Statistische Software (R)

Paul Fink, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Datensatz-Aufbereitung

Datenbeispiel

chickwts: Datensatz über das Gewicht in *Gramm* von 71 Küken, gefüttert mit 6 verschiedenen Beimischungen

ges: Vektor des Geschlechts der Küken (selbst gebaut)

Erzeugung der notwendigen Variablen

```
> data <- chickwts
> ges <- factor(sample(c("m", "f"), size = 71, replace = TRUE))
```

Übersicht über Variablennamen im Datensatz:

```
> names(data)
[1] "weight" "feed"
```


Zugriff auf Variablen in `data.frame`

- Listenzugriff mit Name der Variable

  ```
  data$weight
  ```

- Matrixzugriff mit Name der Variable

  ```
  data[, "weight"]
  ```

- Matrixzugriff mit Index der Variable

  ```
  data[, 1]
  ```

Vorteil von Matrixzugriff: Man kann auch mehrere Variablen gleichzeitig herausholen

Zugriff auf bestimmte Beobachtungen

Indizes bekannt: Matrixzugriff mit Indizes

```
data[c(1, 4, 20), ]
```

Indizes unbekannt: Beobachtungen nach Kriterien finden

- Matrixzugriff:

  ```
  data[data$feed == "casein", ]
  ```

- Funktion `subset()`:

  ```
  subset(data, feed == "casein")
  ```

Logische Operatoren & Verknüpfungen

==, !=	gleich, ungleich
>, >=	größer, größer gleich
<, <=	kleiner, kleiner gleich
!	Negation (nicht)
&, &&	und
|, ||	oder
xor()	entweder oder (ausschließend)
TRUE, T	wahr
FALSE, F	falsch

Nur '&&' und '||' sind **nicht** vektorwertig.

Die Funktion subset()

Befehlssyntax:

```
subset(Data.frame-Objekt, Kriterium an Variablen)
```

Beispiele:

```
subset(data, feed == "casein")
subset(data, feed %in% c("casein", "linseed"))
subset(data, (feed == "casein") & (weight > 240))
```

Spart Schreibarbeit bei komplizierteren Kriterien

Neue Variablen hinzufügen

- Listenzugriff:

  ```
  data$gender <- ges
  ```

- data.frame() Funktion:

  ```
  data <- data.frame(data,ges)
  ```

- cbind() Funktion:

  ```
  data <- cbind(data,ges)
  ```

- Matrixzugriff:

  ```
  data[, 3] <- ges
  ```

`length(ges) < nrow(data)` $\Longrightarrow$ Recycling-Regel

Achtung: Bestehende Variablen werden überschrieben!

Variablen aus data.frame löschen

- Listenzugriff:

  ```
  data$gender <- NULL
  ```

- Mit Matrixzugriff herausfiltern

  ```
  data <- data[, -3]
  ```

Ändern von Variablen in einem `data.frame`

Vorgehen anhand von Beispiel:

Das Gewicht soll in *kg* umgewandelt werden

```
# 1. Variable aus data.frame in Hilfsobjekt speichern
h <- data[,"weight"]
# 2. Aenderungen am Hilfsobjekt durchfuehren
h <- h / 1000
# 3. Hilfsobjekt als Variable in data.frame aufnehmen
data[,"weight"] <- h
```

oder einfacher:

```
# Alle Schritte laufen intern ab
data <- transform(data, weight = weight / 1000)
```


Weitere nützliche Funktionen

Beobachtungen der Größe nach ordnen:

`daten[,order(V1, V2, usw. )]`

`daten` wird nach *V1* geordnet, wenn Werte gleich, dann nach *V2*, usw.

Beobachtungen aggregieren:

`aggregate(zuaggVar ~ nachFaktorVar, FUN = Funktion, data = daten)`

Zum Besipiel für das Gruppenmittel die Funktion `mean()`.

Schreibarbeit sparen:

`with(daten, ... )`

Innerhalb von ... stehen einem direkt die Variablennamen von `daten` zur Verfügung

Aufgaben

1. Lesen Sie den Datensatz *nba.asc* aus dem Datenarchiv des Instituts für Statistik (`http://www.statistik.lmu.de/service/datenarchiv/nba/nba.html`) in R ein als `data.frame` mit Namen *nba*.
2. Wandeln Sie die Variable *Datum* in den Datentyp `date` um
3. Berechnen Sie *hdiff* als neue Variable in *nba* die Differenz der erzielten Punkte der Auswärtsmannschaft von der Heimmannschaft
4. Berechnen Sie die binäre Variable *siegh* in *nba*, die angibt, ob die Heimmannschaft gewonnen hat (1: Sieg, 0: Niederlage)