

Einige Abbildung für die statistische Analyse von Sprechdaten in R (Teil 1)

Jonathan Harrington

0. Einführung

Die grundlegende Funktion für Abbildungen ist `plot()`

```
x = 1:10
plot(x)
y = x^2
plot(x, y)
```

0.1 Einige Parameter für R-Abbildungen

`plot()` sowie die meisten Funktionen, um in R Abbildungen zu erzeugen, können mit mehreren Argumenten ergänzt werden. Hier sind die wichtigsten davon (siehe `help(par)` für alle):

die Farbe. Entweder Ganzzahl oder eine aus `colors()` auswählen
`col="blue"`

Mit Linie, ("l"), Linie + Punkte ("b"), Keine Werte ("n").
`plot(x, type="l")`

Solid, dotted, dashed...(default `lty=1`)
`plot(x, type="l", lty=1)`
 # das gleiche
`plot(x, type="l", lty="solid")`

Liniendichte
`plot(x, type="l", lty="solid", lwd=2)`

Symbole für die Abbildung
`plot(x, pch=2)`
`plot(x, type="b", pch=1)`

Weitere Symbole

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

```
pchShow <-
function(extras = c(" ", ".", "o", "O", "0", "+", "-", "|", "%", "#"),
        cex = 3, ## good for both .Device=="postscript" and "x11"
        col = "red3", bg = "gold", coltext = "brown", cextext = 1.2,
        main = paste("plot symbols : points (... pch = *, cex =",
                      cex, ")"))
{
  nex <- length(extras)
  np <- 26 + nex
  ipch <- 0:(np-1)
  k <- floor(sqrt(np))
  dd <- c(-1,1)/2
  rx <- dd + range(ix <- ipch %/% k)
  ry <- dd + range(iy <- 3 + (k-1) - ipch %/% k)
  pch <- as.list(ipch) # list with integers & strings
  if(nex > 0) pch[26+ 1:nex] <- as.list(extras)
  plot(rx, ry, type="n", axes = FALSE, xlab = "", ylab = "",
        main = main)
  abline(v = ix, h = iy, col = "lightgray", lty = "dotted")
}
```

```

for(i in 1:np) {
  pc <- pch[[i]]
  ## 'col' symbols with a 'bg'-colored interior (where available) :
  points(ix[i], iy[i], pch = pc, col = col, bg = bg, cex = cex)
  if(cextext > 0)
    text(ix[i] - 0.3, iy[i], pc, col = coltext, cex = cextext)
}
}

```

dann

```

pchShow(c("o","O","0"), cex = 2.5)
pchShow(NULL, cex = 4, cextext = 0, main = NULL)

```

```

# Beschriftung
plot(x, type="b", pch=1, main="Überschrift", xlab = "etwas", ylab="etwas")
# Keine Achsen-Beschriftung
plot(x, type="b", pch=1, main="Überschrift", xlab = "", ylab="")

```

```

# Bereich setzen
x = y = 0:10

```

```

ylim = xlim = c(-20, 20)
plot(x, y, type="b", pch=1, main="Überschrift", xlim = xlim, ylim=ylim)

```

```

# Ohne Achsen
plot(x, y, type="l", axes=F)

```

```

# x-Achse hinzufügen
axis(side=1)

```

```

# x-Achse oben (axis(side = 2) und axis(side=4) für die y-Achsen)
axis(side=3)

```

0.2 Überlagerungsfunktionen

```

x = y = 0:10

```

```

# Die Linie y = -x + 5
plot(x, y, type="l")
abline(5, -1, col="red")

```

```

# Linie von Koordinaten überlagern. Hier von [2, 8] nach [4, 1]
lines(c(2, 4), c(8, 1), type="l", col="green", lty="dotted")

```

```

# Ein oder mehrere Werte überlagern
points(5, 9, pch=5)

```

```

# Text überlagern; cex ist character expansion
text(6, 3, "Hallo", cex=2)

```

```

# locator(1) gibt die Koordinaten vom Mausklick zurück

```

```
# Daher text per Mausklick einfügen
text(locator(1), "mein text")
```

```
# Polynom überlagern. Hier  $y = 0.1x^2 + 3$  zwischen  $x = 4$  und  $x = 9$ 
curve(0.1 * x^2 + 3, xlim=c(4, 9), col="blue", add=T)
```

```
# curve() kann auch alleine ohne add=T eingesetzt werden
curve(cos(x) + sin(x), xlim = c(-20, 20))
```

```
#  $y = \cos(x) e^{-0.1x}$ 
curve(cos(x) * exp(-.1*x), xlim=c(0, 50), main = "A decaying sinusoid", ylab="Amplitude")
```

0.3 Abbildungen überlagern, oder nebeneinander...

```
# Eine Abbildung auf eine andere überlagern: par(new=T)
```

```
plot(0:10, 0:10, type="l")
par(new=T)
plot(0:20, 20:0, type="l", col=2)
par(new=F)
```

```
# besser: Bereiche setzen; und die x- und y-Achsen Beschriftungen in
# der ersten Abb. weglassen
xlim = c(0, 20)
ylim = c(0, 20)
plot(0:10, 0:10, type="l", xlim=xlim, ylim=ylim, xlab="", ylab="")
par(new=T)
plot(0:20, 20:0, type="l", xlim=xlim, ylim=ylim, col=2, xlab="x", ylab="y")
par(new=F)
```

```
# Mehrere Abbildungen nebeneinander
# mfrow(m, n) m = Reihen, n = Spalten. Hier 4 Abbildungen in 2 Reihen und 2 Spalten
# (Es gibt auch mfcrow(m, n) )
```

```
par(mfrow=c(2,2))
# oben links
plot(1:10)
# oben rechts
curve(cos(x) + sin(x), from = -20, to=20)
# unten links
curve(cos(x) * exp(-.1*x), from = 0, to= 50, main = "A decaying sinusoid",
ylab="Amplitude")
```

```
# Zurücksetzen auf 1 x 1
par(mfrow=c(1,1))
plot(1:10)
```

1. Tabellarische Daten (count data)

Es handelt sich hier um Anzahlen **kategorialer 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 Informationen, die man auch mit `table()` bekommt. (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 für die Anzahl der /b, d/ Plosive im linken Kontext x Vokal erstellt:

```
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:

```
# 21 Graustufen
g = gray(0:20/20)
# 21 Bars mit diesen Graustufen
barplot(rep(2, 21), col=g)
```

2. Kontinuierliche Daten: Ein Parameter

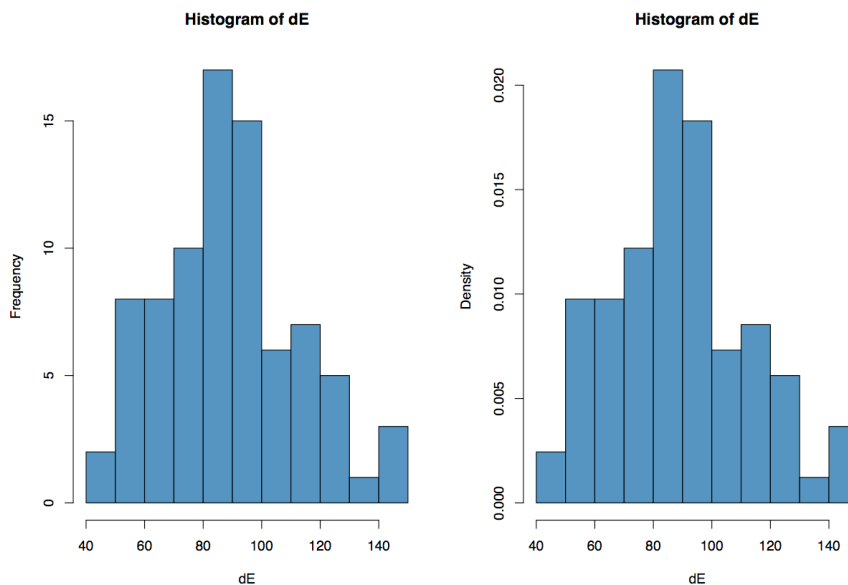
2.1 Allgemein

`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 vowelx
d = end(vowelx) - start(vowelx)
temp = vowelx.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:

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

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?

$$\text{sum}(dE > 70 \ \& \ dE < 80) \\ 10 \\ 10/82 \\ 0.1219512$$

Eine Gaußglocke auf die rechte Abbildung überlagern

$$\text{curve}(\text{dnorm}(x, \text{mean}(d[\text{temp}]), \text{sd}(d[\text{temp}])), \text{xlim}=\text{c}(40, 160), \text{add}=\text{T}, \text{lwd}=2)$$

2.3 Boxplot-Abbildungen

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

11 randomisierte Ganzzahlen zwischen 0 und 100

$$g = \text{sample}(0:100, 11)$$

Median

$$\text{median}(g)$$

Das gleiche

Der sechste Werte (da 11 Werte und wegen $1+p*(n-1)$) in der sortierten Reihenfolge

$$g.s = \text{sort}(g) \\ g.s[6]$$

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

$$\text{quantile}(g)$$

die 37% und 68% Quantale

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

```
# IQR() : interquartile-range
quantile(g, .75) - quantile(g, .25)
# das gleiche
IQR(g)
```

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

```
# Dauerwerte von "E", Sprecher "67"
temp = vowelx.l == "E" & vowelx.spkr == "67"
dE = dur(vowelx[temp,])
```

```
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(vowelx)
#
boxplot(d ~ vowelx.l)
```

```
# * bedeutet 'gekreuzt mit' oder 'Interaktion mit'.
boxplot(d ~ vowelx.l * vowelx.spkr)
```

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

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

```
bwplot(d ~ vowelx.l | vowelx.spkr)
```

Zusammenfassung der wichtigsten Funktionen und Parameter

Einfache Abbildungen und Parameter

```
plot(1:10) # Einfache Abbildung
plot(1:10, 1:10)
plot(1:10, type="l") # Mit Linie
plot(1:10, type="l", lty=2, pch = 3)

plot(1:10, xlab="etwas", ylab="y-Achse", main="Haupt") # Beschriftung
```

```
xlim = c(0, 20) # Bereiche setzen
ylim = c(-5, 30)
plot(1:10, xlim=xlim, ylim=ylim)
```

```
plot(1:10) # Text auftragen
text(locator(1), "hier etwas hinzufügen")
```

```
plot(1:10) # Abbildungen aufeinander überlagern
par(new=T)
plot(10:1, type="l")
```

```
par(mfrow=c(1,2)) # Abbildungen nebeneinander
plot(1:10)
plot(1:10)
par(mfrow=c(1,1))
```

Funktionen für kategorialen Daten

```
table(vowlax.l) # Tabellen
table(vowlax.l, vowlax.left)
prop.table(table(vowlax.l, vowlax.spkr)) # Als Proportionen

res = table(vowlax.l, vowlax.spkr) # Barplot
barplot(res, beside=T)
```

Funktionen für kontinuierliche Daten (Ein-Dimensional)

```
d = end(vowlax) - start(vowlax)
mean(d) # Mittelwert
median(d) # Median
IQR(d) # Interquartaler Bereich
sd(d) # Standardabweichung
hist(d) # Histogramm
boxplot(d) # Boxplot
boxplot(d ~ vowlax.l) # Boxplot als Funktion von Labels
boxplot(d ~ vowlax.l * vowlax.spkr) # Boxplot als Funktion gekreuzter Labels
```

3. Fragen

1. `vowlax.word` und `vowlax.l` sind zwei Schriftzeichen-Vektoren, die zueinander parallel sind d.h. sie sind von derselben Länge und `vowlax.word[n]` ist die Wort-Etikettierung die zu der Vokal-Etikettierung `vowlax.l[n]` passt. Erstellen Sie eine Tabelle wie unten, die die Verteilung der Vokale für die Wörter *Menschen*, *nicht*, *will*, *man*, *Mannschaft* zeigt.

	E	I	a
Mannschaft	0	0	4
Menschen	6	0	0
man	0	0	6
nicht	0	12	0
will	0	6	0

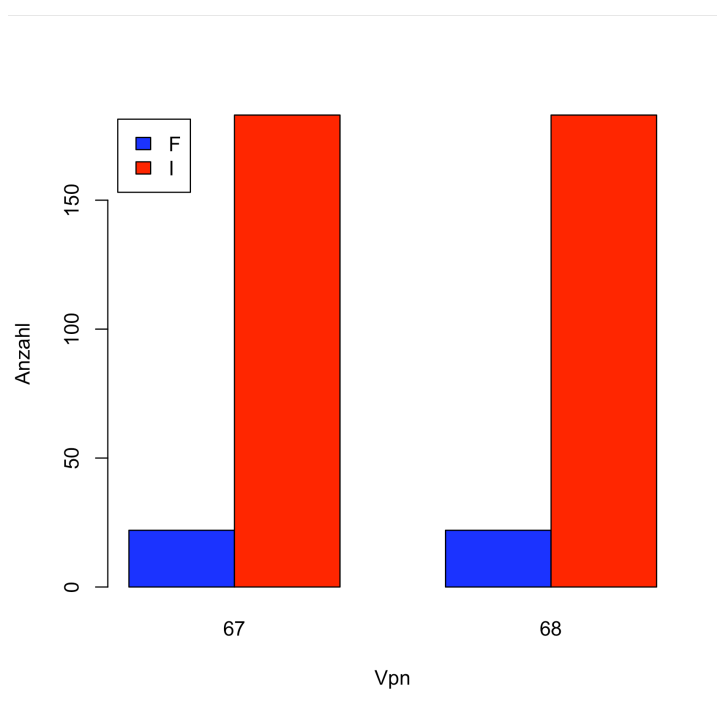
2.

(a) Erstellen Sie einen neuen Vektor `w`, aus `vowlax.word`

`w = vowlax.word`

Ersetzen Sie in `w` die Wörter *doch*, *ich*, *bin*, *ist*, *man*, *will*, *nicht* mit "F" (Funktionswort) und alle andere mit "I" (Inhaltswort). (NB `x[temp] = "etwas"` ersetzt alle Elemente von `x[temp]` mit "etwas").

(b) Verwenden Sie `w` und `vowlax.spkr`, um die Abbildung unten der Verteilung der F/I Wörter getrennt für die beiden Sprecher zu erstellen, wie unten



3. Die folgenden Objekte sind parallel zueinander:

`vowlax.fdat.5`
`vowlax`

Matrix F1-F4 (4 Spalten zum zeitlichen Mittelpunkt)
 Segmentliste

vowlax.spr
vowlax.l
vowlax.left

Sprecher-Etikettierung ("67" oder "68")
Vokal-Etikettierung
Etikettierung, linker Kontext

Schreiben Sie die R-Befehle um die Abbildung wie unten zu erzeugen, die auf diese Weise zusammengesetzt wurden:

- (a) F2/F1 für "I" und "O" Vokale für Sprecherin 68
- (b) F2 von "I" getrennt für die linken Kontexte "r" und "Q" (Glottalverschluss) Sprecherin 68
- (c) Logarithmus der Dauer getrennt für alle Vokale Sprecher 67
- (d) F1 von "a" und "E" für beide Sprecher

Schreiben Sie in 1-2 Zeilen pro Abbildung, inwiefern die Verteilungen Ihren Erwartungen entsprechen.

