Fitting an ARMA Model and Performing Residual Analysis

Practical 9
Ananya Kaushal
1940233

INTRODUCTION:

ARMA model is a model that is combined from the AR and MA models. In this model, the impact of previous lags along with the residuals is considered for forecasting the future values of the time series. Here β represents the coefficients of the AR model and α represents the coefficients of the MA model.

$$Yt = \beta_1 * y_{t-1} + \alpha_1 * \varepsilon_{t-1} + \beta_2 * y_{t-2} + \alpha_2 * \varepsilon_{t-2} + \beta_3 * y_{t-3} + \alpha_3 * \varepsilon_{t-3} + \ldots\ldots\ldots\ldots + \beta_k * y_{t-k} + \alpha_k * \varepsilon_{t-k}$$

An ARMA model, or Autoregressive Moving Average model, is used to describe weakly stationary stochastic time series in terms of two polynomials.

The "residuals" in a time series model are what is left over after fitting a model. The residuals are equal to the difference between the observations and the corresponding fitted values:

$$E_t = y_t - \grave{y}_t.$$

In this report, we use the number of worldwide earthquakes during different years to fit a suitable ARMA model and perform residual analysis.

AIM:

To fit a suitable ARMA model describing number of worldwide earthquakes during different years and verify the properties of residual analysis.

OBJECTIVE:

(a) Fit a suitable model to the yearly earthquakes data using auto.arima.
(b) Perform residual analysis on the fitted model and check whether the assumptions of residuals are satisfied

DATASET:

The dataset used in this study gives the information about the number of occurrences of earthquakes per year, from the year 1916 to 2015.

PROCEDURE:

```
#Importing the dataset:


library(readxl)
earthquakes_data <- read_excel("Quakes.xlsx")
View(earthquakes_data)

#Preview of the dataset:

head(earthquakes_data,10)

## # A tibble: 10 x 2
##     Year Quakes
##    <dbl>  <dbl>
##  1  1916      2
##  2  1917      5
##  3  1918     12
##  4  1919      8
##  5  1920      7
##  6  1921      9
##  7  1922      7
##  8  1923     12
##  9  1924      9
## 10  1925     12
```

#About the dataset:

#The above data shows the number of worldwide earthquakes with magnitude greater than 7 on the Richter scale for a total of 100 years taken from http://earthquake.usgs.gov/. Here zt is the number of earthquakes for n=100 years. The two variables in the data are:
#Year: describes the year in which the earthquakes were recorded
#Quakes: Number of worldwide earthquakes recorded (greater than 7 on Richter Scale)

Analysis:

#1. Converting the data into a time series object:

```
attach(earthquakes_data)
data = ts(Quakes, start = 1916)
```

#2. Plotting the time series graph:

```
ts.plot(data, xlab = "Time", ylab = "Number of Earthquakes")
```

Trend Component - In the above plot, the trend component is very slightly visible as the from the initial value to the final value a miniscule increasing trend can be seen.
Seasonal Component - The time series does not show a particular pattern in the number of earthquakes recorded yearly.
Cyclical Component - Since no periodic cycle in pattern is visible from the above plot, we can say that

there is no cyclical component present in the time series data.
Therefore, since the data does not explicitly contain any of the components, we can describe the model as:
zt = et
where, et = irregular component

```
#3. We perform the Augmented Dickey-Fuller test to check whether the data is stationary:

#H0: The data is non stationary
#H1: The data is stationary

library(tseries)

## Warning: package 'tseries' was built under R version 4.1.2

## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo
```

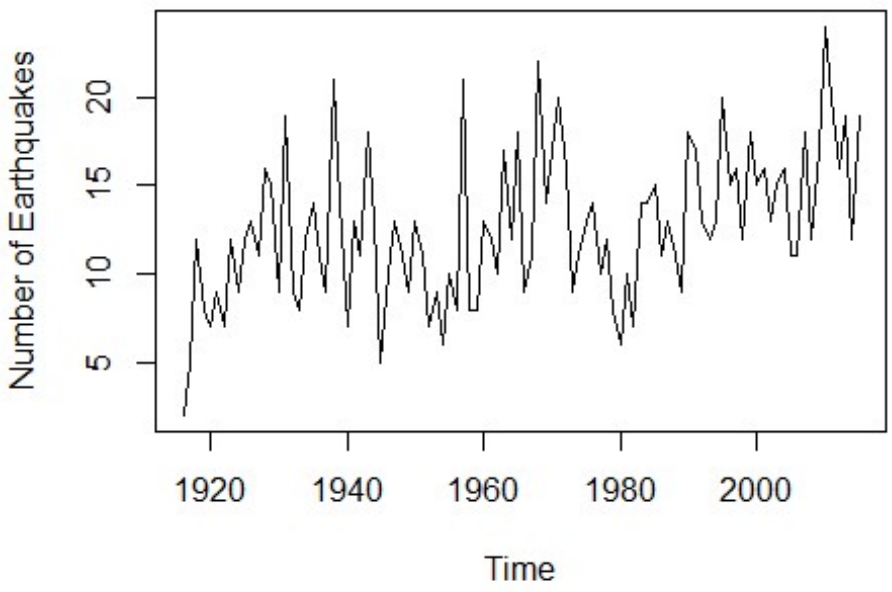

Figure 1: Time series plot of Earth Quakes Dataset

```
adf.test(data)

##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -3.452, Lag order = 4, p-value = 0.04991
## alternative hypothesis: stationary
```

```
#Since the p-value is less than 0.05 we reject the null hypothesis and conclude that the data is stationary

#4. Examining the ACF and PACF plots:

library(forecast)

## Warning: package 'forecast' was built under R version 4.1.2

acf(data)
```

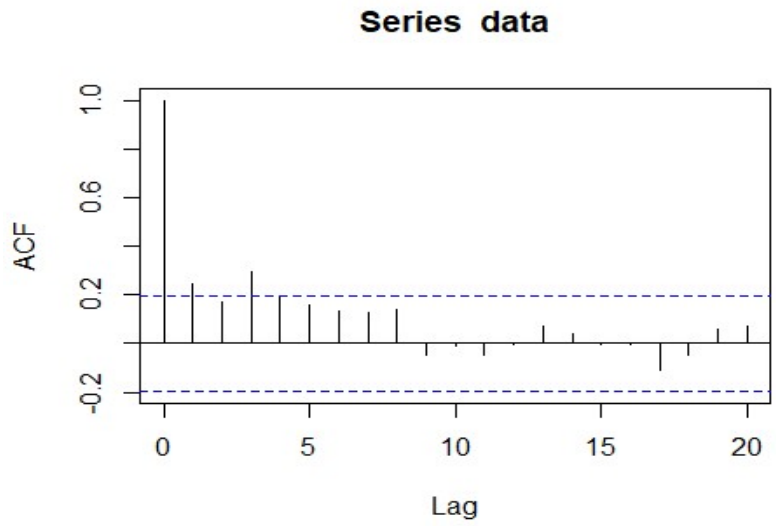

Figure 2: ACF Plot of the Dataset

```
pacf(data)
```

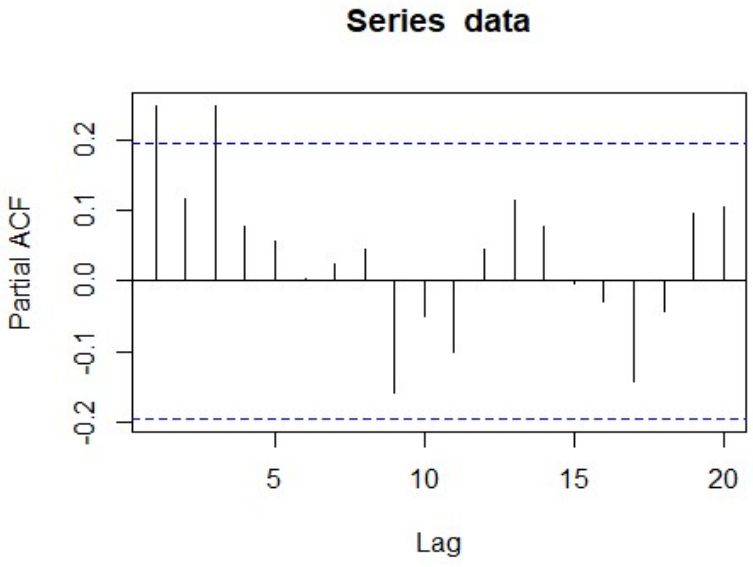

Figure 3: PACF Plot of the dataset

We know that in a moving average model of order q, the PACF plot will initially show a few significant lags and gradually decay either in an exponential or oscillatory manner and the ACF plot will show significant autocorrelation at q number of lags and cut off after the qth lag.
On observing the above graph we can say that:
(a) The PACF plot decaying in an oscillatory manner, showing a few significant lags initially.
(b) There is one spike that crosses the blue line and the acf cuts off after the first lag.

Hence we can conclude that the model is of the form MA(1), moving average model of order 1.

```
#5. Fitting the MA model:

fit=auto.arima(data, seasonal="FALSE")
fit

## Series: data
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       -0.8092
## s.e.   0.0710
##
## sigma^2 = 15.53:  log likelihood = -276.26
## AIC=556.53   AICc=556.65   BIC=561.72

#The fitted value of the co-efficient (theta) is -0.8092. Since the data shows ARIMA(0,1,1) we say that the the AR part is of order 0 and the MA part is of order 1.

#6. Extract the residuals from the fitted model:

res = resid(fit)

#6. Verify all the assumptions:

#(a) Assumption 1: Errors are uncorrelated

#To check whether the errors are uncorrelated, we can get a basic idea using the acf plot but for a more firm conclusion we use the Portmanteu test

#ACF plot:
acf(res)
```

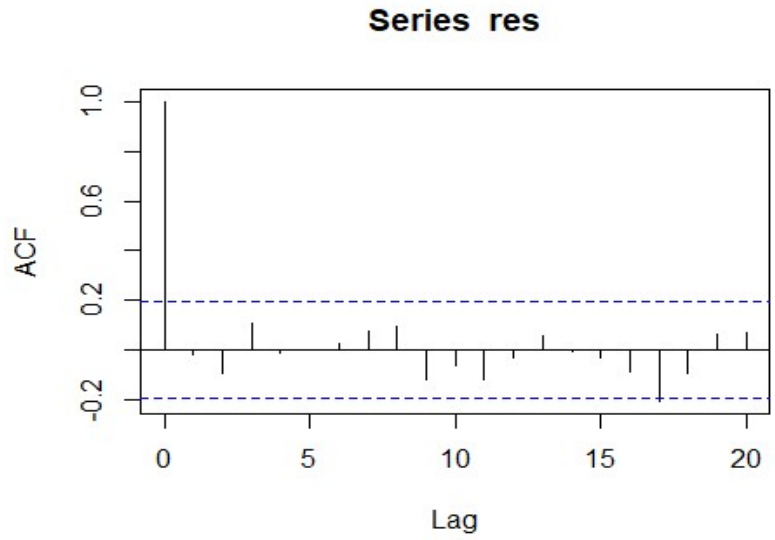

Figure 4: ACF Plot of Residuals

Since the ACF values lie within the threshold value for all lags, we can say that the errors are uncorrelated.

```
#Portmanteu test:
#H0: There is no significant correlation among the residual series
#H1: There is significant correlation among the residuals.

Box.test(res)

##
##  Box-Pierce test
##
## data:  res
## X-squared = 0.037841, df = 1, p-value = 0.8458

#Since the p-value is 0.84 (>0.05) we accept the null hypothesis that there is no significant correlation among the residuals.

#Therefore assumption 1 is satisfied

#(b) Assumption 2: Residuals are normally distributed

#For checking this assumption we can either use qq plots or some statistical test such as Shapiro Wilk test

#QQ or Quantile-Quantile plot:
qqnorm(res)
qqline(res)
```

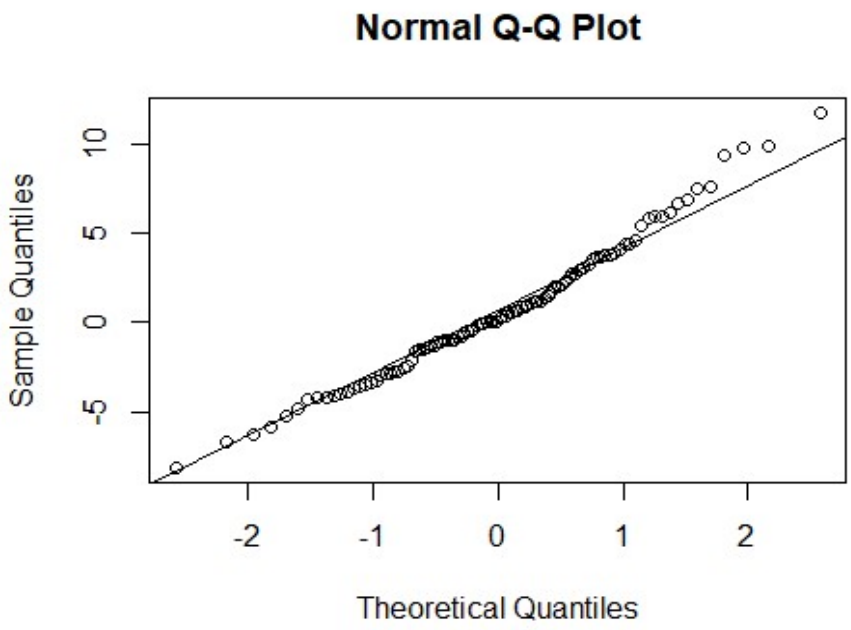

Figure 5: Q-Q Plot

From the qq plot, we see that most of the points lie on the line plotted, to get more clarity, we perform the Shapiro-Wilk test for normality.

```
#Shapiro-Wilk test:
#H0: Data is distributed normally
#H1: Data is not distributed normally

shapiro.test(res)

##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98273, p-value = 0.2156
```

CONCLUSION:

Since p value is >0.05, we will accept the null hypothesis of normality. Therefore, from the qq plot and shapiro test the assumption of normality is satisfied.

After performing the residual analysis on the given dataset, we can conclude that the residuals satisfy all the assumptions.