

Einige nützliche Abbildung für die statistische Analyse von Sprechdaten in R

Jonathan Harrington

R-libraries

```
library(emu)
library(MASS)
library(lattice)
```

1. Tabellarische Daten (count data)

Es handelt sich hier um Aufzählungen von **kategorialen Daten**.

Kategorial: die Anzahl von /I, E, a/.

wie oft wird /r/ im Vergleich zu /R/ in Bayern verwendet? (Kategoriale Daten, weil die Wahl nur zwischen /r/ oder /R/ besteht).

Kontinuierlich: Die F2-Werte von /I/ im Vergleich zu /E/.

Die Zungenposition und -konfiguration in der Erzeugung von /r/-Lauten.

Nützliche Funktionen in der deskriptiven Statistik von kategorialen Daten

```
table(), prop.table(), barplot()
```

```
table(vowlax.l)
```

Als Proportionen

```
prop.table(table(vowlax.l))
```

`table()` kann auch verwendet werden, um Tabellen von miteinander verbundenen Kategorien zu erstellen. zB. `vowlax.l` enthält Etikettierungen vor dem Vokal. Wir wollen die Anzahl der Vokale nach Links-Kontext tabellieren.

```
table(vowlax.left, vowlax.l)
```

Es können auch mehrdimensionale Tabellen erstellt werden - wenn wir z.B. die links x Vokal x rechts Kontexte zählen wollen:

```
table(vowlax.left, vowlax.l, vowlax.right)
```

Ein **Barplot** ist eine graphische Darstellungen der Informationn, die man auch mit `table()` bekommen kann. (Manche verwenden Barplots für kontinuierliche Daten, aber dies ist im allgemeinen nicht zu empfehlen - Boxplots, siehe unten, sind dafür viel besser geeignet).

Hier wird ein Barplot erstellt, für die Anzahl der /b, d/ Plosive im linken Kontext x Vokal:

```
temp = vowlax.left %in% c("b", "d")
left = vowlax.left[temp]
vow = vowlax.l[temp]
o = table(left, vow)
```

```
barplot(o)
```

Besser ist nebeneinander

```
barplot(o, beside=T)
```

Die Abbildung schöner gestalten (siehe auch `help(barplot)` und `example(barplot)`).

```
col = c("green", "red")
barplot(o, beside=T, col=col)
legend(locator(1), c("b", "d"), fill=col, cex=2)
```

Übrigens um die Farb-Möglichkeiten zu sehen, `farben = colors()` . Siehe auch

```
pie(rep(1,30), col=rainbow(30))
pie(rep(1, 6), col=1:6)
```

Vier Graustufen-Abbildungen zwischen schwarz und weiß: `gray(0:3/4)`

Daher:

```
g = gray(0:20/20)
barplot(rep(2, 21), col=g)
```

2. Kontinuierliche Daten: Ein Parameter

2.1 Allgemein

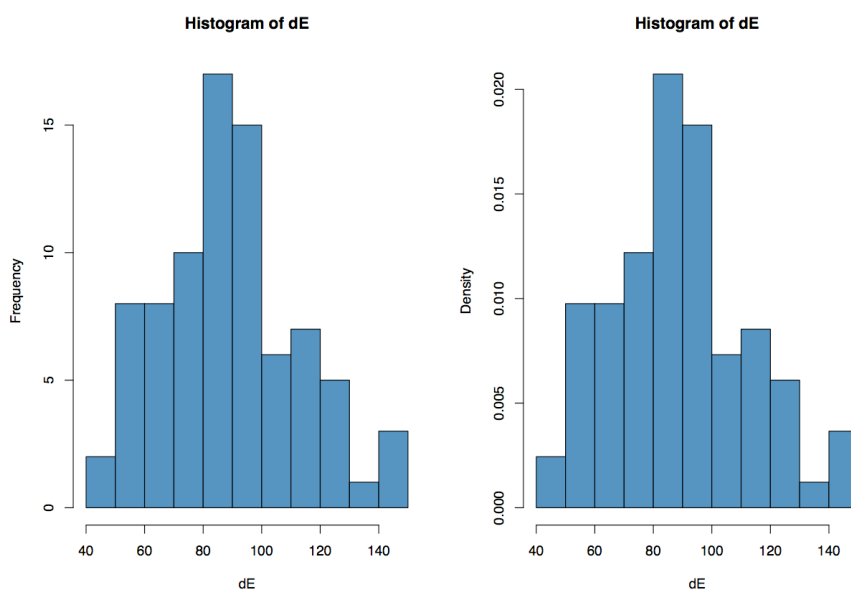
`mean()`, `quantile()`, `median()`, `IQR()`, (auch `sd()` aber das wird später im Zusammenhang mit der Normalverteilung diskutiert).

`boxplot()` und `hist()` sind zwei der nützlichsten Funktionen, um einen graphischen Überblick der Daten zu bekommen.

2.2 Histogramme

Histogramm der Dauer von "E" Vokalen im Dataset `vowlax`

```
d = dur(vowlax)
temp = vowlax.l == "E"
dE = d[temp]
par(mfrow=c(1,2))
col = "steelblue"
hist(dE, col=col)
hist(dE, freq=F, col=col)
```



Der Unterschied zwischen diesen Histogrammen: die Gesamt-Fläche vom Histogramm (rechts) = 1.

Genauer:

$$\text{Density} = \text{Frequency} / (n * \text{Balkenbreite}) \quad (1)$$

Die Balkenbreite ist 10, n ist 82 (`length(dE)`) und z_B ist die Häufigkeit des vierten Balken von links 10 (in der Abbildung links).

Daher ist die Höhe vom entsprechenden Balken in der Abbildung rechts:

```
10 / (82 * 10)
0.01219512
```

Die Fläche im Histogramm (rechts) ist eine Verschlüsselung der **Proportionen**.

$$\text{Fläche} = \text{Balkenbreite} * \text{Density} \quad (2)$$

Für den vierten Balken von links ist die Fläche daher:

```
10 * 0.01219512
0.1219512
```

Also ca. 12% der Werte liegen zwischen 70 und 80. Stimmt das?

```
sum(dE > 70 & dE < 80)
10
10/82
0.1219512
```

2.3 Boxplot-Abbildungen

2.3.1 Quantale (quantiles)

Quantal p ($0 < p < 1$) befindet sich in Position `1+p*(n-1)` in der nach Reihenfolge sortierten Daten.

21 Dauerwerte

```
werte = dur(vowlax)
werte = werte[1:21]
```

sortieren

```
werte.s = sort(werte)
```

Median:

```
median(werte)
```

das gleiche wie:

```
pos = 1+0.5*(21-1)
werte.s[pos]
```

min, 25% Quantal, Median, 75% Quantal, Maximum

```
quantile(werte)
```

der 37% und 68% Quantale

```
quantile(werte, c(0.37, 0.68))
```

IQR(): interquartile-range

```
quantile(werte, .75) - quantile(werte, .25)
```

das gleiche

```
IQR(werte)
```

Boxplot-Abbildung. Der dicke Strich in der Mitte = der Median, die oberen und unteren Kanten vom Rechteck sind die .25 und .75 Quantalen.

```
boxplot(dE)
abline(h=median(dE), col="blue")
abline(h=quantile(dE, c(.25, .75)), col="green")
```

Aus `help(boxplot)` :

the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.

Mehrere Boxplots nebeneinander (zB um Vokal-Kategorien miteinander zu vergleichen):

```
d = dur(vowlax)
#
boxplot(d ~ vowlax.1)
```

* bedeutet 'gekreuzt mit' oder 'Interaktion mit'.

```
boxplot(d ~ vowlax.1 * vowlax.spkr)
```

Eine andere äquivalente, aber etwas schönere Darstellung gibt es mit der `bwplot()` Funktion

```
bwplot(d ~ vowlax.1 | vowlax.spkr)
```

3. Kontinuierliche Daten: Abbildungen von 2 oder mehreren Parametern

```
form = vowlax.fdat.5[,1:4]
```

3.1 Zwei Parameter

```
plot(form[,1:2], pch=vowlax.1)
```

oder

```
plot(form[,1], form[,2], type="n")
text(form[,1], form[,2], vowlax.1)
```

nach Farben kodiert

```
code = as.numeric(factor(vowlax.1))
plot(form[,1], form[,2], pch=vowlax.1, col=code)
```

Farbauswahl

```
farben = c("brown", "grey", "blue", "orange")[code]
plot(form[,1], form[,2], pch=vowlax.1, col=farben)
```

Zeichenauswahl

```
plot(form[,1:2], pch=code)
plot(form[,1:2], pch=code, col=code)
```

Plotting characters (Argumente an `pch`)

```
help(points)
```

Die Funktion `pchShow()` in `help(points)` laden

dann

```
par(mfrow=c(1,2))
pchShow(c("o","O","0"), cex = 2.5)
pchShow({}, cex = 4, cextext=0, main=NULL)
```

zB mit 'plotting characters' 15-18:

```
p = c(15:18)[code]
plot(form[,1:2], pch=p)

# mit Farbe
plot(form[,1:2], pch=p, col=farben)

# selbe Symbolen, unterschiedliche Farben
plot(form[,1:2], pch=21, col=code)
```

3.2 Mehrere Parameter

```
pairs(form)
lab = paste("Formant", 1:4, sep="")
pairs(form, labels=lab)

pairs(form, labels=lab, col=code)
pairs(form, labels=lab, pch=15, col=code)

# usw.
```

4. Daten und data-frames

In der statistischen Analyse von Daten werden oft (aber keineswegs immer) so genannte 'data-frames' verwendet, in denen nicht nur numerische und Schriftzeichen-Daten, sondern auch abhängige und unabhängige Variablen zusammen gespeichert werden können.

Terminologie: wenn wir prüfen wollen, ob die Dauer sich nach Vokal-Kategorie signifikant unterscheidet.

Abhängige Variabel: Dauer

Unabhängige Variabel auch Faktor (in R mit `factor()` erstellt): Vokal-Kategorie

Eine unabhängige Variabel hat mindestens 2 Stufen (levels): zB "I", "E", "a" "O" = 4 Stufen
Es können auch mehrere unabhängige Variablen vorliegen (wenn wir z.B. prüfen wollen, ob sich die Dauer signifikant nach Vokal-Kategorie und Akzentuierung unterscheidet).

Erstellung einer data-frame

```
d = dur(vowlax)
d.df = data.frame(Dauer=d, Vokal=factor(vowlax.1),
Sprecher=factor(vowlax.spkr))
names(d.df)
```

Wichtig: ein data-frame lässt sich mehr oder weniger wie eine Matrix behandeln (vor dem Komma bezieht sich auf die Reihen, nach dem Komma auf die Spalten).

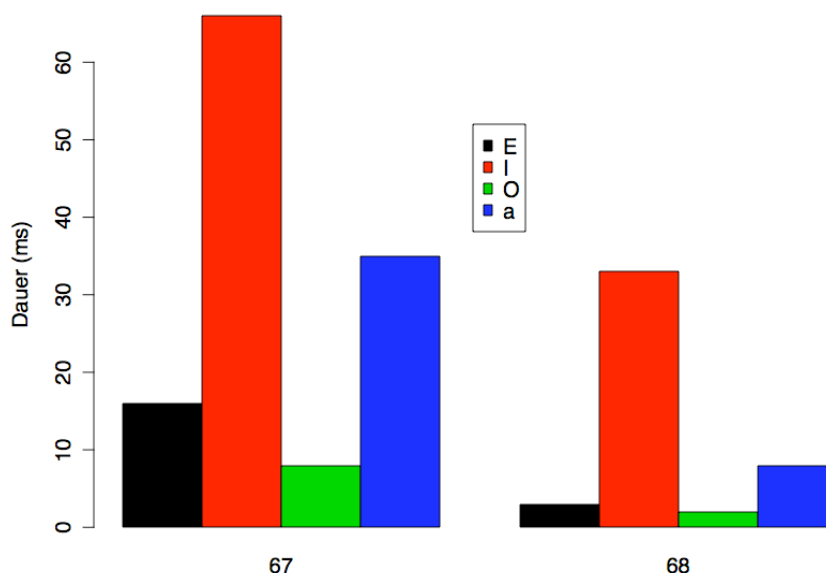
```
attach() ermöglicht Zugang zu den Variablen durch deren Namen zB
attach(d.df)
Dauer[1:5]
length(Vokal)
table(Sprecher)
```

5. Fragen

Diese Fragen beziehen sich auf `vowlax.df` : das sind Vokal-Daten von 2 Sprechern zum zeitlichen Mittelpunkt. Die Aufgaben gehen vielleicht schneller nach `attach(vowlax.df)` . Dann zum Schluss: `detach(vowlax.df)` .

Schreiben Sie R-Befehle für:

- 1 Die Anzahl der Reihen.
- 2 Die Dimensionen vom Data-frame (Reihen x Spalten).
- 3 Die ersten fünf Reihen.
- 4 Die Namen vom Data-frame.
- 5 Alle Reihen im Data-frame für die F1 - (minus) f0 in Bark größer als 3.5 Bark ist (Die Funktion `bark(x)` konvertiert von Hertz nach Bark, und `bark(x, inv=T)` von Bark nach Hertz)
- 6 All Reihen mit "I" Vokalen.
- 7 Die Anzahl von "I" und "E" Vokalen für die F2 weniger als 1800 Hz ist.
- 8 Alle Reihen von "I" für die F3 (Bark) - F2 (Bark) weniger also als F1 (Bark) ist.
- 9 Die Reihe mit der größten Dauer.
- 10 Die Reihen mit einer Dauer größer als 100 ms und mit f0 kleiner als 90 Hz ist.
- 11 Die Anzahl von "I" und "E" Vokalen mit F1 größer als 400 Hz und F2 weniger als 1600 Hz.
12. Wieviele "I" Vokale haben F1 höher als der Median F1 von "E" Vokalen?
13. Der Barplot zeigt die Anzahl der Vokale getrennt für die beiden Sprecher für alle Vokale mit einer Dauer weniger als 72 ms. Schreiben Sie die Befehle, um diese Abbildung zu erzeugen.



14. Erstellen Sie eine Abbildung wie unten, F1 x f0 x RMS von "a" Vokalen der Beiden Sprecher zu differenzieren. NB: eine Farbkodierung um die Sprecher zu differenzieren, gibt es mit

```
cols = c(1:2)[speaker]
```

