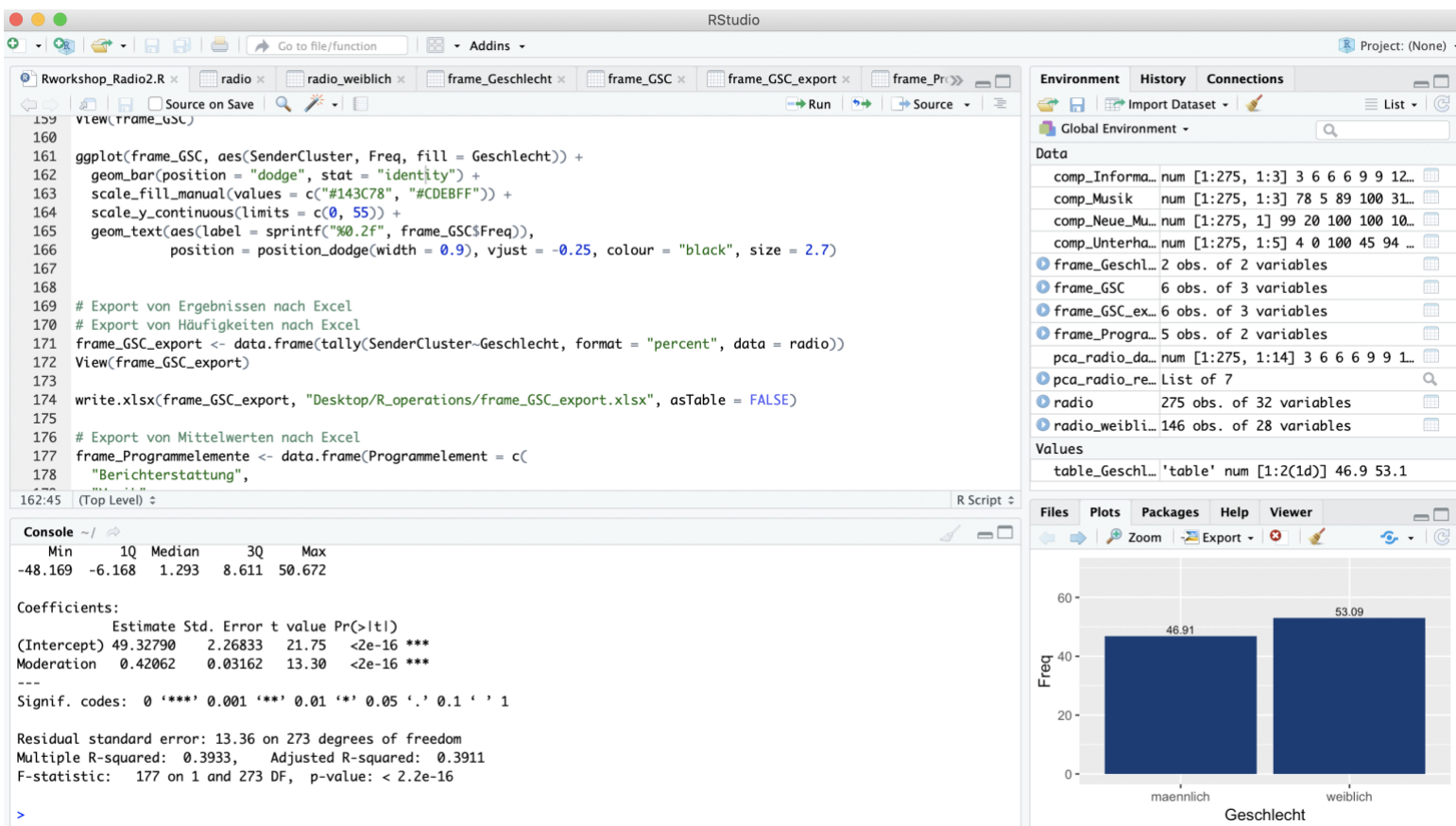


Hendrik Godbersen

Angewandte Statistik & R

Das Buch zum Kompaktkurs mit R-Befehlsblatt für
alle gängigen Analyseverfahren, Datensatz zum
Üben & Video-Tutorials



Hendrik Godbersen

Angewandte Statistik & R

© Prof. Dr. Hendrik Godbersen 2020

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Autors. Dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Nicht genehmigte Vervielfältigungen und Verbreitungen des Werks werden straf- und zivilrechtlich verfolgt.

Der Autor geht davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Der Autor übernimmt, ausdrücklich oder implizit, keine Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Inhaltsverzeichnis

1 Einleitung.....	4
2 Installation & Grundstruktur von R Studio.....	6
2.1 Installation von R, R Studio & relevanten Paketen	6
2.2 Grundstruktur von R Studio	8
3 R-Befehle & Übungsdaten.....	10
3.1 R-Befehle.....	10
3.2 Übungsdaten.....	13
4 Datenmanagement.....	16
4.1 Datenvorbereitung & Laden von Daten.....	16
4.2 Daten bearbeiten in R	18
5 Deskriptive Statistik.....	22
5.1 Skalenniveaus.....	23
5.2 Häufigkeiten.....	24
5.3 Lageparameter & Streuungsmaße	26
6 Inferenzstatistische Verfahren.....	32
6.1 Überblick & Systematisierung der Verfahren	32
6.2 Chi-Quadrat-Test.....	34
6.3 t-Test	37
6.4 ANOVA	39
6.5 Shapiro-Wilk-Test.....	40
6.6 Wilcoxon-Test	42
6.7 Korrelation	44
6.8 Regression	46
6.9 Hauptkomponentenanalyse	50
7 Graphiken & Export von Ergebnissen nach Excel	56
7.1 Graphiken in R.....	56
7.2 Export von Ergebnissen nach Excel.....	58

1 Einleitung

Dieser Kurs vermittelt das Wissen und die Fähigkeiten:

- die gängigen Verfahren der deskriptiven & inferenzstatistischen Auswertung zu verstehen und anzuwenden.
- mit R zu arbeiten und die für statistische Auswertungen notwendigen Befehle anzuwenden.

Somit sollten Sie nach diesem Kurs, **ein empirisches Projekt mit den „gängigen“ statistischen Verfahren in R auszuwerten**.

Um diesen Kurs durchzuführen, benötigen Sie neben diesem Buch die folgenden Unterlagen und Dateien, die Sie auf www.godbersen.online erhalten:

- das **Befehlssammlung mit allen notwendigen R-Befehlen**, um ein empirisches Projekt mit den „gängigen“ Verfahren auszuwerten
- den **Datensatz**, mit dem die Übungsaufgaben zu diesem Kurs realisiert werden

Es wird empfohlen, das R-Befehlsblatt auszudrucken, damit die Übungsaufgaben so einfach wie möglich umgesetzt werden können und Sie sich auf die Anwendung der statistischen Verfahren bestmöglich konzentrieren können.

Ferner sind Video-Tutorials Teil des Kurses, die an den entsprechenden Stellen in jedem Kapitel dieses Buchs verlinkt sind. Sie benötigen also eine Internetverbindung, wenn Sie diesen Kurs durchführen. Es wird nachdrücklich dazu geraten, die Video-Tutorials als erstes anzuschauen und den Text in diesem Buch eher als Nachschlagewerk zu betrachten.

Hier gelangen Sie zum **Video-Tutorial** des ersten Kapitels:



Kern des Kurses sind die statistischen Verfahren in den **Kapiteln fünf und sechs**. Jedes einzelne Unter-Kapitel behandelt ein Auswertungsverfahren und gliedert sich jeweils in drei Abschnitte:

- 1) Erklärung der statistischen Verfahren
- 2) Übungsaufgabe mit einem Datensatz aus einem Forschungsprojekt
- 3) Umsetzung der statischen Analyse in R

Für den größt- und schnellstmöglichen Lernerfolg ist die **Herangehensweise** in diesem Kurs wichtig. Im Kern sollten Sie sich die Fragen stellen, **wann wende ich welches Verfahren an und wie interpretiere ich die Ergebnisse**. Die Frage, wie programmiere ich mit R, sollte nicht im Vordergrund stehen. Die R-Programmierung werden Sie „nebenbei“ lernen.

Darüber hinaus ist es wichtig, dass Sie den **Kurs Schritt für Schritt durchgehen**. Lassen Sie kein Kapitel aus und gehen Sie erst zum nächsten Kapitel über, wenn Sie das vorherige auch verstanden haben. Dies ist wichtig, da die einzelnen Kapitel aufeinander aufbauen. Wenn Sie den Kurs einmal beendet haben, können Sie das Buch, die Video-Tutorials und die Übungsaufgaben als Nachschlagewerk nutzen, z.B. wenn Sie ein bestimmtes Auswertungsverfahren in einem konkreten empirischen Projekt anwenden möchten.

Mit dieser Herangehensweise werden Sie (abhängig von Ihrer Lerngeschwindigkeit etwas schneller oder langsamer) **innerhalb von sieben Stunden solide Kenntnisse und Fähigkeiten in der anwendungsorientierten empirischen Forschung** erhalten. Ziel ist es, dass Sie ein **anspruchsvolles empirisches Projekt mit den gängigen statistischen Verfahren in R analysieren** können.

2 Installation & Grundstruktur von R Studio

In diesem Kapitel lernen Sie, wie Sie R, R Studio und alle notwendigen Zusatzpakete für statistische Analysen installieren. Ferner wird im zweiten Teil ein Überblick über den Aufbau von R Studio, der Benutzeroberfläche für Auswertungen gegeben.

2.1 Installation von R, R Studio & relevanten Paketen

Um Analysen mit R vornehmen zu können, benötigen Sie R, R Studio und einige Zusatzpakete. Im folgenden **Video-Tutorial** sind die notwendigen Installationsschritte beschrieben:

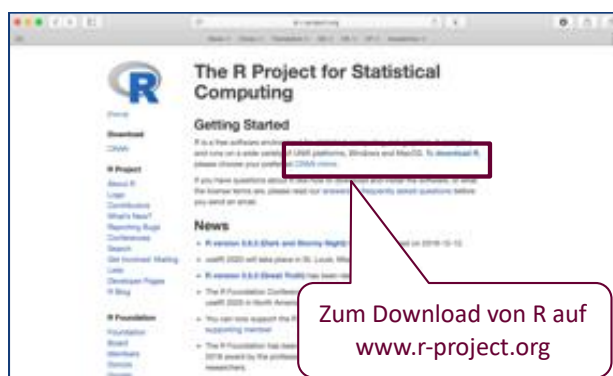


Im Folgenden sind die drei Installationsschritte noch einmal beschrieben:

1) **Download & Installation von R** (R ist das Basisprogramm)

- 1) Gehen sie zu: <https://www.r-project.org/>
- 2) Wählen Sie einen Mirror (Quelle zum Herunterladen) zum Download aus
 - Achten Sie darauf, dass Sie R passend zu Ihrem Betriebssystem wählen
- 3) Laden Sie R herunter
- 4) Installieren Sie R auf Ihrem Computer

In der folgenden Abbildung sehen Sie die Einstiegsseite www.r-project.org, über die Sie zum Download von R gelangen:



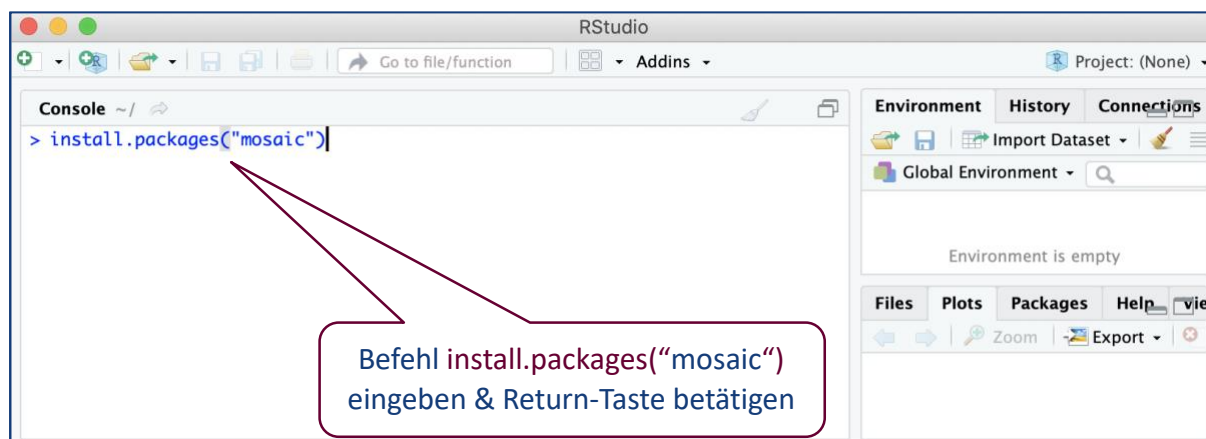
2) Download & Installation von R Studio (R Studio ist eine Entwicklungsumgebung für R und erleichtert die Bedienung)

- 1) Gehen Sie zu: <https://rstudio.com/products/rstudio/download/>
- 2) Laden Sie R Studio herunter
- 3) Installieren Sie R auf Ihrem Computer

3) Installieren Sie die Zusatzpakete in R Studio

- 1) Stellen Sie sicher, dass Ihr Computer mit dem Internet verbunden ist
- 2) Geben Sie in die Console folgenden Befehl ein: `install.packages("mosaic")`
- 3) Betätigen Sie die Return-Taste
- 4) Wiederholen Sie dieses Vorgehen für alle relevanten Pakete

In der folgenden Abbildung ist dargestellt, wo Sie in R Studio den Befehl `install.packages("")` eingeben müssen, um die notwendigen Zusatzpakete zu installieren:



Die folgenden Pakete benötigen Sie und müssen diese mit dem Befehl `install.packages("")` über die Console installieren:

- mosaic
- psych
- psy
- ggplot2
- openxlsx
- readxl

2.2 Grundstruktur von R Studio

In diesem Kapitel geht es um die Grundstruktur von R Studio. Schauen Sie sich dazu das folgende **Video** an:



R Studio ist in vier Bereiche eingeteilt.

Oben rechts unter Environment werden die geladenen Datensätze angezeigt. (Wie Datensätze geladen werden, wird in Kapitel 4.1 erklärt.)

Oben links wird das Script angezeigt. Im Script werden die Befehle, mit denen ein Datensatz analysiert wird, eingetragen und ausgeführt. Das Erstellen, Öffnen und Speichern von Scripten erfolgt wie bei anderen Anwendungen über das Haupt-Menü. Um einen Befehl auszuführen, muss der Run-Button oberhalb des Scripts betätigt werden; wichtig ist, dass die Zeilen im Script, in denen die auszuführenden Befehle stehen, markiert sind. Im Bereich oben links können Sie sich auch den Datensatz anzeigen lassen; beachten Sie aber, dass der Datensatz nur geladen ist und auch für die Auswertung verwendet werden kann, wenn er oben rechts unter Environment angezeigt wird.

Unten links befindet sich die Console. In dieser werden die Ergebnisse der statistischen Analysen angezeigt. Die Auswertungs-Befehle können auch hier eingegeben und ausgeführt werden; davon wird aber abgeraten, da die Befehle beim Schließen von R verloren gehen. Geben Sie die Befehle deshalb immer im Script (oben links) ein, das Sie speichern, wieder öffnen und dann auch wieder laufen lassen können.

Unten rechts werden unter Plots Graphiken angezeigt, die Sie mit R erstellen. (Das Erstellen von Graphiken wird in Kapitel 7.1 erklärt.)

Die folgende Abbildung gibt einen Überblick über die vier Felder in R Studio:

The image shows the R Studio interface with four main panes, each annotated with a callout box:

- Script Editor (Top Left):** Contains R code. An annotation points to the top-left corner: "Öffnen, Speichern etc. von Dateien erfolgt wie bei anderen Programmen über die Navigation". Another annotation points to the "Run" button: "Button, um Befehle auszuführen (Zeilen im Skript müssen markiert sein)".
- Environment (Top Right):** Shows loaded datasets and generated objects. An annotation points to it: "Geladener Datensatz & generierte Objekte".
- Console (Bottom Left):** Shows the results of commands. An annotation points to it: "Anzeige des Datensatzes (Achtung: Anzeige ≠ geladener Datensatz)". Another annotation points to the input area: "Console mit Ergebnisse (Befehlseingabe auch hier möglich; Befehle werden aber nicht gespeichert)".
- Plots (Bottom Right):** Shows generated graphics. An annotation points to it: "Generierte Graphiken".

3 R-Befehle & Übungsdaten

In diesem Kapitel werden die R-Befehlssammlung, die Sie auf www.godbersen.online herunterladen können, und die grundsätzliche Befehlsstruktur in R und insbesondere dem Zusatzpaket mosaic, welches das wesentliche Paket für die statistischen Auswertungen ist, erklärt. Ferner wird ein Überblick über die Übungsdaten, die Sie auch auf www.godbersen.online herunterladen können, und das dahinterliegende Forschungsprojekt gegeben.

3.1 R-Befehle

Die R-Befehlssammlung besteht aus drei Seiten:

- **Seite 1:** Alle notwendigen Befehle für die Auswertungen von “normalen” Forschungsprojekten
- **Seite 2:** Erstellen von Graphiken und Export von Ergebnissen nach Excel
- **Seite 3:** PLSPM (Partial least path square modelling) – über Standardverfahren hinausgehend und nicht relevant in diesem Kurs

Eine detailliertere Erklärung der R-Befehlssammlung finden Sie im folgenden **Video-Tutorial**:



Die wichtigste Seite (für diesen Kurs) ist die erste, da auf dieser Seite die R-Befehle aller statistischen Standard-Verfahren zu finden sind. Die folgende Abbildung gibt einen Überblick über die erste Seite der R-Befehlssammlung:

Installieren & Laden der (Zusatz-)Pakete

Bearbeiten von Variablen & Datensätzen

Häufigkeiten

Lageparameter & Streuungsmaße

Inferenzstatistische Verfahren

Hauptkomponentenanalyse & Cronbachs Alpha

Das oben abgebildete **erstes Befehlsblatt** gliedert sich in **sechs Bereiche**. Oben links finden sich unter „Get going“ die Befehle, die benötigt werden, um Zusatzpakete zu installieren und zu laden. Die Installation muss nur einmal vorgenommen werden. **Das Laden der Pakete (require(mosaic) etc.) muss jedes Mal, wenn R Studio gestartet wird, wieder vorgenommen werden.**

Im nächsten Abschnitt darunter („Recoding, computing, formatting variables & creating new datasets“) stehen die Befehle, um die Codes (Merkmalsausprägungen) von Variablen zu verändern, neue Variablen zu berechnen, das Format von Variablen (numerische und nicht numerische Merkmalsausprägungen) anzupassen und aus einem bestehenden Datensatz einen neuen Datensatz zu generieren.

In den nächsten beiden Abschnitten unten rechts auf dem ersten Befehlsblatt finden sich die Befehle zur Auswertung eines Datensatzes nach Häufigkeiten („Frequencies“) und die Bestimmung von Lageparametern (z.B. Arithmetische Mittelwerte) und Streuungsmaßen (z.B. Standardabweichung) („Median, mean, variance, standard deviation, maximum, minimum“).

Die rechte Seite des ersten Befehlsblatts ist in zwei Bereiche geteilt. Zuerst werden die Befehle für alle Standard-Verfahren der Inferenzstatistik aufgeführt („Chi²-Test“, „t-Test“ etc.). Daran anschließend werden unten links die Befehle für die Hauptkomponentenanalyse („Principal component analysis“) als strukturentdeckendes Verfahren und die Berechnung von

Cronbachs alpha, die im Zusammenhang mit der Hauptkomponentenanalyse durchgeführt wird („Cronbach’s alpha“), dargestellt.

Die **Grundstruktur der Befehle** im R-Zusatzpaket mosaic, welches das hauptsächliche Auswertungs-Tool ist, stellt sich, wie folgt, dar:

- `command(y~x, z)`

Als erstes wird der Befehl für das Auswertungsverfahren (Häufigkeit, arithmetischer Mittelwert, Chi²-Test, t-Test etc.) angegeben. Dann folgt eine öffnende Klammer. Die abhängige Variable wird angegeben. Nach dem Schlangenzeichen folgt die unabhängige Variable. Es kommt ein Komma und der Datensatz, aus dem R sich die Variablen ziehen soll, wird eingetragen. Der Befehl wird mit einer abschließenden Klammer beendet. Wenn es keine zwei Variablen oder keine abhängige Variable gibt, entfällt das y und der Befehl beginnt mit dem Schlagzeichen.

Ein Beispiel soll diese Struktur verdeutlichen. Es soll herausgefunden werden, wieviele Frauen und Männern, die in der Variable „Geschlecht“ erfasst wurden, im Datensatz „radio“ gegeben sind. Dazu wäre folgender Befehl notwendig (eine detailliertere Einführung in die Auswertung mit Häufigkeiten finden Sie in Kapitel 5.2):

- `tally(~Geschlecht, data = radio)`

Die Farben in der Befehlssammlung sind für die Anwendung von Bedeutung. Alle Buchstaben und Zeichen, die rot dargestellt sind, werden als fixe Befehle von R so gefordert und dürfen nicht verändert werden. Die blauen Wörter sind Variablen oder der auszuwertende Datensatz, die je nach ihrer individuellen Bezeichnung angepasst werden müssen.

Die soeben beschriebene Befehlsstruktur ist noch einmal in der folgenden Abbildung dargestellt:

R (mosaic) – Key Commands

```

Get online
install.packages("r")
require(r) <- starting mosaic, psych, pty, ggplot2, openssl, plotly (required for each)

Recording, converting, formatting variables & creating new datasets:
Standardize (new) variables:
dataset$variable[dataset$variable=="000"] <- "new_value"
→ Please note: categorical values require quotes (""), numerical values
Calculating new variables:
dataset$variable <- dataset$variable1 + dataset$variable2
Changing the variable format:
dataset$variable <- as.numeric(dataset$variable)
→ factor to numeric
(1) dataset$variable <- as.factor(dataset$variable)
(2) levels(dataset$variable) <- c("attribute1", "attribute2")
→ numeric to factor
Creating new datasets (to email, is not equal etc.):
new_dataset <- dataset[dataset$variable == "value"]
new_dataset <- dataset[dataset$variable != "value"]
new_dataset <- dataset[dataset$variable > value]

```

General command structure

`command(y~x, z)`

Frequencies

`tally(~variable1, data = dataset)`

`tally(~variable1, format = "percent", data = dataset)`

`tally(variable1~variable2, data = dataset)`

`tally(variable1~variable2, format = "percent", data = dataset)`

General command structure

`command(y~x, z)`

Exercises

`tally(~variable1, data = dataset)`

`tally(~variable1, format = "percent", data = dataset)`

`tally(variable1~variable2, data = dataset)`

`tally(variable1~variable2, format = "percent", data = dataset)`

Median, mean, variance, standard deviation, maximum, minimum

`median~variable, data = dataset` / `mean~variable, data = dataset`

`var~variable, data = dataset` / `sd~variable, data = dataset`

`max~variable, data = dataset` / `min~variable, data = dataset`

`round~variable, data = dataset`

`→ result: Min, Q1, median, Q3, Max, mean, standard deviation, n, missing values`

Hinweis zu den Befehlen

rot R-Befehl (darf nicht verändert werden)

blau Bezeichnungen von Variablen oder des Datensatzes, die individuell angepasst werden muss

y Abhängige Variable

x Unabhängige Variable

z Datensatz

3.2 Übungsdaten

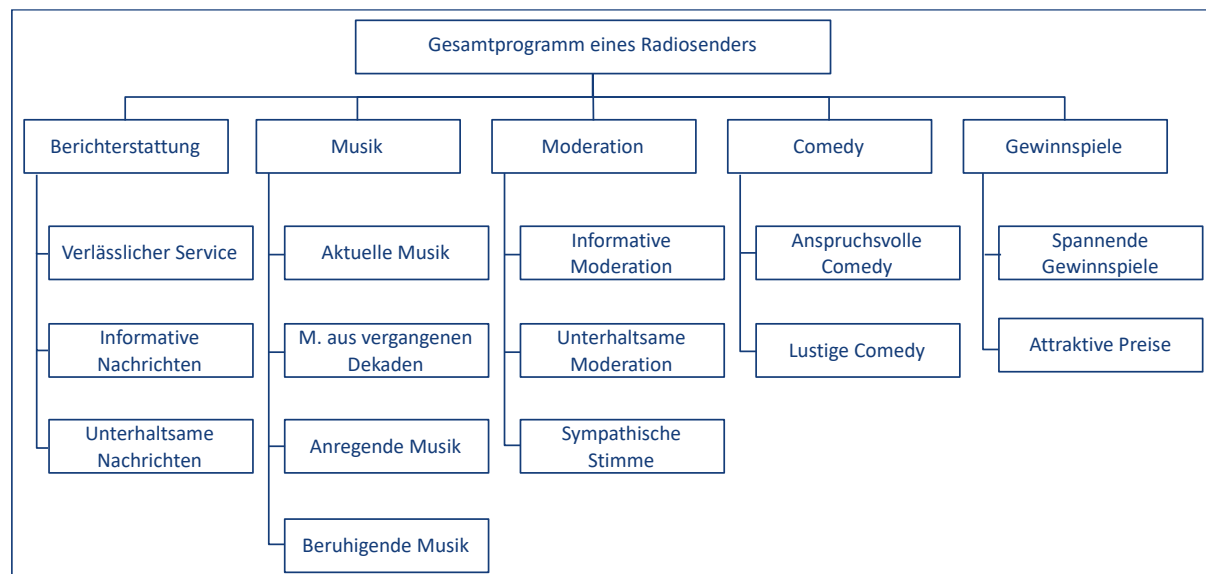
Im Rahmen dieses Kurses müssen Sie alle gängigen Auswertungsverfahren in Übungsaufgaben anwenden und so selbst eine statistische Auswertung vornehmen. Diese Auswertung erfolgt mit einem vereinfachten Datensatz aus einem Forschungsprojekt, das in folgendem Artikel veröffentlicht wurde:

- Godbersen, H. (2019): Hörererwartungen, Programmqualität und Optimierungspotenzial von musikbasierten Formatradios – Eine empirische Analyse mit der Means-End Theory of Complex Cognitive Structures, in: transfer – Zeitschrift für Kommunikation und Markenmanagement, Jg. 65, Nr. 3, S. 12-21

Zur näheren Erläuterung des Datensatzes und der dahinterliegenden inhaltlichen Überlegungen schauen Sie sich das folgende **Video-Tutorial** an:



Das o.a. Forschungsprojekt behandelt die Beurteilung von Formatradiosendern. Die Grundstruktur des Beurteilungsmodells ist in der folgenden Abbildung dargestellt:



Das Modell besteht aus drei Ebenen. An der Spitze steht die Qualität des Gesamtprogramms aus Hörsicht. Auf der darauffolgenden Ebene stehen die Programmelemente Berichterstattung, Musik, Moderation, Comedy und Gewinnspiele. Auf der untersten und konkretesten Ebene sind die Programmeigenschaften (z.B. aktuelle Musik, Musik aus vergangenen Dekaden, anregende und beruhigende Musik) zu finden. Die Variablen auf den genannten drei Ebenen wurden mit Continuous Rating Scales von 0 (nicht gut) bis 100 (sehr gut) erfasst. Daneben wurden weitere Variablen erfasst:

- Stammsender: in den letzten zwei Wochen am häufigsten gehörter Sender
- SenderCluster: neugebildete Variable mit den Merkmalsausprägungen Privat jung, Privat alt und SWR
- Geschlecht: mit den Merkmalsausprägungen männlich und weiblich
- Alter: angegeben in Jahren
- Altersgruppen: zwei Gruppen mit den Merkmalsausprägungen jung und alt
- Affinität zu aktueller Musik (Variablen-Bezeichnung: Aktuelle_Musik_A): zwei Gruppen mit den Merkmalsausprägungen „eher affin zu aktueller Musik“ und „eher nicht affin zu aktueller Musik“

Eine Gesamtübersicht zu den Variablen des Datensatzes finden Sie in der folgenden Abbildung, wobei die Variablen-Bezeichnungen rot geschrieben sind:

- **Stammsender**
 - Messung: in den letzten 2 Wochen am häufigster gehörter Sender
 - **SenderCluster**
 - Messung: neu gebildete Variable mit drei Ausprägungen
 - Merkmalsausprägungen: SWR / Privat jung / Privat alt
 - **Programmeigenschaften**
 - Messung: Continuous Rating Scale von 0 (nicht gut) bis 100 (sehr gut)
 - 14 Items

• Verlässlicher_Service	• Informative_Moderation
• Informative_Nachrichten	• Unterhaltsame_Moderation
• Unterhaltsame_Nachrichten	• Sympathische_Stimme
• Aktuelle_Musik	• Anspruchsvolle_Comedy
• Alte_Musik	• Lustige_Comedy
• Anregende_Musik	• Spannende_Gewinnspiele
• Beruhigende_Musik	• Attraktive_Preise
- Hinweis: Variablennamen in rot
- **Programmelemente**
 - Messung: Continuous Rating Scale von 0 (nicht gut) bis 100 (sehr gut)
 - 5 Items
 - **Berichterstattung**
 - **Musik**
 - **Moderation**
 - **Comedy**
 - **Gewinnspiele**
 - **Gesamtprogramm**
 - Messung: Continuous Rating Scale von 0 (nicht gut) bis 100 (sehr gut)
 - **Geschlecht**
 - Messung: Auswahlfrage
 - **Alter**
 - Messung: offene Altersangabe
 - **Altersgruppen** (alt vs. jung, 2 Gruppen)
 - **Aktuelle_Musik_A** (Affinität, 2 Gruppen)

4 Datenmanagement

In diesem Kapitel lernen Sie, wie Sie Daten für die Analyse mit R vor- und aufbereiten und wie Sie diesen Datensatz dann in R laden und für weitere R-Sitzungen bereithalten. Daran anschließend wird die Bearbeitung von Daten in R Studio erklärt. In diesem Kapitel beginnen Sie mit der Bearbeitung der Übungsaufgaben und werden selbst mit R arbeiten.

4.1 Datenvorbereitung & Laden von Daten

Die Auswertung in einem Forschungsprojekt kann nur so gut sein wie die dahinterliegenden Daten. Dementsprechend ist es wichtig, dass Sie bei jedem Forschungsprojekt die Daten „in Ordnung halten/bringen“ und dann in R laden. Im folgenden **Video-Tutorial** werden die Datenvorbereitung und das Laden des Datensatzes in R erklärt und demonstriert:



Inhalt		Prof. Dr. Hendrik Godbersen
1	Einleitung	
2	Installation & Grundstruktur von R Studio	
2.1	Installation von R, R Studio & relevanten Paketen	
2.2	Grundstruktur von R Studio	
3	R Befehle & Übungsdaten	
3.1	R Befehle	
3.2	Übungsdaten	
4	Datenmanagement	
4.1	Datenvorbereitung & Laden von Daten	
4.2	Daten bearbeiten in R	
5	Deskriptive Statistik	
5.1	Skalenniveau	
5.2	Häufigkeiten	
5.3	Lagemaße & Streuungsmaße	
6	Inferenzstatistische Verfahren	
6.1	Überblick & Systematisierung der Verfahren	
6.2	Chi-Quadrat-Test	
6.3	t-Test	
6.4	ANOVA	
6.5	Shapiro-Wilk-Test	
6.6	Wilcoxon-Test	
6.7	Korrelation	
6.8	Regression	
6.9	Hauptkomponentenanalyse	
7	Graphiken & Export von Ergebnissen nach Excel	
7.1	Graphiken in R	
7.2	Export von Ergebnissen nach Excel	

Im ersten Schritt geht es um die Datenvorbereitung. Laden Sie sich die Daten vom Server, auf dem Sie Ihre Online-Befragung durchgeführt haben, im Excel-Format herunter. Andere Formate sind auch möglich, jedoch zeigt die Erfahrung, dass im Excel-Format die wenigsten Probleme auftreten. Sollten Sie einen schriftlichen Fragebogen eingesetzt haben oder die Daten über ein anderes Medium erhoben haben, überführen Sie diese in eine Excel-Tabelle.

Die Datenstruktur sollte, wie folgt, aussehen und ist auch in der nachfolgenden Abbildung dargestellt:

- Erste Zeile = Variablenbezeichnungen
- 1 Spalte = 1 Variable
- 1 Zeile = 1 Befragter

Stammnummer	SenderCluster	Verlässlicher_Service	Informative_Nachrichten	Unterhalts_Aktuelle_Mitgliedschaft	Unterhalts_Aktuelle_Mitgliedschaft
1	Energy	Privat jung	6	86	100
2	SWR3	SWR	6	7	20
3	Energy	Privat jung	6	86	100
4	Energy	Privat jung	6	54	100
5	BigFM	Privat jung	9	28	53
6	Antenne1	Privat alt	14	27	86
7	BigFM	Privat jung	15	18	100
8	egoFM	Privat jung	16	27	63
9	BigFM	Privat jung	16	17	94
10	BigFM	Privat jung	18	38	7
11	BigFM	Privat jung	18	24	89
12	egoFM	Privat jung	19	37	43
13	BigFM	Privat jung	19	8	4
14	Das Ding	Privat jung	20	80	40
15	Antenne1	Privat alt	22	17	16
16	Energy	Privat jung	28	26	6
17	Das Ding	Privat jung	28	77	62
18	SWR3	SWR	29	37	33
19	SWR3	SWR	30	70	10
20	Ohr	Privat alt	33	47	33
21	SWR3	SWR	35	23	28
22	BigFM	Privat jung	35	78	6
23	BigFM	Privat jung	36	47	15
24	Energy	Privat jung	37	0	27
25	SWR3	SWR	37	87	87

Eine Empfehlung ist, nur vollständig ausgefüllte Fragebögen in den Datensatz und die spätere Analyse aufzunehmen. Dies vereinfacht den Umgang mit fehlenden Werten, die bei diesem Vorgehen schlichtweg nicht mehr existieren. Die Entscheidung darüber hängt jedoch vom Forschungsprojekt ab.

Voraussetzung dafür, dass Sie die Daten mit R auswerten können, ist hinsichtlich:

- **Variablenbezeichnungen:** keine Leerzeichen & keine Umlaute
 - Empfehlung: kurze & bedeutungsvolle Bezeichnungen
- **Merkmalsausprägungen:**
 - Metrische Variablen (z.B. Alter) als numerischen Wert
 - Kategoriale Variablen (z.B. Geschlecht) als Text

Im zweiten Schritt müssen Sie die **Daten in R Studio laden** und auch für Ihre nächsten R-Sitzungen bereitstellen. Dazu sind die folgenden fünf Schritte notwendig, die Sie im Detail und ohne Ausnahme anwenden sollten/müssen:

0) Öffnen Sie ein neues Script über das Hauptmenü

1) Vorbereitung (wichtig):

- Speichern Sie die Daten in einem Ordner auf Ihrem Computer, der extra für das Projekt angelegt ist

2) Daten aus Excel importieren:

- Unter „Environment“ Daten über „Import Dataset“ Excel-Daten auswählen

3) Daten im R-Format speichern:

- Unter Environment über Speichersymbol Daten im Projektordner speichern

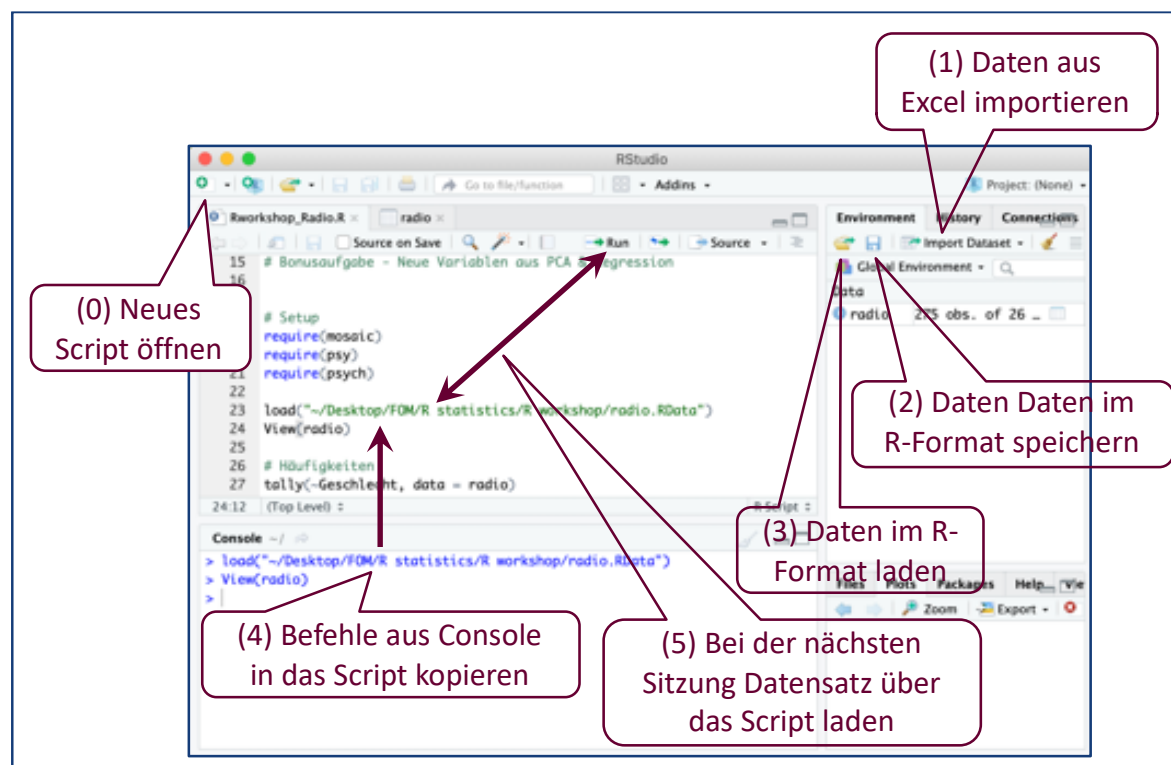
4) Daten im R-Format laden:

- Unter „Environment“ Daten über Öffnungssymbol R-Daten auswählen
- Unter „Environment“ auf den Datensatz klicken
- In der Console generierte Befehle in das Script kopieren

5) Bei der nächsten Anwendung können die Daten über das Script gestartet werden

- Zeile mit dem load-Befehl markieren und Run-Button betätigen

Die fünf oben beschriebenen Schritte sind im Folgenden noch einmal graphisch dargestellt:

**4.2 Daten bearbeiten in R**

In vielen Forschungsprojekten ist es notwendig den erhobenen Datensatz zu bearbeiten. Es kann geboten sein, Variablen zu recodieren, neue Variablen zu berechnen oder einen neuen Datensatz zu erstellen. Im Folgenden finden Sie für die genannten drei Fälle Beispiele:

- Recodieren von Variablen:
 - Inverse Skalen, z.B. 1 zu 6, 2 zu 5, 3 zu 4...
 - Zusammenfassen von Werten, z.B. Einwohner von Bayern und Baden-Württemberg (erhobene Variable) zu Süddeutschland (neu zu bildende Variable)
 - Ändern von metrischen zu kategorialen Werten (siehe Datenvorbereitung für R)
 - ...
- Berechnen neuer Variablen:
 - Bildung eines neuen („übergeordneten“) Konstrukts aus mehreren Variablen, z.B. Berechnung einer Gesamtbewertung aus mehreren Einzelbewertungen über das arithmetische Mittel
 - Berechnung einer neuen Variablen, die sich aus dem Verhältnis mehrerer Variablen bestimmt, z.B. Berechnung der Rentabilität aus den Variablen Gewinn und Umsatz
 - ...
- Erstellen neuer Datensätze:
 - Ausschluss von Befragten mit bestimmten Merkmalsausprägungen, z.B. zu alte oder zu junge Befragte
 - Analyse nur eines Teils der Befragten, z.B. ausschließliche Analyse der Antworten von Frauen
 - ...

Schauen Sie sich nun das **Video-Tutorial** an, in dem die Bearbeitung von Daten in R erklärt wird und zu allen drei oben genannten Datenbearbeitungen eine Übungsaufgabe gestellt wird:



Wie im Video-Tutorial dargestellt, finden Sie im Folgenden noch einmal die Aufgaben und Lösungen zur Bearbeitung von Daten.

Aufgaben:

- Generieren Sie aus SenderCluster eine neue Variable (SenderCluster2): „Privat jung“ & „Privat alt“ zu „Privat“ und „SWR“ zu „SWR“.
- Berechnen Sie aus den Variablen „Comedy“ und „Gewinnspiele“ in einer neuen Variablen (CoGe) den Mittelwert.
- Generieren Sie einen neuen Datensatz, der nur aus Frauen besteht.

Benötigte Befehle:Recoding (new) variables:

```
dataset$new_variable[dataset$old_variable=="XXX"] <- "new_value"
```

→ Please note: categorial values require quotes (""), numerical values not

Calculating new variables:

```
dataset$new_variable <- dataset$variable1 + dataset$variable2
```

Changing the the variable formats:

...

Creating new datasets (is equal, is not equal etc.):

```
new_dataset <- dataset[dataset$variable == "value",]
```

```
new_dataset <- dataset[dataset$variable != "value",]
```

```
new_dataset <- dataset[dataset$variable > value,]
```

Relevante Variablen:

- SenderCluster (Aufgabe 1)
- Comedy und Gewinnspiele (Aufgabe 2)
- Geschlecht (Aufgabe 3)

Lösung in R:

R-Befehle

```

32 # Setup
33 require(mosaic)
34 require(psy)
35 require(psych)
36 require(openxlsx)
37 load("~/Desktop/R_workshop/radio.RData")
38 View(radio)
39
40 # Generieren von neuen Variablen & Datensätzen
41 radio$SenderCluster2[radio$SenderCluster=="Privat jung"] <- "Privat"
42 radio$SenderCluster2[radio$SenderCluster=="Privat alt"] <- "Privat"
43 radio$SenderCluster2[radio$SenderCluster=="SWR"] <- "SWR"
44
45 radio$CoGe <- (radio$Comedy + radio$Gewinnspiele) / 2
46
47 radio_weiblich <- radio[radio$Geschlecht == "weiblich",]
48 View(radio_weiblich)
49
50
51 # Überprüfen, ob alles funktioniert
52 View(radio)
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Neuer Datensatz (nur aus Frauen bestehend) (Aufgabe 3)

Neu gebildete Variablen (Aufgaben 1 & 2)

id	Gewinnspiele	Gesamtprogramm	Geschlecht	Alter	Altersgruppen	Aktuelle_Musik_A	SenderCluster2	CoGe
7	3	54	weiblich	31	alt	eher nicht affin zu aktueller Musik	Privat	5.0
2	2	2	maennlich	34	alt	eher affin zu aktueller Musik	SWR	2.0
95	2	96	maennlich	23	jung	eher nicht affin zu aktueller Musik	Privat	48.5
72	9	86	weiblich	24	jung	eher nicht affin zu aktueller Musik	Privat	40.5
33	40	33	maennlich	22	jung	eher nicht affin zu aktueller Musik	Privat	36.5
6	100	79	weiblich	23	jung	eher nicht affin zu aktueller Musik	Privat	53.0
12	9	84	maennlich	30	alt	eher affin zu aktueller Musik	Privat	10.5
42	5	86	maennlich	25	jung	eher affin zu aktueller Musik	Privat	23.5
21	9	74	weiblich	33	alt	eher nicht affin zu aktueller Musik	Privat	15.0
14	14	85	maennlich	26	jung	eher affin zu aktueller Musik	Privat	14.0
20	13	78	weiblich	23	jung	eher nicht affin zu aktueller Musik	Privat	16.5
8	6	73	maennlich	24	jung	eher nicht affin zu aktueller Musik	Privat	7.0
64	80	81	weiblich	24	jung	eher nicht affin zu aktueller Musik	Privat	72.0

5 Deskriptive Statistik

Mit dem fünften Kapitel beginnt der Kern des Kurses, die Vermittlung der statistischen Verfahren und deren Anwendung in R. Als Voraussetzung dafür werden im ersten Unterkapitel die Skalenniveaus erklärt. Darauf aufbauend werden Häufigkeiten sowie Lageparameter und Streuungsmaße behandelt. Auch wenn Sie schon mit diesen deskriptiven statistischen Verfahren vertraut sind, wird nachdrücklich empfohlen, diese Kapitel sorgfältig durchzugehen, da hier die Arbeitsweise in R deutlich wird. Dementsprechend finden Sie in diesem Kapitel auch mehrere Übungsaufgaben, so dass Sie nach Abschluss des Kapitels mit der Benutzung von R vertraut sein sollten.

Wichtiger Hinweis: Stellen Sie sicher, dass Sie vor der Auswertung **bei jeder neuen Sitzung** die benötigten **Zusatzpakete geladen** haben über den Befehl **require(mosaic)** (für das Laden der anderen Pakete ersetzen Sie mosaic mit dem entsprechenden Paket).

Schauen Sie sich auch das folgende **Video-Tutorial** an:



5.1 Skalenniveaus

Für die Anwendung von statistischen Verfahren ist es wichtig, welches Skalenniveau die zu analysierenden Variablen haben. Schauen Sie sich dazu das folgende **Video** an:



Für Anwendung von Häufigkeiten und Lageparametern sowie auch für die in Kapitel sechs erklärten inferenzstatistischen Verfahren ist es notwendig, zwischen zwei Arten von Variablen zu unterscheiden:

- **Kategoriale Variablen** mit nominalem und ordinalem Skalenniveau
- **Metrische Variablen** mit intervallskaliertem und verhältnisskaliertem Niveau

Eine Aufteilung der Skalenniveaus mit mathematischen Eigenschaften, Messwertcharakteristik, Beispiel und möglichen Lageparametern finden Sie in der folgenden Tabelle:

Skalenniveau		Math. Eigenschaften	Beschreibung der Messwertcharakteristika	Beispiel	Lageparameter
Kategorial	Nominal	$\neq$	Die Messwerte sind identisch oder nicht identisch.	Geschlecht	• Modus
	Ordinal	$\neq$; $</>$	Die Messwerte sind größer, kleiner oder identisch.	Olympische Ränge	• Modus • Median
Metrisch	Intervall	$\neq$; $</>$; - ; +	Die Distanz zwischen den Messwerten kann angegeben werden.	Temperatur	• Modus • Median • Arithm. Mittel
	Verhältnis	$\neq$; $</>$; + / - ; * / ÷	Die Distanz zwischen den Messwerten und deren Verhältnis kann angegeben werden.	Körpergröße	

5.2 Häufigkeiten

In den meisten Auswertungen sind drei Arten von Häufigkeiten relevant, wobei i.d.R. die relative Häufigkeit am wichtigsten ist:

- **Absolute Häufigkeit:** gibt an, wie häufig ein Wert auftritt
- **Relative Häufigkeit:** gibt das Verhältnis eines untersuchten Merkmalsträgers zu einem Wert oder einer Klasse von Werten an
- **Kreuztabelle:** stellt die gemeinsame Verteilung zweier Variablen dar

Im folgenden **Video-Tutorial** werden die Häufigkeiten, ihre statistische Verwendung und die Umsetzung in R, inklusive Übungsaufgaben, erläutert:



Aufgaben 1:

- Wie viele Männer und Frauen gibt es in der Stichprobe?
- Wie hoch ist der Anteil der Männer und Frauen in %?
- Wie hoch ist der Anteil der SenderCluster in %?

Benötigte Befehle 1:

Frequencies

```
tally(~variable1, data = dataset)
```

```
  tally(~variable1, format = "percent", data = dataset)
```

```
tally(variable1~variable2, data = dataset)
```

```
  tally(variable1~variable2, format = "percent", data = dataset)
```

Relevante Variablen 1:

- SenderCluster
- Geschlecht

Lösung in R 1:

The screenshot shows the RStudio interface. The script editor on the left contains the following R code:

```

48 View(radio_weiblich)
49
50
51 # Häufigkeiten, Lageparameter & Streuungsmasse
52 # Häufigkeiten
53 tally(~Geschlecht, data = radio)
54 tally(~Geschlecht, format = "proportion", data = radio)
55 tally(~Geschlecht, format = "percent", data = radio)
56
57 tally(~SenderCluster, format = "percent", data = radio)
58
59 tally(SenderCluster~Geschlecht, format = "percent", data = radio)
60
61 # Lageparameter & Streuungsmasse
62 mean(~Alter, data = radio)
63 mean(~Alter, data = radio)
64
65 # Top Level
66
67

```

Two callout boxes highlight specific parts of the code:

- R-Befehle** points to lines 53-57, which are the `tally` commands.
- Ergebnisse** points to the console output of these commands.

The console output shows the following results:

```

> # Häufigkeiten, Lageparameter & Streuungsmasse
> # Häufigkeiten
> tally(~Geschlecht, data = radio)
Geschlecht
maennlich weiblich
129      146
> tally(~Geschlecht, format = "proportion", data = radio)
Geschlecht
maennlich weiblich
0.4690909 0.5309091
> tally(~Geschlecht, format = "percent", data = radio)
Geschlecht
maennlich weiblich
46.90909 53.09091
> tally(~SenderCluster, format = "percent", data = radio)
SenderCluster
Privat alt Privat jung      SWR
23.63636 45.81818 30.54545

```

The Environment pane on the right shows the loaded data objects:

- `radio`: 275 obs. of 28 variables
- `radio_weiblich`: 146 obs. of 28 variables

Aufgaben 2:

- Wie verteilen sich Männer und Frauen auf die SenderCluster (%)?

Benötigte Befehle 2:**Frequencies**

`tally(~variable1, data = dataset)`

`tally(~variable1, format = "percent", data = dataset)`

`tally(variable1~variable2, data = dataset)`

`tally(variable1~variable2, format = "percent", data = dataset)`

Relevante Variablen 2:

- SenderCluster
- Geschlecht

Lösung in R 2:

The screenshot shows the RStudio interface. The script editor contains the following code:

```

49
50
51 # Häufigkeiten, Lageparameter & Streuungsmasse
52 # Häufigkeiten
53 tally(~Geschlecht, data = radio)
54 tally(~Geschlecht, format = "proportion", data = radio)
55 tally(~Geschlecht, format = "percent", data = radio)
56
57 tally(~SenderCluster, format = "percent", data = radio)
58
59 tally(SenderCluster~Geschlecht, format = "percent", data = radio)
60
61 # Lageparameter & Streuungsmasse
62 mean(~Alter, data = radio)
63 median(~Alter, data = radio)
64 var(~Alter, data = radio)
65.1 (Top Level)

```

The console shows the output of the first three commands:

```

> tally(~Geschlecht, format = "proportion", data = radio)
Geschlecht
maennlich weiblich
0.4690909 0.5309091
> tally(~Geschlecht, format = "percent", data = radio)
Geschlecht
maennlich weiblich
46.90909 53.09091
> tally(SenderCluster, format = "percent", data = radio)
SenderCluster
Privat alt Privat jung SMR
73.63636 45.81818 30.54545

```

The fourth command is highlighted in the script editor, and its output is shown in the console:

```

> tally(SenderCluster~Geschlecht, format = "percent", data = radio)
SenderCluster maennlich weiblich
Privat alt 20.15584 26.71233
Privat jung 44.96124 46.57534
SMR 34.88372 26.71233

```

5.3 Lageparameter & Streuungsmaße

Im Folgenden sind die **Lageparameter** definiert:

- **Arithmetisches Mittel:** Division der Summe aller Werte durch die Anzahl der Werte; metrisches Skalenniveau notwendig
- **Median:** Wert, der die Grenze zwischen oberer und unterer Hälfte markiert; mindestens ordinales Skalenniveau
- **Modalwert:** Häufigster Wert einer Häufigkeitsverteilung; mindestens nominales Skalenniveau

Neben den Lageparametern sind die **Streuungsmaße** für die Interpretation von Daten relevant. Die beiden zentralen Streuungsmaße Varianz und Standardabweichung sind, wie folgt, definiert:

- **Streuungsparameter**
 - **Varianz:** Durchschnittliche quadrierte Abweichung der gemessenen Werte vom arithmetischen Mittel (beachte: empirische Varianz $\rightarrow /n$; Stichprobenvarianz $\rightarrow /n-1$)

- **Standardabweichung:** Durchschnittliche Abweichung der gemessenen Werte vom arithmetischen Mittel

Schauen Sie sich auf jeden Fall das nächste Video-Tutorial an, in dem unter anderem die Lageparameter und die Streuungsmaße detaillierter erklärt werden. Es sei an dieser Stelle darauf verwiesen, dass die Varianz (Streuungsmaß) die mathematische Grundlage für die meisten inferenzstatistischen Verfahren ist. („**Wenn Sie die Varianz verstanden haben, haben Sie den Kern der (inferenz-)statistischen Analyseverfahren verstanden**“):



Aufgaben 1:

- Was ist das durchschnittliche Alter der Befragten (arith. Mittel)?
- Welchen Wert hat der Median bei der Variable Alter?
- Wie stark streuen die gemessenen Werte der Variable Alter um das arithmetische Mittel (Varianz & Standardabweichung)?

Benötigte Befehle 1:

Median, mean, variance, standard deviation, maximum, minimum

```
median(~variable, data = dataset) / mean(~variable, data = dataset)
```

```
var(~variable, data = dataset) / sd(~variable, data = dataset)
```

```
max(~variable, data = dataset) / min(~variable, data = dataset)
```

```
favstats(~variable, data = dataset)
```

Relevante Variablen 1:

- Alter

Lösung in R 1:

The screenshot shows the RStudio interface. The script editor contains the following code:

```

56 tally(~SenderCluster, format = "percent", data = radio)
57
58
59 tally(SenderCluster-Geschlecht, format = "percent", data = radio)
60
61 # Lagrange-Multiplikator & Streuungsmaße
62 mean(~Alter, data = radio)
63 median(~Alter, data = radio)
64 var(~Alter, data = radio)
65 sd(~Alter, data = radio)
66
67 mean(~Gesamtprogramm, data = radio)
68 median(~Gesamtprogramm, data = radio)
69 var(~Gesamtprogramm, data = radio)
70 sd(~Gesamtprogramm, data = radio)
71

```

The console shows the output of the commands:

```

SenderCluster
Privat alt Privat jung      SWR
23.63636  45.81818  30.54545
> tally(SenderCluster-Geschlecht, format = "percent", data = radio)
Geschlecht
SenderCluster männlich weiblich
Privat alt  20.15084 26.71233
Privat jung 44.96124 46.57534
SWR         34.88372 26.71233
> mean(~Alter, data = radio)
[1] 28.84727
> median(~Alter, data = radio)
[1] 26
> var(~Alter, data = radio)
[1] 46.30506
> sd(~Alter, data = radio)
[1] 6.804782

```

Red boxes in the original image highlight the R commands (lines 62-65) and the corresponding console output (lines 62-65). A red box also highlights the console output for the first set of commands (lines 62-65).

Aufgaben 2:

- Wie hoch ist die durchschnittliche Bewertung des Gesamtprogramms (arith. Mittel)?
- Welchen Wert hat der Median bei der Variable Gesamtprogramm?
- Wie stark streuen die gemessenen Werte der Variable Gesamtprogramm um das arithmetische Mittel (Varianz & Standardabweichung)?

Benötigte Befehle 2:

Median, mean, variance, standard deviation, maximum, minimum

`median(~variable, data = dataset)` / `mean(~variable, data = dataset)`

`var(~variable, data = dataset)` / `sd(~variable, data = dataset)`

`max(~variable, data = dataset)` / `min(~variable, data = dataset)`

`favstats(~variable, data = dataset)`

Relevante Variablen 2:

- Gesamtprogramm

Lösung in R 2:

The screenshot shows the RStudio interface. The script editor contains the following code:

```

61 # Lageparameter & Streuungsmasse
62 mean(~Alter, data = radio)
63 median(~Alter, data = radio)
64 var(~Alter, data = radio)
65 sd(~Alter, data = radio)
66
67 mean(~Gesamtprogramm, data = radio)
68 median(~Gesamtprogramm, data = radio)
69 var(~Gesamtprogramm, data = radio)
70 sd(~Gesamtprogramm, data = radio)
71
72 favstats(~Alter, data = radio)
73 favstats(~Gesamtprogramm, data = radio)
74
75 mean(Gesamtprogramm-Geschlecht, data = radio)
76
77.31 (Top Level)

```

The console shows the following output:

```

SNR      34.88372 26.71233
> # Lageparameter & Streuungsmasse
> mean(~Alter, data = radio)
[1] 28.84727
> median(~Alter, data = radio)
[1] 26
> var(~Alter, data = radio)
[1] 46.30506
> sd(~Alter, data = radio)
[1] 6.804885
> mean(~Gesamtprogramm, data = radio)
[1] 77.53818
> median(~Gesamtprogramm, data = radio)
[1] 78
> var(~Gesamtprogramm, data = radio)
[1] 292.9356
> sd(~Gesamtprogramm, data = radio)
[1] 17.11536

```

Annotations in the image:

- A red box highlights the R commands (lines 61-76) in the script editor, with a callout labeled "R-Befehle".
- A red box highlights the console output (lines 1-10), with a callout labeled "Ergebnisse".

Aufgaben 3:

- Bestimmen Sie für die Variable Alter das arithmetische Mittel, den Median, die Standardabweichung etc. mit nur einem R-Befehl.
- Bestimmen Sie für die Variable Gesamtprogramm das arithmetische Mittel, den Median, die Standardabweichung etc. mit nur einem R-Befehl.

Benötigte Befehle 3:

Median, mean, variance, standard deviation, maximum, minimum

`median(~variable, data = dataset)` / `mean(~variable, data = dataset)`

`var(~variable, data = dataset)` / `sd(~variable, data = dataset)`

`max(~variable, data = dataset)` / `min(~variable, data = dataset)`

`favstats(~variable, data = dataset)`

Relevante Variablen 3:

- Alter
- Gesamtprogramm

Lösung in R 3:

The screenshot shows the RStudio interface with a script editor on the left and a console on the bottom left. The script editor contains R code for calculating summary statistics for variables 'Alter' and 'Gesamtprogramm' in a dataset named 'radio'. The console shows the output of these commands. Two callout boxes highlight specific parts: 'R-Befehle' points to the code lines, and 'Ergebnisse' points to the console output.

R-Befehle

```

65 sd(~Alter, data = radio)
66
67 mean(~Gesamtprogramm, data = radio)
68 median(~Gesamtprogramm, data = radio)
69 var(~Gesamtprogramm, data = radio)
70 sd(~Gesamtprogramm, data = radio)
71
72 favstats(~Alter, data = radio)
73 favstats(~Gesamtprogramm, data = radio)
74
75 mean(Gesamtprogramm-Geschlecht, data = radio)
76
77
78 # Inferenzstatistik
79 # Chi-Quadrat-Test
80 # chisq.test(Gesamtprogramm-Geschlecht, data = radio)
81
74.1 (Top Level) >

```

Ergebnisse

```

> var(~Alter, data = radio)
[1] 46.30506
> sd(~Alter, data = radio)
[1] 6.804782
> mean(~Gesamtprogramm, data = radio)
[1] 77.53818
> median(~Gesamtprogramm, data = radio)
[1] 78
> var(~Gesamtprogramm, data = radio)
[1] 292.9356
> sd(~Gesamtprogramm, data = radio)
[1] 17.11536
> favstats(~Alter, data = radio)
  min  Q1 median  Q3  max   mean    sd   n missing
19  24   26  32   48 28.84727 6.804782 275      0
> favstats(~Gesamtprogramm, data = radio)
  min  Q1 median  Q3  max   mean    sd   n missing
 2  70   78  90 100 77.53818 17.11536 275      0

```

Aufgaben 4:

- Stellen Sie das durchschnittliche Alter von Männern und Frauen gegenüber (arith. Mittelwerte).
- Stellen Sie die durchschnittliche Bewertung des Gesamtprogramms von Männern und Frauen gegenüber (arith. Mittelwerte).

Benötigte Befehle 4:

Median, mean, variance, standard deviation, maximum, minimum

`median(~variable, data = dataset)` / `mean(~variable, data = dataset)`

`var(~variable, data = dataset)` / `sd(~variable, data = dataset)`

`max(~variable, data = dataset)` / `min(~variable, data = dataset)`

`favstats(~variable, data = dataset)`

Relevante Variablen 4:

- Alter
- Gesamtprogramm
- Geschlecht

Lösung in R 4:

The screenshot shows the RStudio interface with a script editor on the left and a console on the bottom. The script editor contains R code for calculating various statistics. The console shows the output of these commands. Two red boxes highlight specific parts of the code and output, with callout labels 'R-Befehle' and 'Ergebnisse'.

R-Befehle (R Commands):

```
sd(-Alter, data = radio)
mean(-Gesamtprogramm, data = radio)
median(-Gesamtprogramm, data = radio)
var(-Gesamtprogramm, data = radio)
sd(-Gesamtprogramm, data = radio)
favstats(-Alter, data = radio)
favstats(-Gesamtprogramm, data = radio)
mean(Alter-Geschlecht, data = radio)
mean(Gesamtprogramm-Geschlecht, data = radio)
```

Ergebnisse (Results):

```
> median(-Gesamtprogramm, data = radio)
[1] 78
> var(-Gesamtprogramm, data = radio)
[1] 292.9356
> sd(-Gesamtprogramm, data = radio)
[1] 17.11536
> favstats(-Alter, data = radio)
  min  Q1 median  Q3  max   mean    sd  n missing
19  24    26  32   48 28.84727 6.804782 275      0
> favstats(-Gesamtprogramm, data = radio)
  min  Q1 median  Q3  max   mean    sd  n missing
 2  20    29  40  100 27.53818 17.11536 275      0
> mean(Alter-Geschlecht, data = radio)
      moennlich  weiblich
29.28682  28.45890
> mean(Gesamtprogramm-Geschlecht, data = radio)
      moennlich  weiblich
74.51163  80.21233
```

6 Inferenzstatistische Verfahren

In diesem Kapitel wird zuerst ein Überblick über die Verfahren der strukturprüfenden Analyse (Inferenzstatistik) und der strukturentdeckenden Analyse gegeben, so dass Sie wissen, wann Sie welches Verfahren anwenden müssen. Daran anschließend werden die einzelnen Verfahren in jeweils einem Unterkapitel erklärt, Sie wenden diese in Übungsaufgaben an und lernen, wie Sie die Ergebnisse interpretieren müssen.

6.1 Überblick & Systematisierung der Verfahren

In der Inferenzstatistik wird geprüft, ob ein signifikanter (nicht zufälliger) Unterschied zwischen zwei oder mehreren Gruppen vorliegt oder ob eine oder mehrere unabhängige Variablen einen signifikanten (nicht zufälligen) Einfluss auf eine abhängige Variable hat/haben. An dieser Stelle sei darauf verwiesen, dass **Forschungsprojekte i.d.R. immer inferenzstatistische Auswertungen erfordern**. Die Voraussetzung für die Wahl des richtigen Auswertungsverfahrens ist das Wissen über die Skalenniveaus der unabhängigen und abhängigen Variablen. Die Skalenniveaus wurden in Kapitel 5.1 erläutert und sind in der folgenden Tabelle noch einmal dargestellt:

Skalenniveau		Math. Eigenschaften	Beschreibung der Messwertcharakteristika	Beispiel	Lageparameter
Kategorial	Nominal	$\neq$	Die Messwerte sind identisch oder nicht identisch.	Geschlecht	• Modus
	Ordinal	$\neq$; $</>$	Die Messwerte sind größer, kleiner oder identisch.	Olympische Ränge	• Modus • Median
Metrisch	Intervall	$\neq$; $</>$; - ; +	Die Distanz zwischen den Messwerten kann angegeben werden.	Temperatur	• Modus • Median • Arithm. Mittel
	Verhältnis	$\neq$; $</>$; + / - ; * / ÷	Die Distanz zwischen den Messwerten und deren Verhältnis kann angegeben werden.	Körpergröße	

Schauen Sie sich nun das **Video-Tutorial** zu den statistischen Analyseverfahren an:



Zu Beginn einer „statistischen Karriere“ fällt es vielen schwer, zwischen unabhängiger und abhängiger Variablen zu unterscheiden. Folgende Faustregel kann helfen: **die unabhängige Variable beeinflusst die abhängige Variable**, zum Beispiel: Wie stark beeinflussen die Werbeausgaben (unabhängige Variable) den Absatz eines Produktes (abhängige Variable)?

Die folgende Abbildung gibt einen Überblick über die einzelnen Verfahren und welches Verfahren bei welchen Skalenniveaus der unabhängigen und abhängigen Variablen angewandt wird:

Strukturprüfende Verfahren		Unabhängige Variable	
		nicht-metrisch	metrisch
Abhängige Variable	nicht-metrisch	Chi ² -Test	Diskriminanzanalyse
	metrisch	t-Test*** (bei 2 Gruppen) & Varianzanalyse (bei ≥ 3 Gruppen)	Regressionsanalyse (Dependenz) & Korrelationsanalyse (Interdependenz)

*** „Ergänzende“ Tests:

- Shapiro-Wilk-Test
(prüft Stichproben auf Normalverteilung, die für den t-Test notwendig ist)
- Mann-Whitney-Test/Wilcoxon-Test
(prüft mind. ordinal skalierte Daten auf Unterschiede ohne Berücksichtigung der Verteilungsform)

Strukturentdeckende Verfahren	• Hauptkomponentenanalyse (Faktoranalyse) • Clusteranalyse
-------------------------------	---

Im Folgenden werden zur Verdeutlichung beispielhafte Forschungsfragen gegeben, die mit den einzelnen Analyse-Verfahren untersucht werden könnten:

- Chi²-Test: Haben Frauen und Männer eine unterschiedliche Lieblingsfarbe (Unterscheidung in unabhängige und abhängige Variable ist aus statistischer Sicht nicht notwendig)?

- t-Test (zwei Kontrastgruppen): Haben Frauen und Männer (unabhängige Variable) ein unterschiedliches Einkommen (abhängige Variable)?
- Varianzanalyse (ANOVA) (mehr als zwei Kontrastgruppen): Haben die Bewohner der 16 Bundesländer (unabhängige Variable) ein unterschiedliches Einkommen (abhängige Variable)?
- Shapiro-Wilk-Test & Wilcoxon-Test:
 - wie t-Test, werden aber nur bei kleinen Stichproben eingesetzt (Faustregel: Stichprobenumfang < 50)
 - Shapiro-Wilk-Test prüft, ob die Stichprobe normalverteilt ist
 - Wilcoxon-Test prüft – wie der t-Test – Unterschiede zwischen zwei Gruppen
- Korrelationsanalyse: Gibt es einen (signifikanten / nicht zufälligen) Zusammenhang zwischen Alter und Einkommen (bei diesem Test gibt es keine abhängigen und unabhängigen Variablen)?
- Regressionsanalyse: Wie stark beeinflussen die Werbeausgaben (unabhängige Variable) den Absatz eines Produktes (abhängige Variable)?
- Hauptkomponentenanalyse (strukturentdeckendes Verfahren ohne abhängige und unabhängige Variablen): Können mehrere metrische Items (abgefragte Eigenschaften) zu Komponente („neuen Variablen“) zusammengefasst werden?

6.2 Chi-Quadrat-Test

Der χ^2 -Test prüft, ob nominale (kategoriale) Variablen (un-)abhängig voneinander sind – in anderen Worten: ob die gemessenen Werte nicht zufällig von den erwarteten Werten (Werte, die bei „vollkommener“ Unabhängigkeit der Variablen gegeben sein müssten) abweichen. Folgende **Fragestellung** liegt ein χ^2 -Test zugrunde:

- Gibt es einen „nicht-zufälligen“ statistischen Zusammenhang zwischen zwei nominalskalierten Merkmalen? Sind zwei kategoriale Variablen unabhängig?

Eine konkrete Forschungsfrage könnte zum Beispiel lauten:

- Haben Frauen und Männer eine unterschiedliche Lieblingsfarbe (Unterscheidung in unabhängige und abhängige Variable ist aus statistischer Sicht nicht notwendig)?

Das **Vorgehen bei der Analyse** besteht aus zwei Schritten:

- 1) Ist der p-value < 0.05 (< 0.01 , < 0.001)? → Signifikanz?
- 2) Wie unterscheiden sich die Häufigkeiten?

Auch die „**statistischen**“ **Analyseschritte („im Hintergrund“)** bestehen aus zwei Schritten:

- 1) Berechnung der erwarteten Häufigkeiten (e), wenn kein Unterschied zwischen den Variablen bestehen würde (Spaltensumme * Zeilensumme / Merkmalsträger)
- 2) Bestimmung, ob die gemessenen Werte (h) signifikant von den erwarteten Werten abweichen

Schauen Sie sich in jedem Fall das **Video-Tutorial** zum χ^2 -Test an, da in diesem auch die Grundprinzipien der Inferenzstatistik ausführlicher erklärt werden, die auch allen weiteren Verfahren zugrunde liegen:



Aufgaben:

- Besteht ein signifikanter Unterschied zwischen Männern und Frauen bei der Präferenz für die SenderCluster?
- Unterscheidet sich das junge und alte Alterssegment bei der Affinität zu aktueller Musik? Und wenn ja, wie fällt dieser Unterschied aus?

Benötigte Befehle:

Chi² test

```
xchisq.test(variable1~variable2, data = dataset)
```

Relevante Variablen:

- SenderCluster
- Geschlecht
- Alter
- Altersgruppen
- Aktuelle_Musik_A

Lösung in R:**R-Befehle und Ergebnisse zu Aufgabe 1:**

R-Befehle

```

77
78 # Inferenzstatistik
79
80 # Chi-Quadrat-Test
81 xchisq.test(SenderCluster-Geschlecht, data = radio)
82
83 xchisq.test(Aktuelle_Musik_A-Altersgruppen, data = radio)
84
85 (Top Level) 2

```

Ergebnisse

```

> xchisq.test(SenderCluster-Geschlecht, data = radio)

Pearson's Chi-squared test

data: tally(x, data = data)
X-squared = 2.7819, df = 2, p-value = 0.2488

  26      39
(30.49) (34.51)
[0.663] [0.584]
<-0.81> <-0.70>

  58      68
(59.11) (66.89)
[0.621] [0.618]
<-0.14> <-0.14>

  45      39
(39.48) (44.68)
[0.795] [0.702]
<-0.89> <-0.84>

key:
observed
(expected)
[contribution to X-squared]
<-Pearson residual>

```

R-Befehle und Ergebnisse zu Aufgabe 2:

R-Befehle

```

76
77 # Inferenzstatistik
78 # Chi-Quadrat-Test
79 xchisq.test(SenderCluster-Geschlecht, data = radio)
80
81 xchisq.test(Aktuelle_Musik_A-Altersgruppen, data = radio)
82 tally(Aktuelle_Musik_A-Altersgruppen, format = "percent", data = radio)
83
84 # t-Test
85 t.test(Gesamtprogramm-Geschlecht, data = radio)
86
87 (Top Level) 2

```

Ergebnisse

```

> xchisq.test(Aktuelle_Musik_A-Altersgruppen, data = radio)

Pearson's Chi-squared test with Yates' continuity correction

data: tally(x, data = data)
X-squared = 6.7412, df = 1, p-value = 0.009421

  55      78
(66.26) (66.74)
[1.75] [1.73]
<-1.38> <-1.38>

  82      60
(70.74) (71.26)
[1.64] [1.62]
<-1.34> <-1.33>

key:
observed
(expected)
[contribution to X-squared]
<-Pearson residual>

> tally(Aktuelle_Musik_A-Altersgruppen, format = "percent", data = radio)

      Altersgruppen
Aktuelle_Musik_A  alt  jung
  eher affin zu aktueller Musik  40.14599 56.52174
  eher nicht affin zu aktueller Musik 59.85401 43.47826

```

6.3 t-Test

Der t-Test ist ein parametrisches Verfahren. Beim Zweistichproben-t-Test wird überprüft, ob sich eine metrische Variable zwischen zwei Stichproben signifikant (nicht zufällig) unterscheidet. Folgende **Fragestellungen** liegen einem t-Test zugrunde:

- Gibt es einen „nicht-zufälligen“ statistischen Unterschied zwischen zwei Stichproben in Bezug auf ein metrisches Merkmal?
- Gibt es einen „nicht-zufälligen“ Einfluss einer nominalen Variablen (2 Gruppen) auf eine metrische Variable?

Eine konkrete Forschungsfrage könnte zum Beispiel lauten:

- Haben Frauen und Männer (unabhängige Variable) ein unterschiedliches Einkommen (abhängige Variable)?

Der t-Test kann weiterhin **differenziert** werden in:

- t-Test bei unabhängigen Stichproben – Messungen bei zwei Gruppen
- t-Test bei abhängigen Stichproben – Messungen bei derselben Gruppe (z.B. vorher – nachher)

Das **Vorgehen bei der Analyse** besteht aus zwei Schritten (wie auch beim χ^2 -Test):

- Ist der p-value < 0.05 (< 0.01 , < 0.001)? → Signifikanz?
- Wie unterscheiden sich die Mittelwerte?

Im Folgenden finden Sie das **Video-Tutorial**, in dem der t-Test erklärt werden und wie man diesen in R anwendet:



Aufgaben:

- Beurteilen Männer und Frauen das Gesamtprogramm von Radiosendern (signifikant) unterschiedlich?
- Beurteilen Männer und Frauen die Musik von Radiosendern (signifikant) unterschiedlich?

Benötigte Befehle:**t-test**

```
t.test(variable~grouping_variable, data = dataset)
```

Relevante Variablen:

- Musik
- Gesamtprogramm
- Geschlecht

Lösung in R:

The screenshot shows the RStudio interface. The script editor contains the following code:

```
84 tally(Aktuelle_Musik_A-Altersgruppen, format = "percent", data = radio)
85
86 # t-test
87 t.test(Gesamtprogramm-Geschlecht, data = radio)
88 t.test(Musik-Geschlecht, data = radio)
89
90 # Anova
91 summary(aov(Gesamtprogramm-SenderCluster, data = radio))
92 main(Gesamtprogramm-SenderCluster ~ Anova, data = radio)
93 (Top Level)
```

Two callouts point to specific parts of the code:

- A callout labeled "R-Befehle" points to lines 87 and 88.
- A callout labeled "Ergebnisse" points to the console output for the two t-tests.

The console output shows the results of the two t-tests:

```
> t.test(Gesamtprogramm-Geschlecht, data = radio)

Welch Two Sample t-test

data: Gesamtprogramm by Geschlecht
t = -2.7355, df = 229.18, p-value = 0.006715
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -9.806829 -1.594572
sample estimates:
mean in group männlich mean in group weiblich
74.51163 80.21233

> t.test(Musik-Geschlecht, data = radio)

Welch Two Sample t-test

data: Musik by Geschlecht
t = -2.5525, df = 223.81, p-value = 0.01136
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -11.296221 -1.453168
sample estimates:
mean in group männlich mean in group weiblich
77.19380 83.56849
```

6.4 ANOVA

Die Varianzanalyse (Analysis of Variance, ANOVA) ist ein parametrisches Verfahren und untersucht, ob sich die Mittelwerte mehrerer (mehr als zwei) unabhängiger Stichproben systematisch unterscheiden. Folgende **Fragestellung** liegt der ANOVA zugrunde:

- Gibt es einen „nicht-zufälligen“ statistischen Unterschied zwischen mehr als zwei Stichproben in Bezug auf ein metrisches Merkmal?

Eine konkrete Forschungsfrage könnte zum Beispiel lauten:

- Haben die Bewohner der 16 Bundesländer (unabhängige Variable) ein unterschiedliches Einkommen (abhängige Variable)?

Die ANOVA weist somit Ähnlichkeiten zum t-Test auf, lässt sich aber, wie folgt, vom **t-Test** abgrenzen:

- Während der t-Test metrische Merkmale von maximal zwei Stichproben auf signifikante Unterschiede prüft, werden bei der Varianzanalyse mehrere Mittelwerte verglichen.

Hier finden Sie das **Video-Tutorial** mit Aufgaben in R und Lösungen:



Aufgaben:

- Beurteilen die Hörer der verschiedenen SenderCluster das Gesamtprogramm von Radiosendern (signifikant) unterschiedlich?
- Beurteilen die Hörer der verschiedenen SenderCluster die Moderation von Radiosendern (signifikant) unterschiedlich?

Benötigte Befehle:**ANOVA**

```
summary(aov(variable~grouping_variable, data = dataset))
```

Relevante Variablen:

- SenderCluster
- Moderation
- Gesamtprogramm

Lösung in R:

The screenshot shows the RStudio interface. The script editor on the left contains R code for ANOVA tests. A red box highlights the ANOVA commands, with a callout labeled 'R-Befehle'. The console on the bottom left shows the output of these commands, with a red box highlighting the ANOVA results, with a callout labeled 'Ergebnisse'. The Environment pane on the right shows the loaded datasets: 'radio' (275 obs. of 28 variables) and 'radio_weibli' (146 obs. of 28 variables).

```

85
86 # t-Test
87 t.test(Gesamtprogramm-Geschlecht, data = radio)
88 t.test(Musik-Geschlecht, data = radio)
89
90 # ANOVA
91 summary(aov(Gesamtprogramm-SenderCluster, data = radio))
92 mean(Gesamtprogramm-SenderCluster, data = radio)
93
94 summary(aov(Moderation-SenderCluster, data = radio))
95
96 # Shapiro-Wilk-Test
97 shapiro.test(radio$Alter)
98
99 (Top Level)

```

Console output:

```

mean in group männlich mean in group weiblich
77.19388                83.56849

> summary(aov(Gesamtprogramm-SenderCluster, data = radio))
Df Sum Sq Mean Sq F value Pr(>F)
SenderCluster  2  3206  1603.0  5.658 0.00391 **
Residuals    272  77058  283.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> mean(Gesamtprogramm-SenderCluster, data = radio)
Privat alt Privat jung      SWR
83.23077    76.96825    73.98810

> summary(aov(Moderation-SenderCluster, data = radio))
Df Sum Sq Mean Sq F value Pr(>F)
SenderCluster  2  3014  1506.9  2.337 0.0986 .
Residuals    272 175422  644.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

6.5 Shapiro-Wilk-Test**Verwendung:**

Der Shapiro-Wilk-Test **vergleicht** die **Werte einer Stichprobe** mit **normalverteilten Werten**, um zu überprüfen, ob eine Normalverteilung angenommen werden kann. **Bei kleinen Stichproben (ca. $n < 50$) ist die Normalverteilung Voraussetzung für die Anwendung des t-Tests.** Folgende **Fragestellung** liegt dem Shapiro-Wilk-Test zugrunde:

- Sind die Werte einer mindestens intervallskalierten Variable normalverteilt?

Die **Ergebnisse des Shapiro-Wilk-Test** werden, wie folgt, **interpretiert**:

- Ist der Test **nicht signifikant** ($p > .10$), **unterscheidet sich die Verteilung** der vorliegenden Stichprobe **nicht von einer normalverteilten Stichprobe** (→ **Normalverteilung**)
- Ist der Test **signifikant** ($p \leq .10$), **unterscheidet sich die Verteilung** der vorliegenden Stichprobe **von einer normalverteilten Stichprobe** (→ **keine Normalverteilung**)

Sollte bei einer kleinen Stichprobe der Shapiro-Wilk-Test ergeben, dass **keine Normalverteilung** vorliegt, ergibt sich die folgende **Konsequenz**:

- U.a. werden statt des t-Tests Rangverfahren wie Wilcoxon-/Mann-Whitney-Test angewendet (siehe nächstes Kapitel 6.6)
- Beachte: I.d.R. ist es sinnvoll, nicht nur den Shapiro-Wilk-Test durchzuführen, sondern auch einen Blick auf die (graphische) Verteilung zu werfen.

Schauen Sie sich nun das **Video-Tutorial** zum Shapiro-Wilk-Test an und lösen die darin gestellten Aufgaben mit R:



Aufgaben:

- Ist die Variable Alter normalverteilt?
- Ist die Variable Gesamtprogramm normalverteilt?

Benötigte Befehle:

Shapiro-Wilk test

```
shapiro.test(dataset$variable)
```

Relevante Variablen:

- Gesamtprogramm
- Alter

Lösung in R:

The screenshot shows the RStudio interface. The script editor on the left contains R code for an ANOVA and Shapiro-Wilk tests. The console on the bottom left shows the output of the Shapiro-Wilk tests. Two callouts highlight specific parts: 'R-Befehle' points to the test commands in the script, and 'Ergebnisse' points to the test results in the console.

```

89 # Anova
90 summary(aov(Gesamtprogramm~SenderCluster, data = radio))
91 mean(Gesamtprogramm~SenderCluster, data = radio)
92
93 summary(aov(Moderation~SenderCluster, data = radio))
94
95
96 # Shapiro-Wilk-Test
97 shapiro.test(radio$Alter)
98 shapiro.test(radio$Gesamtprogramm)
99
100 # Wilcoxon-Test
101 wilcox.test(Gesamtprogramm~Geschlecht, data = radio)
102
103 ## Top Level:
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Console output:

```

> # Shapiro-Wilk-Test
> shapiro.test(radio$Alter)

Shapiro-Wilk normality test

data:  radio$Alter
W = 0.87431, p-value = 3.079e-14

> shapiro.test(radio$Gesamtprogramm)

Shapiro-Wilk normality test

data:  radio$Gesamtprogramm
W = 0.91737, p-value = 3.418e-11

```

6.6 Wilcoxon-Test

Der Rangsummen-Test ist ein non-parametrisches Verfahren und vergleicht die *mittleren Rangzahlen* zweier Gruppen. (Es liegt keine Verteilungsannahme vor, u.a. Normalverteilung wurde auf Grund von Shapiro-Wilk-Test abgelehnt.) Folgende **Fragestellung** liegt dem Rangsummen-Test zugrunde:

- Gibt es einen „nicht-zufälligen“ statistischen Unterschied zwischen zwei Gruppen in Bezug auf ein mindestens ordinalskaliertes Merkmal?

Schauen Sie sich nun das **Video-Tutorial** zum Wilcoxon-Test an und lösen die darin gestellten Aufgaben mit R:



Aufgaben:

- Liegt nach dem Wilcoxon-Test ein signifikanter Unterschied zwischen Männern und Frauen bei der Beurteilung des Gesamtprogramms vor?
- Liegt nach dem Wilcoxon-Test ein signifikanter Unterschied zwischen Männern und Frauen bei der Beurteilung der Musik vor?

Benötigte Befehle:

Mann-Whitney & Wilcoxon test

```
wilcox.test(variable~grouping_variable, data = dataset)
```

(→Mann-Whitney test)

Relevante Variablen:

- Musik
- Gesamtprogramm
- Geschlecht

Lösung in R:

```

93 summary(aov(Moderation~SenderCluster, data = radio))
94
95 # Shapiro-Wilk-Test
96 shapiro.test(radio$Alter)
97 shapiro.test(radio$Gesamtprogramm)
98
99
100 # Wilcoxon-Test
101 wilcox.test(Gesamtprogramm~Geschlecht, data = radio)
102 wilcox.test(Musik~Geschlecht, data = radio)
103
104 # Korrelationsanalyse
105 cor.test(Musik~Moderation, data = radio)
104.22 (TopLevel) :

Console ~/
data: radio$Gesamtprogramm
W = 0.91737, p-value = 3.418e-11

> wilcox.test
> wilcox.test(Gesamtprogramm~Geschlecht, data = radio)

Wilcoxon rank sum test with continuity correction

data: Gesamtprogramm by Geschlecht
W = 8852, p-value = 0.03801
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(Musik~Geschlecht, data = radio)

Wilcoxon rank sum test with continuity correction

data: Musik by Geschlecht
W = 8227.5, p-value = 0.07007
alternative hypothesis: true location shift is not equal to 0

```

6.7 Korrelation

Die Korrelationsanalyse untersucht den (linearen) Zusammenhang zwischen zwei oder mehreren metrischen Variablen und beantwortet die folgende **Fragestellung**:

- Wie stark ist der lineare Zusammenhang zwischen den Variablen – und in welcher Richtung besteht er? → Wird der Wert von Variable A erhöht / gesenkt oder bleibt er gleich, wenn sich der Wert von Variable B ändert?

Eine konkrete Forschungsfrage könnte zum Beispiel lauten:

- Gibt es einen (signifikanten / nicht zufälligen) Zusammenhang zwischen Alter und Einkommen (bei diesem Test gibt es keine abhängigen und unabhängigen Variablen)?

Das **Vorgehen bei der Analyse** besteht aus zwei Schritten (vgl. auch Kap. 6.2 Chi2-Test und 6.3 t-Test):

- 1) Ist der p-value < 0.05 (< 0.01 , < 0.001)? → Signifikanz?
- 2) Wie hoch ist der Korrelationskoeffizient?

Eine **mögliche Interpretation des Korrelationskoeffizienten** finden Sie im Folgenden:

- $0.2 < |r| \leq 0.5$: geringe Korrelation
- $0.5 < |r| \leq 0.7$: mittlere Korrelation
- $0.7 < |r| \leq 1.0$: hohe Korrelation

Dabei ist das Vorzeichen des Korrelationskoeffizienten zu beachten:

- $0 < r \leq +1.0$: gleichgerichteter Zusammenhang
- $-1.0 \geq r > 0$: entgegengesetzter Zusammenhang

Im folgenden **Video** wird die Korrelationsanalyse genauer erklärt. Schauen Sie deshalb auf jeden Fall dieses Video und lösen Sie die darin gestellten Aufgaben:



Aufgaben:

- Gibt es einen nicht-zufälligen Zusammenhang zwischen der Beurteilung der Musik und der Moderation?
- Gibt es einen nicht-zufälligen Zusammenhang zwischen der Beurteilung von Comedy und Gewinnspielen?

Benötigte Befehle:

Correlation

```
cor.test(variable1~variable2, data = dataset)
```

Relevante Variablen:

- Musik
- Moderation
- Comedy
- Gewinnspiele

Lösung in R:

The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
102 wilcox.test(Musik-Geschlecht, data = radio)
103
104 # Korrelationsanalyse
105 cor.test(Musik-Moderation, data = radio)
106 cor.test(Comedy-Gewinnspiele, data = radio)
107
108 # Einfache Regression
109 lm.test(Musik-Moderation, data = radio)
110
111 (Top Level) :
```

A red box highlights the correlation analysis code (lines 104-106), with a callout labeled "R-Befehle". The console on the right shows the output of these commands:

```
> cor.test(Musik-Moderation, data = radio)

Pearson's product-moment correlation

data: Musik and Moderation
t = 7.1019, df = 273, p-value = 1.068e-11
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2901642 0.4902779
sample estimates:
cor
0.3948947

> cor.test(Comedy-Gewinnspiele, data = radio)

Pearson's product-moment correlation

data: Comedy and Gewinnspiele
t = 11.041, df = 273, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4680666 0.6323200
sample estimates:
cor
0.5555905
```

A red box highlights the console output, with a callout labeled "Ergebnisse". The Environment pane on the right shows the loaded data objects: "radio" (275 obs. of 28 variables) and "radio_weiblich" (146 obs. of 28 variables).

6.8 Regression

Die lineare Regression untersucht den linearen Einfluss einer metrischen Variablen (Einfachregression) oder mehrerer metrischer Variablen (Multiple Regression) auf eine abhängige metrische Variable. Dabei wird folgende **Fragestellung** beantwortet:

- Inwieweit kann der Wert einer abhängigen Variablen durch die Werte einer (Einfachregression) oder mehrere (Multiple Regression) unabhängigen Variablen erklärt werden?

Eine konkrete Forschungsfrage könnte zum Beispiel lauten:

- Wie stark beeinflussen die Werbeausgaben (unabhängige Variable) den Absatz eines Produktes (abhängige Variable)?

Das **Ergebnis der Analyse** ist die **Regressionsgerade** (Ergebnis der Analyse): $y = a + b \cdot x$

- y: Abhängige Variable („Wirkung“)
- x: Unabhängige Variable („Ursache“)
- a: y-Achsenabschnitt
- b: Regressionskoeffizient („um wieviel steigt y, wenn x sich um eine Einheit erhöht“)

Statistische Analyse („im Hintergrund“): Im Rahmen der Regressionsanalyse wird die Gerade ermittelt, bei der die quadrierten Abweichungen der gemessenen Werte von der Geraden am geringsten ist (Methode der kleinsten Quadrate)

Das **Vorgehen bei der Analyse** besteht aus drei Schritten:

- 1) Sind die p-values < 0.05 (< 0.01 , < 0.001)?
 - Liegt ein signifikanter Einfluss von x_i auf y vor (p-values)?
- 1) Wie hoch sind die Regressionskoeffizienten (b)?
 - Wie stark ist der Einfluss von x_i auf y?
- 2) Welchen Wert hat das Bestimmtheitsmaß (R^2)?
 - Wie hoch ist der Erklärungsgehalt des Modells?
 - R^2 = Anteil der Varianz der abhängigen Variablen, den die unabhängigen Variablen erklären

Schauen Sie sich nun das **Video-Tutorial** zur Regressionsanalyse mit einer genaueren Erklärung des Verfahrens sowie den Übungsaufgaben und deren Lösung in R an:



Aufgaben:

- Welchen Einfluss hat die Musik auf das Gesamtprogramm?
- Welchen Einfluss hat die Comedy auf das Gesamtprogramm?
- Welchen Einfluss haben die fünf Programmelemente auf das Gesamtprogramm?

Benötigte Befehle:**Linear regression**

```
summary(lm(dependent_variable~independent_variable, data =  
dataset))
```

Multiple regression

```
summary(lm(dependent_variable~ind_variable1 + ind_variable2,  
data = dataset))
```

Relevante Variablen:

- Berichterstattung
- Musik
- Moderation
- Comedy
- Gewinnspiele
- Gesamtprogramm

Lösung in R:

R-Befehle und Ergebnisse zu Aufgabe 1:

R-Befehle (Aufgabe 1)

```

107 # Einfache Regression
108 summary(lm(Gesamtprogramm~Musik, data = radio))
109 summary(lm(Gesamtprogramm~Comedy, data = radio))
110
111 # Multiple Regression
112 summary(lm(Gesamtprogramm ~ Musik + Berichterstattung + Moderation + Comedy + Gewinnspiele, data = radio))
113
114 # Hauptkomponentenanalyse
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Ergebnisse (Aufgabe 1)

```

> summary(lm(Gesamtprogramm~Musik, data = radio))

Call:
lm(formula = Gesamtprogramm ~ Musik, data = radio)

Residuals:
    Min       1Q   Median       3Q      Max
-71.132  -6.736   1.397   9.401  60.053

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.9474    3.4996   11.41  <2e-16 ***
Musik         0.4665    0.0421   11.08  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.24 on 273 degrees of freedom
Multiple R-squared:  0.3102,    Adjusted R-squared:  0.3077
F-statistic: 122.8 on 1 and 273 DF,  p-value: < 2.2e-16

> summary(lm(Gesamtprogramm~Comedy, data = radio))

Call:
lm(formula = Gesamtprogramm ~ Comedy, data = radio)

```

R-Befehle und Ergebnisse zu Aufgabe 2:

R-Befehle (Aufgabe 2)

```

107 # Einfache Regression
108 summary(lm(Gesamtprogramm~Musik, data = radio))
109 summary(lm(Gesamtprogramm~Comedy, data = radio))
110
111 # Multiple Regression
112 summary(lm(Gesamtprogramm ~ Musik + Berichterstattung + Moderation + Comedy + Gewinnspiele, data = radio))
113
114 # Hauptkomponentenanalyse
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Ergebnisse (Aufgabe 2)

```

> summary(lm(Gesamtprogramm~Musik, data = radio))

Call:
lm(formula = Gesamtprogramm ~ Musik, data = radio)

Residuals:
    Min       1Q   Median       3Q      Max
-63.176  -7.154   1.460   8.540  35.399

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.0085    1.64017   39.17  <2e-16 ***
Musik         0.28754    0.03072    9.36  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.92 on 273 degrees of freedom
Multiple R-squared:  0.243,    Adjusted R-squared:  0.2402
F-statistic: 87.61 on 1 and 273 DF,  p-value: < 2.2e-16

> summary(lm(Gesamtprogramm~Comedy, data = radio))

Call:
lm(formula = Gesamtprogramm ~ Comedy, data = radio)

```

R-Befehle und Ergebnisse zu Aufgabe 3:

R-Befehle (Aufgabe 3)

```

107 # Einfache Regression
108 summary(lm(Gesamtprogramm ~ Musik, data = radio))
109 summary(lm(Gesamtprogramm ~ Comedy, data = radio))
110
111 # Multiple Regression
112 summary(lm(Gesamtprogramm ~ Musik + Berichterstattung + Moderation + Comedy + Gewinnspiele, data = radio))
113
114 # Hauptkomponentenanalyse (PCA)
115
116

```

Ergebnisse (Aufgabe 3)

```

> summary(lm(Gesamtprogramm ~ Musik + Berichterstattung + Moderation + Comedy + Gewinnspiele, data = radio))

Call:
lm(formula = Gesamtprogramm ~ Musik + Berichterstattung + Moderation + Comedy + Gewinnspiele, data = radio)

Residuals:
    Min       1Q   Median       3Q      Max
-41.961  -5.098   0.189   5.538  51.239

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  23.99332    2.95161   8.129 1.59e-14 ***
Musik         0.27606    0.03585   7.877 8.34e-14 ***
Berichterstattung 0.24768    0.03220   7.691 2.76e-13 ***
Moderation    0.14962    0.03782   3.957 9.74e-05 ***
Comedy        0.07963    0.03112   2.558  0.0111 *
Gewinnspiele   0.01454    0.02600   0.559  0.5764

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

Residual standard error: 10.83 on 269 degrees of freedom
Multiple R-squared:  0.6872, Adjusted R-squared:  0.5999
F-statistic: 83.16 on 5 and 269 Df, p-value: < 2.2e-16

```

6.9 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse wird verwendet, um Fragen/Variablen/Indikatoren zu wenigen Dimensionen zusammenzufassen, und beantwortet die folgende **Fragestellung**:

- Können multidimensionale metrische Daten auf wenige wichtige Hauptkomponenten verdichtet werden?

Die **Grundidee** der Hauptkomponentenanalyse ist:

- Die Items einer Komponente korrelieren hoch miteinander.
- Die Items unterschiedlicher Komponenten korrelieren nicht hoch miteinander.

Die **Umsetzung der Hauptkomponentenanalyse** in R erfolgt in einem Vorbereitungsschritt und dann fünf Analyseschritten:

- 0) Vorbereiten der Datenbasis (neuer Datensatz)
- 1) Prüfung, ob eine Hauptkomponentenanalyse durchgeführt werden kann/soll mittels KMO (Kaiser-Meyer-Olkin-Test: misst den Anteil der Varianz zwischen Variablen, der auf gemeinsame Varianz zurückzuführen ist)

- 2) Bestimmung der Anzahl der Komponenten anhand der Eigenwerte
 - Auswahl: So viele Komponenten wie Eigenwerte > 1
- 3) Bestimmung der Faktorladungen (Korrelation eines Items mit einer Komponente) & Zuordnung der Items zu Komponenten
 - Voraussetzung: Ladung > 0.5
 - Entscheidung: Zuordnung des Items zu der Komponente, auf der es am höchsten lädt
- 4) Interpretation der Komponenten
- 5) Testen der einzelnen Komponenten mittels Cronbachs α (Reliabilität; durchschnittliche Korrelation aller Split-half-tests)

Im Folgenden **Video-Tutorial** wird die Hauptkomponentenanalyse im Detail und an einem Beispiel erklärt; ferner wird in dem Video auch die Übungsaufgabe gestellt und die dazugehörige Lösung in R erläutert:



Aufgaben:

- Zu welchen Komponenten lassen sich die Programmeigenschaften zusammenfassen? („Wie können aus den Programmeigenschaften neue übergeordnete Variablen gebildet werden?“)
- Wie ist die Reliabilität der neuen Komponenten zu beurteilen?

Benötigte Befehle:**Principal Component analysis** (package: psych)

```
dataset_new <- cbind(dataset$variable1, dataset$variable2...)
colnames(dataset_new) <- c("Item_1", "Item_2")
KMO(dataset_new)
pcaX <- princomp(dataset_new, scores = TRUE, cor = TRUE)
summary(pcaX)                plot(pcaX)
principal(dataset_new, nfactors = x, rotate = "varimax")
```

Cronbach's Alpha (package: psy)

```
factorX <- cbind(dataset$variable1, dataset$variable2...)
cronbach(factorX)
```

Relevante Variablen:

- Verlaesslicher_Service
- Informative_Nachrichten
- Unterhaltsame_Nachrichten
- Aktuelle_Musik
- Alte_Musik
- Anregende_Musik
- Beruhigende_Musik
- Informative_Moderation
- Unterhaltsame_Moderation
- Sympathische_Stimme
- Anspruchsvolle_Comedy
- Lustige_Comedy
- Spannende_Gewinnspiele
- Attraktive_Preise

Lösung in R entlang der fünf Schritte der Hauptkomponentenanalyse:

0) Vorbereiten der Datenbasis (neuer Datensatz)

R-Befehle

```

115 # Hauptkomponentenanalyse
116 pca_radio_data <- cbind(radio$Verlaesslicher_Service, radio$Informative_Nachrichten, radio$Unterhaltsame_Nachrichten, radio$Aktuelle_Musik)
117 colnames(pca_radio_data) <- c("Verlaesslicher_Service", "Informative_Nachrichten", "Unterhaltsame_Nachrichten", "Aktuelle_Musik")
118
119 KMO(pca_radio_data)
120
121 pca_radio_results <- princomp(pca_radio_data, scores = TRUE, cor = TRUE)
122 summary(pca_radio_results)
123
124 principal(pca_radio_data, nfactors = 4, rotate = "varimax")
125
126 # Cronbachs Alpha
127 cronbachs_alpha <- cbind(radio$Verlaesslicher_Service, radio$Informative_Nachrichten, radio$Unterhaltsame_Nachrichten, radio$Aktuelle_Musik)
128
129 Console:
Multiple R-squared: 0.6872, Adjusted R-squared: 0.5999
F-statistic: 83.16 on 5 and 269 Df, p-value: < 2.2e-16

> # Hauptkomponentenanalyse
> pca_radio_data <- cbind(radio$Verlaesslicher_Service, radio$Informative_Nachrichten, radio$Unterhaltsame_Nachrichten, radio$Aktuelle_Musik,
radio$Alte_Musik, radio$Anregende_Musik, radio$Beruhigende_Musik, radio$Informative_Moderation, radio$Unterhaltsame_Moderation,
radio$Sympathische_Stimme, radio$Anspruchsvolle_Comey, radio$Lustige_Comey, radio$Spannende_Gewinnspiele, radio$Attraktive_Preise)
> colnames(pca_radio_data) <- c("Verlaesslicher_Service", "Informative_Nachrichten", "Unterhaltsame_Nachrichten", "Aktuelle_Musik", "Alte_Musik",
"Anregende_Musik", "Beruhigende_Musik", "Informative_Moderation", "Unterhaltsame_Moderation", "Sympathische_Stimme", "Anspruchsvolle_Comey",
"Lustige_Comey", "Spannende_Gewinnspiele", "Attraktive_Preise")
>
> KMO(pca_radio_data)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = pca_radio_data)
Overall MSA = 0.85
MSA for each item =
  Verlaesslicher_Service Informative_Nachrichten Unterhaltsame_Nachrichten Aktuelle_Musik
1 0.82 0.78 0.92 0.89
2 Alte_Musik Anregende_Musik Beruhigende_Musik Informative_Moderation
3 0.89 0.73 0.70 0.90
4 Unterhaltsame_Moderation Sympathische_Stimme Anspruchsvolle_Comey Lustige_Comey
5 0.86 0.92 0.87 0.87
6 Spannende_Gewinnspiele Attraktive_Preise
7 0.80 0.80

```

1) Prüfung mittels KMO

R-Befehle

```

119 KMO(pca_radio_data)
120
121 pca_radio_results <- princomp(pca_radio_data, scores = TRUE, cor = TRUE)
122 summary(pca_radio_results)
123
124 principal(pca_radio_data, nfactors = 4, rotate = "varimax")
125
126 # Cronbachs Alpha
127 cronbachs_alpha <- cbind(radio$Verlaesslicher_Service, radio$Informative_Nachrichten, radio$Unterhaltsame_Nachrichten, radio$Aktuelle_Musik)
128
129 Console:
Multiple R-squared: 0.6872, Adjusted R-squared: 0.5999
F-statistic: 83.16 on 5 and 269 Df, p-value: < 2.2e-16

> # Hauptkomponentenanalyse
> pca_radio_data <- cbind(radio$Verlaesslicher_Service, radio$Informative_Nachrichten, radio$Unterhaltsame_Nachrichten, radio$Aktuelle_Musik,
radio$Alte_Musik, radio$Anregende_Musik, radio$Beruhigende_Musik, radio$Informative_Moderation, radio$Unterhaltsame_Moderation,
radio$Sympathische_Stimme, radio$Anspruchsvolle_Comey, radio$Lustige_Comey, radio$Spannende_Gewinnspiele, radio$Attraktive_Preise)
> colnames(pca_radio_data) <- c("Verlaesslicher_Service", "Informative_Nachrichten", "Unterhaltsame_Nachrichten", "Aktuelle_Musik", "Alte_Musik",
"Anregende_Musik", "Beruhigende_Musik", "Informative_Moderation", "Unterhaltsame_Moderation", "Sympathische_Stimme", "Anspruchsvolle_Comey",
"Lustige_Comey", "Spannende_Gewinnspiele", "Attraktive_Preise")
>
> KMO(pca_radio_data)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = pca_radio_data)
Overall MSA = 0.85
MSA for each item =
  Verlaesslicher_Service Informative_Nachrichten Unterhaltsame_Nachrichten Aktuelle_Musik
1 0.82 0.78 0.92 0.89
2 Alte_Musik Anregende_Musik Beruhigende_Musik Informative_Moderation
3 0.89 0.73 0.70 0.90
4 Unterhaltsame_Moderation Sympathische_Stimme Anspruchsvolle_Comey Lustige_Comey
5 0.86 0.92 0.87 0.87
6 Spannende_Gewinnspiele Attraktive_Preise
7 0.80 0.80

> pca_radio_results <- princomp(pca_radio_data, scores = TRUE, cor = TRUE)
> summary(pca_radio_results)
Importance of components:
Standard deviation 2.359856 1.212494 1.160619 1.007396 0.980609 0.844324 0.789766 0.738

```

Ergebnisse

2) Bestimmung der Anzahl der Komponenten anhand der Eigenwerte

R-Befehle

```

121 pca_radio_results <- prcomp(pca_radio_data, scores = TRUE, cor = TRUE)
122 summary(pca_radio_results)
123
124 principal(pca_radio_data, nfactors = 4, rotate = "varimax")
125
126 # Cronbachs Alpha
127 cronr(Unterhaltsame_Moderation, Sympathische_Stimme, Anspruchsvolle_Comey, Lustige_Comey,
128       (Top Level))

```

Ergebnisse

```

> principal(pca_radio_data, nfactors = 4, rotate = "varimax")
Principal Components Analysis
Call: principal(r = pca_radio_data, nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
      RC1  RC4  RC3  RC2  h2  u2  com
Verlaesslicher_Service  0.16  0.78  0.18 -0.10 0.67 0.33 1.2
Informative_Nachrichten 0.12  0.90  0.03  0.02 0.83 0.17 1.0
Unterhaltsame_Nachrichten 0.48  0.47  0.04  0.35 0.58 0.42 2.8
Aktuelle_Musik          0.19 -0.02 -0.02  0.77 0.62 0.38 1.1
Alte_Musik              0.29  0.16  0.65 -0.12 0.55 0.45 1.6
Anregende_Musik         0.00 -0.01  0.53  0.63 0.68 0.32 1.9
Beruhigende_Musik       0.01  0.09  0.84  0.18 0.75 0.25 1.1
Informative_Moderation  0.35  0.72  0.17  0.22 0.71 0.29 1.8
Unterhaltsame_Moderation 0.61  0.42 -0.01  0.44 0.74 0.26 2.7
Sympathische_Stimme     0.48  0.36  0.09  0.47 0.58 0.42 2.9
Anspruchsvolle_Comey    0.64  0.29  0.41  0.08 0.66 0.34 2.2
Lustige_Comey           0.71  0.19  0.22  0.16 0.62 0.38 1.5
Spannende_Gewinnspiele  0.83  0.11  0.10  0.05 0.71 0.29 1.1
Attraktive_Preise       0.81  0.13 -0.02  0.10 0.68 0.32 1.1

```

3) Bestimmung der Faktorladungen und

4) Interpretation der Komponenten

R-Befehle

```

124 principal(pca_radio_data, nfactors = 4, rotate = "varimax")
125
126 # Cronbachs Alpha
127 cronr(Unterhaltsame_Moderation, Sympathische_Stimme, Anspruchsvolle_Comey, Lustige_Comey,
128       (Top Level))

```

Ergebnisse

```

> principal(pca_radio_data, nfactors = 4, rotate = "varimax")
Principal Components Analysis
Call: principal(r = pca_radio_data, nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
      RC1  RC4  RC3  RC2  h2  u2  com
Verlaesslicher_Service  0.16  0.78  0.18 -0.10 0.67 0.33 1.2
Informative_Nachrichten 0.12  0.90  0.03  0.02 0.83 0.17 1.0
Unterhaltsame_Nachrichten 0.48  0.47  0.04  0.35 0.58 0.42 2.8
Aktuelle_Musik          0.19 -0.02 -0.02  0.77 0.62 0.38 1.1
Alte_Musik              0.29  0.16  0.65 -0.12 0.55 0.45 1.6
Anregende_Musik         0.00 -0.01  0.53  0.63 0.68 0.32 1.9
Beruhigende_Musik       0.01  0.09  0.84  0.18 0.75 0.25 1.1
Informative_Moderation  0.35  0.72  0.17  0.22 0.71 0.29 1.8
Unterhaltsame_Moderation 0.61  0.42 -0.01  0.44 0.74 0.26 2.7
Sympathische_Stimme     0.48  0.36  0.09  0.47 0.58 0.42 2.9
Anspruchsvolle_Comey    0.64  0.29  0.41  0.08 0.66 0.34 2.2
Lustige_Comey           0.71  0.19  0.22  0.16 0.62 0.38 1.5
Spannende_Gewinnspiele  0.83  0.11  0.10  0.05 0.71 0.29 1.1
Attraktive_Preise       0.81  0.13 -0.02  0.10 0.68 0.32 1.1

```

5) Testen der einzelnen Komponenten mittels Cronbachs α

The screenshot shows the RStudio interface with a script editor and a console. The script editor contains R code for calculating Cronbach's alpha for three components: `comp_Unterhaltung`, `comp_Information`, and `comp_Musik`. The console shows the output of these calculations.

R-Befehle

```
127 comp_Unterhaltung <- cbind(radioUnterhaltsame_Moderation, radioAnspruchsvolle_Comedy, radioLustige_Comedy, radioSpannende_Ge
128 comp_Information <- cbind(radioVerlaesslicher_Service, radioInformative_Nachrichten, radioInformative_Moderation)
129 comp_Musik <- cbind(radioAlte_Musik, radioAnregende_Musik, radioBeruhigende_Musik)
130 comp_Neue_Musik <- cbind(radioAktuelle_Musik)
131
132 cronbach(comp_Unterhaltung)
133 cronbach(comp_Information)
134 cronbach(comp_Musik)
135 cronbach(comp_Neue_Musik)
```

Ergebnisse

```
> cronbach(comp_Unterhaltung)
$sample.size
[1] 275

$number.of.items
[1] 5

$alpha
[1] 0.8511712

> cronbach(comp_Information)
$sample.size
[1] 275

$number.of.items
[1] 3

$alpha
[1] 0.8882828

> cronbach(comp_Musik)
```

7 Graphiken & Export von Ergebnissen nach Excel

Bei der Ergebnisdokumentation in einem Forschungsartikel, einer Abschlussarbeit oder einem Projektbericht ist es i.d.R. sinnvoll, Graphiken zu verwenden, um dem Leser eine leichtere Nachvollziehbarkeit zu ermöglichen. In den folgenden beiden Unterkapiteln lernen Sie, wie Sie Graphiken in R erstellen können und Ergebnisse nach Excel exportieren können, um dort Graphiken zu erstellen.

7.1 Graphiken in R

Im folgenden **Video** wird erläutert, wie Sie Ergebnisgraphiken in R erstellen:



Aufgaben:

- Erstellen Sie ein Säulendiagramm, das die relativen Häufigkeiten (%) der Variable Geschlecht wiedergibt.
- Erstellen Sie ein Säulendiagramm, dass die relativen Häufigkeiten (%) der Variable SenderCluster in Abhängigkeit vom Geschlecht wiedergibt.

Benötigte Befehle:

Bar graph – frequencies with 1 variable

```
table <- tally(~variable, format = "proportion", data = dataset)
frame <- as.data.frame(table)
View(frame)

ggplot(frame, aes(variable, Freq)) + geom_bar(position = "dodge", stat = "identity", fill = "#143C78") +
  scale_y_continuous(limits = c(min, max)) + geom_text(aes(label = sprintf("%.2f", frame$Freq)), position =
    position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
```

Bar graph – frequencies with 2 variables

```
table <- tally(variable_1~variable_2, format="proportion", data = dataset)
frame <- as.data.frame(table)
View(frame)

ggplot(frame, aes(variable_1, Freq, fill = variable_2)) + geom_bar(position = "dodge", stat = "identity") +
  scale_fill_manual(values = c("#143C78", "#CDEBFF")) + scale_y_continuous(limits = c(min, max)) + geom_text(aes(label =
    sprintf("%.2f", frame$Freq)), position = position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
```

Relevante Variablen:

- SenderCluster
- Geschlecht

Lösung in R:

R-Befehle und Ergebnisse zu Aufgaben 1:

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation and visualization.


```
# Häufigkeiten mit 2 Variablen
table_Geschlecht <- tally(~Geschlecht, format = "percent", data = radio)
frame_Geschlecht <- data.frame(table_Geschlecht)
View(frame_Geschlecht)

# Häufigkeiten mit 1 Variable
frame_GSC <- data.frame(tally(SenderCluster~Geschlecht, format = "percent", data = radio))
View(frame_GSC)
```
- Environment:** Shows the objects created in the workspace:
 - comp_Information: num [1:275, 1:3] 3 6 6 6 9 9 12 14 15 16 ...
 - comp_Musik: num [1:275, 1:3] 78 5 89 100 31 51 16 88 1...
 - comp_Neue_Musik: num [1:275, 1] 99 20 100 100 100 91 60 86 ...
 - comp_Unterhaltung: num [1:275, 1:5] 4 0 100 45 94 53 22 70 8 ...
 - frame_Geschlecht: 2 obs. of 2 variables
 - frame_GSC: 6 obs. of 3 variables
 - pca_radio_data: num [1:275, 1:14] 3 6 6 6 9 9 12 14 15 16 ...
 - pca_radio_resul: List of 7
 - radio: 275 obs. of 28 variables
 - radio_weiblich: 146 obs. of 28 variables
- Console:** Shows the execution of the following code:


```
> ggplot(frame_GSC, aes(SenderCluster, Freq, fill = Geschlecht)) +
+   geom_bar(position = "dodge", stat = "identity") +
+   scale_fill_manual(values = c("#143C78", "#CDEBFF")) +
+   scale_y_continuous(limits = c(0, 55)) +
+   geom_text(aes(label = sprintf("%0.2f", frame_GSC$Freq)),
+   position = position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
> # Graphiken - Häufigkeiten
> # Häufigkeiten mit 1 Variable
> table_Geschlecht <- tally(~Geschlecht, format = "percent", data = radio)
> frame_Geschlecht <- data.frame(table_Geschlecht)
> View(frame_Geschlecht)
>
> ggplot(frame_Geschlecht, aes(Geschlecht, Freq)) +
+   geom_bar(position = "dodge", stat = "identity", fill = "#143C78") +
+   scale_y_continuous(limits = c(0, 70)) +
+   geom_text(aes(label = sprintf("%0.2f", frame_Geschlecht$Freq)),
+   position = position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
```
- Plots:** Displays a bar chart titled "Geschlecht" showing the frequency of responses for "männlich" and "weiblich". The y-axis is labeled "Freq" and ranges from 0 to 60. The x-axis is labeled "Geschlecht". The bars are dark blue. The frequency for "männlich" is 46.91 and for "weiblich" is 53.09.

Geschlecht	Freq
männlich	46.91
weiblich	53.09

R-Befehle und Ergebnisse zu Aufgabe 2:

The screenshot displays the RStudio interface. The top-left pane shows the R script editor with code for data manipulation and plotting. The top-right pane shows the Environment window with a list of objects. The bottom-left pane shows the Console output. The bottom-right pane shows a bar chart titled 'Ergebnisse'.

R-Befehle (R Commands):

```

155 geom_text(aes(label = sprintf("%0.2f", frame_Geschlecht$Freq)),
156           position = position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
157
158 # Häufigkeiten mit 2 Variablen
159 frame_GSC <- data.frame(tally(SenderCluster~Geschlecht, format = "percent", data = radio))
160 View(frame_GSC)
161
162 ggplot(frame_GSC, aes(SenderCluster, Freq, fill = Geschlecht)) +
163   geom_bar(position = "dodge", stat = "identity") +
164   scale_fill_manual(values = c("#143C78", "#CDEBFF")) +
165   scale_y_continuous(limits = c(0, 55)) +
166   geom_text(aes(label = sprintf("%0.2f", frame_GSC$Freq)),
167             position = position_dodge(width = 0.9), vjust = -0.25, colour = "black", size = 2.7)
168
169 # Export von Ergebnissen nach Excel
170 # Export von Häufigkeiten nach Excel
171 # (Top Level) :
  
```

Environment:

- comp_Information: num [1:275, 1:3] 3 6 6 9 9 12 14 15 16 ...
- comp_Musik: num [1:275, 1:3] 78 5 89 100 31 51 16 88 1 ...
- comp_Neue_Musik: num [1:275, 1] 99 20 100 100 100 91 60 86 ...
- comp_Unterhaltu.: num [1:275, 1:5] 4 0 100 45 94 53 22 70 8 ...
- frame_Geschlecht: 2 obs. of 2 variables
- frame_GSC: 6 obs. of 3 variables
- pca_radio_data: num [1:275, 1:14] 3 6 6 6 9 9 12 14 15 16 ...
- pca_radio_result: List of 7
- radio: 275 obs. of 28 variables
- radio_weiblich: 146 obs. of 28 variables

Values:

```

table_Geschlecht 'table' num [1:2(1d)] 46.9 53.1
  
```

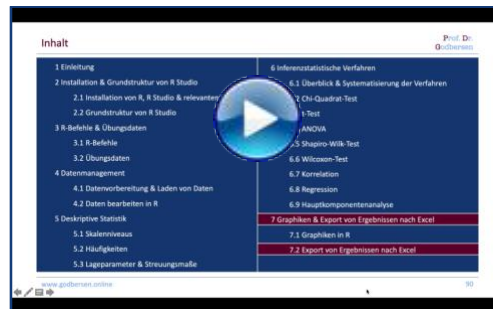
Ergebnisse (Results):

The bar chart shows the frequency (Freq) of SenderCluster (Private alt, Private jung, SWR) by Geschlecht (maennlich, weiblich). The y-axis ranges from 0 to 40. The x-axis categories are Private alt, Private jung, and SWR. The legend indicates that dark blue bars represent 'maennlich' (male) and light blue bars represent 'weiblich' (female).

SenderCluster	maennlich (Freq)	weiblich (Freq)
Private alt	20.16	26.71
Private jung	44.96	40.58
SWR	34.88	26.71

7.2 Export von Ergebnissen nach Excel

Im folgenden **Video** wird erläutert, wie Sie Analyseergebnisse aus R nach Excel exportieren können, um dort Ergebnisgraphiken zu erstellen:



Aufgaben:

- Exportieren Sie die relativen Häufigkeiten (%) der Variable SenderCluster in Abhängigkeit vom Geschlecht nach Excel.
- Exportieren Sie die Mittelwerte der Bewertungen der Programmelemente nach Excel.

Benötigte Befehle:

Export R results to Excel (package: openxlsx)

(1.1) Preparing frequencies

```
frame <- data.frame(tally(variable_1~variable_1, data = dataset))
```

(1.2) Preparing mean, median etc.

```
frame <- data.frame(new_variable_name_1 =  
c(mean(~variable_1, data = dataset), mean(~variable_2, data =  
dataset)), new_variable_name_2 = c(mean(~variable_3, data =  
dataset), mean(~variable_4, data = dataset)))
```

(2) Export to Excel

```
write.xlsx(frame, "Desktop/R_operations/name.xlsx", asTable =  
FALSE)
```

→ Please note: The name of the R frame & the directory / name of the xlsx-file have to be adjusted

Relevante Variablen:

- SenderCluster
- Berichterstattung
- Musik
- Moderation
- Comedy
- Gewinnspiele
- Gesamtprogramm
- Geschlecht

Lösung in R:

R-Befehle und Ergebnisse zu Aufgabe 1:

R-Befehle

```

172 frame_GSC_export <- data.frame(tally(SenderCluster~Geschlecht, format = "percent", data = radio))
173 View(frame_GSC_export)
174
175 write.xlsx(frame_GSC_export, "Desktop/R_operations/frame_GSC_export.xlsx", asTable = FALSE)
176
177 # Export von Mittelwerten nach Excel
178 frame_Programmelemente <- data.frame(Programmelement = c(
179   "Berichterstattung",
180   "Musik",
181   "Moderation",
182   "Comedy",
183   "Gewinnspiele"),
184   Mittelwert = c(
185     20.15504,
186     44.96124,
187     34.88372,
188     26.71233,
189     46.57534,
190     26.71233)
191   )
192 
```

Ergebnisse: generierte Excel-Datei

SenderCluster	Geschlecht	Freq
Privat alt	maennlich	20.15504
Privat jung	maennlich	44.96124
SWR	maennlich	34.88372
Privat alt	weiblich	26.71233
Privat jung	weiblich	46.57534
SWR	weiblich	26.71233

R-Befehle und Ergebnisse zu Aufgabe 2:

The screenshot displays the RStudio interface with three main panels. The top-left panel shows the R script editor with code for creating a data frame and calculating mean values. The top-right panel shows the Environment pane with a list of objects. The bottom-right panel shows a preview of the generated Excel file.

R-Befehle

```
175 write.xlsx(frame_GSC_export, "Desktop/R_operations/frame_GSC_export.xlsx", asTable = FALSE)
176
177 # Export von Mittelwerten nach Excel
178 frame_Programmelemente <- data.frame(Programmelement = c(
179   "Berichterstattung",
180   "Musik",
181   "Moderation",
182   "Comedy",
183   "Gewinnspiele"),
184   Mittelwert = c(
185     mean(-Berichterstattung, data = radio),
186     mean(-Musik, data = radio),
187     mean(-Moderation, data = radio),
188     mean(-Comedy, data = radio),
189     mean(-Gewinnspiele, data = radio)))
190
191 View(frame_Programmelemente)
192
193 write.xlsx(frame_Programmelemente, "Desktop/R_operations/frame_Programmelemente.xlsx", asTable = FALSE)
194
```

Environment

- comp_Information num [1:275, 1:3] 3 6 6 6 9 9 12 14 15 16 ...
- comp_Musik num [1:275, 1:3] 78 5 89 100 31 51 16 88 1 ...
- comp_Neue_Musik num [1:275, 1] 99 20 100 100 100 91 60 86 ...
- comp_Unterhaltu... num [1:275, 1:5] 4 0 100 45 94 53 22 70 8 ...
- frame_Geschlecht 2 obs. of 2 variables
- frame_GSC 6 obs. of 3 variables
- frame_GSC_export 6 obs. of 3 variables
- frame_Programme... 5 obs. of 2 variables
- pca_radio_data num [1:275, 1:14] 3 6 6 6 9 9 12 14 15 16 ...
- pca_radio_resul... List of 7

Ergebnisse: generierte Excel-Datei

Programmelement	Mittelwert
Berichterstattung	69.24727273
Musik	80.57818182
Moderation	67.06909091
Comedy	44.99272727
Gewinnspiele	36.57818182

Abschluss des Kurses und Nachwort

Sie sind am Ende des Kurses Angewandte Statistik & R angelangt. Wenn Sie die einzelnen Kapitel Schritt für Schritt durchgegangen sind und alle Übungsaufgaben lösen konnten, sollten Sie nun in der Lage sein, ein empirisches Forschungsprojekt erfolgreich auszuwerten.

Falls Sie Ihre neu gewonnenen Kenntnisse und Fähigkeiten festigen wollen, können Sie alle Übungsaufgaben noch einmal in R bearbeiten. Legen Sie auch hier den Fokus auf die beiden Fragen: Wann wende ich welches Verfahren an und wie interpretiere ich die Ergebnisse?

Der Autor würde sich über Ihr Feedback zu diesem Kurs freuen. Senden Sie dazu eine Email an: hendrik.godbersen@godbersen.online.

Ich wünsche Ihnen viel Erfolg bei Ihrem nächsten empirischen Projekt.

Ihr Prof. Dr. Hendrik Godbersen