BIKE-SHARING – TIME SERIES, DECISION TREES & LINEAR MODELS

1, Clean the data – correct outliers based on the count variable

The R codes for clean the data is as follow:

```
setwd("C:/Users/zizhe/Desktop)

library(forecast)

library(tseries)

 daily = read.csv('bike.csv', header=TRUE, stringsAsFactors=FALSE)

daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")

count_ts = ts(daily[, c('count')])

daily$count = tsclean(count_ts)

data<- daily
```

Discuss why these measurements have been altered.

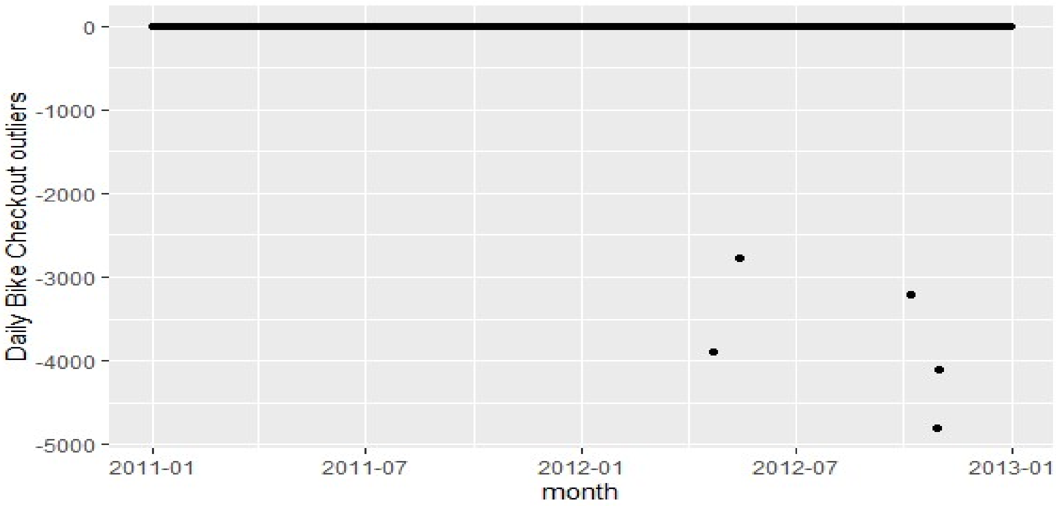
In some cases, the number of bicycles checked out dropped below 100 on the day and rose to over 4,000 the next day. These are suspected outliers that could bias the model by skewing statistical summaries. The outliers may affect the estimation of the final regression results of the model.

Produce a time series plot showing where these outliers occur.

R code for the plot:

```
daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")

count_ts = ts(daily[, c('count')])

daily$count_clean = tsclean(count_ts)

outlier=daily$count-daily$count_clean

ggplot(daily, aes(dteday, outlier)) + geom_point() +scale_x_date('month') + ylab("Daily Bike Checkout outliers") + xlab("")
```

The following plot shows where these outliers occur:

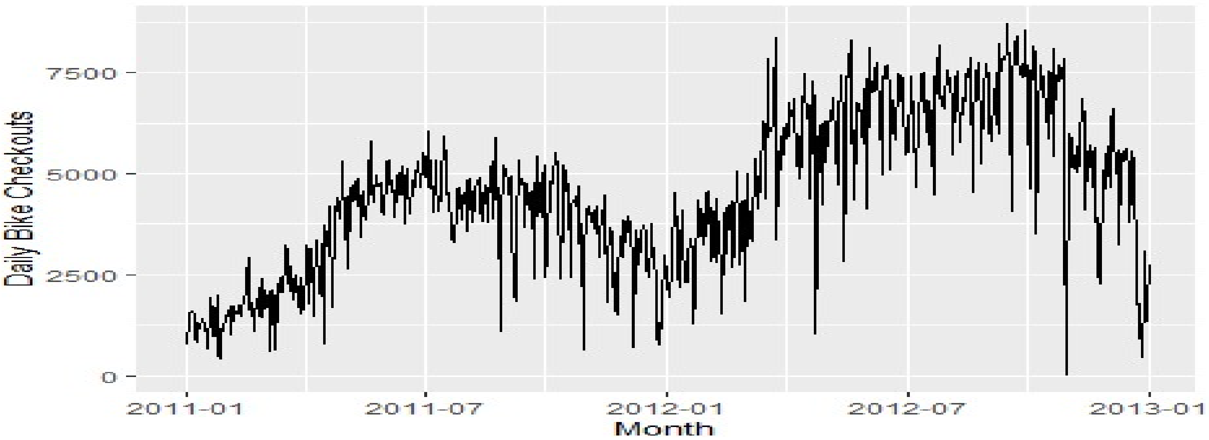


2, Describe the data and present summary visualizations/statistics for the bike-sharing count response over weekdays/weekends, holidays, weather patterns, etc.

```
setwd("C:/Users/zizhe/Desktop/INFO )

library(ggplot2)

library(forecast)

 daily = read.csv('bike.csv', header=TRUE, stringsAsFactors=FALSE)

daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")

count_ts = ts(daily[, c('count')])

daily$count = tsclean(count_ts)

data<- daily
```

Plot the original data:

```
ggplot(daily, aes(dteday, count)) + geom_line() + scale_x_date('month') + ylab("Daily Bike Checkouts")
+ xlab("")
```

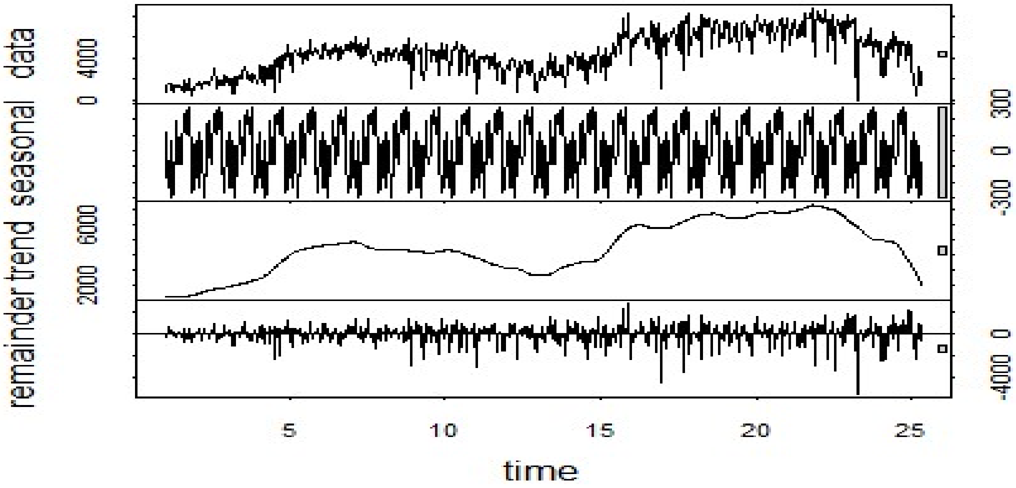


Do a monthly seasonal decomposition:

count <- ts(daily$count,frequency=30)

decomp = stl(count, s.window="periodic")

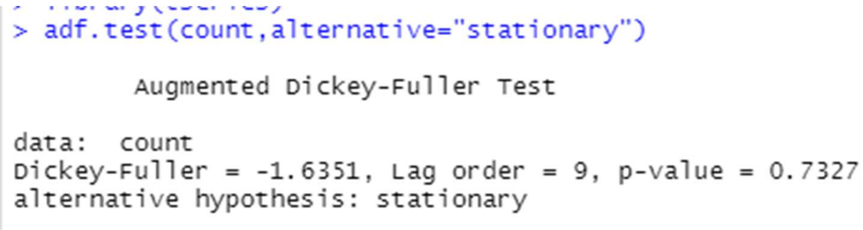
plot(decomp)



Next step to assess if the time series is stationary by using adf.test.

library(tseries)

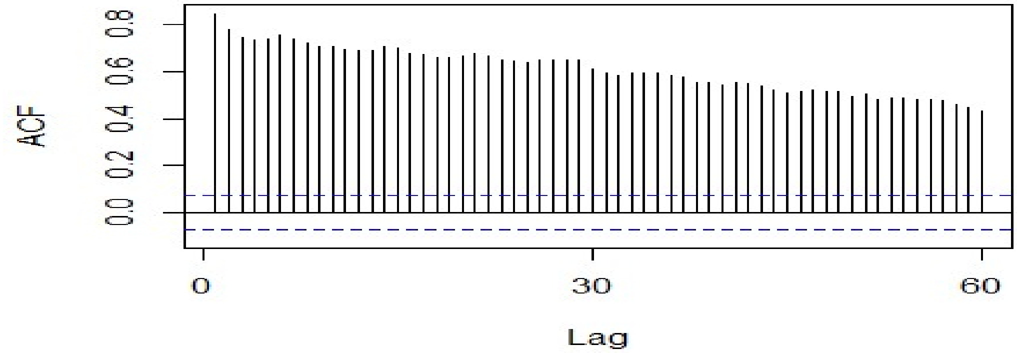
adf.test(count,alternative="stationary")



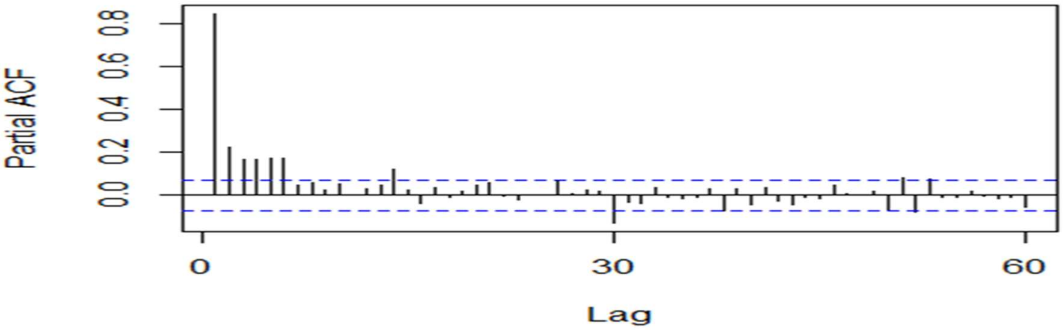
```
> library(tseries)
> adf.test(count,alternative="stationary")

        Augmented Dickey-Fuller Test

data:  count
Dickey-Fuller = -1.6351, Lag order = 9, p-value = 0.7327
alternative hypothesis: stationary
```

Next step, measure the autocorrelation

Acf(count, main='')



Pacf(count, main='')



Describe the data:

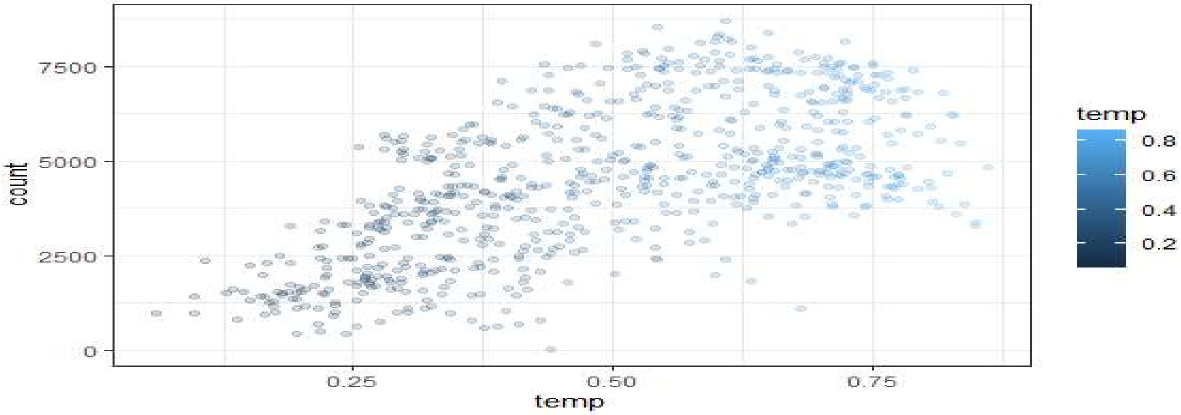
From the above different screenshots, we can see that the original data has a strong seasonality, so the original data need to do seasonal decomposition to extract the factor seasonal and look at other factors that affected the variable "count". After moving out the outliers by cleaning the data, I call the function STL in the additive model to do a monthly seasonal decomposition. From the decomposition plot, we can see that without the seasonal component, the overall trend for the variable "count" in the summer is still increasing from the year 2011 to the year 2012. Next step I call the function adf.test to test whether the time series is stationary. We can find that the P-value of adf. test is 0.7327> 0.05, this means the time series is not stationary. I need to used the Acf function to test whether the time series is stationary. The Acf plot shows the first column value in the Lag is similar to the original data value, which means the correlation in the time series is strong. The correlation reduced gradually month by month. After call the Pacf function, we can see that the bigger spikes represent numbers that we can consider to use in the Arima to forecast model.

Present summary visualizations/statistics:

Hypothesis: The count of bike rentals increases as the temperature rises. but when the temperature rises to a certain extent, the count of bicycle rentals decreases.

R code for the plot is as follow:

```
ggplot(data,aes(temp,count))+geom_point(aes(color=temp),alpha=0.2)+theme_bw()
```

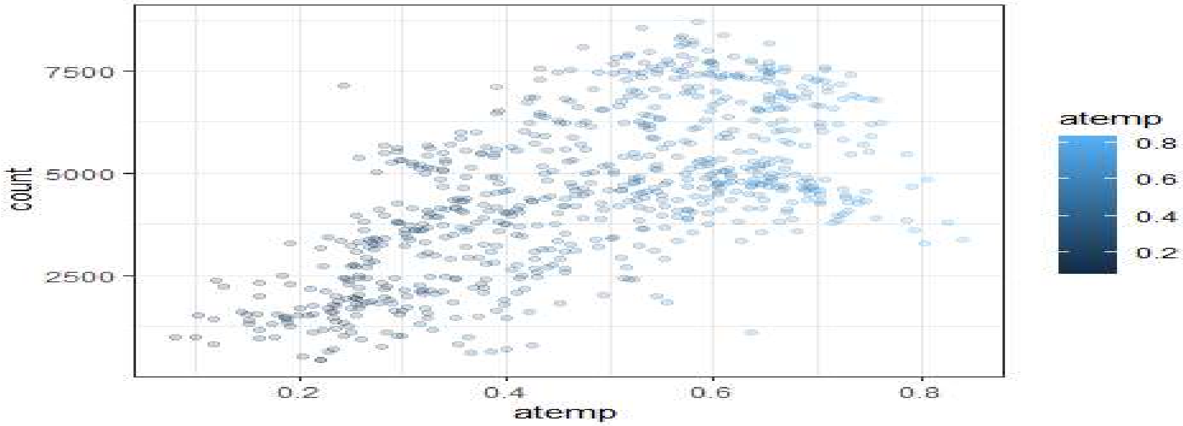


From the plot above, we can see that there is some degree of linearity between the count of bikes rented and the temperature; The warmer the weather, the more bikes are rented. The count of bike rentals gradually increases when the temperature is between 0.2 and 0.75. After 0.75, the count of bike rentals drops sharply. This verifies my hypothesis. (The temperature values have been normalized, the value is just between 0 and 1),

Hypothesis: The count of bike rentals increases as the atemp rises. but when the atemp rises to a certain extent, the count of bicycle rentals decreases.

R code for the plot is as follow:

```
ggplot(data,aes(atemp,count))+geom_point(aes(color=atemp),alpha=0.2)+theme_bw()
```



From the plot above, we can see that there is some degree of linearity between the count of bikes rented and the atemp; The warmer the atemp, the more bikes are rented. The count of bike rentals gradually increases when the atemp is between 0.2 and 0.7. After 0.7, the count of bike rentals drops sharply. This verifies my hypothesis. (The atemp values have been normalized, the value is just between 0 and 1)

We can see that the plot for temp and atemp looks very similar.

Call the correlation function: cor(data$temp,data$atemp), get the answer is 0.9917016.

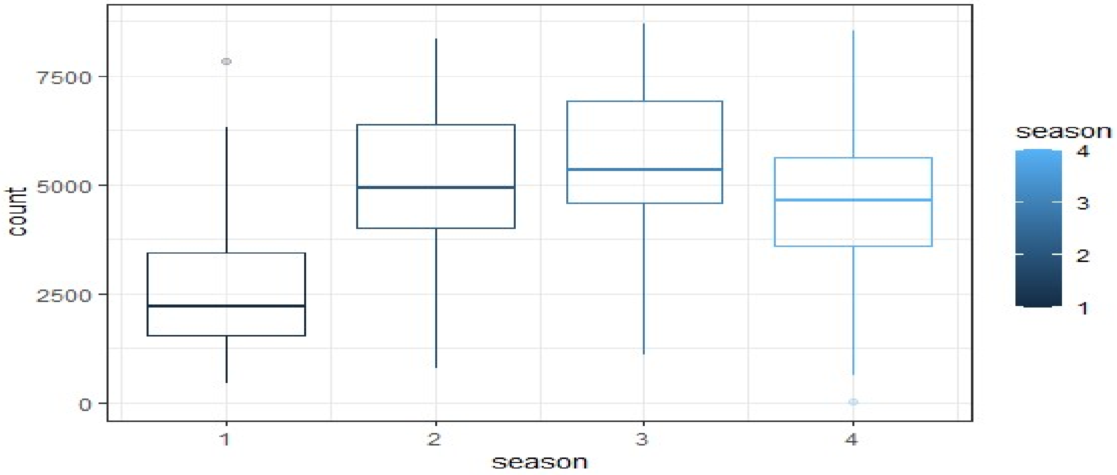
We can see that the temp and atemp are strongly correlated

Hypothesis: The count of rental bikes is lowest in winter and highest in summer.

R code for the plot is as follow:

```
ggplot(data,aes(season,count))+geom_boxplot(aes(group=season,color=season),alpha=0.2)+theme_bw()
```

From the Attribute Information of the Bike Sharing Dataset, **season (1:winter, 2:spring, 3:summer, 4:fall)**. Reference: https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset

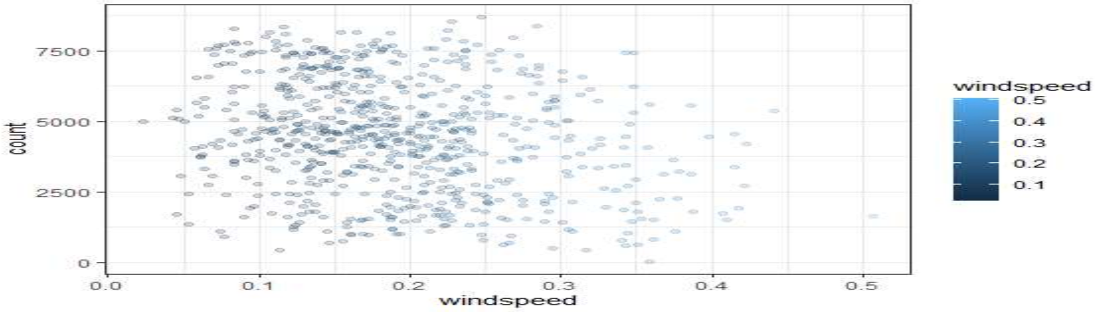


From the plot above, we can see that the count of rental bikes is lowest in winter and highest in summer. In the season that Spring, Summer, and Fall, the count of bike rentals are more than twice as many as the Winter count of bike rentals. This verifies my hypothesis.

Hypothesis: People are willing to rent bikes within a reasonable wind speed range.

R code for the plot is as follow:

```
ggplot(data,aes(windspeed,count))+geom_point(aes(color=windspeed),alpha=0.2)+theme_bw()
```

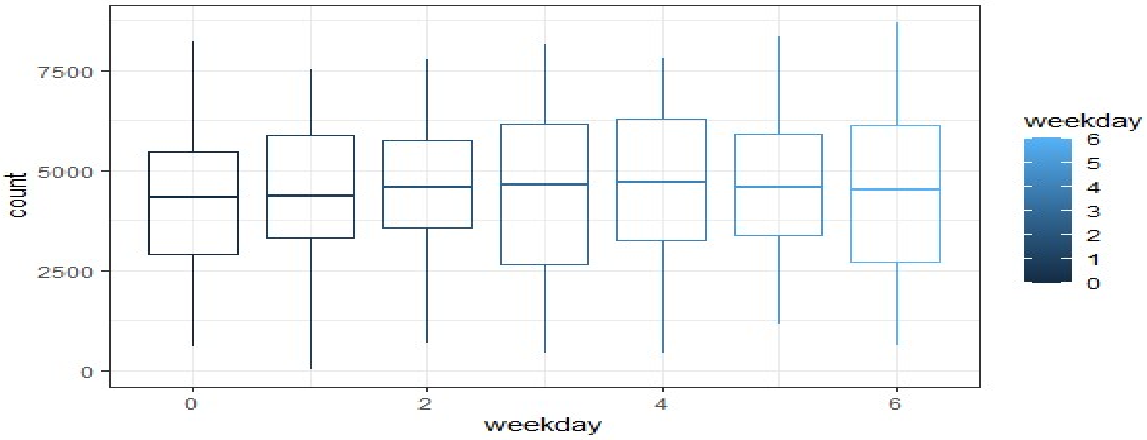


From the plot above, we can see that the relationship between wind speed and the count of bike rentals does show a non-linear relationship, the higher the wind speed, the fewer bike rentals. When the wind speed is between 0.1 and 0.3, the count of rental bikes is the largest. After 0.3, the count of bike rentals drops sharply. This verifies my hypothesis. (The windspeed values have been normalized, the value is just between 0 and 1).

Hypothesis: Most people rent bikes may be on any weekday.

R code for the plot is as follow:

ggplot(data,aes(weekday,count))+geom_boxplot(aes(group=weekday,color=weekday),alpha=0.2)+theme_bw()



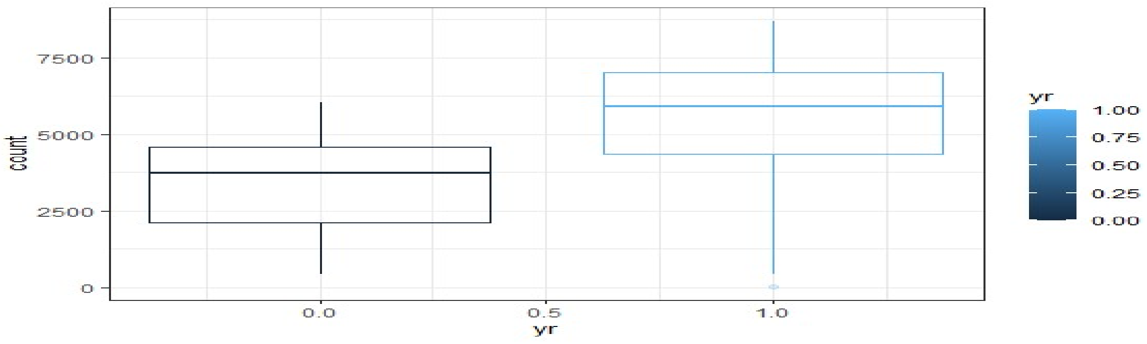
From the plot above, we can see that the count of bike rentals is highest on Saturday and Wednesday, and lowest on Tuesday. We found no particular pattern in renting bikes every day. This verifies my hypothesis.

Hypothesis: When people get familiar with bike rentals, the number of bike rentals starts to increase.

R code for the plot is as follow:

ggplot(data,aes(yr,count))+geom_boxplot(aes(group=yr,color=yr),alpha=0.2)+theme_bw()

From the Attribute Information of the Bike Sharing Dataset, **yr: year (0: 2011, 1:2012).**

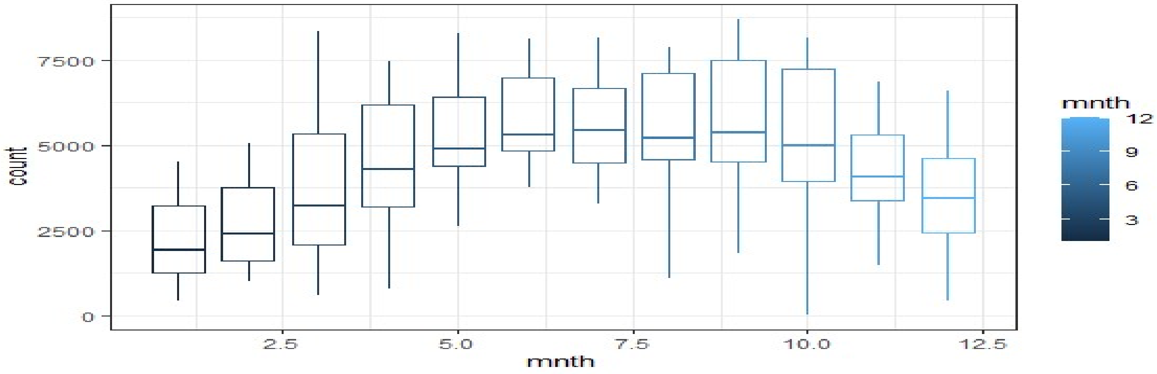


From the plot above, we can see that the count of bike rentals in the year 2012 roughly doubled from the year 2011. This verifies my hypothesis.

Hypothesis: People prefer to rent bikes in the months when the temperature is good.

R code for the plot is as follow:

ggplot(data,aes(mnth,count))+geom_boxplot(aes(group = mnth,color=mnth),alpha=0.2)+theme_bw()



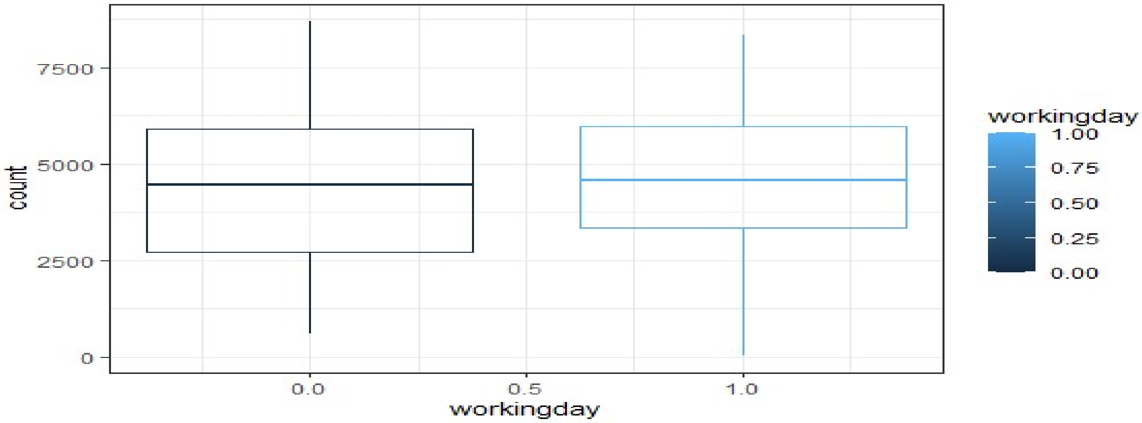
From the plot above, we can see that the count of bike rentals was highest in March, April, August, September, and October, the temperatures in these months are perfect for outdoor cycling, which was consistent with previous seasonal analysis results. This verifies my hypothesis.

Hypothesis: people maybe rent bikes on any day between the working days and non-working days.

R code for the plot is as follow:

ggplot(data,aes(workingday,count))+geom_boxplot(aes(group=workingday,color=workingday),alpha =0.2)+theme_bw()

From the Attribute Information of the Bike Sharing Dataset,(workingday : if day is neither weekend nor holiday is 1, otherwise is 0.)

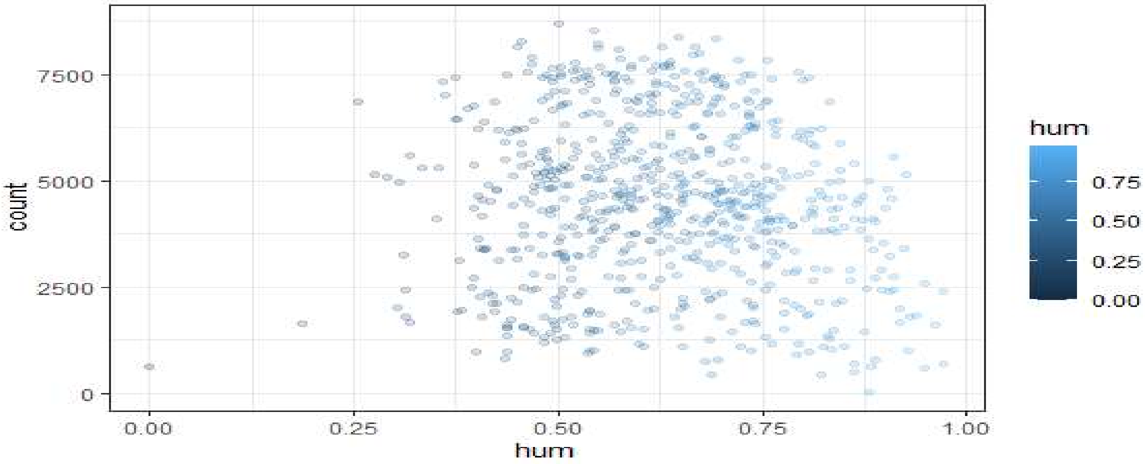


From the plot above, we can see that the count of the bike rentals within workingdays are a little bit more than non-workingdays, there is not too much difference. This verifies my hypothesis.

Hypothesis: People are willing to rent bikes within a reasonable humidity range.

R code for the plot is as follow:

ggplot(data,aes(hum,count))+geom_point(aes(group=hum,color=hum),alpha=0.2)+theme_bw()



From the plot above, we can see that when the humidity value is between 0.4-0.8, with the largest count of bicycle rentals. This verifies my hypothesis. (The humidity values have been normalized, the value is just between 0 and 1).

Hypothesis: People like to rent bikes in good weather, but less in bad weather.

R code for the plot is as follow:

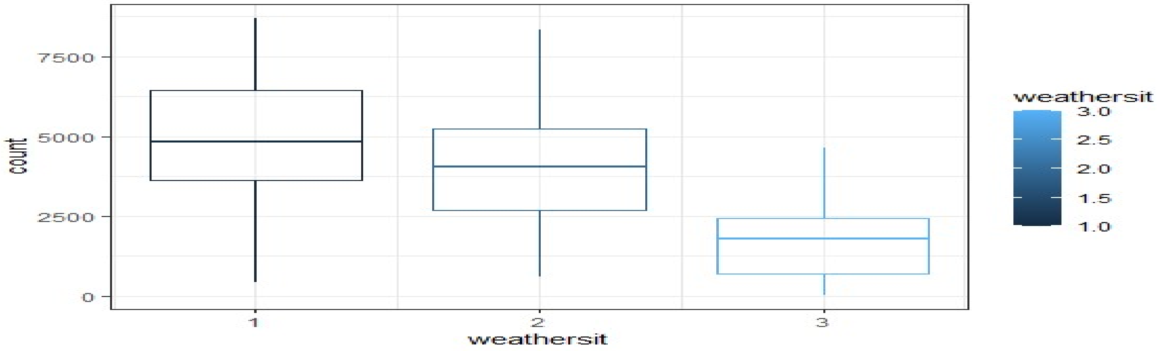
ggplot(data,aes(weathersit,count))+geom_boxplot(aes(group=weathersit,color=weathersit),alpha=0.2)+theme_bw()

From the Attribute Information of the Bike Sharing Dataset,

(weathersit : - 1: Clear, Few clouds, Partly cloudy

- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds)

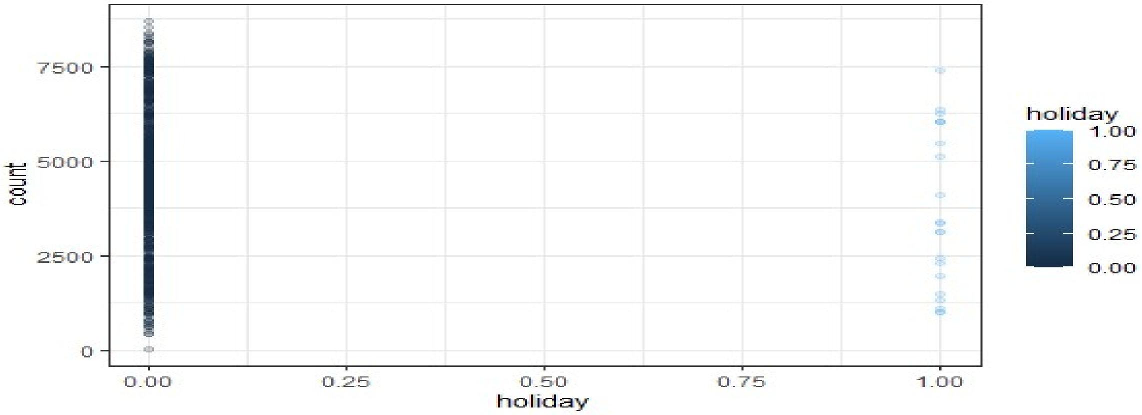


From the plot above, we can see that when the weather condition is good, the count of bike rentals are high, when the weather condition is not good, the count of bike rentals are very low. The weather has a big impact on bike rental count. This verifies my hypothesis.

Hypothesis: People rent bikes mainly for the convenience of work,

R code for the plot is as follow:

```
ggplot(data,aes(holiday,count))+geom_point(aes(group=holiday,color=holiday),alpha=0.2)+theme_bw()
```

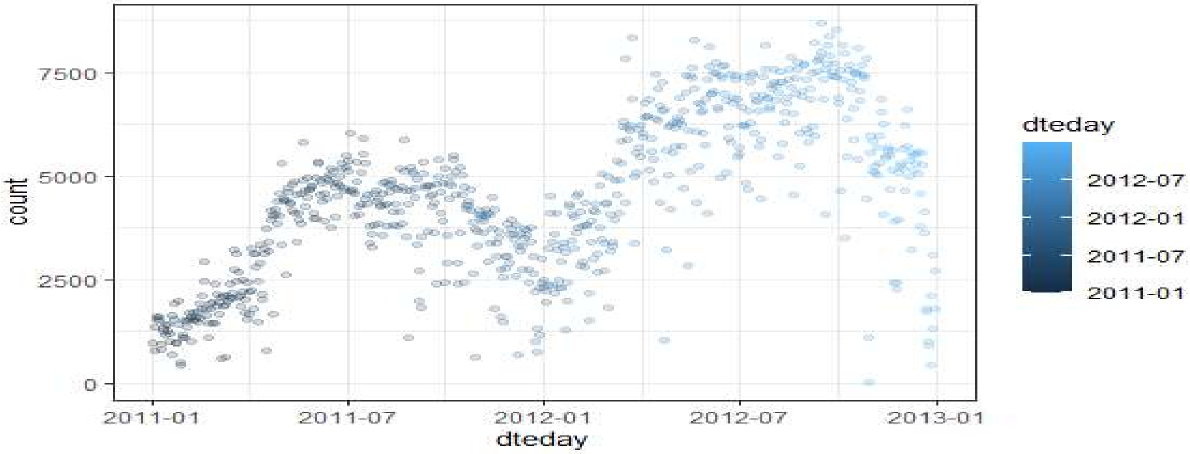


From the plot above, we can see that the count of rental bikes on non-holiday days is much larger than that on holiday days. This verifies my hypothesis.

Hypothesis: The bike rental numbers fluctuate seasonally and have time series characteristics.

R code for the plot is as follow:

```
ggplot(data,aes(dteday,count))+geom_point(aes(group=dteday,color=dteday),alpha=0.2)+theme_bw()
```



From the plot above, we can see that between January 2011 and December 2012, the daily change in the count of bike rentals fluctuated with the seasons. the plot shows that the count of bike rentals is gradually increasing from winter to summer and then gradually decreasing from summer to winter. Overall, the count of bike rental for the year 2012 increased compared to the year 2011. This verifies my hypothesis.

3, Build a linear model for predicting the response count.

First step, show the R code for run the linear model without convert any variables to factors

```
setwd("C:/Users/zizhe/Desktop/INFO)
 daily = read.csv('bike.csv', header=TRUE, stringsAsFactors=FALSE)
View(daily)
daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")
count_ts = ts(daily[, c('count')])
 daily$count = tsclean(count_ts)
# REMOVE the explanatory dteday
daily<- daily[,-1]
perc.train <- 0.6 # 60 percent training
num.samples <- 50
rsqr1.lm <- vector()
rsqr1<- 1 - sum((y - yhat)^2) / sum((y - mean(y))^2)
for (run in 1:num.samples)
{ train.rows <- sample(1:nrow(daily),nrow(daily)*perc.train,replace=FALSE)
train<- daily[train.rows,]
 test<- daily[-train.rows,]
mdl <- lm(count ~ . , train)
y <- test$count # test data
 yhat <- predict(mdl, test) # prediction using all test data
rsqr1 [run]<- 1 - sum((y - yhat)^2) / sum((y - mean(y))^2)}
```

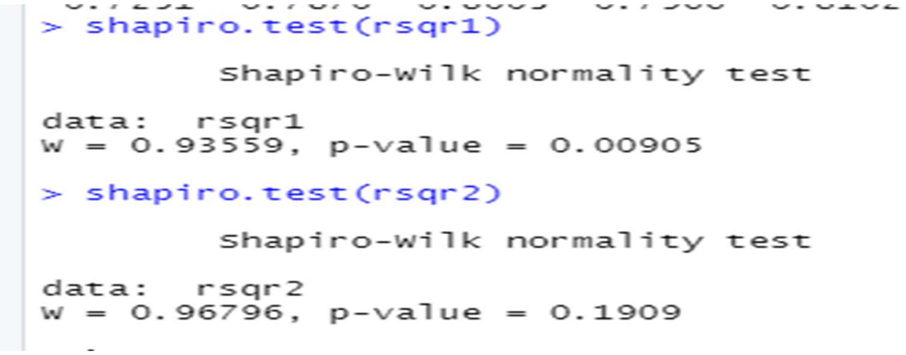
Second step, show the R code for run the linear model convert variables to factors.

```
setwd("C:/Users/zizhe/Desktop/INFO 424/Lab6")
 daily = read.csv('bike.csv', header=TRUE, stringsAsFactors=TRUE)
View(daily)
daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")
count_ts = ts(daily[, c('count')])
 daily$count = tsclean(count_ts)
# REMOVE the explanatory dteday
daily<- daily[,-1]
daily$season<-as.factor(daily$season)
```

```
 daily$yr<-as.factor(daily$yr)
daily$mnth<-as.factor(daily$mnth)
 daily$holiday<-as.factor(daily$holiday)
 daily$weekday<-as.factor(daily$weekday)
daily$workingday<-as.factor(daily$workingday)
daily$weathersit<-as.factor(daily$weathersit)
perc.train <- 0.6 # 60 percent training
num.samples <- 50
rsqr2.lm <- vector()
rsqr2 <- 1 - sum((y - yhat)^2) / sum((y - mean(y))^2)
for (run in 1:num.samples)
{ train.rows <- sample(1:nrow(daily),nrow(daily)*perc.train,replace=FALSE)
train<- daily[train.rows,]
 test<- daily[-train.rows,]
mdl <- lm(count ~ . , train)
y <- test$count # test data
 yhat <- predict(mdl, test) # prediction using all test data
rsqr2 [run]<- 1 - sum((y - yhat)^2) / sum((y - mean(y))^2)}
```

Compare rsqr1 and rsqr2:

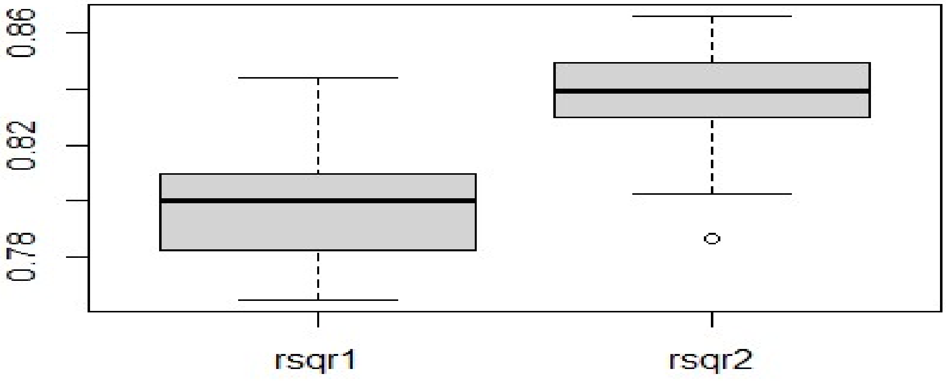
The first step is to do the shapiro.test to check if rsqr1 and rsqr2 distributions are normal or not.



```
> shapiro.test(rsqr1)

        Shapiro-Wilk normality test

data:  rsqr1
W = 0.93559, p-value = 0.00905

> shapiro.test(rsqr2)

        Shapiro-Wilk normality test

data:  rsqr2
W = 0.96796, p-value = 0.1909
```

From the above screenshot, we can see that the rsqr1 P-value = 0.00905 < 0.05, rsqr2 P-value =0.1909>0.05. this means only rsqr2 has a normal distribution, so we don’t need to do a further test. After comparing the rsqr1 and rsqr2 P-value results, we can find that rsqr1 and rsqr2 are different.

boxplot(cbind(rsqr1,rsqr2),yim=c(0,1))   We can see the plot below shows the rsqr result prediction is better when we convert variables to factors..



Explain why there are differences (if any)

1, The values that we selected for the training dataset and test datasets are random. So, after calculating, the R^2 results are random results.

2, After using factors for variables, the values of the variables would be more meaningful and more accurate. The R^2 result was found to conform to the normal distribution, The R^2 of the normal distribution can show the best fit line optimized within the data.

Explain why certain variables are now factors.
In R, factors are used to work with categorical variables, variables that have a fixed and known set of possible values. Factors are used to represent the category in a set of data, which can record the category name and the number of categories in this set of data. They are also useful when you want to display character vectors in a non-alphabetical order. Historically, factors were much easier to work with than characters. The reason certain variables are now factors is that it is convenient (have orders) to summarize, visualize and model the related variables. Factor variables can have less storage space.

In this case, the variables named season, yr, mnth, holiday, weekday, workingday, weathersit should be treated as factors, as they are the data that are treated as categorical data.

4, Build a decision tree model for the data

```
library(rpart)

library(rpart.plot)

library(forecast)

library(tseries)

setwd("C:/Users/zizhe/Desktop)

 daily = read.csv('bike.csv', header=TRUE, stringsAsFactors=TRUE)
```

```
View(daily)
daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")
count_ts = ts(daily[, c('count')])
 daily$count = tsclean(count_ts)
# REMOVE the explanatory dteday
daily<- daily[,-1]
```

# The R code for the decision tree is as follow:

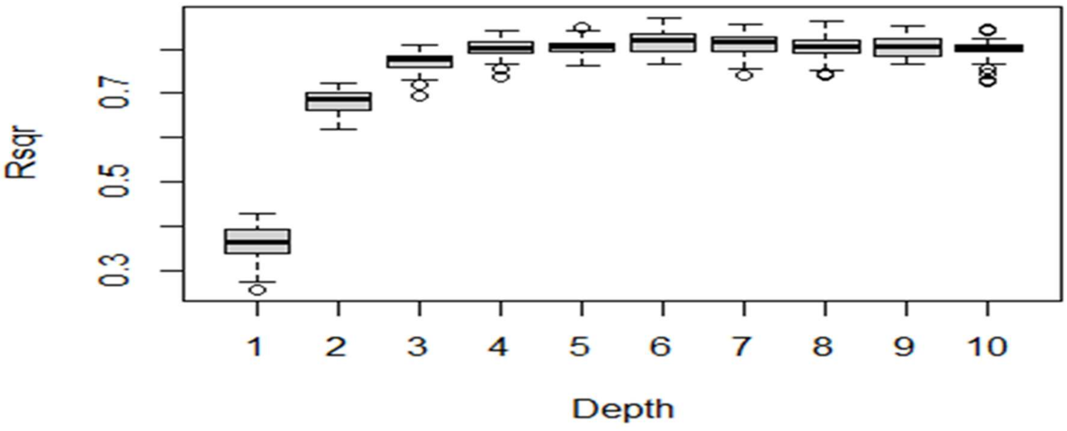
```
tree <- function(data,formula,perc.train=0.6,...)
{
   train.rows <- sample(1:nrow(data),nrow(data)*perc.train,replace=FALSE)
  train <- data[train.rows,]
  test <- data[-train.rows,]
   rp <- rpart(formula, data=train,
             control=rpart.control(...))
  yhat <- predict(rp,
                  newdata=test,
                   type="vector")
   rsqr <- function(y, yhat)
{1 - sum((y - yhat)^2) / sum((y - mean(y))^2)}
  return(rsqr(test$count, yhat))
}
```

# The R code for maxdepth from 1 to 10 is as follow:

```
rsqr.dp<-matrix(nrow=50,ncol=10)
for (depth in 1:10)
{
  res<-replicate(50, tree(daily, count ~ .,maxdepth=depth,
                                  minsplit=2,
                                  minbucket=2,
                                  cp=0))
    rsqr.dp[,depth]<-cbind(res)
  }
```

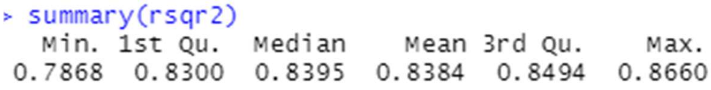
# Boxplot get the rsqr plot:

boxplot(rsqr.dp,xlab="Depth",ylab="Rsqr")



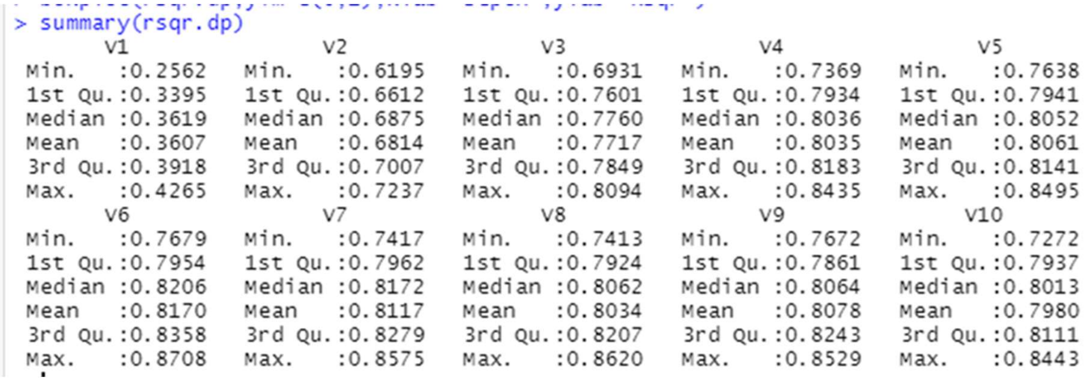
Compare the resulting prediction quality to the linear model,

I would like to use the mean value for each depth of rsqr to compare with the linear model mean Rsqr2.



```
> summary(rsqr2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.7868  0.8300  0.8395  0.8384  0.8494  0.8660 
```

From the above screenshot, we can get that the linear model mean of Rsqr2= 0.8384



```
> summary(rsqr.dp)
       V1               V2               V3               V4               V5        
 Min.   :0.2562   Min.   :0.6195   Min.   :0.6931   Min.   :0.7369   Min.   :0.7638  
 1st Qu.:0.3395   1st Qu.:0.6612   1st Qu.:0.7601   1st Qu.:0.7934   1st Qu.:0.7941  
 Median :0.3619   Median :0.6875   Median :0.7760   Median :0.8036   Median :0.8052  
 Mean   :0.3607   Mean   :0.6814   Mean   :0.7717   Mean   :0.8035   Mean   :0.8061  
 3rd Qu.:0.3918   3rd Qu.:0.7007   3rd Qu.:0.7849   3rd Qu.:0.8183   3rd Qu.:0.8141  
 Max.   :0.4265   Max.   :0.7237   Max.   :0.8094   Max.   :0.8435   Max.   :0.8495  
       V6               V7               V8               V9              V10        
 Min.   :0.7679   Min.   :0.7417   Min.   :0.7413   Min.   :0.7672   Min.   :0.7272  
 1st Qu.:0.7954   1st Qu.:0.7962   1st Qu.:0.7924   1st Qu.:0.7861   1st Qu.:0.7937  
 Median :0.8206   Median :0.8172   Median :0.8062   Median :0.8064   Median :0.8013  
 Mean   :0.8170   Mean   :0.8117   Mean   :0.8034   Mean   :0.8078   Mean   :0.7980  
 3rd Qu.:0.8358   3rd Qu.:0.8279   3rd Qu.:0.8207   3rd Qu.:0.8243   3rd Qu.:0.8111  
 Max.   :0.8708   Max.   :0.8575   Max.   :0.8620   Max.   :0.8529   Max.   :0.8443  
```

We can see that the mean of rsqr results for each maxdepth from 1 to 10 are shown in the above Summary screenshot separately are 0.3607, 0.6814, 0.7717, 0.8035, 0.8061, 0.8170, 0.8117, 0.8034, 0.8078, 0.7980. All the results are lower than the linear model mean of Rsqr2 which is 0.8384.

So after the comparison, we can get that the linear model performs better. Because a higher result in the R^2 means the prediction value is closer to the actual value.

Discuss what maxdepth of the tree is suitable to be used to build the decision tree over all of the data.

In this task, I think when **the decision tree maxdepth= 6**, the tree is suitable to be used to build the decision tree over all of the data. Because when the decision tree **maxdepth= 6**, the mean Rsqr result= 0.8170. It is the highest mean Rsqr result compare with other mean Rsqr results. The task is to obtain an accurate prediction of the response variable. There will be an error in the prediction, the more accurate the prediction, the less error happened. A higher Rsqr value means less model error. This is why we choose the highest Rsqr result to get the best-fitted decision tree model for all of the data.

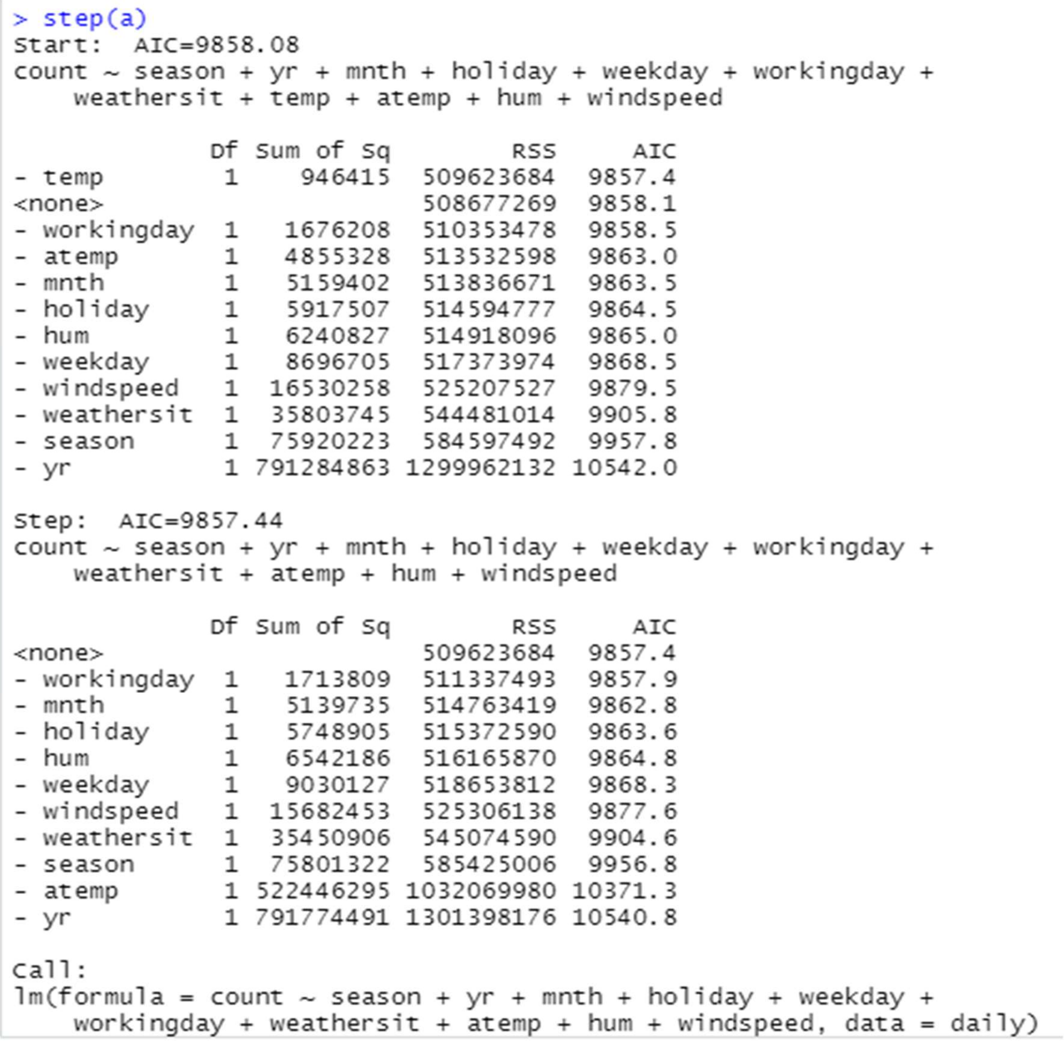
5, Fit a single (stepwise) linear model to the entire dataset

The R code is as follow:

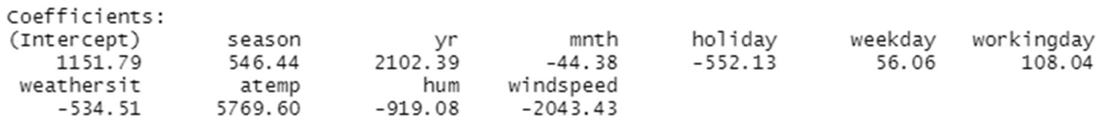
a<-lm(count ~ . , daily)

# Show the stepwise for the factor linear model

step(a)



```
> step(a)
Start:  AIC=9858.08
count ~ season + yr + mnth + holiday + weekday + workingday + 
    weathersit + temp + atemp + hum + windspeed

             Df Sum of Sq        RSS     AIC
- temp        1    946415  509623684  9857.4
<none>                     508677269  9858.1
- workingday  1   1676208  510353478  9858.5
- atemp       1   4855328  513532598  9863.0
- mnth        1   5159402  513836671  9863.5
- holiday     1   5917507  514594777  9864.5
- hum         1   6240827  514918096  9865.0
- weekday     1   8696705  517373974  9868.5
- windspeed   1  16530258  525207527  9879.5
- weathersit  1  35803745  544481014  9905.8
- season      1  75920223  584597492  9957.8
- yr          1 791284863 1299962132 10542.0

Step:  AIC=9857.44
count ~ season + yr + mnth + holiday + weekday + workingday + 
    weathersit + atemp + hum + windspeed

             Df Sum of Sq        RSS     AIC
<none>                     509623684  9857.4
- workingday  1   1713809  511337493  9857.9
- mnth        1   5139735  514763419  9862.8
- holiday     1   5748905  515372590  9863.6
- hum         1   6542186  516165870  9864.8
- weekday     1   9030127  518653812  9868.3
- windspeed   1  15682453  525306138  9877.6
- weathersit  1  35450906  545074590  9904.6
- season      1  75801322  585425006  9956.8
- atemp       1 522446295 1032069980 10371.3
- yr          1 791774491 1301398176 10540.8

Call:
lm(formula = count ~ season + yr + mnth + holiday + weekday + 
    workingday + weathersit + atemp + hum + windspeed, data = daily)
```



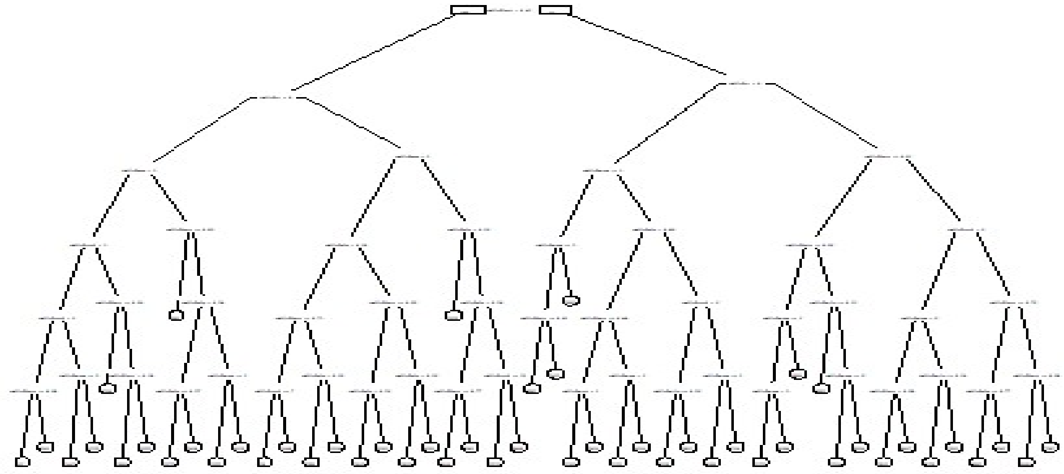
```
Coefficients:
(Intercept)       season           yr         mnth      holiday      weekday   workingday  
    1151.79       546.44      2102.39       -44.38      -552.13        56.06       108.04  
 weathersit        atemp          hum    windspeed  
    -534.51      5769.60      -919.08     -2043.43  
```

Fit decision tree (using the most suitable maxdepth that is determined in question 4) to the entire dataset.

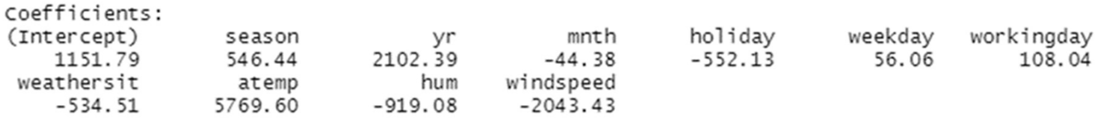
The R code is as follows:

b=rpart (count ~ ., daily, maxdepth=6, minsplit=2, minbucket=2, cp=0)

prp(b)



Discuss/interpret the meaning of the coefficients. What do they tell you about which factors influence the high/low behaviour with bike-sharing?



```
Coefficients:
(Intercept)       season           yr         mnth      holiday      weekday   workingday
    1151.79       546.44      2102.39       -44.38      -552.13        56.06       108.04
 weathersit        atemp          hum    windspeed
    -534.51      5769.60      -919.08     -2043.43
```

The coefficient reflects the degree of correlation between the explanatory variable X and the response variable Y. The coefficient represents the parameter of the influence of explanatory variable X on response variable Y in the regression equation. The larger the coefficient is, the more influence x has on Y. The positive coefficient means that y increases with the increase of x, and the negative coefficient means that y decreases with the increase of x.

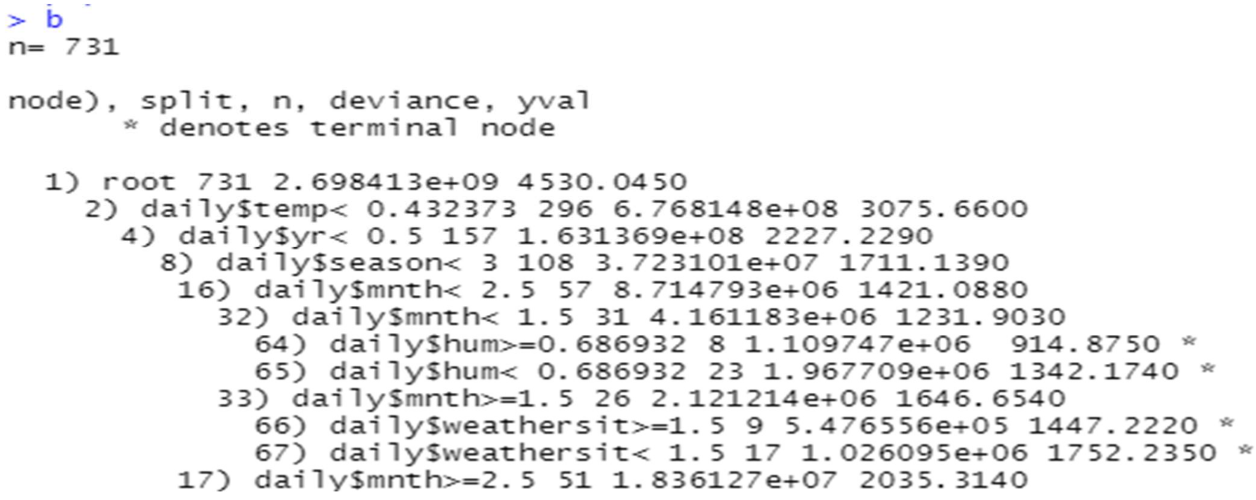
From the coefficients shown above, we can see that the explanatory variables have a positive impact on the response variable Y including **season, year, weekday, workingday, and atemp**. This means the count of bike renting numbers will increase with the increase of the above listed explanatory variables.

The explanatory variables have a negative impact on the response variable Y including **month, holiday, weather, humidity, and windspeed**. This means the count of bike renting numbers will decrease with the increase of the above-listed explanatory variables.

We can see that the factor “atemp=5769.60” has the highest positive influence on bike-sharing, and the factor “windspeed= (-2043.43)” has the highest negative influence on bike-sharing.

Discuss/interpret the meaning of the decision tree. What do they tell you about which factors influence the high/low behaviour with bike sharing?

In the decision tree, questions are asked to each node, and nodes are bifurcated according to the answers, so as to achieve the purpose of data classification. The decision tree is a kind of supervised learning algorithm. Make a series of decisions based on attributes, and each decision either leads to the next decision or produces the final result.



```
> b
n= 731

node), split, n, deviance, yval
      * denotes terminal node

  1) root 731 2.698413e+09 4530.0450
    2) daily$temp< 0.432373 296 6.768148e+08 3075.6600
      4) daily$yr< 0.5 157 1.631369e+08 2227.2290
        8) daily$season< 3 108 3.723101e+07 1711.1390
         16) daily$mnth< 2.5 57 8.714793e+06 1421.0880
           32) daily$mnth< 1.5 31 4.161183e+06 1231.9030
             64) daily$hum>=0.686932 8 1.109747e+06  914.8750 *
             65) daily$hum< 0.686932 23 1.967709e+06 1342.1740 *
           33) daily$mnth>=1.5 26 2.121214e+06 1646.6540
             66) daily$weathersit>=1.5 9 5.476556e+05 1447.2220 *
             67) daily$weathersit< 1.5 17 1.026095e+06 1752.2350 *
         17) daily$mnth>=2.5 51 1.836127e+07 2035.3140
```

From the result of the decision tree shown above, N =731 means there are 731 data points. The indentation of the decision tree represents the new branch, * represents the leaf node.

Each row in the output has five columns, for example:

2) daily$temp< 0.432373 296 6.768148e+08 3075.6600

1. 2) is the number of a node branch
2. daily$temp< 0.432373 is the splitting rule being applied to the parent node
3. 296 is the number of points that would be at this node of the tree
4. 6.768148e+08 is the deviance at this node (used to decide how the split was made)
5. 3075.6600 is what you would predict for points at this node if you split no further.

The decision tree is divided into 6 levels of depth. Examples of the leaf node of the sixth level of depth is 64) daily$hum>=0.686932 8 1.109747e+06 914.8750 *

65) daily$hum< 0.686932 23 1.967709e+06 1342.1740 *

We can see that the factor "temp" as the second depth node (number of points = 296) has a high positive influence on bike-sharing, and the factor "windspeed" as the leaf node(number of points are less than 20 in each leaf node) has a high negative influence on bike-sharing.

Do they agree in terms of the relationships of the variables?

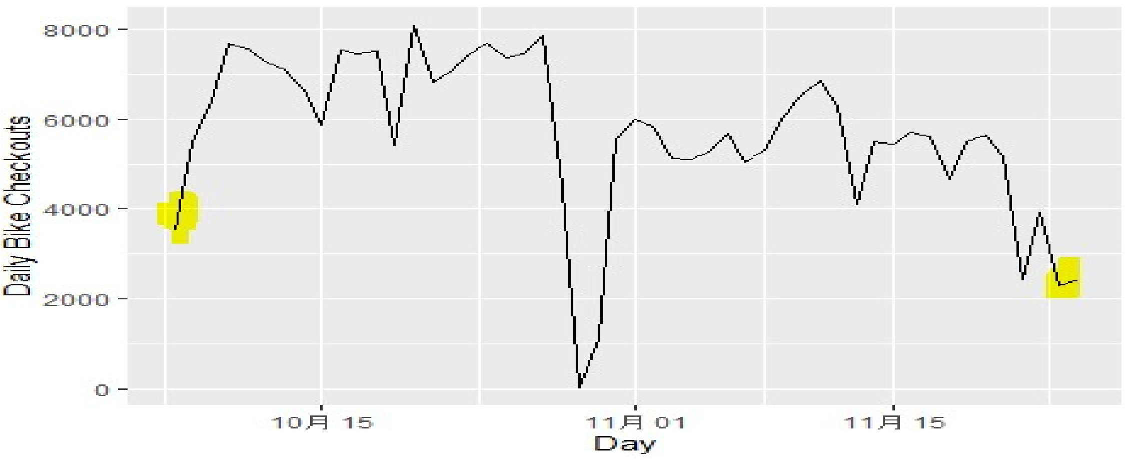
I think the coefficients of the linear model and the result of the decision tree model. They agree in terms of the relationship of variables. Because in the linear model, the factors that have the greatest impact on the count of bicycle rentals are atemp and year (atemp and temperature have a very strong positive correlation, and they mean similar things), In the decision tree model, the factors that have the greatest influence on the count of bicycle rentals are also temperature and year. We can see that in the hierarchical depth node of the decision tree, the first and second depth of the nodes are temperature and year respectively, indicating that temperature and year have the greatest influence on the count result.

6, Explain why these events (dates) have been labelled as anomalous and briefly discuss what are the properties of an anomalous event.

The two events selected from Table 4 are “Unseasonably cool weather”, date = 07/10/2012. and “The coldest morning of the season”, date = 25/11/2012.

The R code for showing these two dates is as follow:

```
daily <- read.csv('bike.csv', header=TRUE, stringsAsFactors=FALSE)

daily$dteday = as.Date(daily$dteday,format="%d/%m/%Y")

library(ggplot2)

rownames(daily)<-daily[,1]

monthstart<- which(rownames(daily)=="2012-10-07")

monthsend<- which(rownames(daily)=="2012-11-25")

month<-daily[monthstart:monthsend,]

ggplot(month, aes(dteday, count)) + geom_line() + scale_x_date('Day') + ylab("Daily Bike Checkouts")
+ xlab("")
```



From the screenshot above, we can see that The events “Unseasonably cool weather” date = 07/10/2012 and “The coldest morning of the season” date = 25/11/2012 have been labelled as anomalous (Yellow painting part), because these events are random events that occur without warning, with low probability, these events are just like the outliers, outliers refer to sample points in which some values in a sample deviate significantly from other values. The event “Unseasonably cool weather” does not always happen every day, every week, and every month. It does not satisfy the rules of seasonal weather.

The properties of an anomalous event: Anomalous events occur outside the range of normal and reasonable events. An anomalous event is one that is difficult to predict under normal circumstances. The occurrence of anomalous events has no pattern to follow. It does not follow the general pattern in the time series.