



Programmieren für Psychologie

MSc Umweltpsychologie/Mensch-Technik-Interaktion

MSc Klinische Psychologie und Psychotherapie

MSc Psychologie

Joram Soch | Wintersemester 2026/2027

(2) R und VS Code

R und Visual Studio Code

Quarto und Markdown

Zotero

Tidyverse

R und Visual Studio Code

Quarto und Markdown

Zotero

Tidyverse

Was ist R?

- Eine Programmiersprache und ein Softwarepaket.
- Entwickelt von Ihaka and Gentleman (1996).
- Freier Dialekt der proprietären Software S (Becker, Chambers, and Wilks (1988)).
- Weiterentwickelt und gepflegt durch [R Core Team](#) und [R Foundation](#)
- Interpretierte imperative 4GL Sprache.
- Optimierte und populär für statistische Datenanalysen.
- Große Community mit etwa 20.000 beigetragenen [R-Paketen](#) (Erweiterungen)
- Evolviert und konservativ im Kern, konsistent und progressiv in R-Paketen.

Wie bekomme ich R?

Über cran.r-project.org die geeignete Version herunterladen und installieren.



CRAN
Mirror
What's new?
Search
[CRAN Team](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Task Views](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

Download and Install R
Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R: • Download R for Linux (Debian, Fedora/Redhat, Ubuntu) • Download R for macOS • Download R for Windows
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.
Source Code for all Platforms
Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it! • The latest release (2022-06-23, Funny-Looking Kid) R-4.2.1.tar.gz, read what's new in the latest version. • Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release). • Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports. • Source code of older versions of R is available here. • Contributed extension packages
Questions About R
• If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the [CRAN Repository Policy](#) and then use the [web form](#).

If this fails, send an email to CRAN-submissions@R-project.org following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

Was können wir mit R machen?

Was kann R?

- Datensätze laden, manipulieren, und speichern.
- Eine Vielzahl von Berechnungen an verschiedenen Datenstrukturen durchführen.
- Eine Vielzahl statistischer Analysemethoden auf Daten anwenden.
- Datenanalyseskripte schreiben und Abbildungen generieren.
- Präsentationen [RMarkdown](#) und Bücher [RBookdown](#) erstellen.
- Wissenschaftliche Berichte mit [Quarto](#) erstellen.

Was kann R (bisher) nicht so gut?

- In einer ansprechenden Umgebung programmieren (⇒ Visual Studio Code).
- Scientific Computing (⇒ Python, Matlab, Julia).
- Psychologische Experimente programmieren (⇒ Python, Matlab).

Wie bekomme ich Hilfe zu R?

- während der Programmierung und bei bekanntem Funktionsnamen über die Kommandozeile:

```
?mean          # Zeigt Hilfe zu der Funktion "mean()"
help(mean)      # Zeigt Hilfe zu der Funktion "mean()"
browseVignettes("knitr") # Zeigt Vignetten des Paktes "knitr"
browseVignettes()   # Zeigt Vignetten aller installierten Pakete im Browser
```

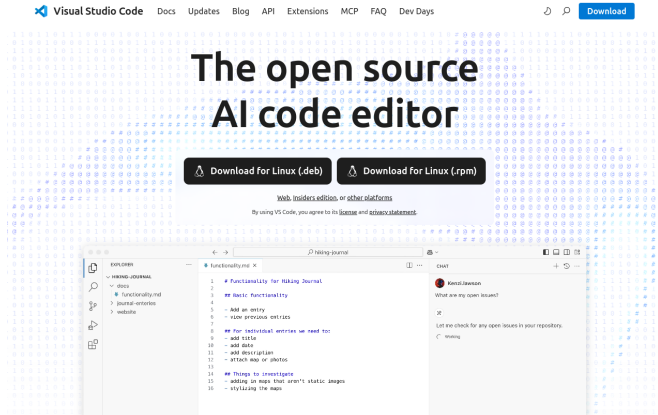
- Googlen
- stackoverflow.com
- r-project.org/help.html
- rstudio.com/resources/cheatsheets
- rseek.org
- r-bloggers.com

Was ist Visual Studio Code (VSCode)?

- VSCode ist ein kostenloser Quelltext-Editor von Microsoft.
- VSCode ist eine Softwareentwicklungsumgebung (Integrated Development Environment, IDE).
- Seit 2015 für Windows, macOS und Linux verfügbar.
- Seit 2018 ist VSCode der beliebteste Editor laut jährlicher [Stackoverflow-Umfragen](#).
- Ein Microsoftprodukt ist damit auch der beliebteste Editor der Linuxwelt.
- Über Extensions kann VSCode als IDE für beliebige Sprachen genutzt werden.
- Zum Beispiel funktioniert VSCode als IDE für R, Python, Julia, Shell, Quarto, etc.
- VSCode ist Community-based und sehr konfigurierbar.
- VSCode ist über Microsoft's GitHub über Endgeräte synchronisierbar.

Wie bekomme ich VSCode?

Über code.visualstudio.com herunterladen und installieren.



Wie benutze ich VSCode?

Ausführliche Dokumentation mit Anleitungen zur Nutzung: <https://code.visualstudio.com/docs>

 **Visual Studio Code** Docs Updates Blog API Extensions MCP FAQ Dev Days



Overview

SETUP

GET STARTED

CONFIGURE

EDIT CODE

BUILD, DEBUG, TEST

GITHUB COPILOT

SOURCE CONTROL

TERMINAL

LANGUAGES

NODE.JS / JAVASCRIPT

TYPESCRIPT

PYTHON

JAVA

C++

C#

CONTAINER TOOLS

DATA SCIENCE

INTELLIGENT APPS

AZURE

Visual Studio Code documentation

Get familiar with Visual Studio Code and learn how to code faster with AI.

Getting started



Download Visual Studio Code

Download VS Code for Windows, macOS, or Linux.



Getting started

Discover the key features of VS Code with the step-by-step tutorial.



Code faster with AI

Get started with GitHub Copilot, your AI coding assistant.

Code with rich features



Code in any language

Write code in your favorite programming language.



Version control

Built-in support for git and many other source control providers.



Debugging

Debug your code without leaving your editor.



Python



C++



C#

Wie benutze ich R in VSCode?

Anleitungen zum Arbeiten mit R in VSCode: <https://code.visualstudio.com/docs/languages/r>

 Visual Studio Code Docs Updates Blog API Extensions MCP FAQ Dev Days



Overview

SETUP

GET STARTED

CONFIGURE

EDIT CODE

BUILD, DEBUG, TEST

GITHUB COPILOT

SOURCE CONTROL

TERMINAL

LANGUAGES

Overview

JavaScript

JSON

HTML

Emmet

CSS, SCSS and Less

TypeScript

Markdown

PowerShell

C++

Java

PHP

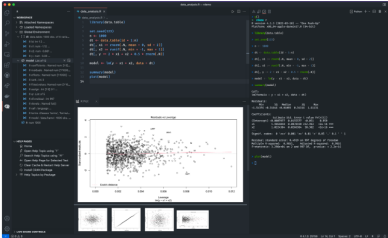
Python

R in Visual Studio Code

Edit

The [R programming language](#) is a dynamic language built for statistical computing and graphics. R is commonly used in statistical analysis, scientific computing, machine learning, and data visualization.

The [R extension](#) for Visual Studio Code supports extended syntax highlighting, code completion, linting, formatting, interacting with R terminals, viewing data, plots, workspace variables, help pages, managing packages and working with [R Markdown](#) documents.



ON THIS PAGE

Getting started

Running R code

Code completion (IntelliSense)

Linting

Workspace viewer

Debugging

Next steps

[RSS Feed](#)

[Ask questions](#)

[Follow #vscode](#)

[Request features](#)

[Report issues](#)

[Watch videos](#)

1. R installieren.
2. VSCode installieren.
3. VSCode für R startklar machen (R in Visual Studio Code).

Aufgaben für die Mittagspause

4. Mit dem eigenen oder Labor-Rechner vertraut machen.
 - a) Wie viel Festplatten- und Arbeitsspeicher (RAM) besitzt der Rechner? Notieren Sie die Werte in GB.
 - b) Welches Betriebssystem in welcher Version ist installiert? (z.B. Windows 11, macOS 14.5, Ubuntu 22.04)
 - c) Erstellen Sie einen Ordner für alle Materialien und Übungen dieses Kurses (z.B. "PDS_2026") an einem selbst gewählten Speicherort auf Ihrer Festplatte (nicht auf dem Desktop oder in Downloads).
 - d) Wie lautet der vollständige absolute Pfad zu diesem Kursordner? (z.B. "C:\Users\IhrName\Dokumente\PDS_2026" oder "/home/username/Documents/PDS_2026")
 - e) Laden Sie die Dateien "Beispiel_Datenanalyseskript_1.R" und "Beispieldaten.csv" herunter und speichern Sie diese in Ihrem Kursordner.
 - f) Testen Sie, mit welchen Programmen sich diese Dateien jeweils öffnen lassen und mit welchen nicht. Welche Programme zeigen den Inhalt sinnvoll lesbar an?

5. Erkunden Sie die VSCode User Interface und bearbeiten Sie nachstehenden Fragen (Tipp: Konsultieren Sie dafür die Online-Dokumentation [VS Code user interface](#)):
- a) Welche verschiedenen Views (Ansichten) können in der Activity Bar (vertikale Leiste ganz links) ausgewählt werden? Listen Sie mindestens fünf auf.
 - b) Wählen Sie in der *Activity Bar* den *Explorer* aus. Wie können Sie Ihren Kursordner in VSCode öffnen, sodass dessen Inhalt in der Primary Sidebar (Hauptseitenleiste) dauerhaft sichtbar ist?
 - c) Installieren Sie die R Extension für VSCode, falls noch nicht geschehen. Wählen Sie in der *Activity Bar* die *R Extension* aus. Welche Bereiche und Funktionen werden nun in der Primary Sidebar angezeigt?
 - d) Standardmäßig befindet sich das Panel (mit Terminal, R-Console, etc.) am unteren Bildschirmrand. Wie können Sie es so konfigurieren, dass das Panel stattdessen rechts (oder links) vertikal angeordnet ist?
 - e) Wie können Sie eine Secondary Sidebar öffnen und wofür könnte diese nützlich sein?

R und Visual Studio Code

Quarto und Markdown

Zotero

Tidyverse

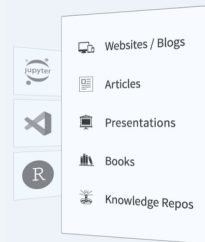
Quarto

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	7b07d62
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](#)

Markdown

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
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><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.
Text attributes <i>_italic_</i>, **bold**, <code>"monospace"</code>. Horizontal rule: ---	<pre><p>Text attributes italic, bold, <code>monospace</code>.</p> <p>Horizontal rule:</p> <hr /></pre>	Text attributes <i>italic</i>, bold, <code>monospace</code>. 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  $\epsilon$ 

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)

- Zusätzlich zum 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

R und Visual Studio Code

Quarto und Markdown

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 various icons representing research tools. The text on the website reads: "Your personal research assistant", "Zotero is a free, easy-to-use tool to help you collect, organize, annotate, cite, and share research.", and "Download". Below the download button, it says "Available for Mac, Windows, Linux, and ". A link "Just need to create a quick bibliography? Try [ZoteroBib](#)." is also present. The screenshot of the Zotero application shows a sidebar with "My Library" and "Book Reviews". The main window displays a list of books with columns for "Title", "Creator", and "Year". The selected item is "Circulation of Medicine in the Early Modern Atlantic World" by Cook and Walker, 2013. The right pane shows details for this item, including the item type "Journal Article", title "Circulation of Medicine in the Early Modern Atlantic World", author "Cook, Harold J.", and a description of the article's content.

zotero

Groups Documentation Forums Get Involved [Log In](#) [Upgrade Storage](#)

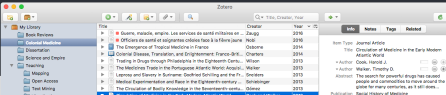
  Your personal research assistant  

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 .

Just need to create a quick bibliography? Try [ZoteroBib](#).



My Library

- Book Reviews
- Personal Knowledge
- Science and Empire
- Teaching
- Writing
- Open Access
- Test Mining
- Visualization

| Title | Creator | Year |
|--|-----------------|------|
| Quatre, marais, engins. Les services de santé militaires en... | Zhang | 2016 |
| Officers de santé et assignés colonies face à la fièvre jaune | Ngô | 2016 |
| The Emergence of Tropical Medicine in France | Osborne | 2014 |
| Colonial Disease, Translation, and Enlightenment | Francis-Sin | 2014 |
| Trading in Groups through Philadelphia in the Eighteenth Century | Wilson | 2013 |
| The Medicine Trade in the Portuguese Atlantic World | Walker | 2013 |
| Learning and Teaching in Botany: Botanical Gardening and the... | Swenson | 2013 |
| Medical Experimentation and Race in the Eighteenth-century | Reinberger | 2013 |
| The Circulation of Botany Knowledge in the Seventeenth-century | Osborne | 2013 |
| Circulation of Medicine in the Early Modern Atlantic World | Cook and Walker | 2013 |

Info Notes Tags Related

Item Type: Journal Article

Title: Circulation of Medicine in the Early Modern Atlantic World

+ Author: Cook, Harold J.

+ Author: Walker, Timothy D.

Abstract: The search for powerful drugs has caused 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

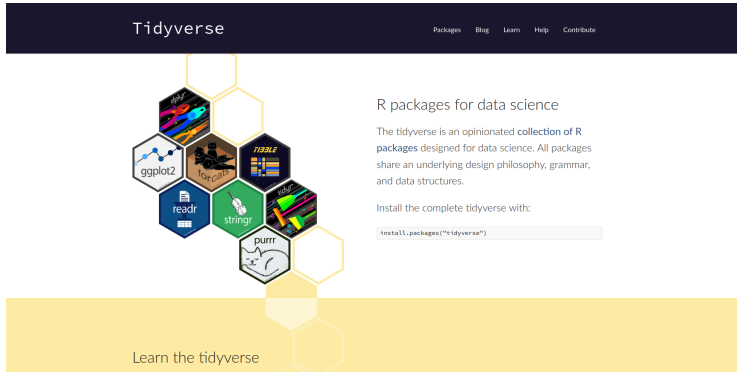
R und Visual Studio Code

Quarto und Markdown

Zotero

Tidyverse

Tidyverse



The screenshot shows the Tidyverse website. At the top is a dark blue header with the word "Tidyverse" in white on the left and navigation links "Packages", "Blog", "Learn", "Help", and "Contribute" on the right. Below the header is a large yellow hexagonal grid. A cluster of eight hexagons in the center contains logos for the following R packages: **ggplot2**, **readr**, **stringr**, **purrr**, **dplyr**, **tidyr**, **forcats**, and **tidbale**. To the right of this grid, the text reads: "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.", and "Install the complete tidyverse with:". Below this text is a light gray box containing the R code `install.packages("tidyverse")`. At the bottom of the page is a solid yellow bar with the text "Learn the tidyverse" on the left.

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)**

pipes

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.
mtcars > summarize(avg = mean(mpg))
```



```
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()
mtcars > count(cyl)
```

Group Cases

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



```
mtcars > group_by(cyl) > summarize(avg = mean(mpg))
```

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.



```
starwars > rowwise() > mutate(film_count = length(film))
```

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.
mtcars > filter(mpg > 20)
```



```
distinct(data, ..., keep_all = FALSE)
# Remove rows with duplicate values.
mtcars > distinct(gear)
```



```
slice(data, ..., preserve = FALSE)
# Select rows by position.
mtcars > slice(10:15)
```



```
slice_sample(data, ..., n, prop, weight, by = NULL, replace = FALSE)
# Randomly select rows. Use n to select a number of rows and prop to select a fraction of rows.
mtcars > slice_sample(n = 5, replace = TRUE)
```



```
slice_min(data, order_by, ..., n, prop, with_ties = TRUE) and slice_max()
# Select rows with the lowest and highest values.
mtcars > slice_min(mpg, prop = 0.25)
```



```
slice_head(data, ..., n, prop) and slice_tail()
# Select the first or last rows.
mtcars > slice_head(n = 3)
```

Logical and boolean operators to use with filter()

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

See [7base:Logic and 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.
mtcars > arrange(mpg)
mtcars > arrange(desc(mpg))
```

ADD CASES



```
add_row(data, ..., before = NULL, after = NULL)
# Add one or more rows to a table.
cars > add_row(speed = 1, dist = 1)
```

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.
mtcars > pull(wt)
```



```
select(data, ...)
# Extract columns as a table.
mtcars > select(mpg, wt)
```



```
relocate(data, ..., before = NULL, after = NULL)
# Move columns to new position.
mtcars > relocate(mpg, cyl, after = last_col())
```

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

| | | |
|--------------------|----------------------------------|-----------------|
| contains(match) | num_range(prefix, range) | 1, e.g. mpg:cyl |
| ends_with(match) | all_of(x) any_of(x, ..., vars) | 1, e.g. log |
| starts_with(match) | matches(match) | everything() |

MANIPULATE MULTIPLE VARIABLES AT ONCE

```
df <- tibble(x_1 = c(1, 2), x_2 = c(3, 4), y = c(4, 5))
```



```
across(cols, funs, ..., names = NULL)
# Summarize or mutate multiple columns in the same way.
df > summarize(across(everything(), mean))
```



```
c_across(cols)
# Compute across columns in row-wise data.
df > rowwise() > mutate(x_total = sum(c_across(1:2)))
```

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).



```
mutate(data, ..., keep = "all", before = NULL, after = NULL)
# Compute new column(s). Also add_column()
mtcars > mutate(gpm = 1 / mpg)
mtcars > mutate(gpm = 1 / mpg, keep = "none")
```



```
rename(data, ...)
# Rename columns. Use rename_with() to rename with a function.
mtcars > rename(miles_per_gallon = mpg)
```



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 (aes(mapping)) +  
  SCALE POSITION COORDINATE (FUNCTION) +  
  COORDINATE FUNCTION +  
  FACTOR FUNCTION +  
  SCALE 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", "RRRRGGGB")
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")
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 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



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(xend = lat + 1, xend = long + 1, curvature = 1)) - x, yend, alpha, angle, color, curvature, linetype, size
c = geom_path(aes(xend = "last", linetype = "round", linetype = 2))
d = geom_point(aes(x = alpha, color, fill, group, linetype, size))
e = geom_rect(aes(xmin = long, ymin = lat, xmax = long + 2, ymax = lat + 2)) - xmin, ymin, xmax, ymax, alpha, color, fill, linetype, size
f = 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
a = geom_abline(aes(intercept = 0, slope = 1))
b = geom_hline(aes(yintercept = lat))
c = geom_vline(aes(xintercept = long))
d = geom_segment(aes(xend = lat + 1, xend = long + 1))
e = geom_spoke(aes(angle = 1:155, radius = 2))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)
a = geom_area(aes(x = long, y = hwy))
b = geom_bar(aes(x = long, y = hwy))
c = geom_density(aes(x = long, y = hwy))
d = geom_dotplot(aes(x = long, y = hwy))
e = geom_freqpoly(aes(x = long, y = hwy))
f = geom_histogram(aes(x = long, y = hwy))
g = geom_line(aes(x = long, y = hwy))
h = geom_point(aes(x = long, y = hwy))
i = geom_smooth(aes(x = long, y = hwy))

discrete

d <- ggplot(mpg, aes(class))
a = geom_bar(aes(class))
b = geom_dotplot(aes(class))
c = geom_freqpoly(aes(class))
d = geom_histogram(aes(class))
e = geom_line(aes(class))
f = geom_point(aes(class))
g = geom_smooth(aes(class))

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, linetype, size, vjust
b = geom_point(aes(x = alpha, color, fill, shape, size, stroke))
c = geom_quantile(aes(x = alpha, color, group, linetype, size, weight))
d = geom_rug(aes(x = "lat", x = alpha, color, linetype, size))
e = geom_smooth(method = lm)(x = alpha, color, fill, group, linetype, size, weight)
f = geom_text(aes(label = cty, nudje, x = 1, nudje, y = 2)) - x, y, label, alpha, angle, color, family, fontface, hjust, linetype, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))
a = geom_bar(aes(class, hwy))
b = geom_boxplot(aes(class, hwy))
c = geom_dotplot(aes(class, hwy))
d = geom_histogram(aes(class, hwy))
e = geom_jitter(aes(class, hwy))
f = geom_violin(aes(class, hwy))

both discrete

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

THREE VARIABLES

aes(x = with(mtcars, sqrt(displacement) * delta, lat = 22); f <- ggplot(mtcars, aes(long, lat))
a = geom_contour(aes(x = z))
b = geom_contour_filled(aes(x = z))
c = geom_raster(aes(x = z, y = y, fill = z))
d = geom_tile(aes(x = z, y = y, fill = z))

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))
a = geom_bin2d(binsize = c(0.25, 500))
b = geom_density_2d(aes(x = alpha, color, group, linetype, size))
c = geom_hex(aes(x = alpha, color, fill, size))

continuous function

i <- ggplot(economics, aes(date, unempLOY))
a = geom_area(aes(x = alpha, color, fill, linetype, size))
b = geom_line(aes(x = alpha, color, group, linetype, size))
c = geom_step(aes(x = alpha, color, group, linetype, size))

visualizing error

d <- data.frame(p = c("A", "B"), fit = 4.5, se = 1.2)
j <- ggplot(d, aes(p, fit, ymin = fit - se, ymax = fit + se))
a = geom_crausertion(aes(x = y, ymin, ymax, alpha, color, fill, group, linetype, size))
b = geom_errorbar(aes(x = y, ymin, ymax, alpha, color, fill, group, linetype, size, width))
c = geom_errorbarh(aes(x = y, ymin, ymax, 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(lon, lat))
b = geom_map(aes(map_id = state), map = map)

Beispieldatensatz

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

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

Die ersten 12 Zeilen des Dataframes:

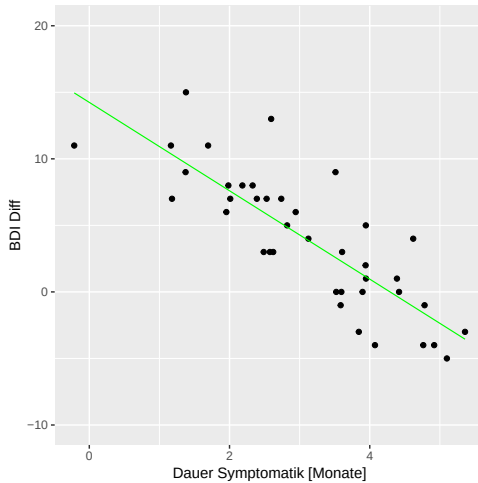
| DUR | dBDI | PatientIn |
|------|------|-----------|
| 1.37 | 9 | 1 |
| 2.18 | 8 | 2 |
| 1.16 | 11 | 3 |
| 3.60 | 0 | 4 |
| 2.33 | 8 | 5 |
| 1.18 | 7 | 6 |
| 2.49 | 3 | 7 |
| 2.74 | 7 | 8 |
| 2.58 | 3 | 9 |
| 1.69 | 11 | 10 |
| 3.51 | 9 | 11 |
| 2.39 | 7 | 12 |

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)

- Becker, Richard A., John M. Chambers, and Allen Reeve Wilks. 1988. *The New S Language: A Programming Environment for Data Analysis and Graphics*. Reprint. London: Chapman & Hall.
- Ihaka, Ross, and Robert Gentleman. 1996. "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics* 5 (3): 2999–2314.