Final Project Report

Shrusti Ghela

6/2/2022

Introduction:

La Pino'z Pizza is India's fastest growing and the most popular Pizza chain. The first pizzeria opened up in Chandigarh in 2011 and now there are more than 350 outlets across India. A few of these outlets belong to my father. Even though this is a very up and coming, new generation Pizza place in India, I have seen first hand how the business works - the same old traditional way. This project is motivated out of personal curiosity and a desire to suggest that these businesses can profit a lot more if they analyze the data that is generated on a daily basis and use the insights from this analysis in a fruitful way. "Currently, few restaurants use R or any other readily available statistical software to analyze their data. Instead many use subscriptions to services that may preform basic analysis such as averaging and then display that information in basic, though visually appealing visualizations." [1.] This is exactly what I want to fix by the medium of this project.

Two of the most important aspects in functioning of any restaurant is number of visitors and total sales. Forecasting the sales and number of visitors is an essential component in the decision-making process for the owners of these restaurants. For this reason, I try to investigate the performance of the sales and visitors for 2 different restaurants of the same La Pino'z franchise. It aims at analyzing the time series models of sales for at least 2 restaurants and number of visitors for at least 2 restaurants.

I am interested in this project because I have seen this business function very first hand and if by the concepts learnt in this class, there is something that I could add to it that helps the business grow, I would be happy to help my family business.

I aim to work on the analysis of Time Series (daily) data for Sales and number of visitors of 2 restaurants of the La Pino'z franchise (at Kalol, Gujarat and Mehsana, Gujarat) with data from March 2021 to May 2022.

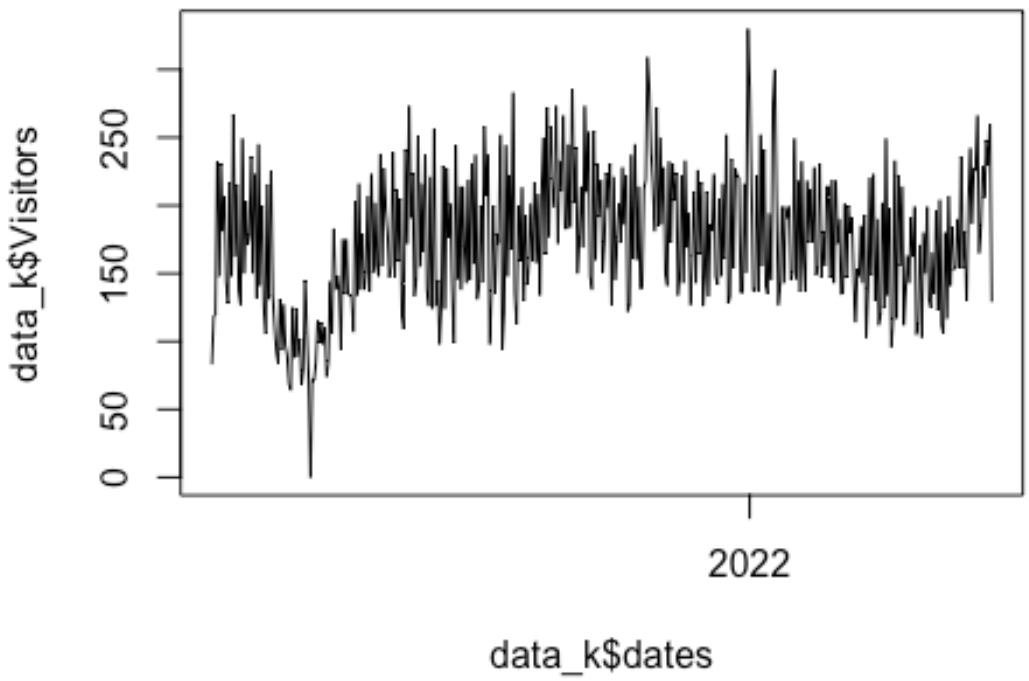
I recieved the relevant data from the owners of these restaurants. This data is not publicly available. I have obtained permission to use this data for this project.

Exploratory Analysis:

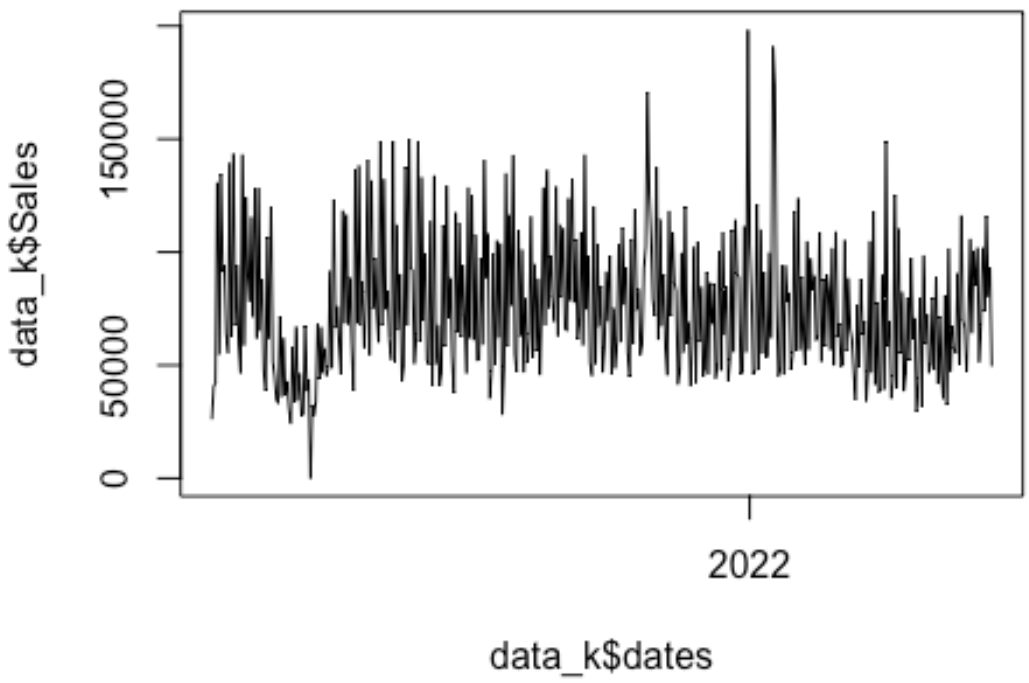
```
# data for the restaurant in Kalol
data_k <- read.csv("~/Downloads/Kalol.csv")
head(data_k)
```

```
##         Date Visitors  Sales
## 1 2021-03-07       83  26039
## 2 2021-03-08      118  40542
## 3 2021-03-09      120  42043
```

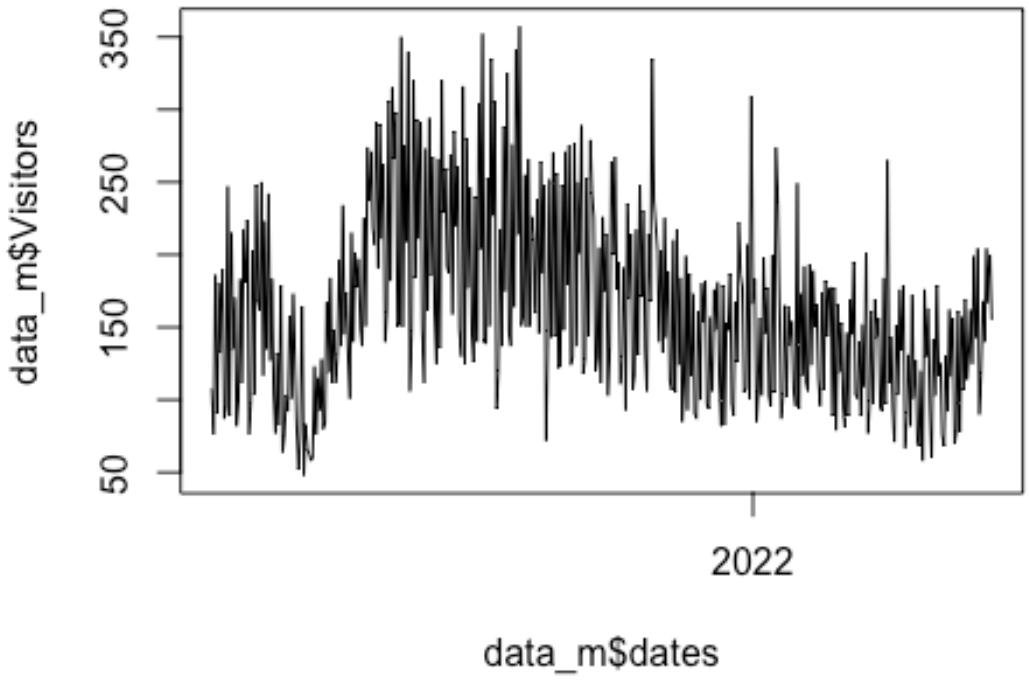
```
## 4 2021-03-10      232 130706
## 5 2021-03-11      148  54828
## 6 2021-03-12      231 134485

# data for the restaurant in Mehsana
data_m <- read.csv("~/Downloads/Mehsana.csv")
head(data_m)

##         Date Visitors  Sales
## 1 2021-03-01      108  45361
## 2 2021-03-02       76  32053
## 3 2021-03-03      186 129386
## 4 2021-03-04       91  38132
## 5 2021-03-05      180 115446
## 6 2021-03-06      133  54881

# converting dates to date-datatype for Kalol
data_k <- cbind(data_k, dates = ymd(data_k$Date))

# converting dates to date-datatype for Mehsana
data_m <- cbind(data_m, dates = ymd(data_m$Date))

plot(data_k$dates, data_k$Visitors, type='l')
```



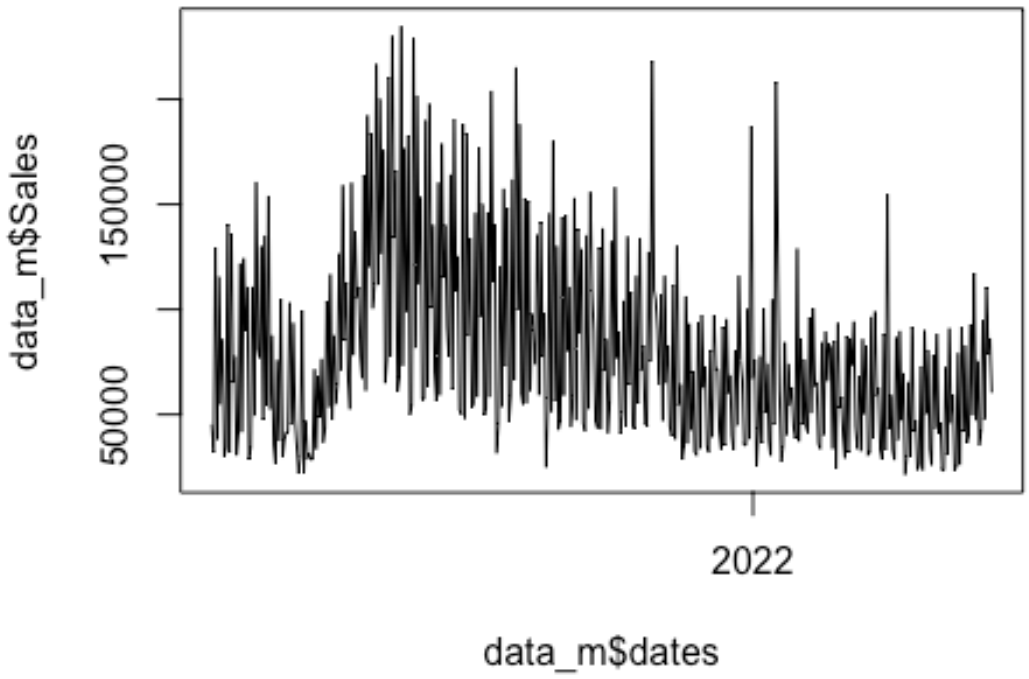

```
plot(data_k$dates, data_k$Sales, type='l')
```




```
plot(data_m$dates, data_m$Visitors, type='l')
```



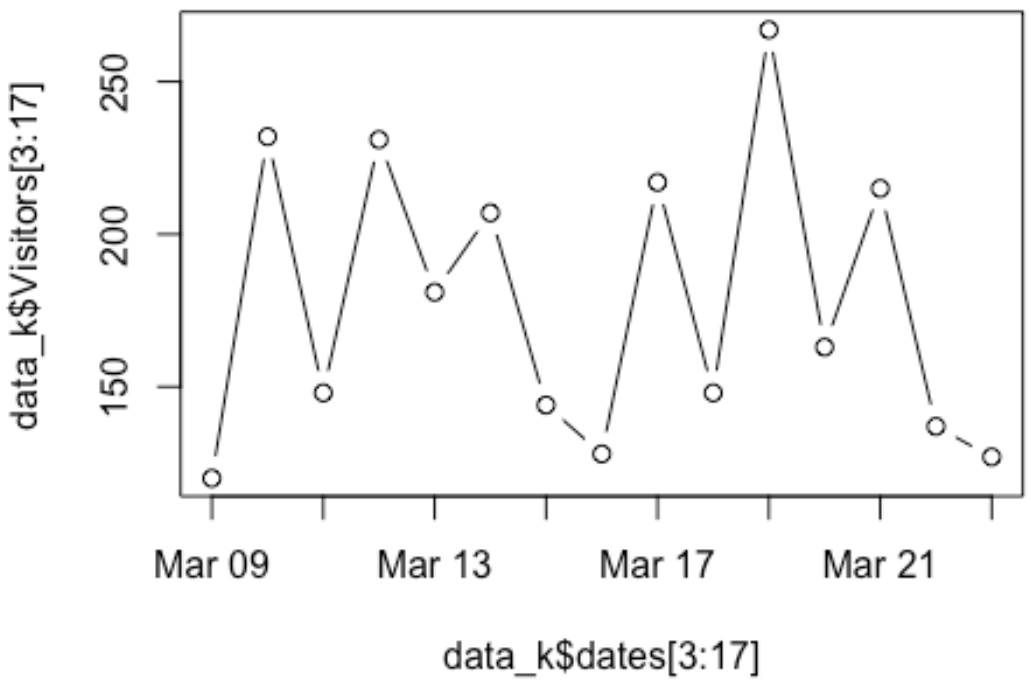

```
plot(data_m$dates, data_m$Sales, type='l')
```

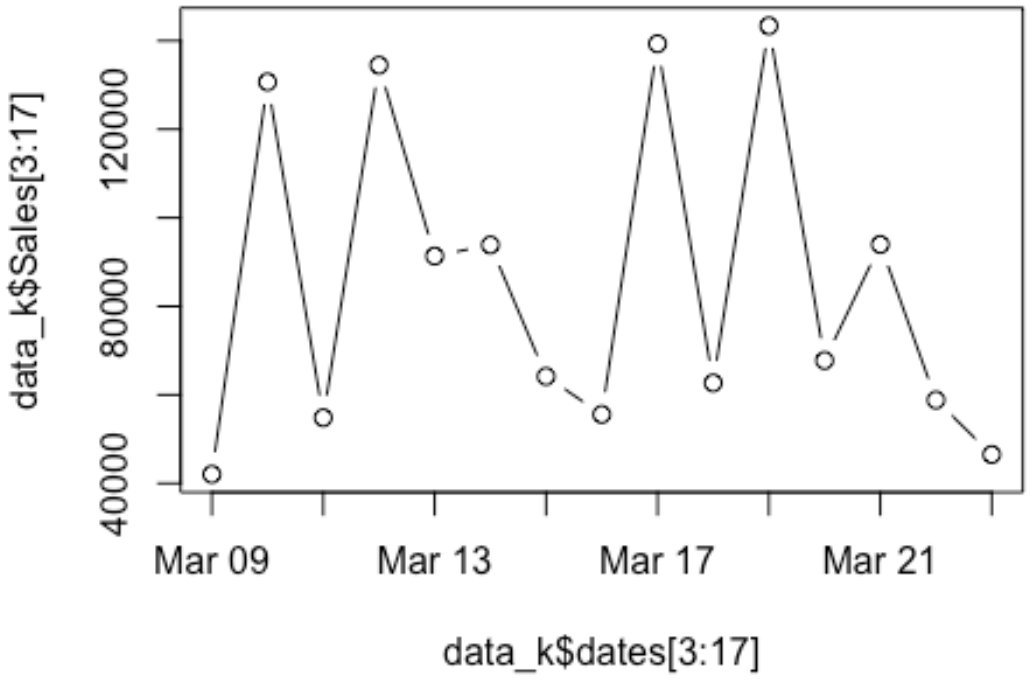


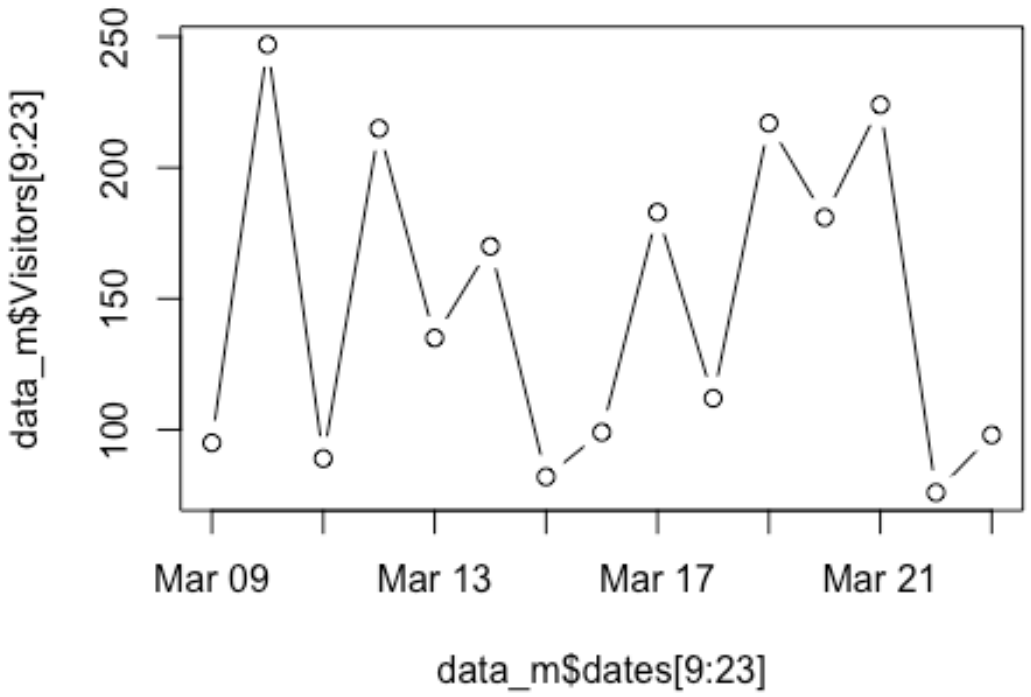
On eyeballing these series, there are a few things that I observe:

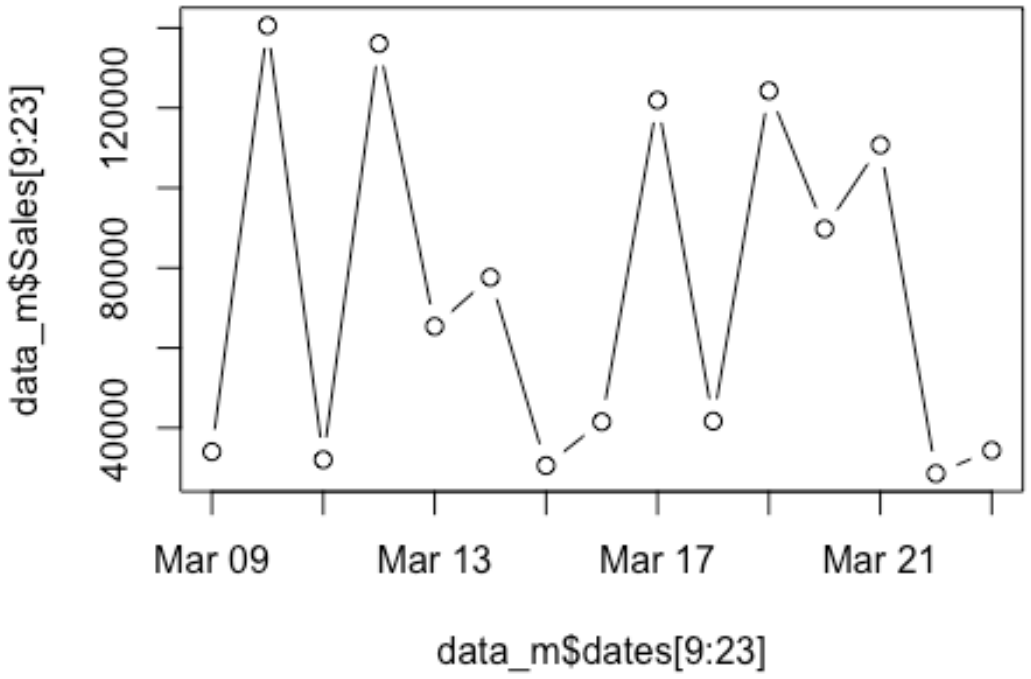
- There seems to be some sort of (weekly) seasonality. This is also intuitive becasue the data consists of restaurants' visitors and sales, and there would be difference in visitors and sales depending on the day of the week.
- The restaurant has offers like "Buy One Get One (BOGO)" free on Pizzas on every Wednesdays and Fridays. So, we expect to have more visitors and sales during those days.
- One other observation is that we see a dip in sales and visitors in April 2021. This might be because of the fact that India faced second wave of COVID and saw an increase in COVID cases. This affected the restaurants heavily too. But as the wave subsided, we see a heavy boom again.

Let's just zoom in to a particular time frame so that we could verify our seasonality observation a little better


data_k$Visitors[3:17]
150
200
250
Mar 09
Mar 13
Mar 17
Mar 21
data_k$dates[3:17]


data_k$Sales[3:17]
40000
80000
120000
Mar 09
Mar 13
Mar 17
Mar 21
data_k$dates[3:17]

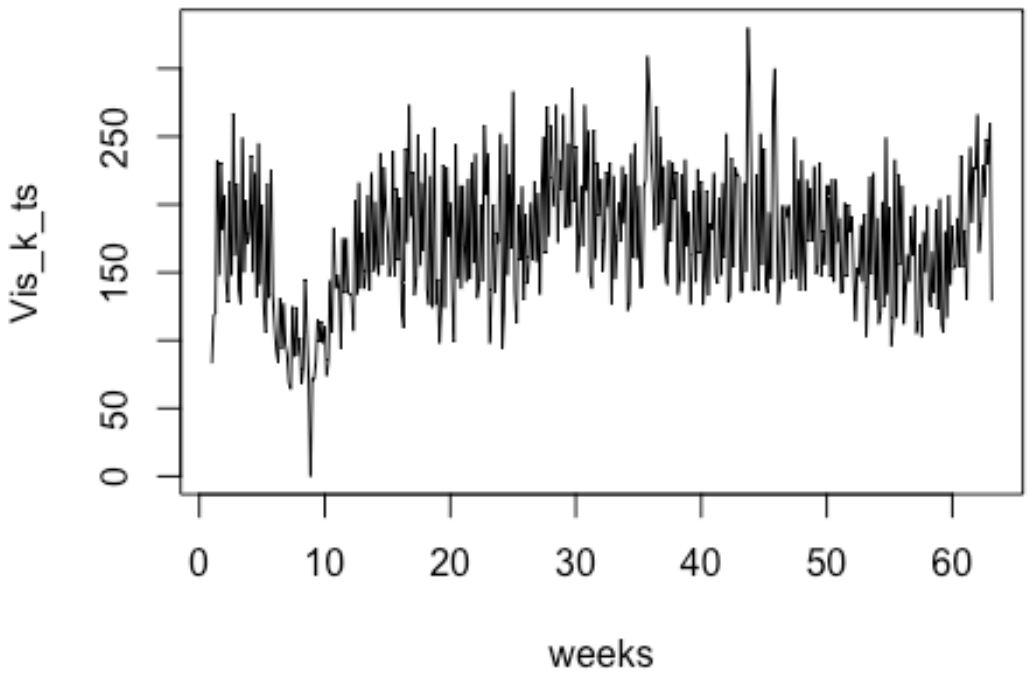

data_m$Visitors[9:23]
100
150
200
250
Mar 09
Mar 13
Mar 17
Mar 21
data_m$dates[9:23]


data_m$Sales[9:23]
40000
80000
120000
Mar 09
Mar 13
Mar 17
Mar 21
data_m$dates[9:23]

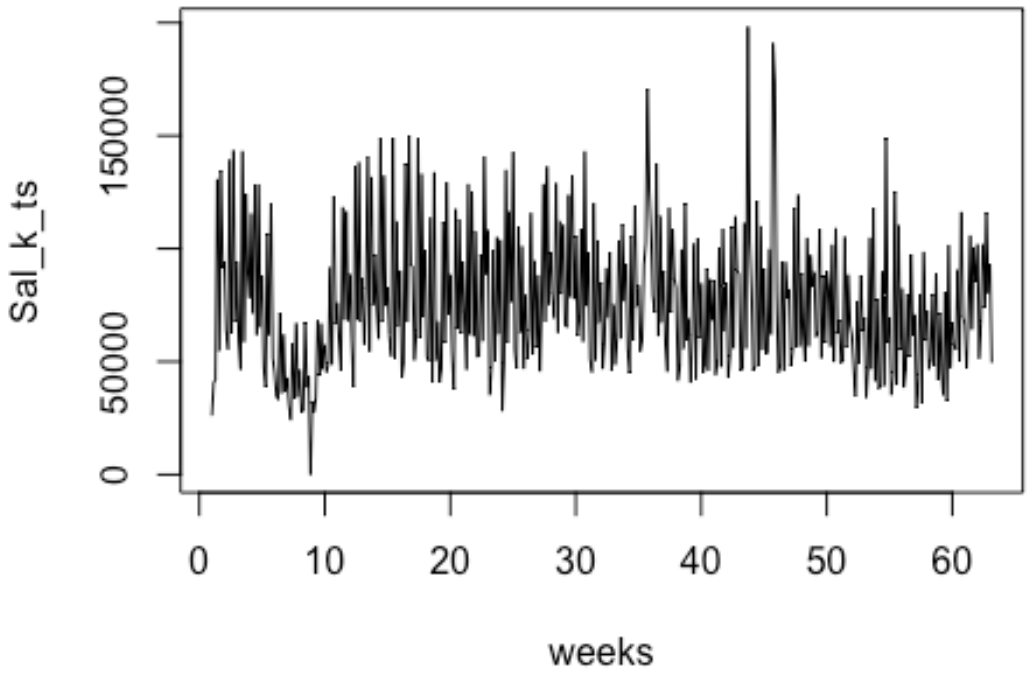
From the above plots, we see that there is some sort of seasonality. Some sort of behavior repeats every seven observations. So, we observe a weekly seasonality that we suspected above. I just zoomed in to a particular section to verify that we need to consider a seasonality while modeling and forecasting for this time-series which we know because of the domain-knowledge.

I also think that there has to be very high correlation between the number of visitors and total sales.

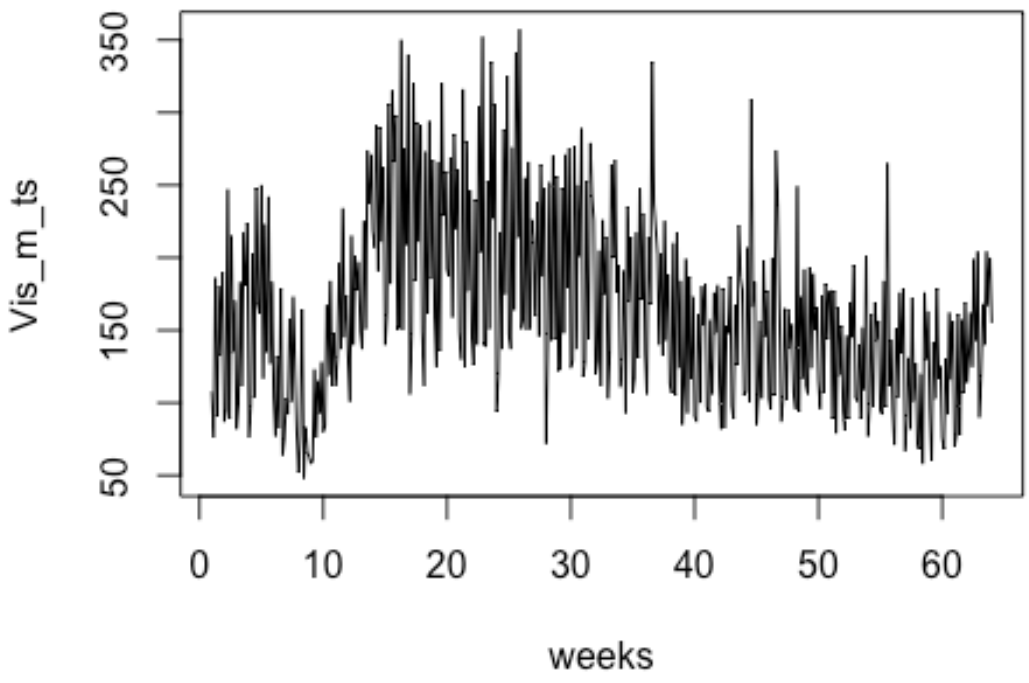
```
#time-series datatype

#considering frequency 7 because of the seasonal behaviour
Vis_k_ts <- ts(data_k$Visitors, freq=7, start=c(1))
Sal_k_ts <- ts(data_k$Sales, freq=7, start=c(1))

plot(Vis_k_ts, xlab='weeks')
```



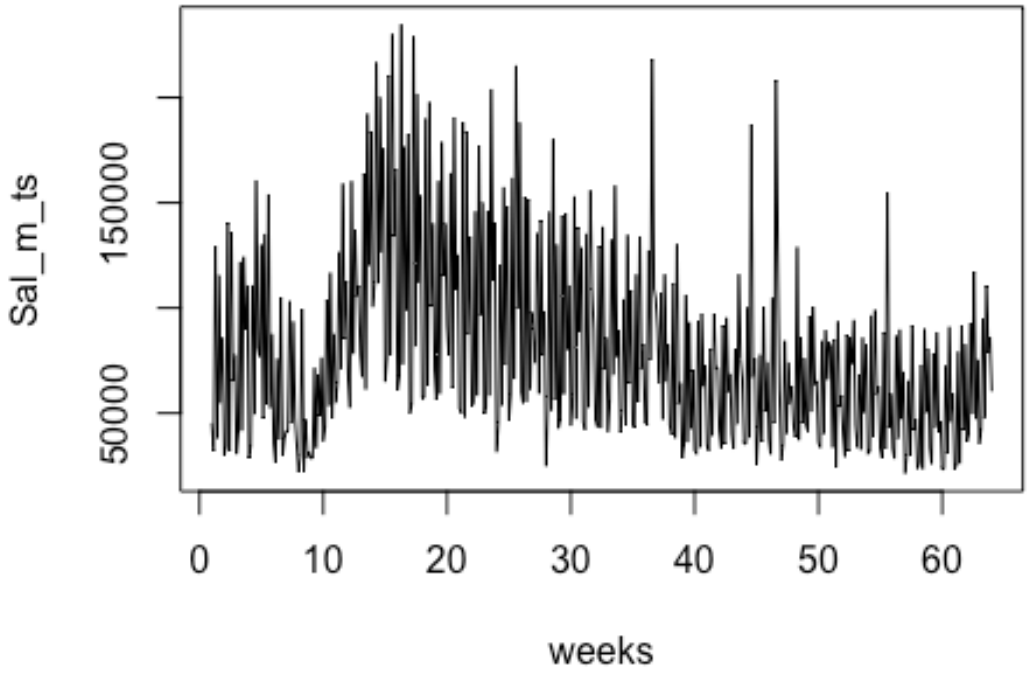

```
plot(Sal_k_ts, xlab='weeks')
```




```r
#time-series datatype

#considering frequency 7 because of the seasonal behaviour
Vis_m_ts <- ts(data_m$Visitors, freq=7, start=c(1))
Sal_m_ts <- ts(data_m$Sales, freq=7, start=c(1))

plot(Vis_m_ts, xlab='weeks')
```

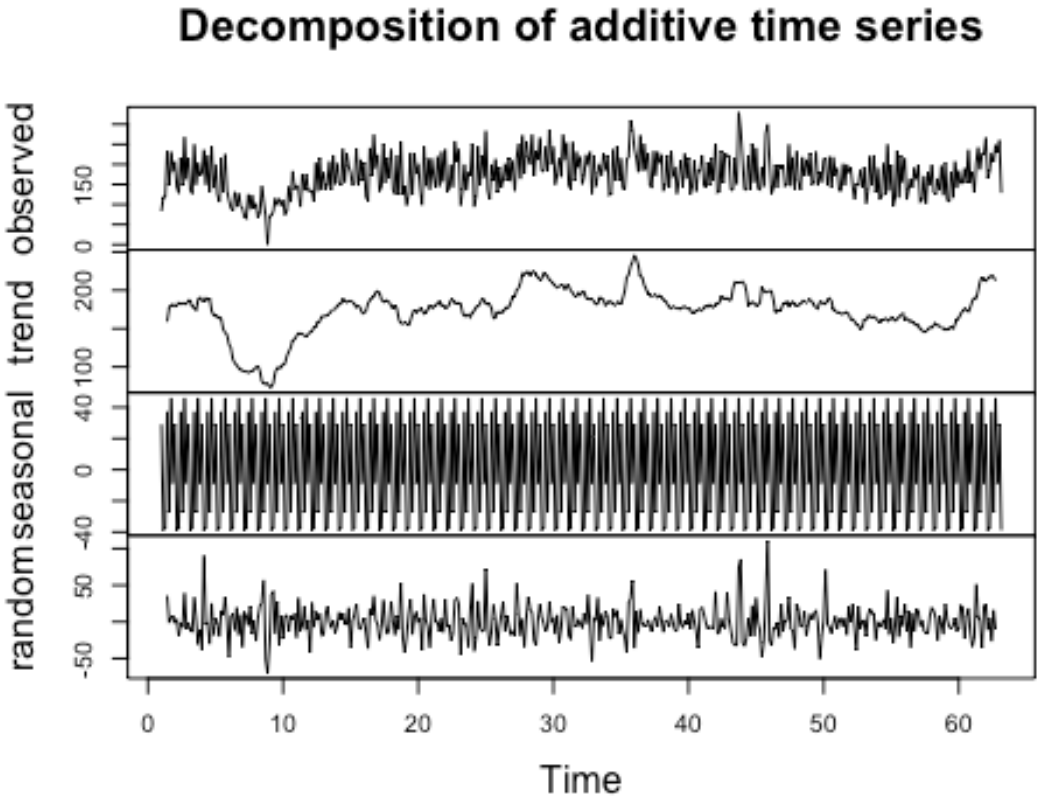



```
plot(Sal_m_ts, xlab='weeks')
```

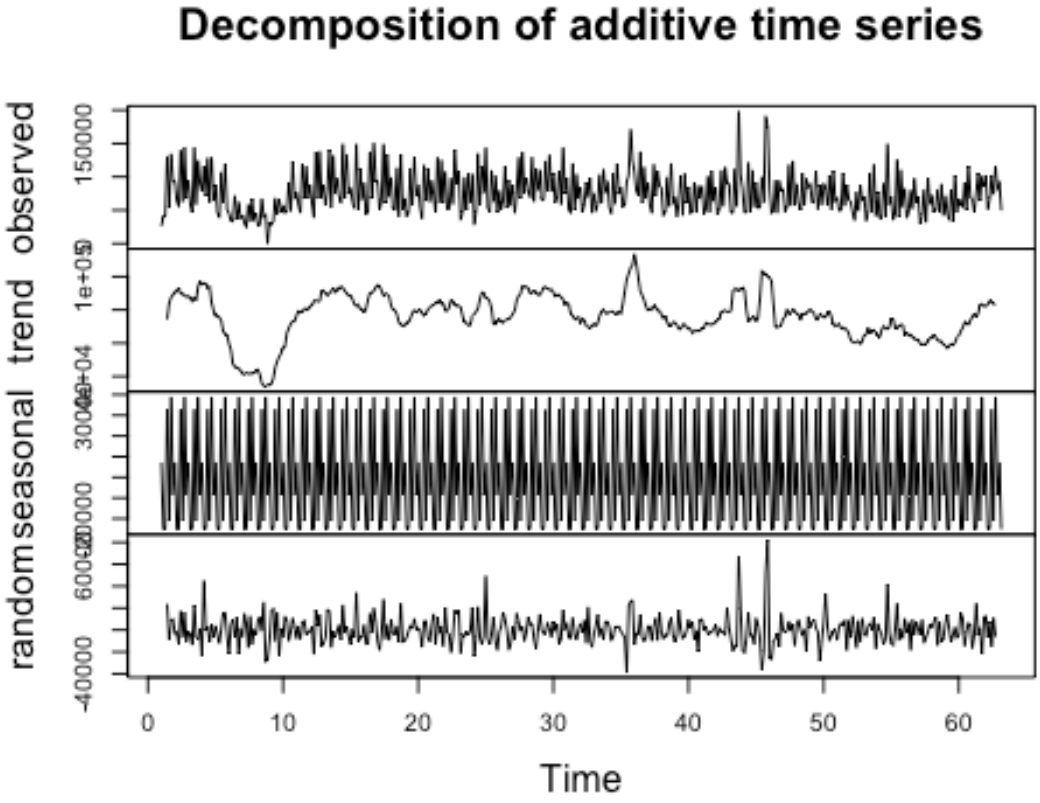


Let us see the trends and seasonality by decomposing the time-series into these components.

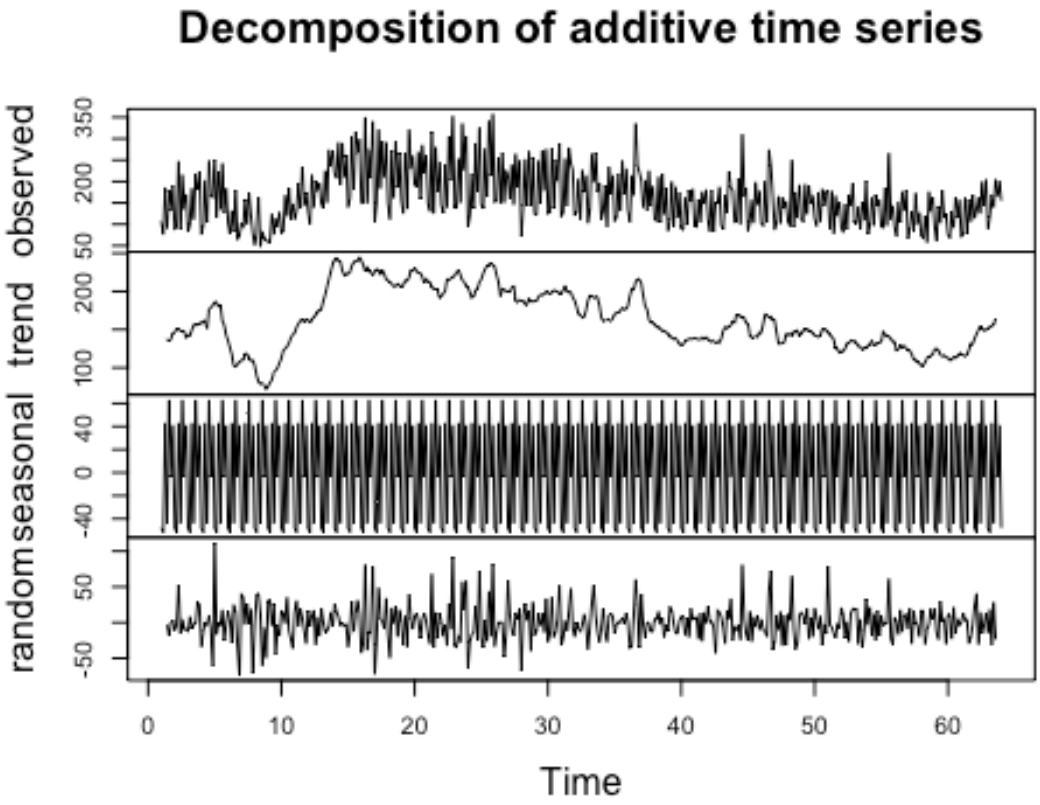
```
plot(decompose(Vis_k_ts))
```



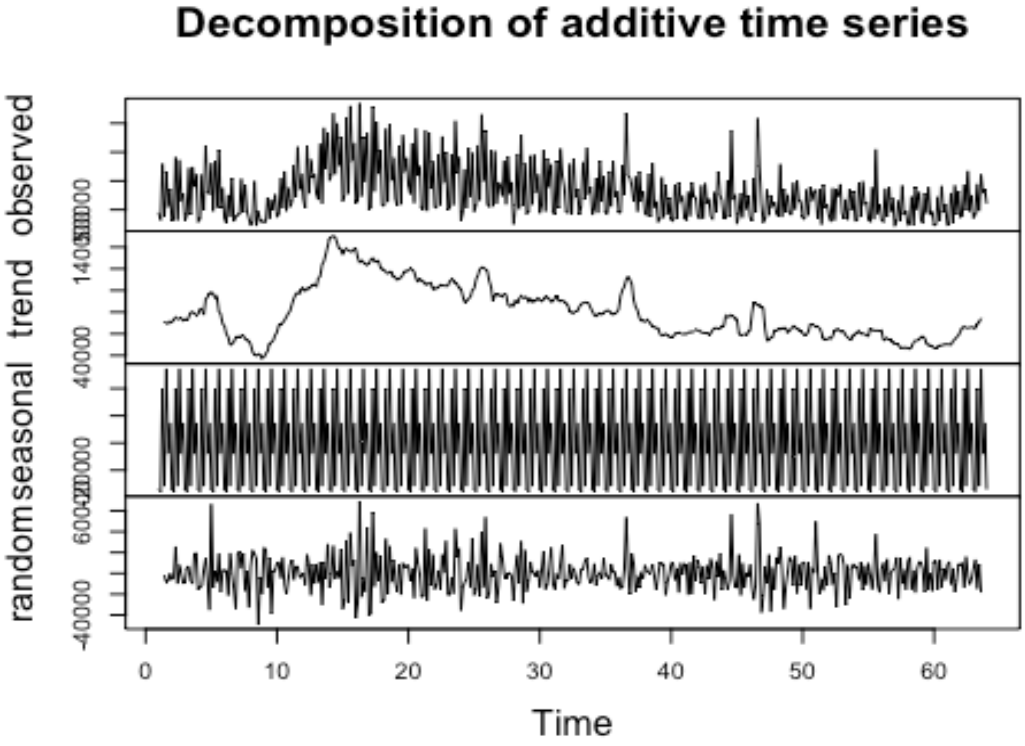

```
plot(decompose(Sal_k_ts))
```



```
plot(decompose(Vis_m_ts))
```




```
plot(decompose(Sal_m_ts))
```



We verify that there is some sort of seasonality and trend. And hence, non-stationarity involved in these time-series. Further verification could be done using kpss test.

KPSS test H_0 : Series is stationary H_α :Series is NOT stationary

```
#Hypothesis testing for stationarity
kpss.test(Vis_k_ts)
```

```
## Warning in kpss.test(Vis_k_ts): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  Vis_k_ts
## KPSS Level = 1.2036, Truncation lag parameter = 5, p-value = 0.01
```

p-value = 0.01, p-value < 0.05, $\implies$ we reject the null-hypothesis

```
#Hypothesis testing for seasonally differenced series for stationarity
kpss.test(diff(Vis_k_ts, lag=7))
```

```
## Warning in kpss.test(diff(Vis_k_ts, lag = 7)): p-value greater than printe
d p-
## value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(Vis_k_ts, lag = 7)
## KPSS Level = 0.050822, Truncation lag parameter = 5, p-value = 0.1
```

The seasonaly differenced time-series has p-value 0.1 which implies we fail to reject the null-hypothesis.

```
#Hypothesis testing for stationarity
kpss.test(Sal_k_ts)
```

```
## Warning in kpss.test(Sal_k_ts): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  Sal_k_ts
## KPSS Level = 0.32616, Truncation lag parameter = 5, p-value = 0.1
```

This shows that our Sales time-series might be stationary. But on eyeballing I don't observe it to be stationary. And observing the decomposed time-series also says that the time-series is not stationary. I am unsure as to why this is happening. So, for this project I am not working with this series. I need to read up on this to further work on this series.

```
#Hypothesis testing for stationarity
kpss.test(Vis_m_ts)
```

```
## Warning in kpss.test(Vis_m_ts): p-value smaller than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  Vis_m_ts
## KPSS Level = 1.7422, Truncation lag parameter = 5, p-value = 0.01
```

p-value = 0.01, p-value < 0.05, $\Longrightarrow$ we reject the null-hypothesis

```
#Hypothesis testing for seasonally differenced series for stationarity
kpss.test(diff(Vis_m_ts, lag=7))

## Warning in kpss.test(diff(Vis_m_ts, lag = 7)): p-value greater than printe
d p-
## value

##
##  KPSS Test for Level Stationarity
##
## data:  diff(Vis_m_ts, lag = 7)
## KPSS Level = 0.072539, Truncation lag parameter = 5, p-value = 0.1
```

The seasonaly differenced time-series has p-value 0.1 which implies we fail to reject the null-hypothesis.

```
kpss.test(Sal_m_ts)

## Warning in kpss.test(Sal_m_ts): p-value smaller than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  Sal_m_ts
## KPSS Level = 2.4692, Truncation lag parameter = 5, p-value = 0.01
```

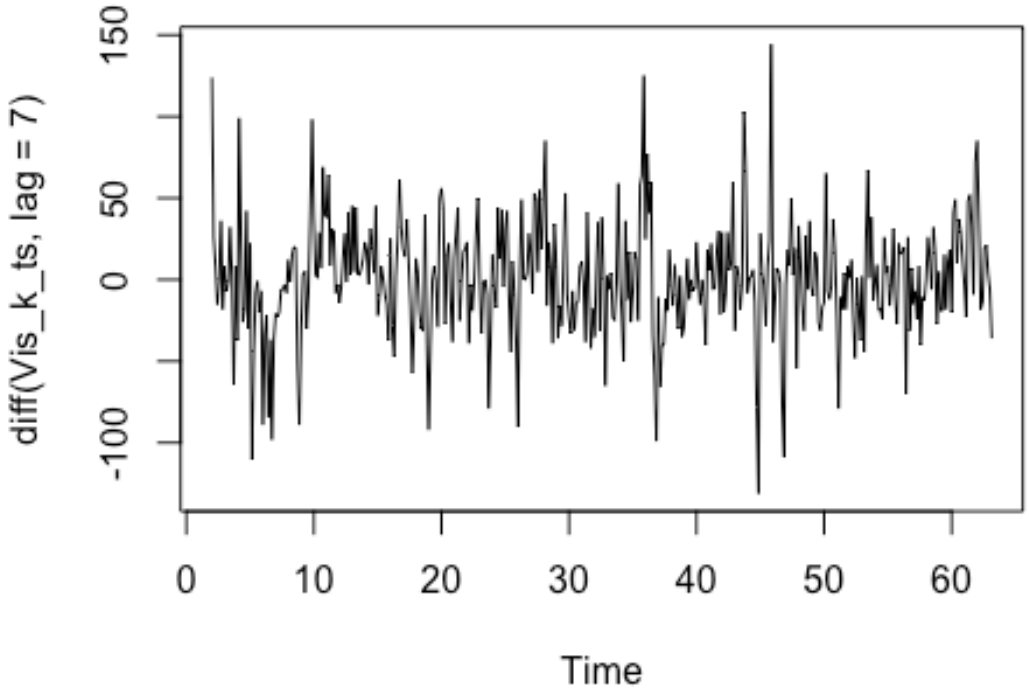
p-value = 0.01, p-value < 0.05, $\Longrightarrow$ we reject the null-hypothesis

```
#Hypothesis testing for seasonally differenced series for stationarity
kpss.test(diff(Sal_m_ts, lag=7))

## Warning in kpss.test(diff(Sal_m_ts, lag = 7)): p-value greater than printe
d p-
## value

##
##  KPSS Test for Level Stationarity
##
## data:  diff(Sal_m_ts, lag = 7)
## KPSS Level = 0.064598, Truncation lag parameter = 5, p-value = 0.1
```

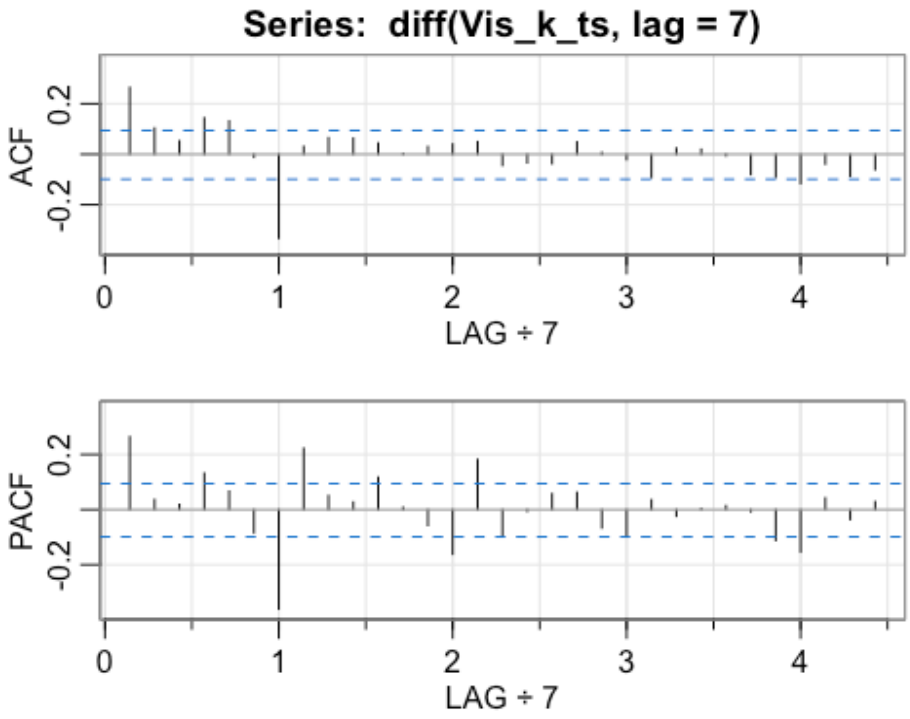
The seasonaly differenced time-series has p-value 0.1 which implies we fail to reject the null-hypothesis.

ARIMA Modelling:

```
plot(diff(Vis_k_ts, lag=7))
```




```
acf2(diff(Vis_k_ts, lag=7))
```

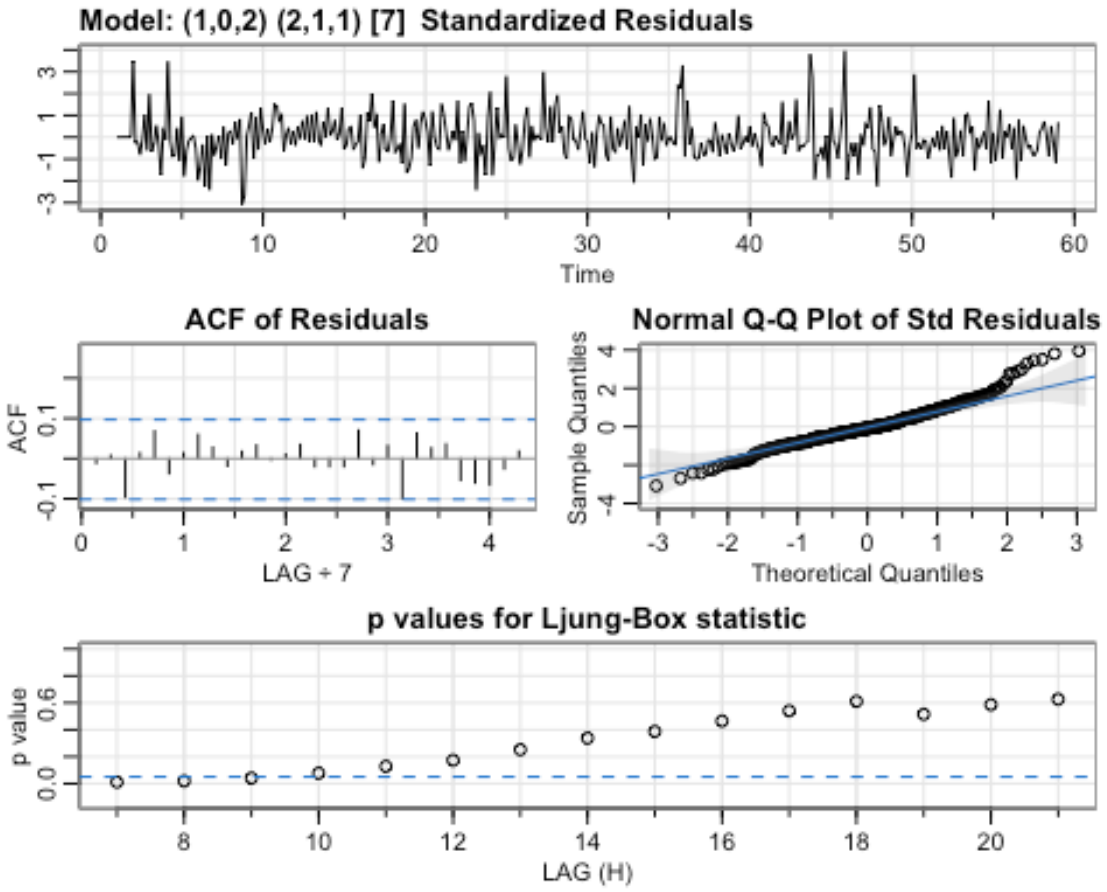


For ACF, We see 2 significant non-seasonal lag and 1 significant seasonal lag. We also see some significant non-seasonal lag closer to the seasonal lag. This might be due to the first significant seasonal lag.

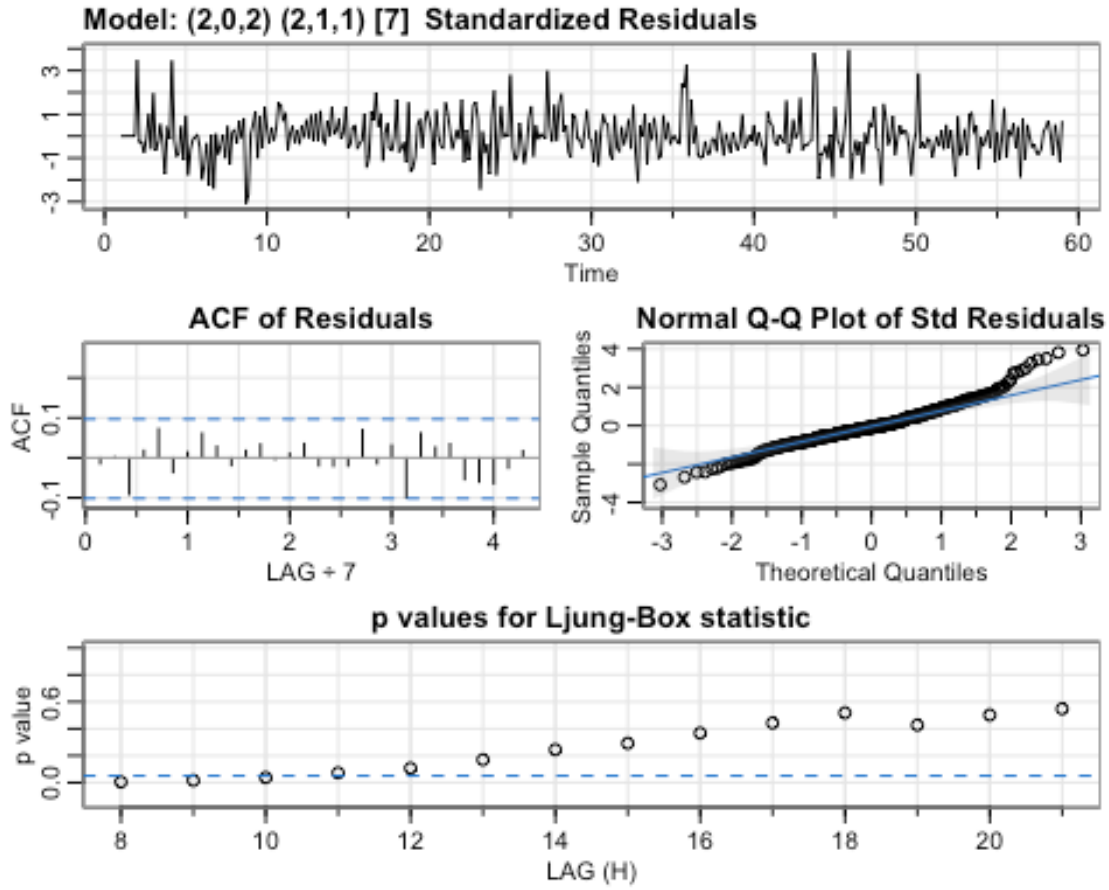
For PACF, We see 1 significant non-seasonal lag and 2 significant seasonal lag. We also see some significant non-seasonal lag closer to the seasonal lag. This might be due to the first significant seasonal lag.

So, for Visitor time-series we might want to model our series somewhere around (1,0,2)(2,1,1)[7] from the ACF, PACF and the hypothesis tests. I will model multiple sarima to check which model has better results.

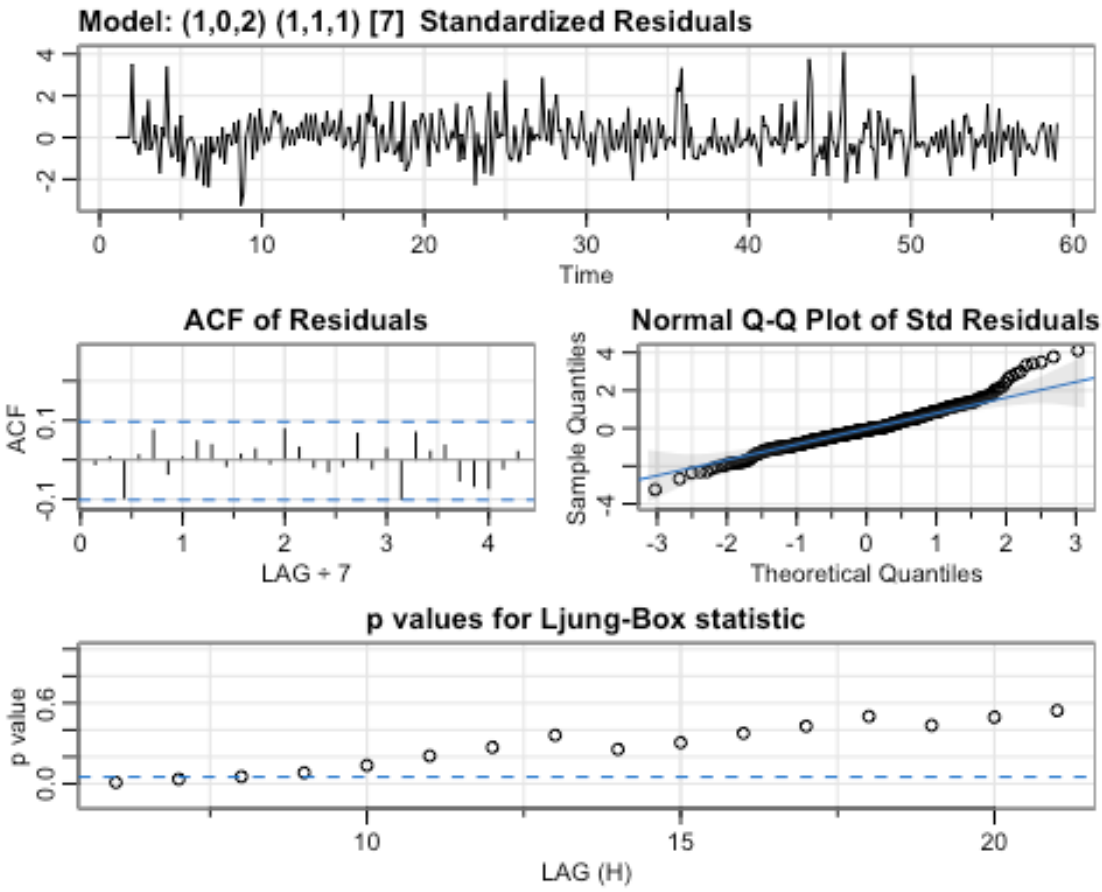
```
Vis_k_train = window(Vis_k_ts, end = c(59, 1))
Vis_k_test = window(Vis_k_ts, start = c(59,2))
Vis_m_train = window(Vis_m_ts, end = c(60, 1))
Vis_m_test = window(Vis_m_ts, start = c(60,2))
Sal_m_train = window(Sal_m_ts, end = c(60, 1))
Sal_m_test = window(Sal_m_ts, start = c(60,2))

Vis_k_m1 = sarima(Vis_k_train, S=7,
                  p=1, d=0, q=2,
                  P=2, D=1, Q=1)
```



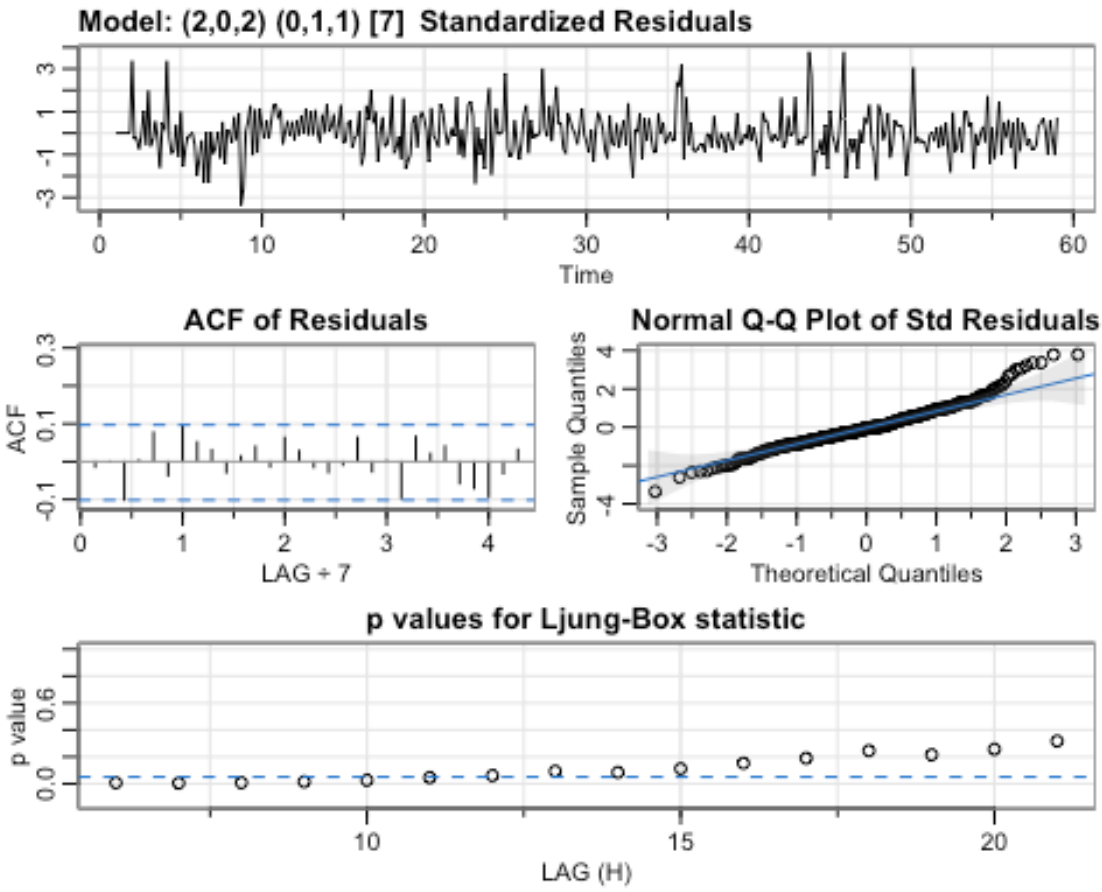
```
Vis_k_m2 = sarima(Vis_k_train, S=7,
                  p=2, d=0, q=2,
                  P=2, D=1, Q=1)
```



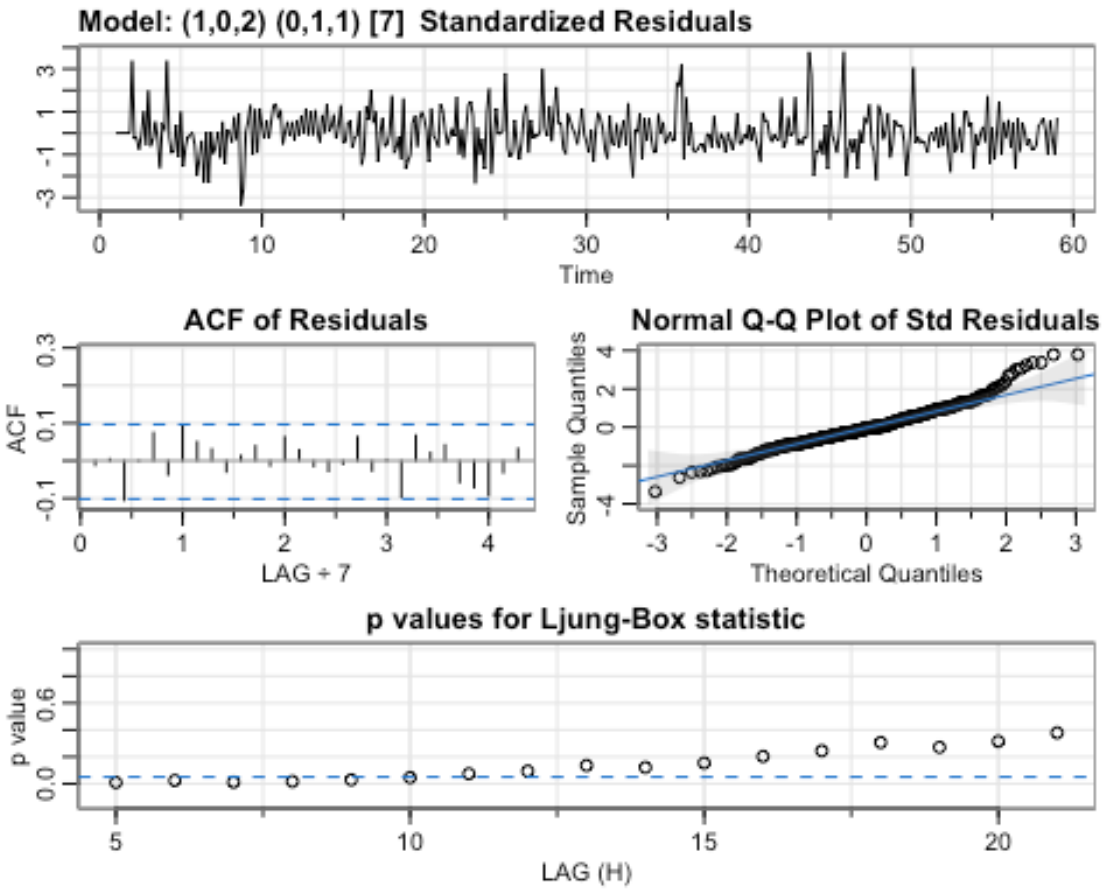

```
Vis_k_m3 = sarima(Vis_k_train, S=7,
                  p=1, d=0, q=2,
                  P=1, D=1, Q=1)
```




```
Vis_k_m4 = sarima(Vis_k_train, S=7,
                  p=2, d=0, q=2,
                  P=0, D=1, Q=1)
```




```
Vis_k_m5 = sarima(Vis_k_train, S=7,
                  p=1, d=0, q=2,
                  P=0, D=1, Q=1)
```

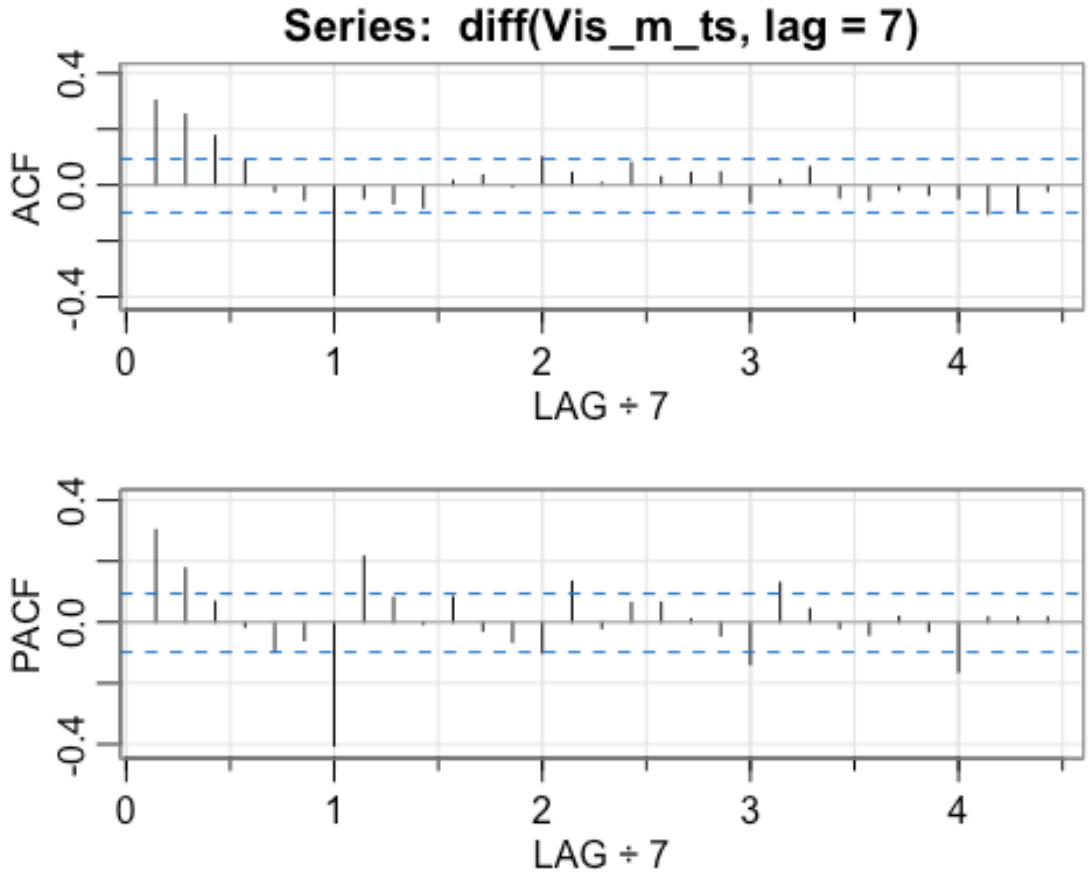



```
Vis_k_m1$AICc

## [1] 9.493664

Vis_k_m2$AICc

## [1] 9.498498

Vis_k_m3$AICc

## [1] 9.495198

Vis_k_m4$AICc

## [1] 9.504271

Vis_k_m5$AICc

## [1] 9.499406
```

The AICc of all the models are almost similar. But the Ljung-box of model 3 (1,0,2)(1,1,1)[7] looks fairly better when compared with other models. Its still not satisfactory enough, because for initial lags, the p-value are close to the significant levels.

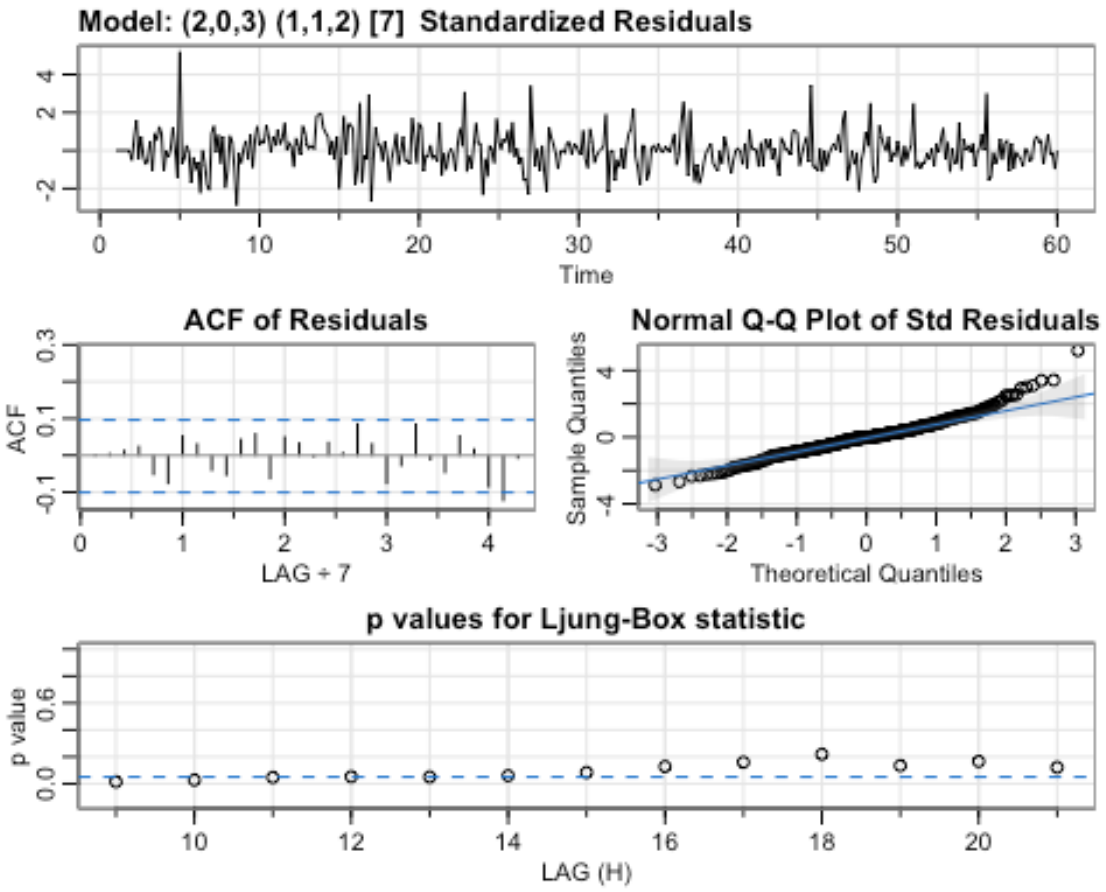
Furthermore, for the Normal Q-Q Plot of Std Residuals, my interpretation would be that the middle values of the sample are close to what you'd expect from normally distributed data, as it follows the straight line from the diagram closely. However, it seems the underlying data distribution presents extreme values more often than a normal one, that's why we see the points going under the line for big negative values and over it for big positive ones.

```
acf2(diff(Vis_m_ts, lag=7))
```

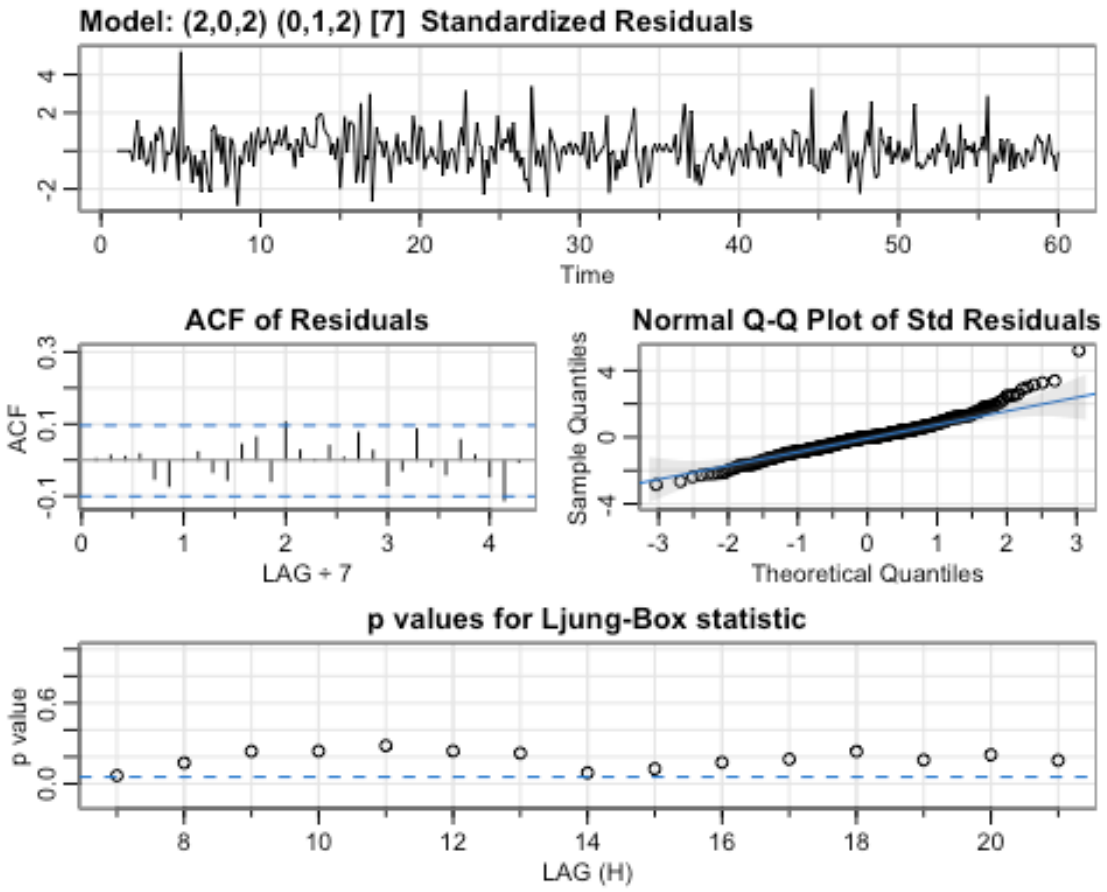


For seasonally differenced time seires, according to the ACF and PACF graphs, the model sould be close to (2,0,3)(1,1,2)[7]. I will plot multiple models as to compare the AICc and the Ljung-Box statistic to see which model outperforms the rest.

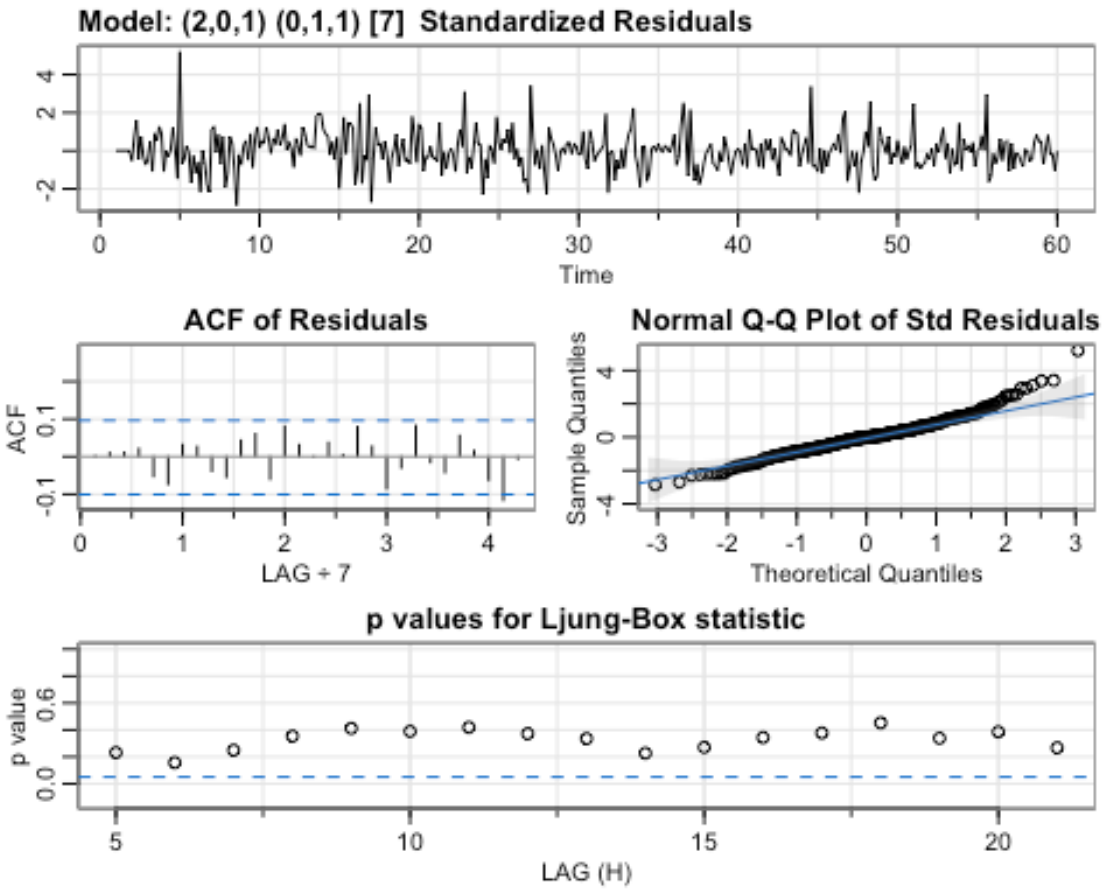
```
Vis_m_m1 = sarima(Vis_m_train, S=7,
                  p=2, d=0, q=3,
                  P=1, D=1, Q=2)
```



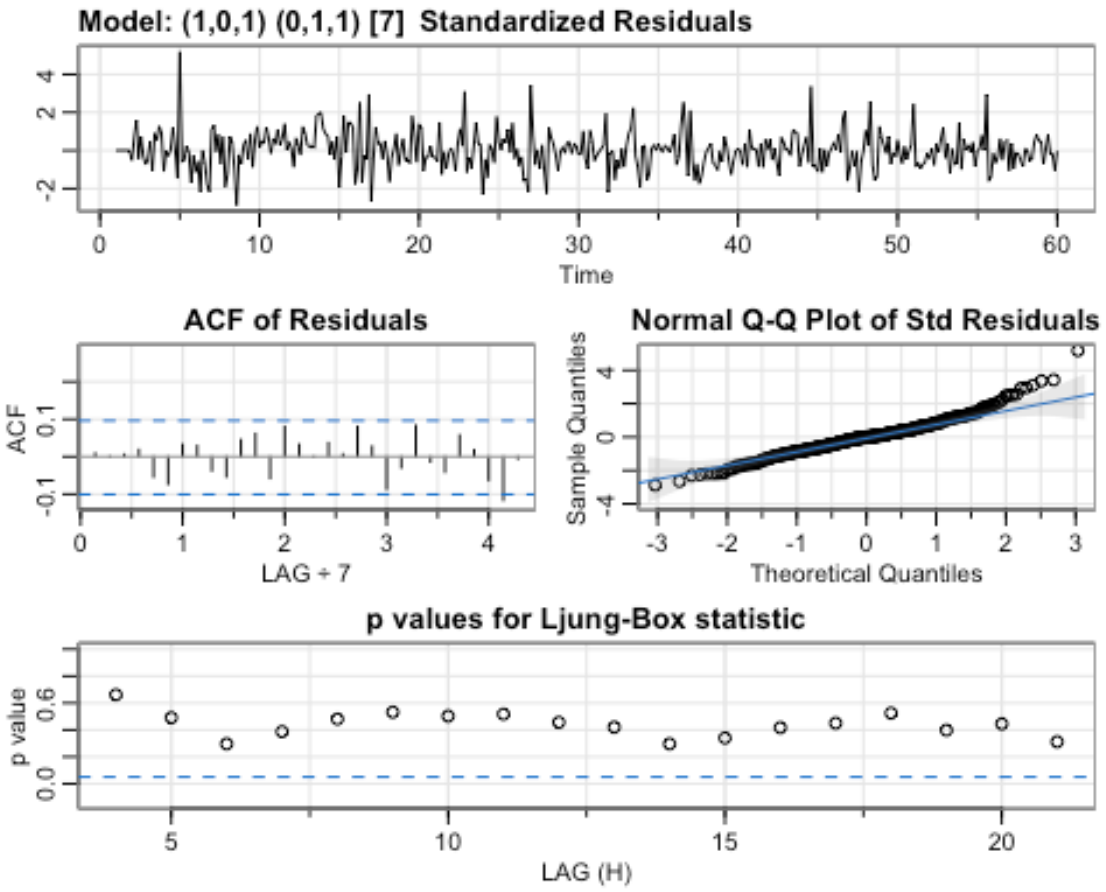

```
Vis_m_m2 = sarima(Vis_m_train, S=7,
                  p=2, d=0, q=2,
                  P=0, D=1, Q=2)
```




```
Vis_m_m3 = sarima(Vis_m_train, S=7,
                  p=2, d=0, q=1,
                  P=0, D=1, Q=1)
```




```
Vis_m_m4 = sarima(Vis_m_train, S=7,
                  p=1, d=0, q=1,
                  P=0, D=1, Q=1)
```



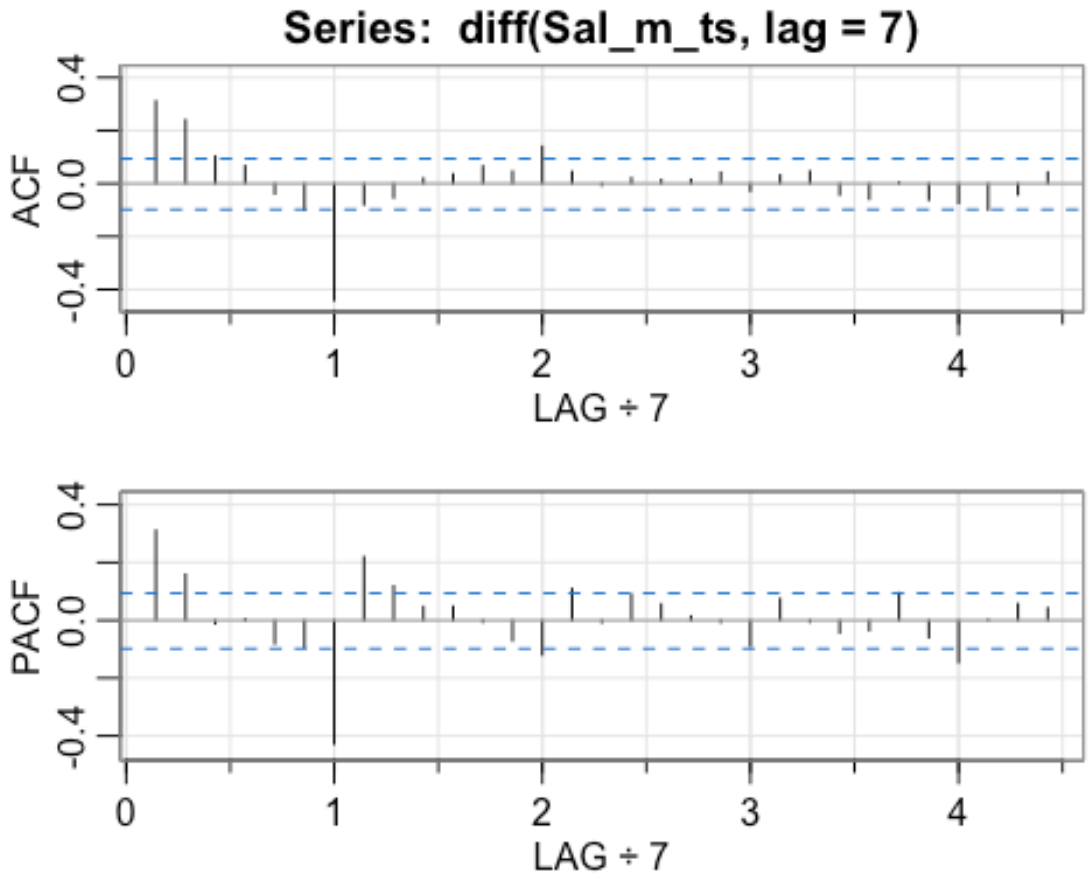

```
Vis_m_m1$AICc

## [1] 9.758893

Vis_m_m2$AICc

## [1] 9.748644

Vis_m_m3$AICc

## [1] 9.740514

Vis_m_m4$AICc

## [1] 9.735671
```

The model 4 (1,0,1)(0,1,1)[7] has a satifactory Ljung-Box statistic and outperforms the other models when compared.

Furthermore, for the Normal Q-Q Plot of Std Residuals, my interpretation would be that the middle values of the sample are close to what you'd expect from normally distributed data, as it follows the straight line from the diagram closely. However, it seems the underlying data distribution presents extreme values more often than a normal one, that's why we see

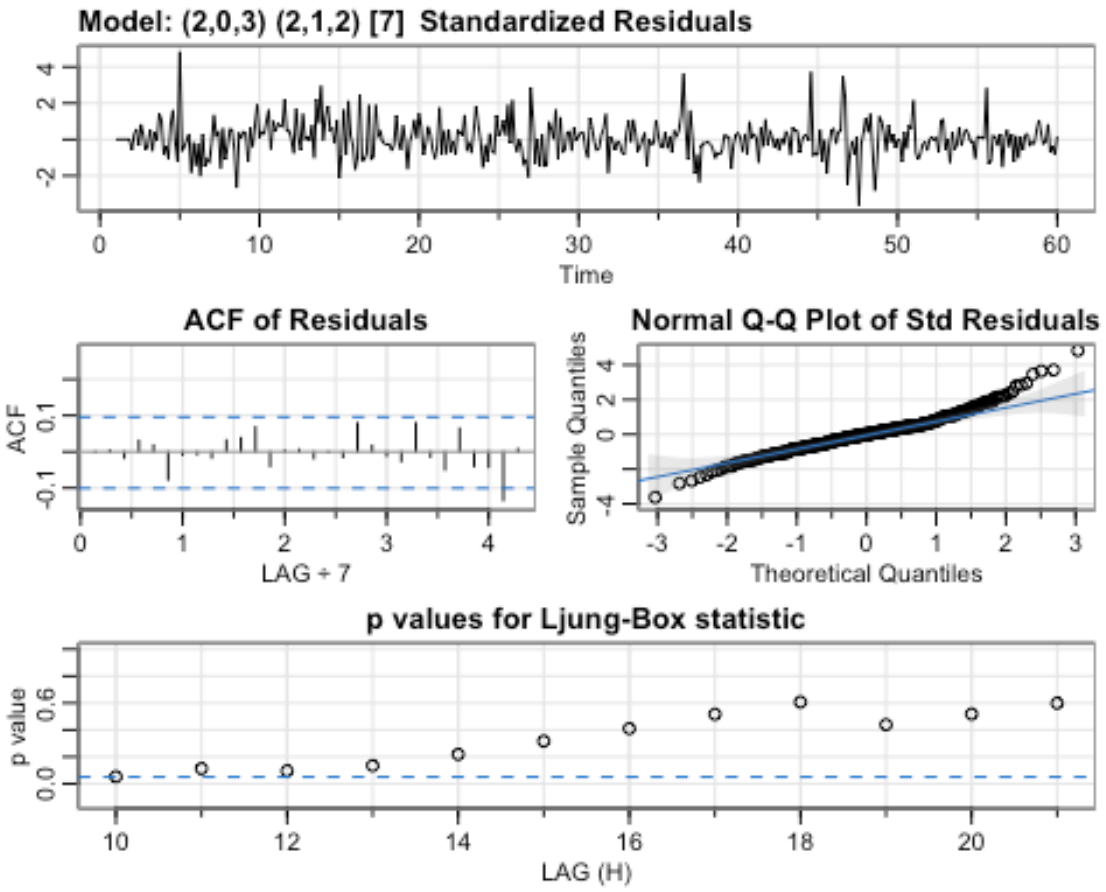
the points going under the line for big positive ones. And for big negative values, it still follows the straight line.

```
acf2(diff(Sal_m_ts, lag=7))
```

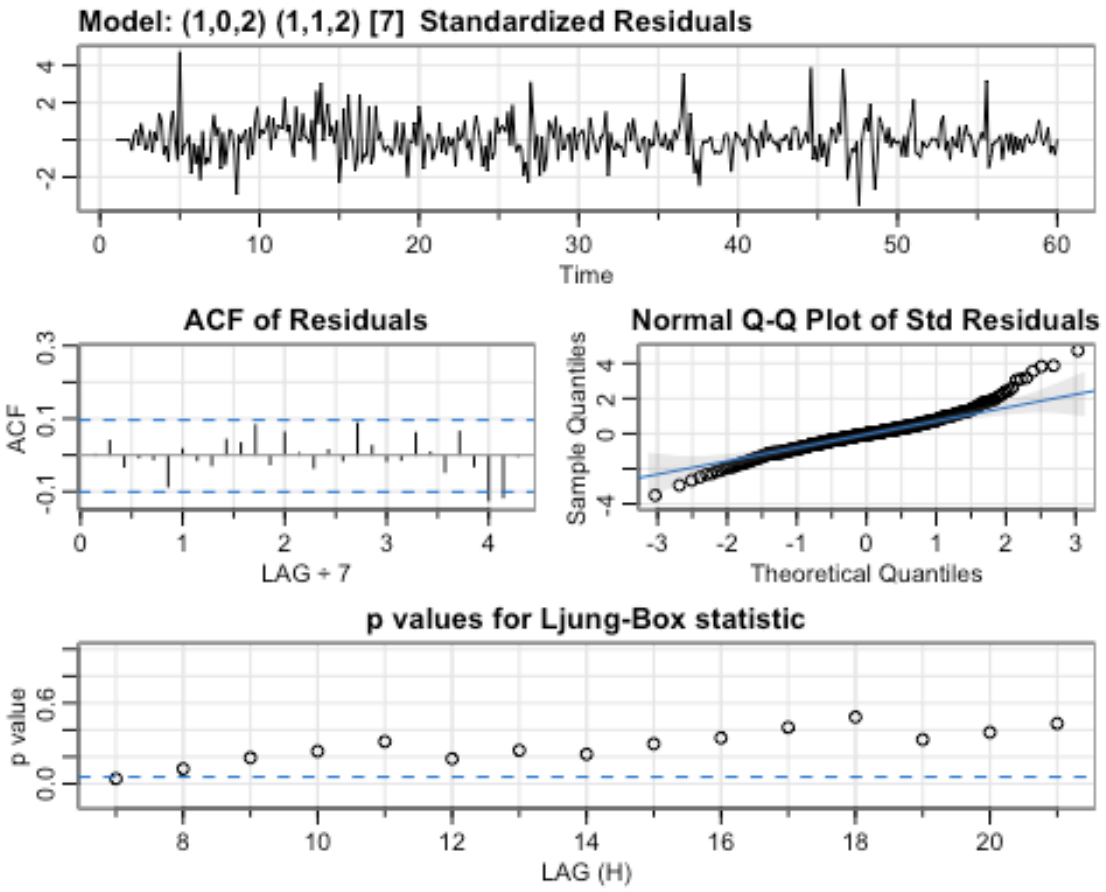


For seasonally differenced time seires, according to the ACF and PACF graphs, the model sould be close to (2,0,3)(2,1,2)[7]. I will plot multiple models as to compare the AICc and the Ljung-Box statistic to see which model outperforms the rest.

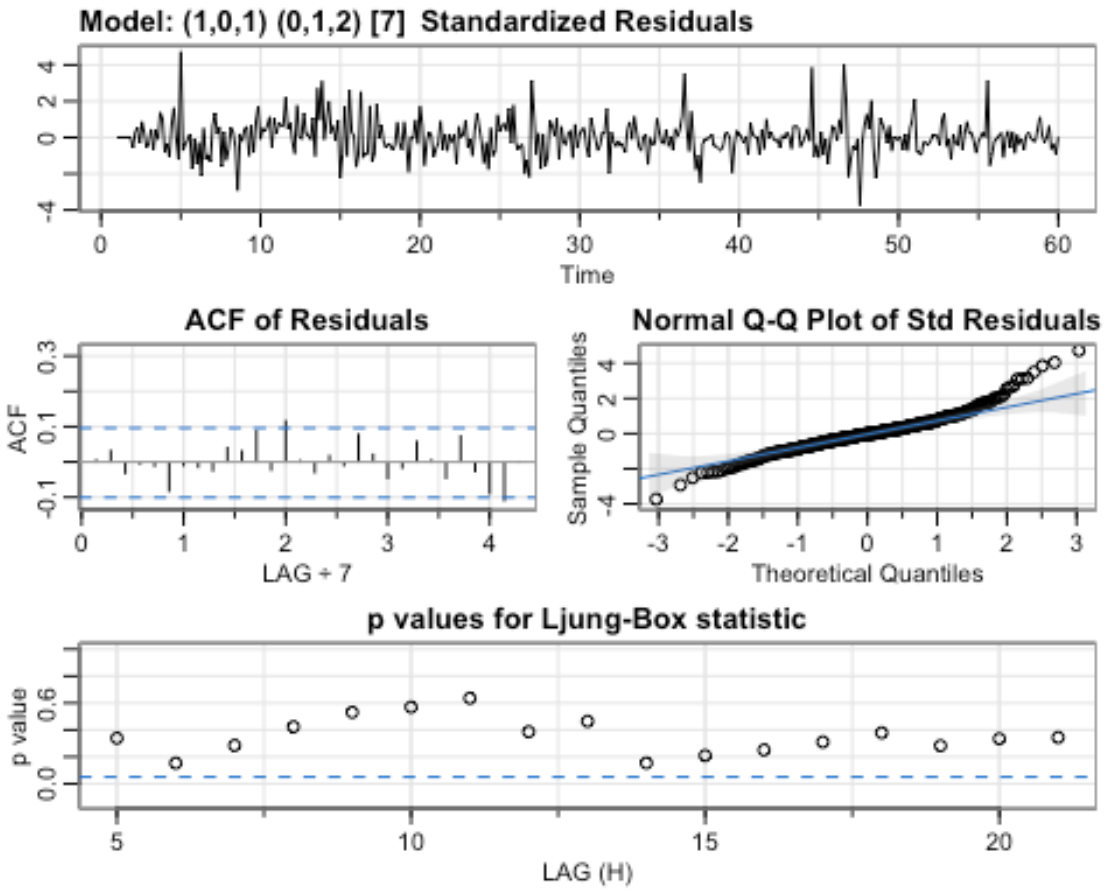
```
Sal_m_m1 = sarima(Sal_m_train, S=7,
                  p=2, d=0, q=3,
                  P=2, D=1, Q=2)
```



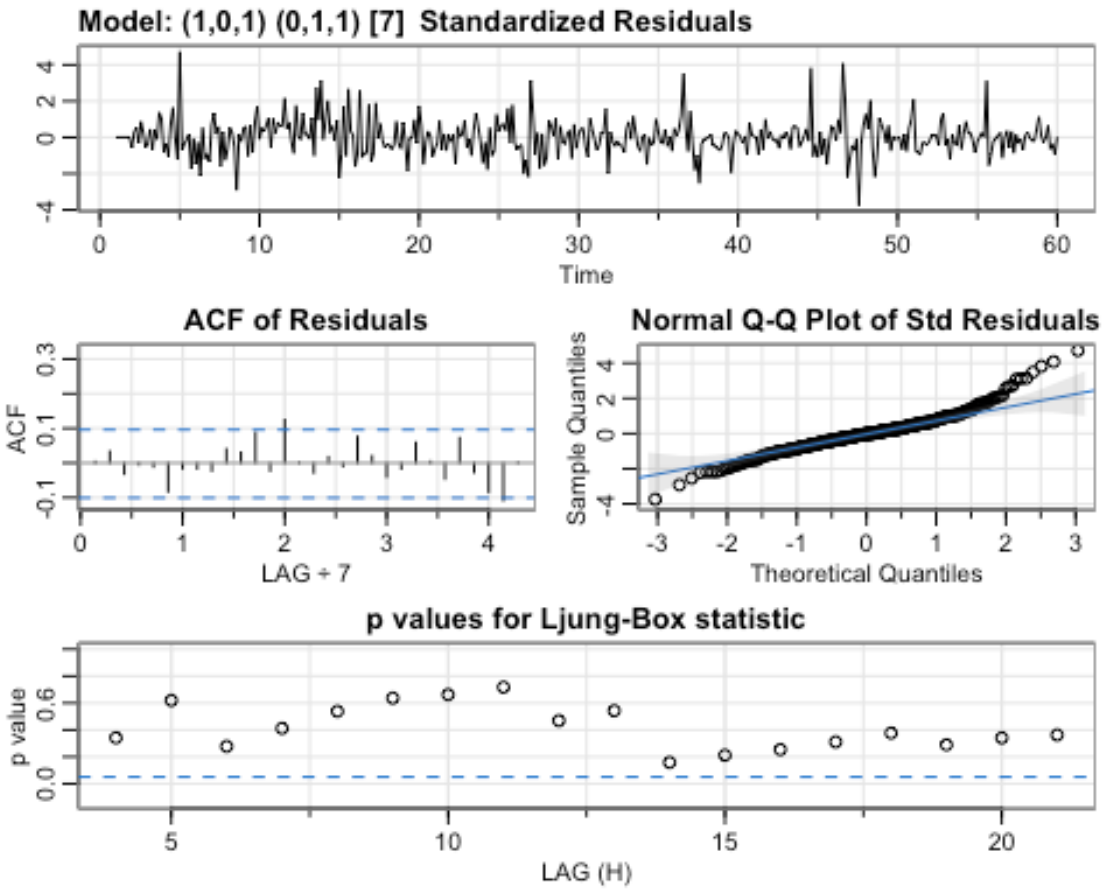

```
Sal_m_m2 = sarima(Sal_m_train, S=7,
                  p=1, d=0, q=2,
                  P=1, D=1, Q=2)
```




```
Sal_m_m3 = sarima(Sal_m_train, S=7,
                  p=1, d=0, q=1,
                  P=0, D=1, Q=2)
```




```
Sal_m_m4 = sarima(Sal_m_train, S=7,
                  p=1, d=0, q=1,
                  P=0, D=1, Q=1)
```




```
Sal_m_m1$AICc

## [1] 22.68504

Sal_m_m2$AICc

## [1] 22.71225

Sal_m_m3$AICc

## [1] 22.70906

Sal_m_m4$AICc

## [1] 22.70421
```

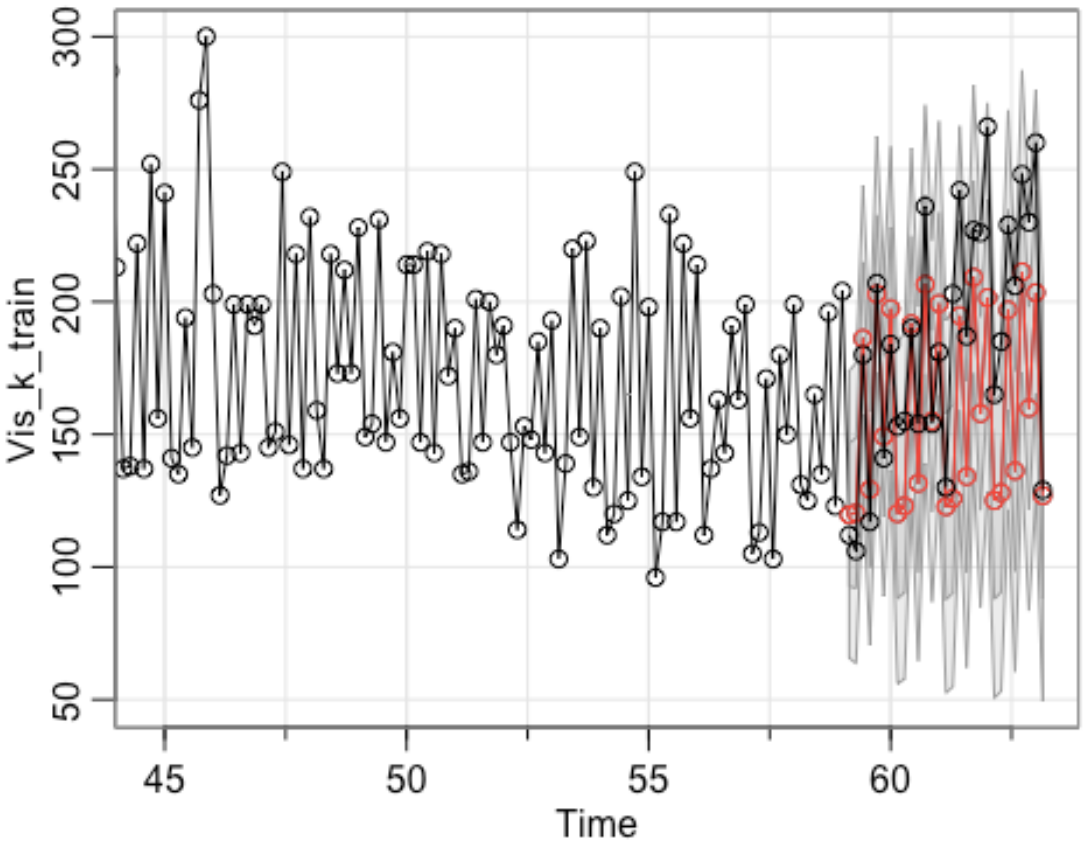
The model 4 (1,0,1)(0,1,1)[7] performs good enough. It has higher AICc when compared to the model that resulted from reading the AFC adn PACF graph, however, the Ljung_Box statistic suggests that model 4 is a better model.

Furthermore, for the Normal Q-Q Plot of Std Residuals, my interpretation would be that the middle values of the sample are close to what you'd expect from normally distributed data, as it follows the straight line from the diagram closely. However, it seems the underlying

data distribution presents extreme values more often than a normal one, that's why we see the points going under the line for big negative values and over it for big positive ones.

Additional Analysis:

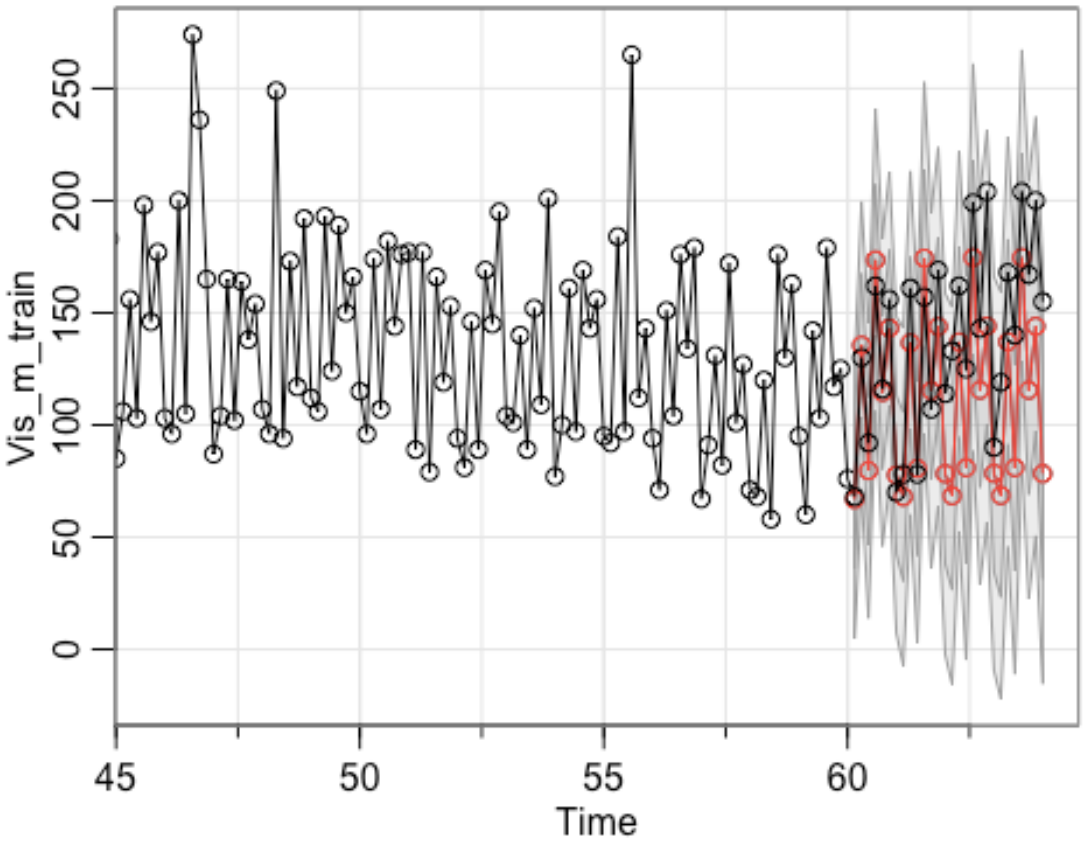
```
length(Vis_k_test)

## [1] 29

#forecasting for visitors of Kalol based on the model that we identified in t
he previous section
Vis_k_for = sarima.for(Vis_k_train, n.ahead = 29, S=7, p=1, d=0, q=2,  P=1, D
=1, Q=1 )
lines(Vis_k_test, type='o')
```



We see that the forecasts are very close to the observed values at least for the initial few observations.

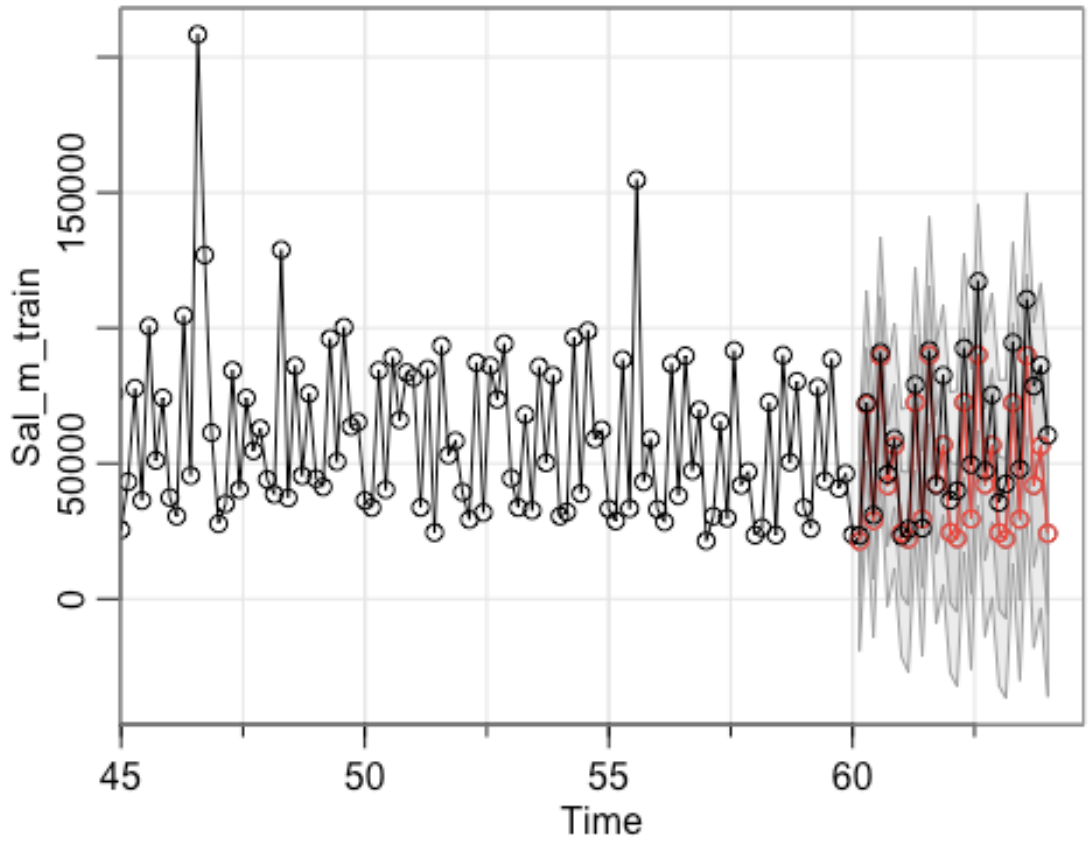
```
length(Vis_m_test)

## [1] 28

#forecasting for visitors of Mehsana based on the model that we identified in
the previous section
Vis_m_for = sarima.for(Vis_m_train, n.ahead = 28, S=7, p=1, d=0, q=1,  P=0, D
```

```
=1, Q=1 )
lines(Vis_m_test, type='o')
```



We see that the forecasts are very close to the observed values at least for the initial few observations.

```
#forecasting for sales of Mehsana based on the model that we identified in th
e previous section
Sal_m_for = sarima.for(Sal_m_train, n.ahead = 28, S=7, p=1, d=0, q=1,  P=0, D
=1, Q=1 )
lines(Sal_m_test, type='o')
```



We see that the forecasts are very close to the observed values at least for the initial few observations.

```
accuracy(Vis_k_for$pred, Vis_k_test)

##                ME     RMSE     MAE      MPE     MAPE     ACF1 Theil's U
## Test set 25.40522 39.34201 31.1425 11.62529 15.70408 0.504819 0.6398808

accuracy(Vis_m_for$pred, Vis_m_test)

##                ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 24.30965 35.27167 28.10173 16.16783 19.35185 0.4869416  0.527526

accuracy(Sal_m_for$pred, Sal_m_test)

##                ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 12918.06 17396.12 13256.58 20.24439 21.55571 0.6094488   0.30221
```

Summary and Implications:

- The restaurant visitors and sales was successfully analyzed and forecasted.
- Each time-series was individually decomposed, differenced to achieve stationarity, ACF and PACF graph was observed and multiple ARIMA models were fit based on the ACF and PACF of the differenced series, the most apppropriate ARIMA model was identified using metrics like AICc and observing the residual plots.
- This selected appropriate ARIMA model was used to forecast for a set of test observations. And the accuracy of these forecasts were checked by comparing them to the actual observed values. The accuracy was not great for all these forecasts.
- One of the major observations was that the forecasts and the observed values were close to each other for the initial few test-observations and moved further apart for the later test-observations.
- There is a further scope of imporvement on these predictions and models by applying other Non-ARIMA models or maybe some Machine Learning models.
- I also tried out applying Facebook's Prophet to model these time-series as the application of holidays would play a major role in Visitors and Sales of the restaurant but it wasn't a great hit either. It performed worse than ARIMA models that we selected here. I believe using ML models would be better. I will check that out in the future.
- And finally, the results from this project (predicted number of visitors and predicted total sales for 3 consecutive days) were shared with the stakeholders (the results were quite impressive considering how simple models were used) and these results were used in business decision-making.
- One of the most important lessons that I learnt practically while working on this project was that simple model with fewer model parameters is always easier to interpret compared to an overly complex models. And simple models even though not as accurate as complex models are still far more interpretable.
- This is an interesting project and I will continue to find that perfect sweet spot of accurate and interpretable model for this particular dataset.

References:

[1.] https://saas.berkeley.edu/rp/a-shallow-dive-into-time-series-analysis-of-local-restaurant-data-using-r

[2.] Boomija, G., Anandaraj, A., Nandhini, S., & Lavanya, S. (2018). Restaurant visitor time series forecasting using autoregressive integrated moving average. Journal of Computational and Theoretical Nanoscience, 15(5), 1590-1593.

[3.] Liu, L. M., Bhattacharyya, S., Sclove, S. L., Chen, R., & Lattyak, W. J. (2001). Data mining on time series: an illustration using fast-food restaurant franchise data. Computational Statistics & Data Analysis, 37(4), 455-476.

[4.] https://medium.com/@lhagenau/a-brief-outline-of-time-series-analysis-forecasting-with-excel-using-restaurant-sales-data-518e5de1a12e