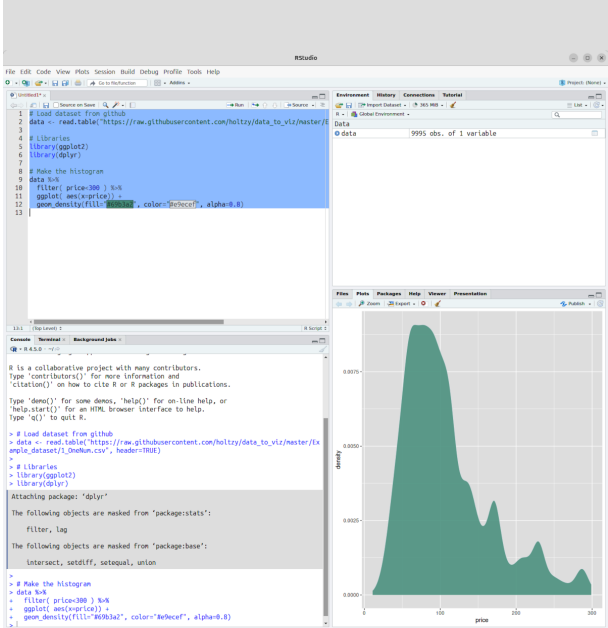
Einführung in R

[Termin 1]

Miguel Alvarez

kamapu.net

15. Oktober 2025



Miguel Alvarez

- Geobotaniker
- Vegetationsökologist
- Kartograph
- Computerfreak
  - Statistisches Programmieren
  - Data-Science/KI
  - GIS

https://kamapu.net/

R

Statistische Programmiersprache

- Programmiersprache
  - Mathematik/Statistik
  - Visualisierung (Grafiken)
- Interface/Terminal
- REPL-Umgebung (Read-Eval-Print-Loop)

- Open Source
- Freeware

R

```r
# Arithmetik
5 * (20 + 8)
```

```
[1] 140
```

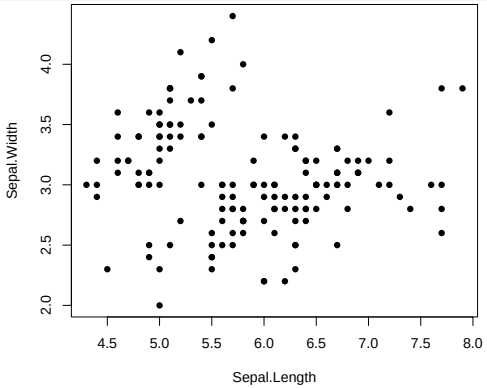
```r
# Statistik
summary(iris$Sepal.Length)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.
  4.300   5.100   5.800   5.843   6.400
   Max.
  7.900
```

```r
# Eulersche Identität
round(exp(1i * pi) + 1)
```

```
[1] 0+0i
```

```r
# Visualisierung
plot(Sepal.Width ~ Sepal.Length,
  data = iris,
  pch = 16)
```



R

Geschichte

- S (1975)
  - S-PLUS
  - TIBCO Spotfire S+
- R (1992)

R ist die kostenlose Alternative zu **S**

Ross Ihaka
Robert Gentleman

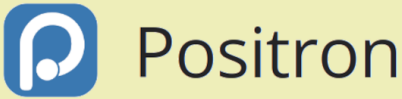
R versus Python

- Spezialist für Statistiken	- Allgemeine Verwendung
- Vektor als zentrale Datenstruktur	- Liste als zentrale Datenstruktur
- Index fängt bei 1	- Index fängt bei 0
- In der Regel interaktiv	- In der Regel nicht interaktiv

Beide Sprachen sind Bestandteile von Project Jupyter

R IDEs

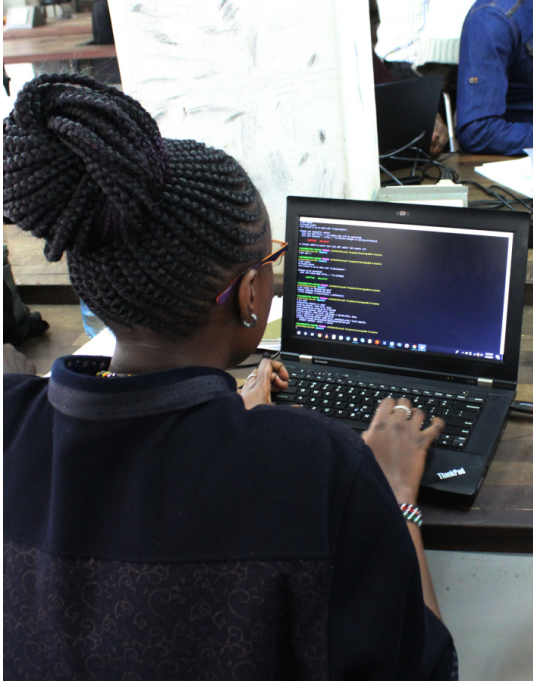
- **RStudio**
- **Positron**
- **Tinn-R**
- **Visual Studio Code**
  - R Extension



R Anwendung

Einführungskurse in **R**

- ☒ Syntax
- ☒ Schleifen
- ☒ Funktionen
- ☒ `source()`
- ☐ Pakete
- ☐ Objekt-Orientiertes Programmieren
- ☐ Literarisches Programmieren



R Anwendung

Arbeit mit **Skripten**

- Dokumentation
- Reproduzierbarkeit
- Automatisierung

```r
# Laden von Paket dplyr
library(dplyr)

# Laden von Datensatz iris
data(iris)

# Übersicht des Datensatzes
glimpse(iris)
```

```
Rows: 150
Columns: 5
$ Sepal.Length <dbl> 5.1, 4.9, 4.7, 4.6,~
$ Sepal.Width  <dbl> 3.5, 3.0, 3.2, 3.1,~
$ Petal.Length <dbl> 1.4, 1.4, 1.3, 1.5,~
$ Petal.Width  <dbl> 0.2, 0.2, 0.2, 0.2,~
$ Species      <fct> setosa, setosa, set~
```

R Pakete

Comprehensive R Archive Network
CRAN

```
install.packages()
```

Weitere Repositorien

```
devtools::install_github()
devtools::install_gitlab()
devtools::install_bitbucket()
```

CRAN
Mirrors
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

Donations
Donate

The Comprehensive R Archive Network

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 (2025-06-13, Great Square Root) R-4.5.1.tar.gz, read what's new in the latest version.
- The CRAN directory src/base-prerelease contains R alpha, beta, and rc releases as daily snapshots in time periods before a planned release.
- Between releases, the same directory src/base-prerelease contains snapshots of current patched and development versions. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Alternatively, daily snapshots are available here.
- 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.

Supporting CRAN

- CRAN operations, most importantly hosting, checking, distributing, and archiving of R add-on packages for various platforms, crucially rely on technical, emotional, and financial support by the R community.

  Please consider making financial contributions to the R Foundation for Statistical Computing.

What are R and CRAN?

OOP

S3

Ohne formale Klassendefinition.

S4

Mit formaler Definition von Klassen.
Klassen bestehen aus Slots.
Validierung.
(Siehe hier)

R6

Kapselung und mutable Objekten (wie bei Python).
Implementiert in Paket R6.

R6 2.6.1 Reference Articles Changelog Search for

Introduction

Source: vignettes/Introduction.Rmd

The R6 package provides an implementation of encapsulated object-oriented programming for R (also sometimes referred to as classical object-oriented programming). It is similar to R's reference classes, but it is more efficient and doesn't depend on S4 classes and the *methods* package.

R6 classes

R6 classes are similar to R's reference classes, but are lighter weight, and avoid some issues that come along with using S4 (R's reference classes are based on S4). For more information about speed and memory footprint, see `vignette("Performance")`.

Unlike many objects in R, instances (objects) of R6 classes have reference semantics. R6 classes also support:

- public and private methods
- active bindings
- inheritance (superclasses) which works across packages

Why the name R6? When R's reference classes were introduced, some users, following the names of R's existing class systems S3 and S4, called the new class system R5 in jest. Although reference classes are not actually called R5, the name of this package and its classes takes inspiration from that name.

The name R5 was also a code-name used for a different object system started by Simon Urbanek, meant to solve some issues with S4 relating to syntax and performance. However, the R5 branch was shelved after a little development, and it was never released.

Basics

Here's how to create a simple R6 class. The `public` argument is a list of items, which can be functions and fields (non-functions). Functions will be used as methods.

```
library(R6)
```

ON THIS PAGE

- R6 classes
  - Basics
  - Private members
  - Active bindings
  - Inheritance
  - Fields containing reference objects
- Other topics

Sauber Codieren

lintr

```
lintr::lint_package()
```

styler

```
styler::style_pkg()
```

goodpractice

```
goodpractice::gp()
```

lintr

R-CMD-check passing | codecov 99% | CRAN 3.2.0 | lifecycle stable

lintr provides static code analysis for R. It checks for adherence to a given style, identifying syntax errors and possible semantic issues, then reports them to you so you can take action. Watch lintr in action in the following animation:

lintr is complementary to the {styler} package which automatically restyles code, eliminating some of the problems that lintr can detect.

Installation

Install the stable version from CRAN:

```
install.packages("lintr")
```

Links

View on CRAN
Browse source code
Report a bug

License

Full license
MIT + file LICENSE

Community

Contributing guide
Code of conduct
Getting help

Citation

Citing lintr

Developers

Jim Hester
Author

Florent Angly
Author

Russ Hyde
Author

Michael Chirico
Author, maintainer

Kun Ren
Author

Alexander Rosenstock
Author

Indrajeet Patil
Author

More about authors...


Fragen?