Time Series Project

Natalia Pludra, Gasper Pust

```r
library(xts)
library(forecast)
library(tseries)
library(astsa)
library(ggplot2)
library(dplyr)
library(knitr)
# library(gridExtra)
```

Introduction

Forecasting real-world time series data is a fundamental theme in statistical modeling. This project focuses on analyzing and forecasting website traffic for an academic teaching notes website using robust statistical methods. The objective is to develop accurate models for predicting web traffic, leveraging patterns in the data. The report documents the full analytical process, including data preparation, model building, and validation.

Dataset

This dataset contains five years of daily time series data capturing various traffic metrics for a statistical forecasting teaching notes website (https://regressit.com/statforecasting.com/). The data was collected using StatCounter, a web traffic monitoring tool.

The dataset contains 2 167 rows of data from **September 14, 2014**, to **August 19, 2020** and includes daily counts of:

- **Page Loads:** Total pages accessed on the site.
- **Unique Visitors:** Distinct users visiting the site, identified by IP address.
- **First-Time Visitors:** Users accessing the site for the first time, identified by the absence of prior cookies.
- **Returning Visitors:** Users with prior visits, identified through cookies when accepted.

The data exhibits complex seasonality influenced by both the day of the week and the academic calendar.

The source of the data is Kaggle (https://www.kaggle.com/datasets/bobnau/daily-website-visitors).

```r
df_website <- read.csv("daily-website-visitors.csv")

df_website$Page.Loads <- as.numeric(gsub(",", ".", gsub("\\.", "", df_website$Page.Loads)))
df_website$Date <- as.Date(df_website$Date,format = "%m/%d/%Y")

kable(head(df_website), caption="Table1: Sample data")
```

Table 1: Table1: Sample data

Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
1	Sunday	1	2014-09-14	2.146	1,582	1,430	152
2	Monday	2	2014-09-15	3.621	2,528	2,297	231
3	Tuesday	3	2014-09-16	3.698	2,630	2,352	278
4	Wednesday	4	2014-09-17	3.667	2,614	2,327	287
5	Thursday	5	2014-09-18	3.316	2,366	2,130	236
6	Friday	6	2014-09-19	2.815	1,863	1,622	241

We decided to focus on Daily Page Loads.

Exploratory Data Analysis

The first step of the project was EDA. Figure 1 shows our time series.

```r
ts_website <- xts(df_website$Page.Loads, df_website$Date)

plot(ts_website, main = "Daily Page Loads", ylab = "Page Loads",lwd=1.2)
```

We divide the data into a training set and a test set. The test set will contain the last 6 months of observations.

```r
# Training set: First 4.5 years, Test set: Last 6 months
cutoff_date <- as.Date("2020-02-19")
train_data <- window(ts_website, end = cutoff_date)
test_data <- window(ts_website, start = cutoff_date + 1)

plot(train_data, main = "Daily Page Loads", ylab = "Page Loads", xlab = "Date",lwd=1.2)
```

The plot of the training set does not indicate the presence of a trend or heteroscedasticity. However, there is evidence of cyclic patterns in the data. We can also notice unusual observations in 2017. The number of page loads was significantly lower than in other years.

Basic statistics and distribution of the Daily Page Loads are presented in Table 2.

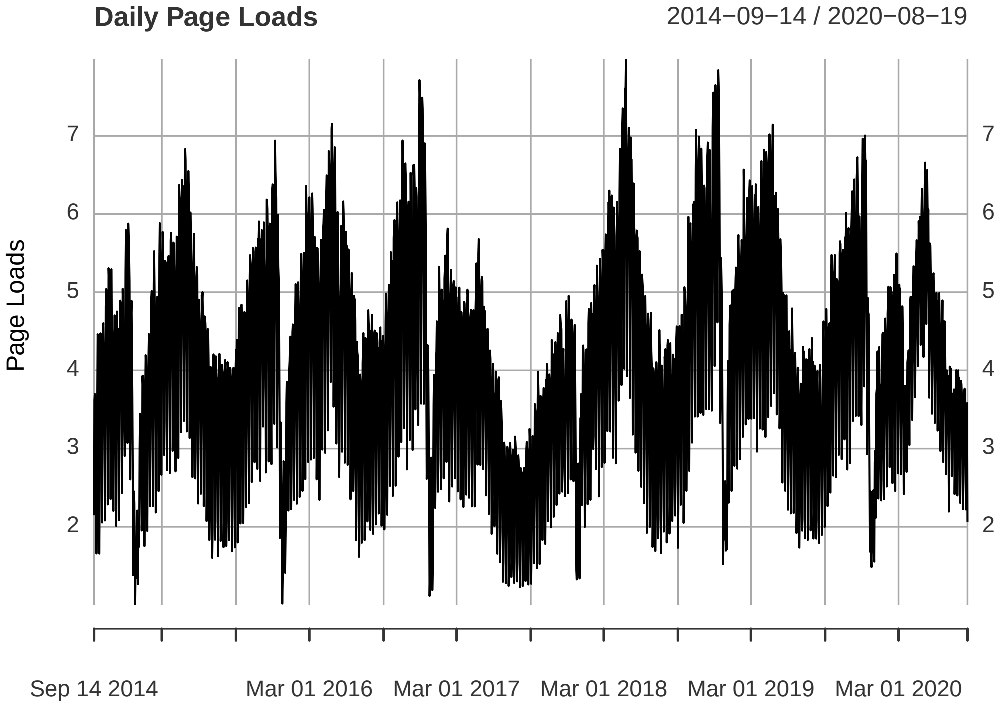


Figure 1: Daily Page Loads

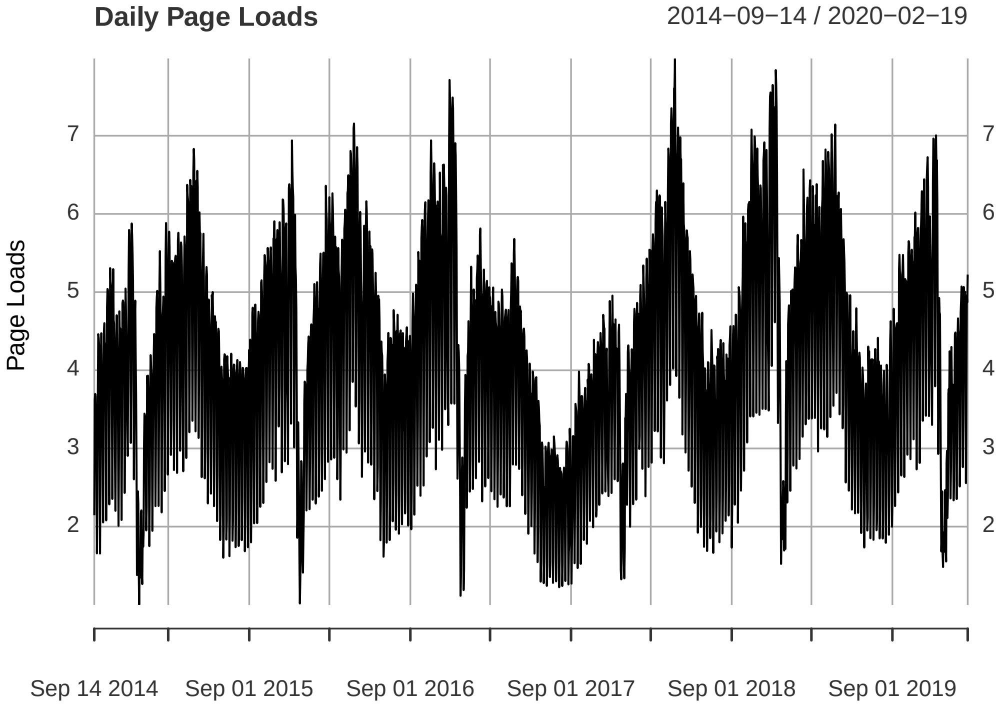


Figure 2: Daily Page Loads - training set

```r
summary(df_website$Page.Loads)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.002   3.115   4.106   4.117   5.021   7.984
```

```r
ggplot(df_website, aes(Page.Loads))+
  geom_histogram(fill="peachpuff3",color="black")+
  labs(x="Page Loads")+
  theme_minimal()
```

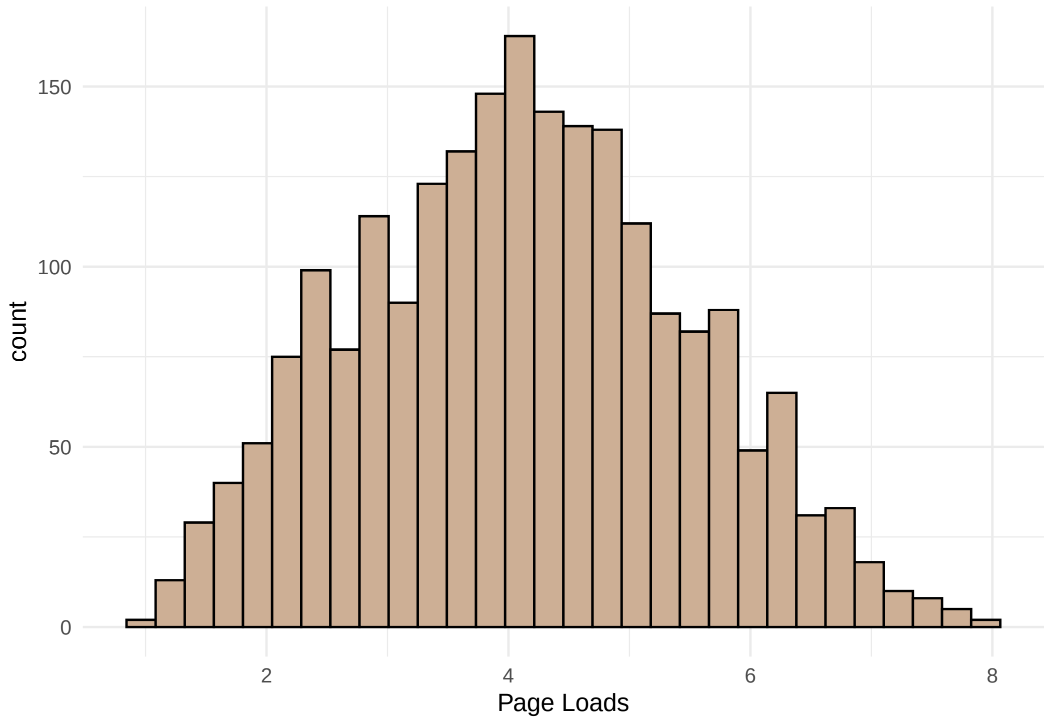


Figure 3: Histogram of Daily Page Loads

There is no missing values in our data.

In Figure 3 we present boxplots of Page Loads by day of the week.

```r
ggplot(df_website,aes(x=factor(Day,levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Fr
  geom_boxplot(aes(fill=Day))+
  labs(x="Day of the week", y="Page Loads")+
  theme_minimal()+
  theme(legend.position = "none")
```

We can observe that website traffic is lower during weekends.

We will check if the time series is stationary using Augmented Dickey-Fuller (ADF) test.

```r
adf.test(ts_website, alternative = "stationary")
```

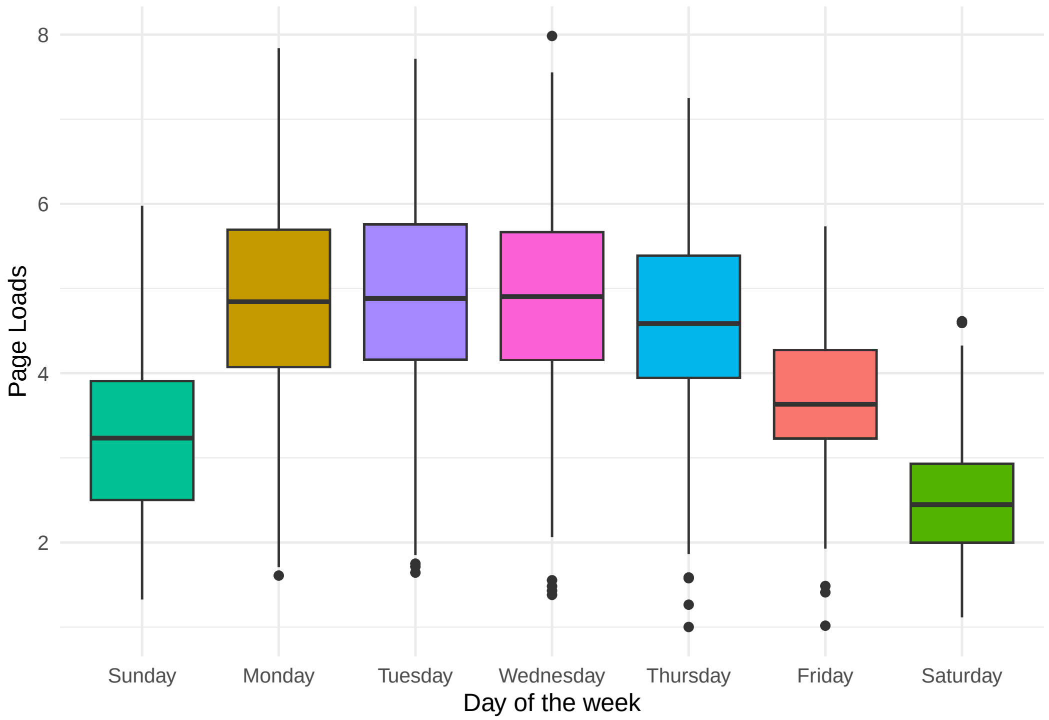


Figure 4: Boxplot of Page Loads by day of the week

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_website
## Dickey-Fuller = -5.4532, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

Small p-value indicates that the time series is stationary (assuming significance level of 0.05).

Next, we proceed to compute the sample ACF and PACF for further analysis.

```
invisible(astsa::acf2(ts_website,max.lag= 80))
```

The seasonality feature are present in the sample ACF which shows cycles of 7 days - we have weekly seasonality.

```
lag1.plot(ts_website,16)
```

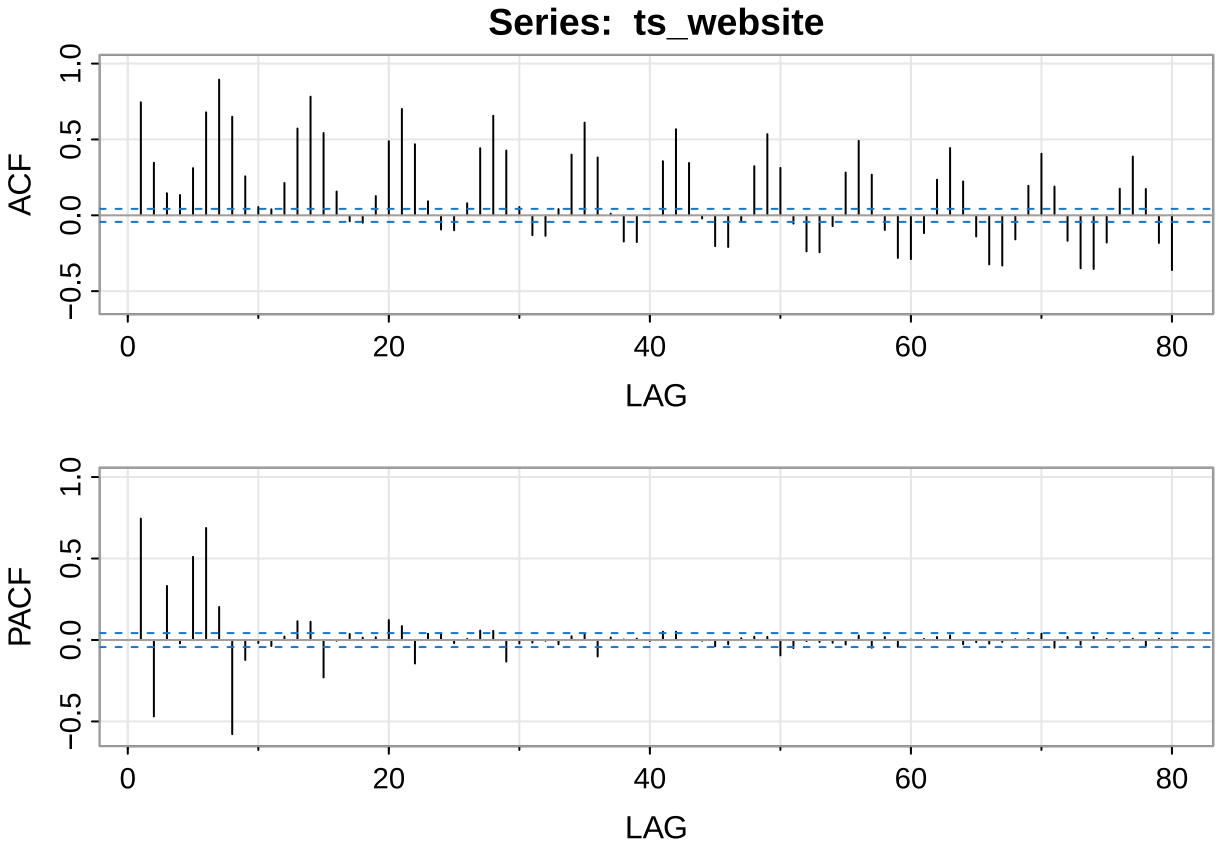
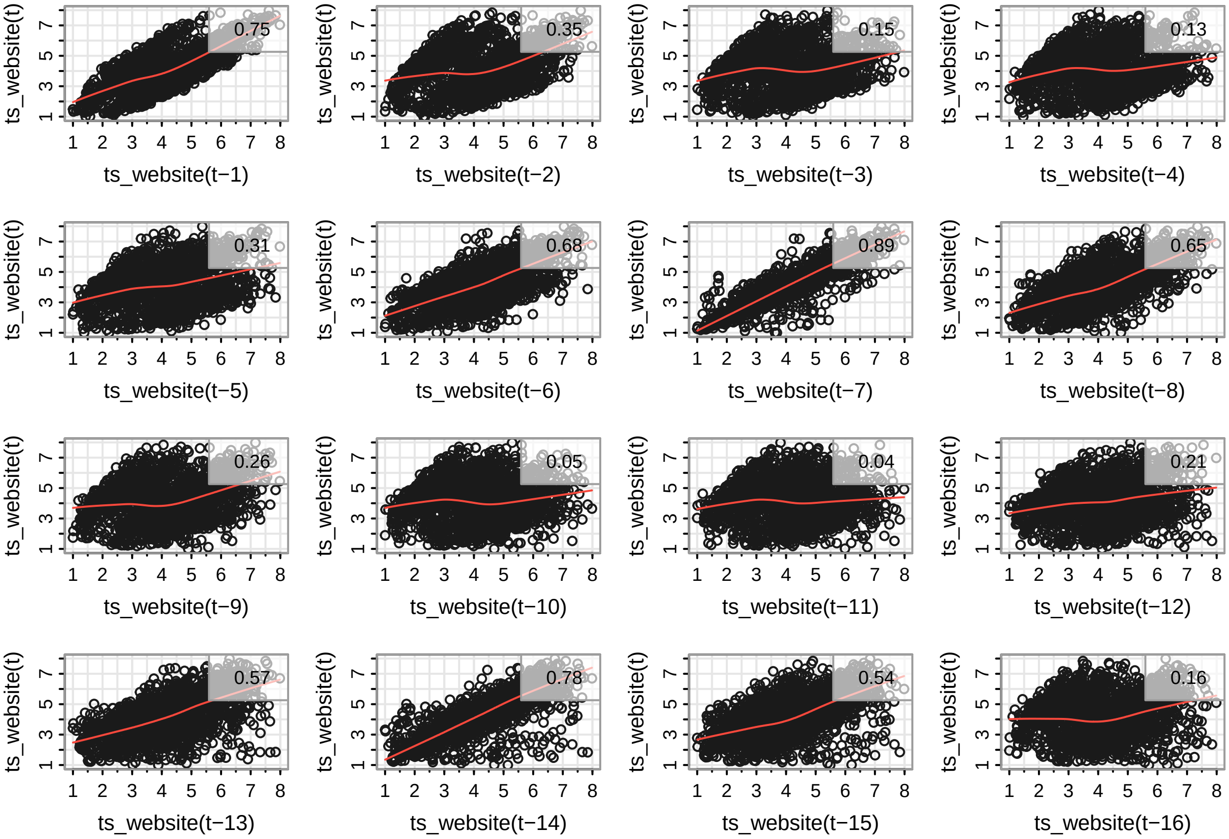


Figure 5: Figure 3: ACF and PACF



We can see the strongest correlation at lag 1, 7 and 14.

Seasonal Component

As we observed in the previous part, there is a weekly seasonal component in our time series. We will try various types of differencing to remove this component.

```
dlm1 <- diff(train_data,1)
plot(dlm1,lwd=1,main=expression(paste(Delta, "Page Loads train")))
```

```
invisible(acf2(dlm1,main=expression(paste(Delta, "Page Loads train"))))
```

```
dlm7 <- diff(train_data,7)
plot(dlm7,lwd=1,,main=expression(paste(Delta[7], "Page Loads train")))
```

```
invisible(acf2(dlm7,main=expression(paste(Delta[7], "Page Loads train"))))
```

```
dlm7.1 <- diff(diff(train_data,7),1)
plot(dlm7.1,lwd=1,main=expression(paste(Delta,Delta[7], "Page Loads train")))
```

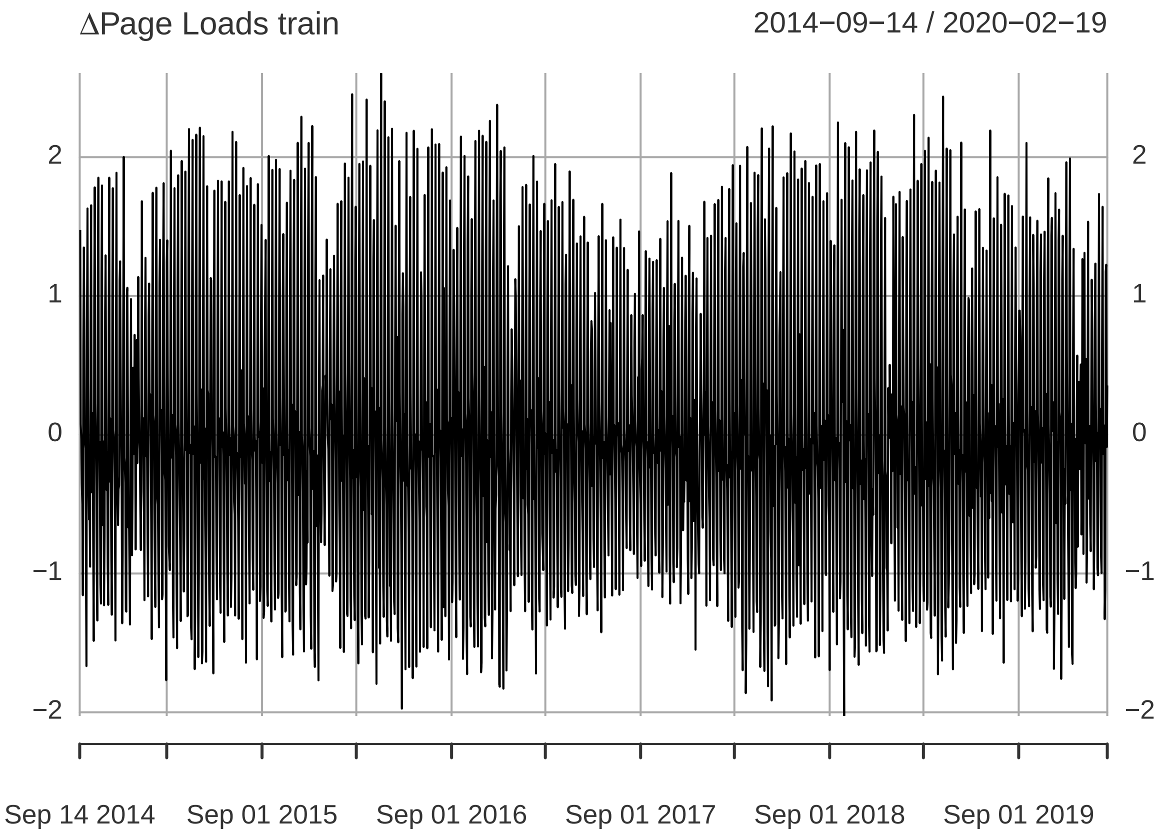

ΔPage Loads train
2014−09−14 / 2020−02−19
2
1
0
−1
−2
2
1
0
−1
−2
Sep 14 2014
Sep 01 2015
Sep 01 2016
Sep 01 2017
Sep 01 2018
Sep 01 2019

Figure 6: ...

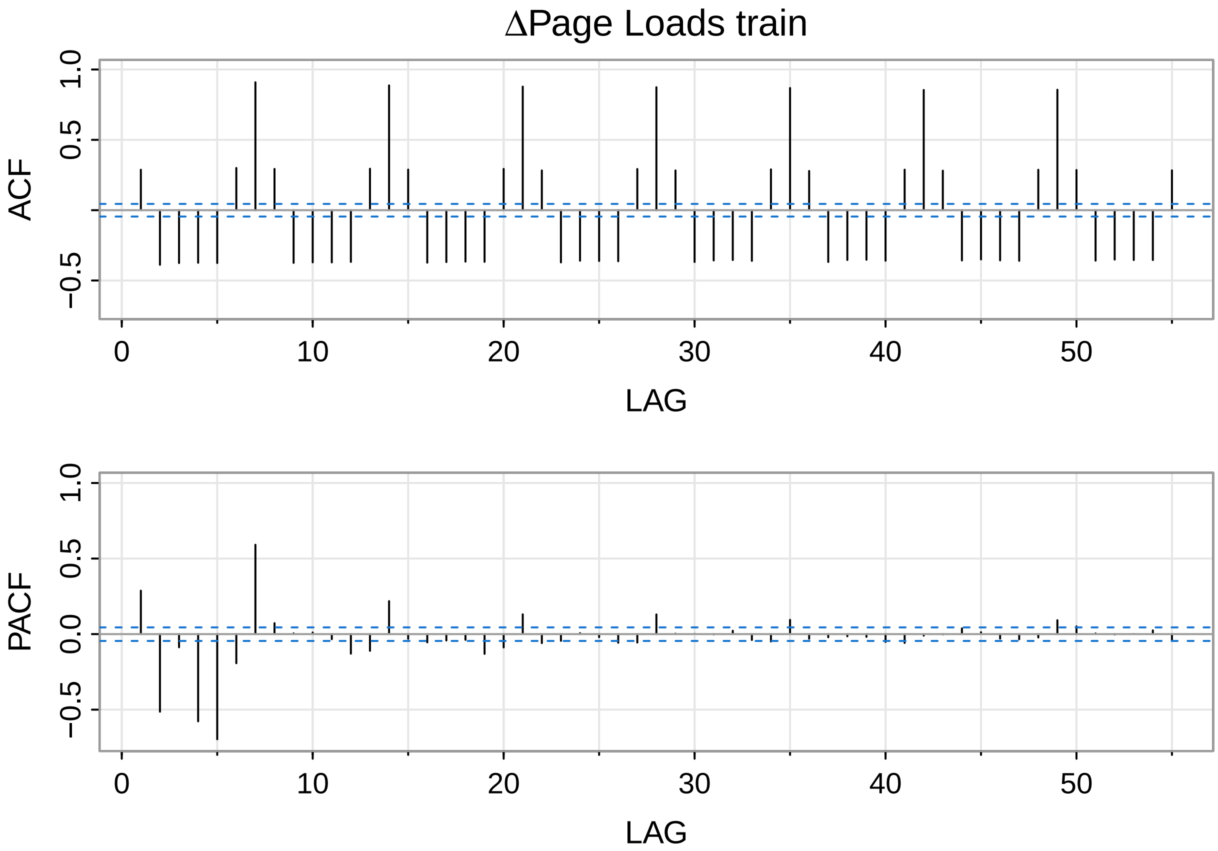

ΔPage Loads train
ACF
1.0
0.5
−0.5
0
10
20
30
40
50
LAG
PACF
1.0
0.5
0.0
−0.5
0
10
20
30
40
50
LAG

Figure 7: ...

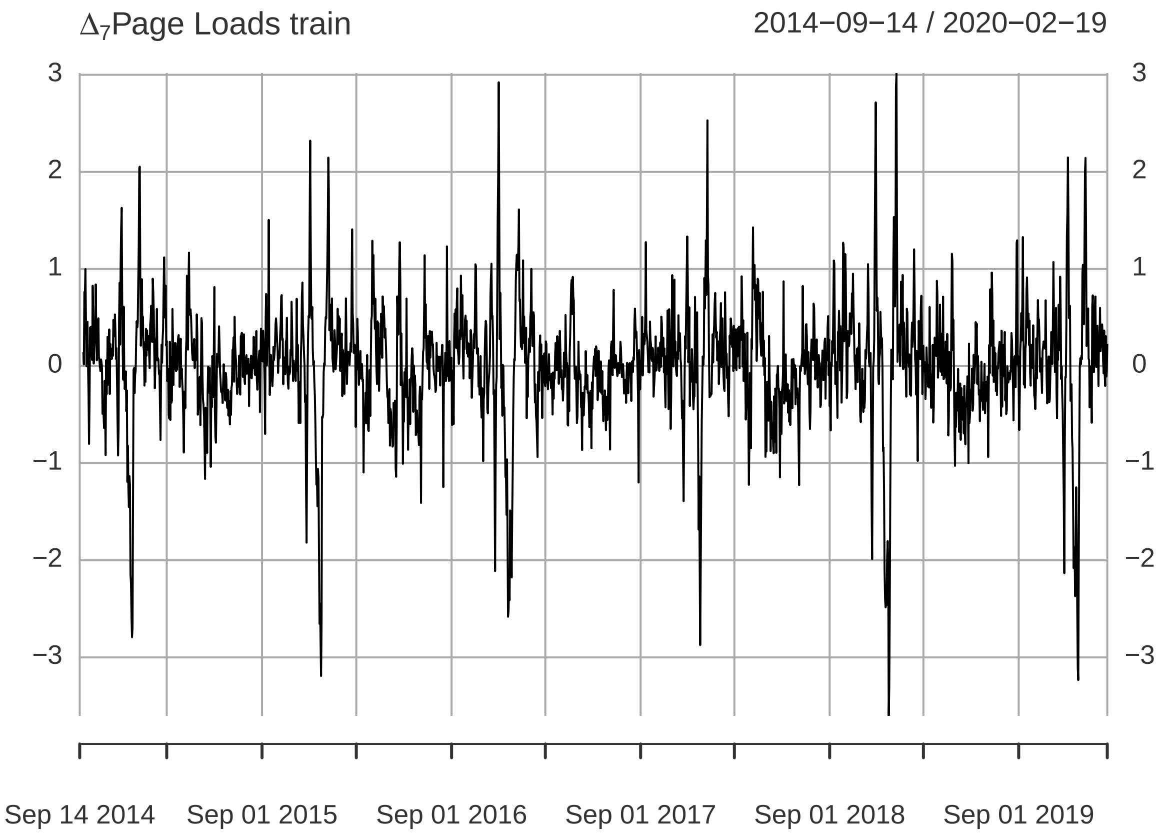

Δ_7Page Loads train
2014−09−14 / 2020−02−19
3
2
1
0
−1
−2
−3
Sep 14 2014
Sep 01 2015
Sep 01 2016
Sep 01 2017
Sep 01 2018
Sep 01 2019

Figure 8: ...

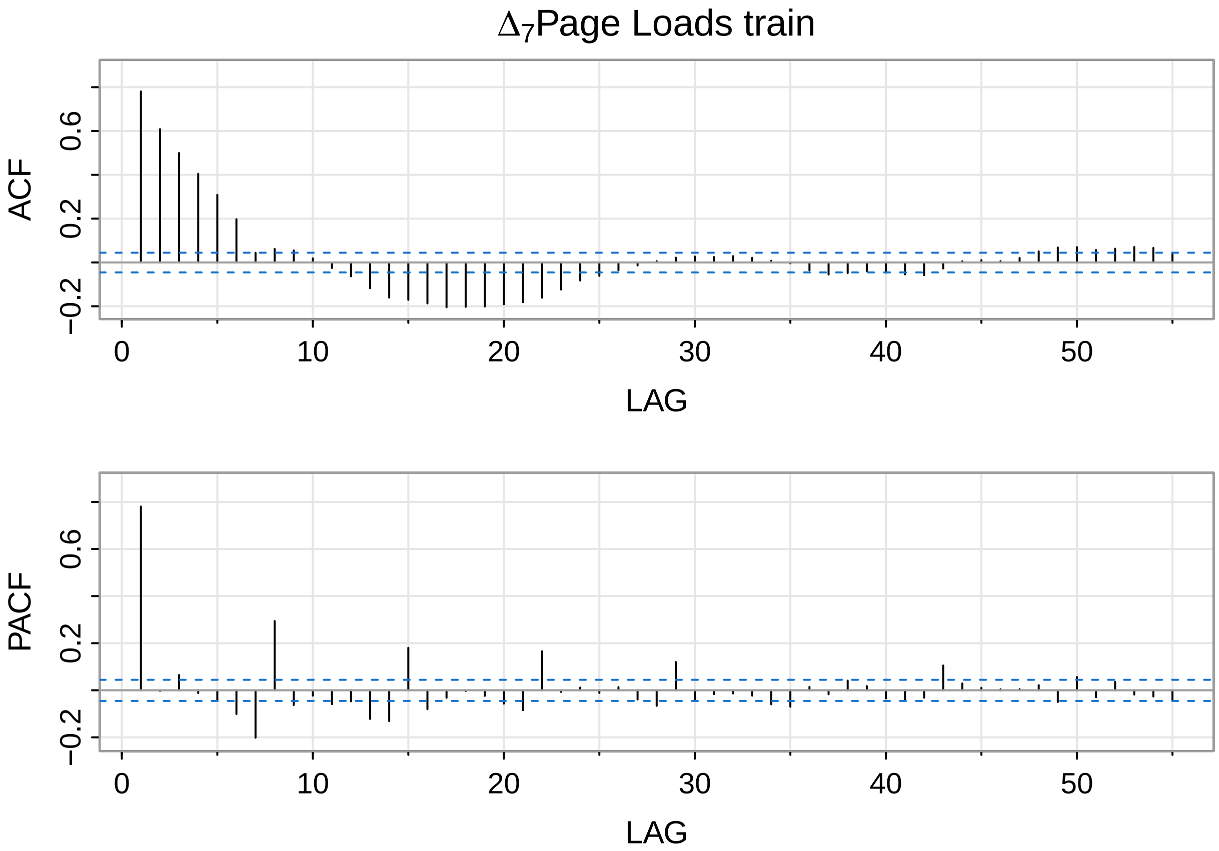

Δ_7Page Loads train
ACF
0.6
0.2
−0.2
0
10
20
30
40
50
LAG
PACF
0.6
0.2
−0.2
0
10
20
30
40
50
LAG

Figure 9: ...

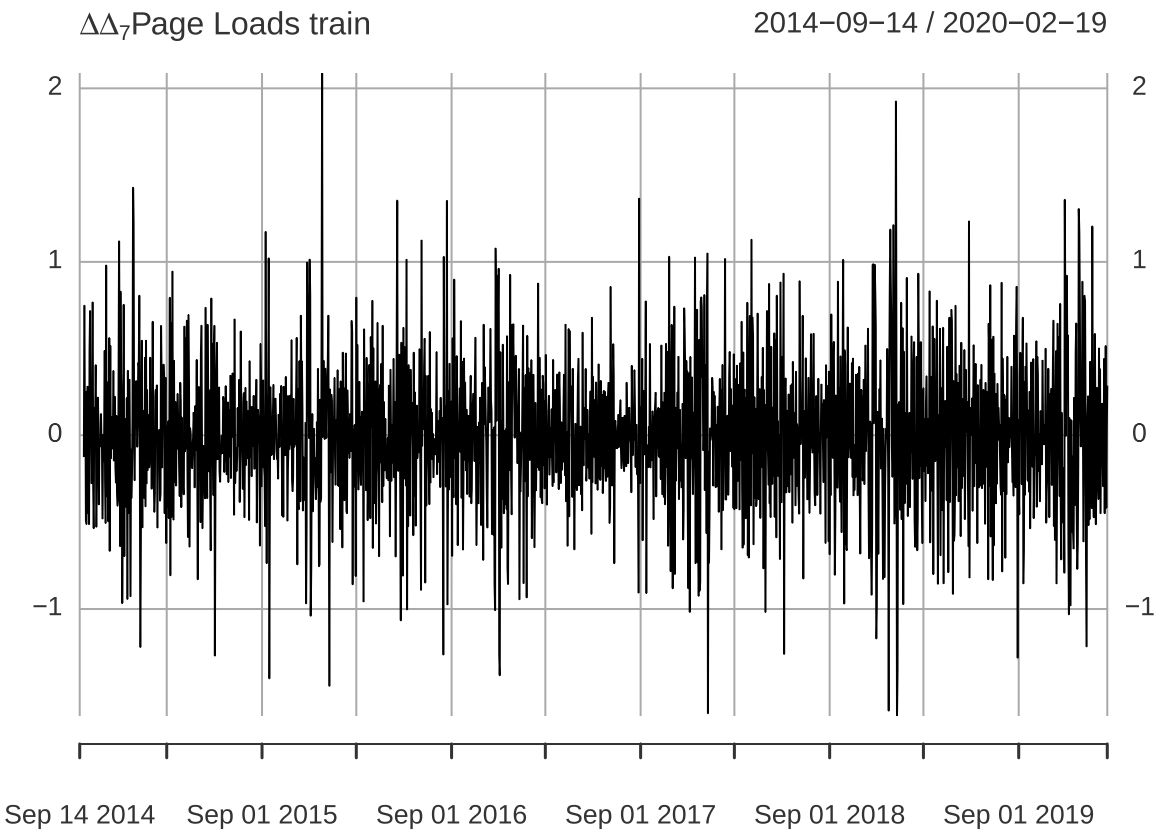

$\Delta\Delta_7$Page Loads train
2014−09−14 / 2020−02−19
2
1
0
−1
Sep 14 2014
Sep 01 2015
Sep 01 2016
Sep 01 2017
Sep 01 2018
Sep 01 2019

Figure 10: ...

```r
invisible(acf2(dlm7.1,main=expression(paste(Delta,Delta[7], "Page Loads train"))))
```

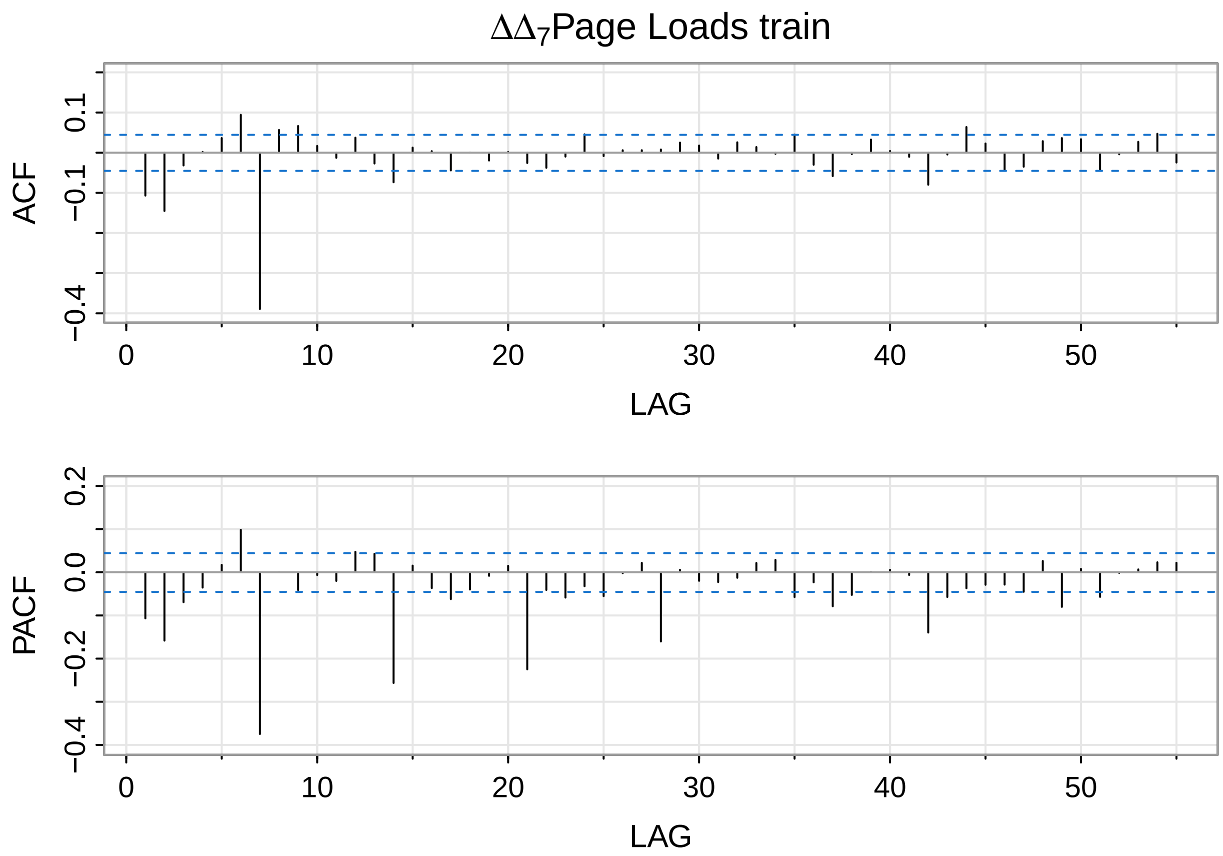


Figure 11: ...

COMMENTS

Modelling strategy

TO DO

Analysis of Weekly Page Loads

Now, we are aggregating our data to weekly time series.

```r
ts_website_weekly <- apply.weekly(ts_website, mean)
plot(ts_website_weekly, main = "Weekly Average Page Loads", ylab = "Page Loads", xlab = "Date")
```

We divide the data into a training set and a test set. The test set will contain the last 6 months of observations.

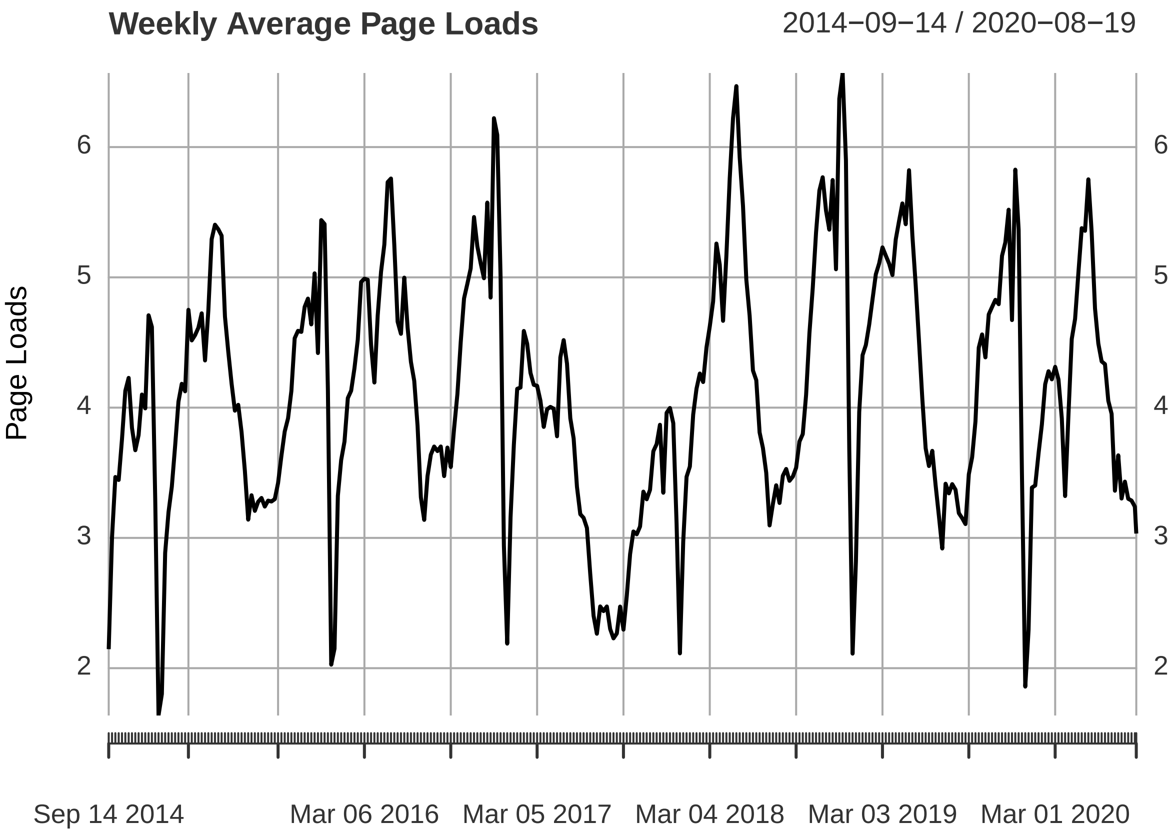


Figure 12: Weekly Page Loads

```
# Training set: First 4.5 years, Test set: Last 6 months
cutoff_date <- as.Date("2020-02-19")
train_data_weekly <- window(ts_website_weekly, end = cutoff_date)
test_data_weekly <- window(ts_website_weekly, start = cutoff_date + 1)

plot(train_data_weekly, main = "Weekly Page Loads", ylab = "Page Loads", xlab = "Date")
```

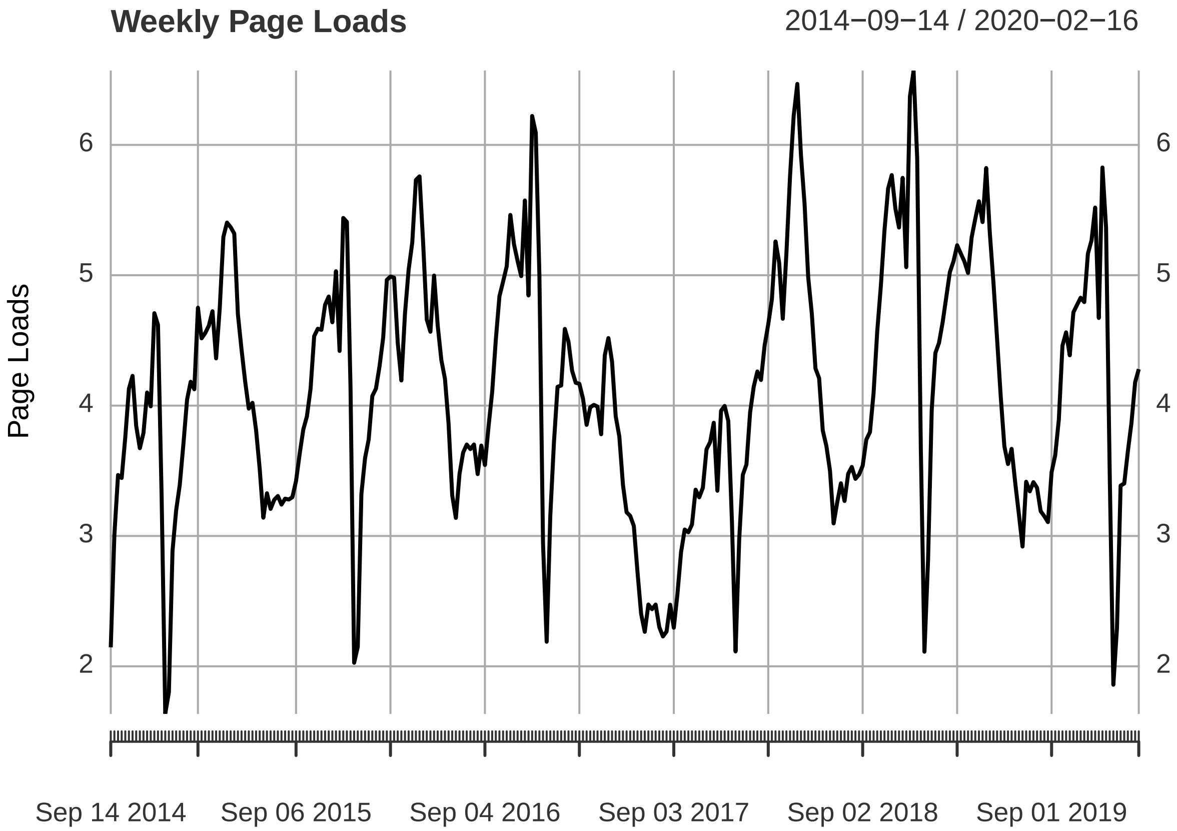


Figure 13: Weekly Page Loads - training set

```
invisible(acf2(ts_website_weekly,max.lag= 80))
```

```
adf.test(ts_website_weekly, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_website_weekly
## Dickey-Fuller = -4.4322, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

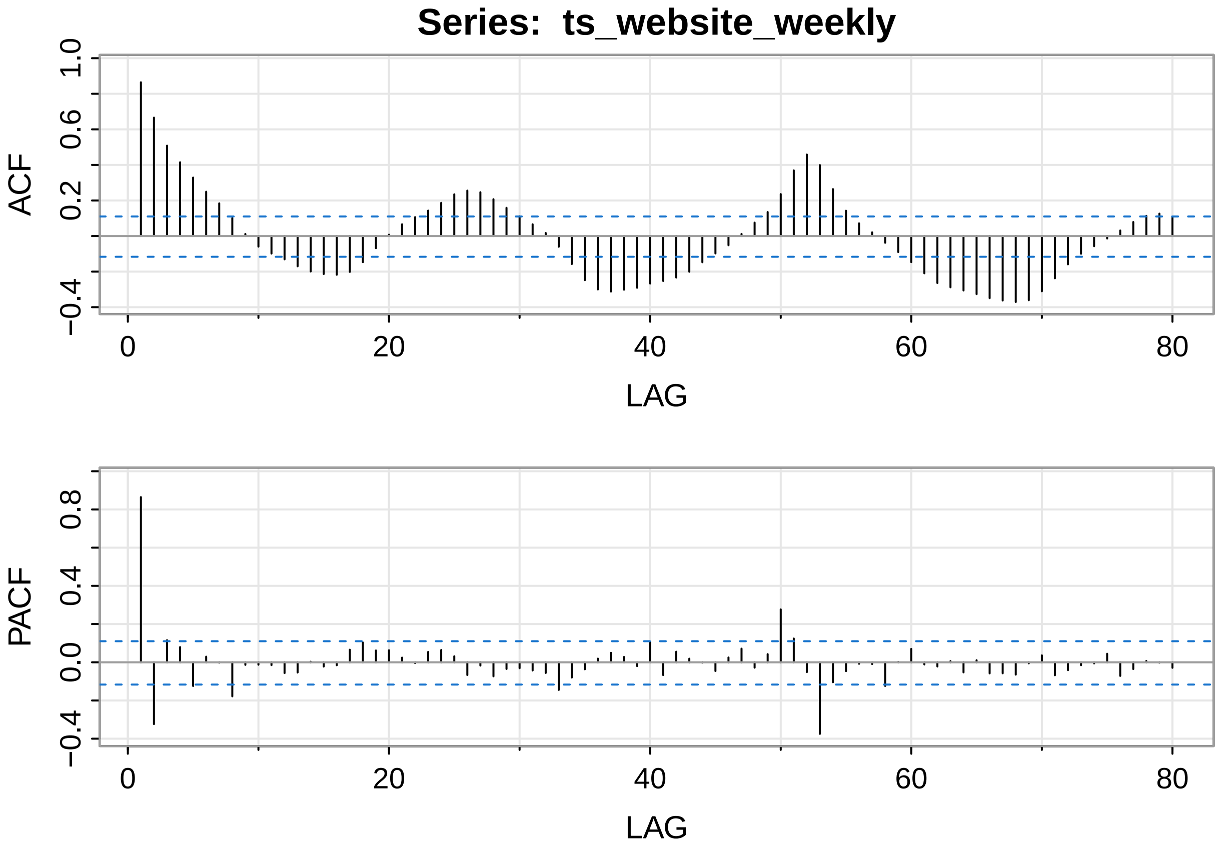
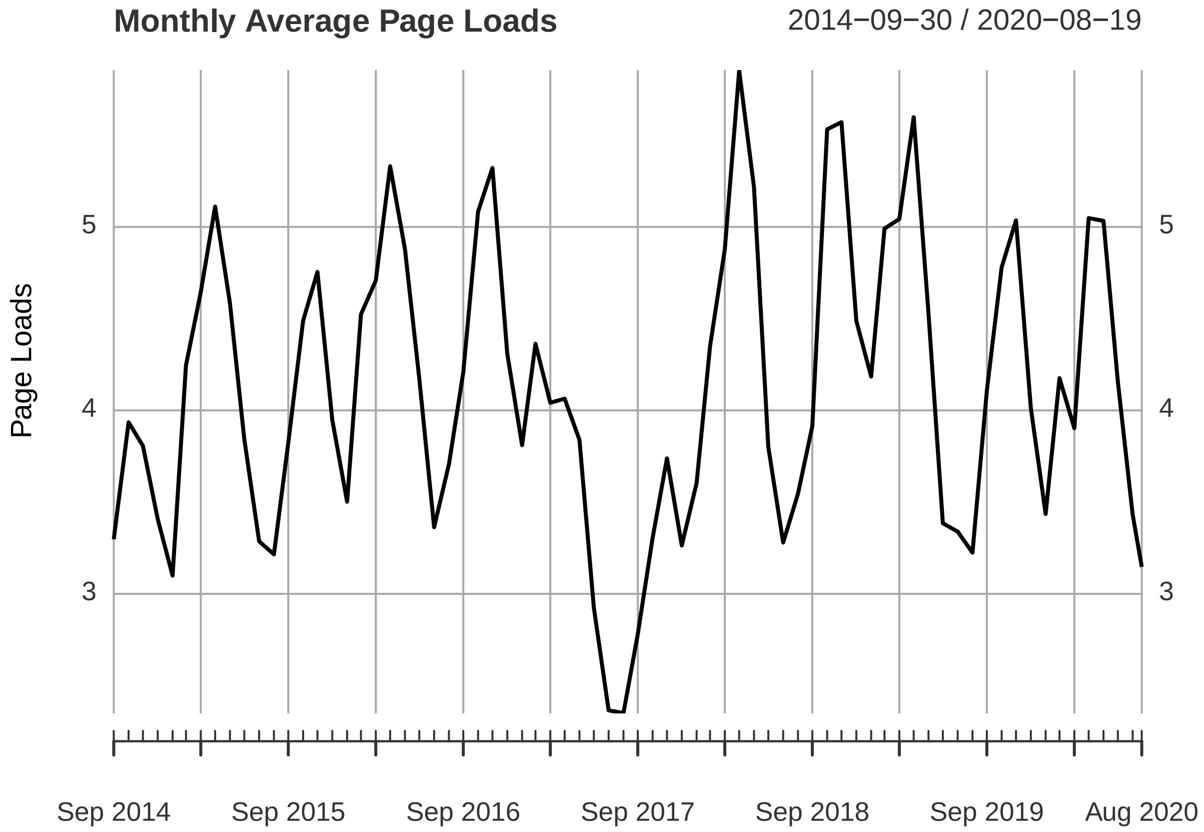


Figure 14: ACF and PACF

Analysis of Monthly Page Loads

```r
# Monthly Aggregation
ts_website_monthly <- apply.monthly(ts_website, mean)
plot(ts_website_monthly, main = "Monthly Average Page Loads", ylab = "Page Loads", xlab = "Date")
```



We divide the data into a training set and a test set. The test set will contain the last 6 months of observations.

```r
# Training set: First 4.5 years, Test set: Last 6 months
cutoff_date <- as.Date("2020-02-29")
train_data_monthly <- window(ts_website_monthly, end = cutoff_date)
test_data_monthly <- window(ts_website_monthly, start = cutoff_date + 1)

plot(train_data_monthly, main = "Weekly Page Loads", ylab = "Page Loads", xlab = "Date")
```

```r
invisible(acf2(ts_website_monthly))
```

```r
adf.test(ts_website_monthly, alternative = "stationary")
```

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

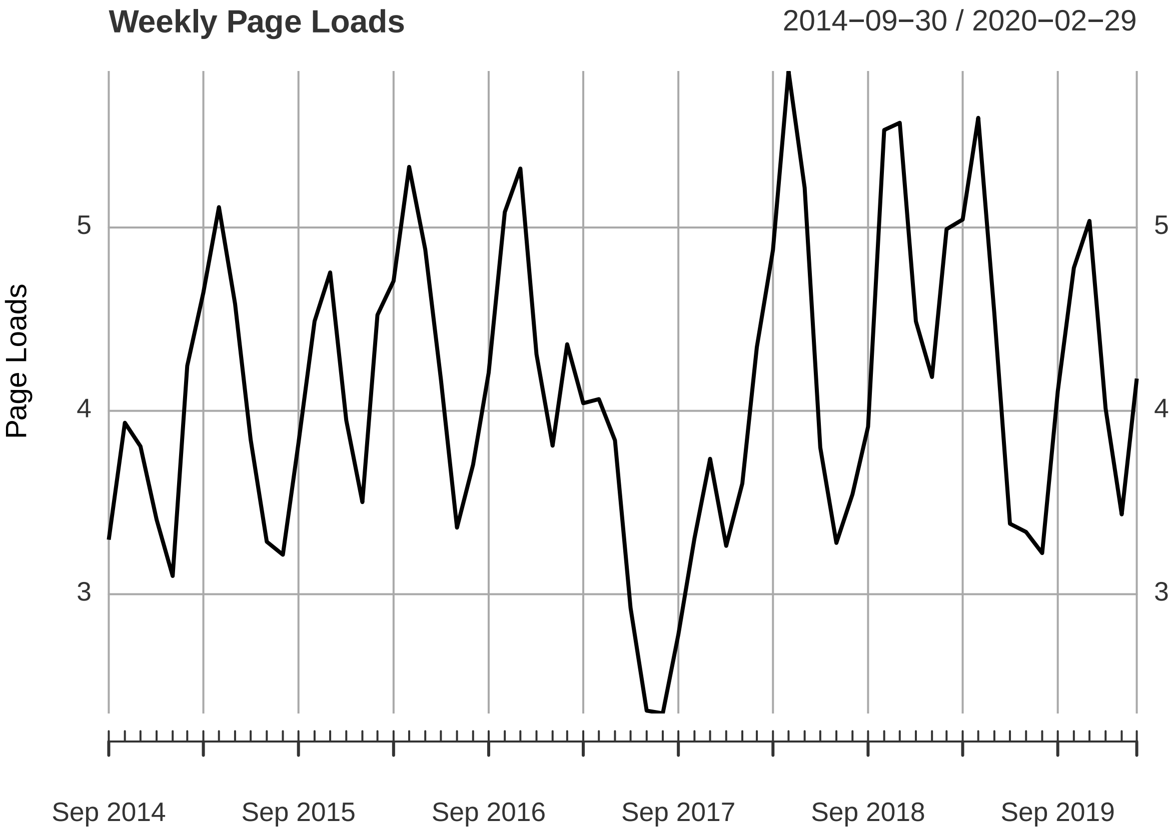


Figure 15: Weekly Page Loads - training set

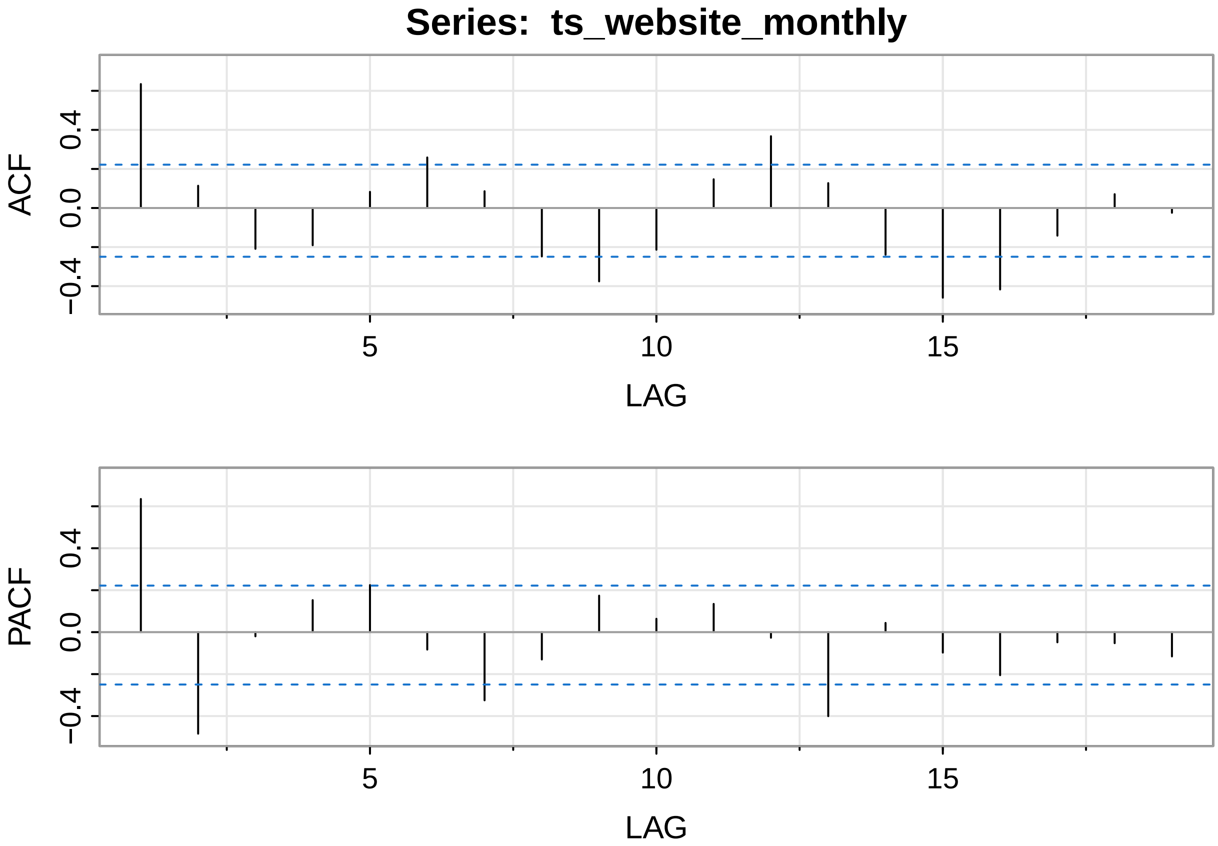


Figure 16: ACF and PACF

```
## data:  ts_website_monthly
## Dickey-Fuller = -2.5767, Lag order = 4, p-value = 0.3405
## alternative hypothesis: stationary
```