Computergestützte Datenanalyse: DATA-Übung mit R

Tag 1 – 17.07.2025

WER BIN ICH?

1. **Forscherin am Center for Advanced Internet Studies (CAIS) in Teams:**
   - Research Data & Methods
     - Forschung zu Methoden, insbesondere Computational Social Science
     - Beratungsangebote und Workshop Organisation

   - Societal and Computational Perspectives on Public Discourses Politische Soziologie und Forschungsethik
     - Migration, Polarisierung, Rassismus, Hassrede, usw.

2. **Doktorandin an der Universität Wien am Computational CommScience Lab**
   - Automatisierte Analyse von Rassismus in Texten – Chancen und Herausforderungen

WER BIN ICH?

1. **Forscherin am Center for Advanced Internet Studies (CAIS) in Teams:**
   - Research Data & Methods
     - Forschung zu Methoden, insbesondere Computational Social Science
     - Beratungsangebote und Workshop Organisation

   - Societal and Computational Perspectives on Public Discourses Politische Soziologie und Forschungsethik
     - Migration, Polarisierung, Rassismus, Hassrede, usw.

2. **Doktorandin an der Universität Wien am Computational CommScience Lab**
   - Automatisierte Analyse von Rassismus in Texten – Chancen und Herausforderungen

3. **Externe Lehrkraft an der Heinrich-Heine-Universität Düsseldorf**

ahrabhi.kathirgamalingam@cais-research.de

VERANSTALTUNGSZIELE

- Grundlegendes Verständnis der Programmiersprache R, z.B. Skripte, Packages und Funktionen
- Umgang mit Ressourcen für das Selbststudium
- Organisation der Datenanalyse
- Selbstständige Bearbeitung einer empirischen Fragestellung, d.h. Datenimport, -aufbereitung, -visualisierung und –analyse

WER SEID IHR?

1. **Untereinander vorstellen mit Sitznachbar*in (5 MIN)**
   - Erwartungen
   - Befürchtungen
   - Vorerfahrungen

2. **Kurze Vorstellrunde im Plenum**
   - Name, (optional: Pronomen), Studiengang
   - Erfahrungen, Erwartungen, Befürchtungen
   - Irgendetwas, worauf ich achten kann?

Erwartungen
vs. Befürchtungen

HERAUSFORDERUNGEN

- Heterogene Gruppe
- Kaum Wiederholung statistischer Grundlagen
- Langsamer Einstieg, steile Lernkurve

DIE NÄCHSTEN VIER TAGE

- **Tag 1**
  - Einführung in R und RStudio
  - „Basics“:
  - Coding Konventionen
  - Objekte, Datenimport & Co
- **Tag 2**
  - Skalenniveau
  - Troubleshooting
  - Datenaufbereitung
  - Datenvisualisierung
  - Deskriptive Statistik
- **Tag 3**
  - Inferenzstatistik I
  - Bivariate Analyse
- **Tag 4**
  - Indexbildung
  - Inferenzstatistik II
  - Abschluss

Genereller Ablauf

- Vier Tage geblockt
- Mischung aus Input- und Übungssessions
- Anwesenheitsabfrage alle 90 Minuten

BETEILIGUNGSNACHWEIS

BN (2 CP): Anwesenheit + aktive Teilnahme + Take-Home Exam

- Anwesenheit:
  - maximal zwei Zeitslots à 90 Minuten bzw. in der Summe 180 Minuten verpassen (egal, ob dies unentschuldigt oder entschuldigt geschieht)
  - Überprüft mit Eintrag in Liste
- Aktive Teilnahme: Im Plenum, Mitarbeit an Übungen
- Take-Home Exam:
  - Aufgabenblatt, das eigenständig zu bearbeiten ist
  - Abgabe: **Dienstag, 12.08.2025 - 23:59 Uhr via E-Mail an ahrabhi.kathirgamalingam@cais-research.de**
  - Sie müssen die Hälfte der Punkte erreichen, um das Übungsblatt zu bestehen
  - Wenn Sie das Übungsblatt nicht bestehen, dann erhalten Sie einen Zweitversuch
  - Der Zweitversuch ist nur möglich, wenn Sie das erste Übungsblatt eingereicht haben!
  - Wir besprechen das Übungsblatt am Ende der letzten Sitzung gemeinsam

…CHATGPT & CO?

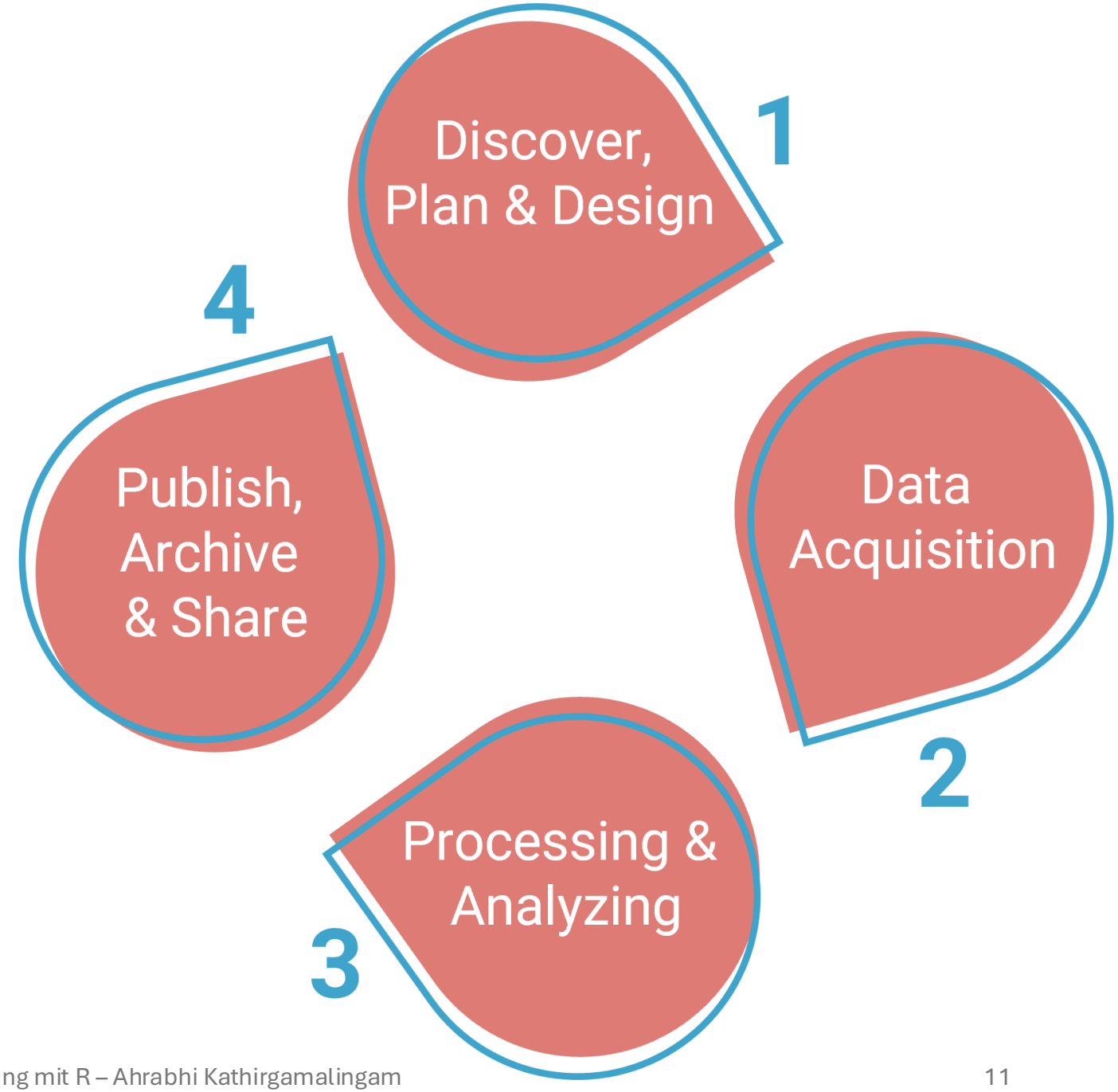
- Kann beispielsweise genutzt werden,
  - um Fehlermeldungen zu analysieren
  - um sich Code zu erklären
  - um Übungen für sich lösen zu lassen

- Auf den ersten Blick sind die Code-Ergebnisse beeindruckend
  - Auf den zweiten Blick aber auch häufig nutzlos: LLMs „fabulieren“ antworten herbei
  - Vorsicht bei der Aktualität der Testdaten
  - Kommen auf das Thema beim *Troubleshooting* zurück

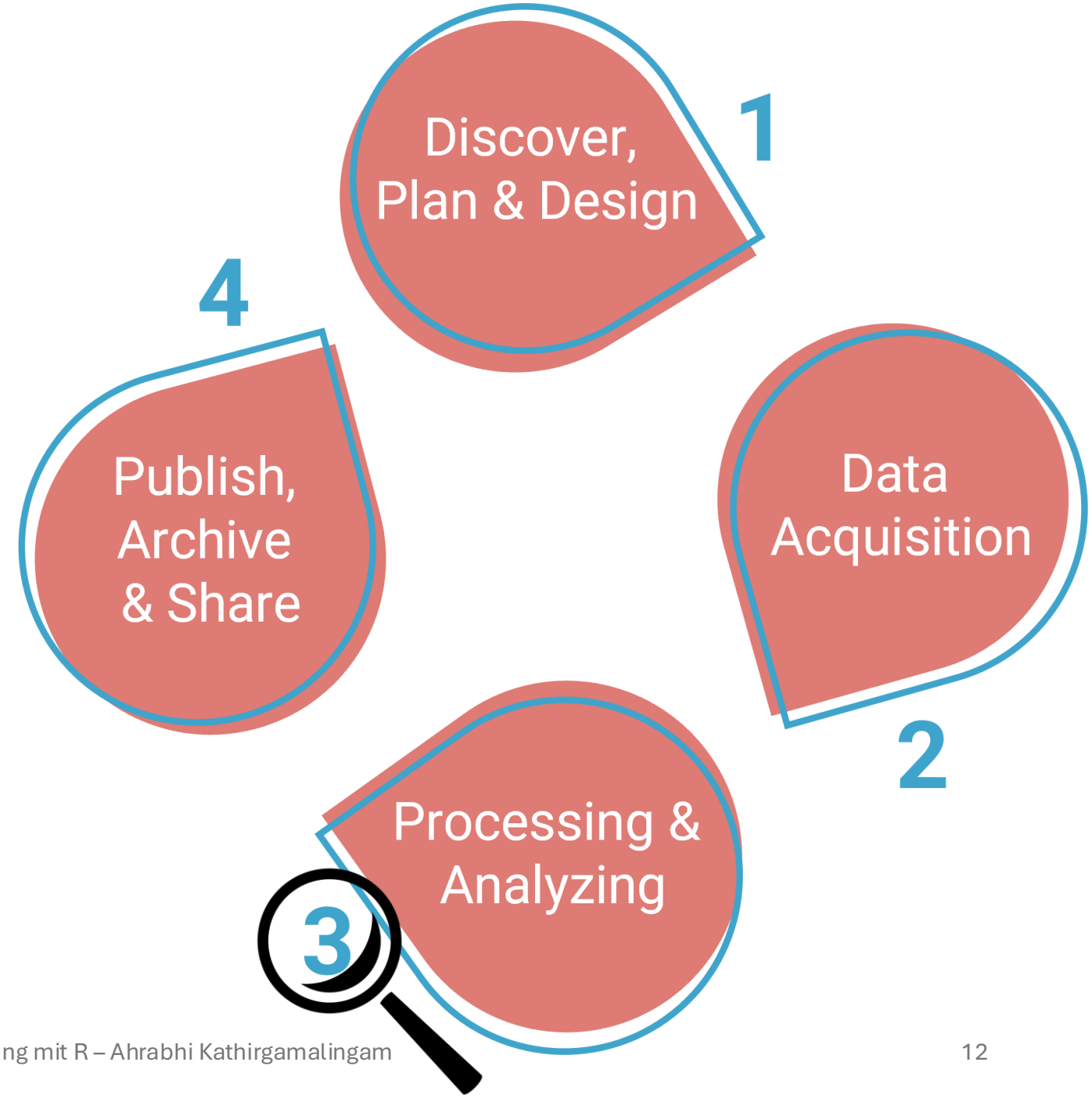
- Transparenter Umgang: Was klappt, was nicht? Versteht ihr die Outputs? Was prompten wir überhaupt? Und vor allem: *Nutzung reporten!*

Einführung: Datenauswertung mit R

Research Data Lifecycle



Research Data Lifecycle



DATENAUSWERTUNG

- Nach Forschungsdesign und Datenerhebung

- Prozess der Untersuchung von Daten, um Muster, Beziehungen, Erkenntnisse zu identifizieren und sie zu veranschaulichen

- Daten organisieren, vorbereiten, „processen“ – aufbereiten!
- Daten analysieren (Anwendung der Vorlesungsinhalte)
- Daten visualisieren

DATENAUSWERTUNG

- Nach Forschungsdesign und Datenerhebung

- Prozess der Untersuchung von Daten, um Muster, Beziehungen, Erkenntnisse zu identifizieren und sie zu veranschaulichen

- Daten organisieren, vorbereiten, „processen“ – aufbereiten!
- Daten analysieren (Anwendung der Vorlesungsinhalte)
- Daten visualisieren

…mit?

WAS IST R?

- R ist eine Statistik- und Programmiersprache
  - Version 1.0.0 im Jahr 2000 veröffentlicht
  - Entwicklung durch R Core Group
  - Frei („free as freedom“) und Open-Source

UND WAS IST RSTUDIO?

- RStudio ist eine Entwicklungsumgebung und Benutzeroberfläche (engl. Integrated Development Environment oder kurz: *IDE*)
  - RStudio wird seit etwa 2011 von Posit entwickelt
  - Als Desktop-Version verfügbar
  - For free but not Open-Source („free as free beer“)

- Es gibt zahlreiche Alternativen, z.B. Positron, Visual Studio Code oder Jupyter Notebooks, wir nutzen jedoch RStudio!

WARUM R?

- Für Statistik und Datenanalyse entworfen
- Open-Source-Programmiersprache und Interface
- Riesige Sammlung an Erweiterungen („Packages“)
- Aktive Nutzung im Feld und etabliert im Feld:
  - wird weiterentwickelt
  - viele Packages
  - viele Tutorials
  - Viel Unterstützung, Foren , usw.
- Reproduzierbarkeit & Automatisierung

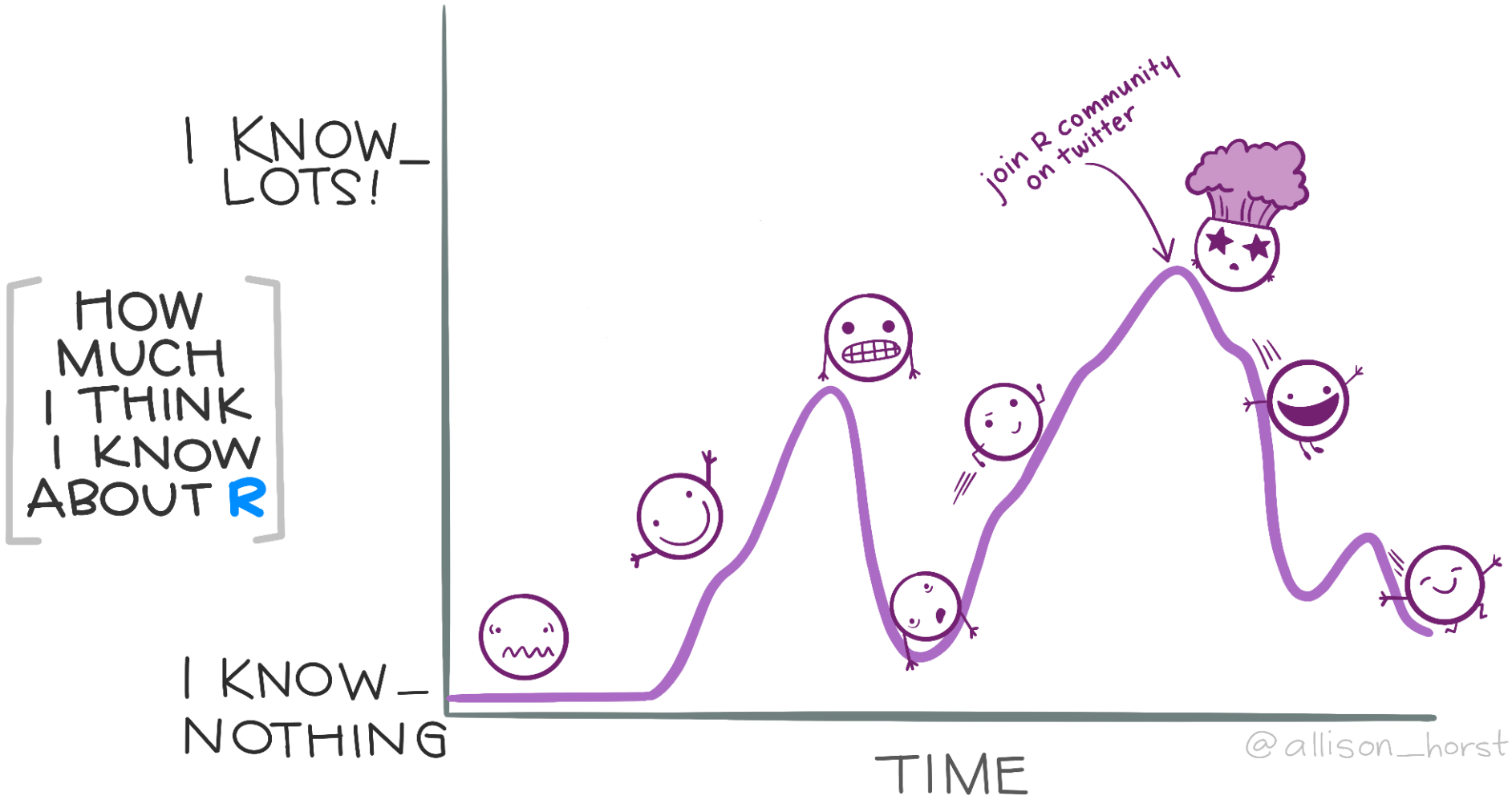
WARUM R LERNEN IN ZEITEN VON KI?

- Daten spielen zentrale Rolle, egal, in welchem Feld
- Einstieg in Programmiersprachen sinnvoll, um analytisches Denken zu fördern
  - Strukturiert denken
  - Lösungen finden
  - Recherchieren!
  - ChatGPT Outputs verstehen ☺
- Basics verstehen ist sehr hilfreich
- R ist für mehr als nur Statistik gut, zum Beispiel:
  - Computational Social Science
  - Datenvisualisierung
  - Automatisierung von Aufgaben

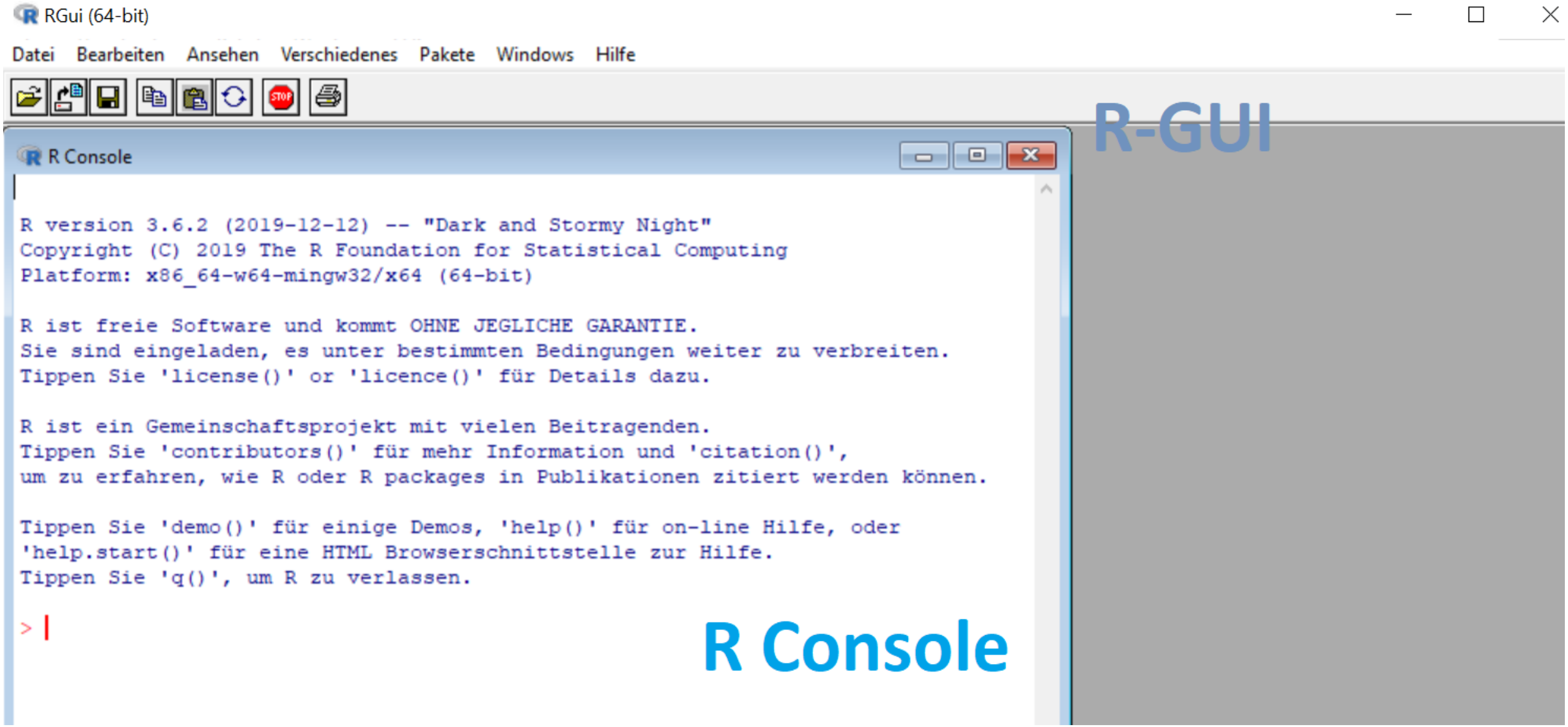
WARUM R LERNEN IN ZEITEN VON KI?

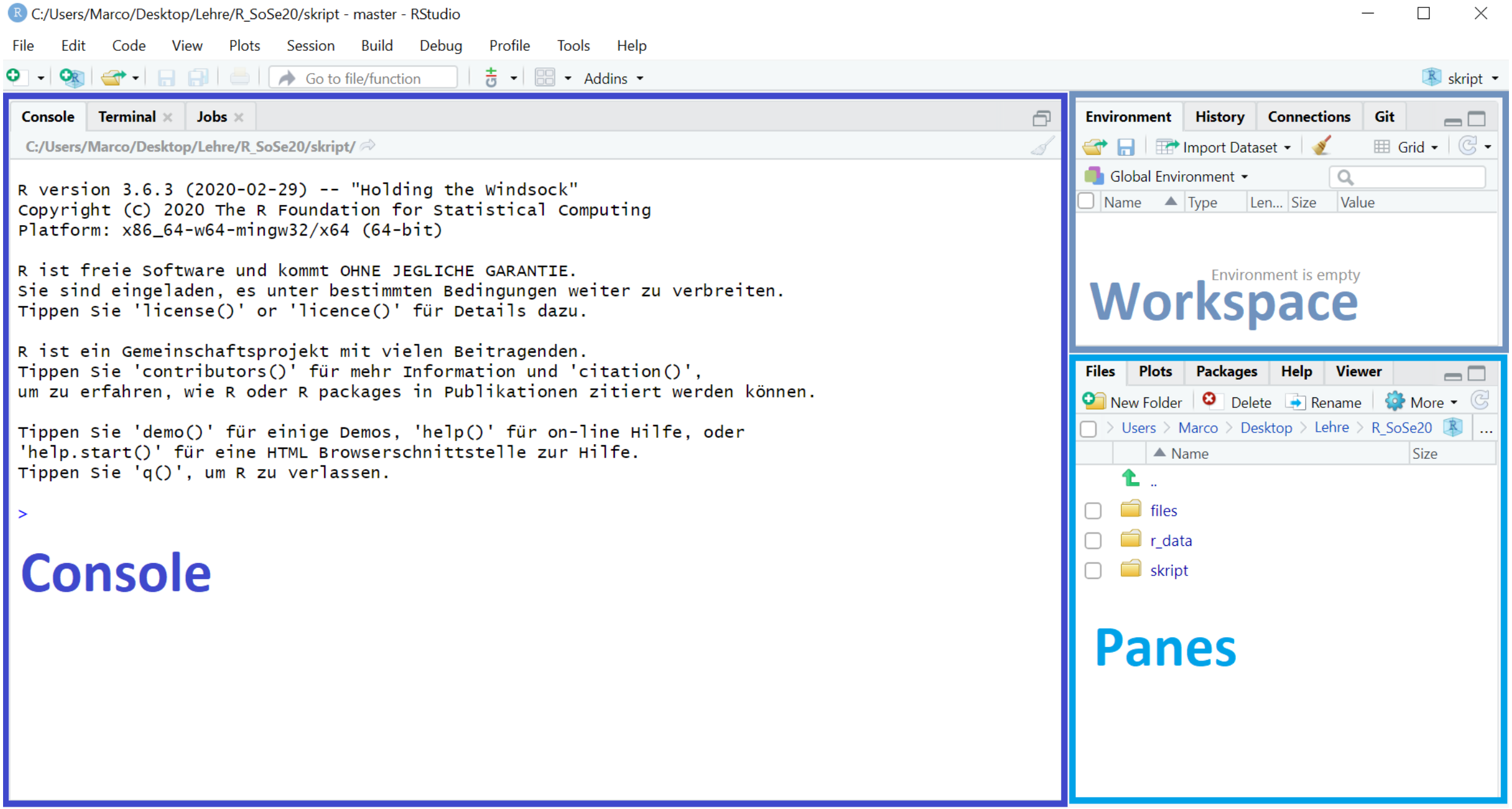
- Daten spielen zentrale Rolle, egal, in welchem Feld
- Einstieg in Programmiersprachen sinnvoll, um analytisches Denken zu fördern
  - Strukturiert denken
  - Lösungen finden
  - Recherchieren!
  - ChatGPT Outputs verstehen ☺
- Basics verstehen ist sehr hilfreich
- R ist für mehr als nur Statistik gut, zum Beispiel:
  - Computational Social Science
  - Datenvisualisierung
  - Automatisierung von Aufgaben

...Spaß?



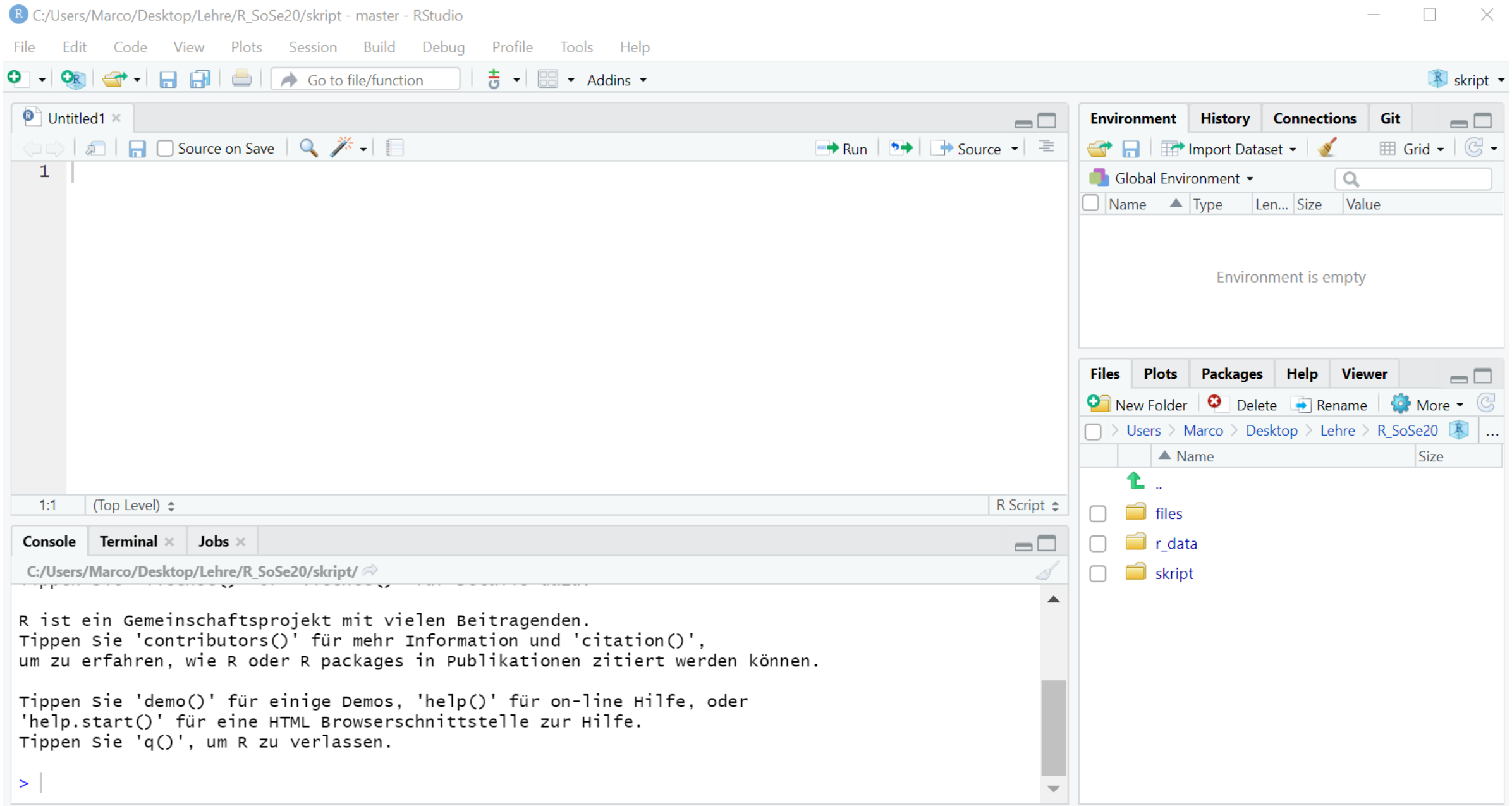
https://github.com/allisonhorst/stats-illustrations


RGui (64-bit)
Datei Bearbeiten Ansehen Verschiedenes Pakete Windows Hilfe
R-GUI
R Console
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
>
R Console


C:/Users/Marco/Desktop/Lehre/R_SoSe20/skript - master - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
skript
Console Terminal Jobs
C:/Users/Marco/Desktop/Lehre/R_SoSe20/skript/
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
>
Console
Environment History Connections Git
Import Dataset
Grid
Global Environment
Name Type Len... Size Value
Environment is empty
Workspace
Files Plots Packages Help Viewer
New Folder Delete Rename More
Users Marco Desktop Lehre R_SoSe20
Name Size
..
files
r_data
skript
Panes

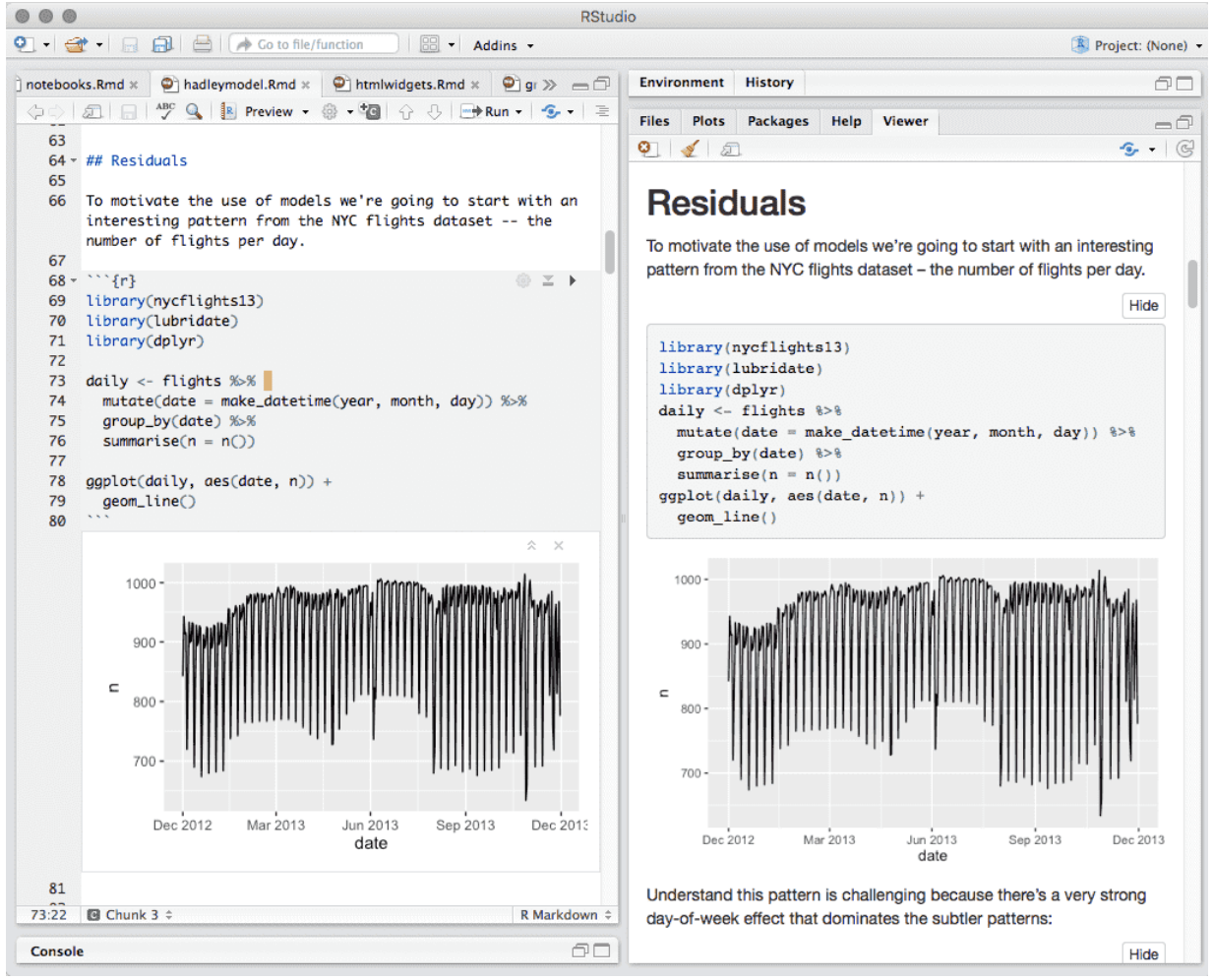
R Script

- Code wird allerdings nicht in der Console, sondern in Skripten (File -> new File -> R Script) geschrieben
- R Script: Quasi eine Text-File mit code
- File -> New File -> R Script
- Haben die Endung „.R“ oder „.r“
- Curser muss in die Zeile, die ausgeführt werden soll. Code wird anschließend mit einem Klick auf „run“ oder mit dem Shortcut „Strg“ + „Enter“ ausgeführt
- Kommentare werden mit einem „#“ eingeleitet


C:/Users/Marco/Desktop/Lehre/R_SoSe20/skript - master - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
skript
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Console
Terminal
Jobs
C:/Users/Marco/Desktop/Lehre/R_SoSe20/skript/
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
>
Environment
History
Connections
Git
Import Dataset
Grid
Global Environment
Name
Type
Len...
Size
Value
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Users
Marco
Desktop
Lehre
R_SoSe20
Name
Size
..
files
r_data
skript

R Markdown / Quarto

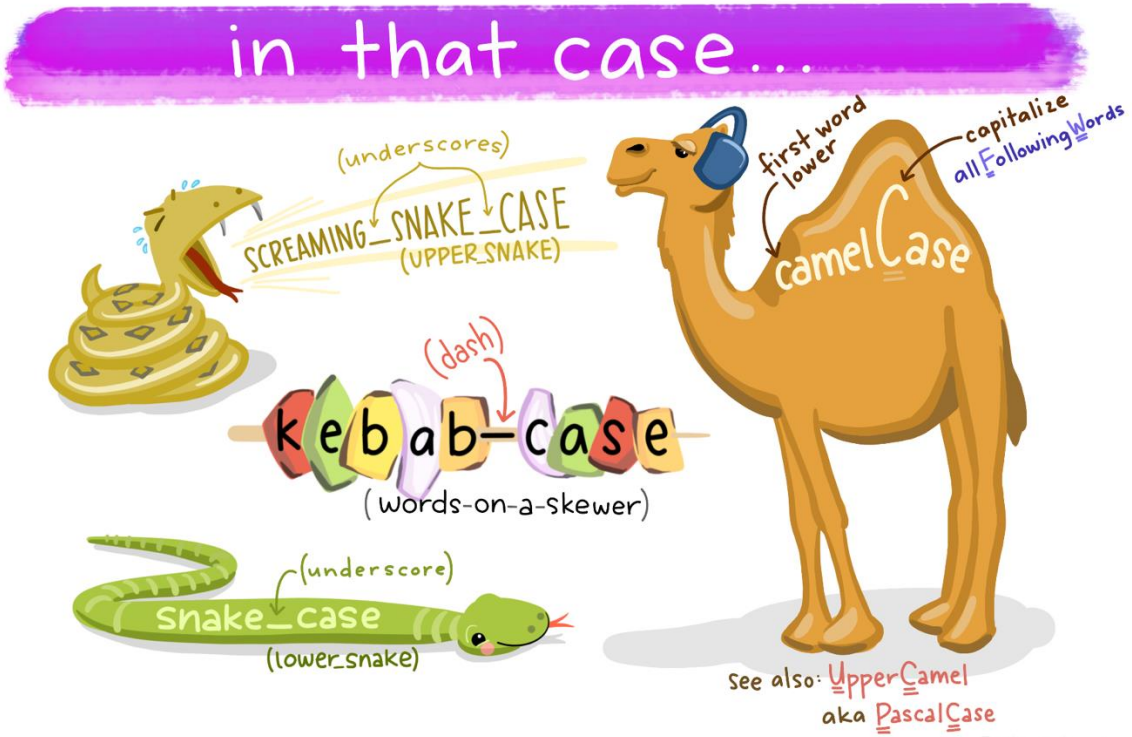
- Kombination aus Code und Text
- Gut zum Teilen, Veranschaulichen, etc.
- Wir bleiben bei R Script ☺



Konventionen – die Basics

- R* ist… sehr empfindlich!
  - Groß- und Kleinschreibung
  - Leerzeichen
  - Underscore, dash, etc.
  - Sonderzeichen
  - Umlaute (ä, ö, ü) -> lieber nicht
- Warum? Weil alles etwas ‚eigenes' ist
- …Mehr dazu später

*gilt für die meisten (Programmier)sprachen



https://github.com/allisonhorst/stats-illustrations

R in a nutshell:
Objects, Functions, Packages

Objects/ Objekte

- In R wird alles in *Objekten* gespeichert (Variablen, Daten, Funktionen etc.)
- Objekte haben verschiedene Klassen, Typen und Formate
- Objekte erhalten einen Wert durch Zuweisung!
  - Dafür wird das „assignment“-Zeichen verwendet “<-“
    - „Kleiner-als“ + „minus“
    - Shortcut: „alt“ + „minus“

Objects

Aufgabe:

x

x <- 3

x <- 3+4

X <- 3 *(Achtung, X großgeschrieben!)*

y <- 1, 5, 6, 7

y <- c(1, 5, 6, 7)

z <- "Hallo" *(Achtung, Anführungszeichen!)*

texts <- c("Hallo", "WER", "BIST", "du")

Objects

```
d <- 0
a <- b <- c <- d
print(a)

z <- 5
x <- y <- z
print(x)
```

Functions/ Funktionen

- Auch Funktionen werden in Objekten gespeichert
- Funktionen führen automatisiert Anweisungen/Aufgaben aus
  - z.B. Mittelwerte mit mean(x), Tabellen mit table(x), usw.
  - „?“ Funktion für Hilfe

- Auf Funktionsname folgte eine Klammer
  - In die Klammer kommt die Variable gefolgt von weiteren Argumenten
  - z.B. mean(x, na.rm = TRUE)

- Funktion auf Variable eines Datensatzes anwenden
  - Dollarzeichen nutzen „$“
  - mean(Datensatz$Variable, na.rm = TRUE)

Functions

Aufgabe:

?c

?mean

mean(x)

mean(y)

mean(z)

Base R

- Kernfunktion(en) von R (keine weitere Installation notwendig)
  - z.B. mean() oder table()
  - Aufgabe: library(help = "base")

- Verantwortung im „R Core Team“
  - Wird eher selten und sehr konservativ erweitert/verändert

Packages

- R bietet einen Kern an Funktionen (base R), die über Packages erweitert werden
  - Vergleich: Smartphone -> Apps

- Packages werden i.d.R. über Cran (The Comprehensive R Archive Network) installiert
  - Ca. 18.900 Packages werden auf Cran gelistet
  - Qualitätskontrolle und Standards u.a. Dokumentation zur Funktionsweise
  - Sie können auch Packages programmieren und anbieten

- Packages müssen einmalig installiert und für jede Session aktiviert werden
  - install.packages(„package“) und library(package)
  - Deinstallation über remove.packages(„package“)
  - Update aller Packages über update.packages()
  - sessionInfo() zeigt aktivierte Packages

Packages

Aufgabe:

sessionInfo()

install.packages(“dplyr“)

library(dplyr)

sessionInfo()

remove.packages(“dplyr“)

Packages

Aufgabe:

sessionInfo() zeigt aktivierte Packages

install.packages("dplyr") installiert dplyr

library(dplyr) aktiviert dplyr

sessionInfo() zeigt aktivierte Packages (neu dabei: dplyr)

remove.packages("dplyr") deinstalliert dplyr

PAUSE?

ÜBERRASCHUNGSTEST, juhu!

- Was ist base?
- Was ist ein Package?
- Wie weisen wir Objekten Werte zu?
- Funktioniert dieser Code: Library()?
- Funktioniert dieser Code: sessionInfo()?

Wieder/ weiter: Objects

Wiederholung: Objekte

- In R wird alles in Objekten gespeichert (Variablen, Daten, Funktionen etc.)
- Objekte erhalten einen Wert durch Zuweisung
  - Dafür wird das „assignment“-Zeichen verwendet <-
  - „Kleiner-als“ + „minus“
  - Shortcut: „alt“ + „minus“
- Objekte haben verschiedene **Klassen, Typen und Formate**

Objekte

Aufgabe:

class(x)

class(y)

class(z)

is.numeric(z)

is.character(z)

Objekte

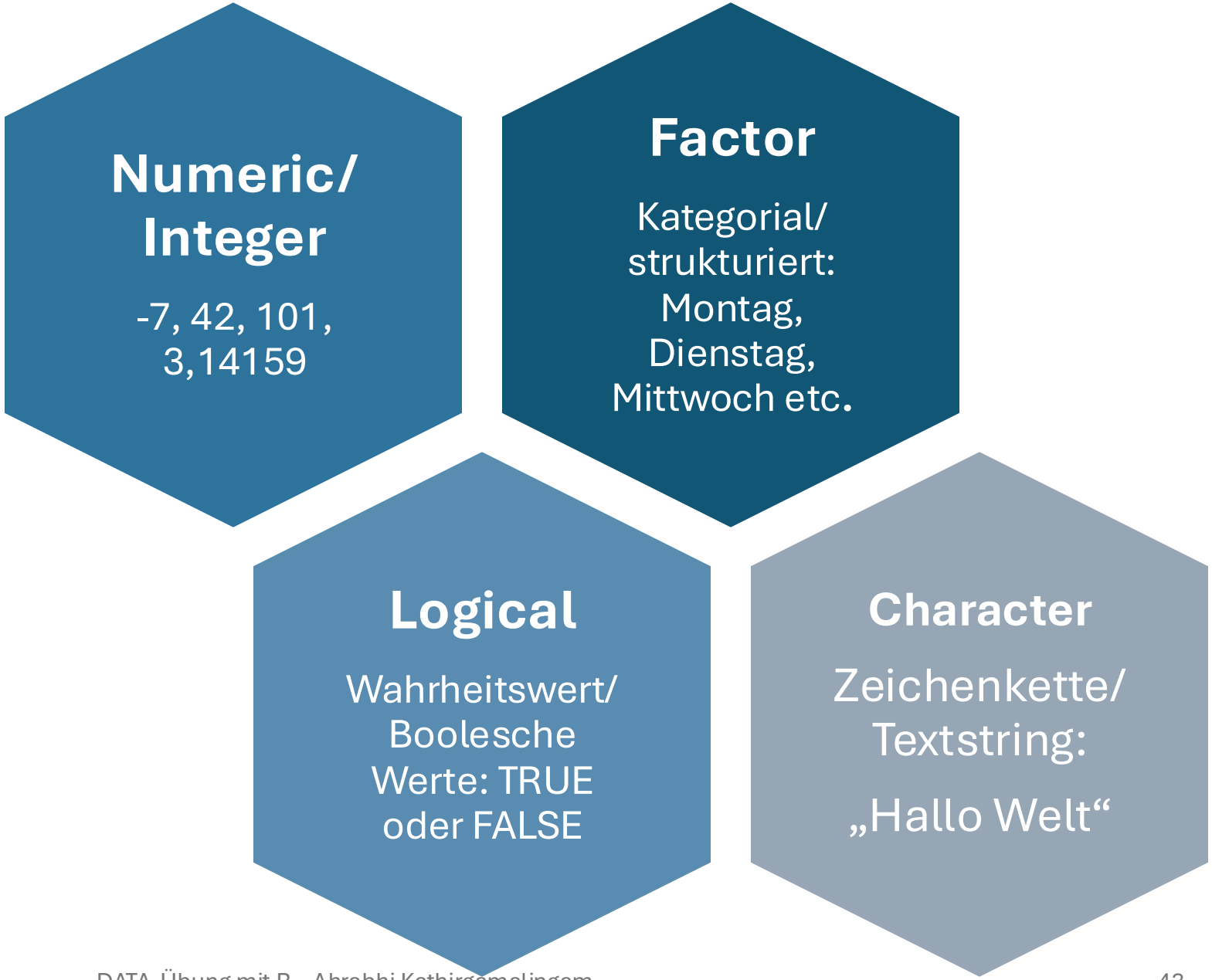
- Objekte haben unterschiedliche Klassen

```
#Objekte und Klassen
number <- 1 + 1
class(number)

## [1] "numeric"
```

```
#Objekte und Klassen
text <- "hello world!"
class(text)

## [1] "character"
```

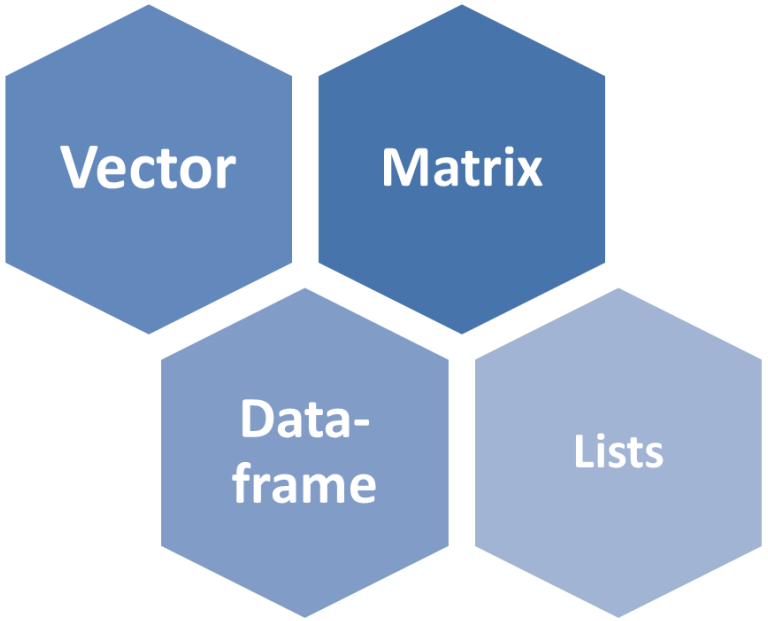
Datentypen/ Klassen

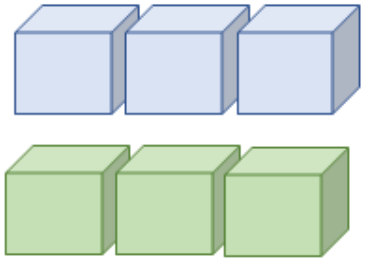
Zum Nachlesen



Objekte und Formate

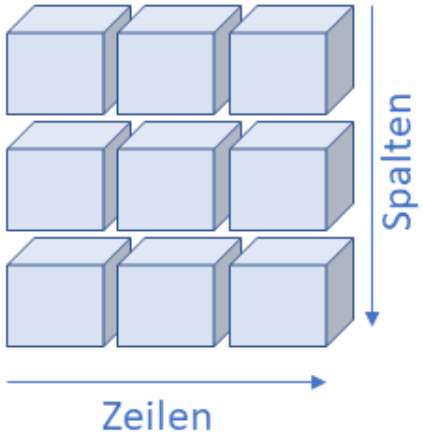
- Objekte (mit der jeweiligen Klasse) werden in unterschiedlichen Formaten gespeichert, dazu zählen …





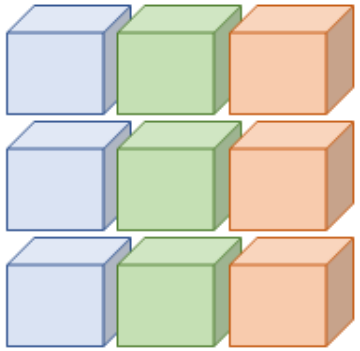
Vektor

Eine ganz einfache Liste von Werten einer Klasse!



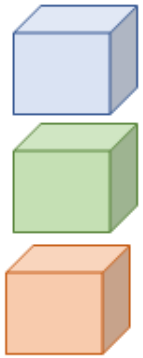
Matrix

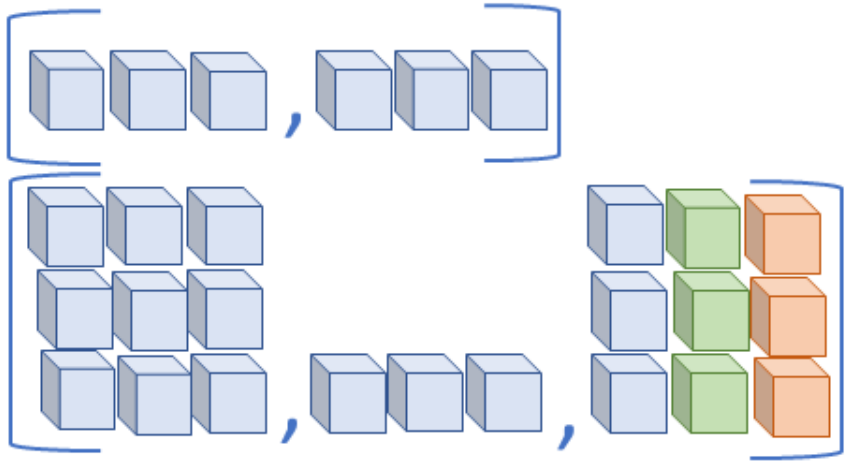
Eine Tabelle mit Werten einer Klasse!



Data Frame

Eine Tabelle, in der jede Spalte (Variable) eine andere Klasse sein kann!





Lists

Kann unterschiedliche Formate und Klassen beinhalten

Quelle: https://devopedia.org/r-data-structures

Fehlende Werte (Missing Values)

- Missing Values als spezifische Klasse
  - R kennt nur einen Wert (und nicht wie bei SPSS 77,777,7777) für Missing Values: NA (not available)
  - NA hat nicht die Klasse „character“
  - Aufgabe:

```
missing <- NA

class(missing)

is.na(missing)

var <- c(1, 3, NA, 6, NA, 100)

is.na(var)
```

Fehlende Werte (Missing Values)

```r
#Klasse NA
missing <- NA
class(missing)
```

```
## [1] "logical"
```

```r
#Variable mit Missing Values
var_1 <- c(1:25, NA, NA, NA, NA, NA)
#test auf Missing Values
is.na(var_1)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
#Summe NA
sum(is.na(var_1))
```

```
## [1] 5
```

Wieder/ weiter:
Konventionen & Good Practices

Konventionen

- Style Guides
  - Google‘s R Style Guide
  - Tidyverse Style Guide
- Bezeichnung für Variablen, Funktionen, Daten etc.
  - R ist case-sensitive! Es macht also einen Unterschied ob die Bezeichnung mit a oder mit A anfängt
  - Darf nicht mit einer Zahl beginnen (zB: 1var, lieber var_1)
  - Darf manche Sonderzeichen nicht beinhalten (z.B. *, #, !, $, @)
  - Besser keine Umlaute (ä,ö,ü)
  - Konvention: mit einem Unterstrich (my_variable), Bindestrich (my-variable), einem “Höcker” (“camel case”, myVariable) oder einem Punkt (my.variable), statt myvariable
  - **Bestenfalls einheitlich**! z.B. häufig wird der Name „df“ für dataframe vergeben

Konventionen

- *I will remember this code without comments (And Other Hilarious Jokes You Can Tell Yourself, Volume II)*
- Bitte stets den Code kommentieren!
  - Kommentare mit einen „#“ einleiten
- Chunks im Skript: Abgegrenzter Bereich z.B. Objekte testen
- Mit ### ---- Chunk-Name ----
  - z.B. ### ---- Testing-Objects ----

Good Practice

- R und R-Packages zitieren
  - R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
  - Citation()-Funktion, z.B. citation(„package“)
    - Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Hands on

- Gehen wir unser Skript nochmal durch und:
  - Fügen Comments hinzu (mit #)
  - Sortieren in Chunks (### ---- XYZ ----)

- Wie habt ihr sortiert?

Versionskontrolle

- Bewährter Workflow (Ironie!):
  - Analyse.r
  - Analyse_neu.r
  - Analyse_neu_1.r
  - Analyse_ganz_neu.r
  - Analyse_wirklich_ganz_neu.r
  - Analyse_wirklich_ganz_neu_final.r
  - Analyse_wirklich_ganz_neu_final_25.03.24.r
- Versionskontrolle als Lösung!
- Happy Git and GitHub for the useR – Jennifer Bryan

R Projects

- Organisieren den Workflow in R; z.B. Ordner, Unterordner, Skripte, Grafiken, etc.
  - Erstellen einen relativen Pfad
  - Ermöglichen die Reproduktion des Codes

- What They Forgot to Teach You About R – Jennifer Bryan & Jim Hester
- The Missing Semester of Your CS Education

R Projects organisieren

```
C:./User/Desktop/Projekt/Projekt_1
|    mein-R-Project.Rproj
|    set-up.R
|    script_1.R
|    script_2.R
+---data
|       allbus_2018_gesamt.sav
|       allbus_fb.pdf
+---export
|       data_export_1.csv
+---graphics
|       plot_1.png
|       plot_.png
```

Working Directory

- Absoluter Pfad: Working Directory ist der Ordner „R_MA_SoSe_22“
  - C:\Users\Ahrabhi\Desktop\Lehre\R_SoSe_2024\data
- Relativer Pfad: Working Directory ist der Ordner „R_SoSe_2024“ (Shortcut „.\“)
  - .\data
- Relative Pfade sind absoluten Pfaden vorzuziehen,
  - Weil Ordner verschoben werden und der Pfad nicht aktuell ist
  - Weil Projekte sonst nicht von anderen Systemen genutzt werden können
- **Falsche Pfade zählen zu den häufigsten Fehlerquellen am Anfang:**
  - / oder \
  - Wo liegen meine Daten?

Working Directory

Herausfinden, in welchem wd wir sind:
getwd()
Einen working directory setzen wir mit der Funktion:
setwd(„PFAD“)

- Aufgabe:
  - Erstelle einen Ordner auf deinem Desktop namens „DATA_SoSe2025“
  - Checke die WD mit getwd()
  - Setze deine WD zu dem Ordner mit setwd(“PFAD“)

Daten und Datenimport

Datenimport

- Test- und Beispieldaten
  - Werden häufig von Ressourcen genutzt, um Beispiele zu zeigen
  - Überblick über data()
    - Mtcars
    - Titanic
    - Iris

- Unterschiedliche Daten-Typen können importiert werden
  - .RData
  - . sav (SPSS)
  - .dta (STATA)
  - .csv
  - .xlsx

Datenimport Beispieldaten

- Aufgabe:
  - df <- mtcars
  - Was ist das für ein Datensatz?
  - mean(mtcars) – was passiert?
  - mean(df$cyl)

Datenimport Beispieldaten

- Aufgabe:
  - df <- mtcars
  - Was ist das für ein Datensatz? Was sind obs? Was sind variables?
  - mean(mtcars) – was passiert?
  - mean(mtcars$cyl)

REMINDER:
Funktion auf Variable eines Datensatzes anwenden:
Dollarzeichen nutzen „$“
mean(Datensatz$Variable, na.rm = TRUE)

Welche Daten kennen Sie?

Datenquellen

- Es gibt unzählige Datenquellen (, die teilweise frei zugänglich sind)
  - Google Dataset Search
  - gesisDataSearch
  - Harvard Dataverse
- Open-Data
  - Daten als „Allgemeingut“
  - Open-Data-Gesetz: Daten maschinenlesbar und entgeltfrei zur Verfügung stellen
  - Kartenmaterial und Geodaten z.B. OpenStreetMap
  - Bevölkerungs-, Verwaltungs-, Wahl-, Wetter- und Umweltdaten
  - Z.B. über Open-Data-Portale

Datenquellen

- Wir arbeiten mit existierenden Datensätze zum Beispiel
  - ALLBUS (Allgemeine Bevölkerungsumfrage der Sozialwissenschaften)
  - GLES (German Longitudinal Election Study)
  - ESS (European Social Survey)

ALLBUS

- Allgemeine Bevölkerungsumfrage der Sozialwissenschaften (kurz: ALLBUS)
  - Seit 1980 alle zwei Jahre: Erhebung zur gesellschaftlichen Dauerbeobachtung von Einstellungen, Verhalten und sozialem Wandel in Deutschland, z.B. Mediennutzung, politische Einstellungen, soziales Kapital, Soziodemografie etc.
  - Aufgrund von Corona wurde die Befragung 2020 nicht durchgeführt
  - Weitere infos: https://www.gesis.org/allbus/inhalte-suche/studienprofile-1980-bis-2018/2018

Aufgabe

- Recherchieren Sie Informationen zum ALLBUS 2018. Treffen Sie Aussagen zum
  - Erhebungszeitraum
  - Grundgesamtheit
  - Anzahl der Befragten
  - Auswahlverfahren

ALLBUS

- ALLBUS 2018:
  - Erhebungszeitraum: April 2018 bis September 2018
  - 3.477 Befragte/708 Variablen
- Grundgesamtheit und Auswahl:
  - Untersuchungsgebiet: Bundesrepublik Deutschland
  - Personenstichprobe:
    - Grundgesamtheit: Personen (Deutsche und Ausländer), die zum Befragungszeitpunkt in Privathaushalten lebten und vor dem 01.01.2000 geboren sind.
  - Auswahl:
    - Zweistufige, disproportional geschichtete Zufallsauswahl in Westdeutschland (incl. West-Berlin) und Ostdeutschland (incl. Ost-Berlin). In der ersten Auswahlstufe wurden Gemeinden in Westdeutschland und in Ostdeutschland mit einer Wahrscheinlichkeit proportional zur Zahl ihrer erwachsenen Einwohner ausgewählt. In der zweiten Auswahlstufe wurden Personen aus den Einwohnermeldekarteien zufällig gezogen. Zielpersonen mit nicht hinreichend guten Deutschkenntnissen zählen zu den systematischen Ausfällen.

ALLBUS

- Erhebungsverfahren: ALLBUS 2018
  - Persönlich-mündliche Befragung mit standardisiertem Frageprogramm (CAPI – Computer Assisted Personal Interviewing)
- Ausschöpfungsquote:
  - Gesamt: 32,4 %
  - Westdeutschland: 32,6 %
  - Ost: 31,7 %

Datenimport

- Nutzung des Rio-Packages macht das Leben einfach
  - Install.packages(" rio ") und library(rio)
  - Wahrscheinlich ist der zusätzliche Befehl install_formats() notwendig

```
library(rio)
allbus <- import("data/allbus_2018_gesamt.sav")
```

- Häufigste Fehlerquelle: Arbeitsverzeichnis beachten!
  - setwd()
  - getwd()

Was sehen wir? ☺

REFRESHER

- Was ist eine working directory?
- Warum sollten wir Code kommentieren?
- Sollten wir eine Variable varfünf nennen?
- Welche Klassen gibt es?
- Welche Formate gibt es?
- Was ist NA?

FRAGEN, UNKLARHEITEN, FEEDBACK?

VIELEN DANK ☺

UND BIS MORGEN!

ahrabhi.kathirgamalingam@cais-research.de