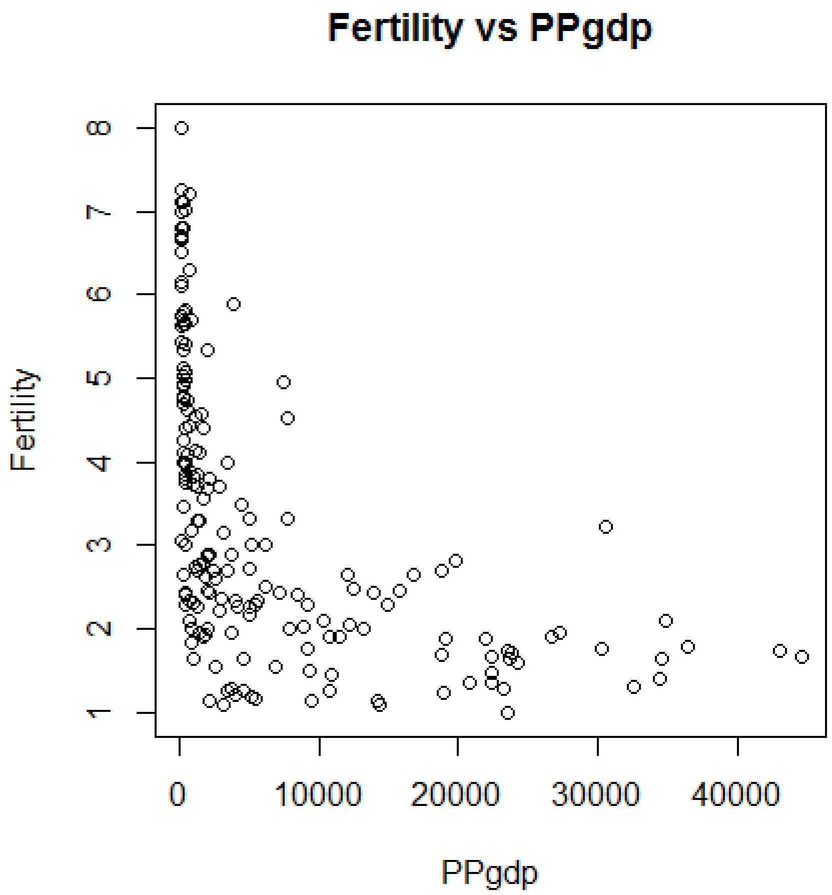
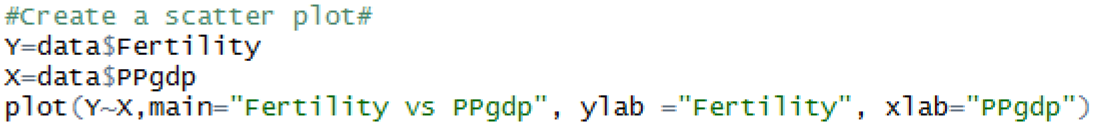
Statistics 108, Project 1

Data visualization and pre-processing:

1.

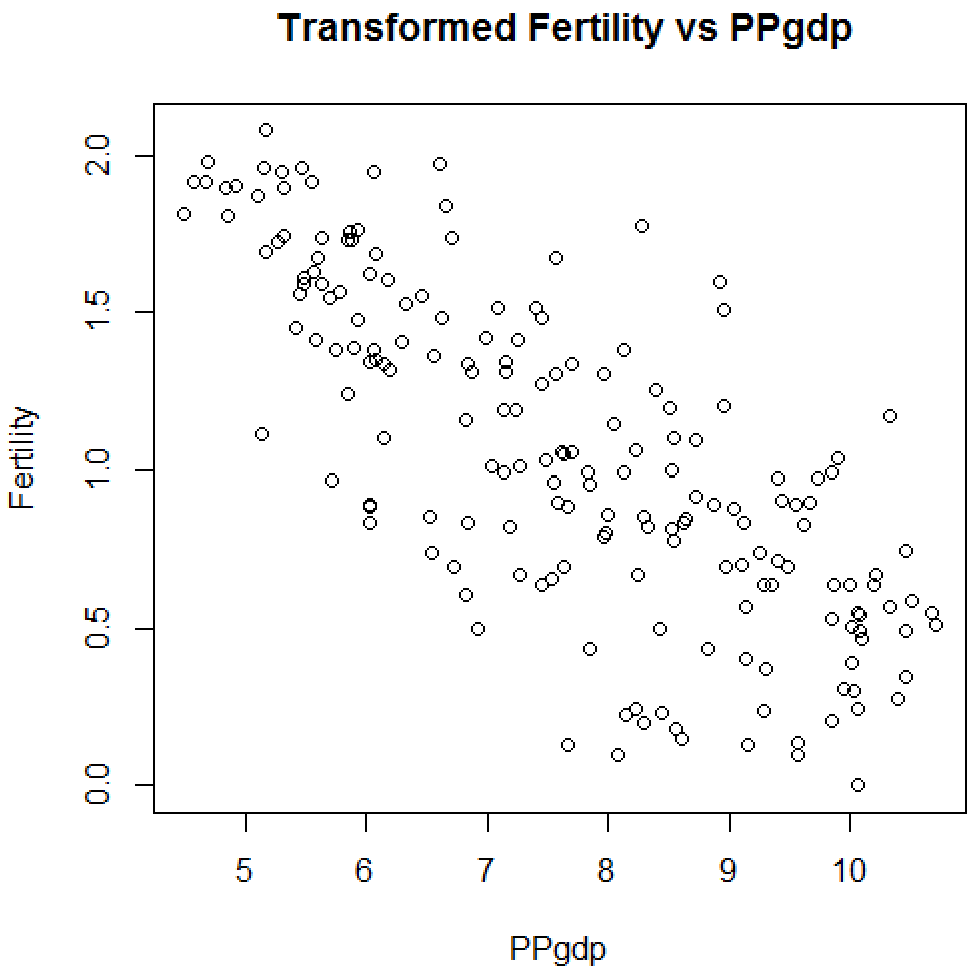




```
#Create a scatter plot#
Y=data$Fertility
X=data$PPgdp
plot(Y~X,main="Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
```

On the scatter plot, the shape of the spread of the data is clearly not linear. Most of the points gather on the left side and concentrated. Thus, a simple linear regression model seems not to be a plausible for a summary for this graph.

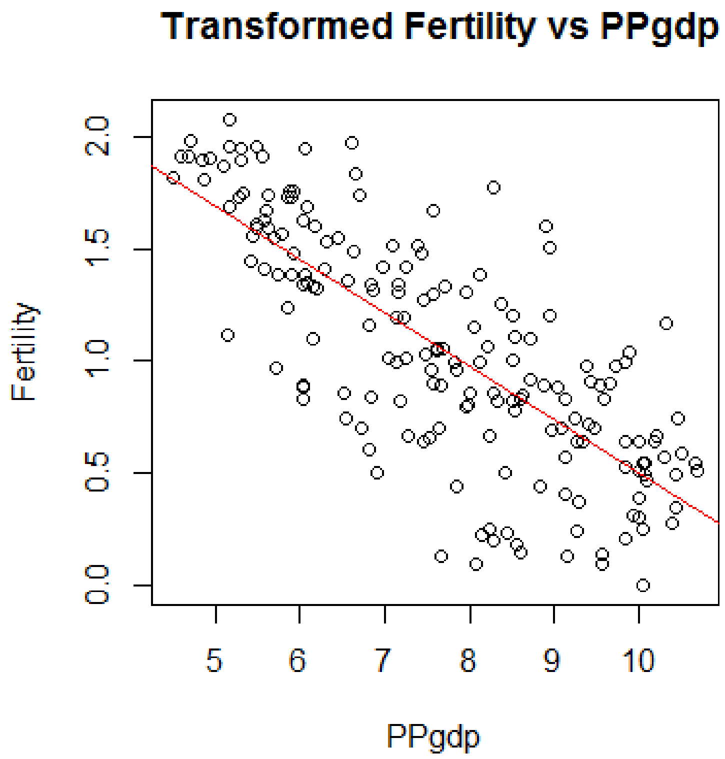
2.The transformations I made on this dataset are log(y) and log(x). Since the equal variance assumption doesn't hold on the original data, I used the boxcox() function to determine the appropriate transformation on Y. The result of the boxcox() function indicates that log(Y) is a good transformation on Y. However, after the transformation on Y, the scatter plot of the data was still not in a good shape. Most of the data points still gathered on the upper left corner. Thus, log(x) is reasonable transformation on X. The following graph is the transformed scatter plot of the dataset.



After the transformations, the points on the scatter plot are more spread out. We can barely see that the shape of the spread of the data is linear with a negative slope. The linearity assumption seems hold and a fitted line with a negative slope seems to be fitted on this plot.

Model fitting and diagnostic:

3.

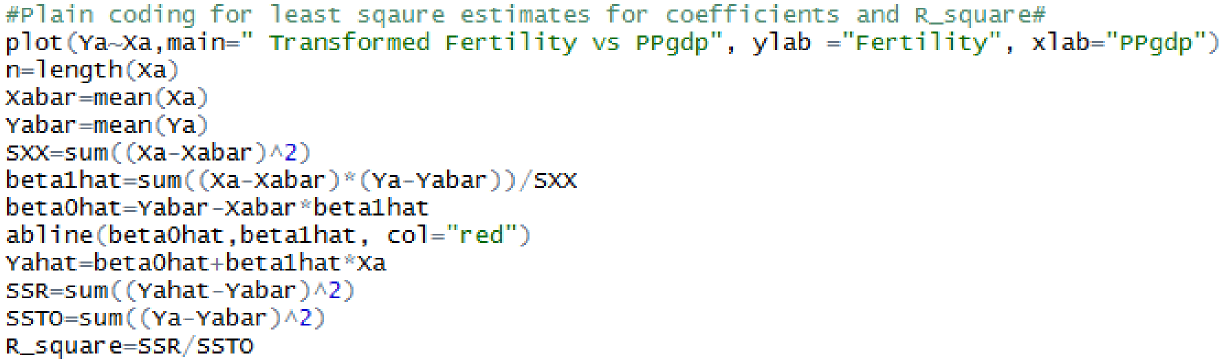


Least square estimator: $\hat{\beta}_1 = -0.237$ $\hat{\beta}_0 = 2.876$

Coefficient of determination: $R^2 = 0.581$

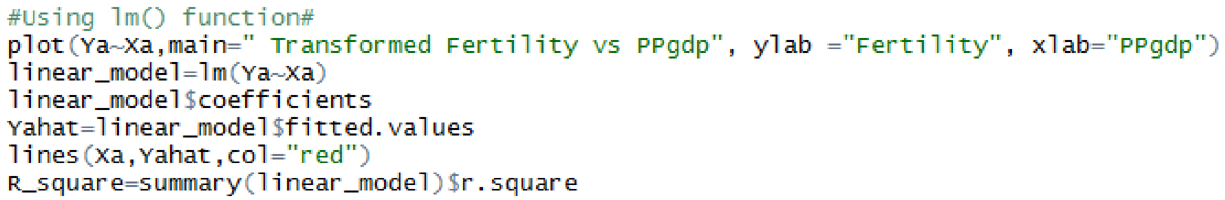
The fitted line goes through the center of the spread of the data. Points are spread roughly even above and below the fitted line; however, R^2 is only about 0.6, so it seems like the data doesn't have a very strong linear relationship.

(a)



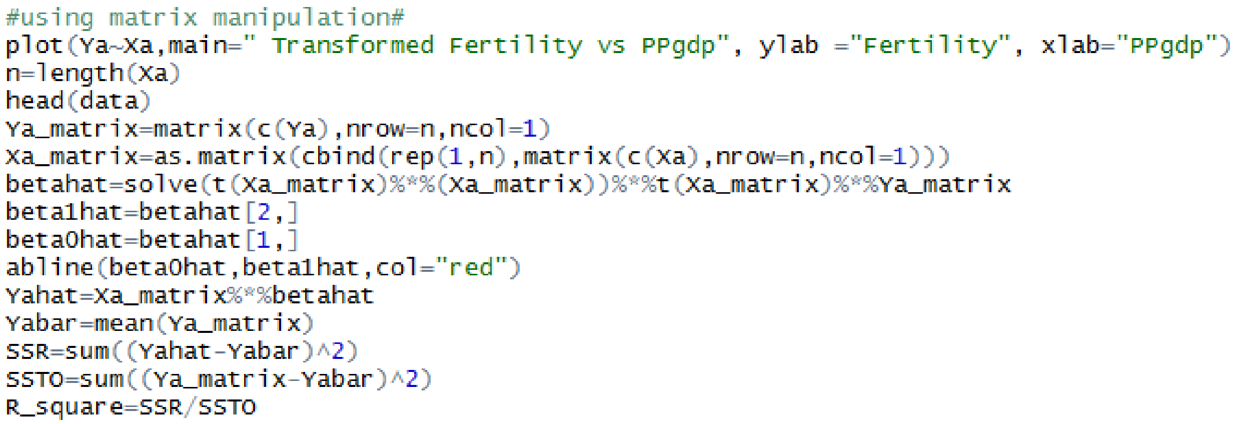
```
#Plain coding for least sqaure estimates for coefficients and R_square#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
n=length(Xa)
Xabar=mean(Xa)
Yabar=mean(Ya)
SXX=sum((Xa-Xabar)^2)
beta1hat=sum((Xa-Xabar)*(Ya-Yabar))/SXX
beta0hat=Yabar-Xabar*beta1hat
abline(beta0hat,beta1hat, col="red")
Yahat=beta0hat+beta1hat*Xa
SSR=sum((Yahat-Yabar)^2)
SSTO=sum((Ya-Yabar)^2)
R_square=SSR/SSTO
```

(b)



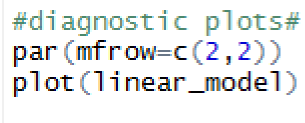
```
#Using lm() function#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
linear_model=lm(Ya~Xa)
linear_model$coefficients
Yahat=linear_model$fitted.values
lines(Xa,Yahat,col="red")
R_square=summary(linear_model)$r.square
```

(c)

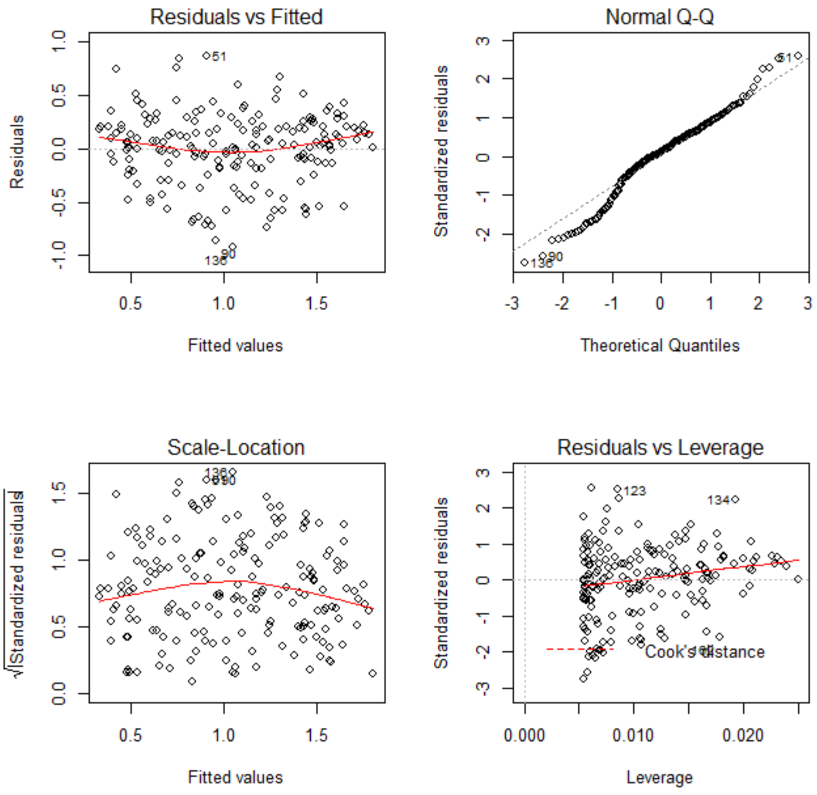


```
#using matrix manipulation#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
n=length(Xa)
head(data)
Ya_matrix=matrix(c(Ya),nrow=n,ncol=1)
Xa_matrix=as.matrix(cbind(rep(1,n),matrix(c(Xa),nrow=n,ncol=1)))
betahat=solve(t(Xa_matrix)%*%(Xa_matrix))%*%t(Xa_matrix)%*%Ya_matrix
beta1hat=betahat[2,]
beta0hat=betahat[1,]
abline(beta0hat,beta1hat,col="red")
Yahat=Xa_matrix%*%betahat
Yabar=mean(Ya_matrix)
SSR=sum((Yahat-Yabar)^2)
SSTO=sum((Ya_matrix-Yabar)^2)
R_square=SSR/SSTO
```

4.



```
#diagnostic plots#
par(mfrow=c(2,2))
plot(linear_model)
```



On the Residuals vs Fitted plot, the reference line is approximately horiztonal at residuals=0, and the points are roughly randomly spread and don't form any particular partern. Thus, the linearity holds on the transfomed data.

On the Noromal Q-Q plot, points are below the reference line at the left end and points are above the reference line at the right end. This indicates that the spread of the standardized residuals has heavy tails at the both ends, so the normality assumption doesn't hold on the transformed data.

On the Scale-Location plot, the reference line is roughly horizontal, so the equal variance is fairly hold on the transformed data.

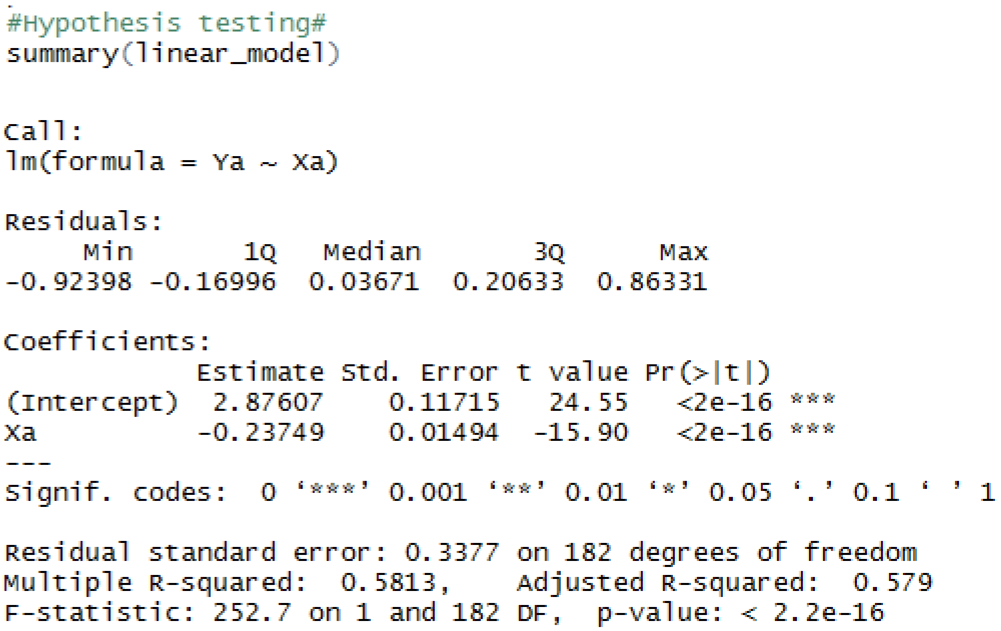
On the Residuals vs Leverage, we don't see any outlier.

Making inferences based on the model:

5.

$$H_0: \hat{\beta}_1 = 0, \qquad \alpha = 0.05$$

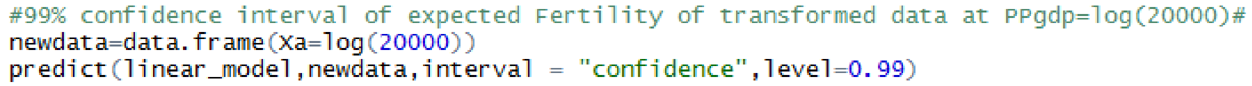
$$H_a: \hat{\beta}_1 \neq 0$$



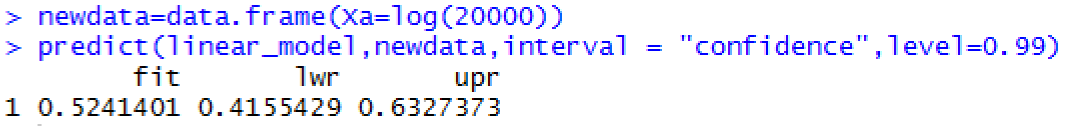
```
#Hypothesis testing#
summary(linear_model)


Call:
lm(formula = Ya ~ Xa)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.92398 -0.16996  0.03671  0.20633  0.86331 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.87607    0.11715   24.55   <2e-16 ***
Xa          -0.23749    0.01494  -15.90   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3377 on 182 degrees of freedom
Multiple R-squared:  0.5813,	Adjusted R-squared:  0.579 
F-statistic: 252.7 on 1 and 182 DF,  p-value: < 2.2e-16
```

In the linear model summary, we can see that the $p - value \ < 0.05$, so we reject the null hypothesis. Hence, there is a significant linear relationship between the transformed variables at $\alpha = 0.05$.

6.



```
#99% confidence interval of expected Fertility of transformed data at PPgdp=log(20000)#
newdata=data.frame(Xa=log(20000))
predict(linear_model,newdata,interval = "confidence",level=0.99)
```



```
> newdata=data.frame(Xa=log(20000))
> predict(linear_model,newdata,interval = "confidence",level=0.99)
        fit       lwr       upr
1 0.5241401 0.4155429 0.6327373
```

99% confidence interval for expected fertility of the transformed data at PPgdp=log (20000) is [0.416,0.633]

Lower bound of the 99% confidence interval of original data

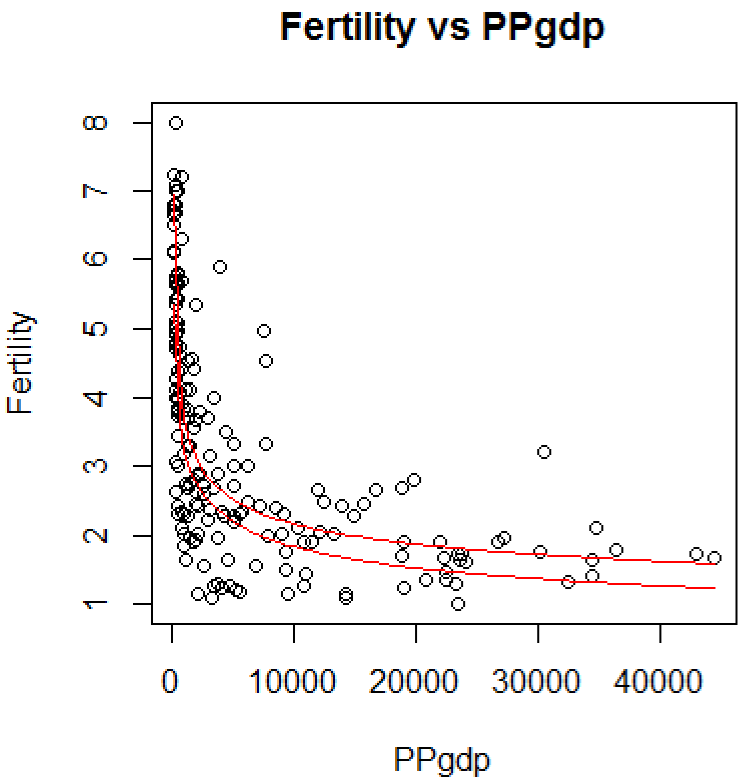
$= e^{0.4155329} = 1.515$

Upper bound of the 99% confidence interval of original data

$= e^{0.6327373} = 1.883$

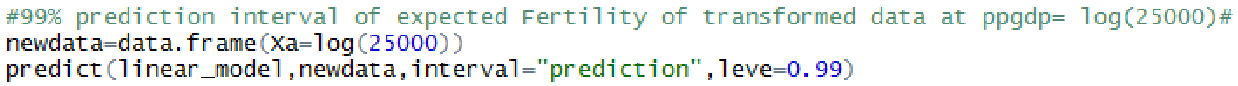
99% confidence interval on the expected Fertility for a region with PPgdp 20,000US dollars in 2001 is [1.515,1.883]

7.

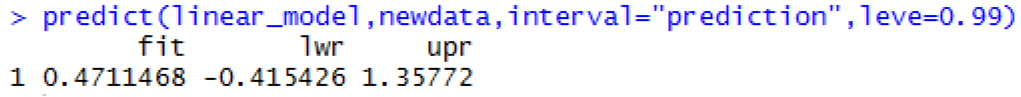


```
#95% confidence band#
plot(Y~X,main="Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
sigmahat=sqrt(sum(residuals(linear_model)^2)/df.residual(linear_model))
W=sqrt(2*qf(0.95,df1 = 2, df2 = n-2))
range=seq(from=min(X),to=max(X),length.out = 100)
upperbound=beta0hat+beta1hat*log(range)+W*sigmahat*sqrt(1/n+(log(range)-Xabar)^2/SXX)
lowerbound=beta0hat+beta1hat*log(range)-W*sigmahat*sqrt(1/n+(log(range)-Xabar)^2/SXX)
lines(exp(upperbound)~range,col="red")
lines(exp(lowerbound)~range,col="red")
```

8.



```
#99% prediction interval of expected Fertility of transformed data at ppgdp= log(25000)#
newdata=data.frame(Xa=log(25000))
predict(linear_model,newdata,interval="prediction",leve=0.99)
```



```
> predict(linear_model,newdata,interval="prediction",leve=0.99)
        fit       lwr     upr
1 0.4711468 -0.415426 1.35772
```

99% prediction interval for future fertility of the transformed data at PPgdp=log (2500) is [-0.4154,1.3578]

Lower bound of the 99% prediction interval of original data

$= e^{-0.415426} = 0.660$

Upper bound of the 99% prediction interval of original data

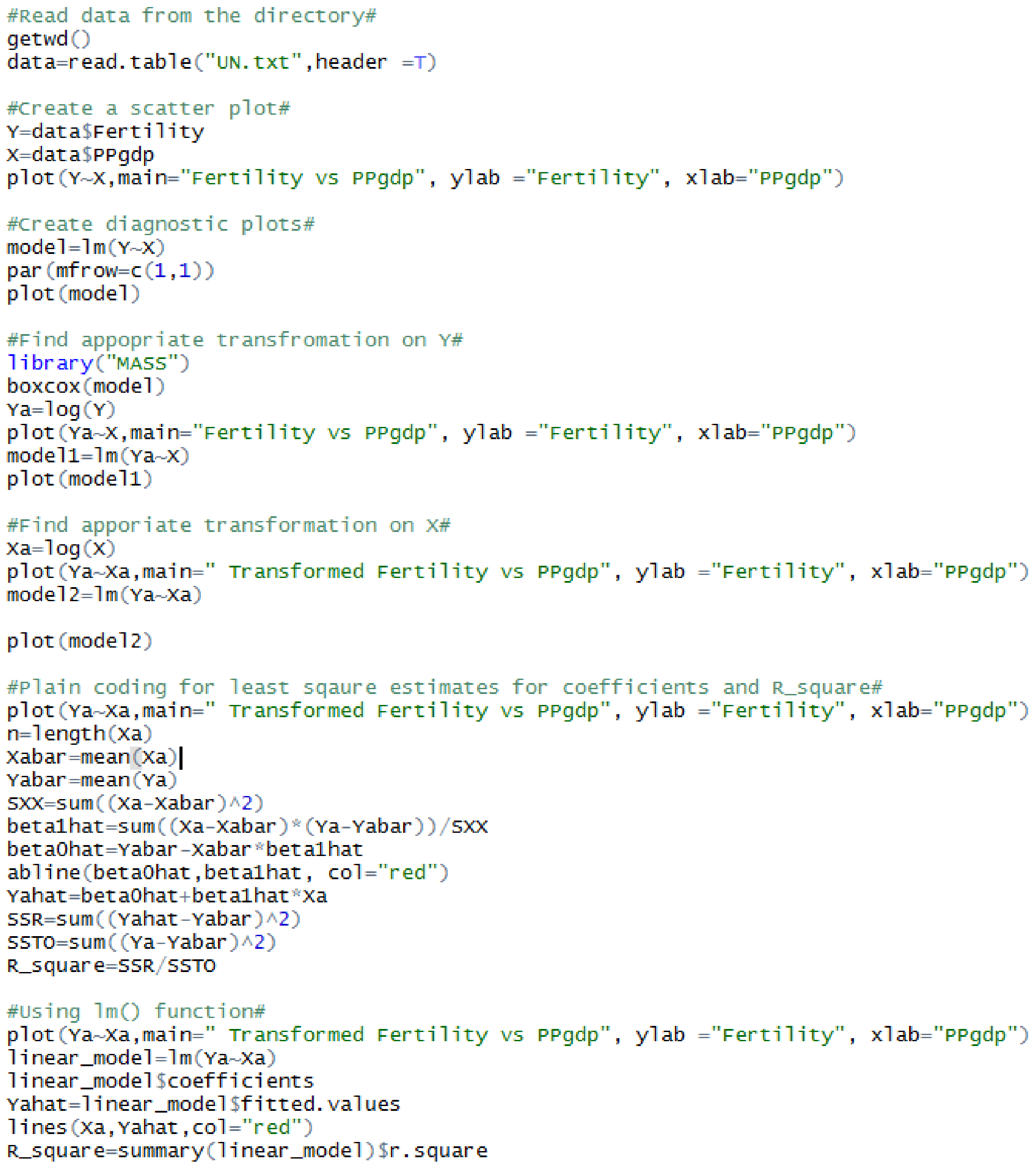
$= e^{1.35772} = 3.887$

99% prediction interval on the future Fertility for a region with PPgdp 25,000 US dollars in 2008 is [0.660,3.887]

9.

Based on the diagnostic plots in Part 4, we know that normality assumption doesn't hold on the transformed data, because the spread of the standardized residuals has two heavy tails. Since inference and hypothesis testing are based on normality assumption, the inferences and the hypothesis testing made on this dataset may not be accurate.

R code



```
#Read data from the directory#
getwd()
data=read.table("UN.txt",header =T)

#Create a scatter plot#
Y=data$Fertility
X=data$PPgdp
plot(Y~X,main="Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")

#Create diagnostic plots#
model=lm(Y~X)
par(mfrow=c(1,1))
plot(model)

#Find appopriate transfromation on Y#
library("MASS")
boxcox(model)
Ya=log(Y)
plot(Ya~X,main="Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
model1=lm(Ya~X)
plot(model1)

#Find apporiate transformation on X#
Xa=log(X)
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
model2=lm(Ya~Xa)

plot(model2)

#Plain coding for least sqaure estimates for coefficients and R_square#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
n=length(Xa)
Xabar=mean(Xa)
Yabar=mean(Ya)
SXX=sum((Xa-Xabar)^2)
beta1hat=sum((Xa-Xabar)*(Ya-Yabar))/SXX
beta0hat=Yabar-Xabar*beta1hat
abline(beta0hat,beta1hat, col="red")
Yahat=beta0hat+beta1hat*Xa
SSR=sum((Yahat-Yabar)^2)
SSTO=sum((Ya-Yabar)^2)
R_square=SSR/SSTO

#Using lm() function#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
linear_model=lm(Ya~Xa)
linear_model$coefficients
Yahat=linear_model$fitted.values
lines(Xa,Yahat,col="red")
R_square=summary(linear_model)$r.square
```

```
#using matrix manipulation#
plot(Ya~Xa,main=" Transformed Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
n=length(Xa)
head(data)
Ya_matrix=matrix(c(Ya),nrow=n,ncol=1)
Xa_matrix=as.matrix(cbind(rep(1,n),matrix(c(Xa),nrow=n,ncol=1)))
betahat=solve(t(Xa_matrix)%*%(Xa_matrix))%*%t(Xa_matrix)%*%Ya_matrix
beta1hat=betahat[2,]
beta0hat=betahat[1,]
abline(beta0hat,beta1hat,col="red")
Yahat=Xa_matrix%*%betahat
Yabar=mean(Ya_matrix)
SSR=sum((Yahat-Yabar)^2)
SSTO=sum((Ya_matrix-Yabar)^2)
R_square=SSR/SSTO

#diagnostic plots#
par(mfrow=c(2,2))
plot(linear_model)

#Hypothesis testing#
summary(linear_model)

#99% confidence interval of expected Fertility of transformed data at PPgdp=log(20000)#
newdata=data.frame(Xa=log(20000))
predict(linear_model,newdata,interval = "confidence",level=0.99)

#95% confidence band#
plot(Y~X,main="Fertility vs PPgdp", ylab ="Fertility", xlab="PPgdp")
sigmahat=sqrt(sum(residuals(linear_model)^2)/df.residual(linear_model))
W=sqrt(2*qf(0.95,df1 = 2, df2 = n-2))
range=seq(from=min(X),to=max(X),length.out = 100)
upperbound=beta0hat+beta1hat*log(range)+W*sigmahat*sqrt(1/n+(log(range)-Xabar)^2/SXX)
lowerbound=beta0hat+beta1hat*log(range)-W*sigmahat*sqrt(1/n+(log(range)-Xabar)^2/SXX)
lines(exp(upperbound)~range,col="red")
lines(exp(lowerbound)~range,col="red")

#99% prediction interval of expected Fertility of transformed data at ppgdp= log(25000)#
newdata=data.frame(Xa=log(25000))
predict(linear_model,newdata,interval="prediction",leve=0.99)
```