To: Business Operations Director

From: Shreya, Urvish, Adi, Salma

Subject: Demand Forecast

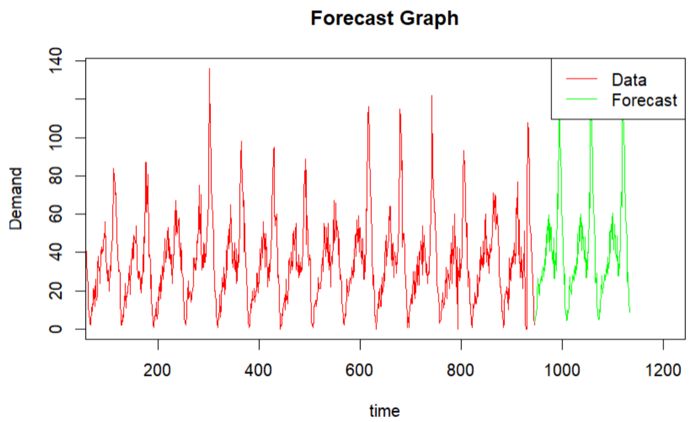
Date: 03/04/2023

Problem Statement: To create a forecast of future demand in planning to acquire new buses and extend its terminals to meet expected demand.

EXECUTIVE SUMMARY

Major Findings

1. Number of passengers arriving at the terminal at the beginning of the week is higher than end of the week.
2. There is a consistent increase in demand every week as the number of passengers at the terminal increase from 100 passenger on 1st march to 120 passengers on 21st march with over 160 passengers on 8th March.
3. Distribution of demand is right skewed which means that majority of demand falls in certain range with low instances of high demand. This high demand can also be seasonal, which occurs during the first few days of the week and low demand for rest of the week.
4. Highest number of passengers arrive between 5:30 PM and 8:00 PM at the terminals.
5. The highest average number of passengers arrive at 6 PM followed by 7 PM and 1 PM. The least average number of passengers arrive at 6 AM.
6. The highest average demand for transportation is on Mondays followed by other working days, while least demand is on Sundays.



7. High autocorrelation at the lags indicate that data is nonstationary and needs to be addressed.
8. The forecasted data indicates that the demand continues the same pattern and numbers until 24th March 2005 from double seasonality.

Recommendations for Action

1. The number of buses should be increased during weekdays especially during the busiest times of the days at 6PM,7Pm and 8PM.
2. The number of buses can be decreased during weekends and can be utilized during busy hours to reduce cost.
3. The performance of the resources is to be tracked to make sure buses and terminals are well equipped for the changing demand.
4. Develop a contingency plan to support sudden spikes in the passengers arriving at the terminals.

Analytical Overview

The dataset has 990 rows observations and 3 variables with date, time and demand that shows instances of passengers every 15 minutes from 6 am to 10 pm every day. The date columns are converted into date time and calculating differences between dates.

The dataset is visualized for exploratory data analysis through distribution of data, demand over time, demand by day, demand by hours of week, demand by hour of day etc.

ACF plot is implemented for the dataset. The resulting plot shows the strong correlation between the time series and its lagged values. The plot has the lag and correlation coefficient. The plot has two horizontal lines, representing the 95% and 99% confidence intervals for the correlation coefficients. Any correlation coefficients that fall outside of these confidence intervals are considered statistically significant.

The time series is later decomposed into components depending on season, level and trend. The seasonal component is assumed to be periodic with a fixed period. The dataset is later divided into training and validation for 63 observations for 7 days. We get total observations as 63 because of instances being recorded veery 15 minutes from 6 am to 10 pm.

The regression model, Holt Winter's, ARIMA and SARIMA models are applied to the dataset to get the best results. Ensemble model takes the average of ARIMA and SARIMA methods to give the best average predictions possible. The next approach was to divide the model into weekdays and weekends, through double seasonality to achieve the best accuracy.

Documentation

Exploratory Data Analysis:

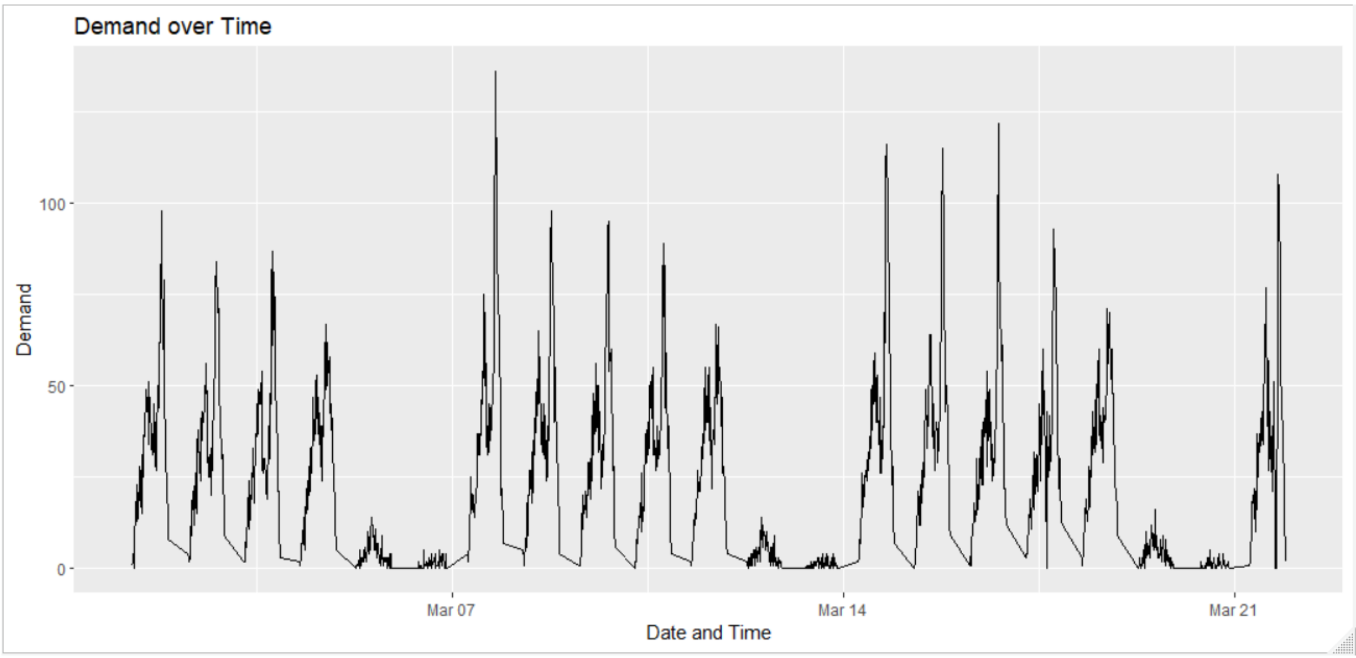
Getting one column for date time & summarising it.

```
library(readxl)
library(forecast)
library(ggplot2)
library(tidyverse)
library(lubridate)
bicup.data<-read_excel("bicup2006.xls")

data = data.frame(bicup.data)
data$DATE=as.Date(data$DATE)
data$TIME=format(data$TIME, format = "%H:%M")
data$DATETIME <- as.POSIXct(paste(data$DATE, data$TIME), format = "%Y-%m-%d %H:%M")

summary(data)

```

```
> summary(data)
      DATE                TIME               DEMAND           DATETIME
 Min.   :2005-03-01   Length:1323        Min.   :  0.00   Min.   :2005-03-01 06:30:00
 1st Qu.:2005-03-06   Class :character   1st Qu.:  4.00   1st Qu.:2005-03-06 10:22:30
 Median :2005-03-11   Mode  :character   Median : 23.00   Median :2005-03-11 14:15:00
 Mean   :2005-03-11                      Mean   : 25.87   Mean   :2005-03-11 14:15:00
 3rd Qu.:2005-03-16                      3rd Qu.: 40.00   3rd Qu.:2005-03-16 18:07:30
 Max.   :2005-03-21                      Max.   :136.00   Max.   :2005-03-21 22:00:00
> 
```

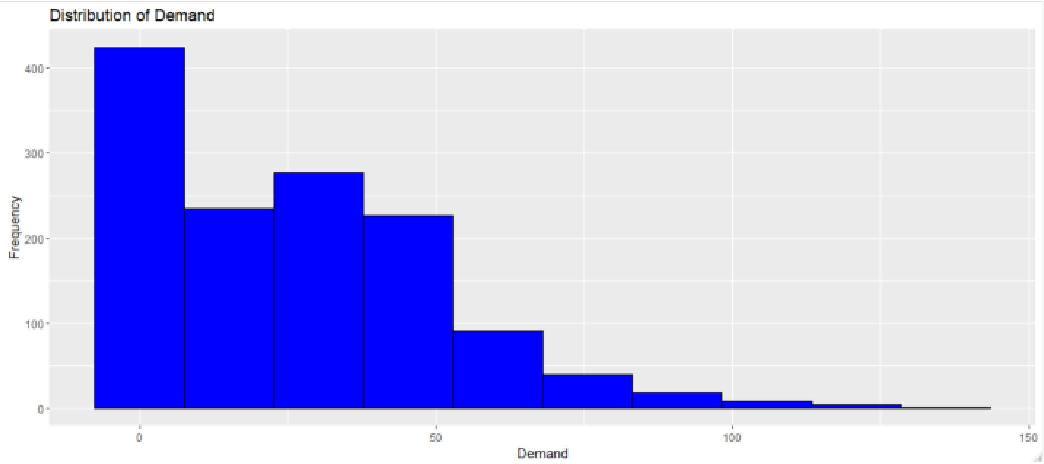
Visualizing Demand over time:

```
# Visualize the demand over time
ggplot(data, aes(x = DATETIME, y = DEMAND)) +
  geom_line() +
  labs(title = "Demand over Time", x = "Date and Time", y = "Demand")

```



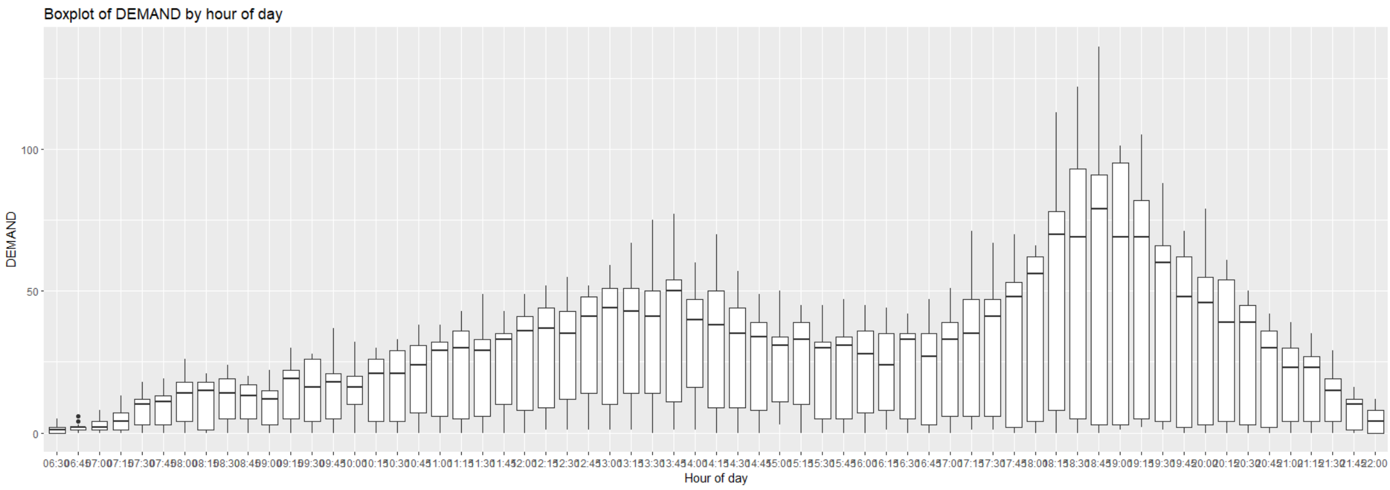
Plotting the distribution of demand.

```r
# plot the distribution of demand
ggplot(data, aes(x = DEMAND)) +
  geom_histogram(bins = 10, fill = "blue", color = "black") +
  labs(title = "Distribution of Demand", x = "Demand", y = "Frequency")

```



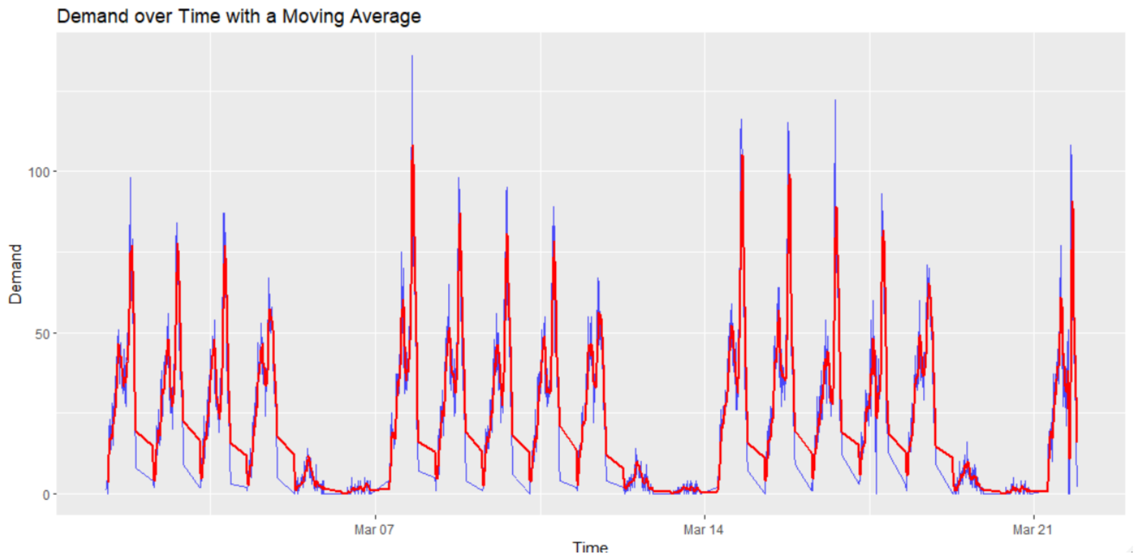
Box plot of demand for each hour.

```r
data$HOUR <-(format(data$TIME,format =  "%H"))
ggplot(data, aes(x = HOUR, y = DEMAND)) +
  geom_boxplot() +
  labs(title = "Boxplot of DEMAND by hour of day", x = "Hour of day", y = "DEMAND")

```



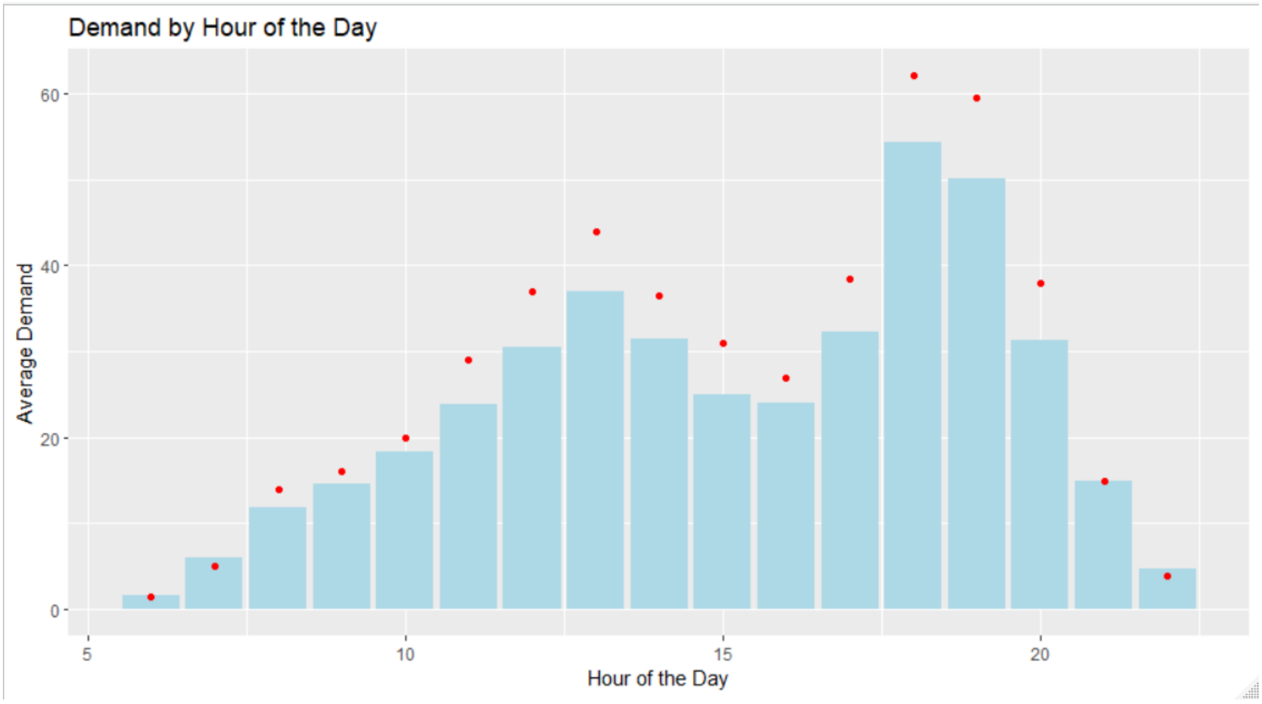
Moving average of Demand over time

```r
data0 <- data %>%
  mutate(MOV_AVG = zoo::rollmeanr(DEMAND, k = 5, fill = NA))

ggplot(data0, aes(x = DATETIME)) +
  geom_line(aes(y = DEMAND), color = "blue", alpha = 0.7) +
  geom_line(aes(y = MOV_AVG), color = "red", size = 1) +
  labs(title = "Demand over Time with a Moving Average", x = "Time", y = "Demand")

```



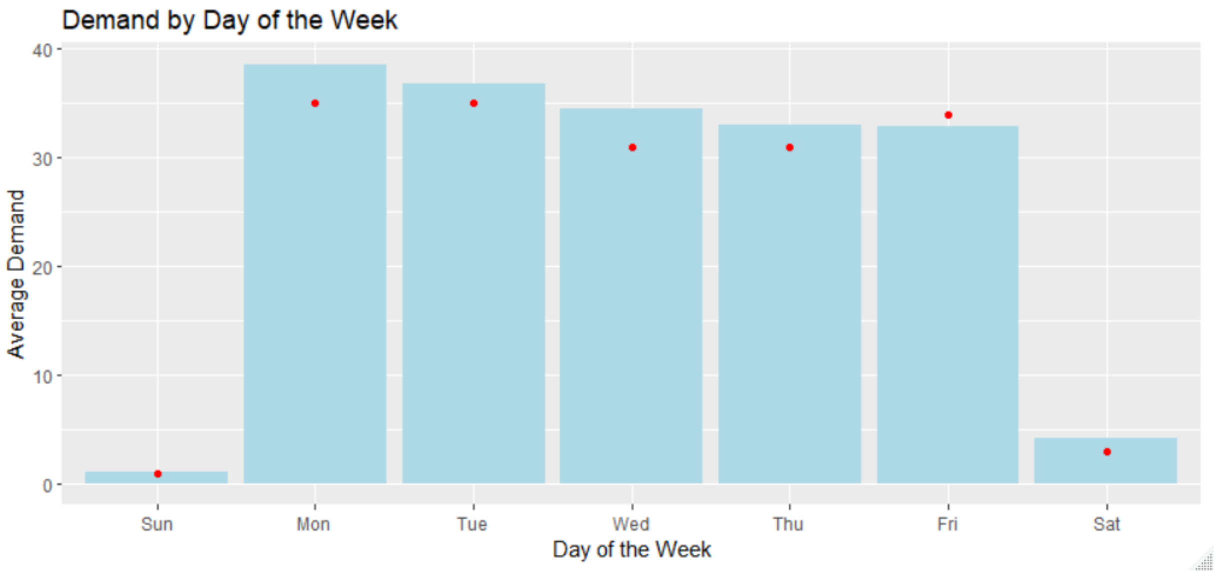
Average Demand by an hour of the day.

```
# Plot demand by hour of the day
data1 <- data %>%
  mutate(HOUR = hour(DATETIME)) %>%
  group_by(HOUR) %>%
  summarise(AVG_DEMAND = mean(DEMAND), MED_DEMAND = median(DEMAND))

ggplot(data1, aes(x = HOUR, y = AVG_DEMAND)) +
  geom_col(fill = "lightblue") +
  geom_point(aes(y = MED_DEMAND), color = "red") +
  labs(title = "Demand by Hour of the Day", x = "Hour of the Day", y = "Average Demand")
```



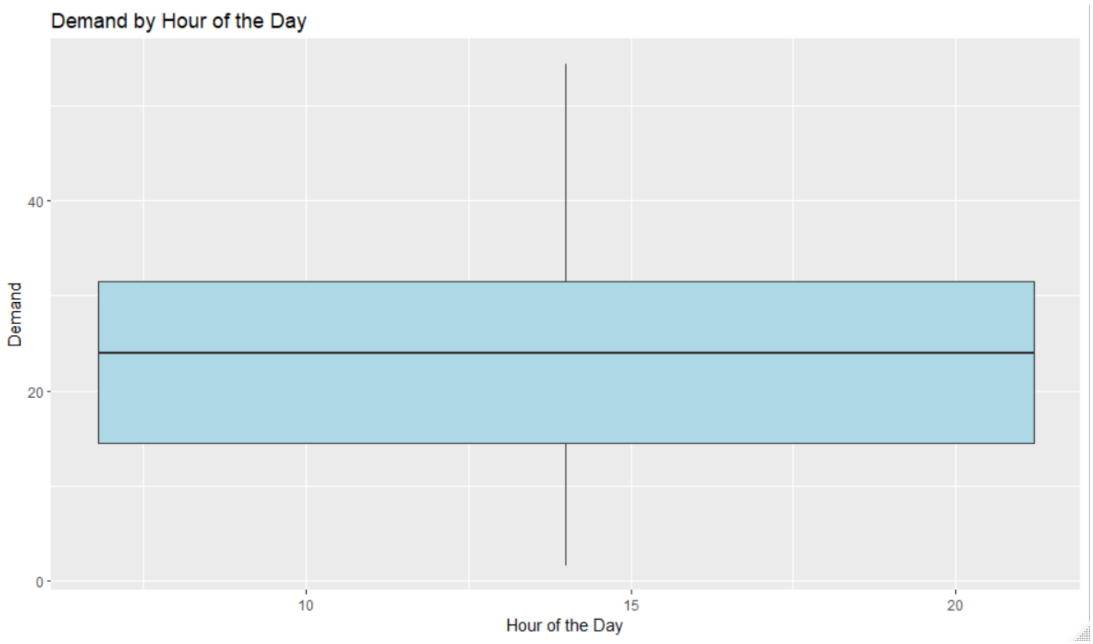
Average Demand by day of week

```
# Plot demand by day of the week
data2 <- data %>%
  mutate(DAY = wday(DATETIME, label = TRUE)) %>%
  group_by(DAY) %>%
  summarise(AVG_DEMAND = mean(DEMAND), MED_DEMAND = median(DEMAND))

ggplot(data2, aes(x = DAY, y = AVG_DEMAND)) +
  geom_col(fill = "lightblue") +
  geom_point(aes(y = MED_DEMAND), color = "red") +
  labs(title = "Demand by Day of the Week", x = "Day of the Week", y = "Average Demand")
```



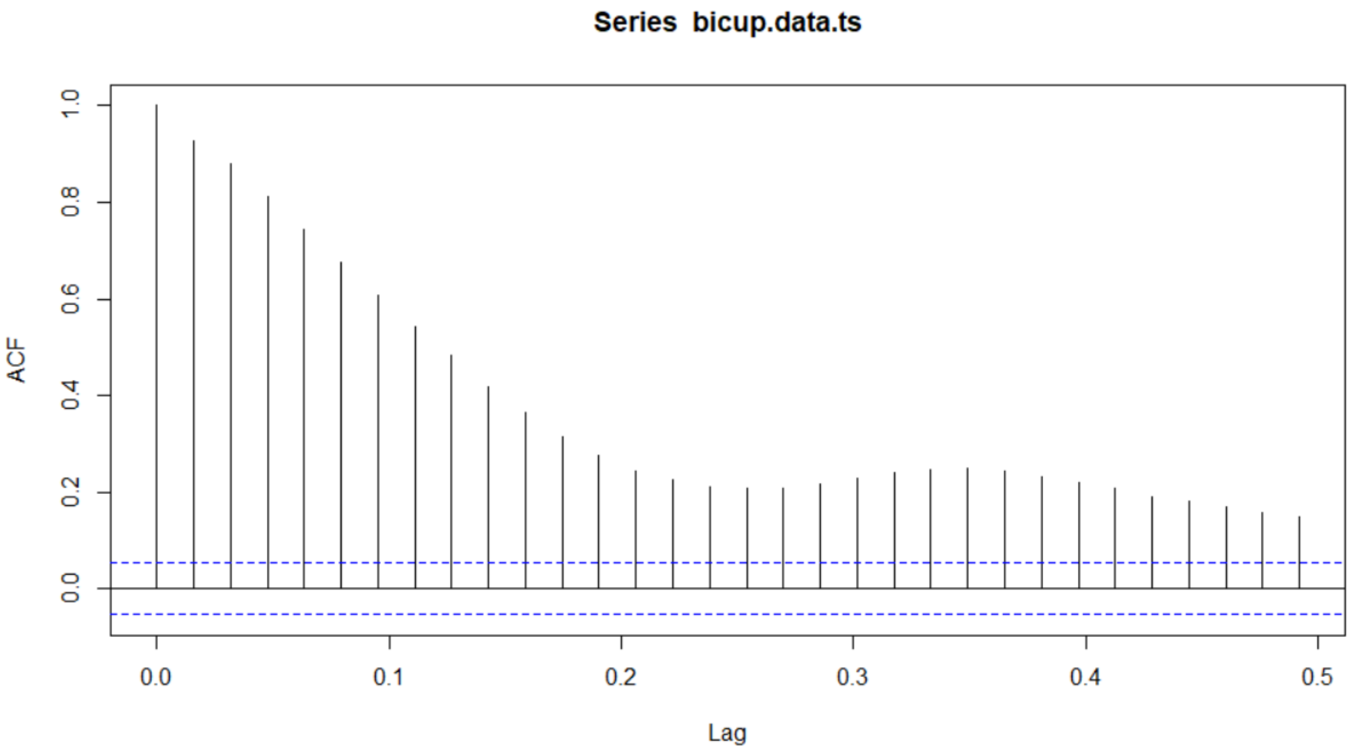
Boxplot of average demand

```
# Plot a boxplot of demand by hour of the day
ggplot(data1, aes(x = HOUR, y = AVG_DEMAND )) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Demand by Hour of the Day", x = "Hour of the Day", y = "Demand")
```



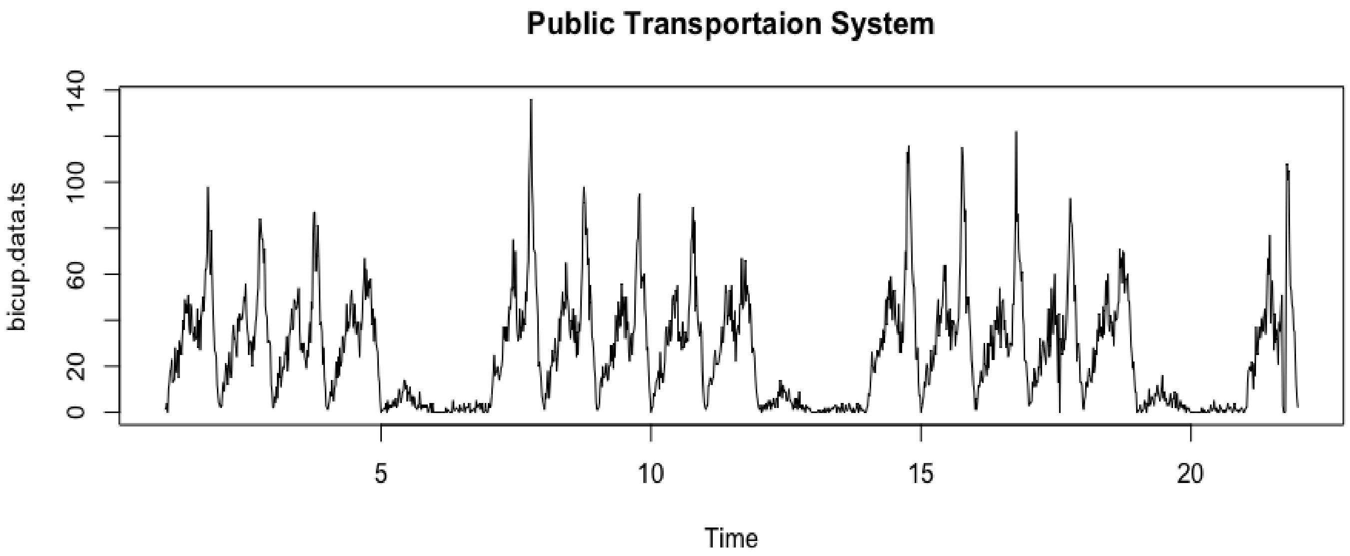
Autocorrelation in Demand

```
bicup.data<-read_excel("bicup2006.xls")
bicup.data.ts <- ts(bicup.data$DEMAND, start= c(2005, 3, 1, 6, 30), frequency = 63)

acf(bicup.data.ts) #strong autocorrelation present
```



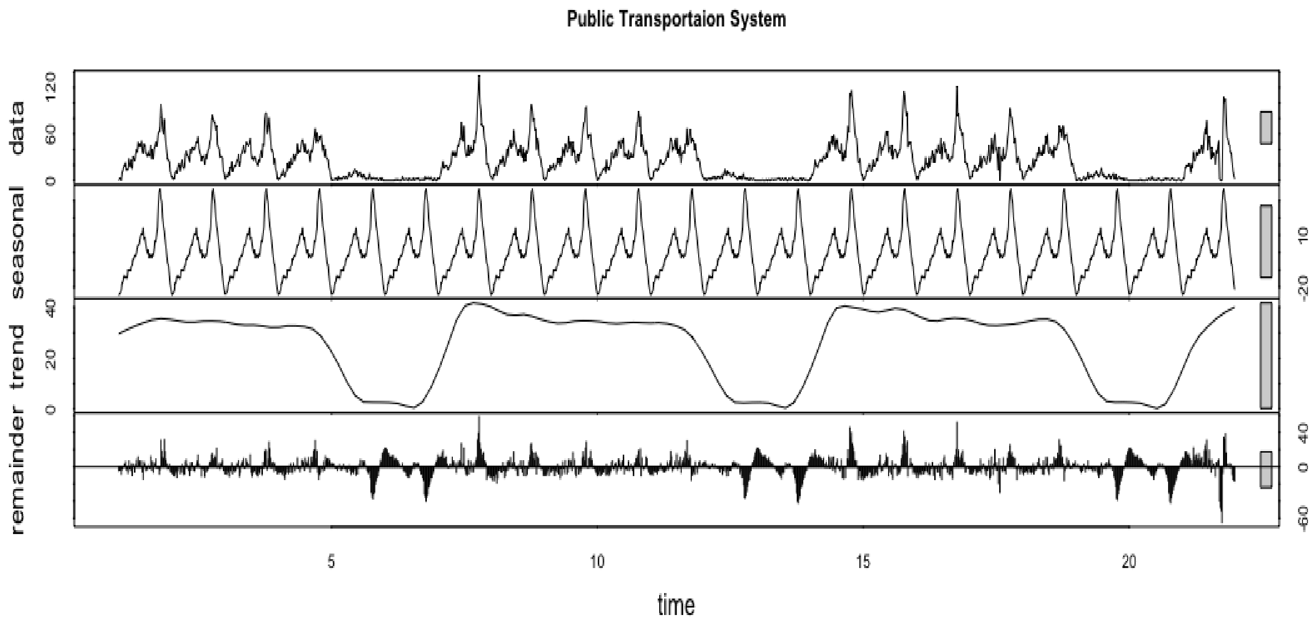
Time series plot:

plot(bicup.data.ts, main = "Public Transportation System",)



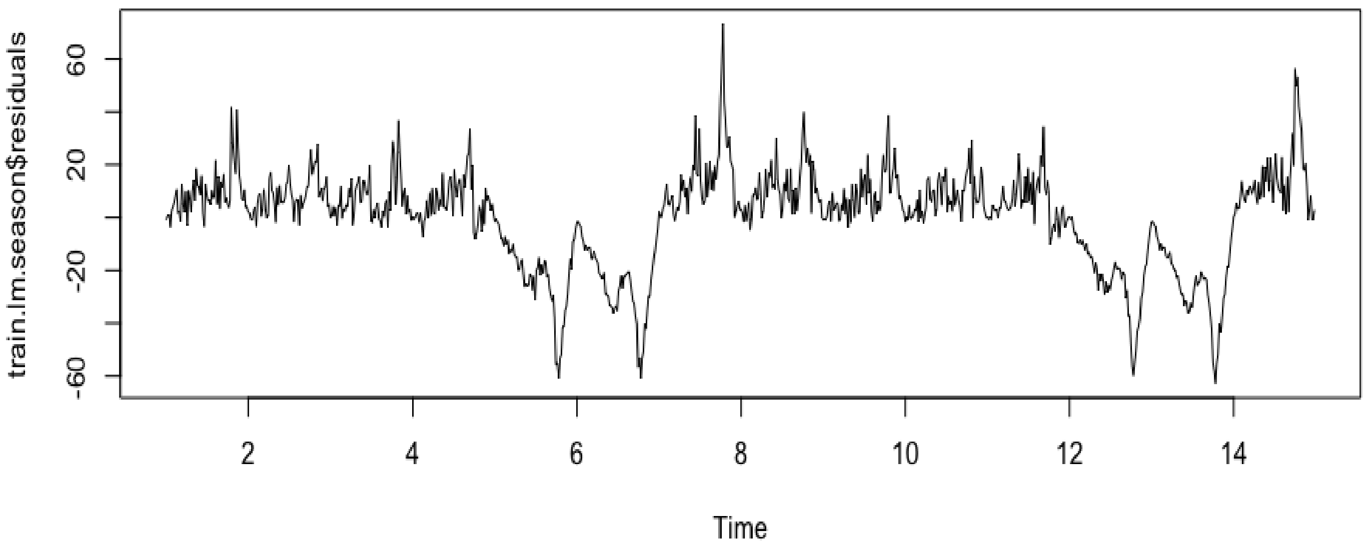
Trend, seasonal & level plot:

```
#trend, season and level present
plot(stl(bicup.data.ts, s.window = "periodic"), main = "Public Transportaion System")
```



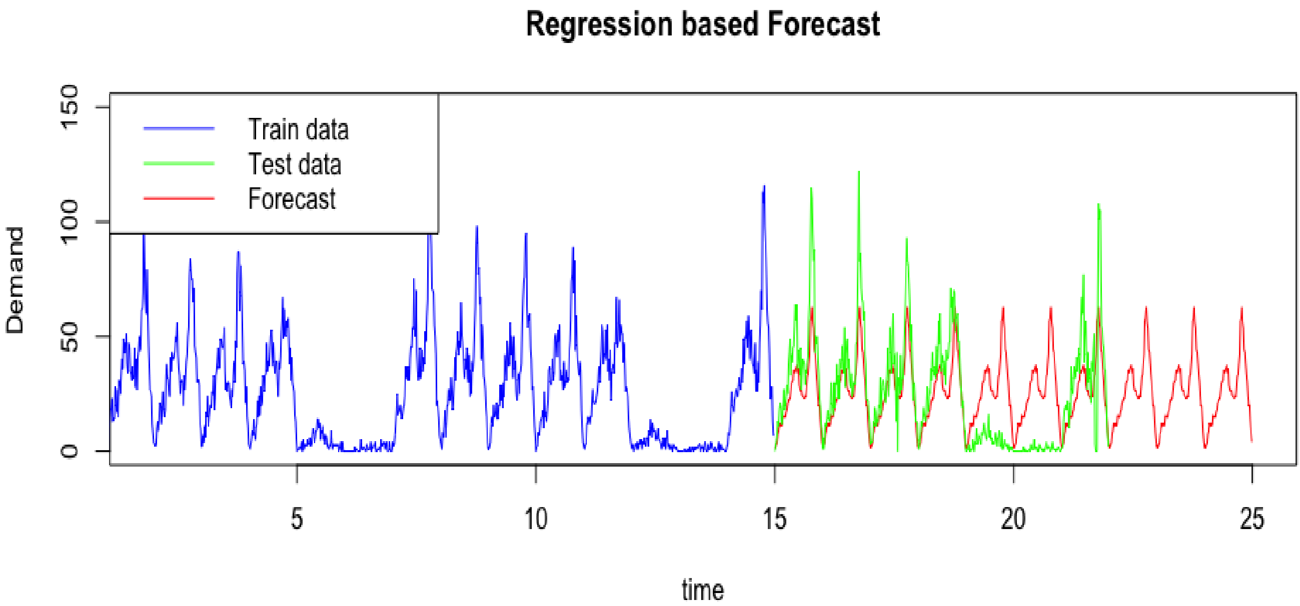
Residual plot of seasonal model:

```
#partition the data
nValid = 63*7
nTraining = length(bicup.data.ts)-nValid
bicup.train.ts = window(bicup.data.ts, end = c(1,nTraining))
bicup.valid.ts = window(bicup.data.ts, start = c(1,nTraining+1))

#Regression based model
train.lm.season = tslm(bicup.train.ts~season)
train.trend.season.pred = forecast(train.lm.season, h=63*7)

plot(train.lm.season$residuals)
```



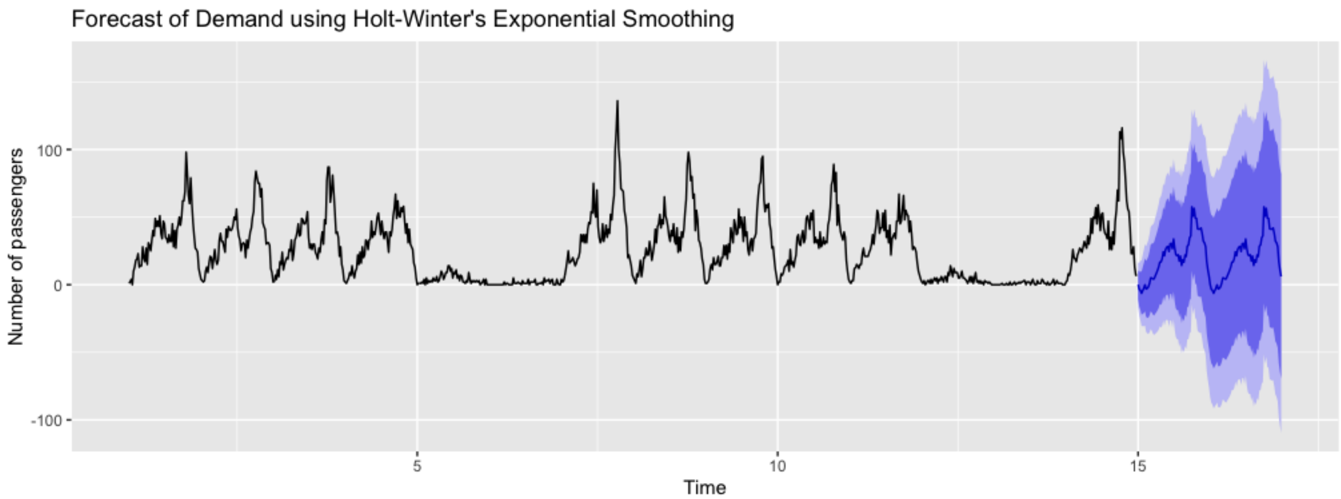
Forecasted graph for seasonal model

```
model1=accuracy(train.lm.season,bicup.valid.ts)

train.lm.season.forecast=forecast(train.lm.season, h=63*10)

plot(train.lm.season.forecast$mean, main = "Regression based Forecast", xlim= c(2,25),
     xlab = "time", ylab = "Demand", col= "red")
lines(bicup.train.ts, col = "blue")
lines(bicup.valid.ts, col = "green")
legend("topleft", legend = c("Train data", "Test data", "Forecast"),
       lty = c(1, 1, 2), col = c("blue", "green", "red"))
```



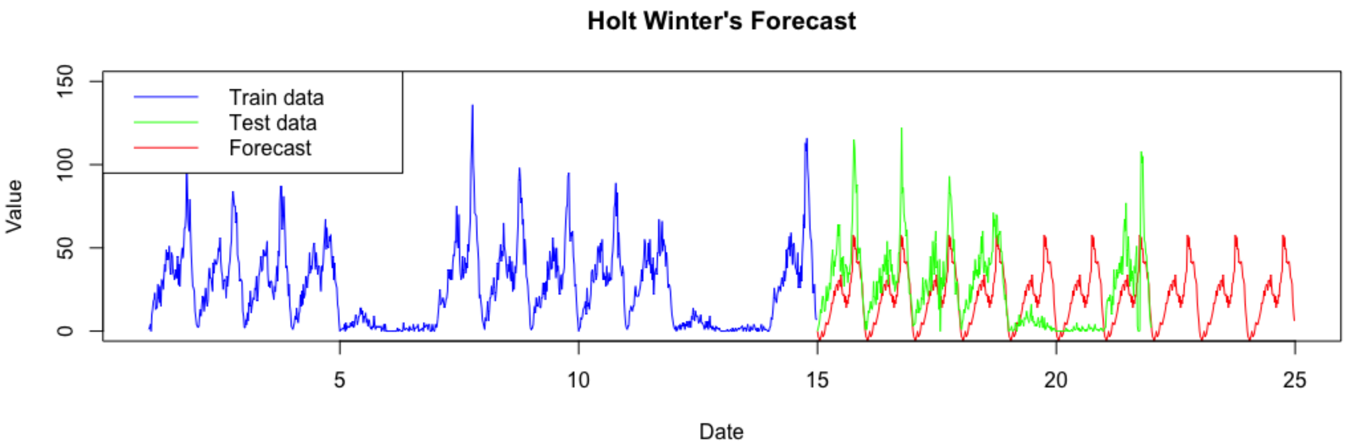
Holt-Winters model forecast.

```
#holt winter's
bicup.data.hw = HoltWinters(bicup.train.ts, beta=FALSE)
autoplot(forecast(bicup.data.hw)) + xlab("Time") + ylab("Number of passengers") +
  ggtitle("Forecast of Demand using Holt-Winter's Exponential Smoothing")
```



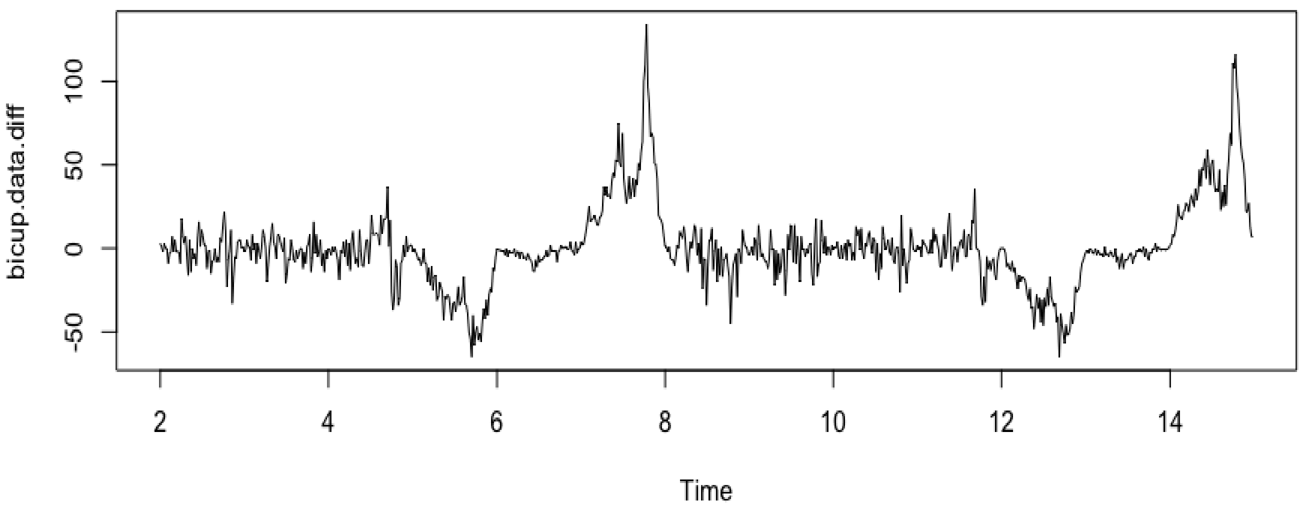
Holt-Winters model forecast.

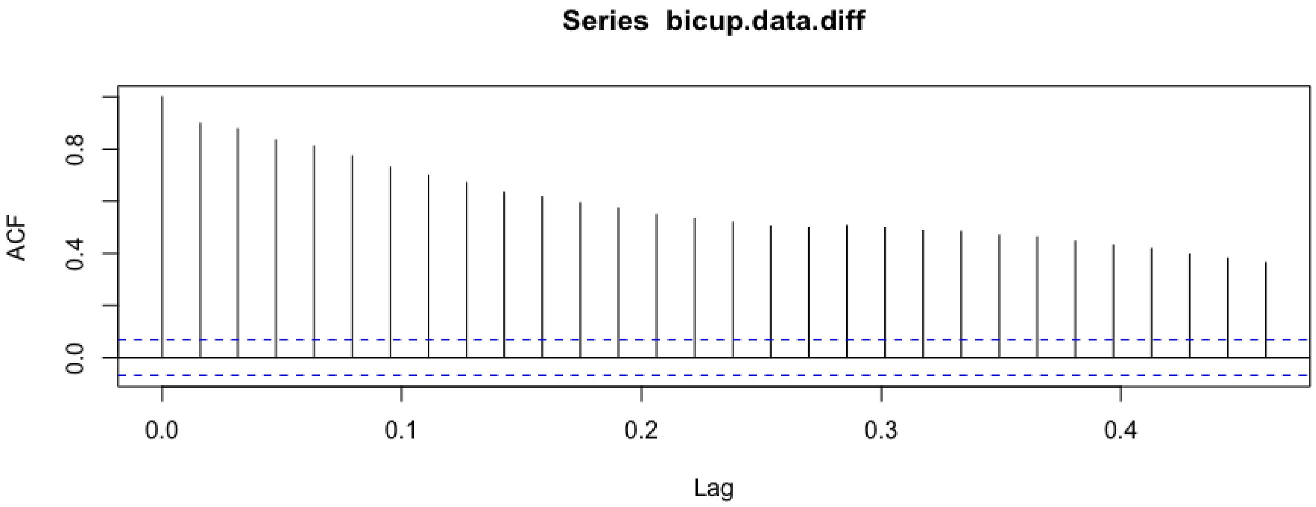
```
bicup.data.hw.forecast1 = forecast(bicup.data.hw,h=63*10)
plot(bicup.data.hw.forecast1$mean, main = "Holt Winter's Forecast", xlim= c(1,25)
     , ylim=c(0,150), xlab = "Date", ylab = "Value", col= "red")
lines(bicup.train.ts, col = "blue")
lines(bicup.valid.ts, col = "green")
legend("topleft", legend = c("Train data", "Test data", "Forecast"),
       lty = c(1, 1, 1), col = c("blue", "green", "red"))
```

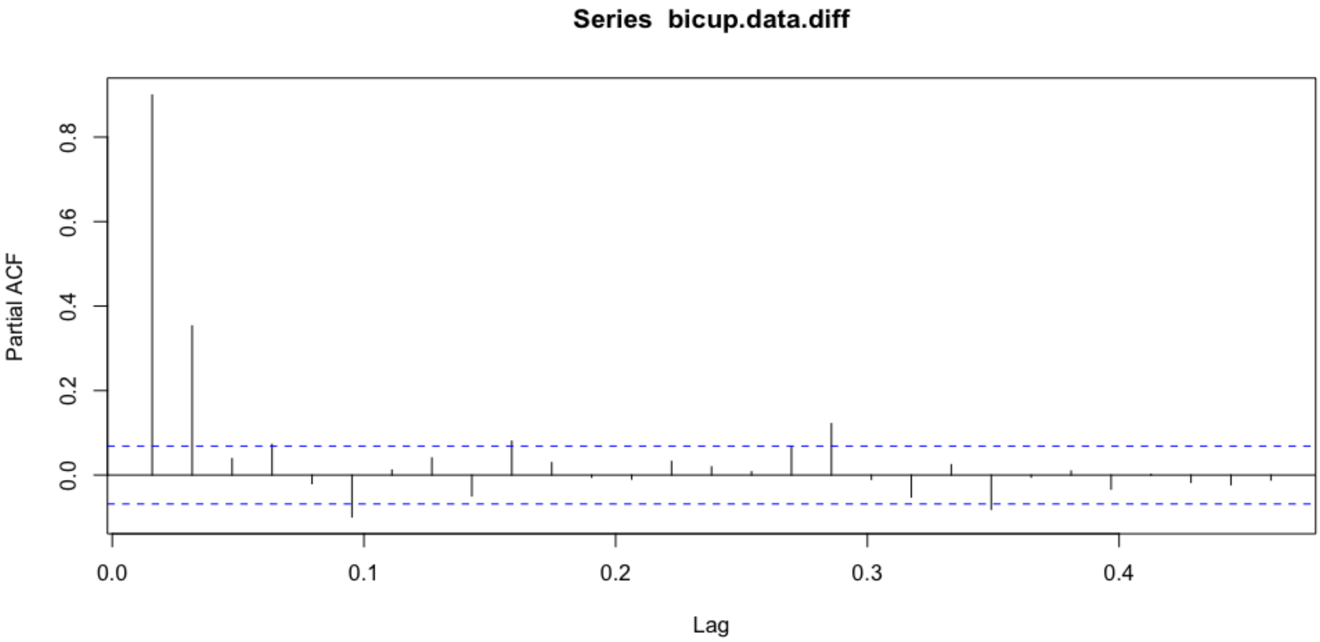


Plots of Arima: differenced, ACF, pacf.

```
#ARIMA model
bicup.data.diff=diff(bicup.train.ts,lag=63)

plot(bicup.data.diff)

acf(bicup.data.diff)
pacf(bicup.data.diff)
```



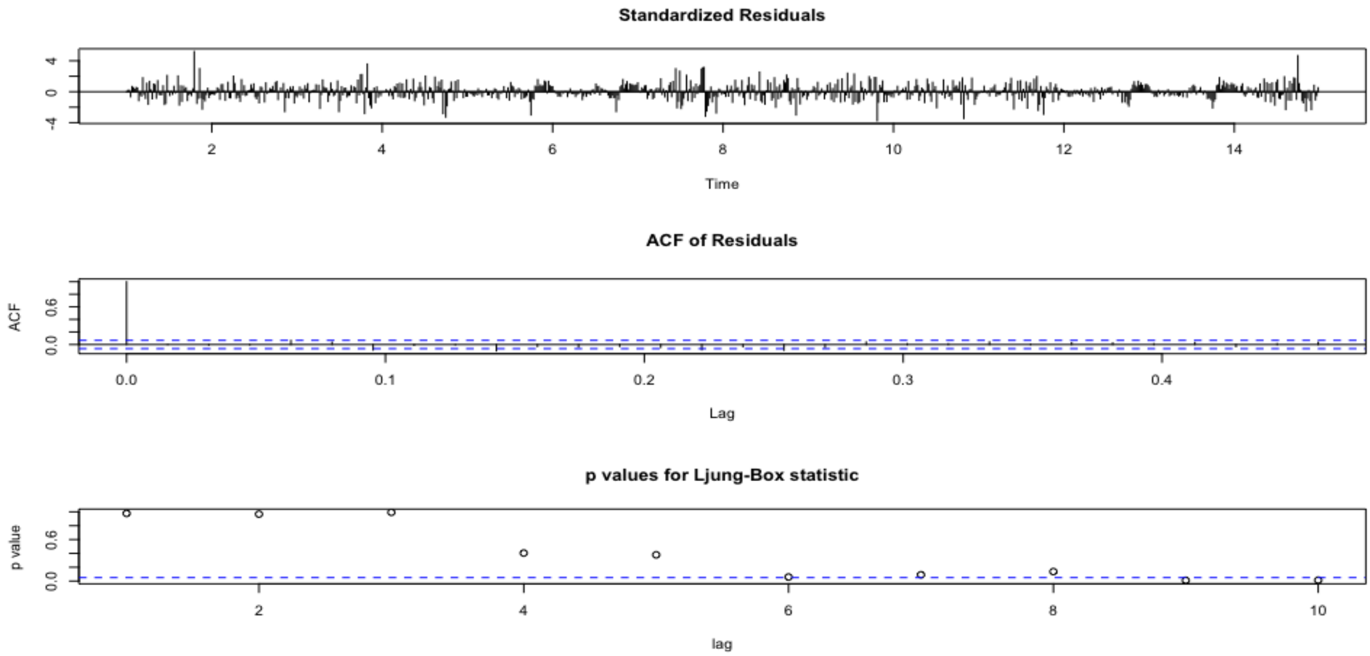




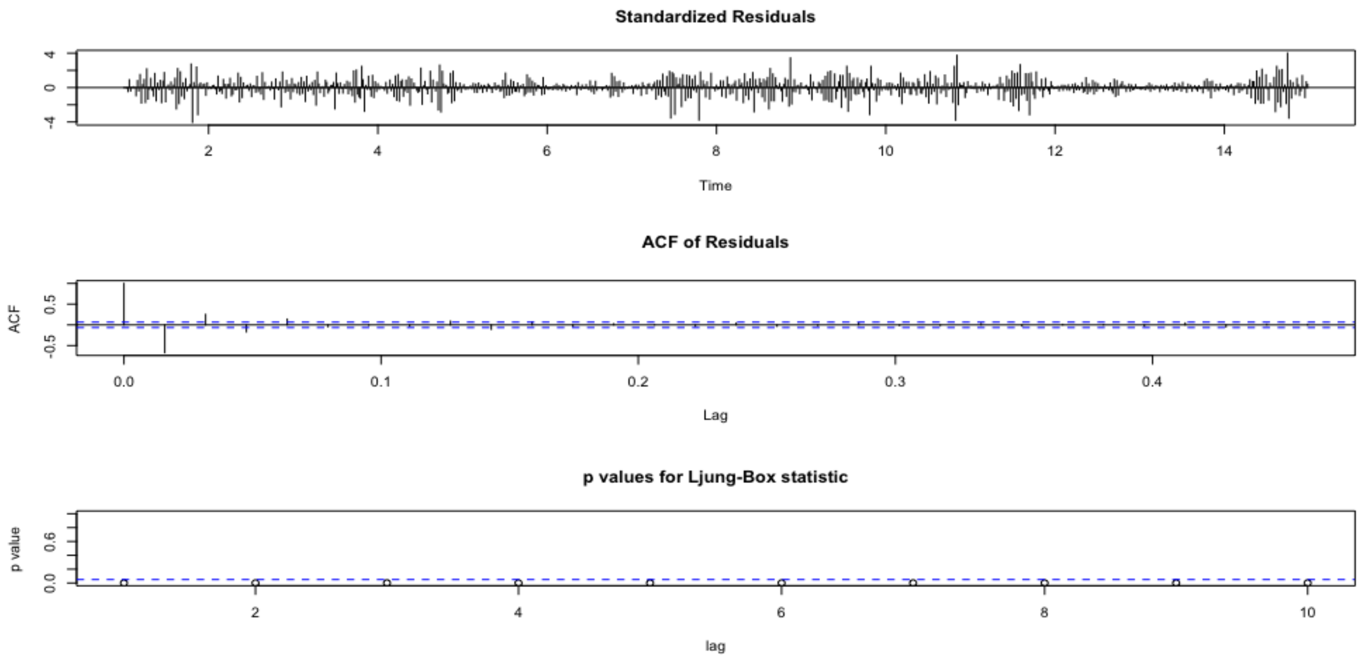
Arima Model

```
bicup.data.res.arima <- Arima(train.lm.season$residuals, order = c(1,1,2))
bicup.data.res.arima1 <- Arima(train.lm.season$residuals, order = c(0,2,0))

summary(bicup.data.res.arima)
tsdiag(bicup.data.res.arima)
```

```
> summary(bicup.data.res.arima)
Series: train.lm.season$residuals
ARIMA(1,1,2)

Coefficients:
          ar1     ma1      ma2
      -0.7939  0.4392  -0.1875
s.e.   0.1176  0.1260   0.0632

sigma^2 estimated as 47.53:  log likelihood=-2942.93
AIC=5893.86   AICc=5893.9   BIC=5912.97

Training set error measures:
                      ME     RMSE      MAE      MPE     MAPE     MASE         ACF1
Training set 0.004271021 6.878814 5.137308 22.41397 174.7559 0.345874 0.0008731013
```

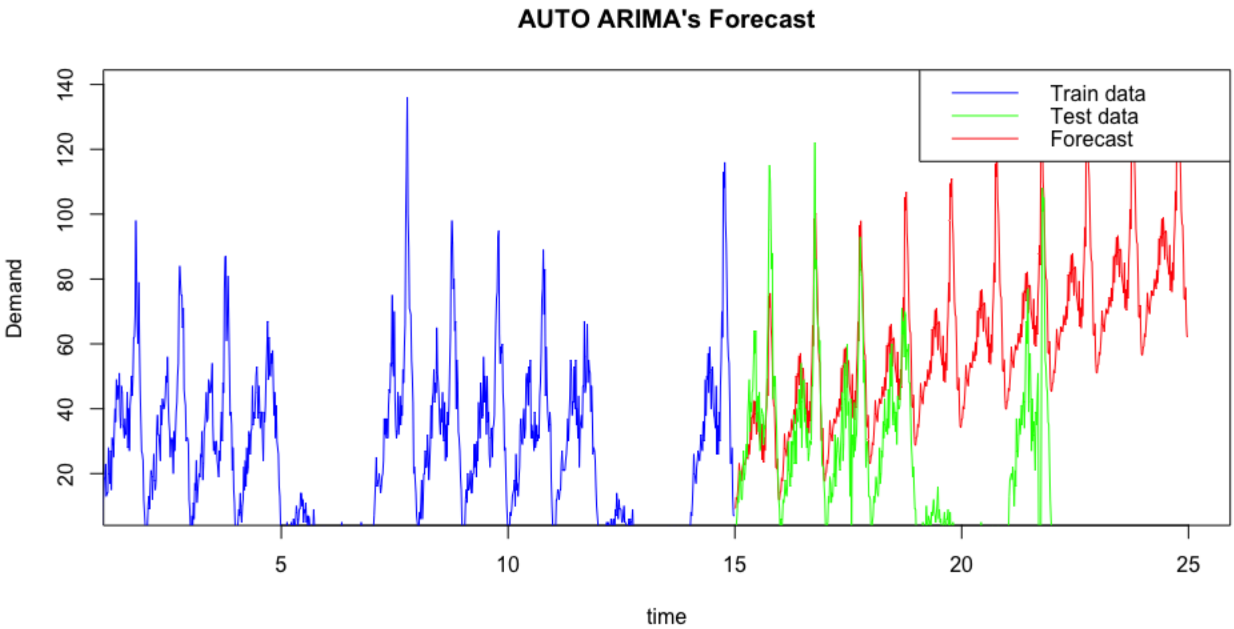



```
> summary(bicup.data.res.arima1)
Series: train.lm.season$residuals
ARIMA(0,2,0)

sigma^2 estimated as 147.3:  log likelihood=-3437.38
AIC=6876.76   AICc=6876.76   BIC=6881.54

Training set error measures:
                      ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
Training set -0.01135975 12.12093 8.746333 39.33333 276.4666 0.5888549 -0.6710168
> tsdiag(bicup.data.res.arima1)
```



Auto Arima model:

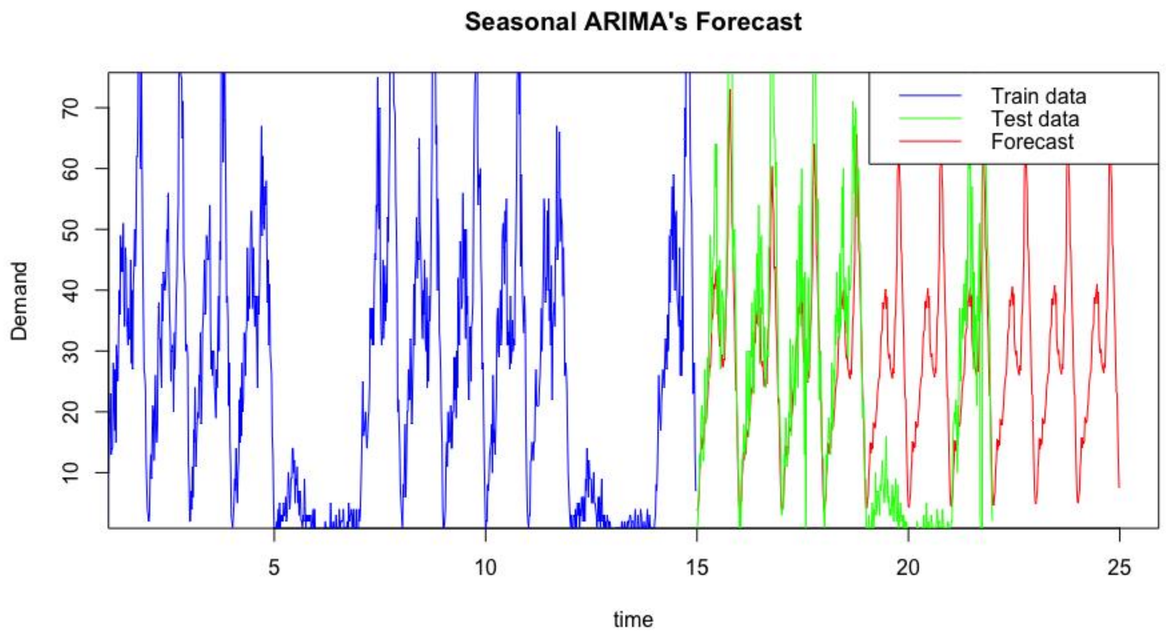
```r
#Auto ARIMA model
bicup.data.arima <- auto.arima(bicup.train.ts)
bicup.data.arima.pred <- forecast(bicup.data.arima, h = 63*7)

summary(bicup.data.arima)
tsdiag(bicup.data.arima)

model31 = accuracy(bicup.data.arima.pred,bicup.valid.ts)

bicup.data.arima.forecast <- forecast(bicup.data.arima, h = 63*10)

plot(bicup.data.arima.forecast$mean, main = "AUTO ARIMA's Forecast", xlim= c(2,25),
     xlab = "time", ylab = "Demand", col= "red")
lines(bicup.train.ts, col = "blue")
lines(bicup.valid.ts, col = "green")
legend("topright", legend = c("Train data", "Test data", "Forecast"),
        lty = c(1, 1, 2), col = c("blue", "green", "red"))
```



Seasonal Arima(SARIMA) model:

```r
bicup.data.sarima=arima(bicup.train.ts,order=c(1,1,2),seasonal=list(order=c(2,1,1),period=63))

summary(bicup.data.sarima)

bicup.data.res.sarima.pred <- forecast(bicup.data.sarima, h = 63*7)

model4 = accuracy(bicup.data.res.sarima.pred,bicup.valid.ts)

bicup.data.res.sarima.forecast = forecast(bicup.data.sarima, h=63*10)

plot(bicup.data.res.sarima.forecast$mean, main = "Seasonal ARIMA's Forecast",
     xlim= c(2,25), xlab = "time", ylab = "Demand", col= "red")
lines(bicup.train.ts, col = "blue")
lines(bicup.valid.ts, col = "green")
legend("topright", legend = c("Train data", "Test data", "Forecast"),
        lty = c(1, 1, 1), col = c("blue", "green", "red"))
```

```
> summary(bicup.data.sarima)

Call:
arima(x = bicup.train.ts, order = c(1, 1, 2), seasonal = list(order = c(2, 1,
    1), period = 63))

Coefficients:
          ar1     ma1      ma2     sar1      sar2      sma1
      -0.7776  0.4089  -0.1907   0.0428  -0.0944   -1.0000
s.e.   0.1267  0.1356   0.0678   0.0401   0.0393    0.0522

sigma^2 estimated as 50.1:  log likelihood = -2847.85,  aic = 5709.7

Training set error measures:
                        ME     RMSE      MAE MPE MAPE      MASE         ACF1
Training set -0.003648636 6.816549 4.862251 NaN  Inf 0.8459011 0.0007931953
```

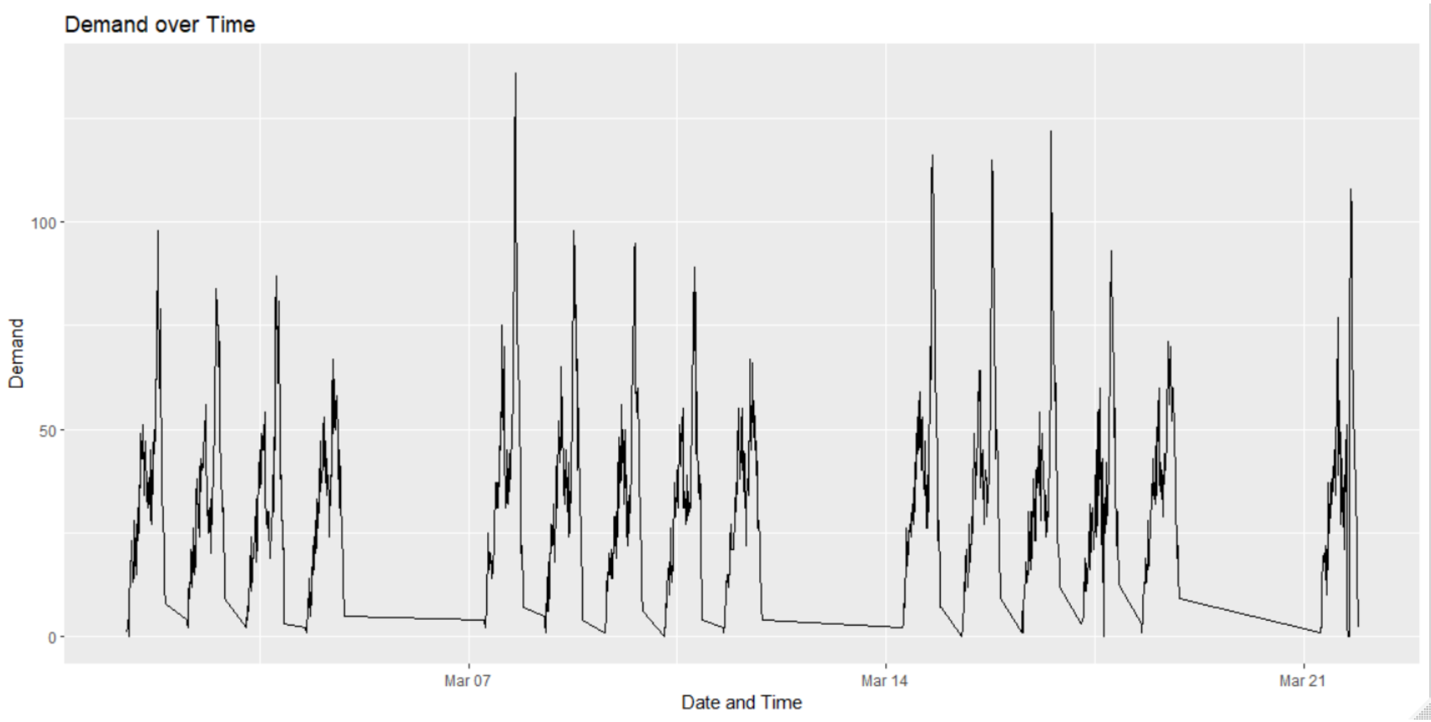


Models when separated weekdays & weekends:

```
#models separated data for weekends & weekdays by Urvish.
df = data.frame(read_excel("bicup2006.xls"))
df$DATE=as.Date(df$DATE)
df$TIME=format(df$TIME, format = "%H:%M")
df$DATETIME = as.POSIXct(paste(df$DATE, df$TIME), format = "%Y-%m-%d %H:%M")

df = subset(df, select = -c(DATE,TIME))
library(lubridate)
library(dplyr)
df$day <- wday(df$DATETIME, label=TRUE)

# filter data based on weekdays and weekends using dplyr
weekday_data <- df %>% filter(day %in% c("Mon", "Tue", "Wed", "Thu", "Fri"))
weekend_data <- df %>% filter(day %in% c("Sat", "Sun"))
```

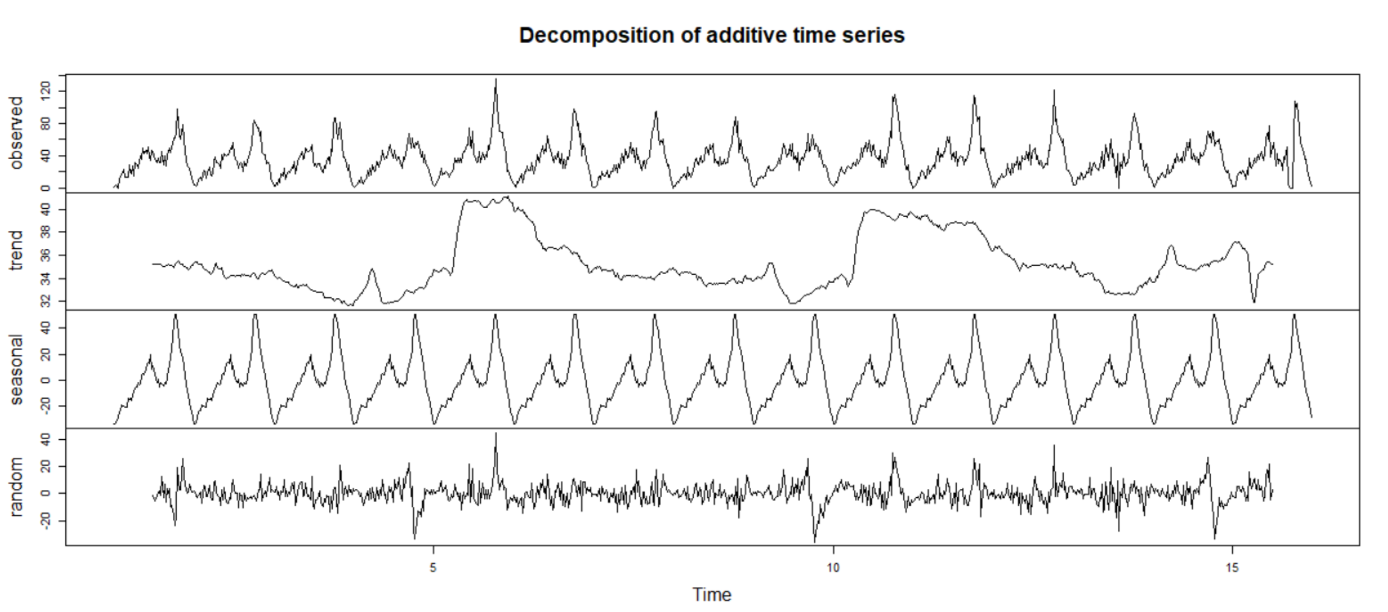
Weekdays plot:

```
# Visualize the demand over time weekdy
ggplot(weekday_data, aes(x = DATETIME, y = DEMAND)) +
  geom_line() +
  labs(title = "Demand over Time", x = "Date and Time", y = "Demand")

```



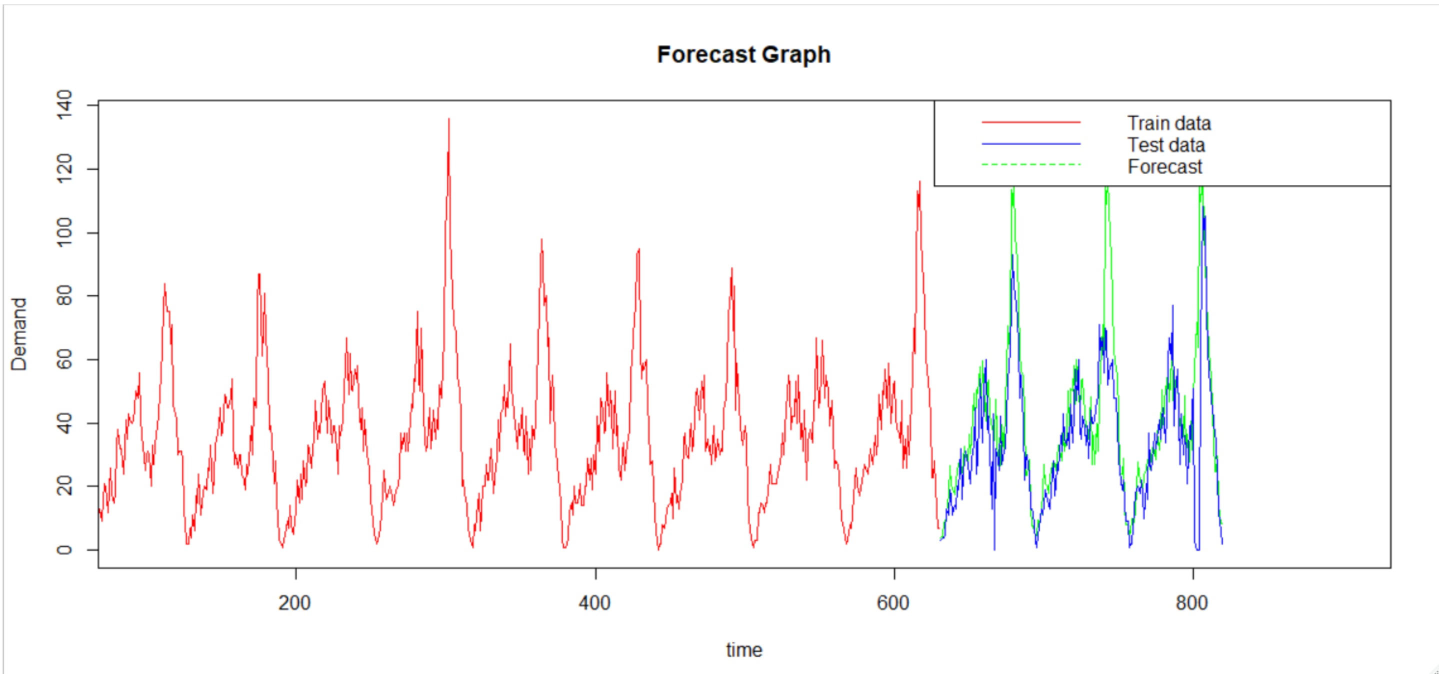
Decomposition pot of weekday data:

```
weekday_data.ts=ts(weekday_data$DEMAND, frequency = 63)

df1 <- data.frame(head(weekday_data$DEMAND, n = 630))
# select remaining rows for second dataframe
df2 <- data.frame(tail(weekday_data$DEMAND, n = 189))

plot(decompose(weekday_data.ts))
```



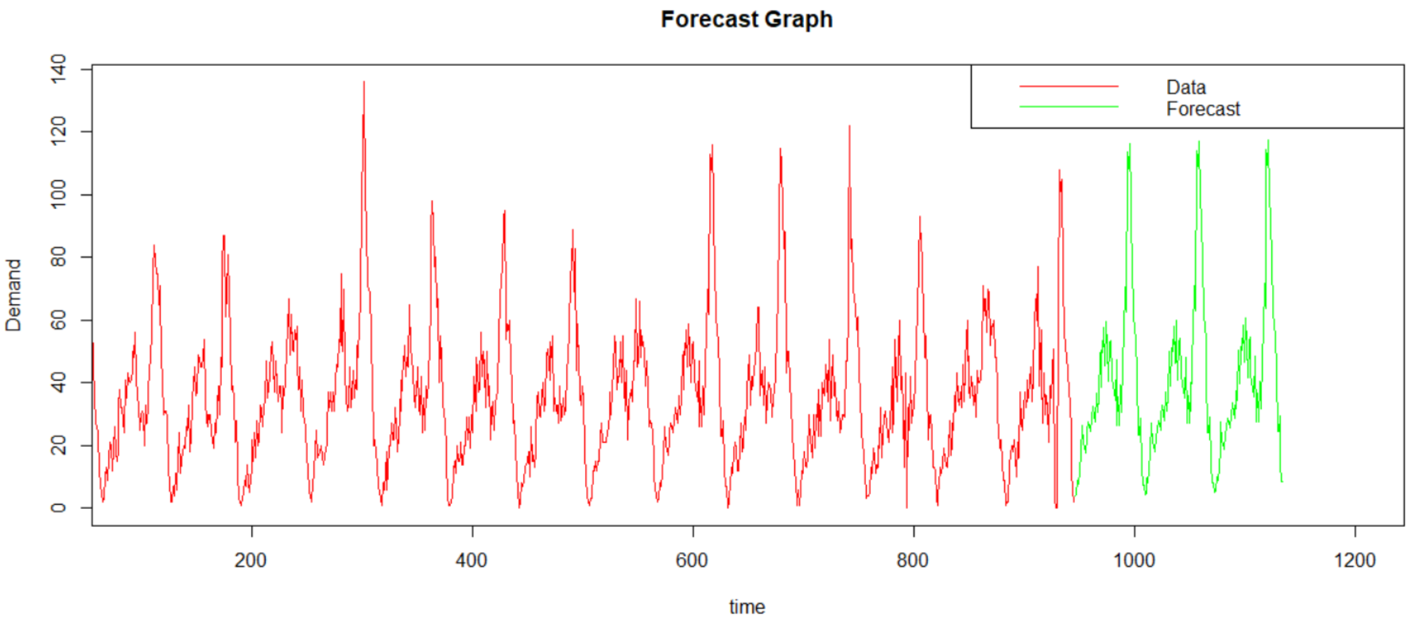
Auto arima model for weekday data:

```r
weekday_data.train.ts = ts(df1, frequency = 63)
weekday_data.valid.ts = ts(df2, frequency = 63)

mod= auto.arima(weekday_data.train.ts)

f_values=forecast(mod, h = 189)
f_values1 = ts(f_values$mean,frequency = 63)
accuracy(f_values1,weekday_data.valid.ts)

index <- c(631:819)
my_df <- data.frame(index,f_values1)


index1 = c(1:630)
my_df1 = data.frame(index1,df1$head.weekday_data.DEMAND..n...630.)

my_df2= data.frame(index,df2$tail.weekday_data.DEMAND..n...189.)
plot(my_df1$index1,my_df1$df1.head.weekday_data.DEMAND..n...630.,
     col="red",type="l", xlim = c(100, 900), main = "Forecast Graph",
     xlab ="time", ylab = "Demand")
lines(my_df$index,my_df$f_values1, col="green")
lines(my_df2$index,my_df2$df2.tail.weekday_data.DEMAND..n...189., col = "blue")
legend("topright", legend = c("Train data", "Test data", "Forecast"),
       lty = c(1, 1, 2), col = c("red","blue", "green"))
```



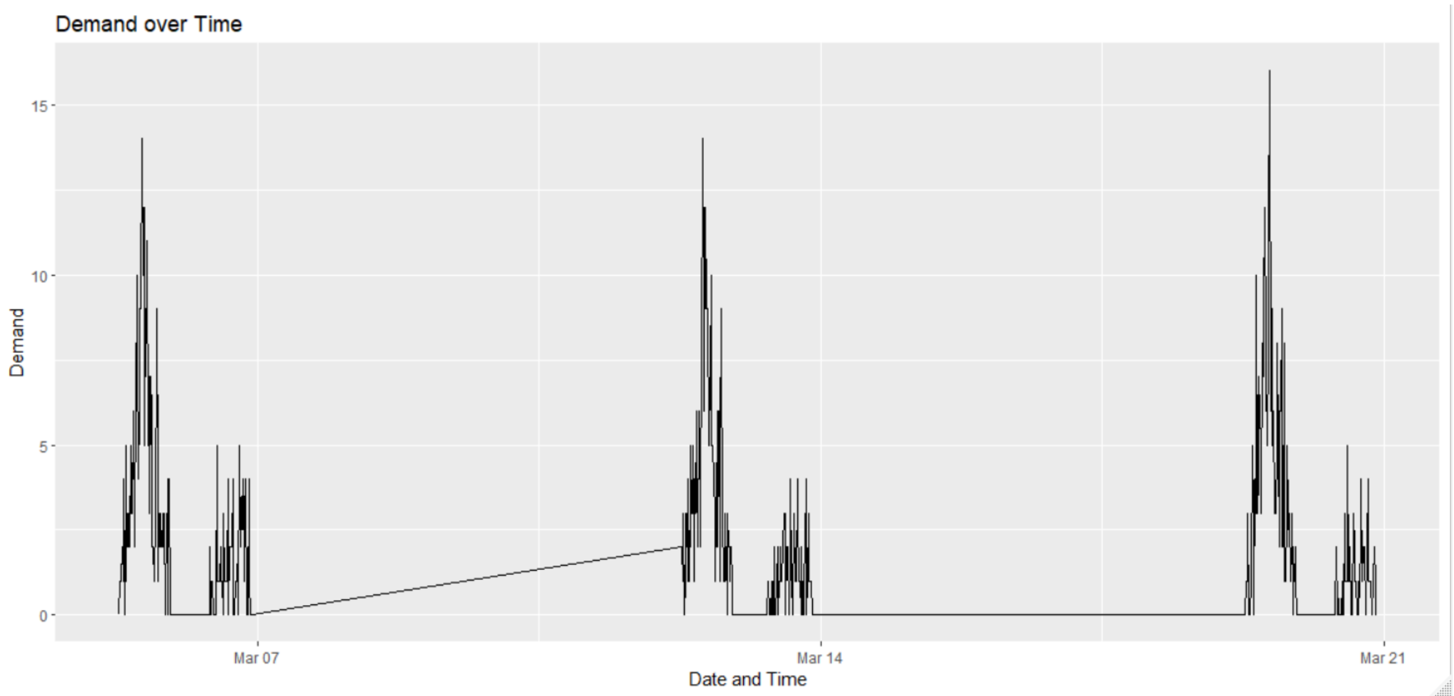
The whole fit of data into model:

```r
#whole_fit
whole_fit=  auto.arima(weekday_data.ts)

f_values=forecast(mod, h = 189)
f_values1 = ts(f_values$mean,frequency = 63)
accuracy(f_values1,weekday_data.valid.ts)

index <- c(946:1134)
my_df <- data.frame(index,f_values1)


index1 = c(1:945)
my_df1 = data.frame(index1,weekday_data$DEMAND)

plot(my_df1$index1,my_df1$weekday_data.DEMAND, col="red",type="l",
     xlim = c(100, 1200), main = "Forecast Graph", xlab ="time", ylab = "Demand")
lines(my_df$index,my_df$f_values1, col="green")
legend("topright", legend = c("Data", "Forecast"),
       lty = c(1, 1, 2), col = c("red", "green"))
```



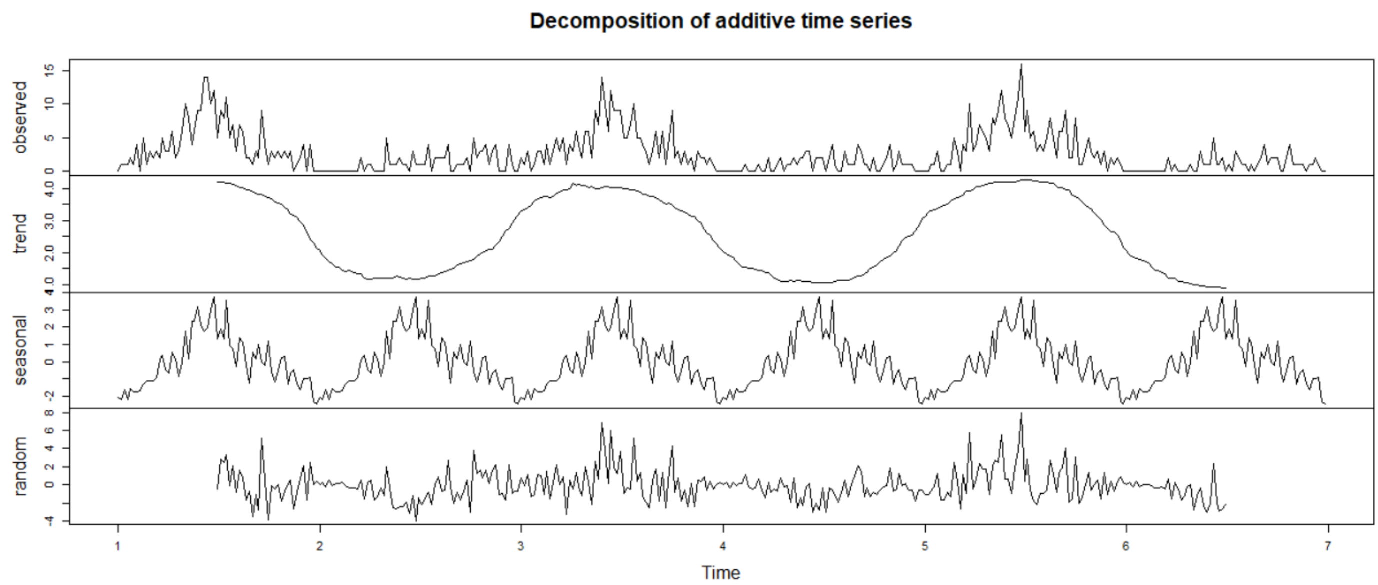
Weekends plot:

```r
# Visualize the demand over time weekends
ggplot(weekend_data, aes(x = DATETIME, y = DEMAND)) +
  geom_line() +
  labs(title = "Demand over Time", x = "Date and Time", y = "Demand")
```



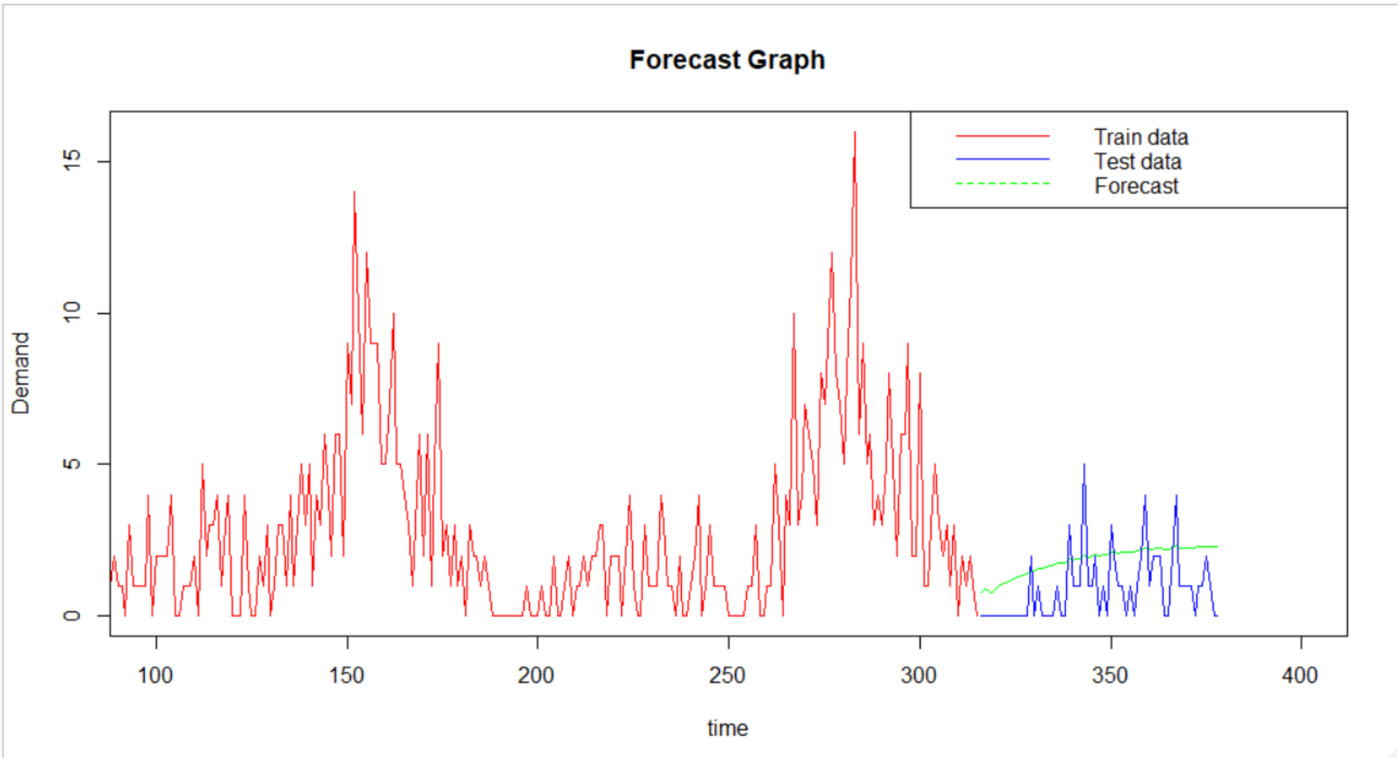
Decomposition plot of weekends:

```
weekend_data.ts=ts(weekend_data$DEMAND, frequency = 63)

df1 <- data.frame(head(weekend_data$DEMAND, n = 315))
# select remaining rows for second dataframe
df2 <- data.frame(tail(weekend_data$DEMAND, n = 63))

plot(decompose(weekend_data.ts))
```



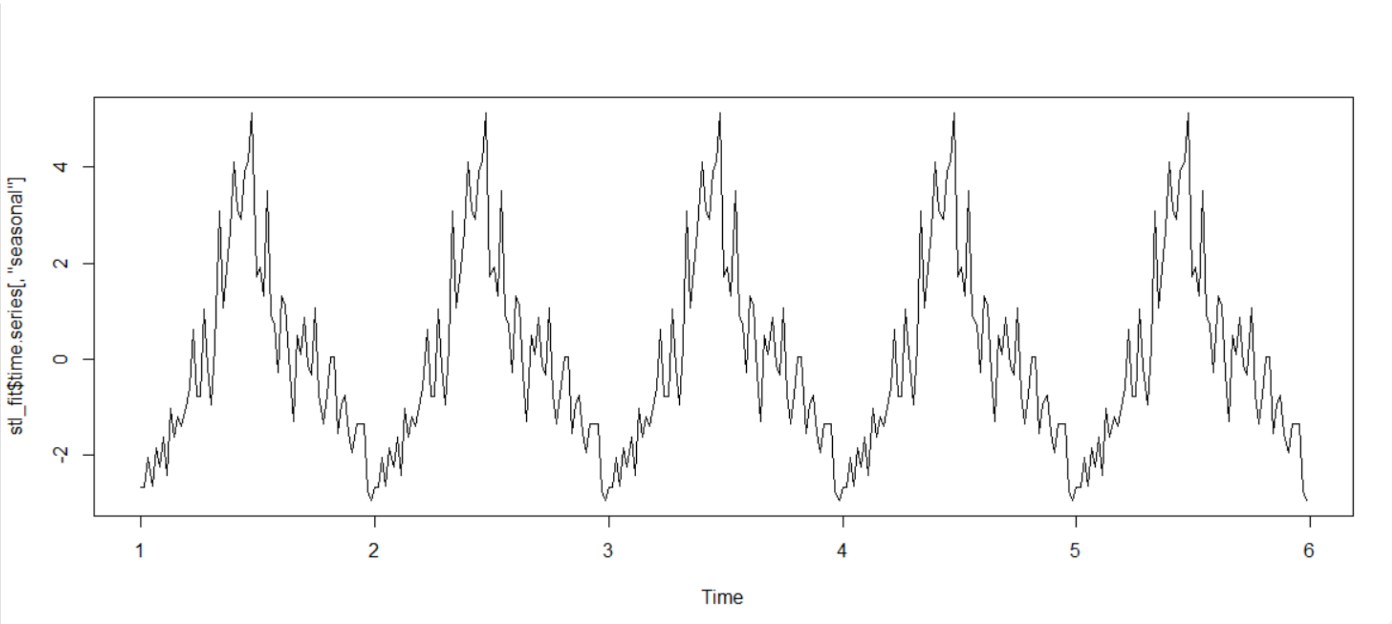
Arima Model for weekends (NOT A GREAT FIT):

```r
weekend_data.train.ts = ts(df1, frequency = 63)
weekend_data.valid.ts = ts(df2, frequency = 63)

mod= auto.arima(weekend_data.ts)

f_values=forecast(mod, h = 63)
f_values1 = ts(f_values$mean,frequency = 63)
accuracy(f_values1,weekend_data.valid.ts)

index <- c(316:378)
my_df <- data.frame(index,f_values1)


index1 = c(1:315)
my_df1 = data.frame(index1,df1$head.weekend_data.DEMAND..n...315.)

my_df2= data.frame(index,df2$tail.weekend_data.DEMAND..n...63.)
plot(my_df1$index1,my_df1$df1.head.weekend_data.DEMAND..n...315., col="red",type="l",
     xlim = c(100, 900), main = "Forecast Graph", xlab ="time", ylab = "Demand")
lines(my_df$index,my_df$f_values1, col="green")
lines(my_df2$index,my_df2$df2.tail.weekend_data.DEMAND..n...63., col = "blue")
legend("topright", legend = c("Train data", "Test data", "Forecast"),
       lty = c(1, 1, 2), col = c("red","blue", "green"))

```



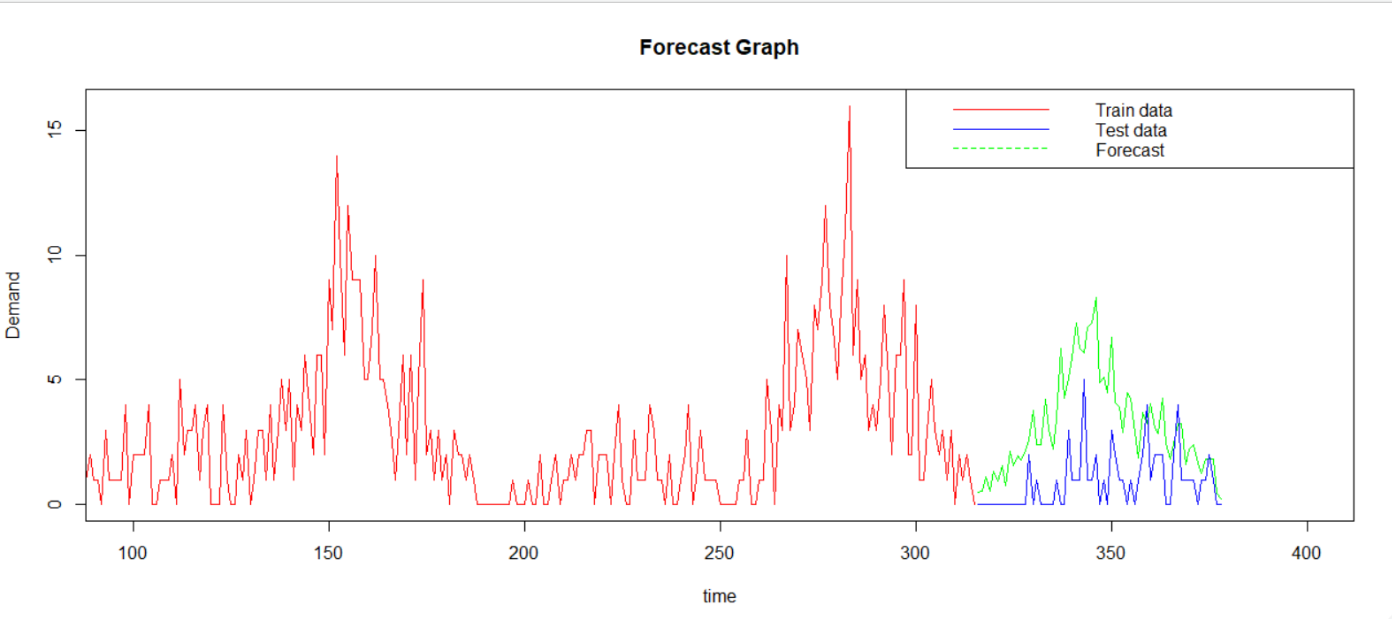
Plot of STL model for Weekends:

```r
weekday_data.train.ts=ts(df1$head.weekend_data.DEMAND..n...315., start = c(1,1), frequency = 63)
weekend_data.valid.ts=ts(df2$tail.weekend_data.DEMAND..n...63., start = c(1,1), frequency = 63)

# Apply the STL algorithm to decompose the data
stl_fit <- stl(weekday_data.train.ts, s.window = "periodic")

# Plot the seasonal component
plot(stl_fit$time.series[, "seasonal"], type = "l")
```



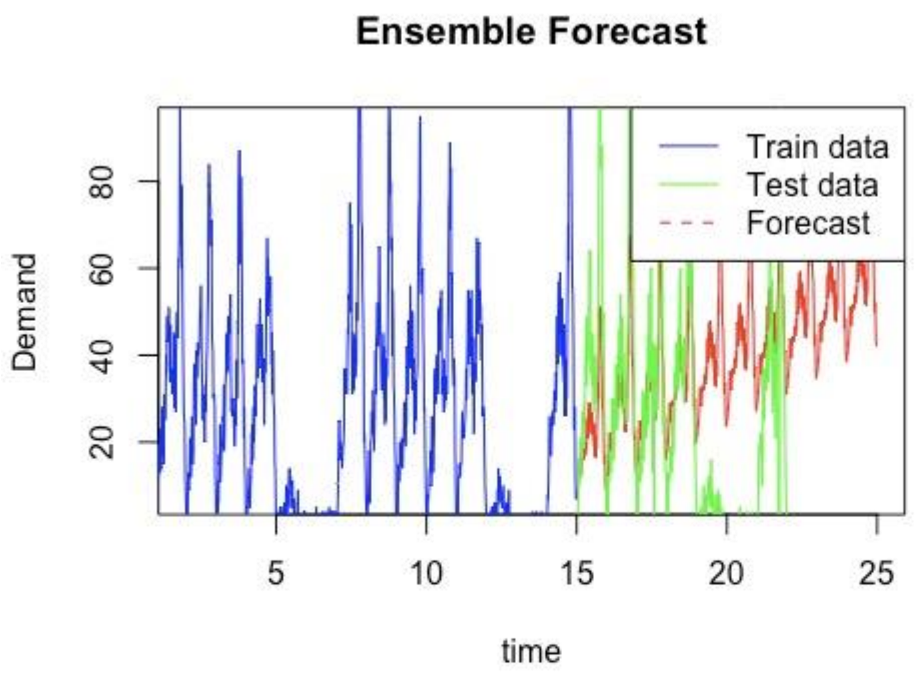
Fit the whole model:

```r
f_values=forecast(stl_fit, h = 63)
f_values1 = ts(f_values$mean,frequency = 63)
accuracy(f_values1,weekend_data.valid.ts)

index <- c(316:378)
my_df <- data.frame(index,f_values1)


index1 = c(1:315)
my_df1 = data.frame(index1,df1$head.weekend_data.DEMAND..n...315.)

my_df2= data.frame(index,df2$tail.weekend_data.DEMAND..n...63.)
plot(my_df1$index1,my_df1$df1.head.weekend_data.DEMAND..n...315., col="red",type="l",
     xlim = c(100, 400), main = "Forecast Graph", xlab ="time", ylab = "Demand")
lines(my_df$index,my_df$f_values1, col="green")
lines(my_df2$index,my_df2$df2.tail.weekend_data.DEMAND..n...63., col = "blue")
legend("topright", legend = c("Train data", "Test data", "Forecast"),
       lty = c(1, 1, 2), col = c("red","blue", "green"))
```



```r
#ENSEMBLE MODEL-SALMA

bicup.data <- read_excel("bicup2006.xls")
df <- data.frame(bicup.data$DEMAND)

bicup.data<-read_excel("bicup2006.xls")
bicup.data.ts <- ts(bicup.data$DEMAND, start= c(1,1), frequency = 63)
nValid = 63*7
nTraining = length(bicup.data.ts)-nValid
bicup.train.ts = window(bicup.data.ts, end = c(1,nTraining))
bicup.valid.ts = window(bicup.data.ts, start = c(1,nTraining+1))

# Split the data into training and validation sets
#train <- df[1:1058,]
#valid <- df[1059:1323,]

# Fit ARIMA and SARIMA models
arima_fit <- auto.arima(bicup.train.ts, seasonal = TRUE)
sarima_fit <- Arima(bicup.train.ts, order = c(1,1,2), seasonal = list(order = c(2,1,1), peri

# Forecast with the ARIMA and SARIMA models
arima_fcst <- forecast(arima_fit, h = 63*10)
sarima_fcst <- forecast(sarima_fit, h = 63*10)

# Ensemble the forecasts
ensemble_fcst <- (arima_fcst$mean + sarima_fcst$mean) / 2

# Compute accuracy measures for the ensemble forecast
accuracy(ensemble_fcst, valid)
summary(ensemble_fcst)
ensemble_forecast <- (forecast::meanf((arima_fcst$mean), (sarima_fcst$mean),h=63*10))
accuracy(ensemble_forecast)
ensemble.forecast <- (bicup.data.res.arima.forecast$mean + bicup.data.arima.forecast$mean +

plot(ensemble.forecast, main = "Ensemble Forecast", xlim= c(2,25), xlab = "time", ylab = "De
lines(bicup.train.ts, col = "blue")
lines(bicup.valid.ts, col = "green")
```

```
> accuracy(ensemble_fcst, valid)
                ME     RMSE      MAE  MPE MAPE
Test set -15.65019 33.29386 28.12455 -Inf  Inf
> summary(ensemble_fcst)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  6.512  32.058  42.950  44.596  54.631 102.783
```


Ensemble Forecast
Train data
Test data
Forecast
Demand
20
40
60
80
5
10
15
20
25
time