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 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 th 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.

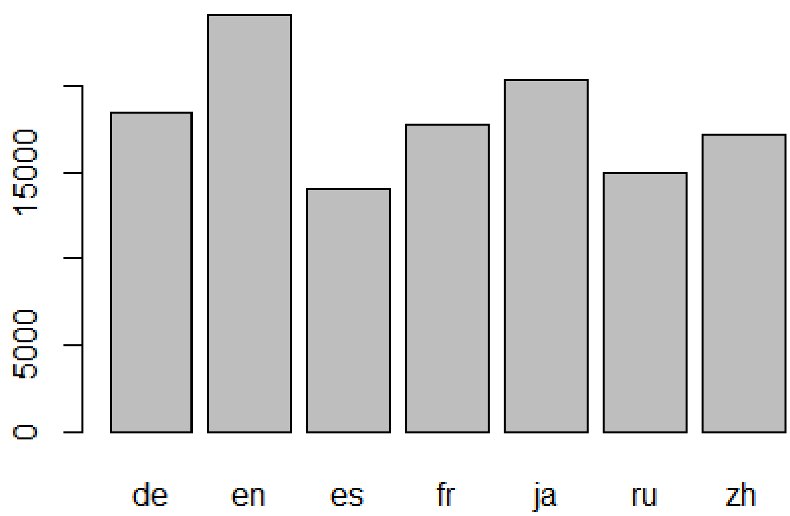
```
getLang <- function(page){
  res <- stri_extract(str = page, regex = '[a-z][a-z].wikipedia.org')
  if(!is.na(res))
    return(substr(res,0,2))
  return('na')
}
```

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

The plot shows the counts of each of the languages in the data set.

```
langCnt <- table(getLang(wtd$Page))

barplot(langCnt)
```



Next I have written a function : grpByLang that groups the data set based on the language of the page and stores the data into seperate 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 os then transposed so that the dates act as rows and number of visits becomee the column. Finally, it is converted into a time series object with a frequency of 7 as it is a daily data set.

```
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
```

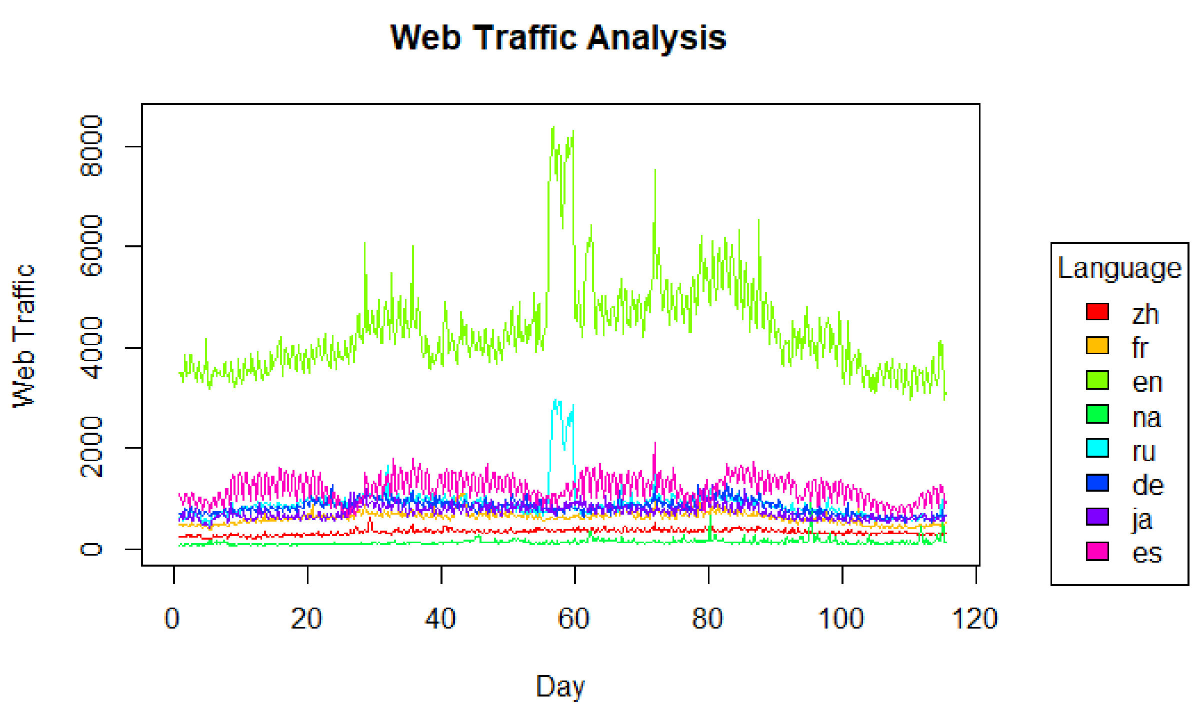
```r
langCodes <- unique(wtd$lang)
wtd_lang <- data.frame()

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

library(lubridate)

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))
```

```r
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 the web traffic of each language in a different colours. 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.

```r
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 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.

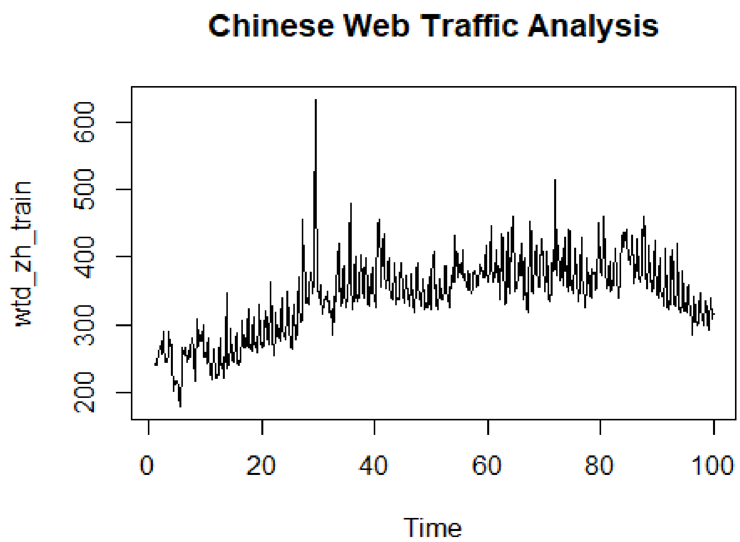
I have briefly described the results of each step and my decision process behind selecting a particular model.

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:

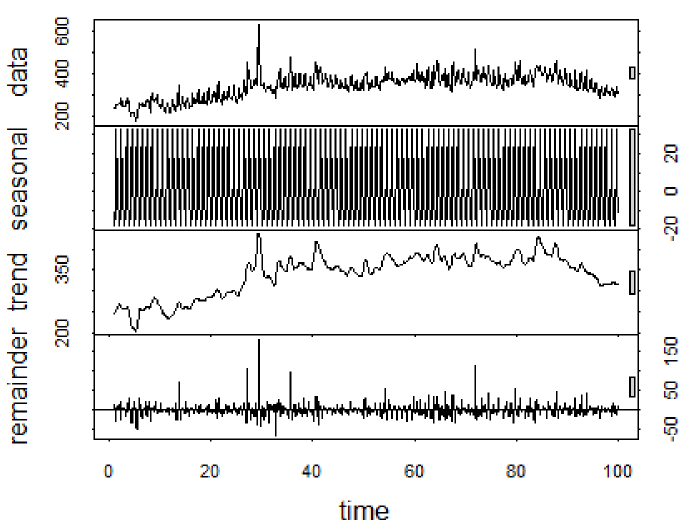
```
wtd_zh_train <- window(wtd_zh_ts, end = 100)
wtd_zh_test <- window(wtd_zh_ts, start = 100)
plot(wtd_zh_train, main = "Chinese Web Traffic Analysis")
```



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:

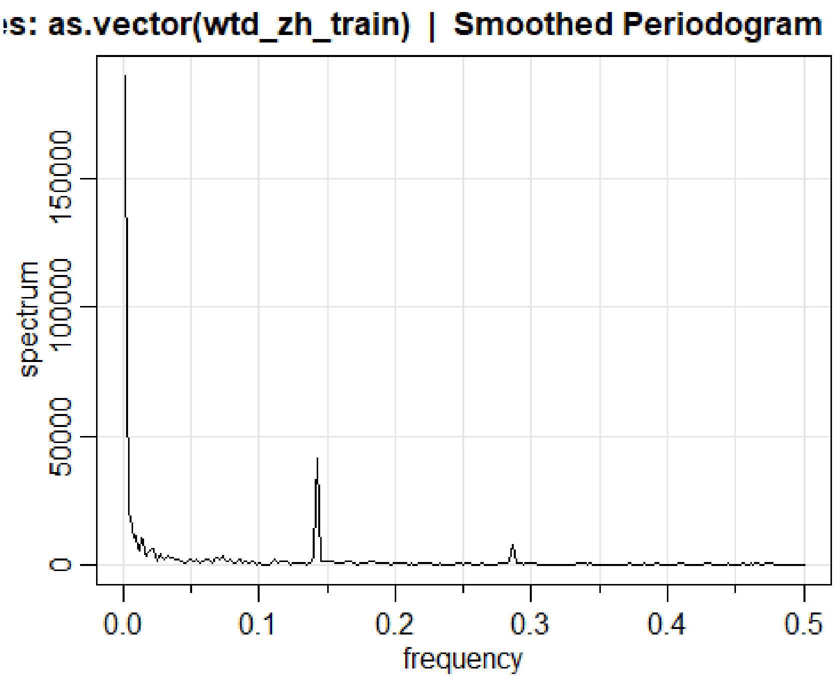
```
kpss.test(wtd_zh_train)

## 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 = 5.6342, 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:

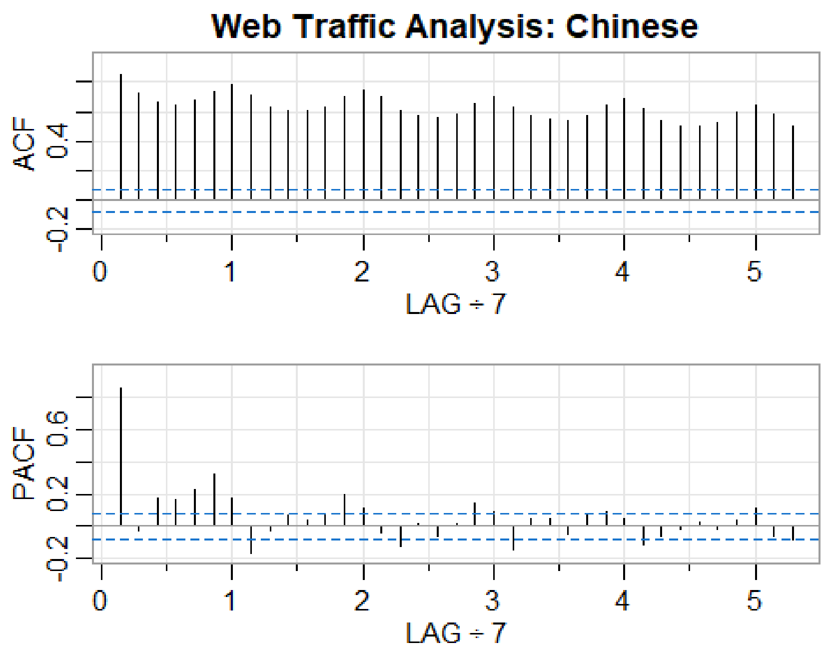
```
wtd_zh.spec <- mvspec(as.vector(wtd_zh_train),detrend = TRUE, spans = 3)
```



The plot shows one major peak around the 140th day (approx.) and a small peak arounf the 280th . Representative of quarterly seasonality?

Plotting the Autocorrelation plot:

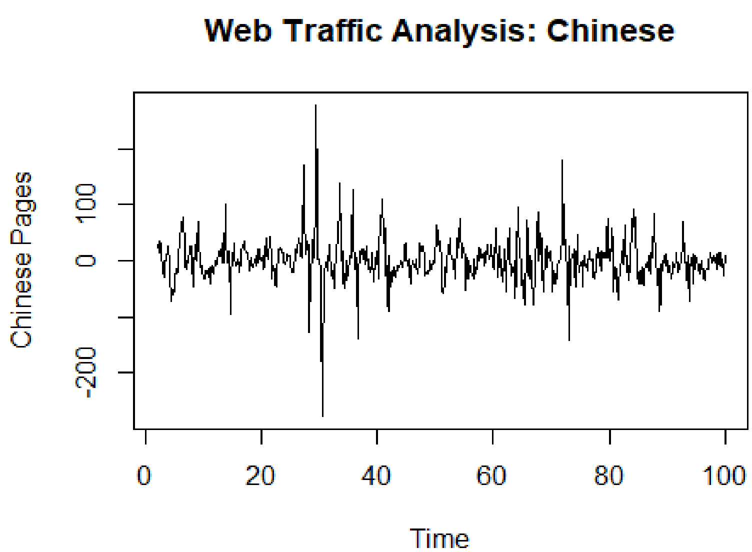
```
acf2(wtd_zh_train, main = "Web Traffic Analysis: Chinese")
```



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

Performing Seasonal Differencing:

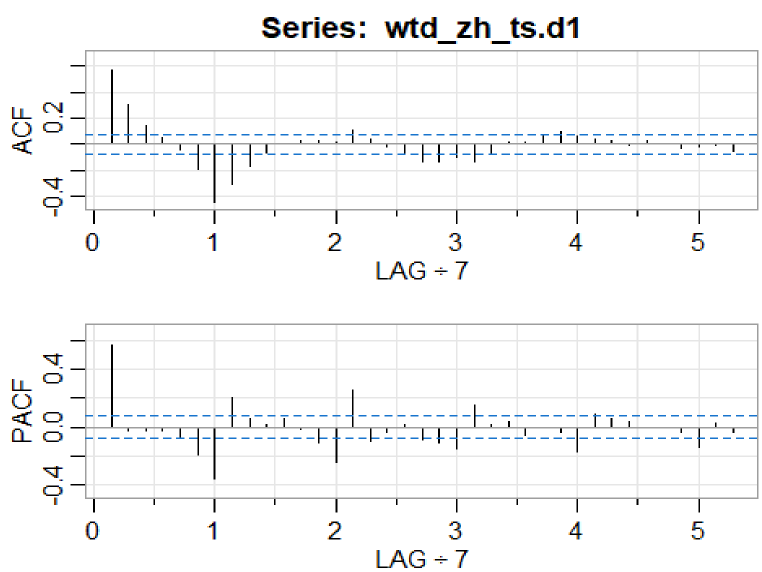
```
wtd_zh_ts.d1 <- diff(wtd_zh_train, lag = 7)
plot(wtd_zh_ts.d1,
     main = "Web Traffic Analysis: Chinese",
     ylab = "Chinese Pages", type = 'l')
```




```
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 = 0.10539, Truncation lag parameter = 6, p-value = 0.1
```

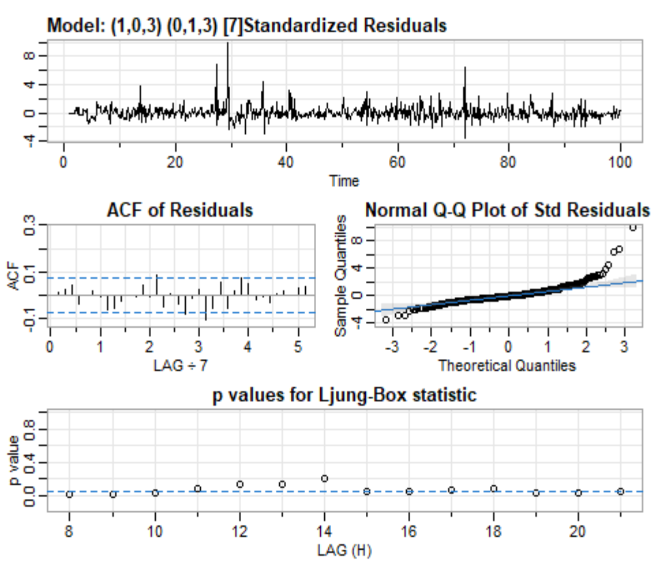
```
acf2(wtd_zh_ts.d1)
```



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

ARIMA Modeling:

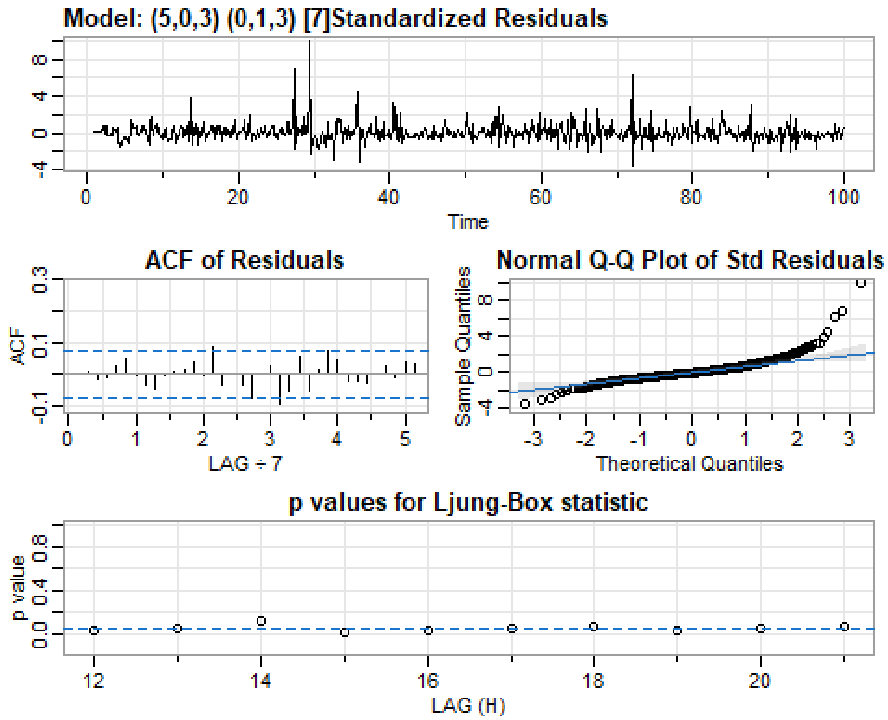
```
wtd_zh_sm1 <- sarima(wtd_zh_train, S = 7,
                     p = 1, d = 0, q = 3,
                     P = 0, D = 1, Q = 3)
```



```r
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     sma2    sma3  constan
t
##       0.9935  -0.3518  -0.2507  -0.1693  -0.9553  -0.0619  0.0319    0.120
2
## s.e.  0.0092   0.0386   0.0437   0.0392   0.0481   0.0542  0.0392    0.110
2
##
## sigma^2 estimated as 510:  log likelihood = -3127.24,  aic = 6272.49
##
## $degrees_of_freedom
## [1] 679
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.9935 0.0092 107.9705  0.0000
## ma1       -0.3518 0.0386  -9.1066  0.0000
## ma2       -0.2507 0.0437  -5.7342  0.0000
## ma3       -0.1693 0.0392  -4.3148  0.0000
## sma1      -0.9553 0.0481 -19.8730  0.0000
## sma2      -0.0619 0.0542  -1.1423  0.2537
## sma3       0.0319 0.0392   0.8128  0.4166
## constant   0.1202 0.1102   1.0911  0.2756
##
## $AIC
## [1] 9.130256
##
## $AICc
## [1] 9.130565
##
## $BIC
## [1] 9.189632
```

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



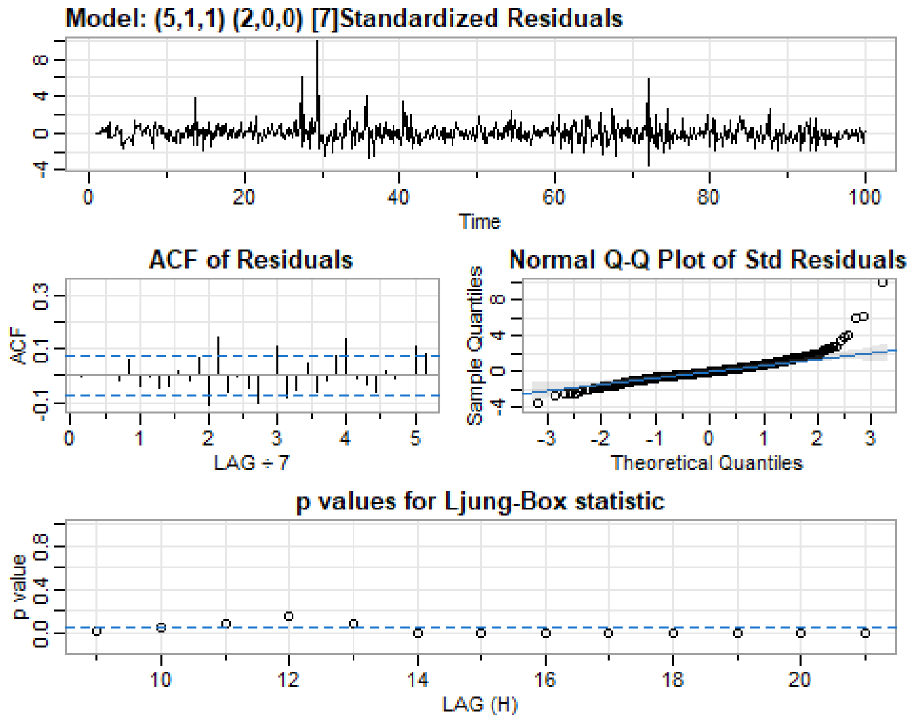

```
wtd_zh_sm2

## $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
##        -0.3982  1.4918  0.3854  -0.5294  0.0439  1.0596  -0.8227  -0.9204
## s.e.    0.0424  0.0464  0.0669   0.0463  0.0412  0.0151   0.0360   0.0196
##           sma1     sma2    sma3  constant
##        -0.9317  -0.0720  0.0197    0.1105
## s.e.    0.0573   0.0539  0.0449    0.1246
##
## sigma^2 estimated as 501.4:  log likelihood = -3122.54,  aic = 6271.08
##
## $degrees_of_freedom
## [1] 675
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1       -0.3982 0.0424  -9.3975  0.0000
## ar2        1.4918 0.0464  32.1179  0.0000
```

```
## ar3        0.3854 0.0669   5.7578  0.0000
## ar4       -0.5294 0.0463 -11.4366  0.0000
## ar5        0.0439 0.0412   1.0660  0.2868
## ma1        1.0596 0.0151  69.9905  0.0000
## ma2       -0.8227 0.0360 -22.8649  0.0000
## ma3       -0.9204 0.0196 -46.9554  0.0000
## sma1      -0.9317 0.0573 -16.2720  0.0000
## sma2      -0.0720 0.0539  -1.3370  0.1817
## sma3       0.0197 0.0449   0.4395  0.6605
## constant   0.1105 0.1246   0.8871  0.3753
##
## $AIC
## [1] 9.128204
##
## $AICc
## [1] 9.128878
##
## $BIC
## [1] 9.213969
```

```
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
##       0.6055  -0.0468  -0.0282  0.0104  0.0475  -0.9684  0.2873  0.2041
## s.e.  0.0397   0.0464   0.0445  0.0449  0.0397   0.0106  0.0390  0.0399
##
## sigma^2 = 615.9:  log likelihood = -3206.06
## AIC=6430.13   AICc=6430.39   BIC=6470.99
```

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



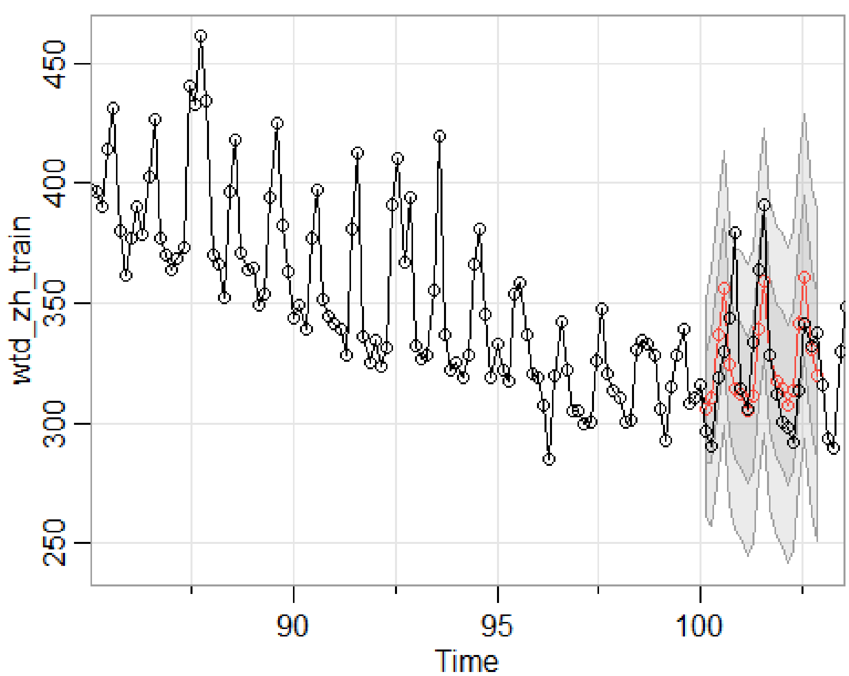

```
wtd_zh_sm3

## $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
##       0.6067  -0.0462  -0.0278  0.0110  0.0484  -0.9708  0.2877  0.2044
## s.e.  0.0397   0.0464   0.0446  0.0449  0.0397   0.0107  0.0390  0.0399
##       constant
##         0.1191
## s.e.    0.1370
##
## sigma^2 estimated as 608.2:  log likelihood = -3205.71,  aic = 6431.43
##
## $degrees_of_freedom
## [1] 684
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.6067 0.0397  15.2770  0.0000
## ar2       -0.0462 0.0464  -0.9942  0.3205
```

```
## ar3        -0.0278 0.0446  -0.6228  0.5336
## ar4         0.0110 0.0449   0.2437  0.8075
## ar5         0.0484 0.0397   1.2192  0.2232
## ma1        -0.9708 0.0107 -90.3720  0.0000
## sar1        0.2877 0.0390   7.3711  0.0000
## sar2        0.2044 0.0399   5.1196  0.0000
## constant    0.1191 0.1370   0.8695  0.3849
##
## $AIC
## [1] 9.28056
##
## $AICc
## [1] 9.28094
##
## $BIC
## [1] 9.346087
```

Looking at the above plots, ARIMA(5,0,3)(0,1,3)[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)(0,1,3)[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 = 20,S = 7,
                             p = 1, d = 0, q = 3,
                             P = 0, D = 1, Q = 3)
lines(wtd_zh_test, type = 'o')
```



Estimating the accuracy:

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

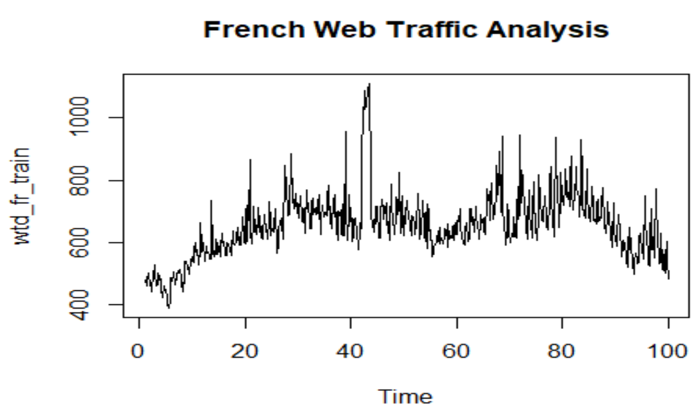
```
##                 ME     RMSE      MAE        MPE     MAPE      ACF1 Theil's
U
## Test set 0.7262922 23.00949 17.93274 -0.2014689 5.337733 0.4559495  0.9029
54
```

The RMSE value is 23.00949.

French Web Traffic

Splitting the data set into train and test sets:

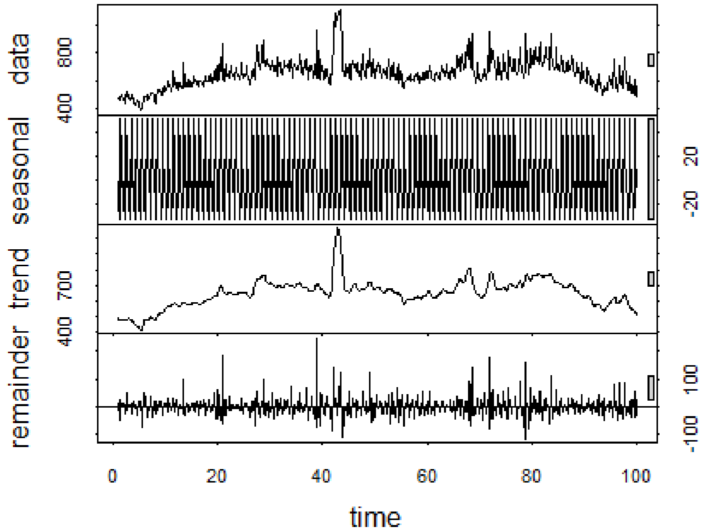
```
wtd_fr_train <- window(wtd_fr_ts, end = 100)
wtd_fr_test <- window(wtd_fr_ts, start = 100)
plot(wtd_fr_train, main = "French Web Traffic Analysis")
```



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)
```



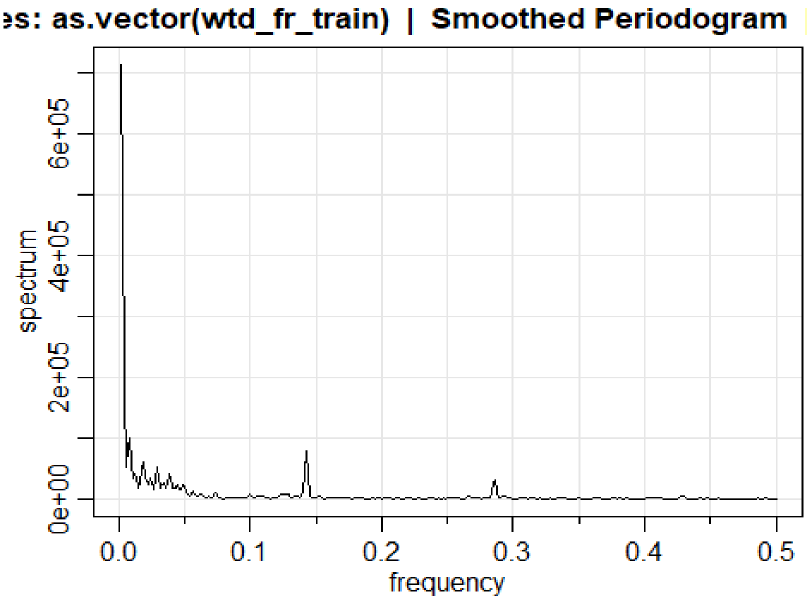
Performing the KPSS Test for stationarity:

```
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 = 1.9486, 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:

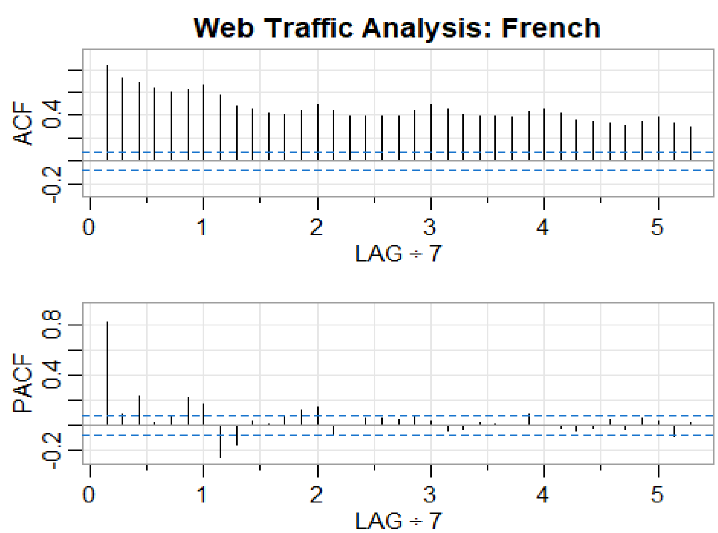
```
wtd_fr.spec <- mvspec(as.vector(wtd_fr_train),detrend = TRUE, spans = 2)
```



There is a weekly seasonality that can be seen in the spectral analysis. However, there seem to be two other smaller spikes around the 140th and the 280th day (approx.) but not any other significant peaks. This may signify some kind of quarterly seasonality.

Plotting the Autocorrelation plot:

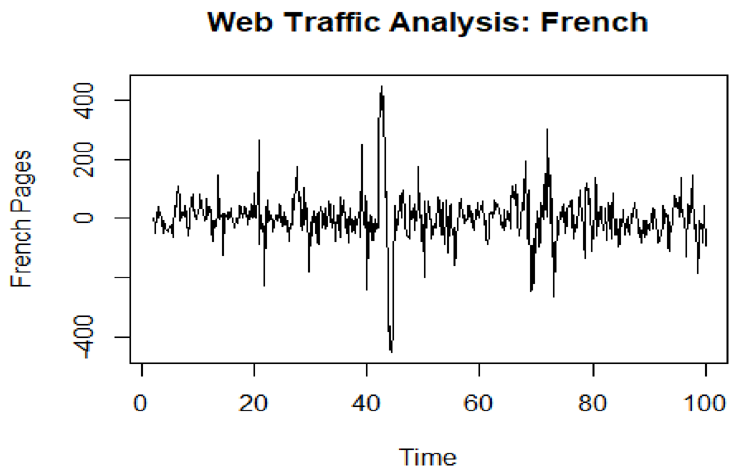
```
acf2(wtd_fr_train, main = "Web Traffic Analysis: French")
```



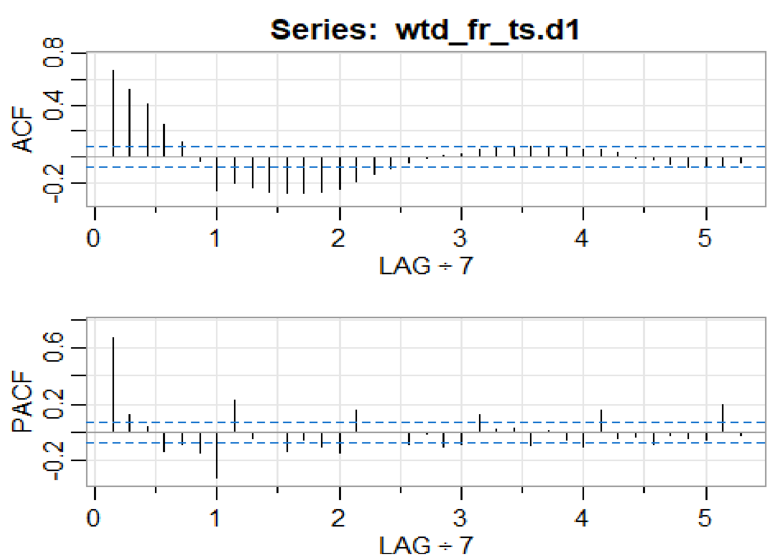
The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

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')
```



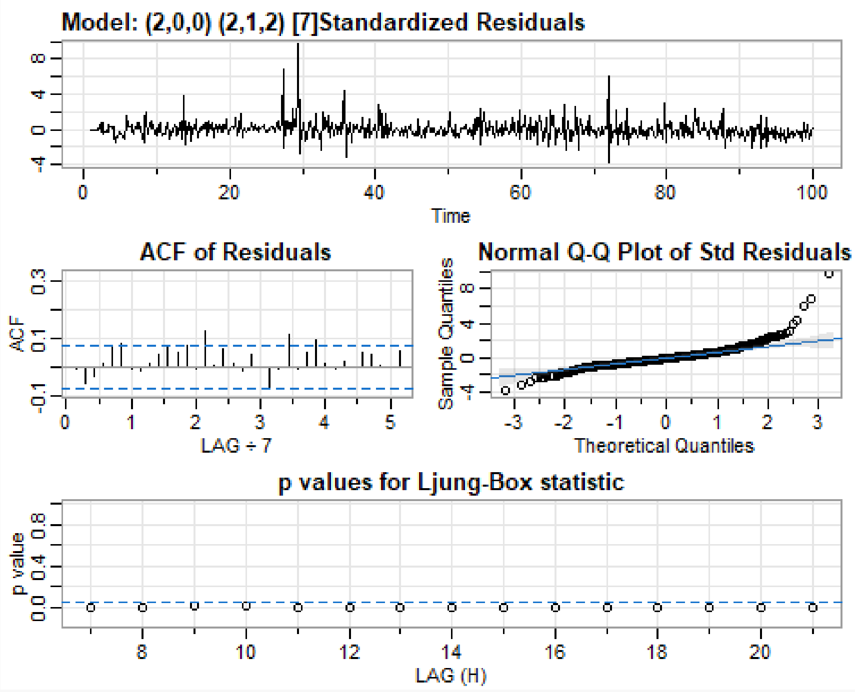

```
kpss.test(wtd_fr_ts.d1)

## 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 = 0.084302, 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.

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




```r
wtd_fr_sm1
```

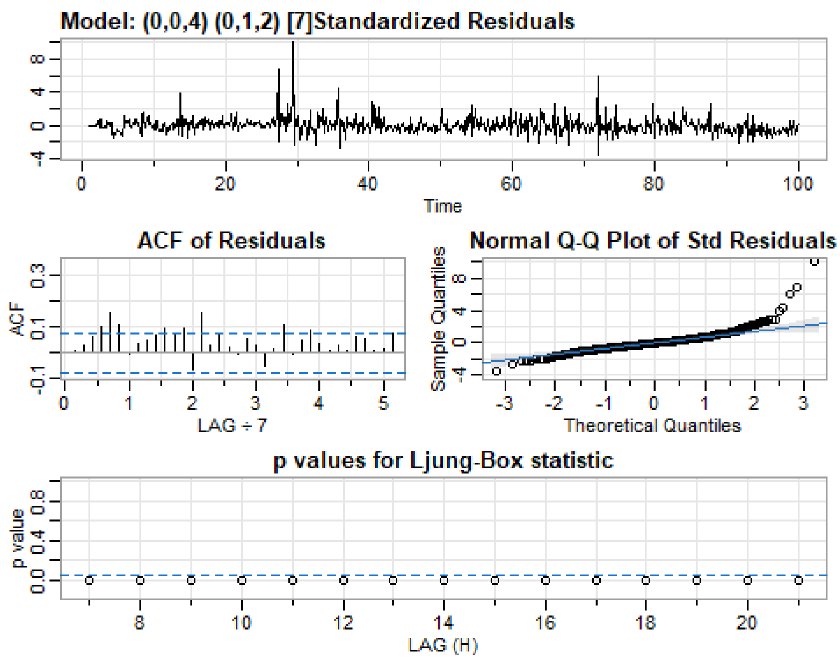
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2     sar1     sar2     sma1     sma2  constant
##       0.6673  0.0562  -0.1158  -0.0774  -0.7194  -0.1019    0.1376
## s.e.  0.0400  0.0393   0.4447   0.0497   0.4437   0.3898    0.0730
##
## sigma^2 estimated as 541.9:  log likelihood = -3142.06,  aic = 6300.11
##
## $degrees_of_freedom
## [1] 680
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.6673 0.0400 16.6728  0.0000
## ar2        0.0562 0.0393  1.4288  0.1535
## sar1      -0.1158 0.4447 -0.2604  0.7946
```

```
## sar2      -0.0774 0.0497 -1.5585  0.1196
## sma1      -0.7194 0.4437 -1.6213  0.1054
## sma2      -0.1019 0.3898 -0.2615  0.7938
## constant   0.1376 0.0730  1.8838  0.0600
##
## $AIC
## [1] 9.170468
##
## $AICc
## [1] 9.170708
##
## $BIC
## [1] 9.223246
```

```r
auto.arima(wtd_fr_train, D = 1, seasonal = TRUE)
```

```
## Series: wtd_fr_train
## ARIMA(0,0,4)(0,1,2)[7]
##
## Coefficients:
##          ma1     ma2     ma3     ma4     sma1     sma2
##       0.6615  0.4392  0.4166  0.2325  -0.7040  -0.1150
## s.e.  0.0401  0.0463  0.0376  0.0351   0.0411   0.0395
##
## sigma^2 = 2466:  log likelihood = -3658.56
## AIC=7331.11   AICc=7331.28   BIC=7362.84
```

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



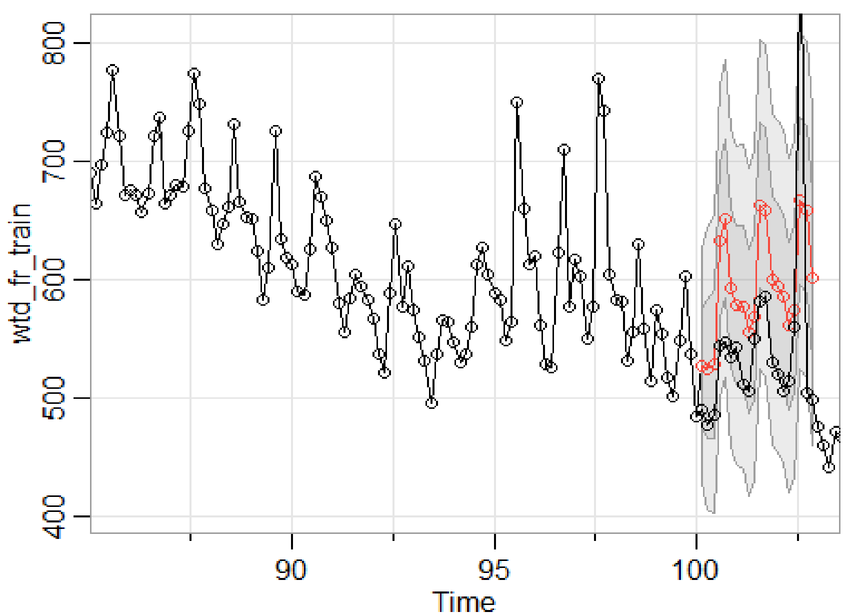

```
wtd_fr_sm2

## $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:
##          ma1     ma2     ma3     ma4     sma1     sma2  constant
##       0.6593  0.4222  0.2290  0.0909  -0.7780  -0.0331    0.1323
## s.e.  0.0398  0.0462  0.0403  0.0362   0.0423   0.0420    0.0619
##
## sigma^2 estimated as 565.6:  log likelihood = -3155.8,  aic = 6327.59
##
## $degrees_of_freedom
## [1] 680
##
## $ttable
##          Estimate     SE  t.value p.value
## ma1        0.6593 0.0398  16.5500  0.0000
## ma2        0.4222 0.0462   9.1449  0.0000
## ma3        0.2290 0.0403   5.6843  0.0000
## ma4        0.0909 0.0362   2.5071  0.0124
## sma1      -0.7780 0.0423 -18.4122  0.0000
## sma2      -0.0331 0.0420  -0.7877  0.4311
## constant   0.1323 0.0619   2.1372  0.0329
```

```
##
## $AIC
## [1] 9.210471
##
## $AICc
## [1] 9.210711
##
## $BIC
## [1] 9.263249
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : ARIMA(0,0,4)(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 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 = 20,S = 7,
                             p = 0, d = 0, q = 4,
                             P = 0, D = 1, Q = 2)
lines(wtd_fr_test, type = 'o')
```



Evaluating the accuracy:

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

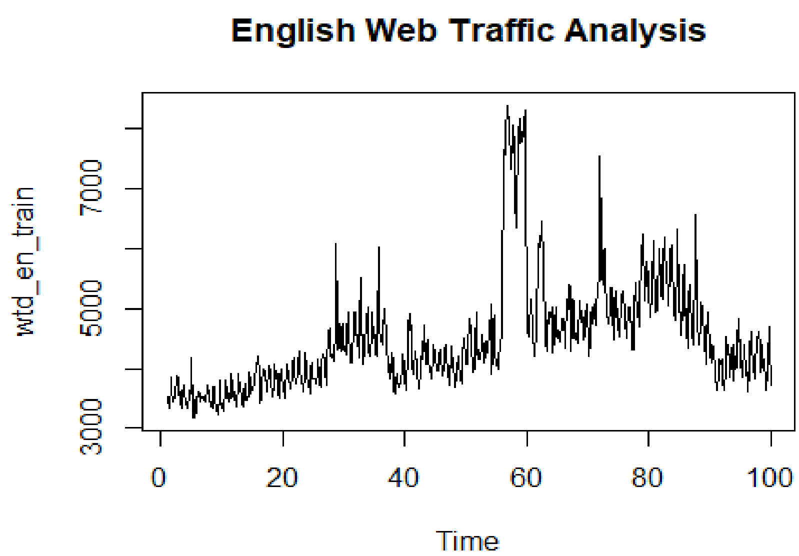
```
##                  ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's
U
## Test set -50.85587 88.62662 74.23264 -10.65307 13.24803 -0.1031373 0.88485
24
```

The RMSE of the model is pretty high : 88.62662.

English Web Traffic

Splitting the data set into train and test sets:

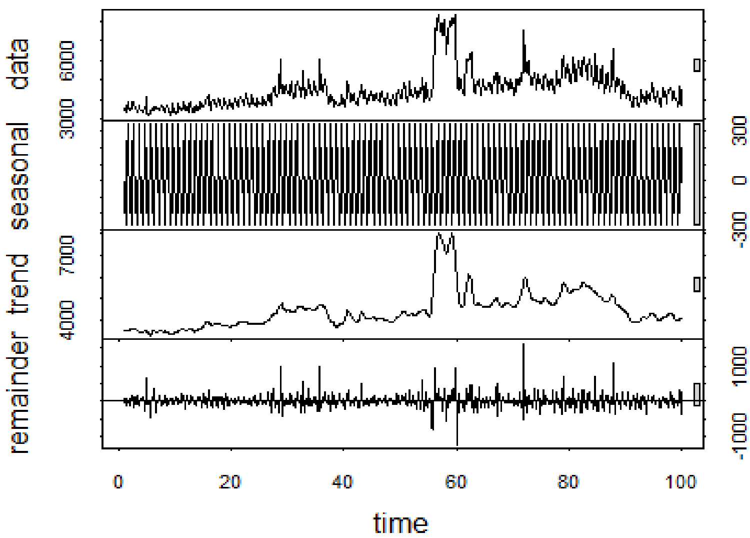
```
wtd_en_train <- window(wtd_en_ts, end = 100)
wtd_en_test <- window(wtd_en_ts, start = 100)
plot(wtd_en_train, main = "English Web Traffic Analysis")
```



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)
```



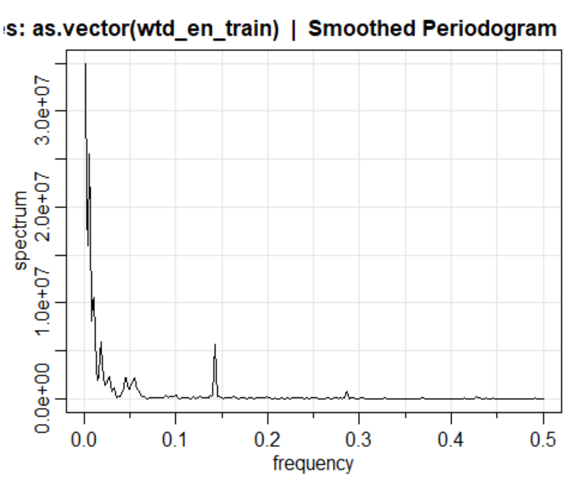
Performing the KPSS Test for stationarity:

```
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 = 3.2137, 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:

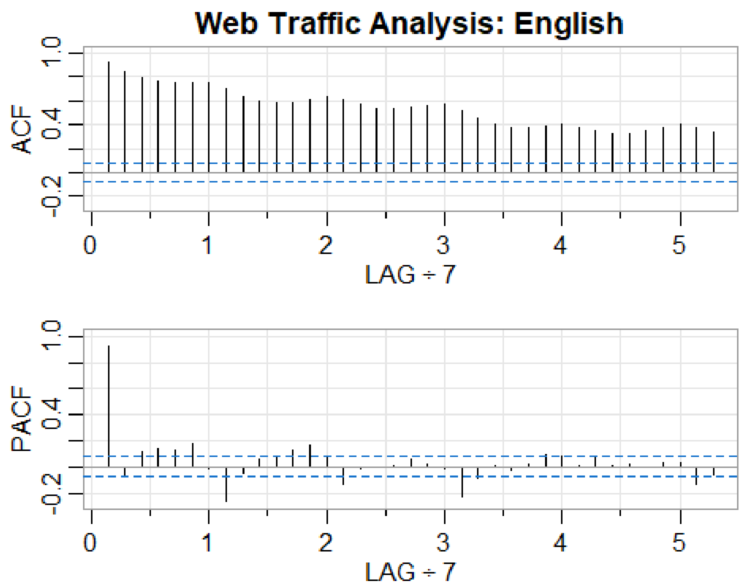
```
wtd_en.spec <- mvspec(as.vector(wtd_en_train),detrend = TRUE, spans = 3)
```



There is a weekly seasonality that can be seen in the spectral analysis. There are also multiple peaks in the beginning of the plot followed by significant spikes around the 140th day.

Plotting the Autocorrelation plot:

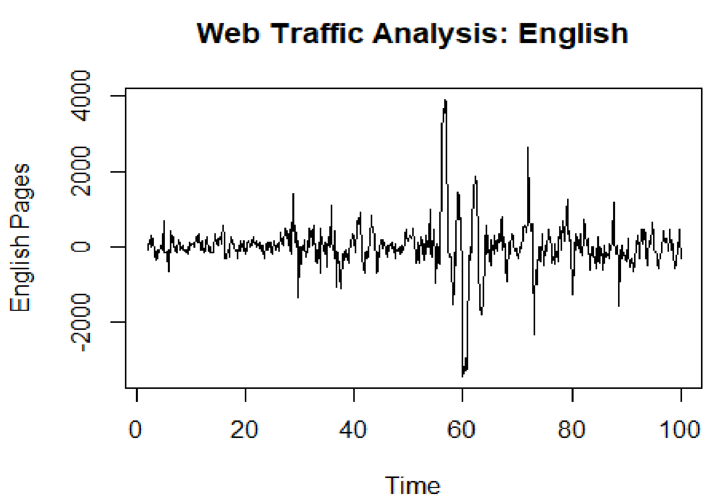
```
acf2(wtd_en_train, main = "Web Traffic Analysis: English")
```



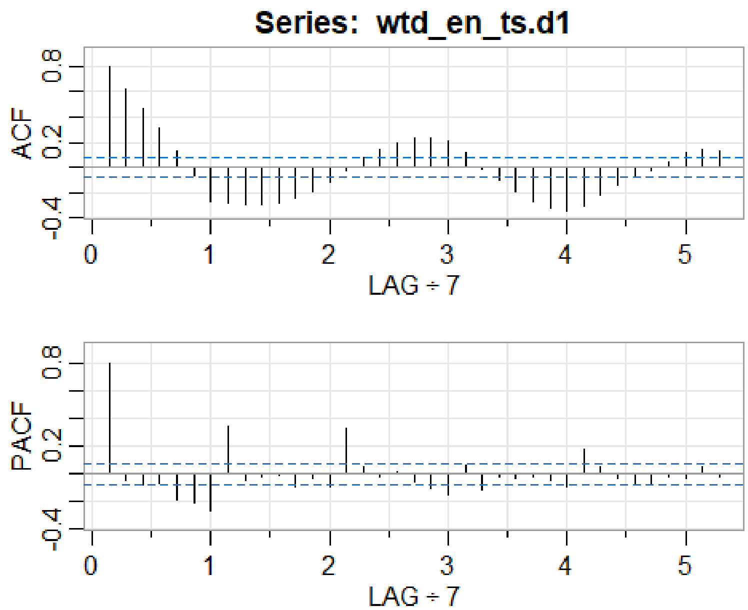
The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

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')
```



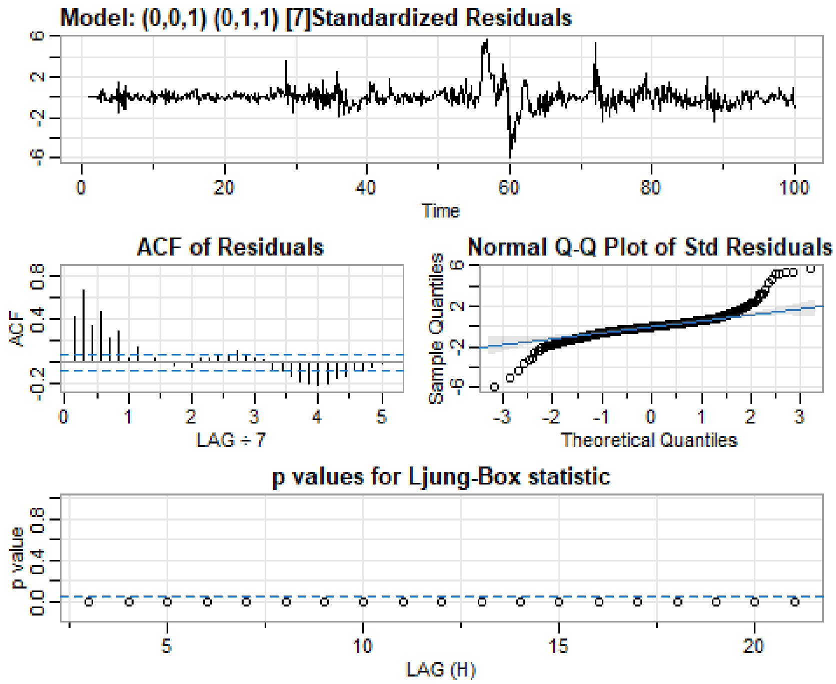
```
kpss.test(wtd_en_ts.d1)

## 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 = 0.045783, 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)
```



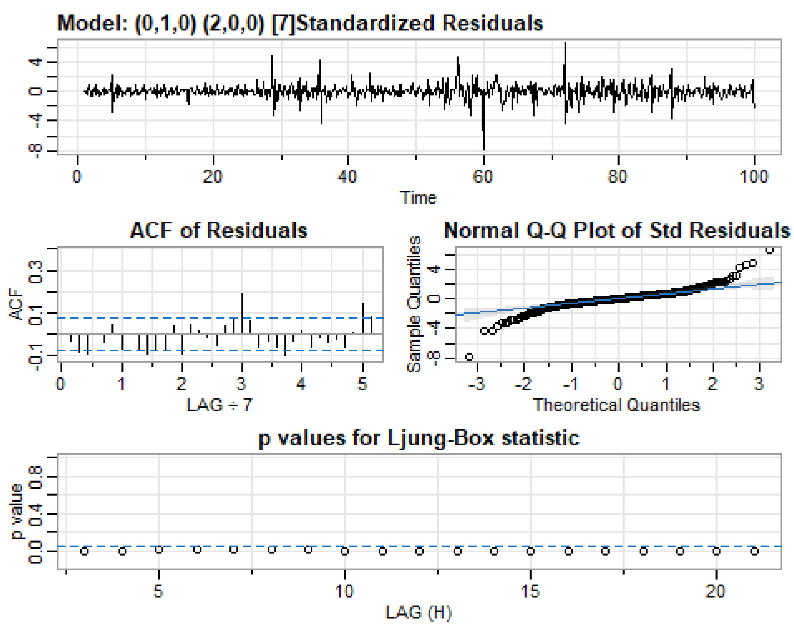

```
wtd_en_sm1

## $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
##       0.6621  -0.4943    0.7967
## s.e.  0.0211   0.0410    2.0186
##
## sigma^2 estimated as 190489:  log likelihood = -5152.1,  aic = 10312.2
##
## $degrees_of_freedom
## [1] 684
##
## $ttable
##          Estimate     SE  t.value p.value
## ma1        0.6621 0.0211  31.3775  0.0000
## sma1      -0.4943 0.0410 -12.0576  0.0000
## constant   0.7967 2.0186   0.3947  0.6932
##
## $AIC
## [1] 15.01049
```

```
##
## $AICc
## [1] 15.01054
##
## $BIC
## [1] 15.03688
```

```
auto.arima(wtd_en_train, D = 1, seasonal = TRUE)
```

```
## Series: wtd_en_train
## ARIMA(1,0,0)(2,1,0)[7]
##
## Coefficients:
##          ar1     sar1     sar2
##       0.8757  -0.6711  -0.3808
## s.e.  0.0187   0.0358   0.0353
##
## sigma^2 = 98394:  log likelihood = -4924.92
## AIC=9857.85   AICc=9857.9   BIC=9875.97
```

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

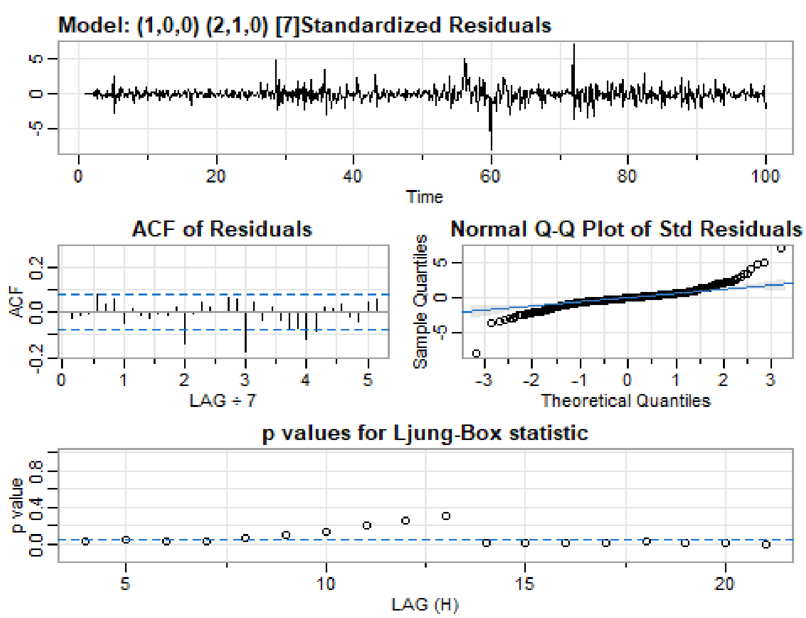



```
wtd_en_sm2
```

```
## $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:
##          sar1    sar2  constant
##        0.2466  0.2395   -0.6157
## s.e.  0.0368  0.0368   23.3603
##
## sigma^2 estimated as 102733:  log likelihood = -4982.7,  aic = 9973.4
##
## $degrees_of_freedom
## [1] 690
##
## $ttable
##           Estimate      SE t.value p.value
## sar1        0.2466  0.0368  6.7001   0.000
## sar2        0.2395  0.0368  6.5009   0.000
## constant   -0.6157 23.3603 -0.0264   0.979
##
## $AIC
## [1] 14.39163
##
## $AICc
## [1] 14.39168
##
## $BIC
## [1] 14.41784
```

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

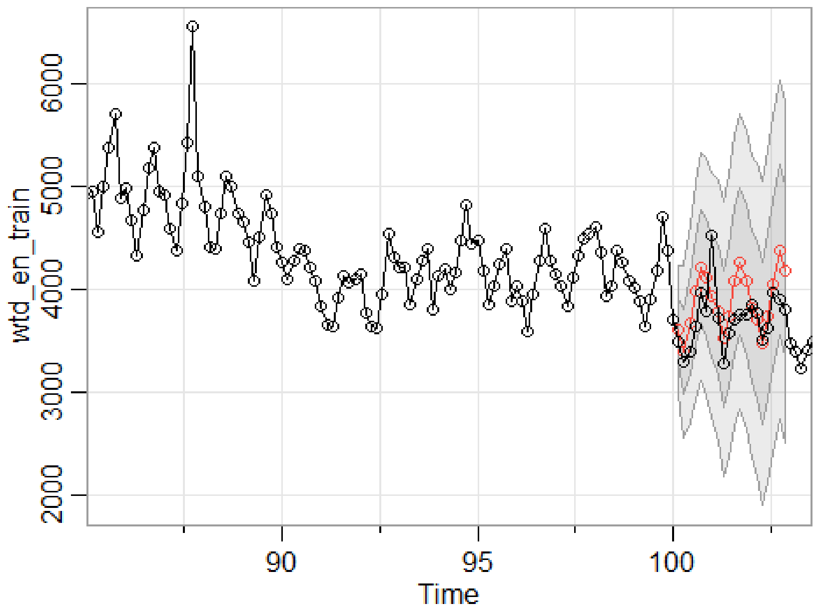


```
wtd_en_sm3

## $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
##       0.8756  -0.6710  -0.3808    0.4100
## s.e.  0.0187   0.0358   0.0353    6.6632
## 
## sigma^2 estimated as 97964:  log likelihood = -4924.92,  aic = 9859.84
## 
## $degrees_of_freedom
## [1] 683
## 
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.8756 0.0187  46.8693   0.000
## sar1      -0.6710 0.0358 -18.7326   0.000
## sar2      -0.3808 0.0353 -10.7930   0.000
## constant   0.4100 6.6632   0.0615   0.951
## 
## $AIC
## [1] 14.35203
## 
## $AICc
## [1] 14.35211
## 
## $BIC
## [1] 14.38501
```

Looking at the above plots, I have decided to go ahead with model generated by auto ARIMA : ARIMA(1,0,0)(2,1,0)[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 models.

Forecasting:

```
wtd_en_sm1_for <- sarima.for(wtd_en_train,n.ahead = 20,S = 7,
                      p = 1, d = 0, q = 0,
                     P = 2, D = 1, Q = 0)
lines(wtd_en_test, type = 'o')
```



Evaluating accuracy:

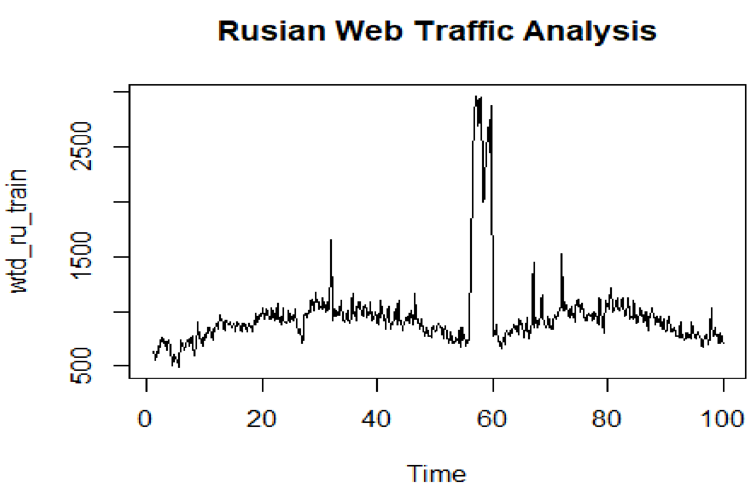
```
accuracy(wtd_en_sm1_for$pred,x = wtd_en_test)

##                    ME     RMSE     MAE       MPE    MAPE      ACF1 Theil's U
## Test set -173.1482 296.1758 243.634 -4.839356 6.44958 0.1968077 0.9826485
```

The RMSE value is 296.1758.

Russian Web Traffic

Splitting the data set into train and test sets:

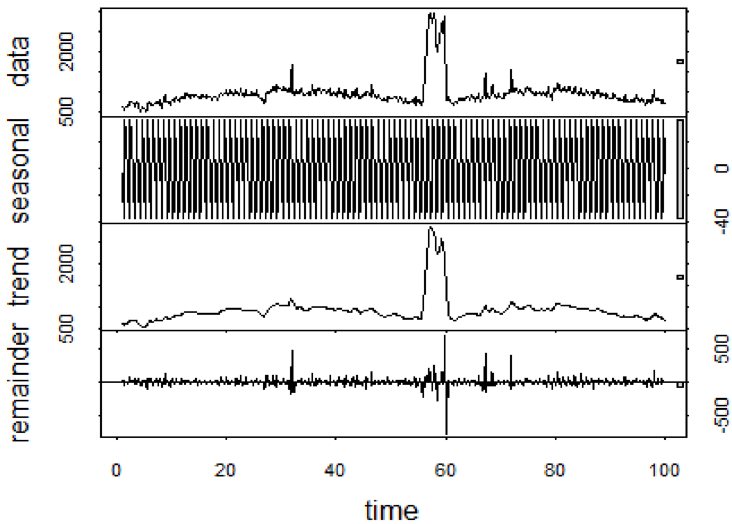
```
wtd_ru_train <- window(wtd_ru_ts, end = 100)
wtd_ru_test <- window(wtd_ru_ts, start = 100)
plot(wtd_ru_train, main = "Rusian Web Traffic Analysis")
```



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)
```



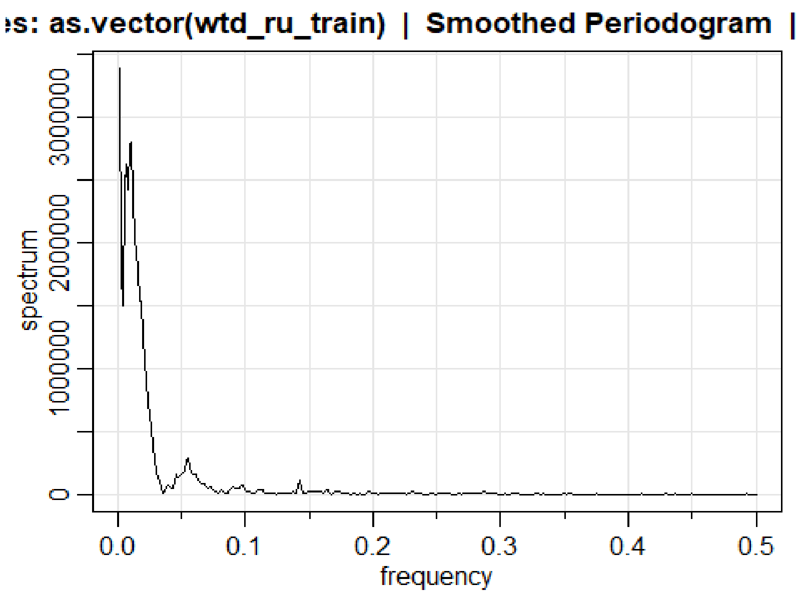
Performing the KPSS Test for stationarity:

```
kpss.test(wtd_ru_train)

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_ru_train
## KPSS Level = 0.51683, Truncation lag parameter = 6, p-value = 0.03788
```

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

Spectral Analysis:

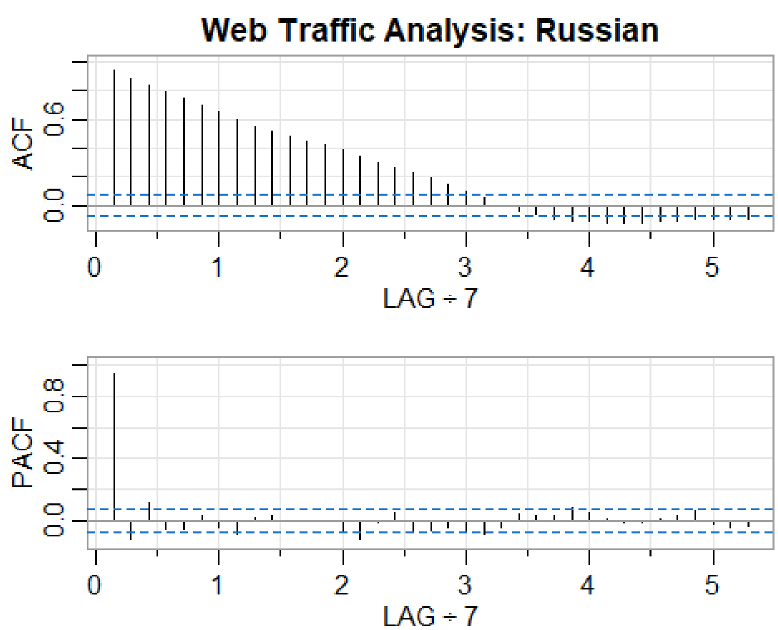
```
wtd_ru.spec <- mvspec(as.vector(wtd_ru_train),detrend = TRUE, spans = 2)
```



There is a weekly seasonality that can be seen in the spectral analysis. There are multiple peaks in the begginning of the plot followed by a peak at around the 60th day ans a small peak around the 140th day. There are no other significant peaks.

Plotting the Autocorrelation plot:

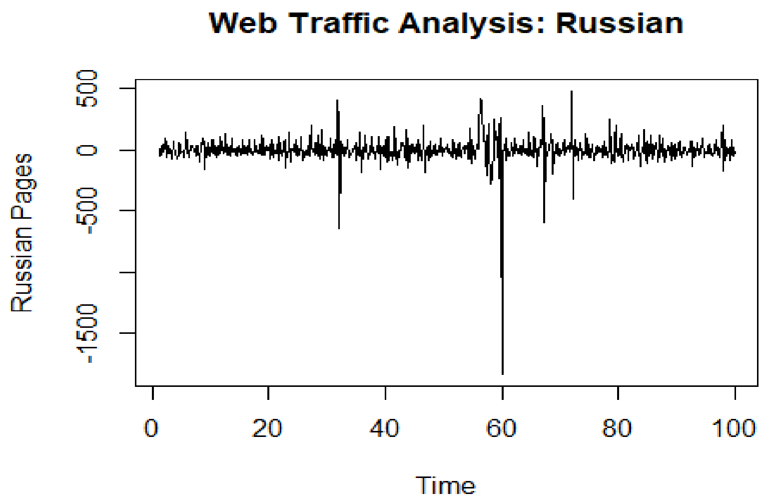
```
acf2(wtd_ru_train, main = "Web Traffic Analysis: Russian")
```



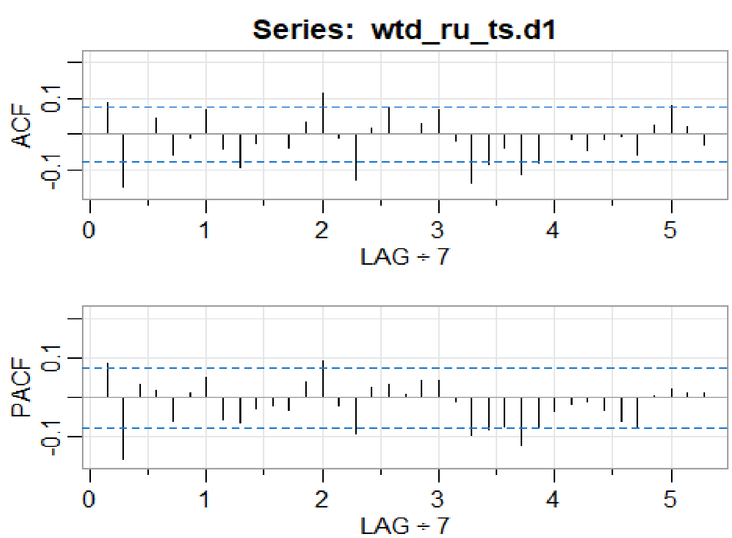
The autocorrelations plot is much different from the other plots that I have seen so far. This cannot be interpreted as an obvious weekly seasonality. However, there is an obvious correlations among the lags. In this case, I have decided to apply the non-seasonal differencing first and check if that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



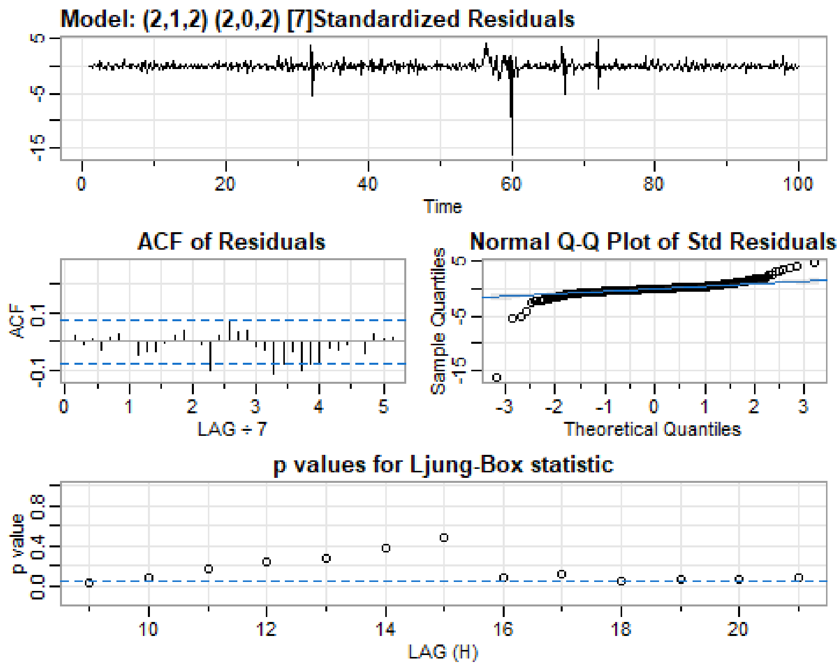

```
kpss.test(wtd_ru_ts.d1)

## 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 = 0.024552, 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)
```



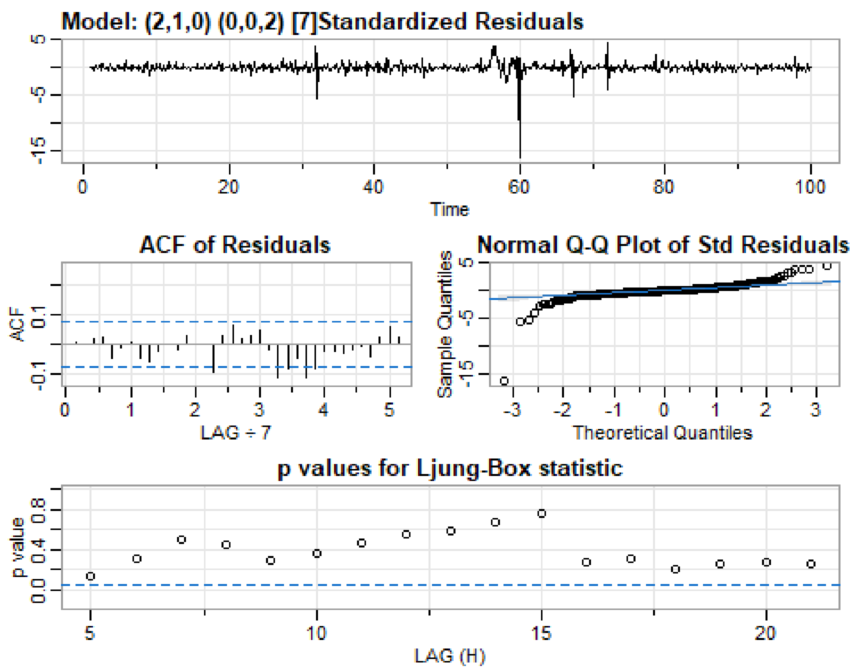

```
wtd_ru_sm1

## $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
##       -0.4577  -0.7330  0.5367  0.6478  0.6681  0.3307  -0.7133  -0.2738
## s.e.   0.0963   0.2122  0.1215  0.2180  0.4466  0.4466   0.4646   0.4620
##       constant
##         0.2647
## s.e.    9.8492
##
## sigma^2 estimated as 11292:  log likelihood = -4220.79,  aic = 8461.57
```

```
##
## $degrees_of_freedom
## [1] 684
##
## $ttable
##          Estimate     SE t.value p.value
## ar1       -0.4577 0.0963 -4.7542  0.0000
## ar2       -0.7330 0.2122 -3.4541  0.0006
## ma1        0.5367 0.1215  4.4174  0.0000
## ma2        0.6478 0.2180  2.9710  0.0031
## sar1       0.6681 0.4466  1.4959  0.1351
## sar2       0.3307 0.4466  0.7405  0.4592
## sma1      -0.7133 0.4646 -1.5352  0.1252
## sma2      -0.2738 0.4620 -0.5926  0.5537
## constant   0.2647 9.8492  0.0269  0.9786
##
## $AIC
## [1] 12.21006
##
## $AICc
## [1] 12.21044
##
## $BIC
## [1] 12.27559
```

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

```
## Series: wtd_ru_train
## ARIMA(2,1,0)(0,0,2)[7]
##
## Coefficients:
##          ar1      ar2    sma1    sma2
##       0.1011  -0.1385  0.0431  0.0965
## s.e.  0.0377   0.0379  0.0378  0.0395
##
## sigma^2 = 11865:  log likelihood = -4232.04
## AIC=8474.09   AICc=8474.18   BIC=8496.79
```

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



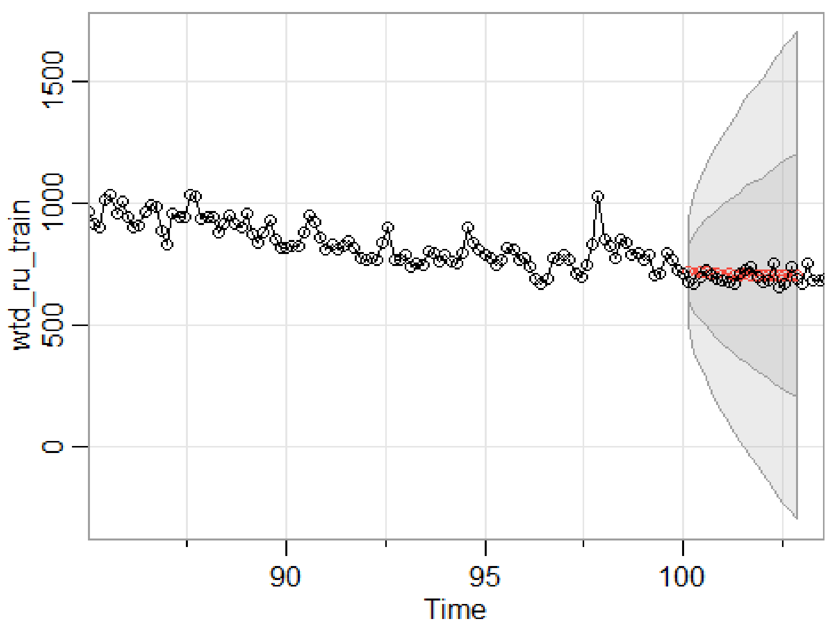

```
wtd_ru_sm2

## $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
##       0.1011  -0.1385  0.0431  0.0964    0.1498
## s.e.  0.0377   0.0379  0.0378  0.0395    4.5237
##
## sigma^2 estimated as 11796:  log likelihood = -4232.04,  aic = 8476.09
##
## $degrees_of_freedom
## [1] 688
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.1011 0.0377  2.6816  0.0075
## ar2       -0.1385 0.0379 -3.6539  0.0003
## sma1       0.0431 0.0378  1.1419  0.2539
## sma2       0.0964 0.0395  2.4401  0.0149
## constant   0.1498 4.5237  0.0331  0.9736
##
## $AIC
```

```
## [1] 12.23101
##
## $AICc
## [1] 12.23113
##
## $BIC
## [1] 12.27032
```

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 = 20,S = 7,
                             p = 2, d = 1, q = 0,
                             P = 0, D = 0, Q = 2)
lines(wtd_ru_test, type = 'o')
```



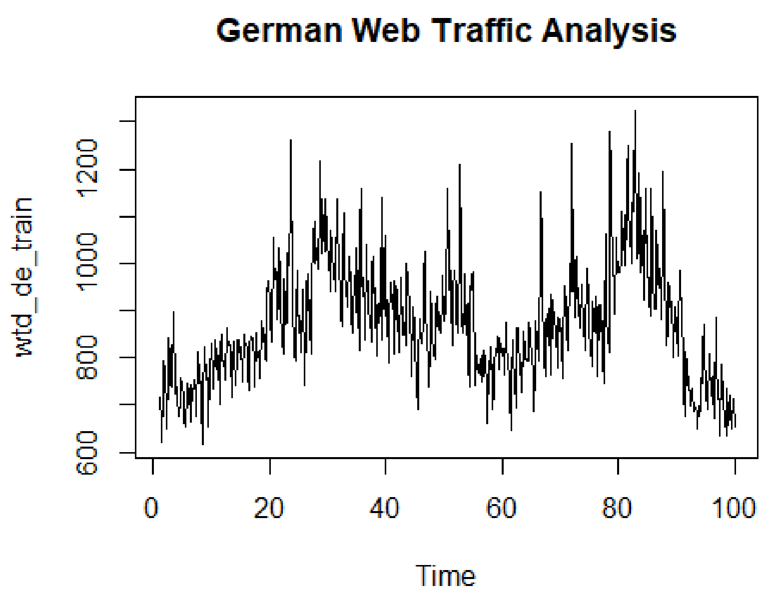
Evaluating accuracy:

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

##                  ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's
U
## Test set -10.71845 28.37902 25.01438 -1.675551 3.606785 0.03478481 0.74942
59
```

German Web Traffic

Splitting the data set into train and test sets:

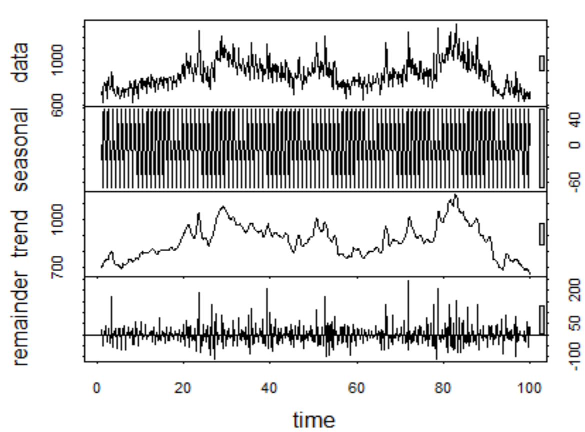
```
wtd_de_train <- window(wtd_de_ts, end = 100)
wtd_de_test <- window(wtd_de_ts, start = 100)
plot(wtd_de_train, main = "German Web Traffic Analysis")
```



Similar to the previous time series, there is a noticeable upward trend in the first few months, followed by a slightly doenward 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)
```



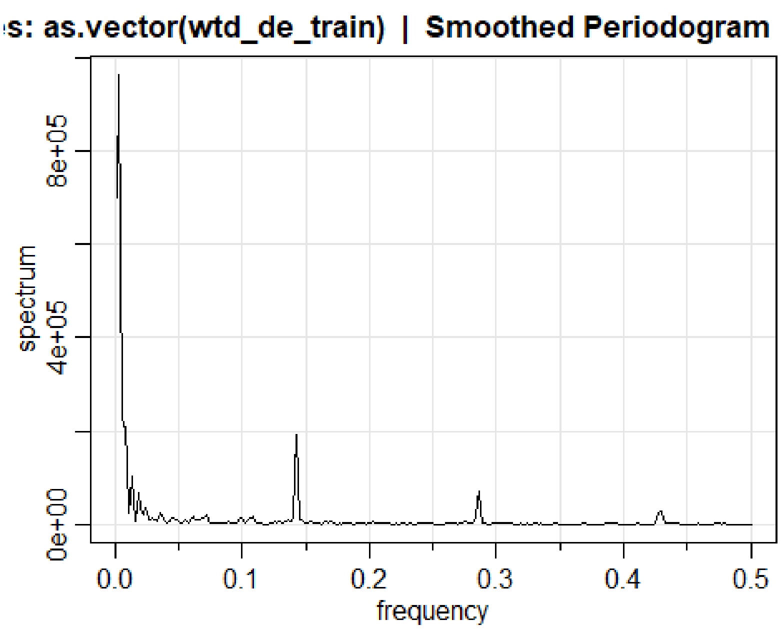
Performing the KPSS stationarity test:

```
kpss.test(wtd_de_train)

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_de_train
## KPSS Level = 0.71898, Truncation lag parameter = 6, p-value = 0.01182
```

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

Spectral Analysis:

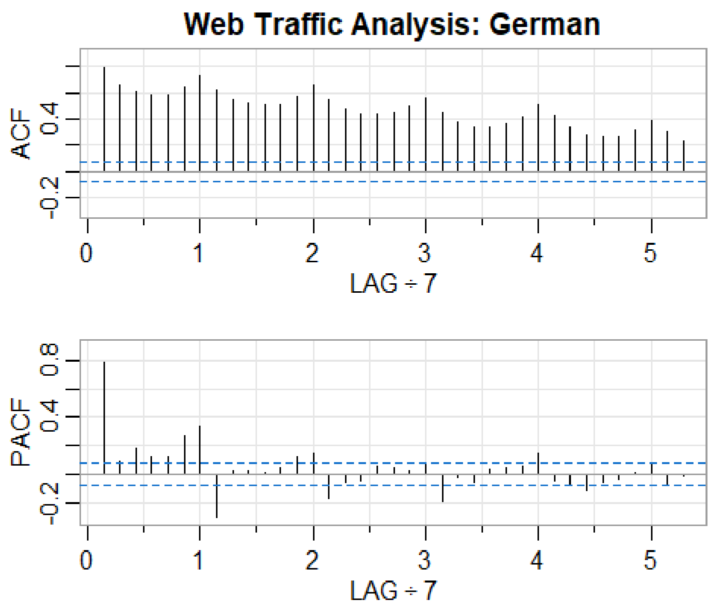
```
wtd_de.spec <- mvspec(as.vector(wtd_de_train),detrend = TRUE, spans = 2)
```



There is a weekly seasonality that can be seen in the spectral analysis. There are also small peaks at the beggining of the plot followed by three significant peaks around 140th, 280th and 420th days (approx). This may signify some kind of quarterly seasonality.

Plotting the autocorrelation plot:

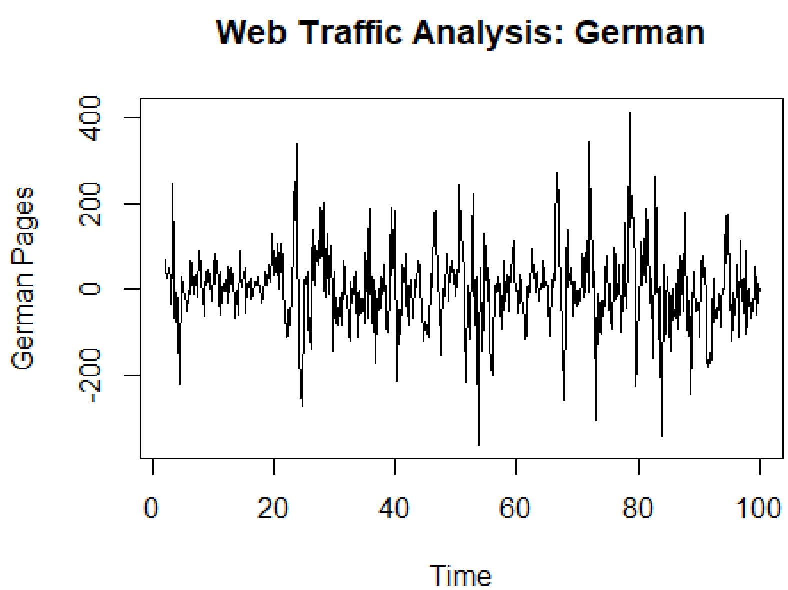
```
acf2(wtd_de_train, main = "Web Traffic Analysis: German")
```



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

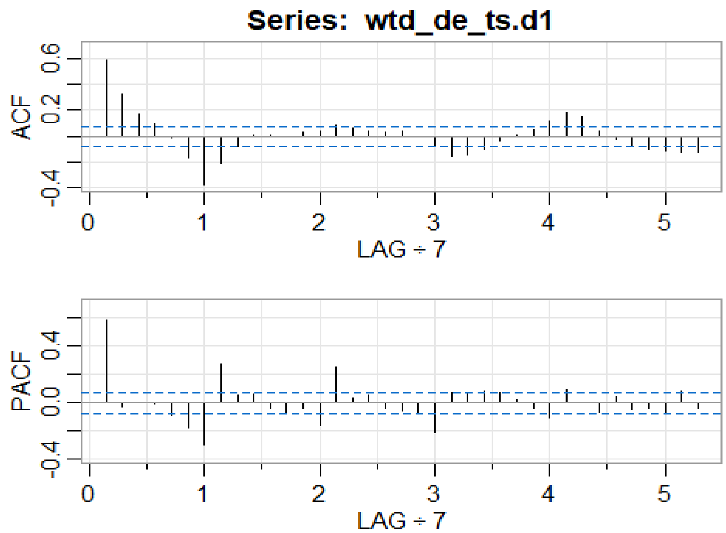
Seasonal Differencing:

```
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)
```

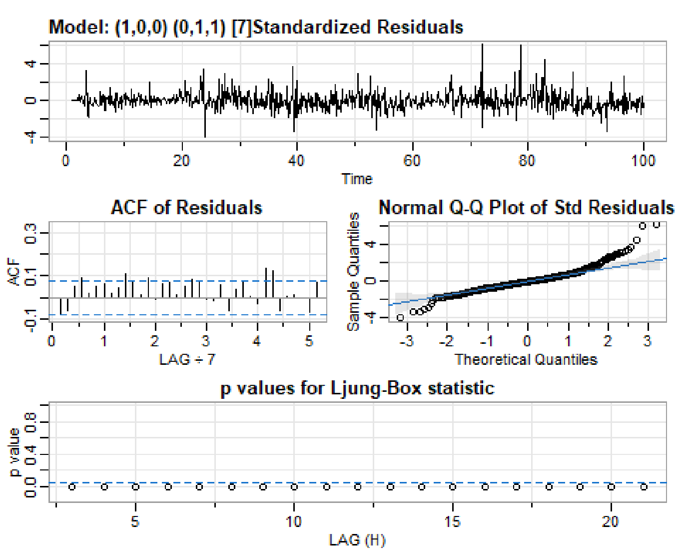
```
## Warning in kpss.test(wtd_de_ts.d1): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_de_ts.d1
## KPSS Level = 0.14813, 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)
```



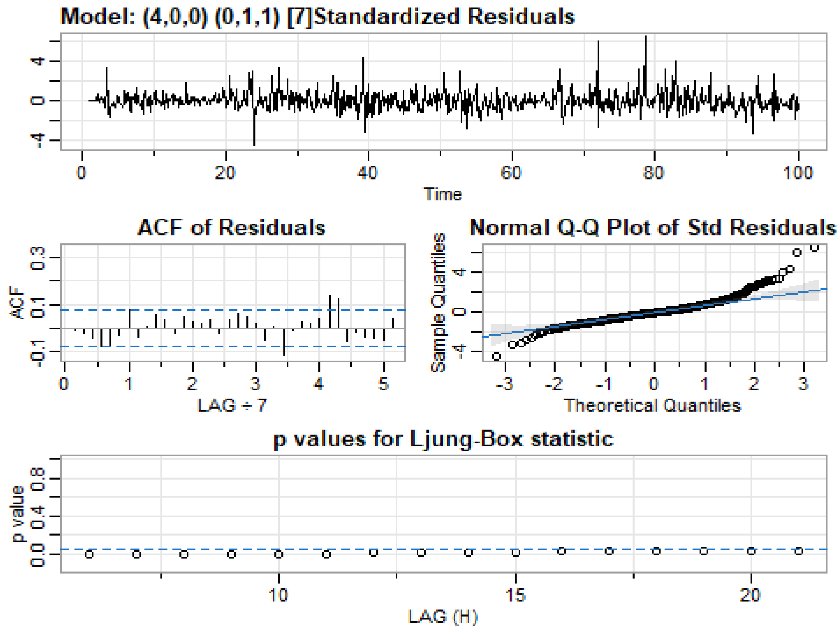
```r
wtd_de_sm1
```

```
## $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
##       0.7661  -0.7948   -0.0115
## s.e.  0.0331   0.0396    0.2851
##
## sigma^2 estimated as 3300:  log likelihood = -3761.51,  aic = 7531.02
##
## $degrees_of_freedom
## [1] 684
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.7661 0.0331  23.1789  0.0000
## sma1      -0.7948 0.0396 -20.0603  0.0000
## constant  -0.0115 0.2851  -0.0403  0.9678
##
## $AIC
## [1] 10.96218
##
## $AICc
## [1] 10.96223
##
## $BIC
## [1] 10.98857
```

```r
auto.arima(wtd_de_train, D=1)
```

```
## Series: wtd_de_train
## ARIMA(4,0,0)(0,1,1)[7]
##
## Coefficients:
##          ar1     ar2     ar3     ar4     sma1
##       0.6712  0.0160  0.1061  0.1128  -0.9204
## s.e.  0.0391  0.0458  0.0456  0.0399   0.0635
##
## sigma^2 = 3122:  log likelihood = -3742.56
## AIC=7497.12   AICc=7497.25   BIC=7524.32
```

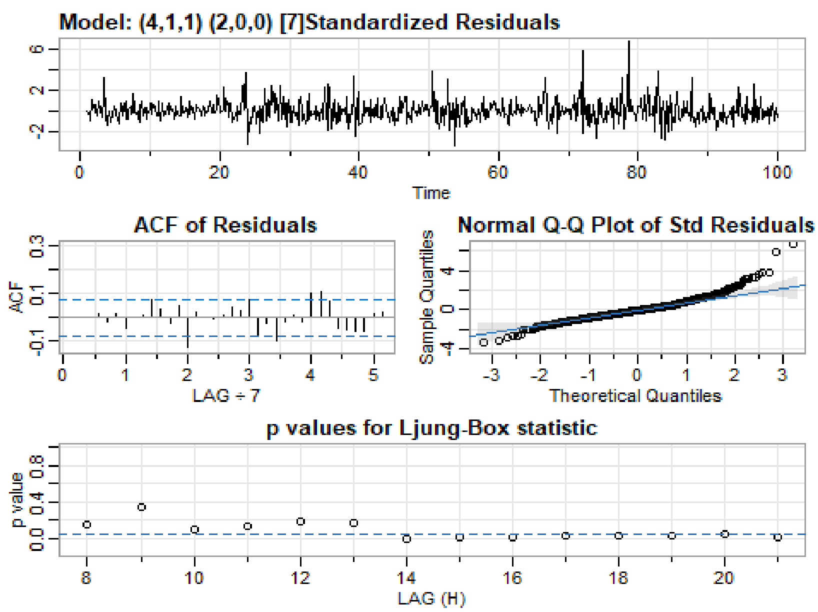
```r
wtd_de_sm2 <- sarima(wtd_de_train,S = 7,
                     p = 4, d = 0, q = 0,
                     P = 0, D = 1, Q = 1)
```




```r
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
##       0.6712  0.0160  0.1061  0.1129  -0.9210    0.0276
## s.e.  0.0392  0.0458  0.0456  0.0400   0.0651    0.2854
##
## sigma^2 estimated as 3099:  log likelihood = -3742.56,  aic = 7499.11
##
## $degrees_of_freedom
## [1] 681
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.6712 0.0392 17.1266  0.0000
## ar2        0.0160 0.0458  0.3503  0.7262
## ar3        0.1061 0.0456  2.3284  0.0202
## ar4        0.1129 0.0400  2.8236  0.0049
```

```
## sma1       -0.9210 0.0651 -14.1385  0.0000
## constant    0.0276 0.2854   0.0968  0.9229
##
## $AIC
## [1] 10.91574
##
## $AICc
## [1] 10.91592
##
## $BIC
## [1] 10.96192
```

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




```
wtd_de_sm3
```

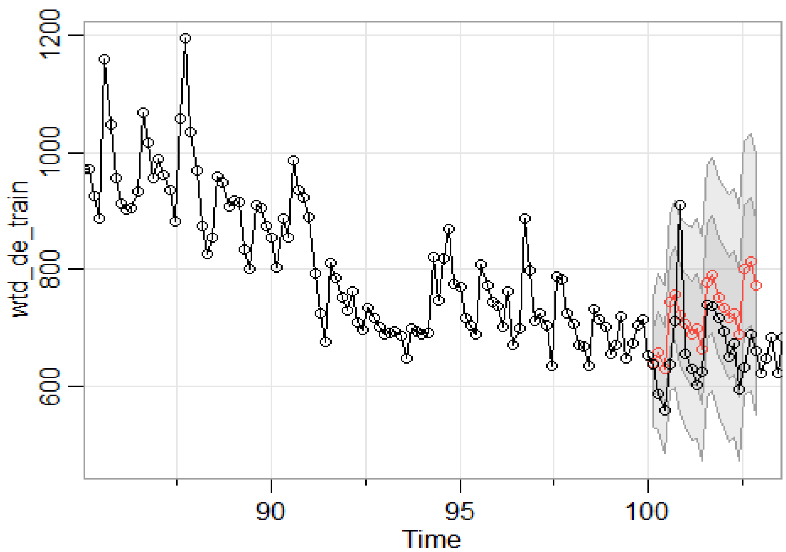
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ma1    sar1    sar2  constant
##       0.5927  -0.0254  0.0423  0.0177  -0.9659  0.3756  0.2685   -0.0443
## s.e.  0.0407   0.0447  0.0446  0.0397   0.0150  0.0375  0.0374    0.5817
##
```

```
## sigma^2 estimated as 3464:  log likelihood = -3809.02,  aic = 7636.05
##
## $degrees_of_freedom
## [1] 685
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.5927 0.0407  14.5680  0.0000
## ar2       -0.0254 0.0447  -0.5670  0.5709
## ar3        0.0423 0.0446   0.9493  0.3428
## ar4        0.0177 0.0397   0.4469  0.6551
## ma1       -0.9659 0.0150 -64.2929  0.0000
## sar1       0.3756 0.0375  10.0110  0.0000
## sar2       0.2685 0.0374   7.1731  0.0000
## constant  -0.0443 0.5817  -0.0762  0.9393
##
## $AIC
## [1] 11.01883
##
## $AICc
## [1] 11.01913
##
## $BIC
## [1] 11.0778
```

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 = 20,S = 7,
                    p = 4, d = 0, q = 0,
                    P = 0, D = 1, Q = 1)
lines(wtd_de_test, type = 'o')
```



Evaluating accuracy:

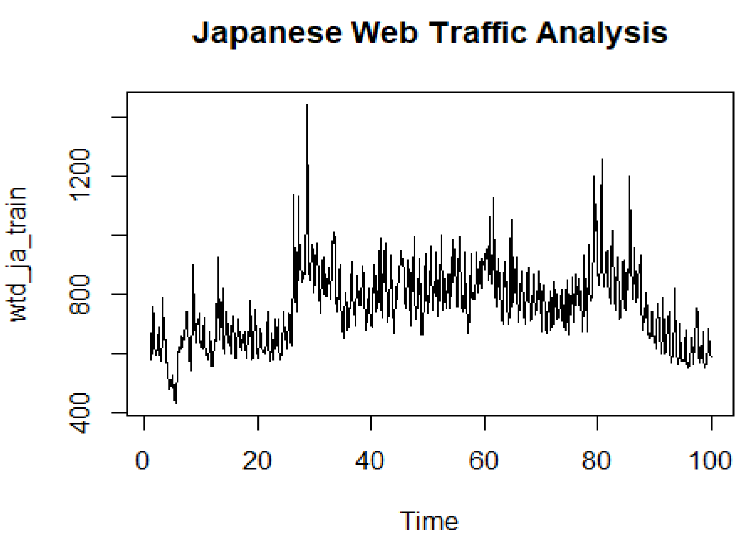
```
accuracy(wtd_de_sm1_for$pred,x = wtd_de_test)

##                     ME     RMSE     MAE      MPE     MAPE      ACF1 Theil's U
## Test set -57.28774 88.50616 76.1618 -9.32977 11.40234 0.2103274  1.173924
```

The RMSE value is 88.50616.

Japanese Web Traffic

Splitting the data set into train and test sets:

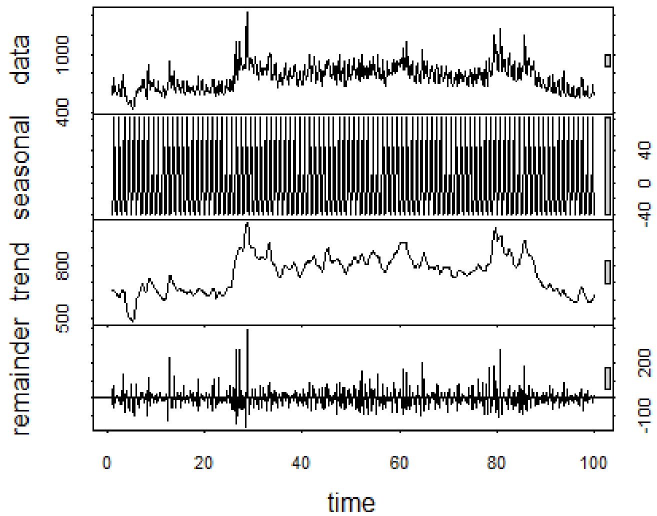
```
wtd_ja_train <- window(wtd_ja_ts, end = 100)
wtd_ja_test <- window(wtd_ja_ts, start = 100)
plot(wtd_ja_train, main = "Japanese Web Traffic Analysis")
```



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)
```



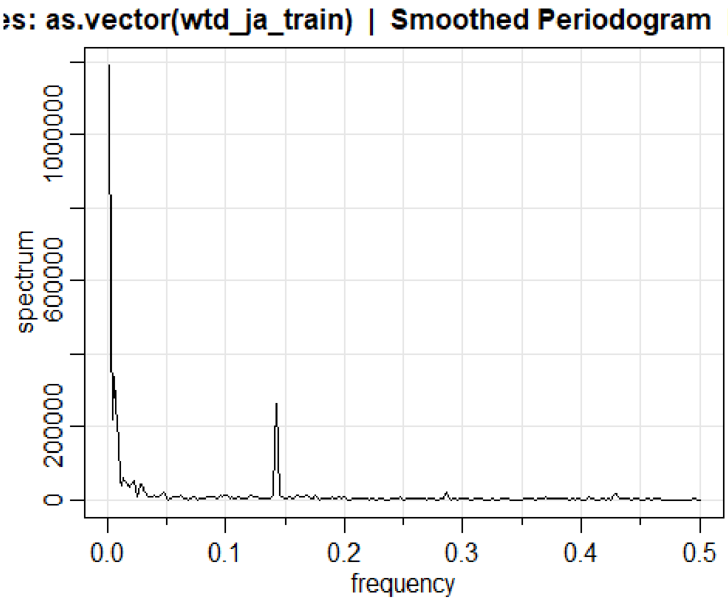
Performing the KPSS stationarity test:

```
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 = 1.8086, 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:

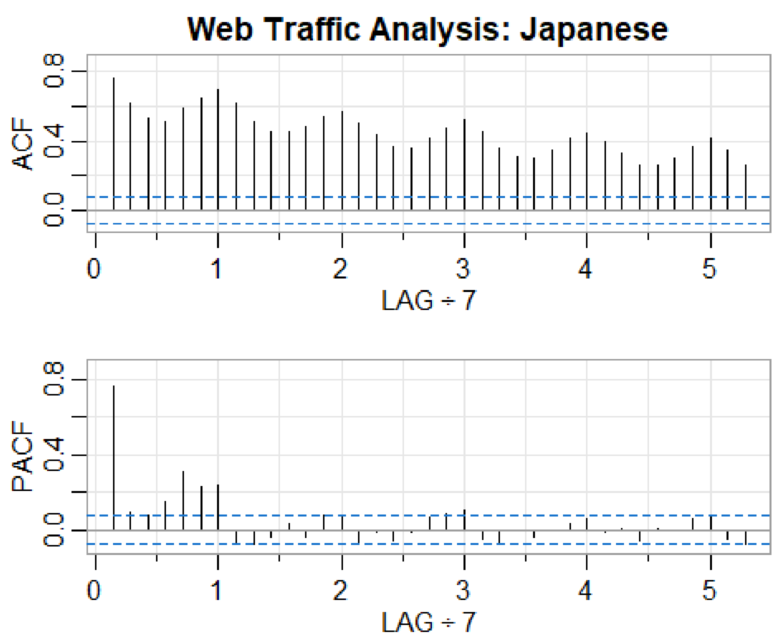
```
wtd_ja.spec <- mvspec(as.vector(wtd_ja_train),detrend = TRUE, spans = 3)
```



There is a weekly seasonality that can be seen in the spectral analysis and a significant spike around the 140th (approx.) but no other significant peaks.

Plotting the autocorrelation plot:

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

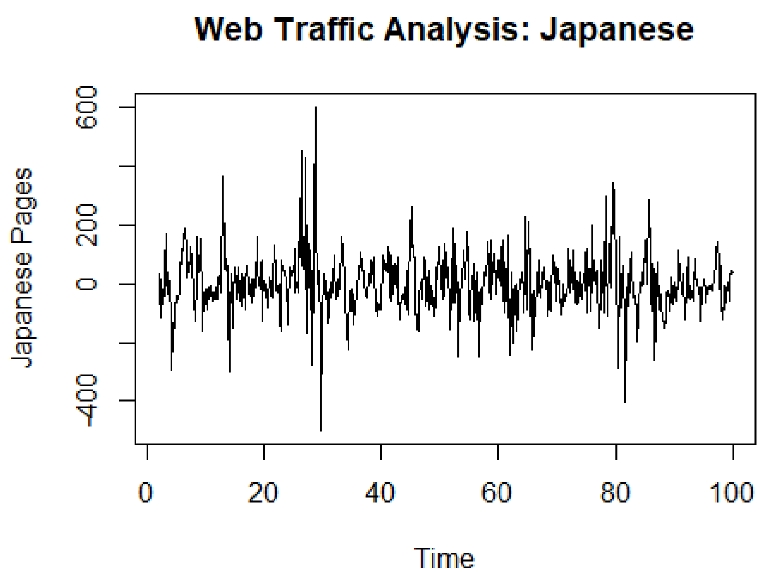


The autocorrelations shows a high lag every 7 days which is an indication of a weekly seasonality.

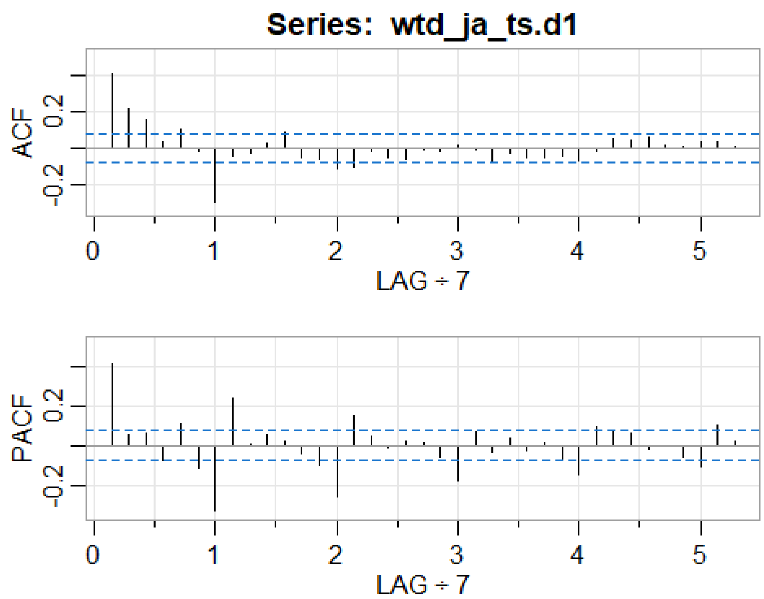
Seasonal Differencing:

```
wtd_ja_ts.d1 <- diff(wtd_ja_train, lag = 7)
plot(wtd_ja_ts.d1,
```

```r
     main = "Web Traffic Analysis: Japanese",
     ylab = "Japanese Pages", type = 'l')
```

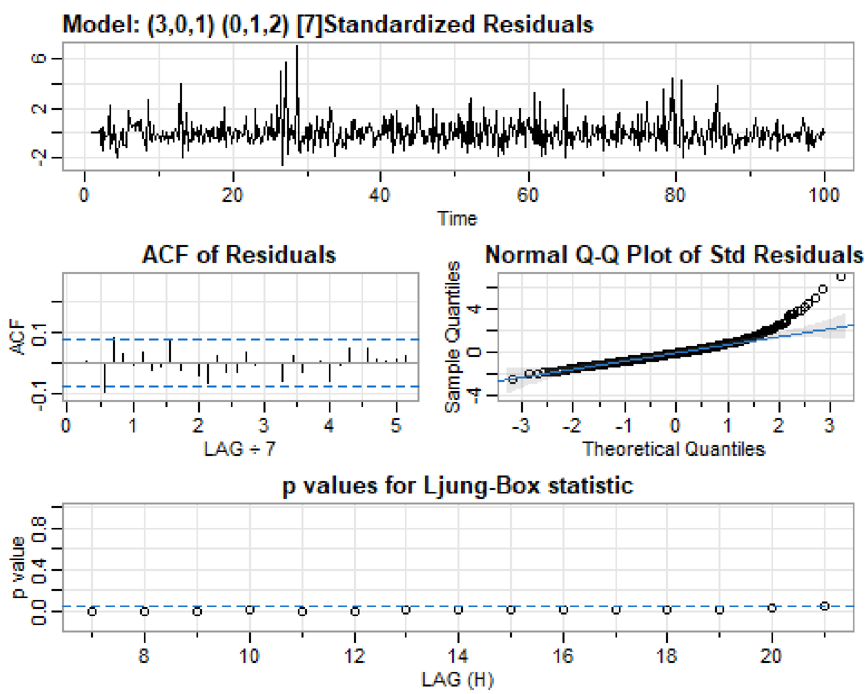



```r
kpss.test(wtd_ja_ts.d1)

## 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 = 0.095455, 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 auti ARIMA to find a good fot for the model.

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



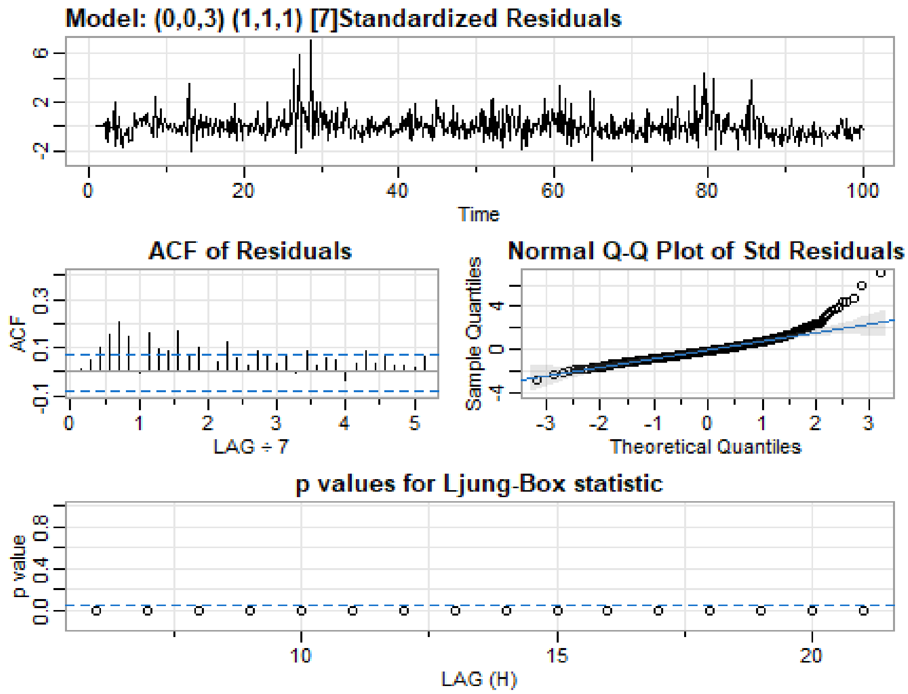

```
wtd_ja_sm1
```

```
## $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
##       1.2944  -0.3118  0.0067  -0.8147  -0.8950  -0.0915    0.0593
## s.e.  0.0610   0.0634  0.0468   0.0480   0.0445   0.0426    0.2085
##
## sigma^2 estimated as 4484:  log likelihood = -3874.11,  aic = 7764.22
##
## $degrees_of_freedom
## [1] 680
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        1.2944 0.0610  21.2184  0.0000
## ar2       -0.3118 0.0634  -4.9188  0.0000
## ar3        0.0067 0.0468   0.1428  0.8865
## ma1       -0.8147 0.0480 -16.9835  0.0000
## sma1      -0.8950 0.0445 -20.1163  0.0000
## sma2      -0.0915 0.0426  -2.1475  0.0321
## constant   0.0593 0.2085   0.2846  0.7761
##
## $AIC
## [1] 11.30163
##
## $AICc
## [1] 11.30187
##
## $BIC
## [1] 11.35441
```

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

```
## Series: wtd_ja_train
## ARIMA(0,0,3)(1,1,1)[7]
##
## Coefficients:
##          ma1     ma2     ma3     sar1     sma1
##       0.5465  0.3574  0.2188  0.1719  -0.8556
## s.e.  0.0402  0.0431  0.0332  0.0498   0.0298
##
## sigma^2 = 5323:  log likelihood = -3923.29
## AIC=7858.57   AICc=7858.7   BIC=7885.77
```

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



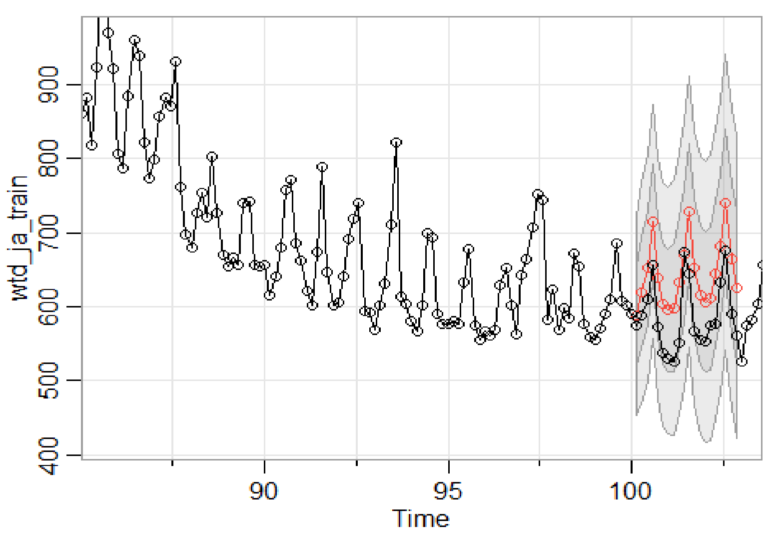

```
wtd_ja_sm2

## $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
##       0.5468  0.3577  0.2189  0.1731  -0.8569    0.0490
## s.e.  0.0402  0.0432  0.0332  0.0499   0.0301    0.1556
##
## sigma^2 estimated as 5284:  log likelihood = -3923.24,  aic = 7860.48
##
## $degrees_of_freedom
## [1] 681
##
## $ttable
##          Estimate     SE  t.value p.value
## ma1        0.5468 0.0402  13.6019  0.0000
## ma2        0.3577 0.0432   8.2873  0.0000
```

```
## ma3         0.2189 0.0332   6.5932  0.0000
## sar1        0.1731 0.0499   3.4674  0.0006
## sma1       -0.8569 0.0301 -28.4711  0.0000
## constant    0.0490 0.1556   0.3148  0.7530
##
## $AIC
## [1] 11.44174
##
## $AICc
## [1] 11.44192
##
## $BIC
## [1] 11.48792
```

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 = 20,S = 7,
                             p = 3, d = 0, q = 1,
                            P = 0, D = 1, Q = 2)
lines(wtd_ja_test, type = 'o')
```



Evaluating accuracy:

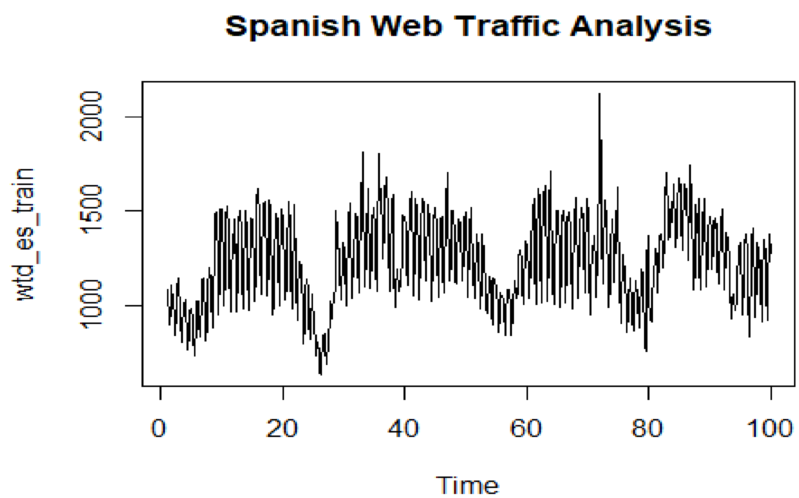
```
accuracy(wtd_ja_sm1_for$pred,x = wtd_ja_test)

##                 ME     RMSE      MAE       MPE     MAPE        ACF1 Theil'
s U
## Test set -56.49645 60.98373 57.06423 -9.761464 9.845564 -0.01553993  1.268
637
```

The RMSE value is 60.98373.

Spanish Web Traffic

Splitting the data set into train and test sets:

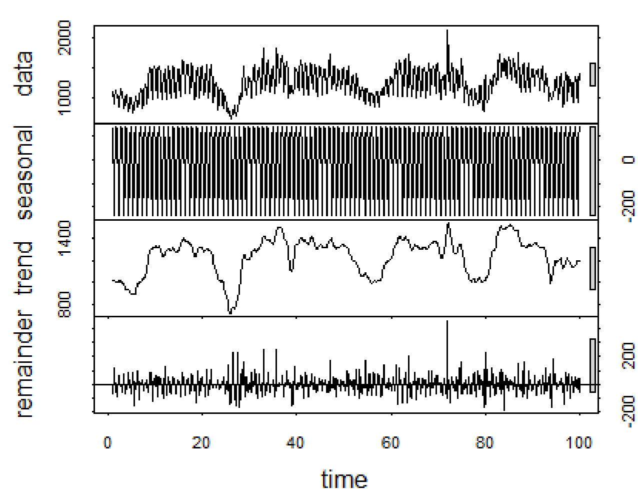
```
wtd_es_train <- window(wtd_es_ts, end = 100)
wtd_es_test <- window(wtd_es_ts, start = 100)
plot(wtd_es_train, main = "Spanish Web Traffic Analysis")
```



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)
```



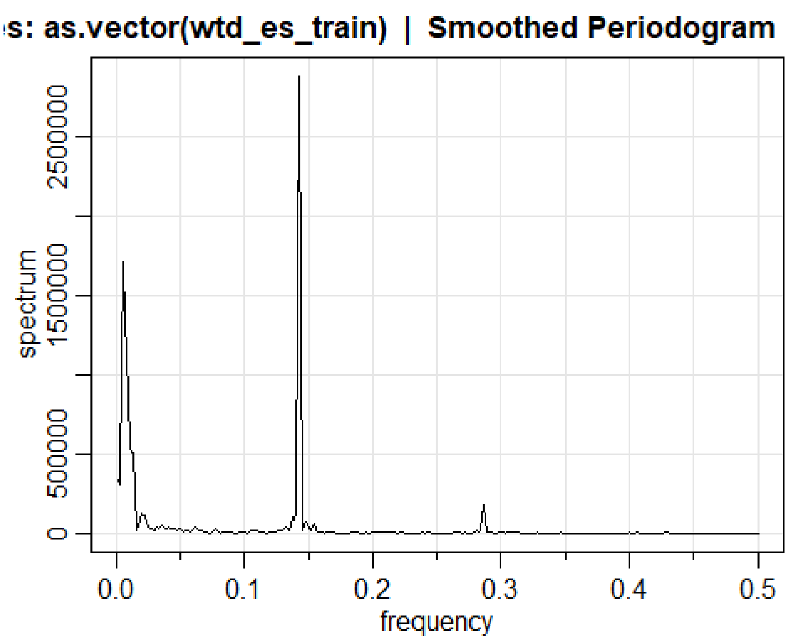
Performing the KPSS test for stationarity:

```
kpss.test(wtd_es_train)

##
##  KPSS Test for Level Stationarity
##
## data:  wtd_es_train
## KPSS Level = 0.69788, Truncation lag parameter = 6, p-value = 0.01374
```

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

Spectral Analysis:

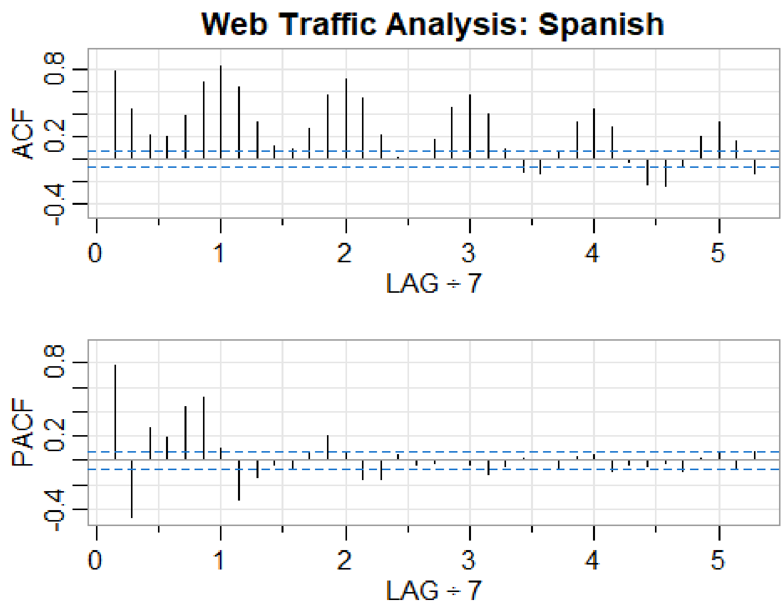
```
wtd_es.spec <- mvspec(as.vector(wtd_es_train),detrend = TRUE, spans = 3)
```



There is a weekly seasonality that can be seen in the spectral analysisand a very large spike around the 140th day (approx.). There is also another small spike around 280th day but no other significant peaks. The spike is larger than seen on any of the other time series. I am not really sure how to interpret this data.

Plotting the autocorrelation plot:

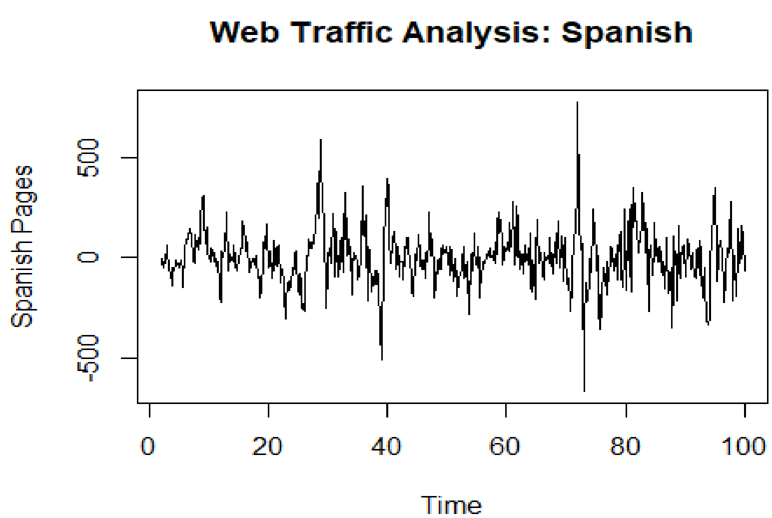
```
acf2(wtd_es_train, main = "Web Traffic Analysis: Spanish")
```



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

Seasonal Differencing:

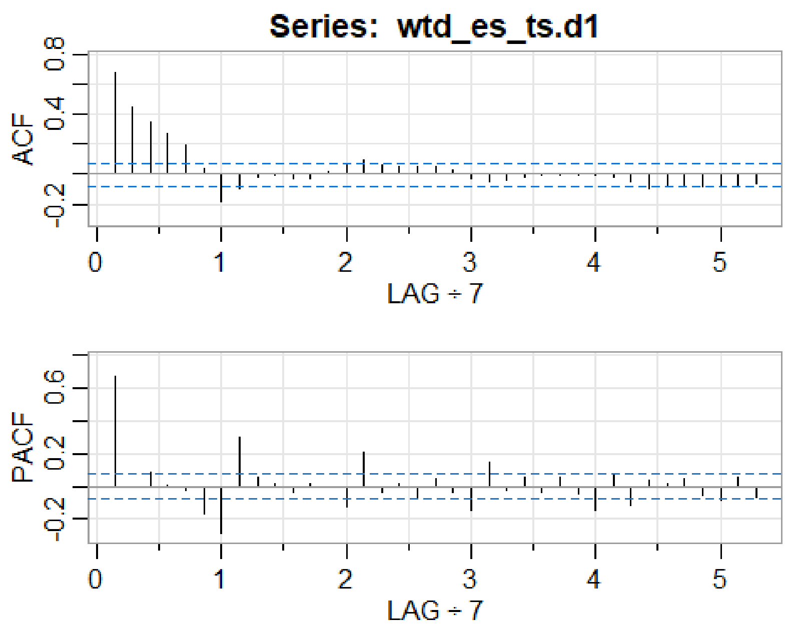
```
wtd_es_ts.d1 <- diff(wtd_es_train, lag = 7)
plot(wtd_es_ts.d1,
     main = "Web Traffic Analysis: Spanish",
     ylab = "Spanish Pages", type = 'l')
```




```
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 = 0.050917, Truncation lag parameter = 6, p-value = 0.1
```

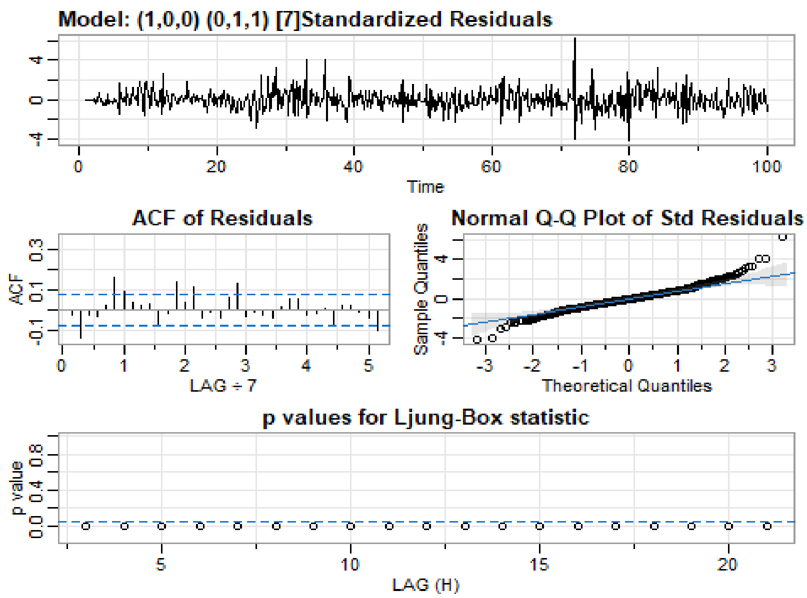
```
acf2(wtd_es_ts.d1)
```



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)
```



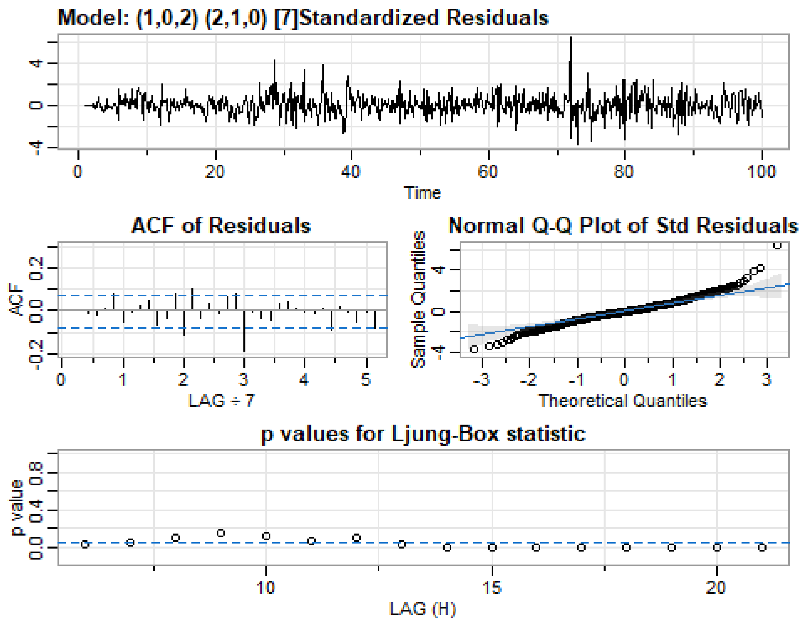

```
wtd_es_sm1

## $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
##       0.8643  -0.7854    0.2618
## s.e.  0.0239   0.0627    0.7429
##
## sigma^2 estimated as 7009:  log likelihood = -4020.21,  aic = 8048.41
##
## $degrees_of_freedom
## [1] 684
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.8643 0.0239  36.1525  0.0000
## sma1      -0.7854 0.0627 -12.5209  0.0000
## constant   0.2618 0.7429   0.3524  0.7246
##
## $AIC
## [1] 11.7153
##
## $AICc
## [1] 11.71535
```

```
##
## $BIC
## [1] 11.74169
```

```r
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
##       0.9016  -0.1740  -0.1820  -0.5587  -0.2356
## s.e.  0.0249   0.0467   0.0442   0.0382   0.0379
##
## sigma^2 = 7452:  log likelihood = -4036.48
## AIC=8084.97   AICc=8085.09   BIC=8112.16
```

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




```r
wtd_es_sm2
```

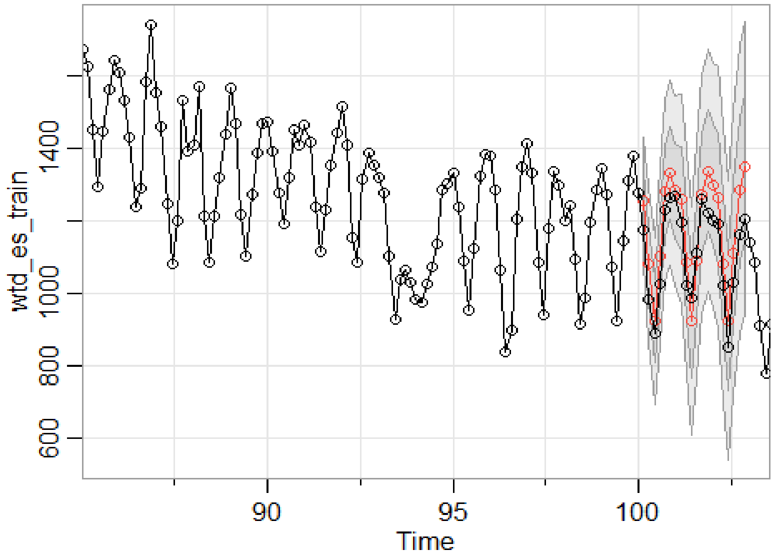
```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), p
eriod = S),
##     xreg = constant, transform.pars = trans, fixed = fixed, optim.control
= list(trace = trc,
```

```
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1      ma2     sar1     sar2  constant
##       0.9015  -0.1739  -0.1820  -0.5587  -0.2356    0.2135
## s.e.  0.0249   0.0467   0.0442   0.0382   0.0379    1.6983
##
## sigma^2 estimated as 7398:  log likelihood = -4036.48,  aic = 8086.95
##
## $degrees_of_freedom
## [1] 681
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1        0.9015 0.0249  36.2294   0e+00
## ma1       -0.1739 0.0467  -3.7273   2e-04
## ma2       -0.1820 0.0442  -4.1160   0e+00
## sar1      -0.5587 0.0382 -14.6146   0e+00
## sar2      -0.2356 0.0379  -6.2099   0e+00
## constant   0.2135 1.6983   0.1257   9e-01
##
## $AIC
## [1] 11.7714
##
## $AICc
## [1] 11.77158
##
## $BIC
## [1] 11.81758
```

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 = 20,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 -62.80452 78.99408 71.34132 -5.636945 6.478437 0.4279138 0.601212
1
```

The RMSE value is 78.99408.

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.

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/