Rahul Ranjan (ID – 111448179)

Professor Finch

AMS 315 Project 1: Part A Report

Introduction

The conflict of this project is to obtain the function that was used to produce the dependent variable values that depend on the independent variable values. This function will be derived by utilizing R as a programming platform for inputting and outputting important data. I was given two Excel files containing subject IDs and either only independent or dependent variable values for each ID. These files will be merged, and appropriate statistical functions will be used on this data to approach a conclusion. Along with the files, I was also given a useful internet source handout by my fellow TA's. This was very useful in helping me code, but there was one problem. Every time the two data files were merged and imputed, the data would be merged differently. (Example – ID # 237 had different IV values but the same DV value for two different times I merged the same data files) This difference resulted in different models for the same data. (Different R-squared values and function parameters) Despite this problem, I still decided to follow the handout given by the TA's since upon emailing them I was told that there would be no point reduction. The least squares method will be used to minimize the sum of squares coming from residuals and will ultimately assist in producing a linear model. Analysis of variance will be displayed as a table and its results along with confidence intervals and a lack of fit test will determine the accuracy of the linear model. The motive behind the completion of this project is the experience one can acquire. Learning about regression analysis is one thing but applying it will help me better understand how it is used in the real world.

Methodology

Firstly, the data from the two excel files were read in R by using the read.csv() function. The two files were read, and the data was inserted into two separate variables . Afterwards, the data was merged and automatically sorted by ID using the merge() function. The merged data was accessed using a new variable - PartA. This data was then imputed by the "linear regression using bootstrap" method through the installation and use of the mice() function. This method allows for the data to be sampled through replacement and eventually it will estimate the statistical information on the entire population of the data. Now, all data missing independent and dependent variable values have been removed from the merged data. I used the R command lm() to fit a regression model to my merged data set. From here, the ANOVA table, regression line, confidence intervals, and prediction intervals were constructed to provide conclusions of the model. The ANOVA was used to gather information about the variability levels for the regression model. Lastly, a lack of fit test based on the data from the ANOVA table will be generated to test the model.

Results

There were 715 observations in total. Out of this, it was determined that 696 subject ID's were associated with at least one independent or dependent variable value, 644 subject ID's were associated with independent variable values, 543 subject ID's were associated with dependent variable values, and 491 subject ID's were associated with independent and dependent variable values. There were 19 data sets with no independent or dependent variable values. After imputation there were a total of 696 observations. (As shown in the merged data diagram below) The regression model had an adjusted r-squared value of 0.5304, a multiple r-squared value of 0.5298, and a r value of 0.7283. These values were backed up by ANOVA. The model also gave the fitted function DV = 11.219 + 4.759(IV). The ANOVA table gave an F-value of 783.9868 and a p value of 0. Therefore, there Is a significant difference in the mean of both independent and dependent variable values. (Model p-value was 2.2 e^-16 which is basically 0) This p-value is very low so is extremely significant and as a result, the ANOVA favors this model. The division of the independent variable sum of squares by the total sum of squares generated a r-squared value of 0.530442394 which rounds off to 0.5304. This value is identical to the multiple r-squared value generated by the model. The multiple r-squared value was moderate/strong so a transformation of the variables would not be necessary. This r-squared value tells us that 53.04% of the dependent variable variance is explained by the independent variable. The r value of 0.7283 tells us that there exists a moderately strong relationship between the dependent and independent variables. The 95% confidence interval for the slope of my estimated regression line was 4.425696 5.093175. This p-value from the ANOVA table makes us reject the null hypothesis that the slope of the estimated regression equals 0. (Due to the 0.10, 0.05, and 0.01 significance levels being higher than the near 0 p-value) Since the slope was non-zero and the confidence interval does not contain 0, we can conclude that the independent and dependent variables have a relation and the regression model can be a useful predictor. The lack of fit test generated a lack of fit F value of 0.9627 and a lack of fit p-value of 0.5306. The p-value is higher than the significance levels of 0.10, 0.05, and 0.01. Therefore, we fail to reject the null hypothesis that the regression model is appropriate.

Conclusion

The reasonably high r value shows that there exists a strong relationship between the two variables. The moderate r-squared value shows us that more than half of the variance from the dependent variable can be explained by the independent variable. Therefore, there is a moderate/strong association between the independent and dependent variables. As shown in the results, the confidence interval for the slope, and test of the slope equaling 0 also assisted me. These helped to show that the slope had to be non-zero and there must co-exist a relationship between the variables. Lastly, the lack of fit test determines the reliability of the model and its function. The results showed that we failed to reject the model. According to all of this, I can conclude that the fitted function DV = 11.219 + 4.759(IV) from my model was used to generate the dependent variable values from the independent variable and I accept this model. Before ending this report, I would like to state, if I were a statistician, I would have reported the problem mentioned above to a higher up before starting this part of the project. After the completion of this part of the project, I found out that instead of merging the two data files (IV and DV) first and then imputing, you can do the opposite. I imputed both data files separately first and then merged them together. I repeated this step multiple times and the code is shown in the appendix below. For every repetition, the model ended up being the same with the same R-squared values and function parameters. (Unlike the

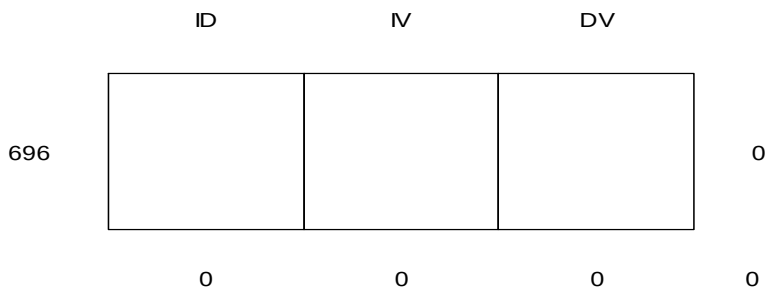
models generated from the same data when using the handout procedure) Despite this problem, I still accept the model from above which I obtained through the handout's procedure.

Appendix – Tables, Graphs, R Code and Other Important Data

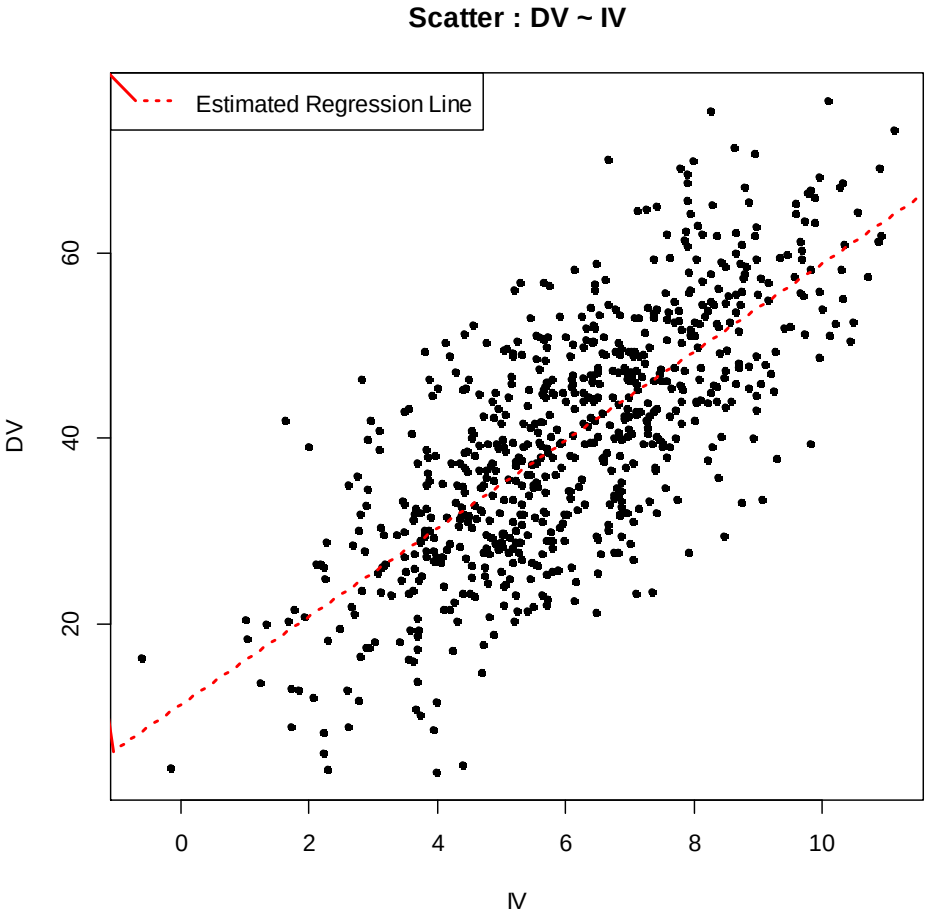
Observation Data Set Before Imputation

ID	IV	DV	
491			0
153			1
52			1
19			2
0	71	172	243

Observation Data Set After Imputation (MICE)



Estimated Regression Line Scatter Plot



Regression Model (IV and DV)

Call:

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

Residuals:

```
    Min      1Q  Median      3Q     Max
-27.4459  -6.2754   0.1104   5.8702  27.2303
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   11.219      1.090   10.29   <2e-16 ***
IV             4.759      0.170   28.00   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 9.04 on 694 degrees of freedom

Multiple R-squared: 0.5304,   Adjusted R-squared: 0.5298

F-statistic: 784 on 1 and 694 DF, p-value: < 2.2e-16

ANOVA Table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
IV	1	64068.97	64068.967	783.9868	0
Residuals	694	56715.06	81.722	NA	NA

Lack of Fit

Analysis of Variance Table

```
Response: y
            Df Sum Sq Mean Sq  F value Pr(>F)
x            1  63943   63943 779.2950 <2e-16 ***
Residuals    694  56841      82
 Lack of fit  34   2686      79   0.9627 0.5306
 Pure Error  660  54155      82
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R Code

```
> data_IV <- read.csv('P1A_IV48179.csv', header = TRUE)
> data_DV <- read.csv('P1A_DV48179.csv', header = TRUE)
> PartA <- merge(data_IV, data_DV, by = 'ID')
> str(PartA)
'data.frame':   715 obs. of  3 variables:
 $ ID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ IV: num  7.21 6.49 5.01 7.55 6.69 ...
 $ DV: num  39.3 40.1 29.7 NA 33 ...
> install.packages(‘mice’)
> library(mice)
> PartA_incomplete <- PartA
> md.pattern(PartA_incomplete)
    ID IV  DV
491  1  1   1   0
153  1  1   0   1
52   1  0   1   1
19   1  0   0   2
     0 71 172 243
```

```
>  PartA_impute <- PartA[!is.na(PartA$IV) == TRUE | !is.na(PartA$DV) == TRUE,]
> md.pattern(PartA_impute)
   ID IV  DV
491  1  1   1   0
153  1  1   0   1
52   1  0   1   1
     0 52 153 205
> imp <- mice(PartA_impute, method = "norm.boot", printFlag = FALSE)
> PartA_complete <- complete(imp)
> md.pattern(PartA_complete)
 /\     /\
{  `---'  }
{  O   O  }
==>  V <==  No need for mice. This data set is completely observed.
 \  \|/  /
  `-----'

   ID IV DV
696  1  1  1 0
     0  0  0 0
>  model <- lm(DV ~ IV, data = PartA_complete)
> summary(model)

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

Residuals:
     Min       1Q   Median       3Q      Max
-27.4459  -6.2754   0.1104   5.8702  27.2303
```

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	11.219	1.090	10.29	<2e-16	***
IV	4.759	0.170	28.00	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.04 on 694 degrees of freedom

Multiple R-squared: 0.5304, Adjusted R-squared: 0.5298

F-statistic: 784 on 1 and 694 DF, p-value: < 2.2e-16

```
> install.packages('knitr')
> library(knitr)
Warning message:
package 'knitr' was built under R version 3.5.3
> kable(anova(model), caption='ANOVA Table')
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
IV	1	64068.97	64068.967	783.9868	0
Residuals	694	56715.06	81.722	NA	NA

```
> plot(PartA_complete$DV ~ PartA_complete$IV, main='Scatter : DV ~ IV', xlab='IV', ylab='DV', pch=20)
> abline(model, col='red', lty=3, lwd=2)
> legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
> confint(model, 'IV', level=0.95)
    2.5 %   97.5 %
IV 4.425696 5.093175
> groups <- cut(PartA_complete$IV,breaks=c(-Inf,seq(min(PartA_complete$IV)+0.3,
max(PartA_complete$IV)-0.3,by=0.3),Inf))
```

```
> table(groups)
groups
  (-Inf,-0.324] (-0.324,-0.0244]  (-0.0244,0.276]    (0.276,0.576]
              1               1               0               0
  (0.576,0.876]    (0.876,1.18]     (1.18,1.48]     (1.48,1.78]
              0               2               2               5
   (1.78,2.08]     (2.08,2.38]     (2.38,2.68]     (2.68,2.98]
              4               9               6              15
   (2.98,3.28]     (3.28,3.58]     (3.58,3.88]     (3.88,4.18]
              9              14              33              20
   (4.18,4.48]     (4.48,4.78]     (4.78,5.08]     (5.08,5.38]
             30              33              39              38
   (5.38,5.68]     (5.68,5.98]     (5.98,6.28]     (6.28,6.58]
             46              35              28              37
   (6.58,6.88]     (6.88,7.18]     (7.18,7.48]     (7.48,7.78]
             43              40              31              25
   (7.78,8.08]     (8.08,8.38]     (8.38,8.68]     (8.68,8.98]
             31              24              21              23
   (8.98,9.28]     (9.28,9.58]     (9.58,9.88]     (9.88,10.2]
             10               6              13               8
   (10.2,10.5]     (10.5,10.8]     (10.8, Inf]
              8               2               4
> a <- ave(PartA_complete$IV, groups)
> bin <- data.frame(x=a, y = PartA_complete$DV)
> install.packages('alr3')
> library(alr3)
> fit <- lm(y ~ x, data = bin)
> pureErrorAnova(fit)
Analysis of Variance Table
```

```
Response: y
            Df Sum Sq Mean Sq  F value Pr(>F)
x            1  63943   63943 779.2950 <2e-16 ***
Residuals    694  56841      82
 Lack of fit  34   2686      79   0.9627 0.5306
 Pure Error  660  54155      82
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
>
```

R Code (My Procedure and Not Used In the Report) (Just For Show)

```
> PartA_IV <- read.csv('P1A_IV48179.csv', header = TRUE)
> PartA_DV <- read.csv('P1A_DV48179.csv', header = TRUE)
> library(mice)

> md.pattern(PartA_IV)
   ID IV
644  1  1  0
71   1  0  1
     0 71 71

> PartA_IV_imp <- PartA_IV[!is.na(PartA_IV$IV)== TRUE,]
> imp <- mice(PartA_IV_imp, method = "norm.boot", printFlag = FALSE)
> PartA_IV_Complete = complete(imp)
> md.pattern(PartA_IV_Complete)
 /\     /\
{  `---'  }
```

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

   ID IV
644  1  1 0
    0  0 0
> md.pattern(PartA_DV)
   ID  DV
543  1   1   0
172  1   0   1
    0 172 172
> PartA_DV_imp <- PartA_DV[!is.na(PartA_DV$DV)== TRUE,]
> imp2 <- mice(PartA_DV_imp, method = "norm.boot", printFlag = FALSE)
> PartA_DV_Complete = complete(imp2)
> md.pattern(PartA_DV_Complete)
 /\     /\
{  `---'  }
{ O   O }
==>  V <==  No need for mice. This data set is completely observed.
 \ \|/ /
  `-----'

   ID DV
543  1  1 0
    0  0 0

> PartA_Merged <- merge(PartA_IV_Complete, PartA_DV_Complete, by = 'ID')
```

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

    ID IV DV
491  1  1  1 0
     0  0  0 0
> M <- lm(DV ~ IV, data=PartA_Merged)
Warning message:
semi-transparency is not supported on this device: reported only once per page
> summary(M)
```

Call: \\ Models like the one below was generated multiple times from the same data, and they were all the same. However, unlike these models, models generated with the same data using the handout procedure were completely different.

```
lm(formula = DV ~ IV, data = PartA_Merged)

Residuals:
    Min      1Q  Median      3Q     Max
-27.4142  -6.3342   0.1079   5.9110  27.1831

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.0334     1.2966   8.509   <2e-16 ***
```

IV 4.7944 0.2054 23.346 <2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.146 on 489 degrees of freedom

Multiple R-squared: 0.5271, Adjusted R-squared: 0.5261

F-statistic: 545 on 1 and 489 DF, p-value: < 2.2e-16

Rahul Ranjan (ID – 111448179)

Professor Finch

AMS 315 Project 1: Part B Report

Introduction

This part of the problem carries the same conflict from part A. This time, I was given merged and sorted data and the dependent and independent variables were presented as y and x respectively. I will find the function that was used to generate the y values. However, this time we will follow a slightly different methodology to obtain a model and test its accuracy. I will perform linear regression on the merged data after finding an appropriate transformation of the data. A transformation will be needed if the r-squared value is low. An ANOVA table, confidence interval, and lack of fit test will be generated to assist in making conclusions about the generated model. If the results support the model, then its fitted function will be valid.

Methodology

Since the data was already merged, sorted, and stripped of any missing data, I did not need to use mice for imputation. I immediately generated a model using the lm() function. The model outputted a multiple r-squared value of 0.4141 which is moderate to low. I transformed my data many times to find the appropriate transformation. An appropriate transformation will be one that has the highest r-squared value and/or one that has a higher r-squared value than the r-squared value of the original data. A transformation will not always help data with a low r-squared value, but after trying multiple

transformations, I decided to use the reciprocal transformation. I constructed a model, and this generated the highest r-squared values out of all the other transformations. From here, I generated an ANOVA table, confidence interval, and lack of fit test to back up the new model. This time, the lack of fit test was performed by binning the new data into groups in order to reduce any consequences from observation errors. This worked by groups x and y values based off their proximity to one another. (Close values were put into the same group) After obtaining the results, I was able to make accurate conclusions about the new model and its function.

Result

Firstly, the total number of observations was 596. The function for the original data was y = 0.72000 – 0.36591(x) with an r-squared value of 0.4141 and r value of 0.6435. This r-squared value tells us that 41.41% of y variance is explained by x. The r value shows a moderate linear relationship. Given this data and after trying multiple other transformations, I decided to transform x and y through the reciprocal model. This was done by holding all the x values the same, but transforming all the y values into 1/y. (The transformation of 1/y for all the y values was set equal to ytran variable in my code and although x was not transformed, I set all the values of x from the original data to equal xtran variable) I tried multiple other transformation as shown in the code in the appendix, but these transformations either had lower r-squared values than the original data or were not higher than the reciprocal transformation r-squared value. The function for this new data was ytran = -18.2731 + 16.3247(xtran) with an r-squared value of 0.4864 and r value of 0.6974. (xtran is the same as x) This new r-squared value tells us that 48.64% of y variance is explained by x. The new r value shows a moderate linear relationship. This is a slight improvement over the original data, so the transformation helped. I generated the ANOVA table for this transformed data, and it backed up the data form the model. The division of the independent variable sum of squares by the total sum of squares generated a r-squared value of 0.486442439 which rounds off to 0.4864. (Same r-squared value from the model which was 2.2e^-16) The ANOVA table gave an F-value of 562.6376 and a p value of 0. Therefore, there Is a significant difference in the mean of both independent and dependent variable values. This p-value is very low so is extremely significant and as a result, the ANOVA favors this model. I constructed a 95% confidence interval for the slope of the estimated regression line was 14.97301 to 17.6763. The p value from the ANOVA table allows us to reject the null hypothesis that the slope is 0. (Due to significant levels of 0.01, 0.05, and 0.10 being higher than the p value) Since the confidence interval does not contain 0 and the recent test concluded that the slope is non-zero, it can be concluded that x and ytran have a relationship and the model can be a useful predictor. The corr(xtran, ytran) = 0.6974543 which rounded off is just 0.6974. (Same as the r value from the ANOVA table and model) This suggests that there is strong linear correlation between xtran and ytran. The lack of fit test showed a lack of fit F value of 0.8882 and a lack of fit p value of 0.7564. This p value is higher than the significant levels of 0.10, 0.05, and 0.01. Therefore, we fail to reject the null hypothesis that the model is appropriate.

Conclusion

Clearly after the transformation using the reciprocal model, there was a slight improvement of the r-squared value. The r-squared still isn't large, but it is moderate. Therefore, there exists a moderate association between x and ytran. As shown in the results, the confidence interval for the slope, and test of the slope equaling 0 also helped me. These helped to show that the slope had to be non-zero and there must co-exist a relationship between the variables. Finally, the lack of fit test determines the reliability of

the model and its function. The results showed that we failed to reject the model after binning the data into groups. According to all of this, I can conclude that the fitted function ytran = -18.2731 + 16.3247(xtran) from my model was used to generate the dependent variable values from the independent variable and I accept this model.

Appendix – Tables, Graphs, R Code and Other Important Data

Observation Data Set (Used MICE and There Are 596 Total Observations)

	ID	x	y	
596				0
	0	0	0	0

Regression Model Generated from Original Data (x and y)

Call:

lm(formula = y ~ x, data = PartB)

Residuals:

```
    Min      1Q  Median      3Q     Max
-0.08826 -0.02003 -0.00379  0.01691  0.38722

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.72000    0.02786   25.84   <2e-16 ***
x           -0.36591    0.01786  -20.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03395 on 594 degrees of freedom
Multiple R-squared:  0.4141,    Adjusted R-squared:  0.4132
F-statistic: 419.9 on 1 and 594 DF,  p-value: < 2.2e-16
```

Regression Model Generated from Transformed Data Using Reciprocal Model (xtran = x and ytran= 1/y)

(I Used This One)

```
Call:
lm(formula = ytran ~ xtran, data = data_tran)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7370 -0.9128 -0.0891  0.8323  5.1330

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -18.2731     1.0737  -17.02   <2e-16 ***
xtran        16.3247     0.6882   23.72   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 1.309 on 594 degrees of freedom

Multiple R-squared: 0.4864, Adjusted R-squared: 0.4856

F-statistic: 562.6 on 1 and 594 DF, p-value: < 2.2e-16

Regression Models Generated from Other Transformed Data Using Other Models (R-squared Value Lower Than Reciprocal Model)

(I Did Not Use Any of These)

Transformation : Quadratic Model (xtran = x and ytran = y^(1/2))

Call:

lm(formula = ytran ~ xtran, data = data_tran)

Residuals:

Min	1Q	Median	3Q	Max
-0.11883	-0.02586	-0.00342	0.02302	0.32620

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.10194	0.03212	34.31	<2e-16	***
xtran	-0.46111	0.02059	-22.40	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03914 on 594 degrees of freedom

Multiple R-squared: 0.4579, Adjusted R-squared: 0.457

F-statistic: 501.7 on 1 and 594 DF, p-value: < 2.2e-16

Transformation : Exponential Model (xtran = x and ytran = log(y))

Call:

lm(formula = ytran ~ xtran, data = data_tran)

Residuals:

Min	1Q	Median	3Q	Max
-0.6517	-0.1363	-0.0068	0.1246	1.1299

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.7517	0.1573	11.14	<2e-16	***
xtran	-2.3661	0.1008	-23.47	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1917 on 594 degrees of freedom

Multiple R-squared: 0.4812, Adjusted R-squared: 0.4803

F-statistic: 550.9 on 1 and 594 DF, p-value: < 2.2e-16

Transformation : Power Model (xtran = log(x) and ytran = log(y))

Call:

lm(formula = ytran ~ xtran, data = data_tran)

Residuals:

Min	1Q	Median	3Q	Max
-0.6520	-0.1341	-0.0057	0.1213	1.1256

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-0.30624	0.06968	-4.395	1.31e-05	***
xtran	-3.68309	0.15653	-23.529	< 2e-16	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1914 on 594 degrees of freedom

Multiple R-squared: 0.4824,   Adjusted R-squared: 0.4815

F-statistic: 553.6 on 1 and 594 DF, p-value: < 2.2e-16

Transformation : Logarithmic Model (xtran = log(x) and ytran = y))

Call:

lm(formula = ytran ~ xtran, data = data_tran)

Residuals:

Min	1Q	Median	3Q	Max
-0.08837	-0.02007	-0.00389	0.01651	0.38646

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.40234	0.01233	32.64	<2e-16	***
xtran	-0.57090	0.02769	-20.62	<2e-16	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03386 on 594 degrees of freedom

Multiple R-squared: 0.4171,   Adjusted R-squared: 0.4162

F-statistic: 425.1 on 1 and 594 DF, p-value: < 2.2e-16

Tranformation : Reciprocal of Both x and y (xtran = 1/x and ytran = 1/y))

Call:

lm(formula = ytran ~ xtran, data = data_tran)

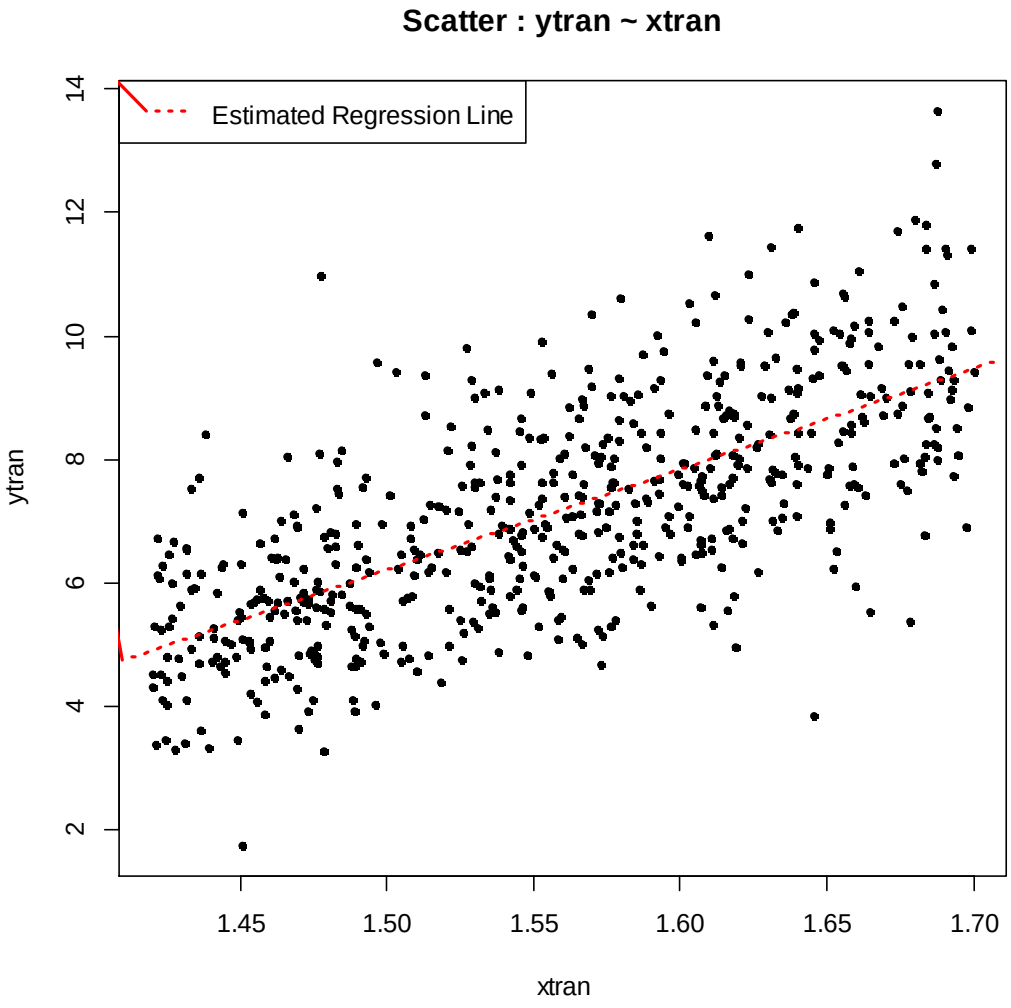
Residuals:

```
   Min     1Q  Median     3Q    Max
-4.7125 -0.8984 -0.0899  0.8250  5.1288

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   32.433      1.072   30.27   <2e-16 ***
xtran        -39.275      1.663  -23.61   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.311 on 594 degrees of freedom
Multiple R-squared:  0.4841,    Adjusted R-squared:  0.4833
F-statistic: 557.4 on 1 and 594 DF,  p-value: < 2.2e-16
```

Estimated Regression Line with Scatter Plot (xtran and ytran from Reciprocal Model Transformation)



ANOVA Table (xtran and ytran)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
xtran	1	963.3562	963.356152	562.6376	0
Residuals	594	1017.0553	1.712214	NA	NA

Lack of Fit (xtran and ytran)

Analysis of Variance Table

Response: y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x	1	962.62	962.62	551.9732	<2e-16 ***
Residuals	594	1017.79	1.71		
Lack of fit	93	144.06	1.55	0.8882	0.7564
Pure Error	501	873.73	1.74		

R Code

```
> PartB <- read.csv('P1B48179.csv', header = TRUE)
> library(mice)
> md.pattern(PartB)
/\     /\
{  `---'  }
{  O   O  }
==>  V <==  No need for mice. This data set is completely observed.
 \  \|/  /
```

`-----'

ID x y

596 1 1 1 0

0 0 0 0

> M <- lm(y ~ x, data = PartB)

> summary(M) //This is the model summary for the original data which. (Output shown above)

data_tran <- data.frame(xtran = PartB$x, ytran = PartB$y^(1/2))

> N <- lm(ytran ~ xtran, data = data_tran)

> summary(N) //This is the model summary for the transformed data by the quadratic model. (Output shown above)

> data_tran <- data.frame(xtran = PartB$x, ytran = log(PartB$y))

> N <- lm(ytran ~ xtran, data = data_tran)

> summary(N) //This is the model summary for the transformed data by the exponential model. (Output shown above)

> data_tran <- data.frame(xtran = log(PartB$x), ytran = log(PartB$y))

> N <- lm(ytran ~ xtran, data = data_tran)

> summary(N) //This is the model summary for the transformed data by the power model. (Output shown above)

> data_tran <- data.frame(xtran = log(PartB$x), ytran = PartB$y)

> N <- lm(ytran ~ xtran, data = data_tran)

> summary(N) //This is the model summary for the transformed data by the logarithmic model. (Output shown above)

> data_tran <- data.frame(xtran = PartB$x, ytran = 1/PartB$y)

> N <- lm(ytran ~ xtran, data = data_tran) //This is the model summary for the transformed data by the reciprocal model which ended up being my final model. (Output shown above)

> summary(N)

> data_tran <- data.frame(xtran = 1/PartB$x, ytran = 1/PartB$y)

> N <- lm(ytran ~ xtran, data = data_tran)

> summary(N) //This is the model summary for the transformed data by reciprocal of x and y. (Output shown above)

```
> groups <- cut(data_tran$xtran,breaks=c(-Inf,seq(min(data_tran$xtran)+0.3, max(data_tran$xtran)-0.3,by=-0.003),Inf))

> table(groups)

(-Inf,1.402] (1.402,1.405] (1.405,1.408] (1.408,1.411] (1.411,1.414]
           0            0            0            0            0
(1.414,1.417]  (1.417,1.42]  (1.42,1.423] (1.423,1.426] (1.426,1.429]
           0            1            8            8            4
(1.429,1.432] (1.432,1.435] (1.435,1.438] (1.438,1.441] (1.441,1.444]
           8            4            6            4            4
(1.444,1.447]  (1.447,1.45]  (1.45,1.453] (1.453,1.456] (1.456,1.459]
           5            4            6            7            7
(1.459,1.462] (1.462,1.465] (1.465,1.468] (1.468,1.471] (1.471,1.474]
           7            6            5           10            9
(1.474,1.477]  (1.477,1.48]  (1.48,1.483] (1.483,1.486] (1.486,1.489]
          13            6            9            3            8
(1.489,1.492] (1.492,1.495] (1.495,1.498] (1.498,1.501] (1.501,1.504]
           9            6            3            3            2
(1.504,1.507]  (1.507,1.51]  (1.51,1.513] (1.513,1.516] (1.516,1.519]
           5            6            5            5            4
(1.519,1.522] (1.522,1.525] (1.525,1.528] (1.528,1.531] (1.531,1.534]
           6            2            8            9            4
(1.534,1.537]  (1.537,1.54]  (1.54,1.543] (1.543,1.546] (1.546,1.549]
           9            6            8           11            6
(1.549,1.552] (1.552,1.555] (1.555,1.558] (1.558,1.561] (1.561,1.564]
           5            9            6            9            6
(1.564,1.567]  (1.567,1.57]  (1.57,1.573] (1.573,1.576] (1.576,1.579]
          12            6            9            7           11
(1.579,1.582] (1.582,1.585] (1.585,1.588] (1.588,1.591] (1.591,1.594]
           7            8            6            5            8
```

```
(1.594,1.597]   (1.597,1.6]   (1.6,1.603] (1.603,1.606] (1.606,1.609]
         5           2           8           6          10
(1.609,1.612] (1.612,1.615] (1.615,1.618] (1.618,1.621] (1.621,1.624]
        11           9           9          11           5
(1.624,1.627]  (1.627,1.63]  (1.63,1.633] (1.633,1.636] (1.636,1.639]
         3           4          10           4           5
(1.639,1.642] (1.642,1.645] (1.645,1.648] (1.648,1.651] (1.651,1.654]
         9           3           6           3           5
(1.654,1.657]  (1.657,1.66]  (1.66,1.663] (1.663,1.666] (1.666,1.669]
         8           9           6           6           3
(1.669,1.672] (1.672,1.675] (1.675,1.678] (1.678,1.681] (1.681,1.684]
         1           4           6           4           8
(1.684,1.687]  (1.687,1.69]  (1.69,1.693] (1.693,1.696] (1.696,1.699]
         7           7           8           3           3
(1.699,1.702] (1.702,1.705] (1.705,1.708] (1.708,1.711] (1.711,1.714]
         2           0           0           0           0
(1.714,1.717]  (1.717,1.72]   (1.72, Inf]
         0           0           0
> library(alr3)
a <- ave(data_tran$xtran, groups)
bin <- data.frame(x=a, y=data_tran$ytran)
> fit <- lm(y ~ x, data = bin)
> pureErrorAnova(fit)
Analysis of Variance Table


Response: y
            Df  Sum Sq Mean Sq  F value Pr(>F)
x            1  962.62  962.62 551.9732 <2e-16 ***
Residuals    594 1017.79    1.71
```

```
 Lack of fit  93  144.06    1.55   0.8882 0.7564
 Pure Error  501  873.73    1.74
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> confint(N, 'xtran', level = 0.95)
        2.5 %  97.5 %
xtran 14.97301 17.6763
> library(knitr)
Warning message:
package ‘knitr’ was built under R version 3.5.3
> kable(anova(N), caption = 'ANOVA Table')



|          |  Df|    Sum Sq|    Mean Sq|  F value| Pr(>F)|
|:---------|---:|---------:|----------:|--------:|------:|
|xtran     |   1|  963.3562| 963.356152| 562.6376|      0|
|Residuals | 594| 1017.0553|   1.712214|       NA|     NA|
> install.packages('Hmisc') //All the downloads and packages installed are not shown.
> library(Hmisc)
> cor(xtran,ytran)
[1] 0.6974543
>  plot(ytran ~ xtran, main='Scatter : ytran ~ xtran', xlab='xtran', ylab='ytran', pch=20)
> abline(N, col='red', lty=3, lwd=2)
> legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
```