Begleitskript zur Statistik für AB und NawaRo

Einführung in das Statistikprogramm

SAS

(Statistical Analysis System)

Anhand der Beispiele des Statistikskriptes

B3003

Modul (3403-032/1101-032)

Sebastian Bökle

(begleitet von Jens Möhring & Prof. Dr. Hans-Peter Piepho)

Hohenheim, im Juni 2009

Inhaltsverzeichnis

Vorwort

Dieses Skript soll der Einführung in das Statistikprogramm SAS dienen. Grundlegende Funktionen von SAS werden durch Nachvollziehen und Durchrechnen der Beispiele aus dem Statistikskript erklärt. Die Kapitel in diesem Skript sind analog zur Gliederung des Statistikskriptes. Seitenangaben können mit neuen Auflagen des Statistikskriptes variieren.

Zunächst gehe ich auf den Aufbau und das grundlegende Eingabemuster von SAS ein. Dann kommen fortlaufend immer weitere Funktionen für verschiedene statistische Operationen hinzu. Somit sind die Kapitel inhaltlich aufeinander aufbauend, weswegen es empfehlenswert ist sich vorige Kapitel, um des Verständnisses willen, anzueignen.

Der Aufbau und grundlegende Eingabestrukturen sind einfach zu verstehen. Um in der Hilfe (F1) auch wirklich eine Hilfe zu finden benötigt es zu Beginn ein wenig Ausdauer und Geduld, bis man sich zu Recht findet und weiß wo was steht. Bei Fehlermeldungen sind es zumeist nur kleinste Eingabefehler deren Behebung das Problem schon lösen. Ziel ist es am Ende Datensätzen mit SAS statistisch auswerten zu können.

1. SAS Einführung

Bei SAS haben wir es mit einem Statistikprogramm zu tun, das die Programmanweisungen und Dateneingaben zeilenorientiert einliest und durchläuft. Je nach gewünschter Analyse (graphische Darstellung der Daten, t- Tests, Vertrauensintervalle…) sind einzelne Prozeduren (PROC), die das Programm vorgibt, zu wählen. Wobei wir im Rahmen unserer Analysen mit manchen Prozeduren dieselben Analyseschritte durchführen können. Bei der Auswertung von Daten programmieren wir demzufolge in der Regel immer einen Datenschritt (oder Datastep), in dem die Rohdaten eingelesen bzw. eingegeben werden. Und einen Prozedurschritt (oder Procstep), in dem wir die Prozedur und ihre Einstellungen (Statements, Optionen und Funktionen der einzelnen Prozeduren- mehr dazu im weiteren Verlauf) eingeben, um die gewünschte Auswertung zu vollziehen.

Wenn man SAS öffnet (durch Doppelklick auf ein Desktop- SAS- Symbol oder über START → PROGRAMME → SAS → SAS version (…)) erscheinen auf der rechten Seite drei Fenster und auf der linken 2 Tabs („Results“ und „Explorer“).
Bei den drei Fenstern handelt es sich um das Output- Fenster, das Log-Fenster, und das Fenster des Editors.
Der **Editor** ist das Fenster in dem man schreibt und arbeitet. Es werden hier Daten und Programmanweisungen eingegeben.
Im **Log- Fenster** wird buchstäblich das Logbuch geführt. Hier werden alle Schritte angezeigt, die das Programm durchläuft. Wichtig sind hier die rot geschriebenen Zeilen. Das sind Fehlermeldungen. Wenn keine rote Farbe im Log- Fenster auftaucht ist alles in Ordnung. Es können allerdings auch grün geschriebene Zeilen erscheinen. Dies sind Warnungen und nur gegebenenfalls wichtig.
Im **Output- Fenster** werden Ergebnisse (also Errechnetes aus eingegebenen oder importierten Daten und Programmbefehlen) angezeigt. Wenn wir im Editor nur Daten eingeben, erscheint im Output zunächst nichts. Mit einer bestimmten Anweisung (Erläuterung siehe unter „Importieren von Daten“) können jedoch auch diese Datensätze ins Output- Fenster gebracht werden. Wenn also im Output Fenster nichts eingetragen wurde sind die Datensätze links im Explorer Tab → Libraries/Bibliothek → Work zu finden. Auf die gewünschte Datei einen Doppelklick und es öffnet sich ein Viewtable- Fenster rechts neben dem Editor, in dem dann eine Tabelle mit den eingegebenen Daten ausgefüllt ist. Diese wird auch erstellt, wenn im Output- Fenster etwas angezeigt wird.

Wie bekomme ich meine Daten in mein SAS Programm bzw. in den Editor?

Es geht jetzt um das Erstellen des Datasteps, in dem wir unsere Rohdaten eingeben. Später erstellen wir den Procsteps mit dem wir die Daten dann analysieren.

1. Möglichkeit: Erstellen eines Datensatzes „von Hand“ im Editor:

Wir kennen bspw. den Weißzuckerertrag der Rübenernten einzelner Bundesländer im Jahr 2006 pro ha. [Auszugsweise aus: *Institut für Zuckerrübenforschung Göttingen; Jahresbericht 2006/07; Tabelle S.38*]

Bundesland (Beobachtung)	Ertrag in t/ha
Baden- Württemberg	10,301
Bayern	10,545

Nordrhein- Westfalen	9,296
Brandenburg	5,512
Sachsen- Anhalt	7,366

Folgender Datastep kann im SAS Editor angelegt werden.

```
data Zgehalt_Rueben;
length land $20;
input land$ anteil;
datalines;
Baden_Württemberg 10.301
Bayern 10.545
NRW 9.296
Brandenburg 5.512
Sachsen_Anhalt 7.366
;
run;
proc print data=Zgehalt_rueben;
run;
```

Die Daten mussten hier von Hand eingegeben werden. Hinter **`data`** steht der Name unter dem die Datei angelegt wird. Wichtig ist hier den Unterstrich (_) zwischen den einzelnen Wörtern des Namens nicht zu vergessen und keine Umlaute zu verwenden. Also nicht: „Zgehalt_Rüben“ sondern „Zgehalt_Rueben“. Sonst treten Fehlermeldungen auf und das Programm läuft nicht ab.
Die Anweisung `length` definiert uns die Länge der Worte (Anzahl der Buchstaben) von der Variablen `land`. Sind die Daten nicht als Zahlen (numerisch) sondern bspw. als Buchstaben (alphanumerisch) angegeben, muss in der `input`- Anweisung hinter die entsprechende Variable ein Dollarzeichen $ stehen. `input` gibt die eingegebene(n) Variable(n)/Spaltennamen an. Hier benannt mit „anteil“- Weißzucker von einem Hektar Rüben. `datalines` oder auch `cards` sind Anweisungen dafür, dass die Rohdaten folgen. Wenn man Dezimalzahlen eingibt, muss darauf geachtet werden, dass sie durch einen Punkt und nicht durch ein Komma getrennt werden. **`run`** lässt das Programm laufen und die Rohdaten werden in einer Tabelle im work- Ordner angelegt (s.o.). Wenn in einer Zeile der Datentabelle unter `datalines`/ `cards` mehrere Beobachtungen stehen, müssen in der `input`- Anweisung zwei @ hinter den Variablennamen stehen.
Wichtig: Hinter jedem Befehl/ jeder Anweisung muss ein „;“ stehen um ihn/sie abzuschließen.
Der **`data`** `Zgehalt_Rueben;` Datensatz im Output- Fenster (Erklärung s.u.):

```
                    Das SAS System        19:19 Saturday, January 10, 2009  19

Beob.    land         anteil

  1      Baden_Wü     10.301
  2      Bayern       10.545
  3      NRW           9.296
  4      Brandenb      5.512
  5      Sachsen_      7.366
```

2. Möglichkeit: Importieren von Daten:

SAS bietet die Möglichkeit, Daten, die in anderen Dateien gespeichert sind, zu importieren. Dies geschieht mit dem Import Wizzard. Man findet ihn unter File→ Import Data. Wenn er dann geöffnet ist wird man zunächst nach dem Dateityp, den man importieren möchte, gefragt. Excel-Dateien, aber auch viele andere Formate, sind importierbar. Je nach Version wird diese in der Scrollleiste ausgewählt. Man klickt auf Next/ Weiter und muss dann den genauen Pfad angeben in dem die Datei gespeichert ist. Danach muss das Tabellenblatt der Excel-Datei, in der unsere Daten stehen, angegeben werden. Nun müssen wir einen Platz in SAS bestimmen an dem die Daten gespeichert werden sollen. Zuerst muss eine Library gewählt werden. Für gewöhnlich „Work“, zu beachten ist hier aber, dass die Daten in Work nach jeder SAS- Session wieder gelöscht werden. Entweder man wählt gleich eine andere Library, einen ganz anderen Speicherort oder man speichert die Work Datei später noch einmal gesondert an einem anderen Ort ab. Unter „Member“ wird der gewünschte Name für die Datendatei eingegeben.
Links im Explorer Fenster ist nun unter Active Libraries→ Work ein Table mit den Daten gespeichert. Um sie im Output Fenster stehen zu haben wird die Print-Prozedur mit folgenden Eingaben angewendet.

```
proc print data=/* Name der Work. Datei*/;
run;
```

Dieselben Daten können später auch einfach und schneller über einen Procstep importiert werden. Diese Importschritte müssen also für ein- und dieselben Daten nur einmal durchlaufen werden. Im jetzigen Schritt des Import Wizzards muss der Pfad angegeben werden aus dem SAS die Daten dann importieren könnte bzw. die Work-Datei unter dem wir vorhin die SAS Daten haben abspeichern lassen. Finish klicken.
Der genannte Procstep zum Importieren von Daten sieht folgendermaßen aus:

```
PROC IMPORT OUT= WORK.WEISSZUCKER /* Hier wir der Name der Work.Datei
eingetragen. Hier Bsp. WORK.WEISSZUCKER */
            DATAFILE= "Hier den Pfad in dem die ursprüngliche Excel Tabelle
gespeichert ist eintragen.Bsp: C:/Eigene Dateien/stat/daten…"
            DBMS=EXCEL REPLACE;/*DBMS heißt Datenbank Management System und sagt
nur aus welchem "System" importiert wird. Replace sagt dass eventuell in SAS
vorhandene Dateien mit dem oben angegebenen Namen überschrieben werden.*/
run;
```

Generell ist zum Importieren noch hinzuzufügen, dass die Daten in den eingeführten Tabellen den einzelnen Variablen nach in Spalten nebeneinander stehen sollten (also: In der ersten Spalte stehen die Daten der ersten Variablen untereinander, daneben in der zweiten Spalte die der zweiten Variablen usw.). So können die Variableneingaben hinter `var`, je nachdem welche Prozedur, einfach den Spalten der Tabelle entsprechend nebeneinander eingegeben werden. Auch sonst sollten unnötig viele alphanumerische (in Buchstaben geschrieben) Angaben vermieden werden. Also: Klare, übersichtliche Daten importieren.

In den Beschreibungen zum Importieren sind schon einige Angaben über den oben genannten Procstep erschienen. Wie leicht zu erkennen war, können mit **`proc print`** und der angegebenen gewünschten Datei, Daten im Output Fenster sichtbar gemacht werden. Eine andere wichtige Prozedur zu Beginn ist **`Proc sort`**. Mit dieser Prozedur können Daten der Größe nach geordnet werden. Die Anweisung für unsere Zuckerdaten sieht im Editor so aus:

```
Proc sort data=weisszucker;
by weisszuckerertrag_in_t_ha;
run;
```

Mit `by` wird die Variable benannt über die der Datensatz sortiert werden soll. Wir haben nur eine Variable, trotzdem muss diese angegeben werden. Ohne erscheint im Log-Fenster eine rote Fehlermeldung.
Der Procstep sieht also im Allgemeinen so aus, dass zuerst die Prozedur und dann die zu analysierende Datei angegeben wird. Je nach Prozedur müssen dann verschiedene Variablen- und Syntaxanweisungen folgen. Wie schon erwähnt werden die einzelnen Angaben mit einem „;“ Semikolon abgeschlossen. Wenn alle Eingaben gemacht sind kommt das **`run;`** mit dem die Analyse zum Ablaufen bereit ist.

Farben in SAS:

Die Farbe eines Befehls in einem SAS Programm, wie wir es oben in einfachen Ausführungen schon kennen gelernt haben, folgt einer Ordnung. Befehle die einen Data- (`data Zgehalt_Rueben;`) oder Procstep (`Proc sort data=weisszucker;`) bezeichnen und diesen auch beenden (`run;`) sind dunkelblau.
Statements und Optionen wie `data=` oder `by` werden hellblau geschrieben.
Anweisungen in „“ oder ,’ sind immer lila und Kommentare in /* */ werden grün geschrieben.
Zahlen, Dateinamen, Variablen und Funktionen, wie Rechenzeichen oder verteilungsdefinierende Funktionen sind schwarz.
Erscheint eine Eingabe in rot, macht sie für das Programm keinen Sinn. Bis einzelne Befehle allerdings fertig ausgeschrieben sind erscheinen die Buchstaben auch rot.

1.1 Hilfen

Folgende Quellen waren mir große Hilfen:

- Die Online Hilfe von SAS und die Programm Hilfe selbst. Wenn das Programm geöffnet ist, öffnet sich die Hilfe mit F1, und wenn man online ist öffnet sich die online- Hilfe mit F1.
- Der SAS Kurs von Carina Osterseifen. Gibt es zu downloaden unter: http://web.urz.uni-heidelberg.de/statistik/sas-ah/05.01/SAS-Kurs.html
- Statistik mit SAS von Dufner, Jensen, Schumacher; Teubner Verlag, 2.Auflage; 390 Seiten; ISBN 3-519-12088-7; Kosten ca: 24,50€. Ist in der Universitätsbibliothek als Lehrbuch vorhanden

2. Beschreibende Statistik und metrische Daten

2.1 Histogramm

Beispiel 1, S.9 Maispflanzen: Erstellen eines Histogramms mit SAS

Es erfolgt zunächst das Eingeben der Daten unter den Angaben des Datennamens hinter **`data`**, der Variablen hinter `input` und der einzelnen Datenpunkte unter `cards;` oder `datalines;`.

```
data Laenge_Maispflanzen;
input laenge;
cards;
175
```

```
172
179
167
163
154
163
164
157
177
186
165
175
194
176
162
166
169
170
181
168
166
180
164
179
170
150
192
170
173
170
150
174
164
182
188
157
165
172
168
179
179
164
162
178
162
182
171
182
183
;
run;
```

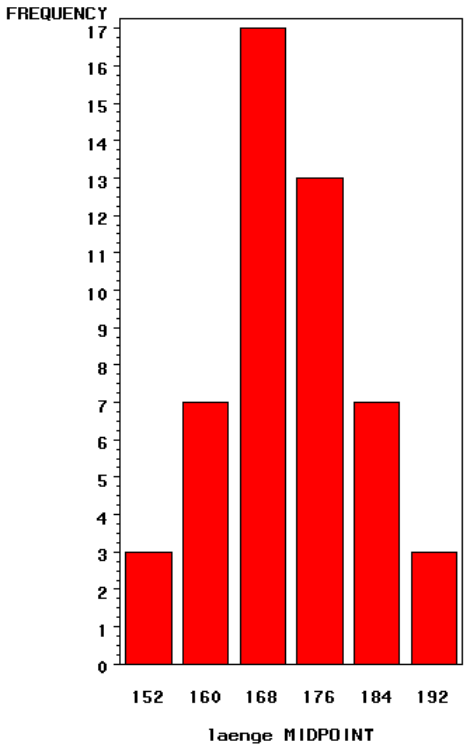
Um ein Histogramm zeichnen zu lassen gibt man die SAS Prozedur **`Proc gchart`** an. Mit der Anweisung `vbar` sagen wir dem System dass es die Balken des Diagramms vertikal zeichnen und außerdem auch die Balken der Häufigkeiten nach, nach der Variablen `laenge`, vertikal auftragen soll. Durch `Type=` sagen wir ob die aufgetragene Häufigkeit auf den Balken die relative oder die absolute Häufigkeit sein soll. Wir haben mit `freq` die absolute Häufigkeit gewählt. D.h. bspw.: Genau „3 Pflanzen für die Länge 152 cm bis

159,9cm). Relative Angaben wären dann der prozentuale Anteil der Stichprobe. Den würde man mit `Type=percent` auf der „y-Achse“ erhalten.

Nachdem alles markiert wurde, kann das Programm laufen gelassen werden (Anklichen das Männchens auf der oberen Menüleiste) und man erhält das unten abgebildete Histogramm. Es öffnet sich ein neues Fenster in der Leiste des Editors, Log-Fensters…. Ebenso ist es in den Libraries des Explorers unter Work angelegt im Ordner „Gseg“. Bevor ein neuer Procstep laufen gelassen wird, sollten allgemein immer jegliche Graphikfenster wieder geschlossen werden, da dadurch Fehlermeldungen auftreten können.

Der Procstep für das Histogramm:

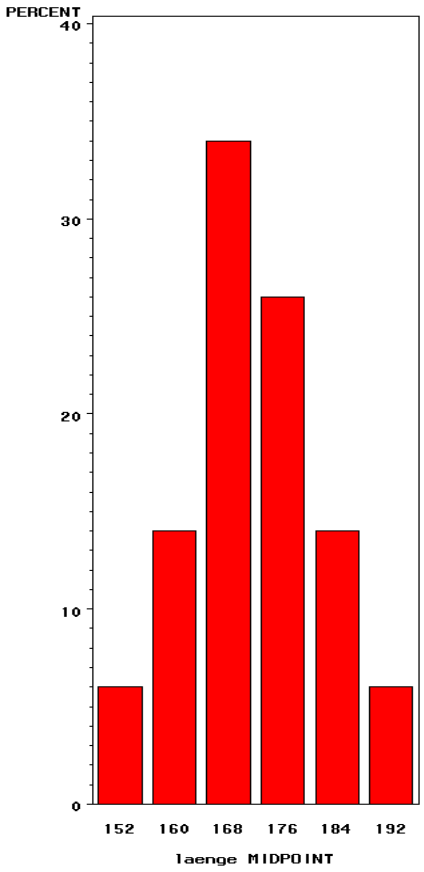
```
proc gchart data=Laenge_Maispflanzen;
vbar laenge / Type=freq;
run;
```



In diesem Histogramm sind nun auf der y-Achse die absoluten Häufigkeiten, d.h. die konkrete Anzahl der Pflanzen aus unserer Stichprobe in ihrem jeweiligen Längenbereich. Um die relative Häufigkeit (die prozentualen Anteile der Längenbereiche an der Stichprobe) zu erhalten ist, wie schon erwähnt, nur ein kleine Änderung von Nöten: `Type=freq` wird zu `Type=percent`.

```
proc gchart data=Laenge_Maispflanzen;
vbar laenge / Type=percent;
run;
```

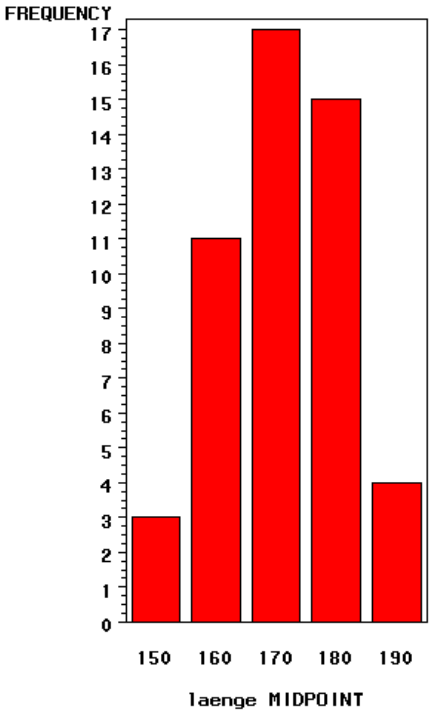
Folgendes Histogramm mit der rel. Häufigkeit in % auf der y-Achse wird erstellt:



Des Weiteren können die Klassenmitten mit der Option Midpoints auf der x- Achse angezeigt werden. Dann stehen unter den Balken die Mitten der Klassen und nicht die oberen Grenzen.

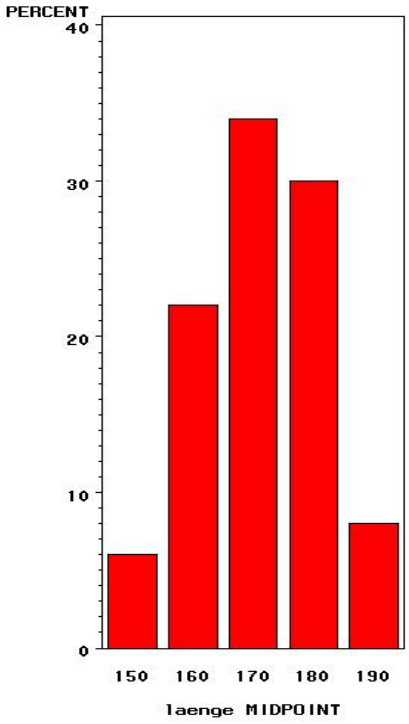
```
proc gchart data=Laenge_Maispflanzen;
vbar laenge / Type=freq midpoints=150 to 190 by 10;
run;
```

Ergebnis absolute Häufigkeit:



Ergebnis relative Häufigkeit:

```
proc gchart data=Laenge_Maispflanzen;
vbar laenge / Type=percent midpoints=150 to 190 by 10;
run;
```



S.11, Beispiel 2: Stem-and-leaf Plot:

Mit der Prozedur **proc univariate** und der Option plot kann ein Stem-and-leaf Plot erstellt werden. Wieder muss die Variable angegeben werden über die die Graphiken erstellt werden sollen. Der Procstep dazu:

```
proc univariate data=Laenge_Maispflanzen plot;
var laenge;
run;
```

Dabei wird außerdem ein Box and Whiskers Plot und ein Normalwahrscheinlichkeits Plot erstellt. Diese erscheinen dann unter dem Stem-and-leaf Plot im Output. Hier nur der Stem-and-leaf und der Box and Whiskers Plot:

```
Stamm Blatt                    #            Box-Plot
  194 0                        1                |
  192 0                        1                |
  190                                           |
  188 0                        1                |
  186 0                        1                |
  184                                           |
  182 0000                     4                |
  180 00                       2                |
  178 00000                    5             +-----+
  176 00                       2             |     |
  174 000                      3             |     |
  172 000                      3             |     |
  170 00000                    5             *--+--*
  168 000                      3             |     |
  166 000                      3             |     |
  164 000000                   6             +-----+
  162 00000                    5                |
  160                                           |
  158                                           |
  156 00                       2                |
  154 0                        1                |
  152                                           |
  150 00                       2                |
      ----+----+----+----+
```

2.2 Statistische Maßzahlen mit SAS:

2.2.1 Quantile (Perzentile)

Auf Seite 12 in Beispiel 1 werden nun nur noch 16 der vorigen 50 Pflanzen (n) der Stichprobe verwendet. Mit einer Syntaxanweisung kann SAS alle Beobachtungen bis auf die ersten 16 Pflanzenlängen löschen.

```
data Laenge_Maispflanzen;set Laenge_Maispflanzen;
if _n_>16 then delete;
run;
```

In diesem Datastep (Datenschritt) haben wir nun folgendes in der sog. Syntaxanweisung, definiert: Mit der **data**- Anweisung benennen wir unseren neuen Datensatz mit den 16 Pflanzen. Mit der set- Anweisung wird SAS mitgeteilt, dass es die Werte für diese neue Datei aus der Datei nehmen soll, die hinter set steht. Durch die nun folgende Syntaxanweisung sagen wir SAS, dass es Werte über der Anzahl Pflanzen n=16 löschen soll: `if _n_>16 then delete;` (*wenn n>16 dann löschen*. Die Werte 17- 50 werden also im neuen Datensatz „gelöscht“). Weiter geht es mit dem Procstep um die Quantile 25%, 50% (Median) und 75% zu berechnen. Um dies zu erhalten wird die Prozedur **univariate** aufgerufen. Die Quantile sollen über der Variablen laenge berechnet werden also „`var laenge;`“

```
proc univariate data=Laenge_Maispflanzen;
var laenge;
run;
```

Im Output erscheint nun:

```
The SAS System          10:22 Tuesday, July 29, 2008   5

                          The UNIVARIATE Procedure
                             Variable:  laenge

                                Moments

N                          16    Sum Weights                16
Mean                   170.5625    Sum Observations         2729
Std Deviation      10.6581346    Variance            113.595833
Skewness           0.54929991    Kurtosis            0.08195578
Uncorrected SS         467169    Corrected SS         1703.9375
Coeff Variation    6.24881472    Std Error Mean      2.66453365

                      Basic Statistical Measures

            Location                      Variability

      Mean     170.5625     Std Deviation           10.65813
      Median   169.5000     Variance               113.59583
      Mode     163.0000     Range                   40.00000
                            Interquartile Range     13.50000

NOTE: The mode displayed is the smallest of 2 modes with a count of 2.

                      Tests for Location: Mu0=0

      Test          -Statistic-    -----p Value------
```

```
Student's t    t  64.01214    Pr > |t|    <.0001
Sign           M         8    Pr >= |M|   <.0001
Signed Rank    S        68    Pr >= |S|   <.0001


            Quantiles (Definition 5)

             Quantile      Estimate

             100% Max         194.0
             99%              194.0
             95%              194.0
             90%              186.0
             75% Q3           176.5
             50% Median       169.5
             25% Q1           163.0
             10%              157.0
             5%               154.0
             1%               154.0
             0% Min           154.0
```

Der auf S.14, Beispiel 2 berechnete Median ist hier mit enthalten und wird deswegen nicht gesondert berechnet. (Sortierschritt vgl. nächstes Beispiel)

S.14, Beispiel 3: Lebendgewichte einer Stichprobe von Milchkühen:

Wieder geben wir die Daten in einem Datastep ein und benennen die Variable:

```
data Lebendgewichte;
input gewicht;
cards;
663
644
656
671
665
659
656
647
642
647
657
632
649
;
run;
```

Um wie im Skript vorzugehen können die Daten mit der Prozedur **proc sort** folgendermaßen sortiert und dann mit **proc print** im Output angezeigt werden.

```
proc sort data=Lebendgewichte;
```

```
by gewicht;
run;
proc print data=Lebendgewichte;
run;
```

Desweiteren wollen wir nun die Quantile Q_{10}, Q_{20} und Q_{50} berechnen. Die Quantile Q_{10} und Q_{50} haben wir im vorigen Beispiel schon erfolgreich bestimmen können da diese von UNIVARIATE automatisch angezeigt werden, stehen sie auch jetzt wieder im OUTPUT (s.u.). Für das Quantil Q_{20} muss ein neuer Befehl eingegeben werden:

```
proc univariate data= Lebendgewichte;
var gewicht;
output out= Lebendgewichte1 pctlpts=20 pctlpre=gewicht;
run;
```

Mit „`output out=`“ sagen wir SAS, dass es das von uns ausgewählte Quantil in die Datei `Lebendgewichte1` schreiben soll, weil es im vordefinierten Output nicht einbezogen ist. `pctlpts=` gibt die Prozentzahl an, die das gewünschte Perzentil haben soll und `pctlpre=` veranlasst SAS in der gewünschten Variablen zu suchen.

Dieses Quantil erscheint nun wie gesagt nicht im Output- Fenster sondern wird als gesonderte Datei in den Ordnern der Librairie des Explorers (auf der linken Seite in den Tabs *Explorer und Results*) angelegt und zwar unter dem Namen den wir hinter `output out` eingegeben haben, es ist ein neuer „gesonderter“ data file, deswegen Lebendgewichte1. In der Librairy im Ordner „Work“ wurde nun die folgende Tabelle angelegt:

Lebendgewichte von Milchkuehen

gewicht20
644

Output für:

```
Lebendgewichte von Milchkuehen
                                                    15:09 Tuesday, April 15, 2008

                          The UNIVARIATE Procedure
                             Variable:  gewicht

                                 Moments

N                          13    Sum Weights                13
Mean                652.923077    Sum Observations         8488
Std Deviation       10.6806799    Variance            114.076923
Skewness            -0.2107311    Kurtosis            -0.1649146
```

```
Uncorrected SS        5543380    Corrected SS        1368.92308
Coeff Variation    1.63582515    Std Error Mean      2.96228762


                    Basic Statistical Measures

          Location                    Variability

      Mean     652.9231     Std Deviation          10.68068
      Median   656.0000     Variance              114.07692
      Mode     647.0000     Range                  39.00000
                            Interquartile Range    12.00000

NOTE: The mode displayed is the smallest of 2 modes with a count of 2.


                    Tests for Location: Mu0=0

   Test           -Statistic-    -----p Value------

   Student's t    t  220.4118    Pr > |t|    <.0001
   Sign           M       6.5    Pr >= |M|   0.0002
   Signed Rank    S      45.5    Pr >= |S|   0.0002


                 Quantiles (Definition 5)

                  Quantile      Estimate

                  100% Max           671
                  99%                671
                  95%                671
                  90%                665
                  75% Q3             659
                  50% Median         656
                  25% Q1             647
                  10%                642
                  5%                 632
                  1%                 632
                  0% Min             632
```

S.15, Beispiel 4: Die Quantile Q_0 und Q_{100} sind im Output von Bsp3 enthalten.

2.2.2 Lagemaße

Median: s.o.

Das arithmetisches Mittel: S.16, Beispiel 1: 16 Maispflanzen:

Das arithmetische Mittel kann mit der Prozedur **`proc means`** berechnet werden. Folgende Anweisungen sind einzugeben:

```
proc means data=Laenge_Maispflanzen;
var laenge;
run;
```

Hier ist jetzt nur der Procstep angegeben. Der Datastep ist derselbe wie der in 2.2.1 für die Quantile. Dieser wiederum war ein Auszug aus dem Datastep mit den 50 Maispflanzen. Mit **`proc means`** erhält man dann die Ergebnisse im Output:

```
The SAS System           10:22 Tuesday, July 29, 2008  17

                              The MEANS Procedure

                          Analysis Variable : laenge

 N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------
16      170.5625000      10.6581346      154.0000000      194.0000000
-------------------------------------------------------------------
```

Das geometrische Mittel:

Zusatzbeispiel:
Zusätzlich zu dem Beispiel der Keimzahlen in Milch berechnen wir das geometrische Mittel unserer 16 Maispflanzen. Da für das geometrische Mittel die Daten logarithmiert werden müssen, müssen auch wir einen neuen Datensatz erstellen indem wir wie vorhin unseren neuen Datensatz mit `set` aus dem vorigen erstellen. Allerdings müssen wir hier noch das Logarithmieren mit einbeziehen, was wir mit der Syntaxanweisung `log_laenge=log(laenge);` bewerkstelligen. Dann sieht unser Datastep so aus:

```
data Laenge_Maispflanzen; set Laenge_Maispflanzen;
log_laenge=log(laenge);
run;
```

Im Output mit Variablenangabe anzeigen lassen durch:

```
proc print data=Laenge_Maispflanzen;
var log_laenge laenge;
run;
```

In diesem Print haben wir zwei Variable angegeben. In der Eingabezeile hinter `var` werden diese nur durch eine Leerzeile (Blank) getrennt und SAS erkennt, dass es sich um mehrere Variable handelt. Somit werden im Output nun die beiden Spalten der Variablen angezeigt:

```
                          log_
```

Obs	laenge	laenge
1	5.16479	175
2	5.14749	172
3	5.18739	179
4	5.11799	167
5	5.09375	163
6	5.03695	154
7	5.09375	163
8	5.09987	164
9	5.05625	157
10	5.17615	177
11	5.22575	186
12	5.10595	165
13	5.16479	175
14	5.26786	194
15	5.17048	176
16	5.08760	162

Zur Übung: Sortieren der Daten nach der Länge und anschließendes erneutes Anzeigen im Output: Procstep:

```
proc sort data=Laenge_Maispflanzen;
by laenge;
run;
proc print data=Laenge_Maispflanzen;
var log_laenge laenge;
run;
```

Sortierter Output von laenge und log_laenge:

The SAS System 10:22 Tuesday, July 29, 2008 19

Obs	log_ laenge	laenge
1	5.03695	154
2	5.05625	157
3	5.08760	162
4	5.09375	163
5	5.09375	163
6	5.09987	164
7	5.10595	165
8	5.11799	167
9	5.14749	172
10	5.16479	175
11	5.16479	175
12	5.17048	176
13	5.17615	177
14	5.18739	179
15	5.22575	186
16	5.26786	194

Das geometrische Mittel nun lässt sich mit der Prozedur „**proc univariate**“ berechnen.

```
proc univariate dat=Laenge_Maispflanzen;
output out=a mean=m;
var log_laenge;
run;
```

Im Output ist das noch logarithmierte(!) geometrische Mittel der **Mean 5.13729948** nicht der Median der Percentilen. Um das geometrische Mittel zu erhalten, kann mit SAS auch wie mit einem Taschenrechner gerechnet werden, indem man einfach einen neuen Datensatz „b“ erzeugt, diesen aus den logarithmierten Daten „a“ zieht und die gewünschte Rechenoperation eingibt. Das geometrische Mittel bezeichnen wir mit klein „g“. Um das Ganze im Output anschauen zu können: **proc print**.

```
data b;
set a;
g=exp(m);
run;

proc print data=b;
run;
```

Das ist der Output nach **proc univariate**:

```
The SAS System          10:22 Tuesday, July 29, 2008  20

                          The UNIVARIATE Procedure
                           Variable:  log_laenge

                                Moments

N                          16    Sum Weights                16
Mean                5.13729948    Sum Observations     82.1967916
Std Deviation       0.06178582    Variance             0.00381749
Skewness            0.40681563    Kurtosis             -0.1290116
Uncorrected SS      422.326797    Corrected SS         0.05726231
Coeff Variation     1.20269064    Std Error Mean       0.01544646

                      Basic Statistical Measures

          Location                      Variability

     Mean     5.137299     Std Deviation            0.06179
     Median   5.132744     Variance                 0.00382
     Mode     5.093750     Range                    0.23091
                           Interquartile Range      0.07957

NOTE: The mode displayed is the smallest of 2 modes with a count of 2.

                      Tests for Location: Mu0=0

     Test           -Statistic-     -----p Value------

     Student's t    t  332.5876    Pr > |t|    <.0001
     Sign           M         8    Pr >= |M|   <.0001
     Signed Rank    S        68    Pr >= |S|   <.0001

                      Quantiles (Definition 5)

                       Quantile      Estimate
```

```
100% Max        5.26786
99%             5.26786
95%             5.26786
90%             5.22575
75% Q3          5.17332
50% Median      5.13274
25% Q1          5.09375
10%             5.05625
5%              5.03695
1%              5.03695
0% Min          5.03695
```

Und dieser nach unserer kleinen Rechenoperation:

```
Das SAS System            21:14 Monday, August 4, 2008   9

                    Beob.       m          g

                      1      5.14105    170.895
```

Mit dem geometrischen Mittel g= 170.895.

S. 19, Beispiel 1: Keimzahlen

Um das geometrische Mittel für die Keimzahlen in Milch zu berechnen, gehen wir wie bei den Maispflanzen vor.

Datastep:

```
data Keimzahlen;
input anzahl;
cards;
5150
57400
14200
35800
12200
26900
61000
6750
51700
1390
285
3150
4000
9600
200
265
13100
5800
56800
440
4750
1170
4350
8200
47
60900
```

835
950
8800
270
1410
5150
3300
4900
995
3950
600
2550
3550
63000
2150
86200
4150
710
580
8250
975
1910
5400
1350
30500
16800
120
380
27500
295
965
32700
3030
56500
890
45700
980
595
765
1340
15700
12100
4550
20500
20500
52000
18700
4350
32900
910
2700
115
5900
33000
19100
1760
415
2450
70000
170
1820
21800
79800
230

```
9150
6800
7050
1940
2750
;
run;
```

Die Daten werden wieder zuerst der Anschaulichkeit halber im Output erzeugt. Dann wird mit `log_anzahl=log(anzahl);` eine Spalte mit den logarithmierten Werten im Output erstellt. Zuvor muss noch festgelegt werden, mit welchen Daten das geschehen soll. Das sagen wir SAS mit der Anweisung `set`. Sie holt die Daten in den Datastep, die wir hinter ihr eingeben, und zwar die Daten von oben, die wir Keimzahlen genannt hatten.

```
proc print data=Keimzahlen;
run;
data Keimzahlen;
set Keimzahlen;
log_anzahl=log(anzahl);
run;
proc print data=Keimzahlen;
var log_anzahl anzahl;
run;
```

Der eigentliche Procstep:

```
proc univariate data=Keimzahlen;
output out=a mean=m; /*output out ist eine Funktion und erzeugt die Datei a und
mit mean wird der Mittelwert m in die neue Datei geschrieben.*/
var log_anzahl;
run;
data b;set a;/*neue Datei a (durch set) wird aus alter Datei b erzeugt.*/
g=exp(m);
run;
proc print data=b;
run;
```

Das kommt raus:

```
                Das SAS System          20:26 Tuesday, August 5, 2008   7

                                Beob.       m          g

                                  1      8.29326    3996.83
```

Das harmonische Mittel: Beispiel 2, S. 22: Bodenproben: Dichte in kg/Liter

Unser Datastep :

```
Data Bodendichte;
input dichte;
cards;
1.3
1.4
1.3
1.2
1.1
1.0
```

```
1.5
1.4
1.4
1.3
;
run;
proc print data=bodendichte;
var dichte;
run;
```

Die Berechnung des Kehrwertes:

```
data bodendichte;set bodendichte;
kehrwert_dichte=1/dichte;
run;
proc print data=bodendichte;
var dichte kehrwert_dichte;
run;
```

Unser Procstep:

```
proc univariate data=bodendichte;
output out=c mean=m;
var kehrwert_dichte;
run;
Data d;set c;
m=1/m;
run;
proc print data=d;
run;
```

Mit „`m=1/m;`“ wird erstens die Variable *m* erzeugt und zweitens durch das 1/*m* der Kehrwert des Mittelwertes m dort gespeichert.
Und das durch „**`proc print`**“ erscheinende Ergebnis im Output:

```
Das SAS System          20:02 Wednesday, August 6, 2008   16

                        Beob.        m

                          1       1.27232
```

2.2.3 Streuungsmaße

Die in diesem Kapitel behandelten Streuungsmaße (Variationsbreite, Interquartilabstand, Varianz, Standardabweichung und Variationskoeffizient) sind von SAS in der Funktion „univariate“ alle enthalten und werden auf einmal zusammen ausgerechnet.
Was neu hinzukommt ist, dass wir mit den 3 Düngemethoden mehrere Variablen in unserem On-Farm Versuch haben. Diese müssen in den schon bekannten Befehlen wie folgt berücksichtigt werden. Sie müssen alle drei, durch ein Leerzeichen getrennt, in der „input“-Anweisung sowie in der „var“-Anweisung unter „proc univariate“ eingegeben werden.

```
data OnFarm_Duengemittel;
input Kein NPK DAP;
cards;
0.30 0.80 1.64
0.34 1.12 1.38
0.39 1.12 1.70
0.40 1.60 2.80
0.40 2.80 2.40
0.42 1.14 1.56
0.48 3.20 1.92
0.54 1.34 1.46
0.56 1.20 1.66
0.58 1.22 1.60
0.62 1.40 2.30
0.68 2.24 2.76
0.74 1.54 1.66
0.74 1.52 2.42
0.78 1.46 1.80
0.82 1.60 2.50
0.96 1.60 2.06
1.02 1.74 2.16
1.06 1.40 1.74
1.10 1.44 1.74
1.44 4.16 3.84
1.60 2.00 2.40
1.68 4.80 2.56
2.40 4.48 3.84
2.40 9.60 3.84
2.56 5.28 3.24
3.60 4.80 5.60
4.50 5.50 6.75
;
run;
proc univariate data=OnFarm_Duengemittel;
var Kein NPK DAP;
run;
```

Man erhält dann im Output:

Variationsbreite (auch: Spannweite) → `Range`

Interquartilabstand → `Interquartile Range`

Varianz → `Variance`

Standardabweichung → `Std Deviation`

Variationskoeffizient → `Coeff Variation`

Stichprobenanzahl → `N`

Mittelwert → `Mean`

The UNIVARIATE Procedure
Variable: DAP

Moments

N	28	Sum Weights	28
Mean	2.5475	Sum Observations	71.33
Std Deviation	1.2602546	Variance	1.58824167
Skewness	1.99545468	Kurtosis	4.26912698
Uncorrected SS	224.5957	Corrected SS	42.882525
Coeff Variation	49.4702494	Std Error Mean	0.23816573

Basic Statistical Measures

Location		Variability	
Mean	2.547500	Std Deviation	1.26025
Median	2.230000	Variance	1.58824
Mode	3.840000	Range	5.37000
		Interquartile Range	1.10000

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	10.69633	Pr > |t|	<.0001
Sign	M	14	Pr >= |M|	<.0001
Signed Rank	S	203	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	6.75
99%	6.75
95%	5.60
90%	3.84
75% Q3	2.78
50% Median	2.23
25% Q1	1.68
10%	1.56
5%	1.46
1%	1.38
0% Min	1.38

The UNIVARIATE Procedure
Variable: DAP

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
1.38	2	3.84	21
1.46	8	3.84	24
1.56	6	3.84	25
1.60	10	5.60	27

```
1.64        1         6.75       28
```

Mit dem oben eingegebenen Datensatz und den angefügten Befehlen für SAS können Mittelwert und Standardabweichung berechnet werden.

```
proc means data=Laenge_Maispflanzen;
var laenge; run;
```

```
                          The SAS System         15:09 Tuesday, April 15, 2008   1

                          The MEANS Procedure

                       Analysis Variable : laenge

 N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------
50      171.1800000       9.9768917     150.0000000     194.0000000
-------------------------------------------------------------------
```

2.3 Box- Plot

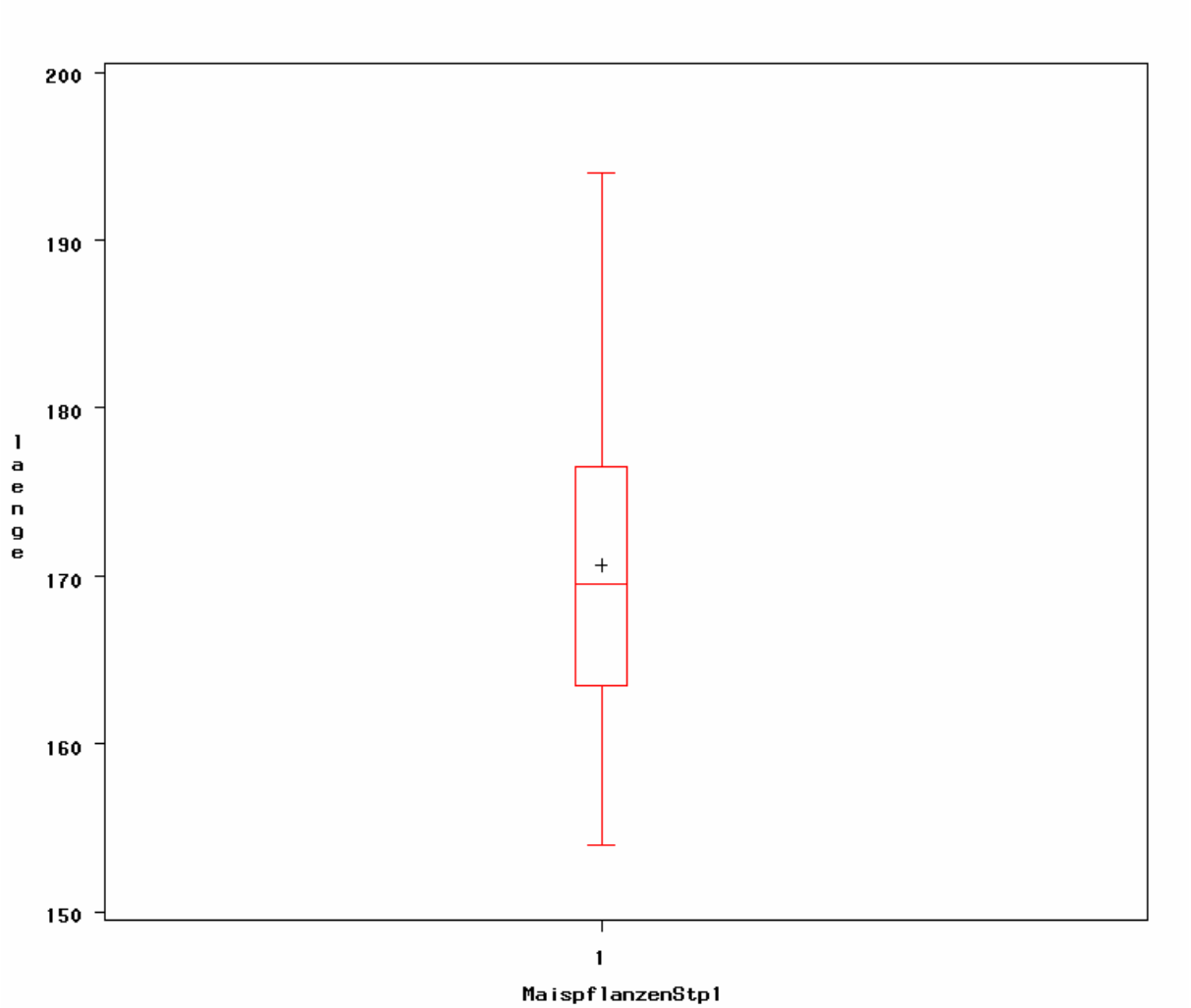
Box-Plot: S.27, Beispiel 1: Stichprobe Maispflanzen

Über die 16 gemessenen Längen der Maispflanzen soll ein Box- Plot erstellt werden. In SAS gibt es dafür die Prozedur Box- Plot. Die Daten werden wie gewohnt benannt, hier mit „**data** BoxPlotMp1;“ und eingegeben. Mit den folgenden Befehlen und Eingaben:

```
data BoxPlotMp1;
MaispflanzenStp1=1;/*Diese Beschreibung gibt die 2. Variable an, damit das
Programm 2 Variablen hat, die es miteinander plotet*/
input laenge;
cards;
175
172
179
167
163
154
163
164
157
177
186
165
175
194
176
164
;
run;
proc boxplot data=BoxPlotMp1;/* Das ist die Prozedureingabe für Box- Plot-
Berechnungen*/
```

```
plot laenge*MaispflanzenStp1;
;run;
```

Erhalten wir mit SAS folgenden Box- Plot:



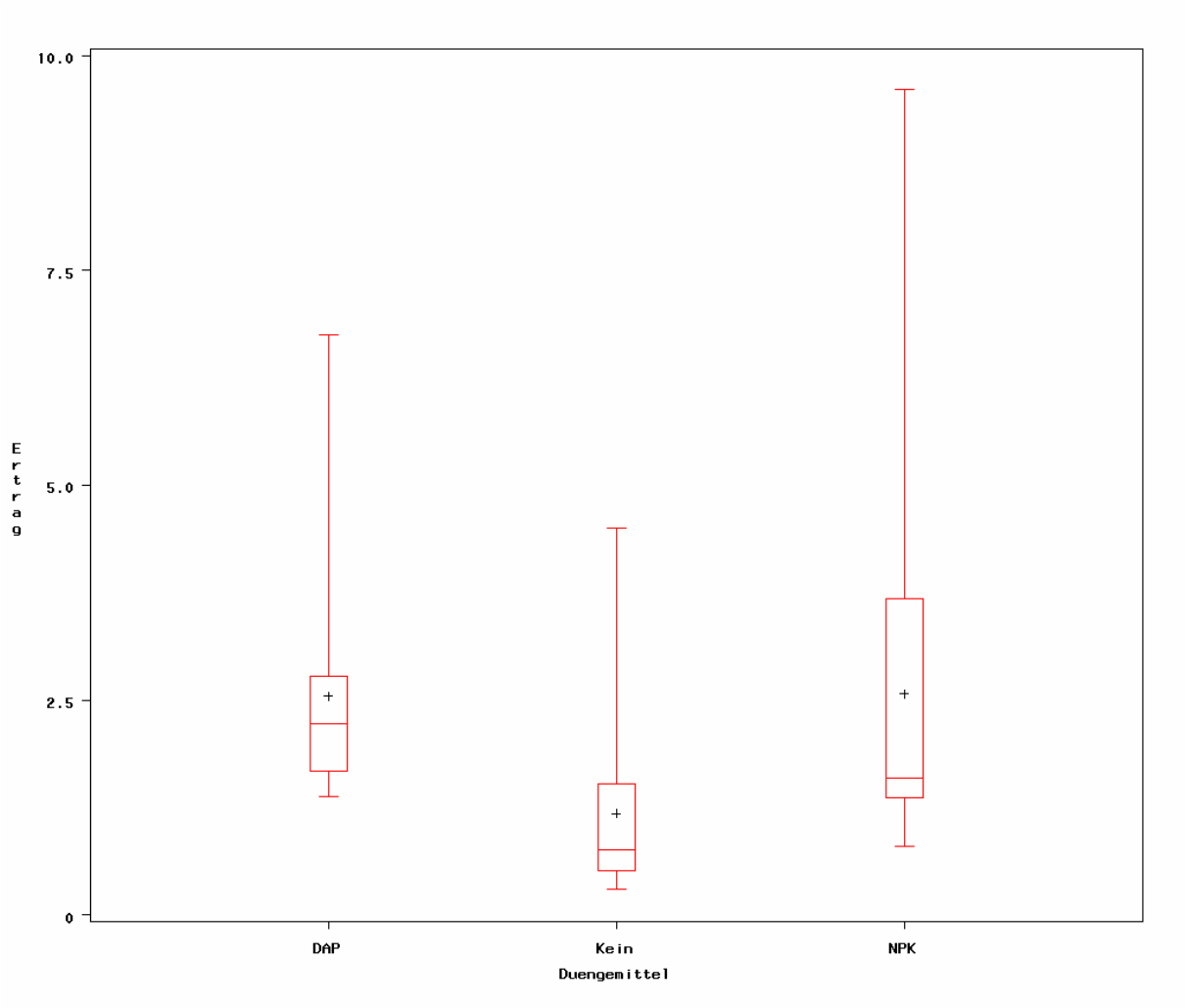
Box- Plot Beispiel über den OnFarm Versuch (S.29):

Mit diesen Eingaben:

```
data OnFarm_Duengemittel;
input Duengemittel $ Ertrag @@;/* das $- Zeichen sagt, dass die Dünemittel Daten
(Kein, NPK und DAP) nicht numerisch, also keine Zahlen sind. Das @@ sagt, dass
mehrere Beobachtungen in einer Zeile sind.*/
cards;
Kein 0.30 NPK 0.80 DAP 1.64
Kein 0.34 NPK 1.12 DAP 1.38
Kein 0.39 NPK 1.12 DAP 1.70
Kein 0.40 NPK 1.60 DAP 2.80
Kein 0.40 NPK 2.80 DAP 2.40
Kein 0.42 NPK 1.14 DAP 1.56
Kein 0.48 NPK 3.20 DAP 1.92
Kein 0.54 NPK 1.34 DAP 1.46
Kein 0.56 NPK 1.20 DAP 1.66
```

```
Kein 0.58 NPK 1.22 DAP 1.60
Kein 0.62 NPK 1.40 DAP 2.30
Kein 0.68 NPK 2.24 DAP 2.76
Kein 0.74 NPK 1.54 DAP 1.66
Kein 0.74 NPK 1.52 DAP 2.42
Kein 0.78 NPK 1.46 DAP 1.80
Kein 0.82 NPK 1.60 DAP 2.50
Kein 0.96 NPK 1.60 DAP 2.06
Kein 1.02 NPK 1.74 DAP 2.16
Kein 1.06 NPK 1.40 DAP 1.74
Kein 1.10 NPK 1.44 DAP 1.74
Kein 1.44 NPK 4.16 DAP 3.84
Kein 1.60 NPK 2.00 DAP 2.40
Kein 1.68 NPK 4.80 DAP 2.56
Kein 2.40 NPK 4.48 DAP 3.84
Kein 2.40 NPK 9.60 DAP 3.84
Kein 2.56 NPK 5.28 DAP 3.24
Kein 3.60 NPK 4.80 DAP 5.60
Kein 4.50 NPK 5.50 DAP 6.75
;run;
proc sort data=OnFarm_Duengemittel; /*es muss zuerst ein Mittelwert berechnet
werden, weil dabei die Daten sortiert werden. Dies ist nötig um den folgenden so
gewünschten Box- Plot zu erhalten.*/
by Duengemittel;run;
proc boxplot data=OnFarm_Duengemittel;
plot Ertrag*Duengemittel;
;run;
```

erhalten wir:



3. Einführung in die schließende Statistik für normalverteilte Daten (univariat)

3.2 Normalverteilung

S.34 Frage 1: Wie groß ist die Wahrscheinlichkeit, dass eine zufällig aus dem Feld gezogene Pflanze x= 185 cm ist oder länger?

Es wird also nach einer Überschreitungswahrscheinlichkeit gesucht.

Wir haben und brauchen dazu einen Mittelwert μ =170 cm, eine Standardabweichung σ_x= 10 und die gegebene Pflanzenlänge als unterste Grenze, Unterschreitungsgrenze. Dann ist es im Prinzip nur die Rechenformel von S.37 die wir SAS geben um unsere Wahrscheinlichkeit zu berechnen. Zusätzlich muss das System noch wissen in welcher Verteilung die Wahrscheinlichkeit sein soll. Das wird mit der Option `p=probnorm (z);`

bewerkstelligt. Sie sagt, dass die Wahrscheinlichkeit p eine Normalverteilung in Form einer z- transformierten Standardnormalverteilung sein soll. All das geschieht mit folgendem Datastep.

```
data MpF1;
z= (185-170)/10;
p=probnorm (z);
run;
```

Ein Procstep ist hier durch die Rechenformel im Datastep nicht mehr nötig.

```
proc print data=MpF1;
run;
```

```
                    Das SAS System         20:27 Thursday, August 14, 2008   1

                                  Beob.      z         p

                                    1       1.5     0.93319
```

Das Ergebnis der Operation steht somit im Output und auch wieder im Work Ordner. Wichtig ist für uns das z = 1,5 mit dem wir in Tabelle 1, S. 209 im Skript, für z(1,5)=0,0668 ablesen. D.h., dass 6,68% der beobachteten Maispflanzen größer oder gleich 185 cm lang sind. p gibt die Unterschreitungswahrscheinlichkeit an, hier 0,93319. Diese können wir durch eine kleine Erweiterung des Datastep direkt berechnen:

```
data MpF1;
z= (185-170)/10;
p=probnorm (z);
q=1-p;
run;

proc print data=MpF1;
run;
```

S. 37 Frage 2:

Jetzt ist nach einem Bereich gefragt. Im Datastep ist leicht nachzuvollziehen, dass wir den z-Wert der oberen und der unteren Grenze berechnen lassen und dann in p3 die beiden Wahrscheinlichkeiten voneinander abziehen. Mit `p=abs(p1-p2);` wird dem System nur gesagt, dass es bei der Berechnung der Differenz den Betrag der Differenz ausgeben soll.

```
data MpF2;
z1= (185-170)/10;/* z1=1,5 Folge: 0,0668 also 6,68% liegen über 185cm*/
z2= (175-170)/10;/* z2=0,5 Folge: 0,3085 also 30,85% liegen über 175cm*/
p1= probnorm (z1);
p2= probnorm (z2);
p3=p1-p2; /* Zwischen 175cm und 185cm liegen nun Mp21 - Mp2= 30,85%-6,68%
Maispflanzen.*/
run;
```

Output:

```
Das SAS System          20:27 Thursday, August 14, 2008   2

Beob.      z1      z2        p1          p2          p3

  1       1.5     0.5     0.93319     0.69146     0.24173
```

Im Viewtable (Libraries→ Work→ Mpf2) unter p3 steht nun der Anteil Pflanzen einer gezogenen Stichprobe die zwischen 175 cm und 185 cm hoch sind. Mit den Definitionen von z1 bis p3 hier im Datastep werden also praktisch die Spalten für unseren Viewtable in Work und was darin steht festgelegt.

S.38 Frage 3:

Gefragt ist nach der Wahrscheinlichkeit, dass eine Pflanze kleiner als 145 cm ist. Dafür gehen wir gleich wie in Frage 1 vor.

```
data MpF3;
z=(145-170)/10;
p=probnorm (z);
run;
```

Output:

```
Das SAS System          20:27 Thursday, August 14, 2008   3

                Beob.       z              p

                  1       -2.5     .006209665
```

S.38 Frage 4:

Mit SAS erzeugen wir die Antwort von Frage 4 wie im Skript in dem wir die Ergebnisse aus den vorigen Fragen zu Rate ziehen. Zuerst müssen wir über die ersten zwei Datasteps SAS die Ergebnisse der vorigen Aufgaben bekannt geben. Dann erzeugen wir einen neuen Datastep für die Beantwortung der Frage 4 und lassen SAS die Werte einfach nur voneinander abziehen. "merge" gibt SAS an, dass es die Werte in der angelegten work Tabelle nebeneinander schreiben soll.

```
data MpF3;
p=0.0062096653;
p2=0.6914624613;
p3=1-p2-p;
run;
```

Output:

```
Das SAS System          20:27 Thursday, August 14, 2008   4
```

```
Beob.            p        p2         p3

  1       .006209665     0.69146     0.30233
```

S.38 Frage 5:

Gleiches Vorgehen wie in den Fragen 1 und 3.

```
data MpF5;
z=(170-170)/10;
p=probnorm (z);
run;

proc print data=MpF5;
run;
```

Output:

```
                Das SAS System         20:27 Thursday, August 14, 2008   5

                                Beob.    z      p

                                  1      0    0.5
```

3.5 Vertrauensintervall für einen Mittelwert

S.42, Beispiel 1:

Der Datensatz für die 50 Maispflanzen:

```
data Laenge_Maispflanzen;
input laenge;
cards;
175
172
179
167
163
154
163
164
157
177
186
165
175
194
176
162
166
169
170
181
168
166
180
164
179
170
```

```
150
192
170
173
170
150
174
164
182
188
157
165
172
168
179
179
164
162
178
162
182
171
182
183
;
run;
proc print data=Laenge_Maispflanzen;
run;
```

Wieder nur angewandt, um die Daten im Output anschauen zu können.

Der Procstep mit der Prozedur Means:

```
proc means data=Laenge_Maispflanzen clm ;
var laenge;
run;
```

Output:

```
                    Das SAS System         20:27 Thursday, August 14, 2008   7

                                  Die Prozedur MEANS

                               Analysis Variable : laenge

                          Untere 95%            Obere 95%
                       KG für Mittelwert     KG für Mittelwert
                       ------------------------------------------
                             168.3445988           174.0154012
                       ------------------------------------------
```

Mit der Option `clm` wird ein Vertrauensintervall in means für den Mittelwert berechnet. Unsere Vertrauensintervalle mit SAS weichen geringfügig vom Ergebnis im Skript ab. Das liegt daran, dass die Means Prozedur schon mit einer t-Verteilung arbeitet und nicht wie im Skript mit einer Normalverteilung. Dennoch liegen die Werte sehr nah beieinander, weil wir einen kritischen Wert der t-Verteilung von 2,010, bei n=50, FG(n-1)=49 und alpha=0.05, ablesen können (siehe Tab.II zweiseitig) und so kein großer Unterschied zu 1,96 besteht.

Noch deutlicher wird dieser Sachverhalt im Folgenden.
Zur weiteren Veranschaulichung des Unterschiedes einer Normal- und einer t- Verteilung: Der folgende Schritt gibt die Annahme von S.43 Mitte wieder und zwar, dass bei unseren Maisdaten die Standardabweichung ungefähr s=10 ist und unser tabellierter t-Wert bei n=10, alpha=0.05 und den FG=(n-1)=(10-1)=9 2,262 ergibt. Dadurch wird der Vertrauensintervall deutlich breiter aufgrund der kleineren Stichprobe des höheren t-Tabellenwertes. Die Berechnung wird mit der Formel aus dem Skript durchgeführt

```
proc means data=Laenge_Maispflanzen clm;
var laenge;
output out=b mean=mean std=std;
run;
data b;
set b;
lower=171.18- 2.262*10/sqrt(10);
upper=171.18+ 2.262*10/sqrt(10);
run;
proc print data=b;
run;
```

Das SAS System 20:27 Thursday, August 14, 2008 9

Beob.	_TYPE_	_FREQ_	mean	std	lower	upper
1	0	50	171.18	9.97689	164.027	178.333

Die Optionen lower und upper können außer mit den abgetippten Zahlen auch mit Anweisungen versehen werden. Das ist möglich, weil wir im Procstep davor definiert haben, was der Mittelwert ist (`mean=mean`) und was die Standardabweichung ist (`std=std`). `sqrt(_freq_)` ist die Quadratwurzel (square root) des Stichprobenumfangs `_freq_`. Der Datastep sieht dann so aus:

```
data b;
set b;
lower=mean-2.262*std/sqrt(_freq_);
upper=mean+2.262*std/sqrt(_freq_);
run;
proc print data=b;
run;
```

Das Ergebnis sollte dann auch dasselbe sein. ☺

3.6 Vertrauensintervall für einen Mittelwert bei kleinen Stichproben

S. 44 Beispiel Weizensorten

```
data Weizensorte;
```

```
input Ertrag;
cards;
44
40
18
20
45
26
55
55
20
46
15
8
41
;
run;
proc print data=Weizensorte;
run;
proc means data=Weizensorte clm;
var Ertrag;
output out=m mean=mean std=std;
run;
```

Output:

```
                    Das SAS System          19:39 Monday, August 18, 2008  11

                                Die Prozedur MEANS

                              Analysis Variable : Ertrag

                       Untere 95%            Obere 95%
                  KG für Mittelwert     KG für Mittelwert
                  ------------------------------------------
                       23.6346322            42.9807524
                  ------------------------------------------
```

Wir haben in den Anweisungen zur Berechnung eines Vertrauensintervalls weder die Kleinheit der Stichprobe (n<30) noch die Tatsache beachtet, dass die Varianz geschätzt werden muss. Proc Means arbeitet direkt mit einer t-Verteilung, deswegen müssen wir hier nichts Zusätzliches eingeben.

```
data m;
set m;
lower=mean-1.96*std/sqrt(_freq_);
upper=mean+1.96*std/sqrt(_freq_);
run;

proc print data=m;
run;
```

Im zweiten Procstep haben wir hier nun unerlaubterweise mit der Normalverteilung gearbeitet (das sehen wir an der 1,96, was der kritische Wert aus der Normalverteilung ist). Dabei soll nur deutlich werden, dass das Vertrauensintervall so bei kleineren Stichprobenumfängen kleiner wird als mit dem korrekten Wert aus der t-Verteilung.

Output:

```
                Das SAS System           19:39 Monday, August 18, 2008  12

Beob.    _TYPE_    _FREQ_      mean       std       lower      upper

  1          0         13     33.3077    16.0072    24.6061    42.0093
```

3.7 Stichprobenumfang zur Schätzung eines Mittelwertes

Zum Berechnen des Stichprobenumfangs zur Stichprobenplanung geben wir in SAS die Formel aus dem Kasten von S.45 ein und benützen SAS erneut wie einen Taschenrechner.

```
data Stpr_Umfang; /* Die einzelnen Variablen der Formel werden SAS genannt*/
sigma=10; /* Die geschätzte Varianz*/
HB=1; /* Die halbe Breite des Vertrauensintervalls*/
n=4*sigma**2 / (HB)**2;  /* Die Fromel von S.45. */
run;
```

Ein Sternchen bedeutet „mal“, zwei Sternchen bedeuten „hoch“ und „/“ bedeutet geteilt. Die grünen Zeilen sind übrigens direkt in SAS selber erzeugt worden. Mit /*…*/ kann Text in das Programm eingefügt werden, ohne dass dieser irgendeinen Einfluss auf die Analyse hat. Zum Merken der programmierten Zeilen ist dies sehr empfehlenswert und hilfreich.

```
proc print data=Stpr_Umfang;
run;
```

Output:

```
                Das SAS System        19:51 Wednesday, August 20, 2008   1

                              Beob.    sigma    HB      n

                                1        10      1     400
```

Und wir erhalten das gewünschte Ergebnis wie im Skript.

3.8 Vertrauensintervall für die Differenz von 2 Mittelwerten (verbundene Stichprobe)

Wir müssen, um den Vertrauensintervall zu bestimmen, zuerst die Differenz unserer Mittelwerte berechnen. Zunächst wird SAS mit den Daten gefüttert die es braucht. Die Anordnung der Daten im Datastep ist spaltenweise. Die Reihenfolge der Variablen hinter „`input`“ muss den Spalten des Datensatzes unter „`cards;`“ entsprechen. Erste Variable definiert erste Spalte usw.

```
data onfarm_Malawi;
input LM CCA;
```

```
cards;
2.2 3.5
2.2 2.0
1.9 2.9
1.2 0.4
1.3 0.6
0.9 0.5
1.0 0.6
0.5 0.3
1.8 2.2
1.1 0.7
1.6 0.9
1.0 0.3
1.6 1.1
0.6 0.3
;
run;
data onfarm_Malawi;
set onfarm_Malawi;
d=LM-CCA;
run;
proc univariate data=onfarm_Malawi all;
var d;
```

„proc univariate“ soll die Variable d (Differenz der Erträge) analysieren bzw. den Vertrauensintervall ermitteln.

```
run;
proc print data=onfarm_Malawi;
run;
```

Die Prozedur UNIVARIATE gibt im Output eine Menge Daten aus. Die Informationen über die Grenzen des Vertrauensintervalls sind aber zum Glück mit als Erstes angegeben (siehe rot geschriebene Zeilen):

```
                Das SAS System        19:51 Wednesday, August 20, 2008   2

                              Die Prozedur UNIVARIATE
                                    Variable:  d

                                      Momente

N                          14    Summe Gewichte             14
Mittelwert          0.18571429    Summe Beobacht.           2.6
Std.abweichung      0.64313791    Varianz            0.41362637
Schiefe             -1.4997287    Kurtosis           1.40591734
Unkorr. Qu.summe          5.86    Korr. Quad.summe   5.37714286
Variationskoeff.    346.305029    Stdfeh. Mittelw.   0.17188584


                              Grundlegende Statistikmaße

                Lage                              Streuung

          Mittelwert   0.185714     Std.abweichung            0.64314
          Median       0.400000     Varianz                   0.41363
          Modalwert    0.400000     Spannweite                2.10000
                                    Interquartilsabstand      0.50000

HINWEIS: Der angezeigte Modalwert ist der kleinste von 2 Modalwerten bei einer Häufigkeit von 2.
```

Modalwerte

Modalwert	Anzahl
0.4	2
0.7	2

Grundlegende Konfidenzgrenzen Normalverteilung vorausgesetzt

Parameter	Schätzwert	95% Konfidenzgrenzen	
Mittelwert	**0.18571**	**-0.18562**	**0.55705**
Std.abweichung	0.64314	0.46625	1.03612
Varianz	0.41363	0.21738	1.07355

Natürlich ist für uns der Vertrauensintervall des Mittelwertes wichtig. Die Werte unserer berechneten Differenzen d entsprechen hier den Mittelwerten. Also ist dieses unser gesuchtes Vertrauensintervall.

3.9 Vertrauensintervall für die Differenz von 2 Mittelwerten (unverbundene Stichprobe)

Es geht nun um unverbundene Stichproben. Deswegen müssen wir die Prozedur wechseln. Vorhin benutzten wir **`proc univariate`** und **`proc means`**. Jetzt müssen wir auf eine Prozedur zurückgreifen, die mit so genannten gemischten Modellen arbeitet. Demnach ist sie auch benannt: **`proc mixed`**. Zunächst erstellen wir wieder einen Datastep, der die Variablen Feld und Länge hat. Diese müssen wir hinter `input` angeben. Nun zum Procstep: Mit dem `class` Statement geben wir all die Variablen an, in denen sich Beobachtungen bei der Analyse unterschieden sollen, z.B. soll SAS die Felder und nicht die einzelnen Längen unterscheiden. Die Prozedur mixed arbeitet wie schon erwähnt mit Matrizen bzw. gemischten Modellen. Im `model` - Statement müssen wir SAS die Variablen der sog. Designmatrix angeben. Wir haben die Variable Länge und Feld. So wie wir es eingegeben haben, wird jeder Länge wird ein Feld zugeordnet. Mit dem Statement `lsmeans` werden die Mittelwerte (mit der Methode der kleinsten Quadrate – method of least squares) errechnet. Die Option `pdiff` bewirkt die Berechnung der Differenz der einzelnen Felder. Die Option **`Cl`** sorgt dafür dass das Confidence Limits, also das Vertrauensintervall berechnet wird.

```
data Laenge_Maispflanzen_2St;
input laenge feld;
cards;
192 2
176 2
179 2
```

```
169 2
193 2
169 2
166 2
188 2
160 2
163 2
159 2
178 2
178 2
172 2
163 2
175 2
162 2
172 2
170 2
183 2
173 2
190 2
182 2
175 2
182 2
168 2
160 2
165 2
163 2
172 2
170 2
172 2
186 2
186 2
172 2
164 2
177 2
173 2
169 2
182 2
175 1
172 1
179 1
167 1
163 1
154 1
163 1
164 1
157 1
177 1
186 1
165 1
175 1
194 1
176 1
162 1
166 1
169 1
170 1
181 1
168 1
166 1
180 1
164 1
179 1
170 1
150 1
```

```
192 1
170 1
173 1
170 1
150 1
174 1
164 1
182 1
188 1
157 1
165 1
172 1
168 1
179 1
179 1
164 1
162 1
178 1
162 1
182 1
171 1
182 1
183 1
;
run;
proc mixed data=Laenge_Maispflanzen_2St;
class feld;
model laenge=feld;
lsmeans feld/pdiff cl;
run;
```

Das steht dann im Output:

Das SAS System 11:02 Tuesday, September 16, 2008 16

Die Prozedur MIXED

Typ 3 Tests der festen Effekte

Effekt	Zähler Freiheitsgrade	Nenner Freiheitsgrade	F-Statistik	Pr > F
feld	1	88	1.52	0.2211

Kleinste-Quadrate-Mittelwerte

Effekt	feld	Schätzwert	Standardfehler	DF	t-Wert	Pr > |t|	Alpha	Untere	Obere
feld	1	171.18	1.3633	88	125.56	<.0001	0.05	168.47	173.89
feld	2	173.70	1.5242	88	113.96	<.0001	0.05	170.67	176.73

Differenzen Kleinste-Quadrate-Mittelwerte

Effekt	feld	_feld	Schätzwert	Standardfehler	DF	t-Wert	Pr > |t|	Alpha	Untere	Obere
feld	1	2	-2.5200	2.0449	88	-1.23	0.2211	0.05	-6.5839	1.5439

Die Grenzen des Intervalls sind rot markiert. Erscheinen sonst schwarz.

3.10 Test zum Vergleich zweier verbundener Stichproben

Der Datastep wird aus 3.8 kopiert. Dann berechnen wir in einem neuen Datastep die Differenz der beiden Sorten. Der dann folgende t- Test im Procstep wird mit **proc means** durchgeführt. Die Option t liefert im Output den t-Wert der Teststatistik und prt liefert den P-Wert. Dazu gleich mehr. Der Test soll mit den Differenzen durchgeführt werden also:

`var d;` für die Differenz.

```
data onfarm_Malawi;
input LM CCA;
cards;
2.2 3.5
2.2 2.0
1.9 2.9
1.2 0.4
1.3 0.6
0.9 0.5
1.0 0.6
0.5 0.3
1.8 2.2
1.1 0.7
1.6 0.9
1.0 0.3
1.6 1.1
0.6 0.3
;
run;
data onfarm_Malawi;
set onfarm_Malawi;
d=LM-CCA;
run;
Proc print data=onfarm_Malawi;
run;
proc means data=onfarm_Malawi t prt;
var d;
run;
```

Wie erwünscht erscheinen im Output die beiden Werte (s.u.). Der t- Wert ist der gleiche wie der der Teststatistik im Skript. Wegen Rundungsfehlern weichen sie etwas voneinander ab. Diesen kann mit einem Tabellenwert verglichen werden um über Signifikanz zu entscheiden. Neu ist der sog. **p-Wert** unter Pr > |t| (Probability > Betrag t-Wert). Er gibt auch Auskunft über die Signifikanz. Bzw. sagt er wie hoch die Überschreitungswahrscheinlichkeit der Teststatistik. Ist der p-Wert größer als das Signifikanzniveau α= 0,05, wird die Nullhypothese beibehalten (es bestehen keine Unterschiede zwischen den Sorten) wenn er kleiner ist, wird sie verworfen (es bestehen signifikante Unterschiede zwischen den Sorten).

Output:

```
                    Das SAS System       11:02 Tuesday, September 16, 2008  18

                      Die Prozedur MEANS

                    Analysis Variable : d
```

t-Wert	Pr > |t|
1.08	0.2996

α = 0,05 < 0,2996= p-Wert. Der p-Wert ist größer als das Signifikanzniveau demnach kann die Nullhypothese nicht verworfen werden und es konnten keine Unterschiede der Sorten nachgewiesen werden.

3.11 Test zum Vergleich zweier unverbundener Stichproben

S. 54 Beispiel Maispflanzen

Alternativ zu 3.10 können zwei unverbundene Stichproben in SAS mittels der Prozedur **`proc ttest`** verglichen werden. Im Versuch mit den Maispflanzen aus zwei Stichproben werden nun die zwei Felder in der Zielvariablen Länge verglichen. Der Datastep kann erneut aus 3.9 kopiert werden.

```
data Laenge_Maispflanzen_2St;
input laenge feld;
cards;
192 2
176 2
179 2
169 2
193 2
169 2
166 2
188 2
160 2
163 2
159 2
178 2
178 2
172 2
163 2
175 2
162 2
172 2
170 2
183 2
173 2
190 2
182 2
175 2
182 2
168 2
160 2
165 2
163 2
172 2
170 2
172 2
186 2
186 2
172 2
```

```
164 2
177 2
173 2
169 2
182 2
175 1
172 1
179 1
167 1
163 1
154 1
163 1
164 1
157 1
177 1
186 1
165 1
175 1
194 1
176 1
162 1
166 1
169 1
170 1
181 1
168 1
166 1
180 1
164 1
179 1
170 1
150 1
192 1
170 1
173 1
170 1
150 1
174 1
164 1
182 1
188 1
157 1
165 1
172 1
168 1
179 1
179 1
164 1
162 1
178 1
162 1
182 1
171 1
182 1
183 1
;
run;
proc ttest data=Laenge_Maispflanzen_2St;
class feld;
var laenge;
run;
```

Der t- Test kann nur 2 Zielgruppen vergleichen. Deswegen darf die `class`-Variable auch nicht mehr als 2 Gruppen beinhalten. In unserem Fall sind das die beiden Felder 1 und 2.

Die Variable in der die beiden Zielgruppen verglichen werden ist die Länge deshalb `var laenge;`.

Im Output erscheint ein negativer t-Wert, das liegt daran, dass SAS nicht mit Beträgen rechnet. Im Skript ist der Wert positiv, weil die Differenz der Mittelwerte in der Formel (S. 54 unter (1)) in Betragsstrichen steht. Da es sich aber um einen zweiseitigen Test handelt gilt dies auch für die positive, obere Grenze. Der t- Wert und die Methode sind wieder rot geschrieben.

Output:

Das SAS System 14:05 Wednesday, September 17, 2008 1

Die Prozedur TTEST

Variable: laenge

feld	N	Mittelwert	Std.abw.	Std.fehler	Minimum	Maximum
1	50	171.2	9.9769	1.4109	150.0	194.0
2	40	173.7	9.1992	1.4545	159.0	193.0
Diff (1-2)		-2.5200	9.6400	2.0449		

feld	Methode	Mittelwert	95% CL Mittelwert		Std.abw.	95% CL Std Dev	
1		171.2	168.3	174.0	9.9769	8.3340	12.4325
2		173.7	170.8	176.6	9.1992	7.5356	11.8121
Diff (1-2)	Gepoolt	-2.5200	-6.5839	1.5439	9.6400	8.4021	11.3091
Diff (1-2)	Satterthwaite	-2.5200	-6.5483	1.5083			

Methode	Varianzen	DF	t-Wert	Pr > |t|
Gepoolt	**Gleich**	**88**	**-1.23**	**0.2211**
Satterthwaite	Ungleich	86.188	-1.24	0.2170

Gleichheit der Varianzen

Methode	Zähler Freiheits-grade	Nenner Freiheits-grade	F-Statistik	Pr > F
Folded F	49	39	1.18	0.6042

Der p- Wert (siehe 3.10) ist größer als das Signifikanzniveau α= 0.05. Die Nullhypothese wird beibehalten womit kein Unterschied zwischen den beiden Feldern nachgewiesen werden konnte. Schön ist auch zu sehen, dass die Prozedur mit einer gemeinsamen Varianz und mit unterschiedlichen Varianzen testet. Siehe Skript wo extra eine gemeinsame Varianz berechnet wird.

S.55 Beispiel Tomatenerträge

In diesem Beispiel ist die Variable der Düngervarianten mit Buchstaben und nicht mit Zahlen unterschieden. Was in der Definition der Variablen hinter `input` mit dem $-Zeichen angegeben sein muss.

```
data Tomatenertraege;
input Ertrag Variante $;
datalines;
29.9 a
11.4 a
25.3 a
16.5 a
21.1 a
26.6 b
23.7 b
28.5 b
14.2 b
17.9 b
24.3 b
;
run;
proc print data=Tomatenertraege; /* Zur Kontrolle ob unser Datensatz richtig
vorhanden ist schauen wir ihn uns im Output an.*/
run;
```

```
                Das SAS System     14:05 Wednesday, September 17, 2008   4

                                Beob.    Ertrag    Variante

                                   1      29.9        a
                                   2      11.4        a
                                   3      25.3        a
                                   4      16.5        a
                                   5      21.1        a
                                   6      26.6        b
                                   7      23.7        b
                                   8      28.5        b
                                   9      14.2        b
                                  10      17.9        b
                                  11      24.3        b
```

```
proc ttest data=Tomatenertraege;
class variante;
var ertrag;
run;
```

Ansonsten wird im Procstep gleich vorgegangen wie im ersten Beispiel. Im Output, erscheint wieder ein negativer t- Wert jedoch mit dem gleichen Betrag wie im Skript.

```
                Das SAS System     14:05 Wednesday, September 17, 2008   5

                                  Die Prozedur TTEST

                                  Variable:  Ertrag

Variante       N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

a              5       20.8400      7.2456        3.2403     11.4000     29.9000
b              6       22.5333      5.4320        2.2176     14.2000     28.5000
Diff (1-2)             -1.6933      6.3028        3.8165
```

Variante	Methode	Mittelwert	95% CL Mittelwert		Std.abw.	95% CL Std Dev	
a		20.8400	11.8435	29.8365	7.2456	4.3410	20.8205
b		22.5333	16.8328	28.2339	5.4320	3.3907	13.3226
Diff (1-2)	Gepoolt	-1.6933	-10.3269	6.9402	6.3028	4.3353	11.5064
Diff (1-2)	Satterthwaite	-1.6933	-10.8923	7.5056			

Methode	Varianzen	DF	t-Wert	Pr > |t|
Gepoolt	Gleich	9	-0.44	0.6677
Satterthwaite	Ungleich	7.3369	-0.43	0.6787

Gleichheit der Varianzen

Methode	Zähler Freiheits-grade	Nenner Freiheits-grade	F-Statistik	Pr > F
Folded F	4	5	1.78	0.5400

Der p- Wert sagt uns bei dem Signifikanziveau von α= 0.05, dass es keinen Unterschied zwischen den Düngevarianten gibt. Die Nullhypothese, dass keine Unterschiede bestehen, kann also nicht verworfen werden.

3.12 Verbundene oder unverbundene Stichprobe?

Im Beispiel auf S.56/57 mit den abgenutzten Schuhen handelt es sich um verbundene Stichproben. Es wird vorgegangen wie unter 3.10.

```
data schuhe;
input matA matB;
datalines;
13.2 14.0
8.2 8.8
10.9 11.2
14.3 14.2
10.7 11.8
6.6 6.4
9.5 9.8
10.8 11.3
8.8 9.3
13.3 13.6
;
run;
data schuhe;
set schuhe;
d=matB-matA;
run;
proc print data=schuhe;
run;
proc means data=schuhe t prt mean;
var d;
run;
```

Mit den Optionen t, prt und mean erhalten wir t-Wert, p- Wert und Mittelwert, dieselben wie im Skript.

```
Das SAS System        21:13 Friday, September 19, 2008  63

                    Die Prozedur MEANS

                  Analysis Variable: d

     t-Wert     Pr > |t|      Mittelwert
  ------------------------------------------
       3.35       0.0085       0.4100000
```

Der t- Wert ist größer als der Tabellenwert 2,62 und der p-Wert kleiner als das Signifikanzniveau von α= 0.05. Der Test ist also signifikant. Die Materialien verschleißen unterschiedlich stark.

3.14 Stichprobenumfang für den unverbundenen t-Test

Für die Berechnung dieses Stichprobenumfangs können wir uns genauso wenig einer Prozedur aus SAS bedienen wie unter 3.7. Allerdings steht uns eine Option zur Verfügung, die uns unsere z- Quantile ausgibt. Die Option heißt `PROBIT(1-alpha/2)` bzw. `PROBIT(1-beta)`. Somit können wir die ganze Berechnung wieder im Datastep durchführen. Wir benennen alle unsere Variablen (Irrtumswahrscheinlichkeit des Tests α, die Güte 1-β, die kleinste nachzuweisende Differenz δ und die Fehlervarianz σ^2) und tippen die Formel aus dem Skript als letzte Funktion ab. Das Ergebnis ist wie immer in einem Table unter Work abgelegt und zusätzlich können wir uns das Ganze als Druckversion mit **`proc print`** ins Output- Fenster schreiben lassen.

```
data n;
delta=5;
STDDEV=sqrt(7.733);     /* Anstatt STDDEV (für Standard Deviation=
Standardabweichung/ -fehler für die kleinste Differenz) können wir auch var (die
Varianz)eingeben, macht aber keinen Unterschied, da es nur um die Definition
geht und nicht um eine Funktion.*/
z1=PROBIT(1-0.05/2);
z2=PROBIT(1-0.9);
n=2*STDDEV**2/delta**2*(z1+z2)**2;
run;
proc print data=n;
run;
```

Output:

```
Das SAS System        15:05 Tuesday, October 21, 2008  11

 Beob.    delta     STDDEV       z1         z2          n

   1        5      2.78083    1.95996    1.28155    6.50031
```

3.17 Test des Parameters μ

Beim Verglleich eines theoretischen Parameter μ_0 mit dem Mittelwert einer Stichprobe machen wir dasselbe wie beim Test zweier Stichproben. Nur fehlt uns für den theoretischen Mittelwert der Datensatz. Das lässt sich mit einer Option für die Nullhypothese lösen. Der theoretische Mittelwert wird einfach hinter die Syntaxelement ‚H0=' geschrieben. Sonst ist alles wie in den zuvor durchgeführten Tests.

```
data studincome;
input student    einkommen;
datalines;
 1       1403.58
 2        958.23
 3       1070.82
 4        721.13
 5       1506.01
 6        829.61
 7       1098.49
 8        956.65
 9        836.71
10        984.65
11        802.77
12        918.00
13       1138.98
14        787.09
15        801.24
16        789.14
17       1057.62
18        906.10
19        658.15
20       1079.36
21       1285.65
22       1313.86
23        878.86
24       1358.58
25        727.17
26        753.02
27       1202.45
28       1069.73
29        959.12
30       1265.52
31        828.05
32        826.88
33        896.77
34        959.09
35       1205.22
36        689.61
37       1257.00
38        895.90
39       1234.73
40        945.14
41        655.87
42        426.22
43        880.30
44       1131.70
45       1071.45
46        866.22
47       1181.82
48        758.84
49       1278.97
50        633.87
```

```
51        1059.56
52         964.03
53         686.94
54        1011.38
55         778.74
56         954.36
57         774.47
58        1157.87
59         987.75
60        1432.78
61         895.25
62        1356.59
63         735.76
64         912.21
65        1019.53
66         783.07
67        1089.14
68         614.35
69        1161.41
70        1180.63
71        1059.22
72        1057.40
73         726.17
74         850.12
75        1112.68
76        1147.10
77        1303.12
78         742.13
79         778.67
80        1165.63
;
run;
proc ttest data=studincome H0=1300;
var einkommen;
run;
```

Im Output erscheint wie gehabt der t-Wert. Der p- Wert ist deutlich kleiner als das Signifikanzniveau α. Die Nullhypothese, dass Studenten in Witzenhausen ein gleich großes Einkommen wie andere haben wird verworfen. Sie bekommen signifikant weniger Geld als der Bundesdurchschnitt.

```
                                Das SAS System        19:22 Saturday, November 1, 2008   1

                                Die Prozedur TTEST

                               Variable:  einkommen

     N     Mittelwert     Std.abw.     Std.fehler      Minimum      Maximum

    80          978.0        221.0        24.7085        426.2       1506.0

     Mittelwert     95% CL Mittelwert     Std.abw.       95% CL Std Dev

          978.0        928.8   1027.2        221.0        191.3    261.8

                            DF      t-Wert     Pr > |t|

                            79      -13.03       <.0001
```

3.18 Vertrauensintervall für eine Varianz

In 3.9 haben wir schon mit **proc mixed** das Vertrauensintervall für die Differenz zweier unabhängiger Stichproben bestimmt. Das Vertrauensintervall für eine Varianz lässt sich fast gleich bestimmen. Der Unterschied in diesem Beispiel ist, dass wir weder eine `class` noch eine `lsmeans` Anweisung (least square means: Kleinst-Quadrat-Mittelwerte) angeben brauchen. Die `class` Anweisung muss nicht angegeben werden weil wir hier keine Variablen haben für die wir einzelne Effekte berechnen wollen bzw. nach denen wir die gesamten Daten unterscheiden/einteilen möchten bspw. in einzelne Felder oder Düngevarianten. Außerdem brauchen wir auch keine Berechnung der Mittelwerte, deswegen fällt auch die `lsmeans` Anweisung weg. Was bleibt ist das Model, das wir angeben müssen: `model y=;`. Jetzt muss man wissen wie das Model aussieht: $y_{ij}= \mu_i+e_{ij}$. Was man noch wissen muss ist, dass in der Modelangabe „y=“ in SAS der Mittelwert μ_i und der Restfehler e_{ij} schon enthalten ist, das heißt das steht schon mit da, wenn wir nur „y=“ schreiben. Wenn wir hinter das = noch etwas schreiben würden, gäben wir ein Kovariable $\beta_{.i}$ an und mixed würde uns eine lineare Regression berechnen. Wollen wir aber nicht, deswegen: Weglassen. Wichtig ist, dass wir hier nur das Model $y_{ij}= \mu_i+e_{ij}$ brauchen und dies in Form von „y=“ angeben. Um die Vertrauensintervalle zu erhalten schreiben wir im Procstep hinter **proc mixed** `data=Mais` noch `cl;`. Cl steht für „confidence limits“= Vertrauensgrenzen.

```
data Mais;
input i      y;
datalines;
 1    63.53
 2    44.18
 3    49.07
 4    33.88
 5    67.98
 6    38.59
 7    50.27
 8    44.11
 9    38.90
10    45.33
11    37.43
12    42.43
13    52.03
14    36.75
15    37.36
16    36.83
17    48.50
18    41.92
19    31.14
20    49.44
21    58.40
```

```
22    59.63
23    40.73
24    61.57
25    34.14
26    35.27
27    54.79
28    49.02
29    44.22
30    57.53
31    38.52
32    38.47
33    41.51
34    44.22
35    54.91
36    32.51
37    57.16
38    41.47
39    56.19
40    43.61
41    31.05
42    21.07
43    40.79
44    51.72
45    49.10
46    40.18
47    53.89
48    35.52
49    58.11
50    30.09
;
proc mixed data=Mais cl;
model y=;
run;
```

Unter der Überschrift „Covariance Parameter Estimates“ im Output stehen die Grenzen des Intervalls.

Der Output sieht wieder folgendermaßen aus:

```
                    Das SAS System          17:17 Friday, October 24, 2008   5

                                  Die Prozedur MIXED

                                 Modellinformationen

                Data Set                     WORK.MAIS
                Dependent Variable           y
                Kovarianzstruktur            Diagonal
                Estimation Method            REML
                Residuenvarianzmethode       Profil
                Feste-Effekte-SE-Methode     Modellbasiert
                Freiheitsgradmethode         Residuum


                                    Dimensionen

                        Kovarianzparameter               1
                        Spalten in X                     1
                        Spalten in Z                     0
                        Subjekte                         1
                        Max Beob. je Subjekt            50
```

Anzahl der Beobachtungen

Number of Observations Read	50
Number of Observations Used	50
Number of Observations Not Used	0

Covariance Parameter Estimates

Kov.Parm	Schätzwert	Alpha	Untere	Obere
Residual	99.9971	0.05	69.7763	155.28

Anpassungsstatistiken

-2 Res Log-Likelihood	368.6
AIC (kleiner ist besser)	370.6
AICC (kleiner ist besser)	370.7
BIC (kleiner ist besser)	372.5

3.19 Test zum Vergleich zweier unabhängiger Stichprobenvarianzen

Für den Vergleich von unabhängigen Stichprobenvarianzen können wir genau wie zum Vergleich zweier unabhängiger Stichproben die Prozedur TTEST verwenden, da sie uns im Output den Vergleich der Varianzen bzw. den Test auf deren Gleichheit angibt. Data- und Procstep werden wie in den vorigen Beispielen mit **`proc ttest`** erstellt.

```
data tomaten;
input ertrag beh$;
cards;
29.9 a
11.4 a
25.3 a
16.5 a
21.1 a
26.6 b
23.7 b
28.5 b
14.2 b
17.9 b
24.3 b
;
run;
proc ttest data=tomaten;
class beh;
var ertrag;
run;
```

Im Output ist das Ergenis wieder als p-Wert angegeben, den es zu interpretieren gilt.

Die Prozedur TTEST

Variable: ertrag

beh	N	Mittelwert	Std.abw.	Std.fehler	Minimum	Maximum
a	5	20.8400	7.2456	3.2403	11.4000	29.9000
b	6	22.5333	5.4320	2.2176	14.2000	28.5000
Diff (1-2)		-1.6933	6.3028	3.8165		

beh	Methode	Mittelwert	95% CL Mittelwert		Std.abw.	95% CL Std Dev	
a		20.8400	11.8435	29.8365	7.2456	4.3410	20.8205
b		22.5333	16.8328	28.2339	5.4320	3.3907	13.3226
Diff (1-2)	Gepoolt	-1.6933	-10.3269	6.9402	6.3028	4.3353	11.5064
Diff (1-2)	Satterthwaite	-1.6933	-10.8923	7.5056			

Methode	Varianzen	DF	t-Wert	Pr > |t|
Gepoolt	Gleich	9	-0.44	0.6677
Satterthwaite	Ungleich	7.3369	-0.43	0.6787

Gleichheit der Varianzen

Methode	Zähler Freiheits-grade	Nenner Freiheits-grade	**F-Statistik**	**Pr > F**
Folded F	4	5	**1.78**	**0.5400**

Unser p-Wert (Überschreitungswahrscheinlichkeit) ist größer als das Signifikanzniveau α von 5%. Demnach können wir die Nullhypothese, dass die Varianzen signifikant verschieden sind, nicht verwerfen. Praktisch bedeutet das, dass die Düngevarianten sich in ihrer Stabilität nicht unterscheiden.

3.20 Einseitige und zweiseitige Tests

S. 76 Beispiel Schnakenproblem

In diesem Beispiel können wir genauso vorgehen wie in 3.17. Der Aufbau ist genau gleich. Es wird wieder ein Stichprobenmittelwert mit einem theoretischen Mittelwert verglichen. Eingabe vom Datastep (die Daten wurden hier simuliert, sie stehen nicht im Skript), Procstep, Nullhypothese nicht vergessen.

```
data schnaken;
input anzahl;
datalines;
18
12
16
9
1
13
11
```

```
1
9
17
5
7
9
19
13
10
20
5
13
12
;
run;
proc ttest data=schnaken H0=10;
var anzahl;
run;
```

```
                                Das SAS System         15:23 Tuesday, November 4, 2008  12

                                Die Prozedur TTEST

                                Variable:  anzahl

 N     Mittelwert     Std.abw.     Std.fehler      Minimum      Maximum

20        11.0000       5.4772         1.2247       1.0000      20.0000

 Mittelwert     95% CL Mittelwert     Std.abw.       95% CL Std Dev

    11.0000       8.4366  13.5634       5.4772       4.1654   7.9999

                          DF      t-Wert     Pr > |t|

                          19        0.82       0.4243
```

Der Schnakenbesatz ist nicht signifikant größer als die Bekämpfungsschwelle von 10 Tieren pro 350 ml da der p-Wert größer 0,05 (Sgnifikanzniveau) ist.

Allerdings haben wir hier jetzt nur zweiseitig getestet und nicht einseitig wie es aus der Fragestellung hervorgeht. Es gibt nun 2 Möglichkeiten einseitig zu testen. Einmal die reguläre Funktion über eine Option, deren Syntaxanweisung da heißt SIDES= 2, U oder L. L steht für Lower, also die untere Grenze, U für Upper, die obere Grenze und 2 steht für einen zweiseitigen Test. In unserem Fall müssen wir Sides=U eingeben, weil die Alternativhypothese besagt, dass der Besatz oberhalb der Schwelle 10 liegt. Zum anderen können wir einseitig testen, indem wir zunächst zweiseitig testen und dann den p-Wert halbieren, was einer Verdopplung des Signifikanzniveaus auf α=0.1 entspricht. Vom Prinzip her macht es keinen Unterschied, ob wir zweiseitig mit α=0.1 oder einseitig mit

α=0.05 testen. Dies aber nur als zusätzliche Information, um den Spielraum darzustellen, den man in SAS hat. Man kann mit vielen Möglichkeiten zum selben Ergebnis kommen. Der reguläre und wichtige Weg:

```
data schnaken;
input anzahl;
datalines;
18
12
16
9
1
13
11
1
9
17
5
7
9
19
13
10
20
5
13
12
;
run;
proc ttest data=schnaken Sides=U H0=10;
var anzahl;
run;
```

Output:

```
                                Das SAS System         15:23 Tuesday, November 4, 2008  15

                               Die Prozedur TTEST

                               Variable:  anzahl

 N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

20       11.0000      5.4772        1.2247      1.0000     20.0000

 Mittelwert    95% CL Mittelwert    Std.abw.      95% CL Std Dev

    11.0000      8.8823  Infty        5.4772      4.1654   7.9999

                        DF     t-Wert    Pr > t

                        19       0.82    0.2122


  N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

 20       11.0000      5.4772        1.2247      1.0000     20.0000
```

Das `Infty` zeigt uns, dass die obere Vertrauensgrenze für den Mittelwert bei Unendlich liegt.
Der t-Wert ist und muss auch derselbe bleiben. Der p-Wert ging nach unten, weil der eineitige Test (bei eindeutig einseitiger Fragestellung) genauer ist (es können mehr Signifikanzen auftreten). Trotz des kleineren p-Wertes ist das Ergebnis aber weiterhin nicht signifikant.

3.21 Äquivalenztest am Beispiel zweier unverbundener Stichproben

Bei einem Äquivalenztest wollen wir testen, ob zwei Werte bzw. hier Mittelwerte mit einem gewissen „Spielraum“ δ=1 gleichwertig (äquivalent) sind. Wir haben hier zwei Möglichkeiten das zu testen. Einmal über einen t-Test und zum anderen über einen Vertrauensintervall der Differenz unserer Mittelwerte, das sich innerhalb der von uns festgelegten Äquivalenzgrenzen befinden muss, um Äquivalenz nachzuweisen. Genau wie im Skript auf Seite 78 unten beschrieben (letzter Abschnitt), führen wir nun die beiden einfachen Tests durch.
Wir beginnen diese Tests in SAS wie immer mit dem Datastep und erreichen unser Ziel mit der Prozedur t-Test. Die Methodik ist dieselbe wie in den vorangegangenen Abschnitten.
Der Datastep:

```
data btToxin;
input n linie$;
cards;
12 b
15 b
13 b
10 b
8 b
10 i
17 i
12 i
9 i
16 i
;
run;
```

Hinter linie muss wieder das $-Zeichen, da diese Variable in Buchstaben (alphanumerisch) angegeben ist. (Alphanumerisch oder character-valued beschreibt in SAS alles außer Zahlen.)

Der Procstep:

In diesen ersten zwei Procsteps testen wir, ob unsere vorgegebenen Grenzen signifikant von den Grenzen des Vertrauensintervalls der Mittelwertdifferenz verschieden sind. Wir führen dazu zwei einseitige Tests durch. Zuerst gegen die untere Grenze [Erinnerung: bei sides=U (upper) ist die untere Grenze fest, also vorgegeben mit -1] und dann gegen die Obere [sides=l ;lower mit +1 nach oben hin festgelegt]. Dabei wird jeweils gegen die Nullhypothesen H_{01}=1 bzw. H_{02}=-1 (die Nullhypothesen entsprechen jeweils δ_1 und δ_2) getestet ob die jeweiligen Grenzen des Vertrauensintervalls der Mittelwertdifferenz signifikant von den betreffenden Äquivalenzgrenzen verschieden sind. Im Skript geschieht dieser Vergleich mit dem Tabellenwert t_{Tab}. Dieses Vorgehen wird hier durch Betrachtung des p-Wertes ersetzt.

```
/* Einseitiger Test mit H0=1 */
proc ttest data=btToxin H0=1 sides=u;
class linie;
var n;
run;
```

Output:

```
Das SAS System          14:34 Friday, November 7, 2008   9

                                  Die Prozedur TTEST

                                    Variable:  n

        linie          N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

        b              5       11.6000      2.7019        1.2083      8.0000     15.0000
        i              5       12.8000      3.5637        1.5937      9.0000     17.0000
        Diff (1-2)             -1.2000      3.1623        2.0000

linie         Methode          Mittelwert    95% CL Mittelwert    Std.abw.      95% CL Std Dev

b                                 11.6000      8.2452  14.9548      2.7019      1.6188   7.7639
i                                 12.8000      8.3751  17.2249      3.5637      2.1351  10.2405
Diff (1-2)    Gepoolt             -1.2000     -4.9191  Infty        3.1623      2.1360   6.0582
Diff (1-2)    Satterthwaite       -1.2000     -4.9546  Infty

                  Methode          Varianzen        DF     t-Wert    Pr > t

                  Gepoolt          Gleich            8      -1.10    0.8483
                  Satterthwaite    Ungleich     7.4564      -1.10    0.8472

                               Gleichheit der Varianzen

                                  Zähler       Nenner
                              Freiheits-   Freiheits-
                Methode           grade        grade    F-Statistik    Pr > F

                Folded F             4            4           1.74    0.6048
```

```
/* Einseitiger Test mit H0=-1 */
proc ttest data=btToxin H0=-1 sides=l;
class linie;
```

```
var n;
run;
```

Output:

```
Das SAS System          14:34 Friday, November 7, 2008  10

                                  Die Prozedur TTEST

                                     Variable:  n

        linie          N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

        b              5       11.6000      2.7019        1.2083      8.0000     15.0000
        i              5       12.8000      3.5637        1.5937      9.0000     17.0000
        Diff (1-2)             -1.2000      3.1623        2.0000

linie         Methode          Mittelwert    95% CL Mittelwert    Std.abw.      95% CL Std Dev

b                                 11.6000      8.2452  14.9548      2.7019      1.6188  7.7639
i                                 12.8000      8.3751  17.2249      3.5637      2.1351  10.2405
Diff (1-2)    Gepoolt             -1.2000    -Infty     2.5191      3.1623      2.1360   6.0582
Diff (1-2)    Satterthwaite       -1.2000    -Infty     2.5546

                Methode          Varianzen        DF     t-Wert    Pr < t

                Gepoolt          Gleich            8      -0.10    0.4614
                Satterthwaite    Ungleich     7.4564      -0.10    0.4615

                              Gleichheit der Varianzen

                                   Zähler       Nenner
                              Freiheits-    Freiheits-
                Methode            grade         grade    F-Statistik    Pr > F

                Folded F               4             4           1.74    0.6048
```

Der p-Wert ist in beiden Tests gößer als α=0.05. Das heißt also, die Nullhypothese, dass die Mittelwerte nicht äquivalent sind wird beibehalten. Äquivalenz konnte nicht nachgewiesen werden.

Nun testen wir nach der 2. Möglichkeit. Und zwar prüfen wir, ob das Vertrauensintervall der Differenz unserer beiden Mittelwerte innerhalb unserer vorgegebenen Äquivalenzgrenzen liegt, wobei die Irrtumswahrscheinlichkeit auf 2α erhöht wird, und zwar durch die Option **`alpha=0.1`**;. Dann wäre die Äquivalenz der Mittelwerte nachgewiesen. Also berechnen wir die Grenzen des Vertrauensintervalls. Deswegen brauchen wir hier auch keine Nullhypothesen, weil wir nur die Grenzen des Vertrauensintervalls suchen. Das machen wir wieder jeweils für beide Seiten einzeln mit einem einseitigen Test, weil wir die Ablehnungsbereiche einzeln betrachten und so eine höhere Genauigkeit/ Teststärke bekommen.

```
proc ttest data=btToxin alpha=0.1; /* Einfache Variante mit
Vertrauensintervall*/
class linie;
var n;
run;
```

Output:

```
                                  Die Prozedur TTEST

                                     Variable:  n

        linie          N    Mittelwert    Std.abw.    Std.fehler     Minimum     Maximum

        b              5       11.6000      2.7019        1.2083      8.0000     15.0000
        i              5       12.8000      3.5637        1.5937      9.0000     17.0000
        Diff (1-2)             -1.2000      3.1623        2.0000

linie         Methode          Mittelwert    90% CL Mittelwert    Std.abw.      90% CL Std Dev

b                                 11.6000      9.0241  14.1759      2.7019      1.7543   6.4098
i                                 12.8000      9.4024  16.1976      3.5637      2.3139   8.4544
Diff (1-2)    Gepoolt             -1.2000     -4.9191   2.5191      3.1623      2.2713   5.4107
Diff (1-2)    Satterthwaite       -1.2000     -4.9546   2.5546

                Methode          Varianzen        DF     t-Wert    Pr > |t|

                Gepoolt          Gleich            8      -0.60      0.5651
                Satterthwaite    Ungleich     7.4564      -0.60      0.5663

                                 Gleichheit der Varianzen

                                    Zähler        Nenner
                              Freiheits-    Freiheits-
                Methode          grade         grade    F-Statistik    Pr > F

                Folded F             4             4           1.74    0.6048
```

Die Grenzen sind -4,9 und 2,5. Unsere Äquivalenzgrenzen -1 und 1. Das Intervall ist darin nicht enthalten, also sind die Mittelwerte auch nicht äquivalent.

4. Die einfache Varianzanalyse

Wir bewerkstelligen die einfach Varianzanalyse in SAS mit der Prozedur GLM. Mit GLM ist es egal, ob unsere Datensätze balanciert oder unbalanciert sind. Es kann in jedem Fall eine globale Nullhypothese getestet werden.

4.4 Die Varianzanalysetabelle

S.91 Beispiel Sortenversuch:

Die Varianzanalysetabelle liefert uns letztlich alle Informationen die wir aus den Daten ziehen wollen bezüglich der Sortenunterschiede bzw. -gleichheit. Dementsprechend wird sie auch im Output nach der Durchführung mit Proc GLM angezeigt.

Folgendermaßen sieht der Datastep aus:

```
data Sortenertraege;
input sorte$ wdh ertrag;
datalines;
a 1 31
a 2 32
a 3 37
a 4 32
b 1 21
b 2 23
b 3 25
b 4 19
c 1 27
c 2 29
c 3 34
c 4 34
d 1 34
d 2 32
d 3 31
d 4 27
e 1 24
e 2 23
e 3 27
e 4 26
;
run;
```

Die Eingabe der Rohdaten unterscheidet sich von den bisher behandelten Prozeduren nicht. Hinter die `input`-Angabe der alphanumerischen Variablen (hier: `sorte`) muss wieder das $- Zeichen, ansonsten die einzelnen Variablen von links nach rechts und deren Daten von oben nach unten eingegeben werden.

Der Procstep:

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
lsmeans sorte/t;
run;
```

Einige der Anweisungen kennen wir schon. Sie haben immer dieselbe Bedeutung. Nach `class` geben wir die Variablen an, nach denen der Datensatz klassifiziert werden soll. Es sind immer unabhängige Variablen mit nominalem Niveau (Also vom „namengebenden“

Niveau bspw. Sortennamen). In diesem Fall heißt das, dass wir die einzelnen Sorten miteinander verglichen haben wollen, deswegen `class sorte;`.

Mit der Anweisung `model` bestimmen wir das Model nachdem wir die Varianzanalyse ansetzen wollen. Der Gesamtmittelwert μ und der Fehler/die Zufallsabweichung e sind in SAS immer schon im Model integriert ($y_{ij}=\mu+e_{ij}$). Wir müssen nur den Effekt τ_i der i-ten Behandlung bzw. des i- ten Sorteneffekts ins Model eintragen. Dann arbeitet SAS nach dem Model $yij= \mu+ \tau_i +e_{ij}$. Vor dem Gleichheitszeichen stehen immer die abhängigen Variablen und nach dem Gleichheitszeichen im Model die Unabhängigen. Das heißt, dass wir hier den Ertrag der verschiedenen Sorten modellieren und vergleichen wollen. Also ist die Sorte unser einziger unabhängiger Effekt der im Modell steht. Folglich geben wir an: `model ertrag= sorte;` was dem Model $yij= \mu+ \tau_i +e_{ij}$ entspricht.

Mit `lsmeans` geben wir die Methode an mit der die Mittelwerte berechnet werden sollen. Bei `lsmeans` ist das die Methode der kleinsten Quadrate (least square und means wie Mittelwerte). Wir könnten auch nur `means` angeben. `sorte/t` besagt, dass ein multipler t-Test über die einzelnen Sorten gemacht wird.

Der Output: Dies ist der gesamte Output wenn ein Procstep mit GLM durchgeführt wird. Die für uns wichtigen Werte bezüglich der Varianzanalysetabelle sind wieder rot geschrieben.

Proc Print liefert:

Das SAS System 15:40 Wednesday, November 19, 2008 1

Beob.	sorte	wdh	ertrag
1	a	1	31
2	a	2	32
3	a	3	37
4	a	4	32
5	b	1	21
6	b	2	23
7	b	3	25
8	b	4	19
9	c	1	27
10	c	2	29
11	c	3	34
12	c	4	34
13	d	1	34
14	d	2	32
15	d	3	31
16	d	4	27
17	e	1	24
18	e	2	23
19	e	3	27
20	e	4	26

Proc GLM liefert:

The GLM Procedure

Class Level Information

Class	Levels	Values
sorte	5	a b c d e

Number of Observations Read	20
Number of Observations Used	20

The GLM Procedure

Dependent Variable: ertrag

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	348.8000000	87.2000000	11.28	0.0002
Error	15	116.0000000	7.7333333		
Corrected Total	19	464.8000000			

R-Square	Coeff Var	Root MSE	ertrag Mean
0.750430	9.791856	2.780887	28.40000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sorte	4	348.8000000	87.2000000	11.28	0.0002

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sorte	4	348.8000000	87.2000000	11.28	0.0002

The GLM Procedure
Least Squares Means

sorte	ertrag LSMEAN	LSMEAN Number
a	33.0000000	1
b	22.0000000	2
c	31.0000000	3
d	31.0000000	4
e	25.0000000	5

Least Squares Means for Effect sorte
t for H0: LSMean(i)=LSMean(j) / Pr > |t|

Dependent Variable: ertrag

i/j	1	2	3	4	5
1		5.594024	1.017095	1.017095	4.068381
		<.0001	0.3252	0.3252	0.0010
2	-5.59402		-4.57693	-4.57693	-1.52564
	<.0001		0.0004	0.0004	0.1479
3	-1.0171	4.576929		0	3.051286
	0.3252	0.0004		1.0000	0.0081
4	-1.0171	4.576929	0		3.051286
	0.3252	0.0004	1.0000		0.0081
5	-4.06838	1.525643	-3.05129	-3.05129	
	0.0010	0.1479	0.0081	0.0081	

```
NOTE: To ensure overall protection level, only probabilities associated with pre-planned comparisons
      should be used.
```

Auffällig ist, dass zwei Varianzanalysetabellen angegeben werden (Typ I und Typ III). In der Statistik- und der Biometrievorlesung wird nur mit Typ I gearbeitet. Wir betrachten jedoch immer die Typ I Tabelle, da sie bei unbalancierten Daten eine höhere Teststärke liefert. Auch bei der Kovarianzanalyse später ist Typ I vorzuziehen. Obendrein sind beide oft identisch (Details dazu in der Biometrie-Vorlesung).

Zum Ergebnis des globalen Tests: Der p-Wert (`Pr > F 0.0002`) ist sehr klein, kleiner 0,05. Das heißt die Nullhypothese, dass alle Sorten (-mittelwerte) gleich sind, wird verworfen. Einzelne signifikante Unterschiede zwischen den Sorten selbst werden im nächsten Abschnitt getestet. Wer genau hingesehen hat, kann schon bemerkt haben, dass dies im letzten Abschnitt des Outputs bereits geschehen ist.

4.5 Multiple Mittelwertvergleiche

4.5.1 LSD-Test

S.93 Beispiel Sortenversuch:

Im letzten Abschnitt wurde der t-Test bzw. der LSD- Test schon mit durchgeführt. Normalerweise werden die durchzuführenden Tests wie in Variante 2 (s.u.) hinter einer `adjust`= Anweisung angegeben. Da der t-Test als „Grundeinstellung“ gemacht wird muss er nur mit einem /t hinter `lsmeans` und der Variablenangabe, über die die Mittelwerte erzeugt werden sollen, angegeben werden.

Variante 1:

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
lsmeans sorte/t;
run;
```

Variante 2:

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
lsmeans sorte/adjust=t;
run;
```

Der Output der Variante 1 ist folgender:

```
                                  Das SAS System      15:40 Wednesday, November 19, 2008  40

                                   Die Prozedur GLM

                      Klassifizierungsausprägungsinformationen

                Klasse          Ausprägungen        Werte

                sorte                      5        a b c d e

                wdh                        4        1 2 3 4


                 Anzahl gelesene Beobachtungen           20
                 Anzahl verwendete Beobachtungen         20


                                  Das SAS System      15:40 Wednesday, November 19, 2008  41

                                   Die Prozedur GLM

Abhängige Variable: ertrag

                                        Summe der       Mittleres
      Quelle                    DF       Quadrate        Quadrat    F-Statistik    Pr > F

      Modell                     4     348.8000000     87.2000000          11.28    0.0002

      Fehler                    15     116.0000000      7.7333333

      Korrigierte Summe         19     464.8000000


                R-Quadrat     Koeff.var    Wurzel MSE    ertrag Mittelwert

                 0.750430      9.791856      2.780887             28.40000


                                                       Mittleres
      Quelle                    DF       Typ I SS        Quadrat    F-Statistik    Pr > F

      sorte                      4     348.8000000     87.2000000          11.28    0.0002


                                                       Mittleres
      Quelle                    DF      Typ III SS       Quadrat    F-Statistik    Pr > F

      sorte                      4     348.8000000     87.2000000          11.28    0.0002


                                  Das SAS System      15:40 Wednesday, November 19, 2008  42
```

Die Prozedur GLM
Kleinste-Quadrate-Mittelwerte

sorte	ertrag LSMEAN	LSMEAN Anzahl
a	33.0000000	1
b	22.0000000	2
c	31.0000000	3
d	31.0000000	4
e	25.0000000	5

Kleinste-Quadrate-Mittelwerte für Effekt sorte
t für HO: LSMean(i)=LSMean(j) / Pr > |t|

Abhängige Variable: ertrag

i/j	1	2	3	4	5
1		5.594024	1.017095	1.017095	4.068381
		<.0001	0.3252	0.3252	0.0010
2	-5.59402		-4.57693	-4.57693	-1.52564
	<.0001		0.0004	0.0004	0.1479
3	-1.0171	4.576929		0	3.051286
	0.3252	0.0004		1.0000	0.0081
4	-1.0171	4.576929	0		3.051286
	0.3252	0.0004	1.0000		0.0081
5	-4.06838	1.525643	-3.05129	-3.05129	
	0.0010	0.1479	0.0081	0.0081	

HINWEIS: To ensure overall protection level, only probabilities associated with pre-planned comparisons should be used.

Der Output der Variante 2 ist:

Die Prozedur GLM

Klassifizierungsausprägungsinformationen

Klasse	Ausprägungen	Werte
sorte	5	a b c d e

Anzahl gelesene Beobachtungen	20
Anzahl verwendete Beobachtungen	20

Die Prozedur GLM

Abhängige Variable: ertrag

Quelle	DF	Summe der Quadrate	Mittleres Quadrat	F-Statistik	Pr > F
Modell	4	348.8000000	87.2000000	11.28	0.0002

Fehler	**15**	**116.0000000**	**7.7333333**
Korrigierte Summe	19	464.8000000	

R-Quadrat	Koeff.var	Wurzel MSE	ertrag Mittelwert
0.750430	9.791856	2.780887	28.40000

Quelle	**DF**	**Typ I SS**	**Mittleres Quadrat**	**F-Statistik**	**Pr > F**
sorte	**4**	**348.8000000**	**87.2000000**	**11.28**	**0.0002**

Quelle	DF	Typ III SS	Mittleres Quadrat	F-Statistik	Pr > F
sorte	4	348.8000000	87.2000000	11.28	0.0002

Die Prozedur GLM
Kleinste-Quadrate-Mittelwerte

sorte	ertrag LSMEAN	LSMEAN Anzahl
a	33.0000000	1
b	22.0000000	2
c	31.0000000	3
d	31.0000000	4
e	25.0000000	5

Kleinste-Quadrate-Mittelwerte für Effekt sorte
Pr > |t| für HO: LSMean(i)=LSMean(j)

Abhängige Variable: ertrag

i/j	**1**	**2**	**3**	**4**	**5**
1		**<.0001**	**0.3252**	**0.3252**	**0.0010**
2	<.0001		**0.0004**	**0.0004**	**0.1479**
3	0.3252	0.0004		**1.0000**	**0.0081**
4	0.3252	0.0004	1.0000		**0.0081**
5	0.0010	0.1479	0.0081	0.0081	

HINWEIS: To ensure overall protection level, only probabilities associated with pre-planned comparisons should be used.

In der ersten Variante, der Grundeinstellung, werden für die einzelnen Sorten die t- Werte (t- Werte → `Kleinste-Quadrate-Mittelwerte für Effekt sorte. t für HO: LSMean(i)=LSMean(j) / Pr > |t|`) und p- Werte angegeben. Was zusätzlich noch einen Vergleich des t-Wertes für jede Sorte mit t_{TAB} erlaubt. Für die Information über Signifikanzen, reicht der p- Wert jedoch

völlig aus. Ist der p-Wert kleiner als das Signifikanzniveau α= 0,05 besteht ein signifikanter Sortenunterschied.

Die Wahl der Variante bleibt eurer Vorliebe überlassen. Sicherer scheint mir jedoch die Variante mit den zusätzlichen t- Werten, da so einer Fehlinterpretation eines p- Wertes entgegen gewirkt werden kann, falls man sich mal vertut ob Signifikanz bei großem oder kleinem p- Wert besteht. Ein hoher t- Wert deutet da gleichzeitig auf Signifikanz hin und bietet doppelte Absicherung.

4.5.1.1 Buchstabendarstellung im LSD Test

Um eine Buchstabendarstellung im Output zu erzeugen, müssen im Procstep folgende Änderungen vorgenommen werden:

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
means sorte/lsd;
run;
```

Wir müssen `lsmeans` in `means` ändern und hinter **`sorte`** die Option `lsd` eingeben.

Im Output erscheint zusätzlich dann die Buchstabendarstellung:

```
                                  Das SAS System          16:28 Monday, December 1, 2008   34

                                   Die Prozedur GLM

                                t-Tests (LSD) für ertrag

HINWEIS: Dieser Test kontrolliert den Fehler erster Art pro Vergleich, nicht den Fehler für das
                                      gesamte Experiment.


                                Alpha                             0.05
                                Freiheitsgrade des Fehlers          15
                                Mittlerer quadratischer Fehler   7.733333
                                Kritischer Wert von t            2.13145
                                Geringste signifikante Differenz  4.1912


                    Mittelwerte mit demselben Buchstaben sind nicht signifikant verschieden.

                          t Gruppierung    Mittelwert      N    sorte

                                     A        33.000      4    a
                                     A
                                     A        31.000      4    d
                                     A
                                     A        31.000      4    c

                                     B        25.000      4    e
                                     B
                                     B        22.000      4    b
```

Auf die Angaben über den rot markierten Zeilen sei extra hingewiesen, die im Skript auf S.93 oberes Beispiel alle ebenfalls aufgeführt sind.

4.5.3 Der Tukey- Test

Mit dem LSD- Test haben wir vergleichsbezogen getestet. Mit dem Tukey- Test testen wir nun versuchsbezogen.
Der Datensatz und der Datastep bleiben gleich. Auch im Procstep ändert sich nur eine Optionsangabe. Den Procstep der für die Varianzanalysetabelle mit lsmeans etc. notwendig ist, gebe ich hier nur der Vollständigkeit halber an. Auf Seite 98/99 ist diese nicht aufgeführt :

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
lsmeans sorte/adjust=tukey;
run;
```

Output:

```
                                  Das SAS System       16:05 Saturday, November 29, 2008  13

                                   Die Prozedur GLM

                      Klassifizierungsausprägungsinformationen

                    Klasse          Ausprägungen        Werte

                    sorte                      5        a b c d e


                     Anzahl gelesene Beobachtungen           20
                     Anzahl verwendete Beobachtungen         20

                                  Das SAS System       16:05 Saturday, November 29, 2008 14

                                   Die Prozedur GLM

Abhängige Variable: ertrag

                                          Summe der       Mittleres
       Quelle                     DF       Quadrate        Quadrat    F-Statistik    Pr > F

       Modell                      4     348.8000000     87.2000000          11.28    0.0002

       Fehler                     15     116.0000000      7.7333333

       Korrigierte Summe          19     464.8000000
```

R-Quadrat	Koeff.var	Wurzel MSE	ertrag Mittelwert
0.750430	9.791856	2.780887	28.40000

Quelle	DF	Typ I SS	Mittleres Quadrat	F-Statistik	Pr > F
sorte	4	348.8000000	87.2000000	11.28	0.0002

Quelle	DF	Typ III SS	Mittleres Quadrat	F-Statistik	Pr > F
sorte	4	348.8000000	87.2000000	11.28	0.0002

Die Prozedur GLM
Kleinste-Quadrate-Mittelwerte
Korrektur für multiple Vergleiche: Tukey

sorte	ertrag LSMEAN	LSMEAN Anzahl
a	33.0000000	1
b	22.0000000	2
c	31.0000000	3
d	31.0000000	4
e	25.0000000	5

Kleinste-Quadrate-Mittelwerte für Effekt sorte
Pr > |t| für HO: LSMean(i)=LSMean(j)

Abhängige Variable: ertrag

i/j	1	2	3	4	5
1		0.0004	0.8436	0.8436	0.0076
2	0.0004		0.0029	0.0029	0.5627
3	0.8436	0.0029		1.0000	0.0535
4	0.8436	0.0029	1.0000		0.0535
5	0.0076	0.5627	0.0535	0.0535	

4.5.3.1 Buchstabendarstellung im Tukey- (HSD-) Test

S. 98 Beispiel Sortenversuch

Die Änderungen um hier eine Buchstabendarstellung zu erzielen entsprechen denen vom LSD- Test. `lsmeans` wird zu `means` und `lsd` muss in `tukey` umgeschrieben werden:

```
proc glm data=sortenertraege;
class sorte;
model ertrag= sorte;
means sorte/tukey;
run;
```

Output:

Die Prozedur GLM

Tukey-Test der Studentisierten Spannweite (HSD) für ertrag

HINWEIS: Dieser Test kontrolliert den Fehler erster Art für das gesamte Experiment weist i.A. jedoch einen höheren Fehler zweiter Art auf als REGWQ.

Alpha	0.05
Freiheitsgrade des Fehlers	15
Mittlerer quadratischer Fehler	7.733333
Kritischer Wert der Studentisierten Spannweite	**4.36698**
Kleinste signifikante Differenz	**6.072**

Mittelwerte mit demselben Buchstaben sind nicht signifikant verschieden.

Tukey Gruppierung	Mittelwert	N	sorte
A	33.000	4	a
A			
B A	31.000	4	d
B A			
B A	31.000	4	c
B			
B C	25.000	4	e
C			
C	22.000	4	b

Wenn man `means sorte/lsd tukey` eingibt werden beide Buchstabendarstellungen (für LSD und HSD) in einem Output angegeben.

5. Einführung in die schließende Statistik für kategoriale Daten

In diesem Kapitel müssen wir viel „zu Fuß“ rechnen. D.h. wir werden SAS oft nur wie einen Taschenrechner benützen indem wir die Formeln in einem Datastep nachvollziehen.

5.1 Kombinatorik

Beispiel S. 100

Wir wollen 3! („Drei Fakultät“) berechnen. Dazu müssen wir zuerst eine Variable definieren. Diese ist hier einfach x. Für Fakultät gibt es die Funktion **`fact();`** mit der wir 3! berechnen können. Mit der print Prozedur bekommen wir das Ergebnis im Output angezeigt.

```
data beh;
x=fact(3);
run;
proc print data=beh;
run;
```

Output:

```
Das SAS System       11:20 Saturday, December 27, 2008   9

    Beob.     x

      1       6
```

S. 101 Variation mit Zurücklegen:

Um die Anzahl der möglichen Basentriplets zu berechnen müssen wir nur 4^3 rechnen. Das geht mit SAS folgendermaßen: Multipliziert wird bekanntermaßen mit * und potenziert mit **. Demnach der Datastep:

```
data vmz;
x=4*4*4;
x1=4**3;
run;
proc print data=vmz;
run;
```

Es kommt heraus:

```
Das SAS System       11:20 Saturday, December 27, 2008  10

Beob.      X     x1

  1       64     64
```

Wir können aber auch die Variablen und die Formel dazu angeben, was v.a. bei komlexeren Rechnungen üblich ist. Darauf zu achten ist, dass die Definitionen der Variablen vor der Formel stehen. Da SAS der Reihe nach einliest müssen die Variablen definiert sein bevor die Formel gelesen wird.

```
data vmzf;
n=4;
k=3;
x=n**k;
run;
proc print data=vmzf;
run;
```

Output:

```
Das SAS System       11:20 Saturday, December 27, 2008  13

Beob.     n     k      x

  1       4     3     64
```

S. 101 Variation ohne Zurücklegen

Die Variation ohne Zurücklegen entspricht der Formel auf S. 102 oben. Der Datastep dazu (zwei Möglichkeiten mit Funktionen fact() und perm()):

```
data vozf;
n=5;
```

```
k=3;
x=fact(n)/(fact(n-k));
x1=perm(n,k);
run;
proc print data=vozf;
run;
```

Output:

```
                                  Das SAS System       11:20 Saturday, December 27, 2008  14

                                 Beob.    n    k     x    x1

                                   1      5    3    60    60
```

S.102 Kombination mit Zurücklegen

Für „n über k“ Kombinationen gibt es die Funktion comb(n,k). Die umständlichere Version wäre die Formel einzugeben wie im Beispiel oben.

Datastep:

```
data koz;
x=comb(49,6);
run;
Proc print data=koz;
run;
```

Output:

```
                                  Das SAS System       11:20 Saturday, December 27, 2008  15

                                  Beob.        x

                                    1      13983816
```

S.102 Kombination mit Zurücklegen

Die Funktion comb wird hier mit dem Eingeben der Formel kombiniert.

```
data kmz;
n=9;
k=6;
x=comb((n+k-1), k);
run;
proc print data=kmz;
run;
```

Output:

```
                                  Das SAS System       11:20 Saturday, December 27, 2008  16

                                 Beob.    n    k      x

                                   1      9    6    3003
```

5.2 Einige wichtige Grundregeln der Wahrscheinlichkeitsrechnung

In den folgenden Beispielen, die im Skript auf den Seiten 104-106 stehen, führen wir die Rechnungen alle im Datastep ohne Prozedur bzw. Procstep durch. D.h. SAS ist wieder

unser Taschenrechner. Wir gehen so vor, dass immer zuerst die Variablen und dann die Formeln definiert werden. Über **`proc print`** erscheint dann alles im Output- Fenster.

S.104 Beispiel Maisfeld

```
data krankerMais;
p=5000/100000*100;
p0=100-5000/100000*100;
p1=5000/100000;
p2=1-5000/100000;
run;
proc print data=krankerMais;
run;
```

Output:

```
                          Das SAS System          20:26 Sunday, December 28, 2008   1

Beob.     p     p0      p1       p2

  1       5     95     0.05     0.95
```

S.104 Beispiel Landsberger Gemenge

```
data lbg;
pG=100000/200000*100;
pW=60000/200000*100;
pK=40000/200000*100;
pG1=100000/200000;
pW1=60000/200000;
pK1=40000/200000;
pWK=pW+pK;
run;
Proc print data=lbg;
run;
```

Output:

```
                          Das SAS System          20:26 Sunday, December 28, 2008   2

Beob.     pG     pW     pK     pG1     pW1     pK1     pWK

  1       50     30     20     0.5     0.3     0.2      50
```

S.105 Würfel Additionssatz

```
data Wuerfel;
p1=1/6;
p2=1/6;
p3=1/6;
p4=1/6;
p5=1/6;
p6=1/6;
pG=p1+p2+p3+p4+p5+p6;
pgG=p2+p4+p6;
puG=p1+p3+p5;
```

```
run;
proc print data=Wuerfel;
run;
```

Output:

```
                                    Das SAS System         20:26 Sunday, December 28, 2008   3

Beob.       p1          p2          p3          p4          p5          p6       pG     pgG     puG

  1      0.16667     0.16667     0.16667     0.16667     0.16667     0.16667      1     0.5     0.5
```

S.105 Supermarkt würfelt

```
Data Multiplikationssatz;
p6=1/6*1/6*1/6*1/6;
pc6=1/6*1/6*1/6*1/6*100;
run;
proc print data=Multiplikationssatz;
run;
```

Output:

```
                                    Das SAS System         20:26 Sunday, December 28, 2008   4

                                Beob.            p6        pc6

                                  1      .000771605     0.077160
```

Wenn Zahlen wie hier p6 angegeben werden, fehlt einfach die 0 vor dem Komma.

S.106 Blutgruppenbeispiel

```
Data lorenz;
p1=0.38*0.85;
p2=0.42*0.85;
p3=0.13*0.85;
p4=0.07*0.85;
p5=0.38*0.15;
p6=0.42*0.15;
p7=0.13*0.15;
p8=0.07*0.15;
pG=p1+p2+p3+p4+p5+p6+p7+p8;
run;
Proc print data=lorenz;
run;
```

Output:

```
                                    Das SAS System         20:26 Sunday, December 28, 2008   5

      Beob.       p1        p2        p3         p4         p5        p6         p7         p8       pG

        1       0.323     0.357     0.1105     0.0595     0.057     0.063     0.0195     0.0105      1
```

5.3 Binomialverteilung

S.109 Beispiel Maiszünsler

Mit den unten eingefügten Datasteps versuchen wir die Tabelle auf S. 111 im Skript nachzuvollziehen. Jeder Stichprobenumfang bekommt einen Datastep. Zunächst definieren wir im Datenschritt den Parameter p (Befallsstärke) und die Variable n (Stichprobenumfang). Mit der Anweisung `KEEP` sagen wir SAS, welche Vraiablen später im Output bzw. in der dafür angelegten Datei Maiszuensler1 ausgegeben werden sollen. Das sind die Anzahl befallener Pflanzen k, w für die Wahrscheinlichkeit wie hoch eine bestimmte Anzahl an Pflanzen befallen ist und ws, die aufsummierte Wahrscheinlichkeit über alle Befallsmöglichkeiten, die am Ende immer 1 ergibt. Nun müssen wir SAS sagen, dass es die Wahrscheinlichkeit für jede einzelne Möglichkeit des Befalls in einer Stichprobe berechnet. Bspw. bei n=1 soll es die Wahrscheinlichkeit von 0 befallenen Pflanzen berechnen und von 1 befallenen Pflanze. Dies erreichen wir, in dem wir mit der `DO TO` - Anweisung eine Schleife programmieren. Einfach übersetzt sagen wir SAS: Mache/ berechne die Wahrscheinlichkeiten für k= 0 befallene Pflanzen bis k= n befallene Pflanzen. Was der Zeile `DO k=0 TO n;` entspricht. Es müssen jetzt noch die Wahrscheinlichkeiten und ihre „Bedingungen“ definiert werden. Die erste Wahrscheinlichkeit ist w. PDF bedeutet *Probability Density (Mass) Function*. Diese Wahrscheinlichkeitsdichtefunktion soll mit einer Binomialverteilung (`'BINOM'`) über die Parameter k, p und n berechnet werden, die wir oben benannt haben. Bei der aufsummierten Wahrscheinlichkeit ws muss die CDF *Cumulative Distribution Function* angewandt werden. Mit der Anweisung `output;` erzeugen wir zum einen eine Datei im Work Ordner, in dem wir die Ergebnisse später abrufen können, und zum anderen sorgen wir dafür, dass alle n Zeilen im Output angezeigt werden. Ohne `output;` erscheint nur die letzte Zeile. `end;` beendet die Schleife. Nun muss für die drei Stichprobenumfänge auf S.111 nur das n im Datastep geändert werden.

```
data Maiszuensler1;
p=0.05; n=1;
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
ws=CDF('BINOM',k,p,n);
output;
end;
run;
```

```
proc print data=Maiszuensler1;
run;
```

Output:

```
                                  Das SAS System           09:58 Monday, January 5, 2009   1

                                Beob.     k       w       ws

                                  1       0     0.95     0.95
                                  2       1     0.05     1.00
```

```
data Maiszuensler2;
p=0.05; n=2;
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
ws=CDF('BINOM',k,p,n);
output;
end;
run;
proc print data=Maiszuensler1;
run;
```

Output:

```
                                  Das SAS System           09:58 Monday, January 5, 2009   2

                                Beob.     k       w         ws

                                  1       0     0.9025    0.9025
                                  2       1     0.0950    0.9975
                                  3       2     0.0025    1.0000
```

```
data Maiszuensler3;
p=0.05; n=3;
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
ws=CDF('BINOM',k,p,n);
output;
end;
run;
proc print data=Maiszuensler1;
run;
```

Output:

```
                                  Das SAS System           09:58 Monday, January 5, 2009   3

                                Beob.     k       w          ws

                                  1       0     0.85738    0.85738
                                  2       1     0.13538    0.99275
                                  3       2     0.00713    0.99988
                                  4       3     0.00013    1.00000
```

S. 113 Beispiel Zucchini

Es wird hier genauso vorgegangen wie im Maiszünslerbeispiel. Es ändern sich nur p und n.

```
data Zucchini;
p=0.75; n=5;
```

```
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
ws=CDF('BINOM',k,p,n);
output;
end;
run;
proc print data= Zucchini;
run;
```

Output:

Das SAS System 09:58 Monday, January 5, 2009 4

Beob.	k	w	ws
1	0	0.00098	0.00098
2	1	0.01465	0.01562
3	2	0.08789	0.10352
4	3	0.26367	0.36719
5	4	0.39551	0.76270
6	5	0.23730	1.00000

```
/* Zucchini mit n=10 */
data Zucchini10;
p=0.75; n=10;
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
ws=CDF('BINOM',k,p,n);
output;
end;
run;
proc print data= Zucchini10;
run;
```

Output:

Das SAS System 09:58 Monday, January 5, 2009 5

Beob.	k	w	ws
1	0	0.00000	0.00000
2	1	0.00003	0.00003
3	2	0.00039	0.00042
4	3	0.00309	0.00351
5	4	0.01622	0.01973
6	5	0.05840	0.07813
7	6	0.14600	0.22412
8	7	0.25028	0.47441
9	8	0.28157	0.75597
10	9	0.18771	0.94369
11	10	0.05631	1.00000

S.115 Beispiel für die Rekursionsformel

Auch werden wieder die oben genannten Befehle verwendet.

```
data Bsp;
p=0.75; n=5;
KEEP k w ws;
DO k=0 TO n;
w=PDF('BINOM',k,p,n);
```

```
ws=CDF('BINOM',k,p,n);
output;
end;
run;
proc print data=Bsp;
run;
```

Output:

```
                    Das SAS System           09:58 Monday, January 5, 2009   7

                Beob.    k       w          ws

                  1      0    0.00098    0.00098
                  2      1    0.01465    0.01562
                  3      2    0.08789    0.10352
                  4      3    0.26367    0.36719
                  5      4    0.39551    0.76270
                  6      5    0.23730    1.00000
```

5.3.1 Mittelwert und Varianz einer Binomialverteilung

S. 116 Beispiel 1

Die Formeln aus dem Kasten werden zu Fuß in einem Datastep wie folgt berechnet:

```
data mwvar;
n=100;
p=0.05;
q=1-p;
MW=n*p;
V=n*p*q;
run;
proc print data=mwvar;
run;
```

Output:

```
                    Das SAS System           09:58 Monday, January 5, 2009   8

                Beob.     n       p       q      MW      V

                  1      100    0.05    0.95     5    4.75
```

5.3.2 Schätzen des Parameters p der Binomialverteilung

S. 116 Beispiel 2

Auch der Parameter p wird anhand der Formeln zu Fuß berechnet:

```
data parmP;
n=100;
x=7;
p=0.07;
VvonP=(p*(1-p))/n;
sfp=sqrt(VvonP);
run;
proc print data=parmP;
run;
```

Output:

```
                          Das SAS System           09:58 Monday, January 5, 2009   9

Beob.      n      x       p       VvonP         sfp

  1       100     7     0.07     .000651     0.025515
```

5.3.3 Test für den Parameter der Binomialverteilung

Asymptotischer Test S.118 Beispiel Zuckerrüben

Um hier mit SAS zu einem z-Wert zu kommen, können wir die Prozedur Freq anwenden. Im Datastep geben wir die Zahl der gesunden und der befallenen Pflanzen an und verwenden nicht die Gesamtzahl wie im Beispiel. Das hängt mit der Vorgehensweise von SAS zusammen. Die erste Anweisung im Procstep ist `weight`. Mit ihr können wir eine Gewichtung in den Daten entsprechend der Anzahl der einzelnen Merkmale erzielen. Mit der Anweisung `table` legen wir die Variable fest, nach der die Häufigkeiten berechnet werden sollen. Danach muss noch die Verteilung (hier binomial) mit der Wahrscheinlichkeit p angegeben werden.

```
data Zuckerrueben;
input zahl fall$;
datalines;
143 befallen
424 gesund
;
run;
proc freq data=Zuckerrueben;
weight zahl;
table fall/binomial(p=0.25);
run;
```

Output:

```
                          Das SAS System           09:58 Monday, January 5, 2009  10

                          Die Prozedur FREQ

                                                 Kumulative      Kumulativer
fall          Häufigkeit     Prozent     Häufigkeit      Prozentwert
-------------------------------------------------------------------------
befallen          143       25.22           143          25.22
gesund            424       74.78           567         100.00


                  Binomialverhältnis
                  für fall = befallen
                ---------------------------------
                Proportion                 0.2522
                ASE                        0.0182
                95% Untere Konf.grenze     0.2165
                95% Obere Konf.grenze      0.2880

                Exakte Konfidenzgrenzen
                95% Untere Konf.grenze     0.2169
                95% Obere Konf.grenze      0.2901
```

```
Test von H0: Verhältnis = 0.25

ASE unter H0                0.0182
Z                           0.1212
Einseitige Pr >  Z          0.4518
Zweiseitige Pr > |Z|        0.9035

      Stichprobengröße = 567


        Das SAS System           09:58 Monday, January 5, 2009  11

    Beob.    zahl      fall

      1       143    befallen
      2       424    gesund
```

Der erhaltene z-Wert entspricht nicht ganz dem im Skript, weil dort der Standardfehler mit dem Wert für p under H0 berechnet wird (p=0,25), während er in SAS mit der Schätzung von p Berechnet wird. Führt aber zum selben Ergebnis: H_0 wird beibehalten.

Alternativ lässt sich das Ganze natürlich auch wieder zu Fuß nach der Formel rechnen:

```
data q;
z=sqrt(567)*((0.0022)/sqrt(0.25*0.75));
run;
proc print data=q;
run;
```

Output:

```
        Das SAS System           09:58 Monday, January 5, 2009  12

         Beob.       z

           1      0.12098
```

Exakter Test S.119 Beispiel Kreuzungsexperiment

Wird beispielhaft mit den Anweisungen aus 5.3.4 siehe S. 125 durchgeführt.

5.3.4 Vertrauensintervall für den Parameter p der Binomialverteilung

Asymptotisches & exaktes Vertrauensintervall

Beide werden mit den Anweisungen, die auf S. 125 angegeben sind, berechnet. **`proc freq`** ist wieder die Prozedur der Wahl mit denselben Anweisungen, die oben erklärt sind. Allerdings wird hier hinter der Option `binomial` kein p angegeben, weil wir alle Werte von p abdecken wollen, für die diese Nullhypothese angenommen würde.

```
data keimen;
input keimfaehig$ zahl;
datalines;
nein 1
ja 9
```

```
;
run;
proc freq data=keimen;
weight zahl;
tables keimfaehig/binomial;
run;
```

Output:

```
                          Das SAS System          09:58 Monday, January 5, 2009  16

                          Die Prozedur FREQ

                                                  Kumulative      Kumulativer
keimfaehig     Häufigkeit      Prozent      Häufigkeit       Prozentwert
-------------------------------------------------------------------------------
ja                      9        90.00               9             90.00
nein                    1        10.00              10            100.00


                  Binomialverhältnis
                  für keimfaehig = ja
                ---------------------------------
                Proportion                 0.9000
                ASE                        0.0949
                95% Untere Konf.grenze     0.7141
                95% Obere Konf.grenze      1.0000

                Exakte Konfidenzgrenzen
                95% Untere Konf.grenze     0.5550
                95% Obere Konf.grenze      0.9975

                   Test von H0: Verhältnis = 0.5

                ASE unter H0               0.1581
                Z                          2.5298
                Einseitige Pr >  Z         0.0057
                Zweiseitige Pr > |Z|       0.0114

                      Stichprobengröße = 10
```

5.3.5 Vergleich von zwei Binomialwahrscheinlichkeiten- unverbundene Stichproben

Mit **`proc freq`** können wir herausfinden, ob die beiden Saatgutpartien sich in ihrer Keimfähigkeit signifikant unterscheiden. Der Datastep dazu hat drei Variablen: Die Partie, ihre Keimfähigkeiten und deren Unterscheidung in keimfähig und nicht keimfähig (fall$). Die Anweisungen im Procstep unterscheiden sich zu den vorher besprochenen nur hinter der `table` Anweisung: Wir haben drei Variablen im Datastep. Nach einer wird gewichtet. Hier: keim. Nach den anderen beiden soll getestet werden. Sie werden mit einem Malzeichen (*) an die `table` Funktion angehängt. Da zwei nominale Variablen/Daten vorliegen muss mit dem Chi²-Test getestet werden. Das gibt man mit der Option `chisq;` hinter den `table` Anweisungen an. Heraus kommt allerdings z^2 und nicht z, weil mit einem

Chi²-Test getestet wurde. Deswegen kommt noch ein Datastep hinten dran, der die Quadratwurzel ausrechnet.

```
data Keimfaehigkeit;
input partie keim fall$;
datalines;
1 289 ja
2 274 ja
1 27 nein
2 42 nein
;
run;
proc freq data=Keimfaehigkeit
weight keim;
table partie*fall/ chisq;
run;
data z;
zvers=sqrt(3.6605);
run;
proc print data=z;
run;
```

Output 1:

```
                    Das SAS System          09:58 Monday, January 5, 2009  17

                    Die Prozedur FREQ

                  Table of partie by fall

partie     fall

Häufigkeit
Prozent
Row Pct
Col Pct    |ja       |nein     |  Summe
-----------+---------+---------+---------
         1 |     289 |      27 |     316
           |   45.73 |    4.27 |   50.00
           |   91.46 |    8.54 |
           |   51.33 |   39.13 |
-----------+---------+---------+---------
         2 |     274 |      42 |     316
           |   43.35 |    6.65 |   50.00
           |   86.71 |   13.29 |
           |   48.67 |   60.87 |
-----------+---------+---------+---------
Summe      |     563 |      69 |     632
           |   89.08 |   10.92 |  100.00


    Statistiken für Tabelle von partie nach fall.

Statistik                       DF        Wert      Prob
------------------------------------------------------------
Chi-Quadrat                      1      3.6605    0.0557
Likelihood-Quot. Chi-Quad.       1      3.6867    0.0548
Kontinuitätskorr. Chi-Quad.      1      3.1887    0.0741
Mantel-Haenszel Chi-Quadrat      1      3.6547    0.0559
Phi-Koeffizient                         0.0761
Kontingenzkoeffizient                   0.0759
Cramers V                               0.0761
```

```
        Exakter Test von Fisher
-------------------------------------------
Zelle (1,1) Häufigkeit (F)          289
Linksseitige Pr <= F              0.9797
Rechtsseitige Pr >= F             0.0368

Tabellenwahrscheinlichkeit (P)    0.0164
Zweiseitige Pr <= P               0.0735

        Stichprobengröße = 632
```

Output 2:

```
                Das SAS System          09:58 Monday, January 5, 2009  18

 Beob.      zvers

   1      1.91324
```

Den Abweichungen im Ergebnis liegt die Tatsache zugrunde, dass im Skript der Standardfehler einer Differenz mit der gemeinsamen (gepoolten) Schätzung von p unter H0 berechnet wird, während SAS die Schätzungen von p je Gruppe verwendet.

5.3.6 Vergleich von zwei Binomialwahrscheinlichkeiten- verbundene Stichprobe

Auch hier wenden wir die Prozedur **freq** an. Zur Unterscheidung von verbundenen Stichproben wird der McNemar Test verwendet. Die Anweisungen unterscheiden sich zu den bereits aufgeführten im table; Statement, hier wurde die Option agree statt chisq verwendet. Dieses Statment sorgt dafür, dass mehrere verschiedene Tests über unsere Daten laufen (siehe SAS Hilfe Stichwort: agree). Darunter ist auch der McNemar Test. Nach diesem Test wird eine so genannte S-Statistik erzeugt, aus der wir wieder in einem Datastep die Wurzel ziehen um zu unserem z_{Vers} zu kommen.

```
Data Labor;
input methodA$ methodB$ anzahl;
datalines;
j j 156
j n 32
n j 18
n n 37
;
run;
Proc freq data=Labor;
weight anzahl;
table methodA*methodB/agree;
run;
data z;
zvers=sqrt(3.92);
run;
proc print data=z;
run;
```

Output:

```
                Das SAS System          09:58 Monday, January 5, 2009  21
```

Die Prozedur FREQ

Table of methodA by methodB

methodA methodB

Häufigkeit Prozent Row Pct Col Pct	j	n	Summe
j	156	32	188
	64.20	13.17	77.37
	82.98	17.02	
	89.66	46.38	
n	18	37	55
	7.41	15.23	22.63
	32.73	67.27	
	10.34	53.62	
Summe	174	69	243
	71.60	28.40	100.00

Statistiken für Tabelle von methodA nach methodB.

Test von McNemar

Statistik (S)	**3.9200**
DF	1
Pr > S	0.0477

Einfacher Kappa-Koeffizient

Kappa	0.4610
ASE	0.0644
95% Untere Konf.grenze	0.3349
95% Obere Konf.grenze	0.5871

Stichprobengröße = 243

Das SAS System 09:58 Monday, January 5, 2009 22

Beob.	zvers
1	1.97990

Für die Beispiele in Kapitel 5.3.7 wird auf die SAS Anweisungen aus dem vorangehenden Kapitel verwiesen.

5.4 Poissonverteilung

5.4.2 Parameterschätzung oder Schätzen des Parameters der Poisson-Verteilung

S.143 Beispiel Leukozyten

Der geschätzte Parameter, den wir suchen, ist ein Mittelwert. Somit ist es nahe liegend die Prozedur **means** für seine Berechnung zu verwenden. Wir wollen den Mittelwert über die Zählwerte k mit der Gewichtung nach den beobachteten Häufigkeiten B_k. Dem entsprechen die Anweisungen im Procstep. Mit `var k;` bilden wir den Mittelwert über k und mit `weight beob;` gewichten wir nach B_k.

```
data LambdaLeuk; /* Mit proc means über Mittelwerte von k gewichtet nach Bk.*/
input k beob;
datalines;
0 158
1 73
2 21
3 3
4 1
;
run;
proc means data=LambdaLeuk;
var k;
weight beob;
run;
```

Output:

Die Prozedur MEANS

Analysis Variable : k

N	Mittelwert	Std. abweichung	Minimum	Maximum
5	0.5000000	5.8309519	0	4.0000000

Erwartete Häufigkeiten einer Poissonverteilung

S.144 Beispiel Leukozytenzählung

Hier können wir ganz genauso vorgehen wie unter 5.3 bei der Binomialverteilung. Mit folgenden Unterschieden: 1. Wir tauschen `'BINOM'` in `'POISSON'` weil wir jetzt mit der Poissonverteilung arbeiten und 2. programmieren wir keine Schleife mehr sondern berechnen jede Schätzung einzeln. Also erstellen wir folgende Datasteps:

```
data poissonLeukE4;
```

```
k=4;
w=PDF('POISSON',k,0.5);
ws=CDF('POISSON',k,0.5);
E4=w*256;
run;
Proc print data=poissonLeukE4;
run;

data poissonLeukE3;
k=3;
w=PDF('POISSON',k,0.5);
ws=CDF('POISSON',k,0.5);
E3=w*256;
run;
Proc print data=poissonLeukE3;
run;

data poissonLeukE2;
k=2;
w=PDF('POISSON',k,0.5);
ws=CDF('POISSON',k,0.5);
E2=w*256;
run;
Proc print data=poissonLeukE2;
run;

data poissonLeukE1;
k=1;
w=PDF('POISSON',k,0.5);
ws=CDF('POISSON',k,0.5);
E1=w*256;
run;
Proc print data=poissonLeukE1;
run;

data poissonLeukE0;
k=0;
w=PDF('POISSON',k,0.5);
ws=CDF('POISSON',k,0.5);
E0=w*256;
run;
Proc print data=poissonLeukE0;
run;
```

Outputs:

```
                    Das SAS System          09:58 Monday, January 5, 2009  34

Beob.     k        w          ws          E0
  1       0     0.60653    0.60653     155.272


Beob.     k        w          ws          E1
  1       1     0.30327    0.90980     77.6359


Beob.     k        w          ws          E2
  1       2    0.075816    0.98561     19.4090


Beob.     k        w          ws          E3
  1       3    0.012636    0.99825     3.23483
```

Beob.	k	w	ws	E4
1	4	.001579507	0.99983	0.40435

Wir haben nun die erwarteten Häufigkeiten berechnet. Es fehlen noch die Summe und E_{Rest}. Folgende Datasteps sind nötig:

```
data poissonLeukest;
n=256;
E0= 155.272;
E1= 77.6359;
E2= 19.4090;
Erest=n-E0-E1-E2;
run;
Proc print data=poissonLeukest;
run;


Data summe;
n=256;
E0= 155.272;
E1= 77.6359;
E2= 19.4090;
E3= 3.23483;
E4= 0.40435;
s=E0+E1+E2+E3+E4;
run;
proc print data=summe;
run;
```

Outputs:

Das SAS System 09:58 Monday, January 5, 2009 44

Beob.	n	E0	E1	E2	Erest
1	256	155.272	77.6359	19.409	3.6831

Beob.	n	E0	E1	E2	E3	E4	s
1	256	155.272	77.6359	19.409	3.23483	0.40435	255.956

Vertrauensintervall für Parameter λ einer Poisson-Verteilung

S.145 Beispiel Leukozytenzählung

Auch hier ist es wieder schwierig, mit einer Prozedur zu verfahren, da diese nur bei großen n genügend genau greifen. Deshalb werden wir alles zu Fuß berechnen. Die gesamten Variablen und Formeln werden in einen Datastep gepackt. Die Funktion PROBIT kennen wir schon um mit gegebener Wahrscheinlichkeit den entsprechenden Wert der z- Verteilung zu berechnen. Das Q bei lambdaUQ bzw. lambdaOQ soll einfach nur Qubik abkürzen um klarzustellen, dass es hier um den Vertrauensintervall für die 3,2mm³ geht.

```
data LambdaLeukCL;
n=256;
lambda=0.5;
alpha=0.05;
z=PROBIT(1-alpha/2);
lambdaU=1/n*((sqrt(n*lambda))-1/2*z)**2;
lambdaO=1/n*((sqrt(n*lambda+1))+1/2*z)**2;
lambdaUQ=lambdaU*256/3.2;
lambdaOQ=lambdaO*256/3.2;
run;
proc print data=LambdaLeukCL;
run;
```

Output:

```
                                Das SAS System          09:58 Monday, January 5, 2009  47

                                                                                 lambda     lambda
Beob.     n     lambda    alpha       z       lambdaU    lambdaO       UQ          OQ

  1      256     0.5      0.05    1.95996    0.41713    0.59461    33.3706     47.5691
```

S.145 Beispiel Kläranlage

Wieder geht alles zu Fuß:

```
data Wasserprobe;
n=5;
lambda=(2+0+1+5+2)/5;
alpha=0.05;
z=PROBIT(1-alpha/2);
lambdaU=1/n*((sqrt(n*lambda))-1/2*z)**2;
lambdaO=1/n*((sqrt(n*lambda+1))+1/2*z)**2;
run;
proc print data=Wasserprobe;
run;
```

Output:

```
                                Das SAS System          09:58 Monday, January 5, 2009  51

Beob.    n    lambda    alpha       z       lambdaU    lambdaO

  1      5       2      0.05    1.95996    0.95248    3.69217
```

Entsprechend wird das *Phleum pratense* Beispiel auf S.145 bearbeitet.

5.4.3 Test für den Vergleich zweier Parameter λ_1 und λ_2

Hier gelangen wir über einen Datastep zu Fuß zu einem p-Wert, der uns sagt ob die Saatgutpartien gleich sauber sind oder nicht. Wir rechnen zuerst einen z_{Vers}-Wert aus und wandeln diesen dann mit der PROBNORM Funktion in einen zweiseitigen p-Wert einer Normalverteilung um. Dies macht die vorletzte Zeile. Durch die Multiplikation mit 2 kommen wir zum zweiseitigen p-Wert.

```
data Saatgut;
l1=3.02;
l2=3.75;
```

```
n1=98;
n2=95;
zvers=abs(l1-l2)/sqrt(((n1*l1+n2*l2)/(n1+n2))*(1/n1+1/n2));
p=2*(1-PROBNORM(zvers));
run;
```

Output:

```
                                Das SAS System          09:58 Monday, January 5, 2009  52

                Beob.      l1       l2      n1      n2      zvers              p

                  1       3.02     3.75     98      95     2.75806       .005814523
```

Auch der p-Wert < 0.05 erfordert H_0 abzulehnen. Die Saatgutpartien sind nicht gleich sauber.

5.5 Der Chi²-Anpassungstest

S.154 Beispiel Leukozyten Daten

Für den Chi²-Anpassungstest-Test gibt es keine Prozedur. Jedoch können wir uns mit **`proc means`** und einigen Statements zu unserem gewünschten χ^2_{Vers} durcharbeiten. Zunächst stellen wir die beobachteten und die erwarteten Häufigkeiten gegenüber und berechnen für jedes Paar einen χ^2_{Vers}-Wert. Dies geschieht für jedes Datenpaar, wenn wir die Formel für χ^2_{Vers} in die Zeile über `datalines` schreiben. Mit **`proc print`** können wir uns diese Daten im Ausgabefenster ansehen. Sie entsprechen bis auf kleine Rundungsfehler den Werten im Skript. Mit **`proc means`** bilden wir nun über die Variable chi, die wir oben im Datensatz erzeugt haben, den Mittelwert und den Standardfehler (siehe Output). Mit den Statements darunter erzeugen wir eine Datei mit dem Namen chiq in der die Variable chi aller Datenpaare aufsummiert wird und die Anzahl der Beobachtungen/Datenpaare als Klassen formuliert werden. Mit dem darauf folgenden Datenschritt wollen wir nun noch einen p-Wert und den χ^2_{Tab} erzeugen. Mit `set chiQ;` holen wir die Datei, die wir oben mit `output out=chiq` erzeugt haben, in die neue Datei, die wir mit **`data`** `chiQ;` benannt haben. Für die Formeln müssen wir zunächst eine Variable für die Freiheitsgrade und eine für α definieren, was mit den Zeilen, `df=2;alpha=0.05;` (df steht für Degrees of Freedom= Freiheitsgrade) geschieht. Der p-Wert errechnet sich mit einer Funktion (probCHI), die uns eine χ^2-Verteilung liefert, aus der wir die Überschreitungswahrscheinlichkeit für unser chi bei 2 Freiheitsgraden berechnen können, sprich den p-Wert. Das gewünschte X^2_{Tab} erhalten wir mit der Funktion **cinv**. In die Klammer dahinter muss das Signifikanzniveau und die Freiheitsgrade.

```
data chiLeuk;
input B E;
chi=((B-E)**2)/E;
datalines;
158 155.3
73 77.6
21 19.4
4 3.6
;
run;
proc means data=chiLeuk;
var chi;
output out=chiq sum=chi n=klassen;
run;

data chiQ;
set chiQ;
df=2;
alpha=0.05;
p=1-probCHI(chi,df);
CHI_TAB=cinv(1-alpha,df);
run;

proc print data=chiQ;
run;
```

Output 1:

```
                          Das SAS System          09:58 Monday, January 5, 2009  78

                          Die Prozedur MEANS

                        Analysis Variable : chi

                                    Std.
N       Mittelwert      abweichung          Minimum          Maximum
-------------------------------------------------------------------
4        0.1240063       0.1071390        0.0444444        0.2726804
-------------------------------------------------------------------
```

Output 2:

```
                          Das SAS System          09:58 Monday, January 5, 2009  79

Beob.    _TYPE_    _FREQ_      chi      klassen    df    alpha       p        CHI_TAB

  1          0         4     0.49603        4        2     0.05    0.78035     5.99146
```

Hier ist nun alles schön übersichtlich auf einen Nenner gebracht.

Der p-Wert ist größer als 0.05 und $X^2_{Tab} > X^2_{Vers}$ demnach wird die Nullhypothese beibehalten und wir haben keine Poissonverteilung.

S. 156 Beispiel Erdnussfeld

Dieses Beispiel ist wohl das komplizierteste. Deswegen werde ich die einzelnen Zeilen mit einem SAS Kommentar erklären, damit jede Erklärung bei ihrer Zeile steht. Das zerpflückt das Programm etwas, vereinfacht aber die Orientierung und kann im Editor weggelassen werden.

Wir fassen das ganze Programm kurz in einem Überblick über die einzelnen Schritte, die uns zu unserem χ^2_{Vers} bringen sollen, zusammen. Grundlegend wollen wir wissen ob die von uns beobachteten Werte ungefähr (bis zu einem Signifikanzniveau von α=5%) mit einer Poissonverteilung übereinstimmen. Folgende Schritte sind in SAS dazu nötig:

1. Datenschritt für die beobachteten (observed) Daten
2. Erzeugen von erwarteten Daten, die eine Poissonverteilung aufweisen .
3. Gegenüberstellen der beiden Datensätze mit der `Merge`-Anweisung.
4. Berechnen der Wahrscheinlichkeiten beider Datensätze (Beobachtete und erwartete).
5. Chi²-Anpassungstest über alle Klassen und aufsummieren zu χ^2_{Vers}.
6. Berechnen des p-Wertes

Vorher noch zum Anschauen und auch noch als Überblick das Programm für dieses Beispiel ohne zeilenerklärende Kommentare.

```
/* S. 156 Beispiel Erdnüsse Inokulumbesatz Chi²-Anpassungstest */
%let n=26;
data erdnuss;
input k Bk;
class=k; if class <3 then class=0;
if class>11 then class=12;
/* Beobachtete Daten */
cards;
0 2
1 5
2 5
3 13
4 5
5 6
6 9
7 6
8 7
9 4
10 5
11 9
12 6
13 2
14 2
15 0
16 1
17 1
18 2
19 0
20 2
21 1
22 2
23 0
24 0
25 0
26 1
;
proc means data=erdnuss noprint;
freq Bk;
var k;
```

```
output out=erdn MEAN=mean;
/* Erzeugen der erwarteten Daten bei Annahme einer Poissonverteilung */
data Poisson;
set erdn;
lambda=mean;
k=0;
prob=EXP(-lambda);
RETAIN lambda pr_old cumul k;
cumul+prob;
pr_old=prob;
output;
Do k=1 to &n-1;
prob=lambda/k*pr_old;
pr_old=prob; /
cumul+prob;
output;
END;
prob=1-cumul;
output;
run;
proc means data=erdnuss noprint;
var Bk;
output out=sum_Bk SUM=sum;
/*Gegenüberstellen der beobachteten und erwarteten Daten und Berechnung der
Wahrscheinlichkeitswerte */
Data erdnussdaten;
Merge Poisson erdnuss;
if _n_=1 then set sum_Bk;
Retain sum;
expected=sum*prob;
observed=Bk;
Keep prob k expected observed Bk sum class;
proc means data=erdnussdaten noprint;
by class;
var observed expected;
output out=test sum=observed expected;
/* Chi² Anpassungstest */
data test;
set test;
chiSQvers=(observed-expected)**2/expected;
run;
/* Aus den 11 chi² Werten wird die Summe zu einem Chi²Vers*/
proc means data=test;
var chiSQvers;
output out=chiq sum=chiSQvers N=gruppen;
run;
/* Berechnung des p-Wertes */
data finale;
set chiq;
p_wert=1-PROBCHI(chiSQvers, gruppen-2);
proc print;
run;
```

Das Programm mit Erklärungen:

```
/* S. 156 Beispiel Erdnüsse Inokulumbesatz Chi²-Anpassungstest */
%let n=26; /*Die Anweisung %let n=26; ist eine so genannte globale Variable.
Global heisst, dass sie in allen Programmen die momentan im Editor stehen mit &n
aufgrufen bzw. eingesetzt werden kann und dann als Synonym für die Zahl hinter
dem = gilt. Wirklich sinnvoll ist dies bspw. für lange Dezimalzahlen.*/
```

```
data erdnuss;
input k Bk;
class=k; /* Wir definieren hier bzw. sagen SAS, dass die Variable class gleich der Variablen k ist. Diese Definition schränken wir in den nächsten zwei Anweisungen noch weiter ein.*/
if class <3 then class=0; /* Auf S.157 unten werden die vom Betrag her kleinen Werte in Klassen zusammengefasst. Das machen wir mit diesen beiden Anweisungen. SAS schaut dabei nicht die Beträge der Werte an, sondern wir sagen es SAS, weil wir die Werte kennen, die wir zu einer Klasse zusammengefasst haben wollen.*/
if class>11 then class=12;
/* Beobachtete Daten */
cards;
0 2
1 5
.
.
.
24 0
25 0
26 1
;
/* Für die erwarteten Werte, die wir später erstellen wollen, brauchen wir die Varianz bzw. den Mittelwert (ist gleichbedeutend, weil diese bei der Poissonverteilung gleich sind (und =lambda)) der beobachteten Daten. Dies bewerkstelligen in diesem Procstep mit proc means. */
proc means data=erdnuss noprint; /* Die Option noprint verhindert, dass für die in diesem Procstep berechneten Daten ein Tabelle gespeichert wird.*/
freq Bk; /* Mit var sagen wir proc means nach welcher Variablen es analysieren soll. Mit freq sagen wir means welchen Wert diese Variable in jeder Beobachtung jeweils hat. Wörtlich übersetzt ist Bk die Frequenz (Anzahl an Beobachtungen) von k.*/
var k;
output out=erdn MEAN=mean; /* Mit output out erzeugen wir eine neue Datei mit dem Namen erdn. In dieser soll der Mittelwert (MEAN) der Daten mean heißen.Wir benennen ihn, damit wir wissen unter welchem Namen wir ihn später wieder einsetzen können.*/
/* Erzeugen der erwarteten Daten bei Annahme einer Poissonverteilung */
data Poisson; /* Name der Datei für die erwarteten Häufigkeiten. */
set erdn; /* Mit der Funktion set holen wir die Daten aus der Datei erdn weil wir sie brauchen (MIttelwert bzw. Varianz) um die Daten Poisson zu erstellen.*/
lambda=mean; /* Lambda in den poissonverteilten Daten soll gliech dem Mittelwert/Varianz mean aus den erdn Daten sein. Oben hatten wir schon erwähnt, dass idese gleich sind. */
k=0; /* Für die Berechnung der erwarteten Häufigkeiten brauchen wir einen Anfangswert E0, der bei k=0 ist. Der Kasten auf S.143 im Skript stellt dies ebenso dar. Also in dieser Zeile defnieiren wir ersteinmal nur dass wir bei k=0 sind bzw. mit der Berechnung anfangen.*/
prob=EXP(-lambda); /*Formel für E0 aus dem Kasten auf S.143.*/
```

```
RETAIN lambda pr_old cumul k; /* Für die weiteren Berechnungen brauchen wir die Variablen lambda, pr_old, cumul und k. cumul und prob_old werden in den nächsten Zeilen definiert. Mit RETAIN sorgen wir dafür, dass diese Variablen in allen weiteren Data- und Procsteps mit ihrem Wert aufrufbar sind.*/
cumul+prob; /* Definitin der Variablen cumul. cumul is hier noch =0 und addiert von k zu k+1 immer die Wahrscheinlichkeiten bis wir am Ende den Chi²Vers haben. Weil cumul =0 ist und immer nur prob addiert wird kann man anstatt cumul=cumul+prob nur cumul+prob anweisen. */
pr_old=prob; /* Das oben mit EXP(-lambda) definierte prob, soll ab hier prob_old heisen.*/
output;     /* Alles was in diesem Datastep bis hier her defniniert wurde, soll sich das System "merken" aber keine Datei dafür abspeichern. Dies geschieht mit output. output out würde alle Variablen in einer Tabelle im Work Ordner speichern.*/
Do k=1 to &n-1; /* Nun leiten wir ein Schleife ein, die uns von allen k nacheinander bis k=25 einen Erwartungswert E berechnet. Ohne diese Schleife müsste man jedes k mit einer extra Zeile berechnen. Der Erwartungswert für k 26 nämlich E26 ist der Wert E_Rest. Er wird zum Schluss gesondert berechnet. Siehe S.144 im Skript: Beispiel Leukozytenzählung. */
prob=lambda/k*pr_old; /* Die Wahrscheinlichkeit für die Erwartungswerte sollen nach der Rekursionsformel S.143 Kasten (2) berechnet werden. Diese Formel tragen wir hier ein. Für den Wert den wir für Ex berechnen steht die Variable prob. */
pr_old=prob; /* Für jeden E-Wert der in der Schleife berechnet wird braucht die Formel den Wert von Ek-1. Jedes prob_old in der Formel des jeweiligen E ist defniert durch das prob aus der Berechnung des Ek-1ten Wertes. Damit dies in unserem Programm hier auch passiert brauchen wir diese Zeile.*/
cumul+prob; /*Diese Anweisung sagt dass jeder neue E-Wert zur Summe aus den schon berechneten E-Werten dazu addiert werden soll.*/
output;
END; /*Ende der Schleife die die Erwartungswerte berechnet.*/
prob=1-cumul; /* Berechnung des oben erwähnten ERest.*/
output;
run;
/* Berechnung der Wahrscheinlichkeiten für die beobachteten Werte Bk. Um die Wahrscheinlichkeit eines k zu ermitteln, muss sein Wert durch die Summe aller Werte geteilt werden. Diese Summe errechnet uns der folgende Procstep mit proc means.*/
proc means data=erdnuss noprint;
var Bk;
output out=sum_Bk SUM=sum; /*Ausgabe der Summe aller beobachteten Werte unter der Bezeichnung sum.*/

/*Gegenüberstellen der beobachteten und erwarteten Daten. In diesem Datenschritt stellen wir die Daten gegenüber und berechnen die Wahrscheinlichkeiten der erwarteten Daten. */
```

```
Data erdnussdaten;
Merge Poisson erdnuss; /* Mit dem Statement merge stellen wir zwei Datensätze nebeneinander. Zur Veranschaulichung kannst du, nachdem du diesen Datensatz berechnet hast, im Work Ordner unter dem Namen erdnussdaten anschauen. Die Daten aus Poisson und erdnuss stehen nebeneinander.*/
if _n_=1 then set sum_Bk; /* Hier setzen wir in jede unserer 26 Zeilen im Datensatz die Variable sum_Bk ein(die Summe aller beobachteten Werte). Mit dem if-then-Statement sagen wir: Wenn _n_ (Zeilenangabe im Datensatz) 1 ist/ wir in der ersten Zeile stehen soll die Summe der beobachteten Werte eingetragen werden. Um die Wahrscheinlichkeiten der erwarteten Werte zu berechnen, brauchen wir sum_Bk in jeder Zeile. Das erreichen wir mit RETAIN sum.*/
Retain sum;
expected=sum*prob; /* Die Wahrscheinlichkeiten der ewarteten Werte heißen expected und errechnen sich aus sum*prob. prob war als Wahrscheinlichkeit der Bk definiert(von jedem n einzeln).*/
observed=Bk; /* Bis hier her haben wir jetzt die Wahrscheinlichkeiten der beobachteten und der erwarteten absoluten Werte berechnet.*/
Keep prob k expected observed Bk sum class;    /* Das KEEP Statement sagt SAS welche Variablen im Output stehen sollen. Die Tabelle mit diesen Variablen eingetragen kann man sich nun entweder unter Explorer->Bibliotheken->Work->erdnussdaten ansehen oder sie sich mit Proc print im Output- Fenster ausgeben lassen.*/
proc means data=erdnussdaten noprint; /* Wir hatten unseren Ursprungsdatensatz in Klassen eingeteilt. In diesem Procstep mit proc means berechnen wir nun Summe, Mittelwert und Varianz pro Klasse bzw. für die 11 Klassen in den entscheidenden Variablen expected und observed, weil wir mit diesen den Chi²-Anpassungstest durchführen.*/
by class;
var observed expected;
output out=test sum=observed expected; /* Die neue Datei heißt test, weil wir ihre Daten für den Chi²-Anpassungstest brauchen. Die Summe von observed und expected soll auch mit ausgegeben werden.*/
 /* Jetzt kommt der chi² Anpassungstest mit dem wir testen ob die beobachteten Daten einer Poisson Verteilung à la den erwarteten Daten entsprechen. */
data test;
set test; /* Einsetzen der Daten mit set aus den schon vorhandenen Daten test.*/
chiSQvers=(observed-expected)**2/expected; /* Was soll berechnet werden? Die Formel für Chi²Vers aus dem Kasten auf S.154.*/
run; /* Bisher haben wir den Chi²Vers für jede Beobachtung einzeln errechnet. Mit dem folgenden Procstep bilden wir die Summe zum endgültigen Chi²Vers, das mit dem Tabellenwertverglichen werden kann.*/
/* Aus den 11 chi² Werten wird die Summe zu einem Chi²Vers*/
proc means data=test;
var chiSQvers; /* Diese Variable haben wir im Datastep test erstellt und über sie soll proc means angewendet werden.*/
```

```
output out=chiq sum=chiSQvers N=gruppen; /* Name der neuen Datei mit output out ist: chiq. Wir wollen die Summe der Variablen chiSQvers und die Anzahl aller Beobachtungen(=N) als Gruppen bezeichnet haben. */
run;
/* Berechnung des p-Wertes */
data finale;
set chiq; /* Daten aus der Datei chiq sollen angewandt werden */
p_wert=1-PROBCHI(chiSQvers, gruppen-2);  /* Formel für den p-Wert. Es geht um eine Fläche unter der Chi²-Wahrscheinlichkeitskurve links und rechts des positiven und negativen Signifikanzniveaus. Die Gesamtfläche darunter hat den Betrag 1. Die Funktion PROBCHI berechnet uns, mit den Vorraussetzungen in Klammern (zu betrachtende Variable, Freiheitsgrade), diese Fläche. Diese Fläche von 1 abgezogen gibt uns den p-Wert wieder. Die Wahrscheinlichkeit, mit der die Nullhypothese fälschlicherweise verworfen wird obwohl sie zutrifft.*/
proc print;
run;  /*  Das  ganze  laufen  lassen  und  es  sich  schön,  übersichtlich, zusammengefasst im Output-Fenster anschauen und interpretieren. Die beobachteten Daten entsprechen nicht einer Poissonverteilung.*/
```

Output:

```
                          Das SAS System        19:19 Saturday, January 10, 2009  12

                          Die Prozedur MEANS

                    Analysis Variable : chiSQvers

                                    Std.
 N      Mittelwert      abweichung          Minimum          Maximum
-------------------------------------------------------------------
11      13.4813793      26.2150629       0.0498691       85.5076685
-------------------------------------------------------------------

                          Das SAS System          01:09 Sunday, January 11, 2009   4

                                                chi
    Beob.     _TYPE_     _FREQ_              SQvers     gruppen     p_wert

      1          0         11               148.295        11          0
```

5.6 Test auf Unabhängigkeit in der 2x2 Feldertafel (4- Feldertafel)

Die Beispiele, die in den folgenden drei Unterpunkten von 5.6 behandelt werden, rechnen wir alle mit demselben Prozedurschritt. Da im Output dieses Prozedurschrittes die Angaben für den χ^2_{Vers} für große Stichproben, die Yateskorrektur/ Kontinuitätskorrektur sowie für Fisher's exakten Test enthalten sind.

5.6.1 Test bei großen Stichproben

S. 167 Beispiel Impfung gegen Typhus

Im Datensatz für die Impfung geben wir als Variablen die Impfung selbst (geimpft oder nicht geimpft), den Gesundheitszustand und die Anzahl der Fälle einer Kombination. Im `tables` Statement geben wir wieder die Variablen an die wir verglichen haben wollen. Dahinter müssen wir, um eine Analyse zu erhalten, nach dem / noch `chisq;` angeben. Damit liefert uns SAS alle Informationen die wir in diesem Kapitel wissen müssen. An dieser Stelle lassen sich noch weitere Optionen eingeben; siehe HILFE (F1). Mit der Variablen hinter `weight` geben wir einmal an nach welcher Variablen gewichtet werden soll, d.h. in welcher Variable die Häufigkeiten stehen. Ohne die `weight` Anweisung müssten wir einen Datensatz mit 18483 Kombinationen eingeben. Data- und Procstep sehen demnach so aus:

```
data typhus;
input Impfung$ GesZustand$ Anzahl;
cards;
geimpft krank 56
geimpft gesund 6759
Ngeimpft krank 272
Ngeimpft gesund 11396
;
run;
proc freq data=typhus;
tables Impfung*GesZustand/chisq;
weight Anzahl;
run;
```

Output:

```
Das SAS System        16:03 Wednesday, January 7, 2009   8

                          Die Prozedur FREQ

                   Table of Impfung by GesZustand

Impfung     GesZustand

Häufigkeit|
Prozent   |
Row Pct   |
Col Pct   |gesund   |krank    |  Summe
----------|---------|---------|---------
Ngeimpft  |   11396 |     272 |   11668
          |   61.66 |    1.47 |   63.13
          |   97.67 |    2.33 |
          |   62.77 |   82.93 |
----------|---------|---------|---------
geimpft   |    6759 |      56 |    6815
          |   36.57 |    0.30 |   36.87
          |   99.18 |    0.82 |
          |   37.23 |   17.07 |
----------|---------|---------|---------
Summe     |   18155 |     328 |   18483
          |   98.23 |    1.77 |  100.00
```

Statistiken für Tabelle von Impfung nach GesZustand.

Statistik	DF	Wert	Prob
Chi-Quadrat	**1**	**56.2341**	**<.0001**
Likelihood-Quot. Chi-Quad.	1	63.1614	<.0001
Kontinuitätskorr. Chi-Quad.	1	55.3714	<.0001
Mantel-Haenszel Chi-Quadrat	1	56.2310	<.0001
Phi-Koeffizient		-0.0552	
Kontingenzkoeffizient		0.0551	
Cramers V		-0.0552	

Exakter Test von Fisher

Zelle (1,1) Häufigkeit (F)	11396
Linksseitige Pr <= F	1.723E-15
Rechtsseitige Pr >= F	1.0000
Tabellenwahrscheinlichkeit (P)	1.134E-15
Zweiseitige Pr <= P	2.526E-15

Stichprobengröße = 18483

Wir erhalten denselben Wert wie im Skript, der eine Abhängigkeit zwischen Impfung und Erkrankung bestätigt.

5.6.2 Die Kontinuitätskorrektur von Yates

S.169 Beispiel Primula

Auch diese ist im Output direkt enthalten. Somit unterscheiden sich Data- und Procstep nur in den Daten und den Namen der Variablen. Also:

```
Data primula;
input Wasser$ Keimung$ Faelle;
cards;
LehmH2O Ngekeimt 5
LehmH2O gekeimt 9
RegenH2O Ngekeimt 15
RegenH2O gekeimt 3
;
run;
proc freq data=primula;
tables Wasser*Keimung/chisq;
weight Faelle;
run;
```

Output:

Das SAS System 16:03 Wednesday, January 7, 2009 9

Die Prozedur FREQ

Table of Wasser by Keimung

Wasser Keimung

Häufigkeit Prozent Row Pct Col Pct	Ngekeimt	gekeimt	Summe
LehmH2O	5	9	14
	15.63	28.13	43.75
	35.71	64.29	
	25.00	75.00	
RegenH2O	15	3	18
	46.88	9.38	56.25
	83.33	16.67	
	75.00	25.00	
Summe	20	12	32
	62.50	37.50	100.00

Statistiken für Tabelle von Wasser nach Keimung.

Statistik	DF	Wert	Prob
Chi-Quadrat	1	7.6190	0.0058
Likelihood-Quot. Chi-Quad.	1	7.8707	0.0050
Kontinuitätskorr. Chi-Quad.	**1**	**5.7228**	0.0167
Mantel-Haenszel Chi-Quadrat	1	7.3810	0.0066
Phi-Koeffizient		-0.4880	
Kontingenzkoeffizient		0.4385	
Cramers V		-0.4880	

Exakter Test von Fisher

Zelle (1,1) Häufigkeit (F)	5
Linksseitige Pr <= F	0.0079
Rechtsseitige Pr >= F	0.9993
Tabellenwahrscheinlichkeit (P)	0.0072
Zweiseitige Pr <= P	0.0100

Stichprobengröße = 32

5.6.3 Fisher's exakter Test

S.169 Beispiel Klinik

Auch dieser ist im Output direkt enthalten. Somit unterscheiden sich Data- und Procstep nur in den Daten und den Namen der Variablen. Also:

```
Data Klinik;
input Beh$ Erfolg$ Faelle;
cards;
X ja 4
X nein 1
Y ja 3
Y nein 7
;
run;
proc freq data=Klinik;
tables Beh*Erfolg/chisq;
weight Faelle;
```

```
run;
```

Output:

```
Das SAS System        16:03 Wednesday, January 7, 2009  10

                        Die Prozedur FREQ

                      Table of Beh by Erfolg

Beh        Erfolg

Häufigkeit
Prozent
Row Pct
Col Pct    |ja       |nein     |  Summe
-----------+---------+---------+---------
X          |      4  |      1  |      5
           |  26.67  |   6.67  |  33.33
           |  80.00  |  20.00  |
           |  57.14  |  12.50  |
-----------+---------+---------+---------
Y          |      3  |      7  |     10
           |  20.00  |  46.67  |  66.67
           |  30.00  |  70.00  |
           |  42.86  |  87.50  |
-----------+---------+---------+---------
Summe      |      7  |      8  |     15
           |  46.67  |  53.33  | 100.00

          Statistiken für Tabelle von Beh nach Erfolg.

Statistik                       DF        Wert      Prob
-------------------------------------------------------
Chi-Quadrat                      1      3.3482    0.0673
Likelihood-Quot. Chi-Quad.       1      3.5064    0.0611
Kontinuitätskorr. Chi-Quad.      1      1.6406    0.2002
Mantel-Haenszel Chi-Quadrat      1      3.1250    0.0771
Phi-Koeffizient                         0.4725
Kontingenzkoeffizient                   0.4272
Cramers V                               0.4725

          WARNUNG: 75% der Zellen haben erwartete
     Häufigkeiten unter 5. Chi-Quadrat ist eventuell kein gültiger Test.

                 Exakter Test von Fisher
-------------------------------------------------
Zelle (1,1) Häufigkeit (F)              4
Linksseitige Pr <= F               0.9930
Rechtsseitige Pr >= F              0.1002

Tabellenwahrscheinlichkeit (P)     0.0932
Zweiseitige Pr <= P                0.1189

                 Stichprobengröße = 15
```

Der **Zweiseitige Pr <= P 0.1189** im Output entspricht dem p_{Vers} auf S. 171 und die **Tabellenwahrscheinlichkeit (P) 0.0932** dem P(Daten) auf S.170.

5.7 Test auf Unabhängigkeit in einer r x c Tafel

Für die r x c Tafel müssen wir ein paar Modifikationen vornehmen, um das aus dem Program zu bekommen, was wir wollen. Wir haben eine 3x3 Kontingenztafel, in der wir die drei Bodentypen mit den drei Besitzformen analysieren wollen. Mit den beiden Schleifen sorgen wir nun dafür, dass die 9 Vergleiche von statten gehen und jedes mit jedem verglichen wird. Die ersten beiden Anweisungen (`length bodentyp $16;`) im Datastep definieren lediglich die Anzahl der Buchstaben, die die Variablen bodentyp und besitzform haben dürfen. Wenn wir nichts angeben definiert SAS die Länge der Variablen nach der ersten Variablen. Also ein Buchstabe beim Bodentyp, weil die erste Variable „I“ ein Buchstabe ist. Bei Besitzform definiert SAS 11 Buchstaben weil „Eigenbesitz“ 11 Buchstaben hat. Demnach müssten wir die Länge der Variablen Besitzform nicht definieren, weil die erste Form das längste Wort ist und so die anderen beiden vollständig eingelesen werden. Zum Problem würde es ohne Längendefinition bei der ersten Variable werden, weil die erste Form die kürzeste ist. Also liest SAS bei Besitzform II und III jeweils auch nur I ein, weil die weiteren Ziffern nicht definiert sind. Willkürlich definieren wir die Länge auf 16 Ziffern obwohl bei der ersten Variablen drei und bei der zweiten 11 ausreichend wären. Um konkret zu sehen, was ohne diese Anweisungen passiert, kann man das Programm einfach mal ohne sie laufen lassen und sich den Output anschauen. Für unsere beiden Variablen erstellen wir nun zwei so genannte `DO END`;-Schleifen. Die erste Schleife liest für den Bodentyp I die Daten für jede Besitzform aus der zweiten Schleife ein. Dann für Bodentyp II jede Besitzform und als letztes für Bodentyp III. Da SAS zeilenorientiert ist, belegt es die Variablen Zeile für Zeile. Bodentyp I erste Zeile usw. Und dann bekommt jede Besitzform eine Zahl der Zeile. Hinter `input;` definieren wir die Zahlen des Datensatzes als die Variable Anzahl. Die beiden @@ sagen SAS, dass in einer Zeile für eine Variable Bodentyp mehrere Einträge stehen. Das `Output`;-Statement sorgt dafür, dass diese Zeilen in der Datei gespeichert werden. Die beiden `END;END;` Anweisungen beenden die Schleifen. Nun kommt ein gewohnter Procstep wie wir ihn aus den vorigen Beispielen kennen.

```
data Iowa;
length bodentyp $16;
length besitzform $16;
DO Bodentyp= 'I', 'II', 'III';
DO Besitzform= 'Eigenbesitz', 'Pacht', 'Gemischt';
input Anzahl@@;
Output;
END;
END;
```

```
cards;
36 67 49
31 60 49
58 87 80
;
run;
proc print;run;
proc freq data=Iowa order=data;
tables Bodentyp*Besitzform/ chisq;
weight anzahl;
run;
```

Output:

Beob.	bodentyp	besitzform	Anzahl
1	I	Eigenbesitz	36
2	I	Pacht	67
3	I	Gemischt	49
4	II	Eigenbesitz	31
5	II	Pacht	60
6	II	Gemischt	49
7	III	Eigenbesitz	58
8	III	Pacht	87
9	III	Gemischt	80

Die Prozedur FREQ

Table of bodentyp by besitzform

bodentyp besitzform

Häufigkeit Prozent Row Pct Col Pct	Eigenbesitz	Pacht	Gemischt	Summe
I	36	67	49	152
	6.96	12.96	9.48	29.40
	23.68	44.08	32.24	
	28.80	31.31	27.53	
II	31	60	49	140
	6.00	11.61	9.48	27.08
	22.14	42.86	35.00	
	24.80	28.04	27.53	
III	58	87	80	225
	11.22	16.83	15.47	43.52
	25.78	38.67	35.56	
	46.40	40.65	44.94	
Summe	125	214	178	517
	24.18	41.39	34.43	100.00

Statistiken für Tabelle von bodentyp nach besitzform.

Statistik	DF	Wert	Prob
Chi-Quadrat	4	1.5431	0.8190
Likelihood-Quot. Chi-Quad.	4	1.5517	0.8175
Mantel-Haenszel Chi-Quadrat	1	0.0109	0.9169
Phi-Koeffizient		0.0546	
Kontingenzkoeffizient		0.0546	
Cramers V		0.0386	

Stichprobengröße = 517

Verglichen mit χ^2_{Tab} wird die Nullhypothese der Unabhängigkeit nicht verworfen. Es können also keine Abhängigkeiten nachgewiesen werden.

Alternativ können die Daten auch wie folgt eingelesen werden:+

```
data Iowa;
input bodentyp$    besitzform$     Anzahl;
datalines;
  I          Eigenbesitz       36
  I          Pacht             67
  I          Gemischt          49
  II         Eigenbesitz       31
  II         Pacht             60
  II         Gemischt          49
  III        Eigenbesitz       58
  III        Pacht             87
  III        Gemischt          80
;
```

6. Korrelation und Regression

6.1 Die Pearsonsche Produkt- Moment Korrelation

S. 183 Beispiel 3

Die Niederschlagsdaten von 1900-1925 sind in üblicher Form im folgenden Datastep abgetragen.

```
Data Regen;
input jahr ns ertrag;
datalines;
1900 177 26.2
1901 96 25.0
1902 144 32.6
1903 105 26.6
1904 111 19.6
1905 135 20.4
1906 209 29.2
1907 161 33.8
1908 246 26.6
1909 108 22.6
1910 137 24.2
1911 71 16.6
1912 119 29.8
1913 108 19.4
1914 132 30.6
```

```
1915 89 16.4
1916 147 30.4
1917 98 19.2
1918 106 20.2
1919 123 25.6
1920 156 31.0
1921 191 35.8
1922 162 31.6
1923 235 33.6
1924 147 30.4
1925 110 30.2
;
run;
```

S. 188 Beispiel Schätzen des Korrelationskoeffizienten

Eine Korrelation lässt sich in SAS mit der Prozedur **corr** bewerkstelligen. Dabei müssen nur die Variablen angegeben werden die miteinander korreliert werden sollen. Der Procstep kann auch noch ergänzt werden indem man hinter **proc corr** noch pearson einträgt. Es macht jedoch keinen Unterschied weil die Pearsonsche Korrelation Default-Einstellung ist.

```
proc corr data=Regen;
var ns ertrag;
run;
```

Folgender Output entsteht dabei:

```
                                  Das SAS System       16:05 Saturday, November 29, 2008  25

                                  Die Prozedur CORR

                              2  Variablen:    ns       ertrag


                                  Einfache Statistiken

                                                    Std.
Variable            N     Mittelwert     abweichung         Summe       Minimum       Maximum

ns                 26      139.34615       43.97085          3623       71.00000      246.00000
ertrag             26       26.44615        5.67304      687.60000       16.40000       35.80000


                  Pearsonsche Korrelationskoeffizienten, N = 26
                            Prob > |r| under H0: Rho=0

                                        ns          ertrag

                     ns           1.00000         0.62913
                                                  0.0006

                     ertrag       0.62913         1.00000
                                  0.0006
```

Das Ergebnis entspricht dem im Skript mit r= 0,63.

S. 189 Test der Korrelation ρ und ihr Vertrauensintervall (S. 192)

Diese beiden Anwendungen können wir in einem Procstep erledigen. Die Korrelation lässt sich mit der Fisher- Option testen und mit der Anweisung `type=twosided` erhalten wir die obere und untere Grenze des Vertrauensintervalls für die Korrelation ρ. `BIASADJ=NO` ist ein Korrekturfaktor, mit dem SAS eine Verzerrung der Daten mit einberechnet. Wir brauchen diesen Faktor nicht also `BIASADJ=NO`. Das Signifikanzniveau α= 0,05 (Grundeinstellung; keine Angabe im Procstep nötig).

Der Procstep:

```
proc corr data=Regen fisher (type=twosided BIASADJ=NO);
var ns ertrag;
run;
```

Output:

Das SAS System 14:09 Tuesday, December 2, 2008 44

Die Prozedur CORR

2 Variablen: ns ertrag

Einfache Statistiken

Variable	N	Mittelwert	Std. abweichung	Summe	Minimum	Maximum
ns	26	139.34615	43.97085	3623	71.00000	246.00000
ertrag	26	26.44615	5.67304	687.60000	16.40000	35.80000

Pearsonsche Korrelationskoeffizienten, N = 26
Prob > |r| under H0: Rho=0

	ns	ertrag
ns	1.00000	**0.62913** 0.0006
ertrag	**0.62913** 0.0006	1.00000

Pearson Korrelationsstatistiken (Fisher-Z-Transformation)

Variable	Mit Variable	N	**Stichprobenkorrelation**	Fisher's z	**95% Konfidenzgrenzen**	
ns	ertrag	26	**0.62913**	0.73997	**0.319683**	**0.817308**

Pearson Korrelationsstatistiken (Fisher-Z-Transformation)

Variable	Mit Variable	**p-Wert für H0:Rho=0**
ns	ertrag	**0.0004**

Die Korrelation ist mit einem p-Wert von **0.0004** < 0,05 und somit signifikant. Auch die Grenzen des Vertrauensintervalls entsprechen dem Ergebnis aus dem Skript.

6.2.1 Regression und Streuungszerlegung

S.196 Schätzung der Regressionsgeraden für Regendaten

SAS verfügt auch über eine Prozedur, mit der sich die Parameter für eine Regressionsgerade schätzen lassen. Diese liefert zusätzlich noch das Bestimmtheitsmaß und eine Streuungszerlegung in einer Varianzanalysetabelle.
Der Datastep bleibt der Gleiche, deswegen hier nur der Procstep:
Bei der Modell- Anweisung ist wieder darauf zu achten, dass die abhängige Variable (Zielvariable; hier: Ertrag) vor dem Gleichheitszeichen steht und die unabhängige Variable (Prädiktorvariable) dahinter.

```
proc reg data=Regen;
model ertrag=ns;
run;
```

```
                                Das SAS System       16:05 Saturday, November 29, 2008  27

                                 Die Prozedur REG
                                  Model: MODEL1
                            Dependent Variable: ertrag

                          Number of Observations Read          26
                          Number of Observations Used          26


                                    Varianzanalyse

                                             Summe der      Mittleres
Quelle                          DF            Quadrate       Quadrat    F-Statistik    Pr > F

Model                            1           318.45795     318.45795          15.72    0.0006
Error                           24           486.12666      20.25528
Korrigierte Summe               25           804.58462


          Root MSE                    4.50059    R-Quadrat       0.3958
          Abhängiger Mittelwert      26.44615    Korr. R-Qu.     0.3706
          Coeff Var                  17.01792


                                Parameter Estimates

                                 Parameter-      Standard-
Variable          DF              schätzer         fehler     t-Wert    Pr > |t|

Intercept          1              15.13554        2.98596       5.07      <.0001
ns                 1               0.08117        0.02047       3.97      0.0006
```

„ns“ entspricht der Steigung b im Skript. Intercept ist der gesuchte Achsenabschnitt. R^2 ist das Bestimmtheitsmaß B. Der ungefähr gleiche F-Wert (=**15.72**) wie im Skript (Rundungsfehler) und der entsprechende p- Wert (=**0.0006**) zeigen eine signifikant von 0 verschiedene Steigung.

6.2.2 t-Tests und Vertrauensintervall

S.203 Test des Regressionskoeffizienten und sein Vertrauensintervall

Auch hier können beide Anwendungen wieder in einem Abwasch gemacht werden. Der Test für den Regressionskoeffizienten und sein Vertrauensintervall werden mit folgendem Procstep in der Prozedur GLM (! nicht REG) ausgegeben.

```
proc glm data=Regen;
model ertrag=ns/clparm;
run;
```

Da mit der Prozedur GLM auch Regressionen berechnet werden, können wir diese hier einsetzen um den Regressionskoeffizienten zu testen. Da wir den Datensatz hier nach keiner Variablen klassifiziert haben wollen, sondern über den gesamten Datensatz testen, benötigen wir keine Angabe der class- Variablen. `clparm;` ist eine Option die SAS sagt, dass ein Vertrauensintervall (cl=confidence limit) für den Parameter (parm) erstellt werden soll. Im Model ist wieder die Aufstellung: Die abhängige Variable des Ertrags wird mit der unabhängigen des Niederschlags modelliert.

Im Output erscheint:

```
  Das SAS System          16:24 Friday, December 5, 2008   1

                                  Die Prozedur GLM

                          Anzahl gelesene Beobachtungen           26
                          Anzahl verwendete Beobachtungen         26


                                  Die Prozedur GLM

Abhängige Variable: ertrag

                                        Summe der       Mittleres
       Quelle                     DF       Quadrate        Quadrat    F-Statistik    Pr > F

       Modell                      1     318.4579524     318.4579524          15.72    0.0006
       Fehler                     24     486.1266630      20.2552776
       Korrigierte Summe          25     804.5846154


                    R-Quadrat     Koeff.var    Wurzel MSE    ertrag Mittelwert

                     0.395804      17.01792      4.500586             26.44615
```

Quelle	DF	Typ I SS	Mittleres Quadrat	F-Statistik	Pr > F
ns	1	318.4579524	318.4579524	15.72	0.0006

Quelle	DF	Typ III SS	Mittleres Quadrat	F-Statistik	Pr > F
ns	1	318.4579524	318.4579524	15.72	0.0006

Parameter	Schätzwert	Standardfehler	t-Wert	Pr > |t|	95% Konfidenzgrenzen	
Konstante	15.13553937	2.98595716	5.07	<.0001	8.97282667	21.29825207
ns	0.08116919	0.02047077	3.97	0.0006	0.03891959	0.12341879

Alles was wir suchen steht wunderschön in einer Zeile. Der t-Wert entspricht dem im Skript, der p-Wert sagt uns ebenfalls, dass die Steigung signifikant von Null verschieden ist (weil p<0,05) und die Vertrauensgrenzen sind auch angegeben.