Dokumentation zum Statistikprogramm `statist`

Dirk Melcher
Institut für Umweltsystemforschung, Universität Osnabrück
Artilleriestr. 34, 49069 Osnabrück
email: Dirk.Melcher@usf.Uni-Osnabrueck.DE

Dokumentation vom 31.1.97 (für `statist v0.1`)
16.2.1998 kleine Anmerkungen von Bernhard Reiter
18.8.1998 Selektive Aktualisierungen von B.Reiter

Achtung! Diese Dokumentation enthält veraltete Informationen!

1 Einleitung

Zuerst einmal: Das Programm `statist` ist ein Laienprogramm, also keine überzogenen Erwartungen! Es wurde geschrieben, um einfache, alltägliche Statistik ausführen zu können, ohne jedesmal einen „Dinosaurier“ wie SAS oder SPSS bemühen zu müssen. Es soll aber keineswegs dazu dienen, um wirklich aufwendige statistische Verfahren durchzuführen.

Das Programm hat den Anspruch

1. *einfach* und schnell zu bedienen zu sein
2. wirklich portabel zu sein. Daher wurde auf aufwendige Ein/Ausgabe, Fenstertechnik, Menüschnüddel u.v.a. verzichtet
3. schnell und simpel um zusätzliche Routinen zu erweitern zu sein
4. halbwegs speicherschonend zu sein

`statist` befindet sich noch in der beta-Phase. Wie bei jedem Programm üblich wird auch bei `statist` ausdrücklich keine Garantie auf richtige Ergebnisse übernommen!

2 Installation

Die Installation ist denkbar einfach: Man kopiert das Programm eben dorthin, wo man's haben will. Günstig wäre es natürlich, wenn zu diesem Verzeichnis auch ein Pfad gelegt wäre ... Es gibt nur einen einzigen Haken: für jede Spalte (s. Abschnitt 5), die eingelesen wird, legt `statist` eine temporäre Datei an. Wenn man Dateien mit vielen Spalten einliest, kann unter DOS die Anzahl der geöffneten Dateien zu groß werden. In diesem Fall, falls es nicht schon geschehen ist, in der `config.sys` mit

dem Befehl `FILES=40` die Anzahl der Dateipuffer auf 40 oder was anderes hochsetzen. Außerdem ist für die DOS-Version darauf zu achten, daß das Verzeichnis `c:\tmp` existiert, andernfalls mit `md c:\tmp` das Verzeichnis anlegen oder die entsprechende Zeile in den `statist` Quellen ändern und neu übersetzen (Datei `data.c` am Ende von Funktion `makefilename()`).

3 Aufruf

`statist [-help -silent -log -nobell -nofile -noplot -thist]` *`Datenfile`*

Die Option `-help` gibt einen (sehr kurzen) Hilfetext aus. Wie ein Datenfile auszusehen hat, wird in Abschnitt 5 beschrieben. Die Optionen `-log` bewirkt, daß die Ergebnisse in der Datei `statist.log` protokolliet werden und die Option `-nobell` bewirkt, daß bei Fehlern und Warnungen kein Piepton ertönt. Die Optionen `-silent`, `-nofile`, `-noplot` und `-thist` werden in den Abschnitten 7, 5, 4 und 8 beschrieben.

4 statist und gnuplot

`gnuplot` ist ein interaktives Graphikprogramm zur Darstellung von Daten und Funktionen. Es kann nicht nur im Dialog, sondern auch mit Hilfe eines Skripts gesteuert werden.

Läuft `statist` unter UNIX, dann werden gewisse Funktionen von `statist` durch eine `gnuplot`-Graphik unterstützt. Voraussetzung dafür ist natürlich, daß das `gnuplot` installiert und sich das entsprechende Verzeichnis in der `PATH`-Variablen befindet.

Unter DOS wird von `statist` eine Kommandodatei namens `stat_gpl.com` erzeugt, mit deren Hilfe nach Beendigung von `statist` eine `gnuplot`-Graphik erzeugt werden kann. Nimmt man eine Windows-Version von `gnuplot`, dann kann man bequem unter MS-Windows `statist` in einer DOS-Box laufen lassen und im anderen Fenster `gnuplot`.

Momentan werden folgende Funktionen (s. Abschnitt 8) unterstützt:

- BOX- UND WHISKER Plot (Median, Standardabweichung etc.)
- Lineare Regression (2- und 3-dimensional)
- Polynomregression
- Test auf Normalverteilung (Häufigkeitshistogramm + Summenfunktion)
- Probitanalyse

Außerdem kann man unter dem Menüpunkt `Datenverwaltung | gnuplot-Befehle eingeben` direkt `gnuplot`-Befehle eingeben, um so eine Graphik interaktiv zu verfeinern oder für den Ausdruck fertig machen (nur unter UNIX). Will man *keine* `gnuplot`-Graphik haben, z.B. weil man im Batch-Betrieb arbeitet (s. Abschnitt 7) oder weil der Rechner zu langsam ist, dann kann das Programm mit der Option `-noplot` aufgerufen werden.

5 Daten

Daten werden dem Programm grundsätzlich in Form von simplen ASCII-Dateien zugeführt. Entweder ruft man das Programm mit einer ASCII-Datei auf, oder das Programm fragt gleich beim Aufruf nach dem Namen einer Datendatei. Ohne Datendatei tut sich nix, es sei denn, man gibt beim Aufruf die Option `-nofile` an, um die Daten direkt über die Tastatur einzugeben (Menüpunkt `Datenverwaltung | Spalte vom Terminal einlesen`). Das macht aber eigentlich nur selten Sinn. Die Option ist mehr dafür gedacht, um unter UNIX Menübefehle zusammen mit den Daten zu `statist` zu pipen.

Eine Datendatei besteht aus einer oder mehreren Zahlenspalten (momentan max. 25). Die Zahlen in der Datei müssen durch ein oder mehrere Leerzeichen voneinander getrennt werden. Es ist auch erlaubt, eine Datei mit verschieden langen Spalten einzugeben. In diesem Fall muß aber in der *kürzeren* Spalte (in der sozusagen Zahlen „fehlen“) ein 'M' (Vor `statist v0.12` mußte dies ein '.' sein. Kann in Quell-Datei `statist.h` bei `#define NODATA` vor Übersetzung von `statist` geändert werden.) stehen, damit `statist` weiß, welche Zahl welcher Spalte zuzuordnen ist. Beispiel:

```
# Beispiel Datendatei fuer statist
  1  3   5  6
  7  8   9 10
 11 12  13 14
 15  M  16  M
```

Wie man dem Beispiel entnehmen kann, sind auch Kommentarzeilen nach Art von `gnuplot` zugelassen, die mit einem '`#`' in der ersten Spalte eingeleitet werden. Leerzeilen werden ebenfalls ignoriert.

Genauso gut hätten die Daten auch so eingetippt werden können:

```
# Beispiel Datendatei fuer statist
# Ich glaube, hier ist was schief gelaufen
  1     3  5  6
 7  8   9      10
 11 12     13 14
15  M            16  M
```

Im Programm werden die Spalten jeweils Variablen zugeordnet. Standardmäßig wird die 1. Spalte mit 'a', die 2. mit 'b', die 3. mit 'c' usw. bezeichnet. Um bei vielspaltigen Datendateien den Überblick zu behalten, ist es aber auch möglich, die Spalten einzeln zu benennen. Das hat den Vorteil, daß man sich dann nicht merken muß, in welcher Spalte eine bestimmte Variable steht. Dies ist innerhalb der ersten Zeile der Datendatei möglich. Die Zeile muß mit einem '`#`' als Kommentarzeile gekennzeichnet sein, gefolgt von einem '`%`'. Dann werden den Zeilen folgendermaßen Namen zugeordnet (Beispiel):

```
#% kow kaw ec50
0.34 4.56 0.23
```

```
1.23 5.45 6.76
6.78 1.34 9.60
```

Dabei ist folgendes zu beachten:

1. Es müssen genauso viele Variablennamen angegeben werden, wie Spalten vorhanden sind.

2. Als Spaltennamen dürfen *nur* Buchstaben, Ziffern und '_' benutzt werden

Ältere Versionen von statist verwendeten die Zeichenkombination `#!`. (Das alte Verhalten läßt sich leicht wieder herstellen, wenn vor dem Kompilieren von statist in der Datei `data.c` in der Funktion `parsecomment()` die Konstante `var_id` geändert wird.)

Manchmal arbeitet man mit Daten, deren einzelne Objekte benannt sind. Den Objekten entspricht in einem `statist`-File eine Zeile. Standardmäßig „duldet" `statist` lediglich Dateien, die nur Zahlenspalten und Kommentarzeilen enthalten. Um jedoch auch mit Dateien zu arbeiten, welche alphanumerische Spalten enthalten, kann man diese Spalten explizit mit einem `$`-Zeichen kennzeichen, so daß `statist` nicht versucht, diese als Zahl zu interpretieren:

```
#% $name kow kaw ec50
2,4-D   0.34 4.56 0.23
Benzol  1.23 5.45 6.76
Atrazin 6.78 1.34 9.60
```

Zu beachten ist, daß in den alphanumerischen Spalten kein Leerzeichen stehen darf, da dies als neue Spalte interpretiert würde! Um beim obigen Beispiel zu bleiben: `2,4 D` wäre falsch.

Bei einigen Prozeduren ist die Anzahl der verwendeten Spalten variabel. Z.B. können bei der multiplen linearen Regression 2 oder auch 10 Spalten angegeben werden. Will man für eine Prozedur alle eingelesenen Spalten verwenden, so tippt man, sobald das Programm nach der Anzahl der Spalten fragt, einfach 'alle' ein. Damit entfällt die explizite Zuordnung der Spalten zu den Variablen.

Man kann auch Daten aus mehreren Dateien gleichzeitig einlesen und somit Daten aus verschiedenen Dateien kombinieren. Dazu wählt man den Menüpunkt `Datenverwaltung | Neue Datei einlesen`.

6 Menü

Durch das Programm wird man mit einem *sehr* einfachen Menü geführt. Grundzätzlich werden Menüpunkte mit Ziffern gewäehlt. '0' führt immer in die nächst höhere Menüebene und beendet konsequenterweise im Hauptmenü das Programm. Ein Schmankerl gibt es aber doch: Man kann immerhin jede Benutzerabfrage mit der Returntaste unterbrechen und landet dann wieder in eines der Menüs.

Wenn man eine Statistikprozedur aufruft, wird man aufgefordert, den Spalten Variablen zuzuordnen, das ist eigentlich selbsterklärend.

7 Batch-Betrieb

Wenn man zahlreiche Datensätze auf immer die gleiche Art und Weise durch `statist` durchnudeln möchte und es einem auf die Nerven geht, sich immer wieder durchs Menü durchzuhangeln, gibt es eine kleine Hilfe: Da das Programm nur mit Standard-Ein/Ausgabe arbeitet, kann man sich eine kleine „Antwort"-Datei basteln. Hierin schreibt man exakt das hinein, was man sonst als Eingabe für `statist` eintippen würde, also in der Regel nur die Zahlen/Buchstaben, die man als Auswahl für das Menü und die Spalten eingibt. Genauso kann man die Ausgabe in eine Datei umleiten, um sich dann die Ergebnisse in Ruhe anzusehen oder aber alternativ die Option `-log` angeben (was bewirkt, daß das Ergebnis nicht nur in die Datei `statist.log` sondern auch auf den Bildschirm ausgegeben wird). Mit der Option `-silent` wird die Ausgabe von Dialogtexten unterdrückt, so daß nur noch das Ergebnis der Berechnungen ausgegeben wird. Außerdem fällt dann die Aufforderung zum Drücken der `Return`-Taste zum Fortfahren des Programmes weg. Will man z.B. im Batch-Modus eine lineare Regression mit den Spalten a und b einer Datei durchführen, dann sähe die „Antwort"-Datei so aus (Vergleiche hierzu die Eingabe beim normalen Menü-Betrieb):

```
2
1
a
b
0
0
```

Der Aufruf für den Batch-Betrieb könnte dann also folgendermaßen aussehen:

```
statist daten.dat -silent < statist.ant > statist.log
```

bzw.

```
statist daten.dat -silent -log < statist.ant
```

8 Funktionen

Momentan stehen folgenden Statistikfunktionen zur Verfügung. (Die Angaben in Klammern beziehen sich auf die Literatur, denen der Algorithmus entnommen wurde.):

1. Lineare Regression
2. Rank-Korrelationskoeffizient von Spearman [1, S. 175 ff]
3. Multiple lineare Korrelation [3, S. 77 ff]
4. Partielle lineare Korrelation (max. 5 Variablen) [5, S. 82 f]
5. Polynomregression [3, S. 65 f]
6. Korrelationsmatrix der linearen Korrelationskoeffizienten

7. Korrelationsmatrix der Spearman'schen Korrelationskoeffizienten
8. Punkt-biserielle (lineare) Korrelation [1, S. 182 ff]
9. t-Test zum Vergleich zweier Mittelwerte aus Stichproben [1, S. 10 ff]
10. t-Test zum Vergleich zweier Mittelwerte bei paarweiser Anordnung der Stichproben [5, S. 175 f]
11. Test auf Normalverteilung (Kolmogoroff-Smirnoff-Lilliefors) [4, S. 100 ff]
12. χ^2-Vierfeldertafel [5, S. 200 ff]
13. χ^2-Mehrfachtafel [5, S. 209 ff]
14. U-Test von Mann und Whitney [5, S. 184 ff]
15. Zweistichprobentest von Wilcoxon [5, S. 340 ff]
16. Test von Kruskal und Wallis auf k unabhängige Stichproben [5, S. 337 ff]
17. Standardabweichung, Mittelwert, Median u.a.
18. Probitanalyse [5, S. 534 ff]
19. Log-Transformation (10er Logarithmus), Invertierung (1/x) und Sortieren
20. Elemenieren von vermuteten Ausreißern [2, S. 835]
21. Kreuz-Validierung multipler linearer Regression (noch experimentell!).

Bei Korrelations- bzw. Regressionsfunktionen wird immer zugleich ein Test auf signifikante Korrelation durchgeführt. Approximationen für t-Verteilung, Normalverteilung, χ^2-Verteilung und t-Verteilung wurden [3] entnommen.

Anmerkung zu den Funktionen:

- Beim Test auf Normalverteilung (Kolmogoroff-Smirnoff-Lilliefors) lautet die Hypothese H_0: die Daten sind normalverteilt. Diese Hypothese wird akzeptiert, wenn die Wahrscheinlichkeit für H_1 (die Daten sind nicht normalverteilt) *nicht* signifikant hoch ist. Die „Beweislast" liegt also bei H_1. Dies bedeutet, daß H_0 desto besser abgesichert ist, je *höher* das Signifikanzniveau α liegt, denn α gibt jetzt die Wahrscheinlichkeit für H_0 statt für H_1 an. Es geht hier also genau umgedreht wie bei den anderen Tests zu!

  Wählt man den Test auf Normalverteilung, so gibt `statist` zuerst ein Häufigkeitsdiagramm aus.

  Bei Angabe der Option `-thist` (oder auch `-noplot`, s. Abschnitt 4) wird dieses als Textgraphik dargestellt, ansonsten als `gnuplot`-Graphik.

Da beim KS-Lilliefors-Test die theoretisch erwartete Normalverteilungsfunktion mit der Summenhäufigkeitsfunktion der Daten verglichen wird, werden diese Funktionen graphisch dargestellt. Zwei waagerechte Linien zeigen die größte 'vertikale' Differenz der beiden Funktionen auf, welche die Prüfgröße D darstellt.

- Bei den t-Tests wird vorrausgesetzt, daß die Varianzen der Grundgesamtheiten, aus denen die Stichproben vorliegen, gleich groß sind. Wenn man paarweise angeordnete Meßwerte testen möchte (z.B. Vergl. des Gewichtes von männl. und weibl. Mäusen aus je einem Wurf, s. [5, S. 175 f]), dann wende man den t-Test zum Vergleich paarweise angeordneter Stichproben an.

- Beim χ^2-Vierfelder-Tafeltest gibt es zwei Möglichkeiten zur Eingabe der Daten:

  1. Wenn die beiden eingelesenen Spalten nur '0' oder '1' enthalten, bedeuted dies 'Merkmal nicht vorhanden' bzw. 'Merkmal vorhanden'. Dementsprechend werden die Merkmalskombinationen für die Vierfeldertafel ausgezählt. Um z.B. eine Vierfeldertafel für zwei Merkmale aufzustellen, könnte man folgende Datei eingeben:

     ```
     # Merkmale einer Blume 1=gross 2=rot
     1 0
     1 0
     1 1
     1 1
     0 1
     0 0
     ```

     `statist` stellt aus dieser Eingabe die Vierfeldertafel auf, wie dies in Tabelle 1 dargestellt ist.

Table 1: Beispiel für eine Vierfeldertafel für die Merkmale A und B.

	A vorhanden	A nicht vorhanden
B vorhanden	2	1
B nicht vorhanden	2	1

  2. Wenn die zwei Spalten aus je nur 2 Werten bestehehen, wird davon ausgegangen, daß die fertig ausgezählte Vierfeldertafel eingelesen worden ist. Die Werte würden dann also wie folgt eingegeben:

     ```
     # Tafel fuer Merkmale 'rot' und 'gross' einer Blume
     2 1
     2 1
     ```

- Beim χ^2-Mehrfachtafel-Test können im Gegensatz zur Vierfeldertafel Merkmale in mehrere Klassen dargestellt werden. Ein Beispiel hierfür wäre die Untersuchung der Verteilung der Merkmale 'Blattgröße' und 'Blütenfarbe' einer

Pflanze. Das Merkmal Blattgröße könnte z.B. in die Klassen 'groß', 'mittel' und 'klein'eingeteilt werden und die Blütenfarbe in die Klassen 'blau', 'rot' und 'weiß'. Im Gegensatz zur Vierfeldertafel werden bei diesem Test nur ausgezählte Tabellen von `statist` angenommen, also z.B.

```
# Tafel fuer die Merkmale 'Bluetenfarbe' und 'Blattgroesse'
# Spalten: Bluete blau   rot    weiss
# Zeilen   Blatt  gross  mittel klein
 29   11   6
273  191  64
  8   31   4
```

- Beim U-Test werden zwei Variable x und y daraufhin untersucht, ob sie sich signifikant voneineander unterscheiden. Er ist somit das parameterfreie Gegenstück zum t-Test. Beim U-Test erfolgt ein Test der Prüfgröße U auf Signifikanz mit Hilfe der Normalverteilung, wenn sowohl für x als auch y mindestens 8 Werte vorhanden sind, sonst benutzt `statist` eine Tabelle der kritischen Werte.

- Beim Test von Kruskal und Wallis handelt es sich wie beim U-Test um einen parameterlosen Test, bei dem geprüft wird, ob drei oder mehr unabhängige Stichproben der gleichen Grundgesamtheit entstammen. Dieser Test ist somit das Gegenstück zum parametrischen F-Test. Wenn die Stichproben jeweils mehr als 4 Werte enthalten, kann ein χ^2-Test durchgeführt werden, ansonsten muß die Prüfgröße H leider mit Hilfe von Tabellen getestet werden.

- Beim Zweistichprobentest von Wilcoxon handelt es sich ebenfalls um einen parameterlosen Test, bei dem zwei Zufallsvariablen x und y paarweise verglichen werden und ist somit daß parameterlose Gegenstück zum t-Test für paarweise angeordnete Stichproben. Er eignet sich z.B. für Fragestellungen, bei denen ein Objekt mit zwei verschiedenen Mitteln behandelt worden ist. x und y charakterisieren in diesem Fall die unterschiedliche Behandlung am gleichen Objekt. Die Hypothese H_0 lautet dann: Es gibt keine Unterschiede in der Behandling x und y.

  Für Stichproben < 25 wird eine Tabelle der kritischen Werte benutzt, ansonsten wird die Wahrscheinlichkeit mit Hilfe einer Approximation an die Normalvewrteilung berechnet.

- Der Punkt-biserielle Korrelationskoeffizient wird benutzt, wenn die Korrelation zwischen einem quantitativen Merkmal und einem alternativen Merkmal berechnet werden soll (Bsp.: Korrelation 'Durchmesser einer Blüte' – 'Blüte ist rot' ($\Longrightarrow$ ja/nein Entscheidung).

- Bei der log-Transformation wird eine neue Spalte erzeugt, welche die logaritmierten Werte einer eingelesenen Spalte enthält. Dies ist nützlich, wenn man z.B. eine log-lineare statt einer linearen Korrelation berechnen und/oder testen will. Das gleiche gilt analog für die Invertierungsfunktion 1/x, der Sortierfunktion und der Ausreißerfunktion.

- Unter dem Menuepunkt `Verschiedens |Ausreisser + Box-Whisker-Plot` wird via `gnuplot` ein sogenannter *Box-Whisker-Plot* [2, S. 835 ff] erstellt (s. Abb. 1). *Box-Whisker-Plots* sind gut geeignet, um auf einen Blick bestimmte Eigenschaften von Verteilungen zu erfassen. Zum Beispiel gibt die Lage des arithmetischen Mittelwertes im Vergleich zum Median einerseits und die Lage des Konfidenzintevalles des Medians zum 25%- und 75%-Quartil Aufschluß über die Schiefe einer Verteilung. Außerdem kann man potentielle Ausreißer mit einem Blick erkennen.

- Unter dem Menüpunkt `Regreesion und Korrelation` finden sich die Punkte `Kreuz-Validierung multipler linearer Regression` und `Randomisierung multipler linearer Regression`. Diese beiden Punkte dienen der Evaluierung der Prognosefähigkiet linearer Modelle [6, 7].

  Die prognostizierte Varianz Q^2 wird beim Menüpunkt `Kreuz-Validierung multipler linearer Regression` folgendermaßen berechnet: Ein Objekt wird aus dem Datensatz herausgenommen und die Regression mit den verbleibenden Daten durchgeführt[1]. Mit Hilfe der so ermittelten Regressionskoeffizienten a_i kann dann die abhängige Variable yo des fehlenden Objektes berechnet werden. Der so berechnete Wert kann als prognostizierter Wert yp bezeichnet werden. Dieses Verfahren wird für alle Datensätze angewendet, so daß für jeden gemessene Wert yo ein prognostizierter Wert yp existiert. Anschließend kann die prognostizierte Varianz Q^2 aus den yo, yp und dem Mittelwert $\bar{y}$ berechnet werden:

$$Q^2 = 1 - \frac{\sum_{i=1}^{n} (yo_i - yp_i)^2}{\sum_{i=1}^{n} (yo_i - \bar{y})^2} \tag{1}$$

  Als weitere Maßnahme zur Validierug wird von Wold die Randomisierung des Response-Vektors genannt (Menüpunkt `Randomisierung multiple linearer Regression`). Bei diesem verfahren werden die unabhängigen Variablen intakt gelassen, während der Vektor der y-Werte mittels Zufallsgenerator randomisiert wird. Dabei werden nicht die y-Werte selber geändert, sondern die Indizes des Vektors werden permutiert, die y-Werte werden also vertauscht. Dieses Randomisierung wird zahlreiche Male wiederholt und für jeden so manipulierten Datensatz das Bestimmtheitsmaß r^2 und die prognostizierte Varianz Q^2 berechnet. Die Verteilungen dieser Werte können in einem Histogramm dargestellt werden, so daß erkennbar wird, ob das r^2 bzw. Q^2 des originalen Datensatzes mit hoher Wahrscheinlichkeit Produkt eines 'Zufalls'-Datensatzes ist oder ob nicht. Der Benutzer kann wählen, wieviel Tupel und somit wieviele aus permutierten Datensätzen erzeugte r^2 und Q^2 prodiziert werden sollen. Dies kann bei größeren Datensätzen durchaus länger dauern! Zum Schluß werden zwei neue Spalten `rquad` (enthält die r^2 Werte) und `qquad` (enthält die Q^2 Werte) erzeugt. Diese Spalten können z.B. mit Hilfe eines Histogrammes (Menüpunkt

[1] Nach Wold ist es günstiger, nicht ein, sondern mehrere Objekte aus dem gesamten Datensatz herauszunehmen. Dies ist bisher noch nicht implementiert

`Verschiedenes|Standardabweichung, Mittelwert, Median uva.`) ausgewertet werden. Man kann dann sehen, ob das 'echte' Q^2 bzw. r^2 in einem Häufigkeitsbereich liegen, in dem auch viele mit Hilfe der Zufallsdatensaetze erzeugte Werte liegen oder nicht. Läßt die Verteilung des Histogrammes darauf schließen, daß das Auftreten des 'echte' Q^2 bzw. r^2 in einem Zufallsdatensatz unwahrscheinlich ist, dann spricht das für eine aussagekräftige Regression.

Literatur

[1] J. L. Bruning and B. L Kintz. *Computational Handbook of Statistics*. Scott, Foresman and Company, Glennview, Illinois, USA, 1977.

[2] J. Hartung, B. Elpelt, and K.-H. Klösener. *Statistik. Lehr- und Handbuch der angewandten Statistik*. R. Oldenbourg Verlag, München, Wien, 5. edition, 1986.

[3] G. W. Müller and T. Kick. *BASIC Programme für die angewandte Statistik*. Oldenbourg Verlag, München, 1985.

[4] H. R. Neave and P. L. B. Worthington. *Distribution-Free Tests*. Unwin Hyman, London, 1988.

[5] E. Weber. *Grundriß der biologischen Statistik*. VEB Gustav Fischer, Jena, 1986.

[6] S. Wold. Validation of QSARs. *Quant. Struct.-Act. Relat.*, 10:191–193, 1991.

[7] S. Wold and L. Eriksson. Validation tools. In H. van de Waterbeemt, editor, *Chemometric Methods in Molecular Design*, volume 2 of *Methods and Principles in Medicinal Chemistry*, pages 309–318. VCH, Weinheim, 1995.

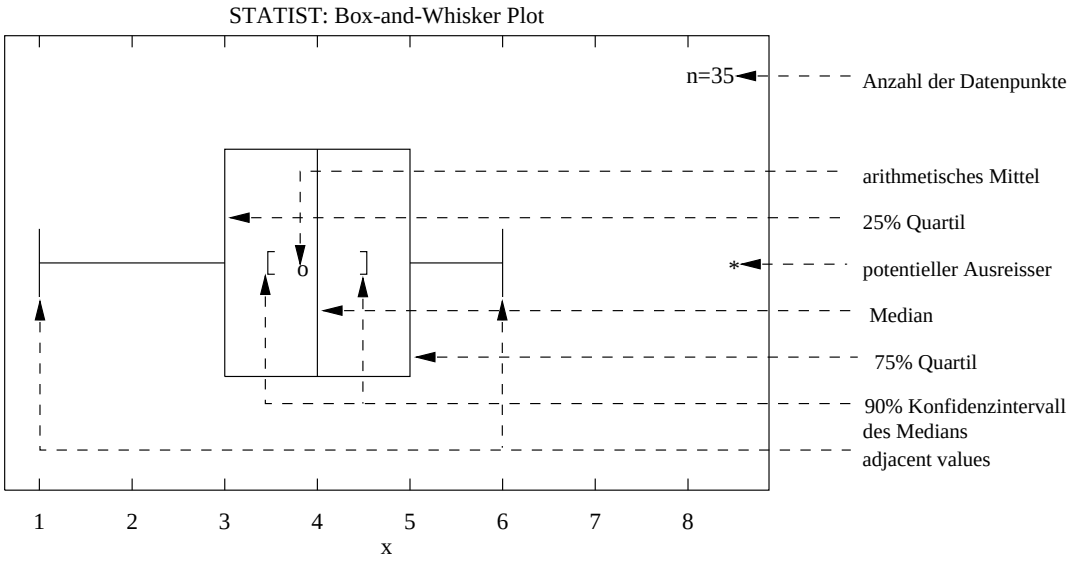

Fig. 1: Beispiel für einen Box-Whisker-Plot. Die *adjacent values* geben Werte an, die am dichtesten am sog. *inner fence* liegen, welcher den 'inneren' Bereich gegen potentielle Ausreißer abgrenzen [2, S. 835].