PART A

Introduction: In part A, the objective is to find the linear regression relationship between the independent variable and the dependent variable. We need to analyze the dataset from two different files and merge them together. The research problem is whether there's an association between IV and DV, and impute the fitted linear regression.

Methods: First, we need to read the files with code read.csv("354111_IV.csv") and read.csv("354111_DV.csv"), and then merge them together based on ID by method md.pattern(). Since there are many missing data in the dataset, I need to drop the observations that miss both IV and DV. The imputation method that I'm going to use is the package called "mice" with the method "norm.boot() to impute the missing value. Then we used lm() to fit a linear regression model, it allows R to return the table that summarizes the linear regression. We got the ANOVA table for the linear model using the anova() function and presented it in a table using the kable() function from the "knitr" package. At last, we plotted a scatter plot of IV and DV using the plot() function and obtained the confidence interval for the intercept and slope of the regression line at the 95% and 99% levels using the confint() function.

Results: In 608 observations, there are 492 complete data sets which have both IV and DV. IV is missing in 62 cases, DV is missing in 72 cases, and both are missing in 18 cases that need to be dropped. We got the fitted regression function 5.0150IV+42.6369, so the null hypothesis should be rejected. The residual standard error is 8.389 on 588 degrees of freedom. The multiple R-squared is 0.5992, the adjusted R-squared is 0.5985, the F-statistic is 879.1 on 1 and 588 DF, and the p-value is smaller than 2.2e-16. The null hypothesis should be rejected since the p-value is a small value. The 95% CI of the intercept is 4.682804 to 5.347187, and the 99% CI of the intercept is 4.577902 to 5.452089, which indicates that we can be 95% (or 99%) confident that the true intercept value falls within this range. The test of the null hypothesis that the slope is zero is rejected when the alpha is 0.05.

Conclusion: In part A, the analysis suggests that there are strong association between IV and DV. The fitted linear regression model equation is DV=5.0150IV+42.6369. About 60% of the variation in the DV can be explained by the IV included in the model, and the F-statistic of 879.1 with a p-value less than 0.05 shows the model is fitted the data. The null hypothesis of zero slopes was rejected since the p-value is very small. And the rejection of the null hypothesis also proves that IV is a significant predictor of DV in the model.

Part B

Introduction: In Part B, we are going to analyze a dataset consisting of ID with its IV and DV, and get a more fitted model for IV and DV by performing the transformation. By applying the transformations, we can create a new transformed model and assess its fit using the Lack of Fit(LOF) test. The objective is to determine whether the transformed model provides a better fit to the data than the original linear regression model.

Methods: I set IV as x, and DV as y. I check the original R-square value first with code Origin<-lm(I(y)~I(x), data=PartB), summary(Origin), and got the R-square value 0.4467. I began the transformation started from transformed y to y^(-1) with code B1<-lm(I(y^(-1)) ~ I(x), data=PartB). Then run summary(B1), R returned the R-square value for this transformation which is 0.3705. Then I just used the same code for the rest transformation, transform y to y^(-½), y^(2), y^(½), log(y), and transform x to x^(-½), x^(2), x^(½), log(x). The result table is shown below as Table 1. Then I select the one with the largest R-square value when y transformed to y^(2) and x stays the same. The R-square value for y^(2) and x is 0.4629 which is greater than the original R-square value. Then I used the tool cut() to create the groups and ave() to find the mean value of each group. At last, I apply the LOF test by using the function los_pure_error_anova() in the package "oslrr" to test the transformed model.

Results: There are 463 observations in the dataset Part B. Accordion to Table 1 below, the largest R-square value I can find is 0.4629 when y is transformed to y^(2) and x stays the same. Therefore, the transformed equation would be y^(2)=5.3718x+38.9760 or DV^(2)=5.3718IV+38.9760. The F-statistic is 397.3 on 1 and 461 DF, and the p-value is smaller than 2.2e-16 which indicates that strong evidence that the slope of the linear regression line is significantly different from zero. According to Table 2 below, the results of the ANOVA test on the pure error model for the linear regression of y on x show that the F-value for the predictor x is 386.0209 with a very small p-value of 6.963848e-63. This indicates that the predictor x has a significant effect on the response y. The lack of fit test also shows that the F-value is 0.6887149 with a p-value of 0.9417984, which is not significant.

Conclusions: In Part B, we found a better linear regression model by transforming IV and DV based on the largest R-square value. The model I got is DV^(2)=5.3718IV+38.9760. The R-square value is changed from 0.4467 to 0.4629, and the p-value is smaller than 2.2e-16 which determine the predictor variable IV has a significant effect on the response variable DV. We could identified the association

between the variables. The lack of fit test suggests that the model fits the data well, and there is no significant difference between the fit of the model and the fit of the data.

Table1

Table 1

	y	y^(-1)	y^(-1/2)	y^(1/2)	y^(2)	log(y)
x	0.4467	0.3705	0.3948	0.4331	0.4629	0.4158
x^(-1)	0.4046	0.3679	0.3824	0.4008	0.4031	0.3934
x^(-1/2)	0.4254	0.3775	0.395	0.419	0.4282	0.4089
x^(1/2)	0.4469	0.3786	0.4011	0.4354	0.4588	0.4202
x^(2)	0.4292	0.3429	0.369	0.4125	0.4521	0.3924
Log(x)	0.4398	0.3812	0.4014	0.4308	0.4472	0.418

Table2

```
> 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    252634.75    252634.75     386.0209    6.963848e-63
Residual       461    292130.47     633.6887
 Lack of fit    47     21184.57     450.7354    0.6887149       0.9417984
 Pure Error    414    270945.91     654.4587
--------------------------------------------------------------------------
```

Part A code

```
install.packages(“ISwR”)
library("ISwR")
setwd("~/Desktop")
iv<-read.csv("354111_IV.csv")
dv<-read.csv("354111_DV.csv")
merge_data<-merge(iv,dv,by="ID",all=TRUE)
str(merge_data)
any(is.na(merge_data[,2])==TRUE)
any(is.nan(PartA[,2])==TRUE)
any(is.na(PartA[,3]) == TRUE)
any(is.nan(PartA[,3]) == TRUE)
incomplete_data<-merge_data
library(mice)
md.pattern(incomplete_data)
A_imp<-merge_data[!is.na(merge_data$IV)==TRUE|!is.na(merge_data$DV)==TRUE,]
imp<-mice(A_imp, method=”norm.boot”, printFlag=FALSE)
A_Complete<-complete(imp)
md.pattern(A_complete)
M<-lm(DV~IV, data=A_complete)
summary(M)
library(knitr)
kable(anova(M), caption= ‘ANOVA Table’)
plot(A_complete$DV ~ A_complete$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')
confint(M, level = 0.95)
confint(M, level = 0.99)
```

Part B code

```
setwd("~/Desktop")
PartB<-read.csv("354111_PartB.csv")
origin<-lm(I(y)~I(x),data=PartB)
summary(origin)
plot(PartB$y~PartB$x, main= ‘Scatter:y~x’, xlab=x,ylab=y,pch=20)
abline(origin,col='red', lty=3, lwd=2)
```

```
legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
B1<-lm(I(y)~I(x^(-1)),data=PartB)
summary(B1)
B2<-lm(I(y)~I(x^(1/2)),data=PartB)
summary(B2)
B3<-lm(I(y)~I(x^(-1/2)),data=PartB)
summary(B3)
B4<-lm(I(y)~I(x^(2)),data=PartB)
summary(B4)
B5<-lm(I(y)~I(log(x)),data=PartB)
summary(B5)
B6<-lm(I(y^(-1))~I(x),data=PartB)
summary(B6)
B7<-lm(I(y^(-1))~I(x^(-1)),data=PartB)
summary(B7)
B8<-lm(I(y^(-1))~I(x^(-1/2)),data=PartB)
summary(B8)
B9<-lm(I(y^(-1))~I(x^(1/2)),data=PartB)
summary(B9)
B10<-lm(I(y^(-1))~I(x^(2)),data=PartB)
summary(B10)
B11<-lm(I(y^(-1))~I(log(x)),data=PartB)
summary(B11)
B12<-lm(I(y^(-1/2))~I(x),data=PartB)
summary(B12)
B13<-lm(I(y^(-1/2))~I(x^(-1)),data=PartB)
summary(B13)
B14<-lm(I(y^(-1/2))~I(x^(-1/2)),data=PartB)
summary(B14)
B15<-lm(I(y^(-1/2))~I(x^(1/2)),data=PartB)
summary(B15)
B16<-lm(I(y^(-1/2))~I(x^(2)),data=PartB)
summary(B16)
B17<-lm(I(y^(-1/2))~I(log(x)),data=PartB)
summary(B17)
B18<-lm(I(y^(1/2))~I(x),data=PartB)
summary(B18)
```

```
B19<-lm(I(y^(1/2))~I(x^(-1)),data=PartB)
summary(B19)
B20<-lm(I(y^(1/2))~I(x^(-½)),data=PartB)
summary(B20)
B21<-lm(I(y^(1/2))~I(x^(½)),data=PartB)
summary(B21)
B22<-lm(I(y^(1/2))~I(x^(2)),data=PartB))
summary(B22)
B23<-lm(I(y^(1/2))~I(log(x)),data=PartB)
summary(B23)
B24<-lm(I(y^(2))~I(x),data=PartB)
summary(B24)
B25<-lm(I(y^(2))~I(x^(-1)),data=PartB)
summary(B25)
B26<-lm(I(y^(2))~I(x^(-1/2)),data=PartB)
summary(B26)
B27<-lm(I(y^(2))~I(x^(1/2)),data=PartB)
summary(B27)
B28<-lm(I(y^(2))~I(x^(2)),data=PartB)
summary(B28)
B29<-lm(I(y^(2))~I(log(x)),data=PartB)
summary(B29)
B30<-lm(I(log(y))~I(x),data=PartB)
summary(B30)
B31<-lm(I(log(y))~I(x^(-1)),data=PartB)
summary(B31)
B32<-lm(I(log(y))~I(x^(-1/2)),data=PartB)
summary(B32)
B33<-lm(I(log(y))~I(x^(1/2)),data=PartB)
summary(B33)
B34<-lm(I(log(y))~I(x^(2)),data=PartB)
summary(B34)
B35<-lm(I(log(y))~I(log(x)),data=PartB)
summary(B35)
plot(PartB$y^(2)~PartB$x, main='Scatter:y~x’,xlab=x,ylab=y,pch=20)
abline(B24,col= ‘red’, lty=3, lwd=2)
legend('topleft', legend='Estimated Regression Line', lty=3, lwd=2, col='red')
```

```
data_trans <- data.frame(xtrans=data$x, ytrans=data$y^(2))
groups <- cut(data_trans$xtrans,breaks=c(-Inf,seq(min(data_trans$xtrans)+0.3,
max(data_trans$xtrans)-0.3,by=0.3),Inf))
table(groups)
x <- ave(data_trans$xtrans, groups)
data_bin <- data.frame(x=x, y=data_trans$ytrans)
install.packages('olsrr')
library("olsrr")
fit_b<-lm(y~x,data=data_bin)
ols_pure_error_anova(fit_b)
fit_b_final <- lm(ytrans ~ xtrans, data = data_trans)
summary(fit_b_final)
```