Forecasting with Seasonal Adjustment using Classical Decomposition - An Application to UK Quarterly Gas Consumption

Course Name: Time Series Analysis and Forecasting

Course Code: PM-ASDS10

Submitted to:

Dr. Rumana Rois

Professor

Department of Statistics, JU

Submitted by:

Mohammad Saiduzzaman Sayed

ID: 20215063

Batch: 5th

Sec: A

Professional Masters in

Applied Statistics and Data Science (PM-ASDS)

JAHANGIRNAGAR UNIVERSITY

Assignment 2

Selected the R Dataset:

```
63%%6

## [1] 3

63%%3

## [1] 0
```

My ID is 20215063 and last two digit is 63. So, 63 %% 6 = 3 and 63 %% 3 = 0. Hence, our given dataset is UKgas and given title for assignment 2 is "Forecasting with Seasonal Adjustment using Classical Decomposition – An Application to UK Quarterly Gas Consumption"

Dataset Details:

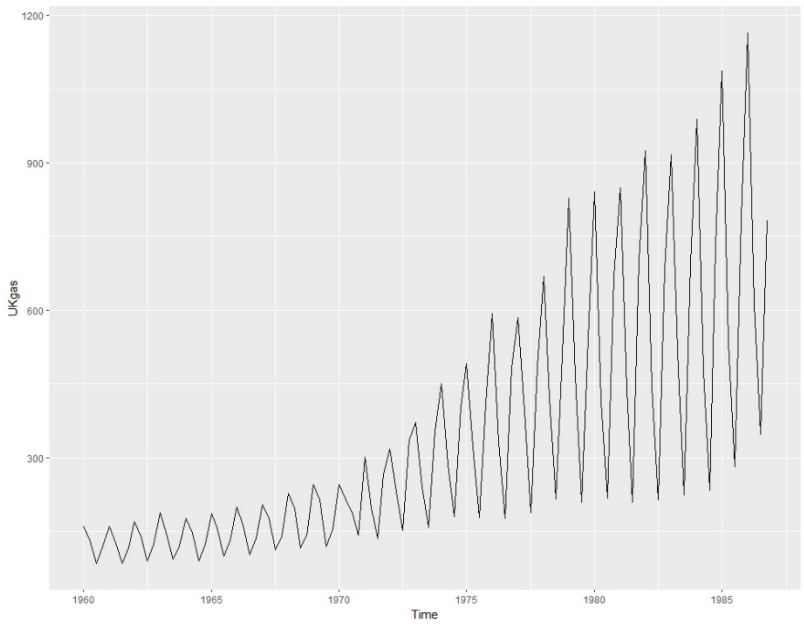
UK Quarterly Gas Consumption

Description: Quarterly UK gas consumption from 1960Q1 to 1986Q4, in millions of therms.
Usage: UKgae
Format: A quarterly time series of length 108.

1. Plots

Time-Plot:

```
plot(UKgas, col='brown', lwd=2, main='Time series plot of UKgas')
```



Comment on Time-plot:

The above time-plot describes some main features of data which is given below:

Trends: The above time series plot shows a clear upward trend.

Seasonal pattern: These data show a multiplicative seasonal pattern because the pattern repeats every 4 quarters. The frequency is 4, and therefore period is quarterly, so a seasonal component is present.

Sharp Changes: The above time series plot shows that there is no sharp change in our data.

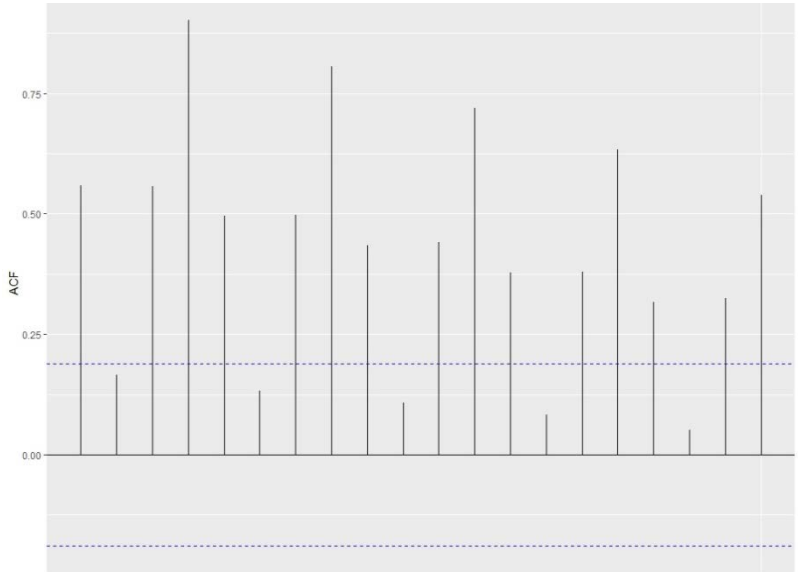
Outliers: Following the above time series plot, we can assume that there are some outliers in the data. We can check outliers with the help of forecast package and tsoutliers function in R.

```
tsoutliers(UKgas)
```

```
## $index
##  [1]  44  57  65  69  73  77  79  81  83  84  85  87  88  89  91  93  95
97  98
## [20]  99 101 103 105 107
##
## $replacements
##  [1] 301.1369 358.9973 404.5556 475.4278 492.0752 531.2751 394.3690 524.91
69
##  [9] 383.0432 565.7420 547.0516 402.9917 585.6832 559.4504 453.9700 643.24
21
## [17] 495.6343 670.9894 559.0268 534.9950 676.2419 551.6219 744.3583 588.41
11
```

ACF(AutoCorrelation Function) Plot:

```
acf(UKgas, col='brown', lwd=2)
```

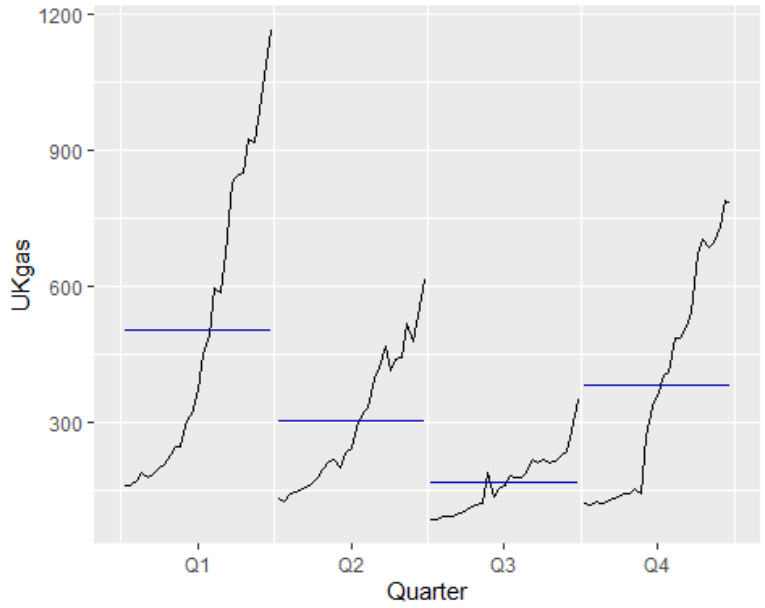


Comment on ACF plot:

ACF plot shows that first two significant spike and the coefficient is high at lag 4,8,12. So, here we can say trend and seasonality present in the dataset.

Seasonal-Subseries Plot:

```
ggsubseriesplot(UKgas)
```



Comment on seasonal subseries plot:

A seasonal subseries plot is used to determine if there is significant seasonality in a time series. For our quarterly data, all the Q1 values are plotted, then all the Q2 values, and so on. So, we can clearly see that from the above seasonal subseries data, there is seasonality present in UKgas data.

Estimated Outliers:

```
new_UKgas<-tsclean(UKgas)
```

Here, we estimated outliers with tsclean function in forecast package.

2. Decompose

To estimate the trend component and seasonal component of a seasonal time series that can be described using an multiplicative model, we can use the "decompose ()" function in R. This function estimates the trend, seasonal, and irregular components of a time series that can be described using an multiplicative model.

Classical Decomposition:

$$y_t = f(S_t, T_t, R_t)$$

Where, y_t = data at period t

S_t = seasonal component at period t

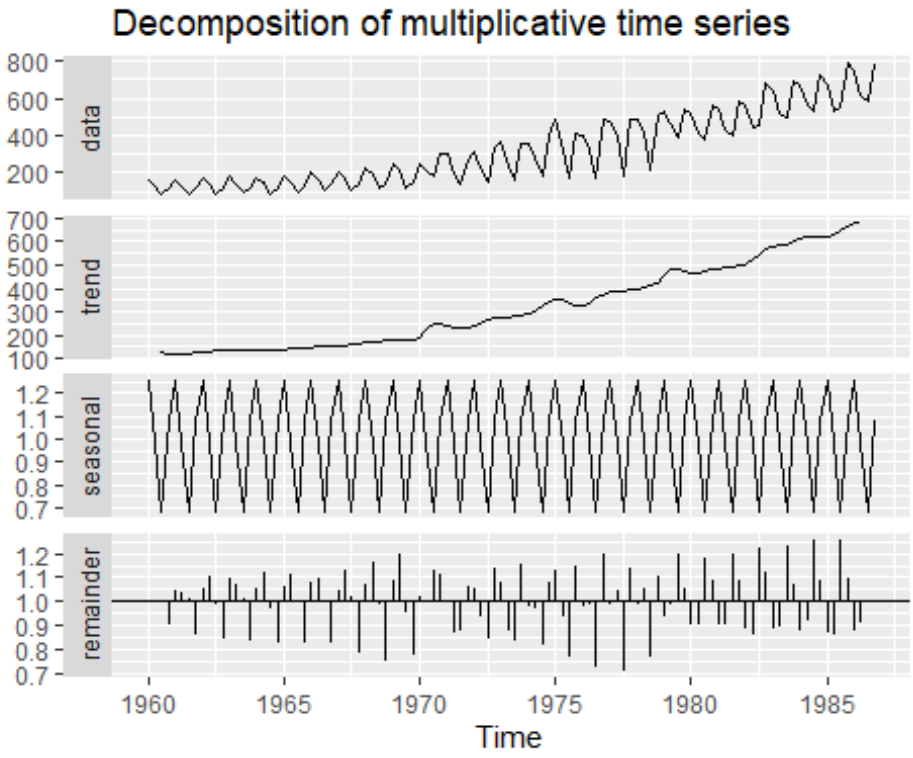
T_t = trend-cycle component at period t

R_t = remainder component at period t

The multiplicative decomposition says that time series data is a function of the product of its components. Thus,

Multiplicative decomposition: $y_t = S_t \times T_t \times R_t$

```
decompose<-decompose(new_UKgas, type = c("multiplicative"))
autoplot(decompose)
```



Comment: The plot shows the observed series, the trend line, the seasonal pattern and the random part of the series. Here, trend no calculated for first and last few values. And seasonal component repeats from year to year.

Seasonal Index using classical decompose:

```
seasonal<-seasonal(decompose)
```

Seasonal Adjusted Data:

We can calculate seasonally adjusted data from decomposition. For Multiplicative decomposition, the seasonally adjusted data are computed by dividing the original observation by the seasonal component.

$$\frac{y_t}{S_t} = T_t \times R_t$$

```
seasadj<-new_UKgas/decompose$seasonal
```

Splitting the Seasonal Adjusted Data into training (70%) and test (30%) data sets:

Training Data:

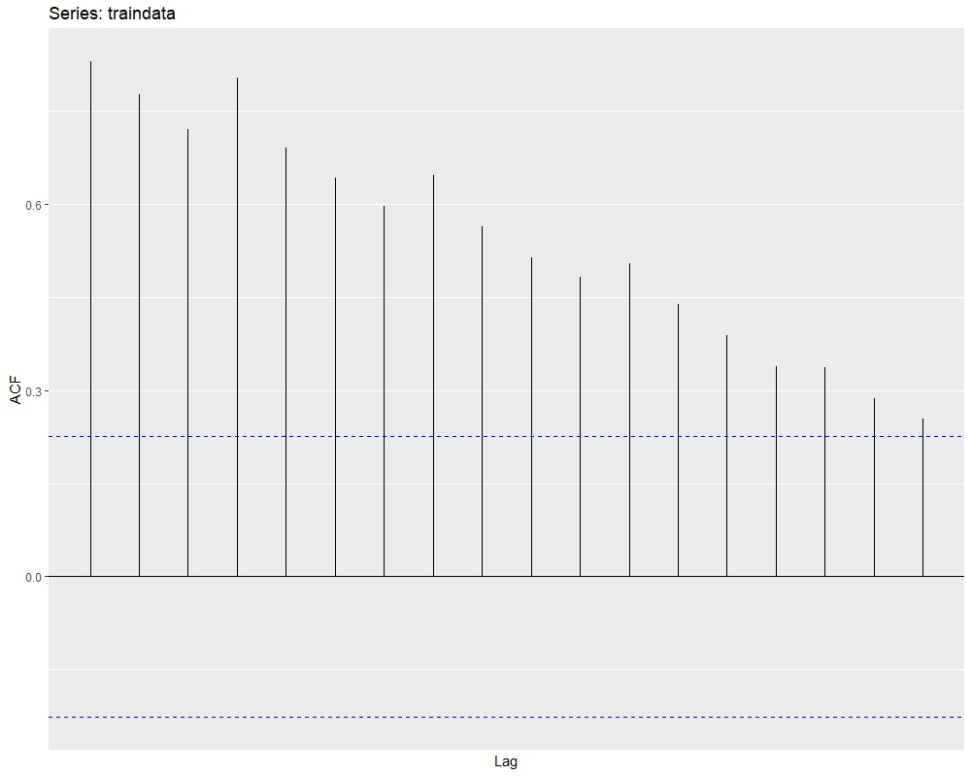
```
traindata<-ts(seasadj[0:75], frequency = 4, start=c(1960,1))
```

Test Data:

```
testdata<-ts(seasadj[76:length(seasadj)], frequency = 4, start=c(1978,4))
```

ACF (AutoCorrelation Function) Plot of Training Data:

```
acf(traindata)
```



Comment: The ACF plot shows that train data has trend and seasonality because of here the seasonal spike is significant.

Checking stationary of Training Data:

Augmented Dickey-Fuller (ADF) test and The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test can be used to test for stationary. These tests are available in tseries package with the function adf.test() for the ADF test and kpss.test() for the KPSS test.

ADF Test:

```
adf.test(traindata)

##
##  Augmented Dickey-Fuller Test
##
## data:  traindata
## Dickey-Fuller = -2.2541, Lag order = 4, p-value = 0.4718
## alternative hypothesis: stationary
```

In the ADF test,
H_0 : the series is not stationary
H_a: the series is stationary
Hence, a small p-value (i.e., less than Alpha=0.05) suggests that the series is stationary. Here, the p-value(0.5043) is greater than Alpha(0.05). So, we can not reject H0. That means the ADF test shows that the Seasonal Adjusted training data of UKgas is not stationary.

KPSS Test:

```
kpss.test(traindata)
##  KPSS Test for Level Stationarity
##
## data:  traindata
## KPSS Level = 1.8334, Truncation lag parameter = 3, p-value = 0.01
```

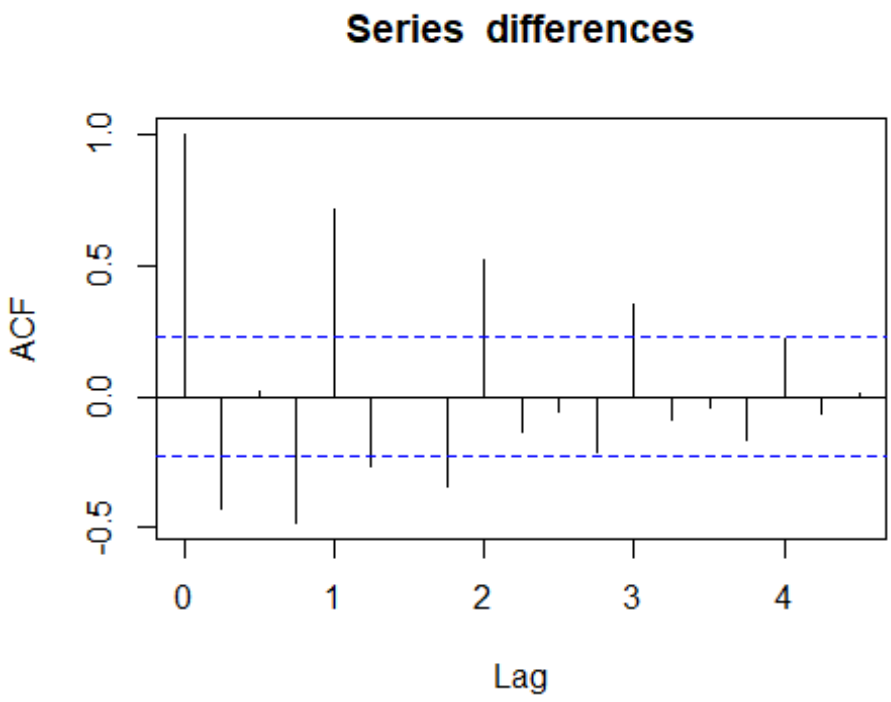
In the KPSS test,
H_0 : the series is stationary
H_a : the series is not stationary
Hence, a small p-value suggests that the series is not stationary, and a differencing is required. Here, the p-value(0.01) is less than Alpha(0.05). So, we can reject H0. That means the KPSS test shows that the Seasonal Adjusted training data of UKgas is not stationary.

So, to get stationary time series, the differences are required. Now, we can check stationary after taking differences.

Differences:

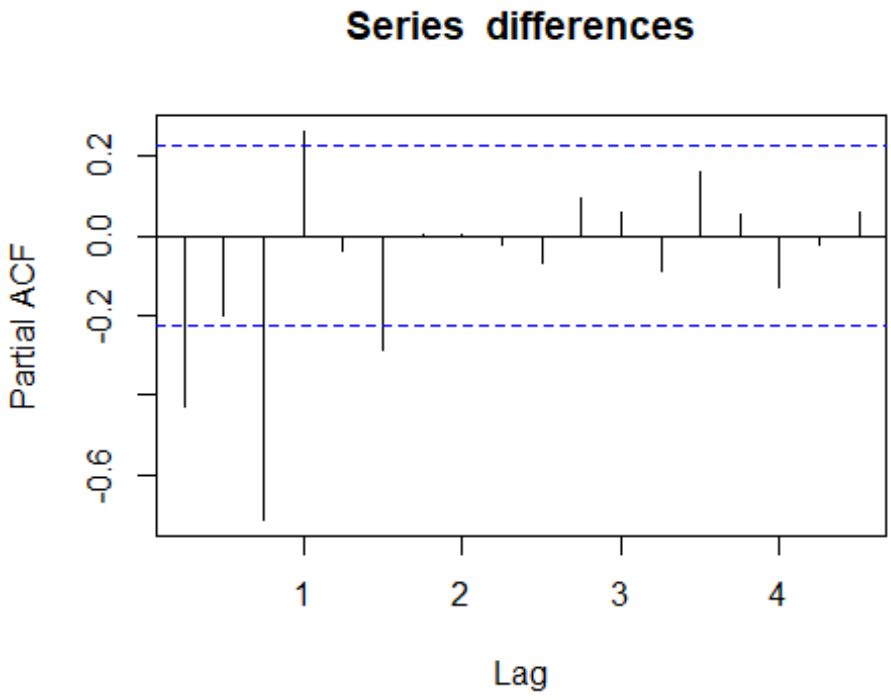
```
differences<-diff(traindata, differences = 1)
acf(differences)
```

ACF and PACF plot of differences:



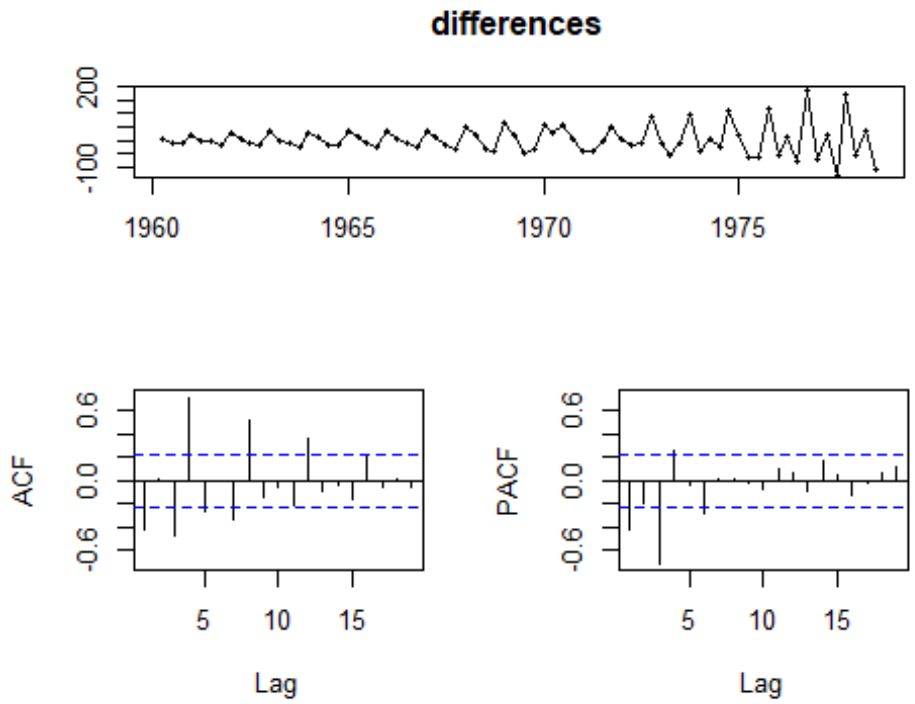
Comment: Here, the seasonal spike is significant, so we need seasonal differences. And MA(q)=2.

```
pacf(differences)
```



Comment: Here, the first two spike is significant, so AR(p)=2

```
tsdisplay(differences)
```



Checking Stationary of taking differences of Training Data:

```
adf.test(differences)

##
##  Augmented Dickey-Fuller Test
##
## data:  differences
## Dickey-Fuller = -5.8806, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

kpss.test(differences)

##
##  KPSS Test for Level Stationarity
##
## data:  differences
## KPSS Level = 0.093964, Truncation lag parameter = 3, p-value = 0.1
```

After taking difference the ADF test shows that the p-value is less than Alpha(0.05) means the data is stationary and the kpss test shows that the p-value is greater than Alpha(0.05) means the data is stationary. So, we can say that one non seasonal difference is enough for the M1 model seasonal adjusted training data of UKgas to get stationary series.

Here, the seasonal differences is re we can see ACF and PACF plot our spike is AR=2, MA=2. So the nonseasonal order is (p=2,d=1,q=2) and seasonal order is (P=0, D=1, Q=0).

Fit Model 1:

```
M1<-Arima(traindata, order = c(2, 1, 2), seasonal = c(0, 1, 0))
summary(M1)
```

```
## Series: traindata
## ARIMA(2,1,2)(0,1,0)[4]
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.2972  -0.6578  -1.8933  0.9316
## s.e.  0.0933   0.0872   0.0863  0.0894
##
## sigma^2 = 595.7:  log likelihood = -323.42
## AIC=656.83   AICc=657.77   BIC=668.07
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 2.739395 22.89534 13.87939 0.8436306 5.572667 0.6298915
##                      ACF1
## Training set -0.06009259
```

Mathematical Expression of M1 model:

ARIMA(2,1,2)(0,1,0)[4]

Estimated Model:

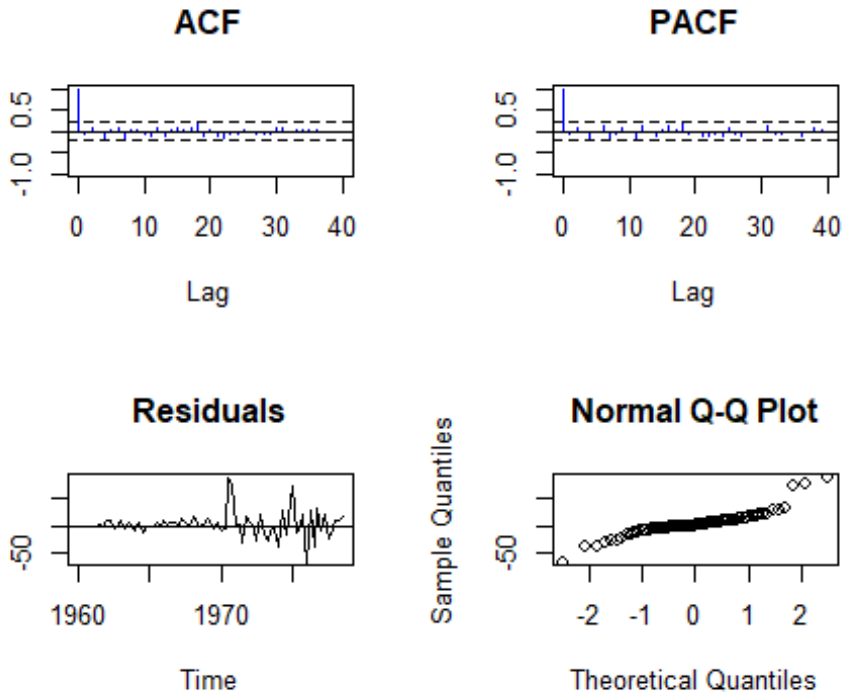
$$(1 - 1.2972B + 0.6578B^2)(1 - B)(1 - B^4)y_t = (1 - 1.8933B + 0.9316)Z_t,$$
$$where\ \{Z_t\} \sim WN(0, \hat{\sigma}^2 = 595.7)$$

Residuals Diagnostic Checking:

```
library(itsmr)

test(M1$residuals)
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     18.89    0.5291
## McLeod-Li Q                Q ~ chisq(20)     22.96    0.2906
## Turning points T   (T-48.7)/3.6 ~ N(0,1)        49    0.9264
## Diff signs S         (S-37)/2.5 ~ N(0,1)        35    0.4268
## Rank P         (P-1387.5)/109.3 ~ N(0,1)      1403    0.8872
```



Assumptions:

i. {et} uncorrelated. Here, we can see the ACF and PACF plot of residuals diagnostics, there is no significant spike. So, we can say that the {et} is uncorrelated.

ii. {et} have mean zero. From our Residuals plot, we can say that mean zero but variance in not much constant.

iii. {et} is normally distributed. Here, the Normal Q-Q plot shows that the residuals are approximately normally distributed because the data is near 45 degrees.

Here, H_0: Residuals are iid noise.

We can see the summary of the residuals test here all p-value is greater than Alpha (0.05). So, here we cannot reject H_0. So, the M1 model satisfied all assumptions of residuals.

Fit Model 2:

```
M2<-auto.arima(traindata)
summary(M2)

## Series: traindata
## ARIMA(1,0,0)(1,1,0)[4] with drift
##
## Coefficients:
```

```
##          ar1     sar1   drift
##       0.4306  -0.3707  3.9052
## s.e.  0.1089   0.1085  0.9763
##
## sigma^2 = 682.2:  log likelihood = -331.25
## AIC=670.5   AICc=671.1   BIC=679.55
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -0.06681973 24.87067 16.77615 -2.069215 7.340374 0.761356
##                     ACF1
## Training set 0.06260969
```

Mathematical Expression of M2 model:

ARIMA(1,0,0)(1,1,0)[4] with drift

Estimated Model:

$$(1-0.4306B)(1+0.3707B^4)(1-B^4)y_t = Z_t, \qquad where\ \{Z_t\}\sim WN(0,\hat{\sigma}^2 = 682.2)$$

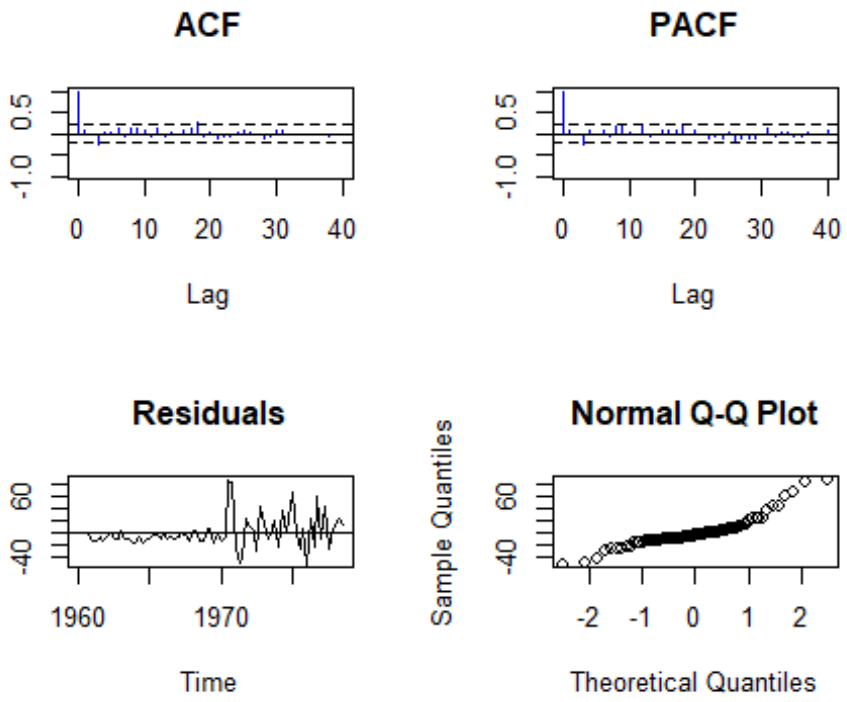
Mean Form:

$$(1-0.4306B)(1+0.3707B^4)(y_t' - 3.9052)(1-B^4)y_t = Z_t,$$
$$where\ \{Z_t\}\sim WN(0,\hat{\sigma}^2 = 682.2)$$

Residuals Diagnostic Checking:

```
test(M2$residuals)
```

```
## Null hypothesis: Residuals are iid noise.
## Test                        Distribution Statistic   p-value
## Ljung-Box Q                Q ~ chisq(20)     24.23    0.2327
## McLeod-Li Q                Q ~ chisq(20)     22.96    0.2909
## Turning points T   (T-48.7)/3.6 ~ N(0,1)        46    0.4597
## Diff signs S         (S-37)/2.5 ~ N(0,1)        37         1
## Rank P         (P-1387.5)/109.3 ~ N(0,1)      1623    0.0312 *
```



Assumptions:

i. {et} uncorrelated. Here, we can see the ACF and PACF plot of residuals diagnostics, there is no significant spike. So, we can say that the {et} is uncorrelated.

ii. {et} have mean zero. From our Residuals plot, we can say that mean zero but variance in not much constant.

iii. {et} is normally distributed. Here, the Normal Q-Q plot shows that the residuals are approximately normally distributed because the data is near 45 degrees.

Here, H_0: Residuals are iid noise.

We can see the summary of the residuals test here all p-value is greater than Alpha (0.05). So, here we cannot reject H_0. So, the M1 model satisfied all assumptions of residuals.

3. Model Selection Criteria:

Akaike Information Criterion:

$$AIC(\beta) = -2 \ln L_X(\beta, S_X(\beta)/n) + 2(p + q + 1)$$

Akaike Information Criterion corrected:

$$AICc(\beta) = -2 \ln L_X(\beta, S_X(\beta)/n) + \frac{2(p + q + 1)n}{(n - p - q - 2)}$$

Bayesian Information Criterion:

$$BIC = (n-p-q)\ln\left[\frac{n\hat{\sigma}^2}{(n-p-q)}\right] + n\left(1+ln\sqrt{2\pi}\right) + (p+q)ln\left[\frac{\sum_{t=1}^{n} X_t^2 - n\hat{\sigma}^2}{p+q}\right]$$

Model	**AIC**	**AICc**	**BIC**
M1	656.83	657.77	668.07
M2	670.5	671.1	679.55

For the same family as ARIMA whose AIC and AICc value among other models is the lowest is the best model. Here, M1 is the best fitted model because its AIC and AICc value is lowest. And, M1 model fulfilled more underlying assumptions of residuals than other.

4. Forecasting with seasonal adjustment

Forecasting with M1 Model:

```
n=length(testdata)
Fore_M1<-forecast::forecast(M1, h=n)$mean
ForeSEA_M1<-Fore_M1+seasonal[60:84]
```

Forecasting with M2 Model:

```
Fore_M2<-forecast::forecast(M2, h=n)$mean
ForeSEA_M2<-Fore_M2+seasonal[60:84]
```

Forecasting accuracy measures based on the test data set:

$$ME = \frac{1}{n}\sum_{t=1}^{n} e_t$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_t|$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n} e_t^2$$

$$MPE = \frac{1}{n}\sum_{t=1}^{n} PE_t$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|PE_t|$$

$$PE_t = \left(\frac{Y_t - F_t}{Y_t}\right) * 100$$

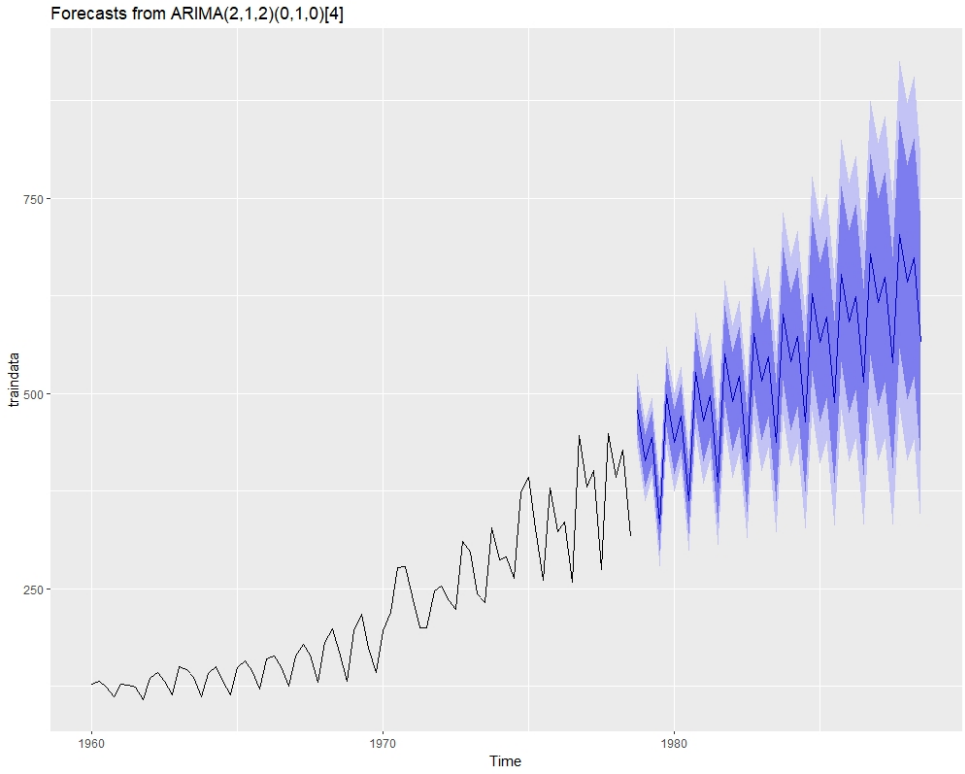
Model	MSE	RMSE	MAE	MPE	MAPE
M1	19718.88	140.4239	87.67655	8.319761	13.50448
M2	28275.07	168.1519	110.6208	14.64417	16.63395

After checking forecasting accuracy measures for the test data set, here the M1 model forecasting accuracy measures values are lower than other model. So, based on that M1 model is best fitted model for forecasting.

5. Forecasting

Using the best model M1, forecasting 10 points ahead:

```
autoplot(forecast::forecast(M1, h= 10*4))
```



THE END