LONDON HOUSING PRICE OVER THE YEARS 2009-2019

A.Mayuri(2348133)

[COMPANY NAME] [Company address]

London Housing Price Over the Years 2009-2019

A.Mayuri(2348133)

2024-02-15

Question:

Choose a real time series data and illustrate the following questions.

1. Perform the various steps to understand the components of the data and comment on it.

2. Discuss the mathematical model for the given data.

3. Comment about the behavior of Autocorrelation function plot.

4. Does the logarithmic transformation help in achieving stationarity of the data set? Justify your answer. Briefly prepare a report based on the above questions.

Objective:

1)To Understand the components of the data
2)Develop a time series model
3)comment on the ACF plot
4)Logarithmic transformation needed to achieve stationarity?

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 through 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 m
essage.

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: 131 Columns: 1
## ── Column specification ──────────────────────────────────────────────────
────────
## Delimiter: ","
## dbl (1): 162673
##
## 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 m
essage.

View(LHP1)
attach(LHP1)
```

Convert data into Time-Series And Plot

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

##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2009 160956 159340 160701 162740 164536 167673 169603 171214 172314 172818
## 2010 175248 174765 176796 177754 178655 180519 180807 180231 178102 176301
## 2011 173811 173046 175490 174668 174838 177164 177335 176783 175171 175200
## 2012 174161 174323 176543 177026 178696 179756 180129 179563 178412 178662
## 2013 177203 178189 179900 180621 182088 184274 185642 186082 185358 186260
## 2014 189347 190037 194251 196171 197951 200825 203406 203639 203311 202704
## 2015 203424 203360 205936 208265 209874 213518 215756 216350 216676 218500
## 2016 220627 222663 223784 226370 228430 230868 231176 230848 229944 231053
## 2017 232696 231760 235021 236727 238595 241406 242628 242041 242003 241086
## 2018 241989 240428 242396 243445 244962 247981 248620 248248 247676 246896
## 2019 244582 243281 245077 245255 246140 248562 249432 249942 249376 248515
##         Nov    Dec
## 2009 174136 174458
## 2010 176036 174442
## 2011 174812 174179
## 2012 178406 176816
## 2013 188544 188265
## 2014 203346 202856
## 2015 219582 220361
## 2016 231922 231593
## 2017 242378 241061
## 2018 246518 244641
## 2019 250410

LHP1

## # A tibble: 131 × 1
##    `162673`
##       <dbl>
##  1   160956
##  2   159340
##  3   160701
##  4   162740
##  5   164536
##  6   167673
```

```
## 7    169603
## 8    171214
## 9    172314
## 10   172818
## # ℹ 121 more rows
```

```
length(LHP1)
```

```
## [1] 1
```

```
ts.plot(data2)
```

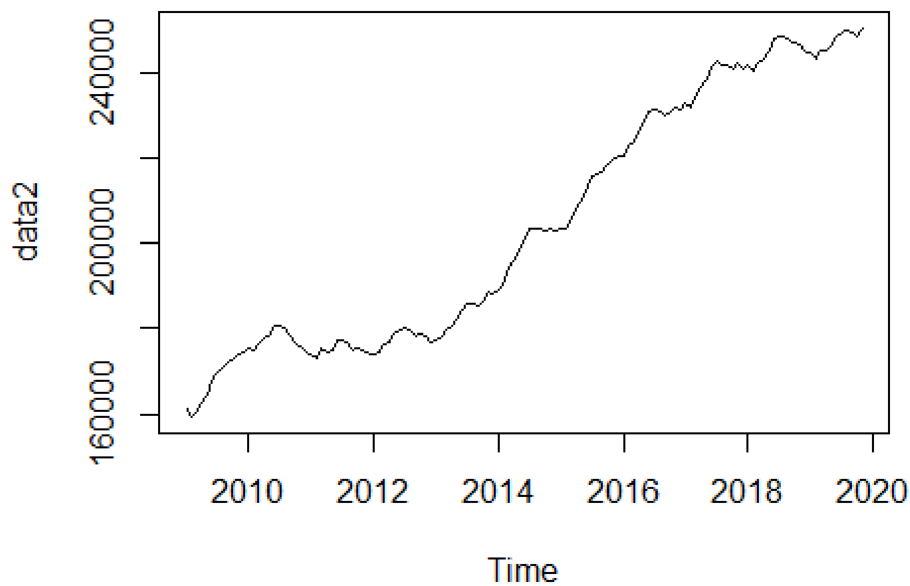


Figure 1 : Time series plot Time Vs price of houses in London

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.

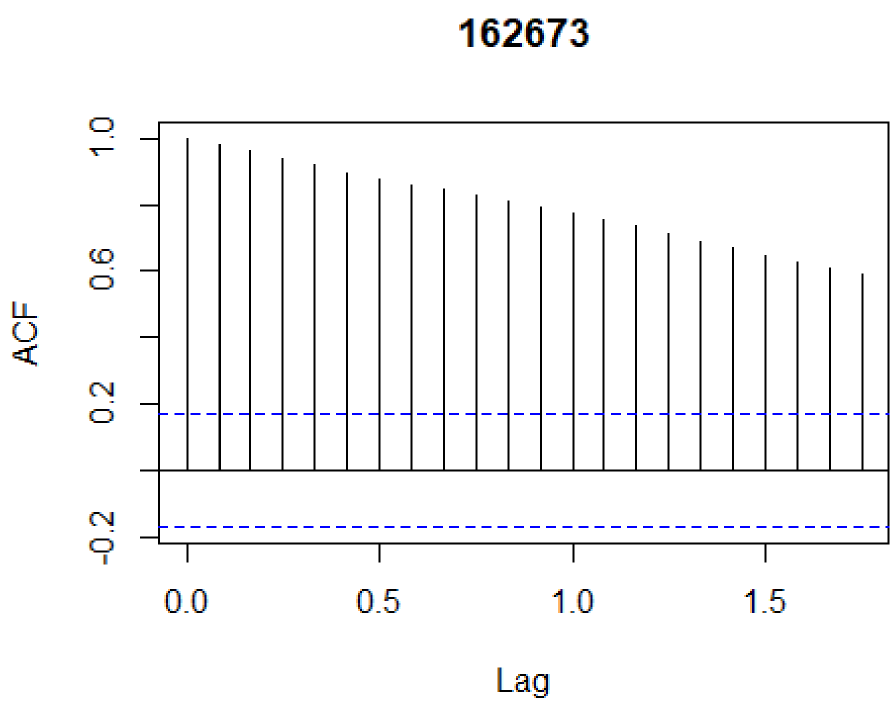
Thus, the model is of additive form with trend and irregularity component.

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 the differencing method below.

Logarithmic Transformation And Stationarity

1) We cannot use logarithmic transformation since it helps us to convert a multiplicative model into an additive model.

2) Since our model is additive, we need not use logarithmic transformation.

3) To achieve stationarity we need to use differencing method as demonstrated below.

stationary classes of time series model. (Justification)

we need to transform the data to stationary form. a polynomial degree say p form model will give a stationary model post differencing p time.

differencing method and Plotting (Demonstration)

```
diffdata=diff(data2)
diffdata

##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   D
ec
## 2009       -1616  1361  2039  1796  3137  1930  1611  1100   504  1318   3
22
## 2010   790  -483  2031   958   901  1864   288  -576 -2129 -1801  -265 -15
94
## 2011  -631  -765  2444  -822   170  2326   171  -552 -1612    29  -388  -6
33
## 2012   -18   162  2220   483  1670  1060   373  -566 -1151   250  -256 -15
90
## 2013   387   986  1711   721  1467  2186  1368   440  -724   902  2284  -2
79
## 2014  1082   690  4214  1920  1780  2874  2581   233  -328  -607   642  -4
90
## 2015   568   -64  2576  2329  1609  3644  2238   594   326  1824  1082   7
79
## 2016   266  2036  1121  2586  2060  2438   308  -328  -904  1109   869  -3
29
## 2017  1103  -936  3261  1706  1868  2811  1222  -587   -38  -917  1292 -13
17
## 2018   928 -1561  1968  1049  1517  3019   639  -372  -572  -780  -378 -18
77
## 2019   -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
```

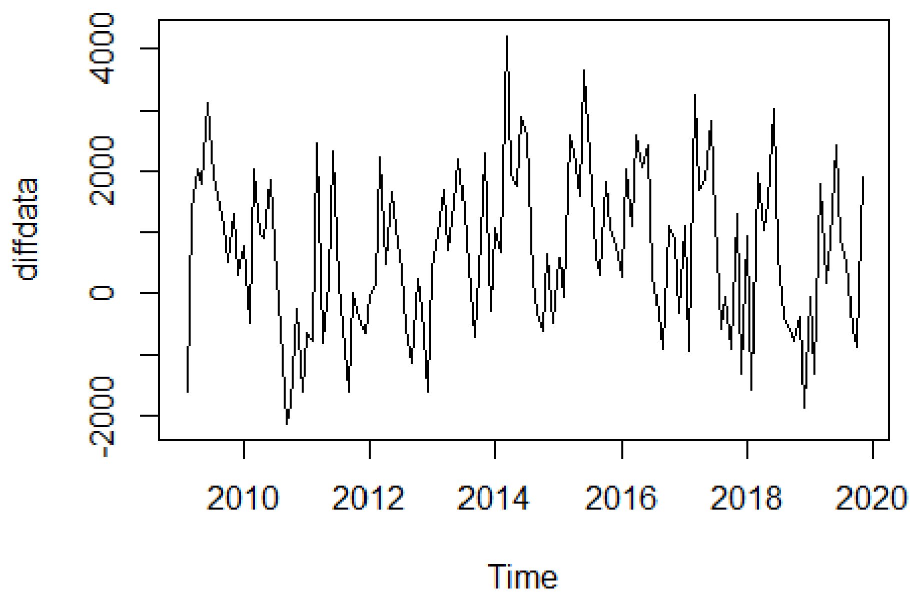


Figure 2: Time Vs 1st Differential Data (To Observe Stationarity)

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 = -6.2648, 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.

Conclusion:

1)Thus the data follows a time series model of the form:

z(t)=m(t)+e(t)
where, z(t)-time series variable dependent on time,
m(t)-trend component
e(t)-irregularity component.

2) There is no stationarity observed in the Acf plot of the time series data set. However, the first differen cing method did yield a stationary plot and it was confirmed using an augmented dicky fuller test.

3)Logarithmic transformation was not used since the model was of an additive type.

The End