WEB TRAFFIC ANALYSIS

DATA 598: PROJECT REPORT

Anuhya B S

INTRODUCTION

CONTEXT AND BACKGROUND:

Analysis and forecasting web traffic has many applications in various areas. It is a proactive approach to provide secure, reliable and qualitative web communication. Web traffic is most generally defined as the amount of data sent and received by visitors to a website, which is representative of the total number of people visiting the site as well. In recent years, emphasis on how to predict traffic of web pages has increased significantly. Predicting web traffic can help web site owners in many ways including: 1. determining an effective strategy for load balancing of web pages residing in the cloud 2. forecasting future trends based on historical data 3. understanding the user behavior.

For this project, web traffic from Wikipedia has been used. Wikipedia is a popular multilingual free content online encyclopedia written and maintained by a community of volunteers through a model of open collaboration. It grants open access to all traffic data and provides lots of additional information in a context network besides single keywords. Wikipedia is often used for deep topical reading. Thus, it is a great platform to forecast the trends of Wiki pages based on historical data.

GOALS:

1. Grouping the data based on the language of the page and seeing if there exist any interesting patterns in web traffic based on language patterns. (ex: English, French, Chinese)
2. Forecasting future traffic for each language of the web pages as a group.

I am interested in this project as it helps me understand the underlying principles of time series forecasting by applying them on a real-world web traffic model. I believe that by understanding this I can also use such models in various other applications such as vehicle traffic forecasting, network packet forecasting etc.

DATA DESCRIPTION

The data set consists of approximately 145k time series. Each of these time series represent a number of daily views of different Wikipedia articles, starting from July, 1st, 2015 up until December 31st, 2016. The data set has 804 columns – except the first column, each column represents a date and the daily traffic for that particular Wikipedia page. The first column contains the name of the page, the language of the page, type of access and agent.

EXPLORATORY ANALYSIS

Loading Libraries and Data

```
library(astsa)
library(forecast)library(tseries)
library(stringi)

wtd <- read.csv('train_2.csv',check.names = FALSE)
dim(wtd)

## [1] 145063    804
```

The dimensions of the data set are 145063 rows and 804 columns.

Handling missing values

```
na_counts <- colSums(is.na(wtd))
head(na_counts)

##       Page 2015-07-01 2015-07-02 2015-07-03 2015-07-04 2015-07-05
##          0      20740      20816      20544      20654      20659
```

The data set has several missing values. I believe there are two main reasons for the missing values - first is because the Wikipedia pages were not created for the topics and second because there is actual missing data. For now, I have substituted the NA values with 0 for both the cases.

```
wtd[is.na(wtd)] <- 0
```

Grouping the data by languages

Since the data is humongous, it makes sense to group the data by languages and see if there is an influence of language on the pages. The getLang function is designed to extract the language of each page from the 'Page' column in the data set.

```
getLang <- function(page){
  res <- stri_extract(str = page, regex = '[a-z][a-z].wikipedia.org')
  if(!is.na(res))
    return(substr(res,0,2))
  return('na')
}
```

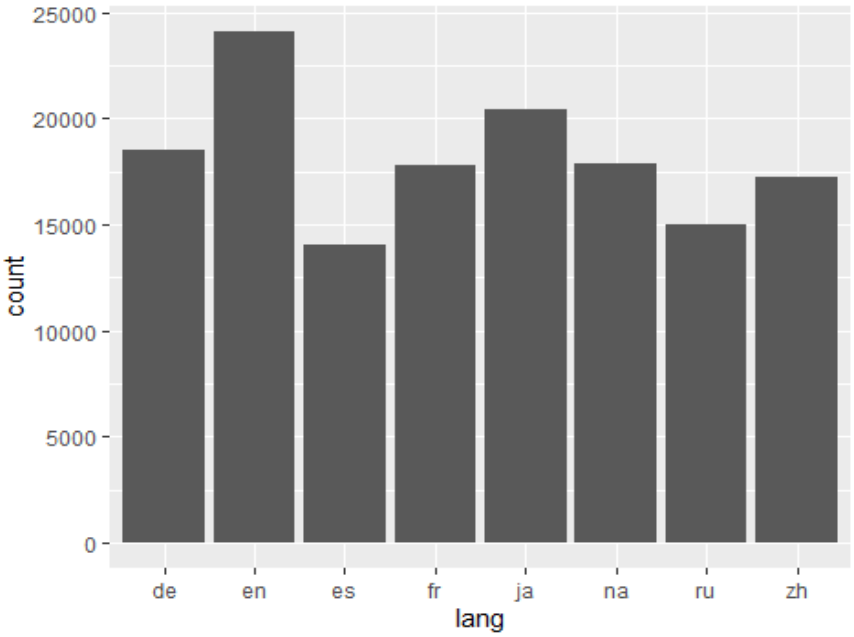
There are 7 distinct languages in the data set. The two letter words correspond to the following languages:

- de - German
- en - English
- es - Spanish
- fr - French
- ja - Japanese
- ru - Russian

- zh - Chinese

Next I have written a function : grpByLang that groups the data set based on the language of the page and stores the data into separate lists. To group the pages by language, I have taken the average of all the views for all pages of each language. Each language list is then transposed so that the dates act as rows and number of visits become the column. Finally, it is converted into a time series object with a frequency of 7 as it is a daily data set. The plot shows the counts of each of the languages in the data set.

```
library(ggplot2)
wtd$lang <- sapply(wtd$Page,FUN = getLang)
table(wtd$lang)

##
##    de    en    es    fr    ja    na    ru    zh
## 18547 24108 14069 17802 20431 17855 15022 17229

ggplot(data=wtd, aes(x=lang)) +
  geom_bar()
```




```
langCodes <- unique(wtd$lang)
wtd_lang <- data.frame()

grpByLang <- function(l, wtd_ln){
  temp <- subset(wtd_ln, lang == l)
  temp <- subset(temp, select = -c(lang))
  wtd_ln_sums <-  colSums(temp[,-1]) / nrow(temp)
  wtd_ln_sums$lang <- l
  return(wtd_ln_sums)
}
res <- list()
```

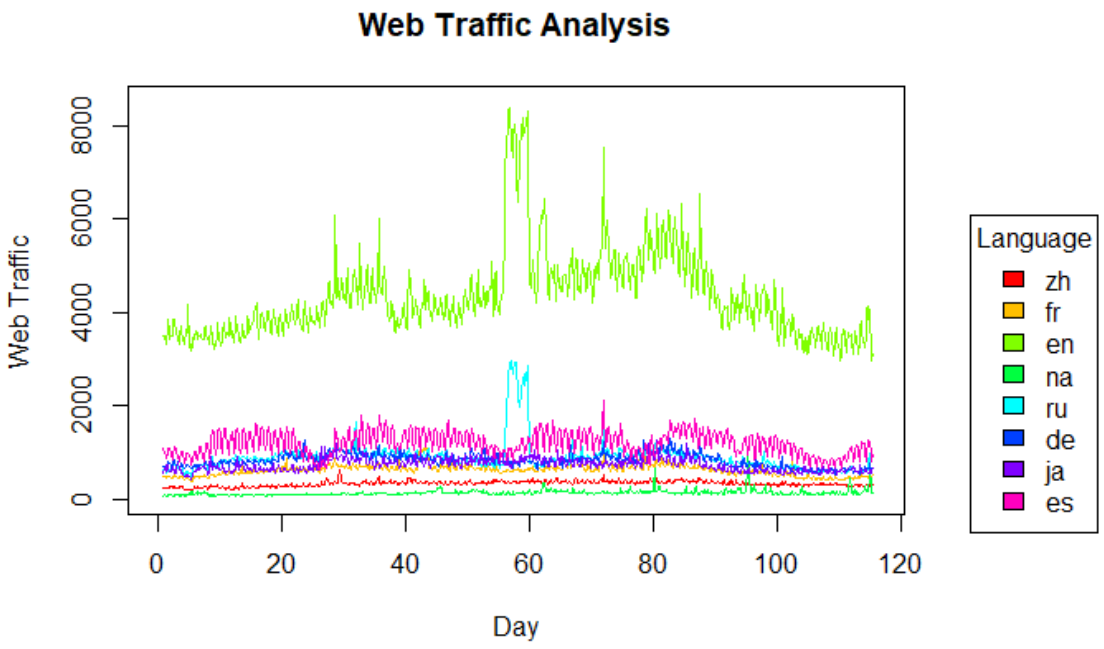
```r
for (i in 1:length(langCodes)){
  res[[i]] <- grpByLang(langCodes[i], wtd)
}

library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

wtd_zh <- as.data.frame(res[[1]], check.names = FALSE)
wtd_zh <- as.data.frame(t(wtd_zh[,-804]), check.names = FALSE)
wtd_zh$date <- as.Date(rownames(wtd_zh))
wtd_zh_ts <- ts(wtd_zh$V1, frequency = 7)

wtd_fr <- as.data.frame(res[[2]], check.names = FALSE)
wtd_fr <- as.data.frame(t(wtd_fr[,-804]))
wtd_fr$date <- as.Date(rownames(wtd_fr))
wtd_fr_ts <- ts(wtd_fr$V1, frequency = 7)

wtd_en <- as.data.frame(res[[3]], check.names = FALSE)
wtd_en <- as.data.frame(t(wtd_en[,-804]))
wtd_en$date <- as.Date(rownames(wtd_en))
wtd_en_ts <- ts(wtd_en$V1, frequency = 7)

wtd_na <- as.data.frame(res[[4]], check.names = FALSE)
wtd_na <- as.data.frame(t(wtd_na[,-804]))
wtd_na$date <- as.Date(rownames(wtd_na))
wtd_na_ts <- ts(wtd_na$V1, frequency = 7)

wtd_ru <- as.data.frame(res[[5]], check.names = FALSE)
wtd_ru <- as.data.frame(t(wtd_ru[,-804]))
wtd_ru$date <- as.Date(rownames(wtd_ru))
wtd_ru_ts <- ts(wtd_ru$V1, frequency = 7)

wtd_de <- as.data.frame(res[[6]], check.names = FALSE)
wtd_de <- as.data.frame(t(wtd_de[,-804]))
wtd_de$date <- as.Date(rownames(wtd_de))
wtd_de_ts <- ts(wtd_de$V1, frequency = 7)

wtd_ja <- as.data.frame(res[[7]], check.names = FALSE)
wtd_ja <- as.data.frame(t(wtd_ja[,-804]))
wtd_ja$date <- as.Date(rownames(wtd_ja))
wtd_ja_ts <- ts(wtd_ja$V1, frequency = 7)

wtd_es <- as.data.frame(res[[8]], check.names = FALSE)
wtd_es <- as.data.frame(t(wtd_es[,-804]))
```

```
wtd_es$date <- as.Date(rownames(wtd_es))
wtd_es_ts <- ts(wtd_es$V1, frequency = 7)
```

Plotting the series

I have plotted the web traffic of each language in a different colors. This helps us understand the language that in general have the highest number of visitors as well as identify any patterns in the data which may common across languages.

```
par(mar=c(5, 4, 4, 8), xpd=TRUE)
plot(0,0,xlim = c(0,116), ylim = c(0,8500), type = "n", main = "Web Traffic A
nalysis", xlab= "Day", ylab = "Web Traffic")
cl <- rainbow(8)
lines(wtd_zh_ts, col = cl[1], type = 'l')
lines(wtd_fr_ts,col = cl[2], type = 'l')
lines(wtd_en_ts,col = cl[3], type = 'l')
lines(wtd_na_ts,col = cl[4], type = 'l')
lines(wtd_ru_ts,col = cl[5], type = 'l')
lines(wtd_de_ts,col = cl[6], type = 'l')
lines(wtd_ja_ts,col = cl[7], type = 'l')
lines(wtd_es_ts,col = cl[8], type = 'l')
legend("topright",inset=c(-0.25, 0.3), legend = langCodes,fill = cl, title="L
anguage")
```



We can see form the plot, that the English Wikipedia pages have the most traffic. There is also a significant spike in traffic around the middle of the data set for both the English and the Russian pages which distinctly stands out in the plot.

Analyzing, Forecasting and Modeling each language time series

For each language, I have taken the following steps:

1. Splitting the language data into training and test set

2. Plotting the training data and eyeballing to see if the time series looks stationary

3. Performing the KPSS test to check for stationarity

4. Apply STL decomposition to the time series to understand the trend component, seasonal component and the remainder component.

5. All the language time series have some amount of seasonality so I have applied Spectral Analysis to discover any underlying peaks/ periodicities that are immediately visible from the ACF Plots.

6. Plotted the Autocorrelation plots

7. Applied seasonal/ non-seasonal differencing based on the time series data.

8. Identified and fit potential ARIMA models for the time series data and evaluated the residual plots for each model.

9. Forecasting the time series using the most appropriate model identified in Step 8.

10. Evaluating the accuracy of the forecast.

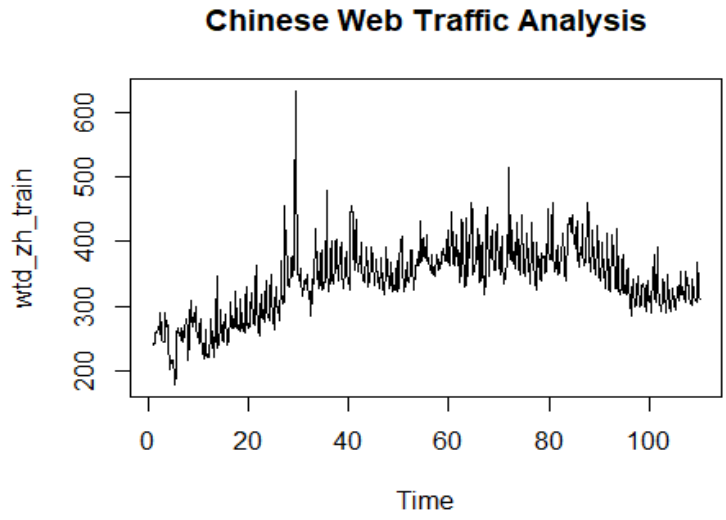
I have briefly described the results of each step and my decision process behind selecting a particular model.

Please note that I have only considered the seven languages (de, en, es, fr, ja, ru, zh) for this project and not the 'na' time series as it is not language related and mainly deals with media links.

Chinese Web Traffic

Splitting the data set into train and test sets:

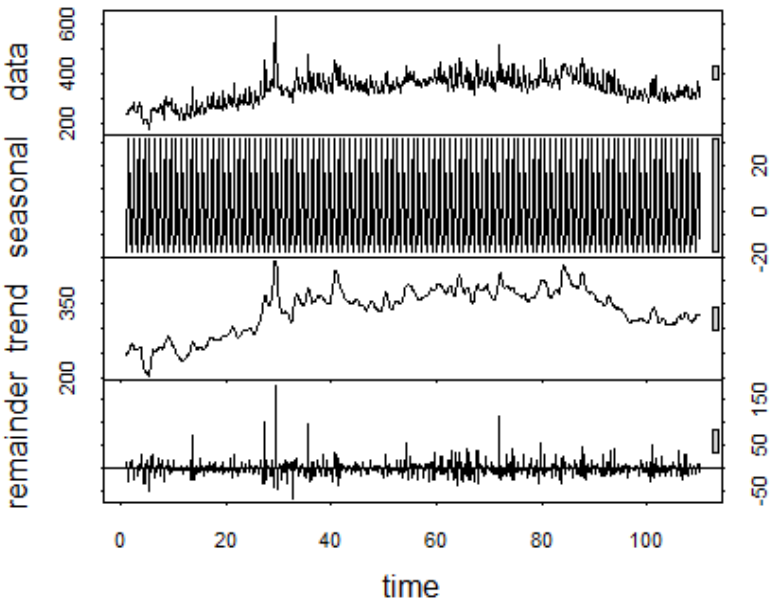
```
wtd_zh_train <- window(wtd_zh_ts, end = c(110,1))
wtd_zh_test <- window(wtd_zh_ts, start = c(110,2))
plot(wtd_zh_train, main = "Chinese Web Traffic Analysis")
```



There is a noticeable upward trend in the first few months, followed by a large spike in the traffic. There also appears to be a seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_zh_stl <- stl(wtd_zh_train, s.window = "periodic")
plot(wtd_zh_stl)
```



Performing the KPSS test to verify the stationarity:

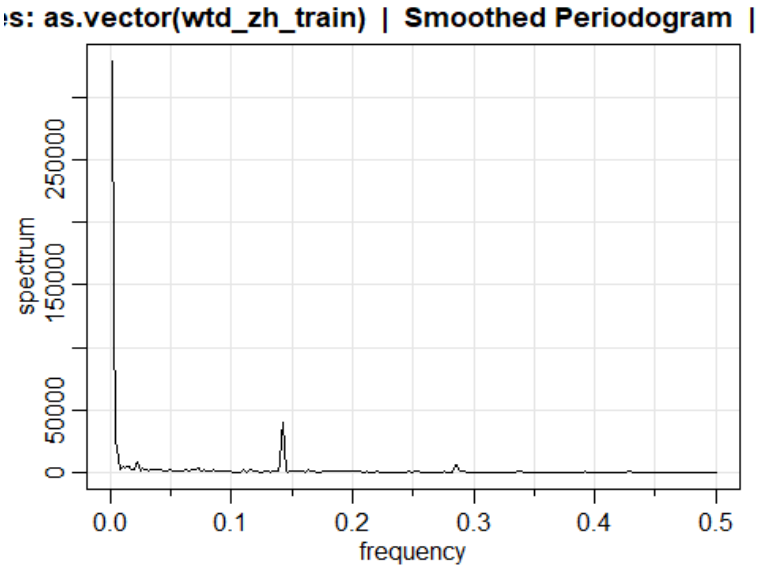
```
kpss.test(wtd_zh_train)
```

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

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_zh_train
## KPSS Level = 4.5459, Truncation lag parameter = 6, p-value = 0.01
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_zh.spec <- mvspec(as.vector(wtd_zh_train),detrend = TRUE, spans = 3)
```

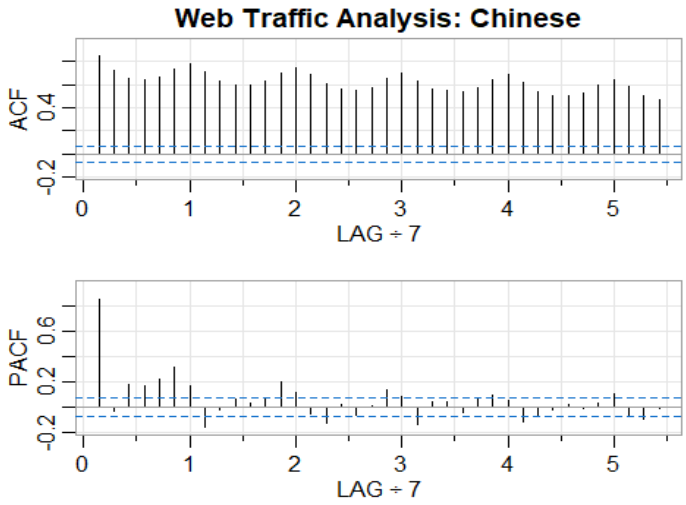



```
head(wtd_zh.spec$details)

##      frequency period   spectrum
## [1,]    0.0013  768.0 329609.820
## [2,]    0.0026  384.0 137328.764
## [3,]    0.0039  256.0  26474.243
## [4,]    0.0052  192.0  18741.488
## [5,]    0.0065  153.6  10304.866
## [6,]    0.0078  128.0   2044.725
```

The plot shows one major peak 1/0.14 which is approx. 7 days. This is indicative of a weekly seasonality. There is also a small peak around 1/0.28 which is approx. 3 days.

Plotting the Autocorrelation plot:

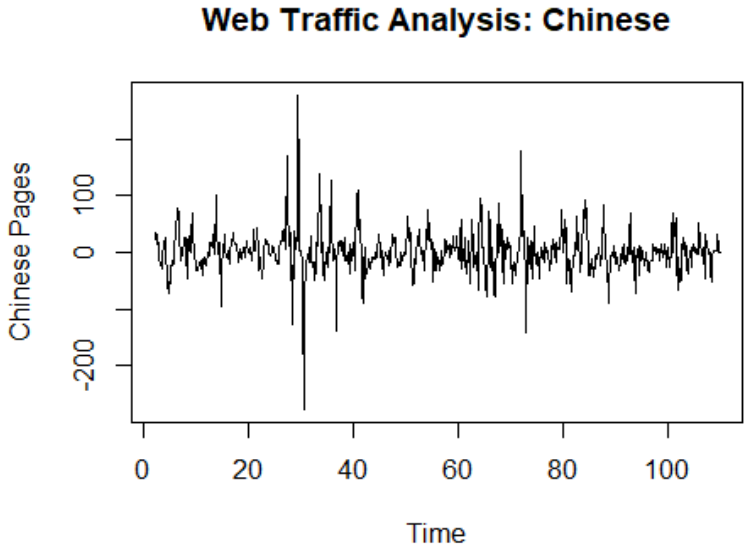
```
acf2(wtd_zh_train, main = "Web Traffic Analysis: Chinese")
```



The autocorrelations show a high lag every 7 days which is an indication of a weekly seasonality.

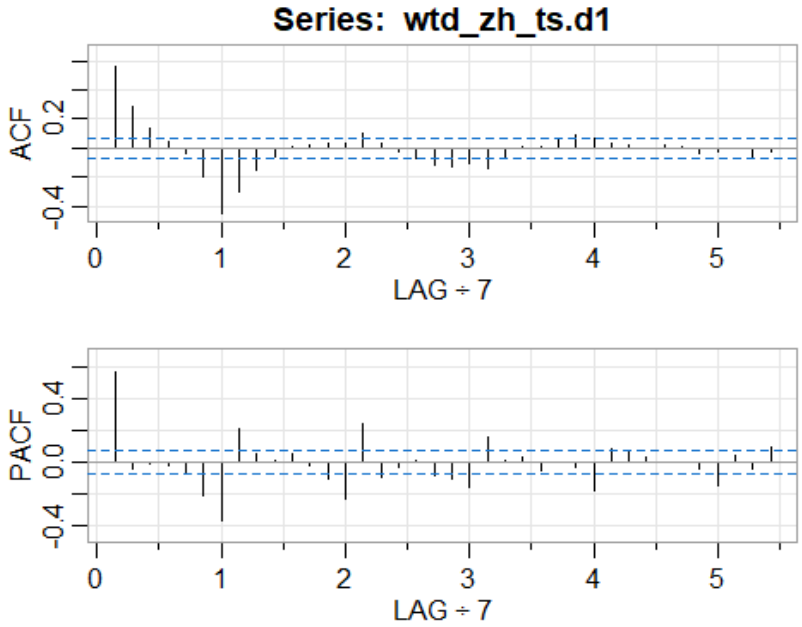
Performing Seasonal Differencing:

```
wtd_zh_ts.d1 <- diff(wtd_zh_train, lag = 7)
plot(wtd_zh_ts.d1,
     main = "Web Traffic Analysis: Chinese",
     ylab = "Chinese Pages", type = 'l')
```




```
kpss.test(wtd_zh_ts.d1)

## Warning in kpss.test(wtd_zh_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
```

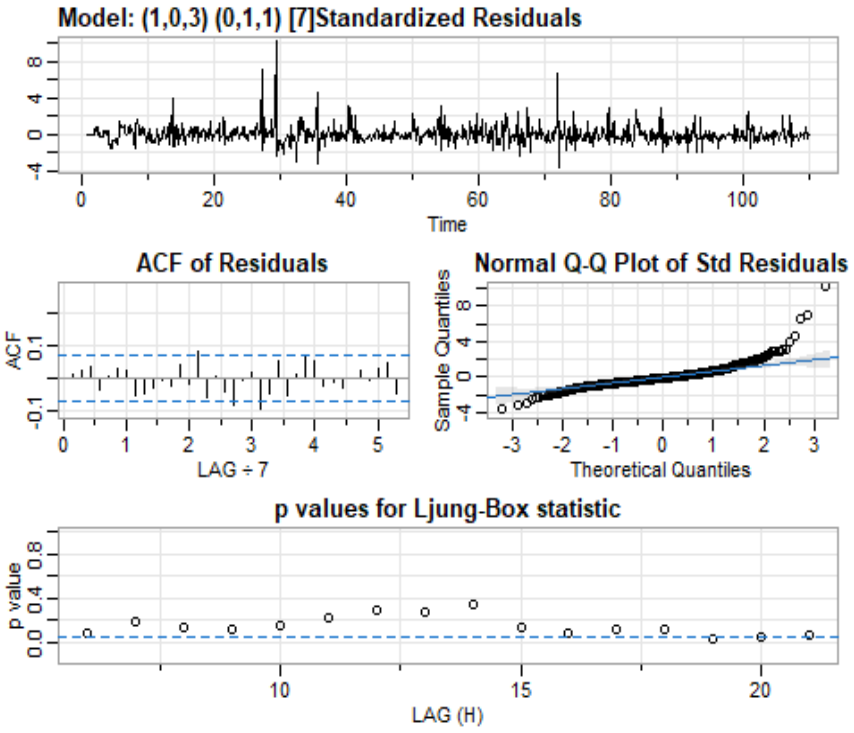
```
## data:  wtd_zh_ts.d1
## KPSS Level = 0.086821, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_zh_ts.d1)
```



From the plot above, intuitively I would pick the following values: Q = 3/1 P = 0 D = 1 q = 3 p = 1/5 d = 0 I would apply ARIMA(1,0,3)(0,1,1)[7], ARIMA(5,0,3)(0,1,1)[7] and run auto ARIMA.

ARIMA Modeling:

```
wtd_zh_sm1 <- sarima(wtd_zh_train, S = 7,
                     p = 1, d = 0, q = 3,
                     P = 0, D = 1, Q = 1)
```

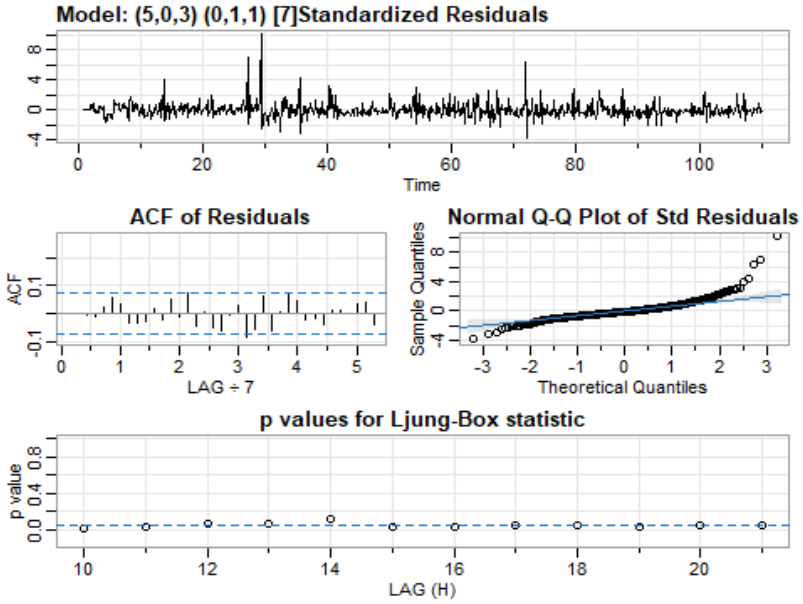



```
wtd_zh_sm1

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1      ma2      ma3     sma1  constant
##       0.9960  -0.3545  -0.2596  -0.1658  -1.0000    0.0996
## s.e.  0.0058   0.0361   0.0399   0.0367   0.0222    0.0979
##
## sigma^2 estimated as 482.7:  log likelihood = -3427.87,  aic = 6869.74
##
## $degrees_of_freedom
## [1] 751
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.9960 0.0058 171.9450  0.0000
## ma1       -0.3545 0.0361  -9.8267  0.0000
## ma2       -0.2596 0.0399  -6.5044  0.0000
## ma3       -0.1658 0.0367  -4.5126  0.0000
## sma1      -1.0000 0.0222 -44.9635  0.0000
```

```
## constant   0.0996 0.0979   1.0173  0.3093
##
## $AIC
## [1] 9.07496
##
## $AICc
## [1] 9.075108
##
## $BIC
## [1] 9.117768
```

```
wtd_zh_sm2 <- sarima(wtd_zh_train, S = 7,
                     p = 5, d = 0, q = 3,
                     P = 0, D = 1, Q = 1)
```

```
## Warning in arima(xdata, order = c(p, d, q), seasonal = list(order = c(P, :
## possible convergence problem: optim gave code = 1
```




```
wtd_zh_sm2
```

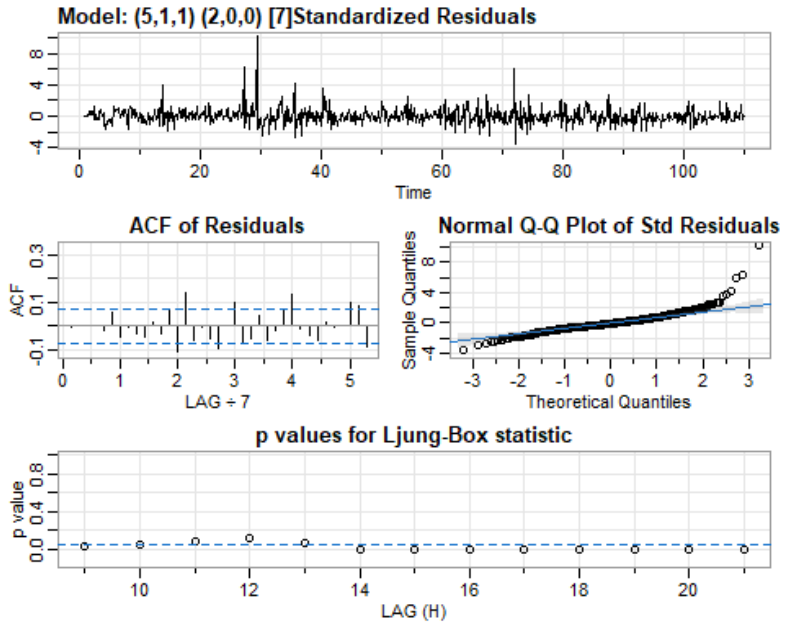
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1    ar2     ar3      ar4     ar5     ma1      ma2      ma3
sma1
```

```
##       0.1029  0.688  0.6808  -0.5351  0.0629  0.5546  -0.3472  -0.9073  -0
.9999
## s.e.  0.0414  0.040  0.0269   0.0405  0.0396  0.0191   0.0338   0.0102   0
.0056
##       constant
##         0.0949
## s.e.    0.1366
##
## sigma^2 estimated as 479.1:  log likelihood = -3425.02,  aic = 6872.04
##
## $degrees_of_freedom
## [1] 747
##
## $ttable
##          Estimate     SE   t.value p.value
## ar1        0.1029 0.0414    2.4873  0.0131
## ar2        0.6880 0.0400   17.2026  0.0000
## ar3        0.6808 0.0269   25.2653  0.0000
## ar4       -0.5351 0.0405  -13.2099  0.0000
## ar5        0.0629 0.0396    1.5873  0.1129
## ma1        0.5546 0.0191   29.0172  0.0000
## ma2       -0.3472 0.0338  -10.2793  0.0000
## ma3       -0.9073 0.0102  -89.2401  0.0000
## sma1      -0.9999 0.0056 -179.0015  0.0000
## constant   0.0949 0.1366    0.6949  0.4874
##
## $AIC
## [1] 9.077986
##
## $AICc
## [1] 9.078376
##
## $BIC
## [1] 9.145256
```

```
auto.arima(wtd_zh_train, seasonal = TRUE)
```

```
## Series: wtd_zh_train
## ARIMA(5,1,1)(2,0,0)[7]
##
## Coefficients:
##          ar1      ar2      ar3     ar4     ar5      ma1   sar1    sar2
##       0.6058  -0.0555  -0.0240  0.0113  0.0501  -0.9691  0.279  0.2164
## s.e.  0.0377   0.0442   0.0425  0.0428  0.0377   0.0097  0.037  0.0379
##
## sigma^2 = 585.5:  log likelihood = -3510.9
## AIC=7039.8   AICc=7040.04   BIC=7081.54
```

```
wtd_zh_sm3 <- sarima(wtd_zh_train, S = 7,
                    p = 5, d = 1, q = 1,
                    P = 2, D = 0, Q = 0)
```




```
wtd_zh_sm3
```

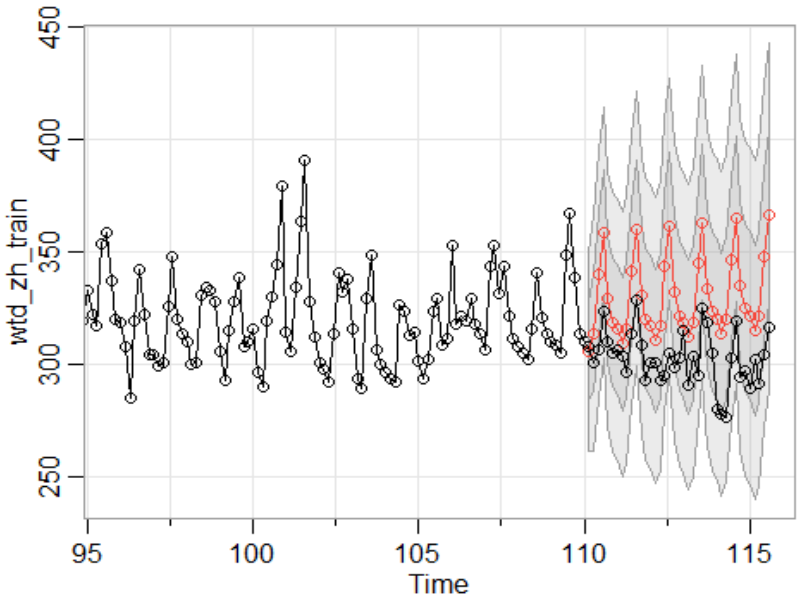
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ar3     ar4     ar5      ma1    sar1    sar2
##       0.6062  -0.0554  -0.0239  0.0115  0.0505  -0.9704  0.2791  0.2163
## s.e.  0.0377   0.0442   0.0425  0.0428  0.0377   0.0096  0.0370  0.0379
##       constant
##         0.1007
## s.e.    0.1280
##
## sigma^2 estimated as 578.9:  log likelihood = -3510.6,  aic = 7041.21
##
## $degrees_of_freedom
## [1] 754
##
## $ttable
##          Estimate     SE   t.value p.value
## ar1        0.6062 0.0377   16.0785  0.0000
```

```
## ar2       -0.0554 0.0442   -1.2527  0.2107
## ar3       -0.0239 0.0425   -0.5620  0.5743
## ar4        0.0115 0.0428    0.2686  0.7883
## ar5        0.0505 0.0377    1.3405  0.1805
## ma1       -0.9704 0.0096 -100.9740  0.0000
## sar1       0.2791 0.0370    7.5488  0.0000
## sar2       0.2163 0.0379    5.7009  0.0000
## constant   0.1007 0.1280    0.7863  0.4320
##
## $AIC
## [1] 9.228318
##
## $AICc
## [1] 9.228631
##
## $BIC
## [1] 9.289095
```

Looking at the above plots, ARIMA(5,0,3)(0,1,1)[7] has the lowest AIC value. However, there is hardly much difference between the AIC value of the other models. I have decided to go ahead with **ARIMA(1,0,3)(0,1,1)[7]** model for forecasting because among all the models it had the best ACF of Residuals and p-values for Ljung-Box statistic and AIC value is also pretty less.

Forecasting:

```
wtd_zh_sm1_for <- sarima.for(wtd_zh_train,n.ahead = 39,S = 7,
                      p = 1, d = 0, q = 3,
                       P = 0, D = 1, Q = 1)
lines(wtd_zh_test, type = 'o')
```



Estimating the accuracy:

```
accuracy(wtd_zh_sm1_for$pred, wtd_zh_test)
```

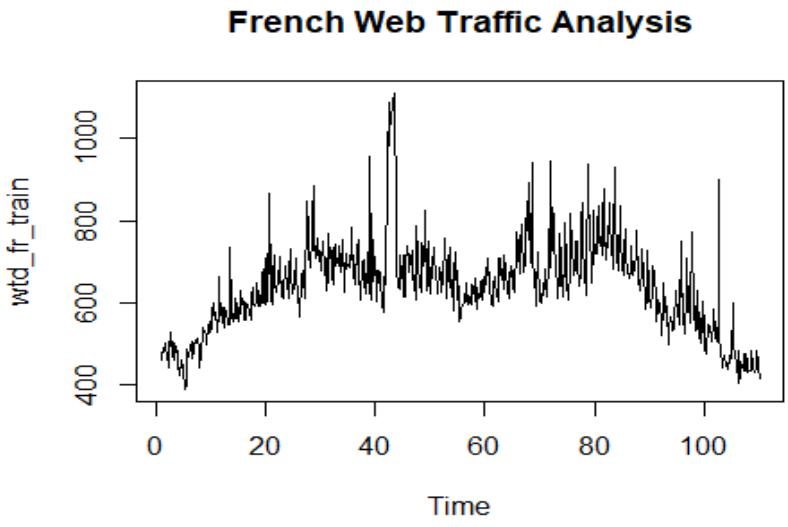
```
##                     ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -27.40973 30.92066 27.5156 -9.093013 9.127398 0.4713231  2.336123
```

The RMSE value is **30.92066**.

French Web Traffic

Splitting the data set into train and test sets:

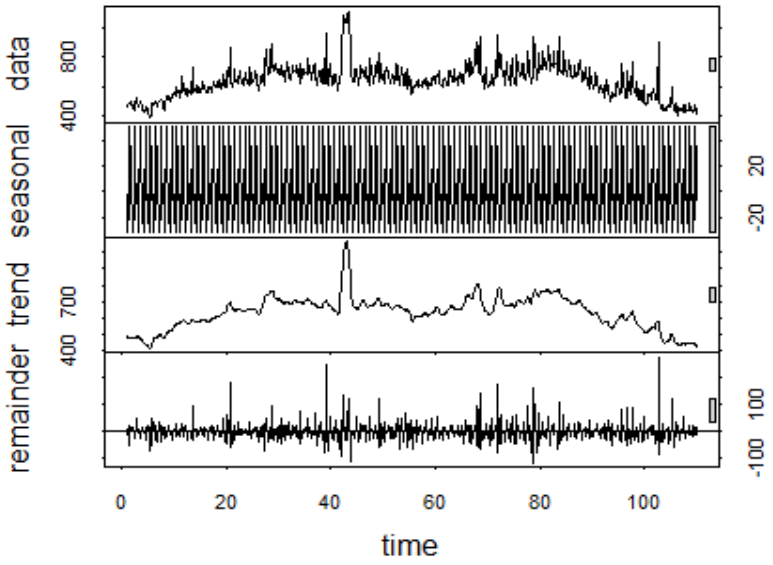
```
wtd_fr_train <- window(wtd_fr_ts, end = c(110,1))
wtd_fr_test <- window(wtd_fr_ts, start = c(110,2))
plot(wtd_fr_train, main = "French Web Traffic Analysis")
```



Similar to the previous time series, there is a noticeable upward trend in the first few months, followed by a large spike in the traffic. There also seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_fr_stl <- stl(wtd_fr_train, s.window = "periodic")
plot(wtd_fr_stl)
```



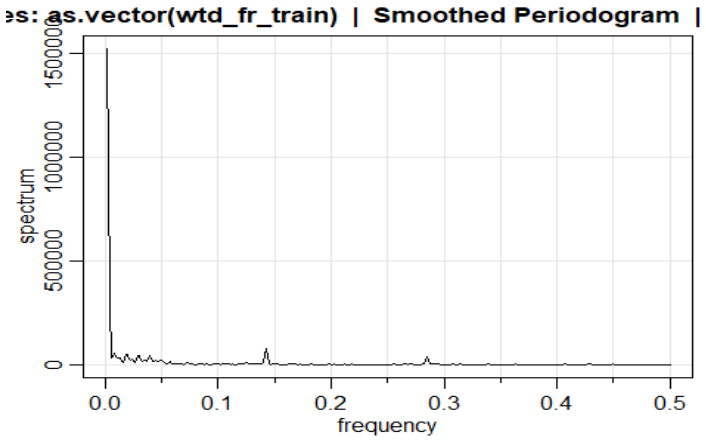
Performing the KPSS Test for stationarity:

```
kpss.test(wtd_fr_train)

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

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_fr_train
## KPSS Level = 1.4725, Truncation lag parameter = 6, p-value = 0.01
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_fr.spec <- mvspec(as.vector(wtd_fr_train),detrend = TRUE, spans = 2)
```

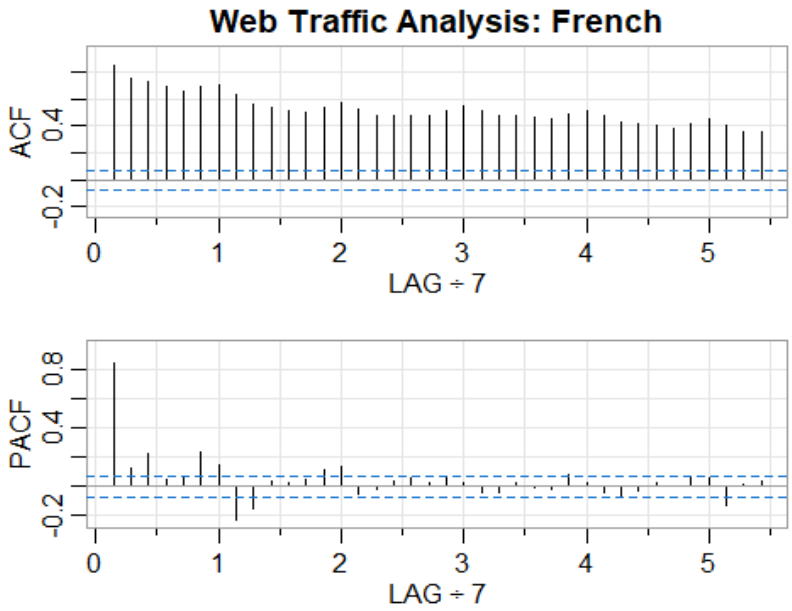


```
head(wtd_fr.spec$details)

##      frequency period   spectrum
## [1,]    0.0013  768.0 1524929.42
## [2,]    0.0026  384.0  951123.81
## [3,]    0.0039  256.0  290629.30
## [4,]    0.0052  192.0   31756.73
## [5,]    0.0065  153.6   39712.76
## [6,]    0.0078  128.0   54385.90
```

The plot shows one major peak 1/0.14 which is approx. 7 days. This is indicative of a weekly seasonality. There is also a small peak around 1/0.28 which is approx. 3 days. There are also some peaks around 1/0.01 (approx. 100 days) to 1/0.04 (approx. 25 days) which is indicative of a quarterly seasonality.

Plotting the Autocorrelation plot:

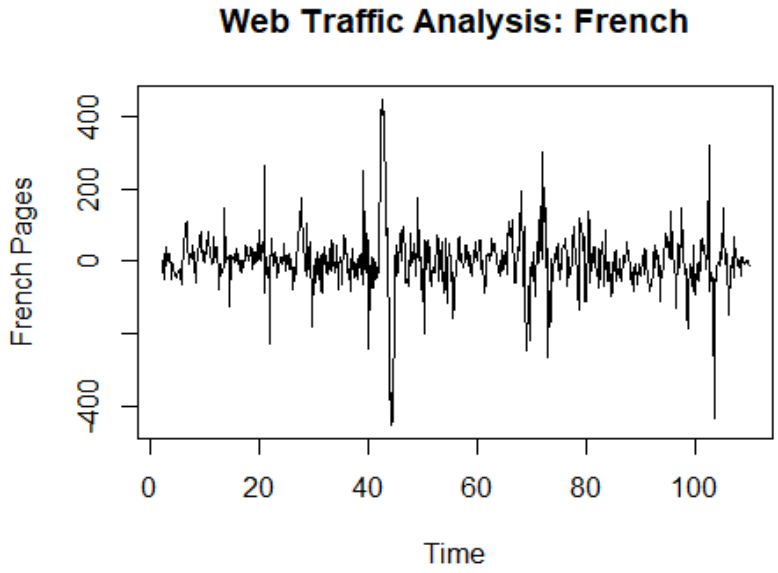
```
acf2(wtd_fr_train, main = "Web Traffic Analysis: French")
```



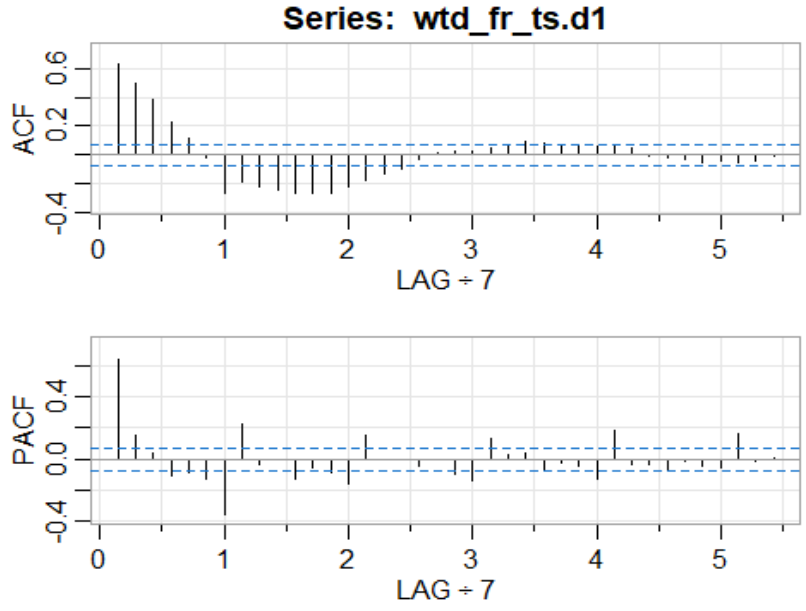
The autocorrelations show a high lag every 7 days which is an indication of a weekly seasonality.

Seasonal Differencing:

```
wtd_fr_ts.d1 <- diff(wtd_fr_train, lag = 7)
plot(wtd_fr_ts.d1,
     main = "Web Traffic Analysis: French",
     ylab = "French Pages", type = 'l')
```



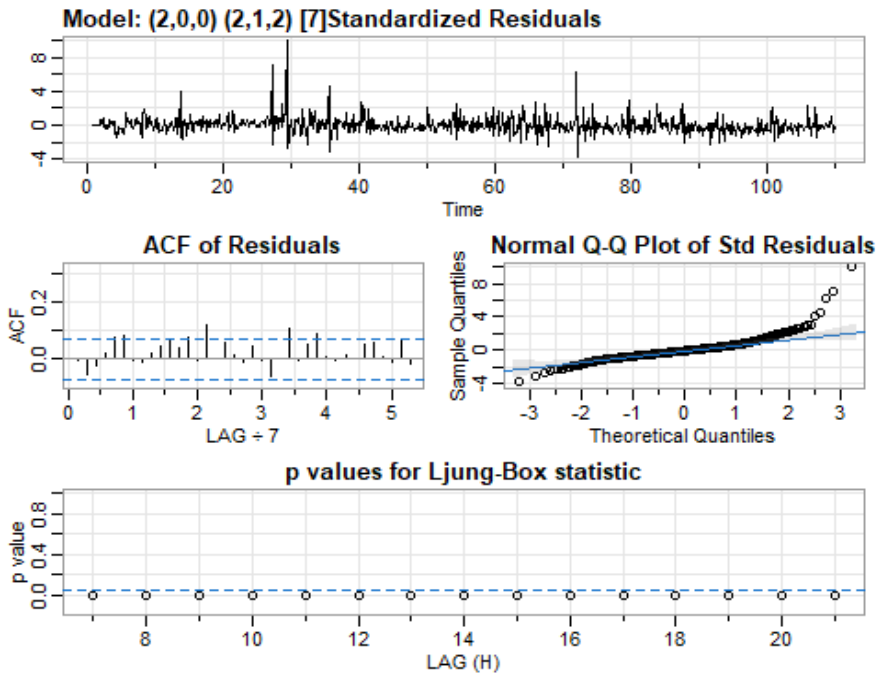

```
kpss.test(wtd_fr_ts.d1)

## Warning in kpss.test(wtd_fr_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_fr_ts.d1
## KPSS Level = 0.11571, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_fr_ts.d1)
```



From the plot above, intuitively I would pick the following values: P = 2 Q = 2 D = 1 d = 0 p = 2 q = 0/4

I would apply ARIMA(2,0,0)(2,1,2)[7] and run auto ARIMA.

```
wtd_fr_sm1 <- sarima(wtd_zh_train, S = 7,
                     p = 2, d = 0, q = 0,
                     P = 2, D = 1, Q = 2)
```



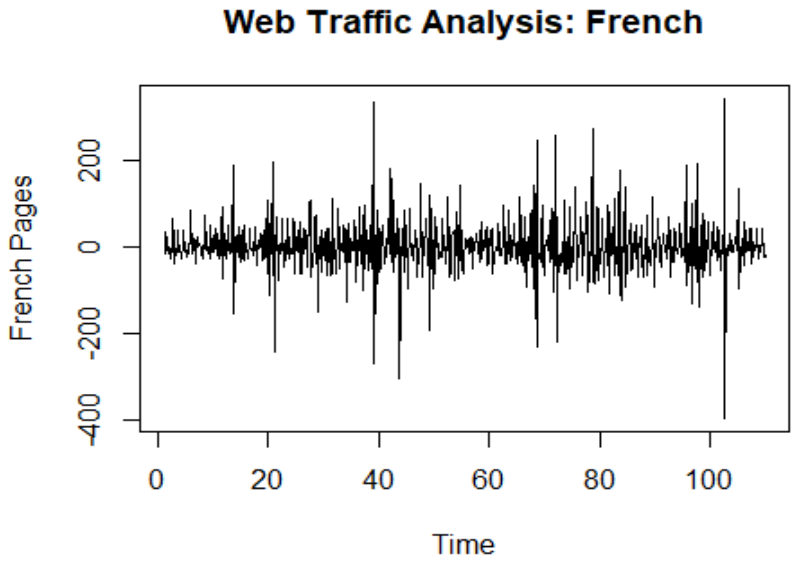

```
wtd_fr_sm1
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2    sar1     sar2     sma1    sma2  constant
##       0.6627  0.0455  0.0583  -0.0721  -0.8931  0.0633     0.102
## s.e.  0.0381  0.0375  0.4915   0.0439   0.4906  0.4166     0.071
##
## sigma^2 estimated as 520.8:  log likelihood = -3446.34,  aic = 6908.69
##
## $degrees_of_freedom
## [1] 750
##
```

```
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.6627 0.0381 17.4144  0.0000
## ar2        0.0455 0.0375  1.2140  0.2251
## sar1       0.0583 0.4915  0.1185  0.9057
## sar2      -0.0721 0.0439 -1.6421  0.1010
## sma1      -0.8931 0.4906 -1.8206  0.0691
## sma2       0.0633 0.4166  0.1520  0.8793
## constant   0.1020 0.0710  1.4364  0.1513
##
## $AIC
## [1] 9.126404
##
## $AICc
## [1] 9.126601
##
## $BIC
## [1] 9.175327
```

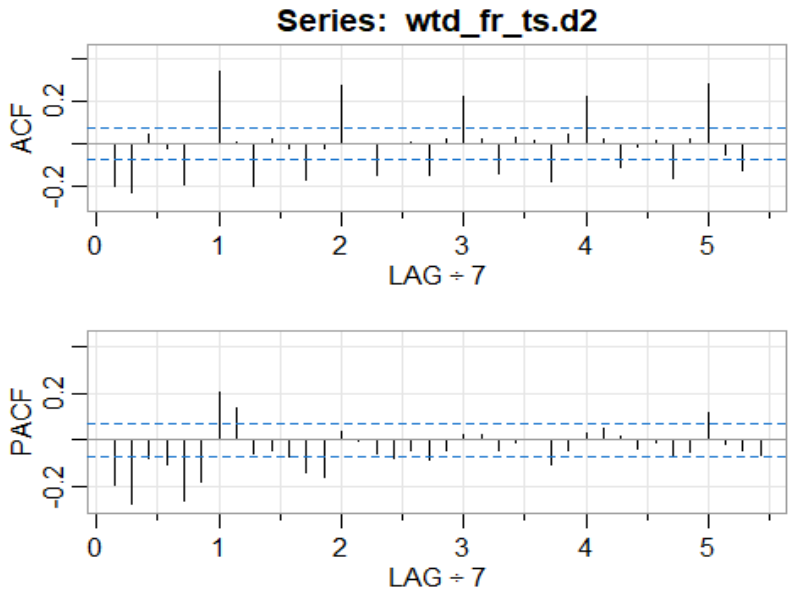
First order differencing on training data as the the time series is very noisy:

```
wtd_fr_ts.d2 <- diff(wtd_fr_train, 1)
plot(wtd_fr_ts.d2,
     main = "Web Traffic Analysis: French",
     ylab = "French Pages", type = 'l')
```




```
kpss.test(wtd_fr_ts.d2)

## Warning in kpss.test(wtd_fr_ts.d2): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
```

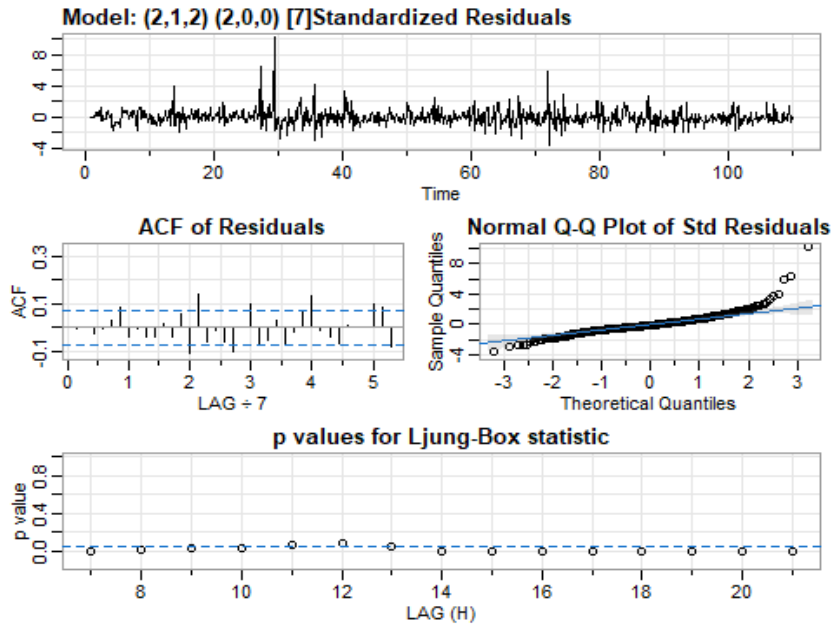
```
##
## data:  wtd_fr_ts.d2
## KPSS Level = 0.06973, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_fr_ts.d2)
```



From the plot above, intuitively I would pick the following values: P = 2 Q = 0 D = 0 d = 1 p = 2 q = 2

I would apply ARIMA(2,0,0)(2,1,2)[7] and run auto ARIMA.

```
wtd_fr_sm2 <- sarima(wtd_zh_train, S = 7,
                     p = 2, d = 1, q = 2,
                     P = 2, D = 0, Q = 0)
```



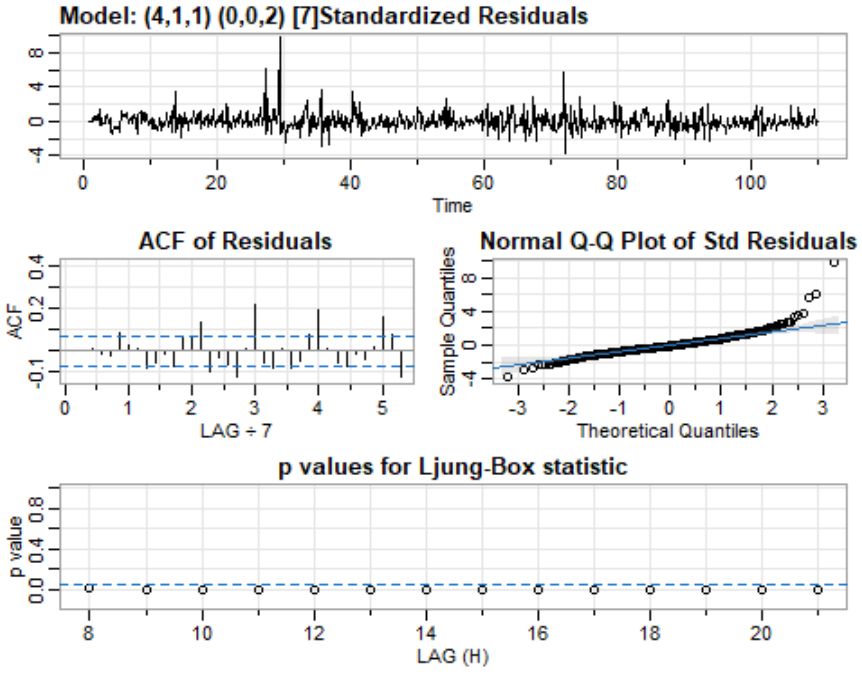

```
wtd_fr_sm2

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1    sar2  constant
##       0.6350  -0.0829  -0.9960  0.0291  0.2913  0.2036    0.0987
## s.e.  0.3719   0.2165   0.3713  0.3602  0.0365  0.0373    0.1308
##
## sigma^2 estimated as 581.1:  log likelihood = -3512.04,  aic = 7040.08
##
## $degrees_of_freedom
## [1] 756
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.6350 0.3719  1.7074  0.0882
## ar2       -0.0829 0.2165 -0.3826  0.7021
## ma1       -0.9960 0.3713 -2.6826  0.0075
## ma2        0.0291 0.3602  0.0809  0.9355
## sar1       0.2913 0.0365  7.9791  0.0000
## sar2       0.2036 0.0373  5.4570  0.0000
## constant   0.0987 0.1308  0.7547  0.4507
##
```

```
## $AIC
## [1] 9.226838
##
## $AICc
## [1] 9.227033
##
## $BIC
## [1] 9.27546
```

```
auto.arima(wtd_fr_train, seasonal = TRUE)
```

```
## Series: wtd_fr_train
## ARIMA(4,1,1)(0,0,2)[7]
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ma1     sma1    sma2
##       0.5972  0.0256  0.1373  -0.0520  -0.9650  0.2860  0.167
## s.e.  0.0379  0.0430  0.0421   0.0371   0.0103  0.0371  0.034
##
## sigma^2 = 2666:  log likelihood = -4089.16
## AIC=8194.32   AICc=8194.51   BIC=8231.41
```

```
wtd_fr_sm2 <- sarima(wtd_zh_train, S = 7,
                     p = 4, d = 1, q = 1,
                     P = 0, D = 0, Q = 2)
```




```
wtd_fr_sm2
```

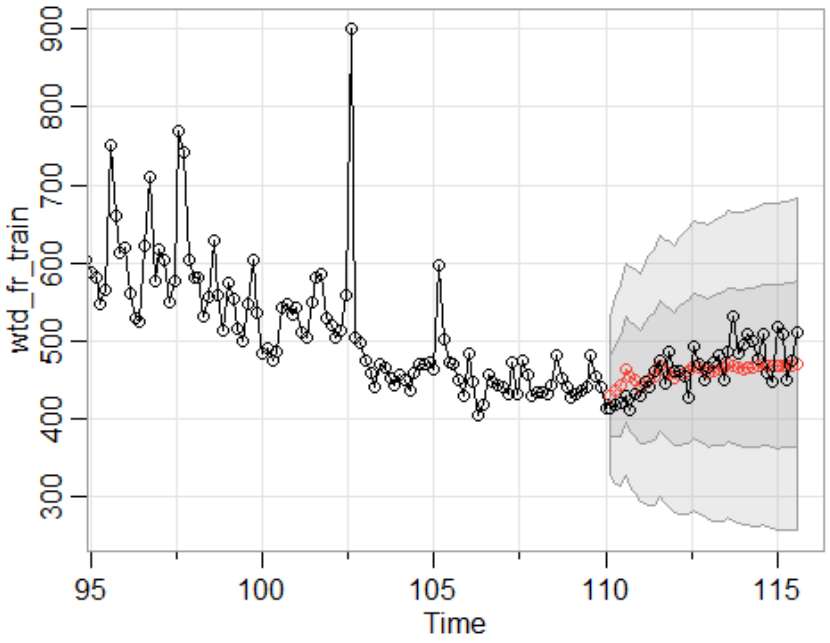
```
## $fit
##
## Call:
```

```
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1    sma1    sma2  constant
##       0.6162  -0.1291  -0.0254  -0.0008  -0.9446  0.2588  0.1362    0.0998
## s.e.  0.0392   0.0439   0.0429   0.0385   0.0151  0.0379  0.0333    0.1304
##
## sigma^2 estimated as 608.4:  log likelihood = -3529.2,  aic = 7076.4
##
## $degrees_of_freedom
## [1] 755
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.6162 0.0392  15.7243  0.0000
## ar2       -0.1291 0.0439  -2.9388  0.0034
## ar3       -0.0254 0.0429  -0.5913  0.5545
## ar4       -0.0008 0.0385  -0.0199  0.9841
## ma1       -0.9446 0.0151 -62.5837  0.0000
## sma1       0.2588 0.0379   6.8288  0.0000
## sma2       0.1362 0.0333   4.0873  0.0000
## constant   0.0998 0.1304   0.7649  0.4446
##
## $AIC
## [1] 9.274443
##
## $AICc
## [1] 9.274693
##
## $BIC
## [1] 9.329142
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : **ARIMA(2,1,2)(2,0,0)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic look better for this model and there is not much relative difference in the AIC value between the two values.

Forecasting:

```
wtd_fr_sm1_for <- sarima.for(wtd_fr_train,n.ahead = 39,S = 7,
                     p = 2, d = 1, q = 2,
                      P = 2, D = 0, Q = 0)
lines(wtd_fr_test, type = 'o')
```



Evaluating the accuracy:

```
accuracy(wtd_fr_sm1_for$pred,x = wtd_fr_test)
```

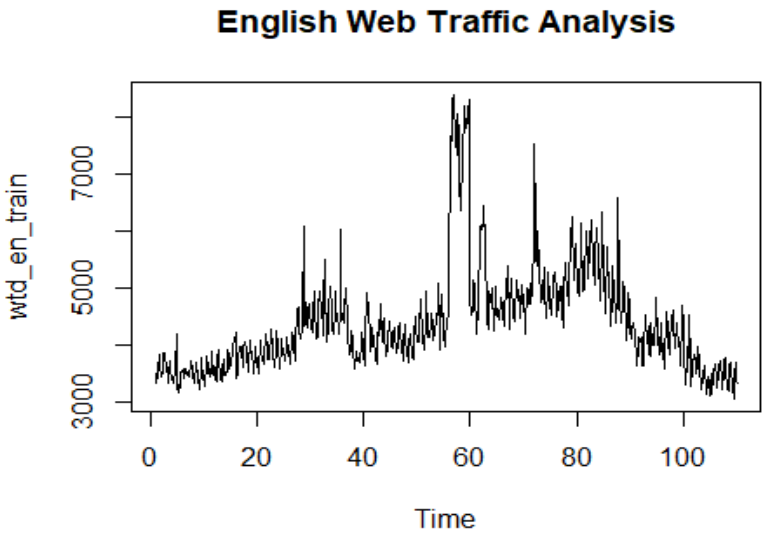
```
##                    ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 5.498776 25.60579 20.87068 0.8478588 4.412506 0.3273653 0.8792066
```

The RMSE of the model is **25.60579**.

English Web Traffic

Splitting the data set into train and test sets:

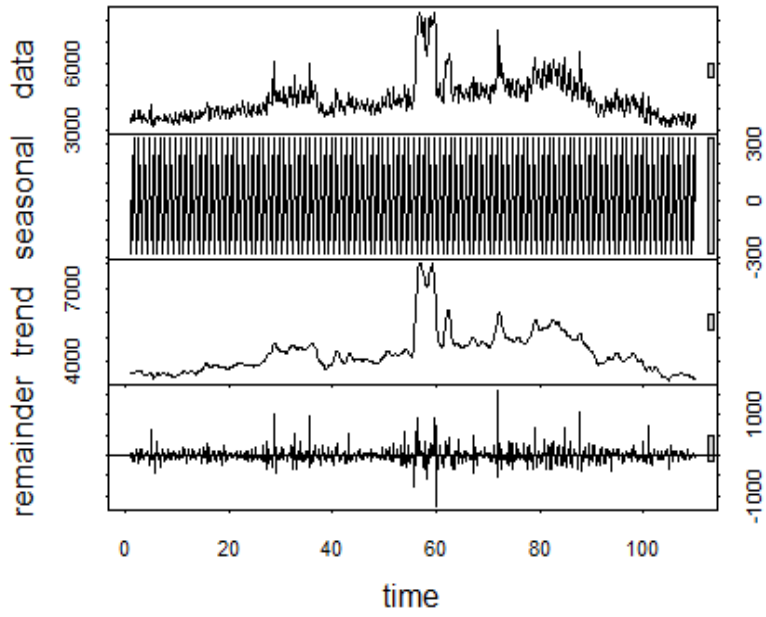
```
wtd_en_train <- window(wtd_en_ts, end = c(110,1))
wtd_en_test <- window(wtd_en_ts, start = c(110,2))
plot(wtd_en_train, main = "English Web Traffic Analysis")
```



Similar to the previous time series, there is an upward trend in the first few months, followed by a very large spike in the traffic. There is seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_en_stl <- stl(wtd_en_train, s.window = "periodic")
plot(wtd_en_stl)
```



Performing the KPSS Test for stationarity:

```
kpss.test(wtd_en_train)
```

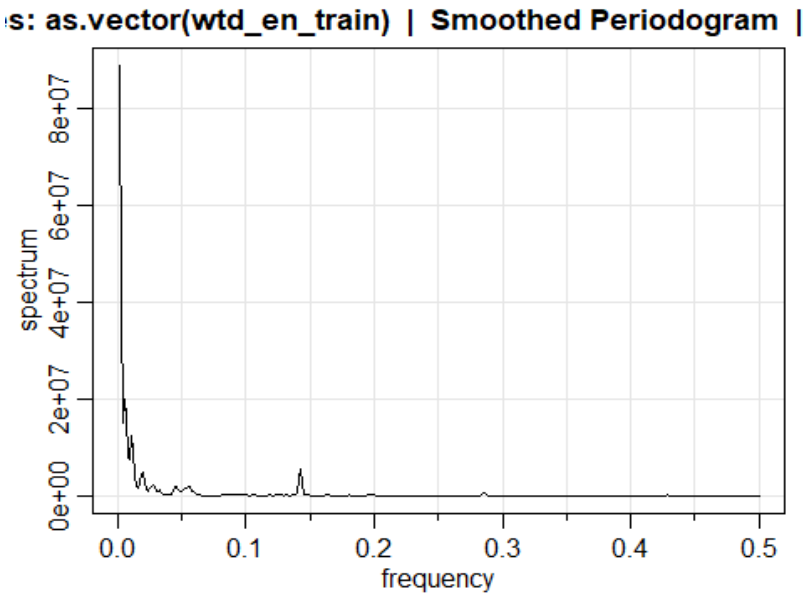
```
## Warning in kpss.test(wtd_en_train): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wtd_en_train
## KPSS Level = 1.996, Truncation lag parameter = 6, p-value = 0.01
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_en.spec <- mvspec(as.vector(wtd_en_train),detrend = TRUE, spans = 3)
```



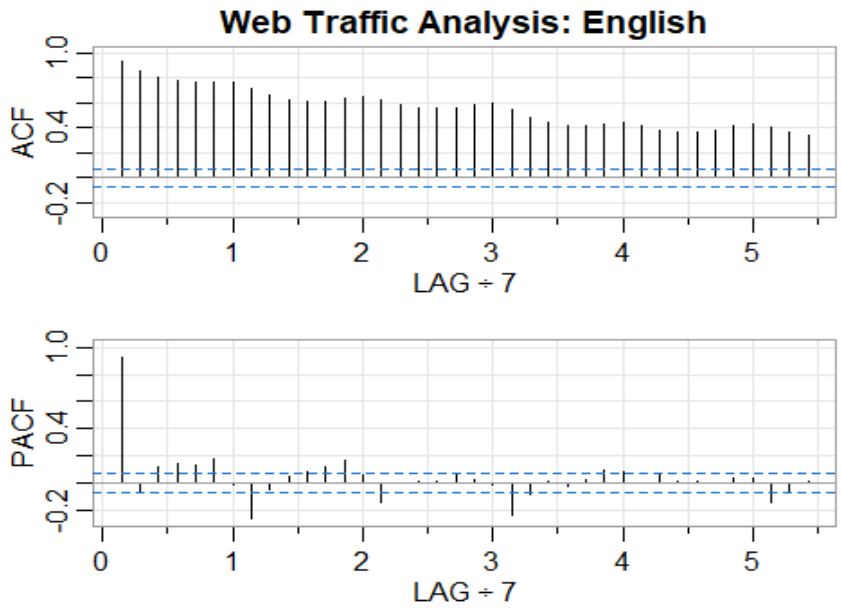

```
head(wtd_en.spec$details)
```

```
##      frequency period spectrum
## [1,]    0.0013  768.0 89107337
## [2,]    0.0026  384.0 39589414
## [3,]    0.0039  256.0 15105724
## [4,]    0.0052  192.0 19981973
## [5,]    0.0065  153.6 16790904
## [6,]    0.0078  128.0  7703258
```

The plot shows one major peak 1/0.14 which is approx. 7 days. This is indicative of a weekly seasonality. The peak at approx. 3 days is hardly noticeable. There are several peaks at the start of plot which is approx. between 20 days to 120 days. This is a stronger indicative of quarterly seasonality.

Plotting the Autocorrelation plot:

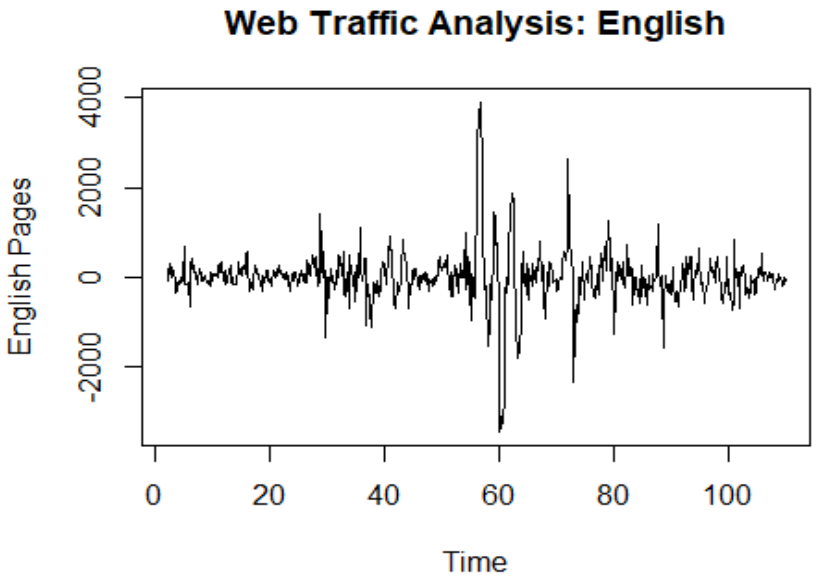
```
acf2(wtd_en_train, main = "Web Traffic Analysis: English")
```



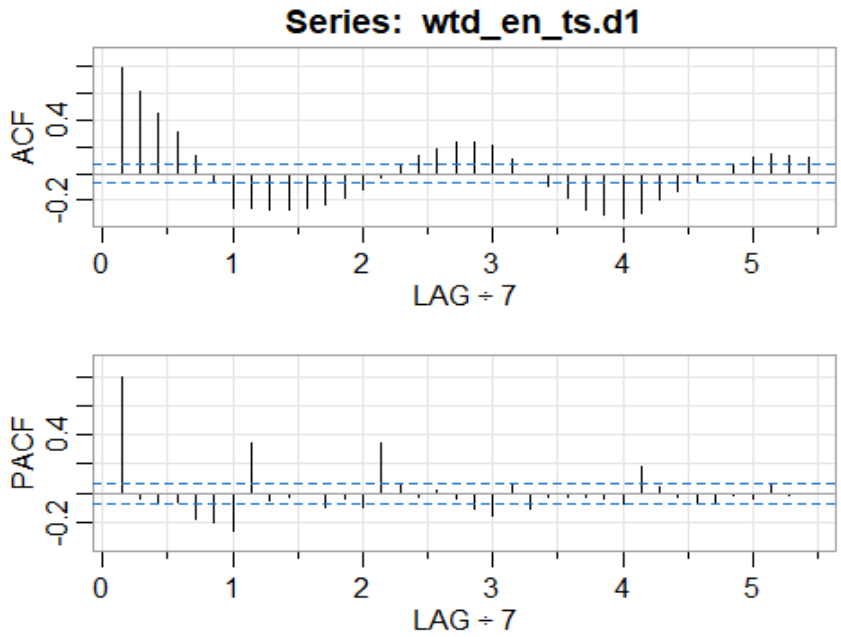
The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

Seasonal Differencing:

```
wtd_en_ts.d1 <- diff(wtd_en_train, lag = 7)
plot(wtd_en_ts.d1,
     main = "Web Traffic Analysis: English",
     ylab = "English Pages", type = 'l')
```



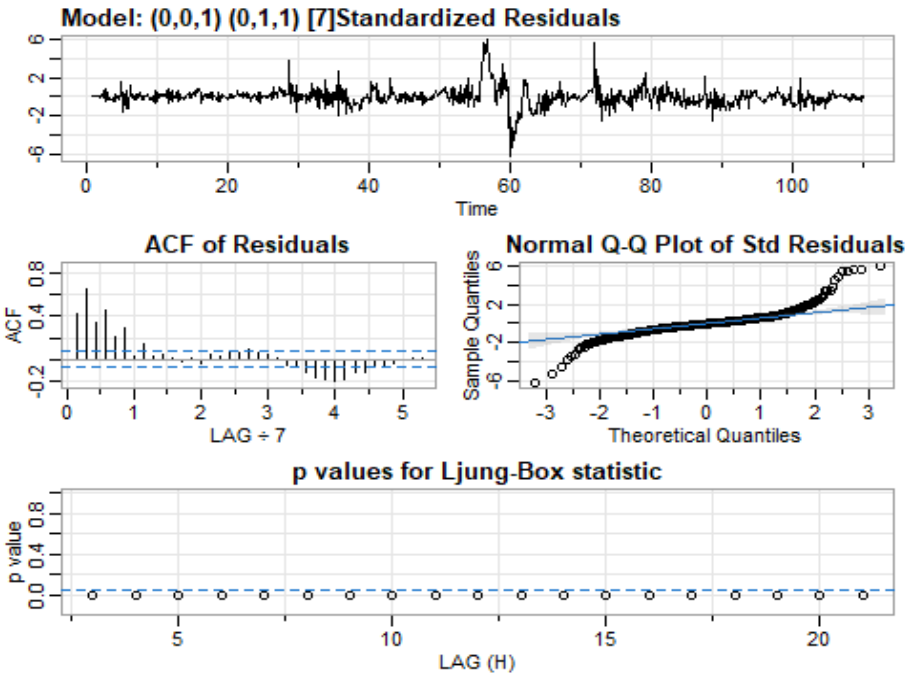
```
kpss.test(wtd_en_ts.d1)

## Warning in kpss.test(wtd_en_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_en_ts.d1
## KPSS Level = 0.069132, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_en_ts.d1)
```



From the plot above, intuitively I would pick the following values: Q = 1 P = 0 D = 1 q = 1 d = 0 p = 0

I would apply the ARIMA(0,0,1)(0,1,1)[7] and run auto ARIMA for this time series.

```
wtd_en_sm1 <- sarima(wtd_en_train, S = 7,
                     p = 0, d = 0, q = 1,
                     P = 0, D = 1, Q = 1)
```

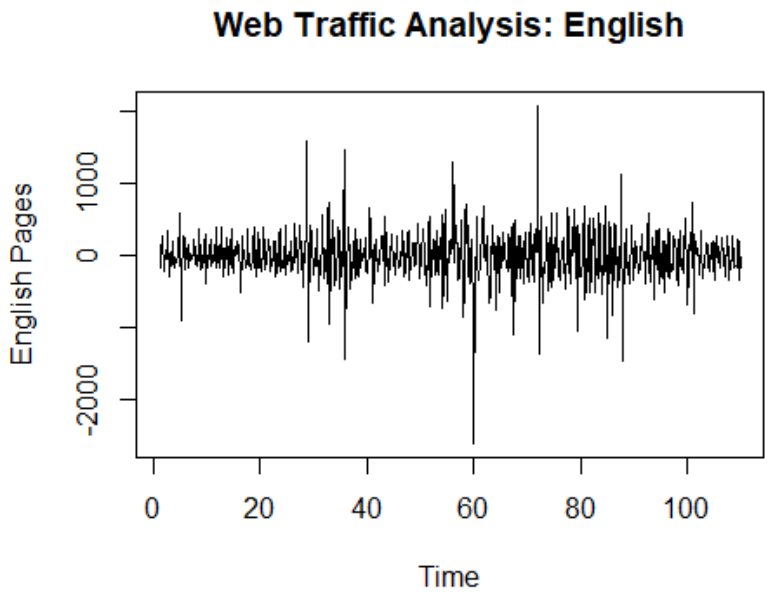



```
wtd_en_sm1

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     sma1  constant
##       0.6585  -0.4934   -0.1959
## s.e.  0.0202   0.0385    1.8540
##
## sigma^2 estimated as 177571:  log likelihood = -5650.35,  aic = 11308.7
##
## $degrees_of_freedom
## [1] 754
##
## $ttable
##          Estimate     SE  t.value p.value
## ma1        0.6585 0.0202  32.5801  0.0000
## sma1      -0.4934 0.0385 -12.8111  0.0000
## constant  -0.1959 1.8540  -0.1057  0.9159
##
## $AIC
## [1] 14.93883
##
```

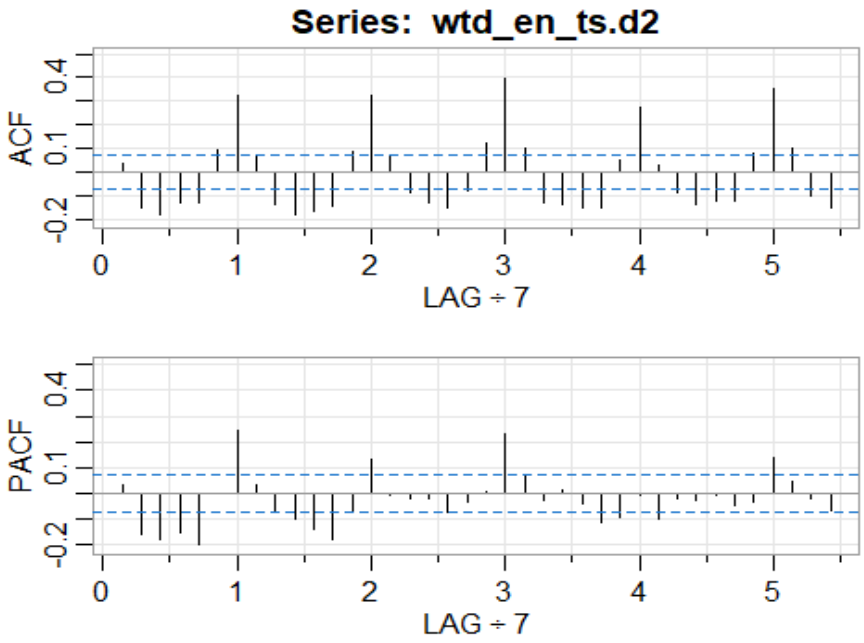
```
## $AICc
## [1] 14.93887
##
## $BIC
## [1] 14.96329
```

First order differencing on training data as the the time series is very noisy:

```
wtd_en_ts.d2 <- diff(wtd_en_train, 1)
plot(wtd_en_ts.d2,
     main = "Web Traffic Analysis: English",
     ylab = "English Pages", type = 'l')
```



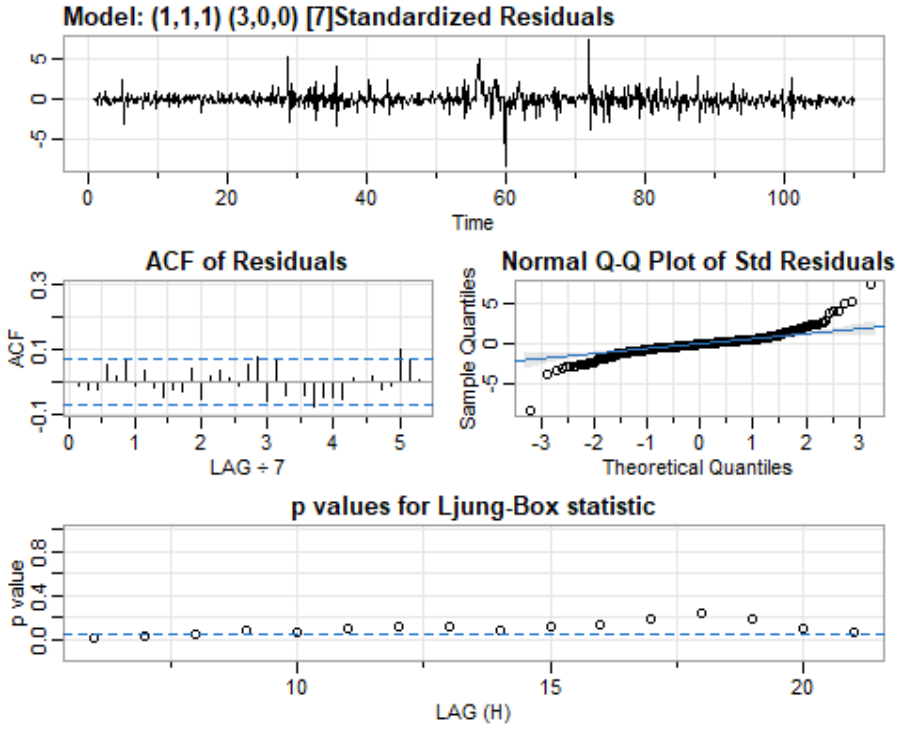

```
kpss.test(wtd_en_ts.d2)

## Warning in kpss.test(wtd_en_ts.d2): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_en_ts.d2
## KPSS Level = 0.043725, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_en_ts.d2)
```



From the plot above, intuitively I would pick the following values: Q = 0 P = 3 D = 0 q = 1 d = 1 p = 1

I would apply the ARIMA(1,1,1)(3,0,0)[7] and run auto ARIMA for this time series.

```
wtd_en_sm4 <- sarima(wtd_en_train, S = 7,
                     p = 1, d = 1, q = 1,
                     P = 3, D = 0, Q = 0)
```



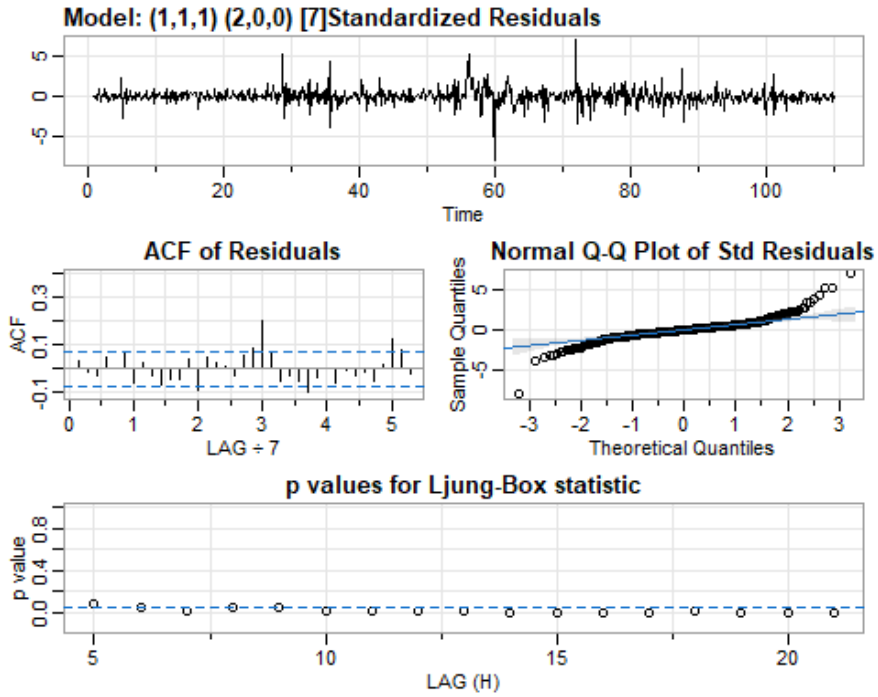
```
wtd_en_sm4
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1    sar1    sar2    sar3  constant
##       0.8810  -0.9935  0.2026  0.1740  0.2750    0.3753
## s.e.  0.0214   0.0083  0.0354  0.0353  0.0352    2.0502
##
## sigma^2 estimated as 84455:  log likelihood = -5412.22,  aic = 10838.44
##
## $degrees_of_freedom
## [1] 757
##
## $ttable
##          Estimate     SE   t.value p.value
## ar1        0.8810 0.0214   41.1808  0.0000
## ma1       -0.9935 0.0083 -120.1453  0.0000
## sar1       0.2026 0.0354    5.7250  0.0000
## sar2       0.1740 0.0353    4.9363  0.0000
## sar3       0.2750 0.0352    7.8182  0.0000
## constant   0.3753 2.0502    0.1831  0.8548
##
## $AIC
## [1] 14.20503
##
## $AICc
## [1] 14.20518
##
## $BIC
## [1] 14.24757
```

```
auto.arima(wtd_en_train, seasonal = TRUE)
```

```
## Series: wtd_en_train
## ARIMA(1,1,1)(2,0,0)[7]
##
## Coefficients:
##          ar1      ma1    sar1    sar2
##       0.8548  -0.9853  0.2712  0.2408
## s.e.  0.0223   0.0074  0.0357  0.0354
##
```

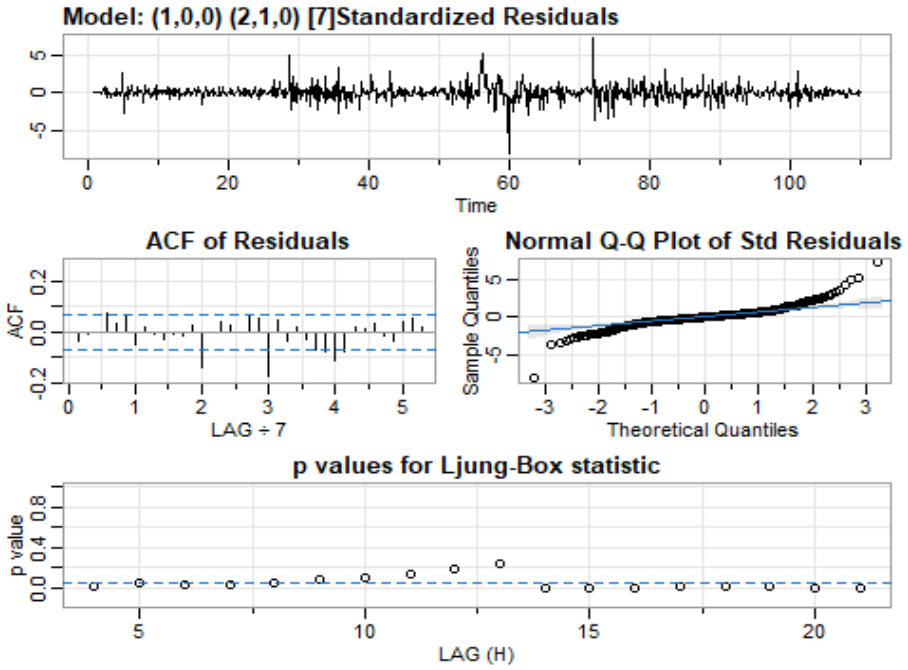
```
## sigma^2 = 91927:  log likelihood = -5441.73
## AIC=10893.46   AICc=10893.54   BIC=10916.64

wtd_en_sm2 <- sarima(wtd_en_train, S = 7,
                     p = 1, d = 1, q = 1,
                     P = 2, D = 0, Q = 0)
```




```
wtd_en_sm2

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1    sar1    sar2  constant
##       0.8549  -0.9854  0.2713  0.2408    0.1408
## s.e.  0.0224   0.0075  0.0357  0.0354    2.4419
##
## sigma^2 estimated as 91444:  log likelihood = -5441.73,  aic = 10895.45
##
## $degrees_of_freedom
## [1] 758
##
## $ttable
```

```
##          Estimate     SE   t.value p.value
## ar1        0.8549 0.0224   38.2313   0.000
## ma1       -0.9854 0.0075 -131.2706   0.000
## sar1       0.2713 0.0357    7.6043   0.000
## sar2       0.2408 0.0354    6.7930   0.000
## constant   0.1408 2.4419    0.0577   0.954
##
## $AIC
## [1] 14.27976
##
## $AICc
## [1] 14.27986
##
## $BIC
## [1] 14.31622
```

```
wtd_en_sm3 <- sarima(wtd_en_train, S = 7,
                     p = 1, d = 0, q = 0,
                     P = 2, D = 1, Q = 0)
```




```
wtd_en_sm3
```

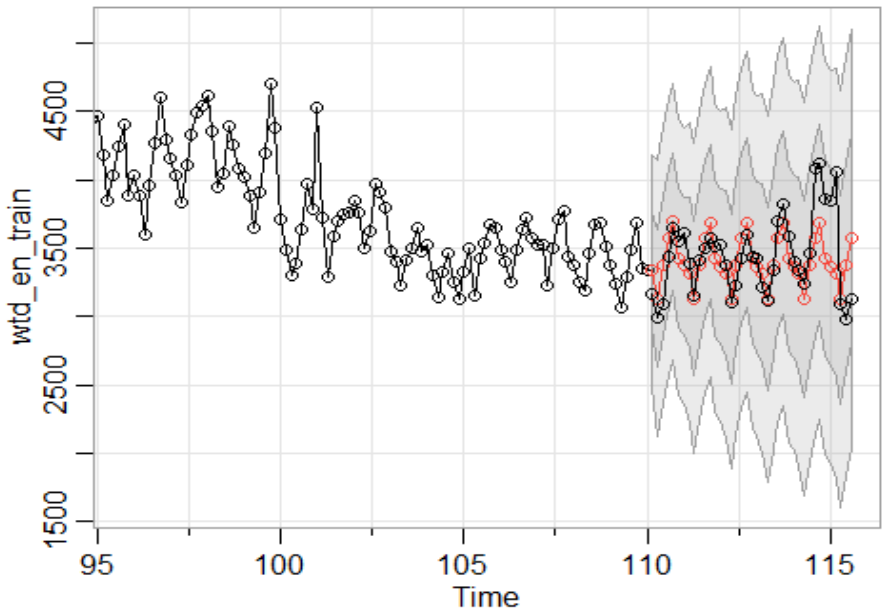
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
```

```
##
## Coefficients:
##          ar1     sar1     sar2  constant
##       0.8738  -0.6752  -0.3876   -0.2673
## s.e.  0.0179   0.0339   0.0334    6.0315
##
## sigma^2 estimated as 91937:  log likelihood = -5402.49,  aic = 10814.98
##
## $degrees_of_freedom
## [1] 753
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.8738 0.0179  48.8005  0.0000
## sar1      -0.6752 0.0339 -19.9312  0.0000
## sar2      -0.3876 0.0334 -11.6176  0.0000
## constant  -0.2673 6.0315  -0.0443  0.9647
##
## $AIC
## [1] 14.28663
##
## $AICc
## [1] 14.2867
##
## $BIC
## [1] 14.3172
```

Looking at the above plots, I have decided to go ahead with model: **ARIMA(0,0,1)(0,1,1)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic don't look great and there is not much relative difference in the AIC value between the models. However, this model has the least number of non-significant terms. Hence, I chose this model.

Forecasting:

```
wtd_en_sm1_for <- sarima.for(wtd_en_train,n.ahead = 39,S = 7,
                      p = 0, d = 0, q = 1,
                     P = 0, D = 1, Q = 1)
lines(wtd_en_test, type = 'o')
```



Evaluating accuracy:

```
accuracy(wtd_en_sm1_for$pred,x = wtd_en_test)
```

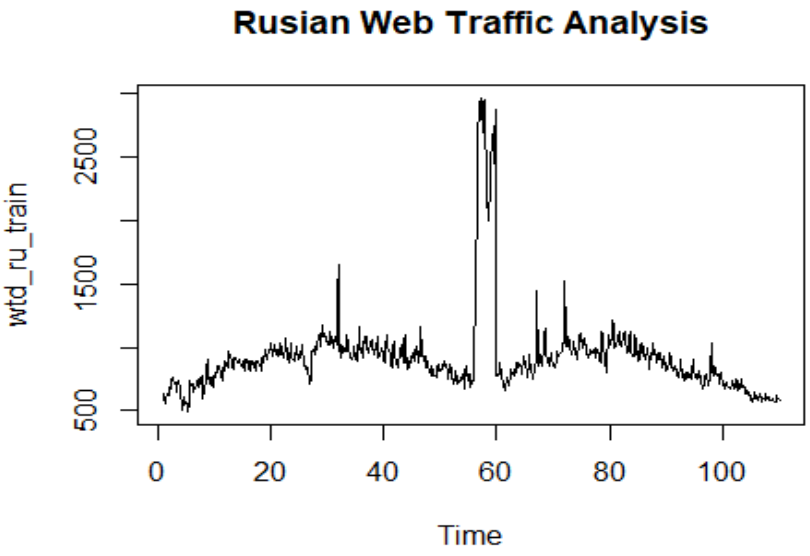
```
##                  ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set 45.47113 238.4469 168.597 0.8815891 4.742445 0.6281423 0.9501652
```

The RMSE value is **238.4469**.

Russian Web Traffic

Splitting the data set into train and test sets:

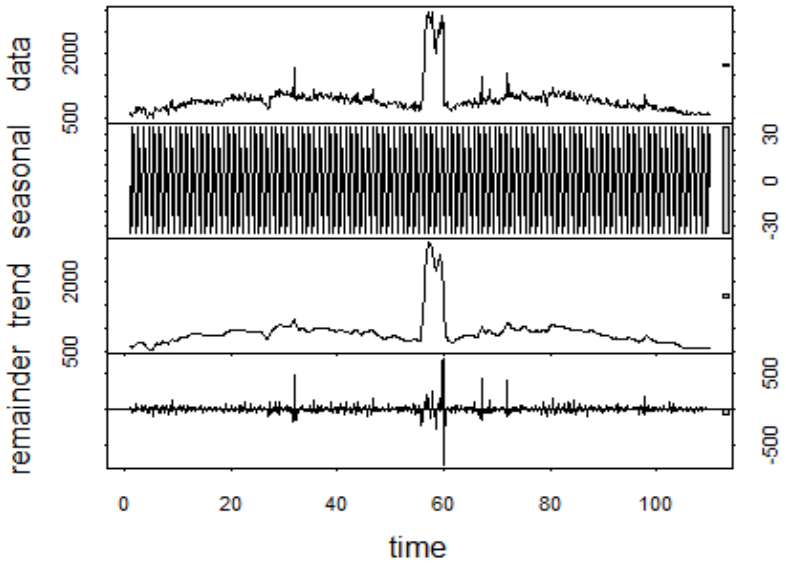
```
wtd_ru_train <- window(wtd_ru_ts, end = c(110,1))
wtd_ru_test <- window(wtd_ru_ts, start = c(110,2))
plot(wtd_ru_train, main = "Rusian Web Traffic Analysis")
```



There is a little bit of upward trend in the first few months, followed by an extremely large spike in the traffic. There is seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_ru_stl <- stl(wtd_ru_train, s.window = "periodic")
plot(wtd_ru_stl)
```



Performing the KPSS Test for stationarity:

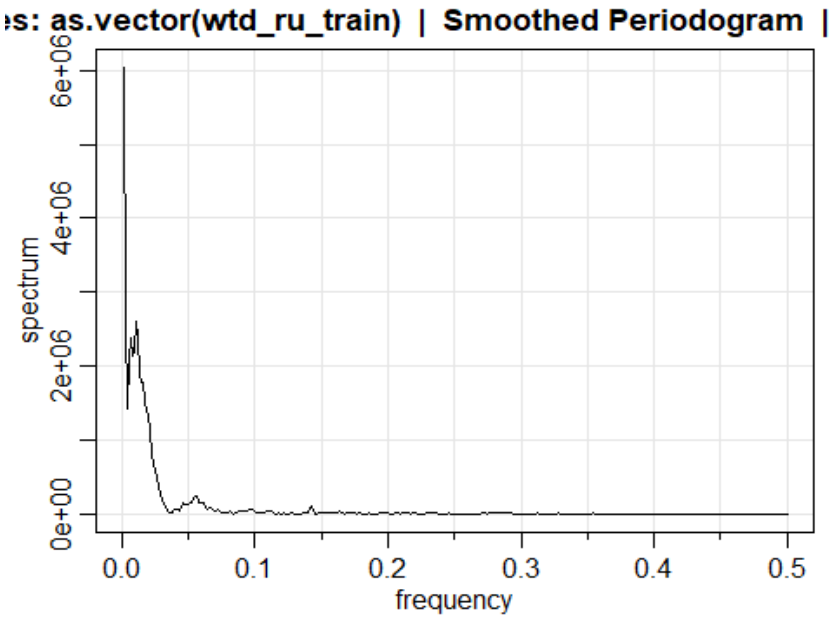
```
kpss.test(wtd_ru_train)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wtd_ru_train
## KPSS Level = 0.62669, Truncation lag parameter = 6, p-value = 0.02021
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_ru.spec <- mvspec(as.vector(wtd_ru_train),detrend = TRUE, spans = 2)
```

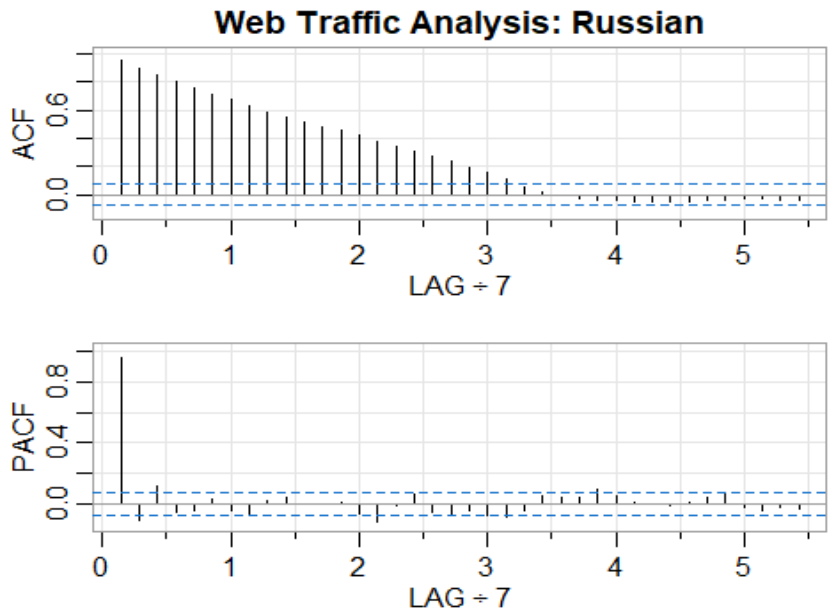



```
head(wtd_ru.spec$details)

##      frequency period spectrum
## [1,]    0.0013  768.0  6045476
## [2,]    0.0026  384.0  2658592
## [3,]    0.0039  256.0  1433215
## [4,]    0.0052  192.0  2068443
## [5,]    0.0065  153.6  2384748
## [6,]    0.0078  128.0  2142030
```

The plot shows a small peak 1/0.14 which is approx. 7 days. This is indicative of a slight weekly seasonality. There are several peaks at the start of plot which is approx. between 20 days to 120 days. This is a strong indicative of quarterly seasonality.

Plotting the Autocorrelation plot:

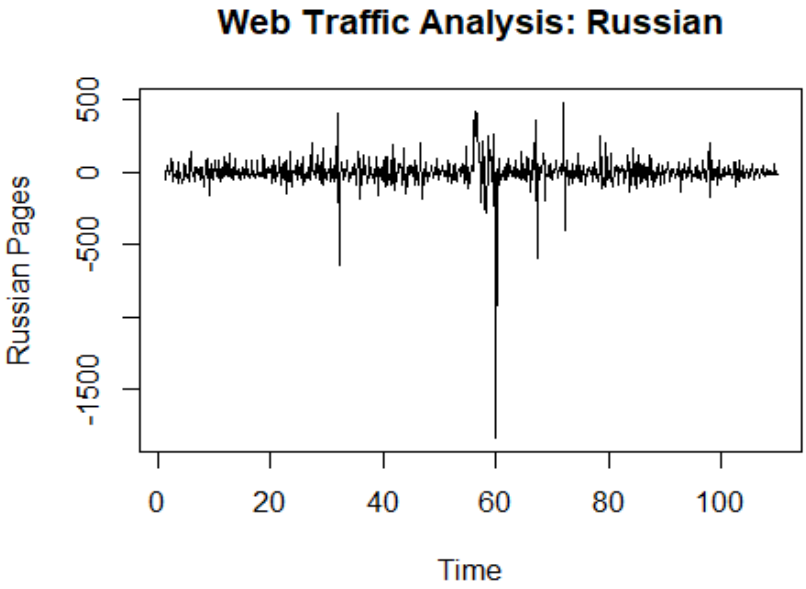
```
acf2(wtd_ru_train, main = "Web Traffic Analysis: Russian")
```



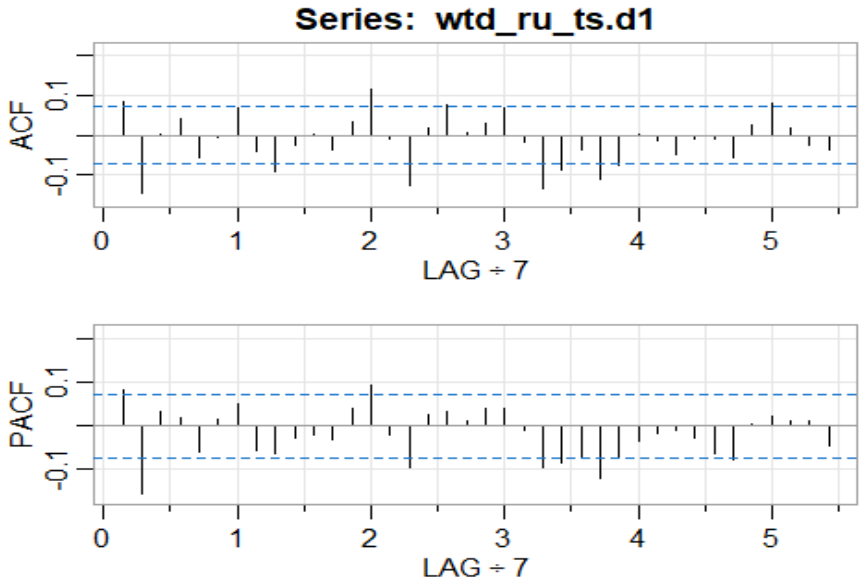
The autocorrelations plot is much different from the other plots that I have seen so far. This cannot be interpreted as an obvious weekly seasonality. However, there is an obvious correlation among the lags. In this case, I have decided to apply the non-seasonal differencing first and check if that is enough to make the time series is stationary.

Non Seasonal Differencing:

```
wtd_ru_ts.d1 <- diff(wtd_ru_train, lag = 1)
plot(wtd_ru_ts.d1,
     main = "Web Traffic Analysis: Russian",
     ylab = "Russian Pages", type = 'l')
```



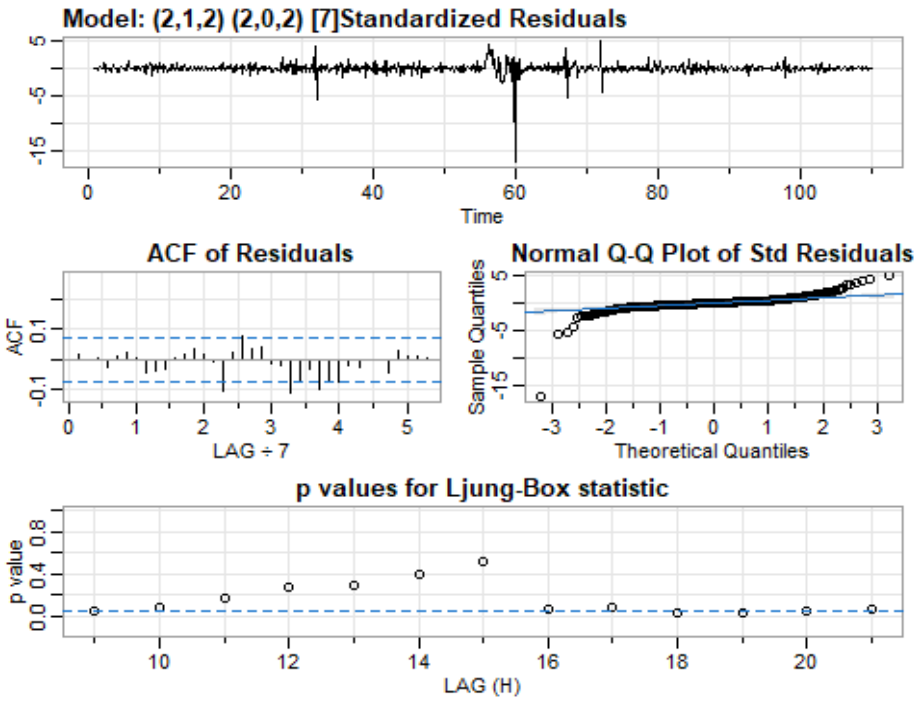
```
kpss.test(wtd_ru_ts.d1)

## Warning in kpss.test(wtd_ru_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_ru_ts.d1
## KPSS Level = 0.026637, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_ru_ts.d1)
```



From the plot above, intuitively I would pick the following values: d = 1 p = 2 q = 2 D = 0 Q = 2 P = 2

I would apply the ARIMA(2,1,2)(2,0,2)[7] to fit the model as well as run the auto ARIMA to see if there are better fits to the models.

```
wtd_ru_sm1 <- sarima(wtd_ru_train, S = 7,
                     p = 2, d = 1, q = 2,
                     P = 2, D = 0, Q = 2)
```



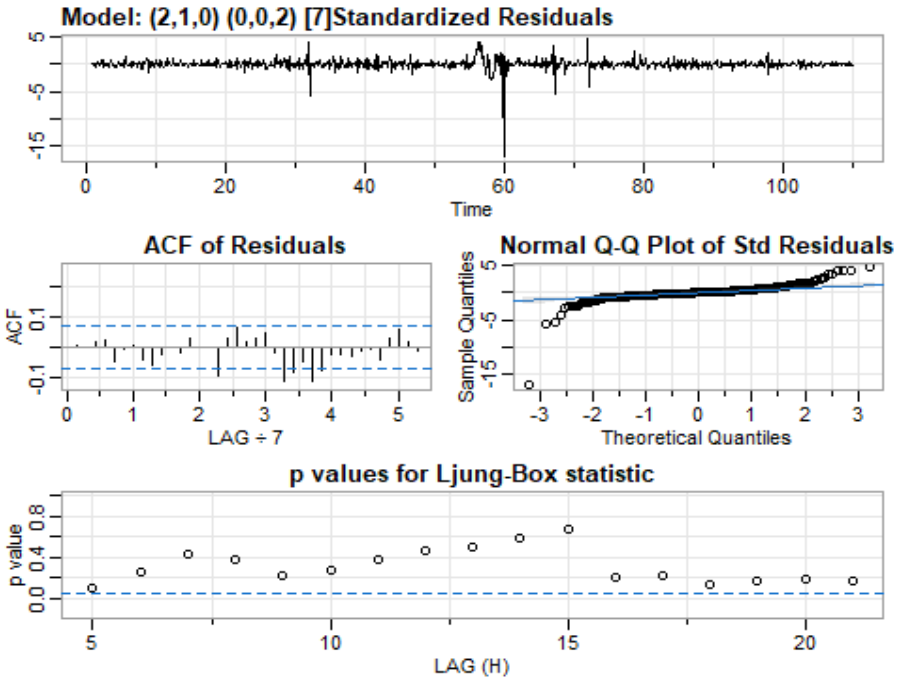

```
wtd_ru_sm1

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ar2     ma1     ma2    sar1    sar2     sma1     sma2
##       -0.4653  -0.6908  0.5500  0.6047  0.6637  0.3352  -0.7059  -0.2816
## s.e.   0.1059   0.1763  0.1234  0.1792  0.4181  0.4182   0.4305   0.4288
##       constant
##         0.1495
## s.e.    9.3339
##
## sigma^2 estimated as 10350:  log likelihood = -4613.61,  aic = 9247.21
##
## $degrees_of_freedom
## [1] 754
##
## $ttable
##          Estimate     SE t.value p.value
## ar1       -0.4653 0.1059 -4.3955  0.0000
## ar2       -0.6908 0.1763 -3.9185  0.0001
## ma1        0.5500 0.1234  4.4590  0.0000
```

```
## ma2        0.6047 0.1792  3.3752  0.0008
## sar1       0.6637 0.4181  1.5872  0.1129
## sar2       0.3352 0.4182  0.8015  0.4231
## sma1      -0.7059 0.4305 -1.6398  0.1015
## sma2      -0.2816 0.4288 -0.6567  0.5116
## constant   0.1495 9.3339  0.0160  0.9872
##
## $AIC
## [1] 12.11955
##
## $AICc
## [1] 12.11986
##
## $BIC
## [1] 12.18032
```

```
auto.arima(wtd_ru_train)
```

```
## Series: wtd_ru_train
## ARIMA(2,1,0)(0,0,2)[7]
##
## Coefficients:
##          ar1      ar2   sma1    sma2
##       0.0969  -0.1383  0.042  0.0979
## s.e.  0.0359   0.0361  0.036  0.0377
##
## sigma^2 = 10862:  log likelihood = -4626.03
## AIC=9262.06   AICc=9262.14   BIC=9285.25
```

```
wtd_ru_sm2 <- sarima(wtd_ru_train, S = 7,
                     p = 2, d = 1, q = 0,
                     P = 0, D = 0, Q = 2)
```



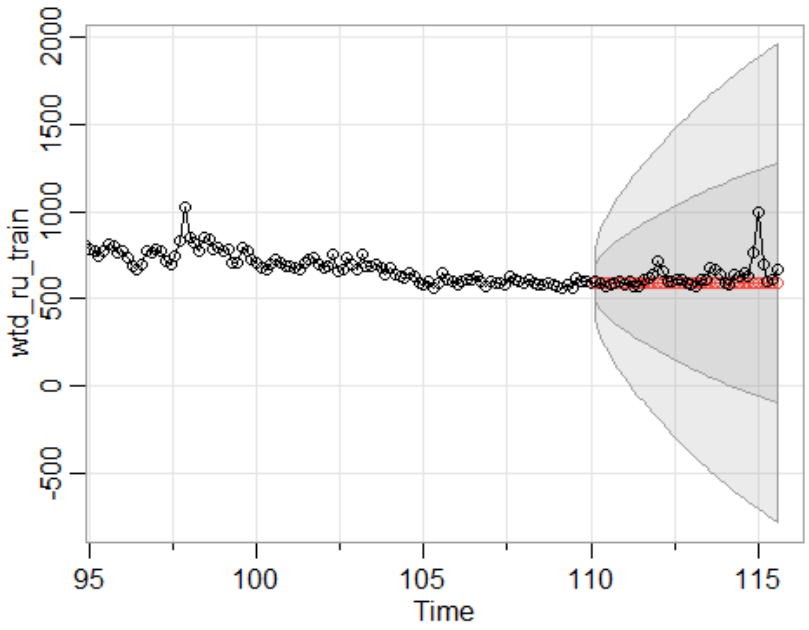

```
wtd_ru_sm2

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2   sma1    sma2  constant
##       0.0969  -0.1384  0.042  0.0979   -0.0150
## s.e.  0.0359   0.0361  0.036  0.0377    4.1123
##
## sigma^2 estimated as 10805:  log likelihood = -4626.03,  aic = 9264.06
##
## $degrees_of_freedom
## [1] 758
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.0969 0.0359  2.6982  0.0071
## ar2       -0.1384 0.0361 -3.8292  0.0001
## sma1       0.0420 0.0360  1.1690  0.2428
## sma2       0.0979 0.0377  2.6011  0.0095
## constant  -0.0150 4.1123 -0.0036  0.9971
##
```

```
## $AIC
## [1] 12.14163
##
## $AICc
## [1] 12.14173
##
## $BIC
## [1] 12.1781
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : **ARIMA(2,1,0)(0,0,2)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic look better for this model and there is not much relative difference in the AIC value between the models.

Forecasting:

```
wtd_ru_sm1_for <- sarima.for(wtd_ru_train,n.ahead = 39,S = 7,
                      p = 2, d = 1, q = 0,
                      P = 0, D = 0, Q = 2)
lines(wtd_ru_test, type = 'o')
```



Evaluating accuracy:

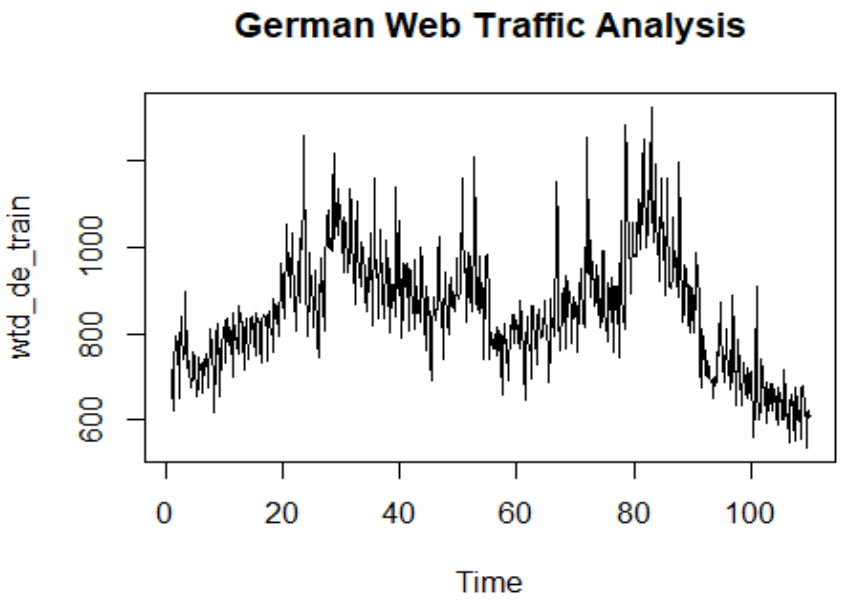
```
accuracy(wtd_ru_sm1_for$pred,x = wtd_ru_test)

##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 38.96517 82.93603 44.56494 5.237445 6.214016 0.4994408  1.244766
```

The RMSE error is **82.93603**.

German Web Traffic

Splitting the data set into train and test sets:

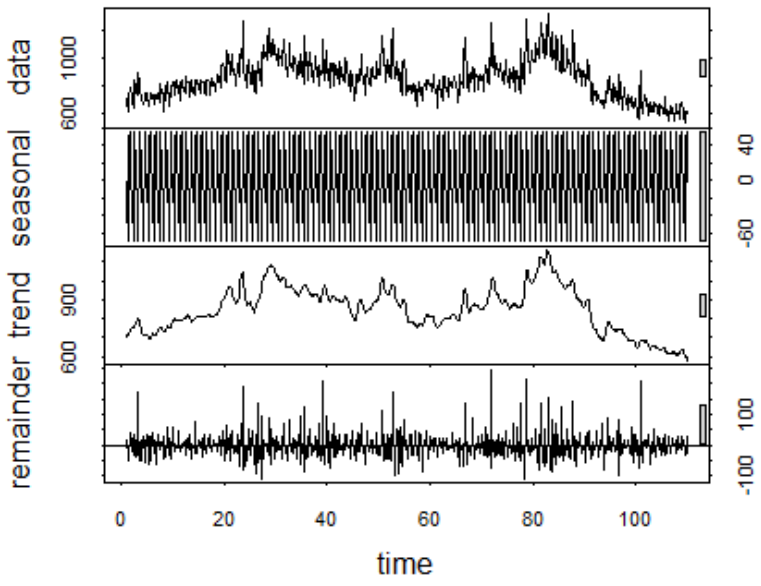
```
wtd_de_train <- window(wtd_de_ts, end = c(110,1))
wtd_de_test <- window(wtd_de_ts, start = c(110,2))
plot(wtd_de_train, main = "German Web Traffic Analysis")
```



Similar to the previous time series, there is a noticeable upward trend in the first few months, followed by a slightly downward trend and then another upward trend. There is seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_de_stl <- stl(wtd_de_train, s.window = "periodic")
plot(wtd_de_stl)
```



Performing the KPSS stationarity test:

```
kpss.test(wtd_de_train)
```

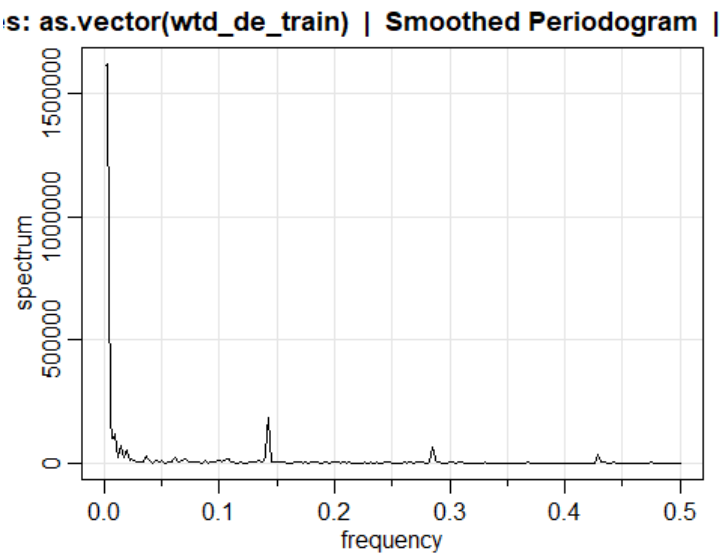
```
## Warning in kpss.test(wtd_de_train): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  wtd_de_train
## KPSS Level = 1.4295, Truncation lag parameter = 6, p-value = 0.01
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_de.spec <- mvspec(as.vector(wtd_de_train),detrend = TRUE, spans = 2)
```

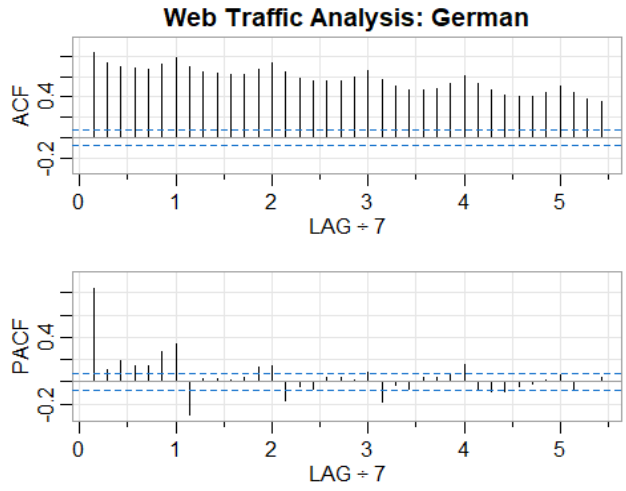



```
head(wtd_de.spec$details)

##      frequency period  spectrum
## [1,]    0.0013  768.0 1611499.2
## [2,]    0.0026  384.0 1620003.8
## [3,]    0.0039  256.0  785023.8
## [4,]    0.0052  192.0  187164.2
## [5,]    0.0065  153.6  102578.9
## [6,]    0.0078  128.0  101037.3
```

The plot shows a big peak at 1/0.14 which is approx. 7 days. This is indicative of a weekly seasonality. There are several small peaks at the start of plot which is approx. between 20 days to 120 days. This is a indicative of some kind of quarterly seasonality. There are also two peaks at 1/0.28 and 1/0.43 which is approx. 2-3 days - there seems to be some mid-weekly seasonality as well.

Plotting the autocorrelation plot:

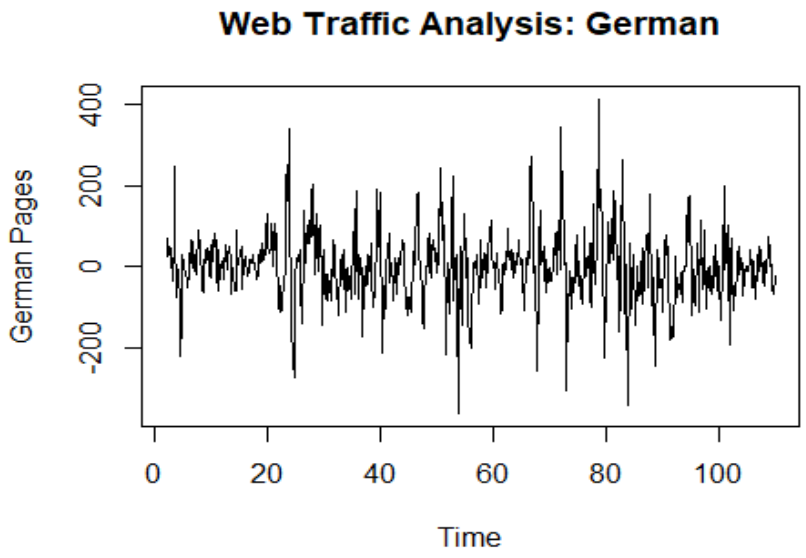
```
acf2(wtd_de_train, main = "Web Traffic Analysis: German")
```



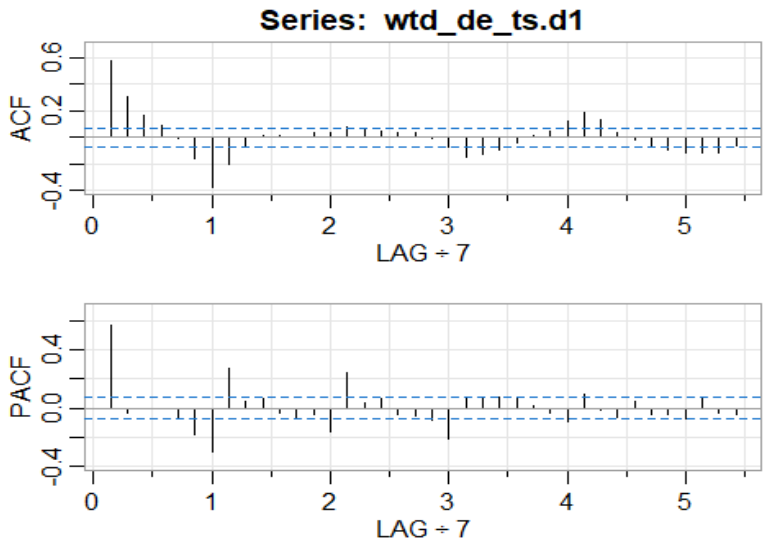
The autocorrelations show a high lag every 7 days which is an indication of a weekly seasonality.

Seasonal Differencing:

```
wtd_de_ts.d1 <- diff(wtd_de_train, lag = 7)
plot(wtd_de_ts.d1,
     main = "Web Traffic Analysis: German",
     ylab = "German Pages", type = 'l')
```



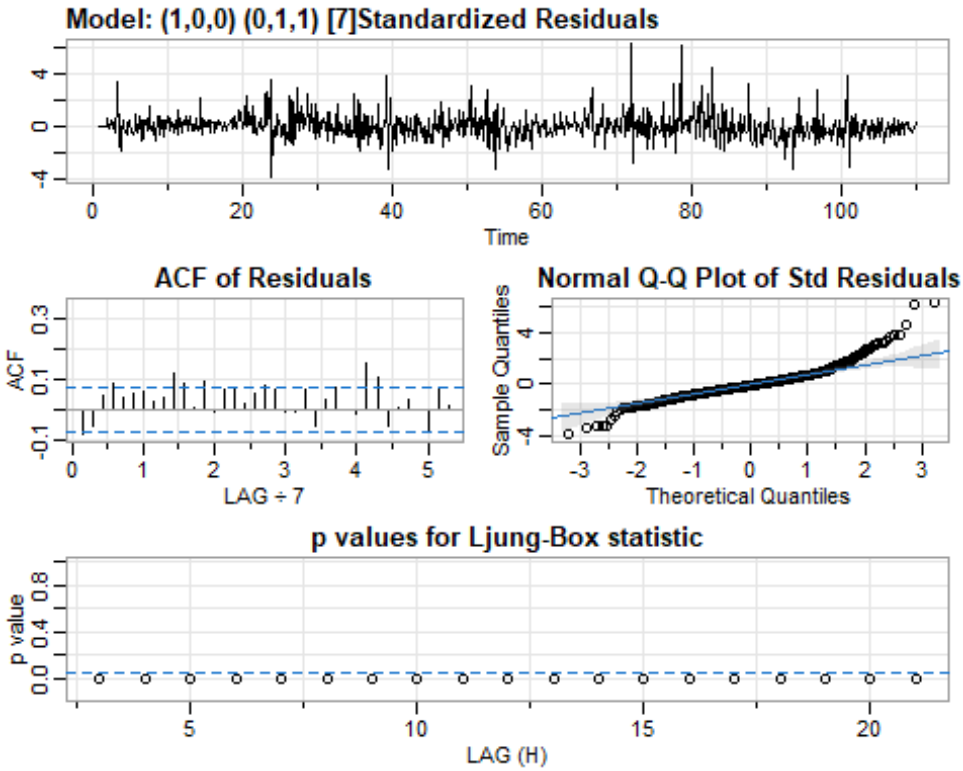

```
kpss.test(wtd_de_ts.d1)

## Warning in kpss.test(wtd_de_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_de_ts.d1
## KPSS Level = 0.18169, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_de_ts.d1)
```



From the plot above, intuitively I would pick the following values: Q= 1 P = 0 D = 1 q = 4/0 p = 1 d = 0

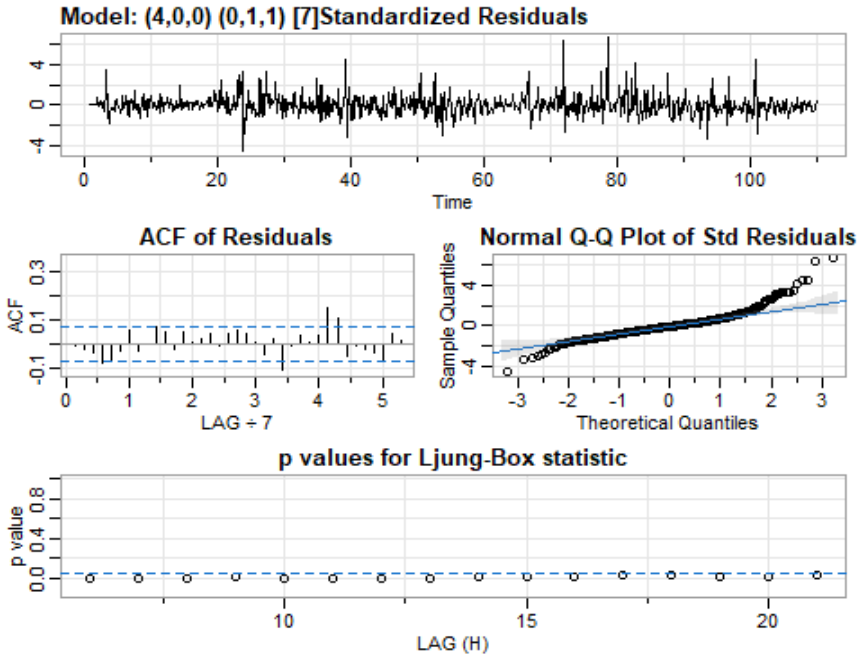
I would apply ARIMA(1,0,0)(0,1,1)[7] and run auto ARIMA on the model.

```
wtd_de_sm1 <- sarima(wtd_de_train, S = 7,
                     p = 1, d = 0, q = 0,
                     P = 0, D = 1, Q = 1)
```




```
wtd_de_sm1
```

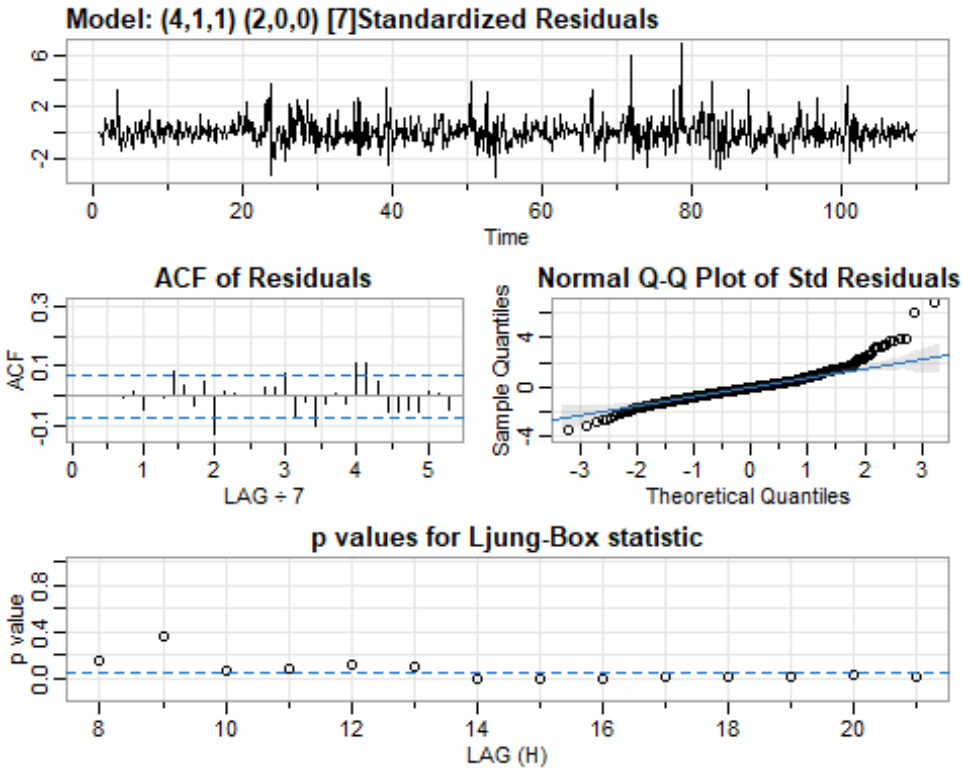
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     sma1  constant
##       0.7517  -0.7777   -0.1510
## s.e.  0.0320   0.0357    0.2704
##
## sigma^2 estimated as 3193:  log likelihood = -4131.74,  aic = 8271.47
##
## $degrees_of_freedom
## [1] 754
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.7517 0.0320  23.4669  0.0000
## sma1      -0.7777 0.0357 -21.7672  0.0000
## constant  -0.1510 0.2704  -0.5585  0.5767
##
## $AIC
## [1] 10.92665
##
## $AICc
## [1] 10.92669
##
## $BIC
## [1] 10.95111

auto.arima(wtd_de_train, D=1)

## Series: wtd_de_train
## ARIMA(5,0,0)(0,1,1)[7]
##
## Coefficients:
##          ar1     ar2     ar3     ar4     ar5     sma1
##       0.6567  0.0128  0.1019  0.0761  0.0692  -0.9086
## s.e.  0.0366  0.0434  0.0432  0.0432  0.0375   0.0380
##
## sigma^2 = 3015:  log likelihood = -4109.33
## AIC=8232.66   AICc=8232.8   BIC=8265.06

wtd_de_sm2 <- sarima(wtd_de_train,S = 7,
                     p = 4, d = 0, q = 0,
                     P = 0, D = 1, Q = 1)
```




```
wtd_de_sm2

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2     ar3     ar4     sma1  constant
##       0.6613  0.0177  0.1024  0.1172  -0.8913   -0.1408
## s.e.  0.0367  0.0433  0.0432  0.0370   0.0357    0.3242
##
## sigma^2 estimated as 3008:  log likelihood = -4110.93,  aic = 8235.87
##
## $degrees_of_freedom
## [1] 751
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.6613 0.0367  18.0353  0.0000
## ar2        0.0177 0.0433   0.4085  0.6830
## ar3        0.1024 0.0432   2.3682  0.0181
## ar4        0.1172 0.0370   3.1667  0.0016
## sma1      -0.8913 0.0357 -24.9609  0.0000
## constant  -0.1408 0.3242  -0.4344  0.6641
##
```

```
## $AIC
## [1] 10.87961
##
## $AICc
## [1] 10.87976
##
## $BIC
## [1] 10.92242
```

```
wtd_de_sm3 <- sarima(wtd_de_train,S = 7,
                     p = 4, d = 1, q = 1,
                     P = 2, D = 0, Q = 0)
```




```
wtd_de_sm3
```

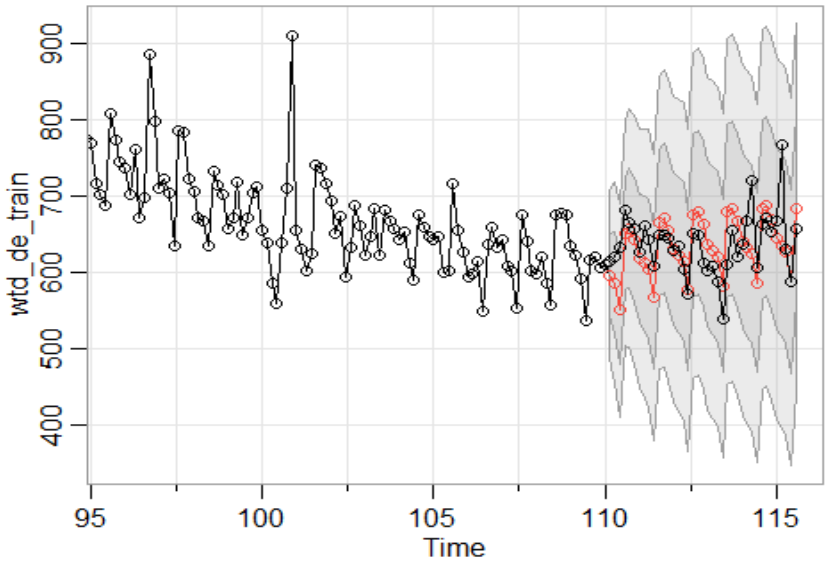
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ma1    sar1    sar2  constant
##       0.5813  -0.0269  0.0322  0.0220  -0.9630  0.3736  0.2666   -0.1432
```

```
## s.e.  0.0387   0.0424  0.0422  0.0378   0.0141  0.0357  0.0356    0.5540
##
## sigma^2 estimated as 3337:  log likelihood = -4179.38,  aic = 8376.77
##
## $degrees_of_freedom
## [1] 755
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.5813 0.0387  15.0085  0.0000
## ar2       -0.0269 0.0424  -0.6345  0.5259
## ar3        0.0322 0.0422   0.7622  0.4462
## ar4        0.0220 0.0378   0.5828  0.5602
## ma1       -0.9630 0.0141 -68.1508  0.0000
## sar1       0.3736 0.0357  10.4520  0.0000
## sar2       0.2666 0.0356   7.4799  0.0000
## constant  -0.1432 0.5540  -0.2584  0.7962
##
## $AIC
## [1] 10.97873
##
## $AICc
## [1] 10.97898
##
## $BIC
## [1] 11.03342
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : **ARIMA(4,0,0)(0,1,1)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic look better for this model (although all look equally bad) and there is not much relative difference in the AIC value between the models.

Forecasting:

```
wtd_de_sm1_for <- sarima.for(wtd_de_train,n.ahead = 39,S = 7,
                      p = 4, d = 0, q = 0,
                      P = 0, D = 1, Q = 1)
lines(wtd_de_test, type = 'o')
```



Evaluating accuracy:

```
accuracy(wtd_de_sm1_for$pred,x = wtd_de_test)
```

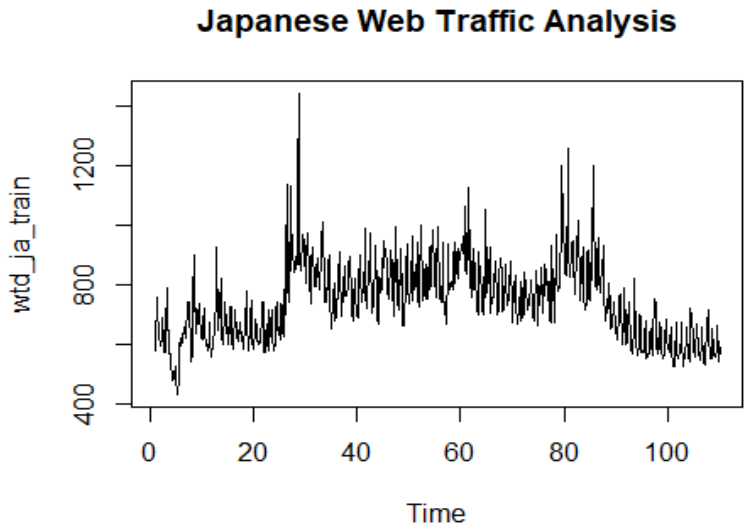
```
##                    ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 2.777242 40.43782 30.37245 0.2268837 4.684784 0.5040598 0.8823819
```

The RMSE value is **40.43782**.

Japanese Web Traffic

Splitting the data set into train and test sets:

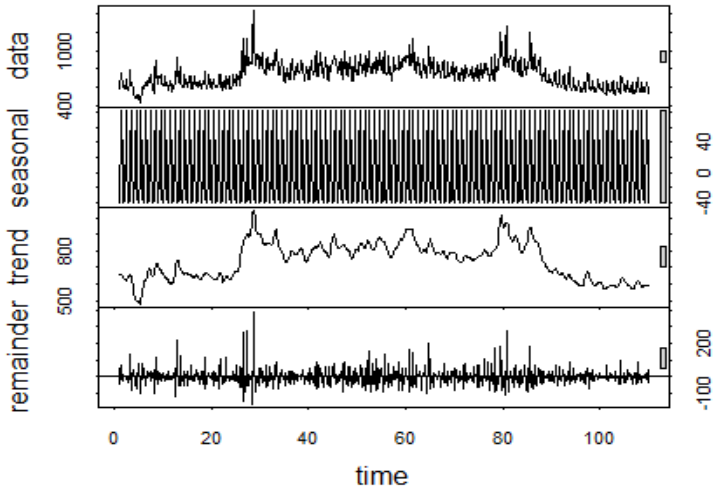
```
wtd_ja_train <- window(wtd_ja_ts, end = c(110,1))
wtd_ja_test <- window(wtd_ja_ts, start = c(110,2))
plot(wtd_ja_train, main = "Japanese Web Traffic Analysis")
```



The upward trend in the first few months is not as noticeable as the previous time series but there is large spike in the traffic. There is seasonality in the data. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_ja_stl <- stl(wtd_ja_train, s.window = "periodic")
plot(wtd_ja_stl)
```



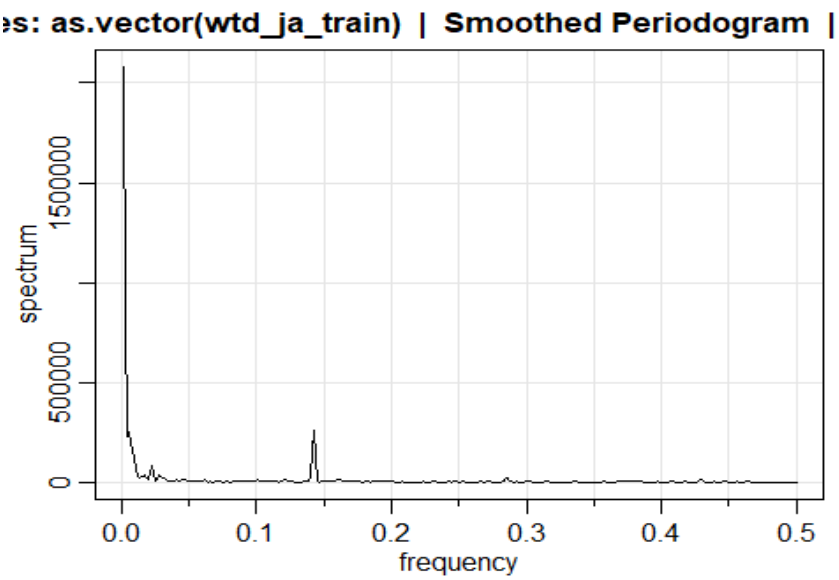
Performing the KPSS stationarity test:

```
kpss.test(wtd_ja_train)

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

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_ja_train
## KPSS Level = 1.6809, Truncation lag parameter = 6, p-value = 0.01
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_ja.spec <- mvspec(as.vector(wtd_ja_train),detrend = TRUE, spans = 3)
```



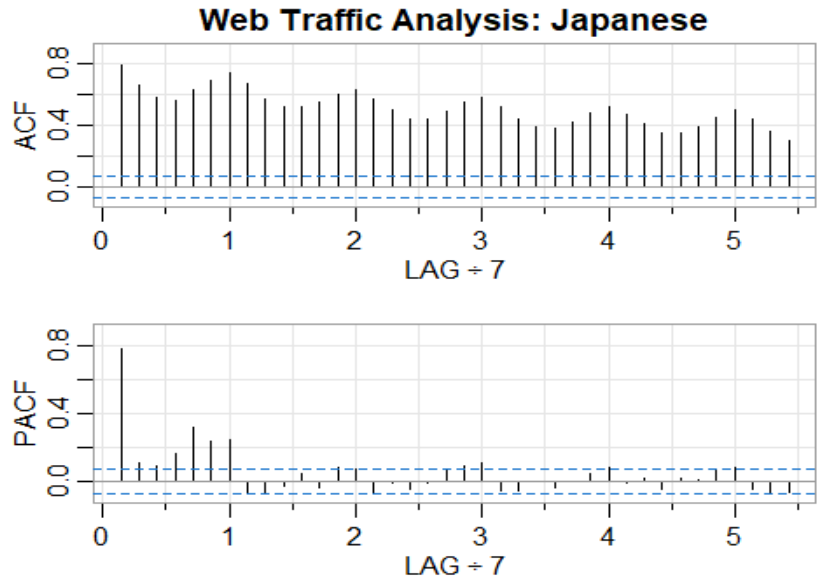

```
head(wtd_ja.spec$details)
```

```
##      frequency period  spectrum
## [1,]    0.0013  768.0 2081618.1
## [2,]    0.0026  384.0  855121.0
## [3,]    0.0039  256.0  228172.7
## [4,]    0.0052  192.0  254728.1
## [5,]    0.0065  153.6  206727.7
## [6,]    0.0078  128.0  163959.6
```

The plot shows a big peak at 1/0.14 which is approx. 7 days. This is indicative of a weekly seasonality. There are some small peaks at the start of plot which is approx. between 20 days to 120 days which is a indicative of some kind of quarterly seasonality. There are no other major peaks in the plot.

Plotting the autocorrelation plot:

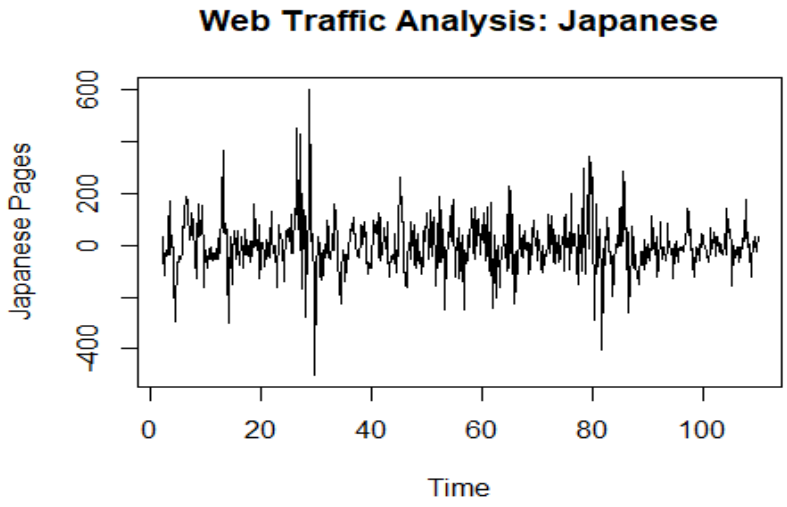
```
acf2(wtd_ja_train, main = "Web Traffic Analysis: Japanese")
```



The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

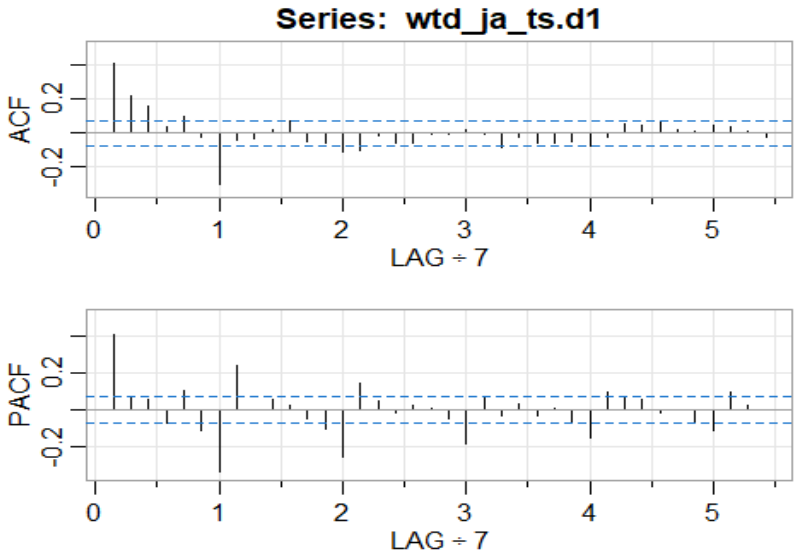
Seasonal Differencing:

```
wtd_ja_ts.d1 <- diff(wtd_ja_train, lag = 7)
plot(wtd_ja_ts.d1,
     main = "Web Traffic Analysis: Japanese",
     ylab = "Japanese Pages", type = 'l')
```



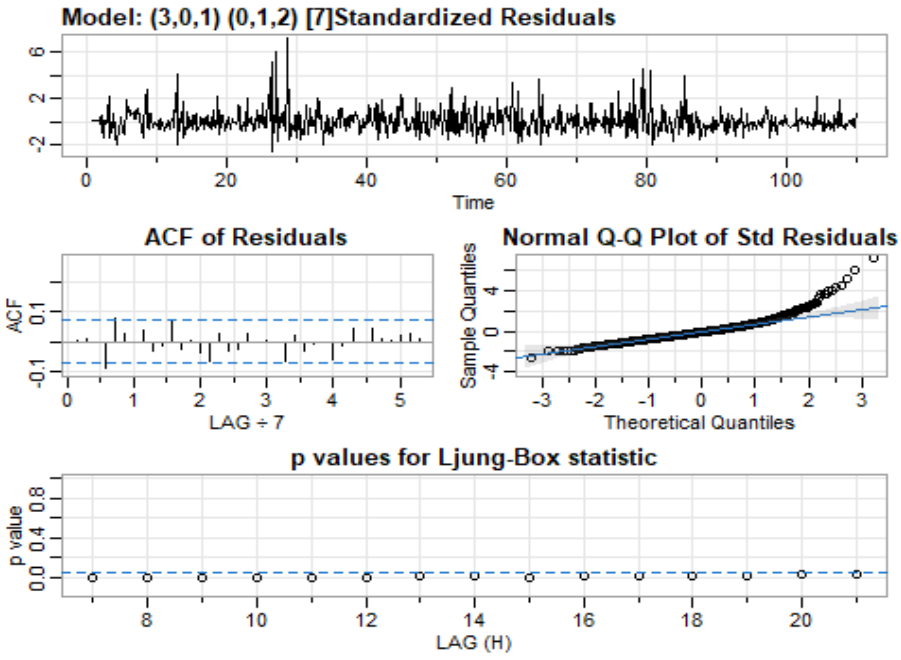

```
kpss.test(wtd_ja_ts.d1)

## Warning in kpss.test(wtd_ja_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
```

```
##
## data:  wtd_ja_ts.d1
## KPSS Level = 0.089372, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_ja_ts.d1)
```



From the plot above, intuitively I would pick the following values: D = 1 P = 0 Q = 2 d = 0 p = 3 q = 1 I would apply ARIMA(3,0,1)(0,1,2)[7] and run auto ARIMA to find a good for the model.

```
wtd_ja_sm1 <- sarima(wtd_ja_train,S = 7,
                     p = 3, d = 0, q = 1,
                     P = 0, D = 1, Q = 2)
```



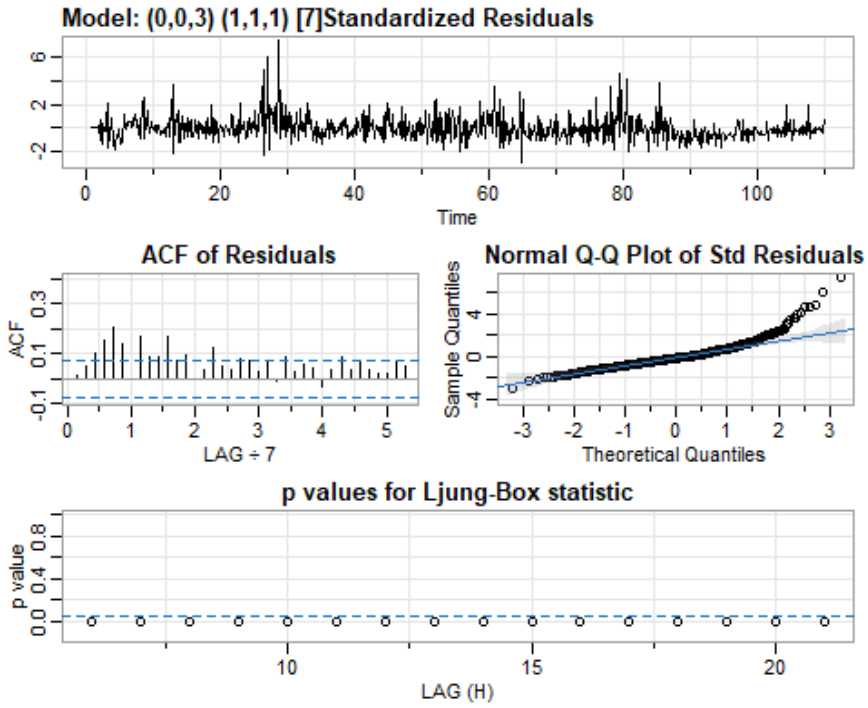
```
wtd_ja_sm1
```

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sma1     sma2  constant
##       1.3024  -0.3057  -0.0041  -0.8272  -0.9078  -0.0921   -0.0350
## s.e.  0.0561   0.0603   0.0449   0.0429   0.0690   0.0397    0.2003
##
## sigma^2 estimated as 4145:  log likelihood = -4241.49,  aic = 8498.98
##
## $degrees_of_freedom
## [1] 750
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        1.3024 0.0561  23.2307  0.0000
## ar2       -0.3057 0.0603  -5.0732  0.0000
## ar3       -0.0041 0.0449  -0.0902  0.9281
## ma1       -0.8272 0.0429 -19.2807  0.0000
## sma1      -0.9078 0.0690 -13.1587  0.0000
## sma2      -0.0921 0.0397  -2.3193  0.0206
## constant  -0.0350 0.2003  -0.1747  0.8614
##
## $AIC
## [1] 11.22719
##
## $AICc
## [1] 11.22738
##
## $BIC
## [1] 11.27611
```

```
auto.arima(wtd_ja_train, D = 1)
```

```
## Series: wtd_ja_train
## ARIMA(0,0,5)(1,1,1)[7]
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5    sar1     sma1
##       0.4679  0.3584  0.2477  0.1089  0.1904  0.1727  -0.8584
## s.e.  0.0433  0.0417  0.0361  0.0378  0.0397  0.0485   0.0247
##
```

```
## sigma^2 = 4823:  log likelihood = -4284.73
## AIC=8585.46   AICc=8585.66   BIC=8622.5

wtd_ja_sm2 <- sarima(wtd_ja_train,S = 7,
                     p = 0, d = 0, q = 3,
                     P = 1, D = 1, Q = 1)
```



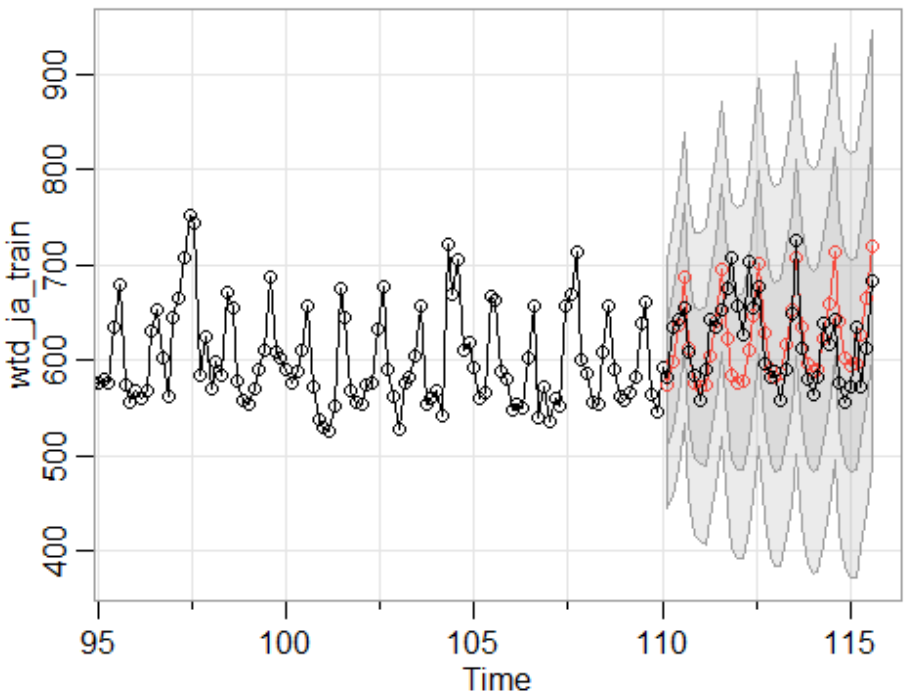

```
wtd_ja_sm2

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     ma2     ma3    sar1     sma1  constant
##       0.5364  0.3554  0.2131  0.1533  -0.8376   -0.0322
## s.e.  0.0382  0.0406  0.0317  0.0467   0.0273    0.1543
##
## sigma^2 estimated as 4937:  log likelihood = -4296.65,  aic = 8607.3
##
## $degrees_of_freedom
## [1] 751
##
## $ttable
```

```
##          Estimate     SE  t.value p.value
## ma1        0.5364 0.0382  14.0440  0.0000
## ma2        0.3554 0.0406   8.7612  0.0000
## ma3        0.2131 0.0317   6.7293  0.0000
## sar1       0.1533 0.0467   3.2804  0.0011
## sma1      -0.8376 0.0273 -30.7112  0.0000
## constant  -0.0322 0.1543  -0.2088  0.8346
##
## $AIC
## [1] 11.37028
##
## $AICc
## [1] 11.37043
##
## $BIC
## [1] 11.41309
```

Looking at the above plots, I have decided to go ahead with my intuitive model: **ARIMA(3,0,1)(0,1,2)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic look better for this model and AIC value is also much lesser for this model.

Forecasting:

```
wtd_ja_sm1_for <- sarima.for(wtd_ja_train,n.ahead = 39,S = 7,
                      p = 3, d = 0, q = 1,
                     P = 0, D = 1, Q = 2)
lines(wtd_ja_test, type = 'o')
```



Evaluating accuracy:

```
accuracy(wtd_ja_sm1_for$pred,x = wtd_ja_test)
```

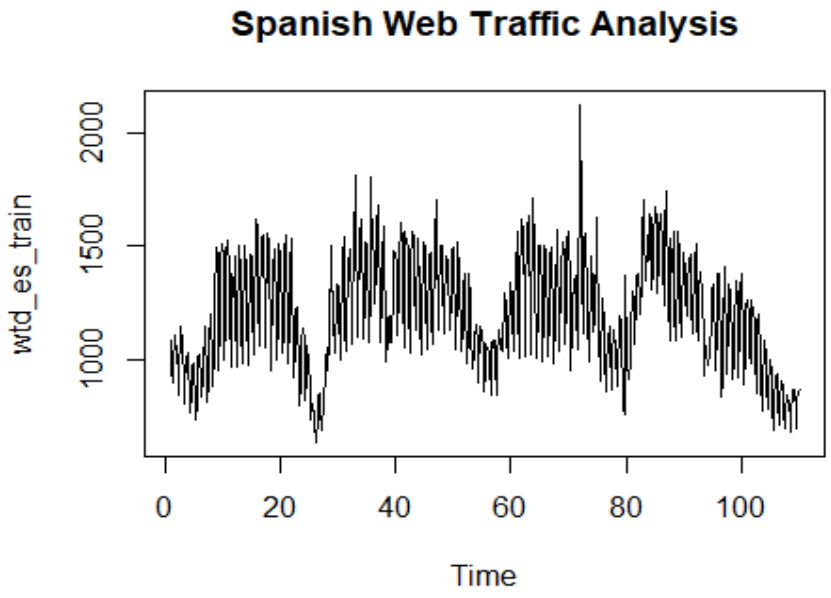
```
##                 ME     RMSE      MAE        MPE     MAPE     ACF1 Theil's U
## Test set -2.05563 42.10344 32.93384 -0.5722104 5.217705 0.553135 0.9180255
```

The RMSE value is **42.10344**.

Spanish Web Traffic

Splitting the data set into train and test sets:

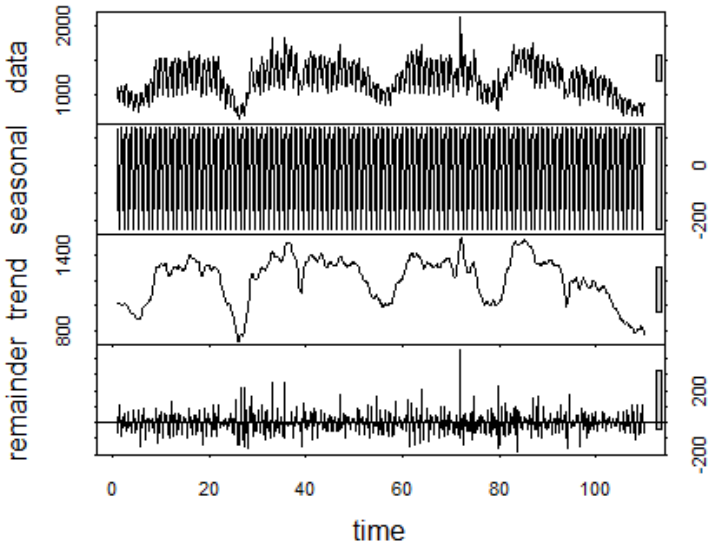
```
wtd_es_train <- window(wtd_es_ts, end = c(110,1))
wtd_es_test <- window(wtd_es_ts, start = c(110,2))
plot(wtd_es_train, main = "Spanish Web Traffic Analysis")
```



There is high seasonality in the data and some spikes in the traffic. The time series does not seem to be stationary.

STL Decomposition:

```
wtd_es_stl <- stl(wtd_es_train, s.window = "periodic")
plot(wtd_es_stl)
```



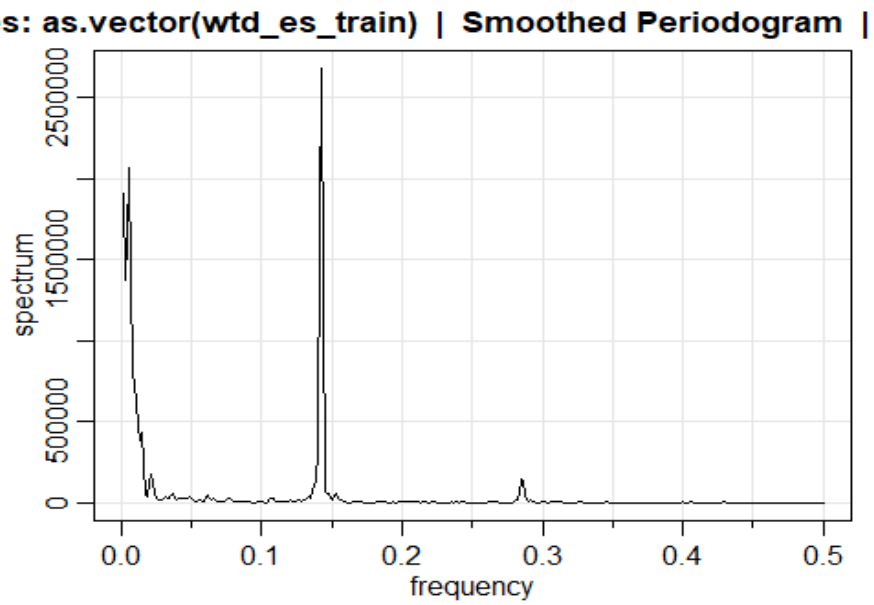
Performing the KPSS test for stationarity:

```
kpss.test(wtd_es_train)

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_es_train
## KPSS Level = 0.62072, Truncation lag parameter = 6, p-value = 0.02075
```

The p-value is less than 0.05, thus we reject the null hypothesis. The time series is not stationary.

Spectral Analysis:

```
wtd_es.spec <- mvspec(as.vector(wtd_es_train),detrend = TRUE, spans = 3)
```



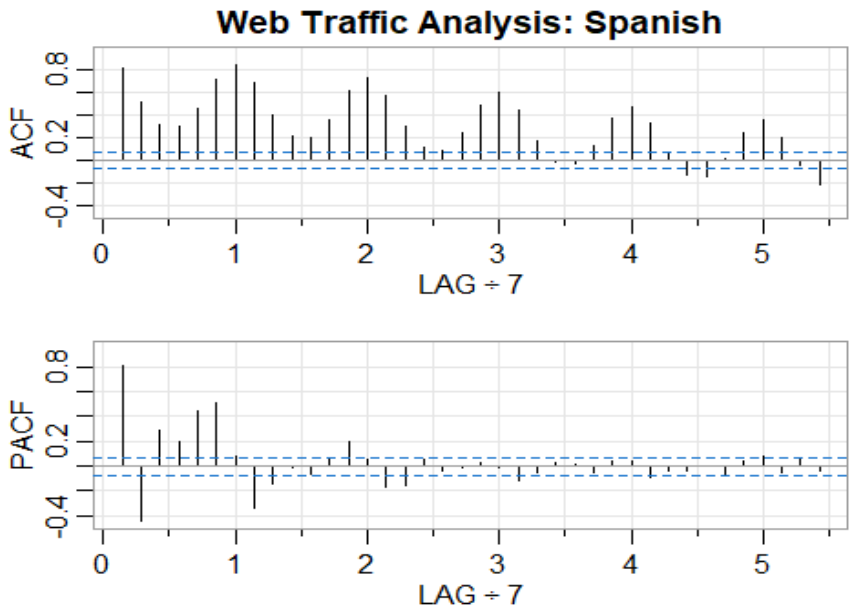

```
head(wtd_es.spec$details)
```

```
##      frequency period  spectrum
## [1,]    0.0013  768.0 1909842.9
## [2,]    0.0026  384.0 1375373.8
## [3,]    0.0039  256.0 1608348.2
## [4,]    0.0052  192.0 2069047.1
## [5,]    0.0065  153.6 1402507.5
## [6,]    0.0078  128.0  813704.4
```

The plot shows an extremely large peak at 1/0.14 which is approx. 7 days. This is indicative of a high weekly seasonality. There is also a small peak at approx. 3 days which is a indicative of some mid-weekly seasonality. There are also some peaks at the start of the plot.

Plotting the autocorrelation plot:

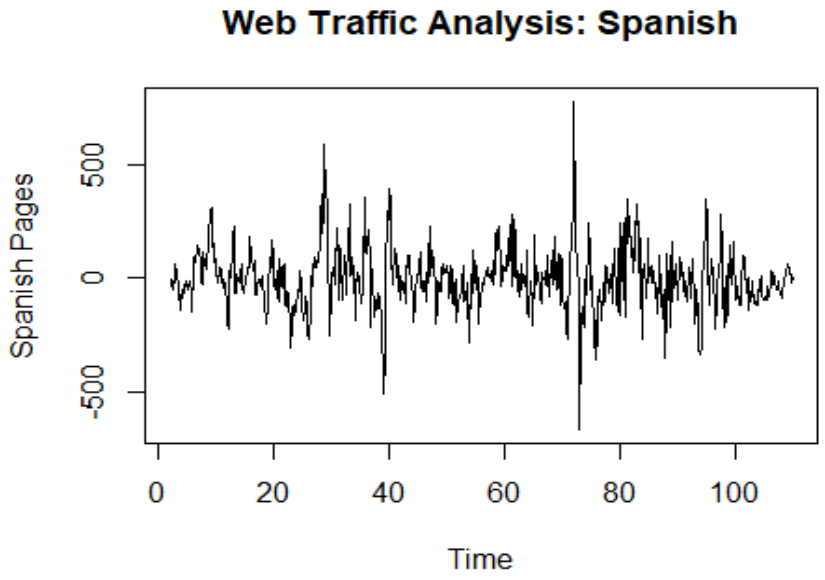
```
acf2(wtd_es_train, main = "Web Traffic Analysis: Spanish")
```



The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

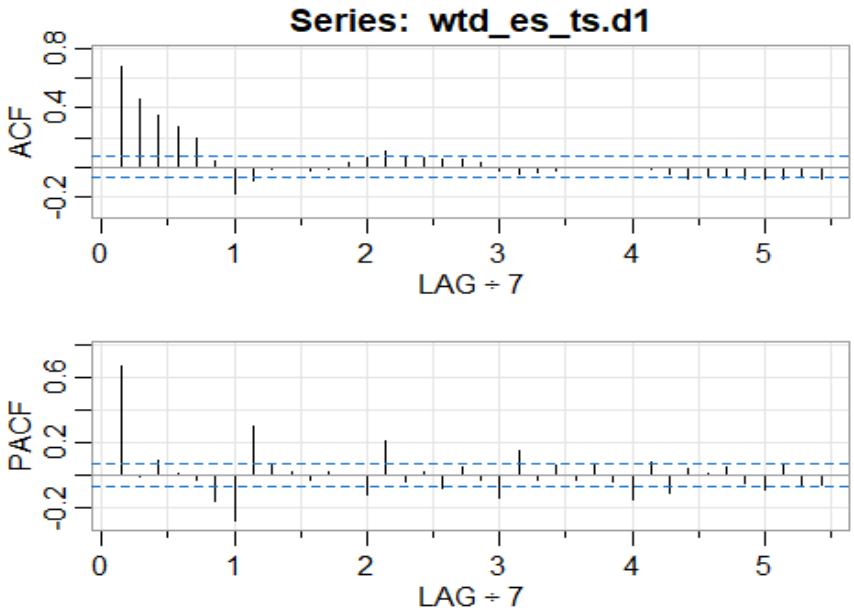
Seasonal Differencing:

```
wtd_es_ts.d1 <- diff(wtd_es_train, lag = 7)
plot(wtd_es_ts.d1,
     main = "Web Traffic Analysis: Spanish",
     ylab = "Spanish Pages", type = 'l')
```




```
kpss.test(wtd_es_ts.d1)

## Warning in kpss.test(wtd_es_ts.d1): p-value greater than printed p-value
```

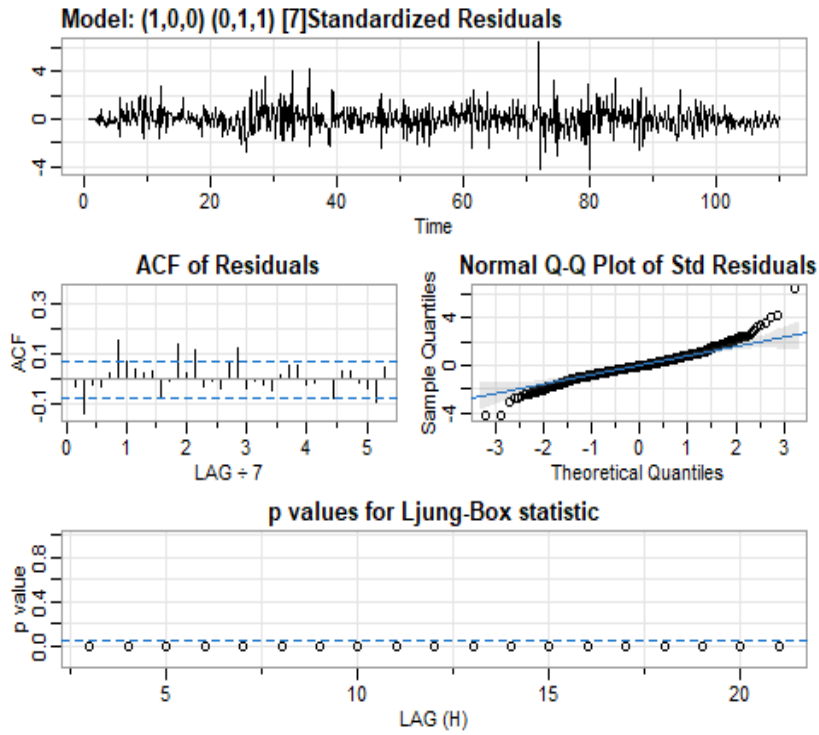
```
##
##  KPSS Test for Level Stationarity
##
## data:  wtd_es_ts.d1
## KPSS Level = 0.13647, Truncation lag parameter = 6, p-value = 0.1

acf2(wtd_es_ts.d1)
```



From the plot above, intuitively I would pick the following values: D = 1 P = 0 Q = 1 d = 0 p = 1 q = 0 I would apply ARIMA(1,0,0)(0,1,1)[7] an run auto ARIMA to find a fit for the model.

Arima Modeling:

```
wtd_es_sm1 <- sarima(wtd_es_train,S = 7,
                     p = 1, d = 0, q = 0,
                     P = 0, D = 1, Q = 1)
```



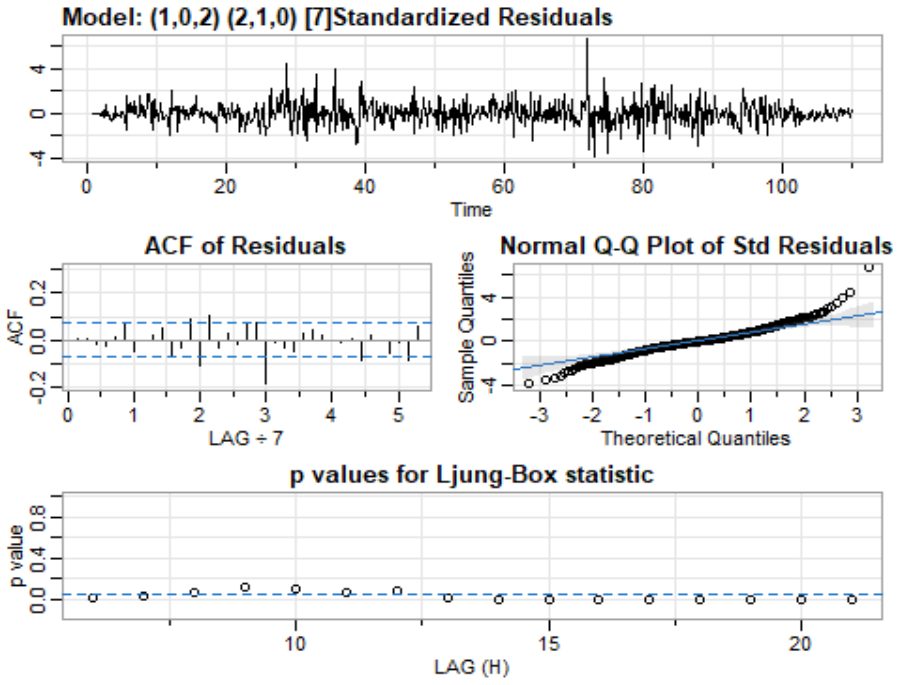

```
wtd_es_sm1

## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     sma1  constant
##       0.8644  -0.7509   -0.2132
## s.e.  0.0220   0.0463    0.7860
##
## sigma^2 estimated as 6523:  log likelihood = -4401.82,  aic = 8811.65
##
## $degrees_of_freedom
## [1] 754
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.8644 0.0220  39.2401  0.0000
## sma1      -0.7509 0.0463 -16.2080  0.0000
## constant  -0.2132 0.7860  -0.2712  0.7863
##
## $AIC
```

```
## [1] 11.64022
##
## $AICc
## [1] 11.64026
##
## $BIC
## [1] 11.66468
```

```
auto.arima(wtd_es_train, seasonal = TRUE)
```

```
## Series: wtd_es_train
## ARIMA(1,0,2)(2,1,0)[7]
##
## Coefficients:
##          ar1      ma1      ma2     sar1    sar2
##       0.9043  -0.1728  -0.1844  -0.5582  -0.236
## s.e.  0.0233   0.0443   0.0421   0.0364   0.036
##
## sigma^2 = 6880:  log likelihood = -4417.63
## AIC=8847.27   AICc=8847.38   BIC=8875.04
```

```
wtd_es_sm2 <- sarima(wtd_es_train,S = 7,
                     p = 1, d = 0, q = 2,
                     P = 2, D = 1, Q = 0)
```




```
wtd_es_sm2
```

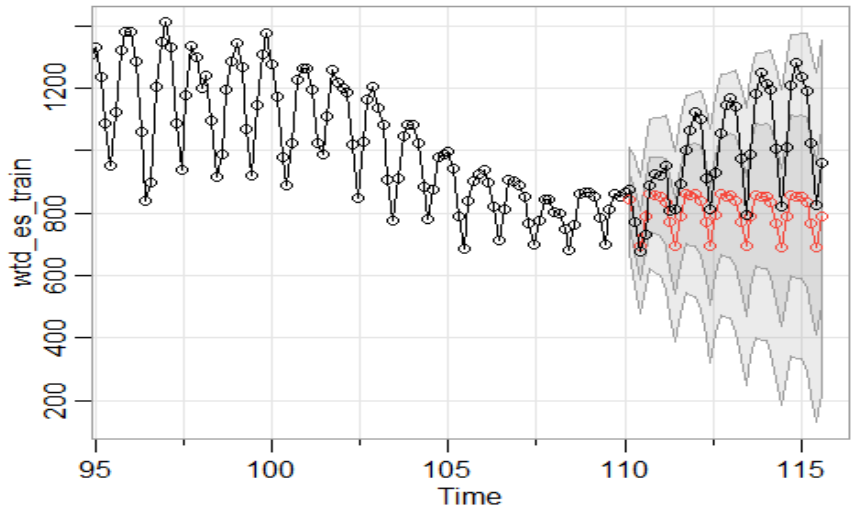
```
## $fit
##
## Call:
```

```
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1      ma2     sar1    sar2  constant
##       0.9040  -0.1724  -0.1843  -0.5581  -0.236   -0.2493
## s.e.  0.0233   0.0443   0.0421   0.0364   0.036    1.5935
##
## sigma^2 estimated as 6834:  log likelihood = -4417.62,  aic = 8849.24
##
## $degrees_of_freedom
## [1] 751
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.9040 0.0233  38.7649  0.0000
## ma1       -0.1724 0.0443  -3.8939  0.0001
## ma2       -0.1843 0.0421  -4.3788  0.0000
## sar1      -0.5581 0.0364 -15.3299  0.0000
## sar2      -0.2360 0.0360  -6.5553  0.0000
## constant  -0.2493 1.5935  -0.1564  0.8757
##
## $AIC
## [1] 11.68988
##
## $AICc
## [1] 11.69003
##
## $BIC
## [1] 11.73269
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : **ARIMA(1,0,2)(2,1,0)[7]** for forecasting because among all the models the ACF of Residuals and p-values for Ljung-Box statistic looks better for this model and there is not much relative difference in the AIC value between the models.

Forecasting:

```
wtd_es_sm1_for <- sarima.for(wtd_es_train,n.ahead = 39,S = 7,
                      p = 1, d = 0, q = 2,
                      P = 2, D = 1, Q = 0)
lines(wtd_es_test, type = 'o')
```



Evaluating Accuracy:

```
accuracy(wtd_es_sm1_for$pred,x = wtd_es_test)

##                  ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 191.5251 229.2526 195.7292 17.72636 18.31138 0.8143218  1.664813
```

The RMSE value is **229.2526**.

OBSERVATIONS AND RESULTS

Non-Normality of the Residuals

A normal Q-Q Plot of standardized residuals is used to assess if the residuals satisfy the assumption of normality. If all the points fall on the straight line, then the data is said to be normally distributed. In this project, all the observed Q-Q plots have residuals that deviate at the far ends of the line, but fall on the line at that center. This is indicative of "Tailedness". This means the distribution has fat tails or data that is distributed farther away from the mean of the data. This in turn means the time series still have a lot of noise element in them and since real world data is not perfect, it rarely follows a perfect normal distribution.

Similarities and Differences in the language models

We can see from the EDA that the English Wikipedia pages have the most traffic (or number of views) and Chinese pages have the least. The other five languages have approx. the same number of views with Russian and Spanish languages being on the higher end. There is also a similarity in the peaking of viewership in Russian and English pages. However, whether there is any relation between the two models has not been explored in the current project and is a can be considered a future work.

All the language models are non-stationary. Each of them has some amount of trend, weekly and quarterly seasonality as well as lot of noise. The Chinese, Japanese, French and German pages have similar spectral density plots with a main peak at weekly seasonality and some

smaller peaks around quarterly seasonality. The English model has a slightly different spectral density with more peaks at the start of the plot demonstrating a higher quarterly seasonality. The spectral density plot for the Spanish model shows the highest weekly seasonality and the one for Russian model shows higher quarterly seasonality and not much of a weekly seasonality.

The same thing is represented in ACF plots as the Chinese, French, German, Japanese and English models have similar ACF Plots but Spanish and Russian Models have a different ACF plot kind of representing their differing seasonalities. All the models had to be differenced to make them stationary. The French and the Russian models were differenced at first-order as the model was noisy and the first-order differencing resulted in better accuracy. All the other models were seasonally differenced.

The German, French, Chinese and Japanese have relatively low RMSE, which makes sense as the number of views for these pages as mentioned earlier is lesser than other pages. The Russian model has a slightly higher RMSE but it also has viewers on the higher end and more seasonality so this was also expected. The English model has a very high RMSE and also most number of viewers so this also did not seem out of the place. However, the Spanish model forecast had a high RMSE and relatively lesser number of viewers. This could attribute to the high weekly seasonality and noise in the data. It can also be seen from the Q-Q plots of the residuals that none of the models follow a strict normal distribution. They have data which are much farther away from the mean of the data which could be due to sudden spikes in the viewership or the noise element.

CONCLUSIONS

The Wikipedia Web Traffic time series data was successfully analyzed and forecasted by grouping together by language. Each time series was individually decomposed, non-stationarity was differenced out and then the most appropriate ARIMA model was identified using AIC metrics and the residual plots. The time series was forecasted for all 7 languages however, the accuracy is not great for all of them. There is definitely a scope of improvement where in different Machine learning models can be applied to forecast future data and a comparison can be done with the results of the ARIMA model. In some models non-significant terms were unavoidable and but I have tried to reduce the number of non-significant terms in the models as much as I could.

REFERENCES

1. N. Petluri and E. Al-Masri, “Web Traffic Prediction of Wikipedia Pages,” 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5427-5429, doi: 10.1109/BigData.2018.8622207.
2. Kämpf M, Tessenow E, Kenett DY, Kantelhardt JW. The Detection of Emerging Trends Using Wikipedia Traffic Data and Context Networks. PLoS One. 2015;10(12):e0141892. Published 2015 Dec 31. doi:10.1371/journal.pone.0141892

3. http://manishbarnwal.com/blog/2017/05/03/time_series_and_forecasting_using_R/#:~:text=ts()%20function%20is%20used,set%20frequency%20of%20the%20data
4. https://towardsdatascience.com/stl-decomposition-how-to-do-it-from-scratch-b686711986ec
5. https://online.stat.psu.edu/stat510/lesson/4/4.2
6. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
7. https://www.wikipedia.org/