Time Series

Malak Gaballa-900201683 , Masa Tantawy-900201312 , Moustafa El Mahdy-900201154

2023-05-22

Intro

It is important to analyze the growing Egyptian population over the years. This data is a time series data that provides annual figures regarding the population of Egypt from 1950 to 2023. The data will be analyzed from 1950 to 2020 and the years 2021-2023 will be left out for forecasting in the end.

Reading the Dataset

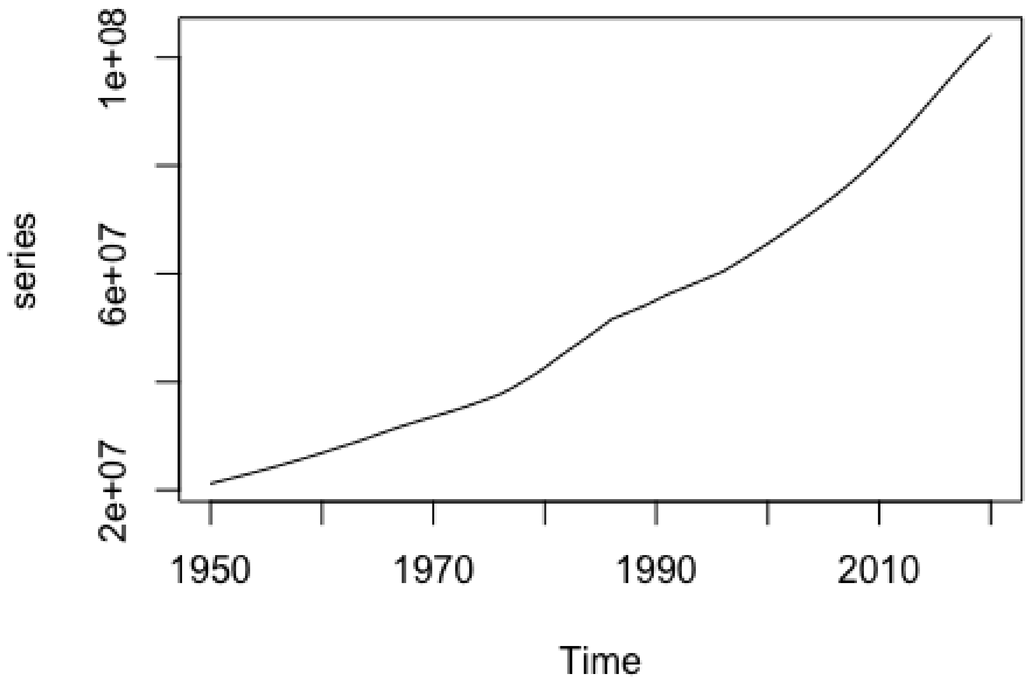
```
df=read.csv('EG Population TS.csv')
str(df); dim(df)

## 'data.frame':    74 obs. of  2 variables:
##  $ Year      : int  1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 ...
##  $ Population: int  21197691 21704443 22223309 22754580 23298551 23855527
24425817 25009741 25607624 26219800 ...

## [1] 74  2

head(df);tail(df)

##   Year Population
## 1 1950   21197691
## 2 1951   21704443
## 3 1952   22223309
## 4 1953   22754580
## 5 1954   23298551
## 6 1955   23855527

##    Year Population
## 69 2018   99834881
## 70 2019  101973865
## 71 2020  103994537
## 72 2021  105919663
## 73 2022  107770524
## 74 2023  109546720
```

Now, the last 3 years will be removed from the main series and used for forecasting.

```
forecast= df[c(72,73,74),]
data = df[-c(72,73,74),]
```

Step 1 : Plot the Series

```
series <- ts(data$Population,start=c(1950))
plot.ts(series)
```



From the plot of the series, it is very visible that there is an upward trend. To support this claim, Dickey Fuller's test will be used and it is expected that the difference will be taken to make it stationary.

Step 2: Stationarity and Trend Check

```
library(urca)
df=ur.df(series,type="trend",lags=1)
summary(df) # not stationary, no trend

##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -706537  -46386    5284   36417  273402
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.291e+04  5.135e+04   1.420    0.160
## z.lag.1     -3.342e-03  4.116e-03  -0.812    0.420
## tt           5.701e+03  4.054e+03   1.406    0.164
## z.diff.lag   9.303e-01  6.234e-02  14.923   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119500 on 65 degrees of freedom
## Multiple R-squared:  0.9589, Adjusted R-squared:  0.957
## F-statistic: 505.2 on 3 and 65 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -0.8118 1.7928 1.9003
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
df=ur.df(series,type="trend",lags=2)
summary(df) # not stationary, no trend
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -693470  -40905    4194   48515  261953
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.617e+04  5.236e+04   1.073    0.287
## z.lag.1     -1.422e-03  4.424e-03  -0.321    0.749
## tt           4.385e+03  4.314e+03   1.017    0.313
```

```
## z.diff.lag1  1.101e+00  1.242e-01   8.869 1.09e-12 ***
## z.diff.lag2 -2.063e-01  1.306e-01  -1.580    0.119
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119000 on 63 degrees of freedom
## Multiple R-squared:  0.9597, Adjusted R-squared:  0.9571
## F-statistic: 374.9 on 4 and 63 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -0.3214 2.0171 2.1054
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
df=ur.df(series,type="trend",lags=3)
summary(df) # not stationary, no trend
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -690164  -42310    5184   51947  258487
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.350e+04  5.378e+04   0.995    0.324
## z.lag.1     -1.127e-03  4.783e-03  -0.236    0.815
## tt           4.304e+03  4.630e+03   0.930    0.356
## z.diff.lag1  1.090e+00  1.286e-01   8.475 6.73e-12 ***
## z.diff.lag2 -1.484e-01  1.889e-01  -0.786    0.435
## z.diff.lag3 -5.642e-02  1.352e-01  -0.417    0.678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 120700 on 61 degrees of freedom
## Multiple R-squared:  0.959,  Adjusted R-squared:  0.9556
## F-statistic: 285.2 on 5 and 61 DF,  p-value: < 2.2e-16
```

```
##
##
## Value of test-statistic is: -0.2355 2.0359 2.1354
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

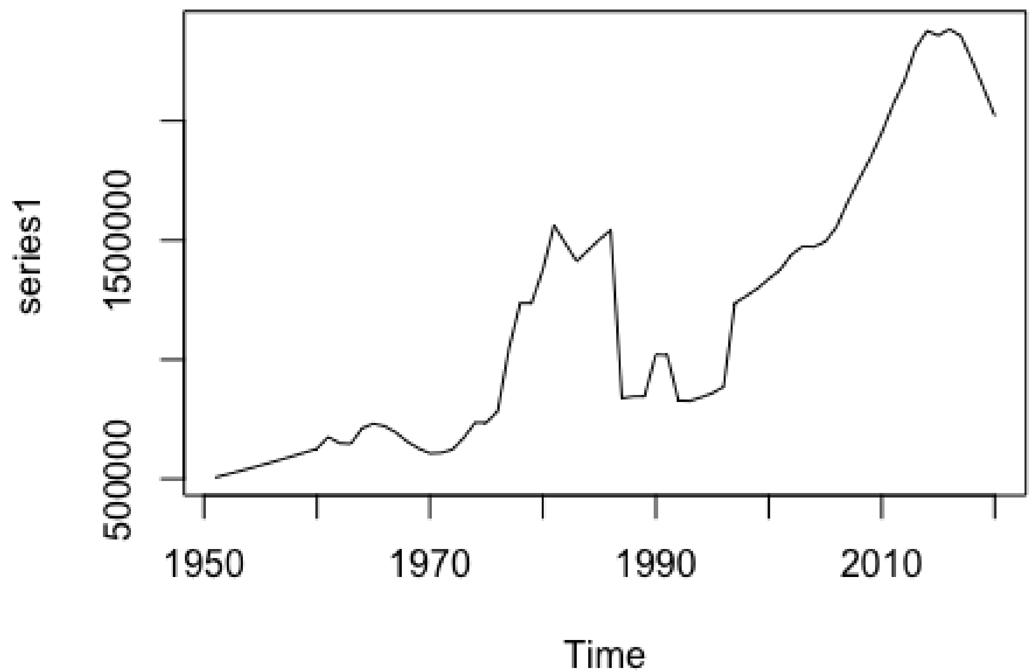
For all lags, the dickey fuller test concludes the following:

- The series is not stationary since the p-value is greater than 0.05, thus failing to reject "H0 : series not stationary".
- The series has no trend since the p-value is greater than 0.05, thus failing to reject "H0: series has no trend".

This suggests that we need to take the required difference to make the series stationary.

Step 3: Take First Difference

```
series1 <- diff(series, differences=1)
plot.ts(series1)
```



It is visible that an upward trend still exists. To support this claim, Dickey Fuller's test will be constructed and it is expected that the second difference will be taken to make it stationary.

Stationarity and Trend Check

```
library(urca)
df=ur.df(series1,type="trend",lags=1)
summary(df) # stationary, has a trend

##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -686860  -40245    3592   52542  260861
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.630e+04  3.314e+04   1.397   0.1672
## z.lag.1     -1.178e-01  5.291e-02  -2.227   0.0294 *
## tt           3.088e+03  1.513e+03   2.041   0.0454 *
## z.diff.lag   2.196e-01  1.230e-01   1.785   0.0790 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118200 on 64 degrees of freedom
## Multiple R-squared:  0.09671,    Adjusted R-squared:  0.05437
## F-statistic: 2.284 on 3 and 64 DF,  p-value: 0.08733
##
##
## Value of test-statistic is: -2.2274 2.1803 2.4985
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
df=ur.df(series1,type="trend",lags=2)
summary(df) # stationary, has a trend
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -684929  -41879    4015   54776  257341
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.709e+04  3.433e+04   1.372   0.1752
## z.lag.1     -1.259e-01  5.576e-02  -2.258   0.0275 *
## tt           3.281e+03  1.586e+03   2.069   0.0427 *
```

```
## z.diff.lag1  2.145e-01  1.252e-01   1.714   0.0916 .
## z.diff.lag2  6.501e-02  1.291e-01   0.503   0.6165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119800 on 62 degrees of freedom
## Multiple R-squared:  0.1007, Adjusted R-squared:  0.04265
## F-statistic: 1.735 on 4 and 62 DF,  p-value: 0.1536
##
##
## Value of test-statistic is: -2.2578 2.2042 2.5658
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
df=ur.df(series1,type="trend",lags=3)
summary(df) # not stationary, has a trend
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -682959  -40172    5525   57956  254215
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.788e+04  3.573e+04   1.340   0.1853
## z.lag.1     -1.336e-01  5.913e-02  -2.259   0.0276 *
## tt           3.459e+03  1.665e+03   2.078   0.0420 *
## z.diff.lag1  2.192e-01  1.275e-01   1.720   0.0907 .
## z.diff.lag2  6.284e-02  1.312e-01   0.479   0.6337
## z.diff.lag3  5.554e-02  1.331e-01   0.417   0.6780
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 121600 on 60 degrees of freedom
## Multiple R-squared:  0.1036, Adjusted R-squared:  0.02888
## F-statistic: 1.387 on 5 and 60 DF,  p-value: 0.2421
```

```
## 
## 
## Value of test-statistic is: -2.2586 2.2042 2.57 
## 
## Critical values for test statistics: 
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

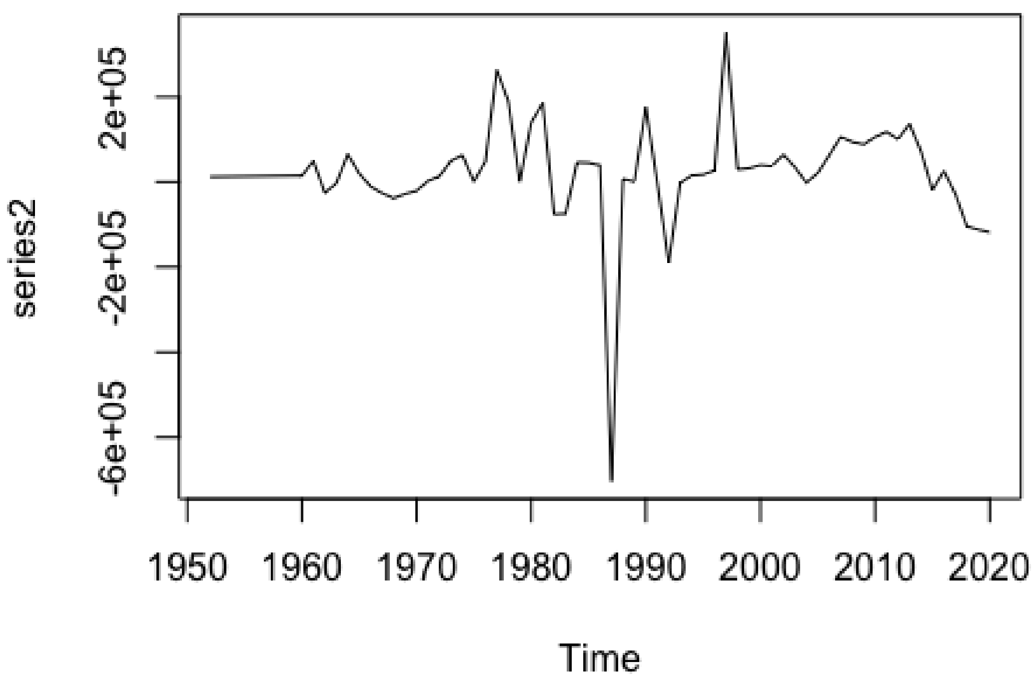
For all lags, the dickey fuller test concludes the following:

- The series is stationary since the p-value is less than 0.05, thus rejecting “H0 : series not stationary”.
- The series has a trend since the p-value is less than 0.05, thus rejecting “H0: series has no trend”.

This suggests that we need to take the required difference to make the series stationary.

Step 4: Take Second Difference

```
series2 <- diff(series, differences=2)
plot.ts(series2) #no pattern visible
```



After taking the second difference, no pattern is visible. To support this, Dickey Fuller's test will be constructed.

Stationarity and Trend Check

```
library(urca)
df=ur.df(series2,type="trend",lags=1)
summary(df) # stationary, no trend

##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
```

```
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726887  -30755   -2329   38184  326487
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.366e+04  3.126e+04   0.437    0.664
## z.lag.1     -8.476e-01  1.678e-01  -5.053 3.99e-06 ***
## tt           1.401e+02  7.857e+02   0.178    0.859
## z.diff.lag   1.107e-02  1.287e-01   0.086    0.932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 123600 on 63 degrees of freedom
## Multiple R-squared:  0.4136, Adjusted R-squared:  0.3857
## F-statistic: 14.81 on 3 and 63 DF,  p-value: 2.103e-07
##
##
## Value of test-statistic is: -5.0526 8.5657 12.8322
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
df=ur.df(series2,type="trend",lags=2)
summary(df) # stationary, no trend
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726474  -32661    -696   37996  325949
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.383e+04  3.272e+04   0.423    0.674
```

```
## z.lag.1     -8.732e-01  2.085e-01  -4.189 9.18e-05 ***
## tt           1.570e+02  8.223e+02   0.191    0.849
## z.diff.lag1  3.588e-02  1.749e-01   0.205    0.838
## z.diff.lag2  2.823e-02  1.321e-01   0.214    0.831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 125600 on 61 degrees of freedom
## Multiple R-squared:  0.414,  Adjusted R-squared:  0.3756
## F-statistic: 10.77 on 4 and 61 DF,  p-value: 1.134e-06
##
##
## Value of test-statistic is: -4.1887 5.9387 8.8822
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

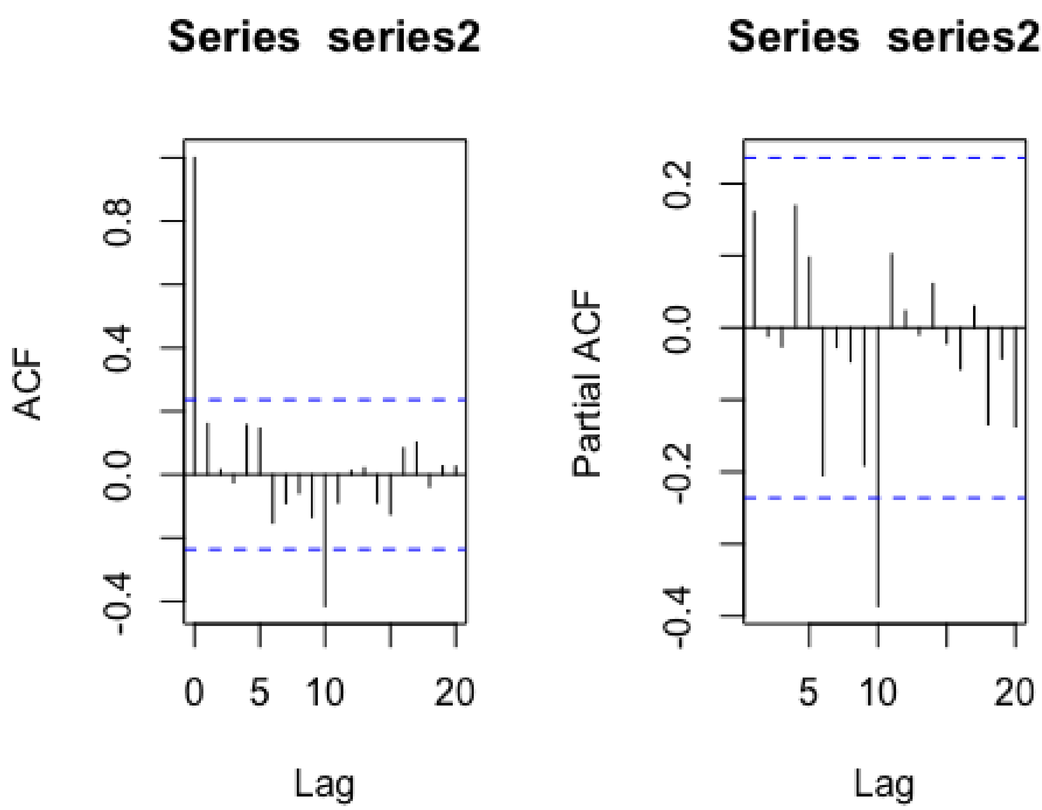
For all lags, the dickey fuller test concludes the following:

- The series is stationary since the p-value is less than 0.05, thus rejecting "H0 : series not stationary".
- The series has no trend since the p-value is greater than 0.05, thus failing to reject "H0: series has no trend".

Since the series is now stationary and has no trend, we shall proceed by plotting the ACF and PACF to determine the order of the model.

STEP 5: examine ACF & PACF Plots

```
op = par(mfrow=c(1,2))
acf(series2, lag.max=20) # 1 spike
pacf(series2, lag.max=20) # 1 spike

par(op)
```



Both ACF and PACF have one spike at lag 10, suggesting that an ARMA model will be fitted ; this is because both functions can be considered decaying. Multiple models will be fitted with several values of p and q.

STEP 6: Chosen Models

```
m1<-arima(series, order=c(1,2,1)); m1 #AIC = 1815.06

##
## Call:
## arima(x = series, order = c(1, 2, 1))
##
## Coefficients:
##          ar1      ma1
##       0.6528  -0.5013
## s.e.  0.3975   0.4536
```

```
##
## sigma^2 estimated as 1.424e+10:  log likelihood = -904.53,  aic = 1815.06
```

```
m2<-arima(series, order=c(2,2,1)); m2 #AIC = 1817.17
```

```
##
## Call:
## arima(x = series, order = c(2, 2, 1))
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1     ar2     ma1
##        0.091  0.0375  0.0959
## s.e.     NaN     NaN     NaN
##
## sigma^2 estimated as 1.427e+10:  log likelihood = -904.59,  aic = 1817.17
```

```
m3<-arima(series, order=c(1,2,2)); m3 #AIC = 1816.9
```

```
##
## Call:
## arima(x = series, order = c(1, 2, 2))
##
## Coefficients:
##           ar1     ma1     ma2
##       -0.4054  0.6040  0.1619
## s.e.   0.5235  0.5174  0.1626
##
## sigma^2 estimated as 1.421e+10:  log likelihood = -904.45,  aic = 1816.9
```

```
m4<-arima(series, order=c(2,2,2)); m4 #AIC = 1812.75
```

```
##
## Call:
## arima(x = series, order = c(2, 2, 2))
##
## Coefficients:
##           ar1      ar2     ma1     ma2
##       -0.2730  -0.8197  0.4717  1.0000
## s.e.   0.0848   0.0778  0.0587  0.1492
##
## sigma^2 estimated as 1.229e+10:  log likelihood = -901.37,  aic = 1812.75
```

```
m5<-arima(series, order=c(3,2,2)); m5 #AIC = 1814.65
```

```
##
## Call:
## arima(x = series, order = c(3, 2, 2))
##
## Coefficients:
```

```
##           ar1     ar2     ar3     ma1     ma2
##       -0.2423  -0.811  0.0393  0.4715  1.0000
## s.e.   0.1291   0.084  0.1225  0.0569  0.1201
##
## sigma^2 estimated as 1.23e+10:  log likelihood = -901.32,  aic = 1814.65
```

```r
m6<-arima(series, order=c(2,2,3)); m6 #AIC = 1814.66
```

```
##
## Call:
## arima(x = series, order = c(2, 2, 3))
##
## Coefficients:
##           ar1      ar2     ma1     ma2     ma3
##       -0.2876  -0.8234  0.5120  1.0188  0.0403
## s.e.   0.1000   0.0790  0.1496  0.1400  0.1374
##
## sigma^2 estimated as 1.23e+10:  log likelihood = -901.33,  aic = 1814.66
```

```r
m7<-arima(series, order=c(3,2,3)); m7 #AIC = 1817.52
```

```
##
## Call:
## arima(x = series, order = c(3, 2, 3))
##
## Coefficients:
##          ar1      ar2     ar3     ma1     ma2      ma3
##       0.0016  -0.2527  0.5522  0.2274  0.4615  -0.6289
## s.e.  0.3239   0.2289  0.2304  0.3338  0.2886   0.3311
##
## sigma^2 estimated as 1.25e+10:  log likelihood = -901.76,  aic = 1817.52
```

```r
# The AIC is increasing, so we will stop fitting larger ARMA models.
# Chosen Model: M4 ARIMA (2,2,2)
```

After fitting several models, it is visible that the AIC is increasing. Thus, there is no need to fit larger models. The best performing model is Model 4 which is ARIMA(2,2,2) since it has the lowest AIC measure at 1812.75. Next , the assumptions will be checked for this chosen model.

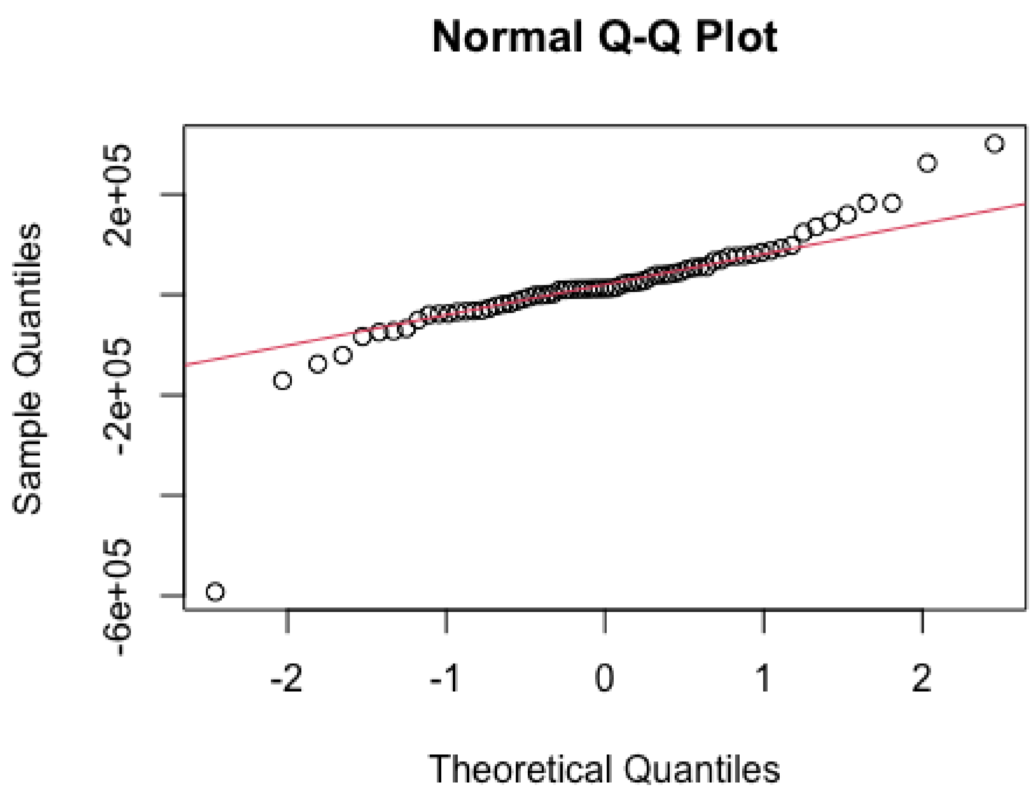
STEP 7: Checking NICE Assumptions

The assumptions that will be checked are as follows:

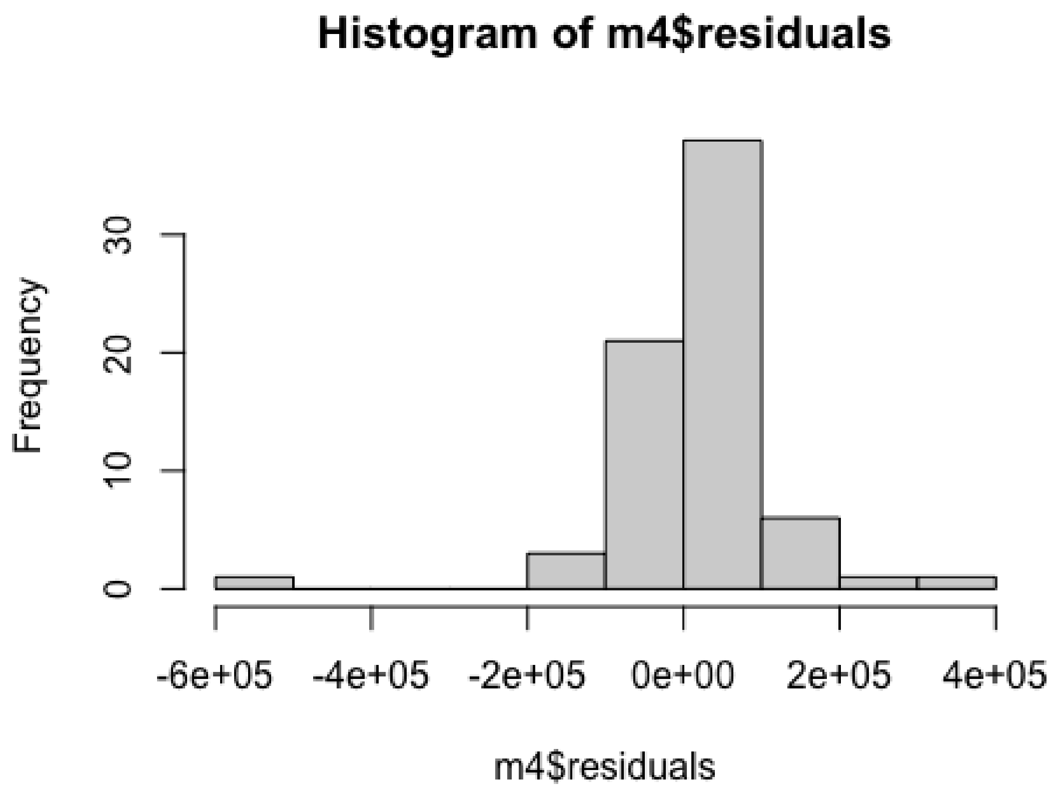
1. **N**ormality of Residuals
2. **I**ndependence of Residuals
3. **C**onstant Variance of Residuals
4. **E**xpectation = 0

Best Model : M4

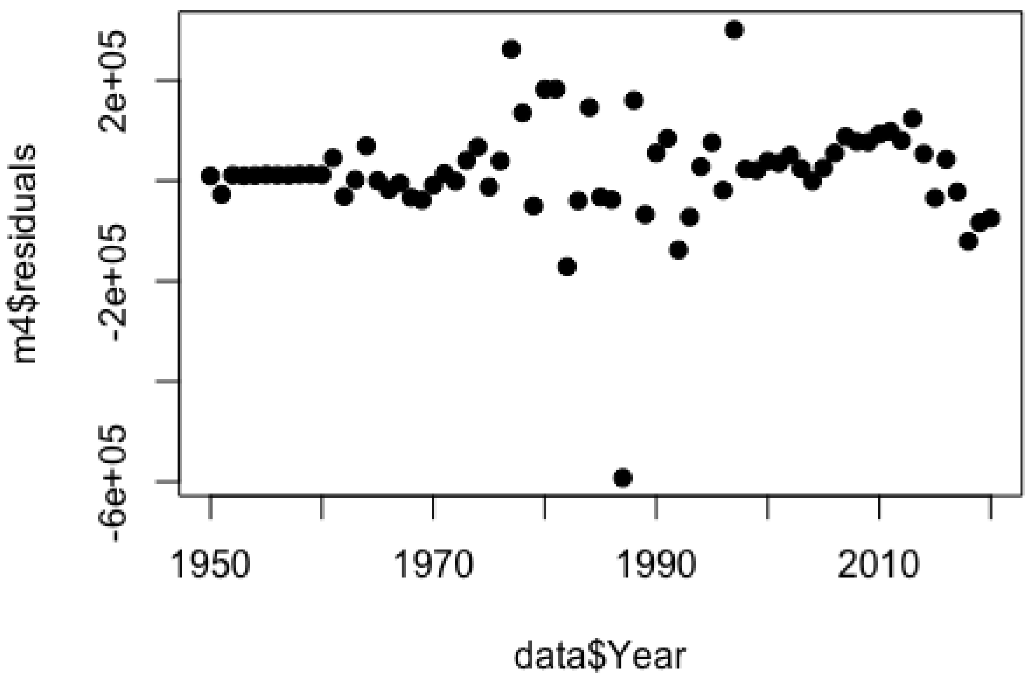
```
# Plot 1: QQ plot of residuals
qqnorm(m4$residuals); qqline(m4$residuals, col = 2)
```



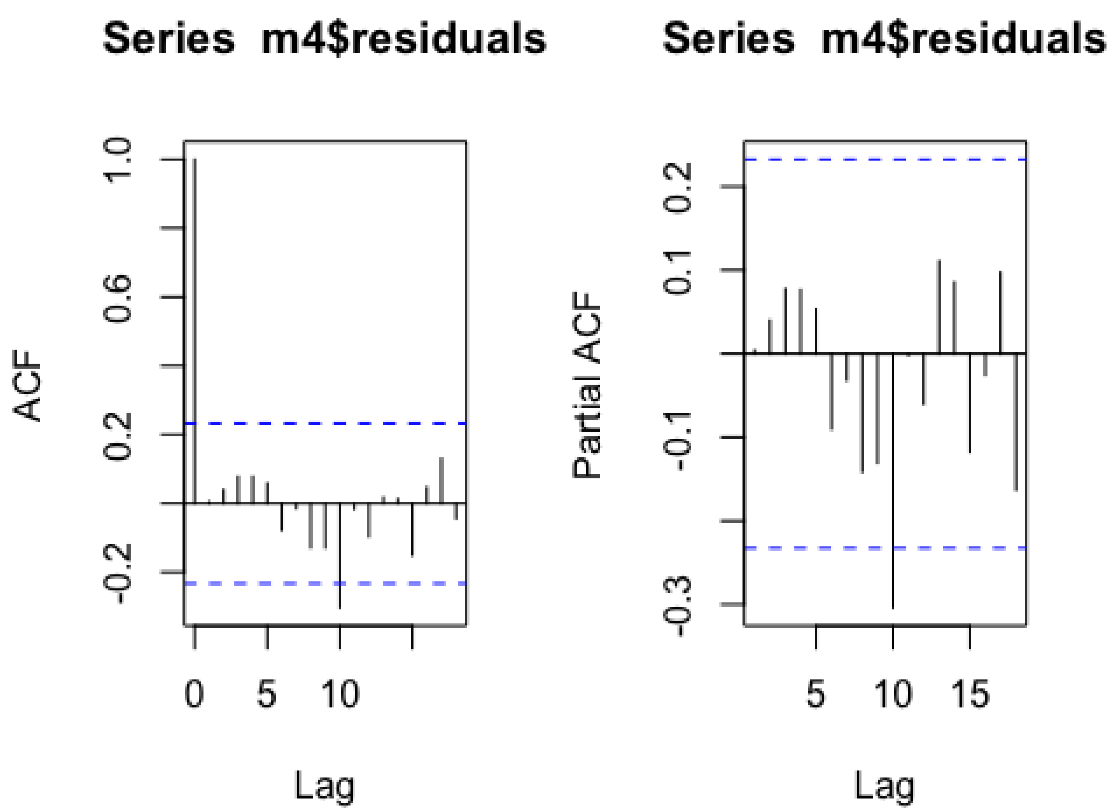
```
#normal but the data is too peaked
# Plot 2: Histogram of residuals
hist(m4$residuals)
```



```r
# ----> Checking for independence
# Plot 3: Index plot of residuals
plot(data$Year, m4$residuals, pch=19) # random scatter of points
```



```
# Plot 4: ACF and PACF of residuals
op = par(mfrow=c(1,2))
acf(m4$residuals); pacf(m4$residuals) #spike at lag 10 in both ACF & PACF
```

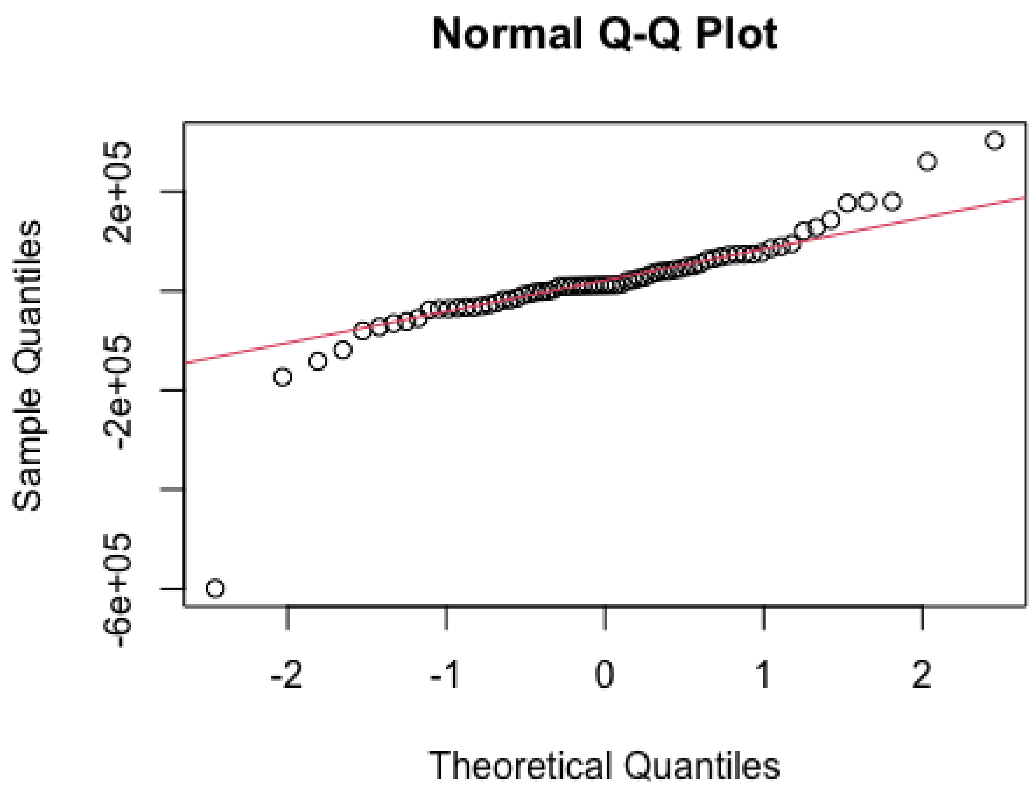



```
par(op)
# Ljung-Box Pierce Test
Box.test(m4$residuals, lag = 20, fitdf = 1)

##
##  Box-Pierce test
##
## data:  m4$residuals
## X-squared = 14.483, df = 19, p-value = 0.7549

#strongly fail to reject H0: independence of residuals
```

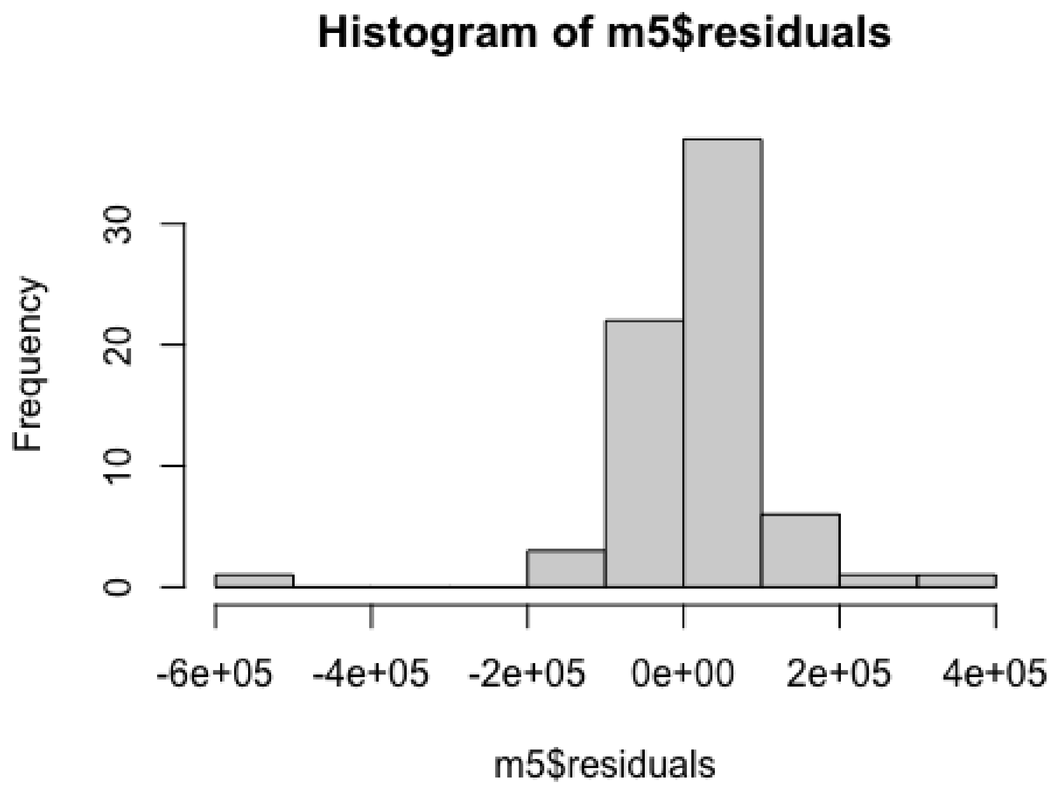
Since all the assumptions are verified except the independence of residuals due to spikes in ACF and PACF, the second best model will be checked. Chosen Model: M5 ARIMA (3,2,2).

Second Best Model : M5

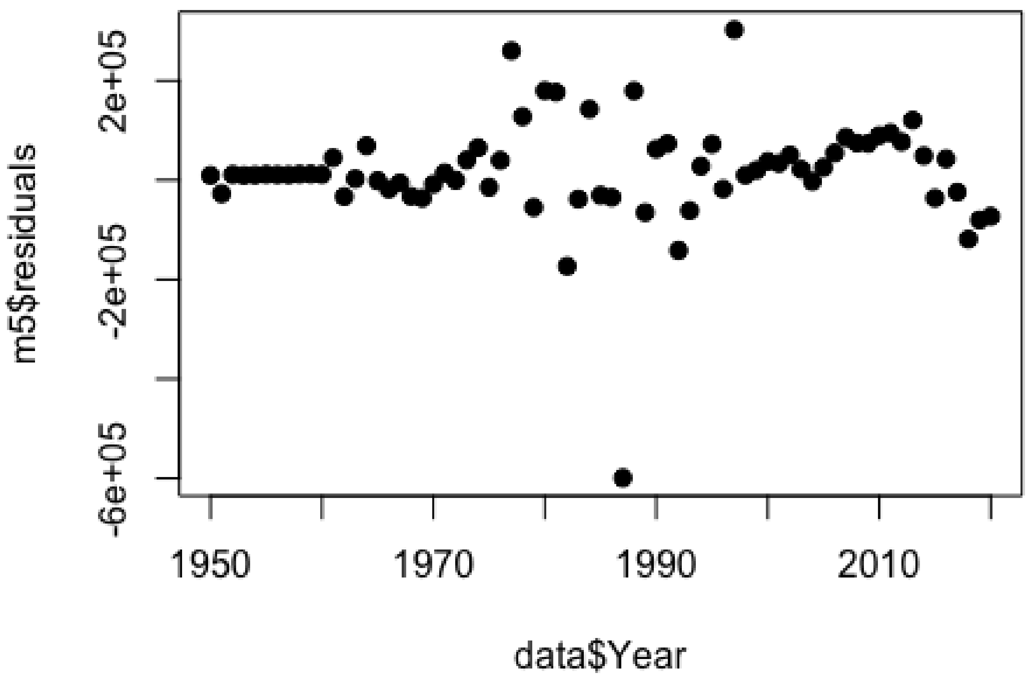
```
# Plot 1: QQ plot of residuals
qqnorm(m5$residuals); qqline(m5$residuals, col = 2)  #normal but the data is
too peaked
```



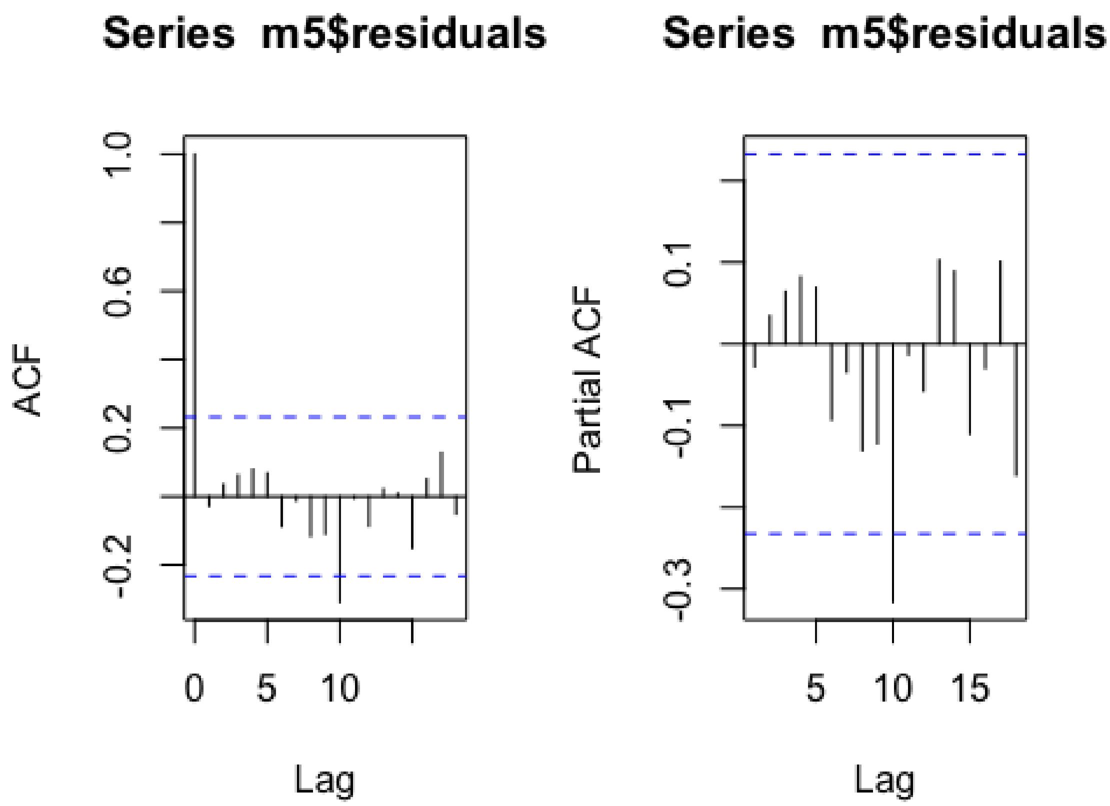
```r
# Plot 2: Histogram of residuals
hist(m5$residuals)
```



```r
# ----> Checking for independence
# Plot 3: Index plot of residuals
plot(data$Year, m5$residuals, pch=19) # random scatter of points
```



```
# Plot 4: ACF and PACF of residuals
op = par(mfrow=c(1,2))
acf(m5$residuals); pacf(m5$residuals) #spike at lag 10 in both ACF & PACF
```




```
par(op)
# Ljung-Box Pierce Test
Box.test(m4$residuals, lag = 20, fitdf = 1)

##
##  Box-Pierce test
##
## data:  m4$residuals
## X-squared = 14.483, df = 19, p-value = 0.7549

#strongly fail to reject H0: independence of residuals
```

The assumptions of this model are similar to the previous model assumptions.Thus, using AIC and parsimony principle, the best model is M4 ARIMA (2,2,2). Finally, this model will be used to forecast the Egyptian Population for the years 2021,2022, and 2023.

STEP 6: Forecasting

```
library(forecast)

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

pred= predict(m4,n.ahead = 3)
pred$pred ; forecast

## Time Series:
## Start = 2021
## End = 2023
## Frequency = 1
## [1] 106021637 108070089 110107443

##    Year Population
## 72 2021  105919663
## 73 2022  107770524
## 74 2023  109546720
```

Year	Actual Population	Forecasted Population
2021	105,919,663	106,021,637
2022	107,770,524	108,070,089
2023	109,546,720	110,107,443

It is clear that the forecasted values are very close to the actual figures, indicating that the model is a plausible decision. It is worth noting that as years go by, the forecasted population will become of less accuracy since the error of prediction and variance increase.