



Analyse und Dokumentation

BSc Psychologie SoSe 2026

Belinda Fleischmann

Datum	Gruppe 1	Gruppe 2	Format	Thema
Fr, 10.04.2026	11-13 Uhr	13-15 Uhr	Seminar	(1) Quarto, Zotero, Tidyverse
Fr, 17.04.2026	11-13 Uhr	13-15 Uhr	Seminar	(2) Forschungsethik
Fr, 24.04.2026	11-13 Uhr	13-15 Uhr	Seminar	(3) Wissenschaftliche Berichte
Mi, 29.04.2026	11-13 Uhr	13-15 Uhr	Praxisseminar	Offene Übung (Quarto und Zotero)
Fr, 08.05.2026	11-13 Uhr	13-15 Uhr	Seminar	(4) Offenheit und Transparenz
Fr, 15.05.2026	11-13 Uhr	13-15 Uhr	Übung	Einfache Lineare Regression
Mo, 18.05.2026	11-13 Uhr	15-17 Uhr	Übung	Korrelation
Fr, 05.06.2026	11-13 Uhr	13-15 Uhr	Übung	Einstichproben-T-Test
Mi, 10.06.2026	13-15 Uhr	17-19 Uhr	Übung	Zweistichproben-T-Test
Fr, 26.06.2026	11-13 Uhr	13-15 Uhr	Übung	Einfaktorielle Varianzanalyse
Fr, 26.06.2026	15-17 Uhr	17-19 Uhr	Übung	Zweifaktorielle Varianzanalyse
Fr, 03.07.2026	11-13 Uhr	13-15 Uhr	Übung	Multiple Regression
Fr, 10.07.2026	11-13 Uhr	13-15 Uhr	Übung	Kovarianzanalyse
Fr, 24.07.2026			Klausurtermin	

(1) Quarto, Zotero, Tidyverse

Quarto

Zotero

Tidyverse

Quarto

Zotero

Tidyverse

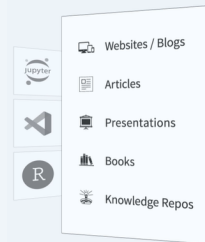
Quarto

Welcome to Quarto

An open-source scientific and technical publishing system

- Author using [Jupyter](#) notebooks or with plain text markdown in your favorite editor.
- Create dynamic content with [Python](#), [R](#), [Julia](#), and [Observable](#).
- Publish reproducible, production quality articles, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Share knowledge and insights organization-wide by publishing to [Posit Connect](#), [Confluence](#), or other publishing systems.
- Write using [Pandoc](#) markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.

Analyze. Share. Reproduce. You have a story to tell with data—tell it with Quarto.

[Get Started](#)[Guide](#)

Was ist Quarto?

- Ein seit 2022 verfügbares freies wissenschaftlich-technisches Publikationssystem
- Eine Weiterentwicklung von [RMarkdown](#) und [RBookdown](#) durch [Posit](#)
- RMarkdown/RBookdown sind RStudio Adaptationen von [Markdown](#) und [Jupyter Notebooks](#)
- Allgemeines Ziel ist hier die einfache Integration von ausführbarem Programmiercode in ein ansprechendes Text-, Tabellen- und Abbildungslayout für Web- und Printdokumente.
- Quarto nutzt [Markdown](#) und [Latex](#) für Layoutprozesse.
- Quarto nutzt [Pandoc](#) für multiple Outputformate (.html, .docx, .pdf, etc.)
- Quarto läuft smoother und schneller als RMarkdown und RBookdown.

Quarto Installation



Overview **Get Started** Guide Extensions Reference Gallery Blog Help ▾



Get Started

Tutorial: Hello, Quarto
Tutorial: Computations
Tutorial: Authoring

Get Started

Install Quarto, then check out the tutorials to learn the basics.

Step 1

Install Quarto

Find your operating system in the table below

Platform	Download	Size	SHA-256
Ubuntu 18+/Debian 10+	quarto-1.4.554-linux-amd64.deb	111.82 MB	7b07062
Linux x86 Tarball	quarto-1.4.554-linux-amd64.tar.gz	113.04 MB	f01203f
Linux Arm64	quarto-1.4.554-linux-arm64.deb	112.52 MB	4291e1b
Linux Arm64 Tarball	quarto-1.4.554-linux-arm64.tar.gz	113.6 MB	43c788d
RHEL 7 Tarball	quarto-1.4.554-linux-rhel7-amd64.tar.gz	113.4 MB	7d5264b
Mac OS	quarto-1.4.554-macos.pkg	186.2 MB	ab6a44c
Windows	quarto-1.4.554-win.msi	108.89 MB	76d281d
Release notes and more downloads...			

Step 2

Choose your tool and get started



Quarto VSCode Tutorial



Overview **Get Started** Guide Extensions Reference Gallery Blog Help ▾



Get Started
Tutorial: Hello, Quarto
Tutorial: Computations
Tutorial: Authoring

Tutorial: Hello, Quarto

Choose
your
tool



Overview

In this tutorial we'll show you how to use Quarto with VS Code. Before getting started, you should install the [Quarto VS Code Extension](#), which includes many tools that enhance working with Quarto, including:

- Integrated render and preview for Quarto documents.
- Syntax highlighting for markdown and embedded languages
- Completion and diagnostics for YAML options
- Completion for embedded languages (e.g. Python, R, Julia, etc.)
- Commands and key-bindings for running cells and selected lines.

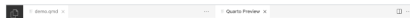
You can install the Quarto extension from within the **Extensions** tab in VS Code, from the [Extension Marketplace](#), the [Open VSX Registry](#) or directly from a [VSIX extension file](#).

Note

This tutorial focuses on editing plain text Quarto `.qmd` files in VS Code. Depending on your preferences and the task at hand there are two other editing modes available for Quarto documents: the [Visual Editor](#) and the [Notebook Editor](#). For the purposes of learning we recommend you work through this tutorial using the VS Code text editor, then after you've mastered the basics explore using the other editing modes.

Basic Workflow

Quarto `.qmd` files contain a combination of markdown and executable code cells. Here's what it might look like in VS Code to edit and preview a `.qmd` file:



On this page

Overview

Basic Workflow

Render and Preview

YAML Options

Markdown

Code Cells

External Preview

Next Up

Edit this page

Report an issue

Was ist Markdown?

- Eine Markup Language (Auszeichnungssprache) zur Erzeugung formatierten Texts
- Eine HTML Alternative zur Erstellung von Webseiten etc. mithilfe einfacher Texteditoren
- Von John Gruber und Aaron Swartz 2004 mit dem Ziel hoher Lesbarkeit entwickelt

Text using Markdown syntax	Corresponding HTML produced by a Markdown processor	Text viewed in a browser
<pre>Heading ***** Sub-heading ----- # Alternative heading ## Alternative sub-heading Paragraphs are separated by a blank line. Two spaces at the end of a line produce a line break.</pre>	<pre><h1>Heading</h1> <h2>Sub-heading</h2> <h1>Alternative heading</h1> <h2>Alternative sub-heading</h2> <p>Paragraphs are separated by a blank line.</p> <p>Two spaces at the end of a line
 produce a line break.</p></pre>	Heading Sub-heading Alternative heading Alternative sub-heading Paragraphs are separated by a blank line. Two spaces at the end of a line produce a line break.
<pre>Text attributes <i>_italic_</i>, **bold**, '<code>monospace</code>'. Horizontal rule: ---</pre>	<pre><p>Text attributes italic, bold, <code>monospace</code>.</p> <p>Horizontal rule:</p> <hr /></pre>	Text attributes <i>italic</i>, bold, monospace . Horizontal rule:

Was ist Latex?

- Ein Softwarepaket zur Vereinfachung von TeX
- TeX ist ein von Donald Knuth ab 1977 entwickeltes Textsatzsystem mit Makrosprache
- LaTeX wurde von Leslie Lamport Anfang 1984 entwickelt
- LaTeX ist insbesondere für mathematische Berichte und Präsentationen (Beamer) nützlich

```
\footnotesize
\begin{theorem}[Datenverteilung des Allgemeinen Linearen Modells]
\justifying
\normalfont
Es sei
\begin{equation}
\upsilon = X\beta + \varepsilon \text{ mit } \varepsilon \sim N(0_n, \sigma^2 I_n)
\end{equation}
das ALM. Dann gilt
\begin{equation}
\upsilon \sim N(\mu, \sigma^2 I_n) \text{ mit } \mu := X\beta \in \mathbb{R}^n.
\end{equation}
\end{theorem}
```



Theorem (Datenverteilung des Allgemeinen Linearen Modells)

Es sei

$$v = X\beta + \varepsilon \text{ mit } \varepsilon \sim N(0_n, \sigma^2 I_n) \quad (7)$$

das ALM. Dann gilt

$$v \sim N(\mu, \sigma^2 I_n) \text{ mit } \mu := X\beta \in \mathbb{R}^n. \quad (8)$$

Quarto Guide



Overview Get Started **Guide** Extensions Reference Gallery Blog Help ▾



Guide ▾
 Authoring >
 Computations >
 Tools >
 Documents >
 Presentations >
 Dashboards >
 Websites >
 Books >
 Manuscripts >
 Interactivity >
 Publishing >
 Projects >
 Advanced >

Guide

Comprehensive guide to using Quarto. If you are just starting out, you may want to explore the [tutorials](#) to learn the basics.

Authoring

Create content with
 markdown
 Markdown Basics
 Figures
 Tables
 Diagrams
 Citations & Footnotes
 Cross References
 Article Layout

Computations

Execute code and display
 its output
 Using Python
 Using R
 Using Julia
 Using Observable
 Execution Options
 Parameters

Tools

Use your favorite tools
 with Quarto
 JupyterLab
 RStudio IDE
 VS Code
 Neovim
 Text Editors
 Visual Editor

Documents

Generate output in many
 formats
 HTML
 PDF
 MS Word
 Typst
 Markdown
 All Formats

Presentations

Present code and
 technical content
 Presentation Basics
 Reveal.js (HTML)
 PowerPoint (Office)
 Beamer (PDF)

Dashboards

Publish data with
 dashboards
 Dashboard Basics
 Layout
 Data Display
 Interactivity
 Deployment

Websites

Create websites and
 blogs
 Creating a Website
 Website Navigation
 Creating a Blog
 Website Search
 Website Listings

Books

Create books and
 manuscripts
 Creating a Book
 Book Structure
 Book Crossrefs
 Customizing Output

Manuscripts

Write and publish
 notebook-first scholarly
 articles
 Getting Started
 Authoring Manuscripts
 Publishing Manuscripts
 Using Manuscripts

Interactivity

Engage readers with
 interactivity
 Overview
 Observable JS
 Shiny
 Widgets
 Component Layout

Publishing

Publishing documents
 and sites
 Publishing Basics
 Quarto Pub
 GitHub Pages
 Posit Connect
 Posit Cloud
 Netlify
 Confluence
 Other Services

Projects

Scale up your work with
 projects
 Project Basics
 Managing Execution
 Project Profiles
 Environment Variables
 Project Scripts
 Virtual Environments

```
---
title: "Quarto Demonstration"
author: "Toni Demo"
date: today
format: pdf
---

# Überschrift zu Kapitel 1.

Hier steht der Text für Kapitel 1. Darin kann auch eine Abbildung enthalten sein.

{width="10%"}

## Überschrift zum Unterkapitel 1.1

Text kann fett, kursiv oder in monospace formatiert werden. Farben lassen sich mit \textcolor{} einbinden,
und Auflistungen über * oder - erstellen:

* \textcolor{blue}{blau}
* \textcolor{green}{grün}
- \textcolor{red}{rot}

Mathematische Ausdrücke werden durch Dollarzeichen in LaTeX formatiert, z.B. die Verteilung
einer Zufallsvariable  $\epsilon \sim N(\mu, \sigma^2 I_n)$  mit  $\mu := X\beta$  in  $\mathbb{R}^n$ .

Code Chunks können mit `echo` und `eval` ausgeführt und angezeigt werden (`eval = TRUE, echo = TRUE`):

```{r, eval = TRUE, echo = TRUE}
x <- 1:100
cat("Der Median der Variable beträgt:", median(x))
```

oder ausgeblendet und nicht ausgeführt werden (`eval = FALSE, echo = FALSE`):

```{r, eval = FALSE, echo = FALSE}
cat("Dieser Code wird weder angezeigt noch ausgeführt.")
```
```

Quarto Demonstration

Toni Demo

2026-04-08

Überschrift zu Kapitel 1.

Hier steht der Text für Kapitel 1. Darin kann auch eine Abbildung enthalten sein.



Überschrift zum Unterkapitel 1.1

Text kann **fett**, *kursiv* oder in `monospace` formatiert werden. Farben lassen sich mit `\textcolor{}` einbinden, und Auflistungen über `*` oder `-` erstellen:

- blau
- grün
- rot

Mathematische Ausdrücke werden durch Dollarzeichen in LaTeX formatiert, z.B. die Verteilung einer Zufallsvariable $v \sim N(\mu, \sigma^2 I_n)$ mit $\mu := X\beta \in \mathbb{R}^n$.

Code Chunks können mit `echo` und `eval` ausgeführt und angezeigt werden (`eval = TRUE`, `echo = TRUE`):

```
x <- 1:100  
cat("Der Median der Variable beträgt:", median(x))
```

Der Median der Variable beträgt: 50.5

oder ausgeblendet und nicht ausgeführt werden (`eval = FALSE`, `echo = FALSE`):

YAML Header

- Jede .qmd-Datei beginnt mit einem YAML Header (zwischen ---)
- Der YAML Header spezifiziert Metadaten und Formatierungsoptionen

```
---  
title: "Mein Bericht"  
author: "Toni Demo"  
date: today  
format: pdf  
bibliography: Referenzen.bib  
csl: apa.csl  
---
```

- format bestimmt das Ausgabeformat (pdf, html, beamer, etc.)
- bibliography und csl binden Literaturverwaltung ein (siehe Abschnitt Zotero)
- Weitere Formatierungsoptionen z.B. fontsize, margin, toc (Inhaltsverzeichnis), number-sections
- Übersicht verfügbarer YAML Optionen (nach Output-Format): [Quarto Reference](#)

Projekt YAML (`_quarto.yml`)

- Neben dem dokumentspezifischen YAML Header kann es eine projektweite `_quarto.yml`-Datei geben
- Diese liegt im Root directory des Projekts und gilt für alle `.qmd`-Dateien
- Einstellungen im Dokument-YAML überschreiben die Projekteinstellungen

```
project:
  title: "Mein Projekt"

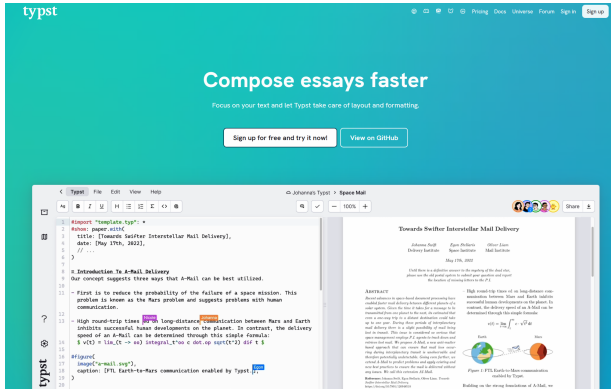
format:
  pdf:
    include-in-header: Header.tex

bibliography: Referenzen.bib
csl: apa.csl
```

- Praktisch um gemeinsame Einstellungen (z.B. Header, Bibliographie) nicht in jeder Datei wiederholen zu müssen

Beispielbericht

Beispielpräsentation



Quarto Typst Integration

- Guide
 - Authoring
 - Computations
 - Tools
- Documents
 - HTML
 - PDF
 - MS Word
 - Typst
 - Typst Basics
 - Custom Formats
- Markdown
- All Formats
- Presentations
- Dashboards
- Websites
- Books
- Manuscripts
- Interactivity
- Publishing
- Projects
- Advanced

Guide > Documents > Typst > Typst Basics

Typst Basics

Overview

[Typst](#) is a new open-source markup-based typesetting system that is designed to be as powerful as LaTeX while being much easier to learn and use. Typst creates beautiful PDF output with blazing fast render times.

Use the `typst` format to create a PDF document via Typst. For example:

```
hello-typst.qnd
---
title: "Hello Typst!"
format:
  typst:
    toc: true
    section-numbering: 1.1.a
    columns: 2
---
```

Rendering or previewing this document will invoke the Typst CLI to create `hello-typst.pdf`, a PDF file, from your markdown source file. Quarto includes the Typst CLI so no separate installation of Typst is required.

The above example highlights a few of the options available for Typst output. This document covers these and other options in detail. See the [Typst format reference](#) for a complete list of all available options.

One of the highlights of Typst is the ease of creating highly customized templates. For example, here are some Typst templates that you can use in Quarto as custom formats:



Learn more about how to use them, and how to create your own in [Custom Formats](#).

On this page

- Overview
- Known Limitations
- Page Layout
- Table of Contents
- Section Numbering
- Code Annotation
- Bibliography
- Typst Blocks
- Raw Typst
- Typst CSS
- Typst File (.typ)
- Fonts Support
- Computation Figure Format
- Includes

[Edit this page](#)
[Report an issue](#)

Quarto

Zotero

Tidyverse

Was ist ein Reference Manager?

- Reference Manager sind Literaturverwaltungsprogramme
- Reference Manager unterstützen Zitationen und das Erstellen von Literaturverzeichnissen
- Zitierstile können automatisch auf bestimmte Spezifikationen (z.B. APA) eingestellt werden
- Reference Manager dienen auch als digitale Bibliotheken
- Kommerzielle Reference Manager sind z.B. EndNote, Citavi, Mendeley und Papers
- Kostenlose/Freemium Reference Manager sind z.B. [JabRef](#) und [Zotero](#)
- Eine Integration in Quarto erlaubt z.B. der Export der eigenen Library in das [BibTeX](#) Format.

Zotero Website

Zotero Documentation

The image shows the Zotero website and a screenshot of the Zotero application interface. The website has a dark background with the text "Your personal research assistant" and "Zotero is a free, easy-to-use tool to help you collect, organize, annotate, cite, and share research." Below this is a "Download" button and text indicating availability for Mac, Windows, Linux, and iOS, along with a link to "Just need to create a quick bibliography? Try ZoteroBib". The application screenshot shows a sidebar with "My Library" and a list of items with columns for Title, Creator, and Year. The selected item is "The Emergence of Tropical Medicine in France" by Osborne, 2014.

zotero

Groups Documentation Forums Get Involved Log In Upgrade Storage

Your personal research assistant

Zotero is a free, easy-to-use tool to help you collect, organize, annotate, cite, and share research.

Download

Available for Mac, Windows, Linux, and iOS
Just need to create a quick bibliography? Try [ZoteroBib](#)

My Library

| Title | Creator | Year |
|--|-----------|------|
| Quatre, maladie, empire. Les services de santé militaires en... | Zajac | 2010 |
| Les Officiers de santé et les maladies coloniales à la Réunion (1890-1914) | Osborne | 2014 |
| The Emergence of Tropical Medicine in France | Osborne | 2014 |
| Colonial Disease, Translation, and Enlightenment: Franco-British... | Osborne | 2014 |
| Tracing in Groups through Philanthropy in the Eighteenth Century | Walker | 2010 |
| The Medicines Trade in the Portuguese Atlantic World, 1500-1700 | Walker | 2010 |
| Leopold and Savoy in Burundi: Gendered Settling and the Ph... | Walker | 2010 |
| Medical Experimentation and Race in the Eighteenth-century | Schneider | 2010 |
| The Circulation of Botanical Knowledge in the Eighteenth-century | Osborne | 2010 |
| Colonialism and the Emergence of Tropical Medicine in France | Osborne | 2014 |

Item Type: Journal Article
Title: Circulation of Medicine in the Early Modern Atlantic World
+ Author: Cook, Harold J.
Walker, Timothy D.
Abstract: The search for powerful drugs has inspired people and communities to move around the globe for many centuries, as it still does.
Publication: Social History of Medicine

BibTeX Export

- Better BibTeX Plugin installieren: `zotero-better-bibtex-[version].xpi`-Datei von retorque.re herunterladen; in Zotero: Tools → Plugins → Zahnrad-Symbol → Install Plugin From File...
- In Zotero: Rechtsklick auf Sammlung → "Export Collection"
- Format: BibTeX → Export als Referenzen.bib
- Die `.bib`-Datei im gleichen Ordner wie die `.qmd`-Datei speichern
- Im YAML-Header der `.qmd`-Datei einbinden:

```
bibliography: Referenzen.bib
csl: apa.csl
```

- APA Citation Style Language (CSL) Datei herunterladen: `apa.csl` via [Zotero Style Repository](https://www.zotero.org/styles)
- Die `apa.csl`-Datei ebenfalls im gleichen Ordner wie die `.qmd`-Datei speichern

Zitieren in Quarto

- Jeder Eintrag in der .bib-Datei hat einen eindeutigen Citation Key, z.B. `noether1918`
- Zitieren im Text mit `@noether1918` → Noether (1918)
- Zitieren in Klammern mit `[@noether1918]` → (Noether, 1918)
- Bei mehreren Autor:innen: `[@weber2020]` → (Weber et al., 2020)
- Mehrere Quellen: `[@noether1918; @weber2020]` → (Noether, 1918; Weber et al., 2020)
- Literaturverzeichnis wird automatisch am Ende des Dokuments eingefügt

Quarto

Zotero

Tidyverse

Tidyverse

Tidyverse

[Packages](#) [Blog](#) [Learn](#) [Help](#) [Contribute](#)



R packages for data science

The tidyverse is an opinionated [collection](#) of [R packages](#) designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

Learn the tidyverse

Cheat Sheets

Data transformation with dplyr : : CHEATSHEET



dplyr functions work with pipes and expect **tidy data**. In tidy data:

| Each variable is in its own column | Each observation , or case , is in its own row | x f(y) becomes f(x, y) |
|--|---|---|
| | | |

Summarize Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

| summary function |
|---|
| summarize(data, ...)
Compute table of summaries.
<code>mtcars > summarize(avg = mean(mpg))</code> |
| count(data, ..., wt = NULL, sort = FALSE, name = NULL) Count number of rows in each group defined by the variables in ... Also tally() , add_count() , add_tally()
<code>mtcars > count(cyl)</code> |

Group Cases

Use **group_by(data, ...)** to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.

| | |
|--|--|
| | <code>mtcars > group_by(cyl) > summarize(avg = mean(mpg))</code> |
|--|--|

Use **rowwise(data, ...)** to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidy cheat sheet for list-column workflow.

| | |
|--|---|
| | <code>starwars > rowwise() > mutate(film_count = length(film))</code> |
|--|---|

ungroup(x, ...) Returns ungrouped copy of table.
`g_mtcars <- mtcars %>% group_by(cyl) %>% ungroup(g_mtcars)`

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

| | |
|--|---|
| | filter(data, ..., preserve = FALSE) Extract rows that meet logical criteria.
<code>mtcars > filter(mpg > 20)</code> |
| | distinct(data, ..., keep_all = FALSE) Remove rows with duplicate values.
<code>mtcars > distinct(gear)</code> |
| | slice(data, ..., preserve = FALSE) Select rows by position.
<code>mtcars > slice(10:15)</code> |
| | slice_sample(data, ..., n, prop, weight, by = NULL, replace = FALSE) Randomly select rows. Use <code>n</code> to select a number of rows and <code>prop</code> to select a fraction of rows.
<code>mtcars > slice_sample(n = 5, replace = TRUE)</code> |
| | slice_min(data, order_by, ..., n, prop, with_ties = TRUE) and slice_max() Select rows with the lowest and highest values.
<code>mtcars > slice_min(mpg, prop = 0.25)</code> |
| | slice_head(data, ..., n, prop) and slice_tail() Select the first or last rows.
<code>mtcars > slice_head(n = 3)</code> |

Logical and boolean operators to use with filter()

| | | | | | | |
|----|---|----|----------|------|---|-------|
| == | < | <= | is.na() | %in% | | xor() |
| != | > | >= | !is.na() | ! | & | |

See **7base::Logic** and **7base::Comparison** for help.

ARRANGE CASES

| | |
|--|---|
| | arrange(data, ..., by, group = FALSE) Order rows by values of a column or columns (low to high), use with desc() to order from high to low.
<code>mtcars > arrange(mpg)</code>
<code>mtcars > arrange(desc(mpg))</code> |
|--|---|

ADD CASES

| | |
|--|--|
| | add_row(data, ..., before = NULL, after = NULL) Add one or more rows to a table.
<code>cars > add_row(speed = 1, dist = 1)</code> |
|--|--|

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

| | |
|--|--|
| | pull(data, var = 1, name = NULL, ...) Extract column values as a vector, by name or index.
<code>mtcars > pull(wt)</code> |
| | select(data, ...) Extract columns as a table.
<code>mtcars > select(mpg, wt)</code> |
| | relocate(data, ..., before = NULL, after = NULL) Move columns to new position.
<code>mtcars > relocate(mpg, cyl, after = last_col())</code> |

Use these helpers with select() and across()
e.g. `mtcars > select(mpg:cyl)`

| | | |
|---------------------------|---------------------------------------|-----------------------|
| contains(match) | num_range(prefix, range) | l_e.g. mpg:cyl |
| ends_with(match) | all_of(x) any of(x, ..., vars) | l_e.g. lpg |
| starts_with(match) | matches(match) | everything() |

MANIPULATE MULTIPLE VARIABLES AT ONCE

| | |
|--|--|
| | across(cols, funs, ..., names = NULL) Summarize or mutate multiple columns in the same way.
<code>df > summarize(across(everything(), mean))</code> |
| | across(cols) Compute across columns in row-wise data.
<code>df > rowwise() > mutate(totals = sum(c_across(1:2)))</code> |

MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).

| vectorized function |
|--|
| mutate(data, ..., keep = "all", before = NULL, after = NULL) Compute new column(s). Also add_column() .
<code>mtcars > mutate(gpm = 1 / mpg)</code>
<code>mtcars > mutate(gpm = 1 / mpg, keep = "none")</code> |
| rename(data, ...) Rename columns. Use rename_with() to rename with a function.
<code>mtcars > rename(miles_per_gallon = mpg)</code> |



Tidyverse dplyr

```
D <- read.csv("../Daten/Daten_1.csv")
```

```
# Daten einlesen
```

| Variable_1 | Variable_2 | Variable_3 |
|------------|------------|------------|
| 34.87 | 34.61 | 33.56 |
| 32.16 | 22.89 | 15.75 |
| 33.95 | 31.82 | 28.83 |
| 28.78 | 25.91 | 20.04 |
| 30.13 | 26.83 | 22.00 |
| 30.50 | 26.50 | 24.42 |
| 32.48 | 26.92 | 22.96 |
| 31.66 | 31.84 | 28.83 |
| 32.76 | 33.00 | 33.28 |
| 31.60 | 26.77 | 21.21 |
| 32.44 | 28.55 | 28.63 |
| 29.48 | 25.33 | 24.19 |
| 31.24 | 28.97 | 25.18 |
| 34.33 | 31.31 | 28.22 |
| 31.56 | 27.11 | 22.92 |
| 31.87 | 30.95 | 30.30 |
| 27.07 | 21.94 | 17.60 |
| 29.36 | 25.41 | 19.32 |
| 36.07 | 33.56 | 33.41 |
| 33.03 | 28.81 | 26.58 |
| 33.12 | 32.20 | 29.44 |

Tidyverse dplyr

Der Pipe operater %>% oder |> ermöglicht es, Funktionen in einer Reihe nacheinander auszuführen.

mutate() erlaubt das Erzeugen neuer Spalten als Funktionen bestehender Spalten.

```
library(dplyr)
n <- nrow(D)                                # Anzahl Beobachtungen
D_processed <- D %>%                         # D wird an nächste Funktion übergeben
  mutate(ID = seq(n)) %>%                   # ID-Spalte hinzufügen
  mutate(Summe = Variable_1 + Variable_2 + Variable_3) # Summen-Spalte hinzufügen
```

| Variable_1 | Variable_2 | Variable_3 | ID | Summe |
|------------|------------|------------|----|--------|
| 34.87 | 34.61 | 33.56 | 1 | 103.04 |
| 32.16 | 22.89 | 15.75 | 2 | 70.79 |
| 33.95 | 31.82 | 28.83 | 3 | 94.60 |
| 28.78 | 25.91 | 20.04 | 4 | 74.74 |
| 30.13 | 26.83 | 22.00 | 5 | 78.96 |
| 30.50 | 26.50 | 24.42 | 6 | 81.42 |
| 32.48 | 26.92 | 22.96 | 7 | 82.37 |
| 31.66 | 31.84 | 28.83 | 8 | 92.34 |
| 32.76 | 33.00 | 33.28 | 9 | 99.05 |
| 31.60 | 26.77 | 21.21 | 10 | 79.58 |
| 32.44 | 28.55 | 28.63 | 11 | 89.62 |
| 29.48 | 25.33 | 24.19 | 12 | 79.00 |
| 31.24 | 28.97 | 25.18 | 13 | 85.40 |
| 34.33 | 31.31 | 28.22 | 14 | 93.86 |
| 31.56 | 27.11 | 22.92 | 15 | 81.59 |
| 31.87 | 30.95 | 30.30 | 16 | 93.12 |
| 27.07 | 21.94 | 17.60 | 17 | 66.61 |
| 29.36 | 25.41 | 19.32 | 18 | 74.09 |
| 36.07 | 33.56 | 33.41 | 19 | 103.04 |
| 33.03 | 28.81 | 26.58 | 20 | 88.42 |
| 33.12 | 32.20 | 29.44 | 21 | 94.75 |

`filter()` erlaubt es, Zeilen gemäß bestimmten Bedingungen auszuwählen.

```
D_selected <- D_processed %>%  
  filter(ID %in% 1:10) %>%           # Auswahl der IDs 1-10  
  filter(Summe > 90)                 # Selektion der Beobachtungen mit Summe > 90
```

| Variable_1 | Variable_2 | Variable_3 | ID | Summe |
|------------|------------|------------|----|--------|
| 34.87 | 34.61 | 33.56 | 1 | 103.04 |
| 33.95 | 31.82 | 28.83 | 3 | 94.60 |
| 31.66 | 31.84 | 28.83 | 8 | 92.34 |
| 32.76 | 33.00 | 33.28 | 9 | 99.05 |

Data visualization with ggplot2 : : CHEATSHEET



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot (data = DATA) +  
  GEOM FUNCTION (mapping = aes(MAPPINGS),  
  size = SIZE, position = POSITION) +  
  COORDINATE FUNCTION +  
  THEME FUNCTION
```

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5" x 5" file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and fill - string ("red", "RRRRGGG")
linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "longdash", 5 = "longdash", 6 = "twodash")
size - integer (line width in mm)
shape - integer/shape name or a single character ("a")

color and fill - string ("red", "RRRRGGG")
linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "longdash", 5 = "longdash", 6 = "twodash")
size - integer (line width in mm)
shape - integer/shape name or a single character ("a")



Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unempLOY))
b <- ggplot(aes(x = long, y = lat))

a = geom_blank() and a = expand_limits() ensure limits include values across all plots.
b = geom_curve(aes(x = lat + 1, y = long + 1, curvature = 1)) - x, y, end, y, yend, alpha, angle, color, curvature, linetype, size
c = geom_path(linetype = "solid")
d = geom_path(linetype = "round", linetype = 2)
e = geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, linewidth, linetype, size
f = geom_rect(aes(xmin = long, ymin = lat, xmax = long + 2, ymax = lat + 2)) - xmin, xmax, ymin, ymax, alpha, color, fill, linetype, size
g = geom_ribbon(aes(ymin = unempLOY - 900, ymax = unempLOY + 900)) - x, ymin, ymax, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size
b = geom_abline(aes(intercept = 0, slope = 1))
c = geom_hline(aes(yintercept = lat))
d = geom_vline(aes(xintercept = long))
e = geom_segment(aes(xend = lat + 1, yend = long + 1))
f = geom_spoke(aes(angle = 1:155, radius = 2))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
a = geom_area(stat = "bin")
b = geom_area(aes(x = long, y = lat, linetype = size))
c = geom_density(binned = "gaussian")
d = geom_dotplot(aes(x = long, y = lat))
e = geom_freqpoly(aes(x = long, y = lat))
f = geom_histogram(binned = 3)
g = geom_histogram(binned = 2, width = 2)
h = geom_qq(aes(sample = hwy))
i = geom_qqline(aes(x = long, y = lat))

discrete

d <- ggplot(mpg, aes(class))
e = geom_bar()
f = geom_bar(aes(fill = size))
g = geom_bar(aes(fill = size))

TWO VARIABLES both continuous

e <- ggplot(mpg, aes(cty, hwy))
a = geom_label(aes(label = cty, nudje_x = 1, nudje_y = 2)) - x, y, label, alpha, angle, color, family, fontface, hjust, linewidth, size, vjust
b = geom_point()
c = geom_quantile()
d = geom_rug(aes(size = "lat"))
e = geom_smooth(method = lm)
f = geom_smooth(method = lm)
g = geom_text(aes(label = cty, nudje_x = 1, nudje_y = 2)) - x, y, label, alpha, angle, color, family, fontface, hjust, linewidth, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))
a = geom_col()
b = geom_boxplot()
c = geom_boxplot(aes(x = long, y = lat, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, width))
d = geom_dotplot(binned = "y", stackdir = "center")
e = geom_violin(aes(scale = "area"))
f = geom_violin(aes(scale = "area"))

both discrete

g <- ggplot(diamonds, aes(carat, color))
a = geom_count()
b = geom_jitter(height = 2, width = 2)
c = geom_jitter(height = 2, width = 2)

THREE VARIABLES

aes(class ~ aes(carat, color, long2 = delta, lat2)); f <- ggplot(diamonds, aes(long, lat))
a = geom_contour(aes(z = 2))
b = geom_contour_filled(aes(fill = 2))
c = geom_raster(aes(fill = 2))
d = geom_tile(aes(fill = 2))

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))
a = geom_bin2d(binned = 0.25, 5000)
b = geom_density_2d()
c = geom_density_2d()
d = geom_hex()
e = geom_hex()

continuous function

i <- geom_area()
j = geom_line()
k = geom_line()
l = geom_step(direction = "hv")
m = geom_step(direction = "hv")

visualizing error

d <- data.frame(murder = c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"),
murder = factor(murder),
murder = factor(murder))
a = geom_crosstab(aes(x = y, y = x, ymax, ymin, alpha, color, fill, group, linetype, size, width))
b = geom_errorbar(aes(x = y, y = x, ymax, ymin, alpha, color, fill, group, linetype, size, width))
c = geom_errorbar(aes(x = y, y = x, ymax, ymin, alpha, color, fill, group, linetype, size, width))
d = geom_errorbar(aes(x = y, y = x, ymax, ymin, alpha, color, fill, group, linetype, size, width))

maps

g <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
a = ggplot(data, aes(long, lat))
b = geom_map(aes(map_id = state), map = map)
c = expand_limits(aes(map_id = state), map = mapState)
d = geom_map(aes(map_id = state), map = mapState)

Beispieldatensatz

```
library(dplyr) # Für Pipe (%>%), mutate()

# Daten vorbereiten
D <- read.csv("../Daten/Daten_2.csv") # Daten einlesen
n_pat <- nrow(D) # Anzahl Patientinnen
D_processed <- D %>% # PatientIn ID hinzufügen
  mutate(PatientIn = seq(n_pat))
```

Die ersten 12 Zeilen des Dataframes:

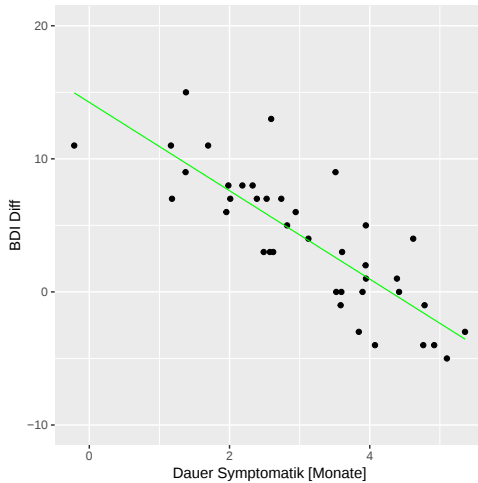
| DUR | BDI | PatientIn |
|------|-----|-----------|
| 1.37 | 9 | 1 |
| 2.18 | 8 | 2 |
| 1.16 | 11 | 3 |
| 3.60 | 0 | 4 |
| 2.33 | 8 | 5 |
| 1.18 | 7 | 6 |
| 2.49 | 3 | 7 |
| 2.74 | 7 | 8 |
| 2.58 | 3 | 9 |
| 1.69 | 11 | 10 |
| 3.51 | 9 | 11 |
| 2.39 | 7 | 12 |

Tidyverse ggplot2

```
library(ggplot2)                                # Für ggplot()

# Visualisierung
ggplot(
  data = D_processed,                           # Daten
  mapping = aes(x = DUR, y = BDI)               # Daten-Axen-mapping
) +
coord_cartesian(ylim = c(-10, 20)) +            # y-limits anpassen
geom_point() +                                  # Datenpunkte zeichnen
geom_smooth(                                    # Ausgleichsgerade zeichnen
  method = "lm",
  color = "green", se = F, linewidth = 0.4
) +
ylab("BDI Diff") + xlab("Dauer Symptomatik [Monate]") # Achsenbeschriftung
graphics.off()                                  # Schließt browser

ggsave(                                         # Abbildung speichern
  filename = "ggplot_beispiel.pdf",
  height = 5, width = 5
)
```



VS Code Website

VS Code-R Wiki

R for Data Science (2e)

ggplot2: Elegant Graphics for Data Analysis (3e)