

Bivariate explorative Datenanalyse in R

Achim Zeileis, Regina Tüchler

2006-10-09

In der LV Statistik 1 haben wir auch den Zusammenhang von 2 Variablen untersucht. Hier werden die dazugehörigen R-Befehle beschrieben:

1 Zwei metrische Merkmale

Zunächst einmal wird wieder der `statlab` Datensatz geladen und der Bequemlichkeit halber auch `attached`.

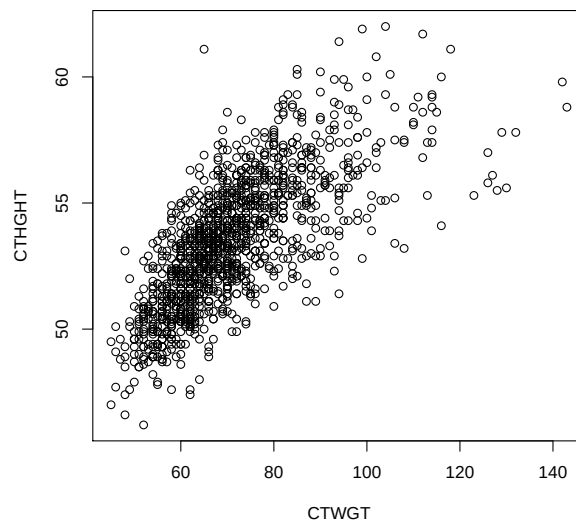
```
R> load("statlab.rda")  
R> attach(statlab)
```

Zur Messung des Zusammenhangs von zwei metrischen Variablen, kann die Korrelation verwendet werden. Der Korrelationskoeffizient zwischen Körpergröße `CTHGHT` und Gewicht `CTWGT` des Kindes wird berechnet durch

```
R> cor(CTHGHT, CTWGT)  
[1] 0.7234727
```

Aber viel anschaulicher wird der Zusammenhang der beiden Variablen in einem Streudiagramm dargestellt.

```
R> plot(CTHGHT ~ CTWGT)
```

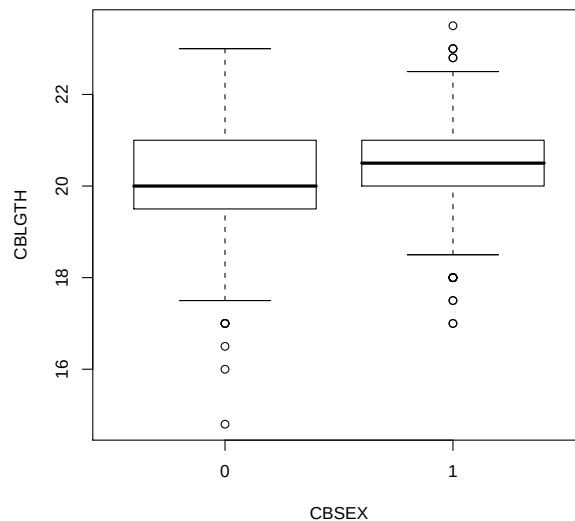


Hier wird die Formelnotation verwendet, in der `CTHGHT` erklärt wird (`~`) durch `CTWGT`. Äquivalent könnte auch `plot(CTHGHT, CTWGT)` verwendet werden. Sowohl Streudiagramm als auch Korrelationskoeffizient zeigen an, dass die beiden Merkmale positiv gekoppelt sind.

2 Ein abhängiges metrisches und ein erklärendes kategoriales Merkmal

Ist die erklärende Variable kategorial, wählt die Funktion `plot` automatisch eine passende Grafik aus, nämlich einen parallelen Boxplot. Die Syntax ist aber identisch zum obigen Beispiel. Wir untersuchen hier die Größe des Babies `CBLGTH` in Abhängigkeit vom Geschlecht `CBSEX` (Codierung 0 für Mädchen und 1 für Buben).

```
R> plot(CBLGTH ~ CBSEX)
```



Aus diesem Boxplot lässt sich ablesen, dass Buben bei der Geburt größer sind als Mädchen. Um numerische Zusammenfassungen der beiden Teilstichproben zu bekommen, wird in R der Befehl `tapply` verwendet. Als erstes Argument wird dabei die metrische Variable, als zweites die gruppierende kategoriale Variable und als drittes die zu evaluierende Funktion übergeben. Um die Mittelwerte in beiden Gruppen zu berechnen:

```
R> tapply(CBLGTH, CBSEX, mean)
```

```
      0      1
20.16590 20.61914
```

liefert einen Vektor mit beiden Mittelwerten zurück. Liefert die zu evaluierende Funktion nicht einen Skalar (also nur einen Wert, wie `mean`) sondern selbst schon einen Vektor, dann returniert `tapply` eine Liste:

```
R> tapply(CBLGTH, CBSEX, summary)
```

\$■0■

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
14.80	19.50	20.00	20.17	21.00	23.00

\$■1■

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
17.00	20.00	20.50	20.62	21.00	23.50

3 Zwei kategoriale Merkmale

Die einfachste numerische Beschreibung für zwei (oder mehr) kategoriale Merkmale, ist ihre Kontingenztafel, in der immer noch die gesamte Information der Originaldaten enthalten ist. In R gibt es zwei Befehle für die Erzeugung von Kontingenztafeln aus Faktoren (d.h. Objekten der Klasse "factor", die qualitative Variablen kodieren), und zwar `table` und `xtabs`. Das Resultat ist äquivalent, aber `xtabs` hat ein Formelinterface, das manchmal bequemer zu benutzen sein kann. Um also die Kontingenztafel von Geschlecht (Codierung `CBSEX`) und Blutgruppe (`CBB` in der Kontingenztafel) zu berechnen, können wir völlig äquivalent sagen:

```
R> table(CBSEX, CBB)
```

	CBB								
CBSEX	1	2	3	4	5	6	7	8	9
0	27	38	14	3	259	165	74	25	43
1	34	26	8	3	222	211	71	31	42

```
R> xtabs(~CBSEX + CBB)
```

	CBB								
CBSEX	1	2	3	4	5	6	7	8	9
0	27	38	14	3	259	165	74	25	43
1	34	26	8	3	222	211	71	31	42

Um mit der Kontingenztafel weiter rechnen zu können, speichern wir sie im Objekt `tab`.

```
R> tab <- table(CBSEX, CBB)
```

Diese enthält nun die absoluten Häufigkeiten, die in den entsprechenden Kombinationen von Kategorien beobachtet wurden. So haben beispielsweise 27 Mädchen Blutgruppe 0 neg (in der Tabelle Zeile 0 und Spalte 1).

Bedingte relative Häufigkeiten, also z.B. den Anteil der Blutgruppe 0 neg bei den Mädchen, können aus der bedingten Häufigkeitstabelle abgelesen werden. Diese bedingte Häufigkeitstabelle kann wieder mit dem Befehl `prop.table` berechnet werden. Als Argumente gibt man die Tabelle der absoluten Häufigkeiten an und die Nummer der Variablen auf die bedingt werden soll. Hier ist das die erste Variable `CBSEX`, daher:

```
R> prop.table(tab, 1)
```

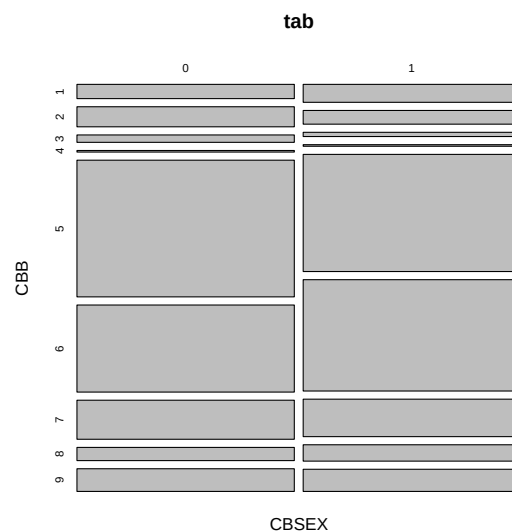
	CBB					
CBSEX	1	2	3	4	5	6
0	0.04166667	0.05864198	0.02160494	0.00462963	0.39969136	0.25462963
1	0.05246914	0.04012346	0.01234568	0.00462963	0.34259259	0.32561728

	CBB		
CBSEX	7	8	9
0	0.11419753	0.03858025	0.06635802
1	0.10956790	0.04783951	0.06481481

Die Zeilensummen in dieser Tabelle sind konsequenterweise genau 1, entsprechen also genau 100%. Aus der Tabelle können wir dann leicht ablesen, daß die empirische Wahrscheinlichkeit für Blutgruppe 0 neg. bei den Frauen bei ca. 4% liegt.

Als Visualisierung für die Kontingenztafel ist der sogenannte Mosaikplot sehr gut geeignet, der all die Information sowohl der Kontingenztafel mit den absoluten Häufigkeiten als auch mit den bedingten relativen Häufigkeiten darstellt. Der Mosaikplot wird in R mit dem Befehl `mosaicplot` erzeugt.

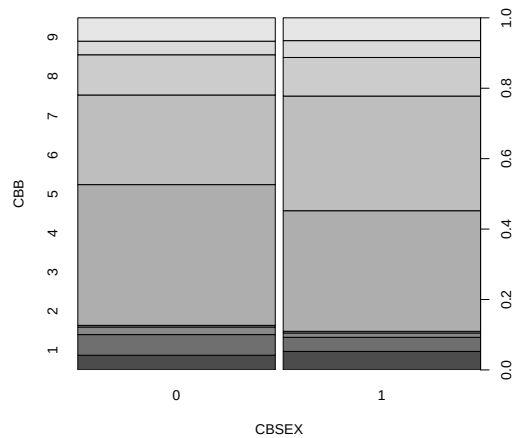
```
R> mosaicplot(tab)
```



Dieser ist in erster Linie eine flächenproportionale Darstellung der Kontingenztafel, d.h. salopp gesprochen: je größer der Eintrag in der Kontingenztafel, desto größer ist die Fläche des entsprechenden Mosaiks im Vergleich zu den übrigen Mosaiks.

Der Mosaikplot lässt sich noch etwas verschönern, wenn man die Mosaikflächen bezüglich der bedingten Variable einfärbt und direkt aneinanderlegt. Eine solche Darstellung nennt man auch Spineplot, sie kann einfach über `spineplot(tab)` generiert werden. Diese Funktion hat außerdem ein angenehmes Formelinterface, sodass man die Kontingenztafel nicht vorher von Hand berechnen muss, sondern direkt `spineplot(CBB ~ CBSEX)` sagen kann. Dasselbe macht auch die `plot()` Funktion selbst, wenn sie eine Formel mit zwei kategorialen Variablen übergeben bekommt:

```
R> plot(CBB ~ CBSEX)
```



Auf der linken Seite steht, wie bei den Befehlen in den vorangegangenen Abschnitten, die abhängige Variable und rechts die erklärende Variable. Die Funktion liefert außerdem die Kontingenztafel, die sie visualisiert hat unsichtbar zurück:

```
R> tab <- plot(CBB ~ CBSEX)
R> tab
```

	CBB								
CBSEX	1	2	3	4	5	6	7	8	9
0	27	38	14	3	259	165	74	25	43
1	34	26	8	3	222	211	71	31	42

4 Ein abhängiges kategoriales und ein erklärendes metrisches Merkmal

Die einfachste Möglichkeit den Zusammenhang eines abhängigen kategorialen Merkmals mit einem erklärenden metrischem Merkmal numerisch und graphisch zu beschreiben, ist die metrische Variable zu diskretisieren und dann fortzufahren als hätte man zwei kategoriale Variablen. Diese Idee ist sehr ähnlich zu der Idee des Histogramms, wo die metrische Variable auch in verschiedene Intervalle (und damit also Kategorien) eingeteilt wird, für die dann wieder absolute und relative Häufigkeiten berechnet werden können. Wir betrachten hier die Körpergröße des Babies `CBLGHT` in Abhängigkeit vom Geschlecht `CBSEX`.

Um eine quantitative Variable in verschiedene Kategorien gleichsam zu “zerschneiden” steht in R der Befehl `cut` zur Verfügung. Dieser erwartet als erstes Argument die metrische Variable und als zweites Argument die Stelle, an denen er schneiden soll. Eine einfache und in der Regel recht gute Wahl ist es, die Variable in vier gleich große Stücke zu zerschneiden. Man teilt dann also einfach mit den Werten der Fünf-Punkt-Zusammenfassung.

```
R> fivenum(CBLGTH)
```

```
[1] 14.8 20.0 20.5 21.0 23.5
```

die uns das Minimum, das untere Quartil, den Median, das obere Quartil und das Maximum liefert. Zwischen jeweils zwei benachbarten Werten liegen also immer 25% der Daten. Die diskretisierte Variable erhalten wir dann mit

```
R> CBLGTH2 <- cut(CBLGTH, fivenum(CBLGTH), include.lowest = TRUE)
R> summary(CBLGTH2)
```

```
[14.8,20] (20,20.5] (20.5,21] (21,23.5]
      577       221       270       228
```

Das Argument `include.lowest` muss man hier auf `TRUE` setzen, damit der kleinste Wert 14.8 am linken Rand mit in das Intervall aufgenommen wird. Und mit der resultierenden kategorialen Variablen `CBLGTH2` verfahren wir jetzt genau wie im vorangegangenen Abschnitt, d.h. wir können Kontingenztafeln absoluter und relativer Häufigkeiten berechnen.

```
R> tab <- table(CBLGTH2, CBSEX)
R> tab
```

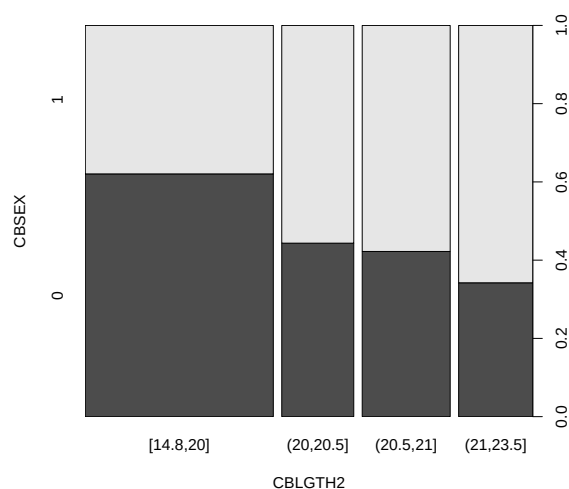
	CBSEX	
CBLGTH2	0	1
[14.8,20]	358	219
(20,20.5]	98	123
(20.5,21]	114	156
(21,23.5]	78	150

```
R> prop.table(tab, 1)
```

	CBSEX	
CBLGTH2	0	1
[14.8,20]	0.6204506	0.3795494
(20,20.5]	0.4434389	0.5565611
(20.5,21]	0.4222222	0.5777778
(21,23.5]	0.3421053	0.6578947

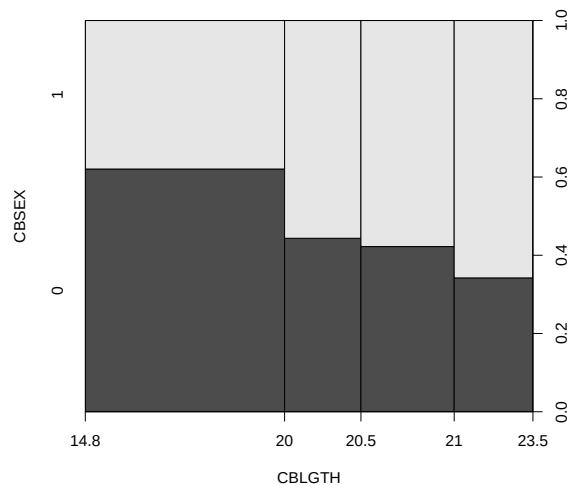
Der Zusammenhang kann auch wieder mit einem Mosaikplot oder Spineplot visualisiert werden:

```
R> spineplot(tab)
```



Dabei wird allerdings die x-Achse unterbrochen, eine etwas schönere Darstellung bekommen wir, wenn wir `spinplot()` mit dem Formelinterface aufrufen und dabei die gewünschten Bruchpunkte mit dem Argument `breaks` übergeben:

```
R> spineplot(CBSEX ~ CBLGTH, breaks = fivenum(CBLGTH))
```



Setzt man das Argument `breaks` nicht, so werden die Bruchpunkte automatisch so gewählt wie bei der Funktion `hist()`. 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 sollten entfernt werden.

```
R> remove(statlab, tab, CBLGTH2)
```