Course of Study Bachelor Computer Science	Exercises Statistics WS 2022/23
Sheet III - Solutions	

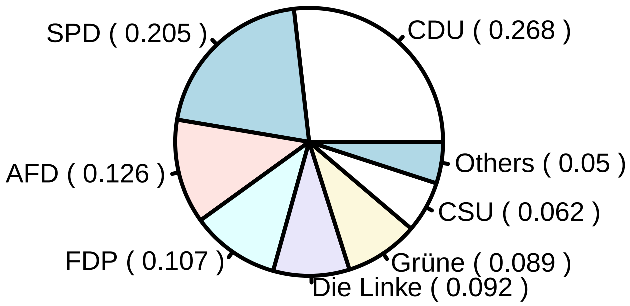
Descriptive Statistics - Frequency Tables and Distributions

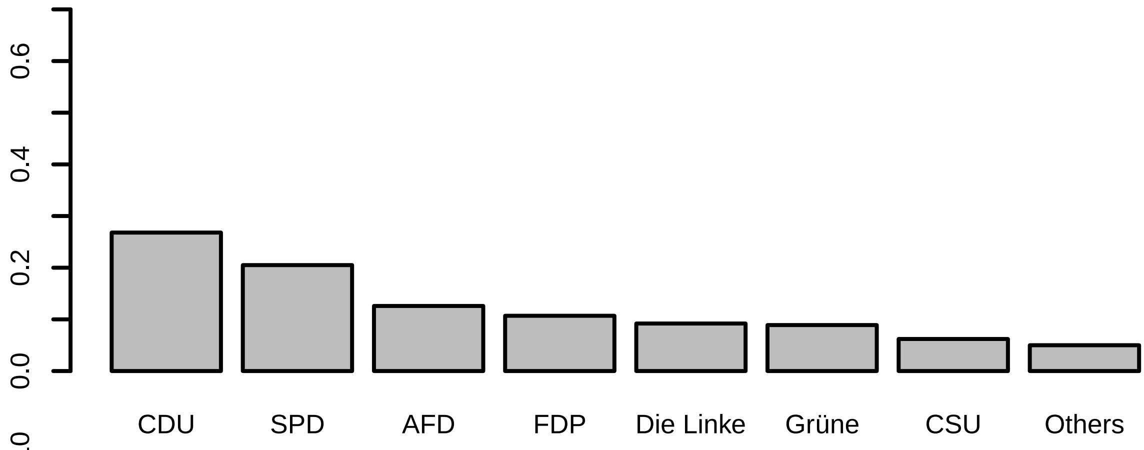
1. Consider the results of the national elections in Germany in 2013 and 2017:

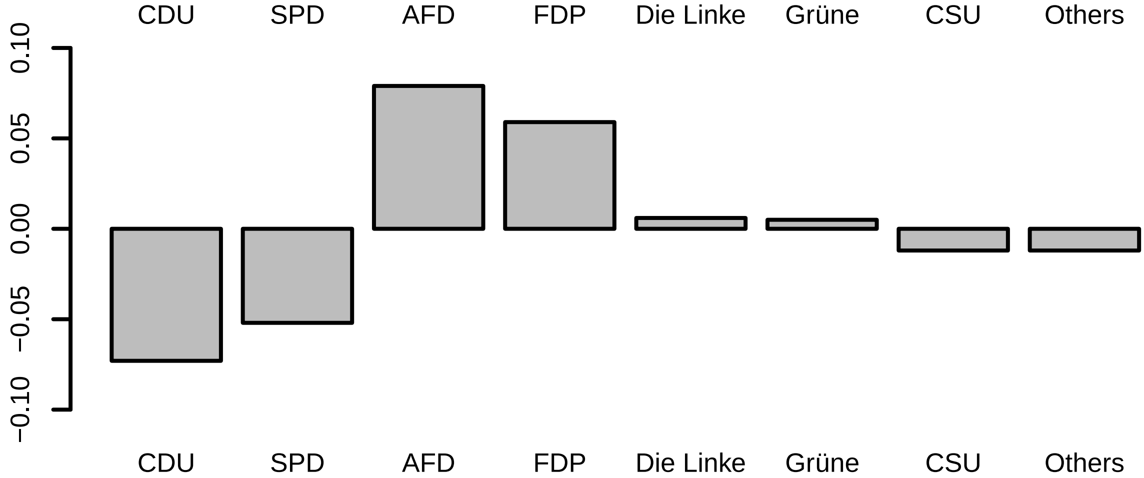
Party	Results 2013 (%)	Results 2017 (%)
CDU	26,8%	34,1%
SPD	20,5%	25,7%
AfD	12,6%	4,7%
FDP	10,7%	4,8%
DIE LINKE	9,2%	8,6%
GRUENE	8,9%	8,4%
CSU	6,2%	7,4%
Others	5,0%	6,2%

Summarize the results of 2017 in a pie and abar chart. Compare the results in 2013 and 2017 with an appropriate bar chart.

Answer: The table shows the relative frequencies of each party. We can draw a pie chart and a barplot with the parties on the x-axis and the relative frequencies on the y-axis. To compare the ersults in 2013 and in 2017 we can show the differences in proportion of votes in barplot.








```
######################################################
# Descriptive Statistics: National Elections
#
# File: des_stat_nat_el.R
#
######################################################
library(tidyverse)

# Results of national elections
```

```
results2017 <- c(0.268,0.205,0.126,0.107,0.092,
                 0.089,0.062,0.05)
results2013 <- c(0.341,0.257,0.047,0.048,0.086,
                 0.084,0.074,0.062)
difference <- results2017-results2013
party <- c("CDU","SPD","AFD","FDP","Die Linke","Gruene","CSU","Others")

# applying tibbles
nat_el <- tibble(
  res.2017 = c(0.268,0.205,0.126,0.107,0.092,0.089,0.062,0.05),
  res.2013 = c(0.341,0.257,0.047,0.048,0.086,0.084,0.074,0.062),
  party = c("CDU","SPD","AFD","FDP","Die Linke","Gruene","CSU","Others")
) %>%
  mutate(
    diff = res.2017 - res.2013
  )
nat_el

# You can adjust the size of the margins by specifying a margin parameter
# using the syntax par(mar = c(bottom, left, top, right)), where the
# arguments bottom, left are the size of the margins. The default value
# for mar is c(5.1, 4.1, 4.1, 2.1). To change the size of the margins of a
# plot you must do so with par(mar) before you actually create the plot.
# to increase plot margins on the side of the figure .

# mfrow A vector of length 2, where the first argument specifies the
# number of rows and the second the number of columns of plots.

# cex: A numerical value giving the amount by which plotting text and
# symbols should be magnified relative to the default. This starts as 1
# when a device is opened, and is reset when the layout is changed, e.g.
# by setting mfrow.

# To ensure that large labels stay in figure we choose mar= c(2, 2, 0.5, 0.5).
# To have the plots below each other we choose mfrow = c(3,1)
# To ensure that the text of labels fits in the diagram we set cex=0.45
par(mar= c(2, 2, 0.5, 0.5), mfrow=c(3,1), cex = 0.45)

pie(results2017, labels = paste(party,"(",results2017,")"))

barplot(results2017,names.arg=party,
        ylim=c(0,0.7), xlab="Parties",ylab="2017 Votes (%)")
barplot(difference, names.arg=party,
        ylim=c(-0.1,0.1),
        xlab="Parties",ylab="Difference to 2103")

# eps-file
dev.copy2eps(file="../pictures/national_elections.eps")

# diagrams with ggplot
# ggplot ordes the bars according to the alphabetic order of the x values, here party.
# The order can be changed by adding a factor to the variably party where the levels
# represents the newly defined order.
nat_el$party <- factor(nat_el$party,
                       c("CDU","SPD","AFD","FDP","Die Linke","Gruene","CSU","Others"))
ggplot(data = nat_el, mapping = aes(x = "", y = results2017, fill = party)) +
  geom_col(width = 1) +
  coord_polar(theta = "y") +
  geom_text(mapping = aes(label = paste(party,"(",results2017*100,")")),
            position = position_stack(vjust = 0.5)) +
  theme_void()
ggplot(data = nat_el) +
  geom_col(mapping = aes(x=party,y=results2017)) +
  xlab("Parties")+
  ylab("2017 Votes (%)") +
  theme_bw()
ggplot(data = nat_el) +
  geom_col(mapping = aes(x=party, y=diff)) +
  xlab("Parties")+
  ylab("Difference to 2013") +
  theme_bw()
```

2. The data shown in the list are the times in milliseconds it took one of us to move the mouse over a small target in a series of 20 trials. The times are sorted from shortest to longest.

   568, 577, 581, 640, 641, 645, 657, 673, 696, 703, 720, 728, 729, 777, 808, 824, 825, 865, 875, 1007

(a) Compute and draw the cumulative frequency distribution.

(b) Compute using the cumulative frequency distribution the proportion of response times

i. less equal 800

ii. greater than 725

iii. greater than 642 and less equal 777

iv. equal 696

in the sample.

(c) Consider the following classes $(500, 600]$, $(600, 700]$, $(700, 800]$, $(800, 900]$, $(900, 1000]$, $(1000, 1100]$.

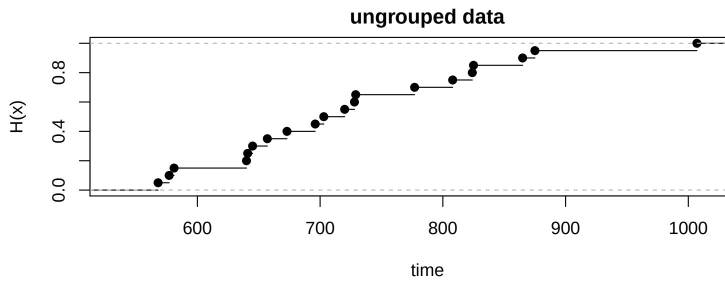
- Compute the grouped frequency distribution and draw the histogram.

(d) The classes are now $(500, 600]$, $(600, 900]$,$(1000, 1200]$. Mention that the classes have different width. Compute the grouped frequency distribution and draw the histogram. Can you interpret the y-values in the diagram?

Answer:

(a) Cumulative frequency distribution with the original data

values	H(x)
568.00	0.05
577.00	0.10
581.00	0.15
640.00	0.20
641.00	0.25
645.00	0.30
657.00	0.35
673.00	0.40
696.00	0.45
703.00	0.50
720.00	0.55
728.00	0.60
729.00	0.65
777.00	0.70
808.00	0.75
824.00	0.80
825.00	0.85
865.00	0.90
875.00	0.95
1007.00	1.00

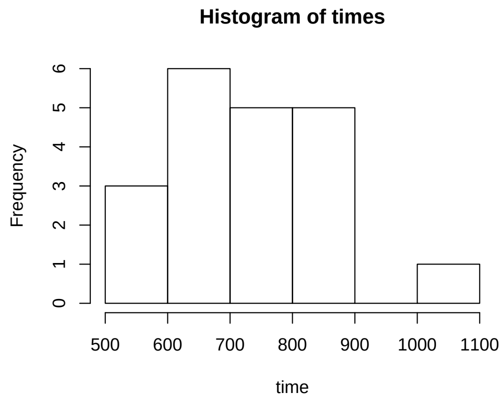


(b) Compute the proportion of response times

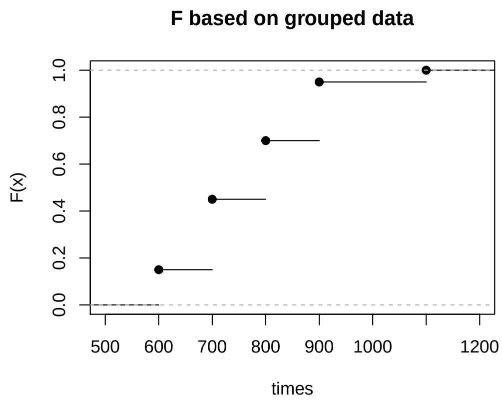
- less equal 800: H(800) = 0.7
- greater than 725: 1-H(725) = 0.45
- greater than 642 and less equal 777: H(777) - H(642) = 0.45
- equal 696: $H(696) - lim_{x\uparrow 696}H(x) = 0.05$
- in [696,800]

(c) Grouped data

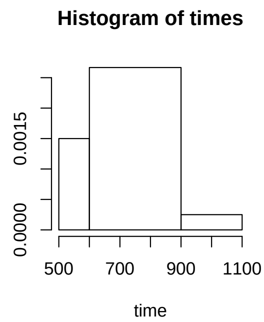
- Cumlative frequncy distribution based on grouped data



values	n	rel	$H_g(x)$
600	3	0.15	0.15
700	6	0.30	0.45
800	5	0.25	0.70
900	5	0.25	0.95
1100	1	0.05	1.00



(d) classes with different widths: $(500, 600], (600, 900], (900, 1100]$ empirical distribution for these classes



values	n	rel	$\hat{H}(x)$
600	3	0.15	0.15
900	16	0.80	0.95
1100	1	0.05	1.00

Since the width of the classes are not equally like the y-values in the histogram are not proportional to the frequencies of the classes. There is no menaningfull interpretation of the y-values.

```
##########################################################
# Descriptive Statistics: Times to move the mouse
#
# File: des_stat_time_mouse.R
#
##########################################################
library(tidyverse)
library(xtable) # only necessary to get a tex-table

# 4) The data shown in the list are the times in
# milliseconds it took one of us to move the mouse
# over a small target in a series of 20 trials.
# The times are sorted from shortest to longest.

times <- c(568, 577, 581, 640, 641, 645, 657, 673, 696,
           703, 720, 728, 729, 777, 808, 824, 825, 865,
           975, 1007, 1007)

##########################################################
# ungrouped data                                         #
##########################################################
df <- tibble(values = times) %>%
  # count the number of observations per observed value
  group_by(values) %>%
  mutate(
    abs.freq = n()
    ) %>%
  unique() %>%  # remove multiple entries
  ungroup() %>% # remove group by to regard all observations
  mutate(
    rel.freq = abs.freq / sum(abs.freq),
    cum.rel.freq = cumsum(rel.freq)
    )
df

# alternative solution applying count()
tibble(values = times) %>%
  # count the number of observations per observed value
  count(values) %>%
  mutate(
    abs.freq = n,
    rel.freq = abs.freq / sum(abs.freq),
    cum.rel.freq = cumsum(rel.freq)
  ) %>%
  select(-n)

# alternative solution applying ecdf()
H <- ecdf(times)  # emp. cum. distr. function
tibble(obs.values = knots(H),       # -> observation values
       cum.rel.freq = H(obs.values)) %>%
  mutate(
    # for the smallest obs. value is cum.rel.freq = rel. freq
    # lag() -> value before the current value
    rel.freq = if_else(is.na(lag(cum.rel.freq)),cum.rel.freq,
                       cum.rel.freq - lag(cum.rel.freq)),
    abs.freq = rel.freq * length(times)  # length(times) = no of of obs.
  )

# alternative solution applying
tab <- table(times)
tibble(
  values = tab %>% names() %>% as.numeric(),
  abs.freq = tab %>% as.integer(),
  rel.freq = abs.freq / length(times) ,
  cdf = cumsum(rel.freq)
)

# b) Compute and draw the cumulative frequency distribution.
H <- ecdf(times)
H(700)

# tex Tabelle erzeugen
xtable(df_tab[,c(1,4)]) %>% print(include.rownames = FALSE, floating = FALSE)

# emp. Verteilungsfkt.
plot.ecdf(times,
          xlab = "time", ylab = "H(x)",
          main = "ungrouped data")
# eps-file
dev.copy2eps(file="../pictures/time_emp_dis1.eps")

# plot the empirical distribution function with ggplot()
ggplot(data = df) +
```

```
  stat_ecdf(mapping = aes(values)) +
  xlab("time") +
  ylab("H(x)") +
  ggtitle("empirical distribution function (step function)")

# modification: function plot
ggplot(data =
         df %>%
         mutate(x1 = values, x2 = c(values[-1],100+max(values)))) +
  geom_point(mapping = aes(x=values, y=cum.rel.freq)) +
  geom_segment(mapping = aes(x = x1, y = cum.rel.freq,
                             xend = x2, yend = cum.rel.freq)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = min(df$values)-100) +
  xlab("time") +
  ylab("H(x)") +
  ggtitle("empirical distribution function")

# c) Compute the proportion of response times
# less equal 800
H(800) # 0.7
# greater than 725
1-H(725) # 0.45
# greater than 642 and less equal 777
H(777) - H(642) # 0.45
# equal 696 --> Grenzwert
# H(696) - H(695) # 0.05
sum(df$values == 696)/length(df$values)
# in [698, 800]
H(800)-H(696)+sum(df$values == 696)/length(df$values)


#####################################################
# grouped data                                      #
#####################################################
# Consider the following classes
# (500,600],(600,700],(700,800],(800,900],(900,1000],
# (1000,1100]
# classbounds:
bounds <- c(500,600,700,800,900,1000,1100)

cut(times, breaks = bounds)

times_cut <- cut(times, breaks = bounds,
                 # labels denotes the names of values
                 # default: classes like (500,60], ...
                 # here: value = upper bound of the class
                 labels = bounds[-1]) # leave the first value

# cut(times, breaks = bounds) # labels are the classes (a,b]

df_cut <-
  tibble(upper_bound = times_cut) %>%
  group_by(upper_bound) %>%
  mutate(n = n(),
         rel = n / length(times)) %>%
  ungroup() %>%
  unique() %>% # remove multiple entries
  mutate(cum_rel_freq = cumsum(rel))
df_cut

# alternative solution applying count()
tibble(upper_bound = times_cut) %>%
  count(upper_bound) %>%
  mutate(rel = n / length(times),
         cum_rel_freq = cumsum(rel))


# alternative solution using the emp. cum. dist. function H
tibble(obs.values = bounds[-1],       # -> upper bounds of the classes
       cum.rel.freq = H(obs.values)   # cum. rel. freq. of the classes
       ) %>%
  mutate(
    # for the smallest obs. value is cum.rel.freq = rel. freq
    # lag() -> value before the current value
    rel.freq = if_else(is.na(lag(cum.rel.freq)),cum.rel.freq,
                       cum.rel.freq - lag(cum.rel.freq)),
    abs.freq = rel.freq * length(times)  # length(times) = no of of obs.
  )

# tex Tabelle erzeugen
xtable(df_cut_tab) %>% print(include.rownames = FALSE, floating = FALSE)
```

```
# Compute the grouped frequency distribution and draw the histogram.
# Histogramm
hist(times, breaks = bounds, xlab = "time")

# histogram plot applying ggplot()
ggplot(data = df) +
  geom_histogram(mapping = aes(x = values), breaks = bounds,
                 color = "grey") +
  theme_classic() +
  ggtitle("Histogram of times")

# eps-file
dev.copy2eps(file="../pictures/time_hist.eps")

# remark: coerce the values of times_cut to character and
# then to integer to get integer values!!!
# H based on grouped data
plot.ecdf(as.integer(as.character(times_cut)),
          xlab = "times", ylab = "F(x)",
          main = "F based on grouped data")

# eps-file
dev.copy2eps(file="d:/Lehre/MATHE/Statistik/Exercises/pictures/time_emp_dis2.eps")

# The following part of the exercise has been skipp ind WS 22/23!
# # H based on grouped data under the assumption of uniformly
# # distributed values in the classesd
# plot(x = c(500,as.integer(as.character(df_cut$upper_bound))),
#      y = c(0,df_cut$cum_rel_freq),
#      type = "b",
#      xlab = "time", ylab = "",
#      main = "H - grouped data",
#      sub = "Assumption: all values are uniformly distributed in the classes")
# # solution wtih ggplot()
# ggplot(data = df_cut, mapping = aes(x= upper_bound, y=cum_rel_freq)) +
#   geom_point() +
#   geom_line(group=1) +
#   xlab("time") +
#   ylab("") +
#   ggtitle("H - grouped data with the assumption: all values are uniformly distributed in the classes") +
#   theme_classic()
#
# # eps-file
# dev.copy2eps(file="../pictures/times_emp_dis3.eps")
#
# # empirical distribution function based on grouped data
# # assumming uniformly distributed values in the classes
# xval <- c(500,as.integer(as.character(df_cut$upper_bound)))
# yval <- c(0,df_cut$cum_rel_freq)
# H_gr_uni <- function(x,xval,yval) {
#   if ( x < xval[1]) {
#     return(0)
#   } else {
#     if (x > max(xval)) {
#       return(1)
#     } else {
#       i <- max(which(x >= xval))
#       return(yval[i] + (yval[i+1]-yval[i])*(x-xval[i])/(xval[i+1]-xval[i]))
#     }
#   }
# }
#
# # Compute the proportion of response times
# # less equal 800
# H_gr_uni(800,xval, yval) # 0.7
# # greater than 725
# 1-H_gr_uni(725,xval, yval) # 0.4875
# # greater than 642 and less equal 777
# H_gr_uni(777,xval, yval) - H_gr_uni(642,xval, yval) # 0.3665
# # equal 696 --> Grenzwert: 0

####################################################
# grouped data: different class widths             #
####################################################
# Consider the following classes
# (500,600],(600,900],(900,1100]
# classbounds:
bounds <- c(500,600,900,1100)
times_cut <- cut(times, breaks = bounds,
                 # labels denotes the names of values
                 # here: value = upper bound of the class
                 labels = bounds[-1])
times_cut
```

```
# cut(times, breaks = bounds) # labels are the classes (a,b]

df_cut_diff <-
  tibble(upper_bound = times_cut) %>%
  count(upper_bound) %>%
  mutate(rel = n / length(times),
         cum_rel_freq = cumsum(rel))
df_cut_diff

# alternative solution using the emp. cum. dist. function H
tibble(obs.values = bounds[-1],      # -> upper bounds of the classes
       cum.rel.freq = H(obs.values)  # cum. rel. freq. of the classes
) %>%
  mutate(
    # for the smallest obs. value is cum.rel.freq = rel. freq
    # lag() -> value before the current value
    rel.freq = if_else(is.na(lag(cum.rel.freq)),cum.rel.freq,
                       cum.rel.freq - lag(cum.rel.freq)),
    abs.freq = rel.freq * length(times)  # length(times) = no of of obs.
  )


# tex Tabelle erzeugen
xtable(df_cut_tab) %>% print(include.rownames = FALSE, floating = FALSE)

# Histogram
hist(times, breaks = bounds, xlab = "time", ylab ="")
# histogram applying ggplot()
ggplot(data = df) +
  geom_histogram(mapping = aes(x = values), breaks = bounds,
                 color = "grey") +
  theme_classic() +
  ggtitle("Histogram of times - new bounds") +
  xlab("time") +
  ylab("")


# eps-file
dev.copy2eps(file="../pictures/time_hist2.eps")
```

Descriptive Statistics - Measures

1. Make up data sets with 5 numbers each that have:

   (a) the same mean but different standard deviations.

   (b) the same mean but different medians.

   (c) the same median but different means.

   Answer:

a)	A	B		A	B
	1	3	mean	5	5
	3	4	variation	10	2,5
	5	5	std. variation	3,16	1,58
	7	6			
	9	7	same mean but different variation		

b)	A	B		A	B
	1	1	mean	5	5
	3	3	median	5	6
	5	6			
	7	7			
	9	8	same mean but different median		

c)	A	B		A	B
	1	1	mean	5	6
	3	3	median	5	5
	5	5			
	7	7			
	9	14	same median but different mean		

```
#######################################################
# Descriptive Statistics: changing measures
#
# File: des_stat_different_measures.R
#
#######################################################
# Make up data sets with 5 numbers each that have:
# a) the same mean but different standard deviations.
xa <- c(1,3,5,7,9)
ya <- c(3,4,5,6,7)
mean(xa); mean(ya)
# 5       2
sd(xa); sd(ya)
# sd(xa)     sd(ya)
# 3.162278   1.581139

# b) the same mean but different medians.
xb <- c(1,3,5,7,9)
yb <- c(1,3,6,7,8)
mean(xb); mean(yb); median(xb); median(yb)
# 5         5         5           6

# c) the same median but different means.
xc <- c(1,3,5,7,9)
yc <- c(1,3,5,7,14)
mean(xc); mean(yc); median(xc); median(yc)
#  5        6          5          5
```

2. Consider a stock portfolio that began with a value of 1000 \$ and had annual returns of 13%, 22%, 12%, -5%, and -13%.

   (a) Compute the value after each of the five years.

   (b) Compute the annual rate of return.

   Use the **geometric mean**: $\sqrt[n]{\prod_{i=1}^{n} x_i}$

(c) Based on the result of b), which annual returns do you expect in the next two years? Would it make sense to prdeict the annual return 20 years later?

Answer:

value: 1000

year	annual return	rate	value	return with geo. mean	return with "mean"
1	13,00%	1,13	1130	1049,98	1058
2	22,00%	1,22	1378,6	1102,45	1119,36
3	12,00%	1,12	1544,03	1157,55	1184,29
4	-5,00%	0,95	1466,83	1215,40	1252,98
5	-13,00%	0,87	1276,14	1276,14	1325,65

geometric mean: $(1.13 \cdot 1.22 \cdot 1.12 \cdot 0.95 \cdot 0.87)^{1/5} \approx (1.276)^{1/5} \approx 1.049977111$
mean: 1,06
If we assume that the annual return in the following is close to the average annual return, we will predict

- return after year 6: $1276.142 \cdot 1.049977111 \approx 1339.92$
- return after year 7: $1276.142 \cdot 1.049977111^2 \approx 1406.886$

To assume that in following 20 years the annual return will be close to the average annual return based on the return on these 5 years is rather unrealisitic. Therefore a prediction of the return in 20 years makes no sense.

```
##########################################################
# Descriptive Statistics: Returns of a portfolio
#
# File: des_stat_returns.R
#
##########################################################
# Consider a stock portfolio that began with a value
# of 1000 and had annual returns of 13%, 22%, 12%, -5%,
# and -13%.
# a) Compute the value after each of the five years.
x <- 1000
ret <- c(0.13,0.22,0.12,-0.05,-0.13)
value <- 1000 * cumprod(1+ret)
value
# solution with a for loop
#value <- rep(0,5)
#for ( i in 1:5 ) {
#  x <- x*(1+ret[i])
#  value[i] <- x
#}
value
#  1130.000 1378.600 1544.032 1466.830 1276.142

# Compute the annual rate of return.
annual_rate <-  (prod(1+ret)^0.2 -1)*100
annual_rate

# wrong annual rate
mean(ret)
# value after 5 years using wrong.rate
1000*(1+mean(ret))**5
```

```
#  4.997711
# expected return after year 6
value[5]* (1+annual_rate/100)
# expected return after year 7
value[5]* (1+annual_rate/100)**2
```

3. A sample of 30 distance scores measured in yards has a mean of 7, a variance of 16, and a standard deviation of 4.

   (a) You want to convert all your distances from yards to feet, so you multiply each score in the sample by 3. What are the new mean, variance, and standard deviation?

   (b) You then decide that you only want to look at the distance past a certain point. Thus, after multiplying the original scores by 3, you decide to subtract 4 feet from each of the scores. Now what are the new mean, variance, and standard deviation?

Answer: Original data: $\begin{array}{rcl} n &=& 30 \\ \bar{x} &=& 7 \\ s^2 &=& 16 \end{array}$

   (a) Every observation is multiplied by 3: $x_i^{neu} = 3 \cdot x_i$.
   We become the mean:
   $\bar{x}^{neu} = \frac{1}{30}\sum_{i=1}^{30} x_i^{neu} = \frac{1}{30}\sum_{i=1}^{30} 3 \cdot x_i = 3 \cdot \bar{x} = 3 \cdot 7 = 21$
   and the variance:
   $s_{neu}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^{neu} - \bar{x}^{neu})^2 = \frac{1}{29}\sum_{i=1}^{30}(3 \cdot x_i - 3 \cdot \bar{x})^2 = \frac{1}{29}\sum_{i=1}^{30} 3^2 \cdot (x_i - \bar{x})^2 = 9 \cdot s^2 = 9 \cdot 16 = 144$ i.e. the new standard deviation is: $\sqrt{144} = 12$

   (b) After the multiplication, subtract 4:
   $x_i^{neu2} = x_i^{neu} - 4$ i.e. $\bar{x}_{neu2} = \frac{1}{30}\sum_{i=1}^{30} x_i^{neu2} = \frac{1}{30}\sum_{i=1}^{30}(x_i^{neu} - 4) = \bar{x}_{neu} - \frac{30 \cdot 4}{30} = \bar{x}_{neu} - 4 = 21 - 4 = 17$
   and the variance:
   $s_{neu2}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^{neu2} - \bar{x}^{neu2})^2 = \frac{1}{29}\sum_{i=1}^{30}(x_i^{neu} - 4 - (\bar{x}^{neu} - 4))^2 = s_{neu}^2 = 144$ i.e. the new standard deviation is: $\sqrt{144} = 12$

4. Which of the following measures of location can be used for a qualitative variable, a quantitative continuous variable resp. an ordinal variable?

   - Mode
   - Median
   - Mean

Answer: qualitative variable: only the mode
quantitative continuous variable: all
ordinal variable: mode, median

5. You have the following 25 observations of the variable Number. Calculate the arithmetic mean, the geometric mean, the harmonic mean and the trimmed 20% mean.

Number	Absolute frequency
1	5
2	4
3	1
4	7
5	2
6	3
7	1
8	2
Sum	25

Answer:

(a) $\bar{x} = \frac{1\cdot5+2\cdot4+3\cdot1+4\cdot7+5\cdot2+6\cdot3+7\cdot1+8\cdot2}{25} = \frac{95}{25} = 3.8$

(b) $G(x) = \sqrt[25]{1^5 \cdot 2^4 \cdot 3 \cdot 4^7 \cdot 5^2 \cdot 6^3 \cdot 7 \cdot 8^2} = \sqrt[25]{1902536294400} \approx 3.099$

(c) $H(x) = \frac{25}{\frac{5}{1}+\frac{4}{2}+\frac{1}{3}+\frac{7}{4}+\frac{2}{5}+\frac{3}{6}+\frac{1}{7}+\frac{2}{8}} = \frac{5250}{2179} \approx 2.409$

(d) removing the upper and lower 10% of the scores, i.e. the firts two ones and the last two eights results in
trimmend 20% mean $= \frac{1\cdot3+2\cdot4+3\cdot1+4\cdot7+5\cdot2+6\cdot3+7\cdot1}{21} = \frac{77}{21} \approx 3.667$

```
########################################################
# Descriptive Statistics: measures of a frequency table
#
# File: des_stat_freq_tab_measures.R
#
########################################################
library(tidyverse)

# frequency table
freq_tab <- tibble(
  no = 1:8,
  nobs = c(5,4,1,7,2,3,1,2)
)
freq_tab

# ordered raw data
x <- rep(freq_tab$no, freq_tab$nobs)

# mean
mean(x)
# geometric mean
prod(x)^(1/length(x))
# harmonic mean
length(x)/sum(1/x)
# trimmed 20% mean
mean(x, trim = 0.1)
```

6. Which of the following measures of dispersion can be used for a qualitative variable resp. a quantitative continuous variable?

   - Variance
   - Standard deviation
   - Interquartile range
   - Range

   Answer:

   - qualtitative variable, i.e. a nominal scaled variable, none
   - ordinal variable: range and interquartile range
   - quantitative continuous variable: all

7. Define for each of the measures mean, quantile, variance, geometric mean, harmonic mean and trimmed mean based on their definitions given in the lecture a R function.

   (a) Use the sample $x_i : 3, 7, 2, 5, 6, 10, 6, 3, 6, 5$ to test the functions. Calculate the 3 quartile and the 10% trimmed mean for the given sample.

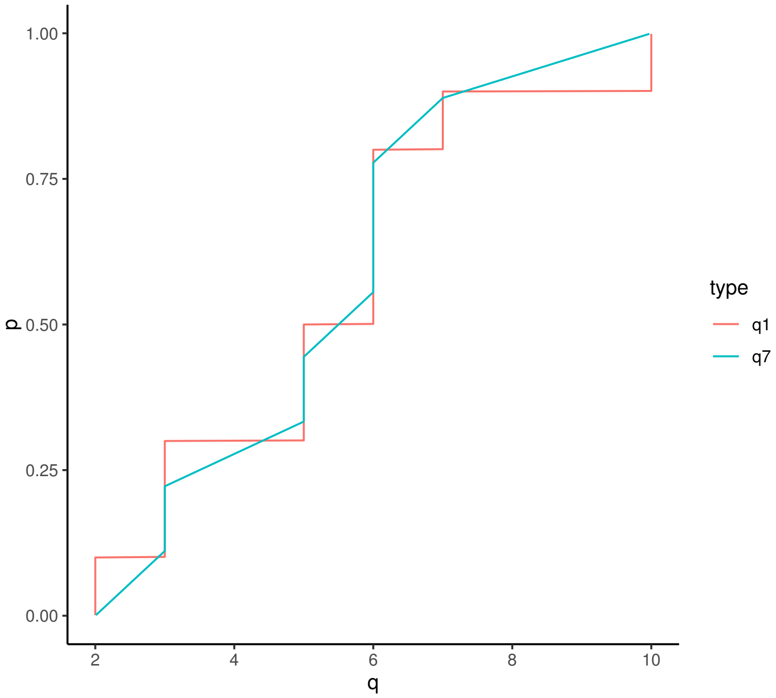
   (b) In R there are several methods offered to compute quantiles. These methods are specified by the argument type $\in 1, 2, ..., 9$. Type 1 correspons to the definition given in the lecture. Type 7 is the default value. Evaluate the p-quantiles of the sample for $p \in \{0.001, 0.002, ..., 0.999\}$ and create a diagram with x-axis = quantile and y-axis = p containing the values of both types. Mention the differences and try to understand the different approaches.

   Answer: sample: $\{x_1, x_2, ..., x_n\} = \{3, 7, 2, 5, 6, 10, 6, 3, 6, 5\}$

measure	formula	value
mean(x)	$\frac{\sum_{i=1}^{n} x_i}{n}$	5.3
quantile(x,p)	$x_{(\lceil n*p \rceil)}$	3, 5, 6
variance(x)	$\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$	5.344444
geometric mean (x)	$\sqrt[n]{\prod_{i=1}^{n} x_i}$	4.822547
harmonic mean (x)	$\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$	4.329897
trimmed mean (x)	$\frac{\sum_{i=1+\lfloor n\cdot 0.5\cdot p \rfloor}^{n-\lfloor n\cdot 0.5\cdot p \rfloor} x_{(i)}}{n - 2\cdot \lfloor n \cdot 0.5 \cdot p \rfloor}$	5.125

The calculation of the quantile is extensively described in R manual quantile().

- type=1: $\tilde{x}_p = \begin{cases} x_{(np)} & np \in \mathbb{N} \\ x_{(\lceil np \rceil)} & \text{else} \end{cases}$
  Thus the calculation corresponds to the definition according to the lecture "inverse of the empirical distribution function".
- type=7: Consider the points $(\frac{k-1}{n-1}, x_{(k)})$ for k=1,2,...,n with $x_{(k)}$ the k-th value of the ordered sample. Let f() be the function defined by linear interpolating these points. Then the p quantile is f(p). So we get for $p = \frac{k}{n-1}, k = 0, 1, ..., n-2$ the values $x_{(k+1)}$ of the ordered sample and for the other values of p we get $x_{(p)}$ the linear interpolation of p between the points $(\frac{k}{n-1}, x_{(k+1)})$, $(\frac{k+1}{n-1}, x_{(k+2)})$ if $\frac{k}{n-1} < p < \frac{k+1}{n-1}$ for k=0,1,..., n-2. If $x_{(k)} = x_{(k+1)}$ you see a vertical line in the diagram.



8. An experiment compared the ability of three groups of participants to remember briefly-presented chess positions. The data are shown below. The numbers represent the number of pieces correctly remembered from three chess positions.

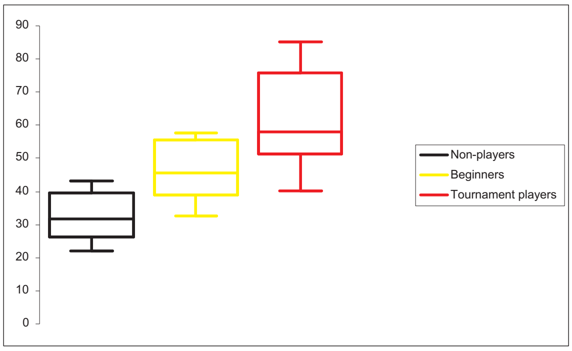
   Compare the performance for each group by computing mean, median, min, max, quartiles, interquartile range, variance. Create side-by-side box plots for these three groups. What can you say about the differences between these groups from the box plots?

Non-players	Beginners	Tournament Players
22.1	32.5	40.1
22.3	37.1	45.6
26.2	39.1	51.2
29.6	40.5	56.4
31.7	45.5	58.1
33.5	51.3	71.1
38.9	52.6	74.9
39.7	55.7	75.9
43.2	55.9	80.3
43.2	57.7	85.3

   Answer: For non-player ($n = 10$):

- Minimum: 22.1
- Maximum: 43.2
- Mean: $\frac{1}{10}\sum_{i=1}^{10}(22.1 + 22.3 + \ldots + 43.2 + 43.2) = 33.04$
- Variance:
$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{9}\left((22.1 - 33.04)^2 + (22.3 - 33.04)^2 + \ldots + (43.2 - 33.04)^2\right) =$ 64.53
- Median: $x_{\left(\frac{n}{2}\right)} = x_{(5)} = 31.7$ or $\frac{1}{2}\left(x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}\right) = \frac{1}{2}(31.7 + 33.5) = 32.6$
- Q1: $n \cdot p = 10 \cdot 0.25 = 2.5$ and 2.5 rounded upwards 3, i.e. we obtain $x_{(3)} = 26.2$
- Q2= Median
- Q3: $n \cdot p = 10 \cdot 0.75 = 7.5$ and 7.5 rounded upwards is 8, i.e. we obtain $x_{(8)} = 39.7$
- IQ= Q3-Q1=39.7-26.2=13.5

min	22,1	32,5	40,1
max	43,2	57,7	85,3
Q1	26,2	39,1	51,2
Q2	31,7	45,5	58,1
Q3	39,7	55,7	75,9
mean	33,04	46,79	63,89
interquartil range	13,5	16,6	24,7
variance	64,53	81,55	244,03




```
##########################################################
# Descriptive Statistics: Chess Players
#
# File: des_stat_chess.R
#
###########################################################
# An experiment compared the ability of three groups of
# participants to remember briefly-presented chess
# positions. The data are shown below. The numbers
# represent the number of pieces correctly remembered
```

```
# from three chess positions.
# Compare the performance for each group by computing
# mean, median, min, max, quartils, interquartil range,
# variance. Create side-by-side box plots for these
# three groups. What can you say about the differences
# between these groups from the box plots?

############################################################
# old fashion solution
############################################################
data <- matrix(c(
  22.1,32.5,40.1,
  22.3,37.1,45.6,
  26.2,39.1,51.2,
  29.6,40.5,56.4,
  31.7,45.5,58.1,
  33.5,51.3,71.1,
  38.9,52.6,74.9,
  39.7,55.7,75.9,
  43.2,55.9,80.3,
  43.2,57.7,85.3), nrow=10, ncol=3, byrow=TRUE)
colnames(data) <- c("Non-players","Beginners","Tournament")

char_numbers <- rbind(
  apply(data,2,min),
  apply(data,2,max),
  c(quantile(data[,1],probs=c(0.25),type=1),
    quantile(data[,2],probs=c(0.25),type=1),
    quantile(data[,3],probs=c(0.25),type=1)),
  c(quantile(data[,1],probs=c(0.5),type=1),
    quantile(data[,2],probs=c(0.5),type=1),
    quantile(data[,3],probs=c(0.5),type=1)),
  c(quantile(data[,1],probs=c(0.75),type=1),
    quantile(data[,2],probs=c(0.75),type=1),
    quantile(data[,3],probs=c(0.75),type=1)),
  apply(data,2,mean),
  apply(data,2,var))
char_numbers <- rbind(char_numbers, char_numbers[5,] - char_numbers[3,])
rownames(char_numbers) <-c("min","max","q1","q2","q3","mean","variance",
                           "interquartil range")
char_numbers

# Boxplots
boxplot(data[,1],data[,2],data[,3], names=colnames(data),
        main = "side by side boxplots",
        xlab = "player type", ylab = "rem. chess positions")

############################################################
# Solution with tibbles and ggplot()
############################################################
library(tidyverse)
data <- tibble(
    type = c(rep("non-player",10), rep("beginner",10),rep("tournament",10)),
    res = c(22.1,22.3,26.2,29.6,31.7,33.5,38.9,39.7,43.2,43.2,
            32.5,37.1,39.1,40.5,45.5,51.3,52.6,55.7,55.9,57.7,
            40.1,45.6,51.2,56.4,58.1,71.1,74.9,75.9,80.3,85.3))
measures <- data %>%
  group_by(type) %>%
  summarise(Min = min(res),Mx=max(res),
            q1=quantile(res,0.25,type=1),q2=quantile(res,0.5,type=1),
            q3=quantile(res,0.75,type=1),
            Mean=mean(res),variance=var(res),
            interquartile_range=q3-q1)

measures

# Boxplots
# changing the order in the side by side boxplots by adding a factor to type
data$type <- factor(data$type, levels = c("non-player", "beginner","tournament"))
ggplot(data = data) +
  geom_boxplot(mapping = aes(x=type, y=res, group = type)) +
  geom_point(mapping = aes(x=type,y=res,group=type)) +
  xlab("player type") +
  ylab("rem. chess positions") +
  ggtitle("side by side boxplots with marked values") +
  theme_bw()

# eps-file
dev.copy2eps(file="../pictures/chess_bp.eps")
```

9. Exercise 3.1 from Heumann, Schomaker: Introduction to Statistics and

Data Analysis, page 63
A hiking entusiast has a app for his smartphone which summarizes his hikes by using a GPS device. The distance hiked (in km) and maximum altitude (in m) for the last 10 hikes:

Distance	12.5	29.9	14.8	18.7	7.6	16.2	16.5	27.4	12.1	17.5
Altitude	342	1245	502	555	398	670	796	912	238	466

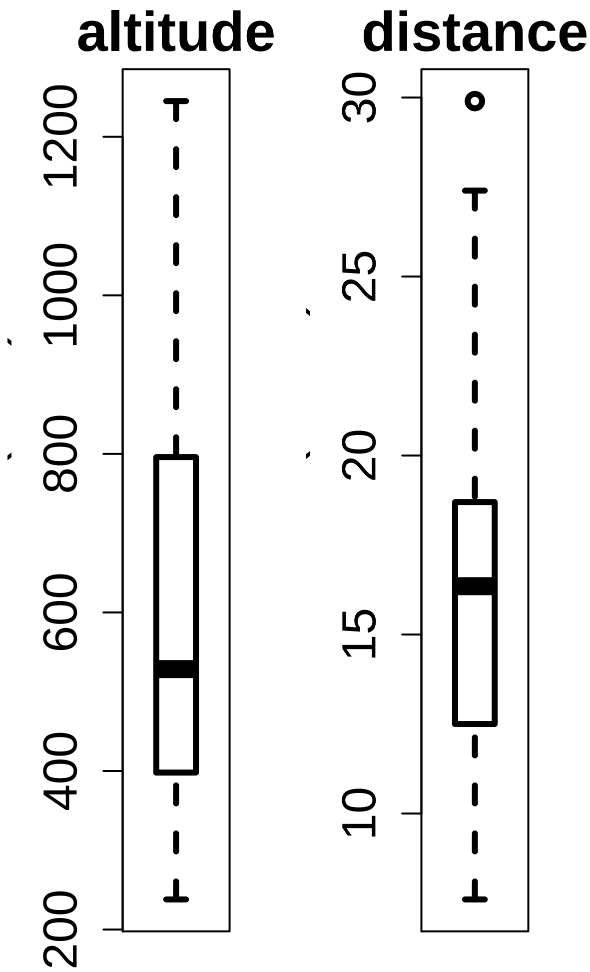
(a) Calculate the arithmetic mean and median for both distance and altitude.
(b) Determine the first and third quartile for both distance and altitude. Discuss the shape of the distribution given the values in a) and b).
(c) Calculate the interquartile range and standard deviation for both variables. Compare the variability of both variables.
(d) Draw the box plot for both distance and altitude.
(e) Assume distance is measured as only short (5-15 km), moderate (15-20 km) and long (20-30 km). Summarize the grouped data in a frequency table. Calculate the weighted arithmetic mean under the assumption that the raw data is not known.

Answer:

(a) mean(distance) = 17.32, mean(altitude) = 612.4
median(distance) = 16.2 resp. 16.35, median(altitude) = 502 resp. 528.5
(b) Q3(distance) = 18.7 , Q1(distance) = 12.5
Q3(altitude) = 796, Q1(altitude) = 398
Since the median of the distance is closer to Q3 it seems to be that the the distributions is skewed to the left. The median of the altitude is closer to Q1. This might indicate that the distribution is right skewed.
(c) Interquartile Range = Q3 -Q1
distance: 18.7-12.5 = 6.2, altitude = 796 - 398 = 398
Variances: altitude = 91460.49, distance = 46.11511
Since the means of both variables are rather different, we use the **coefficient of variation v** which is defined as $v = \frac{s}{\bar{x}}$. For the variables we get

$$v_{distance} = 0.3920791 < v_{altitude} = 0.493847$$

Thus the variability of distance seems to be lower than of altitude.



(d) Boxplots:

(e) Frequency table of the gouped data

values	n	rel	cum_rel_freq
(5,15]	4	0.40	0.40
(15,20]	4	0.40	0.80
(20,30]	2	0.20	1.00

The weighted arithmetic mean is calculated by using the relative frequencies and the middle of the classes:

$$\tilde{d} = 0.4 \cdot 10 + 0.4 \cdot 17.5 + 0.2 \cdot 25 = 16$$

```
##########################################################
# Descriptive Statistics: Exercise 3.1,
# Heumann, Schomaker, page 63
```

```
#
# File: des_stat_hiking.R
#
########################################################
# load tidyverse and xtable
library(tidyverse)
library(xtable)

# generate the data
distance <- c(12.5,29.9,14.8,18.7,7.6,16.2,16.5,27.4,12.1,17.5)
altitude <- c(342,1245,502,555,398,670,796,912,238,466)
# sorted data
sort(distance)
sort(altitude)

# mean and median
mean(distance)
mean(altitude)

# R offers several ways of calculating quantiles. Use type=1
# to apply the method we have introduced.
quantile(distance,probs = c(0.25,0.5,0.75),type=1)
quantile(altitude,probs = c(0.25,0.5,0.75),type=1)

# interquartial range
quantile(distance,probs=0.75,type=1)- quantile(distance,probs=0.25,type=1)
quantile(altitude,probs=0.75,type=1)- quantile(altitude,probs=0.25,type=1)

# variance
var(altitude)
var(distance)

# coefficients of variation
sd(distance)/mean(distance)
sd(altitude)/mean(altitude)

# boxplots
par(mfrow=c(1,2))
boxplot(altitude,xlab="",ylab="Altitude (in m)",
        main = "altitude",
        cex.axis=1.5,lwd=3,cex.lab=1.75,cex.main=1.75)
boxplot(distance,xlab="",ylab="Distance (in km)",
        main = "distance",
        cex.axis=1.5,lwd=3,cex.lab=1.75,cex.main=1.75)

# eps-file
dev.copy2eps(file="../pictures/hiking_bp.eps")

# boxplots with ggplot
ggplot(data = tibble(d=distance)) +
  geom_boxplot(mapping = aes(x=d)) +
  geom_point(mapping = aes(x=d,y=0)) +
  scale_y_continuous(labels = NULL, breaks = NULL) +
  ylab("") +
  xlab("distance (in km)") +
  coord_flip() +
  ggtitle("distance") +
  theme_bw()
ggplot(data = tibble(a=altitude)) +
  geom_boxplot(mapping = aes(x=a)) +
  geom_point(mapping = aes(x=a,y=0)) +
  scale_y_continuous(labels = NULL, breaks = NULL) +
  ylab("") +
  xlab("altitude (in m)") +
  coord_flip() +
  ggtitle("altitude") +
  theme_bw()


# grouped data
bounds <- c(5,15,20,30)
dist_cut <- cut(distance, breaks = bounds)

df_cut <- tibble(values = dist_cut)
df_cut_tab <-
  df_cut %>%
  count(values) %>%
  mutate(rel = n / length(distance),
         cum_rel_freq = cumsum(rel))
df_cut_tab
# tex table
tex_tab <- xtable(df_cut_tab)
print(tex_tab, include.rownames = FALSE, floating = FALSE)
```

```
midpoints.classes <- (bounds[-1]+bounds[-4])/2
mean.grouped <- sum(midpoints.classes * df_cut_tab$rel)
mean.grouped
# weighted.mean(c(10,17.5,25),c(4/10,4/10,2/10))
```

10. The data set mpg of the ggplot package contains a subset of the fuel economy data that the EPA makes available on http://fueleconomy.gov. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

    (a) Inspect the description of the data set using the ?mpg() command.

    (b) Select only the variables displ (engine displacement) and hwy (highway miles per gallon) from the data set. Group the values of the variable displ into the the groups "low" ($1 \leq$ displ <3), "medium" ($3 \leq$ displ <5) and "big" ($5 \leq$ displ <8). Use the cut() command to do this. Add a column displ_class which denotes the belonging to one of the groups.

    (c) Calculate the mean, minimum, maximum and the three quartile of the variable hwy depending on the values of displ and depending on displ_class.

    (d) Draw boxplots of the variable hwy grouped by displ resp. displ_class and interpret the results.

    Answer:

    (b) Selected subset of the data

displ	hwy	displ_class
1.80	29	small
1.80	29	small
2.00	31	small
2.00	30	small
2.80	26	small
2.80	26	small
3.10	27	medium
1.80	26	small
1.80	25	small
2.00	28	small
2.00	27	small
2.80	25	small
2.80	25	small
3.10	25	medium
3.10	25	medium
...	..	...

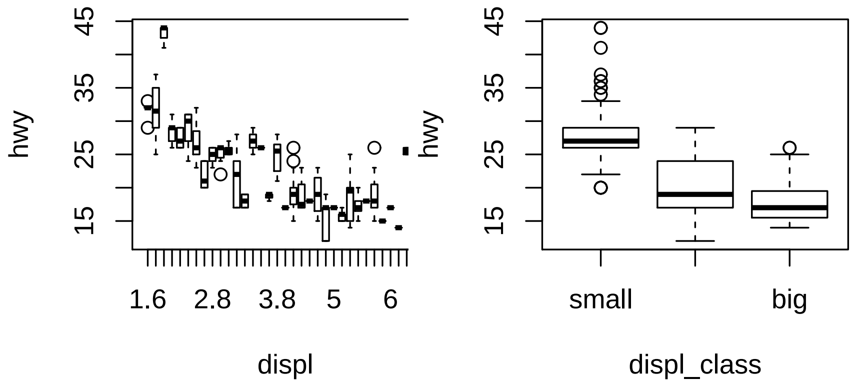
(c) Characteristic numbers of hwy grouped by displ

displ	mean(hwy)	min(hwy)	max(hwy)	q1	q2	q3
1.60	31.60	29.00	33.00	32.00	32.00	32.00
1.80	31.64	25.00	37.00	29.00	31.50	35.00
1.90	43.00	41.00	44.00	41.00	44.00	44.00
2.00	28.24	26.00	31.00	27.00	29.00	29.00
2.20	27.33	26.00	29.00	26.00	27.00	29.00
2.40	28.85	24.00	31.00	27.00	30.00	31.00
2.50	26.80	23.00	32.00	25.00	26.00	28.50
2.70	21.75	20.00	24.00	20.00	21.00	24.00
2.80	24.90	23.00	26.00	24.00	25.00	26.00
3.00	25.12	22.00	26.00	24.50	26.00	26.00
3.10	25.67	25.00	27.00	25.00	25.50	26.00
3.30	22.00	17.00	28.00	17.00	22.00	24.00
3.40	18.00	17.00	19.00	17.00	18.00	19.00
3.50	27.00	25.00	29.00	26.00	27.00	28.00
...	...	...	...	...	...	...

Characteristic numbers of hwy grouped by displ_class

displ_class	mean(hwy)	min(hwy)	max(hwy)	q1	q2	q3
small	27.98	20.00	44.00	26.00	27	29.00
medium	20.11	12.00	29.00	17.00	19	24.00
big	18.14	14.00	26.00	15.50	17	19.50

(d) Boxplots



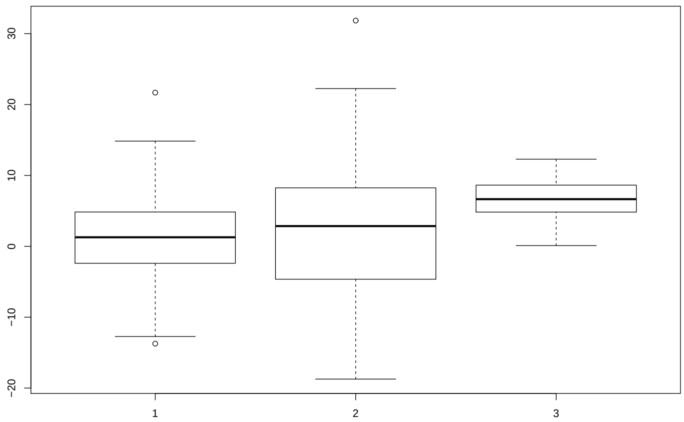
Grouping the data the association between engine displacement and highway miles per gallon becomes more: higher engine displacement correlates with low highway miles per gallon.

```
##########################################################
# Descriptive Statistics: miles per gallon
#
# File: des_stat_miles_gallon.R
#
###########################################################
# load tidyverse
library(tidyverse)

# inspect the description of the data set
?mpg()

# select only the variables displ and hwy and add
# a colummn displ_class which denotes the belonging
# to one of the groups
# low (1 <= displ < 3), medium (3 <= displ < 5),
# big (5 <= displ < 8)
tab <-
  mpg %>%
  select(displ,hwy) %>%
  mutate(displ_class =
              cut(displ,breaks = c(1,3,5,8),
                  labels = c("small","medium","big"))
  )
tab
# tex table
tex_tab <- xtable(tab)
print(tex_tab, include.rownames = FALSE, floating = FALSE)

# calculate mean, min, max Q1, Q2 and Q3 of the variable
# hwy grouped by the values of displ.
stat_hwy_displ <-
  tab %>%
  group_by(displ) %>%
  summarise(mean=mean(hwy),min=min(hwy),max=max(hwy),
               q1=quantile(hwy,0.25, type=2),
               q2=quantile(hwy,0.5, type=2),
               q3=quantile(hwy,0.75, type=2)
               )
stat_hwy_displ
# tex table
tex_tab <- xtable(stat_hwy_displ)
print(tex_tab, include.rownames = FALSE, floating = FALSE)

# calculate mean, min, max Q1, Q2 and Q3 of the variable
# hwy grouped by the values of displ_class.
```

```
stat_hwy_class <-
  tab %>%
  group_by(displ_class) %>%
  summarise(mean(hwy),min(hwy),max(hwy),
            q1=quantile(hwy,0.25, type=2),
            q2=quantile(hwy,0.5, type=2),
            q3=quantile(hwy,0.75, type=2)
  )
stat_hwy_class
# tex table
tex_tab <- xtable(stat_hwy_class)
print(tex_tab, include.rownames = FALSE, floating = FALSE)

# boxplots of hwy grouped by displ
par(mfrow = c(1,1))
boxplot(hwy ~ displ, data = tab, xlab = "displ", ylab = "hwy")
# using ggplot
ggplot(data = tab) +
  geom_boxplot(mapping = aes(group = displ, x = displ, y = hwy)) +
  theme_bw()
# eps files
dev.copy2eps(file = "../pictures/miles_gallon1.eps")

# boxplots of hwy grouped by displ_class
boxplot(hwy ~ displ_class, data = tab, xlab = "displ_class", ylab = "hwy")
# using ggplot
ggplot(data = tab) +
  geom_boxplot(mapping = aes(group = displ_class, x = displ_class, y = hwy)) +
  geom_point(mapping = aes(group = displ_class, x = displ_class, y = hwy)) +
  theme_bw()
# eps files
dev.copy2eps(file = "../pictures/miles_gallon2.eps")

# both boxplots together
par(mfrow = c(1,2))
boxplot(hwy ~ displ, data = tab, xlab = "displ", ylab = "hwy")
boxplot(hwy ~ displ_class, data = tab, xlab = "displ_class", ylab = "hwy")
 # zurücksetzen von mfrow
par(mfrow = c(1,1))
```

Descriptive Statistics - Shape

1. Use the following boxplots to answer the questions below:



   (a) Which of the three distributions has the highest measure of location?

   (b) Which of the three distributions has the largest range?

   (c) Which of the three distributions has the largest interquartile range?

(d) Which of the three distributions has the highest maximum value?

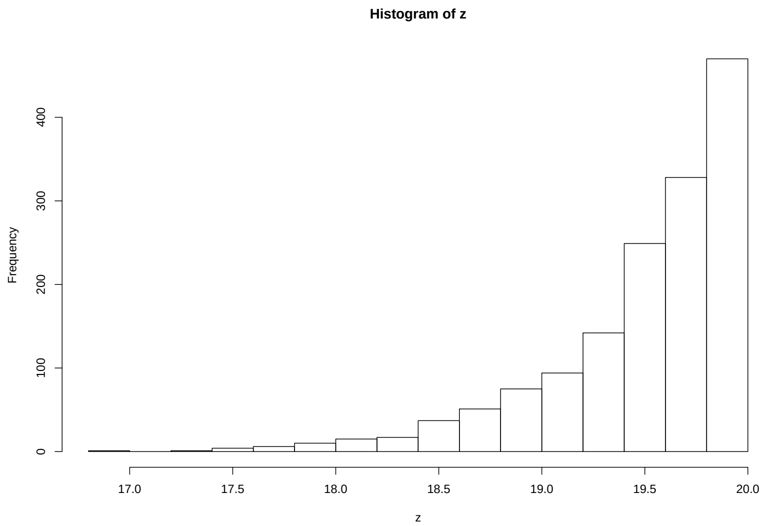
(e) Which of the three distributions has the smallest maximum value?

(f) Discuss skewness/symmetry of the three distributions.

Motivate your answers!

Answer: a) 3, b) 2, c) 2, d) 2, e) 3, f) all symmetric

2. Use the following histogram to answer the questions below:



Is the distribution left-skewed, right-skewed or symmetric? Motivate your answers!

Answer: left-skewed