Bike Sharing Linear Regression Analysis

Jack Bignell, David Chen, Sampreeth Salveru

06/12/2020

Contents

1. Abstract

This project is focused on analyzing the daily count of rental bikes for casual users between the years 2011 and 2012 in the Capital bikeshare system in Washington, D.C. with its corresponding weather and seasonal information. We want to determine which attributes of the dataset can be used to build a linear regression model to predict the number of casual riders. After conducting relevant transformations and model selections, we are interested in determining which factors have the biggest effect on daily casual rider counts and the average casual rider counts for an average day in each season.

2. Problem and Motivation

Bike sharing is a new form of transportation for those who need to travel around short distances or a crowded city. Bicycles are spread across a certain area and users with the app can unlock the bike and ride it to their destination. In short, the program is a bicycle rental system charging by the hour and minute. Bike sharing has seen a sharp rise in popularity thanks to its lower cost and convenience compared to the mainstream forms of public transportation, such as the bus and metro. Furthermore, it offers a way for people to stay healthy and minimize environmental impact. There are over 500 bike sharing programs around the world, including one in UCSB and IV. The Capital bikeshare system in Washington D.C. collected daily rider counts with corresponding weather information between 2011 through 2012. We are interested in the external factors that can affect daily rider counts, such as the type of day and weather. These factors are often outside of the control of bike sharing companies, and being able to scale according to demand is often very important to minimize operating costs.

3. Data

This data was sourced from the UCI Machine Learning Repository. The dataset is titled Bike Sharing and has 731 observations and 16 variables. In the beginning of our analysis we only considered 8 variables and removed the rest that weren't relevant.

Response:

- casual: count of casual users

Predictors:

- season : season (1:winter, 2:spring, 3:summer, 4:fall)
- holiday : weather day is holiday or not
- weekday: day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
- weathersit :
- 1: Clear, Few clouds, Partly cloudy, Partly cloudy
- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- 4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog
- temp : Normalized temperature in Celsius.
- atemp: Normalized feeling temperature in Celsius.
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)

5. Questions of Interest

We want to investigate which variables in our dataset are useful in predicting our response of casual rental bikes users. We will first analyze the relationship between our 9 predictors, listed above, and our casual user count. Would we want to consider all of our predictors or only a select few? Is there a significant relationship between our total count and predictors, and is the relationship positive or negative? Do some predictors have a stronger relationship than others? What is the average casual rider count for an average day in each season? What factors have the biggest effect on daily casual rider counts?

6. Regression Methods

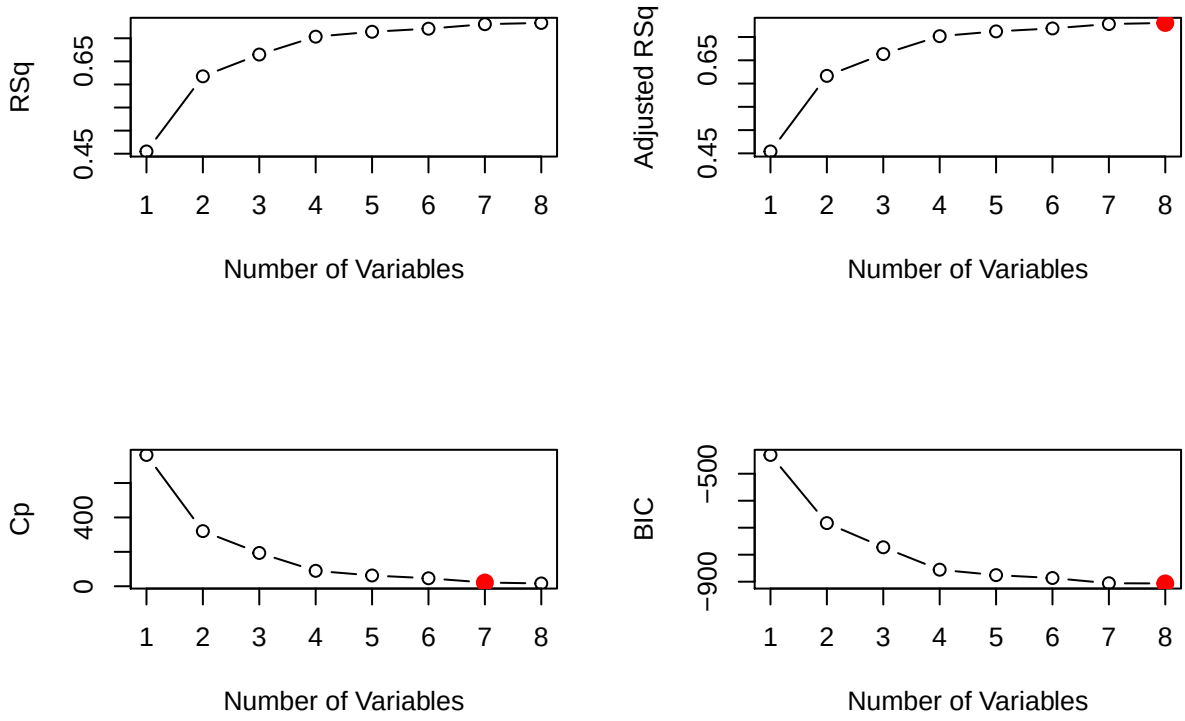
We first begin by looking at the data and see if we need to scale any of the data and deal with categorical variables. Next we do preliminary scatter plots and diagnostics of an initial full model fit to determine if any assumptions are broken and see if the predictors of response need transformations. We apply the required transformations and begin model fitting. Since we are only working with very few predictors, we are going to use regsubsets to see which reduced model offers the best overall performance. We then supplement this with a hybrid BIC selection starting from our regsubsets model, to compare results from regsubsets. Once we have a reduced model, we check the assumptions again for any violations. If none, we proceeded to checking whether our model should be parallel or non parallel, by testing interaction between our categorical season variable and other variables. We use a partial F test to determine which interactions are necessary. This concludes the model building. With our model, we can predict a new interval with average days in each season and see the expected mean response. We can also check which coefficients have the biggest influence on our response.

Exploratory Analysis

Before we begin our regression analysis, we first conduct an exploratory analysis of our data. We begin with plotting our Residuals vs Fitted and Normal Q-Q plot and notice that we have to conduct transformation on both our predictors and response. We run a BoxCox transformation on our response and our plot shows a value close to 0. A value of 0 in BoxCox refers to a natural log transformation on our response. For the predictors we run the PowerTransform function which determines that our predictor, windspeed, will need to be transformed by a square root. Once all of our transformations are finished, we plot an added variable plot and determine that we cannot make adequate assumptions due to the large number of predictors.

7. Regression Analysis

We begin model building after applying some transformations to our response and predictors, from our discoveries in the exploratory data analysis. We will show the full details in the diagnostic checks. Since we are only working with 8 variables, we opt to use regsubets for initial model selection.



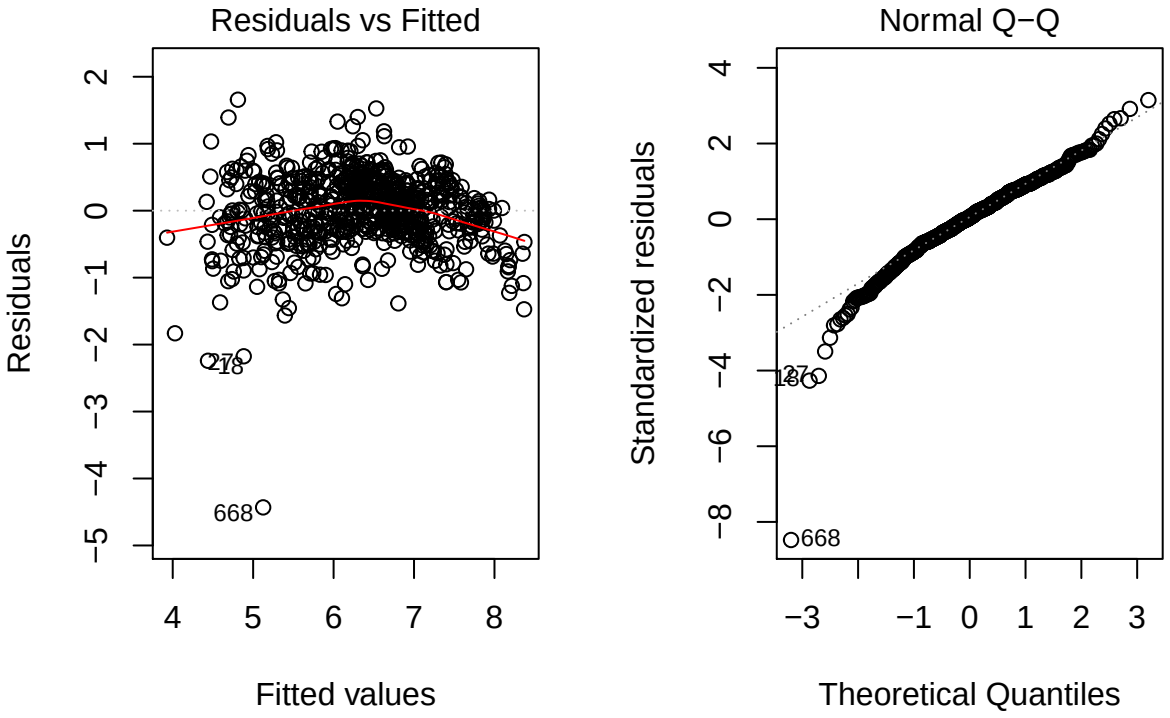
The result shows that a model with 7 and 8 predictors are both very good, with very similar values in adjusted R squared, Cp Mallows, and BIC. We prefer a simpler model and choose to go with a 7 predictor model with season, holiday, workingday, weathersit, temp, hum, and sqrt(windspeed). To confirm these results, we decided to use the step function to perform a hybrid selection starting from the model found in regsubsets.

```
## Start:  AIC=-920.94
## log(casual) ~ season + holiday + workingday + weathersit + temp +
##     hum + sqrt(windspeed)
##
##                   Df Sum of Sq    RSS     AIC
## + atemp            1     0.790 201.00 -921.81
## <none>                         201.79 -920.94
## - holiday          1     1.194 202.98 -918.63
## - hum              1     7.244 209.03 -897.16
## - sqrt(windspeed)  1     9.650 211.44 -888.79
## - weathersit       1    10.388 212.18 -886.24
## - season           3    44.405 246.19 -781.55
## - temp             1    87.639 289.43 -659.28
## - workingday       1   114.899 316.69 -593.48
##
## Step:  AIC=-921.81
## log(casual) ~ season + holiday + workingday + weathersit + temp +
##     hum + sqrt(windspeed) + atemp
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         201.00 -921.81
```

```
## - atemp            1     0.790 201.79 -920.94
## - holiday          1     1.116 202.11 -919.76
## - temp             1     1.395 202.39 -918.75
## - hum              1     7.531 208.53 -896.92
## - sqrt(windspeed)  1     8.353 209.35 -894.04
## - weathersit       1    10.010 211.01 -888.28
## - season           3    41.563 242.56 -790.41
## - workingday       1   114.624 315.62 -593.94
```

```
##
## Call:
## lm(formula = log(casual) ~ season + holiday + workingday + weathersit +
##     temp + hum + sqrt(windspeed) + atemp, data = day)
##
## Coefficients:
##     (Intercept)     seasonspring     seasonsummer     seasonwinter
##         6.94085          0.11629         -0.31465         -0.51108
##         holiday       workingday       weathersit             temp
##        -0.24229         -0.88447         -0.28371          0.04887
##             hum  sqrt(windspeed)            atemp
##        -0.97842         -0.16201          1.64750
```

Using BIC as the metric, the step function running in both direction arrives at the same model that we began with. We decided to settle on these 7 predictors and move on to parallel versus non parallel testing of our season variable.



We add interaction terms of season against all the other variables, and then do a Partial F test of the interaction betas.

```
## Analysis of Variance Table
##
## Model 1: log(casual) ~ season + holiday + workingday + weathersit + temp +
##     hum + sqrt(windspeed)
```

```
## Model 2: log(casual) ~ workingday * season + temp * season + weathersit *
##     season + hum * season + sqrt(windspeed) * season + holiday *
##     season
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    721 201.79
## 2    703 152.03 18     49.76 12.783 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H_0 : $\beta_{season,holiday} = \beta_{season,workingday} = \beta_{season,weathersit} = \beta_{season,hum} = \beta_{season,temp} = \beta_{season,\sqrt{windspeed}} = 0$(all interaction terms equal 0)

H_1: At least one of our interaction β does not equal to zero

F statistic = 12.783, p-value < 2.2e-16

We reject our null hypothesis and conclude that the non parallel model is very significant.

Our next step is to identify which specific interaction terms are significant, so we do partial F-tests of each of the interaction betas using the summary function.

```
##
## Call:
## lm(formula = log(casual) ~ workingday * season + temp * season +
##     weathersit * season + hum * season + sqrt(windspeed) * season +
##     holiday * season, data = day)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9287 -0.2607  0.0174  0.2837  1.7724
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.822483   0.306697  25.506  < 2e-16 ***
## workingday                   -0.989438   0.079203 -12.492  < 2e-16 ***
## seasonspring                 -0.151234   0.447367  -0.338 0.735423
## seasonsummer                  1.377974   0.610462   2.257 0.024298 *
## seasonwinter                 -1.834652   0.417970  -4.389 1.31e-05 ***
## temp                          0.116812   0.008583  13.610  < 2e-16 ***
## weathersit                   -0.333364   0.085950  -3.879 0.000115 ***
## hum                          -2.350073   0.421042  -5.582 3.41e-08 ***
## sqrt(windspeed)              -0.255226   0.048828  -5.227 2.27e-07 ***
## holiday                      -0.422621   0.201228  -2.100 0.036066 *
## workingday:seasonspring       0.091849   0.111075   0.827 0.408565
## workingday:seasonsummer       0.220302   0.110149   2.000 0.045881 *
## workingday:seasonwinter       0.053997   0.110595   0.488 0.625531
## seasonspring:temp            -0.065276   0.011462  -5.695 1.81e-08 ***
## seasonsummer:temp            -0.125394   0.015480  -8.100 2.42e-15 ***
## seasonwinter:temp             0.049617   0.012115   4.096 4.70e-05 ***
## seasonspring:weathersit      -0.037419   0.126777  -0.295 0.767965
## seasonsummer:weathersit       0.171580   0.123483   1.390 0.165121
## seasonwinter:weathersit       0.069606   0.115715   0.602 0.547679
## seasonspring:hum              1.670684   0.526067   3.176 0.001560 **
## seasonsummer:hum              1.109087   0.564490   1.965 0.049836 *
## seasonwinter:hum              1.135780   0.514588   2.207 0.027625 *
## seasonspring:sqrt(windspeed)  0.130103   0.074752   1.740 0.082217 .
```

```
## seasonsummer:sqrt(windspeed)  0.119486   0.076955   1.553 0.120951
## seasonwinter:sqrt(windspeed) -0.005586   0.070131  -0.080 0.936537
## seasonspring:holiday         -0.136821   0.317898  -0.430 0.667043
## seasonsummer:holiday          0.637488   0.315717   2.019 0.043848 *
## seasonwinter:holiday          0.338511   0.275023   1.231 0.218792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.465 on 703 degrees of freedom
## Multiple R-squared:  0.7992, Adjusted R-squared:  0.7915
## F-statistic: 103.6 on 27 and 703 DF,  p-value: < 2.2e-16
```

We see very significant results in all interactions between each season and temperature, and some significance in the interaction between season and humidity. This is probably because season has a big effect on temperature and humidity and are not independent. The p-values between each season interacting with temperature are all close to 0. The p-values for interactions between each season and humidity are all less than 0.05. There are some interactions that are only midly significant in only one season with p-values close to 0.05. Since not every season has singificant interactions with the other predictor, we decided to test it using a partial F-test using annova.

```
## Analysis of Variance Table
##
## Model 1: log(casual) ~ season + holiday + workingday + weathersit + temp +
##     hum + sqrt(windspeed) + season * temp + season * hum
## Model 2: log(casual) ~ workingday * season + temp * season + weathersit *
##     season + hum * season + sqrt(windspeed) * season + holiday *
##     season
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    715 155.96
## 2    703 152.03 12    3.9321 1.5152 0.1133
```

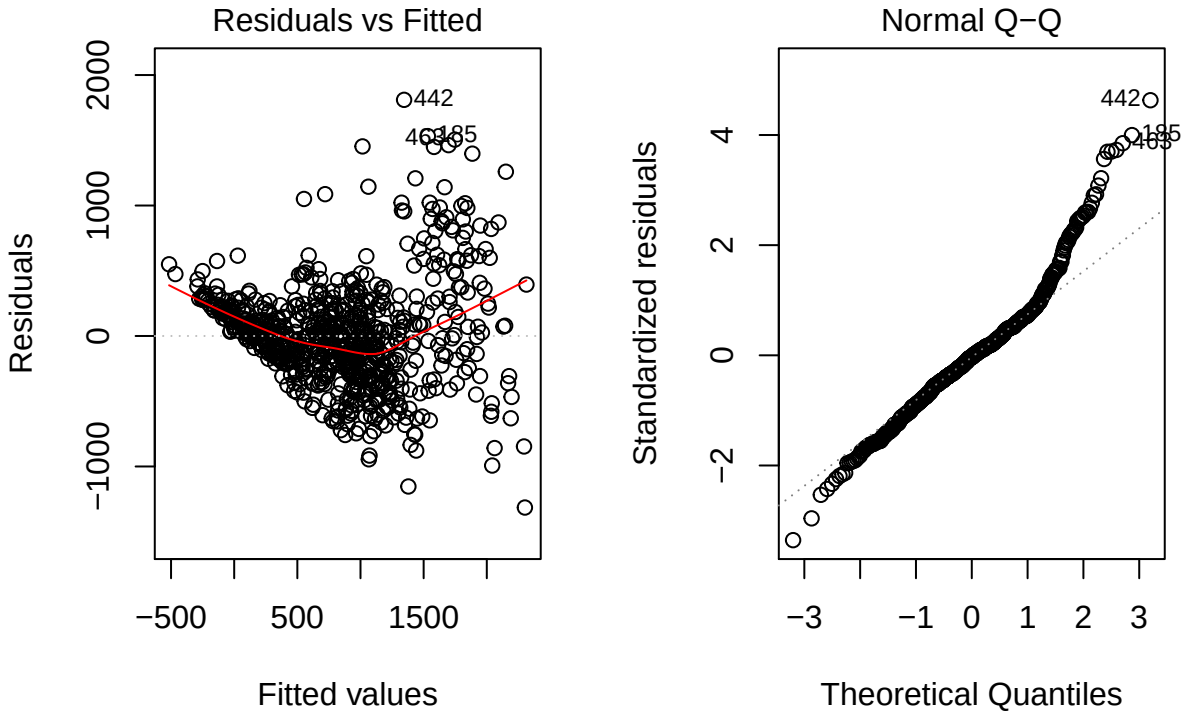
H_0: $\beta_{season,holiday} = \beta_{season,workingday} = \beta_{season,weathersit} = \beta_{season,\sqrt{windspeed}} = 0$(all interaction terms equal 0)
H_1: At least one of our interaction β does not equal to zero

F-statistic = 1.52, p-value = 0.1133

Since the p-value is bigger than our alpha of 0.05, we fail to reject H0. We conclude that our reduced model with only two interactions terms is significant, and prefer it over the full model.

This is our final model which we can use to answer our questions.

Diagnostic Checks

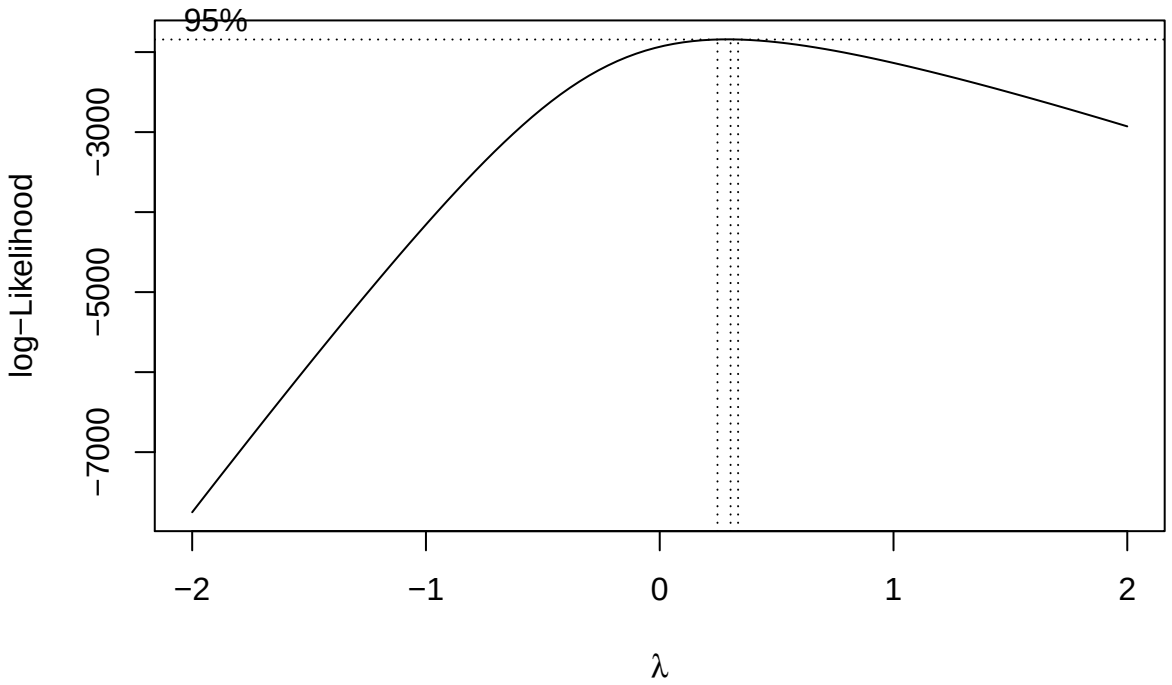


For our diagnostic checks we will determine if our model meets assumptions for a linear regression model. The assumptions are determined by confirming Linearity, Independent, Normality, and Equal Variance (LINE). In our preliminary model we plotted our Residuals vs. Fitted plot and Normal Q-Q plot to check our assumptions. The plots showed that our residuals weren't randomly distributed with equal variance, and also our normality was clearly violated in the Q-Q plot.

To improve our model performance, we began to transform our predictors using the PowerTransformation function. We determined that the only predictor that needed a transformation was windspeed. It had a rounded power transformation value of 0.50, which indicates a square root transformation. The other predictors were close to 1 which determined that no transformation was required. We also ran a transformation on the response by using the BoxCox function. The function returned a plot in which it was seen as the closest round number to pick for a transformation was 0, which indicates a natural log transformation on the response.

```
##           Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## temp      0.7841162        0.78    0.6723005    0.8959318
## atemp     0.9420749        1.00    0.8233793    1.0607706
## hum       0.9199398        1.00    0.7476358    1.0922437
## windspeed 0.4430640        0.50    0.3077701    0.5783579
```

```
boxcox(day.lm)
```



Since our dataset included the categorical variable, season, we added an interaction term with all other predictors. This further improved our model and we plotted our diagnostics for our final model. The plots of Residuals vs. Fitted plot and Normal Q-Q plot greatly improved compared to our preliminary model. The residuals are randomly distributed with equal variance and the Q-Q shows that our model is close to normal, with a few values that are lightly tailed. We confirm that we built a model with good diagnostics and therefore meet our assumptions for a linear regression model.

Interpretation

log(casual) ~ season + holiday + workingday + weathersit + temp + hum + sqrt(windspeed) + season*temp + season*hum

After completing necessary transformations, model selection, and added interaction terms for our categorical variable, we came to a final mode. Using our final model, we can now show our results to answer the questions of interest we stated in the beginning. Based on our coefficients for our predictors, we see that all of them have a significant relationship in predicting the total count of casual rental bike users. All of our predictors also have a positive relationship with the response. The predictor with the strongest relationship in predicting the response while controlling the other predictors is the season. The different categorical variables for season affect the response greater than all of our other variables. The value for the summer season shows the largest increase for the number of casual riders. We also predicted the average number of casual rental bike riders for an average day in each season. For an average day in winter the number of casual riders is 229 with a confidence interval between (204, 257). For an average day in spring the number of casual riders is 976 with a confidence interval between (872,1094). For an average day in summer the number of casual riders is 1125 with a confidence interval between (1013,1250). For an average day in fall the number of casual riders is 539 with a confidence interval between (486,599).

8. Conclusion

Our findings show that in an average day in Winter, Spring, Summer, and Fall, the average number of riders is 229, 976, 1125, 539. We see that during a warmer period there are many more casual riders than in the frigid winter. All of our variables influence our daily rider counts, especially temperature, season and humidity. In general, the daily casual riders can vary a lot by the season, type of day and weather. It is important to recognize that colder seasons such as winter can result in a large decrease in overall users. However, this

model is built around the data from one company in the Washington D.C. climate. The fitted model is only relevant to the climates which are very similar to that of the capital, and cannot be generalized to other parts of the world. Our model is relatively simple and only accounts for the daily average weather. This means that, for example, during lunch time the weather is very cold but warm throughout the rest of the day so we get less riders than expected. The observation for that day only records the average temperature, which hides valuable information for modeling average casual rider counts. These are some of the shortcomings of our model, and future studies should look at data by the hour or minute.

9. Appendix

```
library(readr)
day <- read_csv("C:/Users/jack/Downloads/day.csv")
library(car)
library(MASS)
library(stats)
for (i in 1:length(day$season)){
  if (day$season[i]==1){
    day$season[i]="winter"

  }
  if (day$season[i]==2){
    day$season[i]="spring"

  }
  if (day$season[i]==3){
    day$season[i]="summer"

  }
  if (day$season[i]==4){
    day$season[i]="fall"

  }
}

day$temp=(day$temp*41)
day$windspeed=(day$windspeed*67)
day[c(69),"hum"]=.0000001

day.lm=lm(casual~season+holiday+weekday+workingday
          +temp+weathersit+atemp+hum+windspeed,data=day)
summary(day.lm)
anova(day.lm)
plot(day.lm,which=1)
plot(day.lm,which=2)
boxcox(day.lm)
avPlots(day.lm)

pwtrday=powerTransform(cbind(temp,atemp,hum,windspeed)~1,data=day)
summary(pwtrday)

influenceIndexPlot(day.lm,id=TRUE)
outlierTest(day.lm)
```

```
daylog.lm=lm(log(casual)~season+holiday+workingday
             +temp+weathersit+atemp+hum+sqrt(windspeed),data=day)
avPlots(daylog.lm)
plot(daylog.lm,which=1)
plot(daylog.lm ,which=2)

library(leaps)
summary.reg=summary(regsubsets(log(casual)~season+holiday+weekday
                               +workingday+temp+weathersit+
                                 atemp+hum+sqrt(windspeed),data=day))
par(mfrow = c(2, 2))

plot(summary.reg$rsq, xlab = "Number of Variables", ylab = "RSq", type = "b")

plot(summary.reg$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "b")
best_adj_r2 = which.max(summary.reg$adjr2)
points(best_adj_r2, summary.reg$adjr2[best_adj_r2],
       col = "red",cex = 2, pch = 20)

plot(summary.reg$cp, xlab = "Number of Variables", ylab = "Cp", type = 'b')
best_cp = which.min(summary.reg$cp[-c(length(summary.reg$cp))])
points(best_cp, summary.reg$cp[best_cp],
       col = "red", cex = 2, pch = 20)

plot(summary.reg$bic, xlab = "Number of Variables", ylab = "BIC", type = 'b')
best_bic = which.min(summary.reg$bic)
points(best_bic, summary.reg$bic[best_bic],
       col = "red", cex = 2, pch = 20)
summary.reg$which


day.0=lm(log(casual)~1,data=day)
dayreg=lm(log(casual)~ season + holiday + workingday
          + weathersit + temp + hum + sqrt(windspeed), data=day)
step(dayreg,scope=list(lower=day.0,upper=daylog.lm),direction="both")

day.red = dayreg

plot(day.red)

avPlots(day.red)
summary(day.red)

upday.red=lm(log(casual) ~ workingday*season + temp*season + weathersit*season +
    hum*season + sqrt(windspeed)*season + holiday*season, data = day)
anova(day.red,upday.red)

summary(upday.red)

redupday=lm(log(casual) ~ season + holiday + workingday + weathersit +
    temp + hum + sqrt(windspeed) + season*temp + season*hum, data = day)
anova(redupday,upday.red)
```

```r
summary(redupday)

plot(redupday,which=1)
plot(redupday ,which=2)

new=data.frame(season="winter",workingday=median(day[day$season=="winter",]$workingday),
               temp=mean(day[day$season=="winter",]$temp),
               weathersit=mean(day[day$season=="winter",]$weathersit),
               hum=mean(day[day$season=="winter",]$hum),
               windspeed=mean(sqrt(day[day$season=="winter",]$windspeed)),
               holiday=median(day[day$season=="winter",]$holiday))
ci=predict(redupday,new=new,interval="confidence")
exp(ci)



new=data.frame(season="spring",workingday=median(day[day$season=="spring",]$workingday),
               temp=mean(day[day$season=="spring",]$temp),
               weathersit=mean(day[day$season=="spring",]$weathersit),
               hum=mean(day[day$season=="spring",]$hum),
               windspeed=mean(sqrt(day[day$season=="spring",]$windspeed)),
               holiday=median(day[day$season=="spring",]$holiday))
ci=predict(redupday,new=new,interval="confidence")
exp(ci)


new=data.frame(season="summer",workingday=median(day[day$season=="summer",]$workingday),
               temp=mean(day[day$season=="summer",]$temp),
               weathersit=mean(day[day$season=="summer",]$weathersit),
               hum=mean(day[day$season=="summer",]$hum),
               windspeed=mean(sqrt(day[day$season=="summer",]$windspeed)),
               holiday=median(day[day$season=="summer",]$holiday))
ci=predict(redupday,new=new,interval="confidence")
exp(ci)



new=data.frame(season="fall",workingday=median(day[day$season=="fall",]$workingday),
               temp=mean(day[day$season=="fall",]$temp),
               weathersit=mean(day[day$season=="fall",]$weathersit),
               hum=mean(day[day$season=="fall",]$hum),
               windspeed=mean(sqrt(day[day$season=="fall",]$windspeed)),
               holiday=median(day[day$season=="fall",]$holiday))
ci=predict(redupday,new=new,interval="confidence")
exp(ci)

redupday$coefficients

exp(confint(redupday))
exp(redupday$coefficients)
```