

Das

# Statistik

SOFTWARE SELBSTGEMACHT

buch  
zum  
Commodore 64

$$\delta = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Voss

**EIN DATA BECKER BUCH**



Das

# Statistik

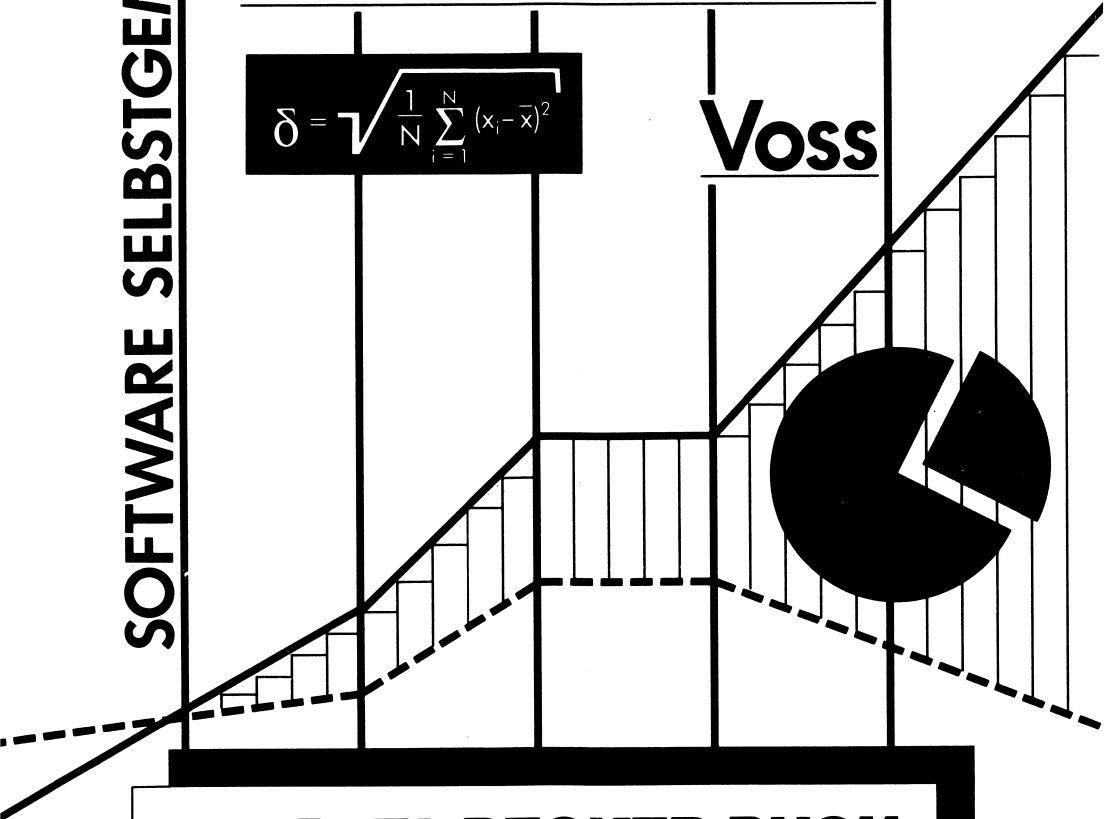
**SOFTWARE SELBSTGEMACHT**

buch  
zum  
Commodore 64

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Voss

**EIN DATA BECKER BUCH**



ISBN 3-89011-102-5

Copyright © 1985 DATA BECKER GmbH  
Merowingerstraße 30  
4000 Düsseldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der DATA BECKER GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

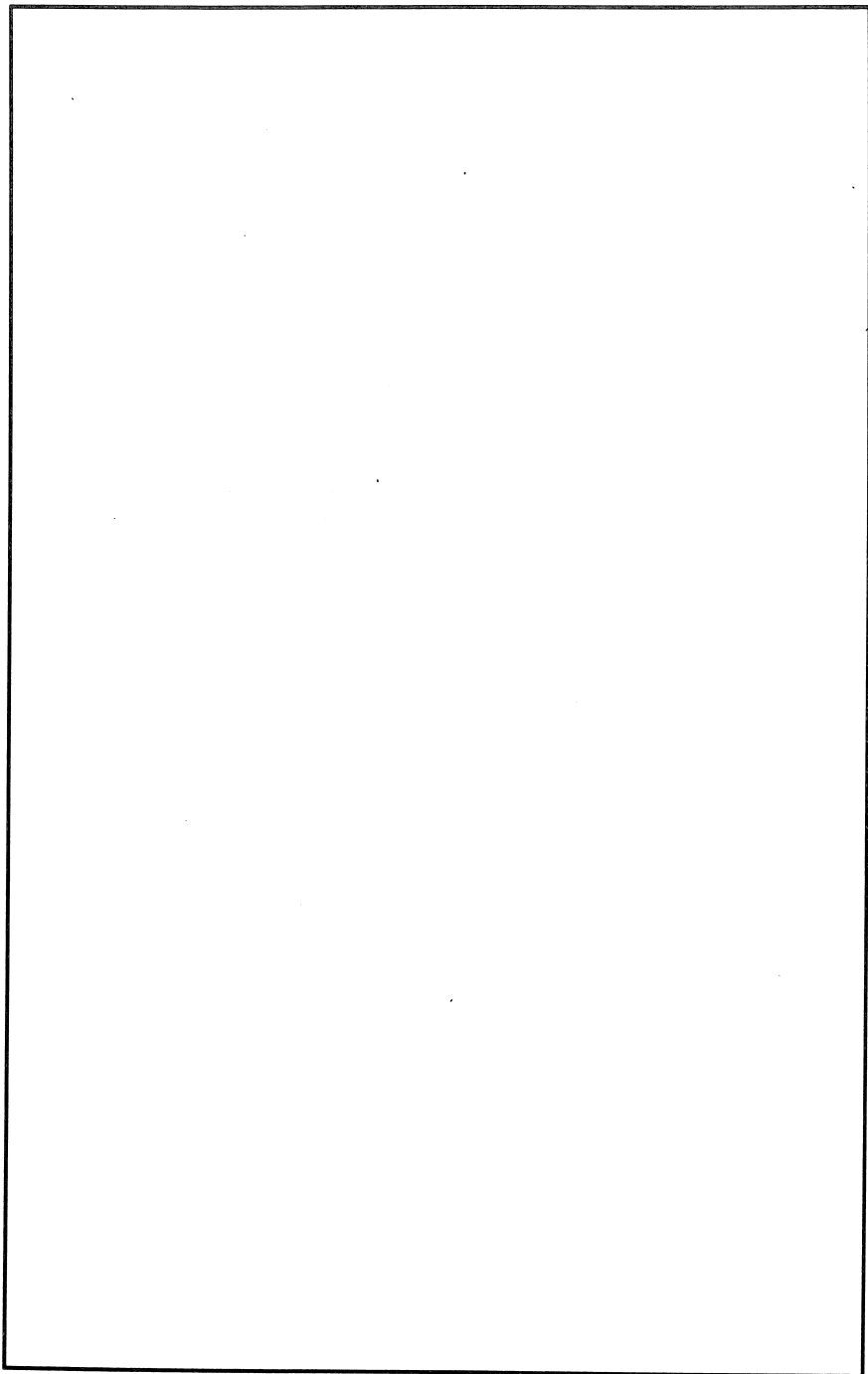
**Wichtiger Hinweis:**

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technischen Angaben und Programme in diesem Buch wurden von dem Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.



Dieses Buch widme ich  
meiner Tochter Sabine,  
die mir mit vielen konstruktiven  
Vorschlägen die Programmierarbeit  
erleichtert hat.



## Inhaltsverzeichnis

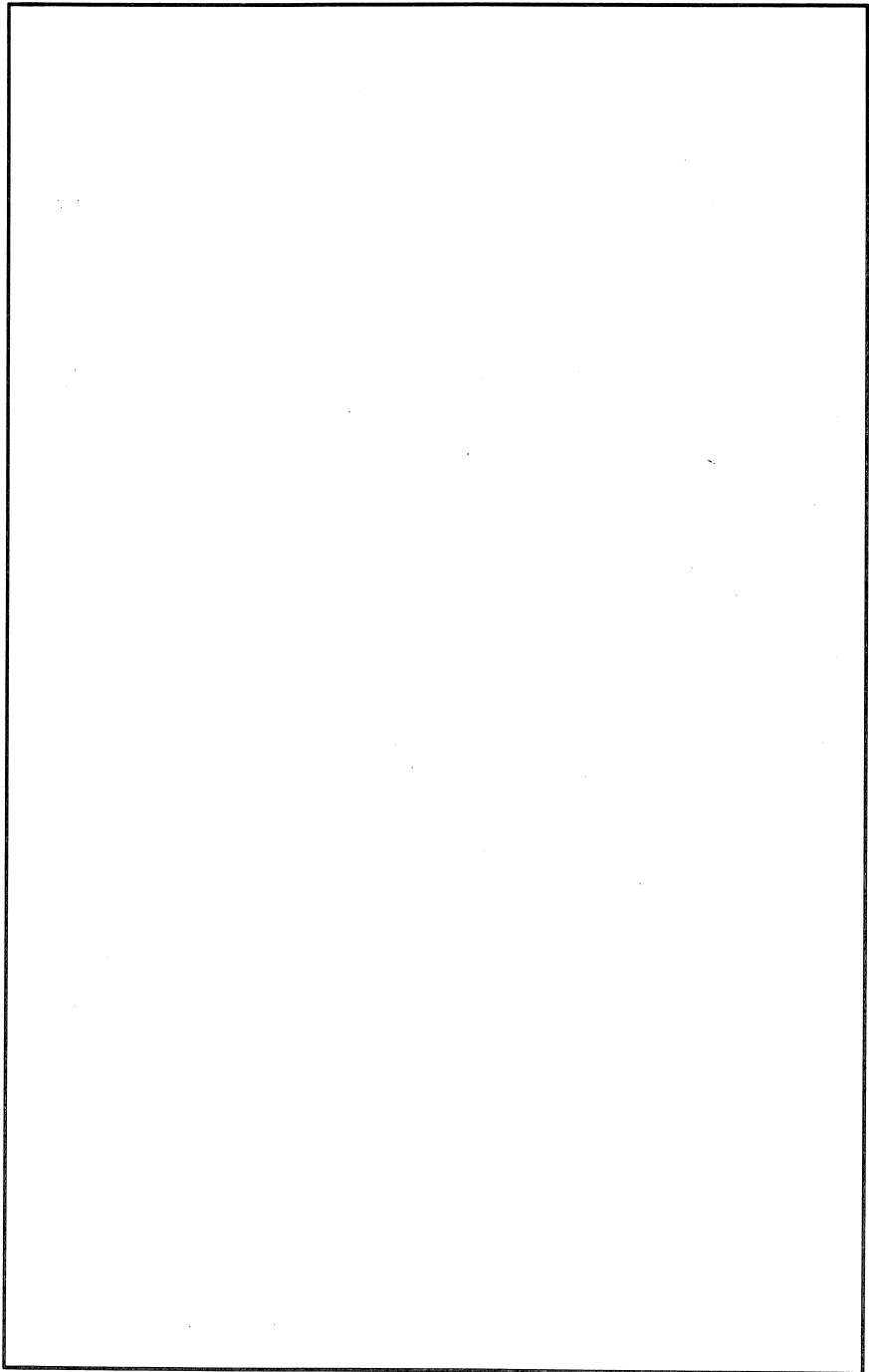
	Seite
Einleitung .....	1
Kapitel 1 : Die Ausgangslage .....	5
1.1 Was ist "Statistik" ? .....	5
1.2 Die Zielsetzungen .....	10
Kapitel 2 : BASIC-Grundlagen .....	19
2.1 Vorbemerkung .....	19
2.2 Ergebnisausgabe .....	20
2.3 Wertzuweisungen .....	23
2.4 Informationseingabe .....	27
2.5 Programmverzweigungen .....	30
2.6 Stringbearbeitung .....	36
2.7 Graphik .....	38
2.8 Diskettenutzung .....	46
Kapitel 3 : Datenaufnahme ohne externe Speicher .....	53
3.1 Vorbemerkung .....	53

3.2	Datengewinnung .....	53
3.3	Dateneingabe .....	58
3.4	Dateneingabe mit LET-Anweisungen ....	59
3.5	Dateneingabe mit INPUT-Anweisungen ..	69
3.6	Dateneingabe über DATA und READ .....	75
3.7	Zählen .....	83
3.8	Ausblick .....	89
Kapitel 4 : Dateneingabe mit externem Speicher .....		91
4.1	Vorbemerkung .....	91
4.2	Speicherung eines neuen Datenbestands .	92
4.3	Dateiverlängerung .....	101
4.4	Generelle Eingabe .....	111
Kapitel 5 : Dateiverwaltung .....		125
5.1	Vorbemerkung .....	125
5.2	Datenausgabe .....	125
5.3	Veränderung von Datenbeständen .....	134
Kapitel 6 : Häufigkeitstabellen .....		153
6.1	Vorbemerkung .....	153
6.2	Statistische Grundlagen .....	154
6.2.1	Verteilung einer diskreten Variablen	155

6.2.2	Verteilung einer stetigen Variablen	157
6.2.3	Verteilung für zwei Variablen .....	158
6.2.4	Multivariate Tabellen .....	159
6.3	Programme .....	160
6.3.1	Sortieren .....	160
6.3.2	Diskrete Häufigkeitstabelle .....	167
6.3.3	Stetige Häufigkeitstabelle .....	174
6.3.4	Bivariate Tabelle (Kreuztabelle) ..	181
Kapitel 7	: Graphische Darstellungen von Häufigkeitsverteilungen ..	189
7.1	Statistische Grundlagen .....	189
7.1.1	Eine diskrete Variable .....	191
7.1.2	Eine stetige Variable .....	193
7.1.3	Zwei Variablen (Streudiagramm) ....	194
7.2	Programme .....	197
7.2.1	Stabdiagramm .....	197
7.2.2	Kreisdiagramm .....	206
7.2.3	Histogramm .....	218
7.2.4	Streudiagramm .....	228
Kapitel 8	: Mittelwerte und Streuungen	235
8.1	Statistische Grundlagen .....	235

8.1.1	Mittelwerte .....	236
8.1.2	Streuungsmaße .....	238
8.2	Programme .....	240
8.2.1	Mittelwerte .....	240
8.2.2	Streuung .....	249
Kapitel 9 : Regressions- und Korrela- tionsrechnung .....		255
9.1	Statistische Grundlagen .....	255
9.1.1	Regressionsrechnung .....	256
9.1.2	Korrelationsrechnung .....	261
9.2	Programme .....	266
9.2.1	Lineare Regression .....	266
9.2.2	Korrelationskoeffizient .....	278
9.2.3	Kontingenzkoeffizient .....	283
Kapitel 10 : Zeitreihenstatistik .....		293
10.1	Statistische Grundlagen .....	293
10.1.1	Begriff und Aufgaben .....	293
10.1.2	Trendprognosen .....	295
10.1.3	Glättungen .....	298
10.2	Programme .....	299
10.2.1	Vorbemerkung .....	299
10.2.2	Methode der halben Durchschnitte .	300

10.2.3	Methode der kleinsten Quadrate	...	316
10.2.4	Reihenglättung	.....	324
Kapitel 11 : Hochrechnungen			..... 339
11.1	Vorbemerkung	.....	339
11.2	Statistische Grundlagen	.....	340
11.3	Schätz-Programm	.....	345
Kapitel 12 : Das Programmpaket			..... 355
12.1	Vorbemerkung	.....	355
12.2	Beschränkungen	.....	356
12.3	Anmerkungen zur Nutzung des Programm- pakets	.....	360
12.4	Genereller Überblick über das Paket		363
12.5	Programmlisten und -beschreibungen		368
12.6	Variablenliste	.....	427
12.7	Programmergebnisse	.....	430
Literaturhinweise			..... 437
Stichwortverzeichnis			..... 439



## Einleitung

Der wesentliche Anstoß zur Entwicklung der Computer, von den riesigen "Elektronengehirnen" der fünfziger Jahre bis hin zu dem modernen und leistungsfähigen Kleinrechner und Homecomputer unserer Tage, kam aus dem Bereich der Datenverarbeitung :

In erster Linie ging es nämlich zunächst einfach nur darum, die immer größer werdenden Datenmengen auf geschickte Weise und möglichst ohne zu große zeitliche Verzögerungen zu verarbeiten. Datenverarbeitung per Hand nämlich war (und ist) mühsam, fehlerträchtig und viel zu zeitaufwendig.

Datenverarbeitung bedeutet nichts anderes als "statistische" Verarbeitung oder anders ausgedrückt : Anwendung geeigneter statistischer Methoden auf vorliegende Datenbestände.

Dies bedeutet, daß Computer von Anfang an vor allem dazu eingesetzt wurden, statistische Auswertungen, die per Hand zu umständlich waren, zu übernehmen. Und selbstverständlich ist dies auch heute noch eines der wichtigsten Einsatzgebiete.

Schaut man sich beispielsweise die Angaben an, die zur Frage der praktischen Nutzung von Rechnern vorliegen, so erkennt man, daß statistische Datenverarbeitungsprozesse nach wie vor eine sehr wichtige Rolle spielen.

Dies gilt auch für Kleinrechner und für Homecomputer. Deshalb ist es eine wichtige Aufgabe, dem Besitzer eines solchen Rechners eine leicht lesbare Anleitung an die Hand zu geben, der er entnehmen kann, wie er seinen Rechner zur Lösung statistischer Aufgabenstellungen einsetzen kann.

Dabei beziehen wir uns in dieser Ausarbeitung auf denjenigen Homecomputer, der in den letzten Jahren dank seiner überzeugenden Leistungen einen bemerkenswerten Marktanteil erworben hat, nämlich auf den

#### Commodore C64

Dies bedeutet, daß die Programmbeispiele dieses Buchs, die ausschließlich in der sehr leicht erlernbaren Programmiersprache BASIC geschrieben sind, Programme für den Commodore C64 sind.

Da jedoch, von wenigen Ausnahmen abgesehen, nur solche BASIC-Sprachelemente verwendet wurden, die auch in anderen "BASIC-Dialekten" Gültigkeit haben, ist ein Umschreiben der Programme auf andere Rechner sicher nicht allzu problematisch (eine Ausnahme bilden die in größerem Umfang verwendeten Graphik-Anweisungen, worauf gesondert aufmerksam gemacht wird).

Darüberhinaus ist anzumerken, daß die grundsätzlichen Ausführungen der folgenden Kapitel zum Aussehen und zur Funktionsweise der verschiedenen statistischen Methoden, die jeweiligen problemanalytischen Überlegungen und die Ausführungen zum Einsatz von Kleinrechnern ganz generell gelten und nicht etwa auf den Commodore C64 beschränkt sind.

Daraus ist zu schließen, daß dieses Buch nicht nur für den Besitzer eines Commodore C64 von Interesse sein dürfte, sondern für alle diejenigen Leser, die sich mit der Frage des Kleinrechner-einsatzes zur Lösung statistischer Aufgaben befassen.

Man könnte nun davon ausgehen, daß das Thema "Computereinsatz in der Statistik" damit kurz abgehandelt werden könnte, daß wir diejenigen statistischen Programme, die es heute in durchaus nennenswerter Zahl schon gibt, vorstellen und ihre Funktionsweise an geeigneten Demonstrationsbeispielen illustrieren.

So einfach wollen wir uns aber das nicht machen. Es geht nämlich in diesem Buch nicht nur darum, einen Beitrag zur Lösung statistischer Probleme der Besitzer von Kleinrechnern zu leisten, sondern um mehr :

Wir wollen gleichzeitig die Leistungsfähigkeit moderner Homecomputer demonstrieren und vor allem dem Leser einen Weg zum Verständnis solcher Geräte aufweisen. Insbesondere geht es dabei darum, zu verdeutlichen, welche gedanklichen Prozesse erforderlich sind, um dazu zu kommen, daß ein Computer für uns die Lösung eines bestimmten Problems übernehmen kann.

Das Grundverständnis für diese so wichtige neue Technologie, die ja Teil der sich rasch verbreitenden Mikroprozessor-Technologie ist, kann sich nämlich nur dann entwickeln, wenn man in die spezifische Denkungsart, die der Umgang mit Computern erfordert, eingeweiht ist, d.h. wenn man sich hinreichend mit ihr vertraut gemacht hat.

Dieses Verständnis erwirbt man nicht dadurch, daß man einfach nur fertige Programme verwendet und sich anschaut, was sie bewirken, sondern in der konkreten Beschäftigung mit Anwendungsproblemen, ihrer gedanklichen Durchdringung, ihrer Umsetzung in eine computeradäquate Form und beispielsweise in der Entwicklung eigener Computerprogramme oder zumindest im gedanklichen Nachvollzug solcher Entwicklungen.

Genau dies ermöglichen die folgenden Kapitel, wobei zusätzlich auch noch die wichtigsten Grundprinzipien statistischer Methoden vorgestellt werden, so daß der Leser aus zwei Gründen verstehen kann, was die präsentierten Programme eigentlich leisten :

1. Er versteht durch die Lektüre des Buchs so viel vom Programmieren in BASIC, daß er die Programme lesen und nachvollziehen kann (gegebenenfalls kann er sie auch nach eigenen Wünschen und Erfordernissen abändern);

2. Er versteht dann auch so viel von den wichtigsten statistischen Methoden, daß er weiß worum es bei den einzelnen Programmen unter den Gesichtspunkten der praktischen Nutzung tatsächlich geht.

So wird es möglich, daß man nicht "hilflos" der nur schwer durchschaubaren Technologie und fremden, meist unverständlichen Programmen ausgeliefert ist, sondern die moderne Technik in Form des Homecomputers gemäß eigener Vorstellungen einsetzen kann und dann auch versteht, was im einzelnen geschieht.

Nur so wird man in die Lage versetzt, zu erkennen, was es mit dieser neuen Technologie auf sich hat, welche Möglichkeiten sie bietet, aber auch wo ihre Einsatzgrenzen liegen.

## Kapitel 1 : Die Ausgangslage.

### 1.1 Was ist "Statistik" ?

Jeder von uns hat täglich mit Statistiken zu tun: Der Fußballfan, der die Bundesligatabelle studiert, die Hausfrau, die Einkaufspreise miteinander vergleicht, der Schüler, der seine Zensuren im Taschenkalender notiert, der Manager, der Aktienkurse verfolgt, die Sekretärin, die jeden Morgen auf die Badezimmerwaage steigt und ihr Gewicht gewissenhaft notiert - sie alle arbeiten statistisch.

Viele Menschen müssen jeden Tag mit großen Mengen an Informationen umgehen, müssen sie zur Kenntnis nehmen, sie verarbeiten und in angemessener Weise darauf reagieren. Je größer nun die Mengen an Informationen werden, die es zu verarbeiten gilt, um so wichtiger werden Grundkenntnisse in statistischen Methoden .

Wer beispielsweise täglich sein Gewicht kontrolliert, hat am Ende eines Jahres 365 Einzelinformationen gesammelt, die insgesamt gesehen kaum noch überschaubar sind. Erst geeignete statistische Verfahren - etwa zur Durchschnittsbildung oder Methoden zur Trendberechnung - lassen den eigentlichen Informationsgehalt dieser 365 Einzeldaten wieder sichtbar werden.

Ein Handwerksmeister, der eine Ertragskontrolle seines Betriebes durchführen will, benötigt geeignete statistische Methoden, um sich aus der großen Zahl von Einzelangaben einen informativen, komprimierten Überblick zu verschaffen.

Der Student, der für seine Examensarbeit eine Umfrage durchführt und dabei 500 Personen je 80 Fragen stellt, braucht statistische Methoden, um aus den 40000 Antworten diejenigen Informationen "herauszuholen", die ihn eigentlich interessieren.

Der Wirtschaftspolitiker, der vor den fast unendlichen "Zahlenfriedhöfen" der statistischen Jahrbücher steht und auf der Grundlage konjunktureller Daten oder auf der Basis von Arbeitsmarktstatistiken Entscheidungen fällen will, muß erst diese Daten mit statistischen Methoden aufbereiten (oder aufbereiten lassen).

Solche und ähnliche Beispiele lassen sich in sehr großer Zahl formulieren. Sie zeigen, daß ohne statistische Methoden der Umgang mit großen Datenbeständen nicht möglich ist, daß also "Statistik" betrieben werden muß.

Was bedeutet nun eigentlich der Begriff "Statistik" ?

Die Fachliteratur bietet uns eine Reihe unterschiedlicher Definitionen an, von denen drei wichtige hier vorgestellt werden sollen, damit der Leser von vornherein weiß, worüber im folgenden gesprochen wird :

"Die Statistik dient dazu, Entscheidungen in Fällen von Ungewißheit zu treffen."

Zur Erläuterung dieser ersten Definition ein einfaches Beispiel :

Der Produktmanager eines Zigarettenkonzerns soll darüber entscheiden, ob eine bisher gut bewährte Zigarettenmarke ab sofort in neuer Verpackung angeboten werden soll. "Ungewißheit" bedeutet bei dieser Entscheidung, daß er nicht weiß, ob ein Umsatzzuwachs erzielt werden kann, wenn er sich für die neue Verpackung entscheidet, oder ob es besser wäre, bei der alten Verpackung zu bleiben.

Es leuchtet unmittelbar ein, wie er seine Ungewißheit verkleinern, vielleicht sogar ganz beseitigen kann : Es müssen Marktuntersuchungen durchgeführt werden, die erkennen lassen, wie die Käufer und eventuelle zukünftige Käufer die unterschiedlichen Verpackungsarten beurteilen, bzw. wie sie in ihrem Kaufverhalten darauf reagieren. Die entsprechenden Methoden stellt die Markt- und Meinungsforschung zur Verfügung, die in den großen Bereich statistischer Methoden einzubeziehen ist.

Die zweite Definition des Begriffs der Statistik, die wir uns anschauen wollen, lautet :

"Statistik dient dazu, Massenerscheinungen zu quantifizieren."

Aus dieser Definition wird deutlich, daß sich der Statistiker nicht für Einzelphänomene interessiert (im Gegensatz zum Beispiel zu einem Arzt, der sich in der Regel ganz speziell mit dem Einzelfall, mit dem Patienten nämlich, der zu ihm zur Untersuchung kommt, befaßt), sondern mit "Massen".

Allerdings ist der Begriff der "Masse", wie er hier benutzt wird, relativ : Schon 10 Elemente können eine "statistische Masse" in diesem Sinne sein, ganz gewiß aber 100, 5000 oder 80000 Elemente.

Unter "Elementen" verstehen wir dabei Personen (bei Befragungen zum Beispiel), Tage oder andere Zeiteinheiten (z.B. bei der Gewichtskontrolle), Nationen (z.B. bei wirtschaftsstatistischen Vergleichen), Zuchtsäue oder Rebstöcke (in der Landwirtschaftsstatistik) und viele andere sog. Merkmalssträger .

"Quantifizieren" bedeutet in diesem Zusammenhang, daß die interessierenden Sachverhalte, Tatbestände oder Entwicklungen mit Zahlen belegt werden sollen, mit statistischen Daten also.

Schließlich noch eine dritte Definition :

"Statistik dient dazu, Daten zu erheben, darzustellen, auszuwerten, zu analysieren und ihre Interpretation vorzubereiten".

Diese Definition beschreibt sehr anschaulich, warum es im einzelnen geht :

Informationen gewinnen wir, indem wir zunächst Daten erheben. Damit sich deren Informationsgehalt aber voll erschließen kann, müssen sie, wie oben schon ausgeführt wurde, in anschaulicher Weise dargestellt (präsentiert) werden; niemandem erschließt sich der Informationsgehalt von 40000 Einzelangaben, wenn diese Daten nicht in überschaubarer Weise dargestellt werden.

Aber auch die beste Art der Datenpräsentation kann noch zu unübersichtlich sein, wenn die Datenbestände zu groß sind. Spätestens dann ist es erforderlich, die Daten in geeigneter Weise auszuwerten. Ein typisches Beispiel dafür ist die Mittelwertberechnung :

Ein Durchschnitt komprimiert in einer einzigen statistischen Zahl die gesamte Information, die in 40000 Einzeldaten vorhanden ist; allerdings gehen bei dieser Durchschnittsbildung auch Einzelinformationen verloren : Informationsgewinn durch zunehmende Übersichtlichkeit wird mit Informationsverlusten erkaufte.

Den eventuellen Auswertungsschritten schließt sich dann die statistische Analyse an : Während Darstellung und Auswertung die Frage beantworten, wie sich ein Datenbestand darstellt, führt die Analyse zur Beantwortung der Frage, warum er sich so darstellt, wie wir ihn vorfinden. Darüber werden wir im Detail bei der Darstellung der einzelnen statistischen Methoden noch sprechen.

Darstellung, Auswertung und Analyse bereiten dann die inhaltliche Interpretation der Befunde vor, bei der es darum geht, die Daten im Hinblick auf die den Statistiker oder seinen Auftraggeber interessierenden Fragestellungen zu beurteilen.

Weitere Anmerkungen zu dieser dritten Definition sind an dieser Stelle nicht erforderlich, wenn man einmal davon absieht, daß es sinnvoll ist, hier schon den mehrfach benutzten Begriff der Daten, dem natürlich in allen weiteren Ausführungen zentrale Bedeutung zukommt, zu erörtern :

Daten sind, ganz allgemein gesprochen, Symbole für Informationen.

In diesem Sinne sind also Daten nicht nur Zahlen, wie man zunächst vielleicht vermuten könnte, sondern es könnten zum Beispiel auch Buchstaben sein, Sonderzeichen irgendwelcher Art, Worte, Texte, Töne u.v.a. Allerdings ist es häufig bei geplanten statistischen Auswertungen sinnvoll, all diese Informationen, sofern dies möglich ist, in Zahlen zu verwandeln, weil dann eine raschere Verarbeitung möglich ist.

Die vorangegangenen Ausführungen verdeutlichen, worum es im folgenden geht : Es soll gezeigt werden, wie statistische Methoden funktionieren, wie und unter welchen Bedingungen sie eingesetzt werden, welche Ergebnisse sie produzieren und vor allem, welche Rolle in diesem Zusammenhang ein Kleinrechner wie zum Beispiel der Commodore C64 spielt.

Es leuchtet unmittelbar ein, daß überall dort, wo große Datenmengen auftauchen und/oder kompliziertere Auswertungsmethoden eingesetzt werden müssen, Computer unschätzbare Dienste leisten können : Wer jemals aus 365 Einzelwerten einen Durchschnitt auszurechnen versuchte, wird dies bestätigen können.

Glücklicherweise sind in den letzten Jahren auch

die kleinsten Rechner schon so leistungsfähig geworden, daß es sich geradezu anbietet, sie für statistische Auswertungen einzusetzen.

## 1.2 Die Zielsetzungen

Die vorangegangenen Ausführungen haben schon auf die wichtigste Absicht, die mit diesem Buch verfolgt wird, aufmerksam gemacht :

Statistische Auswertungen mit Hilfe des C64

Dieser generellen Zielsetzung müssen wir eine etwas einschränkende Anmerkung hinzufügen :

Ein Kleinrechner wie der Commodore C64 ist - trotz seiner noch zu demonstrierenden, bemerkenswerten Fähigkeiten - bezüglich seiner kapazitären Möglichkeiten beschränkt. Riesige Fragebogenaktionen, Volkszählungen, statistische Zeitreihen, die sich über hunderte von Jahren erstrecken, können mit einem solchen Rechner nicht sinnvoll behandelt werden.

Eine entsprechende Einschränkung ist mit Blick auf den Komplexitätsgrad der einzusetzenden statistischen Methoden anzufügen :

Der Bereich der statistischen Auswertungsverfahren und Analysemethoden hat sich in den letzten 20 bis 30 Jahren ungeheuer ausgeweitet und speziell zur Lösung ganz spezifischer und häufig sehr schwieriger Einzelprobleme sind eine Reihe äußerst komplexer und aufwendiger Methoden entwickelt worden, die per Homecomputer kaum bewältigt werden können.

Diese Methoden, die unter den Sammelbegriffen

- "multivariate statistische Methoden",
- "inferenzstatistische Methoden",
- "Wahrscheinlichkeitsstatistik",
- "mathematische Statistik" usw.

in der Fachliteratur auftauchen, sind aber bei der weitaus größten Zahl statistischer Fragestellungen durchaus entbehrlich, d.h. sie werden nur selten benötigt und können deshalb aus unserer weiteren Betrachtung ausgeschlossen werden.

Wir können daraus schließen, daß der Homecomputer-Einsatz immer dann sinnvoll ist, wenn

- a) die Datenfülle nicht zu groß wird,
- b) die Methoden nicht zu kompliziert sind.

Der Leser wird rasch feststellen, daß diese Einschränkungen, die bei den folgenden Kapiteln immer beachtet werden sollten, unter praktischen Gesichtspunkten nicht allzu gravierend sind. Sie betreffen nicht die Tatsache, daß alle statistischen Standardprobleme für Datenmengen, wie sie in der Praxis sehr häufig sind, tatsächlich heute per Homecomputer lösbar sind.

Kleinere statistische Probleme, wie sie zum Beispiel im Haushalt anfallen, Aufgaben, die ein Schüler oder ein Student zu bearbeiten hat, statistische Fragestellungen eines Handwerksbetriebs oder eines kleineren Industrieunternehmens sind allemal mit dem Homecomputer zu bewältigen - und genau solchen statistischen Fragestellungen wenden wir uns im folgenden zu.

Die generelle Zielsetzung, wie sie oben formuliert wurde (statistische Auswertungen mit dem Commodore C64), läßt sich in "Teilziele" zerlegen, bzw. um damit verbundene Teilzielsetzungen ergänzen. Dies wird deutlicher, wenn wir zunächst feststellen, was mit diesem Buch nicht beabsichtigt ist :

Es geht primär nicht darum, statistische Programme für den C64 vorzustellen, die der Leser nur abzutippen braucht, damit er dann seine eigenen Daten statistisch auswerten kann. Allerdings gilt natürlich, daß dieses Ziel - neben anderen - faktisch doch auch erreicht wird, auch wenn es nicht im Mittelpunkt unseres Interesses steht.

Vielmehr geht es darum, dem Leser die Möglichkeit zu bieten,

1. zu verstehen, wie die wichtigsten, grundlegenden statistischen Methoden funktionieren und
2. zu verstehen, welche Schritte vollzogen werden müssen, bevor und damit ein Rechner statistische Arbeiten erfolgreich erledigen kann.

Das erste Argument ist deshalb wichtig, weil nur derjenige, der die Methoden verstanden hat, sachgerecht die Ergebnisse dieser Methoden beurteilen und überhaupt entscheiden kann, welche Methoden bei welchen Fragestellungen einzusetzen sind.

Überdies gilt, daß die einfacheren statistischen Methoden ihrerseits in der Regel die Grundbausteine für komplexere Methoden sind, so daß auch für diese - auch wenn sie hier im einzelnen nicht thematisiert werden - ein gewisses Grundverständnis entstehen wird.

Das zweite Argument ist noch wichtiger :

Wer korrekt die Einsatzmöglichkeiten und -grenzen eines Kleinrechners beurteilen will, und wer sich nicht "hilflos" fremden Programmen ausliefern möchte, der muß nachvollziehen können, wie ein Computerprogramm, das zur Lösung bestimmter Aufgaben eingesetzt werden soll, entstanden ist.

Deshalb sind die statistischen Programme der fol-

genden Kapitel alle so ausführlich dargestellt, daß der Leser beim Nachvollzug der Problembeschreibung, der Problemanalyse, der Programmentwicklung und der Programmbeschreibung erkennen kann, was im einzelnen geschieht.

Nur so wird er übrigens auch in die Lage versetzt, bei Bedarf die hier vorgestellten Programme selbstständig zu verändern, zu ergänzen, zu verbessern oder sie in anderer Weise seinen eigenen speziellen Bedürfnissen anzupassen.

Dieser Zielsetzung entspricht es, daß wir für diese Programme die recht einfache Programmiersprache BASIC verwenden und dabei auf speziellere und schwierigere BASIC-Sprachelemente weitgehend verzichten. Das hat zur Folge, daß die Programme nicht allzu elegant werden, dafür aber überschaubar und leicht nachvollziehbar bleiben. Nicht die größtmögliche Programmeleganz oder größtmögliche Rechengeschwindigkeit sind unsere Zielsetzungen, sondern Verständlichkeit der Programme.

Elegantere Lösungen werden jedem Leser bei den zu erwartenden fortschreitenden Kenntnissen mit hoher Wahrscheinlichkeit selbst gelingen, wenn er daran interessiert ist und sich darum bemüht.

Um dem Einsteiger das Verständnis der Programme zu erleichtern, beschreiben wir jedes der Anwendungsprogramme gemäß des folgenden Schemas :

- o Vorstellung des jeweiligen Problems
- o Problemanalyse
- o Programmliste
- o Liste der im Programm verwendeten Variablen
- o Programmbeschreibung
- o Vorstellung der Programmergebnisse
- o Ergänzungs- und Veränderungsmöglichkeiten

In diesen einzelnen Arbeitsschritten geschieht folgendes :

#### 1. Vorstellung des jeweiligen Problems

Es wird zunächst stichwortartig erläutert, für welche statistische Aufgabenstellung das jeweilige Programm eingesetzt werden kann. Diese Erläuterung bedient sich eines einfachen Anwendungsbeispiels.

#### 2. Problemanalyse

In diesem sehr wichtigen Arbeitsschritt wird klar gestellt, welches die logische Abfolge der Schritte ist, die zur Lösung des jeweiligen Problems erforderlich sind. Nur wenn diese Abfolge korrekt entwickelt worden ist, kann ein entsprechendes Computerprogramm erstellt werden.

#### 3. Programmliste

Selbstverständlich stellen wir dann das BASIC-Programm vor, das als Ergebnis der vorangegangenen Problemanalyse das jeweilige statistische Problem tatsächlich löst.

#### 4. Variablenliste

In jedem Computerprogramm werden zur Kennzeichnung der einzelnen Speicherplätze Variablennamen verwendet, deren Dokumentation wesentlich zum Verständnis eines bestimmten Programms beiträgt.

## 5. Programmbeschreibung

Damit der Leser verstehen kann, was das jeweilige Programm tatsächlich tut, wenn es vom Benutzer gestartet wird, wird es Schritt für Schritt beschrieben, um den Nachvollzug zu ermöglichen.

## 6. Programmergebnisse

Das Verständnis eines gegebenen Programms wird weiterhin dadurch erleichtert, daß wir in diesem Arbeitsschritt darstellen, welche Ergebnisse das jeweilige Programm in einem "Probelauf" produzieren muß.

## 7. Ergänzungen

In diesem letzten Arbeitsschritt schließlich weisen wir darauf hin, in welcher Weise das Programm verändert werden kann, wenn die jeweilige Aufgabenstellung sich von der unseres Demonstrationsbeispiels unterscheiden sollte. Natürlich können dabei nicht alle überhaupt vorstellbaren Alternativen berücksichtigt werden, aber einige der naheliegenden Veränderungswünsche können doch aufgegriffen werden.

Zu den statistischen Auswertungsprogrammen selbst ist noch der folgende Hinweis von Interesse :

In der praktischen statistischen Arbeit bestehen Datenverarbeitungsprozesse häufig nicht nur darin, eine einzige statistische Methode anzuwenden (auch wenn dies sicherlich vorkommt), sondern in der Regel wird ein bestimmter Datenbestand unter mehreren Gesichtspunkten ausgewertet, d.h. es werden mehrere Methoden, unter Umständen sogar sehr viele, sukzessive auf den gegebenen Datenbestand angewandt.

Es ist deshalb angebracht, ein Programmpaket zu entwickeln, welches die wichtigsten statistischen Auswertungsmethoden zusammenfaßt. Dies wird so angelegt werden, daß der Benutzer dieses Pakets je-

weils auswählt, welche Auswertungsmethode er anzuwenden wünscht und diese dann vom Paket bereitgestellt wird.

Die Entwicklung eines solchen Programmpakets ist aber unter den Zielsetzungen, die oben vorgestellt wurden, nicht sehr günstig, weil es so umfangreich wird, daß zumindest der Anfänger es nicht mehr nachvollziehen kann. Viel besser ist es, wenn Schritt für Schritt kleinere Programme präsentiert werden, deren Überschaubarkeit ohne große Schwierigkeiten gewährleistet werden kann.

Deshalb werden in den folgenden Kapiteln zunächst nur kleinere Programme vorgestellt, die in der Regel nur der Bewältigung einer einzigen statistischen Aufgabe dienen, wobei wir übrigens auch demjenigen Leser entgegenkommen, der tatsächlich nur eine einzige statistische Prozedur einsetzen möchte und deshalb auf das Gesamtpaket - zumindest zunächst - gut verzichten kann.

Auf diese Weise erreichen wir, daß man nicht schon von Anfang an "das große Ganze" und alle möglichen Querverbindungen zwischen den verschiedenen Teilprogrammen im Auge haben muß, sondern man kann sich auf die jeweiligen Einzelprobleme konzentrieren und wird erst später, durch die vielen Teilaufgaben dann schon hinreichend trainiert, die Querverbindungen und Koppelungen nachvollziehen können.

Wenn dann diese Einzelprobleme bewältigt sind, werden wir ein "Programmpaket" anschließen, das dann die wichtigsten Auswertungsmethoden zusammenfaßt und dem Leser zur Benutzung anbietet. Der Leser wird dann auch in der Lage sein, die Komponenten dieses Programmpakets nachzuvollziehen.

Zum Abschluß dieses Kapitels noch die folgenden Hinweise :

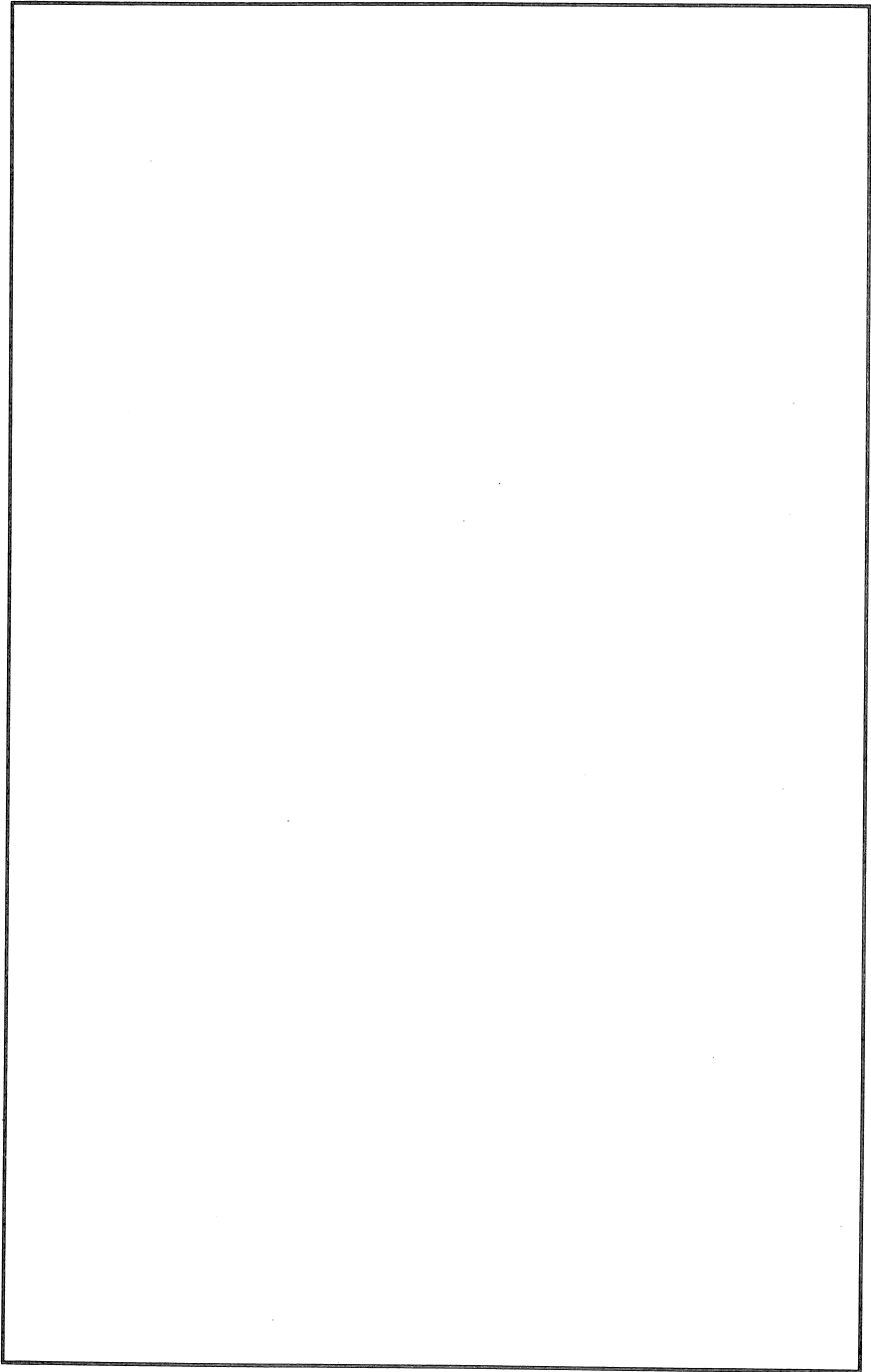
Die Programme wurden nicht so angelegt, daß sie in jedem Fall Eingabefehler des Benutzers "abfangen" können. Wenn ein bestimmtes Programm zum Beispiel

vom Benutzer 4 Zahlen anfordert, dieser aber irrtümlicherweise nur 3 Werte eingibt, dann kommt es zu einer Fehlermeldung und gegebenenfalls zum Programmabbruch. Es ist also wichtig, daß man korrekt den Dialog mit einem Programm abwickelt, bzw. sich mit den Fehlermeldungen des C64-Benutzerhandbuchs vertraut macht.

Entsprechendes gilt für die Programmeingabe : Werden dabei Fehler begangen (und sei es nur, daß an einer bestimmten Stelle ein Komma fehlt oder eines zu viel ist), dann funktioniert das Programm nicht mehr so, wie es soll oder es kommt zu vielleicht unverständlichen Fehlermeldungen. Das Abtippen von Programmen muß also sehr sorgfältig erfolgen oder man besorgt sich einfach die Programmdiskette zu diesem Buch und braucht sich dann um die Fehlervermeidung beim Abtippen keine Gedanken mehr zu machen.

Bevor wir die statistischen Aufgaben in Angriff nehmen und die entsprechenden BASIC-Programme vorstellen, fügen wir ein Kapitel ein, das die Grundelemente der Programmiersprache BASIC vorstellt. Damit soll erreicht werden, daß auch der "Computer-Neuling" die späteren Anwendungskapitel besser bewältigen kann.

Für den Fortgeschritteneren mag dieses Kapitel als eine kurzgefaßte Wiederholung schon bekannter Dinge dienen und der "BASIC-Fachmann" kann es sicherlich überblättern.



## Kapitel 2 : BASIC-Grundlagen

### 2.1 Vorbemerkung

Statistische Auswertungen mit geeigneten Computerprogrammen für den Commodore C64 setzen eine kurze Beschäftigung mit der Programmiersprache BASIC voraus.

BASIC ist derzeit diejenige Programmiersprache, die bei Homecomputern die weiteste Verbreitung erfahren hat und genau dies ist der Grund, warum wir die folgenden Programme in dieser Sprache präsentieren. Allerdings ist mit Nachdruck darauf hinzuweisen, daß vor allem für komplexere Problemstellungen andere Programmiersprachen besser geeignet sind.

Da es aber im folgenden darauf ankommt, verständliche und nachvollziehbare Programme zu präsentieren, bedienen wir uns derjenigen Sprache, die am weitesten verbreitet ist und zudem den Vorteil aufweist, leicht erlernbar zu sein.

Natürlich soll in diesem Kapitel kein kompletter BASIC-Kurs angeboten werden, sondern wir müssen uns mit einem zusammenfassenden Überblick über diejenigen Sprachelemente begnügen, die im folgenden immer wieder benutzt werden.

Wir gehen dabei davon aus, daß der Leser schon über gewisse Grundkenntnisse verfügt, so daß dieses Kapitel quasi der "Auffrischung" dieser Kenntnisse dient. Derjenige Leser, der noch nie in der Sprache BASIC programmiert hat, sollte an dieser Stelle die Lektüre dieses Buches unterbrechen und

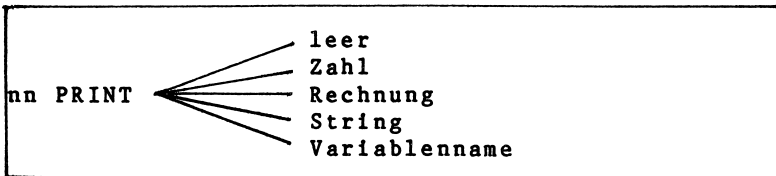
ein einführendes BASIC-Lehrbuch studieren. Derjenige hingegen, der schon perfekter BASIC-Programmierer ist, kann dieses Kapitel übergehen.

## 2.2 Ergebnisausgabe

Jedes BASIC - Programm besteht aus einer Folge von Anweisungen ( Statements ), die dem Rechner zeilenweise einzugeben sind. Eine Programmzeile nennt man Satz .

Jeder Satz in BASIC benötigt eine Satznummer und ist mit der RETURN -Taste abzuschließen. In einem Satz können mehrere Statements stehen, die dann durch Doppelpunkte voneinander zu trennen sind.

Das wesentliche Statement zur Ergebnisausgabe ist das PRINT-Statement, das in allgemeiner Schreibweise folgendermaßen aussieht :



Die verschiedenen Möglichkeiten, die in der allgemeinen Statement-Darstellung genannt wurden (mit Ausnahme der Möglichkeit "leer") können in einem PRINT-Statement mehrfach und auch gemischt verwendet werden. Diese verschiedenen Ausdrücke sind dann durch Kommata oder durch Strichpunkte voneinander zu trennen.

Wird mit Kommata getrennt, so rücken die Ergebnisse auseinander; es erfolgt jeweils ein horizonta-

ler Vorschub in die nächste Bildschirmzone ; wird hingegen der Strichpunkt als Trenner benutzt, geht es unmittelbar weiter.

Jedes PRINT- Statement bewirkt in der Regel einen Zeilenvorschub ; steht das Wort PRINT allein (Möglichkeit "leer"), so erfolgt nur ein Zeilenvorschub (Ausgabe einer Leerzeile). Ein Komma oder ein Strichpunkt am Ende eines PRINT-Statements unterdrückt den Zeilenvorschub.

Die Ergebnisausdrucke, die durch das PRINT-Statement erzeugt werden, beginnen jeweils am linken Bildschirmrand. Will man dies ändern, kann man sich der TAB-Funktion bedienen :

```
10 PRINT TAB(10) 3
```

bewirkt, daß die Zahl 3 zehn Stellen nach rechts verschoben ausgegeben wird.

Jedes BASIC-Programm wird mit dem folgenden Statement beendet :

```
nn END
```

Mit diesen beiden Statements zusammen (PRINT und END) kann man schon erste kleinere Programme entwerfen, die insbesondere zur Lösung von einfachen Rechenaufgaben verwendet werden können (im PRINT-Statement kann ja, wie oben angegeben wurde, auch gerechnet werden).

Dazu ein einfaches Beispiel, das die bisher besprochenen Möglichkeiten illustriert :

```
10 PRINT "ERSTES PROGRAMM"  
20 PRINT  
30 PRINT "3+5 = ";3+5  
40 PRINT  
50 PRINT TAB(4)"ENDE"  
60 END
```

Dieses kleine Programm erzeugt die folgenden Ergebnisse auf dem Bildschirm :

ERSTES PROGRAMM

3+5 = 8

ENDE

Programme werden vom Rechner allerdings erst dann abgearbeitet, wenn sie mit dem folgenden Kommando gestartet werden :

RUN

Für Kommandos gilt die folgende generelle Regel :

Kommandos erhalten keine Satznummer !

Will man ein Programm wieder auf dem Bildschirm erscheinen lassen, zum Beispiel weil es über den oberen Bildschirmrand hinausgewandert ist, so benötigt man das folgende Kommando :

LIST

Häufig ist es sinnvoll, in Programme Kommentare,

Erläuterungen, Zwischenüberschriften einzufügen oder sie mit einer Überschrift zu versehen. Dies ist mit dem folgenden Statement möglich :

```
nn REM Text
```

Dieses REM-Statement taucht mit seinem Text beim Listen eines Programms immer wieder auf, hat aber auf die Programmabarbeitung selbst keinen Einfluß.

Wenn ein Programm im Arbeitsspeicher gelöscht werden soll, so benötigt man das folgende Kommando :

```
NEW
```

Das jeweilige Programm steht nach Eingabe dieses Kommandos nicht mehr zur Verfügung (also ist Vorsicht geboten!).

### 2.3 Wertzuweisungen

Speicherplätze im Rechner müssen mit Namen versehen werden. In diesen Speicherplätzen (Felder) können dann die Werte von Variablen gespeichert werden. Bei der Namensvergabe muß auf den Typ der jeweiligen Variablen geachtet werden.

Wir unterscheiden dabei zwischen numerischen Variablen und Stringvariablen, wobei erstere wiederum in reelle und ganzzahlige Variablen unterteilt werden können. Reelle Variablen haben als Werte Dezimalzahlen (in BASIC verwendet man immer einen Punkt zum Abtrennen von Dezimalstellen), ganzzahlige Variablen hingegen nur ganze Zahlen. Stringvariablen weisen als Ausprägungen Strings (Zeichenketten) auf, die in Anführungszeichen einzuschließen sind.

Der Name einer reellen Variablen besteht aus einem oder zwei alphanumerischen Zeichen, wobei das erste immer ein Buchstabe sein muß. Die Namen ganzzahliger Variablen werden zusätzlich mit dem %-Zeichen, die Namen von Stringvariablen mit dem \$-Zeichen versehen.


In der Statistik ist eine andere Unterteilung üblich, nämlich die in

- diskrete Variablen und
- stetige Variablen

Erstere können als Werte nur endlich viele, streng voneinander getrennte Werte als Ausprägungen annehmen, letztere innerhalb ihres Definitionsbereichs jeden beliebigen Wert.

Beispiel einer diskreten statistischen Variablen ist die Variable "Kinderzahl", Beispiel einer stetigen Variablen ist die Variable "Körpergröße".

Nach diesen Erläuterungen können wir das LET-Statement besprechen, das der Wertzuweisung dient :

<p>nn LET Variablenname =  Zahl Rechnung String Variablenname</p>
--

Das Wort LET darf auch entfallen.

Aus der zweiten Möglichkeit geht hervor, daß mit diesem Statement, wie übrigens auch schon mit dem PRINT-Statement, gerechnet werden kann. Dabei werden die folgenden arithmetischen Operatoren benutzt :

- + Addition
- Subtraktion
- \* Multiplikation
- / Division
- ↑ Potenzierung

Die Rangfolge der Operatoren entspricht der aus der Schulmathematik bekannten; bei Bedarf kann durch geeignete Klammersetzung davon abgewichen werden.

Viele Berechnungen können durch eingebaute Funktionen erleichtert werden. Allgemein sieht ein solcher Funktionsaufruf folgendermaßen aus :

Variablenname = Funktionsname(Argument(e))

Funktionen können nur innerhalb von Anweisungen (z.B. PRINT- oder LET-Anweisung) benutzt werden.

Die wichtigsten dieser Funktionen sind die folgenden :

- SQR(Zahl) : berechnet die Quadratwurzel aus Zahl
- INT(Zahl) : bestimmt den größten, in Zahl enthaltenen ganzzahligen Wert
- RND(0) : bestimmt eine Zufallszahl zwischen 0 bis unter 1
- CHR\$(Zahl) : bestimmt das ASCII-Codezeichen der Codezahl Zahl

Zur zuletztgenannten CHR\$-Funktion soll ein wichtiger Anwendungsfall genannt werden :

## Die Anweisung

```
nn PRINT CHR$(147)
```

kann beim Commodore C64 dazu benutzt werden, im laufenden Programm den Bildschirm zu löschen. Vor jeder Ergebnisausgabe ist dies sehr nützlich.

Auch an dieser Stelle soll ein einfaches Programmbeispiel die bisher vorgestellten BASIC - Elemente erläutern :

```
10 REM PYTHAGORAS
20 LET A = 3
30 LET B = 4
40 LET C = SQR(A*A + B*B)
50 PRINT CHR$(147)
60 PRINT "HYPOTENUSE = ";C
70 PRINT
80 PRINT "ENDE DER AUSGABE":END
```

In diesem Programm wird mit dem Lehrsatz des Pythagoras die Hypotenuse eines rechtwinkligen Dreiecks berechnet, das die Katheten 3 und 4 aufweist. Das Ergebnis erscheint auf dem zuvor gelöschten Bildschirm (Satz 50) in folgender Form :

```
HYPOTENUSE = 5
```

```
ENDE DER AUSGABE
```

In Satz 80 haben wir übrigens von der Möglichkeit Gebrauch gemacht, in einem Satz mehrere Statements (hier zwei Statements) unterzubringen, die durch einen Doppelpunkt voneinander getrennt sind.

## 2.4 Informationseingabe

Konkrete Datenverarbeitungsprobleme erfordern es, daß Daten eingegeben werden. In BASIC eignet sich dazu ganz hervorragend das folgende Statement :

```
nn INPUT ("Text";) Variablenliste
```

Gelangt der Rechner bei der Programmabarbeitung an dieses Statement, so unterbricht er sie und produziert auf dem Bildschirm ein Fragezeichen (vorher den eventuell im Statement vorhandenen Text, der aber auch weggelassen werden kann). Er erwartet dann vom Benutzer so viele Werte (durch Kommata zu trennen), wie Variablennamen in der Variablenliste genannt sind.

Auch dazu ein einfaches Beispiel :

```
10 REM MITTELWERT
20 INPUT "BITTE 3 WERTE : ";A,B,C
30 M = (A+B+C)/3
40 PRINT "MITTELWERT = ";M
50 END
```

Dieses Programm bleibt mit der Meldung

```
BITTE 3 WERTE : ?
```

stehen, so daß der Benutzer dann vor der Notwendigkeit steht, diese drei Werte einzugeben. Hat er sie nacheinander, durch Kommata voneinander getrennt, eingegeben und dann die RETURN-Taste betätigt, kommt die weitere Programmabarbeitung in Gang, die hier nicht näher beschrieben werden muß.

Zu beachten ist, daß der Satz 30 ein LET-Statement darstellt, bei dem wir auf das Schlüsselwort LET verzichtet haben.

Ähnlich wie das INPUT-Statement erwartet auch das folgende Statement eine Informationseingabe, die aber nur aus der Betätigung einer einzigen Taste besteht und nicht durch die RETURN -Taste abgeschlossen werden muß.

```
nn GET Variablenname
```

Zur Eingabe größerer Informationsmengen ist das INPUT-Statement nicht so gut geeignet. Besser geeignet ist das DATA -Statement in Verbindung mit dem READ -Statement :

```
nn DATA Wert1,Wert2,Wert3, ...
```

```
nn READ Name1,Name2,Name3, ...
```

Die Lese - Anweisung READ bewirkt, daß der erste Wert des DATA -Statements in das Feld Name1, der zweite Wert in das Feld Name2, der dritte Wert in das Feld Name3 gelesen wird usw.

Es dürfen in READ-Statements nicht mehr Namen auftauchen als in DATA - Statements Werte vorhanden sind, wohl aber ist es möglich, weniger Werte als vorhanden zu lesen. Eventuell folgende READ-Statements fahren in der (oder den) DATA - Liste(n) dort fort zu lesen, wo die vorhergehende READ-Anweisung aufgehört hat.

Demonstrationsbeispiel :

```
10 DATA 13,12,14
20 READ A,B,C
30 PRINT (A+B+C)/3
40 END
```

Dieses kleine Programm produziert den Wert 13 als Ergebnis.

Bei großen Datenbeständen empfiehlt es sich, unter einem Variablennamen jeweils mehrere Werte zu erfassen und nicht wie bisher immer nur einen Wert. Dies gelingt durch Verwendung sog. indizierter Variablen .

In der Programmiersprache BASIC sieht das folgendermaßen aus :

X(0), X(1), X(2), ... allgemein z.B. X(I)

Bei Verwendung solcher indizierten Variablen muß allerdings dem Rechner vor deren erster Verwendung mitgeteilt werden, wieviele Speicherplätze für die jeweilige(n) Variable(n) freigehalten werden sollen. Dies leistet das folgende Statement :

```
nn DIM Name1(Zahl1),Name2(Zahl2), ...
```

Es kann auch doppelt indiziert werden, also z.B.:

```
10 DIM X(2,3)
```

In diesem Beispiel werden für die Variable X  $3*4 = 12$  Speicherplätze freigehalten, nämlich die Plätze

X(0,0)	X(0,1)	X(0,2)	X(0,3)
X(1,0)	X(1,1)	X(1,2)	X(1,3)
X(2,0)	X(2,1)	X(2,2)	X(2,3)

## 2.5 Programmverzweigungen

Solange in einem BASIC - Programm noch keine Verzweigungen auftreten (sog. Sprünge), erfolgt die Abarbeitung in der Reihenfolge der angegebenen Satznummern. Will man von dieser Reihenfolge abweichen, so sind Programmsprünge erforderlich. Wir unterscheiden unbedingte Sprünge von bedingten Sprüngen.

Der unbedingte Sprung wird mit dem folgenden Statement möglich :

```
nn GOTO mm
```

Erreicht die Programmsteuerung dieses Statement, so erfolgt ein Sprung zum Satz mit der Satznummer mm.

Auch dazu ein einfaches Beispiel :

```

10 REM ENDLOSPROGRAMM
20 I=1
30 Q=I*I
40 PRINT I,Q
50 I=I+1
60 GOTO 30

```

Dieses Programm erzeugt auf dem Bildschirm eine Tabelle von der folgenden Form :

1	1
2	4
3	9
4	16
5	25

usw.

und hört damit erst auf, wenn die Quadratzahlen im Feld Q so groß werden, daß sie vom Rechner nicht mehr gespeichert werden können. Dies ist deshalb ein sog. Endlosprogramm (das der gute Programmierer sinnvollerweise vermeidet), das nur durch die sog. Notbremse abgebrochen werden kann, nämlich durch Betätigung der RUN/STOP-Taste.

Die bedingte Programmverzweigung wird mit dem folgenden Statement ermöglicht :

```
nn IF logische Bedingung THEN Anweisung
```

Hier wird eine Abfrage veranlaßt : Wird die logische Bedingung, die dem Befehlswort IF folgt, erfüllt, so wird die hinter dem THEN stehende Anweisung ausgeführt; ist dies hingegen nicht der Fall, so geht die Programmsteuerung zu dem Satz über, der dem IF-Statement folgt.

Mit dieser Anweisung ist es möglich, eine Endlos-

**schleife, wie sie das vorangegangene Beispiel darstellte, zu vermeiden :**

```
10 I=1
20 Q=I*I
30 PRINT I,Q
40 I=I+1
50 IF I<16 THEN GOTO 20
60 END
```

**Auch hier wird eine Tabelle nach dem obigen Muster erzeugt, die aber mit der Zeile**

```
15          225
```

**beendet wird.**

**Hinter dem Schlüsselwort THEN können noch weitere Anweisungen im gleichen Satz stehen; auch diese werden nur dann ausgeführt, wenn die Bedingung erfüllt ist. Sie sind wieder durch Doppelpunkte voneinander zu trennen.**

**Im IF ... THEN-Statement können sehr gut auch mit Hilfe von String - Variablen logische Bedingungen aufgebaut werden. Dies verdeutlicht das folgende Beispiel :**

```

10 B$ = "SEHR GUT"
20 PRINT B$
30 PRINT
40 INPUT "NOCHMAL (JA ODER NEIN)";A$
50 IF A$ = "JA" THEN GOTO 20
60 PRINT
70 PRINT "ENDE DER AUSGABE":END

```

Dieses Programm druckt so oft den String SEHR GUT auf den Bildschirm, wie der Benutzer auf die Frage des Programms, ob nochmals gedruckt werden soll, mit JA antwortet. Erst wenn er nicht mit JA antwortet, gelangt das Programm über Satz 60 zu Satz 70 und wird damit dann abgeschlossen.

Häufig ist es sinnvoll, daß bestimmte Programmsegmente ganz aus dem eigentlichen Hauptprogramm ausgelagert werden, um dann von diesem aus mehrfach (bei Bedarf) angesprungen zu werden.

Solche ausgelagerten Programmteile nennt man Unterprogramme. Der Sprung vom Hauptprogramm ins Unterprogramm erfordert das folgende Statement :

```
nn GOSUB mm
```

Dieses Statement bewirkt einen Sprung in das Unterprogramm, das mit der Satznummer mm beginnt. Ist dann dieses Unterprogramm abgearbeitet, so muß ein Rücksprung in das Hauptprogramm erfolgen. Zu diesem Zweck wird das Unterprogramm mit dem folgenden Statement beschlossen :

```
nn RETURN
```

Dieses Statement bewirkt, daß ein Rücksprung an diejenige Stelle des Hauptprogramms erfolgt, die

direkt hinter der entsprechenden GOSUB-Anweisung folgt.

Das folgende Programmbeispiel demonstriert diese Möglichkeit schematisch :

```
10 PRINT "DIESES PROGRAMM ZEIGT,"
20 GOSUB 1000
30 PRINT "WIE JEDE ZEILE SEHR"
40 GOSUB 1000
50 PRINT "EINFACH UNTERSTRICHEN"
60 GOSUB 1000
70 PRINT "WERDEN KANN."
80 GOSUB 1000
90 END
1000 REM UNTERPROGRAMM STRICH
1010 PRINT "-----"
1020 RETURN
```

Dieses Programm ergibt auf dem Bildschirm :

```
DIESES PROGRAMM ZEIGT,
-----
WIE JEDE ZEILE SEHR
-----
EINFACH UNTERSTRICHEN
-----
WERDEN KANN.
-----
```

Besonders interessante Möglichkeiten eröffnet das folgende Statement :

```
nn ON Variablenname GOTO m1,m2,m3,...
```

Wird dieses Statement erreicht, erfolgt ein Sprung zum Satz m1, wenn Variablenname den Wert 1 hat, ein Sprung zum Satz m2, wenn Variablenname den

Wert 2 hat, usw. Mit diesem Statement sind also Mehrfachverzweigungen möglich.

In entsprechender Weise funktioniert die folgende Anweisung :

```
nn ON Variablenname GOSUB m1, m2, m3 ...
```

Wir haben weiter oben gesehen, daß durch geeignete Verwendung des IF ... THEN - Statements erreicht werden kann, daß bestimmte Programmteile mehrfach durchlaufen werden können. Man spricht von sog. Schleifen . Derartige Schleifen können mit den folgenden Statements sehr einfach erzeugt werden :

```
nn FOR Name = Anfang TO Ende (STEP Schrittweite)
```

und

```
nn NEXT Name
```

An der Stelle Name steht dabei der Name einer Variablen, die als Laufvariable bezeichnet wird. Sie taucht auch wieder im NEXT-Statement auf und muß vom reellen Typ sein.

Anfang, Ende und Schrittweite können Zahlen, Variablen oder arithmetische Ausdrücke sein.

Gelangt der Rechner an das FOR-Statement, so wird zunächst die Laufvariable mit dem Anfangswert belegt und der Programmteil zwischen dem FOR ...TO- und dem NEXT - Statement wird mit diesem Wert der Laufvariablen durchlaufen.

Bei Erreichen des NEXT - Statements wird die Laufvariable mit dem nächsten Wert, nämlich mit dem aktuellem Wert + Schrittweite belegt und wieder

derselbe Programmteil abgearbeitet, jetzt mit diesem veränderten Wert.

Dies erfolgt so lange, bis bei Erreichen des NEXT-Statements durch weitere Addition der Schrittweite der Endwert überschritten wird. Dann wird die Schleife verlassen und der nach dem NEXT-Statement folgende Satz bearbeitet.

STEP Schrittweite kann im FOR ... TO - Statement entfallen, wenn die Schrittweite gleich eins sein soll.

Das Quadratzahlenbeispiel stellt sich unter Verwendung dieser Statements folgendermaßen dar :

```
10 FOR I=1 TO 15
20 Q=I*I
30 PRINT I,Q
40 NEXT I
50 END
```

## 2.6 Stringbearbeitung

Wir haben schon erwähnt, daß für bestimmte Zwecke Funktionen zur Verfügung stehen. Eine Reihe solcher Funktionen können speziell zur Stringverarbeitung und Stringbearbeitung eingesetzt werden ( String = Zeichenkette). Die wichtigsten dieser Funktionen sollen kurz vorgestellt werden.

CHR\$(Zahl)

Diese Funktion bestimmt das ASCII-Codezeichen, das zur Codezahl Zahl gehört.

VAL(String)

Diese Funktion spaltet die führenden Ziffern aus dem angegebenen String ab.

LEFT\$(String,Zahl)

Diese Funktion schneidet aus String von links beginnend einen Teilstring ab, der so viele Symbole umfaßt, wie durch Zahl angegeben wird.

RIGHT\$(String,Zahl)

Entsprechend schneidet diese Funktion einen Teilstring von rechts beginnend ab.

MID\$(String,Zahl1,Zahl2)

Diese Funktion schneidet aus String beim Symbol Zahl1 beginnend einen Teilstring heraus, der so viele Symbole umfaßt, wie durch Zahl2 angegeben wird.

Für viele Anwendungsbereiche ist auch die folgende Funktion wichtig :

LEN(String)

Diese Funktion bestimmt die Anzahl der Symbole, die in String enthalten sind.

Die letzte Funktion, die hier betrachtet werden soll, ist die folgende :

STR\$(Zahl)

Diese Funktion wandelt Zahl in eine Zeichenkette um.

## 2.7 Graphik

Bei den Programmen der folgenden Kapitel wird auch von den Graphikmöglichkeiten des Commodore C64 Gebrauch gemacht. Hierzu bieten sich zunächst die Graphiksymbole an, die die C64-Tastatur aufweist.

Beispielsweise findet sich auf der Vorderseite der Q-Taste das Zeichen ●. Dieses Zeichen kann mit dem folgenden Programm "gezeichnet" werden :

```
10 PRINT " [SHIFT]    und [Q]  "  
20 END
```

Der Satz 10 erscheint auf dem Bildschirm folgendermaßen :

```
10 PRINT " ● "
```

Entsprechend können die anderen Graphiksymbole der Tastatur verwendet werden. In Verbindung mit der TAB - Funktion und unter Verwendung von Programmschleifen oder von Unterprogrammen können auf diese Weise Striche, Balken, Umrandungen u.ä. erzeugt werden.

Häufig ist es sinnvoll, wenn bei dieser Art der Graphikprogrammierung bestimmte Stellen (Positionen) des Bildschirms gezielt angesprochen werden können. Dies ist sehr leicht folgendermaßen möglich :

Man muß sich den Bildschirm aufgeteilt denken in 40 Spalten (von links nach rechts 0 bis 39) und 25 Zeilen (von oben nach unten 0 bis 24). Somit ergeben sich in dieser sog. Normalgraphik 1000 Bildschirmpositionen (diese Normalgraphik ist

von der sog. hochauflösenden Graphik zu unterscheiden, über die weiter unten kurz gesprochen wird).

Diese 1000 Positionen werden zeilenweise durchnummeriert von der Adressennummer 1024 links oben bis zur Adresse 2023 rechts unten. Jede dieser Adressen, d.h. jede einzelnen Bildschirmposition kann nun mit einem der Graphikzeichen oder mit irgendeinem der anderen Symbole belegt werden. Dazu benötigt man die folgende Anweisung :

nn POKE Adresse,Zahl

Unter Adresse steht eine Zahl zwischen 1024 und 2023 (s.o.), unter Zahl steht die Bildschirm-Codezahl desjenigen Zeichens, das in der entsprechenden Bildschirmposition erscheinen soll (siehe C64-Handbuch, S. 133/134).

Die Adresse einer beliebigen Bildschirmposition findet man mit der folgenden Formel :

$$\text{Adresse} = 1024 + S + 40 * Z$$

Dabei ist S die Spalte und Z die Zeile, in der das Symbol erscheinen soll.

Will man also beispielsweise in der Bildschirmposition, die durch Spalte 20 und Zeile 10 gekennzeichnet wird, ein Kügelchen zeichnen (Codezahl 81), so erhalten wir die Adresse wie folgt :

$$\text{Adresse} = 1024 + 20 + 40 * 10 = 1444$$

und das Zeichenprogramm lautet dann :

10 POKE 1444,81  
20 END

Zusätzlich kann man bei solchen "Zeichnungen" auch die Farbe der Ausgabe beeinflussen :

Den Bildschirmadressen (1024 bis 2023) sind Farb-  
adressen zugeordnet, die die Nummern 55296 bis  
56295 aufweisen. Hat man beispielsweise die Bild-  
schirmadresse 1444 ausgewählt, so ergibt sich die  
zugehörige Farbadresse , indem man rechnet :

$$\text{Farbadresse} = \text{Bildschirmadresse} + 54272$$

also :

$$\text{Farbadresse} = 1444 + 54272 = 55716$$

Die "Einfärbung" einer bestimmten Bildschirmadres-  
se erfolgt dann mit der folgenden Anweisung :

nn POKE Farbadresse,Zahl

An der Stelle Zahl steht ein Wert zwischen 0 und  
15 mit folgender Bedeutung :

Zahl	Farbe	Zahl	Farbe
0	schwarz	8	orange
1	weiß	9	braun
2	rot	10	hellrot
3	türkis	11	grau 1
4	violett	12	grau 2
5	grün	13	hellgrün
6	blau	14	hellblau
7	gelb	15	grau 3

Abschließend sei darauf hingewiesen, daß auch innerhalb von PRINT-Anweisungen ein Farbwechsel bei der Ausgabe von Informationen erzielt werden kann. Dies gelingt unter Benutzung der Zifferntasten zusammen mit der **CTRL**-Taste bzw. mit der **G**-Taste.

Diese Tastenbenutzung muß innerhalb von Anführungszeichen erfolgen.

Dabei bedeuten **CTRL** und **1** bis **CTRL** und **8** das gleiche wie die Zahlen 0 bis 7 in der obigen Farbtabelle, bzw. **G** und **1** bis **G** und **8** das gleiche wie die Zahlen 8 bis 15 in der obigen Tabelle.

Für statistische Anwendungsprobleme ist es sinnvoll, auch einen Blick auf die sog. hochauflösende Graphik zu werfen, mit deren Hilfe es möglich ist, feinere Linien und Kurvenverläufe zu zeichnen.

Man muß sich dabei den Bildschirm aufgeteilt denken in 320 Spalten (0 bis 319 von links nach rechts) und 200 Zeilen (0 bis 199 von oben nach unten). Die so entstehenden 64000 Punkte sind im Prinzip einzeln ansprechbar.

Bei vielen Kleinrechnern stehen zur Programmierung der hochauflösenden Graphik, also zur Ansprache dieser großen Zahl von Einzelpunkten, hilfreiche

Programmanweisungen zur Verfügung, mit deren Hilfe es möglich ist, Einzelpunkte oder auch ganze Linienzüge (Geraden, Kreise u.ä.) auf dem Bildschirm zu zeichnen.

Der Commodore C64 verfügt über derartige Anweisungen leider nicht. Deshalb gibt es nur die folgenden beiden Wege, hochauflösende Graphiken zu programmieren :

Entweder man programmiert tatsächlich Punkt für Punkt die Linienzüge, aus denen sich die gewünschte Graphik zusammensetzt (sicherlich ein recht aufwendiges Verfahren) oder man bedient sich leistungsfähiger Graphik - Hilfsprogramme, die es zu erschwinglichen Preisen zu kaufen gibt.

Da wir davon ausgehen, daß die hier vorzustellenden Programme auch von denjenigen benutzt werden können, die über keine zusätzliche Software, also über keine Hilfsprogramme verfügen, soll der zweite Weg hier nicht weiter verfolgt werden.

Dies bedeutet, daß wir uns zunächst damit beschäftigen müssen, wie man die einzelnen Bildschirmpunkte per Programm ansprechen kann. Wenn dieses Problem nämlich gelöst ist, dann kann man auch Linien, die sich ja aus einzelnen Punkten letztlich zusammensetzen, programmieren, und damit ist man dann auch in der Lage, komplexere Graphiken zu erzeugen.

Bei der Darstellung der notwendigen Programmierschritte werden wir allerdings nicht alle Einzelheiten im Detail erklären, sondern skizzieren nur die Grundzüge dieser Art der Graphik - Programmierung.

Für die hochauflösende Graphik ist beim Commodore C64 der sog. Videochip zuständig, dessen erste Speicheradresse den Wert 53248 aufweist. Diesen Wert weisen wir, wie es in der Standardliteratur zum C64 üblich geworden ist, der Variablen V zu, also :

V = 53248

Der erste Schritt, der bei der Erstellung einer hochauflösenden Graphik erforderlich ist, besteht in der "Umschaltung" vom sog. Textmodus (in dem wir uns bislang immer befunden haben) in den Graphikmodus. Dies geschieht mit der folgenden Anweisung :

```
POKE V+17,59
```

Zusätzlich muß die Speicherstelle V+24 mit dem Wert 24 belegt werden, also :

```
POKE V+24,24
```

Sind diese beiden Anweisungen abgeschickt, dann verschwindet der auf dem Bildschirm eventuell vorhandene Text und wird durch ein Graphikmuster ersetzt, und zwar durch den jeweiligen Inhalt des Graphikspeichers (Adressen 8192 bis 16383). Dieser Speicher muß nun gelöscht werden, bevor eine neue Graphik erzeugt werden kann :

```
FOR A = 8192 TO 16383  
POKE A,0  
NEXT A
```

Wenn diese Anweisungsfolge ausgeführt wird, verschwindet das Graphikmuster auf dem Bildschirm, was eine Weile dauert und in Ruhe beobachtet werden kann.

Auch bei der hochauflösenden Graphik können die Farbmöglichkeiten des C64 ausgenutzt werden, wobei man auf die Bildschirmspeicheradressen 1024 bis 2023 zurückgreift.

Die Angabe der Hintergrundfarbe (HF) und der Punktfarbe (PF; Farbe der Zeichnung) erfolgt gemeinsam in einer einzigen POKE - Anweisung. Diese POKE-Anweisung sieht generell folgendermaßen aus :

```
POKE A, HF+16*PF
```

Als Codezahlen für die einzelnen Farben werden die Werte der weiter oben vorgestellten Farbtabelle verwendet.

Schreiben wir also beispielsweise :

```
FOR A = 1024 TO 2023
POKE A, 14
NEXT A
```

so wird der Bildschirm die Zeichenfarbe schwarz (Codezahl PF = 0) und die Hintergrundfarbe hellblau (Codezahl HF = 14) annehmen.

Diese quasi vorbereitenden Arbeiten haben wir in einem kleinen Unterprogramm zusammengefaßt, das in jedem Beispiel mit hochauflösender Graphik auftauchen wird. Es hat die folgende Form :

```
1900 REM UP SCREEN
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FOR I=1024 TO 2023:POKE I,14:NEXT I
1940 FOR I=8192 TO 16383:POKE I,0:NEXT I
1950 RETURN
```

Zum Zeichnen eines einzigen Punktes dient das folgende Unterprogramm, das wir, wegen der etwas komplizierteren mathematischen Hintergründe, nicht näher erläutern wollen :

```

2000 REM UP PUNKT
2010 XK = 8*INT(X/8)
2020 YK = 320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX = 2 ↑ (7-INT((X/8-INT(X/8))*8))
2040 S = 8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN

```

Auch dieses Unterprogramm taucht in jedem der folgenden Programme, in denen hochauflösende Graphik benutzt wird, auf.

Die Lage des jeweils zu zeichnenden Punktes auf dem Bildschirm wird durch zwei Werte charakterisiert, nämlich durch X (0 bis 319) und Y (0 bis 199). Ein solches Wertepaar muß also vorgegeben sein, bevor per GOSUB-Statement das Unterprogramm 2000 angesprungen werden kann.

Das Unterprogramm transformiert diese beiden Werte dann so, daß der Punkt an der gewünschten Stelle gezeichnet wird.

Springt man dieses Unterprogramm mehrfach an, etwa ausgehend von einer Schleife, in der die X- und die Y- Werte schrittweise verändert werden, dann kann damit ein Linienzug erzeugt werden.

Nachdem man eine hochauflösende Graphik erzeugt hat, muß wieder vom Graphikmodus in den Textmodus "zurückgeschaltet" werden. Dies leistet das folgende kleine Unterprogramm, das als drittes Unterprogramm in allen derartigen Graphikprogrammen auftaucht :

```

2100 REM UP TEXT
2110 POKE V+17,155:POKE V+24,21
2120 RETURN

```

**ACHTUNG :**

Wenn in einem Graphikprogramm ein Fehler passiert, dann stoppt das Programm im Graphikmodus, d.h. die eventuelle Fehlermeldung und auch weitere Eingaben des Benutzers sind nicht lesbar. Aus diesem Graphikmodus gelangt man zurück in den Textmodus, indem man die Tasten RUN/STOP und RESTORE gleichzeitig drückt, oder indem man die folgenden Anweisungen blind eintippt :

POKE 53248+17,155            und RETURN-Taste  
POKE 53248+24,21            und RETURN-Taste

## 2.8 Diskettennutzung

Man kann externe Speicher zur Lösung unterschiedlicher Aufgaben verwenden, beispielsweise um Programme auf Dauer aufzuheben oder auch, um Datenbestände zu speichern. Konkrete Datenverarbeitungsprozesse werden sich dieser Möglichkeiten in der Regel bedienen.

Deshalb soll in diesem letzten Abschnitt über die Nutzung externer Speicher gesprochen werden, wobei wir uns ausschließlich auf die Diskettennutzung beschränken. Der Grund dafür ist die weite Verbreitung der Diskettenlaufwerke und auch die Tatsache, daß etwa die Benutzung von Kassetten als externe Speicher keine prinzipiell anderen Probleme aufwirft.

Weiterhin beschränken wir uns in diesem Abschnitt darauf, nur die wichtigsten Möglichkeiten der Speichernutzung zu besprechen, gehen also nicht auf alle Details und auf alle möglichen Spezialitäten ein.

Prinzipiell sind die beiden folgenden Nutzungsarten voneinander zu unterscheiden :

- Speichern und Laden von Programmen
- Interaktive Nutzung vom laufenden Programm aus.

Für die erste Nutzungsart sind eine Reihe von Kommandos wichtig.

Fabrikneue Disketten müssen zunächst formatiert werden, was durch das folgende Anweisungspaar erreicht wird :

```
OPEN 15,8,15
PRINT#15,"NEW:name,zahl"
```

An der Stelle name kann der Benutzer seiner Diskette einen Namen zuweisen (z.B. STATISTIK, an der Stelle zahl einen zweistelligen Wert, z.B.01).

Will man ein im Arbeitsspeicher befindliches Programm auf der formatierten Diskette speichern, so benötigt man das folgende Kommando :

```
SAVE "name",8
```

An der Stelle name ist jetzt ein Programmname(z.B. TREND) einzugeben.

Das Einlesen eines Programms von der Diskette in den Arbeitsspeicher des Rechners erfordert das folgende Kommando :

```
LOAD "name",8
```

Will man sich zunächst das Inhaltsverzeichnis der Diskette ansehen, also kontrollieren, was auf der jeweiligen Diskette schon vorhanden ist, so benötigt man die folgenden beiden Anweisungen :

LOAD "\$",8  
LIST

Achtung : Wenn das Disketteninhaltsverzeichnis geladen wird, geht ein eventuell im Arbeitsspeicher des Rechners vorhandenes Programm verloren. Also Vorsicht !

Zum Löschen von Dateien verwenden wir die folgenden Kommandos :

OPEN 15,8,15  
PRINT#15,"S:name"

Wichtiger als diese Art der Diskettennutzung ist bei der praktischen Datenverarbeitung, etwa im Rahmen statistischer Auswertungen, die interaktive Nutzung.

Wir verstehen darunter den Zugriff auf die Diskette vom laufenden Programm aus, entweder, um von der Diskette die zu verarbeitenden Daten zu entnehmen (zu lesen), oder um auf der Diskette irgendwelche Daten (oder Verarbeitungsergebnisse) zu speichern (auf die Diskette zu schreiben).

Man muß dabei prinzipiell zwei Speicherungsarten voneinander unterscheiden :

- sequentielle Speicherung
- random-access-Speicherung

Bei der ersten Speicherungsart wird einfach Wert hinter Wert, Satz hinter Satz auf der Diskette gespeichert, unabhängig etwa von der Länge der jeweiligen Datensätze. Im zweiten Fall beansprucht jeder Datensatz einen gleich großen Speicherbereich, was für den späteren eventuellen Zugriff auf einzelne Datensätze von großem Vorteil ist.

Da aber die random - access - Speicherung etwas schwieriger zu durchschauen ist und wir für die Zwecke dieses Buches mit der sequentiellen Speicherung sehr gut auskommen, wollen wir uns nur mit dieser befassen.

Dateien, die über sequentielle Speicherung von einem laufenden Programm aus auf die Diskette geraten sind, weisen im Inhaltsverzeichnis (s.o.) den Zusatz SEQ auf, während Programme, die per SAVE-Kommando auf die Diskette geraten sind, den Zusatz PRG aufweisen. Nur letztere können per LOAD - Kommando wieder gelesen werden; auf SEQ - Dateien hingegen kann auch nur wieder von einem laufenden Programm aus zugegriffen werden.

Will man nun vom laufenden Programm aus eine sequentielle Datei auf Diskette anlegen, benötigt man die folgenden Programmanweisungen (wir wählen hier schon konkrete Beispiele; zur allgemeinen Schreibweise und zur Bedeutung der einzelnen Parameter in dieser allgemeinen Schreibweise vergl. das Commodore Floppy-Disk-Bedienungshandbuch, Seite 17 ff.) :

```
OPEN 2,8,2,"DATEN,S,W"
```

Hierbei ist die erste Zahl (2) die sog. logische Dateinummer, die Zahl 8 ist die Gerätenummer des Laufwerks und die zweite Zahl ist eine Sekundäradresse (eine Zahl zwischen 2 und 14).

In Anführungszeichen eingeschlossen wird der Name der Datei (hier : DATEN), der Hinweis darauf, daß es sich um eine sequentielle Datei handeln soll (Buchstabe S) und der Hinweis darauf, daß geschrieben (gespeichert) werden soll (Buchstabe W).

Alle nun folgenden PRINT-Anweisungen schicken ihre Informationen nicht auf den Bildschirm, sondern in eine sequentielle Datei auf der Diskette, die den Namen DATEN trägt, wenn sie jeweils mit dem Zusatz #2 (log.Dateinummer) versehen werden, also z.B.:

PRINT#2,A\$,B\$

Der Inhalt der beiden Stringfelder A\$ und B\$ wird nun hintereinander auf der Diskette gespeichert.

Ist der Vorgang der Speicherung beendet, muß die geöffnete Datei wieder geschlossen werden, wofür das folgende Statement dient :

CLOSE 2

(es genügt, als Parameter die logische Dateinummer anzugeben; alle weiteren Parameter sind bei der CLOSE-Anweisung entbehrlich).

Häufig ist es sinnvoll, an eine schon existierende Datei weitere Informationen "anzuhängen" - etwa beim Fortschreiben einer statistischen Gewichtstabelle von Woche zu Woche. Wenn diese Datei, die verlängert werden soll, beispielsweise den Namen DATEN (s.o.) trägt, dann gelingt die Verlängerung mit den folgenden Anweisungen :

```
OPEN 2,8,2,"DATEN,S,A"  
PRINT#2 .....  
....
```

CLOSE 2

Der einzige Unterschied zu den bisher benutzten Anweisungen besteht darin, daß als letzter Parameter in der OPEN-Anweisung der Buchstabe A (für APPEND = Anhängen) auftaucht.

Zum interaktiven Lesen einer sequentiellen Datei benutzen wir entsprechend die folgenden Anweisungen :

```
OPEN 2,8,2,"DATEN,S,R"  
INPUT#2 ....  
....
```

```
CLOSE 2
```

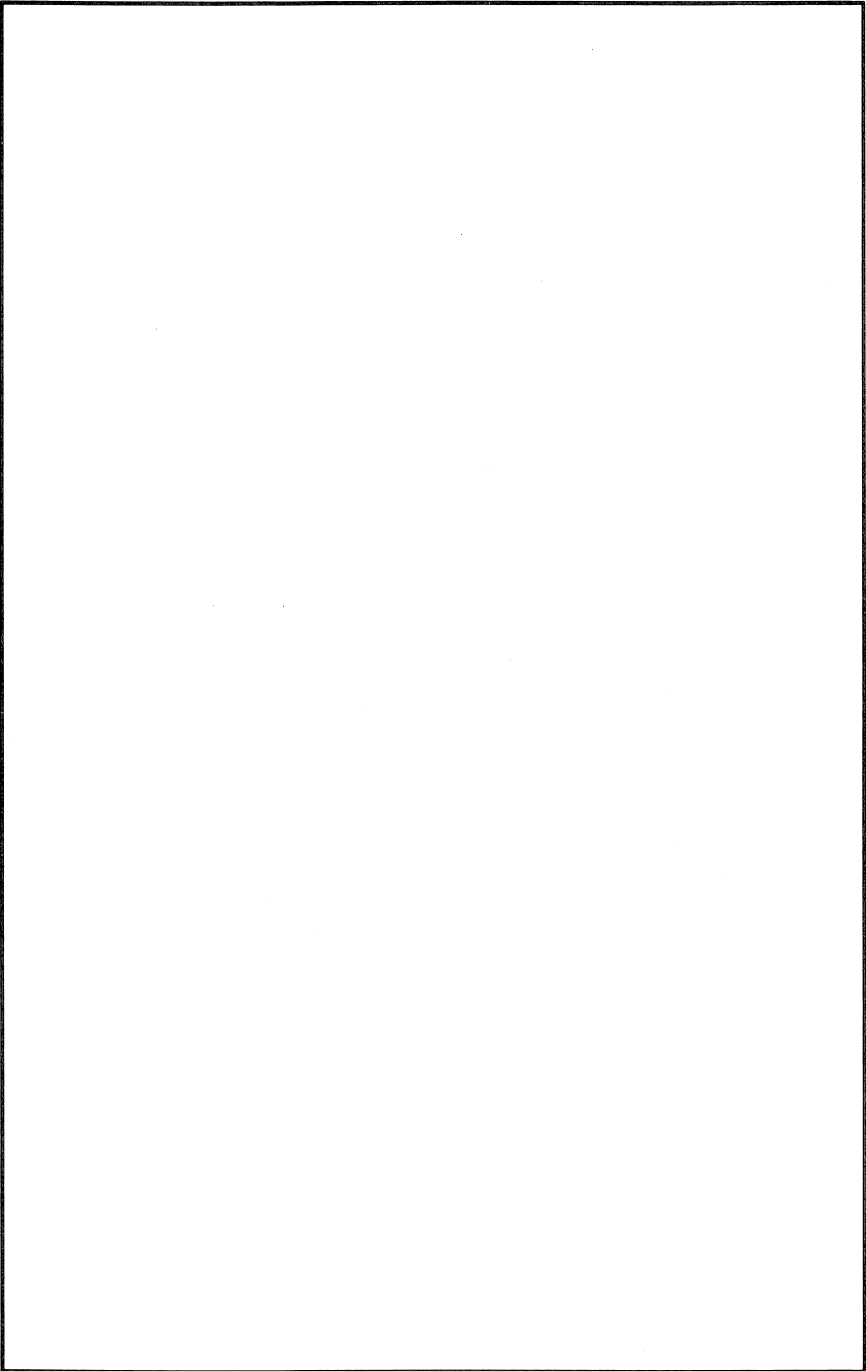
Schließlich ist anzumerken, daß es ab und zu sinnvoll ist, wenn der Name der Datei, auf die zugegriffen werden soll (schreibend oder lesend) erst im laufenden Programm bestimmt wird. Dies gelingt schematisch folgendermaßen :

```
INPUT "BITTE NAME DER DATEI : ";N$  
OPEN 2,8,2,N$+",S,W"  
PRINT#2 ....  
....
```

```
CLOSE 2
```

An der Stelle des Buchstabens W (für Schreiben), kann dabei natürlich auch wieder A (für Anhängen) oder R (für Lesen) stehen.

Auf weitere Einzelheiten soll an dieser Stelle verzichtet werden. Bei der Präsentation der einzelnen Anwendungsbeispiele in den folgenden Kapiteln hat der Leser hinreichend Gelegenheit, die Funktionsweise dieser Anweisungen zu studieren.



## Kapitel 3 : Datenaufnahme ohne externe Speicher

### 3.1 Vorbemerkung

Ein statistischer Auswertungsprozeß beginnt damit, daß die auszuwertenden Daten bereitgestellt werden müssen.

Wir könnten natürlich in einem Buch, welches statistische Programme präsentiert, auch davon ausgehen, daß sich irgendwelche Daten schon im Arbeitsspeicher des Rechners befinden, und bräuchten uns dann nur noch die Auswertungsprozeduren selbst anzuschauen. Dies aber ist nicht so sinnvoll.

Vielmehr ist zu unterstellen, daß derjenige, der eine bestimmte statistische Auswertungsaufgabe mit einem Kleinrechner, wie dem Commodore C64, erledigen möchte, tatsächlich zunächst vor der Aufgabe steht, seine Daten in den Rechner "hinein zu bekommen". Wie dabei sinnvollerweise vorzugehen ist, soll in diesem und im folgenden Kapitel besprochen werden.

### 3.2 Datengewinnung

Wie erwähnt, beginnt die Arbeit des Statistikers mit der Datenaufnahme oder Datengewinnung.

Unter dem Begriff "Daten" versteht man ganz allgemein "Symbole für Informationen".

Wir müssen uns nun mit der Frage befassen, welche vorbereitenden Überlegungen erforderlich sind, um schon die Datengewinnung so zu gestalten, daß die spätere Computerauswertung möglichst erleichtert wird.

Diese Überlegungen erfordern große Sorgfalt, weil sie es ermöglichen, eine Reihe von teilweise sehr schwierigen Problemen, die gegebenenfalls bei der Erstellung der Auswertungsprogramme auftauchen, von vornherin zu vermeiden.

Wie man - mit Blick auf die spätere automatisierte Datenauswertung - die Daten so aufbereitet, daß die Datenverarbeitungsprozesse optimal ablaufen können, soll anhand eines einfachen Beispiels erläutert werden, das zudem die Gelegenheit bietet, die in diesem Zusammenhang wichtigen Grundbegriffe kurz zu klären.

Man stelle sich vor, wir hätten eine statistische Fragebogenaktion gestartet, wobei im Fragebogen unter anderem die folgenden Fragen auftauchen :

1. Bitte Ihr Geburtsjahr . . . .
2. Geschlecht männlich o  
weiblich o
3. Wohnort . . . . .
4. Familienstand ledig o  
verheiratet o  
verwitwet o  
geschieden o
5. Kinderzahl ..
- ...

Wenn nun die ausgefüllten Fragebogen zurückkommen, können wir in einem ersten Arbeitsschritt den Datenbestand folgendermaßen aufschreiben :

Nr.	Geburts- jahr	Geschlecht	Ort	Fam.- stand	Kinder- zahl
001	1942	männlich	Bonn	verheiratet	2
002	1954	weiblich	Köln	ledig	0
003	1961	männlich	Bochum	geschieden	1
004	1955	männlich	Mainz	verheiratet	1

usw.

Diese Daten werden nun sinnvollerweise so transformiert, daß sie einer Datenverarbeitungsanlage in einfacher Form zugänglich gemacht werden können. Diese Transformation geschieht mit Hilfe des sog. Codebuchs , das schematisch das folgende Aussehen hat :

Frage	Inhalt	Werte	Name	Pos.	Typ	m.v.
-	laufende Nummer	001 bis ???	NR	1-3	num.	-
1	Geburtsjahr	1880-1984	GJ	4-7	num.	9999
2	Geschlecht	m = 0 w = 1	GE	8	num.	9
3	Wohnort	Texte	WO	9-20	String	XXX
4	Fam. stand	ledig=0 verh.=1 verw.=2 gesch.=3	FS	21	num.	9
5	Kinderzahl	Zahlen	KZ	22-23	num.	99

Die Inhalte dieses Codebuchs bedürfen keiner weiteren Erklärung, bis auf die letzte Spalte, die mit m.v. überschrieben wurde. Diese Abkürzung steht für "missing value" (fehlender Wert). Hier wird nämlich angegeben, welcher Wert dem Rechner für die Auswertungsprozeduren in dem Fall angegeben werden soll, wenn ein Befragter bei einer bestimmten Frage die Antwort verweigert hat (oder einfach vergessen hat, anzukreuzen).

Auswertungsprogramme müssen solche Fälle identifizieren können, um dann in geeigneter Weise auf die fehlenden Werte reagieren zu können.

Mit Hilfe des Codebuchs kann die ursprüngliche Datentabelle in eine computeradäquate Form gebracht werden, wobei wir zu dem folgenden Ergebnis gelangen :

00119420BONN	12
00219541KOELN	00
00319610BOCHUM	31
00419550MAINZ	11

usw.

Auf diese Weise erhalten wir einen Datenbestand, der in dieser Form direkt einem Rechner zugänglich gemacht werden kann. Diese Form der Datenaufzeichnung bezeichnen wir als Datenmatrix .

Diese Matrix hat so viele Spalten, wie wir Variablen im Datenbestand haben (in unserem Beispiel handelt es sich um 6 Variablen, nämlich um die laufende Nummer NR und um fünf weitere Untersuchungsvariablen); sie weist so viele Zeilen auf, wie wir Untersuchungseinheiten (hier befragte Personen, bzw. Fragebögen) oder allgemein Merkmalsträger in der Untersuchung haben.

In diesem Zusammenhang sollen einige wichtige Begriffe erörtert werden :

Eine Zeile der Datenmatrix nennt man einen Fall (in der englischen EDV - Fachsprache spricht man von record , case oder set ). Ein Fall umfaßt die Werte aller Variablen für einen Merkmalsträger.

Eine Spalte der Datenmatrix nennt man Array . Ein Array umfaßt alle Werte einer Variablen für alle Merkmalsträger.

Die Gesamtheit aller Arrays, also alle Spalten der Datenmatrix zusammen, die identisch ist mit der Gesamtheit aller Records, also aller Zeilen, nennt man Datei = File .

Der Kreuzungsbereich einer Datenmatrixzeile mit einer Datenmatrixspalte, in dem sich ein Wert einer Variablen befindet, wird Feld oder Wert genannt = Field .

Generell können wir also im folgenden von einer  $N * V$  - Matrix ausgehen, wobei  $N$  die Zahl der Fälle und  $V$  die Zahl der Variablen bezeichnet.

Bei einfachen statistischen Auswertungen wird man es häufig nur mit einer einzigen Variablen zu tun haben ( $V = 1$ ), zum Beispiel wenn man sich für die Einkommen von 100 befragten Personen interessiert und für sonst nichts. Die Datenmatrix reduziert sich dann auf einen einzigen Array, in dem nur Einkommensangaben stehen (es sei denn, man numeriert zusätzlich die befragten Personen durch, so wie das in obigem Beispiel geschehen ist, was zu einer zweispaltigen Matrix führen würde).

Auch statistische Zeitreihen, mit denen man es häufiger zu tun hat, können als zweispaltige Datenmatrix interpretiert werden, wobei in der ersten Spalte Zeitpunkte (oder Zeiträume), in der zweiten die zugeordneten Merkmalswerte stehen.

### 3.3 Dateneingabe

Wenden wir uns nun der Frage zu, wie die Daten in den Rechner zur weiteren Auswertung gelangen können. Diese Frage soll im folgenden anhand einfacher Spieldaten beantwortet werden.

Von Anfang an müssen wir dabei die Betrachtungen unter drei verschiedenen Kriterien in unterschiedliche Zweige aufspalten :

1. Zunächst ist danach zu unterscheiden, ob Daten neu eingegeben werden, oder ob ein schon bestehender Datenbestand verlängert werden soll.
2. Desweiteren liegen unterschiedliche Bedingungen vor, je nachdem, ob der Benutzer ein Diskettenlaufwerk oder Magnetbandkassetten benutzt, oder ob er überhaupt keine externen Speicher verwendet.

3. Schließlich muß berücksichtigt werden, ob bei der Nutzung externer Speicher mit sequentiellen Dateien oder mit Random-access-Dateien gearbeitet wird. Wir betrachten hier nur den Fall der sequentiellen Speicherung.

In diesem Kapitel stellen wir zunächst den einfachsten Fall vor, die Datenbereitstellung innerhalb eines BASIC-Programms ohne Nutzung externer Speicher.

### 3.4 Dateneingabe mit LET-Anweisungen

Der einfachste Fall der Dateneingabe ohne Nutzung externer Speicher besteht in der Verwendung von LET-Statements. Dies wird mit dem folgenden Beispiel einer einfachen Mittelwertberechnung demonstriert. Auch bei diesem ersten Beispiel orientieren wir uns schon an der früher vorgestellten Folge von Arbeitsschritten, obwohl diese Aufgabenstellung noch so einfach ist, daß zu den einzelnen Schritten nicht allzu viel zu sagen ist :

- o Vorstellung des Problems

Aus den vier Werten 14, 17, 20 und 29 soll ein arithmetisches Mittel berechnet werden.

- o Problemanalyse

Ein arithmetisches Mittel berechnet sich aus einer Reihe von Merkmalswerten als die Summe dieser Werte geteilt durch deren Anzahl.

Damit gelangt man bei der Problemanalyse zu einer sehr schlichten Problemstruktur.

Ganz generell kann man problemanalytische Überlegungen sehr gut mit dem sogenannten

E V A - Prinzip

in den Griff bekommen. In diesem Stichwort steht E für Eingabe, V für Verarbeitung und A für Ausgabe.

Dies soll bedeuten, daß man sich zunächst Gedanken darüber machen soll, welche Eingabeinformationen das Programm benötigt, um die Aufgaben erledigen zu können, die erledigt werden sollen. Danach sind die Verarbeitungsschritte gedanklich zu konzipieren und schließlich soll man sich überlegen, welche Ergebnisse in welcher Art gewünscht sind.

Wenden wir dieses EVA-Prinzip auf das vorliegende Problem an, so ergibt sich :

E : Es sind die vier Ausgangswerte über LET-Anweisungen bereitzustellen

V : Es wird die Summe dieser Werte bestimmt, die danach durch 4 geteilt werden muß

A : Der sich ergebende Quotient muß als Ergebnis des Programms auf dem Bildschirm ausgegeben werden.

Damit ist die Problemanalyse im Prinzip schon abgeschlossen. Wir wollen aber zusätzlich auf einen speziellen Programmteil zu sprechen kommen, weil dieser in allen folgenden Programmen verwendet wird, und zwar auf ein Unterprogramm zur Unterbrechung der Bildschirmausgabe.

Immer dann, wenn große Ergebnismengen erzeugt werden, ist es sinnvoll, die Bildschirmausgabe zu un-

terbrechen, wenn der Bildschirm voll ist, um dem Benutzer Gelegenheit zu bieten, in Ruhe die Ergebnisse zu studieren. Wenn er sich alles angeschaut hat, soll die Programmabarbeitung weiter fortgesetzt werden.

Diese Möglichkeit der Programmunterbrechung können wir auch schon in dem sehr einfachen Programm verwenden, das wir hier problemanalytisch vorbereiten. Allen Programmen haben wir nämlich einen Titelausdruck und gegebenenfalls Erläuterungen zum Programm selbst vorangestellt, die so lange auf dem Bildschirm stehen bleiben, bis der Benutzer die Leertaste drückt.

Dieses Problem ist sehr einfach zu lösen :

Wir geben auf dem Bildschirm die Meldung aus

**BITTE DIE LEERTASTE DRUECKEN !**

und fordern dann per GET-Anweisung eine Tastatureingabe an, die in einem Stringfeld gespeichert werden kann. Solange in diesem Stringfeld nichts "ankommt" (solange also der Benutzer keine Taste gedrückt hat), was mit der IF ... THEN-Anweisung geprüft werden kann, verzweigen wir zurück zur GET-Anweisung. Erst nach erfolgter Tastaturbetätigung wird die normale Programmabarbeitung fortgesetzt.

o Programm

```
10 REM S31-DURCHSCHNITT
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(7)"■ BEISPIEL S31-DURCHSCHNITT:□"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT" DIESES PROGRAMM BERECHNET DAS ARITHME--"
90 PRINT" TISCHE MITTEL AUS VIER EINGEGEBENEN"
100 PRINTTAB(16)"WERTEN."
110 GOSUB 1000:REM WARTEN
120 LET A=14
130 LET B=17
140 LET C=20
150 LET D=29
160 LET S=A+B+C+D
170 LET AM=S/4
180 PRINTTAB(10)"AUSGANGSWERTE : "
190 PRINT:PRINTA,B,C,D:PRINT:PRINT:PRINT
200 PRINTTAB(10)"■MITTELWERT = ";AM;:PRINT"□"
210 PRINT:PRINT:PRINT:PRINTTAB(10)"ENDE DER AUSGABE"
220 END
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
```

o Variablenliste

A	=	Eingabewert
AM	=	Arithmetisches Mittel
A\$	=	Speicherfeld für die Tastatureingabe im Unterprogramm "Warten"
B	=	Eingabewert
C	=	Eingabewert
D	=	Eingabewert
S	=	Summe der Werte

o Programmbeschreibung

In diesem ersten Beispiel erfolgt die Programmbeschreibung sehr detailliert, damit wir uns dann bei den folgenden Programmen, insbesondere dort, wo schon bekannte Programmsegmente auftauchen, kürzer fassen können :

Satz 10 :

Programmüberschrift

Satz 20 :

Löschen des Bildschirms

Satz 30 :

Ausgabe einer Überschrift mit drei folgenden Leerzeilen

Satz 40 :

Ausgabe des Verfassers und des Jahrs der Programm-  
erstellung

Satz 50 :

Ausgabe von vier Leerzeilen

Satz 60 :

Ausgabe der Unter-Überschrift des Programms; dazu  
wird zunächst in weiße Schriftfarbe und nach der  
Ausgabe wieder in hellblaue Schriftfarbe umge-  
schaltet (die seltsamen Symbole hinter den ersten  
und vor den zweiten Anführungszeichen ergeben sich  
bei gleichzeitiger Betätigung der Tasten **CTRL**  
und **2** (weiß) bzw. **C** und **7** (hell-  
blau))

Satz 70 :

Ausgabe von vier Leerzeilen

Satz 80-100 :

Ausgabe einer Erläuterung zum Programm

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten und er-  
läuternder Hinweis im REM-Statement

Satz 120-150 :

Vorgabe der vier zu mittelnden Werte

Satz 160 :

Berechnung der Summe der vier Werte

Satz 170 :

Berechnung des arithmetischen Mittels

Satz 180-190 :

Ausgabe der vier Ausgangswerte

Satz 200 :

Ausgabe des berechneten Mittelwerts nach drei  
Leerzeilen (in weißer Schriftfarbe)

Satz 210 :

Ausgabe des Hinweises ENDE DER AUSGABE nach drei  
Leerzeilen

Satz 220 :

Ende des Hauptprogramms

Satz 1000 :

Beginn des Unterprogramms zum Abwarten mit einer  
Programmüberschrift

Satz 1010 :

Ausgabe von drei Leerzeilen

Satz 1020 :

Ausgabe des Textes BITTE DIE LEERTASTE DRUECKEN!  
(in schwarzer Schriftfarbe)

Satz 1030 :

Anforderung einer Tastatureingabe mit der GET-An-  
weisung; zu belegen ist das Feld A\$; solange in  
diesem Feld keine Information angekommen ist  
(A\$ = ""), wird zurück verzweigt zu 1030, d.h. es  
wird weiter die Tastatureingabe angefordert -  
das Programm "steht"

Satz 1040 :

Dieser Satz wird nur erreicht, wenn die Tastatureingabe erfolgt ist; ist dies der Fall, wird der Bildschirm gelöscht

Satz 1050 :

Rücksprung ins aufrufende Hauptprogramm

o Programmergebnisse

Wenn dieses Programm mit dem Kommando RUN gestartet wird, erhalten wir auf dem gelöschten Bildschirm als erstes die folgende Ausgabe :

STATISTIK

PROF.DR.W.VOSS, 1985

BEISPIEL S31-SCHNITT

DIESES PROGRAMM BERECHNET DAS ARITHMETISCHE MITTEL AUS VIER EINGEGEBENEN WERTEN.

BITTE DIE LEERTASTE DRUECKEN!

Wenn der Benutzer daraufhin die Leertaste drückt, ergibt sich - wiederum auf einem gelöschten Bildschirm :

AUSGANGSWERTE :

14            17            20            29

MITTELWERT = 20

ENDE DER AUSGABE

Auf die Angaben zu Änderungen der Schriftfarbe ist hier verzichtet worden. Auch in den folgenden Programmen wollen wir darauf nicht mehr eigens auf -

merksam machen, weder bei der Darstellung der Ergebnisse, noch bei der jeweiligen satzweisen Programmbeschreibung.

o Ergänzungen

Wir haben uns mit diesem Programm recht lange aufgehalten, weil es das erste unserer Demonstrationsbeispiele ist, und weil man an diesem einfachen Programm noch sehr übersichtlich die einzelnen Arbeitsschritte beschreiben kann, die erforderlich sind.

Bei den folgenden Programmen können wir uns entsprechend kürzer fassen, Bekanntes sehr knapp abhandeln und nur die jeweils neuen Programmteile einer detaillierteren Betrachtung unterziehen.

Es versteht sich von selbst, daß ein so simples Problem, wie aus vier Werten den Durchschnitt auszurechnen, natürlich sehr viel kürzer programmiert werden kann, als dies oben geschehen ist. Zum Beispiel würde das folgende einfache Programm die gleiche Aufgabe erledigen können :

```
10 PRINT (14+17+20+29)/4
20 PRINT "ENDE":END
```

Gleichwohl haben wir schon bei diesem ersten Programm eine ausführlichere Art der Programmierung gewählt, weil später bei den noch folgenden umfangreicheren Programmen diese Ausführlichkeit sehr der Übersichtlichkeit und dem Nachvollzug dienen wird.

Unter ergänzenden Gesichtspunkten ist darauf hinzuweisen, daß dieses Programm auch verwendet werden kann, um aus mehr als vier Werten den Durchschnitt auszurechnen. Sollte dies gewünscht werden, benötigt man vor Satz 160 weitere LET-Anweisungen und die Sätze 160, 170 und 190 müssen entsprechend verändert werden.

### 3.5 Dateneingabe mit INPUT-Anweisungen

Sehr viel geschickter als mit LET-Anweisungen wie in dem vorangegangenen Beispiel läßt sich die Dateneingabe mit der INPUT-Anweisung bewerkstelligen. Sie erlaubt es darüberhinaus, das Mittelwertberechnungsprogramm so zu verallgemeinern, daß es auf eine beliebige Anzahl von Werten ausgedehnt werden kann.

Dies zeigt exemplarisch das folgende Programm :

#### o Vorstellung des Problems

Es soll ein Programm erstellt werden, welches aus einer beliebigen, vom Benutzer vorzugebenden Anzahl von Werten das arithmetische Mittel berechnet.

#### o Problemanalyse

Bei der Problemanalyse können wir uns nun kürzer fassen als im vorangegangenen Beispiel, da mit Ausnahme des Dateneingabeteils dieses Problem keine prinzipiell neuen Fragen aufwirft.

Damit der Benutzer eine beliebige Anzahl von Werten der Durchschnittsberechnung unterwerfen kann, muß zunächst die gewünschte Anzahl von Werten im Programm mitgeteilt werden.

Ist diese Anzahl dann bekannt, so können in einer Eingabeschleife unter Benutzung des INPUT-State-

ments so viele Werte angefordert werden, wie auch gemittelt werden sollen, und es kann in der gleichen Schleife auch die Summe dieser Werte gebildet werden.

Wann diese Schleife abgebrochen werden muß, kann im Programm festgestellt werden, indem man einen Zählindex bei jeder Schleifenrunde um 1 erhöht und jeweils prüft, ob er nach dieser Erhöhung schon größer geworden ist als der vorgegebene Wert, der die gewünschte Anzahl der zu mittelnden Daten angibt.

Solange er noch nicht größer ist, wird eine erneute INPUT- und Summierungsrunde in Angriff genommen; wenn aber doch, so ist diese Schleife zu verlassen. Die erhaltene Summe kann dann durch die Zahl der Werte dividiert und dieses Ergebnis muß ausgegeben werden.

Es leuchtet unmittelbar ein, daß die Anzahl der zu mittelnden Werte sinnvollerweise eine ganze Zahl sein muß, und daß sie nicht kleiner als 1 sein; um entsprechende Fehleingaben des Benutzers abzufangen, fügen wir eine Abfrage ein, die zur Eingabe der Anzahl der Werte zurück verzweigt, wenn der Benutzer einen unzulässigen Wert eingegeben haben sollte.

Entsprechende Programmteile zum "Abfangen" von fehlerhaften oder nicht zulässigen Eingaben werden auch in den folgenden Programmen verwendet, ohne daß in der Problemanalyse darauf jeweils aufmerksam gemacht werden müßte.

o. Programm

```
10 REM S32-MITTELWERT
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(8)"■BEISPIEL S32-MITTELWERT■"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT" DIESES PROGRAMM BERECHNET DAS ARITHME-"
90 PRINT" TISCHE MITTEL AUS BELIEBIG VIELEN EIN-"
100 PRINTTAB(11)"ZUGEBENDEN WERTEN."
110 GOSUB 1000:REM WARTEN
115 PRINT:PRINT:PRINT
120 PRINT"WIEVIELE WERTE SOLLEN GEMITTELT WERDEN?"
130 PRINT:INPUT"BITTE ZAHL ANGEBEN UND RETURN ";N
134 PRINT:PRINT
135 IF N<1 OR N<>INT(N) THEN PRINT"■EINGABE NICHT
ERLAUBT■":GOTO 115
140 PRINT:PRINT:PRINT
150 FOR I=1 TO N
160 PRINTI;:INPUT ".WERT (UND RETURN): ";X
170 S=S+X
180 NEXT I
190 AM=S/N:PRINT:PRINT:PRINT
200 PRINTTAB(9)"■MITTELWERT = ";AM;:PRINT"■"
210 PRINT:PRINT:PRINT:PRINTTAB(10)"ENDE DER AUSGABE"
220 END
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
```

o Variablenliste

AM = Mittelwert  
A\$ = Tastatureingabe  
I = Laufindex  
N = Anzahl der Werte  
S = Summe (Zwischensumme)  
X = Eingegebener Wert

o Programmbeschreibung

Auch hier können wir uns jetzt wesentlich kürzer fassen, weil sich ja viele Programmteile nicht von dem ersten Programm (S31) unterscheiden. Unter anderem können wir sicherlich auch darauf verzichten, gesondert anzugeben, wenn durch das Programm der besseren Lesbarkeit halber Leerzeilen ausgegeben werden (z.B. Satz 140).

Satz 10-100 :

Überschrift, Titel und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste gedrückt wird

Satz 115-130 :

Anforderung der gewünschten Anzahl der zu mitteln-  
den Werte und Speicherung dieser Angabe im Feld N

Satz 134-135 :

"Abfangen" unzulässiger Eingaben und Rücksprung zur Eingabe (Satz 115), falls dies erforderlich ist

Satz 150 :

Beginn der Eingabe- und Summierungsschleife durch Belegung des Laufindex I mit dem Wert 1

Satz 160 :

Anforderung des I-ten zu mittelnden Wertes und Speicherung im Feld X

Satz 170 :

Erhöhung der (Zwischen-)Summe im Feld S (in der ersten Runde, also für  $I = 1$ , mit dem Wert 0 belegt) um den Wert aus dem Feld X

Satz 180 :

Erhöhung des Laufindex I um 1; solange nach dieser Erhöhung im Feld I ein Wert steht, der kleiner oder gleich dem Endwert im FOR...TO-Statement ist (nämlich N), wird die Schleife (150-180) nicht verlassen; erst wenn I größer wird als N, geht es weiter bei Satz 190

Satz 190 :

Berechnung des arithmetischen Mittels und Speicherung im Feld AM

Satz 200 :

Ausgabe des errechneten Ergebnisses

Satz 210-220 :

Beendigung des Hauptprogramms

Satz 1000-1050

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Programmergebnisse

Wenn dieses Programm gestartet wird, erscheint zunächst wieder der Titelausdruck auf dem Bildschirm, danach Erläuterungen zum Programm und die Aufforderung, die Leertaste zu betätigen.

Dann meldet das Programm :

WIEVIELE WERTE SOLLEN GEMITTELT WERDEN?

BITTE ZAHL ANGEBEN UND RETURN ?

Geben wir daraufhin z.B. den Wert 4 an, so reagiert das Programm mit der folgenden Anforderung :

1.WERT : ?

Geben wir jetzt einen Wert ein, fordert das Programm den nächsten Wert an usw., und zwar insgesamt viermal.

Sind die vier Werte eingegeben, so wird auf dem Bildschirm das arithmetische Mittel dieser vier Werte ausgegeben.

Danach wird mit der Meldung ENDE DER AUSGABE das Programm beendet.

Hätten wir auf die Frage des Programms, wieviele Werte gemittelt werden sollen, z.B. den Wert -1

oder den Wert 14.3 eingegeben, so wäre ausgegeben worden :

EINGABE NICHT ERLAUBT

und es wäre erneut die Zahl der zu mittelnden Werte angefordert worden.

o Ergänzungen

Dieses Programm, das sich zur generellen Mittelwertberechnung eignet, kann ebenfalls stark verkürzt werden, wenn man dies wünscht.

Es muß darauf geachtet werden, daß bei der Anforderung von Zahlenwerten (Satz 130 und Satz 160) tatsächlich nur Zahlen und nicht etwa Buchstaben oder andere Symbole eingegeben werden, weil es sonst zu Fehlermeldungen kommt und das Programm abgebrochen wird.

Nach jeder Eingabe ist die RETURN-Taste zu betätigen.

Weitere Anmerkungen sind zu diesem Programm nicht erforderlich.

### 3.6 Dateneingabe über DATA und READ

Neben den bisher vorgestellten Möglichkeiten der Dateneingabe über LET- bzw. über INPUT-Anweisungen kann man natürlich auch Daten über DATA- und READ-Statements bereitstellen, was sich insbesondere bei größeren Datenbeständen empfiehlt.

Dies demonstriert das folgende Beispiel, bei dem für 8 befragte Personen jeweils 3 Werte (Körpergröße, Körpergewicht und Blutdruck) eingegeben werden sollen.

Damit kommen wir auf den Begriff der Datenmatrix zurück, die in diesem Beispiel aus 8 Zeilen (Fällen) und 3 Spalten (Variablen) besteht. Diese Datenmatrix wird über DATA - Statements vorgegeben und mit READ-Statements eingelesen.

o Vorstellung des Problems

Es soll eine  $8 * 3$  - Datenmatrix eingelesen und zu Kontrollzwecken auf dem Bildschirm wieder ausgegeben werden.

o Problemanalyse

Wir erinnern uns wieder an das EVA - Prinzip und müssen deshalb zunächst die  $8*3 = 24$  Einzelwerte bereitstellen. Für diese Werte muß auch Platz geschaffen werden, was durch eine geeignete Dimensionierungsanweisung geschieht.

Sind die Daten in DATA-Statements bereitgestellt, so können sie per READ-Anweisungen gelesen werden. Man benutzt dazu zweckmäßigerweise zwei geschachtelte Schleifen, arbeitet also mit zwei Laufindizes, von denen der erste den jeweiligen Fall, der zweite die jeweilige Variable indiziert.

Nach dem gleichen Prinzip kann dann die Ausgabe der Werte organisiert werden.

o Programm

```
10 REM S33-DATENMATRIX
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(8)"■BEISPIEL S33-DATENMATRIX■"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM DIEN DATZU, EINE VORGE-"
90 PRINT"GEBENE DATENMATRIX (NUMERISCHE WERTE)"
100 PRINT"EINZULESEN UND ZUR KONTROLLE AUSZUGEBEN"
110 GOSUB 1000:REM WARTEN
120 DIM X(8,3)
130 FOR I=1 TO 8
140 FOR J=1 TO 3
150 READ X(I,J)
160 NEXT J
170 NEXT I
180 PRINT"■FALL", "VAR.1", "VAR.2", "VAR.3":PRINT
190 FOR I=1 TO 8
200 PRINTI,
210 FOR J=1 TO 3:PRINTX(I,J),:NEXT J
220 NEXT I
230 PRINT:PRINTTAB(10)"ENDE DER AUSGABE":END
500 REM DATEN
510 DATA 175,65,125
520 DATA 180,88,132
530 DATA 185,92,128
540 DATA 165,66,140
550 DATA 172,78,138
560 DATA 178,75,118
570 DATA 186,90,130
580 DATA 173,69,143
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
```

o Variablenliste

A\$ = Tastatureingabe

I = Laufindex (Fälle)

J = Laufindex (Variablen)

X(,) = Werte (doppelt indiziert)

o Programmbeschreibung

Satz 10-100 :

Überschrift und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste betätigt wird

Satz 120 :

Dimensionierung der Datenmatrix

Satz 130-170 :

Geschachtelte Einleseschleife;gelesen wird in folgender Reihenfolge :

X(1,1), X(1,2), X(1,3), X(2,1), X(2,2) usw.

bis ... X(8,2), X(8,3)

Satz 180 :

Ausgabe einer Tabellenüberschrift

Satz 190 :

Beginn der äußeren Ausgabeschleife (Fälle)

Satz 200 :

Ausgabe der Fallnummer; weitere Ausgaben erfolgen in der gleichen Bildschirmzeile, weil diese PRINT-Anweisung mit einem Komma endet

Satz 210 :

Innere Ausgabeschleife (Variablen); die drei Variablenwerte (J = 1 bis 3) werden nebeneinander in einer Bildschirmzeile ausgegeben, weil die entsprechende PRINT-Anweisung mit einem Komma endet

Satz 220 :

Ende der äußeren Ausgabeschleife

Satz 230 :

Beendigung des Hauptprogramms

Satz 500-580 :

Daten in DATA-Statements; diese Statements sind satzweise aufgebaut, was nicht obligatorisch ist, aber der Übersichtlichkeit des Programmlistings dient

Satz 1000-1050

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Programmergebnisse

Nach Überschrift und Erläuterungen erhalten wir auf dem Bildschirm die folgenden Ergebnisse :

FALL	VAR.1	VAR.2	VAR.3
1	175	65	125
2	180	88	132
...	...	..	...
8	173	69	143

ENDE DER AUSGABE

o Ergänzungen

Zu diesem Programm sind einige Anmerkungen erforderlich :

Es ist ohne Probleme möglich, es auf mehr Fälle und/oder mehr Variablen auszudehnen; es müssen dann die Dimensionierung (Satz 120) und die Schleifenparameter (Satz 130, 140, 190, 210) geändert werden und natürlich müssen die DATA - Statements ergänzt werden.

Will man das Programm allgemeiner halten, so empfiehlt es sich, sowohl die Zahl der Fälle als auch die Zahl der Variablen durch den Benutzer frei wählen zu lassen und dann die Werte in einer geschachtelten INPUT-Schleife einzugeben, wie es der folgende Programmausschnitt schematisch zeigt:

```

120 INPUT "WIEVIELE FAELLE      ";N
130 INPUT "WIEVIELE VARIABLEN ";V
140 DIM X(N,V)
150 FOR I = 1 TO N
160 PRINT I;".FALL :":PRINT
170 FOR J = 1 TO V
180 PRINT "      VARIABLE NR. ";J;:INPUT X(I,J)
190 NEXT J
200 NEXT I

```

Diese Art der Dateneingabe funktioniert in dieser Weise aber nur dann, wenn alle Variablen vom numerischen Typ sind (oder wenn sie alle Stringvariablen sind; statt mit X(I,J) wäre dann einfach mit X\$(I,J) zu arbeiten).

Wenn die Variablentypen hingegen gemischt auftauchen, wird die Programmierung etwas umständlicher, wie am Beispiel der Dimensionierung verdeutlicht werden soll :

Nehmen wir an, jeder der N Datensätze wäre aus 7 Variablen aufgebaut, nämlich :

- Körpergröße
- Körpergewicht
- Wohnort
- Straße
- Hausnummer
- Blutdruck
- Zigarettenkonsum

Die dritte und die vierte Variable sind Stringvariablen, die anderen sind numerische Variablen. Deshalb kann jetzt nicht folgendermaßen dimensioniert werden :

```
N=8:V=7:DIM X(N,V)
```

sondern zum Beispiel folgendermaßen :

```
N=8:VN=5:VS=2
DIM X(N,VN),X$(N,VS)
```

Dementsprechend müssen auch Einlese- und Ausgabe-  
prozesse etwas umständlicher programmiert werden.

Ein letzter ergänzender Hinweis bezieht sich auf  
die Programmierung der Bildschirmausgabe der Da-  
tenmatrix :

Sie besteht in unserem Beispiel aus genau vier  
Spalten (drei Variablen + der Durchnummerierung in  
der ersten Spalte) und füllt somit den 40-Zeichen  
-Bildschirm des Commodore C64 genau aus, weil das  
Komma in den entsprechenden PRINT-Anweisungen den  
Bildschirm in vier Bildschirmzonen aufteilt.

Hat man mehr Variablen, so müssen die Spalten en-  
ger zusammengerückt werden, was durch geeignete  
Verwendung des Strichpunkts anstelle des Kommas  
möglich ist, oder (bzw. falls auch dies nicht aus-  
reicht) es sind erst die Spalten 1 bis 4, dann die  
Spalten 5 bis 8 usw. auszugeben, d.h. es ist dann  
mit Programmunterbrechungen zu arbeiten.

Entsprechendes gilt, wenn nicht nur 8, sondern z.  
B. 100 Fälle auszugeben sind; die ersten dieser  
Fälle würden zu rasch über den oberen Bildschirm-  
rand hinaus verschwinden, weil fortwährend von  
unten neue nachgeschoben werden. Dies kann man  
verhindern, wenn man beispielsweise nach je 15  
Fällen die Bildschirmausgabe unterbricht, etwa mit  
der folgenden Anweisung :

...

```
215 IF I/15 = INT(I/15) THEN GOSUB 1000
```

Mit dieser Anweisung wird erreicht, daß dann, wenn  
I = 15, 30, 45, ..., die Bildschirmausgabe stehen  
bleibt, bis der Benutzer die Leertaste drückt. Die-  
ser Möglichkeit der Ausgabegestaltung werden wir  
später wieder begegnen.

### 3.7 Zählen

Wenden wir uns schließlich einem letzten Beispiel einfacher Dateneingabe zu. Es geht dabei zusätzlich nun darum, daß es dem Programm überlassen bleibt, festzustellen, wieviele Daten eingegeben werden, was ja in den bisherigen Beispielen immer vom Benutzer vorgegeben werden mußte.

#### o Vorstellung des Problems

Das folgende Programm soll zunächst die Anzahl der eingegebenen Daten selbst bestimmen und aus diesen Daten dann den Mittelwert berechnen.

#### o Problemanalyse

Bezüglich der Mittelwertberechnung ergeben sich hier keinerlei neuen Probleme. Neu ist lediglich die Frage, wie es erreicht werden kann, daß das Programm von sich aus feststellt, wieviele Werte zu mitteln sind.

Zur Beantwortung dieser Frage gibt es verschiedene Möglichkeiten. Der einfachste Weg besteht darin, einen unrealistischen Wert als "Abbruchwert" einzugeben.

Wenn wir zum Beispiel Körpergrößen mitteln wollen, dann könnten wir in dem Moment, in dem keine echten Werte mehr vorhanden sind, weil alle schon eingegeben sind, beispielsweise den Wert -99.99 eingeben - sicherlich keine "echte" Körpergröße.

Nach jeder Dateneingabe lassen wir das Programm überprüfen, ob der gerade eingegebene Wert (schon) der Wert -99.99 ist. Wenn ja, ist die Dateneingabe abzubrechen und die Zahl der echten Werte ergibt sich aus dem Wert, den der mitgelaufene Zählindex gerade aufweist minus 1. Ist hingegen der gerade eingegebene Wert nicht die -99.99, so ist die Eingabeprozedur weiter fortzusetzen.

Wenn wir allerdings gemäß dieser Überlegungen die Daten über DATA- und READ-Anweisungen und nicht über INPUT - Anweisungen eingeben, so ist ein zusätzliches Problem zu lösen :

Wenn nämlich alle Werte zum Zwecke des Zählens gelesen worden sind (bis zum Wert -99.99 einschließlich, der am Ende des Datenbesands stehen muß), so sind weitere READ-Statements erforderlich, um dann die Werte zum Zwecke der Mittelwertberechnung zu lesen.

Während des Zählprozesses sind die Werte ja noch nicht gespeichert worden, weil wir nicht wußten, wieviele es sind und deshalb nicht dimensionieren konnten, was Voraussetzung für die Speicherung aller Werte gewesen wäre.

Dies bedeutet, daß erneut (von Anfang an) gelesen werden muß, wobei jetzt auch, weil ja zwischenzeitlich dimensioniert wurde, gespeichert werden kann. Zu diesem Zweck muß aber der "Lesezeiger" an den Anfang des ersten DATA-Statements "zurückgeschickt" werden (folgende READ-Statements fahren ja dort fort zu lesen, wo vorangegangene aufgehört haben; in unserem Fall hinter der -99.99, wo aber nichts mehr ist, d.h. es käme zu einer Fehlermeldung).

Dieses Zurückschicken gelingt mit dem Statement

```
nn  RESTORE
```

o Programm

```
10 REM S34-ZAEHLEN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(10)"■BEISPIEL S34-ZAEHLEN■"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BESTIMMT DIE ANZAHL GE-"
90 PRINT"GEBENER DATEN UND DANN DEN MITTELWERT"
100 PRINTTAB(11)"AUS DIESEN DATEN."
110 GOSUB 1000:REM WARTEN
120 FOR I=1 TO 10000
130 READ X:IF X=-99.99 THEN 150
140 NEXT I
150 N=I-1
160 PRINT"■ANZAHL DER WERTE : ";N:PRINT"■":PRINT
170 DIM X(N)
180 RESTORE
190 FOR I=1 TO N:READ X(I):NEXT I
200 PRINT"AUSGABE DER WERTE : ":PRINT
210 FOR I=1 TO N:PRINTX(I);:NEXT I:PRINT
220 REM MITTELWERT
230 FOR I=1 TO N:S=S+X(I):NEXT I
240 AM=S/N
250 PRINT:PRINT:PRINT"■MITTELWERT = ";AM;:PRINT"■"
260 PRINT:PRINT:PRINTTAB(10)"ENDE DER AUSGABE":END
500 REM DATEN
510 DATA 180,181,178,175,167,183,190
520 DATA 175,163,188,192,168,171,177
530 DATA 172,177,175
540 DATA -99.99
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
```

o Variablenliste

AM = Mittelwert

A\$ = Tastatureingabe

I = Laufindex

N = Anzahl der Werte

S = Summe der Werte

X = Einzugebender Wert (später indiziert)

o Programmbeschreibung

Satz 10-100 :

Überschrift und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste gedrückt wird

Satz 120 :

Beginn der Zählschleife; ihr Endwert wird beliebig hoch angesetzt, damit alle "echten" Werte vor dem Wert -99.99 tatsächlich erfaßt werden können

Satz 130 :

Lesen des I-ten Wertes in das Feld X; wenn dieser Wert die Zahl -99.99 ist, erfolgt ein Sprung zum Satz 150, andernfalls wird Satz 140 erreicht

Satz 140 :

Erhöhung des Laufindex I um 1 und Fortsetzung des Zählens

Satz 150 :

Satz 150 wird nur erreicht, wenn kein echter Wert mehr vorliegt; die Zahl der echten Werte ist dann I-1 (gespeichert im Feld N)

Satz 160 :

Bekanntgabe der Anzahl echter Werte

Satz 170 :

Dimensionierung

Satz 180 :

Restaurieren des Datenbestands, damit er erneut gelesen werden kann

Satz 190 :

Einlesen aller echten Werte in den X-Array

Satz 200-210 :

Kontrollausgabe aller Werte

Satz 220-250 :

Berechnung des arithmetischen Mittels und Ausgabe auf dem Bildschirm

Satz 260 :

Beendigung des Hauptprogramms

Satz 500-540 :

Spieldatenbestand; der letzte Wert muß die Zahl  
-99.99 sein

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

o Programmergebnisse

Nach Überschrift und Erläuterungen meldet das Pro-  
gramm zunächst :

ANZAHL DER WERTE : 17

Danach erfolgt die Kontrollausgabe :

AUSGABE DER WERTE :

180 181 178 ... usw. ... 177 175

Danach erfolgt die Mittelwertberechnung :

MITTELWERT = 177.17...

ENDE DER AUSGABE

## o Ergänzungen

Hier sind prinzipiell die gleichen Anmerkungen angebracht wie bei den Programmbeispielen zuvor, so daß wir auf weitere Hinweise verzichten können.

### 3.8 Ausblick

Wir haben nun an vier Beispielen gezeigt, wie man Daten zu Zwecken der Datenverarbeitung in den Rechner hineinbekommt, ohne daß man auf externe Speicher, also z.B. auf Disketten zurückgreift.

Diese Art der Dateneingabe könnte allerdings dann zu Schwierigkeiten führen, wenn man einen bestimmten Datenbestand mehrfach nutzen möchte.

Dies ist bei den vorangegangenen Beispielen nur in der Weise lösbar, daß man nun doch externe Speicher, also z.B. Disketten nutzt und das gesamte Programm per SAVE-Kommando auf Kasette oder Diskette speichert.

Wenn man für spätere Verwendungszwecke dieses gespeicherte Programm dann per LOAD-Kommando wieder in den Arbeitsspeicher des Rechners einlädt, dann sind auch die Daten wieder vorhanden, falls diese per DATA- und READ-Statements und nicht per INPUT-Statements bereitgestellt wurden.

Man kann nun für erneute, andere Auswertungen unter Verwendung dieser DATA-Statements neue Auswertungsanweisungen programmieren und dabei die alten Anweisungen, die nun nicht mehr benötigt werden, löschen.

Dies ist zugegebenermaßen eine etwas umständliche

Vorgehensweise, aber sie funktioniert - zumindest bei kleineren Auswertungsaufgaben - sehr gut.

Wie man eleganter, nämlich unter interaktiver Diskettennutzung vorgehen kann, zeigt das folgende Kapitel.

## Kapitel 4 : Dateneingabe mit externem Speicher

### 4.1 Vorbemerkung

Am Ende des vorangegangenen Kapitels haben wir darauf hingewiesen, daß, wenn man einen bestimmten Datenbestand mehrfach nutzen möchte, der Einsatz externer Speicher unumgänglich ist, wenn vermieden werden soll, den betreffenden Datenbestand immer wieder neu (überflüssigerweise) in den Rechner eintippen zu müssen.

Es gibt eine professionellere Art der Datenspeicherung, die zudem einige sehr angenehme zusätzliche Möglichkeiten der Datenverwaltung eröffnet, die wir uns dann im folgenden Kapitel - soweit statistische Aufgabenstellungen davon betroffen sind - anschauen werden.

Es wurde schon erwähnt, daß prinzipiell zwei Arten der Datenspeicherung auf einem Diskettenspeicher voneinander zu unterscheiden sind :

- sequentielle Speicherung
- Random-access-Speicherung

Wir werden uns hier ausschließlich mit der ersten Möglichkeit befassen, weil sie einfacher zu behandeln ist als die zweite, weil sie zudem für die Zwecke einfacher statistischer Auswertungen immer ausreichend ist und weil sie nicht nur auf die Nutzung von Disketten als externen Speichern beschränkt ist.

## 4.2 Speicherung eines neuen Datenbestands

Betrachten wir als erstes den Fall, daß ein neu eingegebener Datenbestand auf einer Diskette gespeichert werden soll.

Zur Verdeutlichung betonen wir nochmals, daß es jetzt nicht mehr darum geht, ein ganzes Programm (auch wenn es vielleicht nur aus DATA-Statements bestehen sollte) per SAVE-Kommando auf einer Diskette zu speichern, sondern ein laufendes BASIC-Programm soll von sich aus dafür sorgen, daß Daten auf eine Diskette übertragen werden.

### o Vorstellung des Problems

Die Daten des Programms S33 (8 \* 3 - Datenmatrix) sollen eingelesen und zur Kontrolle wieder ausgegeben werden (dies war auch die Aufgabenstellung des Programms S33); zusätzlich sollen diese Datensatzweise (fallweise) in einer sequentiellen Diskettendatei unter dem Namen DATEN1 gespeichert werden.

### o Problemanalyse

Gegenüber dem Programm S33 ergibt sich hier nur das zusätzliche Problem, die Datenmatrix auf eine Diskette zu übertragen, was voraussetzt, daß die Daten per READ - Anweisungen vom Programm gelesen worden sind.

Weiterhin muß vorausgesetzt werden, daß der Benutzer eine formatierte Diskette in das Laufwerk einlegt, die wir hinfort "Datendiskette" nennen wollen. Damit er dies nicht vergißt, wird an geeigneter Stelle das Programm unterbrochen, eine entsprechende Meldung ausgegeben und so lange gewartet, bis der Benutzer die Datendiskette eingelegt hat.

Wir empfehlen in diesem Zusammenhang ganz generell, daß Datenbestände auf gesonderten Disketten gespeichert werden sollten und nicht etwa auf der gleichen Diskette, auf der der Benutzer seine Statistik-Programme gespeichert hat.

Wenn die Datendiskette bereitgestellt ist, muß per Programm mit Hilfe der OPEN - Anweisung der Zugang zu ihr geöffnet werden; danach können mit der PRINT#-Anweisung die Daten gespeichert werden und per CLOSE-Anweisung kann dann das Programm beendet werden.

Ob die gewünschte Speicherung erfolgreich war, kann durch Wieder-Einlesen der Daten geprüft werden, worüber im Kapitel über Datenausgabe gesprochen wird.

Eine erste grobe Prüfung ist auch möglich durch Ausgabe des Inhaltsverzeichnisses der Datendiskette mit den Kommandos :

```
LOAD "$",8    und  
LIST
```

ACHTUNG :

Das Laden des Inhaltsverzeichnisses löscht das vorher im Arbeitsspeicher befindliche Programm. Also vorher überlegen, ob dies erlaubt sein soll.

Bei der Benutzung von Disketten können in vielfacher Weise Fehler begangen werden. Beispiels-

weise wird eine noch nicht formatierte Diskette benutzt oder man vergißt überhaupt, eine Diskette in das Laufwerk einzulegen. Oder man versucht eine Datei zu speichern, obwohl unter dem gleichen Namen sich auf der Diskette schon eine Datei befindet : Derartige Fehler werden nur daran erkannt, daß die rote Warnlampe des Diskettenlaufwerks fortwährend blinkt. Der jeweilige Datenverarbeitungsprozeß kann dann nicht fortgesetzt werden.

Eine Sicherung gegen derartige Fehler besteht darin, den sog. Fehlerkanal des Laufwerks abzufragen, um festzustellen, um welche Art des Fehlers es sich handelt.

Zu diesem Zweck werden wir in diesem und auch in allen folgenden Programmen, die mit Diskettenzugriff arbeiten, ein Unterprogramm bereitstellen, welches für eine Fehlermeldung, die auf dem Bildschirm ausgegeben wird, sorgt. Dieses Unterprogramm sieht folgendermaßen aus :

```
2000 REM UP DISK
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN = 0 THEN RETURN
2030 PRINT CHR$(147)
2040 PRINT TAB(12)"DISKETTENFEHLER":PRINT:PRINT
2050 PRINT EN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT "NEUSTART NACH FEHLERBESEITIGUNG MIT
      DEM"
2080 PRINT:PRINT TAB(13)"KOMMANDO RUN"
2090 PRINT:PRINT:PRINT
2100 CLOSE 8:CLOSE 15
2110 PRINT TAB(10)"ENDE DES PROGRAMMS":END
```

Die Nutzung dieses Unterprogramms setzt voraus, daß im Hauptprogramm vor dem ersten Diskettenzugriff der Fehlerkanal 15 geöffnet wird, was die folgende Anweisung erfordert :

nn OPEN 15,8,15

Zudem muß nach jedem Diskettenzugriff die Anweisung

nn GOSUB 2000

folgen und es muß - wie oben geschehen - nach der eventuellen Fehlerausgabe sowohl der Datenkanal 8 wie auch der Fehlerkanal 15 geschlossen werden.

Da bei Auftreten eines derartigen Fehlers das Weiterarbeiten im jeweiligen Datenverarbeitungsprozeß nicht sinnvoll ist, wird im Unterprogramm selbst der gesamte Programmlauf mit einer entsprechenden Meldung beendet.

Im Unterprogramm selbst werden in Satz 2010 Fehlernummer (EN), Fehlerart (EN\$) und weitere Informationen (ET und ES), die das Betriebssystem bereitstellt, angefordert. Ist EN = 0, liegt kein Fehler vor und es kann in das Hauptprogramm zurückgekehrt werden (Satz 2020); ist hingegen ein Fehler aufgetreten, so werden die Fehlerinformationen ausgegeben (Satz 2030-2060) und das Programm wird, wie oben beschrieben, beendet (Satz 2070-2110).

o Programm

```
10 REM S41-DSCHREIBEN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(9)"#BEISPIEL S41-DSCHREIBEN#"
70 PRINT:PRINT:PRINT:PRINT
80 PRINTTAB(2)"MIT DIESEM PROGRAMM WIRD DIE DATEN-"
90 PRINTTAB(1)"MATRIX AUS DEM PROGRAMM S33 AUF EINER"
100 PRINTTAB(3)"DISKETTE GESPEICHERT (SCHREIBEN)."
```

110 GOSUB 1000:REM WARTEN

```
120 DIM X(8,3)
130 FOR I=1 TO 8
140 FOR J=1 TO 3
150 READ X(I,J)
160 NEXT J
170 NEXT I
180 PRINT"=FALL", "VAR.1", "VAR.2", "VAR.3":PRINT
190 FOR I=1 TO 8
200 PRINTI,
210 FOR J=1 TO 3:PRINTX(I,J),:NEXT J
220 NEXT I
230 PRINT:PRINT:PRINT"=ZUM SPEICHERN BITTE
    DATENDISKETTE EIN="
240 PRINT"LEGEN UND DANN DIE LEERTASTE DRUECKEN !:"
250 PRINT:PRINT:PRINT
260 GET A$:IF A$="" THEN 260
265 OPEN 15,8,15
270 OPEN 8,8,8,"DATEN1,S,W"
275 GOSUB 2000:REM DISK
280 FOR I=1 TO 8:FOR J=1 TO 3
290 PRINT#8,X(I,J):GOSUB 2000:NEXT J:NEXT I
300 CLOSE 8,8,8
310 PRINT:PRINTTAB(12)"SPEICHERUNG BEENDET"
320 PRINT:PRINT:PRINTTAB(12)"ENDE DES PROGRAMMS":END
```

```
500 REM DATEN
510 DATA 175,65,125
520 DATA 180,88,132
530 DATA 185,92,128
540 DATA 165,66,140
550 DATA 172,78,138
560 DATA 178,75,118
570 DATA 186,90,130
580 DATA 173,69,143
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
2000 REM UP DISK
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT CHR$(147)
2040 PRINTTAB(12)"■DISKETTENFEHLER!□":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG MIT DEM"
2080 PRINT:PRINTTAB(13)"KOMMANDO RUN"
2090 PRINT:PRINT:PRINT
2100 CLOSE 8:CLOSE 15
2110 PRINTTAB(10)"ENDE DES PROGRAMMS":END
```

o Variablenliste

A\$ = Tastatureingabe

EM\$, EN, ES, ET = Informationen aus dem Disketten-Fehlerkanal

I = Laufindex (Fälle)

J = Laufindex (Variablen)

X(,) = Daten

o Programmbeschreibung

Satz 10-100 :

Überschrift und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste betätigt wird

Satz 120 :

Dimensionieren

Satz 130-170 :

Einlesen in einer geschachtelten Schleife (I=Laufindex für die Fälle; J=Laufindex für die Variablen; vergl. auch Programm S33)

Satz 180 :

Ausgabe einer Tabellenüberschrift

Satz 190-220 :

Ausgabe der Datenmatrix

Satz 230-250 :

Ausgabe der Aufforderung, die Datendiskette einzulegen

Satz 260 :

Abwarten, bis die Leertaste gedrückt wird

Satz 265 :

Eröffnen des Disketten-Fehlerkanals

Satz 270 :

Eröffnen einer Datei mit dem Namen DATEN1 zum Schreiben (Speichern; deshalb Buchstabe W = Write); es handelt sich um eine sequentielle Datei (deshalb Buchstabe S)

Satz 275 :

Sprung ins Unterprogramm 2000, um eventuelle Diskettenfehler festzustellen

Satz 280-290 :

Einschreiben (Speichern) aller Werte mit Hilfe der PRINT#-Anweisung in einer geschachtelten Schleife; anschließend jeweils Sprung ins Unterprogramm 2000 zur Fehlerkontrolle

Satz 300 :

Schließen der Datei (es hätte auch die Anweisung CLOSE 8 genügt)

Satz 310-320 :

Beendigung des Hauptprogramms

Satz 500-580 :

Daten der Datenmatrix (satzweise)

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste gedrückt wird

Satz 2000-2110 :

Unterprogramm zum Abfangen von Fehlern bei der Diskettenbenutzung; ggf. Beendigung des Programms

o Programmergebnisse

Wenn dieses Programm gestartet wird, geschieht zunächst das gleiche wie in Programm S33.

Danach erscheint die folgende Meldung auf dem Bildschirm :

ZUM SPEICHERN BITTE DATENDISKETTE EIN-  
LEGEN UND DANN DIE LEERTASTE DRUECKEN !

Hat der Benutzer diese Anweisung befolgt, kommt das Laufwerk in Gang, d.h. es wird gespeichert und danach meldet das Programm :

SPEICHERUNG BEENDET

ENDE DES PROGRAMMS

Kommt es bei der Benutzung der Disketten zu Fehlern, so wird dagegen der Programmablauf mit einer erläuternden Meldung abgebrochen.

## o Ergänzungen

In diesem Arbeitsschritt können wir uns darauf beschränken auf die entsprechenden Ausführungen beim Programm S33 hinzuweisen. Zusätzlich ist hier anzumerken, daß in Satz 270 (OPEN-Anweisung) der erste und der dritte Parameter und auch der Name der Datei (DATEN1) anders gewählt werden können (vergleiche dazu das Floppy-Disk-Bedienungshandbuch, S.17); gegebenenfalls muß dann entsprechend auch Satz 300 geändert werden.

### 4.3 Dateiverlängerung

Bei statistischen Anwendungsproblemen steht man recht häufig vor der Notwendigkeit, einem Datenbestand, der schon existiert (zum Beispiel wegen eines früheren Datenverarbeitungsprozesses), weitere Daten anzufügen, den Datenbestand also zu verlängern.

Wenn man, den Beispielen des Kapitels 3 folgend, die Daten nur in DATA-Statements bereitgestellt hat, also ohne die interaktive Nutzung externer Speicher, so ist die Vorgehensweise sehr einfach:

Man lädt das entsprechende Programm per LOAD-Kommando, fügt neue DATA-Statements den schon vorhandenen hinzu (auf diese Weise kann man offenbar auch Daten "einschieben" und nicht nur Daten "anhängen") und speichert das so veränderte Programm, bei dem natürlich auch die Anweisungen zur Verarbeitung der Daten auf den geänderten Datenumfang

reagieren müssen, unter neuem Namen per SAVE-Kommando.

Bei interaktiver Diskettennutzung ist aber zumindest das Anhängen von Daten eine recht leicht lösbare Aufgabe, wie das folgende Beispiel zeigt.

Es ist dabei aber besonders zu beachten, daß neue Datensätze an die schon bestehende Datei angefügt und nicht etwa zwischen bestehende Sätze eingeschoben werden. Diese zweite Aufgabe ist bei sequentiellen Dateien, im Gegensatz zu Random - access-Dateien nur recht schwierig zu lösen.

Man müßte zu diesem Zweck die gesamte Datei einlesen, sie mit Hilfe eines eigens entworfenen BASIC-Programms umorganisieren und die so entstehende neue Datei wieder komplett abspeichern, wobei die alte Datei dann gelöscht werden kann. Diese etwas umständliche Prozedur soll hier nicht besprochen werden, sondern wir beschränken uns hier auf das Anfügen von Daten.

#### o Vorstellung des Problems

Die Datenmatrix, die schon im vorangegangenen Beispiel benutzt wurde (Größe, Gewicht und Blutdruck von 8 befragten Personen), soll eingelesen und auf Diskette gespeichert werden.

Zusätzlich soll die Möglichkeit geschaffen werden, weitere Datensätze anzufügen - beispielsweise werden weitere Personen nach den genannten drei Variablen befragt - und diese ebenfalls auf Diskette zu speichern.

## o . Problemanalyse

Bezüglich der ersten Teilaufgabe unterscheidet sich diese Problemstellung nicht von der vorangegangenen.

Was das Anhängen von Datensätzen betrifft, treten auch keine besonderen Schwierigkeiten auf :

Wir veranlassen das Programm, über INPUT - Statements je drei Werte (Größe, Gewicht und Blutdruck) anzufordern, und lassen dann diese drei Werte speichern, wobei in der OPEN-Anweisung als letzter Parameter der Buchstabe A (für APPEND = Anhängen), anstelle von W (für WRITE = Schreiben) auftauchen muß.

Auch bei diesem Programm wird wieder Sorge dafür getragen, daß Fehler bei der Diskettennutzung angezeigt werden und ggf. das Programm abgebrochen wird (siehe dazu die Problemanalyse zum vorangegangenen Beispiel).

o. Programm

```
10 REM S42-ANHAENGEN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(9)"#BEISPIEL S42-ANHAENGEN#"
70 PRINT:PRINT:PRINT:PRINT
80 PRINTTAB(2)"MIT DIESEM PROGRAMM WIRD DIE DATEN-"
90 PRINTTAB(1)"MATRIX AUS DEM PROGRAMM S33 AUF EINER"
100 PRINTTAB(3)"DISKETTE GESPEICHERT (SCHREIBEN). "
102 PRINT:PRINT"DANACH WIRD PER INPUT EIN WEITERER DA-"
104 PRINT"TENSATZ EINGEGEBEN UND AN DIE DATEI AN-"
106 PRINTTAB(15)"GEHAENGT."
110 GOSUB 1000:REM WARTEN
120 DIM X(8,3)
130 FOR I=1 TO 8
140 FOR J=1 TO 3
150 READ X(I,J)
160 NEXT J
170 NEXT I
180 PRINT"=FALL", "VAR.1", "VAR.2", "VAR.3":PRINT
190 FOR I=1 TO 8
200 PRINT I,
210 FOR J=1 TO 3:PRINTX(I,J),:NEXT J
220 NEXT I
230 PRINT:PRINT:PRINT"#ZUM SPEICHERN BITTE
DATENDISKETTE EIN-"
240 PRINT"LEGEN UND DANN DIE LEERTASTE DRUECKEN !:"
250 PRINT:PRINT:PRINT
260 GET A$:IF A$="" THEN 260
265 OPEN 15,8,15
270 OPEN 8,8,8,"DATEN2,S,W"
275 GOSUB 2000:REM DISK
280 FOR I=1 TO 8:FOR J=1 TO 3
290 PRINT#8,X(I,J):GOSUB 2000:NEXT J:NEXT I
300 CLOSE 8,8,8
305 PRINTCHR$(147)
310 PRINT:PRINTTAB(10)"SPEICHERUNG BEENDET"
```

```

320 PRINT:PRINT:PRINT"■ ANHAENGEN EINES WEITEREN
SATZES?"
330 PRINT:PRINT"EINZUGEBEN SIND DREI WERTE : ":PRINT
340 DIM Y(3)
350 INPUT " GROESSE (RETURN)";Y(1)
360 INPUT " GEWICHT (RETURN)";Y(2)
370 INPUT " BLUTDRUCK(RETURN)";Y(3)
380 PRINT:PRINT:PRINT" ES ERFOLGT DIE SPEICHERUNG"
390 OPEN 8,8,8,"DATEN2,S,A"
395 GOSUB 2000:REM DISK
400 FOR J=1 TO 3:PRINT#8,Y(J):GOSUB 2000:NEXT J
410 CLOSE 8,8,8
420 PRINTCHR$(147):PRINTTAB(10)"SPEICHERUNG BEENDET"
430 PRINT:PRINT:PRINT"ENDE DES PROGRAMMS":END
500 REM DATEN
510 DATA 175,65,125
520 DATA 180,88,132
530 DATA 185,92,128
540 DATA 165,66,140
550 DATA 172,78,138
560 DATA 178,75,118
570 DATA 186,90,130
580 DATA 173,69,143
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A#: IF A#="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

```

2000 REM UP DISK
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINTCHR$(147)
2040 PRINTTAB(12)"DISKETTENFEHLER":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG MIT DEM"
2080 PRINTTAB(13)"KOMMANDO RUN"
2090 PRINT:PRINT:PRINT
2100 CLOSE 8:CLOSE 15
2110 PRINTTAB(10)"ENDE DES PROGRAMMS":END

```

o Variablenliste

A\$ = Tastatureingabe

EM\$, EN, ES, ET = Informationen aus dem Disketten-Fehlerkanal

I = Laufindex (Fälle)

J = Laufindex (Variablen)

X(,) = Werte der Datenmatrix (doppelt indiziert)

Y() = Werte des anzuhängenden Datensatzes (einfach indiziert)

o Programmbeschreibung

Satz 10-106 :

Überschrift und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste betätigt wird

Satz 120-170 :

Dimensionieren und Einlesen der ursprünglichen Datenmatrix

Satz 180-220 :

Kontrollausgabe der Datenmatrix auf dem Bildschirm

Satz 230-260 :

Abwarten, bis der Benutzer, nach entsprechender Aufforderung, die Datendiskette eingelegt hat

Satz 265 :

Öffnen des Disketten-Fehlerkanals

Satz 270-310 :

Speicherung der Datenmatrix auf der Datendiskette

Satz 320-330 :

Ausgabe von Erläuterungen zum Anhängen eines Datensatzes

Satz 340 :

Reservierung von Speicherplatz für einen einzugebenden Datensatz

Satz 350-380 :

Anforderung der drei Werte des anzuhängenden Datensatzes und erläuternde Mitteilung

Satz 390-395 :

Eröffnen der Datei, in der schon die Datenmatrix steht (DATEN1), nun aber zum Zwecke des Anhängens (deshalb Parameter A in der OPEN-Anweisung); dazu Sprung ins Unterprogramm 2000 zur Diskettenfehler-Kontrolle

Satz 400-410 :

Speicherung der drei eingegebenen Werte und Schließen der Datei DATEN1

Satz 420-430 :

Beendigung des Hauptprogramms

Satz 500-580 :

Daten der ursprünglichen Datenmatrix

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 2000-2110 :

Unterprogramm zum Abfangen von Fehlern bei der Diskettennutzung und ggf. Programmabbruch

o Programmergebnisse

Dieses Programm verhält sich zunächst genauso, wie das vorangegangene Programm.

Nach der Speicherung der Datenmatrix erfolgt die Meldung :

ANHAENGEN EINES WEITEREN SATZES

EINZUGEBEN SIND DREI WERTE :

GROESSE	?
GEWICHT	?
BLUTDRUCK	?

Bei jedem Fragezeichen muß der Benutzer einen Zahlenwert eingeben. Wenn dies erfolgt ist, erscheint auf dem Bildschirm die Meldung :

ES ERFOLGT DIE SPEICHERUNG

und durch das Anlaufen des Diskettenlaufwerks erkennt man, daß nun tatsächlich der Datensatz gespeichert wird.

Mit den schon bekannten Meldungen wird dann das Programm beendet.

## o Ergänzungen

Unter ergänzenden Gesichtspunkten verweisen wir zunächst auf alle Anmerkungen bei denjenigen Programmen, die sich der gleichen Datenmatrix bedienen.

Zusätzlich sind hier die folgenden Hinweise angebracht :

Es fällt bei dem Studium dieses Programms sofort auf, daß es sinnvoll sein dürfte, wenn dafür Sorge getragen wird, daß nicht nur ein Datensatz, sondern bei Bedarf auch mehrere Datensätze an die schon vorhandene Datei angehängt werden können.

Man kann dies im Prinzip so erreichen, daß der Programmteil 320-420 bei Bedarf mehrfach durchlaufen wird. Im nächsten Abschnitt werden wir ein entsprechendes Programm vorstellen.

Weiterhin ist es sinnvoll, jeden Datensatz, bevor er auf der Diskette gespeichert wird auf dem Bildschirm auszugeben, um dem Benutzer Gelegenheit zu geben, eventuelle Eingabefehler zu korrigieren, bevor ein fehlerhafter Satz gespeichert wird (nachträgliche Korrekturen schon fehlerhaft gespeicherter Daten sind viel zu umständlich). Auch diese Anregung werden wir im folgenden Programm aufgreifen.

Schließlich ist zu berücksichtigen, daß der Benutzer, wenn er Datensätze an eine schon existierende Datei anfügen möchte, nicht immer genau die Datei mit dem Namen DATEN1 meint, sondern vielleicht auch einmal eine andere Datei.

Dies kann dadurch beachtet werden, daß man den Namen der zu verlängernden Datei nicht fest vorgibt (wie wir im Satz 390 des obigen Programms), sondern vom Benutzer über ein INPUT-Statement den Namen der zu bearbeitenden Datei anfordert. Auch

dies soll im folgenden Beispiel mit berücksichtigt werden.

#### 4.4 Generelle Eingabe

Gemäß der vorbereitenden Programme der vorangegangenen Abschnitte und unter Berücksichtigung der dort vorgetragenen Ergänzungswünsche soll nun ein generelles Eingabeprogramm mit Speicherung der eingegebenen Daten auf Diskette durch das Programm selbst vorgestellt werden.

##### o Vorstellung des Problems

Es soll ein Programm erarbeitet werden, das zur Eingabe beliebiger Daten und deren Speicherung auf einer Datendiskette taugt. Dabei sollen Korrekturmöglichkeiten vorgesehen werden und das Programm soll in der Lage sein, zwischen der Erstanlage einer Datei bzw. der Verlängerung einer schon gegebenen Datei zu unterscheiden.

Der Einfachheit halber soll diese Problemstellung auf die Eingabe nur numerischer Daten beschränkt werden.

## o Problemanalyse

Im Prinzip unterscheidet sich die Problemanalyse für dieses Problem nicht wesentlich von den vorangegangenen Aufgaben.

Ein erster Unterschied besteht allerdings darin, daß beliebige Daten eingegeben werden sollen, d.h. wir müssen entsprechende INPUT-Anweisungen vorsehen und können nicht über DATA- und READ-Anweisungen einlesen.

Ein zweiter Unterschied wird dadurch gegeben, daß wir Korrekturmöglichkeiten zulassen wollen. Dies bedeutet, daß zum Beispiel nach der Eingabe eines jeden Datensatzes der Benutzer zunächst vom Programm gefragt wird, ob alles in Ordnung ist. Ist dies der Fall und soll noch ein weiterer Datensatz eingegeben werden, so kann diese Eingabe erfolgen.

Ist hingegen nicht alles in Ordnung, so muß das Programm abfragen, welcher der eingegebenen Werte falsch ist und dieses Speicherfeld dann zur Neubelegerung in einem INPUT-Statement anbieten. Danach wird der Datensatz zu Kontrollzwecken ausgegeben und erneut danach gefragt, ob alles in Ordnung ist usw.

Ein dritter und letzter Unterschied zu den bisherigen Programmen besteht darin, daß sowohl das Neu-Anlegen einer Datei, wie auch die Verlängerung einer Datei möglich sein soll. Dies wird erreicht, indem das Programm eine entsprechende Frage an den Benutzer richtet und je nach Antwort mit dem Parameter W oder A in der OPEN-Anweisung arbeitet.

Diese Unterscheidung ist übrigens in diesem Programm ganz generell einsetzbar, indem wir - auch bei der Neu-Anlage einer Datei - nur den ersten Datensatz mit dem Parameter W in der OPEN-Anweisung eingeben und alle (eventuell) folgenden mit dem Parameter A anhängen.

o Programm

```
10 REM S43-EINGABE
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(10)"#BEISPIEL S43-EINGABE#";
70 PRINT:PRINT:PRINT:PRINT
80 PRINTTAB(3)"GENERELLES PROGRAMM ZUM ANLEGEN "
90 PRINTTAB(3)"UND/ODER ZUR VERLAENGERUNG EINER"
100 PRINTTAB(10)"SEQUENTIELLEN DATEI."
102 PRINT:PRINT:PRINT:PRINT
104 PRINTTAB(10)"#INFORMATIONEN (J/N)#";
106 GET B$:IF B$<>"J" AND B$<>"N" THEN 106
108 IF B$="N" THEN PRINTCHR$(147):GOTO 210
110 PRINTCHR$(147)
120 PRINT"BEI DIESEM PROGRAMM MUSS DER BENUTZER"
130 PRINT"ALS INFORMATIONEN VORGEHEN : ":PRINT
140 PRINT"      NAMEN DER DATEI"
160 PRINT"      ANZAHL DER VARIABLEN":PRINT
170 PRINT"WENN EINE SCHON EXISTIERENDE DATEI VER-"
180 PRINT"LAENGERT WERDEN SOLL, MUSS ER DEREN"
190 PRINT"NAMEN KENNEN UND ANGEBEN.":PRINT:PRINT
192 PRINT:PRINT:PRINT"ACHTUNG : ":PRINT
194 PRINT"ES KOENNEN NUR NUMERISCHE DATEN EINGE-"
196 PRINT"GEBEN WERDEN !"
200 GOSUB 1000:REM WARTEN
210 PRINT"WIRD EINE NEUE DATEI ANGELEGT (N)"
220 PRINT"ODER EINE DATEI VERLAENGERT (V) ?"
230 PRINT:PRINT:PRINTTAB(9)"#BITTE N ODER V#";
235 GET B$
236 IF B$<>"N" AND B$<>"V" THEN 235
238 OPEN 15,8,15
240 PRINT:PRINT:INPUT"NAME DER DATEI (RETURN)";N$
245 PRINT:PRINT:PRINT
250 PRINT:PRINT"DATENDISKETTE EINLEGEN UND LEERTASTE !";
260 GET A$:IF A$="" THEN 260
270 GOSUB 2000:REM EINGABE
280 PRINT:PRINT:PRINT
290 PRINT"      DATENEINGABE UND DATENSPEICHERUNG"
300 PRINTTAB(13)"SIND BEENDET."
310 PRINT:PRINT:PRINT
320 PRINT:PRINTTAB(10)"ENDE DES PROGRAMMS":END
```

```

1000 REM UP WARTEN      *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
2000 REM UP EINGABE    *****
2010 PRINTCHR$(147)
2020 I=1
2030 INPUT "WIEVIELE WERTE PRO FALL (RETURN)";V
2040 DIM X(V) :PRINT:PRINT
2050 PRINT"EINGABE DER WERTE :":PRINT
2060 IF B$="V" THEN I=2
2070 FOR J=1 TO V
2080 PRINT"    VARIABLE NR. ";J;:INPUT"(RETURN)";X(J)
2090 NEXT J
2100 PRINT:PRINT:PRINT
2110 GOSUB 3000:REM FEHLERKONTROLLE
2120 PRINT:PRINT:PRINT
2130 IF B$="N" AND I=1 THEN GOSUB 4000:REM OPEN
2140 IF B$="N" AND I>1 THEN GOSUB 5000:REM APPEND
2150 IF B$="V" THEN GOSUB 5000:REM APPEND
2160 PRINT:PRINTTAB(9)"DATENSATZ GESPEICHERT":PRINT:PRINT
2170 PRINTTAB(7)"NOCH EIN DATENSATZ ? (J/N) "
2180 GET A$
2185 IF A$<>"J" AND A$<>"N" THEN 2180
2190 IF A$="N" THEN 2210
2200 I=I+1: PRINT:PRINT:PRINT:GOTO 2050
2210 RETURN
3000 REM UP FEHLERKONTROLLE *****
3010 PRINTCHR$(147)
3020 FOR J=1 TO V:PRINTJ,X(J) :NEXT J
3030 PRINT:PRINT"ALLE WERTE IN ORDNUNG ? (J/N) "
3040 GET A$
3045 IF A$<>"J" AND A$<>"N" THEN 3040
3050 IF A$="J" THEN 3110
3060 PRINT:PRINT"WELCHER WERT IST FALSCH ?"
3070 INPUT"BITTE NUMMER ANGEBEN (RETURN)";Z
3080 PRINT:PRINT"FALSCHER WERT      : ";X(Z)
3090 INPUT"■KORREKTER WERT (RETURN) ■";X(Z)
3100 GOTO 3010
3110 RETURN

```

```

4000 REM UP NEUANLAGE *****
4010 OPEN 8,8,8,N#+",S,W"
4020 GOSUB 6000:REM DISK
4030 FOR J=1 TO V:PRINT#8,X(J):GOSUB 6000:NEXT J
4040 CLOSE 8,8,8
4050 RETURN
5000 REM UP VERLAENGERN *****
5010 OPEN 8,8,8,N#+",S,A"
5020 GOSUB 6000:REM DISK
5030 FOR J=1 TO V:PRINT#8,X(J):GOSUB 6000:NEXT J
5040 CLOSE 8,8,8
5050 RETURN
6000 REM UP DISK *****
6010 INPUT#15,EN,EM$,ET,ES
6020 IF EN=0 THEN RETURN
6030 PRINTCHR$(147)
6040 PRINTTAB(12)"DISKETTENFEHLER:":PRINT:PRINT
6050 PRINTEN;EM$;ET;ES
6060 PRINT:PRINT:PRINT
6070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG MIT DEM"
6080 PRINTTAB(13)"KOMMANDO RUN"
6090 PRINT:PRINT:PRINT
6100 CLOSE 8:CLOSE 15
6110 PRINTTAB(10)"ENDE DES PROGRAMMS":END

```

o Variablenliste

- A\$ = Tastatureingabe (bzw. J/N für Ja/Nein)  
B\$ = Antworteingabe für neue Datei bzw. Ver-  
längerung einer schon bestehenden Datei  
C\$ = J/N für Ja/Nein bei der Fehlerkontrolle  
EM\$, EN, ES, ET = Informationen aus dem Disket-  
ten-Fehlerkanal  
I = Laufindex (Fälle)  
J = Laufindex (Variablen)  
N\$ = Name der Datei  
V = Anzahl der Variablen pro Datensatz  
X() = Werte eines Datensatzes  
Z = Kenn-Nummer des nach der Fehlerkontrolle  
eventuell zu korrigierenden Wertes

o Programmbeschreibung

Satz 10-100 :

Überschrift und Erläuterungen

Satz 102-108 :

Abfrage, ob nähere Informationen zum Programm ge-  
wünscht werden; wird werder J noch N eingege-  
ben, wartet das Programm; wird N eingegeben, er-  
folgt nach Löschen des Bildschirms ein Sprung zu  
Satz 210, andernfalls (bei Eingabe von J) geht es  
bei Satz 110 weiter

Satz 110 :

Löschen des Bildschirms

Satz 120-196 :

Weitere Erläuterungen

Satz 200 :

Sprung ins Unterprogramm 1000 zum Abwarten

Satz 210-236 :

Abfrage, ob eine neue Datei angelegt werden soll, oder ob eine bestehende Datei verlängert wird; je nach Antwort wird die Variable B\$ mit "N" (neue Datei) oder "V" (Verlängerung) belegt; bei unzulässiger Eingabe wartet das Programm weiter ab (Satz 236)

Satz 338 :

Eröffnen des Fehlerkanals

Satz 240-245 :

Anforderung des Namens, den die Datei trägt (bei Verlängerung) oder tragen soll (bei Neu-Anlage)

Satz 250-260 :

Aufforderung, die Datendiskette einzulegen und Abwarten

Satz 270 :

Sprung ins Unterprogramm 2000 zur Dateneingabe

Satz 280-310 :

Beendigung des Hauptprogramms

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

Satz 2000 :

Beginn des Unterprogramms zur Dateneingabe

Satz 2010 :

Löschen des Bildschirms

Satz 2020 :

Belegung des Laufindex I, der die Fälle zählen  
soll, mit dem Wert 1

Satz 2030 :

Anforderung der Zahl der Werte (Variablen) pro  
Fall; Belegung des Feldes V mit diesem Wert

Satz 2040-2050 :

Dimensionierung für einen Datensatz und Meldung,  
daß nun die Werte eingegeben werden sollen

Satz 2060 :

Wenn die Variable B\$ mit "V" belegt ist, wenn es  
also um die Verlängerung einer existierenden Da-  
tei gehen soll, wird der Zähler I mit dem Wert 2  
belegt, damit das Programm zum Anhängen übergeht

Satz 2070-2100 :

Eingabe des Datensatzes mit INPUT-Anweisungen

Satz 2110 :

Sprung ins Unterprogramm 3000 zur Fehlerkontrolle

Satz 2120-2150 :

Verzweigung in die Unterprogramme 4000 oder 5000, je nachdem, ob eine neue Datei angelegt oder ob eine schon bestehende Datei verlängert werden soll

Satz 2160-2190 :

Meldung, daß der jeweilige Datensatz gespeichert ist und Abfrage, ob noch ein weiterer Datensatz aufgenommen werden soll; wenn nein, weiter bei Satz 2210; wenn ja, weiter bei Satz 2200 (bei unzulässiger Eingabe wartet das Programm ab)

Satz 2200 :

Erhöhung des Zählers I um 1 (nächster Fall) und Rücksprung zum Satz 2050

Satz 2210 :

Beendigung des Unterprogramms Dateneingabe und Rücksprung ins Hauptprogramm

Satz 3000 :

Beginn des Unterprogramms zur Fehlerkontrolle

Satz 3010 :

Löschen des Bildschirms

Satz 3020 :

Ausgabe aller Werte eines Datensatzes mit Numerierung (Variable J)

Satz 3030-3050 :

Abfrage, ob alle Werte in Ordnung sind; wenn ja, weiter bei Satz 3110; wenn nein, weiter bei Satz 3060 (bei unzulässiger Antwort wartet das Programm ab)

Satz 3060-3070 :

Anforderung der laufenden Nummer des falschen Wertes

Satz 3080-3090 :

Kontrollausgabe des als falsch bezeichneten Wertes und Anforderung des korrekten Wertes, der dann in dem gleichen Feld gespeichert wird und somit den dort befindlichen falschen Wert ersetzt

Satz 3100 :

Rücksprung zum Satz 3010 zur erneuten Überprüfung (es könnte ja sein, daß noch ein weiterer Wert in diesem Datensatz falsch ist, oder daß bei der Korrektur selbst wiederum ein Fehler passiert ist)

Satz 3110 :

Rücksprung ins rufende Programm; dieser Satz wird nur erreicht, wenn alle eventuellen Fehler im Datensatz beseitigt sind;erst dann erfolgt der Rücksprung ins Unterprogramm 2000 (hinter den Satz 2110) und erst dann wird deshalb der (nun korrekte) Datensatz gespeichert

Satz 4000 :

Beginn des Unterprogramms zur Neuanlage einer Datei

Satz 4010 :

Eröffnen der Datei mit dem Namen N\$ (siehe Satz 240) zum Schreiben (Parameter W)

Satz 4020 :

Sprung ins Unterprogramm 6000 zur Kontrolle von Fehlern bei der Diskettennutzung

Satz 4030 :

Schreiben (Speichern) des Datensatzes

Satz 4040-4050 :

Schließen der Datei und Rücksprung

Satz 5000 :

Beginn des Unterprogramms zum Verlängern einer schon existierenden Datei

Satz 5010 :

Eröffnen der Datei mit dem Namen N\$ (siehe Satz 240) zum Verlängern (Parameter A)

Satz 5020 :

Sprung ins Unterprogramm 6000 zur Kontrolle von Fehlern bei der Diskettennutzung

Satz 5030 :

Schreiben (Speichern) des Datensatzes

Satz 5040-5050 :

Schließen der Datei und Rücksprung

Satz 6000-6110 :

Unterprogramm zur Kontrolle und eventuellen Ausgabe von Fehlern bei der Diskettennutzung und ggf. Programmabbruch

## o Programmsergebnisse

Bei der Darstellung der Ergebnisse können wir uns kurz fassen, indem wir einfach auf die Inhalte der PRINT- und der INPUT - Anweisungen des Programms verweisen :

Nach der Überschrift und den erläuternden Hinweisen, erscheint auf dem Bildschirm die Frage, ob eine neue Datei angelegt oder eine Datei verlängert werden soll (je nachdem ist "N" oder "V" einzugeben) und es wird der Name der Datei angefordert.

Dann erfolgt die Aufforderung, die Datendiskette einzulegen und danach die Leertaste zu betätigen.

Das Programm fragt dann, wieviele Werte pro Fall einzugeben sind (nehmen wir an, es handelte sich um die drei Variablen Größe, Gewicht und Blutdruck, dann müßten wir also mit 3 antworten) und fordert diese Werte für den ersten Datensatz an.

Haben wir drei Werte eingegeben, so werden diese auf einem geräumten Bildschirm im Rahmen der Fehlerkontrolle ausgegeben, also zum Beispiel :

1	175
2	72
3	134

ALLE WERTE IN ORDNUNG ?  
J ODER N EINGEBEN ?

Nehmen wir an, der zweite Wert wäre falsch, weil diese Person nicht 72 kg, sondern 73 kg wiegt. Wir antworten also mit N, worauf das Programm meldet :

WELCHER WERT IST FALSCH ?  
BITTE NUMMER ANGEBEN ?

Wir geben also die 2 ein und erhalten :

FALSCHER WERT : 72  
KORREKTER WERT : ?

Daraufhin können wir nun den Wert 73 eingeben und  
das Programm meldet :

1	175
2	73
3	134

ALLE WERTE IN ORDNUNG ?  
J ODER N EINGEBEN ?

Wenn wir jetzt J eingeben, so wird der Datensatz  
gespeichert - entweder als erster Datensatz  
einer neu etablierten Datei oder als zweiter oder  
späterer Datensatz einer schon existierenden Da-  
tei. Das Programm meldet dann :

DATENSATZ GESPEICHERT

NOCH EIN DATENSATZ ?  
J ODER N EINGEBEN ?

Geben wir J ein, wird die gesamte Prozedur - nun  
für einen weiteren Datensatz - wiederholt; geben  
wir hingegen N ein, so erhalten wir :

DATENEINGABE UND DATENSPEICHERUNG  
SIND BEENDET.

ENDE DES PROGRAMMS

o Ergänzungen

In diesem Abschnitt sind alle Anmerkungen zu wiederholen, die in den beiden vorhergehenden Programmen vorgetragen wurden. Wir dürfen deshalb auf die entsprechenden Ausführungen einfach verweisen.

## Kapitel 5 : Dateiverwaltung

### 5.1 Vorbemerkung

Wir haben verschiedene Möglichkeiten der Dateneingabe besprochen und müssen und jetzt der Frage zuwenden, wie man gespeicherte Daten wieder ausgeben kann.

Darüberhinaus werden wir in diesem Kapitel auch die Frage erörtern, wie man die Datenbestände, mit denen man statistisch arbeiten möchte, vor der weiteren Verarbeitung verändern kann, was zum Beispiel immer dann erforderlich ist, wenn man Fehler in einem derartigen Datenbestand entdeckt.

### 5.2 Datenausgabe

Wenn man einem BASIC-Programm Daten in der Weise bereitgestellt hat, wie das in Kapitel 3 besprochen wurde, also über LET-, INPUT- oder DATA- und READ-Anweisungen, dann ist die Datenausgabe einfach mit geeigneten PRINT-Anweisungen zu bewerkstelligen. Dazu dürfte ein Beispiel an dieser Stelle entbehrlich sein.

Wenn man hingegen Daten auf eine Diskette vom laufenden Programm aus eingegeben hat, ist anders vorzugehen, denn diese Daten können auch nur von einem laufenden Programm aus wieder geladen werden. Dies illustriert das folgende Programm :

o Vorstellung des Problems

Es soll ein BASIC-Programm entworfen werden, welches gespeicherte Daten von einer Datendiskette liest und sie zu Kontrollzwecken ausgibt.

o Problemanalyse

Prinzipiell sind die gleichen Überlegungen anzustellen wie bei den Eingabeprogrammen.

Bei der Kontrollausgabe stellen wir die Datensatzweise auf dem Bildschirm dar und es ist auch eine Ausgabe der gesamten Datenmatrix möglich, sofern die Zahl der Variablen nicht größer als 3 ist (auch in diesem Fall wäre eine Matrix-Ausgabe möglich, wenn wir die Matrixspalten enger zusammenrückten oder die Matrix immer nur bereichsweise ausgaben; darüber wurde an anderer Stelle schon gesprochen).

Beim Einlesen der Daten von der Diskette wird der Leseprozess dann abgebrochen, wenn keine Daten mehr gefunden werden. Dies ermöglicht die folgende Anweisung :

```
nn IF ST AND 64 THEN ..... Sprung aus der Leseschleife hinaus
```

Bei einer sequentiellen Datei kann es passieren, daß bei nicht korrekten Angaben des Benutzers die gelesenen Werte falsch zugeordnet werden :

Hat man beispielsweise 4 Fälle mit je 3 Werten, also insgesamt 12 Werte und gibt nun beim Start dieses Programms an, es handelte sich um 6 Fälle mit je 2 Werten, dann werden die Daten von der Diskette korrekt gelesen - aber natürlich völ-

lig fehlerhaft zugeordnet. Deshalb wird im folgenden Programm eine entsprechende Kontrollmöglichkeit vorgesehen.

**o Programm**

```
10 REM S51-DLESEN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(10)"BEISPIEL S51-DLESEN"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"PROGRAMM ZUR AUSGABE DES INHALTS EINER"
90 PRINTTAB(10)"SEQUENTIELLEN DATEI."
100 PRINT:PRINT:PRINT
110 GOSUB 1000:REM WARTEN
120 PRINT"BEI DIESEM PROGRAMM MUSS DER BENUTZER"
130 PRINT"FOLGENDE INFORMATIONEN ANGEBEN : "
140 PRINT:PRINT"      NAME DER DATEI"
150 PRINT "      ZAHL DER FAELE"
160 PRINT "      ZAHL DER WERTE PRO FALL"
170 PRINT:PRINT:PRINT
180 PRINT"LIEGEN DIESE INFORMATIONEN VOR ? (J/N)"
190 GET A$
195 IF A$<>"J" AND A$<>"N" THEN 190
200 IF A$="J" THEN GOTO 270
210 PRINTCHR$(147)
220 PRINT"BITTE ERST DIE NOTWENDIGEN INFORMATIO-"
230 PRINT"NEN BESORGEN (NAME DER DATEI, ZAHL DER"
240 PRINT"FAELLE UND ZAHL DER WERTE PRO FALL). "
250 PRINT:PRINT"DANN DAS PROGRAMM MIT DEM
      KOMMANDO RUN"
260 PRINTTAB(12)"ERNEUT STARTEN.":GOTO 530
```

```

270 PRINT:PRINT:PRINT"BITTE EINGEBEN
    (UND JEWEILS RETURN)":PRINT
280 INPUT"      NAME DER DATEI          : ";N#
290 INPUT"      ZAHL DER FAELLE        : ";N
300 INPUT"      ZAHL DER WERTE PRO FALL : ";V
310 DIM X(N,V)
314 PRINT:PRINT:PRINT
315 PRINT"☐DATENDISKETTE EINLEGEN UND LEERTASTE !☐"
316 GET A$:IF A$="" THEN 316
318 OPEN 15,8,15
320 OPEN 8,8,8,N#+",S,R"
325 GOSUB 2000:REM DISK
330 FOR I=1 TO N:FOR J=1 TO V
340 INPUT#8,X(I,J):IF ST AND 64 THEN GOTO 347
345 NEXT J:NEXT I
347 IF I<>N OR J<>V THEN GOSUB 3000
350 CLOSE 8:CLOSE 15:PRINTCHR$(147)
360 FOR I=1 TO N:PRINT"FALL NR. ";I:PRINT
370 FOR J=1 TO V:PRINT"      VARIABLE NR. ";J,X(I,J)
380 NEXT J:GOSUB 1000:REM WARTEN
390 NEXT I
400 IF V>3 THEN 530
420 PRINT"WIRD DIE AUSGABE DER KOMPLETTEN DATEN-"
430 PRINT"MATRIX GEWUENSCHT ? (J/N) ";
450 GET A$
455 IF A$<>"J" AND A$<>"N" THEN 450
460 IF A$="N" THEN 530
470 PRINTCHR$(147):PRINTTAB(11)"☐DATENMATRIX☐":PRINT:PRINT
480 PRINT"FALL",:FOR J=1 TO V:PRINT"VAR";J,:NEXT
    J:PRINT:PRINT
490 FOR I=1 TO N:PRINTI;" ",
500 FOR J=1 TO V:PRINTX(I,J),:NEXT J:PRINT
510 IF I/15=INT(I/15) THEN GOSUB 1000
520 NEXT I
530 PRINT:PRINT:PRINT
540 PRINTTAB(11)"ENDE DER AUSGABE":END

```

```

1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
2000 REM UP DISK
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINTCHR$(147)
2040 PRINTTAB(12)"■DISKETTENFEHLER!□":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG MIT DEM"
2080 PRINTTAB(13)"KOMMANDO RUN"
2090 PRINT:PRINT:PRINT
2100 CLOSE 8:CLOSE 15
2110 PRINTTAB(10)"ENDE DES PROGRAMMS":END
3000 REM UP FEHLER *****
3010 PRINTCHR$(147)
3020 PRINT"DIE ANGABEN ZUR ZAHL DER FAELLE ODER"
3030 PRINT"ZUR ZAHL DER VARIABLEN PRO FALL STIM-"
3040 PRINT"MEN MIT DEM DATEIINHALT NICHT UEBEREIN"
3050 PRINT:PRINT:PRINT:CLOSE 8:CLOSE 15
3060 PRINT:PRINT:PRINT
3070 PRINT" NEUSTART MIT RUN NACH UEBERPRUEFUNG !"
3080 PRINT:PRINT:PRINTTAB(9)"ENDE DES PROGRAMMS":END

```

o Variablenliste

A\$ = Tastatureingabe  
EM\$, EN, ES, ET = Informationen aus dem Disketten-  
Fehlerkanal  
I = Laufindex (Fälle)  
J = Laufindex (Variablen)  
N = Zahl der Fälle  
N\$ = Name der Datei  
V = Zahl der Variablen  
X(,) = Werte in der Datei

o Programmbeschreibung :

Satz 10-100 :

Überschrift und Erläuterungen

Satz 110 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis  
die Leertaste betätigt wird

Satz 120-170 :

Erläuternde Hinweise auf diejenigen Informationen,  
die das Programm zu seiner Fortsetzung benötigt

Satz 180-200 :

Abfrage, ob diese Informationen vorliegen; wenn  
ja, weiter bei Satz 270; wenn nein, weiter bei  
Satz 210; bei unzulässigen Eingaben wartet das  
Programm ab (Satz 195)

Satz 210-260 :

Löschen des Bildschirms und Hinweis darauf, daß die notwendigen Informationen zunächst besorgt werden müssen, bevor das Programm mit dem RUN-Kommando erneut gestartet werden kann; danach Sprung zum Satz 530

Satz 270-300 :

Anforderung der Informationen : Name der Datei (N\$), Zahl der Fälle (N) und Zahl der Variablen (V)

Satz 310 :

Dimensionierung

Satz 314-316 :

Aufforderung, die Datendiskette einzulegen und danach die Leertaste zu betätigen und Abwarten, bis dies geschehen ist

Satz 318 :

Öffnen des Disketten-Fehlerkanals

Satz 320 :

Eröffnen der Datei mit dem Namen N\$ zum Lesen (Parameter R für Read = Lesen)

Satz 325 :

Sprung ins Unterprogramm 2000 zur Prüfung, ob Fehler bei der Diskettennutzung begangen wurden

Satz 330-350 :

Lesen aller Daten mit der dafür zuständigen INPUT# - Anweisung; sind keine Daten mehr vorhanden, so

wird zum Satz 347 gesprungen (Satz 340), andernfalls wird weitergelesen (Satz 345); in diesem Zusammenhang wird auch geprüft, ob die Zahl der gelesenen Fälle oder die Zahl der gelesenen Variablen nicht mit den Angaben des Benutzers übereinstimmt (Satz 347); ist dies der Fall, erfolgt ein Sprung in das Unterprogramm 3000; Schließen der Datei und Löschen des Bildschirms (Satz 350)

Satz 360-390 :

Ausgabe der Werte fallweise, wobei nach jedem Fall das Programm wartet, bis der Benutzer die Leertaste betätigt

Satz 400 :

Wenn die Zahl der Variablen  $V$  größer als 3 ist, erfolgt ein Sprung zum Satz 530

Satz 410-460 :

Satz 410 wird erreicht, wenn  $V$  nicht größer als 3 ist; dann kann die gesamte Datenmatrix in übersichtlicher Form ausgegeben werden und das Programm fragt, ob dies geschehen soll; wenn nein, weiter bei Satz 530; wenn ja, weiter bei Satz 470

Satz 470-520 :

Ausgabe der gesamten Datenmatrix, wobei nach jedem 15. Satz das Programm abwartet, bis der Benutzer die Leertaste betätigt

Satz 530-540 :

Beendigung des Hauptprogramms

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis der Benutzer die Leertaste betätigt

Satz 2000-2110 :

Unterprogramm zur Feststellung eventueller Fehler bei der Diskettennutzung und ggf. Abbruch des Programms

Satz 3000-3080 :

Unterprogramm zur Ausgabe einer Fehlermeldung, wenn die Angaben des Benutzers bezüglich Zahl der Fälle oder Zahl der Variablen nicht zutreffen und Abbruch des Programms

#### o Programmergebnisse

Auch hier sind die Programmergebnisse direkt den Inhalten der PRINT- und INPUT-Anweisungen zu entnehmen.

Wenn wir mit einem früheren Programm die schon mehrfach benutzte  $8 * 3$  - Datenmatrix unter dem Namen DATEN1 auf der Datendiskette gespeichert haben, so können wir diese Daten nun wieder lesen, wenn wir auf die Anforderungen des Programms folgende Informationen eingeben :

BITTE EINGEBEN :

NAME DER DATEI	:	DATEN1
ZAHL DER FAELLE	:	8
ZAHL DER VARIABLEN	:	3

Wir erhalten dann als erste Ausgabe :

FALL NR. 1

VARIABLE NR.	1	175
VARIABLE NR.	2	65
VARIABLE NR.	3	125

BITTE DIE LEERTASTE DRUECKEN!

Betätigen wir danach die Leertaste, so erscheinen die Daten für den zweiten Fall usw.

Danach wird, weil V nicht größer als 3 ist, gefragt, ob wie die Ausgabe der Datenmatrix wünsch. Wenn wir mit J antworten, so ergibt sich :

DATENMATRIX

FALL	VAR1	VAR2	VAR3
1	175	65	125
2	180	88	132

usw.

5.3 Veränderung von Datenbeständen

Wir haben in dem generellen Eingabeprogramm gezeigt, wie man vor der Speicherung von Daten auf der Diskette Datenfehler vermeiden kann.

Trotzdem kann es passieren, daß man nach der Ausgabe mit dem vorangegangenen Programm feststellt, daß doch noch die Notwendigkeit besteht, den Datenbestand zu verändern. Dabei sind prinzipiell die folgenden Fälle zu unterscheiden :

1. Korrektur fehlerhafter Werte
2. Löschen von Werten
3. Hinzufügen von Werten
4. Löschen von Datensätzen
5. Hinzufügen von Datensätzen
6. Löschen von Arrays (Variablen)
7. Hinzufügen von Arrays

Es soll kurz darauf hingewiesen werden, wie bei der hier verwendeten sequentiellen Speicherung von Daten derartige Änderungen im Datenbestand herbeigeführt werden können, bzw. wie in dem Fall vorzugehen ist, wenn einfach nur ein gesamtes Programm samt seinem in DATA-Statements vorliegenden Datenbestand geladen wird.

Die zweite genannte Möglichkeit ergibt sich ja dann, wenn auf die interaktive Diskettennutzung verzichtet wird, wenn also ein Gesamtprogramm in den Arbeitsspeicher des Rechners geladen wird und nun in den DATA - Anweisungen Veränderungen erfolgen sollen.

Die oben genannten sieben Veränderungswünsche sind dann einfach so zu realisieren, daß die DATA -Anweisungen (bzw. die LET-Anweisungen, falls sie zur Dateneingabe benutzt wurden) entsprechend der jeweiligen Wünsche verändert werden und das so entstehende neue Programm per SAVE - Kommando wieder gespeichert wird.

Es ist dabei zu beachten, daß die Veränderung der Anzahl notwendigerweise Rückwirkungen hat auf :

- Dimensionierungen
- Länge von Lese- und Schreibschleifen
- Länge von Rechen- und Verarbeitungsschleifen etc.

Deshalb empfiehlt es sich generell, die Zahl der Fälle und die Zahl der Variablen (oder gegebenenfalls die Gesamtzahl aller Werte) immer als Variablen vorzugeben, die zu Programmbeginn belegt werden - so wie im vorangegangenen Programmbei-

spiel, wo  $N$  die Zahl der Fälle,  $V$  die Zahl der Variablen bezeichnete.

Wenn man nämlich so vorgeht, dann kann ein solches Programm nach der Änderung der DATA - Anweisungen rasch an die neue Lage angepaßt werden.

Wenn ganze Sätze und/oder Arrays gelöscht werden sollen, brauchen nicht unbedingt alle berührten DATA-Anweisungen verändert werden. Einfacher ist es, die zu löschenden Daten im Datenbestand zu belassen, sie aber durch entsprechende Gestaltung der Leseschleifen einfach zu "überlesen".

Folgende Rechen- und Ausgabeschleifen müssen dann entsprechend verkürzt werden.

Schematisch zeigt diese Veränderungsmöglichkeiten das folgende Programm :

o Vorstellung des Problems

Im folgenden Programm soll eine über DATA - Anweisungen bereitgestellte  $5 * 3$  - Datenmatrix nach Wunsch des Benutzers verändert werden.

Wir beschränken uns dabei auf die folgenden drei Möglichkeiten :

1. Löschen eines Datensatzes (Fall)
2. Löschen einer Variablen (Array)
3. Löschen eines Satzes und eines Arrays

Das Löschen einzelner Werte übergehen wir hier, weil es einfach durch Nichtberücksichtigung dieser Werte bei Ausgabe und Rechnungen zu bewerkstelligen ist; ebenso betrachten wir nicht das Hinzufügen von Werten, Sätzen oder Arrays, weil dies nur durch Hinzufügen von DATA-Anweisungen möglich ist.

## o Problemanalyse

Das geplante Programm muß die folgenden Arbeitsschritte durchlaufen :

1. Einlesen der Datenmatrix
2. Frage an den Benutzer, welche Art der Veränderung er wünscht.

Dies gibt uns Gelegenheit, die sog. Menütechnik vorzuführen, die hier so verwendet wird, daß - je nach Auswahl des Benutzers - unterschiedliche Unterprogramme angesprungen werden.

Es muß dabei Sorge dafür getragen werden, daß in dem Fall, in dem das Programm den Benutzer auffordert, eine Ziffer zur Menüauswahl einzugeben, Eingabefehler abgefangen werden.

Ein solcher Fehler kann darin bestehen, daß der Benutzer eine nicht zulässige Ziffer oder z.B. sogar einen Buchstaben eingibt, was in der GET-Anweisung im Gegensatz zur INPUT-Anweisung nicht zu automatischen Fehlermeldungen führt.

Um derartige Fehler abzufangen, benutzen wir in in der entsprechenden GET-Anweisung eine Stringvariable und bestimmen mit der VAL-Funktion die führenden Ziffern in diesem Stringfeld. Wenn sich dabei der Wert 0 ergibt, dann deshalb, weil der Benutzer (irrtümlich) keine Menüziffer, sondern z.B. einen Buchstaben eingegeben hat.

Man kann daraufhin eine erneute Eingabe im Programm vorsehen - genauso, wenn der Benutzer eine nicht zulässige Ziffer eingegeben hat.

Sind beispielsweise nur die Menüziffern 1 bis 5 erlaubt, dann kann die entsprechende Anweisungsfolge so aussehen :

```
255 GET A$ : IF A$ = "" THEN 255
256 A = VAL(A$) : IF A = 0 THEN PRINT "EINGABEFEH
LER; NOCHMAL BITTE" : GOTO 255
257 IF A >5 THEN PRINT "ZU HOHER WERT (MAX.5)" :
GOTO 255
```

Nach diesem Muster werden wir in den folgenden Programmen immer Eingabefehler bei der Eingabe von Menünummern abfangen.

### 3. Vollzug der jeweils gewünschten Änderung

Wie dabei im einzelnen vorzugehen ist, wurde in den einleitenden Bemerkungen zu diesem Abschnitt schon beschrieben.

Programmtechnisch sieht dies so aus, daß der Benutzer zum Beispiel angibt, daß der Fall Nr. Z gestrichen werden soll. In der folgenden Ausgabeschleife (und in eventuellen Rechenschleifen) schlägt sich dies dann einfach so nieder, daß bei der Ausgabe der Fälle die Ausgabe unterdrückt wird, wenn der Laufindex für die Fälle (I) den Wert Z aufweist.

Entsprechend ist auch bei den anderen Wünschen zur Dateiänderung zu verfahren.

o Programm

```
10 REM S52-AENDERN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(10)"#BEISPIEL S52-AENDERN#":PRINT
80 PRINT" IN DIESEM PROGRAMM WIRD GEZEIGT, WIE"
90 PRINT"EINE DATENMATRIX VERAENDERT WERDEN KANN."
100 PRINT:PRINT" EIN SPIELDATENBESTAND WIRD UEBER DATA"
110 PRINTTAB(8)"IN 900 FF. VORGEGEBEN."
120 PRINT:PRINT" DIE AENDERUNGSMOEGlichkeiten WERDEN "
130 PRINT" NACH DER AUSGABE DER GESAMTMATRIX ZUR"
140 PRINTTAB(10)"AUSWAHL ANGEBOten."
150 GOSUB 1000:REM WARTEN
160 DIM X(5,3)
170 FOR I=1 TO 5:FOR J=1 TO 3
180 READ X(I,J):NEXT J:NEXT I
190 PRINTTAB(13)"#DATENMATRIX#":PRINT
200 GOSUB 2000:REM AUSGABE
210 GOSUB 1000:REM WARTEN
220 GOSUB 3000:REM MENUE
230 REM UP EIN FALL *****
240 PRINTCHR$(147);S=0;Z=0:GOSUB 2000
250 PRINT:PRINT "WELCHER FALL (NR. ANGEBEN) : "
255 GET Z#:IF Z#=""THEN 255
256 Z=VAL(Z#):IF Z=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 250
260 IF Z>5 THEN PRINT"ZU HOHER WERT (MAX.5)":GOTO 250
270 PRINTCHR$(147):PRINT"FALL NR.";Z:PRINT:PRINT
280 FOR J=1 TO 3:PRINT" VAR.NR.";J,X(Z,J)
290 NEXT J:RETURN
```

```

300 REM UP EINE VARIABLE *****
310 PRINTCHR$(147):Z=0:S=0:GOSUB 2000
320 PRINT:PRINT"WELCHE VARIABLE (NUMMER) "
325 GET S$:IF S$=""THEN 325
326 S=VAL(S$):IF S=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 320
330 IF S>3 THEN PRINT"ZU HOHER WERT (MAX.3)":GOTO 320
340 PRINTCHR$(147):PRINT"VARIABLE NR. ";S:PRINT:PRINT
350 FOR I=1 TO S:PRINT"  FALL NR. ";I,X(I,S)
360 NEXT I
370 RETURN
400 REM UP FALL *****
410 PRINTCHR$(147):Z=0:S=0:GOSUB 2000
420 PRINT:PRINT "WELCHE ZEILE (FALL-NR.) "
425 GET Z$:IF Z$=""THEN 425
426 Z=VAL(Z$):IF Z=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 420
430 IF Z>5 THEN PRINT"ZU HOHER WERT (MAX.5)":GOTO 420
440 PRINTCHR$(147):PRINT"VERKUERZTE DATEI":PRINT
450 GOSUB 2000:REM AUSGABE
460 RETURN
500 REM UP VARIABLE *****
510 PRINTCHR$(147):Z=0:S=0:GOSUB 2000
520 PRINT:PRINT"WELCHE SPALTE (VAR.-NR.) "
525 GET S$:IF S$=""THEN 525
526 S=VAL(S$):IF S=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 520
530 IF S>3 THEN PRINT"ZU HOHER WERT (MAX.3)":GOTO 520
540 PRINTCHR$(147):PRINT"SCHMALERE DATEI":PRINT
550 GOSUB 2000:REM AUSGABE
560 RETURN
600 REM UP FALL UND VARIABLE *****
610 PRINTCHR$(147):Z=0:S=0:GOSUB 2000
620 PRINT:PRINT "WELCHE ZEILE (FALL-NR.) "
625 GET Z$:IF Z$=""THEN 625
626 Z=VAL(Z$):IF Z=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 620
630 IF Z>5 THEN PRINT"ZU HOHER WERT (MAX.5)":GOTO 620
640 PRINT:PRINT "WELCHE SPALTE (VAR.-NR.) "
645 GET S$:IF S$=""THEN 645
646 S=VAL(S$):IF S=0 THEN PRINT"EINGABEFehler;
    NOCHMAL":GOTO 640
650 IF S>3 THEN PRINT"ZU HOHER WERT (MAX.3)":GOTO 640
660 PRINTCHR$(147):GOSUB 2000:REM AUSGABE
670 RETURN

```

```

700 REM UP ENDE *****
710 PRINT:PRINT:PRINTTAB(10)"ENDE DER AUSGABE"
720 END
730 RETURN
900 DATA 178,65,132
910 DATA 185,84,128
920 DATA 180,78,136
930 DATA 165,59,132
940 DATA 172,76,140
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A#="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
2000 REM UP AUSGABE *****
2010 PRINT"FALL",:FOR J=1 TO 3
2020 IF J=S THEN 2040
2030 PRINT"V";J,
2040 NEXT J:PRINT:PRINT
2050 FOR I=1 TO 5
2060 IF I=Z THEN 2120
2070 PRINTI,
2080 FOR J=1 TO 3
2090 IF J=S THEN 2110
2100 PRINTX(I,J),
2110 NEXT J
2120 PRINT:NEXT I
2130 RETURN

```

```

3000 REM UP MENUE
3010 PRINTCHR$(147)
3020 PRINT"WAEHLEN SIE AUS FOLGENDEN AENDERUNGS-"
3030 PRINT"MOEGlichkeiten ; ":PRINT
3040 PRINT"  AUSGABE NUR EINES FALLES  (1)"
3050 PRINT"  AUSGABE NUR EINER VARIABLEN(2)"
3060 PRINT"  LOESCHEN EINES FALLES    (3)"
3070 PRINT"  LOESCHEN EINER VARIABLEN  (4)"
3080 PRINT"  LOESCHEN EINES FALLES"
3090 PRINT"  UND EINER VARIABLEN      (5)"
3100 PRINT:PRINT"  KEINE AUSWAHL          (6)"
3110 PRINT:PRINT:PRINT
3115 PRINTTAB(18)"BITTE NUMMER:"
3120 GET A$:IF A$="" THEN 3120
3122 A=VAL(A$):PRINT:PRINT
3125 IF A<1 OR A>6 THEN PRINT"
      FALSCH EINGABE; BITTE NOCHMAL":GOTO 3120
3130 ON A GOSUB 230,300,400,500,600,700
3140 GOSUB 1000:REM WARTEN
3150 GOTO 3020
3160 RETURN

```

o Variablenliste

A = Menü-Auswahl (Kennziffer)  
A\$ = Tastatureingabe  
I = Laufindex (Fälle)  
J = Laufindex (Variablen)  
S = Nummer der ausgewählten Matrix-Spalte  
X(,) = Werte in der Datenmatrix  
Z = Nummer der ausgewählten Matrix-Zeile

o Programmbeschreibung

Satz 10-140 :

Überschrift und Erläuterungen

Satz 150 :

Abwarten durch Sprung in das Unterprogramm 1000

Satz 160-180 :

Dimensionierung der Datenmatrix und Einlesen aller Werte

Satz 190 :

Ausgabe einer Überschrift

Satz 200 :

Sprung ins Unterprogramm 2000 zur Ausgabe der Datenmatrix

Satz 210 :

Abwarten

Satz 220 :

Sprung ins Unterprogramm 3000 zur Ausgabe des Menüs (Angebote zur Veränderung der Datenmatrix)

Satz 230 :

Beginn des Unterprogramms zur Ausgabe nur eines Falles; dieses Unterprogramm wird nach entsprechender Menüauswahl angesprungen (siehe Unterprogramm 3000, insbesondere Satz 3130)

Satz 240 :

Löschen des Bildschirms, Belegung der Hilfsvariablen Z und S mit 0 (Z und S bezeichnen die eventuell zu unterdrückenden Zeile und/oder Spalte der Datenmatrix) und Sprung ins Unterprogramm 2000 zur Ausgabe der Datenmatrix

Satz 250 :

Anforderung der Nummer des Falls, der ausgegeben werden soll

Satz 255-260 :

Erfolgt eine unzulässige Eingabe, wird eine Fehlermeldung ausgegeben und es erfolgt ein Rücksprung zu Satz 250

Satz 270-290 :

Ausgabe der gewünschten Fall-Nummer, aller Werte für diesen Fall und Rücksprung ins rufende Programm (d.h. weiter bei Satz 3140 im Unterprogramm 3000)

Satz 300-370 :

Dies ist das Unterprogramm, das die Ausgabe aller Werte für eine ausgewählte Variable bewerkstelligt; es ist im Prinzip genauso aufgebaut wie das vorangegangene Unterprogramm (siehe Satz 230-290)

Satz 400-460 :

Unterprogramm zur Unterdrückung einer Datenmatrix - Zeile (Fall) bei der Ausgabe; dazu wird die entsprechende Zeilennummer (Z) angefordert (Satz 420), damit diese Zeile später bei der Ausgabe (Unterprogramm 2000, angesprungen über Satz 450) übergangen werden kann

Satz 500-560 :

In entsprechender Weise wie das vorangegangene Unterprogramm dient dieses zur Unterdrückung einer Datenmatrix - Spalte (Array) bei der Ausgabe

Satz 600-670 :

Dieses Unterprogramm ermöglicht entsprechend, daß eine Zeile und eine Spalte der Datenmatrix bei der Ausgabe unterdrückt werden können

Satz 700-730 :

Unterprogramm zur Beendigung des Gesamtprogramms

Satz 900-940 :

Spieldaten der Ausgangs-Datenmatrix

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 2000 :

Beginn des Unterprogramms zur Ausgabe der Datenmatrix (entweder komplett oder in verkürzter Form)

Satz 2010-2040 :

Ausgabe der Kopfzeile der Datenmatrix; dabei wird die Spalte Nr. S unterdrückt (Satz 2020); die komplette Matrix wird also nur bei S=0 ausgegeben

Satz 2050-2120 :

Ausgabe der Datenmatrixwerte zeilenweise (I-Schleife; Fälle); dabei wird die Zeile Nr. Z unterdrückt (Satz 2060); alle Zeilen der Matrix (alle Fälle) werden also nur bei Z=0 ausgegeben;

in der J-Schleife (Satz 2080-2110) werden dann die Spaltenwerte pro Zeile ausgegeben mit Unterdrückung der Spalte Nr. S; ist S=0, werden alle Spalten besetzt

Satz 2130 :

Rücksprung ins jeweils rufende Programm

Satz 3000 :

Beginn des Unterprogramms, welches das Menü zur Auswahl gewünschter Veränderungen bereitstellt

Satz 3010-3110 :

Löschen des Bildschirms und Ausgabe der einzelnen Möglichkeiten zur Auswahl; es ist dabei zu beachten, daß wir als letzte Möglichkeit (Nr. 6) auch den Fall vorgesehen haben, daß der Benutzer keine Veränderungen und damit die Beendigung des Programmlaufs wünscht

Satz 3115-3125 :

Anforderung der Menü-Auswahl und Kontrolle auf fehlerhafte Eingabe

Satz 3130 :

Mit dem ON...GOSUB-Statement erfolgt ein Sprung

zum Unterprogramm 230 mit Menü-Nummer 1  
zum Unterprogramm 300 mit Menü-Nummer 2  
zum Unterprogramm 400 mit Menü-Nummer 3  
zum Unterprogramm 500 mit Menü-Nummer 4  
zum Unterprogramm 600 mit Menü-Nummer 5  
zum Unterprogramm 700 (Ende) mit Nummer 6

Dieses Statement erlaubt also die Programmierung von Mehrfachverzweigungen, was wir hier sehr nutzbringend verwenden können

Satz 3140 :

Nachdem das Programm aus dem jeweiligen, über das ON... GOSUB-Statement angesprungene Unterprogramm zurückkehrt, wird Satz 3140 erreicht, der dem Sprung ins Unterprogramm 1000 zum Abwarten dient

Satz 3150 :

Wenn der Benutzer die Leertaste betätigt hat, wird Satz 3150 erreicht, der den Rücksprung zum Satz 3020 bewirkt, d.h. erneut wird das Menü zur Veränderung der Ausgangsmatrix angeboten (da zu Beginn der jeweiligen Unterprogramme zur Veränderung im-

mer S und Z den Wert 0 zugewiesen erhalten, wird immer wieder die ursprüngliche Matrix und nicht etwa eine schon veränderte Matrix verändert; es ist leicht einzusehen, daß man diesen Programmablauf aber auch anders gestalten kann)

Satz 3160 :

Rücksprung ins rufende Programm

o Programmergebnisse

Wenn man dieses Programm startet, wird nach Überschrift und Erläuterungen zunächst die Ausgangsmatrix ausgegeben :

GESAMTE DATENMATRIX

FALL	V1	V2	V3
1	178	65	132
2	185	84	128
3	180	78	136
4	165	59	132
5	172	76	140

Nach einer abwartenden Programmunterbrechung wird dann das Menü ausgegeben :

WAEHLLEN SIE AUS FOLGENDEN AENDERUNGS-  
MOEGLICHKEITEN :

- AUSGABE NUR EINES FALLES (1)
- AUSGABE NUR EINER VARIABLEN(2)
- LOESCHEN EINES FALLES (3)
- LOESCHEN EINER VARIABLEN (4)
- LOESCHEN EINES FALLES  
UND EINER VARIABLEN (5)
- KEINE AUSWAHL (6)

BITTE NUMMER ?

Geben wir jetzt beispielsweise eine 3 ein, so des-  
halb, weil wir einen Fall aus der Matrix löschen  
möchten. Das Programm meldet jetzt :

WELCHE ZEILE (FALL-NR.) ?

Geben wir jetzt zum Beispiel eine 2 ein, so ergibt  
sich :

VERKUERZTE DATEI

FALL	V1	V2	V3
1	178	65	132
3	180	78	136
4	165	59	132
5	172	76	140

Nach dieser Ausgabe wird erneut das Menü angebo-  
ten, falls der Benutzer weitere Veränderungswün-  
sche hat.

In entsprechender Weise ergeben die anderen Unter-  
programme verkürzte Ausgaben. Am einfachsten ist

es, der Benutzer probiert die verschiedenen Möglichkeiten einfach einmal durch.

Wenn wir nach der erneuten Menü-Vorgabe die Auswahlziffer 6 eingeben, dann meldet das Programm

ENDE DER AUSGABE

und ist damit beendet.

o Ergänzungen

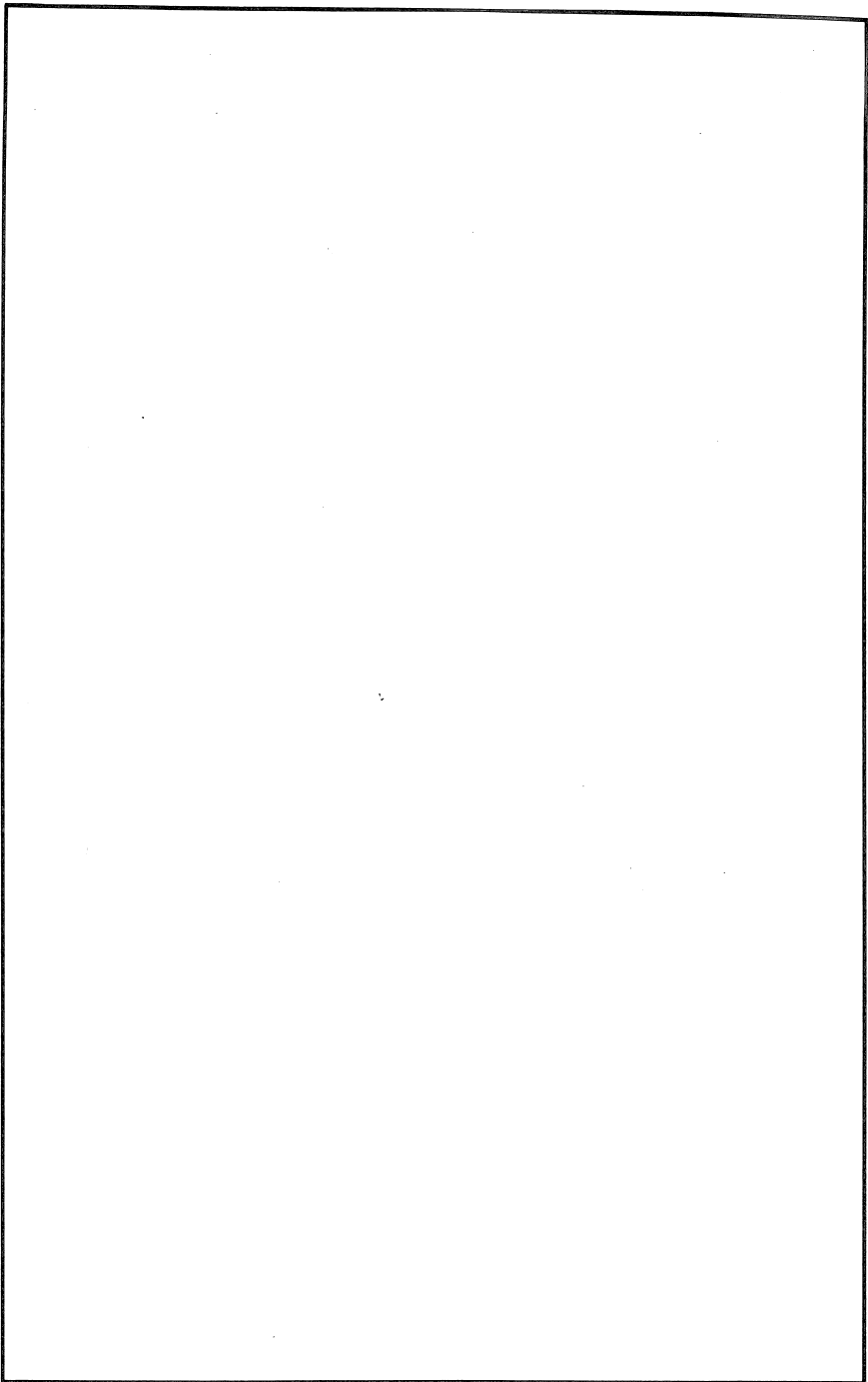
Es ist leicht einsichtig, welche Veränderungen das Programm erfahren müßte, wenn gleichzeitig auch mehrere Spalten und/oder Zeilen der Datenmatrix gelöscht werden sollen :Es müssen nur entsprechend mehr Markiervariablen belegt werden (entsprechend den Variablen S und Z im obigen Programm), die dann bei der Matrixausgabe (wie oben S und Z) steuernd eingreifen (Sätze 2020, 2060, 2090).

Generell ist bei diesem Programm wieder daran zu erinnern, daß es allgemeiner gehalten werden kann, wenn die Zahl der Fälle und die Zahl der Variablen, damit also die Dimension der Datenmatrix allgemein als N und V vorgegeben werden. Dann kann das Programm beliebig große Datenbestände bearbeiten und ist nicht auf eine 5\*3 - Matrix beschränkt (natürlich müssen dann auch die entsprechenden Datenbestände vorliegen).

Im Übrigen ist auf alle Ergänzungen hinzuweisen, die bei den vorangegangenen Programmen im Zusammenhang mit der Be- oder Verarbeitung einer Datenmatrix schon genannt wurden.

Schließlich ist besonders daran zu erinnern, daß das gleiche Programm auch zur Veränderung einer Datenmatrix verwendet werden kann, wenn diese

nicht - wie oben - über DATA-Anweisungen (oder über LET- oder INPUT-Anweisungen) bereitgestellt wird, sondern wenn die Daten interaktiv aus einer Diskettendatei gelesen werden.



## Kapitel 6 : Häufigkeitstabellen

### 6.1 Vorbemerkung

Mit diesem Kapitel wenden wir uns nun zum ersten Mal dem "engeren" Bereich der Statistik zu. Natürlich gehören auch die Erörterungen der vorangegangenen Kapitel in den Bereich statistischer Methoden - allein schon deshalb, weil ja vor jeder statistischen Auswertung Daten erhoben und bereitgestellt werden müssen.

Statistische Datenverarbeitung im engeren Sinne bedeutet aber "Verarbeitung schon vorhandener statistischer Daten"- und genau diesem Themenbereich wenden wir uns jetzt zu.

Es ist dabei im Prinzip unerheblich, ob die Daten direkt in einem BASIC-Programm eingegeben werden, oder ob sie von einer Diskettendatei gelesen werden. Der Einfachheit halber werden wir die folgenden Beispiele so anlegen, daß wir für jede statistische Auswertungsprozedur "Spieldaten" im jeweiligen Programm, zum Beispiel über DATA-Anweisungen vorgeben.

Da davon auszugehen ist, daß der Leser dieses Buches noch nicht Fachmann im Bereich der statistischen Methoden ist, werden wir dieses und die folgenden Kapitel immer zweiteilen :

In einem ersten Teil besprechen wir die jeweils zu behandelnden statistischen Methoden, im zweiten Teil stellen wir dann die entsprechenden BASIC - Programme vor.

Dies ermöglicht es, daß wir uns in den Arbeits-

schritten "Vorstellung des Problems", und "Problem-analyse" sehr kurz fassen können, weil die jeweiligen theoretischen Erörterungen dann schon vorliegen.

## 6.2 Statistische Grundlagen

Hat man einmal einen statistischen Datenbestand zusammengestellt, der dann ausgewertet werden soll, und hat man ihn - wie in den Kapiteln 3 und 4 beschrieben wurde - auf die ein oder andere Art in den Rechner eingegeben, so besteht eine der wichtigsten statistischen Auswertungsmethoden darin, diesen Datenbestand erst einmal übersichtlich darzustellen.

Man unterscheidet zwei Möglichkeiten der Datendarstellung :

- Tabellarische Darstellungen
- Graphische Darstellungen

In diesem Kapitel wenden wir uns ausschließlich den tabellarischen Darstellungen zu.

Weiterhin ist grundsätzlich - sowohl bei tabellarischen wie auch bei graphischen Darstellungen - zu unterscheiden, ob

- nur eine Untersuchungsvariable
- zwei Variablen gleichzeitig
- mehr als zwei Variablen gleichzeitig

zum Gegenstand der Betrachtung gemacht werden.

Im ersten Fall spricht man von univariater, im zweiten Fall von bivariater und im dritten Fall von multivariater Betrachtung.

Schließlich ist zu unterscheiden je nach Typus der

jeweiligen Untersuchungsvariablen. In der Statistik unterscheidet man die folgenden beiden Typen:

- diskrete Variablen
- stetige Variablen

Diskrete Variablen können nur streng voneinander getrennte Werte als Ausprägungen annehmen (typisches Beispiel : Kinderzahl befragter Familien), während stetige Variablen jeden beliebigen Wert annehmen können (Beispiel : Körpergröße befragter Personen).

Zur Vorbereitung der im zweiten Teil des Kapitels vorzustellenden Programme betrachten wir nun einige Beispiele von Häufigkeitsverteilungen, was uns zudem die Gelegenheit bietet, einige zentrale statistische Grundbegriffe vorzustellen.

#### 6.2.1 Verteilung einer diskreten Variablen

Hat man z.B. bei 500 zufällig ausgewählten Familien die jeweilige Kinderzahl erfragt, so erhält man Merkmalswerte dieser Variablen, die ganzzahlig (also diskret) zwischen 0 und vielleicht 6 (die Obergrenze hängt natürlich vom jeweiligen Datenbestand ab) liegen.

Diese Merkmalswerte kann man ordnen und den einzelnen Werten 0, 1, 2, ..., 6 die Häufigkeiten ihres Auftretens zuordnen. Berechnet man zusätzlich zu diesen absoluten Häufigkeiten die relativen Häufigkeiten (Prozentanteile), dann könnte sich eine Tabelle nach folgendem Muster ergeben :

## Häufigkeitsverteilung Kinderzahl

Kinderzahl	Häufigkeit	
	absolut	relativ
0	125	25.0
1	210	42.0
2	105	21.0
3	35	7.0
4	15	3.0
5	5	1.0
6	5	1.0
Summe	500	100.0

In der Statistik hat man sich angewöhnt, die einzelnen interessierenden Sachverhalte abzukürzen. So bezeichnet man mit

- $x_i$  die einzelnen Merkmalswerte
- $f_i$  die Häufigkeiten dieser Merkmalswerte
- $f_i'$  die relativen Häufigkeiten
- $n$  die Gesamtzahl der Beobachtungen  
(im Beispiel gilt :  $n = 500$ )

Offenbar besteht zwischen den absoluten Häufigkeiten  $f_i$  und den relativen Häufigkeiten  $f_i'$  der folgende mathematische Zusammenhang, der aus der Dreisatzrechnung bekannt sein dürfte :

$$f_i' = (f_i/n) \cdot 100$$

### 6.2.2 Verteilung einer stetigen Variablen

Beispiel einer stetigen Variablen ist die Körpergröße befragter Personen. Es gelten hier die gleichen Überlegungen wie oben mit einem wesentlichen Unterschied :

Da eine stetige Variable jeden beliebigen Wert annehmen kann, ist es nicht möglich, in die erste Spalte einer Tabelle nach obigem Muster alle Merkmalswerte aufzunehmen, weil es ja unendlich viele geben kann.

Stattdessen müssen Klassen gebildet werden, etwa nach dem folgenden Muster :

#### Häufigkeitsverteilung Körpergröße

Körpergröße (cm)	Häufigkeit
-----	-----
150 bis unter 160	25
160 bis unter 170	72
usw.	
-----	-----
Summe	n
=====	=====

Natürlich könnte auch eine andere Klasseneinteilung als sinnvoll erachtet werden (zum Beispiel 5 cm-Klassen) und natürlich können auch hier relative Häufigkeiten bestimmt werden.

Als Merkmalswerte  $x_i$  werden in einer derartigen klassifizierten Verteilung die jeweiligen Klassenmittelpunkte interpretiert.

### 6.2.3 Verteilung für zwei Variablen

Eine Häufigkeitsverteilung für zwei Variablen, eine sog. bivariate Verteilung wollen wir mit dem Beispiel von zwei diskreten Variablen zeigen.

Eine solche Verteilung entsteht, wenn man eine Reihe von Merkmalsträgern (zum Beispiel befragte Personen) nach zwei Merkmalen (Variablen) gleichzeitig befragt, also etwa nach Geschlecht und nach der bevorzugten politischen Partei.

Eine bivariate Tabelle sieht dann folgendermaßen aus :

#### Bivariate Häufigkeitsverteilung

Geschlecht Partei	m	w	Summe
-----			
CDU	39	37	76
SPD	44	46	90
FDP	6	5	11
Grüne	7	7	14
Sonstige	1	3	4
-----			
Summe	97	98	195
=====			

Wir sehen hier, daß insgesamt 195 Personen befragt wurden, von denen beispielsweise 76 die CDU als politische Partei bevorzugten, darunter wiederum 39 Männer und 37 Frauen - entsprechend sind alle anderen Werte in dieser Tabelle zu interpretieren.

Nach dem gleichen Muster können auch stetige Variablen in einer bivariaten Tabelle auftauchen; es müßte dann nur wieder, wie schon im univariaten Fall, klassifiziert werden.

#### 6.2.4 Multivariate Tabellen

Multivariate Tabellen werden sehr leicht so unübersichtlich, daß die Arbeit nicht lohnt, derartige Tabellen aufzubauen. In der Regel begnügt man sich in solchen Fällen mit der sog. Urliste, die nichts anderes als die uns schon bekannte Datenmatrix ist.

Deshalb brauchen wir auf diesen Spezialfall hier nicht weiter einzugehen, sondern erinnern stattdessen noch einmal daran, daß auch statistische Zeitreihen in Tabellenform dargestellt werden können (erste Spalte : Zeitpunkte oder Zeiträume; zweite Spalte : Merkmalswerte der Untersuchungsvariablen; vergleichbar also einer  $N * 2$  - Datenmatrix) und wenden uns nun den entsprechenden BASIC-Programmen zu.

## 6.3 Programme

Die obigen Tabellenbeispiele zeigen sehr deutlich, daß die Programmierung einer Häufigkeitstabelle zunächst voraussetzt, daß man den kleinsten und den größten Wert eines Datenbestandes kennt (etwa wenn man klassifizieren will; Beispiel Körpergrösse), und daß man den Datenbestand der Größe nach sortiert vorliegen hat.

In einem ersten Programmbeispiel sollen diese Aufgaben bewältigt werden.

### 6.3.1 Sortieren

#### o Vorstellung des Problems

Ein gegebener Datenbestand soll der Größe der Werte nach sortiert werden.

#### o Problemanalyse

Das Aufsuchen des größten und des kleinsten Wertes kann folgendermaßen vonstattengehen :

Wir geben zwei unrealistische Werte vor : Als größten Wert XG eine sehr kleine Zahl (z.B.-1000), als kleinsten Wert XK eine sehr große Zahl (z.B. 1000). Dann vergleichen wir schrittweise jeden Wert des Datenbestands mit XK und XG.

Wenn sich bei einem solchen Vergleich ergibt, daß

der Zahlenwert aus dem Datenbestand größer als XG ist, dann ersetzt er den bisher bestehenden XG-Wert; ist er hingegen kleiner als XK, so ersetzt er diesen; danach wird der nächste Wert des Datenbestands zum Vergleich herangeholt.

Ist der Wert des Datenbestands weder größer als XG noch kleiner als XK, so erfolgt keine Ersetzung und es wird sofort der nächste Wert des Datenbestands herangeholt.

Durch die Vorgabe der unrealistischen Werte zu Beginn des Vergleichsprozesses wird erreicht, daß schon der erste Wert des Datenbestands als vorläufiger (und vielleicht entgeltiger) XG- bzw. XK-Wert geführt wird und dies gegebenenfalls bleibt.

Sind alle Daten in diesen Vergleichsprozeß einbezogen worden, so finden wir im Feld XK tatsächlich den kleinsten, im Feld XG den größten Wert des Datenbestands.

"Sortieren" bedeutet, daß man je zwei Werte eines Datenbestands miteinander vergleichen muß :

Man vergleicht den ersten Wert mit allen übrigen Schritt für Schritt. Wenn sich bei einem solchen Vergleich ergibt, daß der jeweils erste Wert größer als der jeweils zweite ist, dann müssen beide vertauscht werden.

Dadurch wird erreicht, daß dann, wenn der Inhalt des ersten Feldes mit allen anderen Feldinhalten verglichen wurde, im ersten Feld die insgesamt kleinste Zahl steht.

Vergleicht man dann den jetzt im zweiten Feld stehenden Wert mit allen anderen (außer mit dem im ersten Feld stehenden Wert), so erhält man nach Ablauf aller Vergleiche in dieser Runde im zweiten Feld den insgesamt zweitkleinsten Wert usw.

Haben wir beispielsweise insgesamt 5 Werte zu sortieren, so müssen vier derartiger Vergleichsrunden durchlaufen werden : In der ersten Runde gibt es

vier Vergleiche, in der zweiten drei, in der dritten noch zwei und in der vierten Runde noch ein Vergleich.

Es ist dabei zu beachten, daß nach den ersten Vergleichen die Inhalte der einzelnen Felder immer andere sein können, je nachdem, ob und an welchen Stellen getauscht werden mußte.

#### o Programm

```
10 REM S61-SORTIEREN
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(9)"#BEISPIEL S61-SORTIEREN#"
70 PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BESTIMMT DEN KLEINSTEN"
90 PRINT"UND DEN GROESSTEN WERT EINES DATENBE-"
100 PRINT"STANDS UND SORTIERT IHN DER GROESSE NACH"
130 PRINT TAB(2)"ALS SPIELDATEN WERDEN 15 ANGABEN ZU "
140 PRINT TAB(2)"KOEERPERGROESSEN IN DATA 900 FF. VOR-"
150 PRINTTAB(15)"GEBEBEN."
160 GOSUB 1000:REM WARTEN
170 N=15:DIM X(N)
180 FOR I=1 TO N:READ X(I):NEXT I
190 PRINT:PRINT"#AUSGABE UNSORTIERT# : ":PRINT
200 FOR I=1 TO N:PRINTI,X(I):NEXT I
210 GOSUB 1000:REM WARTEN
220 PRINT"#GROESSTER UND KLEINSTER WERT# :":PRINT
230 XK=1000:XG=0
240 FOR I=1 TO N
250 IF X(I)<XK THEN XK=X(I)
260 IF X(I)>XG THEN XG=X(I)
270 NEXT I
280 PRINT"KLEINSTER WERT : ";XK
290 PRINT"GROESSTER WERT : ";XG
310 GOSUB 1000:REM WARTEN
```

```

320 REM SORTIEREN
330 PRINT"■SORTIERTER DATENBESTAND■ : ":PRINT
340 FOR I=1 TO N-1
350 FOR J=I+1 TO N
360 IF X(I)<X(J) THEN 380
370 H=X(I):X(I)=X(J):X(J)=H
380 NEXT J
390 NEXT I
400 FOR I=1 TO N:PRINTI,X(I):NEXT I
410 PRINT:PRINTTAB(5)"ENDE DER AUSGABE":END
900 DATA 175,178,192,185,188,165,168
910 DATA 173,190,188,162,188,195,172,180
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
H	=	Hilfsfeld beim Tausch
I	=	Laufindex (Werte)
J	=	zweiter Laufindex (bei Vergleichen)
N	=	Zahl der Werte
X()	=	Werte
XG	=	Größter Wert
XK	=	Kleinster Wert

o Programmbeschreibung

Satz 10-160 :

Überschrift und Erläuterungen

Satz 170 :

Vorgabe von N (Zahl der Werte) und Dimensionierung; hier muß geändert werden, wenn mit einer anderen Anzahl von Daten gearbeitet werden soll (unser Spieldatenbestand besteht aus nur 15 Werten)

Satz 180-210 :

Einlesen der Werte und unsortierte Kontrollausgabe

Satz 220-230 :

Vorbereitung der Ermittlung des größten und des kleinsten Wertes durch Ausgabe einer Überschrift und Belegung von XG und XK mit unrealistischen Werten

Satz 240-270 :

Vergleichsschleife : Wenn ein Wert kleiner ist als der bisher kleinste Wert im Feld XK, so ersetzt er diesen in diesem Feld (Satz 250); ist er größer als der bisher größte, ersetzt er diesen im Feld XG (Satz 260)

Satz 280-310 :

Ausgabe des kleinsten und des größten Wertes

Satz 320-330 :

Vorbereitung des Sortierens

Satz 340 :

Beginn der I-Schleife (jeweils erster Wert); diese Schleife endet bei N-1, weil danach ja kein Vergleich mehr möglich ist

Satz 350 :

Beginn der J-Schleife (jeweils zweiter Wert); diese Schleife beginnt immer ein Feld neben dem jeweiligen I-Feld, also bei I+1

Satz 360 :

Vergleich : ergibt sich dabei, daß  $X(I) < X(J)$ , so braucht nicht getauscht zu werden, weshalb das Programm mit Satz 380 (Übergang zum nächsten Wert) fortgesetzt werden kann; muß hingegen getauscht werden, so folgt Satz 370

Satz 370 :

Tausch : Zu diesem Zweck muß ein Hilfsfeld H eingerichtet werden, weil sonst einer der beiden Werte beim Tauschen verlorengehen würde

Satz 380-390 :

Beendigung der Schleifen

Satz 400-410 :

Ausgabe des sortierten Datenbestandes und Beendigung des Hauptprogramms

Satz 900-910 :

Spieldatenbestand (15 Körpergrößen)

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Programmergebnisse

Dieses Programm liefert zunächst den unsortierten Datenbestand und danach :

KLEINSTER WERT	:	162
GROESSTER WERT	:	195

Danach wird der Datenbestand der Größe nach sortiert ausgegeben.

## o Ergänzungen

Es wurde schon darauf hingewiesen, daß der Satz 170 (und natürlich der Datenbestand) geändert werden muß, wenn eine andere Anzahl von Daten sortiert werden soll.

Zu beachten ist weiterhin, daß die Wertvorgabe für XK und XG im Satz 230 verändert werden muß, wenn es echte Werte geben sollte, die diese Grenzen überschreiten. Anders ausgedrückt : die unrealistischen Vorgaben müssen außerhalb des tatsächlichen Datenspektrums liegen.

Schließlich ist anzumerken, daß dieses Programm auch zum alphabetischen Sortieren von Strings verwendet werden kann. Allerdings müssen dann alle benutzten Variablen als Stringvariablen definiert werden.

### 6.3.2 Diskrete Häufigkeitstabelle

#### o Vorstellung des Problems

Für eine diskrete Variable soll per Programm die Häufigkeitstabelle nach dem Muster aus Abschnitt 6.2.1 erstellt werden. Als Beispiel wählen wir die Variable "Kinderzahl" und geben 30 Werte dieser Variablen als Spieldaten über DATA-Anweisungen vor.

## o Problemanalyse

Um die gewünschte Häufigkeitstabelle erzeugen zu können, benötigt man zuerst den kleinsten und den größten Merkmalswert (XK und XG), die wir beide so bestimmen können, wie es in dem vorangegangenen Programm schon durchgeführt wurde.

Danach müssen alle erhobenen Angaben daraufhin untersucht werden, ob sie mit einem Wert zwischen XK und XK (unter Einschluß dieser beiden) übereinstimmen. Ist dies der Fall, wird ein Häufigkeitszähler um 1 erhöht (von denen es XG-XK+1 gibt) und es kann der nächste Wert untersucht werden. Ist dies hingegen nicht der Fall, so muß diese Kinderzahl mit dem nächsten Wert verglichen werden usw.

## o Programm

```
10 REM S62-DISKRETE HV
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(14)"#BEISPIEL S62"
70 PRINT:PRINT TAB(4)"DISKRETE
   HAEUFIGKEITSVERTEILUNG":PRINT
90 PRINT" DIESES PROGRAMM ERZEUGT DIE HAEUFIG-"
100 PRINT"KEITSVERTEILUNG EINER DISKREten VARIAB-"
110 PRINTTAB(17)"LEN.":PRINT
120 PRINT"ALS SPIELDATEN HABEN WIR IN DATA 900 FF"
130 PRINT"30 ANGABEN ZUR KINDERZAHL VON BEFRAGTEN"
140 PRINTTAB(9)"FAMILIEN ANGEGEBEN."
150 GOSUB 1000:REM WARTEN
160 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
```

```

170 N=30
180 DIM X(N)
190 FOR I=1TON:READX(I):NEXTI
200 XG=0: XK=1000
210 FOR I=1TON
220 IF X(I)<XK THEN XK=X(I)
230 IF X(I)>XG THEN XG=X(I)
240 NEXT I
250 K=XG-XK+1: DIM H(K),R(K)
260 FOR I=1 TO N:FOR J=1 TO K
270 IF X(I)=J-1 THEN H(J)=H(J)+1:GOTO 290
280 NEXT J
290 NEXT I
300 FOR J=1 TO K:R(J)=(H(J)/N)*100
310 R(J)=INT(R(J)*100+.5)/100:NEXT J
320 PRINTCHR$(147)
330 PRINT"■KINDER";TAB(12)"HAEUFIGKEIT"
340 PRINT TAB(12)"ABS. IN %";PRINT
350 FOR I=1 TO 25:PRINT"-";:NEXT I:PRINT
360 FOR J=1 TO K
370 PRINTTAB(2)J-1;TAB(12)H(J),R(J)
380 NEXT J
390 PRINT:PRINT:PRINT
400 PRINTTAB(5)"ENDE DER AUSGABE":END
900 DATA 3,4,2,6,0,0,4,2,5,1
910 DATA 2,0,0,0,1,1,0,0,3,1
920 DATA 0,1,2,3,2,1,1,1,0,1
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!";PRINT
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A\$ = Tastatureingabe  
H() = absolute Häufigkeiten  
I = Laufindex (Werte)  
J = Laufindex (Klassen)  
K = Klassenanzahl  
N = Anzahl der Werte  
R() = relative Häufigkeiten  
X() = Werte  
XG = Größter Wert  
XK = Kleinster Wert

o Programmbeschreibung

Satz 10-160 :

Überschrift und Erläuterungen

Satz 170-190 :

Vorgabe der Zahl der Werte (170), Dimensionieren (180) und Einlesen der Spieldaten (190)

Satz 200-240 :

Bestimmung des kleinsten Wertes XK und des größten Wertes XG

Satz 250 :

Bestimmung der Klassenanzahl K und Dimensionierung der absoluten und der relativen Häufigkeiten

Satz 260-290 :

Bestimmung der absoluten Häufigkeiten :

Für jeden Wert  $X(I)$  wird geprüft, ob er mit  $J-1$  ( $= 0, 1, 2, \dots$ ) übereinstimmt (270); ist dies der Fall, wird der Zähler  $H(J)$  (Häufigkeit) um 1 erhöht und der nächste Wert  $X(I)$  in Angriff genommen (Sprung von 270 nach 290); solange hingegen der Vergleich negativ ausfällt, wird  $J$  um 1 erhöht (280)

Satz 300-310 :

Bestimmung der relativen Häufigkeiten  $R(J)$  (Dreisatzrechnung) (300) und Runden dieser Werte auf zwei Dezimalstellen (310)

Satz 320-380 :

Löschen des Bildschirms und Ausgabe der Häufigkeitstabelle

Satz 390-400 :

Beendigung des Hauptprogramms

Satz 900-920 :

Spieldatenbestand

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Programmergebnisse

Nach dem Start dieses Programms erfolgt nach der Ausgabe von Überschrift und Erläuterungen zunächst die Meldung :

MOMENT BITTE, ICH RECHNE

Danach wird auf dem geräumten Bildschirm die Häufigkeitstabelle in folgender Form ausgegeben :

KINDER	HAEUFIGKEIT	
	ABS.	IN %
0	9	30.00
1	9	30.00
2	5	16.67
3	3	10.00
4	2	6.67
5	1	3.33
6	1	3.33

ENDE DER AUSGABE

o Ergänzungen

Wir sehen sofort, wie das Programm geändert werden muß, wenn ein anderer Datenbestand verwendet werden soll :

Zum einen sind natürlich die DATA-Anweisungen in Satz 900 ff. zu ändern, zum anderen muß N in Satz 170 anders belegt werden.

Desweiteren könnte es sinnvoll sein, die relativen Häufigkeiten nicht auf zwei Dezimalstellen, sondern anders zu runden.

Die hier verwendete Rundung kommt ja so zustande, daß jede relative Häufigkeit zunächst mit 100 multipliziert wird :So wird zum Beispiel aus 16.66666 der Wert 1666.666, aus 3.33333 wird entsprechend 333.333;

danach wird jeweils 0.5 dazu addiert, d.h. wir erhalten für den Fall der eben genannten beiden Werte die Zahlen 1667.166 und 333.833;

danach wird die INT - Funktion angewandt, d.h. wir erhalten 1667 und 333;

wird nun schließlich wieder durch 100 dividiert, erhalten wir 16.67 und 3.33.

Will man nun beispielsweise auf drei Dezimalstellen runden, so braucht die Zahl 100 in dem obigen Algorithmus nur durch die Zahl 1000 ersetzt zu werden.

Ein abschließender Hinweis muß sich der Frage zuwenden, wie vorzugehen ist, wenn die Untersuchungsvariable zwar diskret, aber nicht ganzzahlig ist, also nicht wie beim Beispiel "Kinderzahl" die Werte 0, 1, 2,... annimmt, sondern z.B. die Werte 3.5, 4, 4.5, 5, ...

Dann muß in der Vergleichsschleife (J-Schleife; Satz 260, 2.Anweisung bis Satz 280) entsprechend reagiert werden :

Diese Schleife hat als allgemeinen Anfangswert den Wert XK, als Endwert den Wert XG und es muß die jeweilige Schrittweite (im Alternativbeispiel also 0.5) angegeben werden. Der Laufindex J dieser Schleife kann dann nicht gleichzeitig der Zählindex für die Klassen sein, sondern es muß eine gesonderte Zählgröße "mitlaufen".

Im folgenden Programmausschnitt, der eine Lösung

für dieses Problem darstellt, haben wir diesen Hilfszähler mit Z bezeichnet :

```
260 INPUT "SCHRIITWEITE : ";S
262 FOR I = 1 TO N
264 Z = 0
266 FOR J = XK TO XG STEP S
268 Z = Z+1
270 IF X(I) = J THEN H(Z)=H(Z)+1:GOTO 290
280 NEXT J
290 NEXT I
```

Natürlich muß dann auch bei den folgenden Berechnungen berücksichtigt werden, daß es mehr Klassen gibt als ganzzahlige Werte zwischen XK und XG.

### 6.3.3 Stetige Häufigkeitstabelle

#### o Vorstellung des Problems

Es stellt sich hier die gleiche Aufgabe mit dem Unterschied, daß die Untersuchungsvariable jetzt vom stetigen Typ ist. Als Beispiel gehen wir von einem Spieldatenbestand von 30 Körpergrößen aus.

#### o Problemanalyse

Bei der Erstellung dieses Programms stehen wir vor den gleichen Problemen wie im vorangegangenen Abschnitt, so daß wir auf die dortige Problemanalyse verweisen können.

Zusätzlich ist hier nun zu berücksichtigen, daß Klassen gebildet werden müssen ("von ... bis unter ..."). Dieses Problem können wir so lösen, daß wir den kleinsten und den größten Merkmalswert des Datenbestands auf dem Bildschirm ausgegeben lassen, so daß dann der Benutzer entscheiden kann, mit welcher Klassenbreite bei der Fortsetzung des Programms gearbeitet werden soll.

Zusätzlich lassen wir den Benutzer darüber entscheiden, bei welchem Wert die Klassifizierung beginnen soll. Häufig wünscht man ja einen "glatten" Wert, der nicht notwendigerweise mit dem kleinsten Wert des Datenbestands XK identisch sein muß.

Bei der "Einsortierung" der Merkmalswerte in die einzelnen Klassen, also bei der Bestimmung der Häufigkeiten der einzelnen Klassen, wird nun nicht - wie oben - jeder Merkmalswert (Laufindex I) mit jedem Klassenwert (Laufindex J) verglichen, sondern daraufhin geprüft, ob er größer oder gleich dem jeweiligen Klassenanfangspunkt und zugleich kleiner ("bis unter") als der jeweilige Klassenendpunkt ist.

Ist dies der Fall, wird die Häufigkeit dieser Klasse um 1 erhöht und der nächste Merkmalswert betrachtet; ist dies hingegen nicht der Fall, wird die nächste Klasse (für den gleichen Merkmalswert) betrachtet.

o Programm

```
10 REM S63-STETIGE HV
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"#BEISPIEL S63"
70 PRINT:PRINT TAB(4)"STETIGE HAEUFIGKEITSVERTEILUNG"
80 PRINT:PRINT
90 PRINT" DIESES PROGRAMM ERZEUGT DIE HAEUFIG-"
100 PRINT"KEITSVERTEILUNG EINER STETIGEN VARIAB-"
110 PRINTTAB(17)"LEN.":PRINT
120 PRINT"ALS SPIELDATEN HABEN WIR IN DATA 900 FF"
130 PRINT"30 ANGABEN ZUR KOERPERGROESSE VON ZU-"
140 PRINT"FAELLIG AUSGEWAELHTEN PERSONEN ANGEBEBEN"
150 GOSUB 1000:REM WARTEN
160 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
170 N=30
180 DIM X(N)
190 FORI=1TON:READX(I):NEXTI
200 XG=0:XK=1000
210 FOR I=1TON
220 IF X(I)<XK THEN XK=X(I)
230 IF X(I)>XG THEN XG=X(I)
240 NEXT I
250 PRINTCHR$(147)
260 PRINT"KLEINSTER WERT : ";XK
270 PRINT"GROESSTER WERT : ";XG
280 PRINT:PRINT:PRINT
290 INPUT "GEWUENSCHTE KLASSENBREITE (RETURN)";B
300 IF B>(XG-XK)/2 THEN PRINT"ZU GROSSER WERT":GOTO 280
305 IF B<(XG-XK)/20 THEN PRINT"ZU KLEINER WERT":GOTO 280
310 PRINT:PRINT:PRINT
```

```

320 INPUT "START DER KLASSENBIILDUNG (RETURN)";S
330 IF S>XK THEN PRINT"ZU GROSSER WERT":GOTO 310
335 IF S<XK-B THEN PRINT"ZU KLEINER WERT":GOTO 310
340 K=(INT(XG-S)/5)+1:DIM H(K),R(K)
350 PRINTCHR$(147):PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
360 FOR I=1 TO N
370 J=1
380 FOR L=S TO XG+B STEP B
390 IF X(I)<L+B AND X(I)>=L THEN H(J)=H(J)+1:GOTO 420
400 J=J+1
410 NEXT L
420 NEXT I
430 FOR J=1TOK:R(J)=(H(J)/N)*100
440 R(J)=INT(R(J)*100+.5)/100
450 NEXT J
460 PRINTCHR$(147)
470 PRINTTAB(3)"GRÖSSE";TAB(16)"HÄUFIGKEIT"
480 PRINT TAB(16)"ABS. IN %":PRINT
490 FOR I=1 TO 30:PRINT"-";:NEXT I:PRINT
500 FOR J=1 TO K
510 PRINTS+(J-1)*B;" BIS ";S+J*B;TAB(16)H(J);TAB(23)R(J)
520 NEXT J
530 PRINT:PRINT:PRINT
540 PRINTTAB(7)"ENDE DER AUSGABE":END
900 DATA 175,162,188,182,169,165,188,181,192,175
910 DATA 173,174,177,187,185,181,182,169,170,190
920 DATA 170,180,190,185,175,177,183,186,166,170
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"BITTE DIE LEERTASTE DRUECKEN!"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
B	=	Klassenbreite
H()	=	absolute Häufigkeiten
I	=	Laufindex (Werte)
J	=	Laufindex (Klassen)
K	=	Klassenanzahl
L	=	Klassengrenzenindex
N	=	Anzahl der Werte
R()	=	relative Häufigkeiten
S	=	Klassifizierungsanfang
X()	=	Werte
XG	=	Größter Wert
XK	=	Kleinster Wert

o Programmbeschreibung

Satz 10-160 :

Überschrift und Erläuterungen

Satz 170-190 :

Vorgabe der Anzahl der Werte im Datenbestand, Dimensionierungen und Einlesen aller Werte

Satz 200-240 :

Bestimmung des kleinsten und des größten Wertes XK und XG

Satz 250-280 :

Ausgabe von XK und XG

Satz 290-305 :

Anforderung der gewünschten Klassenbreite B;

wenn diese nicht sinnvoll ist, nämlich so groß, daß höchstens zwei Klassen gebildet werden können, oder so klein, daß zu viele Klassen gebildet werden müssen, erfolgt eine entsprechende Meldung und ein Rücksprung zum Satz 280 (erneute Anforderung von B)

Satz 310-335 :

Anforderung des Startpunkts der Klassifizierung, der nicht größer sein darf als XK und nicht kleiner sein sollte als XK-B (bei falschen Eingaben erfolgt eine Meldung und S wird erneut angefordert)

Satz 340 :

Bestimmung der Klassenanzahl K aus der Spannweite XG-S und der Klassenbreite B und Dimensionierung der absoluten und der relativen Häufigkeiten

Satz 350 :

Zwischenmeldung

Satz 360-420 :

Einsortieren der Merkmalswerte in die Klassen :

Der Klassengrenzenindex L läuft dabei vom Startpunkt S bis zum Endpunkt "hinter" XG mit der

Schrittweite B (Klassenbreite) (Satz 390) und es wird geprüft, ob der Merkmalswert Nr. I in die jeweilige Klasse hineinpaßt; wenn ja, wird die Häufigkeit um 1 erhöht und es geht weiter bei Satz 420 (nächster Wert); wenn nein, wird die nächste Klasse probiert (Satz 400 und 410)

Satz 430-450 :

Bestimmung der relativen Häufigkeiten und Runden auf zwei Dezimalstellen

Satz 460-540 :

Ausgabe der Häufigkeitstabelle und Beendigung des Hauptprogramms

Satz 900-920 :

Spieldaten

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wurde

o Programmergebnisse

Bei dem gegebenen Spieldatenbestand liefert das Programm nach Überschrift und Erläuterungen zunächst die folgenden Informationen :

KLEINSTER WERT : 162  
GROESSTER WERT : 192

und fordert dann die Klassenbreite und den Startpunkt der Klassifizierung an.

Aufgrund der obigen Informationen kann der Benutzer nun beispielsweise als Klassenbreite den Wert 5 und als Startpunkt den Wert 160 vorgeben.

Es wird dann eine Häufigkeitstabelle mit folgenden Klassen gebildet :

160 BIS UNTER 165

165 BIS UNTER 170

usw. bis

190 BIS UNTER 195

Die sich ergebende Tabelle entspricht dem Muster, wie wir es in Abschnitt 6.2.2 kennengelernt haben.

o Ergänzungen

Es sind hier die gleichen Ergänzungen sinnvoll, wie sie im vorangegangenen Abschnitt vorgetragen wurden.

#### 6.3.4 Bivariate Tabelle (Kreuztabelle)

o Vorstellung des Problems

Es soll eine Häufigkeitstabelle erstellt werden, bei der die Merkmalsträger nach zwei Merkmalen gleichzeitig ausgezählt werden. Wir gehen dabei

wieder von einem Spieldatenbestand aus :

30 Personen werden nach Geschlecht und nach der bevorzugten politischen Partei befragt; es soll eine Tabelle nach dem Muster der in Abschnitt 6.2.3 vorgestellten erstellt werden.

o Problemanalyse

Die Erstellung einer Häufigkeitsverteilung, bei der die Merkmalsträger nach zwei Merkmalen gleichzeitig ausgezählt werden, bietet keine neuen Probleme. Zu den Arbeitsschritten, die schon in den vorangegangenen beiden Beispielen durchlaufen wurden, tritt nur ein weiterer hinzu :

Es muß für jeden Merkmalswert überprüft werden, ob er mit je zwei Klassenwerten übereinstimmt (diskreter Fall), bzw. in je zwei Klassen hineinpaßt (stetiger Fall). Dies kann durch entsprechende Schleifenschachtelung erreicht werden.

Dem entspricht es, daß nun die Häufigkeiten in einem doppelt indizierten Feld zu speichern sind, denn es entsteht ja eine Häufigkeitstabelle mit  $V_1$  Zeilen ( $V_1$  = Anzahl der Werte der ersten Variablen) und  $V_2$  Spalten ( $V_2$  = Anzahl der Werte der zweiten Variablen).

In unserem Beispiel ist  $V_1 = 5$  (CDU, SPD, FDP, Grüne, Sonstige) und  $V_2 = 2$  (männlich, weiblich).

o . Programm

```
10 REM S64--KREUZTABELLE
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"#BEISPIEL S64"
70 PRINT:PRINT TAB(3)"BIVARIATE HAEUFIGKEITSVERTEILUNG#"
80 PRINT
90 PRINT" DIESES PROGRAMM ERZEUGT DIE HAEUFIG-"
100 PRINT"KEITSVERTEILUNG FUER ZWEI DISKRETE VA-"
110 PRINTTAB(15)"RIABLEN.":GOSUB 1000
120 PRINT"ALS SPIELDATEN HABEN WIR IN DATA 900 FF"
130 PRINT"ANGABEN ZUM GESCHLECHT UND ZUR BEVOR-"
140 PRINT"ZUGTEN POLITISCHEN PARTEI VON 30 ZU-"
150 PRINT"FAELLIG BEFRAGTEN PERSONEN VORGEGBEN."
160 PRINT
180 PRINT"ES IST FOLGENDERMASSEN CODIERT WORDEN.":PRINT
190 PRINT"GESCHLECHT : ":PRINT
200 PRINT"    MAENNlich = 1"
210 PRINT"    WEIBlich  = 2":PRINT
220 PRINT:PRINT"BEVORZUGTE PARTEI : ":PRINT
230 PRINT"    CDU      = 1"
240 PRINT"    SPD      = 2"
250 PRINT"    FDP      = 3"
260 PRINT"    GRUENE   = 4"
270 PRINT"    SONSTIGE  = 5"
280 GOSUB 1000:REM WARTEN
```

```

290 N=30
300 DIM X(N),Y(N),H(5,2),P*(5)
310 PRINTTAB(7)"MOMENT BITTE, ICH LESE"
320 FOR I=1 TO N:READ X(I),Y(I):NEXT I
330 FORP=1TO5:READP*(P):NEXTP
340 PRINTCHR$(147):PRINTTAB(9)"▣KONTROLLAUSGABE▣":PRINT
350 PRINT"FALL", "GESCHLECHT", "PARTEI":PRINT
360 FOR I=1 TO N
370 PRINTTAB(1)I;TAB(14)X(I);TAB(32)Y(I)
380 IF I/10=INT(I/10)THEN GOSUB 1000
390 NEXT I
400 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
410 FOR I=1 TO N
420 FOR P=1 TO 5
430 FOR G=1 TO 2
440 IF X(I)=G AND Y(I)=P THEN H(P,G)=H(P,G)+1:GOTO 470
450 NEXT G
460 NEXT P
470 NEXT I
480 PRINTCHR$(147):PRINTTAB(10)"▣KREUZTABELLE▣":PRINT:PRINT
490 PRINTTAB(7)"MAENNLICH";" WEIBLICH":PRINT
495 FOR I=1 TO 26:PRINT"*";:NEXT I:PRINT:PRINT
500 FOR P=1 TO 5
510 PRINTP*(P),H(P,1),H(P,2)
520 NEXT P
530 PRINT:PRINT:PRINT:PRINT:PRINT
540 PRINTTAB(8)"ENDE DER AUSGABE":END
900 DATA 1,1,2,1,1,3,2,4,1,5,1,2,2,2,2,2,1,1,1,2
910 DATA 1,3,2,5,2,4,1,4,2,3,2,2,1,2,1,1,2,1,1,2
920 DATA 1,1,2,1,2,2,1,3,2,3,2,2,1,2,1,5,2,4,1,4
930 DATA CDU,SPD,FDP,GRUENE,SONSTIGE
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"▣BITTE DIE LEERTASTE DRUECKEN!▣"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

- A\$ = Tastatureingabe  
G = Laufindex (Geschlecht)  
H(,) = absolute Häufigkeiten  
I = Laufindex (Fälle)  
N = Anzahl der Fälle  
P = Laufindex (Parteien)  
P\$( ) = Parteinamen  
X( ) = Werte (Geschlecht)  
Y( ) = Werte (bevorzugte Partei)

o Programmbeschreibung

Satz 10-280 :

Überschrift und Erläuterungen, Hinweise zur Codierung der beiden Variablen

Satz 290 :

Vorgabe der Anzahl der Merkmalsträger (Fälle)

Satz 300 :

Dimensionierung der beiden Untersuchungsvariablen X und Y, der Häufigkeiten H und der Parteina-  
men P\$

Satz 310-330 :

Einlesen aller Werte (in den DATA - Anweisungen sind die Werte paarweise angegeben, also Geschlecht, Partei, Geschlecht, Partei ... usw.)

Satz 340-390 :

Kontrollausgabe der Datenmatrix mit Unterbrechung nach je 10 Fällen

Satz 400-470 :

Einsortieren der Merkmalsträger in die Merkmalsklassen, wobei nun drei Schleifen benötigt werden:

1. Schleife über alle Fälle (Laufindex I)
2. Schleife über die Werte der ersten Variablen (Laufindex P) = Zeilen der Kreuztabelle
3. Schleife über die Werte der zweiten Variablen (Index G) = Spalten der Kreuztabelle

Zu prüfen ist dabei, ob für den Fall Nr. I der Wert der ersten Variablen  $X(I) = G$  und zugleich der Wert der zweiten Variablen  $Y(I) = P$ ; wenn ja, ist die Häufigkeit  $H(P,G)$  um 1 zu erhöhen und der nächste Fall zu untersuchen; wenn nein, ist mit weiteren P,G-Kombinationen zu prüfen

Satz 480-540 :

Ausgabe der entstandenen Kreuztabelle und Beendigung des Hauptprogramms

Satz 900-920 :

Spieldatenbestand (paarweise : X,Y, X,Y, ... usw.)

Satz 930 :

Parteinamen zum Einlesen

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

o Programmergebnisse

Von unserem Spieldatenbestand ausgehend, erzeugt  
dieses Programm die folgende Kreuztabelle :

KREUZTABELLE

	MAENNLICH	WEIBLICH
CDU	4	3
SPD	5	5
FDP	3	2
GRUENE	2	3
SONSTIGE	2	1

ENDE DER AUSGABE

o Ergänzungen

Es ist leicht zu sehen, wie dieses Programm ver-  
allgemeinert werden kann, so daß es für beliebige  
bivariate Datenbestände verwendet werden kann:

Hat man einen anderen Datenbestand, so müssen die  
DATA-Anweisungen (900 ff.) und N in Satz 290 ver-  
ändert werden; denkbar wäre in diesem Zusammen-

hang auch, daß man nicht nur die Bezeichnungen für die eine Variable per DATA-Anweisung vorgibt (Satz 930), sondern auch die für die andere.

Weiterhin ist daran zu erinnern, daß bei der Betrachtung einer (oder zwei) stetigen Variablen zunächst Klassen zu bilden sind (vergl. Abschnitt 6.3.3).

Zu Zwecken der Verallgemeinerung dieses Programms wäre es auch angebracht, die Zahl der Ausprägungen der beiden Variablen nicht - wie in obigem Beispiel mit 5 und 2 - vorzugeben, sondern allgemein als Variablen V1 und V2 über INPUT-Anweisungen zu belegen.

Dabei ist allerdings zu beachten, daß der Bildschirm zur Darstellung der Kreuztabelle nicht ausreicht, wenn V1 oder V2 zu groß wird. Dann muß gegebenenfalls die Kreuztabelle in passende Teile zerlegt werden, um diese dann Schritt für Schritt auszugeben.

## Kapitel 7 : Graphische Darstellungen von Häufigkeitsverteilungen

### 7.1 Statistische Grundlagen

Wir haben im vorangegangenen Kapitel gezeigt, wie man Häufigkeitsverteilungen tabellarisch darstellen kann. Bei umfangreicheren Datenbeständen ist häufig aber eine graphische Darstellung übersichtlicher und deshalb informativer. Vor allem bei Benutzung von unterschiedlichen Schraffuren, Farben und Strichstärken wird der Informationsgehalt von Daten leichter zugänglich.

Deshalb sollen hier nun für uni- und bivariate Häufigkeitsverteilungen Programme für solche graphische Darstellungen vorgestellt werden, wie sie in der Praxis am häufigsten benutzt werden.

Dabei unterteilen wir dieses Kapitel genauso wie das vorhergehende :

- Betrachtung einer diskreten Variablen
- Betrachtung einer stetigen Variablen
- Betrachtung von zwei diskreten Variablen gleichzeitig

Die multivariate Betrachtung (drei Variablen oder mehr gleichzeitig) schließen wir hier aus, weil entsprechende graphische Darstellungen nicht mehr sonderlich übersichtlich sind.

Allerdings muß hier nun ein weiteres Kriterium der Unterscheidung ins Spiel gebracht werden, nämlich die sog. Skalenqualität der in Frage stehenden Untersuchungsvariablen, weil diese die Art der möglichen graphischen Darstellung maßgeblich beeinflußt.

Mit dem Stichwort "Skalenqualität" (häufig auch "Skalenniveau" oder "Datenqualität" genannt) bezeichnet man den Gehalt an Informationen statistischer Daten.

Betrachten wir zur Klärung dieses Begriffs drei einfache Beispiele, die die am häufigsten auftretenden Qualitätsstufen anschaulich illustrieren :

Untersuchungsvariable sei "bevorzugte politische Partei" : Zwei befragte Personen können sich bezüglich der Ausprägungen dieser Variablen voneinander unterscheiden oder nicht, d.h. eine solche Variable beinhaltet die sog. Unterscheidungsinformation. Wir sagen, die Werte dieser Variablen sind nominalskaliert .

In einem zweiten Beispiel betrachten wir die Variable "schulische Leistungen". Diese Variable beinhaltet ebenfalls die Unterscheidungsinformation - zusätzlich können wir aber mit den Werten dieser Variablen eine Rangordnung bilden, d.h. es sind auch Ranginformationen vorhanden : Wir sprechen von einer ordinalskalierten Variablen.

Als letztes Beispiel sei die Variable genannt, mit der wir es auch schon zu tun hatten, die Größe befragter Personen. Diese Variable beinhaltet die Unterscheidungs- und die Ranginformation, läßt aber zusätzlich noch die inhaltliche Interpretation der Abstände (Intervalle) zwischen je zwei Werten zu. Wir nennen deshalb die Werte dieser Variablen intervallskaliert .

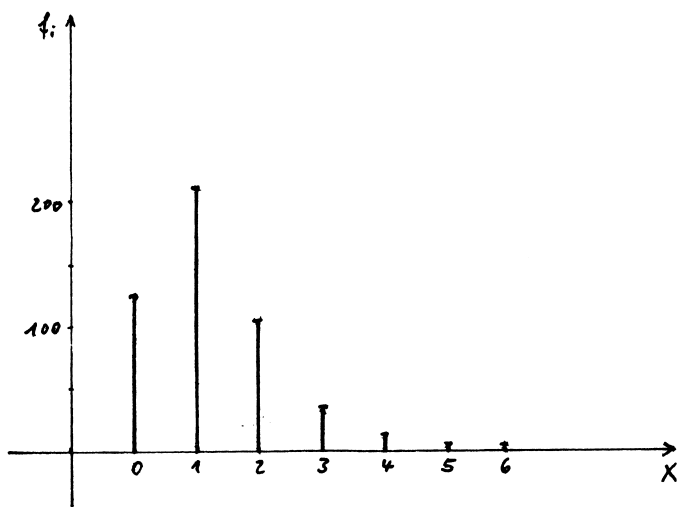
Wir werden im folgenden an einfachen Beispielen erkennen, daß diese Unterscheidungen die Möglichkeiten graphischer Darstellungen mitbestimmen, wie übrigens auch häufig die Auswahl statistischer Auswertungsverfahren, über die in den dann folgenden Kapiteln gesprochen wird.

### 7.1.1 Eine diskrete Variable

Als typisches Beispiel einer diskreten Variablen haben wir die Variable "Kinderzahl" genannt. Sie weist eine hohe Skalenqualität auf und deshalb ist die geeignete Form der graphischen Darstellung einer Häufigkeitsverteilung - wie übrigens für alle diskreten Variablen, die höhere als Ordinalskalenqualität aufweisen - das sog. Stabdiagramm .

Für das Beispiel aus Abschnitt 6.2.1 ergibt sich schematisch das folgende Bild :

Häufigkeitsverteilung Kinderzahl



Ist die Skalenqualität der Untersuchungsvariablen, die graphisch dargestellt werden soll, geringer als eine Intervallskala, eignet sich das Stabdiagramm nicht.

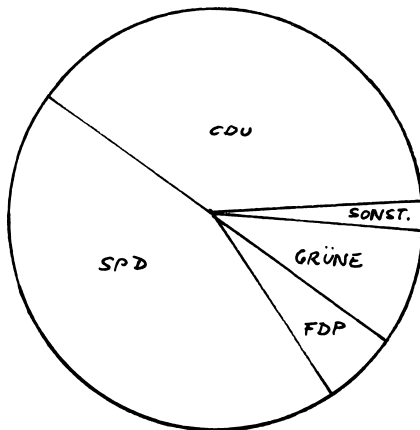
Typisches Beispiel dafür ist die Variable "bevorzugte politische Partei". Wollte man hier ein Stabdiagramm anfertigen, dann müßte man die einzelnen Parteien auf der waagrechten Achse anordnen, was aber letzten Endes völlig willkürlich geschehen könnte.

Deshalb wählt man eine andere Form der graphischen Darstellung, z.B. das Kreisdiagramm, so wie man es aus den Wahlberichterstattungen des Fernsehens kennt.

Hat man beispielsweise die folgenden statistischen Angaben

CDU	39.4 %
SPD	44.6 %
FDP	5.5 %
Grüne	8.1 %
Sonstige	2.4 %

so erhält man ein Bild, das der folgenden Skizze entspricht :



### 7.1.2 Eine stetige Variable

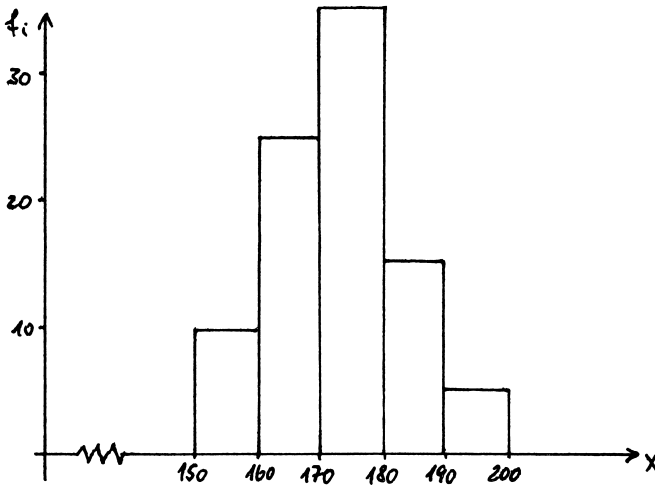
Stetige Variablen weisen in aller Regel eine hohe Skalenqualität auf, so daß wir uns hier auf eine typische Form der Darstellung beschränken können, nämlich auf das sog. Histogramm (Säulendiagramm).

Der Grundgedanke bei dieser Art der Darstellung ist der, daß die Merkmalswertklassen (bei der stetigen Variablen haben wir ja im Gegensatz zur diskreten Variablen Klassen von Merkmalswerten und nicht nur einzelne Punkte) auf der waagrechten Achse eines Achsenkreuzes und die dazugehörigen Häufigkeiten als Rechtecksflächen über diesen Klassen abgetragen werden.

Ein einfaches Beispiel dazu wird durch den folgenden Datenbestand gegeben :

Körpergröße	Häufigkeit
150 bis unter 160	10
160 bis unter 170	25
170 bis unter 180	35
180 bis unter 190	15
190 bis unter 200	5

Diese Häufigkeitsverteilung führt zu dem folgenden Histogramm :



### 7.1.3 Zwei Variablen (Streudiagramm)

Betrachtet man zwei Variablen gleichzeitig, so werden graphische Darstellungen üblicherweise nur dann erstellt, wenn keine Klassifizierungen oder Gruppierungen vorliegen, weil man in diesen Fällen dreidimensionale Achsenkreuze bräuchte, die

sehr leicht unübersichtlich werden.

Liegt jedoch noch die ursprüngliche Datenmatrix vor mit  $N$  Zeilen ( $N =$  Anzahl der Fälle) und 2 Spalten (zwei Variablen  $X$  und  $Y$ ), liegen also noch die einzelnen Wertepaare vor, so ergibt sich eine sehr anschauliche Möglichkeit der graphischen Darstellung.

Man benutzt wieder ein Achsenkreuz, trägt auf der waagrechten Achse (Abszisse) die Werte der Variablen  $X$  und auf der senkrechten Achse (Ordinate) die Werte der Variablen  $Y$  ab und kann dann jeden Merkmalsträger als Punkt im Achsenkreuz durch zwei Koordinaten, die seinem  $X$ - und  $Y$ -Wert entsprechen, fixieren.

Üblicherweise wird dabei auf die  $X$ -Achse diejenige Variable gesetzt, von der man weiß (oder glaubt), daß sie die andere ( $Y$ ) beeinflusst. Man bezeichnet dann  $Y$  als die abhängige,  $X$  als die unabhängige Variable.

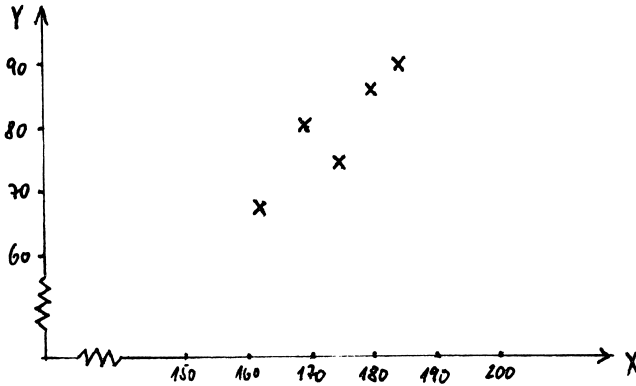
Typisches Beispiel dafür ist die Betrachtung von Körpergröße ( $X$ ) und Gewicht ( $Y$ ) befragter Personen. Die Datenmatrix sieht in diesem Beispiel folgendermaßen aus :

Person	X	Y
1	175	74
2	184	90
3	162	67
4	170	80
5	180	85

usw.

In der graphischen Darstellung, die man Streudiagramm nennt, sieht dies so aus :

### Streudiagramm (Größe/Gewicht)



Natürlich findet man in einer solchen Graphik sehr viel mehr Punkte (eine sog. Punktwolke), wenn man es mit einem echten Datenbestand zu tun hat.

Aus der Lage dieser Punktwolke lassen sich interessante inhaltliche Schlußfolgerungen ziehen :

Steigt sie z.B. von links unten nach rechts oben an, so kann man sagen, daß tendenziell die Werte der Variablen Y zunehmen, wenn auch die der Variablen X zunehmen (je größer, desto tendenziell schwerer). Sinkt sie hingegen von links nach rechts ab, kann man sagen, daß mit steigenden X-Werten die Y-Werte tendenziell fallen (je höher der Zigarettenkonsum, desto geringer tendenziell die Lebenserwartung).

Ist eine Punktwolke sehr eng (schmal), so kann man sagen, daß der Zusammenhang zwischen den beiden Variablen X und Y stark ist; ist die Punktwolke "dick", so kann man von einem nur lockeren Zusammenhang sprechen (oder gar keinem).

## 7.2 Programme

Für die verschiedenen Aufgabenstellungen, wie sie in Abschnitt 7.1 vorgestellt wurden, sollen nun die geeigneten BASIC-Programme entwickelt werden.

### 7.2.1 Stabdiagramm

#### o Vorstellung des Problems

Es soll ein BASIC-Programm entwickelt werden, das die Häufigkeitsverteilung für die Variable Kinderzahl, die in Abschnitt 6.2.1 erörtert wurde, als Stabdiagramm auf dem Bildschirm erzeugt. Die Spieldaten entnehmen wir aus dem erwähnten Beispiel.

#### o Problemanalyse

Bei der Problemanalyse orientieren wir uns zuerst an den Überlegungen, die zur Erstellung der entsprechenden tabellarischen Häufigkeitsverteilung führten. Dort waren die folgenden Arbeitsschritte erforderlich :

1. Eingabe aller Werte
2. Bestimmung des kleinsten und des größten Wertes XK und XG

### 3. Auszählen der Häufigkeiten

Diese Schritte brauchen hier nicht mehr gesondert betrachtet zu werden.

Neu hinzu tritt nun das Problem der Erstellung der Zeichnung auf dem Bildschirm.

Es ist zweckmäßig, bei solchen graphischen Ausgaben immer darauf zu achten, daß der auf dem Bildschirm zur Verfügung stehende Platz gut ausgenutzt wird. Beispielsweise sollten die Merkmalswerte (hier die Kinderzahlen 0 bis 6; allgemein XK bis XG) den waagrechten Platz ausnutzen und die Stäbe des Stabdiagramms den senkrechten Platz.

Dies erreicht man in der sog. Normalgraphik, mit der wir es hier noch zu tun haben, dadurch, daß man die sieben Merkmalswerte (0 bis 6) gleichmäßig auf die 40 zur Verfügung stehenden Bildschirmspalten (von links nach rechts) verteilt und die Stäbe des Stabdiagramms so zeichnet, daß der längste Stab höchstens die 25 zur Verfügung stehenden Bildschirmzeilen ausnutzt.

Deshalb ist es im nächsten Schritt der Problemlösung erforderlich, zunächst den längsten Stab (die größte Häufigkeit) zu bestimmen: Der Problemlösungsweg entspricht demjenigen, den größten Wert eines Datenbestands zu bestimmen, was ja schon bekannt ist (nur geht es jetzt nicht um  $X(I)$  und  $XG$ , sondern um  $H(J)$  und  $HG$ ).

Die übrigen Stäbe (Häufigkeiten) müssen dann mit Hilfe der Dreisatzrechnung unter Nutzung des Maximalwertes  $HG$  normiert werden :

Hat zum Beispiel die größte Häufigkeit  $HG$  den Wert 12, soll sich aber in der Graphik über 24 Bildschirmzeilen erstrecken, und beträgt die erste Häufigkeit z.B. 3, dann muß sich der ihr entsprechen-

de Stab über 6 Bildschirmzeilen erstrecken. Diese Umrechnungen bezeichnen wir als "Normierung".

Wenn die Normierung erfolgt ist, können die Stäbe gezeichnet werden, indem wir die Bildschirmadressen von Stabanfang und Stabende bestimmen und alle Adressen (senkrecht) zwischen diesen beiden Werten (unter Einschluß der beiden) mit kleinen senkrechten Strichen füllen.

Dies ermöglicht die Anweisung

POKE A,66

wobei A die jeweilige Bildschirmadresse und 66 die Codezahl für einen senkrechten Strich darstellen.

o Programm

```
10 REM S71-STABDIAGRAMM
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(14)"BEISPIEL S71"
70 PRINT:PRINT TAB(14)"STABDIAGRAMM"
80 PRINT:PRINT
90 PRINT" DIESES PROGRAMM ERZEUGT DIE HAEUFIG-"
100 PRINT"KEITSVERTEILUNG EINER DISKRETEN VARIAB-"
110 PRINT" LEN IN GRAPHISCHER DARSTELLUNG.":PRINT
120 PRINT"ALS SPIELDATEN HABEN WIR IN DATA 900 FF"
130 PRINT"30 ANGABEN ZUR KINDERZAHL VON BEFRAGTEN"
140 PRINTTAB(11)"FAMILIEN ANGEGEBEN."
150 GOSUB 1000:REM WARTEN
160 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
170 N=30
180 DIM X(N)
190 FORI=1TON:READX(I):NEXTI
200 XG=0:XK=1000
210 FOR I=1TON
220 IF X(I)<XK THEN XK=X(I)
230 IF X(I)>XG THEN XG=X(I)
240 NEXT I
250 K=XG-XK+1:DIM H(K),R(K)
260 FOR I=1 TO N:FOR J=1 TO K
270 IF X(I)=J-1 THEN H(J)=H(J)+1:GOTO 290
280 NEXT J
290 NEXT I
```

```

300 REM BESTIMMUNG DER GROESSTEN HAEUFIGKEIT HG
310 HG=0
320 FOR J=1 TO K: IF H(J)>HG THEN HG=H(J)
330 NEXT J
340 REM NORMIERUNG DER HAEUFIGKEITEN
350 FOR J=1 TO K
360 R(J)=H(J)*(14/HG)
370 R(J)=INT(R(J)+.5)
380 NEXT J
390 REM ZEICHNUNG
400 PRINTCHR$(147)
410 FOR J=1 TO K
420 ZA=17-R(J):ZE=17
430 S=J*3+8
440 FOR Z=ZA TO ZE:A=1024+S+Z*40
450 POKE A,66:NEXT Z
460 NEXT J
470 FOR S=8 TO 32
480 A=1024+S+40*Z
490 POKE A,64:NEXT S
500 W=48
510 FOR J=1 TO K
520 S=J*3+8
530 Z=19:A=1024+S+Z*40
540 POKE A,W
550 W=W+1:NEXT J
560 END
900 DATA 3,4,2,6,0,0,4,2,5,1
910 DATA 2,0,0,0,1,1,0,0,3,1
920 DATA 0,1,2,3,2,1,1,1,0,1
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!■"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A	=	Bildschirmadresse
A\$	=	Tastatureingabe
H()	=	Häufigkeiten
HG	=	Größte Häufigkeit
I	=	Laufindex (Werte)
J	=	Laufindex (Klassen)
K	=	Klassenanzahl
N	=	Anzahl der Werte
R()	=	Normierte Häufigkeiten
S	=	Bildschirmspalte
W	=	Codezahl für Ziffern
X()	=	Werte des Datenbestands
XG	=	Größter Wert
XK	=	Kleinster Wert
Z	=	Bildschirmzeile
ZA	=	Erste Bildschirmzeile (oben)
ZE	=	Letzte Bildschirmzeile (unten)

o Programmbeschreibung :

Satz 10-290 :

Überschrift, Erläuterungen, Vorgabe von N (Zahl der Werte), Dimensionierung, Einlesen der Werte, Bestimmung des kleinsten und des größten Wertes, Bestimmung der Klassenanzahl K, Dimensionierung der Häufigkeiten und Auszählen derselben; dies entspricht alles der detaillierteren Darstellung in Abschnitt 6.3.2

Satz 300-330 :

Bestimmung der größten Häufigkeit HG durch Vergleich aller Häufigkeiten H(J) mit der jeweils schon größten festgestellten Häufigkeit HG (Vorgabe  $HG=0$  (Satz 310), damit schon die erste echte Häufigkeit diesen Wert ersetzen kann)

Satz 340-380 :

Normierung der Häufigkeiten, so daß die Maximalhäufigkeit HG sich über 14 Bildschirmzeilen erstreckt (360); diese normierten Häufigkeiten R(J) werden in Satz 370 ganzzahlig gerundet (es gibt ja nur ganzzahlige Bildschirmadressen)

Satz 390-400 :

Beginn der Zeichnung durch Löschen des Bildschirms

Satz 410 :

Beginn der J-Schleife (J=1 : Erste Häufigkeit = erster Stab; J=2 : Zweite Häufigkeit = zweiter Stab usw.)

Satz 420 :

Bestimmung von Stabanfang (Zeile ZA) und Stabende (Zeile ZE) :

Das Stabende legen wir auf Zeile 17 (dies ist willkürlich; dieser Wert muß aber mindestens so groß sein, wie der maximale normierte Wert, den wir mit 14 festgelegt haben (Satz 360));

der Stabanfang ergibt sich dann aus  $17-R(J)$ ; man muß sich dabei daran erinnern, daß die Bildschirmzeilen von oben nach unten numeriert sind

Satz 430 :

Bestimmung der Stabspalte S :

Wir haben willkürlich festgesetzt :  $S = J*3 + 8$ , d.h. der erste Stab erscheint in Spalte 11 ( $J=1$ ), der zweite in Spalte 14 ( $J=2$ ) usw.

Satz 440-450 :

Belegung aller Adressen zwischen ZA und ZE mit senkrechtem Strich; wir erinnern in diesem Zusammenhang daran, daß sich eine bestimmte Adresse bei gegebener Zeile Z und Spalte S sich ergibt zu :

$$A = 1024 + S + Z*40$$

Satz 470-490 :

Über alle Spalten von Spalte 8 bis Spalte 32 wird in Zeile 17 ein waagrechter Strich gezogen

Satz 500-550 :

Belegung aller Bildschirmadressen in Zeile 19 und den Stabspalten mit der Codezahl W, die die Werte 48, 49, ..., 54 durchläuft und damit stellvertretend steht für die Ziffern 0, 1, 2, ..., 6 (siehe C64-Bedienungshandbuch, S.135);

wir erreichen damit, daß unter den Stäben in der Graphik die Kinderzahlen ausgegeben werden

Satz 560 :

Beendigung des Hauptprogramms

Satz 900-920 :

Datenbestand

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

o Programmergebnisse

Die Ergebnisse dieses Programms brauchen hier  
nicht vorgestellt zu werden (auch bei den fol-  
genden Graphikprogrammen wollen wir darauf ver-  
zichten); sie entsprechen der Skizze, wie wir  
sie in Abschnitt 7.1.1 vorgestellt haben

o Ergänzungen

Auch hier können wir uns kurz fassen und verweisen  
auf die Anmerkungen zu dem Programm zur Erstellung  
der tabellarischen Häufigkeitsverteilung in Ab-  
schnitt 6.3.2

### 7.2.2 Kreisdiagramm

Wir haben im Abschnitt 7.1.1 darauf hingewiesen, daß das Stabdiagramm zur graphischen Darstellung einer univariaten Häufigkeitsverteilung nicht geeignet ist bei niedriger Skalenqualität, also zum Beispiel, wenn die Untersuchungsvariable die bevorzugte politische Partei ist.

In solchen und ähnlichen Fällen ist ein Kreisdiagramm besser geeignet, wie es in Abschnitt 7.1.1 schon vorgestellt wurde. Dafür soll nun das geeignete BASIC-Programm entwickelt werden.

#### o Vorstellung des Problems

Für die in Abschnitt 7.1.1 dargestellte tabellarische Verteilung von Stimmanteilen für politische Parteien soll ein Kreisdiagramm per BASIC-Programm gezeichnet werden.

#### o Problemanalyse

Die Zeichnung, wie sie hier angestrebt wird, kann per BASIC-Programm sinnvoll nur in der sog. hochauflösenden Graphik gezeichnet werden. Es empfiehlt sich deshalb für den ungeübten Leser, den entsprechenden Abschnitt im Kapitel über die Programmiersprache BASIC noch einmal zu studieren.

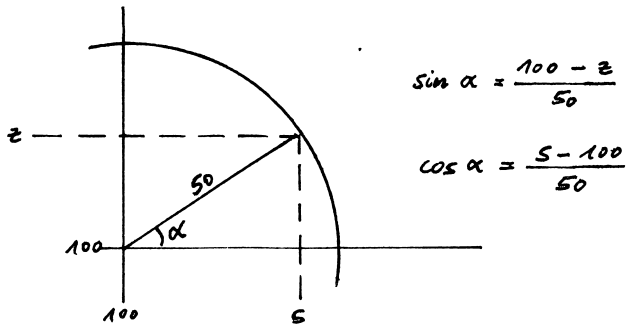
Unter problemanalytischen Gesichtspunkten ergeben sich keine besonderen Schwierigkeiten. Die ent-

scheidenden Punkte sind die folgenden :

Zunächst müssen die eingelesenen Prozentwerte in Kreiswinkel umgerechnet werden, wobei die Dreisatzrechnung davon auszugehen hat, daß 360 Grad insgesamt 100 % entsprechen.

Wenn die Winkel festliegen, kann man mit Hilfe des Sinussatzes und des Cosinussatzes die Koordinaten der Punkte bestimmen, in denen sich die Schenkel der Winkel mit der Kreislinie schneiden, wobei wir von einem Radius der Größe 50 ausgehen (50 Bildschirmspalten bzw. -Zeilen in hochauflösender Graphik).

Dies veranschaulicht die folgende Skizze :



Nach der Vorbereitung des Bildschirms für die hochauflösende Graphik kann dann der Kreis gezeichnet werden, wobei wir von der Polarkoordinaten - Kreisgleichung ausgehen, die folgende Form hat :

$$X = 100 + 50 * \cos(W)$$

$$Y = 100 + 50 * \sin(W)$$

Dabei sind X und Y die Koordinaten der Kreis-

linie und  $W$  bezeichnet die Winkel von 0 Grad bis  $2*PI$  (= 360 Grad) ( $PI$  ist der Kreisparameter, dessen Wert etwa 3.14.. beträgt). Im Programm lassen wir  $W$  in einer Schleife von 0 bis  $2*PI$  mit einer Schrittweite von 0.02 laufen; auf diese Weise erhalten wir genügend eng beieinander liegende Punkte der Kreislinie.

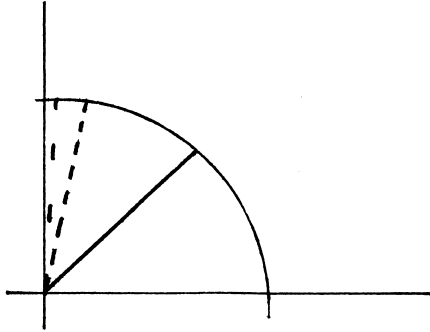
Wir haben den Mittelpunkt des Kreises der Einfachheit halber bei den Koordinaten 100/100 fixiert, den Radius mit 50 festgelegt und erinnern daran, daß auf dem Bildschirm oben die Zeile mit der Nummer 0 und unten die mit der höchsten Nummer ist - also gerade umgekehrt, wie wir es in einem normalen Achsenkreuz aus der Schulmathematik gewohnt sind.

Wenn der Kreis gezeichnet ist, können vom Kreismittelpunkt zu den Sektorenpunkten auf der Kreislinie Strecken gezeichnet werden (wie diese Punkte bestimmt werden können, wurde oben beschrieben).

Im Zusammenhang damit, daß auch diese Strecken aus einzelnen Punkten in der hochauflösenden Graphik zusammengesetzt werden, ist bei der Bestimmung der X- und Y-Koordinaten dieser Punkte folgendes zu beachten :

Wir können eine Strecke in der Weise zeichnen, daß wir eine Schleife vom Kreismittelpunkt bis zum Sektorenpunkt über alle X-Werte laufen lassen und die dazugehörigen Y-Werte per Geradengleichung (es empfiehlt sich dabei die Zwei-Punkte-Gleichung) bestimmen und dann den jeweiligen Punkt zeichnen.

Dies würde aber bei steilen Linien (Steigungswinkel zwischen 45 und 135 Grad) zu einer relativ "lockeren" Punktfolge führen, wie die folgende Skizze verdeutlicht :



Bei einer senkrechten Geraden hätte man auf dem Bildschirm sogar nur zwei Punkte, nämlich den Kreismittelpunkt und den senkrecht darüber (oder darunter) liegenden Sektorenpunkt, d.h. die zu zeichnende Strecke wäre nicht zu sehen.

Deshalb ist es bei steilen Geraden sinnvoller, eine Punkteschleife über alle Y-Werte laufen zu lassen und die dazugehörigen X-Werte per Geradengleichung zu bestimmen.

In Abhängigkeit von dem jeweiligen Winkel ist in dem folgenden Programm eine entsprechende Verzweigung eingebaut.

Weiterhin ist zu beachten, daß in Abhängigkeit vom zu zeichnenden Sektorenwinkel, die Zeichenschleife von links nach rechts (bzw. von oben nach unten) oder von rechts nach links (bzw. von unten nach oben) anzulegen ist. Dies geschieht durch entsprechende Belegung des Schrittweitenparameters (STEP) mit 1 bzw. - 1.

o Programm

```
10 REM S72-KREIS
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"#BEISPIEL S72"
70 PRINTTAB(13)"KREISDIAGRAMM"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM DIENT DAZU, EINE VER-"
100 PRINT"TEILUNG RELATIVER HAEUFIGKEITEN IN"
110 PRINT"FORM EINES KREISDIAGRAMMS (SOG. KUCHEN-"
120 PRINTTAB(8)"DIAGRAMM) DARZUSTELLEN."
130 PRINT:PRINT"ALS AUSGANGSDATEN VERWENDEN WIR PRO-"
140 PRINT"ZENTWERTE (STIMMANTEILE) FUER POLITI-"
150 PRINTTAB(13)"SCHE PARTEIEN."
160 GOSUB 1000:REM WARTEN
165 PRINT"IN DIESEM BEISPIEL SIND 5 PROZENTWERTE"
166 PRINT"VORGEGEBEN. DESHALB IST AUF DIE FOLGEN-"
167 PRINT"DE FRAGE DER WERT 5 EINZUGEBEN.":PRINT:PRINT
170 PRINT"WIEVIELE PROZENTWERTE LIEGEN VOR ?"
180 INPUT "(WIEVIELE KLASSEN) (RETURN) ";N
190 DIM SP(N),ZP(N),W(N),F(N),WK(N),P$(N)
200 FOR I=1TON:READ F(I):NEXT I
210 FOR I=1TON:READ P$(I):NEXT I
220 PRINTCHR$(147):PRINT"KONTROLLAUSGABE":PRINT:PRINT
230 FOR I=1 TO N:PRINTP$(I),F(I):NEXT I
240 GOSUB 1000:REM WARTEN
250 FOR I=1 TO N:W(I)=F(I)*3.6:NEXT I
260 FOR I=1 TO N:WK(I)=WK(I-1)+W(I):NEXT I
```

```

270 REM BESTIMMUNG DER KOORDINATEN
280 SP(0)=150:ZP(0)=100
290 FOR I=1 TO N
300 WK(I)=(WK(I)/180)*π:NEXT I
310 FOR I=1 TO N
320 ZP(I)=100-50*SIN(WK(I))
330 SP(I)=100+50*COS(WK(I))
340 NEXT I
350 REM KREIS
360 GOSUB 1900:REM SCREEN
370 FOR W=0 TO 2*π STEP .02
380 X=100+50*COS(W):Y=100+50*SIN(W)
390 GOSUB 2000:REM PUNKT
400 NEXT W
410 REM SEKTOREN
420 FOR I=1 TO N
430 IF ABS(SP(I)-100)<=34 THEN 510
440 SS=-1
450 IF SP(I)>=100 THEN SS=1
460 FOR X=100 TO SP(I) STEP SS
470 Y=((ZP(I)-100)/(SP(I)-100))*(X-100)+100
480 GOSUB 2000:REM PUNKT
490 NEXT X
500 GOTO 570
510 SS=-1
520 IF ZP(I)>=100 THEN SS=1
530 FOR Y=100 TO ZP(I) STEP SS
540 X=((SP(I)-100)/(ZP(I)-100))*(Y-100)+100
550 GOSUB 2000:REM PUNKT
560 NEXT Y
570 NEXT I
580 FOR I=1 TO 5000:NEXT I
590 GOSUB 2100:REM TEXT
600 PRINTCHR$(147)
610 PRINT"ENDE DER AUSGABE":END

```

```

900 DATA 39.4,44.6,5.5,8.1,2.4
910 DATA CDU,SPD,FDP,GRUENE,SONSTIGE
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !!"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2*(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
EX	=	Hilfsgröße im Unterprogramm Punkt
F()	=	Prozentwerte (Ausgangsdaten)
I	=	Laufindex (Werte, Sektoren)
N	=	Anzahl der Werte
P\$()	=	Parteinamen
S	=	Bildschirmspeicheradresse
SP()	=	Spaltenkoordinaten der Sektorenpunkte
SS	=	Schrittweite in der Zeichenschleife
V	=	Adresse des Videochips
W	=	Laufindex (Kreiswinkel)
W()	=	Sektorenwinkel
WK()	=	Kumulierte Sektorenwinkel
X	=	Koordinate des zu zeichnenden Punktes
XK	=	Koordinate im Unterprogramm Punkt
Y	=	Koordinate des zu zeichnenden Punktes
YK	=	Koordinate im Unterprogramm Punkt
ZP()	=	Zeilenkoordinaten der Sektorenpunkte

o **Programmbeschreibung**

Satz 10-167 :

Überschrift und Erläuterungen

Satz 170-180 :

Anforderung der Zahl der Werte

Satz 190 :

Dimensionierung der Koordinaten der Sektorenpunkte, der Anteilswerte, der Winkel, der kumulierten Winkel und der Parteinamen

Satz 200-210 :

Einlesen der Anteilswerte und der Parteinamen

Satz 220-240 :

Kontrollausgabe der Ausgangsdaten

Satz 250-260 :

Umrechnung der Anteilswerte in Winkel und Bestimmung der kumulierten Winkel (jeder Winkel wird ja in der Zeichnung an den vorhergehenden angefügt)

Satz 270-280 :

Bestimmung der Koordinaten des ersten Sektorenpunktes (er liegt auf der Höhe des Kreismittelpunktes, um 50 Einheiten rechts von ihm, also quasi bei "3 Uhr")

Satz 290-300 :

Berechnung der kumulierten Winkel in Einheiten des

Kreisparameters PI

Satz 310-340 :

Bestimmung der Koordinaten der übrigen Sektorenpunkte auf der Kreislinie mit dem Sinus- und dem Cosinussatz

Satz 350-360 :

Umschalten in den Graphikmodus und Vorbereitung des Bildschirms durch Sprung ins Unterprogramm 1900

Satz 370-400 :

Zeichnen des Kreises durch Bestimmung der Polarkoordinaten in Abhängigkeit vom sich mit der Schrittweite 0.02 verändernden Winkel  $W$  (Satz 380) und jeweils Zeichnen des Punktes durch Sprung ins Unterprogramm 2000 (Satz 390)

Satz 410-420 :

Beginn der Schleife zum Zeichnen der Sektorenlilien

Satz 430 :

Wenn die absolute Differenz zwischen der jeweils zu benutzenden Spaltenkoordinate und der Mittelpunktsspalte (100) kleiner gleich 34 ist, liegt der Steigungswinkel zwischen 45 und 135 Grad; dann geht es weiter bei Satz 510 (senkrechte Zeichenschleife über die Y-Werte); wenn nicht, geht es weiter bei Satz 440

Satz 440 :

Belegung der Schrittweite für die Zeichenschleife mit -1 (Zeichnen von rechts nach links)

Satz 450 :

Wenn aber die Spaltenkoordinate rechts vom Kreis -

Mittelpunkt liegt, wird die Schrittweite mit 1 belegt (Zeichnen von links nach rechts)

Satz 460-490 :

Zeichnen der Strecke vom Kreismittelpunkt bis zum Sektorenpunkt auf der Kreislinie (Zeichenschleife über alle X von 100 (Mittelpunkt) bis SP (Spaltenkoordinate des Sektorenpunktes) mit der Schrittweite SS (siehe auch Satz 440 und 450) unter Nutzung der Zwei-Punkte-Formel (Satz 470))

Satz 500 :

Sprung zum Satz 570, d.h. Überspringen des "senkrechten" Zeichnens

Satz 510-560 :

Satz 510 wird nur erreicht, wenn eine relativ steile Strecke zu zeichnen ist (vergl. Satz 430); dann läuft der gleiche Prozeß ab, wie unter Satz 440 - 490, mit dem Unterschied, daß die Zeichenschleife jetzt über alle Y-Werte läuft und die X-Koordinate jeweils mit der Zwei-Punkte-Formel bestimmt wird

Satz 570 :

Inangriffnahme des nächsten Sektors (zurück zu Satz 430)

Satz 580 :

Warteschleife

Satz 590 :

Sprung ins Unterprogramm 2100 zum Zurückschalten in den Textmodus

Satz 600-610 :

Beendigung des Hauptprogramms

Satz 900-910 :

Daten

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

Satz 1900-1950 :

Unterprogramm zum Vorbereiten des Bildschirms für  
die hochauflösende Graphik

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in der  
hochauflösenden Graphik

Satz 2100-2120 :

Unterprogramm zum "Zurückschalten" in den sog.  
Textmodus

o Programmergebnisse

Das Ergebnis dieses Programms wird nicht vorge-  
stellt; wir verweisen auf die entsprechende Zeich-  
nung in Abschnitt 7.1.1.

## o Ergänzungen

Es ist leicht zu sehen, an welchen Stellen Programmänderungen erforderlich sind, wenn man einen anderen Datenbestand darstellen möchte (Satz 900-910 und veränderte Eingabe im Satz 180) oder wenn man z.B. einen größeren Kreis bevorzugen würde (Änderungen überall dort, wo die 50 auftaucht, z. B. Satz 280, 320, 330, 380 etc.) oder wenn man den Kreismittelpunkt verlagern möchte (überall, wo die 100 auftaucht) oder wenn man gern andere Farben hätte (Satz 1940).

Wer dieses Programm benutzt, wird feststellen, daß die Graphik relativ langsam erzeugt wird. Wesentliche Beschleunigungen können nur erreicht werden, wenn man Unterprogramme in Maschinensprache einbaut.

### 7.2.3 Histogramm

## o Vorstellung des Problems

Für die univariate Häufigkeitsverteilung einer stetigen Variablen soll nun auch die graphische Darstellung programmiert werden. Wie aus Abschnitt 7.1.2 schon bekannt ist, ist dies das Histogramm. Als Ausgangsdaten für dieses Programm wählen wir 50 Angaben zum Körpergewicht zufällig ausgewählter Personen.

## o Problemanalyse

Wir können uns an den Ausführungen in Abschnitt 6.3.3 orientieren, soweit die tabellarische Darstellung des Datenbestands betroffen ist, an den Ausführungen des Abschnitts 7.2.1 (Stabdiagramm) bezüglich der Umsetzung der Tabellenwerte in eine Graphik und am vorangegangenen Abschnitt, was das Zeichnen in hochauflösender Graphik betrifft.

Zusätzlich ist lediglich zu berücksichtigen, daß nun aneinander anschließende Rechtecke gezeichnet werden müssen, was in der hochauflösenden Graphik erfordert, daß für jedes dieser Rechtecke jeweils vier Werte fixiert werden, nämlich :

- Bildschirmspalte links (SA)
- Bildschirmspalte rechts (SE)
- Bildschirmzeile oben (ZE)
- Bildschirmzeile unten (ZA)

Die Spalten links und rechts ergeben sich, wenn man die Zahl der Rechtecke (Anzahl der Klassen der Häufigkeitsverteilung) auf die zur Verfügung stehende Bildschirmbreite "umverteilt" (wir gehen dabei grundsätzlich von konstanter Klassenbreite in der Verteilung aus).

Die Bildschirmzeile unten ist für jedes Rechteck die gleiche, während die obere sich unter Berücksichtigung der darzustellenden Häufigkeiten, normiert auf die zur Verfügung stehende Bildschirmhöhe, ergibt.

o Programm

```
10 REM S73-HISTOGRAMM
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(14)"#BEISPIEL S73"
70 PRINTTAB(15)"HISTOGRAMM"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM DIEN T DAZU, EINE STE-"
100 PRINT"TIGE HAEUEFIGKEITSVERTEILUNG IN HOCHAUF-"
110 PRINTTAB(5)"LOESENDER GRAPHIK AUSZUGEBEN."
130 PRINT:PRINT"ALS AUSGANGSDATEN VERWENDEN WIR 50 AN-"
140 PRINT"GABEN ZUM GEWICHT ZUFAELLIG AUSGEWAEHL-"
150 PRINTTAB(13)"TER PERSONEN."
160 GOSUB 1000:REM WARTEN
170 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
180 N=50
190 DIM X(N)
200 FOR I=1 TO N:READ X(I):NEXT I
210 XG=0:XK=1000
220 FOR I=1 TO N
230 IF X(I)<XK THEN XK=X(I)
240 IF X(I)>XG THEN XG=X(I)
250 NEXT I
260 PRINT:PRINT:PRINT
270 PRINT"KLEINSTER WERT = ";XK
280 PRINT"GROESSTER WERT = ";XG:PRINT:PRINT
290 INPUT"GEWUENSCHTE KLASSENBREITE (RETURN)";B
300 IF B>(XG-XK)/2 THEN PRINT"ZU HOHER WERT":GOTO 290
305 IF B<(XG-XK)/20 THEN PRINT"ZU KLEINER WERT":GOTO 290
310 IF (XG-XK)/B>N THEN PRINT"UNGEEIGNETER WERT":GOTO 290
320 PRINT:INPUT"START DER KLASSENBI L DUNG (RETURN) ";S
330 IF S>XK THEN PRINT"ZU HOHER WERT":GOTO 320
335 IF S<XK-B THEN PRINT"ZU KLEINER WERT":GOTO 320
```

```

340 PRINTCHR$(147):PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
350 K=INT((XG-S)/B)+1
360 DIM F(K),XA(K),XE(K)
370 FOR J=1 TO K:XA(J)=S+(J-1)*B
380 XE(J)=XA(J)+B:NEXT J
390 FOR I=1 TO N:FOR J=1 TO K
400 IF X(I)<XE(J) AND X(I)>=XA(J) THEN F(J)=F(J)+1:GOTO 420
410 NEXT J
420 NEXT I
430 PRINTCHR$(147):PRINT"TABELLE":PRINT:PRINT
440 PRINT"      GEWICHT          HAEUFIGKEIT":PRINT
450 FOR J=1 TO K
460 PRINTXA(J);" BIS UNTER";XE(J);" KG",F(J)
470 NEXT J
475 GOSUB 1000:REM WARTEN
480 FOR J=1 TO K:IF F(J)>FG THEN FG=F(J)
490 NEXT J
500 REM NORMIERUNG (FG=150)
510 FOR J=1 TO K:F(J)=(150/FG)*F(J)
520 F(J)=INT(F(J)+.5):NEXT J
530 REM ZEICHNEN VORBEREITEN
540 DIM SA(K),SE(K),ZE(K)
550 ZA=160:B=INT(300/K)
560 FOR J=1 TO K
570 SA(J)=10+(J-1)*B:SE(J)=SA(J)+B
580 ZE(J)=160-F(J):NEXT J

```

```

590 REM ZEICHNEN
600 GOSUB 1900:REM SCREEN
610 REM ACHSENKREUZ
620 FOR X=3TO310
630 Y=160:GOSUB 2000:REM PUNKT
640 NEXT X
650 FOR Y=3TO160
660 X=3:GOSUB 2000:REM PUNKT
670 NEXT Y
680 FOR J=1TOK
690 FOR Y=ZATOZE(J)STEP-1:X=SA(J)
700 GOSUB 2000:REM PUNKT
710 NEXT Y
720 FOR X=SA(J)TOSE(J):Y=ZE(J)
730 GOSUB 2000:NEXT X
740 FOR Y=ZE(J)TO ZA:X=SE(J)
750 GOSUB 2000:NEXT Y
760 NEXT J
770 FOR I=1TO5000:NEXTI
780 GOSUB 2100:REM TEXT
790 PRINTCHR$(147)
800 PRINT"ENDE DER AUSGABE":END
900 DATA 65,77,88,99,85,62,68,74,76,96
910 DATA 77,75,76,78,91,69,86,86,93,94
920 DATA 78,88,85,83,82,80,90,70,68,69
930 DATA 94,72,73,74,75,76,77,83,82,81
940 DATA 72,75,78,81,82,84,78,74,91,69
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !■"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

```
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2↑(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN
```

o Variablenliste

A\$	=	Tastatureingabe
B	=	Klassenbreite
EX	=	Hilfsgröße im Unterprogramm PUNKT
F()	=	Häufigkeiten
FG	=	Größte Häufigkeit
I	=	Laufindex
J	=	Laufindex
K	=	Klassenanzahl
N	=	Anzahl der Werte
S	=	Startpunkt der Klassenbildung und Speicheradresse im Unterprogramm PUNKT
SA()	=	Bildschirmspalte Klassenanfang
SE()	=	Bildschirmspalte Klassenende
V	=	Speicheradresse des Videochips
X	=	Bildschirmkoordinate Punkt
X()	=	Werte des Datenbestands
XA()	=	Klassenanfang
XE()	=	Klassenende
XG	=	Größter Wert
XK	=	Kleinster Wert und Hilfskoordinate im Unterprogramm PUNKT

Y = Bildschirmkoordinate Punkt  
YK = Hilfskoordinate im Unterprogramm PUNKT  
ZA() = Bildschirmzeile Unterkante Rechteck  
ZE() = Bildschirmzeile Oberkante Rechteck

o Programmbeschreibung

Satz 10-475 :

Erzeugung der Häufigkeitstabelle wie im Programm im Abschnitt 6.3.3 (siehe dort Satz 10-520) :

Überschrift, Erläuterungen (10-170); Vorgabe der Anzahl der Werte (180); Dimensionierung (190) und Einlesen (200); Bestimmung des kleinsten und des größten Wertes (210-280); Anforderung der Klassenbreite und des Klassifizierungs-Startpunktes (290-340); Bestimmung der Klassenanzahl und Dimensionierungen (350-360); Bestimmung der Klassenanfangs- und -endpunkte (370-380); Ermittlung der Häufigkeiten (390-420); Ausgabe der Tabelle (430-475)

Satz 480-520 :

Ermittlung der größten Häufigkeit und Normierung aller anderen Häufigkeiten

Satz 530-550 :

Vorbereitung der Zeichnung durch Dimensionierung der Bildschirmspalten und -zeilen gemäß der Zahl der zu zeichnenden Rechtecke (540); Belegung der unteren Bildschirmzeile für alle Rechtecke (550) und Normierung der Klassenbreite B zur Ausnutzung der Bildschirmbreite (550)

**Satz 560-580 :**

**Bestimmung der linken und der rechten Bildschirm-  
spalte und der oberen Bildschirmzeile für jedes  
Rechteck**

**Satz 590-600 :**

**Vorbereitung des Bildschirms zum Zeichnen in der  
hochauflösenden Graphik**

**Satz 610-640 :**

**Zeichnen einer waagrechten Achse in Bildschirmzei-  
le Nr.160**

**Satz 650-670 :**

**Zeichnen einer senkrechten Achse in Spalte 3**

**Satz 680 :**

**Beginn der Schleife zum Zeichnen der Rechtecke**

**Satz 690-710 :**

**Zeichnen der linken Seite des Rechtecks**

**Satz 720-730 :**

**Zeichnen der oberen Seite des Rechtecks**

**Satz 740-750 :**

**Zeichnen der rechten Seite des Rechtecks**

**Satz 760 :**

**Inangriffnahme des nächsten Rechtecks**

Satz 770-800 :

Warteschleife (770), Umschalten in den Textmodus (780) und Beendigung des Hauptprogramms (790-800)

Satz 900-940 :

Datenbestand

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 1900-1950 :

Unterprogramm zur Vorbereitung des Bildschirms für die hochauflösende Graphik

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in hochauflösender Graphik

Satz 2100-2120 :

Unterprogramm zum Zurückschalten in den Textmodus

o Programmergebnisse

Die Ergebnisse entsprechen der Darstellung in Abschnitt 7.1.2

o Ergänzungen

Wir verweisen wieder auf die Ergänzungen, die im Abschnitt 6.3.3 schon vorgetragen wurden.

7.2.4 Streudiagramm

o Vorstellung des Problems

Zum Abschluß dieses Kapitels soll ein Programm vorgestellt werden, das eine bivariate Häufigkeitsverteilung graphisch darstellt.

Dies ist sinnvoll nur möglich, wenn man von der ursprünglichen Datenmatrix, also von den gegebenen Wertepaaren ausgeht. Dann nämlich kann eine sog. Punktwolke erzeugt werden, wie wir sie in Abschnitt 7.1.3 schon schematisch gezeigt haben.

Als Ausgangsdaten wählen wir 50 Angaben zu Körpergröße und Gewicht zufällig ausgewählter Personen.

o Problemanalyse

Zur Lösung dieser Aufgabe sind die folgenden Arbeitsschritte zu durchlaufen :

- Einlesen der Daten

- Bestimmung der kleinsten und der größten Werte für beide Variablen
- Normierung der Wertepaare zur Ausnutzung des Bildschirms
- Zeichnen eines Achsenkreuzes und der einzelnen Punkte

o Programm

```

10 REM S74-STREUDIAGRAMM
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S74"
70 PRINTTAB(13)"STREUDIAGRAMM"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM STELLT EINE BIVARIATE"
100 PRINT"HAEUFIGKEITSVERTEILUNG GRAPHISCH IN DER"
110 PRINT"    FORM EINES STREUDIAGRAMMS DAR."
130 PRINT:PRINT"ALS AUSGANGSDATEN VERWENDEN WIR 50 AN-"
140 PRINT"GABEN ZU GROESSE UND GEWICHT ZUFAELLIG"
150 PRINTTAB(9)"AUSGEWAELTER PERSONEN."
160 GOSUB 1000:REM WARTEN
170 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
180 N=50
190 DIM X(N),Y(N)
200 FOR I=1 TO N:READ X(I):NEXT I
210 FOR I=1 TO N:READ Y(I):NEXT I
215 PRINTCHR$(147)
220 PRINT"KONTROLLAUSGABE":PRINT
230 FOR I=1 TO N:PRINTI,X(I),Y(I)
240 IF I/15=INT(I/15) THEN GOSUB 1000
250 NEXT I

```

```

255 PRINT:PRINT:PRINTTAB(14)"#MOMENT BITTE#"
260 XK=200:YG=0:YK=200:YG=0
270 FOR I=1 TO N
280 IF X(I)<XK THEN XK=X(I)
290 IF X(I)>XG THEN XG=X(I)
300 IF Y(I)<YK THEN YK=Y(I)
310 IF Y(I)>YG THEN YG=Y(I)
320 NEXT I
330 FOR I=1TON
340 X(I)=(290/(XG-XK))*(X(I)-XK)+10
350 Y(I)=(140/(YG-YK))*(Y(I)-YK)+10
360 Y(I)=150-Y(I)
370 NEXT I
380 REM ZEICHNEN
390 GOSUB 1900:REM SCREEN
400 FOR X=3 TO 315:Y=160
410 GOSUB 2000:REM PUNKT
420 NEXT X
430 FOR Y=3 TO 160:X=3
440 GOSUB 2000:REM PUNKT
450 NEXT Y
460 FOR I=1 TO N
470 X=X(I):Y=Y(I):GOSUB 2000:REM PUNKT
480 NEXT I
490 FOR I=1TO9999:NEXT I
500 GOSUB 2100:REM TEXT
510 PRINTCHR$(147)
520 PRINTTAB(8)"ENDE DER AUSGABE":END

```

```

850 DATA 165,170,190,192,182,168,168,175,172,194
860 DATA 172,175,177,180,190,170,192,195,190,183
870 DATA 180,190,181,178,175,180,183,165,162,164
880 DATA 184,170,170,168,171,179,180,180,175,165
890 DATA 163,165,162,178,175,180,185,176,190,165
900 DATA 65,77,88,99,85,62,68,74,76,96
910 DATA 77,75,76,78,91,69,86,86,93,94
920 DATA 78,88,85,83,82,80,90,70,68,69
930 DATA 94,72,73,74,75,76,77,83,82,81
940 DATA 72,75,78,81,82,84,78,74,91,69
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !□"
1030 GET A$:IF A#="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2↑(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
EX	=	Hilfsgröße im Unterprogramm PUNKT
I	=	Laufindex
N	=	Anzahl der Fälle (Wertepaare)
S	=	Bildschirmadresse
V	=	Adresse des Videochips
X	=	Bildschirmkoordinate Punkt
X()	=	X-Werte
XG	=	Größter X-Wert
XK	=	Kleinster X-Wert und Hilfskoordinate im Unterprogramm PUNKT
Y	=	Bildschirmkoordinate Punkt
YG	=	Größter Y-Wert
YK	=	Kleinster Y-Wert und Hilfskoordinate im Unterprogramm PUNKT

o Programmbeschreibung

Satz 10-170 :

Überschrift und Erläuterungen

Satz 180-210 :

Angabe der Anzahl der Wertepaare, Dimensionierung und Einlesen der Werte (es ist dabei zu beachten, daß wir in den DATA-Anweisungen zuerst alle X-Werte und dann alle Y-Werte angegeben haben; Satz 850-940)

Satz 220-250 :

Kontrollausgabe der Datenmatrix mit Unterbrechung nach je 15 Wertepaaren

Satz 260-320 :

Bestimmung der kleinsten und der größten X- und Y-Werte

Satz 330-370 :

Normierung der X- und der Y-Werte, so daß die Bildschirmspalten 10-300 und die Bildschirmzeilen 10-150 ausgenutzt werden

Satz 380-450 :

Vorbereitung des Bildschirms für die hochauflösende Graphik und Zeichnen eines Achsenkreuzes

Satz 460-480 :

Zeichnen der Punkte des Streudiagramms

Satz 490-520 :

Warteschleife, Umschalten in den Textmodus und Beendigung des Hauptprogramms

Satz 850-940 :

Datenbestand

Satz 1000-2120 :

Unterprogramme (vergl. Abschnitt 7.2.3)

o Ergebnisse

Auch hier verzichten wir auf die Darstellung der Ergebnisse und verweisen auf Abschnitt 7.1.3

o Ergänzungen

Unter ergänzenden Gesichtspunkten dürfen wir auf die Ausführungen bei den vorangegangenen Graphik - Programmen verweisen. Es treten hier keine neuen Aspekte hinzu.

## Kapitel 8 : Mittelwerte und Streuungen

### 8.1 Statistische Grundlagen

Die Statistik stellt nicht nur Methoden bereit, um Daten in geeigneter Form darzustellen, wie das in den vorangegangenen beiden Kapiteln gezeigt wurde, sondern sie ermöglicht es auch, so entstandene Häufigkeitsverteilungen durch Maßzahlen zusammenfassend zu beschreiben.

Dabei wird - wie schon bei den Methoden der Darstellung von Häufigkeitsverteilungen - generell zwischen uni-, bi- und multivariaten Fällen unterschieden, d.h. danach, ob nur eine Variable, zwei Variablen gleichzeitig oder mehr als zwei Variablen gleichzeitig zum Gegenstand der Betrachtung gemacht werden.

In diesem Kapitel wenden wir uns ausschließlich der univariaten Statistik zu, d.h. wir betrachten statistische Maßzahlen, die sich jeweils auf nur eine interessierende Variable beziehen.

Es gibt eine große Zahl derartiger Maßzahlen. Ihre Aufgabe ist es ganz allgemein, mit Hilfe eines zusammenfassenden Ausdrucks, also zum Beispiel mit einem einzigen Zahlenwert, wesentliche Informationen charakterisierender Art aus einem gegebenen Datenbestand "herauszuziehen".

Mit dem Informationsgewinn durch Tabellierung und graphische Darstellung begnügt man sich also nicht, sondern man geht darüber hinaus und "verdichtet" gewissermaßen die vorliegenden Einzelinformationen so weit, daß nur noch ein "Konzentrat" bleibt, das in einprägsamer Weise wichtige Eigen-

heiten einer gegebenen Häufigkeitsverteilung präsentiert.

Die wichtigsten dieser Maßzahlen sind Mittelwerte und Streuungsmaße.

### 8.1.1 Mittelwerte

Mittelwerte dienen dazu, die Mitte einer gegebenen Häufigkeitsverteilung zu charakterisieren, also anzugeben, um welchen Wert herum, sich die Hauptmasse der Merkmalswerte konzentriert. Sie bezeichnen die sog. "zentrale Tendenz" einer Verteilung.

Die Angabe dieser zentralen Tendenz ist eine wichtige Detailinformation aus einem Datenbestand : Er wird in einer einzigen, recht aussagekräftigen Zahl "komprimiert".

Die wichtigsten dieser Mittelwerte sind das arithmetische Mittel und der Modus (Häufigster Wert).

Das arithmetische Mittel eines Datenbestandes ist definiert als die Summe aller Merkmalswerte geteilt durch deren Anzahl, also :

$$\bar{x} = \frac{\sum x_i}{n}$$

Der Modus hingegen ist derjenige Merkmalswert aus einem Datenbestand, der am häufigsten auftritt.

Während das arithmetische Mittel sinnvoll nur aus mindestens intervallskalierten Daten berechnet werden kann (die erforderliche Addition der Merkmalswerte ist bei Nominaldaten (Beispiel : Bevorzugte politische Partei) oder Ordinaldaten (Beispiel: Leistungsbewertungen) nicht sinnvoll durchführbar), ist der Modus auch für die schwächeren Skalenqualitäten geeignet.

Hat man eine symmetrische Häufigkeitsverteilung, dann fallen Modus und arithmetisches Mittel in einem Zahlenwert zusammen. Besteht zwischen beiden Werten hingegen eine Differenz, dann deshalb, weil die Verteilung nicht symmetrisch, also schief ist.

Generell kann man deshalb sagen, daß eine Verteilung um so schief ist, je weiter Modus und arithmetisches Mittel auseinanderfallen.

Dies bedeutet, daß die beiden Werte sich nicht nur dazu eignen, die zentrale Tendenz einer Verteilung zu charakterisieren, sondern gemeinsam auch deren Schiefe .

Ein einfaches Beispiel soll zur Illustration angefügt werden :

Befragt werden 10 zufällig ausgewählte Familien nach der Zahl ihrer Kinder, wobei sich die folgenden Daten ergeben :

0, 0, 1, 1, 1, 1, 2, 2, 3, 5

Das arithmetische Mittel dieser Verteilung liegt offenbar bei 1.6, der Modus hingegen bei 1 (Häufigster Wert).

Die Schiefe dieser Verteilung ist offenkundig: Sie führt dazu, daß das arithmetische Mittel größer ist als der Modus.

Dies ist letzten Endes darauf zurückzuführen, daß das arithmetische Mittel empfindlich auf Extremwerte reagiert, der Modus hingegen überhaupt nicht. Hätte beispielsweise die 10. Familie in unserem kleinen Datenbestand nicht 5, sondern 9 Kinder, so würde das arithmetische Mittel von 1.6 auf 2.0 steigen, der Modus bliebe dagegen unberührt.

### 8.1.2 Streuungsmaße

Der Statistiker begnügt sich nicht damit, eine Häufigkeitsverteilung durch Mittelwerte zu charakterisieren. So wichtig und aussagekräftig solche Mittelwerte auch sind, so ist doch zu berücksichtigen, daß sie nur einen bestimmten Aspekt des Datenbestandes beschreiben und andere vernachlässigen.

Stellt man beispielsweise fest, daß die Kinderzahl im Durchschnitt bei 1.6 liegt, so ist doch noch nichts darüber ausgesagt, wie eigentlich die Verteilung der Kinderzahlen aussieht, die sich hinter dieser Durchschnittszahl quasi "versteckt".

Man interessiert sich deshalb zusätzlich für die Frage, wie weit die einzelnen Merkmalswerte vom Durchschnitt abweichen und hat zur Beschreibung dieses Aspekts einer Verteilung wiederum eine ganze Reihe von Maßzahlen, die sog. Streuungsmaße, von denen wir zwei vorstellen wollen :

Das einfachste Streuungsmaß ist die Spannweite , die als die Differenz zwischen dem größten und dem kleinsten beobachteten Wert definiert ist.

Wichtiger für die praktische statistische Arbeit ist die Standardabweichung .

Sie kommt so zustande, daß man alle Abweichungen der Merkmalswerte vom arithmetischen Mittel notiert, diese Abweichungen quadriert, die quadrierten Abweichungen aufaddiert und diese Summe durch die Anzahl der Werte dividiert (dies ergibt, wenn man so will, eine durchschnittliche quadratische Abweichung). Aus diesem Rechenergebnis wird dann noch die Wurzel gezogen.

Es ergibt sich also die folgende Berechnungsformel für die Standardabweichung :

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

In unserem kleinen Zahlenbeispiel (Kinderzahlen) aus dem vorangegangenen Abschnitt erhalten wir als Spannweite den Wert 5 (= 5 - 0).

Die Standardabweichung kann gemäß der folgenden Arbeitstabelle bestimmt werden :

Kinder	Abweichung	Quadrierte Abweichung
0	- 1.6	2.56
0	- 1.6	2.56
1	- 0.6	0.36
1	- 0.6	0.36
1	- 0.6	0.36
1	- 0.6	0.36
2	0.4	0.16
2	0.4	0.16
3	1.4	1.96
5	3.4	11.56
Summe		20.40

Damit erhalten wir als Standardabweichung :

$$s = \sqrt{20.4/10} = \sqrt{2.04} = 1.428$$

Dies bedeutet, daß die Kinderzahlen im gegebenen Datenbestand im Schnitt um 1.428 (Kinder) vom arithmetischen Mittel 1.6 abweichen.

## 8.2 Programme

Zu den im vorangegangenen Abschnitt vorgestellten Maßzahlen sollen nun geeignete BASIC - Programme vorgestellt werden.

### 8.2.1 Mittelwerte

#### o Vorstellung des Problems

Das folgende Programm soll für beliebig viele, vom Benutzer einzugebenden Werte das arithmetische Mittel und den Modus bestimmen.

#### o Problemanalyse

Wenn dieses Programm für beliebige Datenbestände gelten soll, muß zunächst dem Benutzer Gelegenheit geboten werden, anzugeben, wieviele Werte in die Mittelwertberechnung eingehen sollen; diese Werte sind dann vom Programm anzufordern.

Schon während der Eingabe können die Werte aufsummiert werden, so daß nach der Eingabe sofort das arithmetische Mittel ausgerechnet und ausgegeben werden kann.

Etwas schwieriger ist die Bestimmung des Modus :

Um ihn zu bestimmen, müssen die Daten zuerst sor-

tiert werden (vergl. Abschnitt 6.3.1, wo dieses Problem schon gelöst wurde), danach ist zu prüfen, ob aufeinander folgende Werte identisch sind.

Zur Erläuterung dieser Vorgehensweise schaue sich der Leser noch einmal den kleinen (schon sortierten) Datenbestand oben an (Kinderzahlen), wo also zu prüfen wäre, wie oft die 0 aufeinander folgt, wie oft die 1 aufeinander folgt usw.

Auf diese Weise stellen wir fest, wieviele Folgen identischer Werte es gibt (im obigen Beispiel sind dies fünf Folgen, nämlich die 0-, 1-, 2-, 3- und 5-Folge).

Im nächsten Schritt stellen wir fest, wie lang diese Folgen sind, d.h. wieviele Werte sie jeweils umfassen. In dem obigen Beispiel kommen wir zu dem folgenden Ergebnis :

0 - Folge	:	2 Werte
1 - Folge	:	4 Werte
2 - Folge	:	2 Werte
3 - Folge	:	1 Wert
5 - Folge	:	1 Wert

Dies ist im Grunde nichts anderes als eine einfache Häufigkeitsverteilung.

In dieser Verteilung kann nun durch das Aufsuchen der größten Häufigkeit (4) der Modus (1) identifiziert werden.

Bei klassifiziertem Material wird, wenn die Urliste der Daten nicht mehr vorliegt, der Mittelpunkt der am stärksten besetzten Klasse als Modus bestimmt.

Es versteht sich, daß dann, wenn mehrere gleich stark besetzte Klassen auftreten, bzw. wenn verschiedene Merkmalswerte der Urliste die gleiche (maximale) Häufigkeit aufweisen, ein Modus sinnvollerweise nicht bestimmt werden kann.

o Programm

```
10 REM SB1-MITTELWERTE
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL SB1"
70 PRINT TAB(14)"MITTELWERTE":PRINT:PRINT
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT DAS ARITHMETI-"
100 PRINT"SCHE MITTEL UND DEN MODUS (HAEUFIGSTER"
110 PRINT"WERT) AUS BELIEBIG VIELEN EINZUGEBENDEN"
120 PRINTTAB(16)"WERTEN.":GOSUB 1000
130 PRINT:PRINT:PRINT:INPUT"WIEVIELE WERTE (UND RETURN)";N
135 IF N<2 THEN PRINT:PRINT"FEHLER":PRINT:GOTO 130
140 DIM X(N),Y(N),F(N):PRINT:PRINT
150 FOR I=1 TO N
160 PRINTI;:INPUT ".WERT (RETURN): ";X(I)
170 S=S+X(I)
180 NEXT I
190 AM=S/N
200 PRINT CHR$(147)
210 PRINT"ARITHMETISCHES MITTEL = ";AM;:PRINT"
```

```

220 REM MODUS
224 PRINT:PRINT:PRINT
225 IF N<3 THEN PRINT"BESTIMMUNG DES MODUS
    NICHT MOEGLICH":GOTO 485
230 FOR I=1 TO N-1:FOR J= I+1 TO N
240 IF X(I)<X(J) THEN 260
250 H=X(I):X(I)=X(J):X(J)=H
260 NEXT J:NEXT I
270 K=2:Y(1)=X(1)
280 FOR I=1 TO N-1
290 IF X(I+1)<>X(I) THEN Y(K)=X(I+1):IFI<>N-1THENK=K+1
300 NEXT I
310 FOR I=1TON:FOR J=1 TO K
320 IF X(I)=Y(J) THEN F(J)=F(J)+1:GOTO 340
330 NEXT J
340 NEXT I
350 FG=0
360 FOR J=1 TO K:IF F(J)>FG THEN FG=F(J):M=J
370 NEXT J
390 PRINT"HAEUFIGSTER WERT (MODUS) = ";Y(M);:PRINT"[]"
400 PRINT"      (HAEUFIGKEIT = ";FG;" )"
410 PRINT:PRINT:PRINT
420 PRINT"ACHTUNG : FALLS DIE VERTEILUNG MEHRERE"
430 PRINT"      MODI AUFWEIST, WIRD NUR DER"
440 PRINT"      ERSTE MODALWERT AUSGEGEBEN !"
450 PRINT:PRINT:B$="      DIE VERTEILUNG IST "
460 IF Y(M)=AM THEN PRINTB$;"SYMMETRISCH"
470 IF Y(M)<AM THEN PRINTB$;"LINKSSTEIL"
480 IF Y(M)>AM THEN PRINTB$;"RECHTSSTEIL"
485 PRINT:PRINT:PRINT
490 PRINT:PRINTTAB(12)"ENDE DER AUSGABE":END
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN![]"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

AM	=	Arithmetisches Mittel
A\$	=	Tastatureingabe
F()	=	Häufigkeiten
FG	=	Größte Häufigkeit
H	=	Hilfsfeld beim Tausch
I	=	Laufindex (Werte)
J	=	Laufindex (verschiedene Werte)
K	=	Klassenanzahl
M	=	Markiergröße
N	=	Anzahl der Werte im Datenbestand
S	=	Summe der Werte
X()	=	Werte im Datenbestand
Y()	=	verschiedene Werte im Datenbestand

o Programmbeschreibung

Satz 10-120 :

Überschrift und Erläuterungen

Satz 130-140 :

Anforderung der Zahl der Werte und Dimensionieren der benötigten Arrays

Satz 150-180 :

Schleife zum Anfordern der Werte und zu ihrer Aufsummierung

Satz 190-210 :

Berechnung des arithmetischen Mittels und Ausgabe des Ergebnisses

Satz 230-260 :

Sortieren des Datenbestands

Satz 270 :

Vorbereitung der Bestimmung derjenigen Werte, die voneinander verschieden sind

Satz 280-300 :

Bestimmung der voneinander verschiedenen Werte und Speicherung derselben im Y-Array

Satz 310-340 :

Bestimmung der Häufigkeitsverteilung

Satz 350-370 :

Bestimmung der größten aufgetretenen Häufigkeit

Satz 380-440 :

Ausgabe des Häufigsten Wertes (Modus)

Satz 450-480 :

Ausgabe einer Information über die Schiefe der erteilung durch Vergleich des arithmetischen Mittels mit dem Modus

Satz 490 :

Beendigung des Hauptprogramms

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Ergebnisse

Wenn wir dem Programm beispielsweise mitteilen, daß 10 Werte einzugeben sind und auf die einzelnen Anforderungen dann die Werte aus dem Kinderzahlbeispiel eingeben, dann erhalten wir als Ergebnis :

ARITHMETISCHES MITTEL = 1.6

HAEUFIGSTER WERT (MODUS) = 1  
(HAEUFIGKEIT = 4)

ACHTUNG : FALLS DIE VERTEILUNG MEHRERE  
MODI AUFWEIST, WIRD NUR DER  
ERSTE MODALWERT AUSGEGEBEN !

DIE VERTEILUNG IST LINKSSTEIL

ENDE DER AUSGABE

o Ergänzungen

Zunächst ist in diesem Zusammenhang darauf hinzuweisen, daß die Daten auch anders als über INPUT-Anweisungen eingegeben werden können. Dies wurde in früheren Kapiteln ausführlich besprochen.

Insbesondere erinnern wir daran, daß der auszuwertende Datenbestand beispielsweise auch von einer Diskette gelesen werden kann, wenn er von früheren Auswertungen her schon existiert.

Manchmal steht man auch vor dem Problem, nicht einzelne Merkmalswerte vorzugeben, wovon wir bisher in diesem Beispiel ausgegangen sind, sondern es könnte sein, daß man eine fertige Häufigkeitsverteilung vorliegen hat (also nicht mehr die Urliste der Daten), für die man arithmetisches Mittel und Modus bestimmen will.

Diese veränderte Ausgangslage soll mit folgendem Beispiel illustriert werden, dem wir eine Kurzfassung des entsprechenden Programms anfügen :

Gegeben sei die folgende Häufigkeitsverteilung :

Kinderzahl	Häufigkeit
0	73
1	150
2	85
3	25
4	12
5	7
6	2
Summe	354

Wenn wir das obige Programm benutzen wollten, um das arithmetische Mittel und den Modus dieser Verteilung auszurechnen, müßten wir also 354 Werte eingeben. Dies ist natürlich sehr umständlich.

Einfacher wäre es, nur die verschiedenen Werte 0 bis 6 und die zugeordneten Häufigkeiten einzugeben. Entsprechend wäre bei klassifiziertem Material vorzugehen (stetige Häufigkeitsverteilung), wo dann nur die Klassenmittelpunkte und die Besetzungszahlen der Klassen einzugeben wären.

Eine derartige Vorgehensweise erfordert ein Programm wie das folgende, das wir als "Kurzversion" vorstellen, also ohne Überschriften, Erläuterungen und dergl.

```

10 PRINT CHR$(147)
20 INPUT "WIEVIELE KLASSEN : ";K
30 DIM X(K),F(K)
40 FOR I=1 TO K:PRINT I;" .KLASSE :":PRINT
50 INPUT "WERT UND HAEUFIGKEIT : ";X(I),F(I)
60 S = S + X(I)*F(I):N = N + F(I):NEXT I
70 PRINT CHR$(147)
80 PRINT "ARITHMETISCHES MITTEL = ";S/N
90 FOR I=1 TO K
100 IF F(I)>FG THEN FG = F(I):M = I
110 NEXT I
120 PRINT "MODUS = ";X(M)
130 END

```

Man sieht, daß man hier zu einer sehr einfachen Programmstruktur gelangen kann.

### 8.2.2 Streuung

#### o Vorstellung des Problems

Das folgende Programm soll für einen gegebenen Datenbestand die Spannweite und die Standardabweichung bestimmen.

#### o Problemanalyse

Die Spannweite bestimmen wir durch Aufsuchen des kleinsten und des größten Wertes, wofür in Abschnitt 6.3.1 die notwendigen problemanalytischen Überlegungen schon vorgestellt wurden.

Die Standardabweichung ergibt sich gemäß der For-

mel (und der Problembeschreibung) in Abschnitt 8.1.3 durch Programmierung einer Schleife, in der die quadrierten Abweichungen der Merkmalswerte vom arithmetischen Mittel aufsummiert werden.

Dies setzt voraus, daß zunächst das arithmetische Mittel bestimmt wird, wobei wir so vorgehen können, wie im vorangegangenen Abschnitt besprochen.

o Programm

```
10 REM S82-STREUUNG
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S82"
70 PRINT TAB(15)"STREUUNG":PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT DIE SPANNWEITE"
100 PRINT"UND DIE STANDARDABWEICHUNG AUS BELIEBIG"
110 PRINT"VIELEN EINZUGEBENDEN WERTEN."
120 PRINT:PRINT:PRINT
130 INPUT"WIEVIELE WERTE (UND RETURN)";N
135 IF N<2 THEN PRINT:PRINT"FEHLER":PRINT:GOTO 120
140 DIM X(N):PRINT:PRINT
150 FOR I=1 TO N
160 PRINTI;:INPUT ".WERT (RETURN) ";X(I)
170 S=S+X(I)
180 NEXT I
190 AM=S/N
200 PRINT CHR$(147)
210 PRINT"ARITHMETISCHES MITTEL      = ";AM;" "
```

```

220 REM SPANNWEITE
230 XK=10000: XG=-10000
240 FOR I=1 TO N
250 IF X(I)<XK THEN XK=X(I)
260 IF X(I)>XG THEN XG=X(I)
270 NEXT I
280 PRINT:PRINT:PRINT
290 PRINT"KLEINSTER WERT = ";XK
300 PRINT"GROESSTER WERT = ";XG
310 PRINT:PRINT"■SPANNWEITE = ";XG-XK;"□"
320 PRINT:PRINT:PRINT:S=0
330 REM STANDARDABWEICHUNG
340 FOR I=1 TO N:Q=X(I)-AM:Q=Q*Q
350 S=S+Q:NEXT I
360 SA=SQR(S/N)
370 PRINT"■STANDARDABWEICHUNG = ";SA;"□"
380 PRINT:PRINT:PRINT
390 PRINTTAB(12)"ENDE DER AUSGABE":END
1000 REM UP WARTEN
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(6)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

AM	=	Arithmetisches Mittel
A\$	=	Tastatureingabe
I	=	Laufindex
N	=	Anzahl der Werte
Q	=	Quadrierte Abweichung eines Wertes vom arithmetischen Mittel
S	=	Summe der quadrierten Abweichungen
SA	=	Standardabweichung
X()	=	Werte im Datenbestand
XG	=	Größter Wert
XK	=	Kleinster Wert

o Programmbeschreibung

Satz 10-210 :

Überschrift, Erläuterungen, Anforderung der Zahl der Werte, Dimensionierung, Eingabe der einzelnen Werte, Berechnung des arithmetischen Mittels und Ausgabe des Mittelwertes (vergl. auch Programmbeschreibung des vorangegangenen Programms)

Satz 220-270 :

Bestimmung des kleinsten und des größten Wertes

Satz 280-310 :

Ausgabe der Extremwerte und der Spannweite

Satz 320-350 :

Errechnung der Summe der quadrierten Abweichungen  
zwischen Merkmalswerten und arithmetischem Mittel

Satz 360-390 :

Berechnung der Standardabweichung, Ausgabe des Er-  
gebnisses und Beendigung des Hauptprogramms

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

o Ergebnisse

Geben wir beispielsweise an, daß fünf Werte einzugeben sind und geben wir nach Anforderung durch das Programm die Werte

1, 3, 2, 1, 3

ein, so erhalten wir als Ergebnisse :

ARITHMETISCHES MITTEL = 2

KLEINSTER WERT = 1

GROESSTER WERT = 3

SPANNWEITE = 2

STANDARDABWEICHUNG = 0.89...

ENDE DER AUSGABE

o Ergänzungen

Zur Ergänzung ist hier nicht mehr anzumerken, als dies schon im vorangegangenen Abschnitt geschehen ist.

Insbesondere weisen wir wieder darauf hin, daß eine vereinfachte Dateneingabe möglich ist, wenn schon klassifiziertes Material und nicht mehr die Urliste der Daten vorliegt.

Dann muß allerdings auch wieder das Programm nach dem Muster der Änderung im vorangegangenen Abschnitt verändert werden, was hier sicherlich nicht vorgeführt zu werden braucht.

## Kapitel 9 : Regressions- und Korrelationsrechnung

### 9.1 Statistische Grundlagen

Im vorangegangenen Kapitel haben wir eine Reihe statistischer Maßzahlen vorgestellt, die verwendet werden können, um univariate Häufigkeitsverteilungen zu charakterisieren. Sie werden also dann verwendet, wenn nur eine statistische Untersuchungsvariable betrachtet wird.

In den Kapiteln über tabellarische und graphische Darstellungen von Häufigkeitsverteilungen haben wir aber schon gesehen, daß interessante Aufgaben gerade dann bearbeitet werden können, wenn zwei Variablen gleichzeitig zum Gegenstand der Untersuchung gemacht werden.

Wir erhalten dann bivariate Verteilungen, die in tabellarischer Form Kreuztabellen und in graphischer Form Streudiagramme genannt werden.

Von solchen Datenpräsentationen ausgehend kann man zwei weiterführende Fragen von großer praktischer Bedeutung stellen :

1. Wie kann die Art des statistischen Zusammenhangs zwischen zwei Variablen beschrieben werden ?
2. Wie groß ist die Stärke des Zusammenhangs zwischen zwei statistischen Variablen ?

Auf die erste Frage antwortet die Regressionsrechnung, auf die zweite die Korrelationsrechnung.

### 9.1.1 Regressionsrechnung

Die Regressionsrechnung hat die Aufgabe, die Art eines eventuellen statistischen Zusammenhangs zwischen zwei Variablen zu beschreiben.

Man geht dabei so vor, daß man in die gegebene "Punktwolke" (Streudiagramm) eine Linie hineinzulegen versucht, die sich den Punkten möglichst gut anpassen soll.

Diese Linie, die linear oder auch nichtlinear sein kann (je nach den Erfordernissen des Datenbestands), wird Regressionsfunktion genannt. Diese Funktion stellt einen zusammenfassenden mathematischen Ausdruck zur Beschreibung der bivariaten Verteilung dar.

Wir werden uns im folgenden ausschließlich auf lineare Regressionsfunktionen beschränken, also auf Funktionen von der Form :

$$\hat{y}_i = a + b * x_i$$

Dabei sind :

$x_i$  = alternative Werte der unabhängigen Variablen (waagrechte Achse)

$\hat{y}_i$  = auf der Regressionsgeraden liegende Y-Werte (senkrechte Achse); d.h. Y-Werte, die bei gegebenen X-Werten zu erwarten wären, wenn die Regressionsgerade den bivariaten Zusammenhang korrekt beschreiben würde

$a$  = Ordinatenabschnitt der Regressionsgeraden

b = Steigung der Regressionsgeraden (Tangens des Steigungswinkels)

Die Regressionsrechnung steht nun vor der Aufgabe, die Geradenparameter a und b so zu bestimmen, daß die sich ergebende Gerade "optimal" in den Punkten des Streudiagramms liegt.

Dazu verhilft die sog. Methode der kleinsten Quadrate :

Sie verlangt, daß die Summe der senkrechten quadrierten Abweichungen zwischen tatsächlich beobachteten Y-Werten und den auf der Geraden liegenden Y'-Werten minimiert wird, also :

$$\sum (y_i - y'_i)^2 = \min$$

Daraus ergibt sich :

$$\sum (y_i - a - b \cdot x_i)^2 = \min$$

Das Minimum dieser Funktion erhält man, indem man die obige Beziehung partiell nach a und nach b differenziert und die beiden ersten Ableitungen null setzt. Man erhält auf diese Weise zwei Bestimmungsgleichungen mit den beiden Unbekannten a und b, die man nach diesen Größen auflösen kann.

Dabei ergibt sich :

$$b = \frac{n \cdot \sum x_i y_i - \sum x_i \cdot \sum y_i}{n \cdot \sum x_i^2 - (\sum x_i)^2}$$

$$a = \frac{\sum y_i}{n} - b \cdot \frac{\sum x_i}{n}$$

Wenn diese beiden Werte also aus einem gegebenen Datenbestand ausgerechnet worden sind, dann kann die Regressionsgerade in das Streudiagramm eingezeichnet werden.

Man hat dann nicht nur - wie oben ausgeführt wurde - einen zusammenfassenden Ausdruck für die Punktvolke, sondern man kann diese Beziehung beispielsweise auch für prognostische Aufgaben einsetzen :

Für jeden beliebigen X - Wert kann man mit Hilfe dieser Beziehung ausrechnen, welcher Y - Wert zu erwarten wäre - dies ist im Methoden-Sinn eine Prognose - wenn die Regressionslinie den wahren Zusammenhang zwischen beiden Variablen mit hinreichender Güte repräsentiert.

Betrachten wir zur Illustration dieser vielleicht etwas komplizierten Ausführungen ein einfaches Beispiel, in dem die Punktvolke nur aus 4 Wertepaaren besteht (dies ist für die Praxis natürlich viel zu wenig, aber was für 4 Wertepaare gilt, gilt im Prinzip auch für 40 oder für 7000).

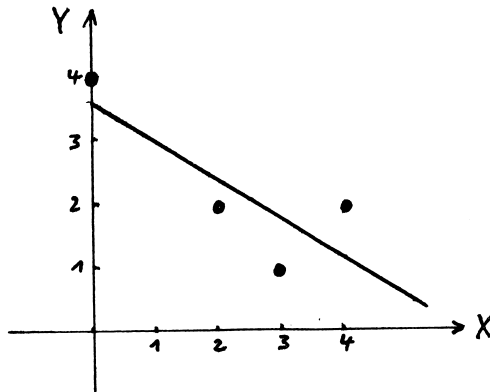
Betrachtet werden die folgenden beiden Variablen:

X = Zahl der Fehler im Deutsch-Diktat  
Y = Zahl der Fehler in einer Mathematikarbeit

Als Daten seien gegeben :

Schüler	Deutsch X	Mathematik Y
A	4	2
B	0	4
C	2	2
D	3	1

Diese Daten stellen sich graphisch folgendermaßen dar :



Die Parameter der Regressionsgeraden  $a$  und  $b$  kann man nun gemäß der oben angegebenen Formeln in einer kleinen Arbeitstabelle leicht bestimmen:

X	Y	X*Y	X
4	2	8	16
0	4	0	0
2	2	4	4
3	1	3	9
9	9	15	29

Somit erhält man :

$$b = (4*15-9*9)/(4*29-81) = - 0.6$$

$$a = 9/4 + 0.6*(9/4) = 3.6$$

Die Regressionsgerade hat also die folgende Form:

$$\hat{y} = 3.6 - 0.6 * x$$

Diese Gerade haben wir in obiger Skizze in die Punktwolke eingezeichnet.

Mit dieser Geraden können nun Prognosen durchgeführt werden : Will ich beispielsweise wissen, wieviele Fehler ein Schüler in Mathematik erwarten läßt, der in der Deutsch-Arbeit 1 Fehler macht, so ergibt sich bei Unterstellung der Regressionsgeraden als gültige Beschreibung des bivariaten Zusammenhangs :

$$\hat{y} = 3.6 - 0.6 * 1 = 3.6 - 0.6 = 3$$

Die zu erwartende Fehlerzahl in der Mathematik-Arbeit liegt also bei 3.

## 9.1.2 Korrelationsrechnung

Im Gegensatz zur Regressionsrechnung, die über die Art eines statistischen Zusammenhangs eine Aussage macht, wird mit der Korrelationsrechnung die Stärke des Zusammenhangs gemessen.

Hat man Daten, die mindestens die Qualität von Intervallskalen aufweisen (so wie in dem eben verwendeten Beispiel), eignet sich dazu ganz besonders gut der Korrelationskoeffizient von Bravais/Pearson.

Er ergibt sich, indem man die sog. Covarianz, die zwischen zwei Untersuchungsvariablen bemessen werden kann, auf das geometrische Mittel aus den beiden Einzelvarianzen bezieht, also auf die Quadratwurzel aus dem Produkt dieser beiden Varianzen.

Die Varianz einer Variablen ist nichts anders als das Quadrat der bekannten Standardabweichung, also:

$$\text{VAR}(X) = \frac{1}{n} \sum (x_i - \bar{x})^2$$

$$\text{VAR}(Y) = \frac{1}{n} \sum (y_i - \bar{y})^2$$

Die Covarianz hingegen sieht folgendermaßen aus :

$$\text{COV}(XY) = \frac{1}{n} \sum (x_i - \bar{x}) * (y_i - \bar{y})$$

Somit erhalten wir als Korrelationskoeffizient :

$$R = \frac{\sum (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 * \sum (y_i - \bar{y})^2}}$$

oder umgeformt :

$$R = \frac{n \cdot \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{[n \cdot \sum x_i^2 - (\sum x_i)^2] \cdot [n \cdot \sum y_i^2 - (\sum y_i)^2]}}$$

Dieser Koeffizient R kann Werte annehmen zwischen +1 und -1 :

Liegt er bei +1, so sprechen wir von vollständiger (maximal starker) positiver (gleichgerichteter) Korrelation, d.h. es ist sehr wahrscheinlich, daß mit zunehmenden X-Werten auch die Y-Werte anwachsen.

Liegt er hingegen bei -1, sprechen wir von vollständiger negativer (gegenläufiger) Korrelation, d.h. es ist sehr wahrscheinlich, daß mit zunehmenden X-Werten die Y-Werte fallen.

Liegt er bei 0, so liegt kein statistischer Zusammenhang vor.

Entsprechend sind alle numerischen Zwischenwerte dieses Koeffizienten R zu interpretieren.

Interessant ist in diesem Zusammenhang das Quadrat von R, der sog. Determinationskoeffizient .

Er gibt den Anteil der Streuung der abhängigen Variablen Y an, der durch die Variation der Variablen X, bei Unterstellung der Gültigkeit einer linearen Beziehung zwischen beiden Variablen, bestimmt wird.

In dem kleinen Zahlenbeispiel von oben ergibt sich - wiederum auf der Grundlage einer Arbeitstabelle,

die wir hier aber nicht präsentieren wollen - als Korrelationskoeffizient :

$$R = -0.814 \quad (R = 0.663)$$

Wir erhalten also einen relativ starken, aber gegenläufigen Zusammenhang zwischen X und Y.

Der Determinationskoeffizient besagt, daß 66.3 % der Streuung der Y-Werte durch die Variation der X-Werte statistisch erklärt wird.

Es muß in diesem Zusammenhang betont werden, daß wir von "statistischer" Erklärung sprechen : Wenn wir feststellen, daß die Variable Y mit der Variablen X variiert und zu einem Teil von dieser erklärt wird, dann heißt dies nicht, daß Y variiert, weil X variiert.

Wir beschreiben mit diesen Methoden nur einen Zusammenhang, können ihn aber nicht etwa begründen, legen mit diesen Methoden also keine Kausalerklärung vor. Dies kann die Statistik auch prinzipiell nicht leisten.

Wenn eine der Variablen (oder beide zusammen) eine geringere Skalenqualität aufweisen, kann das Zusammenhangsmaß R nicht verwendet werden. Es bietet sich dann der sog. Kontingenzkoeffizient (C) an, der auf Überlegungen beruht, die an einem Beispiel, das wir schon kennen, erläutert werden sollen :

Untersucht werden die beiden Variablen Geschlecht und bevorzugte politische Partei. Beide Variablen weisen nur Nominalskalenqualität auf und es soll die Stärke des Zusammenhangs zwischen beiden, ausgehend von dem folgenden Datenbestand, bemessen werden :

	männlich	weiblich	Summe
CDU	20	22	42
SPD	25	20	45
FDP	5	6	11
Grüne	7	7	14
Sonstige	3	2	5
Summe	60	57	117

Wie eine solche Tabelle zu lesen ist, haben wir an anderer Stelle schon dargestellt :

Befragt wurden hier 117 Personen (N), von denen 60 männlichen und 57 weiblichen Geschlechts waren; 42 von ihnen waren CDU-Wähler, 45 SPD-Wähler usw. Weiterhin erkennen wir, daß 20 männliche CDU-Wähler erfaßt wurden, 22 weibliche CDU-Wähler usw.

Wenn nun zwischen den beiden Variablen keine Abhängigkeit bestünde, dann müßten sich beispielsweise die insgesamt 42 CDU-Wähler im Verhältnis 60:57 auf die beiden Geschlechter aufteilen. Das gleiche Verhältnis müßte dann natürlich auch für die Wähler der anderen Parteien gelten.

Mit einer einfachen Dreisatzrechnung können also diejenigen Häufigkeiten in der bivariaten Tabelle bestimmt werden, die zu erwarten wären, wenn Unabhängigkeit bestünde. Diese zu erwartenden Häufigkeiten (e) haben wir in der folgenden Tabelle zusammengestellt :

	männlich	weiblich
CDU	21.54	20.46
SPD	23.08	21.92
FDP	5.64	5.36
Grüne	7.18	6.82
Sonstige	2.56	2.44

Je weiter diese Werte  $e_i$  von den tatsächlich beobachteten Häufigkeiten  $b_i$  der ersten Tabelle abweichen, desto größer ist die Abweichung von der Unabhängigkeit zwischen beiden Variablen, d.h. desto stärker ist die Abhängigkeit, der Zusammenhang zwischen ihnen.

Deshalb werden diese Abweichungen in der folgenden Weise zu einer Summe relativierter quadratischer Abweichungen (S) zusammengefaßt :

$$s = \sum \frac{(b_i - e_i)^2}{e_i}$$

Mit Hilfe dieser Größe S wird der Kontingenz - koeffizient C wie folgt definiert :

$$c = \sqrt{\frac{S}{n+1}}$$

Dieser Koeffizient C bemißt - wie der Koeffizient R - die Stärke des Zusammenhangs zwischen zwei Variablen, die jetzt aber nicht mehr, wie bei R verlangt war, mindestens Intervallskalengualität aufweisen müssen.

In unserem Beispiel ergibt sich für S :

$$S = 0.868$$

und für C :

$$C = 0.086$$

Wir erhalten hier also einen sehr schwachen Zusammenhang zwischen den Variablen Geschlecht und bevorzugte politische Partei.

## 9.2 Programme

Für die oben vorgestellten Aufgaben aus den ansich sehr umfangreichen Bereichen der Regressions- und der Korrelationsrechnung sollen nun die entsprechenden BASIC-Programme entwickelt werden.

### 9.2.1 Lineare Regression

#### o Vorstellung des Problems

Für eine bivariate Häufigkeitsverteilung, die sich graphisch als Streudiagramm darstellt (Punktwolke) soll die lineare Regressionsfunktion (man nennt sie manchmal auch Ausgleichslinie) bestimmt und in das Streudiagramm eingezeichnet werden.

Als Ausgangsdaten verwenden wir die gleichen Angaben zu Körpergröße und Körpergewicht 50 zufällig

ausgewählter Personen, die wir schon in Abschnitt 7.2.4 verwendet haben.

o Problemanalyse

Unter problemanalytischen Gesichtspunkten sind die folgenden Arbeitsschritte zu durchlaufen :

1. Einlesen aller Daten
2. Bestimmung der Parameter der Regressionsgeraden gemäß der Formeln aus Abschnitt 9.1.1
3. Erstellung des Streudiagramms (vergl. dazu Abschnitt 7.2.4, wo sich eine detaillierte Erläuterung zu dieser Aufgabenstellung findet)
4. Einzeichnen der Regressionsgeraden in das Streudiagramm

Im Zusammenhang mit diesem letzten Arbeitsschritt erinnern wir daran, daß bei der Erstellung des Streudiagramms die X- und Y-Werte transformiert wurden, um die Bildschirmgröße ausnutzen zu können.

Bevor nun die Gerade gezeichnet werden kann, muß man deren Parameter (Ordinatenabschnitt und Steigung) aus den so transformierten Werten berechnen.

Aus diesem Grund wird das Programm zur Berechnung der Parameter als Unterprogramm konzipiert, das sowohl im Arbeitsschritt 2 wie auch im vierten Schritt verwendet wird.

Mit Hilfe der bekannten Geradengleichung können dann für alternative X-Werte die zugehörigen Y-Werte bestimmt werden und die Gerade kann dann in die Punktwolke eingezeichnet werden.

o Programm

```
10 REM S91-REGRESSION
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S91"
70 PRINTTAB(14)"REGRESSION"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT EINE REGRES-"
100 PRINT"SIONSGERADE UND STELLT SIE GRAPHISCH"
110 PRINTTAB(5)"MIT DEM STREUDIAGRAMM DAR."
130 PRINT:PRINT"ALS AUSGANGSDATEN VERWENDEN WIR 50 AN-"
140 PRINT"GABEN ZU GROESSE UND GEWICHT ZUFAELLIG"
150 PRINT"AUSGEWAELTER PERSONEN (SIEHE S74)."
```

```

220 PRINT"KONTROLLAUSGABE":PRINT
230 FOR I=1 TO N:PRINTI,X(I),Y(I)
240 IF I/15=INT(I/15) THEN GOSUB 1000
250 NEXT I:GOSUB 1000:GOSUB 3000
255 PRINTCHR$(147):PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
260 XK=200: XG=0: YK=200: YG=0
270 FOR I=1 TO N
280 IF X(I)<XK THEN XK=X(I)
290 IF X(I)>XG THEN XG=X(I)
300 IF Y(I)<YK THEN YK=Y(I)
310 IF Y(I)>YG THEN YG=Y(I)
320 NEXT I
330 FOR I=1TON
340 X(I)=(300/(XG-XK))*(X(I)-XK)+10
350 Y(I)=(140/(YG-YK))*(Y(I)-YK)+10
360 Y(I)=150-Y(I)
370 NEXT I:H=1:GOSUB 3000
380 REM ZEICHNEN
390 GOSUB 1900:REM SCREEN
400 FOR X=3 TO 315:Y=160
410 GOSUB 2000:REM PUNKT
420 NEXT X
430 FOR Y=3 TO 160:X=3
440 GOSUB 2000:REM PUNKT
450 NEXT Y
460 FOR I=1 TO N
470 X=X(I):Y=Y(I):GOSUB 2000:REM PUNKT
480 NEXT I
490 FOR X=3 TO 310
500 Y=0+M*X:GOSUB 2000:REM PUNKT
510 NEXT X
520 FOR I=1TO9999:NEXT I
530 GOSUB 2100:REM TEXT
540 PRINTCHR$(147)
550 PRINTTAB(8)"ENDE DER AUSGABE":END

```

```
850 DATA 165,170,190,192,182,168,168,175,172,194
860 DATA 172,175,177,180,190,170,192,195,190,183
870 DATA 180,190,181,178,175,180,183,165,162,164
880 DATA 184,170,170,168,171,179,180,180,175,165
890 DATA 163,165,162,178,175,180,185,176,190,165
900 DATA 65,77,88,99,85,62,68,74,76,96
910 DATA 77,75,76,78,91,69,86,86,93,94
920 DATA 78,88,85,83,82,80,90,70,68,69
930 DATA 94,72,73,74,75,76,77,83,82,81
940 DATA 72,75,78,81,82,84,78,74,91,69
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !■"
1030 GET A$: IF A#="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
```

```

2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2↑(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN
3000 REM UP REGRESSION *****
3010 S1=0:S2=0:S3=0:S4=0:FORI=1TON
3020 S1=S1+X(I)*Y(I):S2=S2+X(I)
3030 S3=S3+Y(I):S4=S4+X(I)*X(I)
3040 NEXT I
3050 ZA=N*S1-S2*S3
3060 NE=N*S4-S2*S2
3070 M=ZA/NE
3080 O=S3/N-M*S2/N
3090 IF H=1 THEN 3150
3100 PRINTCHR$(147)
3110 PRINT"PARAMETER DER REGRESSIONSGERADEN :";PRINT
3120 PRINT:PRINT"ACHSENABSCHNITT      : ";O
3130 PRINT:PRINT"STEIGUNG              : ";M
3140 GOSUB 1000:REM WARTEN
3150 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
EX	=	Hilfsgröße im Unterprogramm PUNKT
H	=	Hilfsgröße zur Steuerung der Ausgabe der berechneten Regressionsparameter
I	=	Laufindex
M	=	Steigung der Regressionsgeraden
N	=	Anzahl der Wertepaare
NE	=	Nenner
O	=	Ordinatenabschnitt der Regressionsgeraden
S	=	Speicheradresse
S1	=	Summe aller Produkte von X- und Y-Werten
S2	=	Summe aller X-Werte
S3	=	Summe aller Y-Werte
S4	=	Summe aller quadrierten X-Werte
V	=	Speicheradresse (Videochip)
X	=	Bildschirmkoordinate (Punkt)
X()	=	X-Werte
XG	=	Größter X-Wert
XK	=	Kleinster X-Wert und Hilfsgröße im Unterprogramm PUNKT
Y	=	Bildschirmkoordinate (Punkt)

Y() = Werte  
YG = Größter Y-Wert  
YK = Kleinster Y-Wert und Hilfsgröße im Unterprogramm PUNKT  
ZA = Zähler

o Programmbeschreibung

Satz 10-150 :

Überschrift und Erläuterungen

Satz 160 :

Sprung ins Unterprogramm 1000 zum Abwarten, bis die Leertaste betätigt wird

Satz 170 :

Ausgabe einer erläuternden Meldung

Satz 180-210 :

Vorgabe der Anzahl der Wertepaare (N; Satz 180), Dimensionierungen und Einlesen der Werte (entsprechend der Anordnungen in den DATA - Statements zuerst alle X-, dann alle Y-Werte)

Satz 220-250 :

Ausgabe aller Werte mit Unterbrechungen nach je 15 Wertepaaren; Abwarten (Satz 250) und Sprung in das Unterprogramm 3000 zur Berechnung der Parameter der Regressionsgeraden (ebenfalls Satz 250)

Satz 260-320 :

Bestimmung der kleinsten und der größten X- und Y-Werte (dieser Arbeitsschritt wurde schon in einem früheren Programm beschrieben)

Satz 330-370 :

Normierung aller X- und Y-Werte, so daß der Bildschirm gut ausgenutzt wird (auch darüber wurde in einer früheren Programmbeschreibung schon gesprochen);

es ist dabei zu beachten, daß die Bildschirmzeilen von oben nach unten durchnummeriert sind, weshalb in Satz 360 die Y-Werte entsprechend verändert werden;

in Satz 370 wird zusätzlich eine Hilfsvariable H mit 1 belegt, um den Ausdruck der Parameter der Regressionsgeraden, die aus den so transformierten Werten berechnet werden, und die nur für die Zeichnung der Geraden wichtig sind, unterdrücken zu können (vergl. Satz 3090);

weiterhin erfolgt der Sprung in das Unterprogramm 3000 zur Bestimmung der Regressionsparameter für die transformierten X- und Y-Werte

Satz 380-390 :

Vorbereitung des Bildschirms für die hochauflösende Graphik (Sprung ins Unterprogramm 1900)

Satz 400-450 :

Zeichnen eines Achsenkreuzes (Punkt für Punkt, indem jeweils in das Unterprogramm 2000 gesprungen wird)

Satz 460-480 :

Einzeichnen der einzelnen Punkte (Wertepaare) ins Streudiagramm

Satz 490-510 :

Zeichnen der Regressionsgeraden unter Nutzung der aus der Schulmathematik bekannten Geradengleichung und der im Unterprogramm 3000 berechneten Parameter

Satz 520-550 :

Warteschleife, Umschalten in den Textmodus durch Sprung ins Unterprogramm 2100, Schlußmeldung und Beendigung des Hauptprogramms

Satz 850-890 :

X-Werte

Satz 900-940 :

Y-Werte

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 1900-1950 :

Unterprogramm zum Umschalten in den hochauflösenden Graphikmodus

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in der hochauflösenden Graphik

Satz 2100-2120 :

Unterprogramm zum Umschalten in den Textmodus

Satz 3000 :

Beginn des Unterprogramms zur Berechnung der Parameter der Regressionsgeraden mit der Methode der

kleinsten Quadrate

Satz 3010 :

Belegung der vier benötigten Summenfelder mit 0 (da dieses Unterprogramm zweimal benötigt wird, ist diese Nullsetzung erforderlich, um beim zweiten Durchlauf nicht noch die Summenwerte des ersten mitzuschleppen);

weiterhin wird in diesem Satz die Programmschleife zur Erzeugung der einzelnen Summen begonnen

Satz 3020-3040 :

Bildung der vier Summen, die in der Berechnung der Steigung der Regressionsgeraden benötigt werden

Satz 3050-3060 :

Berechnung des Zählers und des Nenners in der Formel für die Steigung

Satz 3070-3080 :

Berechnung der Steigung  $M$  und des Ordinatenabschnitts  $O$  (in der Formel  $b$  und  $a$  genannt) der Regressionsgeraden

Satz 3090 :

Sprung zum Satz 3150, wenn die Hilfsgröße  $H$  den Wert 1 aufweist, d.h. Überspringen der Ergebnisausgabe, wenn die Parameter aus den transformierten Werten berechnet wurden

Satz 3100-3140 :

Löschen des Bildschirms und Ausgabe der Parameter (allerdings nur dann, wenn diese aus den ursprünglichen  $X$ - und  $Y$ -Werten berechnet wurden; vergleiche Satz 370 und 3090) und Abwarten durch Sprung ins Unterprogramm 1000

Satz 3150 :

Rücksprung ins rufende Programm

o Programmergebnisse

Dieses Programm liefert zunächst eine Kontrollausgabe der Wertepaare, zeichnet danach ein Achsenkreuz, zeichnet in diesem Achsenkreuz die Punktwolke und schließlich auch die Regressionsgerade.

Als Rechenergebnisse erhalten wir :

ACHSENABSCHNITT : -54.202613

STEIGUNG : .756774595

Es ist in diesem Zusammenhang zu beachten, daß der Achsenabschnitt beim X-Wert  $X=0$  abzutragen ist, wenn man per Hand zeichnen wollte.

Danach wird die Zeichnung angefertigt. Sie entspricht der Skizze in Abschnitt 9.1.2

o Ergänzungen

Es ist hier daran zu erinnern, daß das gleiche Programm natürlich auch für andere bivariate Datenbestände verwendet werden kann.

Man muß dann natürlich die DATA-Anweisungen verändern und gegebenenfalls die Variable N in Satz 180 neu belegen.

## 9.2.2 Korrelationskoeffizient

### o Vorstellung des Problems

Es soll ein BASIC-Programm vorgestellt werden, das aus einer gegebenen bivariaten Häufigkeitsverteilung den Korrelationskoeffizienten von Bravais/Pearson berechnet (vergl. Abschnitt 9.1.3).

Als Spieldaten verwenden wir wieder den gleichen Datenbestand wie im vorangegangenen Beispiel (50 Angaben zu Körpergröße und Körpergewicht zufällig ausgewählter Personen).

### o Problemanalyse

Allein schon aus der Ähnlichkeit der Berechnungsformel für  $r$  (siehe Abschnitt 9.1.3) mit derjenigen für die Steigung einer Regressionsgeraden  $b$  (siehe Abschnitt 9.1.2 und Programm im Abschnitt 9.2.1) ergibt sich, daß hier keine neuen problemanalytischen Fragestellungen zu verzeichnen sind.

Darüberhinaus vereinfacht sich diese Problemstellung dadurch, daß hier keine Zeichnung angefertigt wird.

o Programm

```
10 REM S92-KORRELATION
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S92"
70 PRINTTAB(14)"KORRELATION:"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT EINEN KORRELA-"
100 PRINTTAB(10)"TIONSKOEFFIZIENTEN."
130 PRINT:PRINT"ALS AUSGANGSDATEN VERWENDEN WIR 50 AN-"
140 PRINT"GABEN ZU GROESSE UND GEWICHT ZUFAELLIG"
150 PRINT"AUSGEWAELTER PERSONEN (SIEHE S74,S91)."
```

```

270 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
280 FOR I=1 TO N
290 S1=S1+X(I)*Y(I):S2=S2+X(I)
300 S3=S3+Y(I):S4=S4+X(I)*X(I)
310 S5=S5+Y(I)*Y(I)
320 NEXT I
330 H1=N*S1-S2*S3
340 H2=N*S4-S2*S2
350 H3=N*S5-S3*S3
360 R=H1/SQR(H2*H3)
370 PRINT:PRINT:PRINT CHR$(147)
380 PRINT"KORRELATIONSKOEFFIZIENT R = ";R;" "
390 PRINT:PRINT:PRINT
400 PRINTTAB(8)"ENDE DER BERECHNUNGEN":END
850 DATA 165,170,190,192,182,168,168,175,172,194
860 DATA 172,175,177,180,190,170,192,195,190,183
870 DATA 180,190,181,178,175,180,183,165,162,164
880 DATA 184,170,170,168,171,179,180,180,175,165
890 DATA 163,165,162,178,175,180,185,176,190,165
900 DATA 65,77,88,99,85,62,68,74,76,96
910 DATA 77,75,76,78,91,69,86,86,93,94
920 DATA 78,88,85,83,82,80,90,70,68,69
930 DATA 94,72,73,74,75,76,77,83,82,81
940 DATA 72,75,78,81,82,84,78,74,91,69
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"BITTE DIE LEERTASTE DRUECKEN !"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
H1-H3	=	Hilfsgrößen in der Berechnung von R
I	=	Laufindex
N	=	Anzahl der Wertepaare
R	=	Korrelationskoeffizient
S1	=	Summe aller Produkte aus X- und Y-Werten
S2	=	Summe aller X-Werte
S3	=	Summe aller Y-Werte
S4	=	Summe aller quadrierten X-Werte
S5	=	Summe aller quadrierten Y-Werte
X()	=	X-Werte
Y()	=	Y-Werte

o Programmbeschreibung

Satz 10-170 :

Überschrift, Erläuterungen und Abwarten

Satz 180-210 :

Vorgabe der Zahl der Wertepaare N, Dimensionierungen und Einlesen aller Werte (zuerst alle X-, dann

alle Y-Werte entsprechend der Anordnung der DATA-Anweisungen)

Satz 220-260 :

Kontrollausgabe der Wertepaare mit Unterbrechung nach jedem 15. Paar und Abwarten durch Sprung ins Unterprogramm 1000

Satz 270-320 :

Bildung der fünf in der Berechnungsformel für R benötigten Summen

Satz 330-360 :

Berechnung des Korrelationskoeffizienten R

Satz 370-400 :

Ausgabe des Ergebnisses und Beendigung des Hauptprogramms

Satz 850-890 :

X-Werte

Satz 900-940 :

Y-Werte

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o **Programmergebnisse**

Aus dem gegebenen Spieldatenbestand berechnet das Programm :

KORRELATIONSKOEFFIZIENT  $R = .829040348$

o **Ergänzungen**

In diesem Arbeitsschritt können wir auf die Anmerkungen des vorangegangenen Abschnitts verweisen. Es ergibt sich hier nichts prinzipiell Neues.

9.2.3 Kontingenzkoeffizient

o **Vorstellung des Problems**

Wenn die einer bivariaten Statistik zugrundeliegenden Daten nicht mindestens Intervallskalengüte aufweisen, kann zur Bemessung der Stärke des statistischen Zusammenhangs der Korrelationskoeffizient  $R$  nicht verwendet werden.

Stattdessen kann mit dem Kontingenzkoeffizienten  $C$  gearbeitet werden, den wir in Abschnitt 9.1.3 vorgestellt haben.

Es soll also nun ein Programm entwickelt werden, welches für einen Spieldatenbestand (30 Angaben zu Geschlecht und bevorzugter politischer Partei zufällig ausgewählter Personen) diesen Koeffizienten berechnet.

o Problemanalyse

Die problemanalytischen Überlegungen müssen daran ansetzen, daß zunächst die bivariaten Häufigkeiten bestimmt werden müssen, also die Felderbesetzungen der bivariaten Häufigkeitstabelle.

Da der Kontingenzkoeffizient C zur Berechnung die (bei Unabhängigkeit der beiden Variablen voneinander) zu erwartenden theoretischen Häufigkeiten benötigt, müssen zunächst die Randsummen der bivariaten Tabelle erzeugt werden, weil diese theoretischen Häufigkeiten sich für ein beliebiges Feld in der Tabelle folgendermaßen bestimmen :

Zu erwartende Häufigkeit =

Zeilensumme \* Spaltensumme / Gesamtanzahl

Sind diese Randsummen bekannt, können also die zu erwartenden Häufigkeiten bestimmt werden und dann der Koeffizient C gemäß der in Abschnitt 9.1.3 zu findenden Formel.

o Programm

```
10 REM S93-KONTINGENZ
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S93"
70 PRINT:PRINT TAB(8)"KONTINGENZKOEFFIZIENT C:"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT DEN KONTIN-"
100 PRINT"GENZKOEFFIZIENTEN C (ZUSAMMENHANGSMASS"
110 PRINT"FUER NOMINALSKALIERTE DATEN).":PRINT
120 PRINT"ALS SPIELDATEN HABEN WIR IN DATA 900 FF"
130 PRINT"ANGABEN ZUM GESCHLECHT UND ZUR BEVOR-"
140 PRINT"ZUGTEN POLITISCHEN PARTEI VON 30 ZU-"
150 PRINT"FAELLIG AUSGEWAELHTEN PERSONEN VORGEGE-"
160 PRINTTAB(16)"BEN."
170 GOSUB 1000:REM WARTEN
180 PRINT"ES IST FOLGENDERMASSEN CODIERT WORDEN:"
190 PRINT:PRINT"GESCHLECHT : ":PRINT
200 PRINT"      MAENNlich = 1"
210 PRINT"      WEIBlich  = 2":PRINT:PRINT
220 PRINT:PRINT"BEVORZUGTE PARTEI : ":PRINT
230 PRINT"      CDU       = 1"
240 PRINT"      SPD       = 2"
250 PRINT"      FDP       = 3"
260 PRINT"      GRUENE    = 4"
270 PRINT"      SONSTIGE  = 5"
280 GOSUB 1000:REM WARTEN
```

```

290 N=30:V1=5:V2=2
300 DIM X(N),Y(N),H(V1,V2),P*(V1),V*(V2)
310 PRINTTAB(7)"MOMENT BITTE, ICH LESE"
320 FOR I=1 TO N:READ X(I),Y(I):NEXT I
330 FORP=1TOV1:READP*(P):NEXTP
335 FOR G=1 TO V2:READ V*(G):NEXT G
340 PRINTCHR*(147):PRINT"≡KONTROLLAUSGABE≡":PRINT
350 PRINT"FALL","GESCHLECHT","PARTEI":PRINT
360 FOR I=1 TO N
370 PRINTTAB(1)I;TAB(14)X(I);TAB(32)Y(I)
380 IF I/10=INT(I/10)THEN GOSUB 1000
390 NEXT I
400 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
410 FOR I=1 TO N
420 FOR P=1 TO V1
430 FOR G=1 TO V2
440 IF X(I)=G AND Y(I)=P THEN H(P,G)=H(P,G)+1:GOTO 470
450 NEXT G
460 NEXT P
470 NEXT I
480 PRINTCHR*(147):PRINT"≡KREUZTABELLE≡":PRINT:PRINT
490 PRINTTAB(5);:FORG=1 TO V2:PRINT" ";V*(G);:NEXTG:PRINT
500 FOR P=1 TO V1
510 PRINTP*(P),:FORG=1TOV2:PRINTH(P,G),:NEXTG
520 PRINT:NEXT P
530 GOSUB 1000:REM WARTEN

```

```

540 REM C
550 DIM S(V1+V2),T(V1,V2)
560 FOR P=1 TO V1:FOR G=1 TO V2
570 S(P)=S(P)+H(P,G):NEXTG:NEXTP
580 FOR G=1TO V2:FOR P=1 TO V1
590 S(V1+G)=S(V1+G)+H(P,G):NEXT P:NEXT G
600 PRINTCHR$(147)
610 PRINT"■KREUZTABELLE MIT RANDSUMMEN :□":PRINT
620 PRINTTAB(5);:FORG=1TOV2:PRINT " ";V$(G);:NEXTG:PRINT"
SUMME"
630 PRINT:FOR P=1 TO V1
640 PRINTP$(P),:FOR G=1 TO V2:PRINHT(P,G),:NEXT G
650 PRINTS(P):NEXTP
660 PRINT:PRINT"SUMME",
670 FOR G=1 TO V2:PRINTS(V1+G),:NEXTG
680 PRINTN:PRINT:PRINT
690 FOR P=1 TO V1:FOR G=1 TO V2
700 T(P,G)=(S(P)*S(V1+G))/N
710 NEXT G:NEXT P
720 S=0:FOR P=1TOV1:FOR G=1TOV2
730 D=H(P,G)-T(P,G):D=D*D:D=D/T(P,G)
740 S=S+D:NEXT G:NEXT P
750 C=SQR(S/(S+N))
760 PRINT:PRINT:PRINT
770 PRINT"■KONTINGENZKOEFFIZIENT C = ";C;"□"
780 PRINT:PRINT:PRINT
790 PRINTTAB(8)"ENDE DER BERECHNUNGEN":END
900 DATA 1,1,2,1,1,3,2,4,1,5,1,2,2,2,2,2,1,1,1,2
910 DATA 1,3,2,5,2,4,1,4,2,3,2,2,1,2,1,1,2,1,1,2
920 DATA 1,1,2,1,2,2,1,3,2,3,2,2,1,2,1,5,2,4,1,4
930 DATA CDU,SPD,FDP,GRUENE,SONSTIGE
940 DATA MAENNLICH,WEIBLICH
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
C	=	Kontingenzkoeffizient C
D	=	Hilfsgröße in der Berechnung der relativen Quadratsummen
G	=	Laufindex der Variablen V2 (Geschlecht)
H(,)	=	Bivariate Häufigkeiten
I	=	Laufindex (Fälle)
N	=	Anzahl der Fälle (Wertepaare)
P	=	Laufindex der Variablen V1 (Partei)
P\$( )	=	Bezeichnungen der Variablen V1 (Partei)
S	=	Summe
S( )	=	Randsummen der bivariaten Tabelle
T(,)	=	Zu erwartende Häufigkeiten
V1	=	Anzahl der Kategorien der Variablen 1
V2	=	Anzahl der Kategorien der Variablen 2
V\$( )	=	Bezeichnungen der Variablen 2
X( )	=	Werte der ersten Variablen
Y( )	=	Werte der zweiten Variablen

o **Programmbeschreibung**

**Satz 10-280 :**

**Überschrift und Erläuterungen zum Programm**

**Satz 290 :**

**Vorgabe der Anzahl der Fälle (N), der Ausprägungen der ersten Variablen (V1) und der zweiten Variablen (V2)**

**Satz 300-335 :**

**Dimensionierungen und Einlesen aller Wertepaare und aller Bezeichnungen**

**Satz 340-390 :**

**Kontrollausgabe mit Unterbrechungen nach je zehn Wertepaaren**

**Satz 400-470 :**

**Bestimmung der bivariaten Häufigkeiten**

**Satz 480-530 :**

**Ausgabe der erzeugten Kreuztabelle**

**Satz 540-550 :**

**Kommentar und Dimensionierung der Randsummen und der zu erwartenden Häufigkeiten**

**Satz 560-570 :**

**Bestimmung der Zeilensummen**

Satz 580-590 :

Bestimmung der Spaltensummen

Satz 600-680 :

Ausgabe der Kreuztabelle mit Randsummen

Satz 690-710 :

Berechnung der bei Unabhängigkeit der beiden Variablen voneinander zu erwartenden Häufigkeiten

Satz 720-740 :

Berechnung der Summe quadrierter, relativierter Abweichungen zwischen beobachteten und zu erwartenden Häufigkeiten

Satz 750 :

Berechnung des Kontingenzkoeffizienten C

Satz 760-790 :

Ausgabe des Ergebnisses und Beendigung des Hauptprogramms

Satz 900-920 :

Paarweise Angabe der X- und der Y-Werte

Satz 930 :

Bezeichnungen der einen Variablen

Satz 940 :

Bezeichnungen der anderen Variablen

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

o Programmergebnisse

Bei dem gegebenen Spieldatenbestand erhalten wir zunächst die folgende Kreuztabelle :

KREUZTABELLE

	MAENNLICH	WEIBLICH
CDU	4	3
SPD	5	5
FDP	3	2
GRUENE	2	3
SONSTIGE	2	1

Danach wird die Kreuztabelle erneut, nun aber mit den Randsummen zusammen ausgegeben und schließlich der interessierende Wert des Kontingenzkoeffizienten C :

KONTINGENZKOEFFIZIENT C = .155784543

o Ergänzungen

Durch Veränderung der Datenbestände (Satz 900 ff) und der Belegung der Variablen N (Satz 290) mit einem anderen Wert kann man auch für andere bivariate Verteilungen den Kontingenzkoeffizienten C berechnen.

Man muß allerdings bei der Ausgabe der Kreuztabellen darauf achten, ob die Bildschirmgröße zur Tabellendarstellung ausreicht. Gegebenenfalls sind die entsprechenden PRINT-Anweisungen zu verändern oder es muß dafür Sorge getragen werden, daß die Tabellen stückweise ausgegeben werden, was durch geeignete Gestaltung der Ausgabeschleifen realisiert werden kann.

Wenn man letzten Endes nur am Wert des Kontingenzkoeffizienten selbst interessiert ist, kann man diese eventuellen Probleme natürlich umgehen, wenn man einfach auf die Ausgabe der beiden Kreuztabellen verzichtet.

## Kapitel 10 : Zeitreihenstatistik

### 10.1 Statistische Grundlagen

In den vorangegangenen Kapiteln wurden ausschließlich solche Fragestellungen behandelt, die mit der statistischen Auswertung von Häufigkeitsverteilungen zu tun hatten.

Dies ist aber nur einer von mehreren Anwendungsbereichen statistischer Methoden. Ein weiterer für die Praxis nicht unwichtiger Bereich ist der der Zeitreihenstatistik.

In diesem Kapitel wird deshalb für diesen Bereich ein Überblick über einige der wesentlichen Auswertungsmethoden geboten und es werden die dafür geeigneten BASIC-Programme entwickelt.

#### 10.1.1 Begriff und Aufgaben

Unter einer statistischen Zeitreihe versteht man die Zuordnung der Merkmalswerte einer Variablen zu Zeitpunkten oder Zeiträumen.

Ein typisches Beispiel ist die Betrachtung der Bevölkerungsentwicklung eines Landes über mehrere Jahre hinweg, die Entwicklung der Arbeitslosenquote über die Monate eines Jahres hinweg, tägliche meteorologische Angaben, die Notierung des täglichen Körpergewichts usw.

Es ist klar, welche Zielsetzungen mit der Erstellung solcher Zeitreihen verfolgt werden :

Nur mit derartigen Zeitreihen wird die Kontrolle zeitlicher Verläufe, wie sie z.B. im Unternehmensbereich unentbehrlich ist, möglich; nur so werden für Prognosen und damit für planerische Entscheidungen die Grundlagen geschaffen.

Als Beispiel einer solchen Zeitreihe stellen wir den jährlichen Zuwachs des Pro-Kopf-Einkommens (in nominaler Betrachtung, also unter Einschluß der Preisteigerungen) in der Bundesrepublik Deutschland dar und erhalten dafür die folgenden Daten :

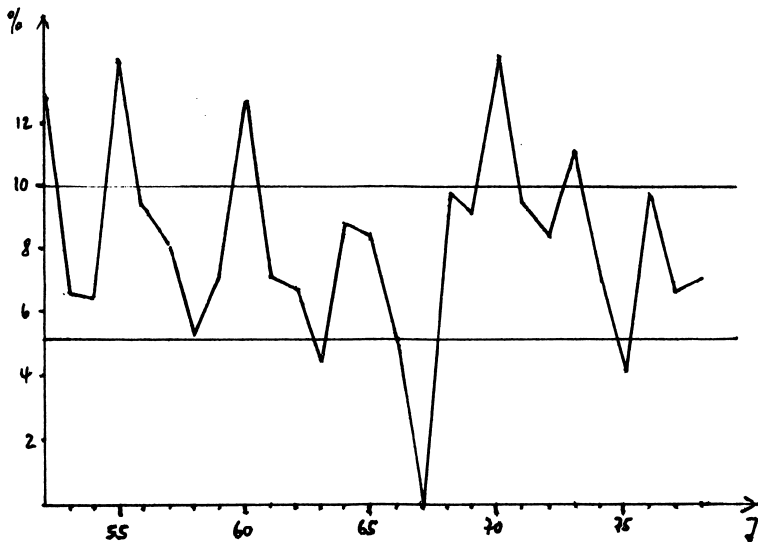
Jahr	Zuwachs(%)	Jahr	Zuwachs(%)
1952	12.7	1966	5.0
1953	6.6	1967	0
1954	6.4	1968	9.8
1955	14.0	1969	9.2
1956	9.7	1970	14.1
1957	8.2	1971	9.7
1958	5.5	1972	8.5
1959	7.2	1973	11.1
1960	12.6	1974	7.0
1961	7.2	1975	4.2
1962	6.9	1976	9.9
1963	4.6	1977	6.7
1964	8.8	1978	7.0
1965	8.3		

Quelle : W.Voß, Die Bundesrepublik Deutschland - Daten und Analysen -, 2.Auflage, Stuttgart u.a., 1981, Seite 83

Diese Angaben eignen sich recht gut dazu, die statistischen Methoden, um die es in diesem Kapitel geht, zu illustrieren, auch wenn diese Daten nicht allzu aktuell sind - aber darauf kommt es in unserem Zusammenhang ja nicht in erster Linie an.

Solche Zeitreihen kann man sehr anschaulich graphisch mit Hilfe eines Achsenkreuzes darstellen, wie die folgende Skizze verdeutlicht :

Daten der obigen Tabelle :



In den folgenden Abschnitten wollen wir uns jetzt anschauen, welches die wichtigsten Aufgaben und Methoden der Zeitreihenstatistik sind.

### 10.1.2 Trendprognosen

Als Prognose bezeichnen wir den Versuch, zu einer begründeten Aussage über zukünftige Entwicklungen zu gelangen.

Es gibt eine große Zahl unterschiedlicher Prognosemethoden, von denen wir hier ausschließlich die sog. Trendprognose betrachten wollen.

Als Trend bezeichnet man die langfristige Entwicklungstendenz einer gegebenen Zeitreihe. Wenn man also beispielsweise bei der täglichen Gewichtskontrolle auf der Badezimmerwaage nach einigen Monaten eine Zeitreihe erhält, die trotz mehrfacher Auf- und Abbewegungen im Prinzip nach oben zeigt, dann spricht man von einem steigenden Trend.

Trendprognosen machen es erforderlich, daß aus der gegebenen Zeitreihe zunächst der Trend bestimmt wird.

Auch dafür gibt es unterschiedliche Berechnungsmethoden, von denen wir zwei Verfahren kurz vorstellen wollen, nämlich

1. die Methode der halben Durchschnitte
2. die Methode der kleinsten Quadrate

Bei der Methode der halben Durchschnitte geht man so vor, daß die gegebene Zeitreihe in zwei Hälften geteilt wird; dann werden sowohl aus der ersten als auch aus der zweiten Hälfte der Zeitreihenwerte die jeweiligen arithmetischen Mittel berechnet.

Der erste Mittelwert wird dem Zeitpunkt zugeordnet, der in der Mitte der ersten Reihenhälfte zu finden ist, der zweite der Mitte der zweiten Reihenhälfte.

Wenn man in einer graphischen Darstellung diese beiden Punkte dann miteinander durch eine gerade Linie verbindet, so erhält man eine die Trendentwicklung ungefähr beschreibende Gerade, mit der dann weitergearbeitet werden kann.

Bei der Methode der kleinsten Quadrate geht man so vor, wie wir es in dem Kapitel über die Regressionsrechnung schon beschrieben haben :

Man kann sich die Zeitreihe ja gewissermaßen als Punktwolke im Koordinatensystem vorstellen, in die - im einfachsten Fall - eine lineare Funktion hin-

ingelegt werden soll - jetzt keine Regressionsfunktion, sondern eine Trendfunktion.

An den grundlegenden Überlegungen ändert dieser inhaltliche Unterschied aber nichts, so daß es also auch hier darum geht, Steigung und Ordinatenabschnitt der Funktion zu bestimmen. Wir können dabei die gleichen Formeln benutzen, wie sie oben schon vorgestellt wurden.

Wenn dann eine Trendlinie bestimmt ist, entweder mit der einen oder mit der anderen Methode, dann kann man Prognosen durchführen, indem man in die Trendgleichung

$$YT = a + b * X$$

einfach an der Stelle X den gewünschten Prognosezeitwert einsetzt.

In dieser Geradengleichung bedeuten :

YT = Trendwerte (Prognosewerte)

a = Ordinatenabschnitt

b = Tangens des Steigungswinkels

X = Zeitwerte

Ab und zu muß man bei diesen Überlegungen allerdings in Rechnung stellen, daß die Trendentwicklung nicht linear ist, mit einer linearen Trendfunktion deshalb nicht besonders gut beschrieben werden kann; logischerweise sind auch dann die mit der Trendgeraden durchgeführten Prognosen nicht allzu treffsicher.

Mit der Methode der kleinsten Quadrate können auch nichtlineare Trendfunktionen bestimmt werden, was aber wegen des wesentlich erhöhten mathematischen Aufwands hier nicht besprochen werden soll.

Ersatzweise kann man eine offenkundig nichtlineare zeitliche Entwicklung in Unterabschnitte in der Weise einteilen, daß innerhalb dieser Abschnitte dann wieder mit linearen Teilfunktionen gearbeitet werden kann, was uns den Rückgriff auf das jetzt besprochene Instrumentarium erlauben würde.

### 10.1.3 Glättungen

Mit dem Problem der Trendberechnung eng verwandt sind Aufgaben der Glättung von Zeitreihen.

Hier geht es darum, die eventuellen Ausschläge der Zeitreihe, die häufig mehr oder weniger regelmäßig nach oben und nach unten abwechselnd auftreten, auszugleichen.

Eine dafür besonders gut geeignete Methode ist das Verfahren der gleitenden Durchschnitte :

Hat man beispielsweise Monatsangaben über mehrere Jahre hinweg, so geht man bei dieser Methode gemäß den folgenden Arbeitsschritten vor :

1. Addition der ersten 12 Werte;
2. Bildung des arithmetischen Mittels, indem diese Summe durch 12 dividiert wird;
3. Zuordnung dieses Mittelwerts zur Mitte des angegebenen Zeitraums (also zwischen dem 6. und dem 7. Monat);
4. Subtraktion des ersten Werts von der Summe aus Schritt 1 und Addition des 13. Werts;
5. Erneute Mittelwertbildung und Zuordnung dieses Werts zwischen dem 7. und dem 8. Monat;

usw.

Auf diese Weise entsteht eine Glättungslinie, die kurzfristige, z.B. saisonale Schwankungen ausgleichen kann.

Es versteht sich, daß je nach Zeitreihe auch mit anderen "Schnittlängen" gearbeitet werden kann.

Hat man beispielsweise Quartalsangaben, so empfehlen sich gleitende Viererschnitte, hat man Jahresangaben, wird oft mit gleitenden Siebenerschnitten gearbeitet, weil man weiß, daß sich konjunkturelle Schwankungen häufig in Siebenjahreszyklen niederschlagen u.ä.

## 10.2 Programme

### 10.2.1 Vorbemerkung

In den folgenden Programmen zur Auswertung statistischer Zeitreihen gemäß den im vorangegangenen Abschnitt vorgetragenen Überlegungen, werden immer jeweils drei Aufgaben gleichzeitig erledigt :

1. Tabellarische Darstellung der Zeitreihe
2. Trendberechnung und Prognosen oder Glättung
3. Graphische Darstellung der Zeitreihe, des Trends oder der Glättungslinie in hochauflösender Graphik

Wir benötigen dafür keine neuen Programmelemente, weil alle diese Teilaufgaben auch schon in den vorangegangenen Kapiteln behandelt wurden.

## 10.2.2 Methode der halben Durchschnitte

### o Vorstellung des Problems

Es soll ein BASIC-Programm entwickelt werden, welches für die Zeitreihendaten, die in dem Abschnitt 10.1.1 vorgestellt wurden, die Trendlinie mit der Methode der halben Durchschnitte bestimmt.

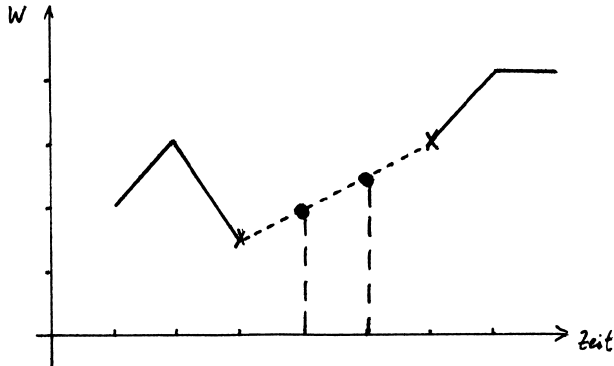
### o Problemanalyse

Nach dem Einlesen der Zeitreihendaten müssen diese zunächst tabellarisch dargestellt werden, was keine besonderen Schwierigkeiten verursacht.

Schwierigkeiten gibt es allerdings dann, wenn für bestimmte Zeitangaben keine Werte vorliegen, wenn also in der Zeitreihe "Lücken" auftreten (was aber bei unseren Beispieldaten nicht der Fall ist).

Für den Fall solcher "Lücken" sehen wir in dem nun folgenden Programm vor, daß der "unechte" Wert -99 einzugeben ist.

Wenn das Programm diesen Markierungswert findet - gegebenenfalls auch mehrere hintereinander - so wird linear interpoliert, wie durch die folgende Skizze verdeutlicht wird :



Diese Interpolation geht also so vonstatten, daß der nicht vorhandene "Lücken"-Wert durch die jeweils benachbarten "echten" Werte durch Mittelwertbildung unter Beachtung der eventuell unterschiedlichen Abstände der "echten" Werte von der "Lücke" geschätzt wird.

Nachdem also auf diese Weise eine eventuell nicht vollständige Zeitreihe komplettiert ist, kann sie gezeichnet werden, wobei - wie schon in früheren Programmen - zur Ausnutzung des Bildschirms zunächst die kleinsten und die größten Werte bestimmt werden und eine Normierung der Ausgangswerte vorgenommen wird.

Das Zeichnen der Zeitreihe geht dann so vor sich, daß gerade Linienstücke von Zeitpunkt zu Zeitpunkt gezeichnet werden, was durch Bestimmung von Geradengleichungen mittels der aus der Schulmathematik bekannten Zwei-Punkte-Formel ermöglicht werden kann (die "zwei Punkte" sind die jeweils benachbarten Zeitreihenwerte).

Mittels eines einfachen Rechenalgorithmus können dann die beiden Mittelwerte der beiden Reihenhälften bestimmt werden und danach kann die Trendlinie

- wiederum mit Hilfe der Zwei-Punkte-Formel - gezeichnet werden.

Schließlich wird dem Benutzer angeboten, Prognosewerte zu bestimmen, wobei einfach auf die Geradengleichung des Trends zurückgegriffen wird.

#### o Programm

```
10 REM S101-TRENDHD
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S101"
70 PRINTTAB(3)"METHODE DER HALBEN DURCHSCHNITTE"
80 GOSUB 1000:REM WARTEN
90 PRINT"DIESES PROGRAMM BESTIMMT EINE TRENDLI-"
100 PRINT"NIE MIT DER METHODE DER HALBEN DURCH-"
110 PRINT"SCHNITTE UND STELLT SIE GRAPHISCH DAR."
120 PRINT:PRINT"DARUEBERHINAUS SIND PROGNOSEN MOEGLICH."
130 PRINT:PRINT"ALS AUSGANGSDATEN WERDEN JAHRESDATEN"
140 PRINT"VERWENDET : ZUWACHS DES VOLKSEINKOMMENS"
150 PRINT"PRO KOPF (NOM.) DER BRD 1952-1978 IN %"
160 GOSUB 1000:REM WARTEN
170 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
180 N=27
190 DIM X(N),Y(N)
200 FOR I=1 TO N:READ Y(I):NEXT I
210 FOR I=1 TO N:READ X(I):NEXT I
```

```

215 PRINTCHR$(147)
220 PRINT"■KONTROLLAUSGABE■":PRINT
230 FOR I=1 TO N:PRINTX(I),Y(I)
240 IF I/15=INT(I/15) THEN GOSUB 1000
250 NEXT I:GOSUB 1000:REM WARTEN
260 FOR I=1 TO N
270 IF Y(I)=-99 THEN GOSUB 3000:REM INTERPOL
280 NEXT I
290 GOSUB 4000:REM UP HALBE SCHNITTE
300 XK=X(1):XG=X(N):YK=20:YG=-10
310 FOR I=1 TO N
320 IF Y(I)<YK THEN YK=Y(I)
330 IF Y(I)>YG THEN YG=Y(I)
340 NEXT I
350 FOR I=1TON
360 X(I)=(200/(XG-XK))*(X(I)-XK)
370 Y(I)=(150/(YG-YK))*(Y(I)-YK)
380 Y(I)=150-Y(I)
390 NEXT I
400 M1=(200/(XG-XK))*(M1-XK):M2=(200/(XG-XK))*(M2-XK)
410 D1=(150/(YG-YK))*(D1-YK):D1=150-D1
420 D2=(150/(YG-YK))*(D2-YK):D2=150-D2
430 REM ZEICHNEN
440 GOSUB 1900:REM SCREEN
450 FOR X=1 TO 300
460 Y=150:GOSUB 2000:REM PUNKT
470 NEXT X
480 FOR I=1 TO N
490 X=X(I):Y=151:GOSUB 2000:REM PUNKT
500 NEXT I
510 FOR Y=1 TO 150
520 X=0:GOSUB 2000:REM PUNKT
530 NEXT Y

```

```

540 FOR I=1 TO N-1
550 X1=X(I):X2=X(I+1)
560 Y1=Y(I):Y2=Y(I+1)
570 FOR X=X1 TO X2
580 Y=((Y2-Y1)*(X-X1))/(X2-X1)+Y1
590 GOSUB 2000:REM PUNKT
600 NEXT X
610 SW=1:IF Y1=Y2 THEN 670
620 IF Y1>Y2 THEN SW=-1
630 FOR Y=Y1 TO Y2 STEP SW
640 X=((Y-Y1)*(X2-X1))/(Y2-Y1)+X1
650 GOSUB 2000:REM PUNKT
660 NEXT Y
670 NEXT I
680 REM TREND
690 FOR X=1 TO 300
700 Y=((D2-D1)*(X-M1))/(M2-M1)+D1
710 GOSUB 2000:REM PUNKT
720 NEXT X
730 FOR I=1 TO 2000:NEXT I
740 GOSUB 2100:REM TEXT
750 PRINTCHR$(147)
760 PRINTTAB(8)"ENDE DER AUSGABE":END
900 DATA 12.7,6.6,6.4,14,9.7,8.2,5.5
910 DATA 7.2,12.6,7.2,6.9,4.6,8.8,8.3
920 DATA 5.0,0,9.8,9.2,14.1,9.7,8.5
930 DATA 11.1,7,4.2,9.9,6.7,7
950 DATA 1952,1953,1954,1955,1956,1957,1958,1959,1960
960 DATA 1961,1962,1963,1964,1965,1966,1967,1968,1969,1970
970 DATA 1971,1972,1973,1974,1975,1976,1977,1978

```

```

1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !□"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2↑(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN
3000 REM UP INTERPOL
3010 FOR J1=I TO I-10 STEP-1
3020 IF J1=1 THEN 3050
3030 IF Y(J1)<>-99 THEN 3050
3040 NEXT J1
3050 FOR J2=I TO I+10
3060 IF J2=N THEN 3090
3070 IF Y(J2)<>-99 THEN 3090
3080 NEXT J2
3090 Y(I)=(Y(J1)*(J2-I)+Y(J2)*(I-J1))/(J2-J1)
3100 RETURN

```

```

4000 REM UP HALBE SCHNITTE *****
4010 S=0:Z=INT(N/2)
4020 FOR I=1 TO Z:S=S+Y(I):NEXT I
4030 D1=S/Z:M1=Z/2+.5+X(1)-1
4040 S=0
4050 FOR I=Z+1 TO N:S=S+Y(I):NEXT I
4060 D2=S/(N-Z):M2=(Z+1+N)/2+X(1)-1
4070 B=(D2-D1)/(M2-M1)
4080 A=(D1*M1-D2*M1)/(M2-M1)+D1
4090 PRINTCHR$(147)
4100 PRINT"PARAMETER DER LINEAREN TRENDFUNKTION :";
4110 PRINT:PRINT" (ACHTUNG : ACHSENSCHNITT IM JAHR 0 !)"
4120 PRINT:PRINT"ACHSENABSCHNITT = ";A
4130 PRINT:PRINT"STEIGUNG = ";B
4140 PRINT:PRINT:PRINT
4150 PRINTTAB(6)"PROGNOSE GEWUENSCHT (J/N)"
4155 GET A$:IF A$="" THEN 4155
4156 IF A$<>"J" AND A$<>"N" THEN 4155
4160 IF A$="N" THEN 4220
4170 PRINT:PRINT:INPUT "WELCHES JAHR (UND RETURN)";J
4180 Y=A+B*J:Y=INT(Y*100+.5)/100
4190 PRINT:PRINT:PRINT
4200 PRINT"JAHR : ";J,Y;" "
4210 GOTO 4140
4220 RETURN

```

o Variablenliste

- A = Ordinatenabschnitt
- A\$ = Tastatureingabe
- B = Steigung
- D1,D2 = Ordinatenwerte der Reihenhälften (halbe Durchschnitte)
- EX = Hilfsgröße im Unterprogramm PUNKT (UP 2000)
- I = Laufindex
- J1,J2 = Laufindizes bei der Interpolation
- M1,M2 = Abszissenwerte der Reihenhälften
- N = Anzahl der Werte
- S = Speicheradresse im Unterprogramm PUNKT (UP 2000) und Summe im Unterprogramm HALBE SCHNITTE (UP 4000)
- SW = Schrittweite
- V = Speicheradresse Videochip
- X = Abszissenwert im Unterprogramm PUNKT (UP 2000)
- X() = Abszissenwerte (Zeitangaben)
- XG = Größter X-Wert
- XK = Kleinster X-Wert und Hilfsgröße im Unterprogramm PUNKT (UP 2000)

X1,X2 = Hilfsgrößen beim Zeichnen von Reihenwert  
zu Reihenwert

Y = Ordinatenwert im Unterprogramm PUNKT (UP  
2000) und Prognosewert im Unterprogramm  
HALBE SCHNITTE (UP 4000)

Y() = Ordinatenwerte (Merkmalswerte der Reihe)

YG = Größter Y-Wert

YK = Kleinster Y-Wert und Hilfsgröße im Unter-  
programm PUNKT (UP 2000)

Y1,Y2 = Hilfsgrößen beim Zeichnen von Reihenwert  
zu Reihenwert

Z = Zahl der Merkmalswerte in der Reihenhäl-  
fte

o Programmbeschreibung

Satz 10-170 :

Überschrift und Erläuterungen

Satz 180-210 :

Vorgabe der Anzahl der Werte, Dimensionierungen  
Einlesen der Werte gemäß ihrer Anordnung in den  
DATA-Anweisungen

Satz 220-250 :

Kontrollausgabe mit Unterbrechung nach je 15 Wer-  
ten

Satz 260-280 :

Sprung ins Unterprogramm 3000 zum Interpolieren,  
wenn ein Reihenwert -99 ist

Satz 290 :

Sprung ins Unterprogramm 4000 zur Berechnung der halben Durchschnitte, der Trendlinie und eventueller Prognosen

Satz 300-340 :

Bestimmung der kleinsten und der größten Werte

Satz 350-390 :

Normierung der Werte zur Ausnutzung des Bildschirms

Satz 400-420 :

Normierung der beiden Reihenhälften - Mittelwerte und der Abszissenpunkte, denen sie zugeordnet werden

Satz 430-440 :

Vorbereitung des Bildschirms für die hochauflösende Graphik (UP 1900)

Satz 450-530 :

Zeichnen eines Achsenkreuzes mit besonderer Markierung der Zeitangaben auf der waagrechten Achse (Satz 480-500)

Satz 540 :

Beginn der Zeichenschleife für die Zeitreihe

Satz 550-560 :

Bestimmung der Anfangs- und Endkoordinaten für ein gerades Linienstück zwischen je zwei Reihewerten

Satz 570-600 :

Zeichnen des jeweiligen geraden Linienstücks nach Zuordnung von Y-Werten zu alternativen X-Werten gemäß der Zwei-Punkte-Formel

Satz 610-660 :

Erneutes Zeichnen des gleichen Linienstücks wie in Satz 570-600, jetzt aber werden alternativen Y-Werten die über die Zwei-Punkte-Formel berechenbaren X-Werte zugeordnet;

dadurch wird erreicht, daß bei steilen Linienstücken (Winkel größer als 45 Grad) keine Leerstellen in der Linie entstehen;

es muß dabei darauf geachtet werden, daß wir der Schrittweite für diese Zeichenschleife den Wert -1 zuweisen müssen, wenn  $Y_1 < Y_2$ , weil dann das Linienstück steigen muß, d.h. die Bildschirmzeilennummer wird immer kleiner

Satz 670 :

Übergang zum nächsten Zeitreihenpunkt

Satz 680-720 :

Zeichnen der Trendlinie, indem gemäß der Zwei-Punkte-Formel jedem X-Wert der entsprechende Wert der Trendfunktion zugeordnet wird (Satz 700)

Satz 730-760 :

Warteschleife, Zurückschalten in den Textmodus und Beendigung des Hauptprogramms

Satz 900-930 :

Daten (Merkmalswerte)

Satz 950-970 :

Daten (Zeitangaben)

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

Satz 1900-1950 :

Unterprogramm zum Umschalten des Bildschirms in  
die hochauflösende Graphik

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in hoch-  
auflösender Graphik

Satz 2100-2120 :

Unterprogramm zum Zurückschalten in den Textmodus

Satz 3000 :

Beginn des Unterprogramms zur linearen Interpolati-  
on fehlender Zeitreihenwerte

Satz 3010 :

Beginn einer Schleife, welche alle Werte vom feh-  
lenden Wert Nr.I beginnend bis zum Wert Nr. I-10  
überprüft;

bei diesem Interpolationsprogramm wird also unter-  
stellt, daß die "Lücke" im Datenbestand höchstens  
20 Werte umfaßt (vergl. auch Satz 3050); damit  
dürften alle praktischen Fälle abgedeckt sein (die  
Prüfung erstreckt sich auf den Bereich "Lücke" +/-  
10 Reihenwerte)

Satz 3020 :

Ist schon der erste Wert ein "fehlender" Wert, so erfolgt ein Sprung zum Satz 3050; es braucht nicht nach "links" geprüft zu werden

Satz 3030 :

Ist der jeweilige Reihenwert ein "echter" Wert, so kann bei Satz 3050 fortgesetzt werden; der Laufindex J1 gibt dann an, wie groß die "Lücke" nach "links" hin war

Satz 3040 :

Überprüfung des nächsten Wertes "links" von der "Lücke"

Satz 3050-3080 :

Die gleichen Überlegungen (vergl. Satz 3010-3040) gelten für den Bereich "rechts" von der "Lücke"

Satz 3090 :

Der zu interpolierende Wert ergibt sich als ein mit den Abständen zwischen dem fehlenden Wert zu den links und rechts anzutreffenden "echten" Werten gewogenes arithmetisches Mittel aus diesen benachbarten "echten" Werten

Satz 3100 :

Rücksprung ins rufende Programm

Satz 4000 :

Beginn des Unterprogramms zur Bestimmung der halben Durchschnitte

Satz 4010 :

Vorbereitung des Summenfeldes S und Bestimmung der

Anzahl der Reihenwerte in der halben Reihe

Satz 4020 :

Berechnung der Summe der Werte der ersten Reihenhälfte

Satz 4030 :

Berechnung des halben Durchschnitts ( $D_1$ ) und Bestimmung des Zeitpunktes, dem dieser Durchschnitt zuzuordnen ist ( $M_1$ ), in Abhängigkeit von  $Z$  und von der ersten Zeitangabe  $X(1)$

Satz 4040-4060 :

Entsprechend gleiche Vorgehensweise wie in Satz 4010-4030 für die zweite Reihenhälfte

Satz 4070-4080 :

Berechnung von Steigung und Ordinatenabschnitt der linearen Trendfunktion, die durch die beiden Punkte  $M_1, D_1$  und  $M_2, D_2$  verläuft (an einer einfachen Skizze kann man sich, falls nötig, die hier verwendeten Beziehungen verdeutlichen)

Satz 4090-4140 :

Ausgabe der Ergebnisse der Berechnungen

Satz 4150-4160 :

Abfrage, ob eine Prognose gewünscht wird;

wenn nein, verzweigt das Programm zum Rücksprung in das rufende Programm (Satz 4220)

Satz 4170 :

Anforderung des gewünschten Prognosezeitpunkts

Satz 4180-4200 :

Berechnung des Prognosewertes und Ausgabe

Satz 4210 :

Rücksprung zum Satz 4140, d.h. Möglichkeit einer erneuten Prognose

Satz 4220 :

Rücksprung ins rufende Programm

o Programmergebnisse

Wenn wir dieses Programm starten, so wird der Ausgangsdatenbestand zunächst tabellarisch ausgegeben.

Danach erhalten wir die Parameter der linearen Trendfunktion :

ACHSENABSCHNITT = 95.4348391

STEIGUNG = -.0444037444

PROGNOSE GEWUENSCHT (J/N) ?

Antworten wir auf diese Frage mit J und geben wir auf Anforderung des Programms z.B. 1986 ein, so erhalten wir :

JAHR : 1986 7.25

Danach erhalten wir die Möglichkeit für weitere Prognosen.

Wenn wir davon keinen Gebrauch machen, dann wird vom Programm die Zeitreihe und die Trendfunktion in einem Achsenkreuz in hochauflösender Graphik gezeichnet.

o Ergänzungen

Der Datenbestand kann natürlich verändert werden, was auch eine Änderung von  $N$  in Satz 180 erfordert.

Allerdings kann die Reihe nicht beliebig lang werden - es sei denn, es wird auf das Zeichnen verzichtet. Wenn aber gezeichnet werden soll, ist es nicht sinnvoll, mehr als 150 Reihenwerte aufzunehmen, weil sonst die einzelnen Zeitpunkte auf dem Bildschirm nicht mehr voneinander getrennt werden können.

Hat man mehr Zeitwerte, empfiehlt es sich, nur die aktuelleren Werte zu verwenden oder z.B. nur jeden zweiten Wert o.ä.

Weiterhin machen wir darauf aufmerksam, daß man nicht unbedingt die Zeitwerte selbst, also wie im gegebenen Beispiel die Jahreszahlen, über DATA-Anweisungen vorgeben muß :

Man kann sie auch rechnerisch im Programm erzeugen oder ganz darauf verzichten und sie einfach nur mit einem Laufindex numerieren (1,2,3,...). Es ist dann in den Leseanweisungen entsprechend darauf zu reagieren.

### 10.2.3 Methode der kleinsten Quadrate

Wir beschäftigen uns in diesem Programm noch einmal mit der gleichen Aufgabenstellung, also mit der Bestimmung einer linearen Trendfunktion, verwenden jetzt aber die Methode der kleinsten Quadrate.

#### o Vorstellung des Problems

Für den gleichen Datenbestand wie im vorangegangenen Programm soll ein Programm erstellt werden, welches die Trendlinie mit der Methode der kleinsten Quadrate erzeugt.

#### o Problemanalyse

Es ergeben sich hier keine neuen Probleme. Lediglich die Trendberechnung selbst (im Unterprogramm 4000) erfolgt jetzt nach anderen Berechnungsformeln, nämlich nach denen, die wir im Kapitel über die Regressionsrechnung schon kennengelernt haben.

#### o Programm

```
10 REM S102-TRENDKQ
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S102"
70 PRINTTAB(4)"METHODE DER KLEINSTEN QUADRATE?"
80 PRINT:PRINT
```

```

90 PRINT"DIESES PROGRAMM BESTIMMT EINE TRENDLI-"
100 PRINT"NIE MIT DER METHODE DER KLEINSTEEN QUA-"
110 PRINT"DRATE UND STELLT SIE GRAPHISCH DAR."
120 PRINT:PRINT"DARUEBERHINAUS SIND PROGNOSEN MOEGLICH."
130 PRINT:PRINT"ALS AUSGANGSDATEN WERDEN JAHRESDATEN"
140 PRINT"VERWENDET : ZUWACHS DES VOLKSEINKOMMENS"
150 PRINT"PRO KOPF (NOM.) DER BRD 1952-1978 IN %"
160 GOSUB 1000:REM WARTEN
170 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
180 N=27
190 DIM X(N),Y(N)
200 FOR I=1 TO N:READ Y(I):NEXT I
210 FOR I=1 TO N:READ X(I):NEXT I
215 PRINTCHR$(147)
220 PRINT"KONTROLLAUSGABE":PRINT
230 FOR I=1 TO N:PRINTX(I),Y(I)
240 IF I/15=INT(I/15) THEN GOSUB 1000
250 NEXT I:GOSUB 1000:REM WARTEN
260 FOR I=1 TO N
270 IF Y(I)=-99 THEN GOSUB 3000:REM INTERPOL
280 NEXT I
290 GOSUB 4000:REM UP TREND
300 XK=X(1):XG=X(N):YK=20:YG=-10
310 FOR I=1 TO N
320 IF Y(I)<YK THEN YK=Y(I)
330 IF Y(I)>YG THEN YG=Y(I)
340 NEXT I
350 FOR I=1TON
360 X(I)=(200/(XG-XK))*(X(I)-XK)
370 Y(I)=(150/(YG-YK))*(Y(I)-YK)
380 Y(I)=150-Y(I)
390 NEXT I

```

```

400 REM ZEICHNEN
410 GOSUB 1900:REM SCREEN
420 FOR X=1 TO 300
430 Y=150:GOSUB 2000:REM PUNKT
440 NEXT X
450 FOR I=1 TO N
460 X=X(I):Y=151:GOSUB 2000:REM PUNKT
470 NEXT I
480 FOR Y=1 TO 150
490 X=0:GOSUB 2000:REM PUNKT
500 NEXT Y
510 FORI=1TON-1
520 X1=X(I):X2=X(I+1)
530 Y1=Y(I):Y2=Y(I+1)
540 FOR X=X1 TO X2
550 Y=((Y2-Y1)*(X-X1))/(X2-X1)+Y1
560 GOSUB 2000:REM PUNKT
570 NEXT X
580 SW=1:IF Y1=Y2 THEN 640
590 IF Y1>Y2 THEN SW=-1
600 FOR Y=Y1 TO Y2 STEP SW
610 X=((Y-Y1)*(X2-X1))/(Y2-Y1)+X1
620 GOSUB 2000:REM PUNKT
630 NEXT Y
640 NEXT I
650 Z=1:GOSUB 4000:REM TREND
660 FOR X=1 TO 300
670 Y=A+B*X
680 GOSUB 2000:REM PUNKT
690 NEXT X
700 FOR I=1 TO 2000:NEXT I
710 GOSUB 2100:REM TEXT
720 PRINTCHR$(147)
730 PRINTTAB(8)"ENDE DER AUSGABE":END
900 DATA 12.7,6.6,6.4,14,9.7,8.2,5.5
910 DATA 7.2,12.6,7.2,6.9,4.6,8.8,8.3
920 DATA 5.0,0,9.8,9.2,14.1,9.7,8.5
930 DATA 11.1,7,4.2,9.9,6.7,7
950 DATA 1952,1953,1954,1955,1956,1957,1958,1959,1960
960 DATA 1961,1962,1963,1964,1965,1966,1967,1968,1969,1970
970 DATA 1971,1972,1973,1974,1975,1976,1977,1978
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !■"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN

```

```

1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FOR I=1024 TO 2023:POKE I,14:NEXT I
1940 FOR I=8192 TO 16383:POKE I,0:NEXT I
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2*(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN
3000 REM UP INTERPOL *****
3010 FOR J1=I TO I-10 STEP-1
3020 IF J1=1 THEN 3050
3030 IF Y(J1)<>-99 THEN 3050
3040 NEXT J1
3050 FOR J2=I TO I+10
3060 IF J2=N THEN 3090
3070 IF Y(J2)<>-99 THEN 3090
3080 NEXT J2
3090 Y(I)=(Y(J1)*(J2-I)+Y(J2)*(I-J1))/(J2-J1)
3100 RETURN

```

```

4000 REM UP TREND *****
4010 S1=0:S2=0:S3=0:S4=0
4020 FOR I=1 TO N
4030 S1=S1+X(I)*Y(I):S2=S2+X(I)
4040 S3=S3+Y(I):S4=S4+X(I)*X(I)
4050 NEXT I
4060 B=(N*S1-S2*S3)/(N*S4-S2*S2)
4070 A=S3/N-B*S2/N
4080 IF Z=1 THEN 4220
4090 PRINTCHR$(147)
4100 PRINT"PARAMETER DER LINEAREN TRENDFUNKTION :";
4110 PRINT:PRINT" (ACHTUNG : ACHSENSCHNITT IM JAHR 0 !)"
4120 PRINT:PRINT"ACHSENABSCHNITT = ";A
4130 PRINT:PRINT"STEIGUNG = ";B
4140 PRINT:PRINT:PRINT
4150 PRINTTAB(5)"PROGNOSE GEWUENSCHT (J/N)"
4155 GET A$:IF A$="" THEN 4155
4156 IF A$<>"J" AND A$<>"N" THEN 4155
4160 IF A$="N" THEN 4220
4170 PRINT:PRINT:INPUT "WELCHES JAHR (UND RETURN)";J
4180 Y=A+B*J:Y=INT(Y*100+.5)/100
4190 PRINT:PRINT:PRINT
4200 PRINT"JAHR : ";J,Y;" "
4210 GOTO 4140
4220 RETURN

```

o Variablenliste

Es kann hier einfach die Variablenliste des vorangegangenen Programms übernommen werden, wobei die folgenden Variablen hier nicht benötigt werden :

D1, D2, M1, M2 und Z

Stattdessen benötigen wir zusätzlich :

S1 = Summe aller Produkte aus X- und Y-Werten

S2 = Summe aller X-Werte

S3 = Summe aller Y-Werte

S4 = Summe aller quadrierten X-Werte

Die Variable Z wird anders verwendet als im vorangegangenen Programm, nämlich als Markiervariable zur Steuerung der Ausgabe der Parameter der Trendgeraden.

o Programmbeschreibung

Satz 10-390 :

Überschrift, Erläuterungen, Dimensionierung, Einlesen der Daten, Kontrollausgabe, Interpolationen, Trendberechnung (Unterprogramm 4000) und Normierung zur Vorbereitung der Bildschirmausgabe wie im vorangegangenen Programm

Satz 400-640 :

Zeichnen der Zeitreihe in hochauflösender Graphik (Details sind der Beschreibung des vorangegangenen Programms, Satz 430-670, zu entnehmen)

Satz 650 :

Sprung ins Unterprogramm 4000 zur Berechnung der Trendlinie aus den normierten X- und Y-Werten; die Markiervariable erhält dabei jetzt den Wert 1 (im Gegensatz zum entsprechenden Sprung in Satz 290), um die Ausgabe zu unterdrücken (vergl. auch Satz 4080)

Satz 660-730 :

Zeichnen der Trendlinie, Warteschleife, Umschalten in den Textmodus und Beendigung des Hauptprogramms

Satz 900-970 :

Daten

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 1900-1950 :

Unterprogramm zum Umschalten in den Graphikmodus

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in hochauflösender Graphik

Satz 2100-2120 :

Unterprogramm zum Umschalten in den Textmodus

Satz 3000-3100 :

Unterprogramm zur linearen Interpolation eventuell fehlender Werte (Details sind beim vorangegangenen Programm beschrieben)

Satz 4000-4010 :

Beginn des Unterprogramms, das die Trendlinie mit der Methode der kleinsten Quadrate bestimmt und Nullsetzen der vier in der Berechnungsformel für die Steigung  $b$  benötigten Summen

Satz 4020-4050 :

Berechnung der vier benötigten Summen

Satz 4060-4070 :

Berechnung der Parameter  $a$  und  $b$  (im Programm A und B genannt)

Satz 4080 :

Wenn die Markiervariable  $Z$  den Wert 1 aufweist, so sind die Parameter aus den normierten Werten berechnet worden; Ergebnisausgabe und Prognosemöglichkeiten werden dann unterdrückt (vergl. Satz 650) und es erfolgt der Rücksprung aus diesem Unterprogramm

Satz 4090-4140 :

Ausgabe der Ergebnisse (Markiervariable  $Z=0$ )

Satz 4150-4210 :

Angebot eventueller Prognosen (vergl. vorangegangenes Programm)

Satz 4220 :

Rücksprung in das rufende Programm

o Ergebnisse

Die Ergebnisse dieses Programms entsprechen denen des vorangegangenen Beispiels mit dem Unterschied, daß nun die Trendlinie unter Umständen eine etwas andere Lage aufweist, weil sie ja mit einer anderen Methode bestimmt wurde.

Nähere Einzelheiten sind hier entbehrlich.

o Ergänzungen

Auch hier verweisen wir auf das vorangegangene Beispiel.

#### 10.2.4 Reihenglättung

o Vorstellung des Problems

Mit Hilfe der Methode der gleitenden Durchschnitte, soll eine Glättungslinie bestimmt werden (man bezeichnet sie übrigens in der Fachliteratur ebenfalls als Trendlinie).

Als Ausgangsdaten wählen wir die monatlichen Anga-

ben zur Arbeitslosenquote in der Bundesrepublik Deutschland von 1978 bis 1983.

o Problemanalyse

Da wir in diesem Beispiel mit Monatsdaten arbeiten, liegt die Vermutung schon nahe, daß sinnvollerweise mit gleitenden Zwölfer-Schnitten zu arbeiten ist.

Allerdings ist es zweckmäßig, wenn die Schnittlänge variabel gehalten wird, um mit diesem Programm auch andere Möglichkeiten der Glättung bieten und ausprobieren zu können.

Man muß dabei beachten, daß dann, wenn die Länge des gleitenden Schnitts geradzahlig ist (z.B. 12), der jeweils berechnete Durchschnitt immer genau zwischen zwei Zeitpunkten einzuordnen ist (zwischen dem 6. und dem 7. Zeitpunkt, zwischen dem 7. und dem 8. Zeitpunkt usw.).

Ist hingegen die Schnittlänge ungeradzahlig (z.B. 7), kann der jeweilige Durchschnitt immer direkt einem Zeitpunkt zugeordnet werden.

Im Programm muß darüberhinaus vorgesehen werden, daß keine unsinnigen Schnittlängen (z.B. 0 oder 1 oder eine Schnittlänge, die größer ist als die Anzahl der Zeitreihenwerte selbst) verwendet werden. Deshalb werden entsprechende Abfragen eingebaut.

Darüberhinaus entstehen bei dieser Aufgabenstellung keine Probleme, wenn man davon absieht, daß nicht nur die Zeitreihe selbst Punkt für Punkt (wie in den vorangegangenen Beispielen auch) gezeichnet werden muß, sondern jetzt auch die Trendlinie (Glättungslinie), weil diese ja keine lineare Funktion, sondern ebenfalls eine vielfach geknickte Linie ist.

In diesem Zusammenhang muß man aber berücksichtigen, daß zwar die Zeitreihe beim ersten Zeitwert beginnt und beim letzten endet, nicht aber die Glättungslinie : Wenn z.B. die Schnittlänge 7 beträgt, beginnt die Glättungslinie erst beim 4. Zeitwert und endet beim viertletzten Zeitwert.

Durch entsprechende Belegung der Felder W1 und W2 werden wir dies im folgenden Programm berücksichtigen.

#### o Programm

```
10 REM S103-GLAETTUNG
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"#BEISPIEL S103"
70 PRINTTAB(15)"GLAETTUNG#"
80 PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT EINE TRENDLI-"
100 PRINT"NIE MIT DER METHODE DER GLEITENDEN"
110 PRINT"SCHNITTE UND STELLT SIE GRAPHISCH DAR."
120 PRINT
130 PRINT"AUSGANGSDATEN SIND DIE MONATLICHEN AR-"
140 PRINT"BEITSLÖSENQUOTEN (%) IN DER BRD 1978-83"
160 GOSUB 1000:REM WARTEN
170 PRINTTAB(7)"MOMENT BITTE, ICH RECHNE"
180 N=72
190 DIM X(N),Y(N),T(N)
200 FOR I=1 TO N:READ Y(I):Y(I)=Y(I)/10:NEXT I
210 FOR I=1 TO N:X(I)=I:NEXT I
220 GOSUB 5000:REM UEBERSCHRIFT
230 FOR I=1 TO N:PRINTX(I),Y(I)
240 IF I/15=INT(I/15) THEN GOSUB 1000:GOSUB 5000
250 NEXT I:GOSUB 1000:REM WARTEN
260 FOR I=1 TO N
270 IF Y(I)=-99 THEN GOSUB 3000:REM INTERPOL
280 NEXT I
290 GOSUB 4000:REM UP TREND
292 GOSUB 1000:REM WARTEN
```

```

293 PRINTCHR$(147)
294 PRINTTAB(8)"BITTE ETWAS GEDULD FUER DIE"
295 PRINTTAB(8)"VORBEREITUNG DER ZEICHNUNG"
296 PRINT:PRINT"ACHTUNG:":PRINT
297 PRINT"DIE ABSZISSE IST NICHT DIE NULLACHSE !"
298 GOSUB 1000:REM WARTEN
300 XK=X(1):XG=X(N):YK=20:YG=-10
310 FOR I=1 TO N
320 IF Y(I)<YK THEN YK=Y(I)
330 IF Y(I)>YG THEN YG=Y(I)
340 NEXT I
350 FOR I=1TON
360 X(I)=(200/(XG-XK))*(X(I)-XK)
370 Y(I)=(150/(YG-YK))*(Y(I)-YK):Y(I)=150-Y(I)
380 T(I)=(150/(YG-YK))*(T(I)-YK):T(I)=150-T(I)
390 NEXT I
400 REM ZEICHNEN
410 GOSUB 1900:REM SCREEN
420 FOR X=1 TO 300
430 Y=150:GOSUB 2000:REM PUNKT
440 NEXT X
450 FOR I=1 TO N
460 X=X(I):Y=151:GOSUB 2000:REM PUNKT
470 NEXT I
480 FOR Y=1 TO 150
490 X=0:GOSUB 2000:REM PUNKT
500 NEXT Y
510 W1=1:W2=N-1:GOSUB 6000:REM ZEICHNEN

```

```

520 REM TREND
530 FOR I=1 TO N:Y(I)=T(I):NEXT I
540 W1=1+INT(L/2):W2=N-INT(L/2)-1
550 GOSUB 6000:REM ZEICHNEN
560 FOR I=1 TO 4000:NEXT I
570 GOSUB 2100:REM TEXT
580 PRINTCHR$(147)
590 PRINT"ENDE DER AUSGABE":END
900 DATA 54,54,49,44,40,39,40,40,38,39
910 DATA 41,44,51,50,42,38,34,33,35,35
920 DATA 37,33,35,38,45,43,38,36,33,34
930 DATA 37,37,35,38,42,48,56,56,52,49
940 DATA 48,48,53,55,54,59,64,73,87,81
950 DATA 76,72,69,68,72,74,75,79,84,91
960 DATA 102,104,98,92,88,87,89,89,86
970 DATA 87,95,99
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN !□"
1030 GET A$:IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
1900 REM UP SCREEN *****
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN
2000 REM UP PUNKT *****
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2↑(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT *****
2110 POKEV+17,155:POKEV+24,21
2120 RETURN

```

```

3000 REM UP INTERPOL *****
3010 FOR J1=I TO I-10 STEP-1
3020 IF J1=1 THEN 3050
3030 IF Y(J1)<>-99 THEN 3050
3040 NEXT J1
3050 FOR J2=I TO I+10
3060 IF J2=N THEN 3090
3070 IF Y(J2)<>-99 THEN 3090
3080 NEXT J2
3090 Y(I)=(Y(J1)*(J2-I)+Y(J2)*(I-J1))/(J2-J1)
3100 RETURN
4000 REM UP TREND *****
4010 PRINTCHR$(147)
4020 INPUT "SCHNITTLAENGE : ";L
4025 IF L<1 OR L<>INT(L) OR L>N/2 THEN PRINT"EINGABEFehler":
PRINT:GOTO 4020
4030 PRINTCHR$(147)
4040 PRINTTAB(5)"MOMENT BITTE, ICH RECHNE"
4050 IF L/2=INT(L/2) THEN 4130
4060 FOR I=1 TO N-L+1
4070 S=0
4080 FOR K=I TO I+L-1:S=S+Y(K):NEXT K
4090 P=I+L/2-.5
4100 T(P)=INT(S/L*100+.5)/100
4110 NEXT I
4120 GOTO 4210
4130 FOR I=1 TO N-L
4140 S1=0:S2=0
4150 FOR K=I TO I+L-1:S1=S1+Y(K)
4160 S2=S2+Y(K+1):NEXT K
4170 P=I+L/2
4180 T(P)=(S1/L+S2/L)/2
4190 T(P)=INT(T(P)*100+.5)/100
4200 NEXT I
4210 REM KONTROLLAUSDRUCK
4220 GOSUB 5000:REM UEBERSCHRIFT
4230 FOR I=1 TO N
4240 IF I/15=INT(I/15) THEN GOSUB 1000:GOSUB 5000
4250 PRINTI,Y(I),T(I)
4260 NEXT I
4270 RETURN

```

```

5000 REM UP UEBERSCHRIFT *****
5010 PRINTCHR$(147)
5020 PRINT"ZEIT","WERT","TREND":PRINT
5030 FOR J=1 TO 30:PRINT"-";:NEXT J:PRINT
5040 RETURN
6000 REM UP ZEICHNEN *****
6010 FORI=W1 TO W2
6020 X1=X(I):X2=X(I+1)
6030 Y1=Y(I):Y2=Y(I+1)
6040 FOR X=X1 TO X2
6050 Y=((Y2-Y1)*(X-X1))/(X2-X1)+Y1
6060 GOSUB 2000:REM PUNKT
6070 NEXT X
6080 SW=1
6090 IF Y1>Y2 THEN SW=-1
6100 IF Y1=Y2 THEN 6150
6110 FOR Y=Y1 TO Y2 STEP SW
6120 X=((Y-Y1)*(X2-X1))/(Y2-Y1)+X1
6130 GOSUB 2000:REM PUNKT
6140 NEXT Y
6150 NEXT I
6160 RETURN

```

o Variablenliste

A\$	=	Tastatureingabe
EX	=	Hilfsgröße im Unterprogramm PUNKT (UP 2000)
I,J	=	Laufindizes
J1,J2	=	Laufindizes im Unterprogramm INTERPOLATION (UP 3000)
K	=	Laufindex über die Schnittlänge
L	=	Schnittlänge
N	=	Anzahl der Werte
P	=	Zeitliche Position des Durchschnitts
S	=	Adresse im Unterprogramm PUNKT (UP 2000) und Summe bei der Schnittberechnung
S1,S2	=	Hilfssummen bei der Durchschnittsberechnung bei geradzahliger Schnittlänge
SW	=	Schrittweite bei der Zeichnung der Reihen
T()	=	Trendwerte (Werte der Glättungslinie)
V	=	Speicheradresse Videochip
W1,W2	=	Anfangs- und Endzeitpunkt der Glättungsreihe
X	=	Koordinatenwert beim Zeichnen (UP 2000)
X()	=	Zeitwerte

XG = Größter X-Wert  
XK = Kleinster X-Wert und Hilfsgröße im Unterprogramm PUNKT (UP 2000)  
X1,X2 = Anfangs- und Endspalte beim Zeichnen eines geraden Linienstücks zwischen zwei Zeitwerten  
Y = Koordinatenwert beim Zeichnen (UP 2000)  
Y() = Merkmalswerte  
YG = Größter Y-Wert  
YK = Kleinster Y-Wert und Hilfsgröße im Unterprogramm PUNKT (UP 2000)  
Y1,Y2 = Anfangs- und Endzeile beim Zeichnen eines geraden Linienstücks zwischen zwei Zeitpunkten

o **Programmbeschreibung**

**Satz 10-190 :**

**Überschrift, Erläuterungen, Vorgabe von N (Anzahl der Werte) und Dimensionierungen**

**Satz 200 :**

**Einlesen der Merkmalswerte und Division durch 10, weil die Werte in den DATA-Anweisungen der Einfachheit halber ohne Dezimalpunkte eingegeben wurden**

**Satz 210 :**

**Durchnumerierung der Zeitwerte**

Satz 220 :

Sprung ins Unterprogramm 5000 zur Ausgabe einer Tabellenüberschrift

Satz 230-250 :

Ausgabe der Tabelle mit Unterbrechung nach je 15 Werten und jeweils erneuter Ausgabe der Tabellenüberschrift (UP 5000)

Satz 260-280 :

Sprung in das Unterprogramm zum Interpolieren (UP 3000), falls dies erforderlich ist, falls also an einer Stelle der Zeitreihe der Markierwert -99 auftritt

Satz 290 :

Sprung ins Unterprogramm 4000 zur Bestimmung der Glättungslinie (bzw. ihrer Werte T(I))

Satz 292-298 :

Zusätzliche Erläuterungen

Satz 300-340 :

Bestimmung der größten und der kleinsten X- und Y-Werte

Satz 350-390 :

Normierung der X-, Y- und T-Werte zur Ausnutzung des Bildschirms

Satz 400-500 :

Vorbereitung des Bildschirms zur hochauflösenden Graphik und Zeichnen eines Achsenkreuzes

Satz 510 :

Sprung ins Unterprogramm 6000 zum Zeichnen der Zeitreihe vom ersten ( $W1=1$ ; 1. bis 2. Zeitpunkt) bis zum letzten Zeitpunkt ( $W2=N-1$ ; vorletzter bis letzter Zeitpunkt)

Satz 520-530 :

Belegung des Y-Arrays mit den Trendwerten  $T(I)$ , um das Unterprogramm 6000 (Zeichnen einer Reihe) erneut benutzen zu können (siehe Satz 550)

Satz 540-550 :

Neubelegung des Anfangswertes  $W1$  und des Endwertes  $W2$  (die Glättungslinie beginnt ja später und endet früher als die ursprüngliche Zeitreihe in direkter Abhängigkeit von  $N$  und  $L$ ) und Sprung ins Unterprogramm 6000 zum Zeichnen

Satz 560-590 :

Warteschleife, Umschalten in den Textmodus und Beendigung des Hauptprogramms

Satz 900-970 :

Daten (jeweils das Zehnfache der eigentlichen Werte, um das Eingeben der Dezimalpunkte zu sparen; vergl. auch Satz 200)

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 1900-1950 :

Unterprogramm zur Vorbereitung des Bildschirms für die hochauflösende Graphik

Satz 2000-2060 :

Unterprogramm zum Zeichnen eines Punktes in hochauflösender Graphik

Satz 2100-2120 :

Unterprogramm zum Umschalten in den Textmodus

Satz 3000-3100 :

Unterprogramm zur linearen Interpolation fehlender Werte (siehe zur detaillierten Erläuterung dieses Unterprogramms das vorletzte Programm in Abschnitt 10.2.2)

Satz 4000-4040 :

Beginn des Unterprogramms zur Bestimmung gleitender Durchschnitte mit der Anforderung der Schnittlänge  $L$  und einer Wartemeldung

Satz 4050 :

Wenn  $L$  geradzahlig ist, erfolgt ein Sprung zum Satz 4130, andernfalls geht es bei Satz 4060 weiter

Satz 4060 :

Beginn einer Schleife, die alle gleitenden Schnitte bildet (sie muß bei 1 beginnen und bei  $N-L+1$  enden)

Satz 4070-4080 :

Bestimmung der jeweiligen gleitenden Summe durch Addition aller Werte zwischen  $I$  und  $I+L-1$

Satz 4090 :

Bestimmung des Zeitpunktes, dem der jeweilige gleitende Schnitt zuzuordnen ist

Satz 4100 :

Berechnung des Durchschnitts und Runden auf zwei Dezimalstellen

Satz 4110 :

Übergang zum nächsten gleitenden Schnitt

Satz 4120 :

Sprung zum Satz 4210, d.h. es wird derjenige Programmbereich übersprungen, der für die geradzahli- gen Schnitte benötigt wird

Satz 4130-4160 :

Bei geradzahligen Schnitten werden zwei Hilfssummen benötigt, aus denen ein gemeinsamer Mittelwert errechnet wird, der dann wieder genau einem Zeitpunkt zugeordnet werden kann (die beiden Einzelschnitte, die sich aus diesen Hilfssummen ergeben, würden jeweils zwischen zwei Zeitpunkten stehen)

Satz 4170-4190 :

Bestimmung des Trendwertes und der zeitlichen Position, der er zuzuordnen ist

Satz 4200 :

Übergang zum nächsten gleitenden Schnitt

Satz 4210-4260 :

Ausgabe der Ursprungswerte und der Trendwerte mit Unterbrechungen nach je 15 Wertepaaren und Wiederholung der Tabellenüberschrift

Satz 4270 :

Rücksprung ins rufende Hauptprogramm

Satz 5000-5040 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 6000-6160 :

Unterprogramm zum Zeichnen einer Zeitreihe; die ausführliche Erörterung dieses Programmteils findet sich in Abschnitt 10.2.2

o Ergebnisse

Dieses Programm liefert zunächst eine tabellarische Darstellung der Ausgangsdaten und danach wird die gewünschte Schnittlänge angefordert.

Geben wir daraufhin z.B. den Wert 12 ein, so erhalten wir :

ZEIT	WERT	TREND
1	5.4	0
2	5.4	0
3	4.9	0
4	4.4	0
5	4	0
6	3.9	0
7	4	4.34
8	4	4.31
9	3.8	4.26
10	3.9	4.21
11	4.1	4.16
12	4.4	4.11
usw.		

Wir sehen hier sehr deutlich, daß die gleitenden

Schnitte erst beim Zeitpunkt 7 beginnen, was bei einem Zwölferschnitt, wie er hier gewählt wurde, auch nicht anders zu erwarten ist.

Danach fertigt das Programm eine Zeichnung in der hochauflösenden Graphik an, die in einem Achsenkreuz die Ursprungsreihe und die Glättungsreihe gemeinsam darstellt.

o Ergänzungen

Im Prinzip gelten die gleichen Überlegungen, wie sie schon bei dem ersten Trendberechnungsprogramm (Abschnitt 10.2.2) vorgetragen wurden.

## Kapitel 11 : Hochrechnungen

### 11.1 Vorbemerkung

Wir haben in den vorangegangenen Kapiteln nur für solche statistischen Fragestellungen BASIC - Programme entwickelt, die man zum Bereich der sogenannten deskriptiven Statistik zählt. Es handelte sich also um Verfahren, die der Beschreibung statistischer Sachverhalte oder Entwicklungen dienen.

Wie aber schon erwähnt wurde, leistet die Statistik mehr als nur Beschreibungen - so wichtig diese auch sind.

Insbesondere im Bereich der sog. Wahrscheinlichkeitsstatistik werden Verfahren bereitgestellt, für die es eine Vielzahl interessanter Anwendungsmöglichkeiten gibt.

Wir wollen uns darauf beschränken, ein einziges dieser Verfahren - allerdings eines der wichtigsten aus diesem Bereich - in die Betrachtung mit aufzunehmen.

Es handelt sich um die sog. Hochrechnungen, die der Statistiker Intervallschätzungen nennt.

Diese Verfahren sind von erheblicher Praxisrelevanz. Man erinnere sich nur einmal daran, daß an Wahlsonntagen sehr rasch sehr präzise Wahlergebnisse auf der Basis solcher Hochrechnungen vorgelegt werden können.

## 11.2 Statistische Grundlagen

Von einer Hochrechnung spricht man dann, wenn auf der Grundlage einer Teilauswertung schon Gesamtergebnisse vorgelegt werden können.

Konkret geht das so vor sich, daß nicht die komplette Gesamtheit aller Merkmalsträger untersucht und ausgewertet wird, die sog. Grundgesamtheit, sondern nur eine Teilmenge. Man spricht dann von einer Stichprobe.

Es leuchtet unmittelbar ein, daß Stichprobenerhebungen viel billiger sind und schneller vonstatten gehen als Totalerhebungen, so daß vor allem in der Markt- und Meinungsforschung, in der betrieblichen Qualitätskontrolle, in der Wahlforschung und in ähnlichen Bereichen derartigen Stichprobenerhebungen große Bedeutung zukommt.

Das entscheidende Problem dabei ist, daß die aus einer Grundgesamtheit gezogene Stichprobe nach Möglichkeit repräsentativ für eben diese Grundgesamtheit sein soll, d.h. sie soll gewissermaßen ein verkleinertes, getreues Abbild dieser Grundgesamtheit sein.

Es gibt unterschiedliche Methoden, wie man diese gewünschte Repräsentativität herstellen kann :

Die einfachste Vorgehensweise besteht darin, daß man eine Zufallsstichprobe zieht.

Eine solche Zufallsstichprobe liegt dann vor, wenn jedes Element der Grundgesamtheit die gleiche und von null verschiedene Chance hatte, in die Stichprobe aufgenommen zu werden.

Für Zufallsstichproben kann nämlich mathematisch bewiesen werden, daß sie repräsentativ für die Grundgesamtheit sind, aus der sie stammen, wenn

sie nicht zu klein sind.

Für Stichprobenumfänge, die größer als 30 sind, kann eine sehr hohe Wahrscheinlichkeit dafür berechnet werden, daß diese Stichproben repräsentativ sind.

Dieser wahrscheinlichkeitsstatistische Hintergrund kann auch für die gewünschten Hochrechnungen verwendet werden, wie im folgenden ansatzweise skizziert werden soll, wobei auf mathematische Details weitgehend verzichtet wird.

Betrachten wir zunächst ein einfaches Beispiel, um auch noch einmal genau die Fragestellung, um die es hier geht, zu präzisieren :

Nehmen wir einmal an, 100 zufällig ausgewählte Erwachsene werden danach befragt, welche politische Partei sie wählen würden, wenn am kommenden Sonntag Bundestagswahlen wären.

Bei der Auswertung dieser Stichprobenbefunde möge sich ergeben, daß  $P = 40\%$  der Befragten die SPD wählen würden.

Die Frage, die mit den Hochrechnungen nun beantwortet werden soll, lautet folgendermaßen :

In welchen Grenzen liegt (mit berechenbarer Wahrscheinlichkeit) der entsprechende unbekannte Anteilswert in der Grundgesamtheit ?

Diese Frage kann mit der folgenden gedanklichen Überlegung beantwortet werden :

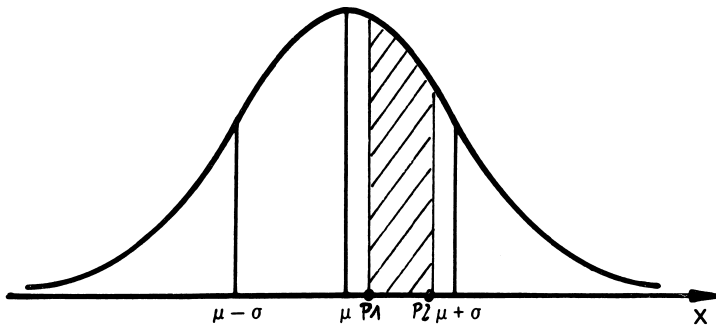
Man stelle sich einmal vor, ein Statistiker zieht alle möglichen Zufallsstichproben vom Umfang  $N$  ( $N = 100$ ), die man überhaupt aus der Grundgesamtheit ziehen kann.

Wie nicht anders zu erwarten ist, erhält man dann sehr, sehr viele und durchaus unterschiedliche Anteilswerte  $P$  (Stichprobenanteilswerte) für SPD - Wähler.

Wenn man für alle diese Anteilswerte eine Häufigkeitsverteilung erstellt, dann erhält man immer (und dieses "immer" ist in diesem Zusammenhang von großer Wichtigkeit) die sog. Gauß-Verteilung, die man auch Normalverteilung nennt.

Dies ist die Aussage des (mathematisch bewiesenen) Zentralen Grenzwerttheorems von Laplace und Liapunoff, dem in der Wahrscheinlichkeitsstatistik, genauso wie natürlich der Normalverteilung, große Bedeutung zukommt.

Die Normalverteilung sieht schematisch so aus, wie es die folgende Skizze zeigt :



Dichtefunktion einer normalverteilten Zufallsvariablen  $X$

Die Gestalt dieser glockenförmigen Kurve wird mit zwei Parametern festgelegt :

1. Der Mittelwert kennzeichnet die Mitte und damit die Lage der Symmetrieachse der Verteilung ( $\mu$ )
2. Die Standardabweichung der Verteilung bestimmt den Abstand der beiden Wendepunkte der Verteilung von der Mittelachse und

entscheidet deshalb darüber, wie breit  
oder schmal die Verteilung ist ( $\sigma$ )

Mit Hilfe dieser Verteilung, deren mathematische  
Funktion bekannt ist, können nun also Häufigkeits-  
aussagen über Stichprobenanteils-  
werte etwa von der  
folgenden Art gemacht werden :

10 % aller Stichprobenanteils-  
werte sind im Bereich  
zwischen  $P_1$  und  $P_2$  zu erwarten (siehe Skizze).

Das Zentrale Grenzwerttheorem besagt nicht nur,  
daß die Normalverteilung die angemessene Häufig-  
keitsverteilung für diese (gedankliche) Aufgaben-  
stellung ist, sondern gibt uns auch eine Informa-  
tion darüber, wie groß diese die Gestalt der Ver-  
teilung charakterisierenden Parameter sind :

Untersucht man beispielsweise Stichprobenanteils-  
werte, so gilt :

Mittelwert der Verteilung =  
(unbekannter) Anteilswert der Grundgesamtheit

Standardabweichung der Verteilung =  
Wurzel aus  $P \cdot (1-P) / N$

Untersucht man hingegen Stichprobenmittelwerte,  
so gilt :

Mittelwert der Verteilung =  
(unbekannter) Mittelwert der Grundgesamtheit

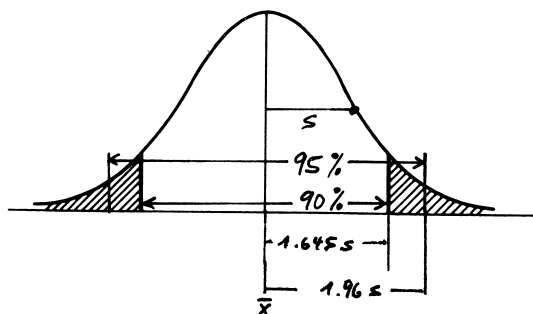
Standardabweichung der Verteilung =  
Standardabweichung der Stichprobe ( $S$ ) divi-  
diert durch die Wurzel aus  $N$

Niemand macht sich aber in der Wirklichkeit die  
Mühe, alle verschiedenen Zufallsstichproben vom  
Umfang  $N$  zu ziehen, sondern in der statistischen  
Praxis hat man nur eine Stichprobe vorliegen.

In diesem Fall ist es aber möglich, Häufigkeitsaussagen in Wahrscheinlichkeitsaussagen zu transformieren :

Wenn bei allen Stichproben 10% der Anteilswerte im Bereich zwischen  $P_1$  und  $P_2$  zu erwarten sind, so gilt für eine einzige Zufallsstichprobe, daß mit einer Wahrscheinlichkeit von 10% deren Anteilswert im Bereich zwischen  $P_1$  und  $P_2$  zu erwarten ist.

In der praktischen statistischen Arbeit sind die 90%- und 95%-Bereiche interessant, die wir in die folgende Skizze eingetragen haben :



Für den 90%-Bereich gilt, daß er das Intervall von  $\bar{X} \pm 1.645 \cdot S$  umfaßt ( $S$  ist jetzt die Standardabweichung der Verteilung, d.h. der Abstand von der Symmetrieachse (Mittelwert) bis zum Wendepunkt);

für den 95%-Bereich gilt entsprechend das Intervall  $\bar{X} \pm 1.96 \cdot S$ .

Aus diesen Überlegungen ergibt sich der folgende zentrale Gedankengang :

Mit 90% Wahrscheinlichkeit ergibt sich in einer Zufallsstichprobe der interessierende Parameter (Anteilswert oder Mittelwert) so, daß das darum aufgebaute Intervall von der Länge  $\pm 1.645 \cdot S$  den wahren (unbekannten) Grundgesamtheitswert erfaßt (der ja mit dem Fußpunkt der Symmetrieachse zusammenfällt) und mit 95% Wahrscheinlichkeit liegt das Stichprobenergebnis so, daß das darum aufgebaute Intervall von der Länge  $\pm 1.96 \cdot S$  den wahren Wert mit erfaßt.

Wenn sich also beispielsweise ein Stichprobenanteilswert von 40% ergibt, so kann festgestellt werden, daß der wahre Grundgesamtheitsanteilswert mit einer Wahrscheinlichkeit von 90% (oder 95%) zwischen A und B zu erwarten ist, wobei die Intervallgrenzen A und B gemäß den obigen Überlegungen berechnet werden können.

Ergänzend sei angemerkt, daß natürlich auch mit anderen Wahrscheinlichkeitswerten außer 90% und 95% gearbeitet werden kann, und daß dieses Instrumentarium nicht nur bei Stichprobenanteilswerten und -mittelwerten verwendet werden kann, sondern auch bei anderen statistischen Maßzahlen.

Diesen Ausführungen folgend soll nun ein BASIC - Programm entwickelt werden.

### 11.3 Schätz-Programm

#### o Vorstellung des Problems

Mit einem BASIC-Programm sollen die 90%- und 95%-Schätzintervalle für Anteilswerte oder für Mittelwerte aus Zufallsstichproben bestimmt werden.

## o Problemanalyse

Die Ausführungen des vorangegangenen Abschnitts haben die wesentlichen problemanalytischen Überlegungen, die hier anzustellen sind, schon vorweggenommen.

Das Programm muß dem Benutzer die Möglichkeit bieten, Schätzintervalle wahlweise für einen Mittelwert oder für einen Anteilswert zu bestimmen; es muß zudem darauf achten, daß zu kleine Stichprobenumfänge ( $N^2=30$ ) als unzulässig abgelehnt werden, weil in diesem Fall die Aussage des Zentralen Grenzwerttheorems nicht mehr mit hinreichender Güte gilt.

Das Programm kann dann das 90%- und das 95%-Intervall bestimmen und als Ergebnis ausgeben.

Dies setzt voraus, daß der Benutzer den jeweiligen Stichprobenparameter eingibt, den Stichprobenumfang und bei der Mittelwertschätzung zusätzlich die Standardabweichung in der Stichprobe, die zur Berechnung der Standardabweichung der zuständigen Normalverteilung benötigt wird.

Gegebenenfalls sind also einige Berechnungen erforderlich, bevor dieses Programm gestartet werden kann.

o Programm

```
10 REM S111-HOCHRECHNUNG
20 PRINTCHR$(147)
30 PRINTTAB(15)"STATISTIK":PRINT:PRINT
40 PRINTTAB(10)"PROF.DR.W.VOSS, 1985"
50 PRINT:PRINT:PRINT
60 PRINTTAB(13)"BEISPIEL S111"
70 PRINT:PRINT TAB(13)"HOCHRECHNUNGEN:"
80 PRINT:PRINT:PRINT
90 PRINT"DIESES PROGRAMM BESTIMMT SCHAEZTZINTER-"
100 PRINT"VALLE FUER MITTELWERTE ODER FUER AN-"
110 PRINT"TEILSWERTE AUS ZUFALLSSTICHPROBEN."
120 GOSUB 1000:REM WARTEN
125 PRINT"WAS SOLL HOCHGERECHNET WERDEN?":PRINT
130 PRINT"MITTELWERT (1)":PRINT
140 PRINT"ODER":PRINT
150 PRINT"ANTEILSWERT (2)":PRINT
158 PRINT:PRINT
160 PRINT "BITTE ZAHL : "
165 GET A$:IF A$="" THEN 165
166 A=VAL(A$):IF A<1 OR A>2 THEN PRINT"EINGABEFehler"
:PRINT:GOTO 160
170 PRINT:PRINT:PRINT CHR$(147)
180 PRINT"DAS PROGRAMM BENDETIGT DIE FOLGENDEN"
190 PRINT"INFORMATIONEN (JEWEILS MIT RETURN) : ":PRINT
:PRINT
200 INPUT "STICHPROBENUMFANG : ";N
210 IF N<30 THEN 460
220 IF Z=2 THEN 340
230 INPUT "STICHPROBENMITTELWERT : ";X
240 PRINT "STANDARDABWEICHUNG"
250 INPUT " DER STICHPROBE : ";S
```

```

260 S=S/SQR(N)
270 X1=X-1.645*S:X2=X+1.645*S
280 B$=" MITTELWERT ";Q=90
290 GOSUB 2000:REM AUSGABE
300 X1=X-1.96*S:X2=X+1.96*S
310 Q=95
320 GOSUB 2000:REM AUSGABE
330 GOTO 530
340 PRINT"ANTEILSWERT DER":INPUT"STICHPROBE (IN %)   : ";X
350 X=X/100
360 S=SQR((X*(1-X)/N))
370 X1=X-1.645*S:X2=X+1.645*S
380 X1=X1*100:X2=X2*100
390 B$=" PROZENTWERT ";Q=90
400 GOSUB 2000:REM AUSGABE
410 X1=X-1.96*S:X2=X+1.96*S
420 X1=X1*100:X2=X2*100
430 Q=95
440 GOSUB 2000:REM AUSGABE
450 GOTO 530
460 PRINTCHR$(147)
470 PRINT"DER STICHPROBENUMFANG IST ZU KLEIN FUER"
480 PRINT"DIE ANWENDUNG DER GAUSS-VERTEILUNG."
490 PRINT:PRINT"ES MUESSEN IN DIESEM FALL SPEZIELLE "
500 PRINT"VERFAHREN DER HOCHRECHNUNG VERWENDET"
510 PRINT"WERDEN, DIE DIESES PROGRAMM NICHT ENT-"
520 PRINTTAB(16)"HAELT."
530 PRINT
540 PRINTTAB(8)"ENDE DER AUSGABE":END

```

```
1000 REM UP WARTEN *****
1010 PRINT:PRINT:PRINT
1020 PRINTTAB(5)"■BITTE DIE LEERTASTE DRUECKEN!□"
1030 GET A$: IF A$="" THEN 1030
1040 PRINTCHR$(147)
1050 RETURN
2000 REM UP AUSGABE *****
2010 PRINT:PRINT:PRINT
2020 PRINT"MIT EINEM VERTRAUEN VON ";Q;" %"
2030 PRINT"LIEGT DER UNBEKANNTE ";B$
2040 PRINT"DER GRUNDGESAMTHEIT ZWISCHEN "
2050 PRINT:PRINT:PRINT"■";X1;" UND ";X2;"□"
2060 PRINT
2070 RETURN
```

o Variablenliste

- A\$ = Tastatureingabe  
B\$ = Bezeichnung (Mittelwert bzw. Anteilswert)  
N = Stichprobenumfang  
Q = Vertrauensniveau (Schätzwahrscheinlichkeit : 90% bzw. 95%)  
S = Standardabweichung in der Stichprobe und Standardabweichung der Normalverteilung  
X = Einzugebender Stichprobenparameter  
X1,X2 = Grenzen des Schätzintervalls  
Z = Auswahlvariable im Menü

o Programmbeschreibung

Satz 10-120 :

Überschrift und Erläuterungen

Satz 130-170 :

Vorgabe des Menüs, damit der Benutzer zwischen der Mittelwert- und der Anteilswertschätzung auswählen kann; entsprechend wird Variable Y mit 1 oder mit 2 belegt; unzulässige Eingaben werden ignoriert (Satz 165-166)

Satz 180-200 :

Anforderung des Stichprobenumfangs N

Satz 210 :

Wenn N zu klein angegeben wurde, erfolgt eine Verzweigung zum Satz 460

Satz 220 :

Wenn  $Z=2$  (Anteilswertschätzung gewünscht), erfolgt eine Verzweigung zum Satz 340

Satz 230-250 :

Anforderung des Stichprobenmittelwerts und der Stichprobenstandardabweichung für die Hochrechnung des Mittelwerts

Satz 260 :

Bestimmung der Standardabweichung der Normalverteilung

Satz 270 :

Berechnung der Grenzen des 90%-Intervalls

Satz 280-290 :

Vorgabe der Informationen für die Ergebnisausgabe und Sprung ins Unterprogramm 2000 zur Ausgabe

Satz 300-320 :

Entsprechende Vorgehensweise für das 95%-Intervall

Satz 330 :

Sprung zum Satz 530, d.h. Überspringen der Anteilswertschätzung

Satz 340 :

Anforderung des Stichprobenanteilswerts (siehe auch Satz 220)

Satz 350-440 :

Berechnung der Intervallgrenzen und Ergebnisausgabe für die Hochrechnung des Anteilswertes ganz entsprechend wie oben für Mittelwerte dargestellt

Satz 450 :

Sprung zum Satz 530, d.h. Überspringen der nachfolgenden Meldung

Satz 460-520 :

Satz 460 wird erreicht, wenn ein zu kleiner Stichprobenumfang eingegeben wurde (siehe Satz 210); es erfolgt dann ein erläuternder Hinweis

Satz 530-540 :

Beendigung des Hauptprogramms

Satz 1000-1050 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 2000-2070 :

Unterprogramm zur Ausgabe der Intervallgrenzen (Ergebnisse der Hochrechnung)

o Ergebnisse

Wenn wir dieses Programm starten, wird zunächst abgefragt, ob eine Mittelwert- oder eine Anteilswertschätzung gewünscht wird.

Entscheiden wir uns beispielsweise für letzteres durch Eingabe der Zahl 2, so fordert das Programm

den Stichprobenumfang an und danach den Anteilswert der Stichprobe.

Nehmen wir an, es läge eine Zufallsstichprobe vom Umfang  $N=1000$  vor, in der sich ein Anteilswert von  $P = 40\%$  (z.B. SPD-Wähler) ergeben hätte.

Das Programm meldet aufgrund dieser Eingaben als Grenzpunkte des 90%-Intervalls die Werte

37.45... und 42.55... (%)

und als Grenzpunkte des 95%-Intervalls die Werte

36.96... und 43.04... (%)

ENDE DER AUSGABE

Ganz generell ist in diesem Zusammenhang anzumerken, daß das Schätzintervall um so kleiner wird, die Hochrechnung also um so präziser, je größer der Stichprobenumfang  $N$  ist (dann wird die Untersuchung aber natürlich teurer) und/oder je kleiner die Schätzwahrscheinlichkeit gewählt wird, wodurch die Aussage dann aber unsicherer wird.

o Ergänzungen

Wir haben schon auf die drei wesentlichen Punkte hingewiesen, die hier eine Rolle spielen :

1. Dieses Instrumentarium zur Hochrechnung von Stichprobenergebnissen kann auch bei anderen Stichprobenparametern benutzt werden (z.B. bei Standardabweichungen, Korrelationskoeffizienten usw.)

2. Wenn der Stichprobenumfang zu klein wird, kann dieses Instrumentarium nicht verwendet werden (Faustregel :  $N$  sollte größer als 30 sein); ersatzweise kann dann mit speziellen Verfahren, die auf kleine Stichprobenumfänge zugeschnitten sind, gearbeitet werden.
3. Statt mit 90%- oder 95%-Intervallen können auch andere Schätzwahrscheinlichkeiten vorgegeben werden (beispielsweise 80 % u.ä.).

Wer sich für diese und weitere Möglichkeiten interessiert, muß auf die statistische Spezialliteratur verwiesen werden (siehe z.B. : M.TIEDE/W.VOSS, Induktive Statistik, 2 Bände, Bochum 1982).

## Kapitel 12 : Das Programmpaket

### 12.1 Vorbemerkung

Wir haben in den vorangegangenen Kapiteln einzelne statistische Auswertungs- und Analyseverfahren als BASIC-Programme vorgestellt.

Die optimale Nutzung dieser Programme wird dadurch ermöglicht, daß man diese Einzelprogramme zu einem Gesamtpaket zusammenschnürt, so daß der Benutzer, der einen bestimmten Datenbestand auswerten möchte, auf alle Methoden, die ihn interessieren, im Ablauf eines einzigen Programms zugreifen kann.

Diese Verknüpfung bedeutet nun allerdings nicht, daß die verschiedenen Einzelprogramme einfach aneinandergehängt werden können, sondern sie müssen aufeinander abgestimmt werden.

Dies führt dazu, daß die Programmlisten des Gesamtpakets, die weiter unten vorgestellt werden, sich deutlich von den Listen der Einzelprogramme unterscheiden. Das gleiche gilt für die Liste der verwendeten Variablen.

Aber auch bei diesen Programmen wurde darauf geachtet, daß die Programmstrukturen möglichst einfach gehalten sind, so daß der Leser bei Bedarf diese Programme nachvollziehen und - was noch wichtiger ist - gemäß seinen eigenen Bedürfnissen verändern kann.

Er sollte bei derartigen Änderungen aber darauf achten, immer auch eine Version (oder besser noch zwei Versionen) des ursprünglichen Programms auf Disketten in Reserve zu behalten.

Es ist es nicht erforderlich, dieses Gesamtpaket in allen Details zu erörtern, weil die sachlichen Einzelprobleme in den vorangegangenen Kapiteln diskutiert wurden. Wir begnügen uns deshalb mit einer generellen Beschreibung.

Einige einschränkende Anmerkungen sind jedoch zunächst vorzuschicken.

## 12.2 Beschränkungen

Trotz seiner vielseitigen Nutzungsmöglichkeiten und seiner erstaunlichen Fähigkeiten sind die kapazitären Möglichkeiten des Commodore C64 doch recht beschränkt : Sehr große Datenmengen können mit diesem Rechner nicht bewältigt werden und vor allem die Nutzung der hochauflösenden Graphik in einigen der Analyseprogrammen nimmt viel des vorhandenen Speicherplatzes in Anspruch.

Diesen Beschränkungen haben wir in dem Programmpaket in der Weise Rechnung getragen, daß wir zunächst die Datenaufnahme generell auf 200 Werte beschränkt haben (bzw. 200 Wertepaare in der bivariaten Statistik).

Desweiteren haben wir auf die hochauflösende Graphik nur dann zurückgegriffen, wenn sie unbedingt erforderlich ist, also beispielsweise in der graphischen Darstellung statistischer Zeitreihen und Trendlinien.

Schließlich ist anzumerken, daß nicht alle statistischen Prozeduren, die in den vorangegangenen Kapiteln vorgestellt wurden, sich in dem Programmpaket wiederfinden. Wir haben uns vielmehr auf die wichtigsten dieser Prozeduren beschränkt, wobei aber immer gewährleistet ist, daß die üblichen statistischen Analysen durchgeführt werden können.

Der Notwendigkeit, Speicherplatz zu sparen, entspricht es auch, daß nicht alle hier zusammengefaßten Prozeduren in einem einzigen Paket versammelt sind, sondern wir stellen quasi "Teilpakete" bereit, die je nach Wunsch des Benutzers aufgerufen werden können.

Diese Teilpakete werden im Abschnitt 12.4 im einzelnen vorgestellt.

Die genannten Beschränkungen können in vielen Fällen (wenn auch nicht in allen, wie gleich noch gezeigt wird) vom Benutzer, wenn dies erforderlich sein sollte, umgangen werden.

Will der Benutzer beispielsweise aus 500 Werten (statt der vorgesehenen maximal 200 Werten) einen Durchschnitt (arithmetisches Mittel) ausrechnen, so ist es ja nicht erforderlich, alle 500 Einzelwerte gleichzeitig im Arbeitsspeicher des C64 zu halten; vielmehr genügen vier Speicherfelder

- ein Zählfeld (I)
- ein Wertfeld (X)
- ein Summenfeld (S)
- ein Hilfsfeld (N)

um die gewünschte Mittelwertbildung - etwa mit dem folgenden Programm - zu vollziehen :

```
10 REM MITTELWERT AUS BELIEBIG VIELEN WERTEN
20 INPUT "WIEVIELE WERTE : ";N
30 FOR I=1 TO N
40 PRINT I;" .WERT BITTE : ";INPUT X
50 S=S+X
60 NEXT I
70 PRINT:PRINT "MITTELWERT = ";S/N
80 PRINT:PRINT "ENDE":END
```

In entsprechender Weise kann man bei vielen anderen statistischen Prozeduren auch verfahren; die entsprechenden notwendigen Programmänderungen kann

der Benutzer sicherlich leicht selbst erfolgreich zustandebringen.

Leider sind derartige Ausweitungen aber nicht bei allen Analyseverfahren möglich. Es gibt Verfahren, bei denen alle in Frage kommenden Merkmalswerte im Arbeitsspeicher benötigt werden, so z.B. dann, wenn ein Datenbestand der Größe nach sortiert werden soll.

In einem solchen Fall kann ein größerer Datenbestand nur dann bewältigt werden, wenn man völlig andere Programmstrukturen als die hier vorgestellten wählt. Diese hier zu besprechen, würde aber die Aufgabenstellungen dieses Buches weit überschreiten.

Auch die andere erwähnte Beschränkung, die darin besteht, daß nicht alle vorgestellten Programme sich in dem Programmpaket wiederfinden, ist ohne große Schwierigkeiten zu umgehen :

Beispielsweise haben wir darauf verzichtet, das Programm in das Paket aufzunehmen, welches ein Kreisdiagramm (Kuchendiagramm) zur graphischen Veranschaulichung einer Häufigkeitsverteilung erzeugt (Programm S72-KREIS).

Wenn nun ein Benutzer unbedingt ein derartiges Kreisdiagramm erstellen möchte, so kann er, anstatt das Programmpaket zu benutzen, dieses Programm S72 laden und es mit seinen eigenen Daten, die er über DATA-Anweisungen eingeben kann, starten.

In einer etwas komfortableren Vorgehensweise kann er in der folgenden Weise auf dieses Programm zugreifen, wobei auch das Programmpaket selbst genutzt wird :

1. Aufruf des Programmpakets  
(vergleiche dazu den Abschnitt 12.4)
2. Auswahl des Teilpakets ESP

(ESP = Eingabe und Speichern von Daten; siehe Abschnitt 12.4)

3. Erzeugung und Speichern der Datei mit den Daten des Benutzers (dies leistet das Teilpaket ESP)
4. Laden des Programms S72-KREIS
5. Verändern dieses Programms, so daß es selbst auf die in 3. erzeugte Datei zugreift und die dort gespeicherten Daten liest, um mit diesen dann das Kreisdiagramm zu erzeugen.

Gemäß dieser schematischen Beschreibung kann prinzipiell unter Nutzung des Programmpakets auf jeder der in den vorangegangenen Kapiteln vorgestellten Teilprogramme zugegriffen werden.

Diese Erörterungen verdeutlichen einen sehr wichtigen Sachverhalt :

\*\*\*\*\*

Das Programmpaket kann nur verwendet werden, wenn man über ein externes Speichergerät verfügt.

\*\*\*\*\*

Wir gehen im folgenden vom Vorhandensein eines Diskettenlaufwerks aus. Ist kein externer Speicher vorhanden, können nur die Einzelprogramme nach Eintippen derselben verwendet werden.

In diesem Zusammenhang sei mit besonderem Nachdruck daran erinnert, daß es sich immer empfiehlt, sowohl Programme, wie auch Daten zweimal (und natürlich auf zwei verschiedenen Disketten) zu speichern, um bei Informationsverlusten oder bei Dis-

kettenbeschädigungen nicht vor unlösbaren Problemen zu stehen.

### 12.3 Anmerkungen zur Nutzung des Programmpakets

Das Programmpaket "STATISTIK" und seine einzelnen Teilprogramme verlangen vom Benutzer den Einstieg in einen Dialog. Dies bedeutet, daß der Benutzer auf Fragen der Programme antworten muß.

Dabei ist eine gewisse Sorgfalt notwendig, um Programmabbrüche zu vermeiden.

Beispielsweise wird bei der Dateneingabe der Benutzer aufgefordert, statistische Daten einzugeben. Diese Eingaben müssen Zahlen (also nicht etwa Worte oder Texte) sein, wenn Fehler vermieden werden sollen.

Will der Benutzer das Körpergewicht 75 kg eingeben, so erwartet das Programm die Zahl 75 und nicht etwa den String "75 kg".

Wenn vom Programm jeweils zwei Zahlenwerte angefordert werden (dies ist in der bivariaten Statistik der Fall, also dann, wenn zwei statistische Variablen, zwei Merkmale gleichzeitig untersucht werden sollen), so sind je zwei Zahlen durch ein Komma voneinander getrennt einzugeben und danach die RETURN-Taste zu betätigen.

Bei Dezimalzahlen (reelle Zahlen) werden bekanntlich die Dezimalstellen durch einen Punkt abgetrennt.

Werden bei der Dateneingabe Fehler gemacht, so sind Korrekturen mit dem Programm DSP möglich, welches über die Menüziffer 2 aufgerufen werden kann.

Eine andere Art der Informationseingabe im Dialog

mit dem jeweiligen Programm liegt dann vor, wenn der Benutzer aufgefordert wird, mit J (für JA) oder N (für NEIN) zu antworten - so z.B. im eben erwähnten Korrekturmodus :

Das Programm fragt :

SIND ALLE DATEN IN ORDNUNG (J/N) ?

und der Benutzer braucht jetzt nur die J- oder die N-Taste zu drücken (JA oder NEIN), ohne daß danach die RETURN-Taste erforderlich ist. Andere Tasten - eingaben akzeptiert das Programm nicht.

In einem komplizierten dialogorientierten Programm ist es aber nicht möglich, alle denkbaren Fehlermöglichkeiten, die der vielleicht noch ungeübte Benutzer begehen kann, vorherzusehen und entsprechende Sicherungen einzubauen. Es ist also durchaus vorstellbar, daß das Programm einmal mit einer Fehlermeldung stehenbleibt. Dem Benutzer kann dann nur geraten werden, die Bedienungshandbücher zu studieren, um die Fehlerquelle zu finden und dann das Programmpaket erneut zu starten.

Nicht selten ist es bei statistischen Datenbeständen der Fall, daß bestimmte Werte fehlen. So gibt es beispielsweise Lücken in Zeitreihen oder man stellt in einer Fragebogenaktion fest, daß einzelne Befragte auf bestimmte Fragen nicht geantwortet haben, weil sie es vielleicht vergessen oder diese Frage übersehen haben, oder weil sie eben die Antwort verweigerten.

Solche Fälle müssen in Statistikprogrammen identifiziert werden können, weshalb man ihnen in der Regel einen Zahlenwert außerhalb des realistischen Wertebereichs zuweist.

Erfaßt man beispielsweise die Kinderzahl verschiedener Familien zum Zwecke der statistischen Auswertung und hat Familie Nr. 423 keine Angabe auf die entsprechende Frage gemacht, so könnte man in diesem Fall den Wert -1 zuweisen, denn dieser Wert kann mit einer "echten" Kinderzahl nicht verwechselt werden.

Bei einer Mittelwertberechnung könnte man dann wie folgt programmieren (wir setzen bei dieser schematischen Darstellung voraus, daß die Kinderzahlen X(I) schon über DATA- und READ-Anweisungen eingelesen wurden) :

```
10 N=500 : REM ANZAHL DER FAMILIEN
20 FOR I=1 TO N
30 IF X(I)=-1 THEN MV=MV+1:GOTO 50
40 S=S+X(I)
50 NEXT I
60 PRINT "MITTELWERT = ";S/(N-MV)
70 END
```

Einen derartigen Ersatzwert nennt man "missing value". Wir haben in den einzelnen Programmen des Programmpakets standardmäßig den Wert -99 vorgegeben und im Feld MV% gespeichert. Sollte der Wert -99 im konkreten Anwendungsfall in den realen Bereich der Untersuchungsvariable gehören, muß der Benutzer diese Voreinstellung natürlich ändern, indem er vor dem jeweiligen Programmstart MV% neu belegt.

Entsprechend haben wir als Standardwert für das Ende eines Datenbestands den Wert -111 vorgegeben und im Feld EF% gespeichert. Auch dies muß im Einzelfall geändert werden, wenn -111 zum realen Wertebereich gehören sollte.

Dieser Wert ist notwendig, damit es möglich ist, Eingaberoutinen abzurechnen, wenn der Benutzer keine Werte mehr einzugeben hat. Er gibt dann den Wert -111 ein, und das Programm erkennt daran das Ende des Datenbestandes und gelangt dann zu den weiteren Auswertungsschritten.

## 12.4 Genereller Überblick über das Paket

Das Programmpaket, um das es hier geht, besteht aus den acht folgenden Teilpaketen :

### 0. STATISTIK

Dieses Programm dient dazu, dem Benutzer eines der folgenden sieben Programme zur Auswahl anzubieten.

#### 1. ESP

Dieses Programm dient der Eingabe von Daten (E), und zu ihrer Speicherung (SP) auf Diskette.

Es besteht dabei die Möglichkeit, eine Datei mit den Daten des Benutzers neu anzulegen oder eine schon bestehende Datei zu verlängern.

Es können Zeitreihendaten, univariate Daten (Betrachtung nur eines Merkmals) oder bivariate Daten (Betrachtung von zwei Merkmalen gleichzeitig) behandelt werden (dies gilt auch für alle folgenden Programme).

#### 2. DSP

Dieses Programm dient dazu, einen Datenbestand von einer Diskette zu lesen (D), ihn eventuell zu korrigieren und ihn, falls dies gewünscht wird, wieder auf Diskette zu speichern (SP).

Verlängerungen des Datenbestandes sind nur im Programm ESP möglich (s.o.).

#### 3. DZA

Dieses Programm dient dazu, einen Datenbestand von einer Diskette einzulesen (D) und einer zeitreihenstatistischen Auswertung zu unterziehen (ZA).

Dies umfaßt die tabellarische Darstellung der Daten, die lineare Interpolation fehlender Werte, die Berechnung einer linearen Trendlinie mit der Methode der kleinsten Quadrate, Trendprognosemöglichkeiten und die graphische Darstellung der Reihe und der Trendlinie.

#### 4. DUT

In diesem Programm wird ein von einer Diskette gelesener Datenbestand (D) einer univariaten statistischen tabellarischen Analyse (UT) unterzogen.

Dies umfaßt die Erstellung der Verteilungen absoluter und relativer Häufigkeiten (wobei im stetigen Fall Klassifizierungsbreite und Startpunkt der Klassifizierung vom Benutzer frei vorgegeben werden können), die Berechnung von arithmetischem Mittel, Spannweite und Standardabweichung und die Erstellung von Häufigkeitstabellen. Graphische Darstellungen der Häufigkeitsverteilungen (Stabdiagramm oder Histogramm) sind nur mit dem folgenden Programm möglich.

#### 5. DUG

Mit diesem Programm werden univariate Daten (U) von Diskette gelesen (D) und die jeweiligen Verteilungen werden graphisch dargestellt (G) (Stabdiagramm oder Histogramm)

#### 6. DBT

in diesem Programm werden Daten, die aus einem externen Speicher (Diskette) gelesen werden (D), bivariaten statistischen Auswertungsverfahren unterworfen (B), wobei auf graphische Ergebnisse zugunsten tabellarischer (T) Ausgaben und der Ausgabe statistischer Maßzahlen verzichtet wird.

Dies bedeutet, daß in diesem Programm zunächst die eingelesenen Daten präsentiert werden. Danach wird im diskreten Fall eine Kreuztabelle ausgegeben und es werden für alle Fälle die relevanten statisti -

schen Maßzahlen berechnet, nämlich die Parameter der linearen Regressionsfunktion, der Korrelationskoeffizient und im diskreten Fall zusätzlich der Kontingenzkoeffizient.

## 7. DBG

Dieses Programm erfüllt die gleichen Aufgaben wie das Programm DBT mit dem Unterschied, daß nun auf bivariate Tabellen zugunsten graphischer Darstellungen (G) verzichtet wird.

In diesem Programm wird also ein Streudiagramm zur graphischen Veranschaulichung der gegebenen bivariaten Häufigkeitsverteilung erstellt.

\*\*\*\*\*

Bei der Benutzung ist, wenn man sich der entsprechenden Programmdiskette bedient, folgendermaßen vorzugehen :

\*\*\*\*\*

### 1. Laden des Programms STATISTIK

Wenn dieses Programm mit dem Kommando RUN nach dem Laden von der Programmdiskette gestartet wird, so wird auf dem Bildschirm ein Menü angeboten, welches dem Benutzer erlaubt, eines der Programme

ESP, DSP, DZA, DUT, DUG, DBT oder DBG

(1) (2) (3) (4) (5) (6) (7)

auszuwählen (als zusätzliche Wahlmöglichkeit wird darüberhinaus der "Ausstieg" aus dem gesamten Programmpaket angeboten).

Wenn sich der Benutzer für eines der oben genannten Programme entschieden hat, so wird dieses, ohne daß eine weitere Anweisung erforderlich wäre, automatisch geladen, verdrängt dabei das nun nicht mehr benötigte Programm STATISTIK und steht dann

dem Benutzer zur Verfügung (sog. Overlay-Technik).

Diese Vorgehensweise ist nur dann möglich, ohne daß Programmfehler auftauchen, wenn

1. das nachgeladenen Programm vom vorhergehenden keine Informationen übernehmen muß (dafür ist hier Sorge getragen) und wenn
2. das nachgeladene Programm mit folgendem Satz eingeleitet wird :

```
POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
```

Dieser Satz befindet sich deshalb am Anfang aller Teilprogramme, vom Programm STATISTIK abgesehen.

(Zur näheren Erläuterung der Overlay-Technik siehe ANGERHAUSEN u.a.: 64 für Profis - ein DATA BECKER Buch).

Nach Ablauf des gewählten Programms wird wiederum automatisch das Programm STATISTIK geladen, so daß erneut ausgewählt werden kann, bzw. der "Ausstieg" aus dem Gesamtpaket beschlossen werden kann.

Diese Art der Nutzung des Programmpakets setzt also voraus, daß der Benutzer die Programmdiskette besitzt.

Diese Diskette nennen wir die PROGRAMMDISKETTE.

Wenn der Benutzer mit dem Programm ESP arbeitet, muß er eine formatierte Diskette bereithalten, damit der erzeugte Datenbestand gespeichert werden kann. Diese Diskette wird dann auch bei allen anderen Programmen benötigt.

Diese Diskette nennen wir die DATENDISKETTE.

Es ist generell zu empfehlen, die Dateien, die die

auszuwertenden Datenbestände beinhalten, nicht auf der Programmdiskette, sondern auf einer eigenen, nämlich auf der eben genannten Datendiskette anzulegen.

Bei der Verwendung des Programmpakets ist also generell folgendermaßen vorzugehen :

- \* 1. Laden des Programms STATISTIK
- \*
- \* 2. Starten dieses Programms nach der READY-Meldung mit dem Kommando RUN
- \*
- \* 3. Auswahl eines der Teilpakete

Nehmen wir einmal an, der Benutzer wollte einen schon auf Diskette vorhandenen Datenbestand einlesen, korrigieren und einer univariaten Analyse unterziehen. Er müßte dazu folgendermaßen vorgehen :

1. Laden von STATISTIK und Starten mit RUN
2. Auswahl von DSP (Menüziffer 2)
3. Korrigieren des Datenbestands
4. Speichern des korrigierten Datenbestands auf der Datendiskette
5. Nach dem Wiederauftauchen des Auswahlmenüs aus dem Programm STATISTIK (dies erfolgt automatisch), Auswahl des Programms DUT (Menüziffer 4) und Durchführung der statistischen Analyse.
6. Nach dem Wiederauftauchen des Auswahlmenüs aus dem Programm STATISTIK Beendigung des Programmlaufs durch Auswählen der Menüziffer 9.

Ist der Benutzer ganz sicher, daß keine Korrekturen erforderlich sind, kann natürlich in Schritt 2 direkt das Programm DUT ausgewählt werden,

Schritt 5 würde zu Schritt 3 und der Programmablauf könnte auf diese Weise verkürzt werden.

## 12.5 Programmlisten und -beschreibungen

In diesem Abschnitt werden die Programmlisten des Gesamtpaketes vorgestellt, die jeweils stichwortartig beschrieben werden.

Im Gegensatz zu den Beschreibungen der einzelnen Programme der vorangegangenen Kapitel beschränken wir uns hier auf eine grobe Skizzierung; die Detailprobleme und die einzelnen Programmschritte sind ja in den vorangegangenen Kapiteln dargelegt worden.

### Programm STATISTIK

```
10 REM STATISTIK
20 PRINTCHR$(147)
30 PRINTTAB(10)"STATISTIK":PRINT:PRINT:PRINT
40 PRINTTAB(6)"PROF.DR.WERNER VOSS, 1985"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"PROGRAMMPAKET ZUR STATISTISCHEN ANALYSE"
70 PRINT"VON ZEITREIHEN, UNIVARIATEN UND BIVA-"
80 PRINT"TEN DATENBESTAENDEN MIT STANDARDMETHO-"
90 PRINTTAB(17)"DEN.":PRINT
92 PRINT" EINZELHEITEN ZU DEN VERFAHREN FINDEN"
93 PRINT" SICH BEI ":PRINT
94 PRINT"  *****"
95 PRINT"    VOSS : DAS STATISTIK-BUCH ZUM C64"
96 PRINT"    (EIN DATA-BECKER BUCH,      1985)"
97 PRINT"  *****"
100 GOSUB 700:REM WARTEN
```

```

120 REM
130 PRINT"DER BENUTZER KANN EINE DER FOLGENDEN"
140 PRINT"MOEGlichkeiten WAELHEN :":PRINT
150 PRINT"DATENEINGABE UND SPEICHERUNG (1)"
170 PRINT:PRINT"EINLESEN DER DATEN VON DISKETTE,"
180 PRINT" KORREKTUR UND NEU-SPEICHERN (2)"
190 PRINT:PRINT"EINLESEN DER DATEN VON DISKETTE UND"
200 PRINT:PRINT" ZEITREIHENSTATISTIK (3)"
210 PRINT:PRINT" UNIVARIATE TABELLEN (1 MERKMAL) (4)"
215 PRINT:PRINT" UNIVARIATE GRAPHIKEN (1 MERKMAL) (5)"
220 PRINT:PRINT" BIVARIATE TABELLEN (2 MERKMALE) (6)"
230 PRINT:PRINT" BIVARIATE GRAPHIK (2 MERKMALE) (7)"
240 PRINT:PRINT:PRINT"NICHTS DAVON (9)"
250 PRINT:PRINT"BITTE NUMMER EINGEBEN"
252 GET Z9$:IF Z9$="" THEN 252
254 Z9=VAL(Z9%):Z$=" EINGABEFehler !"
256 IF Z9=0 THEN PRINT:PRINTZ$:GOTO 120
260 IF Z9>7 AND Z9<>9 THEN PRINT:PRINTZ$:GOTO 120
270 IF Z9=9 THEN 950
280 ON Z9 GOTO 300,350,400,450,470,500,550
300 GOSUB 800:REM MELDUNG
310 LOAD "ESP",8
350 GOSUB 800:REM MELDUNG
360 LOAD "DSF",8
400 GOSUB 800:REM MELDUNG
410 LOAD "DZA",8
450 GOSUB 800:REM MELDUNG
460 LOAD "DUT",8
470 GOSUB 800:REM MELDUNG
480 LOAD "DUG",8
500 GOSUB 800:REM MELDUNG
510 LOAD "DBT",8
550 GOSUB 800:REM MELDUNG
560 LOAD "DBG",8
700 REM UP WARTEN *****
710 POKE 214,21:POKE 211,5:SYS 58640
720 PRINT:PRINT:PRINTTAB(5)"BITTE DIE LEERTASTE
DRUECKEN !"
730 GET A9$:IF A9$="" THEN 730
740 PRINTCHR$(147):RETURN
800 REM UP MELDUNG *****
810 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
820 PRINTTAB(5)"DAS PROGRAMM WIRD GELADEN."
830 RETURN
950 PRINT CHR$(147):PRINTTAB(12)"ENDE DER AUSGABE"
960 PRINT:PRINT:PRINTTAB(10)"NEUSTART NUR MIT RUN"
970 PRINT:PRINT:PRINTTAB(13)"PROF. DR. W. VOSS":END

```

## Beschreibung des Programms STATISTIK

Satz 10-100 :

Überschrift, Erläuterungen und Abwarten

Satz 120-250 :

Ausgabe des Menüs und Anforderung einer Menünummer

Satz 252-260 :

Kontrolle der Menünummer

Satz 270 :

Verzweigung zum Programmende, falls dies gewünscht wurde

Satz 280-560 :

Verzweigung zu den einzelnen Teilprogrammen entsprechend der eingegebenen Menünummer; zunächst wird jeweils eine Meldung ausgegeben (Unterprogramm 800) und danach wird das entsprechende Programm nachgeladen

Satz 700-740 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 800-830 :

Unterprogramm zur Ausgabe einer Wartemeldung

Satz 950-970 :

Beendigung des Programms STATISTIK

**Programm ESP**

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(10)"STATISTIK":
    PRINT:PRINT:PRINT
1020 PRINT"PROGRAMM ZUR EINGABE STATISTISCHER DATEN"
1030 PRINTTAB(12)"AUF DISKETTE":PRINT:PRINT
1040 PRINT:PRINT:PRINT
1050 PRINT"KORREKTUREN DER DATEN SIND MIT DEM PRO-"
1060 PRINT"GRAMM DSP (MENUE-NR.2 IM PROGRAMM STA-"
1070 PRINTTAB(10)"TISTIK) MOEGLICH":GOSUB 7000
1260 N%=201:EF%=-111:MV%=-99
1270 DIM X(N%),Y(N%)
1280 PRINTCHR$(147):PRINT"AUSWAHL DER EINGABE : "
1290 PRINT:PRINT:PRINT" - ZEITREIHENDATEN (1)"
1310 PRINT:PRINT" - UNIVARIATE DATEN (1 MERKMAL) (2)"
1320 PRINT:PRINT" - BIVARIATE DATEN (2 MERKMALE) (3)"
1330 PRINT:PRINT:PRINT" NICHTS DAVON (9)"
1340 PRINT:PRINT:PRINT"BITTE NUMMER ANGEBEN"
1342 GET Z1$:IF Z1$="" THEN 1342
1344 Z1%=VAL(Z1$):Z$=" FALSCH EINGABE !"
1346 IF Z1%=0 THEN PRINT:PRINT:PRINTZ$:GOTO 1290
1350 IF Z1%>3 AND Z1%<>9 THEN PRINT:PRINTZ$: GOTO 1290
1360 IF Z1%=9 THEN GOSUB 6600
1370 ON Z1% GOTO 1500,2500,3000
```

```

1500 REM ZEITREIHENSTATISTIK -----
1510 PRINTCHR$(147):PRINTTAB(8)"#ZEITREIHENSTATISTIK#":
PRINT:PRINT
1520 GOSUB 8500:REM MENUE2
1530 GOSUB 6600:REM ENDE
2500 REM UNIVARIATE STATISTIK -----
2510 PRINTCHR$(147):PRINTTAB(8)"#UNIVARIATE STATISTIK#":
2520 PRINT:PRINTTAB(9)"(NUR EIN MERKMAL)":PRINT:PRINT
2530 GOSUB 8500:REM MENUE2
2540 GOSUB 6600:REM ENDE
3000 REM BIVARIATE STATISTIK -----
3010 PRINTCHR$(147):PRINTTAB(8)"#BIVARIATE STATISTIK#":
3020 PRINT:PRINTTAB(8)"(ZWEI MERKMALE ZUGLEICH)":
PRINT:PRINT
3030 GOSUB 8500:REM MENUE2
3040 GOSUB 6600:REM ENDE
3200 REM UP TASTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 PRINT:PRINT:RETURN
4000 REM UP DISK *****
4010 INPUT#15,EN,EM$,ET,ES
4020 IF EN=0 THEN RETURN
4030 PRINT:PRINT:PRINT
4040 PRINTTAB(12)"#DISKETTENFEHLER#":PRINT:PRINT
4050 PRINTEN;EM$;ET;ES
4060 PRINT:PRINT:PRINT
4070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
4080 PRINT"LADEN DES PROGRAMMS STATISTIK !":
4090 PRINT:PRINT:PRINT
4100 PRINTTAB(10)"ENDE DES PROGRAMMS":END

```

```

5000 REM UP EINGABE *****
5010 PRINTCHR$(147):B$="WERT":C$=" (UND RETURN)"
5020 IF Z1%=3 THEN B$="PAAR(X,Y)"
5030 GOSUB 7150
5040 FOR I=IA TO IE
5050 IF I=N% THENPRINT"LETZTER WERT !":PRINT"-111
    EINGEBEN!":PRINT
5060 PRINTI:",";B$;C$;" : ";
5070 IF Z1%=3 THEN INPUT X(I),Y(I):GOTO 5090
5080 INPUT X(I)
5090 IF X(I)=EF% THEN KA=IA:KE=I-1:GOSUB 6000:GOTO 5130
5110 NEXT I
5120 KA=IA:KE=I-1:GOSUB 6000
5130 RETURN
6000 REM UP SPEICHERN *****
6010 IF KA>KE THEN RETURN
6020 PRINT:PRINT"▣ MOMENT BITTE, ES WIRD GESPEICHERT!▣"
6025 NN$="A"
6030 IF M$="N" AND IA=1 THEN NN$="W"
6105 OPEN 15,8,15
6110 OPEN 8,8,8,N$+" ,S,"+NN$
6115 GOSUB 4000:REM DISK
6120 FOR K=KA TO KE
6130 IF Z1%=3 THEN PRINT#8,X(K):PRINT#8,Y(K):GOTO 6150
6140 PRINT#8,X(K)
6150 NEXT K
6160 GOSUB 4000:CLOSE 8:CLOSE 15:RETURN
6500 REM UP DATENDISK *****
6510 PRINTCHR$(147)
6520 POKE 214,21:POKE 211,3:SYS 58640
6530 PRINT"▣FORMATIERTE DATENDISKETTE EINLEGEN"
6540 PRINT"      UND LEERTASTE DRUECKEN !▣"
6550 GET A$:IF A$="" THEN 6550
6560 PRINTCHR$(147):RETURN

```

```

6600 REM UP PROGRAMMDISK UND ENDE *****
6610 PRINTCHR$(147)
6620 POKE 214,21:POKE 211,3:SYS 58640
6630 PRINT"PROGRAMMDISKETTE WIEDER EINLEGEN"
6640 PRINT"          UND LEERTASTE DRUECKEN !!"
6650 GET A$:IF A$="" THEN 6650
6660 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE!"
6670 LOAD "STATISTIK",8
7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "BITTE DIE LEERTASTE DRUECKEN !!"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7150 REM UP STATUSZEILE EINGABE *****
7160 POKE 214,3:POKE 211,2:SYS 58640
7170 PRINT"FEHLENDER WERT = -99":PRINT"  LETZTER WERT
    = -111!":PRINT
7180 RETURN
7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"NR.", "X-WERT", "Y-WERT!"
7420 FOR JJ=1 TO 26:PRINT"-":NEXT JJ:PRINT:RETURN

```

```

8500 REM UP MENUE2 *****
8510 PRINT"NEU-ANLEGEN EINER DATEI (N)"
8520 PRINT:PRINT"ODER":PRINT:PRINT"VERLAENGERN EINER DATEI
(V)"
8530 PRINT:PRINT:PRINT"BITTE N ODER V : ?"
8535 GET M$:IF M$="" THEN 8535
8540 IF M$<>"N" AND M$<>"V" THEN 8535
8550 GOSUB 6500:REM DATENDISK
8560 IF M$="V" THEN 8630
8570 PRINT:PRINT"NAME DER NEUEN DATEI":INPUT "(UND RETURN)
":N$
8580 J=1
8590 IA=(J-1)*10+1:IE=IA+9
8600 GOSUB 5000:REM EINGABE
8610 IF X(I)=EF% THEN GOSUB 6600
8620 J=J+1:GOTO 8590
8630 PRINT:PRINT "NAME DER ZU VERLAENGERNDEN":INPUT
"DATEI (UND RETURN)":N$
8640 PRINT:PRINT:PRINTTAB(13)"MOMENT BITTE?":PRINT:PRINT
8645 OPEN 15,8,15
8650 OPEN 8,8,8,N$+",S,R"
8655 GOSUB 4000:REM DISK
8660 IF Z1%=3 THEN 8700
8670 I=1
8680 INPUT#8,X:IF ST AND 64 THEN 8740
8690 I=I+1:GOTO 8680
8700 I=1
8710 INPUT#8,X
8720 INPUT#8,Y:IF ST AND 64 THEN 8740
8730 I=I+1:GOTO 8710
8740 GOSUB 4000:CLOSE 8:CLOSE 15
8750 PRINT:PRINT:PRINT"ANZAHL DER FAELLE IN DER DATEI :
":I:NA=I
8760 PRINT:PRINT"ES KOENNEN NOCH":N%-I-1;" EINGABEN
ERFOLGEN.":GOSUB 7000
8770 J1=1:J=INT(NA/10)+1
8780 IF J1=1 THEN IA=NA+1:IE=J*10:GOTO 8800
8790 IA=(J-1)*10+1:IE=IA+9
8800 GOSUB 5000
8810 IF X(I)=EF% THEN GOSUB 6600
8820 J=J+1:J1=J1+1:GOTO 8780
8830 RETURN

```

## Beschreibung des Programms ESP

Satz 1000-1070 :

Vorbereitung des Nachladens, Überschrift und Erläuterungen

Satz 1260-1270 :

Vorgabe der Voreinstellungen und Dimensionierungen

Satz 1280-1350 :

Auswahlmenü zur Festlegung, welche Art statistischer Daten vorliegen; Abfangen unzulässiger Eingaben

Satz 1360-3040 :

Verzweigungen zur Eingabe von Zeitreihendaten, von univariaten oder von bivariaten Daten je nach Menüauswahl

Satz 3200-3230 :

Unterprogramm zum Abwarten, bis J oder N (für Ja oder Nein) eingegeben wird

Satz 4000-4100 :

Unterprogramm zum Abfangen von Fehlern bei der Diskettennutzung (Abfragen des Fehlerkanals)

Satz 5000-5130 :

Unterprogramm zur Eingabe von Daten (Einzelwerte bei Zeitreihen und univariater Statistik; Wertepaare bei bivariater Statistik), beginnend mit der Ausgabe einer sog. Statuszeile (Unterprogramm 7150), Anforderung von je 10 Werten (Wertepaaren)

und Verzweigung ins Unterprogramm 6000 zum Speichern der eingegebenen Daten

Satz 6000-6160 :

Unterprogramm zum Speichern der eingegebenen Daten; ab dem zweiten Datenblock wird die Datei verlängert (NN\$ = "A")

Satz 6500-6560 :

Unterprogramm mit der Aufforderung zum Einlegen der Datendiskette

Satz 6600-6670 :

Unterprogramm mit der Aufforderung zum Einlegen der Programmdiskette und zum Nachladen des Programms STATISTIK (Ende des Programms ESP)

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 7150-7180 :

Unterprogramm zur Ausgabe der Statuszeile (Hinweise für die Dateneingabe)

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 8500-8830 :

Unterprogramm zur Steuerung der Dateneingabe :

Es wird zunächst abgefragt, ob ein Datenbestand neu eingegeben werden soll, oder ob der Benutzer einen schon vorhandenen Datenbestand zu verlängern wünscht;

im zweiten Fall wird der Name der zu verlängern-

den Datei angefordert (8630) und es wird zunächst festgestellt, wieviele Werte oder Wertepaare (also wieviele Fälle) in dieser Datei schon vorhanden sind (8650-8760);

danach oder bei Neueingabe direkt erfolgt der An-  
sprung des Unterprogramms 5000 (in Zehnerschrit-  
ten bzw. bis zur Eingabe des Abbruchwertes -111);

über dieses Unterprogramm 5000 wird Unterprogramm  
6000 erreicht, mit dem gespeichert wird, womit das  
Programm ESP beendet wird

Programm DSP

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(10)"STATISTIK"
:PRINT:PRINT:PRINT
1020 PRINT"EINLESEN STATISTISCHER DATEN AUS EINER"
1030 PRINTTAB(14)"DISKETTE"
1040 PRINT:PRINT:PRINT:PRINT
1050 PRINTTAB(8)"PROF.DR.W. VOSS, 1985"
1060 PRINT:PRINT:PRINT:PRINT
1062 PRINT"WERDEN INFORMATIONEN GEWUENSCHT (J/N) ?"
1064 GOSUB 3200
1068 IF A$="N" THEN 1260
1069 PRINTCHR$(147)
1070 PRINT"MIT DIESEM PROGRAMM KOENNEN":PRINT
1080 PRINT" - ZEITREIHENWERTE"
1090 PRINT" - UNIVARIATE DATEN (NUR 1 MERKMAL)"
1100 PRINT" - BIVARIATE DATEN (2 MERKMALE)":PRINT
1110 PRINT"GELESEN WERDEN.":PRINT:PRINT
1120 PRINT" DIE GELESENEN WERTE WERDEN DANN NACH "
1130 PRINT" EVENTUELLEN KORREKTUREN WIEDER GE-"
1140 PRINT" SPEICHERT, SO DASS MIT ANDEREN PRO-"
1150 PRINT" GRAMMEN AUF SIE ZUGEGRIFFEN WERDEN"
1160 PRINTTAB(17)"KANN.":GOSUB 7000
1170 PRINT"WEGEN DER BESCHRAENKTEN KAPAZITAEREN"
1180 PRINT"MOEGlichkeiten IN DEN ANALYSEPROGRAMMEN"
1190 PRINT"WERDEN DIE DATEN AUF MAXIMAL 200 WERTE"
1200 PRINT"(BZW. WERTEPAARE) BEGRENZT."
1210 PRINT:PRINT:PRINT"WENN WERTE FEHLEN (LUECKEN),
SO IST GE-"
1220 PRINT"MAESS VOREINSTELLUNG -99 VERWENDET."
1230 PRINT:PRINT:PRINT"ALS LETZTER WERT EINES
DATENBESTANDS"
1240 PRINT"IST DER WERT -111 FESTGELEGT."
1250 GOSUB 7000:REM WARTEN
```

```

1260 N%=201:EF%=-111:MV%=-99
1270 DIM X(N%),Y(N%)
1280 PRINTCHR$(147):PRINT"  AUSWAHL DER STATISTIK : "
1290 PRINT:PRINT:PRINT" - ZEITREIHENDATEN (1)"
1310 PRINT:PRINT" - UNIVARIATE DATEN (1 MERKMAL) (2)"
1320 PRINT:PRINT" - BIVARIATE DATEN (2 MERKMALE) (3)"
1330 PRINT:PRINT:PRINT"      NICHTS DAVON (9)"
1340 PRINT:PRINT:PRINT"  BITTE NUMMER ANGEBEN:"
1342 GET Z1$:IF Z1$="" THEN 1342
1344 Z1%=VAL(Z1%):Z$="  FALSCH EINGABE !:"
1346 IF Z1%=0 THEN PRINT:PRINT:PRINTZ$:GOTO 1290
1350 IF Z1%>3 AND Z1%<>9 THEN PRINT:PRINT:PRINTZ$: GOTO 1290
1360 IF Z1%=9 THEN GOSUB 6600
1370 ON Z1% GOTO 1500,2500,3000
1500 REM ZEITREIHENSTATISTIK -----
1510 PRINTCHR$(147):PRINTTAB(10)"  ZEITREIHENSTATISTIK:"
      :PRINT:PRINT
1520 GOSUB 8500:REM MENUE2
1530 GOSUB 6600:REM ENDE
2500 REM UNIVARIATE STATISTIK -----
2510 PRINTCHR$(147):PRINTTAB(10)"  UNIVARIATE STATISTIK:"
2520 PRINT:PRINTTAB(11)"(NUR EIN MERKMAL)":PRINT:PRINT
2530 GOSUB 8500:REM MENUE2
2540 GOSUB 6600:REM ENDE
3000 REM BIVARIATE STATISTIK -----
3010 PRINTCHR$(147):PRINTTAB(10)"  BIVARIATE STATISTIK:"
3020 PRINT:PRINTTAB(8)"(ZWEI MERKMALE ZUGLEICH)"
      :PRINT:PRINT
3030 GOSUB 8500:REM MENUE2
3040 GOSUB 6600:REM ENDE

```

```

3200 REM UP TASTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 PRINT:PRINT:RETURN
4000 REM UP DISK
4010 INPUT#15,EN,EM$,ET,ES
4020 IF EN=0 THEN RETURN
4030 PRINT:PRINT:PRINT
4040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
4050 PRINTEN;EM$;ET;ES
4060 PRINT:PRINT:PRINT
4070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
4080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
4090 PRINT:PRINT:PRINT
4100 PRINT"ENDE DES PROGRAMMS":END
5000 REM UP EINGABE *****
5010 FOR I=IA TO IE
5020 IF X(I)=EF% THEN KA=IA:KE=I-1:GOSUB 7200:GOTO 5050
5030 NEXT I
5040 KA=IA:KE=I-1:GOSUB 7200
5050 RETURN
6500 REM UP DATENDISK *****
6520 POKE 214,21:POKE 211,9:SYS 58640
6530 PRINT"DATENDISKETTE EINLEGEN"
6540 PRINTTAB(9)"UND LEERTASTE DRUECKEN:"
6550 GET A$:IF A$="" THEN 6550
6560 PRINTCHR$(147):RETURN
6600 REM UP PROGRAMMDISK UND ENDE *****
6610 PRINTCHR$(147)
6620 POKE 214,21:POKE 211,3:SYS 58640
6630 PRINT"PROGRAMMDISKETTE WIEDER EINLEGEN"
6640 PRINT" UND LEERTASTE DRUECKEN !:"
6650 GET A$:IF A$="" THEN 6650
6660 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE:"
6670 LOAD "STATISTIK",8

```

```

7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "■BITTE DIE LEERTASTE DRUECKEN !□"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7200 REM UP KONTROLLE *****
7210 IF KA>KE THEN 7390
7220 GOSUB 7400:FOR K=KA TO KE
7230 IF Z1%=3 THEN PRINTK,X(K),Y(K):GOTO 7250
7240 PRINTK,X(K)
7250 NEXT K
7260 PRINT:PRINT"■ALLE WERTE IN ORDNUNG (J/N)□ ":GOSUB3200
7270 IF A$="J" THEN 7390
7280 D$="FALSCHER WERT ":E$="(BITTE NR.UND RETURN)"
7290 IF Z1%=3 THEN D$="FALSCHER ZEILE "
7300 PRINTD$;E$;:INPUT Z%
7305 IF Z%<KA OR Z%>KE THEN PRINT:PRINT"FEHLER":GOTO 7300
7310 PRINT:PRINT"FEHLER":PRINT
7320 IF Z1%=3 THEN PRINT"NR. ";Z%,X(Z%),Y(Z%):GOTO 7340
7330 PRINT"NR. ";Z%,X(Z%)
7340 PRINT:PRINT"■KORREKTUR : □":PRINT"(EINGABE MIT
RETURN BEENDEN)"
7350 IF Z1%=3 THEN PRINT"(PAARE DURCH KOMMA TRENNEN)"
7360 IF Z1%=3 THEN INPUT" ";X(Z%),Y(Z%):GOTO 7380
7370 INPUT" ";X(Z%)
7380 GOTO 7220
7390 RETURN

```

```

7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"NR.", "X-WERT", "Y-WERT"
7420 FOR JJ=1 TO 26:PRINT"-";:NEXT JJ:PRINT:RETURN
8500 REM UP MENUE2 *****
8510 GOSUB 6500:REM DATENDISK
8520 PRINT:PRINT"NAME DER DATEI":INPUT "(UND RETURN) ";N$
8530 PRINTCHR$(147):PRINTTAB(8)"DATEI WIRD GELESEN"
8535 OPEN 15,8,15
8540 OPEN 8,8,8,N$+",S,R"
8545 GOSUB 4000
8550 IF Z1%=3 THEN 8590
8560 I=1
8570 INPUT#8,X(I):IF ST AND 64 THEN 8630
8580 I=I+1:GOTO 8570
8590 I=1
8600 INPUT#8,X(I)
8610 INPUT#8,Y(I):IF ST AND 64 THEN 8630
8620 I=I+1:GOTO 8600
8630 GOSUB 4000:CLOSE 8:CLOSE 15
8635 N1%=I:X(I+1)=EF%:Y(I+1)=EF%
8640 FOR J=1 TO INT(N1%/10)+1
8650 IA=(J-1)*10+1:IE=IA+9

```

```
8660 GOSUB 5000:REM EINGABE
8670 IF X(I)=EF% THEN 8690
8680 NEXT J
8690 PRINT:PRINT:PRINT
8700 PRINT"SOLL DIE DATEI WIEDER UNTER DEM NAMEN"
8710 PRINTN$;" GESPEICHERT WERDEN ? (J/N) ":GOSUB 3200
8720 IF A$="N" THEN GOSUB 6600:REM ENDE
8730 PRINT:PRINT:PRINTTAB(8)"DATEI WIRD GESPEICHERT"
8735 OPEN 15,8,15
8740 OPEN 8,8,8,"@:"+"N$+",S,W"
8745 GOSUB 4000
8750 FOR I= 1 TO N1%
8760 IF Z1%=3 THEN PRINT#8,X(I):PRINT#8,Y(I):GOTO 8780
8770 PRINT#8,X(I)
8775 GOSUB 4000
8780 NEXT I
8790 CLOSE 8:CLOSE 15:RETURN
```

## Beschreibung des Programms DSP

Satz 1000-1250 :

Vorbereiten des Nachladens, Überschrift, Erläuterungen (falls gewünscht) und Abwarten

Satz 1260-1270 :

Voreinstellungen und Dimensionierungen

Satz 1280-3030 :

Menü zur Auswahl der jeweiligen Statistik (Zeitreihenstatistik, univariate, bivariate Statistik) und Verzweigungen je nach Menüauswahl

Satz 3040 :

Beendigung des Programms

Satz 3200-3230 :

Unterprogramm zum Abwarten der Eingabe J oder N (für Ja oder Nein)

Satz 4000-4100 :

Unterprogramm zum Abfragen des Disketten-Fehlerkanals

Satz 5000-5050 :

Unterprogramm zur Übergabe von Daten an das Korrekturprogramm 7200 in Zehnerblöcken

Satz 6500-6560 :

Unterprogramm zur Ausgabe der Aufforderung, die Datendiskette einzulegen

Satz 6600-6670 :

Unterprogramm zur Ausgabe der Aufforderung, die Programmdiskette einzulegen und zum Nachladen des Programms STATISTIK (Ende des Programms DSP)

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 7200-7390 :

Unterprogramm zur Kontrolle und eventuellen Korrektur der Daten

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 8500-8790 :

Unterprogramm zum Einlesen der Daten von der Datendiskette (8520-8630) und zur Übergabe an das Korrekturprogramm (Unterprogramm 7200) in Zehnerblöcken (8640-8680);

danach wird die Möglichkeit geboten, den Datenbestand erneut unter dem alten Dateinamen zu speichern (8690-8780)

Programm DZA

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(10)"STATISTIK":
PRINT:PRINT:PRINT
1020 PRINT"PROGRAMM ZUR EINFACHEN STATISTISCHEN"
1030 PRINTTAB(8)"ANALYSE VON ZEITREIHEN."
1040 PRINT:PRINT:PRINT:PRINT
1050 PRINTTAB(9)"PROF.DR.W. VOSS, 1985"
1060 PRINT:PRINT:PRINT"DIE DATENEINGABE ERFOLGT
IN DIESEM PRO-"
1070 PRINT"GRAMM AUS EINER DISKETTENDATEI"
1090 NZ=81:EF%=-111:MV%=-99
1100 DIM Y(NZ)
1110 GOSUB 5000:REM EINGABE
1120 FOR I=1 TO N1%:IF Y(I)=MV% THEN GOSUB 7500:
REM INTERPOL
1130 NEXT I
1140 PRINTCHR$(147)
1142 GOSUB 7400:FORI=1 TO N1%:PRINTI,Y(I)
1144 IF I/10=INT(I/10) THEN GOSUB 7000:GOSUB 7400
1146 NEXT I
1150 GOSUB 7000:REM WARTEN
1160 M=1:GOSUB 8250:REM TRENDKQ
1170 PRINT:PRINT:PRINT"GRAPHIK (J/N) ? ":GOSUB 3200
1180 IF A$="N" THEN 1270
1190 GOSUB 7800:REM TRANSFORMIEREN
1200 GOSUB 7600:REM SCREEN
1210 GOSUB 7650:REM ACHSEN
1220 GOSUB 8000:REM REIHE
1230 M=0:GOSUB 8250:REM TRENDKQ
1240 FOR X=1 TO 300 STEP3:Y=A+B*X
1250 GOSUB 7700:NEXT X
1260 GOSUB 8150:REM TEXT
1270 GOSUB 4100:REM PROGRAMMDISK
1280 PRINT"ENDE":END
```

```

2000 REM UP DISK *****
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT:PRINT:PRINT
2040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
2080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
2090 PRINT:PRINT:PRINT
2100 PRINTTAB(10)"ENDE DES PROGRAMMS":END
3200 REM UP TATSTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 PRINT:PRINT:PRINT:RETURN
4000 REM UP DATENDISK *****
4020 POKE 214,22:POKE 211,9:SYS 58640
4030 PRINT"DATENDISKETTE EINLEGEN"
4040 PRINT"          UND LEERTASTE DRUECKEN"
4050 GET A$:IF A$="" THEN 4050
4060 PRINTCHR$(147):RETURN
4100 REM UP PROGRAMMDISK *****
4110 PRINTCHR$(147)
4120 POKE 214,22:POKE 211,6:SYS 58640
4130 PRINT"PROGRAMMDISKETTE EINLEGEN"
4140 PRINT"          UND LEERTASTE DRUECKEN !"
4150 GET A$:IF A$="" THEN 4150
4160 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
4170 LOAD "STATISTIK",8

```

```

5000 REM UP EINLESEN *****
5010 GOSUB 4000:REM DATENDISK
5020 PRINTCHR$(147):INPUT "NAME DER DATEI
      (UND RETURN) : ";N$
5030 PRINT TAB(8)"DATEI WIRD GELESEN"
5035 OPEN 15,8,15
5040 OPEN 8,8,8,N$+" ,S,R"
5045 GOSUB 2000
5050 I=1
5060 INPUT#8,Y(I)
5070 IF ST AND 64 THEN 5090
5080 I=I+1:GOTO 5060
5090 GOSUB 2000:CLOSE 8:CLOSE 15:N1%=I:Y(I+1)=EF%
5100 RETURN
7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "■BITTE DIE LEERTASTE DRUECKEN !■"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"■NR. ","WERT■"
7420 FOR J1=1 TO 16:PRINT"-";NEXT J1:PRINT:RETURN
7500 REM UP INTERPOL *****
7510 FOR J1=I TO I-1 STEP-1:IF J1=1 THEN 7540
7520 IF Y(J1)<>MV% THEN 7540
7530 NEXT J1
7540 FOR J2=I TO I+10:IF J2=N1% THEN 7570
7550 IF Y(J2)<>MV% THEN 7570
7560 NEXT J2
7570 Y(I)=(Y(J1)*(J2-I)+Y(J2)*(I-J1))/(J2-J1)
7580 RETURN
7600 REM UP SCREEN *****
7610 V=53248:POKE V+17,59:POKE V+24,24
7620 FOR I=1024 TO 2023:POKE I,14:NEXTI
7630 FOR I=8192 TO 16383:POKE I,0:NEXTI
7640 RETURN
7650 REM UP ACHSENKREUZ *****
7660 FOR X=1 TO 300 STEP5:Y=150:GOSUB 7700:NEXT X
7670 FOR Y=1 TO 150 STEP5:X=0:GOSUB 7700:NEXT Y
7680 RETURN
7700 REM UP PUNKT *****
7710 XL=8*INT(X/8)
7720 YL=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
7730 EX=2↑(7-INT((X/8-INT(X/8))*8))
7740 S=8192+XL+YL:POKE S,PEEK(S) OR EX
7750 RETURN

```

```

7800 REM UP NORM *****
7810 GOSUB 7900:REM MINMAX
7820 FOR I=1 TO N1%
7830 Y(I)=(150/(YG-YK))*(Y(I)-YK)
7840 Y(I)=150-Y(I):NEXT I:RETURN
7900 REM UP MINMAX *****
7910 YK=Y(1):YG=Y(1)
7920 FOR I=1 TO N1%:IF Y(I)<YK THEN YK=Y(I)
7930 IF Y(I)>YG THEN YG=Y(I)
7940 NEXT I:RETURN
8000 REM UP REIHE *****
8010 FOR I=1 TO N1%-1:X1=(200/N1%)*I:X2=(200/N1%)*(I+1)
8020 Y1=Y(I):Y2=Y(I+1)
8030 FOR X=X1 TO X2:Y=((Y2-Y1)*(X-X1))/(X2-X1)+Y1
8040 GOSUB 7700:REM PUNKT
8050 NEXT X
8060 NEXT I
8070 RETURN
8150 REM UP TEXT *****
8160 FOR J=1 TO 2000:NEXT J
8170 POKE V+17,155:POKE V+24,21
8180 RETURN

```

```

8250 REM UP TRENDKQ *****
8260 S1=0:S2=0:S3=0:S4=0
8270 FOR I=1 TO N1:X=(200/N1)*I:IF M=1THEN X=I
8275 S1=S1+X*Y(I):S2=S2+X:S3=S3+Y(I):S4=S4+X*X:NEXT I
8280 B=(N1*S1-S2*S3)/(N1*S4-S2*S2)
8290 A=S3/N1-B*(S2/N1)
8300 B=INT(B*100+.5)/100:A=INT(A*100+.5)/100
8310 IF M=0 THEN 8460
8320 PRINT:PRINT:PRINT"PARAMETER DER TRENDLINIE :":PRINT
8330 PRINT"ACHSENABSCHNITT = ";A
8340 PRINT"STEIGUNG      = ";B
8350 PRINT:PRINT:PRINT"PROGNOSE (J/N) ?":GOSUB 3200:
    REM TASTATUR
8360 IF A*="N" THEN 8460
8370 PRINT:PRINT"BEI DER ANGABE DER GEWUENSCHTEN ZEIT"
8380 PRINT"IST ZU BEACHTEN, DASS DER ERSTE WERT"
8390 PRINT"MIT 1, DER ZWEITE MIT 2 USW. CODIERT"
8400 PRINT"IST."
8410 PRINT:PRINT:INPUT "GEWUENSCHTER ZEITWERT : ";X
8420 Y=A+B*X
8430 PRINT:PRINT"ZEITPUNKT      : ";X
8440 PRINT"PROGNOSEWERT : ";Y;"?"
8450 PRINT:PRINT:GOTO 8350
8460 RETURN

```

## Beschreibung des Programms DZA

Satz 1000-1080 :

Vorbereitung des Nachladens, Überschrift und Erläuterungen

Satz 1090-1100 :

Voreinstellungen und Dimensionierung

Satz 1110 :

Einlesen der Daten durch Sprung ins Unterprogramm 5000)

Satz 1120-1130 :

Interpolation fehlender Werte, falls dies erforderlich ist (Unterprogramm 7500)

Satz 1140-1150 :

Ausgabe der Daten in Blöcken

Satz 1160 :

Berechnung des Trends mit der Methode der kleinsten Quadrate (Unterprogramm 8250)

Satz 1170-1260 :

Ausgabe einer graphischen Darstellung, falls dies gewünscht wird, nach Transformieren der Daten (Unterprogramm 7800) :

Vorbereiten des Bildschirms für die hochauflösende Graphik (Unterprogramm 7600), Zeichnen eines Achsenkreuzes (Unterprogramm 7650), Zeichnen der Zeitreihe (Unterprogramm 8000), Neuberechnung der Pa-

parameter der Trendlinie aus den transformierten Werten (Unterprogramm 8250) mit Unterdrückung der Parameterausgabe (Steuerung über die Hilfsvariable M) und Zeichnen der Trendlinie (1240-1250)

Satz 1270-1280 :

Beendigung des Programms

Satz 2000-2100 :

Unterprogramm zum Abfragen des Disketten-Fehlerkanals

Satz 3200-3230 :

Unterprogramm zum Abwarten der J- oder N-Eingabe

Satz 4000-4060 :

Unterprogramm zur Aufforderung, die Datendiskette einzulegen

Satz 4100-4170 :

Unterprogramm zur Aufforderung, die Programmdiskette einzulegen und zum Nachladen des Programms STATISTIK (Ende des Programms DZA)

Satz 5000-5100 :

Unterprogramm zum Einlesen der Daten von der Datendiskette

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 7500-7580 :

Unterprogramm zur linearen Interpolation fehlender  
Zeitreihenwerte

Satz 7600-7640 :

Unterprogramm zur Vorbereitung des Bildschirms für  
die hochauflösende Graphik

Satz 7650-7680 :

Unterprogramm zum Zeichnen eines Achsenkreuzes

Satz 7700-7750 :

Unterprogramm zum Zeichnen eines Punktes in hoch-  
auflösender Graphik

Satz 7800-7840 :

Unterprogramm zum Normieren der Zeitreihenwerte,  
um den Bildschirm in der graphischen Darstellung  
auszunutzen

Satz 7900-7940 :

Unterprogramm zum Aufsuchen des größten und des  
kleinsten Zeitreihenwertes

Satz 8000-8070 :

Unterprogramm zum Zeichnen der Zeitreihe

Satz 8150-8180 :

Unterprogramm zur Rückkehr in den Textmodus

Satz 8250-8460 :

Unterprogramm zur Berechnung der Parameter der Trendlinie (8260-8300); Ausgabe derselben (wenn M=1) (8310-8340) und Angebot einer Prognosemöglichkeit (8350-8450)

Programm DUT

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(10)"STATISTIK":
PRINT:PRINT:PRINT
1020 PRINT"PROGRAMM ZUR UNIVARIATEN TABELLARISCHEN"
1030 PRINTTAB(8)"ANALYSE (NUR 1 MERKMAL)":PRINT:PRINT
1040 PRINTTAB(9)"PROF.DR.W. VOSS, 1985"
1050 PRINT:PRINT:PRINT"DIE DATEN WERDEN AUS
EINER DISKETTEN-"
1060 PRINTTAB(13)"DATEI GELESEN"
1070 GOSUB 7000:REM WARTEN
1080 N%=201:EF%=-111:MV%=-99
1090 DIM Y(N%),HZ(10),HR%(10),YE(10)
1100 GOSUB 5000:REM EINGABE
1110 PRINTCHR$(147):PRINT"AUSGABE DER URLISTE ? (J/N)"
:GOSUB 3200
1120 IF A$="N" THEN 1160
1130 GOSUB 7400:FOR I=1 TO N1%:PRINTI,Y(I)
1140 IF I/10 = INT(I/10) THEN GOSUB 7000:GOSUB 7400
1150 NEXT I
1160 GOSUB 7000:REM WARTEN
1170 GOSUB 7900:REM MINMAX
1180 PRINTCHR$(147):PRINT"SIND DIE WERTE IM BEREICH
ZWISCHEN 0"
1190 PRINT"UND 9 GANZZAHLIG CODIERT (J/N) ?":GOSUB 3200
1200 IF A$="N" THEN 1240
1210 IF YK<0 OR YG>9 THEN PRINT:PRINT"FEHLER":GOSUB 7000
:GOTO 1240
1220 GOSUB 4000:REM DISKRETE HV
1230 GOTO 1250
1240 GOSUB 4500:REM STETIGE HV
1250 GOSUB 3100
1260 END
```

```

2000 REM UP DISK *****
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT:PRINT:PRINT
2040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
2080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
2090 PRINT:PRINT:PRINT
2100 PRINTTAB(10)"ENDE DES PROGRAMMS":END
3000 REM UP DATENDISK *****
3010 PRINTCHR$(147)
3020 POKE 214,22:POKE 211,9:SYS 58640
3030 PRINT"DATENDISKETTE EINLEGEN"
3040 PRINTTAB(9)"UND LEERTASTE DRUECKEN"
3050 GET A$:IF A$="" THEN 3050
3060 PRINTCHR$(147):RETURN
3100 REM UP PROGRAMMDISK *****
3110 PRINTCHR$(147)
3120 POKE 214,22:POKE 211,7:SYS 58640
3130 PRINT"PROGRAMMDISKETTE EINLEGEN"
3140 PRINTTAB(8)"UND LEERTASTE DRUECKEN"
3150 GET A$:IF A$="" THEN 3150
3160 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
3170 LOAD "STATISTIK",8

```

```

3200 REM UP TASTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 RETURN
4000 REM UP DISKRETE HV *****
4010 PRINT CHR$(147):PRINTTAB(10)"MOMENT BITTES"
4020 FOR I=1 TO N1%:FOR J=YK TO YG
4030 IF Y(I)=MV% THEN MV=MV+1:GOTO 4060
4040 IF Y(I)=J THEN H%(J)=H%(J)+1:GOTO 4060
4050 NEXT J
4060 NEXT I
4070 N=N1%-MV
4080 FOR J=YK TO YG:HR%(J)=(H%(J)/N)*100:HR%(J)=INT(HR%(J)
*10+.5)/10:NEXT J
4090 B$="WERT":J1=YK:J2=YG:M9=0:GOSUB 8000
4100 S=0:GOSUB 7000:REM WARTEN
4110 RETURN
4500 REM UP STETIGE HV *****
4505 PRINTCHR$(147)
4510 PRINT:PRINT:PRINT"KLEINSTER WERT = ";YK:D=YG-YK
4520 PRINT"GROESSTER WERT = ";YG:PRINT"DIFFERENZ = ";D
4530 PRINT:PRINT:PRINT
4580 INPUT "KLASSENBREITE (RETURN): ";DY
4590 IF DY<D/10 OR DY>D/2 THEN PRINT:PRINT"NICHT SINVOLL"
:GOTO 4510
4600 PRINT "KLASSIFIZIERUNGSANFANG: ":INPUT "(UND RETURN)
";SY
4610 IF SY>YK THEN PRINT:PRINT"NICHT ZULAESSIG":GOTO 4510
4620 FOR J=0 TO 9:YE(J)=SY+(J+1)*DY:NEXT J
4630 MV=0:FOR J=0 TO 9:H%(J)=0:HR%(J)=0:NEXT J
4640 FOR I=1 TO N1%:IF Y(I)=MV% THEN MV=MV+1:GOTO 4670
4650 FOR J=0 TO 9:IF Y(I)<YE(J) THEN
H%(J)=H%(J)+1:GOTO 4670
4660 NEXT J
4670 NEXT I
4680 N=N1%-MV
4690 FOR J=0 TO 9:HR%(J)=(H%(J)/N)*100:HR%(J)=
INT(HR%(J)*10+.5)/10:NEXT J
4700 B$="KLASSE":J1=0:J2=9:M9=1:GOSUB 8000
4710 GOSUB 7000
4720 RETURN

```

```

5000 REM UP EINGABE *****
5010 PRINTCHR$(147):INPUT "NAME DER DATEI (UND RETURN) ";N$
5020 GOSUB 3000:REM DATENDISK
5030 PRINTTAB(8)"DATEI WIRD GELESEN"
5035 OPEN 15,8,15
5040 OPEN 8,8,8,N$+",S,R"
5045 GOSUB 2000
5050 I=1
5060 INPUT#8,Y(I)
5070 IF ST AND 64 THEN 5090
5080 I=I+1:GOTO 5060
5090 GOSUB 2000:CLOSE 8:CLOSE 15
5100 N1%=I:Y(I+1)=EF%:RETURN
7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "■BITTE DIE LEERTASTE DRUECKEN !■"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"■NR. ","WERT■"
7420 FOR J1=1 TO 16:PRINT"-";:NEXT J1:PRINT:RETURN
7900 REM UP MINMAX *****
7910 YK=Y(1):YG=Y(1)
7920 FOR I=1 TO N1%:IF Y(I)=MV%THEN 7950
7930 IF Y(I)<YK THEN YK=Y(I)
7940 IF Y(I)>YG THEN YG=Y(I)
7950 NEXT I:RETURN

```

```

8000 REM UP TABELLE *****
8010 PRINTCHR$(147)
8020 PRINT"=";B$;TAB(23)"HAEUFIGKEIT":PRINTTAB(20)"ABS.
REL.(%)";"
8030 FOR I=1 TO 40:PRINT"-";:NEXT I
8040 FOR J=J1 TO J2
8050 IF M9=0 THEN PRINTJ,,H%(J),HR%(J):GOTO 8065
8060 PRINTYE(J)-DY;" B.U. ";YE(J),H%(J),HR%(J)
8065 NEXT J
8070 FOR I=1 TO 40:PRINT"-";:NEXTI
8080 PRINT"SUMME";TAB(26)N
8090 FOR I=1 TO 40:PRINT"="";:NEXTI
8100 PRINT"M.V. = ";MV
8110 S=0:FOR I=1 TO N1%:IF Y(I)=MV% THEN 8130
8120 S=S+Y(I)
8130 NEXT I
8140 AM=S/N:PRINT:PRINT"MITTELWERT = ";AM
8150 S=0:FOR I=1 TO N1%:IF Y(I)=MV% THEN 8170
8160 S=S+(Y(I)-AM)2
8170 NEXT I
8180 PRINT:PRINT"STANDARDABWEICHUNG = ";SQR(S/N);" "
8190 RETURN

```

**Beschreibung des Programms DUT**

**Satz 1000-1070 :**

**Vorbereitung des Nachladens, Überschrift, Erläuterungen und Abwarten**

**Satz 1080-1090 :**

**Voreinstellungen und Dimensionierung**

**Satz 1100 :**

**Einlesen der Daten von Diskette (Unterprogramm 5000)**

**Satz 1110-1160 :**

**Ausgabe des Datenbestands, wenn dies gewünscht ist**

**Satz 1170 :**

**Bestimmung des kleinsten und des größten Wertes im Unterprogramm 7900**

**Satz 1180-1230 :**

**Ausgabe einer diskreten Häufigkeitstabelle, wenn dies zulässig ist (Unterprogramm 4000)**

**Satz 1240-1260 :**

**Ausgabe einer stetigen Häufigkeitstabelle (Unterprogramm 4500) und Beendigung des Programms**

**Satz 2000-2100 :**

**Unterprogramm zum Abfragen des Disketten-Fehlerkanals**

Satz 3000-3060 :

Unterprogramm zur Aufforderung, die Datendiskette einzulegen

Satz 3100-3170 :

Unterprogramm zur Aufforderung, die Programmdiskette einzulegen, Nachladen des Programms STATISTIK (Ende des Programms DUT)

Satz 3200-3230 :

Unterprogramm zum Abwarten der J- oder N-Eingabe

Satz 4000-4110 :

Unterprogramm zur Erstellung einer diskreten Häufigkeitsverteilung, wenn die Merkmalswerte zwischen 0 und 9 (Maximalgrenzen) ganzzahlig codiert sind und Ausgabe der statistischen Maßzahlen (Unterprogramm 8000)

Satz 4500-4720 :

Erzeugung einer stetigen Häufigkeitsverteilung, d. h. einer Verteilung durch Klassifizierung und Ausgabe der statistischen Maßzahlen (Unterprogramm 8000)

Satz 5000-5100 :

Unterprogramm zum Einlesen der Daten von der Datendiskette

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 7900-7950 :

Unterprogramm zur Bestimmung des kleinsten und des größten Merkmalswertes

Satz 8000-8190 :

Unterprogramm zur Ausgabe der Häufigkeitstabelle und zur Bestimmung der statistischen Maßzahlen und Ausgabe derselben

Programm DUG

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(10)"S T A T I S T I K":
PRINT:PRINT:PRINT
1020 PRINT" PROGRAMM FUER UNIVARIATE GRAPHIKEN."
1050 PRINTTAB(9)"PROF.DR.W. VOSS, 1985"
1060 PRINT:PRINT:PRINT" DIE DATEN WERDEN
AUS EINER DISKETTEN-"
1070 PRINTTAB(13)"DATEI GELESEN."
1080 GOSUB 7000:REM WARTEN
1090 N%=201:EF%=-111:MV%=-99
1100 DIM Y(N%),H%(10),HR%(10),YE(10)
1110 GOSUB 5000:REM EINGABE
1120 PRINTCHR$(147):PRINT"AUSGABE DER URLISTE ?
(J/N)":GOSUB 3200:REM TASTATUR
1130 IF A*="N" THEN 1170
1140 GOSUB 7400:FOR I=1 TO N1%:PRINTI,Y(I)
1150 IF I/10 = INT(I/10) THEN GOSUB 7000:GOSUB 7400
1160 NEXT I
1170 GOSUB 7000:REM WARTEN
1200 GOSUB 7900:REM MINMAX
1210 PRINTCHR$(147):PRINT"SIND DIE WERTE IM BEREICH
ZWISCHEN 0"
1220 PRINT"UND 9 GANZZAHLIG CODIERT ? (J/N)":GOSUB 3200
1260 IF A*="N" THEN 1300
1270 IF YK<0 OR YG>9 THEN PRINT:PRINT"FEHLER":GOSUB
7000:GOTO 1300
1280 GOSUB 4000:REM DISKRETE HV
1290 GOTO 1310
1300 GOSUB 4500:REM STETIGE HV
1310 GOSUB 3100
1320 END
```

```

2000 REM UP DISK *****
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT:PRINT:PRINT
2040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
2080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
2090 PRINT:PRINT:PRINT
2100 PRINTTAB(10)"ENDE DES PROGRAMMS":END
3000 REM UP DATENDISK *****
3010 PRINTCHR$(147)
3020 POKE 214,22:POKE 211,9:SYS 58640
3030 PRINT"DATENDISKETTE EINLEGEN"
3040 PRINTTAB(9)"UND LEERTASTE DRUECKEN:"
3050 GET A$:IF A$="" THEN 3050
3060 PRINTCHR$(147):RETURN
3100 REM UP PROGRAMMDISK *****
3110 PRINTCHR$(147)
3120 POKE 214,22:POKE 211,7:SYS 58640
3130 PRINT"PROGRAMMDISKETTE EINLEGEN"
3140 PRINTTAB(8)"UND LEERTASTE DRUECKEN:"
3150 GET A$:IF A$="" THEN 3150
3160 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
3170 LOAD "STATISTIK",8

```

```

3200 REM UP TASTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 RETURN
4000 REM UP DISKRETE HV *****
4010 PRINT CHR$(147):PRINTTAB(10)"MOMENT BITTE"
4020 FOR I=1 TO N1$:FOR J=YK TO YG
4030 IF Y(I)=MV% THEN MV=MV+1:GOTO 4060
4040 IF Y(I)=J THEN H%(J)=H%(J)+1:GOTO 4060
4050 NEXT J
4060 NEXT I
4070 N=N1%-MV
4080 HG=0:FOR J=YK TO YG:IF H%(J)>HG THEN HG=H%(J)
4090 NEXT J
4100 FOR J=YK TO YG:H%(J)=H%(J)*(14/HG):NEXT J
4110 PRINTCHR$(147):PRINT"STABDIAGRAMM":PRINT:ZE=17
4120 FOR J=YK TO YG:ZA=17-H%(J):S=J*3+8
4130 FOR Z=ZA TO ZE:A=1024+S+Z*40:POKE A,66:NEXT Z:NEXT J
4140 FOR S=8 TO 32:A=1024+S+Z*40:POKE A,64:NEXT S
4150 Z=19:FOR J=YK TO YG:S=J*3+8:A=1024+S+Z*40:POKE A,48+J:
NEXT J
4160 S=0:GOSUB 7000:REM WARTEN
4170 RETURN

```

```

4500 REM UP STETIGE HV *****
4505 PRINTCHR$(147)
4510 PRINT:PRINT:PRINT"KLEINSTER WERT = ";YK:D=YG-YK
4520 PRINT"GROSSTER WERT = ";YG:PRINT"DIFFERENZ      = ";D
4530 PRINT:PRINT:PRINT
4580 INPUT "KLASSENBREITE (RETURN): ";DY
4590 IF DY<D/10 OR DY>D/2 THEN PRINT:PRINT"NICHT SINNVOLL":
      GOTO 4510
4600 PRINT "KLASSIFIZIERUNGSANFANG: ":INPUT "(UND RETURN)
      ";SY
4610 IF SY>YK THEN PRINT:PRINT"NICHT ZULAESSIG":GOTO 4510
4620 FOR J=0 TO 9:YE(J)=SY+(J+1)*DY:NEXT J
4630 MV=0:FOR J=0 TO 9:H%(J)=0:HR%(J)=0:NEXT J
4640 FOR I=1 TO N1%:IF Y(I)=MV% THEN MV=MV+1:GOTO 4670
4650 FOR J=0 TO 9:IF Y(I)<YE(J) THEN H%(J)=H%(J)+1:
      GOTO 4670
4660 NEXT J
4670 NEXT I
4680 N=N1%-MV
4880 HG=0:FOR J=0 TO 9:IF H%(J)>HG THEN HG=H%(J)
4890 NEXT J
4900 FOR J=0 TO 9:H%(J)=H%(J)*(15/HG):NEXTJ
4910 PRINTCHR$(147):PRINTTAB(6)"HISTOGRAMM":PRINT:PRINT
4920 ZE=17:FOR J= 0 TO 9:ZA=18-H%(J):S=J*3+6
4930 IF ZA>=ZE THEN 4950
4940 FOR Z=ZA TO ZE:A=1024+S+Z*40:POKE A,160:POKE A+1,
      160:NEXT Z
4950 NEXT J
4960 Z=18:FOR S=3 TO 36:A=1024+S+Z*40:POKE A,64:NEXT S
4970 Z=19:FOR J=0 TO 9:S=J*3+6:A=1024+S+Z*40:POKE A,
      48+J:NEXT J
4980 GOSUB 7000
4990 RETURN

```

```

5000 REM UP EINGABE *****
5010 PRINTCHR$(147):INPUT "NAME DER DATEI
      (UND RETURN) ";N$
5020 GOSUB 3000:REM DATENDISK
5030 PRINTTAB(8)"DATEI WIRD GELESEN"
5035 OPEN 15,8,15
5040 OPEN 8,8,8,N$+"",S,R"
5045 GOSUB 2000
5050 I=1
5060 INPUT#8,Y(I)
5070 IF ST AND 64 THEN 5090
5080 I=I+1:GOTO 5060
5090 GOSUB 2000:CLOSE 8:CLOSE 15
5100 N1%=I:Y(I+1)=EF%:RETURN
7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "■BITTE DIE LEERTASTE DRUECKEN !□"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"■NR.", "WERT:□"
7420 FOR J1=1 TO 16:PRINT"-";:NEXT J1:PRINT:RETURN
7900 REM UP MINMAX *****
7910 YK=Y(1):YG=Y(1)
7920 FOR I=1 TO N1%:IF Y(I)=MV%THEN 7950
7930 IF Y(I)<YK THEN YK=Y(I)
7940 IF Y(I)>YG THEN YG=Y(I)
7950 NEXT I:RETURN

```

## Beschreibung des Programms DUG

### Satz 1000-1080 :

Vorbereitung des Nachladens, Überschrift, Erläuterungen und Abwarten

### Satz 1090-1110 :

Voreinstellungen, Dimensionierungen und Einlesen der Daten durch Sprung ins Unterprogramm 5000

### Satz 1120-1170 :

Ausgabe der Ausgangsdaten in Zehnerblöcken, falls diese Ausgabe gewünscht wird

### Satz 1200 :

Bestimmung des kleinsten und des größten Merkmalswertes durch Sprung ins Unterprogramm 7900

### Satz 1210-1290 :

Verzweigung in den Programmbereich, der für diskrete Daten vorgesehen ist (Unterprogramm 4000), es sei denn, die betrachtete Variable ist im Gegensatz zur Angabe des Benutzers (Satz 1210-1220) doch nicht ganzzahlig-diskret

### Satz 1300-1320 :

Verzweigung in den Programmbereich, der für alle anderen Variablen (außer ganzzahlig-diskret) vorgesehen ist (Unterprogramm 4500) und Beendigung des Programms

Satz 2000-2100 :

Unterprogramm zum Abfragen des Diskettenfehlerkanals

Satz 3000-3060 :

Unterprogramm mit der Aufforderung, die Datendiskette einzulegen

Satz 3100-3170 :

Unterprogramm zur Aufforderung, die Programmdiskette einzulegen und Nachladen des Programms "STATISTIK" (Ende des Programms DUG)

Satz 4000-4170 :

Unterprogramm zur Behandlung des ganzzahlig-diskreten Falls (Merkmalswerte zwischen 0 und 9) : Bestimmung der Häufigkeiten (4020-4060), der maximalen Häufigkeit (4080-4090), Zeichnen des Stabdiagramms (4110-4150), Abwarten (4160) und Rücksprung (4170)

Satz 4500-4990 :

Unterprogramm zur Behandlung einer beliebigen Variablen : Ausgabe des kleinsten und des größten Werts und der Differenz zwischen beiden (4505 - 4530), Anforderung von Klassenbreite und Klassifizierungsanfang mit Zurückweisung unzulässiger oder unsinniger Werte (4580-4610), Bestimmung der Klassenendpunkte (4620), der Häufigkeiten und der relativen Häufigkeiten (4630-4670), Bestimmung der Maximalhäufigkeit (4880-4890), Normierung der Häufigkeiten (4900), Ausgabe des Histogramms (4910 - 4970), Abwarten (4980) und Rücksprung (4990)

Satz 5000-5100 :

Unterprogramm zur Eingabe der Daten von der Datendiskette mit Festlegung der Anzahl der gelesenen Werte

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste be-  
tätigt wird

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüber-  
schrift

Satz 7900-7950 :

Unterprogramm zur Bestimmung des kleinsten und des  
größten Wertes im Datenbestand

**Programm DBT**

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(11)"#S T A T I S T I K#":
    PRINT:PRINT:PRINT
1020 PRINT" PROGRAMM ZUR EINFACHEN STATISTISCHEN"
1030 PRINT" ANALYSE BIVARIATER VERTEILUNGEN OHNE"
1040 PRINT TAB(8)"GRAPHISCHE DARSTELLUNG":PRINT:PRINT:PRINT
1050 PRINTTAB(9)"PROF.DR.W. VOSS, 1985"
1060 PRINT:PRINT:PRINT" DIE DATEN WERDEN AUS EINER
    DISKETTEN-"
1070 PRINTTAB(13)"DATEI GELESEN"
1080 GOSUB 7000:REM WARTEN
1090 N%=201:EF%=-111:MV%=-99
1100 DIM X(N%),Y(N%),S(20),T(10,10),H(10,10)
1110 GOSUB 5000:REM EINGABE
1120 PRINTCHR$(147):PRINT"AUSGABE DES DATENBESTANDS ?
    (J/N) ":GOSUB 3200
1130 IF A$="N" THEN PRINTCHR$(147):GOTO 1180
1140 GOSUB 7400:REM UEBERSCHRIFT
1150 FOR I=1 TO N1%:PRINTI,X(I),Y(I)
1160 IF I/10=INT(I/10) THEN GOSUB 7000:GOSUB 7400
1170 NEXT I:GOSUB 7000
1180 PRINT"WENN BEIDE VARIABLEN IM WERTEBEREICH"
1190 PRINT"ZWISCHEN 0 UND 9 GANZZAHLIG CODIERT "
1200 PRINT"SIND, KANN EINE KREUZTABELLE AUSGEGE-"
1205 PRINT"WERDEN, ANDERNFALLS WERDEN NUR MASSZAH-"
1210 PRINT"LEN BERECHNET.":PRINT:PRINT
1220 PRINT"IST DIESE BEDINGUNG ERFUELLT ? (J/N)":GOSUB 3200
1230 IF A$="N" THEN M=1:GOTO1250
1240 GOSUB 8000:GOTO 1280
1250 PRINTCHR$(147)
1260 PRINT:PRINT:PRINT
1270 GOSUB 4000:REM KORR
1280 GOSUB 3100:REM PROGRAMMDISK
1290 END
```

```

2000 REM UP DISK *****
2010 INPUT#15,EN,EM#,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT:PRINT:PRINT
2040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
2050 PRINTEN;EM#;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
2080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
2090 PRINT:PRINT:PRINT
2100 PRINTTAB(10)"ENDE DES PROGRAMMS":END
3000 REM UP DATENDISK *****
3010 PRINTCHR$(147):POKE 214,22:POKE 211,9:SYS 58640
3020 PRINT"DATENDISKETTE EINLEGEN":PRINTTAB(9)"UND
LEERTASTE DRUECKEN"
3030 GET A$:IF A#="" THEN 3030
3040 PRINTCHR$(147):RETURN
3100 REM UP PROGRAMMDISK *****
3110 PRINTCHR$(147):POKE 214,22:POKE 211,7:SYS 58640
3120 PRINT"PROGRAMMDISKETTE EINLEGEN":PRINTTAB(8)"UND
LEERTASTE DRUECKEN"
3130 GET A$:IF A#="" THEN 3130
3140 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
3150 LOAD "STATISTIK",8

```

```

3200 REM UP TASTATUR *****
3210 GET A$:IF A$="" THEN 3210
3220 IF A$<>"J" AND A$<>"N" THEN 3210
3230 RETURN
4000 REM UP KORR (R UND C) *****
4010 PRINTTAB(8)"# MOMENT BITTE, ICH RECHNE#":
      PRINT:PRINT:PRINT:IF M=1 THEN 4120
4020 FORI=0T09:FORJ=0T09:S(I)=S(I)+H(I,J):NEXTJ:NEXTI
4030 FORJ=0T09:FORI=0T09:S(10+J)=S(10+J)+H(I,J):NEXTI:NEXTJ
4040 FORI=0T09:FORJ=0T09:IFS(I)=OORS(10+J)=0THEN 4060
4050 T(I,J)=(S(I)*S(10+J))/(N1%-MV)
4060 NEXTJ:NEXTI
4070 S=0:FORI=0T09:FORJ=0T09:IF T(I,J)=0THEN 4090
4080 D=H(I,J)-T(I,J):D=D*D:D=D/T(I,J):S=S+D
4090 NEXT J:NEXT I
4100 C=SQR(S/(S+N1%-MV)):C=INT(C*100+.5)/100
4110 PRINT:PRINT:PRINT"KONTINGENZKOEFFIZIENT      C = ";C
4120 GOSUB 4200:REM REGRESSION
4130 PRINT:PRINT"KORRELATIONSKOEFFIZIENT  R = ";R
4140 PRINT:PRINT:PRINT"REGRESSIONSGERADE  : "
4150 PRINT:PRINT"    - ACHSENABSCHNITT  A = ";A
4160 PRINT:PRINT"    - STEIGUNG          B = ";B
4170 PRINT:PRINT:PRINT"M.V. (FEHLENDE FAELLE)    = ";MV
4180 GOSUB 7000:REM WARTEN
4190 RETURN
4200 REM UP REGRESSION *****
4210 S1=0:S2=0:S3=0:S4=0:S5=0:MV=0
4220 FOR I=1 TO N1%:IF X(I)=MV% OR Y(I)=MV% THEN
      MV=MV+1:GOTO 4240
4230 S1=S1+X(I)*Y(I):S2=S2+X(I):S3=S3+Y(I):S4=S4+X(I)
      *X(I):S5=S5+Y(I)*Y(I)
4240 NEXT I
4250 N=N1%-MV:Z=N*S1-S2*S3:NE=N*S4-S2*S2:B=Z/NE:
      A=S3/N-B*(S2/N)
4260 R=Z/SQR(NE*(N*S5-S3*S3))
4270 A=INT(A*100+.5)/100:B=INT(B*100+.5)/100:R=
      INT(R*100+.5)/100
4280 RETURN

```

```

5000 REM UP EINGABE *****
5010 GOSUB 3000:REM DATENDISK
5020 PRINTCHR$(147):INPUT "NAME DER DATEI (UND RETURN) ";N$
5030 PRINT:PRINT:PRINTTAB(8)"DATEI WIRD GELESEN"
5035 OPEN 15,8,15
5040 OPEN 8,8,8,N$+",S,R"
5045 GOSUB 2000
5050 I=1
5060 INPUT#8,X(I):INPUT#8,Y(I)
5070 IF ST AND 64 THEN 5090
5080 I=I+1:GOTO 5060
5090 GOSUB 2000:CLOSE 8:CLOSE 15
5100 N1%=I:X(I+1)=EF%:Y(I+1)=EF%:RETURN
7000 REM UP WARTEN *****
7010 POKE 214,23:POKE 211,5:SYS 58640
7020 PRINT "■BITTE DIE LEERTASTE DRUECKEN !□"
7030 GET A$:IF A$="" THEN 7030
7040 PRINTCHR$(147):RETURN
7400 REM UP UEBERSCHRIFT *****
7410 PRINTCHR$(147):PRINT"NR. ","X-WERT","Y-WERT□"
7420 FOR J=1 TO 26:PRINT"-";:NEXT J:PRINT:RETURN

```

```

7900 REM UP MINMAX *****
7910 XK=X(1):XG=X(1):YK=Y(1):YG=Y(1)
7920 FOR I=1 TO N1%
7930 IF X(I)=MV% OR Y(I)=MV% THEN 7980
7940 IF Y(I)<YK THEN YK=Y(I)
7950 IF Y(I)>YG THEN YG=Y(I)
7960 IF X(I)<XK THEN XK=X(I)
7970 IF X(I)>XG THEN XG=X(I)
7980 NEXT I
7990 RETURN
8000 REM UP KREUZTABELLE *****
8010 PRINTCHR$(147):PRINTTAB(5)"MOMENT BITTE,
      ICH RECHNE":GOSUB 7900
8020 IF XK<0 OR YK<0 OR XG>9 OR YG>9 THEN PRINT"TABELLE
      UNMOEGLICH":GOTO 8130
8030 FOR I=XK TO XG:FOR J=YK TO YG:H(I,J)=0:NEXT J:NEXT I
8040 FOR K=1 TO N1%:FOR I=XK TO XG:FOR J=YK TO YG
8050 IF X(K)=MV% OR Y(K)=MV% THEN MV=MV+1:GOTO 8080
8060 IF X(K)=I AND Y(K)=J THEN H(I,J)=H(I,J)+1:GOTO 8080
8070 NEXT J:NEXT I
8080 NEXT K
8090 PRINTCHR$(147):PRINT"V2 ";:FOR J=YK TO YG
8100 PRINTJ;" ";:NEXT J:PRINT:PRINT
8110 PRINT"V1":PRINT:FOR I=XK TO XG:PRINTI;" ";
8120 FOR J=YK TO YG:PRINTSTR$(H(I,J));" ";:NEXTJ:PRINT:
      NEXT I
8130 GOSUB 7000:GOSUB 4000:REM KORR
8140 RETURN

```

## Beschreibung des Programms DBT

Satz 1000-1080 :

Vorbereitung des Nachladens, Überschrift, Erläuterungen und Abwarten

Satz 1090-1100 :

Voreinstellungen und Dimensionierungen

Satz 1100-1170 :

Einlesen des Datenbestandes (Unterprogramm 5000) und Ausgabe, wenn dies gewünscht wird

Satz 1180-1240 :

Wenn beide Variablen im Wertebereich zwischen 0 und 9 ganzzahlig codiert sind, kann eine Kreuztabelle ausgegeben werden (Unterprogramm 8000)

Satz 1250-1290 :

Berechnung der Zusammenhangsmaße und Beendigung des Programms

Satz 2000-2100 :

Unterprogramm zum Abfragen des Disketten-Fehlerkanals

Satz 3000-3040 :

Unterprogramm zur Aufforderung, die Datendiskette einzulegen

Satz 3100-3150 :

Unterprogramm zur Aufforderung, die Programmdis-

kette einzulegen und zum Nachladen des Programms  
STATISTIK (Ende des Programms DBT)

Satz 3200-3230 :

Unterprogramm zur Eingabe von J oder N

Satz 4000-4190 :

Unterprogramm zur Berechnung des Kontingenzkoeffizienten C (dies unterbleibt, wenn  $M=1$ , d.h. wenn keine Kreuztabelle erstellt wurde), des Korrelationskoeffizienten R und der Parameter der Regressionsgeraden A und B im Unterprogramm 4200

Satz 4200-4280 :

Unterprogramm zur Bestimmung der Parameter der Regressionsgeraden A und B mit Hilfe der Methode der kleinsten Quadrate und zur Berechnung des Korrelationskoeffizienten R

Satz 5000-5100 :

Unterprogramm zum Einlesen der Daten von der Datendiskette

Satz 7000-7040 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 7400-7420 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 7900-7990 :

Unterprogramm zur Berechnung der Extremwerte im Datenbestand

Satz 8000-8140 :

Unterprogramm zur Ausgabe einer Kreuztabelle

**Programm DBG**

```
1000 POKE 45,PEEK(174):POKE 46,PEEK(175):CLR
1010 PRINTCHR$(147):PRINTTAB(11)"S T A T I S T I K":
PRINT:PRINT:PRINT
1020 PRINT" PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG"
1030 PRINT" EINER BIVARIATEN VERTEILUNG."
1040 PRINT:PRINT:PRINT:PRINT
1050 PRINTTAB(9)"PROF.DR.W. VOSS, 1985"
1060 PRINT:PRINT:PRINT" DIE DATEN WERDEN AUS EINER
DISKETTEN-"
1070 PRINT TAB(13)"DATEI GELESEN."
1080 GOSUB 2700:REM WARTEN
1090 N%=201:EF%=-111:MV%=-99
1100 DIM X(N%),Y(N%)
1110 GOSUB 5000:REM EINGABE
1130 GOSUB 2500:GOSUB 2700
1140 GOSUB 3200:REM MINMAX
1150 GOSUB 3100:REM TRANSFORMIEREN
1160 GOSUB 2800:REM SCREEN
1170 GOSUB 2900:REM Achsen
1180 FOR I=1TON1%:IF X(I)=MV% OR Y(I)=MV% THEN 1210
1190 X=X(I):Y=Y(I):GOSUB 3000:REM PUNKT
1200 X=X+1:GOSUB 3000:Y=Y+1:GOSUB 3000:X=X-1:GOSUB3000
1210 NEXT I
1220 M=1:GOSUB 2500:REM REGRESSION
1230 FOR X=1 TO 300:Y=A+B*X:GOSUB 3000:NEXT X
1240 GOSUB 3300: REM TEXT
1250 PRINTCHR$(147):GOSUB 2300:PRINT"ENDE DER AUSGABE":END
```

```

2000 REM UP DISK *****
2010 INPUT#15,EN,EM$,ET,ES
2020 IF EN=0 THEN RETURN
2030 PRINT:PRINT:PRINT
2040 PRINTTAB(12)"DISKETTENFEHLER !":PRINT:PRINT
2050 PRINTEN;EM$;ET;ES
2060 PRINT:PRINT:PRINT
2070 PRINT"NEUSTART NACH FEHLERBESEITIGUNG DURCH"
2080 PRINT"LADEN DES PROGRAMMS STATISTIK !"
2090 PRINT:PRINT:PRINT
2100 PRINTTAB(10)"ENDE DES PROGRAMMS":END
2200 REM UP DATENDISK *****
2210 PRINTCHR$(147):POKE 214,22:POKE 211,9:SYS 58640
2220 PRINT"DATENDISKETTE EINLEGEN":PRINTTAB(9)"UND
LEERTASTE DRUECKEN"
2230 GET A$:IF A$="" THEN 2230
2240 PRINTCHR$(147):RETURN
2300 REM UP PROGRAMMDISK *****
2310 PRINTCHR$(147):POKE 214,22:POKE 211,7:SYS 58640
2320 PRINT"PROGRAMMDISKETTE EINLEGEN":PRINTTAB(9)"UND
LEERTASTE DRUECKEN"
2330 GET A$:IF A$="" THEN 2330
2340 PRINTCHR$(147):PRINTTAB(12)"MOMENT BITTE"
2350 LOAD "STATISTIK",8
2400 REM UP TASTATUR *****
2410 GET A$:IF A$="" THEN 2410
2420 IF A$<>"J" AND A$<>"N" THEN 2410
2430 RETURN

```

```

2500 REM UP REGRESSION *****
2503 IF M=1 THEN 2510
2505 PRINTCHR$(147):PRINTTAB(6)"MOMENT BITTE, ICH RECHNE":
PRINT:PRINT:PRINT
2510 S1=0:S2=0:S3=0:S4=0:S5=0:MV=0
2520 FOR I=1 TO N1%:IF X(I)=MV% DR Y(I)=MV%
THEN MV=MV+1:GOTO 2540
2530 S1=S1+X(I)*Y(I):S2=S2+X(I):S3=S3+Y(I):S4=S4+X(I)
*X(I):S5=S5+Y(I)*Y(I)
2540 NEXT I
2550 N=N1%-MV:Z=N*S1-S2*S3:NE=N*S4-S2*S2:
B=Z/NE:A=S3/N-B*(S2/N)
2560 R=Z/SQR(NE*(N*S5-S3*S3))
2570 A=INT(A*100+.5)/100:B=INT(B*100+.5)
/100:R=INT(R*100+.5)/100
2580 IF M=1 THEN 2630
2590 PRINT:PRINT:PRINT
2600 PRINT:PRINT"ORDINATENABSCHNITT A = ";A
2610 PRINT:PRINT"STEIGUNG B = ";B
2620 PRINT:PRINT"KORRELATIONSKOEFFIZIENT R = ";R
2630 RETURN
2700 REM UP WARTEN *****
2710 POKE 214,23:POKE 211,5:SYS 58640
2720 PRINT "■BITTE DIE LEERTASTE DRUECKEN !□"
2730 GET A$:IF A$="" THEN 2730
2740 PRINTCHR$(147):RETURN
2800 REM UP SCREEN *****
2810 V=53248:POKE V+17,59:POKE V+24,24
2820 FOR I=1024 TO 2023:POKE I,14:NEXTI
2830 FOR I=8192 TO 16383:POKE I,0:NEXTI
2840 RETURN
2900 REM UP ACHSENKREUZ *****
2910 FOR X=1 TO 300 STEP5:Y=150:GOSUB 3000:NEXT X
2920 FOR Y=1 TO 150 STEP5:X=0:GOSUB 3000:NEXT Y
2930 RETURN

```

```

3000 REM UP PUNKT *****
3010 XL=8*INT(X/8)
3020 YL=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
3030 EX=2↑(7-INT((X/8-INT(X/8))*8))
3040 S=8192+XL+YL:POKE S,PEEK(S) OR EX
3050 RETURN
3100 REM UP NORM *****
3110 GOSUB 3200:REM MINMAX
3120 FOR I=1 TO N1%:IF X(I)=MV% OR Y(I)=MV% THEN 3160
3130 X(I)=(200/(XG-XK))*(X(I)-XK)
3140 Y(I)=(150/(YG-YK))*(Y(I)-YK)
3150 Y(I)=150-Y(I)
3160 NEXT I:RETURN
3200 REM UP MINMAX *****
3210 XK=X(1):XG=X(1):YK=Y(1):YG=Y(1)
3220 FOR I=1 TO N1%
3230 IF X(I)=MV% OR Y(I)=MV% THEN 3280
3240 IF Y(I)<YK THEN YK=Y(I)
3250 IF Y(I)>YG THEN YG=Y(I)
3260 IF X(I)<XK THEN XK=X(I)
3270 IF X(I)>XG THEN XG=X(I)
3280 NEXT I:RETURN
3300 REM UP TEXT *****
3310 FOR J=1 TO 2000:NEXT J
3320 POKE V+17,155:POKE V+24,21
3330 RETURN

```

```

4000 REM UP TABELLE *****
4010 PRINTCHR$(147):PRINT" TABELLE (J/N) ?":GOSUB 2400
4020 IF A$="N" THEN 4080
4030 GOSUB 4100:REM UEBERSCHRIFT
4040 FOR I=1 TO N1%:PRINTI,X(I),Y(I)
4050 IF I/10=INT(I/10)THEN GOSUB 2700:GOSUB 4100
4060 NEXT I
4070 GOSUB 2700:REM WARTEN
4080 RETURN
4100 REM UP UEBERSCHRIFT *****
4110 PRINTCHR$(147):PRINT" NR.", "X-WERT", "Y-WERT"
4120 FOR J=1 TO 26:PRINT"-":NEXT J:PRINT:RETURN
5000 REM UP EINGABE *****
5010 GOSUB 2200:REM DATENDISK
5020 PRINTCHR$(147):INPUT"NAME DER DATEI (UND RETURN) ";N$
5030 PRINT:PRINT:PRINTTAB(8)"DATEI WIRD GELESEN"
5035 OPEN 15,8,15
5040 OPEN 8,8,8,N$+",S,R"
5045 GOSUB 2000
5050 I=1
5060 INPUT#8,X(I):INPUT#8,Y(I)
5070 IF ST AND 64 THEN 5090
5080 I=I+1:GOTO 5060
5090 GOSUB 2000:CLOSE 8:CLOSE 15
5100 N1%=I:X(I+1)=EF%:Y(I+1)=EF%:RETURN

```

## Beschreibung des Programms DBG

Satz 1000-1080 :

Vorbereiten des Nachladens, Überschrift, Erläuterungen und Abwarten

Satz 1090-1100 :

Voreinstellungen und Dimensionierungen

Satz 1100-1170 :

Einlesen der Daten (Unterprogramm 5000), Ausgabe (Unterprogramm 3500), Bestimmung der statistischen Maßzahlen (Unterprogramm 4200), Bestimmung der Extremwerte (Unterprogramm 7900), Normieren der Werte zur Ausnutzung des Bildschirms bei der graphischen Darstellung (Unterprogramm 7800), Vorbereiten des Bildschirms für hochauflösende Graphik (Unterprogramm 7600) und Zeichnen eines Achsenkreuzes (Unterprogramm 7650)

Satz 1180-1230 :

Zeichnen des Streudiagramms und der Regressionsgeraden

Satz 1240-1250 :

Zurückschalten in den Textmodus und Beendigung des Programms

Satz 2000-2100 :

Unterprogramm zum Abfragen des Diskettenfehlerkanals

Satz 2200-2240 :

Unterprogramm zur Aufforderung, die Datendiskette einzulegen

Satz 2300-2350 :

Unterprogramm zur Aufforderung, die Programmdiskette einzulegen und zum Nachladen des Programms STATISTIK (Ende des Programms DBG)

Satz 2400-2430 :

Unterprogramm zum Abwarten einer Tastatureingabe

Satz 2500-2630 :

Unterprogramm zur Berechnung und Ausgabe der Parameter der Regressionsgeraden A und B (Ordinatenabschnitt und Steigung) und des Korrelationskoeffizienten R

Satz 2700-2740 :

Unterprogramm zum Abwarten, bis die Leertaste betätigt wird

Satz 2800-2840 :

Unterprogramm zur Vorbereitung des Bildschirms für die hochauflösende Graphik

Satz 2900-2930 :

Unterprogramm zum Zeichnen eines Achsenkreuzes

Satz 3000-3050 :

Unterprogramm zum Zeichnen eines Punktes in der hochauflösenden Graphik

Satz 3100-3160 :

Unterprogramm zum Normieren der Werte, um in der hochauflösenden Graphik den Bildschirm ausnutzen zu können

Satz 3200-3280 :

Unterprogramm zur Bestimmung der größten und der kleinsten X- und Y-Werte

Satz 3300-3330 :

Unterprogramm zur Rückkehr in den Textmodus

Satz 4000-4080 :

Unterprogramm zur Ausgabe des ursprünglichen Datenbestandes in tabellarischer Form, falls dies gewünscht wird

Satz 4100-4120 :

Unterprogramm zur Ausgabe einer Tabellenüberschrift

Satz 5000-5100 :

Unterprogramm zum Einlesen der Daten von der Datendiskette

## 12.6 Variablenliste

In den oben vorgestellten Programmen haben wir die folgenden Variablen verwendet :

- A = Bildschirmadresse beim Zeichnen;  
Ordinatenabschnitt in der Regressions- und  
Trendrechnung
- AM = Arithmetisches Mittel
- A\$ = Tastatureingabe (J/N)
- A9\$ = Tastatureingabe im Programm STATISTIK
  
- B = Steigung in der Regressions- und Trendrech-  
nung
- B\$ = Stringfeld für standardisierte Textausgabe
  
- C = Kontingenzkoeffizient
- C\$ = Stringfeld für standardisierte Textausgabe
  
- D = Differenz zwischen dem kleinsten und dem  
größten Merkmalswert; Differenzen zwischen  
beobachteten und theoretischen Häufigkeiten
- DY = Klassenbreite
- D\$ = Stringfeld für standardisierte Textausgabe
  
- EF% = Endwert des Datenbestands
- E\$ = Stringfeld für standardisierte Textausgabe
- EM\$ = Fehlerinformation im Disketten-Fehlerkanal
- EN = Fehlernummer im Disketten-Fehlerkanal
- ES = Zusatzinformation im Disketten-Fehlerkanal
- ET = Zusatzinformation im Disketten-Fehlerkanal
- EX = Hilfsfeld im Unterprogramm PUNKT (Punkt in  
der hochauflösenden Graphik)

HG = Größte Häufigkeit  
HZ = Absolute Häufigkeit  
HRZ= Relative Häufigkeit

I = Laufindex  
IA = Anfangswert des Laufindex I  
IE = Endwert des Laufindex I

J = Laufindex  
JJ = Laufindex  
J1 = Hilfszählindex  
J2 = Hilfszählindex

K = Laufvariable  
KA = Anfangszählwert eines Datenblocks  
KE = Endzählwert eines Datenblocks

M = Markiervariable zur Steuerung der Ausgabe  
M9 = Markiervariable zur Steuerung der Ausgabe  
M\$ = Menüvariable (N/V)  
MV = Anzahl der missing values  
MV%= Voreinstellung für missing values

N = Anzahl der Werte in den Berechnungen  
NA = Anzahl der Fälle in der Datei  
NE = Nenner in der Berechnung der Regressions-  
oder Trendsteigung  
N% = Voreinstellung (Maximalzahl der Fälle)  
N1% = Anzahl der Werte in einer Datei  
N\$ = Name der Datei  
NN\$= Hilfsfeld zur Aufnahme von "W" (oder "R")  
bzw. "A" beim Öffnen oder Verlängern von  
Dateien

R = Korrelationskoeffizient

**S** = Hilfsfeld im Unterprogramm PUNKT (Punkt in der hochauflösenden Graphik;  
 Summe(n);  
 Bildschirmspalte bei Graphiken  
**ST** = Hilfsfeld beim Erfassen des Dateiendes  
**SY** = Startpunkt der Klassifizierung  
**S1-S5** = Summen in der Regressions- und Trendrechnung  
  
**T** = Theoretische Häufigkeiten  
  
**V** = Startadresse des Videochips  
  
**X** = Wert;  
 Koordinate in der hochauflösenden Graphik  
**XG** = Größter X-Wert  
**XK** = Kleinster X-Wert  
**XL** = Transformierter X-Wert für hochauflösende Graphik  
**X1,X2** = X-Koordinaten benachbarter Reihenwerte  
  
**Y** = Wert;  
 Koordinate in der hochauflösenden Graphik  
**YE** = Klassenendpunkte  
**YG** = Größter Y-Wert  
**YK** = Kleinster Y-Wert  
**YL** = Transformierter Y-Wert für hochauflösende Graphik  
**Y1,Y2** = Y-Koordinaten benachbarter Reihenwerte  
  
**Z** = Bildschirmzeilenindex;  
 Zähler von B in der Regressions- und Trendrechnung  
**Z\$** = Menü-Eingabe bzw. Stringfeld für Informationstext  
**Z%** = Nummer einer fehlerhaften Tabellenzeile  
**ZA** = Anfangs-Bildschirmzeile  
**ZE** = End-Bildschirmzeile  
**Z1\$** = Stringfeld für Menüeingabe  
**Z1%** = Menüziffer (= VAL(Z1\$))

Z9 = Menüziffer (= VAL(Z9\$))  
Z9\$= Stringfeld für die Menüeingabe im Programm  
STATISTIK

## 12.7 Programmgergebnisse

An einem einfachen schematischen Beispiel soll abschließend erläutert werden, welche Ergebnisse das Programmpaket produziert.

Nehmen wir an, der Benutzer möchte 30 Gewichtsangaben eingeben und unter dem Namen WAAGE speichern, um sie später in einem zweiten Schritt einer univariaten statistischen Auswertung (Mittelwerte, Streuung, Häufigkeitsverteilung usw.) zu unterziehen.

Zu diesem Zweck gibt er nach Einlegen seiner Programmdiskette das folgende Kommando :

```
LOAD "STATISTIK",8
```

Nach dem Kommando RUN erscheint auf dem Bildschirm nach Überschrift und erläuternden Hinweisen das folgende Menü :

DER BENUTZER KANN EINE DER FOLGENDEN  
MOEGlichkeiten WAELHEN :

DATENEINGABE UND SPEICHERUNG (1)  
EINLESEN DER DATEN VON DISKETTE,  
KORREKTUR UND NEU-SPEICHERN (2)

EINLESEN DER DATEN VON DISKETTE UND

ZEITREIHENSTATISTIK (3)

UNIVARIATE TABELLEN (1 MERKMAL) (4)

UNIVARIATE GRAPHIKEN (1 MERKMAL) (5)

BIVARIATE TABELLEN (2 MERKMALE) (6)

BIVARIATE GRAPHIK (2 MERKMALE) (7)

NICHTS DAVON (9)

BITTE NUMMER EINGEBEN

Geben wir daraufhin die Nummer 1 ein, so antwortet  
das Programm :

MOMENT BITTE  
DAS PROGRAMM WIRD GELADEN

und stellt uns dann das nachgeladene Programm ESP  
zur Verfügung.

Dieses meldet sich nach der Ausgabe einer Über-  
schrift und verschiedenen Erläuterungen mit dem  
folgenden Menü :

AUSWAHL DER EINGABE :

- ZEITREIHENDATEN (1)
- UNIVARIATE DATEN (1 MERKMAL) (2)
- BIVARIATE DATEN (2 MERKMALE) (3)
  
- NICHTS DAVON (9)

Wählen wir nun mit der Zielsetzung einer späteren univariaten Auswertung die Ziffer 2, so erhalten wir ein weiteres Menü :

UNIVARIATE STATISTIK

- NEU-ANLEGEN EINER DATEI (N)
- ODER
- VERLAENGERN EINER DATEI (V)

Dieses Menü veranlaßt uns, den Buchstaben N einzugeben.

Wenn dies geschehen ist, meldet das Programm :

FORMATIERTE DATENDISKETTE EINLEGEN  
UND LEERTASTE DRUECKEN !

Wenn wir auch dieser Aufforderung gefolgt sind, erfragt das Programm den Namen der neuen Datei.

Auf diese Frage antworten wir mit WAAGE und das Programm beginnt dann mit der Anforderung der Werte :

FEHLENDER WERT = -99  
LETZTER WERT = -111

1. WERT (UND RETURN) : ?

Wir geben nun den ersten Wert ein und betätigen die RETURN-Taste. Daraufhin wird der zweite Wert angefordert usw.

Wenn 10 Werte eingegeben sind, so meldet das Programm :

MOMENT BITTE, ES WIRD GESPEICHERT

und am Anlaufen des Diskettenlaufwerks erkennen wir, daß die jetzt eingegebenen Daten gespeichert werden.

Ist dies erfolgt, fordert das Programm den 11. Wert an usw.

Diese Prozeduren wiederholen sich solange, bis der Benutzer den Wert -111 eingibt (dies wäre in unserem Beispiel also als 31. Wert einzugeben), oder bis das Programm meldet, daß keine weiteren Werte eingegeben werden können, weil die Dimensionierungsgrenzen erreicht sind.

Danach fordert das Programm zum Einlegen der Programmdiskette auf und lädt dann automatisch wieder das Programm STATISTIK, bietet uns also wieder das erste Auswahlmenü an.

Wenn wir von der Vermutung ausgehen, daß bei der Dateneingabe Fehler passiert sind, die nun vor der weiteren statistischen Analyse korrigiert werden sollen, so wählen wir die Menünummer 2. Dies bewirkt das Laden des Programms DSP, mit dessen Hilfe es möglich ist, Daten von der Diskette zu lesen, ggf. zu korrigieren und unter dem gleichen Dateinamen wieder "zurückzuspeichern".

Ist dies erfolgt, wird uns wieder das Menü aus dem Programm STATISTIK angeboten.

Wählen wir jetzt die Ziffer 4 (UNIVARIATE TABELLEN), so wird das Programm DUT nachgeladen, welches als erstes nach dem Namen der zu lesenden Datei fragt.

Wir geben den Namen WAAGE ein und werden daraufhin aufgefordert, die Datendiskette einzulegen.

Ist dies erfolgt, erscheint die Meldung :

DATEI WIRD GELESEN

und das Programm fragt, ob die Ausgabe der Urliste (des ursprünglichen Datenbestands also) gewünscht wird.

Antworten wir mit J, wird der Datenbestand in Zehnerblöcken ausgegeben.

Danach wird angeboten, eine einfache Häufigkeitstabelle für den Fall auszugeben, daß die Werte im Bereich zwischen 0 und 9 ganzzahlig codiert sind, d.h. es wird zunächst gefragt, ob diese Bedingung erfüllt ist.

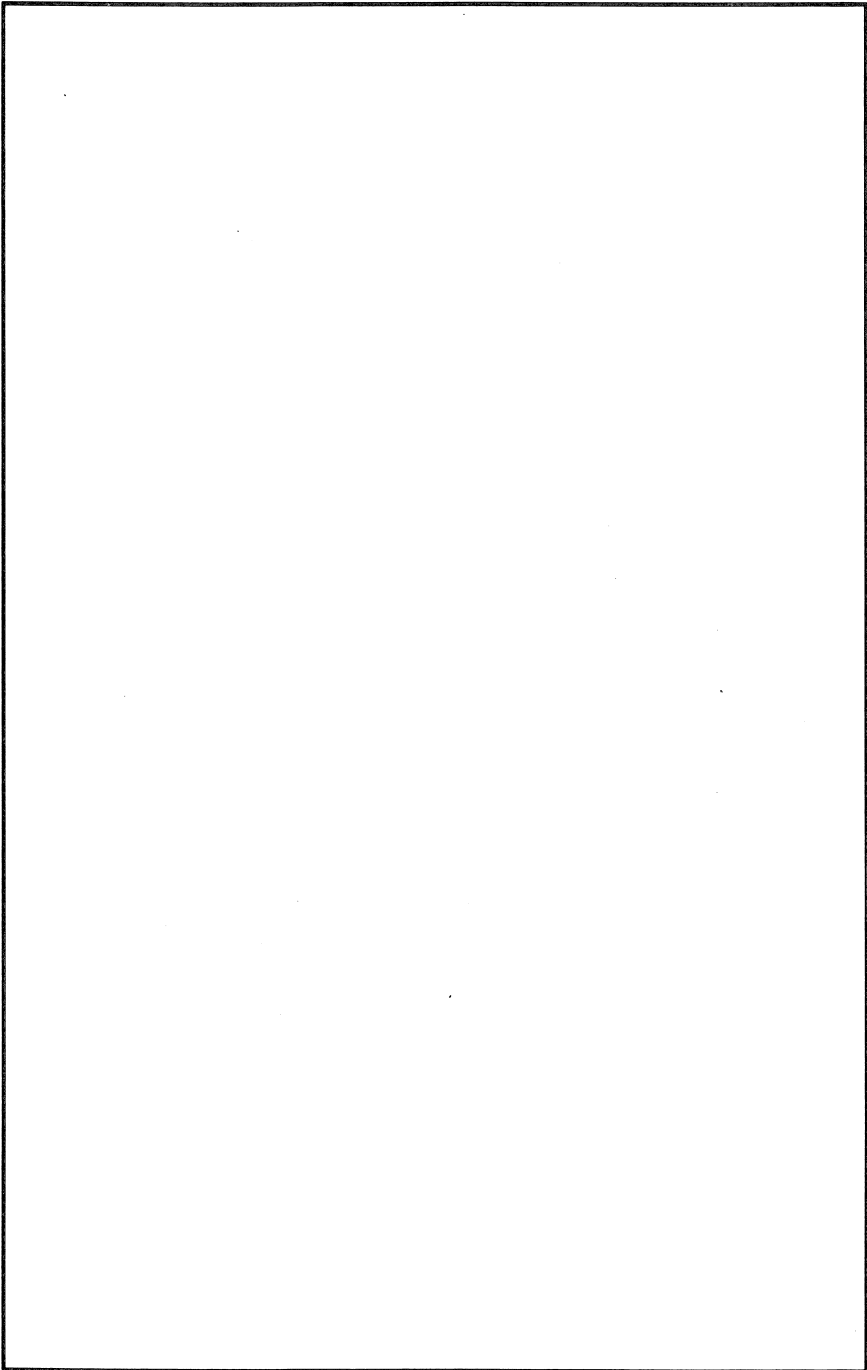
Da wir hier mit N (für Nein) antworten müssen, bringt uns das Programm zur Information den kleinsten und den größten Merkmalswert auf den Bildschirm und fordert dann die Klassenbreite und den Startpunkt der Klassifizierung für die zu erstellende Häufigkeitsverteilung an.

Geben wir passende Werte ein, erzeugt das Programm eine tabellarische Häufigkeitsverteilung der eingegebenen Daten und bestimmt Mittelwert und Standardabweichung dieser Verteilung.

Wollen wir diese Verteilung auch graphisch sehen, so muß nach Wiedererscheinen des Auswahlmenüs aus

dem Programm STATISTIK die Menüziffer 5 eingegeben werden (UNIVARIATE GRAPHIKEN).

Durch Auswahl der Ziffer 9 in diesem ersten Menü können wir dann den gesamten Programmlauf beenden.



## Literaturhinweise

In der folgenden Literaturübersicht beschränken wir uns auf eine Auswahl einführender Titel. Es wird also nicht auf größtmögliche Vollständigkeit dieser Übersicht Wert gelegt, sondern dem Leser sollen einige wenige Titel an die Hand gegeben werden, die es ihm erlauben, die hier angesprochenen Themenbereiche weiter zu vertiefen.

Bei Bedarf wird er in diesen angegebenen Büchern weitere Literaturhinweise finden, die dann Detailprobleme sicherlich klären können, falls das erforderlich oder gewünscht sein sollte.

### 1. Statistik

BARTEL, H. (1978) : Statistik I, Stuttgart/New York

CLAUSS, G./EBNER, H. (1975) : Grundlagen der Statistik, Frankfurt

FRIEDRICHS, J. (1977) : Methoden der empirischen Sozialforschung, Reinbek

HELLER, K./ROSEMANN, B. (1974) : Planung und Auswertung empirischer Untersuchungen, Stuttgart

KELLERER, H. (1972) : Statistik im modernen Wirtschafts- und Sozialleben, Reinbek

KRIZ, J. (1973) : Statistik in den Sozialwissenschaften, Reinbek

KROMREY, H. (1983) : Empirische Sozialforschung, 2. Auflage, Opladen

NOELLE-NEUMANN,E. (1963) : Umfragen in der Massengesellschaft, Reinbek

TIEDE,M./VOSS,W. (1982) : Induktive Statistik, 2 Bände, 2.Auflage, Bochum

VOSS,W. (1979) : Bausteine der Statistik, Köln

## 2. Programmieren

Wir beschränken uns in diesem Abschnitt auf die in diesem Zusammenhang wichtigsten Veröffentlichungen des Hauses DATA-BECKER, Düsseldorf, die in großer Zahl zur Programmierung des Commodore 64 in BASIC erschienen sind.

ANGERHAUSEN/BECKER/GERITS/SHELLENBERGER : 64 für Profis

ANGERHAUSEN/BRÜCKMANN/ENGLISCH/GERITS : 64 intern

ANGERHAUSEN/ENGLISCH/GERITS/THRUN : 64 Tips und Tricks (2 Bände)

BALOUÏ : Alles über Datenbanken und Dateiverwaltung für den Commodore 64

ENGLISCH/SCZCEPANOWSKI : Das große Floppy-Buch

KAMPOW : Das BASIC-Trainingsbuch zum Commodore 64

PLENGE : Das Grafikbuch zum Commodore 64

SCZCEPANOWSKI : 64 für Einsteiger

SEVERIN : Commodore 64 für Technik und Wissenschaft

VOSS : Einführung in die Künstliche Intelligenz

VOSS : Das Schulbuch zum Commodore 64

## STICHWORTVERZEICHNIS

---

### \*\* A \*\*

ABFRAGE	31
ABRUNDEN	173
ABSOLUTE HÄUFIGKEIT	155, 364
ADDITION	25
ADRESSE	39
ANFÜGEN	102
ANTEILSWERT	345
APPEND	50, 103
ARITHMETISCHER OPERATOR	24
ARITHMETISCHES MITTEL	59, 69, 236, 240, 357, 364
ARRAY	57
AUFRUNDEN	173
AUSGLEICHSLINIE	266
AUSPRÄGUNG	24

### \*\* B \*\*

BASIC	2, 13, 19
BEDINGTER, SPRUNG	30
BILDSCHIRMADRESSE	39
BILDSCHIRMPOSITION	38
BILDSCHIRMZONE	21
BIVARIAT	154
BIVARIATE TABELLE	181
BIVARIATE VERTEILUNG	158, 255

\*\* C \*\*

C64, COMMODORE	2
CASE	57
CHR\$	25, 36
CLOSE	50
CODEBUCH	55
COMMODORE-TASTE	41
COVARIANZ	261
CTRL-TASTE	41

\*\* D \*\*

DARSTELLUNG, GRAPHISCHE	189, 364
DARSTELLUNG, TABELLARISCHE	364
DATA	28, 75
DATEI	57
DATEIVERLÄNGERUNG	101
DATEIVERWALTUNG	125
DATEN	8, 9, 53
DATENAUFNAHME	53
DATENAUSGABE	125
DATENDISKETTE	93, 366
DATENEINGABE	58, 59, 69, 91
DATENFEHLER	134
DATENGEWINNUNG	53
DATENMATRIX	57, 76, 126, 159
DATENVERARBEITUNG	1
DESKRIPTIVE STATISTIK	339
DETERMINATIONSKOEFFIZIENT	262
DEZIMALZAHL	360
DIM	29
DIMENSIONIERUNG	80
DISKETTENLAUFWERK	359
DISKETTENNUTZUNG	46
DISKRETE VARIABLE	24, 154, 191
DISKRETE VERTEILUNG	167
DIVISION	25
DOPPELPUNKT	20
DURCHSCHNITT	357

**\*\* E \*\***

EINGABE	363
EINGABE, GENERELLE	111
EINGABEFEHLER	137
EINGABEPROGRAMM	111
EINGEBAUTE FUNKTION	25
EINZELPUNKT	42
END	21
ERGEBNISAUSGABE	20
ERGÄNZUNGEN	15
EVA-PRINZIP	60
EXTERNER SPEICHER	46, 91

**\*\* F \*\***

FALL	57
FARBADRESSE	40
FARBE	40
FEHLERKANAL	94
FEHLERMELDUNG	17, 361
FELD	23, 57
FIELD	57
FILE	57
FOR...TO	35
FORMATIEREN	47
FUNKTION	25, 36
FUNKTION, EINGEBAUTE	25

**\*\* G \*\***

GANZZAHLIGE VARIABLE	23
GAUSS-VERTEILUNG	342
GENERELLE EINGABE	111
GESCHACHELTE SCHLEIFE	76
GET	28
GLÄTTUNG	298, 324

GLÄTTUNGSLINIE	324
GOSUB	33
GOTO	30
GRAPHIK	38
GRAPHIK, HOCHAUFLÖSENDE	39, 41
GRAPHIKMODUS	43
GRAPHIKSYMBOL	38
GRAPHISCHE DARSTELLUNG	189, 364
GRUNDGESAMTHEIT	340

**\*\* H \*\***

HAUPTPROGRAMM	33
HINTERGRUNDFARBE	44
HISTOGRAMM	193, 218, 364
HOCHAUFLÖSENDE GRAPHIK	39, 41
HOCHRECHNUNG	339, 340
HÄUFIGKEIT	155
HÄUFIGKEIT, ABSOLUTE	155, 364
HÄUFIGKEIT, RELATIVE	155, 364
HÄUFIGKEITSTABELLE	153, 364
HÄUFIGKEITSVERTEILUNG	189

**\*\* I \*\***

IF...THEN	31
INDIZIERTE VARIABLE	29
INFORMATION	5
INFORMATIONSEINGABE	27
INFORMATIONSVENLUST	359
INHALTSVERZEICHNIS	47, 93
INPUT	27, 69
INT	25
INTERAKTIVE NUTZUNG	47, 48
INTERPOLATION	301, 364
INTERVALLSCHÄTZUNG	339
INTERVALLSKALIERT	190

**\*\* K \*\***

KLASSE	157, 193
KLASSENBREITE	175
KLEINRECHNER	10
KOMMANDO	22
KONTINGENZKOEFFIZIENT	263, 283, 365
KORREKTUR	360
KORREKTURMÖGLICHKEIT	112
KORRELATIONSKOEFFIZIENT	261, 278, 365
KORRELATIONSRECHNUNG	255, 261
KREISDIAGRAMM	192, 206, 358
KREUZTABELLE	181, 255, 364
KUCHENDIAGRAMM	358

**\*\* L \*\***

LADEN	47
LAUFVARIABLE	35
LEFT\$	37
LEN	37
LESEN	363
LESEPROZESS	126
LET	24, 59
LINEARE REGRESSION	266
LINIENZUG	42
LIST	22
LOAD	47
LÖSCHEN	48

**\*\* M \*\***

MASSZAHL	235, 365
MEHRFACHVERZWEIGUNG	35, 147
MENÜ	365
MENÜTECHNIK	137
MERKMALSTRÄGER	7
METHODE, STATISTISCHE	153
METHODEN, STATISTISCHE	1, 5

MID\$	37
MIKROPROZESSOR-TECHNOLOGIE	3
MISSING-VALUE	56, 362
MITTEL, ARITHMETISCHES	59, 69, 236, 240, 357, 364
MITTELWERT	235, 236, 240, 345
MITTELWERTBERECHNUNG	75, 83
MODUS	236, 240
MULTIPLIKATION	25
MULTIVARIAT	154
MULTIVARIATE TABELLE	159

\*\* N \*\*

NAME	23, 24
NEU-ANLAGE	112
NEW	23
NEXT	35
NOMINALSKALIERT	190
NORMALGRAPHIK	38, 198
NORMALVERTEILUNG	342
NUMERISCHE VARIABLE	23
NUTZUNG, INTERAKTIVE	47, 48

\*\* O \*\*

ON...GOSUB	35
ON...GOTO	34
OPEN	47, 49
OPERATOR, ARITHMETISCHER	24
ORDINALSKALIERT	190
OVERLAY-TECHNIK	366

\*\* P \*\*

POKE	39
POLARKOORDINATEN	207
POTENZIERUNG	25
PRG	49
PRINT	20
PROBLEM	14
PROBLEMANALYSE	14, 60
PROGNOSE	258, 294, 295
PROGRAMMBESCHREIBUNG	15
PROGRAMMDISKETTE	17, 366
PROGRAMMERGEBNISSE	15
PROGRAMMLISTE	14
PROGRAMMPAKET	15, 355
PROGRAMMVERZWEIGUNG	30
PUNKT	44, 360
PUNKTFARBE	44

\*\* R \*\*

RANDOM-ACCESS-SPEICHERUNG	48, 91
READ	28, 75
RECORD	57
REELLE VARIABLE	23
REGRESSION, LINEARE	266
REGRESSIONSFUNKTION	256, 365
REGRESSIONSRECHNUNG	255, 256
REIHENGLÄTTUNG	324
RELATIVE HÄUFIGKEIT	155, 364
REM	23
REPRÄSENTATIV	340
RESTORE	84
RESTORE-TASTE	46
RETURN	33
RETURN-TASTE	20, 28
RIGHT\$	37
RND	25
RUN	22
RUN-STOP-TASTE	31, 46
RUNDUNG	173
RÜCKSPRUNG	33

\*\* S \*\*

SATZ	20
SAVE	47
SCHIEFE	237
SCHLEIFE	35
SCHLEIFE, GESCHACHELTE	76
SCHÄTZPROGRAMM	345
SEQ	49
SEQUENTIELLE SPEICHERUNG	48, 91
SET	57
SHIFT	38
SKALENQUALITÄT	189
SORTIEREN	160, 161
SPALTE	38
SPANNWEITE	238, 249, 364
SPEICHER, EXTERNER	46, 91
SPEICHERN	47, 363
SPEICHERUNG	92, 363
SPEICHERUNG, SEQUENTIELLE	48, 91
SPRUNG	30
SPRUNG, BEDINGTER	30
SPRUNG, UNBEDINGTER	30
SQR	25
STABDIAGRAMM	191, 197, 364
STANDARDABWEICHUNG	238, 249, 364
STATEMENT	20
STATISTIK	5
STATISTIK, DESKRIPTIVE	339
STATISTISCHE METHODE	153
STATISTISCHE METHODEN	1, 5
STATISTISCHE ZEITREIHE	293
STEP	35
STETIGE VARIABLE	24, 154, 157, 193
STETIGE VERTEILUNG	174, 248
STICHPROBE	340
STR\$	37
STREUDIAGRAMM	194, 195, 228, 255, 365
STREUUNG	235, 249
STREUUNGSMASS	238
STRING	23, 36, 167
STRINGBEARBEITUNG	36
STRINGVARIABLE	23
SUBTRAKTION	25

**\*\* T \*\***

TAB	21
TABELLARISCHE DARSTELLUNG	364
TABELLE, BIVARIATE	181
TABELLE, MULTIVARIATE	159
TEXTMODUS	43
TREND	296
TRENDFUNKTION	297
TRENDLINIE	324, 364
TRENDPROGNOSE	295, 364
TYP	23

**\*\* U \*\***

UNBEDINGTER SPRUNG	30
UNIVARIAT	154
UNTERPROGRAMM	33
URLISTE	159

**\*\* V \*\***

VAL	37
VARIABLE	23
VARIABLE, DISKRETE	24, 154, 191
VARIABLE, GANZZAHLIGE	23
VARIABLE, INDIZIERTE	29
VARIABLE, NUMERISCHE	23
VARIABLE, REELLE	23
VARIABLE, STETIGE	24, 154, 157, 193
VARIABLENLISTE	14
VARIANZ	261
VERLÄNGERUNG	112
VERTEILUNG	155, 364
VERTEILUNG, BIVARIATE	158, 255
VERTEILUNG, DISKRETE	167
VERTEILUNG, STETIGE	174, 248
VERÄNDERUNG	134
VIDEOCHIP	42

**\*\* W \*\***

<b>WAHRSCHEINLICHKEITSSTATISTIK</b>	<b>339</b>
<b>WERT</b>	<b>57</b>
<b>WERTZUWEISUNG</b>	<b>23</b>
<b>WRITE</b>	<b>103</b>

**\*\* Z \*\***

<b>ZAHL</b>	<b>360</b>
<b>ZEILE</b>	<b>38</b>
<b>ZEILENVORSCHUB</b>	<b>21</b>
<b>ZEITREIHE</b>	<b>58, 159, 293, 363</b>
<b>ZEITREIHE, STATISTISCHE</b>	<b>293</b>
<b>ZEITREIHENSTATISTIK</b>	<b>293</b>
<b>ZUFALLSSTICHPROBE</b>	<b>340</b>
<b>ZÄHLEN</b>	<b>83</b>
<b>ZÄHLINDEX</b>	<b>70</b>



**Angerhausen/Brückmann/Englisch/Gerits  
64 Intern**

**Das große Buch zum Commodore 64 mit dokumentiertem Schaltplan**

Die Herausforderung für jeden ernsthaften Anwender! Alles über Technik, Betriebssystem und fortgeschrittene Programmierung des Commodore 64. Mit ausführlichem ROM-Listing, sorgfältig dokumentierten Originalschaltplänen zum Ausklappen, zahlreichen Abbildungen, Schaltbildern, Blockdiagrammen und - natürlich - mit anspruchsvollen Programmen. Mit diesem unentbehrlichen Buch lernen Sie Ihren C 64 erst richtig kennen.

**352 Seiten, 2 Schaltpläne, DM 69,-  
ISBN 3-89011-000-2**



**Brückmann/Gerits/  
Wiens**

**Das große Druckerbuch  
369 Seiten, DM 49,-  
ISBN 3-89011-020-7**

Mit diesem Buch meistern Sie absolut jedes Drucker-Problem. Ob Sekundäradressen, Schnittstellen, Steuerzeichen, formatierte Datenausgabe oder Grafik-Hardcopy: alles hervorragend erklärt. Selbstverständlich wieder viele nützliche Programme zum Abtippen; außerdem wichtige Hilfen zur Druckeranpassung, ein Betriebssystemlisting des MPS 801 und ein eigenes Kapitel zum VC-1520. Jetzt holen Sie das Optimum aus Ihrem Drucker heraus!

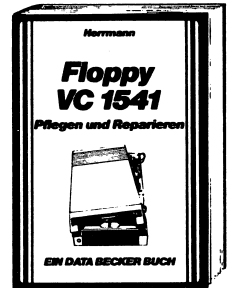
Das Standardwerk zur Floppy VC 1541. Alles über Diskettenprogrammierung vom Einsteiger bis zum Profi. Neben grundlegenden Informationen zum DOS, zu den Systembefehlen und Fehlermeldungen stehen mehrere Kapitel zur praktischen Dateiverwaltung mit der Floppy. Umfangreiches, dokumentiertes DOS-Listing. Dazu eine Fundgrube verschiedenster Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender einfach zur Pflichtlektüre machen.



**Englisch/Szczepanowski**

**Das große Floppy-Buch  
482 Seiten, DM 49,-  
ISBN 3-89011-006-3**

Selbsthilfe spart Zeit, Ärger und Geld - gerade Probleme wie Floppy-Justage oder Reparaturen der Platine sind mit oft einfachen Mitteln zu lösen. Anleitungen zur Behebung der meisten Störungen, Ersatzfeillisten und eine Einführung in Mechanik und Elektronik des Laufwerks. Natürlich gehören auch genaue Angaben zu Werkzeug und Arbeitsmaterial zum Buch, das in jeder Beziehung für "effektiv und preiswert" steht.



**Herrmann  
VC-1541 Pflegen und  
Reparieren  
ca. 200 Seiten, DM 49,-  
ISBN 3-89011-079-7**

Das Superbuch, das Ihnen zeigt, was alles in Ihrem Rekorder steckt. Informiert detailliert und leichtverständlich über Datensette und Cassetten-Speicherung. Mit den Spitzenprogrammen Autostart, Catalog (sucht und lädt automatisch!), Backup von und auf Floppy, Save von Speicherbe-reichen und einem neuen Cassetten-Betriebssystem mit dem 10-20 mal schnelleren (!) Fasttape. Außerdem weitere nützliche Hinweise (Kopfjustage, Kontroll-Lautsprecher) und Programme.



**Paulissen  
Das Cassettenbuch zum  
Commodore 64 und VC-20  
190 Seiten, DM 29,-  
ISBN 3-89011-030-4**

**Brückmann  
Der Commodore 64 und  
der Rest der Welt  
229 Seiten, DM 49,-  
ISBN 3-89011-015-0**



Literatur speziell für den engagierten Hobbyelektroniker vom fähigen Techniker zusammengestellt. Schwerpunkt sind ausgesuchte Ideen zu verschiedenen Einsatzmöglichkeiten des C 64: Motorsteuerung, A/D-Wandler, Spannungs- und Temperaturmessung und Lichtorgel. Dazu eine Reihe hochinteressanter Schaltungen zum Nachbau: EPROM-Programmer, Sprachsynthesizer, Frequenzzähler und noch mehr.

Ein Bestseller, der erfolgreich und umfassend in die Maschinensprache einführt. Sie lernen Aufbau und Arbeitsweise des 6510 Mikroprozessors kennen, erfahren Wichtiges über Eingabe und Start von Maschinenprogrammen sowie über den Umgang mit Monitor, Assembler und Disassembler. Assembler und Disassembler sind im Buch als Programme ebenso enthalten wie ein Einzelschrittssimulator. Viele ausführlich beschriebene Beispielprogramme und Routinen machen Ihnen den Einstieg leicht.



**Englisch**  
**Das Maschinensprachebuch zum Commodore 64**  
 201 Seiten, DM 39,-  
 ISBN 3-89011-008-8

Sie haben den Einstieg in die Maschinensprache geschafft? Dann werden Sie jetzt zum Profi! Von der Problemanalyse bis zum Maschinensprachealgorithmus werden Sie umfassend in die Grundlagen der professionellen Maschinenspracheprogrammierung eingeführt. Dazu wieder viele Beispielprogramme, komplette Maschinenroutinen und wichtige Tips & Tricks zur Maschinenspracheprogrammierung und zur Arbeit mit dem Betriebssystem.



**Englisch**  
**Das Maschinensprachebuch für Fortgeschrittene zum Commodore 64**  
 206 Seiten, DM 39,-  
 ISBN 3-89011-022-3



**Liesert**  
**Peeks & Pokes zum Commodore 64**  
 177 Seiten, DM 29,-  
 ISBN 3-89011-032-0

Leichtverständlich wird hier der Umgang mit PEEK- und POKE-Befehlen erklärt, die vieles vereinfachen, was sonst komplizierte Maschinenroutinen nötig machen würde. Dazu nützliche POKEs und Ihre Anwendungsmöglichkeiten. Außerdem Grundlegendes zum Aufbau des C-64: Betriebssystem, Interpreter, Zeropage, Pointer und Stacks, Charakter-Generator, Sprite-Register und vieles mehr. Mit einer ersten Einführung in die Maschinensprache und etlichen Beispielprogrammen.

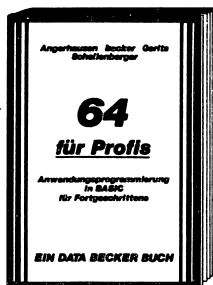
**Schmidt**  
**Assembler Trainingsbuch**  
 ca. 250 Seiten, DM 39,-  
 erscheint April 1985  
 ISBN 3-89011-071-1



Dem interessierten Anfänger werden hier die weitverbreiteten Assembler Profimat, MAE 64 und T.EX.AS. ausführlich anhand von Übungen und Beispielen erklärt und aufbauend eine konsequente Einführung in die Maschinensprache vermittelt. Gleichzeitig ein fundiertes Nachschlagewerk: Ein umfassender und übersichtlicher Anhang mit Erläuterungen aller wichtigen Begriffe sowie ein reichhaltiges Stichwortverzeichnis ergänzen dieses Trainingsbuch optimal.

Bedienerfreundlich und erfolgreich in BASIC programmieren ist kein Privileg von Fachleuten. Wie man es macht, verraten die Software-Autoren aus dem Hause DATA BECKER: Menüsteuerung, Maskenaufbau, Parametrisierung und Dokumentation sind die Stichworte. Dazu die neue leistungsfähige Datenstruktur QUISAM mit lauffertigen Beispielprogrammen.

**Angerhausen/Becker/ Gerits/Schellenberger**  
**64 für Profis**  
 302 Seiten, DM 49,-  
 ISBN 3-89011-007-X

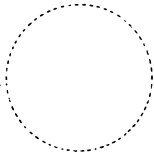


**Einfach besser programmieren!**

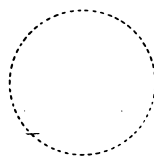
Eine umfassende, praxisorientierte Einführung in den Komplex Dateiverwaltung, Datenbanken, Datenbanksprachen und Expertensysteme. Erklärt werden logische und physische Datenstrukturen oder sequentieller und Direktzugriff. Wer wissen will, wie ein Hashing-Algorithmus aufgebaut ist oder wie man ein komplettes Dateiverwaltungsprogramm erstellt (das im Buch als ausführliches Listing enthalten ist), der braucht dieses Superbuch.



**Baloui**  
**Alles über Datenbanken und Dateiverwaltung für den Commodore 64**  
 222 Seiten, DM 39,-  
 ISBN 3-89011-054-1



DM 39,-- Pf für Postgirkonto Nr. 789 - 436  
Absender: \_\_\_\_\_



Für Vermerke des Absenders

**Empfängerabschnitt**

DM 39,--	Pf ---
für Postgirkonto Nr. 789 - 436	
Absender (mit Postleitzahl) _____	
Verwendungszweck <b>Diskette z. Buch</b>	
366 044	

**Zahlkarte**

(Mit Schreibmaschine, Tinte oder Kugelschreiber deutlich ausfüllen)  
(DM-Betrag in Buchstaben wiederholen)

DM 39,-- Pf ---

**Neununddreißig**

für <b>DATA BECKER GmbH</b>	
Merowingerstr. 30	Postgirkonto Nr. 789 - 436
4000 Düsseldorf	Postgirkonto Essen

**Entlieferungsschein**

- Bitte sorgfältig aufbewahren -

DM 39,-- Pf ---

für <b>DATA BECKER GmbH</b>	
Merowingerstr. 30	Postgirkonto Nr. 789 - 436
4000 Düsseldorf	Postgirkonto Essen

**LADEN, STARTEN - KLAR!**

Für alle diejenigen, die sich fleißiges Abtippen der Programme ersparen und trotzdem das komplette Programmpaket nutzen wollen, gibt es einen Ausweg:

Mit der nebenstehenden Post - Zahlkarte einfach die Diskette zum Buch bestellen !

Diskette zum STATISTIKBUCH DM 39,--



Postvermerk

Diskette zum Buch  
„Das Statistikbuch zum C - 64“  
366 044

Für Mitteilungen an den Empfänger

Bedienen Sie sich der Vorteile  
eines eigenen Post girokontos!  
Auskunft hierüber erteilt jedes Postamt

Feld  
für  
postdienstliche  
Zwecke

### Einlieferungsschein

(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte  
(wird bei der Einlieferung bar erhoben)  
bis 10 DM ..... 90 Pf  
über 10 DM (unbeschränkt) 1,50 DM

**Sparen mit  
wachsendem  
Zins.  
Bei der Post.**

**PostSparen**



### ***DAS STEHT DRIN:***

Statistik auf dem C-64. Ein Lehr- und Arbeitsbuch zu den Grundlagen der Statistik mit zahlreichen Problemlösungen und fertigen Basic-Programmen. Alle, die viel mit umfangreichen statistischen Berechnungen zu tun haben, werden dieses Buch zu schätzen wissen.

Aus dem Inhalt:

- Häufigkeitstabellen
- Mittelwerte und Streuungen
- Regressions- und Korrelationsrechnung
- Zeitreihenstatistik
- Hochrechnungen  
und vieles mehr

Die Einzelprogramme sind sehr gut und ausführlich dokumentiert und so leicht an eigene Anwendungen anpassbar. Das ist Software zum Selbermachen! Das komplette Programmpaket ist auch auf Diskette erhältlich.

### ***UND GESCHRIEBEN HAT DIESES BUCH:***

Werner Voß ist Professor für Statistik an der Universität Bochum und Autor zahlreicher didaktisch gelungener Veröffentlichungen im Bereich Statistik und Datenverarbeitung.

***ISBN 3-89011-102-5***