

Statistik 1 – Tutorate

Einführung in R

Marco Giesselmann, Aurelia De Martinis, Alex Geistlich, Dominic Truxius, Nora Zumbühl

Heute:



Organisatorisches

Semesterplan

Vorbereitung

Literatur

Website



R-Basics

Was kann R?

Starten von RStudio

Arbeiten mit Skripten

R-Syntax Basics

Teil 1:

Organisatorisches



1.1

Semesterplan

Termine (Grau = Standardreferenz)

		Benninghaus	Gehring/Weins	Diaz-Bone	Jann	Tutorat (Folgewoche)
	Block 1: Grundlagen und univariate Analyseverfahren					
1.1	Einführung und Übersicht					
1.2	Vor der Statistik: Datenerhebung, Frage- und Variablentypen	1;2	3	2	2	R-Basics
1.3	Mittelwerte, Anteilswerte und Streuungsmasse	3	6.1	3-3.4	3.2	Datenintegration und -Inspektion I
1.4	Sozialstatistik und Ungleichheitsindikatoren		6.2	3.3-3.5, 3.8	3.2	Datenintegration und -Inspektion II
	Block 2: Inferenzstatistik für univariate Verteilungen					
1.5	Der Standardfehler		10; 11.2	6.3-6.5		Datenmanagement I
1.6	Konfidenzintervalle für Anteils- und Mittelwerte		10; 11.3	6.3-6.5		Datenmanagement II
1.7	Der p-Wert		12-12.2	7-7.1		Univariate Statistik I
1.8	Der Hypothesentest		12-12.2	7-7.1		Univariate Statistik II
	Block 3: Tabellenanalyse					
1.9	Bivariate Analyse: Grundlagen	4, 5.3	5.1.2; 7.1	4.1-4.2.4	4-4.1	Univariate Statistik III
1.10	Einfache Masse der Tabellenanalyse und sozialstrukturelle Anwendungen	5	7.3.2	4.2.5	4.2	Tabellenanalyse I
1.11	Tabellenmass Lambda					
1.12	Chi ² basierte Masse der Tabellenanalyse		7.1; 12.4	4.2.5; 7.4	4.2	Tabellenanalyse II
	Block 4: Mittelwertvergleiche					
1.13	Mittelwertdifferenz und t-test	8	12.3			Mittelwertvergleich
1.14	Puffer					
	<i>Schriftliche Prüfung (siehe VVZ)</i>					

1.2 Vorbereitung

☐ Haben alle R und R-Studio installiert?

- <https://posit.co/download/rstudio-desktop/>



☐ Alle automatisch im OLAT-Kurs registriert / Mail(s) bekommen?

☐ URL zur Website erreichbar (<https://www.suz.uzh.ch/dataforstat/>)?

☐ Support neben dem Tutorat:

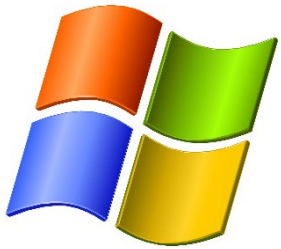
- Zentrale Informatik (<https://www.zi.uzh.ch/de/support.html>): Generelle Hardware & Software Probleme
- Methodenberatung SUZ (methodenberatung@soziologie.uzh.ch): Anwendungsprobleme R

☐ Fragen zur Organisation?

1.2

Vorbereitung/Installation von R und R-Studio

- <https://posit.co/download/rstudio-desktop/>
- Download R for ...



- [base](#)
- [Download R 4.0.0 for Windows](#)



- «Latest Release»: R-4.0.0.pkg
- Ebenso: [XQuartz](#)

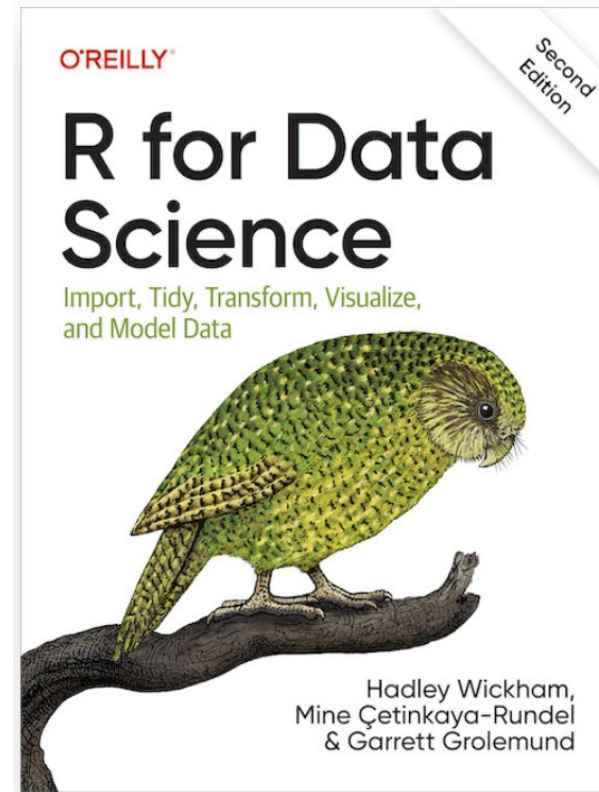
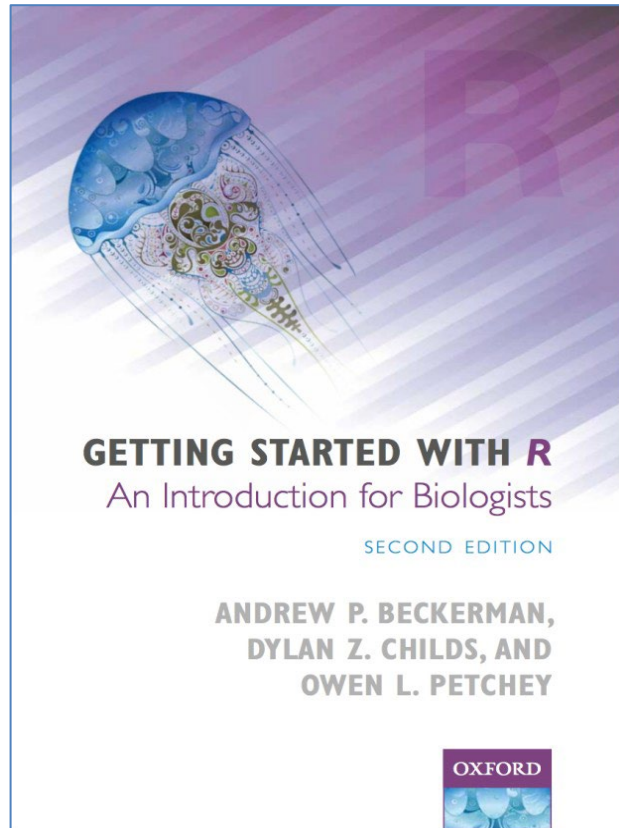
- Starten der Installation (wähle *alle Komponenten*)
- Starten von R-Studio! (und nicht von R...)
 - R ist das Betriebssystem und R-Studio ist die Nutzungsoberfläche

1.3 Webseite

<http://www.suz.uzh.ch/dataforstat/>

- Ziele:
 - Bereitstellung einheitlicher und tutoratsübergreifender Arbeitsressourcen
 - Optimierung der selbständigen Vor- und Nachbereitung
- Darauf findet ihr:
 - Links zu R-Literatur
 - Tutoratsfolien und R-Skripte
 - **Übungsaufgaben**
 - **Musterlösungen und Interpretation!**
- Achtung: **Work in Progress - wird regelmässig aktualisiert**





**Working With Descriptive and
Inferential Statistics in R**

Chair of Political Methodology
Institut für Politikwissenschaft
University of Zurich

Fall 2015

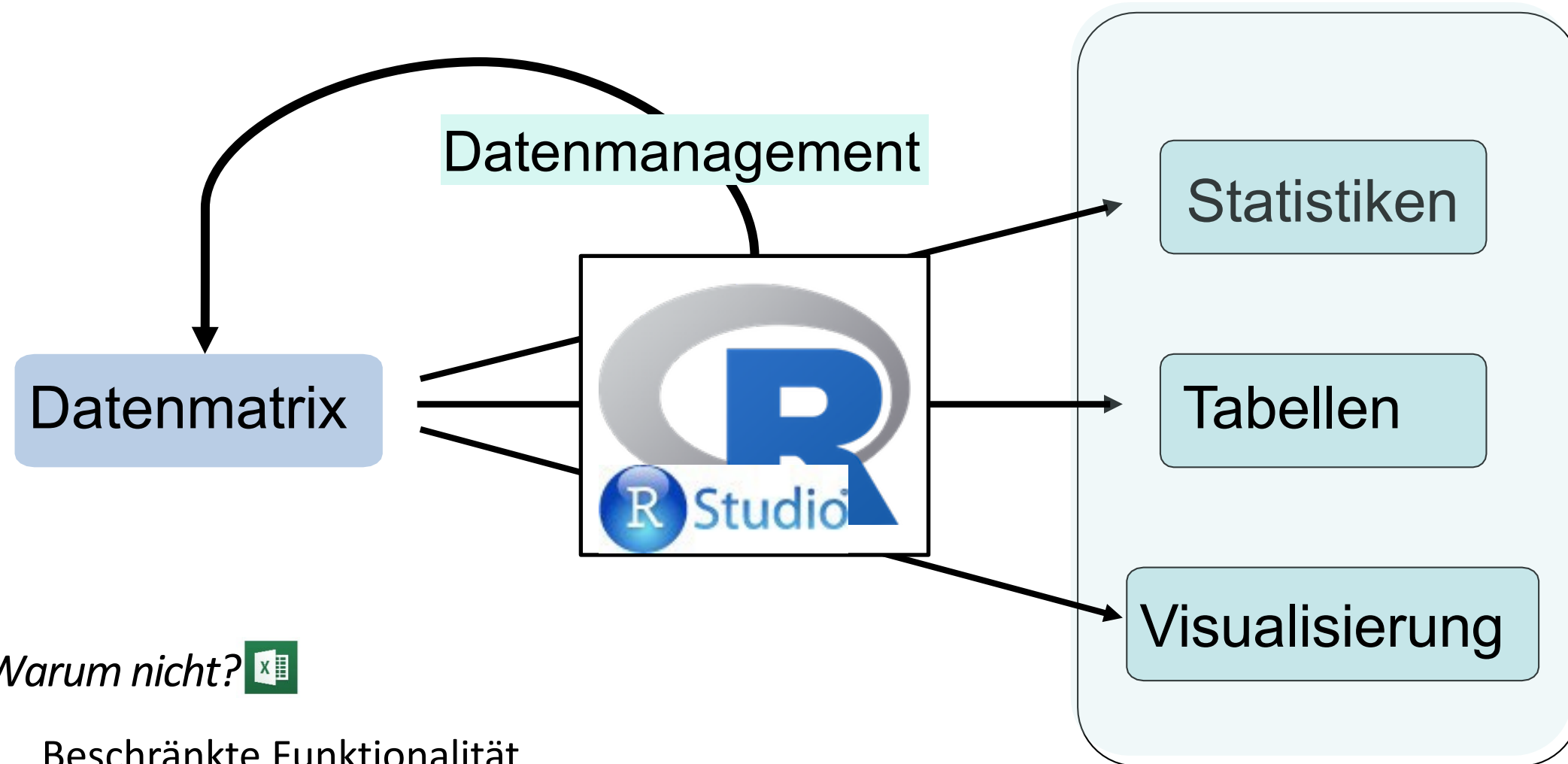
Die Links zur Literatur findet ihr auf der Webseite und/oder auf OLAT

Teil 2: R-Basics



2.1

Was kann R?

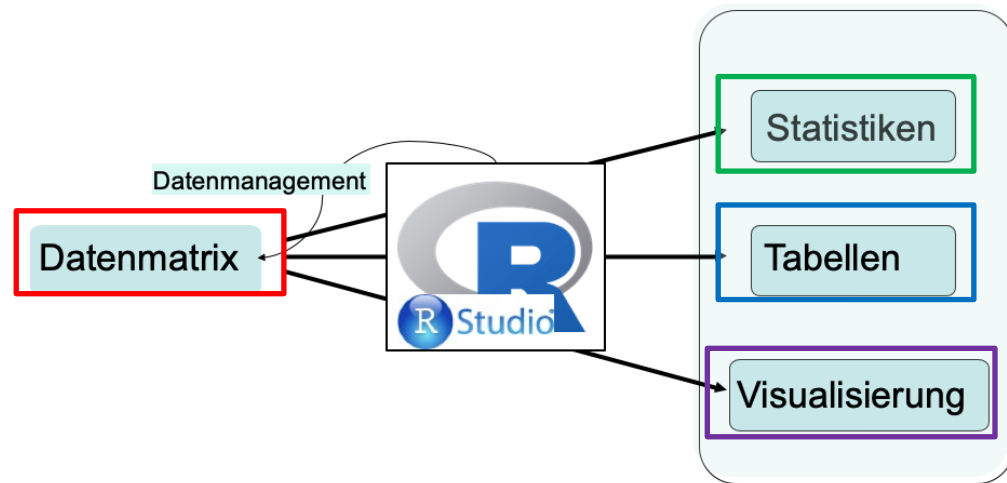


Warum nicht? 

- Beschränkte Funktionalität
- Keine Replikationsstandards

2.1

Was kann R?



	cntry Country	brncntr Born in country	trstplc Trust in the police
1	CH	1	5
2	CH	1	7
3	CH	1	9
4	CH	1	10
5	CH	1	2
6	CH	1	10
7	CH	1	9
8	CH	1	8
9	CH	1	6
10	CH	2	7
11	CH	1	9
12	CH	1	8
13	CH	2	9
14	CH	2	7

Two Sample t-test

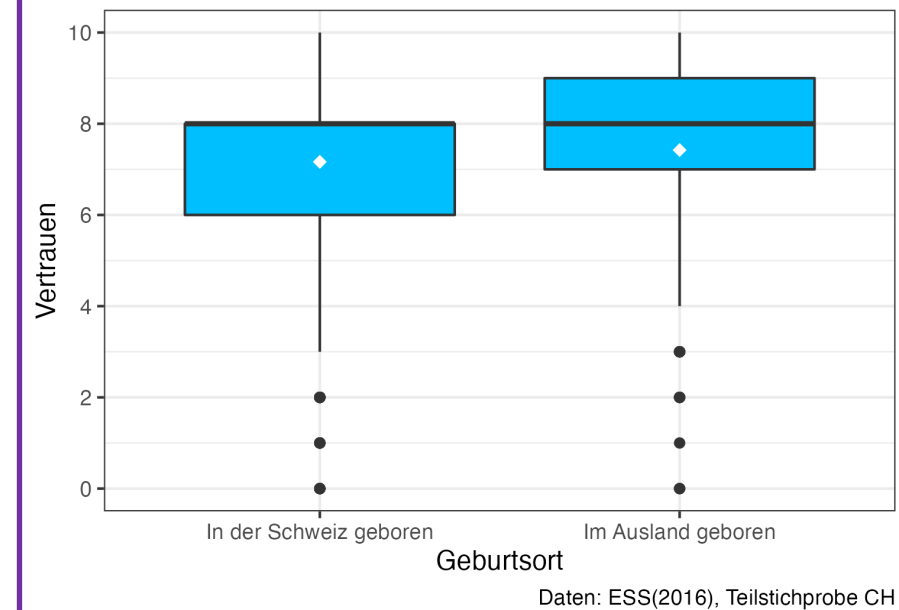
```

data: trstplc by brncntr_kat
t = 2.3356, df = 1516, p-value = 0.01964
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04144965 0.47621055
sample estimates:
mean in group Im Ausland geboren
                        7.423077
mean in group In der Schweiz geboren
                        7.164247
  
```

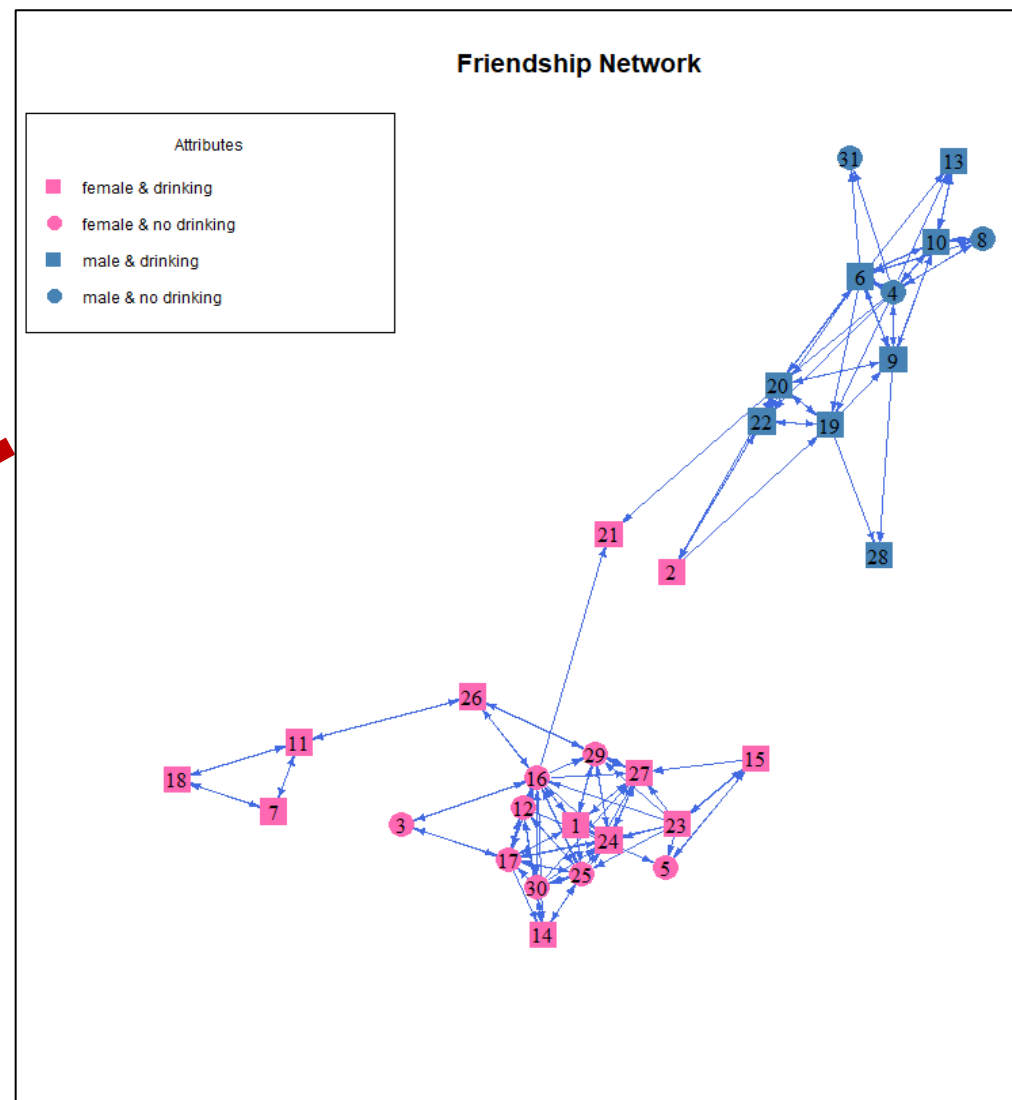
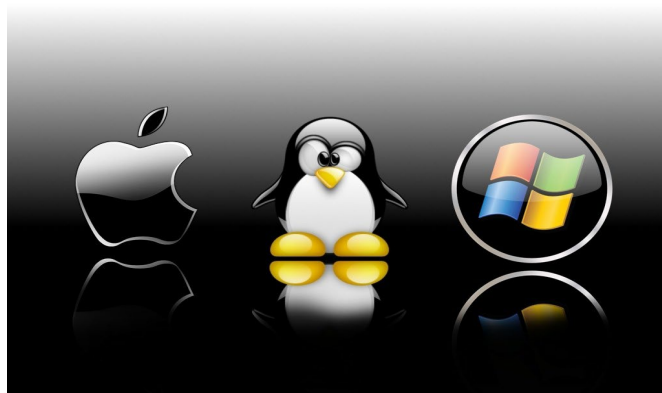
	Im Ausland geboren (N=421)	In der Schweiz geboren (N=1103)	Overall (N=1525)
Trust in the police			
Mean (SD)	7.42 (1.99)	7.16 (1.90)	7.24 (1.93)
Median [Min, Max]	8.00 [0, 10.0]	8.00 [0, 10.0]	8.00 [0, 10.0]
Missing	5 (1.2%)	1 (0.1%)	6 (0.4%)

Vertrauen in die Polizei und Geburtsort

10 = Volles Vertrauen, 0 = Gar kein Vertrauen



2.1 Was kann R? - Vorteile



2.1

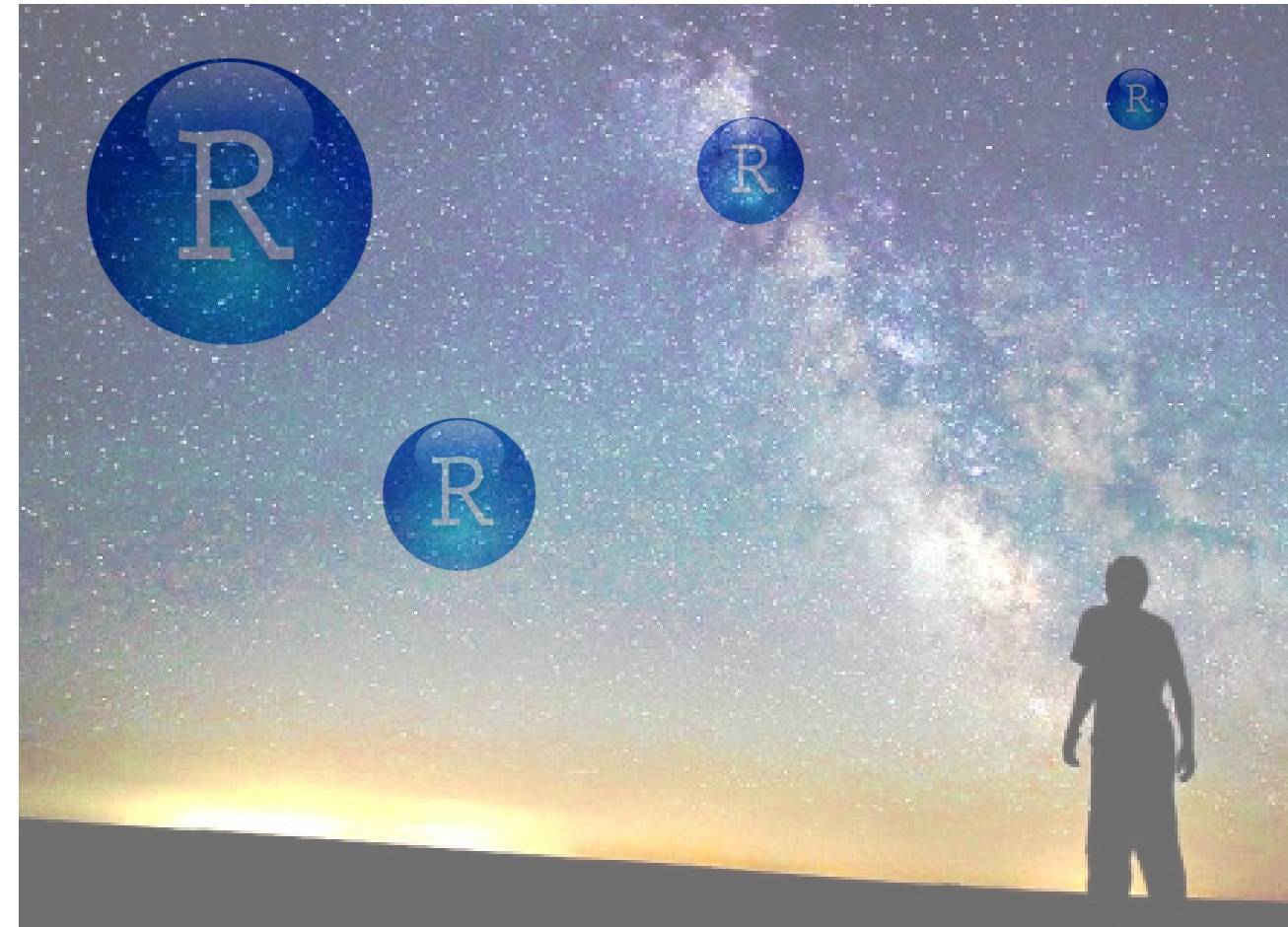
Was kann R? - Nachteile

- R ist ein Open Source Programm, daher:
 - Keine Befehlskonsistenz
 - Keine Funktionsgarantien
 - Befehlskonflikte
 - Fehlerdiagnose ist oft schwierig

You're not alone...

R ist ein Gemeinschaftsprojekt, zu dem zahllose Nutzerinnen und Nutzer weltweit aus den unterschiedlichsten Disziplinen beitragen. Es gibt unter anderem R-Konferenzen, das *The R Journal* und eine Reihe von Blogs, die euch während der Statistikausbildung eine grosse Hilfe sein können:

- [Stackoverflow](#)
- [Datascience+](#)
- [rdr.io](#)
- [R-Bloggers](#)
- [Quick-R](#)
- [\(... Google\)](#)
- [\(... ChatGPT\)](#)



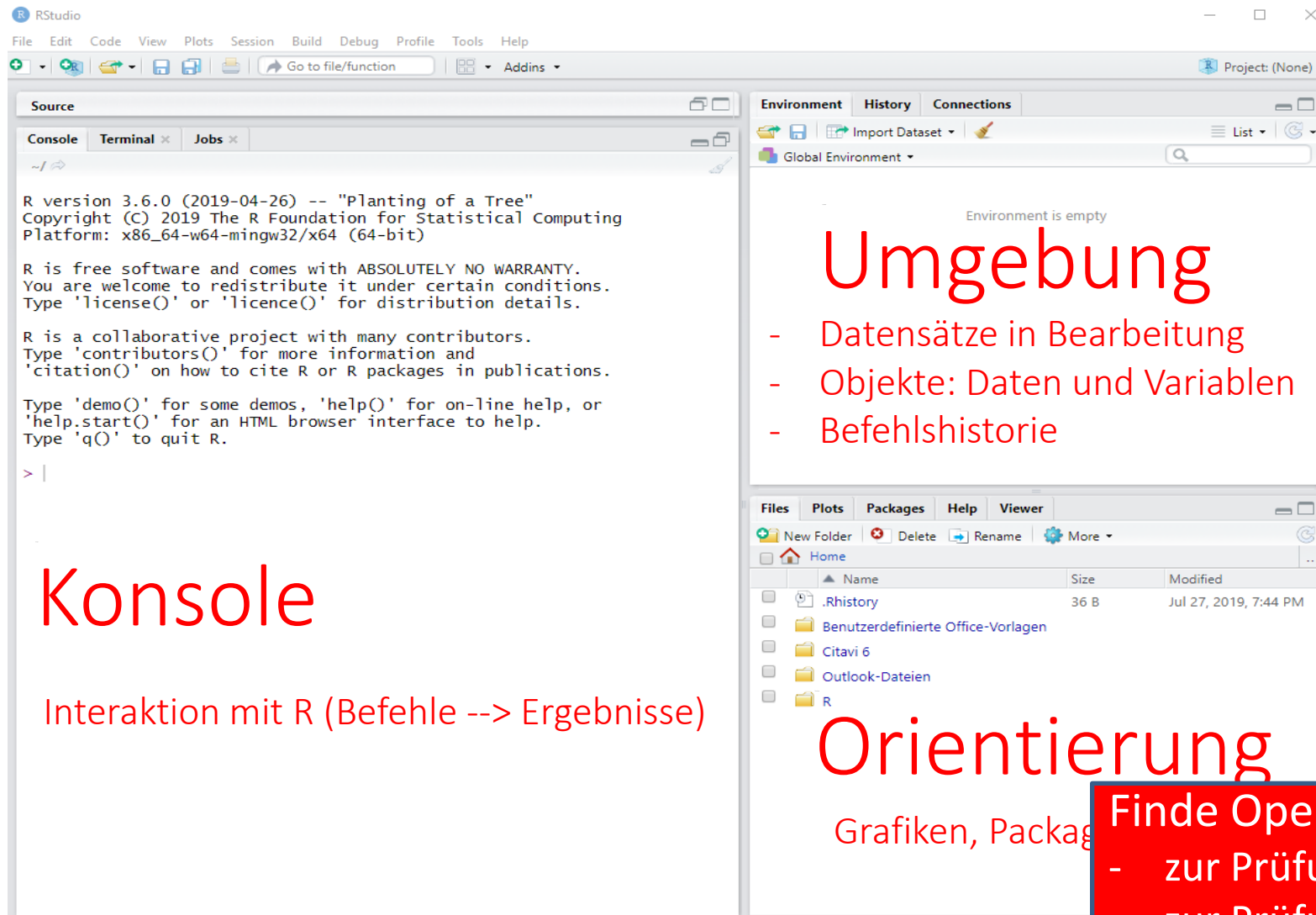
2.1

LLM (ChatGPT, ...) als Co-Pilot beim Coding

- LLMs erleichtern das Coding, im Rahmen des Tutorats zeigen wir euch sinnvolle Einsatzmöglichkeiten.
- Typische Einsätze von LLMs beim Coding:
 - Troubleshooting (Kopie von Code und Fehlermeldung nach ChatGPT)
 - Detailmodifikation z.B. bei Grafiken
 - ChatGPT als Tutorin (Codeerklärungen, Syntaxkonzeption, Funktionsrecherche)
- **Wichtig:**
 - **Starte stets mit Kontextprompts:**
 - ✓ Kreiere einen Standardprompt der ChatGPT mitteilt, was du studierst, in welchem Kontext du R typischerweise verwendest, auf welcher Lernstufe du stehst, auf welchem Niveau (Informatikprof, aufgeschlossener R-Beginner, Statistikphobiker, interessierte Grossmutter) Erklärungen angeboten werden sollen.
 - ✓ Kreiere zudem projektspezifische Prompts: Um welche Daten geht es, in welchem Rahmen sollen die Ergebnisse veröffentlicht oder präsentiert werden?
 - **Alle für eure Analysen genutzten Befehle und Befehlselemente – ob produziert durch euch, in Kooperationen mit anderen Personen oder mit Hilfe von ChatGPT – müssen stets vollständig verstanden sein und ihre Funktion erklärt werden können (Prüfungsanforderung)!**

2.2

Starten von R-Studio und Grundfunktionen



Konsole

Interaktion mit R (Befehle --> Ergebnisse)

Umgebung

- Datensätze in Bearbeitung
- Objekte: Daten und Variablen
- Befehlshistorie

Orientierung

Grafiken, Package

In die Konsole schreiben:

`1+1`

`3*3`

`4/6`

`2<8`

`2>8`

`2=8`

`2==8`

`2!=2`

`2!=8`

`2>8 | 2<8`

`2>8 & 2<8`

Was sind dies für Operationen?

Was bedeuten «&» und «|»

Finde Operationen...

- zur Prüfung, ob 3 mal 3 gleich 9 ist
- zur Prüfung, ob sowohl 3 mal 3 gleich 9 als auch 4 mal 4 gleich 16 ist
- zur Berechnung: Quadrat von 10
- zum Ziehen der Quadratwurzel aus 81

`3*3 == 9`

`3*3 == 9 & 4*4 == 16`

`10^2`

`sqrt(81)`

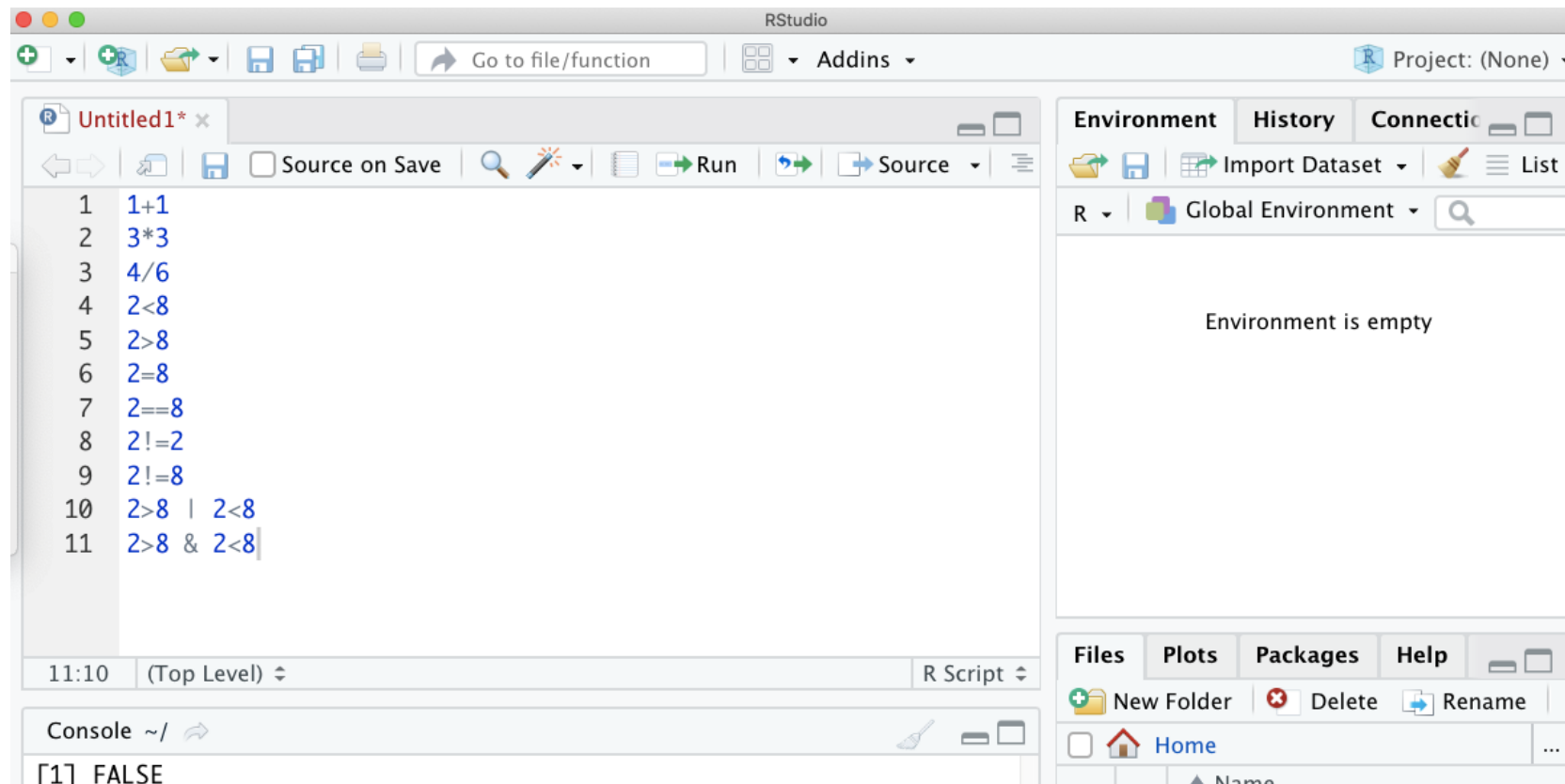
2.3 Arbeiten mit Skripten

Programmierung über die Konsole ist eher unüblich!

→ Befehlssequenzen sind nicht wiederherstellbar und kopierbar,

→ Analysen damit nicht mehr replizierbar oder modifizierbar!

Besser: Programmierung über Skripte.



2.3

Arbeiten mit Skripten

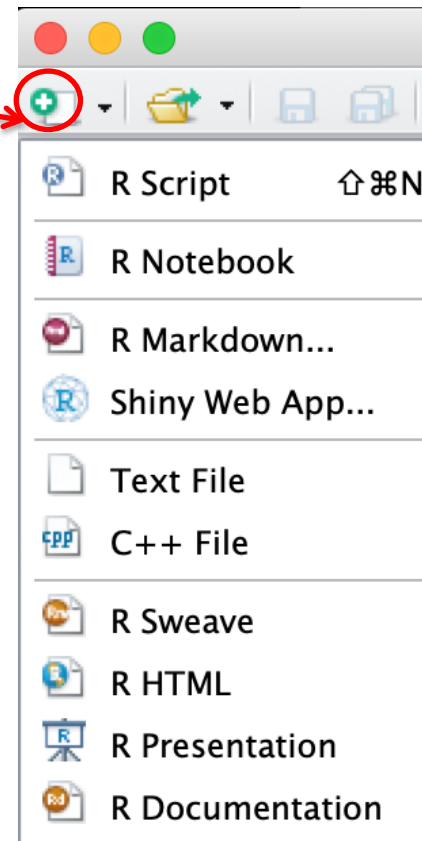
Programmierung über die Konsole ist eher unüblich!

→ Befehlssequenzen sind nicht wiederherstellbar und kopierbar,

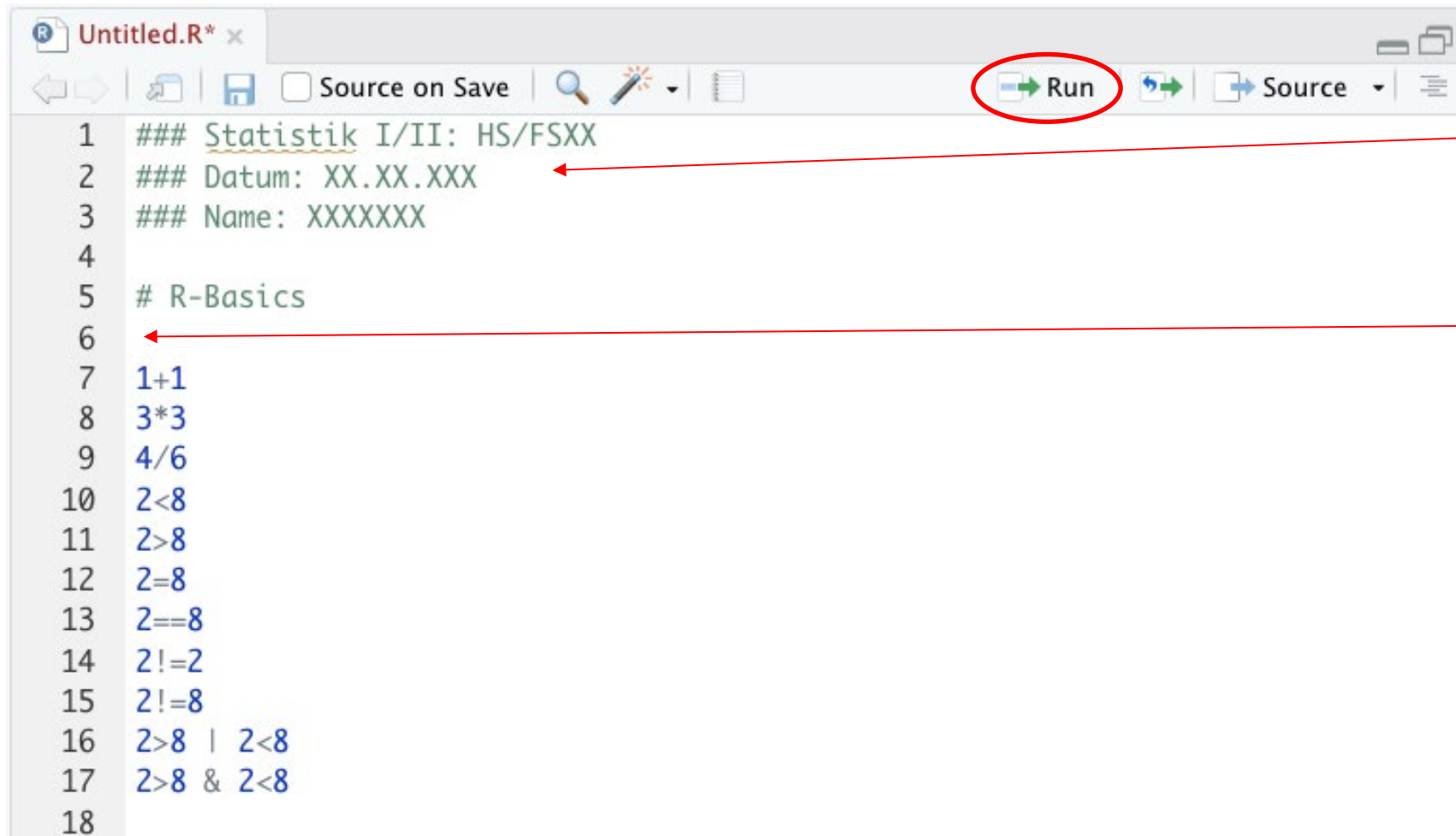
→ Analysen damit nicht mehr replizierbar oder modifizierbar!

Besser: Programmierung über Skripte.

...über File/New File/R Script



2.3 Arbeiten mit Skripten



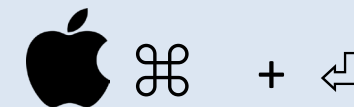
```
1  ### Statistik I/II: HS/FSXX
2  ### Datum: XX.XX.XXX
3  ### Name: XXXXXXXX
4
5  # R-Basics
6
7  1+1
8  3*3
9  4/6
10 2<8
11 2>8
12 2=8
13 2==8
14 2!=2
15 2!=8
16 2>8 | 2<8
17 2>8 & 2<8
18
```

Wichtig: **Beschriftung**
des Skriptes

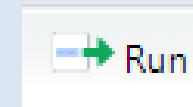
Tipp: **Leerzeilen**

Um einen Teil des Skripts
laufen zu lassen:

a) Markieren und...



b) Oder alternativ:



2.4 R-Syntax Basics: Wir generieren einen (Phantasie-) Datensatz

Erstellen von Wertelisten bzw. (Phantasie-)Variablen

```
# Personnummer
```

```
1:5
```

```
pid <- 1:5
```

```
pid
```

```
c(1, 2, 3, 4, 5)
```

```
pid2 <- c(1, 2, 3, 4, 5)
```

```
pid2
```

```
seq (from = 1, to = 5, by = 1)
```

```
pid3 <- seq (from = 1, to = 5, by = 1)
```

```
# Kanton
```

```
c("ZH", "BE", "LU", "UR", "SZ")
```

```
kid <- c("ZH", "BE", "LU", "UR", "SZ")
```

```
kid
```

- Beschreibe die Funktionalität von «:», «c» und «seq»
- Was macht die Anweisung «<-»
- Unterscheiden sich die Variablen pid, pid2 und pid3?
- Grundlegender Unterschied zwischen pid und kid?

Der Pfeil `<-` Zentrales Zuweisungssymbol in R

Objekt1 `<-` Objekt2

Bedeutung 1:

- Existiert bereits ein Objekt1, wird dieses durch Objekt2 überschrieben

Bedeutung 2:

- Existiert kein Objekt1, wird dieses als Kopie von Objekt2 neu angelegt

«Objekte» in R: Variablen/Wertelisten, Datensätze, Grafiken...

ChatGPT 4o 

3. **Alltagssprache:** Vielleicht könnte man als zusätzliche Eselsbrücke sagen, dass R „merkt“, ob es Objekt1 schon gibt. Wenn ja, dann wird es „aktualisiert“, ansonsten „erstellt“ es das Objekt.

Die Erwähnung, dass Objekte in R nicht nur Variablen, sondern auch Datensätze oder Grafiken umfassen können, ist ebenfalls wichtig. Dies hilft, den Studierenden die Breite der Anwendungsgebiete von Zuweisungen bewusst zu machen.

2.4 R-Syntax Basics: Wir generieren einen (Phantasie-) Datensatz

Erstellen von Wertelisten bzw. (Phantasie-)Variablen

```
# Personennummer
```

```
1:5
```

```
pid <- 1:5
```

```
pid
```

```
c(1, 2, 3, 4, 5)
```

```
pid2 <- c(1, 2, 3, 4, 5)
```

```
pid2
```

```
seq (from = 1, to = 5, by = 1)
```

```
pid3 <- seq (from = 1, to = 5, by = 1)
```

```
# Kanton
```

```
c("ZH", "BE", "LU", "UR", "SZ")
```

```
kid <- c("ZH", "BE", "LU", "UR", "SZ")
```

```
kid
```

Weitere (Phantasie-)Variablen...

```
# Körpergrösse Eltern
```

```
vg <- seq(from = 176, to = 184, by = 2)
```

```
mg <- seq(from = 171, to = 175, by = 1)
```

```
# Geburtsjahr und Befragungsjahr
```

```
yrbrn <- 2001:2005
```

```
date <- 2022
```

```
# Monatliches Einkommen in CHF
```

```
minc <- seq(from = 4500, to = 9000, by = 1000)
```

- Beschreibe die Funktionalität von «:», «c» und «seq»
- Was macht die Anweisung «<-»
- Unterscheiden sich die Variablen pid, pid2 und pid3?
- Grundlegender Unterschied zwischen pid und kid?

2.4 R-Syntax Basics: Wir generieren einen (Phantasie-) Datensatz

Variablenklassen

```
# Attribute der Variablen  
class(pid)  
class(minc)  
class(kid)  
kid_f<-as.factor(kid)  
class(kid_f)
```

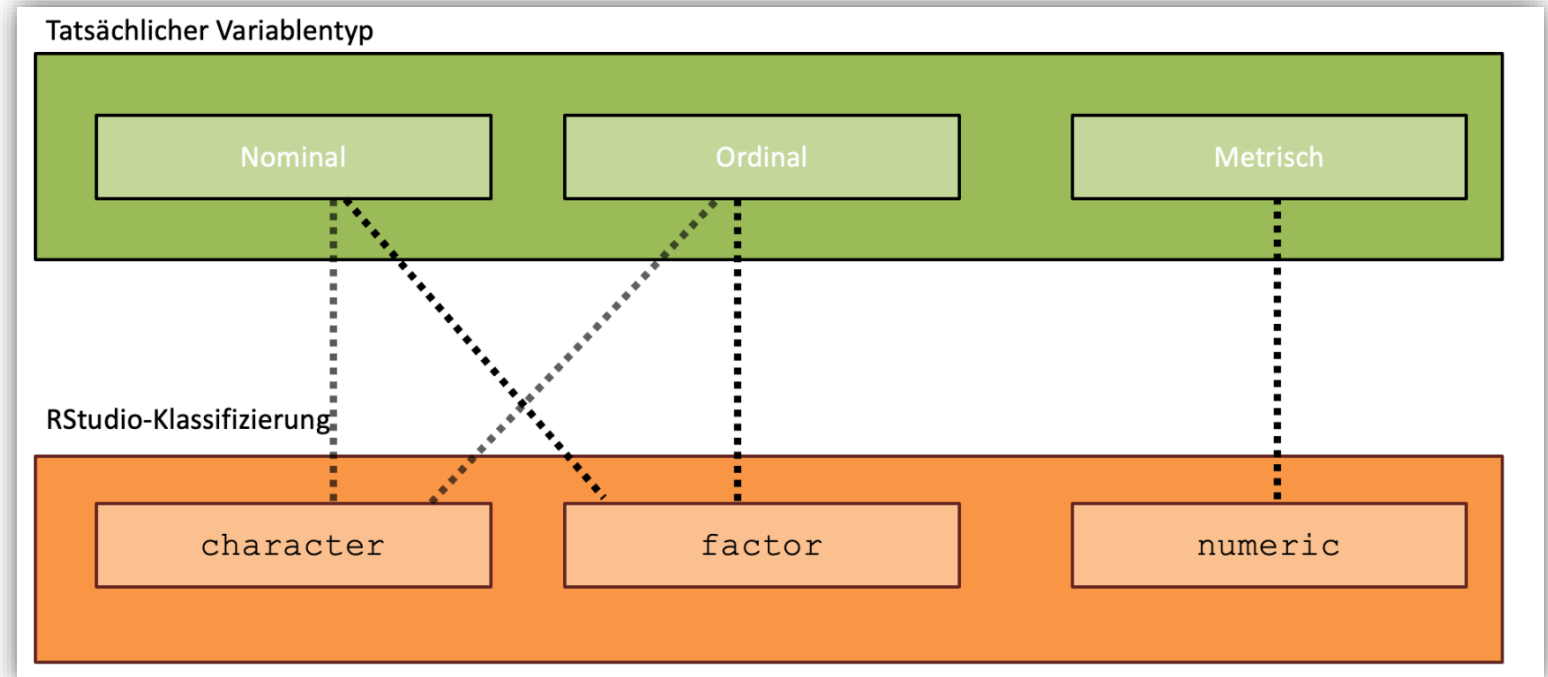
Was bedeuten «integer», «numeric»,
«character» und «factor»?

2.4 R-Syntax Basics: Wir generieren einen (Phantasie-) Datensatz

Variablenklassen

```
# Attribute der Variablen  
class(pid)  
class(minc)  
class(kid)  
kid_f<-as.factor(kid)  
class(kid_f)
```

Was bedeuten «integer», «numeric»,
«character» und «factor»?



- «factors» und «characters» repräsentieren beide kategoriale Variablen
- «factors» haben eine andere Speicherlogik als «characters» und sind für viele Anwendungen/Packages besser geeignet.
- «numerics» repräsentieren metrische Variablen
- «integers» sind eine Untergruppe der «numerics»

2.4 R-Syntax Basics: Wir generieren einen (Phantasie-) Datensatz

Variablenklassen

```
# Attribute der Variablen
class(pid)
class(minc)
class(kid)
kid_f<-as.factor(kid)
class(kid_f)
```

Was bedeuten «integer», «numeric»,
«character» und «factor»?

Rechnen mit Variablen

```
# Monatliches Einkommen in Jahreseinkommen
minc * 12
yinc <- minc * 12
yinc

# Alter zum Zeitpunkt der Befragung
age <- date - yrbrn
age

# Durchschnittliche elterliche Körpergrösse
pg <- (vg + mg)/2
pg
```

Wie verrechnet R Variablen?

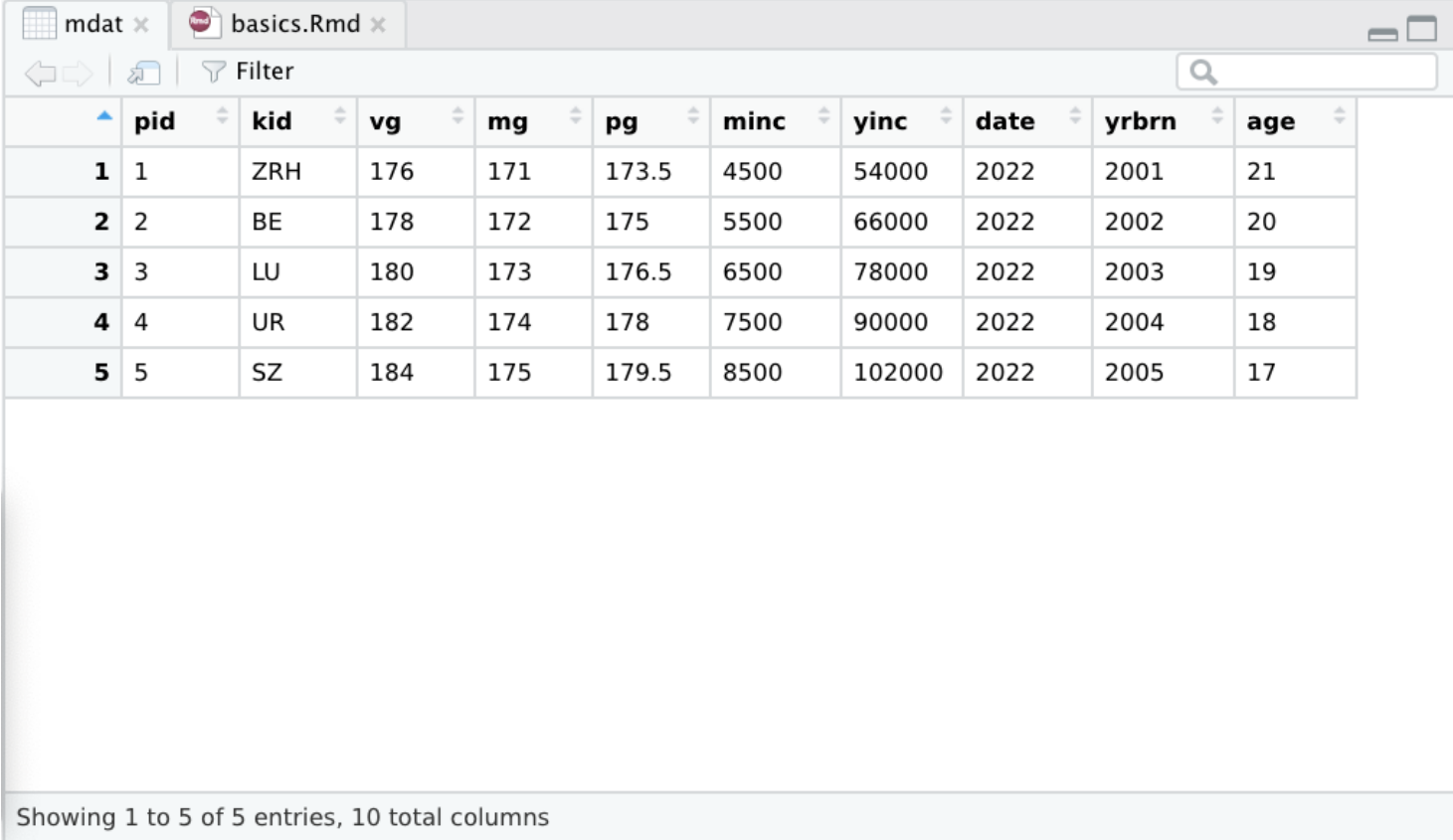
2.4

R-Syntax Basics

Verbindung von Variablen zu Datenmatrizen

```
m <- cbind(pid, kid, vg, mg, pg, minc, yinc, date, yrbrn, age)
class(m)
mdat <- as.data.frame(m)
class(mdat)
View(mdat)
```

- Wie wurde hier Information kombiniert?
- Beschreibe die Datenmatrix



	pid	kid	vg	mg	pg	minc	yinc	date	yrbrn	age
1	1	ZRH	176	171	173.5	4500	54000	2022	2001	21
2	2	BE	178	172	175	5500	66000	2022	2002	20
3	3	LU	180	173	176.5	6500	78000	2022	2003	19
4	4	UR	182	174	178	7500	90000	2022	2004	18
5	5	SZ	184	175	179.5	8500	102000	2022	2005	17

Showing 1 to 5 of 5 entries, 10 total columns

2.4

R-Syntax Basics

Verbindung von Variablen zu Datenmatrizen

```
m <- cbind(pid, kid, vg, mg, pg, minc, yinc, date, yrbrn, age)
class(m)
mdat <- as.data.frame(m)
class(mdat)
View(mdat)
```

- Wie wurde hier Information kombiniert?
- Beschreibe die Datenmatrix

Variablenmanagement in der Datenmatrix

```
class (mdat$minc)
mean (mdat$minc)
```

- Was macht «\$»?
- Was ist mit der Variable «minc» passiert, warum ist das ein Problem?
- Wie können wir sie wieder «numerisieren»?

	pid	kid	vg	mg	pg	minc	yinc	date	yrbrn	age
1	1	ZRH	176	171	173.5	4500	54000	2022	2001	21
2	2	BE	178	172	175	5500	66000	2022	2002	20
3	3	LU	180	173	176.5	6500	78000	2022	2003	19
4	4	UR	182	174	178	7500	90000	2022	2004	18
5	5	SZ	184	175	179.5	8500	102000	2022	2005	17

Showing 1 to 5 of 5 entries, 10 total columns

In meinem data frame in R namens "mdat" gibt es eine Variable "minc" die grade ein character ist, aber numeric sein soll. was tun?


Das ist ein sehr häufiger Fall – und leicht zu beheben. Hier sind die **sicheren und sauberen Wege**, je nachdem, wie die Variable aussieht.

1. Standardfall: Nur Zahlen als Text

Wenn `mdat$minc` aussieht wie `"1200"`, `"1500"`, `"900"` usw.:

r

```
mdat$minc <- as.numeric(mdat$minc)
```

 Copy code

Das reicht völlig aus.

Danach prüfen:

r

```
str(mdat$minc)
```

 Copy code

→ sollte `num` anzeigen.

2.4

R-Syntax Basics

Verbindung von Variablen zu Datenmatrizen

```
m <- cbind(pid, kid, vg, mg, pg, minc, yinc, date, yrbrn, age)
class(m)
mdat <- as.data.frame(m)
class(mdat)
View(mdat)
```

- Wie wurde hier Information kombiniert?
- Beschreibe die Datenmatrix

Variablenmanagement in der Datenmatrix

```
class (mdat$minc)
mean (mdat$minc)
```

- Was ermöglicht «\$»?
- Was ist mit der Variable «minc» passiert, warum ist das ein Problem?
- Wie können wir sie wieder «numerisieren»?

```
mdat$minc <- as.numeric(mdat$minc)
class(mdat$minc)
mean(mdat$minc)
```

	pid	kid	vg	mg	pg	minc	yinc	date	yrbrn	age
1	1	ZRH	176	171	173.5	4500	54000	2022	2001	21
2	2	BE	178	172	175	5500	66000	2022	2002	20
3	3	LU	180	173	176.5	6500	78000	2022	2003	19
4	4	UR	182	174	178	7500	90000	2022	2004	18
5	5	SZ	184	175	179.5	8500	102000	2022	2005	17

Showing 1 to 5 of 5 entries, 10 total columns

Achtung: Sollen bei der Umwandlung «factor» → «numeric» die Werte einer Variable erhalten bleiben, muss erst der Umweg über den «character» gegangen werden

2.4

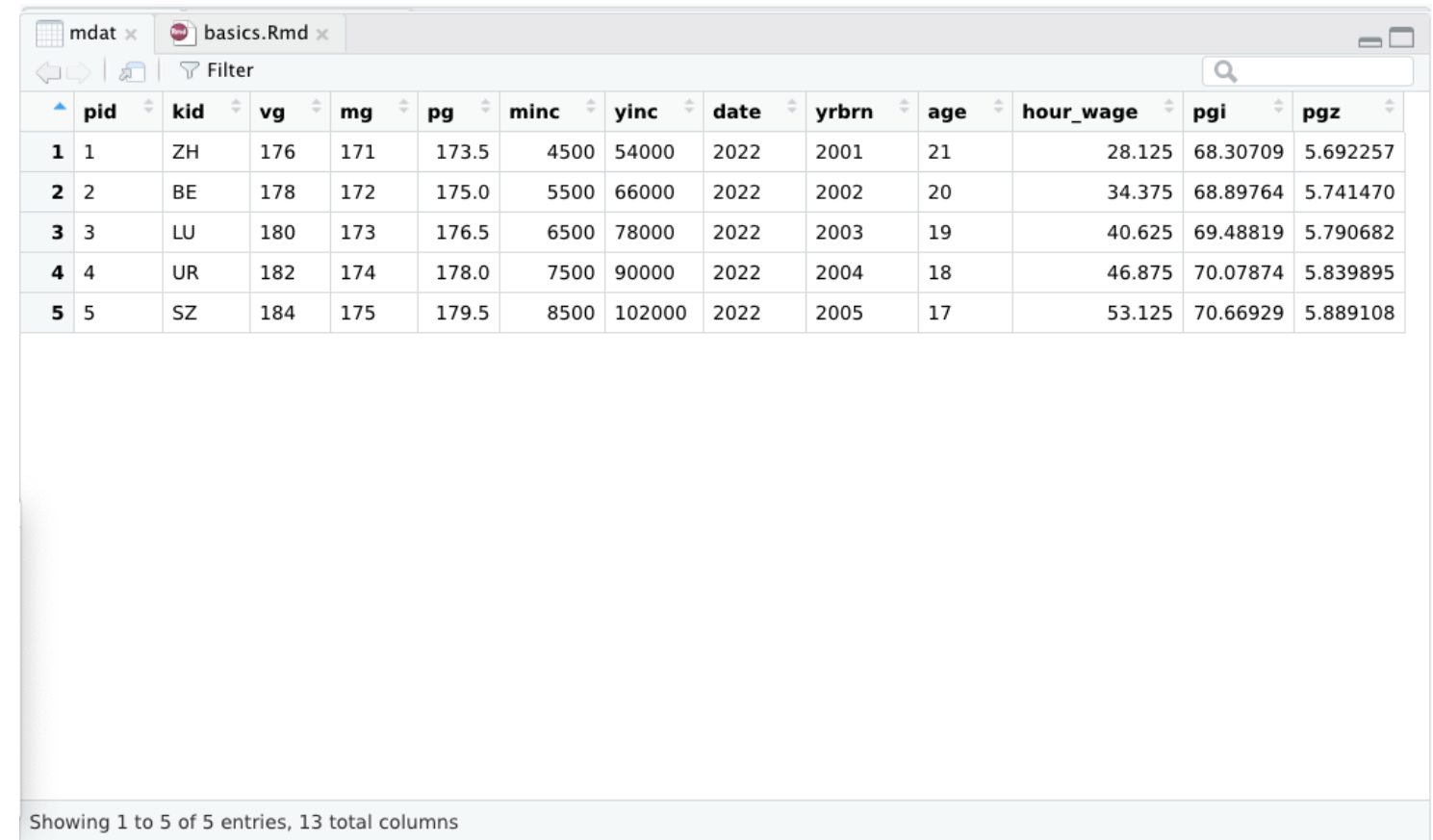
R-Syntax Basics

Variablenmanagement in der Datenmatrix

```
# Stundenlohn  
mdat$hour_wage <- mdat$minc/160  
mdat$hour_wage  
View(mdat)
```

Bilde nach gleichem Muster eine Variable, die die
elterliche Durchschnittsgrösse in Zoll angibt
(Umrechnungsquotient: 2.54)

```
# Grösse in Zoll  
mdat$pg <- as.character(mdat$pg)  
mdat$pg <- as.numeric(mdat$pg)  
mdat$pgi <- mdat$pg/2.54  
mdat$pgi  
View(mdat)
```



	pid	kid	vg	mg	pg	minc	yinc	date	yrbrn	age	hour_wage	pgi	pgz
1	1	ZH	176	171	173.5	4500	54000	2022	2001	21	28.125	68.30709	5.692257
2	2	BE	178	172	175.0	5500	66000	2022	2002	20	34.375	68.89764	5.741470
3	3	LU	180	173	176.5	6500	78000	2022	2003	19	40.625	69.48819	5.790682
4	4	UR	182	174	178.0	7500	90000	2022	2004	18	46.875	70.07874	5.839895
5	5	SZ	184	175	179.5	8500	102000	2022	2005	17	53.125	70.66929	5.889108

Showing 1 to 5 of 5 entries, 13 total columns

- **Variablen** sind **Wertelisten**
- Variablen bzw. Wertelisten sind üblicherweise eingebunden in Datenmatrixen
- Identische Positionen in den Variablen – und somit Zeilen in der Datenmatrix – repräsentieren Einheiten
- Datenmatrixen und Variablen können in unterschiedlichen Formaten gespeichert werden. Dementsprechend gibt es verschiedene Variablenklassen und Datenmatrix-Klassen
- Variablen und Datenmatrixen lassen sich je nach Bedarf zwischen den Klassen transformieren
- **<-** ist der **Zuweisungs-Operator**: Was links davon steht wird zu dem, was rechts davon steht
- nicht-numerischen Werte werden grundsätzlich in Anführungszeichen gesetzt.
- In der R-Sprache fungiert das Dollarzeichen als Trennzeichen zwischen Datensatz (steht vor **\$**) und Variable (steht nach **\$**)
- Interne Hilfe zu einem bestimmten Befehl: **«?»** + Befehl in die Eingabe (siehe nächste Folie)

2.5 Konventionen in R – Interne Hilfe

?seq

Sequence Generation

Description

Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for two common cases.

Arguments

<code>...</code>	arguments passed to or from methods.
<code>from, to</code>	the starting and (maximal) end values of the sequence. Of length 1 unless just <code>from</code> is supplied as an unnamed argument.
<code>by</code>	number: increment of the sequence.
<code>length.out</code>	desired length of the sequence. A non-negative number, which for <code>seq</code> and <code>seq.int</code> will be rounded up if fractional.
<code>along.with</code>	take the length from the length of this argument.

? + Befehl ruft die interne Hilfe in R mit Erläuterungen zum Befehl auf.

Generelle Beschreibung für welchen Zweck der Befehl verwendet werden kann.

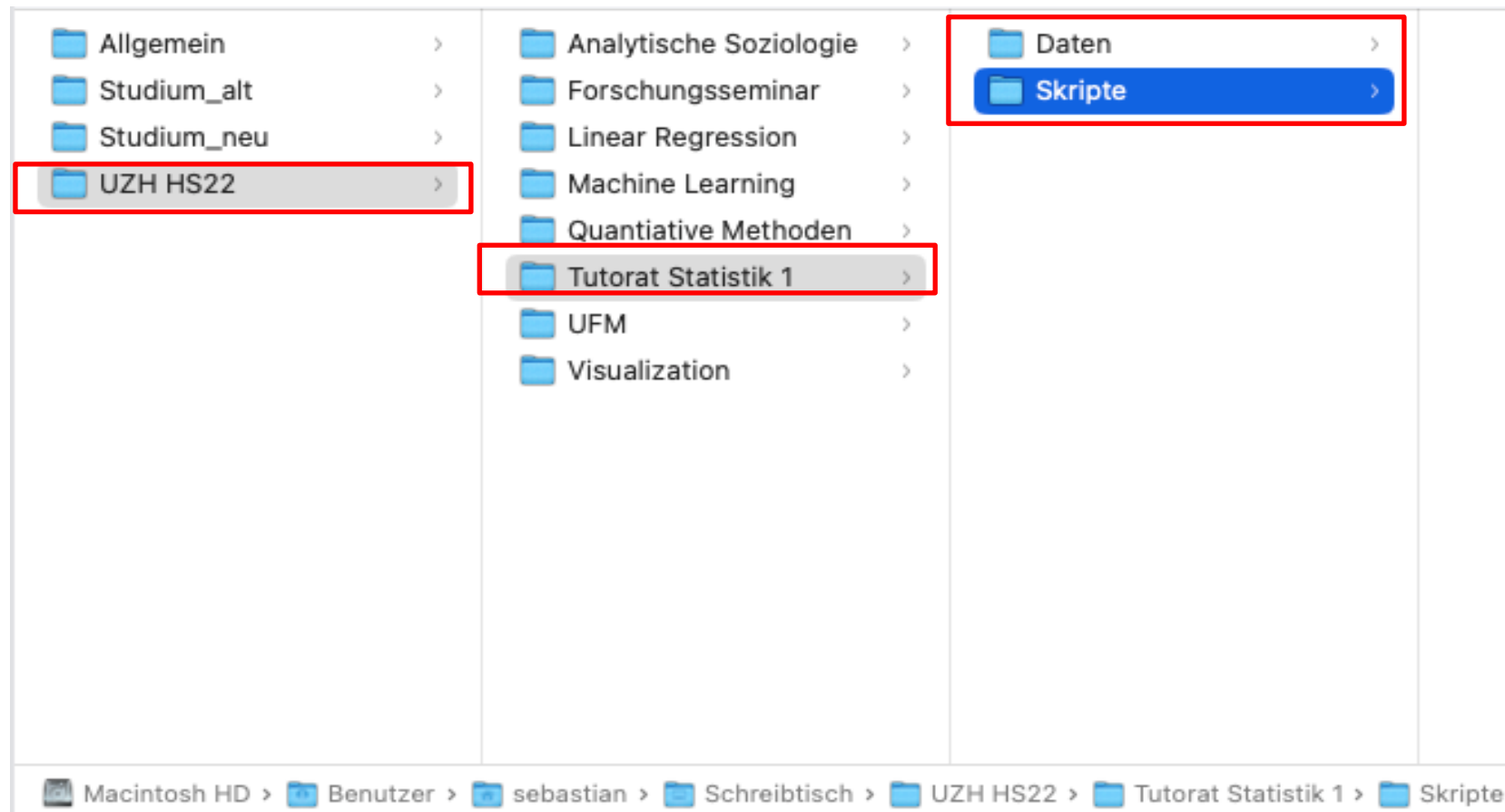
Unter «Argumente» kann man sehen, aus welchen Bestandteilen sich ein Befehl zusammensetzt.

Gleichwohl: Selbst die Logik von einfachen Befehlen wie »seq« ist mittels interner Hilfe oft schwierig zu verstehen. Es lohnt sich aber, sich mit dieser «Grundarchitektur» von R auseinanderzusetzen

2.5

Konventionen in R – Speichern von Skripten

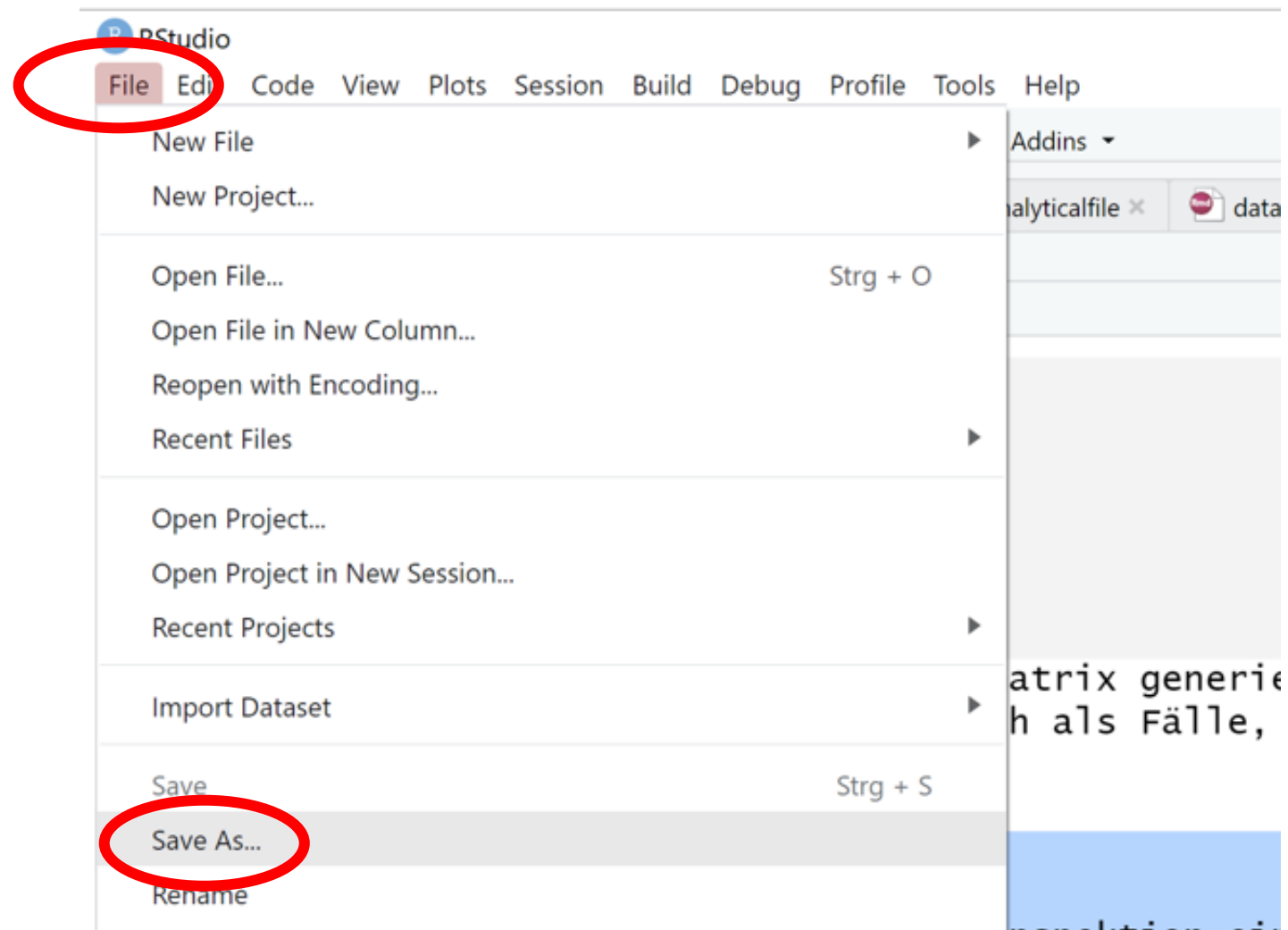
Schritt 1: lege im Veranstaltungs- bzw. Projektordner **Unterordner für deine Skripte** sowie einen **Datenordner** an



2.5

Konventionen in R – Speichern von Skripten

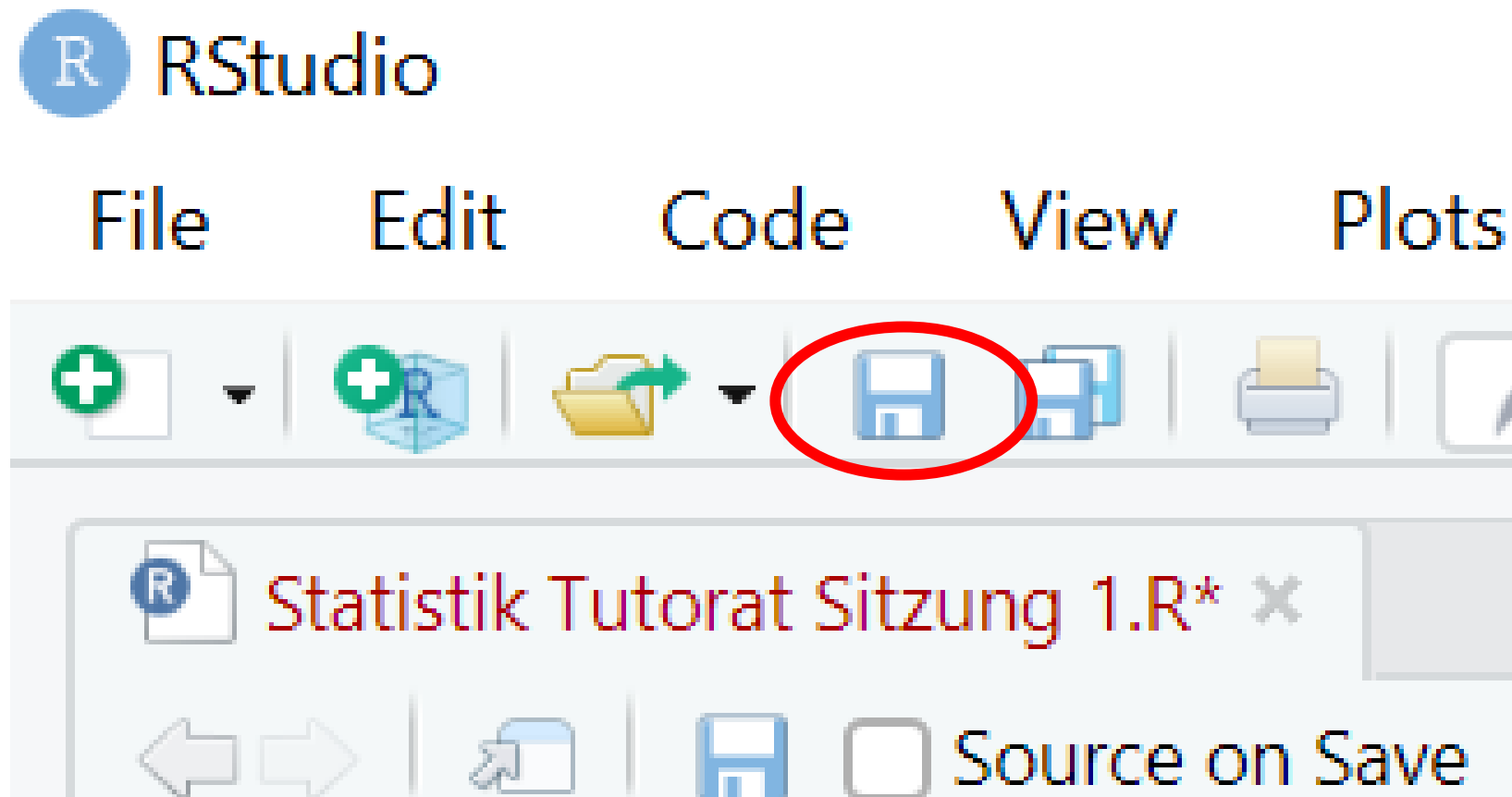
Schritt 2: Nach einer Sitzung solltet ihr die aktualisierten bzw. neu erstellten **Skripte** stets abspeichern.



2.5

Konventionen in R – Speichern von Skripten

Schritt 3: Bei Anlage einer Working Directory (kommt später) oder wiederholtem Speichern auch einfach über das «Speicher»-Symbol...





Fragen?



Ihr findet alle Materialien & Infos der heutigen Sitzung auf der Webseite



<http://www.suz.uzh.ch/dataforstat/index.html>