Regression Model and Exploratory Data Analysis using R Rstudio

Bernardo Vimpi

10/11/2020

In this project we build a Linear Regression Model. We also perform the Exploratory Data Analysis (EDA) for our dataset using three common steps: Analyze the raw data, generate the Summary Statistics, and Data Visualization.

Packages

```
library(tinytex)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# loading the tidyverse package is sufficient but I decided to also load the other packages.

library(ggplot2)
library(dplyr)
library(tidyr)
library(readr)
library(purrr)
library(tibble)
library(stringr)
library(forcats)
library(skimr)
library(moderndive)
library(gapminder)
```

(LC5.1) Conduct a new Exploratory Data Analysis with the same outcome variable y being *score* but with *age* as the new exploratory variable x. Remember, this involves three things: Understand the data, Data Summary, and Data Visualization

Let's first create our dataset and store it in *evals_ch6* .

```
evals_ch6 <- evals %>%
  select(ID, score, bty_avg, age)
```

Exploratory Data Analysis

Step 1: Understand the Data.

```
evals_ch6 %>%
  select(score, age) %>%
  glimpse()
```

```
## Rows: 463
## Columns: 2
## $ score <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3, 2.8, 4.1, 3.4, 4.5, 3.8, 4.5, 4...
## $ age   <int> 36, 36, 36, 36, 59, 59, 59, 51, 51, 40, 40, 40, 40, 40, 40, 4...
```

Answer: By selecting only *score* and *age* we see that both variables are numerical. The dataset contains 463 rows/observations each corresponding to an individual course. We also have 2 columns, age and scores.

We can also go further by checking if there are any missing values in the dataset using the *is.null()* function. The "FALSE" output indicates that there are no missing values in our dataset.

```
evals_ch6 %>%
  select(score, age) %>%
  is.null()
```

```
## [1] FALSE
```

Step 2: Summary Statistics

Note: First, let's generate a random sample of 15 observations from our dataset.

```
evals_ch6 %>%
  select( score, age)%>%
  sample_n(size = 15)
```

```
## # A tibble: 15 x 2
##    score   age
##    <dbl> <int>
##  1   3.5    33
##  2   4.1    52
##  3   4.9    47
##  4   4.5    32
##  5   4.5    52
##  6   4.5    43
##  7   4.9    39
##  8   4      32
##  9   3.9    47
## 10   4.6    33
```

```
## 11   4.2    54
## 12   3.3    62
## 13   4.1    32
## 14   4.5    50
## 15   3.3    57
```

Let's generate the mean and median values for score and age

```
 evals_ch6 %>%
  select(score, age)%>%
  summarize( Mean_score = mean(score), Median_score = median(score),
             Mean_Age = mean(age), Median_Age = median(age))
```

```
## # A tibble: 1 x 4
##   Mean_score Median_score Mean_Age Median_Age
##        <dbl>        <dbl>    <dbl>      <int>
## 1       4.17          4.3     48.4         48
```

We can expand on the summary statistics by applying the *skim()* function for additional statistical outputs such as the measure of spread.

```
evals_ch6 %>%
  select(score, age)%>%
  skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	463
Number of columns	2
Column type frequency:	
numeric	2
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
score	0	1	4.17	0.54	2.3	3.8	4.3	4.6	5	
age	0	1	48.37	9.80	29.0	42.0	48.0	57.0	73	

Answer: Using the *skim()* function for the summary statistics we get additional details regarding the entire dataset and the two variables. First, we have 463 rows or observations and 2 columns, as stated earlier. Second, the column types are both numeric. Third, the *n_missing* column of the output shows that there are no missing values neither in Age nor in Score. Fourth, besides the mean and standard deviation for both age and score, we also have p0 or the 0th percentile, minimum value; the p25 or 25th percentile, 1st quantile; the p50 or 50th percentile, 2nd quantile or the median; the p75 of 75th percentile or 3rd quantile; and the p100 or 100th percentile, the maximum value (for example the highest age is 73 while the highest score is

5).Note also that the output shows a small plot for each variable and their respective skeweness. We'll do further analysis on that during the data visualization part.

Since we have two numeric variables, we can also analyze the strength of their linear relationship via the correlation coefficient.

Correlation Coefficient

```
evals_ch6%>%
  get_correlation(score ~ age)
```

```
## # A tibble: 1 x 1
##      cor
##    <dbl>
## 1 -0.107
```

```
# Alternatively we can also run the below line of code and it will produce the same result:

evals_ch6 %>%
  get_correlation(formula = score ~ age)
```

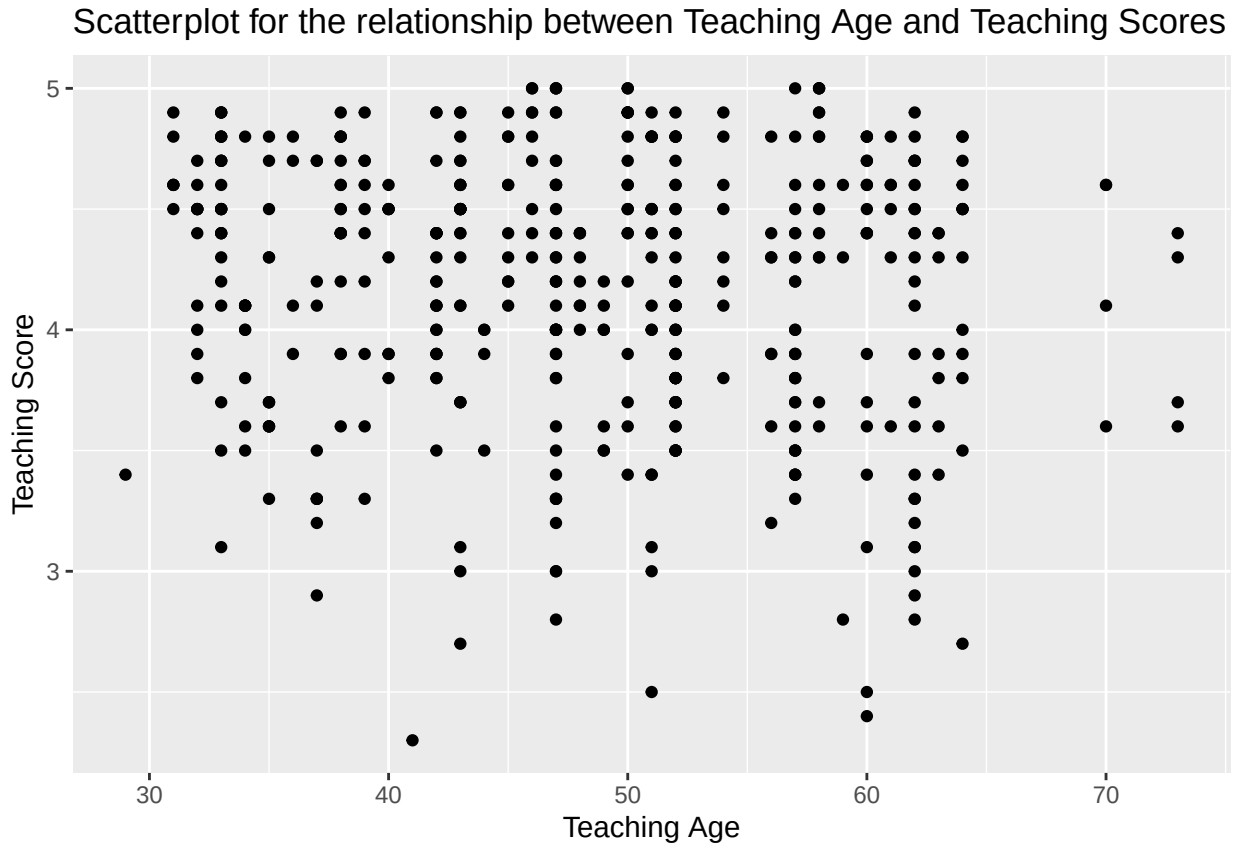
```
## # A tibble: 1 x 1
##      cor
##    <dbl>
## 1 -0.107
```

Answer: Based on the correlation coefficient score of -0.107, we see that both age and score have a negative linear relationship. Since the strength of the linear relationship between two numerical values range from -1 (perfectly negative) to 1(perfectly positive), with 0 indicating no relationship, in the data visualization of this exploratory data analysis we'll further determine if this negative linear relationship between age and score is strong of weak.

Step 3: DATA VISUALIZATION

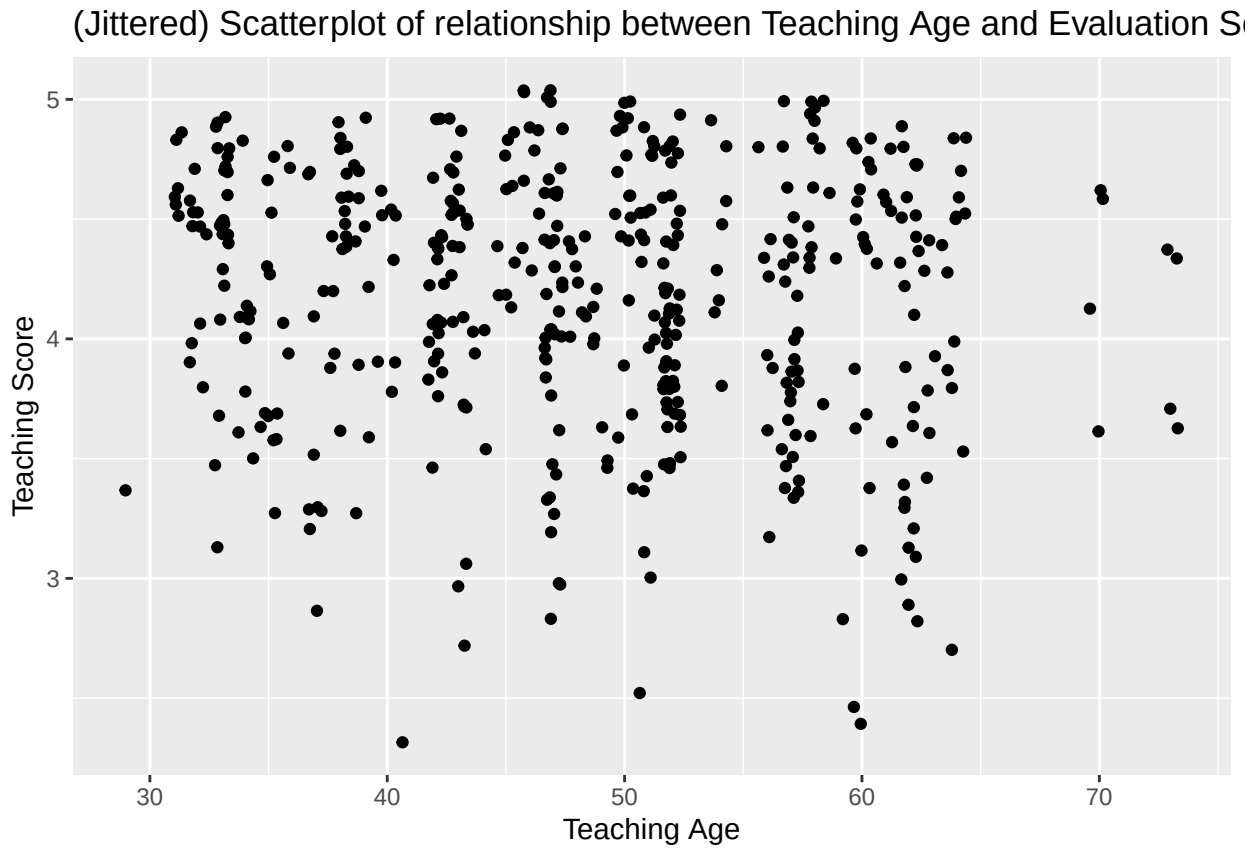
Since both variables, score and age, are numerical, we can visualize the relationship using a scatterplot as bellow:

```
ggplot(evals_ch6, aes( x= age, y= score)) +
  geom_point() +
  labs( x= "Teaching Age", y= "Teaching Score", title = "Scatterplot for the relationship between Teachi
```



Answer: Using the output from the scatterplot, we can see that the linear relationship between age and score is both negative(using the correlation coefficient of -0.107) and week based on the distribution of the points/observations on the plot. There are additional analysis we can perform here: a) most teaching age fall within the range of 31 and 65 years old. We notice also a few observations outside of that range; b) most teaching scores fall within the range of 3 and 5, with some points falling bellow 3; c) if we look carefully, we notice that some observations lie on line of 5 score, if we draw and imaginary line. It is very possible that some of the observations have been plotted on top of the other: an issue commonly known as *overplotting*; d) to to decipher and mitigate *overplotting* our data, we'll use the *geom_jitter()* method to plot the same scatterplot as below:

```
ggplot(evals_ch6, aes(x= age, y = score)) +
  geom_jitter() +
  labs( x = "Teaching Age", y= "Teaching Score", title= "(Jittered) Scatterplot of relationship between
```

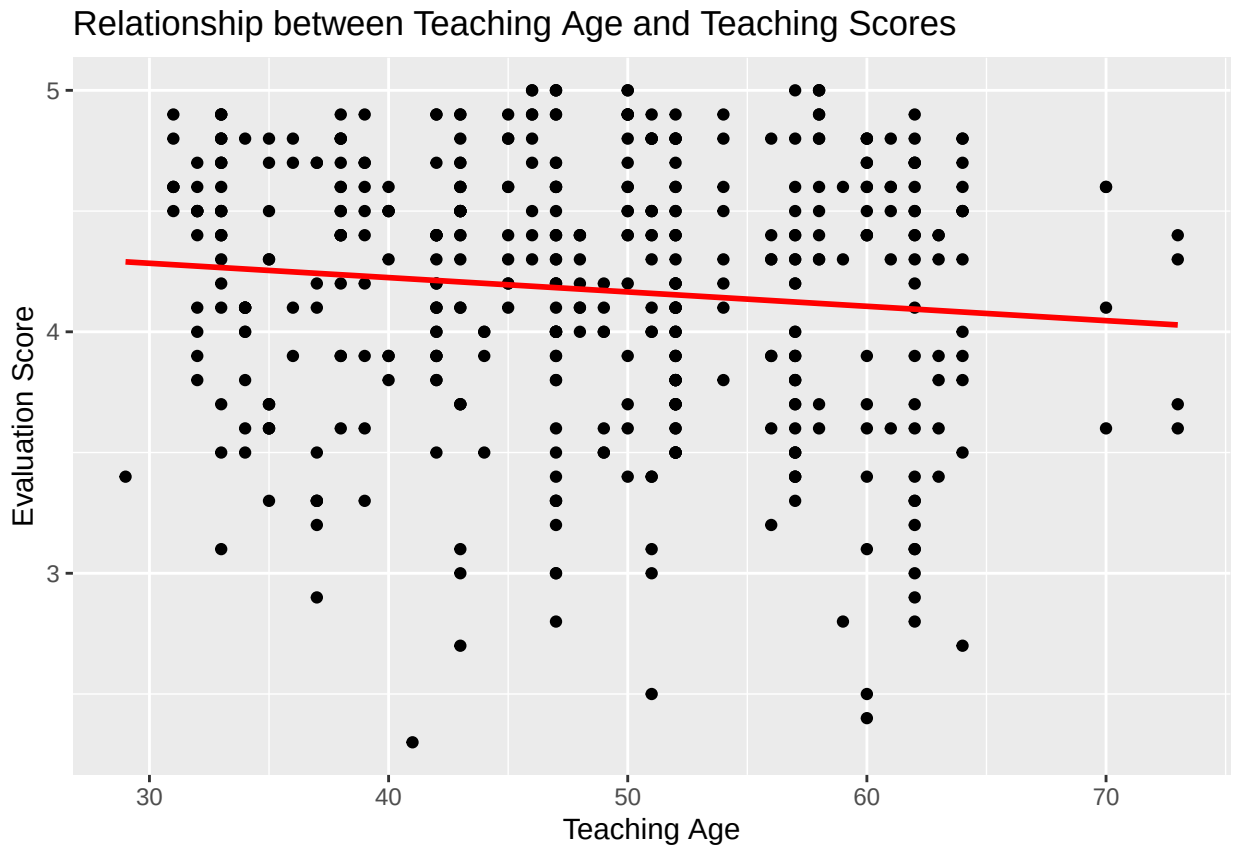


Answer: The jittered plot now displays the observations that were previously overplotted when we used *geom_point()*. This is specially evident when we compare both the *geom_point()* and geom_jittered()_ for observations with teaching score of 5 or slightly above it and the teaching age between 35 and 60.

"Best-Fitting Line"

```
ggplot(evals_ch6, aes(x = age, y = score))+
  geom_point()+
  labs( x= "Teaching Age", y= "Evaluation Score", title="Relationship between Teaching Age and Teaching
  geom_smooth( method = "lm",color= "red", se= FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



What can you say about the relationship between age and teaching scores based on this exploration?

Answer: So far in our analysis, after applying the "regression line" to our model, we note that the relationship between the outcome variable *score* and the explanatory variable *age* is linear and weak. The negative slope of the red line corroborates with the previously analyzed correlation coefficient of -0.107032. This suggests that there is a negative relationship between these two variables: as instructors' age goes up, the teaching score goes down.

(LC5.2) Fit a new simple linear regression using lm(score ~ age, data = evals_ch6) where age is the new explanatory variable x. Get information about the "best-fitting" line from the regression table by applying the get_regression_table() function. How do the regression results match up with the results from your earlier exploratory data analysis?

#Regression Model Building

Let's perform this in steps:

Step 1: Linear Regression using *lm()* function and generate the regression table using *get_regression_table()* function.

```
score_linear_model<- lm(score ~ age, evals_ch6)
```

Let's generate the regression table:

```
get_regression_table(score_linear_model)
```

```
## # A tibble: 2 x 7
##   term      estimate std_error statistic p_value lower_ci upper_ci
##   <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept    4.46      0.127     35.2    0        4.21     4.71
## 2 age         -0.006     0.003     -2.31   0.021   -0.011   -0.001
```

Step 2: Analysis

Equation of the regression line Based on the output of the regression table, and using the linear regression equation formula, we have the following equation:

y-hat = b0 + b1 * X

y-hat = 4.462 -0.006 * x

Although the intercept b0 = 4.462 is worth of our analysis, the one factor we are interested in right now is the b1 = -0.006 or the slope. First, the intercept is the average teaching score (y-hat) for those courses where the instructor had an age = 0. Graphically, this is where the line intersects the y axis when x = 0. However, this is unrealistic since one cannot have a teacher with an age =0. This leads us to our second point: the slope. The equation of the regression line shows that the slope is negative (-0.006). This implies that there is a negative relationship between teaching evaluation score and age. This suggests that on average, teachers with higher age tend to have lower teaching scores. This inverse or negative linear relationship between teaching age and score was also implied in the correlation coefficient we calculated earlier which also has a negative value of -0.107.

(LC5.3) Generate a data frame of the residuals of the model where you used age as the explanatory x variable.

Step 1: Generate the Residual table using *get_regression_points()* function.

```
score_age_model_residuals <- get_regression_points(score_linear_model)
score_age_model_residuals
```

```
## # A tibble: 463 x 5
##       ID score   age score_hat residual
##    <int> <dbl> <int>     <dbl>    <dbl>
##  1     1   4.7    36      4.25    0.452
##  2     2   4.1    36      4.25   -0.148
##  3     3   3.9    36      4.25   -0.348
##  4     4   4.8    36      4.25    0.552
##  5     5   4.6    59      4.11    0.488
##  6     6   4.3    59      4.11    0.188
##  7     7   2.8    59      4.11   -1.31
##  8     8   4.1    51      4.16   -0.059
##  9     9   3.4    51      4.16   -0.759
## 10    10   4.5    40      4.22    0.276
## # ... with 453 more rows
```

Step 2 Analysis: Let's perform further calculations of y-hat and y to calculate the residual for course ID 5 (age 59) and course ID 10 (age 40), respectively.

Our regression equation is: Y = b0 + b1 . x

Y= 4.462 - 0.006 . X

Lets calculate y-hat for course ID 5 and 10, that is when age is 59 and 40, respectively.

```
y_hat_ID5 <- c(4.462 - 0.006 * 59)
y_hat_ID5 =round( y_hat_ID5, digits= 2)
y_hat_ID5
```

```
## [1] 4.11
```

Our calculation of the y_hat for the course ID 5 is 4.11 which is the same as the one generated by the *get_regression_points()* function.

```
y_hat_ID10 <- 4.462 - 0.006 * 40
y_hat_ID10 = round (y_hat_ID10, digits= 2)

y_hat_ID10
```

```
## [1] 4.22
```

Also, our calculations of y_hat for course ID 10 gives us the same result as the one by the *get_regression_points()* function, which is 4.22

Calculating the residuals for course ID 5 and 10

The observed scores for courses ID 5 and 10 are 4.6 and 4.5, respectively

Improve the print output here.

```
residual_ID5 <- 4.6 - y_hat_ID5
residual_ID5
```

```
## [1] 0.49
```

```
residual_ID10 <- 4.5 -y_hat_ID10
residual_ID10
```

```
## [1] 0.28
```

The residual for course ID is 0.49 while the residual for course ID 10 is 0.28, respectively as also indicated by the output from the *get_regression_points()* function.

Answer In this two observations alone, our model had an error or "lack of fit" of 0.49 and 0.28 points for course ID 5 and 10 respectively.

Thank You!!!

Bernardo Vimpi