IE 360 PROJECT REPORT

FORECASTING GASOLINE AND DIESEL SALES

Ayşe Korkmaz - 2018402027
Ece Tuana Hezer - 2018402186
Fatmanur Yaman - 2019402204
Murat Tutar - 2020402264

TABLE OF CONTENT

1. Introduction

Time series forecasting involves creating a statistical model based on available data to make predictions. There are multiple models available for time series forecasting, and the choice of model depends on its performance. However, to improve the accuracy of predictions, certain steps need to be followed, and the results of each step should be analyzed.

The applications of time series forecasting are extensive, encompassing not only business decisions but also decisions in various aspects of social life. Business-related decisions, in particular, rely heavily on time series forecasting. Regardless of the industry, there is always a need to predict variables such as ice-cream for the upcoming summer or water consumption for the following week. These predictions aid in making informed decisions and planning strategies accordingly.

In this project we are expected to predict gasoline and diesel sales. Gasoline and diesel sales depend on various things, some of which are the average price (adjusted with an index) of a liter of unleaded and diesel gasoline, number of unleaded and diesel gasoline using vehicles in the traffic, agriculture and commerce component of Gross National Product, and grand total for GNP.

Based on the available data and our understanding of the problem, we have chosen to experiment with various models in order to improve our understanding and make more accurate interpretations.

Approach

We visually examined the data's attributes and analyzed its characteristics. Firstly, we looked for plots of UGS and DGS values. As it can be seen from Appendix-1, UGS values show a seasonality. It has its peak values in July, and lowest values in January. Overall, it has a decreasing trend and its variability is not changing significantly. On the contrary, DGS values have been increasing over the years. DGS also has the peak values in July and lowest values in January. Data shows seasonality over 7 years. Both plots do not look stationary because both have trends. Expected value of the data is time dependent and changing over the years. Variance of the data is not increasing or decreasing on both plots. However, strong correlation can be seen at each season which means both UGS and DGS have non stationary series.

When we analyze the autocorrelation of the UGS data (Appendix-3), lag 0 is 1 naturally. At lag 1 and lag 4, there is correlation. Lag 4 correlation can be explained by seasonality since every quarter is in a relation with previous years. Partial autocorrelation shows at lags 1, 3 and 5 there is correlation (Appendix-4). There is a strong autocorrelation in the data. There is a need to include these lags in the model to make valuable predictions. We can explain this relation with the decreasing trend.

Autocorrelation plot of DGS data shows strong correlation at lags 4 and 8 (Appendix-5). This means each summer is affected by last year's summer and the year before summer values. This explains the seasonality effect on the plot of DGS. Partial autocorrelation plot shows that lags 3,4 and 5 are correlated with current value (Appendix-6). Which means this year's summer value is affected by last year's summer, spring and fall values. This can be explained with the trend over the years.

2. Different Models

2.1 Forecasting With Time Series Analysis

In method A, we are expected to make forecasts with time series analysis. First, we searched if the data is stationary. The Augmented Dickey-Fuller Test was performed to assess the stationarity of the UGS time series data. The test statistic value obtained was -2.3684, with a lag order of 3. The p-value associated with the test was 0.4316. Therefore, we fail to reject the null hypothesis: data is not stationary. To make this data stationary, we took the first difference of the values. The plot of the first difference UGS does not show a trend but it has seasonality (Appendix-7). When we look at the autocorrelation function, at lags 2, 4, 6, 8, 10 and 12 there is strong correlation (Appendix-8). This is the effect of seasonality. Partial autocorrelation function also shows that lags 2 and 3 are affecting the current value (Appendix-9). To check if 1st differenced data is stationary, we performed Augmented Dickey-Fuller Test resulting in p-value smaller than 0.01. This indicates that we can reject the null hypothesis and say data is stationary. To remove the correlation from the data, we took the 4th difference of the 1st difference data. Then, the autocorrelation plot shows no significant relation with this data (Appendix-10). Similarly, the partial autocorrelation function does not show any significant effect (Appendix-11). To make sure this data is stationary, we performed a KPSS test and the p-value obtained was greater than 0.1. For alpha value 0.05 we fail to reject null hypothesis: data is trend stationary. This indicates that current data is stationary.

Similar to methods applied to UGS data, we tried to make sure we will forecast on stationary DGS data. First, we performed Augmented Dickey-Fuller Test and this test showed that p-value was 0.911 meaning that we fail to reject that data is not stationary. To make this data stationary, we took the first difference and checked the plot, autocorrelation and partial autocorrelation function. The plot doesn't show a significant trend but still there is an increase of the values over the years (Appendix-12). We can see the seasonality on this graph too. Autocorrelation graphs show correlations with lags 2, 4, 6, 8, 10 and 12 (Appendix-13). Partial autocorrelation graph shows a correlation at lags 2, 3, 4 and 5 (Appendix-14). To check if this data is stationary, we performed Augmented Dickey-Fuller Test and p-value is smaller than 0.01. Therefore, we reject the stationarity of the data. Furthermore, to get the stationary data, we took the 4th difference of the 1st differenced DGS values. The plot does not show any trends or seasonality (Appendix-15). The autocorrelation function and partial autocorrelation function does not show any significant correlation at any lags (Appendix-16 & 17). To make sure that we have stationary data, a KPSS test is performed and the p-value is

greater than 0.1. For alpha 0.05 we fail to reject that data is trend stationary meaning data is stationary.

When we analyze the UGS data, we can say that both autocorrelation and partial autocorrelation functions do not show AR or MA features. For AR, we would expect to see slowly dying out autocorrelation and cut off in partial autocorrelation. For MA, we want to see a cut off at autocorrelation and slowly dying out to 0 in partial autocorrelation. UGS data does not show these features, therefore we started our Arima models with order (0,1,0) and seasonal order (0,1,0). In this model, 1 seasonal 1 regular differencing is done. This model gives AIC=576.67 AICc=576.86 BIC=577.8. After the main model, we started neighborhood search to get the minimum value of AIC, AICc and BIC. Keeping the seasonal order as (0,1,0) we added an autoregressive factor to the regular model (1,1,0). This model gave us AIC=572.59 AICc=573.19 BIC=574.86. Then we added the moving average factor by order (1,1,1) and this model presented AIC=574.58 AICc=575.84 BIC=577.99. In the third model we removed autoregressive order (0,1,1) and the result was AIC=573.35 AICc=573.95 BIC=575.62. The fourth model removes the moving average order and adds another integration (0,2,0). This model gave us AIC=579.43 AICc=579.63 BIC=580.52. In our fifth model we changed the seasonal order to (1,1,0) and kept regular order as (0,1,0) and this model gave us the results as AIC=577.32 AICc=577.92 BIC=579.59. The final model has seasonality order as (0,1,1) and regular order as (0,1,0). This model gives AIC=577.32 AICc=577.92 BIC=579.59. We would like to choose the model that gives the minimum AIC, AICc nd BIC values. Therefore the best model is the model1 with regular order (1,1,0) and seasonal order as (0,1,0). Model 1 autocorrelation and partial autocorrelation graphs do not give any correlation and this is what we want from the model. Also residuals look stationary around 0. Based on these findings, the best model we will continue forecasting UGS is model 1.

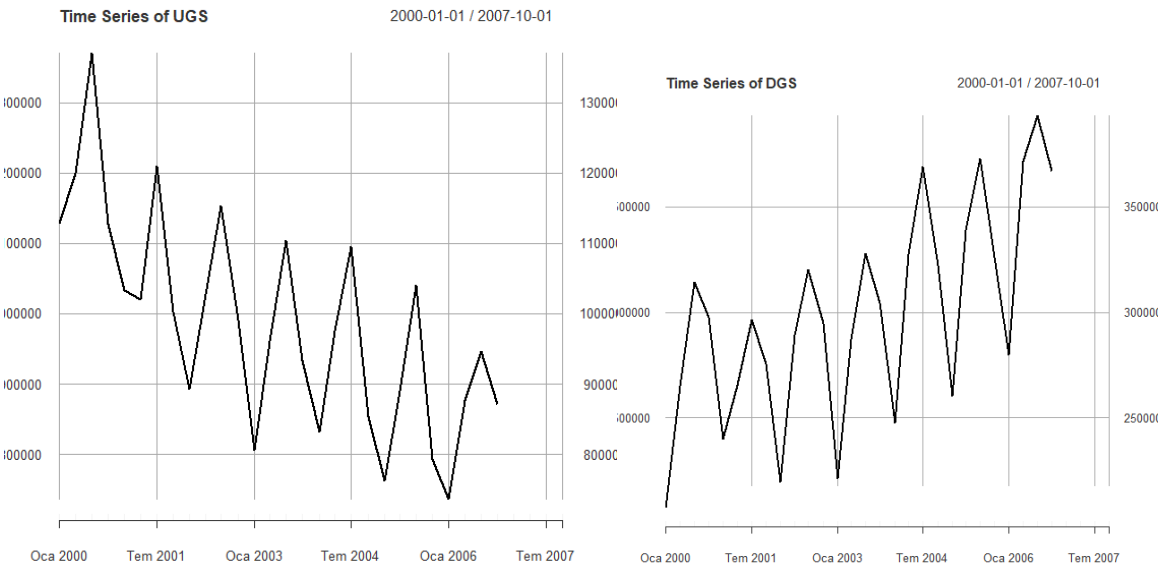
DGS data shows similar characteristics to UGS data. The autocorrelation and partial autocorrelation plots do not show AR or MA features. Therefore the main model we are starting with has regular order (0,1,0) and seasonal order as (0,1,0). This model gives AIC=618.4 AICc=618.59 BIC=619.53. Then we started neighborhood searching by adding AR factor ro the regular order as (1,1,0) and the model gives AIC=619.48 AICc=620.08 BIC=621.75. We added the MA factor to the second model (1,1,1) and this model resulted in AIC=621.32 AICc=622.58 BIC=624.72. On the third model we subtracted AR factor and (0,1,1) model gave us AIC=619.32 AICc=619.92 BIC=621.59. We subtracted MA factor and doubled integration on the fourth model (0,2,0) and this model gave us AIC=607.55 AICc=607.75 BIC=608.64. On the fifth model we kept regular order as (0,1,0) and added AR to seasonal order (1,1,0). This model shows AIC=616.23 AICc=616.83 BIC=618.5. While keeping regular order as (0,1,0) we changed seasonal order to (0,1,1) for the last model. This model gives AIC=616.82 AICc=617.42 BIC=619.09. By comparing AIC, BIC and AICc values the minimum is provided by model 4. Therefore to make forecasts the best model is the model 4 with regular order (0,2,0) and seasonal order (0,1,0). This model does not show significance on partial autocorrelation and autocorrelation plots and its residual values are around 0 and variance is lower than other models.

After finding the best models to make forecasts for UGS and DGS data, we used the forecast function to retrieve 2007 forecasts. At the very beginning, we deleted the last 4 rows since they were empty on UGS and DGS cells. Now we are making forecasts for these values using the best Arima models. The UGS and DGS point forecast for all quarters of 2007 are shown in Table 1 below and plots can be found in Appendix-18 & 19.

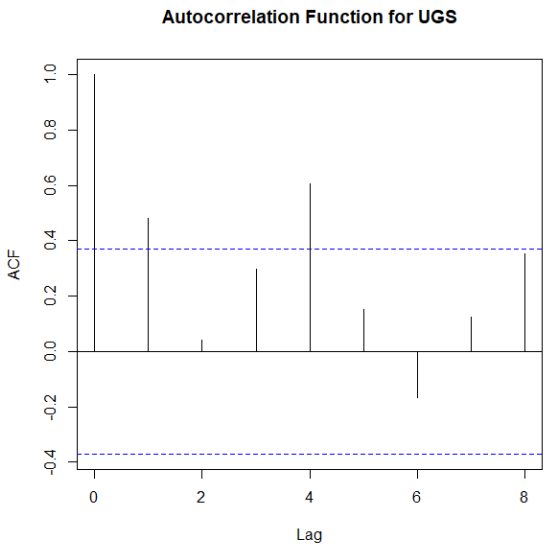
Point Forecasts	UGS	DGS
2007_Q1	713644.6	3414795
2007_Q2	914353.3	4538119
2007_Q3	948450.4	4959466
2007_Q4	894279.8	4903948

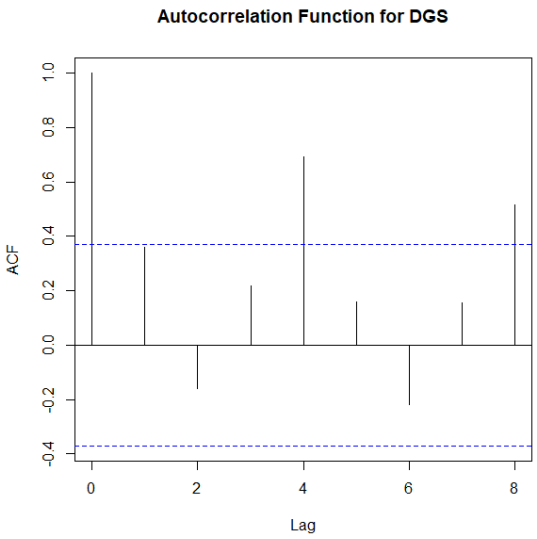
Table:1 Point Forecasts using Time Series Analysis

2.2 Forecasting with Regression



The time series for UGS and DGS are above. It is observed that both of them are non-stationary. UGS seems to have a decreasing trend with a significant seasonality, whereas DGS has an increasing trend with seasonality. There seems to be no need for logarithmic transformation for regression. At first sight, it seems that the use of DGS is replacing the use of UGS. Also, the seasonality of each time series are similar. The autocorrelation functions below also support this claim.





The autocorrelation functions are very similar. Autocorrelation at lag 4 is highest in both, indicating that the period for seasonality is 4 quarters, or a year. So, the use of UGS and DGS at a quarter in a year is correlated with the use of UGS and DGS at the same quarter of another year. Additionally, autocorrelations are also high for lag 1, indicating that each quarter of a year is correlated with the previous quarter or the next quarter.

We first write a code (Code 1 in the appendix) to find a regression but at first we did not include variables for seasonality or time. We first checked the correlations between existing independent variables and eliminated the ones that have high correlation except one of them. This is done by the code snippet below:

```
correlation_matrix <- cor(df[,4:12])  # variables are in columns 4
to 12
highly_correlated <- findCorrelation(correlation_matrix, cutoff =
0.3, verbose=TRUE)
#removing the highly correlated variables
df <- df[-(highly_correlated+3)]
```

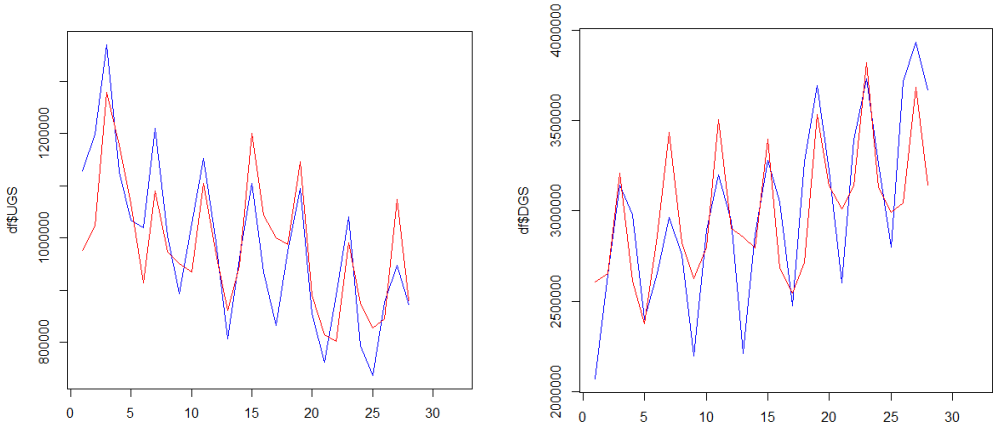
This way, we eliminated most of the independent variables and only had three independent variables remaining for the forecasting. Then, by looking at the p-values, and if $p<0.05$ we chose the variable as significant for UGS (and then DGS). This is done by the code below:

```
#for UGS
UGS_fit <- stepAIC(lm(UGS[1:28] ~ ., data = df[1:28,c(2,4,5,6)]),
direction = "both")
summary(UGS_fit)
#removing RNUV as it has high p-value
UGS_reg<-df[c(2,5,6)]
UGS_reg_model<-lm(UGS ~ ., data = UGS_reg)
summary(UGS_reg_model)

# for DGS
DGS_fit <- stepAIC(lm(DGS[1:28] ~ ., data = df[1:28,c(3,4,5,6)]),
direction = "both")
summary(DGS_fit)
#RNUV has high p-value, therefore removed
DGS_reg<-df[c(3,5,6)]
DGS_reg_model<-lm(DGS~., data=DGS_reg)
summary(DGS_reg_model)
```

After that, we observed that the best predictor variables for the sales of both UGS and DGS were PU and GNPA. Then, based on these variables, we performed a regression. Before forecasting, we would like to visually observe the performance of our regression by comparing it to the known data. This is done by the code below:

```
#Comparing regression to actual observations for UGS
plot(df$UGS, type="l", col="blue")
lines(predict(UGS_reg_model), col="red")

#Comparing regression to actual observations for DGS
plot(df$DGS, type="l", col="blue")
lines(predict(DGS_reg_model), col="red")
```

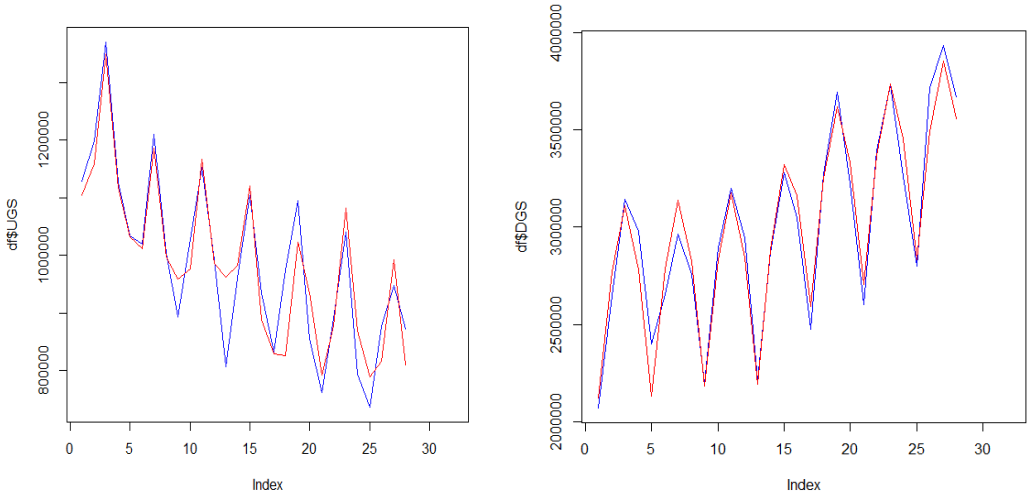
The output graphs of the code are below. The first graph is for UGS and the second is for DGS. In each graph, red line corresponds to regression and blue line corresponds to actual observations.



Although the regression above is not extremely bad, it is seen that it is not very successful as well. We evaluated the possible reasons for this. Then, to improve the regression, we decided to add a seasonality component to our regression equation. We decided not to add a trend component because it seems unnecessary, the errors don't seem to be due to the lack of trend component. Also, we thought about a variable for autocorrelation lags, however they will have very similar effects with the seasonality component we add; and they will be very correlated so therefore we decided to add only the seasonality component.

For seasonality, we added four variables corresponding to each quarter of the year; since the seasonalities are per quarter (and autocorrelation is lag 4, supporting this). The variables are called q1, q2, q3, and q4. They are binary variables, their value is 1 if the corresponding quarter is their quarter of the year; and otherwise 0. To see exactly how we added them, Code 2 in the Appendix can be seen. After adding them, we performed the same operations we did before. This time, the best variables for predicting UGS were RNUV, LPG, and q3. The best variables for predicting DGS were LPG, q1, and q3. It is easily seen that the quarter variables have significant effect; indicating that we did the right thing to add them as new variables and the effect of seasonality is strong.

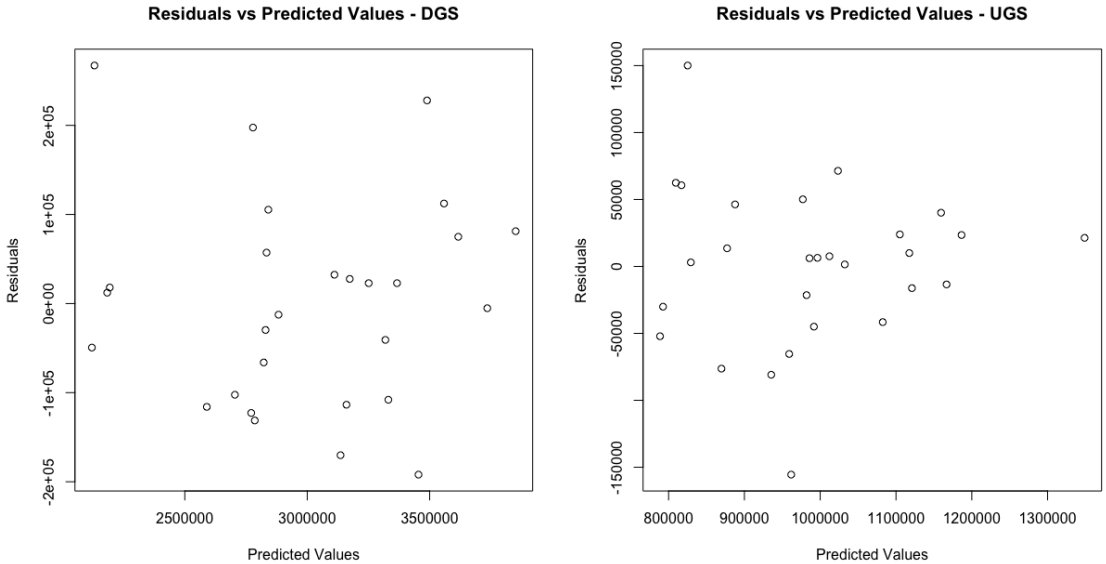
The actual(blue) vs forecast(red) graphs for UGS and DGS are below, respectively.



It is easily observed that the new regression including variables for seasonality is much more accurate than the previous one, although it still has some errors. We can add more variables to decrease this error, however adding new variables to the regression has a cost. It increases the computational time needed and it makes the calculations more complex and difficult. It may also cause overfitting. The regressions above are sufficient enough in the sense that they catch the level, trend, and seasonality of the actual sales of UGS and DGS; therefore we conclude that this is an appropriate regression model for forecasting the sales of UGS and DGS.

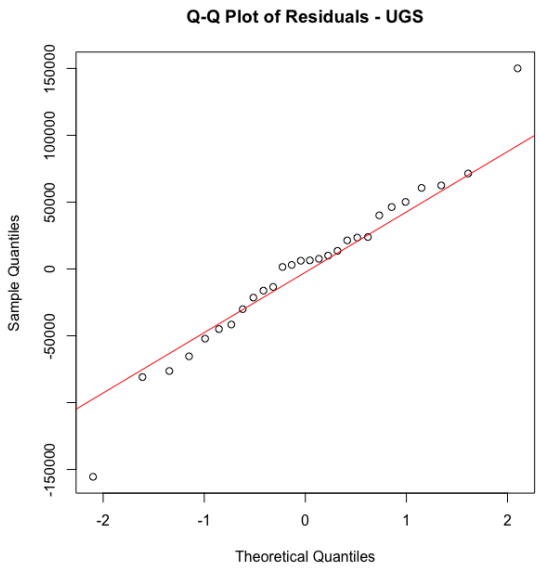
Residual Analysis

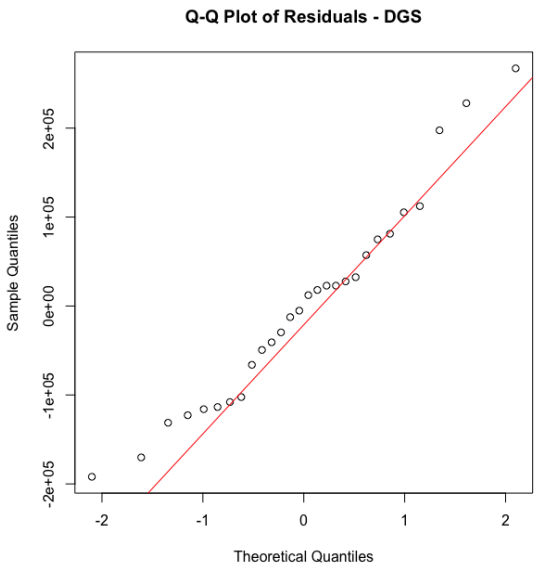
Below are the scatter plots of Residuals vs Predicted Values for both the DGS and UGS models. It can be observed that there is no apparent pattern, non-linearity, or heteroscedasticity in the plots. It appears that the residuals are randomly distributed around zero, indicating that the models are adequately capturing the relationship between the predictors and the response variables.



Q-Q plots

Below are the Q-Q plots for both the UGS and DGS models. The points are roughly following a straight line. The number of outliers are small.





Durbin-Watson Test

Below is the output of the Durbin-Watson Test for USG model. The DW statistic has a value of 1.8359, which is close to 2 and the p-value is 0.2342, which means there is no significant evidence to reject the null hypothesis of there is an autocorrelation present in the UGS model's residuals. The independence of residuals is reasonably satisfied in the UGS model.

```
data:  UGS_reg_model
DW = 1.8359, p-value = 0.2342
alternative hypothesis: true autocorrelation is greater than 0
```

Below is the output of the Durbin-Watson test for DSG model. With DW statistic of 1.3418 and a value of 0.03771, with 0.05 as a significance level, there is significant evidence to reject the null hypothesis. The test results suggest that there is a significant positive autocorrelation present in the DGS model's residuals.

```
data:  DGS_reg_model
DW = 1.3418, p-value = 0.03771
alternative hypothesis: true autocorrelation is greater than 0
```

Collinearity Checks

Below is the results of Variance Inflation Factor to quantify the multicollinearity for UGS model, as VIF value closer to 0 indicates low multicollinearity (little correlation between the variables). Here, all VIF values for three variables are close to 1, so that there is no significant multicollinearity among these variables of UGS model.

```
    RNUV       LPG        q3
1.037483 1.035526 1.001964
```

Below is the results of VIF. For DGS model, again all the VIF values are close to 1, so that there is no significant multicollinearity among these variables.

```
     LPG        q1        q3
1.005471 1.131071 1.125300
```

Model Coefficients

Below is the summary of UGS model. All the p-values are very small which indicates that all three variables are statistically significant. Also, R-squared value is ~84.9%, relatively high percentage of

the variance in UGS can be explained by these variables. The F-statistic of 45.23 with p-value of 4.969e-10 indicates that the overall model is statistically significant.

```
Residuals:
    Min      1Q  Median      3Q     Max
-155510  -32950    6276   27948  150056

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.323e+06  5.444e+04  24.305  < 2e-16 ***
RNUV         9.232e+06  2.008e+06   4.598 0.000115 ***
LPG         -3.752e-01  4.302e-02  -8.721 6.62e-09 ***
q3           1.889e+05  2.713e+04   6.961 3.37e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 62110 on 24 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.8497,     Adjusted R-squared:  0.8309
F-statistic: 45.23 on 3 and 24 DF,  p-value: 4.969e-10
```

Below is the summary of the DGS model. Again all the p-values are very small, which indicates that all the variables are statistically significant to explaing the DGS’s. The R-squared is ~94.5% (higher than UGS model) is very high that 94.5% of the variance in DGS can be explained by these variables. Also, F-statistic value of 139.1 with p-value of 2.633e-15 indicates that the model is statistically significant.

```
Residuals:
    Min      1Q  Median      3Q     Max
-192000 -103775    3497   61633  267157

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.790e+06  1.098e+05  16.304 1.74e-14 ***
LPG          1.043e+00  8.372e-02  12.460 5.71e-12 ***
q1          -6.497e+05  5.693e+04 -11.412 3.51e-11 ***
q3           3.372e+05  5.679e+04   5.939 3.96e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 122700 on 24 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.9456,     Adjusted R-squared:  0.9388
F-statistic: 139.1 on 3 and 24 DF,  p-value: 2.633e-15
```

After deciding an appropriate model, we made the forecasts for the four quarters of 2007. This is done by the code below, its output is also seen. According to that, the forecasts for each quarter of 2007 for

UGS and DGS can be seen below.

```
> UGS_forecast <- predict(UGS_reg_model, newdata = df[29:32,c(2,4,5,8)])
> UGS_forecast #UGS sales forecasts for 2007
       1        2        3        4
747789.3 773513.0 947987.5 762893.5
> DGS_forecast <- predict(DGS_reg_model, newdata = df[29:32, c(3,5,6,8)])
> DGS_forecast #DGS sales forecasts for 2007
      1       2       3       4
2928861 3589281 3953391 3664490
```

3. Comparison of Methods A and B

To compare the method A with time series to method B with regression, we decided to use several error metrics such as RMSE and MAE. The results for method A and method B are below, respectively.

Method A:

```
> RMSE(rep(0,22),diff(ts.ugs.diff.diff4)[2:23])
[1] 101931.1
> MAE(rep(0,22),diff(ts.ugs.diff.diff4)[2:23])
[1] 79666.45
> RMSE(rep(0,22),diff(ts.dgs.diff.diff4)[2:23])
[1] 193163.2
> MAE(rep(0,22),diff(ts.dgs.diff.diff4)[2:23])
[1] 154461.3
```

Method B:

```
> RMSE(predict(UGS_reg_model),df$UGS[1:28])
[1] 57501.23
> MAE(predict(UGS_reg_model),df$UGS[1:28])
[1] 42714.89
> RMSE(predict(DGS_reg_model),df$DGS[1:28])
[1] 113555.5
> MAE(predict(DGS_reg_model),df$DGS[1:28])
[1] 89982.59
```

As can be seen from the results, method B has lower error rates than method A in both calculations, indicating that it is a more accurate method having a better performance. This was indeed the expected result. To understand why, instead of a quantitative perspective, we can approach the problem with a qualitative perspective. In our time series forecasting, we did not use the benefits of using regression whereas in our regression analysis we added several components for seasonality as independent variables (q1, q2, q3, q4 for each quarter). By using -so to say- a hybrid version of the two approaches, we combined the advantages of them and minimized the effect of the drawbacks of each method. This way, we obtained a more accurate result in Method B.

4. APPENDIX

CODE 1:

```
library(readxl)
library(xts)
library(MASS)
library(tseries)
library(ggplot2)
library(forecast)
library(caret)

# Read the data
data_path <- "C:/Users/ASUS/Downloads/IE360-ProjectData.xlsx"
df <- read_excel(data_path)

# Assign shorter names to the columns
df$UGS <- df$`Unleaded Gasoline Sale (UGS)`
df <- df[, !(colnames(df) %in%"Unleaded Gasoline Sale (UGS)")]
df$DGS <- df$`Diesel Gasoline Sale (DGS)`
df <- df[, !(colnames(df) %in% "Diesel Gasoline Sale (DGS)")]
colnames(df)
df$NDGV <- df$`# of Diesel Gasoline Vehicles (NDGV)`
df <- df[, !(colnames(df) %in% "# of Diesel Gasoline Vehicles
(NDGV)")]
df$PU <- df$`Price of Unleaded Gasoline (PU)`
df <- df[, !(colnames(df) %in% "Price of Unleaded Gasoline (PU)")]
df$PG <- df$`Price of Diesel Gasoline (PG)`
df <- df[, !(colnames(df) %in% "Price of Diesel Gasoline (PG)")]
df$NUGV <- df$`# Unleaded Gasoline Vehicles (NUGV)`
df <- df[, !(colnames(df) %in% "# Unleaded Gasoline Vehicles
(NUGV)")]
df$GNPA <- df$`GNP Agriculture`
df <- df[, !(colnames(df) %in% "GNP Agriculture")]
df$GNPC <- df$`GNP Commerce`
df <- df[, !(colnames(df) %in% "GNP Commerce")]
df$GNP <- df$`GNP Total`
df <- df[, !(colnames(df) %in% "GNP Total")]
df$LPG <- df$`# LPG Vehicles (NLPG)`
df <- df[, !(colnames(df) %in% "# LPG Vehicles (NLPG)")]

#creating a Date sequence
dates <- seq(as.Date("2000/1/1"), by = "quarter", length.out =
nrow(df))

#generating time series for UGS and DGS
ugs_ts <- xts(df$UGS, order.by = dates)
dgs_ts <- xts(df$DGS, order.by = dates)

#plotting time series of UGS
plot(ugs_ts, main = "Time Series of UGS")

#plotting time series of DGS
plot(dgs_ts, main = "Time Series of DGS")
```

```
#plotting autocorrelation function for UGS
acf(ugs_ts[1:28], lag.max = 8, main = "Autocorrelation Function for
UGS")

#plotting autocorrelation Function for DGS
acf(dgs_ts[1:28], lag.max = 8, main = "Autocorrelation Function for
DGS")


#Method B
#creating a date sequence
df$Time <- dates
df <- subset(df,select = -c(...1))

df <- df[, c("Time", "UGS", "DGS", "RNUV", "LPG", "PU", "PG",
"NUGV", "NDGV", "GNPA", "GNPC", "GNP")]

#checking for multicollinearity
correlation_matrix <- cor(df[,4:12])  # variables are in columns 4
to 12
highly_correlated <- findCorrelation(correlation_matrix, cutoff =
0.3, verbose=TRUE)
#removing the highly correlated variables
df <- df[-(highly_correlated+3)]


#select most important variables and build the model
#for UGS
UGS_fit <- stepAIC(lm(UGS[1:28] ~ ., data = df[1:28,c(2,4,5,6)]),
direction = "both")
summary(UGS_fit)
#removing RNUV as it has high p-value
UGS_reg<-df[c(2,5,6)]
UGS_reg_model<-lm(UGS ~ ., data = UGS_reg)
summary(UGS_reg_model)

# for DGS
DGS_fit <- stepAIC(lm(DGS[1:28] ~ ., data = df[1:28,c(3,4,5,6)]),
direction = "both")
summary(DGS_fit)
#RNUV has high p-value, therefore removed
DGS_reg<-df[c(3,5,6)]
DGS_reg_model<-lm(DGS~., data=DGS_reg)
summary(DGS_reg_model)

# forecast sales for 2007
predict(UGS_reg_model) #just to see that the regression is accurate
UGS_forecast <- predict(UGS_reg_model, newdata = df[29:32,c(2,5,6)])
UGS_forecast #UGS sales forecasts for 2007

predict(DGS_reg_model)
DGS_forecast <- predict(DGS_reg_model, newdata = df[29:32,
c(3,5,6)])
DGS_forecast #DGS sales forecasts for 2007
```

```
#Comparing regression to actual observations for UGS
plot(df$UGS, type="l", col="blue")
lines(predict(UGS_reg_model), col="red")

#Comparing regression to actual observations for DGS
plot(df$DGS, type="l", col="blue")
lines(predict(DGS_reg_model), col="red")
```

CODE 2:

```
library(readxl)
library(xts)
library(MASS)
library(tseries)
library(ggplot2)
library(forecast)
library(caret)

# Read the data
data_path <- "C:/Users/ASUS/Downloads/IE360-ProjectData.xlsx"
df <- read_excel(data_path)

# Assign shorter names to the columns
df$UGS <- df$`Unleaded Gasoline Sale (UGS)`
df <- df[, !(colnames(df) %in%"Unleaded Gasoline Sale (UGS)")]
df$DGS <- df$`Diesel Gasoline Sale (DGS)`
df <- df[, !(colnames(df) %in% "Diesel Gasoline Sale (DGS)")]
df$NDGV <- df$`# of Diesel Gasoline Vehicles (NDGV)`
df <- df[, !(colnames(df) %in% "# of Diesel Gasoline Vehicles
(NDGV)")]
df$PU <- df$`Price of Unleaded Gasoline (PU)`
df <- df[, !(colnames(df) %in% "Price of Unleaded Gasoline (PU)")]
df$PG <- df$`Price of Diesel Gasoline (PG)`
df <- df[, !(colnames(df) %in% "Price of Diesel Gasoline (PG)")]
df$NUGV <- df$`# Unleaded Gasoline Vehicles (NUGV)`
df <- df[, !(colnames(df) %in% "# Unleaded Gasoline Vehicles
(NUGV)")]
df$GNPA <- df$`GNP Agriculture`
df <- df[, !(colnames(df) %in% "GNP Agriculture")]
df$GNPC <- df$`GNP Commerce`
df <- df[, !(colnames(df) %in% "GNP Commerce")]
df$GNP <- df$`GNP Total`
df <- df[, !(colnames(df) %in% "GNP Total")]
df$LPG <- df$`# LPG Vehicles (NLPG)`
df <- df[, !(colnames(df) %in% "# LPG Vehicles (NLPG)")]


df$q1 <- rep(c(1,0,0,0),8)
df$q2 <- rep(c(0,1,0,0),8)
df$q3 <- rep(c(0,0,1,0),8)
df$q4 <- rep(c(0,0,0,1),8)
```

```
df <- df[, c("...1", "UGS", "DGS", "RNUV", "LPG", "PU", "PG",
"NUGV", "NDGV", "GNPA", "GNPC", "GNP", "q1", "q2", "q3","q4")]

#creating a Date sequence
dates <- seq(as.Date("2000/1/1"), by = "quarter", length.out =
nrow(df))

#generating time series for UGS and DGS
ugs_ts <- xts(df$UGS, order.by = dates)
dgs_ts <- xts(df$DGS, order.by = dates)

#plotting time series of UGS
plot(ugs_ts, main = "Time Series of UGS")

#plotting time series of DGS
plot(dgs_ts, main = "Time Series of DGS")

#plotting autocorrelation function for UGS
acf(ugs_ts[1:28], lag.max = 8, main = "Autocorrelation Function for
UGS")

#plotting autocorrelation Function for DGS
acf(dgs_ts[1:28], lag.max = 8, main = "Autocorrelation Function for
DGS")


#Method B
#creating a date sequence
#df$Time <- dates

#checking for multicollinearity
correlation_matrix <- cor(df[1:28,c(4:16)])  # variables are in
columns 4 to 16
highly_correlated <- findCorrelation(correlation_matrix, cutoff =
0.4, verbose=TRUE)
#removing the highly correlated variables
df <- df[-(highly_correlated+3)] #+3 is for indexing


#select most important variables and build the model
#for UGS
UGS_fit <- stepAIC(lm(UGS[1:28] ~ ., data = df[1:28,c(2,4:9)]),
direction = "both")
summary(UGS_fit)
UGS_reg<-df[c(2,4,5,8)]
UGS_reg_model<-lm(UGS ~ ., data = UGS_reg)
summary(UGS_reg_model)

#for DGS
DGS_fit <- stepAIC(lm(DGS[1:28] ~ ., data = df[1:28,3:9]), direction
= "both")
summary(DGS_fit)
#RNUV has high p-value, therefore removed
DGS_reg<-df[c(3,5,6,8)]
DGS_reg_model<-lm(DGS~., data=DGS_reg)
```

```
summary(DGS_reg_model)

#forecast sales for 2007
predict(UGS_reg_model) #just to see that the regression is accurate
UGS_forecast <- predict(UGS_reg_model, newdata =
df[29:32,c(2,4,5,8)])
UGS_forecast #UGS sales forecasts for 2007

predict(DGS_reg_model)
DGS_forecast <- predict(DGS_reg_model, newdata = df[29:32,
c(3,5,6,8)])
DGS_forecast #DGS sales forecasts for 2007

#Comparing regression to actual observations for UGS
plot(df$UGS, type="l", col="blue")
lines(predict(UGS_reg_model), col="red")

#Comparing regression to actual observations for DGS
plot(df$DGS, type="l", col="blue")
lines(predict(DGS_reg_model), col="red")

#Residual analysis
plot(fitted(UGS_reg_model), residuals(UGS_reg_model), xlab =
"Predicted Values", ylab = "Residuals", title("Residuals vs
Predicted Values - UGS"))

plot(fitted(DGS_reg_model), residuals(DGS_reg_model), xlab =
"Predicted Values", ylab = "Residuals", title("Residuals vs
Predicted Values - DGS"))

#qq plots
qqnorm(residuals(UGS_reg_model), main = "Q-Q Plot of Residuals -
UGS")
qqline(residuals(UGS_reg_model), col = "red")

qqnorm(residuals(DGS_reg_model), main = "Q-Q Plot of Residuals -
DGS")
qqline(residuals(DGS_reg_model), col = "red")

#Durbin-Watson Test
library(lmtest)
residuals_UGS <- residuals(UGS_reg_model)
dw_test_UGS <- dwtest(UGS_reg_model)
dw_test_UGS

residuals_DGS <- residuals(DGS_reg_model)
dw_test_DGS <- dwtest(DGS_reg_model)
dw_test_DGS


#Collinearity Checks
library(car)

vif_values_USG <- vif(UGS_reg_model)
vif_values_USG
```

```
vif_values_DSG <- vif(DGS_reg_model)
vif_values_DSG

#Checking model coefficients
summary(UGS_reg_model)
summary(DGS_reg_model)
```

CODE 3 (METHOD A)

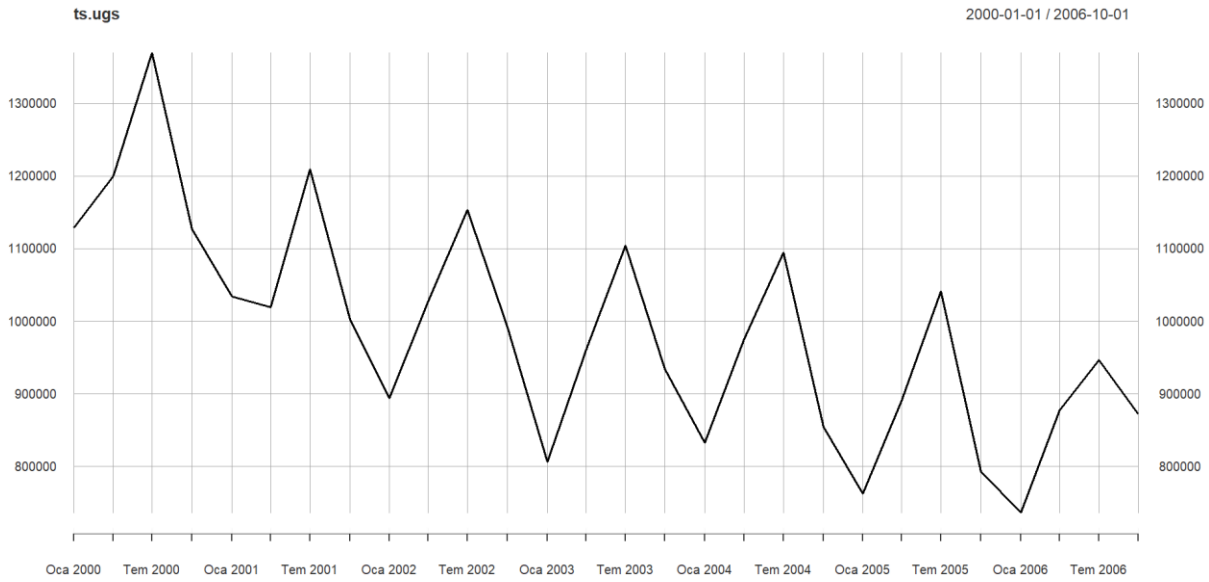
```
library(readxl)
library(xts)
library(MASS)
library(tseries)
library(ggplot2)
library(forecast)
library(caret)

#Load the data.
data<-read_excel("C:/Users/ASUS/Downloads/IE360-ProjectData.xlsx")
head(data) #The first 5 column of the data
tail(data) #The last 5 column of the data
data <- head(data, n=nrow(data)-4) #Remove the last 4 rows because
of NaN values.
tail(data)
#QUESTION 1

#Convert data into time-series.
dates <- seq(as.Date("2000/1/1"), by = "quarter", length.out =
               nrow(data))
ts.ugs <- xts(data$UGS, order.by = dates)
ts.dgs <- xts(data$DGS, order.by = dates)

#Plot the time series for the sales for 2 products.
plot(ts.ugs)
plot(ts.dgs)

#Plot the acf and pacf in order to determine regular and seasonal
differencing to be applied.

acf(ts.ugs,28) #There is a spike at lag 4 indicating an
autocorrelation and seasonality
#Also, strong positive correlation between consecutive observations.

pacf(ts.ugs,28) #There is a spike at lag 1
#on the value 2 time steps in the past after removing the influence
of shorter lags.

acf(ts.dgs,28) #There are significant spikes at seasonal lags of 4.
#The presence of seasonality and need for seasonal differencing.

pacf(ts.dgs,28) #There are spikes at lag 2&3 which indicate that the
current value is dependent
```

```
#on the value 2&3 time steps in the past after removing the
influence of shorter lags.

#The acf and pacf plots for both of 2 products look very similar to
each other.
#The same model can be used to forecast the sales for the upcoming
months.

#METHOD A
#UGS Forecasting

#Question 1 - Try different
adf.test(ts.ugs) #Augmented Dickey-Fuller Test to test the
stationary.

#Taking the first difference.
ts.ugs.diff<-diff(ts.ugs)
ts.ugs.diff<-ts.ugs.diff[-1]
plot(ts.ugs.diff)
acf(ts.ugs.diff)
pacf(ts.ugs.diff)
adf.test(ts.ugs.diff) #showing that we obtained stationary data

#The acf shows the seasonality at lag 4.
ts.ugs.diff.diff4 <- diff(ts.ugs.diff,4)
plot(ts.ugs.diff.diff4)
ts.ugs.diff.diff4<-ts.ugs.diff.diff4[-1:-4]
ts.ugs.diff.diff4
acf(ts.ugs.diff.diff4) #no significantly strong autocorrelation
pacf(ts.ugs.diff.diff4)
kpss.test(ts.ugs.diff.diff4) #showing that the data is stationary


#DGS Forecasting
#Question 1 - Try different
adf.test(ts.dgs) #Augmented Dickey-Fuller Test to test the
stationary.

#Taking the first difference.
ts.dgs.diff<-diff(ts.dgs)
plot(ts.dgs.diff)
ts.dgs.diff<-ts.dgs.diff[-1]
acf(ts.dgs.diff)
pacf(ts.dgs.diff)
adf.test(ts.dgs.diff) #checking stationarity

#The acf shows the seasonality at lag 4.
ts.dgs.diff.diff4 <- diff(ts.dgs.diff,4)
ts.dgs.diff.diff4 <- ts.dgs.diff.diff4[-1:-4]
plot(ts.dgs.diff.diff4)
acf(ts.dgs.diff.diff4)
pacf(ts.dgs.diff.diff4)
kpss.test(ts.dgs.diff.diff4) #checking stationarity
```

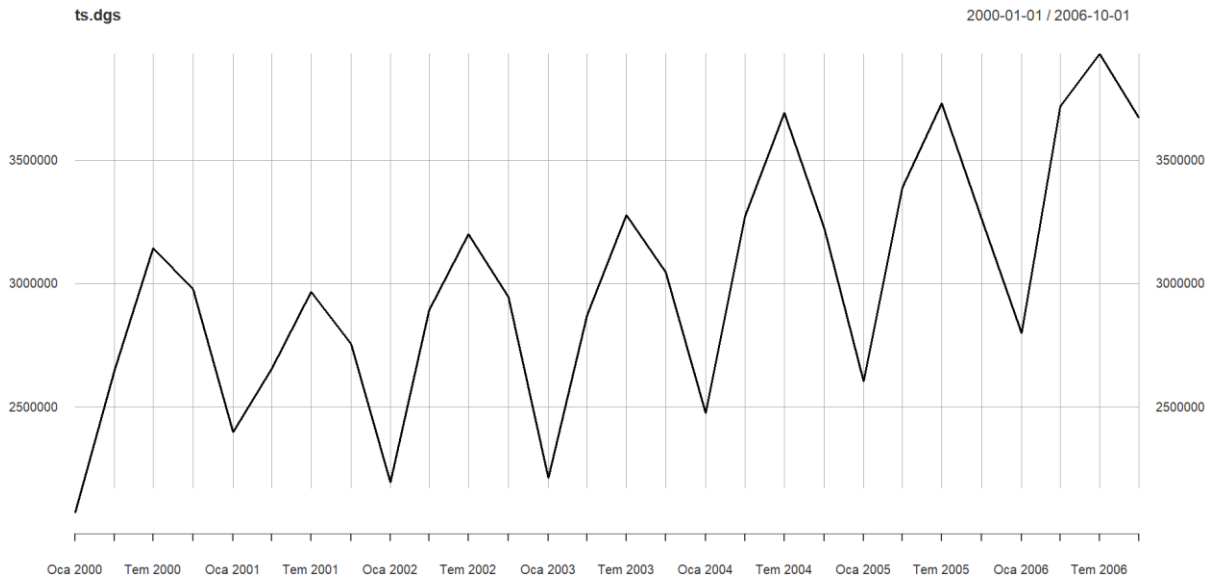
```
#Question 3
#ARIMA(0,1,0)(0,1,0)[4] model is chosen.
#It is chosen because one regular and one seasonal differencing is
done
#As we saw in the ACF and PACF functions, there were no need for MA
or AR
ugs.model <- Arima(ts.ugs, order=c(0, 1, 0), seasonal = list(order =
c(0, 1, 0), period = 4))
ugs.model
tsdisplay(ugs.model$residuals)


#Neighborhood Search for ugs
#1. ARIMA(1,1,0)(0,1,0)[4]
model.1 <- Arima(ts.ugs, order=c(1, 1, 0), seasonal = list(order =
c(0, 1, 0), period = 4))
tsdisplay(model.1$residuals)

#2. ARIMA(1,1,1)(0,1,0)[4]
model.2 <- Arima(ts.ugs, order=c(1, 1, 1), seasonal = list(order =
c(0, 1, 0), period = 4))
tsdisplay(model.2$residuals)

#3. ARIMA(0,1,1)(0,1,0)[4]
model.3 <- Arima(ts.ugs, order=c(0, 1, 1), seasonal = list(order =
c(0, 1, 0), period = 4))
tsdisplay(model.3$residuals)

#4. ARIMA(0,2,0)(0,1,0)[4]
model.4 <- Arima(ts.ugs, order=c(0, 2, 0), seasonal = list(order =
c(0, 1, 0), period = 4))
tsdisplay(model.4$residuals)

#5. ARIMA(0,1,0)(1,1,0)[4]
model.5 <- Arima(ts.ugs, order=c(0, 1, 0), seasonal = list(order =
c(1, 1, 0), period = 4))
tsdisplay(model.5$residuals)

#6. ARIMA(0,1,0)(0,1,1)[4]
model.6 <- Arima(ts.ugs, order=c(0, 1, 0), seasonal = list(order =
c(0, 1, 1), period = 4))
tsdisplay(model.6$residuals)

ugs.model
model.1 #this turns out to be the best as it has the lowest AIC,
AICc, and BIC
model.2
model.3
model.4
model.5
model.6
tsdisplay(model.1$residuals)
tsdisplay(model.2$residuals)
tsdisplay(model.3$residuals)
tsdisplay(model.4$residuals)
```

```
tsdisplay(model.5$residuals)
tsdisplay(model.6$residuals)

#Create a table to compare the different models.


#Neighborhood Search for dgs
dgs.model <- Arima(ts.dgs, order=c(0, 1, 0), seasonal = list(order =
c(0, 1, 0), period = 4))
dgs.model
tsdisplay(dgs.model$residuals)

#1. ARIMA(1,1,0)(0,1,0)[4]
model.dgs.1 <- Arima(ts.dgs, order=c(1, 1, 0), seasonal = list(order
= c(0, 1, 0), period = 4))
tsdisplay(model.dgs.1$residuals)

#2. ARIMA(1,1,1)(0,1,0)[4]
model.dgs.2 <- Arima(ts.dgs, order=c(1, 1, 1), seasonal = list(order
= c(0, 1, 0), period = 4))
tsdisplay(model.dgs.2$residuals)

#3. ARIMA(0,1,1)(0,1,0)[4]
model.dgs.3 <- Arima(ts.dgs, order=c(0, 1, 1), seasonal = list(order
= c(0, 1, 0), period = 4))
tsdisplay(model.dgs.3$residuals)

#4. ARIMA(0,2,0)(0,1,0)[4]
model.dgs.4 <- Arima(ts.dgs, order=c(0, 2, 0), seasonal = list(order
= c(0, 1, 0), period = 4))
tsdisplay(model.dgs.4$residuals)

#5. ARIMA(0,1,0)(1,1,0)[4]
model.dgs.5 <- Arima(ts.dgs, order=c(0, 1, 0), seasonal = list(order
= c(1, 1, 0), period = 4))
tsdisplay(model.dgs.5$residuals)

#6. ARIMA(0,1,0)(0,1,1)[4]
model.dgs.6 <- Arima(ts.dgs, order=c(0, 1, 0), seasonal = list(order
= c(0, 1, 1), period = 4))
tsdisplay(model.dgs.6$residuals)

dgs.model
model.dgs.1
model.dgs.2
model.dgs.3
model.dgs.4 #best model. it has by far the lowest AIC, AICc, and BIC
model.dgs.5
model.dgs.6
tsdisplay(model.dgs.1$residuals)
tsdisplay(model.dgs.2$residuals)
tsdisplay(model.dgs.3$residuals)
tsdisplay(model.dgs.4$residuals)
tsdisplay(model.dgs.5$residuals)
tsdisplay(model.dgs.6$residuals)
```

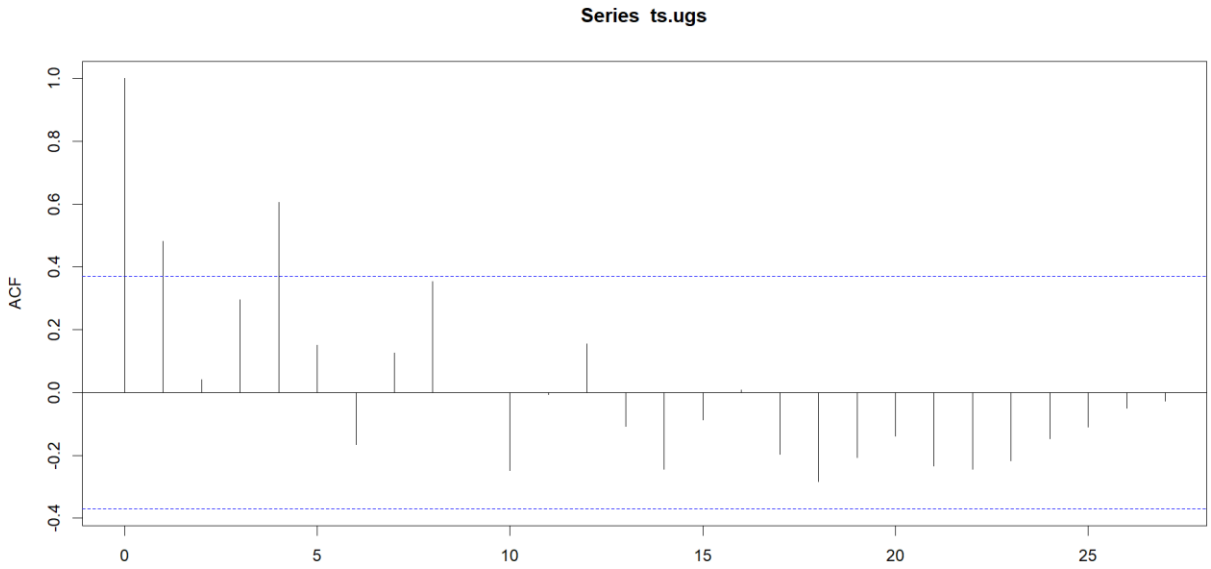
```
#Forecasts UGS
data.ts.forecast.ugs<-forecast(model.1, h=4) #ugs forecast for 2007
data.ts.forecast.ugs
plot(data.ts.forecast.ugs)

#Forecasts DGS
data.ts.forecast.dgs<-forecast(model.dgs.4, h=4) #dgs forecast for
2007
data.ts.forecast.dgs
plot(data.ts.forecast.dgs)
```



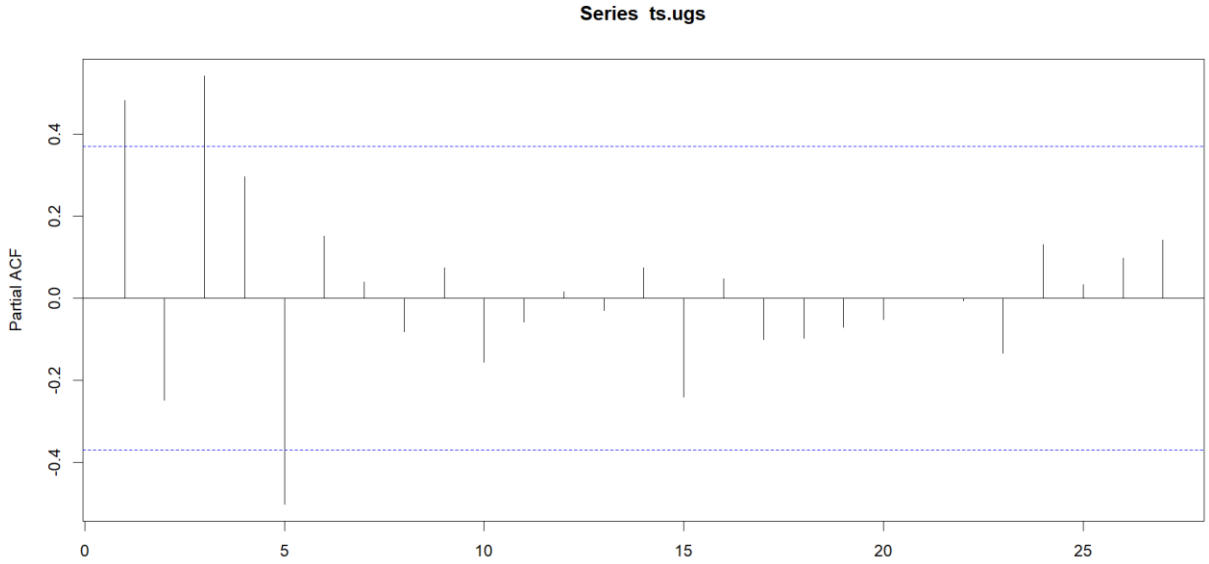
Appendix 1- Plot of UGS



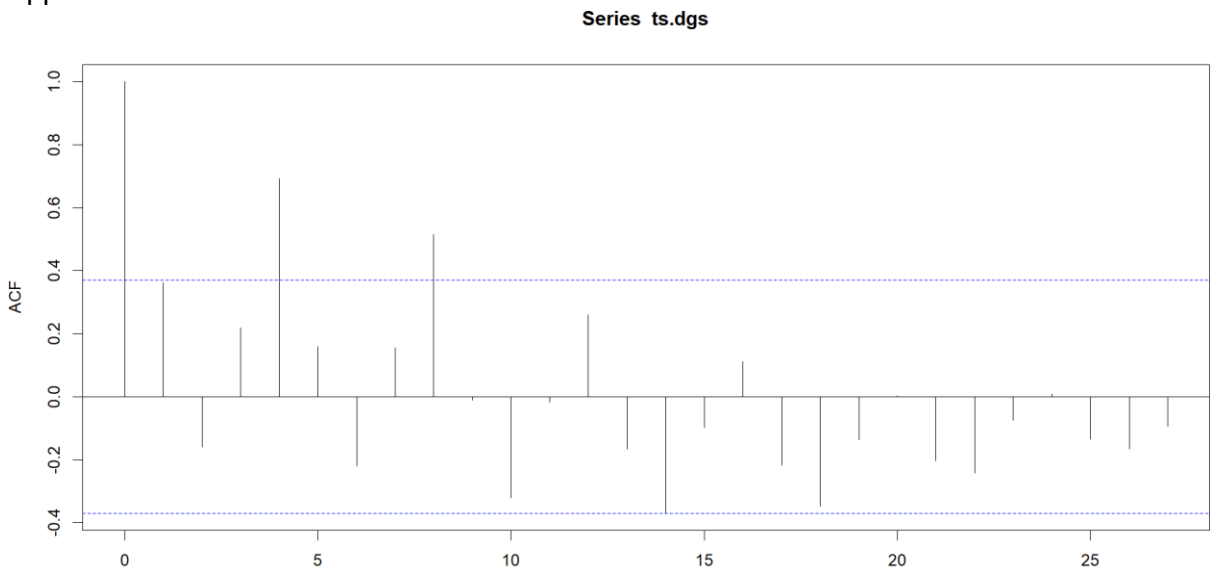
Appendix 2- Plot of DGS



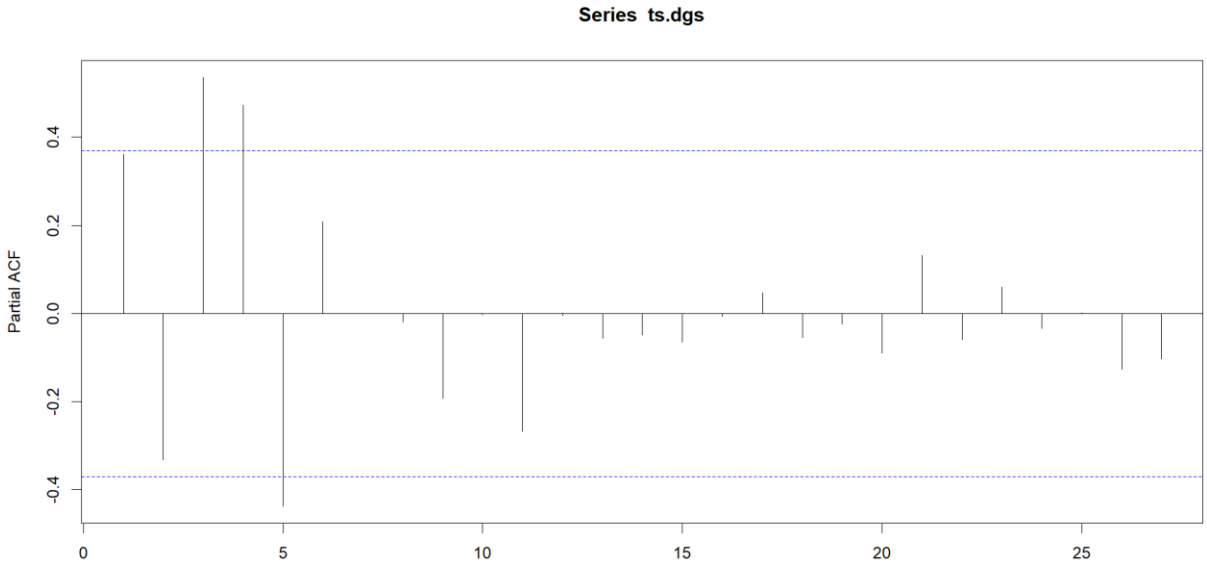
Appendix 3- Autocorrelation of UGS



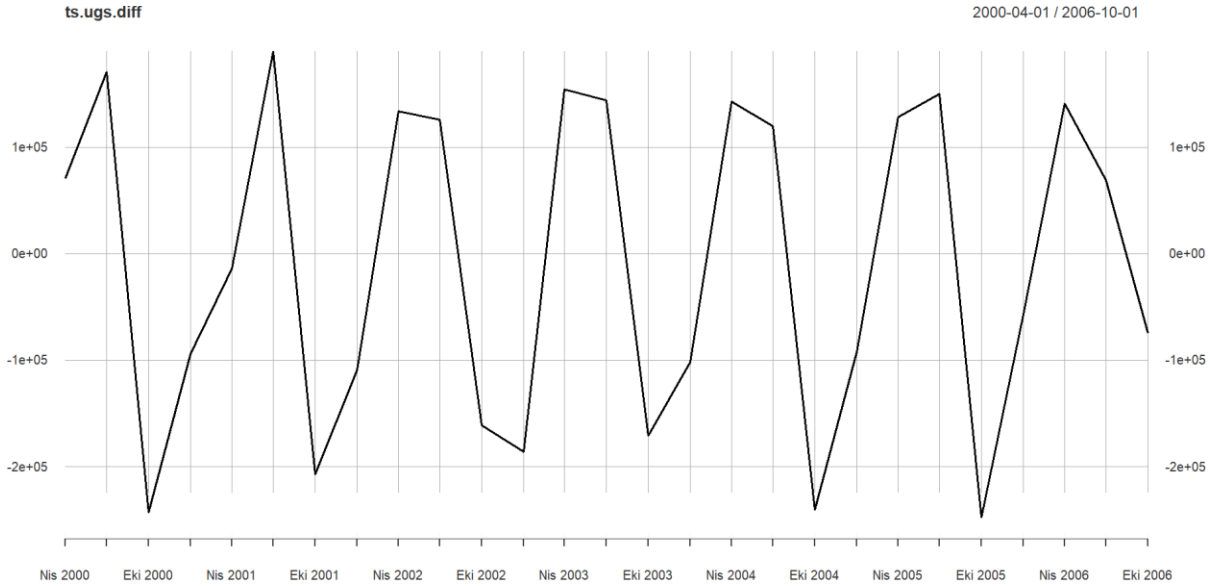
Appendix 4- Partial Autocorrelation of UGS



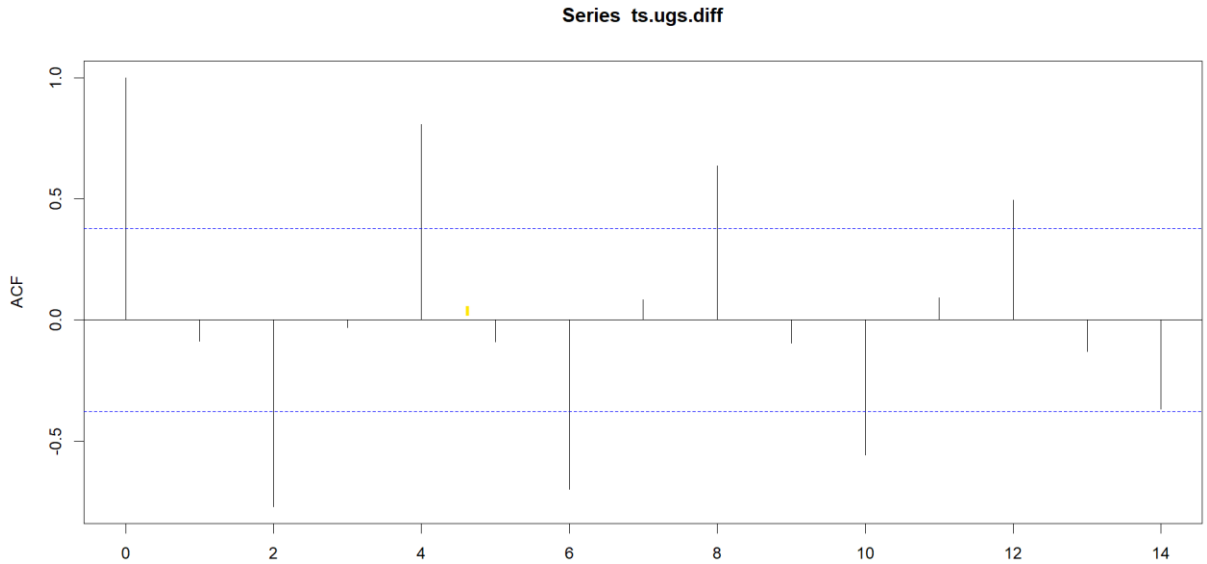
Appendix 5- Autocorrelation of DGS



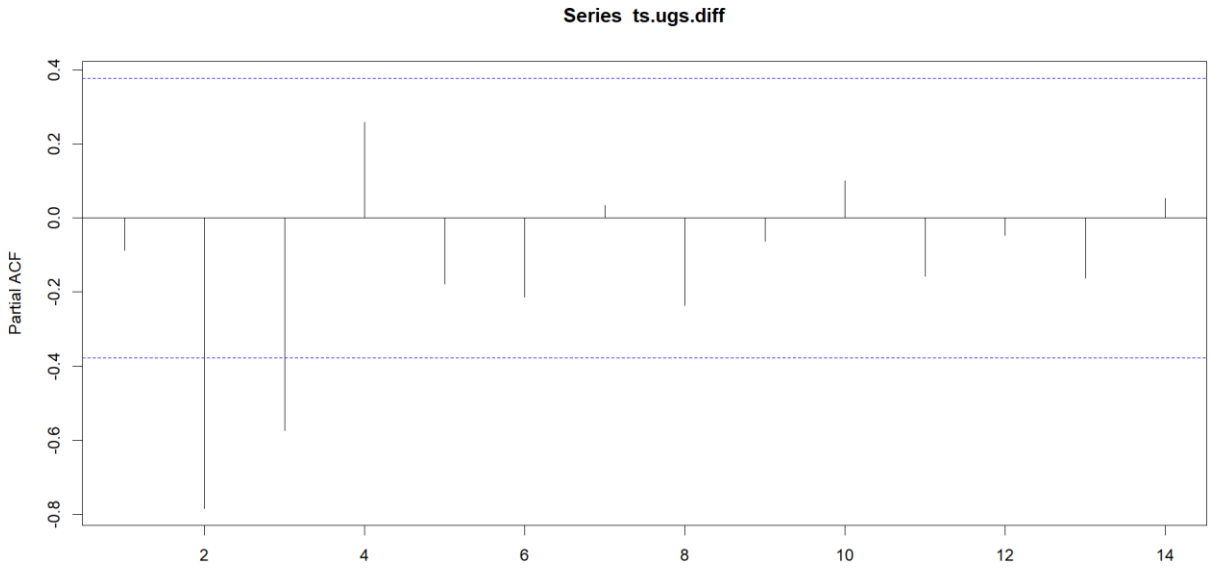
Appendix -6 Partial Autocorrelation of DGS



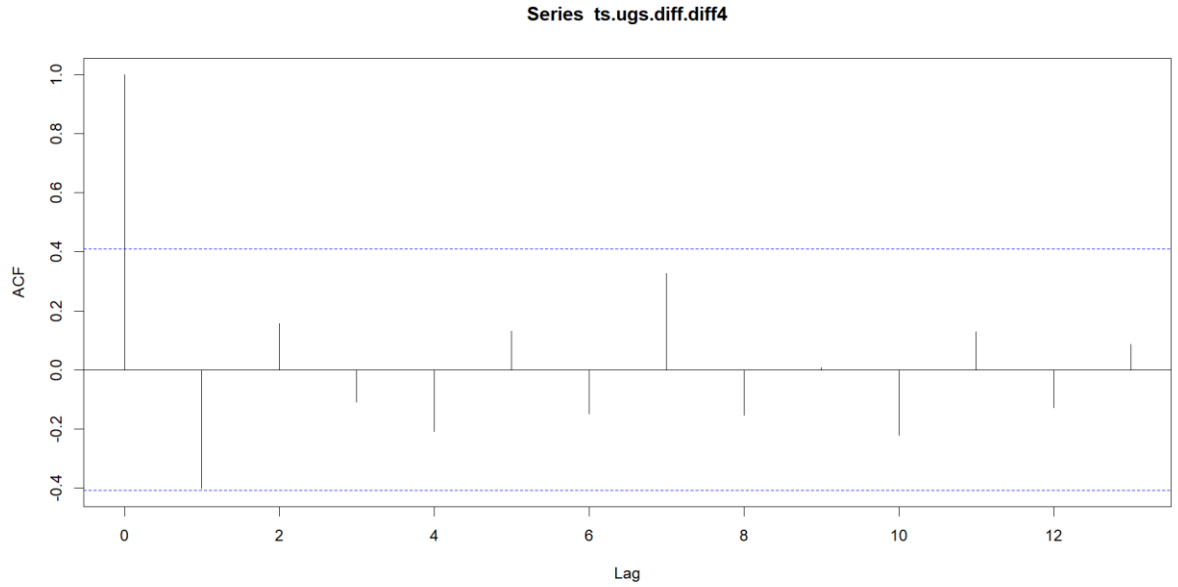
Appendix -7 Plot of 1st Difference UGS



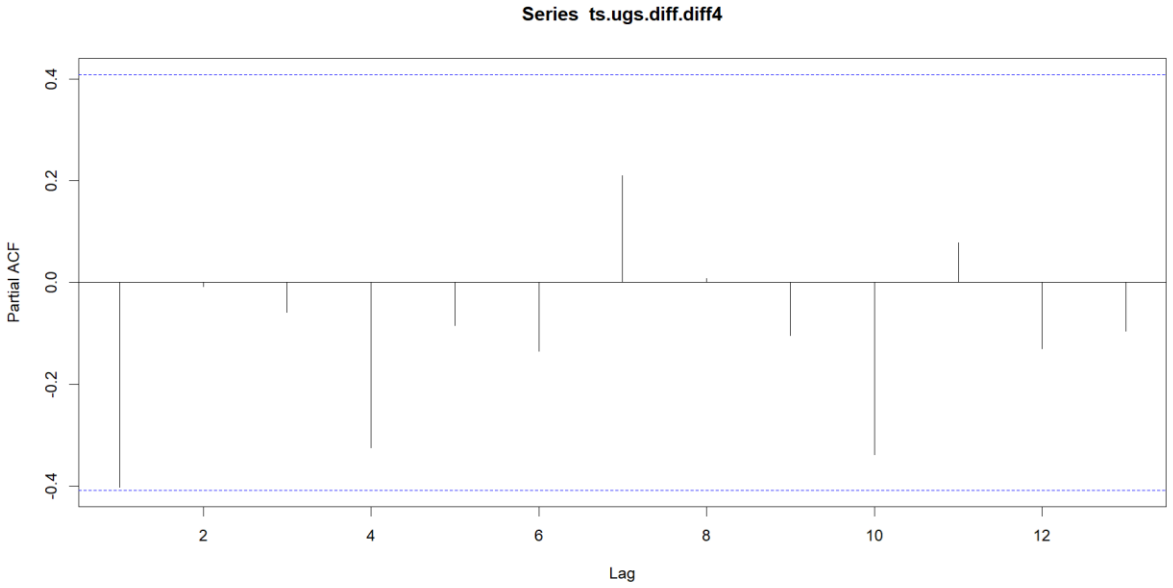
Appendix -8 Autocorrelation of 1st Difference UGS



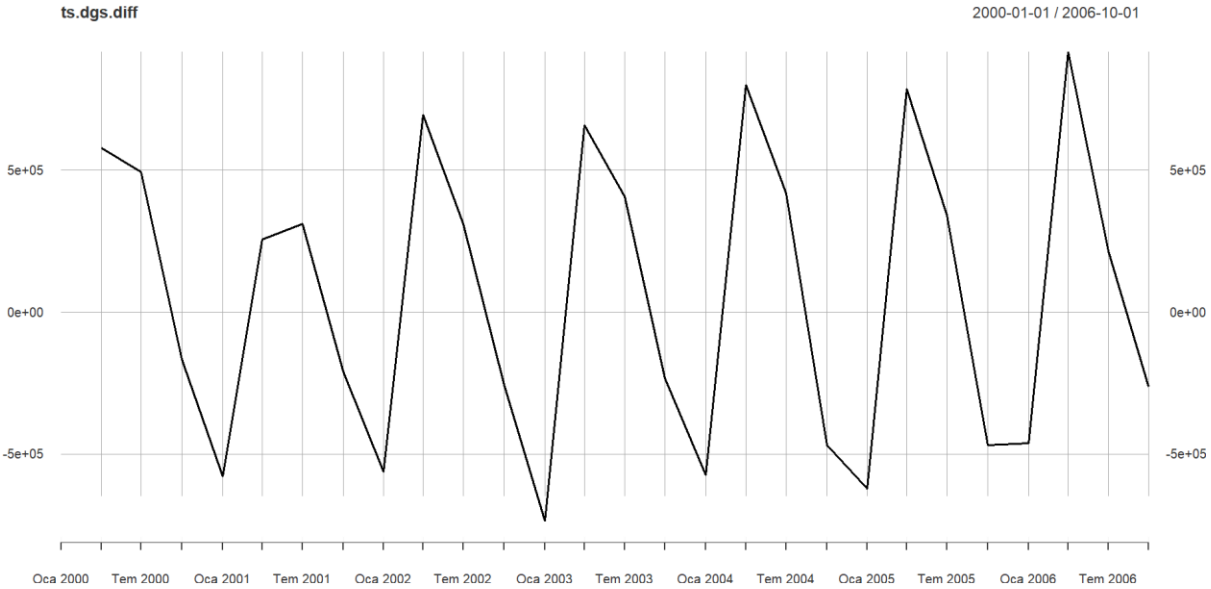
Appendix -9 Partial Autocorrelation of 1st Difference UGS



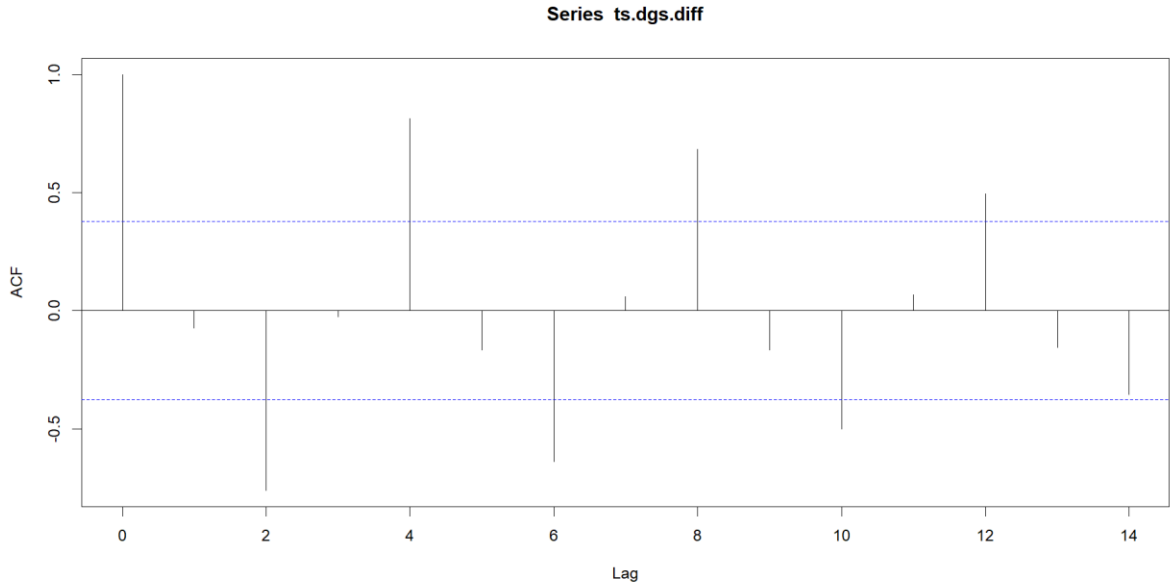
Appendix -10 Autocorrelation of 4th Difference of 1st Difference UGS



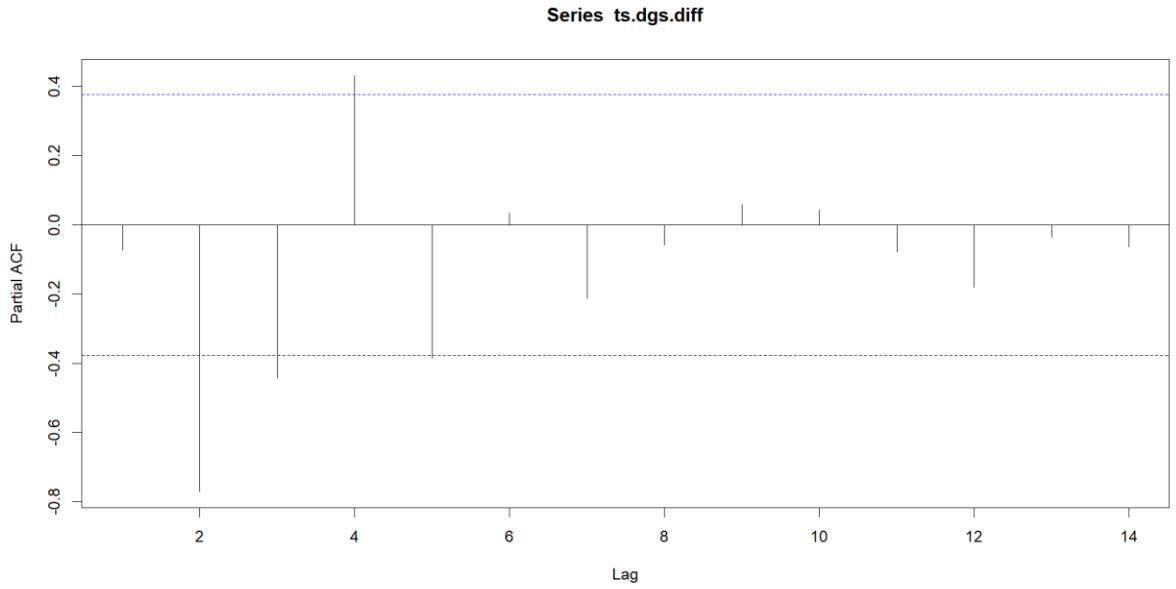
Appendix -11 Partial Autocorrelation of 4th Difference of 1st Difference UGS



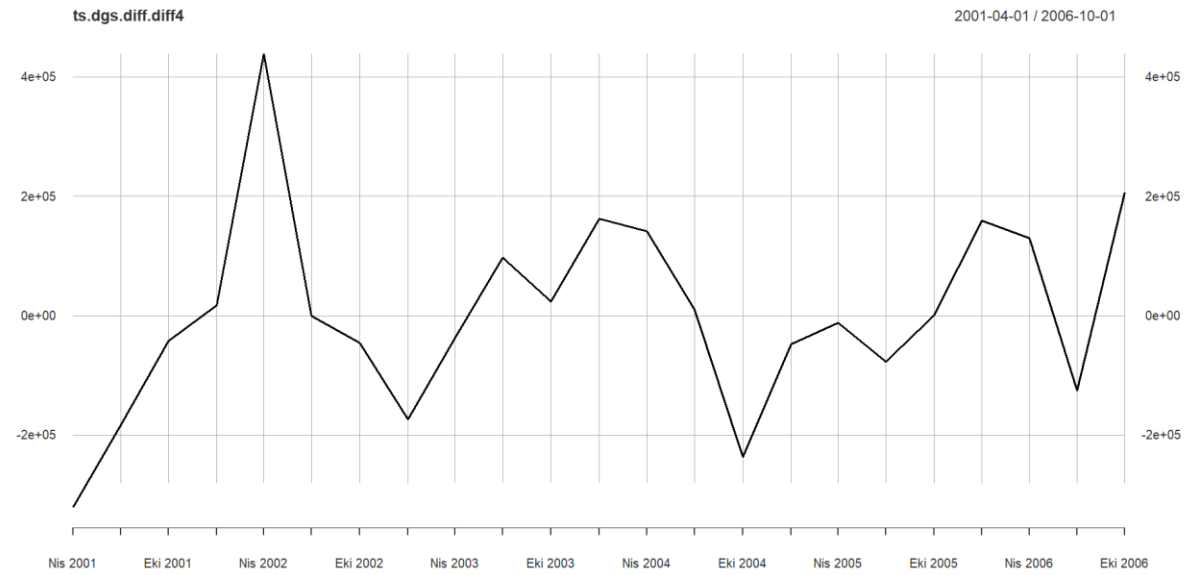
Appendix -12 Plot of 1st Difference DGS



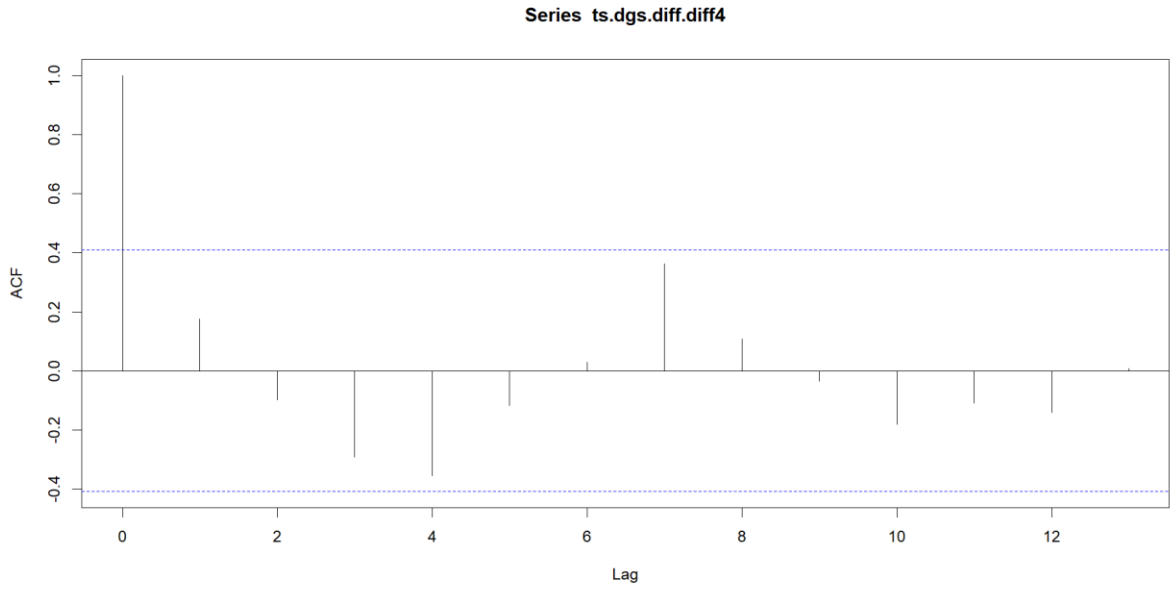
Appendix -13 Autocorrelation of 1st Difference DGS



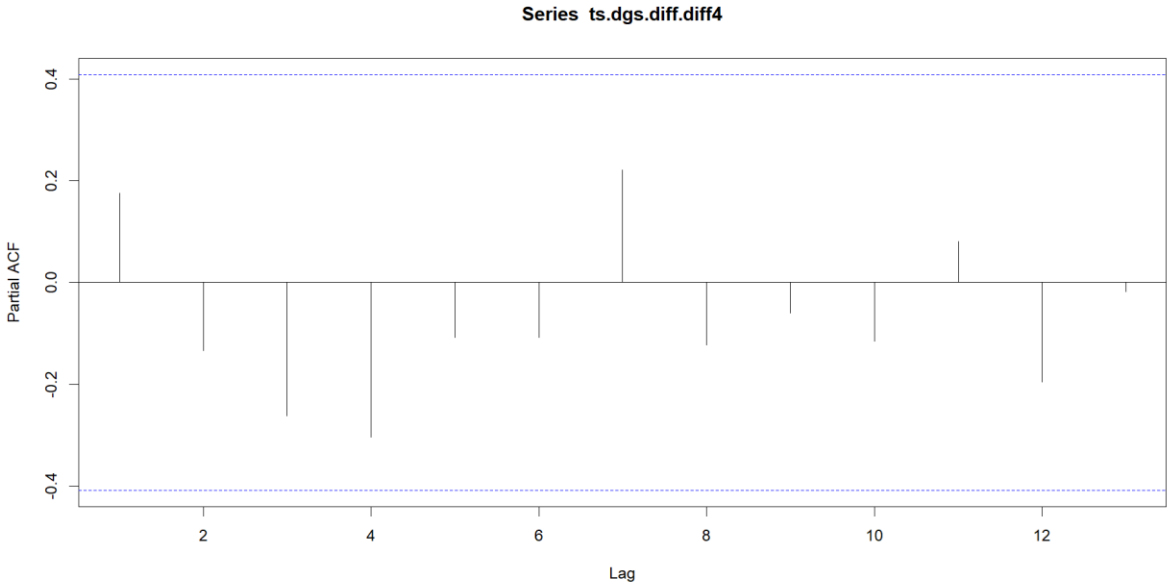
Appendix -14 Partial Autocorrelation of 1st Difference DGS



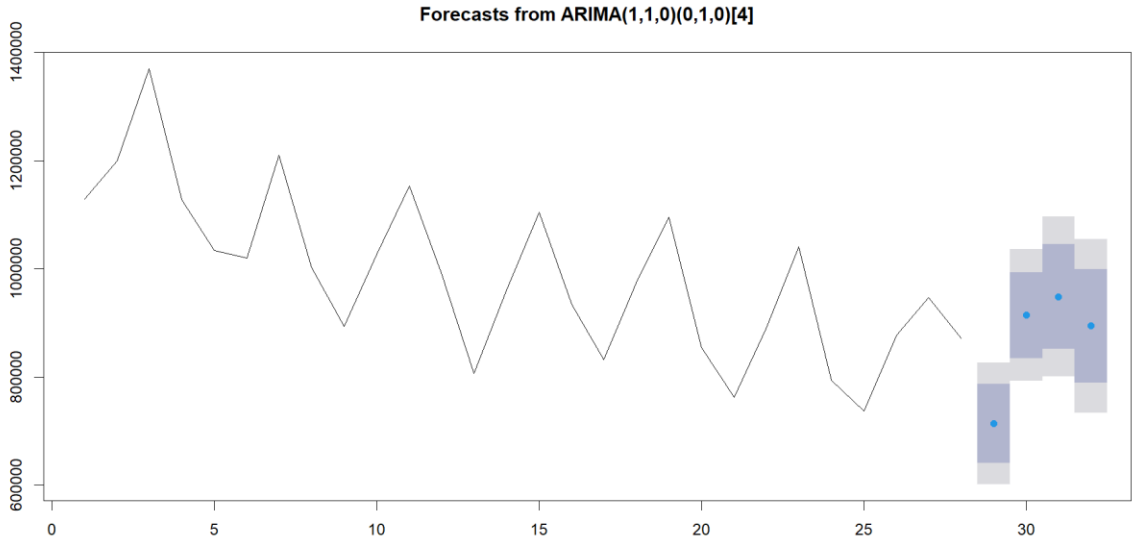
Appendix -15 Plot of 4th Difference of 1st Difference DGS



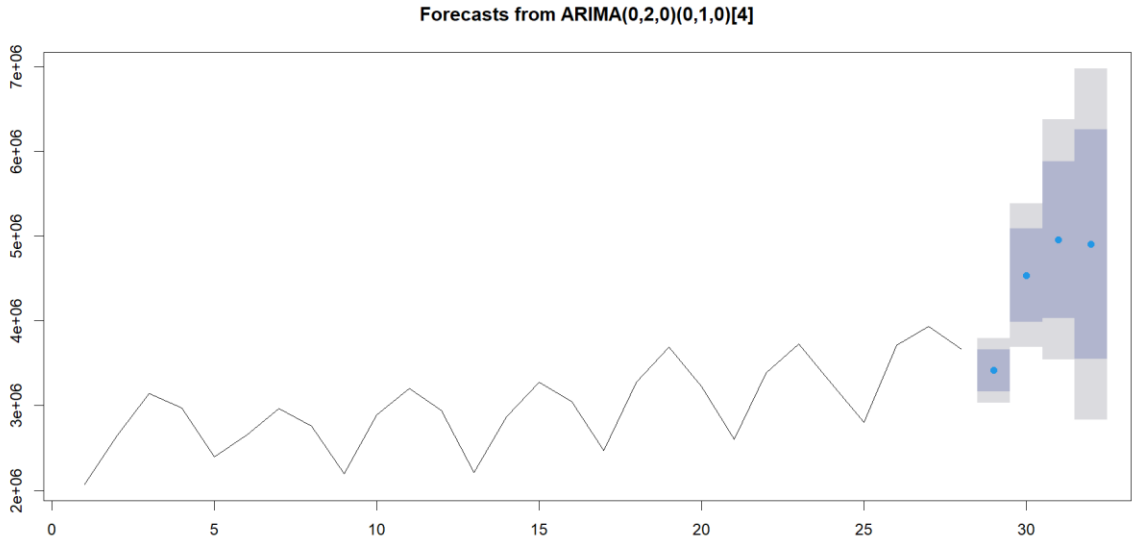
Appendix -16 Autocorrelation of 4th Difference of 1st Difference DGS



Appendix -17 Partial Autocorrelation of 4th Difference of 1st Difference DGS



Appendix -18 UGS Plot with forecasts



Appendix -19 DGS Plot with forecasts