QRM II Graded Assignment (2), Period 1 2025

Material by Sjoerd van Alten and Klervie Toczé

Group 21 - Carlijn Calori, Julia Koeleman, Marit Springer & Collin Veenstra

25-09-2025

Introduction

This assignment is to be completed in groups of 3-4. Further, all students in your group need to be assigned to the same R tutorial group (Friday’s tutorial). You can sign yourself up for a group on Canvas. Please do so You can use the Discussion Board in Canvas if you do not have a group yet or if your group is incomplete.

The assignment has 5 parts, and each part corresponds to the course material of that week (with the exclusion of week 6, for which there is no R programming material).

You are supposed to hand in these assignments on Canvas at the following dates:

* **Deadline 1** *Thursday September 25th, at 23:59pm*: you are supposed to hand in weeks 1, 2, and 3 of this assignment. This will determine 18% of your overall course grade
* **Deadline 2** *Thursday October 9th, at 23:59pm*: you are supposed to hand in weeks 4, and 5 of this assignment. This will determine 12% of your overall course grade

The R tutorials (each Friday) will consist of two halves. During the first half, you will discuss the tutorial exercises. These can be downloaded separately from Canvas. During the second half, you can work on this graded assignment within your own group. The purpose is that you find out how to work with R for doing statistical analyses by yourself. The tutorial exercises are meant to teach you basic commands to get you started, but to answer the problem sets in this assignment, you might need to research your own solutions, and use functions and commands not described in the tutorial exercises. Learning how to solve your own research problems is integral part of learning R. When you and your group get stuck on how to approach an exercise, the hierarchy in finding your way is as follows:

* use the concepts from the tutorial exercises;
* use the cheat sheets available on Canvas;
* use Google, YouTube, StackOverflow, or another website;
* ask the teacher.

The use of generative AI is permitted and may result in a grade of 0. See the AI protocol in the course manual for details.

To answer the assignment, you can simply fill out this R markdown document. There are designated places which you can fill with R code. There are also designated spaces for you to answer each question. Often, the structure of an answer will be as follows. First, you type the R code in the designated box. This will show how you analyzed the data to get the answer to the question. Below the box for the R code, you will then summarize your answer to the question, i.e. what are the conclusions that you draw from the data analysis?

When handing in, you are supposed to submit this .Rmd file, and a knitted version of this document. You can knit this document to pdf, word, or html. Knitting to pdf requires you to have a .tex distribution installed on your computer. Knitting to Word requires you to have Word installed.

The exercises are designed such that you should be able to finish the majority of them during the tutorial each week. If you are not able to finish them fully during that time, you are expected to work on it in your own time using the computers on campus or your own device. It is best to meet as a group in-person when working together. If you want to work remotely, github is a good platform to guarantee smooth collaboration. Alternatively, you can email this .Rmd file back and forth to one another as a group, but this is not recommended as it is more cumbersome.

We encourage you to keep your code blocks, printing statements, and final answers, as short as possible. In any case, there is a page limit of 6 pages per week, which encompasses the total length of this document which consists of the questions, your coding lines, and your answers. When your answers to questions of the respective week exceed this page limit, they will not be graded, resulting in zero points.

Each week consists of 1, 2, or 3 subquestions. The total amount of points you can earn per week is 20 points.

# Week 1

1. Find the dataset “movies1.tsv” on Canvas. Describe your data set: How many observations does it have. How many variables are there? How many subjects? What consists of a subject?

movies <- read.table("movies1.tsv", header=TRUE)  
ncol(movies)

## [1] 19

nrow(movies)

## [1] 505

**Your Answer**

We conclude that the data set has 19 variables and 505 subjects. Every subject consists of a unique movie.

1. Which of the following types of variables are present in your data set? (i) nominal; (ii) ordinal; (iii); interval; (iv) ratio. If present, name one example of such a variable present in your data set.

#WRITE YOUR CODE HERE  
str(movies)

## 'data.frame': 505 obs. of 19 variables:  
## $ index : int 1773 2540 1174 3262 4324 214 1737 610 2738 963 ...  
## $ budget : num 2.70e+07 1.50e+07 4.00e+07 0.00 0.00 1.20e+08 2.80e+07 7.00e+07 1.25e+07 4.00e+07 ...  
## $ keywords : chr "fbi island serial killer series of murders" "fire winter santa claus snow storm christmas tree" "police sequel police officer brother-in-law brother-in-law relationship black men" "masseuse thanksgiving party romance mother daughter relationship" ...  
## $ original\_language : chr "en" "en" "en" "en" ...  
## $ title : chr "Mindhunters" "Krampus" "Ride Along 2" "Enough Said" ...  
## $ popularity : num 17.194 31.565 25.136 14.969 0.148 ...  
## $ release\_date : chr "2004-05-07" "2015-11-26" "2016-01-14" "2013-09-18" ...  
## $ revenue : num 2.11e+07 6.15e+07 1.25e+08 2.53e+07 0.00 ...  
## $ runtime : int 106 98 102 93 97 130 108 99 100 99 ...  
## $ vote\_average : num 6.3 5.9 6.1 6.6 6 6.2 5.5 4.4 6 6.2 ...  
## $ vote\_count : int 333 584 555 348 9 597 640 533 210 371 ...  
## $ genre : chr "Thriller" "Comedy" "Comedy" "Drama" ...  
## $ release\_year : int 2004 2015 2016 2013 2006 2000 2016 2014 2014 2009 ...  
## $ release\_month : int 5 11 1 9 10 3 2 1 2 9 ...  
## $ release\_day : int 7 26 14 18 27 15 4 10 14 29 ...  
## $ first\_actor : chr "LL Cool" "Adam Scott" "Kevin Hart" "Julia Louis-Dreyfus" ...  
## $ first\_actor\_gender : chr NA "male" "male" "female" ...  
## $ director\_first\_name: chr "Renny" "Michael" "Tim" "Nicole" ...  
## $ director\_gender : chr "male" "male" "male" "female" ...

**Your Answer:**

Nominal: present, for example genre and title Ordinal: not present

Interval: present, for example release\_year Ratio: present, for example budget and revenue

1. A movie studio wants to know which types of movies give maximal profit. Perform the following steps to provide the movie studio with an analysis which corresponds to their request:
2. Create the variable profits as the revenue of a movie minus its budget. Report its mean, median, maximum, and minimum.
3. Which movie has the highest profits in your data set and how much are these profits. Which movie has the lowest and how much are its profits? If multiple movies have the exact same highest or lowest profits, give only one example.
4. Create a boxplot of the variable profits. Make sure it has an appropriate title, and appropriate titles and labels for the x- and y-axis. Give Q1, Q2, Q3, and Q4. What does this tell you about the nature of making money in the movies industry?
5. Add a new variable to your data set the log of profits. When creating this variable, what happens to movies for which profits is zero or negative? What then happens when you calculate the mean of log of profits?
6. For movies that have a profit of zero or less, replace log of profits with “NA”. What is now the mean of log of profits? Create a boxplot for log of profits, again with an appropriate title, x- and y-axis labels. How does it compare to the boxplot you made under c.)?
7. Create a scatterplot of with the runtime of movies on the x-axis and the average vote of movies on the y-axis. What do you conclude from the scatterplot? Are movies with a longer runtime considered worse or better by the audience, or does the audience not have a preference? Why do you think this is the case?

For each step, you should provide first all the code you used to answer the question and then formulate an answer using full sentences.

*Step a*

movies$profit = movies$revenue - movies$budget  
  
movies %>%  
 summarise(mean(profit),  
 median(profit),  
 min(profit),  
 max(profit))

## mean(profit) median(profit) min(profit) max(profit)  
## 1 63121475 1900000 -9e+07 2550965087

**Your Answer:**

The mean of the variable profit is 63.121.475 euros. The median of the variable profit is 1.900.000 euros. the minimum of the variable profit is -90.000.000 euros the maximum of the variable profit is 2.550.965.087 euros

*Step b*

#WRITE YOUR CODE HERE  
max\_profit = max(movies$profit)  
min\_profit = min(movies$profit)  
  
movies %>%  
 filter(profit == max\_profit) %>%  
 select(title, profit)

## title profit  
## 1 Avatar 2550965087

movies %>%  
 filter(profit == min\_profit) %>%  
 select(title, profit)

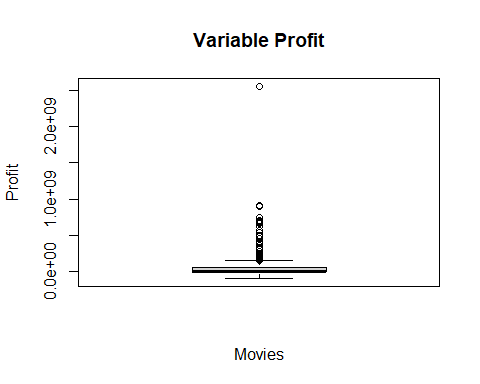
## title profit  
## 1 Mighty Joe Young -9e+07

**Your Answer:**

The movie with the highest profit is Avatar, reporting a profit of 2.550.965.087. The movie with the lowest profit/highest loss is Mighty Joe Young, reporting a profit of -90.000.000

*Step c*

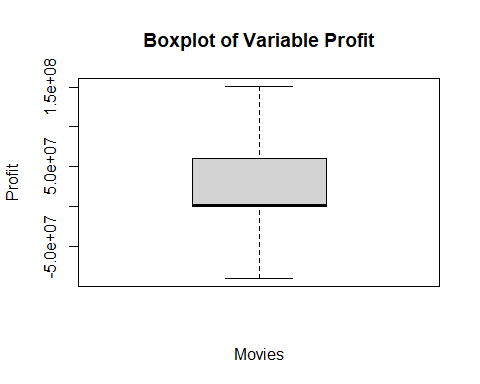
#WRITE YOUR CODE HERE  
boxplot(movies$profit,   
 main="Variable Profit",  
 ylab="Profit",  
 xlab="Movies")



movies %>%  
 summarise(Q1 = quantile(movies$profit, 0.25),  
 Q2 = quantile(movies$profit, 0.50),  
 Q3 = quantile(movies$profit, 0.75),  
 Q4 = max(movies$profit))

## Q1 Q2 Q3 Q4  
## 1 0 1900000 60514050 2550965087

boxplot(movies$profit,  
 main='Boxplot of Variable Profit',  
 ylab="Profit",  
 xlab="Movies",  
 outline = FALSE)



**Your Answer:**

We created two boxplots. The first code is for all variable data in the dataset. The resulting boxplot is very compact with many outliers. In the second code, which was written to create the boxplot, the outliers were removed. We did this using the line #outline = FALSE#. This provides a more specific view of the distribution of variable profit.

We calculated the quantiles based on all the data, including the outliers. These are the quantiles: Q1 = 0 Q2 = 1.900.000 Q3 = 60.514.050 Q4 = 2.550.965.087

This data tells us that the movie industry is a high-risk, high-reward business, where the majority of films earn little, and profitability is driven by a small fraction of massive blockbusters.

*Step d*

#WRITE YOUR CODE HERE  
  
movies$log\_profits = log(movies$profit)

## Warning in log(movies$profit): NaNs produced

movies %>%  
 summarise(mean\_profits = mean(movies$log\_profits))

## mean\_profits  
## 1 NaN

movies %>%  
 summarise(mean\_profits2 = mean(log\_profits[is.finite(log\_profits)], na.rm = TRUE))

## mean\_profits2  
## 1 17.38094

**Your Answer:**

All films without profit are converted to -infinity, because the value of log(0) is not defined for any base of the logarithm All films with a negative profit are converted to NaN, which means that the result of a mathematical calculation is not a valid number. This is because the logarithm is only defined for positive real numbers.

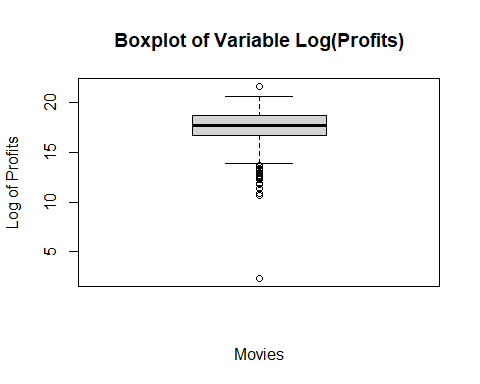
As a result, when calculating the average logarithm of profits, it returns NaN. To obtain an average, films with zero or negative profits should be excluded from the calculation, as their logarithm values are undefined. The average then only reflects films with positive profits. The only problem is that this distorts the result because loss-making or break-even films are not taken into account.

*Step e*

#WRITE YOUR CODE HERE  
movies$log\_profits[movies$log\_profits<=0]<- NA  
movies$log\_profits[is.nan(movies$log\_profits)] <- NA  
  
mean(movies$log\_profits, na.rm = TRUE)

## [1] 17.38094

boxplot(movies$log\_profits,  
 main = "Boxplot of Variable Log(Profits)",  
 xlab = "Movies",  
 ylab = "Log of Profits")



**Your Answer:** The log scale compresses extremely large profits and expands smaller ones, which places the values on a more manageable scale. Therefore the boxplot provides more information than in c. Additionally, all negative values have been removed from the data, so only positive values are visible in the boxplot. However, there are still outliers, which can be seen as the points below and above the boxplot.

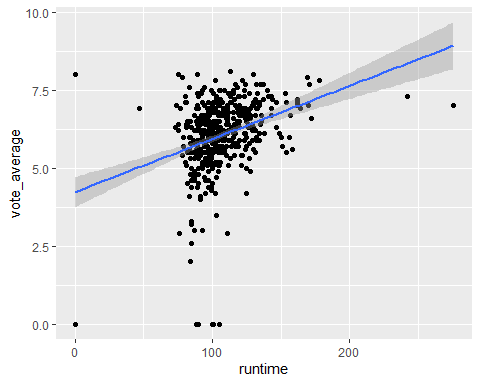
*Step f*

#WRITE YOUR CODE HERE  
ggplot(movies, aes(x=runtime, y=vote\_average))+  
 geom\_point()+  
 geom\_smooth(method="lm", se=T)

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

## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_smooth()`).

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`geom\_point()`).



**Your Answer:**

The average lies around the 100 minutes, movies with a longer runtime are considered better. This suggests that viewers may perceive longer movies as more developed or higher quality.

# Week 2

1 Is your dataset movies1.tsv the full population, or is it a sample of a larger population? If the latter, how would you describe the full population?

#WRITE YOUR CODE HERE  
head(movies$index, n = 10)

## [1] 1773 2540 1174 3262 4324 214 1737 610 2738 963

nrow(movies)

## [1] 505

**Your Answer:**

These numbers are the first ten index numbers of the responders of this data set. As you can see, many numbers are missing. This shows that this is an sample and not an entire population. There are 505 subjects (movies), while the index is showing there are many more. Furthermore, you can logically reason that more than 505 films have been made worldwide.

2 a. For which actor in your data set do you observe the most movies? b. What is the average revenue of the movie in which this actor plays and does the revenue lie above or below the revenue of an average movie according to your data set?   
c. How trustworthy do you consider your conclusion to answer 2b? Use the term “law of large numbers” in your explanation.

*step a*

#WRITE YOUR CODE HERE  
actor\_counts <- sort(table(movies$first\_actor), decreasing = TRUE)  
head(actor\_counts, 1)

##   
## Bruce Willis   
## 7

**Your Answer:**

Bruce Willis is the actor who plays in the most movies. He plays in 7 movies.

*step b*

#WRITE YOUR CODE HERE  
movies %>%   
 filter(first\_actor=="Bruce Willis") %>%   
 summarise(avg\_revenue\_Bruce=mean(revenue, na.rm=T))

## avg\_revenue\_Bruce  
## 1 116280090

movies %>%  
 summarise(avg\_revenue\_data = mean(movies$revenue))

## avg\_revenue\_data  
## 1 94815482

**Your Answer:**

The average revenue of the films of Bruce Willis is 116.280.090. The average revenue of all the films is 94.815.482. The average revenue of the films of Bruce Willis lies above the average revenue of all the films.

*step c*

**Your Answer:** The conclusion in 2b is not very trustworthy, because it is based only on the subset of movies with Bruce Willis. According to the law of large numbers, the reliability of an average increases as the sample size grows. Since the number of Bruce Willis films in the dataset is much smaller (7) than the total number of films (505), his average revenue may not accurately reflect the true expectation and could be influenced by a few exceptionally high-grossing films.

3 For this question, you will assume that your data set is the full population. a. Recode profits such that it is expressed in millions. What is the variance of the variable profits (in millions) in your data set? b. Create a new data set, called movies\_sample. Make sure that it is a random sample of your data set of 25 movies. What is the variance of profits in this random sample? How does it compare to the variance of profits in 2a? c. In a for loop, create 100 different samples of 25 movies, as in b, and estimate the variance within each sample. Save the variance of each sample in a vector called sample\_vars. So the first position of the vector would have the variance of the first sample, the second position the variance of the second sample, etc. Print the start of this vector. d. Summarize and make a histogram of sample\_vars. What is the mean, standard deviation and shape of its distribution? e. In your opinion, is a sample of 25 movies sufficient to get a reliable estimate of the population variance of profits, using the sample variance? Explain?

*step a*

#WRITE YOUR CODE HERE  
  
movies$profit = movies$profit/1000000  
variance = var(movies$profit)

**Your Answer:**

The variance of of the variable profits (in millions) is 30463,13

*step b*

#WRITE YOUR CODE HERE  
  
set.seed(123)  
  
movies\_sample <- movies[sample(nrow(movies), 25), ]  
variance\_sample = var(movies\_sample$profit)

**Your Answer:**

The variance of profits in the sample of 25 movies is 5156 million, which is much lower than the variance of the full dataset. When we computed a sample multiple times, we saw a lot of changes in the variance. Sometimes there were no outliers which made the variance very low, but sometimes the variance was higher than the population variance. Normally we would expect the population variance to be lower, because it it a more precise variable. But in this case the sample chooses movies with less/no outliers, causing the variable to be lower.

*step c*

#WRITE YOUR CODE HERE  
  
library(dplyr)  
  
set.seed(123)  
  
sample\_vars <- c()  
for (i in 1:100) {  
 sample\_data <- sample(movies$profit, size = 25, replace = TRUE)  
 sample\_vars[i] <- var(sample\_data)  
}  
  
print(sample\_vars)

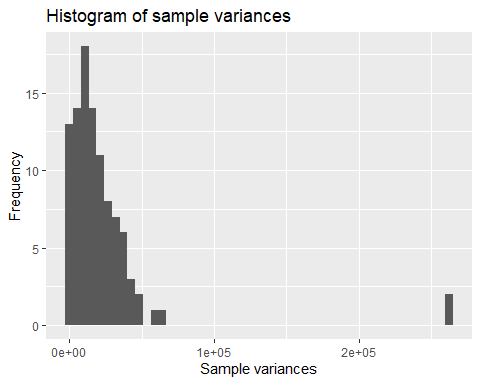
## [1] 5427.987 2599.269 1786.823 4188.200 19863.914 2262.348  
## [7] 3726.957 14556.274 21766.930 6832.605 1087.340 65016.009  
## [13] 47405.135 26631.279 30006.369 24206.674 36152.296 8018.320  
## [19] 49524.980 3851.321 20117.246 26119.756 23950.476 10173.782  
## [25] 36207.858 10970.705 1781.194 19492.944 34422.145 18251.876  
## [31] 2517.222 5662.617 13437.085 18254.151 1927.698 21495.217  
## [37] 1199.210 11642.164 1076.289 20104.961 16522.001 13616.902  
## [43] 259978.030 2494.030 5680.998 23620.087 21488.611 27307.159  
## [49] 39659.398 10125.730 41241.568 34881.842 42593.662 30249.814  
## [55] 11788.002 2526.887 8144.473 3685.787 14342.990 20354.465  
## [61] 10290.612 27415.819 9400.046 7981.578 25090.506 16657.474  
## [67] 36231.807 4069.141 14573.631 11745.178 17524.721 10498.433  
## [73] 2284.862 11283.803 40114.139 263450.409 4958.309 11272.150  
## [79] 12774.466 8603.828 23926.725 7253.008 14657.673 25011.299  
## [85] 30413.328 11457.522 10852.733 33606.377 60474.045 41136.172  
## [91] 10983.993 2295.091 13451.089 8751.209 16105.003 7259.011  
## [97] 29709.776 14968.661 27246.490 34187.963

**Your Answer:**

Above are the profit variances of 100 different samples.

*step d*

#WRITE YOUR CODE HERE  
dfsample\_vars = as.data.frame(sample\_vars)  
  
ggplot(dfsample\_vars, aes(x = sample\_vars)) +  
 geom\_histogram(bins = 50) +  
 labs(title = "Histogram of sample variances", x = "Sample variances", y = "Frequency")



dfsample\_vars %>%  
 summarise(mean\_sample = mean(sample\_vars),  
 sd\_sample = sd(sample\_vars))

## mean\_sample sd\_sample  
## 1 22739.86 36935.63

library(e1071)

## Warning: package 'e1071' was built under R version 4.5.1

skewness\_sample = skewness(sample\_vars, na.rm=T)  
kurtosis\_sample = kurtosis(sample\_vars, na.rm=T)  
print(skewness\_sample)

## [1] 5.414064

print(kurtosis\_sample)

## [1] 32.11739

**Your Answer:**

The mean of the sample variances is 22739,86. The standard deviation of the sample variances is 36935,63. The shape of the distribution is right-skewed, because of the positive skewness value. The kurtosis value is very high, because of the heavy outlier.

*step e*

**Your Answer:**

A sample of 25 movies is relatively small for estimating the population variance of profits. The variance is very sensitive to extreme values, and as seen in the histogram, movie profits are highly skewed with large outliers. According to the law of large numbers, larger samples are needed for the sample variance to converge to the true population variance.

# Week 3

For the next part of the assignment, assume that the movies in your data frame are a random sample of a larger population of movies.

1 a. Create a new data set that only includes movies that are of the genre “Thriller”. For these thriller movies, give a 99 percent confidence interval for the variable *runtime*. Interpret the result. b. Now, assume that the variance of *runtime* amongst thriller movies in your data is exactly the same as the variance of *runtime* in the population. Under this assumption, give a 99 percent confidence interval for the variable *runtime* among thriller movies. Interpret the result. Is you confidence interval wider or less wide than the one you found under question 1a? Why is that the case?

*step a*

#WRITE YOUR CODE HERE  
  
thriller\_movies <- subset(movies, grepl("Thriller", genre, ignore.case = FALSE))  
  
t\_result = t.test(thriller\_movies$runtime, conf.level = 0.99)  
conf\_interval = t\_result$conf.int  
print(conf\_interval)

## [1] 99.94343 111.01042  
## attr(,"conf.level")  
## [1] 0.99

**Your Answer:**

We are 99% confident that the true mean runtime of all thriller movies in the population falls between 99.94 and 111.01 minutes.

*step b*

#WRITE YOUR CODE HERE  
  
mean\_runtime = mean(thriller\_movies$runtime)  
sd\_runtime = sd(thriller\_movies$runtime)  
n = length(thriller\_movies$runtime)  
  
sd\_population = sd\_runtime  
  
alpha <- 0.01  
z = qnorm(1-alpha/2)  
error\_margin = z\*(sd\_population/sqrt(n))  
LOW = mean\_runtime - error\_margin  
UP = mean\_runtime + error\_margin  
  
print(c(LOW, UP))

## [1] 100.1081 110.8457

**Your Answer:**

We are 99% confident that the true mean runtime of thriller movies in the entire population falls between 100.11 and 110.85 minutes.

2 a. Using an appropriate five-step procedure, set up a test for the null hypothesis that the variance of runtime equals . Clearly state your null hypothesis, alternative hypothesis your test statistic, your critical value, and your conclusion. b. For the validity of your test in 2a, what assumption about the distribution of revenue needs to hold? Make an appropriate plot to test this assumption. What do you conclude?

*step a*

#WRITE YOUR CODE HERE  
s2 = var(movies$runtime, na.rm = TRUE)  
n = sum(!is.na(movies$runtime))  
  
sigma2 <- 500  
  
test\_statistic <- (n - 1) \* s2 / sigma2  
  
# Print test statistic  
cat("test statistic:", round(test\_statistic, 2), "\n")

## test statistic: 486

alpha2 <- 0.05  
lower <- qchisq(alpha / 2, df = n - 1)  
upper <- qchisq(1 - alpha / 2, df = n - 1)  
  
cat("Critical values:\n")

## Critical values:

cat("Lower:", round(lower, 2), "\n")

## Lower: 425.06

cat("Upper:", round(upper, 2), "\n")

## Upper: 588.45  
if (test\_statistic < lower || test\_statistic > upper) {  
 cat("Reject the null hypothesis.\n")  
} else {  
 cat("Do not reject the null hypothesis.\n")  
}

## Do not reject the null hypothesis.

**Your Answer:**

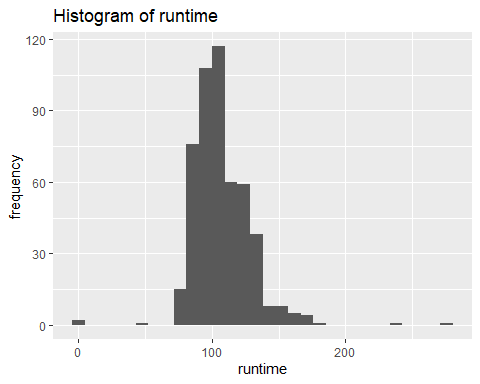
The null hypothesis (H0) is equal to 500. The alternative hypothesis (H1) is NOT equal to 500. We computed the test statistic and the critical values. Our conclusion is that we do not reject the null hypothesis.

*step b*

#WRITE YOUR CODE HERE  
ggplot(movies, aes(x = runtime))+  
 geom\_histogram()+  
 labs(title = "Histogram of runtime",  
 y = "frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

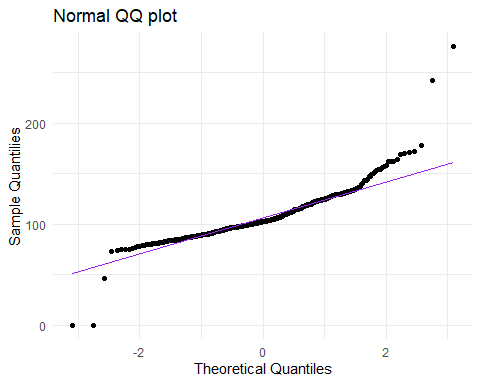
## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_bin()`).



ggplot(movies, aes(sample = runtime)) +  
 stat\_qq() +  
 stat\_qq\_line(color = "purple") +  
 labs(title = "Normal QQ plot",  
 x = "Theoretical Quantiles",  
 y = "Sample Quantilies") +  
 theme\_minimal()

## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



**Your Answer:**

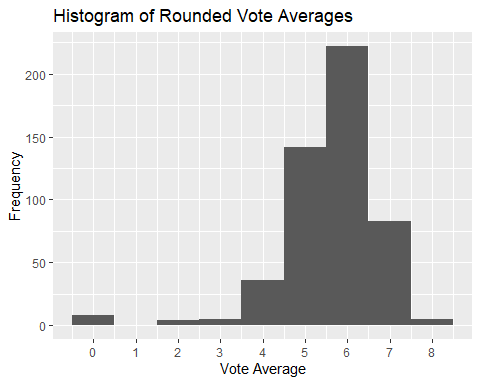
\*I think there is a typo in question 2b, because for 2a we are doing hypothesis testing for the runtime. It would be weird to reason about assumptions of revenue, which has nothing to do with the runtime.

The assumption for the chi-square variance test is that the runtime variable is normally distributed. The histogram of runtime shows a concentration around 90–120 minutes, but the distribution is slightly right-skewed and has a few extreme outliers at both short and very long runtimes. The Q-Q plot also shows clear deviations from the diagonal line, especially in the tails: very short and very long movies fall far from what would be expected under normality Together, these plots indicate that runtime is not perfectly normally distributed. Therefore the normality assumption does not hold, and the result of the variance test should be interpreted with caution.

1. There is an argument going on in the movie studio. *Bob* claims that they should make higher-quality movies, as this will bring in more profits. *Chantal* disagrees. She tells Bob that mediocre movies bring in the most profits. You are asked to advise on who is right.
2. Create a new variable called vote\_average\_rounded. Make sure this variable is the same as vote\_average, but without any decimals (i.e., a 6.3 becomes a 6, a 8.7 an 8, etc.). Display a histogram of vote\_average\_rounded.
3. Create a scatter plot with vote\_average\_rounded on the x axis and the mean of profits within each category of vote\_average\_rounded on the y-axis. Make sure it has an appropriate title, and appropriate titles and labels for the x- and y-axis. At which rating of movies are profits the highest?
4. Recreate the scatter plot with year on the x axis and mean\_profits on the y-axis, but now add bars around each point, indicating the 95% confidence interval.
5. Write an advice to settle the argument between Bob and Chantal.

*step a*

#WRITE YOUR CODE HERE  
movies$vote\_average\_rounded <- floor(movies$vote\_average)  
  
ggplot(movies, aes(x = vote\_average\_rounded)) +  
 geom\_histogram(binwidth = 1) +  
 scale\_x\_continuous(breaks = seq(0,8, by = 1)) +  
 labs(title = "Histogram of Rounded Vote Averages",  
 x = "Vote Average",  
 y = "Frequency")

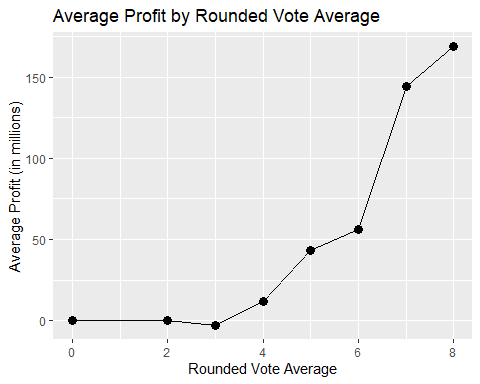


**Your Answer:**

With this code you create the new variable and a histogram.

*step b*

#WRITE YOUR CODE HERE  
profit\_movies <- movies %>%  
 group\_by(vote\_average\_rounded) %>%  
 summarise(profit\_mean = mean(profit, na.rm = TRUE))  
  
  
ggplot(profit\_movies, aes(x = vote\_average\_rounded, y = profit\_mean)) +  
 geom\_point(size = 3) +  
 geom\_line() +  
 labs(  
 title = "Average Profit by Rounded Vote Average",  
 x = "Rounded Vote Average",  
 y = "Average Profit (in millions)"  
 )

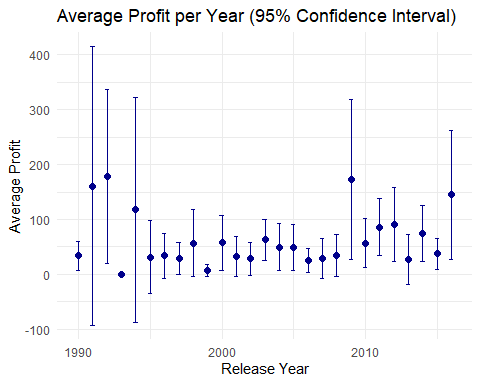


**Your Answer:**

The profits are the highest by a rating of 8.

*step c*

#WRITE YOUR CODE HERE  
  
profit\_by\_year <- movies %>%  
 group\_by(release\_year) %>%  
 summarise(  
 profit\_mean = mean(profit, na.rm = TRUE),  
 sd = sd(profit, na.rm = TRUE),  
 n = n(),  
 se = sd / sqrt(n),  
 lower = profit\_mean - 1.96 \* se,  
 upper = profit\_mean + 1.96 \* se  
 )  
  
ggplot(profit\_by\_year, aes(x = release\_year, y = profit\_mean)) +  
 geom\_point(size = 2, color = "darkblue") +  
 geom\_errorbar(aes(ymin = lower, ymax = upper), width = 0.3, color = "darkblue") +  
 labs(  
 title = "Average Profit per Year (95% Confidence Interval)",  
 x = "Release Year",  
 y = "Average Profit"  
 ) +  
 theme\_minimal()



**Your Answer:**

Above you see the scatterplot created, with the 95% confident interval. We are not sure if it is a typo that we now have to compare the mean profit to the release year, because it is not relevant information for our dilemma.

*step d*

**Your Answer:**

The plot shows that average profits rise with higher vote averages, peaking at ratings around 8. This supports Bob’s claim that higher-quality movies tend to be more profitable, rather than Chantal’s view that mediocre films bring in the most profit. The yearly profit plot shows high variation and wide confidence intervals, suggesting that other factors like market conditions and economic environment also matter. Still, overall the evidence favors Bob, primarily based on the scatterplot of question b.