Statistik und Graphiken mit R

[Termin 1]

Miguel Alvarez

27. Februar 2024

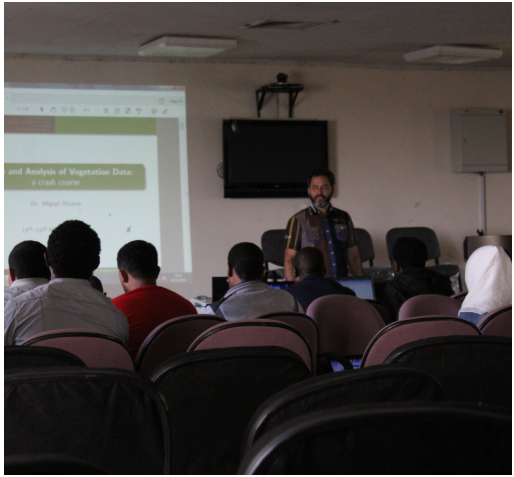
Der Kurs

18:00 – 20:30

Pause 19:15 – 19:30

- Methoden
  - Vortrag
  - Life-Codierung
  - Übungen

https://kamapu.github.io/GrundkursR/



Der Kurs

- **Termin 1 & 2**
  - Grundlagen
  - Datentypen
- **Termin 3 & 4**
  - Objekten (Datenstrukturen)
  - Lesen und Schreiben
- **Termin 5 & 6**
  - Statistiken
  - Graphiken (1)
- **Termin 7 & 8**
  - Graphiken (2)
  - Fortgeschrittenes Programmieren
  - Abschluss

Der Kurs

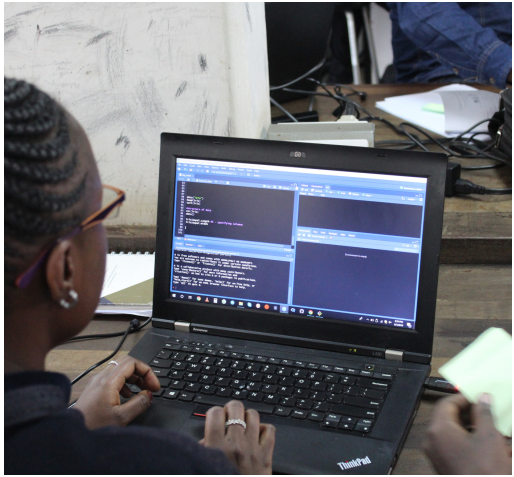
Miguel Alvarez

- Vegetationsökologist
- Datenwissenschaftler
- KI-Experte
- R (und Python) Programmierer
  - Umwelt und Biodiversität
  - Geographische Informationssystheme
  - Datenbanken
  - Reproduzierbarkeit

Open Source Aktivist

Die Teilnehmer?

- Warum R?
- Erwartung von dem Kurs



Geschichte

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

R ist die kostenlose Alternative zu **S**

Ross Ihaka
Robert Gentleman

Was ist R?

- Programmiersprache
  - Open Source (offene Quelle)
  - Freeware (kostenlos)
- Statistische Umgebung
  - Interface
  - Terminal
  - *Editoren*

https://www.r-project.org/

Einführung in R

Was ist R?

- Statistisches Programmieren
  - Mathematik
  - Grafiken
- Kommandozeilen
- REPL (Read-Eval-Print-Schleife)

```
(5 + 100) * 25
```

```
## [1] 2625
```

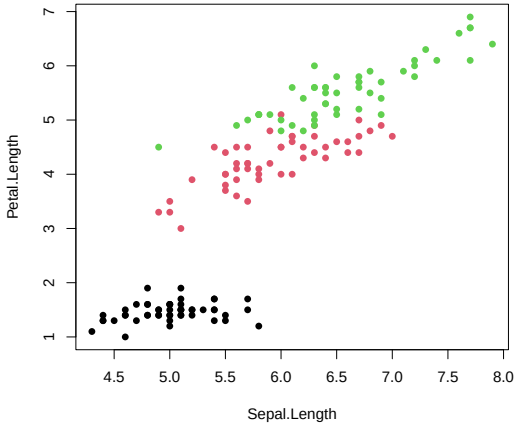
```
50/10
```

```
## [1] 5
```

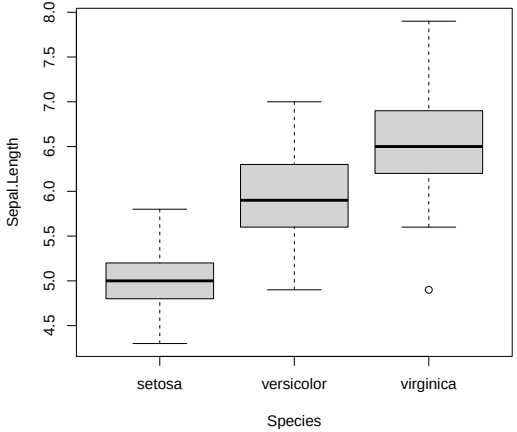
```
60 > 15
```

```
## [1] TRUE
```

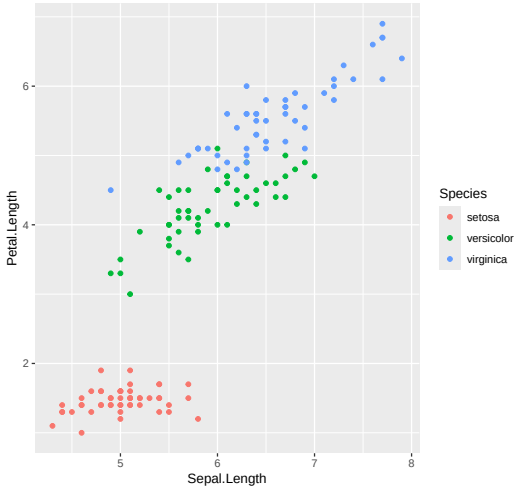
```
plot(Petal.Length ~ Sepal.Length,
      pch = 16, col = Species,
      data = iris)
```



```
boxplot(Sepal.Length ~ Species,
    data = iris)
```



```
library(ggplot2)
ggplot(iris,
    aes(x = Sepal.Length,
        y = Petal.Length,
        color = Species)) +
  geom_point()
```



Warum R?

- Kostet nichts
- Steuerung von Analysen
- Skripten
  - Protokolle
  - Reproduzierbarkeit
- Vielseitig

```
# Wir laden zunächst die Daten
data(iris)

# Wir berechnen die Mittelwerte für
# Länge und Breite von Petalen
# per Schwertlilienart
aggregate(cbind(Petal.Length, Petal.Width) ~ Species,
    FUN = mean, data = iris)
```

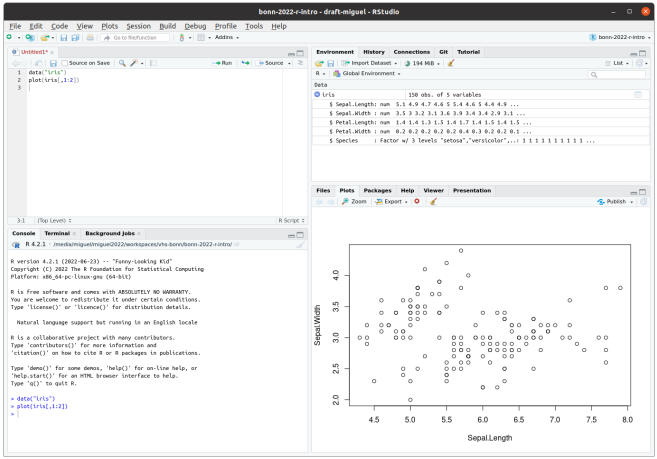
```
##      Species Petal.Length Petal.Width
## 1     setosa        1.462       0.246
## 2 versicolor        4.260       1.326
## 3  virginica        5.552       2.026
```

Wichtige Elemente

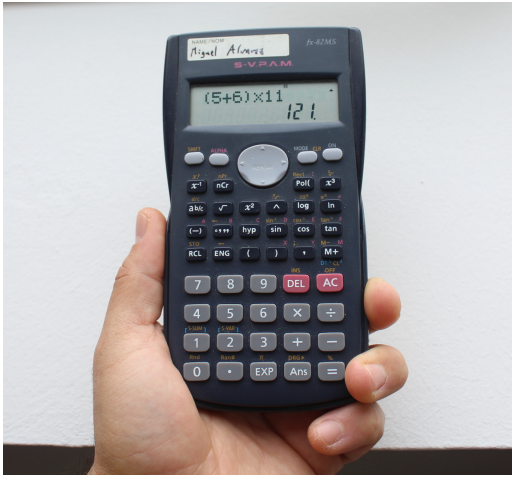
- Konsole
- Sitzung
- Workspace (Environment)
- Arbeitsverzeichnis (working directory)
- Skript
- Paket (Package)

RStudio

- Text Editor
- IDE (integrated development environment)



R als Taschenrechner



- Mathematische Operatoren
- Klammern

```
5+6*11
```

```
## [1] 71
```

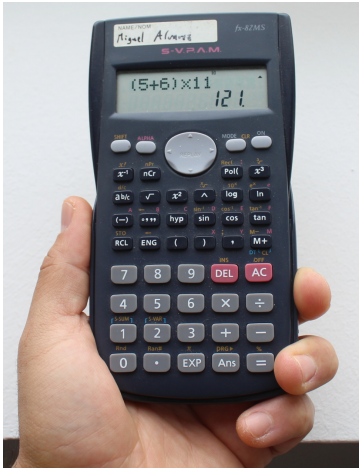
```
(5+6)*11
```

```
## [1] 121
```

- Zuweisung <-

```
M <- 5
M + 2
```

```
## [1] 7
```



Aufgabe 1

Haben/Hatten Sie einen Taschenrechner?
Erzählen Ihre Geschichte.

Aufgabe 2

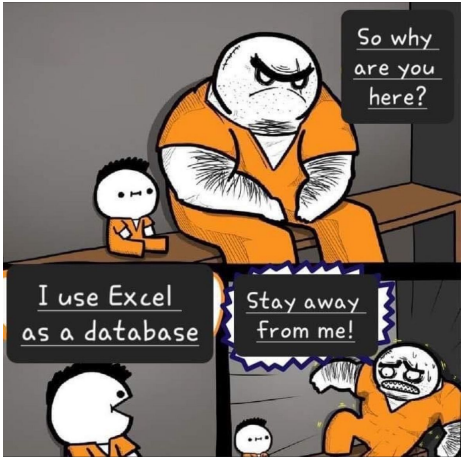
Taschenrechnen mit R

Logische Operatoren


```
==
!=
>
>=
<
<=
&
|
%in%
!
any()
all()
```

```
10 > 15
## [1] FALSE
10 < 15
## [1] TRUE
```

Daten in R



Which types of data do you know?

Daten in R

Vektoren

Der Vektor ist die grundlegende Datenstruktur in **R**

- Länge `length()`
- Typ `class()`
- Evtl. Namen `names()`

```
c(1:10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
rep(5, times = 10)
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5
```

```
LETTERS[1:5]
```

```
## [1] "A" "B" "C" "D" "E"
```

Daten in R

Indexieren

- Eckige Klammern
- Index
  - integer
  - logical (Bedingung)
  - character (Namen)

```
# Mit integer
letters[15]
```

```
## [1] "o"
```

```
# Mit logischen Werten
letters[!letters %in% c("a", "b", "c")]
```

```
##  [1] "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
## [20] "w" "x" "y" "z"
```

```
# Mit Namen
names(letters) <- letters
letters["m"]
```

```
##   m
## "m"
```

Daten in R

Datentypen

- integer
- numeric
- logical
- factor
- character

```
A <- c(1:10)
is.numeric(A)
```

```
## [1] TRUE
```

```
B <- as.character(A)
B
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"
```

```
is.numeric(B)
```

```
## [1] FALSE
```

Daten in R

Sonderklassen

- NA
- NaN
- NULL
- Inf
- -Inf

```
5/0
```

```
## [1] Inf
```

```
log(0)
```

```
## [1] -Inf
```

```
sqrt(-1)
```

```
## Warning in sqrt(-1): NaNs produced
## [1] NaN
```

Vielen Dank!

```
library(fortunes)
fortune(283)
```

```
##
## The good way to do it is to include the following comment at the beginning:
## # This is a holy Script, please edit it not
##    -- Kenn Konstabel (on "... how to protect R Script files from inadvertent
##       editing by users.")
##       R-help (April 2011)
```