**TIME SERIES ANALYSIS**

**ARVIND PAWAR**

**AZAR NAJAFLI**

**DHRUVIN KOTHARI**

**NORTHEASTERN UNIVERSITY**

**ALY6015 INTERMEDIATE ANALYTICS**

**INSTRUCTOR NAME- TENGLONG LI**

**DATE- 03/24/2019**



**ANALYSIS**

**PART- A**

**1) Make use of birth dataset and plot a timeseries keeping frequency=12 and start=c(1946,1)**

**births <- scan("https://robjhyndman.com/tsdldata/data/nybirths.dat", skip=0)**

**Now decompose the timeseries and plot the components.**

**Solution:**

> library(ggplot2)

> library(TTR)

> library(forecast)

> library(tseries)

‘tseries’ version: 0.10-46

‘tseries’ is a package for time series analysis and computational

finance.

See ‘library(help="tseries")’ for details.

> births\_data<- scan("https://robjhyndman.com/tsdldata/data/nybirths.dat", skip=0)

Read 168 items

> births\_times <- ts(births\_data, frequency = 12, start = c(1946,1))

> births\_times

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov

1946 26.663 23.598 26.931 24.740 25.806 24.364 24.477 23.901 23.175 23.227 21.672

1947 21.439 21.089 23.709 21.669 21.752 20.761 23.479 23.824 23.105 23.110 21.759

1948 21.937 20.035 23.590 21.672 22.222 22.123 23.950 23.504 22.238 23.142 21.059

1949 21.548 20.000 22.424 20.615 21.761 22.874 24.104 23.748 23.262 22.907 21.519

1950 22.604 20.894 24.677 23.673 25.320 23.583 24.671 24.454 24.122 24.252 22.084

1951 23.287 23.049 25.076 24.037 24.430 24.667 26.451 25.618 25.014 25.110 22.964

1952 23.798 22.270 24.775 22.646 23.988 24.737 26.276 25.816 25.210 25.199 23.162

1953 24.364 22.644 25.565 24.062 25.431 24.635 27.009 26.606 26.268 26.462 25.246

1954 24.657 23.304 26.982 26.199 27.210 26.122 26.706 26.878 26.152 26.379 24.712

1955 24.990 24.239 26.721 23.475 24.767 26.219 28.361 28.599 27.914 27.784 25.693

1956 26.217 24.218 27.914 26.975 28.527 27.139 28.982 28.169 28.056 29.136 26.291

1957 26.589 24.848 27.543 26.896 28.878 27.390 28.065 28.141 29.048 28.484 26.634

1958 27.132 24.924 28.963 26.589 27.931 28.009 29.229 28.759 28.405 27.945 25.912

1959 26.076 25.286 27.660 25.951 26.398 25.565 28.865 30.000 29.261 29.012 26.992

Dec

1946 21.870

1947 22.073

1948 21.573

1949 22.025

1950 22.991

1951 23.981

1952 24.707

1953 25.180

1954 25.688

1955 26.881

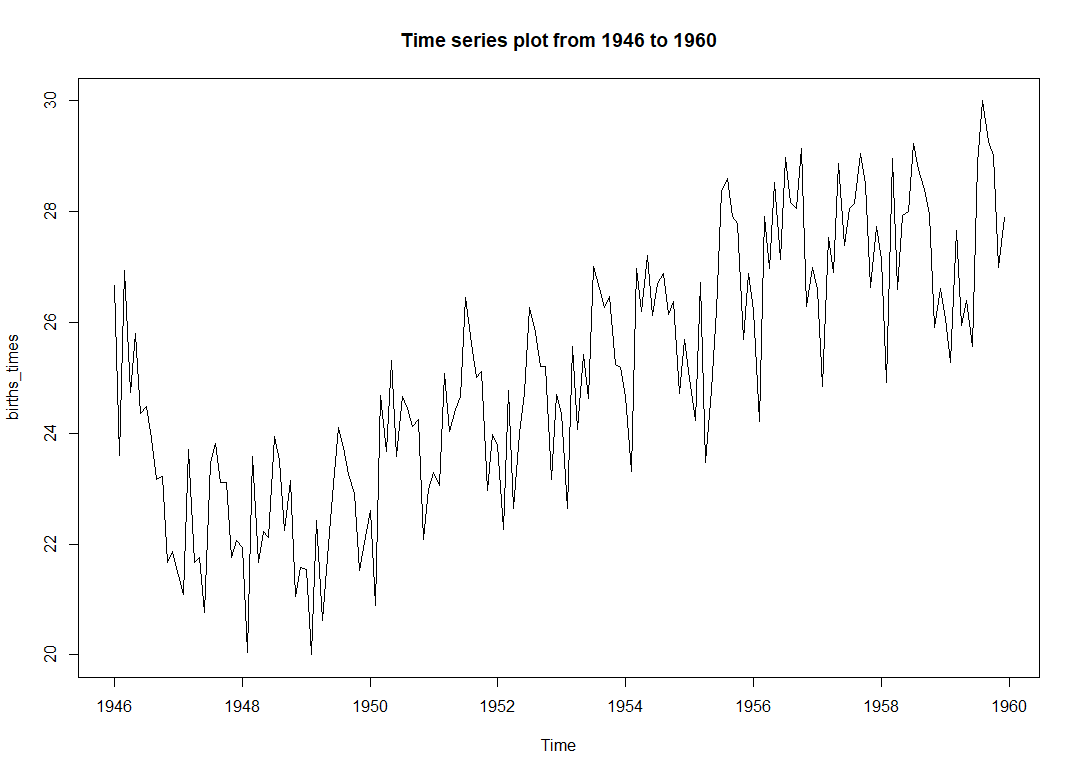
1956 26.987

1957 27.735

1958 26.619

1959 27.897

> plot(births\_times, main="Time series plot from 1946 to 1960")



First, we have installed the required libraries for the time series analysis such as TTR, forecast, series, ggplot2. We retrieved the data from given web link and loaded the dataset into births\_data. The variable births\_data is then given as input to Time Series function ts and stored in births\_times with frequency of12 and series year starting from 1946 to 1960. Using the time series function and variable births\_times we have an output for 12 months from 1946-1960. We have also used plot function for plotting the time series of births\_times shown above.

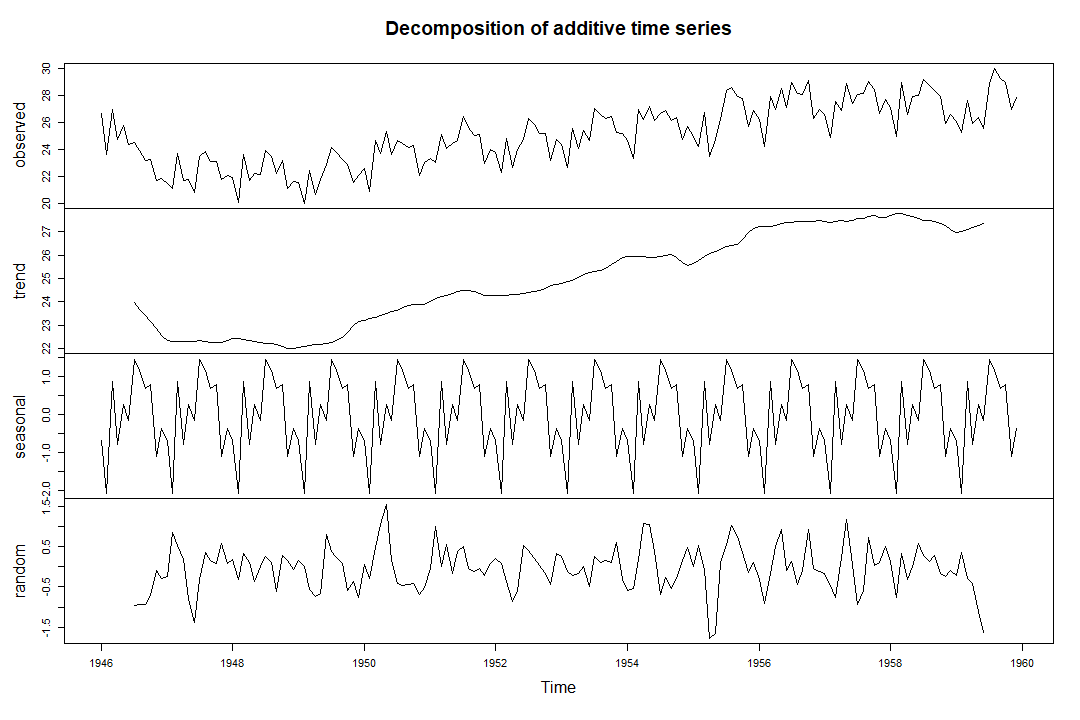
**2) Is the time series additive or a multiplicative model?**

**Remove the seasonality so that the time series would contain only the trend component and an irregular component.**

**Solution:**

> components<- decompose(births\_times)

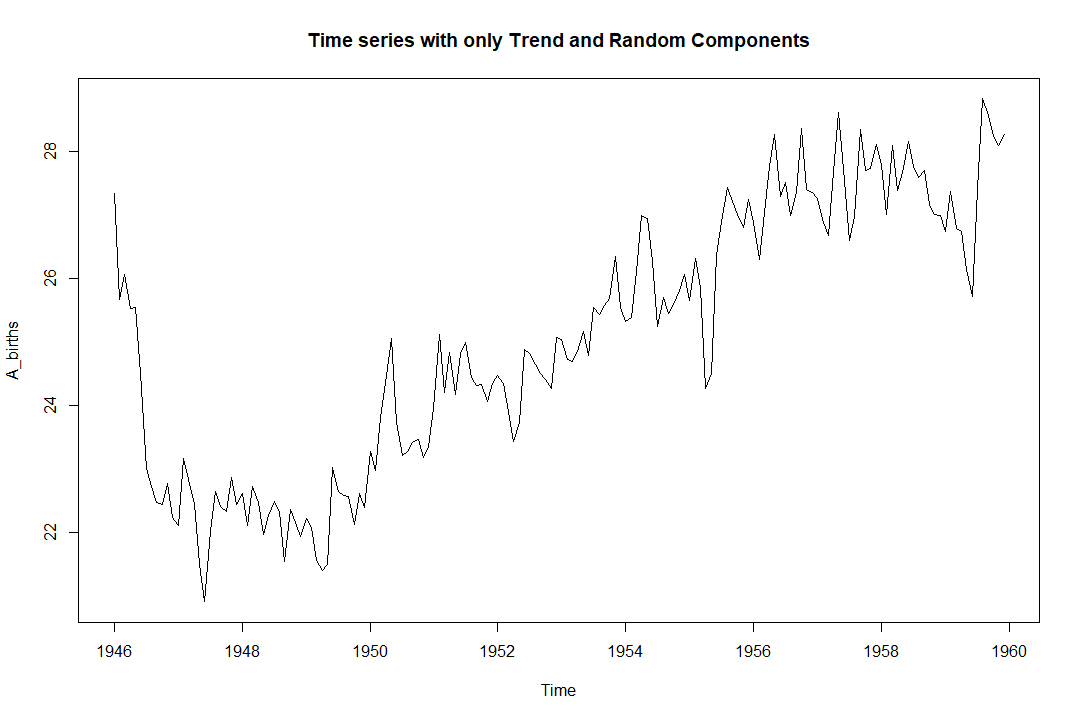
> plot(components)



Here, we used decompose() function, as the data is seasonal we can observe the different components such as trends, seasonal and random. We have created a variable “components” to decompose the births dataset. Viewing them graphically using plot function.

> A\_births<- births\_data-components$seasonal

> plot(A\_births, main="Time series with only Trend and Random Components")



**PART-B**

**1) Make use of birth dataset and plot a timeseries keeping start=c(1500) volcanodust < scan("http://robjhyndman.com/tsdldata/annual/dvi.dat", skip=1). Plot ACF and PACF and add comment about it**

> volcanodust <- scan("http://robjhyndman.com/tsdldata/annual/dvi.dat", skip=1)

Read 470 items

> volcanodust\_times<-ts(volcanodust, start = c(1500))

> volcanodust\_times

Time Series:

Start = 1500

End = 1969

Frequency = 1

[1] 200 150 100 50 0 0 0 0 0 0 0

[12] 0 0 0 0 0 0 0 0 0 0 0

[23] 0 0 0 0 0 0 0 0 0 0 0

[34] 0 0 50 50 50 0 0 0 0 0 0

[45] 0 0 0 0 0 0 0 0 0 100 500

[56] 350 200 100 0 0 0 0 0 0 0 0

[67] 0 0 0 0 0 0 0 0 0 0 0

[78] 0 0 0 0 0 0 0 0 0 200 150

[89] 100 50 0 0 0 200 150 100 50 40 30

[100] 20 10 400 300 210 110 10 20 50 50 50

[111] 40 30 20 10 200 150 100 50 0 0 0

[122] 0 0 0 0 100 75 50 25 0 0 120

[133] 90 60 30 0 40 30 120 85 150 400 275

[144] 175 75 0 60 45 30 15 100 75 50 25

[155] 0 0 0 0 0 0 340 255 170 85 130

[166] 100 65 30 0 0 0 0 200 150 100 50

[177] 0 0 0 0 280 210 140 70 0 0 0

[188] 0 0 0 0 0 0 140 285 205 105 45

[199] 0 0 0 0 0 0 0 0 0 300 225

[210] 150 75 0 80 60 40 20 0 120 90 60

[221] 30 100 75 50 55 15 15 15 15 15 160

[232] 130 90 50 0 0 0 0 0 0 0 0

[243] 0 0 60 45 30 15 0 0 0 0 200

[254] 150 160 255 150 95 40 80 110 77 45 13

[265] 0 0 0 0 0 0 0 0 50 37 25

[276] 13 0 0 0 180 135 90 45 400 300 200

[287] 160 45 30 15 0 0 0 0 0 120 130

[298] 90 50 130 90 60 30 0 0 0 0 0

[309] 0 0 0 80 180 170 170 695 490 375 195

[320] 30 15 0 200 150 100 70 80 65 50 75

[331] 50 200 130 80 40 525 450 375 300 225 150

[342] 75 0 0 0 100 205 140 90 30 0 0

[353] 0 0 0 0 140 105 70 35 0 160 120

[364] 80 40 0 0 0 160 120 80 40 0 0

[375] 0 120 90 60 30 0 0 0 0 400 300

[386] 240 170 50 170 125 85 45 20 15 10 5

[397] 0 0 30 25 15 5 180 135 90 45 0

[408] 60 45 30 15 0 60 45 30 15 0 0

[419] 0 0 0 0 0 0 0 0 0 0 0

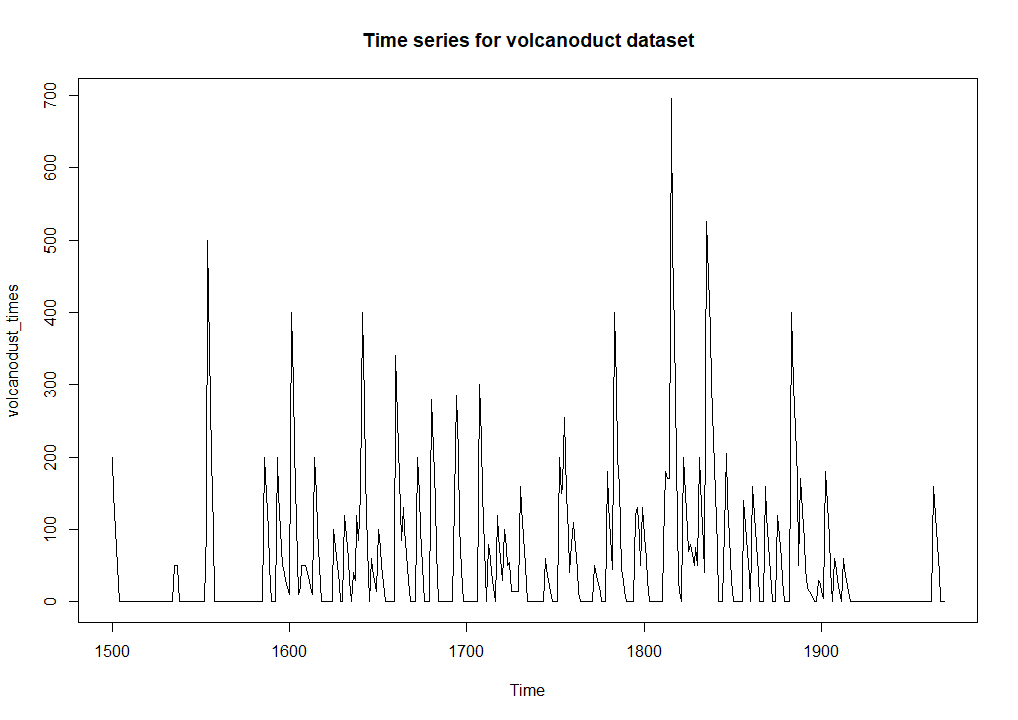
[430] 0 0 0 0 0 0 0 0 0 0 0

[441] 0 0 0 0 0 0 0 0 0 0 0

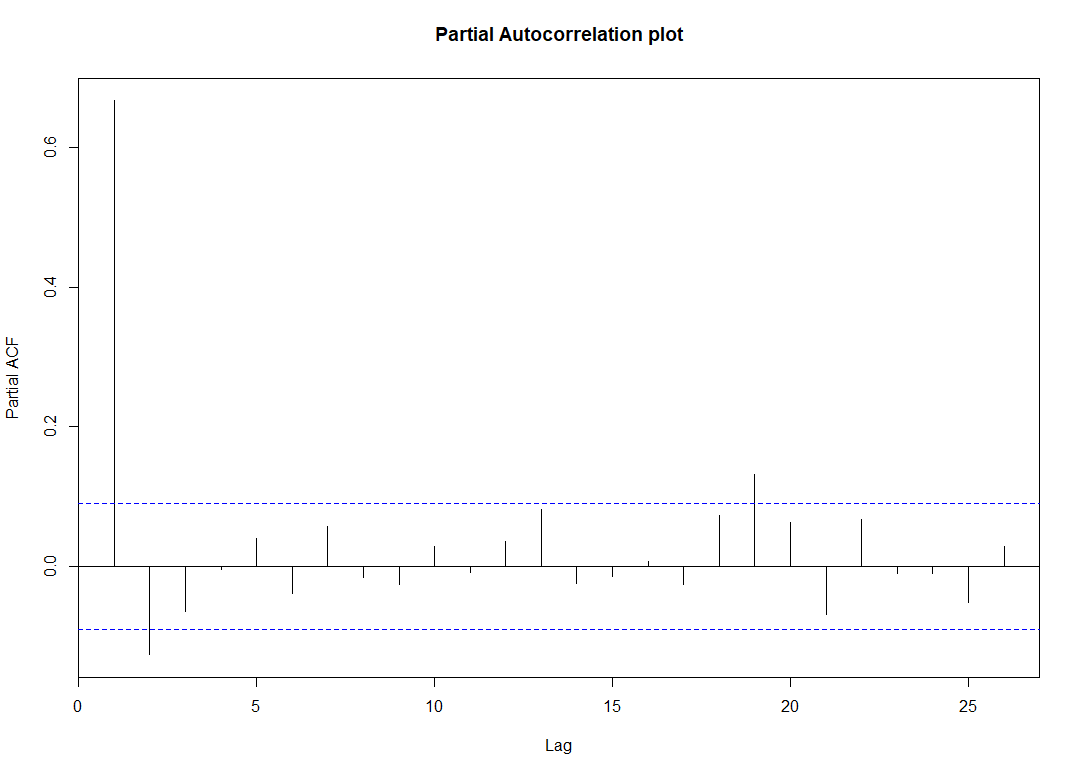
[452] 0 0 0 0 0 0 0 0 0 0 0

[463] 0 160 120 80 40 0 0 0

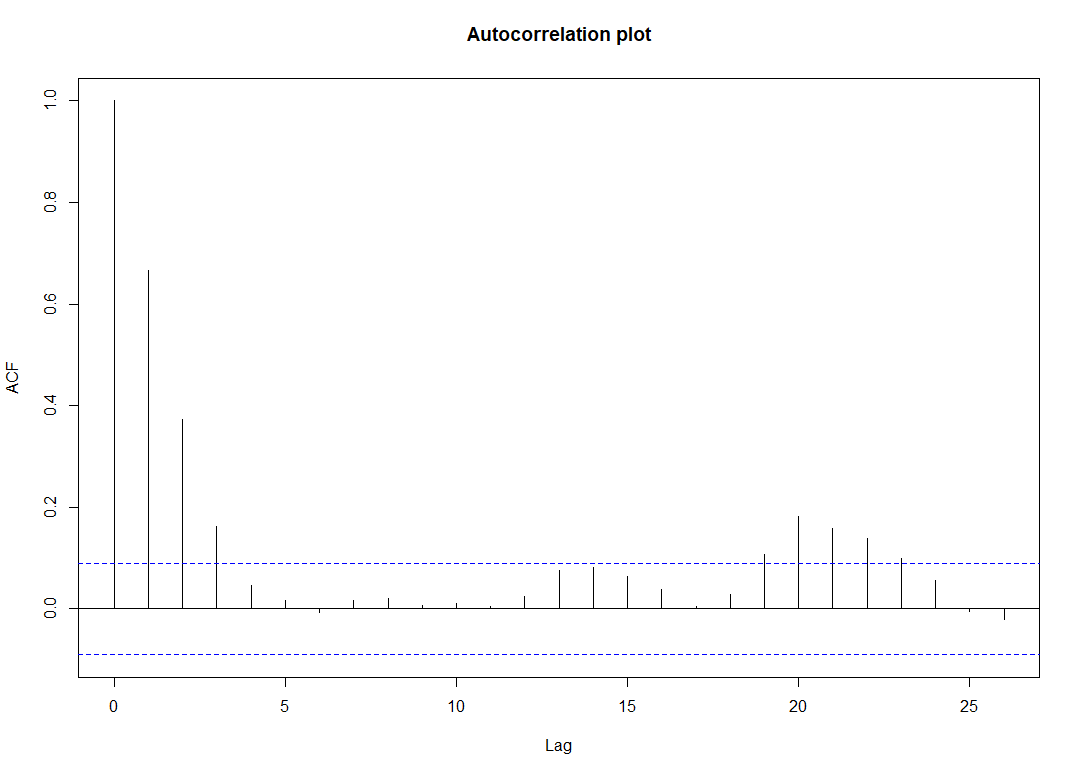
> plot(volcanodust\_times, main="Time series for volcanoduct dataset")



> pacf(volcanodust\_times, main="Partial Autocorrelation plot")



> acf(volcanodust\_times, main="Autocorrelation plot")



In the above steps, we have plotted ACF and PACF from which we have determined p and q values. In PACF plot we can see that the first line is cutting the blue line from the baseline, the line number that cuts the blue line is considered as the p-value. Similarly, in ACF the third line from baseline is cutting the blue line.

**2) Fit an ARIMA model and make use of appropriate p, d, q values.**

ARIMA model is a class of statistical models for analysis and forecasting time series data. This is the model that uses the dependent relationship between an observation and some number of logged observations. Here we used arima function to build the ARIMA model with p, d, q values as 1, 1, 3 respectively. We then stored the arima model variable and has resulted aic values as 5326.62 which is best value achieved with q, d, p values.

> Arima\_Model<- arima(volcanodust\_times, order = c(1,1,3))

> Arima\_Model

Call:

arima(x = volcanodust\_times, order = c(1, 1, 3))

Coefficients:

ar1 ma1 ma2 ma3

0.4323 -0.6863 -0.1496 -0.1383

s.e. 0.1049 0.1062 0.0660 0.0744

sigma^2 estimated as 4884: log likelihood = -2658.31, aic = 5326.62

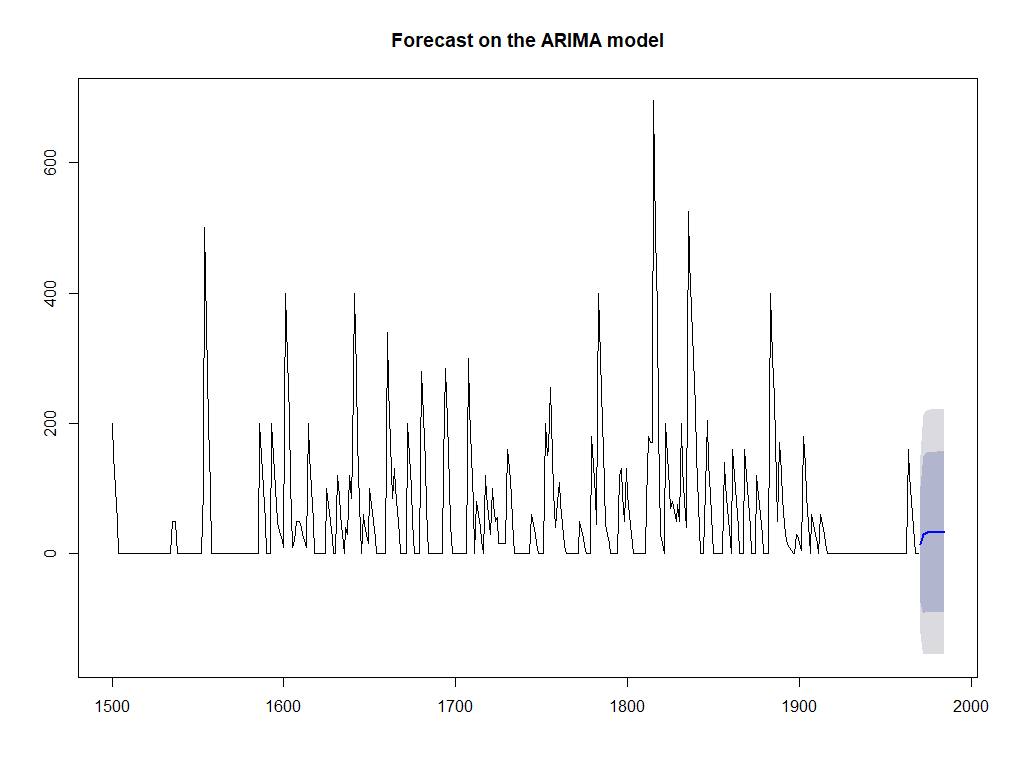
**3) Perform forecast on the model and plot the forecast**

**Solution:**

**In the below code we have passed ARIMA model as input to the forecast to get the forecast of the volcanoduct dataset. We have graphically forecasted the forecast model.**

> Forcast<- forecast(Arima\_Model, h=15)

> plot(Forcast, main="Forecast on the ARIMA model")



REFERENCES

* Time Series Analysis With R. (2019). R-statistics.co. Retrieved 25 March 2019, from <http://r-statistics.co/Time-Series-Analysis-With-R.html>
* v0.23-4, T. (2019). TTR package | R Documentation. Rdocumentation.org. Retrieved 25 March 2019, from <https://www.rdocumentation.org/packages/TTR/versions/0.23-4>
* Quick-R: Time Series. (2019). Statmethods.net. Retrieved 25 March 2019, from <https://www.statmethods.net/advstats/timeseries.html>
* decompose function | R Documentation. (2019). Rdocumentation.org. Retrieved 25 March 2019, from <https://www.rdocumentation.org/packages/stats/versions/3.5.3/topics/decompose>
* Arima function | R Documentation. (2019). Rdocumentation.org. Retrieved 25 March 2019, from <https://www.rdocumentation.org/packages/forecast/versions/8.5/topics/Arima>