Modeling Monthly Dow Jones Industrial Average (DJIA) Index to Forecast the Future Value.

A coursework project.

Niranjan Adhikari, Kevin Johnston & Bir Chhetri

MATH 1307 Forecasting

RMIT University

October 2017

Abstract

This study aims to fit different forecasting models for the monthly Dow Jones Industrial Average (DJIA or DJI) index, and make two years ahead forecasts by using the best-fitted model. As crude oil is an important commodity traded in the financial markets globally, time series linear regression is also used by considering the price of crude oil as a predictor variable. Through careful examination of each model fitted on the time series, state space model was found to yield better fit for the DJI series. The maximum likelihood criterion (AIC & BIC), mean absolute scaled error (MASE), BG test and diagnostic checks are used for evaluating the accuracy of each model. In addition, the dataset was divided into training and test sets to cross-validate each model. We use time series plots as a cross-validation procedure to evaluating the model accuracy. Although the prediction of the stock market is a challenging task, the forecast of the state space model did capture the trend of the index. The forecast shows there is potential for the index to continue its uptrend in the next two years. However, financial markets are influenced by numerous indicators, and our study is purely quantitative and focuses only on historical prices.

Table of Content

1 Introduction

There are several factors that affect the stock market. Fundamental analysts examine economic and financial indicators to determine the performance of a company's stock price, whilst technical analysts (or chartists) study only the historical prices to find a trend or pattern to forecast the future prices.

This study focuses on forecasting the Dow Jones Industrial Average(DJIA) or simply DJI. The DJI is the American stock market index which consists of 30 blue chip companies. The study is aimed at fitting a forecasting model to the DJI series of last 30 years historical data ranging from 1987 to 2017. We investigate the association between crude oil price and DJI series and fit time series regression model considering crude oil as a predictor. Oil price used here is the price of crude oil per barrel as traded in the international markets. This study is based on our intuition of possible correlation between these two markets.

We implement different types of forecasting models such as time series regression, exponential smoothing, state space for a best fit of the Dow Jones Index. For the regression models, oil price series is used as the predictor and distributed Lag models (dLagM) and dynamic lag models (dynLM) are fitted.

This report is organized as follows: In the next session, we will briefly describe the dataset. Section 3 contains the overall methodology to perform the project. Section 4 gives brief introduction about the proposed methods and evaluation techniques. Section 5 presents the results. The final section contains the summary of the project, issue, and the scope.

2 Dataset

The data set contains the information of the DJI series and oil price series from September 1987 to June 2017. The dataset is publicly available and obtained from two-public websites *Au.finance.yahoo.com* and *Indexmundi.com*.

There is a positive correlation of 0.64 between the two series (source R code). The plots in Fig 2.1 and Fig 2.2 compare the trends of series DJI and Oil. There appears to be a noticeable intervention in 2009 in both series though it is more apparent in case of oil series. This could be attributed to the impact of the global financial crisis which occurred during this time. DJI index has trended upward from 2000 levels in the 1990s to a current level of 21000. There is a gradual increase in index except for that historic fall where almost 50% of index value was lost during 2009. With regards to Oil price, it has increased from \$15 in 1987 to over \$100 just before that big fall in around the same time. However, the fall was preceded by DJI index.

Multiple trends can be observed in the series, and there is no apparent effect of seasonality. The increasing and decreasing points in the series suggests autoregressive behavior in the series. Fluctuation around mean level suggests the moving average behavior in the series. Change of the variance is not apparent in case of DJI until 2008, however, the oil price consistently exhibits changing variances throughout the series.

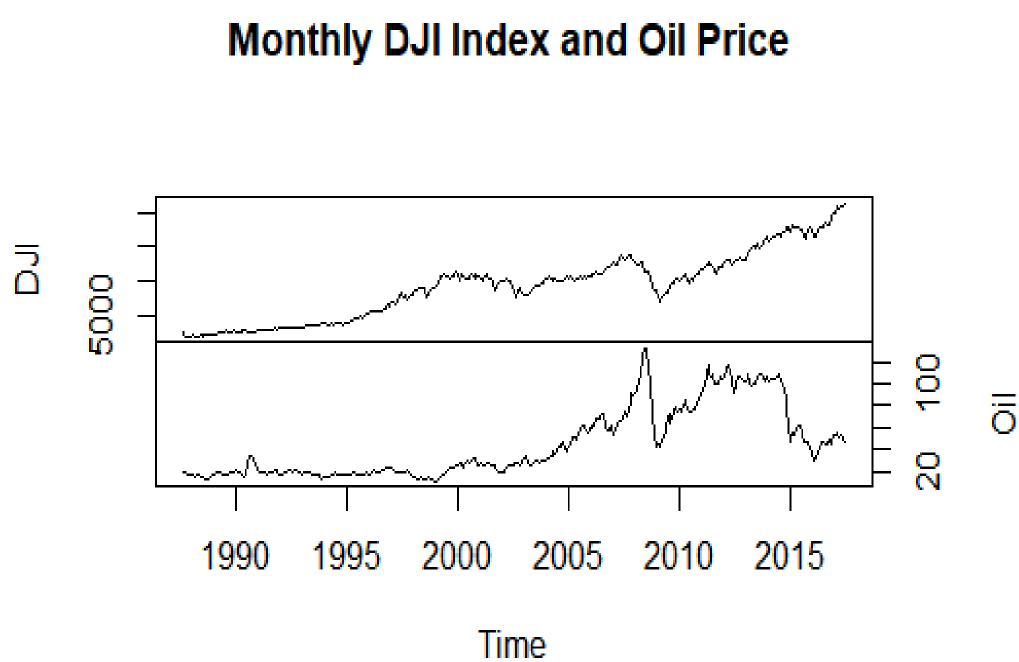


Figure 2.1 Time series Plot of Both Series

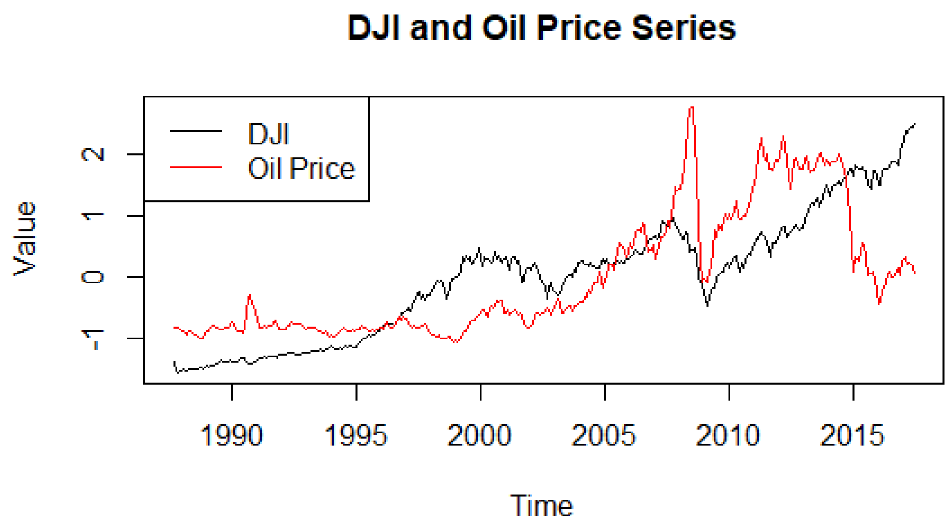


Figure 2.2 Combine Times Series Plot

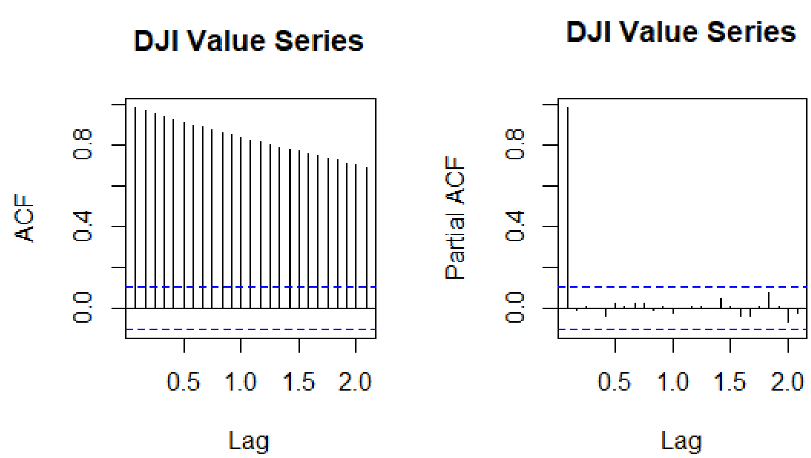


Figure 2.3 ACF and PACF of DJI Series

All the ACF lags of DJI series are significant but slowly decaying. The First lag of the PACF is significant. Visual inspection of the ACF and PACF indicate that there is no evidence of seasonality as there are no repeating patterns in the ACF. However, we observed the multiple trends in the DJI series which is indicated by the slow decay in the series.

3 Methodology

The first step of model fitting was to split the data into training and test set. Data from 1987-2015 was used as the training set and 2015- 2017 was used as the test. Then various models were fitted to the training dataset. These included modeling approaches dlagM, dylnm, Exponential smoothing and State space model. Subsequently, the best model from each of these models was selected. The selected model is then used for forecasting two years ahead (2015 to 2017) and compared to the test series (with visual plots). Finally, the Best model fitted in entire DJI time series is used to forecast the index value from 2017 to 2019.

The overall step of the methodology is listed below.

Step1: Plotting the two-time series together to understand the series

Step2: Partition whole series into training and testing sets (95%:5%)

Step3: Build the all possible models on training series

Step4: Select the best model from each class for models using AIC, BIC or MASE values, residual check, BG test, and forecast the training set

Step5: Comparison of models (Eg: regression (dlagM and dynlm), Exponential and State Space) using cross-validation procedure using test series. During this procedure, we forecast the training series and select the model whose forecast is closest to the original values.

Step6: Fit the selected model from step 5 to entire DJI series.

Step7: Forecast two years ahead DJI using the best model fitted to entire DJI series

4 Model Building and Selection

4.1 Proposed Models

Three different types of modeling approaches were adopted, namely time series regression model, exponential smoothing and state space model. In each of these models, all possible models were explicitly fitted by varying the parameters of the case by case basis in accordance with forecasting approach. This is to ensure the model on the training set captures important trends and patterns of the series and yields the maximum accuracy in the test set. The brief introduction of the models is as follows.

4.1.1 Time Series Regression Model

An explanatory variable is one that provides the forecast with additional information (Rob et al. 2008). In general, the lags of the explanatory variable are treated as predictor while dealing the time series models. In a similar way if we introduce another series as the predictor series, and the lags of the predictive variable will explain the overall variation and correlation structure of the dependent variables. Such kind of regression approach is known as *time series regression*. There are two types of regression approach in time series: linear innovation of time series regression model and dynamic time series regression model.

The one which deals with the linear time series is distributed lag models. There are two types of distributed lags models such as finite distributed lag model and infinite distributed lag model.

Another time series regression which can be applied to the dynamic nature of time series and more compatible with the intervention analysis is known as a dynamic time series regression model. This model performs well for both trend and seasonally affected series with large intervention.

In this study, we first try to implement as much time series regression models as possible to fit DJI series. In this approach, oil price series of same nature and time interval is introduced as a predictor series.

4.1.2 Exponential Smoothing Model

The exponential smoothing methods are used to analyze the time series data by fitting different trend and seasonal patterns together. There are the variety of methods falling under the family of exponential smoothing, each having the property that forecasts are weighted combination of past observations, with recent observations are weightier than older (Rob et al. 2008). The first idea of this concept was originated by Robert G. Brown in 1944 while he was working for the US Navy as an Operation Research Analyst (Rob et al. 2008).

Trend component	Seasonal component		
	N (None)	A (Additive)	M (Multiplicative)
N (None)	N,N	N,A	N,M
A (Additive)	A,N	A,A	A,M
A_d (Additive damped)	A_d,N	A_d,A	A_d,M
M (Multiplicative)	M,N	M,A	M,M
M_d (Multiplicative damped)	M_d,N	M_d,A	M_d,M

Figure 4.1 Possible combination of exponential smoothing method

In our study, we fit the most popular exponential smoothing model such as simple exponential smoothing (N, N), Holt's linear method (A, N), the damped trend model (Ad, N) and Holt-Winter's seasonal method (A, A and A, M).

4.1.3 State Space Model

The state space model is the composition of linear and Gaussian models; therefore, properties and results of multivariate normal distribution are applied. A prominent feature of the state space models is it allows considerable flexibility in the specification of the overall parametric structure (Rob et al. 2008). State space methods offer a unified approach to a wide range of models and techniques such as dynamic regression and many ad-hoc filters. Since the exponential smoothing method is an algorithm for point forecasts, unlike the exponential smoothing method, stochastic state space model provides the framework for computing prediction intervals and other properties (Rob et al. 2008). For each exponential smoothing model, there are two corresponding state space models.

In our study, we try to implement the different combination of the innovation of the state space methods while we do not prefer to choose the seasonal component since our data is free from the seasonality.

4.2 Model Selection and Forecast Evaluation

We build all possible set of models for mentioned three types of popular forecasting approaches discussed above using the training series. We use the AIC, BIC, MASE, BG test and residual analysis selecting the best model amongst the different types of models fitted. Essentially, we forecast from the training series and compare the forecasted series with the test series for cross-validation by visually inspecting each forecasted series. A brief introduction to each method of model evaluation is described below.

Akaike information criterion (AIC) and Bayesian information criterion (BIC): These are the maximum likelihood criterion method. These methods resolve the issue of overfitting by introducing the penalty term for the number of parameters in the model. Under these methods, we chose the model with the lower AIC and BIC values.

Mean Absolute Scaled Error (MASE): This is used to test the forecast accuracy of the models without the problems seen in the other measurements (Rob 2005). A more accurate model is reflected by a lower MASE value.

Breusch-Godfrey (BG) Test: BG test is used to test higher-order serial correlation. The null hypothesis (H_0) of the test is that there is no serial correlation in the residuals up to the specified order. If type is set to "F", the test statistics follows the F distribution. *(source of information: David & Achim from R studio)*

Residual Analysis: Our theoretical beliefs are residual of the model are randomly scattered. Ideal time series plot of the residuals are required to be free from trend and equal variance. Another prominent assumption is that residuals follow the normal distribution. To observe the normality, we plot the histogram of the residuals along with trend line. Finally, the output from either a Breusch-Godfrey test or a Ljung-Box test is printed to see the freedom of the model.

Cross-Validation Procedure: The dataset was split (95:5) into training and test series respectively. The model was built on training series forecasts were made for respective models. We use time series plot and visually inspect how close each model's forecast is to the original series (test series) as cross-validation procedure.

5 Result and Discussion

5.1 Model comparison on test data

For further examination and evaluation, four models were selected and forecasts of respective models were plotted against the original series (test set). As we can see in fig.5.1, the plots were from 2015 to 2017 consisting of the original index value of DJI and display the forecasts series of each model. The Koyck-DLM and ETS with multiplicative error and multiplicative trend are a linear line because they are point forecast value (mean value).

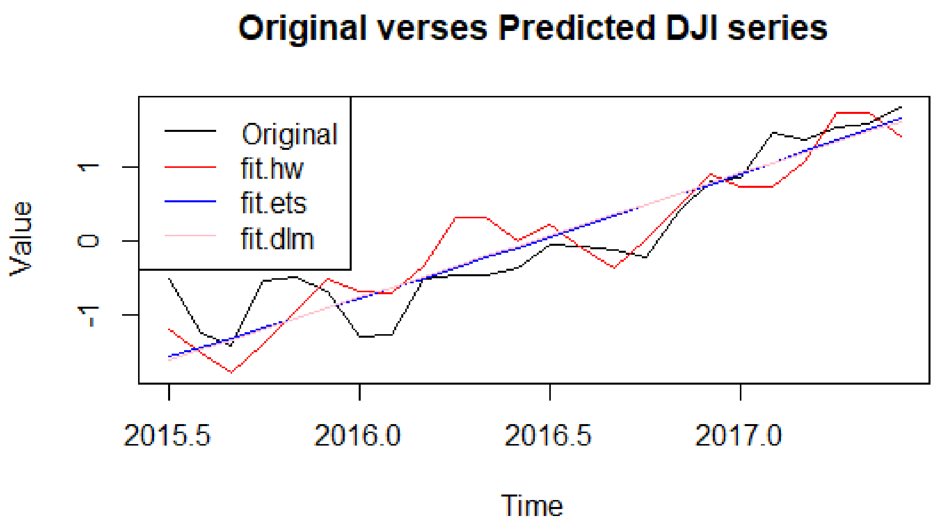


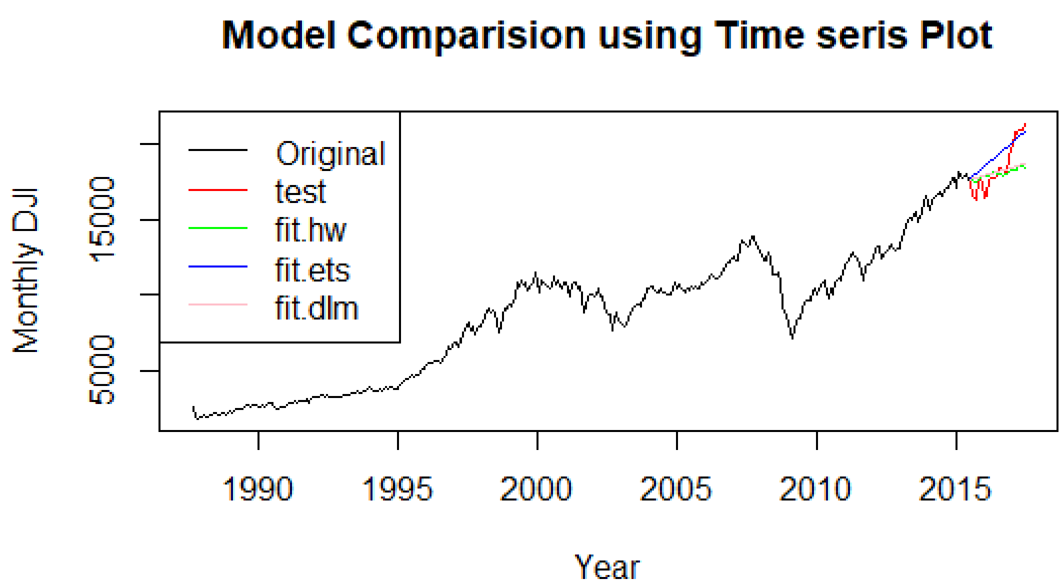
Figure 5.1 Cross-validation of models

These plots provide an indication of how close the forecasted values are with the original test series. From the figure, the ETS model, though a point forecast, seems to accurately capture the true trend of DJI by closely following the test series. In subsequent sections, we further investigate and validate this model.

5.2 Comparison of Forecasts

The forecasts for each type of models are computed from training series which are compared to the test set in red. This plot combines both training series and test series of the original DJI series on which the forecast of the selected models are overlayed from 2015 to 2017.

This visualization of the plot helps us further decide on the best model and use it for forecasting. This process is known as cross-validation procedure. Yet again, ETS appears to best reflect the original trend of DJI when the entire series is visualized.



5.3 Model Accuracy of the best model

This section provides a summary of the test results and information with regards to the model selection. From the table below, the ETS(MMN) model has the lowest AIC, BIC, and MASE value. As we see (as highlighted) the AIC, BIC and MASE values are the least for ETS model. From the residuals analysis, ETS model produces a histogram which appears to be normal and the ACF is well within the bound.

Table 5.1Model Comparison

Selected Model	Model Type	MASE Value	AIC	AIC
DLAGM (Regression Models)	KOYCK	0.9761	-	-
Exponential Smoothing model	HW, Trend (M)	0.238	5913	5978
State Space Models	ETS (MMN)	0.2234	5792	5811

s

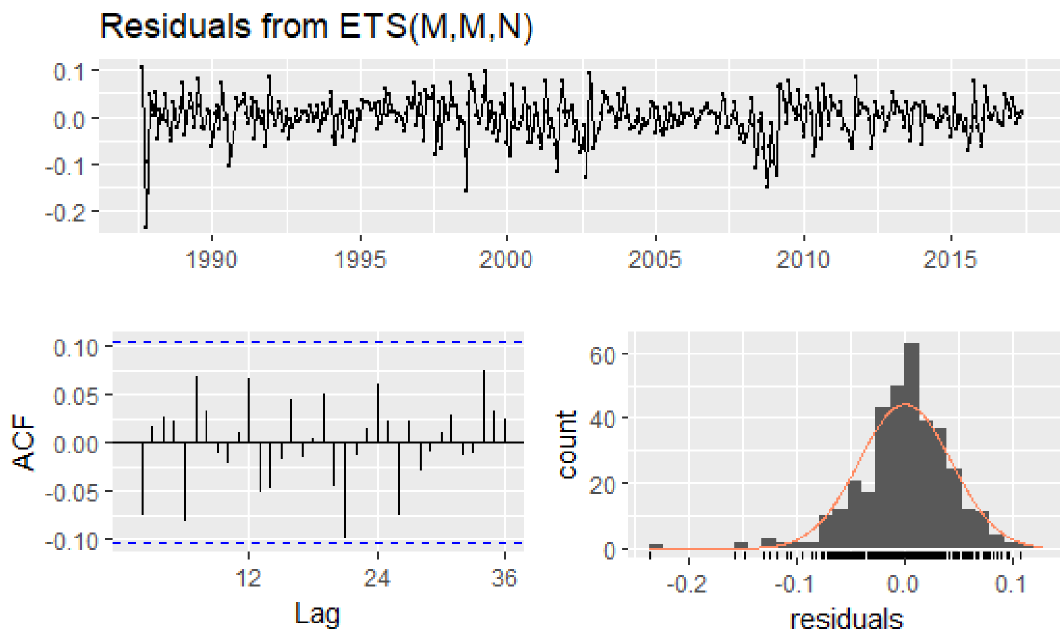


Figure 5.2Resisual analysis of the state space model with multiplicative error and multiplicative trend

5.4 Forecast Using the best Model

The visualization in fig 5.3 is the forecast for next 24 months using the state space model with multiplicative errors and multiplicative trends and no seasonality. With this forecast, we observed that the forecasted value best captured the upward trend of the DJI series. The confidence interval of the forecast increases with time as seen in the figure.

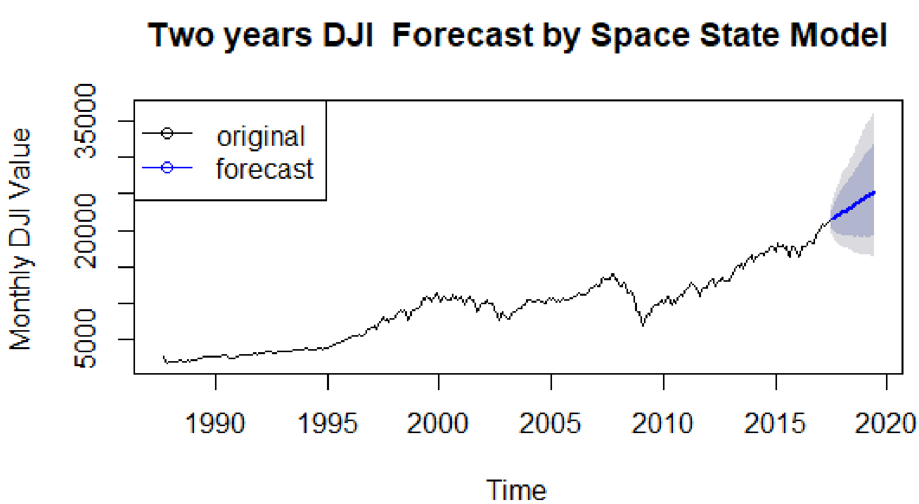


Figure 5.3 Future forecast using entire series

6 Conclusion

The DJI time series was modeled using all possible and appropriate forecasting methods. The state space model with multiplicative errors, multiplicative trend and no seasonality outperformed other modeling approaches. The results of our study indicate the Dow Jones index is most likely to continue its upward trend in the coming 24 months.

We found that the oil price as a predictor variable did not yield a good model on contrary to what we expected. This could be due to very volatile nature of oil prices multiple interventions present in the series.

While there clearly exists a positive correlation between DJI index and the price of oil, the linear regression did not yield a good fit. An improved approach would be to consider other advanced approaches removing the intervention in the oil series before regressing it with DJI if oil is to be used as the predictor. It may also be concluded that due to the of the unique nature of the markets further research on oil as a predictor in the time series regression models may be required to further understand the nature of oil price movement. In saying so, there is a significant scope of study which may be undertaken particularly in this area.

References

Cryer, J. D. & Chan, K. S. 2008, “*Time Series Analysis with Application in R*”, second edition published by Spring

Hyndman, R. J., Koehler, A. B., Ord, J. K., and Snyder. R. J. 2008, "*Forecasting with exponential Smoothing*”, published by Spring

Au.finance.yahoo.com 2017. ^DJI Historical prices | Dow Jones Industrial Average Stock - Yahoo Finance, viewed on 20th Sep 2017 <https://au.finance.yahoo.com/quote/%5EDJI/history?period1=475765200&period2=1506520800&interval=1mo&filter=history&frequency=1mo>.

Indexmundi.com 2017. Crude Oil (petroleum) - Monthly Price - Commodity Prices - Price Charts, Data, and News – IndexMundi, viewed on 20th Sep 2017. <http://www.indexmundi.com/commodities/?commodity=crude-oil&months=360 [Accessed 5 Oct. 2017>.

Appendix

```
rm(list=ls())

## Required Librabries
library(dynlm)
library(ggplot2)
library(AER)
library(Hmisc)
library(forecast)
library(dLagM)
library(TSA)
library(car)
library(expsmooth)
library(urca)
library(x12)
library(tseries)


## Mase Funtion
MASE.dynlm <- function(model, ... ){
  options(warn=-1)
  if(!missing(...)) {# Several models
  models = list(model, ...)
  m = length(models)
  for (j in 1:m){
    if ((class(models[[j]])[1] == "polyDlm") | (class(models[[j]])[1] == "dlm") | (class(models[[j]])[1] == "koyckDlm") | (class(models[[j]])[1] == "ardlDlm")){
      Y.t = models[[j]]$model$model$y.t
      fitted = models[[j]]$model$fitted.values
    } else if (class(models[[j]])[1] == "lm"){
      Y.t = models[[j]]$model[,1]
```

```
      fitted = models[[j]]$fitted.values
    } else if (class(models[[j]])[1] == "dynlm"){
      Y.t = models[[j]]$model$Y.t
      fitted = models[[j]]$fitted.values
    } else {
      stop("MASE function works for lm, dlm, polyDlm, koyckDlm, and ardlDlm objects. Please make sure that you are sending model object directly or send a bunch of these objects to the function.")
    }
    # Y.t = models[[j]]$model$y.t
    # fitted = models[[j]]$fitted.values
    n = length(fitted)
    e.t = Y.t - fitted
    sum = 0
    for (i in 2:n){
      sum = sum + abs(Y.t[i] - Y.t[i-1] )
    }
    q.t = e.t / (sum/(n-1))
    if (j == 1){
      MASE = data.frame( n = n , MASE = mean(abs(q.t)))
      colnames(MASE) = c("n" , "MASE")
    } else {
      MASE = rbind(MASE, c(n , mean(abs(q.t))))
    }
  }
  Call <- match.call()
  row.names(MASE) = as.character(Call[-1L])
  MASE
} else { # Only one model
```

```
    if ((class(model)[1] == "polyDlm") | (class(model)[1] == "dlm") | (class(model)[1] == "koyckDlm") | (class(model)[1] == "ardlDlm")){
      Y.t = model$model$model$y.t
      fitted = model$model$fitted.values
    } else if (class(model)[1] == "lm"){
      Y.t = model$model[,1]
      fitted = model$fitted.values
    } else if (class(model)[1] == "dynlm"){
      Y.t = model$model$Y.t
      fitted = model$fitted.values
    } else {
      stop("MASE function works for lm, dlm, polyDlm, koyckDlm, and ardlDlm objects. Please make sure that you are sending model object directly or send one of these objects to the function.")
    }
    n = length(fitted)
    e.t = Y.t - fitted
    sum = 0
    for (i in 2:n){
      sum = sum + abs(Y.t[i] - Y.t[i-1] )
    }
    q.t = e.t / (sum/(n-1))
    MASE = data.frame( MASE = mean(abs(q.t)))
    colnames(MASE) = c("MASE")
    Call <- match.call()
    row.names(MASE) = as.character(Call[-1L])
    MASE
  }
 }
## ####################################################################
```

```
#####################################################################
## Reading  dataset
setwd("…….")
data <- read.csv("DJI.csv", header = TRUE)
data1 <- data[, c("DJI", "Oil")]
data.ts <- ts(data1, start = c(1987,9), frequency = 12)
#View(data.ts)
class(data.ts)
### -------Non Emperical Analysis----------------------------------------
## Time Sereis Plot
plot(data.ts, yax.flip=T,  main = "Monthly DJI Index and Oil Price " )
cor(data.ts) # Moderately positive correlation
## Plotting both series tigether in the same plot
data.ts.scale= scale(data.ts)
plot(data.ts.scale, plot.type="s",col = c("black", "red"), main = "DJI and Oil Price Series", ylab = "Value")
legend("topleft",lty=1, text.width = , col=c("black","red"), c("DJI", "Oil Price"))
## DJI Series
DJI.ts <- data.ts[,1]
oil.ts <- data.ts[,2]
plot(DJI.ts, main = "Time Series Plot of DJI Series", ylab ="Monthly DJI Value" )
# points(y=DJI.ts,x=time(DJI.ts), pch=as.vector(season(DJI.ts))) ## This will be more messy
par(mfrow=c(1,2)) # Put the ACF and PACF plots next to each other
acf(DJI.ts, main = "DJI Value Series")
pacf(DJI.ts, main = "DJI Value Series")


## Explanation of the DJI series
# Multiple trends exist in the series, no any effect of seasonality, around 2009-2010 there was intervention effect. The following increasing and decreasing points in the series
```

suggest the auto- regressive behavior in the series. Fluctuation around mean level suggests the moving average behavior. change of the variation is obvious throughout the series. Since there are no any repeating patterns in the ACF while slowly decaying patterns of # highly significant lag suggest the absence of seasonal effect and presence of trend accordingly.

Oil Series

oil.ts <- data.ts[,2]

plot(oil.ts, main = "Time Series Plot of Oil Price Series", ylab ="Monthly Oil Price")

points(y=oil.ts,x=time(oil.ts), pch=as.vector(season(oil.ts))) ## This will be more messy

par(mfrow=c(1,2)) # Put the ACF and PACF plots next to each other

acf(oil.ts, main = "Oil Price Series")

pacf(oil.ts, main = "Oil Price Series")

Explanation of the Oil Price series

Multiple trends exist in the series, no any effect of seasonality, in 2010 and 2016 their # # was huge intervention effect. The following increasing and decreasing points in the # # # series suggest the auto- regressive behavior in the series. Fluctuation around means level # suggest the moving average behavior. Change of the variation is obvious throughout the # series. Since there are no any repeating patterns in the ACF while slowly decaying # # # # patterns of highly significant lag suggest the absence of seasonal effect and presence of # trend accordingly.

----------- Modelling the Series --

partition of data set for Cross validation procedure We have partitioned the dataset to # # ## ensure the last 2 years of data is useful for testing purpose

train <- data1[1:334,]   # choose the first 333 row as training set

test <- data1[335:358,] # choose the following 24 row as testing set

DJI.train <- ts(train[,1], start=c(1987,9), frequency = 12)

DJI.test <- ts(test[,1], start=c(2015,7), frequency = 12)

oil.train <- ts(train[,2], start=c(1987,9), frequency = 12)

oil.test <- ts(test[,2], start=c(2015,6), frequency = 12)

```
################################################################################
##############
## ----Regression Model (Distribute Lag Model)----dLagM Package
#Dlm model
dlm.1 = dlm(x = as.vector(oil.train ) , y = as.vector(DJI.train), q = 4, show.summary = TRUE)
vif(dlm.1$model)
checkresiduals(dlm.1$model)
bgtest(dlm.1$model)

# plyDlm model
dlm.2 = polyDlm(x = as.vector(oil.train ) , y = as.vector(DJI.train) , q = 4 , k = 2, show.beta = TRUE , show.summary = TRUE)
vif(dlm.2$model)
checkresiduals(dlm.2$model)
bgtest(dlm.2$model)

#KoyckDlm model
dlm.3 = koyckDlm(x = as.vector(oil.train ) , y = as.vector(DJI.train) , show.summary = TRUE)
vif(dlm.3$model)
checkresiduals(dlm.3$model)
bgtest(dlm.3$model)

## ArdDlm mdoel
dlm.4.1 = ardlDlm(x = as.vector(oil.train) , y = as.vector(DJI.train) , p = 1 , q = 1 , show.summary = TRUE)
checkresiduals(dlm.4.1$model)
bgtest(dlm.4.1$model)
```

BG test: The Breusch-Godfrey test for higher-order serial correlation. The null hypothesis (H0) of the test is

that there is no serial correlation in the residuals up to the specified order. If type is

set to "F", the test statistics follows the F distribution.

dlm.4.1.forecasts = ardlDlmForecast(model = dlm.4.1 , x = oil.test, h = 24)$forecasts

dlm.4.2 = ardlDlm(x = as.vector(oil.train) , y = as.vector(DJI.train) , p = 2 , q = 2 , show.summary = TRUE)

checkresiduals(dlm.4.2$model)

bgtest(dlm.4.2$model)

dlm.4.2.forecasts = ardlDlmForecast(model = dlm.4.2 , x = oil.test, h = 24)$forecasts

dlm.4.3 = ardlDlm(x = as.vector(oil.train) , y = as.vector(DJI.train) , p = 3 , q = 3 , show.summary = TRUE)

checkresiduals(dlm.4.3$model)

bgtest(dlm.4.3$model)

dlm.4.3.forecasts = ardlDlmForecast(model = dlm.4.3 , x = oil.test, h = 24)$forecasts

This forecast is just for experiment

plot(DJI.train, type="l", xlim = range(1987, 2018), ylim = range(DJI.train), ylab = "Monthly DDI", xlab = "Year",

main="Salar Radiation Forecast for Precipitaitn")

lines(ts(dlm.4.3.forecasts, end=c(2017, 6), frequency = 12),col="Red",type="l")

Comparision of Dlm models using aic, bic and mase

aic.models = AIC(dlm.1$model, dlm.2$model, dlm.4.1$model, dlm.4.2$model, dlm.4.3$model)

sortScore(aic.models, score= "aic")

```
bic.models = BIC(dlm.1$model, dlm.2$model, dlm.4.1$model, dlm.4.2$model, dlm.4.3$model)

sortScore(bic.models, score= "bic")

mase <- MASE(dlm.1, dlm.2, dlm.3, dlm.4.1, dlm.4.2, dlm.4.3)

sortScore(mase, score = "mase")


# It is found that the dlm.3 (KoyckDLM) model is better model according mase value

#### Residual of this model is quite impressive

dlm.3.forecasts = koyckDlmForecast(model = dlm.3 , x = oil.test, h = 24)$forecasts

dlm.forecast <- ts(dlm.3.forecasts, end=c(2017, 6), frequency = 12)


plot(DJI.train, type="l", xlim = range(1987, 2018), ylim = c(min(DJI.train),max(DJI.train)+3000),

    ylab = "Salar Radiation", xlab = "Year",

    main="Salar Radiation Forecast for Precipitaitn")

lines(dlm.forecast,col="red",type="l")

legend("topleft",lty=1, pch = 1, text.width = , col=c("black","red"), c("Original", "Forecast"))


############################################################################

## -------Dynamic Linear Model (dynlm package)----------------

## this is the model for intervention analsis using regression approach

Y.t = DJI.train

which(Y.t < -100)

## we god integer == 0

Y.t.1 =  Lag(Y.t,+1)

X.t = oil.train

#which(X.t < -100)

X.t.1 = Lag(X.t,+1)
```

```
dynlm.1 = dynlm(Y.t ~ X.t + L(X.t, k=1) + L(Y.t))
summary(dynlm.1)
checkresiduals(dynlm.1)

dynlm.2 = dynlm(Y.t ~ X.t + X.t.1 + L(Y.t) + trend(Y.t))
summary(dynlm.2)
checkresiduals(dynlm.2)

dynlm.3 = dynlm(Y.t ~ X.t + X.t.1 + L(Y.t) + trend(Y.t)+ trend(X.t))
summary(dynlm.3)
checkresiduals(dynlm.3)

dynlm.4 = dynlm(Y.t ~ X.t + X.t.1 + L(X.t , k = 2) + L(X.t, k=2) + L(Y.t , k = 1) +
trend(X.t))
summary(dynlm.4)
checkresiduals(dynlm.4)
bgtest(dynlm.4)

## comaparing dynlm models
bic.models = BIC(dynlm.1, dynlm.2, dynlm.3, dynlm.4)
aic.models =AIC(dynlm.1, dynlm.2, dynlm.3, dynlm.4)
mase_d <- MASE.dynlm(dynlm.1, dynlm.2, dynlm.3, dynlm.4)

sortScore(bic.models, score = "bic")
sortScore(aic.models,score = "aic")
sortScore(mase_d, score = "mase")

## Accoring to the diagonestic check of residulas, AIC, BIc and MASE values
## We have found that that the dynlm.4 model is the best
```

```
par(mfrow=c(1,1))
plot(Y.t, ylab='Monthly DJI',xlab='Year',main = "Time Series Plot of Monthly
    DJI Series")
lines(dynlm.4$fitted.values,col="red")
legend("topleft",lty=1, pch = 1, col=c("black","red"),c("Original","Fitted"))

####-------Dynlm forecast-------------------
q = 24
n = nrow(dynlm.4$model)
dji.frc = array(NA , (n + q))
dji.frc2 = array(NA , (n + q))
dji.frc[1:n] = X.t[4:length(X.t)]
dji.frc2[1:n] = Y.t[4:length(Y.t)]
trend = array(NA,q)
trend.start = dynlm.4$model[n,"trend(X.t)"]
trend = seq(trend.start , trend.start + q/12, 1/12)

for (i in 1:q){
  dji.frc[n+i] = oil.test[i]
  data.new = c(1,dji.frc[n+i],dji.frc[n-1+i],dji.frc[n-2+i],dji.frc2[n-1+i],trend[i])
  # data.new = c(1, predictor[i], predictor[i-1], predictor[i-2], dji.frc[n-1+i],dji.frc[n-
2+i],trend[i],months)
  dji.frc2[n+i] = as.vector(dynlm.4$coefficients) %*% data.new
}

fit.dynlm <- ts(dji.frc2[(n+1):(n+q)],start=c(2015,7),frequency = 12)

plot(Y.t, xlim=c(1987,2018), ylim = c(min(DJI.train),max(DJI.train)+3000),
    ylab='DJI ',xlab='Year',main = "Time Series Plot of Monthly DIJ Series")
```

```
lines(fit.dynlm,col="red",type="l")
```

think that the dynlm model over fit the series, so it gives the bad forecast. which we can observe

from the r-square value.

```
##########################################################################
########################################################################
## ---Exponential smotheing method-----------------------------
fit.ses = ses(DJI.train, initial="simple", h=24)
summary(fit.ses) ## Mase == 0.2401
checkresiduals(fit.ses)

fit.hw = hw(DJI.train, trend = "multipicative", damped = FALSE, h=24)
summary(fit.hw) ## Mase == 0.2373
checkresiduals(fit.hw)

fit.holt = holt(DJI.train, initial="optimal", exponential=FALSE, h=24)
summary(fit.holt) ## Mase == 0.2382
checkresiduals(fit.holt)
```

when comparing the value of mase and, the resiudal analysis we have observed fit.hw model

best capture the series. The following code demonstrates the forecast

```
## forecast of  exponentila smoothing series
plot(fit.hw, ylab="Monthly DJI", main = "Forecast from Simple Exponential Method",
    plot.conf=FALSE, type="l", fcol="blue", xlab="Year")
```

```
legend("topleft",lty=1, pch = 1, col=c("black","blue"),c("original","forecast"))

##############################################################################
##############################################################################
## State space model
fit.ets.1 = ets(DJI.train, model="MAN", damped = TRUE)
summary(fit.ets.1) ## AIc == 5782.371, BIC == 5805.219, MASE == 0.239
checkresiduals(fit.ets.1)

fit.ets.2 = ets(DJI.train, model="MMN")
summary(fit.ets.2) ## AIc == 5772.443, BIC == 5791.484, MASE == 0.233
checkresiduals(fit.ets.2)

fit.ets.3 = ets(DJI.train, model="ZZZ", opt.crit = "mse")
summary(fit.ets.3) ## AIc == 5770.892, BIC == 5782.316.546, MASE == 0.240
checkresiduals(fit.ets.3) ## this is the best method

## According to mase and residual analysis the fit.ets.2 model best capture the structure
# the series the following chunks of code best describe the series

## Prediction from space state model
plot(forecast(fit.ets.2), ylab="Monthly DJI ", plot.conf=FALSE,
    main="Monthly DJI  Forecast by Space State Model", type="l", xlab="Time")
legend("topleft",lty=1, pch = 1, col=c("black","blue"),c("original","forecast"))

##############################################################################
##############################################################################
## Final model evaluation using cross validation prcedure of ts plot of original
# predicted series
```

```
prediction= ts.intersect(DJI.test, fit.hw[]$mean, forecast(fit.ets.2, h=24)[]$mean,
                      dlm.forecast)
prediction.scale= scale(prediction)
plot(prediction.scale, plot.type="s",col = c("black", "red", "blue", "pink"),
            main = "Original verses Predicted DJI series", ylab = "Value")
legend("topleft",lty=1, text.width = , col=c("black","red", "blue", "pink"),
          c("Original", "fit.hw", "fit.ets", "fit.dlm"))
# # predicted series
#      prediction= ts.intersect(DJI.test, fit.hw[]$mean, forecast(fit.ets.2, h=24)[]$mean,
dlm.forecast, fit.dynlm)
#   prediction.scale= scale(prediction)
#   plot(prediction.scale, plot.type="s",col = c("black", "red", "blue", "pink", "gray"),
#           main = "Original verses Predicted DJI series", ylab = "Value")
#  legend("topleft",lty=1, text.width = , col=c("black","red", "blue", "pink", "gray"),
#                     c("Original", "fit.hw", "fit.ets", "fit.dlm", "fit.dynlm"))


## Cross validation of forecast line in original line
plot(DJI.ts, ylab="Monthly DJI", plot.conf=FALSE, type="l", xlab="Year",
                  main = "Model Comparision using Time seris Plot")
lines(DJI.test, col = "red")
lines(fit.hw[]$mean, col = "green")
lines(forecast(fit.ets.2, h =24)[]$mean, col = "blue", type = "l")
lines(dlm.forecast, col = "pink", type = "l")
# lines(fit.dynlm, col = "gray", type = "l")
legend("topleft",lty=1, text.width = , col=c("black","red", "green","blue", "pink"),
c("Original","test", "fit.hw", "fit.ets", "fit.dlm"))


### Plot of test versus forecast confidence interval of ets model
plot(DJI.ts, ylab="Monthly DJI", plot.conf=FALSE, type="l", xlab="Year",
```

```
    main = "Test and forecast confidence interval of ets model")
lines(DJI.test, col = "red")
lines(forecast(fit.ets.2, h =24)[]$mean, col = "blue", type = "l")
lines(forecast(fit.ets.2, h =24)[]$lower[,2] , col = "green", type = "l")
lines(forecast(fit.ets.2, h =24)[]$upper[,2], col = "green", type = "l")
legend("topleft",lty=1, text.width = , col=c("black","red", "blue", "green"), c("original","test", "fit.ets", "conf.interval"))
# This confidence interval of the forcasted values of state space moel well confined the
# test series. Further, the mean of the forecasted series is pretty looks liek the trend
# line of the test series. Thus we are conformed that the state space model is the best #model according to the forecast accuaracy.
################################################################################################################################################
###-- Two years ahead forecast using whole series ---------------------------
# While performing the diagnestic check of model and cross validation over the
# training and test set of the  series we have found that the space state model
# resonably fit well the series

## buildng model using whole series and two years ahead forecast
best.fit = ets(DJI.ts, model="MMN")
summary(best.fit) ## AIc == 6266.614, BIC == 6286.017, MASE == 0.2372
checkresiduals(best.fit)
plot(forecast(best.fit, h = 24), ylab="Monthly DJI Value ", plot.conf=FALSE,
    main="Two years DJI  Forecast by Space State Model", type="l", xlab="Time")
legend("topleft",lty=1, pch = 1, col=c("black","blue"),c("original","forecast"))

##############################################################################
```