Using Weighted Least Squares Method to Predict Bike Rental Frequency in Washington, D.C.

Shaoxuan Chen, Samuel Greeman, Adrian Kelley, Hanyi Zhang

Abstract

Weighted least squares regression models are developed for predicting the number of bike rentals for both registered users and casual users in Washington D.C. [1] Models are trained on bike rental data from 2011 and validated using 2012 data. Output from R yields highly significant results for both models. Covariates such as season, adjusted temperature and date are selected based on the box plot, scatter plot and correlation matrices, as well as the output from running stepwise variable selections based on Bayesian Information Criterion. Standardized residual plots show a clear pattern, suggesting the models may not be optimal for prediction. In prediction, the final WLS model for registered count performs slightly better than the final WLS model for casual count (RMSE = 0.108 vs. RMSE = 0.257, respectively). Both models consistently underpredict bike rental counts in 2012. There is likely some variables not accounted for in the dataset causing bike rentals to rise from one year to the next. Based on these results, suggestions for bike-sharing businesses to increase profits are discussed in the final section.

1 Introduction

The advancement of technology has led to increased automation in services and industries that traditionally lacked it, as well as more convenience for the consumer. In urban areas, this automation has given the public both easier access to transportation and a greater diversity of options from which to choose. Recently, bike-sharing systems have sprouted throughout the world, with more than 500 bike-sharing programs being utilized[1]. City residents, visitors and tourists can now rent out a bicycle at a location and return it at any other bike-rental stations. As biking becomes a more prevalent mode of travel and its positive effects on the environment and physical health known, it is key to gain a better understanding of what drives people to rent them. Through the analysis, we hope to gain insights on this phenomenon and perpetuate the growth of bike travel.

2 Background

The dataset of this project contains daily bike rental data for Washington D.C. in 2011 and 2012, which is collected by the Capital Bikeshare system. Shown below in Table 1 are the variables from the data we considered for analysis. Additionally, from Fanaee-T and Gama, there is reason to believe that some major events taking place over these two years may have significantly impacted the bike rental counts on days in which the event took place (such as Hurricane Sandy in October 2012)[1]. To reflect this in the analysis, all the data on days with extreme weather in 2012 were filtered out for the validation data set. This is due to the fact that extreme weather is an anomalous event.

The primary objectives for this paper are to create a valid and useful model to predict the number of bikes that both registered users as well as casual users rent out in a given day. This modeling process is helpful in discovering which factors influence the public's transportation choices. If such a model is achieved, the results can be used to provide suggestions on how to increase bike-rental system usage in cities.

[1] Data processing: S.C; Writing and modeling: A.K & S.C & H.Z& S.G; Result analysis: S.C, A.K, H.Z; Organization of results: S.G.

Table 1: Variables : A description of the variables included in the data set

Categorical Variables	**Descriptions**
season	season (1 = spring, 2 = summer, 3 = fall, 4 = winter)
mnth	calendar month
holiday	carries a value of 1 if the day is a federal holiday; 0 if not
weathersit	describes the weather of the day; 1 = clear/few clouds, 2 = cloudy, 3 = light precipitation, 4 = heavy precipitation
workingday	carries a value of 1 if the day is neither a holiday nor a weekend day; 0 otherwise
weekday	day of the week (0 = Sunday, 1 = Monday, etc.)
Numerical Variables	**Descriptions**
instant	the day of the year (from 1 to 365)
temp	the temperature in Celsius, divided by 41 (41 was the max temperature observed)
atemp	the 'feels like' temperature in Celsius, divided by 50 (50 was the max 'feels like' temperature observed)
hum	humidity, as a decimal (ranges from 0 to 1)
windspeed	the wind speed, divided by 67 (67 was the max wind speed observed)
registered	count of registered users
casual	count of casual users

3 Modeling and Analysis

3.1 Training

In this section, two models were built to predict casual user count and registered user count, separately. The candidate predictors in our initial model were selected based on correlation matrices, box plots and scatter plots.

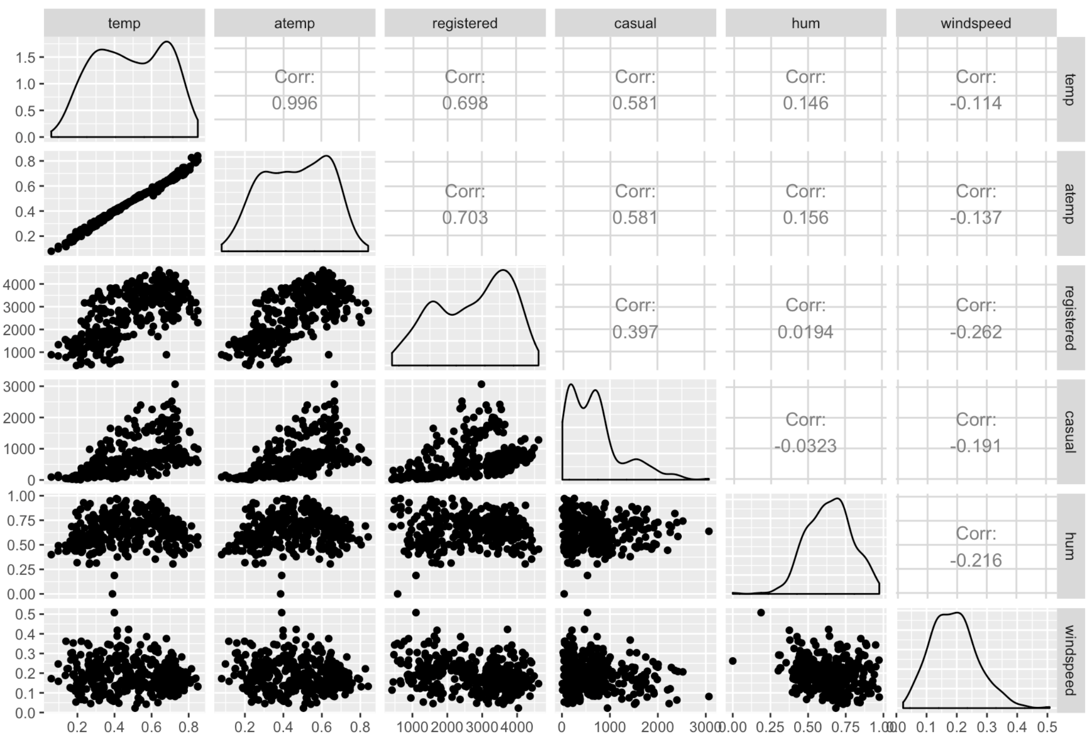

Figure 1: Correlation Plot of continuous variables

The correlation matrix, as shown in Figure 1 above, is produced of continuous variables against response variables, i.e., *registered* and *casual*. It is shown that there is a high correlation between temperature and adjusted temperature. We choose to consider only *atemp* because we speculate feeling temperature is more likely to affect bike rentals than ordinary temperature. Humidity has an extremely weak correlation with both response variables, so it will not be considered in the initial models. Moreover, there is a weak relationship between *windspeed* with both the registered and casual count, so it will be included at first. As for *instant*, since it is just the index of the date from 1/1/2011 to 12/31/2011, it is not considered it in the model.

Boxplots in Figure 2 below and in Figure 8 of the Appendix show that when the data is separated into groups based on categorical variables *season* and *workingday*, the distributions of registered count and casual count for different levels are significantly different, so *season* and *workingday* are included in the initial models.

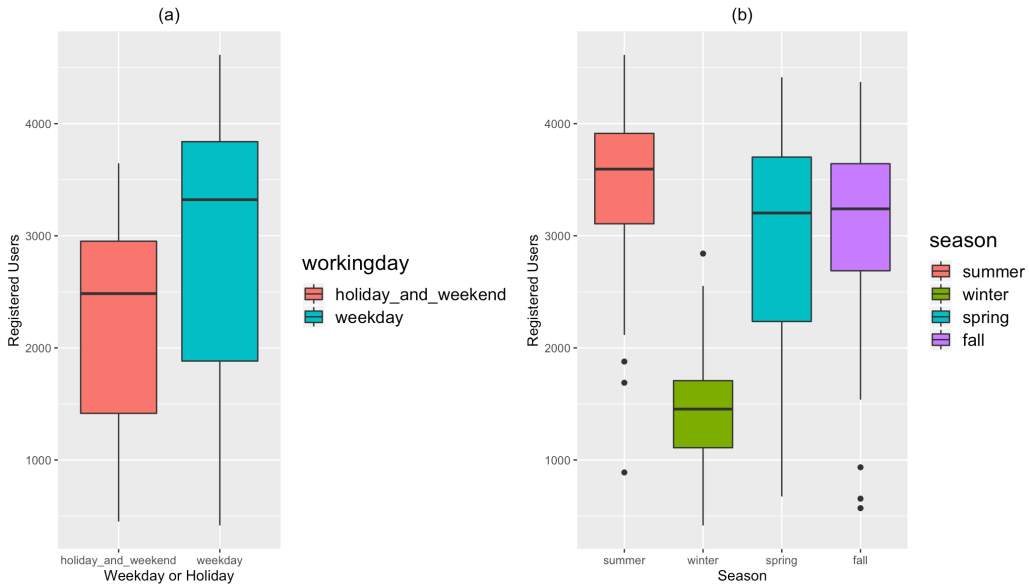

Figure 2: Box Plot of (a) *Workingday* vs. Registered Count; (b) Season vs. Registered Count

The scatter plot in Figure 3(a) shows a quadratic trend between date and registered count, so a quadratic term of *date* will be placed in the initial models. Additionally, Figure 3(b) shows evidence of interaction between *season* and *atemp* on registered count. To be more specific, the slope of the regression lines within each season are significantly different from each other. Interestingly, from Figures 9 and 10 in the Appendix, the same quadratic trend between *date* and *casual* as well as interaction between *season* and *atemp* on casual count were found. So *date*, *season* and *atemp* are included in the initial models.

Therefore, the initial ordinary least squares (OLS) models to predict *registered* and *casual* are run using the following variables: *date*, $date^2$, *atemp*, *windspeed*, *workingday*, *season*, *weathersit* and *atemp*season*.

According to the summary results of our two initial models from R, as shown in Tables 5 and 6 in the Appendix, each variable is highly significant under significance level $\alpha = 0.05$. The global F-test statistics are also extremely high in both initial models. To improve upon the initial models, stepwise variable selection based on Bayesian Information Criterion (BIC) is then performed. The result gives the exact same set of covariates, which suggests that the initial variable selection may have been successful. Now, diagnostics will be performed on the initial models, starting by checking whether the normality assumption is satisfied in both models. From the histograms and qq-plots shown in Figure 4 and Figure 12, the standard residuals are not normally distributed.

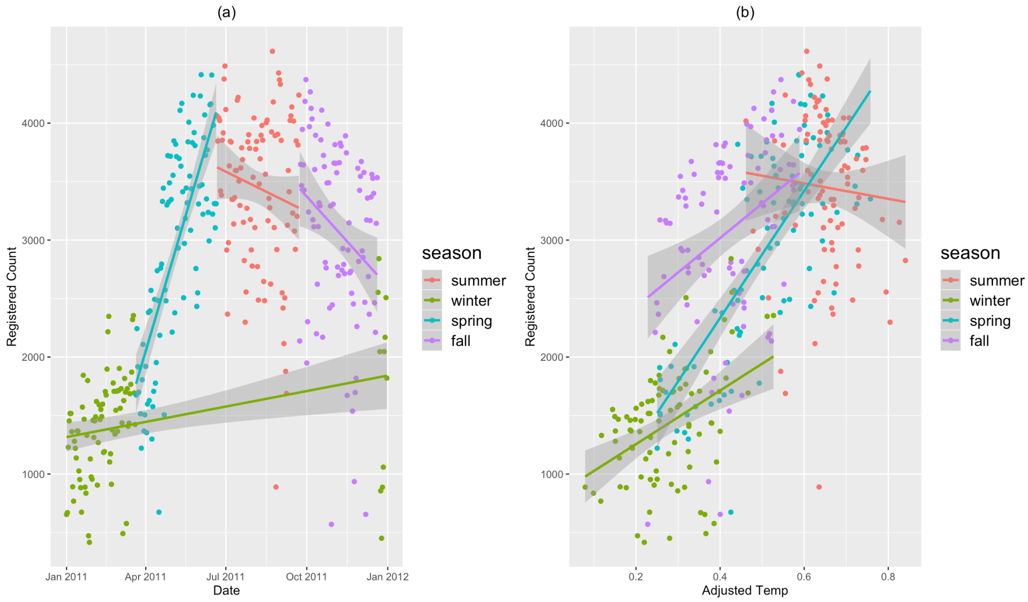

Figure 3: (a) Date vs. Registered Count, grouped by Season; (b) Adjusted Temp. vs. Registered Count, grouped by Season

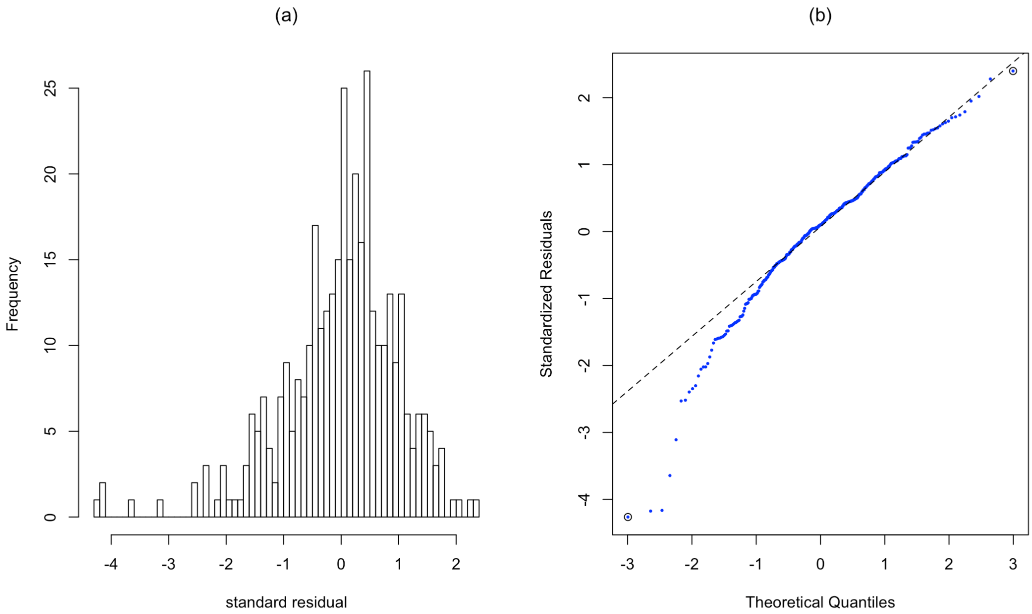

Figure 4: (a) Histogram of Std. Residuals; (b) QQ-Plot of Std. Residuals

Also, in Figure 5 (a) and (c), when the standardized residuals are plotted against fitted values, the data are not evenly distributed within two standard deviations; thus, assumptions for the OLS model may not be met. Weighted Least Squares (WLS) regression is performed to account for this in both models.

The WLS method will place greater weight on lower registered counts (particularly under 3000) and casual counts (particularly under 1000) where there is smaller variance, and less weight on higher counts. The WLS models should now be able to predict higher registered counts with greater accuracy, and the summaries of the R outputs of the two models are shown in Table 2 and Table 3.

From the R output, the WLS models appears to be more or less equivalent with the initial models for predicting registered and casual counts, and the F and t-statistics change little. When looking at the summary output for the WLS model for *casual*, *season* becomes non-significant across the board

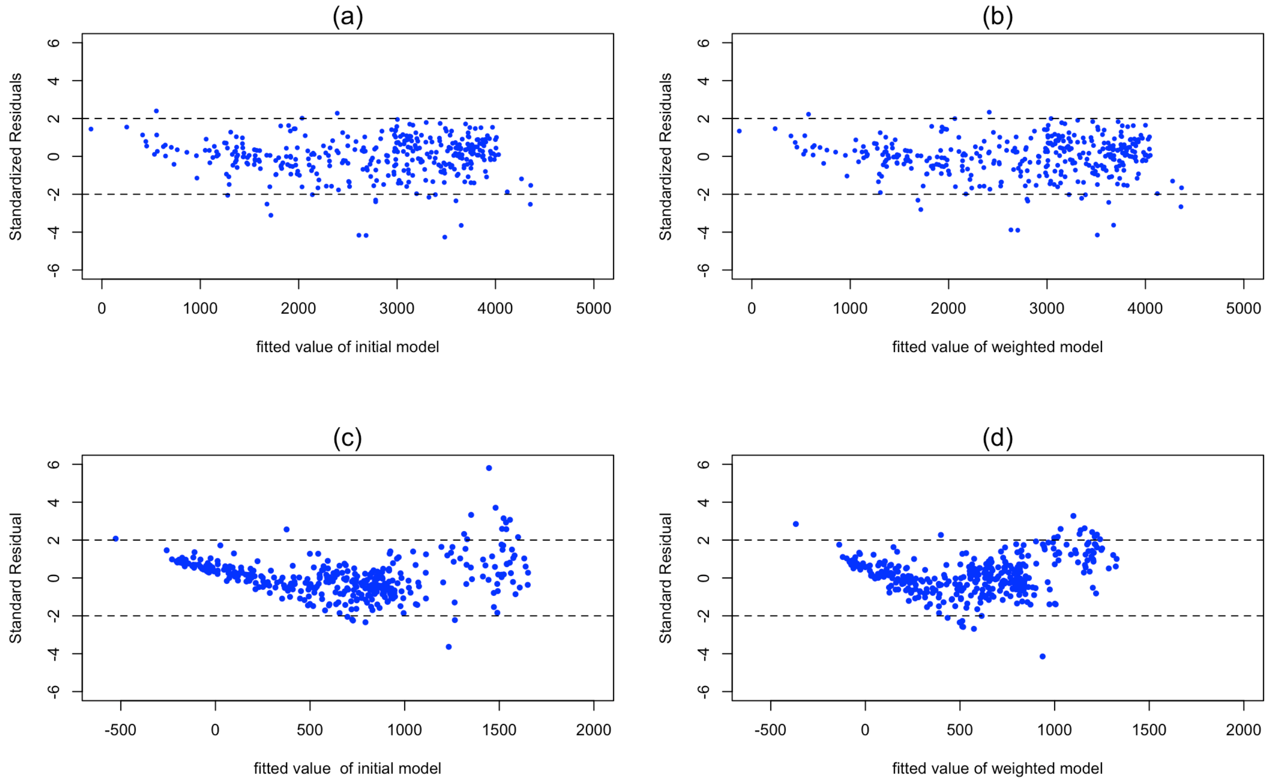

Figure 5: (a) Fitted vs. Std. Residuals, OLS Registered; (b) WLS Registered; (c) OLS Casual; (d) WLS Casual

Table 2: WLS Registered Model Summary : A summary of the predictors in the model and their corresponding coefficients, t-statistics, and p-values, $\alpha = 0.05$

Predictor	**Coefficient**	**t-statistic**	**p-value**	**Significance**
Intercept	4131.53	88.37	1.41×10^{-15}	***
date	9.29	3.47	5.77×10^{-4}	***
$I(date^2)$	-0.02	-3.37	8.28×10^{-4}	***
windspeed	-768.25	-2.54	1.17×10^{-2}	**
weathersit_mist/cloudy	-358.40	-7.41	9.28×10^{-13}	***
weathersit_light_snow/rain/storm	-1681.03	-13.29	$< 2 \times 10^{-16}$	***
workingday_weekday	751.82	15.60	$< 2 \times 10^{-16}$	***
season_winter	-3706.75	-7.26	2.51×10^{-12}	***
season_spring	-4064.66	-8.58	3.07×10^{-16}	***
season_fall	-2708.20	-5.75	2.00×10^{-8}	***
atemp	-2843.64	-4.58	6.46×10^{-6}	***
season_winter$\times atemp$	4628.66	5.18	3.83×10^{-7}	***
season_spring$\times atemp$	6602.89	9.00	$< 2 \times 10^{-16}$	***
season_fall$\times atemp$	4783.70	5.97	5.89×10^{-9}	***
F-statistic: 147, p-value $< 2 \times 10^{-16}$				

but the interaction terms are still highly significant. That may be because of the crossover interaction that exists between *season* and *atemp* in casual counts.

However, in residual analysis, as for registered count shown in Figure 5 (a) and (b), the WLS model has tiny improvement compared with OLS model as several of the most extreme outliers regress towards the middle. As for casual count shown in Figure 5 (c) and (d), compared with OLS model, the WLS model has obvious improvement as the outliers become less extreme overall. Moreover, there are slightly pattern in Figure 5 (b) and obvious pattern in Figure 5 (d), which indicates that some valuable predictors are lacking in both the final WLS models of *casual* and *registered*.

In order to test whether the two WLS models fit the data well, the actual and predicted values are plotted together (as shown in Figure 6 and 7). It appears that the registered WLS model is able to capture the overall trend of the data. Cautiously, this model is moved forward for validation.

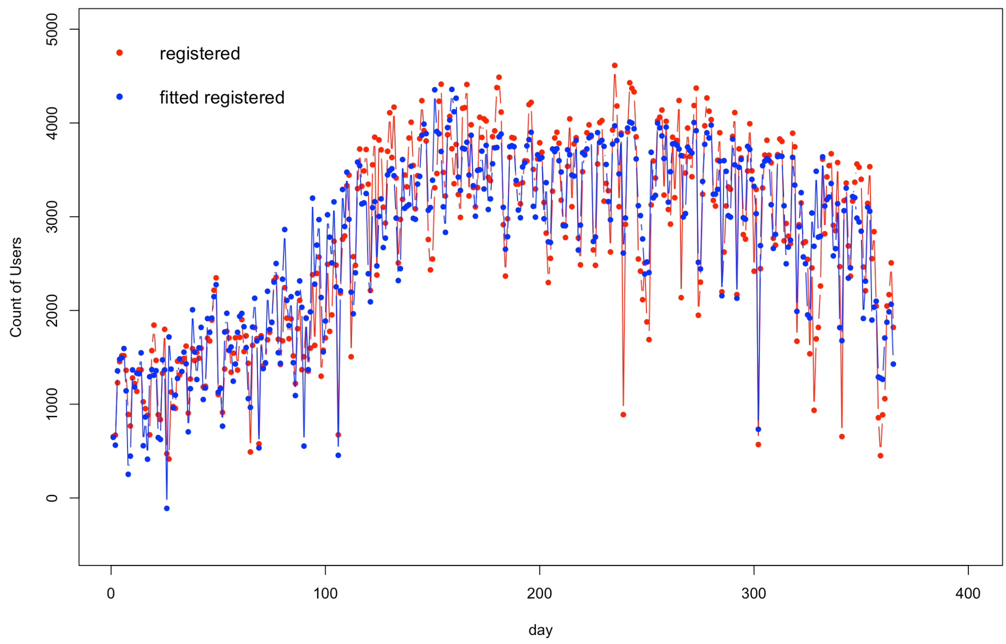

Figure 6: Registered Counts in 2011 vs. Fitted Values

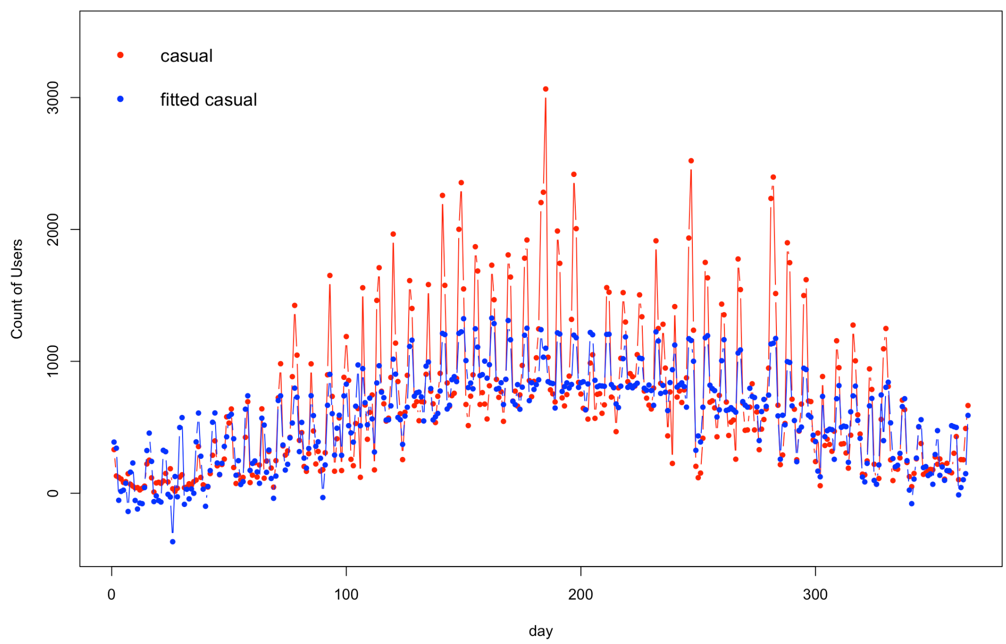

Figure 7: Casual Counts in 2011 vs. Fitted Values

Also, in the casual WLS model, compared with the WLS model for registered users, it does not perform well. To be more specific, it does not capture the general trend, and almost all casual counts are underpredicted, especially in summer and autumn. Therefore, based on what has been discussed above, although the WLS models may not be optimal, it is a clear advancement from the OLS models. Thus, both of the two WLS models were proceeded forward for validation.

Table 3: WLS Casual Model Summary : A summary of the predictors in the model and their corresponding coefficients, t-statistics, and p-values, $\alpha = 0.05$

Predictor	**Coefficient**	**t-statistic**	**p-value**	**Significance**
Intercept	974.04	3.52	4.93×10^{-4}	***
date	3.98	3.64	3.13×10^{-4}	***
$I(date^2)$	-0.01	-4.00	7.83×10^{-5}	***
windspeed	-457.01	-3.09	2.15×10^{-3}	**
weathersit_mist/cloudy	-156.60	-6.29	9.74×10^{-10}	***
weathersit_light_snow/rain/storm	-392.22	-8.16	6.16×10^{-15}	***
workingday_weekday	-386.84	-13.89	$< 2 \times 10^{-16}$	***
season_winter	-733.56	-2.60	9.62×10^{-3}	**
season_spring	-681.80	-2.50	1.28×10^{-2}	*
season_fall	-862.46	-3.12	1.93×10^{-3}	**
atemp	-76.54	-0.20	8.42×10^{-1}	NS
season_winter$\times atemp$	1103.52	2.34	1.96×10^{-2}	*
season_spring$\times atemp$	1171.31	2.73	6.76×10^{-3}	**
season_fall$\times atemp$	1573.41	3.37	8.42×10^{-3}	***
F-statistic: 67.39, p-value $< 2 \times 10^{-16}$				

3.2 Validation

The prediction ability of the two WLS models were evaluated by Relative Mean Square Error (RMSE), which is calculated by $\frac{\sum(Y_i - \hat{Y}_i)^2}{\sum Y_i^2}$. RMSE of both registered and casual WLS models are small, as shown in Table 4. However, in casual WLS model, RMSE is 0.256678, which is almost more than twice of the RMSE for the registered WLS model, which is 0.1084512. This may be because compared with casual users, the behavior of users who have registered may use bike-sharing more regularly, which benefits our registered model.

Table 4: MSE and RMSE for the WLS models

User Type	**MSE for Training**	**MSE for Validation**	**RMSE for Validation**
Registered	173107.7	3765477	0.1084512
Casual	105327.8	406685.6	0.256678

4 Prediction

For the purpose of testing the prediction ability of the two WLS models, the actual bike rental counts in 2012 and the corresponding predicted values are plotted together (as shown in Figure 8 and 9). As for registered count in Figure 8, the predicted values capture the general trend of the real data. That is to say that, in warm seasons, more registered users tend to rent bikes. Also, there are simultaneous upward and downward trends between predicted and actual values. However, almost all registered counts are consistently underpredicted.

As for registered count in Figure 9, it is obvious that although the general trend is slightly captured by the model, the range (amplitude) of fluctuation in predicted values is much smaller than in real data, especially in the summer and autumn seasons. Furthermore, the predicted trend cannot capture weekly spikes in the real trend, even after considering whether or not a particular day is on the weekend. Lastly, almost all casual counts are underpredicted.

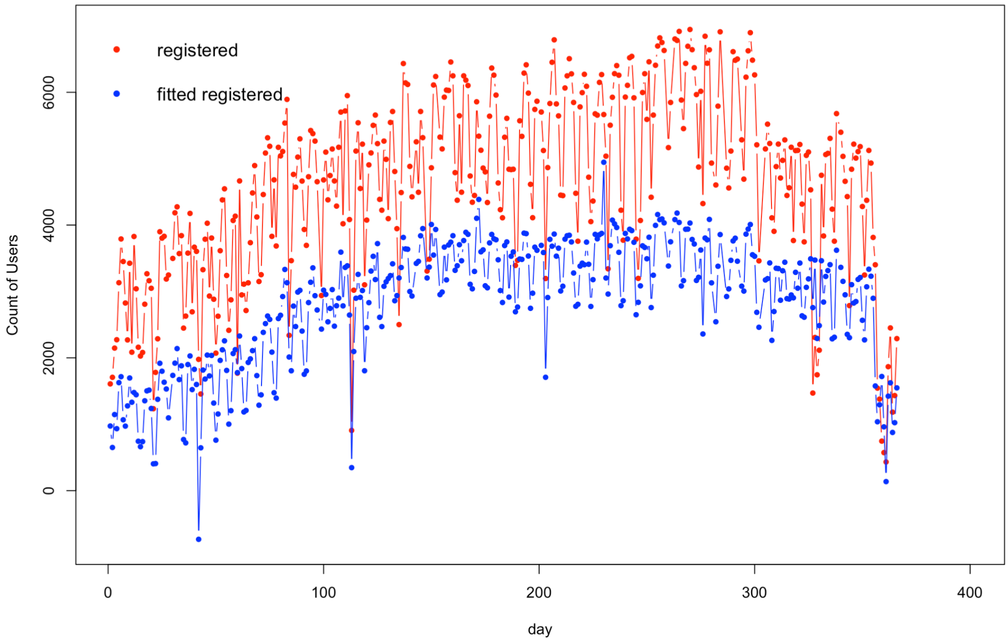

Figure 8: Registered Counts in 2012 vs. Fitted Values

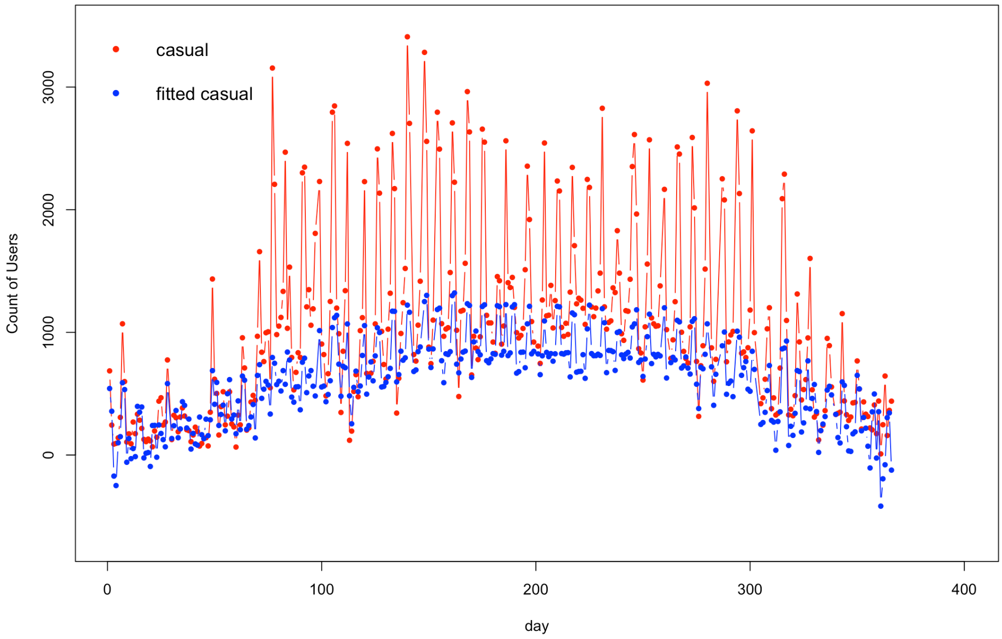

Figure 9: Casual Counts in 2012 vs. Fitted Values

5 Discussion

Overall, our study of bike rentals provided some insight into the factors that go into the number of bikes rented by registered and casual users. It is clear that feeling temperature, wind speed, day of the year, weather situation, whether a day is a working day or not and season are factors that affect bike renting, as our model suggests. However there may still be some factors that go into it that were either unavailable to us, or are altogether immeasurable.

As further diagnostics were performed on our models, the WLS models may not be as effective as previously thought. For the registered model, the RMSE is 0.1084512 and for our casual model, it is 0.256678. The problem for our models is the fact that bike renting is a trend that is gaining more users as time goes by, so even though our models may perform well for predicting 2011 bike rentals, the influx of new users, both casual and registered, is a factor that cannot be quantified for our purposes. Additionally, the models were trained on only one year of data (2011), so the predictability was possibly limited by a lack of available data.

As far as data modification goes, one thing that would have aided predictions for larger registered counts is changing the variables *holiday*. As is, *holiday* only indicates the presence of a holiday, but not the effects of the holiday. Every holiday will have a different effect on bike rentals. For example, July 4 may increase rentals, while Thanksgiving is a holiday that will likely see a decrease in bike rentals, just by nature. If the effect for each holiday on bike rentals could be specified, the results of the models may be improved slightly. Additionally, *holiday* does not account for the duration. That is to say, some holidays are more than one day, but in the dataset are only marked as one day, such as Christmas or spring break.

Finally, the results can offer insights to be applied to bike-sharing businesses. In the summer, when more people are renting bikes, bike rental companies may build more bike stations to meet the growing demand and increase profits. Similarly, businesses may want to lower the cost of maintenance by removing bike stations in the winter when demand for bikes is lower. If a company seeks to increase winter rentals, discounts and coupons may be a good incentive for consumers.

References

[1] Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.

6 Appendix

Table 5: Initial OLS Registered Model Summary : A summary of the predictors in the initial model and their corresponding coefficients, t-statistics, and p-values, $\alpha = 0.05$

Predictor	Coefficient	t-statistic	p-value	Significance
Intercept	4097.45	8.07	1.11×10^{-14}	***
date	8.69	3.43	6.89×10^{-4}	***
$I(date^2)$	-0.02	-3.36	8.52×10^{-4}	***
windspeed	-842.27	-2.74	6.40×10^{-3}	**
weathersit_mist/cloudy	-349.21	-7.06	9.26×10^{-12}	***
weathersit_light_snow/rain/storm	-1639.91	-13.88	$< 2 \times 10^{-16}$	***
workingday_weekday	731.64	15.12	$< 2 \times 10^{-16}$	***
season_winter	-3632.77	-6.98	1.50×10^{-11}	***
season_spring	-4082.84	-8.32	1.98×10^{-15}	***
season_fall	-2584.75	-5.20	3.37×10^{-7}	***
atemp	-2687.93	-4.07	5.82×10^{-5}	***
season_winter$\times atemp$	4496.83	5.05	7.17×10^{-7}	***
season_spring$\times atemp$	6635.91	8.70	$< 2 \times 10^{-16}$	***
season_fall$\times atemp$	4570.83	5.40	1.21×10^{-7}	***
F-statistic: 148.3, p-value $< 2 \times 10^{-16}$				

Table 6: Initial OLS Casual Model Summary : A summary of the predictors in the initial model and their corresponding coefficients, t-statistics, and p-values, $\alpha = 0.05$

Predictor	Coefficient	t-statistic	p-value	Significance
Intercept	1647.13	4.85	1.87×10^{-6}	***
date	5.21	3.07	2.32×10^{-3}	***
$I(date^2)$	-0.01	-3.48	5.70×10^{-4}	***
windspeed	-627.84	-3.06	2.42×10^{-3}	**
weathersit_mist/cloudy	-175.57	-5.30	2.06×10^{-7}	***
weathersit_light_snow/rain/storm	-450.29	-5.69	2.64×10^{-8}	***
workingday_weekday	-657.69	-20.30	$< 2 \times 10^{-16}$	***
season_winter	-1253.49	-3.60	3.68×10^{-4}	***
season_spring	-1121.98	-3.42	7.10×10^{-4}	***
season_fall	-1499.89	-4.51	8.86×10^{-6}	***
atemp	-670.22	-1.52	1.30×10^{-1}	NS
season_winter$\times atemp$	1877.45	3.15	1.78×10^{-3}	**
season_spring$\times atemp$	1787.59	3.50	5.21×10^{-4}	***
season_fall$\times atemp$	2751.43	4.86	1.78×10^{-6}	***
F-statistic: 80.74, p-value $< 2 \times 10^{-16}$				

Table 7: Special Events: A list of weather events considered anomalous

Date	**Special Weather Event**
2012-10-29	Hurricane Sandy
2012-10-30	Hurricane Sandy
2012-10-19	Heavy Storms
2012-09-18	Heavy Rain
2012-07-18	Severe Storms
2012-06-01	Tornado
2012-12-04	Warm Weather Floods
2012-10-07	Cold Temperatures
2012-05-21	Storms
2012-10-19	Heavy Storms

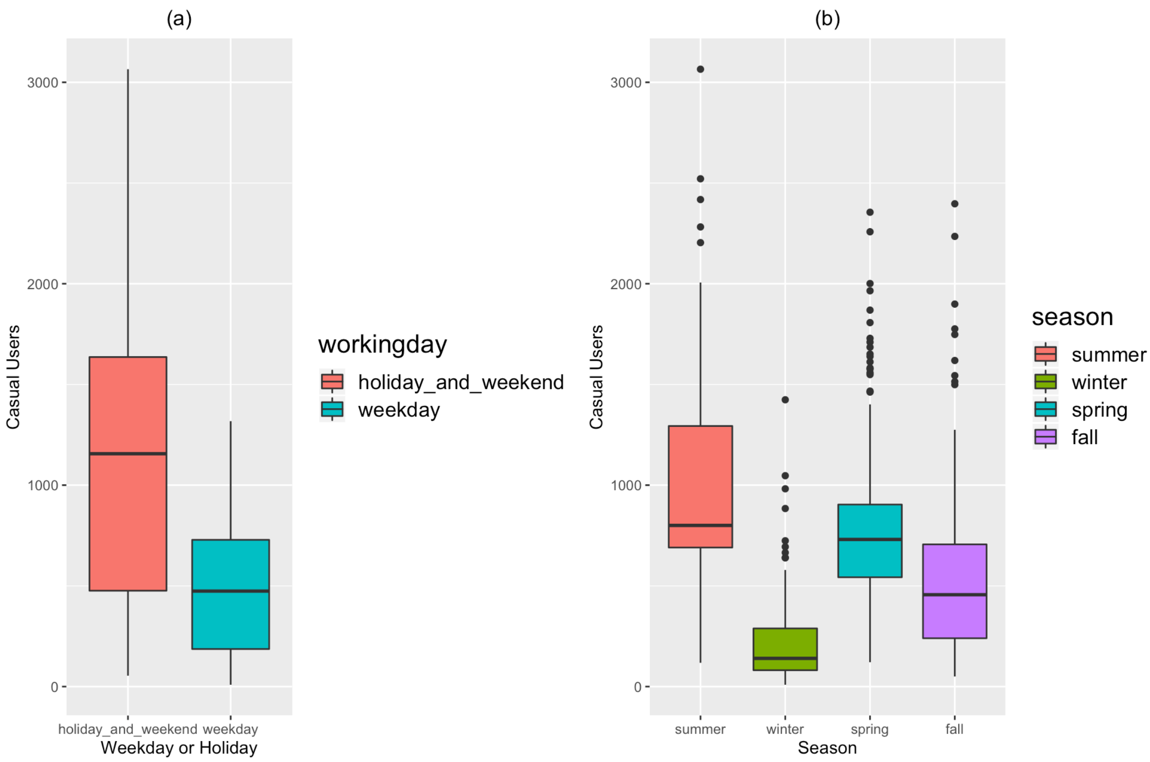

Figure 10: Box Plot of (a) Workingday vs. Casual Count; (b) Season vs. Casual Count

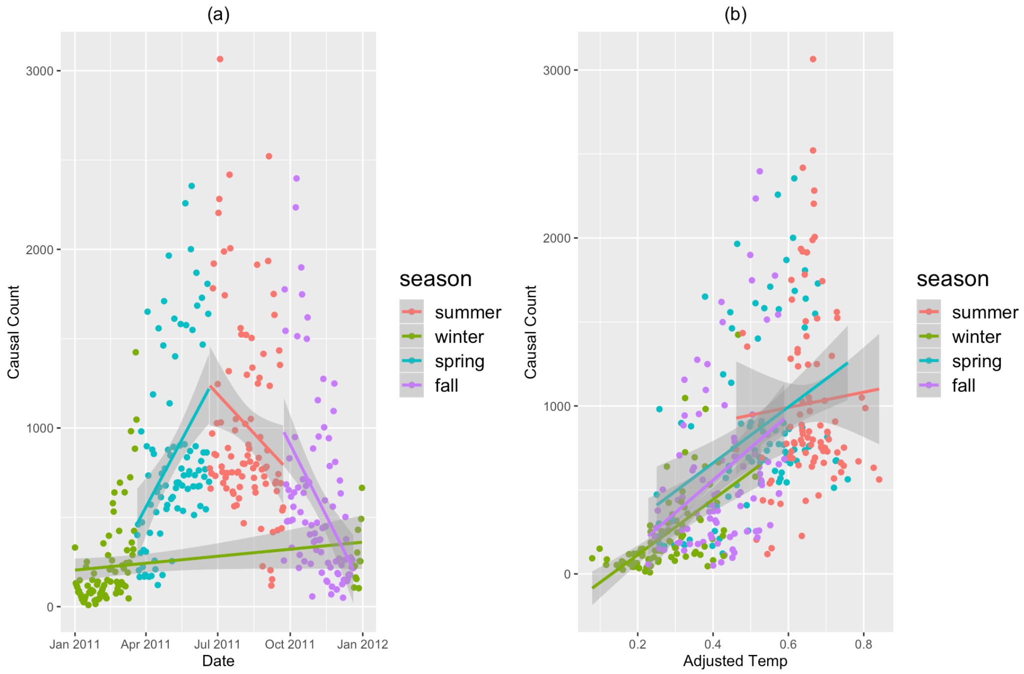

Figure 11: (a) Date vs. Casual Count, grouped by Season; (b) Adjusted Temp. vs. Casual Count, grouped by Season

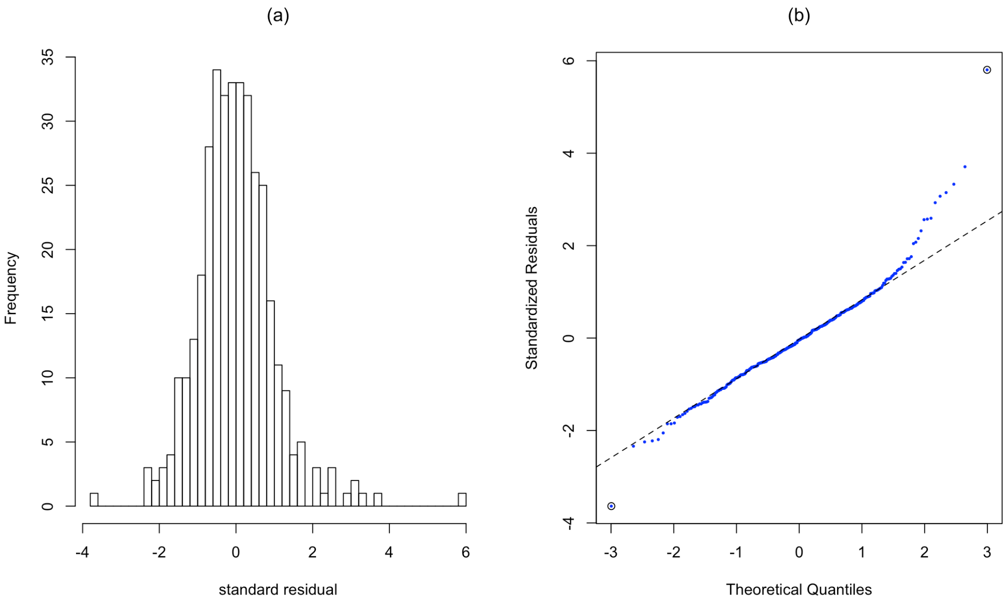

Figure 12: (a) Histogram of Std. Residuals; (b) QQ-Plot of Std. Residuals

```
---
title: "Final Report_Group4"
author: "Chen Shaoxuan"
date: "11/25/2019"
output: html_document
editor_options:
  chunk_output_type: console
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown
```{r}
library(tidyverse)
library(janitor)
library(readxl)
library(readr)
library(ggridges)
library(e1071)
library(ISLR)
library(randomForest)
library(rpart)
library(caret)
library(pls)
library(gbm)
library(pander)
library(zoo)
library(corrplot)
library(lubridate)
library(carData)
library(car)
library(plotly)
library(ggpubr)
library(glmnet)
library(GGally)
```


```{r}
#data processing
options(scipen = 200) #change all number to numeric
options(scipen = 1)
bicycle <- read.csv("~/Dropbox/575 Linear Models/Group Project/day.csv", header = TRUE)
attach(bicycle)
##NOTE## The line below originally read 'format = "%m/%d/%Y"'
bicycle$dteday = as.Date(dteday, format= "%Y-%m-%d")  #change the factor date to dateformat

##NOTE## I commented the below line since it was clearing bike_2011
#year(bicycle$dteday) <- as.numeric(paste("20", as.character(year(bicycle$dteday)), sep=""))

#add special event variable
###NOTE: positive---represents the holidays which is likely that more people go out and use bikecle.
###      extre_weather---represent the day in 2012, which the date of bad weather happened.(We marked them by the information we got from the paper that instructor sent before.)
bicycle= bicycle %>%
  mutate(weathersit = as.factor(weathersit)) %>%
  mutate(special_event = dteday) %>%
  mutate(special_event = as.character(special_event)) %>%
  mutate(special_event = ifelse(special_event %in% c("2011-07-04", "2011-11-23", "2011-12-24", "2011-10-08", "2011-05-27", "2011-11-22", "2011-11-12", "2011-04-16", "2011-03-23", "2011-05-13", "2011-02-11", "2011-01-23", "2011-09-29", "2012-07-04", "2012-11-23", "2012-12-24", "2012-10-08", "2012-05-27", "2012-11-22", "2012-11-12", "2012-04-16", "2012-03-23", "2012-05-13", "2012-02-11", "2012-01-23", "2012-09-29"), "positive",
                ifelse(special_event %in% c("2012-10-29", "2012-10-30", "2012-10-19", "2012-09-18", "2012-07-18", "2012-06-01", "2012-12-04", "2012-10-07", "2012-05-21", "2012-04-07", "2012-05-26", "2012-09-15", "2012-10-11", "2012-10-12", "2012-01-29"),"extre_weather", "normal"))) %>%
  mutate(special_event = as.factor(special_event)) %>%
  mutate(season = ifelse(season %in% c("2"), "spring",
                ifelse(season %in% c("3"),"summer",
                 ifelse(season %in% c("4"),"fall", "winter")))) %>%
  mutate(weathersit = ifelse(weathersit %in% c("1"), "clear/partial_cloudy",
                ifelse(weathersit %in% c("2"),"mist_cloudy",
```
```

```
                  ifelse(weathersit %in% c("3"),"light_snow/rain/storm", "heavy_snow/rain/storm")))) %>%
  mutate(holiday = ifelse(holiday %in% c("1"), "yes","no")) %>%
  mutate(workingday = ifelse(workingday %in% c("0"), "holiday_and_weekend","weekday"))

# change the form of some variables
bicycle$season <- as.factor(bicycle$season)
bicycle$yr <- as.factor(bicycle$yr)
bicycle$mnth <- as.factor(bicycle$mnth)
bicycle$holiday <- as.factor(bicycle$holiday)
bicycle$workingday <- as.factor(bicycle$workingday)
bicycle$weathersit<- as.factor(bicycle$weathersit)
#change the factor level of season variable
bicycle$season = factor(bicycle$season,levels(bicycle$season)[c(3,4,2,1)])
bicycle$weathersit = factor(bicycle$weathersit,levels(bicycle$weathersit)[c(1,3,2)])

bicycle = bicycle %>%
      mutate(date= c(seq(1:365), seq(1:366)))
#seperate the data into two parts: in year 2011/2012
bike_2011= subset(bicycle, dteday>= "2011-01-01" & dteday <= "2011-12-31")
bike_2012= subset(bicycle, dteday>= "2012-01-01" & dteday <= "2012-12-31")
detach(bicycle)
```

```{r}
attach(bike_2011)
#We draw the scatter matrix (without discrete variables)
pairs(~registered+instant+temp+atemp+hum+windspeed+casual+registered,data=bike_2011, upper.panel = NULL)
DataSet<-data.frame(temp,atemp,registered,casual,hum,windspeed)
ggpairs(DataSet)
#Then we have:Correlation Matrices + Correlation Plot
#X <- cbind(bike_2011 %>% select(instant,temp:registered))
#corr_data <- cor(X)
#round(corr_data, 3)
#corrplot(corr_data, method = "number")
```

#box plot

```{r Adrian initial model}
# Make a scatterplot of each covariate against 'count'
# I used these to determine the categorical variables to include in the initial model
# boxplot
p1 =ggplot(bike_2011, aes(x = workingday, y = registered, fill = workingday)) + geom_boxplot() + xlab("Weekday or Holiday") + ylab("Registered Users")+theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ ggtitle("(a)")+
  theme(plot.title = element_text(hjust = 0.5))

p2 =ggplot(bike_2011, aes(x = season, y = registered, fill = season)) + geom_boxplot() + xlab("Season") + ylab("Registered Users")+theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ ggtitle("(b)")+
  theme(plot.title = element_text(hjust = 0.5))
ggarrange(p1,p2, ncol=2, nrow = 1)
```

#scatter plot
```{r}
#date-cnt
p3 <- ggplot(bike_2011, aes(x = dteday, y = registered, color = season)) + geom_point() + xlab("Date") + ylab("Registered Count")+theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ geom_smooth(method = "lm") + ggtitle("(a)")+
  theme(plot.title = element_text(hjust = 0.5))
# temperature - cnt
p4 <- ggplot(bike_2011, aes(x=atemp, y=registered, color= season)) + geom_point() + xlab("Adjusted Temp") + ylab("Registered Count") +theme(legend.title = element_text(size=16)) + theme(legend.text = element_text(size=13))+ geom_smooth(method = "lm") + ggtitle("(b)")+
  theme(plot.title = element_text(hjust = 0.5))
# humandity - cnt
p5 <- ggplot(bike_2011, aes(x=hum, y=registered, color= season)) + geom_point() + xlab("Humidity") + ylab("Registered Count") + geom_smooth(method = "lm") + ggtitle("(c)")+
  theme(plot.title = element_text(hjust = 0.5))
# windspeed -cnt
p6 <- ggplot(bike_2011, aes(x=windspeed, y=registered, color= season)) + geom_point() + xlab("Wind Speed") + ylab("Registered Count") + geom_smooth(method = "lm") + ggtitle("(d)")+
  theme(plot.title = element_text(hjust = 0.5))
```

```

ggarrange(p3,p4, ncol=2, nrow = 1)

#plot in report
#ggplot(bike_2011, aes(x=atemp, y=registered, color= season)) + geom_point() + xlab("Adjusted
    Temp") + ylab("Registered Count") + geom_smooth(method = "lm")
```

#initial model
```{r Adrian initial model}
# Create an initial model based on the variables above:
m.mls.initial <- lm(registered~date+I(date^2)+atemp+season+windspeed+workingday+weathersit+
    season*atemp, data = bike_2011)
options(scipen = 200) #change all number to numeric
options(scipen = 1)
summary(m.mls.initial)

```
###QQ plot and hitogram plot of initial OLS model
```{r}
par(mfrow=c(1,2))
# Standardized Residuals of Initial Model
StanResInitial <- rstandard(m.mls.initial)
# (1) histogram of residual
hist(StanResInitial,50, xlab = "standard residual",main="(a)", font.main =1)

#qqplot
q1_initial <- qqnorm(StanResInitial, plot.it = FALSE)
plot(range(q1_initial$x, q1_initial$x), range(q1_initial$y, q1_initial$y), ylab = "
    Standardized Residuals", xlab ="Theoretical Quantiles")
points(q1_initial, col="blue", pch = 19, cex = 0.3)
qqline(StanResInitial,lty=2)+ title("(b)", font.main =1, adj = 0.5)
```

#Stepwise Variable Selection
```{r Adrian Model v2}
# Variable selection
################################################################################
#Stepwise to choose variable
###refine
StepBIC <- step(m.mls.initial,direction="both", data=bike_2011, k=log(365))
#we get exact same result
```

#Final OLS model
```{r}
#OLS model
m.mls.step_interaction <-lm(registered~date+I(date^2)+atemp+season+windspeed+workingday+
    weathersit+season*atemp, data = bike_2011)
summary(m.mls.step_interaction)
```
###
#WLS model
```{r}
# Calculate fitted values from a regression of absolute residuals vs SensorCO
wts <- 1/fitted(lm(abs(residuals(m.mls.step_interaction)) ~ registered))^2
#refit the model with weight
m.mls.step_interaction_wls <-lm(registered~date+I(date^2)+atemp+season+windspeed+workingday+
    weathersit+season*atemp, data = bike_2011, weights=wts)
options(scipen = 200) #change all number to numeric
options(scipen = 1)
summary(m.mls.step_interaction_wls) #afater checking, RMSE is little bit smaller, but Y vs Y_
    hat almost the same.
```

Training
```{r}
#std residual for WLS
StanResInitial_wls <- rstandard(m.mls.step_interaction_wls)
```

######Combine the residual plot together
```{r}
par(mfrow=c(2,2))
#combine std residual vs. fitted in OLS and Weighted
#OLS
```

```
plot(fitted(m.mls.step_interaction),StanResInitial,xlab="fitted value of initial model", ylab=
     "Standardized Residuals",xlim=c(0, 5000), ylim=c(-6, 6), col="blue", pch = 19, cex = 0.5)
abline(h=2,lty=2)
abline(h=-2,lty=2)+ title("(a)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)
#Weighted
plot(fitted(m.mls.step_interaction_wls),StanResInitial_wls,xlab="fitted value of weighted
     model", ylab="Standardized Residuals",xlim=c(0, 5000), ylim=c(-6, 6), col="blue", pch =
     19, cex = 0.5)
abline(h=2,lty=2)
abline(h=-2,lty=2)+ title("(b)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)

##########
#together with the casual plots
plot(c.mls.initial$fitted.values, StanResInitial_c, col = "blue",xlim = c(-600,2000), ylim = c
     (-6,6), xlab = "fitted value  of initial model", ylab = "Standard Residual", pch = 20)
abline(h=2, lty = 2)
abline(h=-2, lty = 2)+title("(c)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)

plot(c.mls.weighted$fitted.values, StandResWeight_c, xlim = c(-600,2000), ylim = c(-6,6), col
     = "blue", xlab = "fitted value of weighted model", ylab = "Standard Residual", pch = 20)
abline(h=2, lty=2)+title("(d)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)
abline(h=-2, lty=2)
#########

#plot Y v.s. Y_hat in WLS model
par(mfrow=c(1,1))
plot(date, registered, pch = 20, col = "red", xlim = c(1,400), ylim = c(-500, 5000), xlab = "
     day", ylab = "Count of Users")+ title("", adj =0, line = 0.5, cex.main=1.1, font.main =1)
points(date, m.mls.step_interaction$fitted.values, pch = 20, col = "blue", xlim = c(1,400),
     ylim = c(-500, 5000), xlab = "day", ylab  = "Count of Users")
legend ("topleft", legend = c("registered", "fitted registered"), col =c("red", "blue"), lty
     =0, cex=1.2, pch=20, bty = "n")
lines(date, registered, pch = 20, col ="red", type = "b", lty = 1)
lines(date, m.mls.step_interaction$fitted.values, pch = 20, col ="blue", type = "b", lty = 1)
```

```{r}
# Validation
     ------------------------------------------------------------------------------------------

#delete the entries of data which include extre_weather date
bike_2012= bike_2012 %>%
 filter(special_event != "extre_weather")
# Residuals for training data
ResMLS <- resid(m.mls.step_interaction_wls)
# Residuals for validation data
output<-predict(m.mls.step_interaction_wls, se.fit = TRUE, newdata = bike_2012)
ResMLSValidation <- bike_2012$registered - output$fit
```

```{r}
####MSE for OLS model-Training
ResMLS_OLS = resid(m.mls.step_interaction)
# Mean Square Error for training data
mean((ResMLS_OLS)^2)  #summation of (Yi-Yi_hat)^2/n

#####
# Mean Square Error for WLS model-Training
mean((ResMLS)^2)  #summation of (Yi-Yi_hat)^2/n

# Mean Square Error for validation data
mean((ResMLSValidation)^2)

# Relative Mean Square Error for validation data
mean((ResMLSValidation)^2) / mean((bike_2012$cnt)^2)
# summation of (Yi-Yi_hat)^2/summation of Yi^2

# Now the RMSE: 0.1518124, which is smaller than 0.1520601 we got before.
```


#Y vs. Y_hat
```{r}
par(mfrow=c(1,1))
#registered
plot(bike_2012$date,bike_2012$registered, pch = 20, col = "red", xlim = c(1,400), ylim = c
```

```
(-800, 7000), xlab = "day", ylab = "Count of Users", type = 'b')+ title("", adj =0, line = 0.5, cex.main=1.1, font.main =1)
points(bike_2012$date, output$fit, pch = 20, col = "blue", xlim = c(1,400), ylim = c(-800, 7000), xlab = "day", ylab  = "Count of Users", type = 'b')
legend ("topleft", legend = c("registered", "fitted registered"), col =c("red", "blue"), lty =0, cex=1.2, pch=20, bty = "n")
```


#############################
New model: response variable -- casual
```{r}
attach(bike_2011)

# matrix plot of continuos variables
pairs(~casual+instant+temp+atemp+hum+windspeed,data=bike_2011, upper.panel = NULL)
DataSet<-data.frame(temp,atemp,casual,hum,windspeed)
ggpairs(DataSet)

### choose "atemp"  between "temp" and "atemp"

# scatter plot of discrete variables
p2_1 <- ggplot(bike_2011, aes(x = dteday, y = casual, color = season)) + geom_point() + xlab("Date") + ylab("Causal Count") +theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ geom_smooth(method = "lm") + ggtitle("(a)")+
  theme(plot.title = element_text(hjust = 0.5))
# temperature - casual
p2_2 <- ggplot(bike_2011, aes(x=atemp, y=casual, color= season)) + geom_point() + xlab("Adjusted Temp") + ylab("Causal Count") +theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ geom_smooth(method = "lm") + ggtitle("(b)")+
  theme(plot.title = element_text(hjust = 0.5))
# humandity - casual
p2_3 <- ggplot(bike_2011, aes(x=hum, y=casual, color= season)) + geom_point() + xlab("Humidity") + ylab("Casual Count") + geom_smooth(method = "lm") + ggtitle("(c)")+
  theme(plot.title = element_text(hjust = 0.5))
# windspeed-casual
p2_4 <- ggplot(bike_2011, aes(x=windspeed, y=casual, color= season)) + geom_point() + xlab("Wind Speed") + ylab("Causal Count") + geom_smooth(method = "lm") + ggtitle("(d)")+
  theme(plot.title = element_text(hjust = 0.5))

ggarrange(p2_1, p2_2, ncol = 2, nrow = 1)

### consider the interaction of season & humidity and season & windspeed

# boxplot of discrete variables
p2_5 =ggplot(bike_2011, aes(x = workingday, y = casual, fill = workingday)) + geom_boxplot() + xlab("Weekday or Holiday") + ylab("Casual Users")+theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ ggtitle("(a)")+
  theme(plot.title = element_text(hjust = 0.5))

p2_6 =ggplot(bike_2011, aes(x = season, y = casual, fill = season)) + geom_boxplot() + xlab("Season") + ylab("Casual Users")+theme(legend.title = element_text(size=16)) +theme(legend.text = element_text(size=13))+ ggtitle("(b)")+
  theme(plot.title = element_text(hjust = 0.5))
ggarrange(p1,p2, ncol=2, nrow = 1)

ggarrange(p2_5, p2_6, ncol = 2, nrow = 1)
```


Innitial model for casual
```{r}
c.mls.initial <- lm(casual~atemp + season+windspeed+workingday+weathersit+season*atemp + date + I(date^2), data = bike_2011)

options(scipen = 200) #change all number to numeric
options(scipen = 1)
summary(c.mls.initial)
# Residual
Res_c <- resid(c.mls.initial)

par(mfrow = c (1,2))
# (a)standard residual v.s. casual
StanResInitial_c <- rstandard(c.mls.initial)
plot(bike_2011$casual,StanResInitial_c,xlab="Count of Casual Users", ylab="Standardized Residuals", col="blue", xlim = c(0,3500), ylim = c(-6,6), pch = 19, cex = 0.5)
```

```r
abline(h=2,lty=2)
abline(h=-2,lty=2)+ title("(a)", adj =0, line = 0.5, cex.main=1.6, font.main =1)

# (b)standard residual v.s. fitted
plot(fitted(c.mls.initial),StanResInitial_c,xlab="Fitted", ylab="Standardized Residuals", col=
    "blue",xlim = c(-600,2000), ylim = c(-6,6), pch = 19, cex = 0.5)
abline(h=2,lty=2)
abline(h=-2,lty=2)+ title("(b)", adj = 0.5, line = 0.5, cex.main=1.6, font.main =1)


# histgram and qq-plot

# (1) histogram of residual
hist(StanResInitial_c,50, xlab = "standard residual",main="(a)", font.main =1)

# (2) qq-plot & qq-line
qq_initial_c <- qqnorm(StanResInitial_c, plot.it = FALSE)
plot(range(qq_initial_c$x, qq_initial_c$x), range(qq_initial_c$y, qq_initial_c$y), ylab = "
    Standardized Residuals", xlab ="Theoretical Quantiles")
points(qq_initial_c, col="blue", pch = 19, cex = 0.3)
qqline(StanResInitial_c,lty=2)+ title("(b)", font.main =1, adj = 0.5)
```


```{r}
# Variable selection by stepwise
step_model_interaction_c <- step(lm(formula = casual ~ atemp + season+windspeed+workingday+
    weathersit+season*atemp + date + I(date^2),data = bike_2011))

### after the stepwise, variables chosen out are the same as those in initial model.
```

```{r}
# add weight in the initial model
# calculatet the weighte value
wts_c <- 1/fitted(lm(abs(residuals(c.mls.initial)) ~ bike_2011$casual))^2

#refit the model with weight
c.mls.weighted <- lm(casual ~ atemp + season+windspeed+workingday+weathersit+season*atemp +
    date + I(date^2), data = bike_2011,weights=wts_c)

summary(c.mls.weighted)

# the ressidual of weighted model
ResWeight_c <- resid(c.mls.weighted)

# standard residual
StandResWeight_c <- rstandard(c.mls.weighted)

```


validation
```{r}
#delete the entries of data which include extre_weather date
bike_2012= bike_2012 %>%
 filter(special_event != "extre_weather")

# Residaul for validation data
outcome_c <- predict(c.mls.weighted, se.fit = TRUE, newdata = bike_2012)
ResMLSValidation_c <- bike_2012$casual - outcome_c$fit

# compare residual of traing and validation (plot)
#par(mfrow=c(1,1))
#plot(bike_2011$casual,Res_c,xlab="casual ", #ylab="Residuals",xlim=c(0,3500),
#     ylim=c(-600,2500),  col=c("blue"), lty=0, cex=1, pch=19)
#points(bike_2012$casual,ResMLSValidation_c,xlab="casual", ylab="Residuals",xlim=c(0,3500),
    ylim=c(-600,2500),col="red", lty=0, cex=1, pch=19)
#legend(3, 4000, legend=c("Training","Validation"), col=c("blue","red"), lty=0, cex=0.7, pch
    =19)

```

```{r}
#MSE for training OLS
Res_OLS <- resid(c.mls.initial)
```

```
mean((Res_OLS)^2)

# Mean Square Error for training data WLS
mean((ResWeight_c)^2)  #summation of (Yi-Yi_hat)^2/n

# Mean Square Error for validation data
mean((ResMLSValidation_c)^2)

# Relative Mean Square Error for validation data
mean((ResMLSValidation_c)^2) / mean((bike_2012$casual)^2)
# summation of (Yi-Yi_hat)^2/summation of Yi^2
```



```{r}
#training
# plot of
plot(bike_2011$date, bike_2011$casual, col = "red", type = "b", pch = 20,ylim = c(-400, 3500), ylab = "Count of Users",xlab = "day")
points(bike_2011$date, c.mls.weighted$fitted.values, col ="blue" , type = "b",ylim = c(-400, 3500), pch = 20)
legend("topleft", legend = c("casual", "fitted casual"), col = c("red", "blue"), cex = 1.2, lty = 0, pch = 20, bty = "n")

# Prediction
# y and y hat
par(mfrow = c (1,1))
plot(bike_2012$date, bike_2012$casual, pch = 20, col = "red", xlim = c(1,400), ylim = c(-700, 3500), xlab = "day", ylab = "Count of Users", type = 'b')+ title("", adj =0, line = 0.5, cex.main=1.1, font.main =1)
points(bike_2012$date, outcome_c$fit, pch = 20, col = "blue", xlim = c(1,400), ylim = c(-700, 3500), xlab = "day", ylab  = "Count of Users", type = 'b')
legend ("topleft", legend = c("casual", "fitted casual"), col =c("red", "blue"), lty=0, cex =1.2, pch=20, bty = "n")
```

```{r}
# comapre wieghted model and initial model
# fitted value v.s. standard residual
plot(c.mls.initial$fitted.values, StanResInitial_c, col = "red", xlab = "fitted value", ylab = "Standard Residual", pch = 20)
points(c.mls.weighted$fitted.values, StandResWeight_c, col = "blue", xlab = " ", ylab = " ", pch = 20)
abline(h=2, lty=2)
abline(h=-2, lty=2)+title("(a)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)
legend("topleft", legend = c("initial", "weighted"), col = c("red", "blue"), lty = 0, pch = 4, cex = 1.2, bty="n" )


# plot them seperately
par(mfrow = c(1,2))
plot(c.mls.initial$fitted.values, StanResInitial_c, col = "blue",xlim = c(-600,2000), ylim = c(-6,6), xlab = "fitted value  of initial model", ylab = "Standard Residual", pch = 20)
abline(h=2, lty = 2)
abline(h=-2, lty = 2)+title("(a)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)

plot(c.mls.weighted$fitted.values, StandResWeight_c, xlim = c(-600,2000), ylim = c(-6,6), col = "blue", xlab = "fitted value of weighted model", ylab = "Standard Residual", pch = 20)
abline(h=2, lty=2)+title("(b)", adj =0.5, line = 0.5, cex.main=1.6, font.main =1)
abline(h=-2, lty=2)

```

./MA575_Final_Project.Rmd