

Datenanalyse mit Mathematica

Einführung für Studierende
aller Fachrichtungen

Von

Dipl. Math. Werner Sanns

Dipl. Math. Marco Schuchmann

R. Oldenbourg Verlag München Wien

Die Informationen in dieser Dokumentation wurden mit großer Sorgfalt erstellt. Trotzdem können Fehler nicht ausgeschlossen werden. Für fehlerhafte Angaben und deren Folgen werden weder juristische Verantwortung noch irgendeine Haftung übernommen. Für eine Mitteilung eventueller Fehler sind die Autoren dankbar.

e-mail: Silvia0806@AOL.com oder
M.Schuchmann@t-online.de

Bisher im Oldenbourg Verlag erschienene Werke:

Schuchmann, Sanns: Statistik mit Mathematica

Sanns, Schuchmann: Mathematik für Wirtschaftswissenschaftler und
Ingenieure mit Mathematica

Schuchmann, Sanns: Nichtparametrische Statistik mit Mathematica

Schuchmann, Sanns: Statistik transparent mit SAS, SPSS, Mathematica

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Sanns, Werner:

Datenanalyse mit Mathematica : Einführung für Studierende aller
Fachrichtungen / von Werner Sanns und Marco Schuchmann. -
München ; Wien : Oldenbourg, 1999
ISBN 3-486-25280-1

© 1999 Oldenbourg Wissenschaftsverlag GmbH
Rosenheimer Straße 145, D-81671 München
Telefon: (089) 45051-0, Internet: <http://www.oldenbourg.de>

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Gedruckt auf säure- und chlorfreiem Papier
Gesamtherstellung: Druckhaus „Thomas Müntzer“ GmbH, Bad Langensalza

ISBN 3-486-25280-1

Inhalt

| | | |
|-----------|--|------------|
| 1 | <i>Vorbereitende Programme</i> | 7 |
| 1.1 | Daten | 7 |
| 1.2 | Verteilungen | 34 |
| 2 | <i>Kennzahlen einer Stichprobe</i> | 55 |
| 2.1 | Theoretische vs. empirische statistische Kenngrößen | 55 |
| 2.2 | Kennzahlen nichtmetrischer Daten | 59 |
| 2.3 | Berechnung statistischer Kenngrößen bei metrischen Daten | 62 |
| 3 | <i>Streudiagramme</i> | 72 |
| 4 | <i>Tortendiagramme</i> | 88 |
| 5 | <i>Balkendiagramme</i> | 93 |
| 6 | <i>Boxplots</i> | 103 |
| 7 | <i>Karten</i> | 110 |
| 8 | <i>Häufigkeitsauszählungen, Kreuztabellen</i> | 118 |
| 9 | <i>Kovarianz und Korrelation</i> | 125 |
| 10 | <i>Statistische Tests</i> | 133 |
| 10.1 | Einstichproben-t-Test | 135 |
| 10.1.1 | Einseitiges Testproblem..... | 135 |
| 10.1.2 | Zweiseitiges Testproblem..... | 142 |
| 10.2 | Der Zweistichproben t-Test | 146 |
| 10.3 | Test für die Varianz einer Stichprobe | 151 |
| 10.4 | Test zum Vergleich der Varianzen zweier Stichproben | 153 |
| 10.5 | Chi-Quadrat Test auf Unabhängigkeit | 156 |
| 11 | <i>Überprüfung von Verteilungsannahmen</i> | 161 |
| 11.1 | Der Chi-Quadrat Anpassungstest | 161 |
| 11.1.1 | Der Chi-Quadrat Anpassungstest für stetige Verteilungen..... | 161 |
| 11.1.2 | Der Chi-Quadrat Anpassungstest für diskrete Verteilungen..... | 168 |

| | | |
|-------------|---|------------|
| 11.2 | Der Kolmogoroff-Smirnov Anpassungstest | 175 |
| 11.3 | Grafische Möglichkeiten | 179 |
| 11.3.1 | Empirische und theoretische Verteilungsfunktion | 179 |
| 11.3.2 | Zeichnen in Wahrscheinlichkeitspapier..... | 181 |
| 11.3.3 | Der p-p-Plot..... | 186 |
| 12 | <i>Konfidenzintervalle für die Parameter einer Normalverteilung...</i> | 188 |
| 12.1 | Konfidenzintervall für den Parameter μ | 188 |
| 12.2 | Konfidenzintervall für den Parameter σ^2 | 190 |
| 13 | <i>Approximation von Meßwerten.....</i> | 192 |
| 14 | <i>Anhang A: Grundbegriffe der Wahrscheinlichkeitstheorie</i> | 199 |
| 15 | <i>Anhang B: Mathematica kennenlernen</i> | 205 |
| 16 | <i>Literaturverzeichnis</i> | 243 |

Vorwort

Dieses Buch wendet sich an Studierende aller Fachbereiche, die Daten auf einfache Weise darstellen und auswerten wollen. Wir beschäftigen uns mit den elementaren Methoden der beschreibenden und schließenden Statistik und benutzen dazu das Computeralgebrasystem Mathematica.

Als Voraussetzungen für das Arbeiten mit unserem Buch benötigen Sie Zugang zu einem Rechner mit einer installierten Version von Mathematica. Unsere Programme sind getestet unter der derzeit neuesten Version von Mathematica (3.01) in der PC-Version unter Windows 95, Windows 98 und Windows NT. Sie sind aber größtenteils auch unter älteren Mathematica-Versionen, z.B. Version 2.2.3 lauffähig.

Sie können die Grundlagen von Mathematica im Anhang B dieses Buches erlernen. Diejenigen Mathematica-Anweisungen, die Ihnen vielleicht nicht so geläufig sind, und die auch nicht im Anhang erwähnt sind, werden von uns im Text der einzelnen Kapitel erklärt.

Kenntnisse in der Statistik setzen wir nur in ganz geringem Umfang voraus. Wir versuchen die benötigten Begriffe innerhalb der Kapitel möglichst anschaulich zu erklären. Eine mathematische Formulierung der für uns wichtigen Definitionen und Sätze aus der Wahrscheinlichkeitstheorie geben wir für den interessierten Leser im Anhang A.

Schließlich noch ein Hinweis für den Leser, der sich auch für die weiterführenden Methoden der Statistik interessiert: Im gleichen Verlag erschien unser Werk „Statistik mit Mathematica“, in dem die meisten (überwiegend parametrischen) Verfahren behandelt werden. Zu den nichtparametrischen Verfahren der schließenden Statistik ist unser Buch mit dem Titel „Nichtparametrische Statistik mit

Mathematica“ erschienen. Wenn Sie außerdem eines der großen Statistikpakete SAS oder SPSS kennenlernen wollen und deren „Output“ verstehen wollen, finden Sie in unserem Buch „Statistik transparent mit SAS, SPSS, Mathematica“ (OldenbourgVerlag) die mit Mathematica nachvollzogenen und erklärten Ausgaben wieder.

Darmstadt 1999

Werner Sanns, Marco Schuchmann

1 Vorbereitende Programme

Im diesem ersten Kapitel sollen Sie die Grundlagen der Dateneingabe in Mathematica und einige theoretische Aspekte kennenlernen, bevor Sie ab dem 2 Kapitel mit der eigentlichen Aufbereitung ihrer Daten beginnen werden. Wer Mathematica noch nicht kennt, möge bitte zuerst Anhang B lesen.

1.1 Daten

Daten, das sind die gesammelten Informationen, die über einen interessierenden Sachverhalt zur Verfügung stehen, bilden das Arbeitsmaterial des Statistikers. Sein Ziel ist die Aufbereitung und Auswertung von Daten. Dabei kann er vorhandene Daten mit geeigneten Maßzahlen beschreiben und grafisch aufbereiten. Er kann auch versuchen aus dem Datenmaterial mit Hilfe mathematischer Methoden Schlüsse auf den Sachverhalt selbst zu ziehen, der die Daten lieferte. Im ersten Fall sprechen wir von beschreibender (deskriptiver) Statistik, im zweiten Fall von schließender (induktiver) Statistik. Im vorliegenden Buch beschäftigen wir uns überwiegend mit deskriptiven Methoden der Statistik, während von den Methoden der schließenden Statistik hier nur einige der elementarsten beschrieben werden. Weitergehende Methoden der schließenden Statistik beschreiben wir in unseren im Literaturverzeichnis aufgeführten Büchern, welche aber gründliche Kenntnisse in der Statistik voraussetzen.

Daten, die es in irgend einer Form auszuwerten gilt, können verschiedenster Natur sein. Wir wollen daher zunächst Daten nach verschiedensten Kriterien klassifizieren, denn danach richten sich später die Methoden, die wir zur Datenauswertung anwenden können.

Zuerst unterscheiden wir, ob die Daten stetig oder diskret sind. Stetige Daten sind solche, die innerhalb eines gewissen Intervalls jeden reellen Zahlenwert annehmen können. Die Messungen von Längen (z.B. Körpergrößen), Flächen, Volumina etc. liefern stetige Daten, denn sie können innerhalb eines vernünftigen Zahlenbereichs jeden Wert

annehmen, obgleich man im allgemeinen aus praktischen Gründen die Körpergröße nicht mit mehreren Nachkommastellen angibt. Auch ist es in diesem Beispiel nicht klar, wo die Grenzen für den Zahlenbereich liegen. Es gibt (heute noch) keine Menschen mit einer Körpergröße von 3 Metern oder mehr.

Im Gegensatz zu den stetigen Daten können bei diskreten Daten die Werte nicht ein ganzes reelles Intervall überdecken, sondern es können nur endlich viele (oder höchstens „abzählbar unendlich viele“) Werte als Wertebereich zugrunde liegen. Diskrete Daten müssen keineswegs immer Zahlen sein, sondern können auch qualitative Begriffe, wie Farben, Stimmungen etc., repräsentieren. Wir sprechen dabei dann von kategoriellen Daten.

Unabhängig von der bisher vorgestellten Aufteilung, wird auch zwischen metrischen und nichtmetrischen Daten unterschieden. Mit metrischen Daten können numerische Berechnungen sinnvoll vorgenommen werden. Zum Beispiel hat die Körpergröße metrisches Datenniveau, denn es ist hier sinnvoll die Differenz zwischen zwei Körpergrößen zu berechnen. Außerdem ist es bei metrischen Daten zulässig den Mittelwert zu berechnen. Dies ist bei nichtmetrischen Daten nicht so. Nehmen wir einmal an, wir erfassen außer der Körpergröße auch die Augenfarbe von Personen. Auch wenn wir, wie es häufig aus Gründen der Bequemlichkeit der Fall ist, die festgestellten Augenfarben mit Zahlen codieren, macht es keinen Sinn hier zum Beispiel den Mittelwert zu berechnen. Allgemein sind numerische Operationen, wie Addition, Subtraktion usw. hier nicht sinnvoll. Bei nichtmetrischen Daten können wir noch zwischen nominal und ordinal skalierten Daten unterscheiden. Die Augenfarbe wäre zum Beispiel nominal skaliert, da es zwischen den einzelnen Augenfarben keine Rangfolge gibt, wie dies für ordinale Daten gefordert wird. Erfasst man dagegen die Antwort auf die Frage „Wie geht es Ihnen heute?“ mit den Antwortmöglichkeiten „schlecht“, „mittelmäßig“, „gut“, „sehr gut“, so genügt die Variable, die das Befinden erfasst, einer ordinalen Skala, denn hier liegt eine Rangfolge vor. Auch Schulnoten sind ordinale Daten, sie werden mit den Zahlen 1,2,..bis 6 codiert, dahinter stehen aber Beurteilungen von Leistungen, die man zwar ordnen kann, mit denen nicht sinnvoll gerechnet werden kann.

(Codieren Sie selbst die Noten mit anderen Zahlenwerten, die zwar aufsteigend geordnet sind, die aber andere Abstände besitzen, und bilden Sie den „Mittelwert“, wie dies meist in Benotungen von Klassenarbeiten geschieht !).

Wir wollen nun mit Hilfe von Mathematica Daten bearbeiten. Wir benötigen zunächst die Daten, die wir beschreiben wollen, in einer für Mathematica verständlichen Form. Für kleine Datenmengen genügt es, wenn wir die Daten direkt in Mathematica eingeben. Meist werden Sie jedoch Daten auswerten wollen, die in einer Datei vorliegen. Wir werden beide Methoden vorstellen.

Wichtig: Generell werden wir alle Eingaben mit **fettgedruckten** Zeichen darstellen. Die von Mathematica erzeugten Ausgaben werden dagegen in dünnem Druck dargestellt. Eingabezeilen, die jeweils einzeln in einer eigenen Eingabezelle von Mathematica stehen sollten, sind durch eine nachfolgende Leerzeile von der nächste Eingabezeile getrennt. Fehlt die Leerzeile, so können auch mehrere Eingabezeilen in einer Eingabezelle stehen.

Nun die erste kleine Datenreihe, die Sie bitte eingeben und mit Shift- und Enter-Taste abschicken (Mathematica-Neulinge lesen bitte den Einstieg in Mathematica im Anhang B). Darunter erscheint die Ausgabe von Mathematica, die hier nur die eingegebenen Daten wiederholt:

```
daten1={2,1.3,-9.5,4,7}
```

```
{2,1.3,-9.5,4,7}
```

Die Daten erhalten also in ihrer Gesamtheit den Namen `daten1`. Sie sollen jetzt in Tabellenform, das heißt untereinander aufgelistet werden:

```
daten1tabelle=daten1//TableForm
```

```
2
1.3
-9.5
4
7
```

Im nächsten Beispiel sollen Daten eingegeben und dann paarweise zusammengefaßt werden. Dazu erzeugen wir eine Mathematica-Liste namens `daten2` mit einer geraden Anzahl von Daten. Die Anzahl kann mit der `Length`-Anweisung überprüft werden. Die folgende `Partition`-Anweisung bewirkt eine Zusammenfassung zu Datenpaaren. Das Semikolon am Ende einer Programmzeile verhindert die Anzeige einer Ergebnisausgabe in Mathematica:

```
daten2={1,8,3,6,2,11,4,14,5,7};  
n=Length[daten2]
```

```
10
```

```
daten2paar=Partition[daten2,2]
```

```
{{1,8},{3,6},{2,11},{4,14},{5,7}}
```

```
daten2paar//TableForm
```

```
1      8
3      6
2     11
4     14
5      7
```

Auch eine Zusammenfassung zu drei oder mehreren Werten ist mit `Partition` möglich. Die Angabe dafür wird nach dem Komma innerhalb der Klammer der `Partition`-Anweisung geschrieben. Aber Vorsicht, denn unsere ursprüngliche Liste hatte 10 Werte, so daß bei einer Aufteilung in je 3 Werte der letzte Wert entfällt!

```
daten2tripel=Partition[daten2,3];  
daten2tripel//TableForm
```

| | | |
|---|----|----|
| 1 | 8 | 3 |
| 6 | 2 | 11 |
| 4 | 14 | 5 |

Einer der wichtigsten Vorgänge beim Aufbereiten von Daten ist deren Sortierung nach verschiedenen Kriterien. Mathematica stellt dazu den Sort-Befehl zur Verfügung. Zunächst folgt die einfachste Variante für das Beispiel unseres Datensatzes daten1:

```
daten1sort=Sort[daten1]
```

```
{-9.5,1.3,2,4,7}
```

Dann sortieren wir nach absteigender Größe:

```
daten1sort=Sort[daten1,Greater]
```

```
{7,4,2,1.3,-9.5}
```

Auch die Datenpaare können sortiert werden und zwar zunächst nach der ersten Spalte (Mathematica sortiert hier mit höherer Priorität nach der ersten Spalte, dann mit geringerer Priorität nach der zweiten Spalte usw.).

```
daten2psort=Sort[daten2paar];  
daten2psort//TableForm
```

| | |
|---|----|
| 1 | 8 |
| 2 | 11 |
| 3 | 6 |
| 4 | 14 |
| 5 | 7 |

Dann sortieren wir diese in umgekehrter Reihenfolge ebenfalls nach der ersten Spalte:

```
daten2rev=Reverse[daten2psort];  
daten2rev//TableForm
```

```
5      7  
4      14  
3      6  
2      11  
1      8
```

Wir wollen jetzt nach der zweiten Spalte sortieren. Dazu vertauschen wir die Spalten, sortieren dann und machen die Vertauschung rückgängig. Da ein Vertauschen von Spalten in Mathematica nicht direkt möglich ist, sondern nur das Vertauschen von Zeilen, müssen wir zusätzlich noch durch Transponieren die Spalten als Zeilen darstellen:

```
B=Transpose[daten2rev];  
B//MatrixForm
```

```
( 5  4  3  2  1 )  
 ( 7 14  6 11  8 )
```

Die Zeilen werden nun vertauscht:

```
{ B[[1]], B[[2]] } = { B[[2]], B[[1]] };  
B//MatrixForm
```

```
( 7 14  6 11  8 )  
 ( 5  4  3  2  1 )
```

Die vertauschten Zeilen kann man als Spalten darstellen:

```
tausch1=Transpose[B];
```

tausch1//TableForm

| | |
|----|---|
| 7 | 5 |
| 14 | 4 |
| 6 | 3 |
| 11 | 2 |
| 8 | 1 |

Das Sortieren geschieht nun nach dem von uns gewünschten Kriterium:

B=Sort[tausch1];

B//TableForm

| | |
|----|---|
| 6 | 3 |
| 7 | 5 |
| 8 | 1 |
| 11 | 2 |
| 14 | 4 |

Nun wird die Vertauschung wieder rückgängig gemacht:

B=Transpose[B];

{ B[[1]], B[[2]] } = { B[[2]], B[[1]] };

tausch2=Transpose[B];

tausch2//TableForm

| | |
|---|----|
| 3 | 6 |
| 5 | 7 |
| 1 | 8 |
| 2 | 11 |
| 4 | 14 |

Die letzte Tabelle zeigt das gewünschte Resultat.

Oftmals liegen uns Daten zur Auswertung in einer Tabelle vor, die aber leider nicht ganz vollständig ist. Wir sprechen von „missing values“, wenn wir die fehlenden Werte eines Datensatzes meinen. Missing values entstehen zum Beispiel durch unvollständig ausgefüllte Fragebogen, messtechnische Probleme bei bestimmten Werten, fehlerhafte Eingaben bei der Datenerfassung und so weiter. Wir müssen in vielen Fällen solche einzelnen Datensätze, die fehlende Werte enthalten, aus einer bestimmten statistischen Untersuchung herausnehmen. Für unsere Datentabelle bedeutet dies, daß wir unter Umständen eine komplette Datenzeile entfernen müssen, sofern sie einen fehlenden Wert, oder ein Zeichen enthält (meist „.“), das von uns per Konvention als missing value gewertet wird.

Wir erstellen im nächsten Beispiel eine Datenmatrix mit einem missing value, gekennzeichnet mit einem Punkt, und zeigen dann, wie man Datenzeilen löscht.

```
M={{1,2,3,11},{5,.,9,15},{2,5,8,12},{2,2,4,10}};
```

```
M//MatrixForm
```

$$\begin{pmatrix} 1 & 2 & 3 & 11 \\ 5 & . & 9 & 15 \\ 2 & 5 & 8 & 12 \\ 2 & 2 & 4 & 10 \end{pmatrix}$$

Die zweite Zeile von M wird durch folgende Drop-Anweisung gelöscht:

```
M2=Drop[M,{2}];
```

```
M2//MatrixForm
```

$$\begin{pmatrix} 1 & 2 & 3 & 11 \\ 2 & 5 & 8 & 12 \\ 2 & 2 & 4 & 10 \end{pmatrix}$$

Falls man anstelle von {2} nur 2 schreibt, so werden die ersten beiden Zeilen gelöscht und mit -2 die letzten beiden Zeilen. Das Löschen einer bestimmten Spalte der Datenmatrix erfolgt durch das Löschen der

entsprechenden Zeile der transponierten Matrix:

```
M3=Transpose[Drop[Transpose[M],{2}]];  
M3 //MatrixForm
```

$$\begin{pmatrix} 1 & 3 & 11 \\ 5 & 9 & 15 \\ 2 & 8 & 12 \\ 2 & 4 & 10 \end{pmatrix}$$

Es kann mitunter nötig sein, aus einer gegebenen Datentabelle einen bestimmten Bereich herauszugreifen. Die Anweisung dazu lautet „ColumnTake“ und befindet sich im Paket „Statistics`DataManipulation“¹, welches wir nun laden. Die Names-Anweisung zeigt uns die verschiedenen Befehle innerhalb des Pakets:

Hinweis: Die Zeichen ` erzeugen Sie durch die Taste ` (in der oberen Reihe des Hauptfeldes der Tastatur) zusammen mit der Shift-Taste und danach die Leertaste.

```
Needs["Statistics`DataManipulation`"]
```

```
Names["Statistics`DataManipulation`*"]
```

```
{BinCounts, BinLists, BooleanSelect, CategoryCounts,  
  CategoryLists, ColumnDrop, ColumnJoin, ColumnTake,  
  CumulativeSums, DropNonNumeric, DropNonNumericColumn,  
  Frequencies, LengthWhile, Ordering, QuantileForm,  
  RangeCounts, RangeLists, RowJoin, TakeWhile}
```

Wir wählen beispielsweise den Bereich von Spalte 2 bis Spalte 4 der Matrix M. Mit Hilfe von Transponierungen kann man analog auch Zeilenbereich auswählen.

```
ColumnTake[M,{2,4}]/MatrixForm
```

$$\begin{pmatrix} 2 & 3 & 11 \\ . & 9 & 15 \\ 5 & 8 & 12 \\ 2 & 4 & 10 \end{pmatrix}$$

Oft ist es hilfreich sich Daten durch den Zufallsgenerator von Mathematica erzeugen zu lassen. Wir greifen unserer Besprechung der Verteilungen im nächsten Kapitel hier schon etwas vor, denn es geht uns zunächst nur um den technischen Aspekt der Erzeugung von Zufallszahlen als Daten. Wir wollen die Erzeugung und Bearbeitung von Zufallszahlen am Beispiel sogenannter „normalverteilter“ Daten vorführen. Die Anweisung zum Erzeugen einer Zufallszahl lautet `Random[Typ, Bereich]`. Schreibt man nur `Random[]`, so wird eine Zufallszahl zwischen 0 und 1 erzeugt. Dabei handelt es sich um sogenannte „gleichverteilte“ Zufallszahlen. Wir wollen normalverteilte Zufallszahlen erzeugen und brauchen daher die `NormalDistribution`-Anweisung, um den Verteilungstyp festzulegen. Wir müssen zuerst das Paket `Statistics`Master`` laden. Den `Statistics`Master`` sollten Sie künftig immer bei statistischen Berechnungen zu Beginn einer neuen Mathematica-Session laden.

```
Needs["Statistics`Master`"]
```

Mit den folgenden drei Anweisungen wird jeweils eine Variable definiert, die bei jedem Aufruf eine Zufallszahl mit der angegebenen Verteilung erzeugt. Die Zufallszahlen folgen hier einer Normalverteilung, deren Parameter, die in der Klammer stehen, sich allerdings unterscheiden. Auf die Bedeutung dieser Zahlenangaben gehen wir im Kapitel über die Verteilungen (bzw. im Anhang A) genauer ein. Hier sei nur erwähnt, daß die beiden Zahlenwerte den „Mittelwert“ und die „Standardabweichung“ der erzeugten Zufallsdaten charakterisieren.

(* Datensatz aus Zufallszahlen erzeugen *)

```
x1:=Random[NormalDistribution[175,5]]
```

```
x2:=Random[NormalDistribution[172,8]]
```

```
x3:=Random[NormalDistribution[178,7]]
```

Wie erzeugen eine Tabelle mit drei Spalten, je eine Spalte für eine der Variablen x_1 , x_2 und x_3 . Durch die folgende Programmzeile wird mit jeder der durch die Table-Anweisung (vergleiche Anhang B) erzeugten 100 Datenzeilen der Zufallsgenerator 3 mal aufgerufen. Wenn Sie die Ausgabe sehen wollen, lassen Sie das Semikolon am Ende der nächsten Zeile weg.

```
V=Table[{x1,x2,x3},{100}];
```

Oft möchte man aus ein und der selben Tabelle (hier V) einige Zeilen zufällig herausgreifen. Das Ergebnis sei eine Matrix mit Namen VA. Dies ist beispielsweise wie folgt möglich: Zuerst definieren wir eine Funktion h, die bei ihrem Aufruf jeweils eine („gleichverteilte“) ganzzahlige Zufallszahl im Bereich zwischen 1 und 100 liefert.

```
h:=Random[DiscreteUniformDistribution[100]]
```

Durch die Wahl von k (hier 10) können Sie die Anzahl der ausgewählten Zeilen selbst bestimmen. In der Matrix VA stehen die gewünschten zufällig ausgewählten Zeilen der Matrix V.

```
k=10;
```

```
M={};
```

```
While[Length[M]<=k,M=Union[M,{h}]]
```

```
L=If[Length[M]>k,Drop[M,1],M];
```

```
Wahl[i_]:=V[[i]]
```

```
VA=Map[Wahl,L];
```

```
VA//MatrixForm
```

```
( 174.128 181.132 173.709 )  
( 172.574 167.725 174.883 )  
( 175.84 176.538 166.546 )  
( 179.876 173.833 177.029 )  
( 177.775 169.755 180.911 )  
( 171.332 169.018 179.915 )  
( 168.399 197.388 181.541 )  
( 168.127 175.467 186.944 )  
( 173.623 174.331 188.629 )  
( 170.109 173.902 177.425 )
```

Wir speichern diese Zufallsmatrix „VA“ in einer Datei „VAfile“. Legen Sie mit Hilfe des Windows-Explorer (Dateimanager) einen Ordner „Daten“ auf dem Verzeichnis C:\ Ihrer Festplatte an, in welchem Sie diese und die nun folgenden Dateien abspeichern können. Dann können Sie diesen Ordner als Arbeitsverzeichnis in Mathematica festlegen:

```
SetDirectory["C:\Daten"]
```

```
Put[VA,VAfile]
```

So könnte die Matrix später wieder unter dem Namen x geladen werden: `x=Get[VAfile]` und `x//MatrixForm`. Wir wollen dies aber erst später im Kapitel über Streudiagramme verwenden.

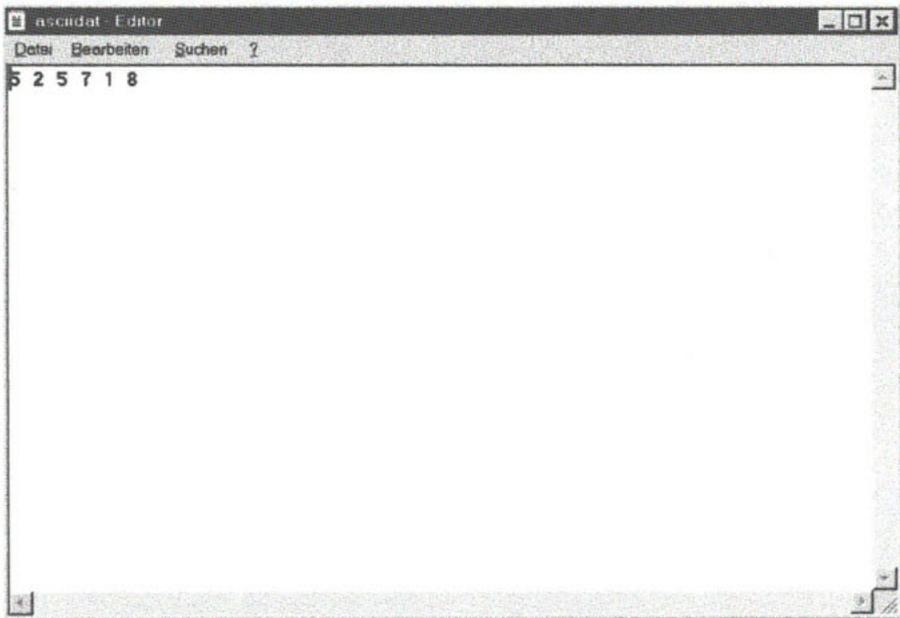
Bisher hatten wir die Daten selbst in Mathematica eingegeben, oder wir haben sie direkt von Mathematica in Form von Zufallszahlen erzeugen lassen. Nun wollen wir Daten lesen, die schon als Datei vorliegen, sei es, daß sie von anderen Programmen erstellt und gespeichert wurden, oder daß sie mit Mathematica aus einer vorangegangenen Berechnung als Datei abgelegt wurden.

Es gibt ein Zusatzpaket für Mathematica „Mathematica Link for Excel“, welches den Datenaustausch zwischen Mathematica und Excel direkt gestattet. Auf diesen Aspekt gehen wir aber erst am Schluß dieses Abschnitts ein, da wir nicht voraussetzen können, daß allen

Lesern dieses Paket zur Verfügung steht.

Zunächst erstellen wir einige Beispieldatensätze mit Hilfe eines ASCII-Editors, wie zum Beispiel dem „Notepad“ aus Windows, oder für diejenigen Leser, die mit Excel arbeiten, auch mit diesem Programm. Gehen Sie dazu folgendermaßen vor:

Sie haben oben schon ein Verzeichnis „C:\Daten“ erstellt. Nun erstellen Sie die erste ASCII-Datei mit dem Editor (Notepad), den Sie über die Menüpunkte ->Start ->Ausführen ->Zubehör ->Editor in Windows 95/98 aktivieren. Geben Sie die Daten 5 2 5 7 1 8 jeweils durch Leerzeichen getrennt ein. Speichern Sie dann diese Werte als Datei „asciidat.txt“ im Verzeichnis c:\Daten ab, wobei die Dateierweiterung .txt vom Editor selbst erzeugt wird. Überzeugen Sie sich mit dem Windows-Explorer (Dateimanager), daß die Datei erstellt wurde. Das Bild zeigt die Ansicht des Editors.



Da wir nicht immer davon ausgehen können, daß wir nur eine einzige Datenzeile in einer Datei vorfinden, deren Werte durch Leerzeichen getrennt sind, wollen wir nun die anschließend benötigten Dateien ebenfalls erstellen.

Die erste Abwandlung wären Daten mit missing values angezeigt durch

einen Punkt. Die Daten seien 5 2 . 6 7 . 1 . 8, also 9 durch Leerzeichen getrennte Werte, wovon 3 missings sind. Wir geben dies ebenfalls im Editor ein und speichern unter „asciidatmiss“ in c:\Daten.

Wir benötigen dann eine Datei, deren Elemente in mehreren Zeilen stehen, und die wir später zeilenweise einlesen wollen.

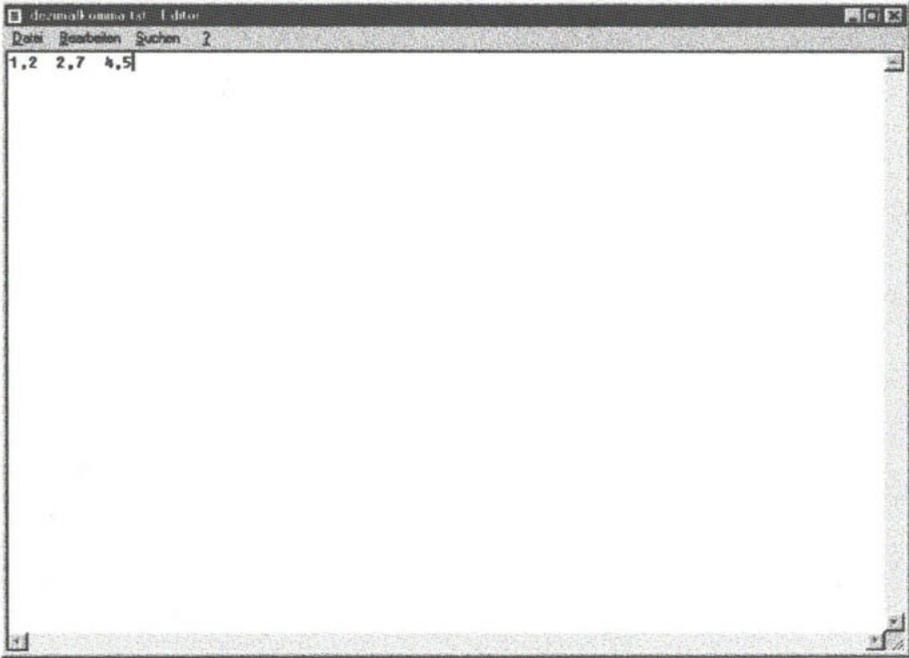
Die Daten seien untereinander in 3 Zeilen stehend :

```
1 5  
3 4  
7 9
```

Wir geben sie im Editor ein und speichern sie unter dem Namen „asciirec.txt“ in unserem Datenverzeichnis c:\Daten

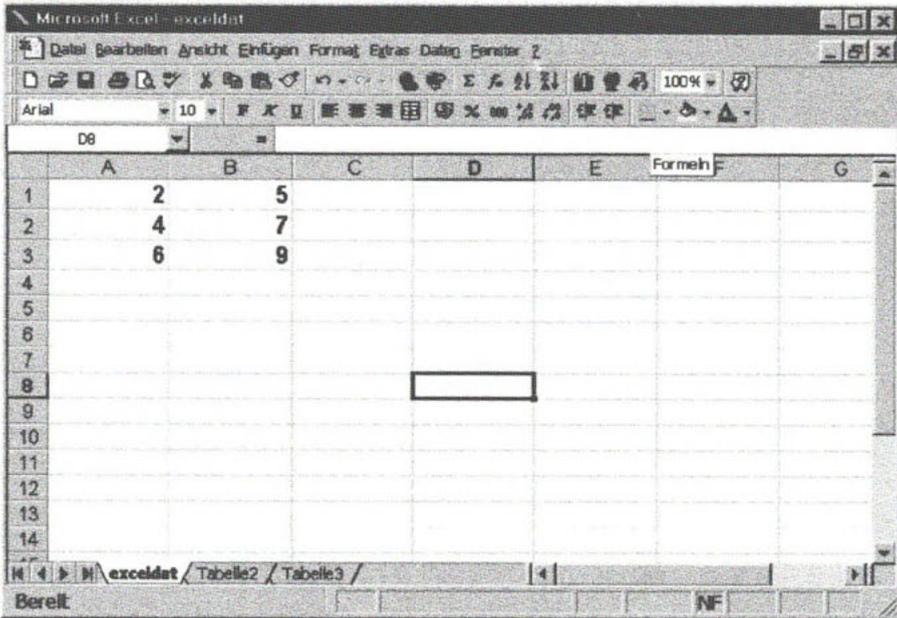
Wir wollen als nächstes Daten erzeugen, die durch Komma getrennt sind. Die Daten seien 2,5,7,9,1. Wir speichern diese mitsamt der Kommata eingegebenen Daten unter „asciikomma.txt“ in c:\Daten.

Oft treten Zahlen auf, die statt des Dezimalpunktes ein Dezimalkomma aufweisen. Erstellen Sie daher im Editor eine kurze Zahlenreihe mit Dezimalkommata. Die Zahlen sind durch Leerzeichen getrennt, wie Sie in der folgenden Ansicht des Editorfensters erkennen können. Speichern Sie dann die Zahlen in der Datei „dezimalkomma“ im Verzeichnis C:\Daten



Für ein weiteres Beispiel brauchen wir die Daten 1001100101010 , die ohne Leerzeichen im Editor erstellt werden und in der Datei „dual.txt“ im üblichen Verzeichnis C:\Daten abgespeichert werden.

Abschließend sollen diejenigen Leser, die mit Excel arbeiten, jedoch „Mathematica Link for Excel“ nicht besitzen, noch eine Datei mit Excel erzeugen. (Wer Excel nicht kennt, kann ab der nächsten Mathematica-Anweisung weiterarbeiten.) Starten Sie Excel und geben sie die Beispieldaten wie im unten stehenden Bild ein.



Beim Speichern beachten Sie hier bitte folgendes: Sie müssen nach der Eingabe der im Bild gezeigten Daten in Excel unter dem Menüpunkt „Datei“ den Unterpunkt „speichern unter“ wählen und dort „exceldat“ als Dateinamen auswählen. Der Dateityp ist als „Formatierter Text (Leerzeichen getrennt)“ auszuwählen. Achten Sie auch darauf, daß das Verzeichnis auf C:\Daten eingestellt ist. Noch ein Hinweis für Benutzer von MS-Access: Auch hier lassen sich die Tabellen über „Speichern unter/Exportieren“ im Textformat abspeichern. Die Datei erhält dann die Extension .txt und kann in Mathematica eingelesen werden. Außerdem können die Daten über Excel und dann, wie unten gezeigt wird, in Mathematica transferiert werden.

Nach diesen Vorbereitungen können wir nun darangehen, die erstellten Dateien in Mathematica einzulesen. Falls Sie zwischenzeitlich Mathematica verlassen haben und nun hier wieder beginnen, müssen Sie zuerst die Verzeichnispfade neu angeben. Die Directory-Anweisung zeigt uns jedenfalls das aktuelle Arbeitsverzeichnis an (Abweichungen von unserer Ausgabe sind möglich).

Directory[]

C:\Programme\Wolfram Research\Mathematica\3.0

Wie setzen das Arbeitsverzeichnis nun auf c:\Daten, falls dies oben noch nicht angezeigt wurde.:

SetDirectory["c:\Daten"];

Directory[]

c:\Daten

Nun können Sie nachsehen, welche Dateien Sie dort schon gespeichert haben:

FileNames[]

asciidat.txt, asciikomma.txt, asciirec.txt, exceldat.prn,....

Die Folge der Daten aus unserer ersten Datei asciidat.txt soll jetzt in eine Mathematica-Liste namens einfacheliste eingelesen werden. Der Befehl hierzu lautet ReadList. Da es sich um ganze Zahlen handelt, die wir einlesen wollen, geben wir nach dem Dateinamen noch die Option „Number“ im ReadList-Befehl an. Für „reelle“ Zahlen (Fließkomma-Zahlen) wählen Sie statt „Number“ die Option „Real“ (bitte nachvollziehen!)

einfacheliste=ReadList["asciidat.txt",Number]

{5,2,5,7,1,8}

Die der Daten aus der zweiten Datei asciidatmiss.txt enthielten missing values. Auch diese Datei soll jetzt in eine Mathematica-Liste, ebenfalls wieder mit „einfacheliste“ bezeichnet, eingelesen werden. Da es sich sowohl um Zahlen, als auch um Zeichen handelt, die wir einlesen wollen, geben wir nach dem Dateinamen noch die Option „Word“ im ReadList-Befehl an.

```
einfacheliste=ReadList["asciidatmiss.txt",Word]
```

```
{5, 2, ., 6, 7, ., 1, ., 8}
```

Um mit dieser Liste sinnvoll weiterarbeiten zu können, wollen wir zunächst die Anzahl der missings, gekennzeichnet mit „.“, auszählen. Dann wollen wir diese aus der Liste entfernen und schließlich die verbleibenden Zahlen, die noch im Word-Format als Text in Mathematica stehen, in ein Zahlenformat umwandeln:

```
anzmiss=Count[einfacheliste,"."]
```

```
3
```

Es waren also 3 missing values vorhanden. Wir entfernen sie mit Hilfe der Select-Anweisung und erhalten eine Liste ohne missing values, die wir mit „omv“ bezeichnen. Dabei werden aus der ursprünglichen Liste nur diejenigen Elemente behalten (Select), die ungleich (Zeichen !=) dem Punkt sind. Hinweis: In der folgenden Anweisung benutzen wir eine sogenannte „reine Funktion“. Eine Erklärung solcher Mathematica-Anweisungen finden Sie im Anhang B.

```
omv=Select[einfacheliste,#!="."&]
```

```
{5, 2, 6, 7, 1, 8}
```

Mit diesen Werten kann man noch nicht rechnen. Erst nach Umwandlung in ein Zahlenformat mit Hilfe der ToExpression-Anweisung ist beispielsweise eine Potenzierung möglich:

```
Zahlen=ToExpression [omv]
```

```
{5, 2, 6, 7, 1, 8}
```

```
Zahlen^2
```

```
{25, 4, 36, 49, 1, 64}
```

Man kann die Folge der Daten aus der Datei einfacheliste beim Einlesen auch in eine Liste von Paaren umwandeln, falls dies erforderlich ist:

```
paarweiseliste=ReadList["asciidat.txt",{Number,Number}]
```

```
{{5,2},{5,7},{1,8}}
```

Befinden sich in einer Datei mehrere Datenzeilen, von denen man jede gleichsam als einen separaten Datensatz in eine Liste einlesen will, so kann man, wie im folgenden für das Beispiel der Datei asciirec.txt gezeigt, eingeben:

```
listerec=ReadList["asciirec.txt",Number,RecordLists->True];  
listerec //TableForm
```

```
1      5  
3      4  
7      9
```

Oft liefern Editoren die Daten durch Kommata getrennt. Das Trennzeichen müssen wir Mathematica mit „WordSeparators“ erklären. Wie oben kann man auch hier erst nach Umwandlung in das Zahlenformat weiterrechnen. Ein Versuch die Werte als Zahlen wie bisher einzulesen scheitert, wie Sie im nächsten Beispiel sehen. Erst in der darauf folgenden Eingabe mit dem Word-Format und mit WordSeparators gelingt das Einlesen der durch Komma getrennten Zahlen:

```
listekomma=ReadList["asciikomma.txt",Number]
```

```
Read::readn:
```

```
Syntax error reading a real number from asciikomma.txt  
{2,$Failed}
```

```
listekomma=ReadList["asciikomma.txt",Word,  
WordSeparators -> {"",""}]
```