Week 5 Assignment

ALY 6015 Intermediate Analytics

Submitted to:

Joseph Manseau

Date :03/22/2020

Submitted by:

Ashlesha Kshirsagar (001082234)

Time series Analysis

Introduction:

Time series arise as recordings of processes which vary over time. A recording can either be a continuous trace or a set of discrete observations. Time series analysis is a statistical technique to analyze the pattern of data points taken over time to forecast the future. Time series analysis has models such as Autoregressive (AR), Moving average (MA) and a combination of both these models ARIMA – Autoregressive Integrated Moving Average. Forecasts in a time series analysis are done based on seasonality, trends and changes. ("Time Series Analysis - an overview | ScienceDirect Topics", 2020). In this assignment we have used two dataset to perform and analye the time series.first data set is the micoreconmic data from tsdl package which is used to analyse thestructure of time series and factors affecting. For 2nd question we have used female birth count from Norfolk County, Boston from 1958 till date . We We have used Durbin Watson test to test the positive autocorrelation. After that we used **ARIMA Model, Holt Winters Method, Exponential Smoothening** for forecasting.And checked which model is the best fit for our data

Time series Analysis

Part A

Quarterly U.S. new plant/equipment expenditure data is extracted from the Macroeconomic data in tsdl. This data is a time series from 1964 to 1968 on new plant/equipment expenditure. Below is the summary statistics table of the extracted data.

Code

```
rm(list=ls())
while (!is.null(dev.list()))  dev.off
library(tsdl)
library(quantmod)
library(forecast)
library(lmtest)
tsdl
Microeconomic <- subset(tsdl,"Microeconomic")
str(Microeconomic)
Microeconomic[2]
```

Interpretation

In the below code that was used to set up and plot a time series dataset, frequency is taken as 4 because the expenditure data is quarterly. The frequency will change based on the data whether it is daily, weekly, monthly or quarterly.

Code :

```
# extract the 2nd microeconomic time series

Microeconomic[[2]]
# for viewing only
expenditure <- as.data.frame(Microeconomic[[2]])
# for Time series analysis
plantexp <- ts(Microeconomic[[2]], frequency = 4, start = c(1964, 1))
str(plantexp)
plot.ts(plantexp,ylab ="Expenditure in billions")
summary(plantexp)
```

Time series Analysis

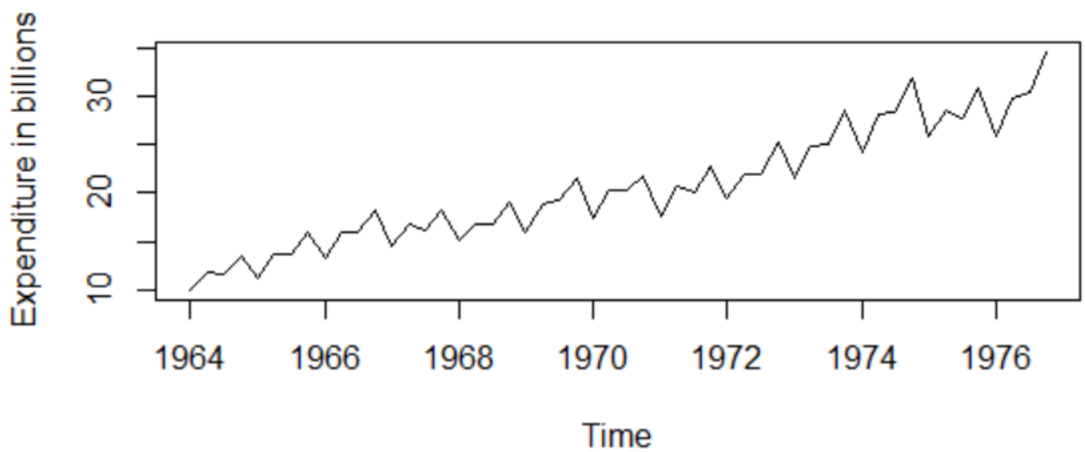

Figure 1 : Time series plot of Expenditure in billions

Interpretation:

From Graph we can see that there is a seasonality every quarter however there is an upward trend as well. In order to train a time series model, trend and seasonality should be excluded.

This is because trends can result into varying mean and seasonality will result in changing variance over time. The expenditure has also increased from 1964 which was about 10 billion to about 35 billion by 1976.

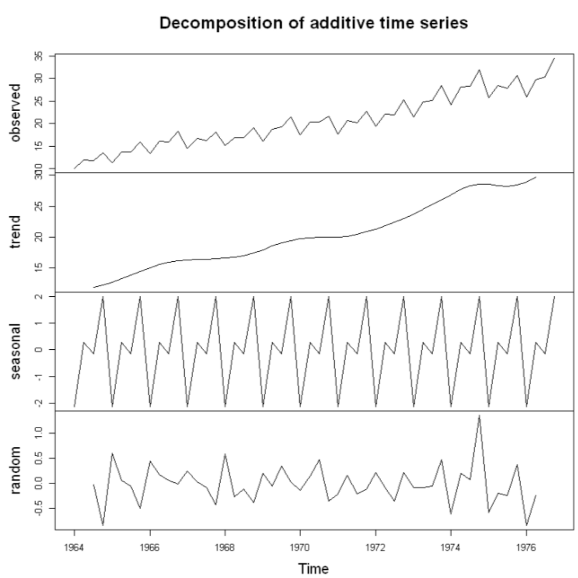

Figure 2: Decomposition of Time series graph

Code :

```
#decomposition of time series
sd <- sapply(plantexp, sd)
sd
mod2 <- decompose(plantexp)
plot(mod2)
```

Time series Analysis

Interpretation:

Decomposition is used to remove the seasonal effect from the time series. It is a mathematical procedure which transforms a time series into 3 different time series as seen in fig. Seasonal line shows pattern repeating with a fixed period of time. Trend line shows increase or decrease and lastly, random shows the noise in the data

Code

```
#removing seasonality
mod3 <-  plantexp - mod2$seasonal
plot((mod3))
plot(mod3, xlab = 'Years', ylab = 'Expenditure in billions', main = 'Expenditure in billions removed seasonality')
```

Output:

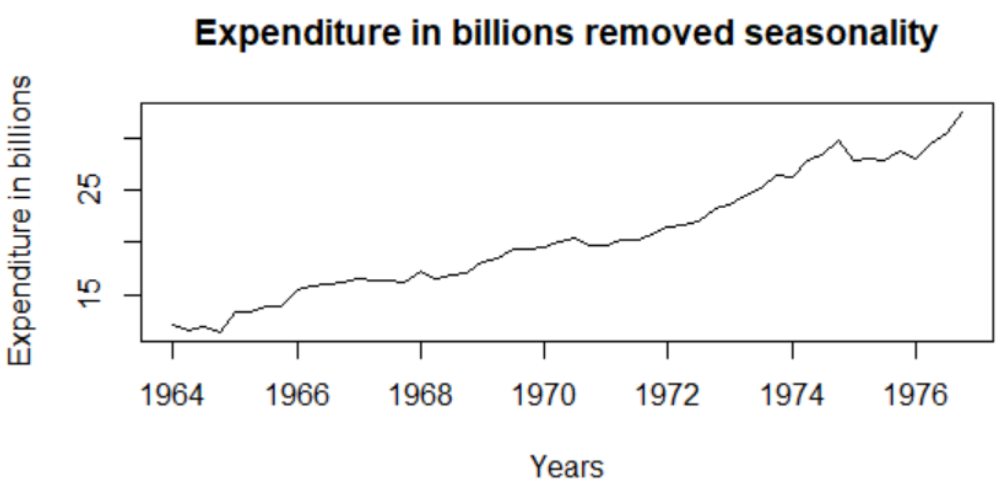

Figure 3: Time series graph of Expenditure in billions after removing seasonality

Interpretation:

This variance should be reduced in order to fit a good time series model. Now we will adjust seasonality in this model that means we are going to remove seasonality in the above model. As the seasonal component is removed. Now when we plot the graph, it shows we have trend and randomness.

Time series Analysis

Part B

In this part we have used Female births in Norfolk County, Boston from 1958 till date

Code:

```
library(forecast)
library(lmtest)
getwd()

setwd('C:\\Users\\ashle\\Desktop\\CPS\\Intermediate\\timeseries')
mydata <- read.csv('Daily Female Births.csv')  #part 2
head(mydata)
with(mydata, plot(Births ~ Date,xlab = "Date", ylab = "Female Births", main = "Date vs. Female Births"))
mod <- lm(mydata$Births ~ mydata$Date)
plot(mod$residuals ~ mod$fitted.values)
abline(0,0)
plot(mod, which = 1)
dwtest(mod)
```

Output

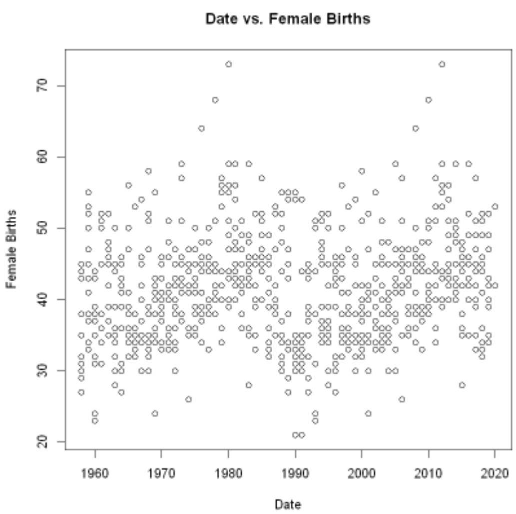

Fig 4: Plot of Date Vs Female Births

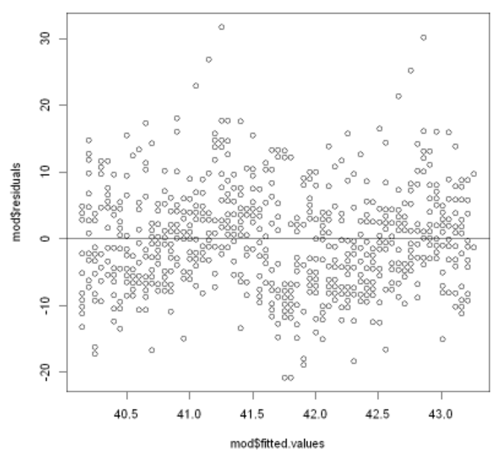

Fig 5 : Residual plot

Time series Analysis

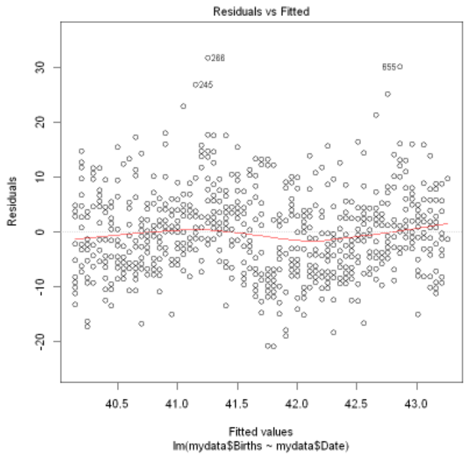

Fig 6: Residuals plot Vs Fitted values

From figure 4 and 5 we can say that we have constant variance in our residuals that means they are evenly spread above and below the center line and there is no visible trend in the data points. Also, from figure 6 we can see that constant variance and no trend, so we will not perform any transformations. We will move forward with this model.

```
        Durbin-Watson test

data:  mod
DW = 1.56, p-value = 7.465e-10
alternative hypothesis: true autocorrelation is greater than 0
```

Interpretation

We have used Durbin Watson test to test the positive autocorrelation. In this test, our null hypothesis that the error terms are not autocorrelated, and the alternative hypothesis is that the error terms have positive autocorrelation. When we test for positive autocorrelation, we will reject the null when our test stat is small. The P value obtained which is very less and we can reject our null hypothesis. And can move ahead with this data for forecasting.

Data Partition

Code

```
Qty_ts <- ts(data=mydata$Births, start=1958, end =2019, freq= 12)
#plotting time series
str(Qty_ts)
plot.ts(Qty_ts)
```

Time series Analysis

```
Qty_ts <- ts(data=mydata$Births, start=1958, end =2019, freq= 12)
#plotting time series
str(Qty_ts)
plot.ts(Qty_ts)

#partition
qty_train<-window(Qty_ts, start = 1958, c(2003,12))
qty_train
sum(qty_train)
qty_test<- window(Qty_ts, start = 2004 )
qty_test
sum(qty_test)
```

Interpretation:

Training data is used to estimate any parameters of a forecasting method and the test data is used to evaluate its accuracy. Because the test data is not used in determining the forecasts, it should provide a reliable indication of how well the model is likely to forecast on new data.

ARIMA Model

ARIMA models are defined for stationary time series. We use auto ARIMA function on both training and test dataset to plot automatic forecasting.

Code

```
#ARIMA model- Traning Dataset
autoArima_train <- auto.arima(qty_train)
plot(forecast(autoArima_train, h=12))
ArimaModel_train <- forecast(autoArima_train, h=12)
#check for accuracy
summary(ArimaModel_train)

#ARIMA model-test dataset
autoArima_test <- auto.arima(qty_test)
plot(forecast(autoArima_test, h=12))
ArimaModel_test <- forecast(autoArima_test, h=12)
#check for accuracy
summary(ArimaModel_test)
```

Output of Forecast model for Traning dataset

```
Forecast method: ARIMA(1,1,2)(1,0,0)[12]

Model Information:
Series: qty_train
ARIMA(1,1,2)(1,0,0)[12]

Coefficients:
         ar1      ma1     ma2     sar1
      0.3937  -1.2458  0.2759  -0.0472
s.e.  0.2656   0.2735  0.2591   0.0442

sigma^2 estimated as 52.96:  log likelihood=-1874.56
AIC=3759.11   AICc=3759.22   BIC=3780.67

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE
Training set 0.1246013 7.244044 5.799458 -2.819743 14.66316 0.7125614
                     ACF1
Training set -0.005659891

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2004       39.64959 30.32364 48.97554 25.38678 53.91240
Feb 2004       39.85546 30.42802 49.28289 25.43743 54.27348
Mar 2004       40.52100 31.05762 49.98438 26.04801 54.99399
Apr 2004       40.51306 31.03035 49.99577 26.01050 55.01561
May 2004       39.91541 30.41850 49.41232 25.39114 54.43968
Jun 2004       39.78000 30.27071 49.28930 25.23679 54.32322
Jul 2004       40.48981 30.96882 50.01080 25.92871 55.05091
Aug 2004       40.63223 31.09981 50.16465 26.05365 55.21081
Sep 2004       40.01951 30.47578 49.56325 25.42364 54.61539
Oct 2004       40.20830 30.65331 49.76329 25.59521 54.82140
Nov 2004       39.92540 30.35918 49.49162 25.29512 54.55567
Dec 2004       40.58567 31.00824 50.16310 25.93825 55.23309
```

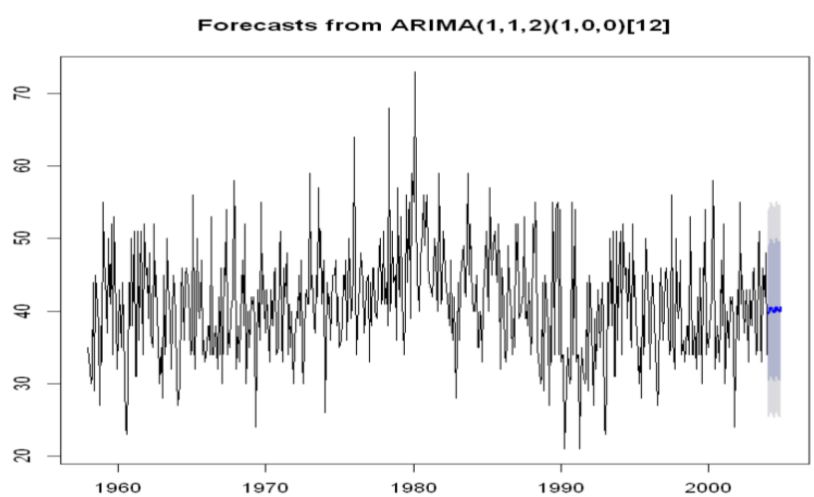

Figure 7: Forecast from ARIMA on Training Dataset

Output of Forecast model for Testing dataset

```
Forecast method: ARIMA(0,1,1)(0,0,1)[12]

Model Information:
Series: qty_test
ARIMA(0,1,1)(0,0,1)[12]

Coefficients:
          ma1     sma1
      -0.8786  -0.2170
s.e.   0.0445   0.0757

sigma^2 estimated as 46.27:  log likelihood=-600.59
AIC=1207.18   AICc=1207.32   BIC=1216.76

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE
Training set 0.2645837 6.745927 5.200891 -1.502399 11.95342 0.6699318
                   ACF1
Training set 0.07251973

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Feb 2019       42.98656 34.26876 51.70436 29.65383 56.31929
Mar 2019       40.94781 32.16600 49.72962 27.51719 54.37842
Apr 2019       42.18401 33.33866 51.02936 28.65621 55.71181
May 2019       44.89253 35.98408 53.80097 31.26824 58.51681
Jun 2019       42.18604 33.21495 51.15713 28.46594 55.90614
Jul 2019       43.49863 34.46533 52.53193 29.68339 57.31387
Aug 2019       44.19953 35.10444 53.29462 30.28979 58.10927
Sep 2019       44.74139 35.58493 53.89785 30.73780 58.74499
Oct 2019       44.37670 35.15927 53.59412 30.27987 58.47352
Nov 2019       41.36625 32.08827 50.64423 27.17680 55.55570
Dec 2019       42.53203 33.19388 51.87018 28.25056 56.81350
Jan 2020       42.78478 33.38685 52.18271 28.41188 57.15768
```

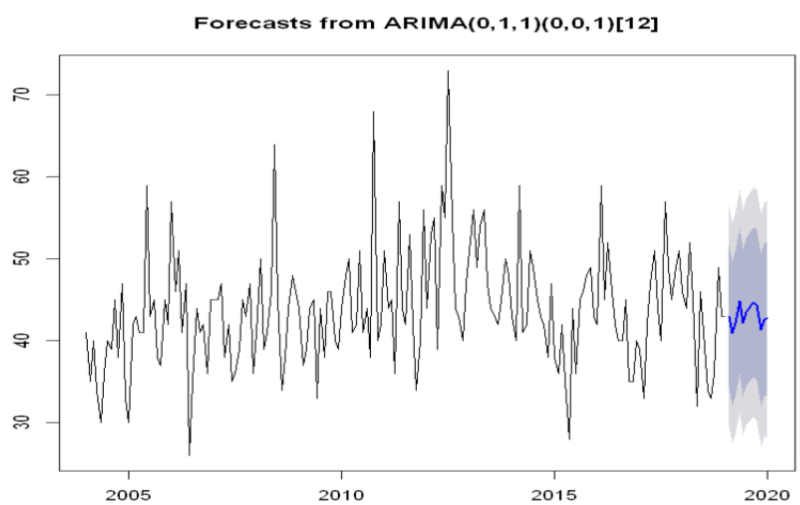

Figure 8 : Forecast From ARIMA on Testing dataset

Time series Analysis

Interpretation

A forecast "error" is the difference between an observed value and its forecast. Here "error" means the unpredictable part of an observation. residuals are calculated on the *training* set while forecast errors are calculated on the *test* set. Second, residuals are based on *one-step* forecasts while forecast errors can involve *multi-step* forecasts.

Holt Winters Method

The Holt-Winters forecasting algorithm allows users to smooth a time series and use that data to forecast areas of interest. This method is used to capture seasonality. The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level ℓtℓt, one for the trend btbt, and one for the seasonal component stst, with corresponding smoothing parameters αα, β∗β∗ and γ

Code

```
#forecast model using HoltWinters method for training data set
model2 <- hw(qty_train, initial='optimal', h=12 )
plot(model2)
accuracy(model2)
summary(model2)

#forecast model using HoltWinters method for test data set
model3 <- hw(qty_test, initial='optimal', h=12)
plot(model3)
accuracy(model3)
summary(model3)
```

Output

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.04603894	7.216054	5.802273	-3.198993	14.75029	0.7129073	0.09633826

```
Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = qty_train, h = 12, initial = "optimal")

  Smoothing parameters:
    alpha = 0.0663
    beta  = 1e-04
    gamma = 1e-04


  Initial states:
    l = 37.6847
    b = 0.0074
    s = 0.5911 -1.0372 3.4185 -1.4484 -0.1377 0.0271
           -0.1649 1.1648 -1.9925 -0.8144 0.4608 -0.0672

  sigma:  7.323

     AIC     AICc      BIC
5700.923 5702.069 5774.253

Error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE
Training set -0.04603894 7.216054 5.802273 -3.198993 14.75029 0.7129073
                   ACF1
Training set 0.09633826
```

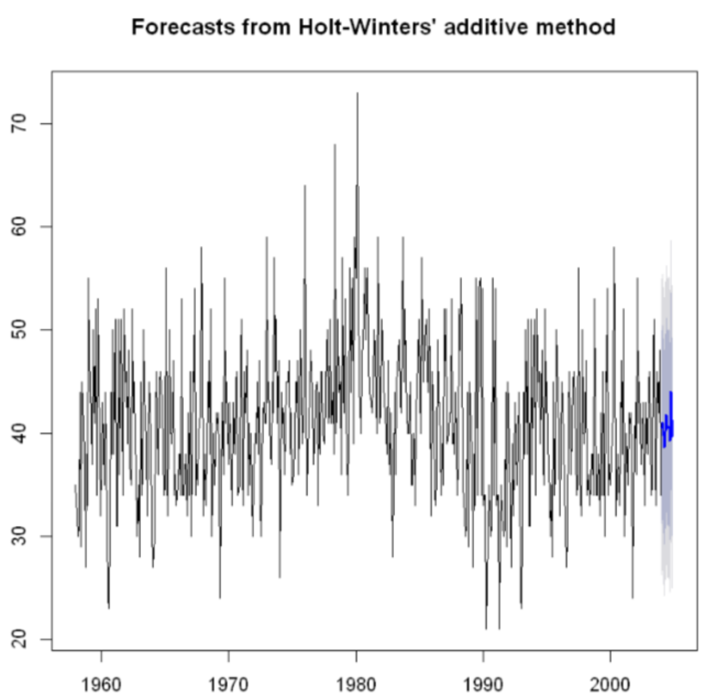

Figure 9: Forecasts from Holts Winters Additive method on training dataset

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.04885039	6.796487	5.193546	-2.295035	12.1118	0.6689857	0.1143723

```
Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = qty_test, h = 12, initial = "optimal")

  Smoothing parameters:
    alpha = 0.1138
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 38.1396
    b = 0.0289
    s = -0.5639 -0.8738 -0.5684 -1.2671 0.1436 -0.426
           1.5623 1.1869 -1.7124 2.2684 0.6527 -0.4022

  sigma:  7.1184

     AIC     AICc      BIC
1668.667 1672.422 1723.041

Error measures:
                      ME     RMSE      MAE       MPE    MAPE      MASE
Training set -0.04885039 6.796487 5.193546 -2.295035 12.1118 0.6689857
                  ACF1
Training set 0.1143723

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Feb 2019       43.49475 34.37216 52.61733 29.54296 57.44653
Mar 2019       45.13890 35.95732 54.32047 31.09688 59.18091
Apr 2019       41.18504 31.94474 50.42533 27.05322 55.31685
May 2019       44.11149 34.81274 53.41024 29.89028 58.33270
Jun 2019       44.51565 35.15871 53.87258 30.20545 58.82584
Jul 2019       42.55551 33.14065 51.97038 28.15672 56.95430
Aug 2019       43.15280 33.68026 52.62533 28.66580 57.63979
Sep 2019       41.77023 32.24027 51.30019 27.19541 56.34505
Oct 2019       42.49694 32.90980 52.08409 27.83467 57.15922
Nov 2019       42.22070 32.57661 51.86478 27.47134 56.97006
Dec 2019       42.55841 32.85761 52.25920 27.72232 57.39449
Jan 2020       42.74808 32.99081 52.50535 27.82562 57.67054
```

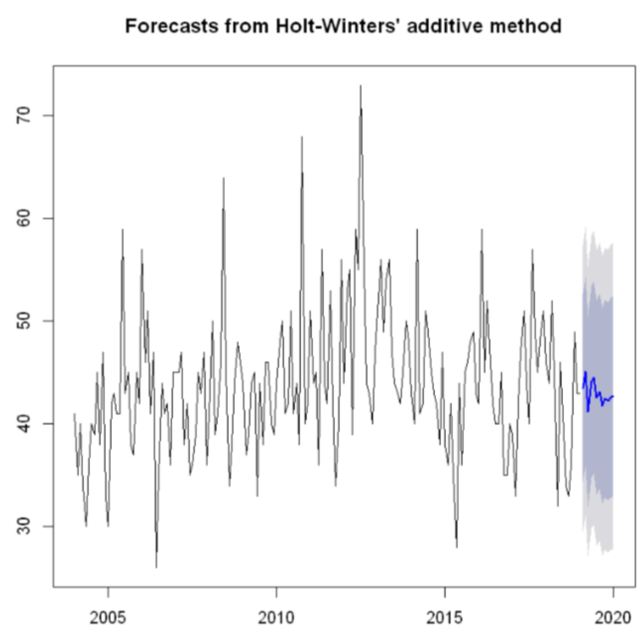

Figure 10: Forecasts from Holts Winters Additive method on Testing dataset

Interpretation

Both the forecast (figure 9 and 10) shows that the seasonal variation in the data increases as the level of the series increases. The RMSE value helps to decide the accuracy of the model which is 6.79.

Exponential Smoothening

Code

```
#Exponential Smoothening training
ets_train <- ets(qty_train)
ets_train
#Forecast for ets component training
fcast_ets_train <- forecast(ets_train, h = 12)
plot(fcast_ets_train)
summary(fcast_ets_train)
#Accuracy
accuracy(fcast_ets_train)

#Exponential Smoothening test
ets_test <- ets(qty_test)
ets_test
#Forecast for ets component
fcast_ets_test <- forecast(ets_test, h = 12)
plot(fcast_ets_test)
summary(fcast_ets_test)
#Accuracy
accuracy(fcast_ets_tes)
```

Output

Time series Analysis

```
ETS(A,N,N)

Call:
 ets(y = qty_train)

  Smoothing parameters:
    alpha = 0.0532

  Initial states:
    l = 38.6468

  sigma:  7.3106

     AIC     AICc      BIC
5685.297 5685.341 5698.238
```

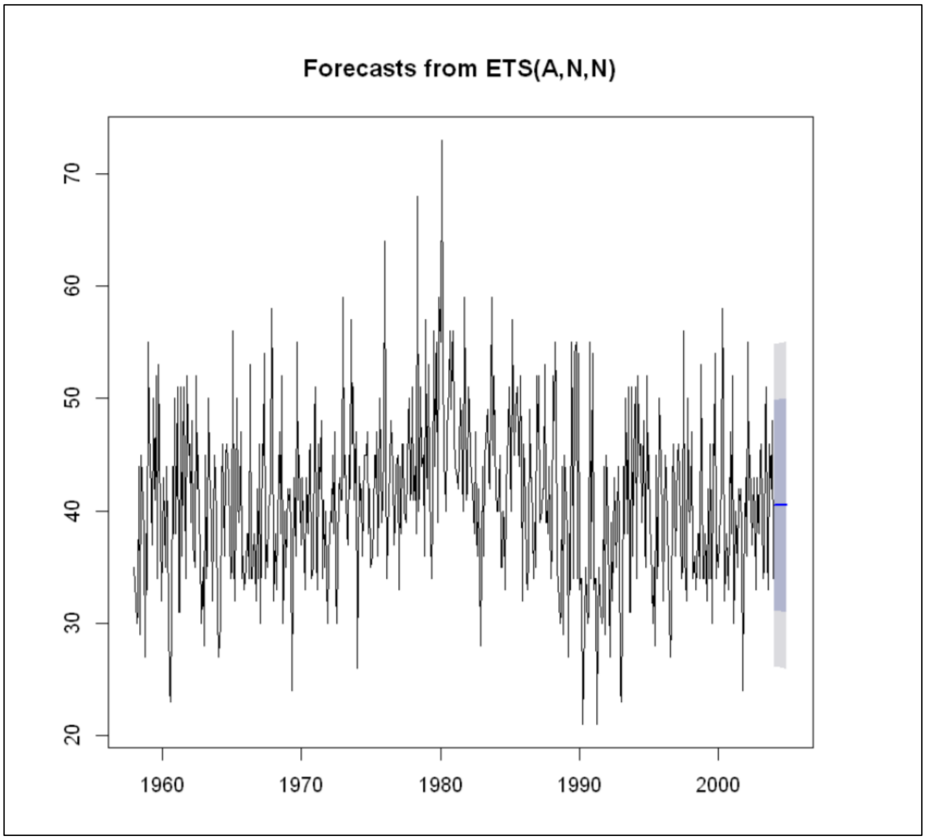

Figure 11: Forecasts from ETS on training dataset

```
Forecast method: ETS(A,N,N)

Model Information:
ETS(A,N,N)

Call:
 ets(y = qty_train)

  Smoothing parameters:
    alpha = 0.0532

  Initial states:
    l = 38.6468

  sigma:  7.3106

     AIC     AICc      BIC
5685.297 5685.341 5698.238

Error measures:
                     ME    RMSE      MAE       MPE     MAPE      MASE
Training set 0.06261119 7.29739 5.870165 -3.046052 14.90038 0.7212489
                   ACF1
Training set 0.09760617

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2004       40.48639 31.11742 49.85536 26.15779 54.81499
Feb 2004       40.48639 31.10416 49.86862 26.13751 54.83528
Mar 2004       40.48639 31.09092 49.88187 26.11725 54.85553
Apr 2004       40.48639 31.07769 49.89509 26.09703 54.87576
May 2004       40.48639 31.06449 49.90830 26.07683 54.89595
Jun 2004       40.48639 31.05130 49.92149 26.05666 54.91612
Jul 2004       40.48639 31.03813 49.93466 26.03652 54.93626
Aug 2004       40.48639 31.02498 49.94781 26.01641 54.95638
Sep 2004       40.48639 31.01184 49.96094 25.99632 54.97646
Oct 2004       40.48639 30.99873 49.97405 25.97627 54.99652
Nov 2004       40.48639 30.98563 49.98715 25.95624 55.01655
Dec 2004       40.48639 30.97255 50.00023 25.93623 55.03655
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.06261119	7.29739	5.870165	-3.046052	14.90038	0.7212489	0.09760617

Time series Analysis

```
ETS(M,N,N)

Call:
 ets(y = qty_test)

  Smoothing parameters:
    alpha = 0.1217

  Initial states:
    l = 38.1904

  sigma:  0.159

     AIC     AICc      BIC
1645.249 1645.384 1654.844
```

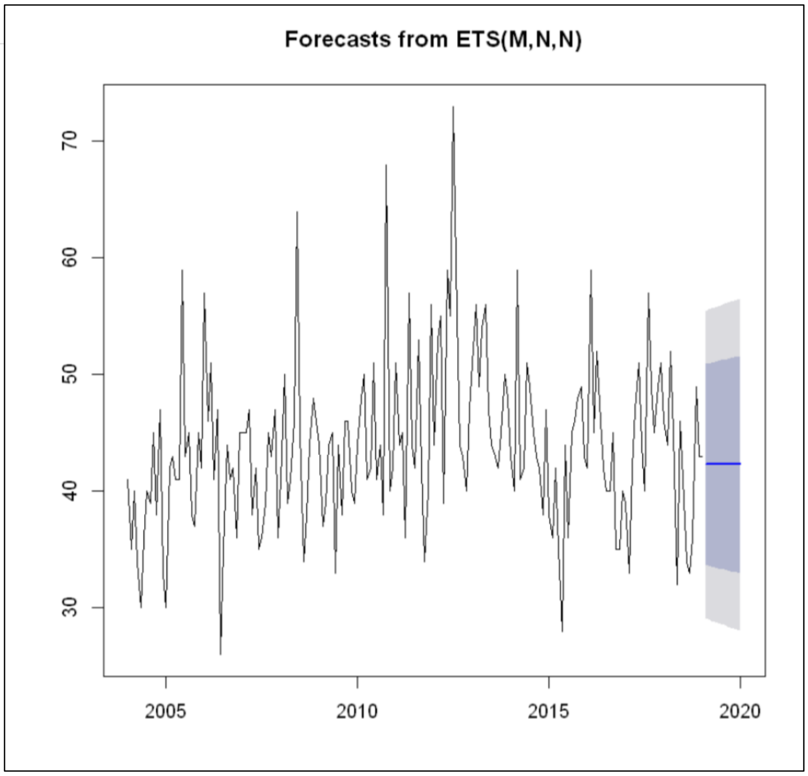

Figure 12: Forecasts from ETS on testing dataset

```
Forecast method: ETS(M,N,N)

Model Information:
ETS(M,N,N)

Call:
 ets(y = qty_test)

  Smoothing parameters:
    alpha = 0.1217

  Initial states:
    l = 38.1904

  sigma:  0.159

     AIC     AICc      BIC
1645.249 1645.384 1654.844

Error measures:
                    ME     RMSE     MAE       MPE     MAPE      MASE       ACF1
Training set 0.1861758 6.902022 5.29185 -1.799537 12.24971 0.6816483 0.09858833

Forecasts:
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Feb 2019       42.29014 33.67083 50.90945 29.10805 55.47223
Mar 2019       42.29014 33.60568 50.97460 29.00840 55.57188
Apr 2019       42.29014 33.54098 51.03930 28.90946 55.67082
May 2019       42.29014 33.47674 51.10354 28.81121 55.76907
Jun 2019       42.29014 33.41294 51.16734 28.71363 55.86665
Jul 2019       42.29014 33.34956 51.23071 28.61671 55.96357
Aug 2019       42.29014 33.28662 51.29366 28.52044 56.05984
Sep 2019       42.29014 33.22408 51.35620 28.42480 56.15548
Oct 2019       42.29014 33.16195 51.41833 28.32978 56.25050
Nov 2019       42.29014 33.10022 51.48006 28.23537 56.34491
Dec 2019       42.29014 33.03888 51.54140 28.14155 56.43872
Jan 2020       42.29014 32.97791 51.60236 28.04832 56.53196
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.1861758	6.902022	5.29185	-1.799537	12.24971	0.6816483	0.09858833

Conclusion:

A forecast method that minimizes the MAE will lead to forecasts of the median, while minimizing the RMSE will lead to forecasts of the mean. Consequently, the RMSE is also widely used, despite being more difficult to interpret. After comparing different models of forecasting we can say that based on RMSE value, the **ARIMA** model is the best fit model for the female birth

Reference:

1. Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, *4*(1), 1–28. https://doi.org/10.1002/for.3980040103

2. Time Series Analysis - an overview | ScienceDirect Topics. (2020). Retrieved 14 February 2020, from https://www.sciencedirect.com/topics/medicine-and-dentistry/time-series-analysis

3.Autoregressive Integrated Moving Average Models (ARIMA).(2020). Retrieved 14 February 2020, from http://forecastingsolutions.com/arima.html