Modern Data Mining: PCA

Bopei Nie

02/12/2023

Contents

Overview

Principle Component Analysis is widely used in data exploration, dimension reduction, data visualization. The aim is to transform original data into uncorrelated linear combinations of the original data while keeping the information contained in the data. High dimensional data tends to show clusters in lower dimensional view.

Clustering Analysis is another form of EDA. Here we are hoping to group data points which are close to each other within the groups and far away between different groups. Clustering using PC's can be effective. Clustering analysis can be very subjective in the way we need to summarize the properties within each group.

Both PCA and Clustering Analysis are so called unsupervised learning. There is no response variables involved in the process.

For supervised learning, we try to find out how does a set of predictors relate to some response variable of the interest. Multiple regression is still by far, one of the most popular methods. We use a linear models as a working model for its simplicity and interpretability. It is important that we use domain knowledge as much as we can to determine the form of the response as well as the function format of the factors on the other hand.

0.1 Objectives

- PCA
- SVD
- Clustering Analysis
- Linear Regression

0.2 Review materials

- Study Module 2: PCA
- Study Module 3: Clustering Analysis
- Study Module 4: Multiple regression

0.3 Data needed

- `NLSY79.csv`
- `brca_subtype.csv`
- `brca_x_patient.csv`

1 Case study 1: Self-seteem

Self-esteem generally describes a person's overall sense of self-worthiness and personal value. It can play significant role in one's motivation and success throughout the life. Factors that influence self-esteem can be inner thinking, health condition, age, life experiences etc. We will try to identify possible factors in our data that are related to the level of self-esteem.

In the well-cited National Longitudinal Study of Youth (NLSY79), it follows about 13,000 individuals and numerous individual-year information has been gathered through surveys. The survey data is open to public here. Among many variables we assembled a subset of variables including personal demographic variables in different years, household environment in 79, ASVAB test Scores in 81 and Self-Esteem scores in 81 and 87 respectively.

The data is store in `NLSY79.csv`.

Here are the description of variables:

Personal Demographic Variables

- Gender: a factor with levels "female" and "male"
- Education05: years of education completed by 2005
- HeightFeet05, HeightInch05: height measurement. For example, a person of 5'10 will be recorded as HeightFeet05=5, HeightInch05=10.
- Weight05: weight in lbs.
- Income87, Income05: total annual income from wages and salary in 2005.
- Job87 (missing), Job05: job type in 1987 and 2005, including Protective Service Occupations, Food Preparation and Serving Related Occupations, Cleaning and Building Service Occupations, Entertainment Attendants and Related Workers, Funeral Related Occupations, Personal Care and Service Workers, Sales and Related Workers, Office and Administrative Support Workers, Farming, Fishing and Forestry Occupations, Construction Trade and Extraction Workers, Installation, Maintenance and Repairs Workers, Production and Operating Workers, Food Preparation Occupations, Setters, Operators and Tenders, Transportation and Material Moving Workers

Household Environment

- Imagazine: a variable taking on the value 1 if anyone in the respondent's household regularly read magazines in 1979, otherwise 0
- Inewspaper: a variable taking on the value 1 if anyone in the respondent's household regularly read newspapers in 1979, otherwise 0
- Ilibrary: a variable taking on the value 1 if anyone in the respondent's household had a library card in 1979, otherwise 0
- MotherEd: mother's years of education
- FatherEd: father's years of education
- FamilyIncome78

Variables Related to ASVAB test Scores in 1981

Test	Description
AFQT	percentile score on the AFQT intelligence test in 1981
Coding	score on the Coding Speed test in 1981
Auto	score on the Automotive and Shop test in 1981
Mechanic	score on the Mechanic test in 1981
Elec	score on the Electronics Information test in 1981
Science	score on the General Science test in 1981
Math	score on the Math test in 1981
Arith	score on the Arithmetic Reasoning test in 1981
Word	score on the Word Knowledge Test in 1981
Parag	score on the Paragraph Comprehension test in 1981
Numer	score on the Numerical Operations test in 1981

Self-Esteem test 81 and 87

We have two sets of self-esteem test, one in 1981 and the other in 1987. Each set has same 10 questions. They are labeled as `Esteem81` and `Esteem87` respectively followed by the question number. For example, `Esteem81_1` is Esteem question 1 in 81.

The following 10 questions are answered as 1: strongly agree, 2: agree, 3: disagree, 4: strongly disagree

- Esteem 1: "I am a person of worth"
- Esteem 2: "I have a number of good qualities"
- Esteem 3: "I am inclined to feel like a failure"
- Esteem 4: "I do things as well as others"
- Esteem 5: "I do not have much to be proud of"
- Esteem 6: "I take a positive attitude towards myself and others"
- Esteem 7: "I am satisfied with myself"
- Esteem 8: "I wish I could have more respect for myself"
- Esteem 9: "I feel useless at times"
- Esteem 10: "I think I am no good at all"

1.1 Data preparation

Load the data. Do a quick EDA to get familiar with the data set. Pay attention to the unit of each variable. Are there any missing values?

Regarding missing values, apart from the entirely missing job type in 1987 in the data set, there are some other missing values present. The data shows that the minimum value of `Income87` is -2. According to the original data set on the National Longitudinal Study of Youth website, the variable `Income87` indicates "Don't Know" with a value of -2 and "Refusal" with a value of -1. Thus, some of the values of `Income87` are missing. In addition, there are some unusual values, such as the minimum value of `HeightFeet05`, which is -4, indicating an abnormal data point.

To ensure the reliability of the results, the rows containing missing or unusual values are deleted, as there isn't a significant amount of missing data.

Besides, there are some variables with incorrect types. The `Subject` variable, which is assigned to each individual, should be a factor instead of a numerical value (integer). `Job05` is a character in the data set, but it should be transformed into a factor. Thus, these incorrect types are corrected.

Additionally, it is worth noticing that the unit of height is inch and feet, the unit of weight is lbs and the unit of income is dollars. Thus, in Question 6, we should first convert the unit before calculating BMI.

```
temp <- read.csv('data/NLSY79.csv', header = T, stringsAsFactors = F)
summary(temp)
temp <-temp[temp$Income87 >= 0, ]
temp <-temp[temp$HeightFeet05 >= 0, ]
```

1.2 Self esteem evaluation

Let concentrate on Esteem scores evaluated in 87.

0. First do a quick summary over all the `Esteem` variables. Pay attention to missing values, any peculiar numbers etc. How do you fix problems discovered if there is any? Briefly describe what you have done for the data preparation.

In order to summarise all the `Esteem` variables, we select the columns containing "Esteem87" and store them in `data.esteem`. Then we apply`summary()` to `data.esteem` to look up the minimum, 1st quantile, median, mean, 3rd quantile and maximum of the variables that we are interested in.

The results shows that there is no missing values or peculiar data in the `Esteem` variables. The minimum and maximum of every esteem score evaluated in 1987 is 1 and 4, respectively.

```r
# select the columns containing "Esteem87", and use summary() to look into the data.
col_names <- colnames(temp)
esteem_cols <- col_names[grep("Esteem87", col_names)]
data.esteem <- temp[, esteem_cols]
summary(data.esteem)
```

```
##    Esteem87_1     Esteem87_2    Esteem87_3     Esteem87_4    Esteem87_5
##  Min.   :1.00   Min.   :1.0   Min.   :1.00   Min.   :1.0   Min.   :1.00
##  1st Qu.:1.00   1st Qu.:1.0   1st Qu.:3.00   1st Qu.:1.0   1st Qu.:3.00
##  Median :1.00   Median :1.0   Median :4.00   Median :1.0   Median :4.00
##  Mean   :1.38   Mean   :1.4   Mean   :3.58   Mean   :1.5   Mean   :3.53
##  3rd Qu.:2.00   3rd Qu.:2.0   3rd Qu.:4.00   3rd Qu.:2.0   3rd Qu.:4.00
##  Max.   :4.00   Max.   :4.0   Max.   :4.00   Max.   :4.0   Max.   :4.00
##    Esteem87_6     Esteem87_7     Esteem87_8    Esteem87_9    Esteem87_10
##  Min.   :1.00   Min.   :1.00   Min.   :1.0   Min.   :1.00   Min.   :1.00
##  1st Qu.:1.00   1st Qu.:1.00   1st Qu.:3.0   1st Qu.:3.00   1st Qu.:3.00
##  Median :2.00   Median :2.00   Median :3.0   Median :3.00   Median :3.00
##  Mean   :1.59   Mean   :1.72   Mean   :3.1   Mean   :3.06   Mean   :3.37
##  3rd Qu.:2.00   3rd Qu.:2.00   3rd Qu.:4.0   3rd Qu.:4.00   3rd Qu.:4.00
##  Max.   :4.00   Max.   :4.00   Max.   :4.0   Max.   :4.00   Max.   :4.00
```

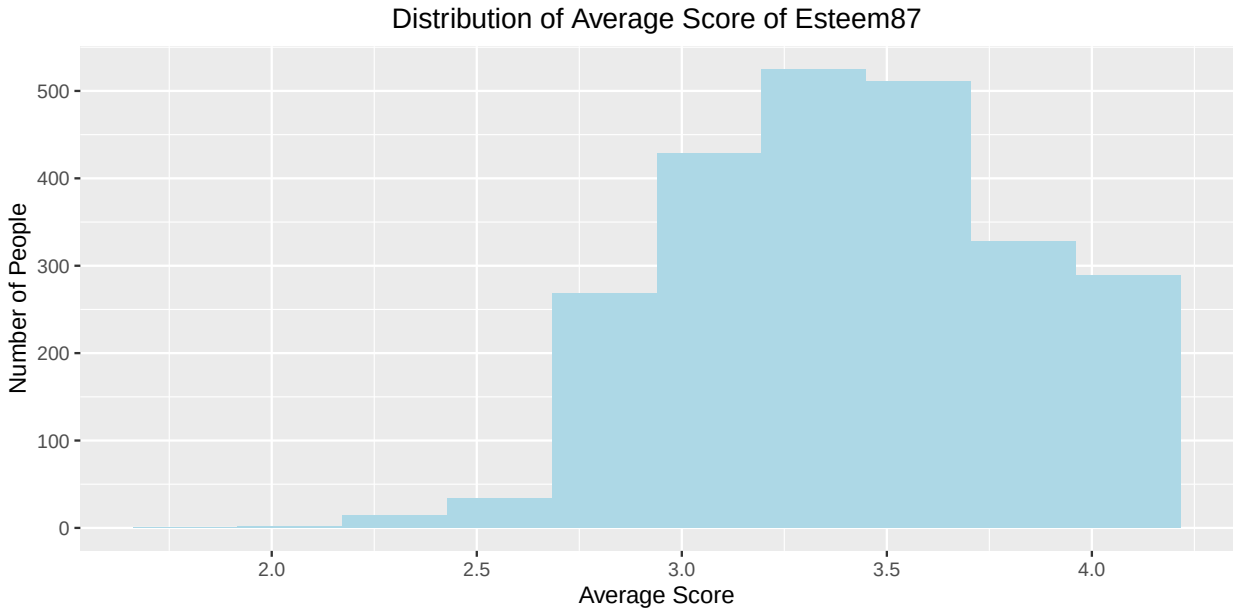
1. Reverse Esteem 1, 2, 4, 6, and 7 so that a higher score corresponds to higher self-esteem.

```r
data.esteem[,  c(1, 2, 4, 6, 7)]  <- 5 - data.esteem[,  c(1, 2, 4, 6, 7)]
```

2. Write a brief summary with necessary plots about the 10 esteem measurements.

The graph shows the distribution of average score of the 10 esteem measurements. Most people score 2.7 to 4.0 on average of the ten questions, and the proportion of people receiving higher score are more than those receive lower scores. The highest average score is 4.0, which means this person obtain the maximum score in each measurement.

```r
data.esteem.2 <- data.esteem %>%  rowwise() %>% mutate(row_mean = mean(c(Esteem87_1,Esteem87_2,Esteem87

ggplot(data.esteem.2) +
geom_histogram(aes(x = row_mean), bins = 10, fill = "lightblue") +
scale_fill_brewer(type = "qual", palette = "Set1") +
labs( title = "Distribution of Average Score of Esteem87", x = "Average Score" , y = "Number of People"
theme(plot.title = element_text(hjust = 0.5))
```



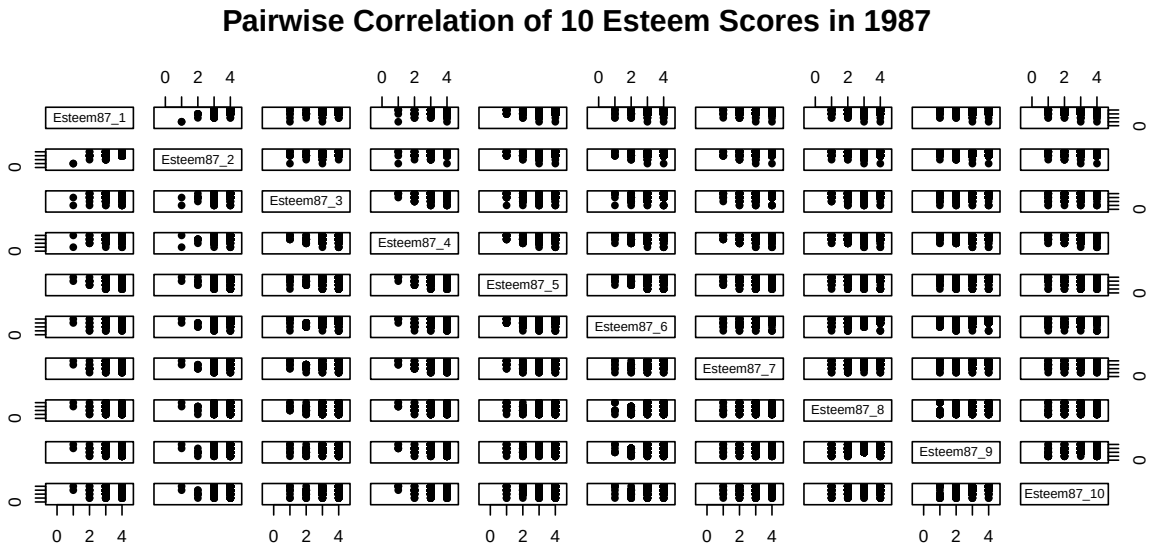
3. Do esteem scores all positively correlated? Report the pairwise correlation table and write a brief summary.

From the pairwise correlation table below, we can see that most esteem scores are clearly positively correlated although some are not strongly correlated.

The pairwise correlation table illustrates the correlation of 10 esteem scores in 1987. As the table reflects the symmetric data, we can focus on the plots on bottom left.

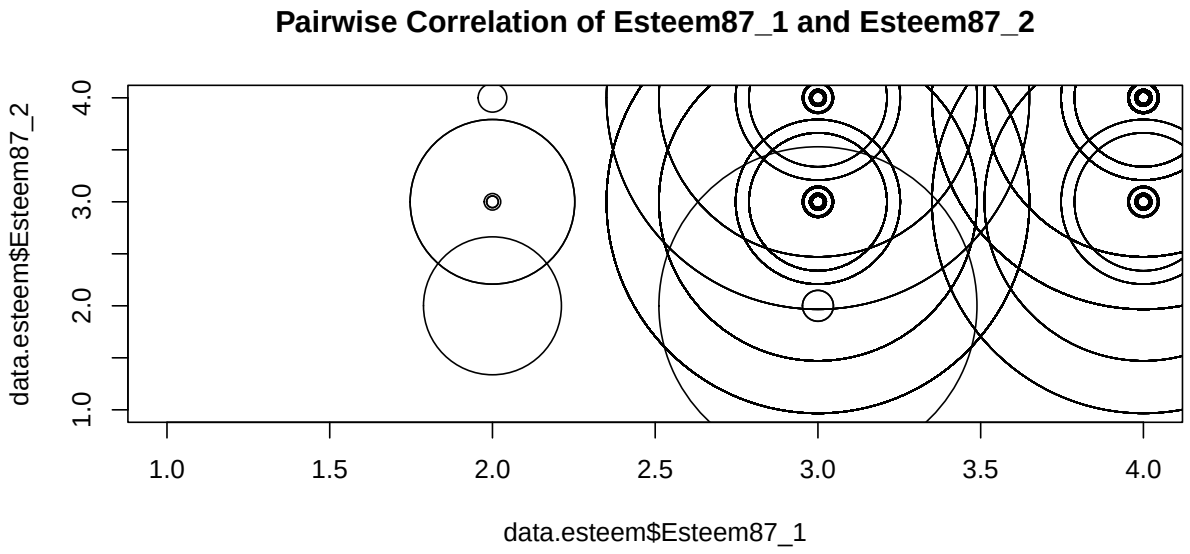
It is worth noticing that all those spots represent the score of the ten esteem questions between one to four.

```
pairs(data.esteem, xlim=c(-0.5, 4.5), ylim=c(-0.5, 4.5), pch=16, main = "Pairwise Correlation of 10 Est
```

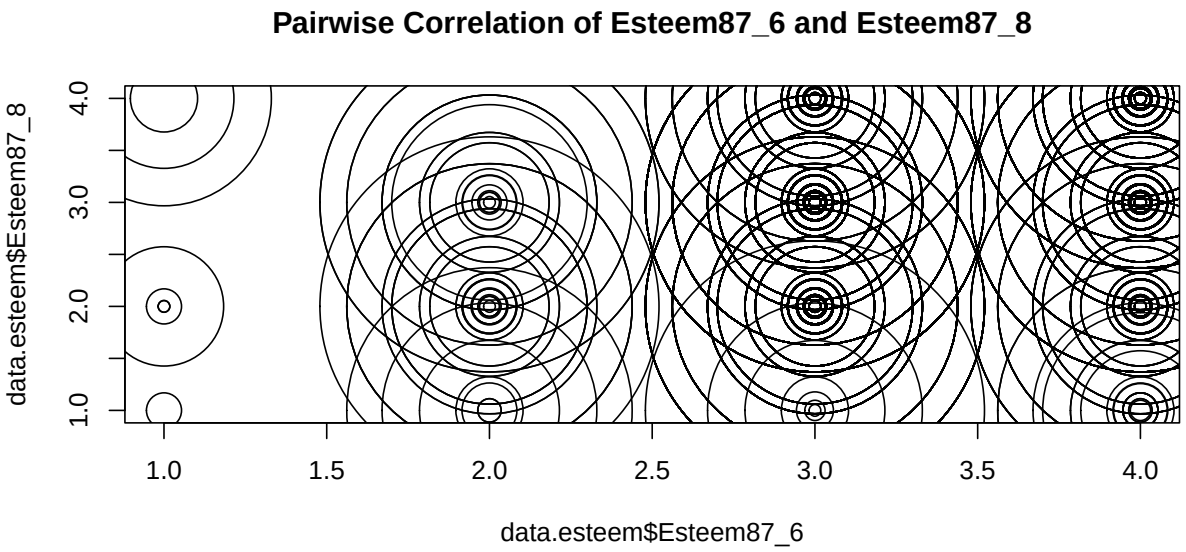


Furthermore, we select some of the pairwise correlation plots and zoom in to see in detail. The circles indicates the number of score pairs. The more frequently a pair of data points appears in the dataset, the larger the circles, and the more circles there are. From those graphs, we can see that most of the pairs are positively correlated.

```
# Calculate the frequency of each data point
point_counts <- table(data.esteem$Esteem87_1, data.esteem$Esteem87_2)
# Create a variable to store the cex values based on the frequency of each data point
cex_values <- sqrt(point_counts)
# Plot the data, using cex values to control the size of the points
p1 <- plot(data.esteem$Esteem87_1, data.esteem$Esteem87_2, cex = cex_values, main = "Pairwise Correlati
```




```
point_counts <- table(data.esteem$Esteem87_6, data.esteem$Esteem87_8)
cex_values <- sqrt(point_counts)
p2 <- plot(data.esteem$Esteem87_6, data.esteem$Esteem87_8, cex = cex_values, main = "Pairwise Correlati
```



4. PCA on 10 esteem measurements. (centered but no scaling)

    a) Report the PC1 and PC2 loadings. Are they unit vectors? Are they orthogonal?

PC1 and PC2 loadings are shown in the table.

The first two principal components (PC1 and PC2) are linear combinations of the original variables in the data. The loadings of PC1 and PC2 represent the weight given to each original variable in the calculation of the first two PCs.

The loadings of PC1 and PC2 are unit vectors, and they represent the relative importance of the variables in the calculation of the PCs.

PC1 and PC2 are orthogonal, which means that they are perpendicular to each other. Orthogonality is a property of principal component analysis (PCA) that ensures that the first PC captures the maximum amount of variance in the data, the second PC captures the maximum amount of variance in the data that is orthogonal to the first PC, and so on. This property helps to simplify the interpretation of the PCs, as each PC captures a unique aspect of the data that is not captured by any of the other PCs.

```
pc_10 <- prcomp(data.esteem, scale=FALSE) # by default, center=True but scale=FALSE!!!

pc_10.loading <- pc_10$rotation
knitr::kable(pc_10.loading[,1:2])
```

	PC1	PC2
Esteem87_1	0.234	-0.376
Esteem87_2	0.244	-0.370
Esteem87_3	0.278	-0.152
Esteem87_4	0.260	-0.323
Esteem87_5	0.312	-0.133
Esteem87_6	0.312	-0.207
Esteem87_7	0.299	-0.159
Esteem87_8	0.394	0.331

	PC1	PC2
Esteem87_9	0.400	0.575
Esteem87_10	0.376	0.260

b) Are there good interpretations for PC1 and PC2? (If loadings are all negative, take the positive loa

Each loadings give us a set of ten numbers which determines the direction of each line. Loadings are unique up to sign, so we can change all of the signs and maintain the information stored in them.

Regarding PC1, it is worth noticing that the ten loadings are approximately the same around 0.3, and thus PC1 is proportional to the total of the ten scores. Also, we should not neglect the fact that loadings of Esteem87_1, 2, 3 and Esteem87_4, which are around 0.25, are slightly smaller than the counterpart of the last six scores. The difference in weight indicates that the first four esteem score contributes slightly less than the last six scores in constructing PC1.

In general, PC1 is approximately the weighted sum of all ten scores. Besides, from the value of loadings of PC1, we can tell that a higher PC1 indicates a higher weighted total score.

As for PC2, notice that the value of PC2 is difference between the sum of the scores of Esteem87_8, Esteem87_9 and Esteem87_10 and the sum of the first 7 scores. So, PC2 is approximately proportional to the difference between the sum of last three scores and that of the first seven scores of 1987. Furthermore, if the total scores are comparable, higher PC2 implies strong self-esteem in the aspects represented by the last three questions relatively, while lower PC2 implies relatively higher level of self-esteem in the aspects represented by the first seven questions.

c) How is the PC1 score obtained for each subject? Write down the formula.

According to the definition, we derive the formula by taking the linear combination of loadings and variables. To be more precise, we multiply the vector of loadings to the vector of variables.

Regarding PC1, take the linear combination according to the loadings, we get:

PC1 = 0.234 × Esteem87_1 + 0.244 × Esteem87_2 + 0.278 × Esteem87_3 + 0.260 × Esteem87_4 + 0.312 × Esteem87_5 + 0.312 × Esteem87_6 + 0.299 × Esteem87_7 + 0.394 × Esteem87_8 + 0.400 × Esteem87_9 + 0.376 × Esteem87_10

Then, we apply the same method to compute the value of PC2:

PC2 = (-0.376 × Esteem87_1) + (-0.370 × Esteem87_2) + (-0.152 × Esteem87_3) + (-0.323 × Esteem87_4) + (-0.133 × Esteem87_5) + (-0.207 × Esteem87_6) + (-0.159 × Esteem87_7) + 0.331 × Esteem87_8 + 0.575 × Esteem87_9 + 0.260 × Esteem87_10

= -(0.376 × Esteem87_1 + 0.370 × Esteem87_2 + 0.152 × Esteem87_3 + 0.323 × Esteem87_4 + 0.133 × Esteem87_5 + 0.207 × Esteem87_6 + 0.159 × Esteem87_7) + (0.331 × Esteem87_8 + 0.575 × Esteem87_9 + 0.260 × Esteem87_10)

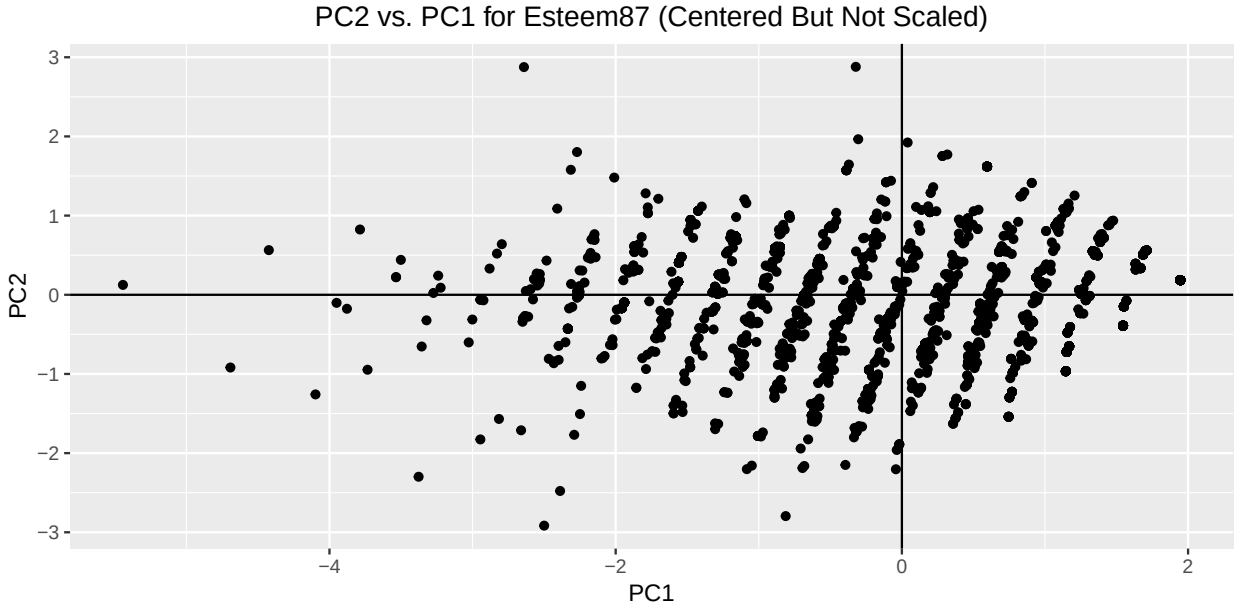
Besides, we can compute the PC1 and PC2 and plot them using the following code.

```
pc_scores <- cbind(scale(data.esteem, scale = FALSE), pc_10$x)
arrange(as.data.frame(pc_scores)) %>%
head()
```

```
##   Esteem87_1 Esteem87_2 Esteem87_3 Esteem87_4 Esteem87_5 Esteem87_6 Esteem87_7
## 1     -0.621      0.401      0.417      0.504     -1.534     -0.407     -0.282
## 2      0.379      0.401      0.417      0.504      0.466      0.593     -0.282
## 3     -0.621     -0.599      0.417     -0.496      0.466     -0.407     -0.282
```

```
## 4      0.379      0.401     -0.583      0.504      0.466     -0.407      0.718
## 5      0.379      0.401      0.417      0.504      0.466      0.593      0.718
## 6      0.379      0.401      0.417      0.504      0.466      0.593     -0.282
##   Esteem87_8 Esteem87_9 Esteem87_10    PC1    PC2      PC3    PC4     PC5
## 1    -0.0995    -0.0646       0.631 -0.318  0.287  0.27199  0.475  0.6088
## 2    -0.0995    -1.0646       0.631  0.453 -1.138 -0.04594 -0.588 -0.2369
## 3     0.9005    -0.0646       0.631  0.196  1.044 -0.52221 -1.018 -0.5108
## 4    -1.0995    -1.0646      -1.369 -0.985 -1.790  0.00627  0.299 -0.1814
## 5     0.9005     0.9354       0.631  1.945  0.185 -0.12272  0.221  0.0309
## 6     0.9005     0.9354       0.631  1.646  0.344  0.01440 -0.271  0.5446
##       PC6     PC7     PC8     PC9    PC10
## 1 -1.2512  0.2634  0.8006 -0.5981 -0.7047
## 2 -0.8949 -0.2353  0.0180  0.4004 -0.0393
## 3 -0.3319  0.0360  0.1077 -0.1332 -0.0303
## 4  0.9288 -0.4360 -0.5162 -0.6290  0.0289
## 5  0.1107 -0.0112  0.0238 -0.0753  0.0074
## 6 -0.0918 -0.2141  0.1542  0.4343 -0.0292
```

```
as.data.frame(pc_10$x) %>%
ggplot(aes(x=PC1, y=PC2)) +
geom_point()+
geom_vline(xintercept = 0) +
geom_hline(yintercept = 0) +
ggtitle("PC2 vs. PC1 for Esteem87 (Centered But Not Scaled)") +
theme(plot.title = element_text(hjust = 0.5))
```



d) Are PC1 scores and PC2 scores in the data uncorrelated?

Theoretically, the first two principal components (PC1 and PC2) are orthogonal to each other, meaning that they are uncorrelated. This means that the correlation between the PC1 scores and PC2 scores should be close to zero.

However, this does not always guarantee that the PC1 scores and PC2 scores will be perfectly uncorrelated, as there may be some slight rounding errors or numerical instability in the PCA computation. In practice,

it depends on the specific data and the way the principal component analysis (PCA) was performed. It is common to use a threshold to define the uncorrelatedness of the two components, such as a correlation coefficient below a certain value (e.g., 0.01 or 0.05). If the correlation between PC1 and PC2 scores is below this threshold, they can be considered uncorrelated.

```
cor(pc_scores[, "PC1"], pc_scores[, "PC2"])
```

```
## [1] -1.6e-15
```

In this specific data set, we can calculate the correlation of PC1 and PC2 to check if they are correlated. The result should that the correlation of PC1 and PC2 is -1.6e-15, which is a rather small number (in absolute value). Therefore, we can conclude that PC1 and PC2 are uncorrelated in this data set.

e) Plot PVE (Proportion of Variance Explained) and summarize the plot.

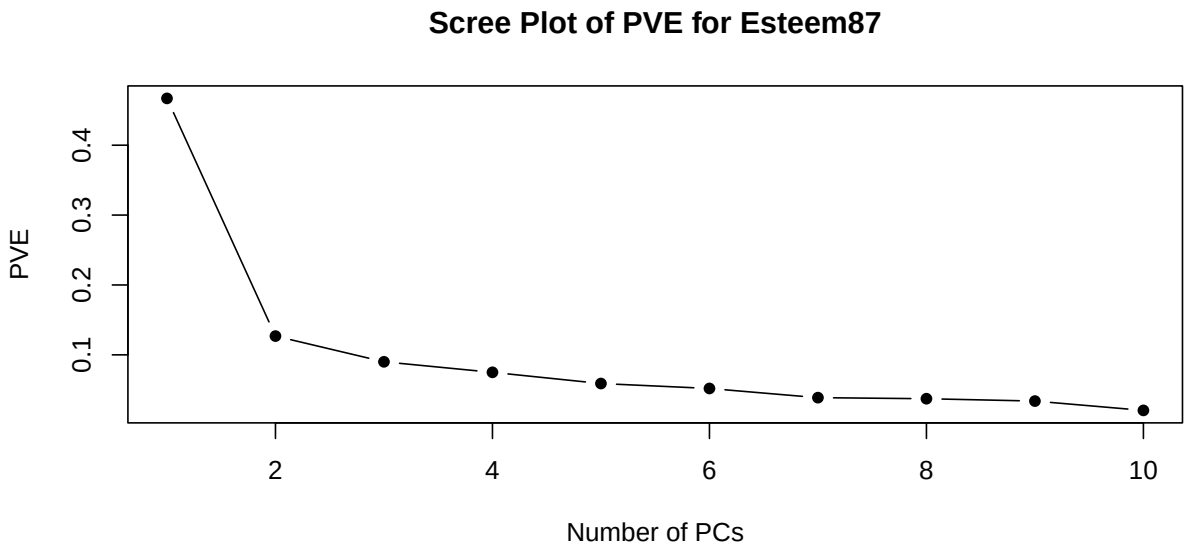
```
summary(pc_10)$importance
```

```
##                           PC1   PC2   PC3   PC4   PC5    PC6    PC7    PC8
## Standard deviation     1.296 0.676 0.569 0.519 0.461 0.4321 0.3734 0.3660
## Proportion of Variance 0.467 0.127 0.090 0.075 0.059 0.0519 0.0387 0.0372
## Cumulative Proportion  0.467 0.594 0.684 0.759 0.818 0.8697 0.9084 0.9456
##                           PC9   PC10
## Standard deviation     0.3490 0.2718
## Proportion of Variance 0.0338 0.0205
## Cumulative Proportion  0.9795 1.0000
```

The summary reports standard deviations, PVE (PVE = Var(PC) / Total Variances) and cumulative proportions and .

From the second row of the table, we can clearly see the proportion of variance of PCs. The leading principal component (PC1) explains almost half of the total variance (46.7%) and the PC2 explains 12.7% of the total variance. Besides, it is worth noticing the relationship: Var(PC1)>Var(PC2)>...>Var(PC10) from data, which is in line with the theoretical expectations.

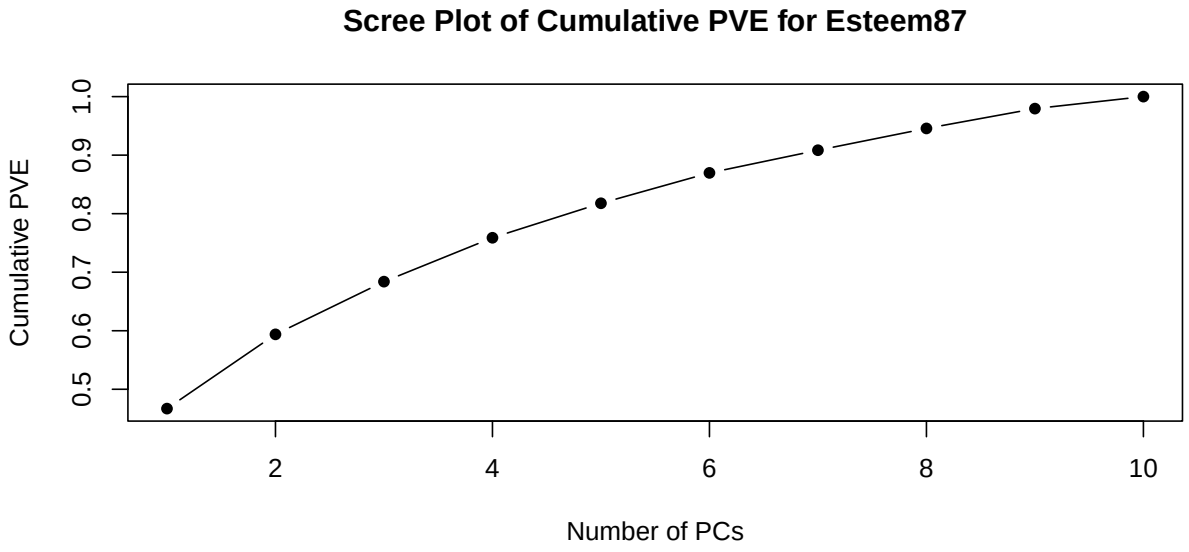
```
plot(summary(pc_10)$importance[2, ], # PVE
     ylab="PVE",
     xlab="Number of PCs",
     pch = 16,
     main="Scree Plot of PVE for Esteem87",
     type="b")
```



f) Also plot CPVE (Cumulative Proportion of Variance Explained). What proportion of the variance in the

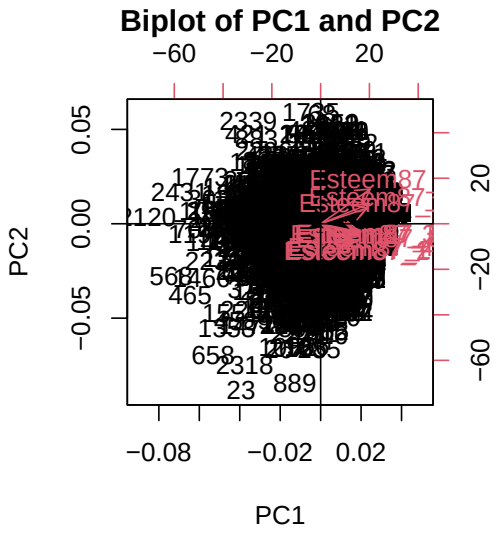
From the data from Cumulative Proportion of Variance Explained, we can see that 59.4% of the variance in the data is explained by the first two principal components.

```
plot(summary(pc_10)$importance[3, ], pch=16,
ylab="Cumulative PVE",
xlab="Number of PCs",
type="b",
main="Scree Plot of Cumulative PVE for Esteem87")
```



g) PC's provide us with a low dimensional view of the self-esteem scores. Use a biplot with the first tw

```
p <- biplot(pc_10, choices=c(1,2),
xlim=c(-0.09,0.05),
ylim=c(-0.09,0.06),
main="Biplot of PC1 and PC2")
abline(v=0, h=0)
```



The biplot indicates:

• PC1 loadings are similar in magnitudes and with same signs. • PC2 captures difference between total of Esteem87_1, 2, . . . , 7 and total of the last three scores(Esteem87_8, 9 and 10). • There are three groups of scores that are highly correlated.

- Esteem87_1, 2 and 4 are highly correlated.
- Esteem87_3, 5, 6 and 7 are highly correlated.
- Esteem87_8, 9 and 10 are highly correlated.

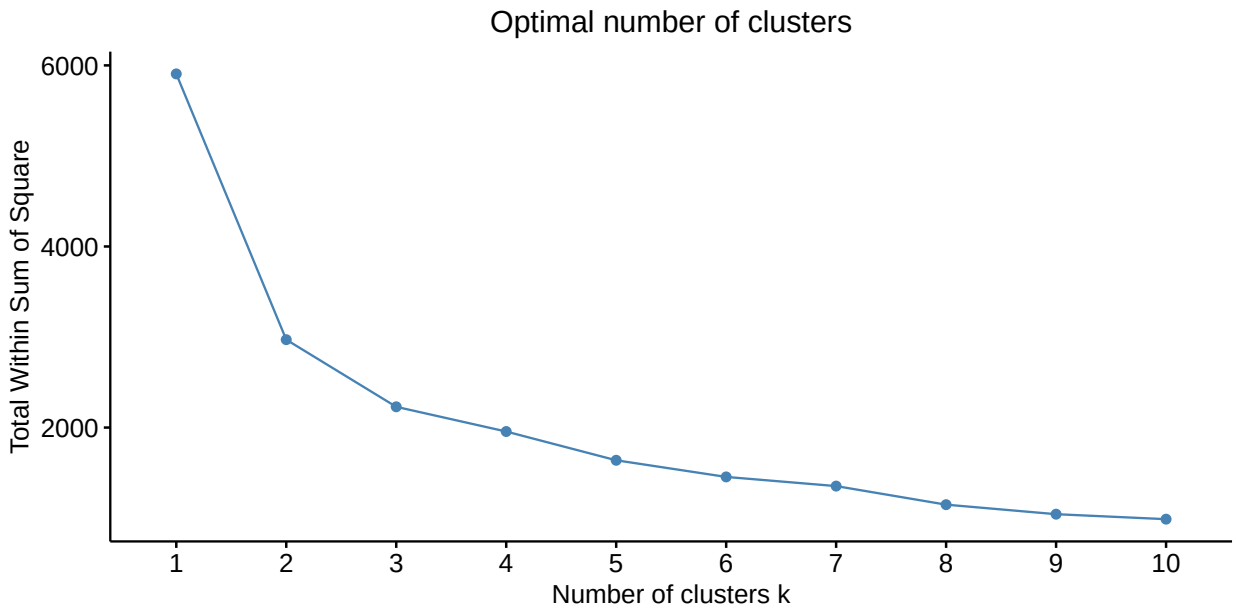
This result provides an insight that inside each group, the scores may represents the similar aspect of self-esteem.

5. Apply k-means to cluster subjects on the original esteem scores

   a) Find a reasonable number of clusters using within sum of squared with elbow rules.

From the plot: "Optimal number of clusters", we can see that as the number of clusters k increases, total within sum square decreases. Furthermore, a sharp drop of total within sum square occur when the number of k equals to two. Also, when k is larger than three, the slope doesn't change much when k increases. According to the elbow rule, it is appropriate to choose the number of k "at the elbow". That is to say, it is reasonable to set k equals to three in order to get the better clustering results.

```r
set.seed(0)
pc_clus <- as.data.frame(pc_10$x)

fviz_nbclust(pc_clus[,1:3], kmeans, method = "wss")+
  theme(plot.title = element_text(hjust = 0.5))
```



b) Can you summarize common features within each cluster?

When using K-Means, the algorithm aims to divide the data into K clusters (here K = 3) such that the data points within each cluster are as similar as possible, based on a similarity metric, such as Euclidean distance, while data points in different clusters are as dissimilar as possible.

```r
pc_kmeans <- kmeans(pc_clus, centers = 3 )
str(pc_kmeans)
```

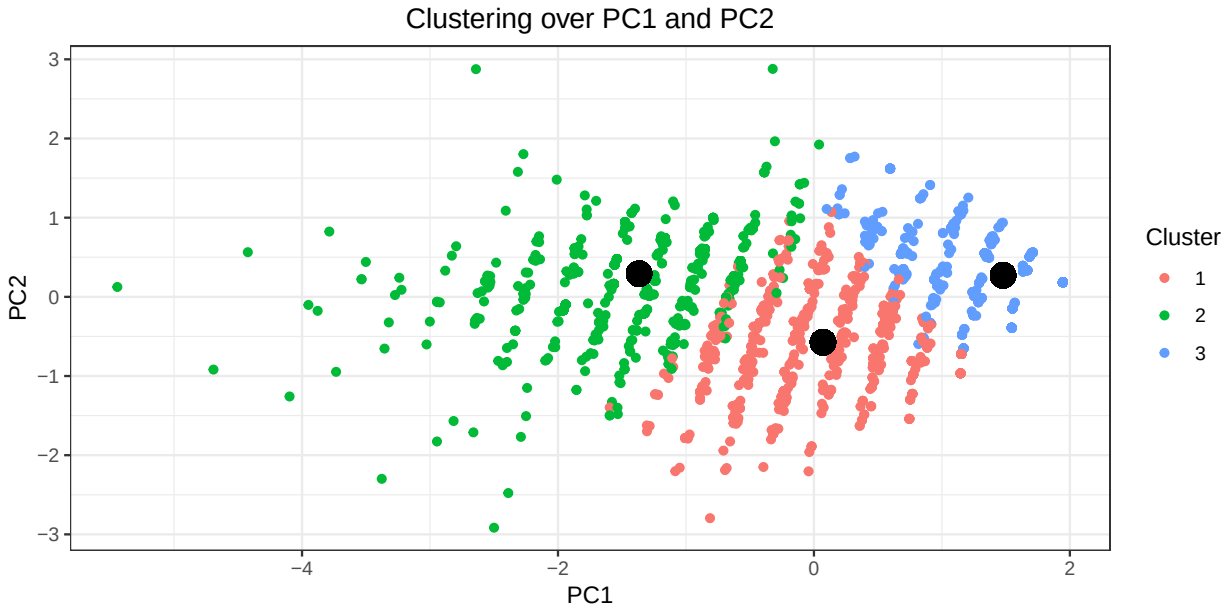
```
## List of 9
##  $ cluster     : Named int [1:2401] 1 1 3 1 3 3 3 2 3 3 ...
##   ..- attr(*, "names")= chr [1:2401] "1" "2" "3" "4" ...
##  $ centers     : num [1:3, 1:10] 0.0709 -1.3661 1.4783 -0.5745 0.293 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:10] "PC1" "PC2" "PC3" "PC4" ...
##  $ totss       : num 8636
##  $ withinss    : num [1:3] 2125 1784 1038
##  $ tot.withinss: num 4948
##  $ betweenss   : num 3689
##  $ size        : int [1:3] 792 856 753
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
p1 <- data.table(x = pc_10$x[,1],
y = pc_10$x[,2],
col = as.factor(pc_kmeans$cluster),
pc1 = pc_kmeans$centers[,1],
pc2 = pc_kmeans$centers[,2]) %>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc1, y = pc2), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC1") +
ylab("PC2")+
ggtitle("Clustering over PC1 and PC2")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning in as.data.table.list(x, keep.rownames = keep.rownames, check.names =
## check.names, : Item 4 has 3 rows but longest item has 2401; recycled with
## remainder.
```

```
## Warning in as.data.table.list(x, keep.rownames = keep.rownames, check.names =
## check.names, : Item 5 has 3 rows but longest item has 2401; recycled with
## remainder.
```

```
p1
```



From the analysis of this graph, it is evident that there are three distinct clusters with well-defined boundaries.

The first cluster, depicted in red, is situated in the mid-bottom section of the graph and encompasses individuals whose total esteem score is approximately equal to the sample mean. The average of Esteem87_8, 9 and 10 is lower than the mean of Esteem87_1, 2, ... , 7.

The green dots make up the second cluster and correspond to individuals whose total esteem score is below the average. The cluster center is positioned above the horizontal line PC2 = 0, indicating that these

individuals generally score higher on the mean of the last three esteem scores compared to the first seven. It is also worth noting that there are some individuals within the cluster whose mean score on the last three esteem scores is lower than the mean score on the first seven.

The third cluster comprises individuals with the highest levels of self-esteem, as evidenced by the largest average score of PC1. This cluster also generally demonstrates a higher average score on the last three esteem scores compared to the first seven.

`c) Can you visualize the clusters with somewhat clear boundaries? You may try different pairs of variabl`

In the following graphs, we plot the clusters over PCs pairwisely. It shows that the cluster over PC1 and PC2 has the best performance and the clearest boundaries.

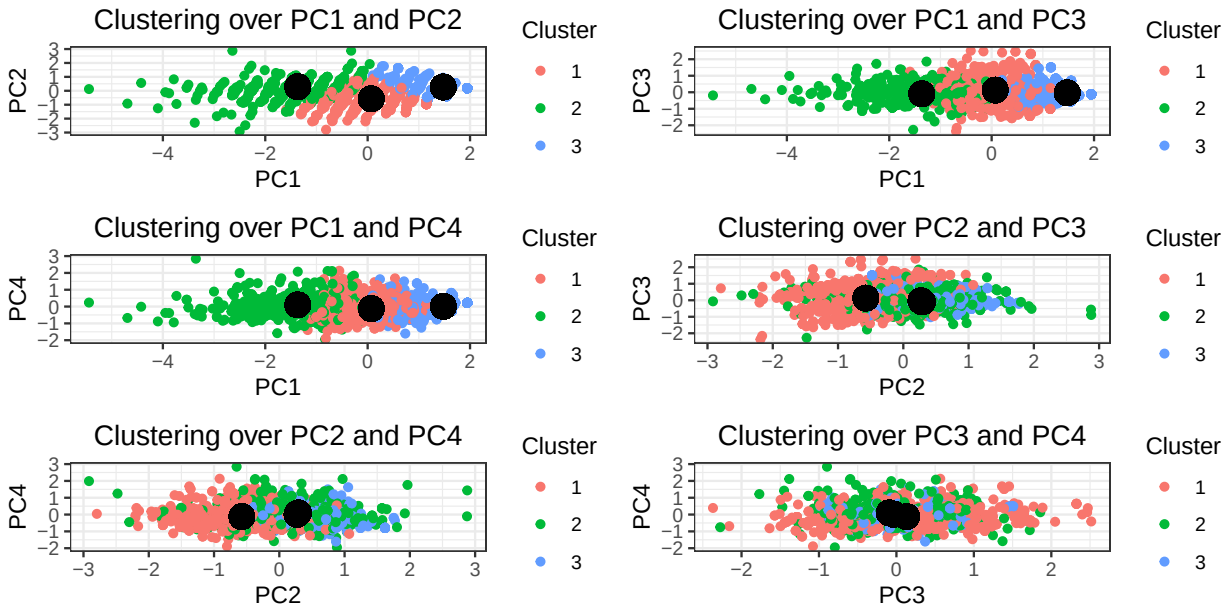
Besides, since the variables Esteem87_1 to Esteem887_10 can only take values: 1, 2, 3 and 4, it's not suitable to put those variables directly to operate clustering.

Thus, we choose the best clustering with the clearest boundaries: the clustering over PC1 and PC2.

```
p1 <- data.table(x = pc_10$x[,1],
y = pc_10$x[,2],
col = as.factor(pc_kmeans$cluster),
pc1 = pc_kmeans$centers[,1],
pc2 = pc_kmeans$centers[,2])%>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc1, y = pc2), size = 5) +

theme_bw() +
labs(color = "Cluster") +
xlab("PC1") +
ylab("PC2")+
ggtitle("Clustering over PC1 and PC2")+
  theme(plot.title = element_text(hjust = 0.5))


p2 <- data.table(x = pc_10$x[,1],
y = pc_10$x[,3],
col = as.factor(pc_kmeans$cluster),
pc1 = pc_kmeans$centers[,1],
pc3= pc_kmeans$centers[,3])%>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc1, y = pc3), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC1") +
ylab("PC3")+
ggtitle("Clustering over PC1 and PC3")+
  theme(plot.title = element_text(hjust = 0.5))

p3 <- data.table(x = pc_10$x[,1],
y = pc_10$x[,4],
col = as.factor(pc_kmeans$cluster),
pc1 = pc_kmeans$centers[,1],
pc4= pc_kmeans$centers[,4])%>%
```

```
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc1, y = pc4), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC1") +
ylab("PC4")+
ggtitle("Clustering over PC1 and PC4")+
  theme(plot.title = element_text(hjust = 0.5))

p4 <- data.table(x = pc_10$x[,2],
y = pc_10$x[,3],
col = as.factor(pc_kmeans$cluster),
pc2 = pc_kmeans$centers[,2],
pc3= pc_kmeans$centers[,3])%>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc2, y = pc3), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC2") +
ylab("PC3")+
ggtitle("Clustering over PC2 and PC3")+
  theme(plot.title = element_text(hjust = 0.5))

p5 <- data.table(x = pc_10$x[,2],
y = pc_10$x[,4],
col = as.factor(pc_kmeans$cluster),
pc2 = pc_kmeans$centers[,2],
pc4= pc_kmeans$centers[,4])%>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc2, y = pc4), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC2") +
ylab("PC4")+
ggtitle("Clustering over PC2 and PC4")+
  theme(plot.title = element_text(hjust = 0.5))

p6 <- data.table(x = pc_10$x[,3],
y = pc_10$x[,4],
col = as.factor(pc_kmeans$cluster),
pc3 = pc_kmeans$centers[,3],
pc4= pc_kmeans$centers[,4])%>%
ggplot() +
geom_point(aes(x = x, y = y, col = col)) +
geom_point(aes(x = pc3, y = pc4), size = 5) +
theme_bw() +
labs(color = "Cluster") +
xlab("PC3") +
ylab("PC4")+
ggtitle("Clustering over PC3 and PC4")+
```

```
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(p1,p2, p3, p4, p5,p6, nrow = 3, ncol = 2)
```



6. We now try to find out what factors are related to self-esteem? PC1 of all the Esteem scores is a good variable to summarize one's esteem scores. We take PC1 as our response variable.

   a) Prepare possible factors/variables:

   - EDA the data set first.

```
# already read in temp
dim(temp)
head(temp)
```

```
  - Personal information: gender, education (05), log(income) in 87, job type in 87. Weight05 (lb) and
```

Because the units of height are inch and feet and the unit of weight is lbs in the original data set, we should first convert the units into meter and kilogram before calculating BMI. (1 feet = 0.3048m, 1inch = 0.0254m). Then, we calculate BMI and take logs to Income87.

```
feet_to_meters <- function(feet, inches) {
  meters <- 0.3048 * feet + 0.0254 * inches
  return(meters)
}
h = feet_to_meters(temp$HeightFeet05, temp$HeightInch05)

lbs_to_kg <- function(lbs) {
  kg <- lbs * 0.45359237
  return(kg)
```

```
}
w = lbs_to_kg(temp$Weight05)

BMI <- w/(h*h)
lg_inc87 <- log(temp$Income87)
```

- Household environment: Imagazine, Inewspaper, Ilibrary, MotherEd, FatherEd, FamilyIncome78. Do set

```
magazine <- as.factor(temp$Imagazine)
newspaper <- as.factor(temp$Inewspaper)
library <- as.factor(temp$Ilibrary)
```

- You may use PC1 of ASVAB as level of intelligence

Apply PCA method to the ten scores in ASVAB to get the level of intelligence. Since PC1 explains a significantly large amount of variance, and all loadings of PC1 are of similar magnitude and same sign, we use PC1 to represent the intelligence.

```
# get PC1 of ASVAB as level of intelligence
pca.asvab <- prcomp(temp[, c(16:25)], scale=T) # all the tests
intell <- pca.asvab$x[,1]
```

Finally, construct the dataframe of all independent variables interested and response variable.

```
data.reg <- data.frame(gender = temp$Gender, educ = temp$Education05, lg_inc87 = lg_inc87, BMI = BMI, ma

head(data.reg)
```

b) Run a few regression models between PC1 of all the esteem scores and suitable variables listed in a

- How did you land this model? Run a model diagnosis to see if the linear model assumptions are reasor

- Write a summary of your findings. In particular, explain what and how the variables in the model af

First, put all the variables into the regression. From the summary, we can see that there are three significant variables level: education, intelligence and newspaper. Education and intelligence are both strongly significant at almost 0.001 confidence level, and the dummy variable newpaper is significant at 5% confidence level.

However, other variables are not significant even at 10% and the R square is rather small(0.0962). This model does not fit the data well, so we move out the insignificant variables one by one and do a few regressions.

```
fit1 <- lm(formula=PC1 ~ educ+ BMI+ magazine+ newspaper+ library+ momed+ daded+ inc_78+ intell, data = d
summary(fit1)
```

```
##
## Call:
## lm(formula = PC1 ~ educ + BMI + magazine + newspaper + library +
##     momed + daded + inc_78 + intell, data = data.reg)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.308 -0.944 -0.056  1.010  2.994
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.26e+00   2.42e-01   -5.20  2.2e-07 ***
## educ         6.36e-02   1.25e-02    5.09  3.9e-07 ***
## BMI         -3.07e-03   4.35e-03   -0.71    0.480
## magazine1    2.63e-02   6.23e-02    0.42    0.673
## newspaper1   1.39e-01   8.04e-02    1.73    0.084 .
## library1     7.83e-02   6.34e-02    1.24    0.217
## momed        1.52e-02   1.30e-02    1.17    0.240
## daded        1.53e-03   9.63e-03    0.16    0.874
## inc_78       2.91e-06   1.99e-06    1.46    0.145
## intell       8.72e-02   1.31e-02    6.65  3.6e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 2391 degrees of freedom
## Multiple R-squared:  0.0962, Adjusted R-squared:  0.0928
## F-statistic: 28.3 on 9 and 2391 DF,  p-value: <2e-16
```

Then, we move out mother education and father education. The regression result is as follow.

```
fit2 <- lm(formula=PC1 ~ educ+ BMI+ magazine+ newspaper+ inc_78+ intell, data = data.reg)
summary(fit2)
```

```
##
## Call:
## lm(formula = PC1 ~ educ + BMI + magazine + newspaper + inc_78 +
##     intell, data = data.reg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.303 -0.953 -0.061  1.005  3.049
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.12e+00   2.26e-01   -4.95  8.0e-07 ***
## educ         6.87e-02   1.22e-02    5.66  1.7e-08 ***
## BMI         -3.43e-03   4.34e-03   -0.79    0.430
## magazine1    4.34e-02   6.15e-02    0.71    0.480
## newspaper1   1.74e-01   7.84e-02    2.22    0.027 *
## inc_78       3.48e-06   1.95e-06    1.78    0.075 .
## intell       9.16e-02   1.28e-02    7.18  9.1e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 2394 degrees of freedom
## Multiple R-squared:  0.0947, Adjusted R-squared:  0.0924
## F-statistic: 41.7 on 6 and 2394 DF,  p-value: <2e-16
```

The final model includes four independent variables: education, intelligence, family income in 1978 and read newspaper or not. Education, intelligence and newspaper are significant at 0.001 confidence level, and family income in 1978 is significant at 0.05 confidence level.

```
fit3 <- lm(formula=PC1 ~ educ+ newspaper+ inc_78+ intell, data = data.reg)
summary(fit3)
```

```
##
## Call:
## lm(formula = PC1 ~ educ + newspaper + inc_78 + intell, data = data.reg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.288 -0.962 -0.057  1.012  3.033
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.22e+00   1.78e-01   -6.82  1.2e-11 ***
## educ         7.03e-02   1.21e-02    5.83  6.2e-09 ***
## newspaper1   1.83e-01   7.73e-02    2.37    0.018 *
## inc_78       3.70e-06   1.94e-06    1.91    0.057 .
## intell       9.33e-02   1.26e-02    7.43  1.5e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.23 on 2396 degrees of freedom
## Multiple R-squared:  0.0942, Adjusted R-squared:  0.0927
## F-statistic: 62.3 on 4 and 2396 DF,  p-value: <2e-16
```

The interpretation of the coefficients are as follow.

1) The estimate of the coefficient of educ is 7.03e-02. This implies that holding other things equal, an additional year of education completed by 2005 is estimated to have additional 7.03e-02 esteem score in terms of PC1 of all self-esteem scores.

2) The estimate of the coefficient of newspaper is 1.83e-01. This shows that holding other things equal, if anyone in the respondent's household regularly read newspapers in 1979, the respondent obtains 1.83e-01 higher esteem score in terms of PC1 of all self-esteem scores than the one whose family does not read newspaper.

3) The estimate of the coefficient of inc_78 is 3.70e-06. This demonstrates that holding other things equal, when a respondent receive one thousand dollars of income in an annual year, he or she is estimated to get 1.94e-03 more esteem score in terms of PC1 of all self-esteem scores.

4) The estimate of the coefficient of intell is 9.33e-02. This indicates that holding other things equal, when the intelligence score increases by 1, the respondant's esteem score is estimated to increase by 9.33e-02.

2 Case study 2: Breast cancer sub-type

The Cancer Genome Atlas (TCGA), a landmark cancer genomics program by National Cancer Institute (NCI), molecularly characterized over 20,000 primary cancer and matched normal samples spanning 33 cancer types. The genome data is open to public from the Genomic Data Commons Data Portal (GDC).

In this study, we focus on 4 sub-types of breast cancer (BRCA): basal-like (basal), Luminal A-like (lumA), Luminal B-like (lumB), HER2-enriched. The sub-type is based on PAM50, a clinical-grade luminal-basal classifier.

- Luminal A cancers are low-grade, tend to grow slowly and have the best prognosis.
- Luminal B cancers generally grow slightly faster than luminal A cancers and their prognosis is slightly worse.
- HER2-enriched cancers tend to grow faster than luminal cancers and can have a worse prognosis, but they are often successfully treated with targeted therapies aimed at the HER2 protein.
- Basal-like breast cancers or triple negative breast cancers do not have the three receptors that the other sub-types have so have fewer treatment options.

We will try to use mRNA expression data alone without the labels to classify 4 sub-types. Classification without labels or prediction without outcomes is called unsupervised learning. We will use K-means and spectrum clustering to cluster the mRNA data and see whether the sub-type can be separated through mRNA data.

We first read the data using `data.table::fread()` which is a faster way to read in big data than `read.csv()`.

```
brca <- fread("data/brca_subtype.csv")

# get the sub-type information
brca_subtype <- brca$BRCA_Subtype_PAM50
```

1. Summary and transformation

    a) How many patients are there in each sub-type?

```
table(brca_subtype)
```

```
## brca_subtype
## Basal  Her2  LumA  LumB
##   208    91   628   233
```

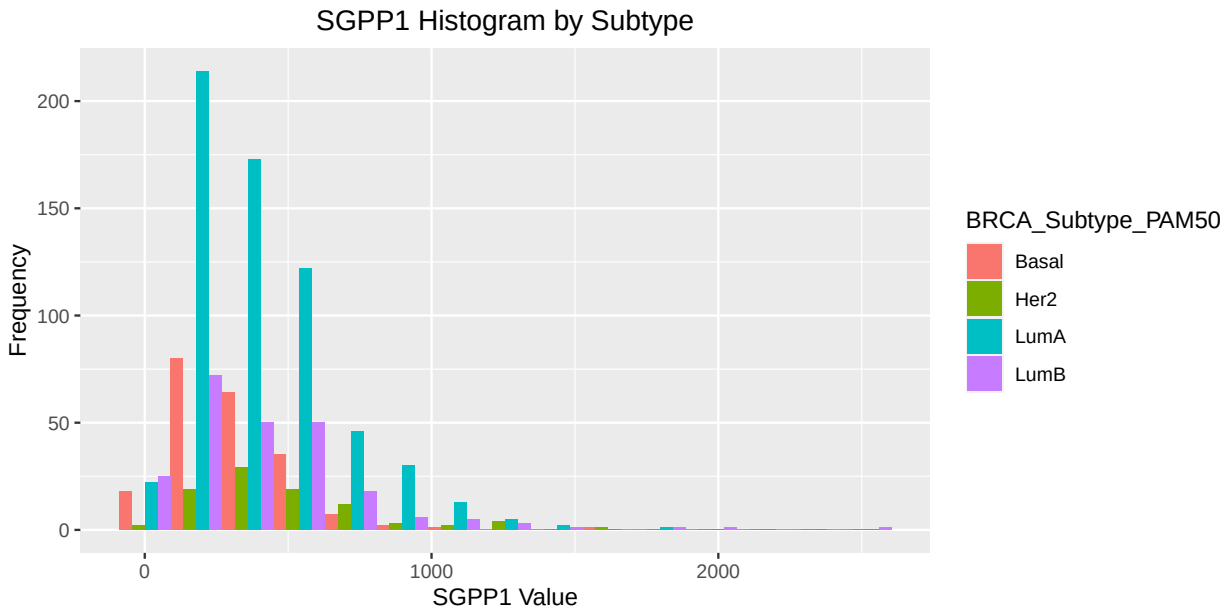
There are 628 LumA, 233 LumB, 91 Her2, and 208 Basal.

b) Randomly pick 5 genes and plot the histogram by each sub-type.

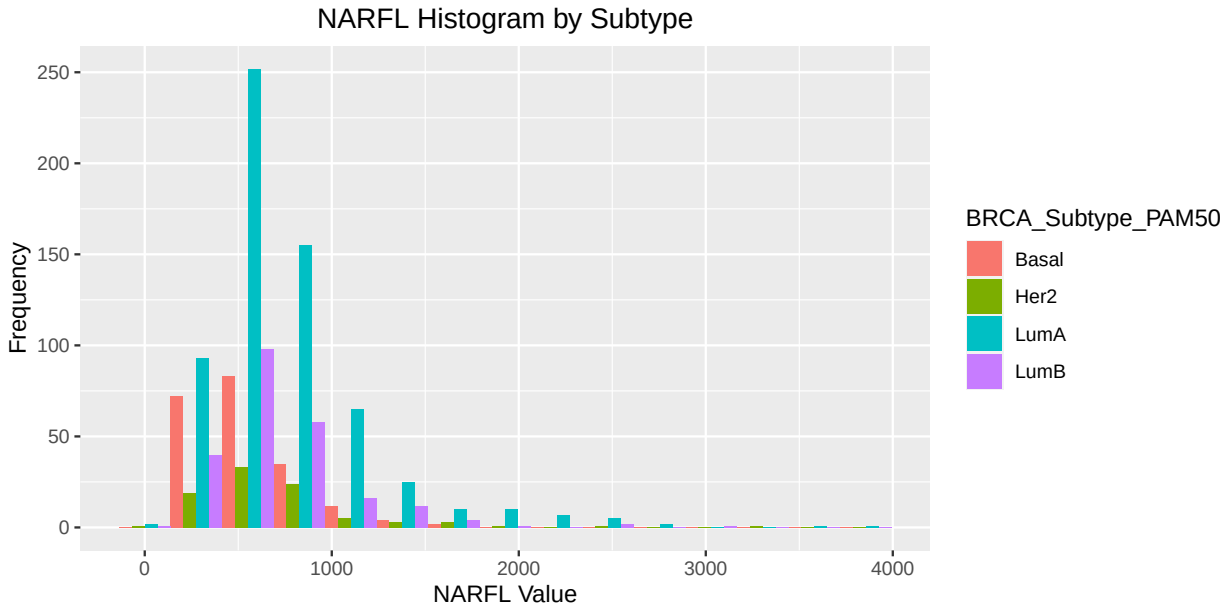
```
set.seed(5)
selected_columns <- sample(colnames(brca[,-1]), 5)
selected_columns
```

```
## [1] "SGPP1"     "NARFL"     "C10orf120" "ABI2"      "MAP1LC3B"
```

```
randgene1<-brca[, c("BRCA_Subtype_PAM50", "SGPP1")]

ggplot(randgene1, aes(x = SGPP1, fill = BRCA_Subtype_PAM50)) +
  geom_histogram(bins = 15, position = "dodge") +
  labs(title = "SGPP1 Histogram by Subtype", x = "SGPP1 Value", y = "Frequency") +
theme(plot.title = element_text(hjust = 0.5))
```

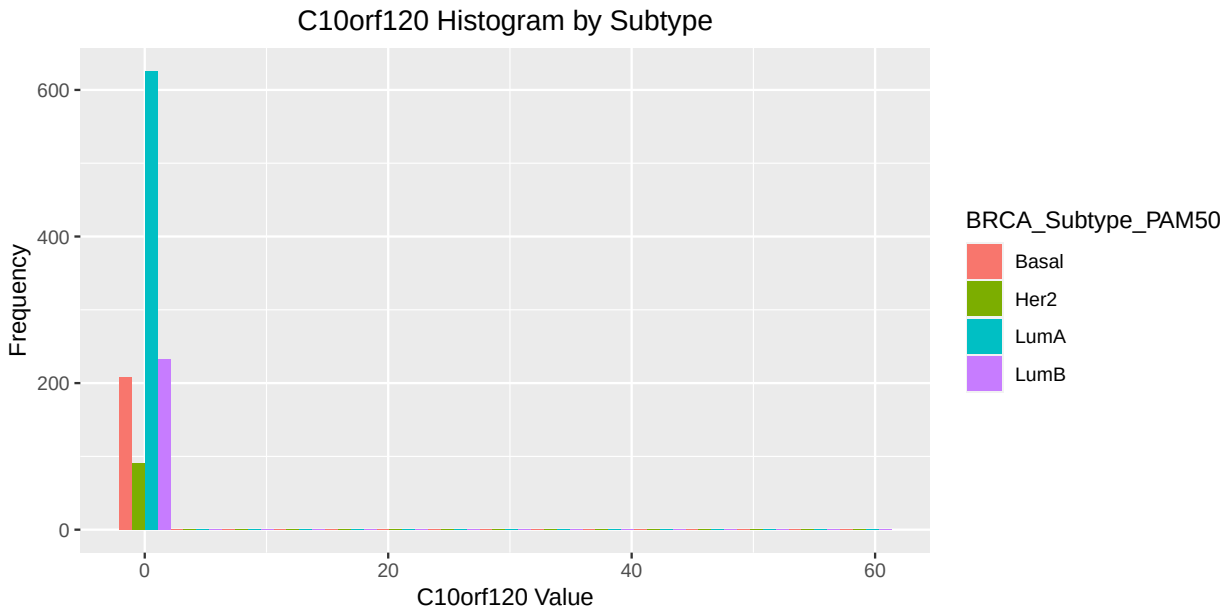



```
randgene2<-brca[, c("BRCA_Subtype_PAM50", "NARFL")]

ggplot(randgene2, aes(x = NARFL, fill = BRCA_Subtype_PAM50)) +
  geom_histogram(bins = 15, position = "dodge") +
  labs(title = "NARFL Histogram by Subtype", x = "NARFL Value", y = "Frequency") +
theme(plot.title = element_text(hjust = 0.5))
```

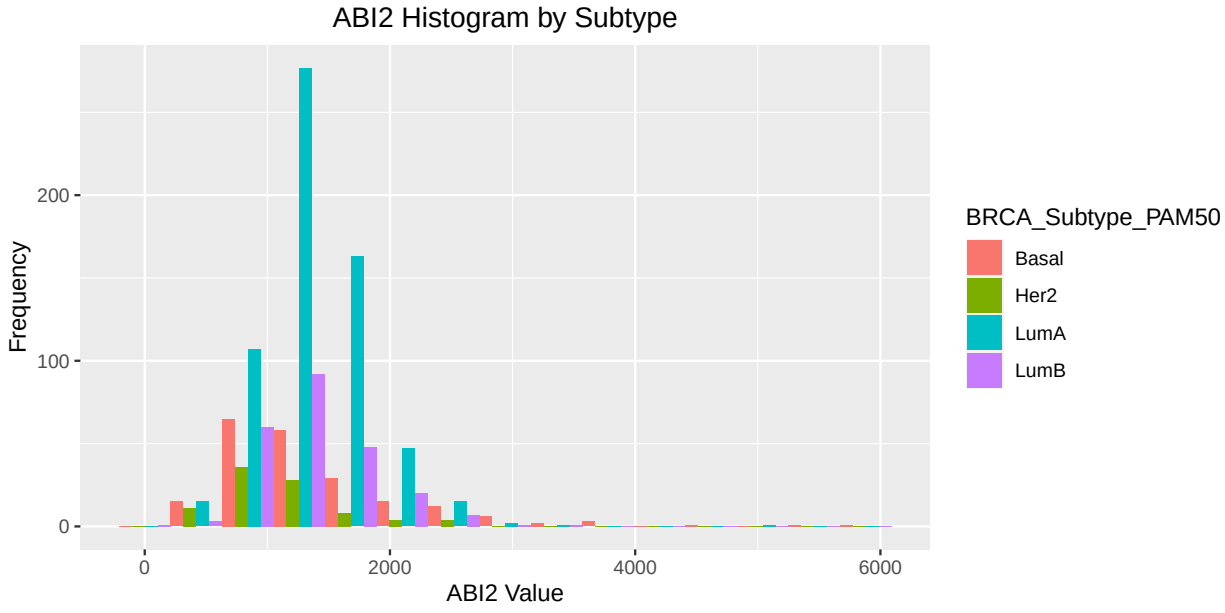



```
randgene3<-brca[, c("BRCA_Subtype_PAM50", "C10orf120")]

ggplot(randgene3, aes(x = C10orf120, fill = BRCA_Subtype_PAM50)) +
  geom_histogram(bins = 15, position = "dodge") +
  labs(title = "C10orf120 Histogram by Subtype", x = "C10orf120 Value", y = "Frequency") +
```

```
theme(plot.title = element_text(hjust = 0.5))
```

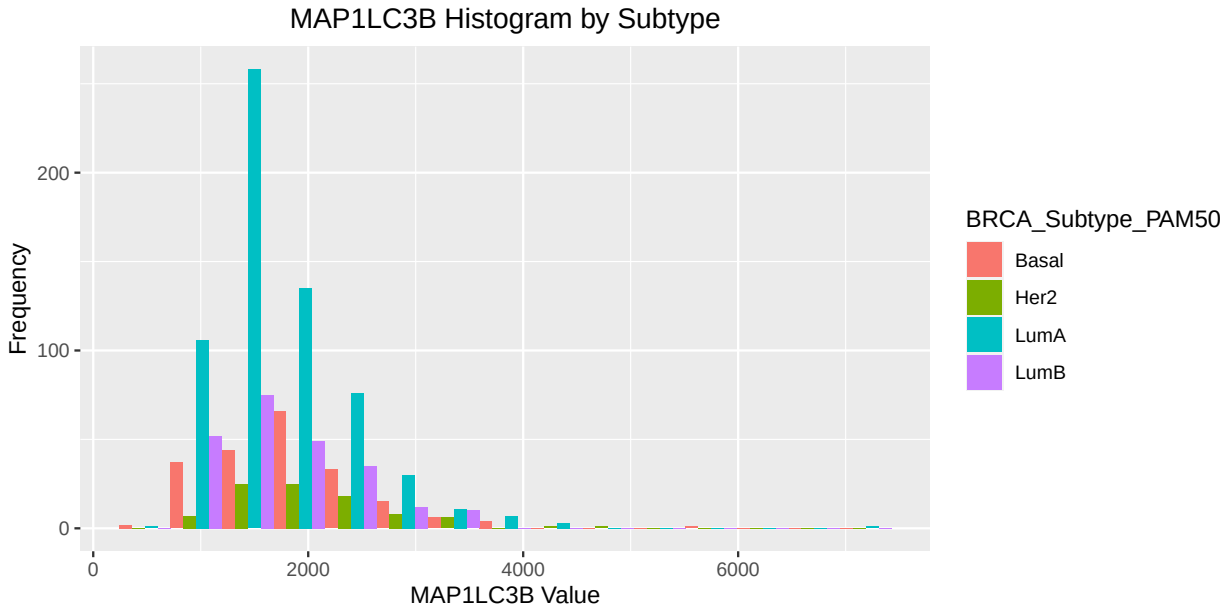



```
randgene4<-brca[, c("BRCA_Subtype_PAM50", "ABI2")]

ggplot(randgene4, aes(x = ABI2, fill = BRCA_Subtype_PAM50)) +
  geom_histogram(bins = 15, position = "dodge") +
  labs(title = "ABI2 Histogram by Subtype", x = "ABI2 Value", y = "Frequency") +
theme(plot.title = element_text(hjust = 0.5))
```




```
randgene5<-brca[, c("BRCA_Subtype_PAM50", "MAP1LC3B")]
```

```
ggplot(randgene5, aes(x = MAP1LC3B, fill = BRCA_Subtype_PAM50)) +
  geom_histogram(bins = 15, position = "dodge") +
  labs(title = "MAP1LC3B Histogram by Subtype", x = "MAP1LC3B Value", y = "Frequency") +
theme(plot.title = element_text(hjust = 0.5))
```



c) Remove gene with zero count and no variability. Then apply logarithmic transform.

```
brca <- brca[,-1]
dim(brca)

sel_cols <- which(colSums(abs(brca)) != 0)
new_brca <- brca[, sel_cols, with=F]

log_brca <- log2(as.matrix(new_brca+1e-10))

nzv <- nearZeroVar(log_brca)
log_brca <- log_brca[, -nzv]

dim(log_brca)
```

```
## [1]  1160 19947
## [1]  1160 18587
```

2. Apply kmeans on the transformed dataset with 4 centers and output the discrepancy table between the real sub-type `brca_subtype` and the cluster labels.

```
kmean_brca <- kmeans(log_brca, centers = 4)
str(kmean_brca)
```

```
## List of 9
```

```
##  $ cluster     : int [1:1160] 3 3 2 3 3 3 3 2 3 3 ...
##  $ centers     : num [1:4, 1:18587] 6.11 7.31 7.28 6.42 14.85 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:18587] "A1BG" "A2M" "NAT1" "NAT2" ...
##  $ totss       : num 8.34e+08
##  $ withinss    : num [1:4] 7.51e+07 2.16e+08 3.29e+08 1.47e+08
##  $ tot.withinss: num 7.68e+08
##  $ betweenss   : num 66553999
##  $ size        : int [1:4] 133 278 543 206
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
cluster <- as.factor(kmean_brca$cluster)
discrepancy_table <- table(brca_subtype, cluster)
discrepancy_table
```

```
##             cluster
## brca_subtype   1   2   3   4
##        Basal  17   3   1 187
##        Her2    9  23  42  17
##        LumA   85 147 396   0
##        LumB   22 105 104   2
```

3. Spectrum clustering: to scale or not to scale?

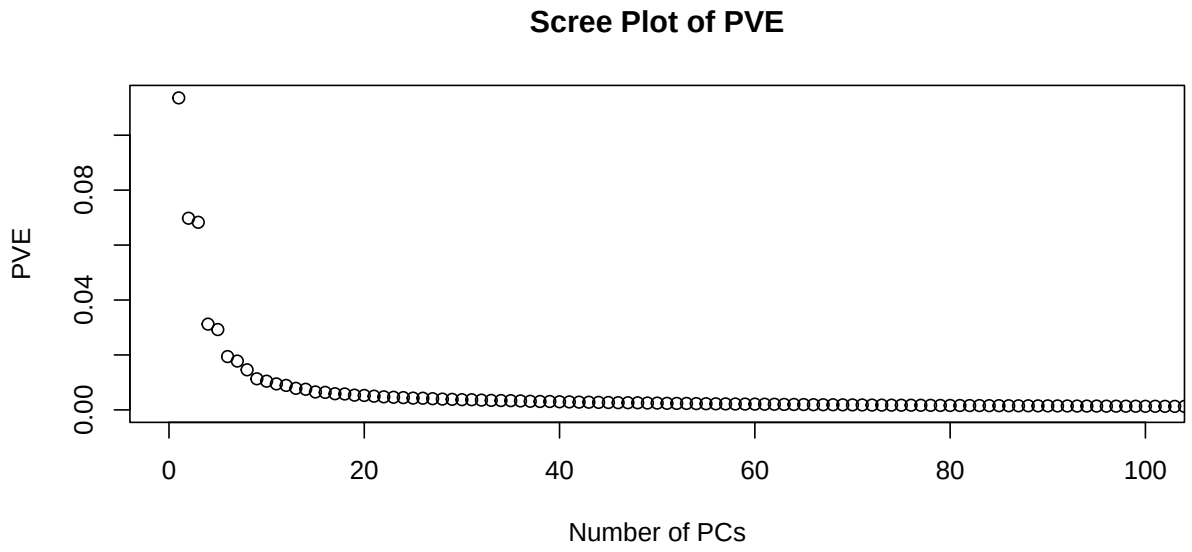
    a) Apply PCA on the centered and scaled dataset. How many PCs should we use and why? You are encouraged to use `irlba::irlba()`.

```
center_sacle_log_brca<-scale(as.matrix(log_brca), center = T, scale = T)
pca_fit1 <-prcomp(center_sacle_log_brca)
summary(pca_fit1)$importance[,1:50]
```

```
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     45.948 36.0118 35.6308 24.0886 23.3160 18.9934 18.1794
## Proportion of Variance  0.114  0.0698  0.0683  0.0312  0.0293  0.0194  0.0178
## Cumulative Proportion   0.114  0.1834  0.2517  0.2829  0.3121  0.3315  0.3493
##                            PC8     PC9    PC10     PC11     PC12     PC13
## Standard deviation     16.4701 14.5052 13.9364 13.26093 12.86785 12.07662
## Proportion of Variance  0.0146  0.0113  0.0104  0.00946  0.00891  0.00785
## Cumulative Proportion   0.3639  0.3752  0.3857  0.39514  0.40405  0.41190
##                           PC14     PC15     PC16     PC17     PC18    PC19
## Standard deviation     11.81444 11.00478 10.87899 10.45855 10.37345 9.98432
## Proportion of Variance  0.00751  0.00652  0.00637  0.00588  0.00579 0.00536
## Cumulative Proportion   0.41940  0.42592  0.43229  0.43817  0.44396 0.44933
##                           PC20    PC21    PC22   PC23    PC24    PC25    PC26
## Standard deviation     9.88443 9.63258 9.40588 9.2472 9.11637 8.93277 8.88580
## Proportion of Variance 0.00526 0.00499 0.00476 0.0046 0.00447 0.00429 0.00425
## Cumulative Proportion  0.45458 0.45957 0.46433 0.4689 0.47341 0.47770 0.48195
##                           PC27    PC28    PC29    PC30   PC31    PC32    PC33
## Standard deviation     8.70856 8.55171 8.44412 8.32013 8.2894 8.13055 8.01585
```

```
## Proportion of Variance 0.00408 0.00393 0.00384 0.00372 0.0037 0.00356 0.00346
## Cumulative Proportion  0.48603 0.48996 0.49380 0.49752 0.5012 0.50478 0.50823
##                           PC34    PC35    PC36    PC37    PC38    PC39    PC40
## Standard deviation     7.95976 7.90149 7.78845 7.62956 7.54286 7.52672 7.40871
## Proportion of Variance 0.00341 0.00336 0.00326 0.00313 0.00306 0.00305 0.00295
## Cumulative Proportion  0.51164 0.51500 0.51826 0.52140 0.52446 0.52750 0.53046
##                           PC41    PC42    PC43    PC44    PC45    PC46    PC47
## Standard deviation     7.35486 7.24217 7.20749 7.12331 7.07602 6.98526 6.96788
## Proportion of Variance 0.00291 0.00282 0.00279 0.00273 0.00269 0.00263 0.00261
## Cumulative Proportion  0.53337 0.53619 0.53898 0.54171 0.54441 0.54703 0.54965
##                           PC48   PC49    PC50
## Standard deviation     6.94236 6.8181 6.78557
## Proportion of Variance 0.00259 0.0025 0.00248
## Cumulative Proportion  0.55224 0.5547 0.55722
```

```
plot(summary(pca_fit1)$importance[2, ],
ylab="PVE",
xlab="Number of PCs",
xlim=c(0,100),
main="Scree Plot of PVE")
```



We should use 8 principal components according to the elbow rules, because after 8 principal components PVE grows very slowly .

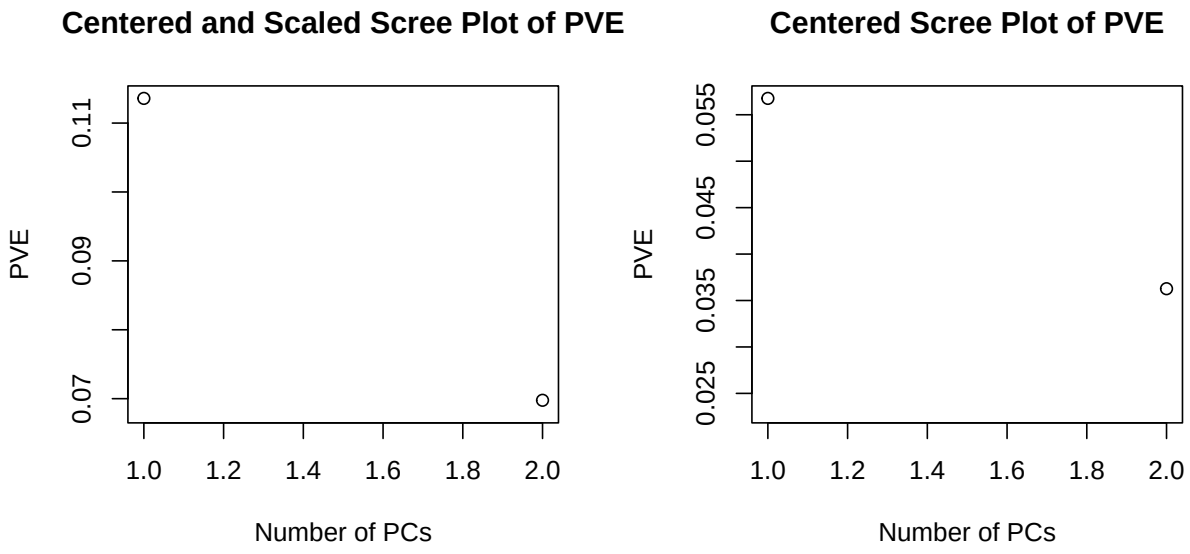
b) Plot PC1 vs PC2 of the centered and scaled data and PC1 vs PC2 of the centered but unscaled data sid

```
center_log_brca<-scale(as.matrix(log_brca), center=T, scale=F)
pca_fit2 <-prcomp(center_log_brca)
summary(pca_fit2)$importance[,1:50]
```

```
##                             PC1      PC2      PC3      PC4     PC5     PC6
## Standard deviation     202.1158 161.5752 129.1453 101.1324 96.9010 92.3639
```

```
## Proportion of Variance   0.0568   0.0363   0.0232   0.0142  0.0131  0.0118
## Cumulative Proportion    0.0568   0.0930   0.1162   0.1304  0.1435  0.1553
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     80.77410 75.97130 73.99719 64.91974 62.83309 61.67422
## Proportion of Variance  0.00907  0.00802  0.00761  0.00586  0.00549  0.00528
## Cumulative Proportion   0.16438  0.17240  0.18001  0.18586  0.19135  0.19663
##                            PC13     PC14    PC15     PC16     PC17     PC18
## Standard deviation     59.50287 56.51087 54.2902 53.42965 50.98997 49.75279
## Proportion of Variance  0.00492  0.00444  0.0041  0.00397  0.00361  0.00344
## Cumulative Proportion   0.20155  0.20599  0.2101  0.21405  0.21766  0.22110
##                            PC19     PC20   PC21     PC22     PC23     PC24
## Standard deviation     48.48054 47.12730 46.475 45.97179 45.73422 44.60989
## Proportion of Variance  0.00327  0.00309  0.003  0.00294  0.00291  0.00276
## Cumulative Proportion   0.22437  0.22745  0.230  0.23339  0.23630  0.23906
##                            PC25     PC26     PC27     PC28     PC29     PC30
## Standard deviation     44.30007 43.43974 42.75269 42.07903 41.65722 41.20326
## Proportion of Variance  0.00273  0.00262  0.00254  0.00246  0.00241  0.00236
## Cumulative Proportion   0.24179  0.24441  0.24695  0.24941  0.25182  0.25418
##                            PC31     PC32     PC33     PC34     PC35     PC36
## Standard deviation     40.73873 40.26344 40.06433 39.96894 39.69904 39.51995
## Proportion of Variance  0.00231  0.00225  0.00223  0.00222  0.00219  0.00217
## Cumulative Proportion   0.25648  0.25874  0.26097  0.26319  0.26538  0.26755
##                            PC37     PC38     PC39     PC40     PC41   PC42
## Standard deviation     39.23025 39.19610 38.76419 38.37802 38.19777 37.918
## Proportion of Variance  0.00214  0.00213  0.00209  0.00205  0.00203  0.002
## Cumulative Proportion   0.26968  0.27182  0.27391  0.27595  0.27798  0.280
##                            PC43     PC44     PC45     PC46     PC47    PC48
## Standard deviation     37.86963 37.66253 37.61954 37.31803 37.23415 37.0045
## Proportion of Variance  0.00199  0.00197  0.00197  0.00193  0.00193  0.0019
## Cumulative Proportion   0.28197  0.28394  0.28591  0.28784  0.28977  0.2917
##                            PC49     PC50
## Standard deviation     36.84626 36.62321
## Proportion of Variance  0.00189  0.00186
## Cumulative Proportion   0.29356  0.29542
```

```r
par(mfrow=c(1,2))

p1<-plot(summary(pca_fit1)$importance[2, 1:3],
ylab="PVE",
xlab="Number of PCs",
xlim=c(1,2),
main="Centered and Scaled Scree Plot of PVE")

p2<-plot(summary(pca_fit2)$importance[2, 1:3],
ylab="PVE",
xlab="Number of PCs",
xlim=c(1,2),
main="Centered Scree Plot of PVE")
```




```
par(mfrow=c(1,1))
```

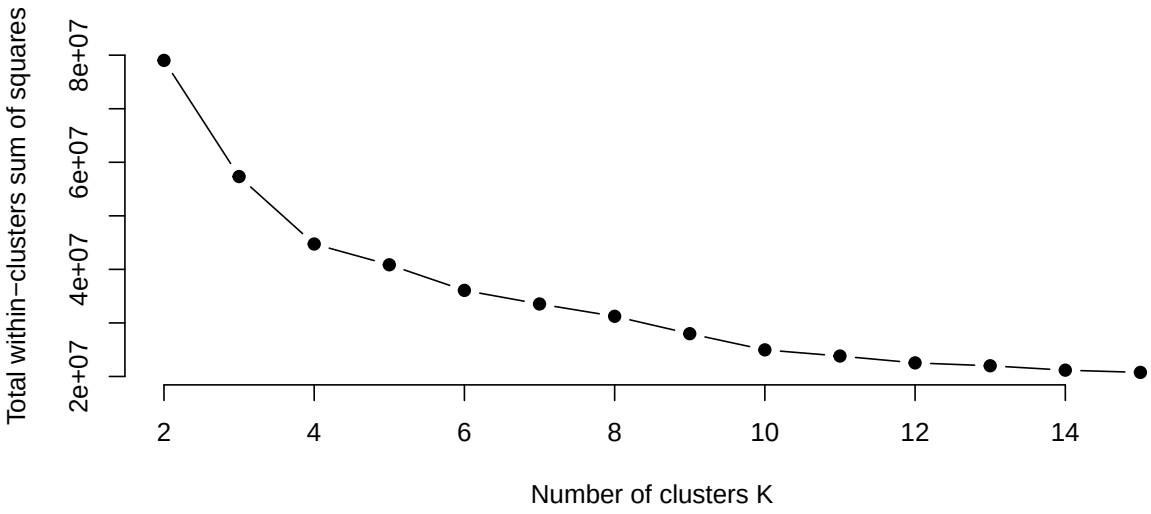
From these two plots above, we can see that both PC1 and PC2 in the centered and scaled PCA explains more PVE than PC1 and PC2 in the centered PCA respectively, so we should scale in the clustering process.

4. Spectrum clustering: center but do not scale the data

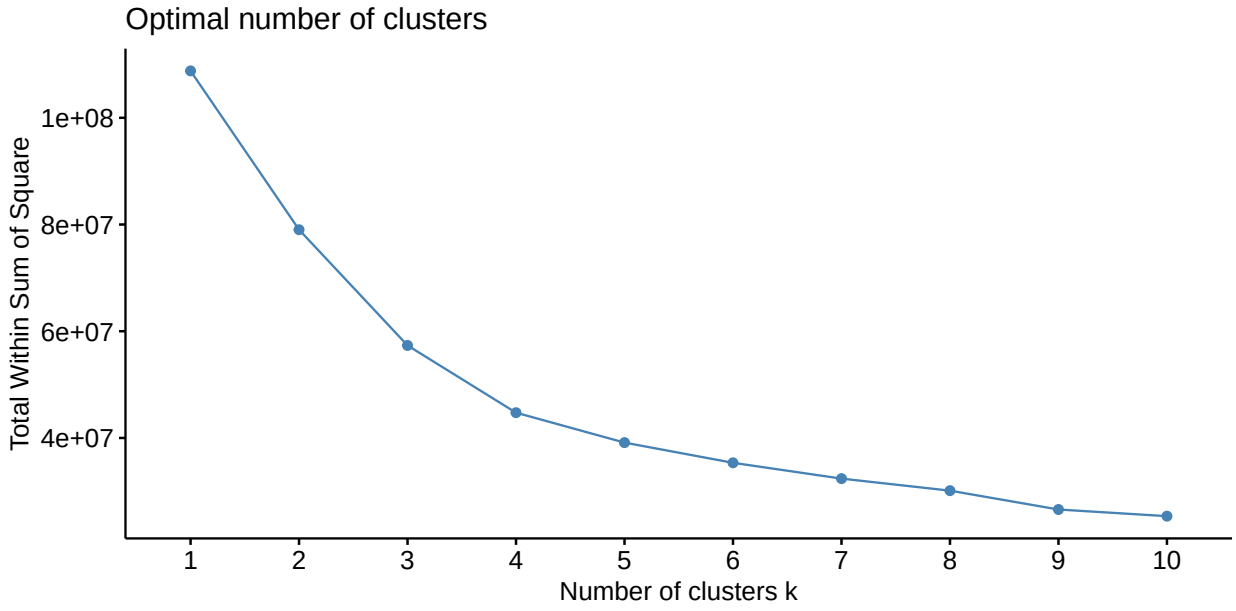
    a) Use the first 4 PCs of the centered and unscaled data and apply kmeans. Find a reasonable number of clusters using within sum of squared with the elbow rule.

```
fourpc<-pca_fit2$x[, 1:4]
```

```
set.seed(0)

wss <- function(df, k) {
kmeans(df, k, nstart = 10)$tot.withinss
}

k.values <- 2:15
wss_values <- sapply(k.values, function(k) kmeans(fourpc, centers = k)$tot.withinss)

plot(k.values, wss_values,
type="b", pch = 19, frame = FALSE,
xlab="Number of clusters K",
ylab="Total within-clusters sum of squares")
```




```
library(factoextra)
fviz_nbclust(fourpc, kmeans, method = "wss")
```



According to the elbow method, we decide to use k=4 to have four clusters.

b) Choose an optimal cluster number and apply kmeans. Compare the real sub-type and the clustering label

Just as above, we choose k=4.

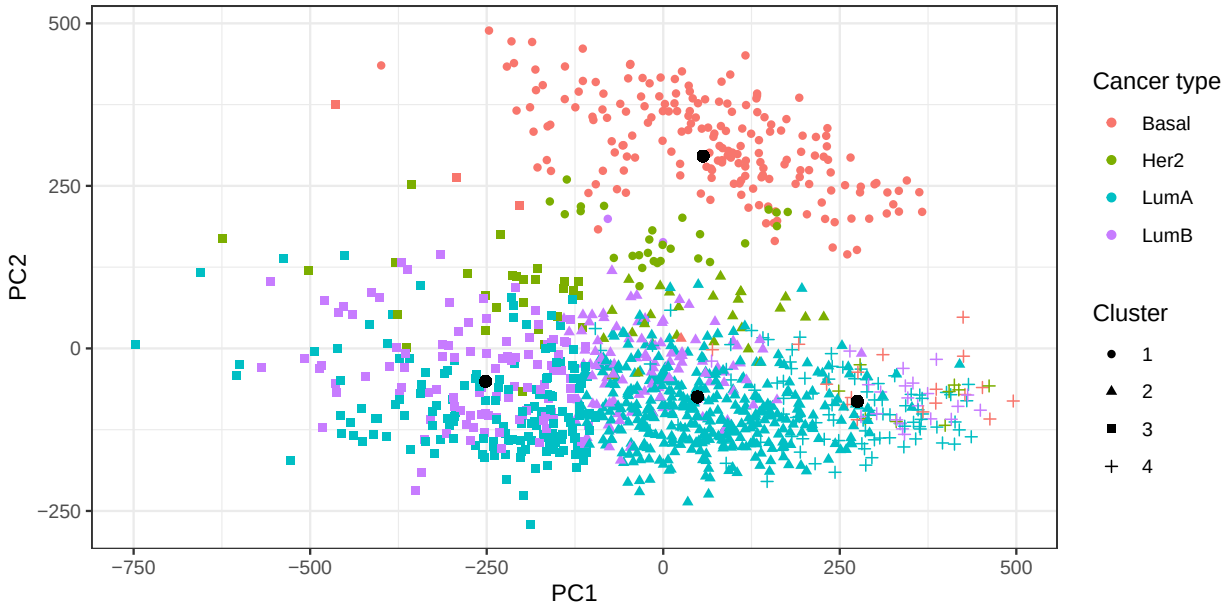
```
kmean_brca2 <- kmeans(fourpc, centers = 4)
str(kmean_brca2)
kmean_brca2$center
```

```
## List of 9
##  $ cluster     : int [1:1160] 2 2 3 2 2 2 2 3 2 2 ...
##  $ centers     : num [1:4, 1:4] 56.6 48.5 -251.6 274.9 295.7 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##  $ totss       : num 1.09e+08
##  $ withinss    : num [1:4] 9494611 14676441 12900480 7665179
##  $ tot.withinss: num 44736710
##  $ betweenss   : num 64051190
##  $ size        : int [1:4] 217 504 299 140
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
##      PC1   PC2     PC3    PC4
## 1   56.6 295.7    2.79 -3.100
## 2   48.5 -74.6   71.33  3.869
## 3 -251.6 -50.7  -21.63 -4.473
## 4  274.9 -81.4 -214.91  0.431
```

```
cluster <- as.factor(kmean_brca2$cluster)
data<-data.frame(fourpc[,1:2])
data2<-data.frame(data, type = brca_subtype, label =  cluster)
kmean_brca2$centers[,1:2]
```

```
##      PC1   PC2
## 1   56.6 295.7
## 2   48.5 -74.6
## 3 -251.6 -50.7
## 4  274.9 -81.4
```

```
p<-data.table(x = data2$PC1,
                y = data2$PC2,
                col = as.factor(data2$type),
                cl = data2$label,
                pc1=kmean_brca2$centers[,1],
                pc2=kmean_brca2$centers[,2]) %>%
  ggplot() +
  geom_point(aes(x = x, y = y, col = col, shape = cl)) +
  geom_point(aes(x = pc1, y = pc2), size = 2, fill = "black") +
  theme_bw() +
  labs(color = "Cancer type", shape = "Cluster") +
  xlab("PC1") +
  ylab("PC2")

p
```



Cluster 1 centers at the center of Basal; cluster 2 centers at around the center of LumA; cluster 3 centers at around the center of LumB; cluster 4 centers deviates quite a bit from the center of Her2. The clustering is good for 3 type of cancers, but not for Her2.

c) Compare the clustering result from applying kmeans to the original data and the clustering result fr

```
discrepancy_table
```

```
##              cluster
## brca_subtype   1   2   3   4
##        Basal  17   3   1 187
##        Her2    9  23  42  17
##        LumA   85 147 396   0
##        LumB   22 105 104   2
```

Looking at the above table from applying kmeans to the original data, we can see that only cluster 4 clearly points to Basal. Cluster 1, 2, 3 all have the highest number of points at LumA. After applying the PCA, the clustering does a better job differentiating cancer types. Original data has a high dimension. PCA helps reduce dimensionality, and having a smaller dimension is better for clustering. Also, since the principal components capture the most important features of the data, using PCA can help reduce the noise in the data.

d) Now we have an x patient with breast cancer but with unknown sub-type. We have this patient's mRNA s

```
x_patient <- fread("data/brca_x_patient.csv")
```

```
dim(x_patient)

loading1<-pca_fit2$rotation[,1]
loading2<-pca_fit2$rotation[,2]
loading1<-data.frame(loading1)
dim(loading1)
```

```
common_cols <- intersect(colnames(x_patient), rownames(loading1))
x_patient <- x_patient[, ..common_cols]
```

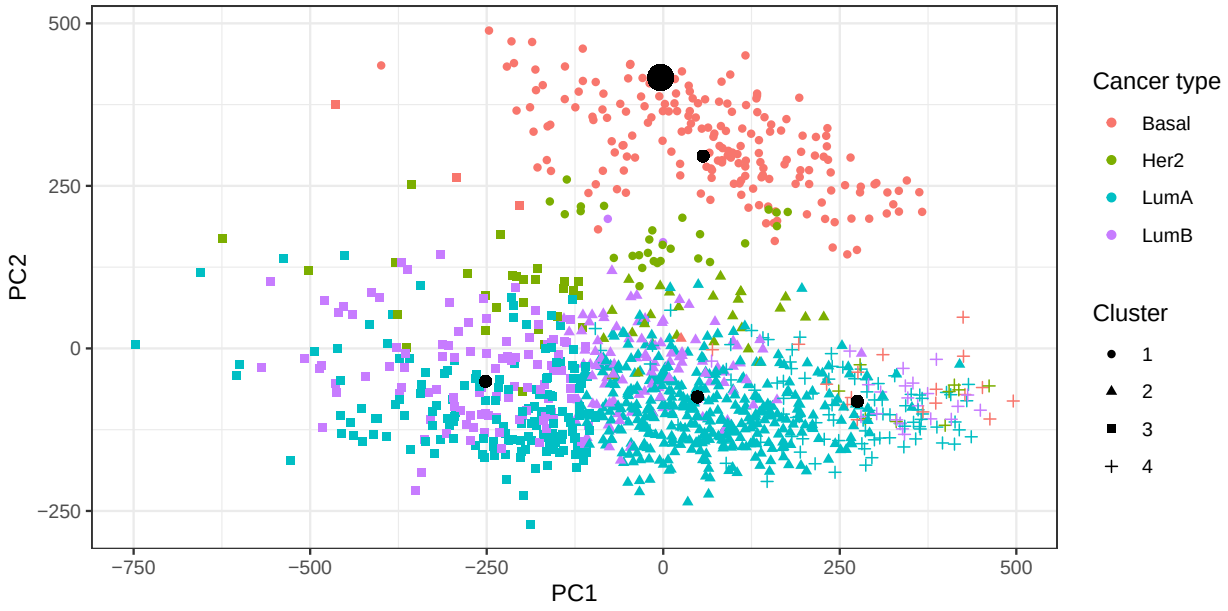
```
## [1]     1 19947
## [1] 18587     1
```

```
log_x_patient <- log2(as.matrix(x_patient+1e-10))
center_log_x_patient<-log_x_patient-colMeans(log_brca)
```

```
pc1score<-as.matrix(center_log_x_patient)%*%as.matrix(loading1)
pc2score<-as.matrix(center_log_x_patient)%*%as.matrix(loading2)
pc1score
pc2score
```

```
##      loading1
## [1,]    -3.75
##      [,1]
## [1,]  417
```

```
data.table(x = data2$PC1,
                y = data2$PC2,
                col = as.factor(data2$type),
                cl = data2$label,
                pc1=kmean_brca2$centers[,1],
                pc2=kmean_brca2$centers[,2],
                pc1score=pc1score,
                pc2score=pc2score )%>%
  ggplot() +
  geom_point(aes(x = x, y = y, col = col, shape = cl)) +
  geom_point(aes(x = pc1, y = pc2), size = 2) +
  geom_point(aes(x = pc1score, y = pc2score), size = 5) +
  theme_bw() +
  labs(color = "Cancer type", shape = "Cluster") +
  xlab("PC1") +
  ylab("PC2")
```



The big black dot is the patient. The small black dots are cluster centers.

```
#library(stats)
patient <- c(pc1score, pc2score)
centroids <- rbind(c(kmean_brca2$centers[,1][1], kmean_brca2$centers[,2][1]), c(kmean_brca2$centers[,1]
distances <- apply(centroids, 1, function(centroid) {
  dist(rbind(patient, centroid))
})
print(distances)
```

```
## [1] 135 494 529 571
```

The distance between the patient and cluster centers is the smallest when it is between the patient and cluster 1. So the patient might have Basal.

3 Case study 3: Auto data set

This question utilizes the `Auto` dataset from ISLR. The original dataset contains 408 observations about cars. It is similar to the CARS dataset that we use in our lectures. To get the data, first install the package ISLR. The `Auto` dataset should be loaded automatically. We'll use this dataset to practice the methods learn so far. Original data source is here: https://archive.ics.uci.edu/ml/datasets/auto+mpg

Get familiar with this dataset first. Tip: you can use the command `?ISLR::Auto` to view a description of the dataset.

3.1 EDA

Explore the data, with particular focus on pairwise plots and summary statistics. Briefly summarize your findings and any peculiarities in the data.

```
head(Auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```
summary(Auto)
```

```
##       mpg          cylinders      displacement   horsepower        weight
##  Min.   : 9.0   Min.   :3.00   Min.   : 68   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.0   1st Qu.:4.00   1st Qu.:105   1st Qu.: 75.0   1st Qu.:2225
##  Median :22.8   Median :4.00   Median :151   Median : 93.5   Median :2804
##  Mean   :23.4   Mean   :5.47   Mean   :194   Mean   :104.5   Mean   :2978
##  3rd Qu.:29.0   3rd Qu.:8.00   3rd Qu.:276   3rd Qu.:126.0   3rd Qu.:3615
##  Max.   :46.6   Max.   :8.00   Max.   :455   Max.   :230.0   Max.   :5140
##
##   acceleration       year        origin                     name
##  Min.   : 8.0   Min.   :70   Min.   :1.00   amc matador       :  5
##  1st Qu.:13.8   1st Qu.:73   1st Qu.:1.00   ford pinto        :  5
##  Median :15.5   Median :76   Median :1.00   toyota corolla    :  5
##  Mean   :15.5   Mean   :76   Mean   :1.58   amc gremlin       :  4
##  3rd Qu.:17.0   3rd Qu.:79   3rd Qu.:2.00   amc hornet        :  4
##  Max.   :24.8   Max.   :82   Max.   :3.00   chevrolet chevette:  4
##                                             (Other)           :365
```

We have in total 392 data in Auto and 9 variables. From the data description and the above summary, we have continuous columns including mpg, displacement, horsepower, weight, acceleration and multi-valued discrete variables including cylinders, year, origin, and a string column of name.

```
colSums(is.na(Auto))
```

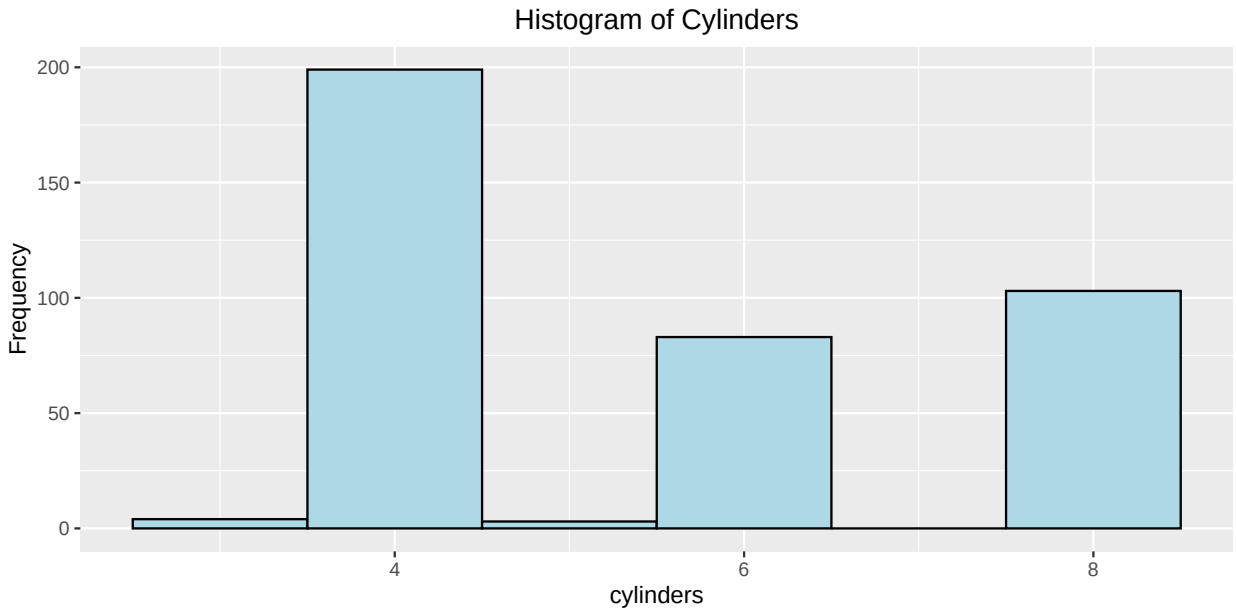
```
##          mpg    cylinders displacement   horsepower       weight acceleration
##            0            0            0            0            0            0
##         year       origin         name
##            0            0            0
```

There are no Null values in the Auto dataset. We then visualize continuous and discrete variables distribution to learn more about the dataset.

```
table(Auto$cylinders)
```

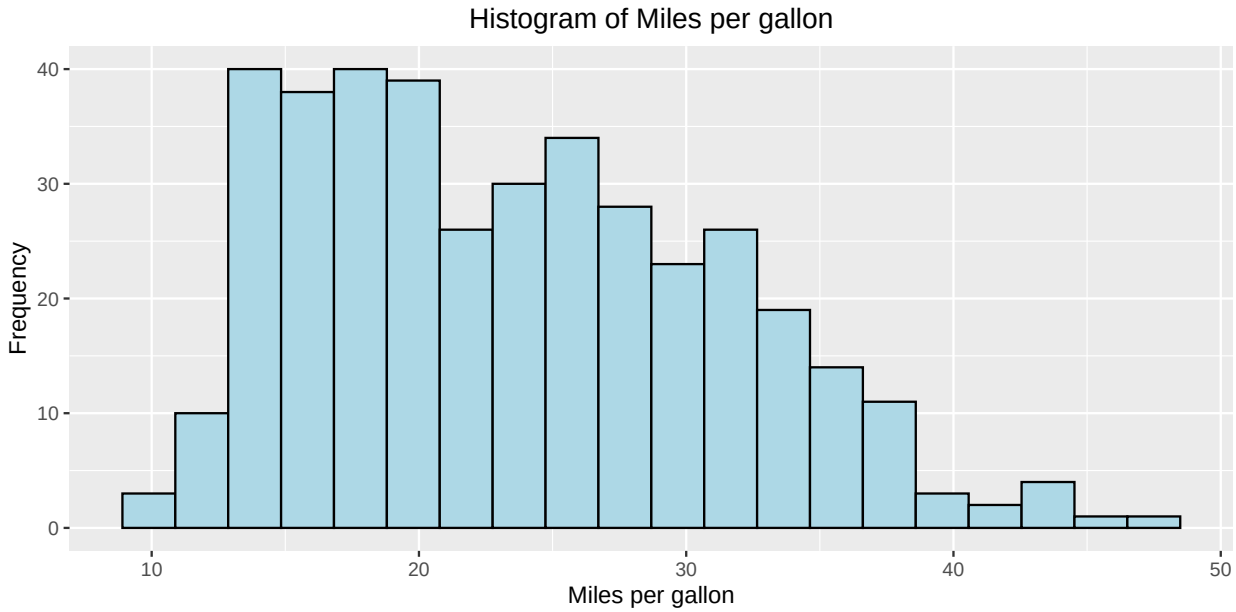
```
##
##   3   4   5   6   8
##   4 199   3  83 103
```

```
ggplot(Auto, aes(x=cylinders)) + geom_histogram(bins=6, fill="lightblue", color="black") +
  xlab("cylinders") + ylab("Frequency") + ggtitle("Histogram of Cylinders")  +
theme(plot.title = element_text(hjust = 0.5))
```



There are seldom car with cylinders number 3 and 5 and no 7 cylinders. Majority of cars have 4 cylinders while there are amount of cars have 6 or 8 cylinders.

```
ggplot(Auto, aes(x=mpg)) + geom_histogram(bins=20, fill="lightblue", color="black") +
  xlab("Miles per gallon") + ylab("Frequency") + ggtitle("Histogram of Miles per gallon")  +
  theme(plot.title = element_text(hjust = 0.5))
```



The above graph shows that a majority of cars have miles per gallon ranging from 16 to 20. The rest of mpg roughly satisfies the normal distribution with mean 26.

```
length(unique(Auto$name))
```

```
## [1] 301
```

We have 392 data points in total but with 301 names. Most of names only have one data point. This column will not be useful in our case.

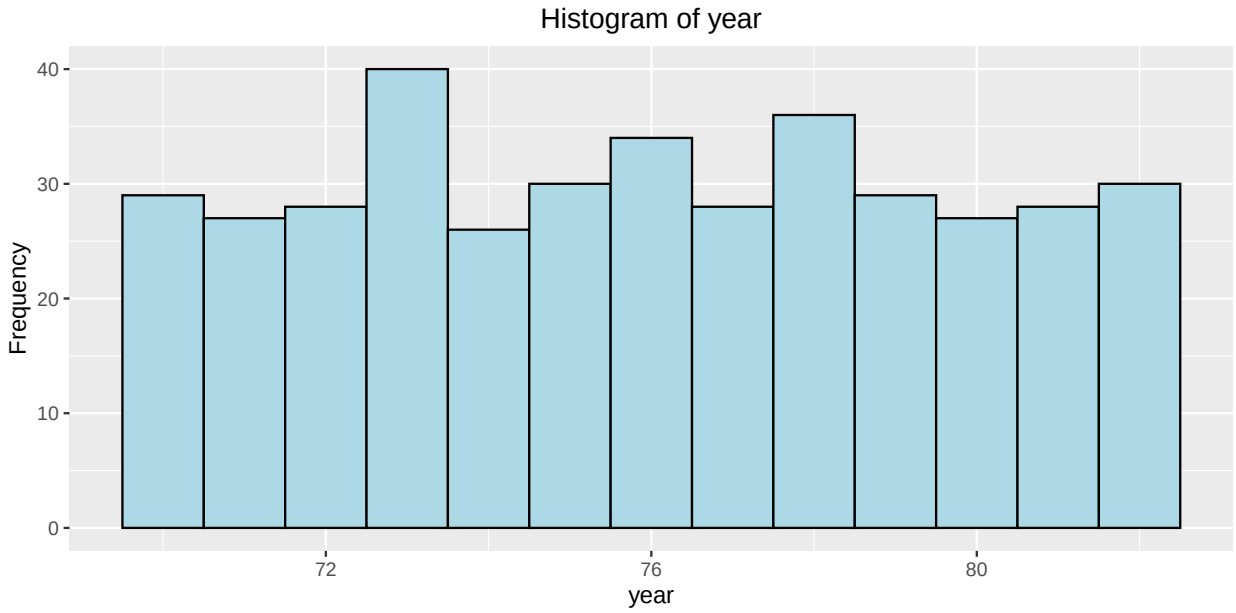
```
table(Auto$year)
```

```
##
## 70 71 72 73 74 75 76 77 78 79 80 81 82
## 29 27 28 40 26 30 34 28 36 29 27 28 30
```

```
length(unique(Auto$year))
```

```
## [1] 13
```

```
ggplot(Auto, aes(x=year)) + geom_histogram(bins=13, fill="lightblue", color="black") +
  xlab("year") + ylab("Frequency") + ggtitle("Histogram of year")  +
theme(plot.title = element_text(hjust = 0.5))
```



The samples evenly scattered across years.

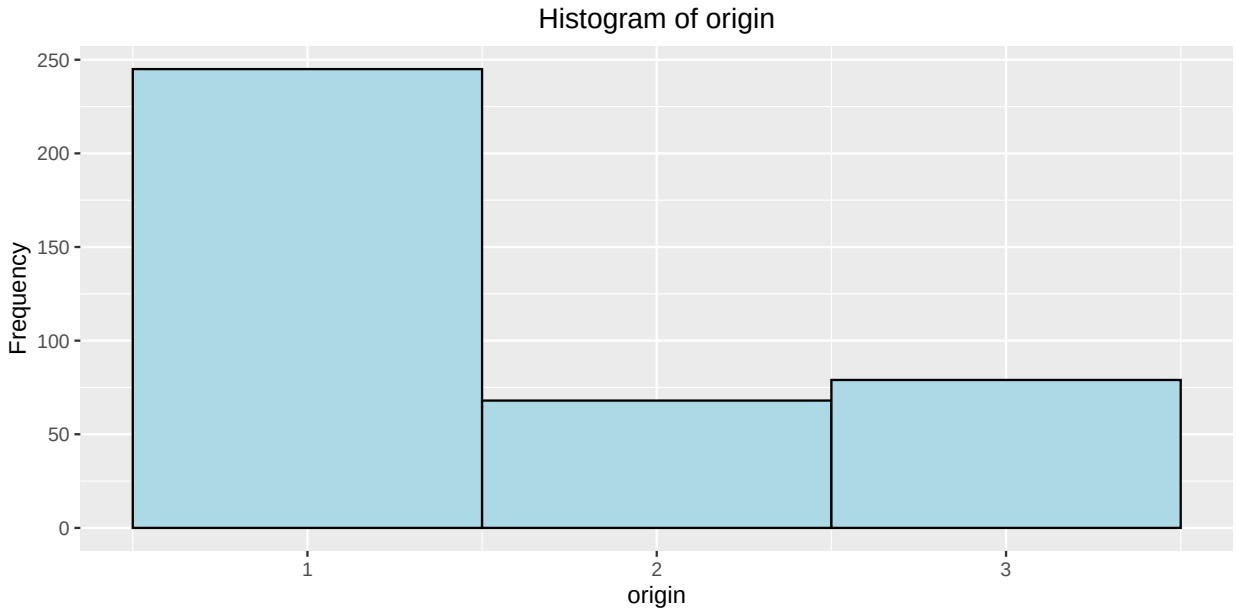
```
table(Auto$origin)
```

```
##
##   1   2   3
## 245  68  79
```

```
length(unique(Auto$origin))
```

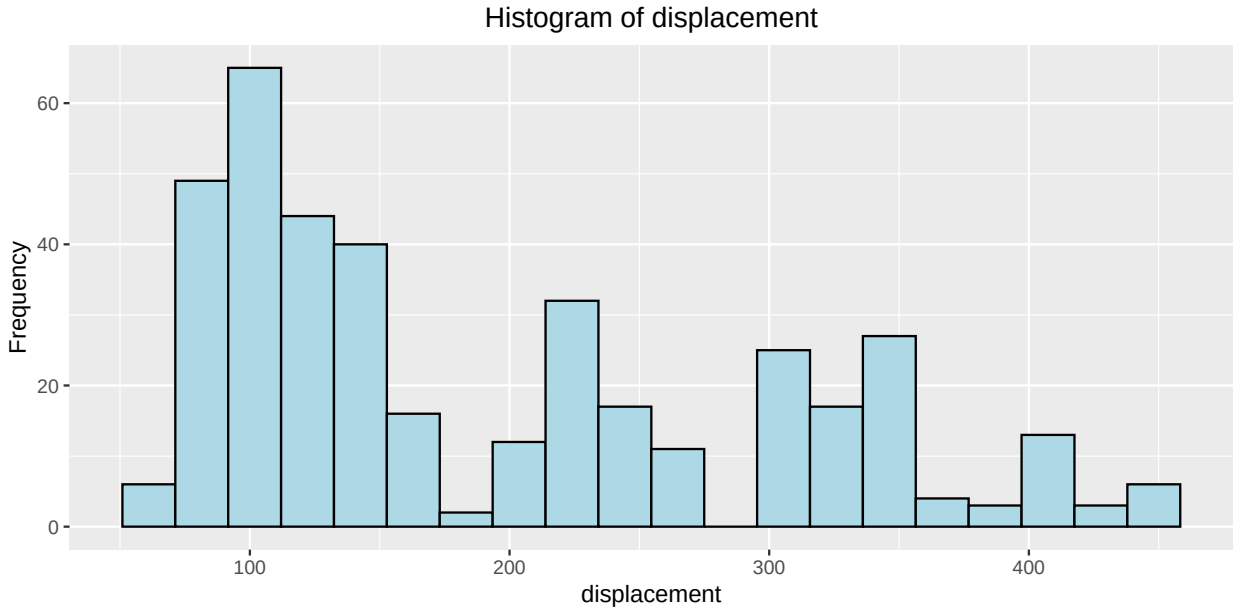
```
## [1] 3
```

```
ggplot(Auto, aes(x=origin)) + geom_histogram(bins=3, fill="lightblue", color="black") +
  xlab("origin") + ylab("Frequency") + ggtitle("Histogram of origin")  +
theme(plot.title = element_text(hjust = 0.5))
```



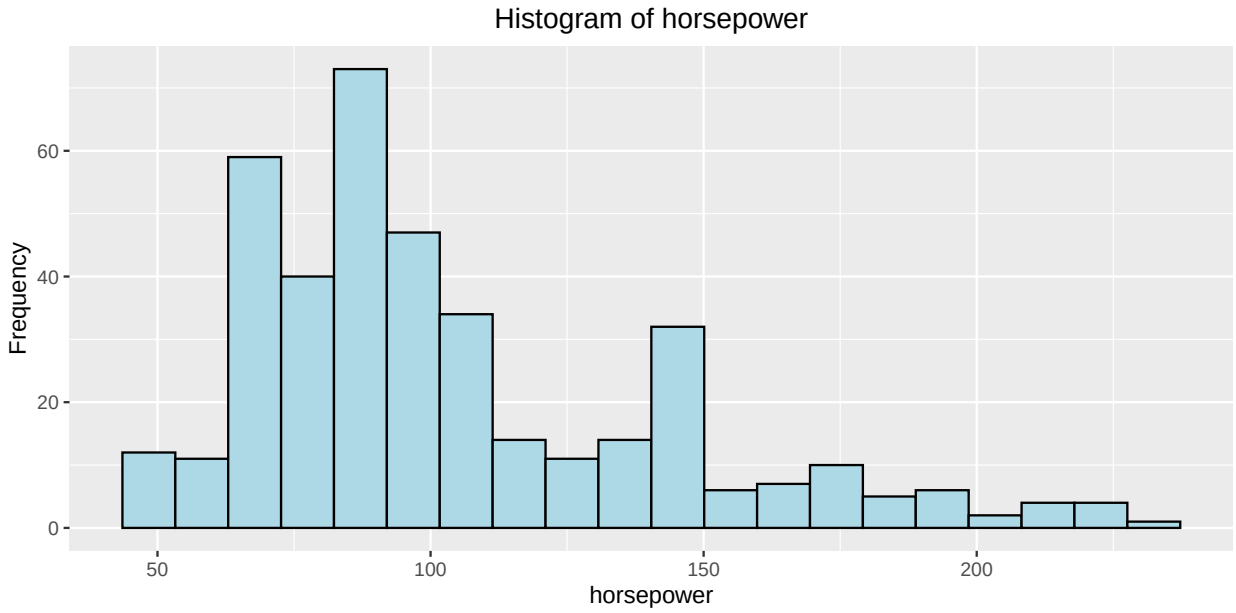
Most samples have origin 1.

```
ggplot(Auto, aes(x=displacement)) + geom_histogram(bins=20, fill="lightblue", color="black") +
  xlab("displacement") + ylab("Frequency") + ggtitle("Histogram of displacement")  +
theme(plot.title = element_text(hjust = 0.5))
```



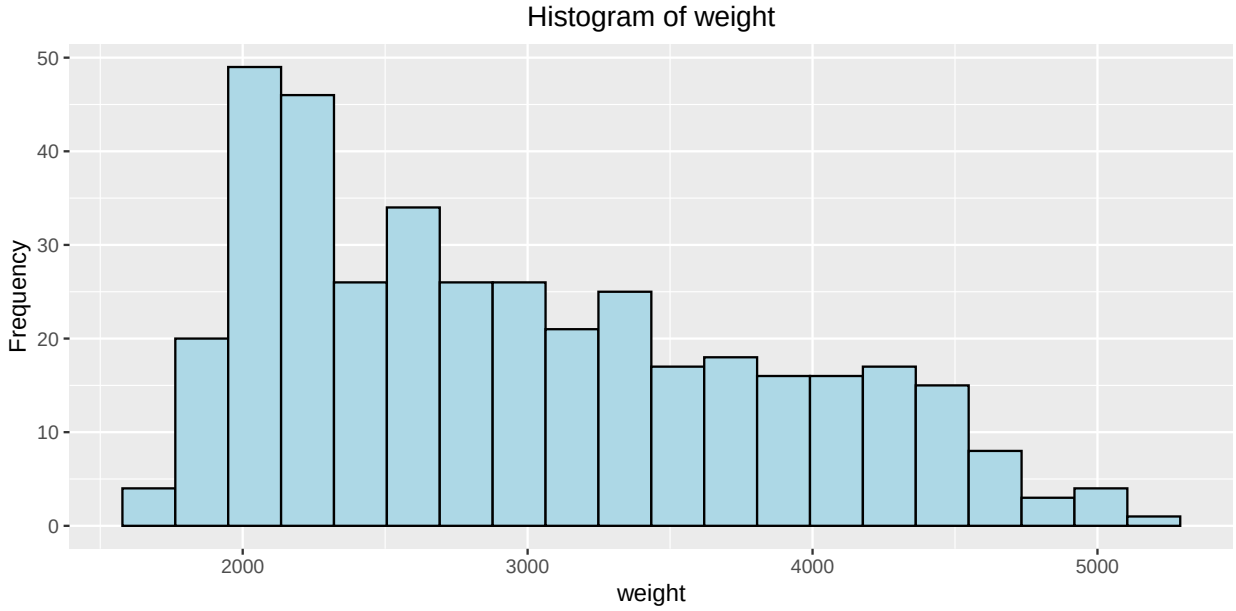
Most samples have displacement around 100. The number of sample decreases with the increase of displacement. But there lacks some samples with displacement from 150 to 200 and 240 to 300 so that the distribution satisfies the normal distribution

```
ggplot(Auto, aes(x=horsepower)) + geom_histogram(bins=20, fill="lightblue", color="black") +
  xlab("horsepower") + ylab("Frequency") + ggtitle("Histogram of horsepower")  +
theme(plot.title = element_text(hjust = 0.5))
```



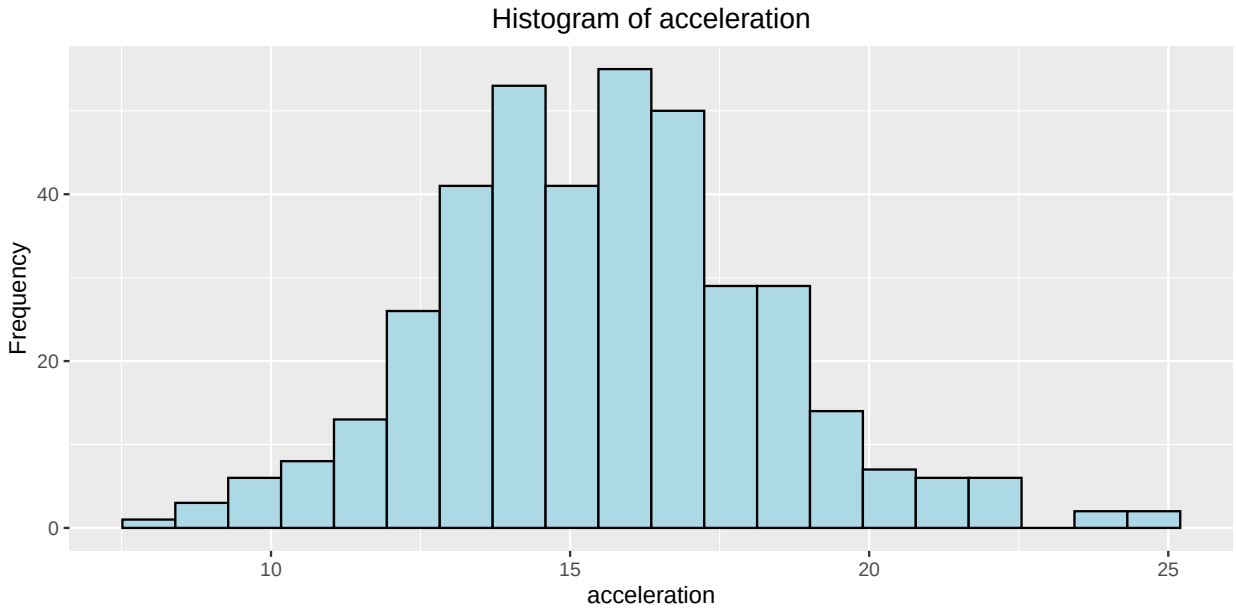
The horsepower are focused from 70 to 100.

```
ggplot(Auto, aes(x=weight)) + geom_histogram(bins=20, fill="lightblue", color="black") +
  xlab("weight") + ylab("Frequency") + ggtitle("Histogram of weight")  +
theme(plot.title = element_text(hjust = 0.5))
```



The weights are roughly evenly distributed from 2000 to 4000 but more samples have weights around 2000.

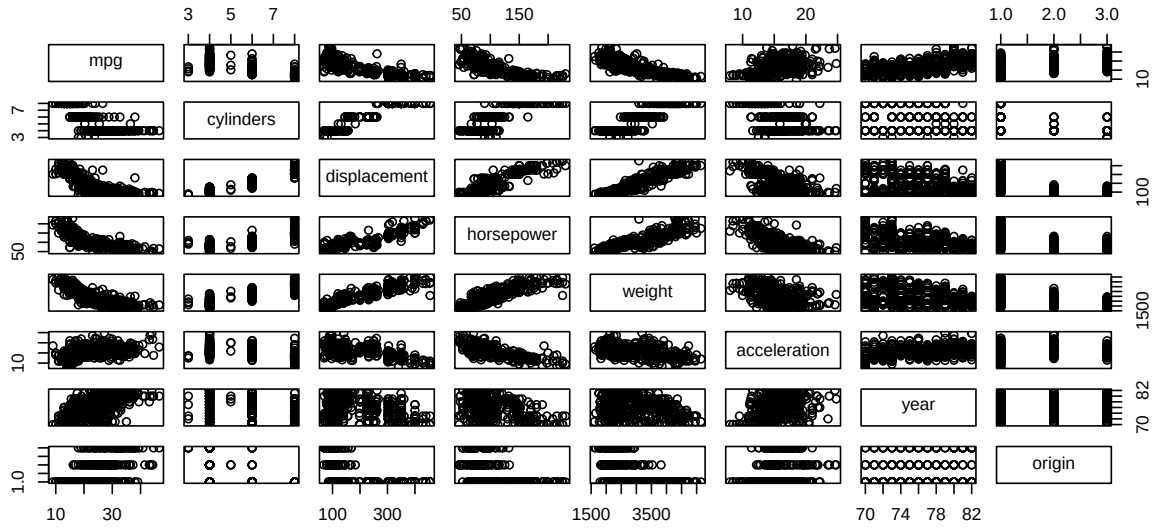
```
ggplot(Auto, aes(x=acceleration)) + geom_histogram(bins=20, fill="lightblue", color="black") +
  xlab("acceleration") + ylab("Frequency") + ggtitle("Histogram of acceleration")  +
theme(plot.title = element_text(hjust = 0.5))
```



The distribution satisfies normal distribution that most samples have acceleration ranging from 14 to 19.

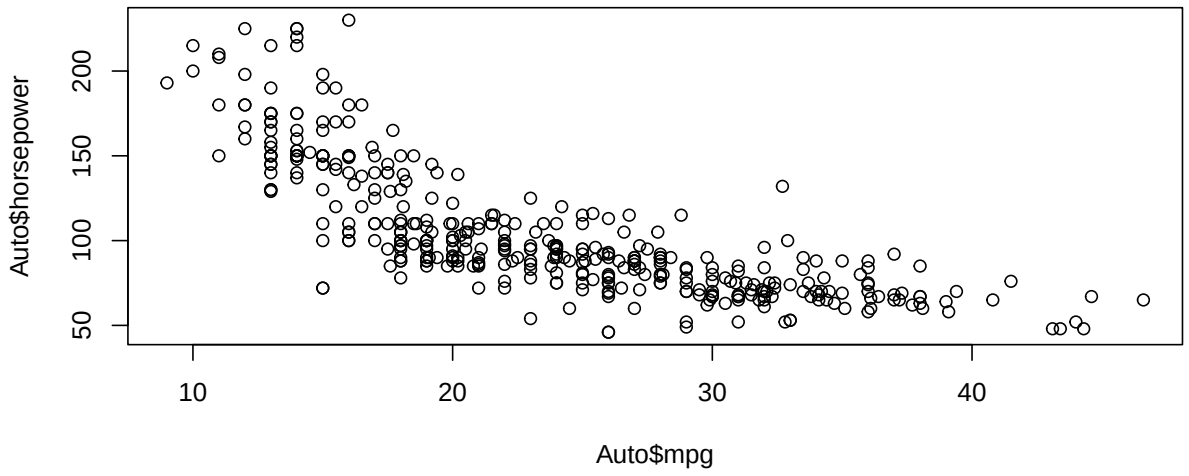
We then do pair analysis by removing the factor variable, i.e. name.

```
pairs(Auto[, -c(9)])
```



From the above scatter plot of all possible pairs, we could find that the plot of mpg pairing with cylinders displacement, horsepower, and weight have significant downward trend, i.e. the sample with a larger displacement, horsepower, and weight, will have smaller mpg. The plots of mpg pairing with acceleration, year, and origin are scattered but a rough upward trend. On the other hand, year has no obvious correlation with other columns. Displacement, horsepower, and weight have obvious correlation with each other.

```
plot(Auto$mpg, Auto$horsepower)
```



Above is the enlarged scatter plot showing mpg vs horsepower.

3.2 What effect does `time` have on `MPG`?

a) Start with a simple regression of `mpg` vs. `year` and report R's `summary` output. Is `year` a significant variable at the .05 level? State what effect `year` has on `mpg`, if any, according to this model.

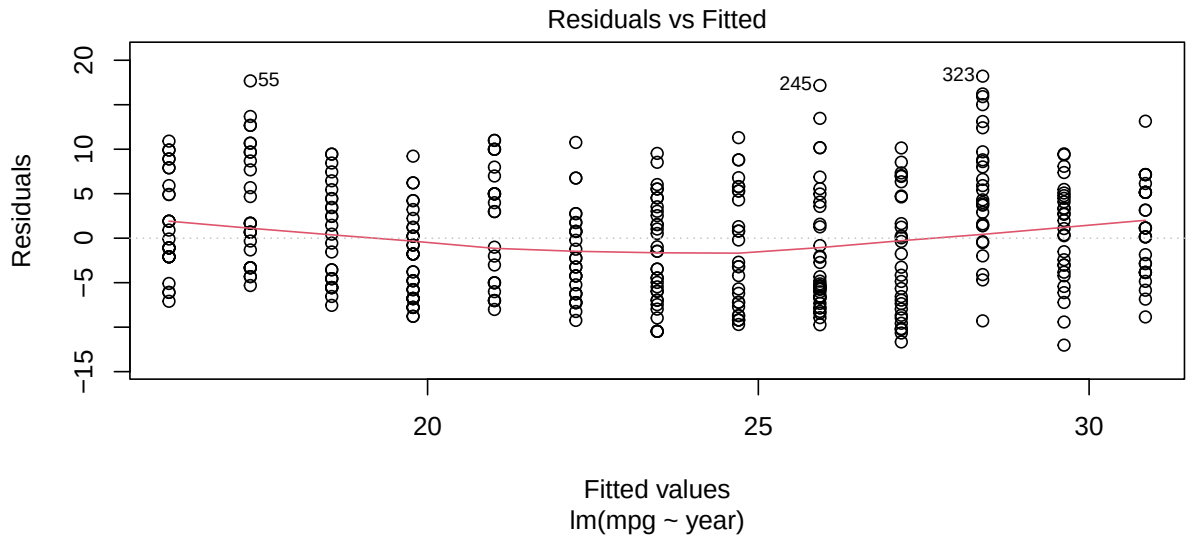
```
fit1 <- lm(mpg ~ year, data=Auto)
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ year, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.021  -5.441  -0.441   4.974  18.209
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70.0117     6.6452   -10.5   <2e-16 ***
## year          1.2300     0.0874    14.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.36 on 390 degrees of freedom
## Multiple R-squared:  0.337,  Adjusted R-squared:  0.335
## F-statistic:  198 on 1 and 390 DF,  p-value: <2e-16
```

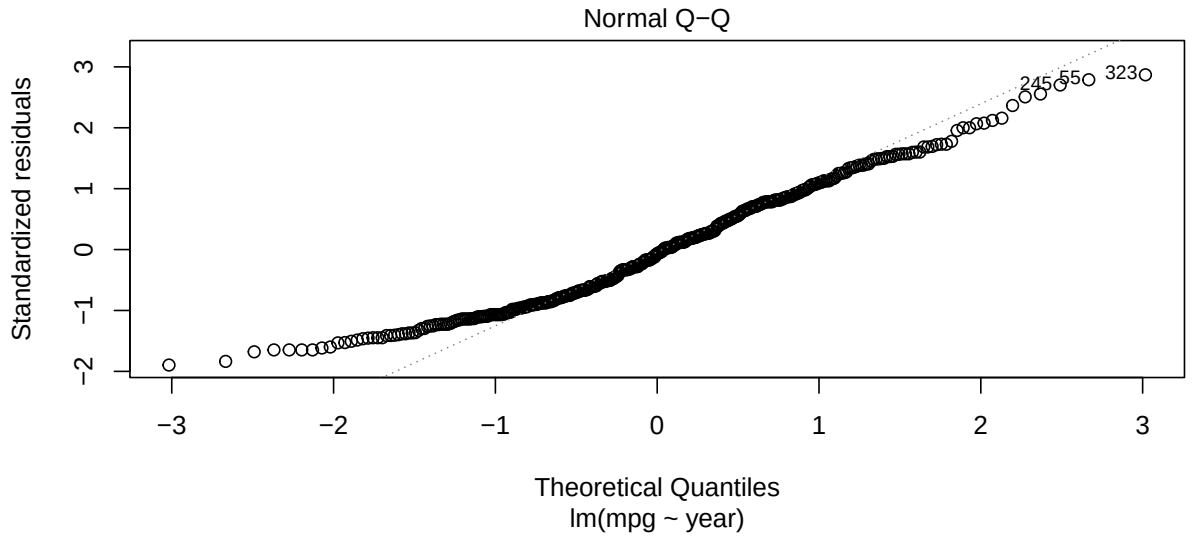
```
summary(fit1)$sigma
```

```
## [1] 6.36
```

```
plot(fit1, 1)
```




```
plot(fit1, 2)
```



Above is result from the regression of mpg and year. The p value of year is less than 2e-16, which is far smaller than 0.05. Thus, year is a significant variable at the 0.05 level. This model indicates that if the year increases by 1, the mpg will increase by 1.23. But the R square is only 0.337 that only explain little variance. From the qq plot, we could found that the lower tails deviates that it doesn't satisfy normal distribution.

b) Add `horsepower` on top of the variable `year` to your linear model. Is `year` still a significant variable at the .05 level? Give a precise interpretation of the `year`'s effect found here.

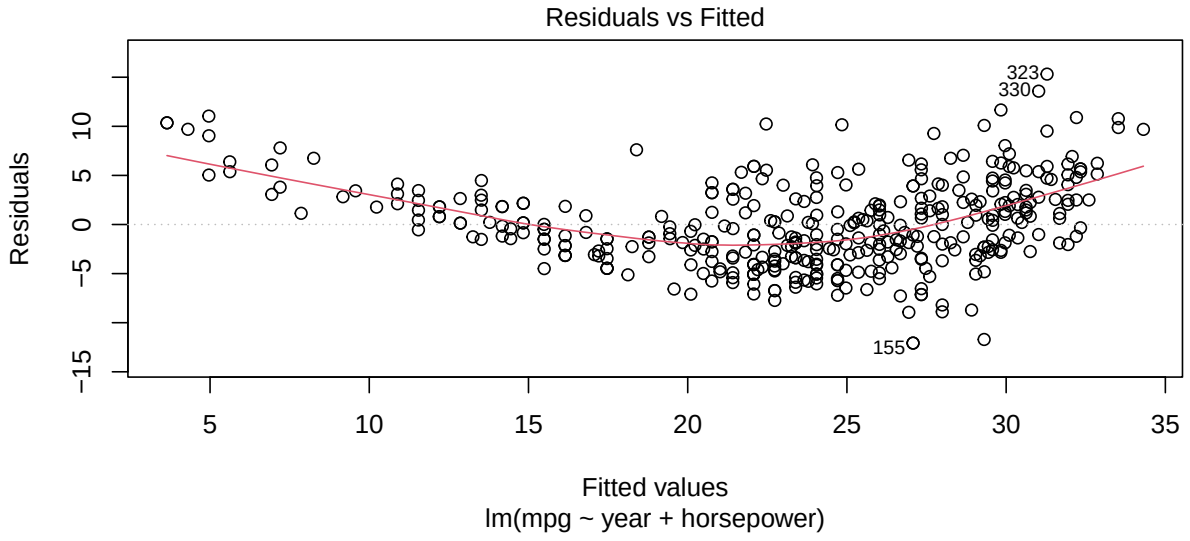
```
fit2 <- lm(mpg ~ year + horsepower, data = Auto)
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ year + horsepower, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.077  -3.078  -0.431   2.588  15.315
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.73917    5.34903   -2.38    0.018 *
## year          0.65727    0.06626    9.92   <2e-16 ***
## horsepower   -0.13165    0.00634  -20.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.39 on 389 degrees of freedom
## Multiple R-squared:  0.685,  Adjusted R-squared:  0.684
## F-statistic:  424 on 2 and 389 DF,  p-value: <2e-16
```

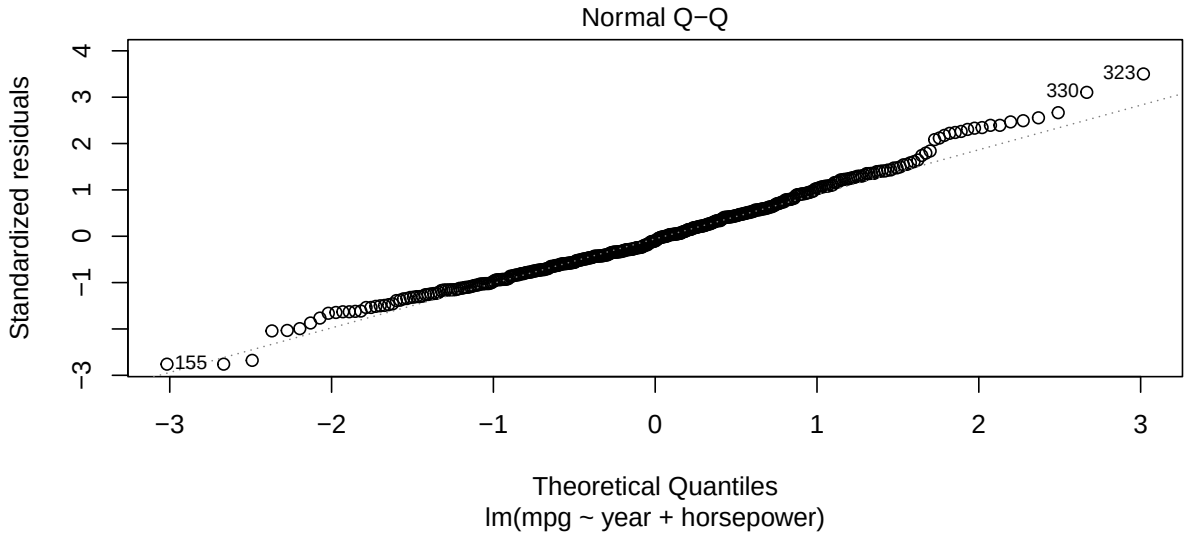
```
summary(fit2)$sigma
```

```
## [1] 4.39
```

```
plot(fit2, 1)
```



```
plot(fit2, 2)
```



Above is result from the regression of year and horsepower on mpg. The p value of year is still less than 2e-16, which is far smaller than 0.05. Thus, year is still a significant variable at the 0.05 level. In this model, if horsepower stays the same and year increases by 1, the mpg will increase by 0.65727, which is smaller than previous model. It is because the added horsepower interpret some variance of mpg and year weights less. The R square increases from 0.337 to 0.685, which is a huge improvement. The RSE decreases from 6.36 to 4.39. The qq plot shows that the residuals roughly satisfy normal distribution.

c) The two 95% CI's for the coefficient of year differ among (i) and (ii). How would you explain the difference to a non-statistician?

For model 1 among (i), we have [1.2300 - 1.96 * 0.0874, 1.2300 + 1.96 * 0.0874], which is [1.059, 1.4]. For model 2 among (ii), we have [0.65727 - 1.96 * 0.06626, 0.65727 + 1.96 * 0.06626], which is [0.5274, 0.787]. The addition of the variable "horsepower" in model 2 allows us to capture the effect of both year and horsepower on the dependent variable, whereas in model 1 we only capture the effect of year. This means that the coefficient of year in model 2 accounts for the effect of year on the dependent variable, controlling for the effect of horsepower. The difference in the two CIs reflects this difference in the models. Model 2 takes into account the effect of horsepower, which can affect the precision of the estimate of the coefficient of year. As a result, the CI for the coefficient of year in model 2 is narrower than the CI in model 1, which only accounts for the effect of year. This indicates that model 2 is a more precise estimate of the effect of year on the dependent variable, as it accounts for the effect of another important variable.

d) Create a model with interaction by fitting `lm(mpg ~ year * horsepower)`. Is the interaction effect significant at .05 level? Explain the year effect (if any).

```
fit3 <- lm(mpg ~ year * horsepower, data = Auto)
summary(fit3)
```

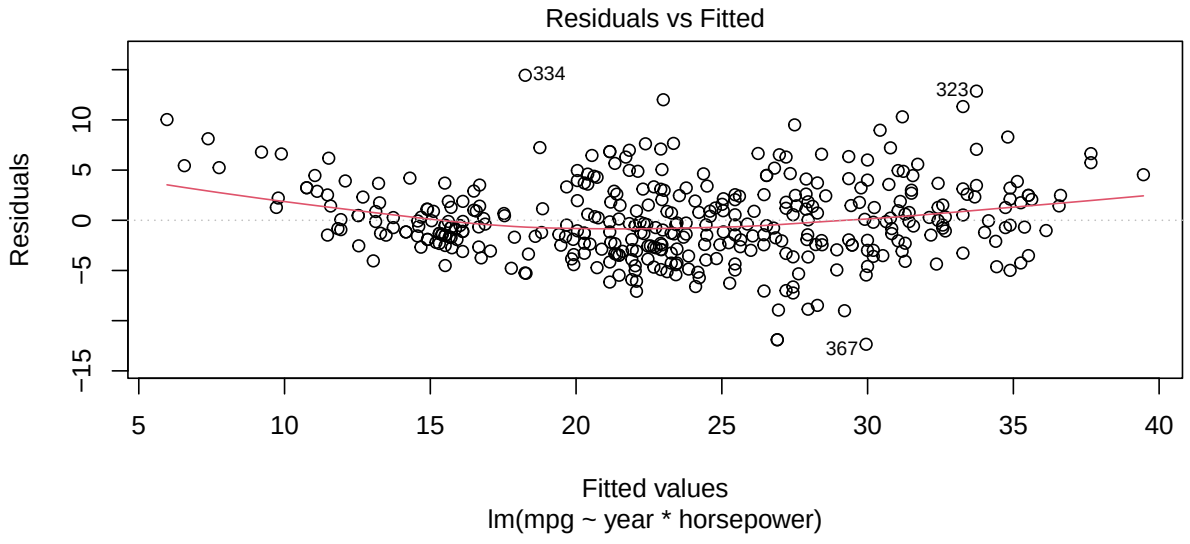
```
##
## Call:
```

```
## lm(formula = mpg ~ year * horsepower, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.349  -2.451  -0.456   2.406  14.444
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.27e+02   1.21e+01  -10.45   <2e-16 ***
## year             2.19e+00   1.61e-01   13.59   <2e-16 ***
## horsepower       1.05e+00   1.15e-01    9.06   <2e-16 ***
## year:horsepower -1.60e-02   1.56e-03  -10.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.9 on 388 degrees of freedom
## Multiple R-squared:  0.752,  Adjusted R-squared:  0.75
## F-statistic:  393 on 3 and 388 DF,  p-value: <2e-16
```

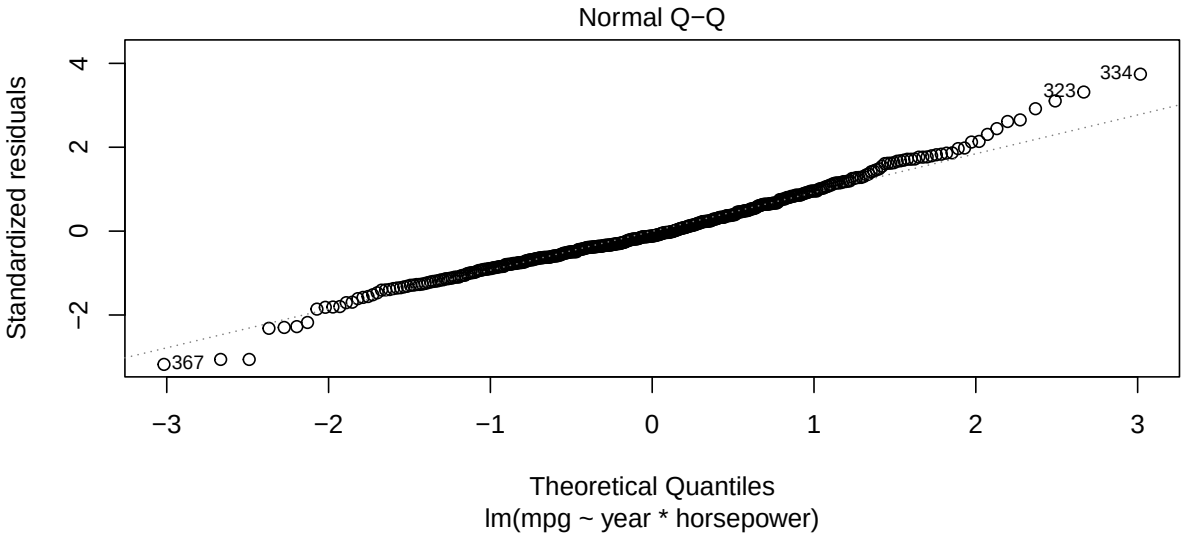
```
summary(fit3)$sigma
```

```
## [1] 3.9
```

```
plot(fit3, 1)
```




```
plot(fit3, 2)
```



Above is result from the regression of year and horsepower and the interaction of both variables on mpg. The p value of all three terms are less than 2e-16, which is far smaller than 0.05. Thus, interaction effect is a significant variable at the 0.05 level. In this model, if horsepower stays the same and year increases by 1, the mpg will increase by 2.19 - 1.6e-02 * horsepower. The relationship between horsepower and mpg is becoming weaker as the year of the car increases. The R square increases from 0.685 to 0.752, which is a huge improvement. The RSE decreases from 4.39 to 3.9. The qq plot shows that the residuals roughly satisfy normal distribution.

3.3 Categorical predictors

Remember that the same variable can play different roles! Take a quick look at the variable `cylinders`, and try to use this variable in the following analyses wisely. We all agree that a larger number of cylinders will lower mpg. However, we can interpret `cylinders` as either a continuous (numeric) variable or a categorical variable.

a) Fit a model that treats `cylinders` as a continuous/numeric variable. Is `cylinders` significant at the 0.01 level? What effect does `cylinders` play in this model?

```
fit_num <- lm(mpg ~ cylinders, data = Auto)
summary(fit_num)
```

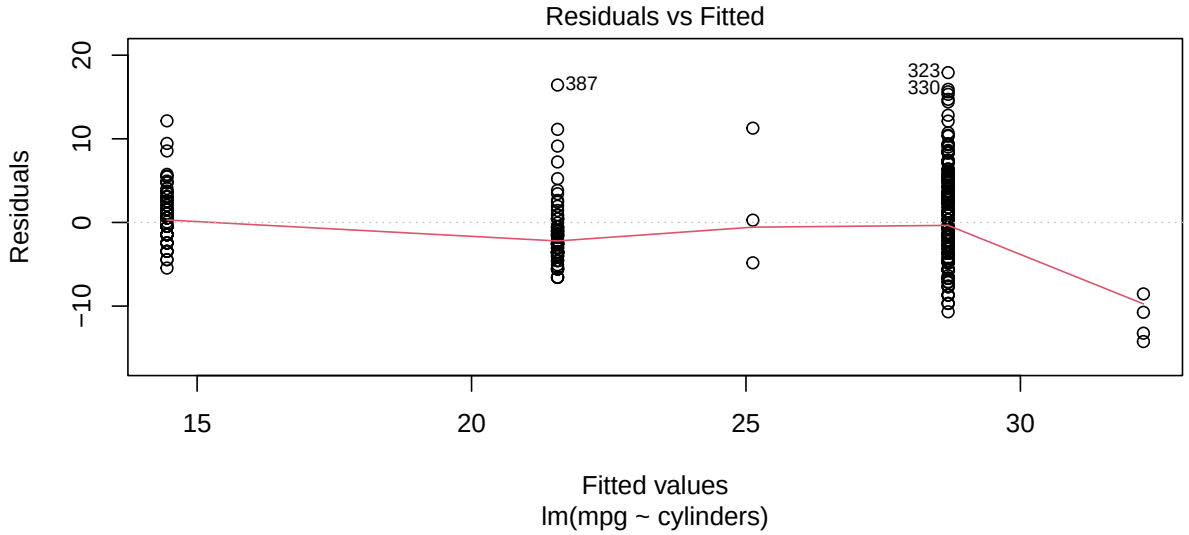
```
##
## Call:
## lm(formula = mpg ~ cylinders, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.241  -3.183  -0.633   2.549  17.917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.916      0.835    51.4   <2e-16 ***
```

```
## cylinders     -3.558      0.146   -24.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.91 on 390 degrees of freedom
## Multiple R-squared:  0.605,  Adjusted R-squared:  0.604
## F-statistic:  597 on 1 and 390 DF,  p-value: <2e-16
```

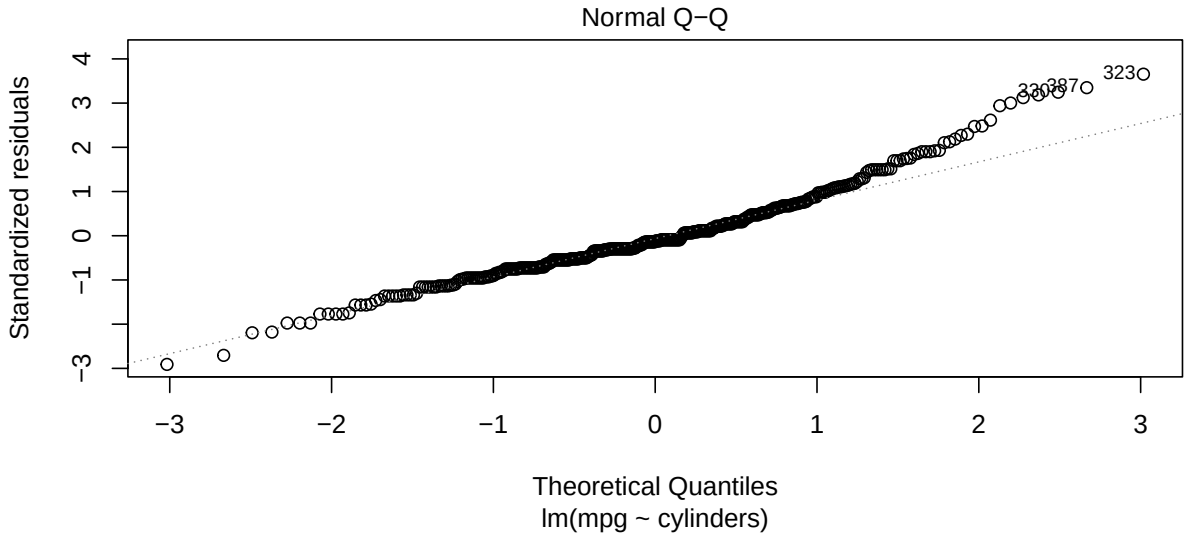
```
summary(fit_num)$sigma
```

```
## [1] 4.91
```

```
plot(fit_num, 1)
```




```
plot(fit_num, 2)
```



Above is result from the regression of numeric cylinders on mpg. The p value of cylinders is still less than 2e-16, which is far smaller than 0.05. Thus, cylinders is still a significant variable at the 0.05 level. In this model, if cylinders increases by 1, the mpg will decrease by 3.558. Adding numeric cylinder, the R square is 0.605. The RSE is 4.91. The qq plot shows that the residuals roughly satisfy normal distribution except for sample 387 and 323.

b) Fit a model that treats `cylinders` as a categorical/factor. Is `cylinders` significant at the .01 level? What is the effect of `cylinders` in this model? Describe the `cylinders` effect over `mpg`.

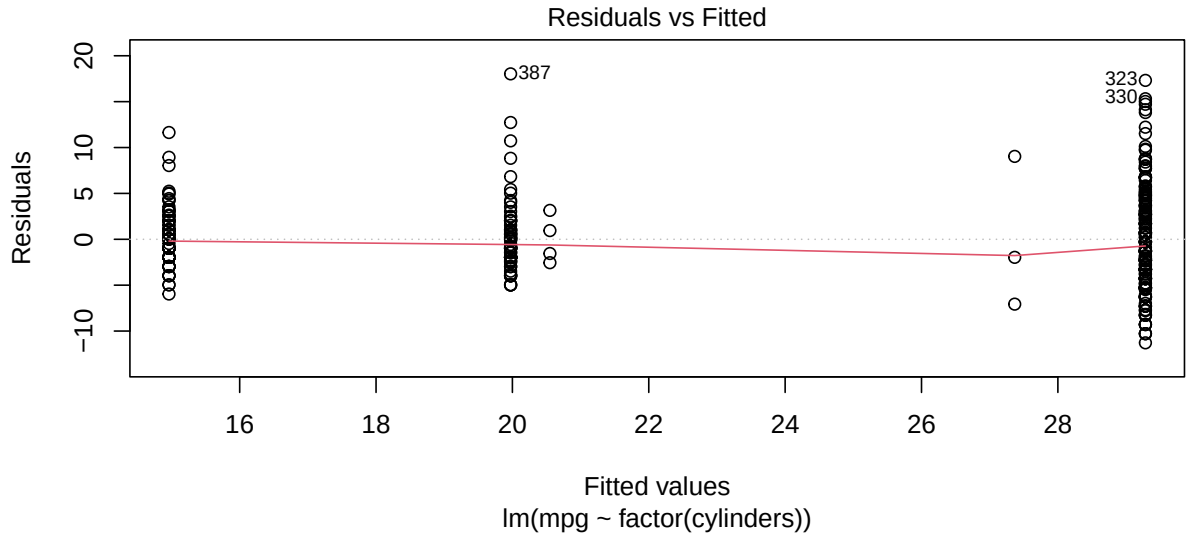
```
fit_factor <- lm(mpg ~ factor(cylinders), data = Auto)
summary(fit_factor)
```

```
##
## Call:
## lm(formula = mpg ~ factor(cylinders), data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.284  -2.904  -0.963   2.344  18.027
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          20.550      2.349    8.75  < 2e-16 ***
## factor(cylinders)4    8.734      2.373    3.68  0.00027 ***
## factor(cylinders)5    6.817      3.589    1.90  0.05825 .
## factor(cylinders)6   -0.577      2.405   -0.24  0.81071
## factor(cylinders)8   -5.587      2.395   -2.33  0.02015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.7 on 387 degrees of freedom
## Multiple R-squared:  0.641,  Adjusted R-squared:  0.638
## F-statistic:  173 on 4 and 387 DF,  p-value: <2e-16
```

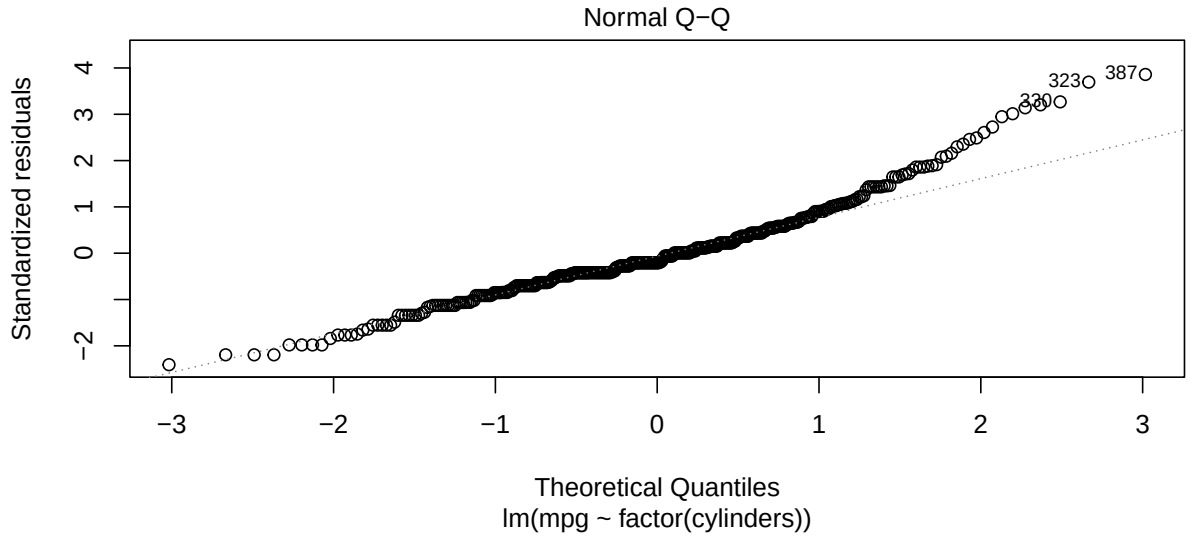
```
summary(fit_factor)$sigma
```

```
## [1] 4.7
```

```
plot(fit_factor, 1)
```




```
plot(fit_factor, 2)
```



Above is result from the regression of factor cylinders on mpg. The p value of cylinder 3 (Intercept) is smaller than 2e-16 and cylinder 4 is 0.00027, which is smaller than 0.01. But the rest of cylinders have p value larger than 0.01. Thus, cylinders 3 and 4 is a significant variable at the 0.01 level while other cylinder

are insignificant. In this model, if it is cylinder 3, then the mpg is the value of Intercept. If it is cylinder 4, then the mpg will increase from Intercept by 8.734, i.e. 8.734 mpg greater than cylinder 3. If it is cylinder 5, then the mpg will increase from Intercept by 6.817, i.e. 6.817 mpg greater than cylinder 3. If it is cylinder 6, then the mpg will decrease from Intercept by 0.577, i.e. 0.577 mpg less than cylinder 3. If it is cylinder 8, then the mpg will decrease from Intercept by 5.587, i.e. 5.587 mpg less than cylinder 3. The model has R square 0.641 and RSE 4.7.

c) What are the fundamental differences between treating `cylinders` as a continuous and categorical variable in your models?

When treating cylinders as a continuous variable, the model assumes that there is a smooth, numerical relationship between the cylinders variable and the target variable, and the goal is to fit a line or curve that best captures this relationship. This approach is usually appropriate when the variable has a natural ordering and there is a clear magnitude difference between the different values.

When treating cylinders as a categorical variable, the model instead creates separate categories for each distinct value of the variable, and the goal is to model the relationship between each category and the target variable. This approach is usually appropriate when the variable does not have a natural ordering, or there is no clear magnitude difference between the values.

d) Can you test the null hypothesis: fit0: `mpg` is linear in `cylinders` vs. fit1: `mpg` relates to `cylinders` as a categorical variable at .01 level?

```
anova(fit_num, fit_factor)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ cylinders
## Model 2: mpg ~ factor(cylinders)
##   Res.Df  RSS Df Sum of Sq    F  Pr(>F)
## 1    390 9416
## 2    387 8544  3       871 13.2 3.4e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the ANOVA table shows that the p-value is less than 0.01, indicating the null hypothesis should be rejected, i.e. Model 2 is a better fit than Model 1. The factor(cylinders) term in Model 2 represents the relationship between mpg and cylinders as a categorical variable, while the cylinders term in Model 1 represents a linear relationship between mpg and cylinders. Thus, this result suggests that the relationship between mpg and cylinders is better represented as a categorical variable than as a linear variable.

3.4 Results

Final modeling question: we want to explore the effects of each feature as best as possible. You may explore interactions, feature transformations, higher order terms, or other strategies within reason. The model(s) should be as parsimonious (simple) as possible unless the gain in accuracy is significant from your point of view.

a) Describe the final model. Include diagnostic plots with particular focus on the model residuals and diagnoses.

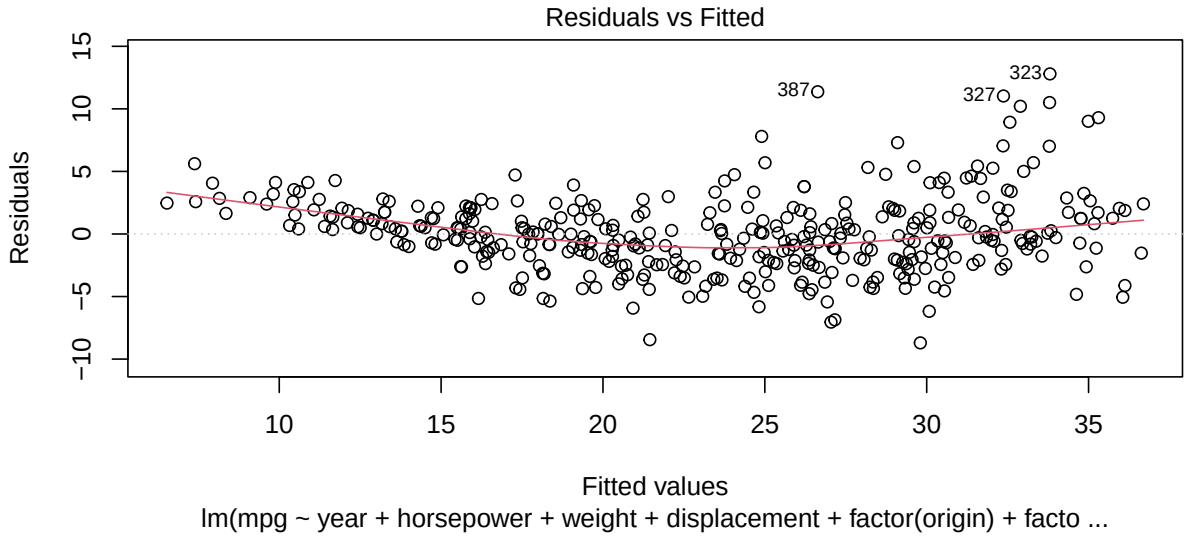
```r
fit_final <- lm(mpg ~ year + horsepower + weight + displacement + factor(origin) + factor(cylinders), da
summary(fit_final)
```

```
##
## Call:
## lm(formula = mpg ~ year + horsepower + weight + displacement +
##     factor(origin) + factor(cylinders), data = Auto)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.704 -1.950 -0.055  1.711 12.793
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.16e+01   4.23e+00   -5.11  5.1e-07 ***
## year                7.36e-01   4.87e-02   15.11  < 2e-16 ***
## horsepower         -3.71e-02   1.07e-02   -3.46   0.0006 ***
## weight             -5.70e-03   5.53e-04  -10.29  < 2e-16 ***
## displacement        1.85e-02   7.17e-03    2.58   0.0103 *
## factor(origin)2     1.76e+00   5.51e-01    3.20   0.0015 **
## factor(origin)3     2.62e+00   5.26e-01    4.98  9.7e-07 ***
## factor(cylinders)4  6.78e+00   1.64e+00    4.14  4.2e-05 ***
## factor(cylinders)5  7.15e+00   2.50e+00    2.86   0.0045 **
## factor(cylinders)6  3.40e+00   1.81e+00    1.88   0.0613 .
## factor(cylinders)8  5.14e+00   2.10e+00    2.44   0.0150 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.09 on 381 degrees of freedom
## Multiple R-squared:  0.847,  Adjusted R-squared:  0.843
## F-statistic:  211 on 10 and 381 DF,  p-value: <2e-16
```

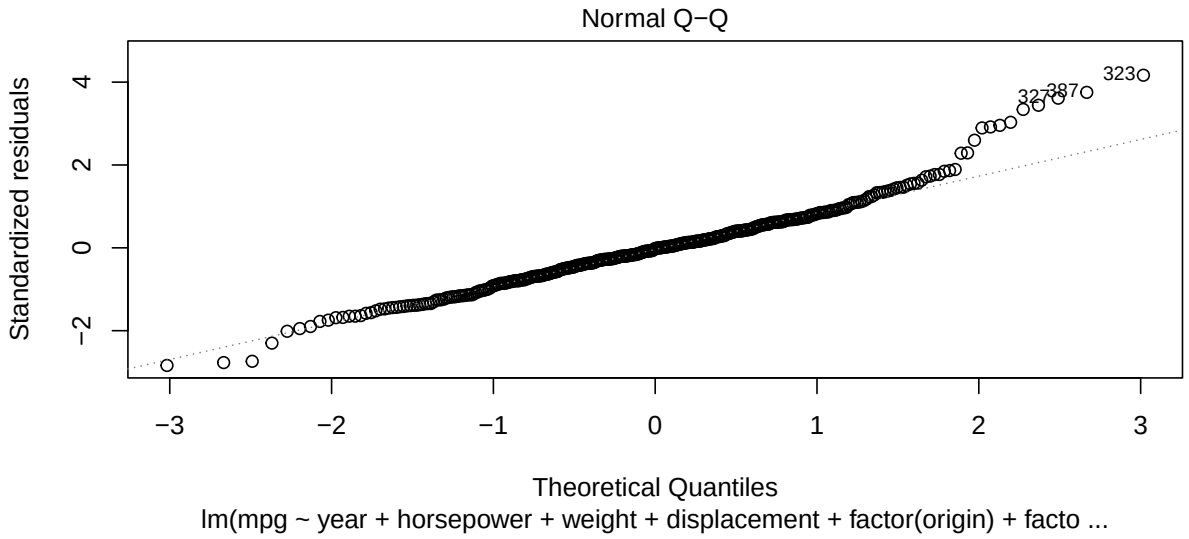
```r
summary(fit_final)$sigma
```

```
## [1] 3.09
```

```r
plot(fit_final, 1)
```




```
plot(fit_final, 2)
```



Above is our final model including the interaction of year and horsepower, weights, and factor variables origin and cylinders. Most of the variables are significant at the level of 0.001 while some factor variables are significant at the level of 0.01. The final R square reaches 0.868, which is a huge improvement from the beginning and this model captures 86.8% variance of the data.

From the plot residuals vs fitted, we found that the red line captures the main trend while there are some outliers such as 334, 387, and 323.

The qq plot shows that the residuals mainly satisfy the normal distribution while the upper tail deviates a little bit with outliers 334, 387, and 323.

b) Summarize the effects found.

The final model finds the following effects on mpg: if other variables remain the same, with year increasing by 1, the mpg will increase by 1.70. If horsepower increases by 1, the mpg will decrease by 0.707. With weight increasing by 1, the mpg will decrease by 0.00485. If origin is 2 instead of 1, the mpg will increase by 1.58. If origin is 3 instead of 1, the mpg will increase by 2.09. With cylinders being 4 instead of 3, the mpg will increase by 6.43. With cylinders being 5 instead of 3, the mpg will increase by 6.55. With cylinders being 6 instead of 3, the mpg will increase by 4.96. With cylinders being 8 instead of 3, the mpg will increase by 7.56. The model also includes an interaction term between year and horsepower, which captures the effect of the combined influence of these two variables on mpg. The interaction term analysis shows that, if other variables remain the same, as year and horsepower both increase by 1 unit, the mpg will decrease by 0.0101. This indicates that the effect of an increase in year and horsepower on mpg is not simply the sum of their individual effects, but rather a combined effect that is different from the effects of each variable alone.

c) Predict the `mpg` of the following car: A red car built in the US in 1983 that is 180 inches long, has eight cylinders, displaces 350 cu. inches, weighs 4000 pounds, and has a horsepower of 260. Also give a 95% CI for your prediction.

```
new_car <- data.frame(year = 83,
                      horsepower = 260,
                      weight = 4000,
                      origin = 1,
                      cylinders = 8,
                      displacement = 350)

prediction <- predict(fit_final, newdata = new_car, interval = "confidence", level = 0.95)
prediction
```

```
##    fit  lwr  upr
## 1 18.6 16.1 21.2
```

4 Simple Regression through simulations

4.1 Linear model through simulations

This exercise is designed to help you understand the linear model using simulations. In this exercise, we will generate (x_i, y_i) pairs so that all linear model assumptions are met.

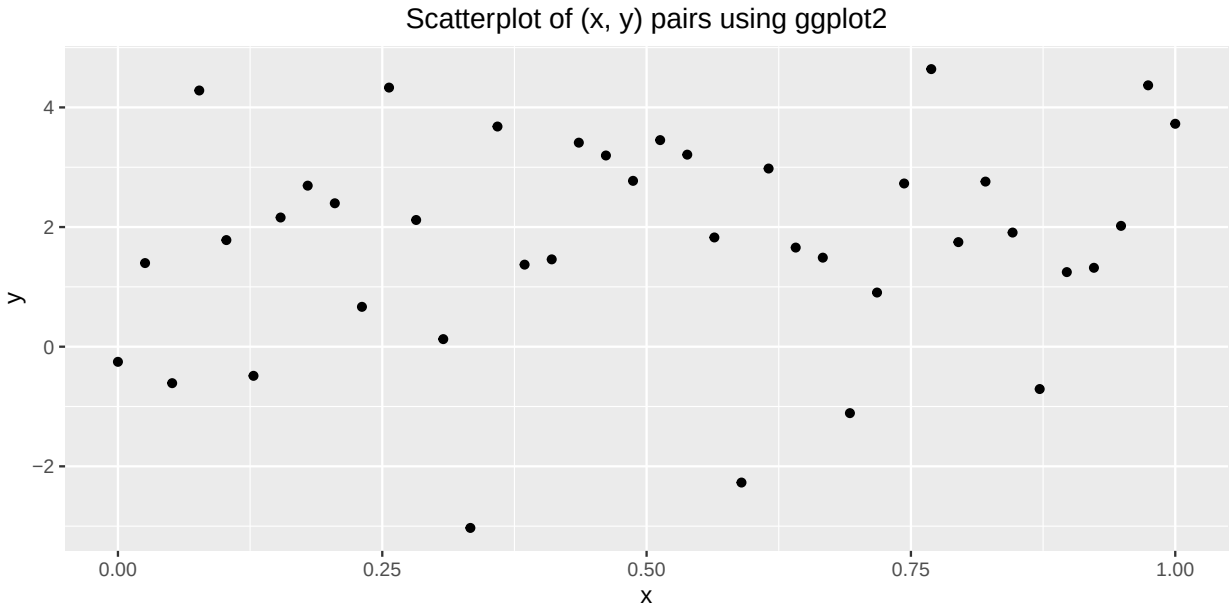
Presume that $\mathbf{x}$ and $\mathbf{y}$ are linearly related with a normal error $\boldsymbol{\varepsilon}$, such that $\mathbf{y} = 1 + 1.2\mathbf{x} + \boldsymbol{\varepsilon}$. The standard deviation of the error ε_i is $\sigma = 2$.

We can create a sample input vector $(n = 40)$ for $\mathbf{x}$ with the following code:

```
# Generates a vector of size 40 with equally spaced values between 0 and 1, inclusive
x <- seq(0, 1, length = 40)
```

4.1.1 Generate data

Create a corresponding output vector for $\mathbf{y}$ according to the equation given above. Use `set.seed(1)`. Then, create a scatterplot with (x_i, y_i) pairs. Base R plotting is acceptable, but if you can, please attempt to use `ggplot2` to create the plot. Make sure to have clear labels and sensible titles on your plots.



4.1.2 Understand the model

i. Find the LS estimates of β_0 and β_1, using the `lm()` function. What are the true values of β_0 and β_1? Do the estimates look to be good?

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.662 -0.880  0.014  1.247  2.882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.331      0.557    2.39    0.022 *
## x              0.906      0.959    0.95    0.350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.79 on 38 degrees of freedom
## Multiple R-squared:  0.023,  Adjusted R-squared:  -0.00272
## F-statistic: 0.894 on 1 and 38 DF,  p-value: 0.35
```

The true value of β_0 is 1 and β_1 is 1.2 while the estimate of β_0 is 1.331 and β_1 is 0.906. The estimate doesn't look good.

ii. What is your RSE for this linear model fit? Is it close to $\sigma = 2$?

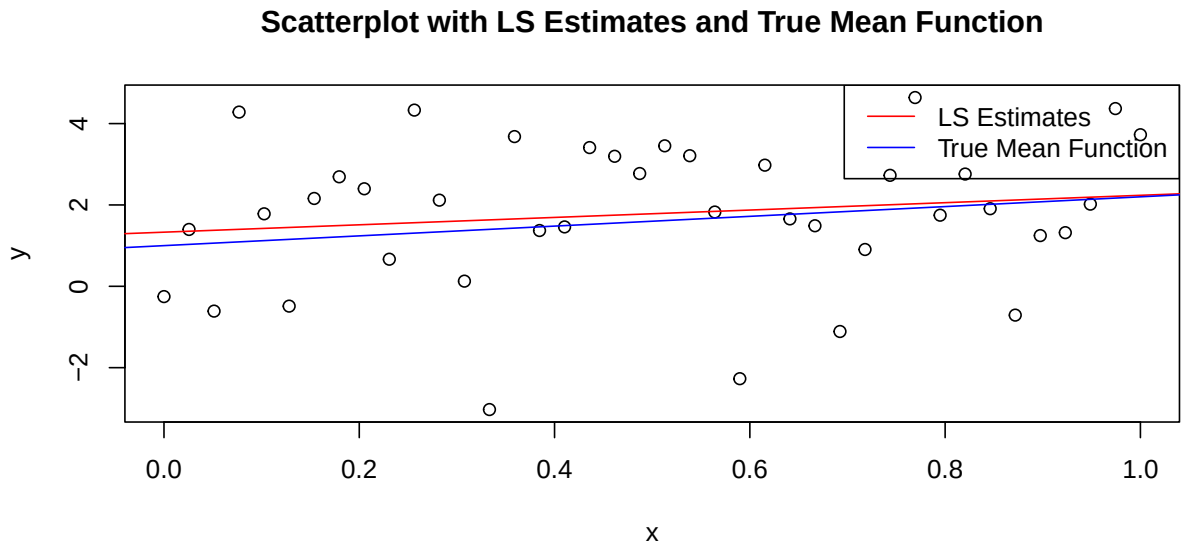
The RSE is approximately 1.79, which is close to $\sigma = 2$

iii. What is the 95% confidence interval for β_1? Does this confidence interval capture the true β_1?

```
##              2.5 % 97.5 %
## (Intercept)  0.203   2.46
## x           -1.034   2.85
```

The 95% confidence interval for β_1 is [-1.034,2.85], which captures the true β_1 1.2.

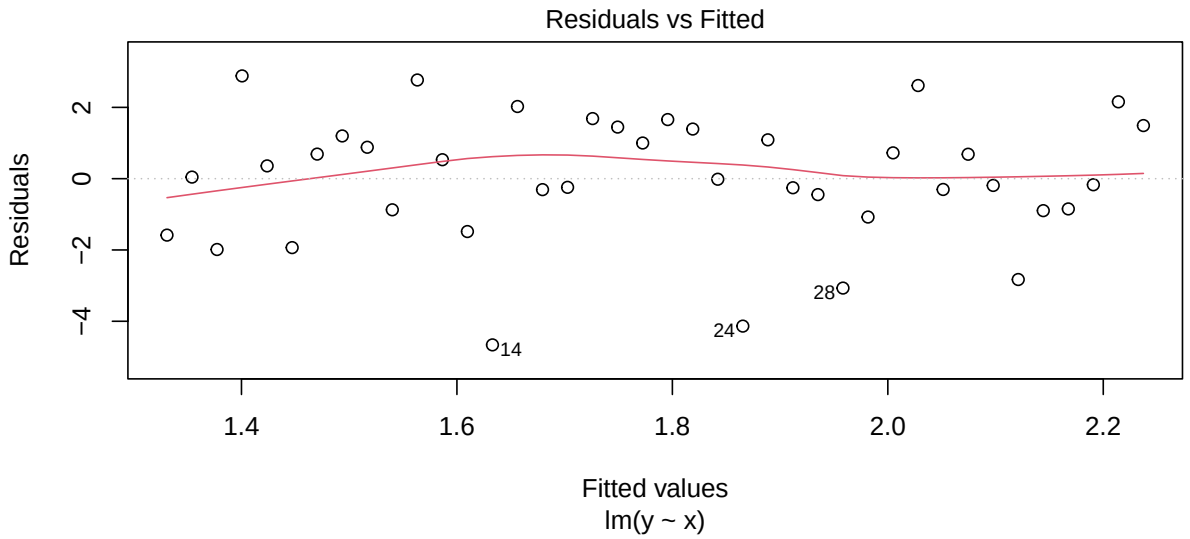
iv. Overlay the LS estimates and the true lines of the mean function onto a copy of the scatterplot you made above.



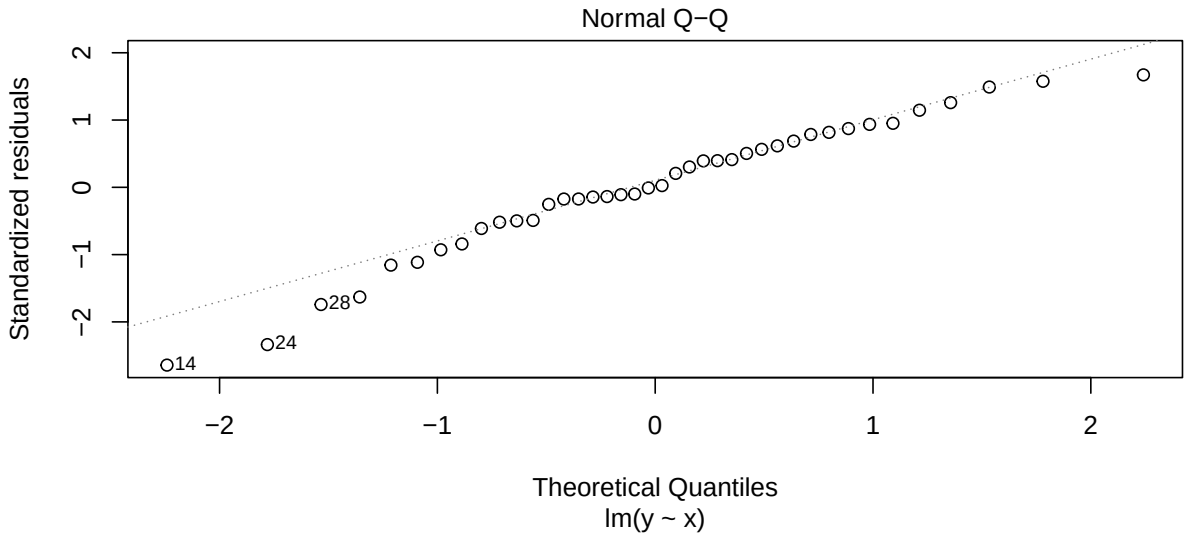
The blue line is the true line of the mean function and the red line is the LS estimate of the mean function. We could conclude that they are similar to each other.

4.1.3 diagnoses

i. Provide residual plot where fitted **y**-values are on the x-axis and residuals are on the y-axis.



ii. Provide a normal QQ plot of the residuals.



iii. Comment on how well the model assumptions are met for the sample you used.

Based on the residual plot and the QQ plot, we can see that the residuals are randomly scattered around 0, and the points form a straight line in the QQ plot except for some outliers such as 28, 24, 14. This indicates that the linear model assumptions of constant variance and normality of residuals are met for the sample used.

4.2 Understand sampling distribution and confidence intervals

This part aims to help you understand the notion of sampling statistics and confidence intervals. Let's concentrate on estimating the slope only.

Generate 100 samples of size $n = 40$, and estimate the slope coefficient from each sample. We include some sample code below, which should guide you in setting up the simulation. Note: this code is easier to follow but suboptimal; see the appendix for a more optimal R-like way to run this simulation.

```
# Inializing variables. Note b_1, upper_ci, lower_ci are vectors
x <- seq(0, 1, length = 40)
n_sim <- 100              # number of simulations
b1 <- 0                   # n_sim many LS estimates of beta_1 (=1.2). Initialize to 0 for now
upper_ci <- 0             # upper bound for beta_1. Initialize to 0 for now.
lower_ci <- 0             # lower bound for beta_1. Initialize to 0 for now.
t_star <- qt(0.975, 38)   # Food for thought: why 38 instead of 40? What is t_star?

# Perform the simulation
for (i in 1:n_sim){
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
  lse <- lm(y ~ x)
  lse_output <- summary(lse)$coefficients
  se <- lse_output[2, 2]
  b1[i] <- lse_output[2, 1]
  upper_ci[i] <- b1[i] + t_star * se
  lower_ci[i] <- b1[i] - t_star * se
}
results <- as.data.frame(cbind(se, b1, upper_ci, lower_ci))
results
summary(results)
# remove unecessary variables from our workspace
# rm(se, b1, upper_ci, lower_ci, x, n_sim, b1, t_star, lse, lse_out)
```

i. Summarize the LS estimates of β_1 (stored in `results$b1`). Does the sampling distribution agree with theory?

After summarizing the estimates of β_1 and its confidence interval, more than 75% lower bound is smaller than 1.2 and more than 75% upper bound is larger than 1.2. Thus, we could conclude that the sampling distribution agrees with theory.

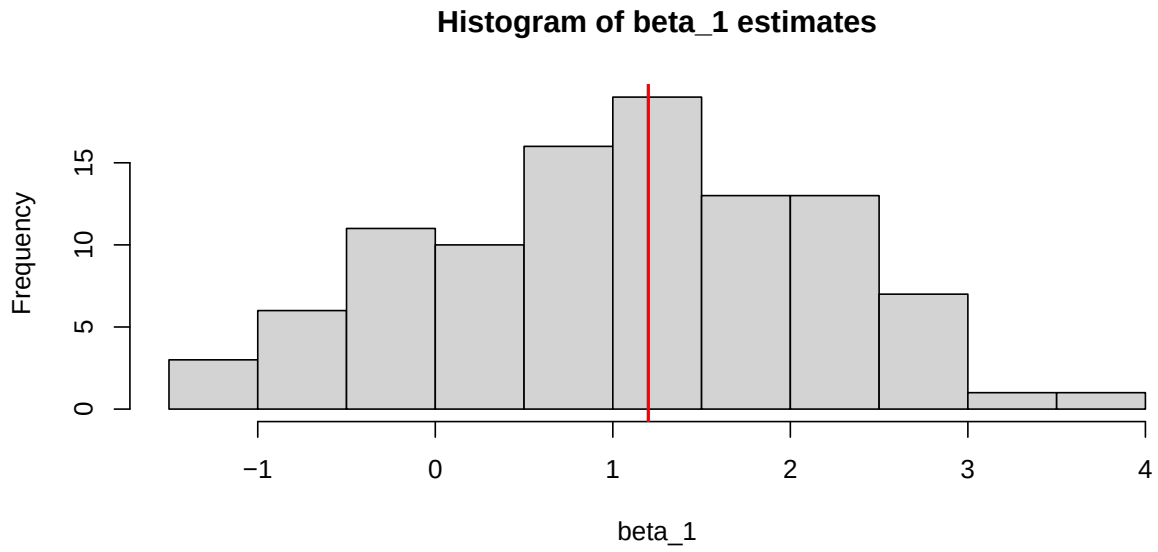
ii. How many of your 95% confidence intervals capture the true β_1? Display your confidence intervals graphically.

```
x <- seq(0, 1, length = 40)
n_sim <- 100              # number of simulations
b1 <- 0                   # n_sim many LS estimates of beta_1 (=1.2). Initialize to 0 for now
upper_ci <- 0             # upper bound for beta_1. Initialize to 0 for now.
lower_ci <- 0             # lower bound for beta_1. Initialize to 0 for now.
t_star <- qt(0.975, 38)   # Food for thought: why 38 instead of 40? What is t_star?

# Perform the simulation
for (i in 1:n_sim){
  y <- 1 + 1.2 * x + rnorm(40, sd = 2)
```

```
  lse <- lm(y ~ x)
  lse_output <- summary(lse)$coefficients
  se <- lse_output[2, 2]
  b1[i] <- lse_output[2, 1]
  upper_ci[i] <- b1[i] + t_star * se
  lower_ci[i] <- b1[i] - t_star * se
}
results <- as.data.frame(cbind(se, b1, upper_ci, lower_ci))
summary(results)
```

```
##        se              b1           upper_ci       lower_ci
##  Min.   :1.16   Min.   :-1.36   Min.   :0.66   Min.   :-3.73
##  1st Qu.:1.16   1st Qu.: 0.23   1st Qu.:2.56   1st Qu.:-2.08
##  Median :1.16   Median : 1.05   Median :3.27   Median :-1.12
##  Mean   :1.16   Mean   : 1.06   Mean   :3.29   Mean   :-1.18
##  3rd Qu.:1.16   3rd Qu.: 1.87   3rd Qu.:4.15   3rd Qu.:-0.39
##  Max.   :1.16   Max.   : 3.97   Max.   :5.98   Max.   : 1.95
```

```
hist(results$b1, main = "Histogram of beta_1 estimates", xlab = "beta_1")
abline(v = 1.2, col = "red", lwd = 2) # add vertical line at true beta_1
```




```
# calculate the proportion of confidence intervals that capture the true beta_1
capture_proportion <- mean(results$upper_ci >= 1.2 & results$lower_ci <= 1.2)
capture_proportion
```

```
## [1] 0.96
```

There are 93% of the 100 simulation captures the actual β_1. Above is the histogram of predicted β_1 that most of the estimated β_1 is around true value.