Statistics for Computer Science

Assignment 1

Martin Gregorík

500359

Services Development Management

Faculty of Informatics
Masaryk University

April 9, 2020

Exercise 1

First, I loaded the data. Then I made a subset of cranial breadth of males from populations AUSTRALI and PERU. This was unsorted, so I sorted it and checked for missing values. There were no missing values.

```r
# I commented setwd because you probably don’t have this directory on your machine.
# And .Rnw needs to be compilable on your machine.
#setwd("/home/martingregorik/R/assignment01")
options(max.print=10000)
library(xtable)

howell <- read.csv("Howell.csv", header = TRUE)
#str(howell)

xcb_unsort <- howell$XCB[howell$Sex == ’M’ & (howell$Population == ’AUSTRALI’ | howell$Population == ’PERU’)]
xcb <- sort(xcb_unsort)
#sum(is.na(xcb)) = 0
#is.na(xcb)
```

Then I created custom functions.

```r
# Task 1
my_sample_min <- function(vec) {
  min <- Inf
  for (i in 1:length(vec)) {
    if (vec[i] < min) {
      min <- vec[i]
    }
  }
  return (min)
}

my_sample_max <- function(vec) {
  max <- -Inf
  for (i in 1:length(vec)) {
    if (vec[i] > max) {
      max <- vec[i]
    }
  }
  return (max)
}

my_sample_mean <- function(vec) {
  sum <- 0
  for (i in 1:length(vec)) {
    sum <- sum + vec[i]
  }
  return (sum / length(vec))
}

# sum of (xi - x~) squared
my_sum_sample_avg <- function(vec, exponent=2) {
  sum <- 0
  div <- 0
  avg <- my_sample_mean(vec)
  for (i in 1:length(vec)) {
    div <- vec[i] - avg
    sum <- sum + (div ^ exponent)
  }
  return (sum)
}

my_decile <- function(vec, k) {
  # k / 10 * 100
  return (vec[1:(k * 10)])
}

my_quartile <- function(vec, q) {
  # denominator
  denom <- 1 / q
  len <- length(vec)
  if (len %% 2 == 0) {
    # even
    return ((vec[len / denom] + vec[len / denom + 1]) / 2)
  } else {
    # odd
    return (vec[(len + 1) / denom])
  }
}
median <- my_quartile(xcb, 0.5)

my_five_num_sum <- function(vec) {
  return (data.frame(min=my_sample_min(vec), lower_q=my_quartile(vec,
     0.25), median=my_quartile(vec, 0.50), upper_q=my_quartile(vec,
     0.75), max=my_sample_max(vec)))
}

my_sample_skew_cramer <- function(vec) {
  nom <- my_sum_sample_avg(vec, 3)
  denom <- length(vec) * (my_sample_variance(vec) ^ (3 / 2))
  return (nom / denom)
}

my_sample_kurtosis <- function(vec) {
  nom <- my_sum_sample_avg(vec, 4)
  denom <- length(vec) * (my_sample_variance(vec) ^ 2)
  return ((nom / denom) - 3)
}
# broad = thick tails = platykurtic

my_sample_variance <- function(vec, exponent=2) {
  if (length(vec) != 0) {
    avg <- my_sample_mean(vec)
    sum <- my_sum_sample_avg(vec, exponent)
    return (sum / (length(vec) - 1))
  }
}

my_sample_standard_deviation <- function(vec, exponent=2) {
  if (length(vec) != 0) {
    pw <- my_sample_variance(vec, exponent)
    return (sqrt(pw))
  }
}

my_sample_range <- function(vec) {
  return (my_sample_max(vec) - my_sample_min(vec))
}

my_sample_decile_range <- function(vec) {
  return (my_quartile(vec, 0.90) - my_quartile(vec, 0.10))
}

my_sample_trimmed_avg <- function(vec) {
  gamma <- 0.1
  n <- length(vec)
  g <- floor(gamma * n)
  # xtg
  return ((1 / (n - 2 * g)) * sum(vec[g + 1:n - g]))
}

my_sample_trimmed_var <- function(vec) {
  gamma <- 0.1
  n <- length(vec)
  g <- floor(gamma * n)
  xtg <- my_sample_trimmed_avg(vec)
  # stg
  return ((1 / (n - (2 * g) - 1)) * sum((vec[g + 1:n - g] - xtg)^2))
}
```

Then I calculated characteristics of each population and stored them in a table.

```
# Australia
xcb_australi_unsorted <- howell$XCB[howell$Sex == ’M’ & howell$
   Population == ’AUSTRALI’]
xcb_australi <- sort(xcb_australi_unsorted)
```

```
xcb_aus_tab <- round(data.frame(
  size=length(xcb_australi),
  mean=my_sample_mean(xcb_australi),
  my_five_num_sum(xcb_australi),
  skew=my_sample_skew_cramer(xcb_australi),
  kurt=my_sample_kurtosis(xcb_australi),
  variance=my_sample_variance(xcb_australi),
  sd=my_sample_standard_deviation(xcb_australi),
  range=my_sample_range(xcb_australi),
  dec_range=my_sample_decile_range(xcb_australi),
  trim_avg=my_sample_trimmed_avg(xcb_australi),
  trim_var=my_sample_trimmed_var(xcb_australi)
), 4)
# Peru
xcb_peru_unsorted <- howell$XCB[howell$Sex == ’M’ & howell$Population == ’PERU’]
xcb_peru <- sort(xcb_peru_unsorted)
xcb_peru_tab <- round(data.frame(
  size=length(xcb_peru),
  mean=my_sample_mean(xcb_peru),
  my_five_num_sum(xcb_peru),
  skew=my_sample_skew_cramer(xcb_peru),
  kurt=my_sample_kurtosis(xcb_peru),
  variance=my_sample_variance(xcb_peru),
  sd=my_sample_standard_deviation(xcb_peru),
  range=my_sample_range(xcb_peru),
  dec_range=my_sample_decile_range(xcb_peru),
  trim_avg=my_sample_trimmed_avg(xcb_peru),
  trim_var=my_sample_trimmed_var(xcb_peru)
), 4)
# Concat them to single data frame
xcb_tab <- xcb_aus_tab
xcb_tab[2, ] <- xcb_peru_tab
rownames(xcb_tab) <- c("AUSTRALI", "PERU")
# Since the table is too long for pdf, I split them into two.
# Yes, I concatenated them earlier, but that was rows.
# Now I cut them in half by columns.
xcb_tab_first_half <- xcb_tab[, 1:(length(xcb_tab) / 2)]
xcb_tab_second_half <- xcb_tab[, ((length(xcb_tab) / 2) + 1):length(xcb_tab)]
```

	size	mean	min	lower$_q$	median	upper$_q$	max
AUSTRALI	52.00	131.94	124.00	128.00	131.00	134.00	144.00
PERU	55.00	137.95	129.00	135.00	138.00	141.00	149.00

Table 1: Characteristics of maximal cranial breadth of AUSTRALI and PERU populations

	skew	kurt	variance	sd	range	dec$_r ange$	trim$_a vg$
AUSTRALI	0.64	-0.34	26.06	5.10	20.00	14.00	163.36
PERU	-0.00	0.06	15.87	3.98	20.00	9.00	168.60

Table 2: Characteristics of maximal cranial breadth of AUSTRALI and PERU populations

Next, I created boxplots of maximum cranial breadth for each population. I set the width of boxes to be proportional to sample sizes.

```r
# Different lengths(australi is 52, peru 55), need to add 0s to the end(neglected in output)
n <- max(length(xcb_australi), length(xcb_peru))
xcb_australi_prolonged <- xcb_australi
length(xcb_australi_prolonged) <- n
max_b <- data.frame(AUSTRALI=xcb_australi_prolonged, PERU=xcb_peru)
```

```r
# Variable australi_peru_cols will be used in following exercises
australi_peru_cols = c("dodgerblue4", "indianred")
boxplot(
  max_b,
  width = c(length(xcb_australi), length(xcb_peru)),
  notch = TRUE,
  main = "Boxplot of maximal cranial breadth",
  xlab = "Countries",
  ylab = "Maximal cranial breadth (mm)",
  col = australi_peru_cols,
  pch = 16
)
points(1:2, xcb_tab$mean, col = "green", pch = 16)
legend(’topleft’, pch = 16, legend = c("Maximum", "Average"), col = c("black", "green"))
```

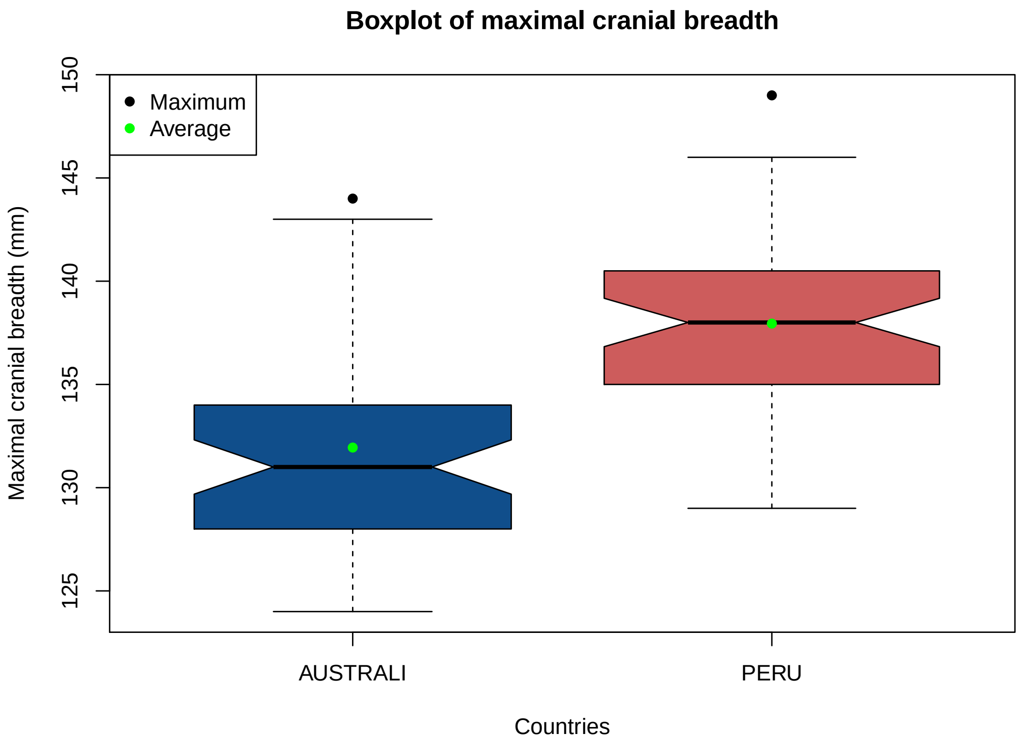

Figure 1: Boxplot of maximal cranial breadth(mm)

Created histogram of maximum cranial breadth for each population.

```
def.par <- par()
layout(matrix(c(1, 2), nrow=1, ncol=2), widths=c(0.5, 0.5))
#layout.show(n=2)
hist(
  xcb_australi,
  main = "Australi population",
  xlab = "Maximal cranial breadth(mm)",
  ylab = "Count",
  col = australi_peru_cols[1]
)
hist(
  xcb_peru,
  main = "Peru population",
  xlab = "Maximal cranial breadth(mm)",
  ylab = "Count",
  col = australi_peru_cols[2]
)
```

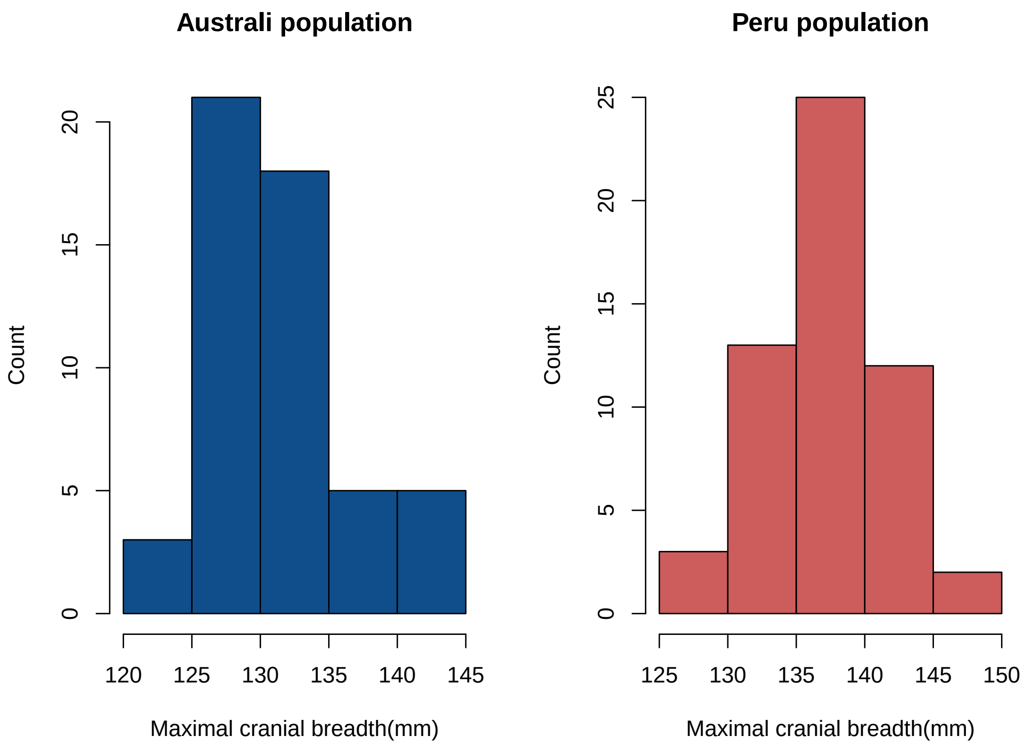

Figure 2: Histogram of maximal cranial breadth(mm)

Created normal qq-plot of maximum cranial breadth for each population.

The observed measurements(Y axis values) are from the same range so the plots can be easily compared.

```r
layout(matrix(c(1, 2), nrow=1, ncol=2), widths=c(0.5, 0.5))
#layout.show(n=2)
qqnorm(
  y = xcb_australi,
  main = "Australi population",
  xlab = "Theoretical quantities",
  ylab = "Maximal cranial breadth(mm)",
  ylim = c(min(xcb_australi, xcb_peru), max(xcb_australi, xcb_peru)),
  pch = 16,
  col = australi_peru_cols[1]
)
qqline(xcb_australi)
qqnorm(
  y = xcb_peru,
  main = "Peru population",
  xlab = "Theoretical quantities",
  ylab = "Maximal cranial breadth(mm)",
  ylim = c(min(xcb_australi, xcb_peru), max(xcb_australi, xcb_peru)),
  pch = 16,
  col = australi_peru_cols[2]
)
qqline(xcb_peru)
```

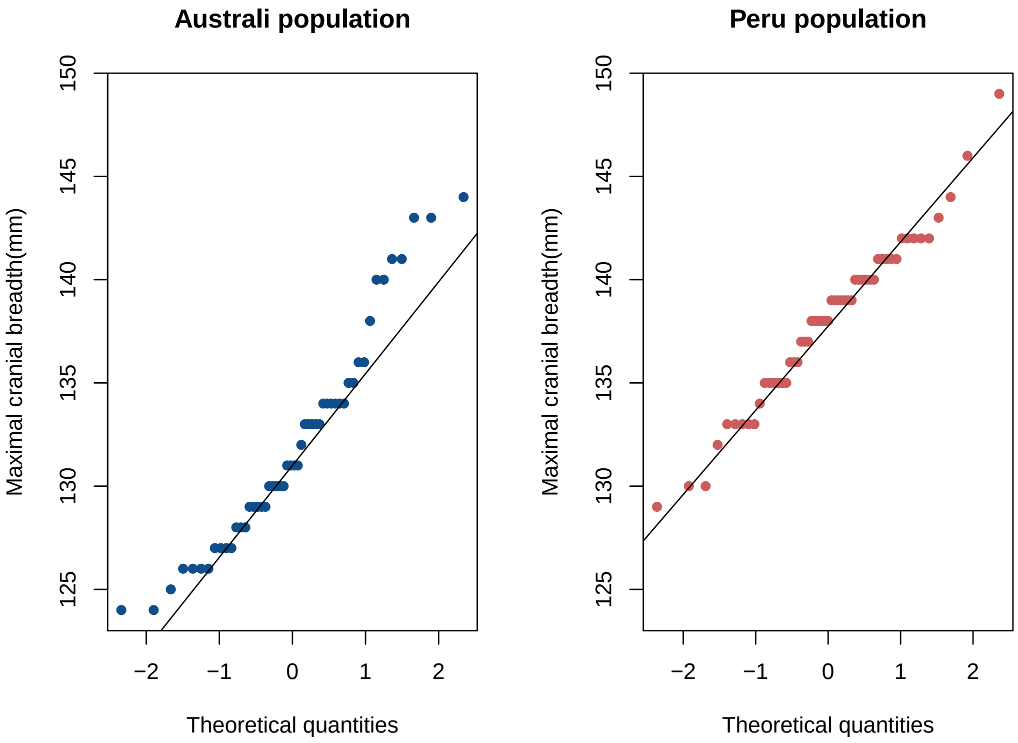

Figure 3: Normal qq-plot of maximal cranial breadth(mm)

Results and interpretation

Although both populations have same maximal cranial breadth reached, the differences can be best seen in histograms - more people of Peru population have bigger breadth than Australian people.

Variable XCB, representing maximal cranial breadth of each population, seems to be normally distributed variable in both cases.

Exercise 2

First, I loaded the data. Then I checked for missing values in desired subset. There were none. So I summed the values in each column, which gave me a total sum of men and women in each year. Lastly, I added margins, which gave me total population in all of the years. Also, I like to keep variables at the beggining of each significant part, in this case Exercise.

```
area_esp <- read.csv("area_spanish_provinces.csv", header = TRUE)
pop_esp <- read.csv2("population-spain-1998-2018.csv", header = TRUE)
#str(area_esp)
#str(pop_esp)
#pop_esp
#sum(is.na(pop_esp[, 2:length(pop_esp)])) = 0
### VARIABLES ###
legend_sex <- c("Women", "Men")
stat_years <-  c("1998", "2018")
total_pop_cols = c("pink", "dodgerblue")
year_cols = c("darkgoldenrod", "chocolate")
### VARIABLES ###
people_each_year <- as.table(colSums(pop_esp[, 2:length(pop_esp)]))
total_people_each_year <- addmargins(people_each_year, FUN = c(Total=sum
   ))
```

Then I created a barplot of total population of Spain in each of the years, with each bar divided between men and women.
The data were stored as table(from previous task), so I converted them into data frame to be able to subset values.
The years were descending from 2018, so I made them ascending.
Then I stored the data in matrix in order for barplot to accept the argument type. Since the data in data frame is ordered by gender, I simply added "byrow = TRUE" to the matrix in order to store the data correctly.
I have also set a few plot parameters:

"cex.axis", to make Y axis values smaller,
"las", to make axis labels horizontal,
"bty" in legend to hide legend box.

```
df_people <- as.data.frame(people_each_year)
# switched ordering of years
df_people <- df_people[dim(df_people)[1]:1, ]
df_people <- matrix(
  df_people[, 2],
  nrow = 2,
  ncol = 5,
  byrow = TRUE,
  dimnames = list(c("F", "M"), c("1998", "2003", "2008", "2013", "2018")
     )
)
# It is clean now
suppressWarnings(par(def.par))
barplot(
  height = df_people,
  main = "Total population of Spain",
  ylim = c(0, max(df_people)),
  beside = TRUE,
  las = 1,
  cex.axis = 0.8,
  col = total_pop_cols)
legend(’topleft’, pch = 15, legend = legend_sex, col = total_pop_cols,
   bty = ’n’)
```

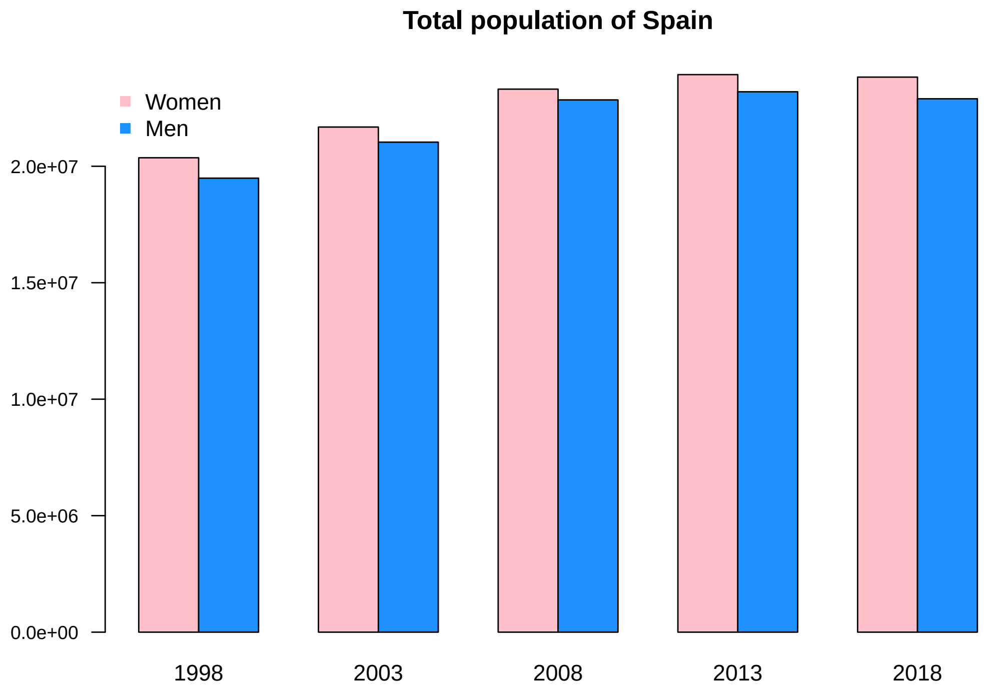

Figure 4: Barplot of total population in Spain

Then I created a barplot of relative proportions of men and women in each province in 2018. I transposed the input matrix in order to show the data by row(men and women side-by-side). I have also set few plot parameters:

"cex.names", to make the provinces' names smaller,

"las", to make axis labels perpendicular to the axis.

```
relative_pop_2018_provinces <- matrix(
  data = c(pop_esp$females.2018, pop_esp$males.2018),
  nrow = 52,
  ncol = 2
)
barplot(
  height = t(relative_pop_2018_provinces),
  names.arg = pop_esp$province,
  cex.names = 0.6,
  cex.axis = 0.8,
  main = "2018 relative population of Spain",
  ylim = c(0, max(relative_pop_2018_provinces)),
  las = 2,
  beside = TRUE,
  col = total_pop_cols)
legend('topleft', pch = 15, legend = legend_sex, col = total_pop_cols,
   bty = 'n')
```

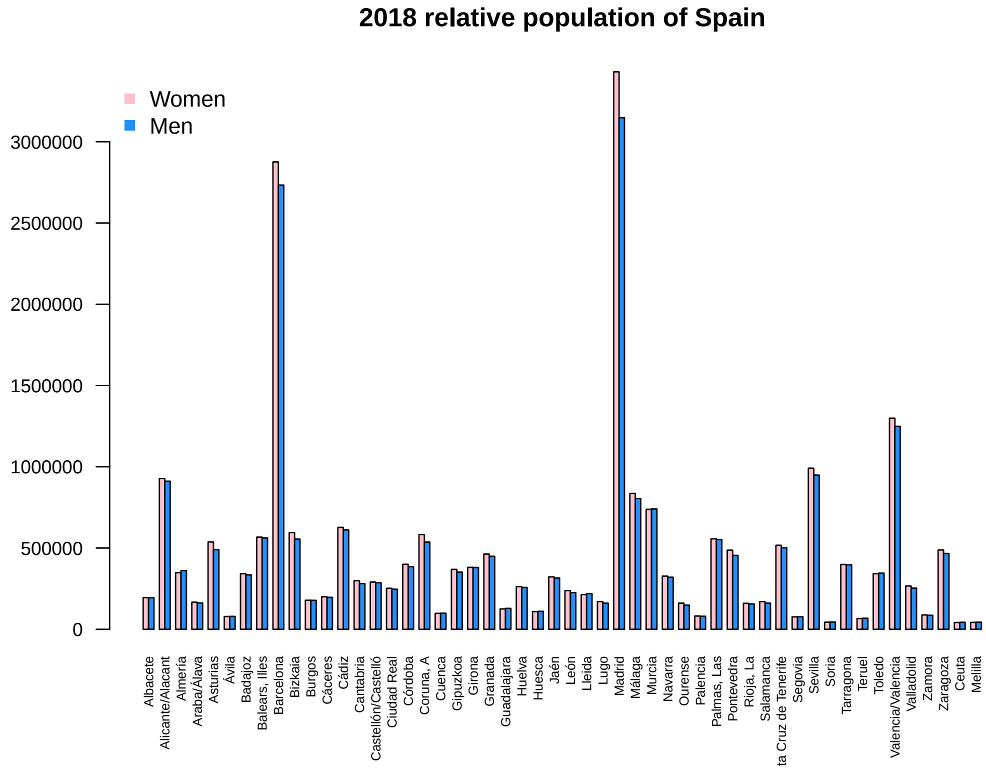

Figure 5: Barplot of relative population of Spain in 2018

Then I calculated population density in 1998 and 2018, for each province. The are of each province was loaded in the beginning of this exercise. I created a helper matrix to store intermediate values.

```r
ppl_in_two_years <- matrix(
  data = c(
    pop_esp$males.1998,
    pop_esp$females.1998,
    pop_esp$males.2018,
    pop_esp$females.2018
    ),
  nrow = nrow(pop_esp),
  ncol = 4,
  dimnames = list(pop_esp$province, c("males.1998", "females.1998", "males.2018", "females.2018"))
)
total_1998 <- as.vector(rowSums(ppl_in_two_years[, 1:2]))
total_2018 <- as.vector(rowSums(ppl_in_two_years[, 3:4]))
density_1998 <- total_1998 / area_esp$Area
density_2018 <- total_2018 / area_esp$Area
den <- data.frame(
  Province=pop_esp$province,
  "Year 1998" = density_1998,
  "Year 2018" = density_2018
)
```

In subtask a, fistly, I sorted the data by year. Next, I calculated the estimates of characteristics of population density, each year separately. Lastly, I bound them together into one table.

```r
# subtask a)
den_sorted_1998 <- sort(den$Year.1998)
den_sorted_1998_tab <- round(data.frame(
  size=length(den_sorted_1998),
  mean=my_sample_mean(den_sorted_1998),
  my_five_num_sum(den_sorted_1998),
  skew=my_sample_skew_cramer(den_sorted_1998),
  kurt=my_sample_kurtosis(den_sorted_1998),
  sd=my_sample_standard_deviation(den_sorted_1998)
), 4)
den_sorted_2018 <- sort(den$Year.2018)
den_sorted_2018_tab <- round(data.frame(
  size=length(den_sorted_2018),
  mean=my_sample_mean(den_sorted_2018),
  my_five_num_sum(den_sorted_2018),
  skew=my_sample_skew_cramer(den_sorted_2018),
  kurt=my_sample_kurtosis(den_sorted_2018),
  sd=my_sample_standard_deviation(den_sorted_2018)
), 4)
density_characteristics <- rbind(den_sorted_1998_tab, den_sorted_2018_tab)
row.names(density_characteristics) <- stat_years
```

	size	mean	min	lower_q	median	upper_q	max	skew	kurt	sd
1998	52.00	282.01	9.17	29.36	79.29	118.94	4623.69	4.12	16.24	848.58
2018	52.00	356.87	9.02	28.23	84.74	151.19	6644.92	4.38	19.20	1128.98

Table 3: Characteristics of population density of Spain in 1998 and 2018

In subtask b, I created a boxplot of population density in 1998 and in 2018. I have set the Y axis limits in order to see the graphs nicely, because there were some outliers which would squish the whole plot.

```
# subtask b)
suppressWarnings(par(def.par))
boxplot(
  den[2:3],
  notch = TRUE,
  main = "Pop. density in 1998 & 2018",
  ylab = "Population density",
  ylim = c(0, 250),
  pch = 16,
  col = year_cols
)
```

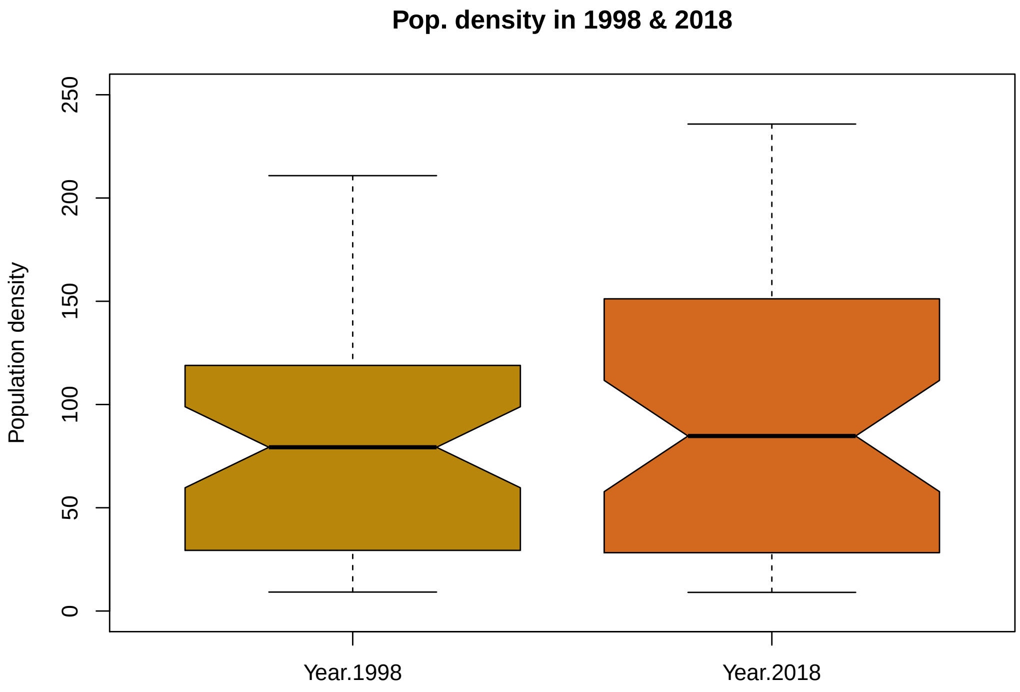

Figure 6: Boxplot of population density in Spain in 1998 and 2018

In subtask c, I created histogram of population density in 1998 and in 2018. There are two histograms, one for each year. Histogram for year 2018 is overlapping histogram for year 1998.

I changed the number of breaks. I was hoping to achieve a plot with province names, just like barplot of relative population of Spain(Figure 5), but it did not turn out that way. I'm really sad because that would make comparisons far better. Now I can't recognize the provinces. I left it as it is because I still think it looks better than the default.

Lastly, I also changed axes a little bit because I didn't like the values inserted by default, they were too sparse, too spread out.

```
# subtask c)
suppressWarnings(par(def.par))
rgb_cols = c(rgb(0.6,0.4,0.2,0.5), rgb(0.3,0.5,0.6,0.5))
hist(
  den$Year.1998,
  main = "Pop. density in 1998 & 2018",
  xlab = "Density(people/km^2)",
  ylab = "Number of provinces",
  las = 2,
  col = rgb_cols[1],
  breaks = density_characteristics$size[1],
  xaxt = ’n’,
  yaxt = ’n’
)
# X axis
axis(
  side = 1,
  at = seq(0, max(density_characteristics$max), by = 250),
  las = 2,
  cex.axis = 0.8
)
# Y axis
axis(
  side = 2,
  at = seq(0, density_characteristics$size[1], by = 5),
  las = 2,
  cex.axis = 0.8
)
hist(
  den$Year.2018,
  col = rgb_cols[2],
  breaks = density_characteristics$size[2],
  add = T)
legend(’topright’, pch = 15, legend = stat_years, col = rgb_cols, bty = ’n’)
```

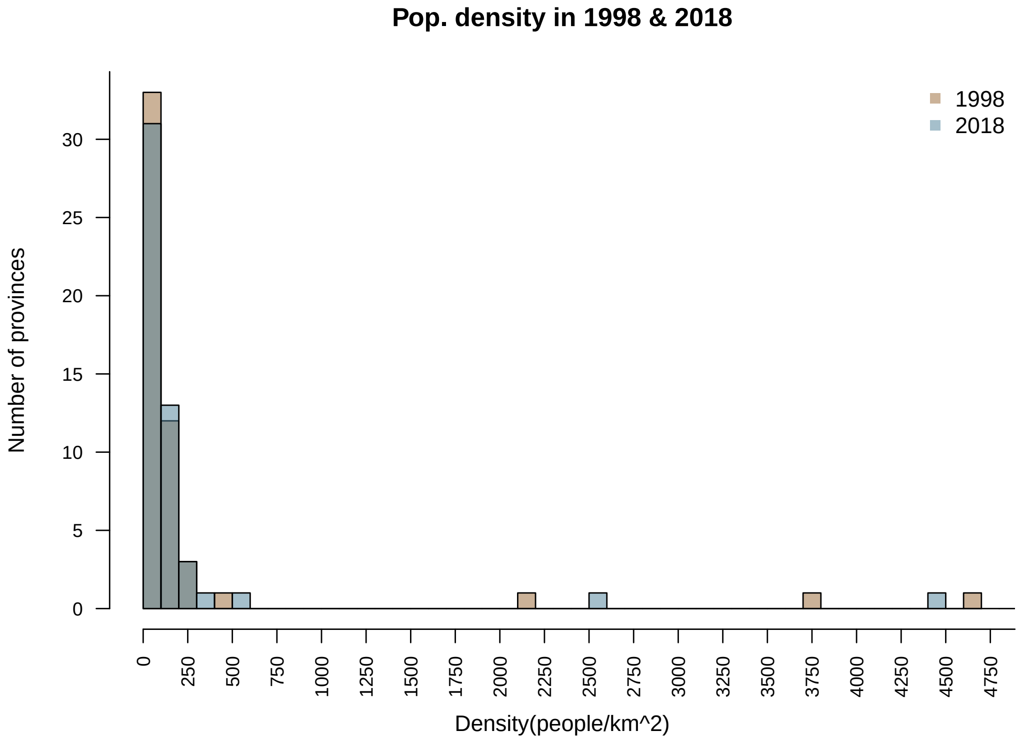

Figure 7: Histogram of population density in Spain in 1998 and 2018

Results and interpretation

Total population of Spain increased in years 1998-2018, even though there is a minor decrease between years 2013 and 2018.

Women are still slightly dominant when it comes to population proportions.

As we can see from the population density characteristics, the maximum population density increased from 4600 to 6600, which can also be seen in histogram. The mean population density increased from 282 to 356, which can be seen better e.g. in boxplot.