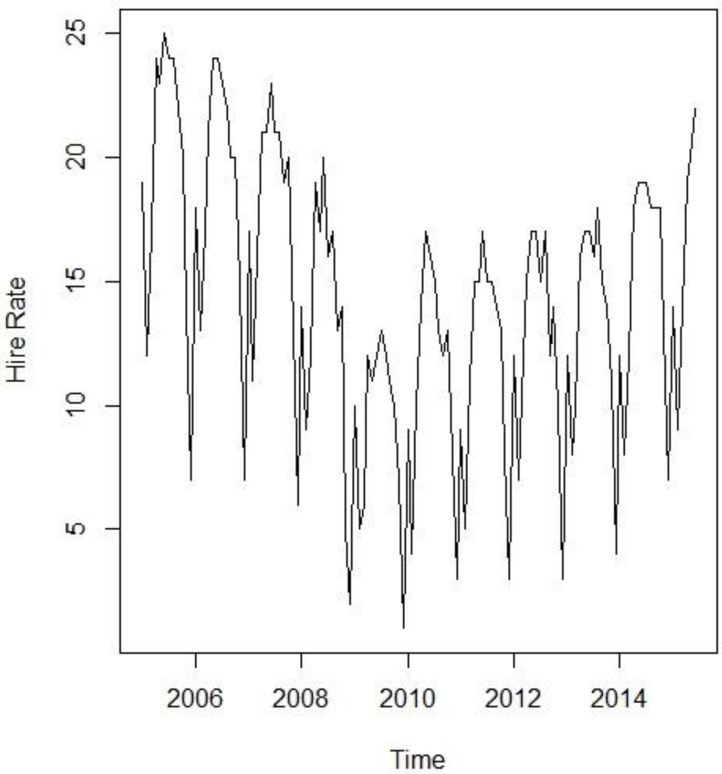
Evgenii Vaniukov
Instructor: Toni Garcia

Forecasting and Time Series Project

In this project, the dataset with the Hire Rates ridership was examined to find the best model to forecast future ridership numbers over four months after the examination period.

Overview of the process

1. The first step is *data partitioning*. The partitioning in forecasting works differently compared to cross-sectional data. In cross-sectional data, the partitioning into training and validation datasets is usually done randomly. However, this approach does not work in time series forecasting because this approach would create two time series with the "holes". Hence, we partition the dataset differently. The series is trimmed into two periods; the earlier period becomes a training dataset and the later period to validation data. Note that the validation dataset is the most important because it carries the most recent information; hence, it is more relevant to predict the future. On the plots that you will e in this project, blue lines indicate the forecast, which is compared to the actual lines below them.
   Here is the plot of hire rates over the period from 2005/01 – 2015/06. The data was partitioned in the way that the validation dataset is equal to two years.



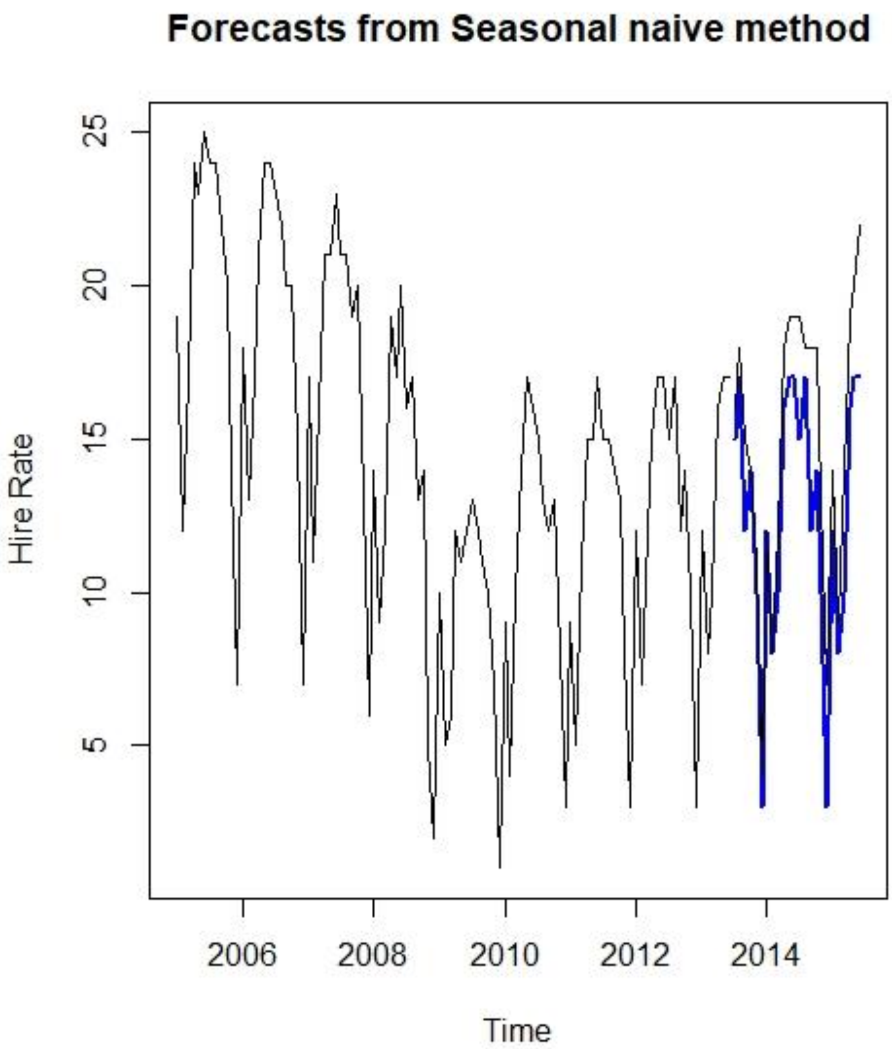
2. The second step is *fitting the forecasting models*. We chose three different models and compared their performances to select the best option to forecast the future period:
   - Naïve Forecast with seasonality (RMSE = 2.84)
   - The linear trend with seasonality (RMSE = 2.48)
   - The quadratic trend with seasonality (RMSE = 1.8)

RMSE stands for Root Mean Square Error. We are going to use it as an evaluation of forecast performances. Essentially, the RMSE measure of how far from the regression line data points are. The lower the error, the better prediction is expected to be in the future. However, the problem of overfitting should be kept in mind.

All forecasts in this project are estimated with the seasonality being included in the model. Seasonality is simply a recurring pattern in a time series. This means that observations, that fall in some seasons, have consistently higher or lower values than those that fall in other seasons.

Naïve Forecast is the simplest model of forecasting. It is used as a performance benchmark to evaluate other models on the same dataset to make sure that more complicated models help to find hidden information instead of overcomplicating things without any value-added. A naïve forecast is simply the most recent value of the series. In other words, at time *t*, our estimate is for any future period t+*K* is simply the value of the series at time *t*. Our Naïve forecast had an *RMSE of 2.84.*

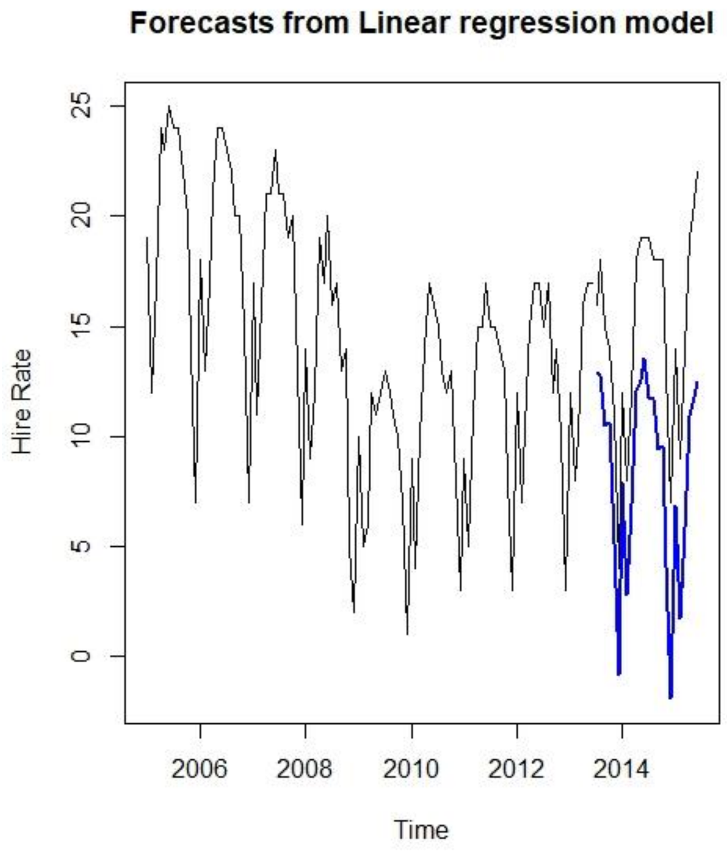
```
Forecast method: Seasonal naive method

Model Information:
Call: snaive(y = train.h, h = Valid, level = 0)

Residual sd: 2.7085

Error measures:
                     ME     RMSE      MAE       MPE     MAPE MASE     ACF1
Training set -0.8888889 2.836273 2.088889 -11.02824 22.00302    1 0.709137

Forecasts:
         Point Forecast Lo 0 Hi 0
Jul 2013             15   15   15
Aug 2013             17   17   17
Sep 2013             12   12   12
Oct 2013             14   14   14
Nov 2013             10   10   10
Dec 2013              3    3    3
Jan 2014             12   12   12
Feb 2014              8    8    8
Mar 2014             10   10   10
Apr 2014             16   16   16
May 2014             17   17   17
Jun 2014             17   17   17
Jul 2014             15   15   15
Aug 2014             17   17   17
Sep 2014             12   12   12
Oct 2014             14   14   14
Nov 2014             10   10   10
Dec 2014              3    3    3
Jan 2015             12   12   12
Feb 2015              8    8    8
Mar 2015             10   10   10
Apr 2015             16   16   16
May 2015             17   17   17
Jun 2015             17   17   17
```



Linear Forecast is a popular forecasting tool, which is based on multiple regression, using suitable predictors to capture the trend and/or seasonality. The model can produce future forecasts by inserting the relevant predictor information into the estimated regression equation. Additionally, a regression model can be used to quantify the correlation between neighboring values in a time series (called autocorrection). Our linear forecast had an *RMSE of 2.48*

```
Forecast method: Linear regression model

Model Information:

Call:
tslm(formula = train.h ~ trend + season)

Coefficients:
(Intercept)        trend      season2      season3      season4      season5      seas
on6      season7      season8      season9     season10     season11
   17.76362     -0.09041     -5.02070     -0.81917      4.38235      5.02832      6.11
874      4.41667      4.38208      2.22249      2.43791     -2.97168
   season12
   -8.88126


Error measures:
                       ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 1.235225e-16 2.474595 1.978149 -6.054011 21.76669 0.9469864 0.8333729

Forecasts:
         Point Forecast       Lo 0       Hi 0
Jul 2013     12.8676471 12.8676471 12.8676471
Aug 2013     12.7426471 12.7426471 12.7426471
Sep 2013     10.4926471 10.4926471 10.4926471
```

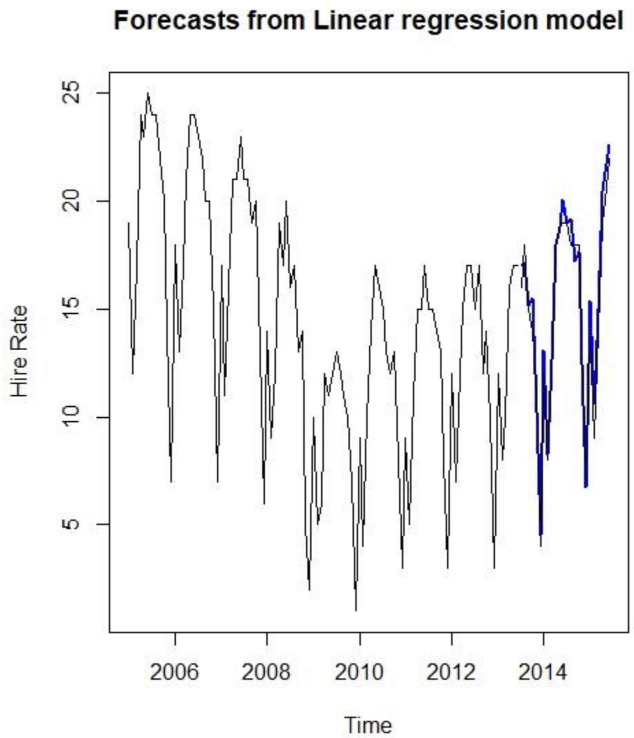
```
Oct 2013      10.6176471 10.6176471 10.6176471
Nov 2013       5.1176471  5.1176471  5.1176471
Dec 2013      -0.8823529 -0.8823529 -0.8823529
Jan 2014       7.9084967  7.9084967  7.9084967
Feb 2014       2.7973856  2.7973856  2.7973856
Mar 2014       6.9084967  6.9084967  6.9084967
Apr 2014      12.0196078 12.0196078 12.0196078
May 2014      12.5751634 12.5751634 12.5751634
Jun 2014      13.5751634 13.5751634 13.5751634
Jul 2014      11.7826797 11.7826797 11.7826797
Aug 2014      11.6576797 11.6576797 11.6576797
Sep 2014       9.4076797  9.4076797  9.4076797
Oct 2014       9.5326797  9.5326797  9.5326797
Nov 2014       4.0326797  4.0326797  4.0326797
Dec 2014      -1.9673203 -1.9673203 -1.9673203
Jan 2015       6.8235294  6.8235294  6.8235294
Feb 2015       1.7124183  1.7124183  1.7124183
Mar 2015       5.8235294  5.8235294  5.8235294
Apr 2015      10.9346405 10.9346405 10.9346405
May 2015      11.4901961 11.4901961 11.4901961
Jun 2015      12.4901961 12.4901961 12.4901961
```



The quadratic trend is a non-linear shape that is easy to fit via linear regression as a polynomial trend. This is done by creating an additional predictor t^2 and fitting a multiple linear regression with two predictors t and t^2. The model was able to capture a U shape of a trend, and that led to a better prediction with an *RMSE of 1.8.*

```
Forecast method: Linear regression model

Model Information:

Call:
tslm(formula = train.h ~ trend + I(trend^2) + season)

Coefficients:
(Intercept)         trend   I(trend^2)      season2      season3      season4      seas
on5      season6      season7      season8      season9     season10
```

```
  21.494689    -0.318350     0.002213    -5.011845    -0.805894     4.395631     5.037
174     6.118736     4.868113     4.842379     2.687219     2.902633
   season11     season12
  -2.511379    -8.429817


Error measures:
                       ME     RMSE      MAE       MPE    MAPE      MASE      ACF1
Training set 2.338475e-18 1.797464 1.398552 -2.72295 13.4309 0.6695195 0.6862318

Forecasts:
         Point Forecast      Lo 0      Hi 0
Jul 2013      17.050166 17.050166 17.050166
Aug 2013      17.164167 17.164167 17.164167
Sep 2013      15.153168 15.153168 15.153168
Oct 2013      15.517169 15.517169 15.517169
Nov 2013      10.256170 10.256170 10.256170
Dec 2013       4.495171  4.495171  4.495171
Jan 2014      13.086853 13.086853 13.086853
Feb 2014       8.241299  8.241299  8.241299
Mar 2014      12.617967 12.617967 12.617967
Apr 2014      17.994634 17.994634 17.994634
May 2014      18.815747 18.815747 18.815747
Jun 2014      20.081303 20.081303 20.081303
Jul 2014      19.019101 19.019101 19.019101
Aug 2014      19.186213 19.186213 19.186213
Sep 2014      17.228326 17.228326 17.228326
Oct 2014      17.645438 17.645438 17.645438
Nov 2014      12.437550 12.437550 12.437550
Dec 2014       6.729663  6.729663  6.729663
Jan 2015      15.374456 15.374456 15.374456
Feb 2015      10.582013 10.582013 10.582013
Mar 2015      15.011792 15.011792 15.011792
Apr 2015      20.441572 20.441572 20.441572
May 2015      21.315795 21.315795 21.315795
Jun 2015      22.634463 22.634463 22.634463
```



<u>Findings</u>: the best model is a Quadratic trend model because it has the lowest RMSE.

3. The third step is *to fit our quadratic model* on the entire dataset to predict the hire rates for the next four months. Here are the estimations that we derived from the quadratic trend with seasonality.

```
4.       Point Forecast      Lo 0      Hi 0
5. Jul 2015       20.99703 20.99703 20.99703
6. Aug 2015       21.27607 21.27607 21.27607
7. Sep 2015       19.45512 19.45512 19.45512
8. Oct 2015       19.73416 19.73416 19.73416
```

Appendix

```
HireRate.df <-read.csv("HireRate.csv", header = T)
View(HireRate.df)
library(forecast)



# Plotting the entire ts
HireRate.ts <- ts(HireRate.df$Hire.rate, start = c(2005,1), end = c(2015,6), frequency = 12)
plot(HireRate.ts,xlab="Time",ylab="Hire Rate")

# Partitioning
Valid <- 24
Training <- length(HireRate.ts)- Valid

train.h <- window(HireRate.ts,start=c(2005,1),end=c(2005,Training))
valid.h <- window(HireRate.ts,start=c(2005,Training+1),end=c(2005,Training+Valid))

# Naïve Forecast with seasonality
naive.forecast <- snaive(train.h, h = Valid, level = 0)
plot(naive.forecast,xlab="Time", ylab ="Hire Rate")
lines(valid.h)
summary(naive.forecast)

# Linear trend with seasonality
HireRate.lm <- tslm(train.h ~ trend + season)
linear.forecast <- forecast(HireRate.lm, h=Valid, level=0)

plot(linear.forecast, xlab="Time", ylab ="Hire Rate")
lines(valid.h)
summary(linear.forecast)

# Quadratic trend with seasonality
HireRate.quad <- tslm(train.h ~ trend + I(trend^2) + season)
quadratic.forecast <- forecast(HireRate.quad, h=Valid, level=0)

plot(quadratic.forecast, xlab="Time", ylab ="Hire Rate")
lines(valid.h)
summary(quadratic.forecast)

#Forecasting the 4-month Hire Rate

HireRate.quad.full <- tslm(HireRate.ts~ trend + I(trend^2) + season)
HireRate.forecast <- forecast(HireRate.quad.full, h = 4, level = 0)

HireRate.forecast
```