Intro to Data Science HW 8


```
# Enter your name here: Bhavya Shah
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

The chapter on **linear models** (Lining Up Our Models) introduces **linear predictive modeling** using the tool known as **multiple regression**. The term multiple regression has an odd history, dating back to an early scientific observation of a phenomenon called ** regression to the mean. ** These days, multiple regression is just an interesting name for using **linear modeling** to assess the **connection between one or more predictor variables and an outcome variable**.

In this exercise, you will **predict food insecurity from three predictors**.

A. We will be using the **Food Insecurity** data set from HW7. Copy it from this URL:

https://data-science-intro.s3.us-east-2.amazonaws.com/FoodInsecurity.csv (https://data-science-intro.s3.us-east-2.amazonaws.com/FoodInsecurity.csv)

into a dataframe called **df** and use the appropriate functions to **summarize the data**.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.1      ✓ purrr   1.0.1
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.3.0      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 1.0.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
df<-data.frame (read_csv('https://data-science-intro.s3.us-east-2.amazonaws.com/FoodInsecurity.c
sv',show_col_types = FALSE))
str(df)
```

```
## 'data.frame':    3142 obs. of  9 variables:
##  $ State            : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ County           : chr  "Autauga County" "Baldwin County" "Barbour County" "Bibb County"
...
##  $ Pop2010          : num  54571 182265 27457 22915 57322 ...
##  $ LAPOP1_10        : num  18503 45789 5634 365 3902 ...
##  $ AveragePovertyRate: chr  "16.13078591" "11.84554563" "29.29932484" "12.19352439" ...
##  $ MedianFamilyIncome: chr  "69337.5" "72665.74194" "44792.44444" "60645.5" ...
##  $ Largest_city     : chr  "Prattville" "Daphne" "Eufaula" "Brent" ...
##  $ city_state       : chr  "Prattville, Alabama" "Daphne, Alabama" "Eufaula, Alabama" "Bren
t, Alabama" ...
##  $ abbr             : chr  "AL" "AL" "AL" "AL" ...
```

B. In the analysis that follows, **LAPOP1_10** will be considered as the **outcome variable**, and **Pop2010**, **AveragePovertyRate**, and **MedianFamilyIncome** as the **predictors**. Add a comment to briefly explain the outcome variable (take a look at HW 7 if needed).

```
# The outcome variable LAPOP1_10 shows how many people in each county in the United States live
more than one mile (for urban areas) or ten miles (for rural areas) from a supermarket.
```

C. Inspect the outcome and predictor variables are there any missing values? Show the code you used to check for that.

```
sum(is.na(df$LAPOP1_10))
```

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

```
sum(is.na(df$Pop2010))
```

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

```
sum(is.na(df$AveragePovertyRate))
```

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

```
sum(is.na(df$MedianFamilyIncome))
```

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

```
# as of now there are no missing values in the variable
```

D. What does it mean when the output of the is.na() function is empty? Explain in a comment. Are all predictors coded as numerical variables? Show your code to check for that and if they are not - find a way to fix this issue, re-check for missing values, and implement a strategy to deal with them if present (Hint - **imputeTS** might help).

```
# there are no missing values in the columns if the output of is.na() function is empty.
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
str(df)
```

```
## 'data.frame':    3142 obs. of  9 variables:
##  $ State            : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ County           : chr  "Autauga County" "Baldwin County" "Barbour County" "Bibb County"
...
##  $ Pop2010          : num  54571 182265 27457 22915 57322 ...
##  $ LAPOP1_10        : num  18503 45789 5634 365 3902 ...
##  $ AveragePovertyRate: chr  "16.13078591" "11.84554563" "29.29932484" "12.19352439" ...
##  $ MedianFamilyIncome: chr  "69337.5" "72665.74194" "44792.44444" "60645.5" ...
##  $ Largest_city     : chr  "Prattville" "Daphne" "Eufaula" "Brent" ...
##  $ city_state       : chr  "Prattville, Alabama" "Daphne, Alabama" "Eufaula, Alabama" "Bren
t, Alabama" ...
##  $ abbr             : chr  "AL" "AL" "AL" "AL" ...
```

```
#We can see from this command that all predictors are not coded as numerical variables. As a res
ult, we must convert the AveragePovertyRate and MedianFamilyIncome into numeric variables.
```

```
df$AveragePovertyRate<-as.numeric(df$AveragePovertyRate)
```

```
## Warning: NAs introduced by coercion
```

```
df$MedianFamilyIncome<-as.numeric(df$MedianFamilyIncome)
```

```
## Warning: NAs introduced by coercion
```

```
sum(is.na(df$LAPOP1_10))
```

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

```
sum(is.na(df$Pop2010))
```

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

```
sum(is.na(df$AveragePovertyRate))
```

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

```
sum(is.na(df$MedianFamilyIncome))
```

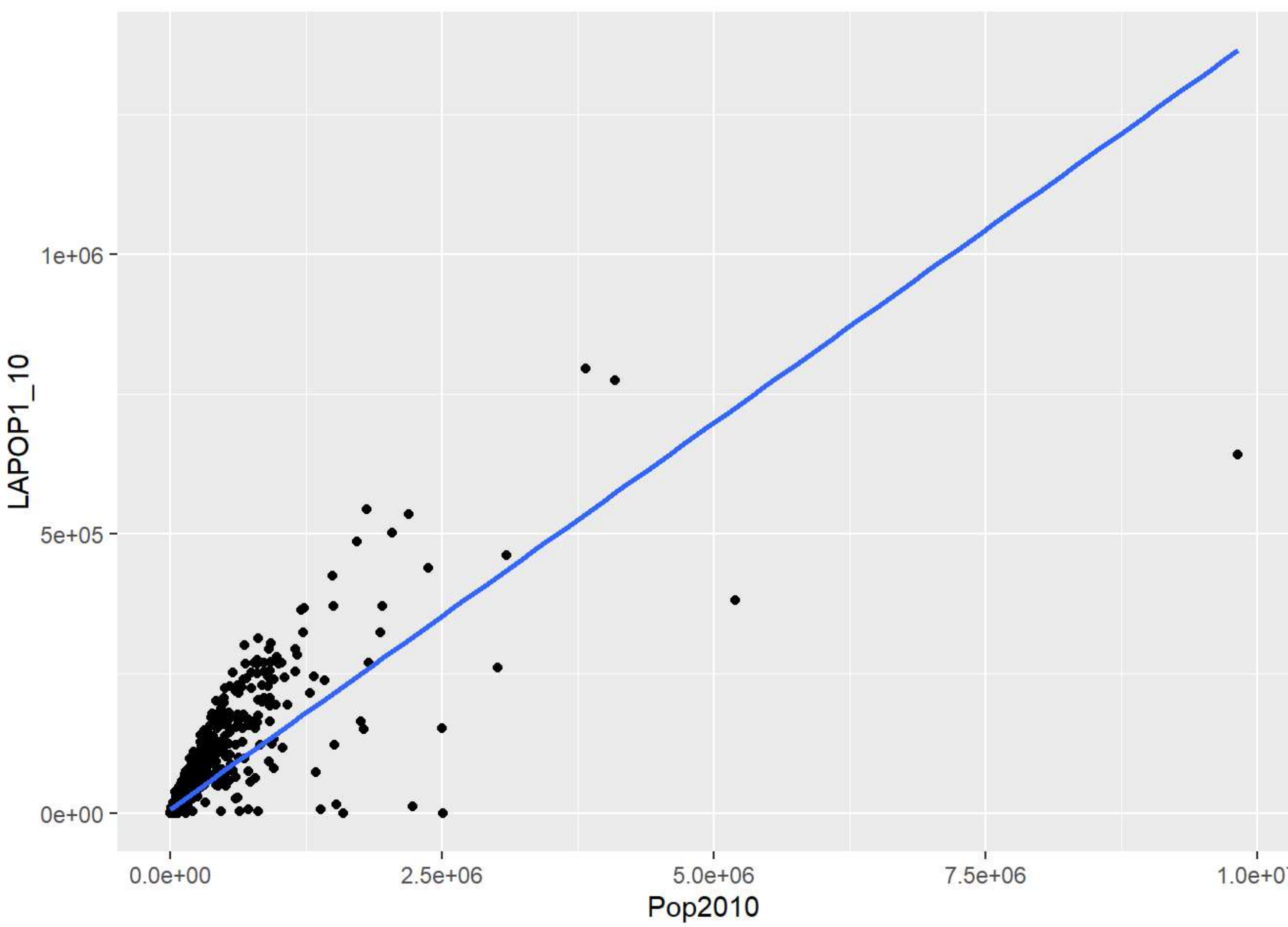
```
## [1] 2
```

```
df$AveragePovertyRate<-na_interpolation(df$AveragePovertyRate)
df$MedianFamilyIncome<-na_interpolation(df$MedianFamilyIncome)
# using na_interpolation the missing values have been replaced
```

E. Create **3 bivariate scatterplots (X-Y) plots** (using ggplot), for each of the predictors with the outcome. **Hint:** In each case, put **LAPOP1_10 on the Y-axis**, and a **predictor on the X-axis**. Add a comment to each, describing the plot and explaining whether there appears to be a **linear relationship** between the outcome variable and the respective predictor.

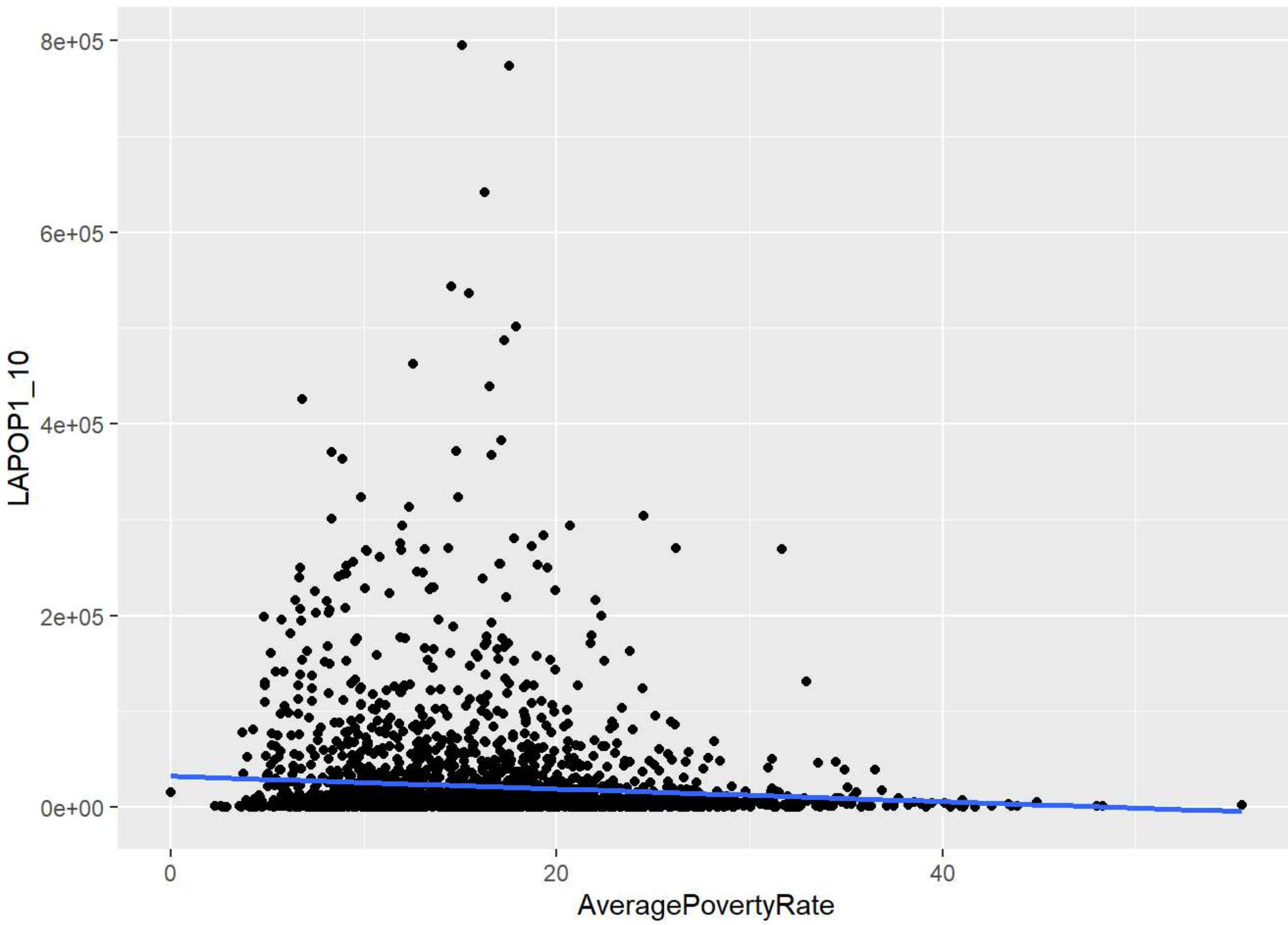
```
ggplot(data=df) + aes(x=Pop2010, y=LAPOP1_10) + geom_point() + geom_smooth(method="lm", se=FALS
E)
```

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



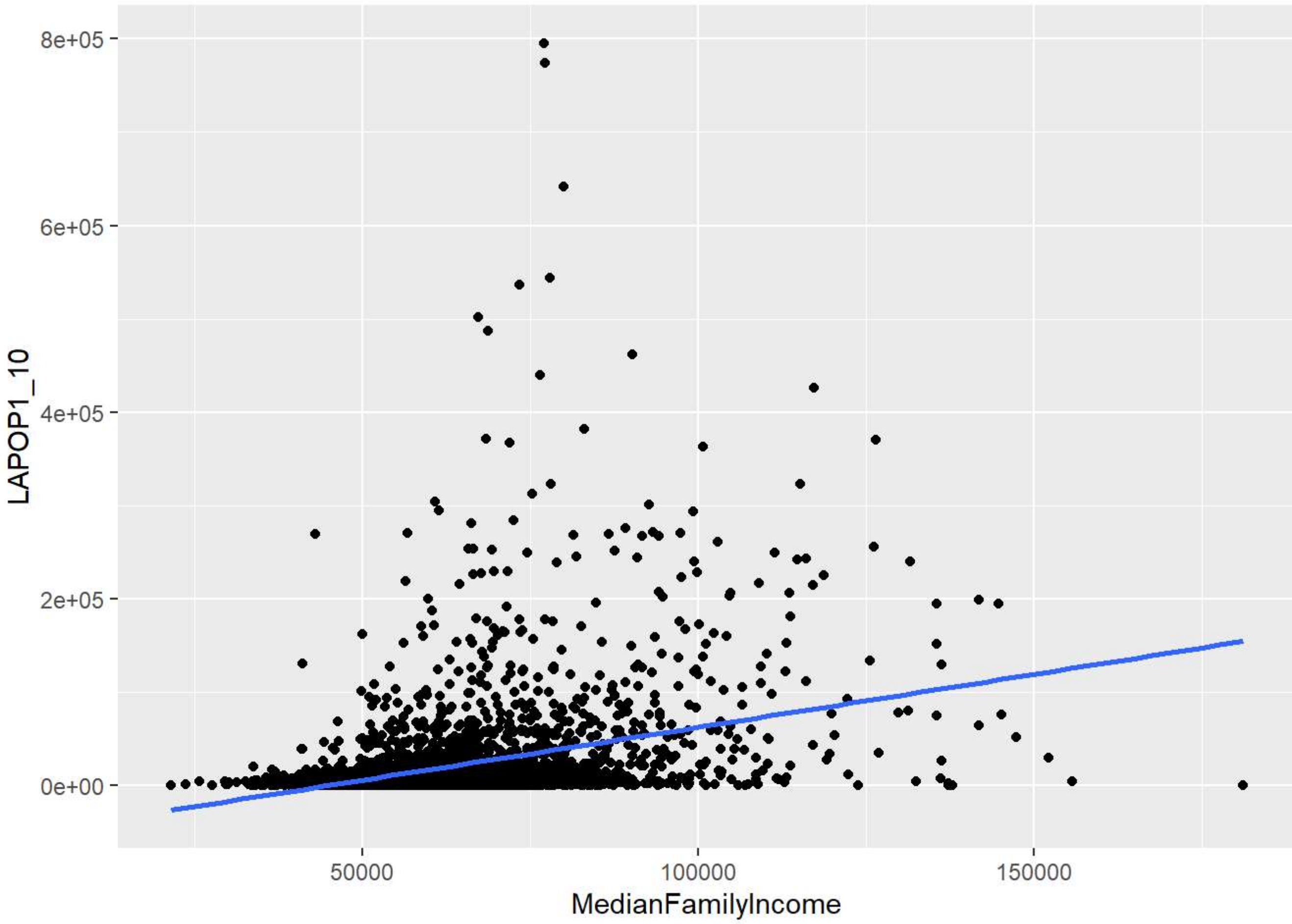
```
# The relationship for pop2010 predictor is linear, with the majority of points in the plot clus
tered together
ggplot(data=df) + aes(x=AveragePovertyRate, y=LAPOP1_10) + geom_point() + geom_smooth(method="l
m", se=FALSE)
```

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




```
# The relationship for average poverty predictor is nearly linear, with some points deviating fr
om the linear line.
ggplot(data=df) + aes(x=MedianFamilyIncome, y=LAPOP1_10) + geom_point() + geom_smooth(method="l
m", se=FALSE)
```

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




```
# for median family income the relationship is somewhat linear with few points away from the lin
ear line
```

F. Next, create a **simple regression model** predicting **LAPOP1_10 based on Pop2010**, using the **lm( )** command. In a comment, report the **coefficient** (aka **slope** or **beta weight**) of **Pop2010** in the regression output and, **if it is statistically significant**, **interpret it** with respect to **LAPOP1_10**. Report the **adjusted R-squared** of the model and try to explain what it means.

```
lol<-lm(LAPOP1_10 ~ Pop2010,data=df)
summary(lol)
```

```
##
## Call:
## lm(formula = LAPOP1_10 ~ Pop2010, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -723574   -8585   -6893   -1977  285519
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.260e+03  6.015e+02   13.73   <2e-16 ***
## Pop2010     1.382e-01  1.834e-03   75.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32160 on 3140 degrees of freedom
## Multiple R-squared:  0.6439, Adjusted R-squared:  0.6437
## F-statistic:  5677 on 1 and 3140 DF,  p-value: < 2.2e-16
```

```
# In the regression output, the slope of Pop2010 is 1.32e-01 and the y intercept is 8.260e+03. B
ecause the predictor pop2010 belongs to the significant code 0, it is statistically significant.
If LAPOP1 10 increases by one, pop2010 rises by 0.1382. The predictor's adjusted r square is 0.6
4, indicating that the model is 64% accurate.
```

G. Create a **multiple regression model** predicting **LAPOP1_10** based on **Pop2010**, **AveragePovertyRate**, and **MedianFamilyIncome**.
Make sure to include all three predictors in one model NOT three different models each with one predictor.

```
lolll<-lm(LAPOP1_10 ~ Pop2010 + AveragePovertyRate + MedianFamilyIncome,data=df)
summary(lolll)
```

```
##
## Call:
## lm(formula = LAPOP1_10 ~ Pop2010 + AveragePovertyRate + MedianFamilyIncome,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -653442   -8161   -4113     733  291793
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -4.299e+04  4.889e+03  -8.792  < 2e-16 ***
## Pop2010             1.299e-01  1.902e-03  68.276  < 2e-16 ***
## AveragePovertyRate  7.083e+02  1.230e+02   5.758 9.34e-09 ***
## MedianFamilyIncome  6.381e-01  5.196e-02  12.280  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31340 on 3138 degrees of freedom
## Multiple R-squared:  0.6621, Adjusted R-squared:  0.6618
## F-statistic:  2049 on 3 and 3138 DF,  p-value: < 2.2e-16
```

H. Report the **adjusted R-Squared** in a comment. How does it compare to the adjusted R-squared from Step F? Is this better or worse? Which of the predictors are **statistically significant** in the model? In a comment, report the coefficient of each predictor that is statistically significant. Do not report the coefficients for predictors that are not significant.

```
# the adjusted R-squared value is 0.6618 which represents the accuracy is 66% while in step F th
e adjusted r-squared value is 0.6437 which represents the accuracy of 64%. As the accuracy of th
e one that we calculated in step G is higher we can say that this a best predictor. The s ignifi
cant predictors are Pop2010, AveragePovertyRate, MedianFamilyIncome because P-value is less than
0.05. The coefficient of significant predictors are as follows
#1.For Predictor Pop2010:
# coefficients: Estimate 1.299e-01, standard error=1.902e-03, t value=68.276, prediction_value=
< 2.2e-16
#2.For Predictor AveragePovertyRate:
#coefficients: Estimate 7.083e+02, standard error=1.230e+02, t value=5.758, prediction_value= 93
4e-09
#3. For Predictor Median Family Income:
#coefficients: Estimate 6.381e-01, standard error-5.196e-02, t value=12.280, prediction_value= <
2.2e-16`
```

I. Create a one-row data frame like this:

```
predDF <- data.frame(Pop2010=100000, AveragePovertyRate=20, MedianFamilyIncome=65000)
```

and use it with the **predict( )** function to predict the **expected value of LAPOP1_10**:

```
predict(lolll,predDF)
```

```
##        1
## 25640.91
```

Describe the accuracy of the prediction.

```
# the prediction accuracy is 66.18%
```

J. Create an additional **multiple regression model**, with **AveragePovertyRate** as the **outcome variable**, and the other **3 variables** as the **predictors**.

Review the quality of the model by commenting on its **adjusted R-Squared**.

```
lmagg<-lm(AveragePovertyRate ~ Pop2010 + LAPOP1_10 + MedianFamilyIncome,data=df)
summary(lmagg)
```

```
##
## Call:
## lm(formula = AveragePovertyRate ~ Pop2010 + LAPOP1_10 + MedianFamilyIncome,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.2831  -2.8680  -0.6914   2.0332  27.9171
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.512e+01  3.425e-01 102.552  < 2e-16 ***
## Pop2010             1.359e-06  4.322e-07   3.144  0.00168 **
## LAPOP1_10           1.476e-05  2.563e-06   5.758 9.34e-09 ***
## MedianFamilyIncome -3.080e-04  5.360e-06 -57.461  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.524 on 3138 degrees of freedom
## Multiple R-squared:  0.5164, Adjusted R-squared:  0.5159
## F-statistic:  1117 on 3 and 3138 DF,  p-value: < 2.2e-16
```

```
# the adjusted r square is 0.5159 which means the accuracy is 51.59% which is comparatively lowe
r.
```