**Introduction:**

The objective of this Individual project is to build a predictive model or algorithm of the response Y based on the predictors X1 through X140. The Data.train file has 5200 observations and 141 variables. The Data.test file also has 5200 observations on the same 140 predictors (not including the Y response). We have no backstory of this large dataset, only to try to run a prediction model to get the best MSE value. MSE between 4 and 10 seems to be a good value. We will perform some exploratory data analysis and look at residual plots to determine if the data meets normal assumptions or if a higher order is needed or transformation is needed.

**Exploratory Data Analysis**

When looking at the summary of Datatrain dataframe most of the values are continuous values. When plotting a graph to see what the range and minimum and maximum values of all the predictors, the minimum values are positive. The range values are around 70s and the maximum values are in 260s (Figure1A1.1). Now we want to see what the linear model regression looks like. Usually, we want to check the normality plot to see how well the data points fall on the line and they should fall closely. Also the residuals vs fitted plot should show data points randomly falling along 0 with no pattern. In Figure2A1.2, we see there is a slight S curve in the normality plot and we see a point that looks like an outlier. Also we see that the residuals vs fitted plot shows a curvilinear pattern and that is no good.

Let's look at a plot to check where the outlier threshold should be and whether the points in our data are within the limits. I made a plot of alpha values between 0.05 and 0.01 and 0.001. For a 95% CI level, the outlier threshold falls within 4 and 5 and at 99% CI level, the outlier threshold falls around 4.8 to below 6 (Figure3A1.3). Looking at the residuals vs leverage plot from Figure 2A1.2, I see that majority of the data points are indeed below 6 and there is one dataset that is at residual value 10. This isn't much of a concern. Next I want to check for any correlation within the variables, multicollinearity can cause problems if there are many predictors that are highly correlated. Figure4A1.4 shows that there is a multicollinearity.

**Pre-Processing**

The stepwise selection procedure helps to reduce the number of predictors and select those that are most significant. This procedure works by building a regression model that includes all predictor variables that are statistically significant comparing to the response variable (Y) in the Data.train file. Figure 5.A1.5 shows the plot of the Stepwise model which still doesn't fix the curvilinear pattern in the residuals vs fits and the MSE is over 300,000. Adding quadratic terms Figure6.A1.6 improved the curvilinear pattern in the residuals vs fits, but the normality plot still needs improvement. The MSE using quadratic terms is 45.609. When I use squared root terms, the normal plot improved and the residual vs fits also display a more constant variance pattern (Figure 7.A1.7). The MSE for the squared root model is 4.499. Final model to try is to add inverse terms, in Figure 8.A1.8, we see that the normality plot improves even more. The MSE for adding inverse terms is 4.038. Using cross validation will help check if we are overfitting with too many variables (Figure 9.A1.9). In Figure 9.A1.9 we see that the MSE is too high for the first two models (with 100 predictors and even with 200 predictors). This confirms that the models using squared root or inverse terms give the best MSE.

## Conclusion

I chose include 3 polynomials in the final formula to calculate the final predictions. This was an example of a nonlinear model and that could be fixed to become linear model by transforming the variables. To visualize if the data meets normal assumptions we look at the residual plots and normality plot. My regression model is

$$f(x) = x + x^2 + x^{\frac{1}{2}} + \ x^{-1}$$

I only included the significant predictors that I have found using stepwise procedure on my full model. I then I predict my MSE will be around 6 even though when I checked the error for the final train model it was around 4. I know there are many more fractional polynomials I could have tried. I decided to go with these because when checking the residual plots, adding the quadratic terms already helped eliminate the curvilinear pattern seen in the residual vs fitted plots.

## Appendix

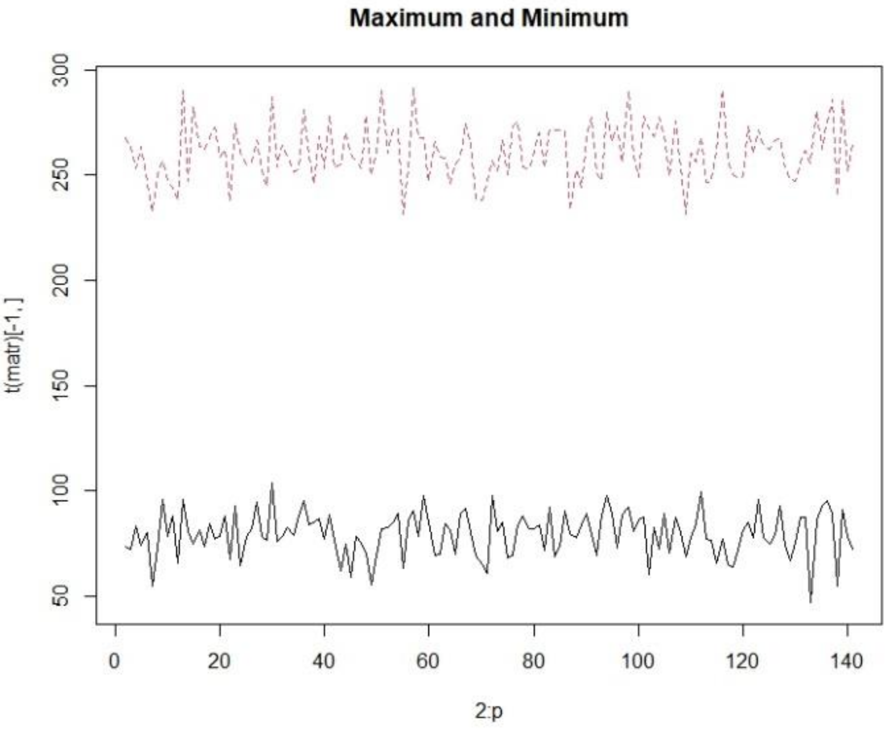


*Figure 1A1.1*

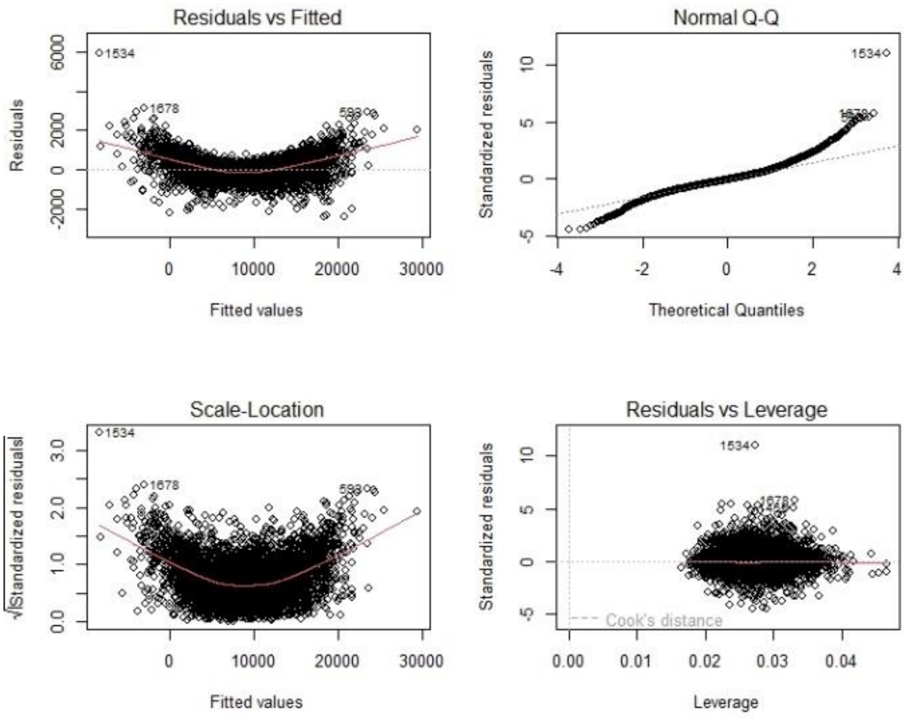
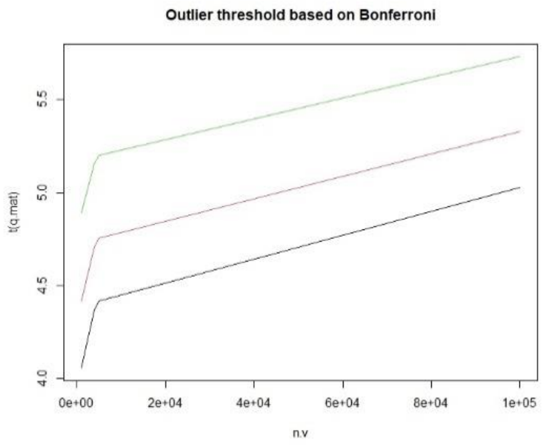


*Figure 2A1.2*



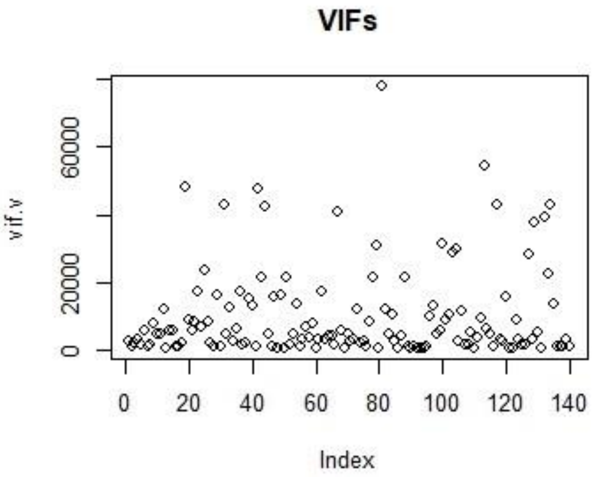


*Figure 3A1.3*

*Figure 4A1.4*

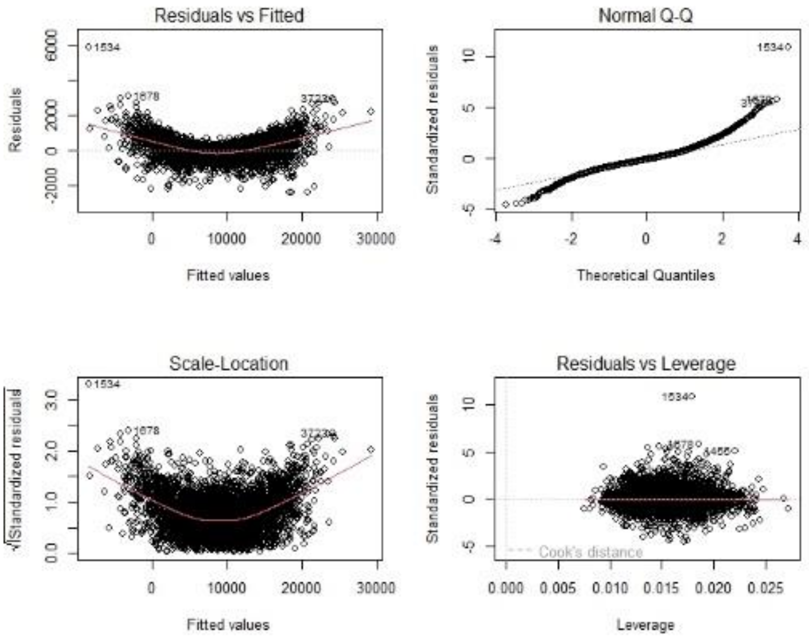


*Figure5.A1.5*

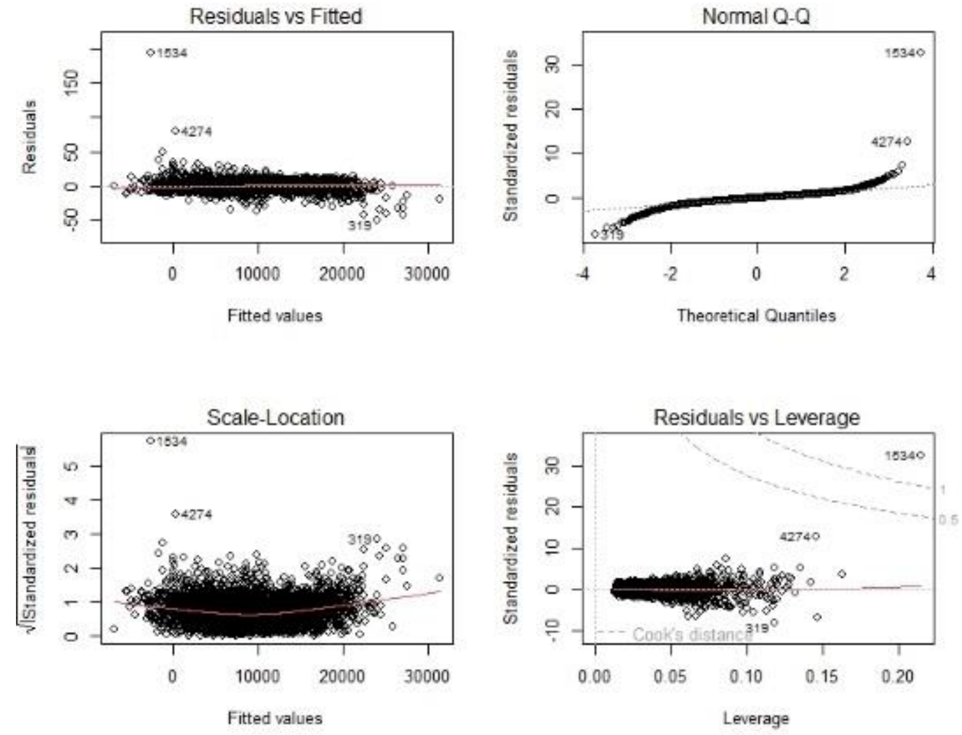


*Figure 6.A1.6*

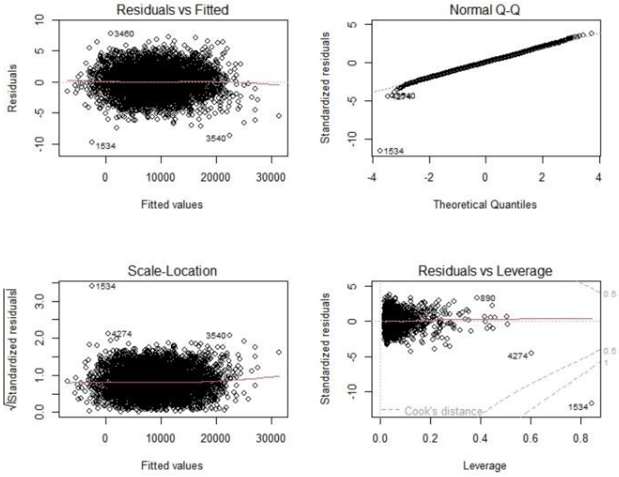


*Figure7.A1.7*

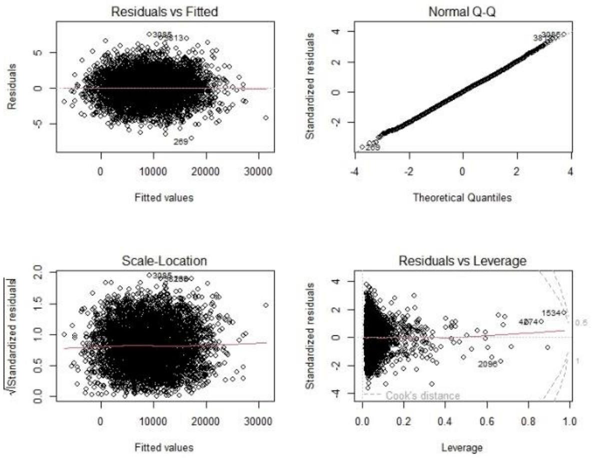


*Figure8.A1.8*

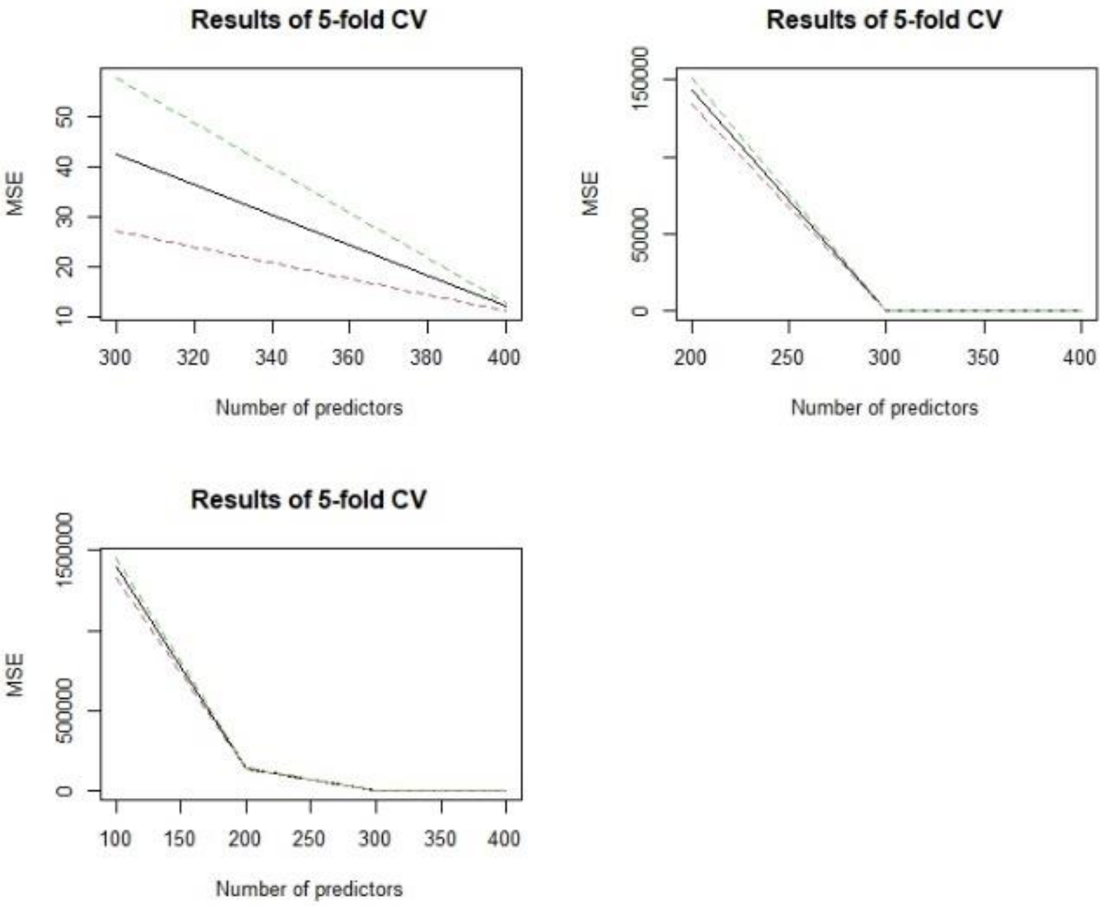
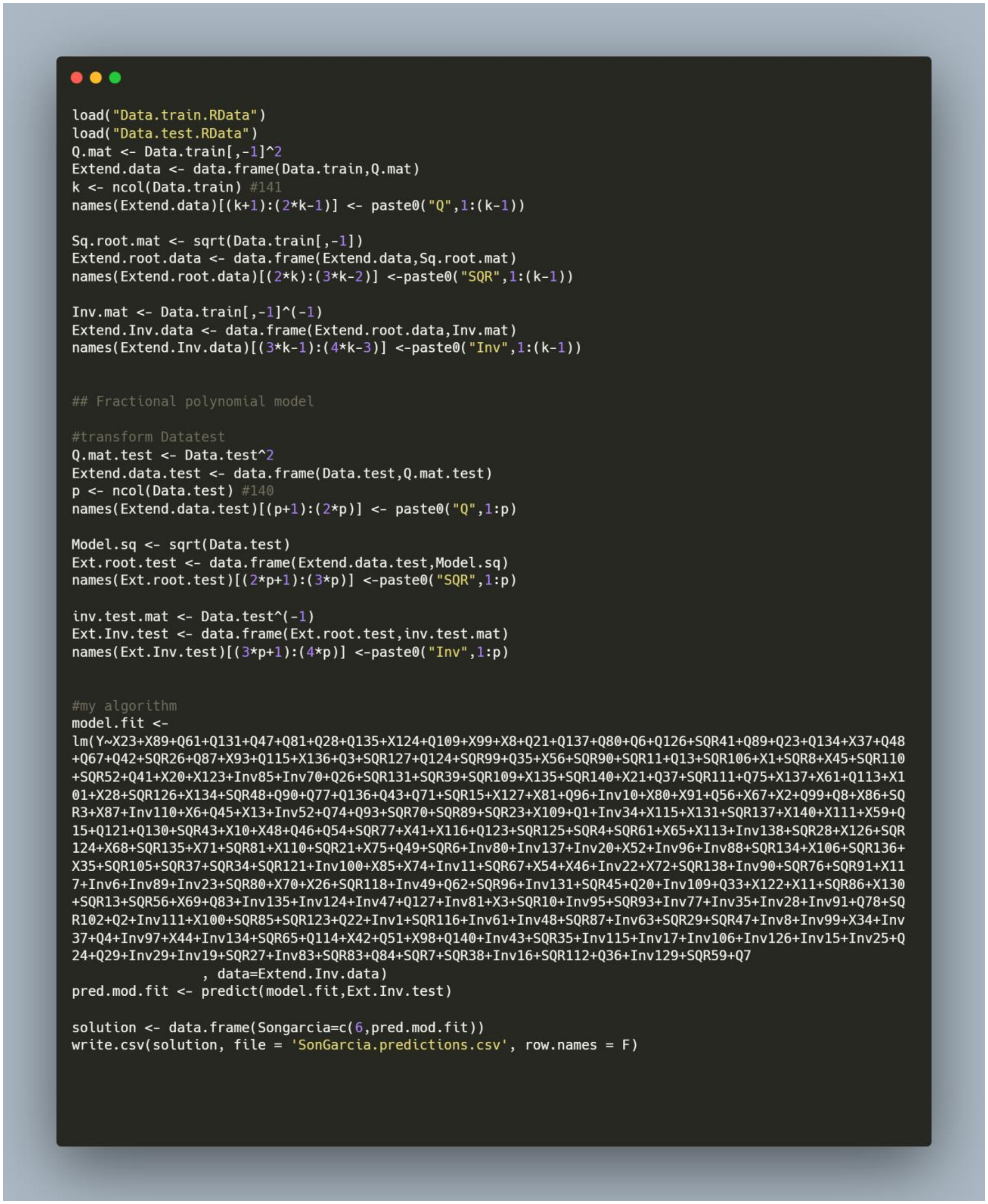


*Figure 9.A1.9*

R Coding:



```r
load("Data.train.RData")
load("Data.test.RData")
Q.mat <- Data.train[,-1]^2
Extend.data <- data.frame(Data.train,Q.mat)
k <- ncol(Data.train) #141
names(Extend.data)[(k+1):(2*k-1)] <- paste0("Q",1:(k-1))

Sq.root.mat <- sqrt(Data.train[,-1])
Extend.root.data <- data.frame(Extend.data,Sq.root.mat)
names(Extend.root.data)[(2*k):(3*k-2)] <-paste0("SQR",1:(k-1))

Inv.mat <- Data.train[,-1]^(-1)
Extend.Inv.data <- data.frame(Extend.root.data,Inv.mat)
names(Extend.Inv.data)[(3*k-1):(4*k-3)] <-paste0("Inv",1:(k-1))


## Fractional polynomial model

#transform Datatest
Q.mat.test <- Data.test^2
Extend.data.test <- data.frame(Data.test,Q.mat.test)
p <- ncol(Data.test) #140
names(Extend.data.test)[(p+1):(2*p)] <- paste0("Q",1:p)

Model.sq <- sqrt(Data.test)
Ext.root.test <- data.frame(Extend.data.test,Model.sq)
names(Ext.root.test)[(2*p+1):(3*p)] <-paste0("SQR",1:p)

inv.test.mat <- Data.test^(-1)
Ext.Inv.test <- data.frame(Ext.root.test,inv.test.mat)
names(Ext.Inv.test)[(3*p+1):(4*p)] <-paste0("Inv",1:p)


#my algorithm
model.fit <-
lm(Y~X23+X89+Q61+Q131+Q47+Q81+Q28+Q135+X124+Q109+X99+X8+Q21+Q137+Q80+Q6+Q126+SQR41+Q89+Q23+Q134+X37+Q48
+Q67+Q42+SQR26+Q87+X93+Q115+X136+Q3+SQR127+Q124+SQR99+Q35+X56+SQR90+SQR11+Q13+SQR106+X1+SQR8+X45+SQR110
+SQR52+Q41+X20+X123+Inv85+Inv70+Q26+SQR131+SQR39+SQR109+X135+SQR140+X21+Q37+SQR111+Q75+X137+X61+Q113+X1
01+X28+SQR126+X134+SQR48+Q90+Q77+Q136+Q43+Q71+SQR15+X127+X81+Q96+Inv10+X80+X91+Q56+X67+X2+Q99+Q8+X86+SQ
R3+X87+Inv110+X6+Q45+X13+Inv52+Q74+Q93+SQR70+SQR89+SQR23+X109+Q1+Inv34+X115+X131+SQR137+X140+X111+X59+Q
15+Q121+Q130+SQR43+X10+X48+Q46+Q54+SQR77+X41+X116+Q123+SQR125+SQR4+SQR61+X65+X113+Inv138+SQR28+X126+SQR
124+X68+SQR135+X71+SQR81+X110+SQR21+X75+Q49+SQR6+Inv80+Inv137+Inv20+X52+Inv96+Inv88+SQR134+X106+SQR136+
X35+SQR105+SQR37+SQR34+SQR121+Inv100+X85+X74+Inv11+SQR67+X54+X46+Inv22+X72+SQR138+Inv90+SQR76+SQR91+X11
7+Inv6+Inv89+Inv23+SQR80+X70+X26+SQR118+Inv49+Q62+SQR96+Inv131+SQR45+Q20+Inv109+Q33+X122+X11+SQR86+X130
+SQR13+SQR56+X69+Q83+Inv135+Inv124+Inv47+Q127+Inv81+X3+SQR10+Inv95+SQR93+Inv77+Inv35+Inv28+Inv91+Q78+SQ
R102+Q2+Inv111+X100+SQR85+SQR123+Q22+Inv1+SQR116+Inv61+Inv48+SQR87+Inv63+SQR29+SQR47+Inv8+Inv99+X34+Inv
37+Q4+Inv97+X44+Inv134+SQR65+Q114+X42+Q51+X98+Q140+Inv43+SQR35+Inv115+Inv17+Inv106+Inv126+Inv15+Inv25+Q
24+Q29+Inv29+Inv19+SQR27+Inv83+SQR83+Q84+SQR7+SQR38+Inv16+SQR112+Q36+Inv129+SQR59+Q7
                , data=Extend.Inv.data)
pred.mod.fit <- predict(model.fit,Ext.Inv.test)

solution <- data.frame(Songarcia=c(6,pred.mod.fit))
write.csv(solution, file = 'SonGarcia.predictions.csv', row.names = F)
```