Statistische Software (R)

Paul Fink, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Datensatz-Aufbereitung

Datenbeispiel

chickwts: Datensatz über das Gewicht 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`

Geht auf 3 verschiedene Arten:

- über Name der Variable mit Listenzugriff
  `data$weight`
- über Name der Variable mit Matrixzugriff
  `data[,"weight"]`
- über Index der Variable mit Matrixzugriff
  `data[,1]`

Vorteil von Matrixzugriff: Man kann auch mehrere Variablen gleichzeitig herausholen

Zugriff auf bestimmte Beobachtungen

Funktioniert wieder mit Matrixzugriff!

Wenn man Index/Indizes der Beobachtung/en schon kennt kann man ihn direkt verwenden:

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

Was macht man, wenn man den Index nicht kennt, aber Beobachtungen anhand von Kriterien finden möchte?

$\Rightarrow$ Teildatensatz extrahieren (subset)

Über Matrixzugriff

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

oder `subset` Funktion

Die `subset` Funktion

Befehlssyntax:

```
subset(data, Kriterium an Variablen)
```

Beispiel:

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

```
subset(data, feed %in% c("casein","linseed"))
```

Das Kriterium ist ein Logischer Ausdruck!

```
subset(data, (feed == "casein") & (weight > 240))
```

Ist deutlich weniger Schreibarbeit

Logische Operatoren & Verknüpfungen

Die folgende Tabelle zeigt die Operationen und Funktionen für logische Vergleiche und Verknüpfungen.

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

Die Operatoren & und | arbeiten vektorwertig.

Neue Variablen hinzufügen

Anlegen neuer Variablen: (ges enthält so viele Elemente wie data Zeilen hat)

- Über Listenzugriff
  `data$gender <- ges`
- Über Matrixzugriff
  `data[,3] <- ges`
- Über cbind Funktion
  `data <- cbind(data,ges)`
- Über data.frame Funktion
  `data <- data.frame(data,ges)`

Variablen aus `data.frame` löschen

- Über Listenzugriff
  `data$gender <- NULL`
- Über Matrixzugriff nur Index/Indizes
  `data <- data[,-3]`

Ändern von Variablen in einem `data.frame`

Man kann eine Datensatz-Variable sehr einfach ändern, indem man sich erst die Variable aus dem Datensatz extrahiert und in ein neues (Hilfs-)Objekt speichert und dann dieses so verändert wie man es haben möchte und abschließend die (alte) Variable im Datensatz durch das Hilfsobjekt ersetzt.

Beispiel:
Die Variable Gewicht soll durch 1000 dividiert werden (entspricht Umwandlung von *g* nach *kg*)

```
h <- data[,"weight"]
h <- h / 1000
data[,"weight"] <- h
```

einfacher:

```
data <- transform(data, weight = weight/1000)
```


Weitere nützliche Funktionen

order: Ordnen
`daten[,order(V1,V2, usw. )]`
`daten` wird nach *V1* geordnet, wenn Werte gleich, dann nach *V2*, usw.

aggregate: Beobachtungen aggregieren
`aggregate(zuaggVar ~ nachFaktorVar, FUN = Funktion, data=daten)`
Zum Besipiel für das Gruppenmittel die Funktion `mean`.

with: 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)