Marriot Hotel Room Demand Forecasting

Business Forecasting Group 2 - Team 10 :Paraskumar, Karthigeyan, Shanmuga Priya, Raju, Niranjan a.k.a GOD

18/07/2019

Data Cleaning

We First clean the data by converting the respective columns as factors and extract the data subset which will be used in the calculations.

```
#### Data Cleaning
library(readxl)
marriot<-read_excel("Marriot_data (1).xlsx")
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
## * `` -> ...9
## * `` -> ...10
## * `` -> ...11
```

```
marriot$`DOW INDICATOR1`<-factor(marriot$`DOW INDICATOR1`)
DOW<-c("Saturday","Sunday","Monday","Tuesday","Wednesday","Thursday","Friday")
levels(marriot$`DOW INDICATOR1`)<-DOW
marriot$WEEK<-factor(marriot$WEEK)
levels(marriot$WEEK)<-c("Week 1","Week 2","Week 3","Week 4","Week 5","Week 6","Week 7",
                        "Week 8","Week 9","Week 10","Week 11","Week 12","Week 13","Week 14")
marriot[,c(7,8,9,10,11)]<-NULL
data.subset<-data.frame(Demand= marriot$DEMAND[1:87], Day = marriot$`DOW INDICATOR1`[1:87], Week = marri
str(data.subset)
```

```
## 'data.frame':    87 obs. of  3 variables:
##  $ Demand: num  1417 924 982 1149 1021 ...
##  $ Day   : Factor w/ 7 levels "Saturday","Sunday",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ Week  : Factor w/ 14 levels "Week 1","Week 2",..: 1 1 1 1 1 1 1 2 2 2 ...
```

Exploratory Data Anlysis

Univariate Analysis

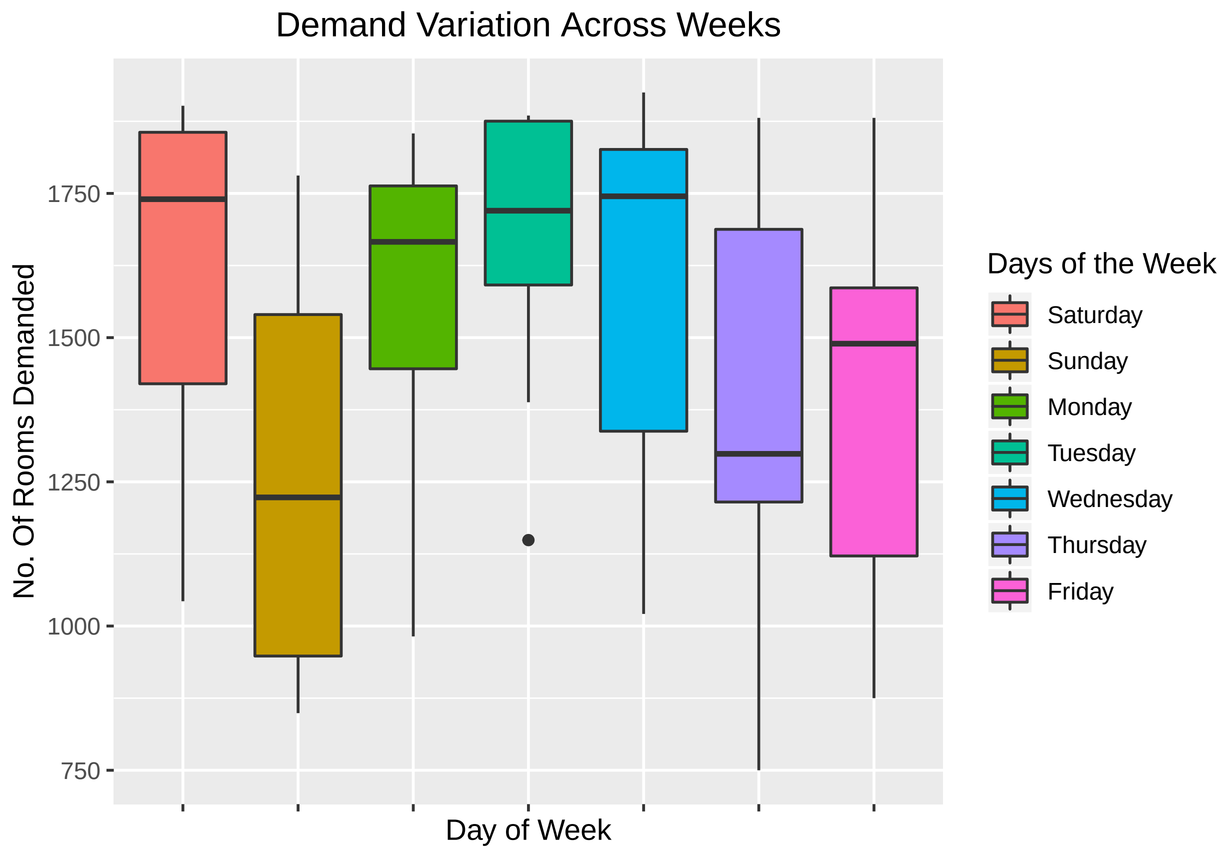
Here we find the descriptive statistics of the hotel room demand.

```
##Univariate Analysis
summary(data.subset$Demand)
```

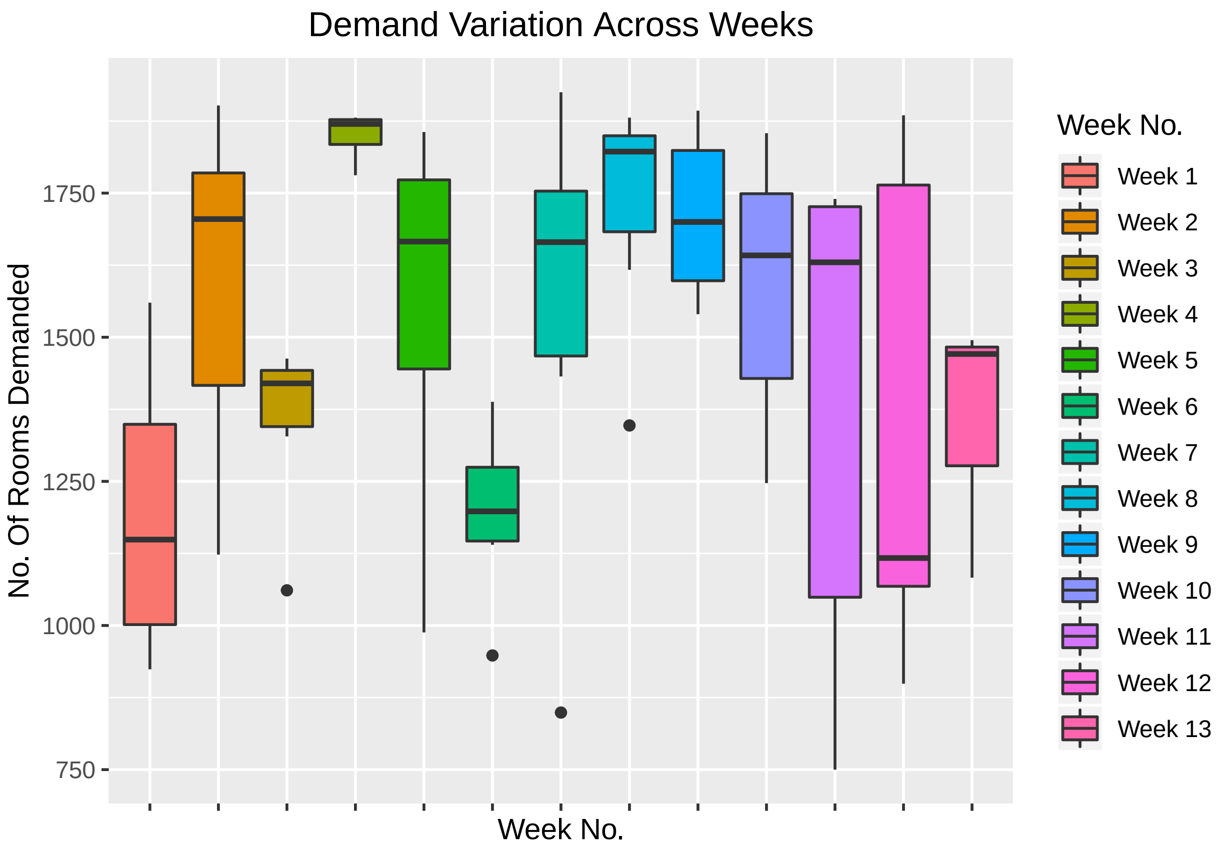
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     750    1244    1553    1501    1780    1925
```

We are also interested in how the demand varies per day and per week. This is captured in the following plots.

```r
library(ggplot2)
demand.day<-ggplot(data.subset,aes(x = data.subset$Day ,
                                   y = data.subset$Demand,
                                   fill=data.subset$Day))+geom_boxplot()
demand.day+labs(title = "Demand Variation Across Weeks",
                x="Day of Week", y="No. Of Rooms Demanded",
                fill="Days of the Week")+
                theme(axis.text.x = element_blank(), plot.title = element_text(hjust = 0.5))
```




```r
demand.week<-ggplot(data.subset,aes(x = data.subset$Week,
                                    y = data.subset$Demand,
                                    fill=data.subset$Week))+geom_boxplot()
demand.week+labs(title = "Demand Variation Across Weeks",
                x="Week No.", y="No. Of Rooms Demanded",
                fill="Week No.")+
                theme(axis.text.x = element_blank(),plot.title = element_text(hjust = 0.5))
```



Time Series Plots

We would like to visually inspect how the demand changes.

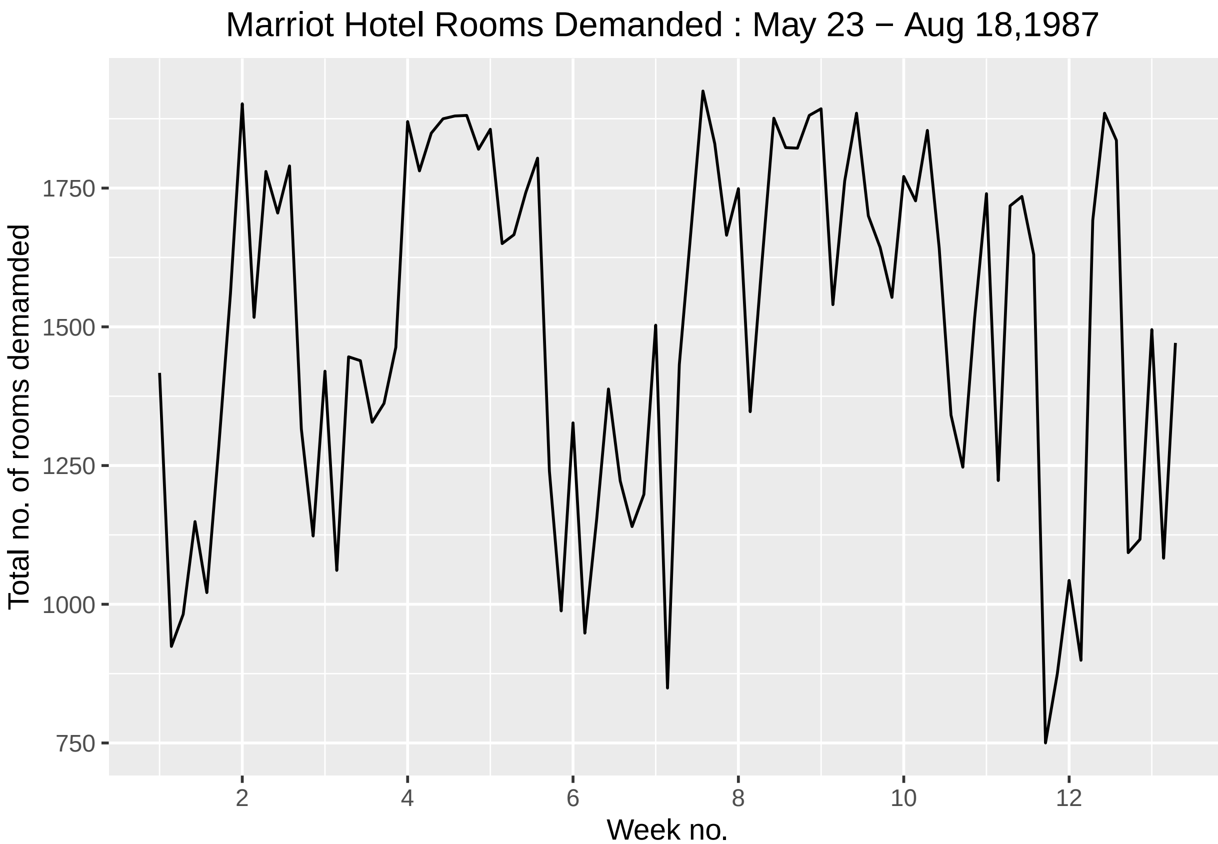
```
### Creating a time-series object
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

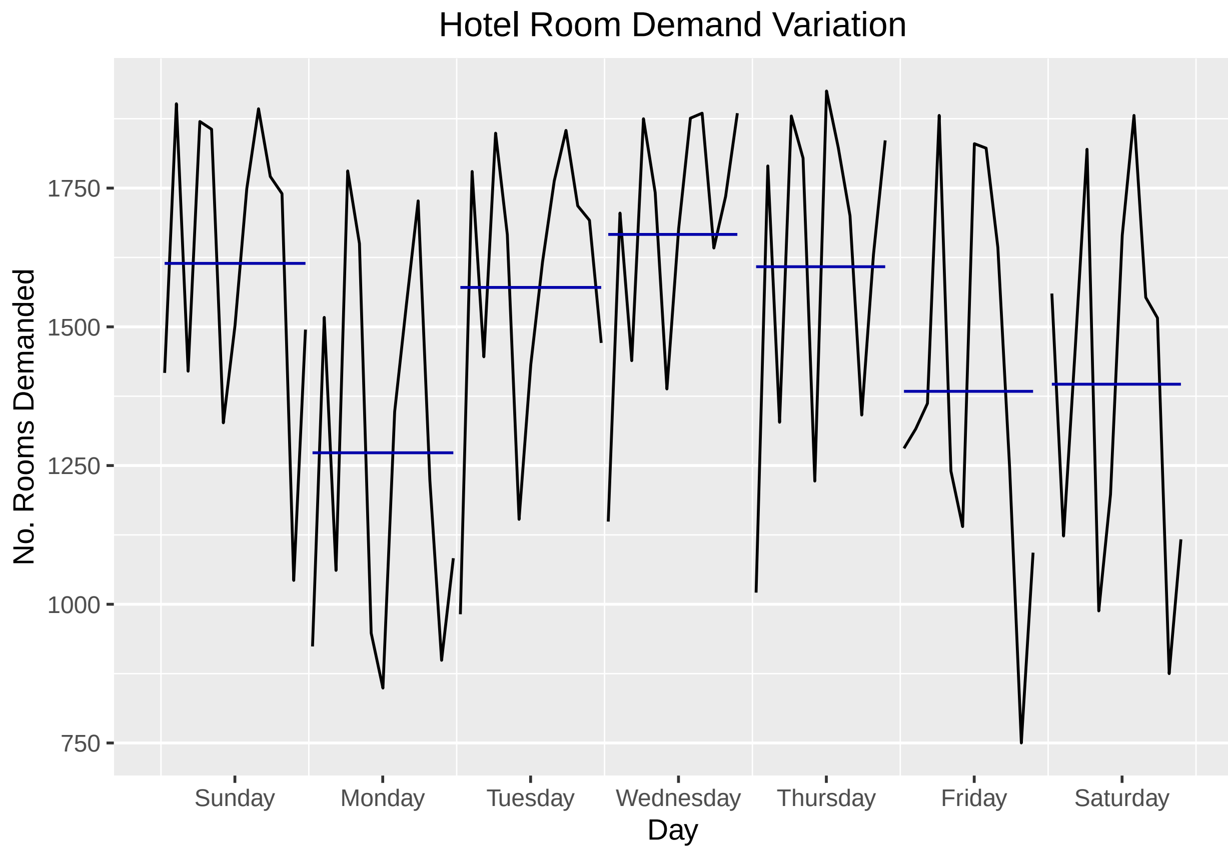
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

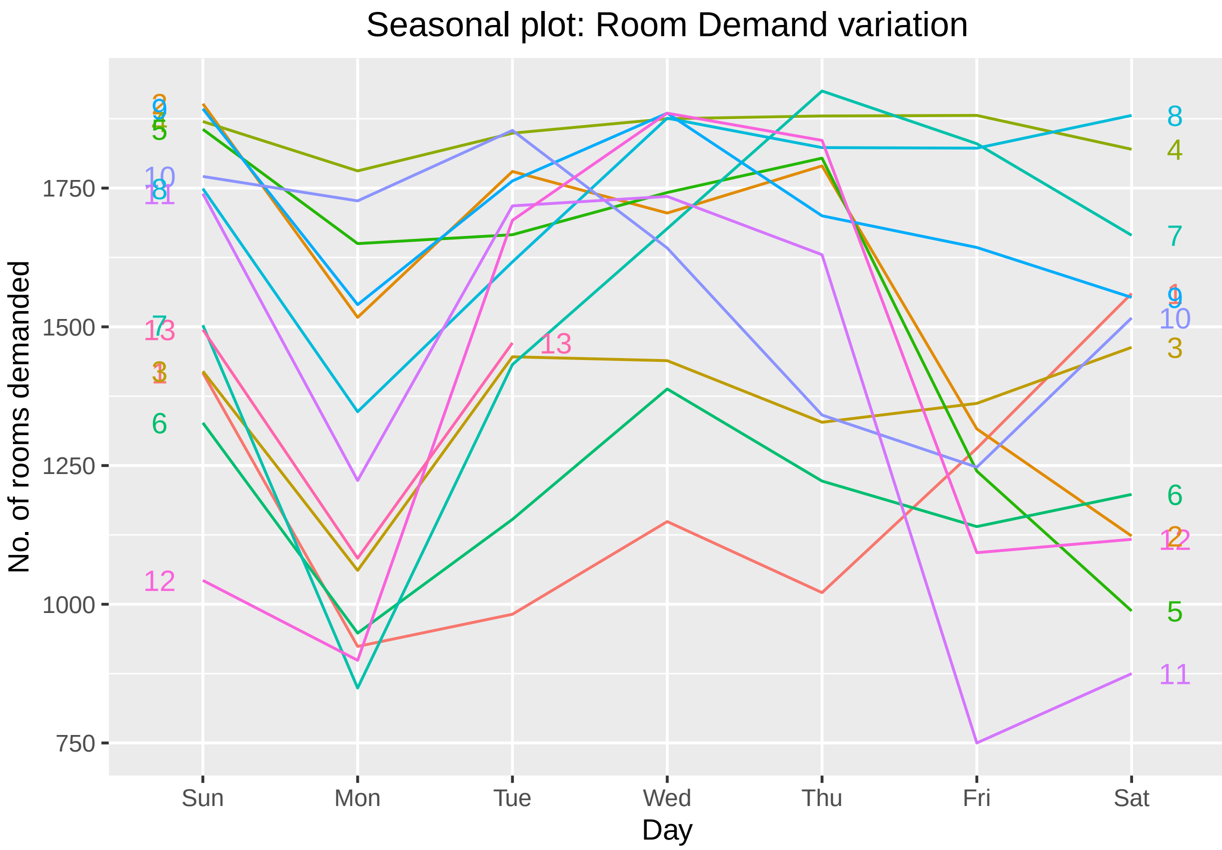
```
demand.ts<-ts(data.subset$Demand,frequency = 7)
autoplot(demand.ts, ylab = "Total no. of rooms demamded", xlab = "Week no.",
         main ="Marriot Hotel Rooms Demanded : May 23 - Aug 18,1987 " )+
         theme(plot.title = element_text(hjust = 0.5))
```




```r
ggsubseriesplot(demand.ts) +
  ylab("No. Rooms Demanded") +
  ggtitle("Hotel Room Demand Variation")+theme(plot.title = element_text(hjust = 0.5))
```




```r
ggseasonplot(demand.ts, year.labels=TRUE, year.labels.left=TRUE) +  ylab("No. of rooms demanded") +
  ggtitle("Seasonal plot: Room Demand variation")+theme(plot.title = element_text(hjust = 0.5))
```



Here we don't notice a trend but a seasonality. To further confirm we conduct the unit root test to identify if there are any random walks present.

#Forcasting Procedure

Stationarity Check

```r
##Staionarity Tests
library(fUnitRoots)
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

## Loading required package: fBasics
```

```r
adfTest(demand.ts) #  #p-value> 0.05 : Fail to reject null hypothesis = Non-stationarity
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 1
##   STATISTIC:
##     Dickey-Fuller: -0.5394
##   P VALUE:
```

```
##     0.4408
##
## Description:
##  Thu Jul 18 16:17:27 2019 by user:
```

```
library(tseries)
kpss.test(demand.ts) #p-value> 0.05 : Fail to reject null hypothesis = Stationarity
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  demand.ts
## KPSS Level = 0.12143, Truncation lag parameter = 3, p-value = 0.1
```

The contradictory results from the above test further leads us to consider not just ARIMA models but also determinsitic trend-seasonality models.

Model Building

Naive

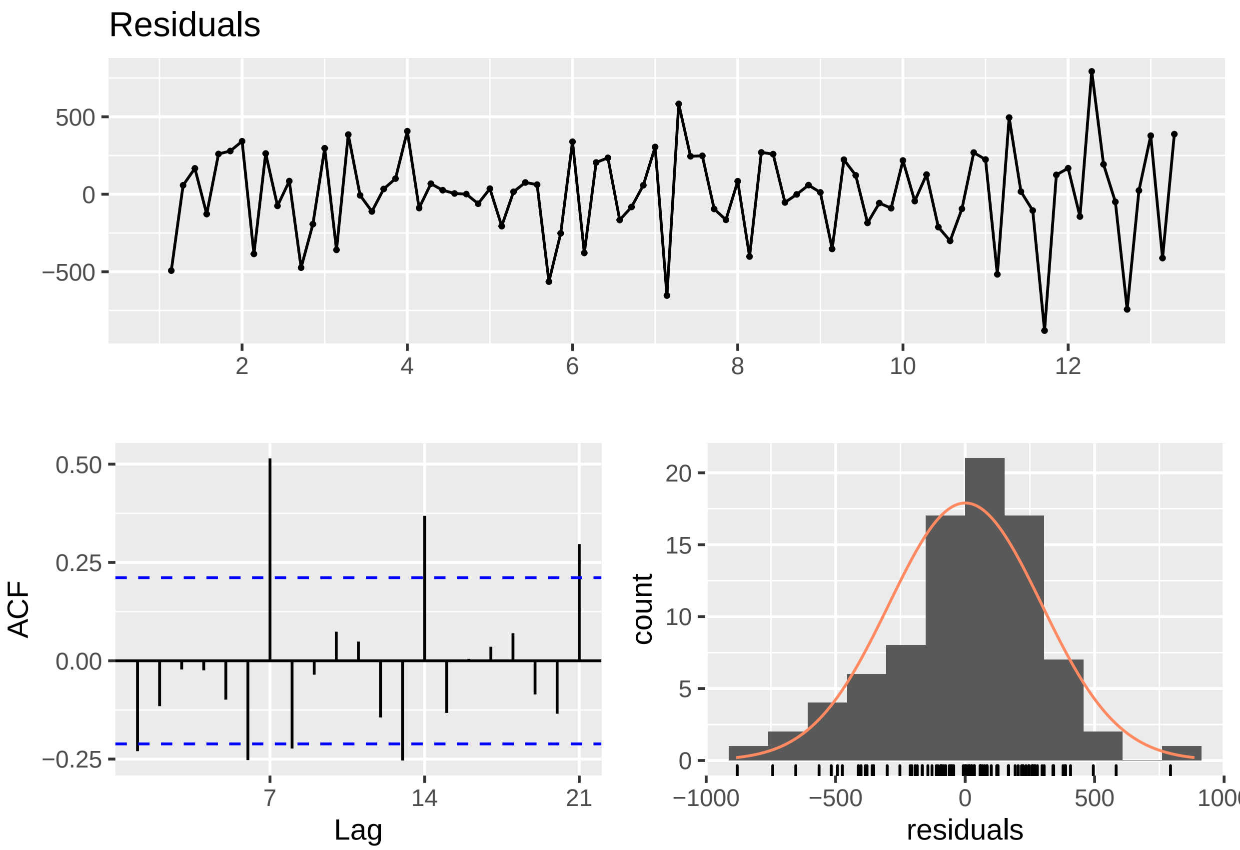
The simplest model to be considered is the naive forcast.

```
#Naive
model.naive<-naive(demand.ts)
model.naive
```

```
##          Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## 13.42857           1471 1095.2444 1846.756  896.33143 2045.669
## 13.57143           1471  939.6014 2002.399  658.29592 2283.704
## 13.71429           1471  820.1722 2121.828  475.64485 2466.355
## 13.85714           1471  719.4888 2222.511  321.66287 2620.337
## 14.00000           1471  630.7850 2311.215  186.00202 2755.998
## 14.14286           1471  550.5906 2391.409   63.35524 2878.645
## 14.28571           1471  476.8442 2465.156  -49.43011 2991.430
## 14.42857           1471  408.2027 2533.797 -154.40816 3096.408
## 14.57143           1471  343.7333 2598.267 -253.00570 3195.006
## 14.71429           1471  282.7565 2659.243 -346.26157 3288.262
```

```
checkresiduals(model.naive$residuals)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



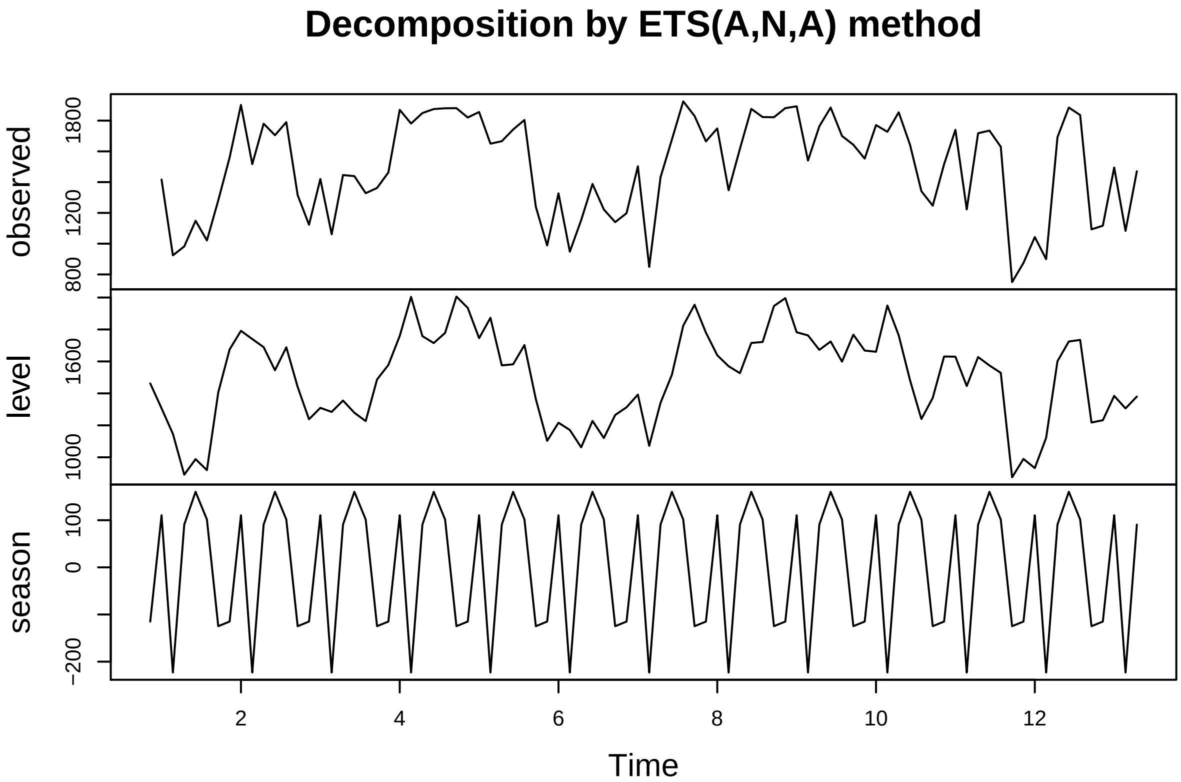
ETS

Next, we consider the Holt-winter model.

```
##Trend-Seasonality Decompositions
model.ets<-ets(demand.ts)
model.ets
```

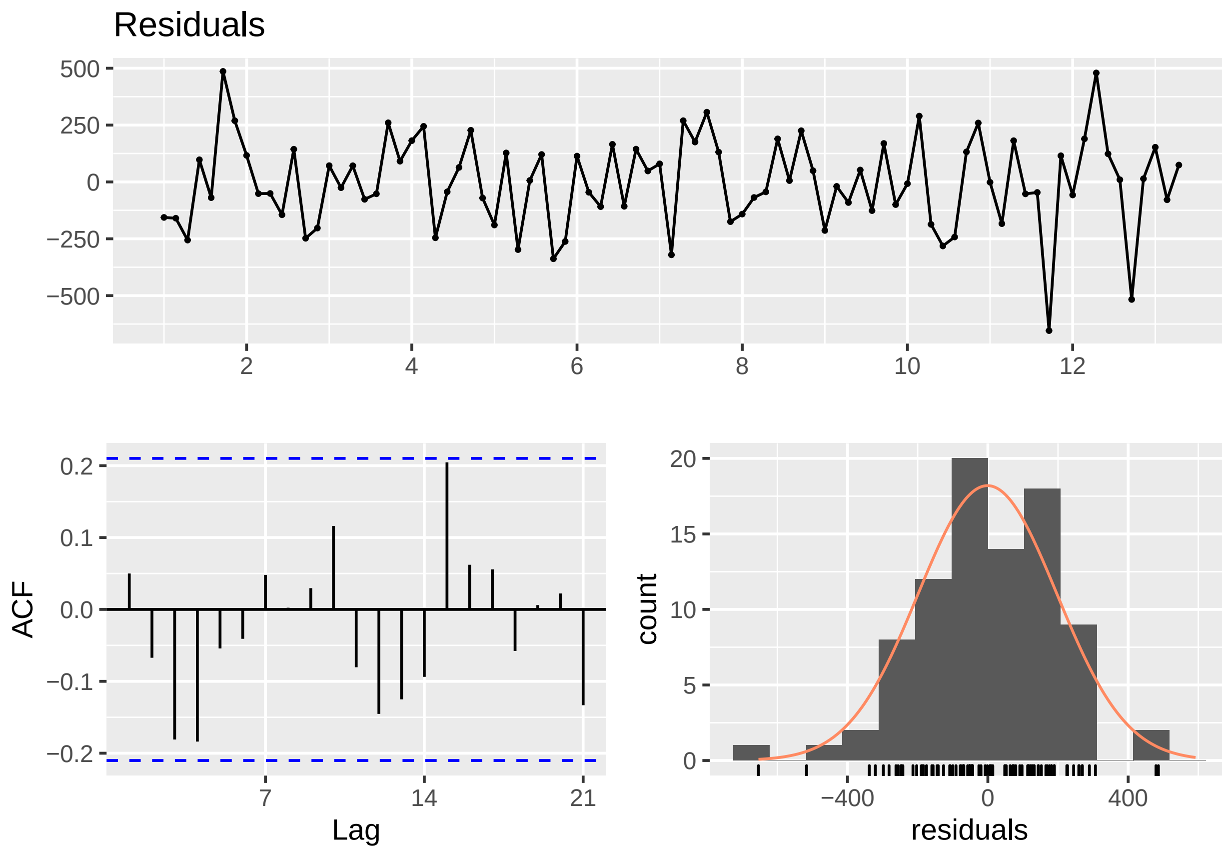
```
## ETS(A,N,A)
##
## Call:
##  ets(y = demand.ts)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     gamma = 1e-04
##
##   Initial states:
##     l = 1462.5945
##     s = -114.956 -124.8259 101.4008 160.1854 90.7313 -223.011
##            110.4753
##
##   sigma:  207.5801
##
##      AIC     AICc      BIC
## 1327.414 1330.308 1352.073
```

```r
plot(model.ets)
```




```r
checkresiduals(model.ets$residuals)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
model.ets$aic
```

```
## [1] 1327.414
```

```
accuracy(model.ets)
```

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.946134 196.5502 155.5432 -1.343285 11.48953 0.4741088
##                    ACF1
## Training set 0.05008858
```

ARIMA

Here, we consider ARIMA models.

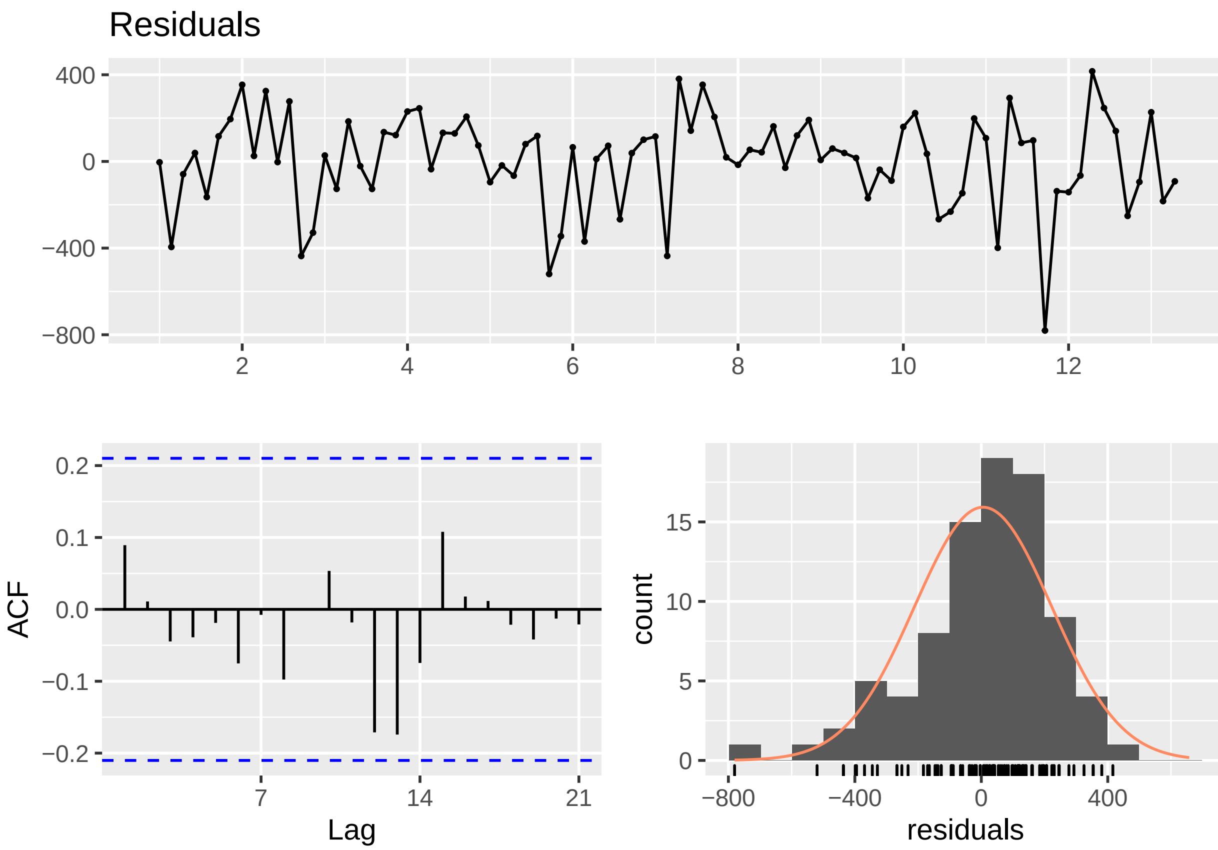
```
#Auto-ARIMA
model.arima<-auto.arima(demand.ts)
model.arima
```

```
## Series: demand.ts
## ARIMA(1,0,0)(2,0,0)[7] with non-zero mean
##
## Coefficients:
##          ar1    sar1    sar2       mean
##       0.7421  0.4322  0.2431  1424.5146
## s.e.  0.0711  0.1072  0.1241   226.0695
##
## sigma^2 estimated as 48984:  log likelihood=-593.43
```

```
## AIC=1196.87   AICc=1197.61   BIC=1209.2
```

```r
checkresiduals(model.arima$residuals)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```r
model.arima$aic
```

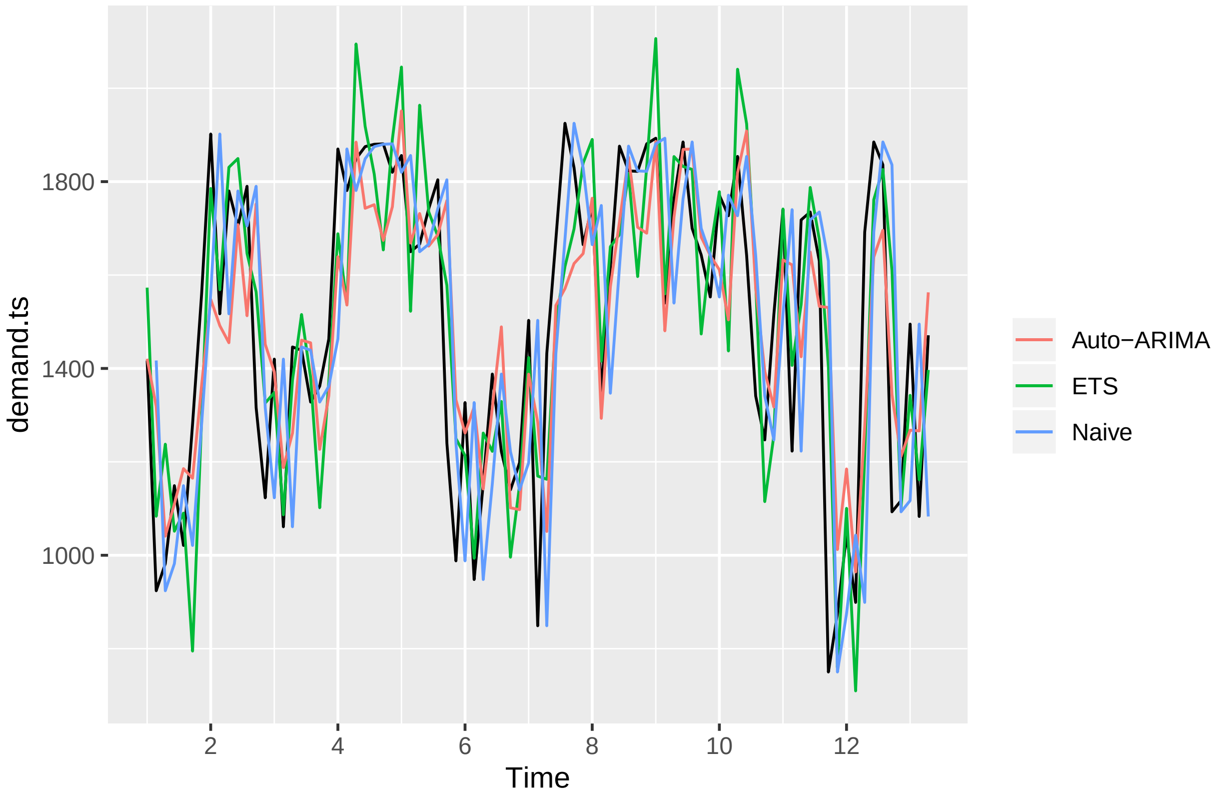
```
## [1] 1196.867
```

Model Evaluation

Here we plot our models.

```r
library(ggplot2)
autoplot(demand.ts) +
  autolayer(fitted(model.ets), series = "ETS") +
  autolayer(fitted(model.arima), series = "Auto-ARIMA") +
  autolayer(fitted(model.naive), series = "Naive")+
  guides(colour = guide_legend(title = " "))
```

```
## Warning: Removed 1 rows containing missing values (geom_path).
```



Now we compare the model AIC values and the MAPE for the three models.

```
accuracy(model.naive)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.627907 293.2036 223.4186 -2.613412 17.11545 0.6809986
##                    ACF1
## Training set -0.2298073
```

```
accuracy(model.arima)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 5.648762 216.1749 165.3773 -2.300914 12.53645 0.5040838
##                    ACF1
## Training set 0.08927444
```

```
accuracy(model.ets)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.946134 196.5502 155.5432 -1.343285 11.48953 0.4741088
##                    ACF1
## Training set 0.05008858
```

By looking at the MAPE values we find the Holt-Winter model to give the best prediction.

Forecast Reccomendation

The last row of the output give the best estimate for Saturday (22 August 1987), the start of week 14.

```r
m2<-forecast(model.ets, h=5);m2
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 13.42857       1540.496 1274.4710 1806.520 1133.6461 1947.345
## 13.57143       1481.691 1105.4938 1857.887  906.3473 2057.034
## 13.71429       1255.465  794.7277 1716.203  550.8280 1960.102
## 13.85714       1265.337  733.3274 1797.346  451.6987 2078.975
## 14.00000       1490.759  895.9578 2085.561  581.0889 2400.430
```

Suggestions To Improve Forecasting

We can further imporve the forecasting by considering a longer time interval.