Wichtige Befehle in R zur Datenanalyse

Ergänzende Unterlagen zur Vorlesung

Prof. Dr. Oliver Gansser

Inhaltsverzeichnis

Datenmanagement

```
#Einlesen der Daten
#Daten müssen im gleichen Verzeichnis liegen wie das Skript
#Download der Daten mit dem Befehl
download.file("https://goo.gl/whKjnl", destfile = "tips.csv")
tips<-read.csv2("tips.csv")

#Alternativ mit Angabe des Pfades, wenn die Daten nicht im Workspace liegen
#Workspace suchen mit getwd()
#Worspace neu setzten mit setwd(Hier den Pfad angeben)

#Datenstruktur betrachten
str(tips)
```

```
## 'data.frame':    244 obs. of  7 variables:
##  $ total_bill: num  17 10.3 21 23.7 24.6 ...
##  $ tip       : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ sex       : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 2 2 2 2 2 ...
##  $ smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day       : Factor w/ 4 levels "Fri","Sat","Sun",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ time      : Factor w/ 2 levels "Dinner","Lunch": 1 1 1 1 1 1 1 1 1 1 ...
##  $ size      : int  2 3 3 2 4 4 2 4 2 2 ...
```

```
#Dimensionen des Datensatzes
dim(tips)
```

```
## [1] 244   7
```

```
#Kopf der Datenmatix betrachten
head(tips)
```

```
##   total_bill  tip    sex smoker day   time size
## 1      16.99 1.01 Female     No Sun Dinner    2
## 2      10.34 1.66   Male     No Sun Dinner    3
## 3      21.01 3.50   Male     No Sun Dinner    3
## 4      23.68 3.31   Male     No Sun Dinner    2
## 5      24.59 3.61 Female     No Sun Dinner    4
## 6      25.29 4.71   Male     No Sun Dinner    4
```

```
#Paket mosaic laden (oder auch andere Pakete, die Sie benötigen)
library(mosaic)
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.4.1
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Loading required package: lattice
## Loading required package: ggplot2
## Loading required package: mosaicData
## Loading required package: Matrix
##
## The 'mosaic' package masks several functions from core packages in order to add addit
## The original behavior of these functions should not be affected by this.
##
## Attaching package: 'mosaic'
## The following object is masked from 'package:Matrix':
##
##     mean
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cov, D, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
#oder require(mosaic) => gilt für alle Pakete

#konvertieren von factor auf nummerisch
tips$sex<-as.numeric(tips$sex)
```

```
#konvertieren von nummerisch auf factor
tips$sex<-as.factor(tips$sex)

#Ausgabe der levels
levels(tips$sex)
```

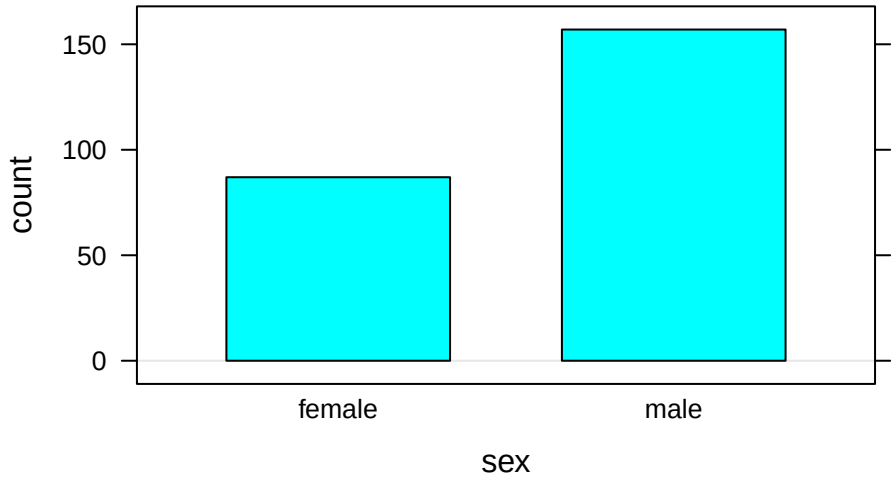
```
## [1] "1" "2"
```

```
#Wertzuweisung von levels
levels(tips$sex)<-c("female","male")
```

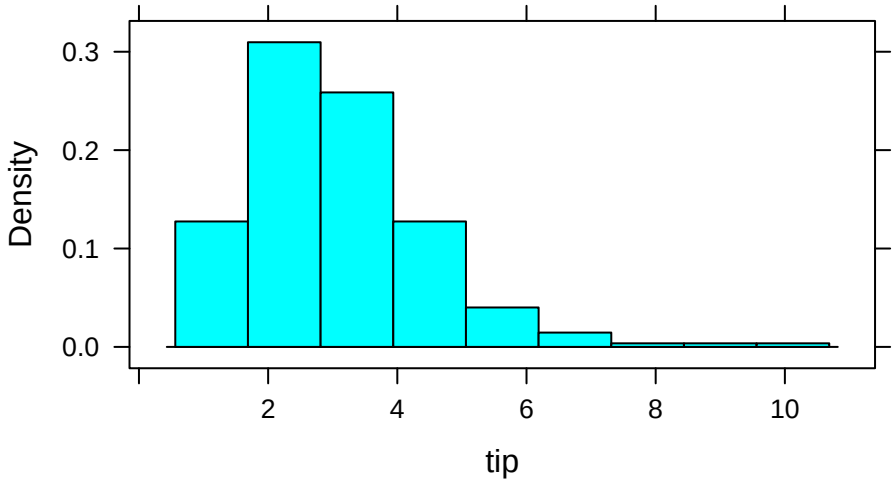
Grafische Verfahren

```
#Balkendiagramm bei kategorialen Daten
bargraph(~sex, data=tips)
```

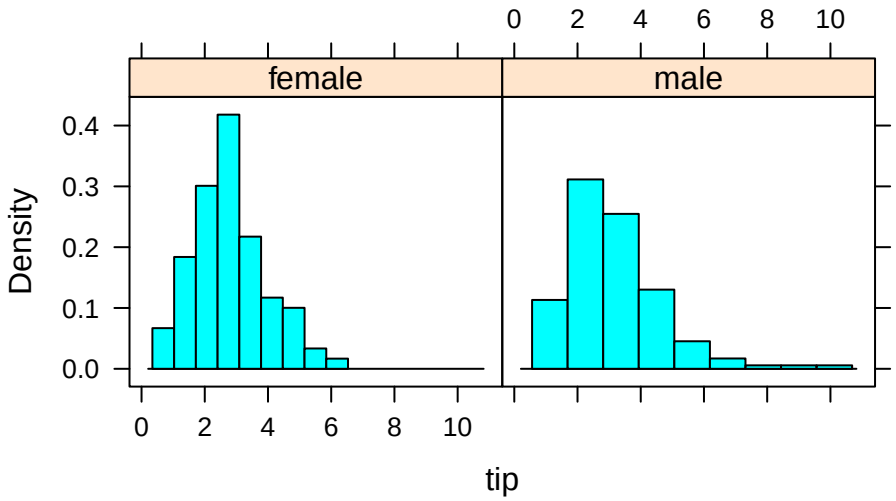
```
## Warning: package 'bindrcpp' was built under R version 3.4.1
```



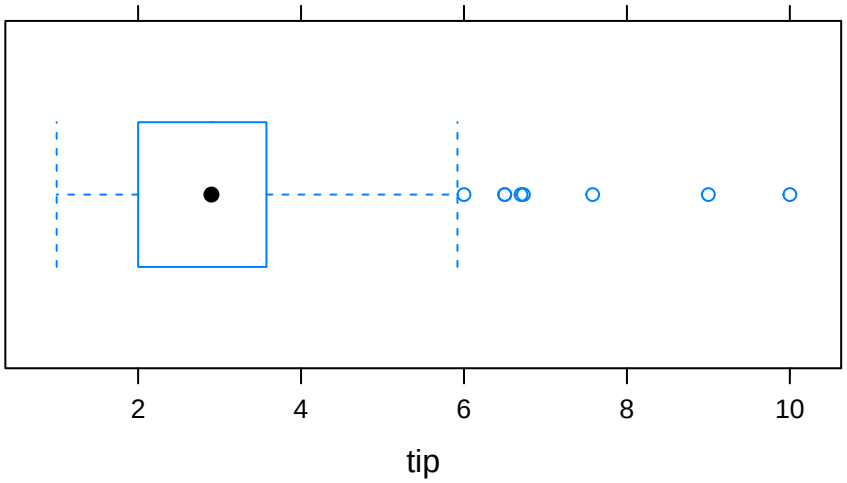

```
#Histogramm bei metrischen Daten
histogram(~tip, data=tips)
```



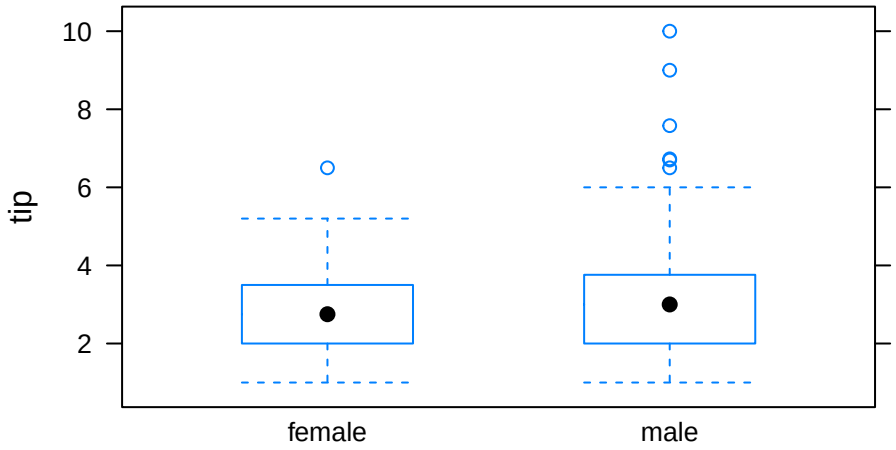

```r
#mit Geschlechtertrennung
histogram(~tip | sex, data=tips)
```



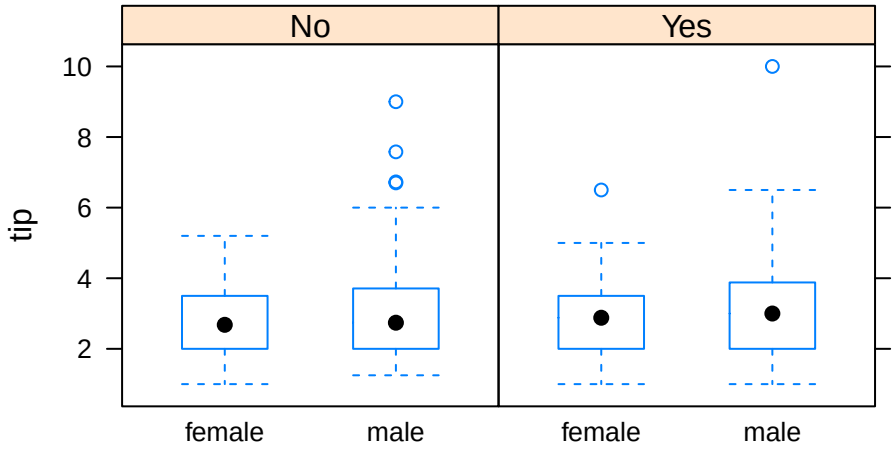

```r
#Boxplot mit metrischen Daten
bwplot(~tip, data=tips)
```



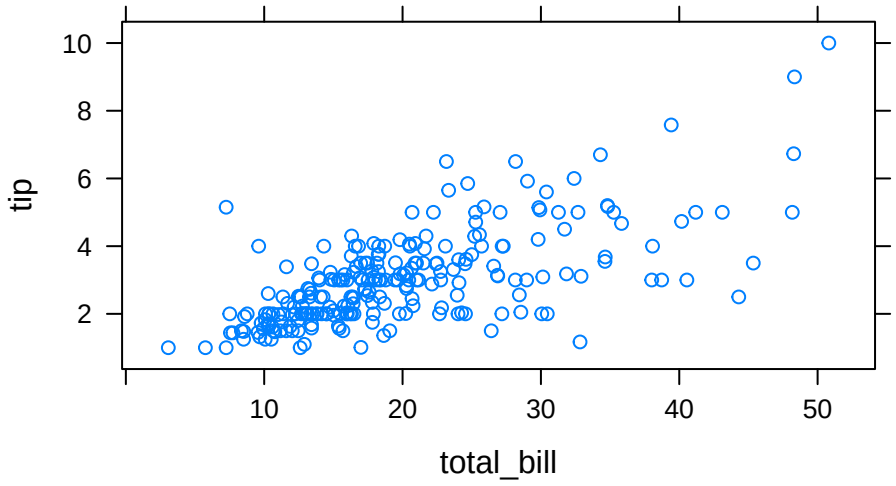

```
#Boxplot mit metrischen Daten für Gruppen
bwplot(tip~sex, data=tips)
```




```
bwplot(tip~sex |smoker, data=tips)
```




```
#Scatterplot (Streudiagramm) mit zwei metrischen Variablen
xyplot(tip~total_bill, data=tips)
```



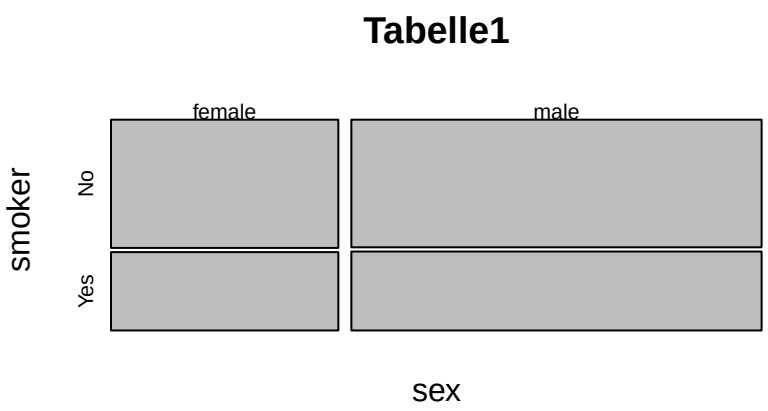

```
#mosaicplot mit zwei kategorialen Variablen
#Vorher muss eine Tabelle mit dem Befehl tally generiert werden
tally(sex~smoker, data=tips)
```

```
##          smoker
## sex       No Yes
##   female 54  33
##   male   97  60
```

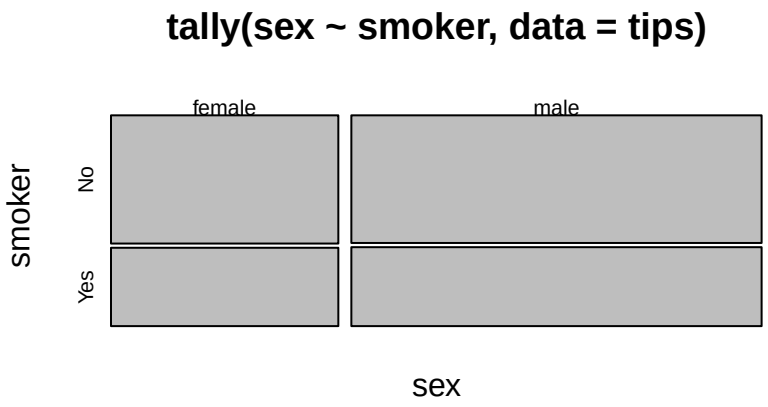
```
#Tabelle muss einem Objekt zugewiesen werden
Tabelle1<-tally(sex~smoker, data=tips)

#Mit Tabelle1 kann nun ein mosaci plot generiert werden
```

```r
mosaicplot(Tabelle1)
```



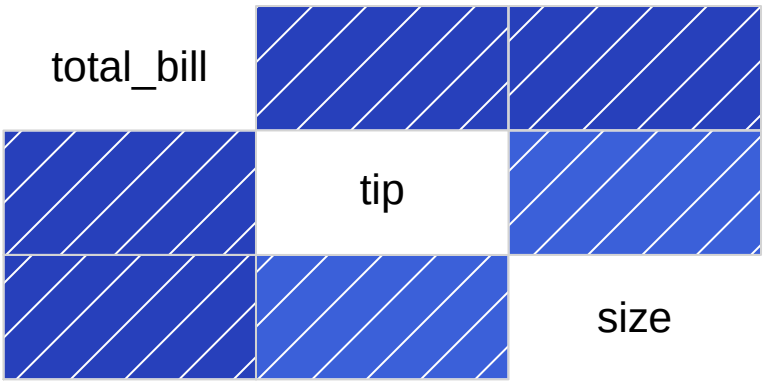

```r
#...oder alles zusammen (Das gilt übrigens grundsätzlich: Es kann über
#die Objektebene gegangen werden oder der Befehle direkt ausgeführt werden,
#siehe hierzu auch weiter unten)
mosaicplot(tally(sex~smoker, data=tips))
```




```r
#Eine Tabelle kann auch relativ berechnet werden mit dem Argument
#format="proportions"
tally(sex~smoker, format="proportion", data=tips)
```

```
##         smoker
## sex             No       Yes
##   female 0.3576159 0.3548387
##   male   0.6423841 0.6451613
```

```
#Korrelationsplot mit metrischen Variablen (hierzu das Paket corrgram laden
library(corrgram)
#Alle nominalen Variablen im Datensatz werden nicht berücksichtigt
corrgram(tips)
```



Kennzahlen

```
#Mittelwert
mean(tip~sex, data=tips)
```

```
##   female     male
## 2.833448 3.089618
```

```
#Anstatt mean können alle Lageparameter und Streumaße errechnet werden
#(min, max, median, sd, var)

#Favorisierte Statistiken werden ausgegeben mit
favstats(tip~sex, data=tips)
```

```
##      sex min Q1 median   Q3  max     mean       sd   n missing
## 1 female   1  2   2.75 3.50  6.5 2.833448 1.159495  87       0
## 2   male   1  2   3.00 3.76 10.0 3.089618 1.489102 157       0
```

```
#Korrelation als Zusammenhangsmaß mit metrischen Variablen
cor(tip~total_bill, data=tips)
```

```
## [1] 0.6757341
```

Aggregation von Variablen

Wenn über mehrere Variablen oder Dimensionen eine neue Variablen oder Dimension (z. B. durch mean) gebildet werden soll, dann eignet sich für die Aggregation der Daten der `apply` Befehl oder der `rowMeans` Befehl.
(Achtung: Im tips Datensatz macht das hier keinen Sinn, deswegen wir nur der Befehl exemplarische angegeben)

Datensatz$Neue_Variable <- apply(Datensatz[,c("Variable1","Variable2", "etc..")],1,mean,na.rm=TRUE)

Datensatz$Neue_Variable <- rowMeans(Datensatz[,c("Variable1","Variable2", "etc.."), na.rm=TRUE]

Praxis: In der Regel wir der Befehl dann benötigt, wenn nach einer PCA oder einer EFA die Dimensionen reduziert werden.

Chi-Quadrat-Test

```
#Test der Unabhängigkeit geht nur mit zwei nominalen Variablen
#Tabelle1 haben wir schon generiert
xchisq.test(Tabelle1)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 0, df = 1, p-value = 1
##
##    54       33
## (53.84)  (33.16)
## [0.00047] [0.00077]
## < 0.022> <-0.028>
##
##    97       60
## (97.16)  (59.84)
## [0.00026] [0.00043]
## <-0.016> < 0.021>
##
## key:
##  observed
##  (expected)
##  [contribution to X-squared]
##  <Pearson residual>
```

t-Test für abhängige Stichproben (Differenzentest)

```
#Variablen müssen beide metrische sein und zwischen beiden Variablen
#wird eine Differenz gebildet
#Forschungsfrage lautet meist:
#- V1 unterscheidet sich von V2 (ungerichtet)
#- V1>V2 (gerichtet)
#- V2>V1 (gerichtet)
t.test(~(tip-total_bill), data=tips)
```

```
##
##  One Sample t-test
##
## data:  tips$(tip - total_bill)
## t = -32.647, df = 243, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -17.80057 -15.77476
## sample estimates:
## mean of x
## -16.78766
```

```
#Wenn die Forschungshypothese (Alternativhypothese) gerichtet ist, und
#V1-V2 < 0 ist, dann wird das Argument alternative="less" hinzugefügt,
#wenn V1-V2 > 0, dann "greater".
t.test(~(tip-total_bill), alternative="less", data=tips)
```

```
##
##  One Sample t-test
##
## data:  tips$(tip - total_bill)
## t = -32.647, df = 243, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##      -Inf -15.9386
## sample estimates:
## mean of x
## -16.78766
```

```
#Achtung: Bei der Dokumentation von t-Tests ist es wichtig, einseitiges
#Testen von zweiseitigem Testen zu unterscheiden (einseitig/zweiseitig).
```

t-Test für unabhängige Stichproben

```
#Bezüglich einer Gruppe (nominale Variable) mit zwei levels wird eine
#metrische Variable getestet auf Unterschiedlichkeit (ungerichtet) oder
#größer oder kleiner (gerichtet).

#Ausgabe der levels (falls nicht mehr präsent)
levels(tips$sex)
```

```
## [1] "female" "male"
```

```
#Bei zweiseitigem Test
t.test(tip~sex, data=tips)
```

```
##
##  Welch Two Sample t-test
##
## data:  tip by sex
## t = -1.4895, df = 215.71, p-value = 0.1378
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5951448  0.0828057
## sample estimates:
## mean in group female   mean in group male
##             2.833448             3.089618
```

```
#Bei einseitigem Test
t.test(tip~sex, alternative="less", data=tips)
```

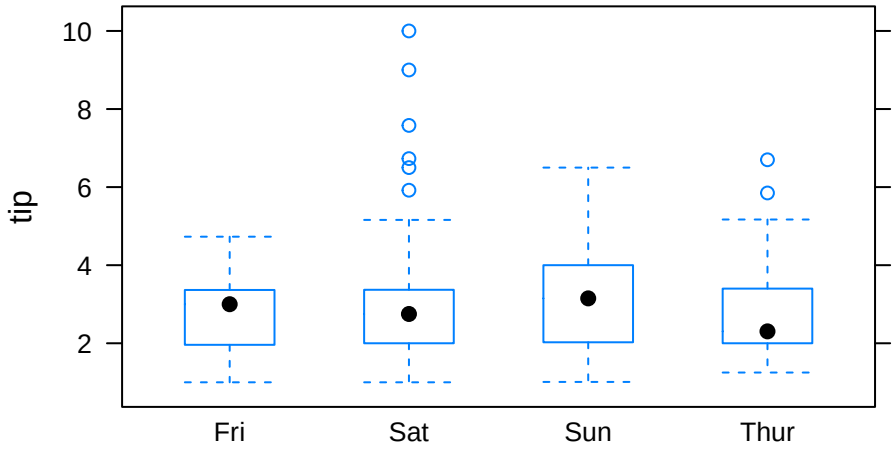
```
##
##  Welch Two Sample t-test
##
## data:  tip by sex
## t = -1.4895, df = 215.71, p-value = 0.0689
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf 0.02793151
## sample estimates:
## mean in group female   mean in group male
##             2.833448             3.089618
```

ANOVA (Varianzanalyse)

```
#Bezüglich einer Gruppe (nominale Variable) mit mehr als zwei levels
#wird eine metrische Variable getestet.
str(tips)
```

```
## 'data.frame':    244 obs. of  7 variables:
##  $ total_bill: num  17 10.3 21 23.7 24.6 ...
##  $ tip       : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ sex       : Factor w/ 2 levels "female","male": 1 2 2 2 1 2 2 2 2 2 ...
##  $ smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day       : Factor w/ 4 levels "Fri","Sat","Sun",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ time      : Factor w/ 2 levels "Dinner","Lunch": 1 1 1 1 1 1 1 1 1 1 ...
##  $ size      : int  2 3 3 2 4 4 2 4 2 2 ...
```

```
bwplot(tip~day, data=tips)
```




```
favstats(tip~day, data=tips)
```

```
##    day  min     Q1 median     Q3   max     mean       sd  n missing
## 1  Fri 1.00 1.9600  3.000 3.3650  4.73 2.734737 1.019577 19       0
## 2  Sat 1.00 2.0000  2.750 3.3700 10.00 2.993103 1.631014 87       0
## 3  Sun 1.01 2.0375  3.150 4.0000  6.50 3.255132 1.234880 76       0
## 4 Thur 1.25 2.0000  2.305 3.3625  6.70 2.771452 1.240223 62       0
```

```
#Forschungshypothese: Es gibt einen Unterschied beim Trinkgeld
#bei/zwischen den Tagen.
summary(aov(tip~day, data=tips))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## day           3    9.5   3.175   1.672  0.174
```

```
## Residuals   240  455.7   1.899
```

Lineare Einfachregression mit metrischer UV

```
#Modellierung einer abhängigen Variable (AV) durch eine unabhängige
#Variable (UV).
Mod1<-lm(tip~total_bill, data=tips)
summary(Mod1)
```

```
##
## Call:
## lm(formula = tip ~ total_bill, data = tips)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1982 -0.5652 -0.0974  0.4863  3.7434
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.920270   0.159735   5.761 2.53e-08 ***
## total_bill  0.105025   0.007365  14.260  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.022 on 242 degrees of freedom
## Multiple R-squared:  0.4566, Adjusted R-squared:  0.4544
## F-statistic: 203.4 on 1 and 242 DF,  p-value: < 2.2e-16
```

```
# Modell lautet immer: AV = interept + Steigung * UV
```

Lineare Einfachregression mit kategorialer UV

```
Mod2<-lm(tip~day, data=tips)
summary(Mod2)
```

```
##
## Call:
## lm(formula = tip ~ day, data = tips)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.2451 -0.9931 -0.2347  0.5382  7.0069
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.73474    0.31612   8.651 7.46e-16 ***
## daySat       0.25837    0.34893   0.740    0.460
## daySun       0.52039    0.35343   1.472    0.142
## dayThur      0.03671    0.36132   0.102    0.919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.378 on 240 degrees of freedom
## Multiple R-squared:  0.02048,    Adjusted R-squared:  0.008232
## F-statistic: 1.672 on 3 and 240 DF,  p-value: 0.1736
```

```
#Achtung: Das nicht ausgegebene level in der Ausgabe ist das Referenzlevel.
```

Multiple Regression

```
str(tips)
```

```
## 'data.frame':    244 obs. of  7 variables:
##  $ total_bill: num  17 10.3 21 23.7 24.6 ...
##  $ tip       : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ sex       : Factor w/ 2 levels "female","male": 1 2 2 2 1 2 2 2 2 2 ...
##  $ smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day       : Factor w/ 4 levels "Fri","Sat","Sun",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ time      : Factor w/ 2 levels "Dinner","Lunch": 1 1 1 1 1 1 1 1 1 1 ...
##  $ size      : int  2 3 3 2 4 4 2 4 2 2 ...
```

```
Mod3<-lm(tip~total_bill + sex + smoker + day + time + size, data=tips)
summary(Mod3)
```

```
##
## Call:
## lm(formula = tip ~ total_bill + sex + smoker + day + time + size,
##     data = tips)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8475 -0.5729 -0.1026  0.4756  4.1076
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.803817   0.352702   2.279   0.0236 *
## total_bill   0.094487   0.009601   9.841   <2e-16 ***
## sexmale     -0.032441   0.141612  -0.229   0.8190
## smokerYes   -0.086408   0.146587  -0.589   0.5561
## daySat      -0.121458   0.309742  -0.392   0.6953
## daySun      -0.025481   0.321298  -0.079   0.9369
## dayThur     -0.162259   0.393405  -0.412   0.6804
## timeLunch    0.068129   0.444617   0.153   0.8783
## size         0.175992   0.089528   1.966   0.0505 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.024 on 235 degrees of freedom
## Multiple R-squared:  0.4701, Adjusted R-squared:  0.452
## F-statistic: 26.06 on 8 and 235 DF,  p-value: < 2.2e-16
```

```
#Mit dem Befehl step führt man eine stufenweise Regressionsanalyse durch,
#bei der die Variablen nach der Reihenfolge ihrer Wichtigkeit entfernt werden.
step(Mod3)
```

```
## Start:  AIC=20.51
## tip ~ total_bill + sex + smoker + day + time + size
##
##              Df Sum of Sq    RSS     AIC
## - day         3     0.609 247.14  15.116
## - time        1     0.025 246.55  18.538
## - sex         1     0.055 246.58  18.568
## - smoker      1     0.365 246.89  18.874
## <none>                    246.53  20.513
## - size        1     4.054 250.58  22.493
## - total_bill  1   101.595 348.12 102.713
##
## Step:  AIC=15.12
## tip ~ total_bill + sex + smoker + time + size
##
##              Df Sum of Sq    RSS    AIC
## - time        1     0.001 247.14 13.117
## - sex         1     0.042 247.18 13.157
## - smoker      1     0.380 247.52 13.490
## <none>                    247.14 15.116
```

```
## - size        1     4.341 251.48 17.365
## - total_bill  1   101.726 348.86 97.232
##
## Step:  AIC=13.12
## tip ~ total_bill + sex + smoker + size
##
##              Df Sum of Sq    RSS    AIC
## - sex         1     0.041 247.18 11.157
## - smoker      1     0.379 247.52 11.491
## <none>                    247.14 13.117
## - size        1     4.342 251.48 15.366
## - total_bill  1   103.327 350.46 96.350
##
## Step:  AIC=11.16
## tip ~ total_bill + smoker + size
##
##              Df Sum of Sq    RSS    AIC
## - smoker      1     0.376 247.55  9.528
## <none>                    247.18 11.157
## - size        1     4.344 251.52 13.408
## - total_bill  1   104.263 351.44 95.029
##
## Step:  AIC=9.53
## tip ~ total_bill + size
##
##              Df Sum of Sq    RSS    AIC
## <none>                    247.55  9.528
## - size        1     5.235 252.79 12.634
## - total_bill  1   106.281 353.83 94.685

##
## Call:
## lm(formula = tip ~ total_bill + size, data = tips)
##
## Coefficients:
## (Intercept)   total_bill         size
##     0.66894      0.09271      0.19260
```

Literatur

Pflichtliteratur:

Intro Stat with Randomization and Simulation (klick here..)

Versionshinweise:

- Datum erstellt: 2017-08-03
- R Version: 3.4.0
- `mosaic` Version: 0.14.4