Ted Zybin

2022-12-1

External Requirements: Data Read-In and Package Loading

```
df = read.csv("Insurance_policy.csv")
```

```
library(mosaic)
library(ggplot2)
library(dplyr)
library(knitr)
```

Teammates' Names: Rohit Nair, Ibrahim Alwishah

Research Question

How do the provided variables associate with the prediction that a customer is high value?

```
insurancedata = df

# There are 48842 rows and 8 columns.
dim(insurancedata)
```

```
## [1] 48842     8
```

```
# The summary function has shown the min, max, median,
#and mean of the variables but there does not seem to be any missing values.
summary(insurancedata)
```

```
##       age        education_num   marital_status      occupation       
##  Min.   :17.00   Min.   : 1.00   Length:48842       Length:48842      
##  1st Qu.:28.00   1st Qu.: 9.00   Class :character   Class :character  
##  Median :37.00   Median :10.00   Mode  :character   Mode  :character  
##  Mean   :38.64   Mean   :10.08                                        
##  3rd Qu.:48.00   3rd Qu.:12.00                                        
##  Max.   :90.00   Max.   :16.00                                        
##     cap_gain     hours_per_week      score        value_flag       
##  Min.   :    0   Min.   : 1.00   Min.   :43.94   Length:48842      
##  1st Qu.:    0   1st Qu.:40.00   1st Qu.:57.50   Class :character  
##  Median :    0   Median :40.00   Median :60.24   Mode  :character  
##  Mean   : 1079   Mean   :40.42   Mean   :60.23                     
##  3rd Qu.:    0   3rd Qu.:45.00   3rd Qu.:62.95                     
##  Max.   :99999   Max.   :99.00   Max.   :76.53                     
```

```
# This has confirmed that there are no missing variables.
colSums( is.na(insurancedata))
```

```
##            age  education_num marital_status     occupation       cap_gain 
##              0              0              0              0              0 
## hours_per_week          score     value_flag 
##              0              0              0 
```

```
# how many rows have 0 NAs, in other words, how many rows are complete?
sum( complete.cases(insurancedata) )
```

```
## [1] 48842
```

```
# There are no missing values
sum( !complete.cases(insurancedata))
```

```
## [1] 0
```

There are no missing values in the table.

Exploratory Data Analysis: Descriptive Statistics and Visualizations

1)

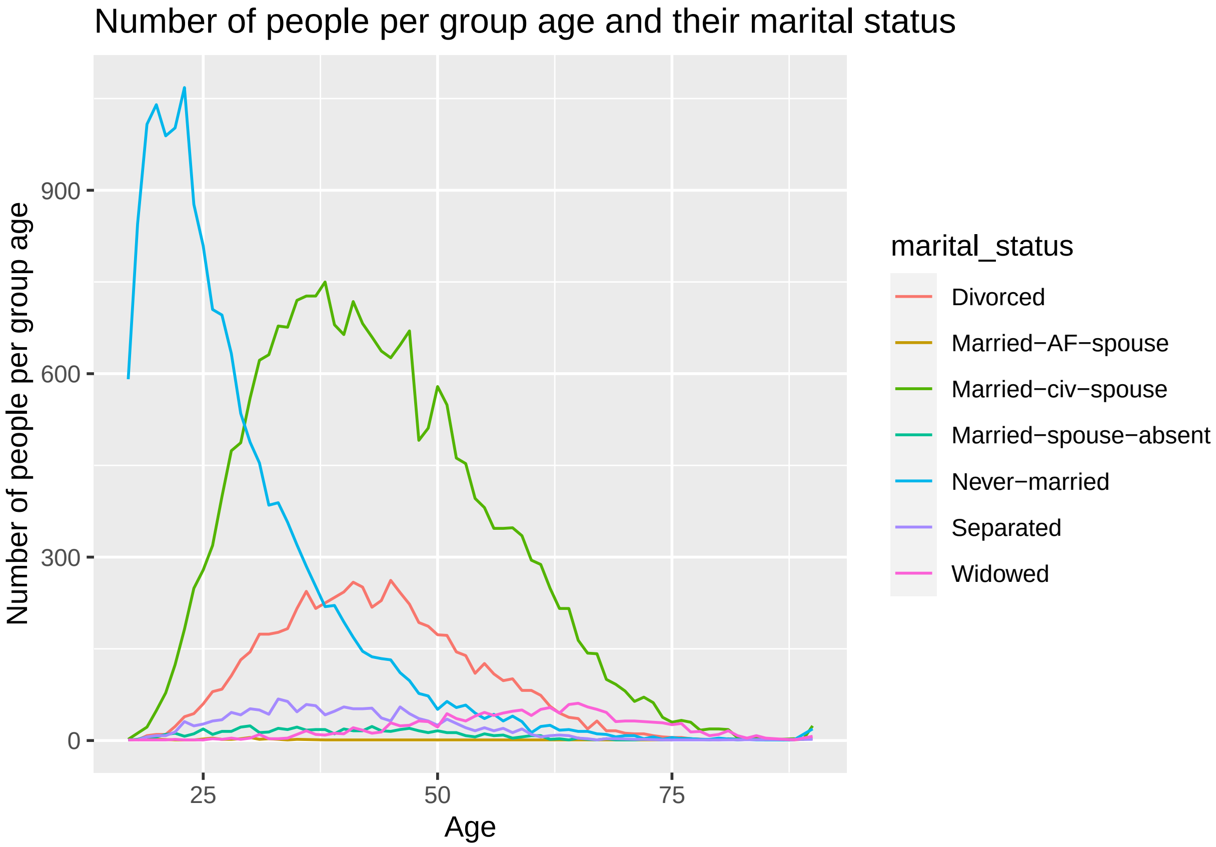
```
insurance = df
insuranceNum <- insurance %>%
  select(age, education_num, cap_gain, hours_per_week, score)


favstats_vec = c()
columns = colnames(insuranceNum)

total_favstats = data.frame()

for (i in 1:ncol(insuranceNum)){
  total_favstats <- rbind(total_favstats, favstats(insuranceNum[,i]))

}

total_favstats <- cbind(names = columns, total_favstats)
rownames(total_favstats) <- NULL

total_favstats
```

```
##            names   min   Q1 median    Q3      max       mean          sd     n
## 1            age 17.00 28.0  37.00 48.00    90.00   38.64359   13.710510 48842
## 2  education_num  1.00  9.0  10.00 12.00    16.00   10.07809    2.570973 48842
## 3       cap_gain  0.00  0.0   0.00  0.00 99999.00 1079.06763 7452.019058 48842
## 4 hours_per_week  1.00 40.0  40.00 45.00    99.00   40.42238   12.391444 48842
## 5          score 43.94 57.5  60.24 62.95    76.53   60.22825    4.025339 48842
##   missing
## 1       0
## 2       0
## 3       0
## 4       0
## 5       0
```

2)

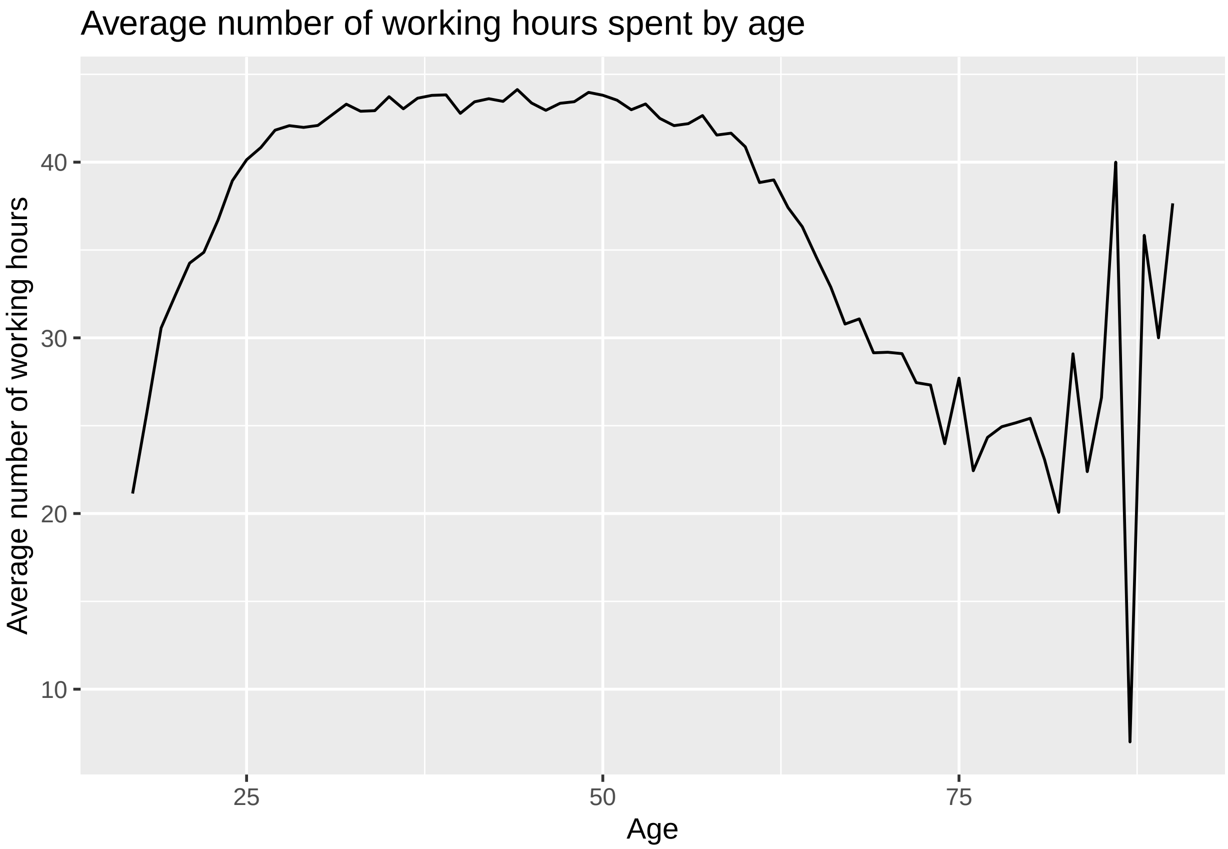
```
insurancedata %>%
  group_by(age, marital_status) %>%
  summarise(number_of_people = n()) %>%
  # creating a graph with respect to marital status
  ggplot() +
  geom_line(aes(x = age, y = number_of_people, color = marital_status)) +
  xlab("Age") + ylab("Number of people per group age") +
  ggtitle("Number of people per group age and their marital status")
```

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



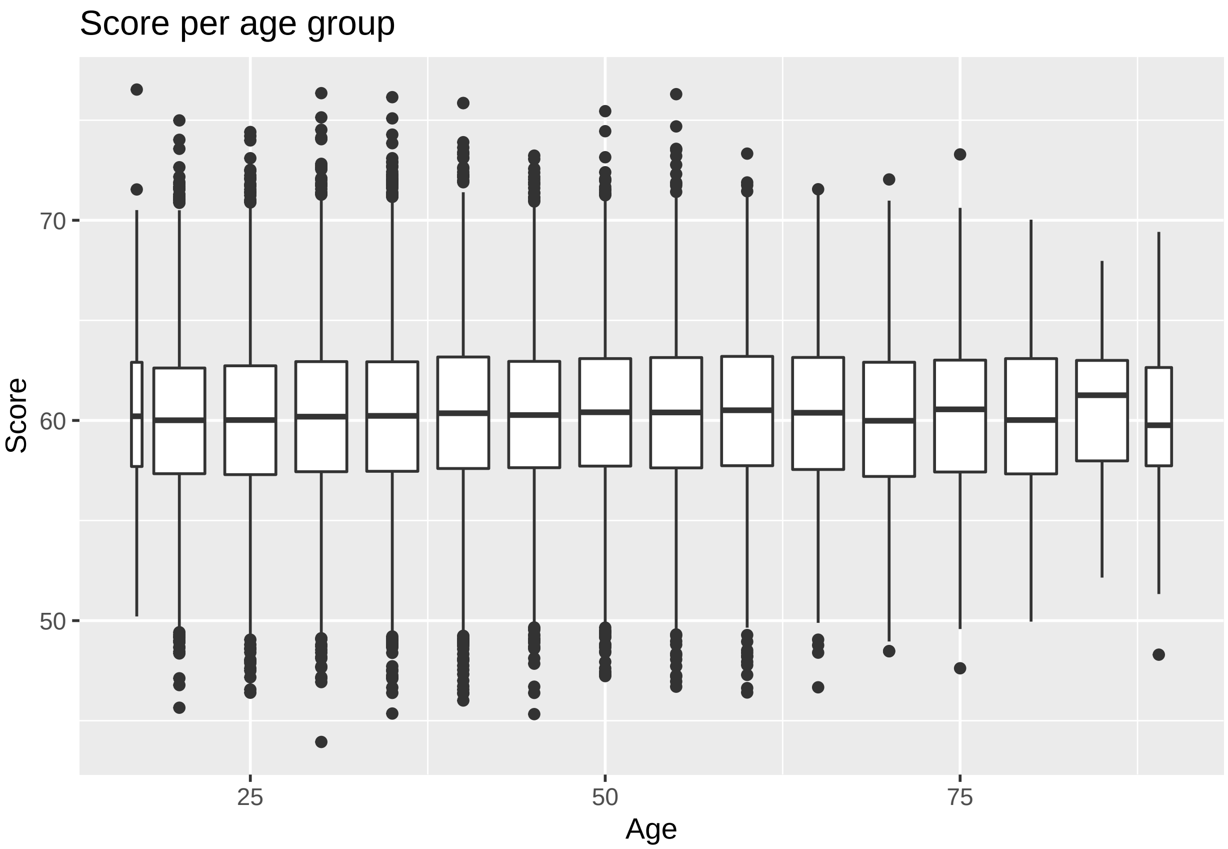
Explanation: The graph represents the number of people of different age groups who have the insurance with different marital status. It is obvious that the proportion of those under 50 and “never-married” is the highest, besides its is relatively right-skewed. According to the graph, the majority of the “never-married” clients are under 25 years old. Another important group is “married-civ-spouse”. The maximum number of people on the graph is 750 . Last but not least is the group of “divorced” clients. The graph, which is relatively even with no serious fluctuations, reaches its maximum at some 300 people.

```
insurancedata %>%
  group_by(age) %>%
  summarise( avg_hours_per_week = mean(hours_per_week))  %>%
  ggplot() +
  geom_line(aes(x = age, y = avg_hours_per_week)) +
  xlab("Age") + ylab("Average number of working hours") +
  ggtitle("Average number of working hours spent by age")
```



Explanation: It is obvious from the given graph that people who are over 25, but under 62.5 years old work almost 45 hours per week. The older they get the less they work. However, it is quite difficult to make any definite statement because in some cases there is not enough evidence (e.g., there is just one person who is 86 years old, three people who are 87 years old and two people who are 89). The fluctuations at the end of the graph may be explained by the volume of the data and its variability.

```
insurancedata %>%
   group_by(age) %>%
  #creating a boxplot
  ggplot(aes(x = age, y = score)) +
  geom_boxplot(aes( group = cut_width(age, 5))) +
  xlab("Age") + ylab("Score") + ggtitle("Score per age group")
```



Explanation: According to the graph the score for each age group is approximately the same (60). Some of the boxplots do not have any outliers because the age groups do not have enough information. Apart from that, all boxplots look quite similar with no exception. The minimum score is below 50, and the maximum - almost 80. It is essential to mention that the size of the boxplots is almost the same among all age groups, meaning that the insurance company has a policy and approves clients with a specific score.

3)

```
# creating a table with minimum, maximum and average values of hours per week
# for each age group
 insurancedata %>%
  group_by(education_num) %>%
  summarise(minimum_value = min(hours_per_week),
            maximum_value = max(hours_per_week),
            average = mean(hours_per_week))
```

```
## # A tibble: 16 x 4
##    education_num minimum_value maximum_value average
##            <int>         <int>         <int>   <dbl>
##  1             1            10            75    36.6
##  2             2             4            96    38.8
##  3             3             3            99    38.9
##  4             4             2            99    39.0
##  5             5             1            99    38.4
##  6             6             1            99    37.0
##  7             7             1            99    34.0
##  8             8             4            99    35.4
##  9             9             1            99    40.6
## 10            10             1            99    38.9
## 11            11             1            99    41.7
## 12            12             1            99    40.8
## 13            13             1            99    42.5
## 14            14             1            99    43.6
## 15            15             2            99    47.6
## 16            16             1            99    46.6
```

Explanation: According to the table, it is obvious that the maximum value of working hours per week among all age groups is 99, while the minimum equals 1 hour per week. The average column shows that the majority of clients tend to spend less than 47 hours per week on average, despite the level of education.

Statistical Analysis: Confidence Interval, Hypothesis Test, and Model or Machine Learning

1)

```
#filtering the data with respect to the marital status
# and number of hours spent at work
Married_over_21_hours_per_week = insurancedata %>%
  filter(marital_status == "Married-civ-spouse" & hours_per_week > 21)
# selecting the number of rows for the future
dim(Married_over_21_hours_per_week) [1]
```

```
## [1] 21223
```

```
#creating a bootstrap algorithm to have 21223 (the number of rows in the initial sample)
# sample means in order to make a confidence interval
bootstrap_samplemeans =
  do(500)*mean(~hours_per_week, data = sample_n(Married_over_21_hours_per_week,
                                           size=dim(Married_over_21_hours_per_week) [1],
                                           replace=TRUE))

bootstrap_samplemeans = bootstrap_samplemeans %>%
  rename(bootstrap_samplemean = mean)
head(bootstrap_samplemeans)
```

```
##   bootstrap_samplemean
## 1             44.87005
## 2             44.87570
## 3             44.88286
## 4             45.00664
## 5             44.87429
## 6             44.93738
```

```
# making the confidence interval with 95% confidence
confidenceinterval = quantile(bootstrap_samplemeans$bootstrap_samplemean,
                              probs = c(.025, .975))
confidenceinterval
```

```
##     2.5%    97.5%
## 44.76037 45.01995
```

I am 95% confident that the average number of working hours of married people mean is more than 44.75 hours, but less than 45.01 hours.

2)

H_0 : The population mean work hours per week equals 40. H_A : The population mean work hours per week does not equal 40.

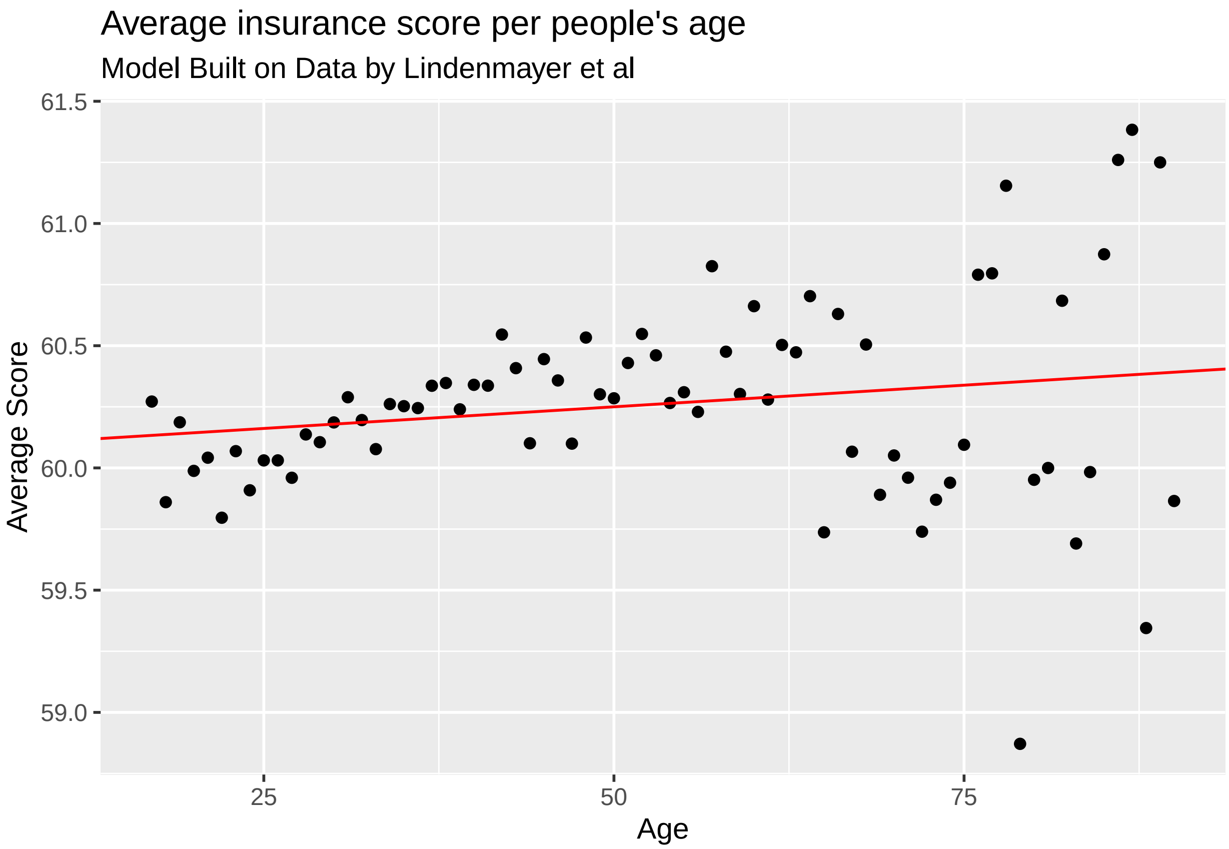
```
# I use built-in test
t.test(insurancedata$hours_per_week, alternative = 'two.sided', mu=40)
```

```
##
##  One Sample t-test
##
## data:  insurancedata$hours_per_week
## t = 7.5332, df = 48841, p-value = 5.036e-14
## alternative hypothesis: true mean is not equal to 40
## 95 percent confidence interval:
##  40.31249 40.53228
## sample estimates:
## mean of x
##  40.42238
```

Explanation: Since my p-val (5.036e-14 = 0.000000000000051) is less than the p-value (0.05), I have enough evidence to reject the initial hypothesis, meaning that the real sample mean does not equal 40 hours.

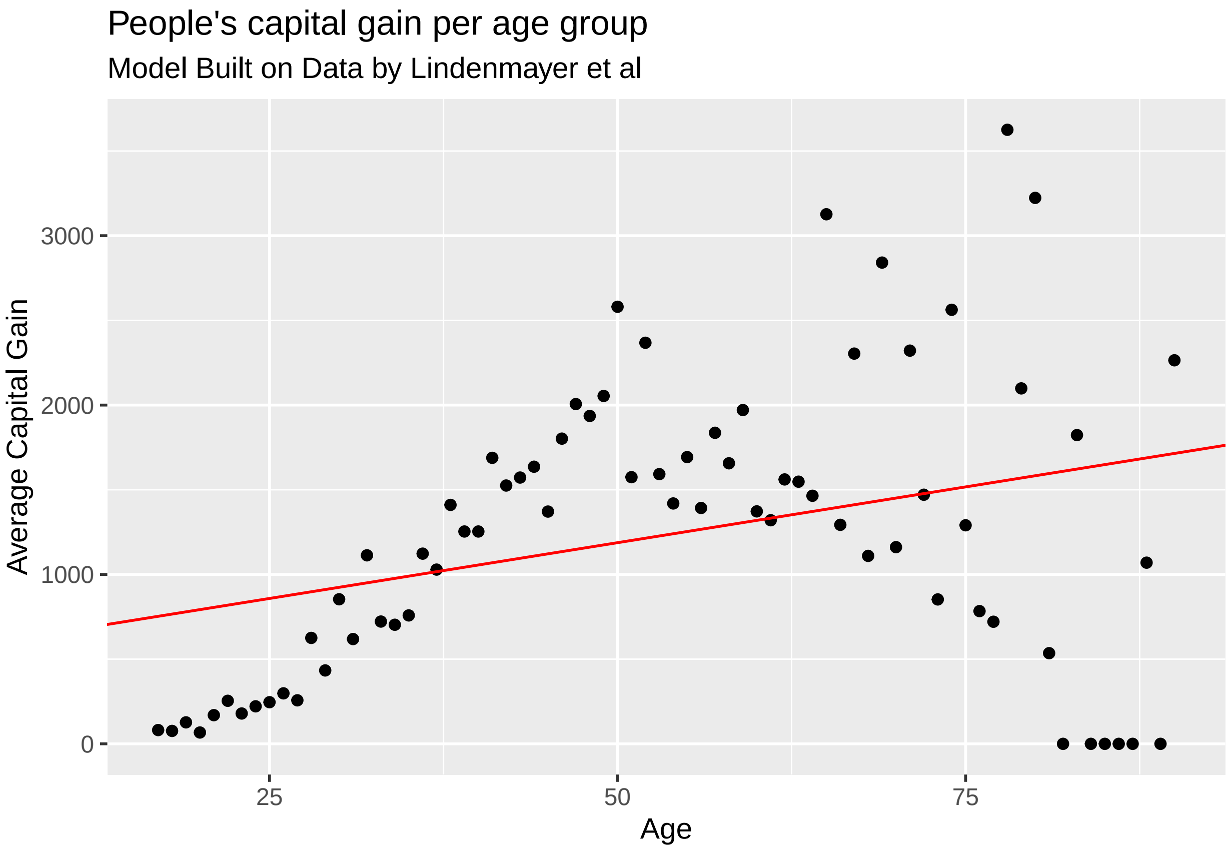
3)

```
# grouping my data by age and summarizing average score per each age group
data = insurancedata %>%
  group_by(age) %>%
  summarise(scoreperage = mean(score))
#creating the regression line
model = lm(data$scoreperage ~ data$age)
#plotting the graph
data %>%
  ggplot() +
  geom_point(aes(x = age,
                 y = scoreperage)) +
  labs(title = "Average insurance score per people's age",
       subtitle = "Model Built on Data by Lindenmayer et al") +
  geom_abline(intercept = model$coefficients[1],
              slope = model$coefficients[2],
              color="red") +
  xlab("Age") + ylab("Average Score")
```



Explanation: The graph illustrates the relationship between the group age and the average score. It is obvious that the slope of the linear regression line is positive, meaning that with an increase in age the average score is expected to increase on average too.

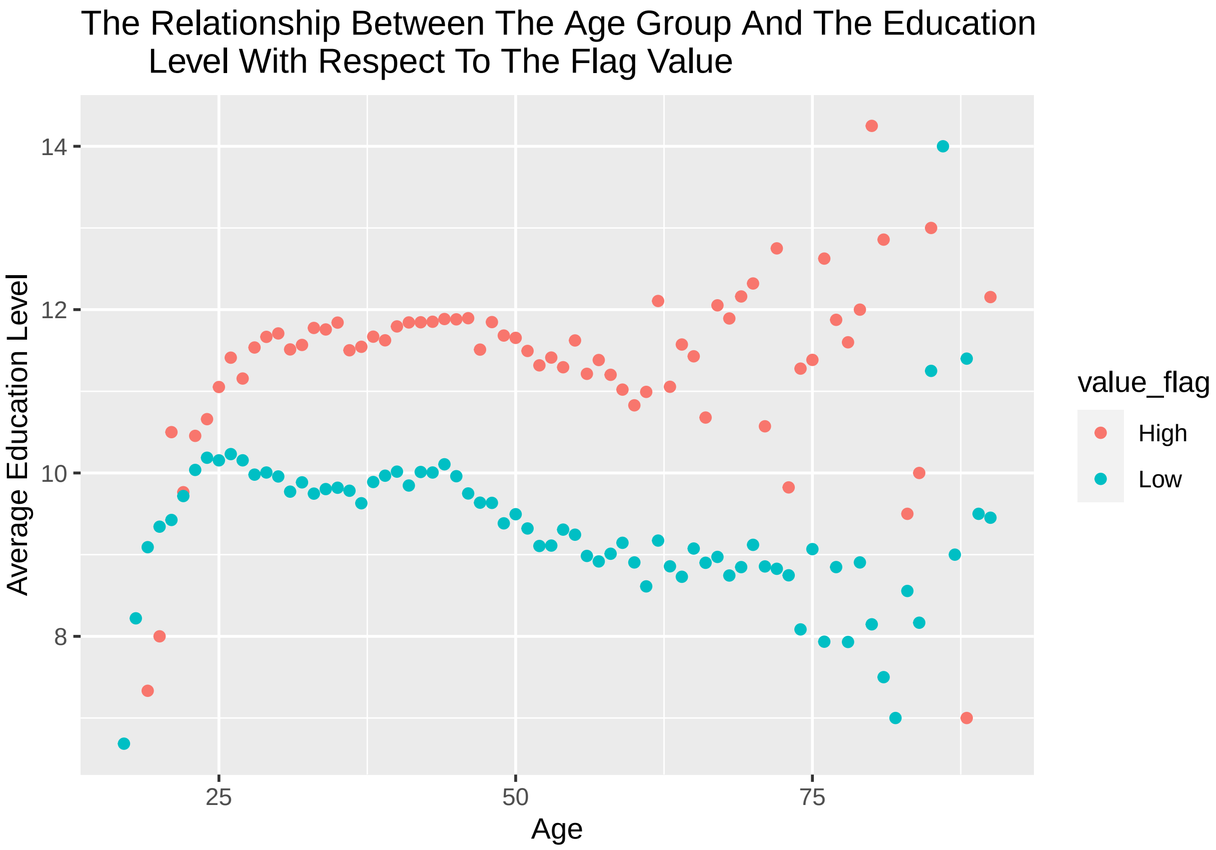
```
#creating a linear model

#predicting the average capital gain per age group by using the regression model.
#grouping my data by age and summarizing the average capital gain for each age group
data = insurancedata %>%
  group_by(age) %>%
  summarise(capgainperage = mean(cap_gain))

model = lm(data$capgainperage ~ data$age)

data %>%
  ggplot() +
  geom_point(aes(x = age,
                 y = capgainperage)) +
  labs(title = "People's capital gain per age group",
       subtitle = "Model Built on Data by Lindenmayer et al") +
  geom_abline(intercept = model$coefficients[1],
              slope = model$coefficients[2],
              color="red") +
  xlab("Age") + ylab("Average Capital Gain")
```



Explanation: The graph shows the relationship between the group age and the average capital gain of insurance clients. It seems that the slope of the suggested linear regression is positive, meaning that with the increase of the age the average capital gain is expected to increase too. Besides, there are some dots which equal zero, meaning that people did not earn any money at all. It is essential to say that the linear regression is not quite suitable because the data is messy.

```
insurancedata %>%
  group_by(age, value_flag) %>%
  summarise(averageeduc = mean(education_num)) %>%
  ggplot() +
  geom_point(aes(x = age, y = averageeduc, colour = value_flag )) +
  labs(title = "The Relationship Between The Age Group And The Education
       Level With Respect To The Flag Value") +
  xlab("Age") + ylab("Average Education Level")
```

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



Explanation: The given graph depicts the relationship between the education level and the value of a person. It is obvious that on average people with better education tend to be more valuable to the insurance company. However, there are people who are considered to be "low value" with a 14 level of education.

Conclusion

Throughout the project I used different coding techniques, which I learnt during classes. I created different graphs and boxplots in order to have a suitable representation of the data. Moreover, I stated a hypothesis and tested it. Apart from that I learnt to create confidence intervals for an estimator and different linear regressions. Also, from the following graphs it could be mentioned that the main audience of the insurance company consists of young and middle-aged people. On average they spend about 42 hours at work per week and have approximately the same score. Another important part of my project was to trace a relationship between the flag status of a client and his/her education level. I noticed that there is a relationship between the flag status of a client and his/her education. Clients with a high value flag level had a better degree almost among all age groups. However, I should look at other parameters, to be sure that there are no other factors that may affect a client's reputation.