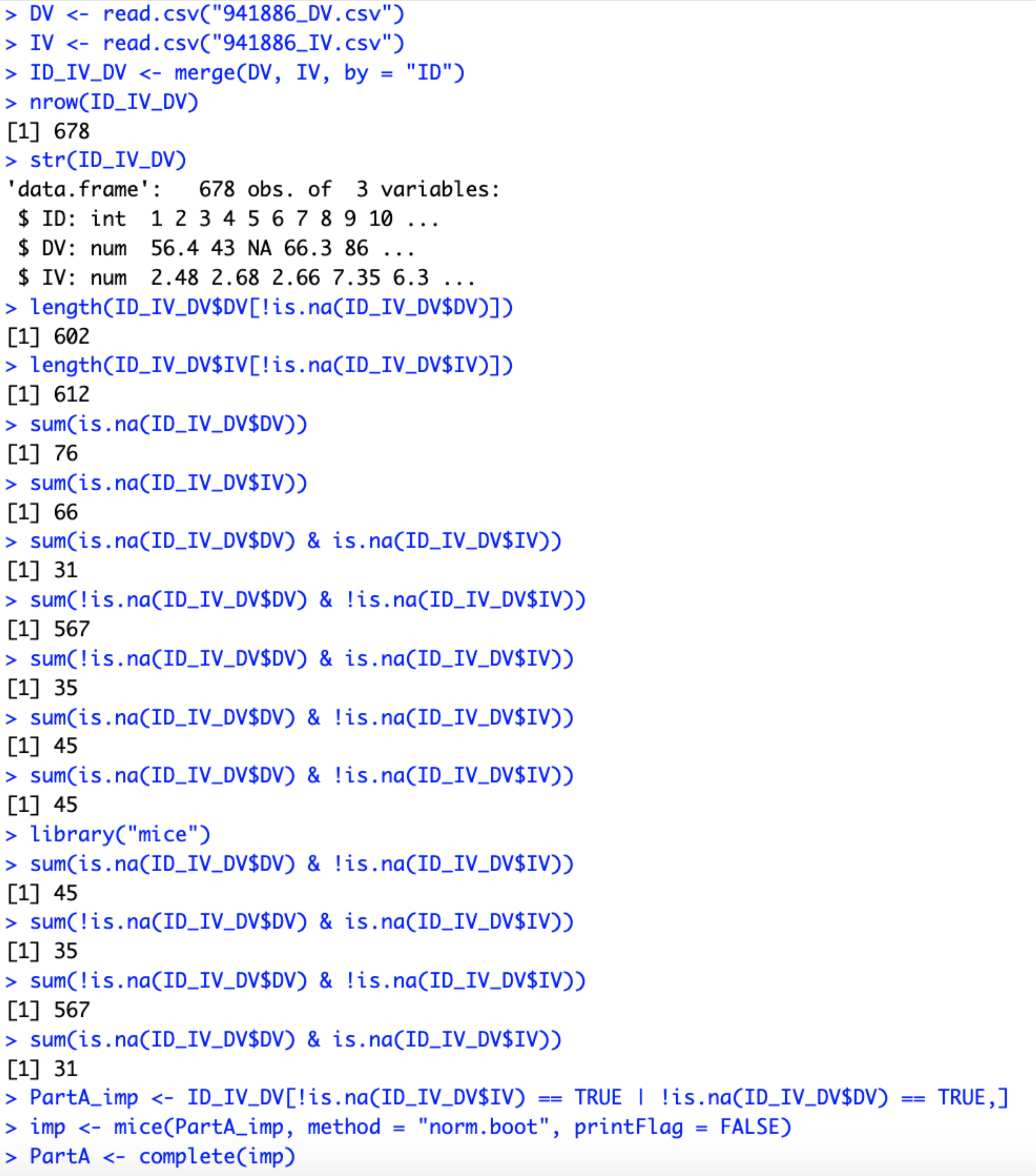
The purpose of Part A wass to combine two datasets and fit a regression model to my combined dataset. Each of the datasets included a column for subject ID and another column for either the dependent or independent variable value. The main problem at hand was because some of the data is missing, I must organize and impute the combined dataset before I can fit a regression model to it.

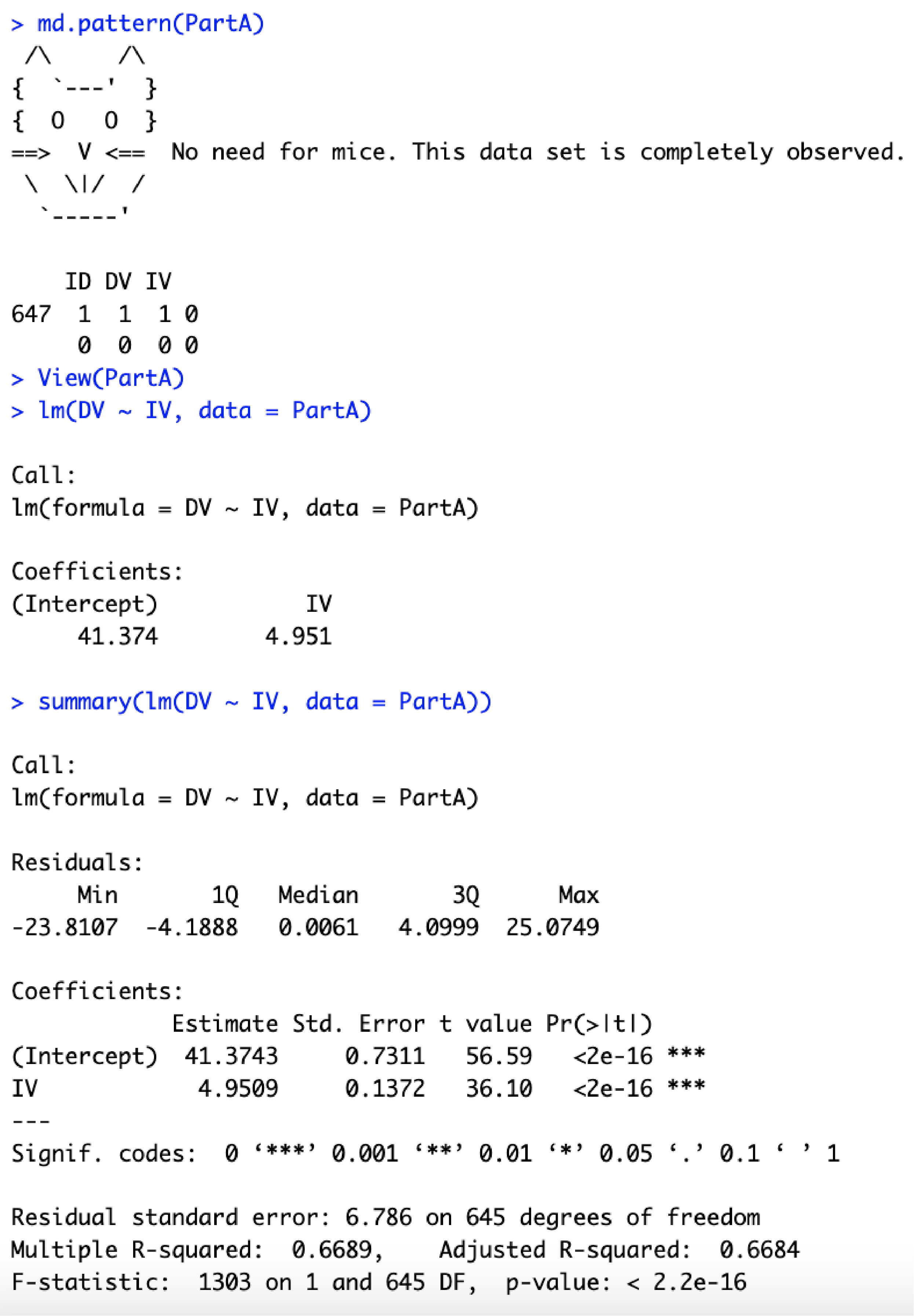
This project was done on RStudio. The first step in completing Part A was to import the two datasets that I was given using "read.csv". One dataset included ID and DV (dependent variable), while the other included ID and IV (independent variable). I then merged the two datasets into one named "ID_IV_DV" by the subject ID. Since some of the subject IDs were missing the DV, IV, or even both, I determined what data I did have. There were a total of 678 observations which meant that there were 678 subject IDs. There were 76 missing values in DV and 66 missing values in IV. Out of those 678 IDs, 567 had both an IV and a DV, 45 had an IV but no DV, 35 had a DV but no IV, and 31 of them had neither an IV nor DV. I adjusted the dataset to exclude the 31 IDs that did not have IV and DV values as they have no data. From there, I used the package called "mice" to impute the data with missing values with a linear regression bootstrap method. I made sure that the data set of 647 IDs was completely observed using the "md.pattern()" tool. To fit a regression model to the imputed data, I used linear regression to determine the slope and intercept of the estimated regression line. After that, I also summarized the regression model, used ANOVA, and calculated the confidence interval of the slope to better understand the results of the test.

Based on my linear regression model, the slope is 4.951 and the y-intercept is 41.374. In Figure 1, the plot of the imputed dataset can be shown as well as the line of best fit. The 95% confidence interval for this model is [4.68, 5.22], which means at the 95% level, we can reject the null hypothesis that the slope is zero. It can be concluded that the slope of the model is not zero as zero is not included in the confidence interval. At the 99% level, the confidence interval is [4.60, 5.31]. Since this interval does not include zero, his result further helps conclude that we are 99% confident that the slope is not zero.

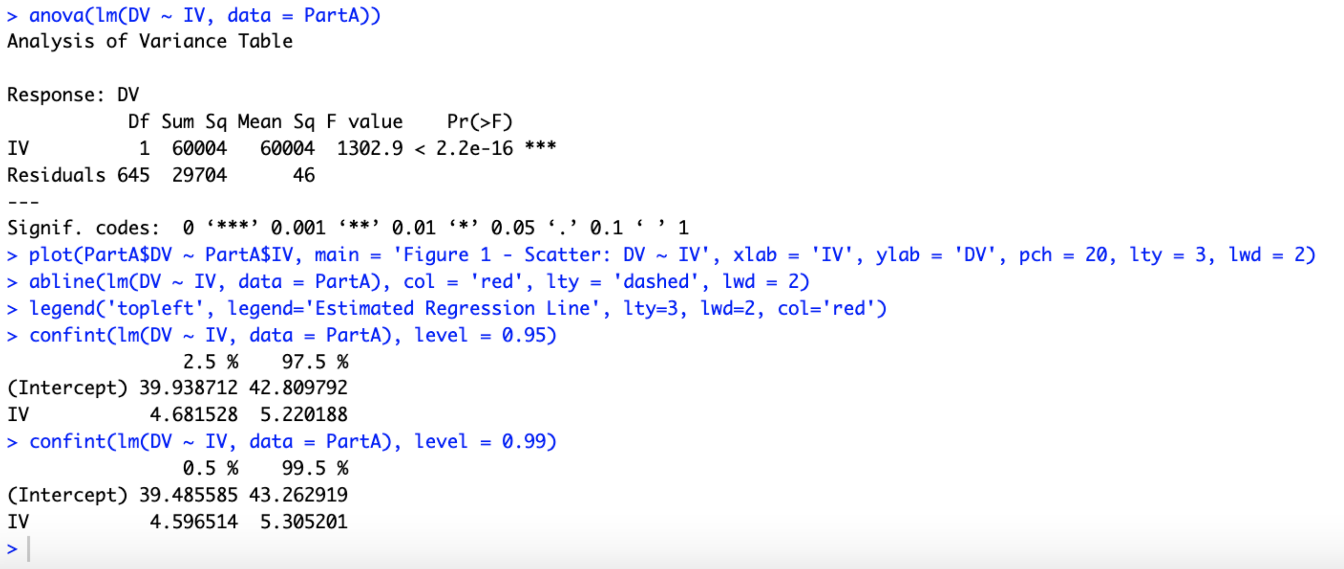
In conclusion, there is an association between the IV and DV of the subject IDs. The summary of this model shows that the R squared is 0.6689, so there is a weak positive correlation. The line of best fit is DV = 4.951*IV + 41.374. Attached below in this report is my RStudio code.



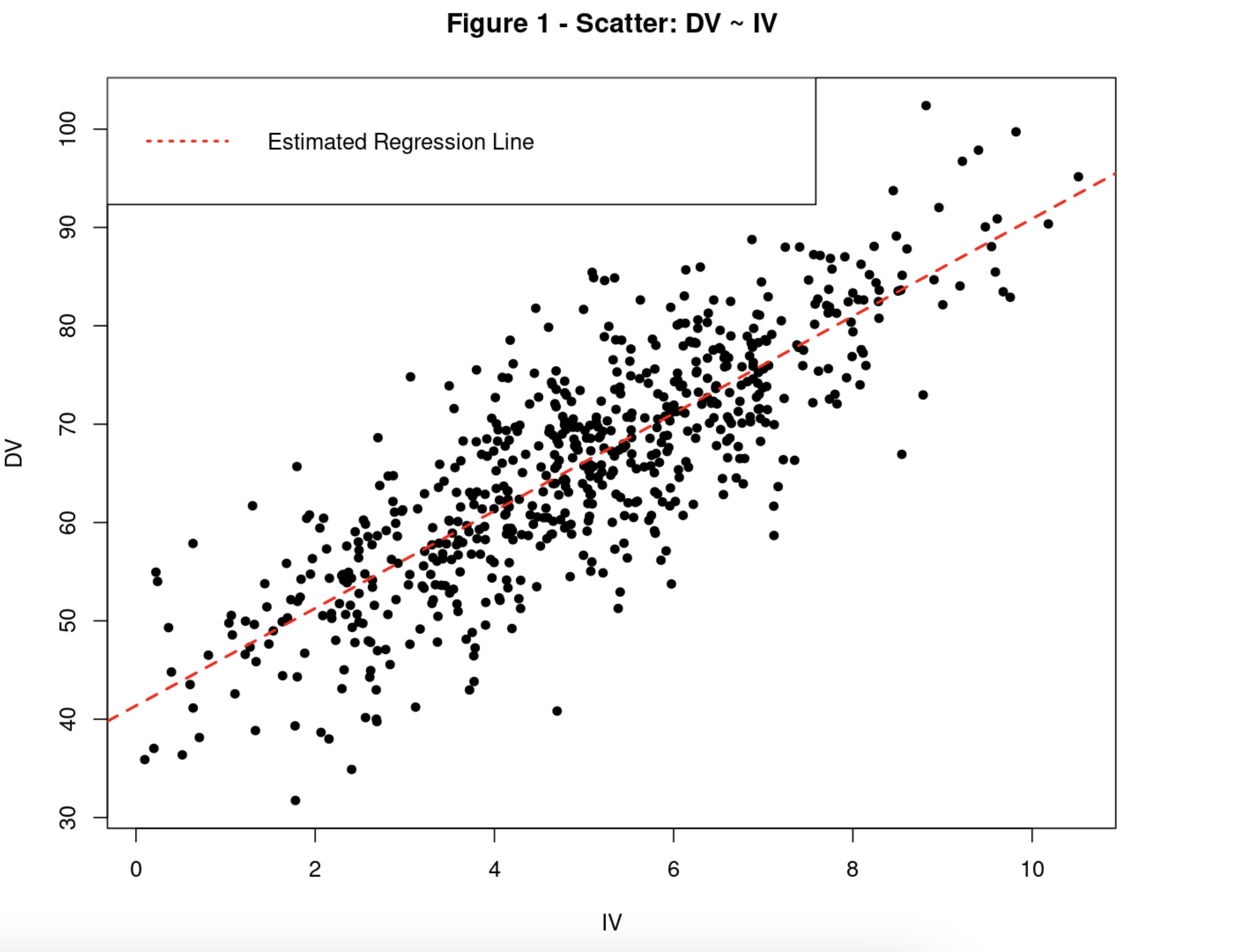
```
> DV <- read.csv("941886_DV.csv")
> IV <- read.csv("941886_IV.csv")
> ID_IV_DV <- merge(DV, IV, by = "ID")
> nrow(ID_IV_DV)
[1] 678
> str(ID_IV_DV)
'data.frame':	678 obs. of  3 variables:
 $ ID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ DV: num  56.4 43 NA 66.3 86 ...
 $ IV: num  2.48 2.68 2.66 7.35 6.3 ...
> length(ID_IV_DV$DV[!is.na(ID_IV_DV$DV)])
[1] 602
> length(ID_IV_DV$IV[!is.na(ID_IV_DV$IV)])
[1] 612
> sum(is.na(ID_IV_DV$DV))
[1] 76
> sum(is.na(ID_IV_DV$IV))
[1] 66
> sum(is.na(ID_IV_DV$DV) & is.na(ID_IV_DV$IV))
[1] 31
> sum(!is.na(ID_IV_DV$DV) & !is.na(ID_IV_DV$IV))
[1] 567
> sum(!is.na(ID_IV_DV$DV) & is.na(ID_IV_DV$IV))
[1] 35
> sum(is.na(ID_IV_DV$DV) & !is.na(ID_IV_DV$IV))
[1] 45
> sum(is.na(ID_IV_DV$DV) & !is.na(ID_IV_DV$IV))
[1] 45
> library("mice")
> sum(is.na(ID_IV_DV$DV) & !is.na(ID_IV_DV$IV))
[1] 45
> sum(!is.na(ID_IV_DV$DV) & is.na(ID_IV_DV$IV))
[1] 35
> sum(!is.na(ID_IV_DV$DV) & !is.na(ID_IV_DV$IV))
[1] 567
> sum(is.na(ID_IV_DV$DV) & is.na(ID_IV_DV$IV))
[1] 31
> PartA_imp <- ID_IV_DV[!is.na(ID_IV_DV$IV) == TRUE | !is.na(ID_IV_DV$DV) == TRUE,]
> imp <- mice(PartA_imp, method = "norm.boot", printFlag = FALSE)
> PartA <- complete(imp)
```



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

    ID DV IV  
647  1  1  1 0
     0  0  0 0
> View(PartA)
> lm(DV ~ IV, data = PartA)

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

Coefficients:
(Intercept)           IV  
     41.374        4.951  

> summary(lm(DV ~ IV, data = PartA))

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

Residuals:
     Min       1Q   Median       3Q      Max 
-23.8107  -4.1888   0.0061   4.0999  25.0749 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  41.3743     0.7311   56.59   <2e-16 ***
IV            4.9509     0.1372   36.10   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.786 on 645 degrees of freedom
Multiple R-squared:  0.6689,    Adjusted R-squared:  0.6684 
F-statistic:  1303 on 1 and 645 DF,  p-value: < 2.2e-16
```



```
> anova(lm(DV ~ IV, data = PartA))
Analysis of Variance Table

Response: DV
           Df Sum Sq Mean Sq F value    Pr(>F)
IV          1  60004   60004  1302.9 < 2.2e-16 ***
Residuals 645  29704      46
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> plot(PartA$DV ~ PartA$IV, main = 'Figure 1 - Scatter: DV ~ IV', xlab = 'IV', ylab = 'DV', pch = 20, lty = 3, lwd = 2)
> abline(lm(DV ~ IV, data = PartA), col = 'red', lty = 'dashed', lwd = 2)
> legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
> confint(lm(DV ~ IV, data = PartA), level = 0.95)
                2.5 %    97.5 %
(Intercept) 39.938712 42.809792
IV           4.681528  5.220188
> confint(lm(DV ~ IV, data = PartA), level = 0.99)
                0.5 %    99.5 %
(Intercept) 39.485585 43.262919
IV           4.596514  5.305201
> 
```



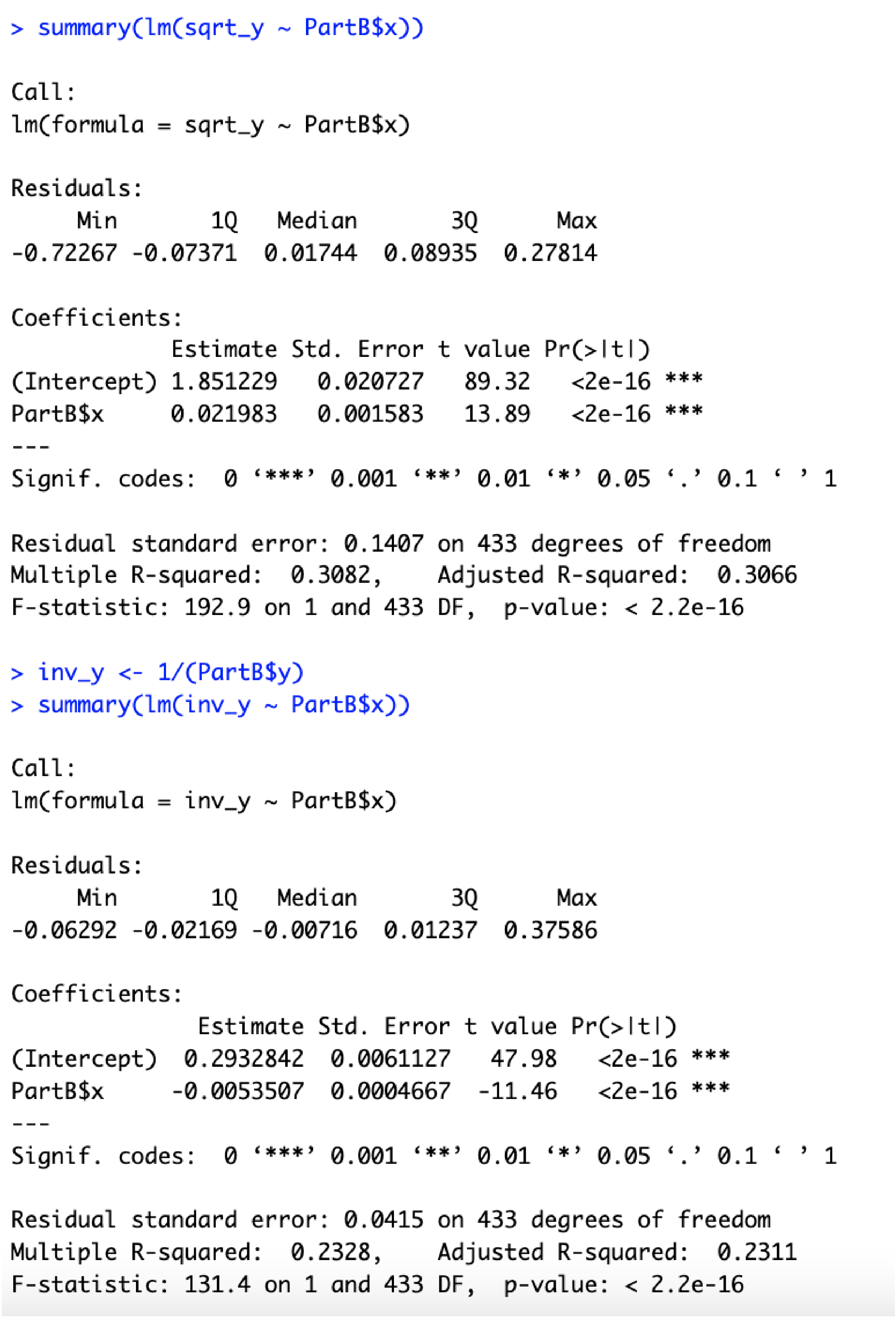
The purpose of Part B is to determine whether or not I need to transform a dataset and fit a regression model to that transformed dataset if necessary. With this, I can recover the function that was used to determine the dependent variable value from the independent variable value. Since this dataset is quite large with similar independent variable values, I need to bin these values first before I fit a regression model as well as a lack of fit test.

For Part B, I also used RStudio. The first thing I did was to import the dataset, which included three variables: ID, x, and y. There were 435 observations and each ID had both an x- and y-value.To see whether or not a transformation was needed, I tested out different types of transformations like the square root of y, inverse of y, square of y, natural logarithm of y, square of y, and the cube of y. With this, I summarized the linear regression of each of these transformations in relation to the x values. To determine which transformation, if any, was better than no transformation at all. I looked at each of the R squared values in the summary. After that, I transformed the data so that the x-values stay the same but all the y-values are changed based on which transformation was best. To bin the data, I used the "cut()" and "ave()" tools to group the x-values by 0.3 intervals. In the package "olsrr", I used "ols_pure_error_anova()" to perform the lack of fit test to determine whether or not my transformation was a good option. If it was, I would then be able to fit a linear regression model to the transformed data.

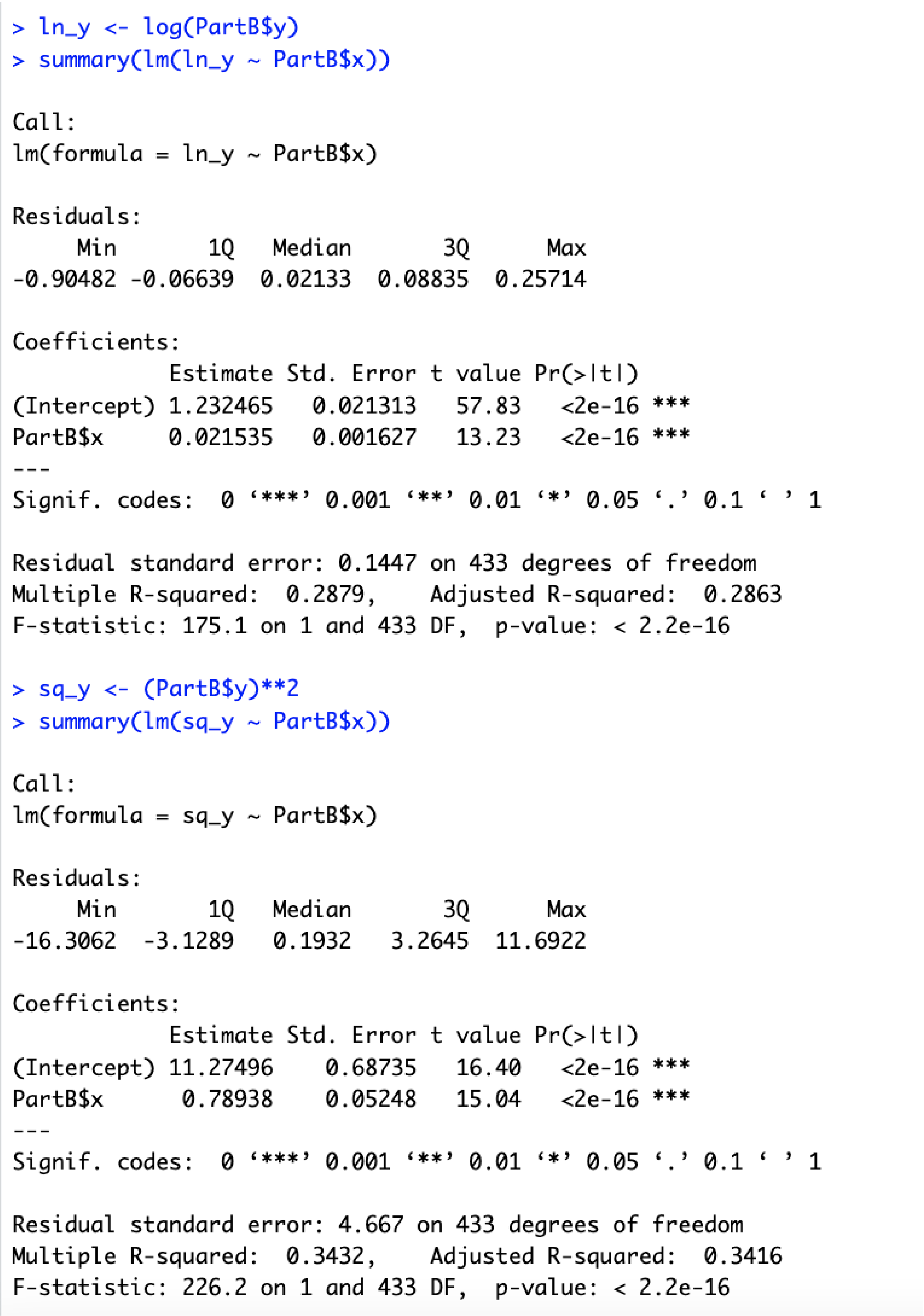
Based on the summaries of the linear regression models of the transformed y-values, I found that the cubed values of y had the highest R squared out of all the possible transformations at 0.3502. This value was also higher than if there was no transformation. The transformed data was now (x, y^3). When I performed the lack of fit test with the binned x-values, I was able to conclude that this transformation of cubed y-values was a good option. Since there was a high p-value of 0.34392, we fail to reject the null hypothesis that the relationship in this model is reasonable. I was able to conclude that there is not enough evidence to show that there is a lack of fit in this linear regression model. With this and the result that the R squared value is higher than the original and the other transformations, this is a good transformation for this dataset. From here, I was able to fit a regression model for this data. The slope is 5.2971 and the y-intercept is 33.7876.

The results of this project are that a good transformation for my dataset is to keep the x-values the same but to cube the y-values. This new and transformed data has a fitted function of y = 5.2971x + 33.7876. Attached below is my RStudio code.

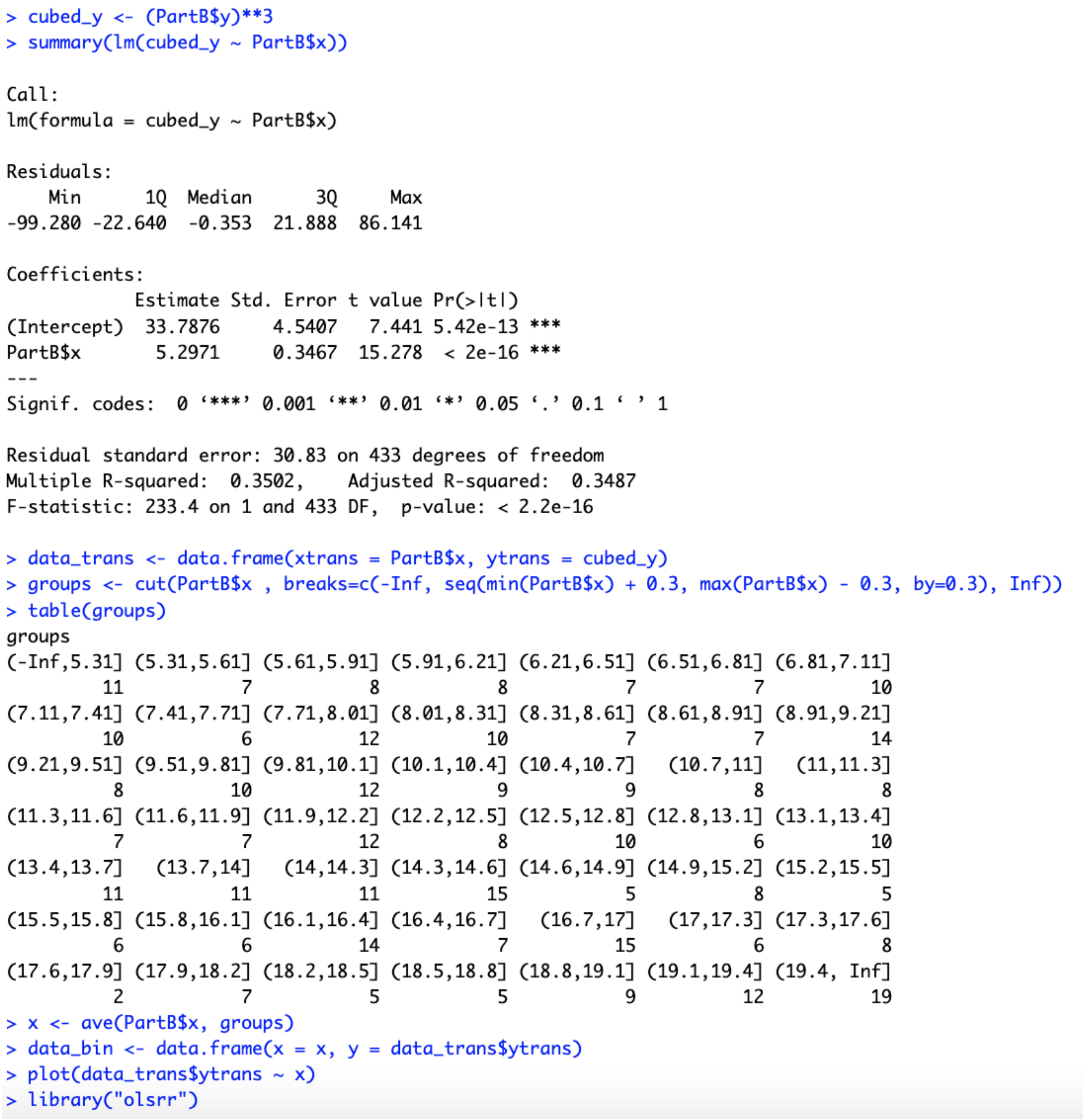
```
> PartB <- read.csv("941886_PartB.csv")
> View(PartB)
> summary(lm(PartB$y ~ PartB$x))

Call:
lm(formula = PartB$y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.35424 -0.32801  0.05547  0.37159  1.21170 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.415189   0.082423   41.44   <2e-16 ***
PartB$x     0.090633   0.006294   14.40   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5596 on 433 degrees of freedom
Multiple R-squared:  0.3238,	Adjusted R-squared:  0.3223 
F-statistic: 207.4 on 1 and 433 DF,  p-value: < 2.2e-16

> sqrt_y <- sqrt(PartB$y)
> summary(lm(sqrt_y ~ PartB$x))

Call:
lm(formula = sqrt_y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.72267 -0.07371  0.01744  0.08935  0.27814 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.851229   0.020727   89.32   <2e-16 ***
PartB$x     0.021983   0.001583   13.89   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



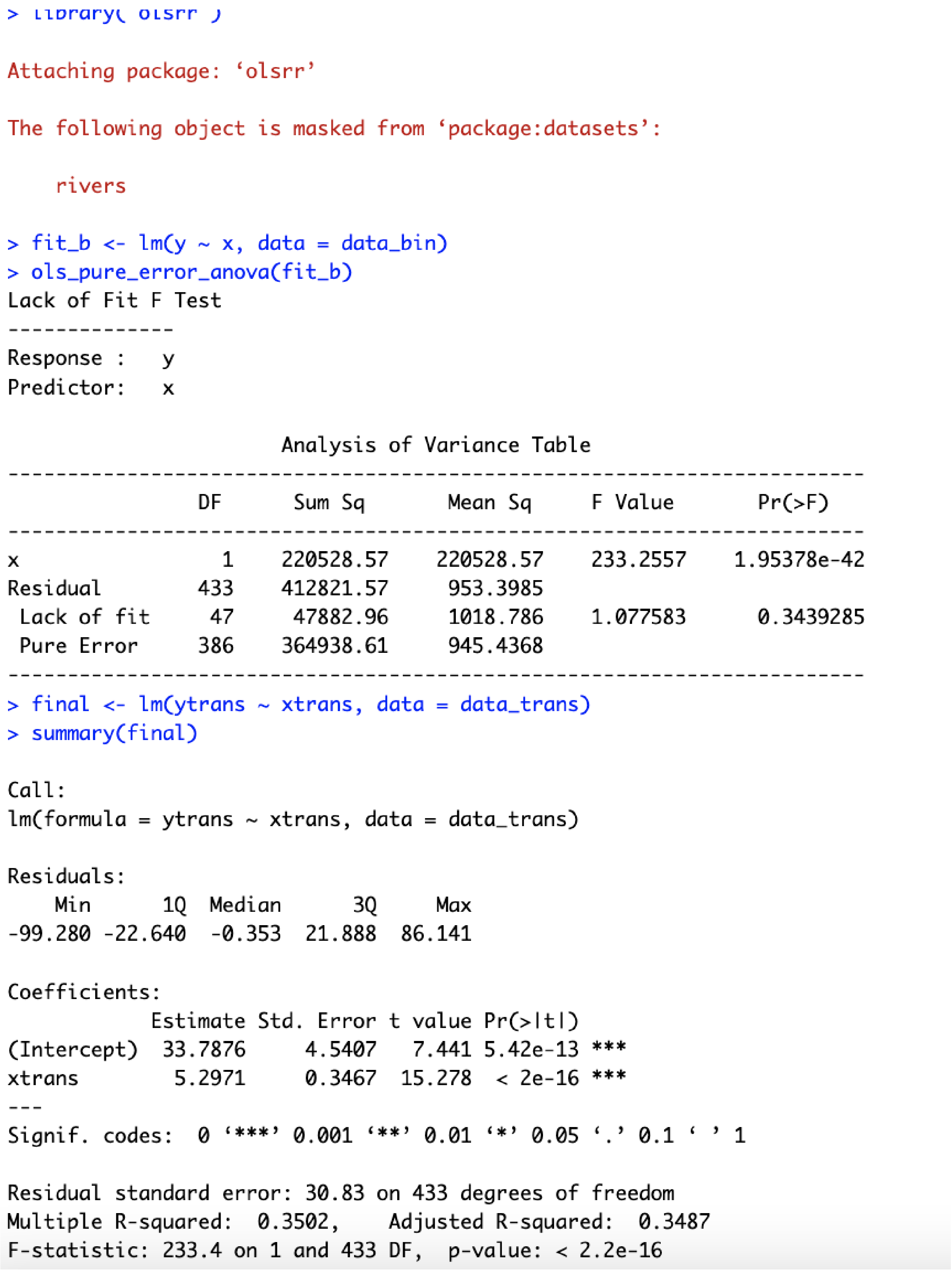
```
> summary(lm(sqrt_y ~ PartB$x))

Call:
lm(formula = sqrt_y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.72267 -0.07371  0.01744  0.08935  0.27814 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.851229   0.020727   89.32   <2e-16 ***
PartB$x     0.021983   0.001583   13.89   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1407 on 433 degrees of freedom
Multiple R-squared:  0.3082,	Adjusted R-squared:  0.3066 
F-statistic: 192.9 on 1 and 433 DF,  p-value: < 2.2e-16

> inv_y <- 1/(PartB$y)
> summary(lm(inv_y ~ PartB$x))

Call:
lm(formula = inv_y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.06292 -0.02169 -0.00716  0.01237  0.37586 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.2932842  0.0061127   47.98   <2e-16 ***
PartB$x     -0.0053507  0.0004667  -11.46   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0415 on 433 degrees of freedom
Multiple R-squared:  0.2328,	Adjusted R-squared:  0.2311 
F-statistic: 131.4 on 1 and 433 DF,  p-value: < 2.2e-16
```



```
> ln_y <- log(PartB$y)
> summary(lm(ln_y ~ PartB$x))

Call:
lm(formula = ln_y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.90482 -0.06639  0.02133  0.08835  0.25714 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.232465   0.021313   57.83   <2e-16 ***
PartB$x     0.021535   0.001627   13.23   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1447 on 433 degrees of freedom
Multiple R-squared:  0.2879,	Adjusted R-squared:  0.2863 
F-statistic: 175.1 on 1 and 433 DF,  p-value: < 2.2e-16

> sq_y <- (PartB$y)**2
> summary(lm(sq_y ~ PartB$x))

Call:
lm(formula = sq_y ~ PartB$x)

Residuals:
     Min       1Q   Median       3Q      Max 
-16.3062  -3.1289   0.1932   3.2645  11.6922 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 11.27496    0.68735   16.40   <2e-16 ***
PartB$x      0.78938    0.05248   15.04   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.667 on 433 degrees of freedom
Multiple R-squared:  0.3432,	Adjusted R-squared:  0.3416 
F-statistic: 226.2 on 1 and 433 DF,  p-value: < 2.2e-16
```



```
> cubed_y <- (PartB$y)**3
> summary(lm(cubed_y ~ PartB$x))

Call:
lm(formula = cubed_y ~ PartB$x)

Residuals:
    Min      1Q  Median      3Q     Max 
-99.280 -22.640  -0.353  21.888  86.141 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  33.7876     4.5407   7.441 5.42e-13 ***
PartB$x       5.2971     0.3467  15.278  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 30.83 on 433 degrees of freedom
Multiple R-squared:  0.3502,	Adjusted R-squared:  0.3487 
F-statistic: 233.4 on 1 and 433 DF,  p-value: < 2.2e-16

> data_trans <- data.frame(xtrans = PartB$x, ytrans = cubed_y)
> groups <- cut(PartB$x , breaks=c(-Inf, seq(min(PartB$x) + 0.3, max(PartB$x) - 0.3, by=0.3), Inf))
> table(groups)
groups
(-Inf,5.31] (5.31,5.61] (5.61,5.91] (5.91,6.21] (6.21,6.51] (6.51,6.81] (6.81,7.11] 
         11           7           8           8           7           7          10 
(7.11,7.41] (7.41,7.71] (7.71,8.01] (8.01,8.31] (8.31,8.61] (8.61,8.91] (8.91,9.21] 
         10           6          12          10           7           7          14 
(9.21,9.51] (9.51,9.81] (9.81,10.1] (10.1,10.4] (10.4,10.7]   (10.7,11]   (11,11.3] 
          8          10          12           9           9           8           8 
(11.3,11.6] (11.6,11.9] (11.9,12.2] (12.2,12.5] (12.5,12.8] (12.8,13.1] (13.1,13.4] 
          7           7          12           8          10           6          10 
(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] 
         11          11          11          15           5           8           5 
(15.5,15.8] (15.8,16.1] (16.1,16.4] (16.4,16.7]   (16.7,17]   (17,17.3] (17.3,17.6] 
          6           6          14           7          15           6           8 
(17.6,17.9] (17.9,18.2] (18.2,18.5] (18.5,18.8] (18.8,19.1] (19.1,19.4] (19.4, Inf] 
          2           7           5           5           9          12          19 
> x <- ave(PartB$x, groups)
> data_bin <- data.frame(x = x, y = data_trans$ytrans)
> plot(data_trans$ytrans ~ x)
> library("olsrr")
```



```
> library( olsrr )

Attaching package: ‘olsrr’

The following object is masked from ‘package:datasets’:

    rivers

> fit_b <- lm(y ~ x, data = data_bin)
> 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    220528.57     220528.57     233.2557    1.95378e-42
Residual       433    412821.57      953.3985
 Lack of fit    47     47882.96      1018.786     1.077583      0.3439285
 Pure Error    386    364938.61      945.4368
------------------------------------------------------------------------
> final <- lm(ytrans ~ xtrans, data = data_trans)
> summary(final)

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

Residuals:
    Min      1Q  Median      3Q     Max
-99.280 -22.640  -0.353  21.888  86.141

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.7876     4.5407   7.441 5.42e-13 ***
xtrans        5.2971     0.3467  15.278  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 30.83 on 433 degrees of freedom
Multiple R-squared:  0.3502,	Adjusted R-squared:  0.3487
F-statistic: 233.4 on 1 and 433 DF,  p-value: < 2.2e-16
```