AMS 315 Project 1

Zian Shang

Instructor: Benjamin Hechtman

11/3/2022

Part A

The problem this project is trying to address is how to recover the function used to generate the dependent variable value based on the independent variable. This part of the project will analyze data arranged by subject ID and impute missing values in the dataset. Students will learn how to solve data and statistics processing problems as statisticians.

For this project, I downloaded R as my software environment and RStudio as the integrated development environment. I created a new repository and verified it with “getwd()”. I used the “read.csv()” command to read both IV and DV files and assign them to two variables: PartA_IV and PartA_DV. I then merged them together by ID to get a new “PartA” using the “merge()” function. With “str(PartA)” and “View(PartA)”, I was able to review the merged data. After that, I used the “any(is.na/nan()==TRUE)” function to check if there is any missing data marked as NA or NAN in the columns. After knowing there was some NAs, I decided to apply the norm.boot method to analyze them. I imported the library mice and used its function “md.pattern()” to find the pattern of missing data. The program showed me a table that counts the number of data of each type. Then, to impute the missing data, I first dropped the 19 observations missing both IV and DV using PartA[!is.na(PartA$IV)==TRUE|!is.na(PartA$DV)==TRUE,] command in order to apply the “linear regression using bootstrap” method. I put “norm.boot” as the method of “mice()” and applied the function “complete()” to the result of it. Then, “md.pattern()” showed my dataset was completed. I also fitted a linear regression model to the dataset with the function “lm()”. The “summary()” function summarized the residuals, coefficients, and some other statistics. After that, I used the function “kable()” in “knitr” library to generate an ANOVA table, “plot()” to generate a scatter plot with a line of best fit, and “confint()” to find the confidence intervals of 95% and 99%. (apdx 1)

According to the result of R, there are a total of 544 observations of 3 variables, with 475 complete datasets. IV is missing in 46 cases, DV is missing in 42 cases, and both are missing in 19 cases (apdx 3). This implies that there are 525 cases in which at least one independent variable or dependent variable exists (apdx 4), 502 cases with a dependent variable, and 498 cases with an independent variable. By summarizing the linear regression model after dropping useless data, I got an estimated intercept of 32.2783 and a slope of 3.8338. This means the equation is Y=32.2783+3.8338x (apdx 5). The ANOVA table of this equation shows an F-value of 387.3644 for the slope of IV, which is far greater than the values on the reference table when the alpha is 0.10, 0.05, and 0.01 (apdx 6). Thus, for the test of H0: slope=0, H0 will be rejected. The 95% confidence interval for the intercept is (30.237190, 34.319389) and for the slope of IV is (3.451149, 4.216492). This means that we can be 95% confident that the true mean of the dataset of either the intercept or the IV is within that range. The same is true for the 99% confidence interval. It is (29.592237,34.96434) for the intercept and (3.330231, 4.33741) for the slope of IV (apdx 8).

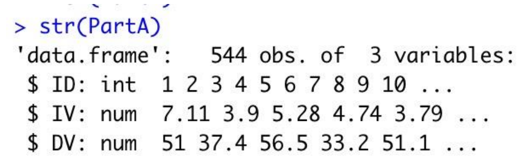
In conclusion, there is an association between the independent variable and the dependent variable. There are two R-squared values given by R. The multiple R-squared equals 0.4255. The adjusted R-squared equals 0.4244. In this project, we can use either one of them since there are no significant differences as we are dealing with a simple linear regression model. My fitted function for this model is Y=32.2783+3.8338x.

Appendix A

1. **Source code:**

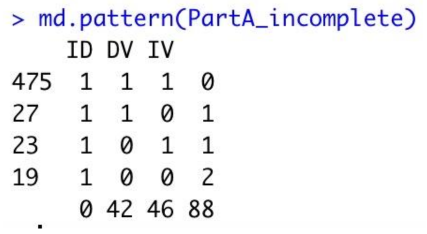
```
getwd()
# read data files and merge by ID
PartA_IV <- read.csv('/Users/zianshang/Downloads/342000_IV.csv', header=TRUE)
PartA_DV <- read.csv('/Users/zianshang/Downloads/342000_DV.csv', header=TRUE)
PartA <- merge(PartA_IV, PartA_DV, by = 'ID')
View(PartA)
str(PartA)
View(PartA_DV)
View(PartA_IV)
# check if there is any NA or NAN in IV and DV
any(is.na(PartA[,2]) == TRUE)
any(is.na(PartA[,3]) == TRUE)
any(is.nan(PartA[,2]) == TRUE)
any(is.nan(PartA[,3]) == TRUE)
PartA_incomplete <- PartA
install.packages('mice')
library(mice)
md.pattern(PartA_incomplete)
# there are 475 complete data sets
# IV is missing in 46 cases, DV is missing in 42 cases, both are missing in 19 cases
# impute data, drop 19 cases that are missing both IV and DV
PartA_imp <- PartA[!is.na(PartA$IV)==TRUE|!is.na(PartA$DV)==TRUE,]
imp <- mice(PartA_imp, method = "norm.boot", printFlag = FALSE)
View(imp)
View(PartA_imp)
PartA_complete <- complete(imp)
md.pattern(PartA_complete)
#fit a regression model to the data set and save it to an object
M <- lm(DV ~ IV, data=PartA_complete)
View(M)
summary(M)
#draw ANOVA table
install.packages('knitr')
library(knitr)
kable(anova(M), caption='ANOVA Table')
# scatter plot
plot(PartA_complete$DV ~ PartA_complete$IV, main='Scatter : DV ~ IV', xlab='IV', ylab='DV', pch=20)
abline(M, col='red', lty=3, lwd=2)
legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
# calculate confidence interval for the slope
confint(M, level = 0.95)
confint(M, level = 0.99)
```

2. **str(PartA)**

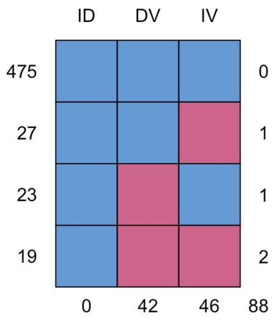


```
> str(PartA)
'data.frame':	544 obs. of  3 variables:
 $ ID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ IV: num  7.11 3.9 5.28 4.74 3.79 ...
 $ DV: num  51 37.4 56.5 33.2 51.1 ...
```

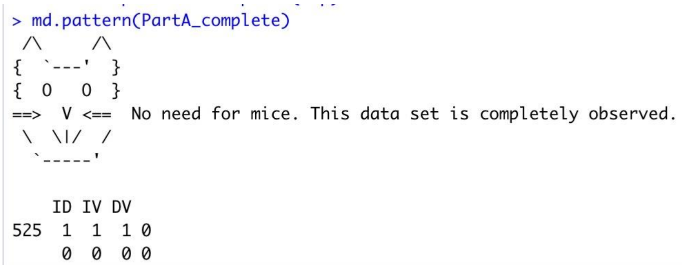
3. missing data pattern of the original merged dataset



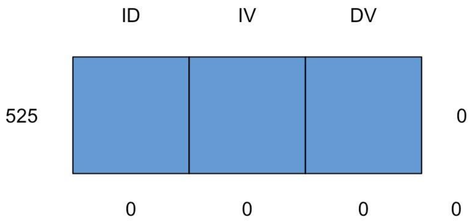
```
> md.pattern(PartA_incomplete)
    ID DV IV
475  1  1  1  0
27   1  1  0  1
23   1  0  1  1
19   1  0  0  2
     0 42 46 88
```



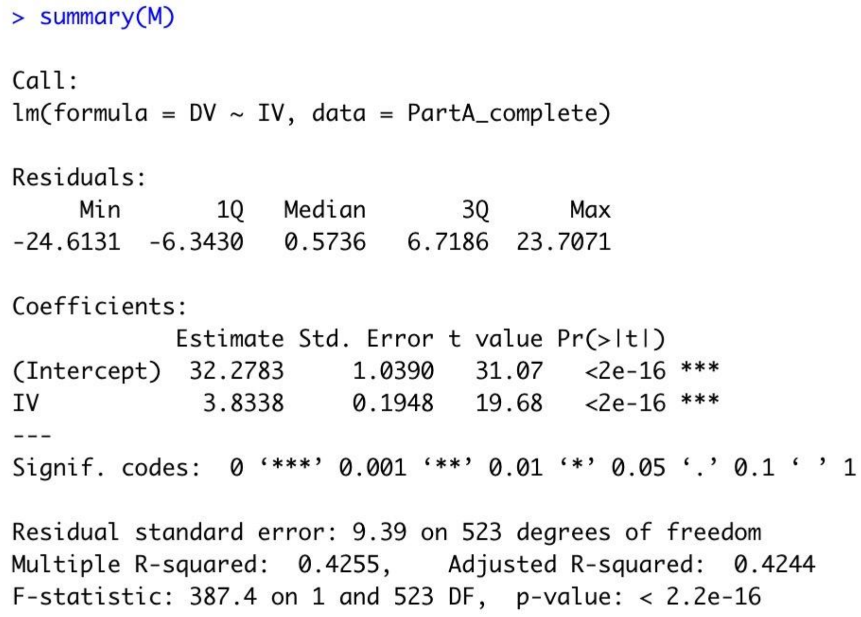
4. missing data pattern of the completed dataset after imputation



```
> md.pattern(PartA_complete)
 /\     /\
{  `---'  }
{  O   O  }
==>  V <==  No need for mice. This data set is completely observed.
 \  \|/  /
  `-----'

    ID IV DV
525  1  1  1 0
     0  0  0 0
```

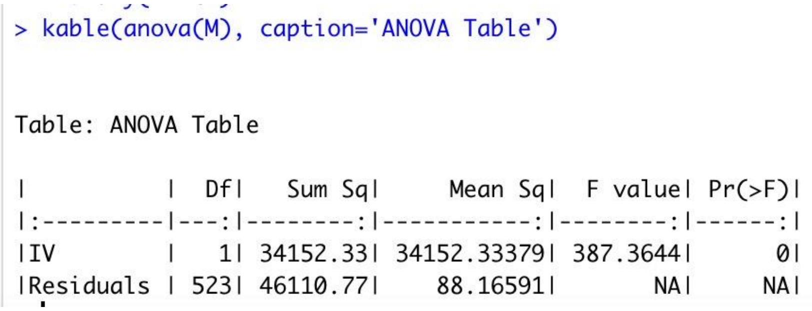


5. summary of the linear regression model M



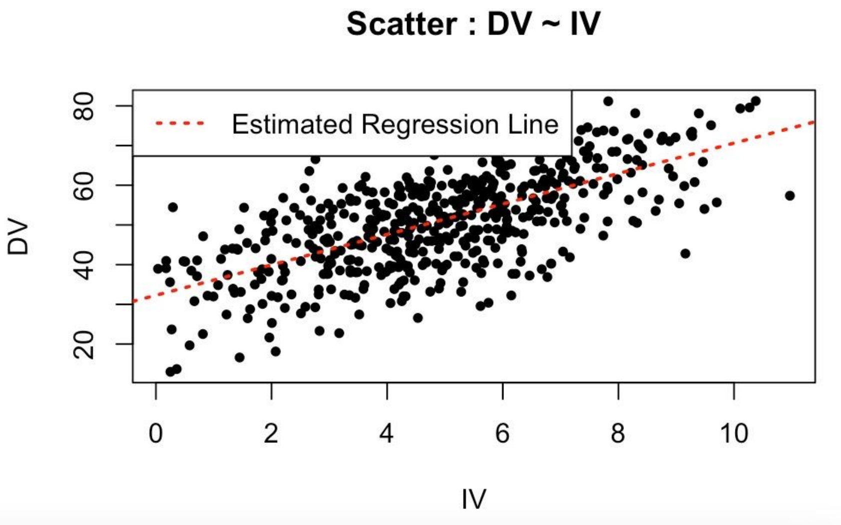
```
> summary(M)

Call:
lm(formula = DV ~ IV, data = PartA_complete)

Residuals:
     Min       1Q   Median       3Q      Max 
-24.6131  -6.3430   0.5736   6.7186  23.7071 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  32.2783     1.0390   31.07   <2e-16 ***
IV            3.8338     0.1948   19.68   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.39 on 523 degrees of freedom
Multiple R-squared:  0.4255,	Adjusted R-squared:  0.4244 
F-statistic: 387.4 on 1 and 523 DF,  p-value: < 2.2e-16
```

6. **ANOVA table**

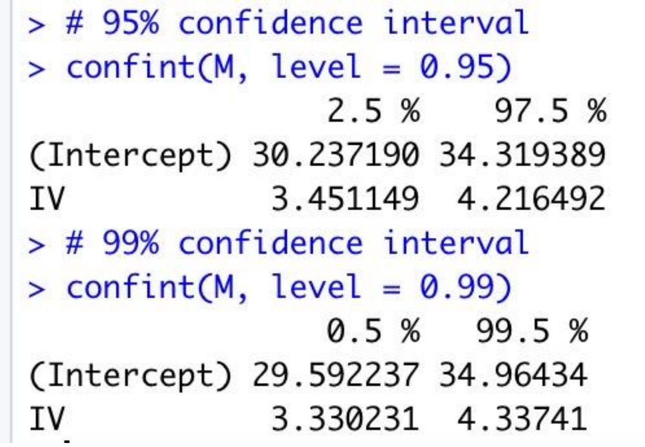


```
> kable(anova(M), caption='ANOVA Table')


Table: ANOVA Table

|          |  Df|   Sum Sq|     Mean Sq|  F value| Pr(>F)|
|:---------|---:|--------:|-----------:|--------:|------:|
|IV        |   1| 34152.33| 34152.33379| 387.3644|      0|
|Residuals | 523| 46110.77|    88.16591|       NA|     NA|
```

7. **scatter plot of the linear regression model**



8. **95% and 99% confidence intervals**



```
> # 95% confidence interval
> confint(M, level = 0.95)
                2.5 %    97.5 %
(Intercept) 30.237190 34.319389
IV           3.451149  4.216492
> # 99% confidence interval
> confint(M, level = 0.99)
                0.5 %   99.5 %
(Intercept) 29.592237 34.96434
IV           3.330231  4.33741
```

Part B

During part B of this project, we will first find a transformation for the given data and fit a linear regression model to it. Then, we need to find repeated measures of independence variable values and bin them to apply an approximate lack of fit test to the transformed data. We will also be using R for this part.

For part B, I first loaded the sample data file into the variable “data” using the function “read.csv()”. The function “str(data)” showed that there are 584 observations of 3 variables for this dataset(apdx 2). Then, I created a linear regression model “original_lm” with the function “lm()”. I created a scatter plot using the function “plot()” (apdx 5) and examined the R-squared data from the result of “summary()”. After that, I tried several transformations such as Y^(½), Y^(⅓), and Y^(⅔), and chose the one with the highest multiple R-squared value. It is Y^(½) with an R-squared value of 0.4401. I fitted a linear model and plotted this “trans_lm” as well(apdx 5). After doing this, I used “cut()” to create groups of similar x values and assigned them to the variable “groups”. I used the function “table()” to show the table of data after grouping. To find the average of the groups, I used the “ave()” function. The result of “ave()”, x, is then used to create “data_bin” using the function “data.frame()”. In the end, I applied the LOF test to the “data_bin”. I installed the package “olsrr” and used its function “ ols_pure_error_anova(fit_b)” to show the ANOVA table for the lack of fit test.

There are a total of 584 observations of 3 variables in the original dataset (apdx 2). The R-squared value of this original dataset’s linear regression model is 0.4245 (apdx 3). The equation is Y=(-1084.36)+962.36X. I tried several transformations, starting from Y^(1/2). It gave me a multiple R-square value of 0.4401. I then tried other transformations on Y. Y^(1/3) gave an R-squared value of 0.4332, and Y^(2/3) gave 0.4399. I wondered if changing the X value will improve the R-squared, so I tried X^(1/2) with Y^(1/2), which gave 0.4358. X^(2) with Y^(1/2) gave 0.4326, X^(-1) with Y^(1/2) gave 0.385, and X^(4) with Y^(1/2) gave 0.3871 (apdx 8). At this point, I decided my transformation should be Y^(1/2). Therefore, I fitted my transformed data into a linear regression model and plotted it. The new scatter plot appeared to be more uniformly distributed, which indicates it is more fitted (apdx 5). The transformed model had an equation of Y=39.4846+4.8104X (apdx 4). The sum of the squared value of X is 262821.60, of residual, is 334099.47, of lack of fit is 23703.94, and of pure error is 310395.53. The lack of fit test provided a p-value of 0.718097 (apdx 7).

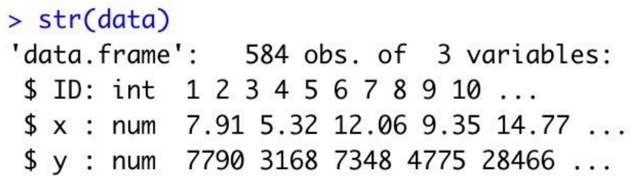
In conclusion, my p-value,0.718097, is relatively large, which indicates that, upon transforming the data, there is little significant lack of fit in the regression model. As a result, my selection of transformation of Y^(1/2) is a good option. The R-squared value supports my choice of transformation since it is the most significant improvement among all of my attempts. My scatter plot of the data after transformation supports my choice by displaying nearly an evenly-spread flat ellipse. My fitted function for the linear regression model of this new dataset is Y=39.4846+4.8104X.

Appendix

1. source code

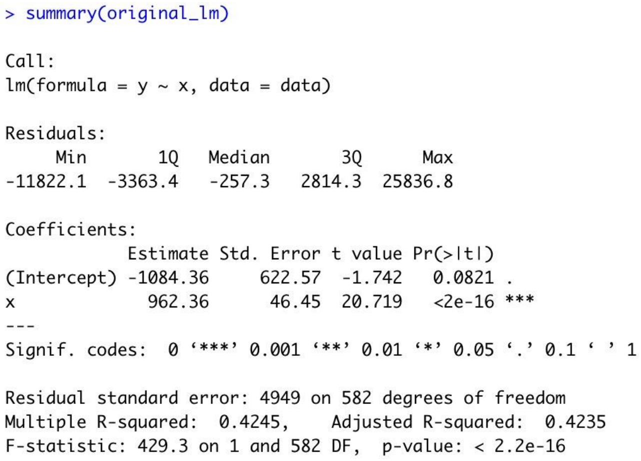
```
> getwd()
> data <- read.csv('/Users/zianshang/Downloads/342000_PartB.csv', header = TRUE)
> View(data)
> str(data)
> original_lm <- lm(y ~ x, data=data)
> View(original_lm)
> # create scatter plot of original lm
> plot(data$y ~ data$x, main='Scatter : y ~ x', xlab='x', ylab='y', pch=20)
> abline(original_lm, col='red', lty=3, lwd=2)
> legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
> summary(original_lm)
> # try trans y^(1/2), R-squared: 0.4401
> data_trans <- data.frame(xtrans=data$x, ytrans=data$y^(1/2))
> trans_lm <- lm(ytrans ~ xtrans, data = data_trans)
> summary(trans_lm)
> # try trans y^(1/3), R-squared: 0.4332
> #data_trans <- data.frame(xtrans=data$x, ytrans=data$y^(1/3))
> # try trans y^(2/3),  R-squared: 0.4399
> #data_trans <- data.frame(xtrans=data$x, ytrans=data$y^(2/3))
> #similar trans…
> # use trans y^(1/2), R-squared: 0.4401
> data_trans <- data.frame(xtrans=data$x, ytrans=data$y^(1/2))
> trans_lm <- lm(ytrans ~ xtrans, data = data_trans)
> summary(trans_lm)
> # create scatter plot for transformed lm
> plot(data_trans$y ~ data_trans$x, main='Scatter Transformed: y ~ x', xlab='x', ylab='y', pch=20)
> abline(trans_lm, col='red', lty=3, lwd=2)
> legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
># create groups using cut()
> groups <- cut(data_trans$xtrans,breaks=c(-Inf,seq(min(data_trans$xtrans)+0.3, max(data_trans$xtrans)-0.3,by=0.3),Inf))
> View(data_trans)
> table(groups)
> # find the average of each group and bind data
> x <- ave(data_trans$xtrans, groups)
> data_bin <- data.frame(x=x, y=data_trans$ytrans)
> View(data_bin)
> # apply LOF test to data_bin
> install.packages("olsrr")
> library("olsrr")
> fit_b <- lm(y ~ x, data = data_bin)
> ols_pure_error_anova(fit_b)
```

2. 584 observations of 3 variables for the original dataset



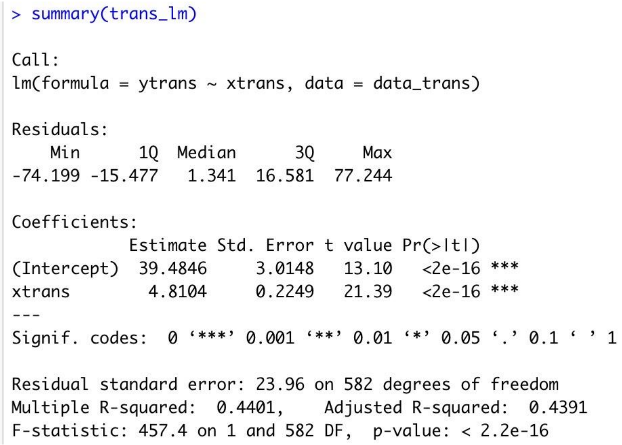
```
> str(data)
'data.frame':   584 obs. of  3 variables:
 $ ID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ x : num  7.91 5.32 12.06 9.35 14.77 ...
 $ y : num  7790 3168 7348 4775 28466 ...
```

3. original dataset summary, multiple R-squared=0.4245



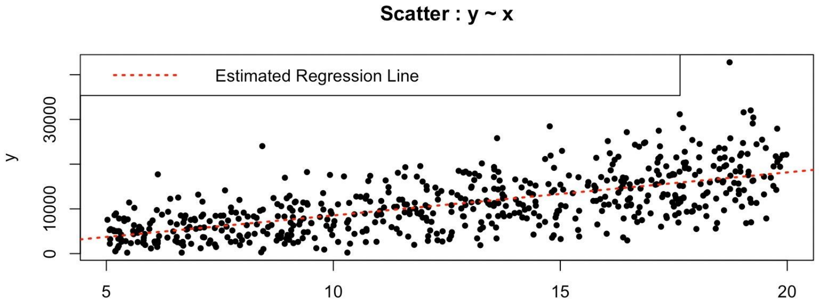
```
> summary(original_lm)

Call:
lm(formula = y ~ x, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-11822.1  -3363.4   -257.3   2814.3  25836.8 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1084.36     622.57  -1.742   0.0821 .  
x             962.36      46.45  20.719   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4949 on 582 degrees of freedom
Multiple R-squared:  0.4245,	Adjusted R-squared:  0.4235 
F-statistic: 429.3 on 1 and 582 DF,  p-value: < 2.2e-16
```

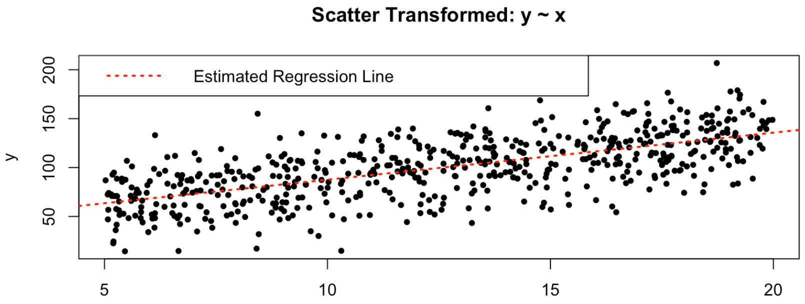
4. dataset summary after transformation, multiple R-squared=0.4401



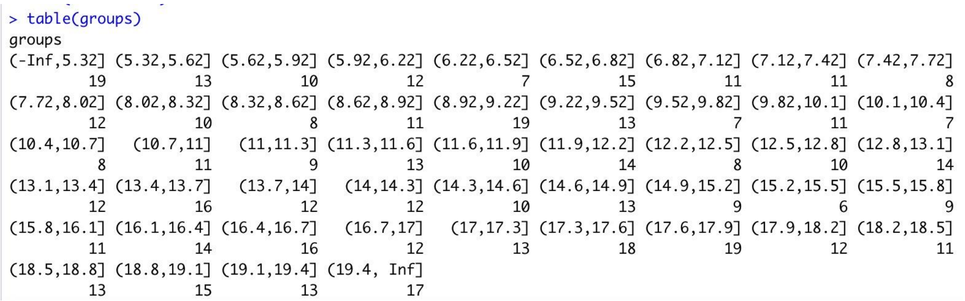
```
> summary(trans_lm)

Call:
lm(formula = ytrans ~ xtrans, data = data_trans)

Residuals:
    Min      1Q  Median      3Q     Max 
-74.199 -15.477   1.341  16.581  77.244 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  39.4846     3.0148   13.10   <2e-16 ***
xtrans        4.8104     0.2249   21.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.96 on 582 degrees of freedom
Multiple R-squared:  0.4401,	Adjusted R-squared:  0.4391 
F-statistic: 457.4 on 1 and 582 DF,  p-value: < 2.2e-16
```

5. scatter plots, original model vs. transformed model



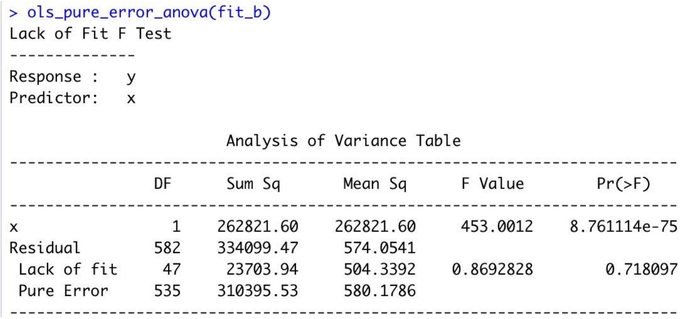


6. Table of transformed data after grouping



```
> table(groups)
groups
(-Inf,5.32] (5.32,5.62] (5.62,5.92] (5.92,6.22] (6.22,6.52] (6.52,6.82] (6.82,7.12] (7.12,7.42] (7.42,7.72] 
         19          13          10          12           7          15          11          11           8 
(7.72,8.02] (8.02,8.32] (8.32,8.62] (8.62,8.92] (8.92,9.22] (9.22,9.52] (9.52,9.82] (9.82,10.1] (10.1,10.4] 
         12          10           8          11          19          13           7          11           7 
(10.4,10.7]   (10.7,11]   (11,11.3] (11.3,11.6] (11.6,11.9] (11.9,12.2] (12.2,12.5] (12.5,12.8] (12.8,13.1] 
          8          11           9          13          10          14           8          10          14 
(13.1,13.4] (13.4,13.7]   (13.7,14]   (14,14.3] (14.3,14.6] (14.6,14.9] (14.9,15.2] (15.2,15.5] (15.5,15.8] 
         12          16          12          12          10          13           9           6           9 
(15.8,16.1] (16.1,16.4] (16.4,16.7]   (16.7,17]   (17,17.3] (17.3,17.6] (17.6,17.9] (17.9,18.2] (18.2,18.5] 
         11          14          16          12          13          18          19          12          11 
(18.5,18.8] (18.8,19.1] (19.1,19.4] (19.4, Inf] 
         13          15          13          17 
```

7. ANOVA table for Lack of Fit test



```
> ols_pure_error_anova(fit_b)
Lack of Fit F Test
--------------
Response :   y
Predictor:   x

                        Analysis of Variance Table
--------------------------------------------------------------------------
                DF      Sum Sq       Mean Sq      F Value        Pr(>F)
--------------------------------------------------------------------------
x                1    262821.60    262821.60     453.0012    8.761114e-75
Residual       582    334099.47     574.0541
 Lack of fit    47     23703.94     504.3392    0.8692828        0.718097
 Pure Error    535    310395.53     580.1786
--------------------------------------------------------------------------
```

8. R-squared values of several transformation attempts

- original_lm R-squared: 0.4245
- trans_Y(1/2) R-squared: 0.4401
- trans_Y(1/3) R-squared: 0.4332
- trans_Y(2/3) R-squared: 0.4399
- trans_X(1/2)_Y(1/2) R-squared: 0.4358
- trans_X(2)_Y(1/2) R-squared: 0.4326
- trans_X(-1)_Y(1/2) R-squared: 0.385
- trans_X(4)_Y(1/2) R-squared: 0.3871