

# Quantitative Methoden mit R & RStudio

Sommersemester 2026 | Sitzung 2

Maura Kratz

# Willkommen zurück!

I) Vorbereitung	01: Einführung in R und R-Studio	02: Projekte und Datenimport	03: Daten aufbereiten	04: Daten zusammenführen
II) Deskriptiv	05: Kannwerte & deren Visualisierung	06: Häufigkeiten & deren Visualisierung	07: The Grammar of Graphics	08: Korrelation & deskriptive Regression
III) Inferenz	09: Einfache und multiple lineare Regression (OLS)	10: Regression Diagnostics	11: Logistische Regression (GLM)	12: Varianzanalysen (ANOVA)
13: Wahlthema (Faktoranalyse, Indexbildung, ...) / Research Notes				
14: Wahlthema (Faktoranalyse, Indexbildung, ...) / Research Notes				

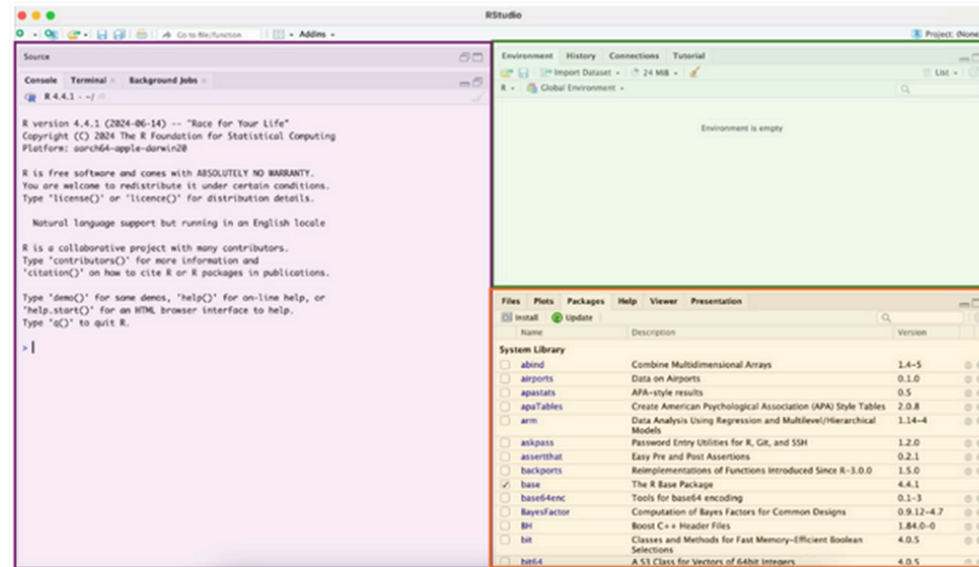
miro

# Was heute ansteht:

- Check-In: Installation & Fragen
- RStudio-Oberfläche
  - Einstellungen
  - Hilfe
- Projektordner anlegen
- Basic R-Syntax & Nützliche Shortcuts
- Daten einlesen

# Die RStudio-Arbeitsumgebung

Bisher sieht es bei euch vermutlich etwa so aus:



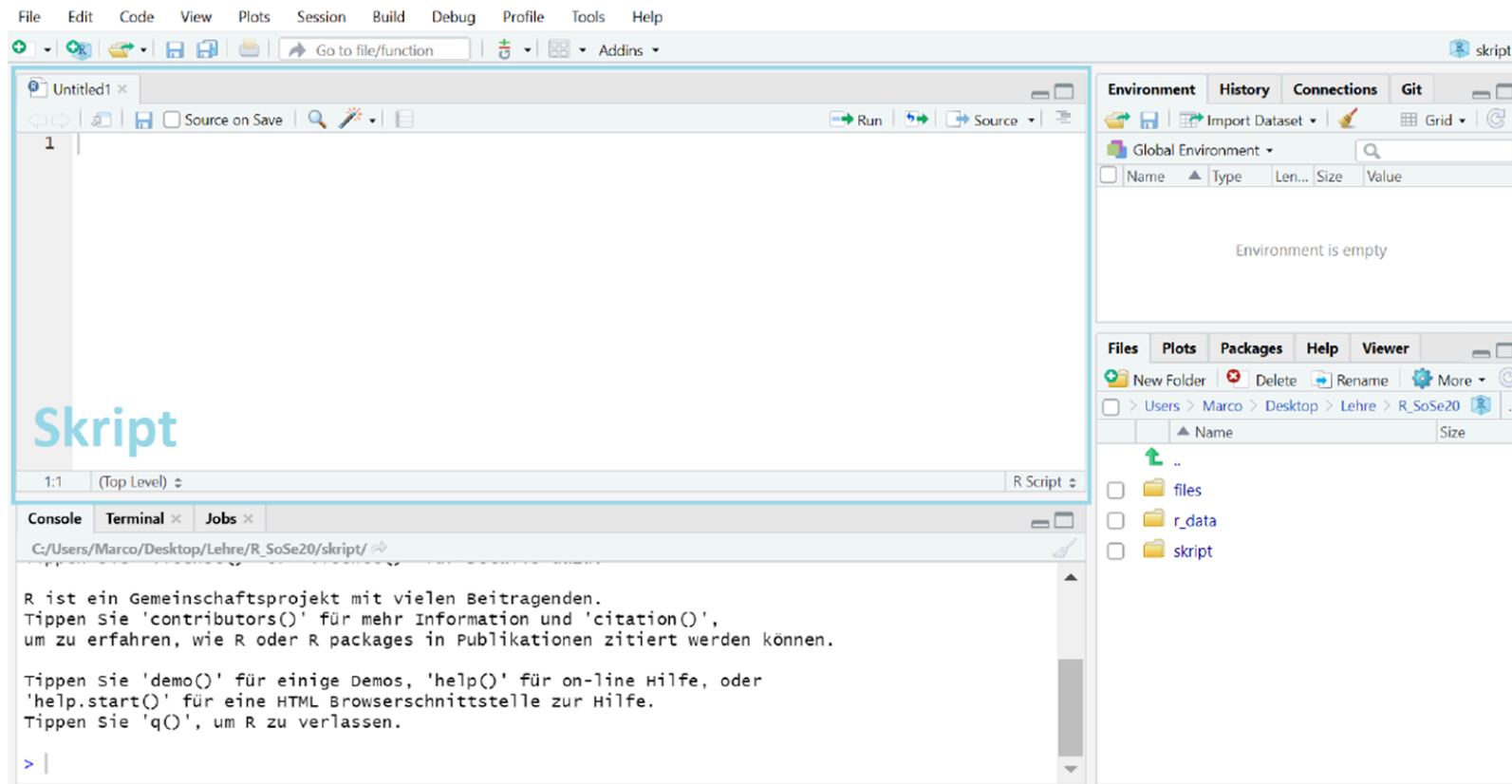
**Console:**  
Hier erscheint der  
ausgeführte Code zusammen  
mit den Ergebnissen

**Environment:**  
Überblick über  
aktuell geladene Objekte

**Files:** Dateien  
**Plots:** Grafiken  
**Help:** Informationen zu R und zu  
Funktionen  
**Packages:** Übersicht über Pakete

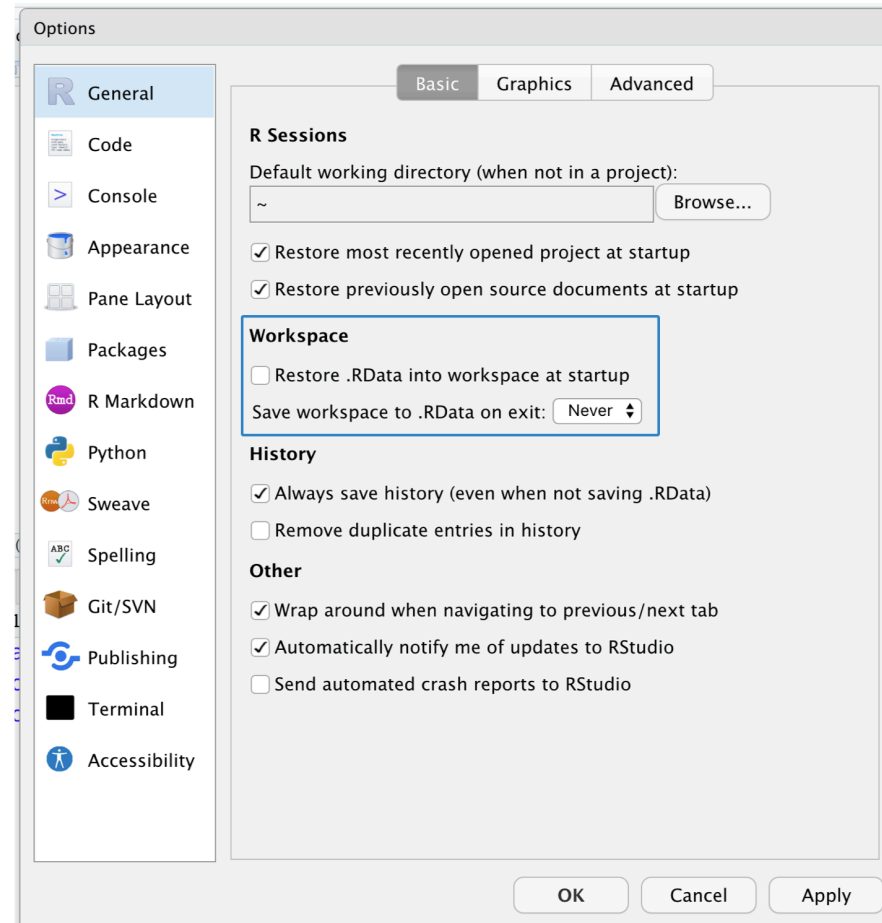
# Die RStudio-Arbeitsumgebung

Da wir in RStudio aber immer mit Skripten arbeiten, sieht die Oberfläche typischerweise so aus:



# Einstellungen

- Tools → Global Options
- Bitte die Optionen zum Workspace deaktivieren: Häkchen nicht setzen und im drop-down-Menü “never” wählen
- mehr Infos zu den Einstellungen im [User Guide](#)



# relative und absolute Pfade

## Windows

### Absoluter Pfad:

```
# Pfad mit "Shift+Rechtsklick+Pfad kopieren" kopieren und einfügen führt zu backward slashes (\)
```

```
# Option 1: händisch zu forward slashes konvertieren
```

```
daten <- rio::import("C:/Forschung/Forschungsprojekt_A/data/beispieldaten.csv")
```

```
## ...
```

### Relativer Pfad:

```
# Wenn wir einen working directory spezifiziert, oder ein Projekt angelegt haben, müssen wir statt des absoluten nur noch den relativen Pfad angeben
```

```
# working directory setzen (falls kein Projekt)
```

```
setwd("C:/Forschung/Forschungsprojekt_A")
```

```
## ...
```

# Relative und absolute Pfade

macOS / Linux

Absoluter Pfad:

```
# Pfad mit "Rechtsklick + Als Pfadname kopieren" kopieren und einfügen führt zu forward slashes

# kein raw string nötig
daten <-
rio::import("/Users/maura/Forschung/Forschungsprojekt_A/data/beispieldaten.csv")
```

Relativer Pfad:

```
# Wenn wir ein Projekt angelegt haben oder setwd() genutzt haben, reicht der relative Pfad

# Working Directory setzen (wenn nicht ohnehin ein Projekt)
setwd("/Users/maura/Forschung/Forschungsprojekt_A")

rio::import("data/beispieldaten.csv")
```

# Ordner und Pfade

- Wir müssen R sagen, wo es Datensätze einlesen und Ergebnisse ablegen soll
- Dazu ist es sinnvoll in Projekten zu arbeiten (1 Projekt pro Kurs/ Hausarbeit/ Forschungsprojekt)
- So müssen nicht immer lange, nicht reproduzierbare absolute Pfade angegeben werden
- Und es muss nicht immer explizit der Arbeitsordner mit `setwd("Pfad-zum-Ordner")` festgelegt werden
- Ein **Projekt** legt einen Ordner als Standard-Arbeitsordner fest
- Von dort aus können dann unterschiedliche Unterordner mit relativen Pfaden angesteuert werden:

```
data_1 <- rio::import("../data/dataset_1.csv")
```

```
R_Kurs/  
├── R_Kurs.Rproj  
├── data/  
│   └── bspdaten.csv  
├── slides/  
│   └── sitzung_01.pdf  
├── skripte/  
│   └── 02_analyse.R  
├── uebungen/  
│   └── uebung_01.R  
└── output/  
    ├── grafik_01.png  
    └── tabelle_01.docx
```

## ! Important

Niemals ein Leerzeichen in Ordner- oder Dateinamen!

# Die wichtigsten Keyboard-Shortcuts

- `Str + Enter` führt Code aus. Dafür:
  - entweder den auszuführenden Code markieren
  - oder den Cursor irgendwo im auszuführenden Code-Chunk platzieren
- `Str + Shift + S` führt das gesamte Skript aus.
- `Str + Shift + M` fügt die Pipe (`%>%`) ein
- `Str + Shift + L` räumt die Console auf

# Guter Stil im Skript

1. erkläre, was du tust in `#` Kommentaren
2. benenne Objekte ausschließlich mit Kleinbuchstaben und trenne Wörter mit `_`

```
i_use_snake_case  
otherPeopleUseCamelCase  
some.people.use.periods  
And_aFew.People_RENOUNCEconvention
```

3. unterteile dein Skript in Sektionen

```
# Load data -----  
  
# Plot data -----
```

## Note

Mehr dazu

im [Tidyverse-Styleguide](#) und im eBook [R For Data Science](#).

# Guter Stil im Code

1. Leerzeichen vor Operatoren (außer `^`), vor `%>%` und vor `+` in ggplot2-Befehlen
2. keine Leerzeichen vor oder nach Klammern
3. neue Argumente/ Befehle kommen in eine neue Zeile
4. Pipes sollten grundsätzlich das letzte in einer Codezeile sein
5. eine Pipe sollte nicht länger als 10-15 Zeilen sein

# Basic R-Syntax

- `#` für Kommentare, also Beschreibungen
- `<-` um etwas einem R-Objekt zuzuweisen
- `[]` um Positionen anzugeben
- Durch Drücken der Tab-Taste erhältst du Vorschläge zur Vervollständigung
- `%>%` bzw. Str. + Shift + M um mehrere Befehle mittels Pipe zu verketteten
- Aufbau von Befehlen: `command_name(argument1 = value1, argument2 = value2, ...)`

# Wo bekomme ich Hilfe ?

## 1. Direkt in RStudio

- `?befehl()`, z.B. `?mean()`
- `?paket`, z.B. `?dplyr`
- `??"stichwort"`, z.B. `??"mean"`

## 2. Von Kommiliton\*innen

## 3. Von mir während der Sitzung und/ oder in Sprechstunden

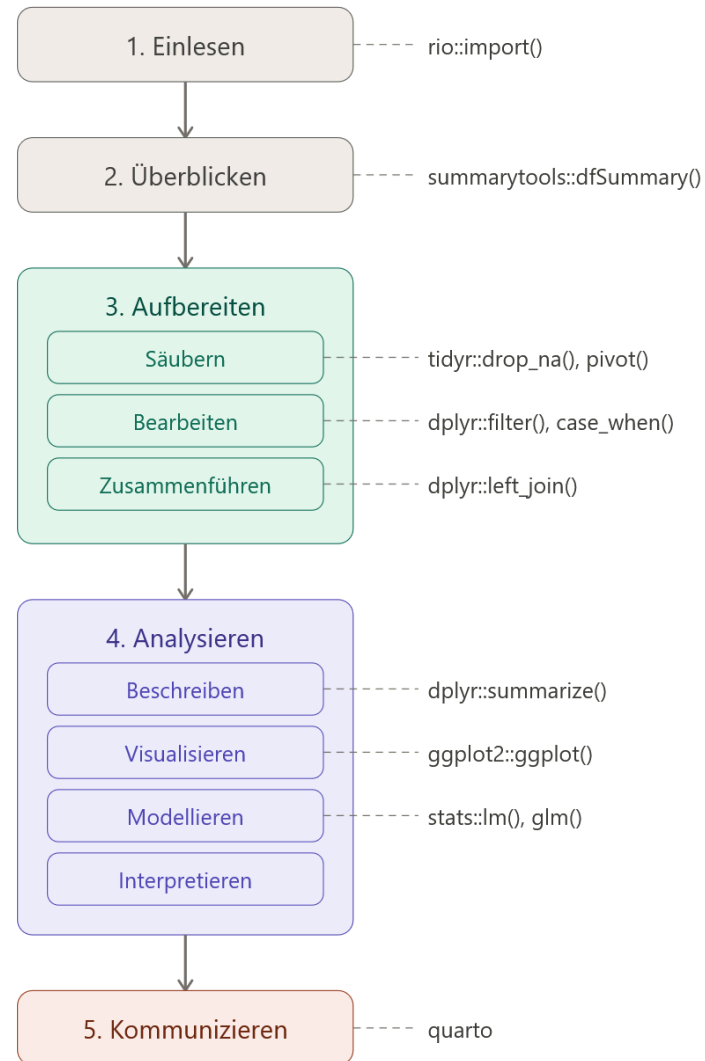


# Wo bekomme ich Hilfe ?

## 4. Online

- bei RStudio/[Posit](#)
- über die von Google betriebene Suchmaschine [Rseek](#)
- auf der [CRAN-Webseite](#) unter “Manuals” gibt es hilfreiche Einstiegshandbücher
- auf [Datacamp](#) für einen schnellen Überblick über Basics
- in den Ressourcen im Syllabus

# Datenanalyse-Workflow



# Hands On - Daten einlesen



# Minute Cards

Bitte füllt die Minute Cards für die heutige Sitzung aus. Das sollt enicht länger als 3 Minuten dauern. Vielen Dank für eure Mitarbeit!



# Vielen Dank und bis kommenden Dienstag!



Sitzung 2 von 14



**Übung 1** zu “Daten einlesen” bis spätestens Sonntagabend!