LONDON HOUSING PRICE OVER THE YEARS 2009-2019

(EXTRACTION OF STATIONARITY USING DIFFERENT METHODS)

A.Mayuri(2348133)

[COMPANY NAME] [Company address]

LONDON HOUSING PRICE OVER THE YEARS 2009-2019

A.Mayuri(2348133)

2024-02-15

QUESTION:

Choose a non seasonal time series data,

1)illustrate the method of differencing.

2)ordinary least squares

3)moving average smoothing to extract the stationary version.

OBJECTIVE:

To Extract Stationary component(Part) from Non-Stationary Data

1)Perform Basic operations to understand the dataset

2)Differencing Method

3)Ordinary Least Square Method

4)Moving Average Smoothing Method.

DATASET:

The dataset has been sourced from **https://www.kaggle.com/datasets/justinas/housing-in-london** where the price of houses has been recorded monthly over the 11 years period. from 2009-2019.

DATA DESCRIPTION:

1)location(london): this is uniform throught out the dataset which confirms that this is a time series data-set and not a panel/cross-sectional data.

2) Date: The dataset is monthly data thus the monthly dates are recorded. with corresponding monthly price averages.

3) Price average: Price average is computed by summing over all the prices by the number of houses sold in that particular month.

Exploratory Data Analyses: Since the

1)dataset taken has the same observation ie (Place-London), code(city=E92000001)

2)the date being a non-disruptive/ no gaps and hence continuous in nature.

3) Average price of the houses (Variable of interest)

Performing a complete time series analysis is the best way to understand the data (EDA).

IMPORT DATASET

```
library(readr)
LHPC <- read_csv("C:/Users/mayur/Desktop/Mstat/Semesters/Tri-sem3/Time series/Dataset/LHPC.csv")

## Rows: 132 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): date, area, code
## dbl (1): average_price
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

View(LHPC)
attach(LHPC)
```

IMPORT DATA SET -(TIME SERIES - ONLY PRICE COMPONENT)

```
library(readr)
LHP1 <- read_csv("C:/Users/mayur/Desktop/Mstat/Semesters/Tri-sem3/Time series/Dataset/LHP1.csv")

## Rows: 132 Columns: 1
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (1): price
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

View(LHP1)
attach(LHP1)
```

CONVERT DATA INTO TIME-SERIES AND PLOT

```
data2=ts(LHP1,start=2009,frequency=12) # converting it into a time series data
data2

##      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct
## 2009 162673 160956 159340 160701 162740 164536 167673 169603 171214 172314
## 2010 174458 175248 174765 176796 177754 178655 180519 180807 180231 178102
## 2011 174442 173811 173046 175490 174668 174838 177164 177335 176783 175171
## 2012 174179 174161 174323 176543 177026 178696 179756 180129 179563 178412
## 2013 176816 177203 178189 179900 180621 182088 184274 185642 186082 185358
## 2014 188265 189347 190037 194251 196171 197951 200825 203406 203639 203311
## 2015 202856 203424 203360 205936 208265 209874 213518 215756 216350 216676
## 2016 220361 220627 222663 223784 226370 228430 230868 231176 230848 229944
## 2017 231593 232696 231760 235021 236727 238595 241406 242628 242041 242003
## 2018 241061 241989 240428 242396 243445 244962 247981 248620 248248 247676
```

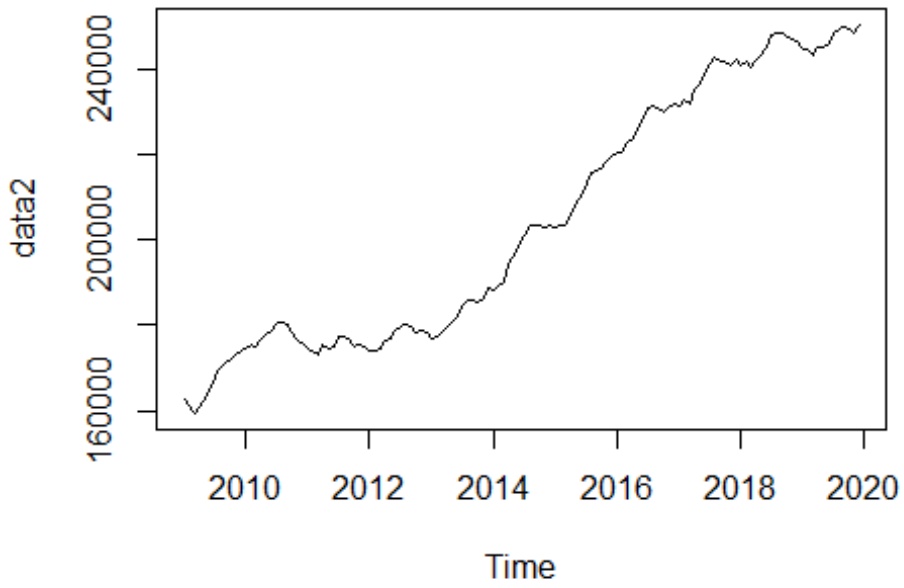
```
## 2019 244641 244582 243281 245077 245255 246140 248562 249432 249942 249376
##         Nov    Dec
## 2009 172818 174136
## 2010 176301 176036
## 2011 175200 174812
## 2012 178662 178406
## 2013 186260 188544
## 2014 202704 203346
## 2015 218500 219582
## 2016 231053 231922
## 2017 241086 242378
## 2018 246896 246518
## 2019 248515 250410
```

```
LHP1
```

```
## # A tibble: 132 × 1
##     price
##     <dbl>
##  1 162673
##  2 160956
##  3 159340
##  4 160701
##  5 162740
##  6 164536
##  7 167673
##  8 169603
##  9 171214
## 10 172314
## # i 122 more rows
```

```
length(LHP1)
```

```
## [1] 1

ts.plot(data2)
```



Interpretation: Here we observe that the dataset only has a trend (upward) and irregularity component. we cannot use a multiplicative model due to the absence of seasonality component.

Thus, the model is of additive form with trend and irregularity component.

Note: The data has no seasonality component involved.

MATHEMATICAL MODEL:

z(t)=m(t)+e(t)

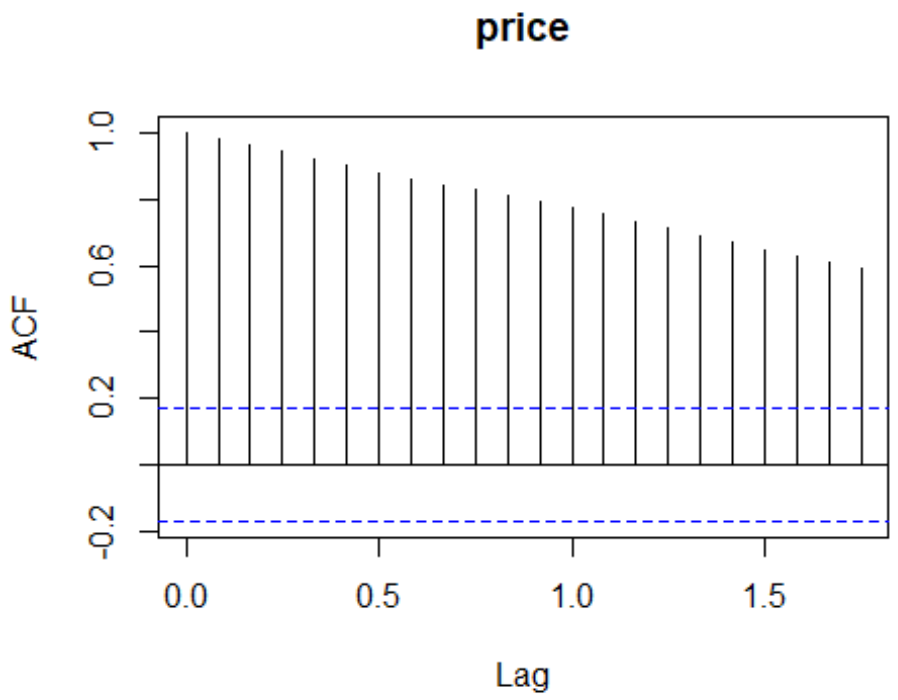
where, z(t)-time series variable dependent on time,

m(t)-trend component

e(t)-irregularity component.

AUTO-CORRELATION FUNCTION PLOT:

```
acf(data2)
```



Here the model has not achieved stationarity because the the acf lines are well above the band line. The conversion into stationary process is demonstrated using differencing method below.

METHOD OF DIFFERENCING.

CONCEPT: WE NEED TO TRANSFORM THE DATA TO STATIONARY FORM. A POLYNOMIAL DEGREE SAY P FORM MODEL WILL GIVE A STATIONARY MODEL POST DIFFERENCING P TIMES.

DIFFERENCING METHOD AND PLOTTING (DEMONSTRATION)

```
diffdata=diff(data2)
diffdata
```

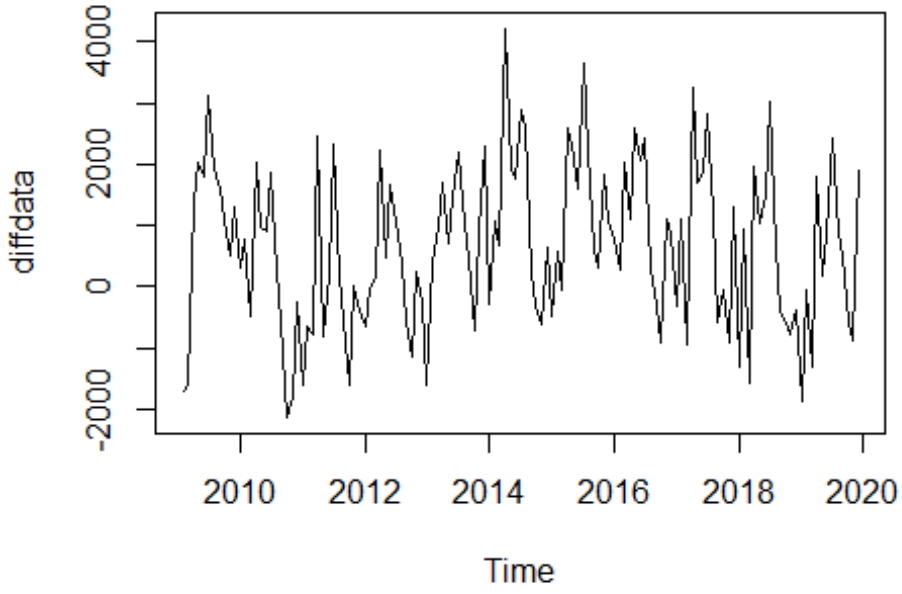
```
##      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 2009     -1717 -1616 1361 2039 1796 3137 1930 1611 1100  504 1318
## 2010  322  790 -483 2031  958  901 1864  288 -576 -2129 -1801 -265
```

```
## 2011 -1594  -631  -765  2444  -822   170  2326   171  -552 -1612    29  -388
## 2012  -633   -18   162  2220   483  1670  1060   373  -566 -1151   250  -256
## 2013 -1590   387   986  1711   721  1467  2186  1368   440  -724   902  2284
## 2014  -279  1082   690  4214  1920  1780  2874  2581   233  -328  -607   642
## 2015  -490   568   -64  2576  2329  1609  3644  2238   594   326  1824  1082
## 2016   779   266  2036  1121  2586  2060  2438   308  -328  -904  1109   869
## 2017  -329  1103  -936  3261  1706  1868  2811  1222  -587   -38  -917  1292
## 2018 -1317   928 -1561  1968  1049  1517  3019   639  -372  -572  -780  -378
## 2019 -1877   -59 -1301  1796   178   885  2422   870   510  -566  -861  1895
```

```
ts.plot(diffdata)
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.3.2
```

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



which is a stationary process, since it has a

1)constant variance

2)Constant mean.

CONFIRM THE STATIONARITY (AUGMENTED DICKY -FULLER TEST)

Hypothesis:

h0: the data trend is not stationary

h1: the data trend is stationary

```
adf.test(diffdata)

## Warning in adf.test(diffdata): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
```

```
## data:  diffdata
## Dickey-Fuller = -6.2938, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

since p value is **less** than the significance value alpha(0.05).

Thus we reject null hypothesis and conclude that the data is stationary in nature.

MOVING AVERAGE SMOOTHING:

CONCEPT:

Moving averages are a series of averages calculated using sequential segments of data points over a series of values. They have a length, which defines the number of data points to include in each average

TYPES

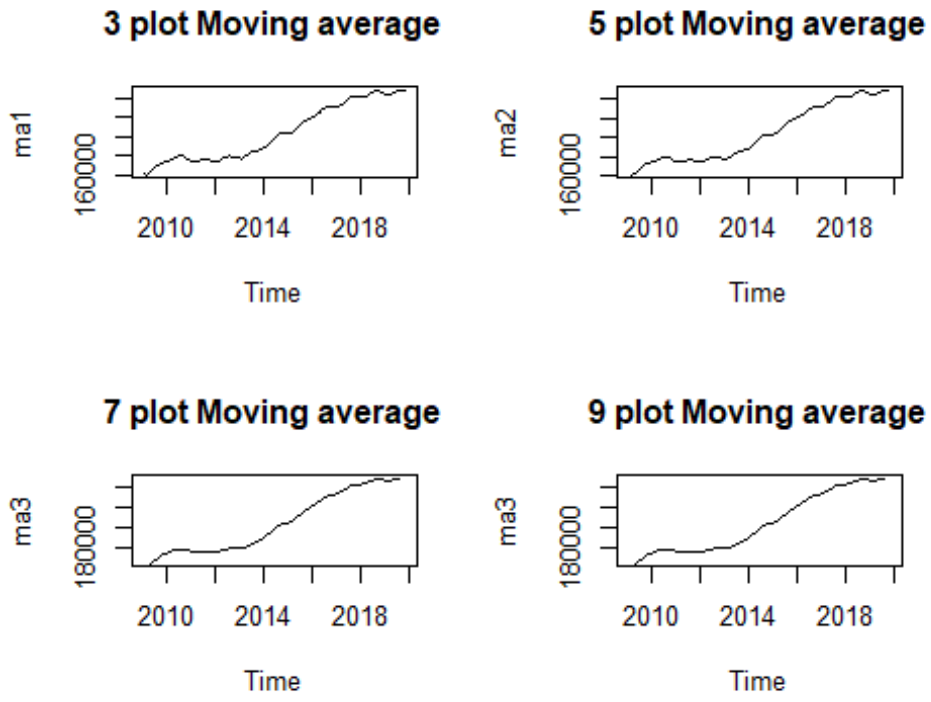
CENTERED MOVING AVERAGES: It includes both previous and future observations to calculate the average at a given point in time.

ONE SIDED MOVING AVERAGE: One-sided moving averages include the current and previous observations for each average. For example, the formula for a moving average (MA) of X at time t with a length of 5 is the following:

MA(5)=X(t-4)+X(t-3)+X(t-2)+X(t-1)+X(t)/5

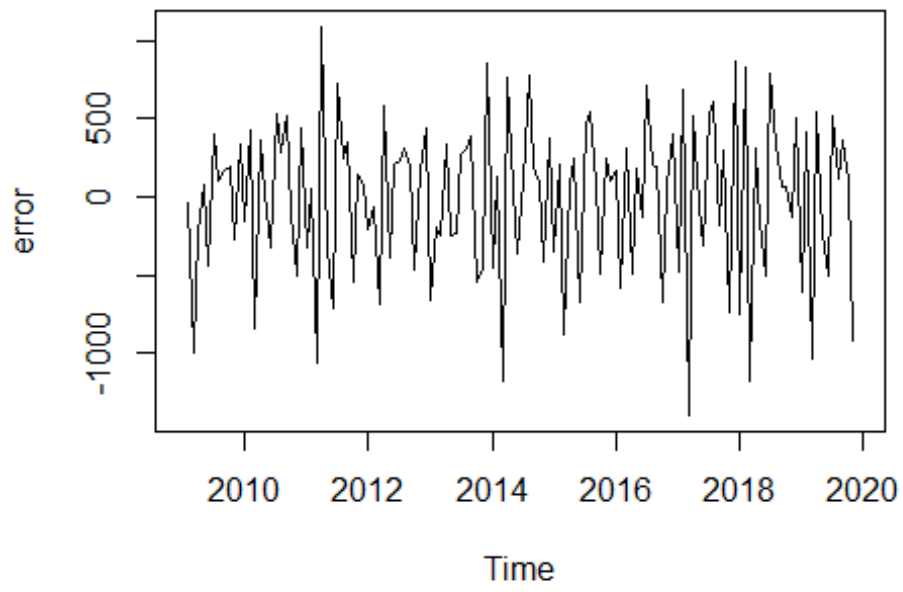
```
library(forecast)

## Warning: package 'forecast' was built under R version 4.3.2

ma1=ma(data2, order=3)
ma2=ma(data2,order=5)
ma3=ma(data2,order=7)
ma3=ma(data2,order=9)
par(mfrow=c(2,2))# combine multiple plots in one
ts.plot(ma1,main="3 plot Moving average")
```

```
ts.plot(ma2,main="5 plot Moving average")
ts.plot(ma3,main="7 plot Moving average")
ts.plot(ma3,main="9 plot Moving average")
```



We observe that as the order MA increases, the smoothing increases. However, to extract the stationarity we need to use the least order MA to reduce the loss of information.

```
error=data2-ma1 #
ts.plot(error)
```



Interpretation: stationarity is achieved in 3 pt moving average (MA). since it has a

1)constant variance

2)Constant mean.

Note: we need not do it for higher order to avoid loss of information. since higher order MA (lesser observation.) we tend to lose quality Information.

ORDINARY LEAST SQUARE METHOD:

CONCEPT:

```
library(readr)
LHP3 <- read_csv("C:/Users/mayur/Desktop/Mstat/Semesters/Tri-sem3/Time series/Dataset/LHP3.csv")

## Rows: 132 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
```

```
## Delimiter: ","
## dbl (3): price, crime, month
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

View(LHP3)
attach(LHP3)

## The following object is masked from LHP1:
##
##     price
```

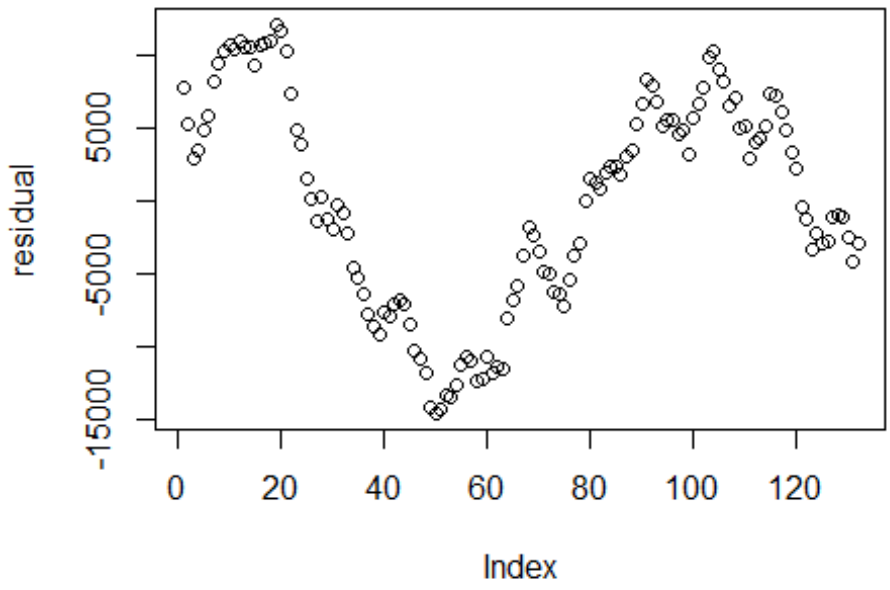
REGRESSION MODEL

```
model=lm(LHP3$price~LHP3$month)
summary(model)

##
## Call:
## lm(formula = LHP3$price ~ LHP3$month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14569.2  -5934.4    553.2   5712.4  12043.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 154196.24    1288.43   119.7   <2e-16 ***
## LHP3$month     751.52      16.81    44.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7359 on 130 degrees of freedom
```

```
## Multiple R-squared:  0.9389, Adjusted R-squared:  0.9385
## F-statistic:  1999 on 1 and 130 DF,  p-value: < 2.2e-16
```

We observe that all the variables are statistically significant(pvalue are significant) and the adjusted r^2 =0.956 shows that the model is a good fit.

```
residual=resid(model)
plot(residual)
```



###Residual

Analysis:

NORMALITY TEST

Hypothesis:

Ho: The residual follows normal distribution

H1: The residual does not follow normal distribution

```
shapiro.test(residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residual
## W = 0.9572, p-value = 0.0003766
```

The model does not follow normal distribution since the pvalue is less than significant alpha

CONSTANT VARIANNCE

From the plot residual plot, we observe that the residual in the model defies the assumption of constant variance. This can be verified using BP test.

Hypothesis: Ho: The residual has a constant Variance H1: The residual does not have a constantt Variance.

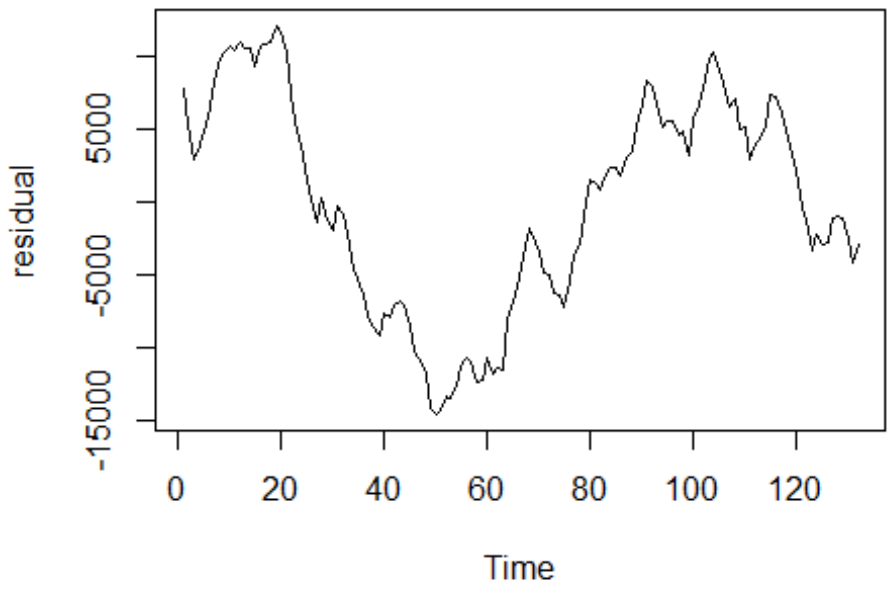
```
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

bptest(model)

##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 18.03, df = 1, p-value = 2.174e-05
```

The value is less than alpha level(0.05) significance, Thus we reject null hypothesis and conclude that the residual does not follow constant variance.

INFERENCE:

Since the model defies the residual analysis thus the assumptions of the error terms. we will not get the stationary plot (stationarity) .

```
ts.plot(residual)
```



The same can be verified using adf test since the data only includes trend components.

ADF TEST

Ho: The residual is not stationary

H1: The residual is stationary.

```
adf.test(residual)

##
##  Augmented Dickey-Fuller Test
##
```

```
## data:  residual
## Dickey-Fuller = -1.5962, Lag order = 5, p-value = 0.7444
## alternative hypothesis: stationary
```

Since the P-value > o.o5 (alpha) level of significance we will fail to reject null hypothesis and conclude that the residual is not stationary.

CONCLUSION:

Higher order of polynomial fitting is not suitable since it was observed that polynomial fitting did not significantly impact in the betterment of residual analysis. Thus we conclude stating that Ordinary differencing method, Moving Average method were viable for the dataset to extract stationarity. However, OLS method is not suitable.

THE END!