**Introduction**

Bike riding is becoming popular as people are using it more and more for recreation. Also, bike riding is becoming part of a healthy lifestyle as well. Even the government promote bike ridership to reduces traffic in major cities. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousand bicycles.

**Data**

The dataset comes from the *UCI machine learning repository* retrieved from: <https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset> . It shows the daily rental activity with parameters such as weather, days, humidity, etc., of Washington D.C. from the year 2011 to 2012. The dataset characteristics are described as the attributes that show us a wide picture of the factors affecting the booking of number of bikes. The following attributes are contained therein:

* **instant:** record index
* **dteday :** date
* **season :** season (1:springer, 2:summer, 3:fall, 4:winter)
* **yr :** year (0: 2011, 1:2012)
* **month :** month ( 1 to 12)
* **holiday:** weather day is holiday or not (extracted from [Web Link])
* **weekday:** day of the week
* **workingday :** if day is neither weekend nor holiday is 1, otherwise is 0.
* **weathersit :**
  + **1:** Clear, Few clouds, Partly cloudy, Partly cloudy
  + **2:** Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  + **3:** Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
* **temp :** Normalized temperature in Celsius. The values are derived via (t-t\_min)/(t\_max-t\_min), t\_min=-8, t\_max=+39 (only in hourly scale)
* **atemp:** Normalized feeling temperature in Celsius. The values are derived via (t-t\_min)/(t\_max-t\_min), t\_min=-16, t\_max=+50 (only in hourly scale)
* **hum:** Normalized humidity. The values are divided to 100 (max)
* **windspeed:** Normalized wind speed. The values are divided to 67 (max)
* **casual:** count of casual users
* **registered:** count of registered users
* **cnt:** count of total rental bikes including both casual and registered

This paper goal is to analyze the dataset and create a predictive model to forecast future ridership. More generally, we’ll seek to answer the following questions:

**Questions**

* How do temperature values change over the seasons? What is mean, standard deviation and median of temperatures for each season?
* For which weather condition the number of total bike rentals are the lowest/highest?
* Is there a correlation between total number of rentals and season? What is the mean, median and standard deviation for total number of rentals (count) per season? Which season is the most popular for the bike rentals?
* Is correlation between felt air temperature (atemp) and number of bike rentals significant? Is there a difference between the correlations for two years (2011 and 2012)?
* Is weather condition correlated to number of bike rentals? What is minimum, maximum, mean, median, standard deviation and number of occurrences for each weather condition? How weather condition influences the distribution of bike rentals?
* Is there a significant difference between total bike rentals on holidays and working days?

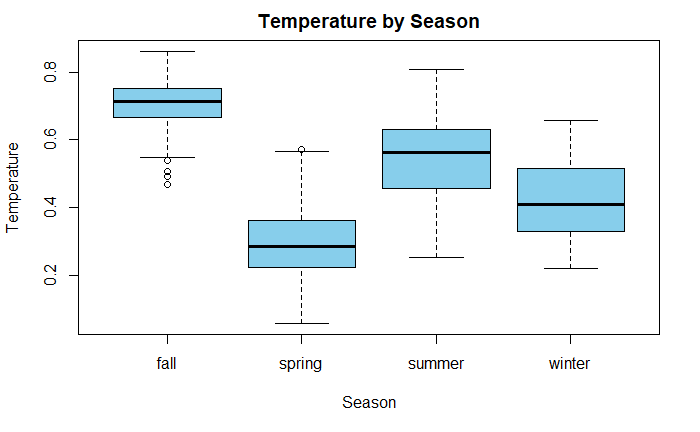
**Analysis**

First, we load up the data into R and view it’s structure as below:

|  |
| --- |
| > data<-read.csv(file.choose(), header=TRUE)  > str(data)  'data.frame': 731 obs. of 16 variables:  $ instant : int 1 2 3 4 5 6 7 8 9 10 ...  $ dteday : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...  $ season : int 1 1 1 1 1 1 1 1 1 1 ...  $ yr : int 0 0 0 0 0 0 0 0 0 0 ...  $ mnth : int 1 1 1 1 1 1 1 1 1 1 ...  $ holiday : int 0 0 0 0 0 0 0 0 0 0 ...  $ weekday : int 6 0 1 2 3 4 5 6 0 1 ...  $ workingday: int 0 0 1 1 1 1 1 0 0 1 ...  $ weathersit: int 2 2 1 1 1 1 2 2 1 1 ...  $ temp : num 0.344 0.363 0.196 0.2 0.227 ...  $ atemp : num 0.364 0.354 0.189 0.212 0.229 ...  $ hum : num 0.806 0.696 0.437 0.59 0.437 ...  $ windspeed : num 0.16 0.249 0.248 0.16 0.187 ...  $ casual : int 331 131 120 108 82 88 148 68 54 41 ...  $ registered: int 654 670 1229 1454 1518 1518 1362 891 768 1280 ...  $ cnt : int 985 801 1349 1562 1600 1606 1510 959 822 1321 ... |

The dataset contains 731 observations/ rows and 16 features/ variables which are a combination of factor, integer and numerical data types.

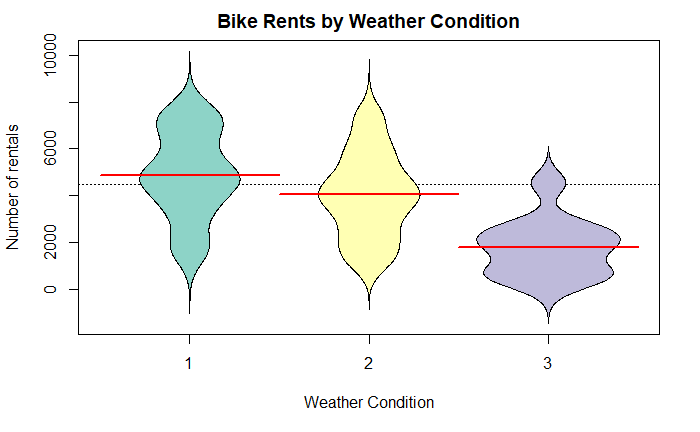
Next we do some data transformations and cleanup to make it suitable for analysis.



As it can be seen from the analysis and from the above boxplots, the lowest minimum temperature as well as the minimum mean temperature applies to spring. Maximum temperature as well as the maximum mean values were experienced during fall.

Next, we would like to know during which weather conditions was the bike rentals highest.

We use a beanplot to for our visualization

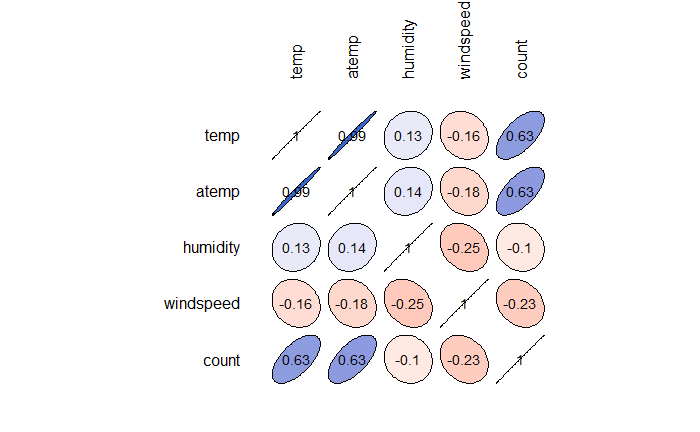


The beanplot demonstrates that the lowest number of rents is typical for the 3rdweather type (rain, thunderstorm etc.) while the highest mean value of rentals have days with the 1st weather type (clear, partly cloudy etc.)

The Analysis of the variance model demonstrates that number of rents and season are significantly correlated with a p-value of 2e-16.

We obtain the correlation coefficients between the variables from the following code output.

The correlation coefficient ranges from -1 to 1. The sign of the correlation coefficient indicates the direction of the association. **The magnitude of the correlation coefficient indicates the strength of the association.**

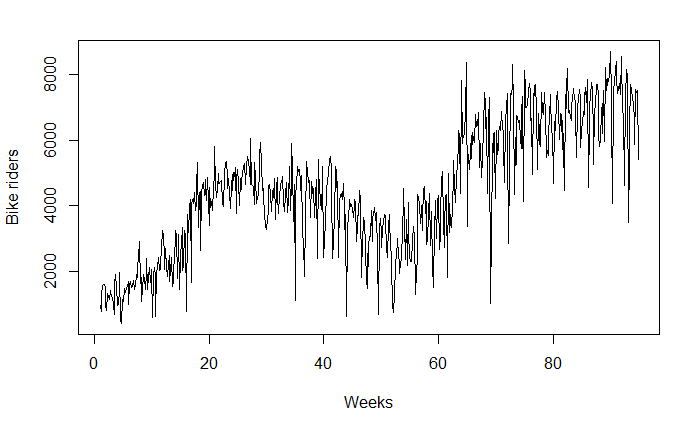


In the initial part, we load the data in R and extract the trend, seasonality and error by decomposition.

Splitting data into training and testing sets. We take 90% of data for the training set and rest

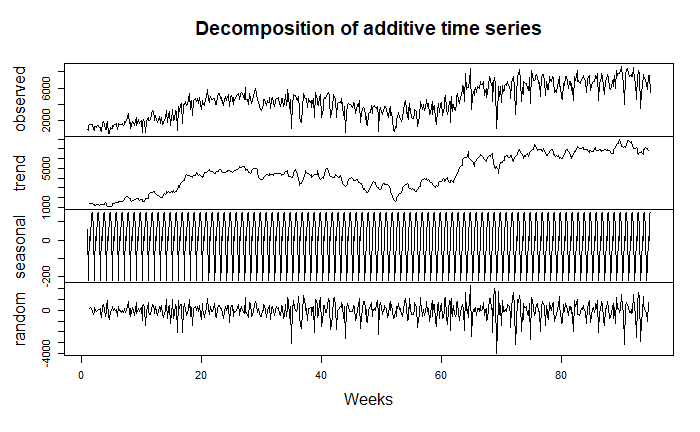
10% for the testing. The plot below shows the trend of the data. The time series described an

additive model since the random ﬂuctuations in the data are roughly constant in size over time.



The decomposition plot below shows the original time series on top, the estimated trend

component, the estimated seasonal component, and the estimated remainder component. We can see, there is a very strong seasonality over the weeks.



We see that the estimated trend component shows a gradual increase from about 1000 from

the first week to about 6000.

Since we have an additive model, we can seasonally adjust the time series by estimating the

seasonal component and subtracting the estimated seasonal component from the original time series.

Fitting an ARIMA model requires the series to be stationary (Dalinina, n.d.). We check for the

stationarity of the series by using the ADF test (Augmented Dickey-Fuller). The test shows that

the p-value is .19 which is less than the 0.05 threshold. This indicates series is non-stationary.

Here are the ACF and PACF plots for the series. ACF is a plot of total correlation between

different lag functions (correlation of x(t) with x(t-1) , x(t-2) and so on).

For an MA series (looking at ACF), if the total correlation chart cuts off at nth lag, our lag is nth

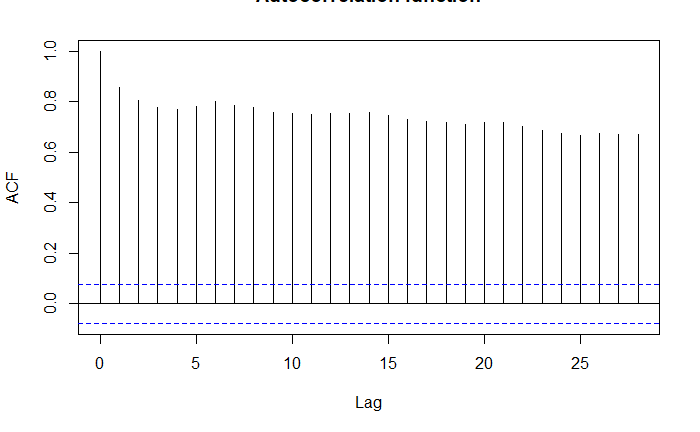
for MA series. However, if the correlation gradually goes down without a cutoff, we have

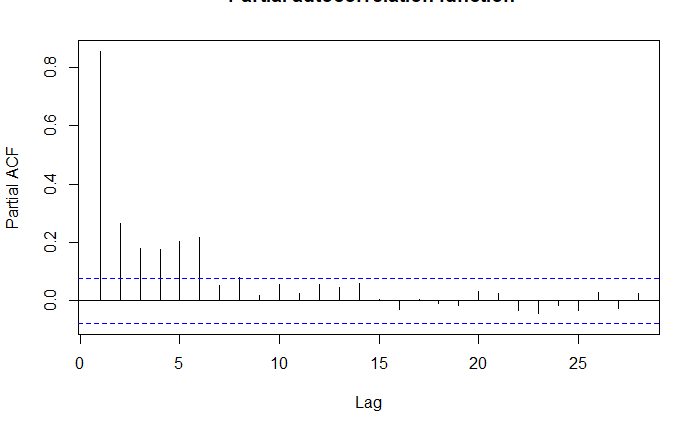
MA(0).

We can estimate the seasonal MA from ACF and AR from PACF for this series. Looking at the

plots we can say that this is an ‘Auto Regressive’ (AR) type of series. Thus, the order will be

AR(6) and MA(0).





As we can see, the decay of ACF chart is very slow, which means that the population is not

stationary.

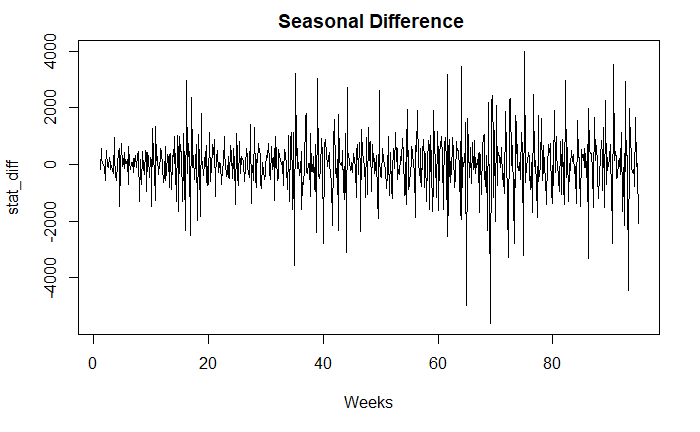
Differencing method is one of the methods to convert non-stationary series to the stationary one. Usually, non-stationary series can be corrected by a simple transformation; differencing.

Differencing the series can help in removing its trend or cycles.

The idea behind differencing is that, if the original data series does not have constant properties over time, then the change from one period to another might (Dalinina, n.d.). After plotting

the differenced series, we see an oscillating pattern and no visible strong trend. This suggests at

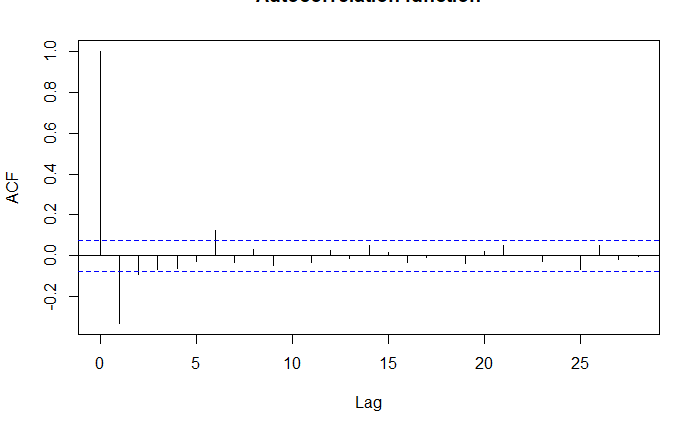
differencing of order 1 terms is enough and should be included in the model.

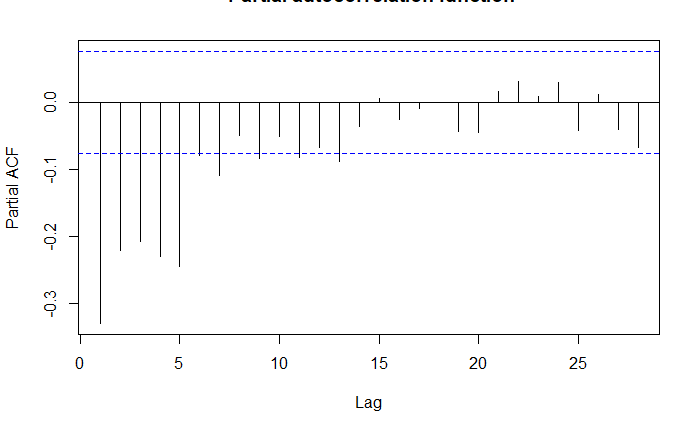


We can estimate the non-seasonal MA from ACF and AR from PACF for this series. Looking at

the plots we can say that this is an ‘Auto Regressive’(AR) type of series. Thus, the order will be

AR (7) and MA (0).





We fitted a model with the order (7,1,0) and seasonal (6,1,0) with period 7 weeks. Forecasts

from the model for the next 10 weeks are also shown in the plot below. The forecasts follow

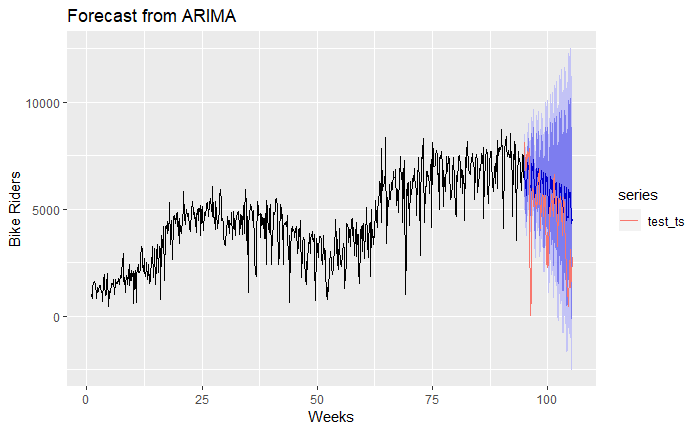
the recent trend in the data, because of the differencing. We can see that the forecast is not

very close to the actual data (blue line).

We can see the model captures the general trend well but is not able to capture sharp ups and downs. It is probably not the best model for decision making on prediction about the number of bikes sharing users.

The few key reasons this model doesn’t perform well are that the model considers only time as a factor, and no other crucial features like whether it was working day or holiday, sunny day or

rainy day etc. weather-related parameters into account.



The output from fitarima() function includes the fitted coefficients and the standard error (s.e.) for each coefficient. Observing the coefficients, we can exclude the insignificant ones. We can

use the function confint() for this purpose.

This is a recursive process and we need to run this arima() function with different (p,d,q) values to find out the most optimized and efficient model.

We use the auto.arima() and ets() functions in the forecast package for the automatic selection of exponential and ARIMA models.

The auto.arima() function in R uses a combination of unit root tests, minimization of the AIC

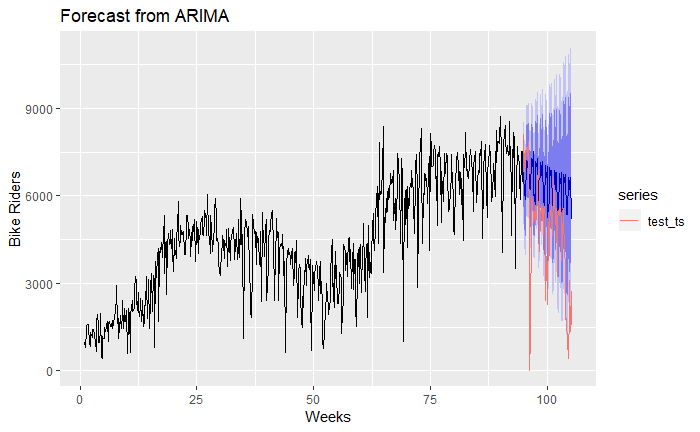
and MLE to obtain an ARIMA model.

The p,d and q are then chosen by minimizing the AIC. The algorithm uses a stepwise search to

traverse the model space to select the best model with smallest AIC.

As we observe, the best ARIMA model would be of order (0,1,3). Predicting using these

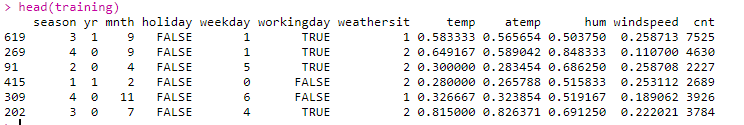
parameters shows a closer fitting model with a high fidelity to the actual data as shown;



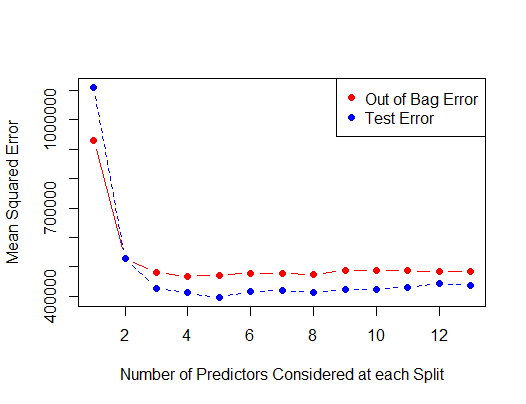
**Random Forest Modeling:**

In simple words, A random forest, is making a bunch of decision trees, and consulting all of them at once to make your decision. In the random forest approach, many decision trees are created. Every observation is fed into every decision tree. The most common outcome for each observation is used as the final output.

Here, in “day” dataset first we have first loaded the dataset and performed cleansing and manipulation operations. After that it is divided into training and testing data set.

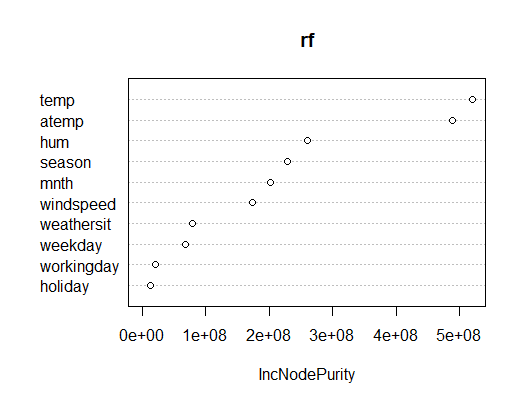


In above picture we can see the glimpse of training dataset. In order to create better random forest model, we have taken the difference in Out of Bag Error and Testing Error on different mtry values. Mtry value is number of variables available for splitting at each tree node. Which is plotted in below picture.



We can notice that at mtry=5, we got minimum number of errors. So, that would be our best model in this case. After putting mtry=5,

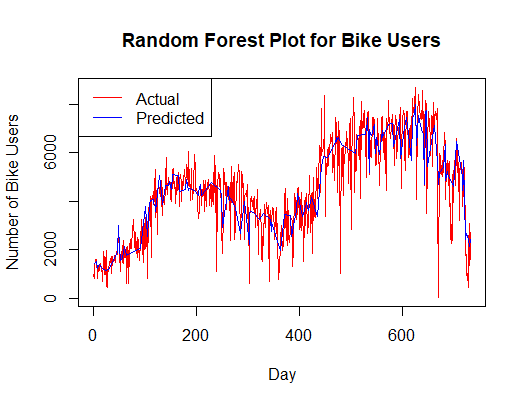
Here, the graph below showing the importance of every attribute which is used by new random forest model.



In that it is noticeable that the weather factors such as temperature, humidity, season of the year, etc. are used most in model. Which can also support by fact that people’s mood of riding bikes is very dependent on weather outside.

Now, let’s predict the value using this model,

The line graph below showing the predicted and actual values of total customers who rented the bike by day.



Now, I used mean absolute error (MAE) to evaluate the model. mean absolute error (MAE) is a measure of difference between two continuous variables.

Here, we got the mean absolute error 479.629 which is quite good. It means that the average difference between actual values and predicted value is about 480.