

LMR_PGPT

Pramod Verma

03 May 2018

Loading all the required packages

```
#install.packages(c("ggplot2", "reshape2", "lattice", "corrplot"))
library(ggplot2) ## Grammar of graphics
#library(reshape2) ## Reshaping data frames
#library(lattice) ## More graphics
library(corrplot) ## For plotting the correlations
```

Reading the file from the computer, change the path accordingly

GDPE is a transformed variable: GDP + Export

```
file = "D:/Projects/Global Container Throughput Index/Data/Final_Pred.csv"
ModelData = read.csv(
  file = file,
  header = TRUE,
  colClasses = c("factor", "numeric", "numeric", "numeric", "numeric",
"numeric", "numeric", "numeric", "numeric", "numeric", "numeric"))

# To remove the decimal places from the year values
ModelData$Period <- as.factor(as.integer(as.character(ModelData$Period)))

# End of the file reading into the memory
```

Check the dataset if everything is correct

```
# Checking the structure of the input data
str(ModelData)

## 'data.frame':    20 obs. of  11 variables:
## $ Period: Factor w/ 20 levels "2000","2001",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ GDP   : num  50030 50988 52089 53602 55986 ...
## $ CPI   : num  69.1 72.1 74.5 77.1 79.9 ...
## $ Oil   : num  30.4 26 26.2 31.1 41.5 ...
## $ AirLog: num  263 342 312 374 453 ...
## $ GDPE  : num  56486 57181 58587 61190 65211 ...
## $ BDI   : num  1599 876 1738 4765 4598 ...
## $ PMI   : num  43.9 45.3 51.6 60.1 57.2 54.7 53.1 52.7 37.9 56.2 ...
## $ Stock : num  2471 1950 1336 2003 2175 ...
## $ Export: num  6456 6193 6498 7588 9224 ...
## $ PGPT  : num  235 246 273 315 362 ...
```

```
# Checking the first five rows
```

```
head(ModelData, 5)
```

```
##      Period      GDP      CPI      Oil AirLog      GDPE      BDI      PMI      Stock
## 1    2000 50030.24 69.14326 30.38 263.17 56486.41 1599 43.9 2470.52
## 2    2001 50988.04 72.10294 25.98 341.67 57181.36  876 45.3 1950.40
## 3    2002 52088.69 74.47020 26.19 312.10 58586.59 1738 51.6 1335.51
## 4    2003 53601.90 77.07507 31.08 374.18 61189.74 4765 60.1 2003.37
## 5    2004 55986.22 79.87456 41.51 453.08 65210.62 4598 57.2 2175.44
##      Export      PGPT
## 1 6456.168 234.6
## 2 6193.314 246.2
## 3 6497.902 272.8
## 4 7587.840 315.0
## 5 9224.393 361.8
```

```
# Summary of the input data
```

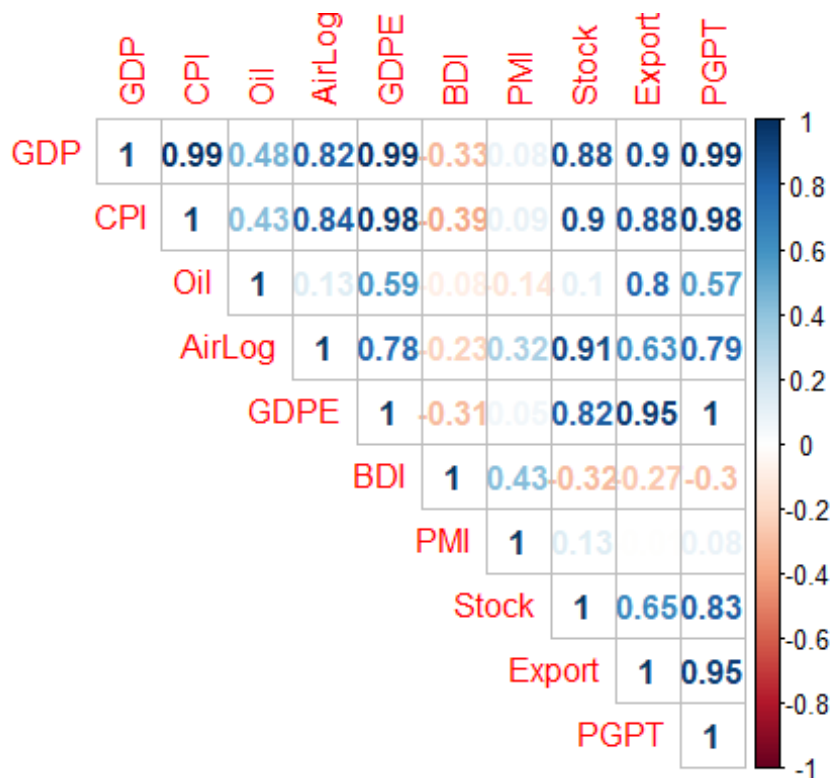
```
summary(ModelData)
```

```
##      Period      GDP      CPI      Oil
## 2000 : 1   Min.   :50030   Min.   : 69.14   Min.   :25.98
## 2001 : 1   1st Qu.:57594   1st Qu.: 82.15   1st Qu.:43.06
## 2002 : 1   Median :65151   Median : 98.25   Median :60.86
## 2003 : 1   Mean    :66162   Mean    :101.55   Mean    :61.96
## 2004 : 1   3rd Qu.:74156   3rd Qu.:119.70   3rd Qu.:82.90
## 2005 : 1   Max.    :85543   Max.    :142.59   Max.    :99.67
## (Other):14
##      AirLog      GDPE      BDI      PMI
## Min.   :263.2   Min.   : 56486   Min.   : 478.0   Min.   :37.90
## 1st Qu.:366.1   1st Qu.: 67783   1st Qu.: 939.8   1st Qu.:51.06
## Median :416.7   Median : 80885   Median :1668.5   Median :52.50
## Mean    :456.0   Mean    : 80081   Mean    :2288.9   Mean    :51.75
## 3rd Qu.:564.6   3rd Qu.: 92339   3rd Qu.:2556.5   3rd Qu.:54.55
## Max.    :754.3   Max.    :103800   Max.    :9143.0   Max.    :60.10
##
##      Stock      Export      PGPT
## Min.   :1336   Min.   : 6193   Min.   :234.6
## 1st Qu.:2198   1st Qu.:10189   1st Qu.:380.2
## Median :2629   Median :15665   Median :524.5
## Mean    :3473   Mean    :13921   Mean    :514.1
## 3rd Qu.:4804   3rd Qu.:18322   3rd Qu.:662.4
## Max.    :7279   Max.    :18968   Max.    :778.5
##
##                                     NA's :1
```

```
# Checking the correlation
```

```
m <- cor(na.omit(ModelData[, -1]))
```

```
corrplot(m, method = "number", type="upper")
```



Defining the training and forecast data

```
# training data
```

```
mt <- ModelData[1:18, -1]
```

```
# forecast data of predictors
```

```
mf <- ModelData[19, -1]
```

```
# Training Data
```

```
mt
```

```
##      GDP      CPI      Oil      AirLog      GDPE      BDI      PMI      Stock      Export
## 1  50030.24  69.14326  30.38  263.17  56486.41  1599  43.9  2470.520  6456.168
## 2  50988.04  72.10294  25.98  341.67  57181.36   876  45.3  1950.400  6193.314
## 3  52088.69  74.47020  26.19  312.10  58586.59  1738  51.6  1335.510  6497.902
## 4  53601.90  77.07507  31.08  374.18  61189.74  4765  60.1  2003.370  7587.840
## 5  55986.22  79.87456  41.51  453.08  65210.62  4598  57.2  2175.440  9224.393
## 6  58130.52  82.90646  56.64  416.19  68640.67  2407  54.7  2205.320  10510.144
## 7  60633.42  85.89470  66.05  422.82  72764.59  4397  53.1  2415.290  12131.177
## 8  63197.03  89.09787  72.34  385.68  77223.31  9143  52.7  2652.280  14026.273
## 9  64346.30  94.25150  99.67  301.34  80511.62   774  37.9  1577.030  16165.311
## 10 63230.51  96.50929  61.95  331.57  75791.14  3005  56.2  2269.150  12560.635
## 11 65956.67  99.99999  79.48  417.17  81257.75  1773  55.3  2652.870  15301.078
## 12 68054.68 105.11726  94.88  393.90  86392.66  1738  51.7  2605.150  18337.982
## 13 69761.54 109.56747  94.05  401.11  88257.52   699  50.2  3019.514  18495.980
## 14 71585.71 114.33436  97.98  556.44  90538.46  2277  53.5  4176.590  18952.750
## 15 73630.02 118.64702  93.17  620.50  92597.72   782  52.4  4736.054  18967.697
```

```
## 16 75733.59 122.86898 48.72 539.23 92252.53 478 49.1 5007.412 16518.941
## 17 77631.27 127.06188 43.58 650.72 93659.97 961 52.6 5383.117 16028.701
## 18 80077.83 131.98427 50.84 754.27 97807.78 1366 54.5 6903.389 17729.953
##      PGPT
## 1  234.6
## 2  246.2
## 3  272.8
## 4  315.0
## 5  361.8
## 6  398.7
## 7  442.2
## 8  497.8
## 9  524.5
## 10 471.2
## 11 550.5
## 12 595.1
## 13 622.3
## 14 644.4
## 15 680.3
## 16 683.5
## 17 701.3
## 18 746.7

#Forecast Data for independent variables
mf

##      GDP      CPI  Oil AirLog      GDPE  BDI  PMI      Stock  Export
## 19 83040.71 137.4369 64.9 589.28 101462.1 1271 51.7 6635.277 18421.42
##      PGPT
## 19 778.5
```

Building Models with different combination of variables: All Variables

```
# Selecting all the independent variables
```

```
fit_all <- lm(PGPT ~ ., data = mt)
```

```
# Keeping fit_all in a common variable for repeatability of code
```

```
fit <- fit_all
```

```
#Checking the summary
```

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = PGPT ~ ., data = mt)
```

```
##
```

```
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
```

```
## -6.576 -3.324  0.582  2.576  6.219
```

```
##
```

```
## Coefficients: (1 not defined because of singularities)
```

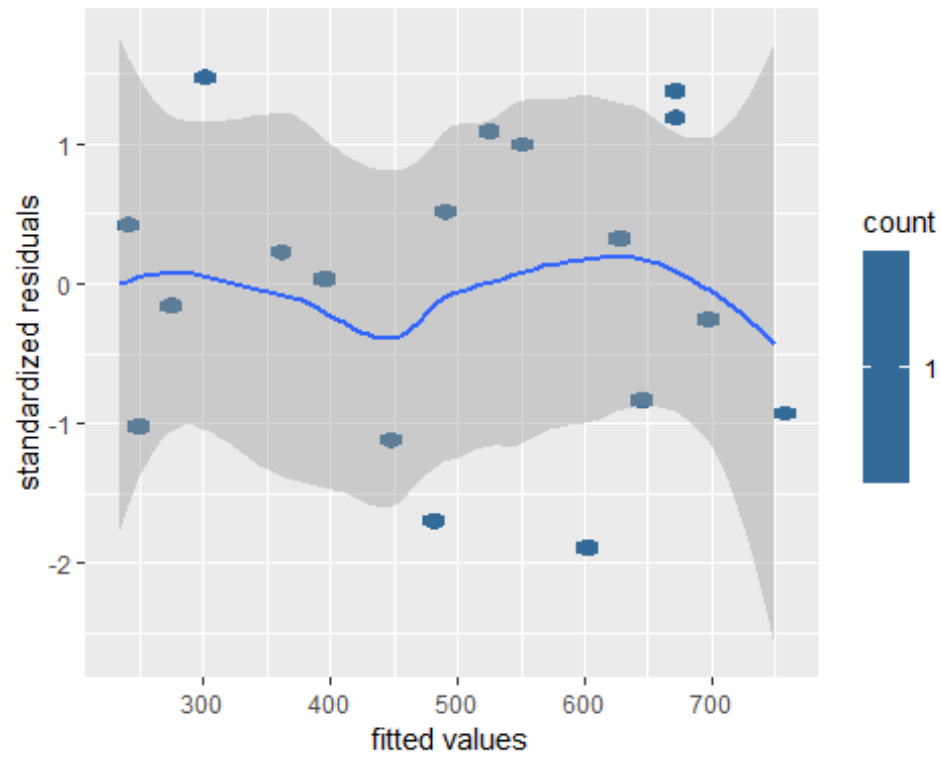
```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.434e+02  5.657e+01  -9.606    5e-06 ***
## GDP         1.834e-03  4.656e-03   0.394   0.70285
## CPI        -2.020e+00  1.197e+00  -1.688   0.12564
## Oil        -4.850e-01  3.457e-01  -1.403   0.19416
## AirLog      5.035e-02  3.537e-02   1.424   0.18832
## GDPE       1.428e-02  3.774e-03   3.783   0.00433 **
## BDI       -1.150e-03  1.359e-03  -0.846   0.41922
## PMI        8.178e-01  4.550e-01   1.797   0.10584
## Stock     -5.693e-03  3.967e-03  -1.435   0.18508
## Export           NA           NA      NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.253 on 9 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.999
## F-statistic: 2127 on 8 and 9 DF,  p-value: 8.011e-14

drop1(fit, test = "F")

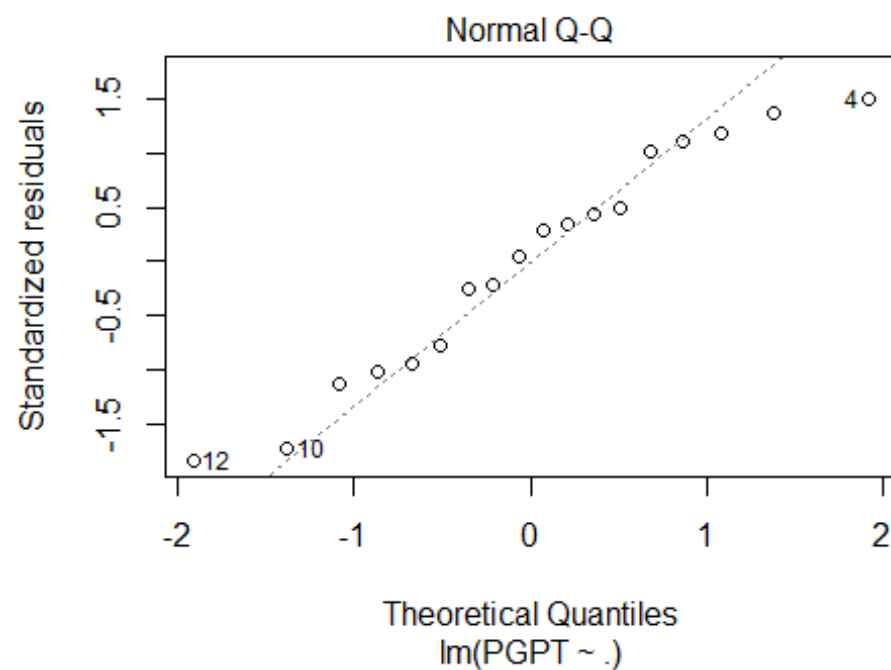
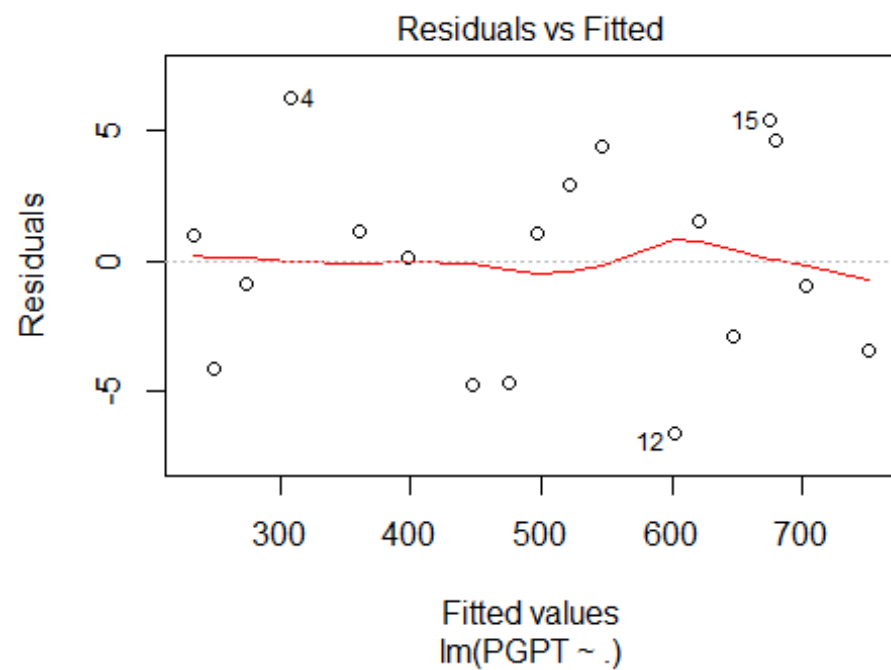
## Single term deletions
##
## Model:
## PGPT ~ GDP + CPI + Oil + AirLog + GDPE + BDI + PMI + Stock +
##      Export
##      Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                248.33 65.239
## GDP      0      0.000 248.33 65.239
## CPI      1    78.641 326.97 68.191  2.8501 0.1256
## Oil      1    54.312 302.64 66.799  1.9684 0.1942
## AirLog   1    55.913 304.25 66.894  2.0264 0.1883
## GDPE     0      0.000 248.33 65.239
## BDI      1    19.771 268.10 64.618  0.7165 0.4192
## PMI      1    89.134 337.47 68.760  3.2304 0.1058
## Stock    1    56.829 305.16 66.948  2.0596 0.1851
## Export   0      0.000 248.33 65.239

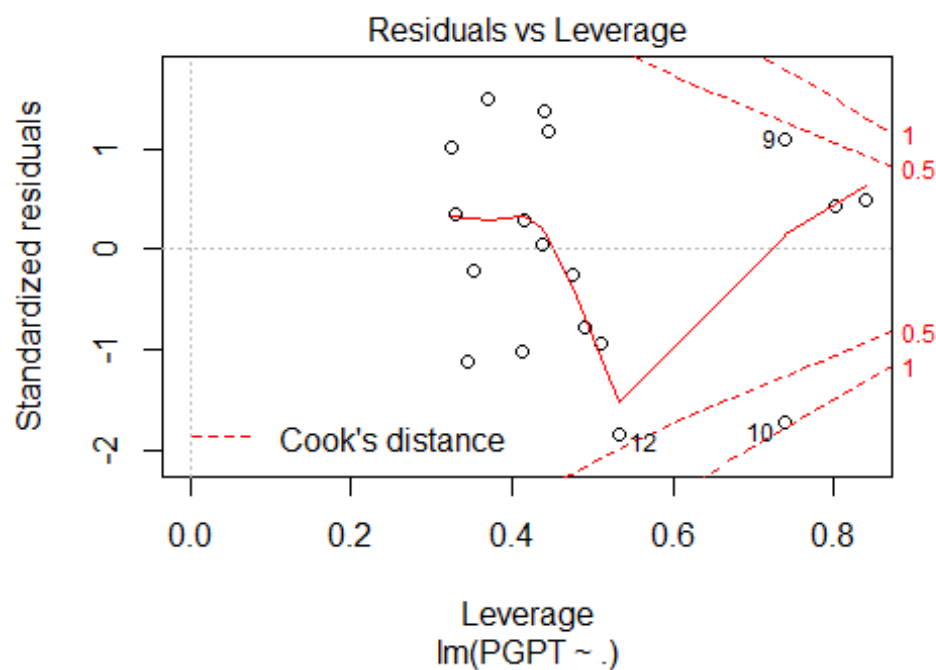
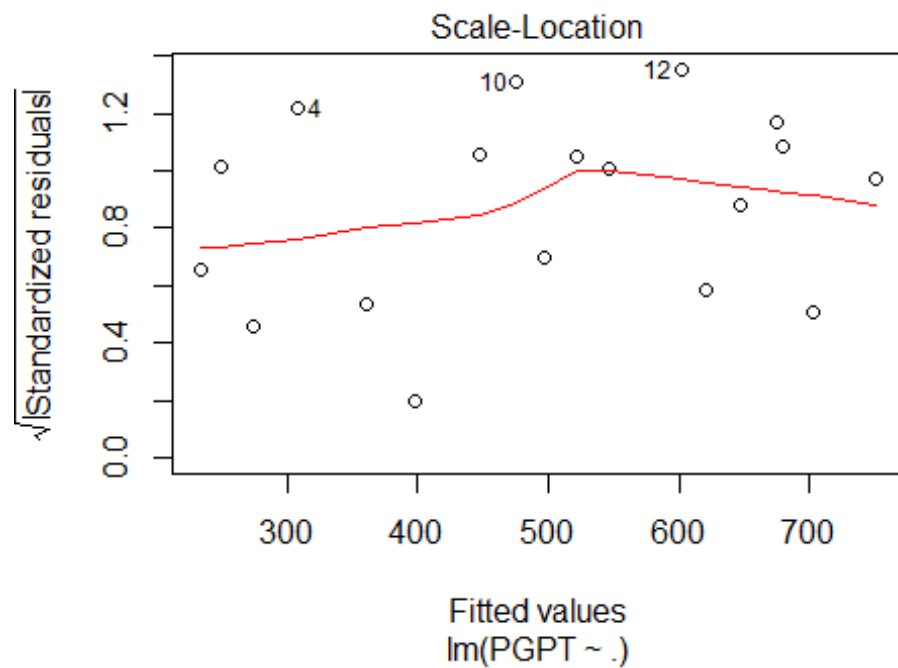
#Plotting the residuals
qplot(.fitted, .stdresid, data = fit, geom = "hex") + geom_smooth(size = 1) +
xlab("fitted values") + ylab("standardized residuals")

## Warning: package 'hexbin' was built under R version 3.4.4
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
plot(fit)
```





```
# Predict the PGPT values using the model
pred <- predict(fit, newdata=mf, interval="confidence", level=.95)
```



```
## Warning in predict.lm(fit, newdata = mf, interval = "confidence", level =  
## 0.95): prediction from a rank-deficient fit may be misleading
```

```
pred1<-pred
```

Building Models with different combination of variables: Only corelated Variables GDP+Export+Stock+AirLog+CPI

```
# Selecting corelated variables
```

```
fit_cor <- lm(PGPT ~ GDPE+Stock+AirLog+CPI, data = mt)
```

```
# Keeping fit_all in a common variable for repeatability of code
```

```
fit <- fit_cor
```

```
#Checking the summary
```

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = PGPT ~ GDPE + Stock + AirLog + CPI, data = mt)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -12.4913  -2.8159  -0.8839   3.0559  10.1324
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -4.378e+02  1.342e+01 -32.620 7.41e-14 ***
```

```
## GDPE         1.130e-02  7.193e-04  15.711 7.79e-10 ***
```

```
## Stock        -5.592e-03  4.147e-03  -1.348  0.2005
```

```
## AirLog        9.073e-02  3.672e-02   2.471  0.0281 *
```

```
## CPI           3.882e-01  6.307e-01   0.615  0.5489
```

```
## ---
```

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

```
##
```

```
## Residual standard error: 7.174 on 13 degrees of freedom
```

```
## Multiple R-squared:  0.9986, Adjusted R-squared:  0.9981
```

```
## F-statistic: 2279 on 4 and 13 DF, p-value: < 2.2e-16
```

```
drop1(fit, test = "F")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## PGPT ~ GDPE + Stock + AirLog + CPI
```

```
##           Df Sum of Sq    RSS    AIC  F value    Pr(>F)
```

```
## <none>                 669.1   75.079
```

```
## GDPE      1   12703.5 13372.5 126.991 246.8314 7.787e-10 ***
```

```
## Stock     1     93.6   762.6  75.436   1.8183   0.2005
```

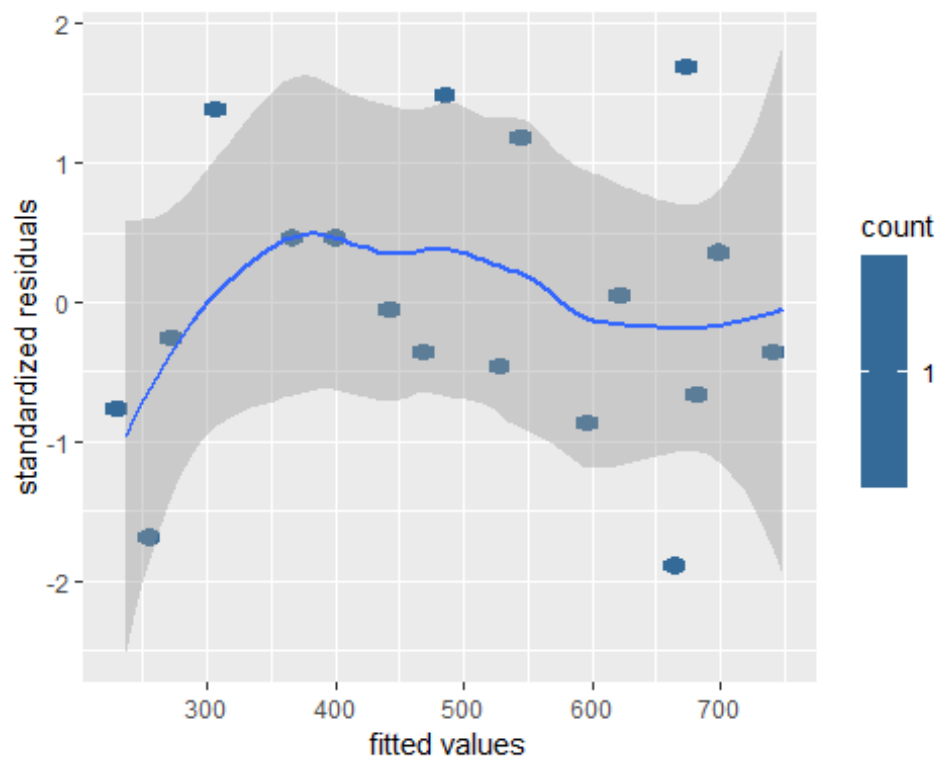
```
## AirLog    1    314.1   983.2  80.008   6.1037   0.0281 *
```

```
## CPI       1     19.5   688.6  73.596   0.3788   0.5489
```

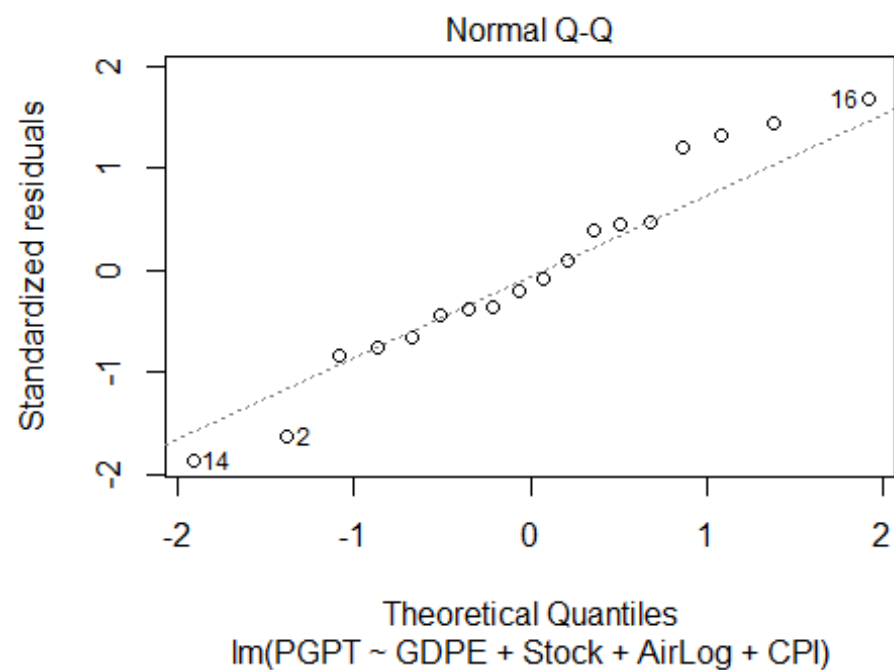
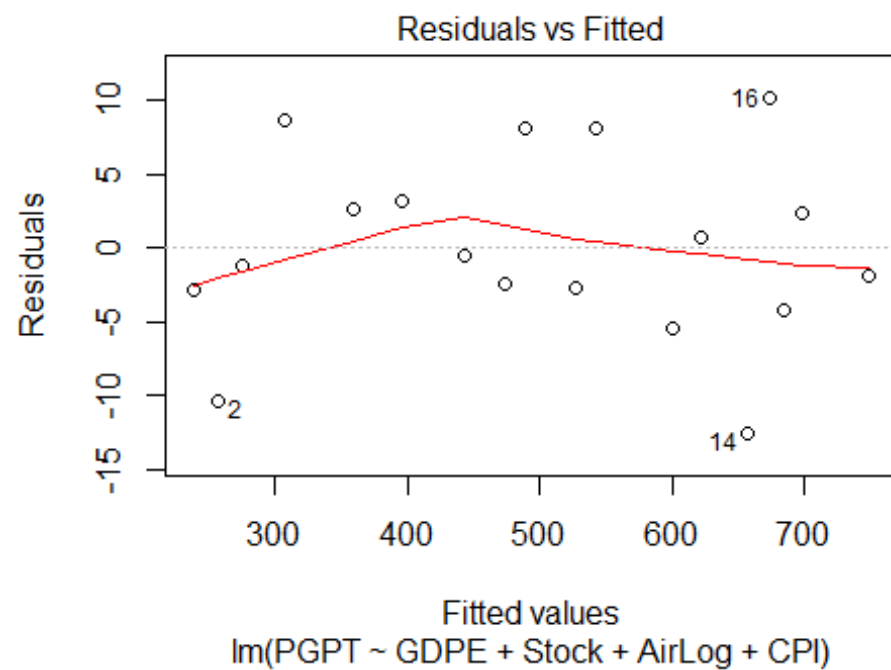
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

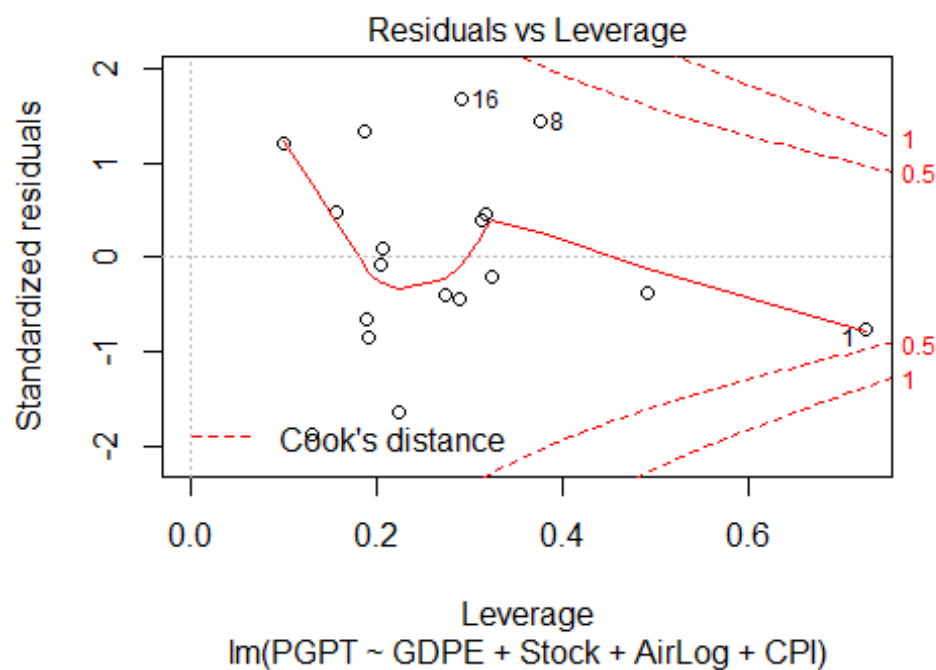
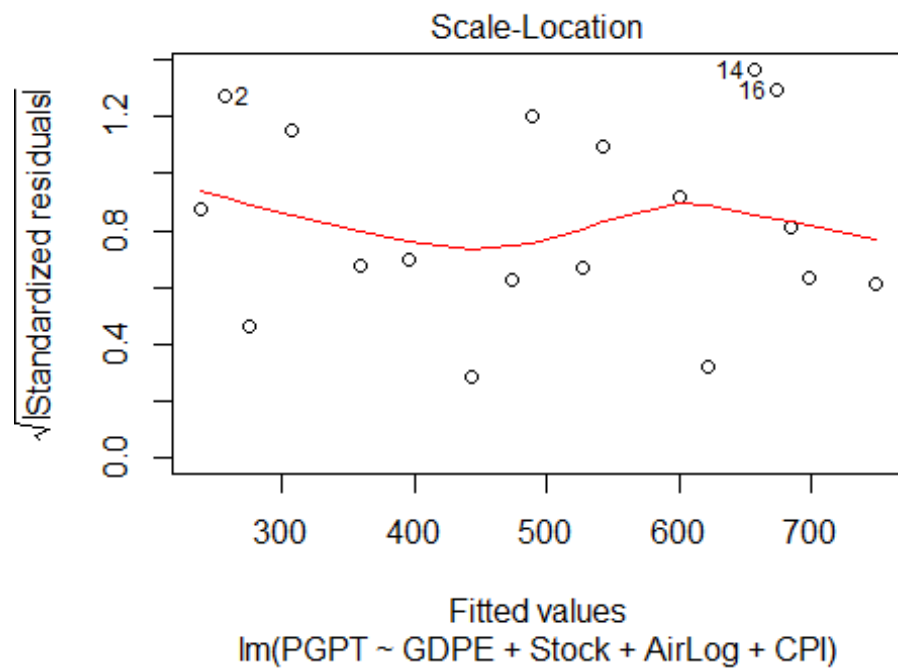
#Plotting the residuals
qplot(.fitted, .stdresid, data = fit, geom = "hex") + geom_smooth(size = 1) +
  xlab("fitted values") + ylab("standardized residuals")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
plot(fit)
```





```
# Predict the PGPT values using the model
pred <- predict(fit, newdata=mf, interval="confidence", level=.95)
pred2 <- pred
```

Building Models with Export and GDP

Selecting correlated variables

```
fit_3 <- lm(PGPT ~ GDPE, data = mt)
```

Keeping fit_all in a common variable for repeatability of code

```
fit <- fit_3
```

#Checking the summary

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = PGPT ~ GDPE, data = mt)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max  
## -11.154  -7.616   1.229   6.811  12.680
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -4.336e+02  1.160e+01  -37.38  <2e-16 ***  
## GDPE         1.203e-02  1.473e-04   81.63  <2e-16 ***
```

```
## ---
```

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

```
##
```

```
## Residual standard error: 8.387 on 16 degrees of freedom
```

```
## Multiple R-squared:  0.9976, Adjusted R-squared:  0.9975
```

```
## F-statistic: 6663 on 1 and 16 DF, p-value: < 2.2e-16
```

```
drop1(fit, test = "F")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## PGPT ~ GDPE
```

```
##           Df Sum of Sq    RSS    AIC F value    Pr(>F)
```

```
## <none>                 1125  78.439
```

```
## GDPE      1     468666 469792 185.054 6663.2 < 2.2e-16 ***
```

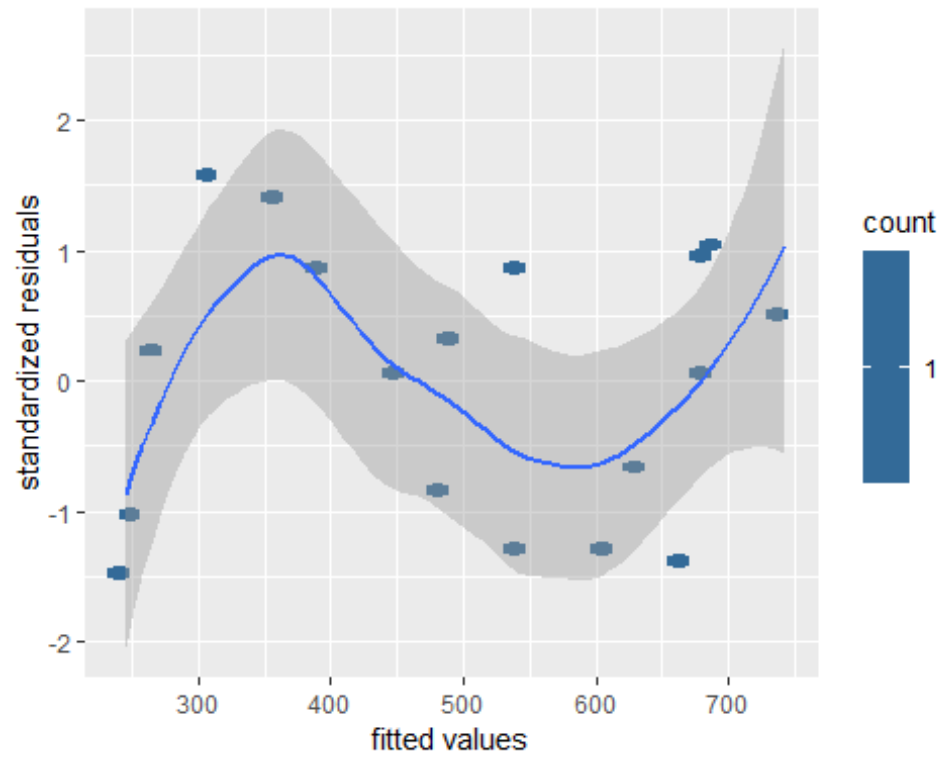
```
## ---
```

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

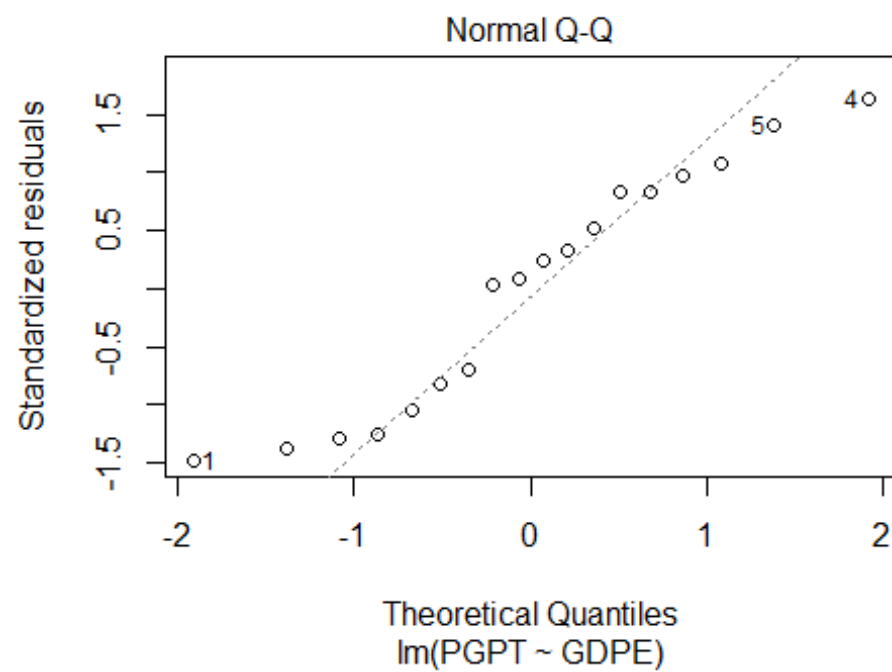
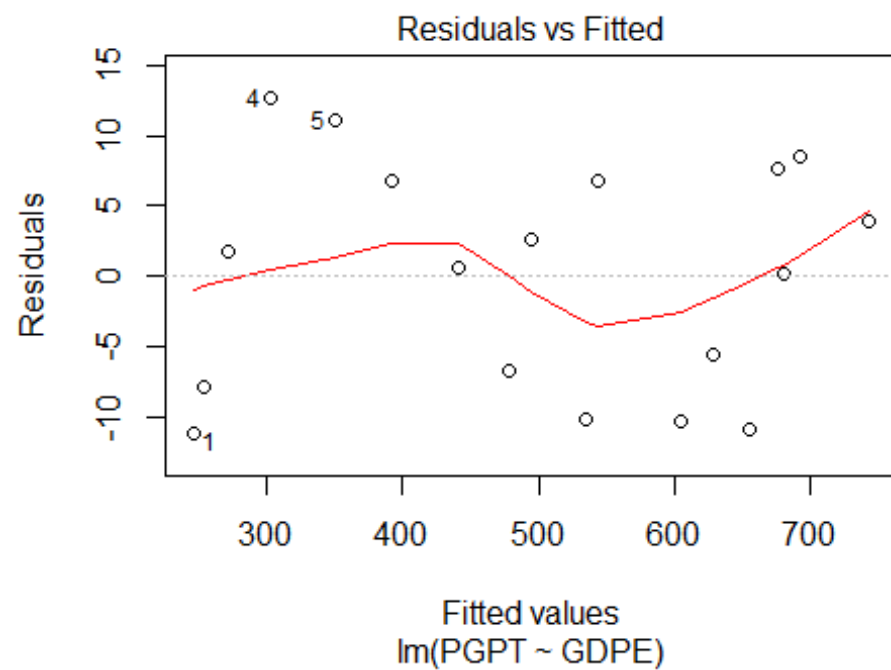
#Plotting the residuals

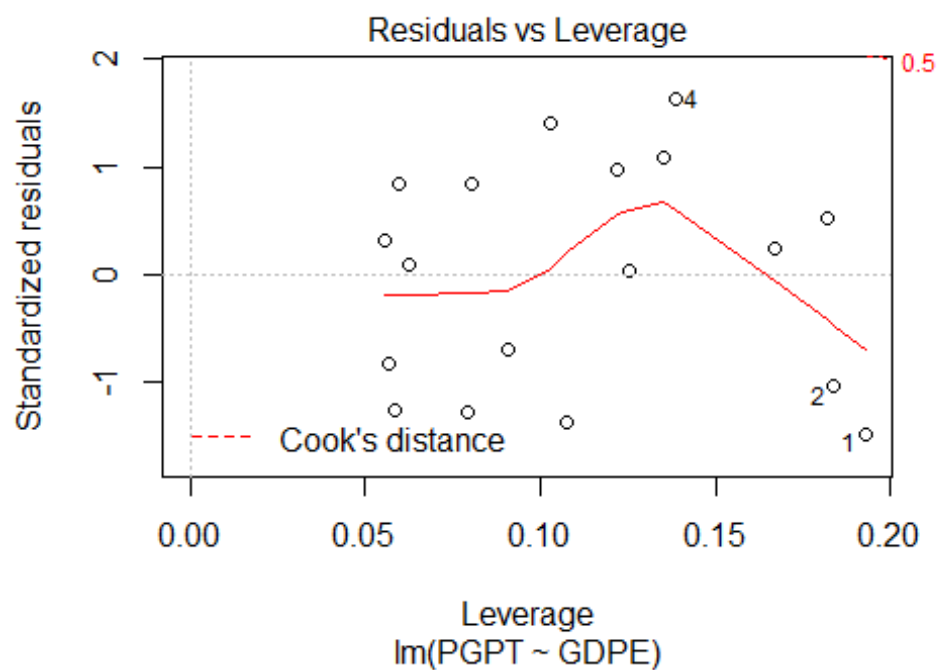
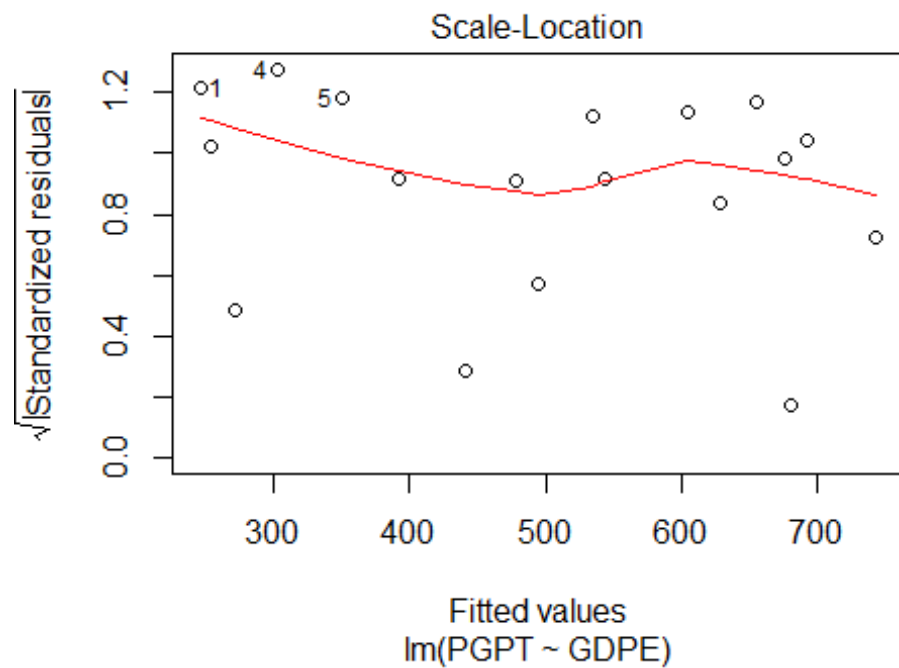
```
qplot(.fitted, .stdresid, data = fit, geom = "hex") + geom_smooth(size = 1) +  
xlab("fitted values") + ylab("standardized residuals")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
plot(fit)
```





```
# Predict the PGPT values using the model
pred <- predict(fit, newdata=mf, interval="confidence", level=.95)
pred3 <- pred
```


Building Models with Export

Selecting correlated variables

```
fit_4 <- lm(PGPT ~ Export, data = mt)
```

Keeping fit_all in a common variable for repeatability of code

```
fit <- fit_4
```

#Checking the summary

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = PGPT ~ Export, data = mt)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -70.876 -26.714  -7.418   7.597 113.661
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 43.90505    39.25574   1.118   0.28
```

```
## Export      0.03392     0.00277  12.245 1.54e-09 ***
```

```
## ---
```

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

```
##
```

```
## Residual standard error: 53.21 on 16 degrees of freedom
```

```
## Multiple R-squared:  0.9036, Adjusted R-squared:  0.8975
```

```
## F-statistic: 149.9 on 1 and 16 DF, p-value: 1.535e-09
```

```
drop1(fit, test = "F")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## PGPT ~ Export
```

```
##      Df Sum of Sq    RSS    AIC F value    Pr(>F)
```

```
## <none>                 45300 144.95
```

```
## Export  1    424491 469792 185.05  149.93 1.535e-09 ***
```

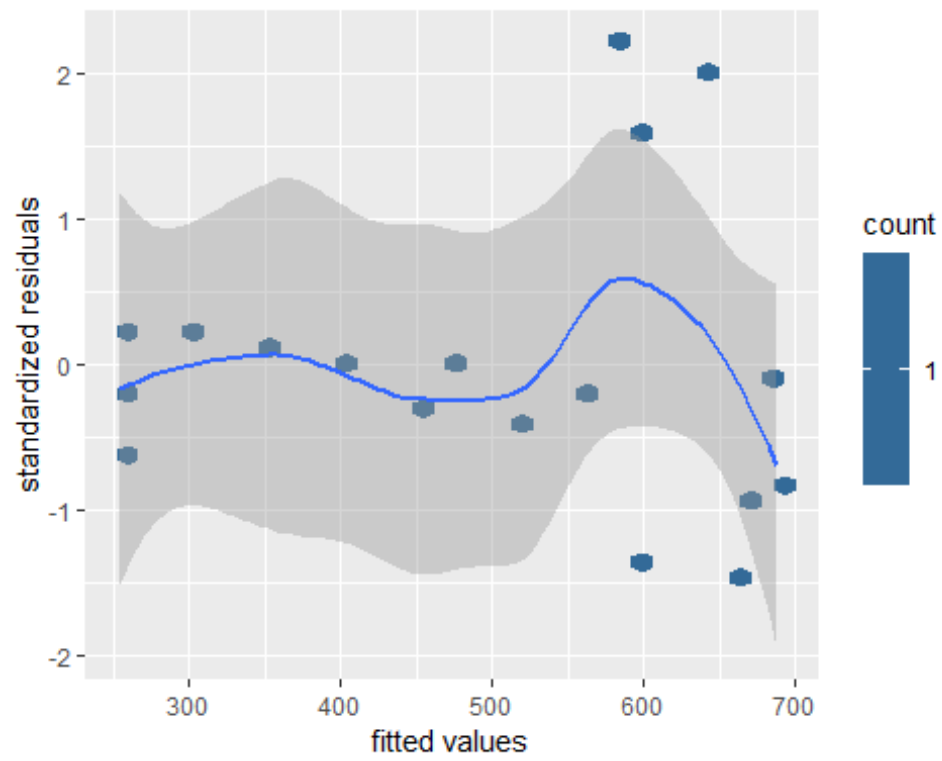
```
## ---
```

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

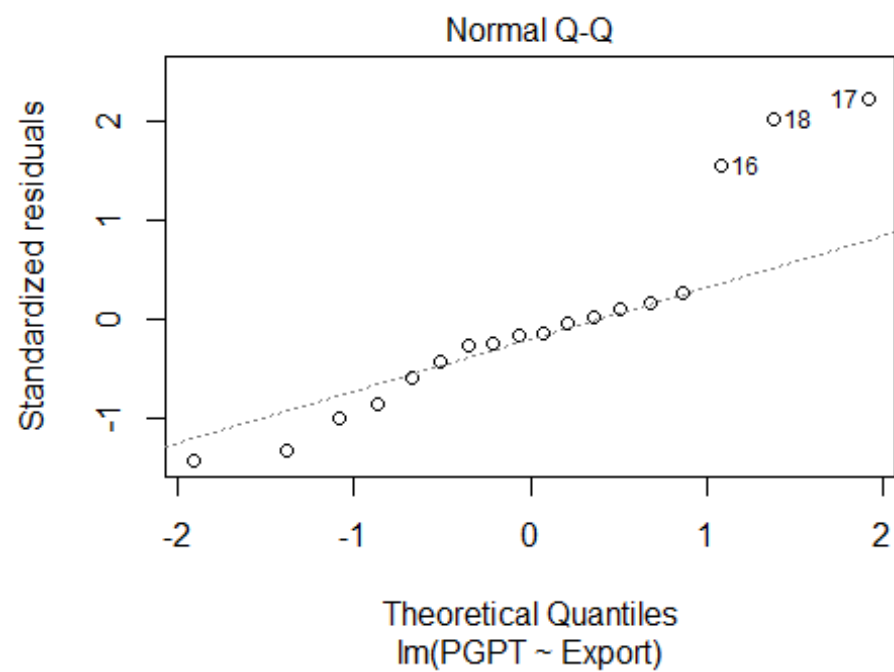
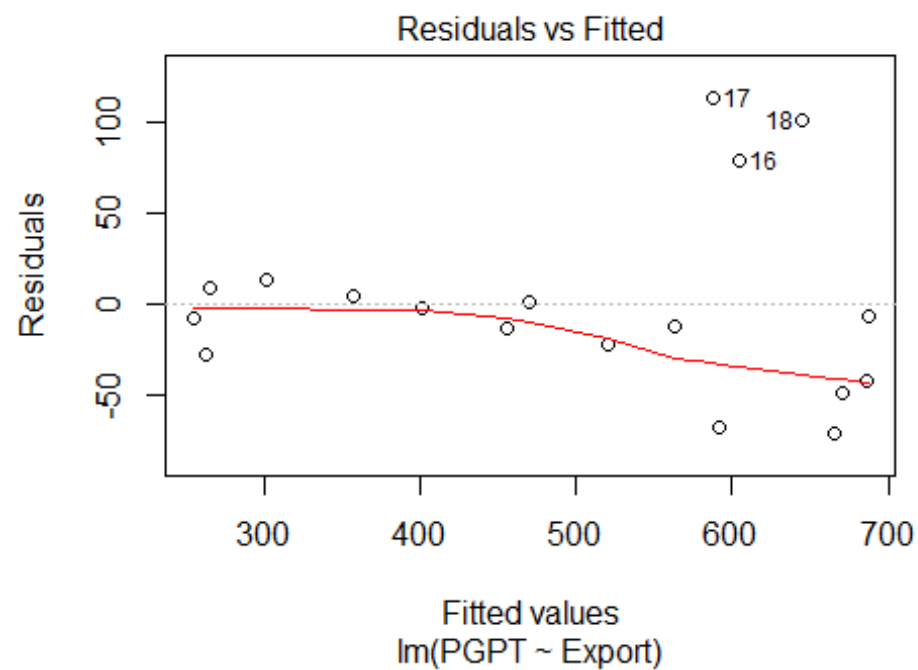
#Plotting the residuals

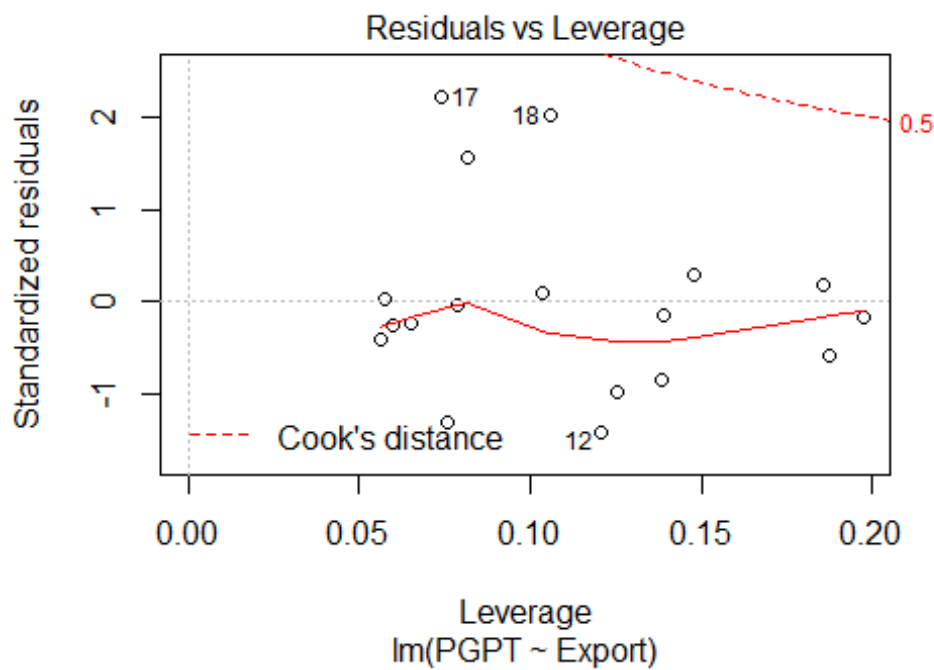
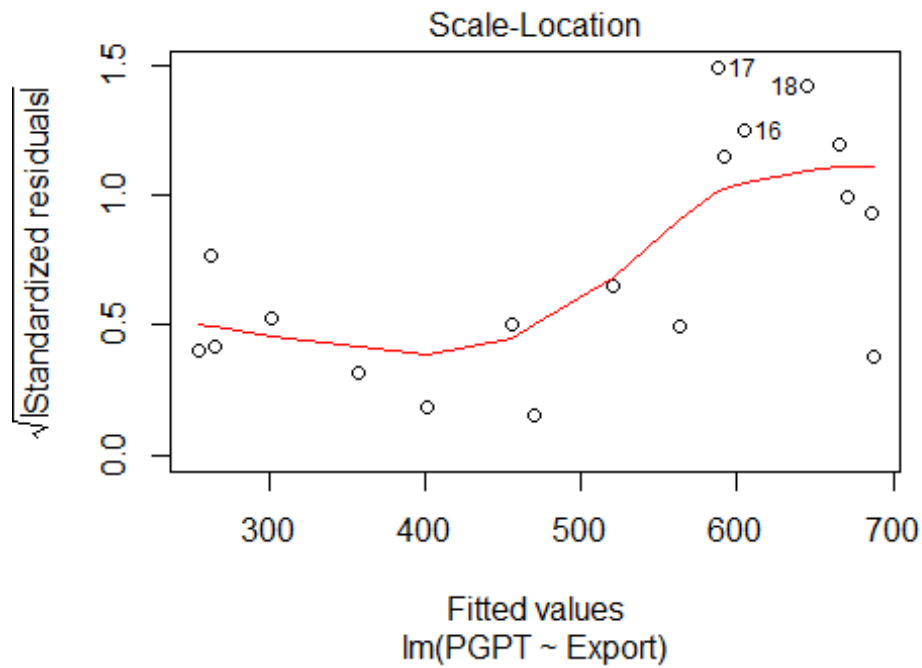
```
qplot(.fitted, .stdresid, data = fit, geom = "hex") + geom_smooth(size = 1) +  
xlab("fitted values") + ylab("standardized residuals")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
plot(fit)
```





```
# Predict the PGPT values using the model
pred <- predict(fit, newdata=mf, interval="confidence", level=.95)
pred4 <- pred
rbind(pred1, pred2, pred3, pred4)
```

```
##          fit          lwr          upr
## 19 780.9715 761.2643 800.6786
## 19 778.5371 763.7335 793.3407
## 19 786.6696 778.1124 795.2267
## 19 668.8063 629.2177 708.3949
```

Building Models with Transformed Variables : GDP + Export and all others

Selecting correlated variables

```
fit_5 <- lm(PGPT ~ . - GDP - Export, data = mt)
```

Keeping fit_all in a common variable for repeatability of code

```
fit <- fit_5
```

#Checking the summary

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = PGPT ~ . - GDP - Export, data = mt)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -7.6535 -3.2816  0.6802  2.7977  5.9120
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.244e+02  2.845e+01 -18.432 4.76e-09 ***
## CPI          -1.968e+00  1.138e+00  -1.730  0.1144
## Oil          -5.957e-01  1.925e-01  -3.095  0.0114 *
## AirLog        5.518e-02  3.175e-02   1.738  0.1129
## GDPE          1.558e-02  1.713e-03   9.098 3.75e-06 ***
## BDI          -1.074e-03  1.287e-03  -0.834  0.4235
## PMI           7.805e-01  4.258e-01   1.833  0.0967 .
## Stock        -6.151e-03  3.628e-03  -1.695  0.1209
```

```
## ---
```

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

```
##
```

```
## Residual standard error: 5.026 on 10 degrees of freedom
```

```
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9991
```

```
## F-statistic: 2655 on 7 and 10 DF, p-value: 1.58e-15
```

```
drop1(fit, test = "F")
```

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## PGPT ~ (GDP + CPI + Oil + AirLog + GDPE + BDI + PMI + Stock +
```

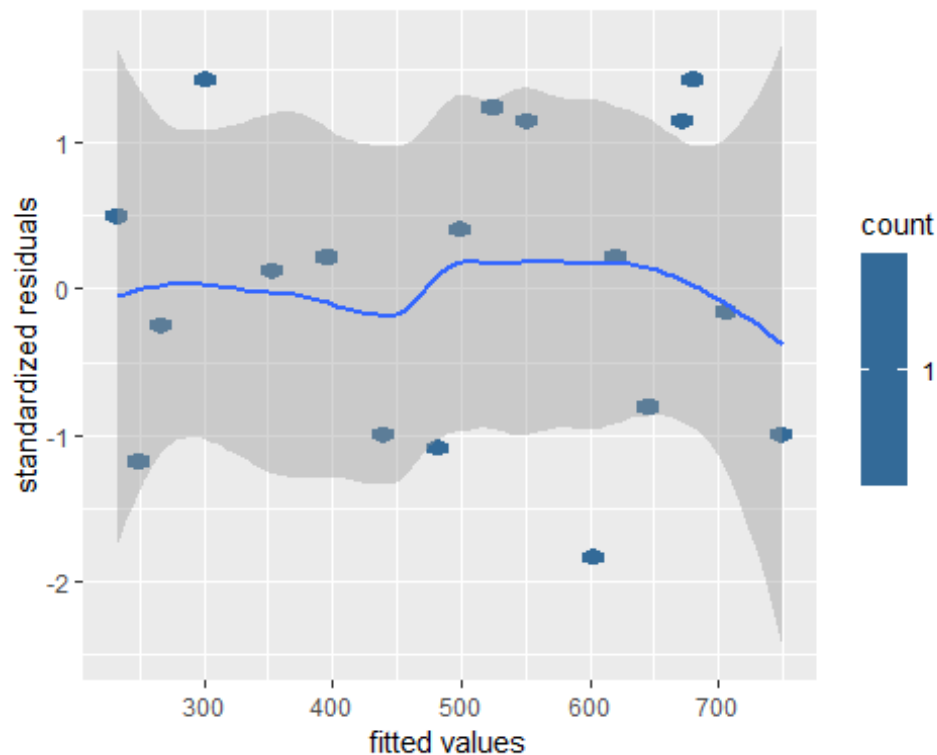
```
##      Export) - GDP - Export
```

```
##           Df Sum of Sq      RSS       AIC F value    Pr(>F)
## <none>                252.61    63.547
## CPI         1       75.56   328.17    66.257    2.9912   0.11441
## Oil         1      241.93   494.54    73.639    9.5770   0.01135 *
## AirLog      1       76.29   328.90    66.297    3.0199   0.11289
## GDPE        1    2090.93  2343.54   101.643   82.7720  3.754e-06 ***
## BDI         1       17.59   270.20    62.758    0.6963   0.42353
## PMI         1       84.86   337.48    66.760    3.3595   0.09672 .
## Stock       1       72.60   325.21    66.094    2.8739   0.12088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

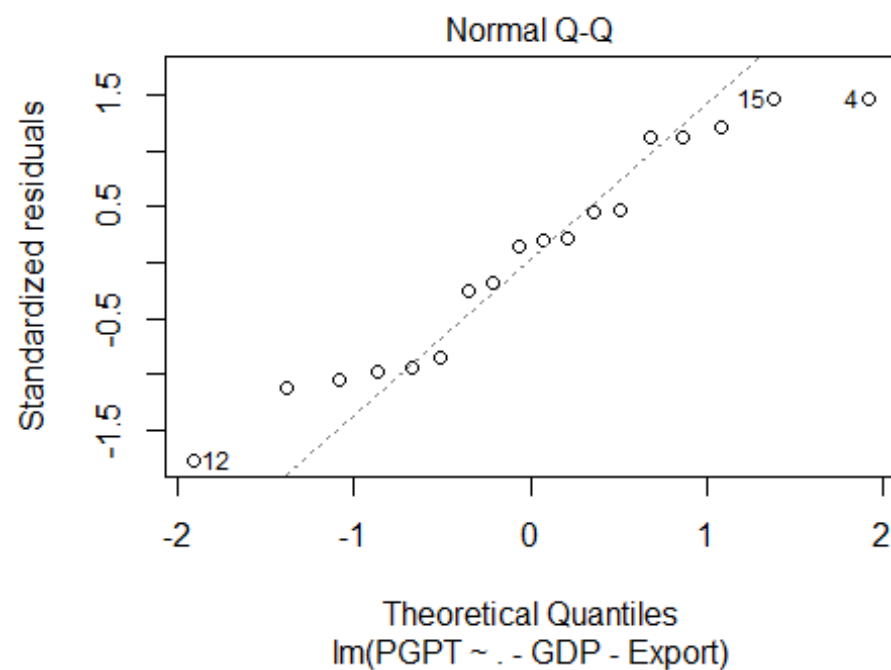
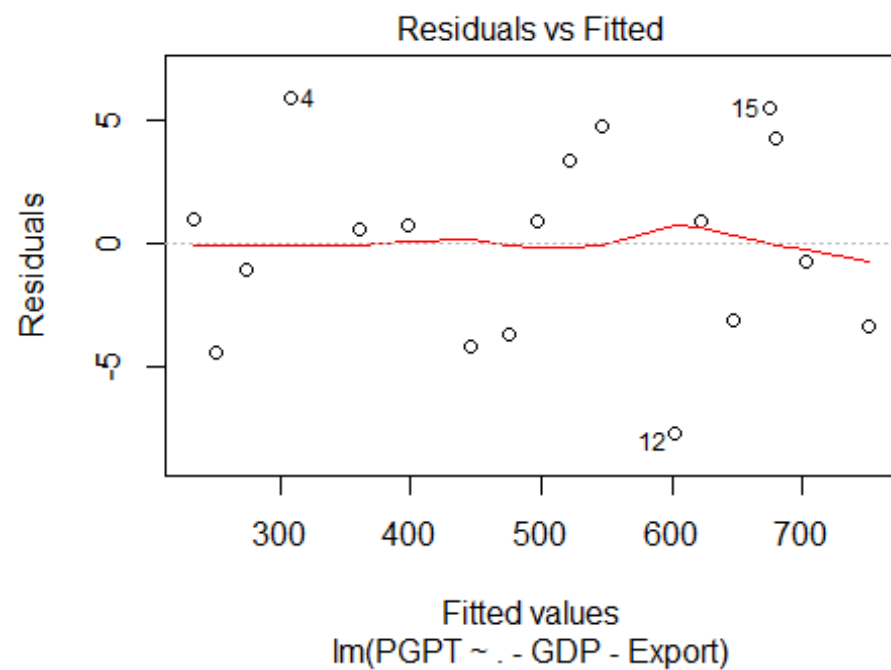
#Plotting the residuals

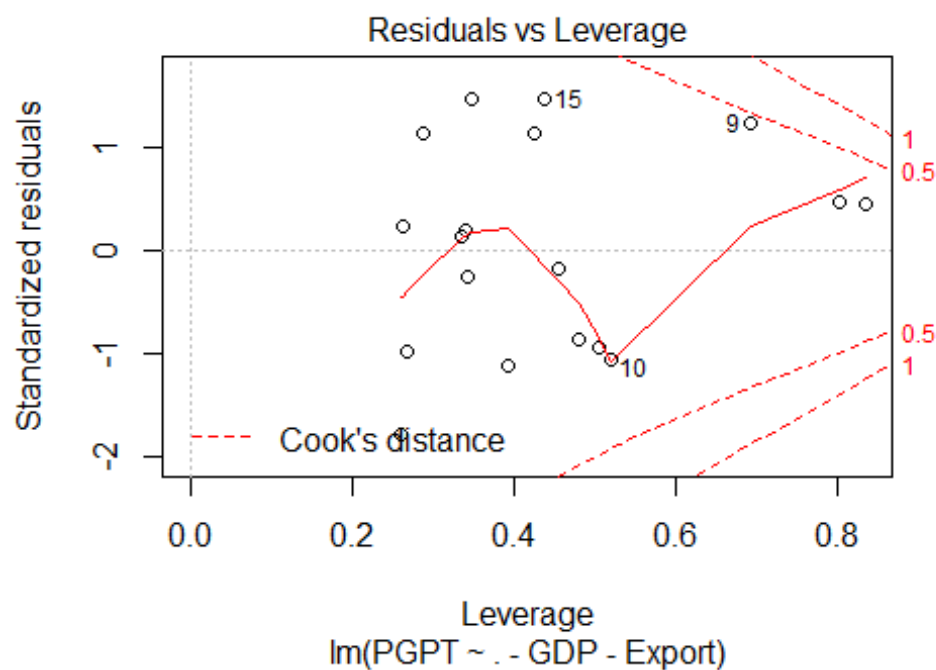
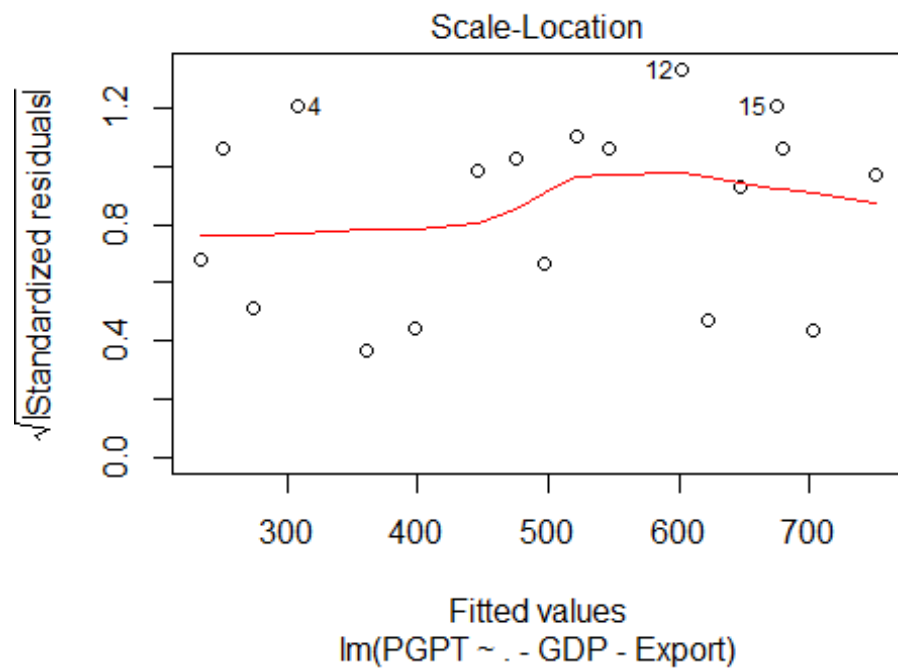
```
qplot(.fitted, .stdresid, data = fit, geom = "hex") + geom_smooth(size = 1) +
xlab("fitted values") + ylab("standardized residuals")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
plot(fit)
```





```
# Predict the PGPT values using the model
pred <- predict(fit, newdata=mf, interval="confidence", level=.95)
pred5 <- pred
rbind(pred1, pred2, pred3, pred4, pred5)
```



```
##          fit      lwr      upr
## 19 780.9715 761.2643 800.6786
## 19 778.5371 763.7335 793.3407
## 19 786.6696 778.1124 795.2267
## 19 668.8063 629.2177 708.3949
## 19 778.3475 766.3801 790.3148
```

Predicting the historical values of throughput based on the model and comparing it with the actual historical data

```
comp <- cbind(as.data.frame(ModelData$Period)[1:18,], mt$PGPT,
as.data.frame(fitted(fit)))
```

```
comp
```

```
##      as.data.frame(ModelData$Period)[1:18, ] mt$PGPT fitted(fit)
## 1      2000      234.6      233.5723
## 2      2001      246.2      250.5979
## 3      2002      272.8      273.8562
## 4      2003      315.0      309.0880
## 5      2004      361.8      361.2407
## 6      2005      398.7      397.8989
## 7      2006      442.2      446.3722
## 8      2007      497.8      496.8955
## 9      2008      524.5      521.1123
## 10     2009      471.2      474.8682
## 11     2010      550.5      545.7349
## 12     2011      595.1      602.7535
## 13     2012      622.3      621.3461
## 14     2013      644.4      647.5052
## 15     2014      680.3      674.8151
## 16     2015      683.5      679.2025
## 17     2016      701.3      702.0001
## 18     2017      746.7      750.0404
```