**Analyzing 120 years of the Olympic Games**

**Name:** Karolina Grodzinska  
**Class:** ALY6010: Probability Theory and Introductory Statistics

**INTRODUCTION**

Exploratory data analysis (EDA) is not a formal process but rather the investigation of many ideas that occur to the analyst at the initial phase of working with a dataset. It helps to understand the dataset better. Some of the initial ideas will end up as dead ends, while some other ones might lead to interesting insights. Exploratory data analysis is an important part of data analysis, it is done through generating questions about the data, searching for answers by visualizing, transforming, and modeling the data, and then using what could be learned from the previous steps to refine the questions or creating new ones. The main goal of EDA is to create many interesting leads that can be explored more in-depth at the later phase of data analysis (R for Data Science).

There are many analytical techniques that allow us to convert data into information. Descriptive analysis tries to answer the question “what happened in the past?”, inferential statistics asks “what about the rest of the population?”, diagnostic analysis tried to uncover what are the reasons for certain trends, predictive analysis answers the question “what’s likely to happen next?”, and prescriptive analysis answers the question “what should we do about it?” (Jones, 2020).

This project will be focused on the analysis of a dataset regarding 120 years of the Olympics. The dataset contains 271116 rows and 15 columns with information about the athletes participating in the Olympic Games (both winter and summer) held between 1896 - 2016.

**ANALYSIS SECTION**

olympics = read.csv("olympics.csv")

summary(olympics)

## ID Name Sex Age

## Min. : 1 Length:271116 Length:271116 Min. :10.00

## 1st Qu.: 34643 Class :character Class :character 1st Qu.:21.00

## Median : 68205 Mode :character Mode :character Median :24.00

## Mean : 68249 Mean :25.56

## 3rd Qu.:102097 3rd Qu.:28.00

## Max. :135571 Max. :97.00

## NA's :9474

## Height Weight Team NOC

## Min. :127.0 Min. : 25.0 Length:271116 Length:271116

## 1st Qu.:168.0 1st Qu.: 60.0 Class :character Class :character

## Median :175.0 Median : 70.0 Mode :character Mode :character

## Mean :175.3 Mean : 70.7

## 3rd Qu.:183.0 3rd Qu.: 79.0

## Max. :226.0 Max. :214.0

## NA's :60171 NA's :62875

## Games Year Season City

## Length:271116 Min. :1896 Length:271116 Length:271116

## Class :character 1st Qu.:1960 Class :character Class :character

## Mode :character Median :1988 Mode :character Mode :character

## Mean :1978

## 3rd Qu.:2002

## Max. :2016

##

## Sport Event Medal

## Length:271116 Length:271116 Length:271116

## Class :character Class :character Class :character

## Mode :character Mode :character Mode :character

##

##

##

##

The dataset consists of 5 numerical variables: ID represented by a unique number for each athlete, age, weight, height, and year. Height is presented in centimeters and weight in kilograms.

The remaining variables are of a type character. They include:

* Name (Athlete’s name)
* Sex (M or F)
* Team (name of the team, mostly it’s a country but sometimes it is a country with a number or an old name such as “East Germany”)
* NOC (National Olympic Committee 3-letter code, more uniform way of referring to a team)
* Games (Year and season)
* Season (winter or summer)
* City (where the Games have been hosted)
* Sport (name of the discipline)
* Event (subcategory - for example running will be divided by distance)
* Medal (Gold, Silver, Bronze, or NA)

**Exploratory Data Analysis**

*# I'd like to check the missing values*

na\_count <-sapply(olympics, **function**(olympics) sum(length(which(is.na(olympics)))))

na\_count

## ID Name Sex Age Height Weight Team NOC Games Year Season

## 0 0 0 9474 60171 62875 0 0 0 0 0

## City Sport Event Medal

## 0 0 0 231333

From the summary, I can see that age has 9474 missing values, height 60171, and weight 62875. Medals have a lot of missing values but that’s due to the fact that when an athlete wasn’t in the top 3, he did not win any medals.

Since the original dataset consisted of 271.116 observations, I can just remove rows that do not include any numerical variables, as I will be interested in exploring these variables.

df = olympics %>% drop\_na(Height, Weight, Age)

Now my dataset includes 206.165 observations.

I’d like to see what’s the distribution of athletes between genders:

a = barplot(table(df$Sex),

main = "How many Olympic athletes are male and female?",

xlab = "Sex",

ylab = "Number of athletes",

ylim = c(0,150000),

col = c("lightcoral", "cornflowerblue"))

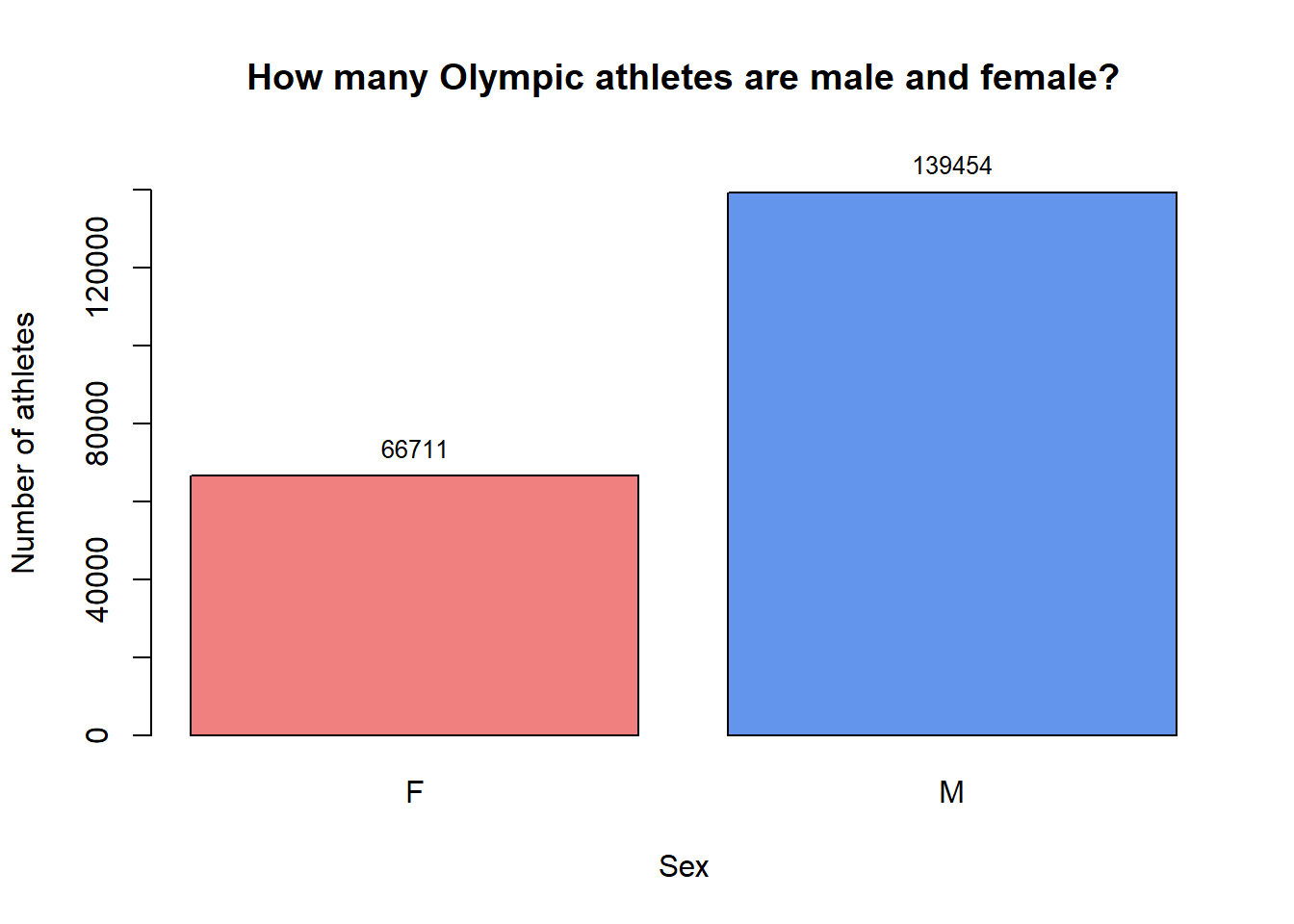
text(y = table(df$Sex),

a,

table(df$Sex),

cex=0.8,

pos = 3)



Now it becomes clear that there are twice as many male athletes than female ones. Thus, it would make sense to look at the distribution of numerical variables per gender.

As we have learned, charts can be either diagnostic or for presentation purposes. Thus, I will start my analysis by showing diagnostic charts such as histograms and box plots.

*# creating subsets to look at the genders separately*

women = dplyr::filter(df, Sex == "F")

men = dplyr::filter(df, Sex == "M")

*# Let's see how the it looks like on histograms and boxplots:*

par(mfrow=c(2,1)) *# showing one graph under the other*

par(mar=c(3,3,2,1)) *# changes margins within the figure*

hist(women$Age,

breaks = 30,

col = brewer.pal(9, "RdPu"),

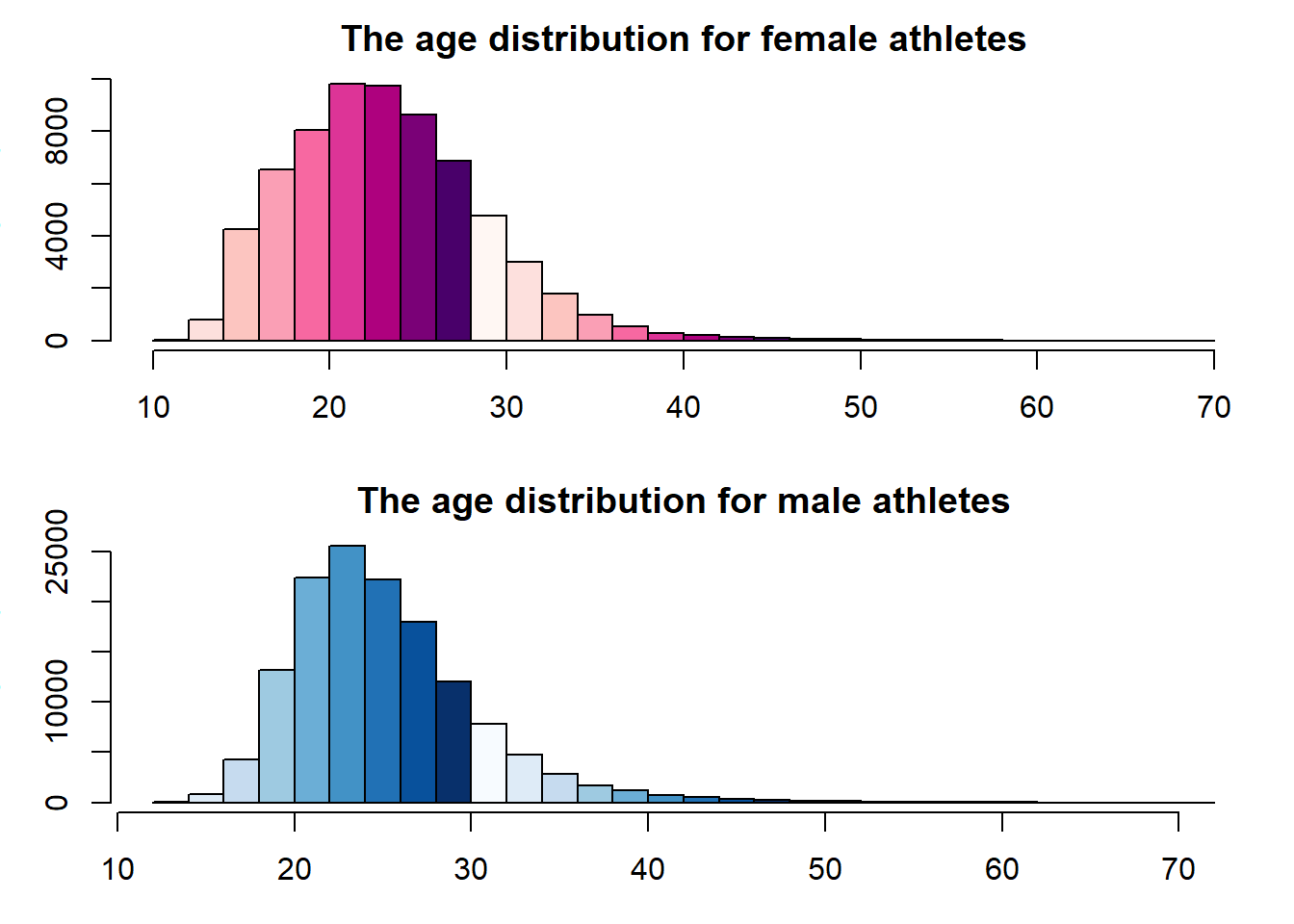
main = "The age distribution for female athletes")

hist(men$Age,

breaks = 30,

col = brewer.pal(9, "Blues"),

main = "The age distribution for male athletes")



While describing data it is important to recognize the shapes of its distribution. A good way to see the distribution is to represent it visually using a histogram. Histograms show frequencies of data points for a specified number of bins (Bluman, 2018) - in this case, it is 30 bins for both genders.

Both distributions are right-skewed. It looks like there are more females under 20 competing at the Olympic Games compared to the histogram representing male athletes. Both distributions have a peak between 20-30 years but the most frequent age (or the mode) of female athletes is lower than for men.

Now, I’d like to explore the other numerical variables included in this dataset - height and weight.

*# Box plots for weight*

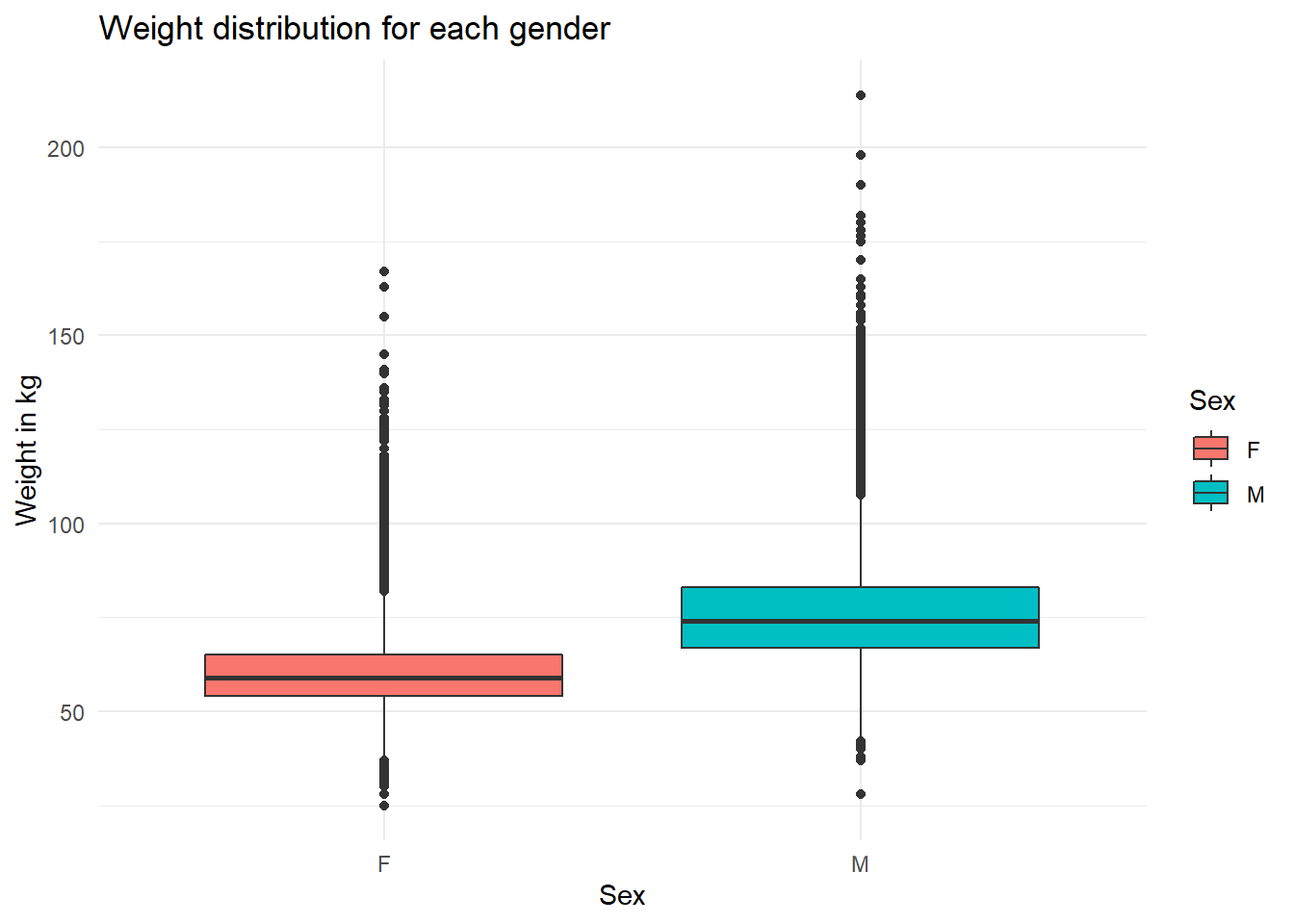
ggplot(df, aes(x = Sex, y = Weight, fill = Sex)) +

geom\_boxplot() +

scale\_color\_manual(values=c("lightcoral", "cornflowerblue")) +

labs(title = "Weight distribution for each gender", y = "Weight in kg") +

theme\_minimal()



As I have mentioned earlier, box plots show a five-number summary of the data. That means that: the minimum value, the first quartile, the median, the third quartile, and the maximum value are represented graphically on a plot (Bluman, 2018).

The figure above shows the weight distribution for both genders. The thick line in the middle represents the median - we can see that it is higher for male athletes. The “boxes” are drawn between the 1st and 3rd quartile, showing the distribution of weights. It is also more dispersed for male athletes. The dots above and below the plots represent outliers. As it can be observed, the data for both genders include outliers both higher and lower than the 1.5 x interquartile range.

*# Let's see how the it looks like on histograms:*

par(mfrow=c(2,1))

par(mar=c(3,3,2,1))

hist(women$Weight,

xlim = c(30, 110),

breaks = 30,

col = brewer.pal(9, "RdPu"),

main = "The weight distribution for female athletes")

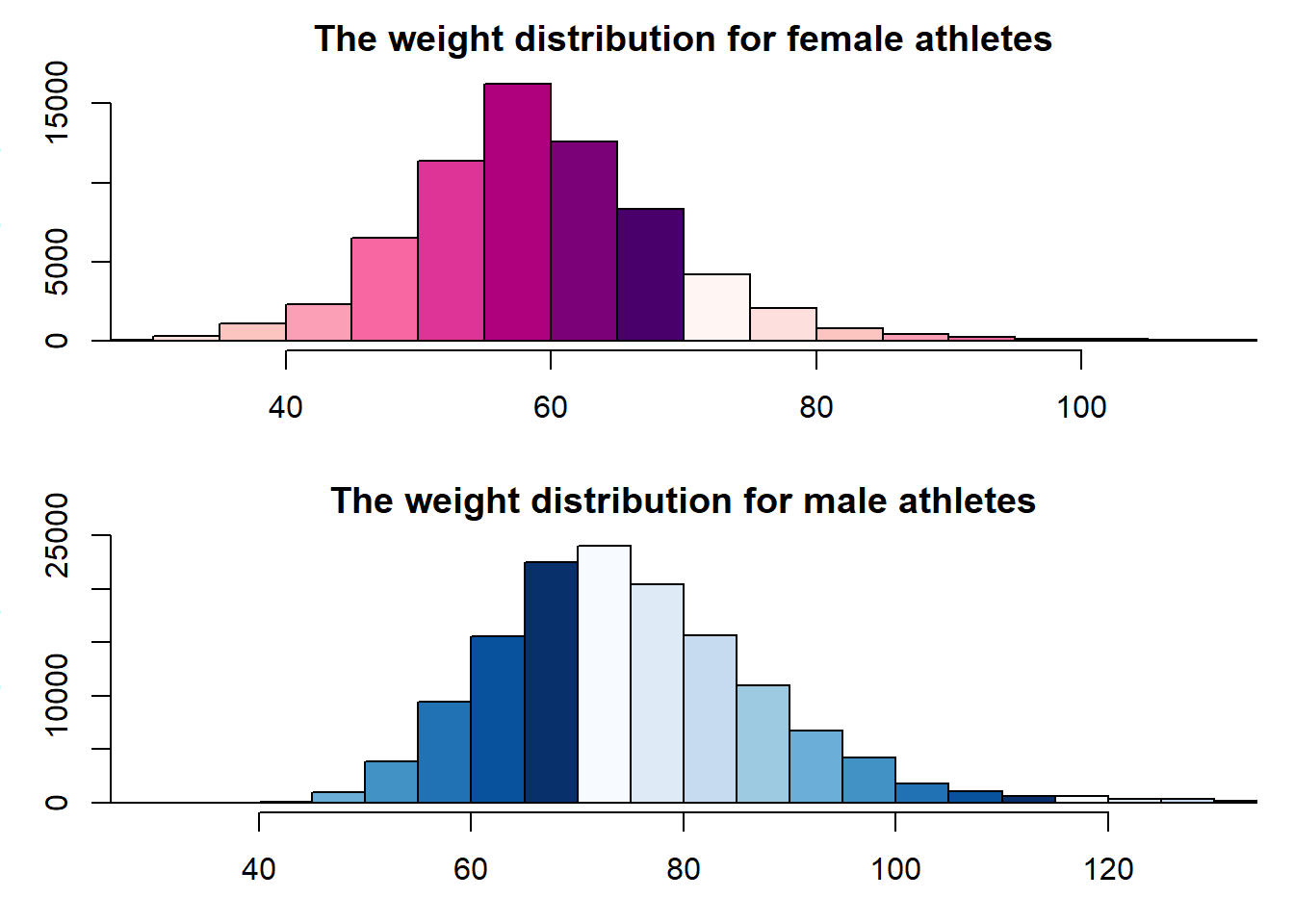
hist(men$Weight,

xlim = c(30, 130),

breaks = 30,

col = brewer.pal(9, "Blues"),

main = "The weight distribution for male athletes")



It seems that the variable storing data about weight is approximately normally distributed, slightly skewed to the right. The mode for weight of male athletes is higher than for female ones.

The histogram also allows us to see that many female athletes weigh 60 kilograms or less but what is the exact percentage?

pnorm(60, mean = mean(women$Weight), sd = sd(women$Weight))

## [1] 0.4991178

Now, I’d like to take a closer look at outliers plotted above the box plot, as there is an overlap between genders. I will examine female athletes that weight more than the third quartile + 1.5 times interquartile range and male athletes that weight less than the value of the first quartile - 1.5 times interquartile range.

*# Compute the first and third quartiles*

q1 <- quantile(women$Weight, 0.25)

q3 <- quantile(women$Weight, 0.75)

iqr <- q3 - q1

*# Calculate the lower and upper cutoffs for outliers*

lower <- q1 - 1.5 \* iqr

upper <- q3 + 1.5 \* iqr

*# Filter weight to find outliers*

out\_upper = women %>%

select(c(1:7)) %>%

filter(Weight > upper)

head(out\_upper)

## ID Name Sex Age Height Weight Team

## 1 5 Christine Jacoba Aaftink F 21 185 82 Netherlands

## 2 5 Christine Jacoba Aaftink F 21 185 82 Netherlands

## 3 5 Christine Jacoba Aaftink F 25 185 82 Netherlands

## 4 5 Christine Jacoba Aaftink F 25 185 82 Netherlands

## 5 5 Christine Jacoba Aaftink F 27 185 82 Netherlands

## 6 5 Christine Jacoba Aaftink F 27 185 82 Netherlands

There are many (1668) observations that are classified as outliers with weight higher than the third quartile + 1.5 times interquartile range. This table actually only shows one athlete that participated in multiple events, thus, she is mentioned many times in this dataset.

Let’s take a look at males that weight less than q1 - 1.5 \* IQR. There should be fewer athletes in this subset.

*# Compute the first and third quartiles (I can overwrite the previous values as I'm not going to use them anymore)*

q1 <- quantile(men$Weight, 0.25)

q3 <- quantile(men$Weight, 0.75)

iqr <- q3 - q1

*# Calculate the lower and upper cutoffs for outliers*

lower <- q1 - 1.5 \* iqr

upper <- q3 + 1.5 \* iqr

*# Filter weight to find outliers*

men %>%

select(c(1:6)) %>%

filter(Weight < lower)

## ID Name Sex Age Height Weight

## 1 18005 Cao Yuan M 17 160 42

## 2 18005 Cao Yuan M 21 160 42

## 3 18005 Cao Yuan M 21 160 42

## 4 24814 Barry Edward Dagger M 39 147 41

## 5 24814 Barry Edward Dagger M 47 147 41

## 6 38251 Wayne Bruce Gammon M 14 155 38

## 7 46484 Ali Ismael Hassan M 21 157 42

## 8 55906 Daniel Jorge M 13 152 41

## 9 73023 Thomas P. Mack, Jr. M 14 147 41

## 10 73023 Thomas P. Mack, Jr. M 18 147 41

## 11 87343 Miguel Nez Lima M 20 170 42

## 12 96860 Robert Archibald "Bob" Prentice M 35 156 42

## 13 99902 Erik Remmerswaal M 31 147 38

## 14 100743 Michel Riendeau M 21 167 41

## 15 101936 Mircea Roger M 13 146 40

## 16 101936 Mircea Roger M 13 146 40

## 17 113413 Alaeddin Soueidan M 13 148 37

## 18 115166 Orlando Maurits Stewart M 34 153 40

## 19 134370 Albert Ferdinand "Al" Zerhusen M 24 183 28

Only 19 male athletes have been classified as outliers on the lower end of weight distribution.

One of the uses of hypothesis testing is checking whether a sample statistic is close or far away from the expected value (Cotton). Now, I am ready to conduct my first statistical test based on the mean - it will be a one-tailed test. There are two statistical tests used for hypotheses related to means - the t-test and the z-test (Bluman, 2018). I will start by using a t-test.

My hypotheses are stated as follows:

*H0 = there is no difference in average weight between male and female athletes*

*H1 = Male athletes weight more (on average) than female athletes*

t.test(men$Weight, mu = mean(women$Weight), alternative = 'greater')

##

## One Sample t-test

##

## data: men$Weight

## t = 446.75, df = 139453, p-value < 2.2e-16

## alternative hypothesis: true mean is greater than 60.02257

## 95 percent confidence interval:

## 75.73249 Inf

## sample estimates:

## mean of x

## 75.79055

Small p-values mean that the statistic is producing unlikely results. P-values measure the evidence - whether we should accept the original assumption or we should favor the alternative hypothesis (Cotton).

The p-value here is very close to 0, which means that I can reject the null hypothesis both for the chosen confidence interval of 95% and I could have also done that if I set the confidence interval to 99%, as the p-value has a lower value than alpha.

Let’s see the box plots for height:

*# Box plots for height*

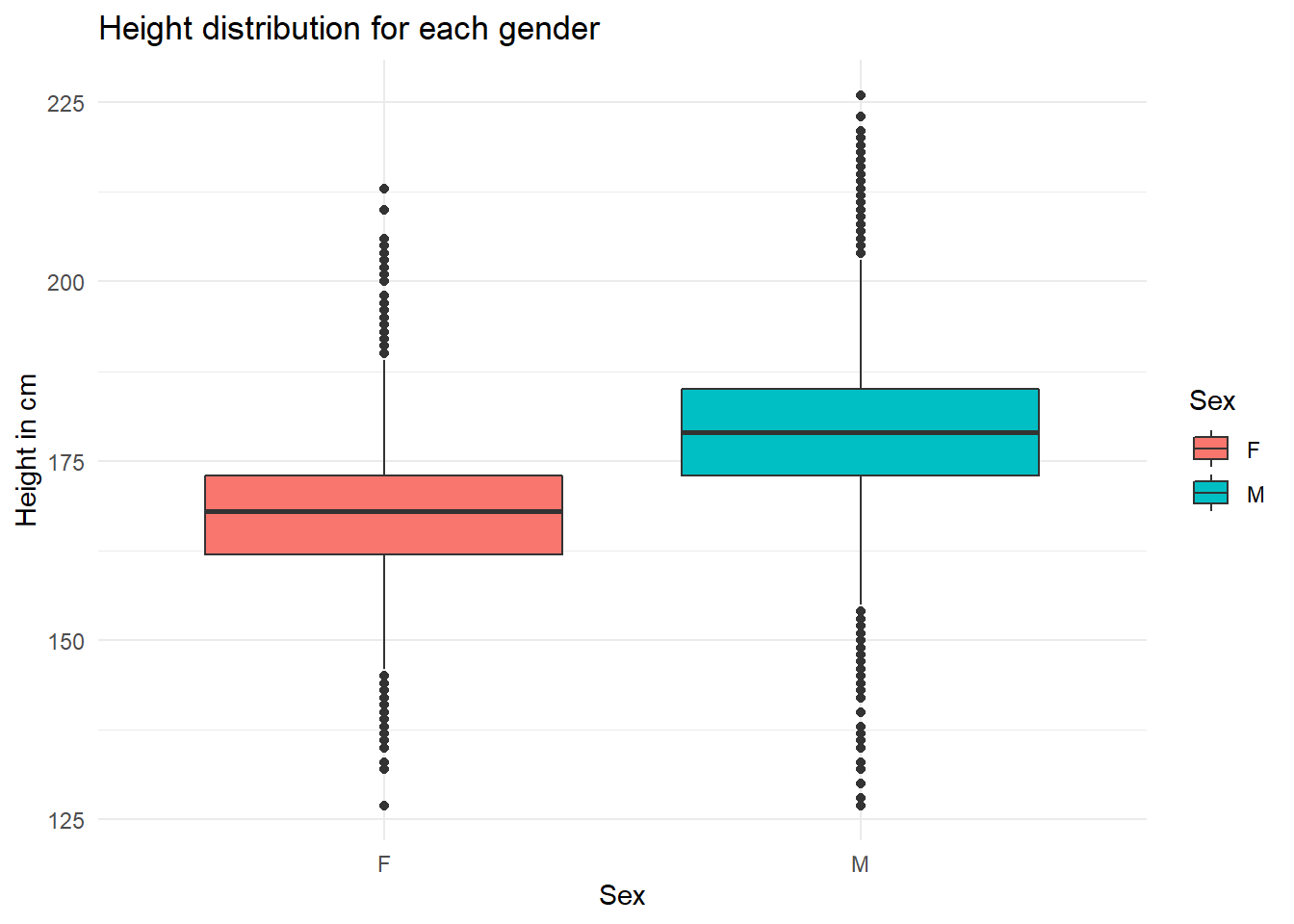
ggplot(df, aes(x = Sex, y = Height, fill = Sex)) +

geom\_boxplot() +

scale\_color\_manual(values=c("lightcoral", "cornflowerblue")) +

labs(title = "Height distribution for each gender", y = "Height in cm") +

theme\_minimal()



Height, similarly to weight, is higher for male athletes. There seem to be athletes of both genders that are just above the 125cm, however, there are no female athletes taller than 215 cm.

Another interesting thing to explore would be looking at a relation between the height and weight. I will illustrate it using a scatter plot.

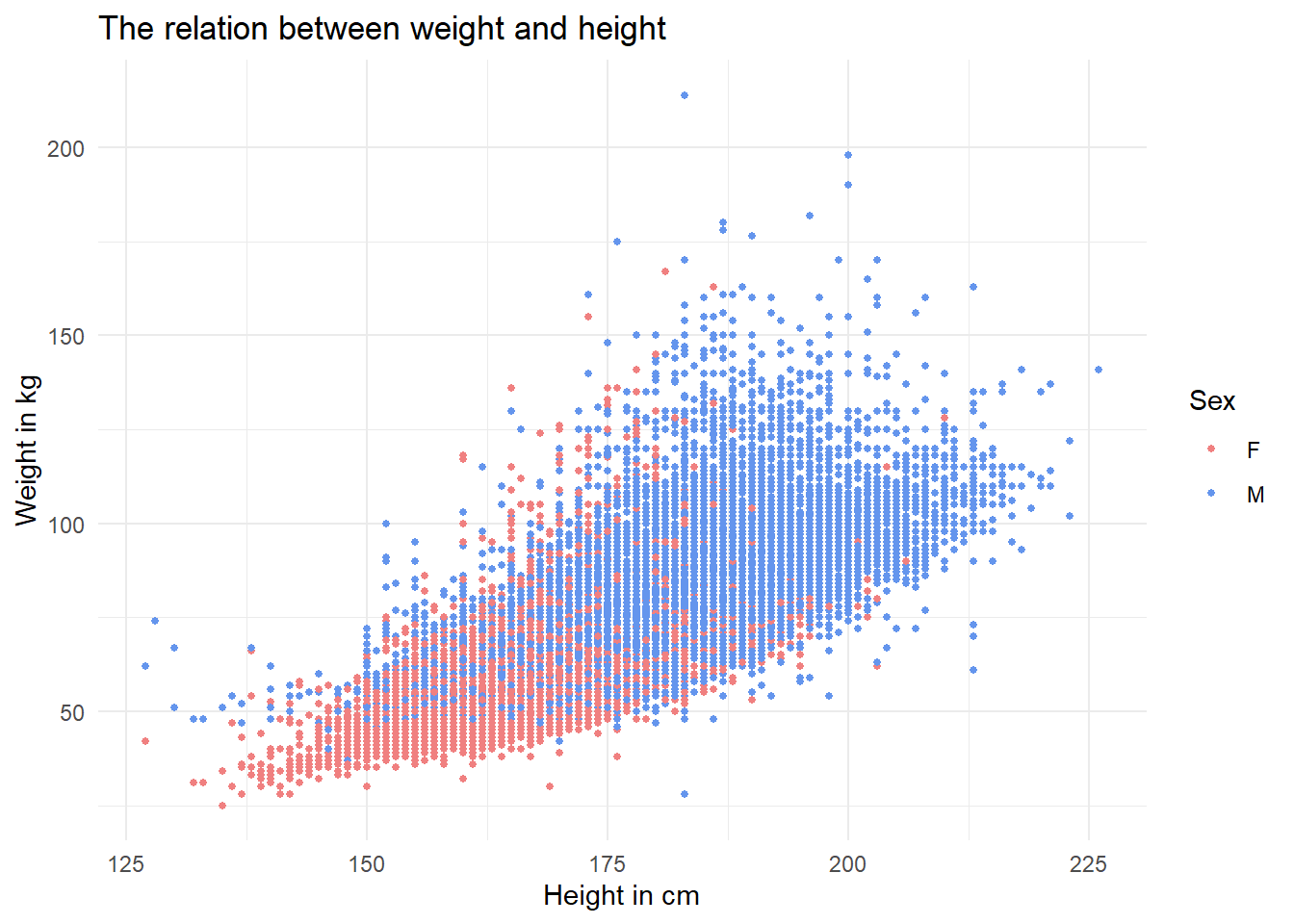
ggplot(df, aes(x = Height, y = Weight, color = Sex)) +

geom\_point(size = 1) +

scale\_color\_manual(values=c("lightcoral", "cornflowerblue")) +

theme\_minimal() +

labs(title = "The relation between weight and height", y = "Weight in kg", x = "Height in cm")



A scatter plot is a graph representing ordered pairs of numbers that represent independent variable x (in this case height) and dependent variable y (in this case weight, as I expect that taller people should also weigh more). Scatter plots are considered a good visual way of describing the nature of the relationship between the two variables (Bluman, 2018).

This plot shows a positive linear relationship between height and weight, so indeed the taller a person is, the higher the weight. Each point is colored based on gender, thus, we can also see that in general men are taller and weigh more, which we already know from the boxplots presented in previous figures. There are some exceptions in this dataset but I would expect that it depends on the sport - gymnasts are usually short, weightlifters would usually weigh more, no matter if it’s a woman or a man, and basketball players would usually be tall.

As the last step in analyzing the numerical variables, I’d like to take a look at height and weight distributions of athletes for different disciplines.

In order to do that, I will create subsets for winter and summer sports. After creating the initial visualization, it appeared to me that ice hockey has been included once during summer Olympics (in 1920), so I decided to remove it from that subset and include hockey only among winter sports. The height and weight for just one year may also not be representative in case athletes got taller or shorter over time.

*# creating subsets to look at summer and winter sports separately for clearer visualizations*

summer = dplyr::filter(df, Season == "Summer")

summer = summer[!grepl("Ice", summer$Sport),] *# in 1920 ice hockey was classified as summer sport for some reason*

winter = dplyr::filter(df, Season == "Winter")

I will be using dot plots - a type of chart where each value is plotted as a point above horizontal axis (Bluman, 2018). However, I will not be treating each data point separately but rather sum them up using average values. In that way, I will not be looking at some outliers but just how tall people competing in that sport are on average. Using just one value (the average) eliminates clutter from the chart.

*# Let's see average heights for summer sports*

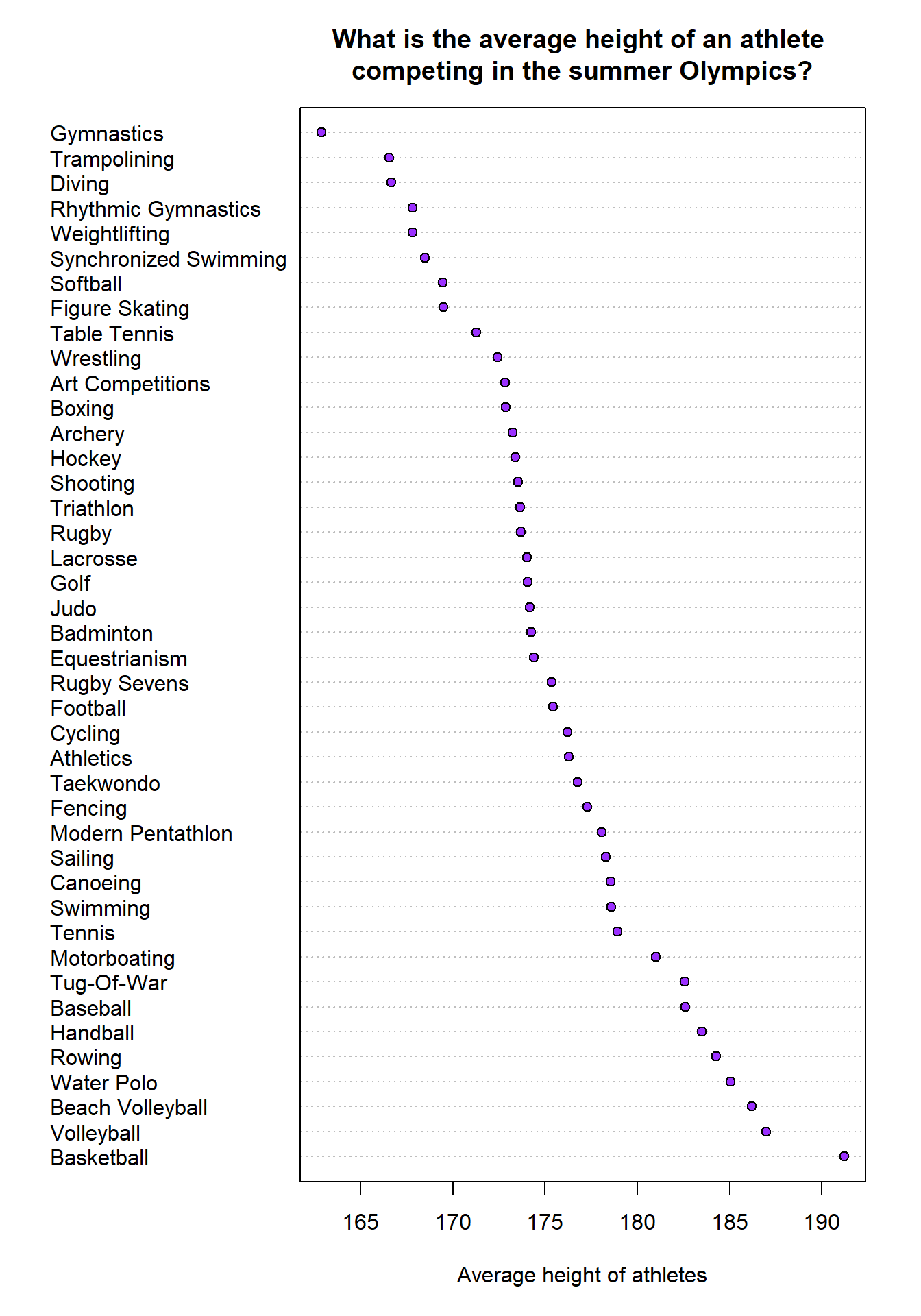
summer\_heights = tapply(summer$Height, summer$Sport, mean)

summer\_heights = sort(summer\_heights, decreasing = TRUE)

dotchart(summer\_heights, pch = 21, bg = "purple1",

xlab="Average height of athletes",

main = "What is the average height of an athlete \ncompeting in the summer Olympics?")



From the figure above, it becomes obvious that there are big height differences between the disciplines - gymnasts are the shortest and basketball players tend to be the tallest.

*# Let's see average heights for winter sports*

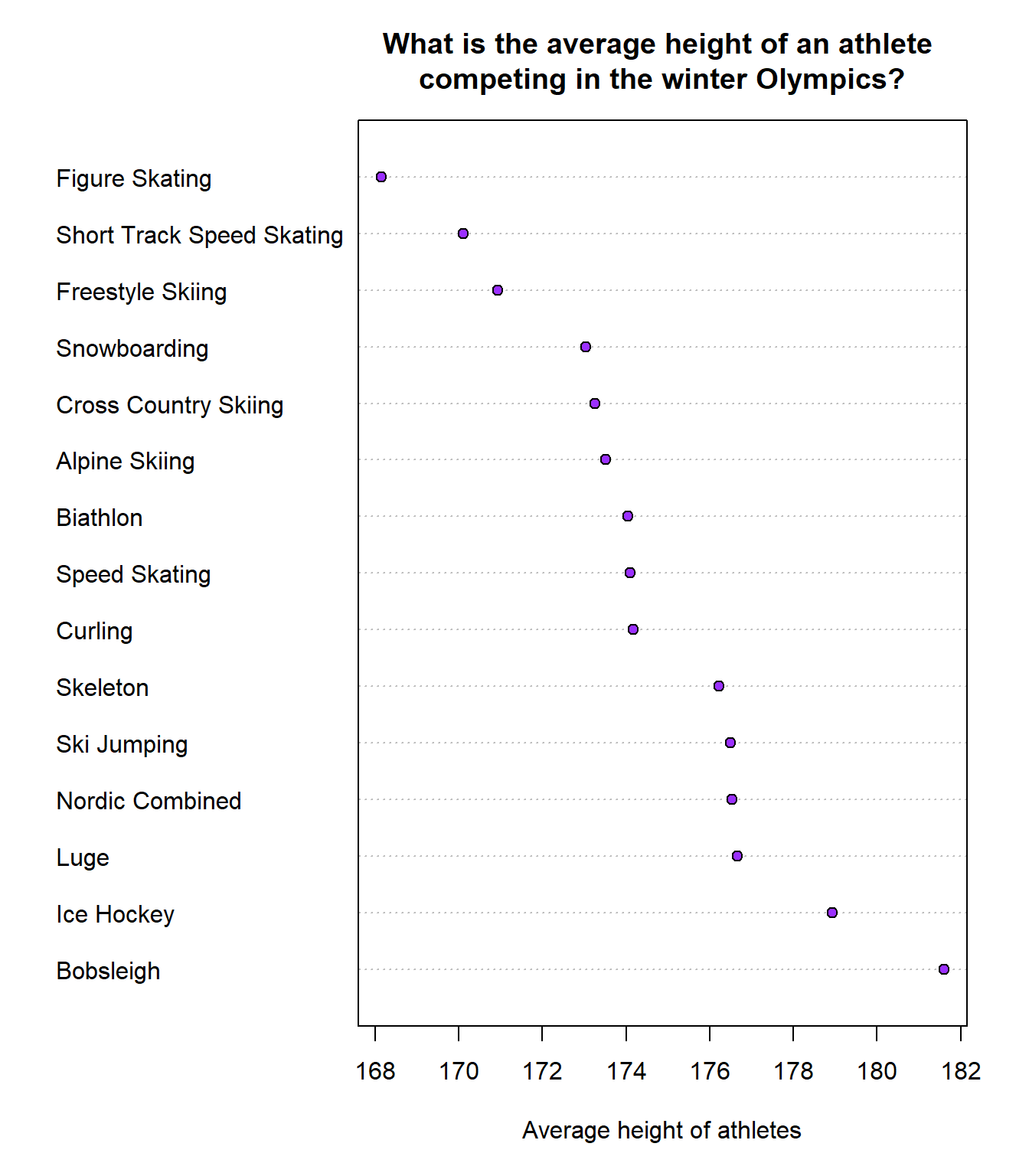
winter\_heights = tapply(winter$Height, winter$Sport, mean)

winter\_heights = sort(winter\_heights, decreasing = TRUE)

dotchart(winter\_heights, pch = 21, bg = "purple1",

xlab="Average height of athletes",

main = "What is the average height of an athlete \ncompeting in the winter Olympics?")



The distribution of heights for the athletes competing in the winter games seems to be less dispersed than between sports played during the summer Olympics. One of the reasons might be that there are fewer disciplines but also some sports require specific physical attributes from the athletes.

It is quite surprising for me that bobsleigh was placed the highest based on the average height for winter sports.

*# Let's see average weights for summer sports*

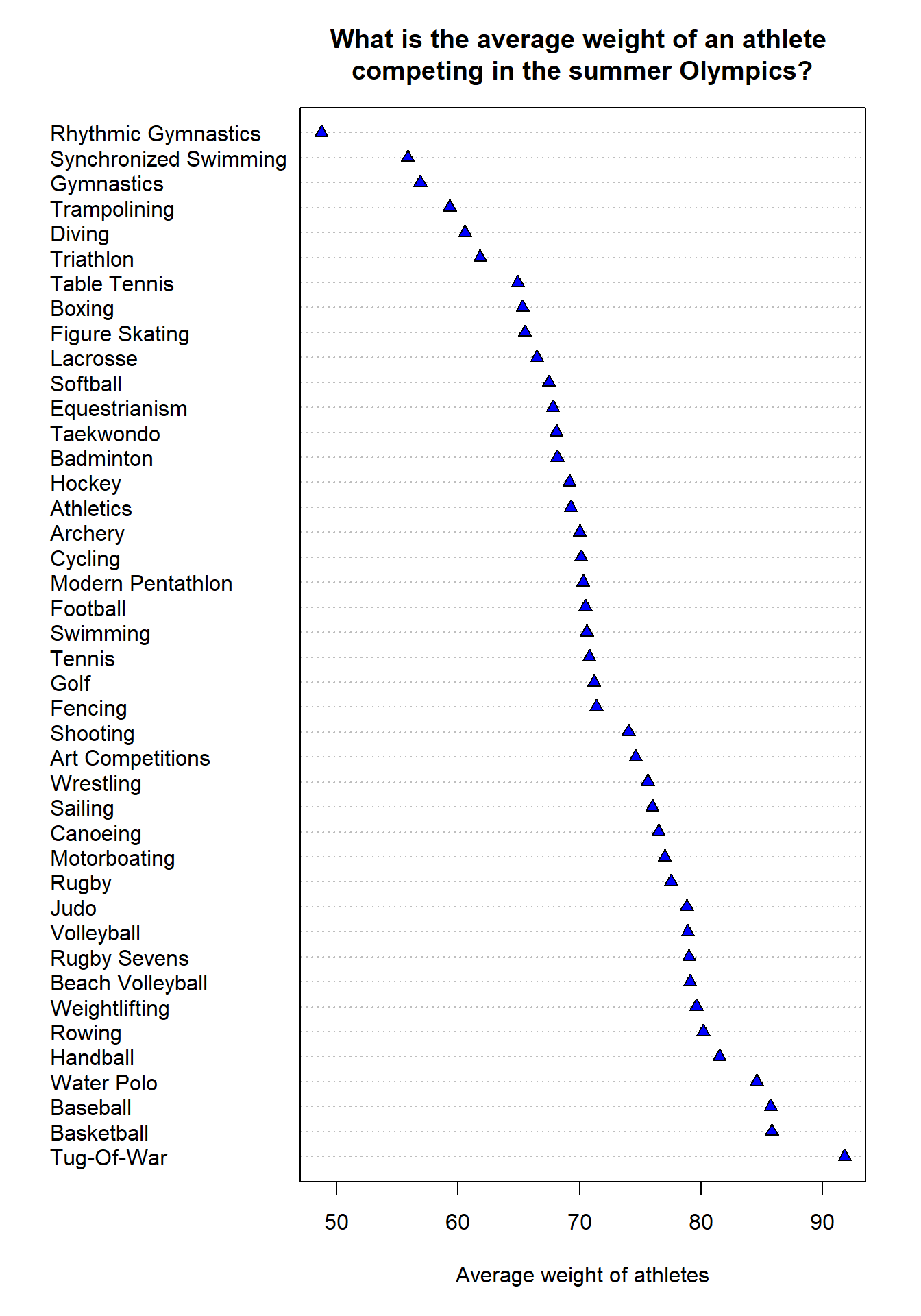
summer\_weights = tapply(summer$Weight, summer$Sport, mean)

summer\_weights = sort(summer\_weights, decreasing = TRUE)

dotchart(summer\_weights, pch = 24, bg = "blue",

xlab="Average weight of athletes",

main = "What is the average weight of an athlete \ncompeting in the summer Olympics?")



This figure becomes much more interesting when compared to the height - for example, athletes competing in Tug-of-War (it was part of the Olympics only between 1900-1920) weren’t the tallest but they seem to weigh the most. I would have expected weightlifters or judo contestants to be in the top 5 based on the average weight but this is not the case.

Looking at the tallest athletes, we can see that basketball players were both leading in terms of height and weight. But volleyball players, on the contrary, are very tall but weigh less.

*# Let's see average weights for winter sports*

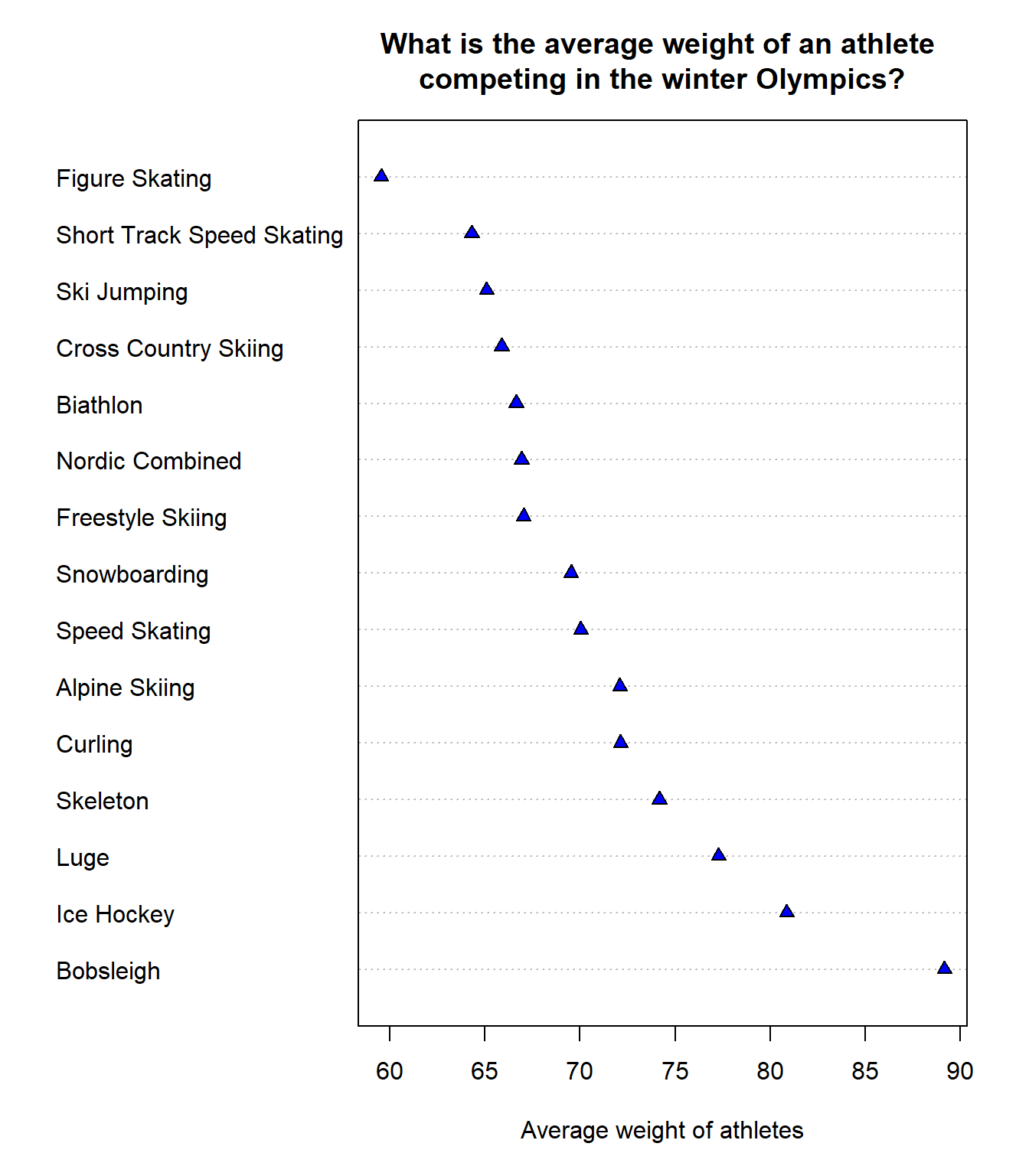
winter\_weights = tapply(winter$Weight, winter$Sport, mean)

winter\_weights = sort(winter\_weights, decreasing = TRUE)

dotchart(winter\_weights, pch = 24, bg = "blue",

xlab="Average weight of athletes",

main = "What is the average weight of an athlete \ncompeting in the winter Olympics?")



The last graph showing the average weight is based on the disciplines played in winter. Here it seems that the top 3 sports with the tallest athletes are also included as the highest average weight.

The minimum weight of an athlete competing in winter games is higher than for the summer games.

**Hypothesis testing, regression, and correlation analysis**

Hypothesis testing is described as a decision-making process that allows evaluating claims about a population (Bluman, 2018). Firstly, I’d like to test something very obvious - are basketball players taller than gymnasts? Those two sports have been on the opposite side of the spectrum in terms of average height when I performed EDA on this dataset so I can set the confidence interval at 99%, as I am certain there is a significant height difference.

df %>%

*# Filter for basketball and gymnastics*

filter(Sport %**in**% c("Basketball", "Gymnastics")) %>%

*# Group by sport*

group\_by(Sport) %>%

*# Get mean and median height*

summarize(mean\_height = mean(Height),

median\_height = median(Height))

## # A tibble: 2 x 3

## Sport mean\_height median\_height

## <chr> <dbl> <dbl>

## 1 Basketball 191. 191

## 2 Gymnastics 163. 164

It can already be seen that there is a big difference between the two disciplines

I will use a t-test with confidence interval of 99% to test the hypotheses that:

*H0 = There is no difference between the height of basketball players and the height of gymnasts*

*H1 = The athletes competing in those two disciplines are characterized by different heights*

*# creating subsets to look at chosen disciplines*

basketball = dplyr::filter(df, Sport == "Basketball")

gymnastics = dplyr::filter(df, Sport == "Gymnastics")

*# Ensuring that my results won't change*

set.seed(123)

*# Taking samples to have equal numbers of observations*

basketball = sample(basketball$Height, 2500, replace = FALSE)

gymnastics = sample(gymnastics$Height, 2500, replace = FALSE)

*# Paired t-test*

t.test(basketball, gymnastics, conf.level = 0.99)

##

## Welch Two Sample t-test

##

## data: basketball and gymnastics

## t = 100.99, df = 4522.2, p-value < 2.2e-16

## alternative hypothesis: true difference in means is not equal to 0

## 99 percent confidence interval:

## 27.83553 29.29327

## sample estimates:

## mean of x mean of y

## 191.0940 162.5296

There have been many athletes that have participated in the Olympics multiple times, so in order to create a sample of heights for unique athletes, I reduced the sample size to 2500 athletes per discipline and created 2 samples.

As large values of the t-scores mean that there is a difference between the two groups, and my p-value has a value very close to 0, I can reject my null hypothesis and state that indeed there is a height difference between the athletes competing in basketball and gymnastics.

As my number of observations is much higher than 30, I could have also used a z-test, as the main difference between the two tests is that a z-test is being used when the standard deviation is known and a t-test is used when the standard deviation is unknown (Bluman, 2018).

I would approach it in the following way:

z.test(basketball, gymnastics, sigma.x = sd(basketball),

sigma.y = sd(gymnastics),

conf.level = 0.99)

##

## Two-sample z-Test

##

## data: basketball and gymnastics

## z = 100.99, p-value < 2.2e-16

## alternative hypothesis: true difference in means is not equal to 0

## 99 percent confidence interval:

## 27.83584 29.29296

## sample estimates:

## mean of x mean of y

## 191.0940 162.5296

The p-value is the same as for the t-test, which confirms that I can reject the null hypothesis and state that the average height of basketball players is higher than the average height of gymnasts. There are slight differences in calculated means and confidence intervals but the outputs of both tests are very similar.

My last question about this dataset is - is there a difference in weight between athletes competing in winter and summer games?

Firstly, I’ll just see the values for the measures of central tendency:

df %>%

group\_by(Season) %>%

summarize(mean\_weight = mean(Weight),

median\_weight = median(Weight))

## # A tibble: 2 x 3

## Season mean\_weight median\_weight

## <chr> <dbl> <dbl>

## 1 Summer 70.7 70

## 2 Winter 70.8 70

It seems like there isn’t much of a difference but I’d like to do the 2 sample t-test and see whether:

*H0 = There is no difference in average weight between the athletes competing in winter and summer sports*

*H1 = There average weight of athletes competing in summer Olympics is different from the average weight of athletes competing in the winter games*

*# T-test*

t.test(summer$Weight, winter$Weight)

##

## Welch Two Sample t-test

##

## data: summer$Weight and winter$Weight

## t = -1.1681, df = 69737, p-value = 0.2428

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## -0.22311222 0.05647813

## sample estimates:

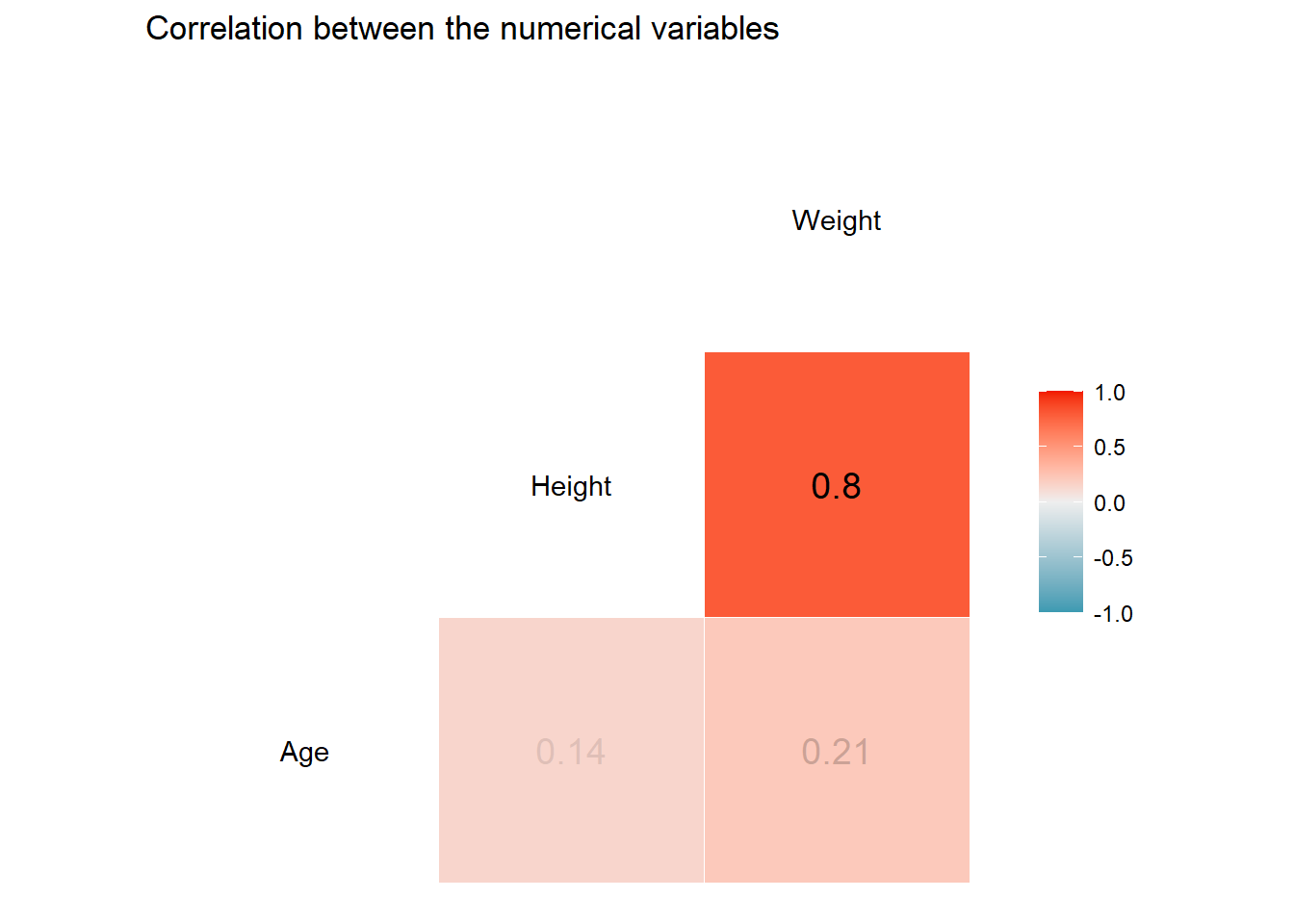
## mean of x mean of y

## 70.67210 70.75542

In this case, I can’t reject the null hypothesis as the p-value is equal to 0.2. Even if I changed my confidence interval to 90%, 0.2428 would still be higher than alpha = 0.1. Thus, I can state that there’s no difference in average weight for athletes competing in winter and summer games.

Let’s see the values of correlation between the 3 numerical variables included in this dataset:

ggcorr(df[4:6], label = TRUE, label\_size = 5, label\_round = 2, label\_alpha = TRUE) + labs(title = "Correlation between the numerical variables")



Correlation is represented by R-Squared (R2), also called a coefficient of determination. It ranges from 0 to 1 (or 0 to 100%) and explains how strong a linear relationship between two variables is. A weak correlation is shown graphically as the points being relatively far away from the best-fit line. The R2 in that case would be represented by a small number. A strong correlation, on the other hand, is one in which the points lie very close to the best-fit line, and the R2 would be closer to 1 (Jones, 2020)(Bluman, 2018).

Here, we can see that the height and weight have a correlation of 0.8, so I can build a linear model to see what’s the relationship between these variables.

ggplot(df, aes(x = Height, y = Weight)) +

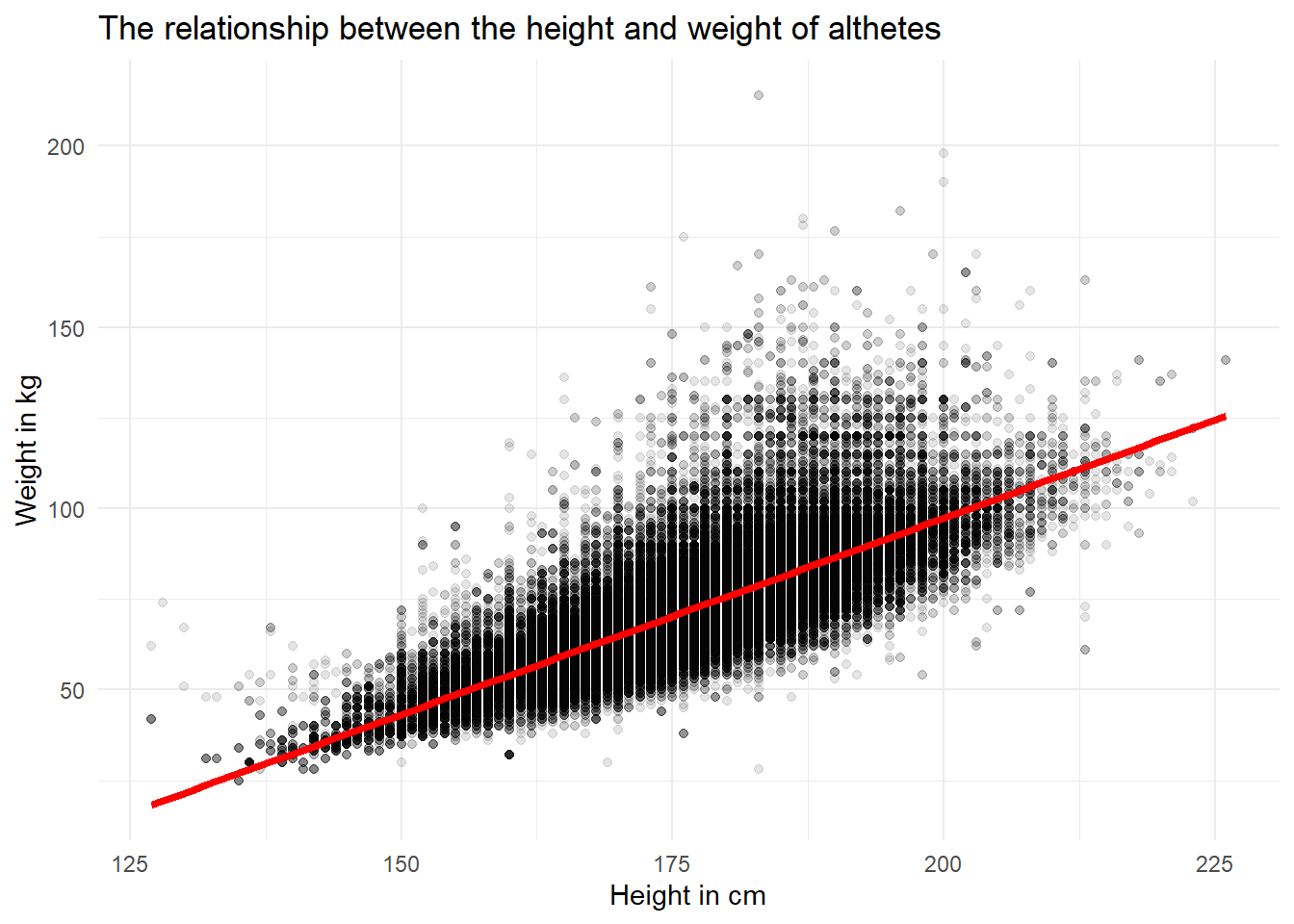
labs(title = "The relationship between the height and weight of althetes", x = "Height in cm", y = "Weight in kg") +

geom\_point(alpha = 0.1) +

geom\_smooth(method = "lm", col = "red", size = 1.5) +

theme\_minimal()

## `geom\_smooth()` using formula 'y ~ x'



This graph is not perfect, even though I tried decreasing the opacity of the points and increasing the thickness on the line to show the trend better. However, we can still observe a positive trend.

I decided to use weight as my response variable, as it made more sense that when the height increases, weight of a person should also increase (the other way around would mean that every fat person should be tall, as with the increase of their weight, their height would also start increasing).

model1 = lm(Weight ~ Height, data = df)

summary(model1)

##

## Call:

## lm(formula = Weight ~ Height, data = df)

##

## Residuals:

## Min 1Q Median 3Q Max

## -50.951 -5.285 -0.868 3.885 135.049

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -119.26786 0.31808 -375.0 <2e-16 \*\*\*

## Height 1.08316 0.00181 598.3 <2e-16 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 8.669 on 206163 degrees of freedom

## Multiple R-squared: 0.6345, Adjusted R-squared: 0.6345

## F-statistic: 3.579e+05 on 1 and 206163 DF, p-value: < 2.2e-16

The regression line equation is represented by **Weight = -119.27 + 1.08 \* height**

This means that this model would be very bad at predicting the weight of the shortest athletes (the shortest athletes in this dataset were 127cm tall). The predicted value for that height would be 18.29 kg, which is definitely too low for an adult. The real weights for those athletes were 42 kg for a female and 62 for a male.

The proportion of the variation in the dependent variable that is explained by the regression line and the independent variable, expressed by R-squared, is equal to 0.63 (or 63% as it is usually expressed by a percentage). That means that 37% of the variation is unexplained. Sometimes a high correlation between the two variables means that it is due to a third variable (Bluman, 2018). So it is possible that a discipline in which the athletes are competing affects their weight, as I know from the exploratory data analysis, there have been big differences between average height and weight between the disciplines. Additionally, there are differences in those measures between the genders.

I can see from the p-values that the results are statistically significant at any confidence interval. The Standard Error is indicating how much the coefficient estimates vary (on average) from the actual average value of the response variable. Ideally, the value of standard error should be lower than its coefficients (Rego, 2015).

I can check some values predicted by my linear model:

predictions = data.frame(Height = seq(150, 200, length = 15)) *# create 15 heights between 150 - 200 cm*

predictions = predictions %>% add\_predictions(model1) *# add predicted values*

*# Creating a function that calculates Body Mass Index*

BMI = **function**(height,weight){

**return**(weight/(height/100)^2)

}

predictions$bmi = BMI(predictions$Height, predictions$pred)

round(predictions,2)

## Height pred bmi

## 1 150.00 43.21 19.20

## 2 153.57 47.07 19.96

## 3 157.14 50.94 20.63

## 4 160.71 54.81 21.22

## 5 164.29 58.68 21.74

## 6 167.86 62.55 22.20

## 7 171.43 66.42 22.60

## 8 175.00 70.29 22.95

## 9 178.57 74.15 23.25

## 10 182.14 78.02 23.52

## 11 185.71 81.89 23.74

## 12 189.29 85.76 23.94

## 13 192.86 89.63 24.10

## 14 196.43 93.50 24.23

## 15 200.00 97.36 24.34

It has definitely been trained on a data for athletes - those are mostly weight for fit and healthy people. I have chosen the body mass index (BMI) as an indicator and according to CDC, the healthy range is between 18.5 to <25. All of my predicted values fit within that range.

As the last part of this task, I will check whether adding categorical variables would improve the linear regression model:

*# Multiple Linear Regression*

model2 <- lm(Weight ~ Height + Sex + Sport, data = df)

summary(model2)

##

## Call:

## lm(formula = Weight ~ Height + Sex + Sport, data = df)

##

## Residuals:

## Min 1Q Median 3Q Max

## -50.924 -4.296 -0.536 3.387 125.174

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -1.053e+02 3.910e-01 -269.210 < 2e-16 \*\*\*

## Height 1.010e+00 2.271e-03 444.561 < 2e-16 \*\*\*

## SexM 3.664e+00 4.512e-02 81.202 < 2e-16 \*\*\*

## SportArchery -1.589e+00 1.966e-01 -8.083 6.33e-16 \*\*\*

## SportArt Competitions 1.718e+00 1.418e+00 1.211 0.22583

## SportAthletics -5.861e+00 1.031e-01 -56.866 < 2e-16 \*\*\*

## SportBadminton -4.355e+00 2.237e-01 -19.469 < 2e-16 \*\*\*

## SportBaseball 2.956e+00 2.747e-01 10.762 < 2e-16 \*\*\*

## SportBasketball -4.465e+00 1.600e-01 -27.903 < 2e-16 \*\*\*

## SportBeach Volleyball -5.506e+00 3.377e-01 -16.307 < 2e-16 \*\*\*

## SportBiathlon -6.066e+00 1.450e-01 -41.848 < 2e-16 \*\*\*

## SportBobsleigh 7.638e+00 1.858e-01 41.097 < 2e-16 \*\*\*

## SportBoxing -7.572e+00 1.486e-01 -50.974 < 2e-16 \*\*\*

## SportCanoeing -1.338e+00 1.380e-01 -9.696 < 2e-16 \*\*\*

## SportCross Country Skiing -5.954e+00 1.277e-01 -46.620 < 2e-16 \*\*\*

## SportCurling -2.769e-01 3.816e-01 -0.726 0.46811

## SportCycling -5.553e+00 1.271e-01 -43.688 < 2e-16 \*\*\*

## SportDiving -4.337e+00 1.922e-01 -22.569 < 2e-16 \*\*\*

## SportEquestrianism -5.749e+00 1.457e-01 -39.468 < 2e-16 \*\*\*

## SportFencing -5.024e+00 1.323e-01 -37.988 < 2e-16 \*\*\*

## SportFigure Skating -6.762e+00 2.146e-01 -31.515 < 2e-16 \*\*\*

## SportFootball -4.242e+00 1.467e-01 -28.911 < 2e-16 \*\*\*

## SportFreestyle Skiing -2.250e+00 2.651e-01 -8.489 < 2e-16 \*\*\*

## SportGolf -1.066e+00 7.265e-01 -1.467 0.14234

## SportGymnastics -4.427e+00 1.119e-01 -39.575 < 2e-16 \*\*\*

## SportHandball -7.349e-01 1.620e-01 -4.537 5.71e-06 \*\*\*

## SportHockey -3.102e+00 1.484e-01 -20.903 < 2e-16 \*\*\*

## SportIce Hockey 2.391e+00 1.457e-01 16.411 < 2e-16 \*\*\*

## SportJudo 5.659e+00 1.598e-01 35.414 < 2e-16 \*\*\*

## SportLacrosse -7.580e+00 5.295e+00 -1.432 0.15228

## SportLuge 1.456e+00 2.227e-01 6.538 6.26e-11 \*\*\*

## SportModern Pentathlon -7.443e+00 2.313e-01 -32.177 < 2e-16 \*\*\*

## SportMotorboating -4.147e+00 7.487e+00 -0.554 0.57965

## SportNordic Combined -9.723e+00 2.485e-01 -39.131 < 2e-16 \*\*\*

## SportRhythmic Gymnastics -1.539e+01 3.170e-01 -48.550 < 2e-16 \*\*\*

## SportRowing -3.277e+00 1.285e-01 -25.498 < 2e-16 \*\*\*

## SportRugby 3.790e+00 1.370e+00 2.766 0.00567 \*\*

## SportRugby Sevens 5.352e+00 4.446e-01 12.038 < 2e-16 \*\*\*

## SportSailing -1.754e+00 1.431e-01 -12.257 < 2e-16 \*\*\*

## SportShooting 1.260e+00 1.290e-01 9.763 < 2e-16 \*\*\*

## SportShort Track Speed Skating -4.020e+00 2.158e-01 -18.628 < 2e-16 \*\*\*

## SportSkeleton -7.898e-01 5.753e-01 -1.373 0.16984

## SportSki Jumping -1.147e+01 1.939e-01 -59.130 < 2e-16 \*\*\*

## SportSnowboarding -1.953e+00 2.637e-01 -7.406 1.31e-13 \*\*\*

## SportSoftball 1.672e+00 3.694e-01 4.525 6.04e-06 \*\*\*

## SportSpeed Skating -2.527e+00 1.470e-01 -17.187 < 2e-16 \*\*\*

## SportSwimming -6.479e+00 1.096e-01 -59.121 < 2e-16 \*\*\*

## SportSynchronized Swimming -8.996e+00 2.745e-01 -32.768 < 2e-16 \*\*\*

## SportTable Tennis -4.626e+00 2.005e-01 -23.070 < 2e-16 \*\*\*

## SportTaekwondo -6.985e+00 3.209e-01 -21.765 < 2e-16 \*\*\*

## SportTennis -6.482e+00 1.923e-01 -33.704 < 2e-16 \*\*\*

## SportTrampolining -5.405e+00 6.207e-01 -8.709 < 2e-16 \*\*\*

## SportTriathlon -1.008e+01 3.398e-01 -29.671 < 2e-16 \*\*\*

## SportTug-Of-War 9.088e+00 1.677e+00 5.420 5.97e-08 \*\*\*

## SportVolleyball -6.612e+00 1.643e-01 -40.256 < 2e-16 \*\*\*

## SportWater Polo 3.673e-03 1.731e-01 0.021 0.98307

## SportWeightlifting 1.235e+01 1.679e-01 73.586 < 2e-16 \*\*\*

## SportWrestling 3.334e+00 1.414e-01 23.588 < 2e-16 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 7.487 on 206107 degrees of freedom

## Multiple R-squared: 0.7275, Adjusted R-squared: 0.7274

## F-statistic: 9655 on 57 and 206107 DF, p-value: < 2.2e-16

Because I added categorical variables, the output is a long list of intercepts, standard errors, t statistics, and p-values for each category. I have added gender and sport as explanatory variables, as it became clear from the exploratory data analysis that weight and height differ both between genders and disciplines. The first important thing to note is that the R2 indeed improved and now 73% of the variation in the response variable weight is explained.

Looking at only p-values, I can see that they have very different values. Small p-values indicate that it is unlikely to observe a relationship between the predictors and the response variables due to chance. Three stars (or asterisks) next to the values represent a highly significant p-value. Thus, a small p-value for the intercept and the slope indicates that there is a relationship between the variables included in this model (Rego, 2015).

With the confidence interval of 95%, for some of the disciplines, I would have to accept the null hypothesis that there is no statistically significant relationship with the response variable in the model. For, disciplines such as curling, golf, lacrosse, motorboating, skeleton, and water polo the relationship between the dependent and independent variables would not be statistically significant. If I changed the confidence level to 99%, the list of sports would be even longer.

**Exploring the number of medals won by each country**

As the last part of this analysis, I would like to focus on the medals.

*# counting the number of medals awarded to each country (in all 120 years)*

winners = df %>% drop\_na(Medal) %>%

group\_by(NOC, Medal, Year) %>%

summarize(Count = length(Medal))

## `summarise()` has grouped output by 'NOC', 'Medal'. You can override using the `.groups` argument.

*# ordering countries by the total medal count and picking top 10*

top\_10 = winners %>%

group\_by(NOC) %>%

summarize(Total = sum(Count)) %>%

arrange(desc(Total)) %>%

slice(1:10)

*# plot*

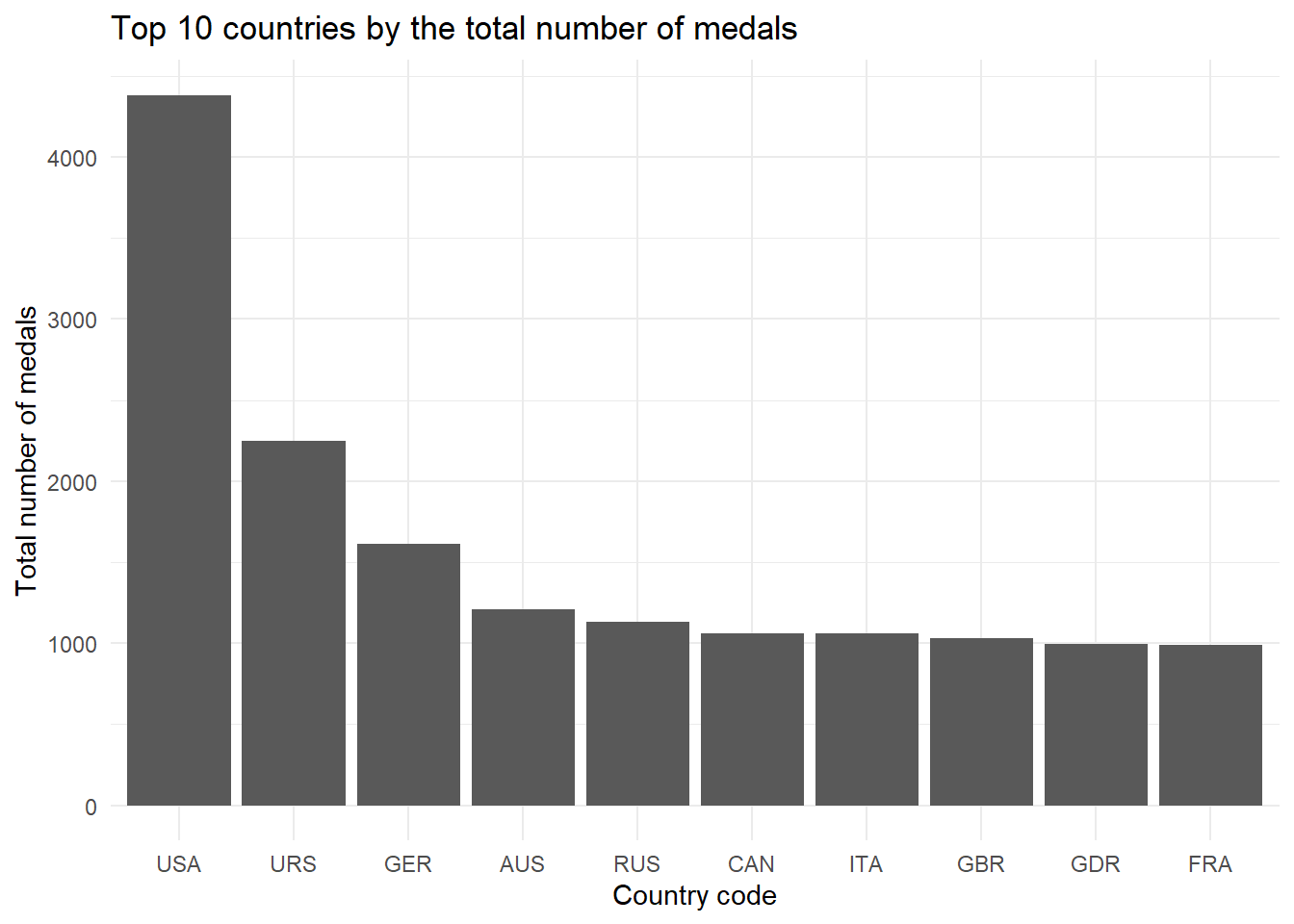
top\_10 %>%

ggplot(aes(reorder(NOC,-Total), Total)) +

geom\_col() +

labs(x= "Country code", y = "Total number of medals", title="Top 10 countries by the total number of medals") +

theme\_minimal()



The top 3 countries that won the most medals are: the United States, the Soviet Union, and Germany. Even when the medals for Soviet Union would be summarized with Russia, the USA would still win the ranking.

*# Let's see what colors of the medals the team US scored over the years*

USA = dplyr::filter(winners, NOC == "USA")

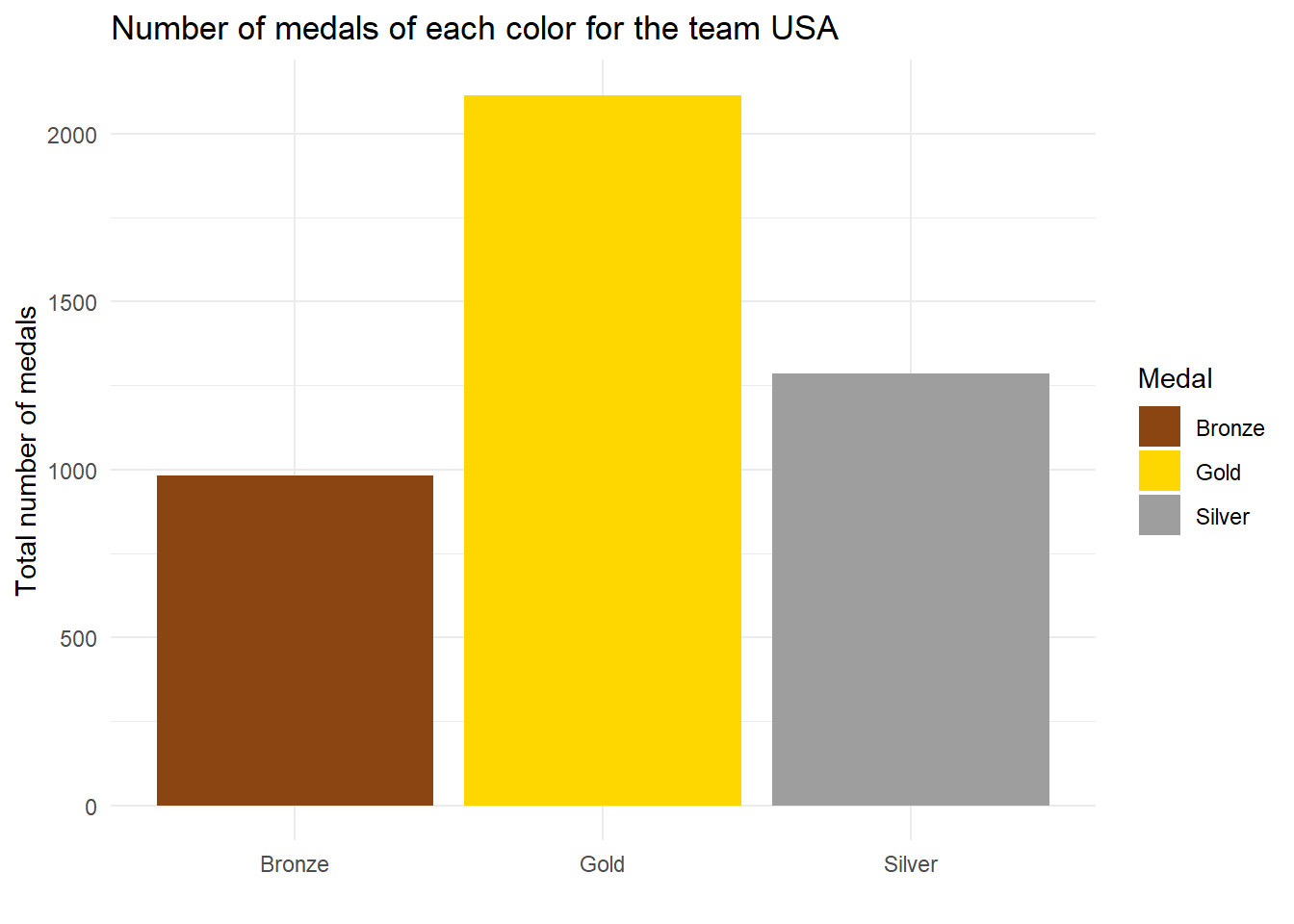
ggplot(USA, aes(x = Medal, y = Count, fill = Medal)) +

geom\_bar(stat="identity") +

theme\_minimal() +

labs(x = " ", y = "Total number of medals", title="Number of medals of each color for the team USA") +

scale\_fill\_manual(values=c("saddlebrown", "gold", "gray62"))



The most medals won by the team USA were gold - many rankings assign different weights to different medals to see how well the country is standing in comparison to their opponents but the United States have also obtained the most medals of the highest value and the fewest bronze medals.

I found out that team USA has been a leader but that made me think about another interesting question - do countries that send more athletes to the Olympics win more medals or can you just have a few very good ones and they will guarantee the success of a particular country?

I will be working on subsets including medals won only in the last winter and last summer games included in this dataset (Rio 2016 and Sochi 2014), as it would not make much sense to sum the numbers across different years.

*# Creating a subset for the 2014 & 2016 Olympics*

Rio = dplyr::filter(olympics, Year == 2016)

Sochi = dplyr::filter(olympics, Year == 2014)

*# Counting the number of athletes per country*

num\_athletes <- Rio %>%

group\_by(NOC) %>%

summarize(count = n\_distinct(Name)) %>%

arrange(desc(count))

num\_athl <- Sochi %>%

group\_by(NOC) %>%

summarize(count = n\_distinct(Name)) %>%

arrange(desc(count))

*# counting the number of medals awarded to each country*

winners = Rio %>% drop\_na(Medal) %>%

group\_by(NOC, Medal, Year) %>%

summarize(Count = length(Medal))

win = Sochi %>% drop\_na(Medal) %>%

group\_by(NOC, Medal, Year) %>%

summarize(Count = length(Medal))

*# ordering countries by the total medal count and picking top 10*

num\_medals = winners %>%

group\_by(NOC) %>%

summarize(Total = sum(Count)) %>%

arrange(desc(Total))

num\_med = winners %>%

group\_by(NOC) %>%

summarize(Total = sum(Count)) %>%

arrange(desc(Total))

*# merging*

last\_summer = left\_join(num\_athletes, num\_medals, by = "NOC")

last\_summer = last\_summer %>% rename(

athletes = count,

medals = Total

)

last\_winter = left\_join(num\_athl, num\_med, by = "NOC")

last\_winter = last\_winter %>% rename(

athletes = count,

medals = Total

)

*# Replacing NAs with zeros*

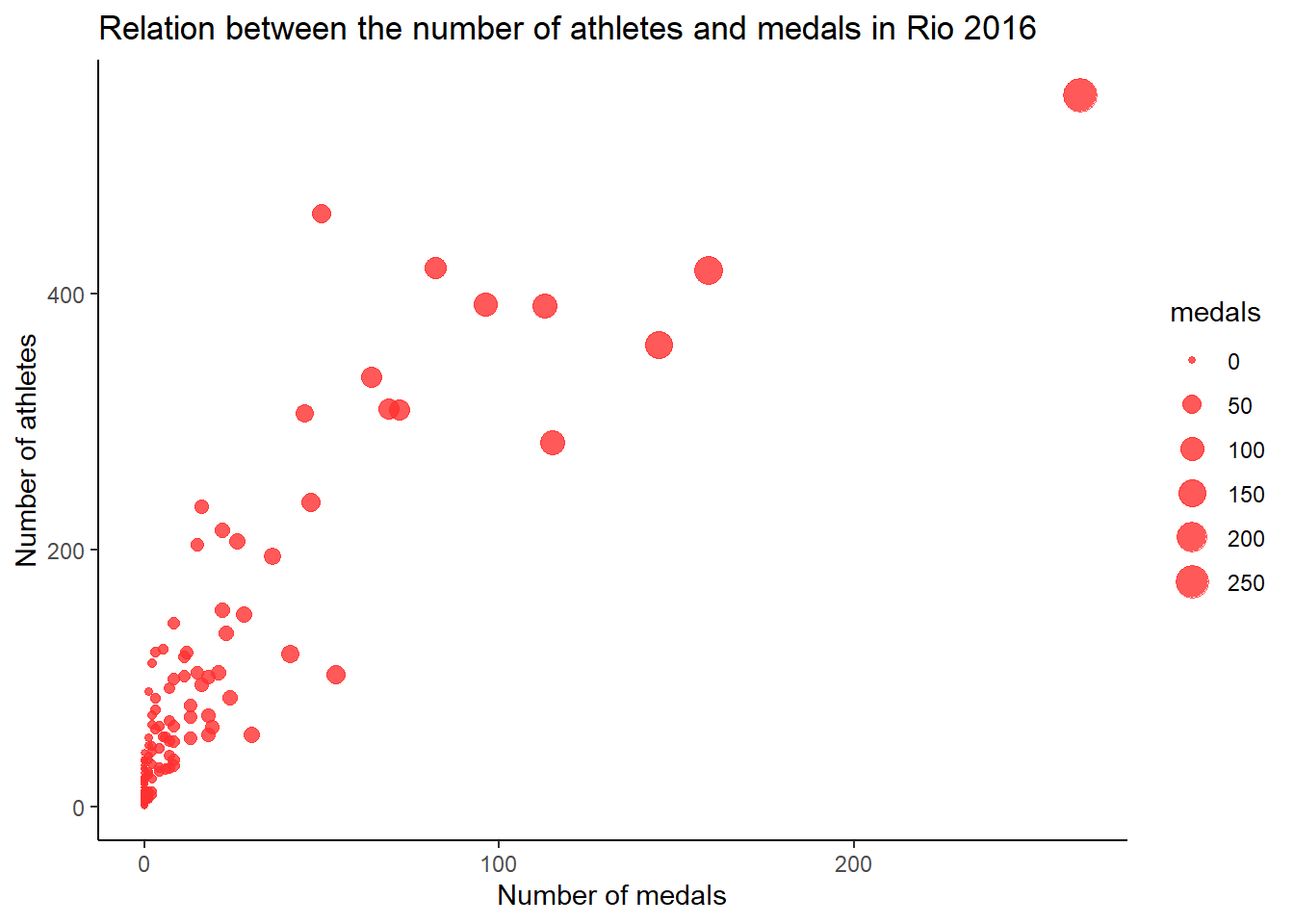
last\_summer = last\_summer %>% mutate\_all(~replace(., is.na(.), 0))

last\_winter = last\_winter %>% mutate\_all(~replace(., is.na(.), 0))

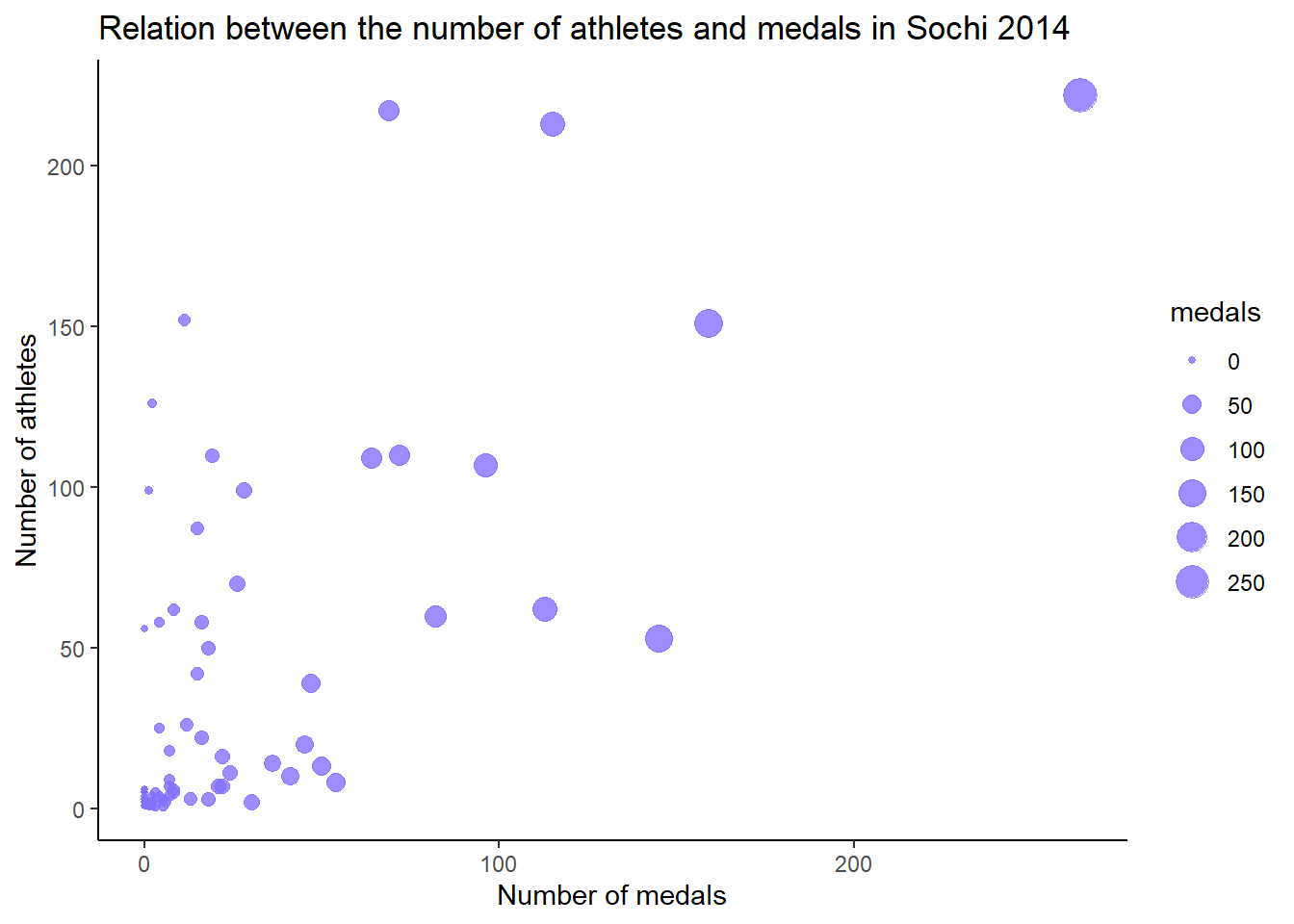
Firstly, I want to visualize the two variables on a scatter plot - my independent variable will be the number of athletes and the dependent variable will be the number of medals (I expect that the number of athletes would affect how many medals the country won).

I made separate plots for the summer and winter games to see whether there are any differences. The size of points is based on the number of medals (the bigger the circle, the more medals a country won).

ggplot(last\_summer, aes(x = medals, y = athletes, size = medals)) + geom\_point(alpha = 0.8, color = "firebrick1") + theme\_classic() + labs(title = "Relation between the number of athletes and medals in Rio 2016", x = "Number of medals", y = "Number of athletes")



ggplot(last\_winter, aes(x = medals, y = athletes, size = medals)) + labs(title = "Relation between the number of athletes and medals in Sochi 2014", x = "Number of medals", y = "Number of athletes") + geom\_point(alpha = 0.8, color = "lightslateblue") + theme\_classic()



Based on the plots, I can conclude that the relationship is more clear for the summer Olympics and I should probably base my model on this subset, as the data points for the winter games in Sochi are more scattered. It seems that there is a positive relationship between the two variables - we can see that as the number of athletes increases, the number of medals increases. Generally, the bigger circles are plotted on the right side of the graphs.

The next step in the regression analysis would be to check the correlation coefficient and determine the significance of the relationship. Determining a regression line when the correlation coefficient is not significant and making predictions would be meaningless (Bluman, 2018). So I’d like to see the correlation for the Rio 2016 subset:

cor(last\_summer$athletes, last\_summer$medals)

## [1] 0.8699587

It seems like there is indeed a positive relationship between the number of athletes per country and the number of medals they won. Now, I can plot a line of best fit (where the sum of squares of the vertical distances between the line and each point is at minimum (Bluman, 2018)) through the data for the summer Olympics in Rio 2016:

ggplot(last\_summer, aes(x = medals, y = athletes)) +

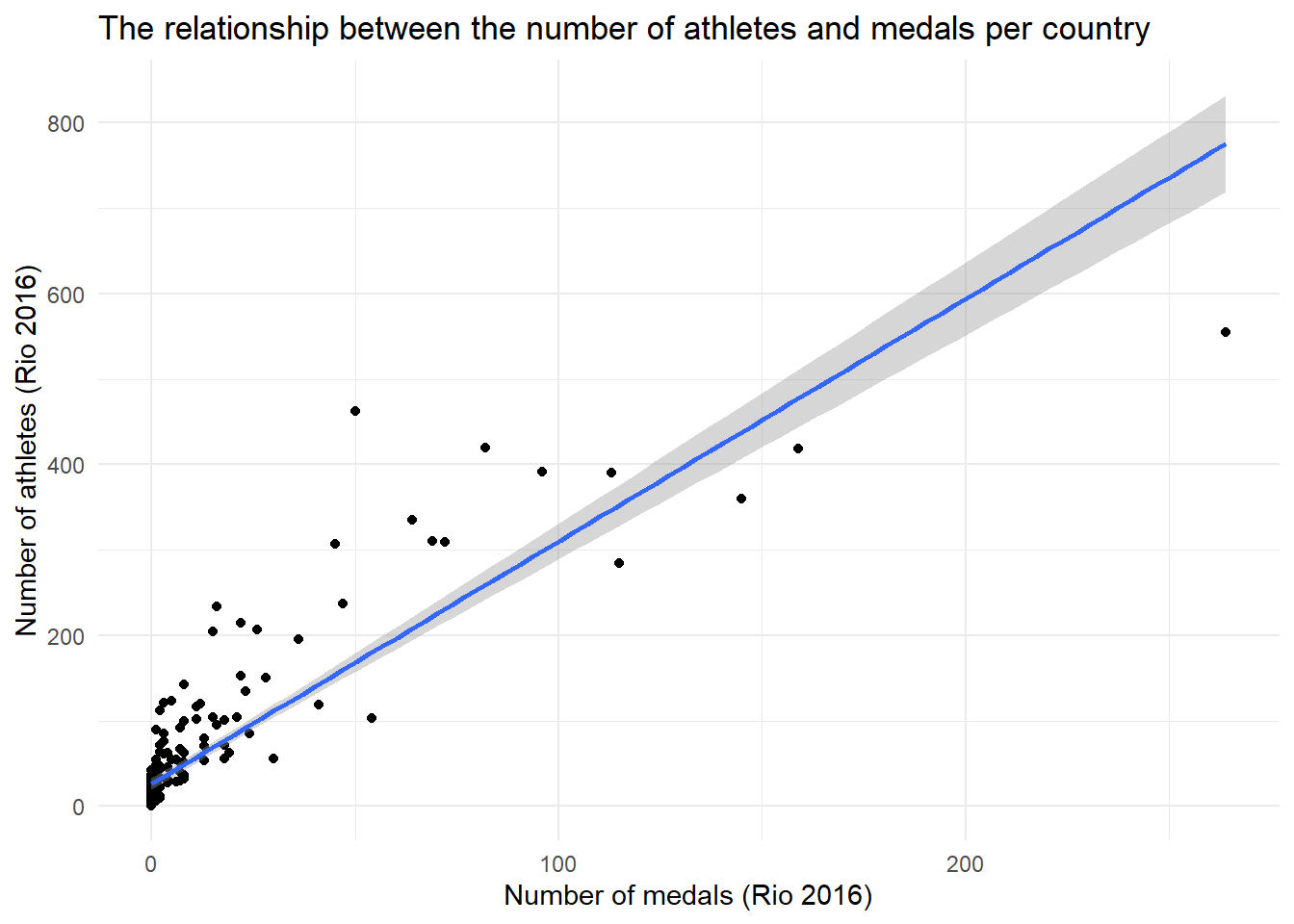
labs(title = "The relationship between the number of athletes and medals per country", x = "Number of medals (Rio 2016)", y = "Number of athletes (Rio 2016)") +

geom\_point() +

geom\_smooth(method = "lm", se = TRUE) + *# showing a straight line and standard error*

theme\_minimal()

## `geom\_smooth()` using formula 'y ~ x'



From this plot, I can see that most points are scattered in the lower left corner, meaning that most countries send fewer than 200 athletes to the Olympics and they come back with very few medals. There is some positive trend in the data - when the number of athletes per country increases, the number of medals also increases.

model3 = lm(medals ~ athletes, data = last\_summer)

summary(model3)

##

## Call:

## lm(formula = medals ~ athletes, data = last\_summer)

##

## Residuals:

## Min 1Q Median 3Q Max

## -68.547 -3.410 2.487 3.488 120.660

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -4.61927 1.14917 -4.02 8.19e-05 \*\*\*

## athletes 0.26659 0.01055 25.26 < 2e-16 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 14.36 on 205 degrees of freedom

## Multiple R-squared: 0.7568, Adjusted R-squared: 0.7556

## F-statistic: 638 on 1 and 205 DF, p-value: < 2.2e-16

So my formula is equal to **number of medals = -4.62 + 0.27 \* number of athletes** which means that when the number of athletes increases by 1, the number of medals would increase by 0.29.

I can already see that this model isn’t perfect for two reasons:

* the intercept is negative, which would mean that if a country has fewer than 18 athletes, it would have a negative number of medals.
* the predicted values would have decimal points and both of the variables are discrete, therefore, they would need to be rounded down to the nearest full number.

The coefficients have highly significant p-values and the R2 shows that 76% of the variation in the number of medals is explained by the number of athletes competing for a particular country.

**CONCLUSIONS**

Based on the analysis of the data about 120 years of Olympic history, I have been able to discover some interesting insights and visually present them. Firstly, the weight and height distribution is not as simple as just assuming that men are taller than women - a female volleyball player will most likely be taller than a male gymnast. So the numerical attributes are more dependent on a discipline than gender. I could also see that there is a big gap in participation between women and men - only half of the athletes included in this dataset are female. According to the International Olympic Committee (IOC), even during the Olympics in Tokyo, only 49% of athletes were female. However, that was the closest to gender equality throughout history. If I was to conduct a full analysis of this dataset, it would also be interesting to see line charts for trends over time, one of which could be a proportion of female athletes.

I have also conducted a series of hypotheses tests to check my assumptions about the weight and height of athletes competing in the Olympic Games. I was able to discover that there’s no difference between the average weight of athletes competing in winter versus athletes competing in the summer, as well as, to confirm that male athletes weight more than female ones, and that basketball players are significantly taller than gymnasts.

I have also discovered what were the top 10 countries with the most medals and analyzed further the number of medals of each color for the winner of that ranking - the United States. This breakdown could have been done for more countries, showing the colors as a stacked bar chart but it would have been difficult to visualize it for multiple countries, as the graphs would have become cluttered.

I have explored linear relationships between the weight of the athletes and their height, as well as, the number of Olympic medals per country and the number of athletes competing for that country. The first model has been improved by adding sports and gender as other explanatory variables, which increased the value of R-squared by 0.09.

The second model predicting the number of medals per country could already explain 76% of the variation in the response variable, based only on one explanatory variable. It could probably be improved after adding additional variables - one idea could be to create a dummy variable checking whether the country in which the Olympics have been hosted in the home country for the team and then check whether competing in front of the spectators from your home country gives any advantage. Another idea could be to combine this data with macroeconomic metrics such as GDP per capita and check whether more wealthy countries bring back home more medals.