Univariate explorative Datenanalyse in R

Achim Zeileis, Regina Tüchler

2006-10-03

1 Ein metrisches Merkmal

Wir laden den Datensatz:

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

und machen die Variablen direkt verfügbar:

```
R> attach(statlab)
```

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

```
R> mean(CTWGT)

[1] 70.93519

R> var(CTWGT)

[1] 194.6614

R> sd(CTWGT)

[1] 13.95211

R> min(CTWGT)

[1] 45

R> max(CTWGT)

[1] 143
```

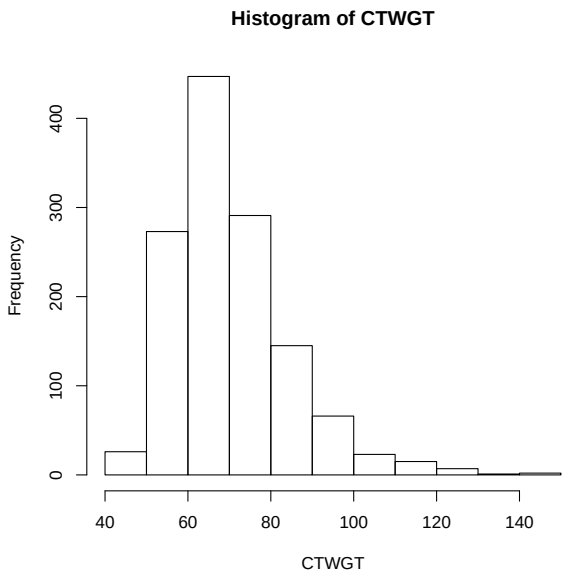
Anmerkung: Falls eine Variable `x` fehlende Werte hat, die in R durch `NA` (für 'not available') kodiert werden, dann ist das Ergebnis obiger Kennzahlen auch `NA`. Um diese `NA`s zu ignorieren, d.h. vor der Berechnung einfach wegzulassen, haben diese Funktionen ein `na.rm` Argument (dies steht für '`NA` 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(CTWGT)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  45.00   61.00   68.00   70.94   78.00  143.00
```

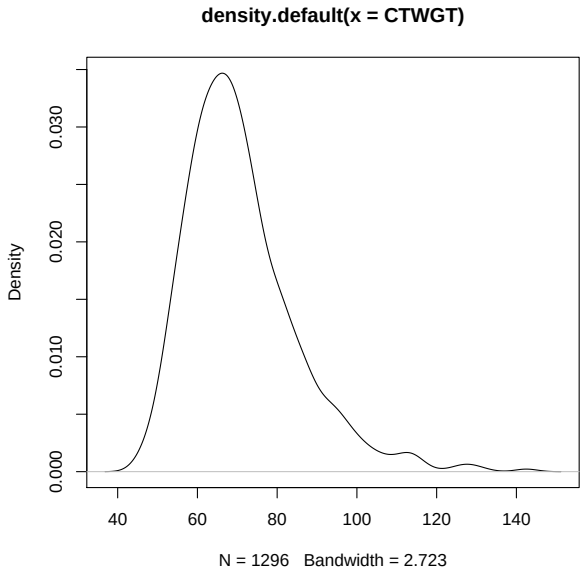
Um dasselbe metrische Merkmal auch zu visualisieren, werden wir nun Histogramme, geglättete Histogramme und Boxplots verwenden. Ein Histogramm für die Variable `CTWGT` wird so erzeugt:

```
R> hist(CTWGT)
```



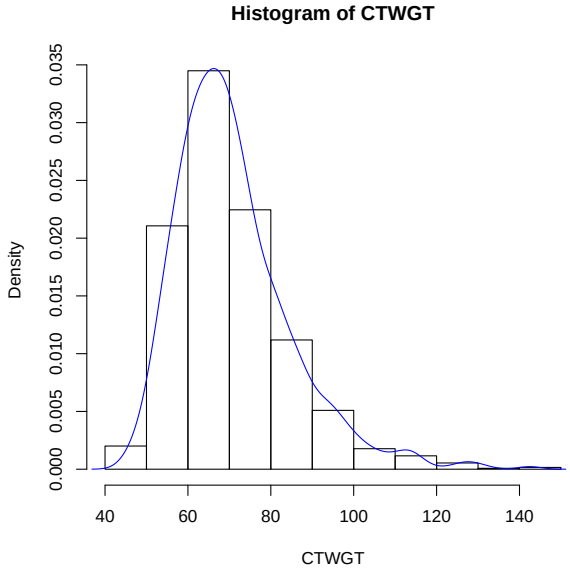
Dabei ist zu beachten, dass auf der y-Achse ‚frequencies', also absolute ‚Häufigkeiten', abgetragen werden. Damit auf der y-Achse die ‚Dichte' abgetragen wird (und damit die Fläche unter dem Histogramm 1 ist, siehe Statistik 1) muss man in R auch noch das Argument `freq` auf `FALSE` setzen (s.u.). Einen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

```
R> plot(density(CTWGT))
```



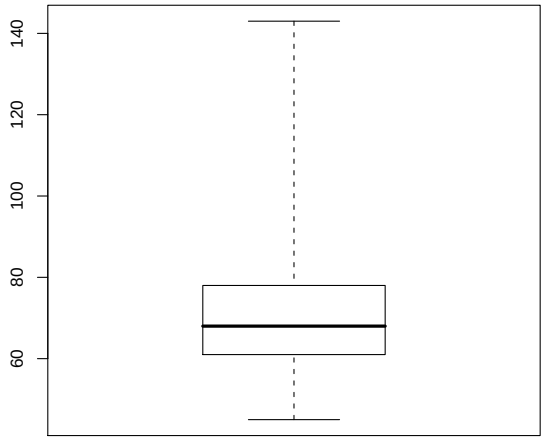
Man kann auch beide Dichteschätzer gemeinsam visualisieren: dabei wird die Dichte nicht durch `plot` in eine neue Grafik gezeichnet, sondern durch `lines` in die bestehende Grafik hinzugefügt:

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



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

```
R> boxplot(CTWGT, range = 0)
```



Falls die zu untersuchende metrische Variable diskret gemessen wurde und nicht allzu viele verschiedene Ausprägungen annimmt, also beispielsweise nur wenige ganzzahlige Werte enthält, kann man die explorative Analyse noch verfeinern. Die Variable `CTWGT` ist zwar diskret erhoben worden, nimmt aber so viele verschiedene Werte an. Wir verwenden daher nur einen Teil der Beobachtungen der Variable `CTWGT`, um die folgenden Befehle zu zeigen:

```
R> subCTWGT <- CTWGT[CTWGT < 54]
```

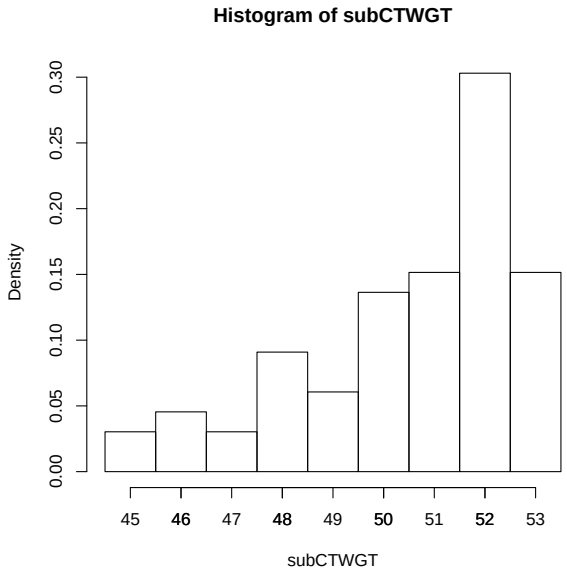
Zunächst kann man sich eine Häufigkeitstabelle anschauen.

```
R> table(subCTWGT)
```

```
subCTWGT
45 46 47 48 49 50 51 52 53
 2  3  2  6  4  9 10 20 10
```

Im Histogramm möchte man üblicherweise diese Kategorien widerspiegeln, aber `hist(subCTWGT)` wählt diese nicht automatisch. Man kann die Intervalleinteilung aber über das Argument `breaks` steuern. Hier setzen wir es auf eine Sequenz $44.5, 45.5, \ldots, 53.5$, sodass die Ausprägungen $45, \ldots, 53$ immer genau in der Intervallmitte liegen.

```
R> hist(subCTWGT, breaks = seq(44.5, 53.5, by = 1), freq = FALSE)
R> axis(1, at = 45:53)
```



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`) $45, \ldots, 53$.

2 Ein kategoriales Merkmal

Nun soll auch die qualitative oder kategoriale Variable `CBD` 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(CBD)

  1   2   3   4   5   6   7
196 167 178 186 187 190 192

R> table(CBD)

CBD
  1   2   3   4   5   6   7
196 167 178 186 187 190 192
```

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.

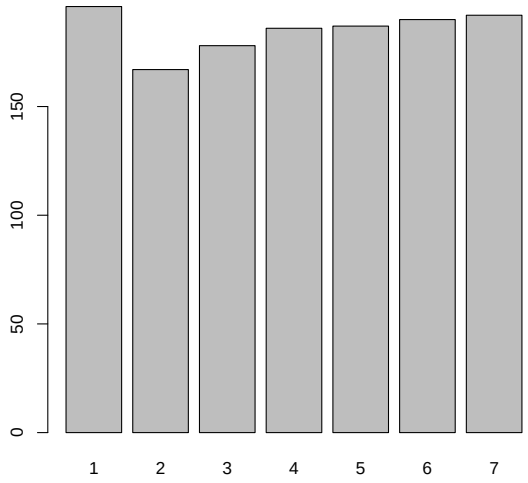
```
R> tab <- table(CBD)
R> prop.table(tab)

CBD
        1         2         3         4         5         6         7
0.1512346 0.1288580 0.1373457 0.1435185 0.1442901 0.1466049 0.1481481
```

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

```
CBD
        1         2         3         4         5         6         7
0.1512346 0.1288580 0.1373457 0.1435185 0.1442901 0.1466049 0.1481481
```

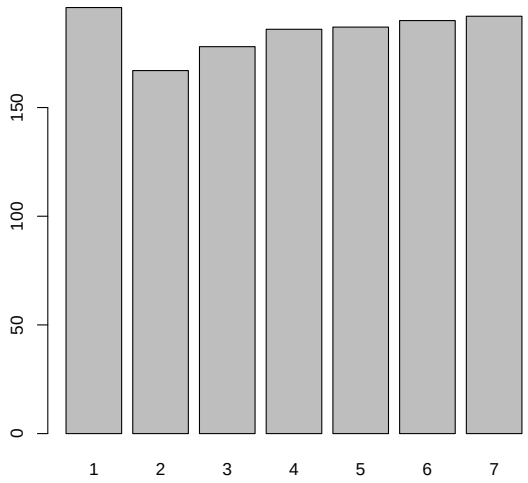
Am besten visualisiert man diese Daten durch ein Balkendiagramm. Dies wird entweder von der generischen Funktion `plot` erzeugt, wenn man sie auf einen `"factor"` anwendet

```
R> plot(CBD)
```



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

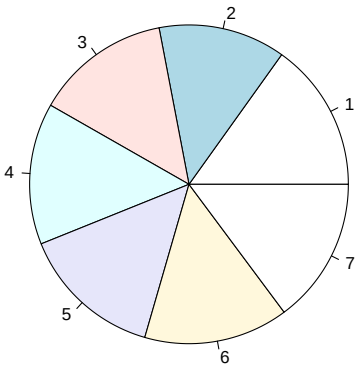
```
R> barplot(tab)
```



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.

3 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(statlab)
```

und die Objekte, die man nicht mehr benötigt

```
R> objects()
```

sollten entfernt werden.

```
R> remove(statlab, tab, subCTWGT)
```