Charles Gauthey

Final DA: Bike Sharing in D.C./Arlington Metro Area

Introduction:

Bike sharing systems are gaining prominence in a current environment where sharing systems are more associated with Uber or Airbnb. Still, bike sharing is becoming ever more important in our transportation system. But, given its recent development, there is a need to analyze the factors into what affects bike sharing user numbers. We are tasked in determining if there is a multivariate regression model that can predict the number of registered bike users in the D.C./Arlington area, given variables pertaining to the day, type of rider, or weather. In particular, we would like to know if casual riders are negatively associated with registered bike riders. Furthermore, we would also like to see if weather and holidays have some sort of interaction as well as hour of day and workdays. Lastly, we are interested to see if feeling temperature is enough to explain weather's effect on registered users, rather than actual temperature, wind speed, and humidity

Exploratory Data Analysis (EDA):

We look at 723 hourly observations of bike sharing statistics in the Washington D.C./Arlington, VA/MD area. Each hourly observation provides us information on the following variables: **Registered** (number of registered bike users), **Date, Year**, **Month**, **Day** (of the week), **Hour** (of day), **Holiday**, **WorkDay**, **Weather**, **Temp** (temperature), **TempFeel** ("feels like" temperature), **Humidity**, **Windspeed**, **Casual**. Our numerical analysis is summarized Figure 1 and 2 and graphically summarized in Figure 3, except for **Date**. We have decided to exclude **Date** from analysis, because the information is inherently explained already through the variables **Year**, **Month**, and **Day**. Looking at are user distribution, both **Registered** and **Casual**, there is a greater range, median, mean, and standard deviation of **Registered** bike sharers compared to **Casual**. This makes sense since **Registered** users are a more frequent target and more analyzed aspect of the bike sharing program. Both are also skewed right. **Year**, **Month**, and **Day** all have nearly even distribution, although we do see lower frequency for earlier months. Comparing **Temp** and **TempFeel**, it is interesting to note that although **Temp** has the smaller range, it does have the higher standard deviation compared to **TempFeel**. Otherwise, **Temp**'s mean and median are both 0.02 higher than **TempFeel**. Looking at the distribution, **Temp** has a symmetrical bimodal bell curve. **TempFeel** is also bimodal, but is not symmetrical and looking more skewed left, with possible outliers at higher values. Other continuous predictor variables have **Humidity** skewed left and **Windspeed** skewed right, with both being unimodal. For our other categorical predictor variables (**Weather**, **Holiday**, **WorkDay**), each variable has a category that has a higher distribution of observations. For **Weather**, that means clear weather (1). **Holiday**'s is non-holidays (0) and **WorkDay**'s are non-weekends or holidays (1).

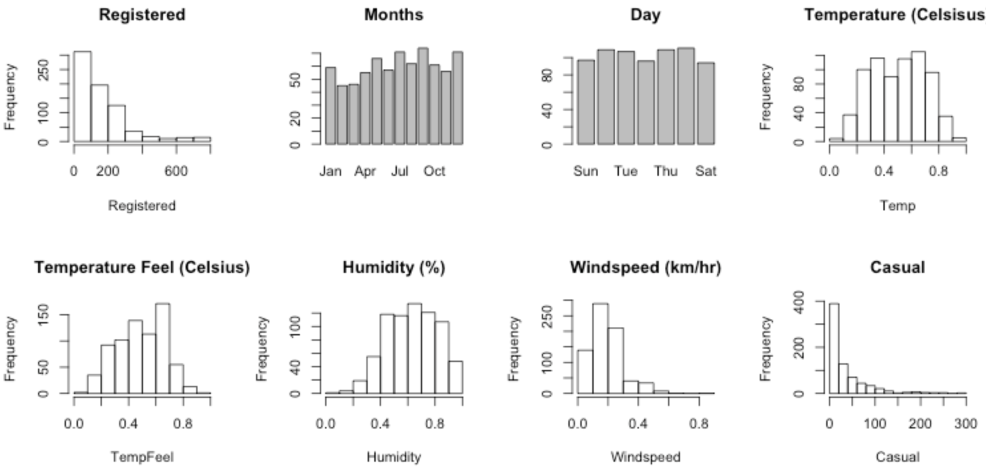
Figure 1: Univariate EDA - Numerical Summaries

Variable	Min.	1st. Qu.	Median	Mean	3rd. Qu.	Max	SD
Registered	1.00	40.00	120.00	159.20	221.00	790.00	160.06
Month	1.00	4.00	7.00	6.80	10.00	12.00	3.40
Day	0.00	1.00	3.00	3.00	5.00	6.00	1.97
Hour	0.00	6.00	11.00	11.60	17.50	23.00	6.84
Temp	0.04	0.34	0.52	0.51	0.66	0.96	0.20
TempFeel	0.03	0.33	0.50	0.49	0.62	0.98	0.18
Humidity	0.00	0.50	0.65	0.64	0.78	1.00	0.18
Windspeed	0.00	0.10	0.19	0.19	0.28	0.85	0.12
Casual	0.00	4.00	18.00	34.52	47.00	291.00	45.69

Figure 2: Categorical Univariate EDA - Numerical Summaries

Variable	Year		Weather			
Factor	**2011**	**2012**	**1**	**2**	**3**	**4**
Count	385	338	478	180	65	0
Percent	53%	47%	66%	25%	9%	0%
Variable	**Holiday**		**WorkDay**			
Factor	**0**	**1**	**0**	**1**		
Count	696	27	218	505		
Percent	96%	4%	30%	70%		

Figure 3: Univariate EDA - Graphical Summaries



We also look at our bivariate relationships in Figure 4, 5, and 6. It is interesting to note that the predictor variables in Figure 4 all demonstrate significant correlation with **Registered**, excep for Day. However, many of these relationships do not have a linear pattern. **Months** and **Day** are a bell curve, while **Hour** has a bimodal relationship. **Registered** and **Casual** also has a positive correlation, despite ideas from people that it might be the opposite. However, **Casual** does not have a linear relationship either, so a transformation we will discuss later will provide more insight. There are also several instances of noticeable evidence of multicollinearity. The most obvious example is **Temp** and **TempFeel** at 0.99, since it makes sense our perception of temperature is determined by actual temperature. **Month** also demonstrates a unique pattern with **Temp** and **TempFeel**, which is because seasons causes certain months to have higher or lower average temperatures.

We also explored several predictor variables conditionally in Figure 5 and 6. For **Year**, **Holiday**, **WorkDay**, and **Weather,** all exhibited difference in average registration within each of its categories. Distribution wise, we do notice that the categories with higher number of observations do tend to have a higher average, higher standard deviations, and more outliers. Because of the information, we cannot prove that each category within the variables are significantly different from one another.

Figure 4: Multivariate EDA – Pairs Plot

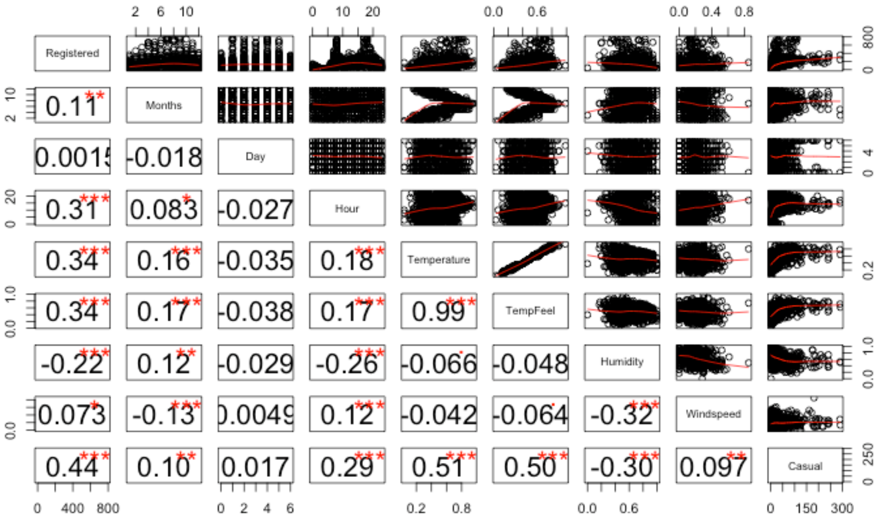
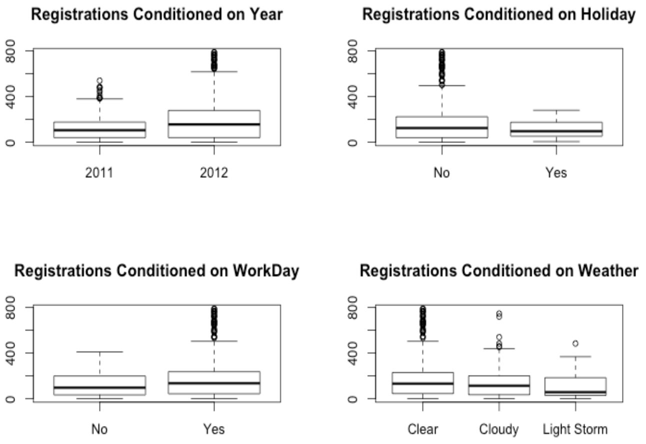


Figure 5: Conditional Distributions

	Mean	**SD**
Year		
2011	120.03	100.68
2012	203.75	198.98
Holiday		
No	161.03	162.09
Yes	111.26	81.72
WorkDay		
No	118.31	95.51
Yes	176.81	178.15
Weather		
1	172.44	173.65
2	141.33	130.71
3	110.98	109.19

Figure 6: Conditional Distributions - Graphical



Modeling the Data & Diagnostics:

We begin creating the initial model by determining whether certain categorical variables be considered ordered categories. We first analyzed **Month** and left it unordered, but collapsed into seasons. During our EDA, we noted that it had a non-linear relationship when plotted with our response. To compensate for that problem, we could reorder our **Month**, but the ordering could be prone to arbitrary metrics. The variable **Month** itself does not have any value and shouldn’t be ordered. Furthermore, an ordered **Month** would have a coefficient of 3.74, but converting **Month** into factors would give us a much wider range of coefficients. For example, using January as a reference, all coefficients of **Month** would be greater than 3.74. Furthermore, we decided to stabilize the categorical variable by splitting 12 months into 4 seasons. We analyzed this by grouping each season by similar temperature and order. Winter months would have the lowest average temperature and summer months have the highest. The groupings were *Winter* (December, January, February), *Spring* (March, April, May), *Summer* (June, July, August), and *Fall* (September, October, November). Our new collapsed **Month** will reference to *Winter*.

Day encountered a similar problem to **Month**. But, as we tried to convert it into a factor, we realized **Day** itself conveys most of the same information as **WorkDay**. **WorkDay** itself is just a more stable version of **Day**, by referencing itself to Weekends. We believe **WorkDay** is sufficient in conveying the relationship among all days and decided to eliminate **Day** instead.

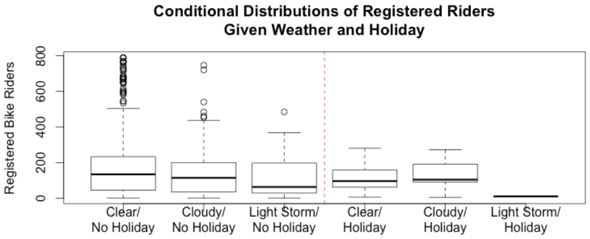
Hour was another variable with a similar issue to **Month**. Like **Month**, **Hour** had a significant correlation with **Registered**, but it was not a linear relationship. Furthermore, an hour itself does not possess a higher value than another hour. If ordered, **Hour** had a coefficient of 4.02. Referencing to midnight, the coefficients now ranged from -52.0422 to 302.9814, which would provide more insight in the relationship between each hour. The issue with 24 factors is that the sample size for each hour was very small. We decided to create a more stable **Hour** by collapsing it by time of the day: *OffHours.Night* (0-6 and 21-23 or 9 PM – 7 AM), *RushHour.Day* (7-9 or 7 AM – 10 AM), *OffHours.Day* (10-15 or 10 AM – 4 PM), *RushHour.Night* (16-20 or 4 PM – 9 PM). However, we kept it as 24 factors, since we found it difficult to collapse **Hour** based on time of day. We feel this categorization would best group hours with similar characteristics. Our **Hour** will have four factors with *OffHours.Night* as the reference group.

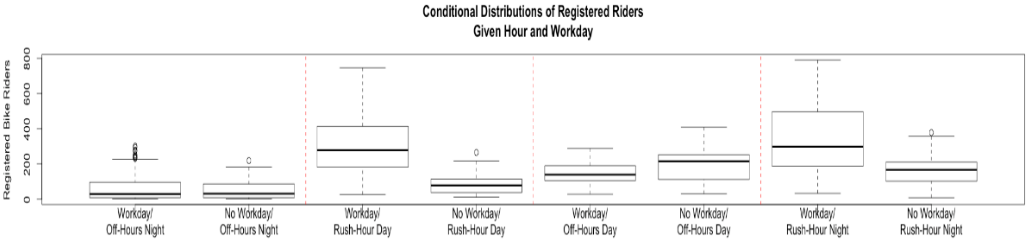
Weather is the only categorical variable that we decide to use as ordered. The way it is structured, each increase in **Weather** intensity (worsening weather) is a steady decrease in registered users. We can note this trend in Figure 6. Even as an ordered variable, we note that **Weather** does not drastically average out the effects of each category. The biggest difference is if **Weather** is at 3 (*Light Storm*) and referenced to 1 (*Clear*), the coefficient is -60.8184. This is slightly different from a total change of -48.3638 (given a coefficient of -24.1819) if **Weather** was ordered. This can be attributed to a smaller sample size for **Weather** at level 3.

Our next step in creating our initial model is to look at possible interaction terms. It has been suggested that the number of registered users and weather is dependent on whether it is a holiday. Furthermore, the relationship between registered users and hour of day could be effected on whether it is a work day. We analyze this relationship in Figure 7. For the interaction term **Weather*Holiday** on the top graph, we notice a difference in our median for each category of **Weather**, depending on whether it is a holiday or not. Furthermore, for **Hour*WorkDay**, we note that except for *OffHours.Night*, all hours have a significant different in median and IQR when comparing to between whether it's a workday or not. As both graphs suggest that the interactions cause some impact to the number of registered users, we decide to include the terms into our initial model.

Our initial model will predict **Registered** users with **Year**, **Months** (collapsed into seasons), **Hour**, **Holiday**, **WorkDay**, **Weather**, **Temp** (temperature), **TempFeel** (temperature feel), **Humidity**, **Windspeed**, **Casual** users. We will also analyze the model with the interaction between **Weather***Holiday** and **WorkDay*Hour** (collapsed into times of day).

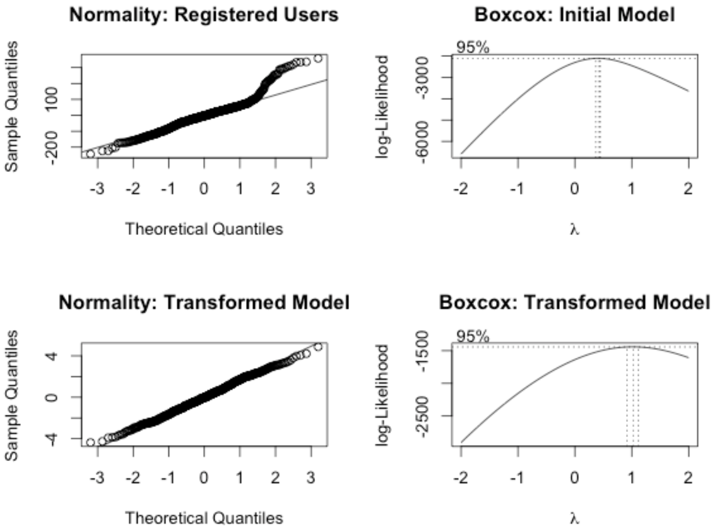
Figure 7: Interaction Terms





We now analyze the feasibility of our initial model. In our normality plot on Figure 8 (top left), we note that the distribution of our residuals demonstrates a skew on both sides, with a notable large skew on the right. We decide to transform **Registered** to the power of 0.4 due to our Box-Cox graph. Our transformed residuals demonstrate on improvement on both sides, but now with a slight truncation on both sides. We decide to check if outliers may affect the normality.

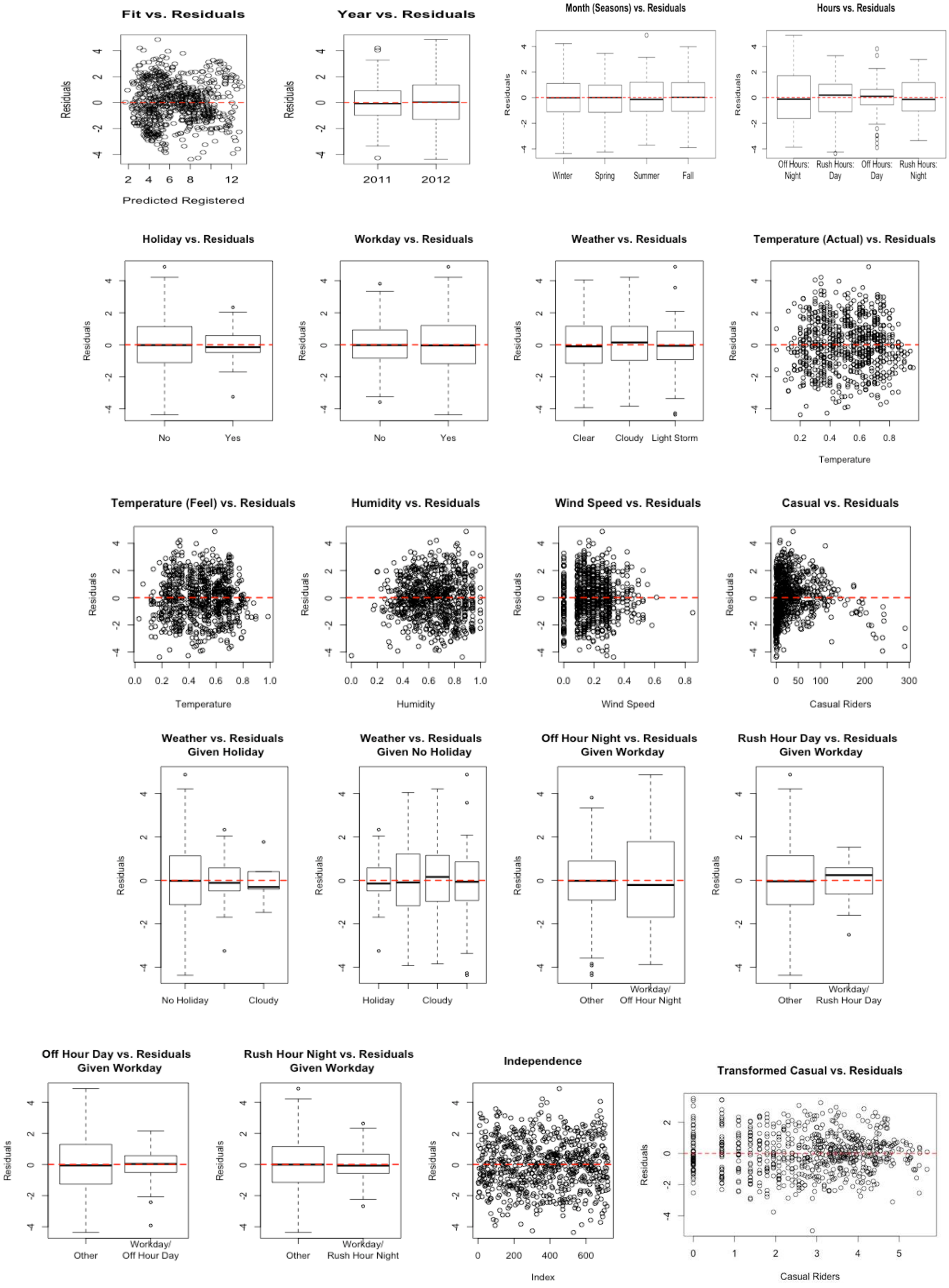
Figure 8: Normality and Boxcox Graphs



To identify outliers, we calculated the threshold for our standardized residual to be 4.003152 and found no outliers that exceeded the threshold. In addition, we decided to screen for observations that exceed twice the mean leverage. There were 36 identified outliers, but we noted six that were significantly different from the other leverage values. However, out of the six, we noted all six to be holidays and four of the observations to be on Labor Day. Due to their common traits, we decided to only throw out only one outlier out of the six, which itself had three times higher leverage than the next closest observation. We eliminated one outlier total, but this has no noticeable effect on our normality. Overall, we are pleased with the normality, as the truncation is small and understandable, because many of our predictor variables themselves are limited by a range of possible values.

Progressing further through our diagnostics, we look at our residual and independence plots in Figure 9. Our residual and independence plots look fine. Most of our residual plots demonstrating a constant variance and expected residual of zero. Our independence plot also demonstrates no group patterns. There are some issues, such as **Windspeed** having an outlier, but even that outlier has a residual near zero. The only residual plot of with significant concerns is **Casual**, with a significant group of higher casual riders having negative residuals. We also noted back in our EDA that **Casual** seems to have a non-linear relationship with **Registered**. We decide to transform **Casual to** log(**Casual**+1). The result is an improved residual plot for **Casual** in the bottom right of Figure 9, which we will include in our model.

Figure 9: Residual Plots and Independence



Before we determine our final model, we decide to check if we can remove any of our predictor variables. It has been noted that **TempFeel** may be more useful than **Temp**, **Windspeed**, and **Humidity**. We conducted a hypothesis test to determine whether **TempFeel**, **Temp**, **Windspeed**, or **Humidity** has a regression relationship with **Registered** users. Interestingly, we found no significant evidence at a p-value of 0.2457, which is much higher than our set alpha level of 0.05. Thus, we decided to remove not only **TempFeel** but also **Temp**, **Windspeed**, and **Humidity** from our model. A possible reason for this is that **Weather** itself may have already explained much of the information our excluded variables were supposed to analyze, just in a more condensed manner.

Results:

We conclude our final model will predict a transformed **Registered** using the variables **Year**, **Month** (in Seasons), **Hour** (in time of day), **Holiday**, **WorkDay**, **Weather**, transformed **Casual**, **Weather*Holiday**, and **WorkDay*Hour**. The results of our model are given in Figure 10. It is interesting to note that transformed **Casual** users has a significantly positive relationship with transformed **Registered** users. This is even though it was previously thought the relationship between the two should be negative. This could have happened because it was wrong to just look at the fact there are only two types of riders. Instead, we should have looked at how both types of riders are affected by the same type of variables. For example, a registered and casual bike rider might be less likely to ride if he or she experienced less than ideal weather.

Other significant non-interaction terms include **Year**, *Spring* and *Fall* months relative to *Winter* months, day off hours and night rush hours relative to night off hours, work days relative to non-work days, and the weather. For **Year,** compared to 2011, 2012 will have an additional 0.8621 transformed registered users. For **Month,** our seasons are interestingly inconsistent, with *Spring* and *Fall* having opposite effects despite having similar weather characteristics. For **Hours**, there is a significant impact to registered users relative to night off hours for all variables except daytime rush hour. This could be because our daytime rush hour is our smallest category under **Hours**. Excluding the effects of interaction, **WorkDay** relative to non-work days adds a transformed 0.819 registered users. Each degree of worsening weather decreased transformed registered users by 0.2149.

For our interaction terms, only **Hours*WorkDay** had categories with a significance evidence of a linear relationship with **Registered**. Compared to night off-hours on a non-workday, a morning rush hour on a workday predicts an additional 4.6138 transformed **Registered**. Day time off-hours on a workday will have an additional 0.9923 transformed **Registered**. Night rush hours on a workday will have an additional 3.9325 transformed **Registered**. It makes sense that rush hours on workdays will have more pronounced effects on **Registered** users, because commuting times are the times **Registered** users are most in need of bike sharing services

Our final regression model is significant at any reasonable alpha level, even our given level of 0.05. This model, given the adjusted R^2, can explain 81.91% of the variation found in the model. We have evidence that the variables **Year**, **Month** (in Seasons), **Hour** (in time of day), **Holiday**, **WorkDay**, **Weather**, transformed **Casual**, **Weather*Holiday**, and **WorkDay*Hour** has a significant relationship with a transformed **Registered** number of bike users.

Figure 10: Final Model Results

				Confidence Interval	
	Estimate	Std. Error	Pr(>|t|)	2.50%	97.50%
(Intercept)	1.3877	0.2150	2.01E-10	0.9656	1.8098
as.factor(Year2)2012	0.8621	0.1006	< 2e-16	0.6647	1.0595
Spring2	-0.3094	0.1528	0.043201	-0.6094	-0.0095
Summer2	-0.1655	0.1593	0.299234	-0.4782	0.1473
Fall2	0.3026	0.1508	0.045233	0.0064	0.5987
RushHour.Day2	-0.1596	0.2883	0.580141	-0.7257	0.4065
OffHours.Day2	0.6994	0.2606	7.46E-03	0.1877	1.2112
RushHour.Night2	1.0070	0.2592	0.000112	0.4982	1.5158
as.factor(Holiday2)1	-0.1591	0.8280	0.847731	-1.7847	1.4666
as.factor(WorkDay2)1	0.8190	0.1735	2.85E-06	0.4783	1.1597
Weather2	-0.2149	0.0782	0.006141	-0.3683	-0.0614
Casual3	1.3499	0.0544	< 2e-16	1.2431	1.4566
as.factor(Holiday2)1:Weather2	-0.2035	0.6542	7.56E-01	-1.4878	1.0808
RushHour.Day2:as.factor(WorkDay2)1	3.9544	0.3304	< 2e-16	3.3057	4.6030
OffHours.Day2:as.factor(WorkDay2)1	-0.5261	0.2692	0.051082	-1.0546	0.0025
RushHour.Night2:as.factor(WorkDay2)1	2.1065	0.2884	7.60E-13	1.5402	2.6728

Residual standard error: 1.297 on 706 degrees of freedom					
Multiple R-squared: 0.8229,	Adjusted R-squared:		0.8191		
F-statistic: 218.7 on 15 and 706 DF, p-value: < 2.2e-16					

Conclusion/Discussion:

Important variables used to predict registered bike users include year, months, hour of day, whether the day is a holiday or a workday, weather, and number of casual riders. We can use these predictor variables to create a multivariate linear regression model with registered users. Other than the number of casual riders, all other variables are dependent on weather or time. There are a couple of curious results. For one, Spring and Fall months seem to have an opposite effect on transformed registered users, despite having similar temperature qualities. In fact, Summer has a negative relationship with transformed registered users relative to Winter months. This seems counterintuitive since people should prefer to bike in better weather months. A possible rationale for this is that workers may be on vacation more often in Summer and Spring. If so, we may consider this further by looking at percentage of registered riders riding instead as a more accurate measure. Understandably, however, the goal of the study is to maximize the number of bike riders, not necessarily the percent usage. Furthermore, the relationship between casual and registered users contradicted our initial hypothesis. As noted in our results, both riders base their decisions by the same type of variables. Instead of analyzing the two variables as competing entities, in the future we might consider combining the two variables by choosing our response variable as total bike users.

R Code:

~~~
---
title: "Final DA Exam — 36-401"
author: "Charles Gauthey"
date: "May 7, 2017"
output: html_document
---

Load
```{r}
bikes = read.table("D:/Documents/CMU Courses/CMU Statistics Courses/Best 36-401/FinalDA3/final-
71.txt")
attach(bikes)
source("D:/Documents/CMU Courses/CMU Statistics Courses/Best 36-
401/FinalDA3/panelfxns(1).R")library(plyr)
library(MASS)
```
Univariate EDA
```{r}
bikes.v2 = apply(bikes, 2, as.numeric)
apply(bikes.v2, 2, summary)
apply(bikes.v2, 2, sd)
apply(bikes, 2, table)

nrow(bikes)

par(mfrow = c(2,4))
hist(Registered, main = "Registered")
barplot(table(Month),names.arg=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"
,"Dec"), main = "Months")
barplot(table(Day),names.arg=c("Sun","Mon","Tue","Wed","Thu","Fri","Sat"), main = "Day")
hist(Temp, main = "Temperature (Celsisus)")
hist(TempFeel, main = "Temperature Feel (Celsius)")
hist(Humidity, main = "Humidity (%)")
hist(Windspeed, main = "Windspeed (km/hr)")
hist(Casual, main = "Casual")

```
Multivariate EDA
```{r}
#Continous EDA
par(mfrow = c(1,1))
cont.vars<-cbind(Registered,Month,Day,Hour, Temp,TempFeel,Humidity, Windspeed,Casual)
colnames(cont.vars)<-
c("Registered","Months","Day","Hour","Temperature","TempFeel","Humidity","Windspeed","Casual")
pairs(cont.vars,upper.panel=panel.smooth,lower.panel=panel.cor)

#Categorical EDA
mean(Registered[Year == 2011]);sd(Registered[Year == 2011])
mean(Registered[Year == 2012]);sd(Registered[Year == 2012])
~~~

~~~~
mean(Registered[Holiday == 0]);sd(Registered[Holiday == 0])
mean(Registered[Holiday == 1]);sd(Registered[Holiday == 1])
mean(Registered[WorkDay == 0]);sd(Registered[WorkDay == 0])
mean(Registered[WorkDay == 1]);sd(Registered[WorkDay== 1])
mean(Registered[Weather == 1]);sd(Registered[Weather== 1])
mean(Registered[Weather == 2]);sd(Registered[Weather == 2])
mean(Registered[Weather == 3]);sd(Registered[Weather == 3])

par(mfrow = c(2,2))
boxplot(Registered~Year,names=c(2011,2012),main = "Registrations Conditioned on Year")
boxplot(Registered~Holiday,names=c("No","Yes"),main = "Registrations Conditioned on Holiday")
boxplot(Registered~WorkDay,names=c("No","Yes"),main = "Registrations Conditioned on WorkDay")
boxplot(Registered~Weather,names=c("Clear","Cloudy","Light Storm"),main = "Registrations
Conditioned on Weather")

```
Initial Modeling
```{r}
#Model 1 Basic: Official Categoricals
mod1.1 =
lm(Registered~Year+Month+Day+Hour+Holiday+WorkDay+Weather+Temp+TempFeel+Humidity+Wi
ndspeed+Casual)

summary(mod1.1)

mod1.2 =
lm(Registered~as.factor(Year)+as.factor(Month)+Day+Hour+Holiday+WorkDay+Weather+Temp+Temp
Feel+Humidity+Windspeed+Casual)
summary(mod1.2)


Winter = ifelse(Month == 1 | Month == 2 | Month == 12, 1, 0)
Spring = ifelse(Month == 3 | Month == 4 | Month == 5, 1, 0)
Summer = ifelse(Month == 6 | Month == 7 | Month == 8, 1, 0)
Fall = ifelse(Month == 9 | Month == 10 | Month == 11, 1, 0)

#Analyzing splitting Months into seasons
plot(Month, Registered)
boxplot(Registered~Winter)
boxplot(Registered~Spring)
boxplot(Registered~Summer)
boxplot(Registered~Fall)
c(mean(Temp[Month==1]),mean(Temp[Month==2]),mean(Temp[Month==3]),mean(Temp[Month==4]),
mean(Temp[Month==5]),mean(Temp[Month==6]),mean(Temp[Month==7]),mean(Temp[Month==8]),m
ean(Temp[Month==9]),mean(Temp[Month==10]),mean(Temp[Month==11]),mean(Temp[Month==12])
)
order(c(mean(Temp[Month==1]),mean(Temp[Month==2]),mean(Temp[Month==3]),mean(Temp[Month
==4]),mean(Temp[Month==5]),mean(Temp[Month==6]),mean(Temp[Month==7]),mean(Temp[Month=
=8]),mean(Temp[Month==9]),mean(Temp[Month==10]),mean(Temp[Month==11]),mean(Temp[Month=
=12]) ))
~~~~

~~~
mod1.3 = lm(Registered~as.factor(Year)+Spring+Summer+Fall+Day+Hour+Holiday+WorkDay+Weather+Temp+TempFeel+Humidity+Windspeed+Casual+Weather*Holiday+WorkDay*Hour)
summary(mod1.3)

'
month.colder = ifelse(Month == 1 | Month == 2 | Month == 3 | Month == 4 | Month == 11| Month == 12, 1, 0)
month.warmer = ifelse(Month == 5 | Month == 6 | Month == 7 | Month == 8 | Month == 9| Month == 10, 1, 0)
mod1.4  = lm(Registered~as.factor(Year)+month.warmer+Day+Hour+Holiday+WorkDay+Weather+Temp+TempFeel+Humidity+Windspeed+Casual+Weather*Holiday+WorkDay*Hour)
summary(mod1.4)'

'mod1.5 = lm(Registered~as.factor(Year)+as.factor(Month)+as.factor(Day)+Hour+Holiday+WorkDay+Weather+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.5)
mod1.6 = lm(Registered~as.factor(Year)+as.factor(Month)+as.factor(Day)+Hour+Holiday+Weather+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.6)'
mod1.7 = lm(Registered~as.factor(Year)+Spring+Summer+Fall+Hour+Holiday+as.factor(WorkDay)+Weather+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.7)

OffHours.Night = ifelse(Hour == 0 | Hour == 1  | Hour == 2 |Hour == 3 | Hour == 4 |Hour == 5|Hour == 6|Hour == 21|Hour == 22|Hour == 23, 1, 0)
RushHour.Day = ifelse(Hour == 7 | Hour == 8  | Hour == 9 , 1, 0)
OffHours.Day = ifelse(Hour == 10 | Hour == 11 | Hour == 12 |Hour == 13 | Hour == 14 |Hour == 15, 1, 0)
RushHour.Night = ifelse(Hour == 16 | Hour == 17  | Hour == 18|Hour == 19  | Hour == 20 , 1, 0)

'mod1.8 = lm(Registered~as.factor(Year)+Spring+Summer+Fall+as.factor(Hour)+as.factor(Holiday)+as.factor(WorkDay)+Weather+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.8)'

mod1.9 = lm(Registered~as.factor(Year)+Spring+Summer+Fall+RushHour.Day+OffHours.Day+RushHour.Night+as.factor(Holiday)+as.factor(WorkDay)+Weather+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.9)

'mod1.10 = lm(Registered~as.factor(Year)+as.factor(Month)+as.factor(Hour)+as.factor(Holiday)+as.factor(WorkDay)+as.factor(Weather)+Temp+TempFeel+Humidity+Windspeed+Casual)
summary(mod1.10)'
```
~~~

~~~
```{r}
#Model 2 Basic: Interactions
par(mfrow = c(1,1))

weather.holiday = rep(NA,nrow(bikes))
weather.holiday[Weather == 1 & Holiday == 0]<-1
weather.holiday[Weather == 2 & Holiday == 0]<-2
weather.holiday[Weather == 3 & Holiday == 0]<-3
weather.holiday[Weather == 1 & Holiday == 1]<-4
weather.holiday[Weather == 2 & Holiday == 1]<-5
weather.holiday[Weather == 3 & Holiday == 1]<-6
boxplot(Registered~weather.holiday,ylab="Registered Bike Riders",names=c("Clear/\nNo
Holiday","Cloudy/\nNo Holiday","Light Storm/\nNo
Holiday","Clear/\nHoliday","Cloudy/\nHoliday","Light Storm/\nHoliday"))
abline(v=3.5,lty=2,col=2)
title("Conditional Distributions of Registered Riders\nGiven Weather and Holiday")

hour.workday = rep(NA,nrow(bikes))
hour.workday[WorkDay == 1 & OffHours.Night == 1]<-1
hour.workday[WorkDay == 0 & OffHours.Night == 1]<-2
hour.workday[WorkDay == 1 & RushHour.Day == 1]<-3
hour.workday[WorkDay == 0 & RushHour.Day == 1]<-4
hour.workday[WorkDay == 1 & OffHours.Day == 1]<-5
hour.workday[WorkDay == 0 & OffHours.Day == 1]<-6
hour.workday[WorkDay == 1 & RushHour.Night == 1]<-7
hour.workday[WorkDay == 0 & RushHour.Night == 1]<-8
boxplot(Registered~hour.workday,ylab="Registered Bike Riders",names=c("Workday/\nOff-Hours
Night","No Workday/\nOff-Hours Night","Workday/\nRush-Hour Day","No Workday/\nRush-Hour
Day",
"Workday/\nOff-Hours Day","No Workday/\nOff-Hours Day","Workday/\nRush-Hour Night","No
Workday/\nRush-Hour Night"))
abline(v=2.5,lty=2,col=2)
abline(v=4.5,lty=2,col=2)
abline(v=6.5,lty=2,col=2)
title("Conditional Distributions of Registered Riders\nGiven Hour and Workday")

'hour.2 = Hour+1
plot(Hour,Registered,col = (WorkDay+1),xlab="Hours of the Day",ylab="Registered Bike Riders")
abline(lm(Registered[WorkDay == 0]~Hour[WorkDay == 0]),col=1,lwd=2)
abline(lm(Registered[WorkDay == 1]~Hour[WorkDay == 1]),col=2,lwd=2)

title("Hour vs. Registered Bike Rides \n Conditioned on Workday")
legend("topleft",c("Weekend","Workday"),col=c(1,2),lwd=2,pch=16)'

mod2.1 =
lm(Registered~as.factor(Year)+Spring+Summer+Fall+RushHour.Day+OffHours.Day+RushHour.Night+
as.factor(Holiday)+as.factor(WorkDay)+Weather+Temp+TempFeel+Humidity+Windspeed+Casual+Wea
ther*as.factor(Holiday)+as.factor(WorkDay)*RushHour.Day+as.factor(WorkDay)*OffHours.Day+as.fact
or(WorkDay)*RushHour.Night)
summary(mod2.1)
~~~

~~~~
```
Diagnostics: Normality/BoxCox
```{r}
#Model 3: Normality/Box Cox
par(mfrow=c(2,2))
qqnorm(mod2.1$res,main="Normality: Registered Users")
qqline(mod2.1$res)
boxcox1 = boxcox(mod2.1)
title("Boxcox: Initial Model")

registered2 = Registered^(.4)
mod3.1 =
lm(registered2~as.factor(Year)+Spring+Summer+Fall+RushHour.Day+OffHours.Day+RushHour.Night+
as.factor(Holiday)+as.factor(WorkDay)+Weather+Temp+TempFeel+Humidity+Windspeed+Casual+Wea
ther*as.factor(Holiday)+as.factor(WorkDay)*RushHour.Day+as.factor(WorkDay)*OffHours.Day+as.fact
or(WorkDay)*RushHour.Night)
qqnorm(mod3.1$res,main="Normality: Transformed Model")
qqline(mod3.1$res)
boxcox2 = boxcox(mod3.1)
title("Boxcox: Transformed Model")

summary(mod3.1)
```
Essay Notes:
Describe prenormality graphs. Boxcox transformation to (.2) causes better treatment of outliers. Very
skewed left however.

Outlier Analysis
```{r}
#Model 4: Outlier analysis
X<-cbind(1,as.factor(Year),Spring, Summer,
Fall,RushHour.Day,OffHours.Day,RushHour.Night,as.factor(Holiday),as.factor(WorkDay),Weather,Tem
p,TempFeel,Humidity,Windspeed,Casual,Weather*ifelse(Holiday == 1,1,0),ifelse(WorkDay ==
1,1,0)*RushHour.Day,ifelse(WorkDay == 1,1,0)*OffHours.Day,ifelse(WorkDay ==
1,1,0)*RushHour.Night)

H<-X%*%solve(t(X)%*%X)%*%t(X)
n<-nrow(X);p<-ncol(X)
SSE<-sum(mod3.1$res^2)
MSE<-SSE/(n-p)
res<-mod3.1$res
del.res<-res*sqrt((n-p-1)/(SSE*(1-diag(H))-res^2))
alpha<-0.05
qt(1-alpha/(2*n),n-p-1)
sort(del.res)[1:10]; sort(del.res)[(n-10):n]

mean.h<-p/n
which(diag(H)>2*mean.h)
sort(diag(H))[(n-10):n]
order(diag(H))[(n-10):n]
~~~~

~~~
#outliers = c(order(diag(H))[(n-5):n], 302, 531)
outliers = 436
registered3 = registered2[-outliers]
Year2 = Year[-outliers]
Spring2 = Spring[-outliers]
Summer2 = Summer[-outliers]
Fall2 = Fall[-outliers]
Winter2 = Winter[-outliers]
Hour2 = Hour[-outliers]
Holiday2 = Holiday[-outliers]
WorkDay2 = WorkDay[-outliers]
Weather2 = Weather[-outliers]
Temp2 = Temp[-outliers]
TempFeel2 = TempFeel[-outliers]
Humidity2 = Humidity[-outliers]
Windspeed2 = Windspeed[-outliers]
Casual2 = Casual[-outliers]
OffHours.Night2 = OffHours.Night[-outliers]
RushHour.Day2 = RushHour.Day[-outliers]
OffHours.Day2 = OffHours.Day[-outliers]
RushHour.Night2 = RushHour.Night[-outliers]


mod4.1 = lm(registered3~as.factor(Year2)+Spring2+Summer2+Fall2+RushHour.Day2+OffHours.Day2+RushHour.Night2+as.factor(Holiday2)+as.factor(WorkDay2)+Weather2+Temp2+TempFeel2+Humidity2+Windspeed2+Casual2+Weather2*as.factor(Holiday2)+as.factor(WorkDay2)*RushHour.Day2+as.factor(WorkDay2)*OffHours.Day2+as.factor(WorkDay2)*RushHour.Night2)
qqnorm(mod4.1$res,main="Normality: Transformed/No Outlier Model")
qqline(mod4.1$res)
boxcox3 = boxcox(mod4.1)
title("Boxcox: Transformed/No Outlier Model")

summary(mod4.1)

```
~~~

Further Diagnostics: Residuals

~~~
```{r}
par(mfrow = c(1,2))
plot(mod4.1$fitted.values, mod4.1$residuals,main = "Fit vs. Residuals", xlab= "Predicted Registered", ylab= "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

boxplot(mod4.1$res~Year2,names=c(2011,2012),ylab="Residuals", main = "Year vs. Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

boxplot(mod4.1$res[Winter2==1],mod4.1$res[Spring2==1],mod4.1$res[Summer2==1],mod4.1$res[Fall2==1],names=c("Winter ","Spring","Summer","Fall"),ylab="Residuals");
title("Month (Seasons) vs. Residuals")
abline(h=0,lty=2, lwd= 2, col = "red")
~~~

```
boxplot(mod4.1$res[OffHours.Night2==1],mod4.1$res[RushHour.Day2==1],mod4.1$res[OffHours.Day2==1],mod4.1$res[RushHour.Night2==1],names=c("Off Hours:\nNight ","Rush Hours:\nDay","Off Hours:\nDay","Rush Hours:\nNight"),ylab="Residuals");
title("Hours vs. Residuals")
abline(h=0,lty=2, lwd= 2, col = "red")

par(mfrow = c(1,4))
boxplot(mod4.1$res~Holiday2,names=c("No","Yes"),ylab="Residuals", main = "Holiday vs. Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

boxplot(mod4.1$res~WorkDay2,names=c("No","Yes"),ylab="Residuals", main = "Workday vs. Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

boxplot(mod4.1$res~Weather2,names=c("Clear","Cloudy","Light Storm"),ylab="Residuals", main = "Weather vs. Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

plot(Temp2,mod4.1$residuals, main = "Temperature (Actual) vs. Residuals", xlab = "Temperature", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

plot(TempFeel2,mod4.1$residuals, main = "Temperature (Feel) vs. Residuals", xlab = "Temperature", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

plot(Humidity2,mod4.1$residuals, main = "Humidity vs. Residuals", xlab = "Humidity", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

plot(Windspeed2,mod4.1$residuals, main = "Wind Speed vs. Residuals", xlab = "Wind Speed", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

plot(Casual2,mod4.1$residuals, main = "Casual vs. Residuals", xlab = "Casual Riders", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

weather.holidayY = Weather2*ifelse(Holiday2 == 1,1,0)
weather.holidayN = Weather2*ifelse(Holiday2 == 0,1,0)
boxplot(mod4.1$residuals~weather.holidayY, main = "Weather vs. Residuals \n Given Holiday", ylab= "Residuals", names = c("No Holiday", "Clear","Cloudy"))
abline(h=0,lty=2, col = "red", lwd = 2)
boxplot(mod4.1$residuals~weather.holidayN, main = "Weather vs. Residuals \n Given No Holiday", ylab= "Residuals",names = c("Holiday", "Clear","Cloudy", "Light Storm"))
abline(h=0,lty=2, col = "red", lwd = 2)

OffHourNight.workday = ifelse(WorkDay2 == 1,1,0)*OffHours.Night2
RushHourDay.workday = ifelse(WorkDay2 == 0,1,0)*RushHour.Day2
OffHourDay.workday = ifelse(WorkDay2 == 1,1,0)*OffHours.Day2
```

~~~~
RushHourNight.workday = ifelse(WorkDay2 == 0,1,0)*RushHour.Night2

boxplot(mod4.1$residuals~OffHourNight.workday, main = "Off Hour Night vs. Residuals \n Given Workday", ylab= "Residuals", names = c("Other", "Workday/\nOff Hour Night"))
abline(h=0,lty=2, col = "red", lwd = 2)
boxplot(mod4.1$residuals~RushHourDay.workday, main = "Rush Hour Day vs. Residuals \n Given Workday", ylab= "Residuals", names = c("Other", "Workday/\nRush Hour Day"))
abline(h=0,lty=2, col = "red", lwd = 2)

par(mfrow = c(1,3))
boxplot(mod4.1$residuals~OffHourDay.workday, main = "Off Hour Day vs. Residuals \n Given Workday", ylab= "Residuals", names = c("Other", "Workday/\nOff Hour Day"))
abline(h=0,lty=2, col = "red", lwd = 2)
boxplot(mod4.1$residuals~RushHourNight.workday, main = "Rush Hour Night vs. Residuals \n Given Workday", ylab= "Residuals", names = c("Other", "Workday/\nRush Hour Night"))
abline(h=0,lty=2, col = "red", lwd = 2)
```
Further Diagnostics: Independence
```{r}

plot(mod4.1$residuals,xlab= "Index",ylab="Residuals", main="Independence")
abline(h=0,lty=2,lwd=2, col= "red")
```
Final Model Analysis:
```{r}
#Model 5: Variable Transformation
par(mfrow = c(1,1))
Casual3 = log(Casual2+1)
mod5.1 = lm(registered3~as.factor(Year2)+Spring2+Summer2+Fall2+RushHour.Day2+OffHours.Day2+RushHour.Night2+as.factor(Holiday2)+as.factor(WorkDay2)+Weather2+Temp2+TempFeel2+Humidity2+Windspeed2+Casual3+Weather2*as.factor(Holiday2)+as.factor(WorkDay2)*RushHour.Day2+as.factor(WorkDay2)*OffHours.Day2+as.factor(WorkDay2)*RushHour.Night2)
summary(mod5.1)
plot(Casual3,mod5.1$residuals, main = "Transformed Casual vs. Residuals", xlab = "Casual Riders", ylab = "Residuals")
abline(h=0,lty=2, col = "red", lwd = 2)

'plot(Windspeed2, mod5.1$residuals)
Windspeed3 = log(Windspeed2+1)
plot(Windspeed3, mod5.1$residuals)

mod5.2 = lm(registered3~as.factor(Year2)+Spring2+Summer2+Fall2+as.factor(Hour2)+as.factor(Holiday2)+as.factor(WorkDay2)+Weather2+Temp2+TempFeel2+Humidity2+Windspeed3+Casual3+Weather2*as.factor(Holiday2)+as.factor(WorkDay2)*as.factor(Hour2))
summary(mod5.2)'

#Model 6: Variable Removal
mod6.1 = lm(registered3~as.factor(Year2)+Spring2+Summer2+Fall2+RushHour.Day2+OffHours.Day2+RushHour.
~~~~

~~~
Night2+as.factor(Holiday2)+as.factor(WorkDay2)+Weather2+Casual3+Weather2*as.factor(Holiday2)+a
s.factor(WorkDay2)*RushHour.Day2+as.factor(WorkDay2)*OffHours.Day2+as.factor(WorkDay2)*Rus
hHour.Night2+Temp2+TempFeel2+Humidity2+Windspeed2)
aov(mod6.1)
summary(mod6.1)
f1 = ((7.9414+0.9219+0.0497+0.2999)/4)/(1187.2488/702)
1-pf(f1,4,652)
#r2.ssr = (0.05661)/(0.05661+0.04569+0.00024+0.00148+17.32803)

#summary(lm(registered3~as.factor(Year2)+as.factor(Month2)+as.factor(Hour2)+as.factor(Holiday2)+as.
factor(WorkDay2)+Weather2+Casual3+Weather2*as.factor(Holiday2)+as.factor(WorkDay2)*as.factor(H
our2)+TempFeel2))

#mod6.2 =
lm(registered3~as.factor(Year2)+as.factor(Month2)+as.factor(Hour2)+as.factor(WorkDay2)+Casual3+as.
factor(WorkDay2)*as.factor(Hour2)+as.factor(Holiday2)+Weather2+Weather2*as.factor(Holiday2))

mod6.2 =
lm(registered3~as.factor(Year2)+Spring2+Summer2+Fall2+RushHour.Day2+OffHours.Day2+RushHour.
Night2+as.factor(Holiday2)+as.factor(WorkDay2)+Weather2+Casual3+Weather2*as.factor(Holiday2)+a
s.factor(WorkDay2)*RushHour.Day2+as.factor(WorkDay2)*OffHours.Day2+as.factor(WorkDay2)*Rus
hHour.Night2)
summary(mod6.2)
aov(mod6.2)

#mod6.3 =
lm(registered3~as.factor(Year2)+as.factor(Month2)+as.factor(Hour2)+as.factor(Holiday2)+as.factor(Wor
kDay2)+Weather2+Casual3+as.factor(WorkDay2)*as.factor(Hour2))
#summary(mod6.3)

```
~~~

Final Model Analysis: CI

~~~
```{r}
#alpha level 0.05 throuhgout analysis.
summary(mod6.2)
confint(mod6.2)
```
~~~