Petrol

Importing libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(car)
library(gridExtra)
library(caTools)
library(xlsx)
```

```
## Loading required package: rJava
## Loading required package: xlsxjars
```

```
library(corrplot)
```

Loading data

```
library(xlsx)
getwd()
```

```
## [1] "/Users/nareshshah/Downloads"
```

```
setwd('/Users/nareshshah/Downloads')
workbook <- "DataPetrolCase1.xlsx"
historical <- read.xlsx(workbook,3)
trimestres <- read.xlsx(workbook,4)
transformed <- read.xlsx(workbook,5)
```

STEP 1: Take a look at the simple scatter plots to see if the relations are linear or if you need to transform the data

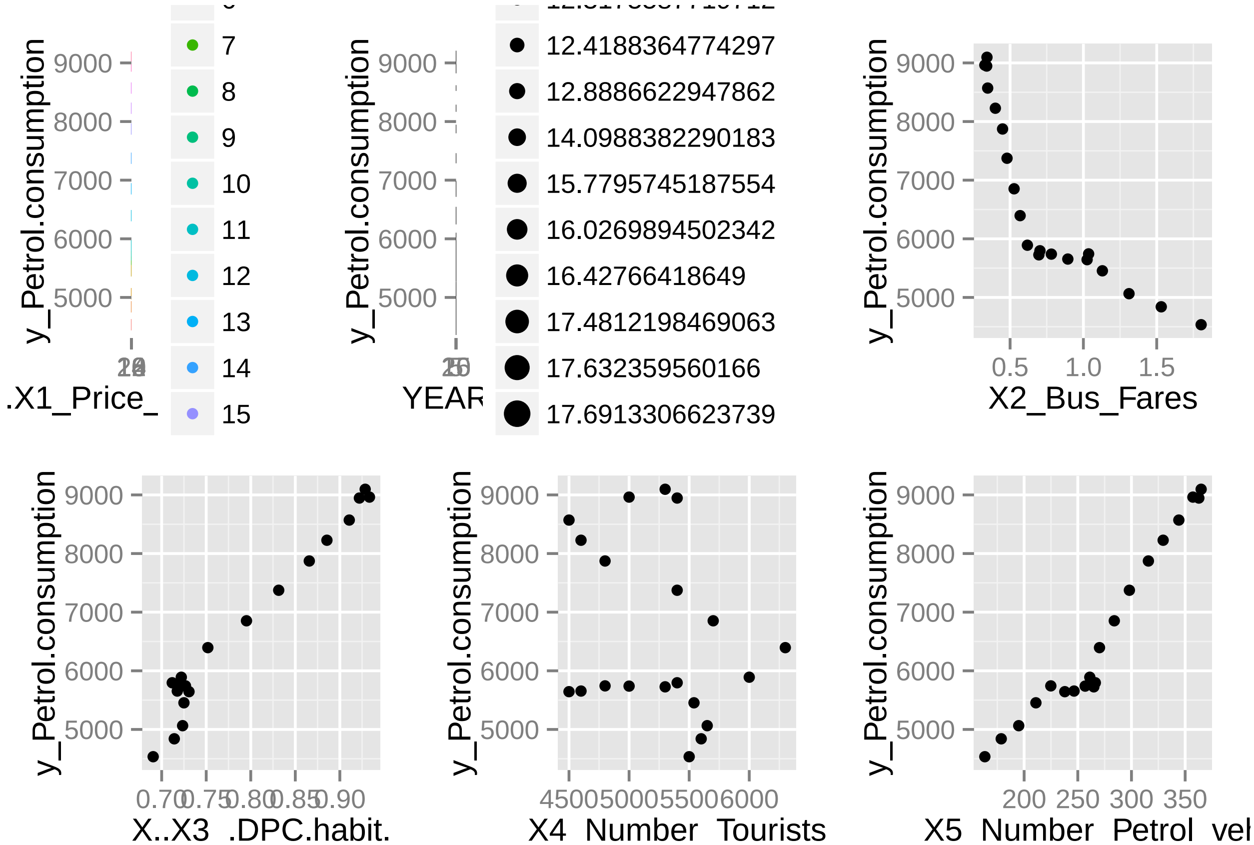
```r
str(transformed)
```

```
## 'data.frame':    20 obs. of  7 variables:
##  $ YEARS                 : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ y_Petrol.consumption  : num  4535 4840 5064 5454 5743 ...
##  $ X..X1_Price_of_Petrol: num  16 17.7 17.5 16.4 15.8 ...
##  $ X2_Bus_Fares          : num  1.8 1.53 1.31 1.13 1.04 ...
##  $ X..X3_.DPC.habit.     : num  0.69 0.714 0.723 0.725 0.727 ...
##  $ X4_Number_Tourists    : num  5500 5600 5650 5540 4800 4500 4600 5000 5300 5400 ...
##  $ X5_Number_Petrol_veh  : num  163 179 195 211 225 ...
```

```r
summary(transformed)
```

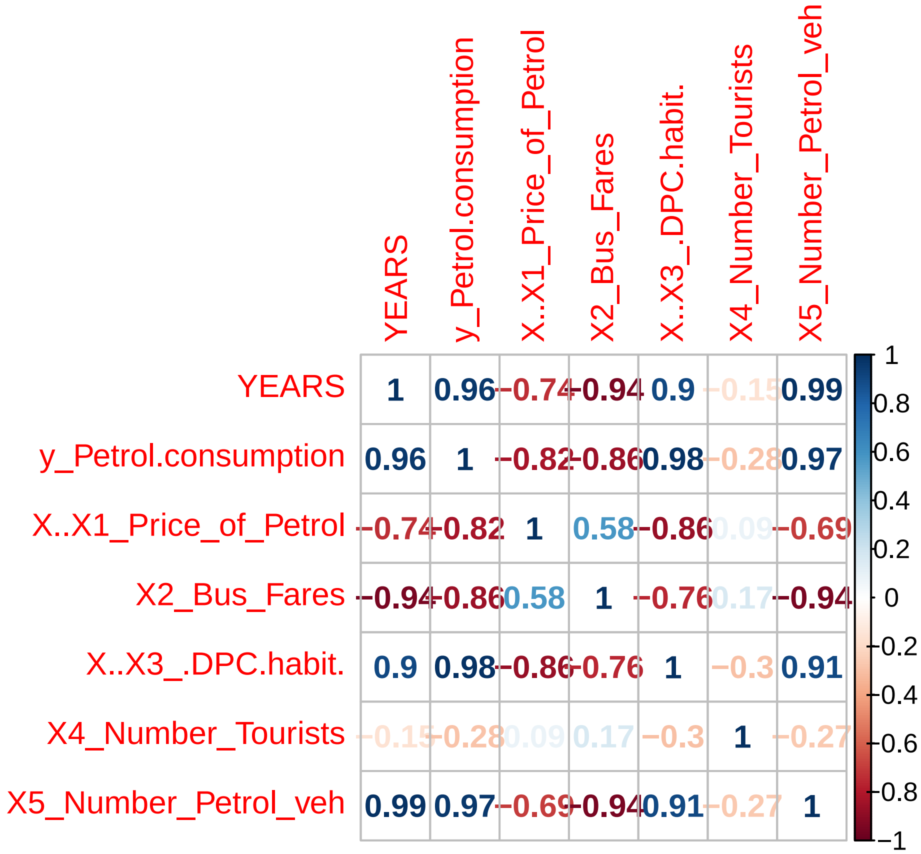
```
##      YEARS       y_Petrol.consumption X..X1_Price_of_Petrol
##  Min.   : 1.00   Min.   :4535         Min.   :11.51
##  1st Qu.: 5.75   1st Qu.:5652         1st Qu.:12.23
##  Median :10.50   Median :5843         Median :15.90
##  Mean   :10.50   Mean   :6619         Mean   :15.40
##  3rd Qu.:15.25   3rd Qu.:7962         3rd Qu.:17.89
##  Max.   :20.00   Max.   :9096         Max.   :19.96
##   X2_Bus_Fares    X..X3_.DPC.habit. X4_Number_Tourists
##  Min.   :0.3278   Min.   :0.6904    Min.   :4500
##  1st Qu.:0.4363   1st Qu.:0.7199    1st Qu.:4800
##  Median :0.6572   Median :0.7287    Median :5350
##  Mean   :0.7654   Mean   :0.7862    Mean   :5244
##  3rd Qu.:1.0278   3rd Qu.:0.8707    3rd Qu.:5555
##  Max.   :1.8030   Max.   :0.9333    Max.   :6300
##  X5_Number_Petrol_veh
##  Min.   :163.4
##  1st Qu.:234.6
##  Median :265.6
##  Mean   :271.7
##  3rd Qu.:319.2
##  Max.   :364.9
```

```r
q1<-qplot(data=transformed, X..X1_Price_of_Petrol, y_Petrol.consumption, color=factor(YEARS))

q2<-qplot(data=transformed, YEARS, y_Petrol.consumption, cex=factor(X..X1_Price_of_Petrol))

q3<-qplot(data=transformed, X2_Bus_Fares, y_Petrol.consumption)

q4<-qplot(data=transformed, X..X3_.DPC.habit., y_Petrol.consumption)

q5<-qplot(data=transformed, X4_Number_Tourists, y_Petrol.consumption)

q6<-qplot(data=transformed, X5_Number_Petrol_veh, y_Petrol.consumption)

grid.arrange(q1, q2, q3, q4, q5, q6, nrow=2)
```

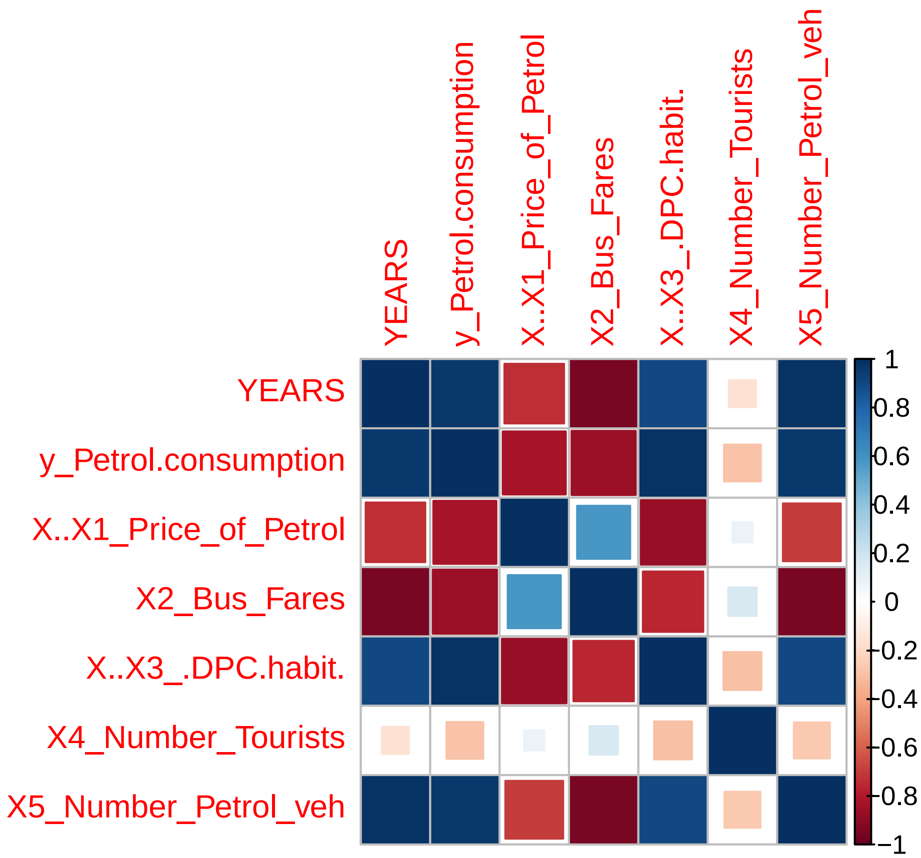


STEP 2/3: Compute all the correlations between all the variables and analyze possible multicollinearity

```
M <-cor(transformed)#Correlation Matrix
#Graficamente:
corrplot(M,method ="number")
```




```
corrplot(M,method ="square")
```



STEP 4: Estimate the regression model.

```
#To build the model we are going to see which variables are more relevant by checking the p-value in th

fit  <- lm(y_Petrol.consumption ~ (X..X1_Price_of_Petrol), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X1_Price_of_Petrol),
##     data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1835.8  -733.0   136.0   731.6  1257.1
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           12715.38    1003.54  12.670 2.09e-10 ***
## X..X1_Price_of_Petrol  -395.88      63.94  -6.191 7.63e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 867.6 on 18 degrees of freedom
## Multiple R-squared:  0.6805, Adjusted R-squared:  0.6627
## F-statistic: 38.33 on 1 and 18 DF,  p-value: 7.634e-06
```

```r
anova(fit) #F value=38.333>Pr(>F), reject H0, the coefficient is not 0.
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                         Df   Sum Sq  Mean Sq F value    Pr(>F)
## X..X1_Price_of_Petrol  1 28851622 28851622  38.333 7.634e-06 ***
## Residuals              18 13547914   752662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Multiple R-squared:  0.6805,   Adjusted R-squared:  0.6627

fit  <- lm(y_Petrol.consumption ~ (X2_Bus_Fares), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X2_Bus_Fares), data = transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1172.55  -638.01   -65.74   568.21  1201.91
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8925.1      374.1  23.861 4.49e-15 ***
## X2_Bus_Fares  -3012.4      430.0  -7.006 1.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 795 on 18 degrees of freedom
## Multiple R-squared:  0.7317, Adjusted R-squared:  0.7168
## F-statistic: 49.09 on 1 and 18 DF,  p-value: 1.535e-06
```

```r
anova(fit) #F value=49.086>Pr(>F), reject H0, the coefficient is not 0.
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##              Df   Sum Sq  Mean Sq F value    Pr(>F)
## X2_Bus_Fares  1 31023272 31023272  49.086 1.535e-06 ***
## Residuals    18 11376264   632015
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Multiple R-squared:  0.7317,   Adjusted R-squared:  0.7168

fit  <- lm(y_Petrol.consumption ~ (X..X3_.DPC.habit.), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X3_.DPC.habit.), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -577.52 -111.62   33.09  186.56  416.63
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -6488.6      590.1  -11.00 2.03e-09 ***
## X..X3_.DPC.habit.  16672.1      746.1   22.34 1.41e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 286.3 on 18 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9633
## F-statistic: 499.3 on 1 and 18 DF,  p-value: 1.409e-14
```

```r
anova(fit) #F value=499.32>>Pr(>F), reject H0, the coefficient is not 0.
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                    Df   Sum Sq  Mean Sq F value    Pr(>F)
## X..X3_.DPC.habit.  1 40924255 40924255  499.32 1.409e-14 ***
## Residuals          18  1475280    81960
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Multiple R-squared:  0.9652,   Adjusted R-squared:  0.9633

fit  <- lm(y_Petrol.consumption ~ (X4_Number_Tourists), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X4_Number_Tourists), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1872.0 -1225.1  -397.2   931.0  2523.2
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        10980.1169  3515.6237   3.123  0.00587 **
## X4_Number_Tourists    -0.8315     0.6674  -1.246  0.22878
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1473 on 18 degrees of freedom
## Multiple R-squared:  0.07939,    Adjusted R-squared:  0.02824
## F-statistic: 1.552 on 1 and 18 DF,  p-value: 0.2288
```

```
anova(fit) #F value=1.5522 slightly higher than Pr(>F).
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                     Df   Sum Sq Mean Sq F value Pr(>F)
## X4_Number_Tourists   1  3366071 3366071  1.5522 0.2288
## Residuals           18 39033464 2168526
```

```
#Multiple R-squared:  0.07939,  Adjusted R-squared:  0.02824


fit  <- lm(y_Petrol.consumption ~ (X5_Number_Petrol_veh), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X5_Number_Petrol_veh), data = transformed)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -732.4 -233.8  181.1  260.3  493.0
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           151.361    408.362   0.371    0.715
## X5_Number_Petrol_veh   23.809      1.469  16.210  3.5e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 388.6 on 18 degrees of freedom
## Multiple R-squared:  0.9359, Adjusted R-squared:  0.9323
## F-statistic: 262.8 on 1 and 18 DF,  p-value: 3.496e-12
```

```
anova(fit) #F value=262.76>>Pr(>F), reject H0, the coefficient is not 0.
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                      Df   Sum Sq  Mean Sq F value    Pr(>F)
## X5_Number_Petrol_veh  1 39681252 39681252  262.76 3.496e-12 ***
## Residuals            18  2718283   151016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Multiple R-squared:  0.9359,   Adjusted R-squared:  0.9323


#Combinamos las dos variables que más explican: X3 y X5
fit  <- lm(y_Petrol.consumption ~ (X..X3_.DPC.habit.+X5_Number_Petrol_veh), data=transformed)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                      Df   Sum Sq  Mean Sq F value    Pr(>F)
## X..X3_.DPC.habit.     1 40924255 40924255 5153.20 < 2.2e-16 ***
## X5_Number_Petrol_veh  1  1340274  1340274  168.77 2.961e-10 ***
## Residuals            17   135006     7942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#F value=5153.20>>Pr(>F), reject H0, the coefficient of X3 is not 0.
#F value=168.77>>Pr(>F), reject H0, the coefficient of X5 is not 0.
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X3_.DPC.habit. + X5_Number_Petrol_veh),
##     data = transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -159.472  -43.944   -4.571   23.496  215.277
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4159.19     256.69  -16.20 9.05e-12 ***
## X..X3_.DPC.habit.    10073.34     558.52   18.04 1.61e-12 ***
## X5_Number_Petrol_veh    10.52       0.81   12.99 2.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.12 on 17 degrees of freedom
## Multiple R-squared:  0.9968, Adjusted R-squared:  0.9964
## F-statistic:  2661 on 2 and 17 DF,  p-value: < 2.2e-16
```

```r
#Multiple R-squared:  0.9968,   Adjusted R-squared:  0.9964
#X1 and X5 are highly correlated cov(x3,x5)=0.9094443. This means that there is multicolinearity, so we

fit  <- lm(y_Petrol.consumption ~ (X..X3_.DPC.habit.+X5_Number_Petrol_veh+X2_Bus_Fares), data=transforme
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X3_.DPC.habit. + X5_Number_Petrol_veh +
##     X2_Bus_Fares), data = transformed)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.344  -36.059   -8.478   35.032  206.466
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3890.715    385.582 -10.090 2.42e-08 ***
## X..X3_.DPC.habit.    10629.526    816.964  13.011 6.32e-10 ***
## X5_Number_Petrol_veh     8.483      2.326   3.647  0.00217 **
## X2_Bus_Fares          -198.185    211.766  -0.936  0.36325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.44 on 16 degrees of freedom
## Multiple R-squared:  0.997,  Adjusted R-squared:  0.9964
## F-statistic:  1761 on 3 and 16 DF,  p-value: < 2.2e-16
```

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                      Df   Sum Sq  Mean Sq   F value    Pr(>F)
## X..X3_.DPC.habit.     1 40924255 40924255 5115.5690 < 2.2e-16 ***
## X5_Number_Petrol_veh  1  1340274  1340274  167.5355 6.821e-10 ***
## X2_Bus_Fares          1     7007     7007    0.8758    0.3633
## Residuals            16   127999     8000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Multiple R-squared:  0.997,    Adjusted R-squared:  0.9964
#There is no big difference between this model and the previous one. We can see in the summary(fit) tha

#In order to avoid multicolinearity we are going to omit X3 and combine X5 with X1
fit  <- lm(y_Petrol.consumption ~ (X..X1_Price_of_Petrol+X5_Number_Petrol_veh), data=transformed)
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X1_Price_of_Petrol +
##     X5_Number_Petrol_veh), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -381.62 -158.13   37.34  149.66  290.91
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3710.811    600.486   6.180 1.01e-05 ***
## X..X1_Price_of_Petrol -141.970     22.166  -6.405 6.53e-06 ***
## X5_Number_Petrol_veh    18.754      1.137  16.498 6.78e-12 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 216.4 on 17 degrees of freedom
## Multiple R-squared:  0.9812, Adjusted R-squared:  0.979
## F-statistic:   444 on 2 and 17 DF,  p-value: 2.124e-15
```

```r
anova(fit) #Multiple R-squared:  0.9812,    Adjusted R-squared:  0.979
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                         Df   Sum Sq  Mean Sq F value    Pr(>F)
## X..X1_Price_of_Petrol  1 28851622 28851622  615.84 8.579e-15 ***
## X5_Number_Petrol_veh   1 12751480 12751480  272.18 6.775e-12 ***
## Residuals             17   796434    46849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(fit) #X..X1_Price_of_Petrol=1.930757;  X5_Number_Petrol_veh=1.930757
```

```
## X..X1_Price_of_Petrol  X5_Number_Petrol_veh
##              1.930757              1.930757
```

```r
#This is a better model and we can check that there is no multicolinearity by using the function "vif"

fit  <- lm(y_Petrol.consumption ~ (X..X1_Price_of_Petrol+X5_Number_Petrol_veh+log(X2_Bus_Fares)), data=
summary(fit)#The three variables are significant
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X1_Price_of_Petrol +
##     X5_Number_Petrol_veh + log(X2_Bus_Fares)), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -172.50  -86.19  -15.43   78.71  216.34
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             363.721    659.474   0.552    0.589
## X..X1_Price_of_Petrol -160.685     13.120 -12.247 1.53e-09 ***
## X5_Number_Petrol_veh    35.060      2.815  12.456 1.20e-09 ***
## log(X2_Bus_Fares)     1968.674    330.540   5.956 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 124.4 on 16 degrees of freedom
## Multiple R-squared:  0.9942, Adjusted R-squared:  0.9931
## F-statistic: 908.1 on 3 and 16 DF,  p-value: < 2.2e-16
```

```r
anova(fit)#Multiple R-squared:  0.9942, Adjusted R-squared:  0.9931
```

```
## Analysis of Variance Table
##
## Response: y_Petrol.consumption
##                         Df   Sum Sq  Mean Sq  F value    Pr(>F)
## X..X1_Price_of_Petrol  1 28851622 28851622 1864.669 < 2.2e-16 ***
## X5_Number_Petrol_veh   1 12751480 12751480  824.123 3.428e-15 ***
## log(X2_Bus_Fares)      1   548869   548869   35.473 2.013e-05 ***
## Residuals             16   247565    15473
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
vif(fit)#The only problem with this model is that there is multi-colinearity.
```

```
## X..X1_Price_of_Petrol  X5_Number_Petrol_veh     log(X2_Bus_Fares)
##              2.048236             35.840072             37.798357
```

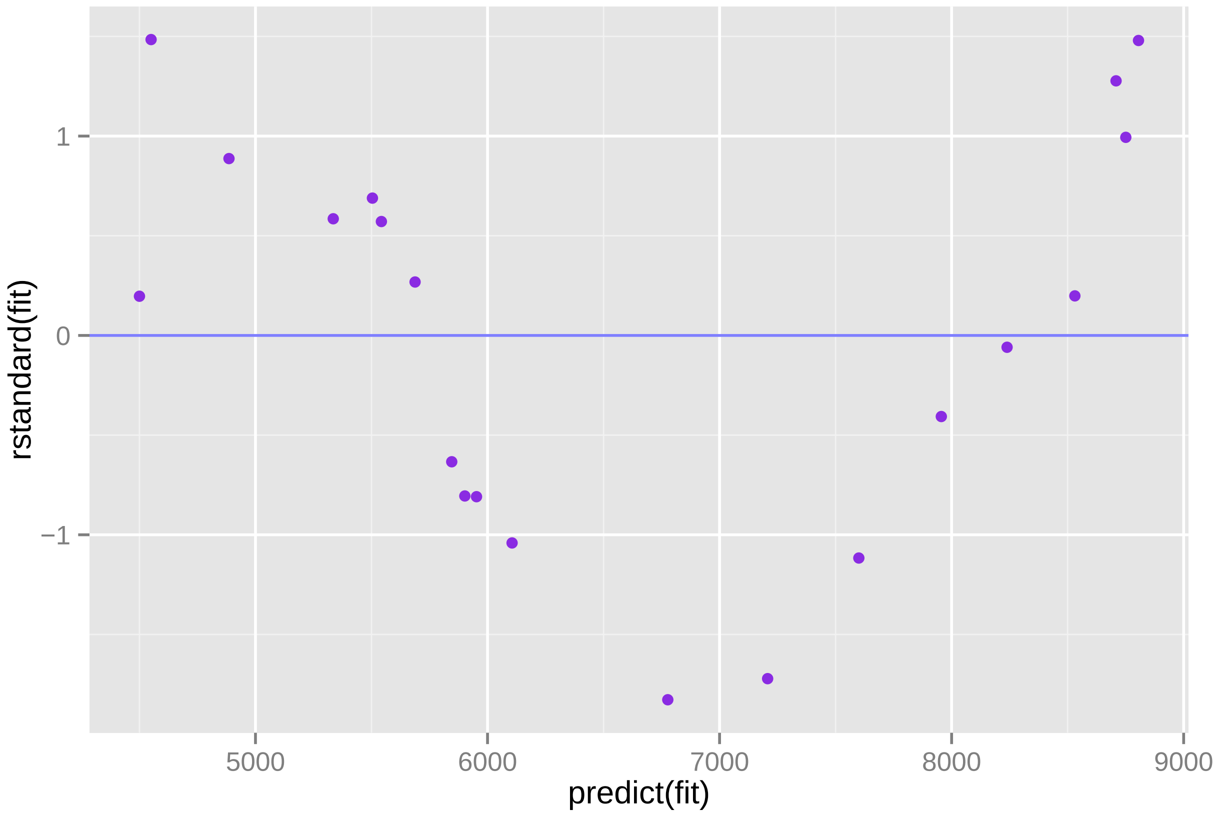
```r
#Our final model is y~a0 + a1x1 + a2x5:

fit  <- lm(y_Petrol.consumption ~ (X..X1_Price_of_Petrol+X5_Number_Petrol_veh), data=transformed)

# We should also note that this model will be the easiest one to explain as it is fairly easy to unders
```

Regression diagnostics

We will now test if the four fundamental assumptions of regression are present in our model.

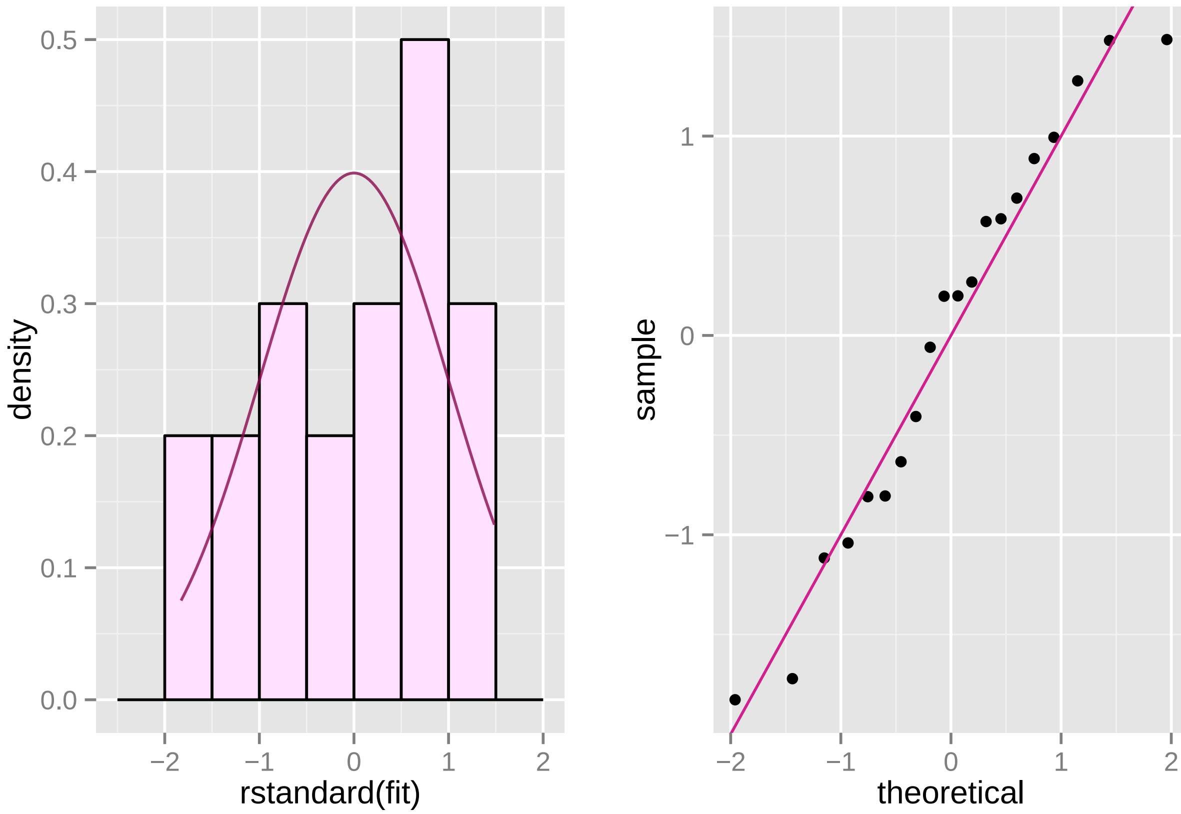
A1: Checking Linear Relation

```r
qplot(predict(fit), rstandard(fit), geom="point", colour=I("blueviolet")) +
  geom_hline(yintercept=0, colour=I("blue"), alpha=I(0.5))#The plot shows the residuals of the model ag
```



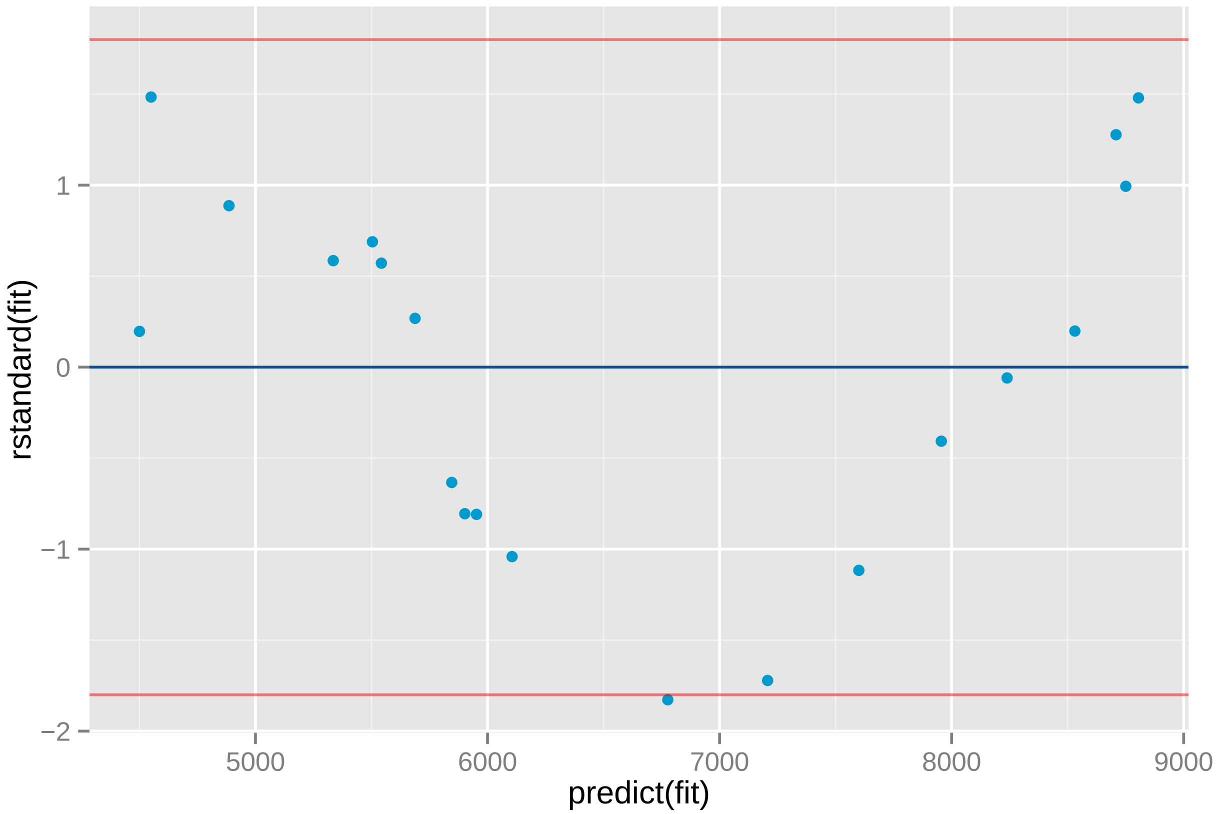
A2: Checking Normality

```
# histogram
q1 = qplot(rstandard(fit), geom="blank") +
  geom_histogram(aes(y=..density..), fill=I("thistle1"), colour=I("black"), binwidth=0.5)+
  stat_function(fun=dnorm, args=list(mean=0, sd=1),
                colour=I("deeppink4"), alpha=I(0.8))
# qqplot
q2 = qplot(sample=rstandard(fit)) +
  geom_abline(slope=1,intercept=0, colour=I("violetred"))

grid.arrange(q1, q2, nrow=1)#The residuals are very close to be distrubuted following a normal distribu
```



A3: Checking Homoscedasticity (equal variance)

```
qplot(predict(fit), rstandard(fit), geom="point", colour=I("deepskyblue3")) + geom_hline(yintercept=0, 
  geom_hline(yintercept=1.8, colour = I("red"), alpha=I(0.5)) +
  geom_hline(yintercept=-1.8, colour = I("red"), alpha=I(0.5)) #se cumple. variances remain similar.
```



A4: Checking Independence

- H0 : errors are not autocorrelated
- H1 : errors are autocorrelated (dependent)

```
durbinWatsonTest(fit)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1        0.753222     0.3857653       0
##  Alternative hypothesis: rho != 0
```

```
#We are using Durbin-Watson test to check independence. If p-value is >= 0.05 then we cannot reject tha
#In this case, p-value is 0, which means that we can reject that the variables are independent.
```

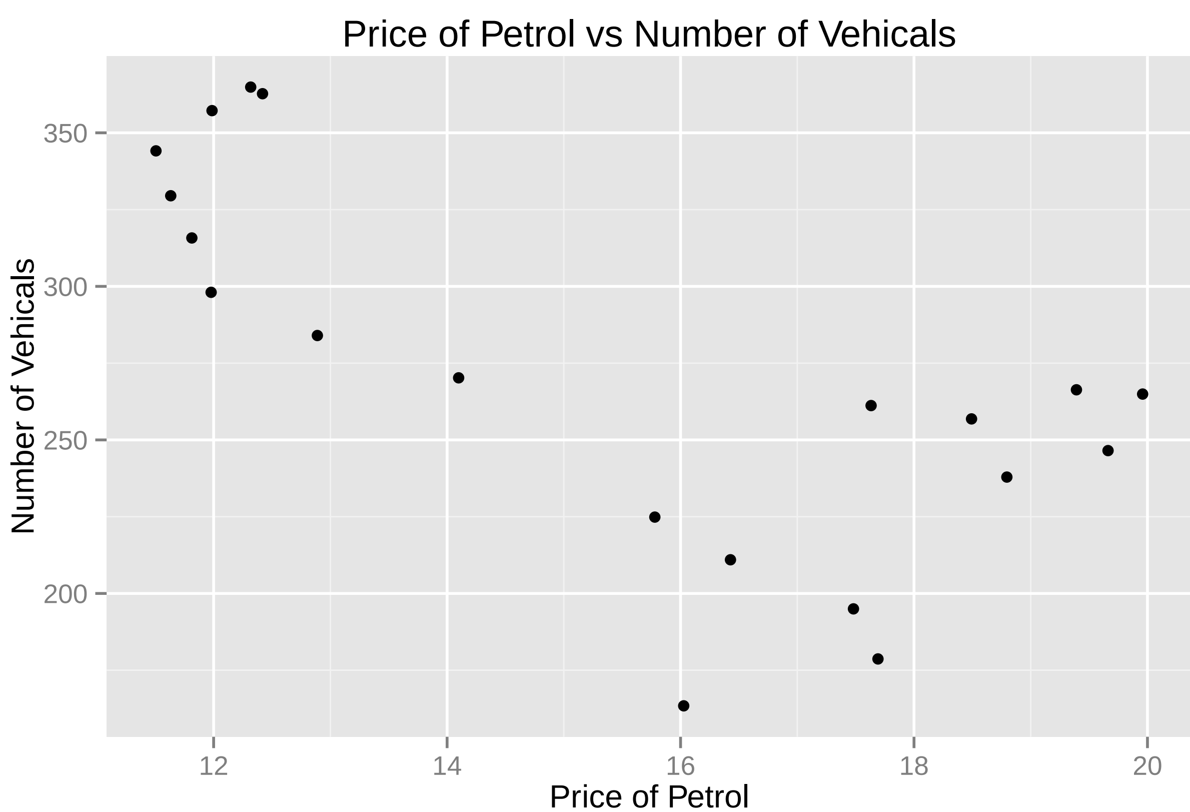
STEP 5: Test the parameters and analyze the multiple correlation and the coefficient of determination for the final model

```
# We again consider the R-squared value and the p-vaules in our final model.
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X1_Price_of_Petrol +
```

```
##     X5_Number_Petrol_veh), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -381.62 -158.13   37.34  149.66  290.91
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3710.811    600.486   6.180 1.01e-05 ***
## X..X1_Price_of_Petrol -141.970     22.166  -6.405 6.53e-06 ***
## X5_Number_Petrol_veh    18.754      1.137  16.498 6.78e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 216.4 on 17 degrees of freedom
## Multiple R-squared:  0.9812, Adjusted R-squared:  0.979
## F-statistic:   444 on 2 and 17 DF,  p-value: 2.124e-15
```

```
# We can see that the p-values for both coefficents are below 0.05 and our R-squared value is 99.64 per

qplot(data = transformed, transformed$X..X1_Price_of_Petrol, transformed$X5_Number_Petrol_veh, xlab = "
```




```
# As we expected, there is no apparent linear relationship between x1 and x2.

# We could also use the VIF function again. As a rule of thumb, a VIF value above 10 indicates multi-co

vif(fit)
```

```
## X..X1_Price_of_Petrol  X5_Number_Petrol_veh
##             1.930757             1.930757
```

STEP 6: Forecast

```r
# Finally, we turn to forecasting using linear regression. We have to assume here that the change to th

# First, we create new dataframe with data to go into row twenty-one We use assumptions in the case to
transformed1 = data.frame(
  YEARS = 21,
  y_Petrol.consumption = NA,
  X..X1_Price_of_Petrol = transformed[20, 3]*1.035,
  X2_Bus_Fares = transformed[20, 4]*1.015,
  X..X3_.DPC.habit. = transformed[20, 5]*1.018,
  X4_Number_Tourists = transformed[20, 6]*0.99,
  X5_Number_Petrol_veh = transformed[20, 7]*1.02
  )

# create a new dataframe with row 21 for use in our model.
transformed1_new = data.frame(transformed1$X..X1_Price_of_Petrol, transformed1$X5_Number_Petrol_veh)

# Predict values with fit
names(transformed1_new) = c("X..X1_Price_of_Petrol","X5_Number_Petrol_veh")
predict_fit = predict(fit, newdata = transformed1_new)
summary(predict_fit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8881    8881    8881    8881    8881    8881
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = y_Petrol.consumption ~ (X..X1_Price_of_Petrol +
##     X5_Number_Petrol_veh), data = transformed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -381.62 -158.13   37.34  149.66  290.91
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3710.811    600.486   6.180 1.01e-05 ***
## X..X1_Price_of_Petrol -141.970     22.166  -6.405 6.53e-06 ***
## X5_Number_Petrol_veh    18.754      1.137  16.498 6.78e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 216.4 on 17 degrees of freedom
## Multiple R-squared:  0.9812, Adjusted R-squared:  0.979
## F-statistic:   444 on 2 and 17 DF,  p-value: 2.124e-15
```

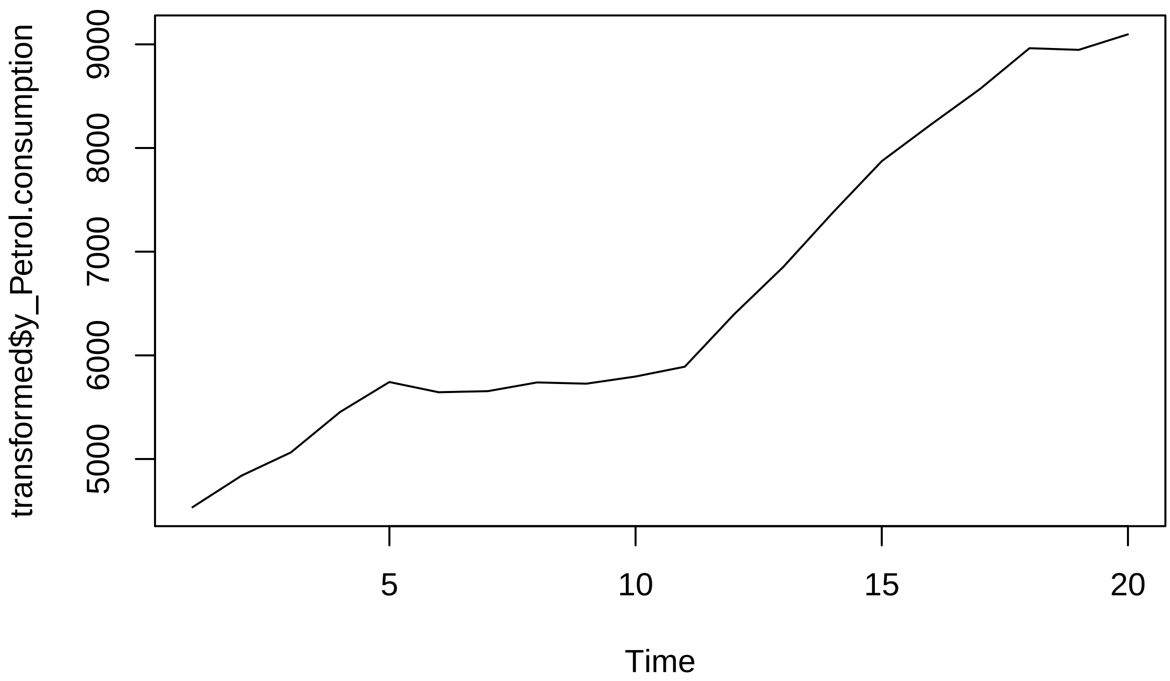
```
# Our predicted value for petrol consumption in year 21 in 8881 (in thousands of tons)

# NOTE: I am really questioning how we got this value. It makes no sense that in part two we are gettin

##########################################################################################
#Part 2

# Part two asks us to refine our prediction assuming that price is given by a normal distribution with 

# First step is to run simulation to get 100 varibles using a normal distribution with mean = 70 and sd
price_data = rnorm(100,70,6.67)
price_data = price_data/7.03

# create new dataframe and run a prediction on it
dataframe_part2 = data.frame(price_data[1:100], transformed1$X5_Number_Petrol_veh)
names(dataframe_part2) = c("X..X1_Price_of_Petrol","X5_Number_Petrol_veh")
predict_part2 = predict(fit, newdata =  dataframe_part2)
summary(predict_part2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9007    9188    9291    9301    9387    9651
```

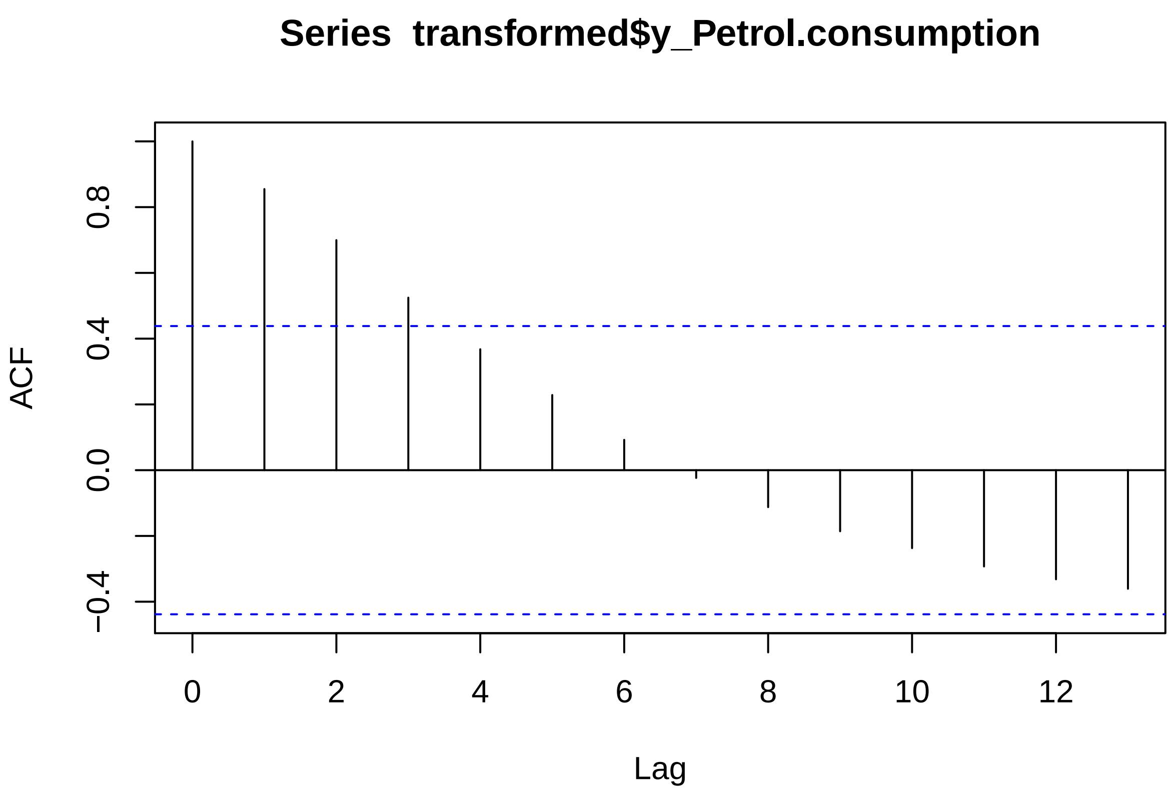
```
# calculate confidence intervals for part 2
lower = mean(predict_part2)-2*sd(predict_part2)/sqrt(length(predict_part2))
upper = mean(predict_part2)+2*sd(predict_part2)/sqrt(length(predict_part2))

# The confidence intervals indicate that we can say that 95% of the time the value of the the consumpti
```

Step 7: Time series based predictions

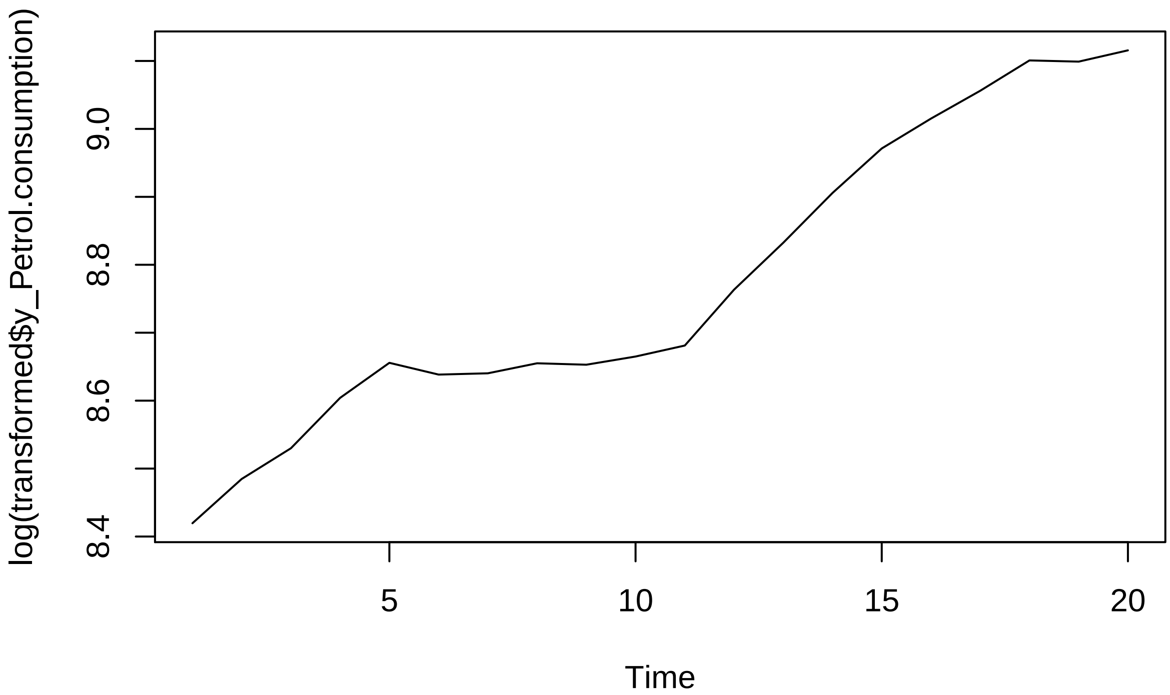
```
#Lets plot the dataset first
ts.plot(transformed$y_Petrol.consumption)
```



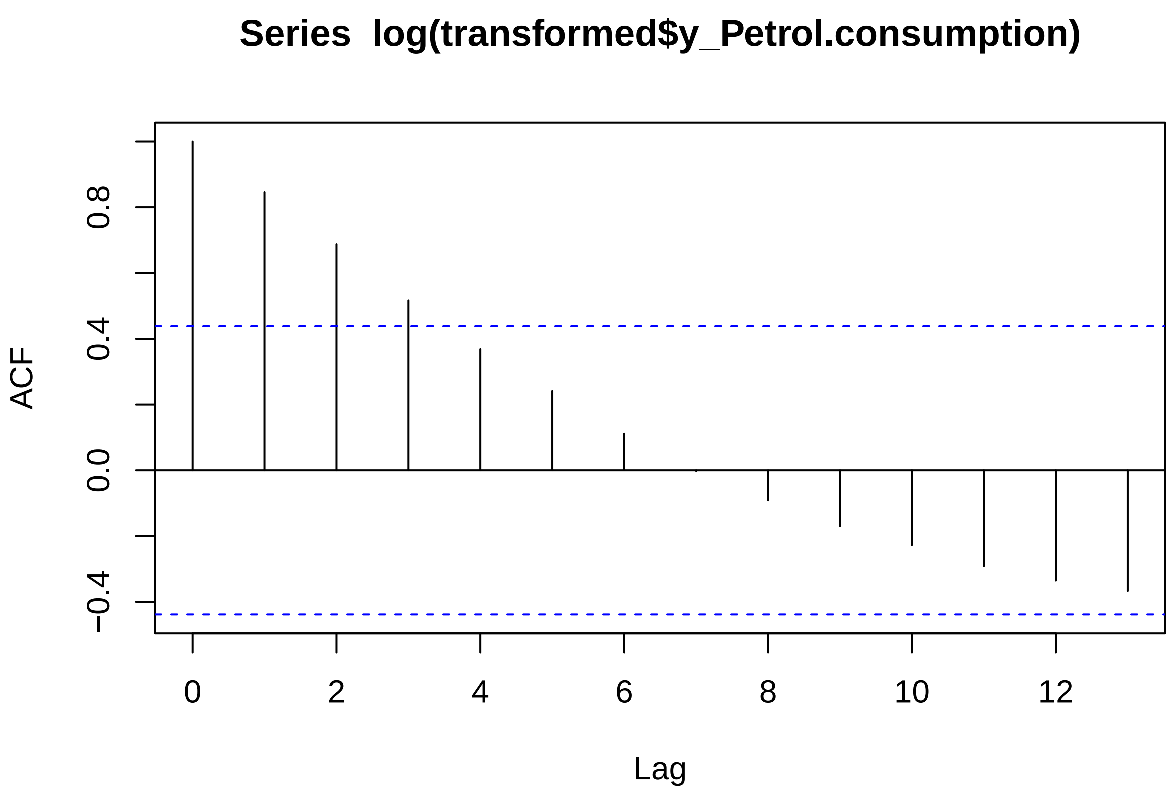

```r
#Lets look at the autocorrelation of the time series
acf(transformed$y_Petrol.consumption)
```




```r
#There seems to be not very much periodicity in the data.
#Lets plot the log of the dataset
ts.plot(log(transformed$y_Petrol.consumption))
```




```r
#There seems to no periodic change in the log graph
acf(log(transformed$y_Petrol.consumption))
```




```r
#Lets fit this to an ARIMA Model
petrol.ar = ar.yw(log(transformed$y_Petrol.consumption))
#Lets look at the order of the ARIMA model
```

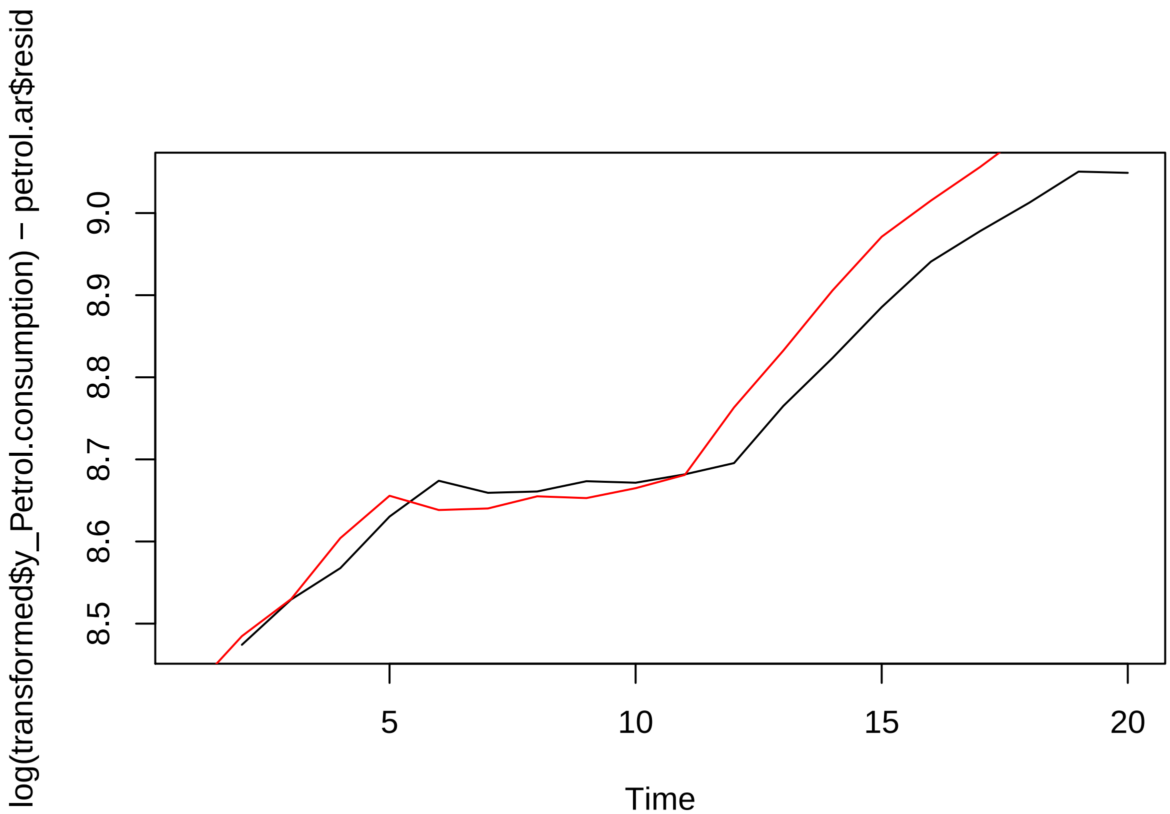
```r
petrol.ar$order.max
```

```
## [1] 13
```

```r
#Lets look at the ARIMA Model arrived at
petrol.ar$aic
```

```
##         0         1         2         3         4         5         6
## 23.142758  0.000000  1.806136  3.419894  5.402111  7.379643  9.083799
##         7         8         9        10        11        12        13
## 11.008043 12.999096 14.884706 16.847036 18.553387 20.519725 22.447443
```

```r
#Lets plot the ARIMA model vs the original time series
ts.plot(log(transformed$y_Petrol.consumption)-petrol.ar$resid)
lines(log(transformed$y_Petrol.consumption),col=2)
```




```r
# As we can see, this is a pretty good fit. The black line is the model predicted by ARIMA
#Lets forecast now
petrol_predict = predict(petrol.ar,n.ahead = 1)
#Since we looked at the logarithmic values we must use a reverse transform
exp(petrol_predict$pred)
```

```
## Time Series:
## Start = 21
## End = 21
## Frequency = 1
## [1] 8630.198
```

Step 8: Forecasting quarterly

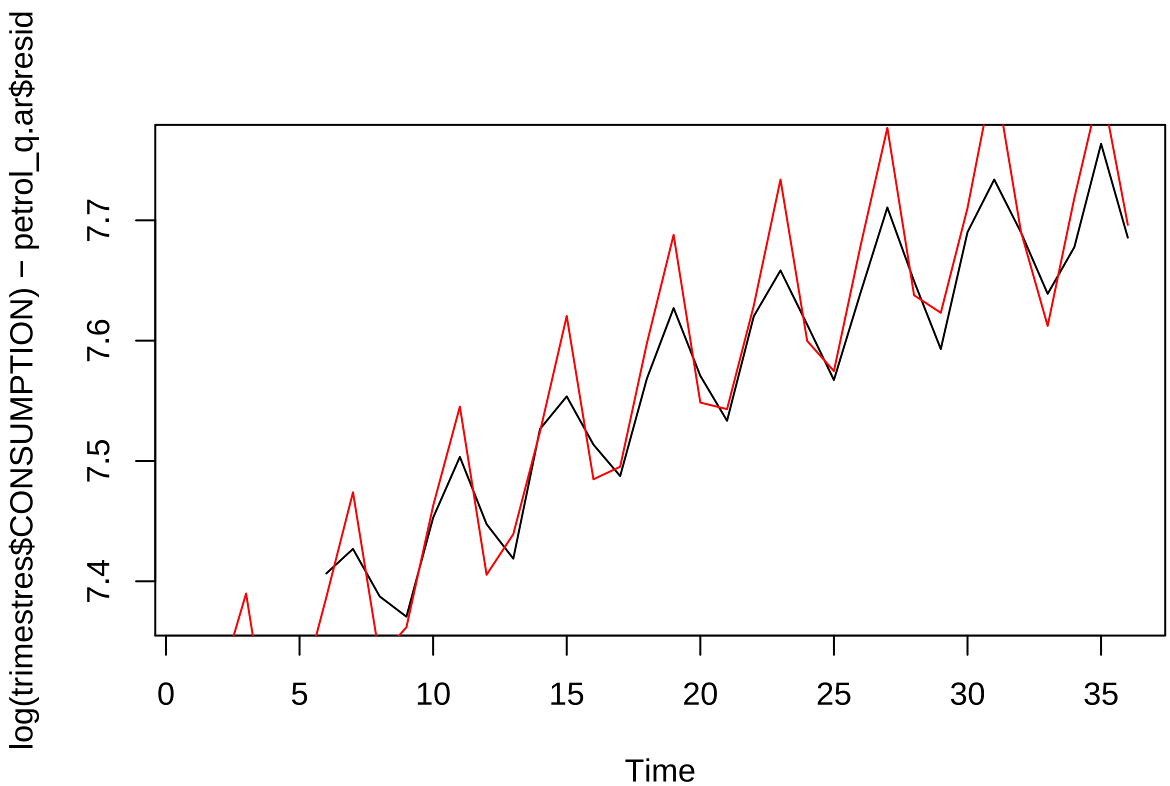
```
petrol_q = ts(trimestres)
#Look at the data
str(trimestres)
```

```
## 'data.frame':    36 obs. of  3 variables:
##  $ YEARS      : num  1 1 1 1 2 2 2 2 3 3 ...
##  $ QUARTERS   : Factor w/ 4 levels "I","II","III",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ CONSUMPTION: num  1361 1502 1619 1408 1484 ...
```

```
petrol_q.ar = ar.yw(log(trimestres$CONSUMPTION))
#Look at the ARIMA model
petrol_q.ar$aic
```

```
##         0         1         2         3         4         5         6
## 38.465635 12.047238 14.024801  4.661135  4.745238  0.000000  1.022254
##         7         8         9        10        11        12        13
##  2.945537  4.522615  5.642824  7.503278  9.400850 11.397470 13.325218
##        14        15
## 15.148184 17.088199
```

```
#Plot the ARIMA Model
ts.plot(log(trimestres$CONSUMPTION)-petrol_q.ar$resid)
lines(log(trimestres$CONSUMPTION),col=2)
```



```r
#Forecast the next year's consumption
petrol_q_predict = predict(petrol_q.ar,n.ahead=4)
#Taking the exponent and adding the four quarterly predicted values
sum(exp(petrol_q_predict$pred))
```

```
## [1] 8581.089
```