Programmieren mit R für Einsteiger

2. Datentypen / 2.5 Pakete

Berry Boessenkool 2022-02-25 11:40

R Pakete

- Viele R Nutzer schreiben Code für spezifische Aufgaben.
- Wenn das auch für andere nützlich sein könnte, wird das oft in einem R Paket verpackt. Das ist eine vorgegebene Form für Quellcode inkl. Dokumentation, Anleitung und Beispiele.
- Wenn umfangreiche Auflagen erfüllt sind, kann das auf CRAN veröffentlicht werden (Comprehensive R Archive Network).
- Dort sind >18'000 Pakete verfügbar, siehe CRAN Task Views

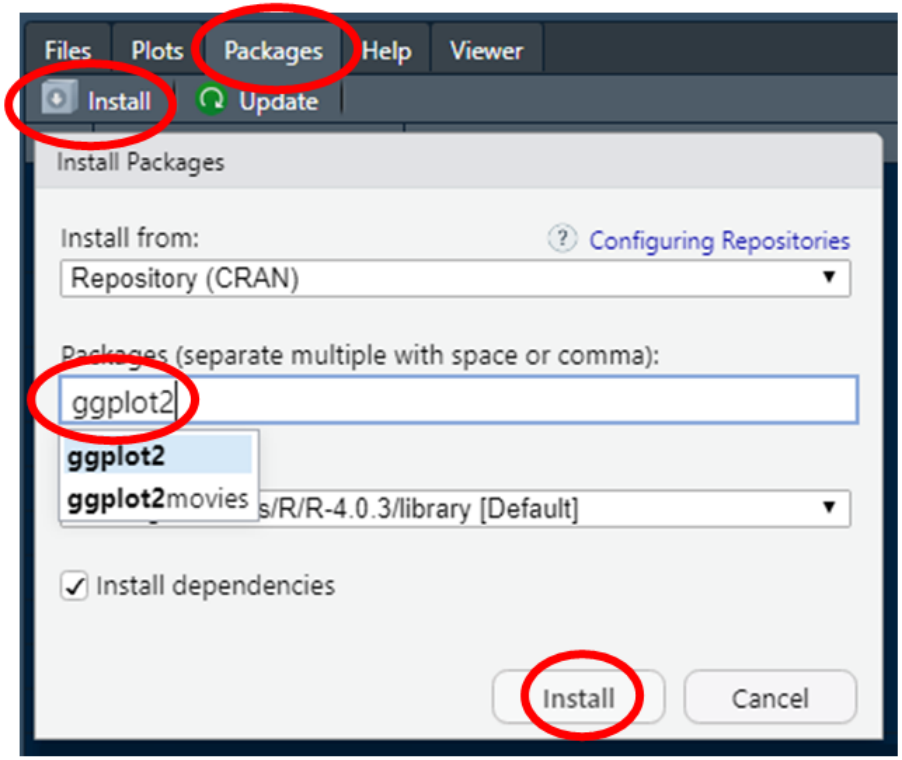
R Pakete installieren

Ein R Package downloaden und installieren:

```
install.packages("ggplot2")
```

Muss nur einmal ausgeführt werden, braucht keine admin Rechte.

Kann auch manuell in Rstudio gemacht werden:



Selten benötigt: Rückgängig machen mit `remove.packages("packagename")`

R Pakete nutzen

Ein Paket aus der lokalen Bibliothek (library) laden:

```
library("ggplot2")
```

Ist in jeder neuen R Session benötigt.
Sollte als Code im Skript stehen, damit das Skript reproduzierbar ist.
Danach sind die Funktionen aus dem Paket direkt nutzbar.

Bei gleichnamigen Funktionen in mehreren Paketen wird das verwendet, welches zuletzt geladen wurde.
Damit klar ist, aus welchem Package eine Funktion verwendet wird, ist die `Paket::Funktion()` Struktur

```
rdwd::findID("Potsdam")
```

eindeutiger (und sicherer) als

```
library(rdwd)
findID("Potsdam")
```

Ein guter Tipp: regelmäßig updaten - mit dem Rstudio button oder

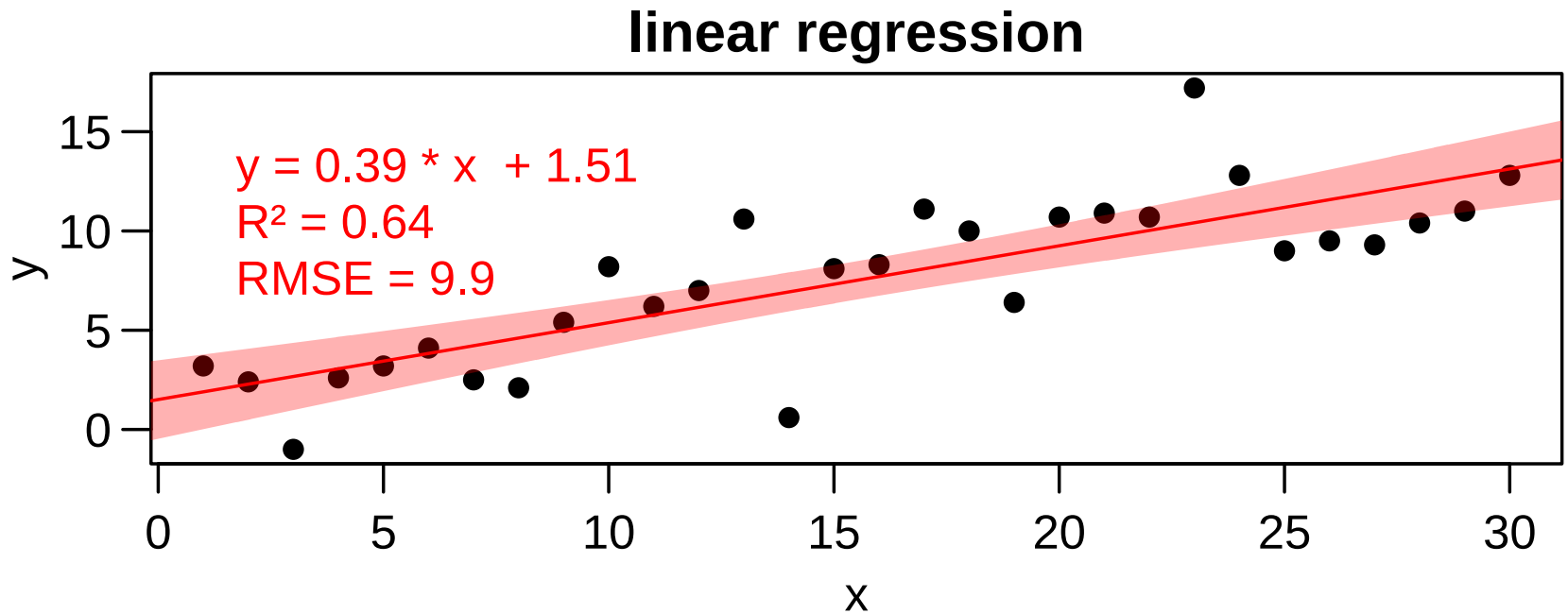
```
update.packages()
```

R Pakete bei Bedarf automatisch installieren

```r
# Wenn ein Paket nicht verfügbar ist, dann installiere es:
if(!requireNamespace("berryFunctions", quietly=TRUE))
  install.packages("berryFunctions")
```

Mehr unter bookdown.org/brry/course/packages

```r
x <- 1:30
y <- c(3.2, 2.4, -1, 2.6, 3.2, 4.1, 2.5, 2.1, 5.4, 8.2,
       6.2, 7, 10.6, 0.6, 8.1, 8.3, 11.1, 10, 6.4, 10.7,
       10.9, 10.7, 17.2, 12.8, 9, 9.5, 9.3, 10.4, 11, 12.8)
berryFunctions::linReg(x, y, pos1="topleft")
```



vorinstallierte Pakete / Quellcode von R Befehlen

Immer mitgelieferte Pakete (base R):

- geladen: `base`, `datasets`, `utils`, `grDevices`, `graphics`, `stats`, `methods`
- manuell zu laden: `compiler`, `grid`, `parallel`, `splines`, `tcltk`, `tools`

Definition (Source code) einer Funktion anschauen:

- ohne Klammern aufrufen: `append` statt `append()`
- auf github öffnen: `berryFunctions::funSource` (`F7` mit rskey)
  - base R: github.com/wch/r-source/src/library/base (stats, parallel, ...)
  - CRAN: github.com/cran
- `head` - UseMethod - `methods(head)` - zB `head.matrix` (mehr)
- `abs` - .Primitive - `do_abs` in src/main/names.c und /arithmetic.c

Online Hilfe CRAN package Funktionen (ohne installieren):
www.rdocumentation.org

Zusammenfassung

R Pakete mit weiterführendem Code (auf CRAN auch geprüft):

- `install.packages` (einmalig, manuell OK)
- `library` (in jedem Script)
- `Paket::Funktion()` macht Ursprung einer Funktion deutlich
- `requireNamespace` prüft, ob ein Paket geladen werden kann
- `berryFunctions::funSource` für Quellcode auf github