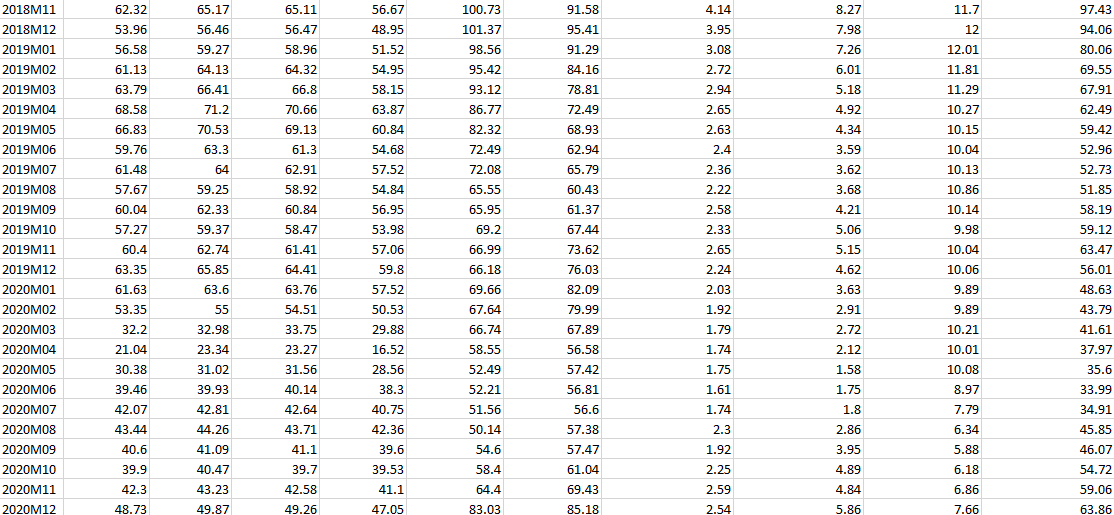
|  |
| --- |
| **TIME SERIES ANALYSIS** Whenever recordings of processes vary over time, Time Series come into the picture. A recording can either be a continuous trace or a set of discrete observations. In this analysis, we will limit our focus to observations which are discrete and equally-spaced (monthly in our case). There are a number of things which are of interest in time series analysis. The most important of these are:  **Modelling**: We observe the pattern in data and fit a time series model to data. This model depends on unknown parameters which need to be estimated.  **Forecasting**: On the basis of available observations we predict the further observations with the help of different time series forecasting techniques which suit our data.  **Stationarity and Non-Stationarity**  A key idea in time series is that of stationarity. Generally speaking, a time series is stationary if its behavior does not change over time. This means that the values always tend to vary about the same level and that their variability is constant over time. Stationary series have a rich theory and their behavior is well understood. This warrants the fundamental role they play in the study of time series. Not all time series that we encounter are stationary. In fact, nonstationary series tend to be the rule rather than the exception.  **Model Development**  A time series is simply a series of data points ordered with regard to time. Time is often the independent variable and the goal is usually to make a forecast for the future. However, there are other aspects that come into play when dealing with time series. Is it stationary? Is there some seasonality in it? Is the target variable auto correlated? How to test if a process is stationary. Employing a Dickey-Fuller test in R is the way out. It test the null hypothesis that a unit root is present. If it is, then p > 0, and the process is not stationary. Otherwise, p = 0, the null hypothesis is rejected, and the process is considered to be stationary.  **Analysis by ARIMA Modelling**  ARIMA stands for Autoregressive Integrated Moving Average. This model is fitted to the time series data either to better understand the data or to predict future points in the series. The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The parameters in ARIMA are p= order of autoregressive model q= order of the moving average model d= degree of differencing  **Differencing**: In statistics, differencing is the transformation applied to the time series data in order to make it stationary. In order to difference the data, the difference between consecutive observations is computed. Differencing removes the changes in the level of time series, eliminating trend and seasonality and consequently stabilizing the mean of the time series.  **Analysis**  Analysis has been done on the data from January 1960 to December 2020. We’ve also used a train-test split where the train set is till 2016 and test set is 2017 to 2018. We’ve very consciously avoided keep the test set as 2019-2020 as thanks to the COVID-19 pandemic, the oil prices went haywire and couldn’t be accurately predicted. However, we have dedicated an entire section to how analyse how COVID-19 affected the Oil Prices as a part of this project.  **Our data at a glance**    ***Head of the Data*** |
|  |

****

***Tail of the Data***

**Our data has 10 Columns, each a different type or index of Oil and each having different start years owing to their respective years of inception like for example: There are no Brent Oil prices in the Head of the Data because Brent Oil came into existence in 1979 January.**

**R CODE:**

***#importing libraries***

> library(forecast)

> library('ggplot2')

> library('tseries')

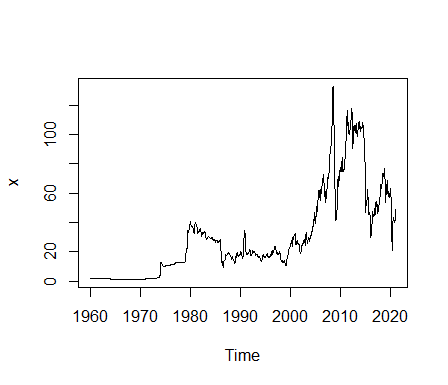
> library(fpp2)

> library(data.table)

> library(stats)

> library(readxl)

***# importing and reading our excel file in the R environment***

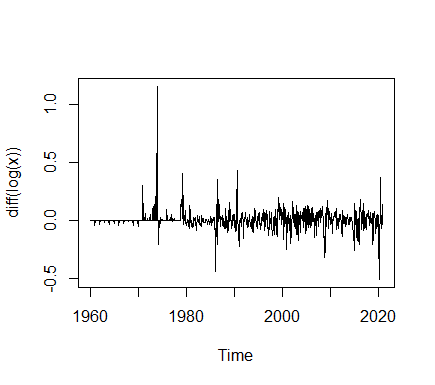
> final\_stat\_proj\_xls <- read\_excel("C:/Users/Paritosh Raikar/Desktop/final stat proj.xls.xlsx")

**CRUDE AVG:**

> x= final\_stat\_proj\_xls$`Crude Avg`

> x=ts(x, start= c(1960,1), end= c(2020,12), frequency = 12)

> plot(x)



***#Log transformation and one-time differencing***

***make variance constant and lose the trend.***

>plot(diff(log(x))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

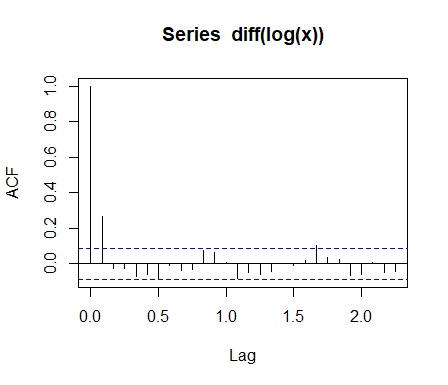
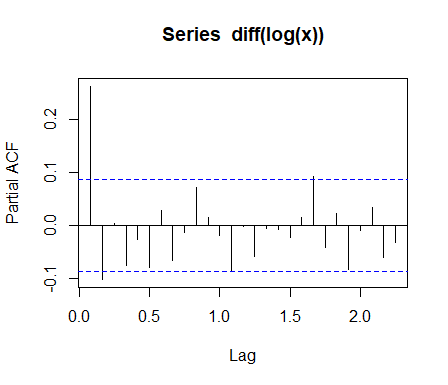
data: diff(log(x))

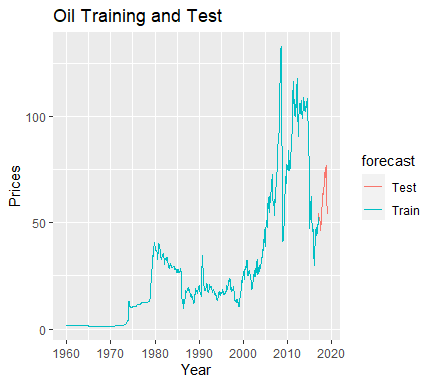
Dickey-Fuller = -8.3585, Lag order = 9, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf (diff(log(x))) > pacf(diff(log(x)))

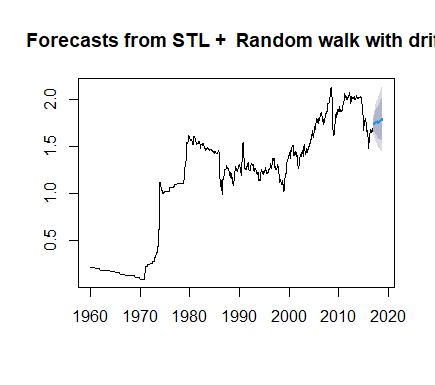


**Train-Test Split:**

> TS\_train <- window(x, start=c(1960,1), end=c(2016,12), freq=12)

> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

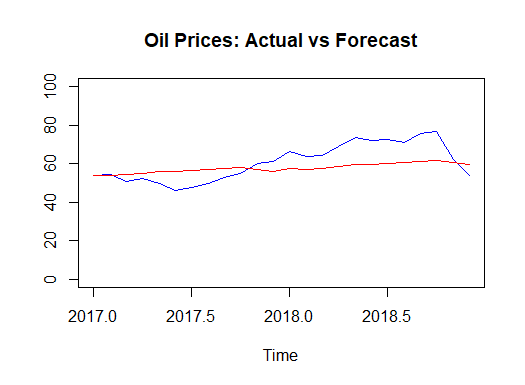
**apply the STL function and forecast it using the ‘random walk**

**method’.**

> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function.**

> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 8.3971

> MAPE

[1] 0.1149

***As the MAPE value comes out to 0.1149, it means that there’s an 11.49 % error in our forecast and our forecast is 88.51% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(0,0,1)

Coefficients:

ma1

0.2348

s.e. 0.0372

sigma^2 estimated as 0.007769: log likelihood=738.67

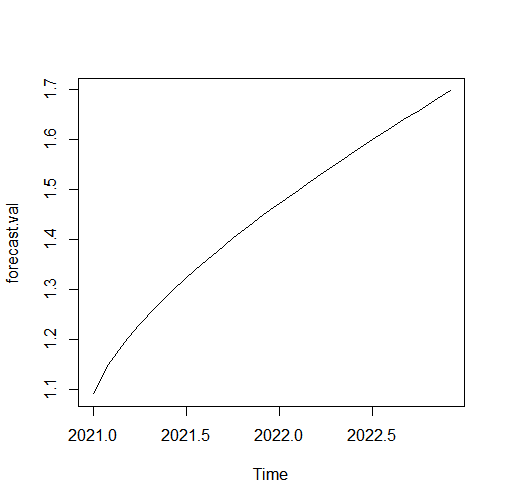
AIC=-1473.35 AICc=-1473.33 BIC=-1464.16

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)



**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(0,0,1))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices increase rapidly at the start of the graph i.e. the January and Ferbruary months of 2021 and becomes completely linear over the course of the next 22 months. So we infer that Crude Oil will grow steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

> Box.test(fit.res,type="Ljung")

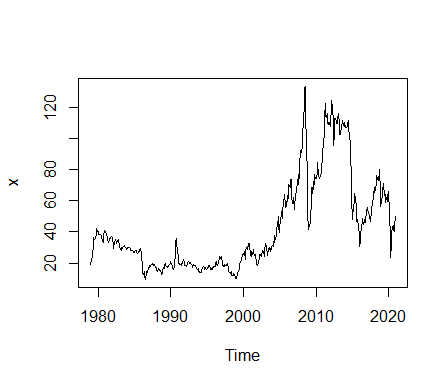
Box-Ljung testf

data: fit.res

X-squared = 0.028798, df = 1, p-value = 0.8652

***Since p value is greater than 0.05, the residuals are independent.***

**BRENT:**

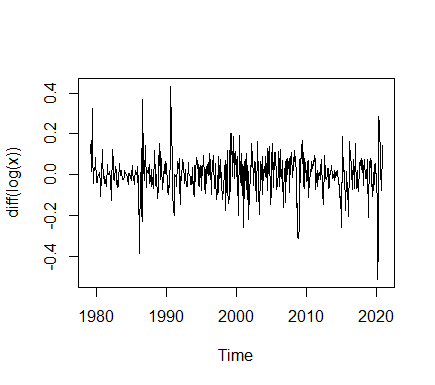
> x= final\_stat\_proj\_xls$Brent

> x=x[229:734]

> x=as.numeric(x)

> x=ts(x, start= c(1979,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

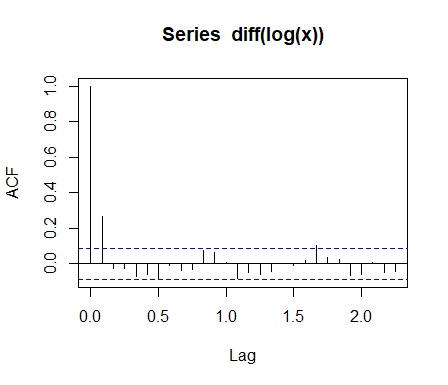
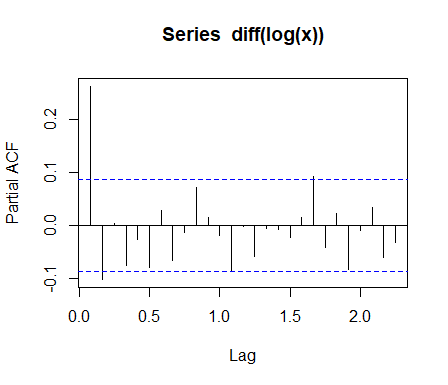
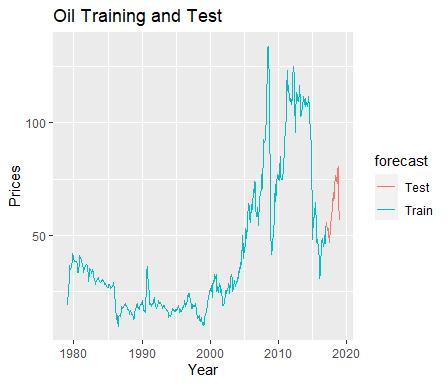
data: diff(log(x))

Dickey-Fuller = -9.0255, Lag order = 7, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))



**Train-Test Split:**

> TS\_train <- window(x, start=c(1979,1), end=c(2016,12), freq=12)

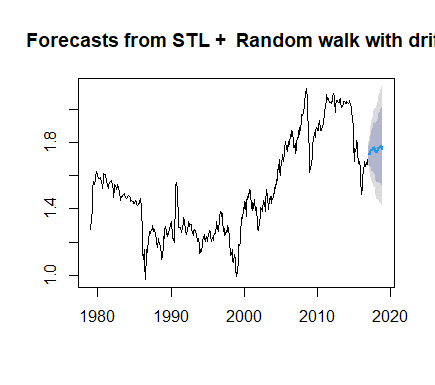
> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

#Our series is Multiplicative by nature. We take the log of our

train set, effectively converting it Into additive series. Then we

apply the STL function and forecast it using the ‘random walk

method’.

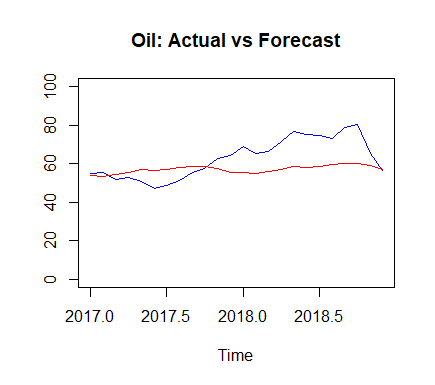


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE<- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 10.9016

> MAPE

[1] 0.1353

***As the MAPE value comes out to 0.1353, it means that there’s an 13.53 % error in our forecast and our forecast is 86.47% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(1,0,2)

Coefficients:

ar1 ma1 ma2

0.8942 -0.6113 -0.3101

s.e. 0.0621 0.0711 0.0430

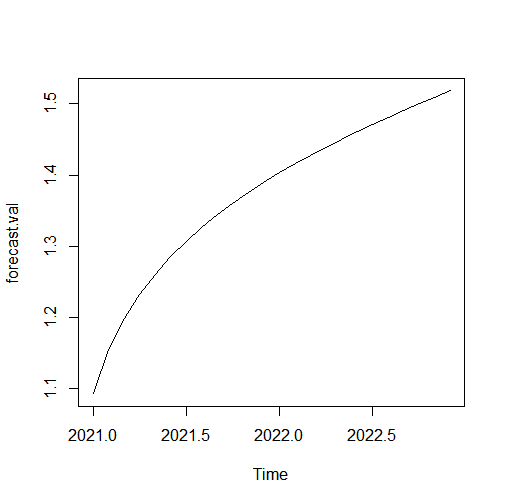
sigma^2 estimated as 0.007924: log likelihood=504.42

AIC=-1000.84 AICc=-1000.76 BIC=-983.95

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=1 and q=2***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(1,0,2))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices increase rapidly in the middle months April, May of 2021 and becomes completely linear over the time. So we infer that Brent Oil prices will grow steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

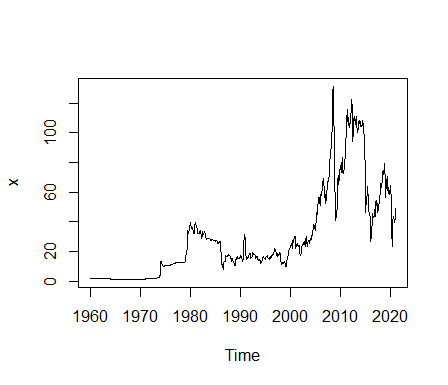
> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.0011425, df = 1, p-value = 0.973

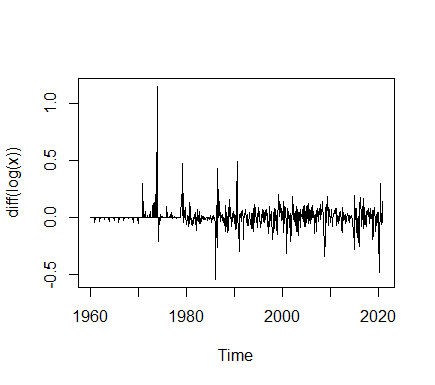
***Since p value is greater than 0.05, the residuals are independent.***

**DUBAI:**

> x= final\_stat\_proj\_xls$Dubai

> x=ts(x, start= c(1960,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>Plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

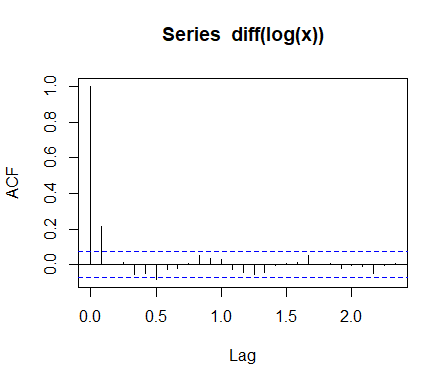
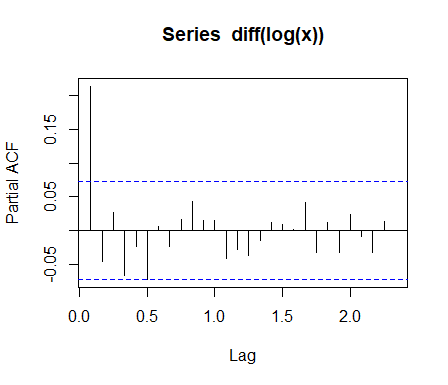
data: diff(log(x))

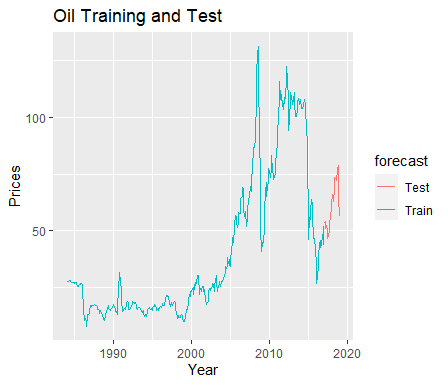
Dickey-Fuller = -8.4913, Lag order = 9, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

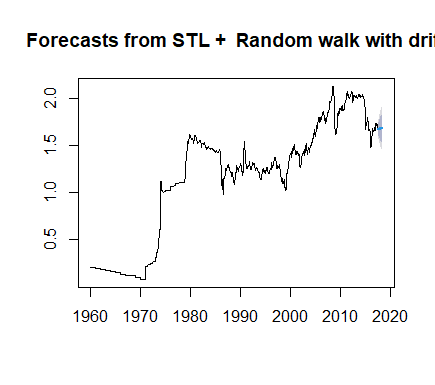
> acf(diff(log(x))) > pacf(diff(log(x)))



**Train-Test Split:**

> TS\_train <- window(x, start=c(1984,1), end=c(2016,12), freq=12)

> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)



**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

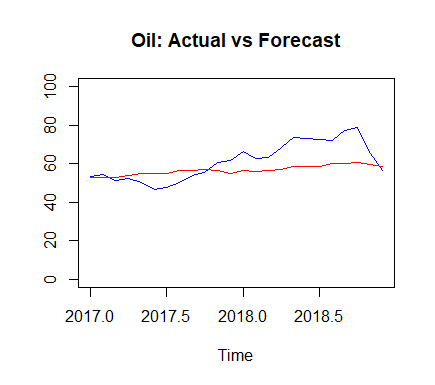
**method’.**

> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 9.2031

> MAPE

[1] 0.1391

***As the MAPE value comes out to 0.1391, it means that there’s an 13.91 % error in our forecast and our forecast is 86.09% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(0,0,1)

Coefficients:

ma1

0.2305

s.e. 0.0369

sigma^2 estimated as 0.008231: log likelihood=717.58

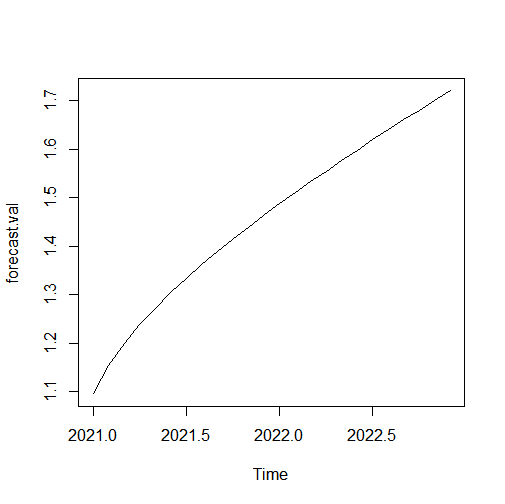
AIC=-1431.16 AICc=-1431.14 BIC=-1421.97

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)



**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(0,0,1))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices increase fast at the start of the graph i.e. the January and Ferbruary months of 2021 and becomes completely linear over the course of the next 22 months. So we infer that Dubai Oil Prices will rise steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

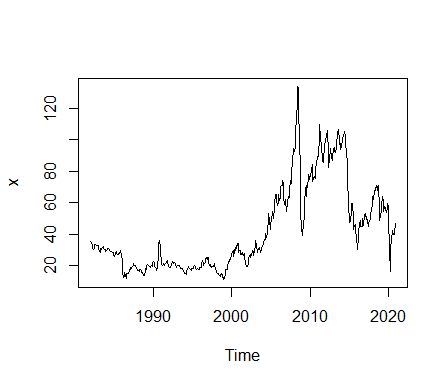
X-squared = 0.011938, df = 1, p-value = 0.913

***Since p value is greater than 0.05, the residuals are independent.***

**WTI:**

> x= final\_stat\_proj\_xls$WTI

> x=x[265:734]

> x=as.numeric(x)

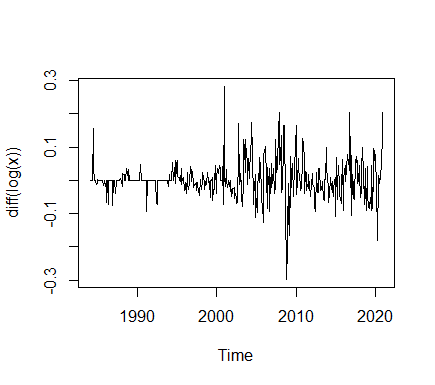
> x=ts(x, start= c(1982,1), end= c(2020,12), frequency = 12)

> plot(x)

**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

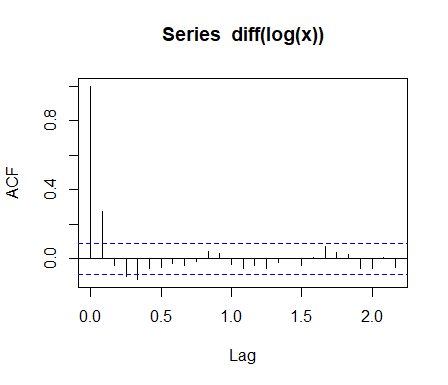
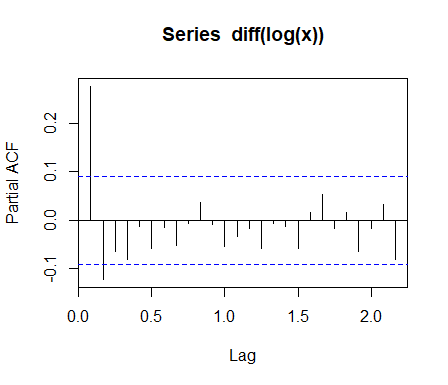
data: diff(log(x))

Dickey-Fuller = -8.6369, Lag order = 7, p-value = 0.01

alternative hypothesis: stationary

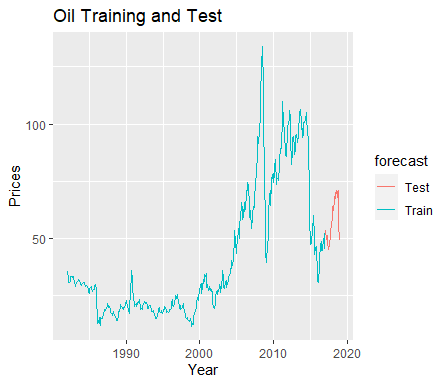
***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) >>pacf(diff(log(x)))



**Train-Test Split:**

> TS\_train <- window(x, start=c(1982,1), end=c(2016,12), freq=12)

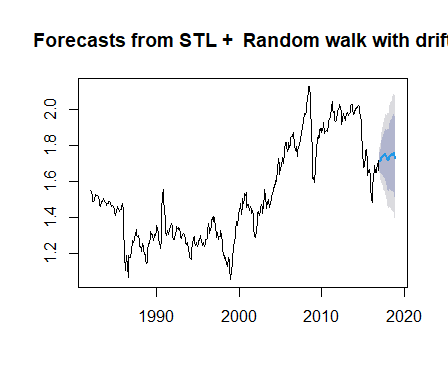
> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

**method’.**

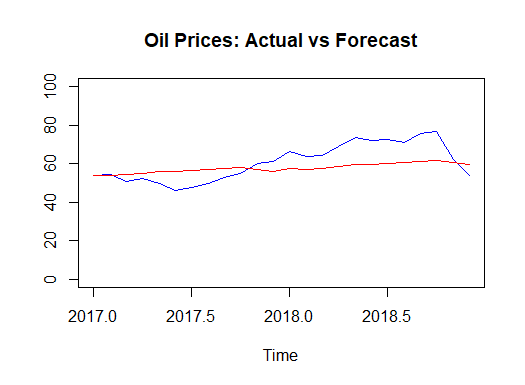


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 9.0315

> MAPE

[1] 0.1314

***As the MAPE value comes out to 0.1314, it means that there’s an 13.14 % error in our forecast and our forecast is 86.86% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(2,0,1)

Coefficients:

ar1 ar2 ma1

1.1287 -0.3232 -0.8487

s.e. 0.1281 0.0438 0.1339

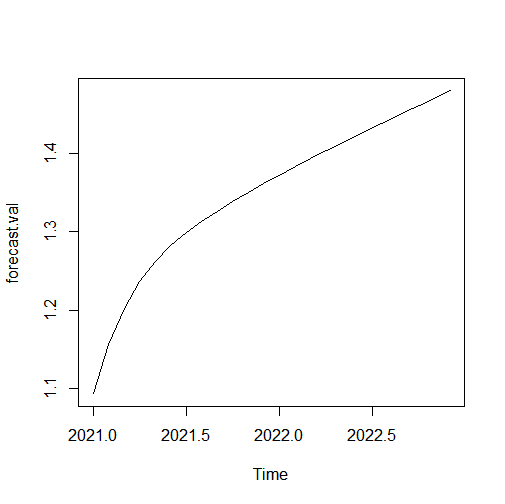
sigma^2 estimated as 0.00807: log likelihood=464.14

AIC=-920.27 AICc=-920.19 BIC=-903.69

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=2 and q=1***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(2,0,1))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices increase rapidly in the first half of 2021 and becomes completely linear over the course of the next 18 months. So we infer that WTI Oil prices will rise steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

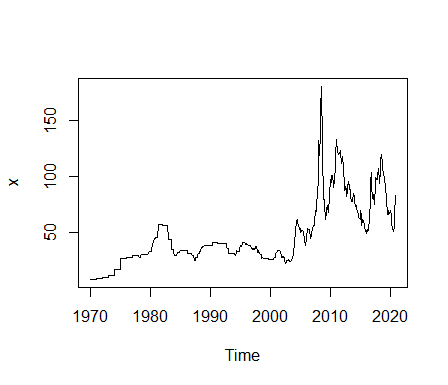
> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.061852, df = 1, p-value = 0.8036

***Since p value is greater than 0.05, the residuals are independent***.



**Coal Australia :**

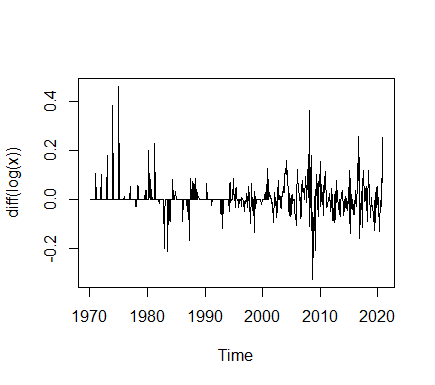
> x= final\_stat\_proj\_xls$`Coal Australia`

> x=x[121:734]

> x=as.numeric(x)

> x=ts(x, start= c(1970,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

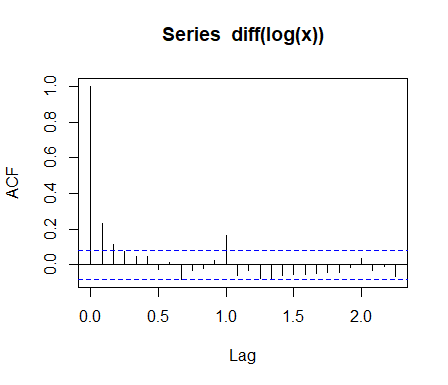
data: diff(log(x))

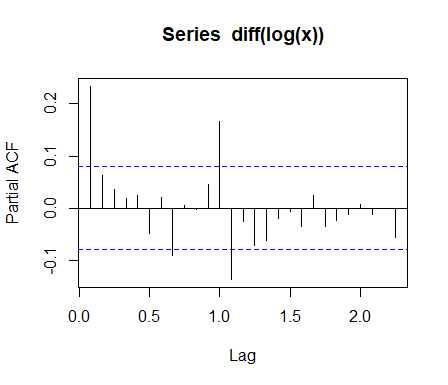
Dickey-Fuller = -7.9865, Lag order = 8, p-value = 0.01

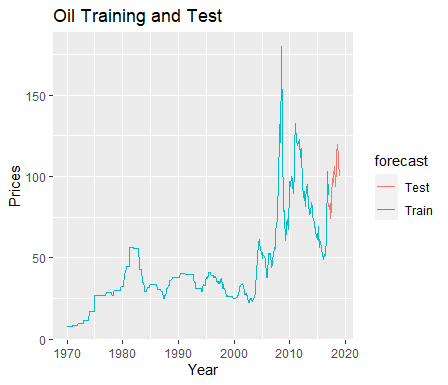
alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))



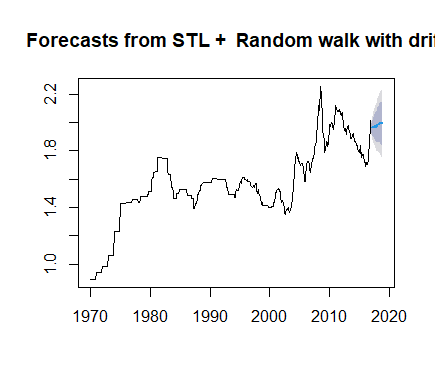




**Train-Test Split:**

> TS\_train <- window(x, start=c(1970,1), end=c(2016,12), freq=12)

> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

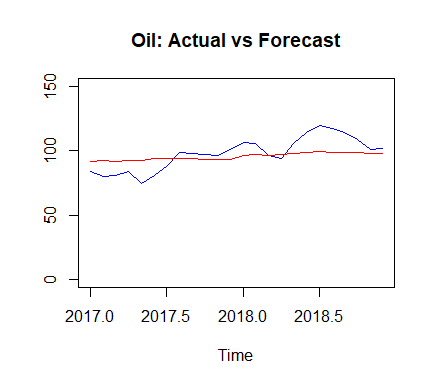
**apply the STL function and forecast it using the ‘random walk**

**method’.**

> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)



**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**

> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 10.5447

> MAPE

[1] 0.0934

***As the MAPE value comes out to 0.0934, it means that there’s an 9.34 % error in our forecast and our forecast is 90.66% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(1,0,1)(0,0,1)[12]

Coefficients:

ar1 ma1 sma1

0.6130 -0.3794 0.2113

s.e. 0.1129 0.1320 0.0406

sigma^2 estimated as 0.003194: log likelihood=889.8

AIC=-1771.59 AICc=-1771.53 BIC=-1753.93

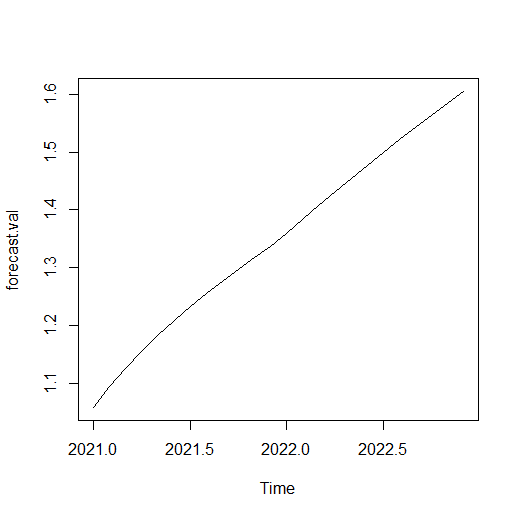
***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=1 and q=1 and Seasonal parameters P=0, D=0, Q=1***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**



> fit=arima(diff(log(x)), order=c(1,0,1),seasonal=list(order=c(0,0,1),period=12))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices first increase but register a very small dip at the start of 2022. They recover from it and become completely linear over the course of the next 12 months. So we infer that apart from the little dip, Australia Coal Prices will grow streadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

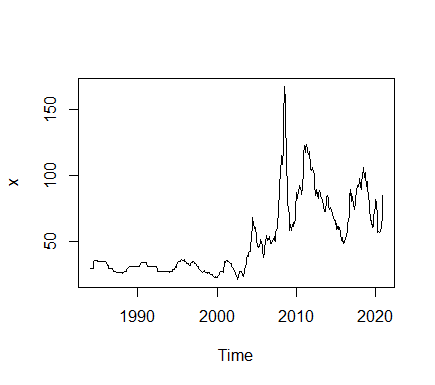
X-squared = 0.024723, df = 1, p-value = 0.8751

***Since p value is greater than 0.05, the residuals are independent.***

**COAL South Africa:**

> x= final\_stat\_proj\_xls$`Coal SA`

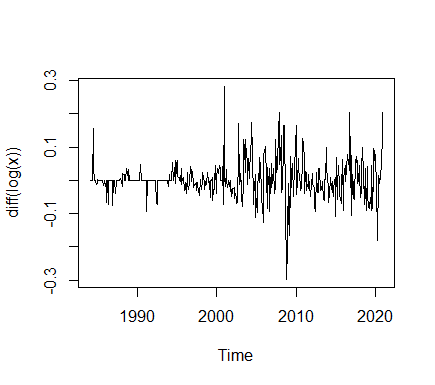
> x=x[289:734]



> x=as.numeric(x)

> x=ts(x, start= c(1984,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

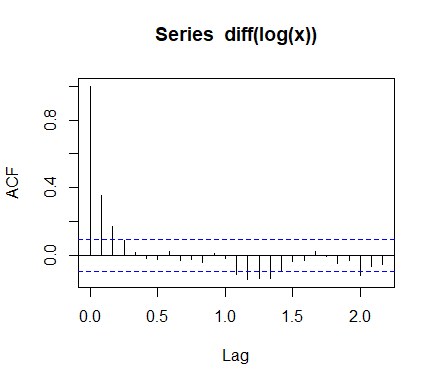
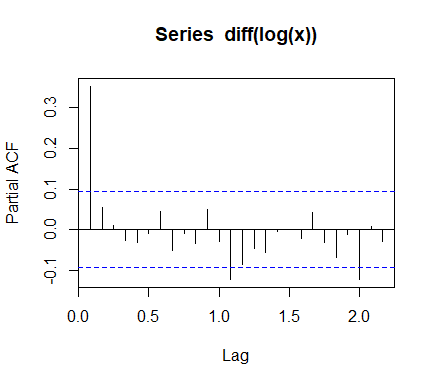
data: diff(log(x))

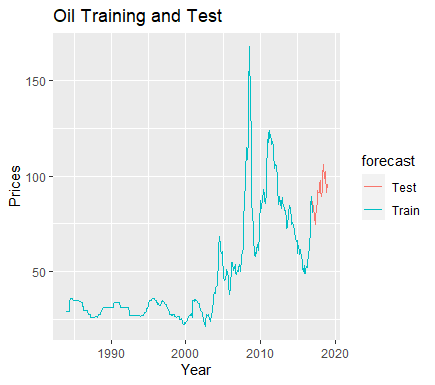
Dickey-Fuller = -6.7497, Lag order = 7, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))





**Train-Test Split:**

> TS\_train <- window(x, start=c(1984,1), end=c(2016,12), freq=12)

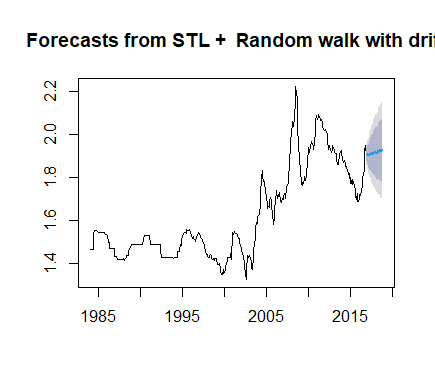
> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

**method’.**

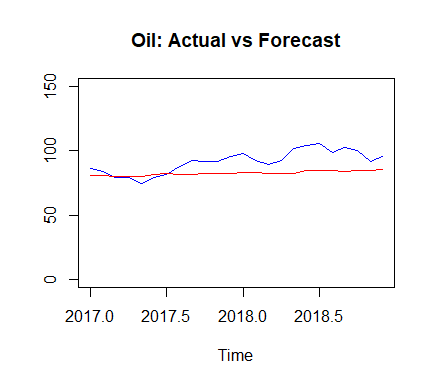


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**

> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 11.4537

> MAPE

[1] 0.101

***As the MAPE value comes out to 0.101, it means that there’s an 10.1% error in our forecast and our forecast is 89.9% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(1,0,0)(0,0,2)[12]

Coefficients:

ar1 sma1 sma2

0.2328 0.0612 -0.0522

s.e. 0.0361 0.0370 0.0356

sigma^2 estimated as 0.003966: log likelihood=985.36

AIC=-1962.71 AICc=-1962.66 BIC=-1944.33

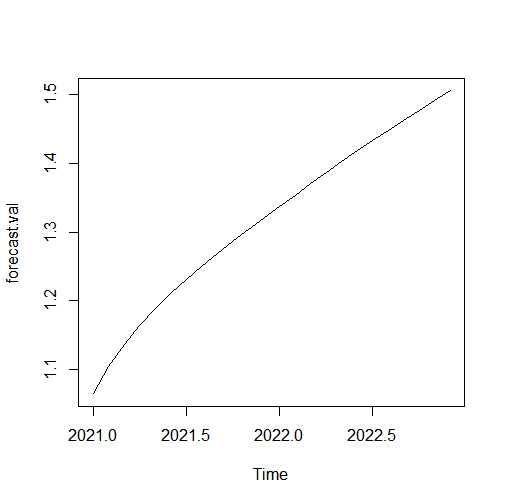
***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1 and Seasonal parameters P=0, D=0, Q=2***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(1,0,0),seasonal=list(order=c(0,0,2),period=12))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices increase rapidly at the start of the graph i.e. the first 4 months of 2021 and becomes completely linear over the course of the next 20 months. So we infer that Coal South Africa Prices will rise steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

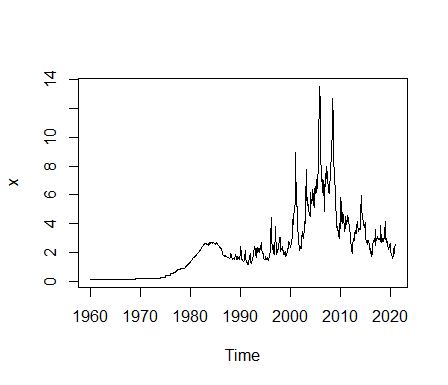
> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.1237, df = 1, p-value = 0.7251

***Since p value is greater than 0.05, the residuals are independent***.

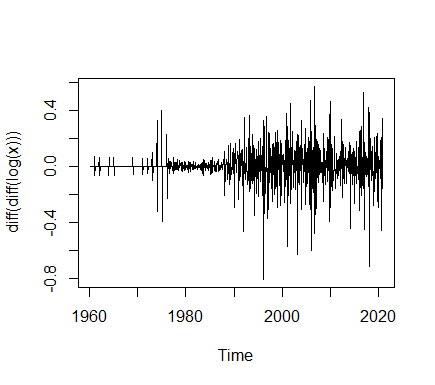


**NATURAL GAS,US:**

> x= final\_stat\_proj\_xls$`Natural Gas, Us`

> x=ts(x, start= c(1960,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(diff(log(x))))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

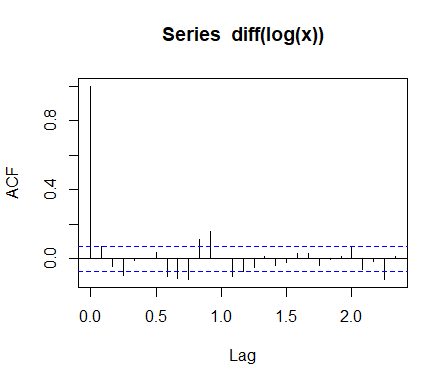
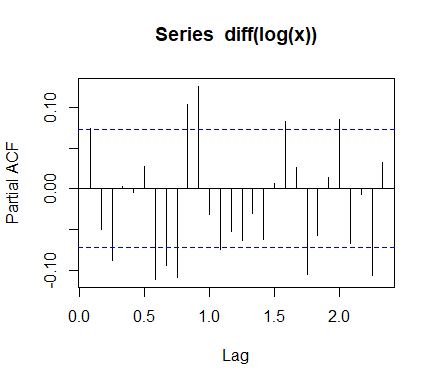
data: diff(log(x))

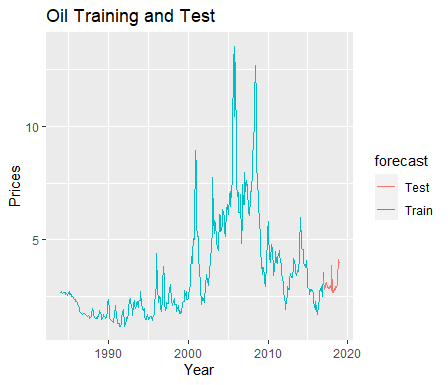
Dickey-Fuller = -9.8144, Lag order = 9, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))





**Train-Test Split:**

> TS\_train <- window(x, start=c(1984,1), end=c(2016,12), freq=12)

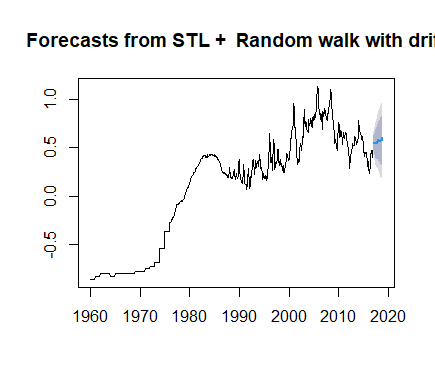
> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

**method’.**

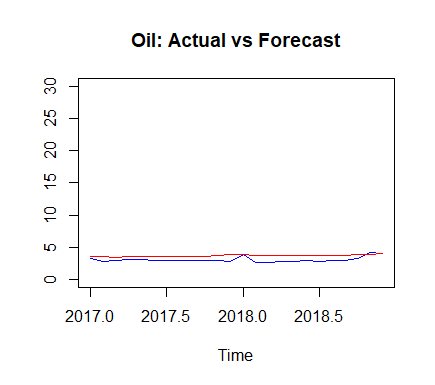


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 0.696

> MAPE

[1] 0.1904

***As the MAPE value comes out to 0.1904, it means that there’s an 19.04% error in our forecast and our forecast is 80.96% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(2,0,2)

Coefficients:

ar1 ar2 ma1 ma2

0.4700 -0.9233 -0.3893 0.9471

s.e. 0.0346 0.0249 0.0322 0.0159

sigma^2 estimated as 0.009551: log likelihood=664.32

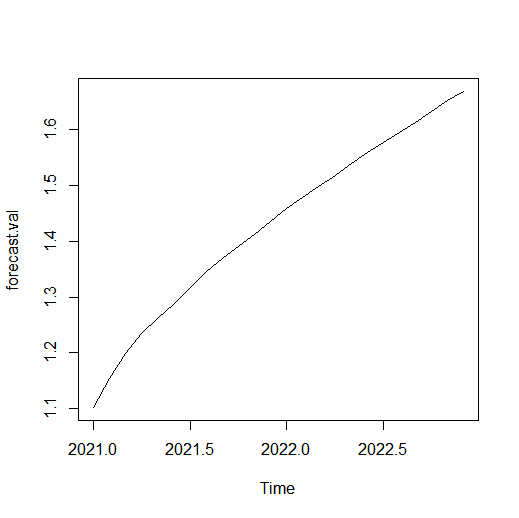
AIC=-1318.63 AICc=-1318.55 BIC=-1295.66

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(2,0,2))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices see a steep increase till the first half of 2021 and becomes approximately linear over the course of the next 18 months. So we infer that Natural Gas(US) Prices will rise steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

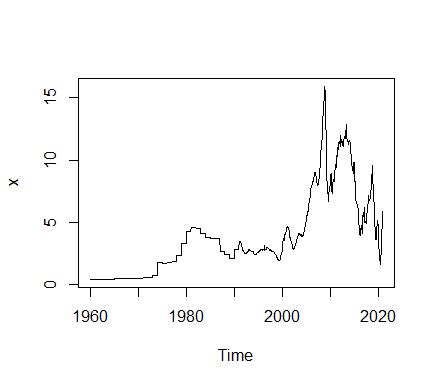
> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.20032, df = 1, p-value = 0.6545

***Since p value is greater than 0.05, the residuals are independent.***

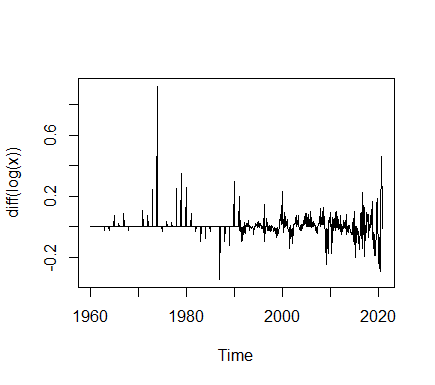


**NATURAL GAS,EUROPE:**

> x= final\_stat\_proj\_xls$`Natural Gas, Europe`

> x=ts(x, start= c(1960,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx)

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

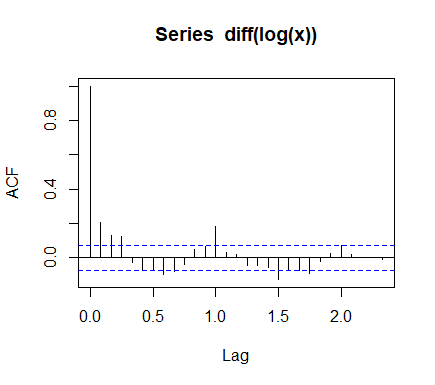
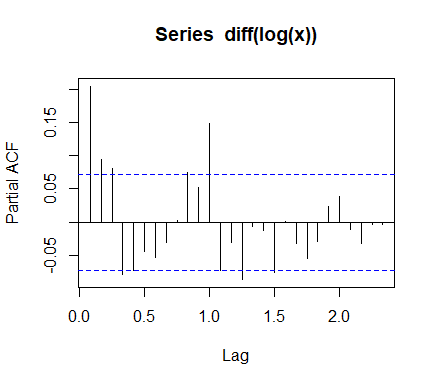
data: diff(log(x))

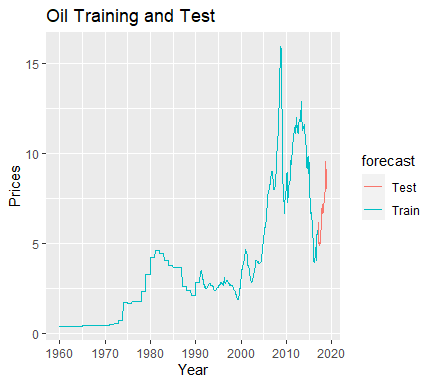
Dickey-Fuller = -8.1917, Lag order = 9, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))





**Train-Test Split:**

> TS\_train <- window(x, start=c(1960,1), end=c(2016,12), freq=12)

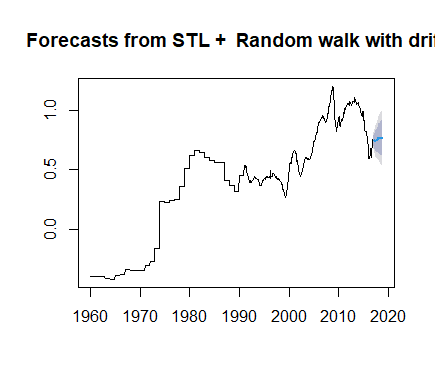
> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

**method’.**

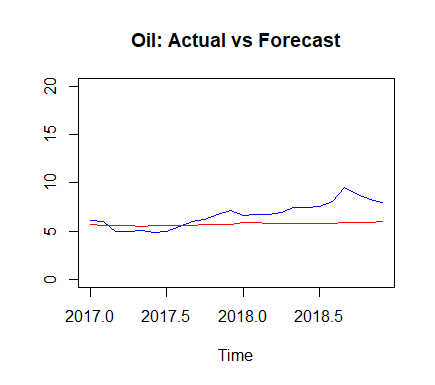


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 1.499

> MAPE

[1] 0.1657

***As the MAPE value comes out to 0.1657, it means that there’s an 16.57 % error in our forecast and our forecast is 83.43% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(1,0,1)(0,0,2)[12]

Coefficients:

ar1 ma1 sma1 sma2

0.6138 -0.4290 0.1934 0.0762

s.e. 0.0917 0.1021 0.0396 0.0450

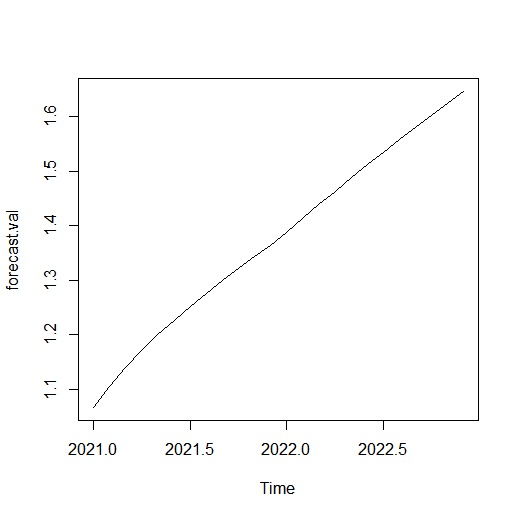
sigma^2 estimated as 0.004201: log likelihood=964.68

AIC=-1919.36 AICc=-1919.27 BIC=-1896.38

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1 and Seasonal parameters P=0, D=0, Q=2***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(1,0,1),seasonal=list(order=c(0,0,2),period=12))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices first increase but register a very small dip at the start of 2022. They recover from it and become completely linear over the course of the next 12 months. So we infer that apart from the little dip, Natural Gas(Europe) will grow steadily over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

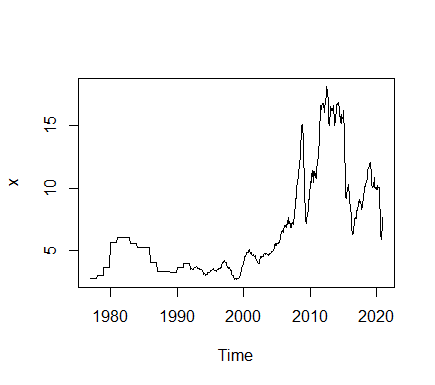
> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.10382, df = 1, p-value = 0.7473

***Since p value is greater than 0.05, the residuals are independent***.

**NATURAL GAS, JAPAN:**

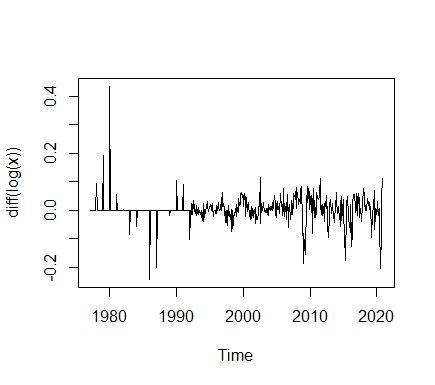
> x= final\_stat\_proj\_xls$`Natural Gas, Japan`

> x=x[205:734]

> x=as.numeric(x)

> x=ts(x, start= c(1977,1), end= c(2020,12), frequency = 12)

> plot(x)



**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

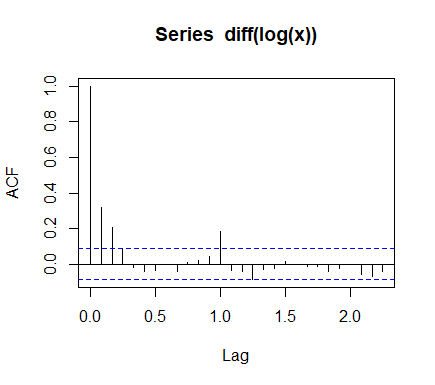
data: diff(log(x))

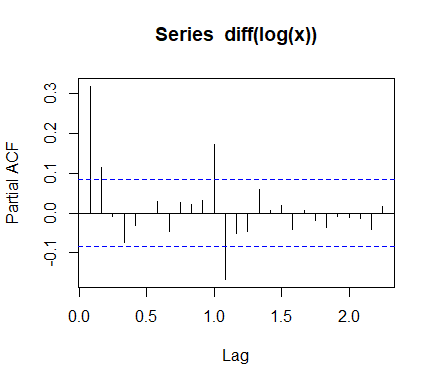
Dickey-Fuller = -7.2279, Lag order = 8, p-value = 0.01

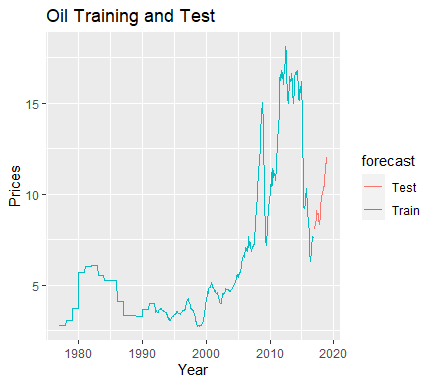
alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))







**Train-Test Split:**

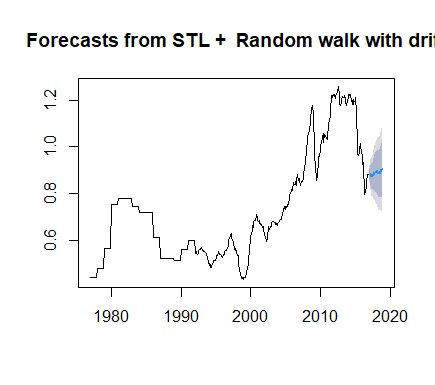
> TS\_train <- window(x, start=c(1977,1), end=c(2016,12), freq=12)

> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

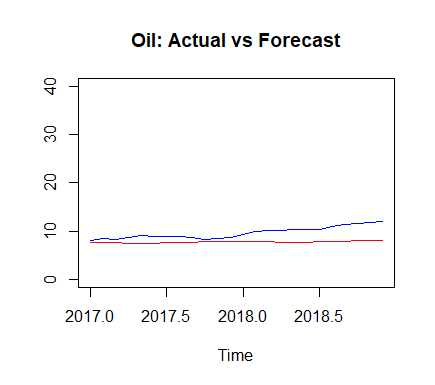
**method’.**

> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

> ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 2.1955

> MAPE

[1] 0.1868

***As the MAPE value comes out to 0.1868, it means that there’s an 18.68 % error in our forecast and our forecast is 81.32% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(0,0,3)(2,0,0)[12]

Coefficients:

ma1 ma2 ma3 sar1 sar2

0.2952 0.2248 0.1560 0.2493 -0.0157

s.e. 0.0427 0.0430 0.0456 0.0453 0.0464

sigma^2 estimated as 0.001692: log likelihood=935.88

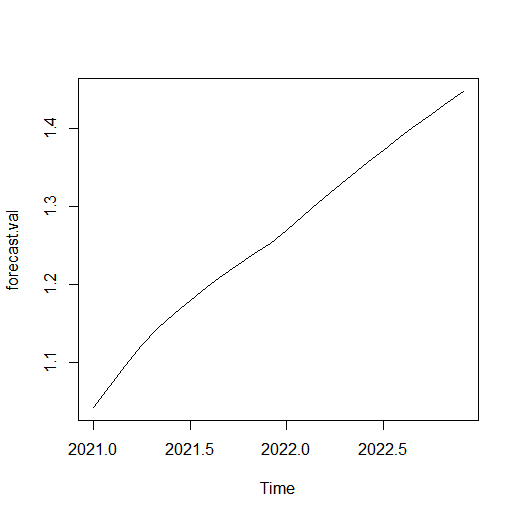
AIC=-1859.77 AICc=-1859.61 BIC=-1834.17

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1 and Seasonal parameters P=2, D=0, Q=0***

**General ARMA Model**



Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(0,0,3),seasonal=list(order=c(2,0,0),period=12))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. We see that the prices first increase but register a very small dip at the start of 2022. They recover from it and become completely linear over the course of the next 12 months. So we infer that Natural Gas(Japan) will rise steadily over the next 2 years.***

**Residual Analysis**

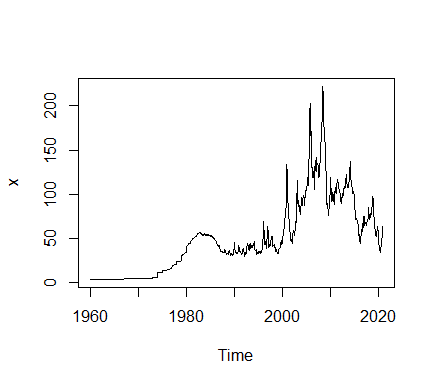
> fit.res=residuals(fit)

> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.014439, df = 1, p-value = 0.9044

***Since p value is greater than 0.05, the residuals are independent***.

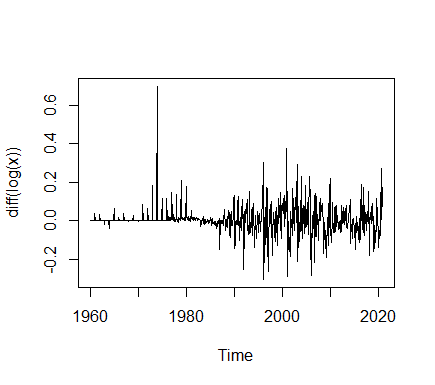
**NATURAL GAS INDEX:**

> x= final\_stat\_proj\_xls$`Natural Gas index`

> x=ts(x, start= c(1960,1), end= c(2020,12), frequency = 12)

> plot(x)

**#Log transformation and one-time differencing**

**make variance constant and lose the trend.**

>plot(diff(logx))

**Checking for Stationarity**

> adf.test(diff(log(x)))

Augmented Dickey-Fuller Test

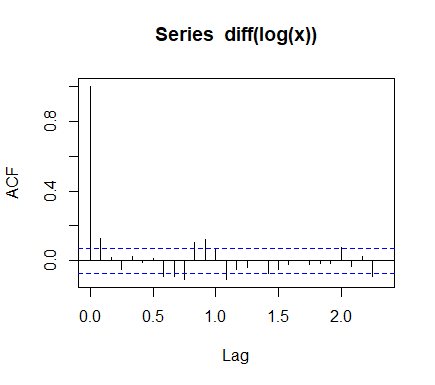
data: diff(log(x))

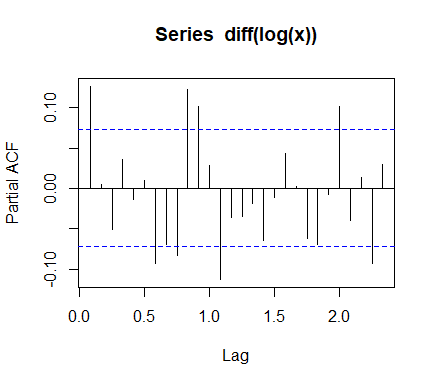
Dickey-Fuller = -8.8753, Lag order = 9, p-value = 0.01

alternative hypothesis: stationary

***Therefore,our transformed time series diff(log(x) is stationary******unlike x and log(x) which on applying adf.test(), gave value of p>0.05.***

> acf(diff(log(x))) > pacf(diff(log(x)))

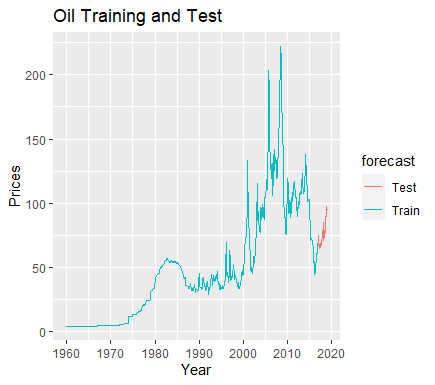




**Train-Test Split:**

> TS\_train <- window(x, start=c(1960,1), end=c(2016,12), freq=12)

> TS\_test <- window(x, start=c(2017,1) ,end=c(2018,12), freq=12)

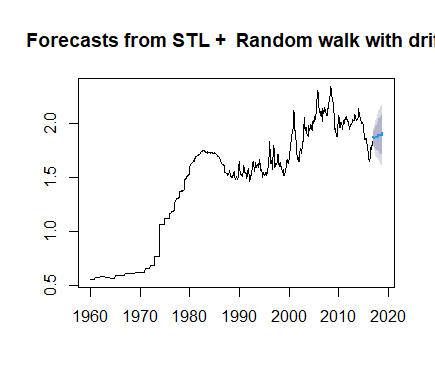


**#Our series is Multiplicative by nature. We take the log of our**

**train set, effectively converting it Into additive series. Then we**

**apply the STL function and forecast it using the ‘random walk**

**method’.**

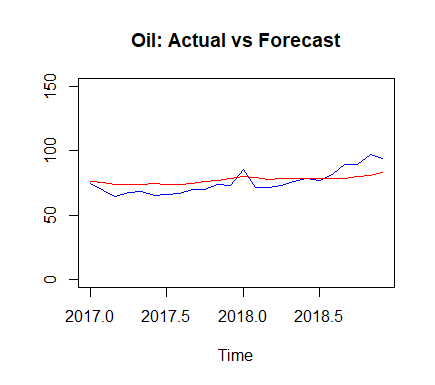


> TSDec\_train\_Log <- stl(log10(TS\_train), s.window = 'p')

> TS\_Train\_stl<- forecast( TSDec\_train\_Log, method ="rwdrift", h=24)

> plot(TS\_Train\_stl)

**#We create a vector which consists of our forecasted values from TS\_Train\_stl function and our original test set values and bind them together using the ‘cbind’ function**



> Vec2 <- 10^(cbind(log10(TS\_test), as.data.frame(forecast(TSDec\_train\_Log, method ="rwdrift", h=24))[,1]))

>ts.plot(Vec2, col=c("blue","red"), main="Oil: Actual vs Forecast")

**#Now, we compare the two using two statistical metrics RMSE and MAPE.**

> RMSE <- round(sqrt(sum(((Vec2[,1] - Vec2[,2])^2)/length(Vec2[,1]))),4)

> MAPE <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4)

> RMSE

[1] 7.4021

> MAPE

[1] 0.0845

***As the MAPE value comes out to 0.0845, it means that there’s an 8.45 % error in our forecast and our forecast is 91.55% accurate.***

**ARIMA Modelling**

> auto.arima(diff(log(x)))

Series: diff(log(x))

ARIMA(0,0,1)(0,0,2)[12]

Coefficients:

ma1 sma1 sma2

0.1273 0.0606 0.0756

s.e. 0.0355 0.0375 0.0350

sigma^2 estimated as 0.005192: log likelihood=886.96

AIC=-1765.91 AICc=-1765.86 BIC=-1747.54

***As we’ve already made our original time series stationary by using transformations, auto.arima() function tells us there’s no need for differencing now and we get an ARMA model with parameters p=0 and q=1 and Seasonal parameters P=0, D=0, Q=2***

**General ARMA Model**

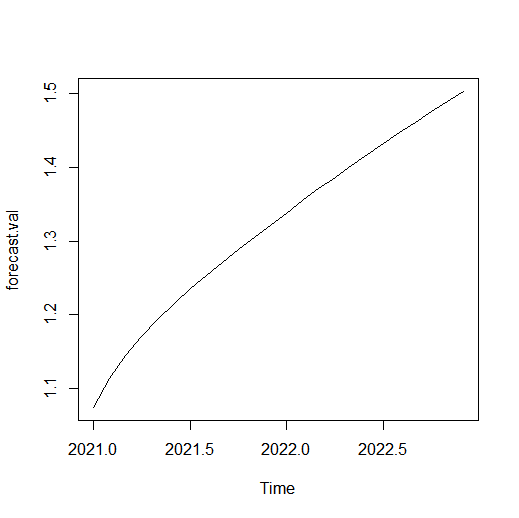


Where, φ 1 , … , φ p {\displaystyle \varphi \_{1},\ldots ,\varphi \_{p}}  are [parameters](https://en.wikipedia.org/wiki/Parameter) of the AR model,  are parameters of the MA model, c c {\displaystyle c} c cis a constant, and the random variable  ε t {\displaystyle \varepsilon \_{t}} is the [white noise](https://en.wikipedia.org/wiki/White_noise)

**Forecasting using our fitted ARMA model**

> fit=arima(diff(log(x)), order=c(0,0,1),seasonal=list(order=c(0,0,2),period=12))

> pred=predict(fit,n.ahead=24)

> pred=diffinv(pred)

> forecast.val=exp(pred)

> plot(forecast.val)

***We see an almost-linear increase suggesting trend but minimal seasonality. The prices increase fast at the start and gradually show linear rise from mid 2021 to end 2022. So we infer that Natural Gas Index prices will rise over the next 2 years.***

**Residual Analysis**

> fit.res=residuals(fit)

> Box.test(fit.res,type="Ljung")

Box-Ljung test

data: fit.res

X-squared = 0.0038098, df = 1, p-value = 0.9508

***Since p value is greater than 0.05, the residuals are independent.***

**Conclusions:**

1. Our original time series for each column wasn’t stationary so log transformation and first order differencing was required to make it stationary.
2. For modelling using ‘Random Walk’ method of the STL function, the actual and forecasted values have small differences as can be inferred from the graph.
3. The forecasts are accurate for all the 10 columns as any forecast with less than 20% Mean Absolute Percentage Error(MAPE) is considered to be good.
4. Forecasted values using ARIMA show an increasing trend for all 10 columns meaning that oil prices will increase gradually in 2021-2022 provided the COVID pandemic subsides.
5. Residuals for all 10 columns are independent.