Dataframes in R

Achim Zeileis, Regina Tüchler

October 3, 2006

Datensätze werden in R typischerweise als Objekte der Klasse `data.frame` dargestellt. Zeilen entsprechen den Beobachtungen und Spalten den verschiedenen Variablen. Zur Illustration verwenden wir den Datensatz `statlab` von der LV-Seite.

```
R> load("statlab.rda")
```

Wir stellen fest, dass es sich um ein Object der Klasse `data.frame` handelt:

```
R> class(statlab)

[1] "data.frame"
```

Die Anzahl an Beobachtungen beträgt 1296 und die Anzahl an Variablen 34:

```
R> dim(statlab)

[1] 1296   34
```

Die Variablennamen erhalten wir mit

```
R> names(statlab)

 [1] "CODE"   "CBSEX"  "CBB"    "CBLGTH" "CBWGT"  "CBMO"   "CBD"    "CBHR"
 [9] "CTHGHT" "CTWGT"  "CTL"    "CTPEA"  "CTRA"   "MBB"    "MBAG"   "MBWGT"
[17] "MBO"    "MBSM"   "MTHGHT" "MTWGT"  "MTE"    "MTO"    "MTSM"   "FBAG"
[25] "FBO"    "FBSM"   "FTHGHT" "FTWGT"  "FTE"    "FTO"    "FTSM"   "FIB"
[33] "FIT"    "FC"
```

Erste Aufschlüsse über die Variablentypen bekommen wir mit

```
R> head(statlab)

  CODE CBSEX CBB CBLGTH CBWGT CBMO CBD CBHR CTHGHT CTWGT CTL CTPEA CTRA MBB
1 1111     0   2   20.0   6.6    3   4    4   55.7    85   5    85   34   2
2 1112     0   2   20.0   6.4    5   7   13   48.9    59   3    74   34   6
3 1113     0   7   19.8   6.1    6   2    4   54.9    70   2    64   25   7
4 1114     0   6   19.5   7.0   10   2   19   53.6    88   3    87   43   6
5 1115     0   5   19.5   7.9    8   7    2   53.4    68   1    87   40   5
6 1116     0   5   22.0   9.5   11   6   12   59.9    93   4    83   37   5
  MBAG MBWGT MBO MBSM MTHGHT MTWGT MTE MTO MTSM FBAG FBO FBSM FTHGHT FTWGT FTE
1   17   119   0    0   66.0   130   1   1   20   19   8   10   70.1   171   3
2   17   130   0   20   62.8   159   3   1   10   23   6   11   65.0   130   1
3   18   134   0   10   66.1   138   2   0    0   21   0   20   70.0   175   2
4   18   135   1    6   61.8   123   2   0   -1   26   5   20   71.8   196   3
5   18   130   0   -1   62.8   146   2   8   -1   21   6   -1   68.0   163   2
6   18   104   0   -1   63.4   116   2   1   -1   17   6   20   74.0   180   3
```

```
  FTO FTSM FIB FIT FC
1   8   10  33 150  6
2   6   20  40 175  6
3   6    0  44 116  1
4   2   -1  42 112  4
5   6   -1  50 129  4
6   0   22   0 214  6
```

Die genauere Struktur sieht man mit:

```
R> str(statlab)
```

Wir unterscheiden kategoriale (oder qualitative) und metrische (oder quantitative) Variablentypen. Kategoriale Merkmale werden in R als Objekte der Klasse `factor` dargestellt, metrische als `numeric` oder `integer` (bei ganzzahligen Merkmalen).

Wir können auf einzelne Variablen zugreifen und z.B. den Typ abfragen:

```
R> class(statlab$CBB)

[1] "factor"
```

Um nicht jedesmal mit dem `$` Operator auf die Variablen eines Datensatzes zugreifen zu müssen, kann man sie auch mit dem Befehl `attach` direkt verfügbar machen.

```
R> attach(statlab)

R> class(CTWGT)

[1] "integer"
```

Mit

```
R> detach(statlab)
```

wird der obige Befehl wieder aufgehoben.

Besonders bei großen Datensätzen kann es sinnvoll sein, nur jene Beobachtungen (=Zeilen) und jene Variablen (=Spalten) auszuwählen, die man tatsächlich für die Analyse benötigt.

Auswahl von Beobachtungen (Zeilen)

Zur Auswahl von Beobachtungen eines Datensatzes verwendet man in aller Regel logische Ausdrücke wie etwa "alle Kinder, die bei ihrer Geburt mindestens 7 pound schwer waren". Dies wird mit dem R-Befehl `subset(data.frame, logical)` gemacht. Das erste Argument ist immer ein `data.frame` und das zweite ein Vektor mit logischen Ausdrücken, die entweder `TRUE` oder `FALSE` sind. Als Vergleichsoperatoren stehen in R u.a. Gleichheit (`==`), Ungleichheit (`!=`), größer (`>` und `>=`) und kleiner (`<` und `<=`) zur Verfügung. Außerdem können diese mit Hilfe der logischen Operatoren 'und' (`&`) und 'oder' (`|`) verknüpft werden.

Einige einfache Illustrationen sind:

```
R> x <- c(3, 5, 2, 9, 1)
R> x > 4

[1] FALSE  TRUE FALSE  TRUE FALSE

R> x > 6 | x < 2

[1] FALSE FALSE FALSE  TRUE  TRUE
```

```
R> x >= 3 & x <= 5

[1]  TRUE  TRUE FALSE FALSE FALSE
```

Wenn wir aus dem Datensatz `statlab` beispielsweise alle Beobachtungen auswählen wollen, die zu weiblichen Babies gehören, die bei der Geburt mindestens 70 pound gewogen haben, so können wir das entweder in zwei Schritten tun

```
R> set1 <- subset(statlab, CBSEX == 0)
R> set2 <- subset(set1, CBWGT >= 7)
```

oder völlig äquivalent auch in einem Schritt

```
R> set2 <- subset(statlab, CBSEX == 0 & CBWGT >= 7)
```

Damit hat sich die Anzahl der Zeilen auf 381 reduziert:

```
R> dim(set2)

[1] 381  34

R> nrow(set2)

[1] 381
```

Zusätzlich kann man so wie bei Matrizen die Beobachtungen auch durch Angabe der Zeilennummern auswählen. Z.B. wählt folgender Befehl nur die ersten sechs Beobachtungen aus:

```
R> set3 <- statlab[1:6, ]
```

Auswahl von Variablen (Spalten)

Die Auswahl kann entweder wieder über die Indices vorgenommen werden, z.B.

```
R> set4 <- statlab[, c(3, 5)]
```

oder man verwendet die Variablennamen, z.B.

```
R> set4 <- statlab[, c("CBB", "CBWGT")]
```

Fehlende Werte

Nicht immer sind alle Merkmale bei allen Merkmalsträgern erhoben worden, etwa wegen eines Messfehlers, weil die Daten verloren gegangen sind oder weil das Merkmal unbeobachtbar war. Mit dem Fehlen von Werten muss man sich bei der Analyse von Daten ebenfalls auseinandersetzen: dabei ist die defensivste Strategie, die Beobachtungen wegzulassen, bei denen nicht alle Merkmale vorhanden sind.

In R werden fehlende Werte durch `NA` (für not available) repräsentiert und es gibt verschiedene Funktionen, die speziell für deren Handhabung bereitgestellt werden. Besonders wichtig ist die Funktion `na.omit`, die in dem übergebenen Datensatz alle Zeilen weglässt, in denen es mindestens einen fehlenden Wert gibt.

Als Beispiel benutzen wir den `GSA` Datensatz, der dann im VK 4 noch öfters verwendet werden wird. Anhand dieses Datensatzes werden noch einmal alle drei Schritte bei der Selektion von Teildatensätzen durchgegangen. Hier sollen nur die Touristen betrachtet werden, die das Burgenland besucht haben. Von diesen wollen wir den Zusammenhang von Einkommen und Ausgaben untersuchen; fehlende Werte sollen weggelassen werden.

```
R> load("GSA.rda")
```

```
R> dim(GSA)

[1] 14571    52

R> gsa <- subset(GSA, province == "Burgenland")
R> dim(gsa)

[1] 1077   52

R> gsa <- gsa[, c("income", "expenditure")]
R> dim(gsa)

[1] 1077    2

R> gsa <- na.omit(gsa)
R> dim(gsa)

[1] 810   2
```

Hierbei ist es besonders wichtig, dass die fehlenden Werte erst ganz am Schluss weggelassen werden, da sonst auch fehlende Werte in möglicherweise irrelevanten Variablen dazu führen, dass einige Beobachtungen weggelassen werden, obwohl `income` und `expenditure` verfügbar wären.

Allerdings kann durch das Weglassen von Beobachtungen unter Umständen wertvolle Information verloren gehen, weshalb in vielen Anwendungen Anstrengungen unternommen werden, fehlende Werte zu imputieren, also durch geeignete Werte zu ersetzen. Hierfür ist häufig Hintergrundwissen hilfreich. In bestimmten Situation kann auch das Fehlen einer Beobachtung selbst von Interesse sein.