Ivo Züchner

# BlueSky Statistics – eine Handreichung zur Anwendung der graphischen Benutzeroberfläche für die Softwareumgebung R

*Stand September 2020*

## Vorwort

Mit diesem Text soll die Statistik-Software *BlueSky Statistics* vorgestellt und eine erste Einführung zu deren Anwendung für interessierte Nutzer*innen gegeben werden. Entstanden ist dies aus Arbeitszusammenhängen an der Philipps-Universität Marburg, über die Anforderungen von Statistikseminaren und in der Zusammenarbeit mit studentischen Hilfskräften.

So ist diese Einführung für interessierte Statistikanwender*innen geschrieben, die mit oder ohne Erfahrung in proprietärer Statistik-Software (wie bspw. SPSS©) einfache oder komplexere statistische Auswertungen mit quantitativen Datensätzen oder auch Ergebnisgrafiken erstellen wollen. Bislang gibt es keine systematische schriftliche Einführung in BlueSky Statistics – allerdings wohl einige Präsentationen und Trainingsvideos auf der eigenen Website https://www.BlueSky Statistics.com und auch auf youtube, daher habe ich hier ein paar Grundlagen zusammengestellt, um Interessierten den Einstieg zu erleichtern. Die nachfolgende inhaltliche Beschreibung lehnt sich an ein Online-Review von Robert A. Muenchen von der University of Tennessee („A Comparative Review of the BlueSky Statistics GUI for R“ (Muenchen 2020a)) an, der sich schon seit längerer Zeit die Mühe macht, grafische Nutzeroberflächen (GUIs) für R zu beschreiben und zu vergleichen. Er hat auch einen sich noch in Erstellung befindenden sehr umfangreichen „User Guide“ für BlueSky Statistics online gestellt (vgl. Muenchen 2020b). Ein Buch über den Einsatz von Raschmodellen mit R und BlueSky Statistics in den Sozialwissenschaften hat Lamprianou (2019) publiziert.

Selbstverständlich ersetzt dieser Text weder eine systematische Einführung in Statistik bzw. in die für die nachfolgende erwähnten statistischen Verfahren, hier wird nur die Anwendung in BlueSky Statistics beschrieben. Auch kann hiermit keine Einführung in R gegeben werden, für einen praktischen Einstieg inklusive Hintergründe und Erklärung der Logik der Programmiersprache vgl. bspw. bspw. Luhmann (2016).

Das vorliegende Dokument bezieht sich auf die Version 7.1 von BlueSky Statistics.

# Inhalt

## Einleitung

*BlueSky Statistics* (im Folgenden auch BSS abgekürzt) ist eine Benutzeroberfläche für die Statistik-Software-Umgebung R. Sie ist sowohl in einer Open-Source Variante als auch in einer kommerziellen Variante (die technischen Support und eine Version für Windows-Terminalserver wie Remote Desktop oder Citrix enthält) erhältlich (Download unter https://www.BlueSkyStatistics.com/Articles.asp?ID=301), derzeit wohl nur für Microsoft Windows. Hinter BSS steht die Firma BlueSky Statistics mit Sitz in Chicago – über die allerdings nicht viel im Internet herauszubekommen ist.

BSS bietet sich für Benutzer*innen an, die mit R arbeiten möchten und bislang vorwiegend Erfahrungen mit SPSS, Stata oder SAS gesammelt haben und/oder vor der R-Syntax zunächst zurückschrecken. Mit BSS kann eine große Bandbreite an Datenmanagement und Auswertungsverfahren als auch grafischen Darstellungen über das Menü aufgerufen werden, die Logik des Programms ist dabei sehr an SPSS orientiert.[1] München (2020) schreibt, dass die Entwickler*innen zum Teil von SPSS kommen und das sieht man dem Programm schnell an, viele Funktionalitäten sind ähnlich oder gleich angelegt, auch wenn die Befehlsstruktur auf R basiert.

BlueSky Statistics bietet sowohl die Möglichkeit, per „point & klick" statistische Auswertungen Datenmanagement und Grafiken vorzunehmen oder diese über R-Syntax zu erstellen. Die grafische Benutzeroberfläche (GUI) orientiert sich in Aussehen, Aufbau und Funktionalität sich stark an SPSS©, so dass gerade SPSS-erfahrende Anwender*innen wenig Einstiegsschwierigkeiten haben sollten.

Die Nutzung von BlueSky Statistics bietet sich an

- wenn Kosten von proprietärer Software gespart werden soll
- wenn sowohl eine vereinfachte Anwendung mit einer GUI angestrebt wird, aber gleichzeitig die vielen Möglichkeiten und Pakete von R genutzt werden sollen
- wenn ein Übergang von SPSS zu R angestrebt wird
- wenn schon mit R gearbeitet wird, aber gerade der Transfer und Weiterbearbeitung von Ausgaben in Office-Programme Probleme bereitet
- wenn mit R gearbeitet werden soll, aber noch viel Datenaufbereitung notwendig ist und R noch nicht gut beherrscht wird
- wenn es in kooperativen Arbeitsformen wichtig ist, die jeweiligen Arbeitsschritte zu dokumentieren und wiederholen zu können.

Unbestritten ist, dass es hierfür auch andere Möglichkeiten gibt - wie bspw. R Commander oder jamovi - aber BSS ist eine funktionale Option.

[1] An dieser Stelle soll nicht die Diskussion geführt werden, ob es nicht sinnvoller ist, R ohne ein Klick- & Point-Menü zu erlernen, weil dieses ein tieferes Verständnis erzeuge und langfristig für die Ausschöpfung der Möglichkeiten von R notwendig ist. Von BSS ausgehend ist ebenfalls ein weiterer Einstieg in R möglich, und gerade für die Arbeit im Team bzw. für Personen, die nicht längerfristig mit statistischen Auswertungen arbeiten, sprechen nach der Meinung des Verfassers die leichte Zugänglichkeit und Vergleichbarkeit mit SPSS, die gute Reproduzierbarkeit der Operationen, ein recht einfaches Datenmanagement als auch der einfache Export von Ergebnissen für die Nutzung von BSS.

In BlueSky Statistics kann als Syntax immer der „klassische“ R-Code verwendet werden– dann erscheint im Ausgabefenster auch „nur“ die klassische R-Ausgabe. Für elegantere und vor allem leichter exportierbare Ausgaben verwendet BlueSky Statistics zum Teil modernere und teilweise auch spezifisch auf BSS angepassten R-Code (s.u.).

# Teil 1: Übersicht über Funktionen und „Handling“ des Programms

## 0. Installation und Aufbau des Programms/ R-Pakete

Die *Hauptinstallation* von BlueSky kann nach dem Download (von https://www.BlueSky Statistics.com/) in einem einzigen Schritt durchgeführt werden. Das Installationsprogramm beinhaltet auch eine Version von R, die die Kompatibilität zwischen BlueSky und dem R Grundprogramm gewährleistet. Diese wird auch dann installiert, wenn sich auf dem Rechner schon eine R Installation befindet.

Zentrale operative Basis von BlueSky Statistics ist R bzw. die Paketstruktur von R. BlueSky Statistics zielt darauf, R „anwendungsfreundlich“ über die Menüführung zugänglich zu machen, kann aber auch über Syntax/R-Code gesteuert werden. Mit dem Syntaxfenster (s.u.) von BlueSky Statistics kann über die in der Menüführung angebotenen Optionen der volle Funktionumfang von R genutzt werden, wenn entsprechende weitere R-Pakete (über die Menüleiste oder die Syntax) installiert werden. Für Pakete, die noch keinen Eingang in die Menüführung gefunden haben muss allerdings mit der Syntax wie in R oder RStudio gearbeitet werden. Die Befehle werden hierfür im Syntaxfenter eingegeben, der Output findet sich im Outputfenster (s.u.), dieser ist dann allerdings nicht extra formatiert und entspricht dann den Ausgaben, wie sie auch in R oder RStudio ausgegeben werden.

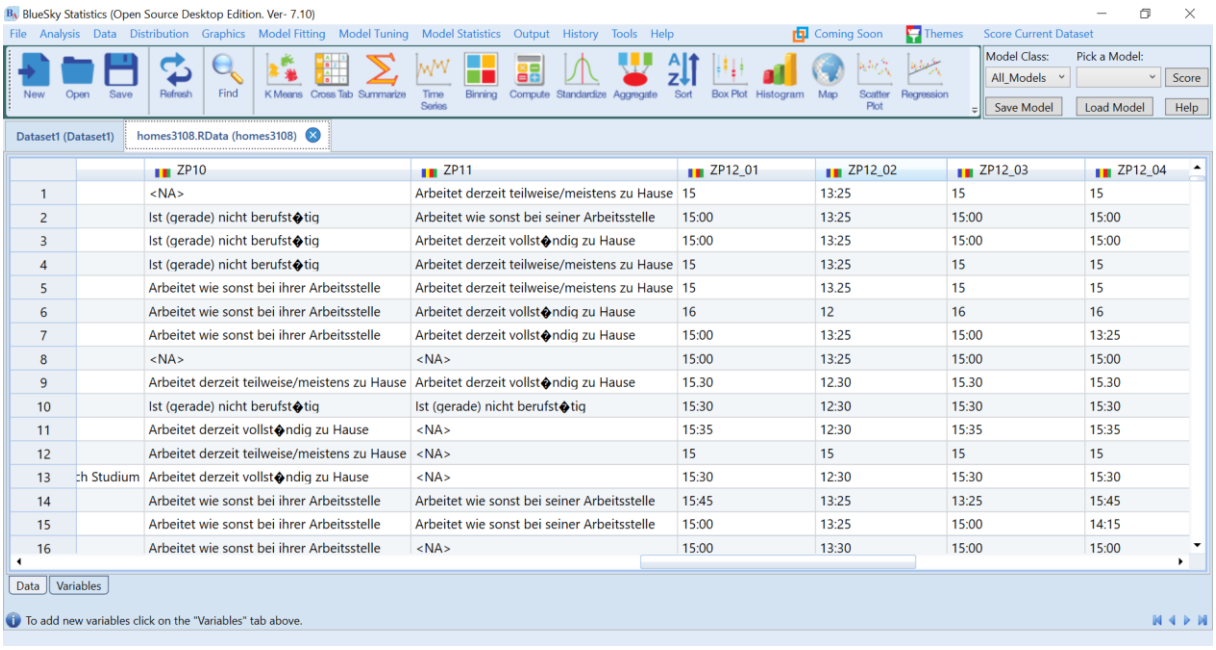
Gleichzeitig arbeitet BlueSky Statistics mit Add-on-Modulen. Die Add-ons dienen dazu, mögliche Auswertungsroutinen aus R-Paketen auch in die GUI des Programms aufzunehmen. Unter https://www.BlueSky Statistics.com/category-s/122.htm finden sich diese als „extensions“ zum Download für die Erweiterung der Menüs – darunter sind auch einige, die derzeit nicht in der Open-Source-edition enthalten sind. Add-ons können aber auch selbst erstellt werden.

## I. Die grafische Benutzeroberfläche (Graphical User interface – GUI)

Beim Start von BlueSky Statistics erscheint als Oberfläche der Dateneditor, der wie SPSS zwei Ebenen hat (Datenansicht und Variablenansicht) – und je nach Wahl auch schon das Ausgabefenster.

### I.1 Dateneditor

BlueSky Statistics beginnt, indem es den Hauptbildschirm mit der Datenansicht (Abbildung 1) anzeigt. Hier können Daten direkt in den leeren Tabellenkalkulations-Dateneditor eingegeben werden. Unten links im Dateneditor-Bildschirm befinden sich zwei Registerkarten mit den Bezeichnungen "Data" und "Variables". Die Registerkarte "Data" wird standardmäßig angezeigt, aber wenn Sie auf die Registerkarte "Variables" klicken, kommt man zur Variablenansicht, die die Metadaten anzeigt (Abbildung 2).




| | | ZP10 | ZP11 | ZP12_01 | ZP12_02 | ZP12_03 | ZP12_04 |
|---|---|---|---|---|---|---|---|
| 1 | | <NA> | Arbeitet derzeit teilweise/meistens zu Hause | 15 | 13:25 | 15 | 15 |
| 2 | | Ist (gerade) nicht berufst�tig | Arbeitet wie sonst bei seiner Arbeitsstelle | 15:00 | 13:25 | 15:00 | 15:00 |
| 3 | | Ist (gerade) nicht berufst�tig | Arbeitet derzeit vollst�ndig zu Hause | 15:00 | 13:25 | 15:00 | 15:00 |
| 4 | | Ist (gerade) nicht berufst�tig | Arbeitet derzeit teilweise/meistens zu Hause | 15 | 13:25 | 15 | 15 |
| 5 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | Arbeitet derzeit teilweise/meistens zu Hause | 15 | 13.25 | 15 | 15 |
| 6 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | Arbeitet derzeit vollst�ndig zu Hause | 16 | 12 | 16 | 16 |
| 7 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | Arbeitet derzeit vollst�ndig zu Hause | 15:00 | 13:25 | 15:00 | 13:25 |
| 8 | | <NA> | <NA> | 15:00 | 13:25 | 15:00 | 15:00 |
| 9 | | Arbeitet derzeit teilweise/meistens zu Hause | Arbeitet derzeit vollst�ndig zu Hause | 15.30 | 12.30 | 15.30 | 15.30 |
| 10 | | Ist (gerade) nicht berufst�tig | Ist (gerade) nicht berufst�tig | 15:30 | 12:30 | 15:30 | 15:30 |
| 11 | | Arbeitet derzeit vollst�ndig zu Hause | <NA> | 15:35 | 12:30 | 15:35 | 15:35 |
| 12 | | Arbeitet derzeit teilweise/meistens zu Hause | <NA> | 15 | 15 | 15 | 15 |
| 13 | ch Studium | Arbeitet derzeit vollst�ndig zu Hause | <NA> | 15:30 | 12:30 | 15:30 | 15:30 |
| 14 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | Arbeitet wie sonst bei seiner Arbeitsstelle | 15:45 | 13:25 | 13:25 | 15:45 |
| 15 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | Arbeitet wie sonst bei seiner Arbeitsstelle | 15:00 | 13:25 | 15:00 | 14:15 |
| 16 | | Arbeitet wie sonst bei ihrer Arbeitsstelle | <NA> | 15:00 | 13:30 | 15:00 | 15:00 |

Abb. 1: Hauptansicht, Fenster 1: Datenansicht.

Es können gleichzeitig mehrere Datensätze geöffnet werden, diese werden dann als Reiter angezeigt. Der jeweils helle Reiter zeigt den für die Menüführung aktuellen Datensatz an.

### I.1.1 Dateneingabe

Mit dem Neustart von BlueSky wird auch ein leeres Datenfenster geöffnet (oder kann mit dem Menü über „File > New“ angefordert werden. Alle Variablen tragen zunächst den Namen var1 var2 usw. Eingegeben werden können zahlen oder Buchstaben, mit „copy to clipboard“ (rechte Maustaste!) können auch Tabellen aus anderen Programmen in das Datenfenster eingefügt werden. Auch können im Datenfenster neue Zeilen eingefügt werden. Bei Faktorvariablen macht es Sinn, diese zunächst als Werte einzugeben. In der Variablenansicht kann dann daraus ein Faktorgemacht und die Ausprägungen als values definiert werden.

Die *Datenspeicherung* erfolgt im Menü mit “File > Save As”, wobei BSS wie SPSS oder Stata jeden Datensatz – im Gegensatz zu R – einzeln speichert). BSS erkennt aber R-Datenfiles mit mehreren Objekten und bietet an, diese getrennt zu speichern.

Die Bearbeitung der Variablen erfolgt – wiederum anlog zu SPSS – in der Variablenansicht (vgl. Abb. 2): Hier können neue Variablen angelegt werden und Datenformate und Anzeigen sowie die Wertelabels festgelegt werden.

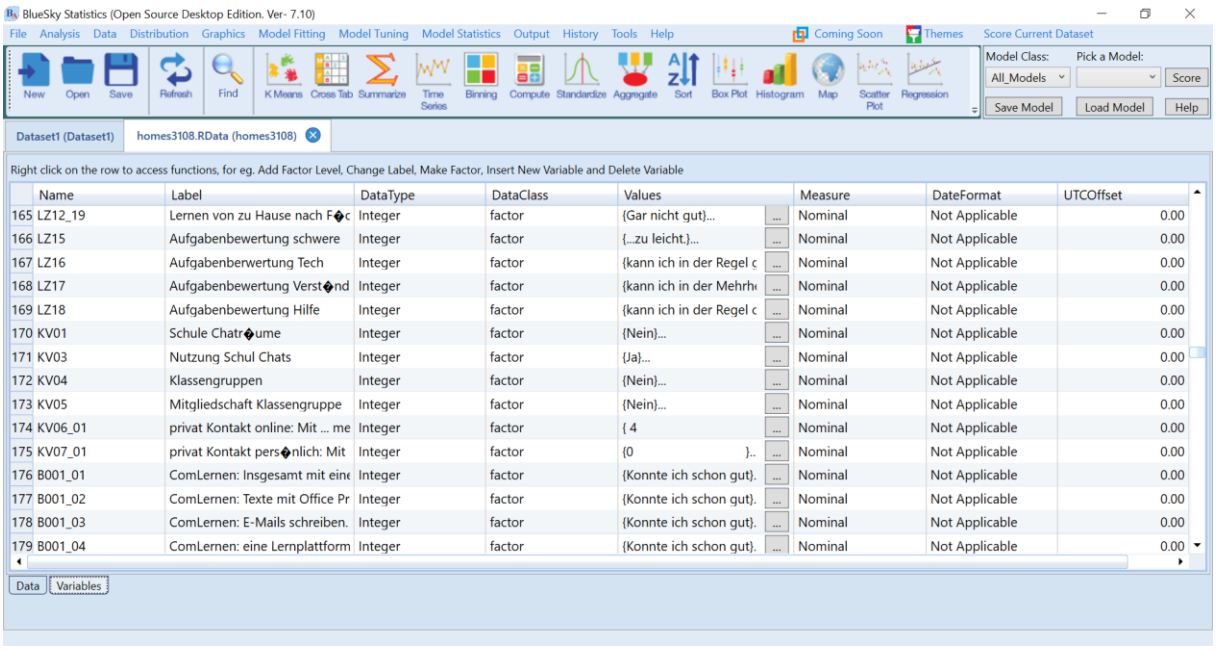



Abb. 2: Die Hauptansicht; Fenster 2: Variablenansicht

Unter der **Variablenansicht** kann – wie bspw. in SPSS – jede einzelne Variable näher definiert werden:

- Name = Kurzname (bspw. Fragebogennummer, sinnvoll sind kurze Namen, gerade wenn auch mit der Syntax gearbeitet wird)
- Label= Langname (frei formulierbar)
- DataTyp= Feindifferenzierung der Datenart, es gibt bei R hier eine Differenzierung numerisch in „integers“ (ganze Zahlen, gilt auch für Faktoren) und double („Gleitkommazahlen“) gemacht
- DataClass= Hier wird zwischen Faktor (gestufter, kategorialer Variable), Vektor (metrischer Variable) oder Character/String (Stringvariable=Zeichenvariable) unterschieden.[2]
- Values= Namen der Ausprägungen bei ordinalen und kategorialen Daten, z.B. 1= Mann und 2 = Frau) (Wertelabels bei SPSS), diese können im Fenster selbst geändert werden, allerdings nur bei Faktoren.
- Measure = Beschreibt das Skalenniveau, mit den klassischen drei Ausprägungen „numeric“, „scale“ und „ordinal“.

Die Variablen können über die rechte Maustaste mit „make numeric oder „make factor“ oder mit „make string“ in ihrer „DataClass“ geändert werden. Falls ein ordinales Skalenniveau festgelegt werden soll (bspw. für eine ordinale Regression), muss dies mit einem R-Syntax Befehl erfolgen (vgl. den Punkt Ordinale Regression).

### I.1.2 Datenimport

Bestehende Daten können leicht in BSS importiert werden, dies geschieht dateiformatübergreifend einfach über das Menü mit “File > Open”.

[2] R unterscheidet zwischen Faktorvariablen (abgestufte Variablen) und Vektorvariablen (metrischen Variablen).

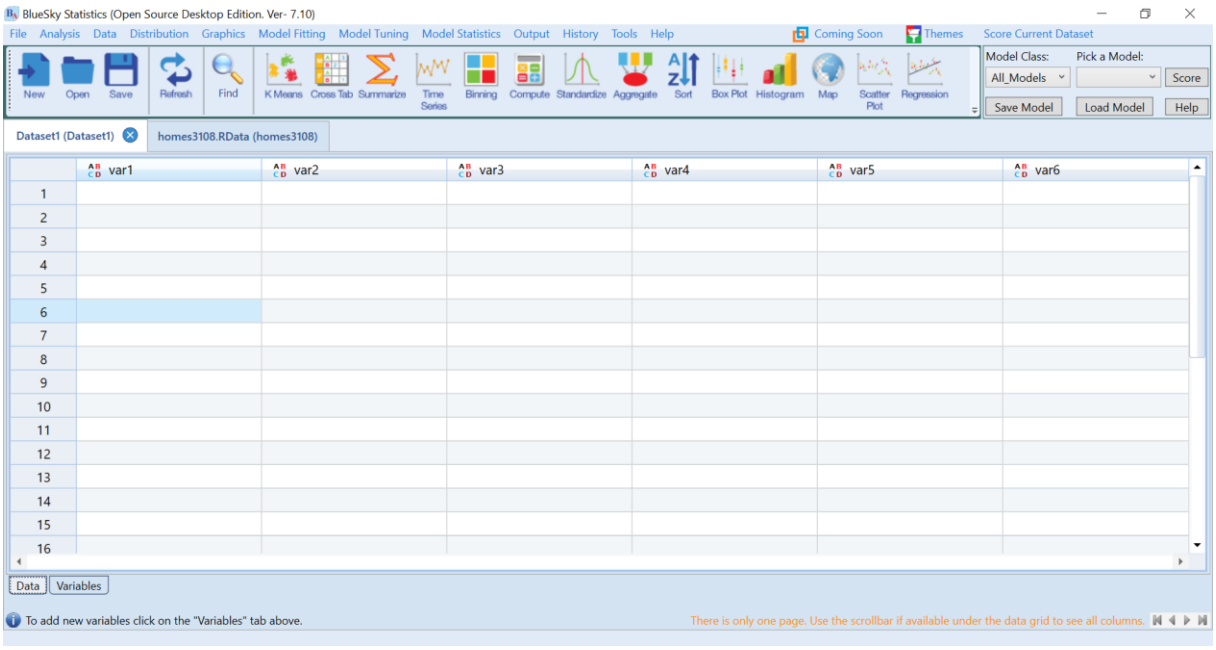



Abb. 3: Öffnen von bestehenden Daten (verschiedener Formate)

Die Open-Source Version von BlueSky Statistics unterstützt dabei die folgenden Formate

- Comma Separated Values (.csv)
- Einfache Textdateien (.txt)
- Excel (alte xls und neue xlsx Dateien)
- dBase's (dbf)
- SPSS (.sav)
- SAS binary files (sas7bdat)
- Standard R workspace Dateien (RData)

Dabei kann die importierte Datendatei dann als R workspace Datei (RData) abgespeichert werden, dies muss aber nicht erfolgen und eine Speicherung im alten Format ist möglich. SQL database Formate müssen über das Menü mit "File > Import Data" importiert werden.

### I.1.3 Datenexport

BlueSky Statistics bietet eine Reihe von Formaten zum Export der Arbeitsdaten in andere Formate/für andere Programme an. Abgespeichert wird die jeweilige Datensatz im Menü mit „Files > Save as". Die angebotenen Formate sind:

- Comma Separated Values – *.csv
- Dbase – *.dbf
- Excel – *.xlsx
- IBM SPSS – *.sav
- R Objects – *.RData

## I.2 Ausgabefenster und Syntaxeditor

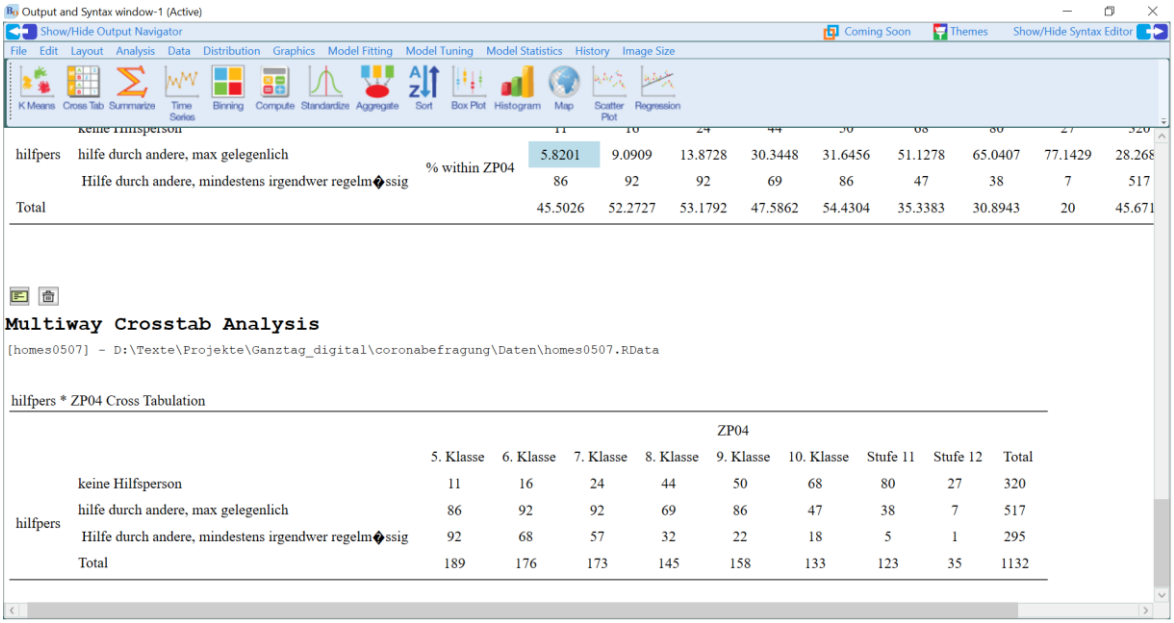



Abb. 4: Ausgabefenster

In diesem Fenster werden alle Ausgaben, Tabellen und auch erstellte Grafiken angezeigt. Ein großer Vorteil von BSS ist, dass die über die BBS-Befehle (Menü oder BSS-Syntax) erzeugten Tabellen auch als Tabellen exportierbar sind – über Copy & Paste oder direkten Export in Word, Excel oder auch pdf.

Oben *links* im Outputfenster findet sich der Pfeil zum Output Navigator. Mit diesem werden die (möglichen) Inhalte der Ausgabe angezeigt, über entsprechendes Anklicken der Fenster kann der Output erweitert werden. So können bspw. noch Hinweise oder die verwendete Syntax im Ausgabefenster zusätzlich angezeigt werden – durch Anklicken oder „Entklicken“ der Elemente in der erscheinenden Leiste.

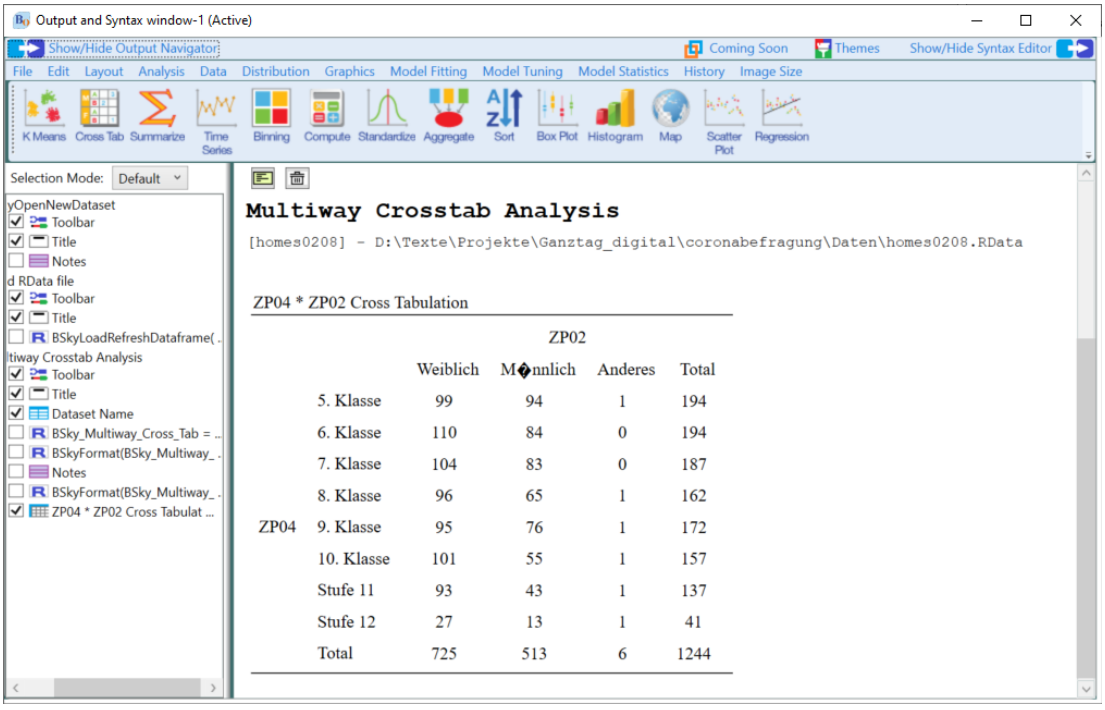



Abb. 5: Ausgabefenster mit Output-Navigator

Oben *rechts* kann mit dem Pfeil das Output-Fenster um den Syntaxeditor erweitert werden (BlueSky Statistics nennt diesen „*BlueSky R Command Editor*“), in die die Syntax/der R-Code durch das Programm eingefügt oder in dem auch selbst Codes geschrieben werden können. Über dieses Syntax-Fenster ist BlueSky Statistics „ganz normal“ mit R zu bedienen (wobei bei Anwendung klassischer Syntax dann allerdings die Formatierung der Tabellen im Outputfenster entfällt, hier lohnt es sich – wenn möglich – BSS Syntax zu verwenden).

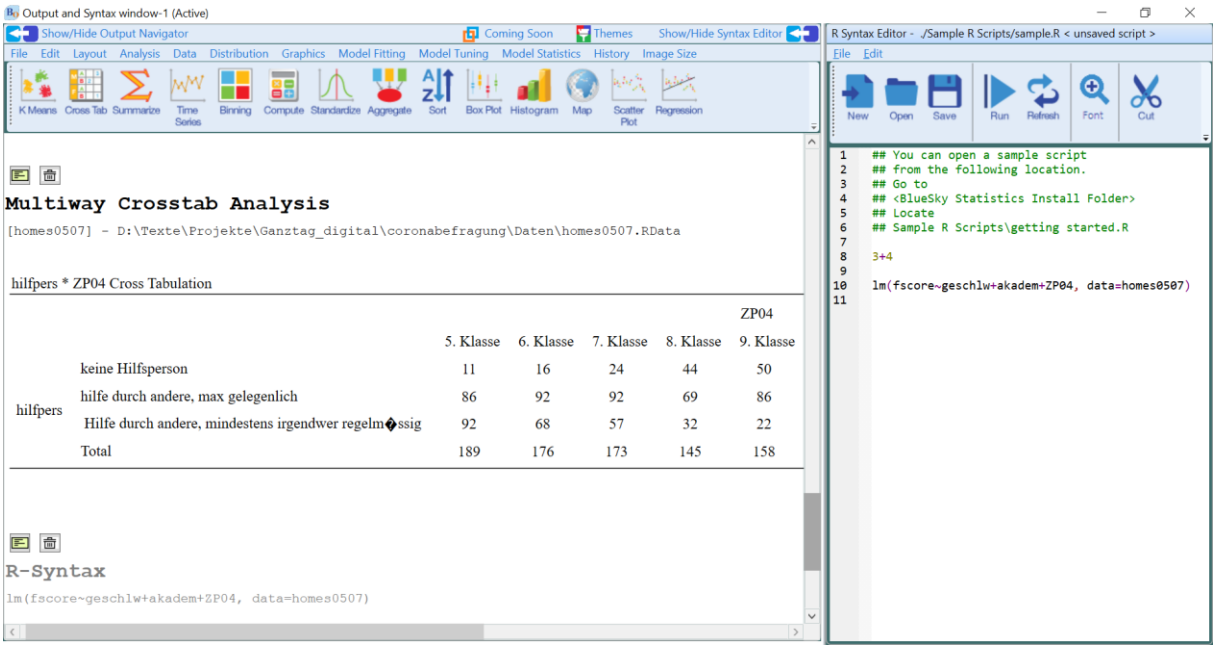



Abb. 6: Ausgabefenster mit Syntaxeditor

## I.3 Menüs & Dialogfenster

BlueSky verwendet wie bspw. SPSS im Menü Hauptkategorien wie „File“ „Data“ Analysis“ „Graphs“, „Modelling“ „Modell Fitting“ „Tools“ etc. Darunter werden dann entsprechende Dialogfelder angezeigt, in denen Variablen ausgewählt werden und in ihren verschiedenen Funktionen eingefügt werden können. Dies geschieht, indem entweder der Variablenname mit der Maus in das Funktionsfeld gezogen oder mit der Maus die Variable ausgewählt und mit dem Pfeil auf das Funktionsfeld geklickt wird. Dann kann mit "OK" der Befehl direkt ausgeführt werden. Wenn man auf die Schaltfläche "Syntax" anstelle von "OK" klickt, fügt BSS (wie SPSS) in die Syntaxfenster den entsprechenden Befehl als Code ein. Der Code wird am unteren Rand des Programmeditors hinzugefügt und hervorgehoben, mit einem Klick auf das "Ausführen"-Symbol (Pfeil) wird dieser ausgeführt.

Wenn mit dem Menüfenster und „OK“ gearbeitet wird, bleibt beim Wiederaufruf desselben Menüpunkts die vorherige Zusammenstellung im Dialogfenster (innerhalb einer Sitzung) erhalten und lässt sich so erneut ablaufen lassen oder anpassen. BSS berücksichtigt bei den Analysen das *Skalenniveau der Variablen* und „verhindert“ zunächst unpassend erscheinende Analysen.

## I.4 Syntax/Code – Steuerung von BSS über Syntax/Code

Über den Syntaxeditor kann grundsätzlich jeder R-Code verwendet werden und auch jedes R-Paket. BlueSky Statistics selbst schreibt einen „modernen“ R-Code. Für die Datenverwaltung werden tidyverse-Pakete (vgl. hierzu https://riptutorial.com/de/r/example/25722/tidyverse) verwendet. Für Grafiken verwendet es ggplot2 und für die Modellabstimmung das Caret-Paket. Der angepasste R-Code verhilft zu den funktionalen Tabellen und dem kopierbaren Output – weicht aber in einigen Fällen vom klassischen Code ab. Wenn mit R oder RStudio (ohne

die Installation von BSS) die Analysen aus BSS anhand der Syntax reproduziert werden sollen, wird eine eigene BSS Installation vorausgesetzt (Muenchen berichtet von einem "BlueSky Statistics R Package" für R oder RStudio-Nutzer*innen, welches der Autor jedoch auch auf der Homepage des Projekts nicht finden konnte).

Um dies zu veranschaulichen folgen einige Beispiele für den BlueSky R-Code: Um Mittelwerte zwischen Gruppen zu vergleichen nutzt BSS folgenden Code:

```
mySummarized <-mydata %>%
dplyr::group_by(workshop,gender) %>%

 dplyr::summarize(mean_pretest=mean(pretest,na.rm =TRUE),
mean_posttest=mean(posttest,na.rm =TRUE))
```

Ein anderes Beispiel ist der R-Code von BlueSky-Code für eine einfache lineare Regression. BlueSky liefert in der zugehörigen Syntax sogar Kommentare, die jeden Schritt erklären. An der Syntax wird deutlich, dass eigene Funktionen verwendet werden, wie der Befehl `BSkyRegression()` anstelle der in R integrierten lm()-Funktion (siehe ausführlich dazu Anhang 1). Die BSS Funktion führt sowohl den Modellierungsschritt als auch den Textformatierungsschritt aus.

```
LinearRegModel1= BSkyRegression(depVars ='posttest',
 indepVars =c('pretest'),dataset="Dataset2")
```

*#Dieser Befehl beschreibt ein lineares Regressionsmodell. Heraus kommt ein Objekt "BSkyLinearRegression". Gleichzeitig werden Modellinformationen ausgegeben wie die geschätzten Koeffizienten und für das Modell eine ANOVA Tabelle als auch die Quadratsummentabelle.*

```
if(TRUE)
{
plot(LinearRegModel1)
}
```

*#Der letzte Befehl ist für die grafische Inspektion der Voraussetzungen, hier werden- bei Nutzung des Pakets car vier Grafiken zur Inspektion der Modellvoraussetzungen erzeugt (residuals vs. fitted, Q-Q Grafik, theoretical quantiles, residuals vs. Leverage).*

BlueSkys Statistics enthält mit dem eigenen Syntaxeditor einen grundlegenden Programmeditor, der das Schreiben und Debuggen von Codes unterstützt. Der Code-Editor wird beim Start ausgeblendet, aber ein Pfeil in der oberen rechten Ecke des Ausgabefensters öffnet das Syntaxfenster (und schließt es). Ein Klick auf die Syntax-Schaltfläche in einem beliebigen Dialogfeld öffnet es ebenfalls. Der Syntaxeditor unterstützt die Syntax erstellen durch farbliche Hervorhebungen von Befehlen, Kommentaren und Variablen. Auch werden bei dem Schreiben der Syntax Vorschläge zur Vervollständigung von Befehl automatischen Ergänzen angeboten. Die Syntax kann gespeichert werden und in jeder R-Umgebung wieder ausgeführt werden – wie gesagt, für spezielle BSS-Syntax braucht es in R oder RStudio ein Zusatzpaket.

## I.5 Paketverwaltung

Ein Thema im Zusammenhang mit der Reproduzierbarkeit ist die *Paketverwaltung*. Einer der Hauptvorteile von R (und damit auch BSS) ist, dass die Funktionalität durch Add-On-Pakete immer mehr erweitert werden kann.

Über die GUI ist es einfach möglich, weitere R Pakete zu installieren oder vorhandene R Pakete zu aktualisieren, im Menü entweder über vorher downgeloadete Pakete oder – wie in R üblich über einen CRAN-Mirror – den es in praktisch jedem Land gibt. Im Menüfenster, das sich dann öffnet, muss ein angebotener ausgewählt werden. Das geht bspw. über das Menü mit „Tools > Package > Install/Update package from CRAN“ – dann öffnet sich folgendes Fenster:

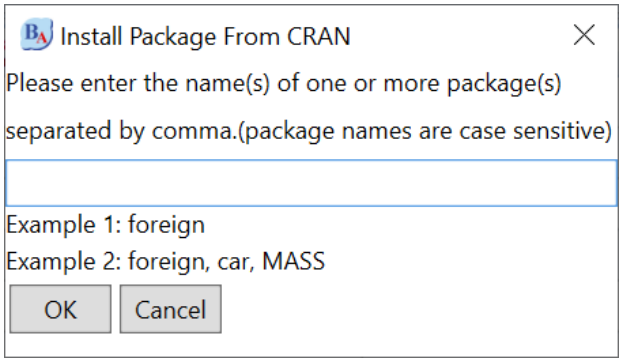



Abb. 7: Fenster für Installation/Updates von R-Paketen

Wie in RStudio kann hier das gewünschte Paket (oder die gewünschten Pakete) direkt in das Feld eingegeben werden.

*Hilfe-Funktion*: Jedes Dialogfeld verfügt über eine Schaltfläche „Hilfe“ in der oberen rechten Ecke, die ein Hilfefenster rechts neben dem Dialogfeld öffnet. Für viele Dialogfelder gibt es hier zusammenfassende Beschreibungen, Hinweise zur Verwendung des Dialogfelds zu den GUI-Einstellungen und zur Funktionsweise der begleitenden Funktion. In der unteren rechten Ecke jedes Dialogfelds befindet sich die Schaltfläche "Get R Help", die Sie zur R-Hilfeseite für die konkrete Standard-R-Funktion führt, die tatsächlich die Berechnungen durchführt. Einige Dialogfelder, die einfach eine R-Funktion aufrufen (z.B. unabhängige Samples t-Test), zeigen die integrierte Hilfedatei von R an.

## I.6 Berichte schreiben und Ausgabe exportieren

Die Analysen von BlueSky Statistics werden im Outputfenster immer mit Menütitel angezeigt, wie z.B. „Linear Regression“. Wenn man auf den Titel doppelklickt lässt sich dieser auch bearbeiten, Anmerkungen kann man im Output allerdings nicht machen, dies muss zuvor bei der Programmierung im R-Code im R-Syntax Editor geschehen.

Die Ausgabequalität von BlueSky Statistics ist höher als die von R, es können Schriftarten gewählt werden und es werden Rich-Text-Tabellen erzeugt (siehe Tab. 1). Die Formatierung der Tabellen kann man bestimmen: Um sie bspw. im Stil der American Psychological Association (APA) anzeigen zu lassen (siehe unten), wählt man im Menü die Einstellung „Tools > configuration settings“ > dann den Reiter „Output“ und darin „tables in APA-style". Wenn dieses angegeben wurde, wird für alle folgende Ausgabetabellen das APA-Format gewählt (vgl. Tab. 1).

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| **q1** | 1 | 42.31 | 42.31 | 48.83 | 0 *** |
| **q2** | 1 | 2.66 | 2.66 | 3.07 | 0.08 |
| **q3** | 1 | 0.56 | 0.56 | 0.65 | 0.42 |
| **Residuals** | 95 | 82.31 | 0.87 | NA | NA |

*Note.* Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tab. 1: Ausgabe in BlueSky

Mit der rechten Maustaste kann auf eine beliebige Tabelle geklickt und diese in Word, in Excel oder in eine pdf-Datei exportiert werden, die Formatierung wird dabei beibehalten. Die gesamte Ausgabe wird in einer einzigen Datei gespeichert, die dann im verknüpften Programm

geöffnet wird und bearbeitet werden kann. Die Ausgabe kann auch mit der rechten Maustaste mit "copy to clipboard“ in die Zwischenablage kopiert werden – in Excel kann dies als richtige Tabelle eingefügt werden (in Word funktioniert dies leider nicht).

Ein Vorteil der BlueSky Statistics-Ausgabetabellen ist ihre Interaktivität. Die Ausgaben lassen sich nach p-Wert, Parametergröße oder einer beliebigen Spalte sortieren.

## II. Variablen- und Datenmanagement

BlueSky Statistics bietet eine Reihe von Optionen zum Datenmanagement auch im Menü an. Sinnvoll ist es, die verschiedenen Optionen auszuprobieren und individuell den favorisierten Zugang zu finden. Über das Einfügen der Befehle in das Syntaxfenster können diese gespeichert und immer wieder reproduziert und angepasst werden. (Anmerkung: Dem Autor erscheint die Arbeit mit entsprechenden Syntaxen oft einfacher, daher sind im Folgenden unter den Fenstern auch die entsprechenden BBS-Syntaxbefehle dargestellt.)

Wichtige Befehle für das Datenmanagement bei Auswertungen sind:

- Aufteilen des Datensatzes

Über „Data“ > „Split Dataset“ wird die Datei *in zwei oder mehr Gruppen aufgeteilt* (also beispielsweise nach Geschlecht), wenn verschiedene Gruppen (Stufen einer Variablen) getrennt analysieren werden sollen. Alle folgenden Analysen werden für jede Ebene der gewählten Faktoren durchgeführt.

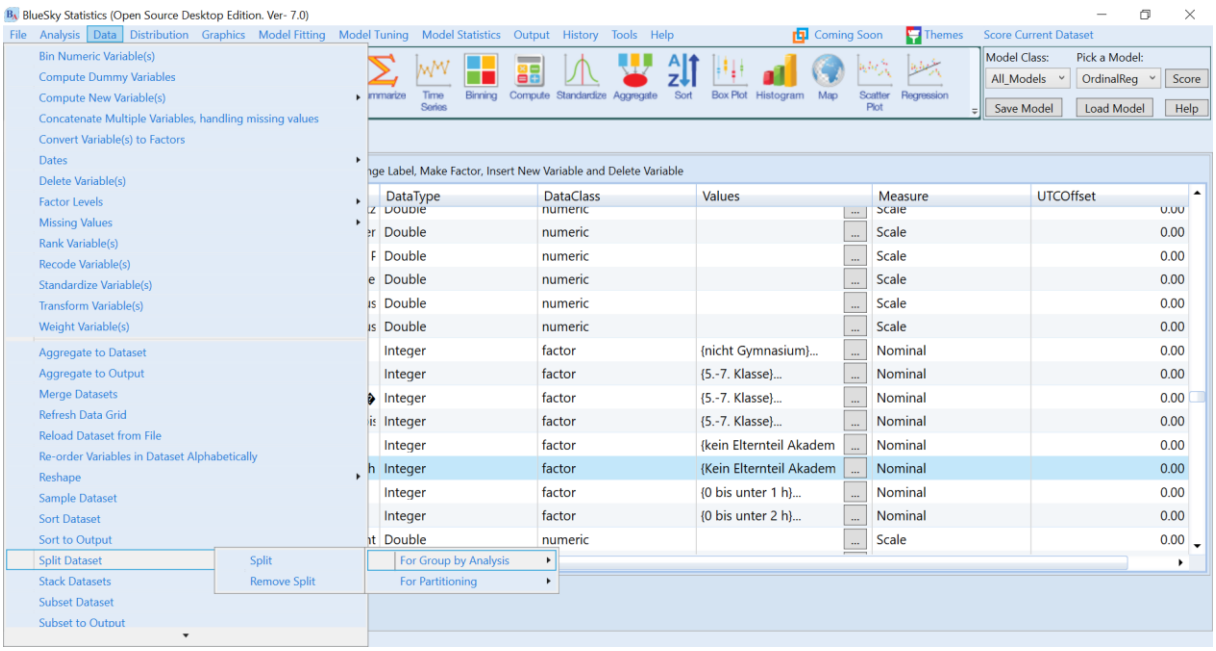



Abb. 8: Menüführung für das Aufteilen des Datensatzes

Die Syntax dafür ist:

```
## [Set Split]
BSkySetDataFrameSplit(c('geschlw'),'Homes.0208gdt.sav')
```

Diese Funktion wird über "Split> For Analysis> Remove Split" deaktiviert.

- Neue Variablen berechnen

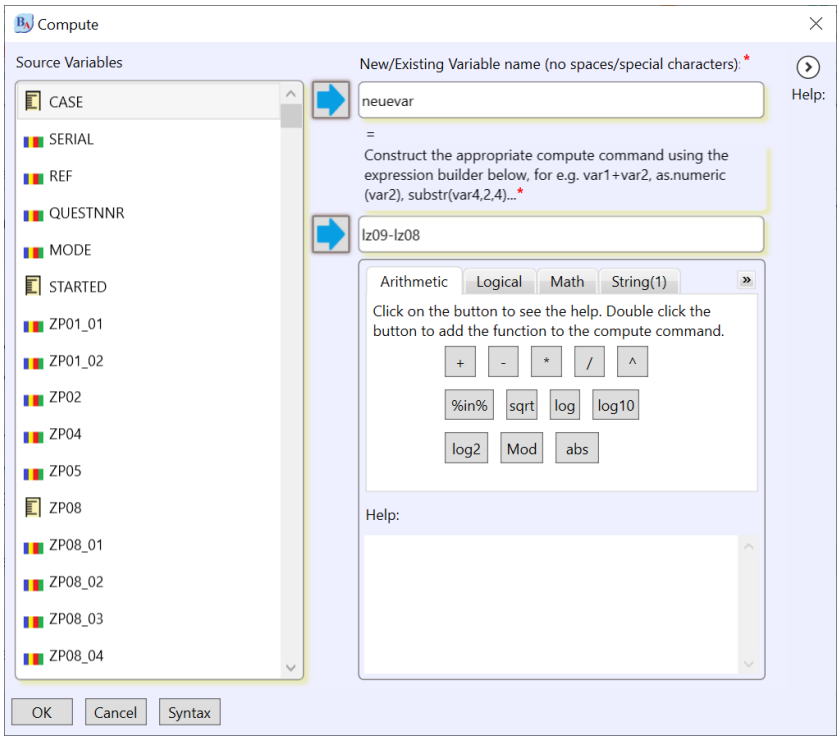



Abb. 9: Unterfenster zur Berechnung neuer Variablen

```
## [Compute]
require(stringr);
require(dplyr);

#Computes the new variable values
homes0208 <-homes0208 %>%
       mutate( neuevar =lz09-lz08)
#Refreshed the dataset in the data grid
BSkyLoadRefreshDataframe(homes0208)
```

Wahlweise kann über „Data > Compute New Variable(s) > Apply a function across all rows" auch eine Variable durch Berechnung mit vorgegebener Formel durchgeführt werden. Das folgende Beispiel erstellt eine Mittelwertsvariable aus drei Variablen:

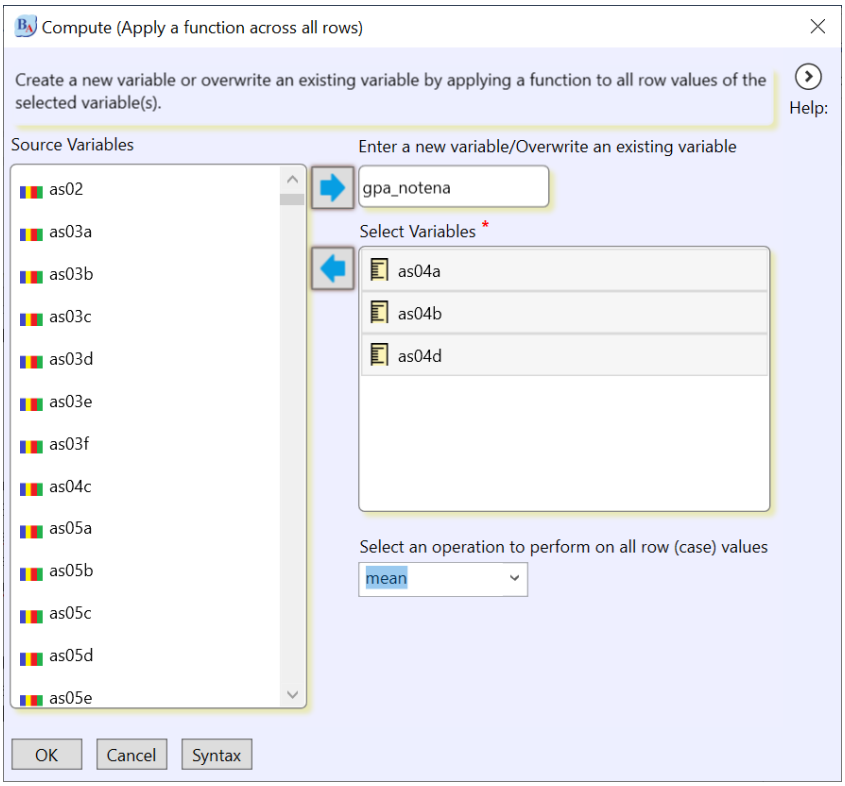



Abb. 10: Unterfenster zur Berechnung neuer Variablen über Funktionen

```
## [Compute (Apply a function across all rows)]
require(dplyr);

#Apply function to all rows
asbscsneudownR$gpanoten <-asbscsneudownR %>%
       select(as04d,as04b,as04a) %>%
       apply(1,mean,na.rm = TRUE)
#Refresh the dataset in the grid
BSkyLoadRefreshDataframe(asbscsneudownR)
```

- Rekodieren einer Variablen

Dies wird im Menüfenster aufgerufen über „Data“ > „Recode Variables“ (= umcodieren**).** Häufig ist es sinnvoll/notwendig, Variablen noch einmal umzucodieren und zum Beispiel Stufen zusammenzufassen oder bei Items für Skalen die Ausprägungen von bspw. 1-5 umzudrehen/zu „invertieren“.

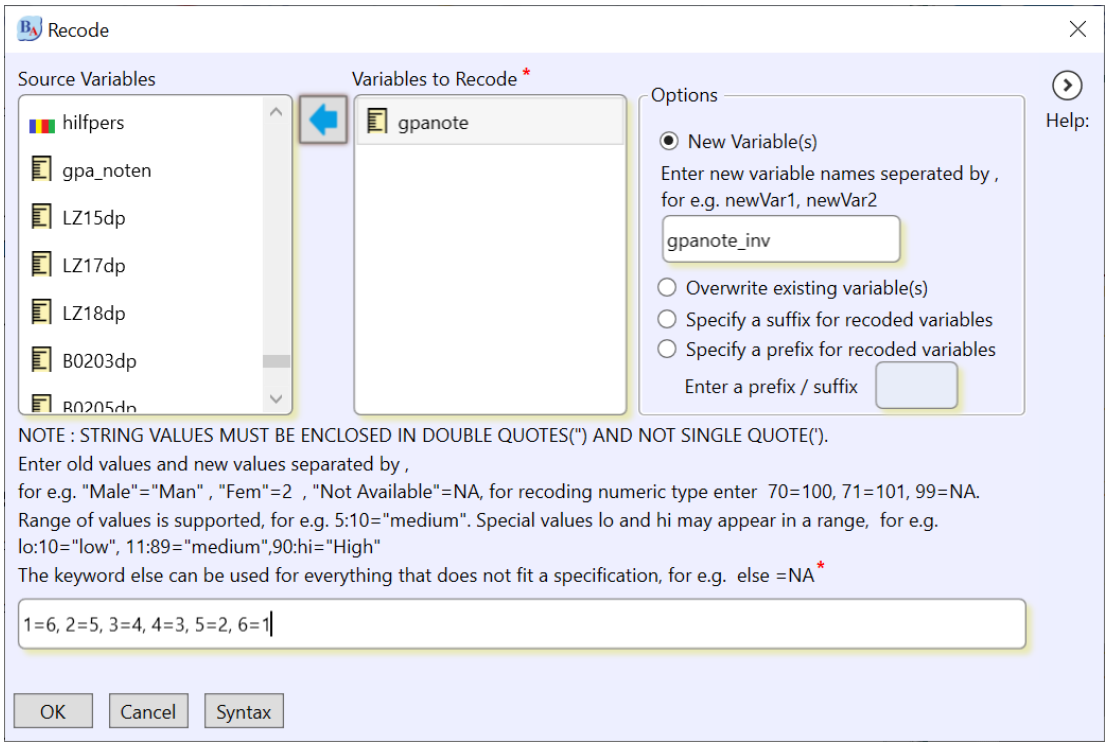



Abb. 11: Unterfenster für die Rekodierung von Variablen

```
Syntax
## [Recode]
require(car);

#Perform the recode
BSkyRecode(colNames=c('gpanote'),newColNames=c(c('gpanote_inv')),Old-
NewVals='1=6,2=5,3=4,4=3,5=2,6=1',prefixOrSuffix      =c(''),NewCol=TRUE,da-
taSetNameOrIndex='homes0208')
#Refresh the dataset in the data grid
BSkyLoadRefreshDataframe(homes0208)
```

- Auswahl einer Teilgruppe aus dem Datensatz

Wenn nur ein Teil des Datensatzes in Analysen verwendet werden soll, können über das Menü mit „Data“ > „Subset Dataset“ bestimmte Fälle ausgewählt werden (bspw. nur Personen einer bestimmten Altersgruppe, Personen aus ausgewählten Regionen oder Personen mit einem bestimmten Geschlecht. Es öffnet sich dann folgendes Fenster:

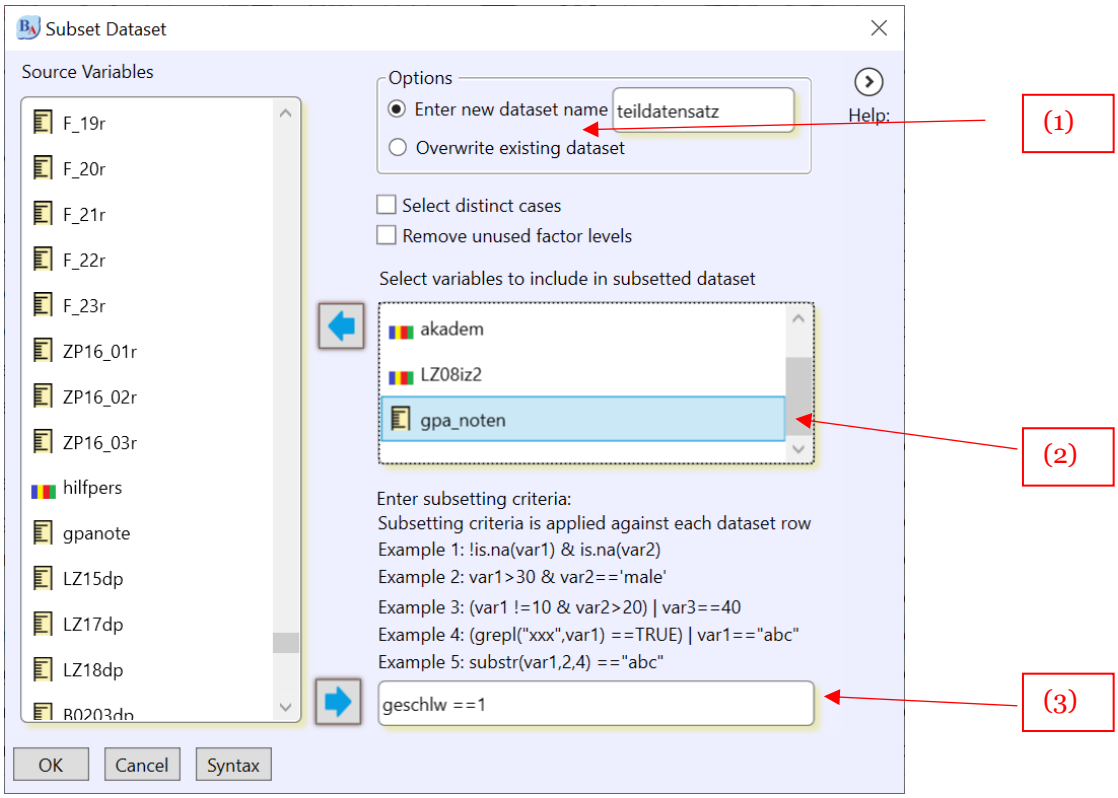



Abb. 12: Untermenü für die Bildung eines Teildatensatzes

Dabei können sowohl die Variablen, die im neuen Datensatz verwendet werden sollen (2), als auch die Bedingungen, die die Fälle erfüllen sollen (3), angegeben werden. Auch muss entschieden werden, ob der bisherige Datensatz überschrieben oder ein neuer Datensatz angelegt werden soll (im letzteren Fall Dabei technisch in BSS ein neuer Teildatensatz erstellt, der auch im Datenfenster als neuer Reiter angezeigt wird).

Die Syntax dafür:

```
## [Subset Dataset]
require(dplyr);

#Creates the subsetted dataset
teildatensatz<-homes0208 %>%
 dplyr::filter(geschlw ==1) %>%
 dplyr::select(gym,akadem,LZ08iz2,gpa_noten)
#Refreshes the subsetted dataset in the data grid
BSkyLoadRefreshDataframe(teildatensatz)
```

- *Erstellung Dummy Variablen*

Für Erstellung von Dummyvariablen(0/1-Variablen) aus gestuften Variablen gibt es eine bei BSS eine eigene Routine „Data“ > „Compute Dummy Variables“.

```
library(fastDummies);
#Dummy coding variables
Homes.0208gdt.sav <- dummy_cols(.data =Homes.0208gdt.sav,select_columns =
c('schulstuf3'),remove_selected_columns = FALSE, ignore_na = TRUE)
BSkyLoadRefreshDataframe(dframe=Homes.0208gdt.sav,load.dataframe=TRUE)
```

- *Nummerische Variable in Stufen aufteilen*

Hierfür hat BSS die sehr nützliche Funktion „Date > Bin numeric variable“, mit der über die Menüführung eine metrische Variable in eine gestufte rekodiert werden können, im entsprechenden Fenster müssen nur die „Grenzwerte“ für die Stufen eingegeben werden.

- *Entfernen fehlender Werte*

Fehlenden Werte sind bei einigen Analysen in R ein Problem. Diese können ausgeschlossen werden, wenn mit der Menüführung unter „Data“ > Missing Values > Remove NA“ eine neuer Teildatensatz angelegt wird, der für die für die aktuellen Analyse relevanten Variablen nur gültige Fälle hat und dieser dann zur Grafikerstellung genutzt wird.

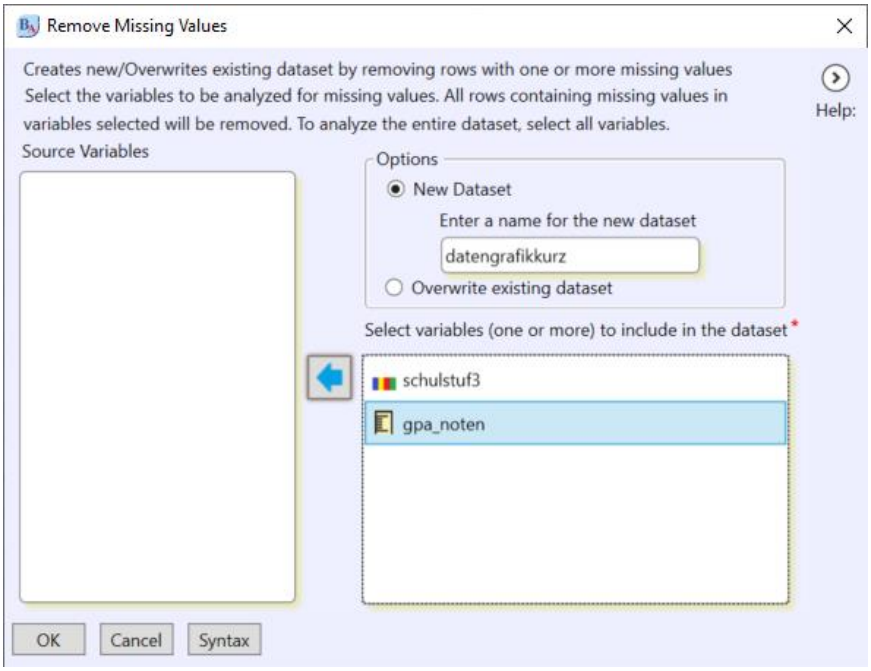



Abb. 13: Menüfenster zur Entfernung von „missings“ in einem neuen (Teil)Datensatz

## III. Aufrufen von Datenauswertungen und -analysen

Viele Datendarstellungs- und Auswertungsroutinen sind – analog zu SPSS mit der Maus „klickbar“. Allerdings ist es wie generell in R bedeutsam, den Variablen vorab das richtige Skalenniveau zuzuweisen – da BSS *einerseits* viele beschreibende Auswertungsroutinen automatisch adäquat wählt und *andererseits* verhindert, dass Auswertungsroutinen mit Variablen mit nicht passendem Skalenniveau durchgeführt werden.

Für viele Auswertungsroutinen gibt es über das Menü mehrere Wege, zu einer entsprechender Auswertungstabelle zu kommen (s.u.), die Wahl hängt eher von den gewünschten Informationen und der Darstellungsart ab (und ist auch Geschmackssache). Viele der Tabellen lassen sich bei der Erstellung auch modifizieren (zusätzliche Angaben, Anzeige oder Weglassen fehlender Werte).

### III.1 Einfache beschreibende Statistiken

1. Einfache Häufigkeitstabellen (für ordinale oder kategoriale Variablen)

- „Summary Analysis > Frequency Tables“ und dann das Klicken der gewünschten Variablen in das entsprechende Fenster erzeugt eine zu SPSS vergleichbare einfache univariate Häufigkeitstabelle (ohne Anpassungsmöglichkeiten). Alternativ kann über

- „Analysis > Tables > Basic“ und dem Einbeziehen nur der gewünschten Variablen erzeugt ebenfalls eine univariate Tabelle, die mit den „Options“ vielfältig angepasst werden kann

Über „Analysis > Tables > Basic“ können auch Kreuztabellen, Mittelwertsausgaben sowie Mittelwertsvergleiche erzeugt werden, je nachdem welche und wie viele Variablen angegeben werden, was diesen Menüweg zu einer sehr vielseitigen Option macht.

2. Mittelwert und Standardabweichung einer metrisch skalierten Variablen

- Über „Analysis > Summary Analysis > Numerical Statistical Analysis” und dann das Auswählen der gewünschten Variablen oder alternativ über
- „Analysis > Tables > Basic“ und dann die Auswahl der gewünschte Variable sind die Darstellung von Mittelwerten und Standardabweichungen möglich

3. Kreuztabelle mit zwei Variablen

Als eine von vielen Möglichkeiten können über

- “Analysis > Contingency Tables > Contingency Tables > Crosstab, Multi-Way”,

verbunden mit der Auswahl von 2 oder mehr Variablen Kreuztabellen in Analogie zu SPSS-Ausgaben erstellt werden (vgl. Abb. 12).

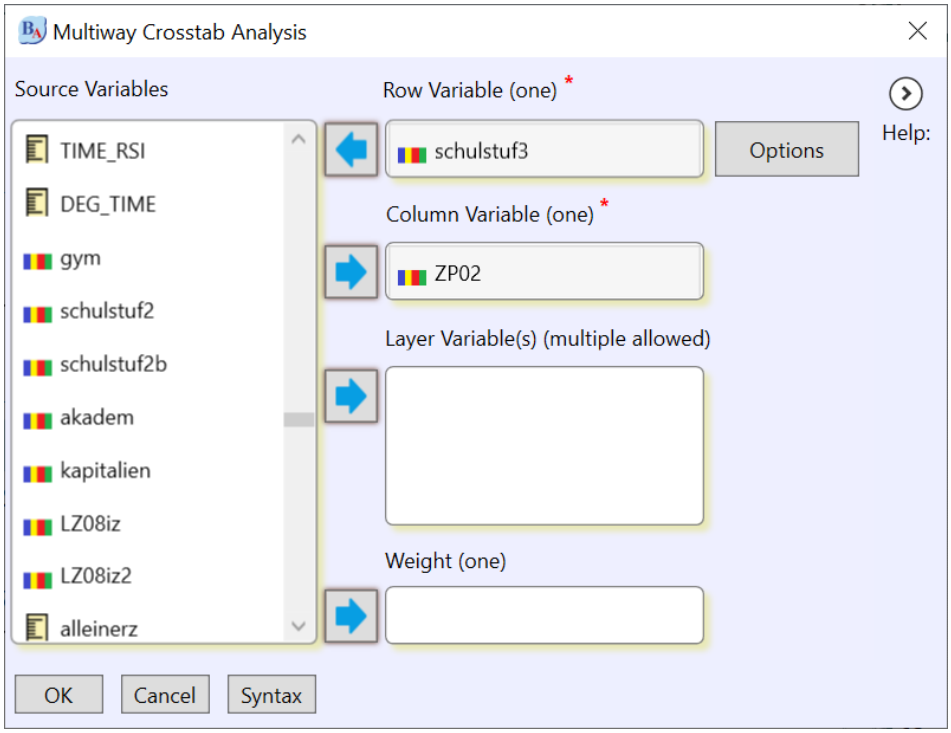



Abb. 14: Untermenü für die Erstellung von Kreuztabellen

Über das Untermenü „Options“ können bspw. noch Prozentwerte und auch der Chi-Quadrat Test angefordert werden.

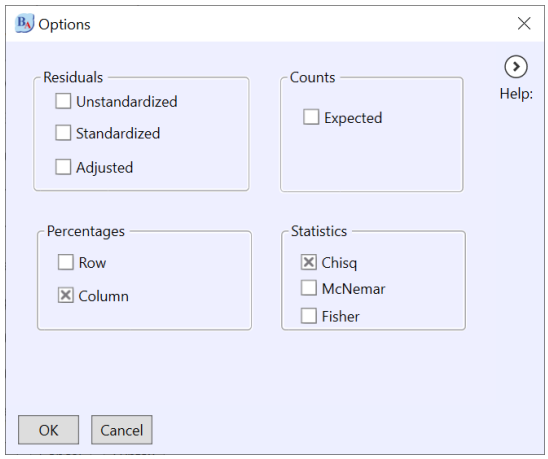



Abb. 15: Unterfenster für die Anforderung von Auswertungsstatistiken bei Kreuztabellen

Nach Befehlseingabe erscheint im Outputfenster folgende Tabelle:

*LZ09 * ZP02 Cross Tabulation*

| | | | ZP02 | | | |
|---|---|---|---|---|---|---|
| | | | Weiblich | Männlich | Anderes | Total |
| LZ09 | Weniger Zeit als sonst | % within ZP02 | 189 | 197 | 1 | 387 |
| LZ09 | Ungefähr gleich viel Zeit | % within ZP02 | 28.421 | 43.202 | 16.667 | 34.339 |
| LZ09 | Mehr Zeit | % within ZP02 | 183 | 100 | 2 | 285 |
| Total | | % within ZP02 | 27.519 | 21.93 | 33.333 | 25.288 |

Statistical Test Results

| | Value | df | Asyp. Sig | Odds ratio | 95%Confidence interval | |
|---|---|---|---|---|---|---|
| Pearson Chi Square | 27.054 | 4 | 0.000 | - | - | - |

Tab. 3: Ausgabe der Funktion „Crosstabs > Multi way“

Die Signifikanzangabe des Chi2 tests findet sich unter „Asyp. Sig”.

- Alternativ „Analysis > Tables > Basic“ und dann die gewünschten Variablen auswählen. Wenn Prozentangaben gewünscht werden, kann unter „Options > Categorial statistics“ Absolutwerte mit „Frequency“ und/oder Prozentwerte mit „Frequency (%)“ angefordert werden. Auch kann unter Options angegeben werden, ob Missings und/oder Signifikanztest mit ausgegeben werden sollen.

*Variable Summaries*

| ZP02 | Weiblich (N=726) | Männlich (N=513) | Anderes (N=6) | p value |
|---|---|---|---|---|
| LZ09 | | | | < 0.001 |
| - N | 665 | 456 | 6 | |
| - Weniger Zeit als sonst | 189 | 197 | 1 | |
| - Ungefähr gleich viel Zeit | 183 | 100 | 2 | |
| - Mehr Zeit | 293 | 159 | 3 | |
| - Weniger Zeit als sonst | 189 (28.4%) | 197 (43.2%) | 1 (16.7%) | |
| - Ungefähr gleich viel Zeit | 183 (27.5%) | 100 (21.9%) | 2 (33.3%) | |
| - Mehr Zeit | 293 (44.1%) | 159 (34.9%) | 3 (50.0%) | |

Tab. 4: Ausgabe der Funktion "Table > Basic"

Die/der User*in muss entscheiden, welche Option favorisiert wird, natürlich können auch noch weitere Darstellungsoptionen von BSS ausprobiert werden.

4. Mittelwerte mehrerer Gruppen/verschiedener Ausprägungen einer Variablen (z.B. Schuhgröße nach Geschlecht)

- Nach „Analysis > Tables > Basic" muss festgelegt werden legen: „Variables to summarize" und bei Bedarf „Groups to compare" (als Gruppenvariable). Unter *Options* können dann die Werte zur Ausgabe ausgewählt als auch Tests wie ANOVA bzw. Kruskal-Wallis-Test angefordert werden.
- *Alternativ* stehen unter „Analysis > Means" für weitergehende Mittelwertsvergleiche eine Reihe von Varianzanalysen und T-Tests zur Verfügung, z.B. auch die SPSS-analoge Option „Analysis > Means > ANOVA, one Way and two Way" (siehe Abschnitt V).

Sinnvoll erscheint es, sich für eine Auswertung mit verschiedenen Wegen vertraut zu machen und individuell das jeweils favorisiertes Vorgehen zu identifizieren.

Grundsätzlich wird über die Menüleiste mit „Analysis > Tables > Basic" ein „one stop shop" als Ausgangspunkt für beschreibende Auswertungen angeboten. Über diesen erschließt sich ein sehr großer Anwendungsbereich, der z.B. zu einfachen Häufigkeiten, Mittelwerten, Kreuztabellen und Mittelwertsvergleichen führt, für den in dem Unterfenster „Options" noch eine sehr große Vielfalt an Anpassungsmöglichkeiten in der Darstellung existieren und es können sogar statistische Tests hinzugefügt werden.

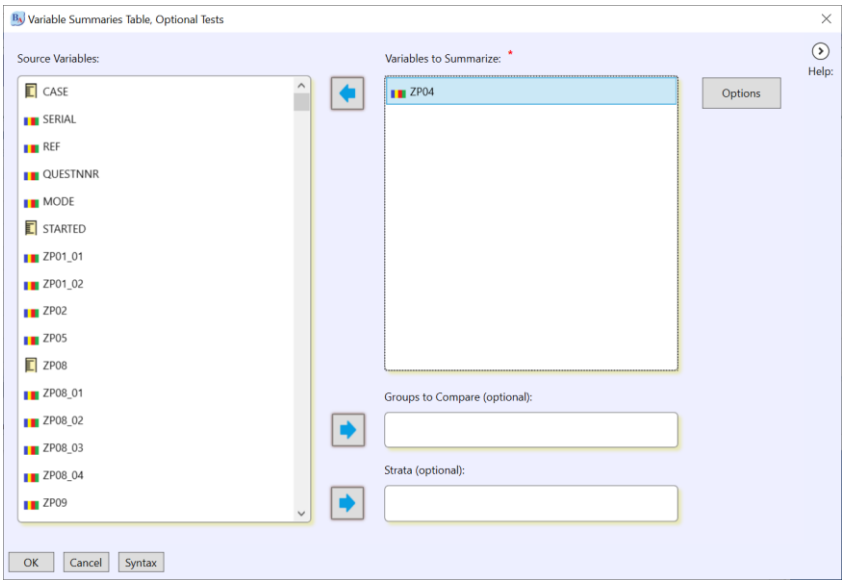



Abb. 16: Menüfenster „*Analysis* > *Tables* > *Basic*“

Über das Feld „Groups to Compare“ können sowohl kategoriale als auch metrische Variablen nach Gruppen verglichen werden, mit Strata lassen sich weitere Ebenen einbeziehen.

Über das Unterfenster „Options“ in dieser Ansicht besteht dann – je nach Skalenniveau (Numerical Statistics, Categorical Statistics, Date Statistics) – die Möglichkeit die Darstellung der einzelnen Ausgabetabellen anzupassen – oder auch Signifikanztests anzufordern.

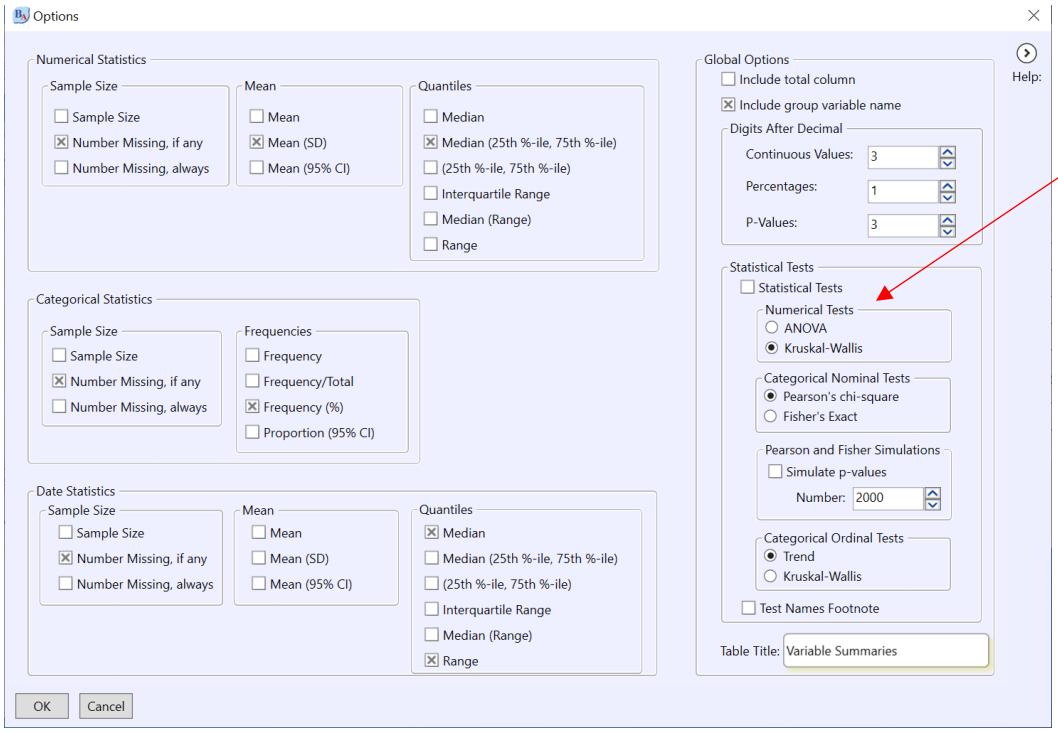



Abb. 17: Unterfenster „Options“ im Menüfenster „Analysis > Tables > Basic“

Ein solcher Mittelwertsvergleich erzeugt bspw. folgende Ausgabe:

Variable Summaries

| schulstuf3 | 5.-7. Klasse (N=584) | 8.-10. Klasse (N=494) | Stufe 11 & 12 (N=179) |
|---|---|---|---|
| gpa_noten | | | |
| - N | 558 | 479 | 169 |
| - Mean | 2.51 | 2.67 | 2.64 |
| - Mean (SD) | 2.51 (0.78) | 2.67 (0.88) | 2.64 (0.83) |
| - Median (Q1, Q3) | 2.50 (2.00, 3.00) | 2.67 (2.00, 3.33) | 2.67 (2.00, 3.33) |

Tab. 5: Ausgabe der Mittelwertsanalyse

Generell sind in den Auswertungsroutinen von BlueSky Statistics keine Optionen für die Berechnung von Effektstärkemaßen für Kontingenztabellen oder Mittelwertsvergleiche implementiert, die aber, wenn gewünscht, mit zusätzlichen R Paketen über die R Syntax angefordert werden können. Anhang 2 zeigt hierfür exemplarische Möglichkeiten.

## III.2 Aufruf weiterer statistischer Auswertungsverfahren

Über die Menüführung können eine Vielzahl statistischer Auswertungsverfahren aufgerufen werden – vor allem über die Hauptpunkte „Analysis“ und „Model Fitting“. Im Folgenden sollen nur de Zugangswege kurz veranschaulicht werden, die Durchführung ausgewählter Methoden wird ausführlich in Abschnitt V beschrieben.

a) Varianzanalysen/T-Test (siehe auch ausführlich Teil V)

Über „Analysis > Means“ gelangt man/frau zu einem ausführlichen Menüfenster für u.a. verschiedene T-Tests und Varianzanalysen. Abschnitt VI enthält hierfür Durchführungsbeispiele.

b) Regressionsmodelle (siehe ausführlich Teil V)

Unter „Modell Fitting“ steht ein Bereich mit vielen Optionen und unterschiedlichen Modellen zur Verfügung. Zum Beispiel ist der Weg für eine *Lineare Regression* (Einflussfaktoren auf eine metrisch skalierte Variable testen z.B. Geschlecht, Lesehäufigkeit und Elternbildung auf Lesekompetenz)

- „Model Fitting > Linear Regression“.

Dabei ist das Fenster zur Modellspezifikation bei allen Regressionen ähnlich (vgl. Abb. 10):

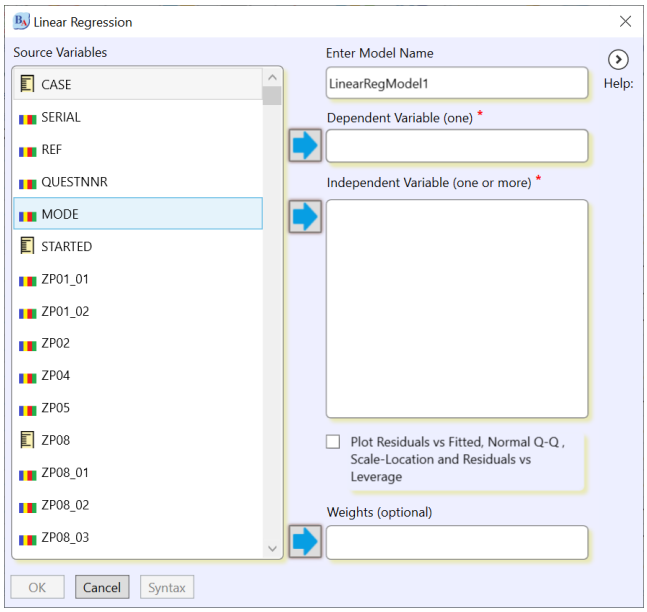



Abb. 18: Menüfenster zur Spezifizierung einer linearen Regression

Es lassen sich über das kleine Klickfenster auch Grafiken zu Inspektion der Voraussetzung der linearen Regression anfordern.

Analog werden weitere Regressionsmodelle mit

- „Model Fitting > Logistic Regression“
- „Model Fitting > Ordinal Regression“.
- „Model Fitting > Mixed Models, basic“ (hier können bislang nur lineare Modelle geschätzt werden)

angefordert.

Der zugehörige R-Code zu allen Analysen kann wie immer angezeigt und vor der Durchführung der Analyse ins Syntaxfenster eingefügt („Einfügen“) und so auch als eigene Syntax gespeichert werden.

Bedeutsam ist, dass BSS die R-Logik der „Modellobjekte“ übernimmt – das heißt, jedes geschätzte Modell wird im Arbeitsspeicher als Objekt mit einem (veränderbaren) Namen wie „LinearRegModel1“ gespeichert, kann entsprechend wieder aufgerufen werden, und über die Funktionen „Model Tuning“ „Model Statistics“ können Modellanpassungen vorgenommen oder weitergehende Informationen über das jeweilige Modell (wie zum Beispiel Gütekriterien, Pseudo-R etc. ) abgerufen werden. Oben rechts im Hauptmenüfenster können die in der Session erzeugten Modell aufgerufen werden (Drop-Down Fenster) und wieder analysiert/deren Informationen abgerufen werden. (Alle dann gewählten Auswahloptionen aus dem Menüfenster Model Statistics wie AIC (Modell Information) oder weitere Modellanalysen (wie „stepwise variable selection“) werden jeweils für das angezeigte Modell durchgeführt.

c) Faktoranalysen

- Unter „Analysis > Factor Analysis“ stehen Faktorenanalysen und Hauptkomponentenanalyse (als *explorative* Faktorenanalysen) zur Verfügung.
- *Konfirmatorische* Faktorenanalysen sind bislang noch nicht in BSS nicht integriert. Hier kann bspw. über die Installation des R-Paketes lavaan und mit der entsprechende R Syntax die Funktionalität erweitert werden – dies setzt dann aber Syntaxarbeit voraus (siehe Abschnitt VI).

d) Clusteranalysen/Klassenanalysen

- Clusteranalysen sind unter „Analysis > Cluster Analysis“ aufrufbar. Zur Verfügung stehen aus dem Basis-Paket die hierarchische Clusteranalyse oder die Clusteranalyse mit K-means-Verfahren. Die Funktionalität ist hier insgesamt beschränkt, bspw. das R-Paket *cluster* bietet deutlich mehr Option für Clusteranalysen.
- Latente Klassenanalysen (LCA) oder auch Latente Profilanalysen (LPA) lassen sich mit BSS nicht durchführen – hier als Ausweichmöglichkeiten auf das R Paket poLCA für die LCA oder das Paket *mclust* (vereinfacht noch in Verfindung mit dem Paket *tidyverse*) für die LPA verwiesen, die aber wiederum ausschließlich über die R-Syntax verwendet werden können (siehe Abschnitt VI).

## Teil 2: Erstellung von Grafiken und Durchführung von statistischen Analysen

## IV. Grafiken

Für Grafiken greift BlueSky Statistics vor allem auf das ggplot2-Paket zurück, hier wird der Original Code verwendet (BlueSky bietet aber auch Grafikfunktionen aus dem R-Grundmodul bei dem Menüunterpunkt "Legacy“ an). Über das Menüfenster „Graphics“ öffnet sich eine Liste von möglichen Grafiken.

**Hinweis:** Fehlenden Werte („NA“) werden bei der Erstellung von Grafiken immer als eine eigene Kategorie ausgegeben. Diese kann verhindert werden, wenn zuvor mit der Menüführung unter „Data“ > Missing Values > Remove NA“ eine neuer Teildatensatz angelegt wurde, der für die betreffenden Variablen nur gültige Fälle hat und dieser dann zur Grafikerstellung genutzt wird (s. Abschnitt II).

## IV.1 Erstellen von Balkendiagrammen

Über „Graphics“> “bar chart, means” können Balkendiagramme von Mittelwerten erstellt werden. Es öffnet sich das Unterfenster

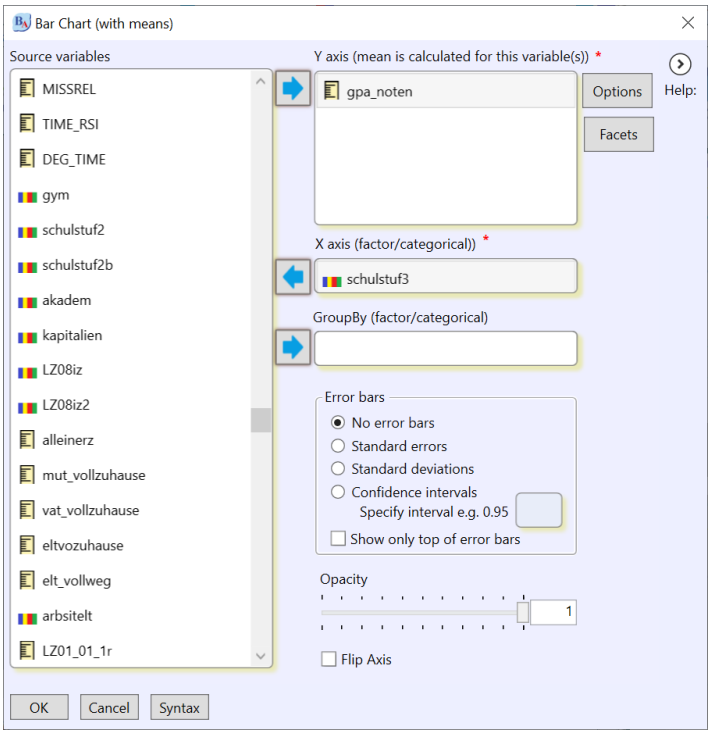



Abb. 19: Menüfenster zur Erstellung von Balkendiagrammen

Mit der Option „Flip Axis“ kann das Säulen- zu einem Balkendiagramm gedreht werden.

Unter dem Untermenü „Options“ kann stets die Beschriftung angepasst und wie im folgenden Beispiel auch die Farbe der Balken angepasst werden


Abb. 20: Untermenü „Options“ bei Balkendiagrammen

Im Ergebnis zeigt sich folgende Grafik:

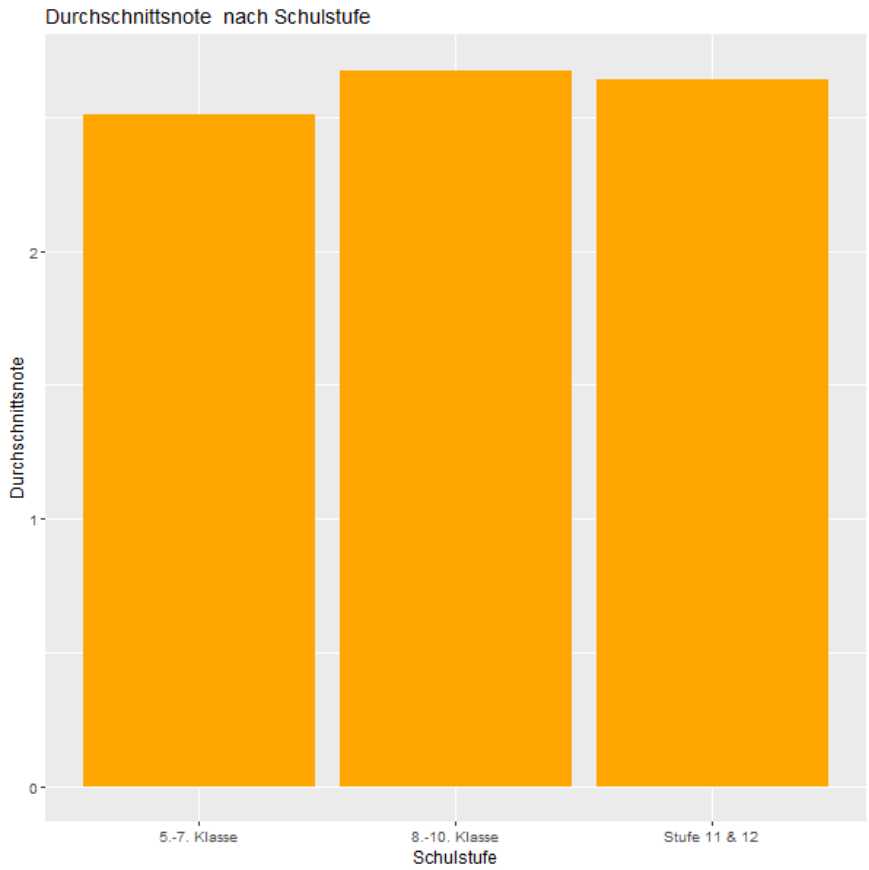



Abb. 21: Einfaches Balkendiagramm

Die Syntax hierfür lautet:

```
## [Bar Chart with means)]
require(ggplot2);
require(ggthemes);
require(Rmisc);

temp <-Rmisc::summarySE(datensatzkurz,measurevar = "gpa_noten",groupvars =
c("schulstuf3"),na.rm = TRUE,.drop = TRUE)
pd <- position_dodge(0.9)
ggplot(data=temp,aes(x = schulstuf3,y = gpa_noten)) +
       geom_bar( position="dodge",alpha=1,fill ="orange",stat="identity") +
       labs(x ="schulstuf3",y ="gpa_noten",title= "Bar Chart (with means)
for Y axis variable gpa_noten,X axis variable schulstuf3") +
       xlab("Schulstufe") +
       ylab("Durchschnittsnote") +
       ggtitle("Durchschnittsnote nach Schulstufe") +
theme_grey() + theme(text=element_text(fam-
ily="sans",face="plain",color="#000000",size=12,hjust=0.5,vjust=0.5))
```

Über die Menüführung gibt es ein paar weitere Veränderungsmöglichkeiten im ersten Unterfenster, für größere Veränderungen muss die Syntax ins Syntaxfenster eingefügt werden – dann können bspw. noch die Schriftgröße, einzelne Farben etc. verändert werden.

Beispiel für Stapelbalken

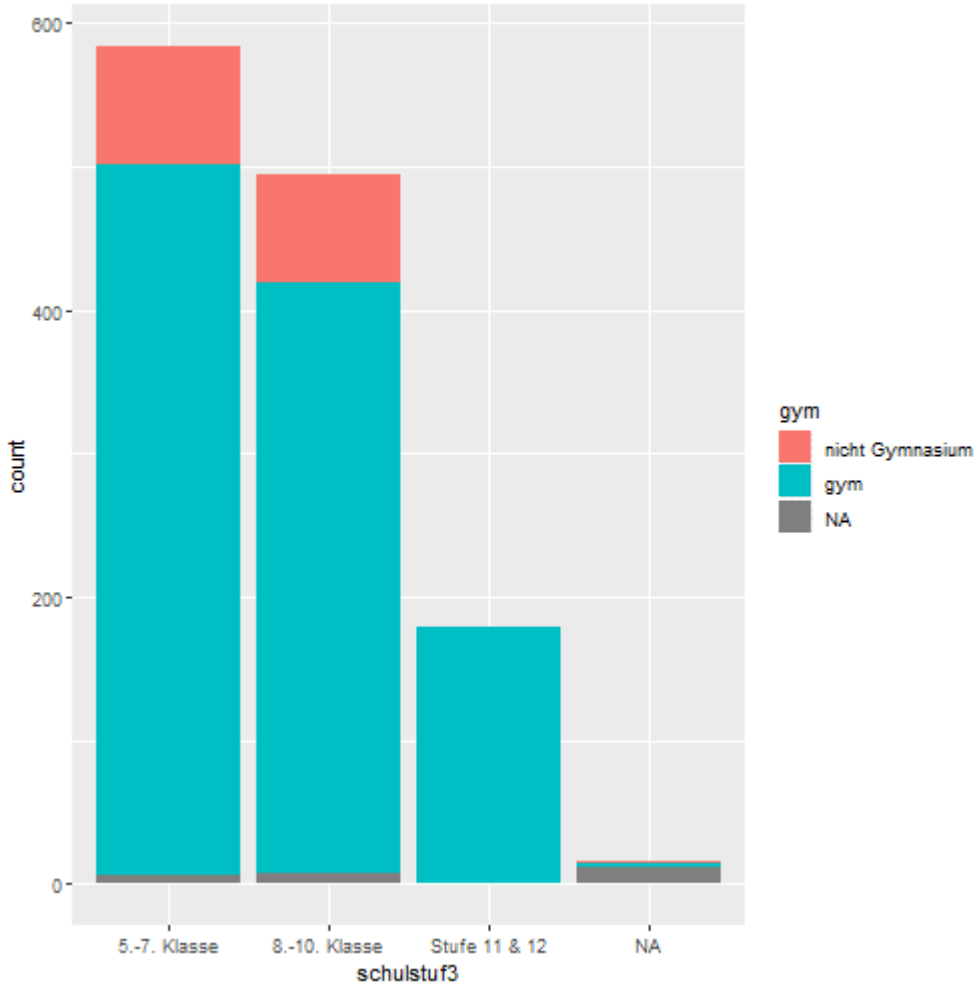



Abb. 22: Gestapeltes Balkendiagramm

```
require(ggplot2)
ggplot(data = homes3108, aes(x = schulstuf3, fill = gym)) + geom_bar ()
```

## IV.2 Erstellen von Diagrammen mit Boxplots

Über die Menüführung „Graphics“> boxplots” kommt man analog zur Erstellung von Boxplots, mit entsprechendem Unterfenster für die Beschriftung.

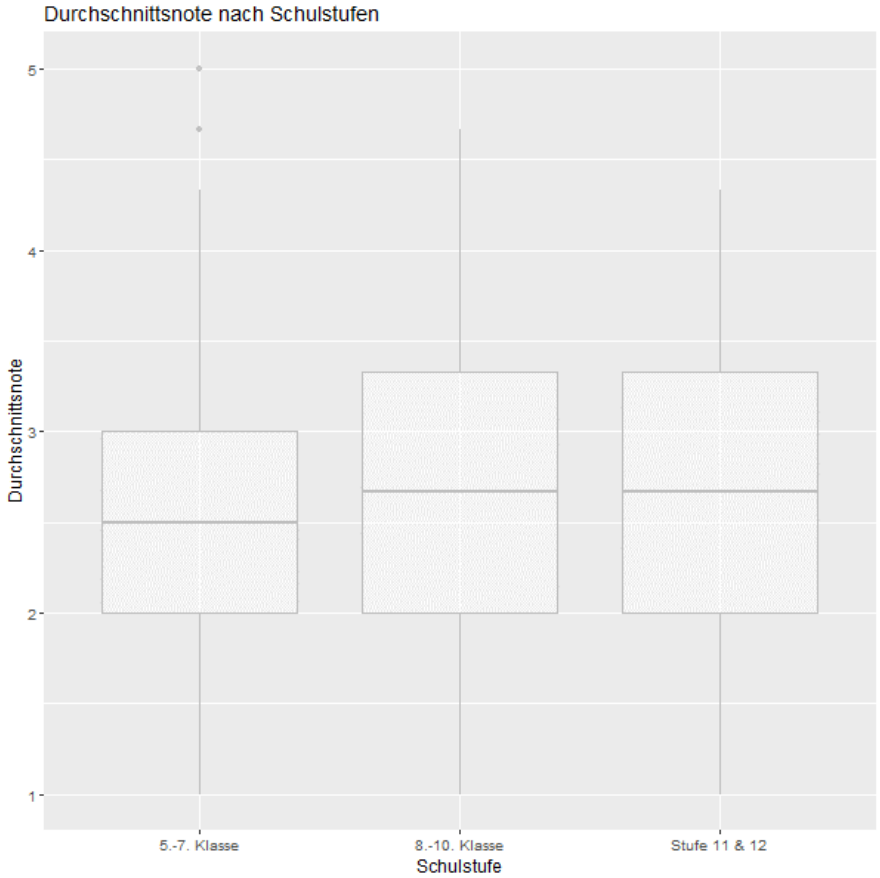



Abb. 23: Diagramm mit Boxplots

Die entsprechende Syntax lautet

```
## [BoxPlot]
require(ggplot2);
require(ggthemes);

ggplot(data=datensatzkurz,aes(x =schulstuf3,y = gpa_noten)) +
 geom_boxplot(col ="gray",alpha =0.5) +
 labs(x ="schulstuf3",y ="gpa_noten",title= "Boxplot for variable
gpa_noten,group by schulstuf3") +
 xlab("Schulstufe") +
 ylab("Durchschnittsnote") +
 ggtitle("Durchschnittsnote nach Schulstufen") +
theme_grey() + theme(text=element_text(fam-
ily="sans",face="plain",color="#000000",size=12,hjust=0.5,vjust=0.5))
```

## IV.3 Erstellen von Histogrammen

Unter „Graphics“> „Histogramm“ kann analog ein einfaches Histogramm erstellt werden.

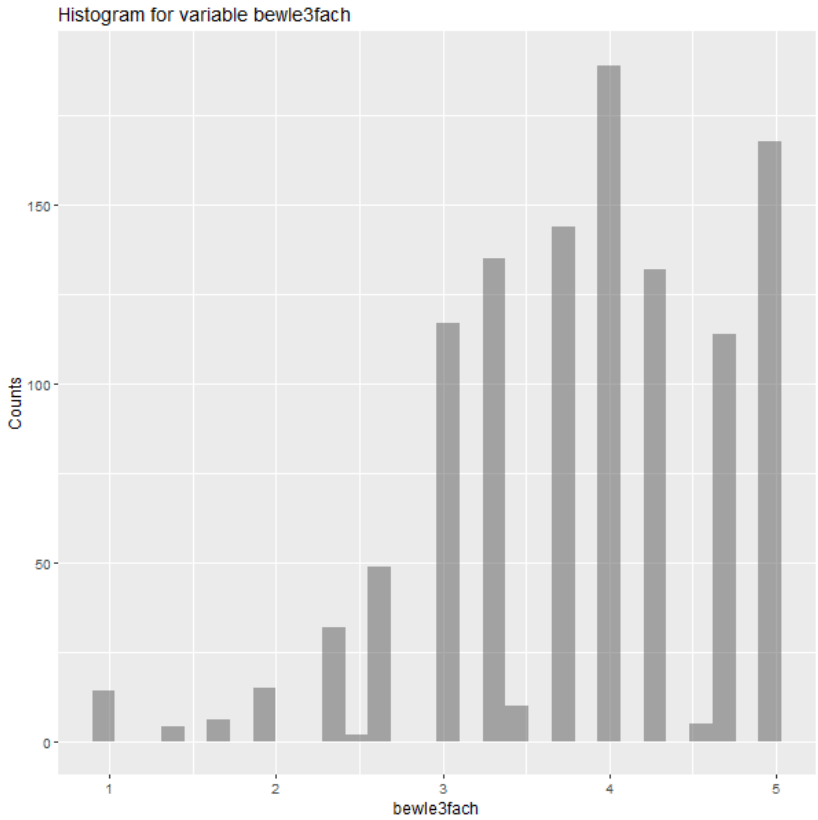



Abb. 24: Histogramm

Über die Syntax wird dies über folgenden Weg erreicht:

```
require(ggplot2);
require(ggthemes);

ggplot(data=datensatzkurz,aes(x =gpa_noten)) +
       geom_histogram(alpha=0.5) +
       labs(x ="bewle3fach",y ="Counts",title= "Histogram for variable
bewle3fach") + theme_grey() + theme(text=element_text(fam-
ily="sans",face="plain",color="#000000",size=12,hjust=0.5,vjust=0.5))
```

Histogramme eignen sich auch zur Prüfung von Normalverteilungsannahmen, ergänzend können hierfür auch Q-Q-Plots erstellt werden (siehe Abschnitt V).

## IV.4 Erstellen von Mehrfachdiagrammen

Bei BSS besteht die Möglichkeit, eine Abbildung mit mehreren Grafiken zu erstellen – mit der Option „facet“ (nebeneinander, untereinander oder auch als beidem („wrap“).

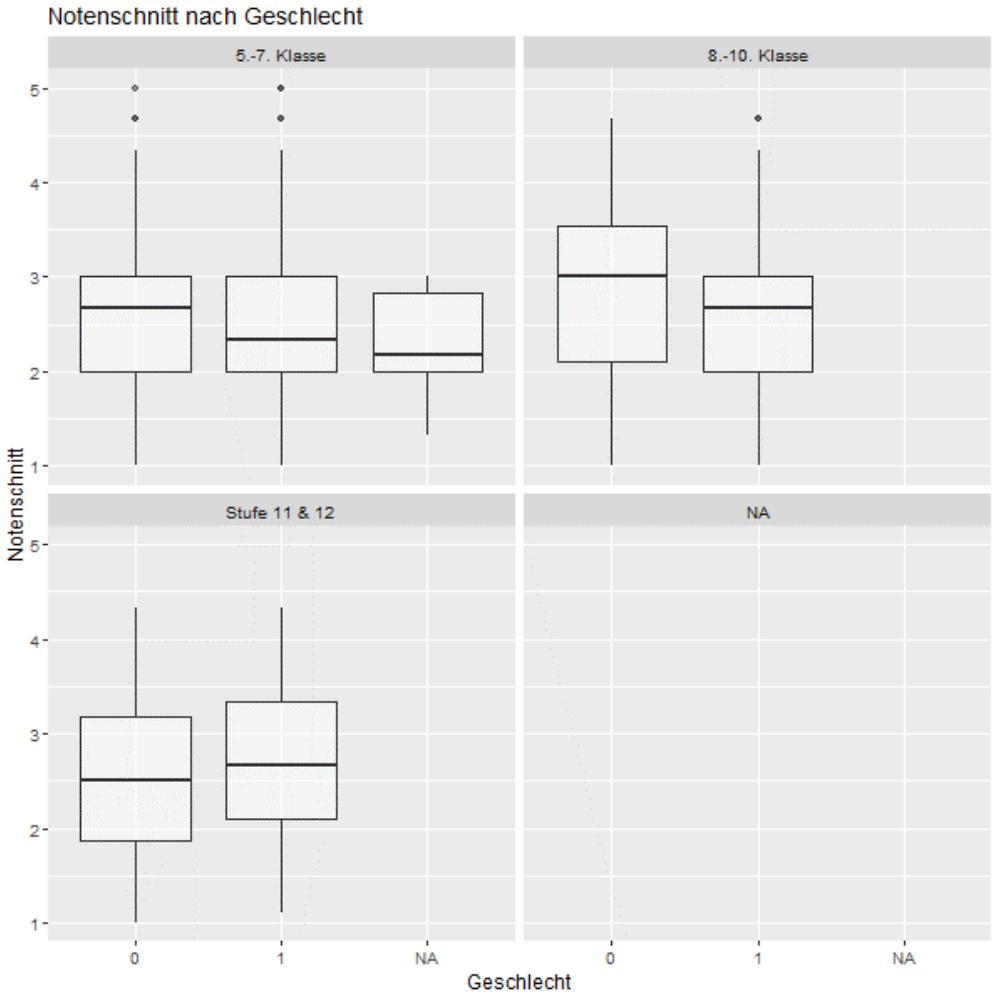



Abb. 25: Mehrfachgrafik mit Boxplots

Hier wurde die NA-Kategorie nicht zuvor entfernt. Die Syntax lautet:

```
## [BoxPlot]
require(ggplot2);
require(ggthemes);

ggplot(data=homes0208,aes(x =geschlw,y =
gpa_noten)) +
 geom_boxplot(alpha =0.5) +
 labs(x ="geschlw",y ="gpa_noten",title=
"Boxplot for variable gpa_noten,group by
geschlw") +
 xlab("Geschlecht") +
 ylab("Notenschnitt") +
 ggtitle("Notenschnitt nach Geschlecht")+
 facet_wrap(~schulstuf3) +
theme_grey() + theme(text=element_text(family="sans",face="plain",
color="#000000",size=12,hjust=0.5,vjust=0.5))}
} )
```

Die Grafiken können über die R Syntax nach Wunsch weiter angepasst werden (Farben, Schriftarten und -größen, Skalierungen). Weitere Hinweise dazu finden sich in der Hilfe zum Paket ggplots2 oder bspw. unter https://r-intro.tadaa-data.de/book/visualisierung.html.

# V. Durchführung von statistischen Analysen

Die Durchführung statistischer Analysen ist schon angeklungen, in diesem Abschnitt sollten exemplarisch statistische Analysen vorgestellt werden, die mit dem Menüführung aufgerufen werden können. Dabei steht die Anwendung des Programms im Mittelpunkt, für eine Erklärung der Verfahren sollte auf entsprechende Lehrbücher (wie bspw. Backhaus et al. 2013, Bortz Döring 200) zurückgegriffen werden. Ausführlich dargestellt werden im Folgenden die Durchführung von 1) T-tests und Varianzanalysen unabhängiger Stichproben, 2) einer linearen Regression, 3) einer binär logistischen Regression sowie 4) einer Faktorenanalyse – mit dem Plan, dies zukünftig weiter auszuweiten.

Die Attraktivität der Nutzung von BSS für entsprechende Analysen liegt für den Autor im Zusammenspiel von den in BlueSky Statistics implementierten Methoden und Routinen und den zugehörigen leicht exportierbaren und weiterverarbeitbaren Ausgaben sowie der Möglichkeit, die Analysen über weitere R Pakete und die R Syntax ergänzen zu können (wird im Folgenden auch dargestellt). Aber auch wenn die Analysen nur über den klassischen R-Code aufgerufen werden, sieht der Autor den erleichterten Datenimport, Datenzugang und das einfache Datenmanagement von BSS als eine Vereinfachung des Arbeitsprozesses an.

## V.1 Durchführung T-test/Varianzanalyse unabhängiger Stichproben

### V.1.1 Durchführung T-Test zweier unabhängiger Stichproben

Für den Vergleich von Mittelwerten zweier Gruppen kann bei Vorliegen entsprechender Voraussetzungen der T-Test eingesetzt werden.

Zwei Voraussetzungen sollten dazu geprüft werden

a) Normalverteilung der abhängigen metrischen Variablen
b) Varianzhomogenität zwischen den beiden Gruppen

Bei der Verwendung des Menüs von BlueSky Statistics sollte die Normalverteilung vorab geprüft werden, die Prüfung der Varianzhomogenität zwischen den Gruppen wird in einem Prozedere mit dem T-test getestet.

**Schritt 1:** Prüfung der Normalverteilung

Dieses als Test kann über den Shapiro-Wilk-Test oder den Kolmogorov-Smirnov-Test erfolgen, diese Tests sind allerdings sehr „empfindlich“ und werden bei großen Stichproben durch die Stichprobengröße schnell signifikant. Empfehlenswert ist daher die grafische Inspektion der Residuen über Histogramm und Q-Q-Plot. Der Aufruf des Histogramm wurde eben beschrieben, ein Q-Q-Plot wird über „Graphics > Q-Q-Plot“ aufgerufen (analog ist auch der Aufruf eines P-P-Plots möglich).

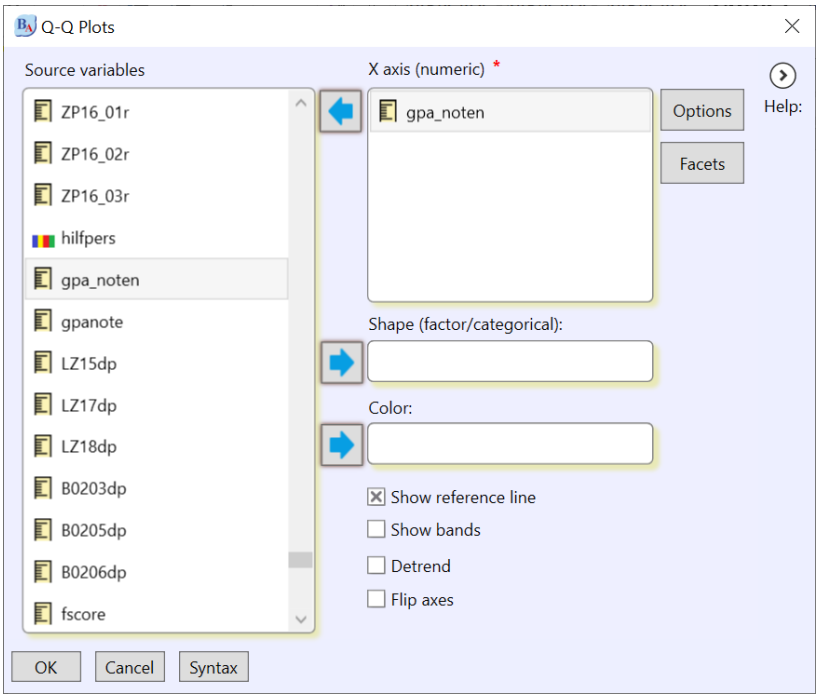



Abb. 26: Menüfenster zur Erstellung eines Q-Q-Plot

Im Ergebnis zeigt sich folgende Grafik – für eine Normalverteilung sollte sich die Verteilung möglichst gut an die Grade „anschmiegen“.

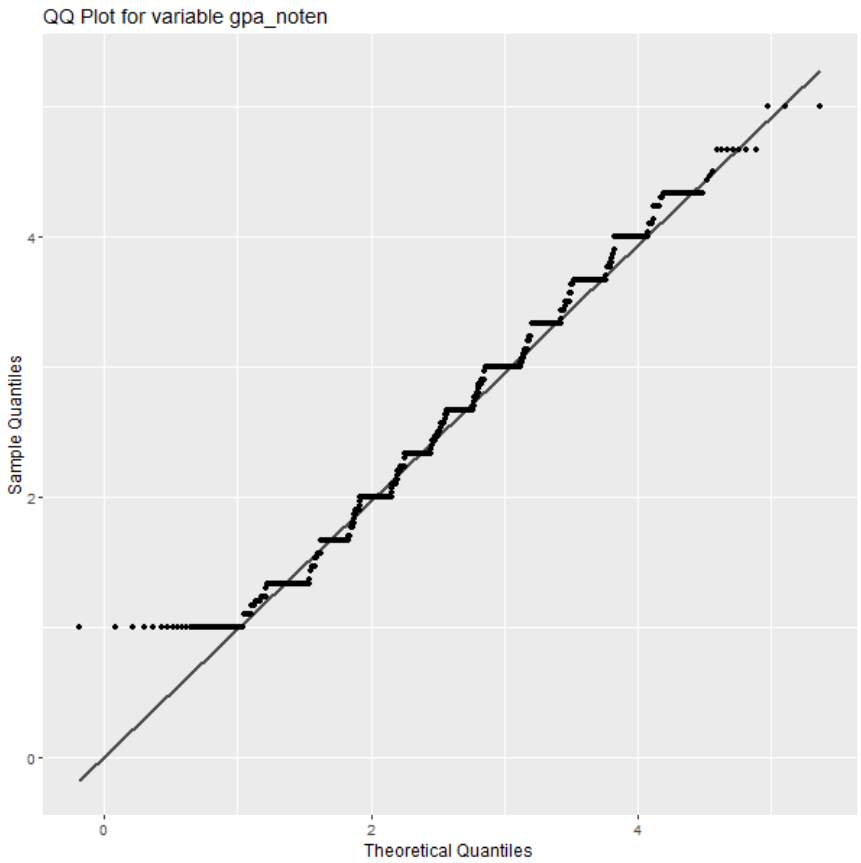



Abb. 27: Q-Q-Plot zur Inspektion der Normalverteilung der Residuen

**Schritt 2:** Voraussetzung aller Varianzanalysen ist die Homogenität der Varianzen in den Teilgruppen. Der Levene-test auf Varianzgleichheit in den zwei Gruppen und der eigentliche T-Test wird in einem Schritt aufgerufen über die Menüleiste mit „Analysis“ > „Means“ > „T-test, independent sample“).

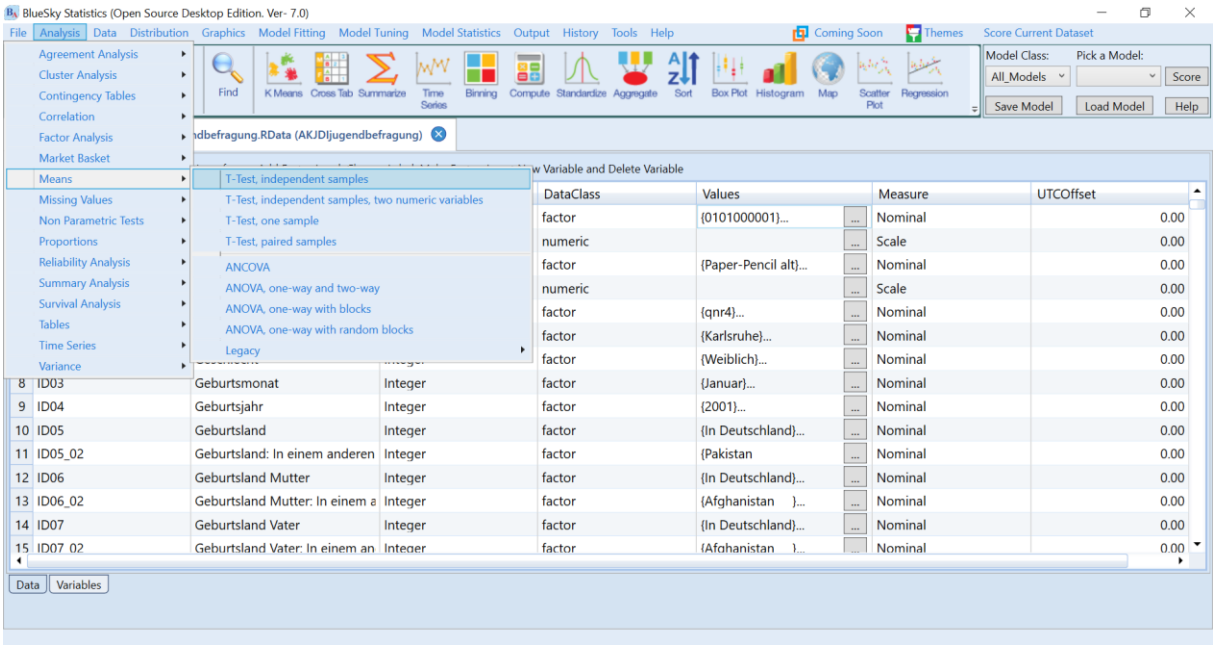



Abb. 28: Menüführung zur Durchführung eines T-Tests

Eingabe der abhängigen Variablen und der Gruppenvariablen

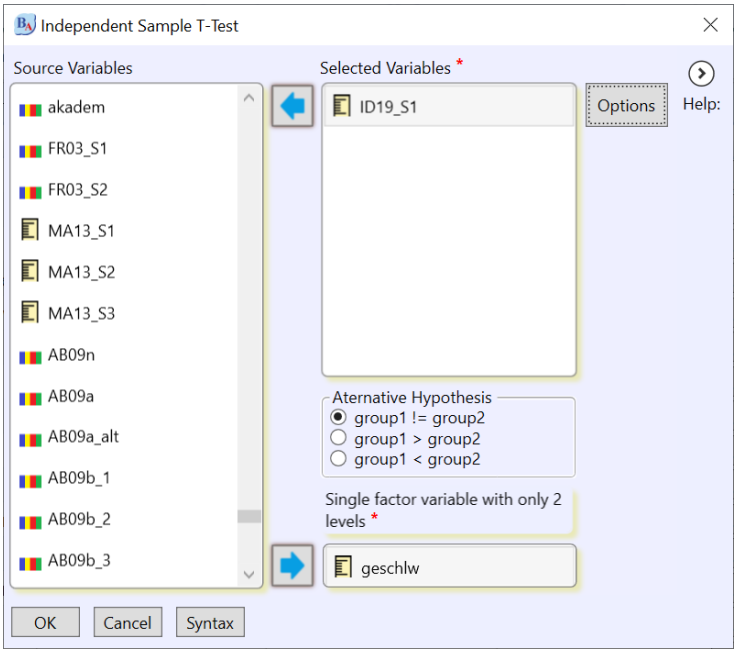



Abb. 29: Unterfenster zur Spezifizierung eines T-Tests für unabhängige Gruppen

In der Option „Alternative Hypotheses“ kann die Richtung des zu testenden Unterschieds vorgegeben werden.

Als Ausgabe zeigt sich folgende Ergebnisdarstellung:

Group Statistics

| | N | Mean | Std Deviation | Std Error Mean |
|---|---|---|---|---|
| ID19_S1 | 173 | 2.2861 | 0.5794 | 0.0441 |

Independent Samples Test

| | | Levene's Test for Equality | | T-Test for equality of means | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | F | Sig. | t | df | Sig.(2-tail) | Mean Difference | Std. Error Difference | Confidence interval of the Diff. lower | Upper |
| ID19_S1 | Equal variances assumed | 2.1385 | 0.1444 | -1.6707 | 400 | 0.0956 | -0.104 | 0.0014 | -0.2263 | 0.0184 |
| | Equal variances not assumed | | | -1.6959 | 388.222 | 0.0907 | -0.104 | | -0.2245 | 0.0166 |

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tab. 6: Ausgaben zum T-Test

Die Tabelle 12 ist ganz offensichtlich der SPSS-Darstellung von T- Test-Ergebnissen nachempfunden: Ist der Levene-Test nicht signifikant, kann der oberen Ergebniszeile gefolgt werden. Die untere Ergebniszeile gibt das Ergebnis des *Welch-Tests* wieder und wird bei signifikantem Unterschied der Varianzen im Levene-Test relevant.

Falls ein Effektstärkemaß gewünscht wird, kann Cohens d berechnet werden, dieses ist aber kein Teil von BSS und muss über R und die Syntax angefordert werden. Dies geht beispielsweise mit dem Paket *lsr*. Nach Installation des Pakets funktioniert das mit folgendem Befehl:

```
require(lsr)
cohensD(daten02$ID19_S1 ~ daten02$geschlw)
```

### V.1.2 Durchführung Einfaktorielle Varianzanalyse/ANOVA

Die Durchführung einer einfaktoriellen oder zweifaktoriellen Varianzanalyse kann mit der Menüleiste über „Analysis“ > „Means“ > „ANOVA, one way and two way“ erfolgen (vgl. Abb. 30).

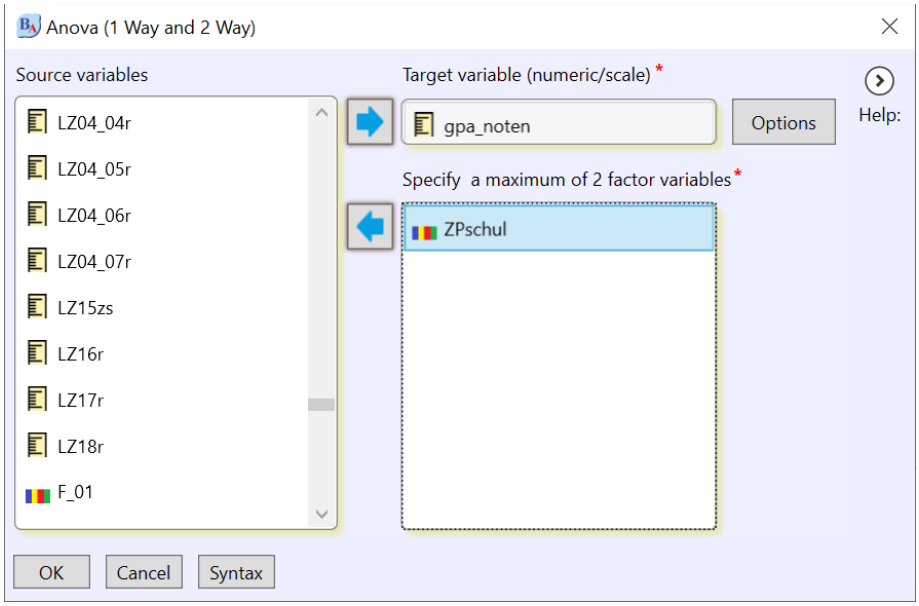



Abb. 30: Menüfenster für eine ein- oder zweifaktorielle Varianzanalyse

Im Unterdialog kann der Levene-Test für Varianzgleichheit angefordert werden und auch weitere Diagnosegrafiken.

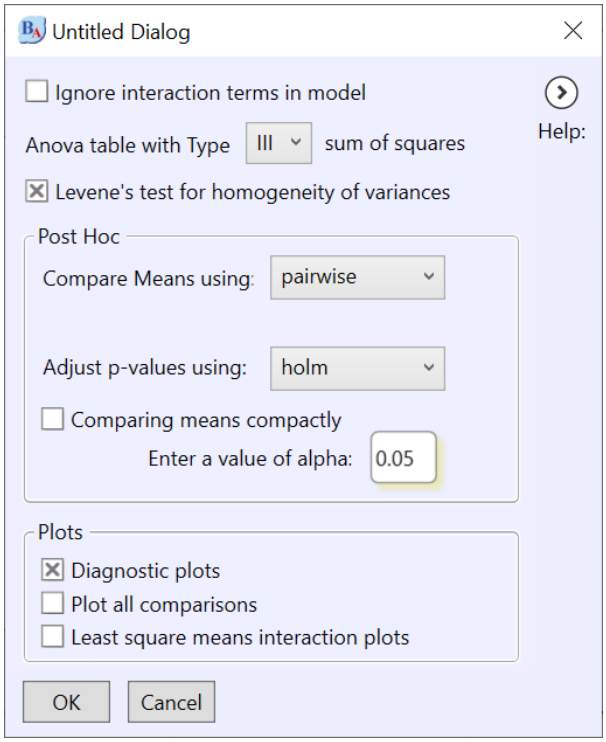



Abb. 31: Untermenü zur Anforderung u.a. des Levene-Tests auf Varianzgleichheit

Im Output wird zunächst eine Tabelle mit den deskriptiven Statistiken abgebildet.

Summaries for gpa_noten by factor variable schulstuf3

| schulstuf3 | n | mean | median | min | max | sd | variance |
|---|---|---|---|---|---|---|---|
| 5.-7. Klasse | 584 | 2.5101 | 2.5 | 1 | 5 | 0.7828 | 0.6128 |
| 8.-10. Klasse | 494 | 2.6738 | 2.6667 | 1 | 4.6667 | 0.8777 | 0.7703 |
| Stufe 11 & 12 | 179 | 2.6422 | 2.6667 | 1 | 4.3333 | 0.8331 | 0.694 |
| NA | 16 | NA | NA | Inf | -Inf | NA | NA |

Dann folgt eine Tabelle mit dem Ergebnis des Levene-Tests auf Varianzgleichheit (der auch getrennt angefordert werden kann – hier für eine Variable mit 7 Stufen).

Levene's test for homogenity of variances center =base::mean

| | Df | F value | Pr(>F) |
|---|---|---|---|
| group | 7 | 1.635 | 0.122 |
| | 1185 | NA | NA |

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dann zeigt BSS eine Tabelle mit dem Ergebnis der Varianzanalyse. Hier wird ein signifikanter Effekt für Faktorstufen ausgewiesen.

Anova table with type III sum of squares for gpa_noten by ZPschul

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| ZPschul | 7 | 93.502 | 13.357 | 21.695 | 0.000 *** |
| Residuals | 1185 | 729.587 | 0.616 | NA | NA |

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tab. 7: Ausgaben zur einfaktoriellen Varianzanalyse

Darüber hinaus gibt BSS auf die Anforderung sowohl die Post-hoc Testergebnisse (Vergleich der Mittelwerte einzelner Gruppenstufen) sowie die Estimated Marginal Means pro Faktorstufe/Gruppe aus (hier nicht dargestellt).

## V.2 Durchführung einer linearen Regression

BSS bietet über die Menüführung zwei Wege zur linearen Regression, der erste (a) ist speziell für BSS entwickelt, während sich der zweite (b) stärker an der klassischen R-Syntax orientiert.

a) Der BSS-Ansatz der linearen Regression (Menüleiste: „Model Fitting > Linear Regression)

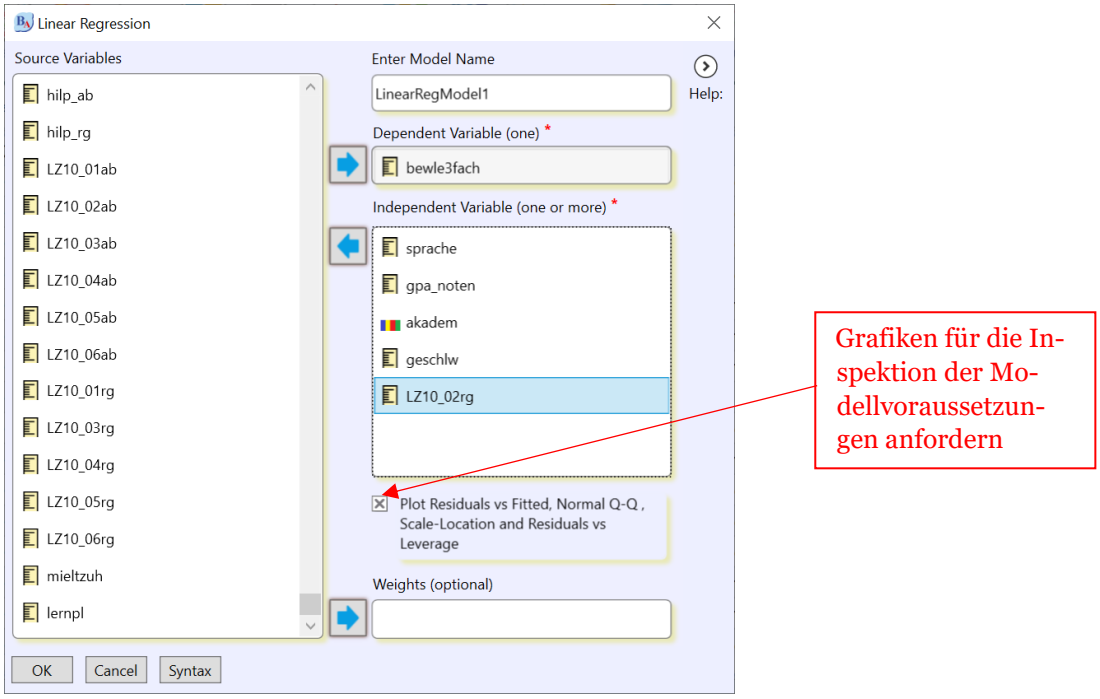



Abb. 32: Untermenü zur Modelformulierung der linearen Regression

*Spezielle BSS-R-Syntax*

```
LinearRegModel1= BSkyRegression(depVars ='bewle3fach',indepVars
=c('sprache','gpa_noten','akadem','geschlw','LZ10_02rg'),dataset="da-
ten03.sav")
```

Output (u.a.)

*Model: lm(formula = bewle3fach ~ sprache + gpa_noten + akadem + geschlw + LZ10_02rg, data = .GlobalEnv$Homes.0208gdt.sav)*

**LM Summary**

| Residual Std. Error | df | R-squared | Adjusted R-squared | F-statistic | numdf | dendf | p-value |
|---|---|---|---|---|---|---|---|
| 0.7708 | 983 | 0.1667 | 0.1625 | 39.3294 | 5 | 983 | 0 |

*Coefficients(284 observations deleted due to missing values)*

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| **(Intercept)** | 4.8152 | 0.0995 | 48.4015 | <.001*** |
| **sprache** | -0.1032 | 0.0744 | -1.3875 | 0.1656 |
| **gpa_noten** | -0.3814 | 0.0306 | -12.468 | <.001*** |
| **akadem** | -0.0132 | 0.05 | -0.2636 | 0.7921 |
| **geschlw** | 0.0221 | 0.0504 | 0.4388 | 0.6609 |
| **LZ10_02rg** | 0.2804 | 0.0612 | 4.585 | <.001*** |

*Note.* Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Tab. 8: Ausgabe zur Linearen Regression

Diese Tabellen können als Tabellen direkt nach Word exportiert werden oder mit der rechten Maustaste („copy to clipboard“) bspw. in MS Excel oder libreoffice calc eingefügt werden (siehe den Punkt Berichte schreiben und Ausgaben exportieren oben) – copy paste in Textverarbeitungsprogramme funktioniert leider nicht.

b) Alternative Menüführung zur linearen Regression (Menüleiste: „Model Fitting > Linear Regression with Formula“)

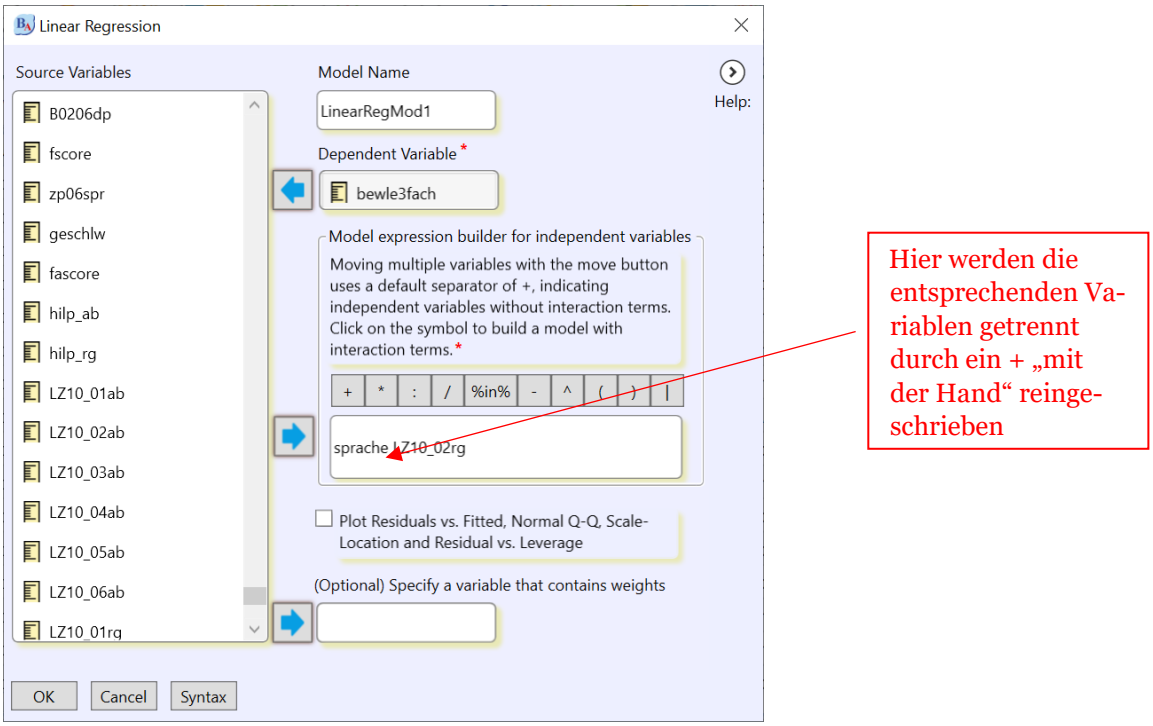



Abb. 33: Unterfenster zur Modellspezifikation einer Linearen Regression (Alternatives Vorgehen)

*Einfache/ „klassische“ R Syntax*:

```
LinearRegMod2 = lm(bewle3fach~sprache+akadem+LZ10_02rg,data=datensatz02)
```

Output:

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.3831 -0.4516  0.0739  0.5441  1.9558

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      4.81518    0.09948  48.402  < 2e-16 ***
sprache                         -0.10320    0.07438  -1.387    0.166
gpa_noten                       -0.38140    0.03059 -12.468  < 2e-16 ***
geschlw                          0.02211    0.05039   0.439    0.661
LZ10_02rg                        0.28038    0.06115   4.585 5.12e-06 ***
akademmind. 1 Elternteil Akadem. -0.01319    0.05003  -0.264    0.792
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7708 on 983 degrees of freedom
  (284 observations deleted due to missingness)
Multiple R-squared:  0.1667,     Adjusted R-squared:  0.1625
F-statistic: 39.33 on 5 and 983 DF,  p-value: < 2.2e-16
```

*Prüfung der Voraussetzungen der Regression*: Zur linearen Regression gehört stets die Prüfung der Voraussetzung der Regression (vgl. hierzu bspw. Backhaus et al. 2016, S. 97ff.). BSS bietet einige Optionen der Prüfung der Voraussetzungen über die Menüfenster, die durch weitere R-Syntaxbefehle ergänzt werden können – hier bietet sich das Paket *lmtest* an. Grafiken zur Inspektion des Modells können im Regressionsfenster schon mit angeklickt werden, andere Modelinformationen können über das Menüfenster „Modell Statistics“ abgerufen werden.

1. Prüfung Normalverteilung der abhängigen Variablen

*(siehe Punkt T-Test/Varianzanalysen)*

2. Prüfung linearer Zusammenhang: Möglich ist hier der Rainbow-test (Paket *lmtest*)

```
raintest(LRModel1)

Rainbow test
data: LRModel1

Rain = 1.0867, df1 = 549, df2 = 542, p-value = 0.1661
```

3. Prüfung Normalverteilung der Residuen: Hier hilft die Grafische Inspektion mit Q-Q-Plots, die im Regressionsfenster mit dem Kreuz bei „Plot Residuals …“ angefordert werden kann.

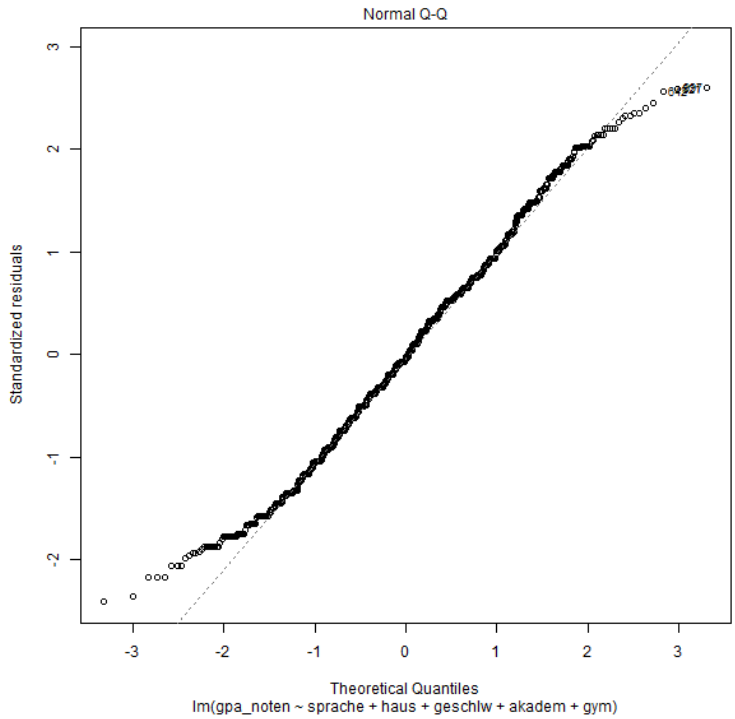



Abb. 34: Plot der studentisierten gegen die theoretischen Residuen zur Untersuchung der Normalverteilung der Residuen

Wenn die Residuen normalverteilt sind, sollten sie auf der gestrichelten Geraden liegen.

4. Prüfung auf Homoskedastizität: Auch dies kann grafisch mit Anklicken der Grafikoption im Menüfenster überprüft werden

Dafür werden die standardisierten Residuen gegen die durch das Modell vorhergesagten Werte geplottet.

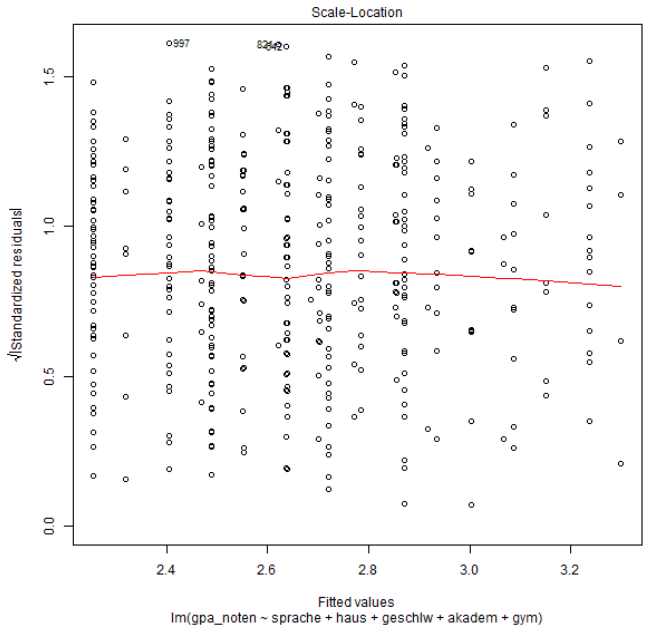



Abb. 35: Standardisierte, absolute Residuen gegen vorhergesagte Werte zur Untersuchung der Homoskedastizität des Fehlers

Bei Homosekdastizität streuen die Punkte eher unsystematisch um die Linie, eine Dreiecksform oder eine U-Form – also systematische Veränderungen – weisen auf Heteroskedastizität zeigt sich (vgl. hierzu bspw. Backhaus et al. 2016, S. 103). Anders formuliert: Wenn sich die Residualwerte mit wachsendem Wert von x systematisch verändern ist davon auszugehen, dass Heteroskedastizität der Residualwerte vorliegt (Janssen & Laatz 2017, S. 444). Zudem lässt sich über den Breusch-Pagan-Test aus dem lmtest-Paket Homoskedastizität testen.

```
bptest(LRModel1)
```

```
studentized Breusch-Pagan test
data: LRModel1
BP = 10.064, df = 5, p-value = 0.07343
```

5. Prüfung Multikollinearität: Über das Menüfenster „Modell statistics“ kann mit der Option „Variance Inflation Factors“ das Modell auf mögliche Multikollinearität der Variablen geprüft werden (Empfehlung: VIF nicht > 5, vgl. Akinwande et al.2015, S. 750).

6. Prüfung Autokorrelation: Vor allem für Panel- und Längsschnittdaten kann sich das Problem der Autokorrelation stellen (Fehlerterme einer Person sind sich ähnlicher), hier wird in der Regel der Durbin-Watson-Test genutzt (auch möglich über das R- Paket lmtest)

```
durbinWatsonTest(LRModel1)
```

```
lag Autocorrelation D-W Statistic p-value
1   0.08728603   1.824524  0.008

 Alternative hypothesis: rho != 0
```

7. Prüfung wie gut das Modell an die Daten angepasst ist

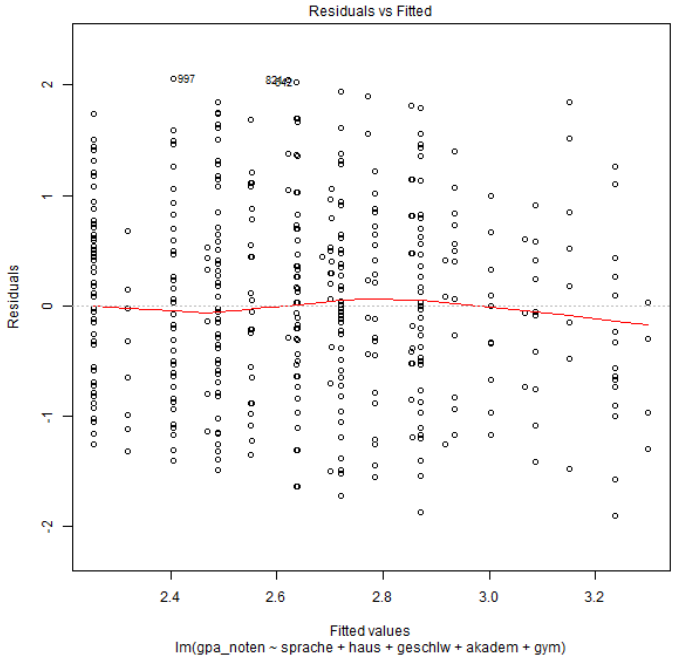



Abb. 36: Plot der Residuen gegen vorhergesagten Werte zur Untersuchung der Anpassung des Modells an die Daten

Dabei sollte keine systematische Abweichung auftreten (U-Form oder Trend). Dazu kann auch ein Test auf korrekte Spezifizierung angefordert werden.

```
resettest(LRModel1)
```

```
      RESET test
data: LRModel1
RESET = 4.2873, df1 = 2, df2 = 1089, p-value = 0.01397
```

Eine facet-Zusammenstellung entsprechender Diagnosegrafiken können auch über das R Paket *performance* angefordert werden:

```
require(performance)
check_model(LRModel1)
```

## V.3 Binäre logistische Regression mit BSS

Auch für die binäre logistische Regression bietet BSS über das Menü zwei Wege an, die derselben Logik wie bei der linearen Regression folgen. Hier wird die Option „Model Fitting > logistic regression“ vorgestellt.

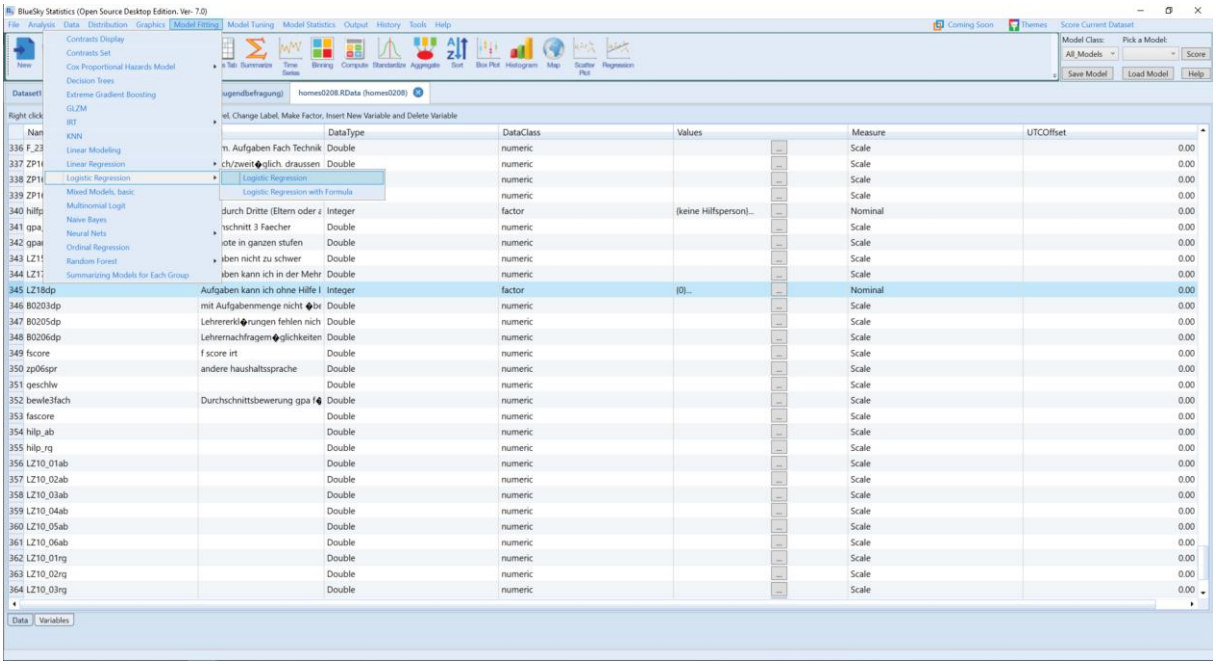



Abb. 37: Menüführung zur Durchführung einer logistischen Regression

Wiederum öffnet sich ein Regressionsfenster zur Modellspezifikation. Im Folgenden ist ein Modell mit sechsunabhängigen Variablen spezifiziert.

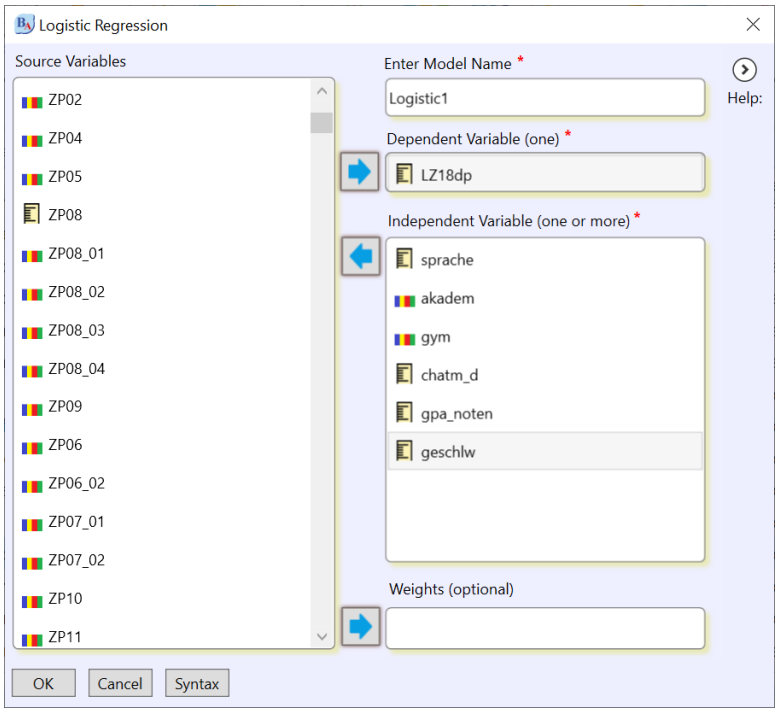



Abb. 38: Untermenü für die logistische Regression

Im Outputfenster zeigen sich anschließend eine Reihe von Angaben zum Modell (Modellformel, Modelstatistiken). Für die geschätzten Koeffizienten gibt BSS zwei Tabellen aus: Zunächst eine Tabelle mit den Koeffizienten, der Standardfehlern und den Werten für die Signifikanztest.

*Coefficients(283 observations deleted due to missing values)*

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| **(Intercept)** | 2.7801 | 0.452 | 6.1502 | <.001*** |
| **sprache** | 0.0989 | 0.2578 | 0.3836 | 0.7013 |
| **mind. 1 Elternteil Akadem.** | 0.0206 | 0.1784 | 0.1157 | 0.9079 |
| **gymgym** | 0.5243 | 0.2362 | 2.2196 | 0.0264 * |
| **chatm_d** | 0.6408 | 0.197 | 3.2527 | 0.0011 ** |
| **gpa_noten** | -0.8028 | 0.1148 | -6.996 | <.001*** |
| **geschlw** | -0.0156 | 0.1764 | -0.0883 | 0.9296 |

*Note.* Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Die Odds ratio und die Konfidenzintervalle der Schätzer werden in einer eigenen Tabelle ausgegeben:

*Odds ratio(OR) and 95% Confidence interval*

| | OR | 2.5 % | 97.5 % |
|---|---|---|---|
| **(Intercept)** | 16.1208 | 6.7363 | 39.7113 |
| **sprache** | 1.104 | 0.6764 | 1.8649 |
| **mind. 1 Elternteil Akadem.** | 1.0209 | 0.7195 | 1.4494 |
| **gym** | 1.6892 | 1.0549 | 2.6678 |
| **chatm_d** | 1.898 | 1.2843 | 2.7833 |
| **gpa_noten** | 0.4481 | 0.3565 | 0.5593 |
| **geschlw** | 0.9845 | 0.6951 | 1.3888 |

Dazwischen gibt es eine Tabelle zur Modellgüte, die unter anderem das Pseudo-$R^2$ Mc Fadden enthält:

*McFadden R2*

| | llh | llhNull | G2 | McFadden | r2ML | r2CU |
|---|---|---|---|---|---|---|
| | -426.59 | -562.6711 | 272.1621 | 0.2418 | 0.2404 | 0.3539 |

Tab. 9: Ausgabe zur logistischen Regression

Weitere Pseudo-$R^2$ Maße können über die R-Syntax angefordert werden (bspw. mit dem Paket *DescTools*, das mit BSS installiert wird). Der R Syntax Befehl für das Cox & Snell $R^2$ sowie das Nagelkerke $R^2$ ist dabei:

```
require(DescTools)
PseudoR2(Logistic1, c("CoxSnell", "Nagel"))
```

Im Outputfenster erscheint dann (unformatiert):

```
CoxSnell    Nagelkerke
0.2403605   0.3539258
```

Die Güte der Klassifikation durch das statistische Modell kann durch den Hosmer-Lemeshow-Test geprüft werden, der beispielsweise mit R-Paket *performance* über die Syntax angefordert werden kann.

```
require(performance)
performance_hosmer(Logistic1)
```

# Hosmer-Lemeshow Goodness-of-Fit Test

```
 Chi-squared: 8.348
df: 8
p-value: 0.400
Summary: model seems to fit well.
```

Für die Interpretation der Koeffizienten/Schätzer kann über die Odds Ratio, die ein Relatives Maß darstellen, hinaus die average marginal effects (AME) verwendet werden (vgl. hierzu Leeper 2018). Dies ist mit der Installation und dem Aufrufen des Pakets „margins“ über die R-Syntax möglich. Die Ausgabe erfolgt wie alle R-Code-gestützten Analysen, die nicht in BSS implementiert sind, unformatiert.

Befehle über die R-Syntax- (nach Paketinstallation):

```
require(margins)
ame_logit_ohilf <- margins(Logistic1)
summary(ame_logit_ohilf)
```

Im Ergebnis

| | factor | AME | SE | z | p | lower | upper |
|---|---|---|---|---|---|---|---|
| mind. 1 | Elternteil Akadem. | 0.0028 | 0.0241 | 0.1157 | 0.9079 | -0.0444 | 0.0499 |
| | chatm_d | 0.0864 | 0.0262 | 3.2964 | 0.0010 | 0.0350 | 0.1378 |
| | geschlw | -0.0021 | 0.0238 | -0.0883 | 0.9296 | -0.0487 | 0.0445 |
| | gpa_noten | -0.1083 | 0.0147 | -7.3769 | 0.0000 | -0.1370 | -0.0795 |
| | gymgym | 0.0789 | 0.0391 | 2.0169 | 0.0437 | 0.0022 | 0.1556 |
| | sprache | 0.0133 | 0.0348 | 0.3837 | 0.7012 | -0.0548 | 0.0815 |

Tab. 10: Ausgabe zur Average Marginal Effects einer logistischen Regression

*Testung der Voraussetzungen der logistischen Regression:*

Bei der logistischen Regression sollte mindestens getestet werden auf

a) Das mögliche Vorhandensein bedeutsamer Ausreißer
b) Mögliche Multikollinearität der unabhängigen Variablen
c) sowie die Linearität des Logits

zu a) Mögliche bedeutsame Ausreißer können mit dem Paket car getestet werden. Einzelne bedeutsame Ausreißer können gerade bei kleineren Stichproben wie in der linearen Regression die Funktionsschätzung ungünstig beeinflussen.

#Laden des Pakets *car* und Durchführung Outliertest

```
require(car)
outlierTest(LogisticR1)

No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:   rstudent unadjusted p-value Bonferroni p725 -2.574994
0.010024      NA
```

#Zudem kann mit dem Paket *car* eine Grafik zur Inspektion von Ausreißern erzeugt werden:

```
influenceIndexPlot(LogisticR1)
```

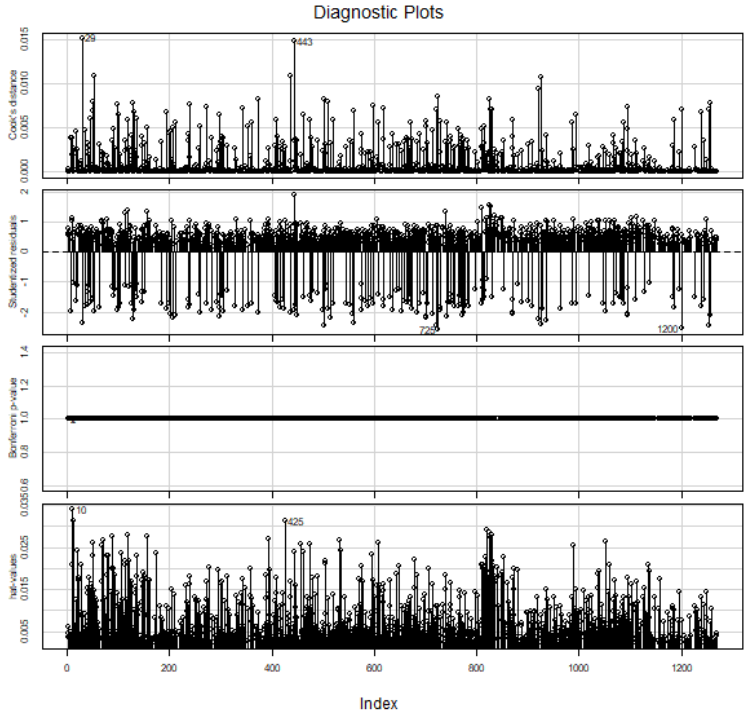



Abb. 39: Grafik zur Ausreißerdiagnostik im Paket *car*

Die Grafik zeigt sowohl the Cook's distance, die studentisierten Residuals, die Bonferroni p Werte sowie hh-Werte.

Zu b) Für die Testung auf Multikollinearität können nach Modellschätzung über die Menüoptition „Model statistics" können die „Variance Inflation Faktors" (VIF) aufgerufen werden (s. auch die Angaben zur linearen Regression, das Modell muss oben rechts im Hauptmenüfenster aufgerufen sein).

Variance-inflation factors

| sprache | akadem | geschlw | haus | gpa_noten | ZP04met |
|---|---|---|---|---|---|
| 1.049 | 1.086 | 1.028 | 1.132 | 1.075 | 1.027 |

Tab. 11: Ausgabe zur Variance Inflation Factors (VIF)

Zu c) Linearität des Logits: Inspiziert werden sollte, ob es jeweils eine lineare Beziehung zwischen der/den metrischen unabhängigen Variable(n) und dem Logit (Log-Funktion) der abhängigen Variable gibt. Dies kann mit dem Paket *car* inspiziert werden

```
residualPlots(LogisticR1)
```

Dabei werden sowohl Linearitätstests sowie Grafiken erzeugt:

```
residualPlots(LogisticR1)

Test stat Pr(>|Test stat|)
geschlw
sprache
gpa_noten     2.2443          0.13411
ZP04m         4.4830          0.03423 *
LZ08_01n      0.3784          0.53846
gym
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

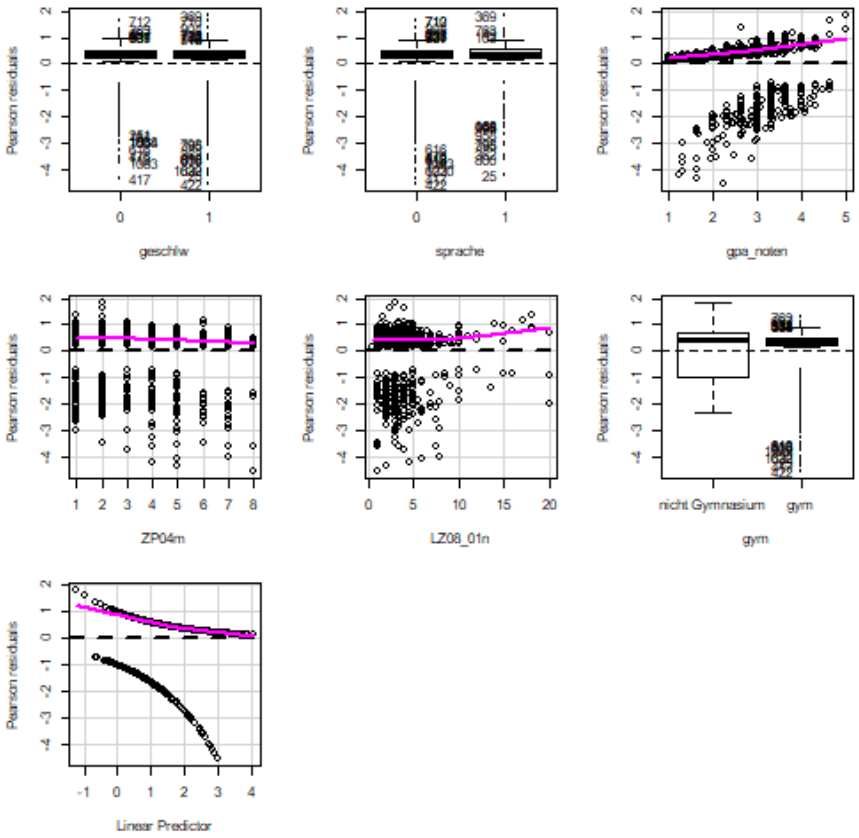



Abb. 40: Facet-Grafik zur Diagnostik der Linearität der Logits mit den Prädiktoren im Paket *car*

## V.3 Ordinale Regression mit BSS

Für die Analyse ordinaler abhängiger Variablen lässt sich die Prozedur unter „Model Fitting > Ordinal Regression“ aufrufen (zur ordinalen Regression vgl. bspw. Janssen & Laatz 2017, S. 465 ff., Große Schlarmann & Galatsch 2014). Wichtige Voraussetzung für die Nutzung der Menüführung ist, dass die Variable vorher als „ordered factor“ definiert worden ist.[3]

Grundsätzlich gibt es hier zwei Modellarten: Proportional Odds (oder Kumulative Logit) Modelle und Sequenzielle Modelle/Continuation Ratio Modelle. Vorgestellt wird hier das kumulative Proportional Odds/Kumulative-Logit-Modell, dass hinter der ordinalen Skala eine dahinterliegende latente metrische Struktur annimmt (vgl. Große Schlarmann & Galatsch 2014).

Ohne hier weiter auf die mathematische Theorie dahinter einzugehen, soll hier nur kurz die Operationalierung in BSS gezeigt werden. Über die Menüführung „Model Fitting > Ordinal Regression“ öffnet sich folgendes Menüfenster:

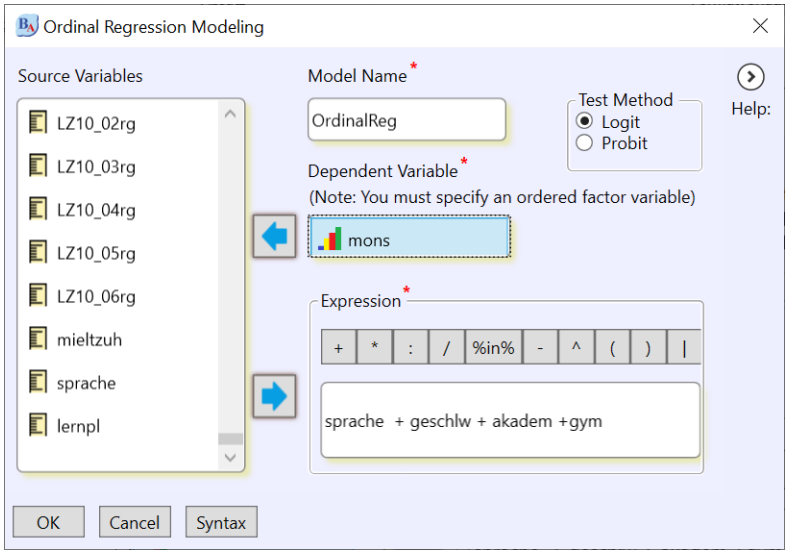



Abb. 41: Unterfenster ordinale Regression

In dem Teilfenster „Expression“ ist es nötig, die unabhängigen Variablen „mit der Hand“ einzugeben (in Verbindung mit einem + Zeichen). Nach Spezifikation und „OK“ erscheint folgender output, der mit der Modellformel beginnt.

*polr(formula=mons ~ sprache + geschlw + akadem + gym, data=homeso208, Hess=TRUE, method='logistic')*

Es zeigen sich zunächst Angaben zur Modellgüte, die vor allem im Modellvergleich interessant werden.

| | SUMMARY.POLR | | |
|---|---|---|---|
| | Residual Deviance | Effective df | AIC |
| | 2987.162 | 8 | 3003.162 |

[3] Das gelingt beispielsweise mit folgender R-Syntax-Logik: `daten02$var24ord <- factor(daten02$var24,ordered = TRUE)`. Und um die neue Variable auch im Datenfenster angezeigt zu bekommen: `BSkyLoadRefreshDataframe(daten02)`

Weiter folgt die Tabelle mit der Koeffizientenschätzung, die SPSS Terminologie unterscheidet hier Schwellen- und Lageschätzer – die Lageschätzer sind die Koeffizienten, die für unabhängigen Variablen geschätzt werden, die Schwellenschätzer Werte für den Übergang von einer Stufe der abhängigen Variablen zur nächsten.

*Coefficients(256 observations deleted due to missing values)*

| | Value | Std. Error | t value | p.value(z) | p.value(t) |
|---|---|---|---|---|---|
| **sprache** | -0.318 | 0.1685 | -1.8871 | 0.0592 . | 0.0594 . |
| **geschlw** | -0.4111 | 0.1156 | -3.5575 | <.001*** | <.001*** |
| **mind. 1 Elternteil Akadem.** | 0.3537 | 0.1147 | 3.0842 | 0.002 ** | 0.0021 ** |
| **Gym** | 0.0591 | 0.1811 | 0.3264 | 0.7441 | 0.7442 |
| **1\|2** | -2.6474 | 0.2154 | -12.2933 | <.001*** | <.001*** |
| **2\|3** | -1.6788 | 0.1961 | -8.5621 | <.001*** | <.001*** |
| **3\|4** | -0.3578 | 0.1885 | -1.8982 | 0.0577 . | 0.058 . |
| **4\|5** | 0.8258 | 0.1907 | 4.3308 | <.001*** | <.001*** |

*Note*. Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Die Odds ratios und die Konfidenzintervalle der Odds ratios für die „Lageschätzer“ werden wie bei der logistischen Regression in eine eigene Tabelle abgebildet.

*Odds Ratio and Confidence Interval*

| | Odds ratio | 2.5 % | 97.5 % |
|---|---|---|---|
| **Sprache** | 0.7276 | 0.523 | 1.0124 |
| **Geschlw** | 0.6629 | 0.5285 | 0.8314 |
| **mind. 1 Elternteil Akadem.** | 1.4243 | 1.1376 | 1.7832 |
| **Gym** | 1.0609 | 0.744 | 1.5128 |

Tab. 12: Ausgaben zur ordinalen Regression

Weitere Angaben zur Modellgüte können über das Menüfenster model statistics aufgerufen werden.

Testung der Voraussetzungen der ordinalen Regression

Zwei Voraussetzungen der ordinalen Regression sollten auf jeden Fall getestet werden.

- Zum Testen der zentralen Annahme *der Proportional odds/“equal slope assumption“* (die Abstände zwischen den Gruppen haben keinen Einfluss auf die Koeffizienten = die logistischen Funktionen für die Wahrscheinlichkeit, mindestens Stufe x zu erreichen, sind für jede Stufe parallel verschoben) bietet sich bspw. das Paket *brant* an, das aus diesem Test „besteht“.

```
require(brant)
brant(oModel1)
```

- Auch sollte auf mögliche Multikollinearität der Prädiktoren geprüft werden – dadurch, dass das ordinale Modell keine Konstante enthält, ist der Aufruf der VIF über die Model Statistics wenig sinnvoll (und erzeugt auch eine Warnmeldung). Sinnvoll ist es, das Ordinale Modell als lineares Modell zu schätzen und sich dann für dieses über das Menü und „Model Statistics“ die VIF ausgeben zu lassen.

## V.4 Lineare Mehrebenenanalyse

Für die Analyse genesteter Daten z.B. „Schüler*innen in Schulen“, „Teilnehmer*innen in Kursen“ bietet sich bei entsprechend hohem Varianzanteil auf Kontextebene Mehrebenenanalysen an (vgl. als Übersicht bspw. Langer 2010). In BSS implementiert sind bislang (einfache) lineare Mehrebenenregressionen, die über „Model Fitting > Mixed Models, basics“ aufgerufen werden können (BSS greift auf das Paket lme4 zurück).

Für die Durchführung bietet sich anfangs immer ein „Leermodell“ an (für das nur die abhängige Variable und die Kontextvariable („Nesting unit“) spezifiziert werden muss).

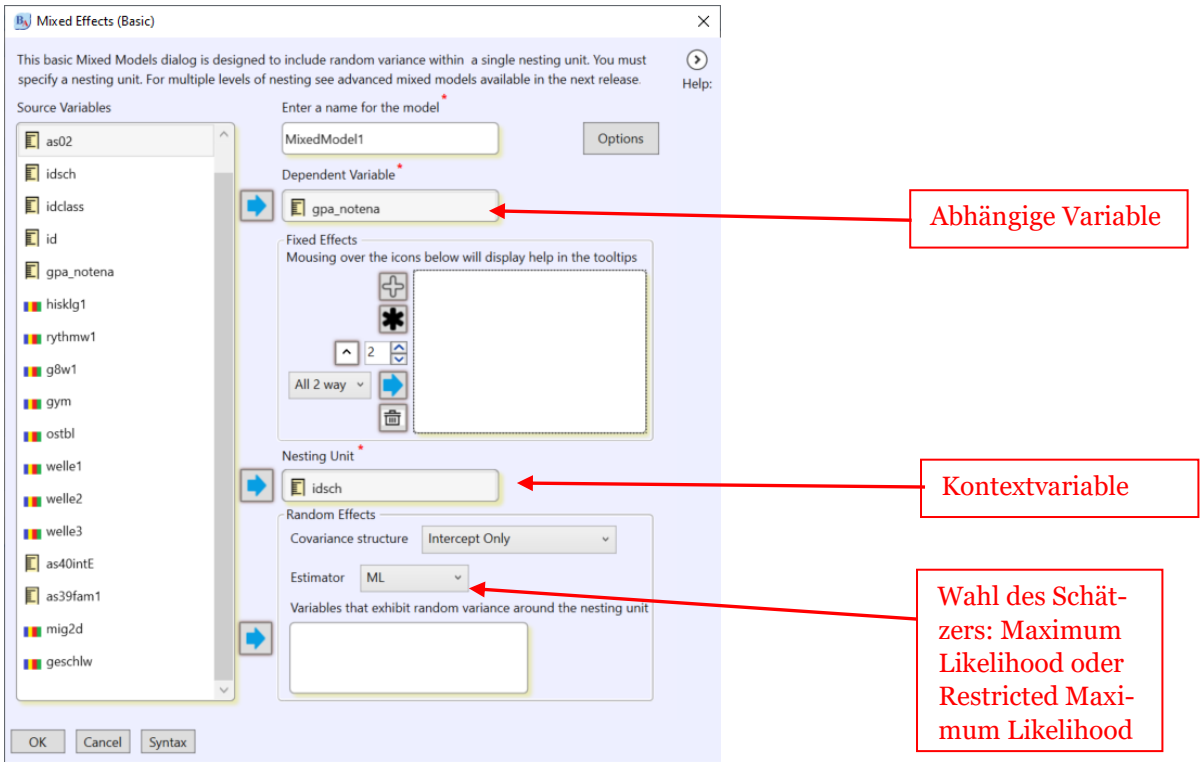



Abb. 42: Menüfenster zur Spezifizierung eines linearen Mehrebenenmodells

Mit diesem Modell ohne unabhängige Variable und nur mit Konstante kann u.a. die Varianzzerlegung geschätzt werden. Als Information muss in dem Unterfenster *Optionen* der ICC angefordert werden, der in einem Leermodell in einem Zwei-Ebenen-Modell (Schüler*innen in Schulen, Patient*innen in Krankenhäusern, Jugendliche in Jugendzentren) den Varianzanteil der Kontextebene abbildet.

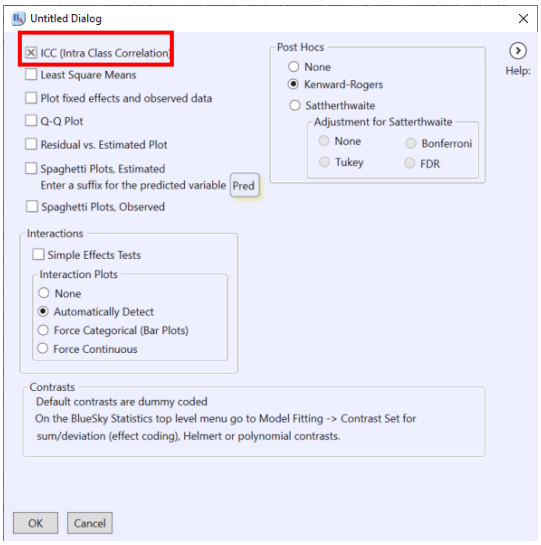


Abb. 43: Untermenü zur Mehrebenenanalyse

In dem umfangreichen Output findet sich dann auch eine entsprechende Angabe zum ICC:

| Intraclass Correlation Coefficient | |
|---|---|
| | Values |
| adjusted | 0.236 |
| conditional | 0.236 |

Tab. 13: Varianzzerlegung im Nullmodell

In diesem Beispiel entfallen 23,6% der Varianz der Durchschnittsnote der drei Hauptfächer auf den Kontext der jeweiligen Einzelschule.

Ein dann im zweiten Schritt spezifiziertes Modell (mit variierendem Intercept aber fixem Slope, ohne Interaktionen der unabhängigen Variablen) wird wie folgt aufgerufen:

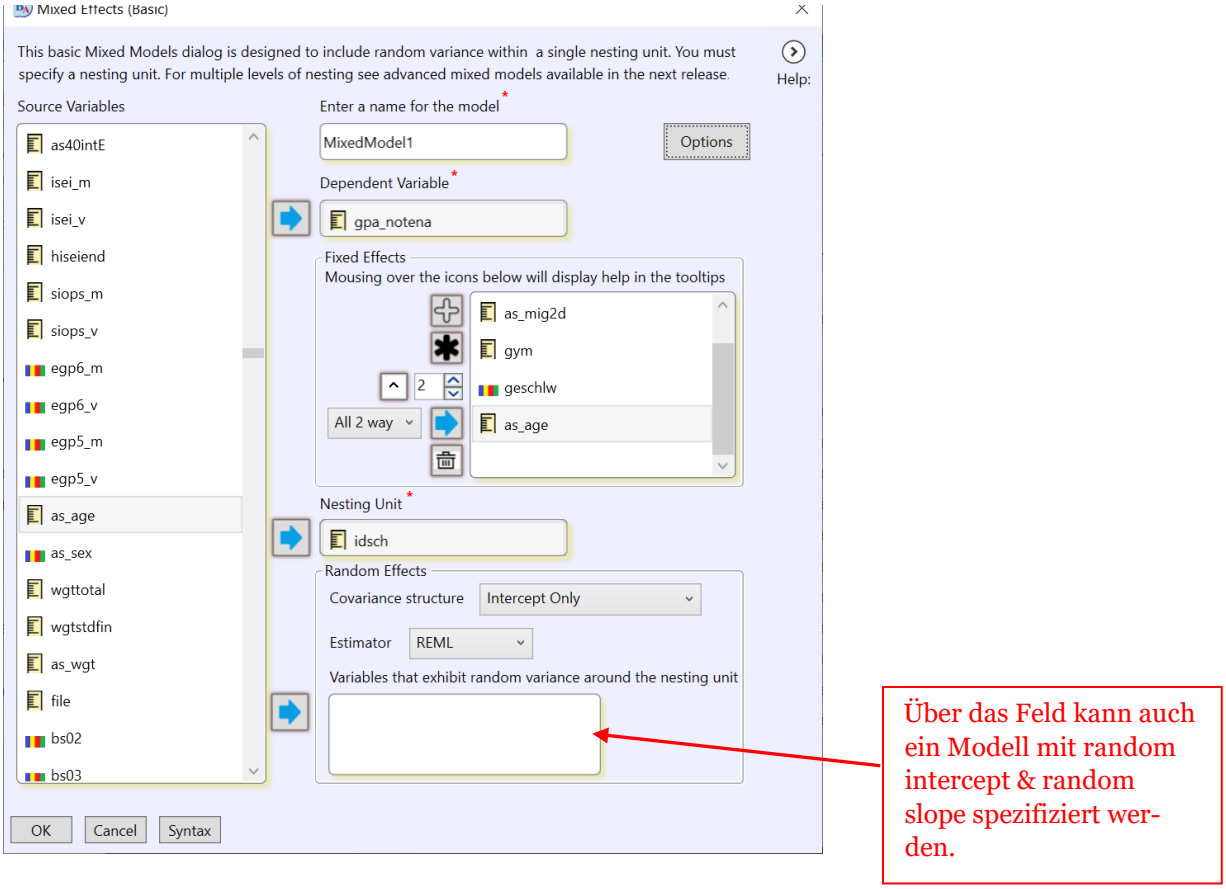



Abb. 44: Aufruf eines linearen Mehrebenenmodells mit random Intercept

Über die „Nesting unit“ wird die Gruppierungs-/Kontextvariable angeben, die mehrere Fälle zusammenfassen (hier: „idsch“ als Identifier für die jeweilige Schule). Über Options können weiter Schritte angefordert werden, wie bspw. Marginal Means oder auch die Varianzzerlegung (ICC) oder auch Grafiken.

Die Syntax für die oben angeklickten Befehle ist

```
## [Mixed Effects (Basic)]
require(lme4);
require(lmerTest);
require(rcompanion);
require(ggplot2);
require(reghelper);
require(emmeans);
require(visreg);
require(performance);

#Creating and summarizing the mixed model
MixedModel1=lmer(gpa_notena~hisklag1+as_mig2d+gym+geschlw+as_age+ (1|
idsch),REML=TRUE,data =daten06)
#Anova and Effect sizes
BSky.Anova.Table <-anova(MixedModel1)
BSkyFormat(as.data.frame(BSky.Anova.Table),singleTableOutputHeader = "ANOVA
Table")
```

#Calculating Effect sizes

```
BSky.effect.sizes <- with(BSky.Anova.Table,NumDF / DenDF * BSky.Anova.Table$"F value" / (1 + NumDF / DenDF * BSky.Anova.Table$"F value"))
names(BSky.effect.sizes) <-row.names(BSky.Anova.Table)
BSkyFormat(BSky.effect.sizes,singleTableOutputHeader = "Effect Sizes: Semi-partial R-squared")

BSkySummaryRes <- summary(MixedModel1)
#We store the results of the print into an object to suppress the plain text output from R
BSkySummaryRes <- BSkyprint.summary.merMod(BSkySummaryRes,correlation =TRUE)
```

Im Folgenden werden nur Teile des Outputs beschrieben. Zunächst zeigt sich eine Grafik, die über Fallzahlen und Zahl der Kontexteinheiten informiert, sowie über die Varianzverteilung wischen den Ebenen (idsch ist hier die Kontextvariable für die Einzelschulen).

Random Effects

| Groups | Name | Variance | Std.Dev. |
|---|---|---|---|
| idsch | (Intercept) | 0.05 | 0.224 |
| Residual | | 0.469 | 0.685 |

Number of obs: 24939, groups: idsch, 272

Tab. 13: Varianzkomponente des Mehrebenenmodells

Wiederum wird – auf Anforderung – auch der ICC ausgegeben, der sich gegenüber dem Leermodell verändert hat (hier nicht dargestellt).

Die Tabelle mit den festen Effekte („Fixed Effects“) zeigen die geschätzten Koeffizienten für die unabhängigen Variablen.

*Fixed Effects*

| | Estimate | Std. Error | df | t value | Pr(>\|t\|) |
|---|---|---|---|---|---|
| **(Intercept)** | 1.907 | 0.038 | 6465.799 | 49.698 | 0 *** |
| **hisklag1 HISEI** | 0.121 | 0.011 | 24838.3 | 11.387 | 0.000 *** |
| **unterstes Quartil** | | | | | |
| **as_mig2d** | 0.023 | 0.013 | 24563.78 | 1.737 | 0.082 . |
| **gym** | -0.315 | 0.037 | 244.114 | -8.544 | 0.000 *** |
| **geschlwn** | -0.139 | 0.009 | 24751.12 | -15.879 | 0.000 *** |
| **as_age** | 0.087 | 0.003 | 24891.03 | 34.511 | 0.000 *** |

*Note.* Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Effect Sizes: Semi-partial R-squared

| hisklag1 | as_mig2d | gym | geschlw | as_age |
|---|---|---|---|---|
| 0.005 | 0 | 0.23 | 0.01 | 0.046 |

Tab. 14: Ausgabe fester Effekte und Effektgrößen im Mehrebenenmodell

Angefordert werden können über das BSS-Menü unter Options bspw. auch noch estimate marginal means für die unabhängigen Variablen.

## V.5 (Explorative) Faktorenanalysen

Über das Menü von BSS können auch explorative Faktorenanalysen und Hauptkomponentenanalysen aufgerufen werden. Ohne hier ausführlich auf den Unterschied einzugehen (vgl. hierzu Bühner 2004, S. 154ff.), kann vereinfacht formuliert werden, dass eine Hauptkomponentenanalyse versucht, möglichst alle Varianz über Hauptkomponenten zu erklären, während die Faktorenanalyse darauf zielt, die Korrelation bzw. die Kovarianz der Variablen zu erklären (Noak 2007, S. 25).

Für die Durchführung bietet sich die Nutzung von BlueSky Statistics mit weiteren R-Pakete an. Exemplarisch soll dieses an einer Faktorenanalyse vorgestellt werden, für die BlueSky Statistics auf das R-Basis-Paket „stats“ und das Paket „GPArotation“ zurückgreift.

Es gibt es in R vielfältige Möglichkeit, Daten aus einem Datensatz für die Analysen auszuwählen, aber als eine pragmatischer Zugang kann für Faktorenanalysen folgende Vorgehensweise vorgeschlagen werden: Falls die Faktorscores nicht direkt im Arbeitsdatensatz benötigt werden, bietet es sich in R an, mit der Subsetfunktion die Items für die Faktorenanalyse in einen eigenen Teildatensatz auszuwählen und dann – bei Bedarf – auch mit diesen eine Z-Standardisierung vorzunehmen. Beides kann mit BSS mit dem Syntaxfenster erfolgen:

Mit „Data > Subset Dataset“ öffnet sich das folgende Unterfenster, in dem die gewünschten Variablen ausgewählt werden können.

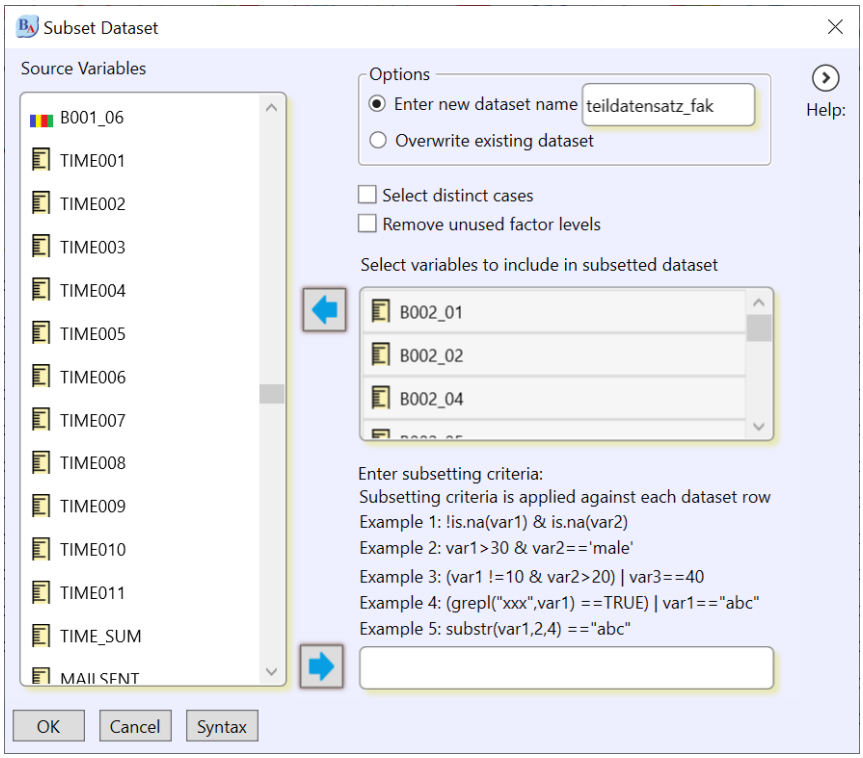



Abb. 45: Menüfenster zur Erstellung eines Teildatensatzes

Anschließend findet sich im Datenfenster ein neuer Reiter mit den entsprechenden Teildatensatz, mit dem die Faktorenanalyse und auch alle gewünschten Informationen leicht erzeugt werden können.

Falls die Arbeit mit z-standardisierten Variablen angestrebt ist kann dann noch weiter über “Data > Standardize Variable(s)” oder den farbige Menüpunkt „Standardize“ kommt man zu folgendem Fenster:

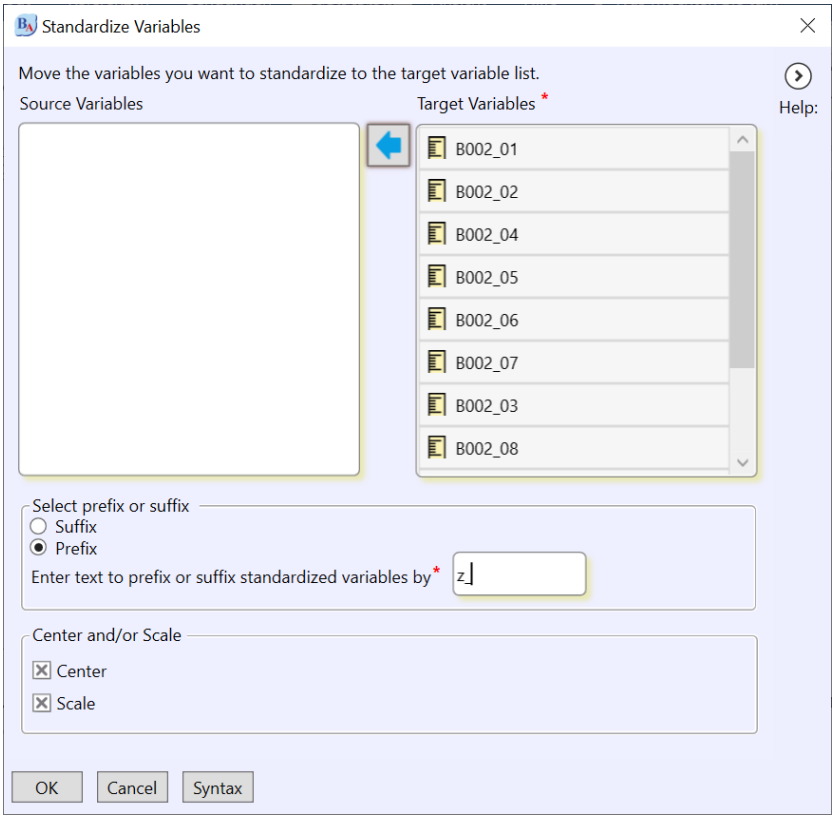



Abb. 46: Menüfenster zur (Z-)Standardisierung von Variablen

Wenn die Optionen „Center“ und „Scale“ angeklickt werden, werden die nach links geklickten Variablen z-standardisiert, dabei werden neue Variablen mit einem entsprechenden Vor- oder Nachsatz im Datensatz angelegt (Mittelwert 0, Standardabweichung =1). Für die weitere Faktorenanalyse, wenn sie mit den z-standardisierten Variablen durchgeführt werden soll, sollte für das nachfolgend beschriebene Vorgehen dann die unstandardisierten Variablen gelöscht werden. Das ist beispielsweise in der Variablenansicht des entsprechenden Datensatzes mit der rechten Maustaste (für jede Variable einzeln) möglich) oder mit der Subsetfunktion ein weiterer Teildatensatz mit nur den z-standardisierten Variablen angelegt werden.

Wenn nun die Daten entsprechend vorliegen, sind die folgenden Schritte der Faktorenanalyse zu gehen (vgl. bspw. Böhner 2004; Backhaus et al. 2016):

*Schritt 1*: Zur Prüfung der Datenqualität für die Faktorenanalyse bietet BSS keine Möglichkeit, die an, daher ist es sinnvoll, entsprechende Kennwerte wie das Kaiser-Maier-Olkin-Kriterium (KMO) mit andere R-Paketen anzufordern. Dies geht u.a. über das R-Paket *psych*, das hier auch im Weiteren für die Faktorenanalyse genutzt wird (und auch eigene Optionen für Faktoranalysen bereit hält). Hier muss der Befehl *KMO()* auf den Teildatensatz mit den standardisierten oder den unstandardisierten Variablen angewendet werden – daher sollten nicht beide in dem Datensatz vorhanden sein

```
#Laden des Pakets und Durchführung
require(psych)
kmo_data <- KMO(teildatensatz_fak)
```

*Schritt 2*: Auswahl der Zahl der Faktoren: Bei der eigentlichen Faktorenanalyse besteht in BSS besteht wie in SPSS die Möglichkeit die Zahl der Faktoren vorzugeben oder den Computer die Auswahl anhand des Eigenwertekriteriums (>=1) zu überlassen (und sich mit einem Screeplot auch das Knee-Kriterium anzuschauen). Da das Eigenwertekriterium auch eher als Anhaltspunkt denn als Determinate verstanden werden sollte und eigentlich auch nur für Hauptkomponentenanalysen sinnvoll ist, ist es ratsam, verschiedene Kriterien für die Auswahl der Zahl der Faktoren heranzuziehen (vgl. Bühner 2004; auch Backhaus et al. 2016). So bietet das eben genannte Paket *psych* weitere Möglichkeiten der die Zahl der auswählenden Faktoren einzugrenzen, was hier vorab genutzt wird. Konkret wird zur Entscheidungsfindung sowohl die Parallelanalyse sowie das VSS- als auch MAP-Informationskriterien (vgl. Luhmann 2011, Bühner 2004)

- Paralellanalyse

Sinnvoll ist es, dieselbe Art der Faktorenanalyse bzw. die Hauptkomponentenanlyse zu verwenden, die auch für die eigentliche Faktorenanalyse geplant ist (Hautkomponentenanalyse, Hauptfaktoranalyse (siehe Punkt C).

```
#Laden des Pakets und Durchführung Parallelanalyse
require(psych)
fa.parallel(teildaten_zfa, fa="fa")
```

## R-Syntax

```
fa.parallel(fa_items, fa="fa")
```

```
Parallel analysis suggests that the number of factors =  3  and the number of components =  NA
Call: fa.parallel(x = fa_items, fa = "fa")
Parallel analysis suggests that the number of factors =  3  and the number of components =  NA

 Eigen Values of

 eigen values of factors
 [1]  2.86  1.12  0.65  0.03 -0.07 -0.14 -0.21 -0.27 -0.34 -0.35 -0.41

 eigen values of simulated factors
 [1]  0.30  0.12  0.09  0.06  0.04  0.01 -0.01 -0.04 -0.06 -0.09 -0.13

 eigen values of components
 [1] 3.56 1.88 1.37 0.82 0.70 0.65 0.56 0.48 0.34 0.33 0.29

 eigen values of simulated components
[1] NA
```

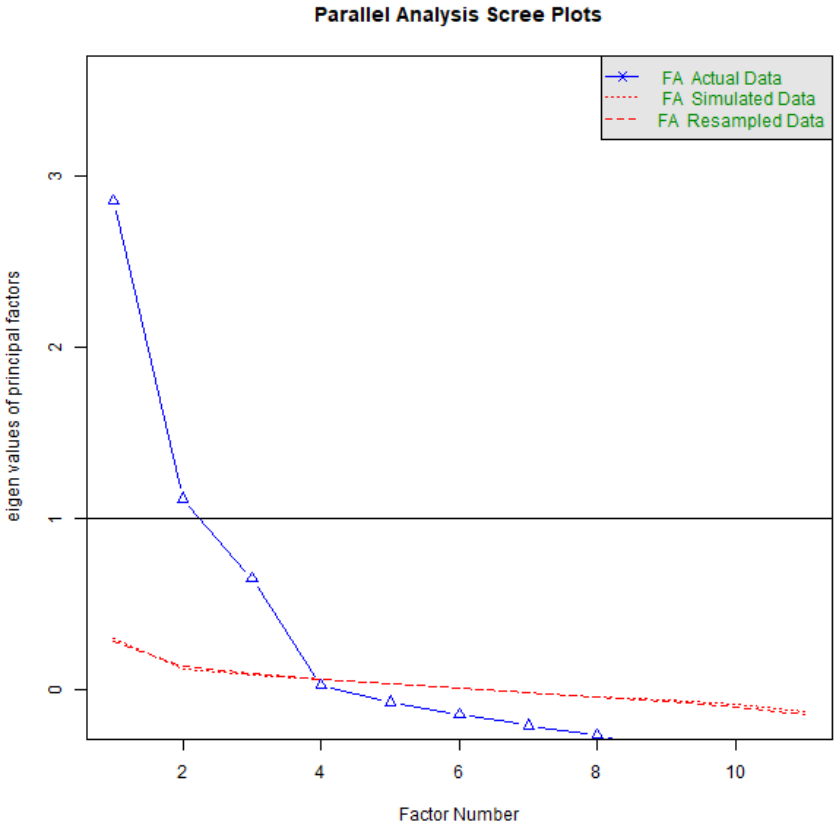



Abb. 47: Screeplot zur Parallelanalyse

c) VSS und MAP

Die VSS-function des Pakets „psych“ vergleicht den Fit einer Reihe von Faktoranalysen mit dender Ladungsmatrix, die vereinfacht wurde indem alle ausser de c höchsten Ladungen per item entfernt wurden, wobei c ein Maß der Faktorkomplexität ist (Revelle & Rocklin (1979). Mitausgeben wird das von Bühner (2004) sehr empfohlene MAP-Kriterium (Minimum Absolute Partial correlation) von Velicer.

#Aufruf

`require(psych)` #wenn nicht schon aufgerufen

`VSS(teildatensatz)`

```
Very Simple Structure
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = plot, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of 0.67  with  4  factors
VSS complexity 2 achieves a maximimum of 0.69  with  3  factors

The Velicer MAP achieves a minimum of 0.05  with  3  factors
BIC achieves a minimum of  NA  with  4  factors
Sample Size adjusted BIC achieves a minimum of  NA  with  5  factors
```

R gibt auch eine Grafik dazu aus (hier nicht dargestellt).

*Schritt 3*: Nach Abwägung der Kriterien und Festlegung der Zahl der Faktoren kann nun über BSS die Faktorenanalyse getstartet werden (kann sie natürlich auch schon vorher, in BSS ist ein einfacher Screeplot als auch das Eigenwertekriterium implementiert und kann über das Menü in BSS angefordert werden). Alternativ – was hier nicht demonstriert wird – kann dann über die Syntax auch eine Faktorenanalyse mit dem Paket „psych“ durchgeführt werden.

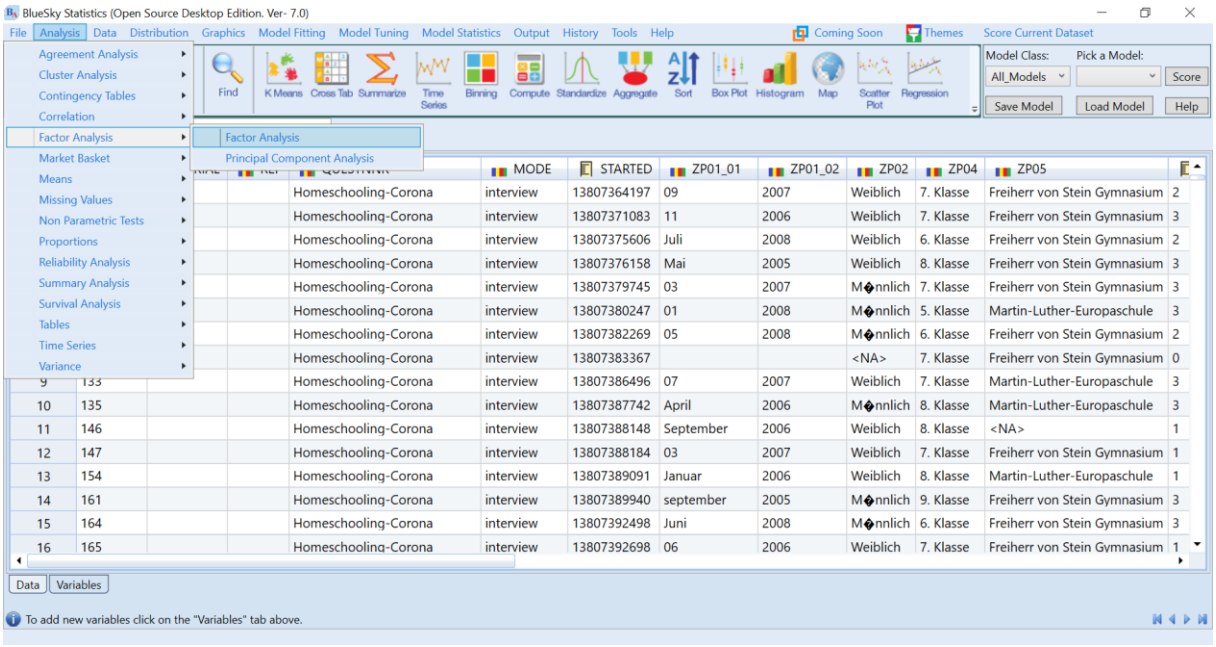



Abb. 48: Aufrufen der Faktorenanalyse in BSS

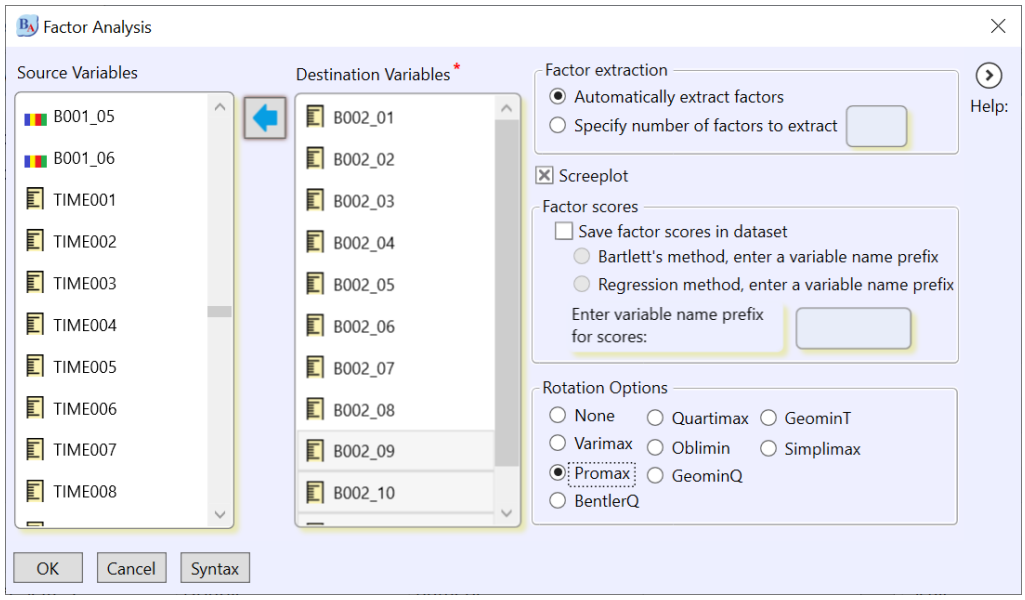



Abb. 49: Menüfenster zur Spezifizierung einer Faktorenanalyse

Ausgewählt weiter werden kann eine Vielzahl von Rotationsmethoden, ein Klick auf die Option Screeplot fordert das entsprechende Diagramm auch noch mal an.

Die von BSS erzeugte Befehlssyntax ist

```
## [Factor Analysis] #Laden der notwendigen Pakete
require(stats);
require(GPArotation);

#Run the factor analysis
BSkyFARes <-BSkyFactorAnalysis(vars=c('B002_01','B002_02','B002_03',
'B002_04','B002_05','B002_06','B002_07','B002_08','B002_09','B002_10','B002
_11'),autoextraction =TRUE,screeplot =TRUE,rotation="promax",saveScores
=FALSE,dataset="daten08")
#Display the results in the output grid
BSkyFormat(BSkyFARes)
#Refresh the dataset in the data grid to show the factor scores
BSkyLoadRefreshDataframe(daten08,FALSE)
```

In der Ergebnisdarstellung werden die unrotierten als auch die rotierten Kommunalitäten ausgegeben

Uniqueness (un-rotated)

| B002_01 | B002_02 | B002_03 | B002_04 | B002_05 | B002_06 | B002_07 | B002_08 | B002_09 | B002_10 | B002_11 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.567 | 0.346 | 0.801 | 0.779 | 0.218 | 0.442 | 0.387 | 0.643 | 0.276 | 0.376 | 0.722 |

Uniqueness ( promax rotation)

| B002_01 | B002_02 | B002_03 | B002_04 | B002_05 | B002_06 | B002_07 | B002_08 | B002_09 | B002_10 | B002_11 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.567 | 0.346 | 0.801 | 0.779 | 0.218 | 0.442 | 0.387 | 0.643 | 0.276 | 0.376 | 0.722 |

Zunächst wird die rotierte Faktormatrix angezeigt:

Rotated Loadings ( promax rotation)

| | Factor1 | Factor2 | Factor3 |
|---|---|---|---|
| B002_01 | 0.518 | | -0.284 |
| B002_02 | | | 0.812 |
| B002_03 | -0.229 | 0.418 | |
| B002_04 | 0.108 | | 0.484 |
| B002_05 | | 0.922 | |
| B002_06 | | 0.75 | |
| B002_07 | | | 0.826 |
| B002_08 | | 0.481 | -0.182 |
| B002_09 | 0.833 | | |
| B002_10 | 0.799 | | |
| B002_11 | 0.509 | 0.119 | |

Factors ( promax rotation)

| | Factor1 | Factor2 | Factor3 |
|---|---|---|---|
| SS loadings | 1.93 | 1.845 | 1.724 |
| Proportion Var | 0.175 | 0.168 | 0.157 |
| Cumulative Var | 0.175 | 0.343 | 0.5 |

Ausgeben wird zudem die Korrelation der Faktoren.

Factor Correlations

| | Factor1 | Factor2 | Factor3 |
|---|---|---|---|
| Factor1 | 1 | 0.281 | 0.517 |
| Factor2 | 0.281 | 1 | 0.288 |
| Factor3 | 0.517 | 0.288 | 1 |

Tab. 14: Ausgabe der Faktorenanalyse

```
Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 150.66 on 25 degrees of freedom.
The p-value is 6.36e-20
```

### Schritt 4: Testung der Reliabilität

In der Regel wird nach der erfolgten Faktorenanalyse die Reliabilität der gefundenen Lösung getestet, in der klassischen Testtheorie meistens über die Ermittlung des Cronbachs alpha.

Dies kann in BSS über „Analysis > Reliabilty > Cronbach's Alpha“ aufgerufen werden. Es erscheint folgendes Fenster:

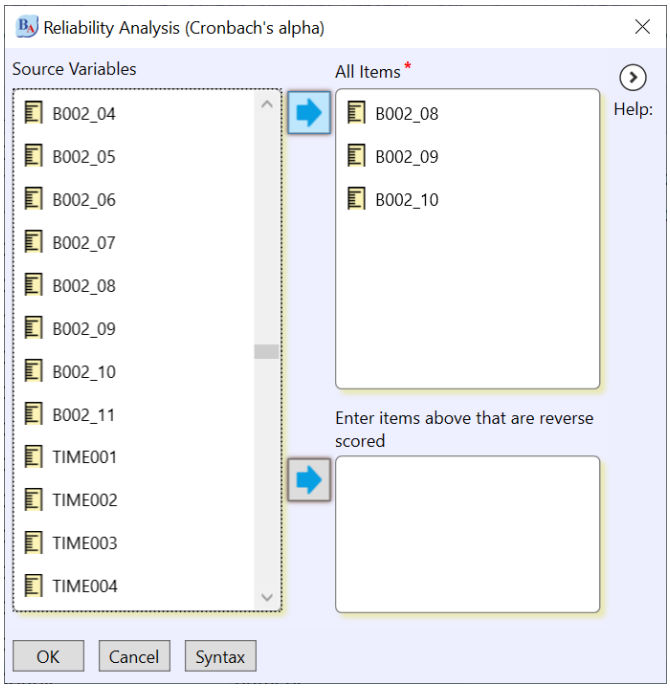



Abb. 50: Menüfenster Durchführung einer Reliablitätsanalyse

Ein nettes Feature ist das untere Fenster, in dem BSS mitgeteilt werden kann, wenn ein Item/mehrere Items „umgekehrt“ ausgerichtet ist/sind, was Rekodierungsarbeit erspart. Alternativ zum Cronbachs alpha gibt es die Option „Analysis > Reliabilty > McDonalds’s Omega“.

# VI. Weitere Analysen auf der Basis der R-Syntax

Um exemplarisch die Potentiale anzudeuten, die in der Verwendung von R und seinen Paketen stecken, die weit über die Möglichkeiten des Menüs von BlueSky Statistics hinausgehen, werden im Folgenden noch exemplarische Analysen vorgestellt, die mit entsprechenden R Paketen über die Syntax von BSS (oder R bzw. RStudio) durchgeführt werden können.

## VI.1 Konfirmatorische Faktoranalyse mit dem Paket lavaan

Konfirmatorische Faktorenanalysen dienen im Gegensatz zur exploratorischen Faktorenanalyse zur Prüfung einer angenommenen Faktorstruktur (vgl. Backhaus et al. 2016, S. 589ff.). Im eigenen Ausblick kündigen die Verantwortlichen von BSS an, dass in zukünftigen Versionen von BSS auch Strukturgleichungsmodelle mit lavaan implementiert werden (Coming soon im Menüfenster, https://www.BlueSkyStatistics.com/Articles.asp?ID=298). Aktuell muss – wenn entsprechende Analyse mit dem Paket lavaan geplant sind – das Paket installiert und die entsprechende Syntax über das Syntaxfenster eingegeben werden. Lavaan eignet sich auch, um konfirmatorische Faktorenanalysen mit einem oder mehreren Faktoren durchzuführen oder auch um latente Wachstumskurvenmodelle durchzuführen (vgl. Rosseel 2020).

Ein Modell mit zwei latenten Faktoren wird wie folgt spezifiziert (natürlich kann auch nur ein Latente Faktor spezifiziert werden):

```
#Laden des Pakets
require(lavaan)
```

*# Das Model formulieren*

```
model1 <- '
faktor1 =~ f01_a + f01_b + f01_c
faktor2 =~ f01_j + f01_k + f01_l
'
```

(wichtig ist das einfache Anführungszeichen am Ende und am Anfang!)

#faktor1 und faktor2 … sind keine erhobenen Variablen, sondern werden als latente Variablen durch die erhobenen Variablen f01_a, f01_b … „konstruiert". Die Namen der latenten Variablen sind frei wählbar)

# Das Modell ausführen

```
fit <- cfa(model1, data=datensatz2)
```

# Anzeige des Outputs

```
summary(fit, fit.measures=TRUE)
```

Über das Paket lavaan sind auch Strukturgleichungsmodelle oder latente Wachstumskurvenmodelle spezifizierbar. Einführungen in das Paket, ausführliche Dokumentation sowie Anwendungsbeispiele finden sich auf der Homepage https://lavaan.ugent.be/.

## VI.2 LCA/Latente Klassenanalyse mit dem Paket poLCA

BlueSky Statistics enthält auch mehrere Optionen zur Durchführung von Clusteranalyen unter „Analysis > Cluster Analysis". Inhaltlich verwandt können Cluster oder Klassen anhand eines statistischen Modells auch mit latenten Klassenanalysen (die das Ziel haben, aus vielen Fällen Typen oder Gruppen „ähnlicher" Fälle zu identifizieren) (vgl. als Einführung bspw. Bacher & Vermunt 2010; Rost 2004)). Diese sogenannten LCA lässt sich mit R über verschiedene Pakete ausführen. Bspw. kann. das Paket poLCA verwendet werden. Zunächst muss auch hier wieder das Paket installiert und dann aufgerufen werden. Im Folgenden findet sich ein Beispiel, wie eine LCA mit 2, eine mit 3 und eine mit 4 latenten Klassen spezifiziert wird.

Ausgegangen wird von einem Datensatz mit Namen „datensatz03"

#Laden des Pakets

```
require(poLCA)
```

#Fehlende Werte löschen (durch Erstellung eines reduzierten Datensatzes)

```
daten03 <-na.omit(datensatz03)
```

#Variablen für die LCA zusammenstellen (ordinale und nominale Skalen möglich, Skalierung der Variablen muss in jedem Fall mit 1 beginnen)

```
f <- cbind(z01a,z01b,z01c,z01d, z01e,z01f,z01g)~1
```

# LCA durchführen (hier für 2, für 3 und für 4 Klassen, Modelle können verglichen werden)

```
z01.lca2 <- poLCA(f,daten03, graphs=TRUE, nclass=2,maxiter=5000,nrep=40)
z01.lca3 <- poLCA(f, daten03, graphs=TRUE, nclass=3,maxiter=5000,nrep=40)
z01.lca4 <- poLCA(f, daten03, graphs=TRUE, nclass=4,maxiter=5000,nrep=40)
```

Insbesondere das AIC und das BIC (niedrigster Wert) werden zur Auswahl der bestmöglichen Lösung herangezogen. Ausführliche Beschreibung sowie Anwendungsbeispiele finden sich u.a. bei den Paketautoren Linzer & Lewis (2011).

## Quellen


Homepage von BlueSky Statistics: https://www.BlueSky Statistics.com/Default.asp

Akinwande, O., H.G Dikko, Samson Agboola (2015). Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis, Open Journal of Statistics, 5 (7), 754-767.

Bacher, J. & Vermunt, J.K. (2010): Analyse latenter Klassen. In C. Wolf & H. Best (Hrsg.), Handbuch der sozialwissenschaftlichen Datenanalyse (S. 553-574). Wiesbaden: VS.

Backhaus, K., Erichson, B., Plinke, W. & Weiber, R. (2016). Multivariate Analysemethoden. 14. Aufl., Berlin: Springer.

Bühner, M. (2004). Einführung in die Test- und Fragebogenkonstruktion. München: Pearson Studium – Psychologie.

Große Schlarmann, J. & Galatsch, M. (2014). Regressionsmodelle für ordinale Zielvariablen. Medizinische Informatik, Biometrie und Epidemiologie, 10(1), 1-9.

Janssen, J. & Laatz, W. (2017). Statistische Datenanalyse mit SPSS. 9. Aufl., Berlin: Springer.

Lamprianou, I. (2019). Applying the Rasch Model in Social Sciences Using R and BlueSky Statistics. London: CRC Press.

Langer, W. (2010). Mehrebenenanalyse mit Querschnittsdaten. In C. Wolf & H. Best (Hrsg.), Handbuch der sozialwissenschaftlichen Datenanalyse (S. 741-774). Wiesbaden: VS.

Leeper, T. J. (2018). Interpreting Regression Results using Average Marginal Effects with R's margins. Verfügbar unter https://cran.r-project.org/web/packages/margins/vignettes/TechnicalDetails.pdf [letzter Zugriff 15.9.2020].

Linzer, D. A. & Lewis, J. B (2011). poLCA – An R Package for Polytomous Variable Latent Class Analysis. *Journal of Statistical Software* 42 (10), unter https://www.jstatsoft.org/article/view/v042i10

Luhmann, M. (2015). R für Einsteiger. Einführung in die Statistiksoftware für die Sozialwissenschaften. 4., vollständig überarbeitete Auflage, Weinheim: Beltz.

Muenchen, R. A. (2020a). A Comparative Review of the BlueSky Statistics GUI for R, updated 8/3/2020 http://r4stats.com/articles/software-reviews/bluesky/

Muenchen, R. A. (2020b). BlueSky Statistics User's Guide. https://onedrive.live.com/?authkey=%21ABRSHNb7jGT219U&cid=6C40810986A4E53F&id=6C40810986A4E53F%21221597&parId=6C40810986A4E53F%2167337&o=OneUp

Noak, M. (2007). Faktorenanalyse. Online Verfügbar unter https://www.uni-due.de/imperia/md/content/soziologie/stein/faktorenanalyse.pdf [letzter Zugriff 15.9.2020]

Rosseel, Y (2020). The lavaan tutorial. Verfügbar unter https://lavaan.ugent.be/tutorial/tutorial.pdf [letzter Zugriff 15.9.2020].

Rost, J. (2004). Lehrbuch der Testtheorie – Testkonstruktion. 2. Aufl., Bern: Huber.


## Schlusswort

Dieses Paper ist Work in Progress – Hinweise, Kritik und Ergänzungswünsche werden gerne entgegengenommen. Angedacht ist, bei weiterem Ausbau dieses Textes die Anwendung ausgewählter Verfahren ausführlicher zu beschreiben (wie ansatzweise bei der linearen Regression im folgenden Anhang).


*Kontakt: Ivo Züchner, Philipps-Universität Marburg, Fachbereich 21, Institut für Erziehungswissenschaft, zuechner@staff.uni-marburg.de*

# Anhang 1: Ansprechen/Aufrufen von Variablen über den R-Code

In R können Variablen auf verschiedene Weisen angesprochen werden. Der „klassische“ Zugang zu Variablen erfolgt mit der Verbindung von

Name des Datensatzes – Dollarzeichen – Kurzname der Variablen, wie z.B.

```
datensatz1$var1
```

So wird mit dem Befehl

```
table(daten02$geschlw)
```

eine einfache Häufigkeitsausgabe für die Variable geschlw aus dem Datensatz daten02 erzeugt. Umfangreichere Befehle haben darüber hinaus oft die Option, erst das „Modell“ zu formulieren in diese Modellformulierung mit dem Zusatz `data=datensatz1` den Datensatz festzulegen wie z.B. `LinMod1 <- lm(skala1 ~ geschlw + alter, data = daten02)`

# Anhang 2: Effektstärken über R-Syntax berechnen

In den beschreibenden Statistiken von BSS sind keine Effektgrößen voreingestellt bzw. aufrufbar, daher hier exemplarische Hinweise, wie über die R Syntax eine Ausgabe von Effektstärken zu erreichen ist – wie immer gibt es verschiedene R Pakete, die dieses ermöglichen.

- **Kreuztabellen**: Phi (2*2 Tabelle) oder Cramers V (2*x Tabelle)

#zunächst Paket *DescTools* laden (ist schon mit BlueSky Statistics installiert)

```
require(DescTools)
```

# Tabelle mit zwei Variablen speichern

```
table1 <- table(datensatz2$ZP02, datensatz2$ZP04)
```

#Cramers V berechnen

```
CramerV(table1, conf.level=0.95)
```

- **Mittelwertsvergleich von 2 Gruppen**: Cohens d

#zunächst Paket lsr (installieren und) laden

```
require(lsr)
cohensD(mathe ~ teacher, data = daten04)
```

(Mathe ist die metrische abhängige Variable, teacher die zweistufige Gruppenvariable)

- **Mittelwertsvergleich von mehr als zwei Gruppen** (aus einer Varianzanalyse): eta²

#zunächst Paket *lsr* (installieren und) laden

```
require(lsr)
```

# zunächst Varianzanalyse mit Befehl „aov“ durchführen

```
anova1 <- aov(mathe ~ gruppenvariable, data=daten04)
```

# Darstellung der ANOVA-Tabelle

```
summary(anova1)
```

# Ausgabe des Eta²

```
etaSquared(anova1)
```

Dies sind nur Beispiele, in R gibt es über viele Pakete weitere Möglichkeiten, diese oder andere Effektstärken zu ermitteln.