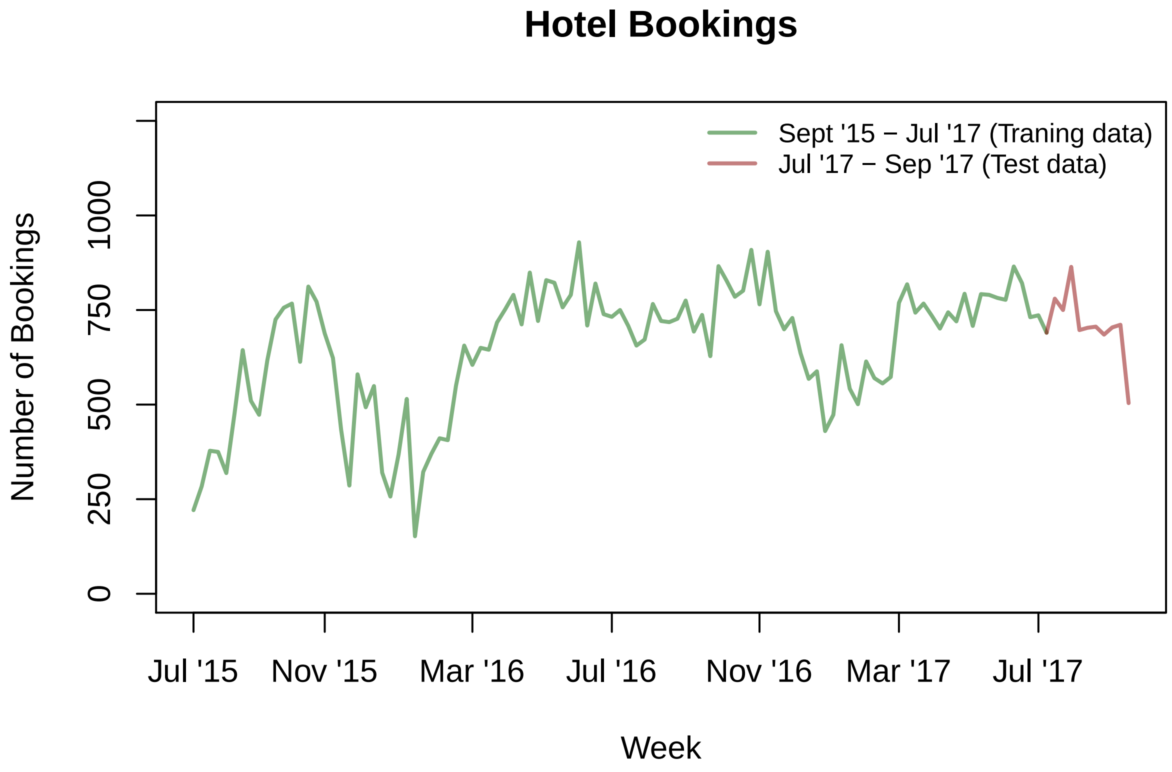
Sonder Hotel Bookings Forecasting

Read in the data and filter for only non-canceled bookings and group by week.

```r
bookings <- read.csv(file = "hotel_bookings.csv", header = TRUE)

library(dplyr)
bookings <- bookings %>%
              filter(is_canceled == 0) %>%
              group_by(arrival_date_year, arrival_date_week_number) %>%
              tally()

bookings <- bookings$n
```

Divide the data into a training set (first 104 weeks) and testing set (last 11 weeks). Provided is a time series plot of the data with training and test sets in different colours.

```r
bookings <- as.numeric(unlist(bookings))
bookings.training <- head(bookings, 105)
bookings.test <- tail(bookings, 11)

plot(bookings.training,
     main = "Hotel Bookings",
     xlab = "Week",
     ylab = "Number of Bookings",
     xlim = c(1, 115),
     ylim = c(0, 1250),
     type = "l",
     lwd = 2,
     col = adjustcolor("darkgreen", 0.5),
     xaxt = "n",
     yaxt = "n")

lines(y = bookings.test,
      x = 105:115,
      lwd = 2,
      col = adjustcolor("darkred", 0.5))

axis(side = 1, at = c(1,17,35,52,70,87,104,122),
     labels = c("Jul '15","Nov '15","Mar '16","Jul '16","Nov '16","Mar '17","Jul '17","Nov '17"))

axis(side = 2, at = (250*0:5), labels = c("0","250","500","750","1000", "1250"))

legend("topright",
       lwd = 2,
       bty = "n",
       cex = 0.8,
       col = c(adjustcolor("darkgreen", 0.5), adjustcolor("darkred", 0.5)),
       legend = c("Sept '15 - Jul '17 (Traning data)", "Jul '17 - Sep '17 (Test data)"),)
```

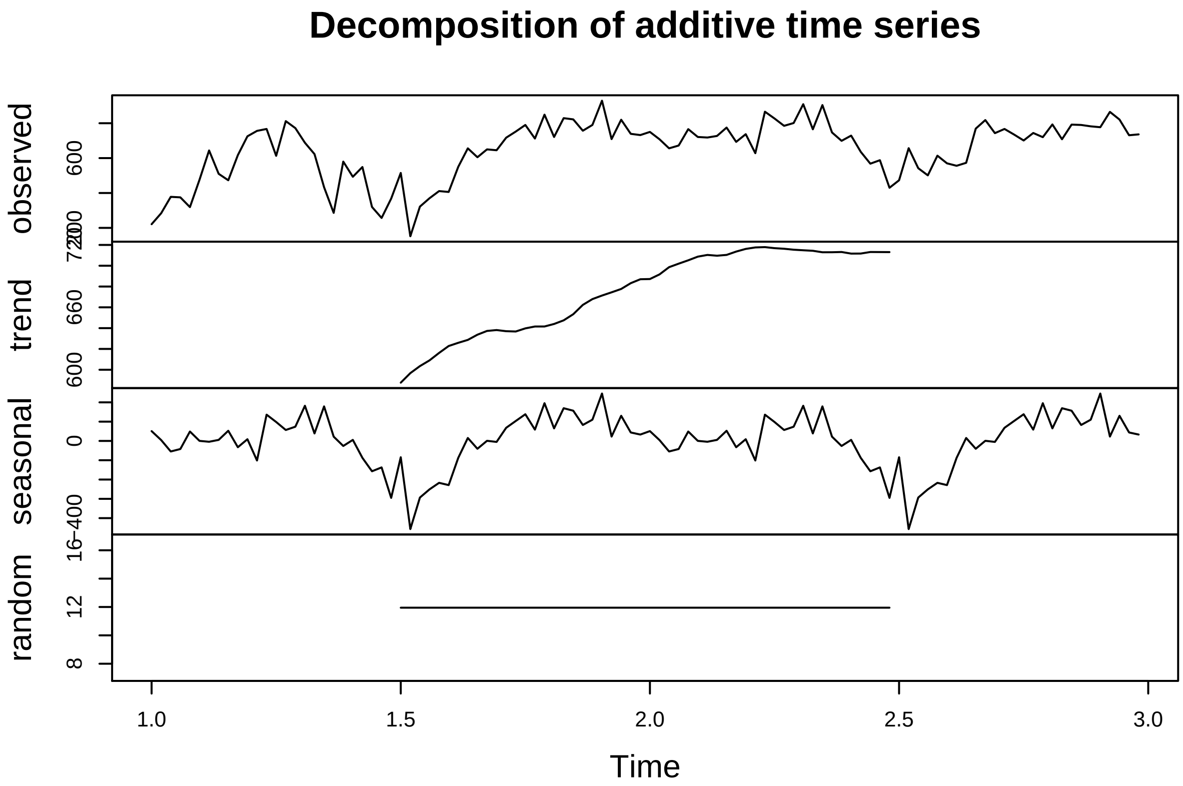


The framework to predict the hotel demand for the remainder of 2017 is as follows:

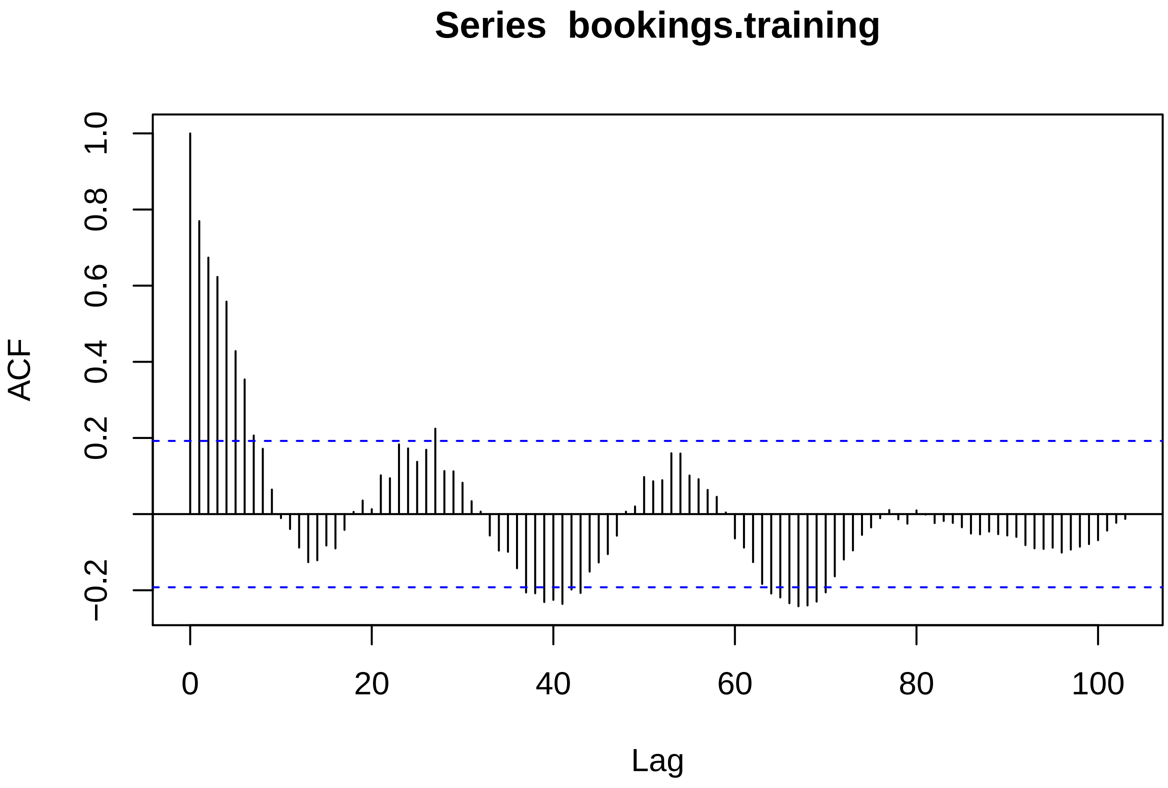
- transform non-stationary data to stationary data
- fit a stationary model
- forecast
- add non-stationarity back

Data with a trend, change points, heteroscedasticity, or seasonality indicate non-stationarity. To better identify non-stationary components, let's decompose the data into its components.

```
bookings.training <- head(bookings, 104)
bookings.ts <- ts(bookings.training, frequency=52) # frequency is 52 since we have weekly data
bookings.decomposed = decompose(bookings.ts, type = "additive")
plot(bookings.decomposed)
```



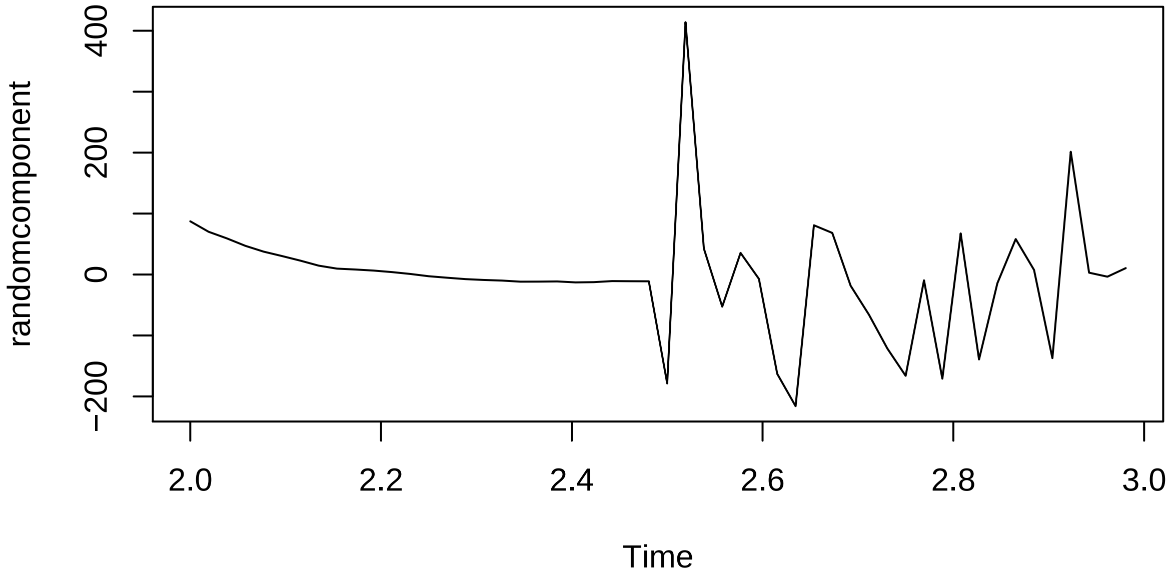
```
acf(bookings.training, lag.max=104)
```



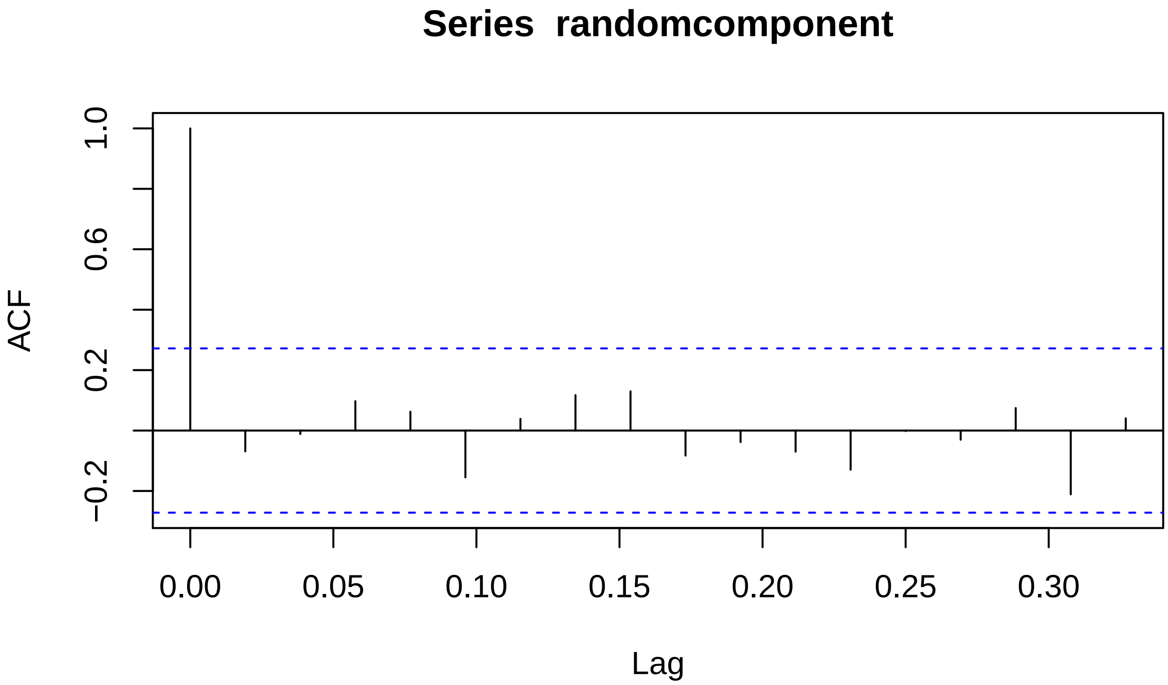
From the plot, there is a clear trend and seasonlity.

There are a few options to remove the non-contant mean including smoothing and differencing. First, we will consider an additive Holt-Winters model.

```
hw.additive <- HoltWinters(bookings.ts, seasonal="additive")
randomcomponent = bookings.ts - hw.additive$fitted[,1]
plot(randomcomponent)
```

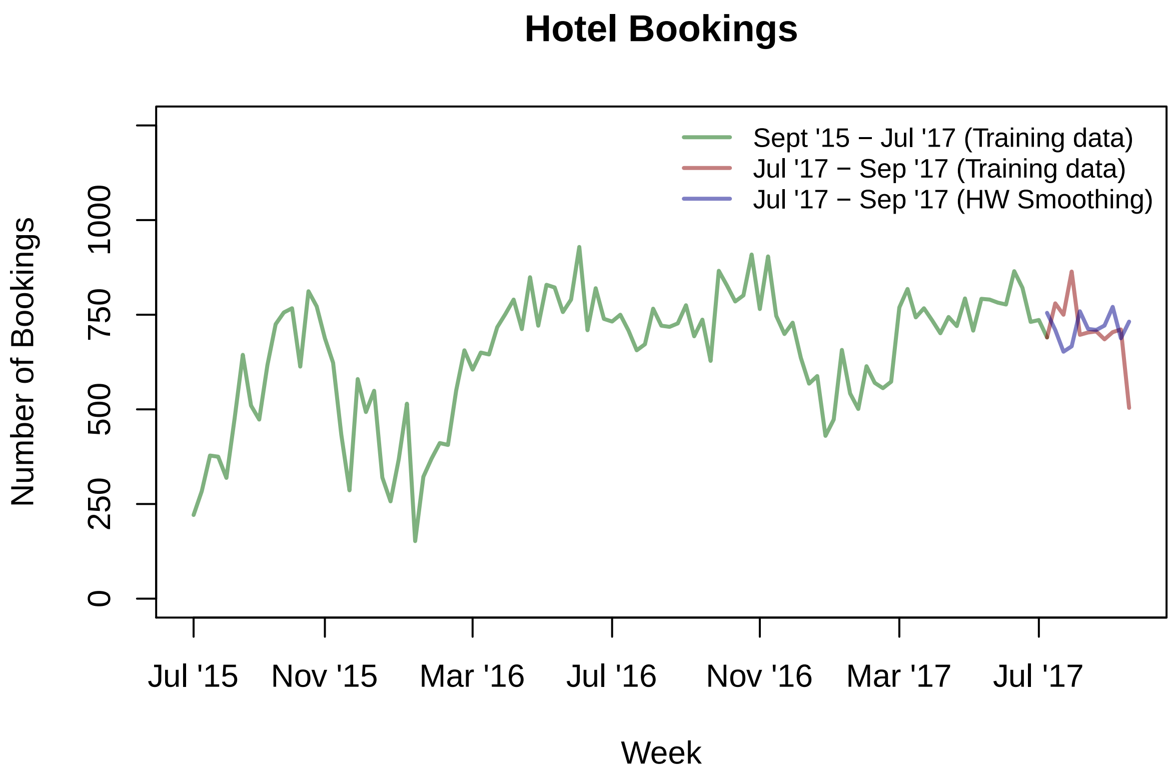


```
acf(randomcomponent)
```



There is no clear trend and/or seasonality in the random component, hence we conclude that the series is stationary.

```
hw.additive <- HoltWinters(bookings.ts, seasonal="additive")
pred.additive <- predict(hw.additive, n.ahead=11)

bookings.training <- head(bookings, 105)
plot(bookings.training,
     main = "Hotel Bookings",
     xlab = "Week",
     ylab = "Number of Bookings",
     xlim = c(1, 115),
     ylim = c(0, 1250),
     type = "l",
     lwd = 2,
     col = adjustcolor("darkgreen", 0.5),
     xaxt = "n",
     yaxt = "n")

lines(y = bookings.test,
      x = 105:115,
      lwd = 2,
      col = adjustcolor("darkred", 0.5))

lines(y = pred.additive,
      x = 105:115,
      lwd = 2,
      col = adjustcolor("darkblue", 0.5))

axis(side = 1, at = c(1,17,35,52,70,87,104,122),
     labels = c("Jul '15","Nov '15","Mar '16","Jul '16","Nov '16","Mar '17","Jul '17","Nov '17"))

axis(side = 2, at = (250*0:5), labels = c("0","250","500","750","1000", "1250"))

legend("topright",
       lwd = 2,
       bty = "n",
       cex = 0.8,
       col = c(adjustcolor("darkgreen", 0.5), adjustcolor("darkred", 0.5), adjustcolor("darkblue", 0.5)
       legend = c("Sept '15 - Jul '17 (Training data)", "Jul '17 - Sep '17 (Training data)", "Jul '17 -
```



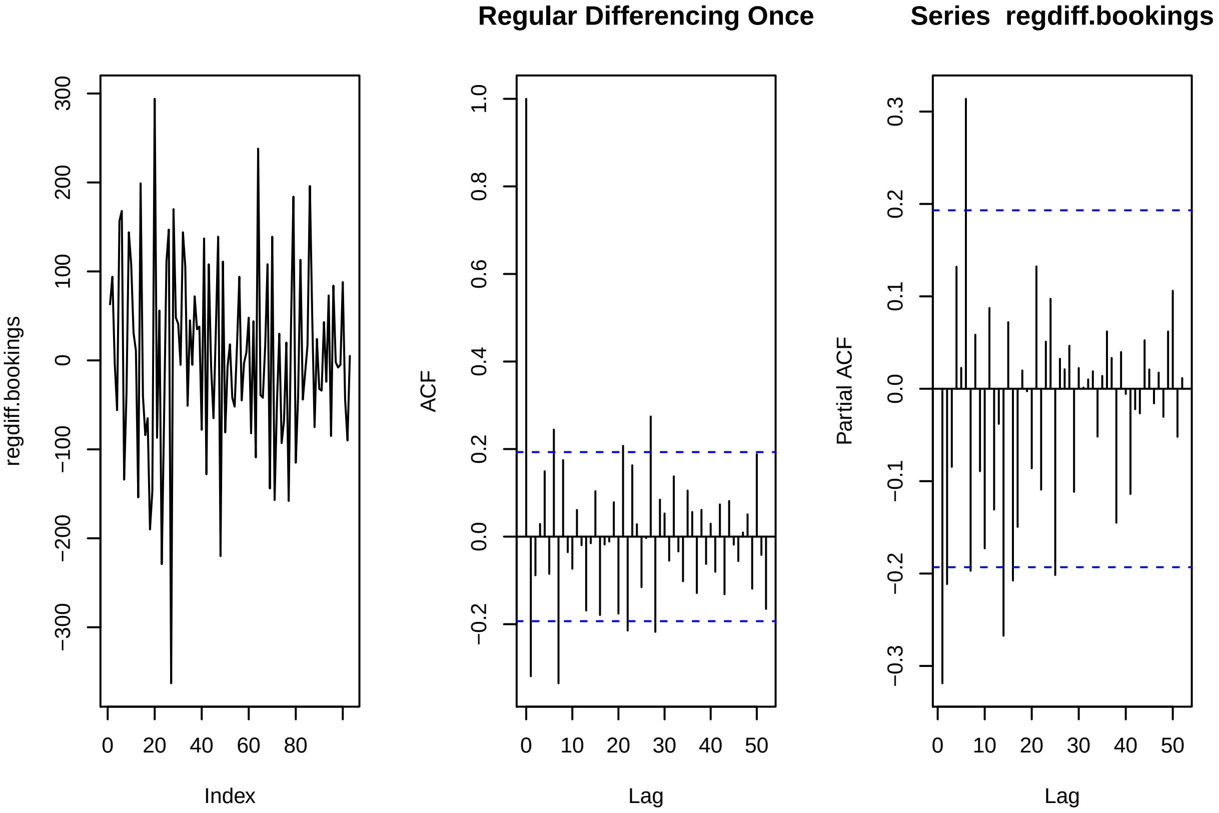
And the SSE_{train} is 119812.8.

```
sum((bookings.test-pred.additive)^2)
```

```
## [1] 119812.8
```

For differencing, we'll propose SARIMA models. Given the acf shows periodicity, we first perform a one time regular differencing and, if necessary, one time seasonal differencing.

```
par(mfrow=c(1,3))

bookings.training <- head(bookings, 104)
regdiff.bookings <- diff(bookings.training, differences=1)
plot(regdiff.bookings, type="l")
acf(regdiff.bookings, lag.max=52, main="Regular Differencing Once")
pacf(regdiff.bookings, lag.max=52)
```



There is no linear decay and/or periodicity left, so we stop here. The times series plot of the differenced series above shows no sign of trend and/or seasonality either.

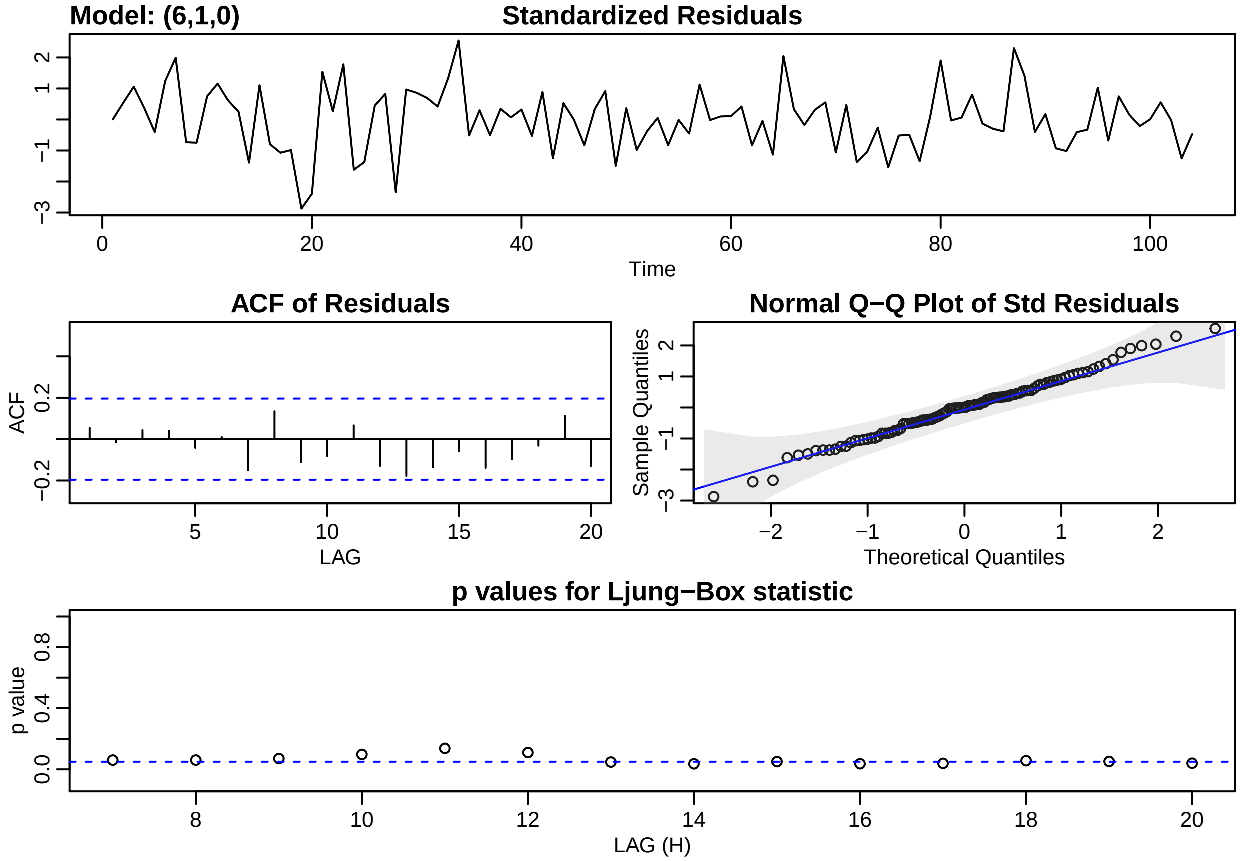
Next, we propose models for the stationary data and perform full residuals diagnostics for the proposed ARIMA models.

- Model 1: the ACF has an exponential decay and the PACF cuts off after lag 6, so we consider ARIMA(6,1,0).
- Model 2: the PACF has an exponential decay and the ACF cuts off after lag 7, so we consider ARIMA(0,1,7).
- Model 3: since in models 1 and 2 we justified exponential decay on the ACF and PACF, it could be exponential decay on both, so we consider ARIMA(1,1,1).

Model 1: ARIMA(6, 1, 0)

```
library(astsa)

model1 <- sarima(bookings.training, p=6, d=1, q=0, details = TRUE)
```

```
## initial  value 4.668937 
## iter   2 value 4.557498
## iter   3 value 4.516215
## iter   4 value 4.514142
## iter   5 value 4.513256
## iter   6 value 4.513223
## iter   7 value 4.513217
## iter   8 value 4.513216
## iter   9 value 4.513216
## iter   9 value 4.513216
## iter   9 value 4.513216
## final  value 4.513216 
## converged
## initial  value 4.531137 
## iter   2 value 4.528791
## iter   3 value 4.527646
## iter   4 value 4.527613
## iter   5 value 4.527545
## iter   6 value 4.527543
## iter   7 value 4.527542
## iter   7 value 4.527542
## iter   7 value 4.527542
## final  value 4.527542 
## converged
```



```
seg <- c(rep(1:6, each=16), rep(7,8))
fligner.test(model1$fit$residuals, seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  model1$fit$residuals and seg
## Fligner-Killeen:med chi-squared = 12.193, df = 6, p-value =
## 0.05779
```

```
shapiro.test(model1$fit$residuals)
```

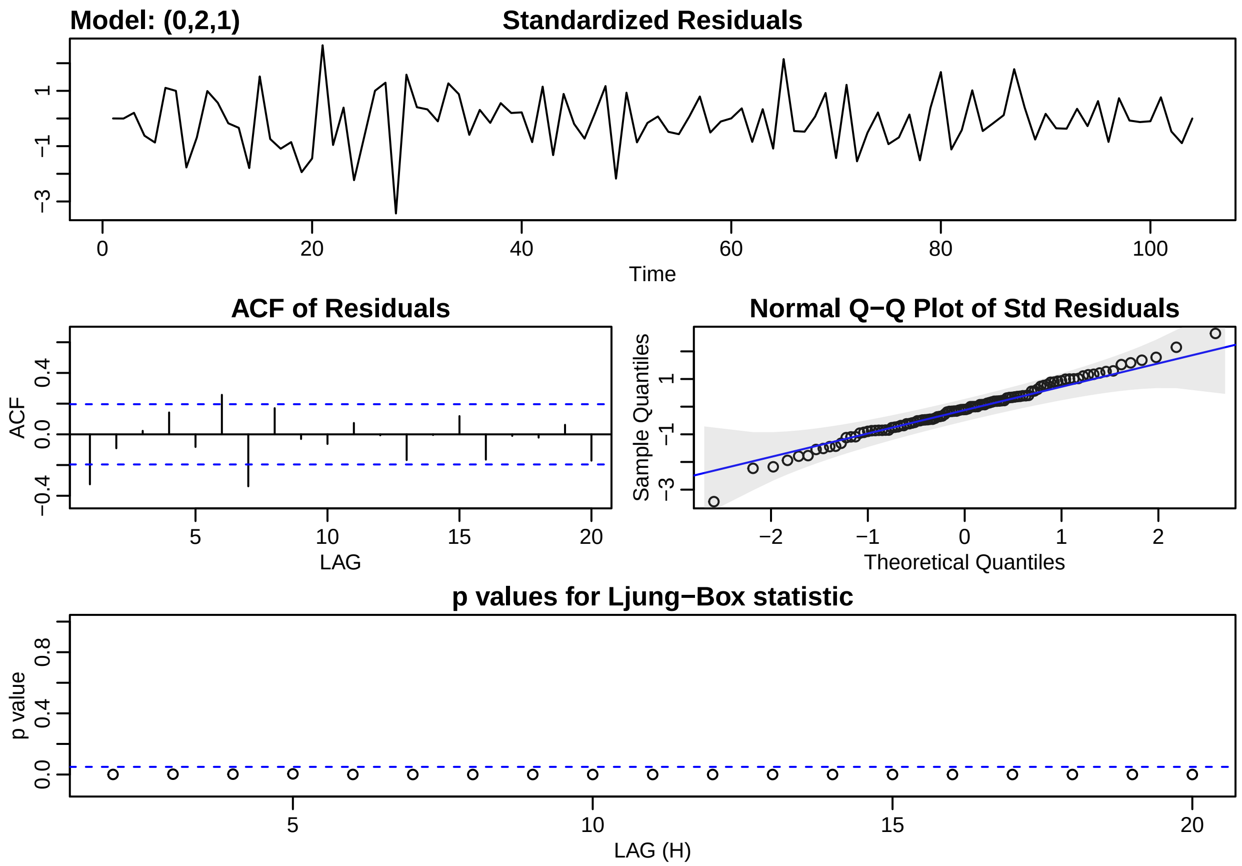
```
##
##  Shapiro-Wilk normality test
##
## data:  model1$fit$residuals
## W = 0.99287, p-value = 0.8653
```

- ARIMA(6, 1, 0)
  - Homoscedasticity: the time series of the residuals appears to have no trend and the p-value of Fligner's test (0.05779) confirms the validity of homoscedasticity.
  - Normality: a few points in the QQplot slightly deviate from the straight line resulting in the p-value of Shapiro-Wilk's test (0.8653). The normality assumption is valid.
  - White Noise: a few p-values of Ljung-Box statistic are below the threshold which raises concerns about the independence of the residuals.

Model 2: ARIMA(0, 1, 7)

```
library(astsa)

model2 <- sarima(bookings.training, p=0, d=2, q=1, details = TRUE)
```

```
## initial  value 5.156737
## iter   2 value 4.850525
## iter   3 value 4.804598
## iter   4 value 4.765159
## iter   5 value 4.730889
## iter   6 value 4.721112
## iter   7 value 4.703716
## iter   8 value 4.702832
## iter   9 value 4.702815
## iter  10 value 4.702814
## iter  10 value 4.702814
## final  value 4.702814
## converged
## initial  value 4.703056
## iter   2 value 4.697225
## iter   3 value 4.695194
## iter   4 value 4.695076
## iter   5 value 4.695073
## iter   6 value 4.695072
## iter   6 value 4.695072
## iter   6 value 4.695072
## final  value 4.695072
## converged
```



```r
seg <- c(rep(1:6, each=16), rep(7,8))
fligner.test(model2$fit$residuals, seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  model2$fit$residuals and seg
## Fligner-Killeen:med chi-squared = 15.533, df = 6, p-value =
## 0.01649
```

```r
shapiro.test(model2$fit$residuals)
```

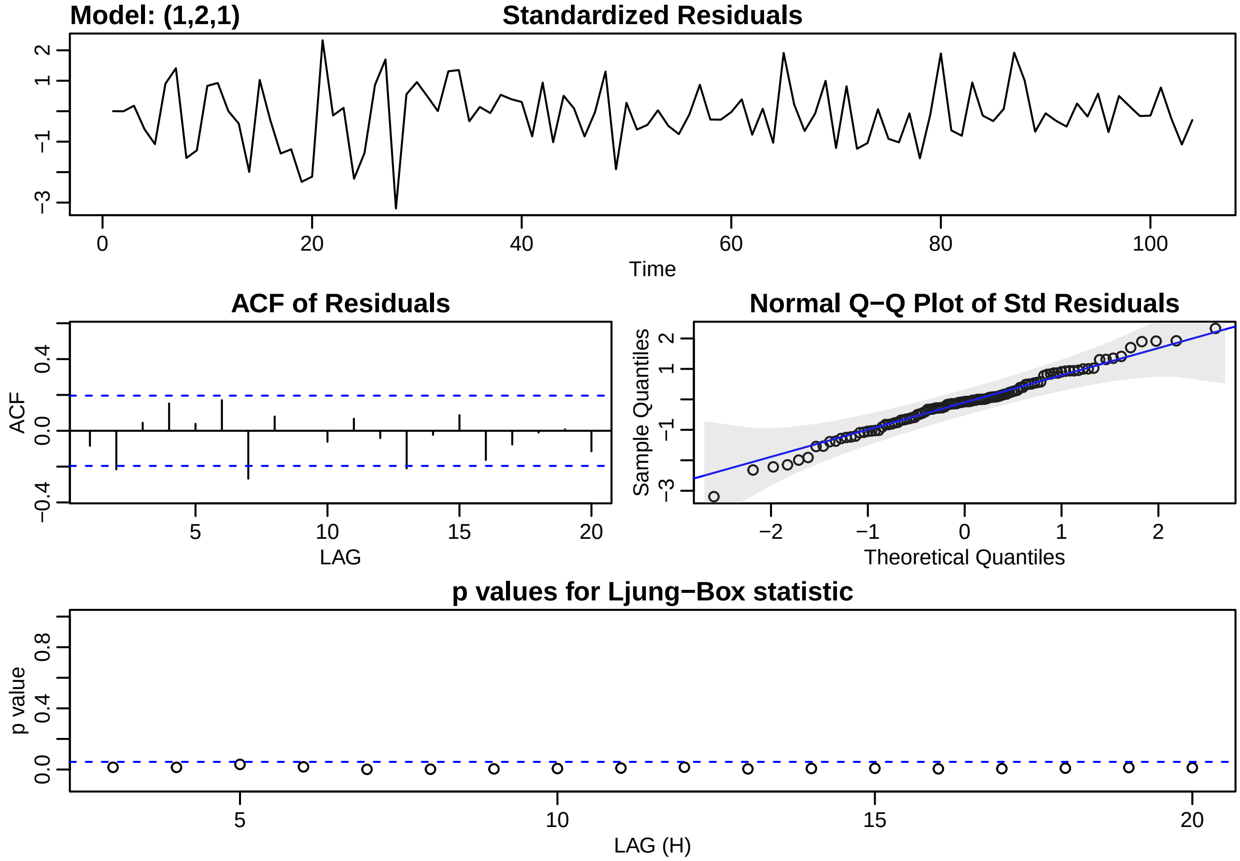
```
##
##  Shapiro-Wilk normality test
##
## data:  model2$fit$residuals
## W = 0.99094, p-value = 0.7161
```

- ARIMA(0, 1, 7)
  - Homoscedasticity: the time series of the residuals has no trend and the p-value of Fligner's test (0.01649) does not raise much concern.
  - Normality: the points in the QQplot don't deviate from the straight line resulting in the large p-value of Shapiro-Wilk's test (0.7161). The normality assumption of the residuals is valid.
  - White Noise: the p-values of the Ljung-Box statistic are below the threshold and there is concern in the ACF plot. We can safely conclude the residuals are not realizations of white noise.

Model 3: ARIMA(1, 1, 1)

```
library(astsa)

model3 <- sarima(bookings.training, p=1, d=2, q=1, details = TRUE)
```

```
## initial  value 5.161506
## iter   2 value 4.774085
## iter   3 value 4.742362
## iter   4 value 4.723250
## iter   5 value 4.711304
## iter   6 value 4.701406
## iter   7 value 4.685902
## iter   8 value 4.676268
## iter   9 value 4.673973
## iter  10 value 4.673110
## iter  11 value 4.672555
## iter  12 value 4.672493
## iter  13 value 4.672490
## iter  13 value 4.672490
## iter  13 value 4.672490
## final  value 4.672490
## converged
## initial  value 4.662495
## iter   2 value 4.651553
## iter   3 value 4.644807
## iter   4 value 4.644586
## iter   5 value 4.644505
## iter   6 value 4.644497
## iter   7 value 4.644496
## iter   8 value 4.644495
## iter   9 value 4.644494
## iter   9 value 4.644494
## iter   9 value 4.644494
## final  value 4.644494
## converged
```



```
seg <- c(rep(1:6, each=16), rep(7,8))
fligner.test(model3$fit$residuals, seg)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  model3$fit$residuals and seg
## Fligner-Killeen:med chi-squared = 19.871, df = 6, p-value =
## 0.00292
```

```
shapiro.test(model3$fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model3$fit$residuals
## W = 0.98941, p-value = 0.5889
```

- ARIMA(1, 1, 1)
  - Homoscedasticity: the time series of the residuals has no trend but the p-value of Fligner's test (0.00292) raises concern. The constant variance assumption is not valid.
  - Normality: the points in the QQplot don't deviate from the straight line resulting in the large p-value of Shapiro-Wilk's test (0.5889). The normality assumption of the residuals is valid.
  - White Noise: the p-values of the Ljung-Box statistic are below the threshold and there is concern in the ACF plot. We can safely conclude the residuals are not realizations of white noise.

Let's compare the three proposed ARIMA models in terms of their prediction power using PRESS and MSE.

```
par(mfrow=c(3,1))
pred.model1 <- sarima.for(bookings.training, n.ahead=11, p=6, d=1, q=0)
pred.model2 <- sarima.for(bookings.training, n.ahead=11, p=0, d=1, q=7)
pred.model3 <- sarima.for(bookings.training, n.ahead=11, p=1, d=1, q=1)
```

```
PRESS.1 <- sum((pred.model1$pred-bookings.test)^2)
PRESS.2 <- sum((pred.model2$pred-bookings.test)^2)
PRESS.3 <- sum((pred.model3$pred-bookings.test)^2)

tab = rbind(1:3, c(PRESS.1, PRESS.2, PRESS.3))
row.names(tab) = c("ARIMA Model", "PRESS")
dimnames(tab)[[2]] = rep("", 3)
tab
```

```
##
## ARIMA Model      1.0      2      3.0
## PRESS       123439.6 203711 150550.7
```

Based on the PRESS statistic, ARIMA(6, 1, 0) predicts the best.

```
MSE.1 <- sum((pred.model1$pred-bookings.test)^2)/3
MSE.2 <- sum((pred.model1$pred-bookings.test)^2)/1
MSE.3 <- sum((pred.model1$pred-bookings.test)^2)/2

tab = rbind(1:3, c(MSE.1, MSE.2, MSE.3))
row.names(tab) = c("ARIMA Model", "MSE")
dimnames(tab)[[2]] = rep("", 3)
tab
```

```
##
## ARIMA Model     1.00      2.0     3.00
## MSE         41146.53 123439.6 61719.79
```

Based on the MSE statistic which accounts for the number of parameters, ARIMA(6, 1, 0) predicts the best.

We choose the superior model to be ARIMA(6, 1, 0).

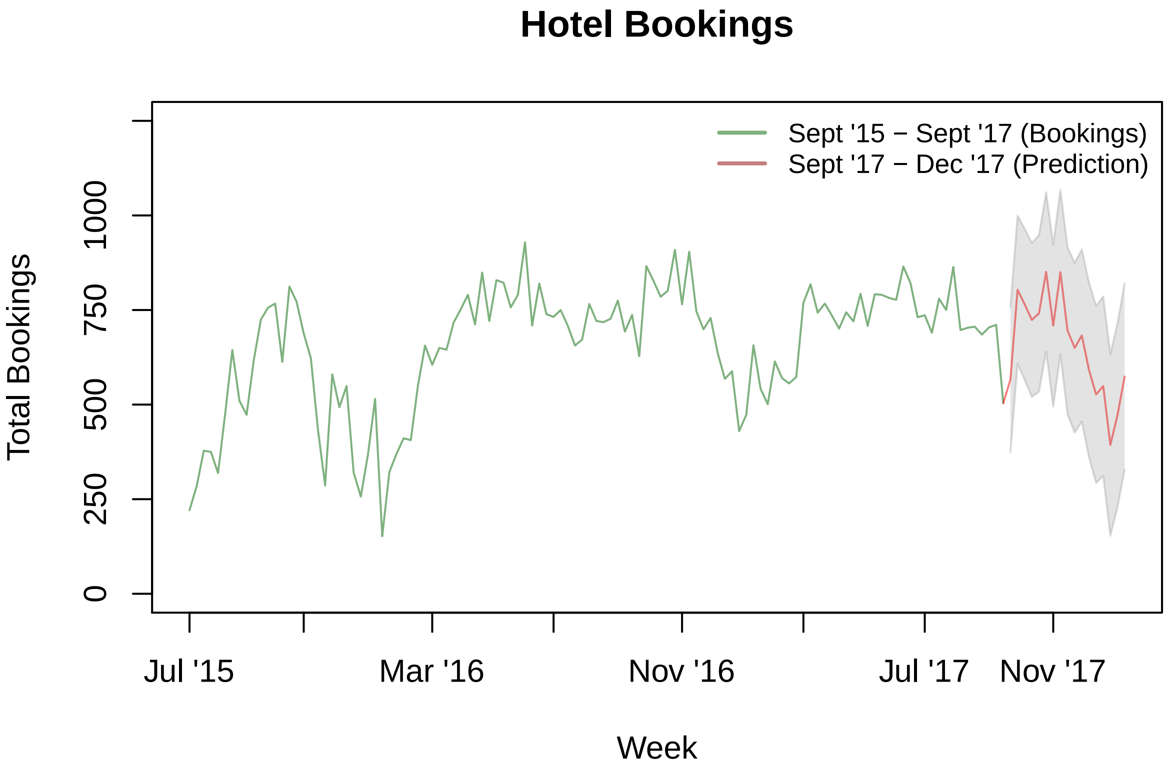
Up until now, we have two models to predict (along with a 95% prediction interval) the hotel demand for the remainder of 2017.

- Holt-Winter Additive
- ARIMA(6,1,0)

It appears that the Holt-Winters Additive model is superior in term of SSE, so that is the model we will choose. Note, we could spend more time finding a better SARIMA model for the time series.

```
bookings.ts <- ts(bookings, frequency=52)
final.model <- HoltWinters(bookings.ts, seasonal="additive")
pred.additive <- predict(final.model, n.ahead=17, prediction.interval = TRUE)
fit <- pred.additive[,1]
lwr <- pred.additive[,2]
upr <- pred.additive[,3]
```

```r
plot(bookings,
     main = "Hotel Bookings",
     xlab = "Week",
     ylab = "Total Bookings",
     xlim = c(1, 132),
     ylim = c(0, 1250),
     type = "l",
     col = adjustcolor("darkgreen", 0.5),
     xaxt = "n",
     yaxt = "n")

lines(y = c(tail(bookings, 1),fit),
      x = 115:132,
      type ="l",
      pch = 19,
      col = adjustcolor("red", 0.5))

lines(y = lwr,
      x = 116:132,
      type ="l",
      pch = 19,
      col = adjustcolor("grey", 0.5))

lines(y = upr,
      x = 116:132,
      type ="l",
      pch = 19,
      col = adjustcolor("grey", 0.5))

polygon(c(116:132, rev(116:132)), c(lwr, rev(upr)), col = "#69696930", border = NA)

axis(side = 1, at = c(1,17,35,52,70,87,104,122),
     labels = c("Jul '15","Nov '15","Mar '16","Jul '16","Nov '16","Mar '17","Jul '17","Nov '17"))

axis(side = 2, at = (250*0:5), labels = c("0","250","500","750","1000", "1250"))

legend("topright",
       lwd = 2,
       bty = "n",
       cex = 0.8,
       col = c(adjustcolor("darkgreen", 0.5), adjustcolor("darkred", 0.5)),
       legend = c("Sept '15 - Sept '17 (Bookings)", "Sept '17 - Dec '17 (Prediction)"),)
```



In the future, we could forecast by hotel vs. resort. Furthermore, instead of weekly forecast, we could forecast by day, month, or quarter and for a specific region.