Capital Bikeshare Dataset Analysis

Michael Alpas, Mohit Singh and Upendra Yadav

August 07, 2020

Introduction

Overview

For this project, we would like to simulate that Team Outlier is a team of Data Scientists that Capital Bikeshare company hired to come up with data driven answers to help them with their decisions.

The following business questions will be guided by the two-year historical log corresponding to years 2011 and 2012.

Use Case 1 - Operational Expenses Analysis

The company need to reduce operational expenses by looking at the manpower to cater extra demand. Hence Team Outlier will predict the total ridership count based on different variables. In order to predict exact operational expenses this model will try to answer below questions -

- Does Season play a significant role in Bike ridership count?
- Does daily environment factors such as weather, temperature or humidity affect the bike ridership count so that Capital Bike share compnay can adjust his human capital?
- Does holidays, weekdays, or weekends play an important role in predicting the ridership count?

Use Case 2 - Targeted Marketing Strategy Analysis

Capital Bikeshare would like to have a targeted marketing strategy to help them efficiently spend their budget. The Team Outlier needs to determine the recommended season to have marketing promotional offers to increase the number of customers.

Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For instance, weather conditions, precipitation, day of week, season, hour of the day, etc. can affect the rental behaviors. The core data set is related to the two-year historical log corresponding to years 2011 and 2012 from Capital Bikeshare system, Washington D.C., USA which is publicly available. Our source (UCI Machine Learning Repository) aggregated the data on two hourly and daily basis. They extracted and added the corresponding weather and seasonal information. Weather information were extracted from http://www.freemeteo.com.

There are two datasets – `hour.csv` and `day.csv`. Both datasets have the same fields except `hr` which is not available in `day.csv`. The `hour.csv` contains bike sharing counts aggregated on hourly basis and it

has records of 17379 hours. The `day.csv` contains bike sharing counts aggregated on daily basis and it has records of 731 days.

Methods

Use Case 1 - Operational Expenses Analysis

Loading the Dataset

```
library(readr)
day_data = read.csv("dataset/day.csv")
#head(day_data)
hour_data = read.csv("dataset/hour.csv")
#head(hour_data)
```

Data Cleaning

- Change Numeric Variables to Factor Variables

```
# Convert Season to Factor
day_data$season = as.factor(day_data$season)
# Set Seasons
levels(day_data$season) = c("Spring", "Summer", "Fall", "Winter")
# Convert Holiday to Factor
day_data$holiday = as.factor(day_data$holiday)
# Set Holiday
levels(day_data$holiday) = c("No", "Yes")
#Convert WorkingDay to Factor
day_data$workingday = as.factor(day_data$workingday)
levels(day_data$workingday) = c("No", "Yes")
#Convert Weather to Factor
day_data$weathersit = as.factor(day_data$weathersit)
levels(day_data$weathersit) = c("Clear", "Mist", "LightPrecip")
# Convert Week days toFactor
day_data$weekday = as.factor(day_data$weekday)
levels(day_data$weekday) = c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
# Convert Month toFactor
day_data$mnth = as.factor(day_data$mnth)
levels(day_data$mnth) = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                          "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
# Check for Missing Field
day_data = na.omit(day_data)
hour_data = na.omit(hour_data)
```

Data Exploration

Before going in to the modelling we will explore the data set to uncover initial patterns, characteristics, and points of interest. This exploratory analysis will look at the distributions of individual predictors, relationships between predictors and the response, correlation and interaction between predictors as related to response.

This exploratory analysis is included in detailed manner in Appendix section.

Model the Data

We will remove few variables such as Instance and date which are unique to each row, these variables dont contribute much to total ridership.

```
data = day_data[, c(-1, -2, -14, -15)]  # removing instance, date, causal and registered variable (thes
```

Define Functions

Function to calculate LOOCV-RMSE

```
calc_loocv_rmse = function(model) {
  temp = (resid(model) / (1 - hatvalues(model))) ^ 2
  temp = temp[is.finite(temp)]
  sqrt(mean(temp))
}
```

Separate Numeric and Categorical Variables.

```
numerical = unlist(lapply(data, is.numeric)) # contains boolean value whether a varible is having a num
```

Modelling with Numeric Predictors Only

Lets begin by creating a model using only numerical predictors

```
data_numerical = data[, numerical] # get all the numerical columns
bike_mod_num = lm(cnt ~ ., data = data_numerical) # model with all numeric variables
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_num)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.7261
```

```
calc_loocv_rmse(bike_mod_num) # get the loocv rmse
```

```
## [1] 1042
```

This model has a very high cross validated RMSE and low Adjusted R^2. Hence we can conclude that model is not explaining the response very well.

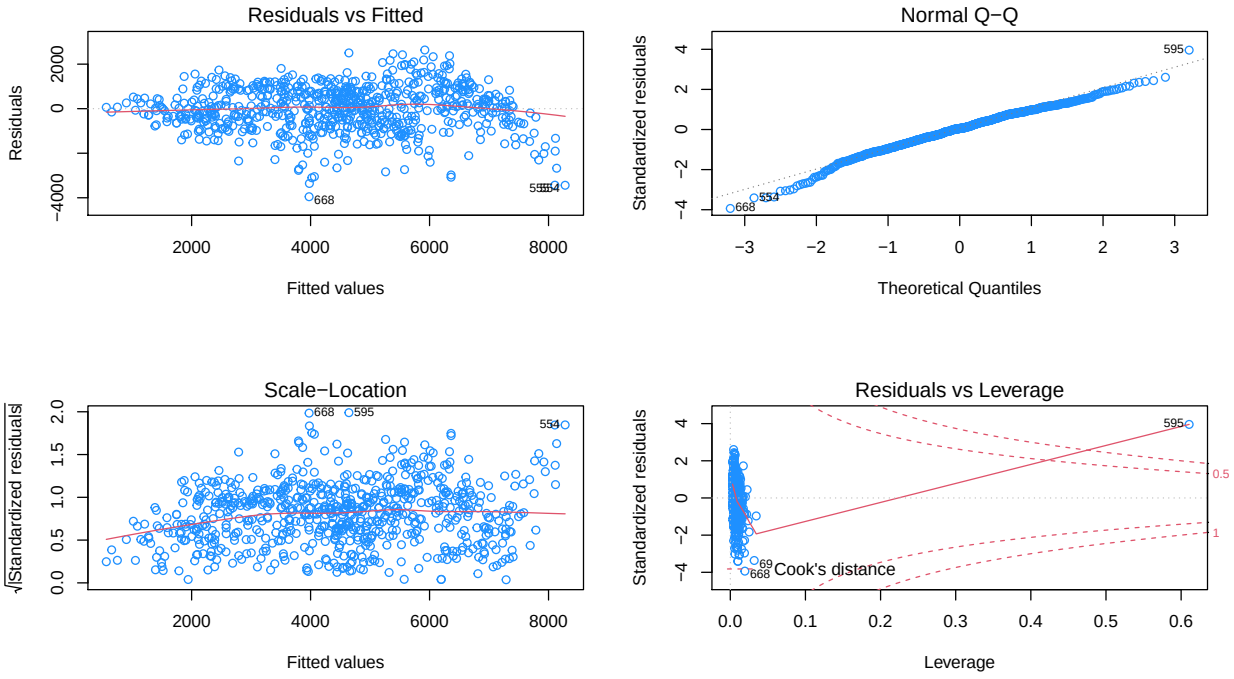
Let's now examine the p-values for the coefficients of the model:

```
summary(bike_mod_num)$coefficients[, 'Pr(>|t|)']
```

```
## (Intercept)          yr        temp       atemp         hum   windspeed
##   7.548e-19  6.263e-109   2.938e-01   4.506e-03   1.193e-15   7.786e-15
```

The above summary results show that the temp is not very significant predictor as it has a high p-value, this may be because of collinearity between `temp` and `atemp` varible.

```
par(mfrow = c(2, 2))
plot(bike_mod_num,col = 'dodgerblue') # create diagnostics model
```



The Fitted vs Residual plot does not show constant variance hence violating the assumption of linear regression model. The leverage plot also highlights some outliers.

Modelling with Numeric and Categorical Predictors

```
bike_mod_all = lm(cnt ~ ., data = data) # modelling with all the variables
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_all)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.8423
```

```
calc_loocv_rmse(bike_mod_all) # get the loocv rmse
```

```
## [1] 794.8
```

By including the categorical variables, Adjusted R^2 has substantially improved and it has also lowered the loocv-rmse.

P-values for coefficients for the model:

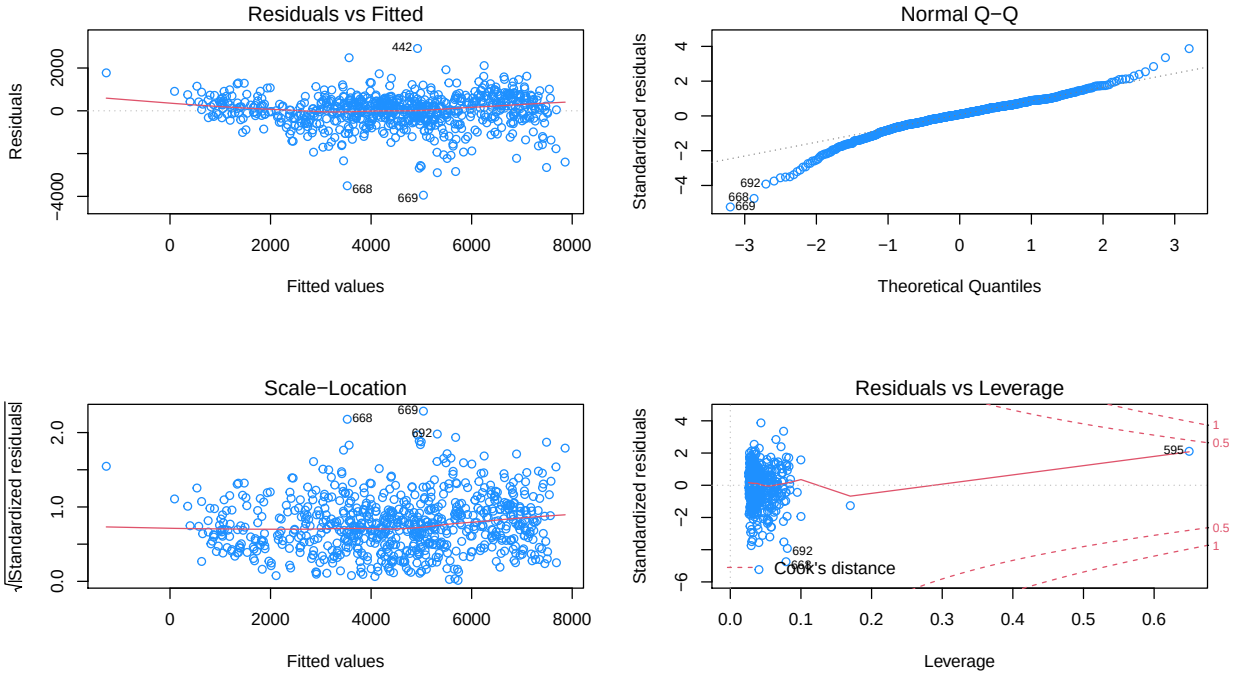
```
summary(bike_mod_all)$coefficients[, 'Pr(>|t|)'] # get the cofficeints
```

```
##            (Intercept)          seasonSummer            seasonFall
##              9.775e-10             1.032e-06             1.025e-04
##          seasonWinter                    yr               mnthFeb
##              2.280e-17            2.295e-154             3.624e-01
##               mnthMar               mnthApr               mnthMay
##              1.085e-03             6.882e-02             6.145e-03
##               mnthJun               mnthJul               mnthAug
##              6.842e-02             9.219e-01             1.426e-01
##               mnthSep               mnthOct               mnthNov
##              1.652e-04             3.179e-02             6.133e-01
##               mnthDec            holidayYes            weekdayMon
##              6.231e-01             1.130e-03             5.319e-02
##            weekdayTue            weekdayWed            weekdayThu
##              3.982e-03             4.136e-04             3.498e-04
##            weekdayFri            weekdaySat       weathersitMist
##              5.296e-05             4.007e-05             3.156e-09
## weathersitLightPrecip                  temp                 atemp
##              5.386e-22             4.153e-02             2.223e-01
##                    hum             windspeed
##              2.014e-07             2.092e-11
```

The p-values of all the parameters of the additive model indicate that not all of the variables are significant.

Diagnostic plots for the model:

```
par(mfrow = c(2, 2))
plot(bike_mod_all,col = 'dodgerblue') # create diagnostics model
```



By including categorical variables also in the model, we still see some issues in the diagnostic plots.

- The Fitted vs Residual plot shows a non linear trend also does not show constant variance, hence violating the assumption of linear regression model. We also see presence of some extreme outlier in the plot.
- We see some fat tails in the Normal Q-Q plot.
- The Residuals vs Leverage plot also indicates presence of some outliers which we might have to check on as we go down the analysis.

Based upon the above results, we will examine below the significance of several of the categorical variables in the response variable:

- Month
- Week Day
- Working Day

```
bike_mod_w_month = lm(cnt ~ . - mnth, data = data) # model without month
bike_mod_w_weekday = lm(cnt ~ . - weekday, data = data) # model without weekday
bike_mod_w_workingday = lm(cnt ~ . - workingday, data = data) # model without workingday
#  anova test to compare above three models
anova(bike_mod_w_month,
      bike_mod_w_weekday,
      bike_mod_w_workingday,
      bike_mod_all)
```

```
## Analysis of Variance Table
##
```

```
## Model 1: cnt ~ (season + yr + mnth + holiday + weekday + workingday +
##     weathersit + temp + atemp + hum + windspeed) - mnth
## Model 2: cnt ~ (season + yr + mnth + holiday + weekday + workingday +
##     weathersit + temp + atemp + hum + windspeed) - weekday
## Model 3: cnt ~ (season + yr + mnth + holiday + weekday + workingday +
##     weathersit + temp + atemp + hum + windspeed) - workingday
## Model 4: cnt ~ season + yr + mnth + holiday + weekday + workingday + weathersit +
##     temp + atemp + hum + windspeed
##   Res.Df       RSS Df Sum of Sq     F  Pr(>F)
## 1    713 472452877
## 2    707 428442679  6  44010198 12.40 2.7e-13 ***
## 3    702 415401688  5  13040991  4.41 0.00059 ***
## 4    702 415401688  0         0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova test shows that `month` and `weekday` are significant predictors, hence we cannot rule them out. Even though the `month` variable is statistically significant, it might be that just few levels are useful and rest of them does not help. We will use model selection schemes later to investigate that.

According to test results working day variable seems to be non significant and we can rule out this variable.

We will now re-fit the model excluding working day:

```
data_2 = data[, c(-6)] # remove working day variable
bike_mod_all_2 = lm(cnt ~ ., data = data_2) # model with all remaining variable
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_all_2)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.8423
```

```
calc_loocv_rmse(bike_mod_all_2) # get the loocv rmse
```

```
## [1] 794.8
```

Excluding working day had no effect on the R^2 of the model, so we can safely remove this variable.

Identifying Collinearity in Model

Now we will using the Variance Inflation Factor to determine if we have multi-collinearity issues in the model:

```
library(faraway)
vif(bike_mod_all_2)
```

```
##          seasonSummer            seasonFall          seasonWinter
##                 7.496                10.720                 7.455
##                    yr              mnthFeb               mnthMar
##                 1.047                 1.836                 2.624
##               mnthApr               mnthMay               mnthJun
##                 5.704                 6.868                 7.423
##               mnthJul               mnthAug               mnthSep
##                 9.444                 8.813                 6.542
```

```
##               mnthOct               mnthNov               mnthDec
##                 5.595                 4.957                 3.184
##           holidayYes            weekdayMon            weekdayTue
##                 1.121                 1.822                 1.730
##           weekdayWed            weekdayThu            weekdayFri
##                 1.741                 1.743                 1.740
##           weekdaySat       weathersitMist weathersitLightPrecip
##                 1.726                 1.642                 1.338
##                temp                 atemp                   hum
##               80.807                70.037                 2.140
##            windspeed
##                 1.273
```

By looking the results of VIF function above, (as we had already suspected by looking to p-value of different variables), temp and atemp have a high level of collinearity. We will now look at the partial correlation coefficient between `temp` and `cnt`:

```
temp_model_1 = lm(temp ~ . - cnt, data = data_2)
temp_model_2 = lm(cnt ~ . - temp, data = data_2)
cor(resid(temp_model_1), resid(temp_model_2))
```

```
## [1] 0.07684
```

While this is relatively small, as `temp` and `atemp` are highly correlated we should check the partial correlation coefficient after removing atemp:

```
temp_model_1 = lm(temp ~ . - cnt - atemp, data = data_2)
temp_model_2 = lm(cnt ~ . - temp - atemp, data = data_2)
cor(resid(temp_model_1), resid(temp_model_2))
```

```
## [1] 0.3801
```

Let us observe the partial correlation coefficient between `atemp` and `cnt`, removing `temp`:

```
temp_model_1 = lm(atemp ~ . - cnt - temp, data = data_2)
temp_model_2 = lm(cnt ~ . - atemp - temp, data = data_2)
cor(resid(temp_model_1), resid(temp_model_2))
```

```
## [1] 0.3758
```

These results indicate that `temp` is more correlated with `cnt` than `atemp` is, so we will remove `atemp` and leave `temp`:

```
data_3 = data_2[, -8]
bike_mod_all_3 = lm(cnt ~ ., data = data_3)
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_all_3)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.8422
```

```
calc_loocv_rmse(bike_mod_all_3) # get the loocv rmse
```

```
## [1] 789.3
```

This change has slightly lowered the adjusted R^2 as we would expect removing a predictor would, but has improved the LOOCV-RMSE, which indicates that it has improved the model.

As we have concerns about the year predictor, we will also look at the partial correlation coefficient between year and count:

```
yr_mod_0 = lm(cnt ~ . - yr, data = data_3)
yr_mod_1 = lm(yr ~ . - cnt, data = data_3)
cor(resid(yr_mod_0), resid(yr_mod_1))
```

```
## [1] 0.7943
```

Partial correlation coefficinet is quite high which indicates that the year has a significant relationship with ridership. We have seen that the ridership seems to be increasing from year to year (with some seasonal cycles within that trend.) Since our data only includes two years this may cause problems with using the model to extrapolate to years that are not included in the training set. However, the high partial correlation coefficient indicates that year is an important predictor so we will keep this it in the model.

Outlier Diagnostics in the Model

Next we would like to check for potential outliers, we have 3 different strategy of doing so :

- Leverage
- Standard Residual
- Cooks Distance

We will be using Cooks Distance to identify any such outlier and see the effect of it on the model.

First, we will calculate the number of observations flagged by Cooks Distance:

```
sum(cooks.distance(bike_mod_all_3) > 4 / length(cooks.distance(bike_mod_all_3)))
```

```
## [1] 44
```

Next step it to refit a model excluding the identified observations:

```
cokks_distance = cooks.distance(bike_mod_all_3)
bike_mod_all_4 = lm(cnt ~ .,
                    data = data_3,
                    subset = cokks_distance <= 4 / length(cokks_distance))
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_all_4)[["adj.r.squared"]] # get the adjusted r-squared
```

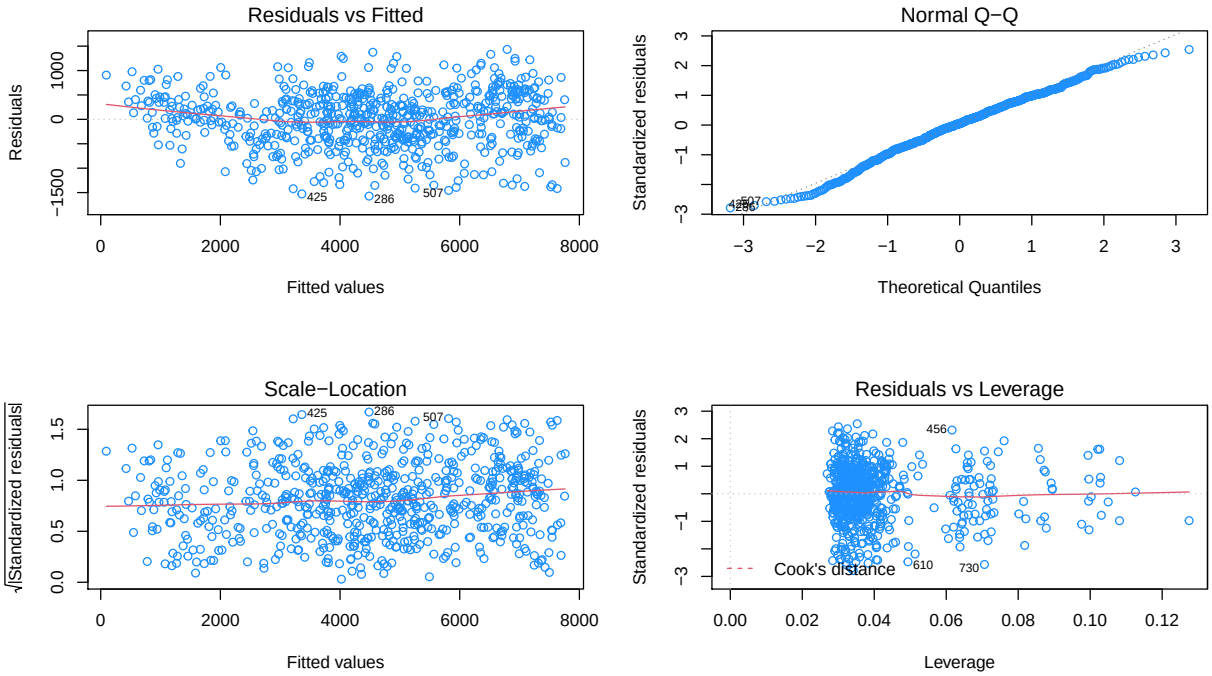
```
## [1] 0.9086
```

```
calc_loocv_rmse(bike_mod_all_4) # get the loocv rmse
```

```
## [1] 586.6
```

Removing these outliers resulted in a substantial increase in R^2 and a substantial decrease in LOOCV-RMSE.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_4,col = 'dodgerblue') # Create diagnostics
```



By looking the Residuals vs Fitted plot, we observed that the distribution of variance looks quite constant, although we do still see some non-linear patterns which indicate that we may want to try some higher order terms or transformations.

The Residuals vs Leverage plot also looks much neater and the Normal Q-Q plot has also improved.

Impact of Interactions in the Model

We will now evaluate including interactions:

```
bike_mod_all_5 = lm(cnt ~ . ^ 2,
                    data = data_3,
                    subset = cokks_distance <= 4 / length(cokks_distance))
# Get Model Parameters - adjusted r-squared and loocv rmse
summary(bike_mod_all_5)[["adj.r.squared"]]
```

```
## [1] 0.9467
```

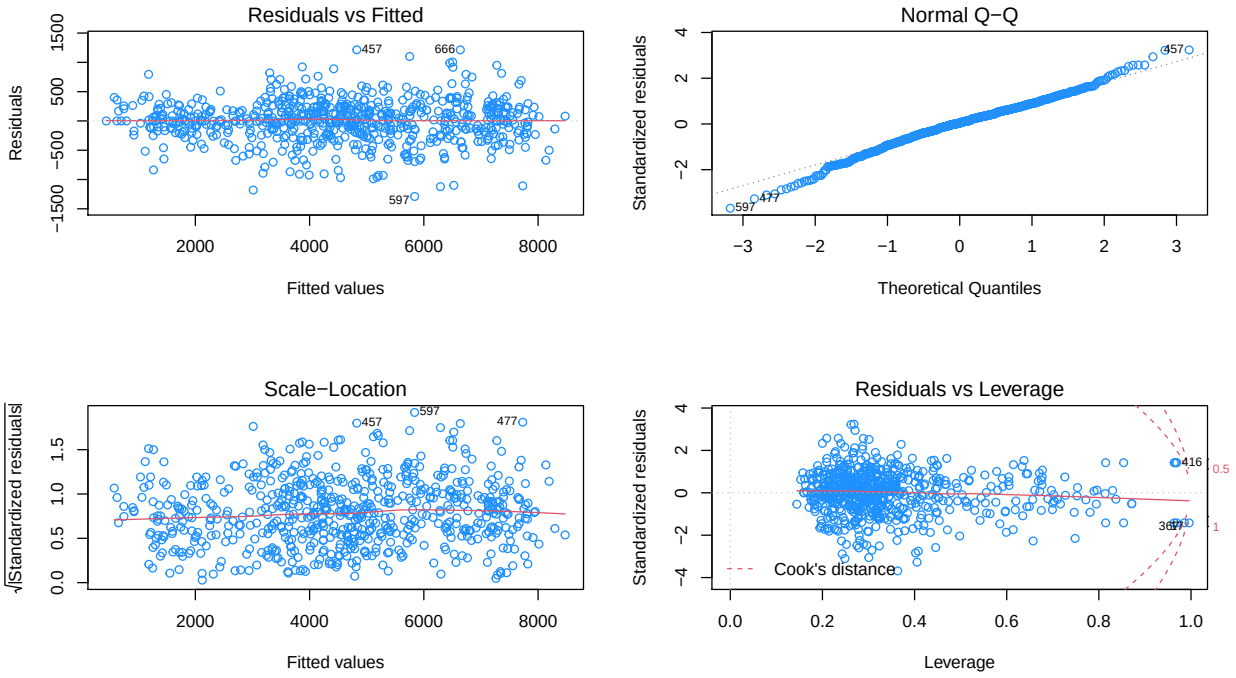
Including all possible interactions resulted in a substantial improvement in adjusted R^2. However, we can not be sure if this improvement is due to the model or merely due to the inclusion of additional predictors.

```
calc_loocv_rmse(bike_mod_all_5) # get the loocv rmse
```

```
## [1] 759.6
```

The LOOCV-RMSE has increased, which indicates that the additional terms could have resulted in over fitting the data.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_5,col = 'dodgerblue') # do diagnostics
```




```
length(coef(bike_mod_all_5)) # get the number of params
```

```
## [1] 305
```

The residual vs fitted plot looks random with errors randomly distributed around 0 line, there are still some outliers which are getting highlighted on the plot.

The Q-Q plot looks more normal.

The leverage plot shows some indication of potential new outliers.

Model Selection

Since the model including all possible interactions increased the LOOCV-RMSE, indicating that it was over fitting to the training data, we will perform a backwards AIC step search to remove the non-significant terms.

```
bike_mod_all_6 = step(bike_mod_all_5, trace = 0, direction = "backward")
length(coef(bike_mod_all_6)) # get the no of params
```

```
## [1] 117
```

```
summary(bike_mod_all_6)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.9482
```

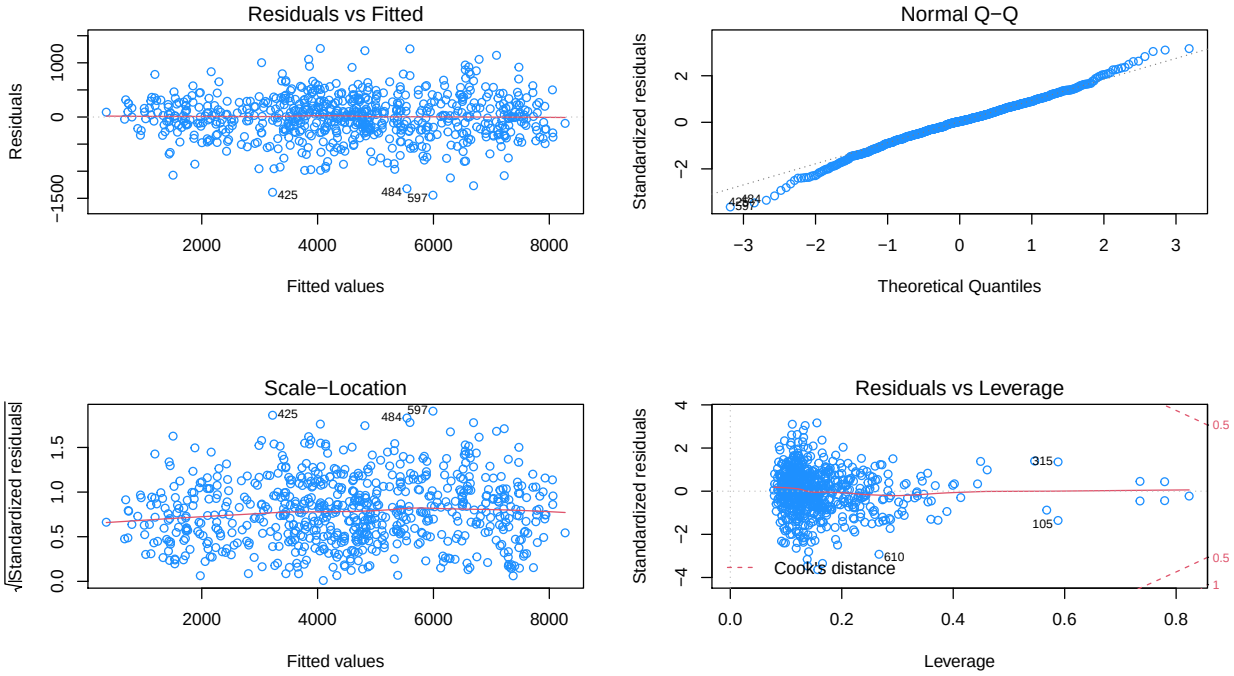
While decreasing the number of predictors from 305 to 117, the backwards step search also resulted in a very small improvement in adjusted R^2.

```
calc_loocv_rmse(bike_mod_all_6) # get the loocv rmse
```

```
## [1] 470.9
```

The LOOCV-RMSE also significantly improved, which indicates that this smaller model is a much better model for inference.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_6,col = 'dodgerblue') # do diagnostics
```



The Residuals vs Fitted plot still looks normal.

Polynomial Fitting

We have previously seen some issues which indicated that polynomial features might improve the model. We will now evaluate including them:

```
temp_mod = lm(
  cnt ~ . + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^ 2),
  data = data_3,
  subset = cokks_distance <= 4 / length(cokks_distance)
)
bike_mod_all_7 = step(temp_mod, trace = 0, direction = "backward")
summary(bike_mod_all_7)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.9203
```

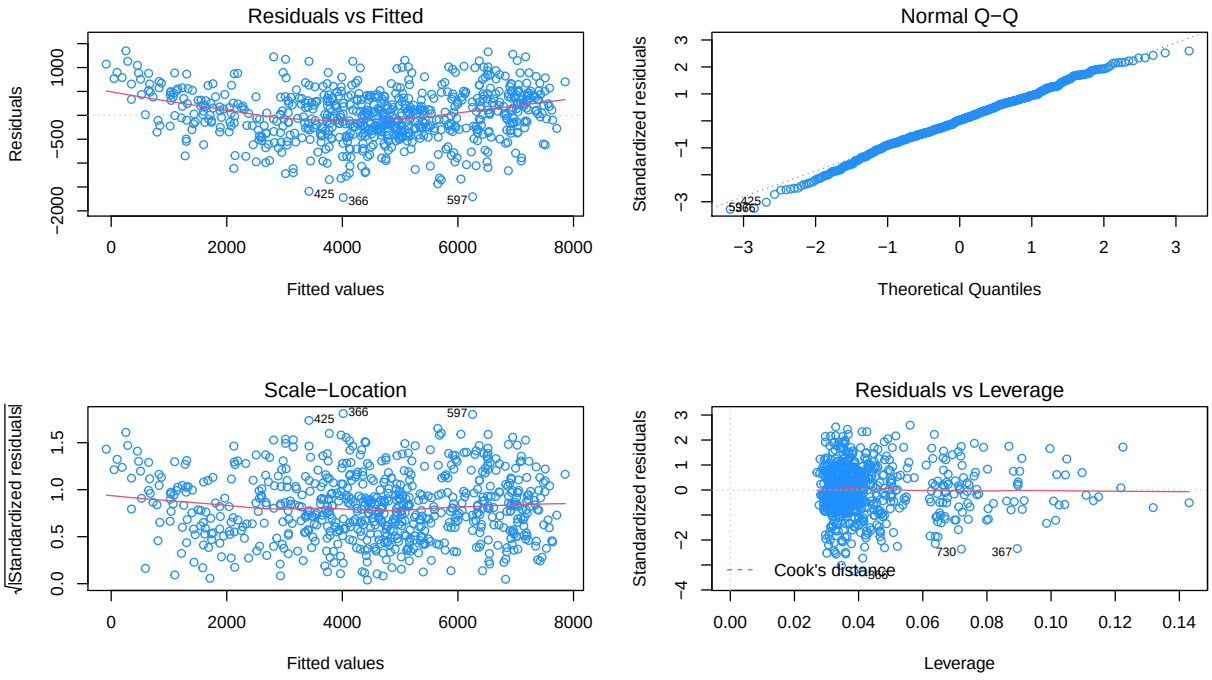
The adjusted R^2 is lower than it was for our interaction model.

```
calc_loocv_rmse(bike_mod_all_7) # get the loocv rmse
```

```
## [1] 548.5
```

The LOOCV-RMSE has also increased.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_7,col = 'dodgerblue') # do diagnostics
```



The Residual vs Fitted plot does not look random and shows some non-linear pattern, which indicate that the inclusion of these terms has not improved the model. However, $temp^2$ has a very low p-value which indicates that it may be useful. We will keep this in mind.

Transformations

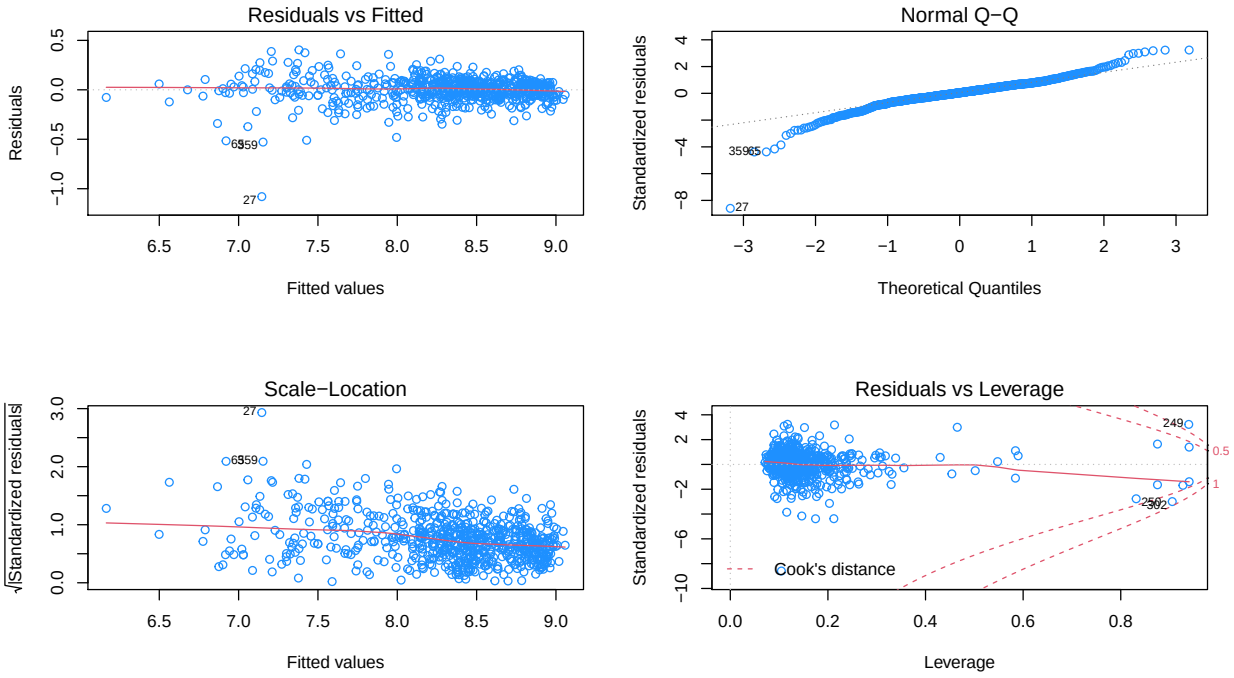
Let us evaluate taking a log transformation of the response.

```
temp_m = lm(log(cnt) ~.^2,
                      data = data_3,
                      subset = cokks_distance <= 4 / length(cokks_distance))
bike_mod_all_8=step(temp_m, trace=0, direction="backward")
summary(bike_mod_all_8)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.9379
```

The adjusted R^2 has been lowered by this transformation.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_8,col = 'dodgerblue') # do diagnostics
```



In addition, we observed issues in both the Residuals vs Fitted plot and the Normal Q-Q plot. We can conclude that this transformation was not helpful.

Interactions with Polynomial Terms

Finally, since some of the polynomial terms seemed as if they could be significant, we will try a model which includes interactions and polynomial terms:

```
bike_mod_int_poly = lm(
  cnt ~ (. ^ 2) + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^ 2),
  data = data_3,
  subset = cokks_distance <= 4 / length(cokks_distance)
```

```
)
bike_mod_all_10 = step(bike_mod_int_poly, trace = 0, direction = "backward")
summary(bike_mod_all_10)[["adj.r.squared"]] # get the adjusted r-squared
```

```
## [1] 0.9498
```

The adjusted R^2 is the best we have found yet.

```
calc_loocv_rmse(bike_mod_all_10) # get the loocv rmse
```

```
## [1] 461.4
```

And this model also results in the lowest LOOCV-RMSE.

Finally we will refit the model using the full data set, find the observations with high leverage and refit the model excluding those items:

```
bike_mod_int_poly_full = lm(cnt ~ (. ^ 2) + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^
                                                                           2),
                            data = data_3)
# Check AIC to fit the best model
bike_mod_all_11 = step(bike_mod_int_poly_full,
                       trace = 0,
                       direction = "backward")

# Use Cook Distance to remove outliers
cooks_distance = cooks.distance(bike_mod_all_11) # find influential observations and exclude them
filter = cooks_distance <= (4 / length(cooks_distance))

# some points have a cooks distance of NA, we will consider these to be outliers
filter[is.na(filter)] <- FALSE
bike_mod_int_poly_full = lm(cnt ~ (. ^ 2) + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^
                                                                           2),
                            data = data_3,
                            subset = filter)
bike_mod_all_11 = step(bike_mod_int_poly_full,
                       trace = 0,
                       direction = "backward")
summary(bike_mod_all_11)[["adj.r.squared"]] # get the adjusted r-squared using AIC
```

```
## [1] 0.9582
```

```
calc_loocv_rmse(bike_mod_all_11) # get the loocv rmse using AIC
```

```
## [1] 426.8
```

```
## Check Same Calculation with BIC

bike_mod_int_poly_bic = lm(cnt ~ (. ^ 2) + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^
                                                                           2),
```

```
                              data = data_3)
# Applying BIC
n = length(coef(bike_mod_int_poly_bic))
bike_mod_all_bic = step(bike_mod_int_poly_bic,
                        trace = 0,
                        direction = "backward", k = log(n))
length(coef(bike_mod_all_11))  # Total no of coef in aic model
```

```
## [1] 151
```

```
length(coef(bike_mod_all_bic)) # Total no of coef in bic model
```

```
## [1] 92
```

```
# Remove Outliers from BIC Model
cooks_distance_1 = cooks.distance(bike_mod_all_bic) # find influential observations and exclude them
filter1 = cooks_distance_1 <= (4 / length(cooks_distance_1))
filter1[is.na(filter)] <- FALSE


bike_mod_int_poly_full_bic = lm(cnt ~ (. ^ 2) + I(temp ^ 2) + I(hum ^ 2) + I(windspeed ^
                                                                              2),
                            data = data_3,
                            subset = filter1)
bike_mod_all_bic_fix = step(bike_mod_int_poly_full,
                        trace = 0,
                        direction = "backward", k = log(n))


# get the adjusted r-squared and
summary(bike_mod_all_bic)[["adj.r.squared"]] # get the adjusted r-squared using BIC without removing Ou
```

```
## [1] 0.9095
```

```
calc_loocv_rmse(bike_mod_all_bic) # get the loocv rmse using BIC without removing Outliers
```

```
## [1] 612.2
```

```
summary(bike_mod_all_bic_fix)[["adj.r.squared"]] # get the adjusted r-squared using BIC after removing
```

```
## [1] 0.946
```

```
calc_loocv_rmse(bike_mod_all_bic_fix) # get the loocv rmse using BIC after removing Outliers
```

```
## [1] 459.9
```

Comment -Finding and removing the observations with high leverage has improved the LOOCV-RMSE and the R^2.

-Here, we see that by using BIC our predictors are reduced from 151 in AIC to 92. so as per the interpretation of the model, BIC will be the good fit. But when we see adjusted R2 and also LOOCV-RMSE, we see that adjusted R2 is higher and LOOCV-RMSE is lower when we select AIC.

-Since, we are fitting this model for prediction, so I will go for the model which gives lower LOOCV-RMSE and higher adj R2. Hence, we will choose model fit via AIC.

Use Case 2 - Targeted Marketing Strategy Analysis

```
library(readr)
day_data = read.csv("dataset/day.csv")
# knitr::kable(head(day_data)[, 1:15])
#head(day_data)
hour_data = read.csv("dataset/hour.csv")
# knitr::kable(head(hour_data)[, 1:15])
#head(hour_data)
```

Filter the data based on season

```
# spring filtered data
spring_data = subset(day_data, day_data$season == 1)
# summer filtered data
summer_data = subset(day_data, day_data$season == 2)
# fall filtered data
fall_data   = subset(day_data, day_data$season == 3)
# winter filtered data
winter_data = subset(day_data, day_data$season == 4)
```

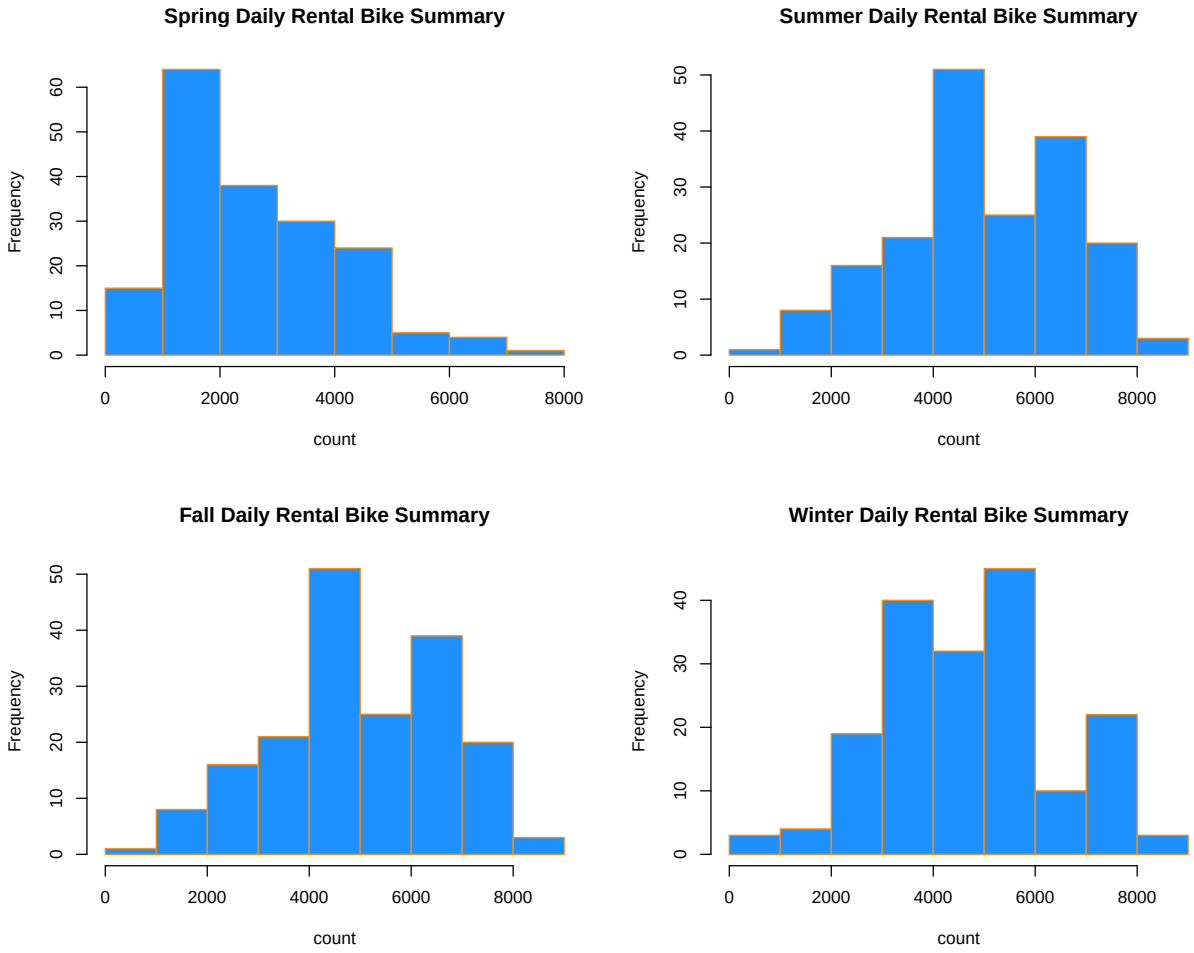
Remove Possible NA

```
# remove NA
spring_data = na.omit(spring_data)
summer_data = na.omit(summer_data)
fall_data = na.omit(fall_data)
winter_data = na.omit(winter_data)
```

Quick Comparison

Let us take a quick look with the filtered data.

```
# plot to see the difference
par(mfrow=c(2,2))
hist(spring_data$cnt, col = "dodgerblue",border = "darkorange", xlab = "count", main = "Spring Daily Re
hist(summer_data$cnt, col = "dodgerblue",border = "darkorange", xlab = "count", main = "Summer Daily Re
hist(summer_data$cnt, col = "dodgerblue",border = "darkorange", xlab = "count", main = "Fall Daily Renta
hist(winter_data$cnt, col = "dodgerblue",border = "darkorange", xlab = "count", main = "Winter Daily Re
```



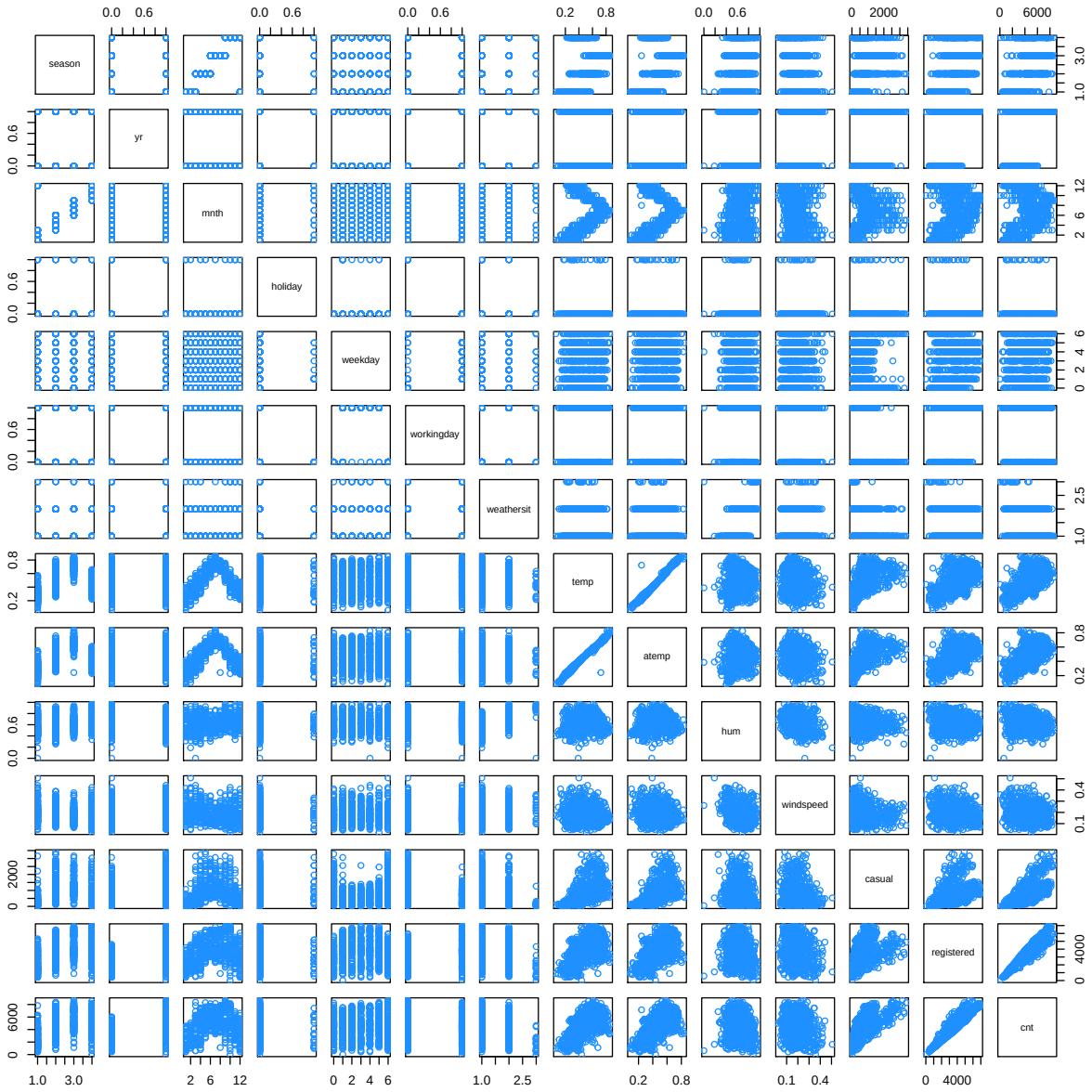
Collinearity Check

We will remove the `instant` and `dteday` variables since these are data reference index and so we can use the `cor()` and `pair()` functions to see what are the predictors that are highly correlated.

```
day_data_converted = day_data[3:16]
# convert all factor variables to numeric in order to call cor()
for(name in colnames(day_data_converted)) {
  if (is.integer(day_data_converted[[name]]))
    day_data_converted[[name]] = as.numeric(day_data_converted[[name]])
}
```

Let us visually check the correlation between the predictors. We would see immediately that there are 2 interesting sets of variables. 1) `temp` and `atemp` 2) `registered` and `cnt`.

```
pairs(day_data_converted, col = "dodgerblue")
```



Now, let us check using `cor()` function. We will use cor values > 0.5 to see the predictors that are correlated

```
all_cor = round(cor(day_data_converted), 2)
(correlated_predictors = sort(abs(all_cor["cnt", abs(all_cor["cnt", ]) > 0.5]), decreasing = TRUE)[-1])
```

```
## registered     casual       temp      atemp         yr
##       0.95       0.67       0.63       0.63       0.57
```

Based on the results of `cor()` computation, the above predictors will not be helpful to predict the recommended `season` to run promotional offers to reach company's goal of increasing the number of customers.

```
#Model after removing collinear variables
full_car_model = lm(cnt ~ season + mnth + holiday + weekday + workingday + weathersit + hum + windspeed
```

Remove influential points

Now we will remove the influential points if any in the data.

```
#Model without influential points
without_inf_data = subset(day_data_converted, subset= cooks.distance(full_car_model) <= 4/length(cooks.
full_car_model = lm(cnt ~ season + mnth + holiday + weekday + workingday + weathersit + hum + windspeed
```

Predictor Selection

We will explore for the best model by using backward search to find that.

```
#AIC Process
best_bod_back_aic = step(full_car_model, direction = "backward", trace = 0)
```

```
#BIC process
best_bod_back_bic = step(full_car_model, direction = "backward", k = log(length(resid(full_car_model)))
```

Comparing models selected by AIC and BIC methods

```
#AIC
extractAIC(best_bod_back_aic)
```

```
## [1]     7 10282
```

```
extractAIC(best_bod_back_bic, k = log(length(resid(full_car_model))))
```

```
## [1]     6 10312
```

By Seeing the AIC value we are going to select the model searched by backward search AIC method.

Test with anova to check the significane of selected model

```
anova(best_bod_back_aic, full_car_model)
```

```
## Analysis of Variance Table
##
## Model 1: cnt ~ season + mnth + holiday + weekday + weathersit + windspeed
## Model 2: cnt ~ season + mnth + holiday + weekday + workingday + weathersit +
##     hum + windspeed
##   Res.Df      RSS Df Sum of Sq    F Pr(>F)
## 1    693 1.64e+09
## 2    691 1.64e+09  2   7779652 1.64   0.19
```

By Seeing the p-value we failed to reject the null hypothesis so we select the model selected by AIC.

```
best_model = lm(cnt ~ season + mnth + holiday + weekday + workingday + weathersit + hum + windspeed, da
```

Compute Prediction Each Season

For computing the mean prediction, we will use the predictors that we identified from the previous process.

```r
library(knitr)
library(kableExtra)
spring_pred = mean(predict(best_model, newdata = spring_data, interval = c("prediction"), level = 0.99))
summer_pred = mean(predict(best_model, newdata = summer_data, interval = c("prediction"), level = 0.99))
fall_pred = mean(predict(best_model, newdata = fall_data, interval = c("prediction"), level = 0.99))
winter_pred = mean(predict(best_model, newdata = winter_data, interval = c("prediction"), level = 0.99))
# create table for prediction results
pred_results = data.frame(
  "prediction" = c(
    "Spring" = spring_pred,
    "Summer" = summer_pred,
    "Fall"   = fall_pred,
    "Winter" = winter_pred
  )
)
kable(pred_results) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
```

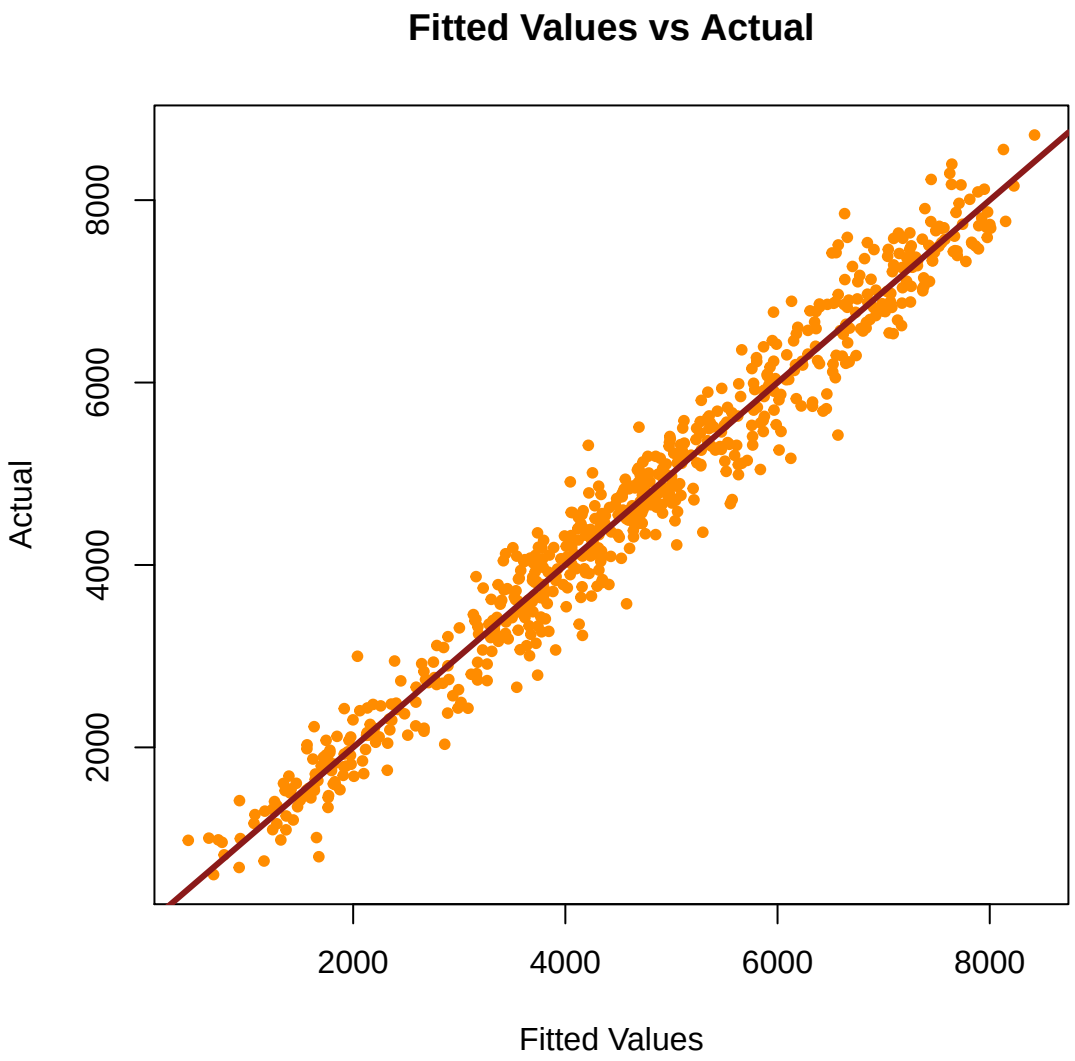
	prediction
Spring	3341
Summer	4175
Fall	5068
Winter	5539

Results

Use Case 1 - Operational Expenses Analysis

Our best model included both interactions and polynomial terms and by seeing the below plot we can say that the fitted values for this model are quite close to the actual values.

```r
plot(
  bike_mod_all_11$fitted.values,
  data_3$cnt[filter],
  main = "Fitted Values vs Actual",
  xlab = "Fitted Values",
  ylab = "Actual",
  col = "darkorange",
  pch = 20
)
abline(0, 1, col = "firebrick4", lwd = 3)
```



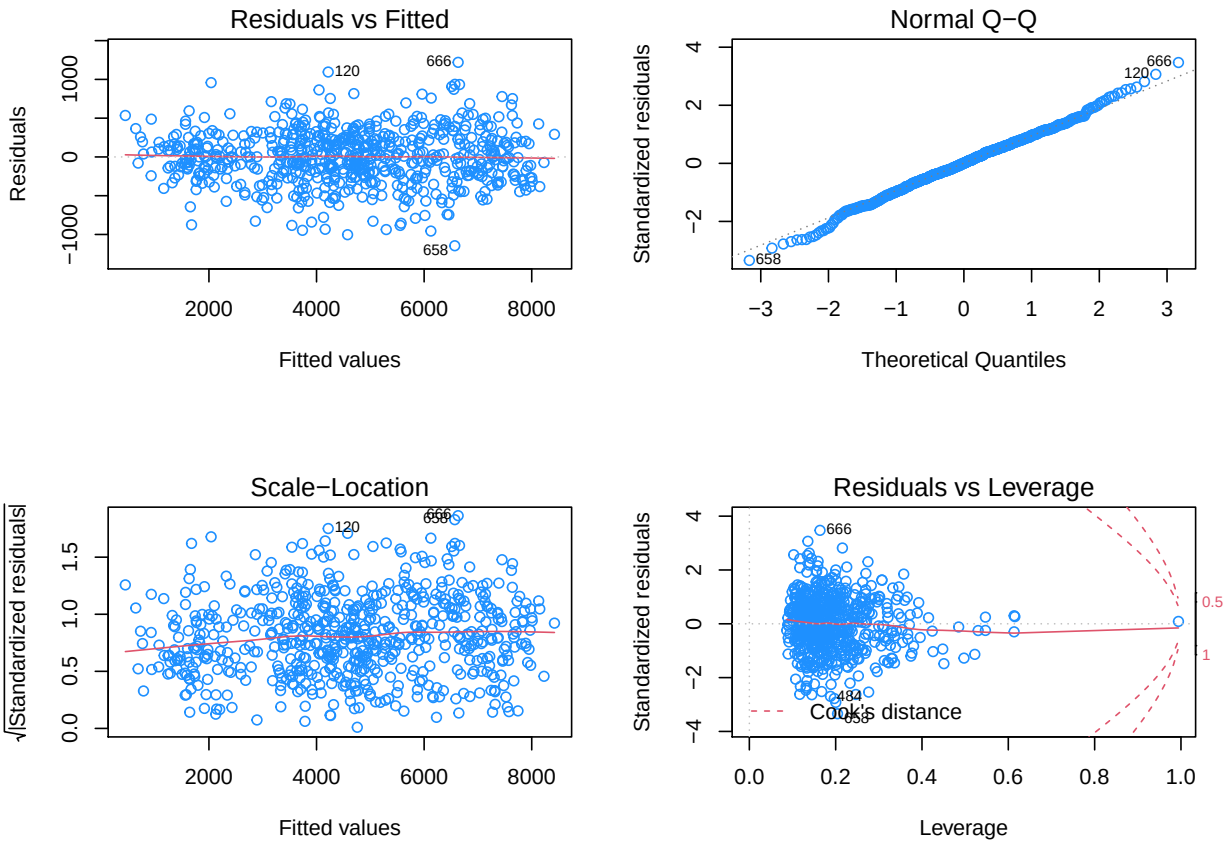

```
summary(bike_mod_all_11)$r.squared
```

```
## [1] 0.9665
```

The adjusted R^2 indicates that the model explains 0.9665 of the variance in the data.

The LOOCV-RMSE of the model is 426.8036.

```
par(mfrow = c(2, 2))
plot(bike_mod_all_11,col = 'dodgerblue') # do diagnostics
```



The diagnostic plots all appear to be acceptable, although there are still some points with very high leverage.

```
sqrt(mean(bike_mod_all_11$residuals^2))
```

```
## [1] 343.8
```

The model has an RMSE of 344 which means that on an average the model predictions will deviate by this factor.

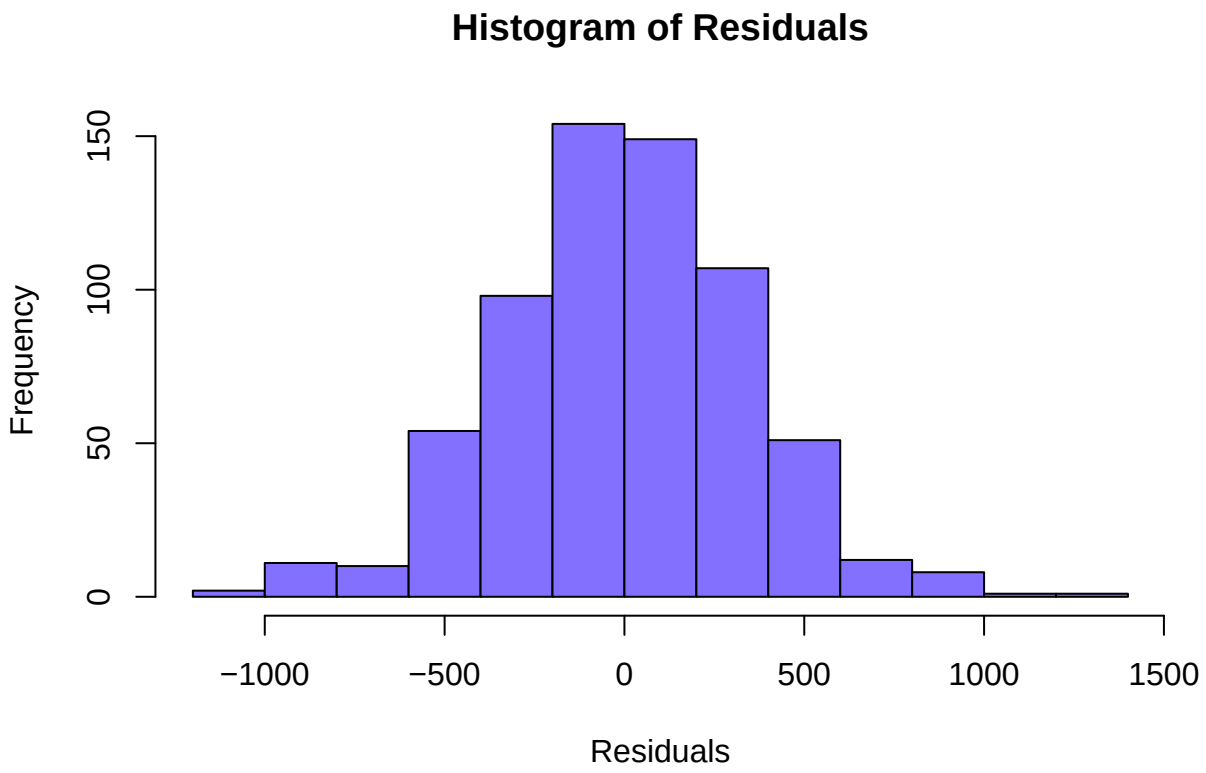
```
# Count the number of high VIF's in the model
sum(vif(bike_mod_all_11) > 5)
```

```
## Warning in v1 * v2: longer object length is not a multiple of shorter object
## length
```

```
## [1] 106
```

This was expected due to high number of categorical variables and their interactions. Since, we are mainly focusing on demand prediction, we tuned our model for better prediction by sacrificing interpretability.

```r
hist(resid(bike_mod_all_11),
     col = "lightslateblue",
     main = "Histogram of Residuals", xlab = "Residuals")
```



The histogram of residuals looks nearly normal which confirms the fact that the model has noise which are normally distributed and centered about 0.

So far, we have tried multiple approaches taught to us in our class to evolve our understanding of what should work to have a model with a good performance without sacrificing the high variance nature of the model. At this point, we have reached a decent model where we have high adjusted R^2 value and low LOOCV RMSE.

Use Case 2 - Targeted Marketing Strategy Analysis

Based on Collinearity check, we observed that these predictors **registered**, **casual**, **temp**, **atemp** and **yr** will not be helpful in predicting the recommended season to run a targeted marketing strategy. We created a model without these predictors and ran AIC and BIC comparison. We observed that backward search AIC has better model that can be used as a baseline to explore which season requires marketing strategy.

Discussion

Use Case 1 - Operational Expenses Analysis

While the final model contained a large number of predictors, making it difficult to interpret. The results indicate that it is very good at explaining the relationship between `weather`, `season` and bike sharing rentals. We tried evaluating using BIC to reduce the size of the final model, however, doing so resulted in a lowered adjusted R2 and higher LOOCV-RMSE, so we preferred the AIC selected model.

As expected by looking this model, the weather situation is an especially important predictor. Both by itself and in its interactions with other predictors, indicating that rain has a significant impact on the number of rentals, especially the interaction between `light rain` and `windspeed`.

The high adjusted R^2 of the model shows that a very large portion of the variance in the data can be explained by this model, which would make it very useful for predicting demand for bikes and consequently predicting the operational cost for the Capital Bike Share.

Future Improvement to this model:

1. Looking at the distribution of data between demand and type of day as well as from the outlier detection through cook's method it will be a better option to build separate models for Holidays, Workdays and Weekends.
2. We can also extend this model to apply Seasonality analysis to see how season and days of week affect the overall demand.

Use Case 2 - Targeted Marketing Strategy Analysis

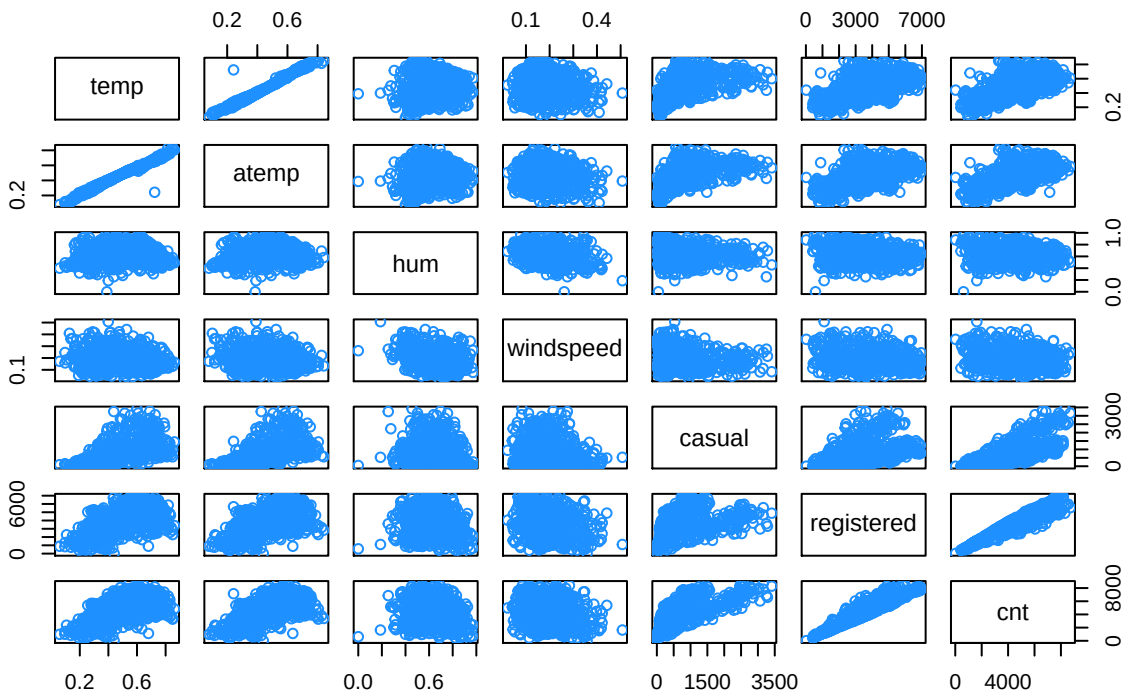
On our preliminary assumption, Winter will be the season that would need an enhanced marketing strategy but it turned out upon finding the best model, Spring is the season that would require a better marketing strategy to attract additional customers.

If given additional time, we would like to explore why Spring season got the lowest prediction compared to our initial assumption of Winter season. Are there any methods that would lead us to a different best model? Are people biking a lot during Winter because they want the excitement of having snow and cold temperature? It will also be interesting to gather data around how often a customer rents a bike, a comparison of `registered` versus `casual` bike riders, and the type of bicycle – mountain bike, fat bike, or road bike.

Appendix

Pairs Plot between different numeric Variables

```
#pairs(day_data, col="dodgerblue")
pairs(day_data[,10:16],
      main = "Pairs Plot for Numeric Features",
      col="dodgerblue"
      )
```



By looking the pairs plot we can deduce that - - there is a relationship between registered users and total ridership count(cnt). - there is also a relationship between temp and cnt.

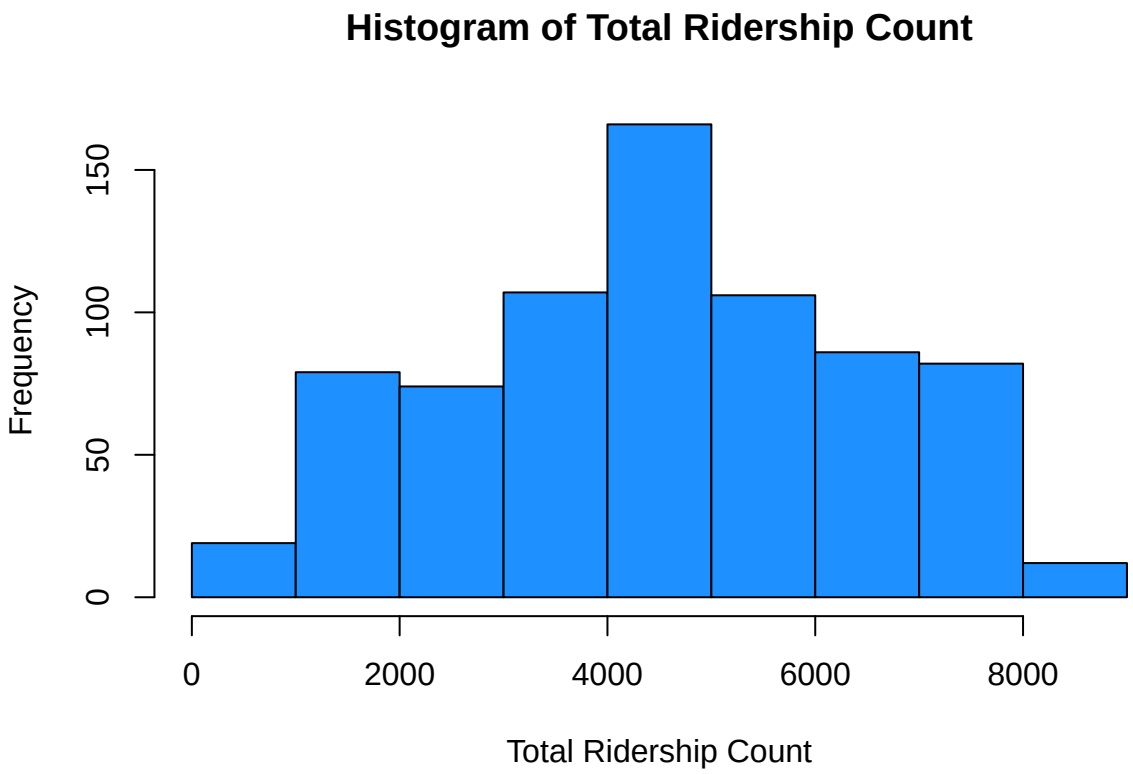
Correlation between numeric Variables

```
cor(day_data[,10:16])
```

```
##                temp    atemp      hum windspeed   casual registered     cnt
## temp         1.0000  0.9917  0.12696   -0.1579  0.54328    0.54001  0.6275
## atemp        0.9917  1.0000  0.13999   -0.1836  0.54386    0.54419  0.6311
## hum          0.1270  0.1400  1.00000   -0.2485 -0.07701   -0.09109 -0.1007
## windspeed   -0.1579 -0.1836 -0.24849    1.0000 -0.16761   -0.21745 -0.2345
## casual       0.5433  0.5439 -0.07701   -0.1676  1.00000    0.39528  0.6728
## registered   0.5400  0.5442 -0.09109   -0.2174  0.39528    1.00000  0.9455
## cnt          0.6275  0.6311 -0.10066   -0.2345  0.67280    0.94552  1.0000
```

Distribution of response variable (Bike Ridership Count)

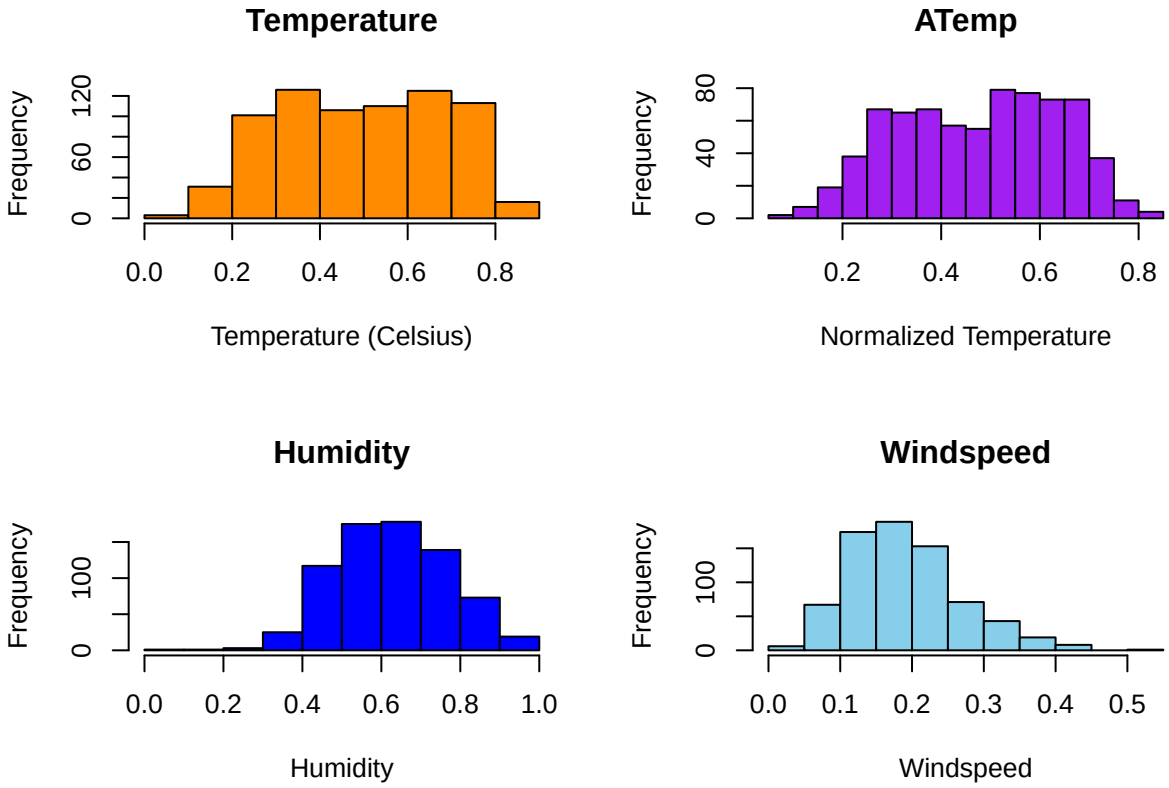
```
hist(day_data$cnt,
     col = "dodgerblue",
     main = "Histogram of Total Ridership Count",
     xlab = "Total Ridership Count")
```



It looks like, the bike ridership count variable is following normal distribution.

Data Distribution of Numeric Variables

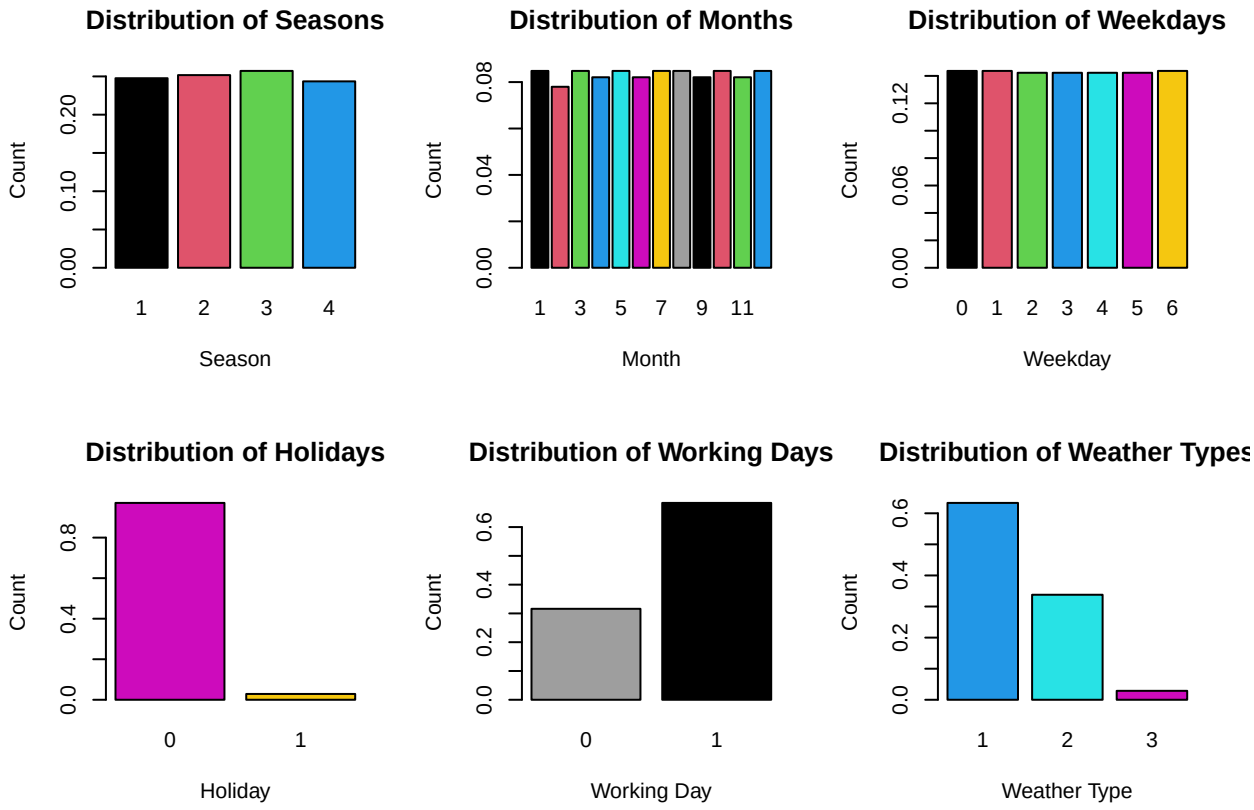
```
par(mfrow=c(2,2))
hist(day_data[,10], main="Temperature", xlab = "Temperature (Celsius)", col = "darkorange")
hist(day_data[,11], main="ATemp", xlab = "Normalized Temperature", col = "purple")
hist(day_data[,12], main="Humidity", xlab = "Humidity", col = "blue")
hist(day_data[,13], main="Windspeed", xlab = "Windspeed", col = "skyblue")
```



These plots do not exactly looks like normally distributed, however windspeed and humidity looks close to Normal Distribution.

Data Distribution of Categorical Variables

```
par(mfrow = c(2,3))
barplot(prop.table(table(day_data$season)), col = 1:4, main = "Distribution of Seasons", xlab = "Season
barplot(prop.table(table(day_data$mnth)), col = 1:12,  main = "Distribution of Months", xlab = "Month",
barplot(prop.table(table(day_data$weekday)), col = 1:12,  main = "Distribution of Weekdays", xlab = "We
barplot(prop.table(table(day_data$holiday)), col = 6:12,  main = "Distribution of Holidays", xlab = "Ho
barplot(prop.table(table(day_data$workingday)), col = 8:12,  main = "Distribution of Working Days", xla
barplot(prop.table(table(day_data$weathersit)), col = 12:15,  main = "Distribution of Weather Types", x
```

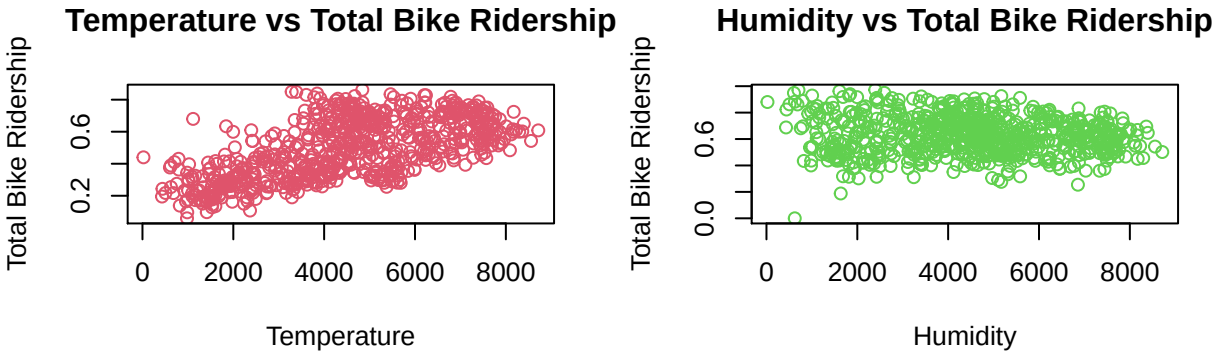


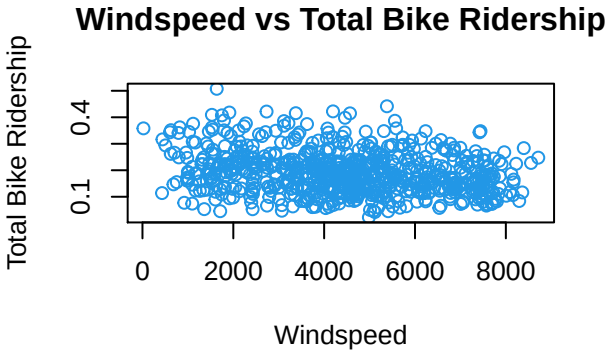
We can conclude follwing points by seeing the above plots -

- Uniformly Distribution - Seasons, Months, Weekday
- Weather seems to be more clear than Mist and Light Precipitation.
- There are very few holidays.
- There are more working days compared to Non working days.

More Plots (Relationship between Bike Ridership and Temp, Humidity, Windspeed)

```
par(mfrow=c(2,2))
plot(day_data$temp ~ day_data$cnt,
     data = day_data,
     main = "Temperature vs Total Bike Ridership",
     xlab = "Temperature",
     ylab = "Total Bike Ridership", col = 2)
plot(day_data$hum ~ day_data$cnt,
     data = day_data,
     main = "Humidity vs Total Bike Ridership",
     ylab = "Total Bike Ridership",
     xlab = "Humidity", col = 3)
plot(day_data$windspeed ~ day_data$cnt,
     data = day_data,
     main = "Windspeed vs Total Bike Ridership",
     ylab = "Total Bike Ridership",
     xlab = "Windspeed", col = 4)
```

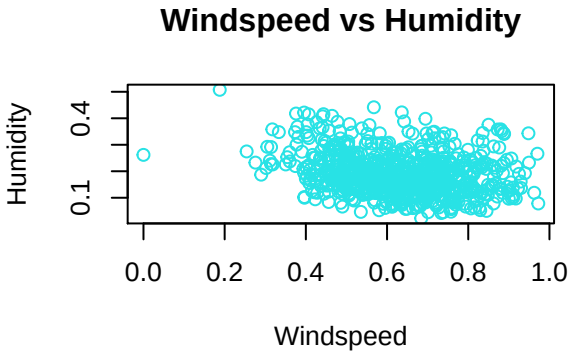


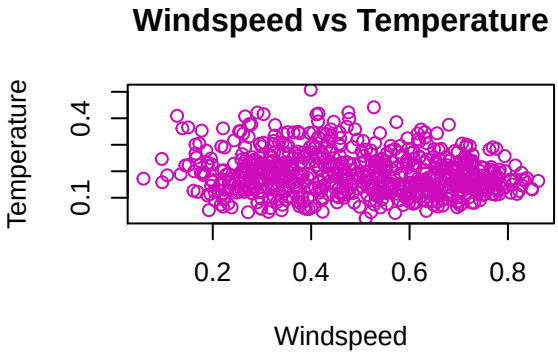


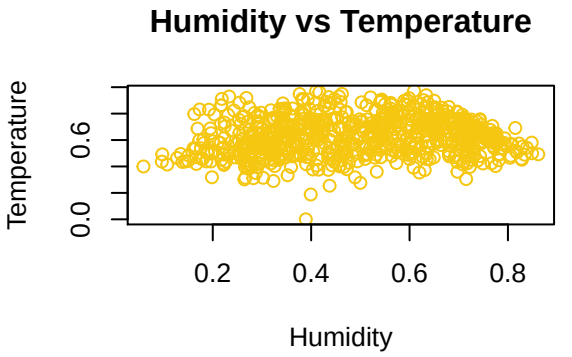
By looking the above plot, it looks like that there is a linear relationship between temperature and total bike ridership, however it could be more polynomial. Nothing concretely can be said about the relationship between Humidity and Total Bike Ridership and Windspeed and Total Bike Ridership.

More Plots (Relationship between Temperature, Humidity, Windspeed)

```
par(mfrow=c(2,2))
plot(day_data$windspeed ~day_data$hum,
     data = day_data,
     main = "Windspeed vs Humidity",
     ylab = "Humidity",
     xlab = "Windspeed", col = 5)
plot(day_data$windspeed ~ day_data$temp,
     data = day_data,
     main = "Windspeed vs Temperature",
     ylab = "Temperature",
     xlab = "Windspeed", col = 6)
plot(day_data$hum ~ day_data$temp,
     data = day_data,
     main = "Humidity vs Temperature",
     ylab = "Temperature",
     xlab = "Humidity", col = 7)
```



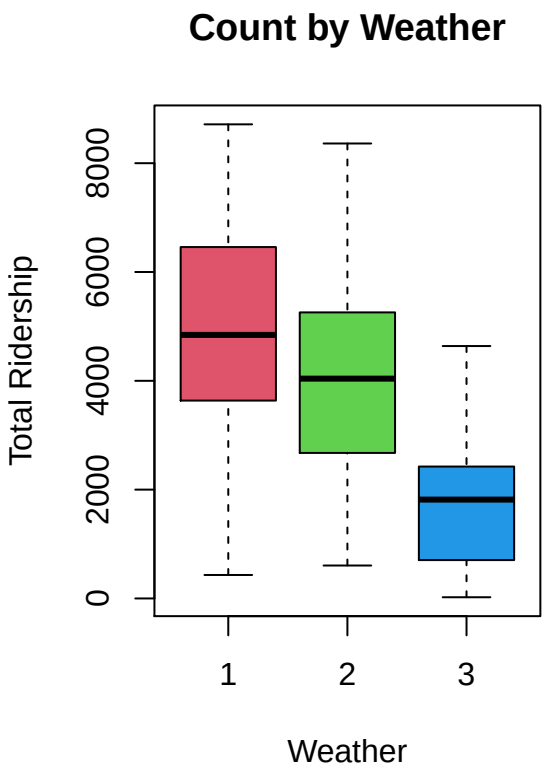


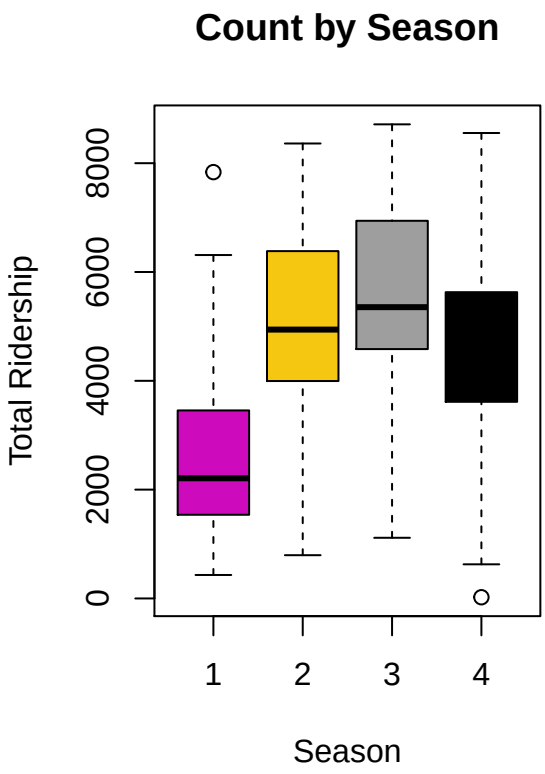


By looking the above plot, it looks like that there is a some relationship between Windspeed, Humidity and Temperature, it could be in some polynomial in nature, we need to explore more to know the true nature of these relationships.

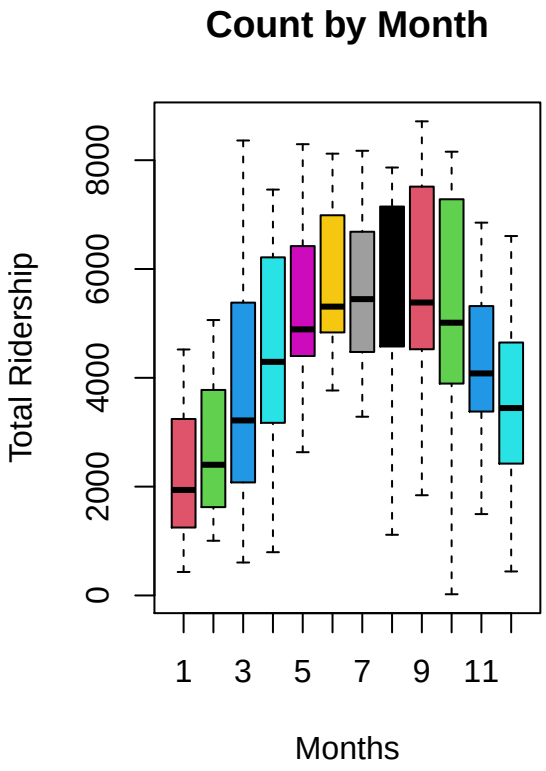
Box Plot Insight for Categorical Variables and its impact on ridership count.

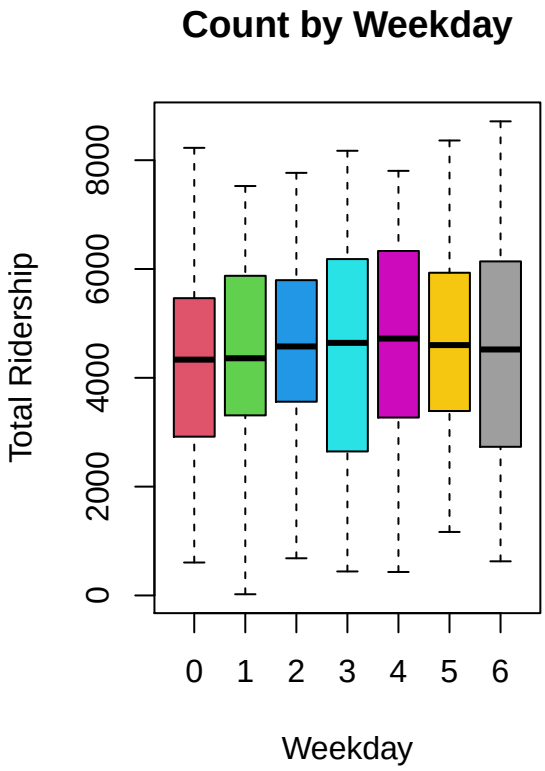
```
par(mfrow = c(1,2))
boxplot(cnt ~ weathersit,
        data = day_data,
        col = 2:4,
        main = "Count by Weather",
        xlab = "Weather",
        ylab = "Total Ridership")
boxplot(cnt ~ season,
        data = day_data,
        col = 6:12,
        main = "Count by Season",
        xlab = "Season",
        ylab = "Total Ridership")
```



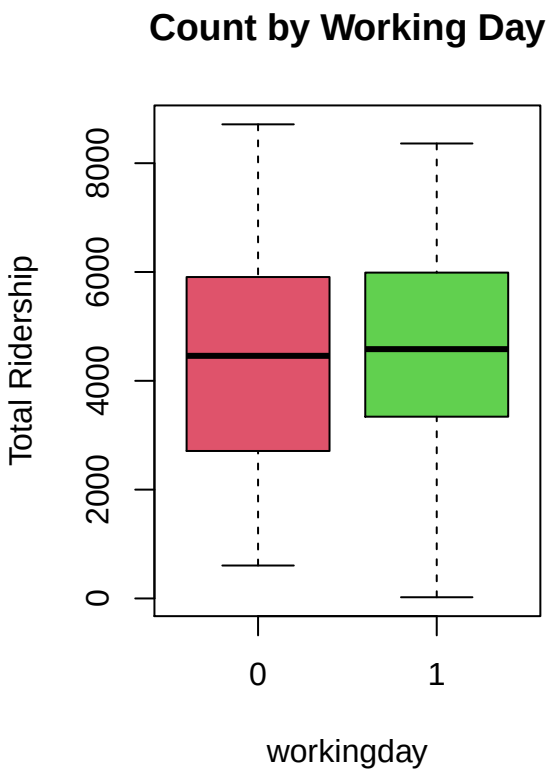


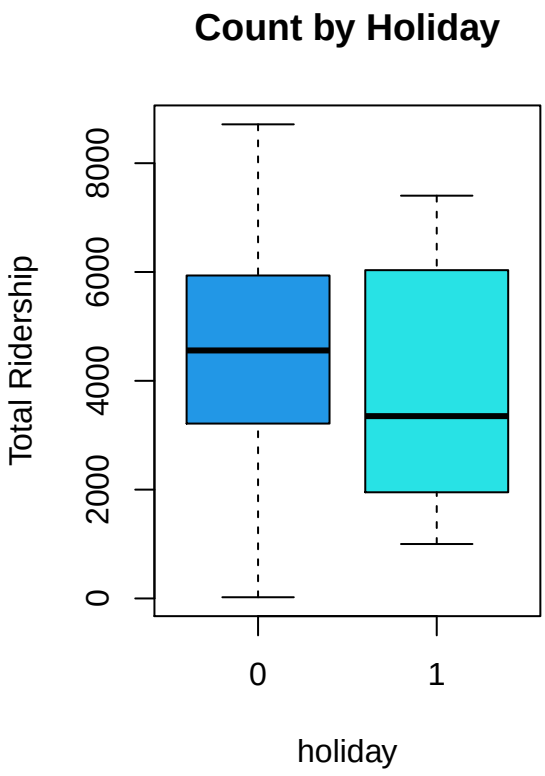

```
boxplot(cnt ~ mnth,
        data = day_data,
        col = 2:14,
        main = "Count by Month",
        xlab = "Months",
        ylab = "Total Ridership")
boxplot(cnt ~ weekday,
        data = day_data,
        col = 2:8,
        main = "Count by Weekday",
        xlab = "Weekday",
        ylab = "Total Ridership")
```






```
boxplot(cnt ~ workingday,
        data = day_data,
        col = 10:12,
        main = "Count by Working Day",
        ylab = "Total Ridership")
boxplot(cnt ~ holiday,
        data = day_data,
        col = 12:15,
        main = "Count by Holiday",
        ylab = "Total Ridership")
```





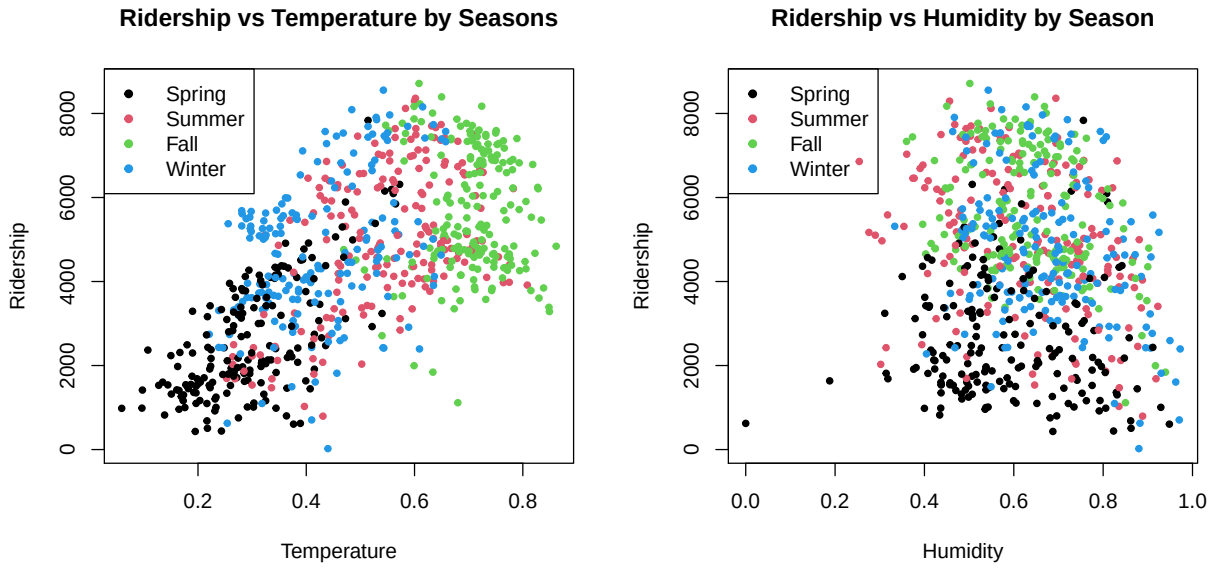
By seeing the above plots we can deduce that - - Total Ridership increases on Clear weather. - Total Ridership increases on Summer and Fall weather. - Total Ridership incrases on June, July, August and Sepetember. - Weekday and Working Day don't offer much interest from boxplot. - Mean Ridership on Holidays is lower than on non-Holidays

Interactions

We will now consider the interactions between the categorical features we identified above and some numerical features.

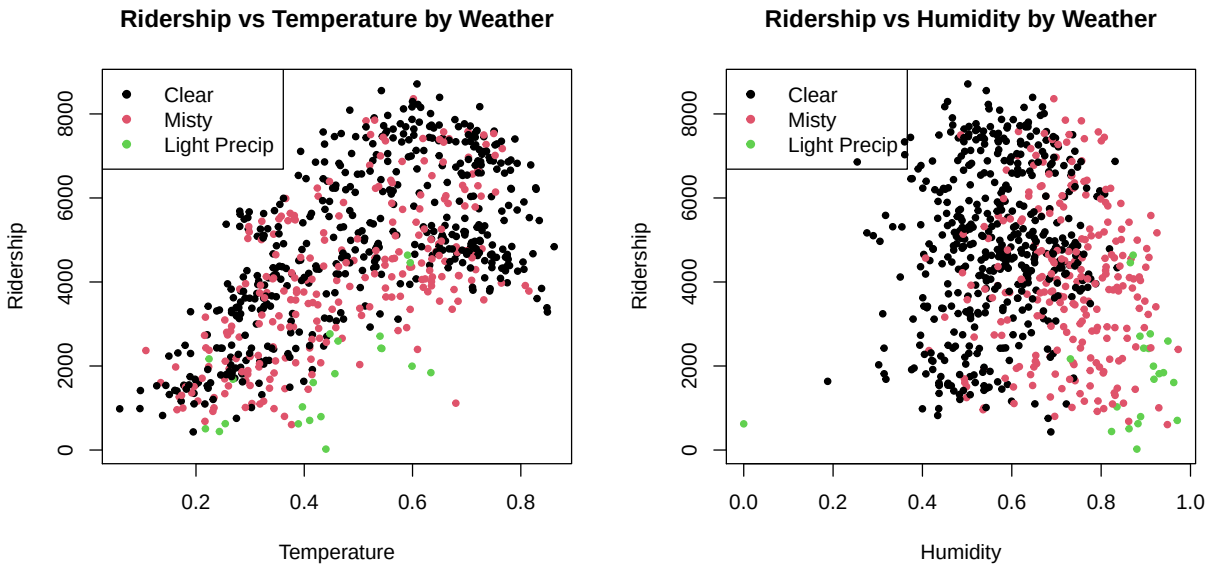
```
par(mfrow = c(1,2))
plot(day_data$temp, day_data$cnt,
     col = day_data$season,
     pch = 20,
     main = "Ridership vs Temperature by Seasons",
     xlab = "Temperature",
     ylab = "Ridership")
legend("topleft", legend = c("Spring", "Summer", "Fall", "Winter"),
       col = 1:5, lwd = 2, lty = c(0,0), pch = 20)
plot(day_data$hum,
     day_data$cnt,
     col = day_data$season,
     pch = 20,
     main = "Ridership vs Humidity by Season",
     xlab = "Humidity",
```

```
     ylab = "Ridership")
legend("topleft", legend = c("Spring", "Summer", "Fall", "Winter"),
       col = 1:5, lwd = 2, lty = c(0,0), pch = 20)
```



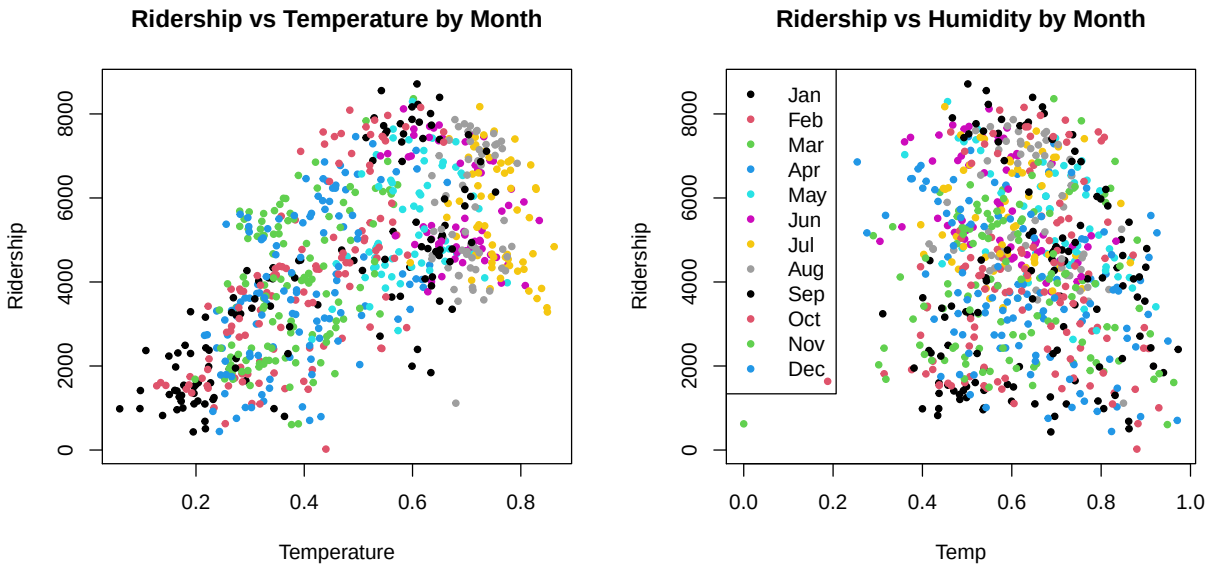
As we would expect the temperature to be correlated to the season the Ridership vs Temperature plot doesn't show any interesting patterns. However, the Ridership vs Humidity plot has vertical clusters which could indicate interesting correlations.

```
par(mfrow=c(1,2))
plot(day_data$temp,
     day_data$cnt,
     col = day_data$weathersit,
     pch = 20,
     main = "Ridership vs Temperature by Weather",
     xlab = "Temperature",
     ylab = "Ridership")
legend("topleft", legend = c("Clear", "Misty", "Light Precip"), col = 1:5, lwd = 2, lty = c(0,0), pch =
plot(day_data$hum,
     day_data$cnt,
     col = day_data$weathersit,
     pch = 20,
     main = "Ridership vs Humidity by Weather",
     xlab = "Humidity",
     ylab = "Ridership")
legend("topleft", legend = c("Clear", "Misty", "Light Precip"), col = 1:5, lwd = 2, lty = c(0,0), pch =
```



Similarly, as we would expect Humidity to be correlated with Weather the Ridership vs Humidity plot doesn't show any interesting patterns while the Ridership vs Temperature plot indicates that there may be some

```
par(mfrow = c(1,2))
plot(day_data$temp,
     day_data$cnt,
     col = day_data$mnth,
     pch = 20,
     main = "Ridership vs Temperature by Month",
     xlab = "Temperature",
     ylab = "Ridership")
plot(day_data$hum,
     day_data$cnt,
     col = day_data$mnth,
     pch = 20,
     main = "Ridership vs Humidity by Month",
     xlab = "Temp",
     ylab = "Ridership")
legend("topleft", legend = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov
```



It is difficult to draw conclusions from the plots above, however both interaction may merit further investigation.

This report was prepared by Team Outlier – Michael Alpas, Mohit Singh and Upendra Yadav.
