Name; Chitipolu Sri Sudheera

Tool: R studio

Washington, D.C Capital Bikeshare Demand

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city. In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

.

Acknowledgements

Kaggle is hosting this competition for the machine learning community to use for fun and practice. This dataset was provided by Hadi Fanaee Tork using data from Capital Bikeshare. We also thank the UCI machine learning repository for hosting the dataset. If you use the problem in publication, please cite:
Fanaee-T, Hadi, and Gama, Joao, *Event labeling combining ensemble detectors and background knowledge*, Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg.

Data

You are provided hourly rental data spanning two years. For this competition, the training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month. You must predict the total count of bikes rented during each hour covered by the test set, using only information available prior to the rental period.

Data Fields

datetime - hourly date + timestamp
season - 1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday - whether the day is considered a holiday
workingday - whether the day is neither a weekend nor holiday
weather - 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
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp - temperature in Celsius
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated
registered - number of registered user rentals initiated
count - number of total rentals

Step 1. Hypothesis Generation

- **Hourly trend**: There must be high demand during office timings. Early morning and late evening can have different trend (cyclist) and low demand during 10:00 pm to 4:00 am.
- **Daily Trend:** Registered users demand more bike on weekdays as compared to weekend or holiday.
- **Rain:** The demand of bikes will be lower on a rainy day as compared to a sunny day. Similarly, higher humidity will cause to lower the demand and vice versa.
- **Temperature:** In India, temperature has negative correlation with bike demand. But, after looking at Washington's temperature graph, I presume it may have positive correlation.
- **Pollution:** If the pollution level in a city starts soaring, people may start using Bike (it may be influenced by government / company policies or increased awareness).
- **Time:** Total demand should have higher contribution of registered user as compared to casual because registered user base would increase over time.
- **Traffic:** It can be positively correlated with Bike demand. Higher traffic may force people to use bike as compared to other road transport medium like car, taxi etc

2. Understanding the Data Set

Training data set has 12 variables (see below) and Test has 9 (excluding registered, casual and count).

Independent Variables

```
datetime:   date and hour in "mm/dd/yyyy hh:mm" format

season:     Four categories-> 1 = spring, 2 = summer, 3 = fall, 4 = winter

holiday:    whether the day is a holiday or not (1/0)

workingday: whether the day is neither a weekend nor holiday (1/0)

weather:    Four Categories of weather

            1-> Clear, Few clouds, Partly cloudy, Partly cloudy
            2-> Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
            3-> Light Snow and Rain + Thunderstorm + Scattered clouds, Light Rain + S
cattered clouds
            4-> Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp:       hourly temperature in Celsius
atemp:      "feels like" temperature in Celsius
humidity:   relative humidity
windspeed:  wind speed
```

Dependent Variables

```
registered: number of registered user
casual:     number of non-registered user
count:      number of total rentals (registered + casual)
```

3. Importing Data set and Basic Data Exploration

1. Import Train and Test Data Set

```
setwd("E:/kaggle data/bike sharing")
```

```
train=read.csv("train_bike.csv")
```

```
test=read.csv("test_bike.csv")
```

4. Combine both Train and Test Data set (to understand the distribution of independent variable together).

```
test$registered=0

test$casual=0

test$count=0
```

```
data=rbind(train,test)
```

Before combing test and train data set, I have made the structure similar for both.

8. Variable Type Identification

```
str(data)

'data.frame':  17379 obs. of  12 variables:

$ datetime  : Factor w/ 17379 levels "2011-01-01 00:00:00",..: 1 2 3 4 5 6 7 8 9 10 ...

$ season    : int  1 1 1 1 1 1 1 1 1 1 ...

$ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
$ workingday: int  0 0 0 0 0 0 0 0 0 0 ...

$ weather   : int  1 1 1 1 1 2 1 1 1 1 ...

$ temp      : num  9.84 9.02 9.02 9.84 9.84 ...

$ atemp     : num  14.4 13.6 13.6 14.4 14.4 ...

$ humidity  : int  81 80 80 75 75 75 80 86 75 76 ...

$ windspeed : num  0 0 0 0 0 ...

$ casual    : num  3 8 5 3 0 0 2 1 1 8 ...

$ registered: num  13 32 27 10 1 1 0 2 7 6 ...
```

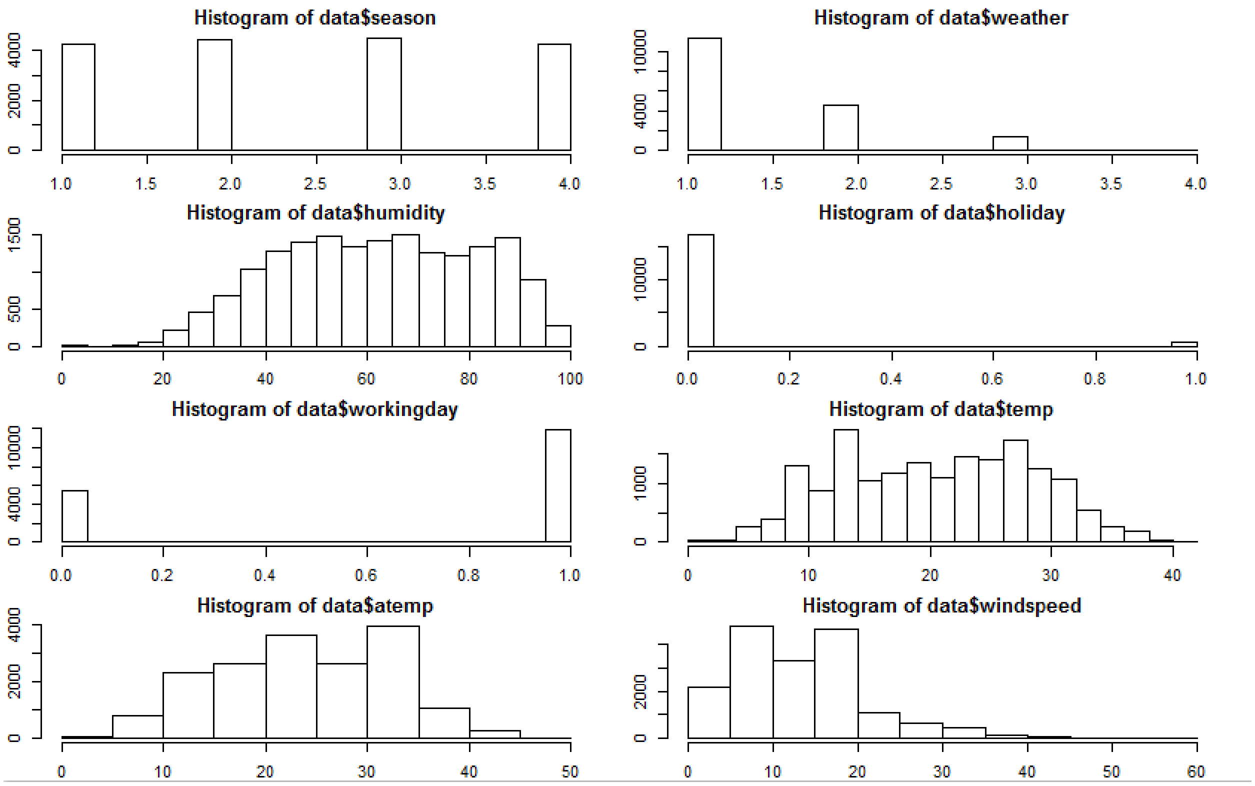
```
$ count     : num  16 40 32 13 1 1 2 3 8 14 ...
```

```
Summary(data)
```

Understand the distribution of numerical variables and generate a frequency table for numeric variables. Now, I’ll test and plot a histogram for each numerical variables and analyze the distribution.

```
par(mfrow=c(4,2))


par(mar = rep(2, 4))
```

```
hist(data$season)

hist(data$weather)

hist(data$humidity)

hist(data$holiday)

hist(data$workingday)

hist(data$temp)

hist(data$atemp)

hist(data$windspeed)
```



1. Few inferences can be drawn by looking at the these histograms:

    - Season has four categories of almost equal distribution
    - Weather 1 has higher contribution i.e. mostly clear weather.

```
○  prop.table(table(data$weather))


○  1     2      3      4
```

```
0.66  0.26   0.08   0.00
```

    - As expected, mostly working days and variable holiday is also showing a similar inference. You can use the code above to look at the distribution in detail. Here you can generate a variable for weekday using holiday and working day. Incase, if both have zero values, then it must be a working day.
    - Variables temp, atemp, humidity and windspeed looks naturally distributed.

2. Convert discrete variables into factor (season, weather, holiday, workingday)

```
3.  data$season=as.factor(data$season)


4.  data$weather=as.factor(data$weather)


5.  data$holiday=as.factor(data$holiday)
```

```
data$workingday=as.factor(data$workingday)
```

4. Hypothesis Testing (using multivariate analysis)

Till now, we have got a fair understanding of the data set. Now, let’s test the hypothesis which we had generated earlier. Here I have added some additional hypothesis from the dataset. Let’s test them one by one:

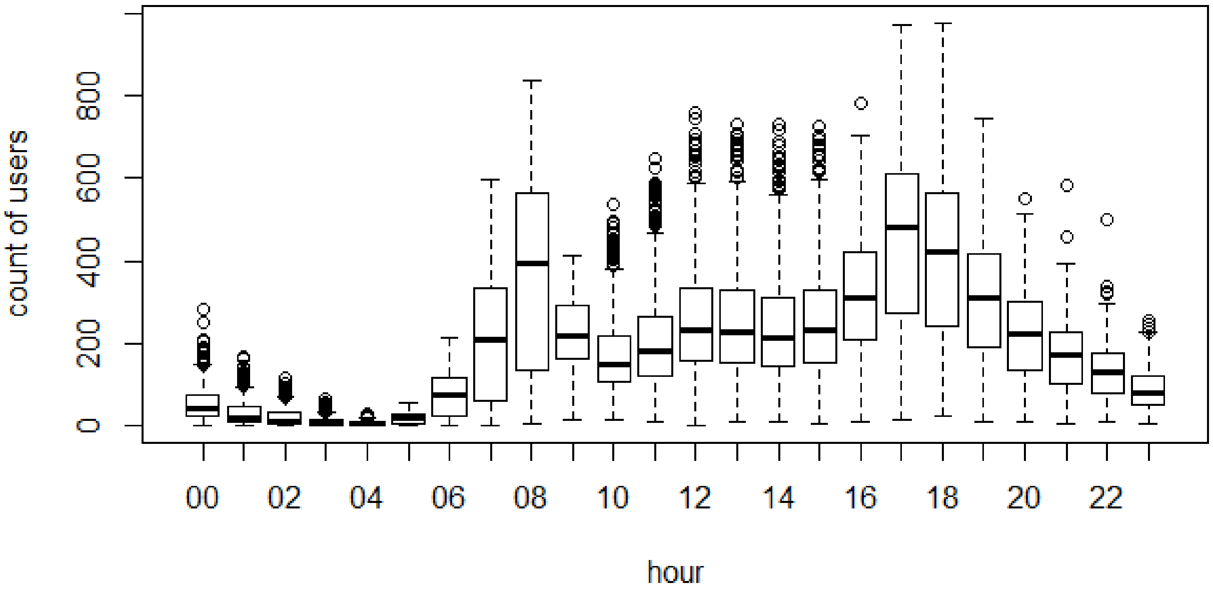
- **Hourly trend**: We don’t have the variable ‘hour’ with us right now. But we can extract it using the datetime column.

```
data$hour=substr(data$datetime,12,13)
```

```
data$hour=as.factor(data$hour)
```

Let’s plot the hourly trend of count over hours and check if our hypothesis is correct or not. We will separate train and test data set from combined one.

```
train=data[as.integer(substr(data$datetime,9,10))<20,]


test=data[as.integer(substr(data$datetime,9,10))>19,]




boxplot(train$count~train$hour,xlab="hour", ylab="count of users")
```

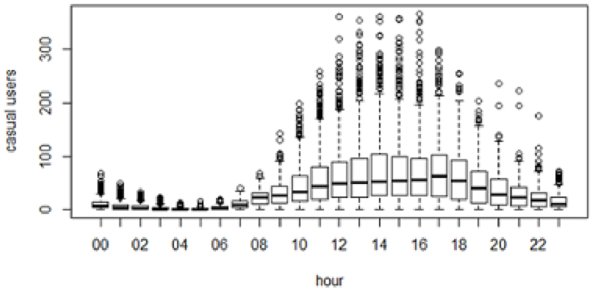


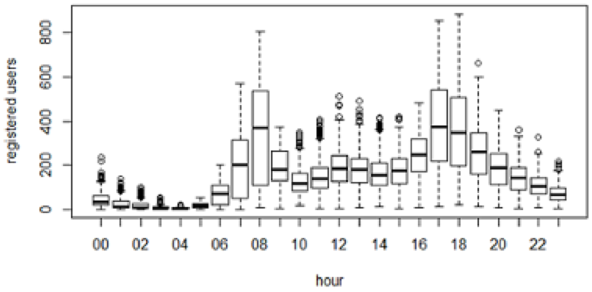
Above, you can see the trend of bike demand over hours. Quickly, I'll segregate the bike demand in three categories:

- High : 7-9 and 17-19 hours
- Average : 10-16 hours
- Low : 0-6 and 20-24 hours

Here I have analyzed the distribution of total bike demand. Let's look at the distribution of registered and casual users separately.

```
boxplot(train$registered~train$hour,xlab='hour',ylab='count of users')




boxplot(train$casual~train$hour,xlab='hour',ylab='count of users')
```

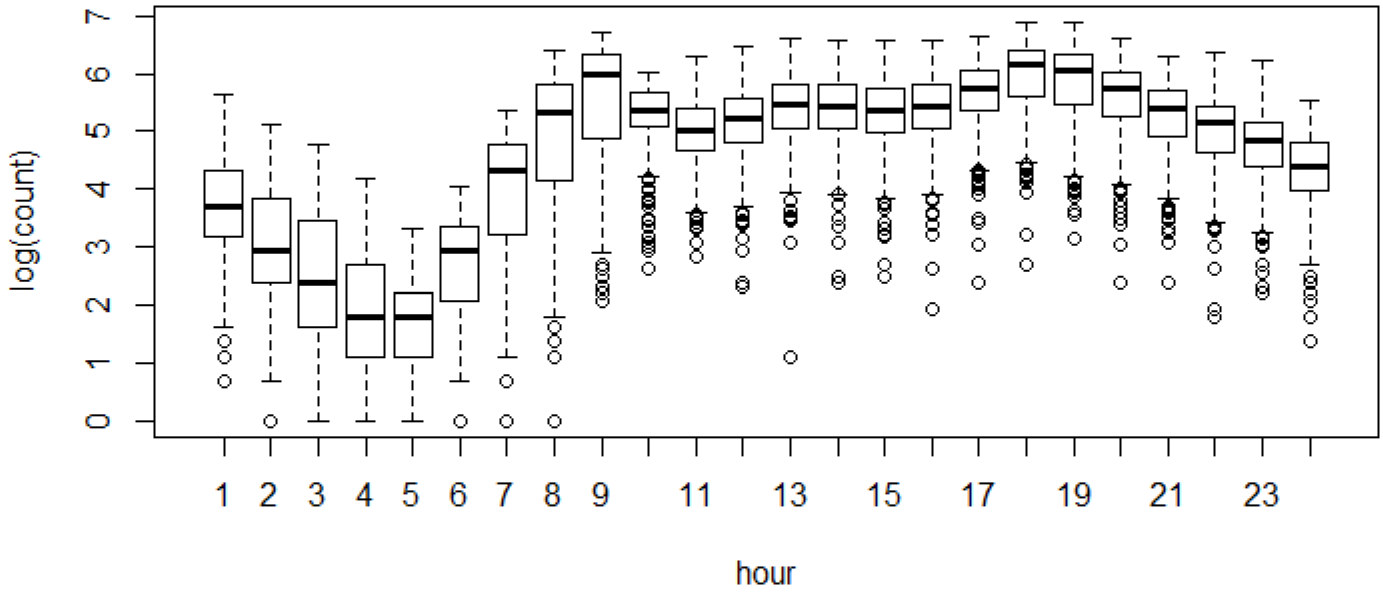




Above you can see that registered users have similar trend as count. Whereas, casual users have different trend. Thus, we can say that ‘hour’ is significant variable and our hypothesis is ‘true’.

You might have noticed that there are a lot of outliers while plotting the count of registered and casual users. These values are not generated due to error, so we consider them as natural outliers. They might be a result of groups of people taking up cycling (who are not registered). To treat such outliers, we will use logarithm transformation. Let’s look at the similar plot after log transformation.

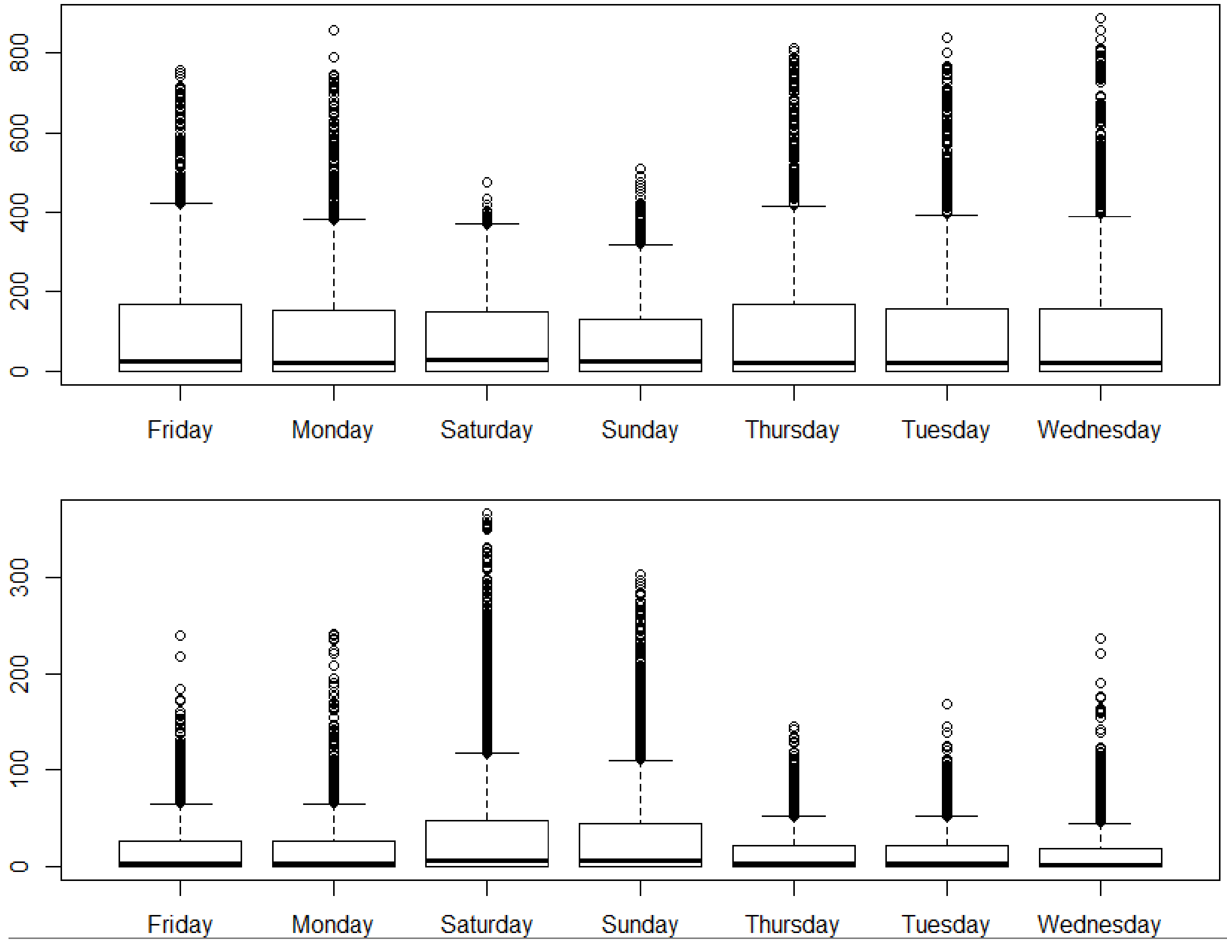
```
boxplot(log(train$count)~train$hour,xlab="hour",ylab="log(count)")
```



Daily Trend: Like Hour, we will generate a variable for day from datetime variable and after that we’ll plot it.

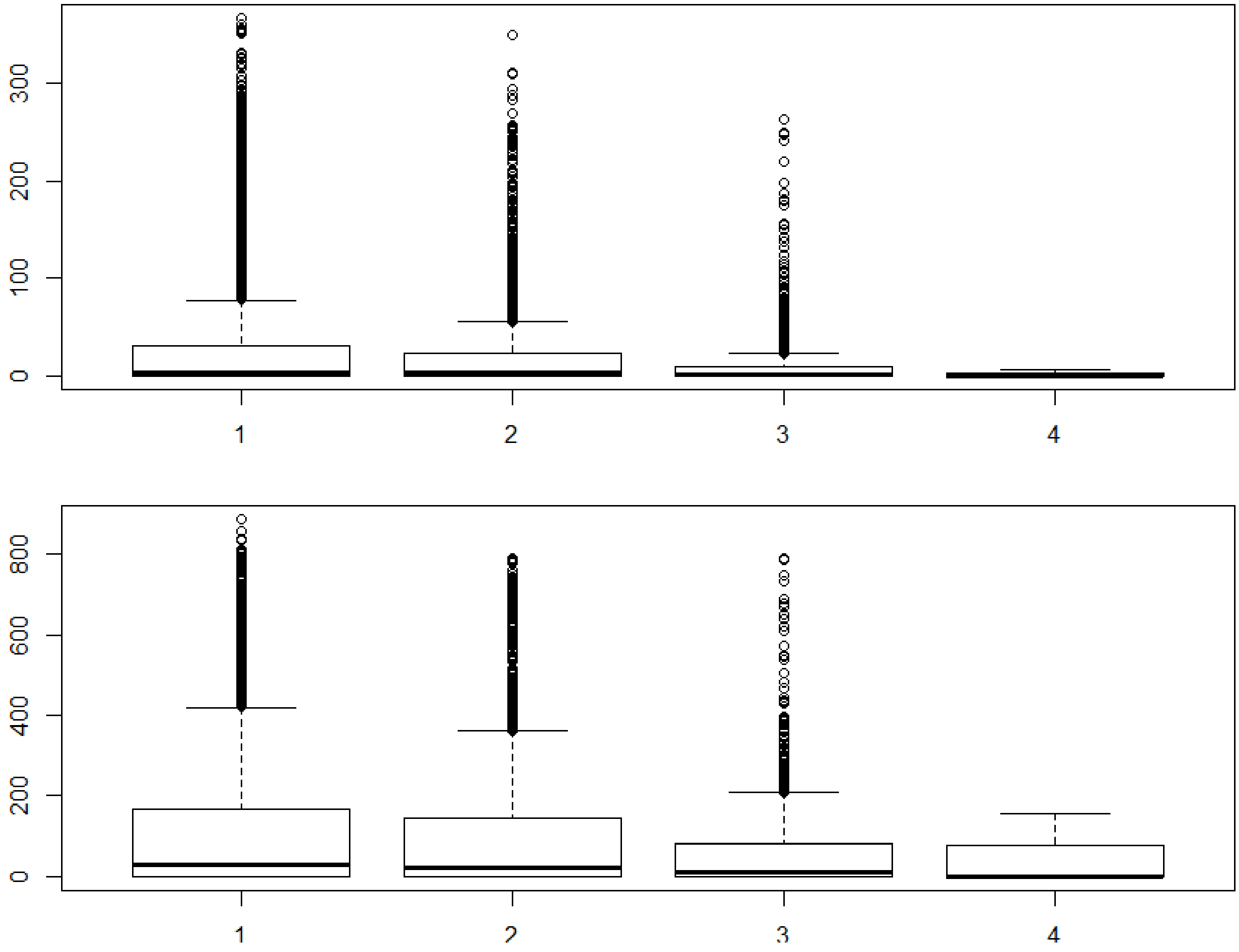
```
date=substr(data$datetime,1,10)

days<-weekdays(as.Date(date))

data$day=days
```

Plot shows registered and casual users' demand over days.



- While looking at the plot, I can say that the demand of causal users increases over weekend.

- **Rain:** We don’t have the ‘rain’ variable with us but have ‘weather’ which is sufficient to test our hypothesis. As per variable description, weather 3 represents light rain and weather 4 represents heavy rain. Take a look at the plot:

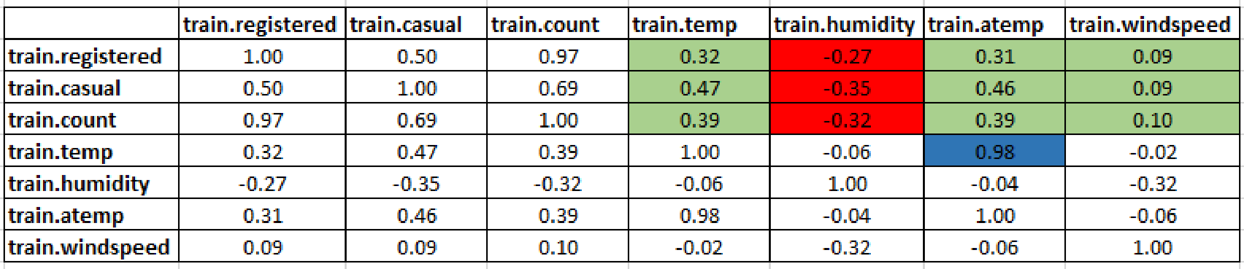


- It is clearly satisfying our hypothesis.

- **Temperature, Windspeed and Humidity:** These are continuous variables so we can look at the correlation factor to validate hypothesis.

```
sub=data.frame(train$registered,train$casual,train$count,train$temp,train$humidity,train$atemp,train$windspeed)
```

```
cor(sub)
```



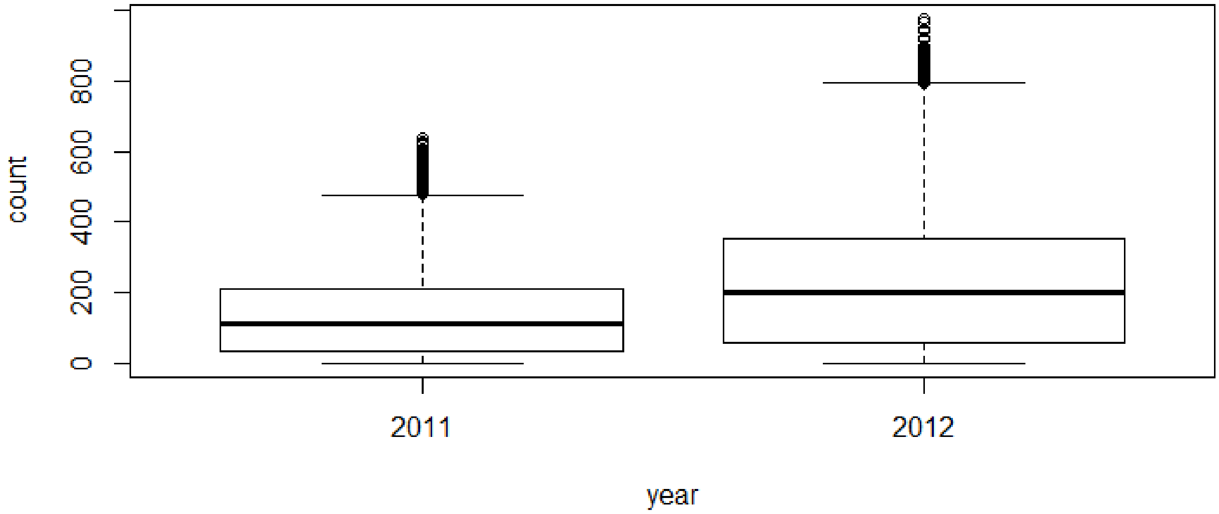
	train.registered	train.casual	train.count	train.temp	train.humidity	train.atemp	train.windspeed
train.registered	1.00	0.50	0.97	0.32	-0.27	0.31	0.09
train.casual	0.50	1.00	0.69	0.47	-0.35	0.46	0.09
train.count	0.97	0.69	1.00	0.39	-0.32	0.39	0.10
train.temp	0.32	0.47	0.39	1.00	-0.06	0.98	-0.02
train.humidity	-0.27	-0.35	-0.32	-0.06	1.00	-0.04	-0.32
train.atemp	0.31	0.46	0.39	0.98	-0.04	1.00	-0.06
train.windspeed	0.09	0.09	0.10	-0.02	-0.32	-0.06	1.00

- Here are a few inferences you can draw by looking at the above histograms:
  - Variable temp is positively correlated with dependent variables (casual is more compare to registered)
  - Variable atemp is highly correlated with temp.
  - Windspeed has lower correlation as compared to temp and humidity

- **Time:** Let's extract year of each observation from the datetime column and see the trend of bike demand over year.

```
data$year=substr(data$datetime,1,4)

data$year=as.factor(data$year)

train=data[as.integer(substr(data$datetime,9,10))<20,]

test=data[as.integer(substr(data$datetime,9,10))>19,]
```

```
boxplot(train$count~train$year,xlab="year", ylab="count")
```



- You can see that 2012 has higher bike demand as compared to 2011.

- **Pollution & Traffic:** We don't have the variable related with these metrics in our data set so we cannot test this hypothesis.

5. Feature Engineering

In addition to existing independent variables, we will create new variables to improve the prediction power of model. Initially, you must have noticed that we generated new variables like hour, month, day and year.

Here we will create more variables, let's look at the some of these:

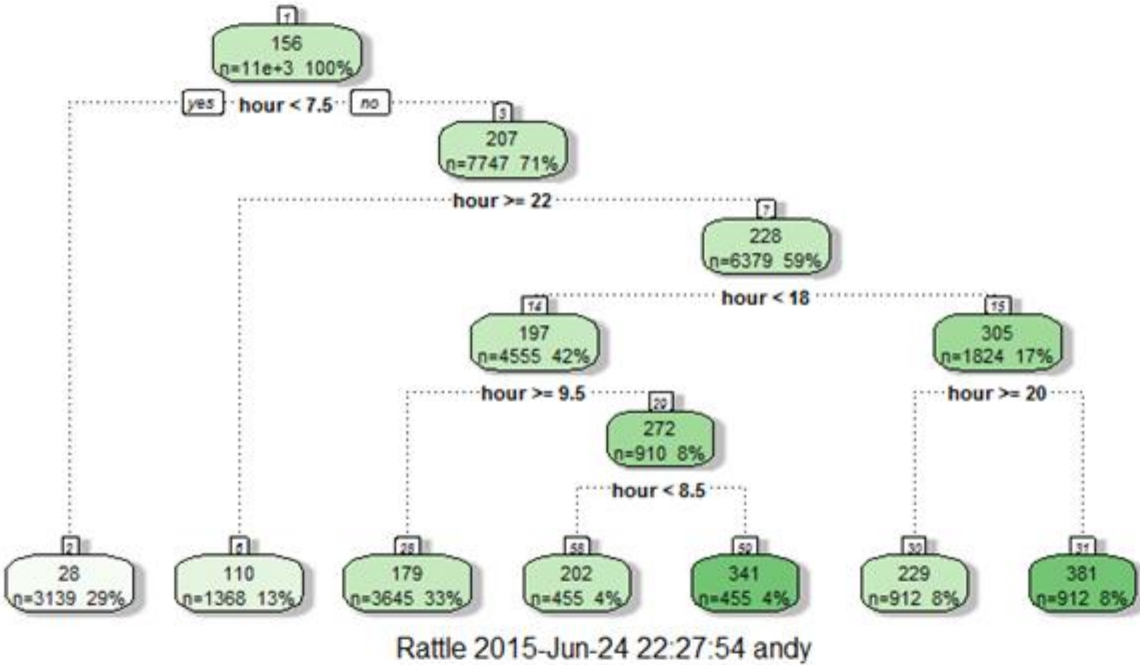
- **Hour Bins:** Initially, we have broadly categorize the hour into three categories. Let's create bins for the hour variable separately for casual and registered users. Here we will use decision tree to find the accurate bins.

```
• train$hour=as.integer(train$hour) # convert hour to integer
```

```
test$hour=as.integer(test$hour) # modifying in both train and test data set
```

We use the library rpart for decision tree algorithm.

```
library(rpart)

library(rattle) #these libraries will be used to get a good visual plot for the decision tree model.

library(rpart.plot)

library(RColorBrewer)

d=rpart(registered~hour,data=train)

fancyRpartPlot(d)
```

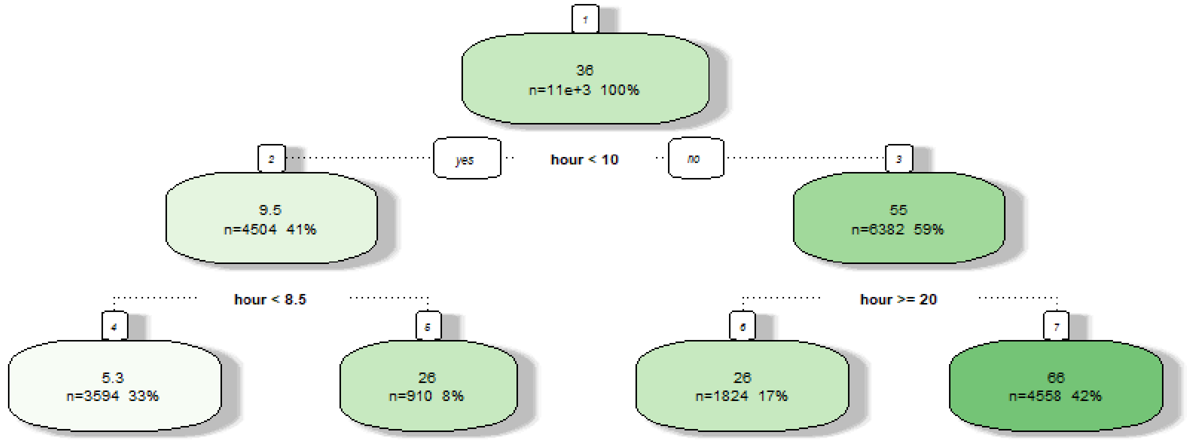


Now, looking at the nodes we can create different hour bucket for registered users.

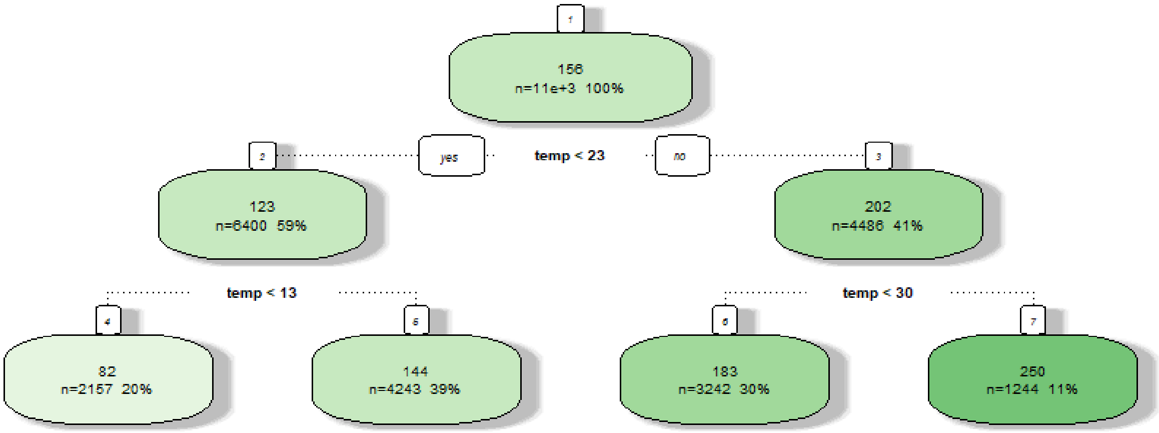
```
data=rbind(train,test)
```

```
data$dp_reg=0

data$dp_reg[data$hour<8]=1

data$dp_reg[data$hour>=22]=2

data$dp_reg[data$hour>9 & data$hour<18]=3

data$dp_reg[data$hour==8]=4

data$dp_reg[data$hour==9]=5

data$dp_reg[data$hour==20 | data$hour==21]=6

data$dp_reg[data$hour==19 | data$hour==18]=7
```

```
c=rpart(registered~hour,data=train)

fancyRpartPlot(c)
```

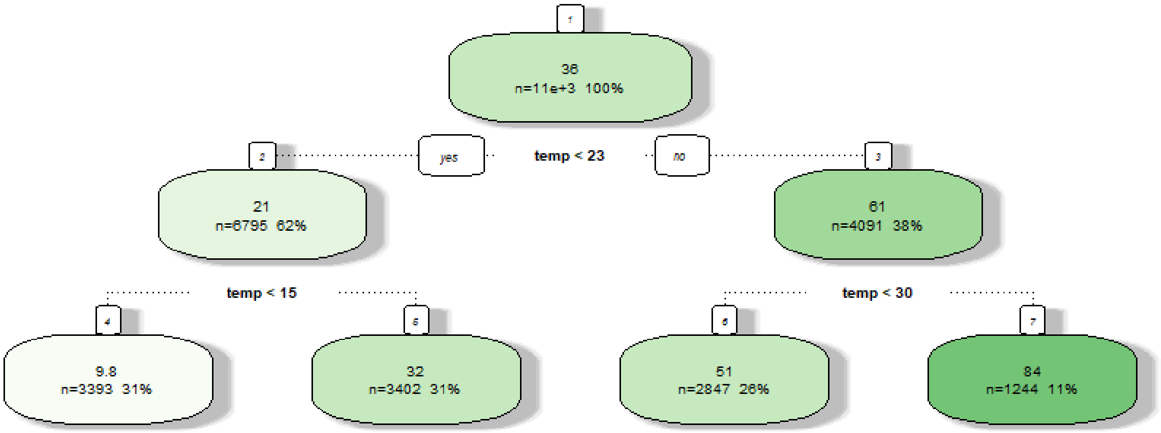


```
data$dp_cas=0

data$dp_cas[data$hour<=8]=1

data$dp_cas[data$hour==9]=2

data$dp_cas[data$hour>=10 & data$hour<=19]=3

data$dp_cas[data$hour>19]=4

d = rpart(registered~temp,data=train)

fancyRpartPlot(f)
```



```
data$temp_reg=0
data$temp_reg[data$temp<13]=1
data$temp_reg[data$temp>=13 & data$temp<23]=2
data$temp_reg[data$temp>=23 & data$temp<=19]=3
data$temp_reg[data$temp>=30]=4
```

```
e = rpart(casual~temp,data=train)
fancyRpartPlot(e)
```



```
data$temp_cas=0

data$temp_cas[data$temp<15]=1

data$temp_cas[data$temp>=15 & data$temp<23]=2

data$temp_cas[data$temp>=23 & data$temp<13]=3

data$temp_cas[data$temp>=30]=4
```

- **Year Bins:** We had a hypothesis that bike demand will increase over time and we have proved it also. Here I have created 8 bins (quarterly) for two years. Jan-Mar 2011 as 1 …..Oct-Dec2012 as 8.

```
•  data$year_part[data$year=='2011']=1

•  data$year_part[data$year=='2011' & data$month>3]=2

•  data$year_part[data$year=='2011' & data$month>6]=3
```

```
data$year_part[data$year=='2011' & data$month>9]=4

data$year_part[data$year=='2012']=5

data$year_part[data$year=='2012' & data$month>3]=6

data$year_part[data$year=='2012' & data$month>6]=7

data$year_part[data$year=='2012' & data$month>9]=8
```

```
table(data$year_part)
```

- **Day Type:** Created a variable having categories like “weekday”, “weekend” and “holiday”.

```
data$day_type=""

data$day_type[data$holiday==0 & data$workingday==0]="weekend"

data$day_type[data$holiday==1]="holiday"
```

```
data$day_type[data$holiday==0 & data$workingday==1]="working day"
```

- **Weekend:** Created a separate variable for weekend (0/1)

```
data$weekend=0
```

```
data$weekend[data$day=="Sunday" | data$day=="Saturday" ]=1
```

6. Model Building

As this was our first attempt, we applied decision tree, conditional inference tree and random forest algorithmsand found that random forest is performing the best. You can also go with regression, boosted regression, neural network and find which one is working well for you.

Before executing the random forest model code, I have followed following steps:

- Convert discrete variables into factor (weather, season, hour, holiday, working day, month, day)

```
train$hour=as.factor(train$hour)


test$hour=as.factor(test$hour)
```

- As we know that dependent variables have natural outliers so we will predict log of dependent variables.
- Predict bike demand registered and casual users separately.
  y1=log(casual+1) and y2=log(registered+1), Here we have added 1 to deal with zero values in the casual and registered columns.

```
#predicting the log of registered users.


set.seed(415)


fit1 <- randomForest(logreg ~ hour +workingday+day+holiday+ day_type +temp_reg+humidi

ty+atemp+windspeed+season+weather+dp_reg+weekend+year+year_part, data=train,importanc

e=TRUE, ntree=250)
```

```
pred1=predict(fit1,test)

test$logreg=pred1

#predicting the log of casual users.

set.seed(415)

fit2 <- randomForest(logcas ~hour + day_type+day+humidity+atemp+temp_cas+windspeed+se
ason+weather+holiday+workingday+dp_cas+weekend+year+year_part, data=train,importance=
TRUE, ntree=250)

pred2=predict(fit2,test)

test$logcas=pred2
```

Re-transforming the predicted variables and then writing the output of count to the file submit.csv

```
test$registered=exp(test$logreg)-1

test$casual=exp(test$logcas)-1

test$count=test$casual+test$registered

s<-data.frame(datetime=test$datetime,count=test$count)

write.csv(s,file="submit.csv",row.names=FALSE)
```