Modul Statistische Aspekte der Analyse molekularbiologischer und genetischer Daten

R-Blatt 2: Deskriptive Statistiken in R

Janne Pott

WS 2021/22

In dieser Übung wird R genutzt, um deskriptive Statistiken zu erstellen.

Deskriptive Statistik in R

Beispiele

In dem ersten Teil der Übung beschäftigen wir uns mit deskriptiven Statistiken. Dazu wird der Datensatz *ergometer.RData* verwendet. Dabei handelt es sich um eine R-spezifische Datei, die schneller in R einlesbar ist und alle R-spezifische Informationen in den Daten (z. B. Attribute, Variablentypen usw.) mitspeichert. Das beinhaltet auch die Variablennamen. Wurde als der Datensatz *myTab* mittels *save(myTab,file="test.RData")* gespeichert, wird wieder die Variable *myTab* erzeugt. Wenn diese schon besteht, wird sie einfach überschrieben! Man kann sich aber den oder die Variablennamen mitangeben lassen.

```
loaded1<-load("data/ergometer.RData")
loaded1
```

```
## [1] "myDat"
```

```
class(myDat)
```

```
## [1] "data.frame"
```

```
var<-colnames(myDat)
dummy1<-c("Durchlaufende ID-Nummer","Geschlecht","Geburtstag","Erhebungsdatum",
          "Größe","Gewicht","Leistung im Ergometer","Milchsäure im Blut")
dummy2<-c(NA,"1 = Mann; 2 = Frau","Monat/Tag/Jahr","Monat/Tag/Jahr",
           "in m","in kg","in Watt/kg","in mg/dl")
dumTab<-data.frame(var,dummy1,dummy2)
knitr::kable(dumTab, position = "!b",
             caption = "Parameterbeschreibung zum Datensatz ergometer",
             col.names = c("Variable","Beschreibung","Codierung / Einheit"))
```

Table 1: Parameterbeschreibung zum Datensatz ergometer

Variable	Beschreibung	Codierung / Einheit
id	Durchlaufende ID-Nummer	NA
sex	Geschlecht	1 = Mann; 2 = Frau
Bday	Geburtstag	Monat/Tag/Jahr
Tday	Erhebungsdatum	Monat/Tag/Jahr
height	Größe	in m
weight	Gewicht	in kg
ergometer	Leistung im Ergometer	in Watt/kg
lactate	Milchsäure im Blut	in mg/dl

In diesem Beispiel heißt der Datensatz also *myDat*. Um Verwechslungen zu verweiden, kann man diesen auch umbenennen:

```
myDat2<-myDat
myDat3<-copy(myDat)
myDat4<-get(loaded1)
```

Der Befehl *copy* ist vor allem in der **data.table** Syntax wichtig, da sonst nicht eine vollständige Kopie angelegt wird (Änderungen in *myDat2* würden sich dann auch auf *myDat* auswirken).

Mit *get* kann man einfach den Variablenname des Objekts angeben, der gesucht und dann einem neuen Namen zugeordnet werden soll. Das ist z.B. bei Schleifen hilfreich, wenn pro Schleife ein RData-Objekt geladen wird, aber die einzelnen Objekte anderes heißen (und man auch nicht weiß wie).

Oft muss man Datensätze noch etwas anpassen, bevor man sie auswerten kann. Wenn das Datum zum Beispiel nicht als solches erkannt wird, kann man das Alter nicht direkt ausrechen. Ich nutze hier die Funktion *mdy()* aus dem Paket **lubridate**, um das Alter zu transformieren, das im Format Monat-Tag-Jahr angegeben ist. Trennzeichen werden hier automatisch erkannt.

```
head(myDat)
```

```
##   id sex      Bday      Tday height weight ergometer lactate
## 1  1   2 5/27/1958 5/27/2005   1.59   59.6      3.34      11
## 2  2   2 4/14/1958 4/14/2005   1.77   76.8      3.19      11
## 3  3   2  1/4/1957  1/4/2005   1.72   72.5      2.76      12
## 4  4   2 2/17/1955 2/17/2005   1.65   63.0      2.87      11
## 5  5   2  4/6/1954  4/6/2005   1.67   60.7      2.27      11
## 6  6   2 3/27/1954 3/27/2005   1.63   71.0      2.93      14
```

```
class(myDat$Bday)
```

```
## [1] "character"
```

```
date1<-mdy(myDat$Bday)
date2<-mdy(myDat$Tday)
class(date1)
```

```
## [1] "Date"
```

```
head(difftime(date2, date1, unit="weeks"))
```

```
## Time differences in weeks
## [1] 2452.429 2452.429 2504.571 2609.000 2661.143 2661.143
```

```
head(difftime(date2, date1, unit="weeks")/52.25)
```

```
## Time differences in weeks
## [1] 46.93643 46.93643 47.93438 49.93301 50.93096 50.93096
```

```
round(head(difftime(date2, date1, unit="weeks")/52.25),2)
```

```
## Time differences in weeks
## [1] 46.94 46.94 47.93 49.93 50.93 50.93
```

```
class(round(head(difftime(date2, date1, unit="weeks")/52.25),2))
```

```
## [1] "difftime"
```

```
setDT(myDat)
myDat[,alter:=as.numeric(round(difftime(date2, date1, unit="weeks")/52.25,2))]
```

Die Deskription umfasst unter anderem das Minimum, Maximum, Mittelwert und die Quartile. Zusätzlich werden Standardabweichung oder Varianz mitangegeben. Man kann rein optisch auf Normalverteilung prüfen (QQ-Plot, Histogramm), oder mittels Kolmogorov-Smirnov Test auf eine signifikante Abweichung davon testen. Ist dieser signifikant, sollte man parameterfreie Test für weitere Analysen verwenden. Der Mann-Whitney U Test liefert beim Geschlechtsvergleich ein signifikantes Ergebnis, d.h. das Altersmittel der Männer ist in diesem Datensatz signikant höher als das in Frauen.

```
myDat[,summary(alter)]
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.94   58.92   64.91   64.77   71.90   76.90
```

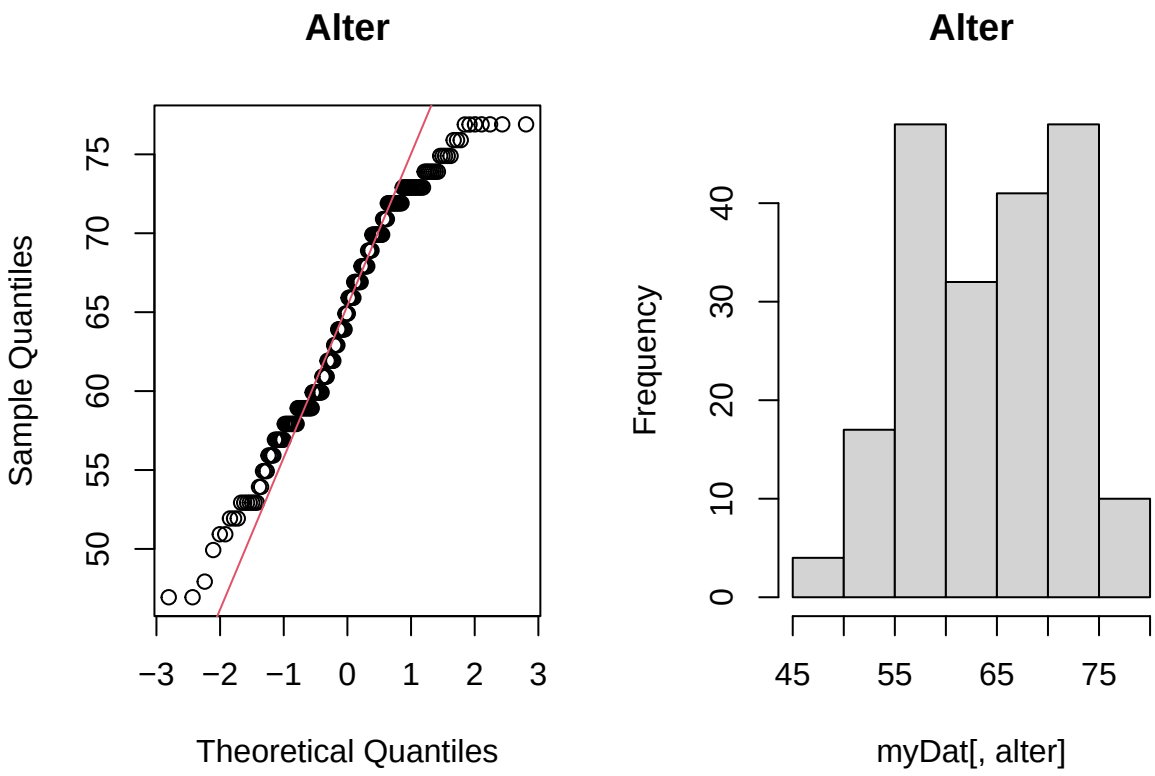
```
myDat[,sd(alter)]
```

```
## [1] 7.368519
```

```
myDat[,var(alter)]
```

```
## [1] 54.29507
```

```
par(mfrow = c(1,2)) # zwei Plots nebeneinander
qqnorm(myDat[,alter],main = "Alter"); qqline(myDat[,alter], col = 2)
hist(myDat[,alter],breaks = 10,main = "Alter")
```




```
ks.test(myDat[,alter],pnorm,mean=mean(myDat[,alter]),sd=sd(myDat[,alter]))
```

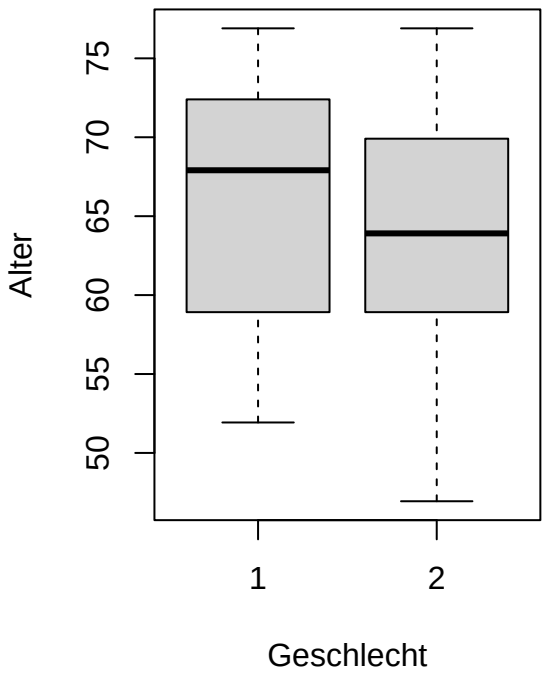
```
## Warning in ks.test(myDat[, alter], pnorm, mean = mean(myDat[, alter]), sd =
## sd(myDat[, : ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  myDat[, alter]
## D = 0.10719, p-value = 0.02019
## alternative hypothesis: two-sided
```

```
wilcox.test(myDat[,alter] ~ myDat[,sex])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  myDat[, alter] by myDat[, sex]
## W = 5870.5, p-value = 0.02685
## alternative hypothesis: true location shift is not equal to 0
```

```
boxplot(myDat[,alter] ~ myDat[,sex],
        xlab="Geschlecht",ylab="Alter")
```



Aufgaben

a) Berechnen Sie den *BMI* der Probanden und bestimmen Sie deskriptive Statistiken für die Größen *ergometer*, *lactate*, *BMI* und *Alter* für Männer und Frauen getrennt.

b) Erstellen Sie QQ-Plots und Histrogramme und testen Sie auf Normalverteilung.

c) Vergleichen Sie *ergometer* zwischen den Geschlechtern unter Verwendung eines geeigneten Tests.

d) Korrelieren Sie *ergometer* mit *lactate*, *BMI* und *Alter*.

Gepaarte Tests

Beispiele

In dem ersten Teil der Übung beschäftigen wir uns mit gepaarten Tests. Dazu wird der Datensatz *haendigkeit.RData* verwendet. Laden Sie den Datensatz **haendigkeit.RData** in R ein.

```
rm(list = setdiff(ls(), c("cor.prob","pathwd","r_on_server")))

loaded2<-load("data/haendigkeit.RData")
loaded2
```

```
## [1] "myDat"
```

```
var<-colnames(myDat)
dummy1<-c("Durchlaufende ID-Nummer","Geschlecht","Länge der Schreibhand",
          "Länge der Nichtschreibhand","Schreibhand",
          "Präferenz für Armverschränkung","Präferenz für Klatschen","Größe")
dummy2<-c(NA,"1 = Mann; 2 = Frau","in cm","in cm",
          "0 = rechts; 1 = links",
          "0 = rechts auf links; 0.5 = keine; 1 = links auf rechts",
          "0 = rechts; 0.5 = keine; 1 = links",
          "in m")
dumTab<-data.frame(var,dummy1,dummy2)
knitr::kable(dumTab, position = "!b",
             caption = "Parameterbeschreibung zum Datensatz haendigkeit",
             col.names = c("Variable","Beschreibung","Codierung / Einheit"))
```

Table 2: Parameterbeschreibung zum Datensatz haendigkeit

Variable	Beschreibung	Codierung / Einheit
id	Durchlaufende ID-Nummer	NA
sex	Geschlecht	1 = Mann; 2 = Frau
WrHnd	Länge der Schreibhand	in cm
NWHnd	Länge der Nichtschreibhand	in cm
WHnd	Schreibhand	0 = rechts; 1 = links
Fold	Präferenz für Armverschränkung	0 = rechts auf links; 0.5 = keine; 1 = links auf rechts
Clap	Präferenz für Klatschen	0 = rechts; 0.5 = keine; 1 = links
height	Größe	in m

```
setDT(myDat)
```

Manchmal sind die Daten unvollständig oder enthalten Ausreißer. Je nach Anteil kann man diese vollständig filtern, oder nur für einzelne Tests.

Fehlende Werte sind in der Regel durch *NA* gekennzeichnet. Mittels *apply* kann man sich die Anzahl der *NA* pro Spalte angeben lassen. Für dieses Beispiel filtere ich die einzelnen *NA*s, aber belasse die *NA*s der Spalte *height*.

Wenn man die Differnez der Handlängen betrachtet, erwartet man ähnliche Größen (1-2 cm). Größere Abweichungen sind unplausibel (z.B. durch falsche Dateneingabe) und/oder könnten die Analyse verzerren,

und sollten daher gefiltert werden. Dazu kann man entweder den Plot nutzen und den "offensichtlichen" Ausreißer filtern, oder man legt eine Grenze fest, zum Beispiel eine Abweichung von mehr als $4 * SD$ vom Mittelwert.

```
apply(myDat, MARGIN = 2, function(x) sum(is.na(x)))
```

```
##     id    sex  WrHnd  NWHnd   WHnd   Fold   Clap height
##      0      1      1      1      1      0      1     28
```

```
filt<-!is.na(myDat$sex) & !is.na(myDat$WrHnd) & !is.na(myDat$NWHnd) & !is.na(myDat$WHnd) & !is.na(myDat$
table(filt)
```

```
## filt
## FALSE  TRUE
##     3   234
```

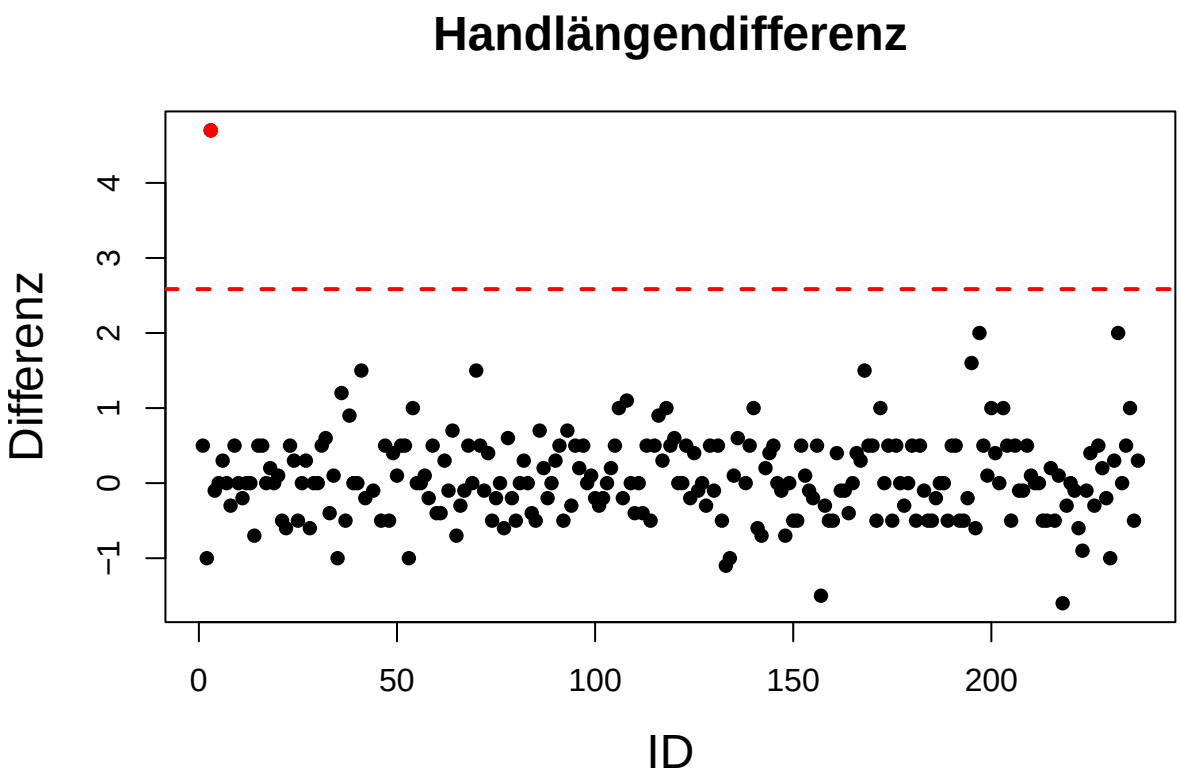
```
myDat<-myDat[filt,]
apply(myDat, MARGIN = 2, function(x) sum(is.na(x)))
```

```
##     id    sex  WrHnd  NWHnd   WHnd   Fold   Clap height
##      0      0      0      0      0      0      0     28
```

```
myDat[,dif:=WrHnd-NWHnd]
filt<-myDat$dif>mean(myDat$dif,na.rm = T) + 4*sd(myDat$dif,na.rm = T)
myDat[filt,]
```

```
##    id sex WrHnd NWHnd WHnd Fold Clap height dif
## 1:  3   1    18  13.3    0    1  0.5     NA 4.7
```

```
plot(myDat$id, myDat$dif, main="Handlängendifferenz",
     xlab = "ID", ylab="Differenz",pch=16,cex.main=1.5,cex.lab=1.5)
points(myDat$id[filt],myDat$dif[filt],col="red",pch=16)
abline(h=mean(myDat$dif,na.rm = T) + 4*sd(myDat$dif,na.rm = T),lwd=2,col="red",lty=2)
```




```
myDat2<-myDat[!filt,]
attach(myDat2)
```

Um binäre Variablen zu beschreiben, eignet sich die absolute bzw. relative Häufigkeit der verwendeten Kategorien. Um zwei Variablen zu vergleichen, benutzt man Kontingenztafel. Es gibt zwei Tests auf Unabhängigkeit:

- Chi-Quadrat-Test (stochastische Unabhängigkeit zweier Merkmale)
- Exakter Test nach Fisher (keine Voraussetzungen an den Stichprobenumfang, robuster)

```
myDat2[,.N,by = .(sex)]
```

```
##    sex   N
## 1:   2 117
## 2:   1 116
```

```
myDat2[,.N,by = .(WHnd)]
```

```
##    WHnd   N
## 1:    0 216
## 2:    1  17
```

```
myDat2[,.N,by = .(sex,WHnd)]
```

```
##    sex WHnd   N
## 1:   2    0 110
## 2:   1    1  10
## 3:   1    0 106
## 4:   2    1   7
```

```
myDat2[,table(sex,WHnd)]
```

```
##    WHnd
## sex   0   1
##   1 106  10
##   2 110   7
```

```
fisher.test(sex,WHnd)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  sex and WHnd
## p-value = 0.4627
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2100113 2.0499856
## sample estimates:
## odds ratio
##  0.6756833
```

```
chisq.test(sex,WHnd)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  sex and WHnd
## X-squared = 0.27267, df = 1, p-value = 0.6015
```

Aufgaben

a) Berechnen Sie geeignete deskriptive Statistiken für die Variablen *sex*, *WrHnd*, *NWHnd*, *WHnd*, *Fold*, *Clap*, und *height*!

b) Testen Sie, ob es Unterschiede in den Handlängen zwischen Männern und Frauen gibt. Konstruieren Sie Boxplots.

c) Testen Sie für Männer und Frauen getrennt, ob es Längenunterschiede zwischen Schreib- und Nichtschreibhand gibt.

d) Analysieren Sie die Beziehung zwischen Schreibhand, Armverschränkung und Klatschen, dabei die unentschiedenen Fälle filtern.

e) Testen Sie für Männer und Frauen getrennt, ob es Beziehungen zwischen Größe, Länge der Hand und Unterschied zwischen Schreib-/Nichtschreibhand gibt.