„Auswertung landwirtschaftlicher Experimente mit SAS“

Eine Einführung...

Dr. Andreas Büchse
Fachgebiet Bioinformatik
Universität Hohenheim
buechse@uni-hohenheim.de

Stand 19.Juli 2006

Vorwort

SAS ist ein Statistikpaket, welches eine zeilenorientierte Eingabe erfordert. Zwar ist in den letzten Jahren auch eine graphische Benutzeroberfläche entwickelt worden, diese bringt nach meiner persönlichen Einschätzung jedoch keine Vorteile für den wissenschaftlichen Nutzer und wird deshalb in diesem Skript nicht behandelt.

Verschiedene statistische Verfahren sind in sog. Prozeduren (PROC) verfügbar. Diese Prozeduren haben eine Reihe von voreingestellten Parametern, die es erlauben, schon mit wenigen grundlegenden Befehlen eine Datenanalyse durchzuführen. Oftmals muss aber von diesen Voreinstellungen abgewichen werden oder es ist sinnvoll, bestimmte Optionen zu nutzen. Dem Einsteiger durch den Dschungel der Möglichkeiten von SAS zu führen und Orientierung zu geben, dass ist das wesentliche Ziel dieses Skripts.

Das Skript soll und kann dagegen keine Lehrbücher und/oder Handbücher ersetzen. Es wird deshalb keinerlei Anspruch auf Vollständigkeit erhoben. Vielmehr werden ausgewählte Prozeduren und Methoden vorgestellt, die sich in meiner mehrjährigen „Zusammen"-Arbeit mit dem SAS-System als nützlich und notwendig erwiesen haben. In Ergänzung zu diesem Skript, sollte immer auch ein Handbuch zu Rate gezogen werden. Neben den Original-Handbüchern von SAS sei für den Einsteiger empfohlen:

- Dufner, Jensen, Schumacher (2002): Statistik mit SAS. 2. Auflage, Verlag Teubner
- Delwiche & Slaughter (2000): The Little SAS Book. SAS-Publishing, ISBN1-58025-239-7 (zu beziehen direkt über SAS Deutschland in Heidelberg (www.sas.com)

Da viele landwirtschaftliche Experimente mit Hilfe Gemischter Modelle auszuwerten sind, wird das Buch „SAS System for Mixed Models" (LITTELL et al. 1996) besonders empfohlen.

Sehr hilfreich sind verschiedene SAS-Newsgroups und e-learning-Foren im Internet. Eine Suche nach dem Stichwort „SAS" und einem weiteren das eigene Problem beschreibenden Schlagwort hilft oftmals bereits weiter. Einige SAS-Links finden sich auf der Homepage des Fachgebietes Bioinformatik (www.uni-hohenheim.de/bioinformatik) sowie unter folgenden Adressen:

- http://home.nc.rr.com/schabenb/
- http://core.ecu.edu/psyc/wuenschk/SAS/SAS-Programs.htm
- http://www.ats.ucla.edu/stat/sas
- SAS-Anwenderhandbuch im Netz vom Rechenzentrum der Uni Heidelberg
- KSFE (Konferenz der SAS-Anwender in Forschung und Entwicklung)

Auf der Seite unseres Fachgebiets sind einige Fallstudien aufgelistet, die in aller Regel mit SAS ausgewertet wurden: http://www.uni-hohenheim.de/bioinformatik/beratung/

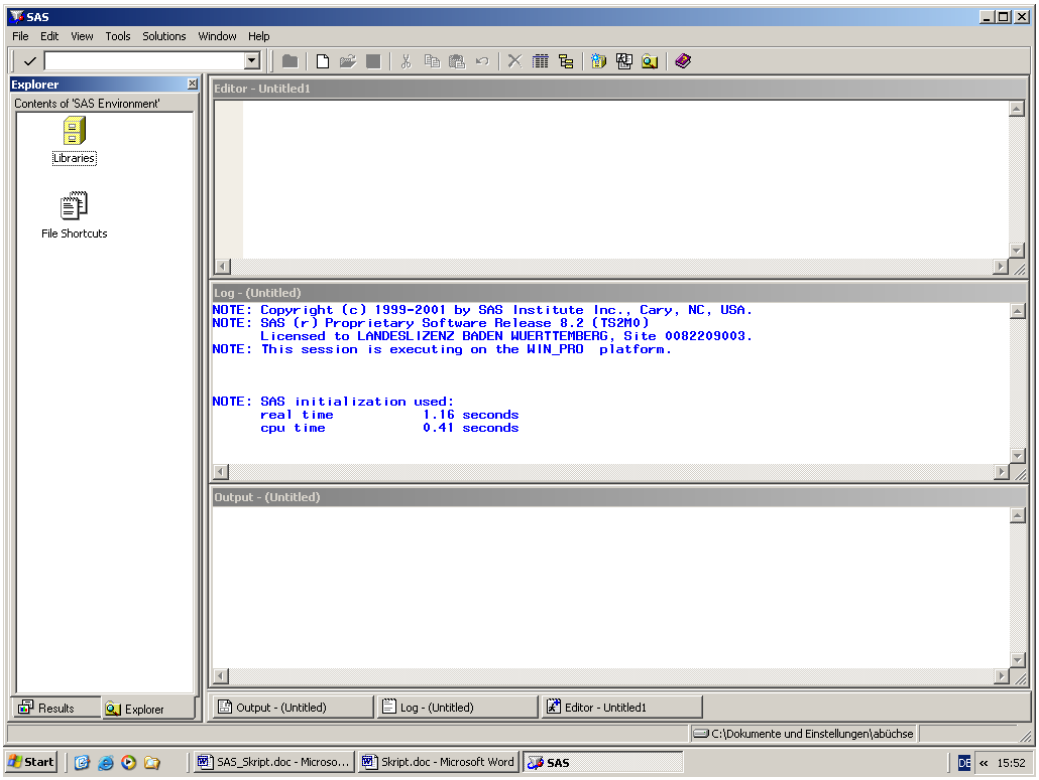
Nicht zuletzt sei natürlich die Nutzung der **Hilfe-Dateien** empfohlen, die Teil des Programms sind. Ja nach Art der Installation kann sich allerdings Aussehen und Erreichbarkeit der Hilfe unterscheiden.

Andreas Büchse, Stuttgart, im Juli 2006

1. Einführung in das SAS-System

Der Aufruf von SAS kann entweder über das Menü erfolgen (START > PROGRAMME > SAS-SYSTEM >...) oder per Doppelklick auf ein SAS-Symbol auf dem Desktop.
Nach dem Start erscheint eine in mehrere Abschnitte unterteilte Oberfläche. Die vier wichtigsten Fenster innerhalb dieser Oberfläche sind:

- EDITOR: Hier werden Anweisungen eingegeben
- OUTPUT: Ausgabe der Ergebnisse
- LOG: Hier wird Buch geführt über erledigte Befehle. Fehlermeldungen erscheinen rot.
- SAS-EXPLORER: Hier können Daten und Ergebnis-Files verwaltet und betrachtet werden.



Typische SAS-Programme haben folgende Grundstruktur:

1. Datenschritt
2. Auflistung der Daten
3. Prozedurschritt

```
Data beispiel;
Input y;
Datalines;
1
2
3
4
;
Proc Print data = beispiel;
Run;
```

Dieses kurze Beispielprogramm kann durch einen Klick auf den Submit-Button (das kleine Männchen in der Symbolleiste) gestartet werden und erzeugt zunächst im Datenschritt ein SAS-internes File „Beispiel“. Nach der Anweisung „datalines“ kommen die Daten und die Prozedur „PROC PRINT“ sorgt dafür, dass die Zahlen 1-4 auf den Bildschirm in das Output-Fenster geschrieben werden.

Anhand dieses kurzen Beispielprogramms kann man bereits einige grundsätzliche Konventionen erläutern:

- Jede Programmzeile muss mit einem Semikolon abschließen.
- Am Ende des Programms muss der Aufruf „run“ erfolgen.
- Jede Beobachtung (1-4) sollte in einer Zeile stehen.

1.1. Hilfe

Sehr gut ist die Hilfe in der SAS-Version 9. Erreichbar über „Help“ > „SAS Help and Documentation“
Die Hilfe zu statistischen Prozeduren steckt unter „SAS-Products“ > „SAS STAT“ > „SAS STAT USERS GUIDE“.

1.2. Import von Daten

Neben der Möglichkeit, die Daten direkt über das Editor-Fenster einzulesen, ist auch ein Import kompletter Dateien (z.B. aus EXCEL oder aus einer Textdatei) möglich. Hierzu bietet SAS einen „Import-Wizard“ der über das Menü „File“ > „Import Data“ gestartet wird.

Import einer Excel-Datei
Wichtig ist, das das Excel-File zum Zeitpunkt des Imports geschlossen ist und möglichst nur die reinen Daten enthält (keine Berechnungen, Leerzeilen usw.). In der ersten Zeile sollten die Variablennamen stehen. Umlaute, Leerzeichen usw. sind zu vermeiden.
Anstatt den Import-Wizard zu benutzen kann auch folgender Code im Editor eingegeben werden:

```
Proc import out=<Name des SAS files>
     Datafile = “<Laufwerk:\Pfad\EXCEL-Datei-Name>”
     DBMS = EXCEL2000 replace;
Getname = yes;
Run;
```

Angenommen es sind Daten aus der Datei “Versuch.XLS” im Ordner „C:\Daten“ zu importieren.

```
PROC IMPORT OUT= WORK.VERSUCH
            DATAFILE= "C:\Daten\Versuch.xls"
            DBMS=EXCEL2000 REPLACE;
     GETNAMES=YES;
RUN;
```

Import einer txt-Datei
Der Import einer txt-Datei (z.B. mit Tabulator als Trenner) läuft ganz ähnlich.

```
PROC IMPORT OUT= WORK.DON
            DATAFILE= "F:\SAS-Praktikum\Daten\DON.txt"
            DBMS=TAB REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

Dateneingabe über den Programm-Editor

Bei kurzen Datensätzen hat das Einlesen über den Editor Vorteile. SAS erwartet, dass Variablen in Spalten und Beobachtungen (records, Fälle) in den Zeilen stehen.

Beispiel: In einem Feldversuch wurden vier Sorten (A-D) auf jeweils vier Parzellen geprüft.

	A	B	C	D
1	21	18	19	14
2	22	16	19	13
3	19	15	16	12
4	18	13	14	11
Summe	80	62	68	50
Mittel	20,0	15,5	17,0	12,5

Die Eingabe kann in folgender Form erfolgen:

```
Data Versuch;
Input Sorte$ Wdh Ertrag;
Datalines;
A     1     21
A     2     22
A     3     19
A     4     18
B     1     18
B     2     16
B     3     15
B     4     13
C     1     19
C     2     19
C     3     16
C     4     14
D     1     14
D     2     13
D     3     12
D     4     11
;
run;
```

Es gibt insgesamt 16 Beobachtungen, also hat der Datenblock 16 Zeilen. Das Dollarzeichen ($) hinter dem Variablen-Namen „Sorte“ gibt SAS den Hinweis, dass dieses eine alphanumerische Variable ist. Im Gegensatz dazu sind „Parzelle“ und „Ertrag“ numerische Variablen. Die Reihenfolge der Variablen im Datenblock muss mit der in der Zeile „Input“ übereinstimmen.

Variablenlänge

Bei alphanumerischen Variablen mit mehr als 8 Zeichen muss die Länge im Input-Schritt definiert werden, sonst werden diese nach 8 Zeichen abgeschnitten.

```
Data Rassen;
Input Rinderrasse$ Gewicht;
Datalines;
Schwarzbunt 550
Rotbunt 450
;
Proc Print;
Run;
```

erzeugt lediglich den Output

Obs	Rinderrasse	Gewicht
1	Schwarzb	550
2	Rotbunt	450

Um die volle Länge der Variablen zu erhalten kann das LENGTH-Statement benutzt werden.

```
Data Rassen;
length Rinderrasse $ 15;
Input Rinderrasse$ Gewicht;
Datalines;
Schwarzbunt 550
Rotbunt 450
;
Proc Print;
Run;
```

Obs	Rinderrasse	Gewicht
1	Schwarzbunt	550
2	Rotbunt	450

Wenn die Datei bereits besteht, so kann die Variablenlänge nicht nachträglich geändert werden. Man kann jedoch eine zweite Variable mit spezifischer Länge definieren.

```
Data Rassen;
set Rassen;
length Rasse_kurz $ 2;
Rasse_kurz = Rinderrasse;
;
Proc Print;
Run;
```

Obs	Rinderrasse	Gewicht	Rasse_kurz
1	Schwarzbunt	550	Sc
2	Rotbunt	450	Ro

Die Länge kann auch beim Import von Daten aus anderen Formaten bestimmt werden. Bei Proc Import steht hierfür die Option „TEXTSIZE“ zur Verfügung.

So erzeugt zum Beispiel

```
PROC IMPORT OUT= WORK.VERSUCH
            DATAFILE= "I:\SAS-Praktikum\Daten\Versuch.xls"
            DBMS=EXCEL REPLACE;
     SHEET="Versuch";
     GETNAMES=YES;
RUN;
Proc Print data=versuch;
run;
```

Obs	Sorte	Block	Ertrag
1	Ariana	1	21
2	Ariana	2	22
3	Ariana	3	19
4	Ariana	4	18
5	Berta	1	18
6	Berta	2	16
7	Berta	3	15
8	Berta	4	13
9	Cojote	1	19
10	Cojote	2	19
11	Cojote	3	16
12	Cojote	4	14
13	Dora	1	14
14	Dora	2	13
15	Dora	3	12
16	Dora	4	11

Dagegen erzeugt

```
PROC IMPORT OUT= WORK.VERSUCH
            DATAFILE= "I:\SAS-Praktikum\Daten\Versuch.xls"
            DBMS=EXCEL REPLACE;
     SHEET="Versuch";
     GETNAMES=YES;
       TEXTSIZE=3;
RUN;
```

Obs	Sorte	Block	Ertrag
1	Ari	1	21
2	Ari	2	22
3	Ari	3	19
4	Ari	4	18
5	Ber	1	18
[...]			
15	Dor	3	12
16	Dor	4	11

Zusammenfügen von Variablen: Konkatenieren

Manchmal soll der Inhalt verschiedener Spalten in einer Spalte zusammengefügt werden. So etwas nennt man Konkatenieren.

```
data temp0;
length a $ 5;
length b $ 3;
input a$ b$ c$;
cards;
nasen bären 47
;
run;
proc print;
run;
```

Obs	a	b	c
1	nasen	bär	47

```
/*Konkatenieren*/
data temp1;
set temp0;
x=a||b||c;
keep x;
run;
proc print;
run;
```

Obs	x
1	nasenbär47

1.3. Permanente SAS-Dateien

Es kann sinnvoll sein, die eingelesenen Daten, für kommende SAS-Sitzungen in einer permanenten SAS-Datei abzulegen. Diese werden nicht in der library „WORK“ (deren Inhalt beim Beenden von SAS gelöscht wird) abgelegt, sondern in einem anderen Verzeichnis.
Dazu ist ein solches Verzeichnis zunächst einmal anzulegen (Name darf max. 8 Buchstaben haben). Das entsprechende Statement lautet: LIBNAME name ’verzeichnis’ ;

Zum Beispiel:

```
LIBNAME meinsas "c:\buechse\sas";
data meinsas.test1;
input x y;
datalines;
1 2
;
run;
```

Innerhalb der SAS-Oberfläche erscheint im Explorer ein library-Symbol mit dem Namen *meinsas*. Hierin befindet sich „virtuell“ die Datei *test1*. Im Windows-Explorer kann man die Datei test1 dagegen innerhalb des Verzeichnisses *c:\buechse.sas* finden. Dieses Verzeichnis muss aber vorher angelegt sein, also physikalisch vorhanden sein! Die entsprechende Meldung hierzu im Log-Fenster:

```
1083  LIBNAME meinsas "c:\buechse\sas";
NOTE: Libref MEINSAS was successfully assigned as follows:
      Engine:        V8
      Physical Name: c:\buechse\sas
1084  data meinsas.test1;
1085  input x y;
1086  datalines;
NOTE: The data set MEINSAS.TEST1 has 4 observations and 2 variables.
NOTE: DATA statement used:
      real time           0.01 seconds
      cpu time            0.00 seconds
1091  ;
1092  run;
```

1.4. Export von SAS-Dateien

Wenn in SAS Ergebnisse produziert wurden, möchte man diese oftmals z.B. in Excel zwecks Tabellierung exportieren. Der folgende Beispiel-Code bewirkt einen Export der SAS-Datei „Versuch“ aus dem SAS-Ordner „Work“ in die Excel-Datei „Versuch.xls“, die sich im Ordner „C:\Daten“ befindet.

```
PROC EXPORT DATA= WORK.Versuch
            OUTFILE= "C:\Daten\Versuch.xls"
            DBMS=EXCEL2000 REPLACE;
RUN;
```

Daneben kann der Export-Wizard benutzt werden. Siehe Menü „File > Export Data“.

1.5. Optionen

In SAS sind verschiedene Optionen verfügbar, um die Arbeit angenehm zu gestalten. So kann man zum Beispiel den Output linksbündig ausrichten (Default ist zentriert) und die Zeilenlänge definieren.

```
/*Linksbündig schreiben*/
Options nocenter linesize=85;
run;
```

Wenn gewünscht kann der Output mit einer informativen Titelzeile überschrieben werden.

```
Title „SAS-Seminar 17.12.2003“;run;
```

Rückgängig machen:

```
Title; run;
```

Weitere Optionen können über das Menü „Tools > Options > ...“ eingestellt werden.

1.6. Output im rtf-Format

Das Layout innerhalb des Output-Editors ist relativ dürftig. Falls ein Einfügen von SAS-Output z.B. in Word-Dokumente geplant ist, so kann die Ausgabe im rtf-Style sehr nützlich sein. Hierzu ist folgender Progamm-Code notwendig:

```
ods rtf body = 'body.rtf';
     […  übriger Programmcode …]
ods rtf close;
```

Dieser Code erzeugt eine temporäre rft-Datei. Diese kann direkt in SAS betrachtet werden. Die rtf-Ansicht sollte jeweils wieder geschlossen werden, bevor ein neuer Output eingefügt werden kann. Noch eleganter ist es, sich die Ausgabe sofort in eine permanente Datei schreiben zu lassen. Zum Beispiel:

```
ods rtf file = 'c:\SAS-Ergebnisse\Versuch.rtf';
     proc print data=Versuch;
     run;
ods rtf close;
```

2. Einfache Prozeduren

Wenn die Daten eingelesen sind und SAS intern in einem File gespeichert sind (in der Regel in der library „work“, dann sollen Auswertungen und Statistiken erzeugt werden. Im Folgenden sind jeweils kurze (selbsterklärende) Programme aufgelistet.

2.1. PROC SORT

```
/*Sortieren*/
Proc sort data=Versuch;
by ertrag;
Proc print data=versuch;
run;
```

2.2. PROC RANK

```
/*Ränge bilden*/
Proc rank data=versuch out=Raenge;
var ertrag;
run;
Proc print data=Raenge;
run;
```

2.3. PROC MEANS

```
/*Mittelwert, Median, Varianz, Standardabweichung, Vertrauensbereich*/
Proc Means data=Versuch mean median var std max min clm N;
var Ertrag;
run;

/*Das gleiche, aber Ergebnis in Datei schreiben*/
Proc Means data=Versuch mean median var std max min clm noprint;
var Ertrag;
output out = Ausgabe mean=mean median=median var=var std=std;run;
Proc Print data=ausgabe;run;
```

Obs	_TYPE_	_FREQ_	mean	median	var	std
1	0	16	16.25	16	10.8667	3.29646

2.4. PROC UNIVARIATE

```
Proc Univariate data=Versuch normal plot;
var Ertrag;
qqplot/normal;
histogram;
run;
```

Univariate erzeugt eine große Vielfalt statistischer Maßzahlen. Mit der Option „normal plot“ wird ein Test auf Normalverteilung durchgeführt. Ein QQ-Plot ist eine graphische Prüfung auf Normalverteilung. Wichtig: Für eine Varianzanalyse sollen nicht die Daten selbst sondern die Residuen normalverteilt sein!

3. Datenmanagement

Sehr wichtig kann es sein, in einem Data-Step mehrere Datenfiles zu kombinieren. Die Daten können entweder untereinander (set) oder nebeneinander (merge) gesetzt werden.

3.1. Der Befehl „set“

Hiermit kann man SAS-Dateien modifizieren oder zwei SAS-Dateien **untereinander** setzen. Angenommen wir haben bereits eine Datei Versuch1 und bekommen jetzt neue Daten von einem Versuch2. Ziel ist es, eine gemeinsame Datei zu erzeugen.

```
Data Versuch2;
Input Sorte$ Wdh Ertrag;
Datalines;
A    1    31
A    2    32
B    1    39
B    2    38
; run;
data versuch1;
set versuch;
ort = 1; run;
data versuch2;
set versuch2;
ort = 2; run;
data Serie;
set versuch1 versuch2; run;
Proc Print data=serie; run;
```

Obs	Sorte	Wdh	Ertrag	ort
1	A	1	21	1
2	A	2	22	1
3	A	3	19	1
4	A	4	18	1
5	B	1	18	1
6	B	2	16	1
7	B	3	15	1
8	B	4	13	1
9	C	1	19	1
10	C	2	19	1
11	C	3	16	1
12	C	4	14	1
13	D	1	14	1
14	D	2	13	1
15	D	3	12	1
16	D	4	11	1
17	A	1	31	2
18	A	2	32	2
19	B	1	39	2
20	B	2	38	2

3.2. Der Befehl „merge“

Im Gegensatz zum untereinander setzen bei „set“ werden durch „merge“ SAS-Dateien **nebeneinander** gesetzt. Die Daten müssen vor dem *mergen* sortiert werden.

Ganz wichtig ist das „by“-Statement. Dieses regelt, dass die Zeilen richtig zugeordnet werden. „By“ ist immer erforderlich! Es muss sich hierbei um eine oder mehrere Variablen handeln, die in beiden zu mergenden Datensätzen vorkommen.

```
Data Versuch2;
Input Sorte$ Wdh Ertrag;
Datalines;
A    1    31
A    2    32
B    1    39
B    2    38
;
run;

Data Bonitur2;
Input Sorte$ Wdh Bonitur;
Datalines;
A    1    4
A    2    2
B    1    7
B    2    8
;
run;

data versuch3;
merge versuch2 Bonitur2;
by Sorte Wdh;
/*!Wichtig: Daten falls nötig vorher sortieren!*/
run;
Proc print data=versuch3;
run;
```

Obs	Sorte	Wdh	Ertrag	Bonitur
1	A	1	31	4
2	A	2	32	2
3	B	1	39	7
4	B	2	38	8

3.3. Rechenoperationen

z.B. Parzellenertrag über Erntefläche auf t/ha umrechen.

```
data versuch;
set versuch;
efl = 12.0;
Ertrag = (ParzErtrag / efl) * 10;
run;
```

3.4. Logische Operatoren

Häufig müssen Daten nach bestimmten Kriterien selektiert werden oder einzelne Einträge gelöscht werden oder man möchte Auswertungen nur für einen Teildatensatz machen. Hier helfen logische Operatoren.

Die wichtisten sind:

Where *[Bedingung]* ;
Befehl wird nur für Objekte ausgeführt wo die where-Abfrage den Wert „wahr“ liefert.

If *[Bedingung]* **then** *[Aktion]* ;
Wenn - Dann

If *[Bedingung]* **then** *[Aktion]* **else** *[Alternative]* ;
Wenn – Dann; Wenn nicht – Dann;

If *[Bedingung]* **then** *[Aktion]* **else if** [Weitere Abfrage einer Bedingung] **else** *[Alternative]* ;
Wenn – Dann; Wenn nicht – Dann Prüfung auf weitere Eigenschaft – wenn beides negativ Dann;

„=“ steht für „gleich“
„ne“ steht für ungleich
Ansonsten „<“ „>“

```
Data Bonitur3;
Input Sorte$ Wdh Bonitur;
Datalines;
A    1    4
A    2    2
B    1    7
B    2    8
C    1    5
C    2    4
;
Data neu;
Set Bonitur2;
Where Sorte ne “A”;
If Bonitur < 3 then Anfaelligkeit = “niedrig“;
else if Bonitur < 6 then Anfaelligkeit = „mittel“;
else Anfaelligkeit = „hoch“;
Proc print;
run;
```

Output:

Obs	Sorte	Wdh	Bonitur	Anfaelligkeit
1	B	1	7	hoch
2	B	2	8	hoch
3	C	1	5	mittel
4	C	2	4	mittel

3.5. Relativierung

Durch ein Zusammenspiel der Prozedur MEANS mit *set* und *merge* in einem Data-Step, kann auch eine Relativierung zum Beispiel auf das Versuchsmittel oder eine Kontrollvariante erreicht werden.

Datenbeispiel:
Im Sommersemester 2004 wurden im Agrarbiologischen Großpraktikum ein Feldversuch mit drei Sorten und vier Düngungsstufen angelegt. Jede Kombination wurde jeweils in vierfacher Wiederholung geprüft. Die Daten sind im Internet verfügbar unter der Adresse

www.uni-hohenheim.de/bioinformatik/lehre/module/saspraktikum/daten/DuengungWeizen.dat

Wdh	Sorte	N	Parz	FM	AnzHalme	FMKoerner	Wdh	Sorte	N	Parz	FM	AnzH	FMK
W1	S1	N4	1	1677	162	473.82	W3	S2	N3	25	1578	177	459
W1	S2	N1	2	1743	204	554.85	W3	S3	N1	26	1128	120	415.32
W1	S3	N3	3	2472	291	823.26	W3	S1	N1	27	1239	132	344.31
W1	S3	N4	4	2319	231	791.94	W3	S3	N4	28	2097	213	649.86
W1	S1	N2	5	1593	195	424.33	W3	S2	N1	29	1287	156	407.13
W1	S2	N2	6	1488	183	491.28	W3	S1	N3	30	1742	183	414.8
W1	S2	N4	7	1674	180	519.09	W3	S1	N4	31	1695	180	395.13
W1	S3	N2	8	1596	162	537.24	W3	S3	N2	32	1620	171	517.59
W1	S1	N1	9	810	138	268.47	W3	S2	N2	33	1539	180	454.83
W1	S2	N3	10	1506	174	472.47	W3	S2	N4	34	1842	189	528.84
W1	S3	N1	11	1254	129	445.29	W3	S1	N2	35	1626	177	425.25
W1	S1	N3	12	1737	186	448.26	W3	S3	N3	36	1836	201	568.56
W2	S1	N4	13	2058	204	521.19	W4	S2	N1	37	1059	141	370.86
W2	S2	N3	14	1479	156	389.31	W4	S1	N4	38	1515	171	430.77
W2	S3	N4	15	1545	168	500.46	W4	S3	N2	39	1197	147	438
W2	S1	N3	16	1587	156	419.5	W4	S1	N2	40	1368	153	432
W2	S2	N1	17	1131	126	380.4	W4	S2	N4	41	1683	213	480
W2	S3	N1	18	1188	123	393	W4	S3	N3	42	2049	195	606.51
W2	S3	N2	19	1764	192	590.28	W4	S3	N1	43	1197	123	414.63
W2	S2	N4	20	1701	185	509.76	W4	S2	N3	44	1833	216	502.8
W2	S1	N2	21	1104	117	327.9	W4	S1	N3	45	1854	195	432
W2	S3	N3	22	1593	168	468.75	W4	S2	N2	46	1428	165	430.11
W2	S1	N1	23	894	105	258	W4	S3	N4	47	2046	216	648.6
W2	S2	N2	24	1239	117	387	W4	S1	N1	48	825	102	216

Düngungsstufe N4 wird zunächst nicht betrachtet. Die Daten werden Versuchsgliedweise sortiert (PROC SORT) und gemittelt (PROC MEANS). Die mittleren Frischmassen werden in eine Datei „Mittel“ exportiert und diese auf den Bildschirm gedruckt. Unter *_FREQ_* ist erkennbar, dass jeder Mittelwert aus vier Einzelwerten gebildet wurde. Anschließend sollen die Frischmassen für jede Sorte jeweils auf den Wert der ungedüngten Stufe relativiert werden.

```
/*Datenmanagement*/
data Weizen; set Weizen;
If N = "N4" then delete;
If Sorte = "S1" then Sortenname="Monopol";
If Sorte = "S2" then Sortenname="Batis";
If Sorte = "S3" then Sortenname="Hybnos";
If N = "N1" then N_Menge=0;
If N = "N2" then N_Menge=80;
If N = "N3" then N_Menge=160;
run;
```

```
/*Mittelwerte je Versuchsglied*/

proc sort data=weizen;
by Sorte N_Menge;
proc means data=Weizen noprint;
by Sorte N_Menge;
var FM;
output out=Mittel mean=m_FM;
run;

proc print data=Mittel;
run;
```

Obs	Sorte	N_Menge	_TYPE_	_FREQ_	m_FM
1	S1	0	0	4	942.00
2	S1	80	0	4	1422.75
3	S1	160	0	4	1730.00
4	S2	0	0	4	1305.00
5	S2	80	0	4	1423.50
6	S2	160	0	4	1599.00
7	S3	0	0	4	1191.75
8	S3	80	0	4	1544.25
9	S3	160	0	4	1987.50

```
/*Relativierung auf ungedüngt*/

Data Ungeduengt;
set Mittel;
if N_Menge=0;
FM_0 = m_FM;
run;

Data Relativ;
merge Mittel Ungeduengt;
by Sorte;
FM_rel = m_FM/FM_0 * 100;
run;
Proc Print data=Relativ;
run;
```

Obs	Sorte	N_Menge	_TYPE_	_FREQ_	m_FM	FM_0	FM_rel
1	S1	0	0	4	942.00	942.00	100.000
2	S1	80	0	4	1422.75	942.00	151.035
3	S1	160	0	4	1730.00	942.00	183.652
4	S2	0	0	4	1305.00	1305.00	100.000
5	S2	80	0	4	1423.50	1305.00	109.080
6	S2	160	0	4	1599.00	1305.00	122.529
7	S3	0	0	4	1191.75	1191.75	100.000
8	S3	80	0	4	1544.25	1191.75	129.578
9	S3	160	0	4	1987.50	1191.75	166.772

3.6. Der Befehl „keep“

Der Output im vorigen Beispiel enthält noch viele Variablen, die nur zur Berechnung erforderlich waren, die man aber später nicht mehr benötigt. Mit Keep kann eine Variablenauswahl selektiert werden.

```
Data Relativ2;
set Relativ;
keep Sorte N_Menge FM_rel;
Proc Print data=Relativ2;
run;
```

Obs	Sorte	N_Menge	FM_rel
1	S1	0	100.000
2	S1	80	151.035
3	S1	160	183.652
4	S2	0	100.000
5	S2	80	109.080
6	S2	160	122.529
7	S3	0	100.000
8	S3	80	129.578
9	S3	160	166.772

3.7. Der Befehl „drop“

Neben einer Positivliste kann man natürlich auch nach einer Negativliste selektieren. Mit dem Befahl Drop lässt sich der gleiche Output erzeugen.

```
/*DROP*/
Data Relativ3;
set Relativ;
drop _TYPE_ _FREQ_ m_FM FM_0 ;
Proc Print data=Relativ3;
run;
```

3.8. Simulationen

Bei bestimmten Fragestellungen ist es sinnvoll, sich basierend auf einem Modell Daten zu simulieren. Hierfür kann der Zufallszahlengenerator von SAS verwendet werden. Über die Funktion „rannor(-1)“ werden normalverteilte Zufallszahlen erzeugt. Diese haben den Erwartungswert 0 und die Varianz 1. Durch Multiplikation mit entsprechenden Varianzkomponenten kann z.B eine Versuchsserie simuliert werden.

```
data sorteneffekte;
      do sorte = 1 to 24 by 1;
            var_s = 1;
            s_eff = sqrt(var_s)*rannor(-1);
            output;
      end;
run;
data ortseffekte;
      do ort = 1 to 4 by 1;
            var_o = 9;
            o_eff = sqrt(var_o)*rannor(-1);
            output;
      end;
run;
data Interaktionen;
      do ort = 1 to 4 by 1;
            do sorte = 1 to 24 by 1;
                  var_int = 3;
                  int_eff = sqrt(var_int)*rannor(-1);
                  output;
            end;
      end;
run;
data simu1;
merge Interaktionen Ortseffekte;
by ort;
run;
proc sort data=simu1;
by sorte;
data simu2;
merge simu1 Sorteneffekte;
by sorte;
      var_e = 1;
      fehler = sqrt(var_e)*rannor(-1);
      y = s_eff + o_eff + int_eff + fehler;
run;
proc mixed data=simu2;
class ort sorte;
model y = ;
random ort sorte ;
run;
```

```
                    The Mixed Procedure

                Covariance Parameter
                      Estimates

                Cov Parm     Estimate

                ort           12.9055
                sorte          1.5920
                Residual       3.5539
```

4. Grafiken und Deskriptive Statistik

SAS-Grafiken können über „***rechte Maustaste > File > Export as Image***“ im gewünschten Format (z.B. jpeg) an den gewünschten Ort gespeichert werden. Alternativ ist ein Speichern als Vektorgrafik möglich (siehe 4.5).

4.1. PROC GPLOT

Bei der Prozedur GPlot wird ein Grafik-Modul aufgerufen, das mit der Anweisung „quit“ wieder geschlossen werden sollte. Durch die Nutzung verschiedener Optionen können die Grafiken im Ausehen an die eigenen Vorlieben angepasst werden.

Datenbeispiel (siehe auch Kapitel 3.5):
Im Sommersemester 2004 wurden im Agrarbiologischen Großpraktikum ein Feldversuch mit drei Sorten und vier Düngungsstufen angelegt. Jede Kombination wurde jeweils in vierfacher Wiederholung geprüft. Die Daten sind im Internet in der Datei DuengungWeizen.dat abgelegt.
Die Daten werden in SAS eingelesen, als weitere Variable wird das Ährengewicht berechnet. Die Sortencodes S1 bis S3 werden in Namen und die Düngungsstufen N1 bis N3 in Düngermengen umgewandelt. Die Menge N4 wird zunächst gelöscht und nicht weiter betrachtet. Dann wird mit der Prozedur GPLOT ein Scatterplot erzeugt. Die Option „Symbol value=dot height=1“ weist den Punkten als Symbol einen fetten Punkt zu.

```
options linesize=80 nodate nonumber nocenter;
Data Weizen;
input Wdh$ Sorte$ N$ FM AnzHalme FMKoerner;
Aehrengew = FMKoerner/AnzHalme;
datalines;
W1   S2   N1   1743 204  554.85
W1   S3   N3   2472 291  823.26
[… mehr Daten ...]
W4   S1   N1   825  102  216.00
;
run;

data Weizen; set Weizen;
If N = "N4" then delete;
If Sorte = "S1" then Sortenname="Monopol";
If Sorte = "S2" then Sortenname="Batis";
If Sorte = "S3" then Sortenname="Hybnos";
If N = "N1" then N_Menge=0;
If N = "N2" then N_Menge=80;
If N = "N3" then N_Menge=160;
run;

goptions reset=all;
goptions ftext=swiss htext=1.5 ;
proc gplot data=Weizen;
symbol value=dot height=1;
plot FM*N_Menge;
run; quit;
```

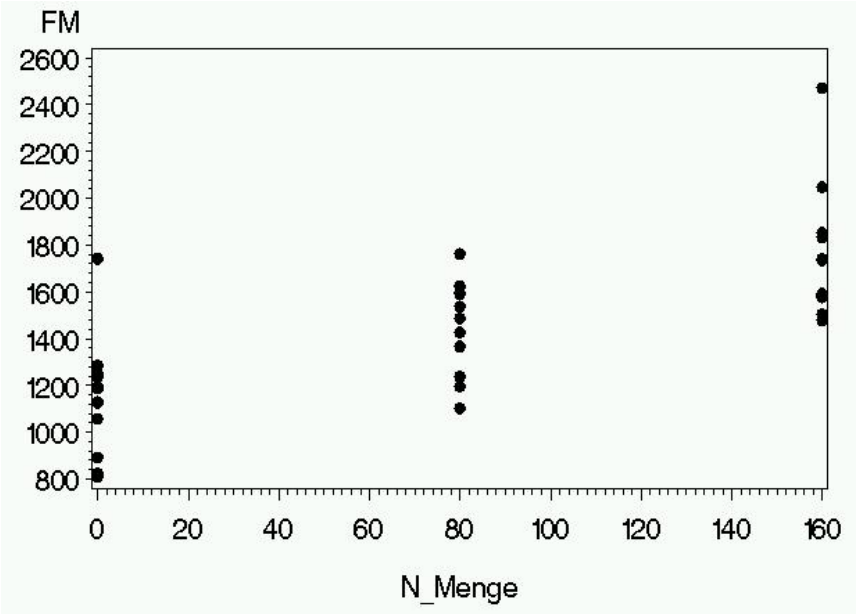


Abb. 2: Scatterplot

Diese Abbildung kann über weitere Optionen noch ansprechender gestaltet werden. Die Option „`i = rl`“ sorgt für Trendlinien auf Basis einer linearen Regression. Mit der Option ´“`width=2`“ erhält man eine schöne fette Linie. Durch „`=Sortenname`“ bekommt jede Sorte ihre eigene Trendlinie. Der ausgiebige Gebrauch der Symbol-Anweisung teilt jeder Sorte ihr eigenes Symbol zu (sonst wird nur eine farbliche Unterscheidung gegeben, die Sie in diesem Schwarz-Weiß-Ausdruck nicht sehen könnten). Schließlich werden noch die Variablenlabel umbenannt. Die resultierende Grafik (Abb. 3) kann in SAS über die rechte Maustaste oder „TOOLS > Graphics Editor“ geöffnet und noch nachbearbeitet werden, so dass man die Abb. 4 erhält.

Die fertige Grafik kann dann über „***rechte Maustaste > File > Export as Image***“ im gewünschten Format (z.B. jpeg) an den gewünschten Ort gespeichert werden.

```
/*verbesserter Scatterplot mit Trendlinie*/
goptions reset=all;
goptions ftext=swiss htext=1.5 ;
proc gplot data=Weizen;
symbol value=dot height=1 i=rl width=2;
plot FM*N_Menge=Sortenname;
label FM = 'Frischmasse [g]';
label N_Menge = 'N [kg/ha]';
run;quit;
```

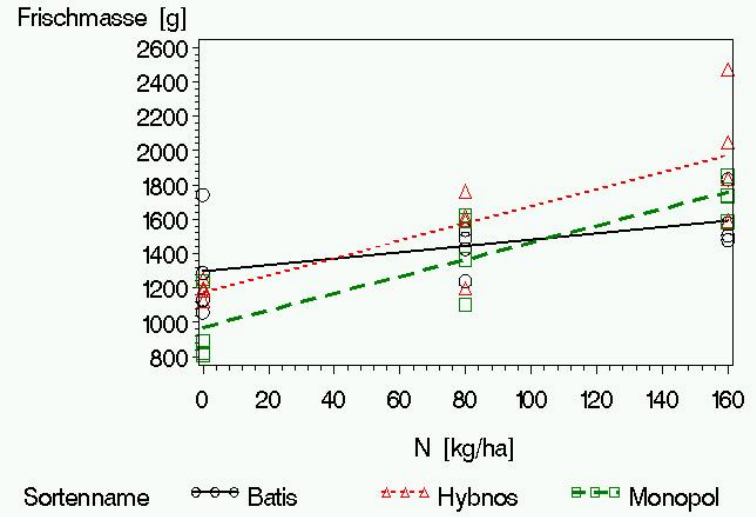


Abb. 3: Lineare Regressionen

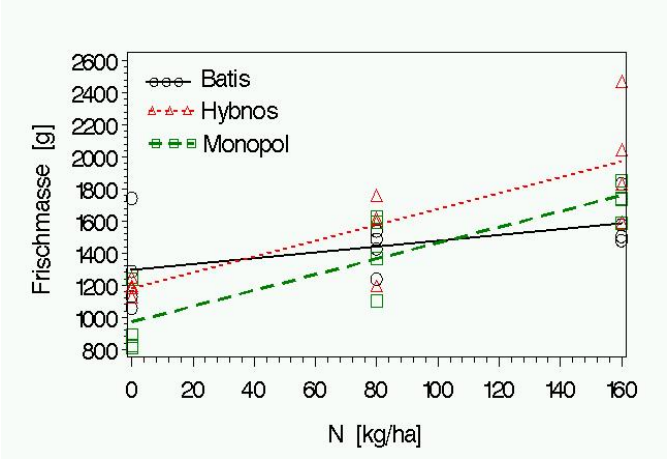


Abb. 4: Lineare Regressionen nach Bearbeitung im SAS-Graphic Editor

4.2. PROC BOXPLOT

Ein Boxplot bietet eine gleichzeitige Darstellung von Mittelwert, Median, Verteilung und Spannweite der Daten. Dicke der Linien und Boxes können eingestellt werden.

```
/*Boxplot*/

goptions reset=all;
goptions ftext=swiss htext=1.5 ftrack=loose hsize=6 vsize=4;
Proc Sort data=weizen;
by N_Menge;
proc boxplot data=Weizen ;
plot FM*N_Menge / boxwidth=10 vaxis=500 to 2500 by 500 cboxes=black
waxis=2 ;
symbol width=3 ;
run;
quit;
```

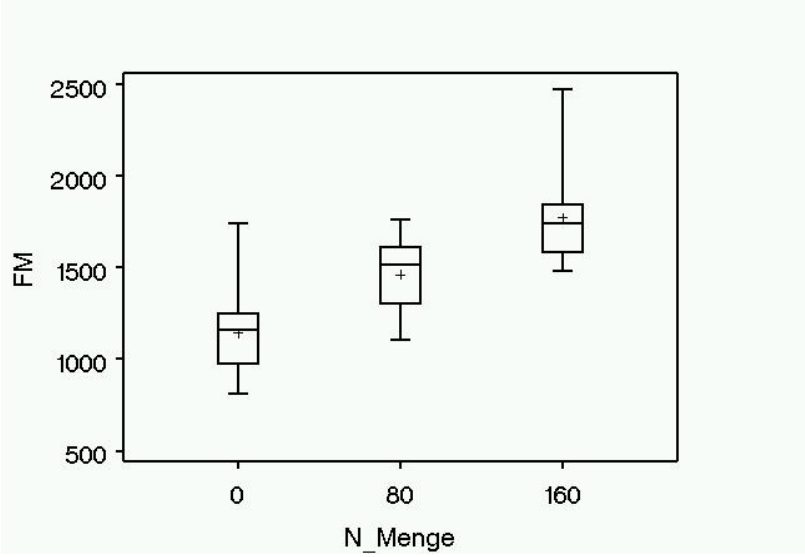


Abb. 5: Boxplot

In der Grundeinstellung wird der Boxplot nach folgenden Einstellungen gezeichnet (Kopie aus der SAS-Online-Hilfe:

Maximum	*Endpoint of upper whisker*
Third quartile (75th percentile)	*Upper edge of box*
Median (50th percentile)	*Line inside box*
Mean	*Symbol marker*
First quartile (25th percentile)	*Lower edge of box*
Minimum	*Endpoint of lower whisker*

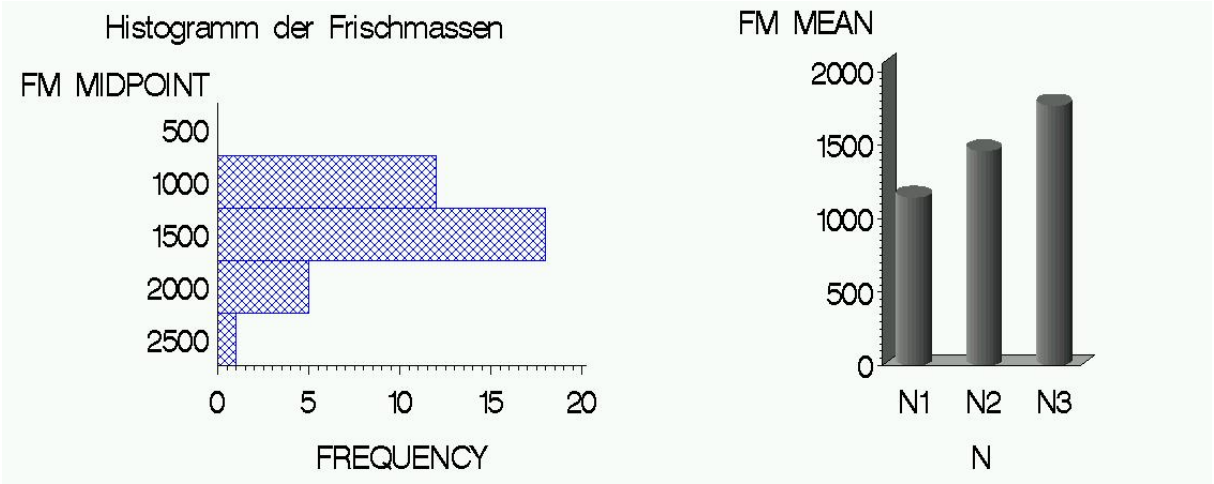
4.3. PROC GCHART

Mit dieser Prozedur können Histogramme und Säulendiagramme erzeugt werden. Vertikale Säulen bekommt man durch „vbar“ horizontale durch „hbar“. Für den der es mag auch in 3D.

```
/*Histogramm*/
Goptions Ftext=swiss htext=1.2;
Pattern Value=x1 color=blue;
Title "Histogramm der Frischmassen";
Proc gchart Data=weizen;
hbar FM /type=freq space=0 noframe nostats midpoints=500 to 2500 by
500;
run; quit;

/*3d Säulendiagramm*/
Goptions Ftext=swiss htext=2;
Pattern Value=x1 color=grey;
Proc gchart Data=Weizen;
vbar3d N /type=mean sumvar=FM space=4
noframe shape=cylinder;
run; quit;
Goptions reset=all;
run;
```

Alternative für Histogramm:

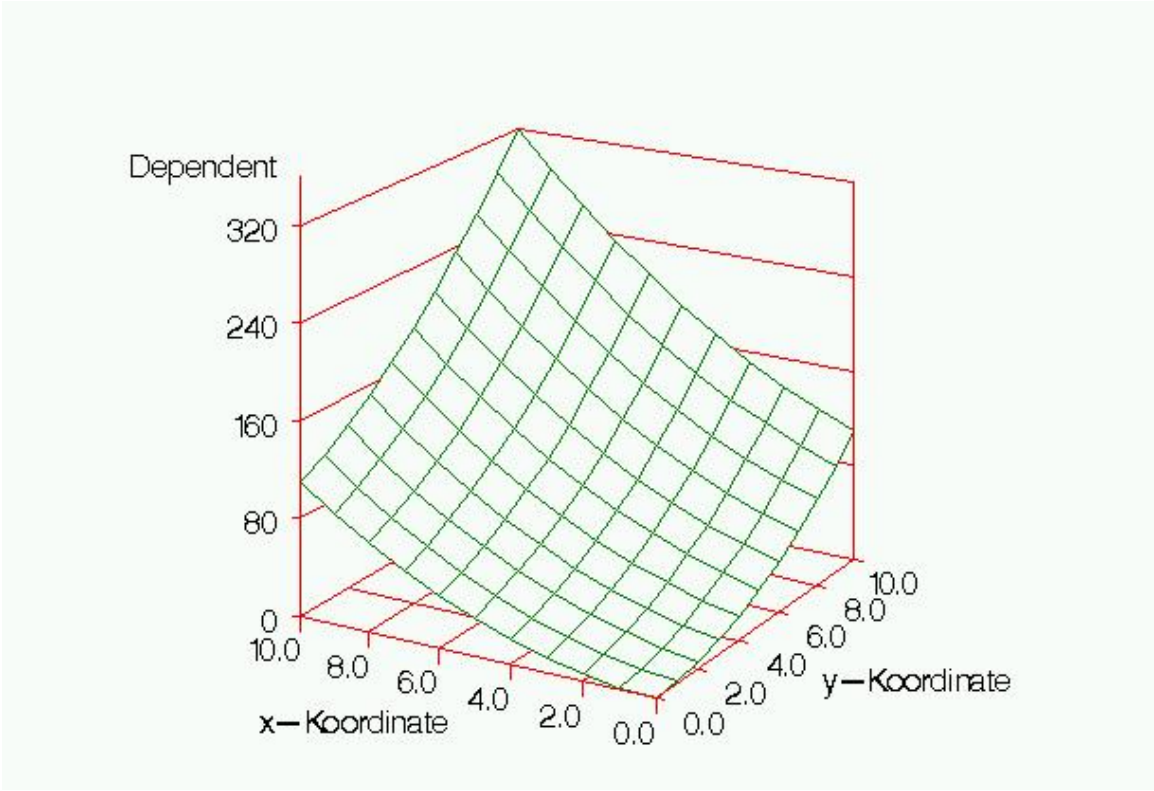
```
/*Histogramm*/
proc sort data=weizen;
by sorte;
proc univariate data=weizen;
by sorte;
var FM;
histogram;
run;
```



4.4. PROC G3D

Diese Prozedur ist speziell für 3D-Grafiken entwickelt. Es werden Flächen gezeichnet. So z.B. eine dreidimensionale Polynom-Oberfläche.

```
data surface;
      do x = 0 to 10 by 1;
            do y = 0 to 10 by 1;
                  z = x + y + x*y + x**2 + y**2;
                  output;
            end;
      end;
run;

goptions reset=all;
goptions htext=1.2 ftrack=tight
ftext=swiss hsize=8 vsize=4;

proc g3d data=surface;
plot x*y=z /grid tilt=75
rotate=60 xticknum=6 yticknum=6 zticknum=5;
format z f3.0; format x  f4.1; format y  f4.1;
label x = "x-Koordinate"; label y = "y-Koordinate";
label z = "Dependent";
run;
```



4.5. SAS-Grafiken als Vektorgrafik in einer Datei ablegen

Damit die Bilder nicht immer so pixelig aussehen, kann man sie im Format .emf in eine Datei schreiben lassen, die dann aus jeder anderen Software heraus verwendet werden können (z.B. Import als Grafik in Word). Hierzu ist der Pfad der Datei zu nennen und als device emf festzulegen.

```
filename f 'c:\buechse\xyz.emf';
goptions gsfname=f gsfmode=replace device=emf
```

Beispiel:

```
/*Boxplot als Vektorgrafik*/

filename f 'c:\buechse\schoenerboxplot.emf';
goptions gsfname=f gsfmode=replace device=emf
ftext='Arial' htext=1.3 hsize=8 cm vsize=6 cm;
symbol value=dot h=1 i=none;

Proc Sort data=weizen;
by N_Menge;

/*Boxplot*/
Proc Sort data=weizen;
by N_Menge;
proc boxplot data=Weizen ;
plot FM*N_Menge / boxwidth=10 vaxis=500 to 2500 by 500 cboxes=black
waxis=2 ;
symbol width=3 ;
run;
quit;



/*Scatterplot als Vektorgrafik*/

filename f 'c:\buechse\schoenerscatterplot.emf';
goptions gsfname=f gsfmode=replace device=emf
ftext='Arial' htext=1.5 hsize=12 cm vsize=8 cm;

/*verbesserter Scatterplot mit Trendlinie*/

proc gplot data=Weizen;
symbol1 value=circle   h=1.5 i=rl cv=black line=1 w=2;
symbol2 value=triangle h=1.5 i=rl cv=red line=2 w=2;
symbol3 value=square   h=1.5 i=rl cv=green line=3 w=2;
plot FM*N_Menge=Sortenname;
label FM = 'Frischmasse [g]';
label N_Menge = 'N [kg/ha]';
run;quit;
```

5. Randomisationspläne in SAS erstellen

Vor der Durchführung eines Versuchs steht die Versuchsplanung. Die Zuordnung der Behandlungen zu den Parzellen, Töpfen, Schalen usw. sollte zufällig (randomisiert) erfolgen. SAS arbeitet mit einem Pseudozufallszahlengenerator. Diesem muss ein (zufälliger) Startwert gegeben werden (Seed-Anweisung). Wird jeweils die gleiche Zahl verwendet, wird auch immer der gleiche Plan generiert. Durch den Befehl Output out wird das Ergebnis der Randomisation in eine Datei geschrieben, die auch exportierbar ist (z.B. nach EXCEL). Abschließend werden noch laufende Parzellennummern zugeordnet. Für verschiedene Versuchsanlagen wird hier jeweils ein Beispiel gegeben.

5.1. Zufallszahlen mittels RANUNI erzeugen

```
/* Zufallszahlen generieren*/
data zufall;
do block =1 to 4;
     do sorte = 1 to 5;
          zahl = ranuni (123);
          output;
     end;
end;
run;
proc print;
run;

proc sort data=zufall;
by block zahl;
run;
proc print;
run;
```

Mit RANUNI kann man für jede Zeile in einem SAS-File eine Zufallszahl erzeugen, die dann anschließend über PROC SORT zu sortieren ist. Durch Kombination von RANUNI, Sortieren und mehreren Data-Steps kann jede gewünschte Struktur erzeugt bzw. nach und nach die Randomisationsschritte eines mehrstufigen Plans vollzogen werden.

5.2. Vollständig randomisierte Anlage (CRD)

Die vollständig randomisierte Anlage ist immer dann von Vorteil, wenn keine großen Störeffekte zu erwarten sind, die einem Blockfaktor zuzuordnen sind und wenn die Zahl der Fehlerfreiheitsgrade beschränkt ist. Diese Anlageform ist deshalb besonders bei kleinen Versuchen sinnvoll.

```
Title "CRD 3 VG auf 6 Parzellen";
Proc Plan seed = 7804193;
Factors parzelle=6 random;
output out = crd;
run;
Data crd;
set crd;
if _N_ = 1 then VG = 1;
```

```
if _N_ = 2 then VG = 1;
if _N_ = 3 then VG = 2;
if _N_ = 4 then VG = 2;
if _N_ = 5 then VG = 3;
if _N_ = 6 then VG = 3;
run;
Proc print data=crd;
run;
```

Obs	parzelle	VG
1	4	1
2	1	1
3	3	2
4	2	2
5	5	3
6	6	3

5.3. Blockanlage (RCBD)

Die Blockanlage besteht aus vollständigen Wiederholungen (block) und Parzellen innerhalb des Blocks. Die Behandlungen (Treatment) werden zufällig auf die Parzellen innerhalb des Blocks verteilt.

```
Proc Plan seed = 7804193;
Factors block=3 ordered
          Treatment=12 random;
output out = blockanlage;
run;
Data blockanlage;
set blockanlage;
Parzelle = _N_;
run;
```

```
The PLAN Procedure

Factor         Select      Levels     Order

block               3           3    Ordered
Treatment          12          12    Random

   block      -------------Treatment-------------

       1       8  1  3  4  2  9  5 12  6 10 11  7
       2      11  6  1  4  5 12 10  9  2  8  3  7
       3       7  6  9 11 10  2  8  4  1  5  3 12
```

5.4. Spaltanlage

Die Spaltanlage besteht aus vollständigen Wiederholungen (block), Großteilstücken (Mainplot) und Kleinteilstücken (Subplot).

```
/*Spaltanlage*/
Proc Plan seed = 7804193;
Factors block=4 ordered
          Mainplot=4 random
          Subplot=4 random ;
output out = splitplot;
run;
Data splitplot;
set splitplot;
Parzelle = _N_;
run;
```

```
The PLAN Procedure

Factor        Select      Levels     Order

block              4           4    Ordered
Mainplot           4           4    Random
Subplot            4           4    Random


   block Mainplot      --Subplot-

       1        3      3  2  1  4
                1      2  4  3  1
                2      3  4  1  2
                4      1  4  2  3
       2        4      4  2  3  1
                3      3  4  2  1
                2      3  1  2  4
                1      3  1  2  4
       3        4      3  2  4  1
                1      4  1  3  2
                3      1  4  2  3
                2      1  2  3  4
       4        2      1  2  4  3
                1      4  1  3  2
                3      4  2  1  3
                4      2  4  1  3
```

5.5. Eine Serie von Spaltanlagen

```
Title "Versuchsserie mit Splitplot";
Proc Plan seed = 7804193;
Factors Ort=3 ordered
          block=2 ordered
          Mainplot=2 random
          Subplot=8 random ;
output out = serie;
run;
```

5.6. Anlagen in unvollständigen Blöcken

Hier ein Beispiel basierend auf dem STAT-USERS-Guide.
Man sieht, es ist prinzipiell möglich. Ich persönlich bevorzuge zur Erstellung von Gitterplänen allerdings die Programme ALPHA+ und CycDesigN.

Example 55.3: An Incomplete Block Design
Jarrett and Hall (1978) give an example of a generalized cyclic design with good efficiency characteristics. The design consists of two replicates of 52 treatments in 13 blocks of size 8. The following statements use the PLAN procedure to generate this design in an appropriately randomized form and store it in a SAS data set. Then, the TABULATE procedure is used to display the randomized plan. The following statements produce Output 55.3.1 and Output 55.3.2:

```
   title 'Generalized Cyclic Block Design';
   proc plan seed=33373;
      treatments trtmt=8 of 52 cyclic (1 2 3 4 32 43 46 49) 4;
      factors block=13 plot=8;
      output out=c;
   quit;
options nocenter;
   proc sort data=c;
   by Block plot;
   proc print data=c;
   run;
```

Obs	block	plot	trtmt
1	1	1	33
2	1	2	34
3	1	3	26
4	1	4	29
5	1	5	12
6	1	6	23
7	1	7	35
8	1	8	36
[…usw…]			
101	13	5	46
102	13	6	38
103	13	7	41
104	13	8	48

6. Korrelation und Regression

6.1. PROC CORR

Zur Berechnung von Korrelationen steht die Prozedur **PROC CORR** zur Verfügung.
Datenbeispiel: Von einem Feld auf dem Ihinger Hof wurden an 80 Positionen in einem 2 x 2 Meter Raster 80 Bodenproben entnommen (Daten von S. Graeff, Inst. 340). Im Labor wurde der Nitrat-, Kalium, Magnesium- und Phosphorgehalt bestimmt. Bestehen zwischen den Nährstoffgehalten Abhängigkeiten? Die Daten sind in der Datei Boden.dat abgelegt.

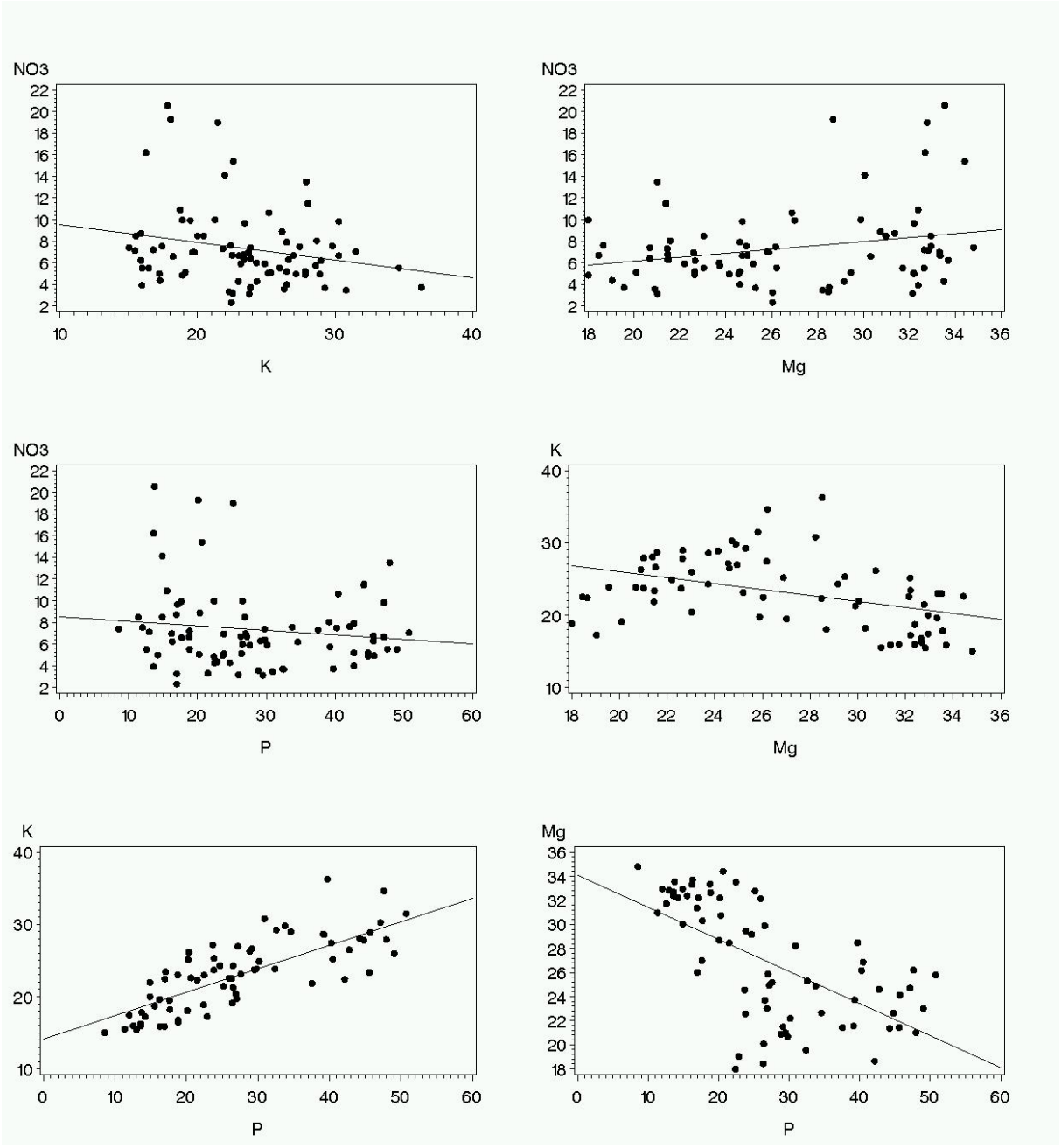
```
Data Boden;
input NO3 K Mg P;
datalines;
10.635  25.2    26.89   40.495
7.313   21.85   21.445  37.56
11.441  28.055  21.385  44.27
13.51   27.915  21.02   47.995
4.949   28.9    24.145  45.725
4.975   27.18   24.565  23.695
3.571   26.315  20.9    28.86
3.708   23.86   19.56   32.41
6.3     26.64   21.505  29.165
6.014   24.3    23.725  26.545
14.132  21.99   30.065  14.905
7.412   15.035  34.82   8.57
3.915   15.965  32.4    13.6
8.505   20.02   32.96   14.905
5.508   16.46   32.665  18.855
10.909  18.735  32.4    15.56
4.292   24.315  29.175  24.7
4.288   22.985  33.525  22.47
2.334   22.47   26.04   17
3.68    29.265  25.3    32.57
6.769   23.38   21.465  45.625
6.291   23.38   21.465  45.625
11.552  28.055  21.385  44.27
7.621   22.42   18.655  42.16
7.506   27.445  26.185  40.28
6.207   28.995  22.66   34.59
7.409   23.865  20.69   29.78
6.398   23.865  20.69   29.78
7.566   29.81   24.895  33.785
5.55    34.67   26.22   47.655
9.681   23.44   32.22   17.105
8.493   15.52   30.99   11.365
5.51    15.985  31.73   12.6
4.996   17.23   32.22   14.23
19.009  21.48   32.795  25.2
6.974   19.635  33.33   16.25
6.659   23.025  33.36   18.79
8.887   26.16   30.76   20.345
3.738   36.295  28.51   39.74
8.05    28.685  21.575  39.165
;
run;

Proc corr data=boden;
var NO3 K Mg P;
run;
```

Pearson Correlation Coefficients, N = 80
Prob > |r| under H0: Rho=0

	NO3	K	Mg	P
NO3	1.00000	-0.21769	0.24683	-0.13339
		0.0524	0.0273	0.2382
K	-0.21769	1.00000	-0.42024	0.78182
	0.0524		0.0001	<.0001
Mg	0.24683	-0.42024	1.00000	-0.62685
	0.0273	0.0001		<.0001
P	-0.13339	0.78182	-0.62685	1.00000
	0.2382	<.0001	<.0001	

Die erste Zahl gibt jeweils den Korrelationskoeffizienten also die Stärke des Zusammenhangs an, die Zahl darunter gibt an, ob die Korrelation signifikant ist. Man sieht, dass bei 80 Wertepaaren auch eine Korrelation von r=0.25 signifikant sein kann.
Die Korrelationsrechnung prüft jeweils, ob ein linearer Zusammenhang besteht. Dieses ist aus den Korrelationskoeffizienten nicht ersichtlich. Es empfiehlt sich deshalb, immer auch einen Plot der Punktewolken zu machen.

```
Proc gplot data=boden;
symbol value=dot h=1 i=rl;
plot NO3*K NO3*Mg NO3*P K*Mg K*P Mg*P;
run; quit;
```



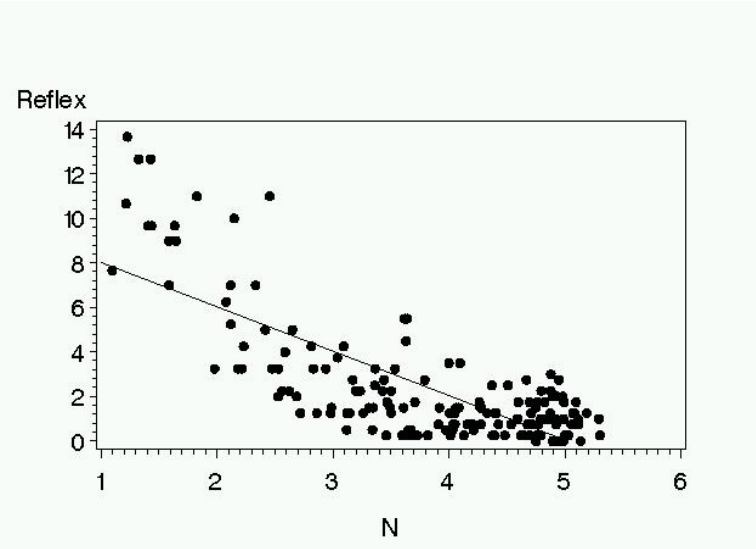
Neben der Maßkorrelation können auch die parameterfreien Zusammenhangsmaße “Kendalls Tau” und „Spearman-Rang-Korrelation“ berechnet werden.

```
Proc Corr data=Boden pearson spearman kendall;
var K P;
run;
```

6.2. Regressionen: PROC REG

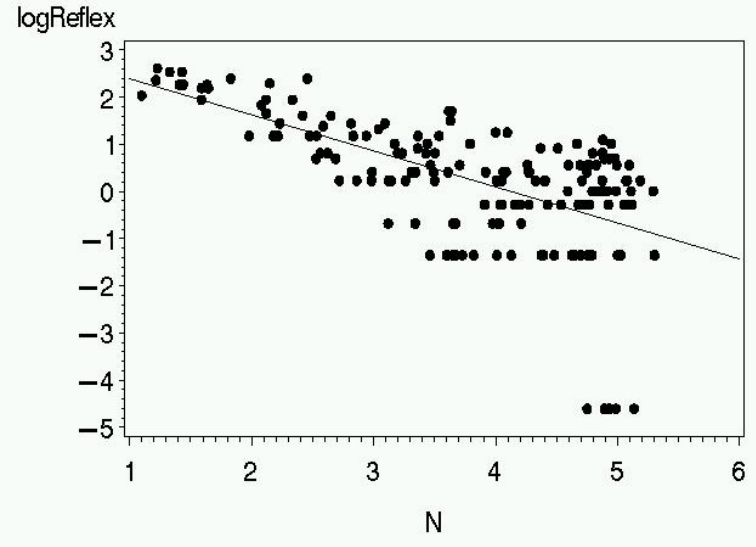
Im Gegensatz zur Korrelation ist bei der Regression die Richtung des Zusammenhangs zu definieren. Beispiel: In dem im Kapitel 6.1 vorgestellten Versuch wurden neben den Bodennährstoffen auch an verschiedenen Positionen der Stickstoffgehalt von Weizenpflanzen und die Reflexion in einem bestimmten Wellenlängenbereich gemessen (Daten unter Reflexion.dat). Zunächst fertigen wir einen Plot der Daten an.

```
data reflexion;
input N    Reflex;
datalines;
1.1        7.6667
1.218      10.6667
1.23       13.6667
[...]
5.303      0.25
;
run;

Proc gplot data=reflexion;
symbol value=dot h=1 i=rl;
plot Reflex*N;
run;quit;
```



Mit abnehmendem Stickstoffgehalt nimmt die Reflexion zu. Allerdings scheint der Zusammenhang keinesfalls linear zu sein. Falls die Beziehung einer Exponentialfunktion folgt, so würde eine Logarithmierung zu einem linearen Verhalten führen. Im vorliegenden Fall tritt das Problem auf, dass teilweise eine Reflexion von 0 gemessen wurde, was nicht logarithmierbar ist. Eine Option in solchen Fällen ist die Addition eines kleinen Wertes (z.B. 0.01).

```
Data reflexion;
set reflexion;
logReflex = log(Reflex+0.01);
run;
Proc gplot data=reflexion;
symbol value=dot h=1 i=rl;
plot logReflex*N;
run;quit;
```



Nun erhalten wir eine nahezu lineare Beziehung, lediglich „gestört“ durch die vorherigen Nullwerte. Mit der Prozedur **PROC REG** können wir nun den Regressionskoeffizienten und das Bestimmtheitsmaß ermitteln. Innerhalb der Prozedur REG steht eine Vielzahl an Anweisungen und Optionen zur Verfügung. Diese alle zu erläutern würde hier den Rahmen sprengen. Mindestens anzugeben sind folgende Statements:

PROC REG < options > **;**
MODEL dependents=<regressors> < / options > **;**
VAR variables **;**

In unserem Falle reicht folgender Code:

```
Proc REG data=reflexion;
model logReflex = N;
run;
```

```
The REG Procedure
Model: MODEL1
Dependent Variable: logReflex

                          Analysis of Variance

                                    Sum of           Mean
Source                     DF       Squares         Square    F Value    Pr > F
Model                       1     108.72496      108.72496      95.14    <.0001
Error                     156     178.27003        1.14276
Corrected Total           157     286.99499

Root MSE              1.06900    R-Square     0.3788
Dependent Mean        0.26927    Adj R-Sq     0.3749
Coeff Var           396.99787

                       Parameter Estimates
                    Parameter       Standard
Variable     DF      Estimate          Error    t Value    Pr > |t|
Intercept     1       3.14674        0.30701      10.25      <.0001
N             1      -0.76350        0.07827      -9.75      <.0001
```

Das komplette Modell ist signifikant (Globaler Test, F-Wert = 95.14 mit p-Wert < 0.0001). Auch der Achsenabschnitt ist signifikant von 0 verschieden, was mit einem t-Test geprüft wird. Das Bestimmtheitsmaß ist R^2=0.38 und die Regressionsfunktion lautet:

$$\log(\text{Reflexion}) = 3.147 - 0.7635\ N$$

Um auf die Beziehung auf der Originalskala zu kommen, können wir die Regressionsgleichung rück-transformieren. Unter Berücksichtigung der Addition von 0.01 erhalten wir:

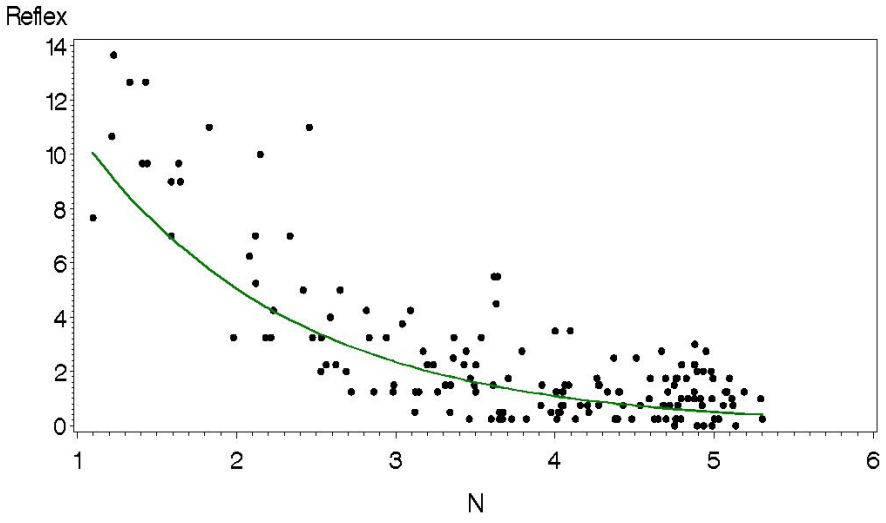
$$\log(\text{Reflexion} + 0.01) = 3.147 - 0.7635 * N$$

$$\Rightarrow \quad \text{Reflexion} + 0.01 = \exp(3.147) * \exp(-0.7635 * N)$$

$$\Rightarrow \quad \text{Reflexion} = - 0.01 + 23.27 * \exp(-0.7635 * N)$$

Diese Funktion können wir mit PROC GPLOT in die Ausgangsdaten zeichnen lassen.

```
Data reflexion;
set reflexion;
Funktion = - 0.01 + 23.27 * exp(-0.7635 * N);
run;
goptions reset=all;
goptions ftext=swiss htext=1.5;
run;
Proc gplot data=reflexion;
plot Reflex*N Funktion*N / overlay;
symbol1 i=none value=dot c=black h=1;
symbol2 i=join value=none c=green w=3;
run;quit;
```



Wir sehen, dass die angepasste Funktion scheinbar zu tief liegt. Mögliche Ursache hierfür ist evtl. die notwendige Addition eines „kleinen Wertes“ vor der Logarithmierung, wodurch das Ergebnis der Regressionsanalyse nicht unerheblich beeinflusst wurde.

Im Kapitel zur Prozedur NLIN wird eine alternative Auswertung vorgestellt, die die nichtlineare Funktion direkt modelliert.

6.2.1. Multiple Regression

Bei multipler Regression (mehr als eine Einflussvariable) stellt sich häufig die Frage der Modellwahl. Eine wichtige Option in der Model-Zeile ist deshalb **SELECTION=*name*** (mit **name** = *FORWARD (or F), BACKWARD (or B), STEPWISE, MAXR, MINR, RSQUARE, ADJRSQ, CP,* oder *NONE* (volles Modell).
Bei FORWARD wird zunächst ein Regressor ins Modell genommen und dann ein zweiter und so weiter. Bei BACKWARD wird zunächst ein Modell mit allen Regressoren angepasst und dann schrittweise vereinfacht. Die Methode „CP" sucht ein Modell, wo Mallows CP minimiert wird. Dieses entspricht einem Modell mit dem geringsten Vorhersagefehler für neue Daten. Das Adjustierte R^2 favorisiert dagegen das beste Modell, für die gegebenen Daten.
Beispiel:

```
Proc Reg;
model y = a b c/selection=backward;
run;
```

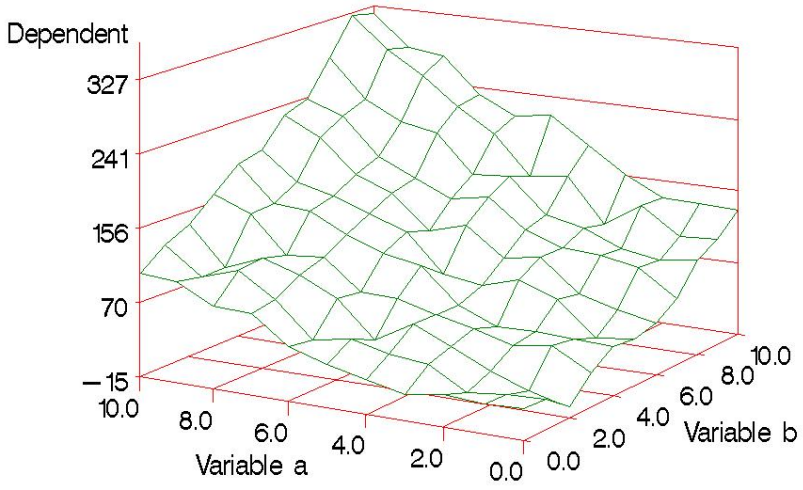
Angenommen wir beobachten Daten die auf das folgende Modell zurückgehen (hier wurden auf Basis eines Modells Daten simuliert):

```
data multreg;
do a = 0 to 10 by 1;
      do b = 0 to 10 by 1;
            do c = 0 to 10 by 1;
            y = a + b + c + a*b + a**2 + b**2 + 10*rannor(-1);
            output;
            end;
      end;
end;
run;

data multreg;
set multreg;
aa = a**2;
bb = b**2;
cc = c**2;
ab = a*b;
ac = a*c;
bc = b*c;
aab = a*a*b;
aac = a*a*c;
abb = a*b*b;
abc = a*b*c;
bbc = b*b*c;
bcc = b*c*c;
run;
```

Angenommen wir wissen, dass die Variablen a, b und c Einfluss auf die abhängige Variable y haben, wir wissen aber nicht genau, wie die Funktion aussieht. Hier hilft die Modellselektion.

```
proc reg data=multreg;
model y = a b c aa bb cc ab ac bc aab aac abc bbc bcc
      /selection = cp ADJRSQ;
run;
```

Das „wahre" Modell wird in diesem Fall tatsächlich als das zweitbeste erkannt!




```
                                The REG Procedure
                                  Model: MODEL1
                              Dependent Variable: y

                                C(p) Selection Method

                         Number of Observations Read        1331
                         Number of Observations Used        1331

Number in                                  Adjusted
  Model          C(p)    R-Square         R-Square   Variables in Model

      7        3.4734      0.9810           0.9809   a b c aa bb ab abc
      6        3.7362      0.9809           0.9808   a b c aa bb ab
      7        3.9666      0.9810           0.9809   a b c aa bb ab bcc
      8        4.3230      0.9810           0.9809   a b c aa bb cc ab abc
      7        4.5467      0.9810           0.9809   a b aa bb cc ab abc
      7        4.5858      0.9810           0.9809   a b c aa bb cc ab
      7        4.6600      0.9810           0.9809   a b c aa bb ab bc
      8        4.6949      0.9810           0.9809   a b c aa bb ab aac abc
      7        4.8309      0.9809           0.9808   a b c aa bb ab bbc
      8        4.8732      0.9810           0.9809   a b c aa bb ab ac abc
      8        5.1455      0.9810           0.9809   a b c aa bb ab abc bcc
      8        5.4305      0.9810           0.9809   a b c aa bb ab aab abc
      7        5.4614      0.9809           0.9808   a b aa bb cc ab bc
```

6.2.2. Regressionen mit der Prozedur GLM

Regressionen können auch mit der Prozedur GLM berechnet werden. Die Schätzwerte für die Regressionsparameter werden erhalten wenn man in der Model-Zeile die Option „Solution“ wählt (siehe auch nächstes Kapitel).

```
Proc Reg data=reflex;
model reflex=n_Gehalt;
run;

Proc GLM data=reflex;
model reflex=n_Gehalt/Solution;
run;
```

Die Verwendung von GLM empfiehlt sich besonders bei komplexeren Modellen und Polynomregression, da ein Modell

$Y = a + x + x*x$

in der Prozedur REG nicht ohne weiteres darstellbar ist (x^2 muss als neue Variable berechnet werden). Allerdings stehen in GLM nicht die Verfahren der Modellselektion zur Verfügung.

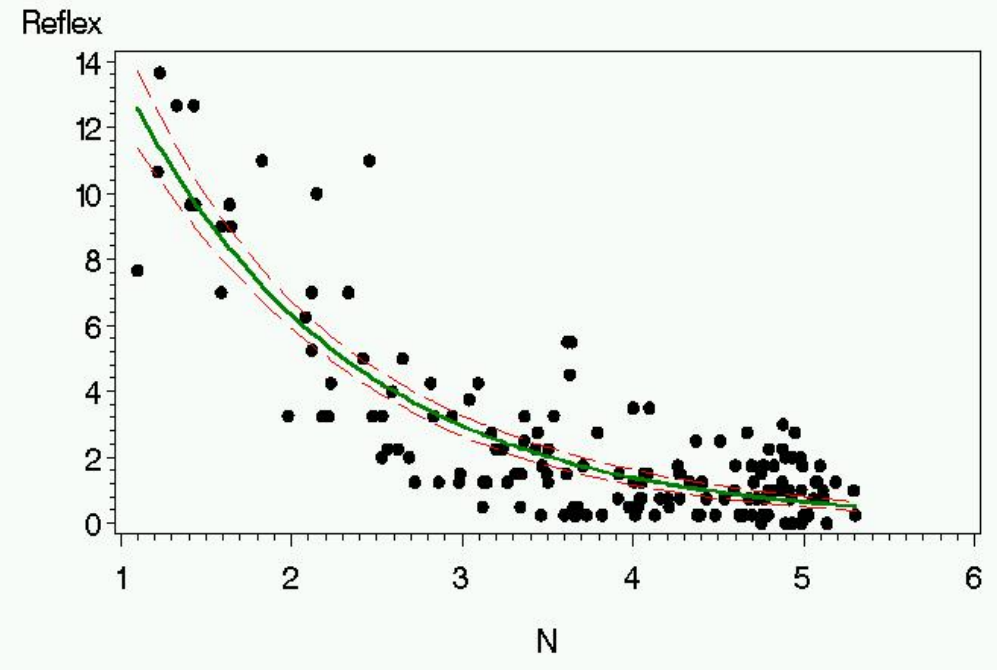
6.3. Nichtlineare Regression: PROC NLIN

Die Prozedur NLIN arbeitet iterativ. Das heißt schrittweise werden die Schätzwerte für die zu bestimmenden Parameter der Funktion so verändert, bis die Summe der Abweichungsquadrate, minimiert ist. Wichtig ist die Vorgabe von Startwerten. Je besser die Startwerte, desto sicherer konvergiert der Iterationsalgorithmus zur minimalen Fehlerquadratsumme. Im Beispiel mit den Reflexionsmessungen haben wir bereits Schätzwerte aus der Regressionsrechnung, die wir als Startwerte verwenden können. Im folgenden SAS-Code wird auch gezeigt, wie man sich Schätzungen für ein 95%-Konfidenzintervall ausgeben lassen kann und wie dieses in eine Abbildung übertragen wird.
In der Zeile „parms“ werden die Startwerte definiert. In der Model-Zeile das zu schätzende nichtlineare Modell. Es können auch Grenzen vorgegeben werden. Dann erfolgt die Schätzung unter Nebenbedingungen. In diesem Datenbeispiel könnte man postulieren, dass der Achsenabschnitt größer Null sein muss.

```
/*Funktion mit PROC NLIN schätzen*/
Proc Nlin data=reflexion;
parms int=23.27 beta=-0.7635;
bounds int > 0;
model reflex = int * exp(beta*N);
output out=funktion predicted=predicted U95M=U95M L95M=L95M;
run;
goptions reset=all;
goptions ftext=swiss htext=1.5;
run;
Proc gplot data=funktion;
plot Reflex*N Predicted*N L95M*N U95M*N/ overlay;
symbol1 i=none value=dot c=black h=1;
symbol2 i=join value=none c=green w=3;
symbol3 i=join value=none c=red w=0.5 line=2;
symbol4 i=join value=none c=red w=0.5 line=2;
label estimate=TM [dt/ha];
run;quit;
```

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	2	1923.7	961.9	462.97	<.0001
Error	156	324.1	2.0776		
Uncorrected Total	158	2247.8			

Parameter	Estimate	Std Error	Approximate 95% Confidence Limits	
int	28.9728	2.4178	24.1968	33.7487
beta	-0.7613	0.0394	-0.8391	-0.6836


Reflex
14
12
10
8
6
4
2
0
1
2
3
4
5
6
N

7. PROC GLM

In den ersten Kapiteln wurde ein Sortenversuch in Blockanlage vorgestellt. Ertragsunterschiede zwischen den Sorten kann man mittels Varianzanalyse auf Signifikanz prüfen. Hierfür stehen in SAS drei Prozeduren zur Verfügung: PROC ANOVA, PROC GLM und PROC MIXED. ANOVA ist nicht empfehlenswert, MIXED besprechen wir später.
Mit der Prozedur GLM können Varianzanalysen, Regressionen und Kovarianzanalysen gerechnet werden. Der folgende Code erlaubt uns eine einfache Varianzanalyse und multiplen Mittelwertvergleich nach t-Test.

```
Data Versuch;
Input Sorte$ Wdh Ertrag;
cards;
A     1     21
A     2     22
A     3     19
A     4     18
B     1     18
B     2     16
B     3     15
B     4     13
C     1     19
C     2     19
C     3     16
C     4     14
D     1     14
D     2     13
D     3     12
D     4     11
;
run;

options linesize=85 nocenter;

      ods rtf body = 'body.rtf';
proc glm data=versuch;
class Sorte Wdh;
model Ertrag = Sorte Wdh;
means Sorte / t;
run;
      ods rtf close;
```

Wir erhalten folgenden Output. Der F-Test ist ebenso wie alle Einzelvergleiche signifikant:

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	158.0000000	26.3333333	47.40	<.0001
Error	9	5.0000000	0.5555556		
Corrected Total	15	163.0000000			

R-Square	Coeff Var	Root MSE	Ertrag Mean
0.969325	4.586806	0.745356	16.25000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Sorte	3	117.0000000	39.0000000	70.20	<.0001
Wdh	3	41.0000000	13.6666667	24.60	0.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Sorte	3	117.0000000	39.0000000	70.20	<.0001
Wdh	3	41.0000000	13.6666667	24.60	0.0001

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	9
Error Mean Square	0.555556
Critical Value of t	2.26216
Least Significant Difference	1.1923

Means with the same letter are not significantly different.			
t Grouping	Mean	N	Sorte
A	20.0000	4	A
B	17.0000	4	C
C	15.5000	4	B
D	12.5000	4	D

Die Programmzeilen im Einzelnen:

PROC GLM < options > **;**

Der Prozeduraufruf. Er steht an erster Stelle. Folgende Optionen sind unter anderem verfügbar:

ALPHA=p	Unter ALPHA kann das Niveau für den Fehler erster Art definiert werden.
DATA=SAS-data-set	Hier kann der Datensatz spezifiziert werden, der ausgewertet werden soll.
NOPRINT	Die Ergebnisse werden nicht auf den Bildschirm geschrieben. Kann sinnvoll sein falls OUTSTAT genutzt wird.
OUTSTAT=SAS-data-set	Die Ergebnisse werden in eine separate SAS-Datei geschrieben (und können später z.B. in EXCEL exportiert werden)

CLASS variables **;**

Hier sind die Klassifizierungsvariablen zu nennen. Dieses sind die **unabhängigen Variablen**, die nominales Niveau haben bzw. als Variablen mit nominalem Niveau betrachtet werden (z.B. Sortennamen, Ortsnummern, Herbizide).

MODEL dependents=independents < / options > **;**

In der Model-Zeile sind die abhängigen Variablen vor einem Gleichheitszeichen und die unabhängigen Variablen nach dem Gleichheitszeichen aufzuführen. Wie die Name bereits sagt, ist hier das „Modell" zu spezifizieren. Angenommen wir möchten einen Sortenversuch als Blockanlage auswerten. Die abhängige Variable sei der Ertrag, die unabhängigen Variablen sind die Sorte und der Block. Wir postulieren, dass Sorte und Block unabhängig voneinander den Ertrag einer Parzellen beeinflussen. Dieses Modell kann folgendermaßen beschrieben werden:

$$y_{ij} = \mu + \tau_i + \beta_j + e_{ij}$$

y_{ij} Ertrag der Parzelle mit i-ter Sorte im j-ten Block
μ Allgemeiner Mittelwert
τ_i Effekt der i-ten Sorte
β_j Effekt des j-ten Blocks
e_{ij} Fehler der Parzelle mit i-ter Sorte im j-ten Block ~$N(0,\sigma^2_e)$

Bei der Beschreibung des Modells in SAS wird der allgemeine Mittelwert und der Restfehler automatisch berücksichtigt. Der User muss also in diesem Fall nur noch den Sorten- und den Blockeffekt aufführen.

$y_{ij} = \mu + \tau_i + \beta_j + e_{ij}$ $\Rightarrow$ **MODEL** Ertrag = Sorte Block ;

Haupteffekte werden einfach der Reihe nach aufgeführt $\Rightarrow$ **A B C**.

Interaktionen werden mit einem Stern verknüpft. Z.B. Interaktion zwischen A und B $\Rightarrow$ **A*B**.

Geschachtelte Effekte[1] (keine Kreuzklassifikation), z.B. „B geschachtelt in A" $\Rightarrow$ **B(A)**

Der „Bar-Operator" ist eine SAS-Kurzschreibweise. Wenn man sehr viele Effekte im Modell hat, kann eine Abkürzung hilfreich sein.
A | B | C = { A | B } | C = { A B A*B } | C $\Rightarrow$ **A B A*B A*C B*C A*B*C**

Durch ein @ kann gesteuert werden, wie viele Effekte maximal in einer Interaktion enthalten sein dürfen. So wird im folgenden Beispiel die Dreifach-Interaktion A*B*C ignoriert.
A | B | C @2 $\Rightarrow$ **A B A*B A*C B*C**

[1] Ein Beispiel für fehlende Kreuzklassifikation sind z.B. die Wiederholungen einer Serie von Blockanlagen. An jedem Versuchsort gibt es eine Wiederholung 1. Diese haben jedoch nichts miteinander zu tun. Evtl. weist Wdh. 1 an Ort 1 einen besonders guten Boden auf, an Ort 2 einen ganz schlechten. Die Wiederholungen sind geschachtelt (engl. „nested") innerhalb der Orte. Demgegenüber sind z.B. Sorten kreuzklassifiziert. Sorte 1 bedeutet an allen Orten den gleichen Genotyp.

In der Model-Zeile können nach einem Schrägstrich unter anderem folgende Optionen spezifiziert werden:

ALPHA=p	es kann erneut das Niveau für den Alpha-Fehler festgelegt werden
CLI, CLM, CLPARM	Konfidenzintervalle für Einzelwerte, Mittelwerte und Parameter
INTERCEPT	es wird ein Signifikanztest für die Nullhypothese $\mu = 0$ berechnet
NOINT	es wird ein Modell ohne Interaktion angepasst
P	Residuen und Erwartungswerte werden ausgegeben
SOLUTION	Schätzwerte für die Parameter (die Effekte) werden ausgegeben
SS1 SS2 SS3 SS4	Wahl der Quadratsummenzerlegung; **sehr wichtig!**

Eine **Type I Analyse** (SS1) beinhaltet die Anpassung einer Sequenz von Modellen. Dabei beginnt man beim einfachsten Modell, das nur einen allgemeinen Term hat (Achsenabschnitt, intercept). Danach werden sukzessive weitere Terme in das Modell aufgenommen. Für jeden zusätzlichen Term kann in der Varianzanalyse eine F-Statistik berechnet werden.
Bei einer Type I Analyse hängt das Ergebnis des Tests von der Reihenfolge ab, in der Terme in das Modell aufgenommen werden. In der GLM Prozedur entspricht die Reihenfolge der Sequenz genau der Reihenfolge der Terme in der MODEL Anweisung. Man muss nicht selber die einzelnen Modelle der Sequenz angeben - das erledigt GLM automatisch. Die Reihenfolge kann allerdings nicht beliebig gewählt werden. Es muss, falls vorhanden, die Hierarchie der Modelleffekte berücksichtigt werden. So müssen immer die Haupteffekte vor den Interaktionen in das Modell aufgenommen werden. Die Haupteffekte sind gewissermaßen den Interaktionen übergeordnet. Ebenso ist es bei Polynomen sinnvoll, die Reihenfolge x, x^2, x^3 einzuhalten. Eine weitere zu berücksichtigende Regel ist die folgende: Wird ein allgemeiner Regressionsterm βx_j und ein gruppenspezifischer Regressionsterm $\gamma_i x_j$ angepasst, so muss der allgemeine Term vor dem gruppenspezifischen Term angepasst werden, um die „richtigen“ Tests zu bekommen.

Bei der **Type III Analyse** (SS3), werden die Teststatistiken anders berechnet: Es werden immer alle anderen Effekte des Modells „herausgerechnet“, d.h. ausgeschaltet. Außerdem arbeitet die Type III Analyse mit bestimmten Parameterrestriktionen. Einzelheiten können hier nicht erläutert werden (siehe SAS/STAT User’s Guide, Version 6, Forth Edition, Kapitel 9 sowie der Abschnitt zu PROC GLM zu Einzelheiten der Definition von Type III Sums of Squares). Bei der Type III Analyse muss man sich um die genaue Reihenfolge der Modelleffekte keine Gedanken machen, da bei der Berechnung von Teststatistiken immer alle jeweils anderen Effekte ausgeschaltet werden.

In manchen Fällen stimmen die Ergebnisse von Type I und Type III Analyse auch unabhängig von der Reihenfolge der Effekte im Modell genau überein, z.B. bei balancierten Daten in der Varianzanalyse. Type I Tests haben bei unbalancierten Daten dagegen oft eine bessere Teststärke als Type III. Bei Kovarianzanalysen sollte man immer Type I verwenden!

MEANS effects < / options > **;**
Es werden arithmetische Mittelwerte berechnet. Dieses ist nur zu empfehlen wenn keine fehlenden Werte vorliegen und die Daten balanciert sind. Für Kombinationen von Faktoren können zwar Mittelwerte berechnet werden, jedoch ist ein statistischer Test von Mittelwertdifferenzen („Grenzdifferenz“) nicht möglich. Nach einem Schrägstrich kann das Verfahren für Multiple Vergleich gewählt werden. Die einfachste Wahl ist die Option „T“ oder „LSD“, die einen multiplen t-Test anweist. Bei sehr vielen Versuchsgliedern und infolgedessen vielen Vergleichen wird das globale Niveau für den Alpha-Fehler allerdings nicht mehr eingehalten

(global = bestehen Unterschiede zwischen irgend zwei Prüfglieder?). Wenn alle Vergleiche von Interesse sind, man den globalen Alpha-Fehler kontrollieren will und man eine allgemeingültige Grenzdifferenz haben möchte, so ist Tukeys-HSD-Test („TUKEY“) das Verfahren der Wahl. REGWQ sorgt für einen sequentiellen Test nach Ryan-Einot-Gabriel-Welch. Der Test hat einen geringeren Beta-Fehler als Tukey, aber keine einheitliche Grenzdifferenz.

Beispiele:

Mittelwertdifferenzen mit Tukey-Test	`MEANS Sorte / TUKEY;`
Test gegen Kontrolle (DUNNETT)	`MEANS Sorte / DUNNETT;`
DUNNETT-Test mit Sorte 5 als Kontrolle	`MEANS Sorte / DUNNETT ('5');`

Problem: Wenn Daten unbalanciert sind, wird keine einheitliche Grenzdifferenz berechnet.
Tipp: Dass kann mit der zusätzlichen Option „**LINES**“ erzwungen werden.
z.B.: `MEANS Sorte / TUKEY LINES;`

Problem: Bei unbalancierten Daten sind besser LSMeans als arithmetische Mittelwerte zu verwenden. LSMeans liefert jedoch keine Grenzdifferenzen.
Tipp1: Mittelwerte über LSMeans berechnen und Grenzdifferenzen aus MEANS-Statement holen.
Tipp2: mittels „LSMEANS / pdiff clm“ können Konfidenzintervalle für Vergleiche berechnet werden. Die halbe breite eine Konfidenzintervalls entspricht der Grenzdifferenz.

LSMEANS *effects* < **/** *options* >

Es werden Mittelwerte nach der Methode der Kleinsten Quadrate (**L**east **S**quares) berechnet. Diese sind einfachen arithmetischen Mittelwerten (means-Statement) fast immer vorzuziehen. Weiterer Vorteil von LSMeans ist es, dass auch Mittelwertvergleiche für Interaktionen möglich sind, während „means“ dieses nur für Haupteffekte erlaubt. Alle Effekte für die LSMeans angefordert werden, müssen im CLASS-Statement genannt worden sein.

```
proc glm;
class A B;
model Y=A B A*B;
lsmeans A B A*B;
run;
```

Folgende Optionen können hinter einem Schrägstrich angegeben werden:

ADJUST= *method*	ALPHA=*p*	AT *variable* = *value*
BYLEVEL	CL	COV
E	E=*effect*	ETYPE=*n*
NOPRINT	OBSMARGINS	OUT=*SAS-data-set*
PDIFF<=*difftype*>	SLICE = *fixed-effect*	SINGULAR=*number*
STDERR	TDIFF	

Bei „**ADJUST**“ kann die Art des Tests für Multiple Mittelwertvergleiche bestimmt werden, wenn diese über „**PDIFF**“ angefordert werden. Default-Einstellung ist „ADJUST=T“, was einem einfachen t-Test entspricht. Adjust=Tukey bewirkt einen Tukey-Test. Insgesamt stehen zur Verfügung:

```
ADJUST=BON
ADJUST=DUNNETT
ADJUST=SCHEFFE
ADJUST=SIDAK
ADJUST=SIMULATE <(simoptions)>
ADJUST=SMM | GT2
ADJUST=TUKEY
ADJUST=T
```

„**NOPRINT**“ unterdrückt die Ausgabe der Mittelwerte und Tests auf den Bildschirm. Mit „**PDIFF**“ bekommt man die jeweilige Irrtumswahrscheinlichkeit für den Vergleich von Mittelwerten. „**STDERR**“ bewirkt die Ausgabe von Standardfehlern für die Mittelwerte. Über „**ALPHA**“ kann das Niveau für den Fehler erster Art spezifiziert werden. „**CL**“ erzeugt Vertrauensintervalle für Mittelwerte und Mittelwertdifferenzen (bei gleichzeitiger Angabe von „PDIFF“).

7.1. Zweifaktorielle Varianzanalyse mit PROC GLM

Wir kommen zurück auf den Versuch zur Wirkung der N-Düngung auf den Weizenertrag (siehe Kapitel 4). In Kapitel 4 hatten wir nur drei Düngungsstufen betrachtet, nun wollen wir alle vier Stufen auswerten (Daten im Internet unter „DuengungWeizen.dat“). Die Sorte hat nominales Skalenniveau und kommt deshalb in das Class-Statement. Die N-Menge kann entweder als quantitativ oder als kategorial (=qualitativ) aufgefasst werden.

Stickstoffdüngung als qualitative Variable behandeln

Der Versuch wurde als eine zweifaktorielle Blockanlage durchgeführt. Es ist deshalb sinnvoll einen Blockeffekt in das Modell zu integrieren. Damit können wir für die Daten folgendes Modell postulieren:

$$y_{ijk} = \mu + \alpha_i + \tau_j + \beta_k + \alpha\tau_{ij} + e_{ijk}$$

y_{ij}	Ertrag der Parzelle mit i-ter Sorte und j-ter Düngung im k-ten Block
μ	Allgemeiner Mittelwert
α_i	Effekt der i-ten Sorte
τ_j	Effekt der j-ten Düngung
β_k	Effekt des k-ten Blocks
$\alpha\tau_{ij}$	Interaktion zwischen i-ter Sorte und j-ter Düngung
e_{ijk}	Fehler der ijk-ten Parzelle ~N(0,σ^2_e)

Der entsprechende GLM-Code lautet:

```
Data Weizen;
input Wdh$ Sorte$ N$ Parz FM AnzHalme FMKoerner;
cards;
W1    S1    N4    1     1677  162   473.82
W1    S2    N1    2     1743  204   554.85
[…]
W4    S1    N1    48    825   102   216
;
run;
```

```
/*Datenmanagement*/
data Weizen; set Weizen;
If Sorte = "S1" then Sortenname="Monopol";
If Sorte = "S2" then Sortenname="Batis";
If Sorte = "S3" then Sortenname="Hybnos";
If N = "N1" then N_Menge=0;
If N = "N2" then N_Menge=80;
If N = "N3" then N_Menge=160;
If N = "N4" then N_Menge=240;
run;

/*Varianzanalyse*/
Proc glm data=Weizen;
class Sortenname N Wdh;
model FM = Wdh Sortenname N Sortenname*N / SS3;
run;
```

Dependent Variable: FM

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	14	4679936.792	334281.199	7.32	<.0001
Error	33	1507770.688	45690.021		
Corrected Total	47	6187707.479			

R-Square	Coeff Var	Root MSE	FM Mean
0.756328	13.78398	213.7522	1550.729

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Wdh	3	336525.062	112175.021	2.46	0.0804
Sortenname	2	433779.292	216889.646	4.75	0.0154
N	3	3519518.563	1173172.854	25.68	<.0001
Sortenname*N	6	390113.875	65018.979	1.42	0.2355

Unter Type III SS sehen wir die Summen der Abweichungsquadrate. SS dividiert durch die Freiheitsgrade (DF) ergibt die Mean Square. Die F-Werte sind jeweils der Quotient aus dem Mean Square eines Effekts und dem Mean Square des Fehlers, den wir im oberen Teil der Tabelle sehen. Unter Pr>F ist die jeweilige Überschreitungswahrscheinlichkeit angegeben. Ein F-Wert von 2.46 kann bei 3 Zähler- und 33 Nenner-Freiheitsgraden noch mit 8%iger Wahrscheinlichkeit auftreten, auch wenn die Nullhypothese „Keine Blockeffekte“ gültig ist. Ein F-Wert von 4.75 tritt bei 2 Zähler- und 33 Nennerfreiheitsgraden dagegen nur in 1,5% der Fälle auf. Der Sorteneffekt kann deshalb ebenso wie der Düngungseffekt als signifikant bezeichnet werden. Die Interaktion zwischen Sorte und Düngung ist nicht signifikant.
Aufgrund der fehlenden Interaktion ist es sinnvoll, die Randmittelwerte der Sorten und Düngungsstufen zu vergleichen und Grenzdifferenzen zu berechnen. Hierfür kann das Statement „means“ genutzt werden.

```
/*Varianzanalyse und Mittelwertvergleiche nach Tukey*/
Proc glm data=Weizen;
class Sortenname N Wdh;
model FM = Wdh Sortenname N Sortenname*N / ss3;
means Sortenname N / Tukey;
run;
```

```
                Tukey's Studentized Range (HSD) Test for FM
NOTE: This test controls the Type I experimentwise error rate, but it generally
                has a higher Type II error rate than REGWQ.

                Alpha                                      0.05
                Error Degrees of Freedom                     33
                Error Mean Square                       45690.02
                Critical Value of Studentized Range      3.47019
                Minimum Significant Difference            185.44

          Means with the same letter are not significantly different.

           Tukey Grouping          Mean      N    Sortenname

                        A       1681.31     16    Hybnos
                        A
                   B    A       1513.13     16    Batis
                   B
                   B            1457.75     16    Monopol


Tukey's Studentized Range (HSD) Test for FM
NOTE: This test controls the Type I experimentwise error rate, but it generally
                has a higher Type II error rate than REGWQ.

                Alpha                                      0.05
                Error Degrees of Freedom                     33
                Error Mean Square                       45690.02
                Critical Value of Studentized Range      3.82537
                Minimum Significant Difference            236.04

           Means with the same letter are not significantly different.

            Tukey Grouping          Mean      N    N

                          A       1821.00     12    N4
                          A
                          A       1772.17     12    N3

                          B       1463.50     12    N2

                          C       1146.25     12    N1
```

7.2. Kovarianzanalyse mit PROC GLM

Stickstoffdüngung als quantitative Variable behandeln

Anstatt die Düngung als qualitativen Faktor in diskreten Stufen zu behandeln, kann er als quantitative Größe betrachtet werden. Das Verfahren ist dann als Kovarianzanalyse zu bezeichnen (Düngung = Kovariable). Die Verwendung der Typ3-Quadratsummenzerlegung kann bei einer Kovarianzanalyse zu falschen Resultaten führen, es sollte deshalb die Typ1-Quadratsummenzerlegung verwendet werden. Bei dieser ist auf die Reihenfolge der Effekte im Modell zu achten (siehe auch Kapitel KOVARIANZANALYSE im Skript *Biometrie* von H.-P. Piepho). Die Interaktion zwischen Sorte und N-Menge ist jeweils nach den Haupteffekten anzupassen. Für die beiden Haupteffekte sind zwei Reihenfolgen denkbar. Um den Sortenef-

fekt sauber zu testen, muss dieser nach der N-Menge angepasst werden. Der Test ist dann um den Effekt der N-Menge bereinigt. Andererseits muss man für einen Test der N-Menge, diese nach dem Effekt der Sorte anpassen. Somit muss zweimal eine Analyse mit PROC GLM durchgeführt werden!

```
/*Kovarianzanalyse*/
Proc glm data=Weizen;
class Sortenname Wdh;
model FM = Wdh Sortenname N_Menge Sortenname*N_Menge /ss1;
run;
Proc glm data=Weizen;
class Sortenname Wdh;
model FM = Wdh N_Menge Sortenname Sortenname*N_Menge /ss1;
run;
```

Sorte vor Düngung angepasst

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Wdh	3	336525.063	112175.021	2.29	0.0930
Sortenname	2	433779.292	216889.646	4.44	0.0184
N_Menge	1	3265500.104	3265500.104	66.78	<.0001
N_Menge*Sortenname	2	244965.058	122482.529	2.50	0.0947

Düngung vor Sorte angepasst

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Wdh	3	336525.063	112175.021	2.29	0.0930
N_Menge	1	3265500.104	3265500.104	66.78	<.0001
Sortenname	2	433779.292	216889.646	4.44	0.0184
N_Menge*Sortenname	2	244965.058	122482.529	2.50	0.0947

Die Reihenfolge der Effekte spielt bei diesem speziellen Beispiel keine Rolle. Sobald die Daten unbalanciert sind, was bei Kovarianzanalysen eher die Regel als die Ausnahme ist, ist die Reihenfolge jedoch sehr wichtig.

Merke: Bei Varianzanalyse mit unbalancierten Daten sowie bei Kovarianzanalyse generell Typ-1-Analyse wählen, sonst läuft man Gefahr, falsche Abweichungsquadrate und damit falsche F-Werte zu erhalten!

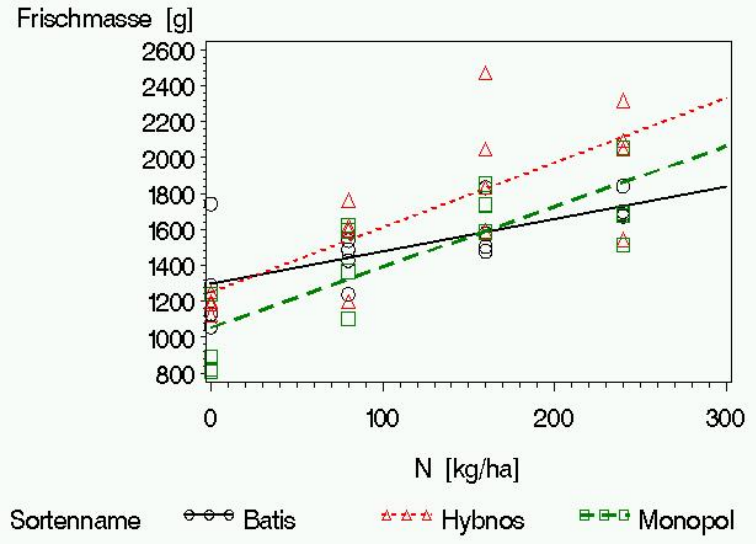
Weiteres Beispiel: Kovarianzanalyse für Futtermittel bei Schweinen
Siehe: http://www.uni-hohenheim.de/bioinformatik/lehre/module/saspraktikum/programme/

```
/*SAS Anweisungen für Kovarianzanalyse*/
options nocenter;
run;

data            1 54 1.28      2 42 1.24      3 42 1.22      4 51 1.48
schweine;       1 57 1.34      2 52 1.29      3 47 1.39      4 41 1.31
input trt       1 45 1.55      2 43 1.43      3 42 1.39      4 40 1.27
x y;            1 41 1.57      2 50 1.29      3 40 1.56      4 45 1.22
cards;          1 40 1.26      2 40 1.26      3 40 1.36      4 39 1.36
1 61 1.40       2 74 1.61      3 80 1.67      4 62 1.40      ;
1 59 1.79       2 75 1.31      3 61 1.41      4 55 1.47      run;
1 76 1.72       2 64 1.12      3 62 1.73      4 62 1.37
1 50 1.47       2 48 1.35      3 47 1.23      4 43 1.15
1 61 1.26       2 62 1.29      3 59 1.49      4 57 1.22

/*Test auf Parallelitaet, Kovariable vor Treatment angepasst*/
proc glm data=schweine;
class trt;
model y= x trt x*trt/solution;
run;

/*Test auf Parallelitaet, Kovariable nach Treatment angepasst*/
proc glm data=schweine;
class trt;
model y= trt x  x*trt/solution;
run;

/*Kovarianzanalyse mit paarweisen Mittelwertvergleichen*/
proc glm data=schweine;
class trt;
model y= x trt/solution;
lsmeans trt/pdiff;
run;

/*Modell mit Interaktion:
Kovarianzanalyse mit paarweisen Mittelwertvergleichen*/
proc glm data=schweine;
class trt;
model y= x trt x*trt/solution;
lsmeans trt/pdiff;
lsmeans trt/pdiff at x= 52.725;
run;

proc means;
var x;
run;

/*Modell mit Interaktion:
Kovarianzanalyse mit paarweisen Mittelwertvergleichen bei x=10*/
proc glm data=schweine;
class trt;
model y= x trt x*trt/solution;
lsmeans trt/pdiff at x=10;
run;
```

7.3. Polynome anpassen

Wir haben bislang bei den Daten der Düngung für Weizensorten eine lineare Beziehung zwischen der Kovariablen und der abhängigen Variablen vermutet, ohne zu überprüfen, ob dieses Modell zutreffend ist.
Es ist sinnvoll, sich zunächst einen Plot der Daten ausgeben zu lassen. Im folgenden SAS-Code werden den Sorten eigene Symbole und Farben zugewiesen.

```
/*Plot der Daten*/
goptions reset=all;
goptions ftext=swiss htext=1.5 ;
proc gplot data=Weizen;
symbol1 value=circle   h=1.5 i=rl cv=black line=1 w=2;
symbol2 value=triangle h=1.5 i=rl cv=red   line=2 w=2;
symbol3 value=square   h=1.5 i=rl cv=green line=3 w=2;
plot FM*N_Menge=Sortenname;
label FM = 'Frischmasse [g]';
label N_Menge = 'N [kg/ha]';
run;quit;
```



Für Polynome anstelle von Geraden können wir relativ einfach über PROC GPLOT eine Grafik erzeugen lassen. Hierzu müssen wir in GPLOT lediglich die Option „i=rl“ durch „i=rq“ ersetzen. Dann wird anstatt einer linearen, eine quadratische Trendlinie angepasst. Das Optimum scheint bei etwa 200 kg/ha zu liegen.

Lack-of-fit-Test
Wir können über einen F-Test prüfen, ob die Linearen Regressionen die Daten hinreichend beschreiben. Hierfür nehmen wir die Stickstoffstufen sowohl als quantitative als auch zusätzlich als qualitative Variable in das Modell. Der qualitative Teil modelliert die systematischen Abweichungen vom linearen Trend. Wichtig ist es, den zusätzlichen qualitativen Term auch in das Class-Statement aufzunehmen.

```
/*Lack of Fit Test*/
Proc glm data=Weizen;
class Sortenname Wdh N;
model FM = Wdh Sortenname N_Menge N Sortenname*N_Menge
Sortenname*N/ss1;
run;
```

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Wdh	3	336525.063	112175.021	2.46	0.0804
Sortenname	2	433779.292	216889.646	4.75	0.0154
N_Menge	1	3265500.104	3265500.104	71.47	<.0001
N (Lack_of_Fit)	2	254018.458	127009.229	2.78	0.0766
N_Menge*Sortenname	2	244965.058	122482.529	2.68	0.0834
Sortenname*N	4	145148.817	36287.204	0.79	0.5375

Bei den Beispieldaten ist der Lack-of-Fit-Term „N“ zwar nicht signifikant, aber ein p-Wert von 8% gibt Anlass zur Vermutung, dass eine lineare Regression allein evtl. nicht hinreichend ist. Wir erweitern daher unser Modell.

```
/*Beziehung nicht linear, passe Polynom an*/
Proc glm data=Weizen;
class Sortenname Wdh;
model FM = Wdh Sortenname N_Menge N_Menge*N_Menge Sortenname*N_Menge
 Sortenname*N_Menge*N_Menge/ss1;
run;
```

Der allgemeine quadratische Term ist signifikant, die sortenspezifische Abweichung Sorte*N*N allerdings nicht und kann auch wieder entfernt werden.

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Wdh	3	336525.063	112175.021	2.58	0.0689
Sortenname	2	433779.292	216889.646	4.98	0.0123
N_Menge	1	3265500.104	3265500.104	75.01	<.0001
N_Menge*N_Menge	1	216142.521	216142.521	4.96	0.0322
N_Menge*Sortenname	2	244965.058	122482.529	2.81	0.0732
N_Meng*N_Meng*Sorten	2	123477.042	61738.521	1.42	0.2554

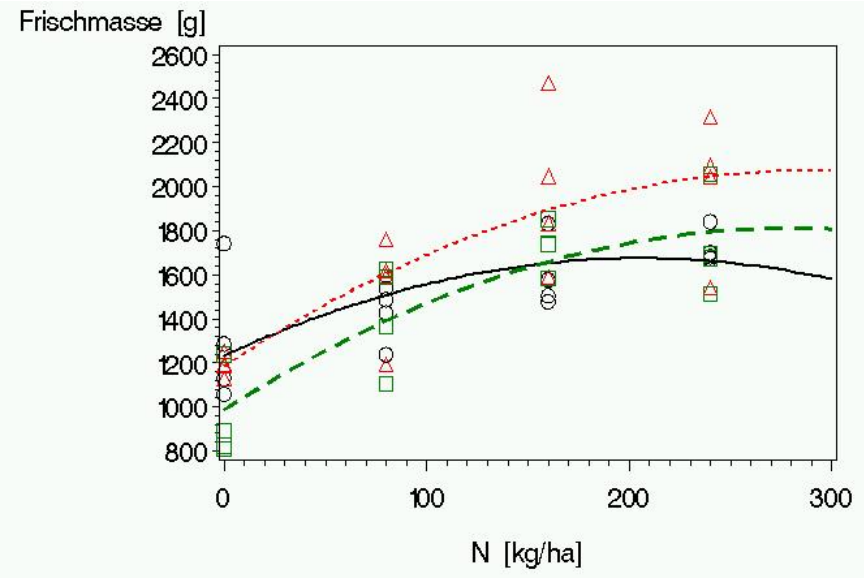
Wir können das Modell also abschließend vereinfachen und erhalten:

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Wdh	3	336525.063	112175.021	2.52	0.0723
Sortenname	2	433779.292	216889.646	4.87	0.0130
N_Menge	1	3265500.104	3265500.104	73.39	<.0001
N_Menge*Sortenname	2	244965.058	122482.529	2.75	0.0765
N_Menge*N_Menge	1	216142.521	216142.521	4.86	0.0337

Der SAS-Code für den Plot der Daten zu diesem Modell ist etwas komplexer. Zunächst einmal lassen wir uns im Zuge der Varianzanalyse die Erwartungswerte in eine Datei „work.pred“ ausgeben.

```
Proc glm data=Weizen;
class Sortenname Wdh;
model FM = Wdh Sortenname N_Menge Sortenname*N_Menge
N_Menge*N_Menge/ss1 solution;
output out=pred predicted=p_FM;
run;
quit;
```

```
/*Plot der Daten*/
data pred;
set pred;
if Sortenname = "Monopol" then x1=FM;
if Sortenname = "Batis" then x2=FM;
if Sortenname = "Hybnos" then x3=FM;
if Sortenname = "Monopol" then x4=p_FM;
if Sortenname = "Batis" then x5=p_FM;
if Sortenname = "Hybnos" then x6=p_FM;
run;
goptions reset=all;
goptions ftext=swiss htext=1.5 ;
proc gplot data=Pred;
symbol1 value=circle   h=1.5 i=none cv=black line=1 w=2;
symbol2 value=triangle h=1.5 i=none cv=red   line=2 w=2;
symbol3 value=square   h=1.5 i=none cv=green line=3 w=2;
symbol4 value=none     h=1.5 i=rq cv=black line=1 w=2;
symbol5 value=none     h=1.5 i=rq cv=red   line=2 w=2;
symbol6 value=none     h=1.5 i=rq cv=green line=3 w=2;
plot x1*N_Menge x2*N_Menge x3*N_Menge
          x4*N_Menge x5*N_Menge x6*N_Menge / overlay;
label x1 = 'Frischmasse [g]';
label N_Menge = 'N [kg/ha]';
run;quit;
```



7.4. Weitere Optionen und Befehle in PROC GLM

BY *variables* **;**

Mit dem By-Statement können Teile eines Datensatzes separat ausgewertet werden. Zum Beispiel die Einzelversuche einer Versuchsserie (`BY versuch;`). Die Daten müssen vorher entsprechend sortiert sein.

WEIGHT *variable* **;**

Hier ist der Name einer zusätzlichen Variable im Datensatz anzugeben, die Gewichte für die Berechnung der Fehlerquadratsumme enthält. Es kann bei bestimmten Auswertungen sinnvoll sein, den Beobachtungen ein unterschiedliches Gewicht zu geben. Die erhaltenen Mittelwerte sind dann Gewichtete Kleinst-Quadrat-Schätzer (weighted least square means). So kann man die einzelnen Beobachtungen z.B. jeweils mit dem Kehrwert ihrer jeweiligen Fehlervarianz zu gewichten.

SOLUTION

Bei Regression- und Kovarianzanalyse ist man häufig an den Schätzwerten der Regressionskoeffizienten interessiert.

```
model FM = Wdh Sortenname N_Menge N_Menge*N_Menge Sortenname*N_Menge
     /ss1 solution;
```

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		940.9166667	B	107.2121388	8.78	<.0001
Wdh	W1	151.2500000	B	86.1148621	1.76	0.0871
Wdh	W2	-64.2500000	B	86.1148621	-0.75	0.4602
Wdh	W3	97.9166667	B	86.1148621	1.14	0.2626
Wdh	W4	0.0000000	B	.	.	.
Sortenname	Batis	243.5500000	B	124.7922911	1.95	0.0584
Sortenname	Hybnos	196.0750000	B	124.7922911	1.57	0.1244
Sortenname	Monopol	0.0000000	B	.	.	.
N_Menge		5.8789062	B	1.2849776	4.58	<.0001
N_Menge*N_Menge		-0.0104850		0.0047572	-2.20	0.0337
N_Menge*Sortenname	Batis	-1.5681250	B	0.8338036	-1.88	0.0677
N_Menge*Sortenname	Hybnos	0.2290625	B	0.8338036	0.27	0.7850
N_Menge*Sortenname	Monopol	0.0000000	B	.	.	.

Hieraus können für bestimmte Kombinationen die Erwartungswerte bestimmt werden. Für die Sorte Monopol würde wir bei 100 kg N erwarten:

Y = 940.917 + 0.25*(151.25-64.25 + 97.917 + 0) + 0.00 + 100*5.8789 - 10000*0.010485 + 0

= 1470.186

ESTIMATE

Über Estimate können „besondere“ Mittelwerte oder Vergleiche berechnet werden. Angenommen, wir möchten den Ertrag für die Sorte Monopol bei 100 kg N bestimmen.

```
estimate "Monopol bei 100 kg N"
intercept 1 wdh 0.25 0.25 0.25 0.25 Sortenname 0 0 1
N_Menge 100 N_Menge*N_Menge 10000 Sortenname*N_Menge 0 0 100;
```

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Monopol bei 100 kg N	1470.18620	65.0164090	22.61	<.0001

Das gleiche Resultat erhalten wir über „LSMEANS“.

```
lsmeans Sortenname / at N_Menge =100;
```

```
                    The GLM Procedure
          Least Squares Means at N_Menge=100

              Sortenname       FM LSMEAN

              Batis           1556.92370
              Hybnos          1689.16745
              Monopol         1470.18620
```

Nützlich ist die Verwendung von ESTIMATE für besondere Vergleiche wie zum Beispiel Mittel von Batis und Hybnos gegenüber Monopol bei 100 kg N. Hierfür schreiben wir uns zunächst die zwei zu schätzenden Mittelwerte auf. Dann wird durch Subtraktion die Differenz gebildet

```
estimate "Batis & Hybnos bei 100 kg N"
     intercept 1 wdh 0.25 0.25 0.25 0.25 Sortenname 0.5 0.5 0
     N_Menge 100 N_Menge*N_Menge 10000 Sortenname*N_Menge 50 50 0;

estimate "Monopol bei 100 kg N"
     intercept 1 wdh 0.25 0.25 0.25 0.25 Sortenname 0 0 1
     N_Menge 100 N_Menge*N_Menge 10000 Sortenname*N_Menge 0 0 100;

estimate "Batis & Hybnos vs Monopol bei 100 kg N"
     intercept 0 wdh 0 0 0 0 Sortenname 0.5 0.5 -1
     N_Menge 0 N_Menge*N_Menge 0 Sortenname*N_Menge 50 50 -100;
```

```
Parameter                                Estimate         Error   t Value
Monopol bei 100 kg N                   1470.18620    65.0164090     22.61
Batis & Hybnos bei 100 kg N            1623.04557    52.6037435     30.85
Batis & Hybnos vs Monopol bei 100 kg N   152.85937    66.1811064      2.31

               Parameter                                Pr > |t|
               Monopol bei 100 kg N                       <.0001
               Batis & Hybnos bei 100 kg N                <.0001
               Batis & Hybnos vs Monopol bei 100 kg N     0.0264
```

SLICE

Die Option „SLICE“ bei Anforderung von LSMEANS bewirkt, ein *Slicing* (in Scheiben schneiden) der Tests. Beispiel: Man möchte Sortenunterschiede getrennt für N-Stufen untersuchen:

```
proc glm data=reflex;
class Sorte Nmenge;
model n_gehalt = Nmenge Sorte Sorte*Nmenge;
lsmeans sorte*Nmenge / Slice=Nmenge;
run;
```

```
          Sorte*NMENGE Effect Sliced by NMENGE for N_Gehalt

                          Sum of
NMENGE          DF        Squares     Mean Square    F Value    Pr > F

0                4      182.700000       45.675000       0.42    0.7910
100              4      251.500000       62.875000       0.58    0.6761
160              4      409.500000      102.375000       0.95    0.4420
220              4     1966.000000      491.500000       4.56    0.0028
```

Interpretation: Sortenunterschiede bestehen nur bei der höchsten Düngung.

Slicing ist wesentlich effektiver als ein „**BY**“-Statement, da beim *Slicen* der Restfehler nach wie vor aus allen Beobachtungen geschätzt wird, während man durch ein „BY“, den Versuch in Teilversuche zerteilen würde!
Weitere Hinweise zum Slicing und viele interessante SAS-Links finden sich auf der Homepage von Prof. Oliver Schabenberger (http://home.nc.rr.com/schabenb/SASSlice.html)

Durch die Option „**OUT=**...“ werden die LSMeans und ihre Standardfehler in eine zu benennende SAS-Datei geschrieben.

Die Option „**AT**“ ist wichtig für Modelle, die quantitative Variablen enthalten, wie z.B. Kovarianzanalysen. Möchte man bei den Beispieldaten eine Schätzung für den Stickstoffgehalt bei einer Düngung von 200 kg N /ha so lautet die Anweisung

```
proc glm data=reflex;
class Sorte;
model n_gehalt = Nmenge Sorte Sorte*Nmenge;
lsmeans sorte / at nmenge = 200; run;
```

OUTPUT < **OUT=**SAS-data-set > keyword=names < ... keyword=names > < / option > **;**

Hier kann eine spezielle SAS-Datei mit Erwartungswerten und Residuen erzeugt werden.
Die wichtigsten drei *keywords* sind RESIDUAL (Residuen = Beobachtung – Erwartungswert), PREDICTED (Erwartungswert) und STUDENT (studentisierte Residuen = Residuen dividiert durch ihre Standardabweichung).

Beispiel: Abspeichern der Erwartungswerte und der studentisierten Residuen in einer Datei „Stud_res“.

```
proc glm ;
class Sorte Nmenge;
model n_gehalt = Nmenge Sorte Sorte*Nmenge;
output out=stud_res predicted=pred student=stud;
run;quit;
```

RANDOM effects < / options > **;**
Ganz wichtig bei gemischten Modellen! Während in der Model-Zeile das komplette Model spezifiziert wird, sind in der Random-Zeile noch einmal zusätzlich alle zufälligen Effekte zu benennen. Die Option **Test** bewirkt, das die korrekten F-Test, entsprechend dem gemischten Modell gemacht werden.

7.5. Analyse einer Spaltanlage mit PROC GLM

In einer Spaltanlage werden die Versuchsglieder nicht in einem Schritt auf die Versuchseinheiten (z.B. Parzellen) verteilt sondern man geht in zwei Schritten vor. Hierbei unterscheidet man Groß- und Kleinteilstücke, sowie Groß- und Kleinteilstückfaktoren.

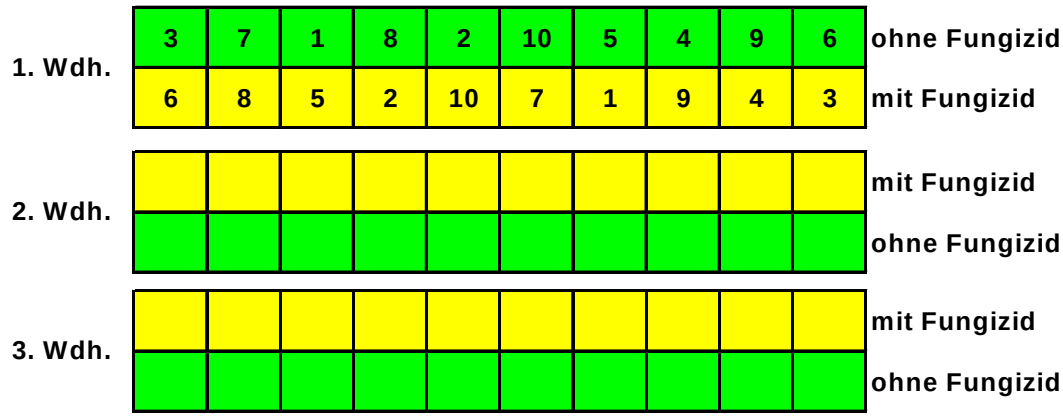


Abb. 18: Beispiel einer zweifaktoriellen Spaltanlage (Drei Wiederholungen, zwei Fungizidstufen, zehn Sorten)

Gegenüber einer zweifaktoriellen Blockanlage besteht der Unterschied, dass für beide Randomisierungseinheiten (Klein- und Großteilstücke) ein zufälliger Fehlereffekt in das Modell zu integrieren ist. Rufen wir uns noch einmal das Modell für die zweifaktorielle Blockanlage in Erinnerung:

$$y_{ijk} = \mu + \alpha_i + \tau_j + \beta_k + \alpha\tau_{ij} + e_{ijk}$$

y_{ij} Ertrag der Parzelle mit i-ter Sorte und j-ter Düngung im k-ten Block
μ Allgemeiner Mittelwert
α_i Effekt der i-ten Sorte
τ_j Effekt der j-ten Düngung
β_k Effekt des k-ten Blocks
$\alpha\tau_{ij}$ Interaktion zwischen i-ter Sorte und j-ter Düngung
e_{ijk} Fehler der ijk-ten Parzelle ~N(0,σ^2_e)

Nehmen mir nun an, die Düngung wäre auf Großteilstücke und die Sorten auf Kleinteilstücke verteilt. Das Modell für eine Spaltanlage lautet:

$$y_{ijk} = \mu + \alpha_i + \tau_j + \beta_k + \alpha\tau_{ij} + ß\tau_{kj} + e_{ijk}$$

y_{ij} Ertrag der Parzelle mit i-ter Sorte und j-ter Düngung im k-ten Block
μ Allgemeiner Mittelwert
α_i Effekt der i-ten Sorte
τ_j Effekt der j-ten Düngung
β_k Effekt des k-ten Blocks
$\alpha\tau_{ij}$ Interaktion zwischen i-ter Sorte und j-ter Düngung
$ß\tau_{kj}$ Interaktion zwischen j-ter Düngung und k-tem Block (=Großteilstückfehler)
e_{ijk} Fehler der ijk-ten Parzelle ~N(0,σ^2_e)

Dieser zusätzliche Fehlerterm ist in der Prozedur GLM in die Model-Zeile aufzunehmen und zusätzlich in einer „Random"-Zeile als zufällig zu definieren. Ein solches Modell mit mehr als einem zufäligen Effekt wird auch als Gemischtes Modell (MIXED MODEL) bezeichnet. Die Anweisung Test sorgt dafür, dass die dem gemischten Modell entsprechenden F-Tests gemacht werden. So wird der Effekt der N-Menge gegen den Großteilstückfehler getestet. Bei der Berechnung von Grenzdifferenzen muss der Fehlerterm explizit definiert werden. Falls Vergleiche zwischen Sorten über N-Stufen hinweg von Interesse sind, so muss der Standardfehler für diese Vergleiche aus einer Kombination von Groß- und Kleinteilstückfehler berechnet werden. Hierfür ist die Prozedur MIXED besser geeignet.

Standardfehler der Differenz für Vergleiche von Kleinteilstückfaktorstufen (A) auf unterschiedlichen Großteilstückfaktorstufen (B)

$$s_d = \sqrt{\frac{2 \cdot [MQ_{ra} + (b-1) \cdot MQ_e]}{rb}} \qquad FG = \frac{a \cdot (r-1) \cdot (a-1) \cdot [(b-1) \cdot MQ_e + MQ_{ra}]^2}{(a-1) \cdot (b-1) \cdot MQ_e^2 + (a) \cdot MQ_{ra}^2}$$

MQ_{ra}	*Großteilstückfehler*
MQ_e	*Kleinteilstück- bzw. Restfehler*
a	*Anzahl Großteilstückfaktorstufen*
b	*Anzahl Kleinteilstückfaktorstufen*
r	*Anzahl Wiederholungen*

```
/*Analyse als Spaltanlage*/
Proc glm data=Weizen;
class Sortenname N Wdh;
model FM = Wdh Sortenname N Sortenname*N wdh*N/ ss3;
random wdh*N/Test;
means N / LSD E=wdh*N;
means Sortenname /LSD;
/*lsmeans Sortenname*N / pdiff E=??? Geht Nicht! MIXED verwenden!*/
run;
```

```
The GLM Procedure
Tests of Hypotheses for Mixed Model Analysis of Variance
Dependent Variable: FM

    Source                     DF   Type III SS   Mean Square  F Value  Pr > F

    Wdh                         3        336525        112175     3.96  0.0472
  * N                           3       3519519       1173173    41.37  <.0001
    Error: MS(N*Wdh)            9        255195         28355
  * This test assumes one or more other fixed effects are zero.

    Source                     DF   Type III SS   Mean Square  F Value  Pr > F

  * Sortenname                  2        433779        216890     4.16  0.0282
    Sortenname*N                6        390114         65019     1.25  0.3187
    N*Wdh                       9        255195         28355     0.54  0.8284
    Error: MS(Error)           24       1252576         52191
  •   This test assumes one or more other fixed effects are zero.
```

t Tests (LSD) for FM

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

Alpha	0.05
Error Degrees of Freedom	9
Error Mean Square	28355.02
Critical Value of t	2.26216
Least Significant Difference	155.51

Means with the same letter are not significantly different.

t Grouping	Mean	N	N
A	1821.00	12	N4
A	1772.17	12	N3
B	1463.50	12	N2
C	1146.25	12	N1

8. Prüfung der Modellvorausetzungen

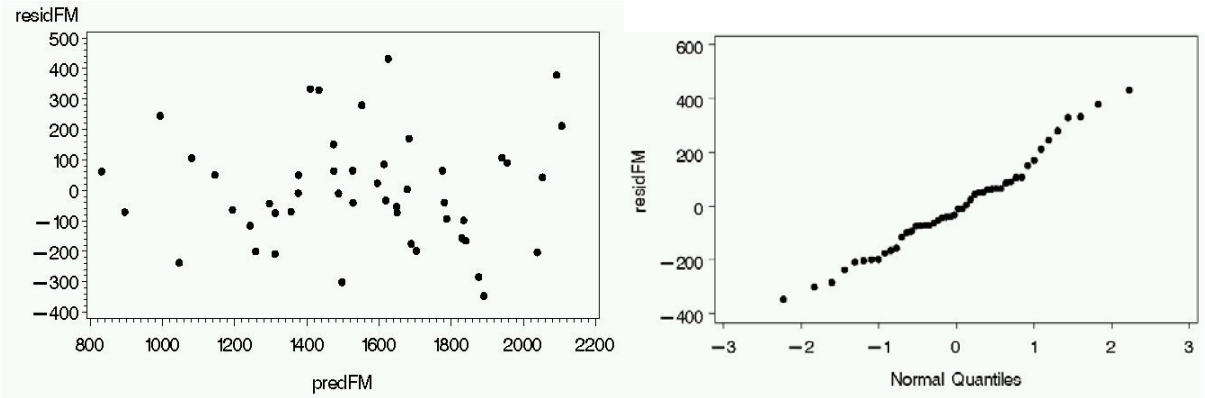
Wichtigste Modellvoraussetzungen für Varianzanalyse, Korrelation und Regression sind:

- Normalverteilung der Fehler
- Varianzhomogenität

Diese sollten am besten über einen Plot der Residuen geprüft werden. Die Residuen, Erwartungswerte und studentisierten Residuen können in den Prozeduren GLM und REG über `OUTPUT out =…` erzeugt und in eine SAS-Datei geschrieben werden.

```
/*Prüfung der Modellvoraussetzungen*/

Proc glm data=Weizen;
class Sortenname N Wdh;
model FM AnzHalme FMKoerner= Wdh Sortenname N Sortenname*N/ss3;
output out=res  residual  = residFM residAnz residFMK
                predicted = predFM  predAnz  predFMK
                student   = studFM  studAnz  studFMK;
run;

proc univariate data=res normal plot;
var residFM residAnz residFMK ;
qqplot /normal;
histogram /normal;
run;

proc gplot data=res;
plot residFM*predFM ;
plot residAnz*predAnz ;
plot residFMK*predFMK ;
symbol value=dot;
run;
```

Als Output der Prozedur GPLOT erhält man einen QQ-Plot, ein Histogramm (hier nicht gezeigt) und einen Plot der Residuen gegen die Erwartungswerte. Der QQ-Plot sollte eine Diagonale zeigen, das Histogramm soll eine Normalverteilung zeigen und im letzten Plot sollte man möglichst eine horizontale Wolke sehen. Strukturen oder Trends im Plot der Residuen gegen die Erwartungswerte zeigen eine Abhängigkeit zwischen Mittelwert und Varianz und damit Varianzheterogenität.



Als Testverfahren für Normalverteilung kann für Stichproben der Shapiro-Wilk-Test verwendet werden. Die Statistik im Shapiro-Wilk-Test sollte möglichst nahe an 1 liegen und mehr als 10% Überschreitungswahrscheinlichkeit aufweisen. Für die Variable FM ist alles in Ordnung.

Output Proc Univariate (gekürzt):

Variable: residFM

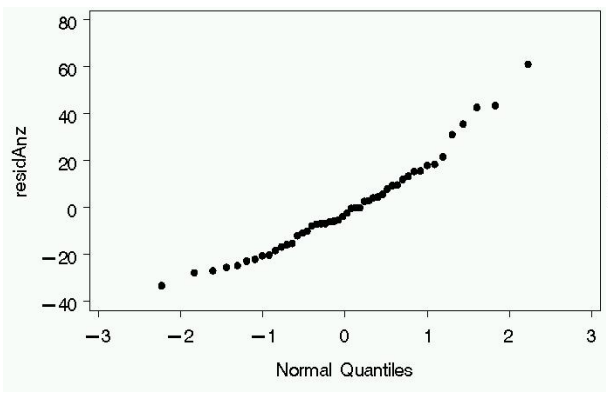
Moments

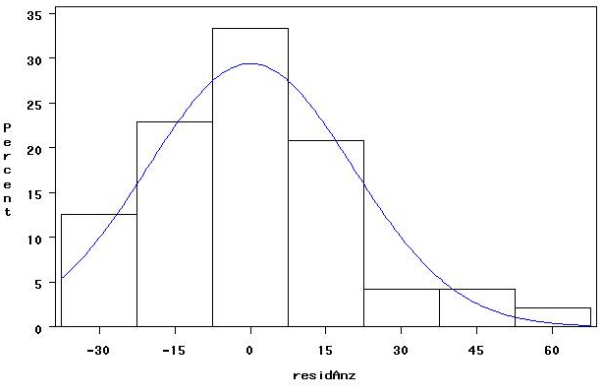
N	48	Sum Weights	48
Mean	0	Sum Observations	0
Std Deviation	179.10954	Variance	32080.2274
Skewness	0.42949886	Kurtosis	-0.0256589
Uncorrected SS	1507770.69	Corrected SS	1507770.69
Coeff Variation	.	Std Error Mean	25.8522353

Tests for Normality

Test	--Statistic---		-----p Value------	
Shapiro-Wilk	W	0.975868	Pr < W	0.4204
Kolmogorov-Smirnov	D	0.086853	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.057975	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.381136	Pr > A-Sq	>0.2500

Bei der Anzahl Halme je qm gibt es allerdings eine Abweichung von der Normalverteilung. Die Verteilung ist schief. Der Shapiro-Wilk-Test ist mit p = 0.046 signifikant. Das heißt, wir haben eine signifikante Abweichung von der Normalverteilung!





Variable: residAnz

Moments

N	48	Sum Weights	48
Mean	0	Sum Observations	0
Std Deviation	20.3616243	Variance	414.595745
Skewness	0.86026721	Kurtosis	0.80754778
Uncorrected SS	19486	Corrected SS	19486
Coeff Variation	.	Std Error Mean	2.93894732

Tests for Normality

Test	--Statistic---		-----p Value------	
Shapiro-Wilk	W	0.951521	Pr < W	0.0460

9. Transformationen

Wenn die Voraussetzungen der varianzanalytischen Verfahren verletzt sind, hilft oftmals eine Transformation der Daten. Hierfür kann ein Data Step genutzt werden. Bei schiefen Verteilungen kann eine log- oder Wurzeltransformation hilfreich sein. Für Prozentzahlen sollte man eine LOGIT- oder Winkeltransformation verwenden. Eine sehr allgemeine Transformation ist die BOX-COX-Power-Transformation. Hier kann über flexible Wahl des Transformationsparameters „phi“ eine ganze Familie von unterschiedlichen Transformationen erreicht werden.

Verschiedene Transformationen

logarithmische Transformation	`ln_y = log(y);`
Wurzeltransformation	`sqrt_y = sqrt(y);`
Logit-Transformation	`logit_y = log(y / (1-y));`
BOX-COX-Transformation	`box_y = ((y**phi)-1)/phi;`
Folded-Exponential	`exp_y=(exp(phi*y)-1)/phi/2-(exp(phi*(1-y))-1)/phi/2;`

Bei der Log- und LOGIT-Transformation treten Probleme bei der Logarithmierung von Nullen auf. In einem solchen Falle hilft die Addition eines kleinen Wertes « c ».

```
ln_y = log(y + c);          logit_y = log(y + c / (1 – y + c));
```

```
/*SAS-Code für Winkeltransformation (Speziell für Prozentzahlen)*/

Data trans; Set daten;
Pi = 2*arsin(1);
z = y / 100;
w = arsin(sqrt(z));
trans_w = w*180/pi;
run;
```

Die Suche nach den optimalen Werten von Phi für die BOX-COX und die Exponential-Transformation kann mittels Macros erfolgen, die auf der Internetseite des FG Bioinfomatik Hohenheim zur Verfügung stehen (http://www.uni-hohenheim.de/bioinformatik/beratung/toolsmacros/macrotool.html).

Um das Macro zu nutzen, muss es in den SAS-Editor geladen werden und einmal ablaufen, dann ist es aktiv. Anschließend müssen das Modell, die Daten und die abhängige Variable definiert werden. Die Suche nach der optimalen Transformation für die Anzahl Halme im Weizenexperiment ergab ein Phi von -0.34.

Allgemein: `%boxcox(phimin=,phimax=, steps=, model=, class=, stmts=, data=, response=);`

für Beispieldaten: `%boxcox(phimin=-1,phimax=1, steps=100, model = Wdh Sortenname N Sortenname*N, class= Wdh Sortenname N, stmts=, data=weizen, response = AnzHalme);`

Die Analyse erfolgt dann mit den transformierten Werten. Man kann sich wiederum Residuen erzeugen lassen und deren Verteilung kontrollieren. Die Signifikanztests für die transformierten Daten sind auch für die Originaldaten valide. Zum Zweck der besseren Darstellung ist eine Rücktransformation möglich.

10. Die Prozedur MIXED

Mixed wurde für gemischte Modelle entwickelt, also Modelle, die neben dem Restfehler weitere zufällige Effekte enthalten. Hierunter fallen Spalt- und Streifenanlagen aber auch Versuchsserien, Dauerversuche, Versuche mit Messwiederholungen in Zeit und/oder Raum. Gemischte Modelle haben für die Auswertung landwirtschaftlicher Versuche damit eine sehr hohe Bedeutung.

10.1. Einführendes Beispiel: Auswertung einer Spaltanlage mit MIXED

```
/*Analyse der Weizendaten als Spaltanlage mit MIXED*/
Proc MIXED data=Weizen;
class Sortenname N Wdh;
model FM = Wdh Sortenname N Sortenname*N / DDFM=KR;
random wdh*N;
lsmeans Sortenname N Sortenname*N / pdiff;
ods output lsmeans=lsmeans diffs=diffs;
run;
```

Im Output ist zunächst auf Meldung zu achten, die das Iterationsverhalten des REML-Algorithmus beschreiben.

„Convergence Criteria met“ = Die Iteration hat konvergiert und eine Lösung gefunden

„Estimated G-Matrix not positiv definit“ = zufällige Effekte im Modell haben Varianz 0; Dieses ist im Output unten der Fall. Der Großteilstückfehler wird auf 0 geschätzt und damit fällt dieser Effekt faktisch aus dem Modell, was nicht immer wünschenswert ist.

„Hessian is not positiv definit“ = Evtl. nur ein lokales Maximum der Likelihood gefunden. Hier ist Vorsicht geboten!

Convergence criteria met.

Covariance Parameter Estimates

Cov Parm	Estimate
N*Wdh	0
Residual	45690

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Wdh	3	33	2.46	0.0804
Sortenname	2	33	4.75	0.0154
N	3	33	25.68	<.0001
Sortenname*N	6	33	1.42	0.2355

Least Squares Means

Effect	Sortenname	N	Estimate	Standard Error	DF	t Value	Pr > |t|
N		N1	1146.25	61.7050	33	18.58	<.0001
N		N2	1463.50	61.7050	33	23.72	<.0001
N		N3	1772.17	61.7050	33	28.72	<.0001
N		N4	1821.00	61.7050	33	29.51	<.0001
Sortenname	Batis		1513.13	53.4381	33	28.32	<.0001
Sortenname	Hybnos		1681.31	53.4381	33	31.46	<.0001
Sortenname	Monopol		1457.75	53.4381	33	27.28	<.0001

10.2. Die Syntax von MIXED im Detail

Die Syntax von PROC MIXED enthält folgende Statements:

PROC MIXED < options > ;
BY variables ;
CLASS variables ;
ID variables ;
MODEL dependent = < fixed-effects > < / options > ;
RANDOM random-effects < / options > ;
REPEATED < repeated-effect > < / options > ;
PARMS (value-list) ... < / options > ;
PRIOR < distribution > < / options > ;
CONTRAST 'label' < fixed-effect values ... >
< | random-effect values ... > , ... < / options > ;
ESTIMATE 'label' < fixed-effect values ... >
< | random-effect values ... >< / options > ;
LSMEANS fixed-effects < / options > ;
MAKE 'table' OUT=SAS-data-set ;
WEIGHT variable ;

PROC MIXED < options > ;
Nach dem Aufruf der Prozedur kann wie üblich die Input-Datei spezifiziert werden. Von der Vielzahl sonstiger Optionen sollen hier vier vorgestellt werden:

METHOD=…
Hier kann die Methode der Parameterschätzung gewählt werden. Default ist REML (Restricted Maximum Likelihood). Alternativen sind ML (Maximum Likelihood), MIVQUE0 (minimum variance quadratic unbiased estimation), sowie entsprechend wie bei PROC GLM Type1, Type2, Type3. **Tipp**: Immer REML nehmen

NOBOUND
Mixed schätzt Varianzkomponenten. Per Definition können Varianzen nicht negativ werden. Man kann aber aufgrund der Erwartungswerte der MQ aus einer klassischen Varianzanalyse nach dem ANOVA-Prinzip die Varianzkomponenten berechnen. Dabei kann es durchaus einmal vorkommen, dass einzelne Komponenten rechnerisch <0 sind. Durch Wählen der Option „NOBOUND“ wird die Restriktion Var>=0 in MIXED fallengelassen und man erhält z.B. die gleichen F-Tests wie in PROC GLM.
Mit der Verwendung von Nobound sollte man allerdings vorsichtig sein, wenn einzelne Faktorstufenkombinationen komplett fehlen (nicht klassifikationsbalanciert, PIEPHO & SPILKE 1999). Dieses ist z.B. bei Lateinischen Quadraten und Rechtecken und Anlagen in unvollständigen Blöcken der Fall. Dagegen sind Blockanlagen, Spalt- und Streifenanlagen klassifikationsbalanciert, nobound darf also angewendet werden.
Tipp: F-Tests soweit möglich mit PROC GLM berechnen und Mittelwertvergleiche in MIXED.

COVTEST
Die Option Covtest bewirkt die Prüfung der Varianzkomponenten auf Signifikanz über einen WALD-Test. Dieser ist bei geringem Stichprobenumfang (beim Faktor Jahr ist das fast immer gegeben) allerdings nicht valide. Besser ist in diesem Fall ein Likelihood-Ratio-Test (siehe LITTELL et al. 1996).

CL<=WALD>
Es wird ein Konfidenzintervall für die Varianzkomponenten berechnet.

BY variables ;

Sorgt für separate Auswertung von Untergruppen. Daten vorher sortieren!

CLASS variables ;

Hier werden die Klassifikationsvariablen (nominales Niveau) aufgeführt.

ID variables ;

Variablen für eine Output-Datei können definiert werden. Ohne ID werden alle Variablen in die Output-Datei geschrieben.

MODEL dependent = < fixed-effects > < / options > ;

Im Gegensatz zur Prozedur GLM werden in MIXED in der Model-Zeile **nur die fixen Effekte** aufgeführt! Ansonsten gelten die gleichen Schreibregeln wie bei PROC GLM. Von der erneuten Vielzahl an Optionen, die nach einem Schrägstrich angegeben werden können, haben folgende besondere Bedeutung:

DDFM=...

Hier wird die Art der Freiheitsgrad-Approximation für die F-Tests und Mittelwertvergleiche festgelegt. Solche Approximationen sind z.B. bei bestimmten Mittelwertvergleichen in Spaltanlagen notwendig. Empfehlenswert ist es hier DDFM=Satterth oder DDFM = KR zu wählen.

HType=

Entsprechend PROC GLM kann die Art der Quadratsummenzerlegung (SS1, SS3) bestimmt werden. Hier kommen HType=1 oder HType=3 in Frage.

NOINT

Es wird ein Modell ohne allgemeinen Mittelwert bzw. bei Regressionen ohne Achsenabschnitt angepasst.

OUTP=...

Residuen und Erwartungswerte werden in eine SAS-Datei geschrieben, deren Name hier zu definieren ist.

SOLUTION

Es werden die Schätzwerte für einzelne Parameter ausgegeben. Wichtig bei Regressions- und Kovarianzanalysen.

RANDOM random-effects < / options > ;

Hier sind die **zufälligen Effekte** im Modell aufzuführen.

Zwei wichtige Optionen sind SUBJECT=... und TYPE=... . Mit SUBJECT sind die Objekte zu benennen für die z.B. eine Messwiederholung durchgeführt wurde. Über die Option TYPE wird die Art der Varianz-Kovarianz-Matrix festgelegt.

Type=VC: Es werden lediglich Varianzkomponenten geschätzt, Kovarianzen sind Null; VC ist die Default-Einstellung.

Type=UN: (unstructured). Jede Stufe eines Subjects (z.B. Termin einer Messung) hat eine eigene Varianz und jedes Paar von Beobachtungen innerhalb eines Subjects hat eine eigene Kovarianz.

Type=CS (Compound Symetrie): Alle Varianzen sind gleich und Kovarianz ist konstant.

Type = VC $\begin{pmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{pmatrix}$

Type = UN $\begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{pmatrix}$

Type = CS $\begin{pmatrix} \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 \end{pmatrix}$

Neben diesen drei Strukturen können viele weitere gewählt werden. Für zeitliche Messwiederholungen ist eine autoregressive Struktur **Type=AR(1)** wichtig. Hierbei hängt die Korrelation zwischen zwei Beobachtungen von Ihrer zeitlichen Distanz ab. Das heißt eng auf einander folgende Messungen sind hoch korreliert.

Wann ist ein Effekt zufällig?
Einen Effekt als *zufällig* zu bezeichnen, ist immer dann zulässig, wenn die Stufen des betreffenden Faktors eine zufällige Stichprobe aus einer Grundgesamtheit von Faktorstufen darstellen. Meist werden Orte und/oder Jahre als zufällig betrachtet. Man kann dieses aber unter Umständen auch für Blöcke in einem Feldversuch postulieren. Mit der Zufälligkeit impliziert man in aller Regel auch eine Verteilungsannahme. Die Effekte sollen normalverteilt sein.
Vorteil zufälliger Effekte ist, dass sie eine Aussage bzw. Prognose für die Grundgesamtheit erlauben, aus der sie gezogen worden sind (Inference Space). Falls diese Prognose nicht gewünscht ist, so ist es durchaus erlaubt, zufällige Effekte als fix zu betrachten. Dadurch erhält man in der Regel kleinere Standardfehler aber eben auch einen kleineren Inference Space.
Interaktionen, die einen zufälligen Haupteffekt enthalten sind automatisch ebenfalls zufällig.

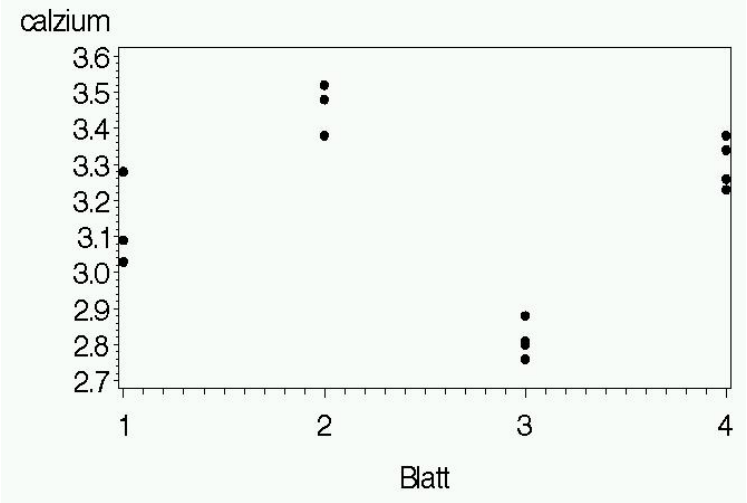
REPEATED < repeated-effect > < / options > ;

Hier kann die Varianz-Kovarianz-Matrix für kleinste Einheit eines Experiments spezifiziert werden. Es sind wie in der RANDOM-Zeile nach einem Schrägstrich das SUBJECT zu bestimmen und der TYPE.

Beispiel:
Bei einzelnen Blättern einer Rübenpflanze wurde der Calciumgehalt gemessen.

```
data pflanze;
input Blatt @@;
do probe=1 to 4;
  input calcium @@; output;
end;
datalines;
1    3.28 3.09 3.03 3.03
2    3.52 3.48 3.38 3.38
3    2.88 2.80 2.81 2.76
```

```
4    3.34 3.38 3.23 3.26
;
run;

goptions ftext=swiss htext=1.8;
proc gplot data=pflanze;
plot calcium*Blatt;
symbol value=dot;
run;
```



Man erkennt, dass sich die Messwerte, die vom gleichen Blatt stammen jeweils ähnlich sind. Die Werte weisen eine sogenannte „Intraclass-Korrelation“ auf. Diese Tatsache muss Berücksichtigung finden, wenn man den mittleren Calcium-Gehalt der Pflanze berechnen möchte.

Warum ist das relevant?
Bei der varianzanalytischen Auswertung linearer Modelle geht man davon aus, dass die einzelnen Beobachtungen unabhängig voneinander sind. Das heißt, wenn man sich willkürlich zwei Messwerte herausgreift, so sollten diese sich nicht ähnlicher sein, als jedes andere Paar von Messwerten.

Drei unterschiedliche Standardfehler
Wir wollen den Mittelwert für diese Pflanze berechnen. Hierbei können wir entweder die 16 Messwerte als völlig unabhängig betrachten, wir können einen festen Blatteffekt modellieren oder einen zufälligen Blatteffekt annehmen.

```
/*unabhängige Messungen*/
Proc mixed data=pflanze;
class Blatt;
model Calcium = /ddfm=kr;
estimate "Mittel" intercept 1 ;  run;

/*fixer Blatteffekt*/
Proc mixed data=pflanze;
class Blatt;
model Calcium = Blatt / ddfm=kr;
estimate "Mittel" intercept 1 Blatt 0.25 0.25 0.25 0.25 ;
run;
```

```
/*zufälliger Blatteffekt*/
Proc mixed data=pflanze;
class Blatt;
model Calcium = /ddfm=kr;
random Blatt;
estimate "Mittel" intercept 1 ;
run;

/*Alternative zufälliger Blatteffekt*/
Proc mixed data=pflanze;
class Blatt;
model Calcium = /ddfm=kr;
repeated / subject=Blatt type=cs;
estimate "Mittel" intercept 1 ;
run;
```

Völlig zufällig

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Mittel	3.1656	0.06350	15	49.86	<.0001

Mit fixem Blatt-Effekt

Mittel	3.1656	0.02031	12	155.84	<.0001

Zufälliger Blatt-Effekt

Mittel	3.1656	0.1360	3	23.27	0.0002

Die erste Auswertung ist auf jeden Fall falsch. Hier werden die Einzelwerte als unabhängige Messungen betrachtet, was nicht stimmt. Das Modell mit fixem Blatt-Effekt beschert uns den kleinsten Standardfehler, wir haben dann allerdings nur den „*Narrow Space of Inference*“ vor uns. Wir können den Mittelwert für die vier Blätter berechnen, aber eine Aussage für die gesamte Pflanze ist nicht möglich. Hierzu müssen wir die analysierten Blätter als zufällige Stichprobe aus der Grundgesamtheit aller Blätter der Pflanze sehen.
Das MIXED-Programm liefert uns Schätzungen für die Varianzkomponenten. Im Modell mit zufälligen Blatteffekten lauten diese:

Covariance Parameter Estimates

Cov Parm	Estimate
Blatt	0.07238
Residual	0.006602

$$Var(\mu) = \frac{\sigma_b^2}{r} + \frac{\sigma_e^2}{rt} = \frac{0{,}07238}{4} + \frac{0{,}006602}{16} = 0{,}01851$$

Der Standardfehler beträgt $\sqrt{Var(\mu)} = \sqrt{0{,}02851} = 0{,}136$.

Korrelation und Kovarianz

Aus den Varianzkomponenten können wir auch die Korrelation zwischen den Messungen vom gleichen Blatt bestimmen.

Die Korrelation ist[2]: $corr(y_{ij}, y_{ij}') = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_e^2} = \frac{0{,}07238}{0{,}07238 + 0{,}006602} = \frac{0{,}07238}{0{,}078982} = 0{,}9164$.

Dass diese Messungen vom gleichen Blatt nicht unabhängig voneinander sind, können wir auch wie eine Messwiederholung auffassen. Ein Blatt wird „wiederholt" gemessen. Je nachdem wie hoch die Korrelation ist, desto geringer ist der Informationsgewinn durch eine weitere Messung. Wenn jede weitere Wiederholung der Messung den gleichen Wert liefert, so wird die Korrelation 1. Wir erhalten allerdings auch keine weiteren Informationen.
Messwiederholung kann durch eine spezielle Syntax in MIXED modelliert werden. Hierbei wird ein Blatt als „Subject" betrachtet, welches wiederholt gemessen wird. Es ist wichtig, eine bestimmte „Korrelationsstruktur" zu wählen. Wir haben hier den Typ *Compound Symmetry* (CS) vor uns. Bei CS sind alle Messungen vom gleichen Subject gleich stark korreliert, was in unserem Falle Sinn macht. Das SAS-Programm liefert bei Verwendung der Optionen „r" und „rcorr" die Varianz-Kovarianz-Matrix und die Korrelationsmatrix.

```
Proc mixed data=pflanze;
class Blatt;
model Calcium = /ddfm=kr;
repeated / subject=Blatt type=cs r rcorr;
run;
```

Estimated R Matrix for Blatt 1

Row	Col1	Col2	Col3	Col4
1	0.07898	0.07238	0.07238	0.07238
2	0.07238	0.07898	0.07238	0.07238
3	0.07238	0.07238	0.07898	0.07238
4	0.07238	0.07238	0.07238	0.07898

Estimated R Correlation Matrix for Blatt 1

Row	Col1	Col2	Col3	Col4
1	1.0000	0.9164	0.9164	0.9164
2	0.9164	1.0000	0.9164	0.9164
3	0.9164	0.9164	1.0000	0.9164
4	0.9164	0.9164	0.9164	1.0000

Die R-Matrix und die R_{corr}-Matrix wird von SAS bei Verwendung der Compound-Symmetry-Struktur folgendermaßen parametrisiert:

[2] Erklärung: Korrelation ist standardisierte Kovarianz (Kovarianz zwischen zwei Messreihen dividiert durch das Produkt der Standardabweichungen der beiden Messreihen). Das was die Messwerte eines Blattes ähnlich macht, ihre Kovarianz, ist die gleichgerichtete Abweichung vom Mittelwert. Diese gleichgerichtete Abweichung ist nichts anderes als die Varianz der Blätter. Diese steht über dem Bruchstrich. Die Varianz der Messwerte selber ergibt sich dagegen als Summe von Fehlervarianz und Blattvarianz. Wenn man die Quadratwurzel aus dieser Summe bildet, erhält man die Standardabweichungen. Diese sind für beide Messreihen gleich. Das Produkt der Standardabweichungen entspricht wiederum der Messwert-Varianz, also das was in der Formel unter dem Bruchstrich steht.

$$\mathbf{R} = \begin{pmatrix} \sigma^2 + \sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2 + \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2 + \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma^2 + \sigma_1 \end{pmatrix} \qquad \mathbf{R}_{corr} = \begin{pmatrix} \rho_{11} & \rho_{21} & \rho_{31} & \rho_{41} \\ \rho_{21} & \rho_{22} & \rho_{32} & \rho_{42} \\ \rho_{31} & \rho_{32} & \rho_{33} & \rho_{43} \\ \rho_{41} & \rho_{42} & \rho_{43} & \rho_{44} \end{pmatrix}$$

Auf der Diagonalen der R-Matrix steht die Summe aus Fehlervarianz und Kovarianz. Jeweils jenseits der Diagonalen stehen die Kovarianzen zwischen den einzelnen Messungen.
Da die Kovarianz zwischen den Messungen der Blattvarianz entspricht können wir diese jenseits der Diagonalen ablesen. Sie beträgt $s^2b = 0.07238$.
Die Fehlervarianz beträgt 0,07898 - 0,07238 = 0,0066.

PARMS (value-list) ... < / options > ;

Hier können Startwerte für die Varianzkomponenten eingegeben werden. Angenommen ein Modell hat zwei zufällige Effekte (Spaltanlage).

```
Parms (1) (4);
```

⇒ für den Großteilstückfehler wird ein Startwert von 1 und für den Restfehler einer von 4 angenommen. Mit

```
Parms (1) (4) / hold = 2;
```

würde man den Wert für die zweite Komponente, also hier den Restfehler, auf 4 fixieren.

ESTIMATE 'label' < fixed-effect values ... > < | random-effect values ... >< / options > ;

Ähnlich wie der gleichnamige Befehl in PROC GLM zu gebrauchen.

LSMEANS fixed-effects < / options > ;

Ähnlich wie der gleichnamige Befehl in PROC GLM zu gebrauchen.

10.3. Streifenanlagen mit MIXED

Bei Streifenanlagen gibt es sozusagen zwei Großteilstücke. Innerhalb jeder Wiederholung wird zunächst ein Faktor in Zeilen randomisiert und dann ein zweiter Faktor in Spalten. Die Abbildung zeigt eine Streifenanlage mit Faktor A (z.B. Saatzeit) in Großteilstücken in horizontaler Richtung, Stufen A-C und Faktor B (z.B. Düngung) in Großteilstücken, in vertikaler Richtung, Stufen 1 –2.
Wenn wir der Regel „*Analyze as randomized*“ folgen und jedem Versuchseinheit einen Fehlerterm zuordnen, so können wir das Modell sehr einfach aufstellen.

$y_{ijk} = \mu + a_i + r_k + ra_{ik} + b_j + ab_{ij} + e_{ijk}$

- μ Allgemeiner Mittelwert
- r_k Effekt der k-ten Wiederholung (fix)
- a_i Effekt der i-ten Saatzeit (fix)
- ra_{ik} Fehler des ik-ten Großteilstücks (Zeilenfehler) ~N(0,σ^2_{ra})
- b_j Effekt der j-ten Düngung (fix)
- rb_{jk} Fehler des jk-ten Großteilstücks (Spaltenfehler) ~N(0,σ^2_{rb})
- ab_{ij} *I*nteraktion zwischen i-ter Saatzeit und j-ter Düngung (fix)
- e_{ijk} Fehler der ijk-ten Parzelle ~N(0,σ^2_e)

Abb.: Beispiel für Streifenanlage

Dieses Modell in MIXED-Code:

```
/*Modell für Streifenanlage*/
proc mixed ;
class A B rep;
model y = rep A B A*B /ddfm=kr;
random A*rep B*rep;
lsmeans A B A*B/ pdiff adjust=tukey cl;
ods output lsmeans=lsmeans0815 diffs=diffs0815;
run;
```

10.4. Streifen-Spalt-Anlage

Es sind für mehrfaktorielle Versuche unzählige Kombinationen von Block-, Spalt und Streifenstrukturen denkbar. Hier soll nur ein Beispiel gegeben werden (C. Pringas, Institut für Zuckerrübenforschung Göttingen). Um die Auswirkungen verschiedener pflanzenbaulicher Maßnahmen auf den Fusariumbefall von Weizen zu untersuchen, wurde Fungizidbehandlung in Zeilen, Bodenbearbeitung in Spalten und Sorte in Unterspalten randomisiert (siehe Abb. Folgeseite).

Das Modell lautet:

$$y_{ijkl} = a_i + b_j + \gamma_k + ba_{ij} + a\gamma_{ik} + b\gamma_{jk} + ba\gamma_{ijk} + r_l + ra_{il} + rb_{jl} + rba_{ijl} + rb\gamma_{jkl} + e_{ijkl}$$

r_l	Effekt der l-ten Wiederholung
a_i	Effekt der i-ten Fungizidstufe
b_j	Effekt der j-ten Bodenbearbeitung
γ_k	Effekt der k-ten Sorte
$ba\gamma_{ijk}$	Interaktionen Fungizid*Bodenbearbeitung*Sorte
ra_{il}	Fehler des il-ten Großteilstücks ~ N(0,□²ra)
rb_{jl}	Fehler des jl-ten Großteilstücks ~ N(0,□²rb)
rba_{ijl}	Fehler des ijl-ten Mittelteilstücks (Kombination Zeile/Spalte) ~ N(0,□²rab)
$rb\gamma_{jkl}$	Fehler des ikl-ten Mittelteilstücks (Kombination Spalte/Unterspalte) ~ N(0,□²rbc)
e_{ijkl}	Fehler der ijkl-ten Parzelle ~ N(0,□²e)

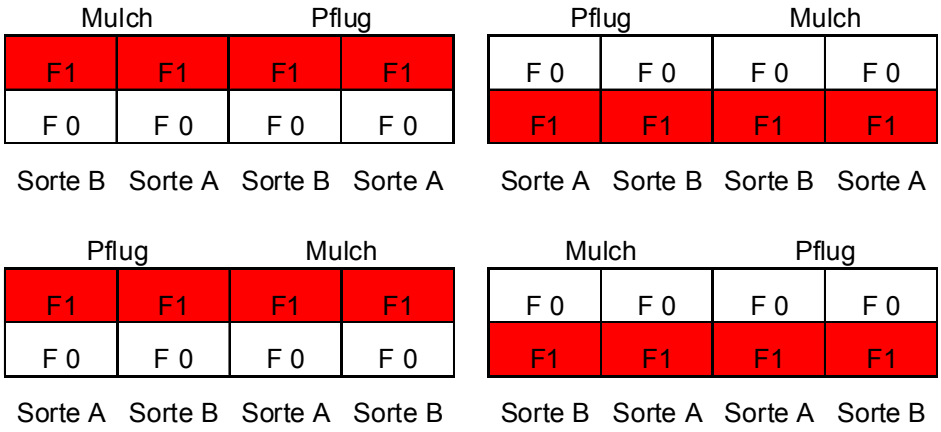


Abb.: Streifen-Spalt-Anlage

Hier existieren nun insgesamt fünf unterschiedliche Versuchseinheiten, die alle ihren eigenen Fehler zugewiesen bekommen. Diese zufälligen Fehlereffekte modellieren wiederum, das sich die Beobachtungen, die vom gleichen Groß-, Mittel- oder Kleinteilstück stammen ähnlich sind.

Zeile*Wdh = Großteilstück

Spalte*Wdh = Großteilstück

Zeile*Spalte*Wdh = Mittelteilstück

Spalte*Unterspalte*Wdh = Mittelteilstück

Zeile*Spalte*Unterspalte*Wdh = Kleinteilstück = Parzelle

MIXED-Code:

```
Proc mixed data = fusarium;
class BB Sorte Fungi WDH;
model DON =  Wdh BB|Sorte|Fungi / ddfm=kr;
random BB*Wdh Fungi*Wdh BB*Fungi*Wdh BB*Sorte*Wdh;
run;
```

*Anmerkung: der Bar-Operator "|“ steht für eine Kreuzklassifikation und dient als Kurzschreibweise. A|B bedeutet A + B + A*B*

10.5. Anlagen in unvollständigen Blöcken

Alpha-Gitter sind eine besondere Form von Anlagen in unvollständigen Blöcken. Hierbei befindet sich in jedem Block nur eine Teilmenge der Gesamtzahl von Prüfgliedern. Im folgenden Beispieldatensatz werden 24 Treatments auf 6 Blöcke je Wiederholung verteilt.

```
options nocenter nodate nonumber;
data alpha;
input rep   block     trt     y;
cards;
 1       1        11    4.1172
 1       1         4    4.4461
 1       1         5    5.8757
 1       1        22    4.5784
 1       2        21    4.6540
 1       2        10    4.1736
 1       2        20    4.0141
 1       2         2    4.3350
 1       3        23    4.2323
 1       3        14    4.7572
 1       3        16    4.4906
 1       3        18    3.9737
 1       4        13    4.2530
 1       4         3    3.3420
 1       4        19    4.7269
 1       4         8    4.9989
 1       5        17    4.7876
 1       5        15    5.0902
 1       5         7    4.1505
 1       5         1    5.1202
 1       6         6    4.7085
 1       6        12    5.2560
 1       6        24    4.9577
 1       6         9    3.3986
 2       1         8    3.9926
 2       1        20    3.6056
 2       1        14    4.5294
 2       1         4    4.3599
 2       2        24    3.9039
 2       2        15    4.9114
 2       2         3    3.7999
 2       2        23    4.3042
 2       3        12    5.3127
 2       3        11    5.1163
 2       3        21    5.3802
 2       3        17    5.0744
 2       4         5    5.1202
 2       4         9    4.2955
 2       4        10    4.9057
 2       4         1    5.7161
 2       5         2    5.1566
 2       5        18    5.0988
 2       5        13    5.4840
 2       5        22    5.0969
 2       6        19    5.3148
 2       6         7    4.6297
 2       6         6    5.1751
 2       6        16    5.3024
 3       1        11    3.9205
 3       1         1    4.6512
 3       1        14    4.3887
```

```
3       1       19    4.5552
3       2        2    4.0510
3       2       15    4.6783
3       2        9    3.1407
3       2        8    3.9821
3       3       17    4.3234
3       3       18    4.2486
3       3        4    4.3960
3       3        6    4.2474
3       4       12    4.1746
3       4       13    4.7512
3       4       10    4.0875
3       4       23    3.8721
3       5       21    4.4130
3       5       22    4.2397
3       5       16    4.3852
3       5       24    3.5655
3       6        3    2.8873
3       6        5    4.1972
3       6       20    3.7349
3       6        7    3.6096
 ;
 run;

/*Randomisierte Vollständige Blockanlage*/
 Proc mixed data=alpha;
 class rep trt;
 model y = rep trt / ddfm=kr;
 lsmeans trt / pdiff ;
 ods output diffs=RCB;
 run;

/*Gitteranlage, Blöcke fix*/
 Proc mixed data=alpha;
 class rep block trt;
 model y = rep block(rep) trt / ddfm=kr;
 lsmeans trt / pdiff ;
 ods output diffs=IBD_intra;
 run;

/*Gitteranlage, Blöcke zufällig*/
 Proc mixed data=alpha;
 class rep block trt;
 model y = rep  trt / ddfm=kr;
 random block(rep);
 lsmeans trt / pdiff ;
 ods output diffs=IBD_inter;
 run;

 title "Mittlerer Stderr der Differenz - Blockanlage";
 proc means data=RCB mean;
 var stderr; run;
 title "Mittlerer Stderr der Differenz - IBD fixe i_Blöcke";
 proc means data=IBD_intra mean;
 var stderr; run;
 title "Mittlerer Stderr der Differenz - IBD zuf. i_Blöcke";
 proc means data=IBD_inter mean;
 var stderr; run;
```

```
Mittlerer Stderr der Differenz ╦ Blockanlage

  Analysis Variable : StdErr Standard Error
        Mean          Minimum          Maximum
_____________________________________________
   0.2995396       0.2995396       0.2995396
_____________________________________________


Mittlerer Stderr der Differenz - IBD fixe i_Blöcke

  Analysis Variable : StdErr Standard Error
        Mean          Minimum          Maximum
_____________________________________________
   0.2766288       0.2643483       0.2857858
_____________________________________________


Mittlerer Stderr der Differenz - IBD zuf. i_Blöcke

  Analysis Variable : StdErr Standard Error
        Mean          Minimum          Maximum
_____________________________________________
   0.2707963       0.2617851       0.2772618
_____________________________________________
```

Bei einem Modell mit vollständigen Blöcken (Wiederholungen) sind die Standardfehler alle gleich. Die Tatsache, ob die Blöcke fix oder zufällig sind, spielt keine Rolle für den Stderr der Differenz.

Bei einem Modell welches vollständige Wiederholungen und unvollständige Blöcke enthält, sind die Standardfehler verschieden. Je nachdem, ob zwei Prüfglieder gemeinsam in einem unvollständigen Block standen oder nicht.

Wenn die unvollständigen Blöcke zufällig sind, so kann auch die Interblock-Information genutzt werden. Hierdurch sinkt der Standardfehler der Differenz ein wenig (Standardfehler der Mittelwerte steigt allerdings an!).

Es ist anzustreben, dass die Standardfehler der Differenz nicht zu unterschiedlich werden. Dazu müssen die Versuchspläne eine möglichst ausgeglichene Verteilung der Prüfglieder haben. Für Alpha-Designs gibt es Spezialsoftware (Alpha+, CycDesigN). Für Zwei- und Dreisatzgitter sind Basispläne in Lehrbüchern tabelliert (siehe z.B. Cochran & Cox, Experimental Design).

10.6. Grenzdifferenzen berechnen

MIXED erledigt zwar sehr schön die Berechnung der Standardfehler, man erhält aber keine Grenzdifferenzen. Man kann sich allerdings Vertrauensintervalle für Differenzen berechnen lassen. Eine Grenzdifferenz entspricht genau der halben Breite eines Vertrauensintervalls für eine Differenz.

Schritt 1: Mittelwerte und Differenzen in SAS-Datei ablegen.

```
/*Spaltanlage mit MIXED mit Mittelwertvergleichen*/
/*Diffs in Datei schreiben und GD berechnen*/
proc mixed data=alpha ;
class rep block trt ;
model y= rep  trt / ddfm=kr;
random block(rep);
lsmeans trt / pdiff adjust=tukey cl;
ods output lsmeans=lsmeans0815 diffs=diffs0815;
run;
```

Schritt 2: Aus Konfidenzintervallen für Vergleiche die LSD berechnen

```
/*Grenzdifferenz ist = halbe Breite
des Konfidenzintervalls für eine Differenz*/
data diffsneu;
set diffs0815;
LSD = (Upper - Lower)/2;
Tukey = (AdjUpper - AdjLower)/2;
keep trt _trt estimate stderr LSD sign_t Tukey sign_tuk;
if abs(estimate)>Tukey then sign_tuk="*";
else sign_tuk = "ns";
if abs(estimate)>LSD then sign_t="*";
else sign_t = "ns";
run;
```

Die erzeugten Dateien **work.lsmeans0815** und **work.diffsneu** können nun exportiert werden (z.B. in EXCEL).

11. Das Output delivery system (ODS)

Seit der SAS-Version 8 steht ODS zum Exportieren von Ergebnissen in SAS-Dateien zur Verfügung. Jedem Ergebnis, dass auf dem Bildschirm erscheint, ist ein spezieller „Table Name“ zugeordnet. Die Tabelle unten ist lediglich ein Auszug aus den verfügbaren Tables in Proc Mixed.

Tab.: ODS Tables Produced in PROC MIXED

Table Name	Description	Required Statement / Option
CovParms	estimated covariance parameters	default output
Diffs	differences of LS-means	LSMEANS / DIFF (or PDIFF)
Estimates	results from ESTIMATE statements	ESTIMATE
LRT	likelihood ratio test	default output
LSMeans	LS-means	LSMEANS
Slices	tests of LS-means slices	LSMEANS / SLICE=
SolutionF	fixed effects solution vector	MODEL / S
Tests1	Type 1 tests of fixed effects	MODEL / HTYPE=1
Tests3	Type 3 tests of fixed effects	default output
Type1	Type 1 analysis of variance	PROC MIXED METHOD=TYPE1
Type3	Type 3 analysis of variance	PROC MIXED METHOD=TYPE3

In der Regel ist man an den F-Tests der fixen Effekte, an den Varianzschätzungen für die zufälligen Effekte und an Mittelwerten von Faktorstufen und Faktorstufenkombinationen interessiert. Dieses ließe sich z.B. erreichen durch:

```
PROC MIXED ….;
MODEL …;
LSMEANS ….;
Ods output Covparms=CP Tests1=Tests1 LSMeans=LSMeans;
RUN; QUIT;
```

⇒ Es werden drei SAS-Dateien mit den Namen “CP”, “Tests1” und “LSMeans” erzeugt. Diese können dann z.B. in EXCEL exportiert werden.

```
PROC EXPORT DATA= WORK.CP
            OUTFILE= "C:\Ergebnisse\CP.xls"
            DBMS=EXCEL2000 REPLACE;RUN;
PROC EXPORT DATA= WORK.Tests1
            OUTFILE= "C:\Ergebnisse\Tests1.xls"
            DBMS=EXCEL2000 REPLACE;RUN;
PROC EXPORT DATA= WORK.LSMeans
            OUTFILE= "C:\Ergebnisse\LSMeans.xls"
            DBMS=EXCEL2000 REPLACE;RUN;
```

Es können auch der sonst im Output-Fenster erscheinenden Teil direkt in eine Datei geschrieben werden (siehe auch Abschnitt 1.6). Zum Beispiel im rtf-Format.

```
ods rtf body = „body.rtf“ ;
     […  Programmcode…]
ods rtf close;
```

Anstelle des Ziels body.rtf kann ein gewöhnlicher Datei-Pfad genannt werden.

12. Nichtparametrische Methoden: PROC NPAR1WAY

Es kann vorkommen, dass die Daten auch nach einer Transformation nicht die Voraussetzungen von varianzanalytischen und regressionsanalytischen Verfahren erfüllen.

Diese sind: Linearität/Additivität, Unabhängigkeit der Fehler, Varianzhomogenität der Fehler, Normalverteilung der Fehler

Bei Verletzung einer oder mehrerer dieser Voraussetzungen sollten alternative Verfahren eingesetzt werden. Die sogenannten NICHTPARAMETRISCHEN VERFAHREN basieren in der Regel auf Rängen (kleinster Wert = 1, zweitkleinster Wert = 2 usw.).
Wenn man es mit mehr als zwei Gruppen (Versuchsglieder) zu tun hat und die Stichproben unverbunden sind, so ist der Kruskal-Wallis-Test ein geeignetes Verfahren. Dieser ist in die SAS-Prozedur NPAR1WAY integriert.

Syntax Prozedur NPAR1WAY

PROC NPAR1WAY < options > ;
BY variables ;
CLASS variable ;
EXACT statistic-options < / computation-options > ;
FREQ variable ;
OUTPUT < OUT=SAS-data-set > < options > ;
VAR variables ;

Die Option “wilcoxon” weist im Zweistichprobenfall einen „Wilcoxon-Rangsummentest“, bei mehr als zwei Stichpoben den Kruskal-Wallis-Test an. Der Befehl „Exact“ ist wichtig wenn zu viele Rangbindungen (Daten mit gleichem Rang) vorliegen (mehr als 25% Bindungen) oder die Stichproben klein sind. Dann sollte man „exakt“ testen.

Beispiel für Kruskal-Wallis-Test

```
PROC NPAR1WAY wilcoxon data=versuch;
CLASS Sorte;
VAR Ertrag;
run;
```

The NPAR1WAY Procedure
Wilcoxon Scores (Rank Sums) for Variable Ertrag
Classified by Variable Sorte

Sorte	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
A	4	54.50	34.0	8.197561	13.6250
B	4	29.50	34.0	8.197561	7.3750
C	4	40.00	34.0	8.197561	10.0000
D	4	12.00	34.0	8.197561	3.0000

Average scores were used for ties.

Kruskal-Wallis Test

Chi-Square	10.7199
DF	3
Pr > Chi-Square	0.0133

Die Irrtumswahrscheinlichkeit liegt bei 1,3%. Also ist die Nullhypothese zu verwerfen.
Die Blockbildung kann im Kruskal-Wallis-Test nicht berücksichtigt werden. Hierfür wäre ein Friedman-Test adäquat. Dieser ist jedoch momentan nicht im Lieferumfang von SAS enthalten. Die Auswertung faktorieller Versuche ist ebenfalls nicht ohne weiteres möglich. Hier gibt es aber einige aktuelle Ansätze.

Weiterführende Literatur: BRUNNER, E. , MUNZEL, U. (2002): Nichtparametrische Datenanalyse. Springer, Heidelberg.

13. Kontingenztafeln: PROC FREQ

Mit Kontingenztafeln überprüft man Häufigkeiten. Sie sind also insbesondere zur Prüfung von Unterschieden oder Zusammenhängen zwischen nominalskalierten Daten geeignet. In SAS kann hierfür die Prozedur FREQ verwendet werden.

PROC FREQ < options > ;
BY variables ;
EXACT statistic-options < / computation-options > ;
OUTPUT < OUT=SAS-data-set > options ;
TABLES requests < / options > ;
TEST options ;
WEIGHT variable ;

Beispiel (nach Bärlocher 1998):
Ein neues Medikament wurde in seiner Wirkung mit Aspirin verglichen.

	Geheilt	Nicht geheilt	Summe
Aspirin	129	55	184
Neues Mittel	80	23	103
Summe	209	78	287

```
Data Daten;
Input Medikament$ Heilung$ Anzahl;
Datalines;
Aspirin    Geheilt   129
Aspirin    NichtGeh   55
Neues      Geheilt    80
Neues      NichtGeh   23
;
run;
Proc freq data=daten;
tables Medikament*Heilung/Chisq;
weight anzahl;
run;
```

Es wird ein Chi² von 1,91 berechnet. Dieses ist nicht signifikant. Die Nullhypothese einer gleichen Wirkung kann damit nicht abgelehnt werden.

Wichtig ist der Befehl „Weight“. Hier wird definiert, dass hinter jeder Kombination mehrere Fälle stehen. Ohne „Weight“ müsste eine Datenmatrix mit 287 Zeilen eingelesen werden.

Für den Befehl „TEST“ sind viele Optionen verfügbar. Hier können vor allem Übereinstimmungsmaße berechnet werden (**`Test Kappa`**; errechnet z.B. den Kappa-Koeffizienten).

Frequency Percent Row Pct Col Pct	Geheilt	NichtGeh	Total
Aspirin	129 44.95 70.11 61.72	55 19.16 29.89 70.51	184 64.11
Neues	80 27.87 77.67 38.28	23 8.01 22.33 29.49	103 35.89
Total	209 72.82	78 27.18	287 100.00

Statistics for Table of Medikament by Heilung

Statistic	DF	Value	Prob
Chi-Square	1	1.9076	0.1672
Likelihood Ratio Chi-Square	1	1.9452	0.1631
Continuity Adj. Chi-Square	1	1.5446	0.2139
Mantel-Haenszel Chi-Square	1	1.9009	0.1680
Phi Coefficient		-0.0815	
Contingency Coefficient		0.0813	
Cramer's V		-0.0815	

14. Generalisierte Lineare Modelle

Eine interessante Alternative zu Nichtparametrischen Methoden für den Fall, dass eine klassische Varianzanalyse nicht möglich ist, stellen die Generalisierten Linearen Modelle dar. Hierunter fällt z.B. auch die Logistische Regression.
Generalisierte Lineare Modelle (oder kurz GLMs) können für die Auswertung bestimmter landwirtschaftlicher Fragestellungen große Bedeutung haben. Bezüglich der theoretischen Hintergründe und weiterer Spezifika sei verweisen auf:

> *PIEPHO, H.P. (1998): Auswertung von Bonituren des Typs „Prozent Befall“ mit Hilfe von SAS Prozeduren für Generalisierte Lineare Modelle. Zeitschrift für Agrarinformatik 6, 26-37*

sowie

> *AGRESTI, A. (1996): An introduction to categorial data analysis. Wiley, New York.*

Hier soll lediglich an einem Beispiel die Umsetzung in SAS erläutert werden.

Beispiel: In einem Feld wurde der Befall von Mohrrüben durch Rübenfliegen in % befallene Pflanzen ermittelt. Der Versuch war als Blockanlage angelegt. Es wurden verschiedene Sorten und verschiedene Behandlungen geprüft.

```
data mohr;
input
genotype    pest_con    block     m     y;
datalines;
      1           1        1      53    44
      1           1        2      48    42
      1           1        3      51    27
      1           2        1      60    16
      1           2        2      52     9
      1           2        3      54    26
      2           1        1      48    24
      2           1        2      42    35
      2           1        3      52    45
      2           2        1      44    13
      2           2        2      48    20
      2           2        3      53    16
      3           1        1      49     8
      3           1        2      49    16
      3           1        3      50    16
      3           2        1      52     4
      3           2        2      51     6
      3           2        3      43    12
      4           1        1      51     4
      4           1        2      42     5
      4           1        3      46    12
      4           2        1      52    15
      4           2        2      56    10
      4           2        3      48     6
      5           1        1      52    11
      5           1        2      51    13
      5           1        3      44    15
      5           2        1      51     4
```

5	2	2	43	6
5	2	3	46	9
6	1	1	50	15
6	1	2	49	5
6	1	3	50	7
6	2	1	51	1
6	2	2	49	8
6	2	3	54	3
7	1	1	52	18
7	1	2	47	13
7	1	3	47	7
7	2	1	52	2
7	2	2	52	4
7	2	3	52	6
8	1	1	47	5
8	1	2	49	15
8	1	3	50	8
8	2	1	56	6
8	2	2	50	4
8	2	3	42	6
9	1	1	52	11
9	1	2	45	6
9	1	3	51	5
9	2	1	54	3
9	2	2	51	8
9	2	3	53	3
10	1	1	51	0
10	1	2	39	10
10	1	3	48	14
10	2	1	50	3
10	2	2	50	0
10	2	3	51	10
11	1	1	52	6
11	1	2	46	4
11	1	3	37	10
11	2	1	52	1
11	2	2	38	7
11	2	3	48	4
12	1	1	52	0
12	1	2	55	4
12	1	3	40	1
12	2	1	50	1
12	2	2	50	3
12	2	3	45	1
13	1	1	45	14
13	1	2	43	18
13	1	3	40	4
13	2	1	51	4
13	2	2	46	7
13	2	3	45	7
14	1	1	52	3
14	1	2	53	12
14	1	3	55	4
14	2	1	52	3
14	2	2	48	7
14	2	3	49	12
15	1	1	52	11

```
     15          1          2          54          6
     15          1          3          49          5
     15          2          1          50          2
     15          2          2          46          4
     15          2          3          53         14
     16          1          1          53          4
     16          1          2          40          1
     16          1          3          52          4
     16          2          1          56          4
     16          2          2          44          1
     16          2          3          42          3
;
run;
```

Die Auswertung kann in SAS mittels der Prozedur **GENMOD** erfolgen. Diese ist in ihrer Syntax den Prozeduren GLM und MIXED relativ ähnlich. Für die abhängige Variable kann eine „Events/Trials“ Schreibweise benutzt werden.
Es wird eine Type1-Schätzung durchgeführt. Hier ist die Reihenfolge der Effekte im Modell wichtig, deshalb werden zwei unterschiedliche Modelle angepasst. Für Prozentzahlen ist in aller Regel eine Binomialverteilung anzunehmen. Die Verbindung zwischen linearem Modell und den Daten stellt die sog. LINK-Funktion her. Für binomialverteilte Daten ist das der LOGIT-Link.

Der Anteil befallener Mohrrüben wird durch die Variable π ausgedrückt.

$$g(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i = \mu + \alpha_r + \tau_s + \nu_b$$

Unterschied zu einer herkömmlichen LOGIT-Tranformation ist, dass diese nicht auf die Daten selbst, sondern auf den Erwartungswert angewendet wird. Für die „neue“ abhängige Variable η wird nun ein lineares Modell angepasst. Diese „neue“ Variable nennt man auch linearer Prediktor. Die geschätzten Mittelwerte sind dann auf der transformierten Skala gemessen, können also theoretisch auch wieder rücktransformiert werden.
Die Anpassung des Modells an die Daten erfolgt über Maximum Likelihood. Die Güte der Anpassung wird mit der Devianz gemessen. Die Devianz ist jeweils die Differenz zwischen dem zweifachen der maximal erreichbaren Likelihood und dem zweifachen der im Modell erreichten Likelihood. Vereinfacht gesagt ist Devianz so etwas wie die SQ in der Varianzanalyse. Effekte mit großer Devianzreduktion haben einen großen Einfluss. Die Signifikanz der Effekte wird über eine Chi²-Statistik geprüft.
Zu einer Devianz-Tabelle gelangt man durch sukzessiven Modellaufbau. Im Beispiel wird zunächst ein Modell angepasst, welches nur den allgemeinen Effekt φ (**INTERCEP**) im linearen Prediktor hat. Hierfür ist die Devianz 901,7. Für ein Modell mit Blockeffekten (**BLOCK**) neben dem allgemeinen Effekt φ beträgt die Devianz 885,3. Die Reduzierung der Devianz beträgt somit 16,4. Die Devianzreduktion kann herangezogen werden, um die Nullhypothese zu testen, dass keine Blockeffekte bestehen. Unter der Nullhypothese ist die Devianzreduktion (asymptotisch) verteilt wie χ^2 mit 2 Freiheitsgraden. Die Anzahl der Freiheitsgrade ergibt sich aus der Anzahl der zusätzlichen Parameter beim Aufbau des Modells. Da der Versuch 3 Blocks hat, werden 3 – 1 = 2 zusätzliche Parameter in das Modell eingeführt. In den nächsten Schritten werden dann weitere Parameter aufgenommen: Hauptwirkungen für Insektizidbehandlung (**PEST_CON**), Hauptwirkungen für Sorten (**GENOTYPE**) und Interaktio-

nen (**GENOTYPE* PEST_CON**). In jedem Schritt wird die Devianzreduktion gegen die χ^2-Verteilung geprüft. Im Beispiel sind alle Effekt signifikant.

```
PROC GENMOD DATA=mohr;
CLASS genotype pest_con block;
MODEL y/m=block pest_con|genotype
          /LINK=logit DIST=bin TYPE1; RUN;

PROC GENMOD DATA=mohr;
CLASS genotype pest_con block;
MODEL y/m=block genotype|pest_con
          /LINK=logit DIST=bin TYPE1; RUN;
```

Output (gekürzt):

```
 Criteria For Assessing Goodness Of Fit

Criterion           DF    Value   Value/DF
Deviance            62   208.1402  3.3571
Scaled Deviance     62   208.1402  3.3571
Pearson Chi-Square  62   199.3440  3.2152
Scaled Pearson X2   62   199.3440  3.2152
Log Likelihood       . -1937.6624   .

    LR Statistics For Type 1 Analysis

Source             Deviance DF ChiSquare Pr>Chi
INTERCEPT          901.7047  0     .      .
BLOCK              885.3002  2   16.4045 0.0003
PEST_CON           794.6180  1   90.6822 0.0001
GENOTYPE           277.0534 15  517.5646 0.0001
GENOTYPE*PEST_CON  208.1402 15   68.9132 0.0001

Source             Deviance  DF ChiSquare Pr>Chi
INTERCEPT          901.7047  0     .      .
BLOCK              885.3002  2   16.4045 0.0003
GENOTYPE           384.7777 15  500.5225 0.0001
PEST_CON           277.0534  1  107.7243 0.0001
GENOTYPE*PEST_CON  208.1402 15   68.9132 0.0001
```

14.1. Überdispersion

Es wurde bisher von einer Binomialverteilung ausgegangen. Diese Annahme impliziert eine definierte Beziehung zwischen Erwartungswert und Varianz. In der Praxis beobachtet man aber oft, dass die Varianz größer ist, als nach diesem Modell zu erwarten wäre. Dieses Phänomen nennt man Überdispersion.
In einer Blockanlage ist Überdispersion aufgrund des Parzellenfehlers zu vermuten, der bisher außer Acht gelassen wurde. Es ist anzunehmen, dass die Befallswahrscheinlichkeiten von Parzelle zu Parzelle schwanken (Parzellenfehler), selbst dann, wenn keine Behandlungsunterschiede bestehen. Wird eine solche Annahme gemacht, so ist die Schwankung zwischen den Befallswerten innerhalb eines Blocks größer, als es nach der Binomial-Verteilung anzunehmen ist. Die Beziehung zwischen Erwartungswert und Varianz bleibt auch bei Überdispersion bestehen, allerdings modifiziert durch einen Proportionalitätsfaktor ϕ, der auch als Überdispersionsparameter bezeichnet werden kann. Dieser geschätzte Überdispersionsparameter wird benutzt, um die Devianzreduktion und die Standardfehler für Parameterschätzungen zu korrigieren: Die Devianzreduktion wird durch $\hat{\phi}$ geteilt, die Varianz-Kovarianz-Matrix der Parameterschätzungen wird mit $\hat{\phi}$ multipliziert.

SAS-Anweisungen und Devianz-Tabellen für Verrechnung der Mohrrübendaten als GLM (Logit-Link, Binomialverteilung, Überdispersion, Skalenparameter mit Pearsonscher Chi-Quadrat Statistik geschätzt).

```
PROC GENMOD DATA=mohr;
CLASS genotype pest_con block;
MODEL y/m=block pest_con|genotype
          /LINK=logit DIST=bin TYPE1 SCALE=pearson; RUN;

PROC GENMOD DATA=mohr;
CLASS genotype pest_con block;
MODEL y/m=block genotype|pest_con
          /LINK=logit DIST=bin TYPE1 SCALE=pearson; RUN;
```

Output (gekürzt):

```
Criteria For Assessing Goodness Of Fit

Criterion           DF     Value  Value/DF
Deviance            62  208.1402    3.3571
Scaled Deviance     62   64.7358    1.0441
Pearson Chi-Square  62  199.3440    3.2152
Scaled Pearson X2   62   62.0000    1.0000
Log Likelihood       . -602.6522     .

LR Statistics For Type 1 Analysis

Source           Deviance NDF ChiSquare Pr>Chi
INTERCEPT          901.70   0      .     .
BLOCK              885.30   2     5.10  0.0780
PEST_CON           794.62   1    28.20  0.0001
GENOTYPE           277.05  15   160.97  0.0001
GENOTYPE*PEST_CON  208.14  15    21.43  0.1235

Source           Deviance NDF ChiSquare Pr>Chi
INTERCEPT          901.70   0      .     .
BLOCK              885.30   2     5.10  0.0780
```

```
GENOTYPE           384.78   15   155.67   0.0001
PEST_CON           277.05    1    33.50   0.0001
GENOTYPE*PEST_CON  208.14   15    21.43   0.1235
```

Der Schätzwert für den Überdispersionsparameter ϕ ist der Zeile **Pearson Chi-Square** unter der Überschrift **Value/DF** zu entnehmen und beträgt $\hat{\phi}$ = 3,2152. Nun sind die Interaktionen nicht mehr signifikant, sondern nur noch die Hauptwirkungen.

14.2. Ein weiteres Beispiel für Generalisierte Lineare Modelle

Wir betrachten die Wahrscheinlichkeit Raps pfluglos zu bestellen in Abhängigkeit des Bundeslandes und der Bodenart (hypothetische Daten).

	Niedersachsen		Mecklenburg-Vorpommern		
	Leichte Böden	Schwere Böden	Leichte Böden	Schwere Böden	Summe
Mit Pflug	129	101	286	58	574
Ohne Pflug	40	76	97	27	240
Summe	169	177	383	85	814

In Niedersachsen bestellen 116 von 346 Betrieben ihren Raps pfluglos = 33,5%.
In Mecklenburg sind dieses 124 von 468 Betrieben = 26,5%.
Auf den ersten Blick scheint die pfluglose Rapsbestellung in Mecklenburg-Vorpommern damit weniger verbreitet zu sein als in Niedersachsen. Es ist jedoch zu berücksichtigen, dass in Mecklenburg leichte Böden vorherrschen und es eine Beziehung zwischen Bodenart und Bodenbearbeitung gibt.
Auf leichten Böden bestellen 137 von 552 Betrieben ihren Raps pfluglos = 24,8%.
Auf schweren Böden bestellen 103 von 262 Betrieben ihren Raps pfluglos = 39,3%.
Die Bedeutung der Bodenart ist damit scheinbar größer als die des Landes. Die berechneten Prozentzahlen sind jedoch infolge der offensichtlichen Assoziation zwischen Bodenart und Bundesland verzerrt. Ziel einer statischen Analyse sollte deshalb sein, die beiden Effekte „Bodenart“ und „Land“ in ihrer Wirkung auf die Wahl der Bodenbearbeitung unabhängig voneinander betrachten zu können. Die mittleren „pfluglos“-Anteile für die beiden Bodenarten sollen so geschätzt werden, als wenn die Verteilung der Böden in beiden Ländern der Rand-Verteilung entspräche, nämlich 552 zu 262 = 67,8% leichte Böden und 32,2% schwere Böden.
Diese Adjustierung kann für normalverteilte Variablen (z.B. Ertrag) mit einem Linearen Modell (Kleinst-Quadrat-Schätzung, Varianzanalyse) und für Anteile (Prozentzahlen) mit einem Generalisierten Linearen Modell erreicht werden, wie es in der Prozedur GENMOD des SAS-Paketes zur Verfügung steht. Die Anpassung des Modells erfolgt hierbei nicht auf der Originalskala sondern auf einer anderen Skala, auf der Linearität und Additivität gegeben ist, der so genannten Logit-Skala.

```
data test;
input Boden$ Land$ pfluglos gesamt;
cards;
leicht Nds 40 169
schwer Nds 76 177
leicht MVP 97 383
schwer MVP 27 85
;
run;

options nocenter;
proc genmod data=test;
class Boden Land;
model pfluglos/gesamt = Land Boden Boden*Land/
dist=bin link=logit type1 type3 scale=pearson;
lsmeans Land Boden Boden*Land / cl;
ods output lsmeans=lsmeans;
run;
```

Wir erhalten folgenden Output:

LR Statistics For Type 1 Analysis

Source	Deviance	DF	Chi-Square	Pr > ChiSq
Intercept	20.7336			
Land	16.0286	1	4.70	0.0301
Boden	2.6651	1	13.36	0.0003
Boden*Land	0.0000	1	2.67	0.1026

LR Statistics For Type 3 Analysis

Source	DF	Chi-Square	Pr > ChiSq
Land	1	1.24	0.2663
Boden	1	11.42	0.0007
Boden*Land	1	2.67	0.1026

Nach dem Typ1-Test ist das Land signifikant nach dem Typ3-Test nicht. Dieses widersprüchliche Resultat ergibt sich aus der unterschiedlichen Philosophie dieser Testverfahren. Nach Typ1 werden Effekte nur hinsichtlich der Effekte adjustiert, die im Modell vor ihnen stehen. Der Test von Boden in der Typ1-Tabelle ist also nicht adjustiert. Hier wird direkt getestet, dass in Niedersachsen 33% pfluglos arbeiten während dieses in Mecklenburg nur 26% sind. Der Test des Bodens ist jedoch hinsichtlich der Länder adjustiert und ebenfalls signifikant. Die Interaktion ist nicht signifikant. Im Typ3-Test sind alle Effekte hinsichtlich aller übrigen adjustiert. Nur der Boden ist signifikant.
Man kann einen adjustierten Test für „Land“ nach der Typ1-Methode erhalten wenn man diesen Effekt hinter Boden anpasst. Er ist nicht signifikant, obwohl die Differenz zwischen Niedersachsen und Mecklenburg 33-26=7% beträgt!

LR Statistics For Type 1 Analysis

Source	Deviance	DF	Chi-Square	Pr > ChiSq
Intercept	20.7336			
Boden	3.2252	1	17.51	<.0001
Land	2.6651	1	0.56	0.4542
Boden*Land	0.0000	1	2.67	0.1026

Da die Interaktion nicht signifikant ist, reduzieren wir das Modell abschließend auf die beiden Haupteffekte und berechnen adjustierte Mittelwerte auf der Logit-Skala und per Rücktransformation auf der Originalskala. Die Wahrscheinlichkeiten sind in der Spalte „p“ im unteren Teil des Outputs abzulesen. Die Differenz zwischen den Ländern ist durch die Adjustierung auf 33% in Niedersachsen und 30% in Mecklenburg zusammengeschrumpft. Der vermeintliche Landeseffekt von 7% kam tatsächlich nur durch die unterschiedliche Situation bezüglich der Böden zustande und ist nun korrigiert.

```
proc genmod data=test;
class Boden Land;
model pfluglos/gesamt = Land Boden /
dist=bin link=logit type3 scale=pearson;
lsmeans Land Boden  / cl;
ods output lsmeans=lsmeans;
run;
data lsmeans;
set lsmeans;
p=exp(estimate)/(1+exp(estimate));
p_ucl = exp(uppercl)/(1+exp(uppercl));
p_lcl = exp(lowercl)/(1+exp(lowercl));run;
data lsmeans;
set lsmeans;
keep boden land p p_ucl p_lcl;run;
proc print data=lsmeans;run;
```

LR Statistics For Type 3 Analysis

Source	Num DF	Den DF	F Value	Pr > F	Chi-Square	Pr > ChiSq
Land	1	1	0.21	0.7247	0.21	0.6444
Boden	1	1	5.08	0.2658	5.08	0.0242

Least Squares Means

Effect	Boden	Land	Estimate	Standard Error	DF	Chi-Square	Pr>ChiSq	Alpha	ConfidenceLimits	
Land		MVP	-0.8332	0.1878	1	19.69	<.0001	0.05	-1.2012	-0.4652
Land		Nds	-0.7082	0.1875	1	14.26	0.0002	0.05	-1.0758	-0.3406
Boden	leicht		-1.0849	0.1671	1	42.13	<.0001	0.05	-1.4125	-0.7573
Boden	schwer		-0.4565	0.2110	1	4.68	0.0305	0.05	-0.8701	-0.0429

Obs	Boden	Land	p	p_ucl	p_lcl
1		MVP	0.30296	0.38575	0.23125
2		Nds	0.33000	0.41567	0.25431
3	leicht		0.25258	0.31924	0.19584
4	schwer		0.38782	0.48927	0.29524