

```

# Bibliotheken und Data-Frames laden
library(lattice)
pfadu = "http://www.phonetik.uni-muenchen.de/~jmh/lehre/Rdf"
asp = read.table(file.path(pfadu, "asp.txt"))
coronal = read.table(file.path(pfadu, "coronal.txt"))
int.df = read.table(file.path(pfadu, "intdauer.txt"))

#####
# 0. Einleitung
#####
# Zur Erinnerung, df ist ein Data-Frame
# Ein Data-Frame besteht aus
# Reihen (Beobachtungen)
# und
# Spalten (Variablen)
#
# class(df), head(df), dim(df), nrow(df), ncol(df), df[n,m]: Reihe n, Spalte m

# Prüfen, ob es ein Data-Frame ist:
class(asp)
# die ersten paar Beobachtungen:
head(coronal)

# Die Spalten können numerische und kategoriale Variablen enthalten

# Numerische Variablen: Ganzzahlen oder kontinuierlich
#
class(asp$d)
# oder
with(asp, class(d))
# [1] "numeric"

class(int.df$Dauer)
# [1] "integer"

# Kategoriale Variablen sind Faktoren mit Stufen:
class(coronal$Socialclass)
# [1] "factor"

# Die Stufen davon. z.B. die ersten paar anschauen...
coronal$Socialclass[1:10]
# oder
levels(coronal$Socialclass)

#####
# Drei typische Sorten von Untersuchungen in der Phonetik:
#####
# Inwiefern ist y von x abhängig? (Wird y von x beeinflusst?)
# y = abhängige Variable, x = unabhängige Variable

# 1. y = numerisch, x = kategorial
# 1.1 z.B. unterscheiden sich /i, e, a/ in der Dauer?
# 1.2 = Inwiefern ist die Dauer (y: abhängige numerische Variable) vom Vokal (x:

```

```

unabhängiger Faktor
# mit drei Stufen) abhängig?
# 1.3 Abbildungen dafür: bwplot(), densityplot(), histogram()

# 2. y = kategorial, x = kategorial
# 2.1 z.B. Wörter wie Sohn, Sonne... können mit /s/ oder /z/ produziert werden.
# Kommt /s/ eher in Bayern und /z/ eher in Hamburg vor?
# 2.2 = Inwiefern ist die Stimmhaftigkeit (y: abhängige kategoriale Variable mit 2
Stufen: s/z)
# vom Dialekt (x: unabhängige kategoriale Variable mit 2 Stufen: Bayern/Hamburg) abhän
ngig?
# 2.3 Abbildungen dafür: barchart(), dotplot()

# 3. y = numerisch, x = numerisch
# 3.1 z.B. gibt es eine größere Dauer bei einer größeren Mundöffnung?
# 3.2 Inwiefern ist die Dauer (y: abhängige numerische Variable)
# von der Mundöffnung (x: unabhängige numerische Variable) abhängig?
# 3.3 Abbildung dafür: xyplot()

# Zusammenfassung:
# 1. y = numerisch, x = kategorial: bwplot(), densityplot(), histogram()
# 2. y = kategorial, x = kategorial: barchart(), dotplot()
# 3. y = numerisch, x = numerisch: xyplot()

#####
# 1. bwplot(), histogram(), densityplot(). y = numerisch, x = kategorial
#####
# Inwiefern ist die Aspirationsdauer von der Artikulationsstelle abhängig?
# Boxplot: Der Punkt liegt beim Median; der 'Box' ist der interquartale Bereich
#
##### Median und Interquartale Bereich
# Median (oder 50% Quantil). Der mittlere Wert in der sortierten Reihenfolge
x = c(15, 19, -1, 10, 11, 18, 90000)
median(x)

# das gleiche
quantile(x, .5)

# Nach Sortierung sieht man, dass 15 an der mittleren Stelle vorkommt.
sort(x)
# -1    10    11    15    18    19 90000

# Es gibt auch andere Bruchteile von Quantilen. Insbesondere
# wird von dem interquartilen Bereich Gebrauch gemacht, um die Streueung einzuschä
tzen
IQR(x)

# das gleich
quantile(x, .75) - quantile(x, .25)
#####

bwplot(d ~ Kons, data = asp)
# Histogram: eine Aufteilung in Proportionen

```

```

histogram(~ d | Kons, data = asp)
# Densityplot: Wahrscheinlichkeitsdichte (die Fläche unter der Kurve = 1)
# siehe auch: http://www.math.utah.edu/~lzhang/teaching/3070summer2008/DailyUpdates/jun26/sec4\_1.pdf
densityplot(~ d | Kons, data = asp)
densityplot(~ d | Kons, data = asp, plot.points=F, ref = T)

# Densityplot: die Stufen vom Faktor aufeinander überlagern: groups = Faktor
densityplot(~ d, group = Kons, data = asp, plot.points=F, ref = T)
densityplot(~ d, group = Kons, data = asp, plot.points=F, ref = T, auto.key=T)

#####
# 2. barchart(), dotplot(): y ist kategorial, x ist kategorial
#####

# Inwiefern ist die Artikulationsstelle von der Region abhängig?

# i. Tabelle: die abhängige Variable (hier Fr) immer
# an letzter Stelle!
tab = with(coronal, table(Region, Fr))
# oder
tab = table(coronal$Region, coronal$Fr)
#
# ii.
# Proportionen. zweites Argument ist 1:n wo n die Anzahl
# der unabhängigen Faktoren ist. Hier 1:1 was auch 1 gleicht
prop = prop.table(tab, 1:1)
# oder
prop = prop.table(tab, 1)
# iii.
barchart(prop, horizontal=F, auto.key=T)
dotplot(prop, horizontal=F, auto.key=T)

#####
# 3. xyplot: y ist numerisch, x ist numerisch
#####
# Inwiefern ist die Dauer von der Intensität abhängig?
xyplot(Dauer ~ dB, data = int.df)

#####
# 4. Bild-Ergänzungen/Erweiterungen
#####

##### 1: Beschriftungen, Bereiche
# ylim, xlim, main, xlab, ylab
# Achsenbeschriftung
histogram(~ d | Kons, data = asp, main = "Histogramm", xlab = "Dauer (ms)")
bwplot(d ~ Kons, data = asp, main = "Meine Daten", ylab = "Dauer (ms)")

# Bereich setzen: ylim =, xlim =
bwplot(d ~ Kons, data = asp, main = "Bplot", ylab = "Dauer (ms)", ylim = c(0, 200))
densityplot(~ d, group = Kons, data = asp, plot.points=F, ref = T, auto.key=T,

```

```

xlim = c(-50, 200))

xyplot(Dauer ~ dB, data = int.df, ylim = c(50, 250), xlim = c(10, 50))

# auch so:
ylim = c(50, 250)
xlim = c(10, 50)
m = "Dauer und Intensität"
xyplot(Dauer ~ dB, data = int.df, ylim = ylim, xlim = xlim, main = m)

##### 2: Bilder gleichzeitig sehen
# Zuerst die Bilder in R ablegen
b = bwplot(d ~ Kons, data = asp)
h = histogram(~ d | Kons, data = asp)
d = densityplot(~ d, group = Kons, data = asp)

# Die Bilder einzeln ansehen
h
b
d

#
dim(h)
h[2]

# split = c(colnumber, rownumber, ncol, nrow), more =T)
# split = (in welcher Spalte?, in welcher Reihe?)
# wieviele Spalten hat das Gesamtbild? wieviele Reihen hat das Gesamtbild?
plot(b, split = c(1, 1, 3, 1), more=T)
plot(h, split = c(2, 1, 3, 1), more=T)
plot(d, split = c(3, 1, 3, 1))

##### 3: Farben für bwplot(), histogram(), densityplot()
farben = c("green", "red")
# um alle Farben zu sehen:
colors()

bwplot(d ~ Kons, data = asp,
par.settings = list(
box.dot = list(col = farben),
box.umbrella = list(col = farben),
box.rectangle = list(col = farben),
plot.symbol = list(col = "slategray")
))

bwplot(d ~ Kons, data = asp,
par.settings = list(
box.dot = list(col = "black"),
box.umbrella = list(col = farben),
box.rectangle = list(col = farben, fill = farben),
plot.symbol = list(col = "slategray")
)

```

```
))
```

```
densityplot(~ d, group = Kons, data = asp, auto.key=T,  
par.settings = list(  
superpose.line = list(col = farben),  
superpose.symbol = list(col = farben)  
))
```

```
barchart(prop, horizontal=F, auto.key=T,  
par.settings = list(  
superpose.polygon = list(col = farben)  
))
```

```
dotplot(prop, horizontal=F, auto.key=T,  
par.settings = list(  
superpose.symbol = list(col = farben, cex = 2)  
))
```

```
xyplot(Dauer ~ dB, data = int.df,  
par.settings = list(  
plot.symbol = list(col = "purple", pch=0, cex = 1.5)  
))
```

```
# um die verschiedenen Plot-Symbole zu sehen:
```

```
source(file.path(pfadu, "pchShow.R"))  
pchShow()
```

```
dotplot(prop, horizontal=F, auto.key=T,  
par.settings = list(  
superpose.symbol = list(col = farben, pch = c(10, 15), cex = 2)  
))
```