Rcode for GV900 Homework 2

GV900 Political Explanation

Pierluigi De Rogatis

09-12-2021

```r
# Firstly, it will be advisable to clean the console and the environment from
previously used data and values by using the following functions:

rm(list=ls(all=TRUE))
cat("\014")
```

1. Load packages

```
#  I load the required packages in order to enable the following code to draw
graphs and analyse regressions by using the "library" function since I have
already installed them (using the "install.packages" function):

library(ggplot2)

# To create graphs and visualise data.

library(gmodels)

# To test, print, or summarise a general linear hypothesis for a regression
model.

library(lattice)

## Warning: package 'lattice' was built under R version 4.1.2

library(survival)

## Warning: package 'survival' was built under R version 4.1.2

library(Formula)
library(Hmisc)

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

# For leveraged data analysis, high-level graphics, and utility operations.
However, I needed to load also the "lattice", "survival", and "Formula"
packages since required by my RStudio to correctly run the Hmisc package.

library(stargazer)

##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary
Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

# To produce LaTeX code, HTML code and ASCII text for well-formatted tables
with regression analysis results.
# For this, I will cite: "Hlavac, Marek (2018). stargazer: Well-Formatted
Regression and Summary Statistics Tables."
```

```
library(carData)
library(effects)

## Warning: package 'effects' was built under R version 4.1.2

## Use the command
##     lattice::trellis.par.set(effectsTheme())
##   to customize lattice options for effects plots.
## See ?effectsTheme for details.

# To construct an "eff" object for a term in a regression that models a
response as a linear function of main effects and interactions of factors and
covariates. However, I needed to load also the "carData" packages since
required by my RStudio to correctly run the gridExtra package.

library(gridExtra)

# To provide user-level functions to work with "grid" graphics.

# Sources: https://www.educba.com/list-of-r-packages/;
https://www.rdocumentation.org/packages/stargazer/versions/5.2.2/topics/starg
azer; https://www.rdocumentation.org/packages/effects/versions/4.2-
0/topics/effect; https://rdrr.io/cran/gridExtra/.
```

2. Load the data set

```
# Now, I can import the data set in my console by using the "read.csv"
function to read the Comma Separated Values file. Moreover, I will use the
"paste0" function to concatenate all elements without a separator (source:
https://r-lang.com/paste0-function-in-r-with-example/). The path to finding
the data is stored in an object called "myPath":

td <- read.csv(paste0(myPath, "titanic.csv"))
```

3. Number of unit of observation

```
# To see the number of observation, i.e. the number of individual passengers
stored in the "td" dataset, I will use the function "dim" which retrieve the
dimension of an object as the number of rows and of columns respectively
(source:
https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/dim):

dim(td)

## [1] 1309   15

# As displayed by the console, the number of observations / individual
passengers / rows in the "td" dataset is:

# 1309
# More than asked in this assignment, another method to display the number of
rows (passengers) could be to use the "nrow" function. I will display it as a
comment to not disrupt the coding:

# nrow(td)
```

4. Frequency table for survival

```
# To create a frequency table to display how many passengers survived or not,
I will use the "data.frame" and "table" functions, and I will save it as an
object called "ft_survived", thus:

ft_survived <- data.frame(table(td$survived))

# More than asked in this assignment, I will change the name of the first
column to have a better understanding of the data displayed in the table by
using the "colnames" function. Thus:

colnames(ft_survived)[colnames(ft_survived) == "Var1"] <- "Survived (1),
Deceased (0)"

# Then, I will print the table using the "print" function:

print(ft_survived)

##   Survived (1), Deceased (0) Freq
## 1                          0  809
## 2                          1  500
```

5. Survival percentage

```
# To add the percentage, I will follow a few steps. First of all, it is
necessary to calculate the relative frequencies using the "prop.table"
function "prop.table(ft_survived$Freq)". Then, we can convert the resulting
relative frequencies into percentages by multiplying them by 100
"prop.table(ft_survived$Freq)*100". Finally, it is essential to insert the
new column Percentage into the already existing ft.survived frequency table.
Thus, the final code:

ft_survived$Percentage <- prop.table(ft_survived$Freq) * 100

# It should be noticed that, in the assignment, there is no request to round
the percentage results. However, it would be advisable to do so in order to
have a more understandable table by using the "round" function. Namely:

ft_survived$Percentage <- round(ft_survived$Percentage, digits = 2)
```

```
# Then, I will print only the percentage column to display the survival
percentage. To do so, I will call only the percentage for the row == 1, since
I am interested in the percentage of survival. Therefore, I will firstly
store only the data of survived people in another table, and store it in an
object called "sur_rate":

sur_rate <- ft_survived[ft_survived == "1",]

# Finally, I will print only the percentage of survived people using the
"print" function:

print(sur_rate$Percentage)

## [1] 38.2

# Besides, I will also print the result as a comment, thus:

# 38.2
```

6. Frequency table for socio-economic class

```
# To create a frequency table to display the different socio-economic class
of passengers, I will use the "data.frame" and "table" functions, and I will
save it as an object called "ft_pclass", thus:

ft_pclass <- data.frame(table(td$pclass))

# More than asked in this assignment, I will change the name of the first
column to have a better understanding of the data displayed in the table by
using the "colnames" function. Thus:

colnames(ft_pclass)[colnames(ft_pclass) == "Var1"] <- "Passenger Ticket
Class"

# Then, I will print the table using the "print" function:

print(ft_pclass)

##   Passenger Ticket Class Freq
## 1                      1  323
## 2                      2  277
## 3                      3  709
```

7. Independent and dependent variables

```
# The research question is: "how does the socio-economic class of passengers
influence the likelihood of passenger survival?". While the hypothesis is
"the socio-economic class of passengers is positively associated with
passenger survival".
```

```
# Therefore, the dependent variable (Y), i.e. the expected outcome, is the
probability of survival (passenger survival).

# On the other hand, the independent variable (X), i.e. the cause, is the
socio-economic class of passengers.
```

8. Cross-tabulation

```
# Since both "survived" and "pclass" are categorical variables, the way to
test their relationship is by using the tabular association test (cross-
tabulation).

# Before to do that, I will transform the "pclass" and "survived" variables
in factor variables to give an ordering to the variable, thus:

td$pclass_ord <- factor(td$pclass,
                        level = c("1", "2", "3"),
                        ordered = TRUE)

td$survived_ord <- factor(td$survived,
                          level = c("0", "1"),
                          ordered = TRUE)

# To do that, I will use the "CrossTable" function, remembering to insert
first the dependent variable (Y) and then the independent variable (X)
(source:
https://www.rdocumentation.org/packages/gmodels/versions/2.18.1/topics/CrossT
able).

# However, I need to adjust the coding.
# Firstly, I will omit row percentages by imposing the argument "prop.r" as
FALSE.
# Then, I will omit the cell percentages by imposing the argument "prop.t" as
FALSE.
# After that, I will omit the chi-squared contributions by imposing the
argument "prop.chisq" as FALSE.
# Finally, I will insert the chi-squared test statistic by imposing the
argument "chisq" as TRUE, thus:

CrossTable(td$survived_ord, td$pclass_ord,
           prop.r = FALSE,
           prop.t = FALSE,
           prop.chisq = FALSE,
           chisq = TRUE)

## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  1309
##
##
##                  | td$pclass_ord
## td$survived_ord |         1 |         2 |         3 | Row Total |
## ----------------|-----------|-----------|-----------|-----------|
##               0 |       123 |       158 |       528 |       809 |
##                 |     0.381 |     0.570 |     0.745 |           |
## ----------------|-----------|-----------|-----------|-----------|
##               1 |       200 |       119 |       181 |       500 |
##                 |     0.619 |     0.430 |     0.255 |           |
## ----------------|-----------|-----------|-----------|-----------|
##    Column Total |       323 |       277 |       709 |      1309 |
##                 |     0.247 |     0.212 |     0.542 |           |
## ----------------|-----------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  127.8592     d.f. =  2     p =  1.720826e-28
##
##
##
```

9. Table information

```
# To see the survival percentage for each class, we have to analyse the
number below the number of passengers for each class ticket. However, we are
interested only in the survival rate, not in the death rate, so we will
display the numbers displayed in the row labelled "1" (since it represents
survived people).

# (a) In the first class, 61.9% (0.619) of passengers survived (in absolute
terms, 200 passengers; in relative terms, 200 passengers over 323).

# (b) In the second class, 43.0% (0.430) of passengers survived (in absolute
terms, 119 passengers; in relative terms, 119 passengers over 277).

# (c) In the third class, 25.5% (0.255) of passengers survived (in absolute
terms, 181 passengers; in relative terms, 181 passengers over 709).
```

10. First hurdle

```
# The hypothesis is: "socio-economic class of passengers is positively
associated with passenger survival."

# Then, if the socio-economic class of passengers is operationalised as the
class tickets of passengers (pclass), we could say that there is a positive
relationship with survival rate indeed.

# In fact, people with the most expensive ticket (1st class, which represents
the highest socio-economic class as proxy) had the highest survival rate
(61.9%), while people with the least expensive ticket (3rd class) had the
lowest survival rate (25.5%). Therefore, the statistics, for now, are
confirming our hypothesis.
```

11. Fill the statement

```
# "Since the test statistic produces a p-value smaller than [0.001], we can
[reject] the null hypothesis of no association at [99.9] % confidence level.
We thus [find] support for our hypothesis."

# 1. (l) 0.001

# 2. (n) reject

# 3. (e) 99.9

# 4. (o) find
```

12. Relationship test

```
# Since "female" is a categorical variable, and "fare" is a
continuous/numeric variable, the most appropriate bivariate hypothesis test
is the "difference of means".

# First of all, I will transform the "female" variable (currently a
character) into a dummy variable (numeric) to display a correct test. To do
so, I will use the "ifelse" function, and I will create a new column in my
dataset called "gender". Further, I will assign the value of "1" to Female
and "0" to Male passengers, thus:

td$gender <- ifelse(td$female == "Female", 1, 0)

# To implement this test, I will use the "t.test" function. Further, I will
assume equal variance between the variables by imposing the option
"var.equal" as TRUE. Moreover, I will store the test as an object called
"t.ff" (source:
https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/t.test).
Thus:
```

```
t_ff <- t.test(fare ~ gender,
               data = td,
               var.equal = TRUE)

# Finally, I will print the test by using the print function

print(t_ff)

##
##  Two Sample t-test
##
## data:  fare by gender
## t = -6.824, df = 1306, p-value = 1.351e-11
## alternative hypothesis: true difference in means between group 0 and group
1 is not equal to 0
## 95 percent confidence interval:
##  -25.80481 -14.28092
## sample estimates:
## mean in group 0 mean in group 1
##        26.15382        46.19668
```

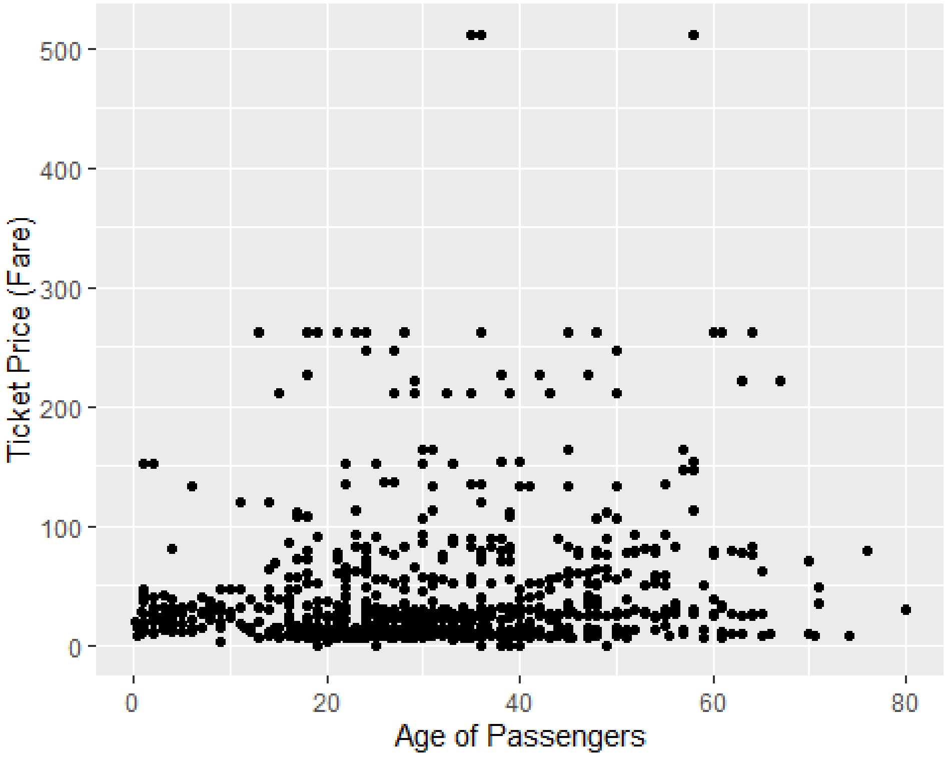
13. Results interpretation

```
# First of all, I have to highlight the in the "gender" variable 1 is for
Female and 0 is for Male. Therefore, the average ticket price for Female
(group 1) is 46.20, while the average ticket price for Male (group 0) is
26.15.

# With this data, I could argue that female passengers, on average, tend to
have a more expensive ticket since the mean in group 1 is greater than the
mean in group 0.

# However, the result could be statistically insignificant. In this case,
fortunately, this is not true. Indeed, the difference of means between the
two is 20.04286 GBP, yielding a t-statistic of -6.824, with a degree of
freedom of 1306 and a p-value of less than 0.001. Hence, we are 99.9%
confident that our hypothesis is statistically significant.
```

14. Age-Fare graphical relation

```
# To graphically display the relationship between "age" and "fare", I will
use the "ggplot" function. However, since both variables are numerical, I
will use a scatterplot by using the "geom_point" function.

# Firstly, I will use the "ggplot" function to tell R in which data frame the
variables are stored, in this case, "td". Besides, I will store the graph as
an object called "g_fage", thus:
```

```
g_fage <- ggplot(td)

# Then, I have to tell to RStudio what variables to plot inside the chart
thanks to the "aes" function, in this case: "age" on the x-axis and "fare" on
the y-axis, thus:

g_fage <- g_fage + aes(x = age, y = fare)

# Then, I use the "geom_boxplot" function to tell R what graph I want, in
this case, a boxplot Furthermore, I added the option "na.rm" to remove all
variables without a measurement. Hence:

g_fage <- g_fage + geom_point(na.rm = TRUE)

# Besides, and more than asked in this assignment, I will use the "ylab"
function to change the name of the y-axis in the boxplot previously created:

g_fage <- g_fage + ylab("Ticket Price (Fare)")

# Further, I will use the "xlab" function to change the name of the X-axis:

g_fage <- g_fage + xlab("Age of Passengers")

# Finally, I will print the graph using the print function:

print(g_fage)
```



15. Age-Fare test statistic

```
# Since we have two numeric variables (age and fare), the best statistic test
is the correlation analysis with the Person's correlation coefficient (r).

# First of all, I have to create a matrix with the two studied variables
using the "as.matrix" function. I will store the matrix in an object called
"fage_mat". Hence:

fage_mat <- as.matrix(td[c("fare", "age")])

# Then, I will use the "rcorr" function to run the correlation analysis, and
I will store it in an object called "r.fage". Further, I will use the "type"
option to tell the programme to run a Person's correlation analysis:

r_fage <- rcorr(fage_mat, type = "pearson")

# Finally, I will print the correlation using the print function:

print(r_fage)

##      fare  age
## fare 1.00 0.18
## age  0.18 1.00
##
## n
##      fare  age
## fare 1308 1045
## age  1045 1046
##
## P
##      fare age
## fare       0
## age  0
```

16. Results interpretation

```
# Based on the correlation analysis above, the correlation coefficient
between age and fare is positive, with a value of 0,18. Thus, older people
pay more for expensive tickets.

# The p-value for this test is so low that the R programme rounded it to 0.
So, we can conclude that our confidence level is 99.9% at least and so we can
reject the null hypothesis with 99.9% confidence.

# Overall, the estimated correlation coefficient between the age of
passengers and ticket fare is positive (0.18) and statistically significant
at a 99.9% confidence level. We thus find support for our hypothesis since
```

```
our question was: "do older passengers tend to have a more expensive ticket
compared with younger passengers?".
```

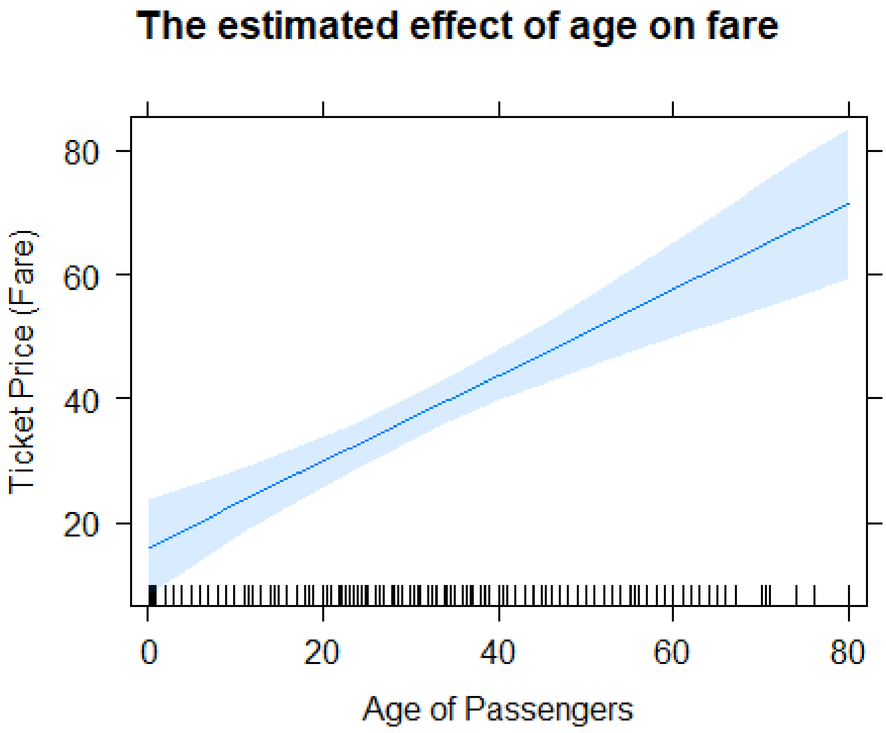
17. Simple Linear Regression

```
# To regress fare on age, I will use the "lm" function since it is a linear
model regression. In the argument, I will put first the dependent variable
(fare) and then the independent variable (age). Moreover, I will store the
regression in an object named "reg_fage". Thus:

reg_fage <- lm(fare ~ age, data = td)

# Then, I will use the "stargazer" function to produce a regression table.
Moreover, I will put "text" as the "type" argument. Hence:

stargazer(reg_fage, type = "text")

##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                fare
## -----------------------------------------------
## age                          0.692***
##                               (0.118)
##
## Constant                     16.021***
##                               (3.909)
##
## -----------------------------------------------
## Observations                   1,045
## R2                             0.032
## Adjusted R2                    0.031
## Residual Std. Error     54.851 (df = 1043)
## F Statistic          34.428*** (df = 1; 1043)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

18. Graphical regression

```
# To graphically display the relationship between age and fare, I will use
the "effect" function. I will first store the result in an object called
"eff_fage". Moreover, in the argument term (the quoted name of a term), I
will put the independent variable (age), while in the argument mod, I will
put the object that stores the regression (reg_fage). (source:
https://www.rdocumentation.org/packages/effects/versions/4.2-
0/topics/effect). Hence:

eff_fage <- effect(term = "age", mod = reg_fage)
```

```
# Then, I will use the "plot" function to graphically display the regression.
Moreover, I will adjust the graph to be more readable. So, I will use the
main, xlab and ylab arguments to change its labels. Hence:

plot(eff_fage,
     main = "The estimated effect of age on fare",
     xlab = "Age of Passengers",
     ylab = "Ticket Price (Fare)")
```



19. Comment of relationship

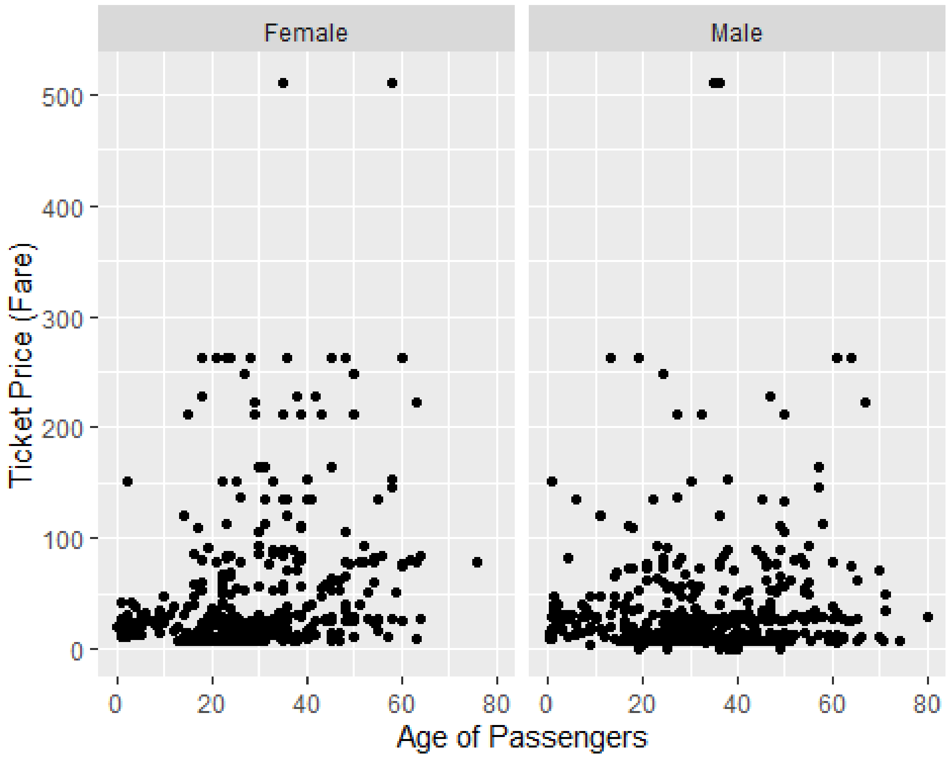
```
# Graphically, it is straightforward to note a positive relationship between
age and fare since the curve is upward.

# Moreover, the estimated regression line is:
# fare = 16.0205 + 0.6922*age
# Since it is a simple regression, this is a straight line expressed as y =
mx + q, and so the m (slope) is 0.6922, which is positive (m > 0), indeed.

# Finally, we can see from the table that both the constant value (intercept)
and the correlation value have a p-value of less than 0.01, thus we have a
confidence of at least 99% that the relationship is statistically
significant.

# However, the adjusted R-squared is 0.031, meaning that only 3.1% of the
```

```
variance in the dependent variable is explained by the independent variable
(source: https://www.investopedia.com/terms/r/r-squared.asp). However, since
our independent variable is statistically significant, we can still conclude
that age and fare are positively correlated (source:
https://statisticsbyjim.com/regression/interpret-r-squared-regression/).
Indeed, our F-statistic has a p-value < 0.01, thus further proving the
hypothesis that our X (age) is related to Y (fare). (source:
https://quantifyinghealth.com/f-statistic-in-linear-regression/).
```

20. Age-Fare relationship on gender

```
# To draw a plot of age and fare based on gender, I will use the "facet_wrap"
function and store the graph as an object called "gg_fage".

# Firstly, I will use the "ggplot" function to tell R in which data frame the
variables are stored, in this case, "td":

gg_fage <- ggplot(td)

# Then, I have to tell R what variables to plot inside the chart with the
"aes" function, in this case, age on the x-axis and fare on the y-axis, thus:

gg_fage <- gg_fage + aes(x = age, y = fare)

# Then, I use the "geom_boxplot" function to tell R what graph I want, in
this case, a boxplot Furthermore, I added the option "na.rm" to remove all
variables without a measurement. Hence:

gg_fage <- gg_fage + geom_point(na.rm = TRUE)

# Besides, and more than asked in this assignment, I will use the "ylab"
function to change the name of the y-axis in the boxplot previously created:

gg_fage <- gg_fage + ylab("Ticket Price (Fare)")

# Further, I will use the "xlab" function to change the name of the X-axis:

gg_fage <- gg_fage + xlab("Age of Passengers")

# To differentiate the graph, I will use the "facet_wrap" option, linking it
to the "female" variable in td. I did not use the dummy variable "gender"
since they are conceptually the same, but if I had used "gender" on the two
panels, I would have only seen the numbers 0 and 1, making it harder to
interpret the graph. For this reason, I used the "female" variable, which
will display the two panels as "Female" and "Male".

gg_fage <- gg_fage + facet_wrap( ~ female)
```

```
# Finally, I use the "print" function to display the boxplot:

print(gg_fage)
```



21. Multiple regression

```
# To estimate a multiple linear regression, I will use the "lm" function.
Further, I will store it as an object called "m.reg.fage". Besides, "gender"
is already a numeric dummy variable, so I do not need to make any other
adjustment to my code, thus:

m_reg_fage <- lm(fare ~ age + gender, data = td)

# Then, I will use the "stargazer" function to produce a regression table.
Moreover, I will put "text" as "type" argument. To display also the previous
regression model (reg.fage), I will add it inside the brackets Hence:

stargazer(reg_fage, m_reg_fage, type = "text")

##
## =====================================================================
##                                     Dependent variable:
##                      ------------------------------------------------
##                                            fare
##                               (1)                        (2)
## ---------------------------------------------------------------------
## age                         0.692***                 0.740***
##                             (0.118)                  (0.116)
##
## gender                                              23.039***
##                                                      (3.447)
##
## Constant                   16.021***                  6.029
##                             (3.909)                  (4.111)
##
## ---------------------------------------------------------------------
## Observations                 1,045                    1,045
## R2                           0.032                    0.072
## Adjusted R2                  0.031                    0.070
## Residual Std. Error    54.851 (df = 1043)       53.737 (df = 1042)
## F Statistic         34.428*** (df = 1; 1043) 40.267*** (df = 2; 1042)
## =====================================================================
## Note:                                     *p<0.1; **p<0.05; ***p<0.01
```

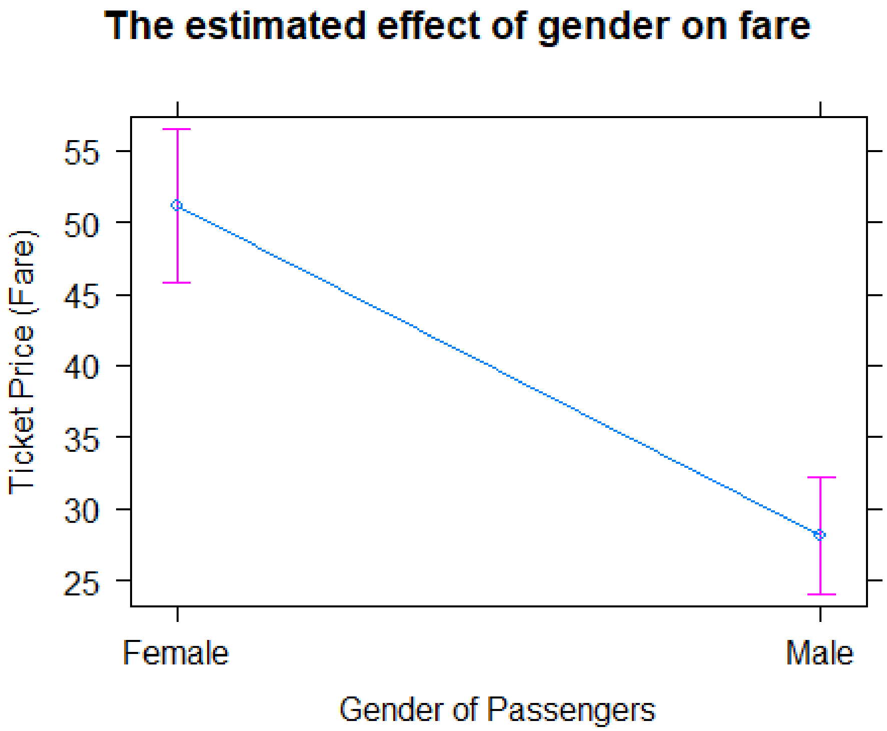
22. Fill the statement (2)

```
# "Since [adjusted R2 (R-squared)] is [smaller] for the [first] model, the
[second] model fits the data better."

# 1. (d) adjusted R2 (R-squared)

# 2. (f) smaller

# 3. (h) first

# 4. (i) second
```

23. Female-Fare relation

```
# To graphically display the relationship between gender and fare, I will use
the "effect" function. Firstly, I will create a new regression with the
variable "female" rather than the numeric variable "gender", storing it in an
object called "g.ref.fage". I will  store the result in an object called
"eff.femage". Moreover, in the argument term (the quoted name of a term), I
will put the independent variable (female), while in the argument mod, I will
put the object that stores the regression (g.reg.fage). (source:
https://www.rdocumentation.org/packages/effects/versions/4.2-
0/topics/effect). Hence:

g_reg_fage <- lm(fare ~ age + female, data = td)
eff_femage <- effect(term = "female", mod = g_reg_fage)

# Then, I will use the "plot" function to graphically display the regression.
Moreover, I will adjust the graph to be more readable. So, I will use the
```

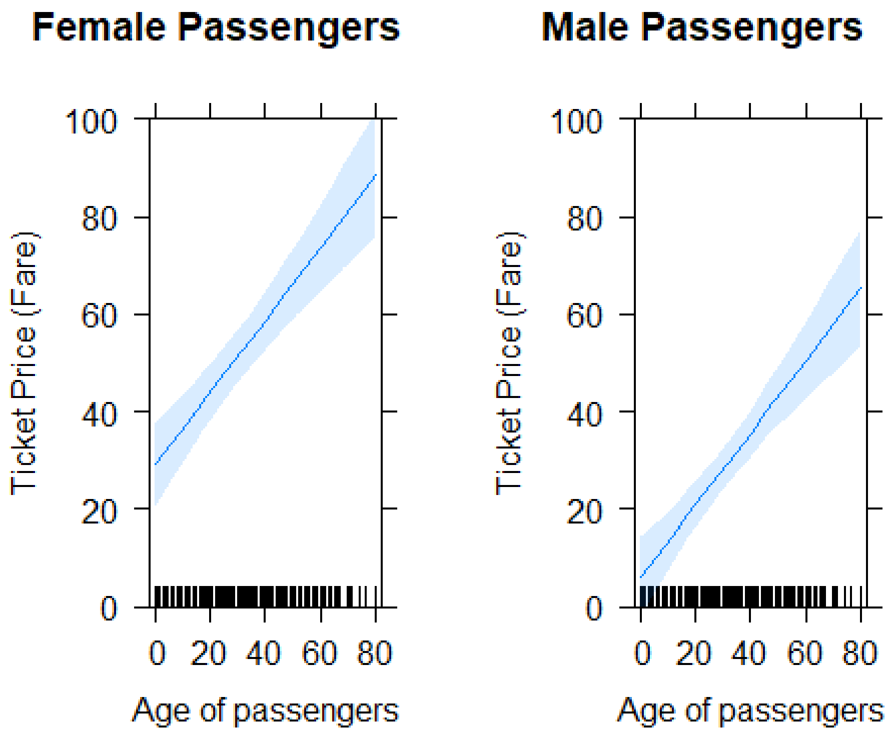
```
main, xlab and ylab arguments to change its labels. Hence:

plot(eff_femage,
     main = "The estimated effect of gender on fare",
     xlab = "Gender of Passengers",
     ylab = "Ticket Price (Fare)")
```



24. Age-Fare(-Female) relation

```
# To graphically display the relationship between age and fare, based on
gender, I will use the "effect" function, inserting the multiple regression
model (m_reg_fage) as the "mod" argument.

# Firstly, I will use use the "effect" function to separate the regression
results based on gender. For female, I will store the result in an object
called "fml_eff_fage", hence:

fml_eff_fage <- effect(term = "age",
                       mod = m_reg_fage,
                       given.values = c("gender" = 1))

# For male, I will store the result in an object called "ml_eff_fage", thus:

ml_eff_fage <- effect(term = "age",
                      mod = m_reg_fage,
                      given.values = c("gender" = 0))
```

```
# Then, I will translate the result into graphs using the "plot" function. I
will store the graph for females in an object called "fml_fage". More than
asked in this assignment, in order to increase the comprehension of the data
displayed in the graph, I will use the "main", "xlab", and "ylab" arguments
to improve them. Further, I added the "ylim" argument in order to have an
equal representation between the two graphs. Hence:

fml_fage <- plot(fml_eff_fage,
                 main = "Female Passengers",
                 xlab = "Age of passengers",
                 ylab = "Ticket Price (Fare)",
                 ylim = c(0, 100))

# While I will store the graph for male in an object called "ml.fage", and I
will do the same improvements, thus:

ml_fage <- plot(ml_eff_fage,
                main = "Male Passengers",
                xlab = "Age of passengers",
                ylab = "Ticket Price (Fare)",
                ylim = c(0, 100))

# Then, I will use the "grid.arrange" function to display the two graphs
together. Moreover, I will use the "ncol" option to represent them one
alongside the other and not one above and below, thus:

grid.arrange(fml_fage, ml_fage, ncol = 2)
```



End of assignment “Homework 2”