Ashley Code Sample

Ashley Gu

9/6/2017

Project 2

Introduction

The dataset comes from the Capital Bikeshare program in Washington, D.C. The goal of the project is to fit a linear regression equation to the total count of bikes rented during a particular hour in terms of the available explanatory variables for the training data (train.csv) and then to use this regression equation to predict the counts for the test dataset.

1. Data Loading

```
#reading csv files

train = read.csv("train.csv", header = T)
test = read.csv("test.csv" ,header = T)
```

In the original dataset, the time is given in terms of year-date-hour, it would be more helpful if we could group them by different time periods of the day and make it a categorical varaible.

Besides hours, the year and month are also provided by first colum. The effect of month on total counts of rental per hour could be captured by the "season" variable. So I decided to only make the additional "time of day" and "year" column.

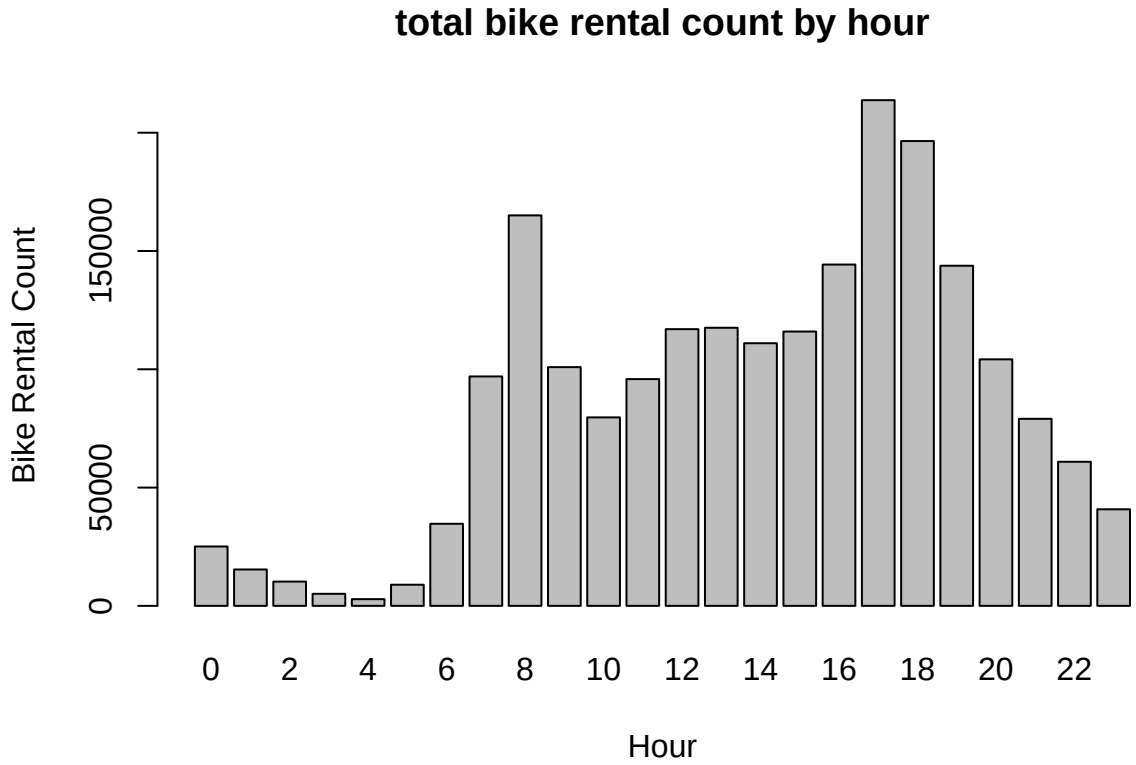
The hour of day may be a very important factor of total rental count because of people's prefered time of traveling, rush hour and etc. So I decided to plot a graph visualizing the count of total rental each hour to see how I should segment a day into several periods.

```
#create an empty list of integers to hold the counts per hour


#this is a function to aggregate the counts per hour, "grepl" is a function in r that checks matching p
segment.func = function(x){
  count_by_hour = vector(mode = "integer", length = 24)
  for (i in c(1:nrow(train))){
    for (n in c(0:9)) {
      string_time = paste("0", toString(n), ":00:00", sep = "")

      if (grepl(string_time, train$datetime[i])) {
        count_by_hour[n+1] = count_by_hour[n+1] + train$count[i]
      }
    }
    for (n in c(10:23)) {
      string_time = paste(toString(n), ":00:00", sep = "")
      if (grepl(string_time, train$datetime[i])) {
        count_by_hour[n+1] = count_by_hour[n+1] + train$count[i]
      }
```

```
    }
  }
  return(count_by_hour)
}

count_by_hour = segment.func(as.character(train$datetime))
summary(count_by_hour)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2832   32300   96410   86890  117100  213800
```

```
barplot(count_by_hour, main = "total bike rental count by hour", xlab = "Hour", ylab = "Bike Rental Cou
```



The barplot above is the sum of all bike rental in the dataset by hour.

According to this barplot, we can see that from 0 - 6am, the count is very low. The count from 7am to 7pm is on the same level besides the rush hours (8am, 5pm,6pm). Then the count decreases from 8pm to 11pm. Therefore, I decided to group the hours into the following categories:

group1(sleep) 0-6 group2(work) 7, 9-16, 19 group3(afterwork) 20-23 group4(traffic) 8, 17, 18

```
#create a seperate variable called "time" which includes sleep, work, afterwork, traffic

time_list = c()


#this function detects whether a certain hourly time shows up in the datetime column and group them int
time.func = function(x){
  for (i in x){
    if (grepl("00:00:00",i) |grepl("01:00:00",i) |grepl("02:00:00",i)|grepl("03:00:00",i)|grepl("04:00:
      list = c(list,"sleep")
    }
```

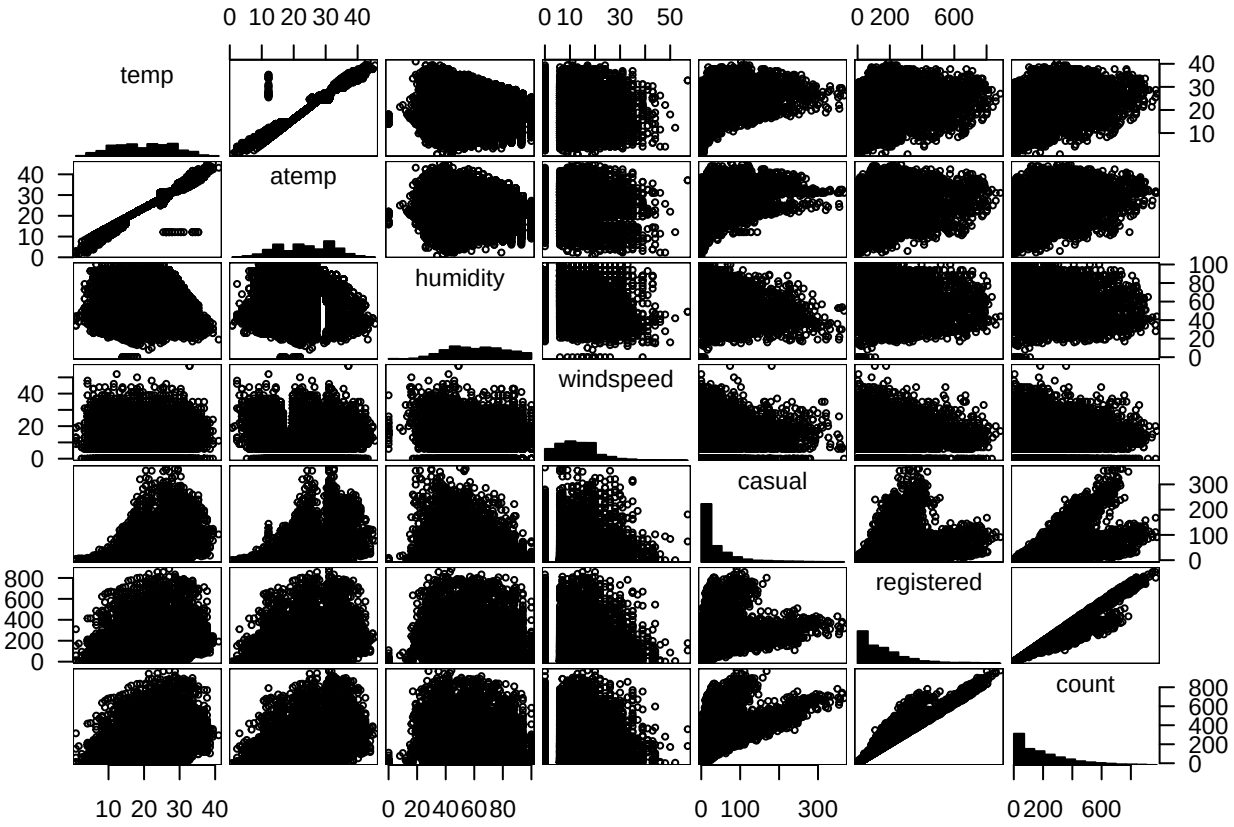
```
    if (grepl("07:00:00",i) |grepl("09:00:00",i) |grepl("10:00:00",i)|grepl("11:00:00",i)|grepl("12:00:
      list = c(list,"work")
    }

    if (grepl("20:00:00",i) |grepl("21:00:00",i) |grepl("22:00:00",i)|grepl("23:00:00",i)) {
      list = c(list,"afterwork")
    }

    if (grepl("08:00:00",i) |grepl("17:00:00",i) |grepl("18:00:00",i)) {
      list = c(list,"traffic")
    }
  }
  return(list)
}

#add the time_list into the train table and name it as "time"
time_list = time.func(as.character(train$datetime))
train$time = unlist(time_list[2:10887])
head(train)
```

```
##              datetime season holiday workingday weather temp  atemp
## 1 2011-01-01 00:00:00      1       0          0       1 9.84 14.395
## 2 2011-01-01 01:00:00      1       0          0       1 9.02 13.635
## 3 2011-01-01 02:00:00      1       0          0       1 9.02 13.635
## 4 2011-01-01 03:00:00      1       0          0       1 9.84 14.395
## 5 2011-01-01 04:00:00      1       0          0       1 9.84 14.395
## 6 2011-01-01 05:00:00      1       0          0       2 9.84 12.880
##   humidity windspeed casual registered count  time
## 1       81    0.0000      3         13    16 sleep
## 2       80    0.0000      8         32    40 sleep
## 3       80    0.0000      5         27    32 sleep
## 4       75    0.0000      3         10    13 sleep
## 5       75    0.0000      0          1     1 sleep
## 6       75    6.0032      0          1     1 sleep
```

2. Exploratory Data Analysis

Pairs Scattered Plots

```
library(gpairs)
gpairs(train[,-c(1,2,3,4,5,13)])
```



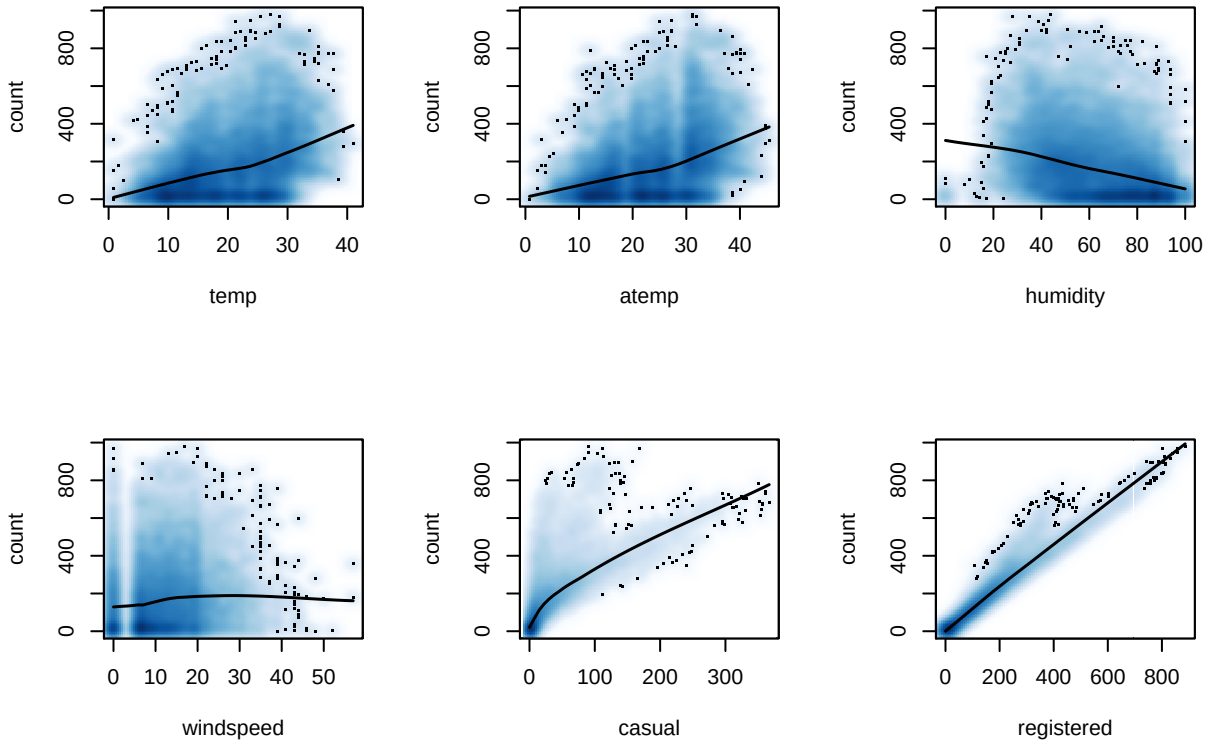
From the pairs plots above, we can see that the temp and atemp has a strong positive correlation and we might want to use only one of these two variables for our prediction.

Also, we can see that the correlation between registered counts and total counts is stronger than the correlation between casual counts and total counts. We might want to investigate these two variables more to determine whether we want to predict them separately or together.

As we can see in the scattered plots above, there are too many data points and the scattered plot just turnded out to be a huge blurb providing us very few information. Thus, I am going to plot 2D density smoothing plots to see where do most data points line.

2D Density Smooth Plots

```
#create a smoothing plot function that loops over all pairs plots between "count" and other continous v

scatter.func = function(){
  for (i in c(6,7,8,9,10,11)) {
    x.lab = names(train)[i]
    smoothScatter(x = train[, x.lab], y = train$count, xlab = x.lab, ylab = "count")
    loess = loess.smooth(x = train[, x.lab], y = train$count)
    lines(loess,lwd = 1.5)

  }
}

par(mfrow = c(2,3))
scatter.func()
```



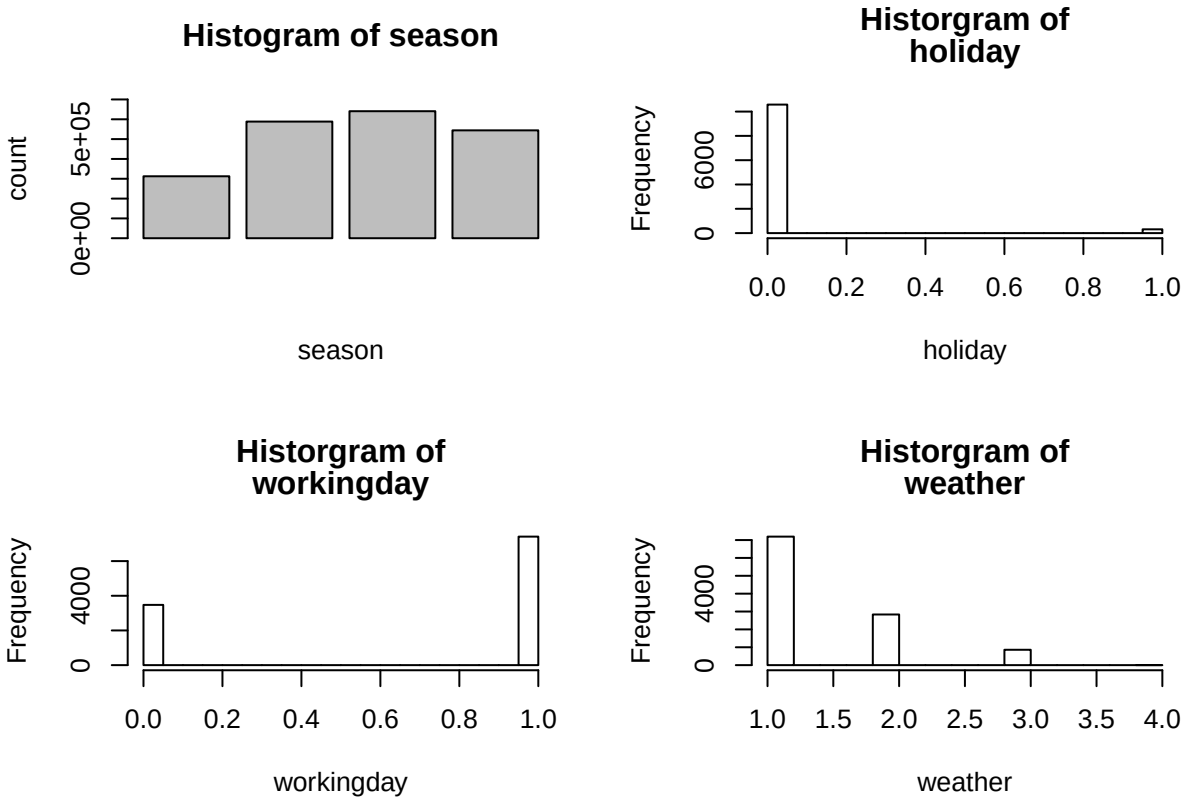
Now, after plotting out the 2D density plots and add the loess smoothing curve on top, we have a much better idea of the relationship of these variables. As we can see from the plots, temp, atemp, casual and registered all have a positive relationship with count. Humidity has a negative relationship with count, which is reasonable. The loess line between windspeed and count is almost flat, which indicates that changes in windspeed doesn't really change the bike rental. Therefore, we could conclude that there is only a very weak relationship between windspeed and bike rental counts.

Histograms

Besides all these continuous variables above, we still haven't investigated the relationships between categorical variables. I am going to include some histograms below to further explore these relationships.

```
hist.func = function(){
  for (i in c(3,4,5)) {
    lab = names(train)[i]
    hist(x = train[, lab], xlab= lab, main = c("Historgram of", lab))
  }
}


#frequency of season doesn't tell us anything.The following function allows us to compute the count of
season.func = function(){
  list_of_season = vector("integer", length = 4)
  for (i in c(1:nrow(train))) {
    for (n in c(1:4)) {
      if (train$season[i] == n) {
        list_of_season[n] = list_of_season[n] + train$count[i]
    }
    }
  }
```

```
  return(list_of_season)
}

par(mfrow = c(2,2))
list_of_season = season.func()
barplot(list_of_season, xlab = "season", ylab = "count", main = "Histogram of season", ylim = c(0,700000
hist.func()
```



As we can see from the histograms above, the bike rental counts is evenly distributed across summer, fall and winter with nearly half of hourly rent in spring. The data is recorded from Jan 2011 to Dec 2012 so every season appears equal amount of times.

There are more bikes being rented on workingday and most bikes are rented on weather 1 (Clear, Few clouds, Partly cloudy, Partly cloudy).

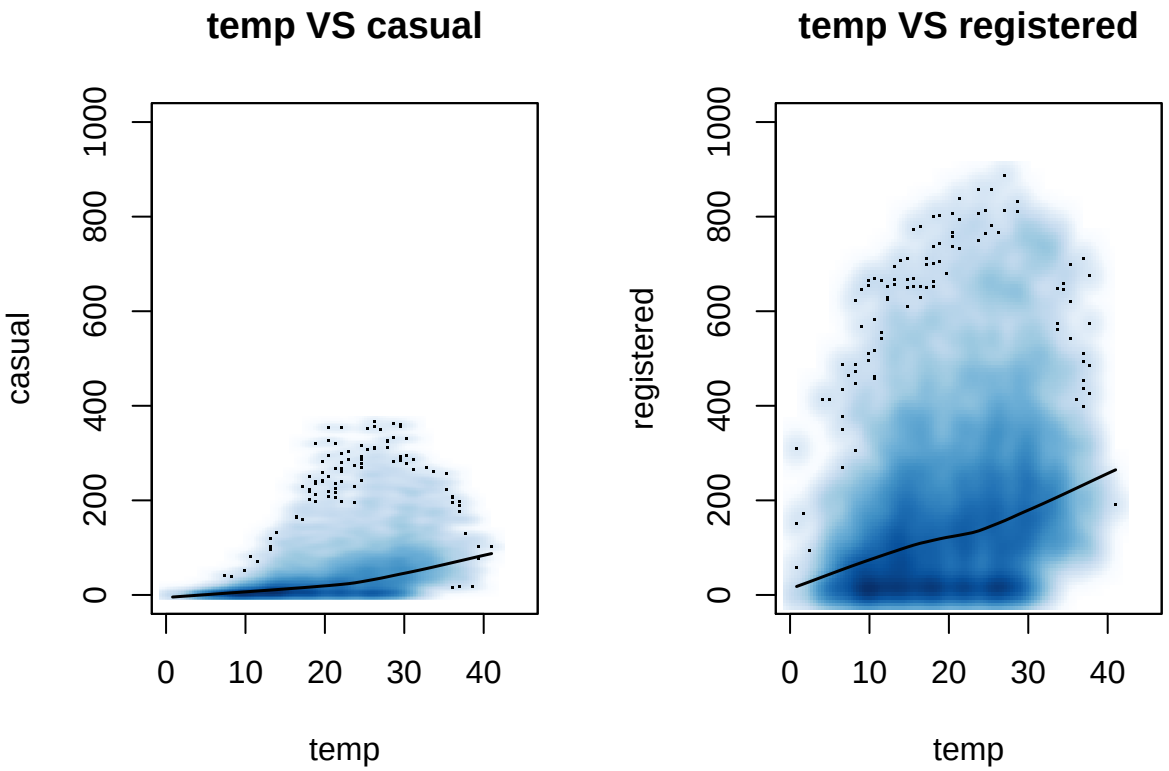
3. Choice of Response in Regression

I make the choice based on whether casual users and registered users show different renting patterns, if so, then it would be more accurate for us to fit a linear regression for each group.

We can find out through visualization. As we can see the in the pairs plot at the very beginning. "casual VS temp" pair plot is noticeably different from "registered VS temp" pair plot. I would like to draw a 2D density plot and fit a loess curve to better visualize the differences.

```
par(mfrow = c(1,2))

smoothScatter(x = train$temp, y = train$casual, ylim = c(0,1000), xlim = c(0,45), xlab = "temp", ylab =
loess.casual = loess.smooth(x = train$temp, y = train$casual)
lines(loess.casual, lwd = 1.5)
```

```
smoothScatter(x = train$temp, y = train$registered, ylim = c(0,1000),xlim = c(0,45), xlab = "temp", yla
loess.registered = loess.smooth(x = train$temp, y = train$registered)
lines(loess.registered, lwd = 1.5)
```



As we can see from the comparison above, the casual bikers are very different from the registered bikers when they react to temperature changes. Registered bikers are more sensitive to temperature changes than casual bikers. Therefore, to better predict the bike rental count, we should keep them separate and make two linear regressions.

Also, from the plot aove, we can see that the majority of data is located in low values, thus, it would be a good idea for us to take the log of counts to better fit the linear regression.

4. Regression Analysis

Basic Linear Regression Model

```
# create a "time" variable for test data as well

time_list_test = time.func(as.character(test$datetime))
test$time = unlist(time_list_test[2:6494])
```

Here, I am going to first combine weather3(light rain) and weather4 (heavy rain& snow) before I run any linear regression because there's only one data point for weather4 = 1 and if I add the interaction term between weather4 and other variables, singularity issues will occur when we add interaction term later on.

```
#predict using casual.lm and registered.lm.

#combine weather 3 and 4 by changing the row that contains weather = 4 into weather = 3
```

```
train$weather[5632] = 3

#because I need to take the log of both casual and registered bikers, I need to take out the rows where

train_nonzero = train[train$casual != 0 & train$registered != 0, ]


# I create my own test dataset of size 200, and put the rest as training data
sample_num = sample(c(1:nrow(train_nonzero)), 200)
my_test = train[sample_num,]
my_train_casual = train[-sample_num, -c(11,12)]
my_train_registered = train[-sample_num, -c(10,12)]



# now I fit a linear regression model for casual bikers and registered bikers separetely, my_test is ex
casual.lm = lm(log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) + as.fa

print("summaries for causual.lm")
```

```
## [1] "summaries for causual.lm"
```

```
summary(casual.lm)
```

```
##
## Call:
## lm(formula = log(casual + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time), data = my_train_casual)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2826 -0.4358  0.0557  0.4987  3.1092
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.9537352  0.0410016  47.650  < 2e-16 ***
## as.factor(season)2       0.3758945  0.0259999  14.458  < 2e-16 ***
## as.factor(season)3       0.0672205  0.0332265   2.023   0.0431 *
## as.factor(season)4       0.4414266  0.0217039  20.339  < 2e-16 ***
## as.factor(holiday)1     -0.2383440  0.0437934  -5.442 5.37e-08 ***
## as.factor(workingday)1 -0.7361281  0.0157054 -46.871  < 2e-16 ***
## as.factor(weather)2     -0.0220929  0.0173816  -1.271   0.2037
## as.factor(weather)3     -0.4935070  0.0289460 -17.049  < 2e-16 ***
## temp                     0.0470178  0.0056360   8.342  < 2e-16 ***
## atemp                    0.0388061  0.0049338   7.865 4.04e-15 ***
## humidity                -0.0108863  0.0004556 -23.892  < 2e-16 ***
## as.factor(time)sleep    -1.3614380  0.0221489 -61.468  < 2e-16 ***
## as.factor(time)traffic   0.5613002  0.0265733  21.123  < 2e-16 ***
## as.factor(time)work      0.5181715  0.0208101  24.900  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7307 on 10672 degrees of freedom
```

```
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7597
## F-statistic:  2600 on 13 and 10672 DF,  p-value: < 2.2e-16
```

```
registered.lm = lm(log(registered + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday)

print("summaries for registered.lm")
```

```
## [1] "summaries for registered.lm"
```

```
summary(registered.lm)
```

```
##
## Call:
## lm(formula = log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + windspeed + as.factor(time), data = my_train_registered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8325 -0.4690  0.0355  0.4755  2.6418
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.0470756  0.0492338  82.201  < 2e-16 ***
## as.factor(season)2       0.2555724  0.0277648   9.205  < 2e-16 ***
## as.factor(season)3       0.1928813  0.0355769   5.422 6.04e-08 ***
## as.factor(season)4       0.5397551  0.0232216  23.244  < 2e-16 ***
## as.factor(holiday)1     -0.0436655  0.0467598  -0.934 0.350414
## as.factor(workingday)1   0.0737833  0.0167685   4.400 1.09e-05 ***
## as.factor(weather)2     -0.0200312  0.0185775  -1.078 0.280947
## as.factor(weather)3     -0.4576041  0.0311654 -14.683  < 2e-16 ***
## temp                     0.0257675  0.0061362   4.199 2.70e-05 ***
## atemp                    0.0082317  0.0053804   1.530 0.126063
## humidity                -0.0048231  0.0005023  -9.602  < 2e-16 ***
## windspeed               -0.0038265  0.0010222  -3.743 0.000183 ***
## as.factor(time)sleep    -1.8415232  0.0236563 -77.845  < 2e-16 ***
## as.factor(time)traffic   0.9672097  0.0284047  34.051  < 2e-16 ***
## as.factor(time)work      0.3239768  0.0222529  14.559  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7802 on 10671 degrees of freedom
## Multiple R-squared:  0.689,  Adjusted R-squared:  0.6886
## F-statistic:  1688 on 14 and 10671 DF,  p-value: < 2.2e-16
```

```
#fit a linear regression model without separating casual and registered bikers
total.lm = lm(log(count + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) + as.facto

print("summaries for total.lm")
```

```
## [1] "summaries for total.lm"
```

```
summary(total.lm)
```

```
##
## Call:
## lm(formula = log(count + 1) ~ as.factor(season) + as.factor(holiday) +
```

```
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time), data = train[-sample_num, -c(10,
##     11)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1139 -0.4442  0.0397  0.4599  2.6707
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.1102601  0.0426803  96.303  < 2e-16 ***
## as.factor(season)2       0.2896710  0.0270644  10.703  < 2e-16 ***
## as.factor(season)3       0.1976799  0.0345869   5.715 1.12e-08 ***
## as.factor(season)4       0.5362481  0.0225925  23.736  < 2e-16 ***
## as.factor(holiday)1     -0.0805759  0.0455864  -1.768 0.077166 .
## as.factor(workingday)1 -0.0766438  0.0163484  -4.688 2.79e-06 ***
## as.factor(weather)2     -0.0266268  0.0180932  -1.472 0.141146
## as.factor(weather)3     -0.4919958  0.0301311 -16.329  < 2e-16 ***
## temp                     0.0250286  0.0058667   4.266 2.01e-05 ***
## atemp                    0.0170281  0.0051358   3.316 0.000918 ***
## humidity                -0.0053779  0.0004743 -11.339  < 2e-16 ***
## as.factor(time)sleep    -1.8167829  0.0230557 -78.800  < 2e-16 ***
## as.factor(time)traffic   0.9331688  0.0276613  33.736  < 2e-16 ***
## as.factor(time)work      0.3734003  0.0216621  17.237  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7606 on 10672 degrees of freedom
## Multiple R-squared:  0.7127, Adjusted R-squared:  0.7123
## F-statistic:  2036 on 13 and 10672 DF,  p-value: < 2.2e-16
```

I created two very basic linear regression models above. From the summaries above, we can see that, overall, our models separating casual and registered bikers (R^2 = 0.7603, 0.6901) perform better than our model not separating the group (R^2 = 0.7139).

Now I am going to explore some interactions between different variables and pick the best model before I use step function and cross validation to further select my variables.

I am going to first incluse the interactions between numerical and categorical variables, if the RSS decreases significantly, I will continue to include interactions between all numericals & numericals and catrgorical & categorical. If not, I am going to stop and start variable selection.

Explore the interactions between different explanatory variables

Include interactions between numerical & categorical

```
casual.lm.numcat = lm(log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +
as.factor(holiday):temp + as.factor(holiday):humidity + as.factor(holiday):atemp +
as.factor(workingday):temp + as.factor(workingday):humidity + as.factor(workingday):atemp +
as.factor(weather):temp + as.factor(weather):humidity + as.factor(weather):atemp +
as.factor(time):temp + as.factor(time):humidity + as.factor(time):atemp
  , data = my_train_casual)

summary(casual.lm.numcat)
```

```
##
## Call:
## lm(formula = log(casual + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):temp + as.factor(season):humidity +
##     as.factor(season):atemp + as.factor(holiday):temp + as.factor(holiday):humidity +
##     as.factor(holiday):atemp + as.factor(workingday):temp + as.factor(workingday):humidity +
##     as.factor(workingday):atemp + as.factor(weather):temp + as.factor(weather):humidity +
##     as.factor(weather):atemp + as.factor(time):temp + as.factor(time):humidity +
##     as.factor(time):atemp, data = my_train_casual)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3381 -0.4127  0.0441  0.4635  2.9988
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.2801171  0.1009306   2.775 0.005524 **
## as.factor(season)2              1.8429293  0.1117102  16.497  < 2e-16 ***
## as.factor(season)3              2.9859468  0.1828563  16.329  < 2e-16 ***
## as.factor(season)4              1.2023965  0.0921007  13.055  < 2e-16 ***
## as.factor(holiday)1            -0.5192844  0.1719091  -3.021 0.002528 **
## as.factor(workingday)1         -0.5428750  0.0685158  -7.923 2.54e-15 ***
## as.factor(weather)2             0.0669142  0.0784748   0.853 0.393853
## as.factor(weather)3            -0.6949181  0.1521411  -4.568 4.99e-06 ***
## temp                            0.1046323  0.0232526   4.500 6.87e-06 ***
## atemp                           0.0520476  0.0204999   2.539 0.011133 *
## humidity                       -0.0029573  0.0012982  -2.278 0.022741 *
## as.factor(time)sleep           -0.3776757  0.0983321  -3.841 0.000123 ***
## as.factor(time)traffic          1.1605959  0.1119196  10.370  < 2e-16 ***
## as.factor(time)work             1.4177966  0.0882810  16.060  < 2e-16 ***
## as.factor(season)2:temp        -0.1240601  0.0215542  -5.756 8.87e-09 ***
## as.factor(season)3:temp        -0.0828102  0.0168745  -4.907 9.37e-07 ***
## as.factor(season)4:temp        -0.0761795  0.0222417  -3.425 0.000617 ***
## as.factor(season)2:humidity    -0.0105980  0.0010308 -10.281  < 2e-16 ***
## as.factor(season)3:humidity    -0.0137822  0.0013241 -10.408  < 2e-16 ***
## as.factor(season)4:humidity    -0.0117444  0.0010565 -11.116  < 2e-16 ***
## as.factor(season)2:atemp        0.0617064  0.0193301   3.192 0.001416 **
## as.factor(season)3:atemp       -0.0072853  0.0140847  -0.517 0.604995
## as.factor(season)4:atemp        0.0534378  0.0195316   2.736 0.006230 **
## as.factor(holiday)1:temp        0.0016193  0.0477921   0.034 0.972971
## as.factor(holiday)1:humidity    0.0046080  0.0025338   1.819 0.068997 .
## as.factor(holiday)1:atemp       0.0013850  0.0439223   0.032 0.974846
## as.factor(workingday)1:temp     0.0440642  0.0154278   2.856 0.004296 **
## as.factor(workingday)1:humidity 0.0040098  0.0007931   5.056 4.35e-07 ***
## as.factor(workingday)1:atemp   -0.0563537  0.0140843  -4.001 6.35e-05 ***
## as.factor(weather)2:temp        0.0018540  0.0124623   0.149 0.881738
## as.factor(weather)3:temp        0.0053736  0.0223430   0.241 0.809944
## as.factor(weather)2:humidity   -0.0022941  0.0009210  -2.491 0.012762 *
## as.factor(weather)3:humidity   -0.0011811  0.0014532  -0.813 0.416372
## as.factor(weather)2:atemp       0.0012417  0.0114290   0.109 0.913484
## as.factor(weather)3:atemp       0.0075137  0.0205331   0.366 0.714426
## temp:as.factor(time)sleep      -0.0409942  0.0181256  -2.262 0.023737 *
## temp:as.factor(time)traffic    -0.0356484  0.0196678  -1.813 0.069933 .
```

```
## temp:as.factor(time)work        -0.0551423  0.0163179  -3.379 0.000729 ***
## humidity:as.factor(time)sleep    0.0025238  0.0012389   2.037 0.041662 *
## humidity:as.factor(time)traffic -0.0041210  0.0013529  -3.046 0.002324 **
## humidity:as.factor(time)work    -0.0059526  0.0010861  -5.480 4.34e-08 ***
## atemp:as.factor(time)sleep      -0.0184405  0.0165258  -1.116 0.264509
## atemp:as.factor(time)traffic     0.0156072  0.0179888   0.868 0.385629
## atemp:as.factor(time)work        0.0240854  0.0148513   1.622 0.104883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6927 on 10642 degrees of freedom
## Multiple R-squared:  0.7849, Adjusted R-squared:  0.7841
## F-statistic: 903.3 on 43 and 10642 DF,  p-value: < 2.2e-16
```

From the summaries above, we can see that for our very basic model without any interactions (causual.lm), the multiple r square is 0.7602. After adding all interactions between numerical and categorical variables, the multiple r square becomes 0.7869. This is a reletively noticable improvement.

Now I am going to do the same thing for registered bikers.

```
registered.lm.numcat = lm(log(registered+1) ~ as.factor(season) + as.factor(holiday) + as.factor(workin
as.factor(holiday):temp + as.factor(holiday):humidity + as.factor(holiday):atemp +
as.factor(workingday):temp + as.factor(workingday):humidity +  as.factor(workingday):atemp +
as.factor(time):temp + as.factor(time):humidity + as.factor(time):atemp , data = my_train_registered)


summary(registered.lm.numcat)
```

```
##
## Call:
## lm(formula = log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):temp + as.factor(season):humidity +
##     as.factor(season):atemp + as.factor(holiday):temp + as.factor(holiday):humidity +
##     as.factor(holiday):atemp + as.factor(workingday):temp + as.factor(workingday):humidity +
##     as.factor(workingday):atemp + as.factor(time):temp + as.factor(time):humidity +
##     as.factor(time):atemp, data = my_train_registered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9751 -0.4570  0.0295  0.4626  2.4827
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.2897166  0.1104884  29.774  < 2e-16 ***
## as.factor(season)2              0.6962087  0.1225154   5.683 1.36e-08 ***
## as.factor(season)3              1.9212308  0.2019458   9.514  < 2e-16 ***
## as.factor(season)4              1.3644634  0.1018263  13.400  < 2e-16 ***
## as.factor(holiday)1             0.0198062  0.1912800   0.104 0.917532
## as.factor(workingday)1          0.1154919  0.0759986   1.520 0.128626
## as.factor(weather)2            -0.0244178  0.0184509  -1.323 0.185732
## as.factor(weather)3            -0.4617745  0.0308433 -14.972  < 2e-16 ***
## temp                            0.0522579  0.0244201   2.140 0.032381 *
## atemp                           0.0247562  0.0214424   1.155 0.248302
## humidity                       -0.0056035  0.0013712  -4.087 4.41e-05 ***
## as.factor(time)sleep           -1.6692547  0.1089954 -15.315  < 2e-16 ***
```

```
## as.factor(time)traffic           1.2132638  0.1244655   9.748  < 2e-16 ***
## as.factor(time)work              0.6084155  0.0978055   6.221 5.14e-10 ***
## as.factor(season)2:temp         -0.0220490  0.0239813  -0.919 0.357894
## as.factor(season)3:temp         -0.0331446  0.0187727  -1.766 0.077496 .
## as.factor(season)4:temp         -0.0058463  0.0247588  -0.236 0.813337
## as.factor(season)2:humidity     -0.0051215  0.0011082  -4.621 3.86e-06 ***
## as.factor(season)3:humidity     -0.0082220  0.0013977  -5.882 4.16e-09 ***
## as.factor(season)4:humidity     -0.0063451  0.0011582  -5.479 4.38e-08 ***
## as.factor(season)2:atemp         0.0060208  0.0215124   0.280 0.779578
## as.factor(season)3:atemp        -0.0175944  0.0156686  -1.123 0.261501
## as.factor(season)4:atemp        -0.0206807  0.0217369  -0.951 0.341419
## as.factor(holiday)1:temp        -0.0636881  0.0532354  -1.196 0.231587
## as.factor(holiday)1:humidity     0.0008622  0.0028198   0.306 0.759775
## as.factor(holiday)1:atemp        0.0520067  0.0489272   1.063 0.287833
## as.factor(workingday)1:temp     -0.0098879  0.0171727  -0.576 0.564769
## as.factor(workingday)1:humidity  0.0032683  0.0008777   3.724 0.000197 ***
## as.factor(workingday)1:atemp    -0.0017689  0.0156794  -0.113 0.910181
## temp:as.factor(time)sleep        0.0161846  0.0189513   0.854 0.393120
## temp:as.factor(time)traffic     -0.0123688  0.0212155  -0.583 0.559901
## temp:as.factor(time)work        -0.0233663  0.0170397  -1.371 0.170315
## humidity:as.factor(time)sleep    0.0029902  0.0013759   2.173 0.029788 *
## humidity:as.factor(time)traffic  0.0020714  0.0015055   1.376 0.168880
## humidity:as.factor(time)work     0.0033353  0.0012071   2.763 0.005737 **
## atemp:as.factor(time)sleep      -0.0289357  0.0172967  -1.673 0.094376 .
## atemp:as.factor(time)traffic    -0.0057944  0.0193933  -0.299 0.765112
## atemp:as.factor(time)work       -0.0006507  0.0154740  -0.042 0.966460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7719 on 10648 degrees of freedom
## Multiple R-squared:  0.6962, Adjusted R-squared:  0.6952
## F-statistic: 659.6 on 37 and 10648 DF,  p-value: < 2.2e-16
```

From the summaries above, we can see that for our very basic model without any interactions (registered.lm), the multiple r square is 0.6901. After adding all interactions between numerical and categorical variables, the multiple r square becomes 0.6993. Adding interaction terms does not improve model that much.

By comparing the p-values of every explanatory variables in registered.lm.numcat with p-values of casual.lm.numcat, we can see a few differences between registered bikers and casual bikers:

- registered bikers are less sensitive to whether the day is a working day or not
- registered bikers are less sensitive to change in atemp("feels like" temperature in Celsius)
- registered bikers are less sensitive to time (sleep, work, traffic, afterwork)

This explains why adding the interaction terms for some categorical variables above doesn't improve R square of registered bikers that much.

Also notice that there are way more significant terms on casual.numcat than in registered.numcat, this also explains why adding interaction isn't neccessary for registered bikers. This also furthur validates the differences in how these two groups of people rent bikes.

So up to this point, I only evaluate the model based on the multiple R squared. In the following section, I am going to predict the count using my own test data and see whether there's improvement in prediction after the modification of model.

Predict on test data and compare RSS

To see the effect of adding interaction on our prediction. I am going to predict the hourly bike rental counts using both modifided model and the basic model and then compare their RSS.

```
predicted.casual.numcat = exp(predict(casual.lm.numcat, newdata = my_test[-c(10,11,12)]) -1)
predicted.registered.numcat = exp(predict(registered.lm.numcat, newdata = my_test[-c(10,11,12)]) -1)
predicted.sum.numcat = predicted.casual.numcat + predicted.registered.numcat
true.value = my_test$count
RSS.numcat = sum((true.value-predicted.sum.numcat)^2)



predicted.casual = exp(predict(casual.lm, newdata = my_test[,-c(10,11,12)]) - 1)
predicted.registered= exp(predict(registered.lm, newdata = my_test[,-c(10,11,12)]) - 1)
predicted.sum = predicted.casual + predicted.registered
RSS = sum((true.value-predicted.sum)^2)



(RSS.numcat - RSS)/RSS
```

```
## [1] -0.01472302
```

As we can see from the RSS values above. The RSS decreases around 1.6% after I include all interaction terms, which means that adding interaction terms doesn't really help that much. However, we do notice some improvements in nultiple R squared and the number of significant terms for casual bikers.

I decided to revert back to the basic model and include the interactions between categorical &categorical variables and see if that will gives us more insights about the casual or registered bikers.

Include interactions between categorical & categorical

```
casual.lm.catcat = lm(log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +

print("summary of casual.lm.catcat")
```

```
## [1] "summary of casual.lm.catcat"
```

```
summary(casual.lm.catcat)
```

```
##
## Call:
## lm(formula = log(casual + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(holiday) +
##     as.factor(season):as.factor(workingday) + as.factor(season):as.factor(weather) +
##     as.factor(season):as.factor(time) + as.factor(holiday):as.factor(workingday) +
##     as.factor(holiday):as.factor(weather) + as.factor(holiday):as.factor(time) +
##     as.factor(workingday):as.factor(weather) + as.factor(workingday):as.factor(time) +
##     as.factor(weather):as.factor(time), data = my_train_casual)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5126 -0.4248  0.0468  0.4741  3.2805
##
## Coefficients: (1 not defined because of singularities)
```

```
##                                              Estimate Std. Error
## (Intercept)                                 1.3835906  0.0587455
## as.factor(season)2                          0.8547103  0.0599944
## as.factor(season)3                          0.7148412  0.0632050
## as.factor(season)4                          0.7849115  0.0579612
## as.factor(holiday)1                        -0.4905402  0.1481055
## as.factor(workingday)1                     -0.3032056  0.0468478
## as.factor(weather)2                        -0.0131981  0.0557480
## as.factor(weather)3                        -0.7225901  0.0901727
## temp                                        0.0519063  0.0055261
## atemp                                       0.0349799  0.0048286
## humidity                                   -0.0106193  0.0004499
## as.factor(time)sleep                       -0.6113801  0.0561850
## as.factor(time)traffic                      0.6242146  0.0675689
## as.factor(time)work                         1.0688088  0.0521593
## as.factor(season)2:as.factor(holiday)1     -0.1703641  0.1407979
## as.factor(season)3:as.factor(holiday)1      0.3699031  0.1190187
## as.factor(season)4:as.factor(holiday)1      0.0150856  0.1187613
## as.factor(season)2:as.factor(workingday)1  -0.1843290  0.0432112
## as.factor(season)3:as.factor(workingday)1  -0.1711553  0.0433668
## as.factor(season)4:as.factor(workingday)1  -0.1907450  0.0434966
## as.factor(season)2:as.factor(weather)2     -0.0525476  0.0453328
## as.factor(season)3:as.factor(weather)2      0.0420954  0.0466617
## as.factor(season)4:as.factor(weather)2     -0.1009620  0.0445191
## as.factor(season)2:as.factor(weather)3      0.1635533  0.0735153
## as.factor(season)3:as.factor(weather)3      0.2576123  0.0752626
## as.factor(season)4:as.factor(weather)3      0.0471056  0.0739979
## as.factor(season)2:as.factor(time)sleep    -0.7025988  0.0599966
## as.factor(season)3:as.factor(time)sleep    -0.9501388  0.0600976
## as.factor(season)4:as.factor(time)sleep    -0.4362002  0.0600072
## as.factor(season)2:as.factor(time)traffic  -0.2020543  0.0726221
## as.factor(season)3:as.factor(time)traffic  -0.4247876  0.0726683
## as.factor(season)4:as.factor(time)traffic  -0.0774523  0.0726952
## as.factor(season)2:as.factor(time)work     -0.3091941  0.0562700
## as.factor(season)3:as.factor(time)work     -0.6176856  0.0564303
## as.factor(season)4:as.factor(time)work     -0.1237751  0.0562828
## as.factor(holiday)1:as.factor(workingday)1         NA         NA
## as.factor(holiday)1:as.factor(weather)2     0.1154590  0.0993207
## as.factor(holiday)1:as.factor(weather)3     0.4844556  0.2140408
## as.factor(holiday)1:as.factor(time)sleep    0.0770677  0.1392140
## as.factor(holiday)1:as.factor(time)traffic  0.2599335  0.1653630
## as.factor(holiday)1:as.factor(time)work     0.1039211  0.1285048
## as.factor(workingday)1:as.factor(weather)2 -0.0499441  0.0356739
## as.factor(workingday)1:as.factor(weather)3 -0.0622979  0.0610682
## as.factor(workingday)1:as.factor(time)sleep -0.3566631  0.0468592
## as.factor(workingday)1:as.factor(time)traffic  0.1074048  0.0568831
## as.factor(workingday)1:as.factor(time)work -0.4567193  0.0440097
## as.factor(weather)2:as.factor(time)sleep   -0.0219937  0.0508393
## as.factor(weather)3:as.factor(time)sleep    0.3952291  0.0815786
## as.factor(weather)2:as.factor(time)traffic  0.0719786  0.0622623
## as.factor(weather)3:as.factor(time)traffic  0.2435678  0.0922985
## as.factor(weather)2:as.factor(time)work     0.1040235  0.0478624
## as.factor(weather)3:as.factor(time)work    -0.0118948  0.0760821
##                                            t value Pr(>|t|)
```

```
## (Intercept)                                    23.552  < 2e-16 ***
## as.factor(season)2                             14.247  < 2e-16 ***
## as.factor(season)3                             11.310  < 2e-16 ***
## as.factor(season)4                             13.542  < 2e-16 ***
## as.factor(holiday)1                            -3.312 0.000929 ***
## as.factor(workingday)1                         -6.472 1.01e-10 ***
## as.factor(weather)2                            -0.237 0.812859
## as.factor(weather)3                            -8.013 1.23e-15 ***
## temp                                            9.393  < 2e-16 ***
## atemp                                           7.244 4.65e-13 ***
## humidity                                      -23.601  < 2e-16 ***
## as.factor(time)sleep                          -10.882  < 2e-16 ***
## as.factor(time)traffic                          9.238  < 2e-16 ***
## as.factor(time)work                            20.491  < 2e-16 ***
## as.factor(season)2:as.factor(holiday)1         -1.210 0.226309
## as.factor(season)3:as.factor(holiday)1          3.108 0.001889 **
## as.factor(season)4:as.factor(holiday)1          0.127 0.898923
## as.factor(season)2:as.factor(workingday)1      -4.266 2.01e-05 ***
## as.factor(season)3:as.factor(workingday)1      -3.947 7.98e-05 ***
## as.factor(season)4:as.factor(workingday)1      -4.385 1.17e-05 ***
## as.factor(season)2:as.factor(weather)2         -1.159 0.246421
## as.factor(season)3:as.factor(weather)2          0.902 0.367003
## as.factor(season)4:as.factor(weather)2         -2.268 0.023359 *
## as.factor(season)2:as.factor(weather)3          2.225 0.026119 *
## as.factor(season)3:as.factor(weather)3          3.423 0.000622 ***
## as.factor(season)4:as.factor(weather)3          0.637 0.524412
## as.factor(season)2:as.factor(time)sleep       -11.711  < 2e-16 ***
## as.factor(season)3:as.factor(time)sleep       -15.810  < 2e-16 ***
## as.factor(season)4:as.factor(time)sleep        -7.269 3.87e-13 ***
## as.factor(season)2:as.factor(time)traffic      -2.782 0.005408 **
## as.factor(season)3:as.factor(time)traffic      -5.846 5.20e-09 ***
## as.factor(season)4:as.factor(time)traffic      -1.065 0.286701
## as.factor(season)2:as.factor(time)work         -5.495 4.00e-08 ***
## as.factor(season)3:as.factor(time)work        -10.946  < 2e-16 ***
## as.factor(season)4:as.factor(time)work         -2.199 0.027888 *
## as.factor(holiday)1:as.factor(workingday)1         NA       NA
## as.factor(holiday)1:as.factor(weather)2         1.162 0.245064
## as.factor(holiday)1:as.factor(weather)3         2.263 0.023632 *
## as.factor(holiday)1:as.factor(time)sleep        0.554 0.579870
## as.factor(holiday)1:as.factor(time)traffic      1.572 0.116004
## as.factor(holiday)1:as.factor(time)work         0.809 0.418709
## as.factor(workingday)1:as.factor(weather)2     -1.400 0.161537
## as.factor(workingday)1:as.factor(weather)3     -1.020 0.307687
## as.factor(workingday)1:as.factor(time)sleep    -7.611 2.94e-14 ***
## as.factor(workingday)1:as.factor(time)traffic   1.888 0.059031 .
## as.factor(workingday)1:as.factor(time)work    -10.378  < 2e-16 ***
## as.factor(weather)2:as.factor(time)sleep       -0.433 0.665306
## as.factor(weather)3:as.factor(time)sleep        4.845 1.29e-06 ***
## as.factor(weather)2:as.factor(time)traffic      1.156 0.247685
## as.factor(weather)3:as.factor(time)traffic      2.639 0.008329 **
## as.factor(weather)2:as.factor(time)work         2.173 0.029773 *
## as.factor(weather)3:as.factor(time)work        -0.156 0.875767
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7097 on 10635 degrees of freedom
## Multiple R-squared:  0.7744, Adjusted R-squared:  0.7733
## F-statistic:   730 on 50 and 10635 DF,  p-value: < 2.2e-16
```

```
registered.lm.catcat = lm(log(registered + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(worki
print("summary of registered.lm.catcat")
```

```
## [1] "summary of registered.lm.catcat"
```

```
summary(registered.lm.catcat)
```

```
##
## Call:
## lm(formula = log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(holiday) +
##     as.factor(season):as.factor(workingday) + as.factor(season):as.factor(weather) +
##     as.factor(season):as.factor(time) + as.factor(holiday):as.factor(workingday) +
##     as.factor(holiday):as.factor(weather) + as.factor(holiday):as.factor(time) +
##     as.factor(workingday):as.factor(weather) + as.factor(workingday):as.factor(time) +
##     as.factor(weather):as.factor(time), data = my_train_registered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8451 -0.4416  0.0192  0.4411  2.7212
##
## Coefficients: (1 not defined because of singularities)
##                                                Estimate Std. Error
## (Intercept)                                   3.6309596  0.0618578
## as.factor(season)2                            0.4930367  0.0631729
## as.factor(season)3                            0.4875360  0.0665536
## as.factor(season)4                            0.6581842  0.0610320
## as.factor(holiday)1                           0.1363775  0.1559521
## as.factor(workingday)1                        0.4240568  0.0493298
## as.factor(weather)2                          -0.1021941  0.0587016
## as.factor(weather)3                          -0.5706000  0.0949500
## temp                                          0.0228953  0.0058189
## atemp                                         0.0114321  0.0050845
## humidity                                     -0.0040582  0.0004738
## as.factor(time)sleep                         -1.2992230  0.0591617
## as.factor(time)traffic                        0.4859756  0.0711486
## as.factor(time)work                           0.6387304  0.0549226
## as.factor(season)2:as.factor(holiday)1       -0.0435643  0.1482573
## as.factor(season)3:as.factor(holiday)1       -0.1242258  0.1253243
## as.factor(season)4:as.factor(holiday)1        0.0709534  0.1250533
## as.factor(season)2:as.factor(workingday)1    -0.1511123  0.0455005
## as.factor(season)3:as.factor(workingday)1    -0.1504860  0.0456644
## as.factor(season)4:as.factor(workingday)1    -0.1256656  0.0458011
## as.factor(season)2:as.factor(weather)2       -0.0793841  0.0477346
## as.factor(season)3:as.factor(weather)2        0.0904452  0.0491339
## as.factor(season)4:as.factor(weather)2        0.0326962  0.0468777
## as.factor(season)2:as.factor(weather)3        0.0371904  0.0774102
## as.factor(season)3:as.factor(weather)3        0.1388942  0.0792500
## as.factor(season)4:as.factor(weather)3        0.0608254  0.0779183
```

```
## as.factor(season)2:as.factor(time)sleep       -0.0933880  0.0631752
## as.factor(season)3:as.factor(time)sleep       -0.1206185  0.0632816
## as.factor(season)4:as.factor(time)sleep       -0.0392418  0.0631864
## as.factor(season)2:as.factor(time)traffic     -0.1316337  0.0764696
## as.factor(season)3:as.factor(time)traffic     -0.2956094  0.0765183
## as.factor(season)4:as.factor(time)traffic     -0.0676033  0.0765466
## as.factor(season)2:as.factor(time)work        -0.1744881  0.0592512
## as.factor(season)3:as.factor(time)work        -0.3432201  0.0594199
## as.factor(season)4:as.factor(time)work        -0.0542030  0.0592647
## as.factor(holiday)1:as.factor(workingday)1            NA         NA
## as.factor(holiday)1:as.factor(weather)2       -0.1501358  0.1045827
## as.factor(holiday)1:as.factor(weather)3        0.0391880  0.2253806
## as.factor(holiday)1:as.factor(time)sleep      -0.4446399  0.1465895
## as.factor(holiday)1:as.factor(time)traffic     0.2650796  0.1741239
## as.factor(holiday)1:as.factor(time)work       -0.0494072  0.1353130
## as.factor(workingday)1:as.factor(weather)2    -0.0131686  0.0375639
## as.factor(workingday)1:as.factor(weather)3    -0.1185367  0.0643036
## as.factor(workingday)1:as.factor(time)sleep   -0.7638616  0.0493418
## as.factor(workingday)1:as.factor(time)traffic  0.8166620  0.0598968
## as.factor(workingday)1:as.factor(time)work    -0.2788443  0.0463414
## as.factor(weather)2:as.factor(time)sleep       0.1168964  0.0535328
## as.factor(weather)3:as.factor(time)sleep       0.3674049  0.0859006
## as.factor(weather)2:as.factor(time)traffic     0.1109985  0.0655609
## as.factor(weather)3:as.factor(time)traffic     0.1458257  0.0971884
## as.factor(weather)2:as.factor(time)work        0.0698195  0.0503981
## as.factor(weather)3:as.factor(time)work        0.0362665  0.0801129
##                                              t value Pr(>|t|)
## (Intercept)                                   58.698  < 2e-16 ***
## as.factor(season)2                             7.805 6.53e-15 ***
## as.factor(season)3                             7.325 2.55e-13 ***
## as.factor(season)4                            10.784  < 2e-16 ***
## as.factor(holiday)1                            0.874 0.381875
## as.factor(workingday)1                         8.596  < 2e-16 ***
## as.factor(weather)2                           -1.741 0.081728 .
## as.factor(weather)3                           -6.009 1.92e-09 ***
## temp                                           3.935 8.39e-05 ***
## atemp                                          2.248 0.024568 *
## humidity                                      -8.566  < 2e-16 ***
## as.factor(time)sleep                         -21.961  < 2e-16 ***
## as.factor(time)traffic                         6.830 8.93e-12 ***
## as.factor(time)work                           11.630  < 2e-16 ***
## as.factor(season)2:as.factor(holiday)1        -0.294 0.768884
## as.factor(season)3:as.factor(holiday)1        -0.991 0.321593
## as.factor(season)4:as.factor(holiday)1         0.567 0.570465
## as.factor(season)2:as.factor(workingday)1     -3.321 0.000900 ***
## as.factor(season)3:as.factor(workingday)1     -3.295 0.000986 ***
## as.factor(season)4:as.factor(workingday)1     -2.744 0.006085 **
## as.factor(season)2:as.factor(weather)2        -1.663 0.096336 .
## as.factor(season)3:as.factor(weather)2         1.841 0.065680 .
## as.factor(season)4:as.factor(weather)2         0.697 0.485518
## as.factor(season)2:as.factor(weather)3         0.480 0.630930
## as.factor(season)3:as.factor(weather)3         1.753 0.079698 .
## as.factor(season)4:as.factor(weather)3         0.781 0.435037
## as.factor(season)2:as.factor(time)sleep       -1.478 0.139374
```

```
## as.factor(season)3:as.factor(time)sleep        -1.906 0.056669 .
## as.factor(season)4:as.factor(time)sleep        -0.621 0.534581
## as.factor(season)2:as.factor(time)traffic      -1.721 0.085210 .
## as.factor(season)3:as.factor(time)traffic      -3.863 0.000113 ***
## as.factor(season)4:as.factor(time)traffic      -0.883 0.377166
## as.factor(season)2:as.factor(time)work         -2.945 0.003238 **
## as.factor(season)3:as.factor(time)work         -5.776 7.86e-09 ***
## as.factor(season)4:as.factor(time)work         -0.915 0.360427
## as.factor(holiday)1:as.factor(workingday)1         NA       NA
## as.factor(holiday)1:as.factor(weather)2        -1.436 0.151154
## as.factor(holiday)1:as.factor(weather)3         0.174 0.861967
## as.factor(holiday)1:as.factor(time)sleep       -3.033 0.002425 **
## as.factor(holiday)1:as.factor(time)traffic      1.522 0.127948
## as.factor(holiday)1:as.factor(time)work        -0.365 0.715020
## as.factor(workingday)1:as.factor(weather)2     -0.351 0.725922
## as.factor(workingday)1:as.factor(weather)3     -1.843 0.065300 .
## as.factor(workingday)1:as.factor(time)sleep   -15.481  < 2e-16 ***
## as.factor(workingday)1:as.factor(time)traffic  13.634  < 2e-16 ***
## as.factor(workingday)1:as.factor(time)work     -6.017 1.83e-09 ***
## as.factor(weather)2:as.factor(time)sleep        2.184 0.029010 *
## as.factor(weather)3:as.factor(time)sleep        4.277 1.91e-05 ***
## as.factor(weather)2:as.factor(time)traffic      1.693 0.090474 .
## as.factor(weather)3:as.factor(time)traffic      1.500 0.133530
## as.factor(weather)2:as.factor(time)work         1.385 0.165972
## as.factor(weather)3:as.factor(time)work         0.453 0.650779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7473 on 10635 degrees of freedom
## Multiple R-squared:  0.7156, Adjusted R-squared:  0.7142
## F-statistic: 535.1 on 50 and 10635 DF,  p-value: < 2.2e-16
```

If we compare the multiple R square of the categorical & categorical interaction with numerical & categorical interaction. We will notice that for casual bikers, adding num&cat interaction improves multiple R square and for registered bikers, adding cat&cat interaction improves multiple R square noticably.

In the following section, I am going to use casual.numcat and registered.catcat to predict the total count of bike rental.

Predict on test data and compare RSS for the combined model using both cat&cat and num&cat interaction

```
predicted.registered.catcat = exp(predict.lm(registered.lm.catcat, newdata = my_test[,-c(10,11,12)]) -
```

```
## Warning in predict.lm(registered.lm.catcat, newdata = my_test[, -c(10,
## 11, : prediction from a rank-deficient fit may be misleading
```

```
predicted.sum.combined = predicted.casual.numcat + predicted.registered.catcat
true.value = my_test$count

RSS.combined = sum((true.value-predicted.sum.combined)^2)
```

```
(RSS.combined - RSS)/RSS
```

```
## [1] -0.08096929
```

The RSS decreases by nearly 7% for the combined model, this is a noticable decrease in RSS and I would choose this model for now and run variable selection procedures on this model to filter out the unneccesary variables.

5. Variable Selection

For variable selection, we could either use stepwise selection or cross validation. For the purpose of practicing, we were told to use stepwise selection. But personally, I prefer cross validation because you select your model entirely based on it's performance. Whereas for stepwise selection, you select based on p-values, which might not always be as intuitive and as relavant to our problem.

```
casual.lm.short <- step(casual.lm.numcat, direction = "both")
```

```
## Start:  AIC=-7802.62
## log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +
##     as.factor(weather) + temp + atemp + humidity + as.factor(time) +
##     as.factor(season):temp + as.factor(season):humidity + as.factor(season):atemp +
##     as.factor(holiday):temp + as.factor(holiday):humidity + as.factor(holiday):atemp +
##     as.factor(workingday):temp + as.factor(workingday):humidity +
##     as.factor(workingday):atemp + as.factor(weather):temp + as.factor(weather):humidity +
##     as.factor(weather):atemp + as.factor(time):temp + as.factor(time):humidity +
##     as.factor(time):atemp
##
##                                   Df Sum of Sq    RSS     AIC
## - as.factor(weather):temp          2     0.033 5106.6 -7806.6
## - as.factor(weather):atemp         2     0.065 5106.6 -7806.5
## - as.factor(holiday):atemp         1     0.000 5106.6 -7804.6
## - as.factor(holiday):temp          1     0.001 5106.6 -7804.6
## <none>                                       5106.6 -7802.6
## - as.factor(holiday):humidity      1     1.587 5108.2 -7801.3
## - as.factor(weather):humidity      2     3.027 5109.6 -7800.3
## - temp:as.factor(time)             3     5.616 5112.2 -7796.9
## - as.factor(workingday):temp       1     3.914 5110.5 -7796.4
## - atemp:as.factor(time)            3     6.427 5113.0 -7795.2
## - as.factor(workingday):atemp      1     7.682 5114.2 -7788.6
## - as.factor(season):atemp          3    13.205 5119.8 -7781.0
## - as.factor(workingday):humidity   1    12.267 5118.8 -7779.0
## - as.factor(season):temp           3    17.607 5124.2 -7771.8
## - humidity:as.factor(time)         3    40.162 5146.7 -7724.9
## - as.factor(season):humidity       3    85.233 5191.8 -7631.7
##
## Step:  AIC=-7806.55
## log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +
##     as.factor(weather) + temp + atemp + humidity + as.factor(time) +
##     as.factor(season):temp + as.factor(season):humidity + as.factor(season):atemp +
##     as.factor(holiday):temp + as.factor(holiday):humidity + as.factor(holiday):atemp +
##     as.factor(workingday):temp + as.factor(workingday):humidity +
##     as.factor(workingday):atemp + as.factor(weather):humidity +
##     as.factor(weather):atemp + temp:as.factor(time) + humidity:as.factor(time) +
##     atemp:as.factor(time)
```

```
##
##                                  Df Sum of Sq    RSS     AIC
## - as.factor(holiday):atemp        1     0.000 5106.6 -7808.6
## - as.factor(holiday):temp         1     0.001 5106.6 -7808.6
## <none>                                        5106.6 -7806.6
## - as.factor(holiday):humidity     1     1.596 5108.2 -7805.2
## - as.factor(weather):humidity     2     3.107 5109.7 -7804.1
## + as.factor(weather):temp         2     0.033 5106.6 -7802.6
## - as.factor(workingday):temp      1     3.938 5110.5 -7800.3
## - atemp:as.factor(time)           3     6.521 5113.1 -7798.9
## - as.factor(weather):atemp        2     5.675 5112.3 -7798.7
## - temp:as.factor(time)            3     6.674 5113.3 -7798.6
## - as.factor(workingday):atemp     1     7.714 5114.3 -7792.4
## - as.factor(season):atemp         3    13.224 5119.8 -7784.9
## - as.factor(workingday):humidity  1    12.342 5118.9 -7782.8
## - as.factor(season):temp          3    17.649 5124.2 -7775.7
## - humidity:as.factor(time)        3    40.212 5146.8 -7728.7
## - as.factor(season):humidity      3    85.475 5192.1 -7635.2
##
## Step:  AIC=-7808.55
## log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +
##     as.factor(weather) + temp + atemp + humidity + as.factor(time) +
##     as.factor(season):temp + as.factor(season):humidity + as.factor(season):atemp +
##     as.factor(holiday):temp + as.factor(holiday):humidity + as.factor(workingday):temp +
##     as.factor(workingday):humidity + as.factor(workingday):atemp +
##     as.factor(weather):humidity + as.factor(weather):atemp +
##     temp:as.factor(time) + humidity:as.factor(time) + atemp:as.factor(time)
##
##                                  Df Sum of Sq    RSS     AIC
## - as.factor(holiday):temp         1     0.194 5106.8 -7810.1
## <none>                                        5106.6 -7808.6
## - as.factor(holiday):humidity     1     1.803 5108.4 -7806.8
## + as.factor(holiday):atemp        1     0.000 5106.6 -7806.6
## - as.factor(weather):humidity     2     3.108 5109.7 -7806.0
## + as.factor(weather):temp         2     0.033 5106.6 -7804.6
## - as.factor(workingday):temp      1     4.196 5110.8 -7801.8
## - atemp:as.factor(time)           3     6.520 5113.1 -7800.9
## - as.factor(weather):atemp        2     5.679 5112.3 -7800.7
## - temp:as.factor(time)            3     6.673 5113.3 -7800.6
## - as.factor(workingday):atemp     1     8.214 5114.8 -7793.4
## - as.factor(season):atemp         3    13.225 5119.8 -7786.9
## - as.factor(workingday):humidity  1    12.364 5119.0 -7784.7
## - as.factor(season):temp          3    17.663 5124.3 -7777.7
## - humidity:as.factor(time)        3    40.234 5146.8 -7730.7
## - as.factor(season):humidity      3    85.477 5192.1 -7637.2
##
## Step:  AIC=-7810.14
## log(casual + 1) ~ as.factor(season) + as.factor(holiday) + as.factor(workingday) +
##     as.factor(weather) + temp + atemp + humidity + as.factor(time) +
##     as.factor(season):temp + as.factor(season):humidity + as.factor(season):atemp +
##     as.factor(holiday):humidity + as.factor(workingday):temp +
##     as.factor(workingday):humidity + as.factor(workingday):atemp +
##     as.factor(weather):humidity + as.factor(weather):atemp +
##     temp:as.factor(time) + humidity:as.factor(time) + atemp:as.factor(time)
```

```
##
##                                  Df Sum of Sq    RSS     AIC
## <none>                                        5106.8 -7810.1
## + as.factor(holiday):temp         1     0.194 5106.6 -7808.6
## + as.factor(holiday):atemp        1     0.194 5106.6 -7808.6
## - as.factor(holiday):humidity     1     2.032 5108.8 -7807.9
## - as.factor(weather):humidity     2     3.060 5109.9 -7807.7
## + as.factor(weather):temp         2     0.032 5106.8 -7806.2
## - as.factor(workingday):temp      1     4.110 5110.9 -7803.5
## - atemp:as.factor(time)           3     6.529 5113.3 -7802.5
## - temp:as.factor(time)            3     6.660 5113.5 -7802.2
## - as.factor(weather):atemp        2     5.738 5112.5 -7802.1
## - as.factor(workingday):atemp     1     8.168 5115.0 -7795.1
## - as.factor(season):atemp         3    13.190 5120.0 -7788.6
## - as.factor(workingday):humidity  1    12.289 5119.1 -7786.5
## - as.factor(season):temp          3    17.619 5124.4 -7779.3
## - humidity:as.factor(time)        3    40.185 5147.0 -7732.4
## - as.factor(season):humidity      3    85.784 5192.6 -7638.1
```

```
print("multiple R squared after varaible selection for causal model")
```

```
## [1] "multiple R squared after varaible selection for causal model"
```

```
summary(casual.lm.short)$r.squared
```

```
## [1] 0.7849232
```

```
registered.lm.short <-  step(registered.lm.catcat, direction = "backward")
```

```
## Start:  AIC=-6173.45
## log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(holiday) +
##     as.factor(season):as.factor(workingday) + as.factor(season):as.factor(weather) +
##     as.factor(season):as.factor(time) + as.factor(holiday):as.factor(workingday) +
##     as.factor(holiday):as.factor(weather) + as.factor(holiday):as.factor(time) +
##     as.factor(workingday):as.factor(weather) + as.factor(workingday):as.factor(time) +
##     as.factor(weather):as.factor(time)
##
##
## Step:  AIC=-6173.45
## log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(holiday) +
##     as.factor(season):as.factor(workingday) + as.factor(season):as.factor(weather) +
##     as.factor(season):as.factor(time) + as.factor(holiday):as.factor(weather) +
##     as.factor(holiday):as.factor(time) + as.factor(workingday):as.factor(weather) +
##     as.factor(workingday):as.factor(time) + as.factor(weather):as.factor(time)
##
##                                           Df Sum of Sq    RSS     AIC
## - as.factor(season):as.factor(holiday)     3      1.62 5941.4 -6176.5
## - as.factor(holiday):as.factor(weather)    2      1.28 5941.1 -6175.2
## - as.factor(workingday):as.factor(weather) 2      1.90 5941.7 -6174.0
## <none>                                               5939.8 -6173.4
## - as.factor(season):as.factor(weather)     6      8.33 5948.1 -6170.5
## - atemp                                    1      2.82 5942.6 -6170.4
```

```
## - as.factor(season):as.factor(workingday)   3      8.37 5948.2 -6164.4
## - temp                                      1      8.65 5948.4 -6159.9
## - as.factor(weather):as.factor(time)        6     17.47 5957.3 -6154.1
## - as.factor(holiday):as.factor(time)        3     14.48 5954.3 -6153.4
## - as.factor(season):as.factor(time)         9     28.67 5968.5 -6140.0
## - humidity                                  1     40.98 5980.8 -6102.0
## - as.factor(workingday):as.factor(time)     3    494.41 6434.2 -5325.1
##
## Step:  AIC=-6176.53
## log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(workingday) +
##     as.factor(season):as.factor(weather) + as.factor(season):as.factor(time) +
##     as.factor(holiday):as.factor(weather) + as.factor(holiday):as.factor(time) +
##     as.factor(workingday):as.factor(weather) + as.factor(workingday):as.factor(time) +
##     as.factor(weather):as.factor(time)
##
##                                            Df Sum of Sq    RSS     AIC
## - as.factor(holiday):as.factor(weather)     2      1.69 5943.1 -6177.5
## - as.factor(workingday):as.factor(weather)  2      1.92 5943.3 -6177.1
## <none>                                                  5941.4 -6176.5
## - as.factor(season):as.factor(weather)      6      8.13 5949.6 -6173.9
## - atemp                                     1      2.83 5944.2 -6173.4
## - as.factor(season):as.factor(workingday)   3      8.39 5949.8 -6167.5
## - temp                                      1      8.66 5950.1 -6163.0
## - as.factor(weather):as.factor(time)        6     17.45 5958.9 -6157.2
## - as.factor(holiday):as.factor(time)        3     14.60 5956.0 -6156.3
## - as.factor(season):as.factor(time)         9     28.66 5970.1 -6143.1
## - humidity                                  1     41.38 5982.8 -6104.4
## - as.factor(workingday):as.factor(time)     3    494.46 6435.9 -5328.3
##
## Step:  AIC=-6177.48
## log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(workingday) +
##     as.factor(season):as.factor(weather) + as.factor(season):as.factor(time) +
##     as.factor(holiday):as.factor(time) + as.factor(workingday):as.factor(weather) +
##     as.factor(workingday):as.factor(time) + as.factor(weather):as.factor(time)
##
##                                            Df Sum of Sq    RSS     AIC
## - as.factor(workingday):as.factor(weather)  2      2.19 5945.3 -6177.5
## <none>                                                  5943.1 -6177.5
## - as.factor(season):as.factor(weather)      6      7.64 5950.8 -6175.8
## - atemp                                     1      2.77 5945.9 -6174.5
## - as.factor(season):as.factor(workingday)   3      8.38 5951.5 -6168.4
## - temp                                      1      8.78 5951.9 -6163.7
## - as.factor(holiday):as.factor(time)        3     14.01 5957.1 -6158.3
## - as.factor(weather):as.factor(time)        6     17.39 5960.5 -6158.3
## - as.factor(season):as.factor(time)         9     28.37 5971.5 -6144.6
## - humidity                                  1     40.86 5984.0 -6106.3
## - as.factor(workingday):as.factor(time)     3    495.53 6438.6 -5327.7
##
## Step:  AIC=-6177.55
## log(registered + 1) ~ as.factor(season) + as.factor(holiday) +
```

```
##     as.factor(workingday) + as.factor(weather) + temp + atemp +
##     humidity + as.factor(time) + as.factor(season):as.factor(workingday) +
##     as.factor(season):as.factor(weather) + as.factor(season):as.factor(time) +
##     as.factor(holiday):as.factor(time) + as.factor(workingday):as.factor(time) +
##     as.factor(weather):as.factor(time)
##
##                                         Df Sum of Sq    RSS     AIC
## <none>                                               5945.3 -6177.5
## - as.factor(season):as.factor(weather)   6      7.72 5953.0 -6175.7
## - atemp                                  1      2.85 5948.2 -6174.4
## - as.factor(season):as.factor(workingday) 3     8.17 5953.5 -6168.9
## - temp                                   1      8.60 5953.9 -6164.1
## - as.factor(weather):as.factor(time)     6     16.44 5961.7 -6160.0
## - as.factor(holiday):as.factor(time)     3     14.09 5959.4 -6158.3
## - as.factor(season):as.factor(time)      9     28.44 5973.7 -6144.5
## - humidity                               1     40.73 5986.0 -6106.6
## - as.factor(workingday):as.factor(time)  3    493.91 6439.2 -5330.8
```

```
print("multiple R squared after varaible selection for registered model")
```

```
## [1] "multiple R squared after varaible selection for registered model"
```

```
summary(registered.lm.short)$r.squared
```

```
## [1] 0.7152997
```

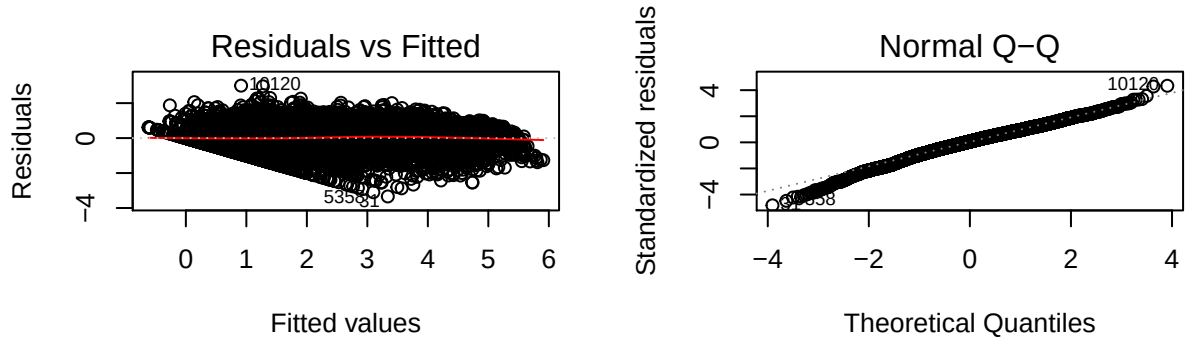
```
predicted.casual.short = exp(predict.lm(casual.lm.short, newdata = my_test[,-c(10,11,12)]) - 1)
predicted.registered.short = exp (predict.lm(registered.lm.short, newdata = my_test[,-c(10,11,12)]) - 1
predicted.sum.short = predicted.casual.short + predicted.registered.short
true.value = my_test$count
RSS.short = sum((true.value-predicted.sum.short)^2)


(RSS.short - RSS)/RSS
```

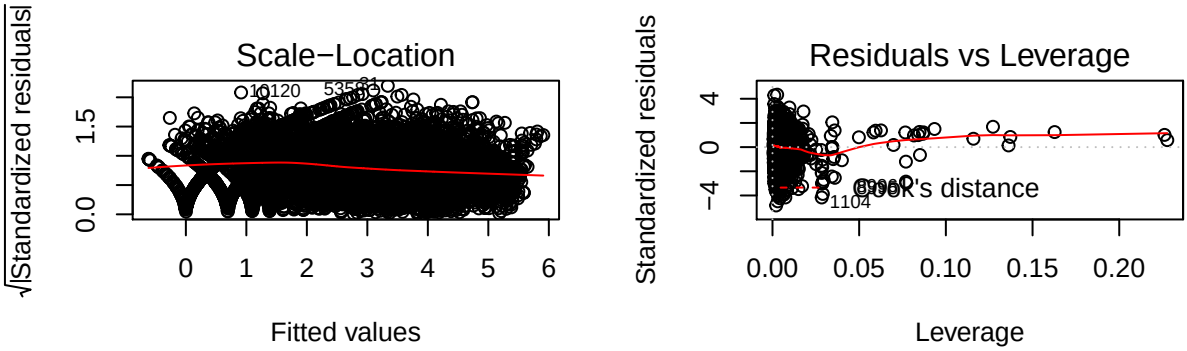
```
## [1] -0.08116805
```

As shown in the comparison above, after I decrease the number of parameters through step function, the RSS of the prediction doesn't change much comparing to the combined but longer model that I previously tested. Therefore, I will use the shorter version of the regression equation.

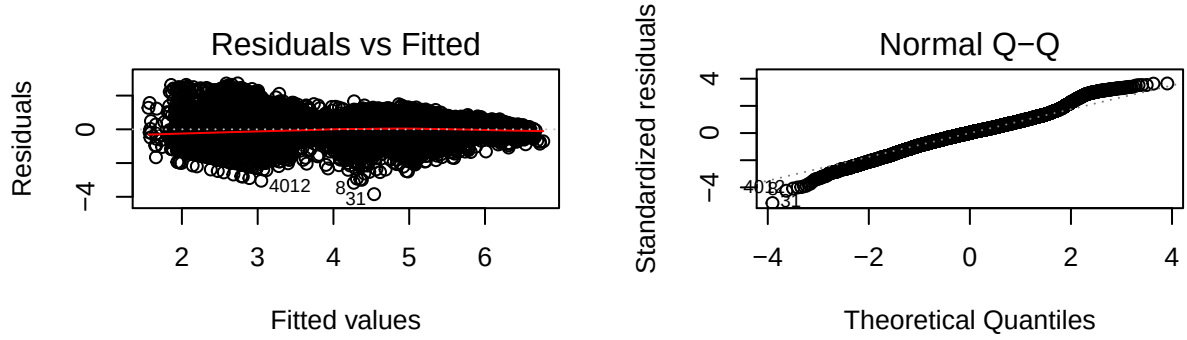
6. Regression Diagnostics

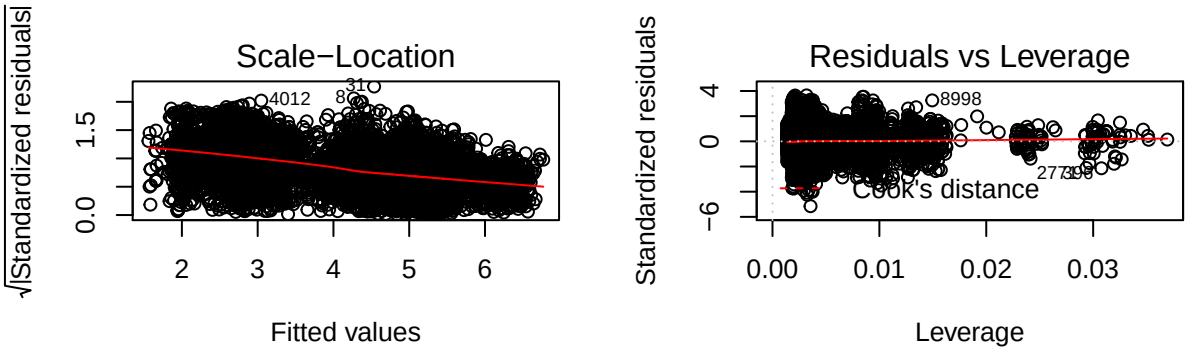
```
par(mfrow = c(2,2))
plot(casual.lm.short)
```






```
par(mfrow = c(2,2))
plot(registered.lm.short)
```





From the regression diagnostic plots shown above. We could see that generally, the assumptions are satisfied.

- For both casual.lm.short and registered.lm.short, the linearity assumptions holds true because the Residual vs Fitted plot shows a horizontal line, which means that the fitted values does not have a linear realtionship with residuals.
- From the Normal QQ plots of casual.lm.short and registered.lm.short, we see that the distribution of residuals are generally normal with slightly heavier tail at the ends. But since we have a very large sample size, the CLT applies and we could assume the errors follow a normal distribution in general.
- The homoscedasticity assumption is slightly violated since for both casual.lm.short and registered.lm.short, the variance of residual decreases as fitted values increases. This might slightly affect the precision of p-values of our variables and the confidence intervals of our prediction.

7. Predictions

As I mentioned above, I combined weather4 into weather3 in my training dataset. To make sure I correctly predict the test data, I am going to combine weather4 into weather3 in my test.csv correspondingly.

```
modify.func = function(){
  for (i in c(1: nrow(test))) {
    if ( test$weather[i] == 4) {
      return(i)
    }
  }
}

modify.func()
```

```
## [1] 155
```

```
test$weather[155] = 3
modify.func()
```

```
## [1] 3249
```

```
test$weather[3249] = 3
sum(test$weather == 4)
```

```
## [1] 0
```

Now I make sure that all weather4 has been included as weather3.

```
predict.log.casual =  predict(casual.lm.short, newdata = test)
predict.log.registered =  predict(registered.lm.short, newdata = test)
predict.norm.casual = expm1(predict.log.casual)
predict.norm.registered = expm1(predict.log.registered)
predict.norm.count = predict.norm.casual + predict.norm.registered


predict.test.1 = data.frame(datetime = test$datetime, count = predict.norm.count)
write.csv(predict.test.1, file = "prediction submission.csv", row.names = F)
```

8.Conclusion & Reflection

In this project, some very intersting findings contributed a lot to my data analysis, they are:

- realizing time of rental is one of the most important feature and segment time into different categories based on the total rental counts per hour

- conducting visualizations on casual and registered bikers separately and discover the differences in how these two groups react to various factors. Fit two separate models for these two groups and largely improve the precision of prediction
- trying out both numerical&categorical and categorical&categorical interactions, realizing that casual and registered biker models behave differently under two types of interactions and combine these two models to increase prediction accuracy
- randomly sample out data in training set to test my model and evaluate model performances based on testing results

I also have some reflections after this project:

Although the project outline specifies to do regression diagnostics at the end. I do think we should do this after EDA and before starting to fit linear regression models. It is true that for normality and homoscaticity assumptions, central limit theorem (CLT) would make our error roughly normally distributed and error distribution is only a part of our assumption. But the linearity assumption is very important and we need to confirm that it is a linear and not other types of relationship before we start to fit our linear regression models.