Univariate explorative Datenanalyse in R

Achim Zeileis

2009-02-20

1 Grundlegende Befehle

Zunächst laden wir den Datensatz (siehe auch *Daten.pdf*) `BBBClub`

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

das den `"data.frame"` `BBBClub` direkt verfügbar macht. Um sich dieses Objekt anzuschauen können einige kleine Befehle hilfreich sein. Die Befehle

```
R> class(BBBClub)
```

```
[1] "data.frame"
```

```
R> dim(BBBClub)
```

```
[1] 1300   11
```

sagen uns, daß das Objekt `BBBClub` von der Klasse `"data.frame"` ist und 1300 Zeilen (= Beobachtungen) und 11 Spalten (= Variablen) hat. Die Variablennamen erhalten wir per

```
R> names(BBBClub)
```

```
 [1] "choice" "gender" "amount" "freq"   "last"   "first"  "child"  "youth"
 [9] "cook"   "diy"    "art"
```

Um die ersten Zeilen eines Datensatzes anzuschauen, gibt es

```
R> head(BBBClub)
```

```
  choice gender amount freq last first child youth cook diy art
1    yes   male    113    8    1     8     0     1    0   0   0
2    yes   male    418    6   11    66     0     2    3   2   3
3    yes   male    336   18    6    32     2     0    1   1   2
4    yes   male    180   16    5    42     2     0    0   1   1
5    yes female    320    2    3    18     0     0    0   1   2
6    yes   male    268    4    1     4     0     0    0   0   0
```

Daraus können wir einige Eigenschaften der Variablen ablesen: die ersten zwei Variablen sind kategorial (oder qualitativ), die uebrigen Variablen sind metrisch (oder quantitativ). Kategoriale Merkmale werden in R als Objekte der Klasse `"factor"` dargestellt, metrische als `"numeric"` oder `"integer"` (bei ganzzahligen Merkmalen).

```
R> class(BBBClub$gender)

[1] "factor"

R> class(BBBClub$amount)

[1] "integer"
```

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(BBBClub)
R> class(gender)

[1] "factor"
```

2 Ein metrisches Merkmal

Um ein metrisches Merkmal numerisch zu beschreiben, gibt es verschiedene statistische Kennzahlen. Hier sollen ein paar der wichtigsten kurz anhand der quantitativen Variablen `amount` illustriert werden: Mittelwert, Varianz, Standardabweichung (die Wurzel aus der Varianz), Minimum und Maximum.

```
R> mean(amount)

[1] 201.3692

R> var(amount)

[1] 8954.526

R> sd(amount)

[1] 94.62836

R> min(amount)

[1] 15

R> max(amount)

[1] 474
```

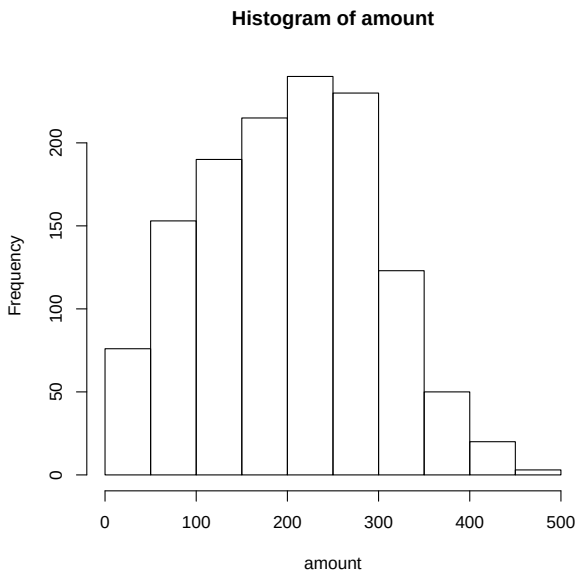
Anmerkung: Falls eine Variable `x` fehlende Werte hat, die in R durch `NA` (für 'not available') kodiert werden, dann ist das Ergebnis obiger Kennzahlen auch `NA`. Um diese `NA`s zu ignorieren, d.h. vor der Berechnung einfach wegzulassen, haben diese Funktionen ein `na.rm` Argument (dies steht für '`NA` remove'). Man kann also bspw. sagen `mean(x, na.rm = TRUE)` um den Mittelwert ohne die fehlenden Beobachtungen zu berechnen.

Die generische Funktion `summary` liefert, wenn man sie auf eine quantitative Variable anwendet, eine Fünf-Punkt-Zusammenfassung plus den Mittelwert, d.h. Minimum, unteres Quartil, Median, Mittelwert, oberes Quartil und Maximum.

```
R> summary(amount)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   15.0   127.0   204.0   201.4   273.0   474.0
```

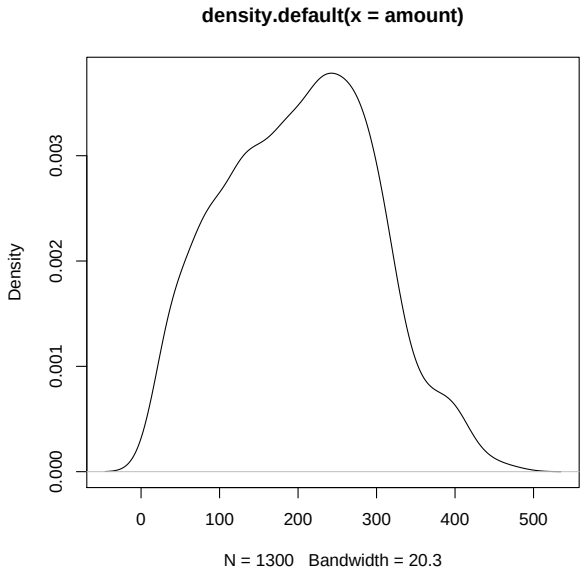
Um dasselbe metrische Merkmal auch zu visualisieren, werden wir nun Histogramme, geglättete Histogramme und Boxplots verwenden. Ein Histogramm für die Variable `amount` wird so erzeugt:

```
R> hist(amount)
```



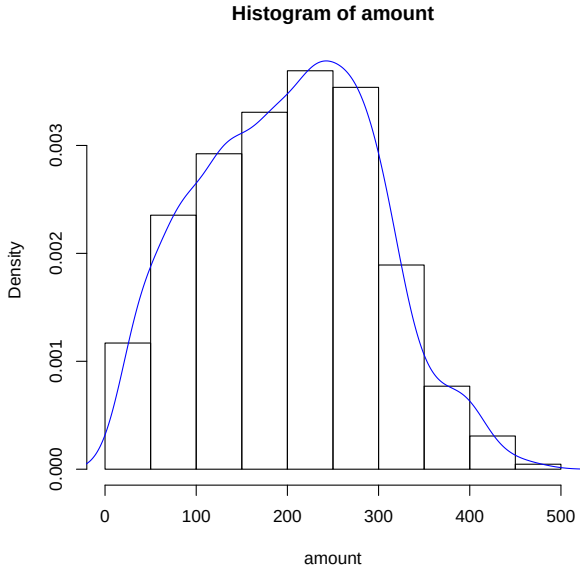
Dabei ist zu beachten, dass auf der y-Achse ‚frequencies', also absolute ‚Häufigkeiten', abgetragen werden. Damit auf der y-Achse die ‚Dichte' abgetragen wird (und damit die Fläche unter dem Histogramm 1 ist) muß man in R auch noch das Argument `freq` auf `FALSE` setzen (s.u.). Einen geglätteten Dichteschätzer erhält man durch `density`, die man durch die generische Funktion `plot` visualisieren kann:

```
R> plot(density(amount))
```



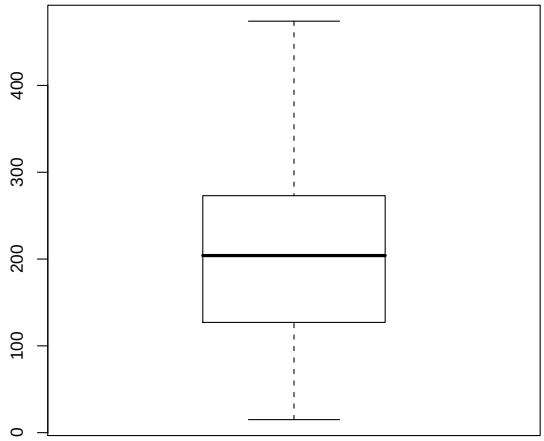
Man kann auch beide Dichteschätzer gemeinsam visualisieren: dabei wird die Dichte nicht durch `plot` in eine neue Grafik gezeichnet, sondern durch `lines` in die bestehende Grafik hinzugefügt:

```
R> hist(amount, freq = FALSE)
R> lines(density(amount), col = 4)
```



Zuletzt visualisieren wir die Variable auch noch mit Hilfe eines Boxplots:

```
R> boxplot(amount)
```

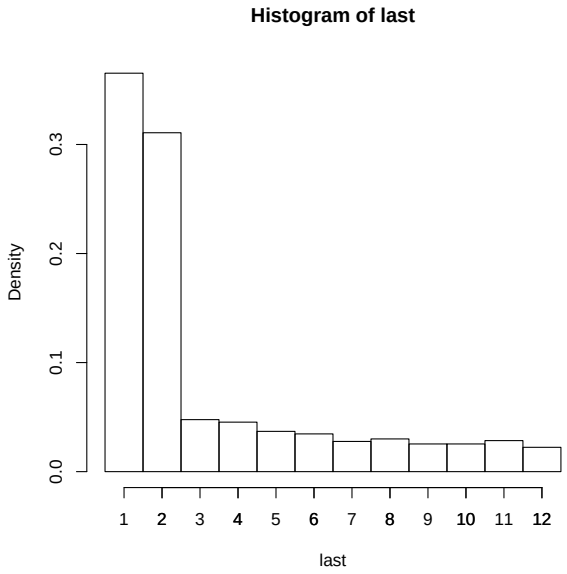


Falls die zu untersuchende metrische Variable diskret gemessen wurde, also beispielsweise nur ganzzahlige Werte enthält, kann man die explorative Analyse noch verfeinern. Dies soll hier beispielhaft für die Variable `last` durchgeführt werden. (**Anmerkung**: Die Variable `amount` ist zwar auch diskret erhoben worden, nimmt aber so viele verschiedene Werte an, daß dies kaum ins Gewicht fällt.) Zunächst kann man sich für diskrete Merkmale, die nur wenige Ausprägungen annehmen, eine Häufigkeitstabelle anschauen

```
R> table(last)

last
  1   2   3   4   5   6   7   8   9  10  11  12
475 404  62  59  48  45  36  39  33  33  37  29
```

Im Histogramm möchte man üblicherweise diese Kategorien widerspiegeln, aber `hist(last)` wählt diese nicht automatisch. Man kann die Intervalleinteilung aber über das Argument `breaks` steuern. Hier setzen wir es auf eine Sequenz $0.5, 1.5, \ldots, 12.5$, so daß die Ausprägungen $1, \ldots, 12$ immer genau in der Intervallmitte liegen. Sequenzen werden in R durch `seq(from, to, by)` erzeugt.

```
R> hist(last, breaks = seq(0.5, 12.5, by = 1), freq = FALSE)
R> axis(1, at = seq(1, 12, by = 1))
```



Der Aufruf von `axis` ist nicht zwingend notwending und dient hier nur einer ‚Verschönerung' der Grafik. Er generiert nochmal eine x-Achse (entspricht Nummer 1) mit Beschriftungen an (`at`) $1, \ldots, 12$.

3 Ein kategoriales Merkmal

Nun soll auch die qualitative oder kategoriale Variable `gender` numerisch und graphisch zusammengefaßt werden. Zur numerischen Beschreibung stehen Häufigkeitstabellen zur Verfügung: in R werden diese sowohl von der generischen Funktion `summary` erzeugt, wenn sie auf einen `"factor"` angewendet wird, als auch von der Funktion `table`.

```
R> summary(gender)

female   male
   456    844

R> table(gender)

gender
female   male
   456    844
```

Man kann diese Häufigkeitstabelle auch in einem Objekt abspeichern und damit weiterrechnen, beispielsweise um relative Häufigkeiten auszurechnen, entweder indem man `prop.table` anwendet oder ‚von Hand' durch die Anzahl der Beobachtungen dividiert.

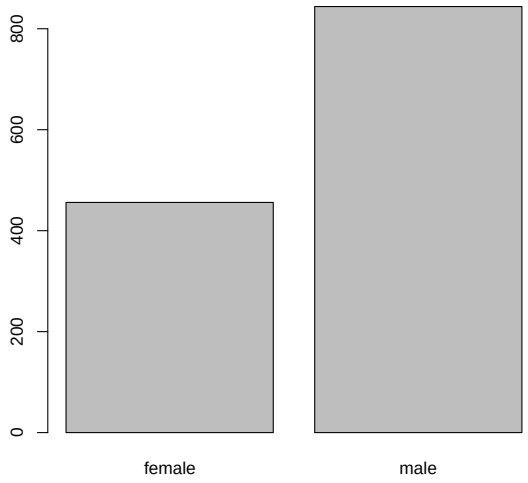
```
R> tab <- table(gender)
R> prop.table(tab)

gender
   female      male
0.3507692 0.6492308
```

```
R> tab/sum(tab)
```

```
gender
   female      male
0.3507692 0.6492308
```

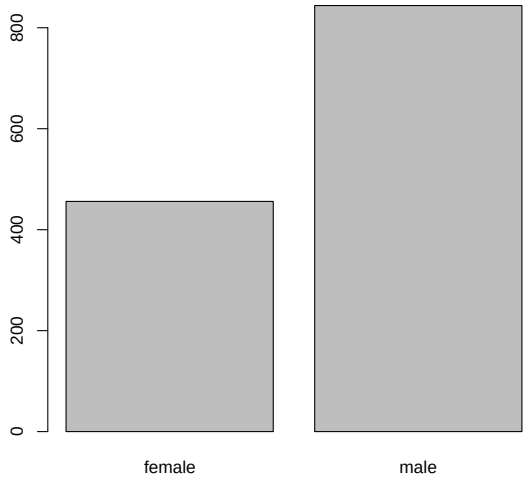
Am besten visualisiert man diese Daten durch ein Balkendiagramm. Dies wird entweder von der generischen Funktion `plot` erzeugt, wenn man sie auf einen `"factor"` anwendet

```
R> plot(gender)
```



oder völlig äquivalent von der Funktion `barplot`, wenn man sie auf die zugehörige Häufigkeitstabelle anwendet.

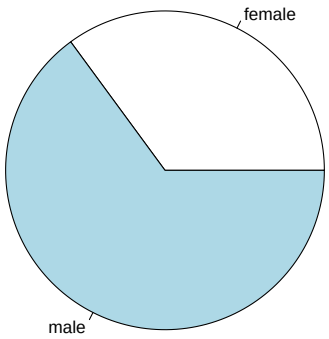
```
R> barplot(tab)
```



Hier könnte natürlich genausogut `barplot(tab/sum(tab))` benutzt werden, um die relativen Häufigkeiten zu visualisieren.

Eine weitere, wenn auch weniger flexible, Visualisierungsmethode ist das Tortendiagramm, das von `pie` angewendet auf eine Häufigkeitstabelle angezeigt wird:

```
R> pie(tab)
```



Bemerkung: Das Tortendiagramm ist allerdings nur gut geeignet, um Mehrheiten zu visualisieren. In fast allen anderen Situationen sind Balkendiagramme besser geeignet.

4 Abschließende Bemerkungen

Nach Abschluß der Analysen soll der Arbeitsplatz aufgeräumt werden. Als erstes wird dafür der Datensatz, der attached wurde, auch wieder detached

```
R> detach(BBBClub)
```

und die Objekte, die man nicht mehr benötigt

```
R> objects()
```

```
[1] "BBBClub" "tab"
```

sollten entfernt werden.

```
R> remove(BBBClub, tab)
```