Course of Study Bachelor Computer Science	Exercises Statistics WS 2023/24
Sheet III - Solutions	

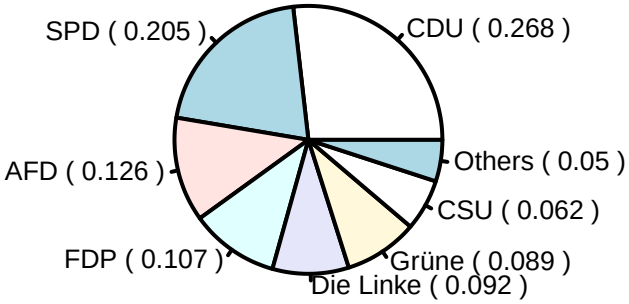
Descriptive Statistics - Frequency Tables and Distributions

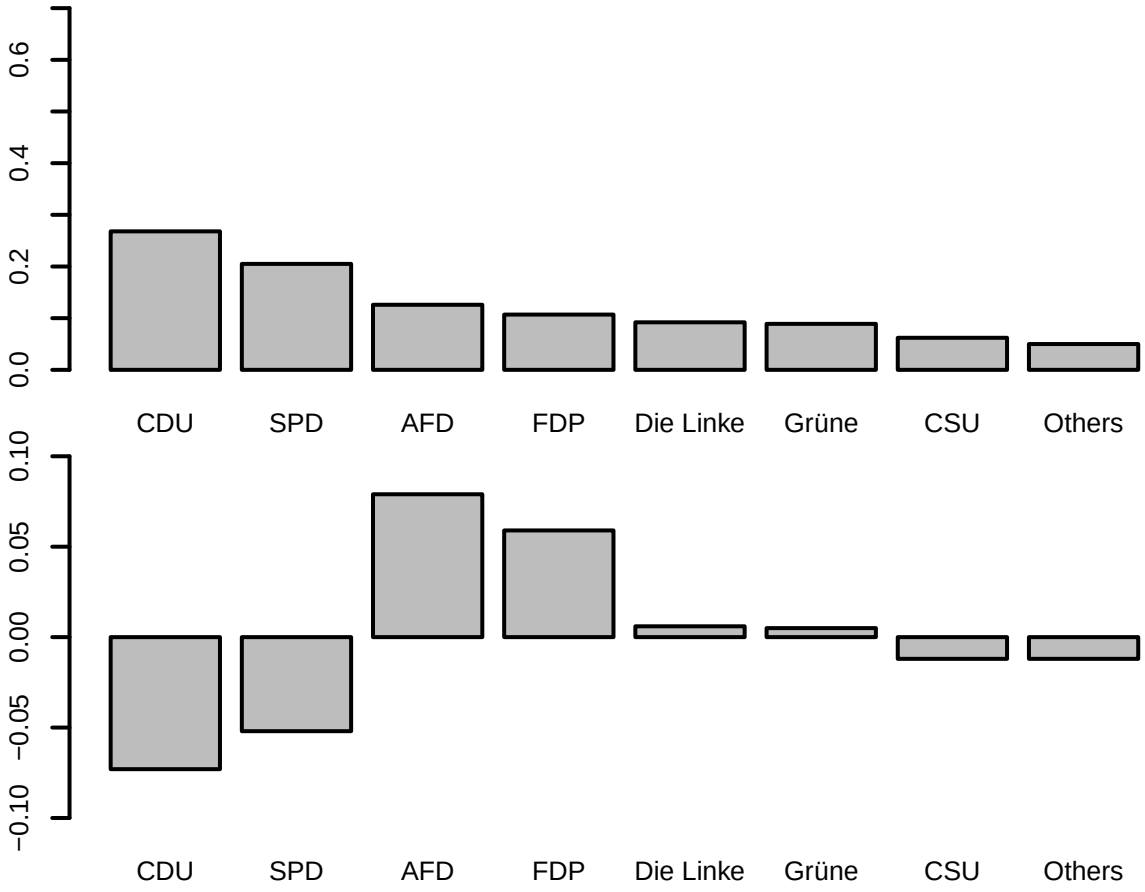
1. Consider the results of the national elections in Germany in 2013 and 2017:

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
# Solution
#
# File: des_stat_nat_el_sol.R
#
#######################################################
library(tidyverse)

# Results of national elections
```

```
results2013 <- c(0.268,0.205,0.126,0.107,0.092,
                 0.089,0.062,0.05)
results2017 <- 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")

# Results of national elections applying tibbles
nat_el <- tibble(
  res.2013 = c(0.268,0.205,0.126,0.107,0.092,0.089,0.062,0.05),
  res.2017 = 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"),
  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")

# reset to the default values of par()
dev.off()

# 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 $(500, 600]$, $(600, 700]$, $(700, 800]$, $(800, 900]$, $(900, 1000]$, $(1000, 1100]$.

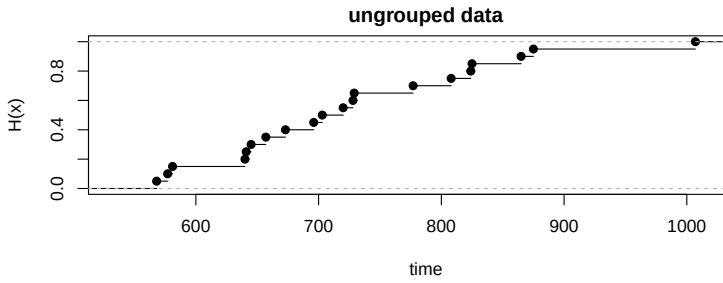
- Compute the grouped frequency distribution and draw the histogram and the distribution function.

(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? Draw the distribution function, too.

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

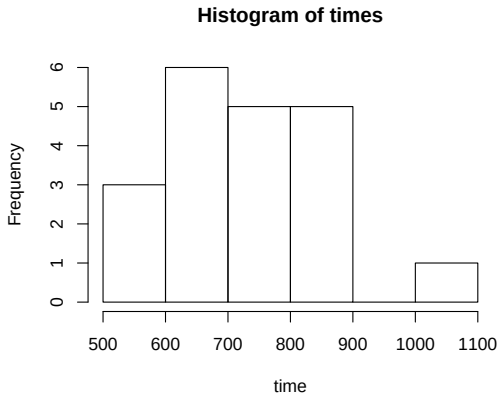


(b) Compute the proportion of response times

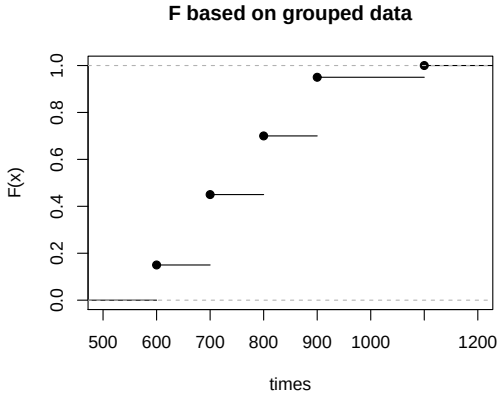
- less equal 800: H(800) = 0.7
- greater than 725: 1-H(725) = 0.45
- greater than 642 and less equal 777: H(777) - H(642) = 0.45
- equal 696: $H(696) - lim_{x \uparrow 696} H(x) = 0.05$
- in [696,800]

(c) Grouped data

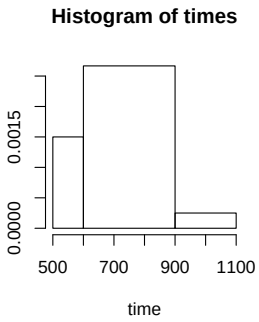
- Cumlative frequncy distribution based on grouped data



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
# Solution
#
# File: des_stat_time_mouse_sol.R
#
#########################################################
library(tidyverse)

# 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                                        #
#########################################################
# solution applying count()
df <- 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)
df

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

# emp. Verteilungsfkt.
plot.ecdf(times,
          xlab = "time", ylab = "H(x)",
          main = "ungrouped data")

# plot the empirical distribution function with ggplot()
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") +
  theme_classic()

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

# solution applying count()
df_cut <-
  tibble(upper_bound =
           # convert factor times cut to numeric
           # necessary for ggplot() line 120 ff
           times_cut %>% as.character() %>% as.numeric()) %>%
  count(upper_bound) %>%
  mutate(rel = n / length(times_cut),
         cum.rel.freq = cumsum(rel))

# 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")

# plot the distrubtion function
# 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")

# plot the distribution function with ggplot()
ggplot(data =
         df_cut %>%
         mutate(x1 = upper_bound, x2 = c(upper_bound[-1],100+max(upper_bound)))) +
  geom_point(mapping = aes(x=upper_bound, 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_cut$upper_bound)-100) +
  xlab("time") +
  ylab("H(x)") +
  ggtitle("empirical distribution function based on classes") +
  theme_classic()

##########################################################
# 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 =
           # convert factor times cut to numeric
           times_cut %>% as.character() %>% as.numeric()) %>%
  count(upper_bound) %>%
  mutate(rel = n / length(times),
         cum.rel.freq = cumsum(rel))
df_cut_diff

# 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("")
```

```
# Since the class widths are not equal, there is non meaning full interpretation
# of the y-values.

# plot the distribution function with ggplot()
ggplot(data =
         df_cut_diff %>%
         mutate(x1 = upper_bound, x2 = c(upper_bound[-1],100+max(upper_bound)))) +
  geom_point(mapping = aes(x=upper_bound, 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_cut$upper_bound)-100) +
  xlab("time") +
  ylab("H(x)") +
  ggtitle("empirical distribution function based on classes with different width") +
  theme_classic()
```

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
# Solution
#
```

```
# File: des_stat_different_measures_sol.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
# Solution
#
# File: des_stat_returns_sol.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 <- x * cumprod(1+ret)
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. Given are the observations *4, 3, 2, 4, 10*. Calculate the

   (a) mean

   (b) median

   (c) mode

   (d) 20%-quantile

   (e) trimmed 40% mean

   of the data! (Include the appropriate formulas in your solution.)

   Answer:

   (a) Mean: $\bar{x} = \frac{2+3+4+4+10}{5} = 4.6$

   (b) Sort:
   2, 3, 4, 4, 10
   $n = 5 \Rightarrow x_{\left(\frac{n+1}{2}\right)} = 3$
   $Median: x_{(3)} = 4$

   (c) Mode: 4 (the most occuring value).

(d) $n \cdot p = 5 \cdot 0.2 = 1 \Rightarrow x_{(1)} = 2$

(e) removing the upper and lower 20% of the scores: 3, 4, 4
trimmed 40% mean $= (3 + 4 + 4)/3 = 11/3$

```
#########################################################
# Descriptive Statistics: Different measure of sample
# Solution
#
# File: des_stat_measures_sample_sol.R
#
#########################################################

library(tidyverse)

# Consider the observations
x <- c(4,3,2,4,10)

# a) mean
mean(x)

# b) median
x.ordered <- sort(x)  # ordered sample
x.ordered[ceiling((length(x))/2)]

# mention that median() function does not always get the same result; for
# example median(c(x,1)) is 3.5 but according to our definition it is 3
# quantile() with type=1 is equivalent to our definition
quantile(x, p=0.5, type=1)

# c) mode
x %>% tibble(val = x) %>%
  # find the absolute frequencies of the values
  group_by(val) %>%
  mutate(n = n()) %>%
  ungroup() %>%
  # find the entries with max. abs. frequency
  filter(n == max(n)) %>%
  # select only value
  select(val) %>%
  # remove duplicate entries
  unique()

# 20% quantile
x.ordered[ceiling(0.2* length(x))]
# alternative
quantile(x, p=0.2, type=1)

# trimmed 40% mean
# mean(x, trim = 0, na.rm = FALSE, ...)
# x     R object
# trim the fraction (0 to 0.5) of observations to be trimmed from each end of x
# before the mean is computed. Values of trim outside that range are taken as
# the nearest endpoint.
mean(x,trim = 0.2)
# alternative
x.ordered[floor(0.2*length(x)+1):ceiling(0.8*length(x))] %>%
  mean()
```

4. 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: Arithmetic mean, the geometric
# mean, the harmonic mean and the trimmed 20% mean from
# a frequency table
# Solution
#
# File: des_stat_freq_tab_measures_sol.R
#
#########################################################

library(tidyverse)

# Consider the frequency table
f.tab <- tibble(
  obs = 1:8,
  n = c(5,4,1,7,2,3,1,2)
)
f.tab
# convert to an ordered sample
x.ordered <- rep(f.tab$obs,f.tab$n)
x.ordered

# a) mean
mean(x.ordered)

# b) geometric mean
(f.tab$obs**f.tab$n %>% prod())**(1/sum(f.tab$n))
prod(x.ordered)^(1/25)

# c) harmonic mean
sum(f.tab$n)/sum(f.tab$n/f.tab$obs)
length(x.ordered)/ ((1/x.ordered) %>% sum())

# d) trimmed 20% mean
# mean(x, trim = 0, na.rm = FALSE, ...)
# x     R object
# trim the fraction (0 to 0.5) of observations to be trimmed from each end of x
# before the mean is computed. Values of trim outside that range are taken as
# the nearest endpoint.
mean(x.ordered,trim = 0.1)
```

5. Evaluation of Quantiles

   (a) Generate a random sample of size n from 1, 2, ..., 20 and determine the empirical distribution function .

   (b) Determine a R function to find quantile according to the definition given in the lecture.

   (c) Compare the results of your quantile function with the results of the R function quantile(), type=1.

   (d) The R function quantile() evaluates quantiles of type 7 if no type is specified. Type 7 quantile are defined by a linear interpolation of the points

$$(0, x_{(1)}), (\frac{1}{n-1}, x_{(2)}), (\frac{2}{n-1}, x_{(3)}), ..., (\frac{n-1}{n-1}, x_{(n)}))$$

   with n = sample size. Visualize the evalution by a diagram which contain these points, the linear interpolation and the R quantile of type 7 of order 0, 0.5, ..., 1.

   (e) Create a table containing the quantiles of type 1 and type 7 of order 0,0.01, ..., 0.99,1. What are the possible values of the quantiles?

   (f) Create a diagram which visualize the empirical distribution function, the function H, which connects the points

$$(0, x_{(1)}), (\frac{1}{n-1}, x_{(2)}), (\frac{2}{n-1}, x_{(3)}), ..., (\frac{n-1}{n-1}, x_{(n)}))$$

   with line segments (the above linear interpolation), type 1 and type 7 quantiles and mention the difference between type 1 and type 7 quantiles.

   (g) Increase the sample size from 10 to 50 and then to 100 and create the above diagram. What happens?

```
#########################################################
# Descriptive Statistics: Quantiles of type 1 and
# type 7 in R
#
# File: des_stat_quantiles_sol.R
#
#########################################################
library(tidyverse)

# a) Generate a random sample of size n from 1, 2, ..., 20
# and determine the empirical distribution function
s.size <- 10
x <- sample(1:20, size = s.size, replace = TRUE)
emp.dist <- tibble(
  obs = x
  ) %>%
  count(obs) %>%
```

```
  mutate(
    cum.rel.freq = cumsum(n)/sum(n)
  )
emp.dist
plot.ecdf(x)

# b) Determine a R function to find quantile according
# to the definition given in the lecture
my.quantile <- function(x,p) {
  x_sort <- sort(x)
  return(x_sort[ceiling(length(x)*p)])
}

# c) Compare the results of your quantile function with
# the results of the R function
my.quantile(x, seq(0,1, by=0.05))
quantile(x, probs = seq(0,1, by=0.05), type = 1)

# d) The R function quantile() evaluates quantiles of type 7
# if no type is specified. Type 7 quantile are defined by a linear
# interpolation of the points
# (0, x_(1)), (1/(n-1),x_(2)), (2/(n-1),x_(3)), ..., (n/(n-1),x_(n))
# with n = sample size. Visualize the evalution by a diagram
# which contain these points, the linear interpolation and the
# R quantile of type 7 of order 0, 0.5, ..., 1

# generate a tibble containing the points
# (0, x_(1)), (1/(n-1),x_(2)), (2/(n-1),x_(3)), ..., (n/(n-1),x_(n)) for type 7
# and
# (1, x_(1)), (1/n,x_(2)), (2/n,x_(3)), ..., (n/n,x_(n))
q.tab <- tibble(
  obs = x,
  obs.ordered = sort(x),
  prop.n = (1:s.size)/s.size,
  prop.n.minus.1 = (0:(s.size-1))/(s.size-1)
)
q.tab
# visualisiation the quantile function type 1 and type 7
# mention, that type 7 apply linear interpolations between the points
# (h1/(n-1),x_(h1)), (h2/(n-1),x_(h2)) for h1/(n-1) < p <= h2/(n-1)
plot(x=q.tab$prop.n.minus.1, y=q.tab$obs.ordered,
     type="l", col = "black",
     ylim=c(0,21), xlim=c(-0.1,1.1),
     xlab="p", ylab="p quantile",
     main="quantiles type=7 and type=1",
     sub = "black = type 7, blue = type 1")
segments(x0=c(0,q.tab$prop.n[-s.size]), y0=q.tab$obs.ordered[ceiling(q.tab$prop.n*s.size)],
         x1=q.tab$prop.n, y1=q.tab$obs.ordered[ceiling(q.tab$prop.n*s.size)], col="blue")
axis(2, at = seq(0,20, by = 1))
axis(1, at = seq(0,1, by = 0.1))

# e) Create a table containing the quantiles of type 1 and type 7 of order
# 0,0.01, ..., 0.99,1. What are the possible values of the quantiles?
q1.q7 <- tibble(
  p = seq(0,1,by=0.01),
  q1 = quantile(x, probs=p,type=1),
  q7 = quantile(x, probs=p,type=7)
)
q1.q7

# feasible value type 1: all sample values
# feasible values type 7: all values in the interval
# [min(sample values, max(sample values)]

# f) Create a diagram which visualize the empirical distribution function,
# the function H, which connects the points
# (0, x_(1)), (1/(n-1),x_(2)), (2/(n-1),x_(3)), ..., (n/(n-1),x_(n))
# with line segments (the above linear interpolation), type 1 and type 7
# quantiles and mention the difference between type 1 and type 7 quantiles.
plot(x=sort(x), y=(0:(s.size-1))/(s.size-1),
     type="b", col = "black",
     xlim=c(0,22), ylim=c(-0.1,1.1),
     ylab="p", xlab="x",
     main="Comparison of type 1 and 7",
     sub = "black = type 7, blue = type 1")
points(y=seq(0,1, by=0.05), x=quantile(x,prob=seq(0,1, by=0.05), type=7), col="black")
points(y=seq(0,1, by=0.05), x=quantile(x,prob=seq(0,1, by=0.05), type=1), col="blue")
lines(x=emp.dist$obs, y=emp.dist$cum.rel.freq, type="s",col="blue")
axis(1, at = seq(0,20, by = 1))
axis(2, at = seq(0,1, by = 0.1))

# g) Increase the sample size from 10 to 50 and then to 100 and create
# the above diagram. What happens?
```

```
s.size <- 200
x <- sample(1:20, size = s.size, replace = TRUE)
emp.dist <- tibble(
  obs = x
) %>%
  count(obs) %>%
  mutate(
    cum.rel.freq = cumsum(n)/sum(n)
  )
#
plot(x=sort(x), y=(0:(s.size-1))/(s.size-1),
     type="l", col = "black",
     xlim=c(0,22), ylim=c(-0.1,1.1),
     ylab="p", xlab="x",
     main=paste("Comparison of type 7 and emp. distribution function, n=",s.size),
     sub = "black = type 7, blue = emp. distr. function")
#points(y=seq(0,1, by=0.02), x=quantile(x,prob=seq(0,1, by=0.02), type=7), col="black")
#points(y=seq(0,1, by=0.02), x=quantile(x,prob=seq(0,1, by=0.02), type=1), col="blue")
lines(x=emp.dist$obs, y=emp.dist$cum.rel.freq, type="s",col="blue")
axis(1, at = seq(0,20, by = 1))
axis(2, at = seq(0,1, by = 0.1))


q.tab <- tibble(
  obs = x,
  obs.ordered = sort(x),
  prop.n = (1:s.size)/s.size,
  prop.n.minus.1 = (0:(s.size-1))/(s.size-1)
)
q.tab

plot(x=q.tab$prop.n.minus.1, y=q.tab$obs.ordered,
     type="l", col = "black",
     ylim=c(0,21), xlim=c(-0.1,1.1),
     xlab="p", ylab="p quantile",
     main=paste("quantiles type=7 and type=1 - n=",s.size),
     sub = "black = type 7, blue = type 1")
#points(x=q.tab$prop.n, y=q.tab$obs.ordered, col="blue", pch=20)
segments(x0=c(0,q.tab$prop.n[-s.size]), y0=q.tab$obs.ordered[ceiling(q.tab$prop.n*s.size)],
         x1=q.tab$prop.n, y1=q.tab$obs.ordered[ceiling(q.tab$prop.n*s.size)], col="blue")
axis(2, at = seq(0,20, by = 1))
axis(1, at = seq(0,1, by = 0.1))


# Observations
# a) The deviations between the empirical distribution function F and the
# function H decrease. For big sample sizes both function are more or less
# identical.
# b) The feasible values of type 1 quantiles are {1,2,...20} whereas
# quantiles of type 7 can take every value from the interval [1,20].
# c) Type 1 quantiles are suited for discrete variables. In case of continous
# variables both types can be used and exspecially for large sample sizes the
# values are more or less identical.
```

6. 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.

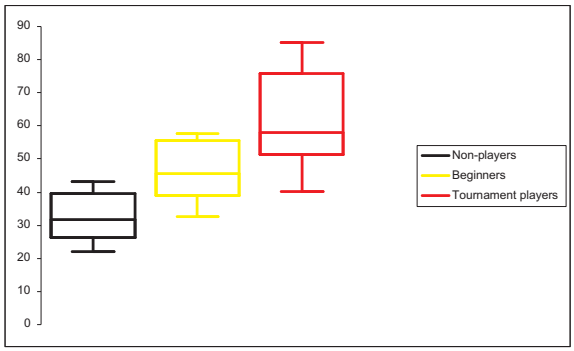
   Compare the performance for each group by computing mean, median, min, max, quartiles, interquartile range, variance. Create side-by-side box plots for these three groups. What can you say about the differences between these groups from the box plots?

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
# Solution
#
# File: des_stat_chess_sol.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?

library(tidyverse)

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")
data

data1 <- 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))
data1

# alternative: tidy the messy dataset data
data %>%
  as_tibble() %>%
  gather(key = "type", value = "res") -> data

measures <- data %>%
  group_by(type) %>%
  summarise(Min = min(res),Max=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
boxplot(data[,1],data[,2],data[,3], names=colnames(data),
        main = "side by side boxplots",
        xlab = "player type", ylab = "rem. chess positions")

# Boxplots with ggplot
boxplot(res ~ type, data = data1)
# solution with ggplot()
# 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 = data1) +
  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()
```

7. 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) Calculate the interquartile range and standard deviation for both variables. Compare the variability of both variables.

   Answer:

   (a) mean(distance) = 17.32, mean(altitude) = 612.4
   median(distance) = 16.2 resp. 16.35, median(altitude) = 502 resp. 528.5

   (b) 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.

```
#########################################################
# Descriptive Statistics: Exercise 3.1,
# Heumann, Schomaker, page 63
# Solution
#
# File: des_stat_hiking_sol.R
#
#########################################################
library(tidyverse)

# 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)

# shape of the distributions
# distance: Q2 closer to Q3, i.e. distance might be left skewed
# altitude: Q2 closer to Q1, i.e. distance might be right skewed
boxplot(altitude, main = "altitude", xlab="altitude")
boxplot(distance, main = "distance", xlab="distance")
boxplot(altitude, distance,
        main = "altitude and distance",
        names = c("distance", "altitude"))

# variance and standard deviation
var(distance)
var(altitude)
sd(distance)
sd(altitude)

# make the values comparable
distance.norm <- distance/mean(distance)
altitude.norm <- altitude/mean(altitude)
boxplot(altitude.norm, distance.norm,
        main = "altitude and distance - normed",
        names = c("distance.norm", "altitude.norm"))

# coefficients of variation
sd(distance)/mean(distance) # = sd(distance.norm)
sd(altitude)/mean(altitude) # = sd(altitude.norm)
```

8. 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" ($1 < \text{displ} \leq 3$), "medium" ($3 < \text{displ} \leq 5$) and "big" ($5 < \text{displ} \leq 8$). 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
...	..	...

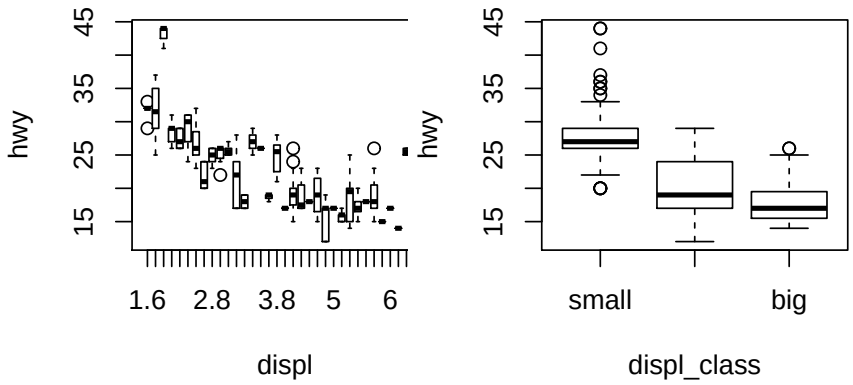
(c) Characteristic numbers of hwy grouped by displ

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



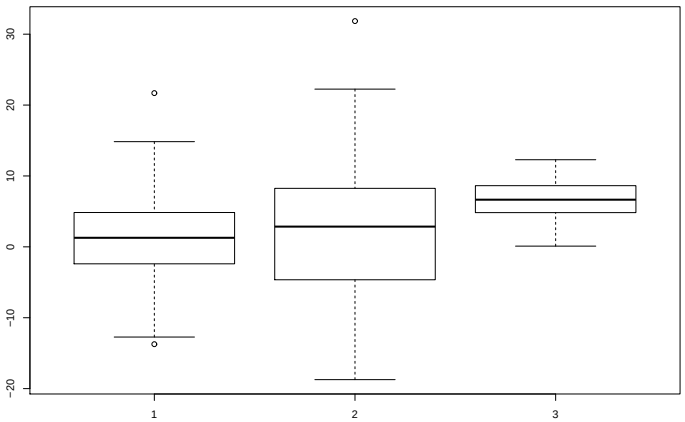
Grouping the data the association between engine displacement and highway miles per gallon becomes more: higher engine displacement correlates with low highway miles per gallon.

```
#########################################################
# Descriptive Statistics: miles per gallon
# Solution
#
# File: des_stat_miles_gallon_sol.R
#
#########################################################
library(tidyverse)

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

# select only the variables displ and hwy and add
# a columnmn 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

# 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=1),
            q2=quantile(hwy,0.5, type=1),
            q3=quantile(hwy,0.75, type=1),
```

```
            nobs = n()
            )
stat_hwy_displ

# 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),
            nobs = n()
  )
stat_hwy_class

# boxplots of hwy grouped by displ
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()

# 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()

# 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")
# zuruecksetzen 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?

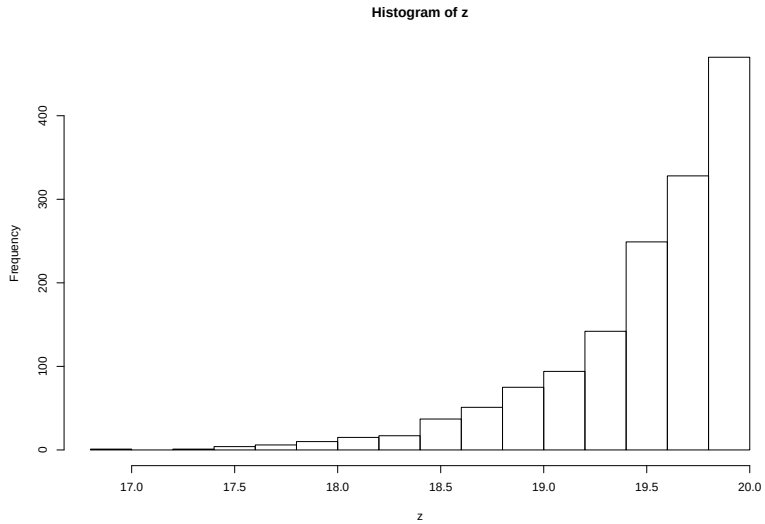
(e) Which of the three distributions has the smallest maximum value?

(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