Bikesharing_Analysis

Rahul Singh

12/16/2017

0. Workplace setup

```r
# import libraries
library(tidyr)
library(MASS)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(plyr)
```

```
## ------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

```r
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(knitr)
library(effects)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'carData'
```

```
## The following objects are masked from 'package:car':
##
##     Guyer, UN, Vocab
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.3
```

```
## Loading required package: lattice
```

```r
library(HistData)
library(gvlma)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# import dataset
bs <- read.csv("hour.csv")
```

```r
# verify dataset has been loaded properly
head(bs,3)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday
## 1       1 2011-01-01      1  0    1  0       0       6          0
## 2       2 2011-01-01      1  0    1  1       0       6          0
## 3       3 2011-01-01      1  0    1  2       0       6          0
##   weathersit temp  atemp  hum windspeed casual registered cnt
## 1          1 0.24 0.2879 0.81         0      3         13  16
## 2          1 0.22 0.2727 0.80         0      8         32  40
## 3          1 0.22 0.2727 0.80         0      5         27  32
```

1. Dataset overview

Source of dataset: http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset Original source: https://www.capitalbikeshare.com/system-data

This dataset contains the hourly and daily record of bike rental counts between year 2011 and 2012 in Washington D.C., provided by Capital Bikeshare, a bike rental company. This dataset also aggregates the weather and the seasonal information particular for that day, including temperature and humidity.

```r
# dataset shape
dim(bs)
```

```
## [1] 17379    17
```

There are 17379 records with 17 columns in this dataset.

```r
# look at structure of dataframe
str(bs)
```

```
## 'data.frame':    17379 obs. of  17 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
##  $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
##  $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
##  $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
##  $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
##  $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

The description of the 17 variables are as follows:

- instant: record index
- dteday: date of bike rental
- season: season of bike rental
  - Note: the original data source indicates that spring is 1, but upon looking at the dates, it seems that 1 is actually winter, meaning the number code is 1: winter, 2: spring, 3: summer, 4: fall
- yr: year of bike rental
- mnth: month of bike rental
- hr: hour of bike rental (0 to 23)
- holiday: whether or not day of rental was a holiday
- weekday: day of week of bike rental
- workingday: whether or not day of rental was netiher a holiday, nor a weekend
- weathersit:
  - Clear, few clouds, partly cloudy
  - Mist, Mist + Cloudy, Mist + broken clouds, Mist + few clouds
  - Light snow, light rain + thundertsorm + scattered clouds, light rain + scattered clouds
  - Heavy rain + ice pallets + thunderstorm + mist, snow + fog
- temp: normalized temperature in celsius (hourly scale)
  - normalization method: (t - t_min) / (t_max - t_min),
  - t_min = -8, t_max = 39
- atemp: normalized feeling temperature in celsius (hourly scale)
  - normalization method: (t - t_min) / (t_max - t_min)
  - t_min = -16, t_max = 50
- hum: normalized humidity (values divided by 100, the max value)
- windspeed: normalzed wind speed (values divided by 67, the max value)
- casual: count of causal (non registered users)
- registered: count of registered users of Capital Bikeshare
- cnt: total count of rental bike (casual + registered)

Let's check if there are any missing values in this dataset

```r
# check for missing values
# source: https://stackoverflow.com/questions/8317231/elegant-way-to-report-missing-values-in-a-data-frame
sapply(bs, function(x) sum(is.na(x)))
```

```
##    instant     dteday     season         yr       mnth         hr 
##          0          0          0          0          0          0 
##    holiday    weekday workingday weathersit       temp      atemp 
##          0          0          0          0          0          0 
##        hum  windspeed     casual registered        cnt 
##          0          0          0          0          0 
```

There are no missing values for any of the columns in this dataset.

2. Cleaning data

There are several columns that need to be cleaned or dropped:

- instant: this is the index of the original data, which is not needed in R, because R has a default indexing applied to dataframes. This column will be dropped.
- dteday: convert to datetypes using as.Date to perform date computations
- season: change to original string value for clarity
- weekday: change to original string value for clarity
- temp: change back to original temperature value, as normalized values are hard to interpret
- atemp: change back to original temperature value
- hum: change back to original humidity
- windspeed: change back to original windspeed
- cnt: verify that casual + registered = cnt

```
# drop instant column
bs <- bs %>%
  dplyr::select(-instant)
```

```
# convert dteday to date time data type
# source: https://www.statmethods.net/input/dates.html
bs$dteday <- as.Date(bs$dteday)

# verify column data type has changed
str(bs)
```

```
## 'data.frame':    17379 obs. of  16 variables:
##  $ dteday    : Date, format: "2011-01-01" "2011-01-01" ...
##  $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
##  $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
##  $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
##  $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
##  $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
##  $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

```
tail(bs, 3)
```

```
##           dteday season yr mnth hr holiday weekday workingday weathersit
## 17377 2012-12-31      1  1   12 21       0       1          1          1
## 17378 2012-12-31      1  1   12 22       0       1          1          1
## 17379 2012-12-31      1  1   12 23       0       1          1          1
##       temp  atemp  hum windspeed casual registered cnt
## 17377 0.26 0.2576 0.60    0.1642      7         83  90
## 17378 0.26 0.2727 0.56    0.1343     13         48  61
## 17379 0.26 0.2727 0.65    0.1343     12         37  49
```

```
# change season back to original string value
bs$season = ifelse(bs$season == 1,"Winter",
                   ifelse(bs$season == 2,  "Spring",
                          ifelse(bs$season == 3, "Summer", "Fall")))

# verify changes
table(bs$season)
```

```
## 
##   Fall Spring Summer Winter 
##   4232   4409   4496   4242
```

```r
# change weekday values back to original string value
bs$weekday = ifelse(bs$weekday == 1, "Mon",
                    ifelse(bs$weekday == 2, "Tues",
                           ifelse(bs$weekday == 3, "Wed",
                                  ifelse(bs$weekday ==4, "Thu",
                                         ifelse(bs$weekday == 5, "Fri",
                                                ifelse(bs$weekday == 6, "Sat", "Sun"))))))

# verify changes
table(bs$weekday)
```

```
##
##  Fri  Mon  Sat  Sun  Thu Tues  Wed
## 2487 2479 2512 2502 2471 2453 2475
```

```r
# change normalized values to original temp values
bs <- bs %>%
  mutate(temp_original = (bs$temp * 47) - 8,
         atemp_original = (bs$atemp * 66) - 16,
         hum_original = hum * 100,
         windspeed_original = windspeed * 67)

# verify changes
bs %>%
  select(temp_original,atemp_original,
         hum_original, windspeed_original) %>%
  head(.,3)
```

```
##   temp_original atemp_original hum_original windspeed_original
## 1          3.28         3.0014           81                  0
## 2          2.34         1.9982           80                  0
## 3          2.34         1.9982           80                  0
```

```r
# verify cnt = casual + registered
# 0 if correct, 1 if incorrect
cnt_ver <- ifelse(bs$cnt == (bs$registered + bs$casual), 0, 1)
# verify that sum of cnt_ver is 0
sum(cnt_ver)
```

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

Some column names are not intuitive, so it is better that they are changed.

```r
bs <- bs %>%
  rename(replace = c('dteday' = 'date',
      'weathersit' = 'weather',
      'cnt' = 'total_bikes'))
```

```r
# check dataframe
head(bs, 3)
```

```
##         date season yr mnth hr holiday weekday workingday weather temp
## 1 2011-01-01 Winter  0    1  0       0     Sat          0       1 0.24
## 2 2011-01-01 Winter  0    1  1       0     Sat          0       1 0.22
## 3 2011-01-01 Winter  0    1  2       0     Sat          0       1 0.22
##    atemp  hum windspeed casual registered total_bikes temp_original
## 1 0.2879 0.81         0      3         13          16          3.28
## 2 0.2727 0.80         0      8         32          40          2.34
## 3 0.2727 0.80         0      5         27          32          2.34
##   atemp_original hum_original windspeed_original
## 1         3.0014           81                  0
## 2         1.9982           80                  0
## 3         1.9982           80                  0
```

3. Problem definition

The big question we want to answer using this dataset is how can we predict the number of bikes rented at a certain date and hour, together with other variables such as weather conditions or the type of user.

In order to build the prediction model, we would need to explore and examine not only individual variables, but also the relationship among multiple variables. The result of the analysis will allow us to choose the most appropriate variables to build a model that would help us predict the number of bikes that will be rented.

Thus the rest of this report will follow the following structure:

4: variable analysis 5. statistical tests using variables 6. regression analysis

4. Variable analysis

In this section, we will look at the important variables of this datset, and examine the relationships among multiple variables.

```
names(bs)
```

```
##  [1] "date"              "season"            "yr"
##  [4] "mnth"              "hr"                "holiday"
##  [7] "weekday"           "workingday"        "weather"
## [10] "temp"              "atemp"             "hum"
## [13] "windspeed"         "casual"            "registered"
## [16] "total_bikes"       "temp_original"     "atemp_original"
## [19] "hum_original"      "windspeed_original"
```

4-1. Who are the users?

There are two type of users of Capital Bikeshare: registered users of the company who have membership, and casual users who borrow bikes for one time purposes.

```
sum(bs$casual) / sum(bs$total_bikes)
```

```
## [1] 0.1883017
```

```
sum(bs$registered) / sum(bs$total_bikes)
```

```
## [1] 0.8116983
```

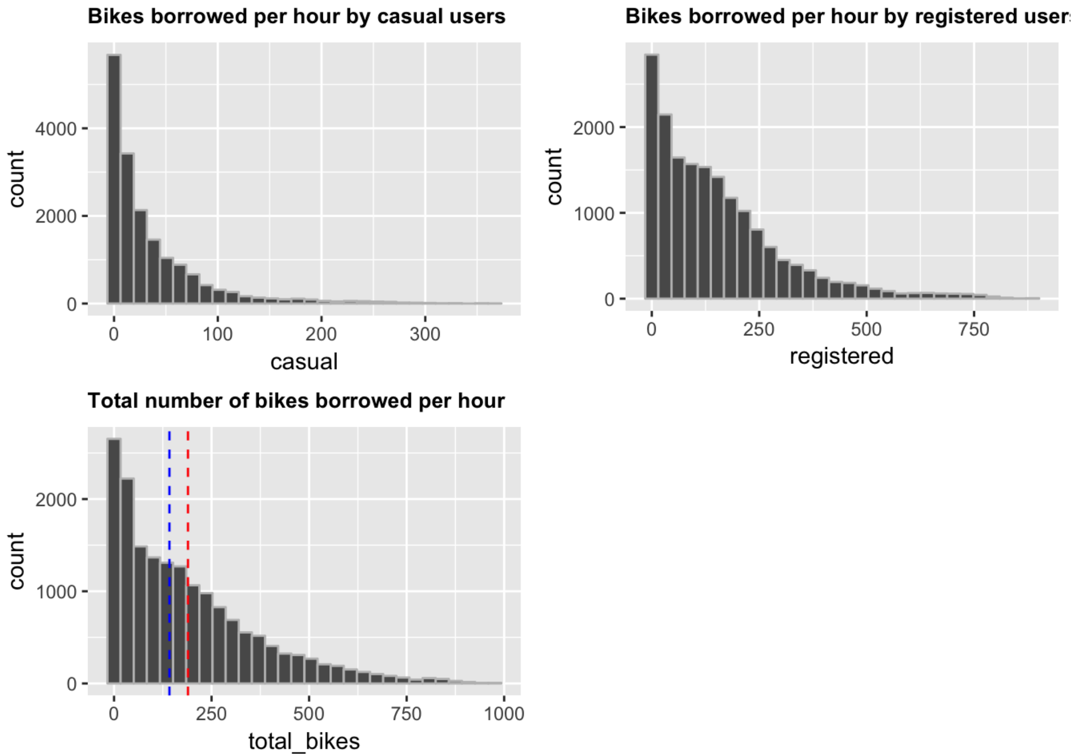
Around 81.2 % of the total bikes were borrowed by registered users, and the rest by casual users. There are a lot more registered users than there are casual users, which is true because not all casual users will be using this company’s bike only (other companies, own bike).

```
# distribution of total bikes per day
summary(bs$total_bikes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    40.0   142.0   189.5   281.0   977.0
```

```
grid.arrange(
  ggplot(bs, aes(casual)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Bikes borrowed per hour by casual users") +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(registered)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Bikes borrowed per hour by registered users")+
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(x = total_bikes)) +
    geom_histogram(color = I('gray')) +
    geom_vline(xintercept = mean(bs$total_bikes), color = I('red'), linetype = 2) +
    geom_vline(xintercept = median(bs$total_bikes), color = I('blue'), linetype = 2) +
    ggtitle("Total number of bikes borrowed per hour")+
    theme(plot.title = element_text(size = 10, face = "bold")),

  ncol = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



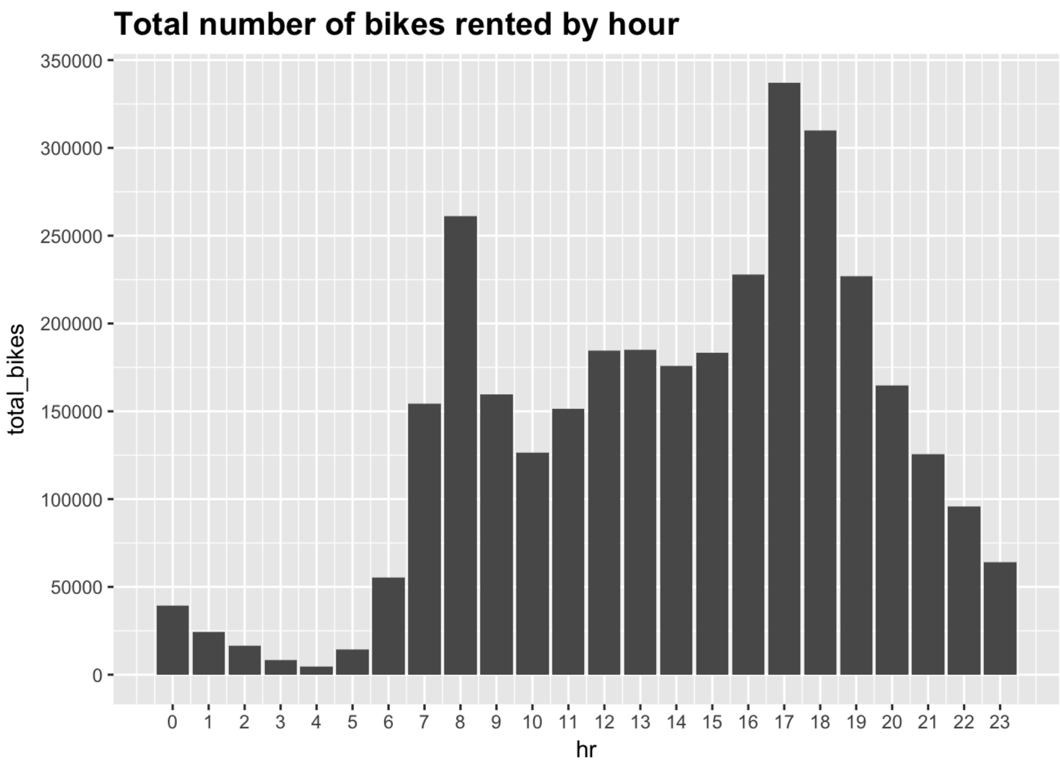
or both types of users, the number is skewed to the left, which makes sense because it would be really rare to have a lot of users (say 700) using the service at the same time. Because there are a lot more registered users, the distribution of total bikes resembles that of a registered user count more than it does the casual user count distribution. The mean number of rides per day 189.5 (red), but the median is 142 (blue), meaning that the number of ridership is skewed to the left, as seen in the histogram.

Because there are so many registered users compared to the casual users, the usage data related to registered would have a much bigger impact on the total usage data. Therefore, in order to ensure that the casual users' unique usage is not hidden by the mass registered users', we will need to look at the data separately between the two types of users for subsequent analysis.

4.2. When do people use bikes?

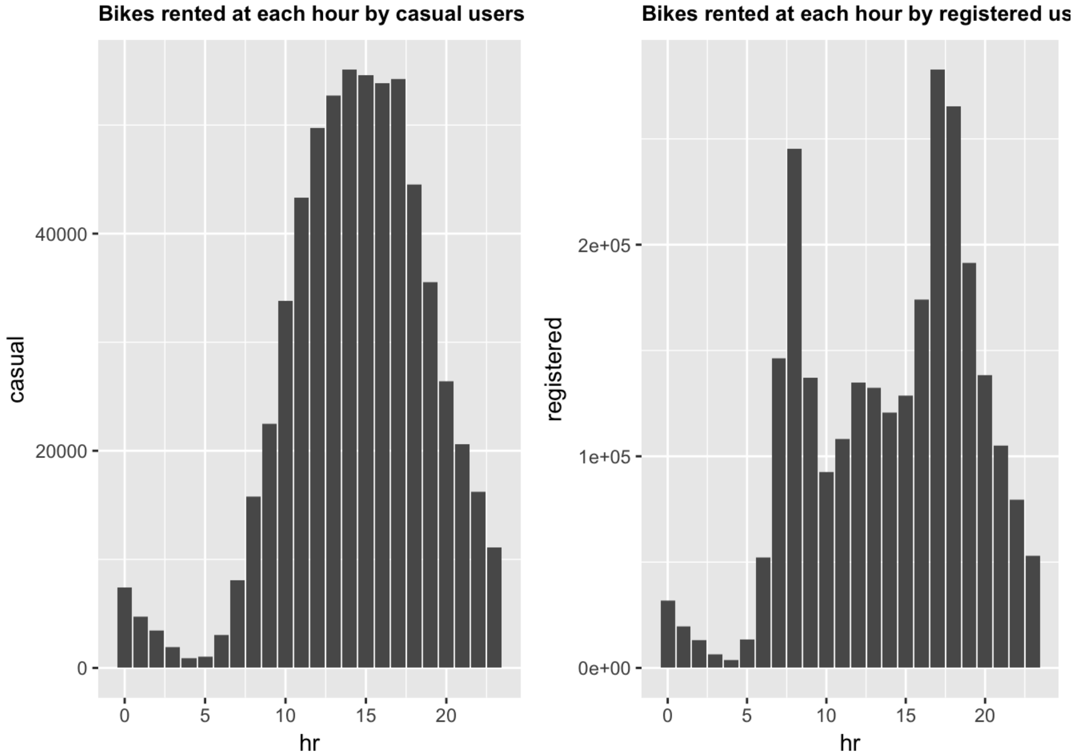
This dataset provides not only date data, but also hourly data, meaning that we can look at the bike usage pattern at different times of the day.

```
ggplot(bs, aes(x = hr, y = total_bikes)) +
  geom_col() +
  ggtitle("Total number of bikes rented by hour") +
  scale_x_continuous(breaks = seq(0,23,1)) +
  scale_y_continuous(breaks = seq(0,350000,50000)) +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



The number peaks at 9am and 5pm~6pm, which is the usual rush hour commute time. Let's look into this further and see if the usage pattern is same for both the registered and the casual users.

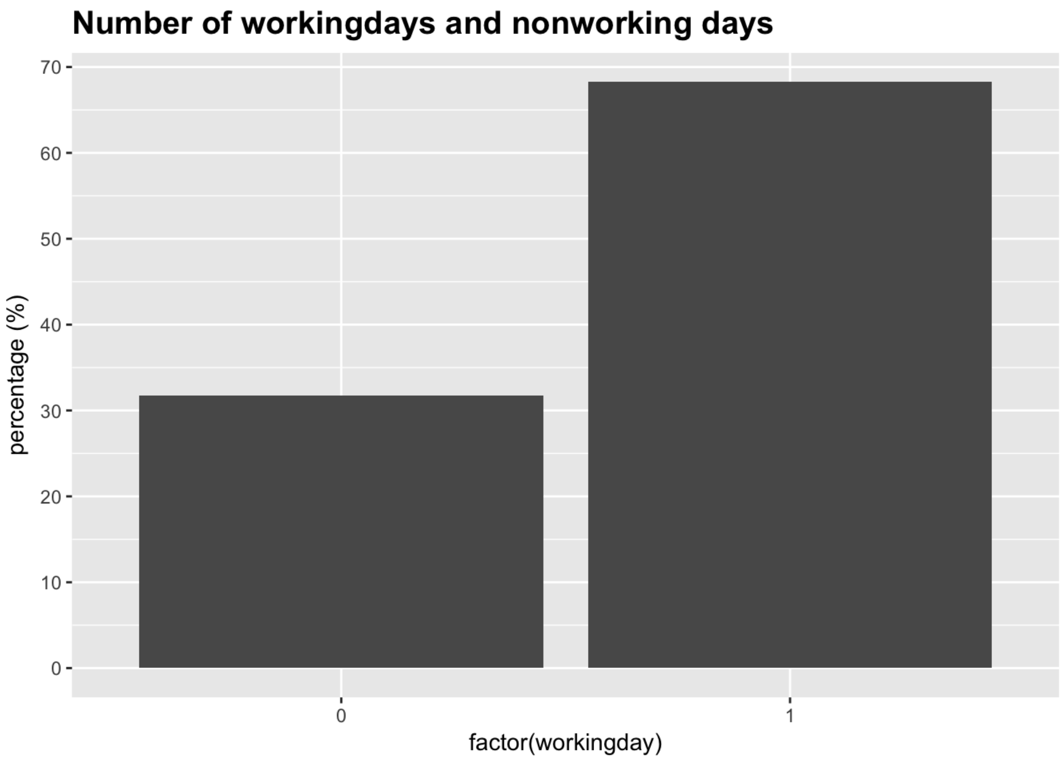
```
# hourly bike rent count for casual vs registered users
grid.arrange(
  ggplot(bs, aes(x = hr, y = casual)) +
    geom_col() +
    ggtitle("Bikes rented at each hour by casual users") +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(x = hr, y = registered)) +
    geom_col() +
    ggtitle("Bikes rented at each hour by registered users") +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ncol = 2
)
```



It is clear that there is a difference in the distribution of number of bikes rented per hour between the casual and regisetered users. This may be because the two types of users have different purposes when borrwing the bike. The peaks seen from total number of rentals are not visible from the casual users' distribution anymore. In fact, there, seems to be a single wide peak for the casual users, which is around the after noon from 12 to 5. This is clearly a working hour during a weekday, which raises another question: do casual and registered users ride primarily on different types of days (i.e., working days vs non working days)?

First, let's look at how many workingdays and non-working days (holidays and weekends) there are.

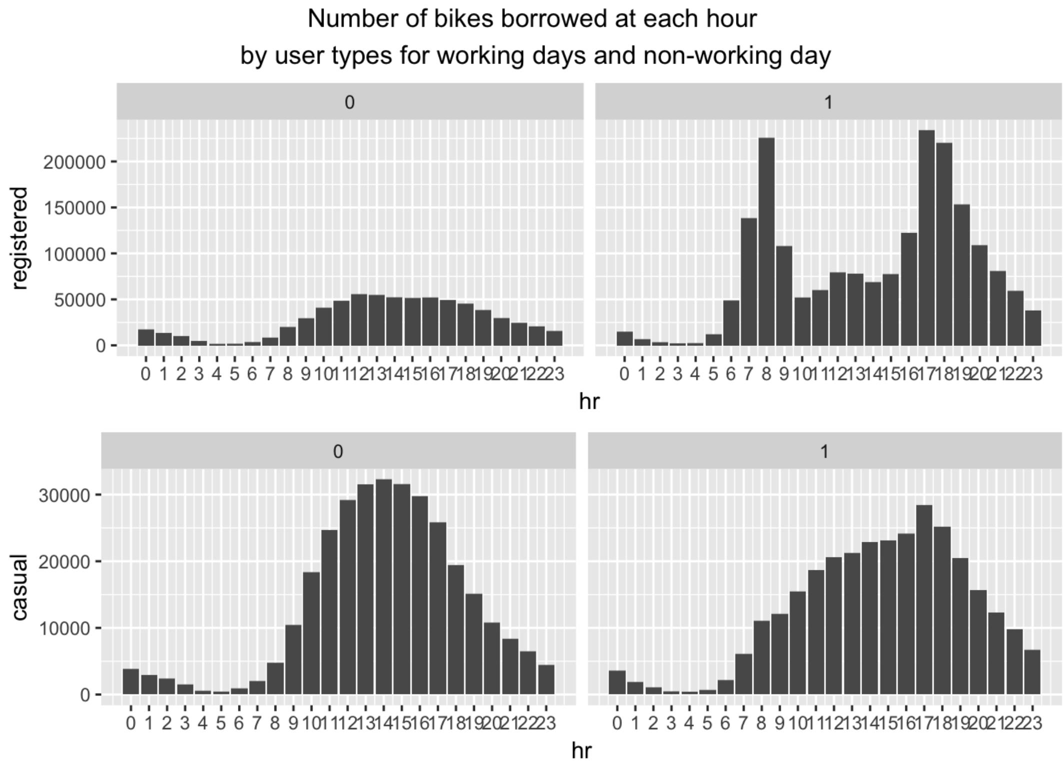
```
ggplot(bs, aes(x = factor(workingday))) +
  geom_bar(aes(y = ..count../sum(..count..) * 100)) +
  scale_y_continuous(breaks = seq(0,80,10)) +
  ggtitle("Number of workingdays and nonworking days") +
  theme(plot.title = element_text(size = 15, face = "bold")) +
  ylab("percentage (%)")
```



A little over 30% of the days are either weekends or holidays.

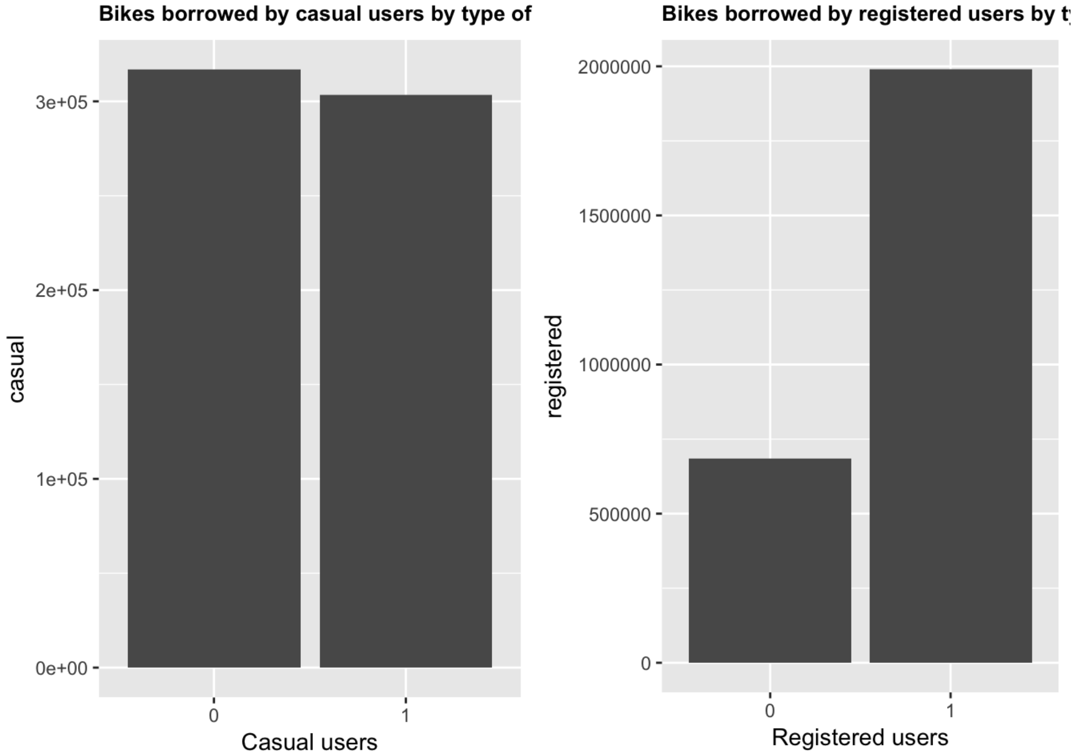
Now, let's look at how the number of bikes borrowed at each hour by the registered and casual users change during workingdays and non-working days.

```
grid.arrange(
  ggplot(bs, aes(x = hr, y = registered)) +
    geom_col() +
    scale_x_continuous(breaks = seq(0,23,1)) +
    facet_wrap(~factor(workingday)),

  ggplot(bs, aes(x = hr, y = casual)) +
  geom_col() +
  scale_x_continuous(breaks = seq(0,23,1)) +
  facet_wrap(~factor(workingday)),

  top = "Number of bikes borrowed at each hour \nby user types for working days and non-working day"
)
```



We can see that while the workingday vs non-working day factor has a huge impact on the hourly number of bikes borrowed, the impact is less evident for the casual users. In fact, judging from the graph, it seems that the number of bikes borrowed are similar for both working and non-working days when it comes to casual users.

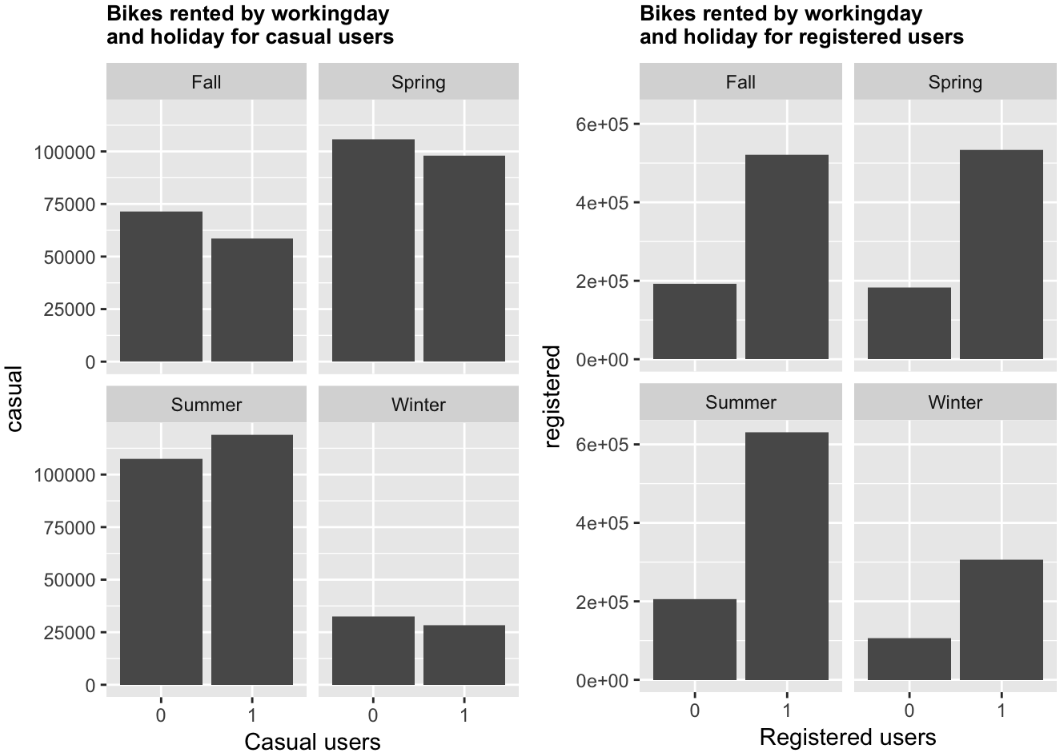
```
grid.arrange(
  ggplot(bs, aes(x = factor(workingday), y = casual)) +
    geom_col() +
    ggtitle("Bikes borrowed by casual users by type of day") +
    xlab("Casual users") +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(x = factor(workingday), y = registered)) +
    geom_col() +
    ggtitle("Bikes borrowed by registered users by type of day") +
    xlab("Registered users") +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ncol = 2
)
```



While the number of bikes rented for working days and non-working days indeed is similar for casual users (slightly more during non-working days), it is clear that for the registered users, they predominantly use the bikes on working days. This would explain why the two peaks in the hourly distribution were at the commute time: many registered users are using the bikes as a transporation method for commuting to and from work (or school). Since the usage pattern for the two groups are clearly different, this may mean that we might need a separate model to predict the number of bikes rented for each types of users.

Since a significant number of registered users use the bike as a transportation method, we would expect that the number of rideships will not vary too much by season. On the same note, because half of the casual users ride during non-work days (i.e., for leisure), there should be some difference in rideships depending on the season, as weather might be a more important consideration.

```
grid.arrange(
  ggplot(bs, aes(x = factor(workingday), y = casual)) +
    geom_col() +
    ggtitle("Bikes rented by workingday \nand holiday for casual users") +
    xlab("Casual users") +
    facet_wrap(~season) +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(x = factor(workingday), y = registered)) +
    geom_col() +
    ggtitle("Bikes rented by workingday \nand holiday for registered users") +
    xlab("Registered users") +
    facet_wrap(~season) +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ncol = 2
)
```



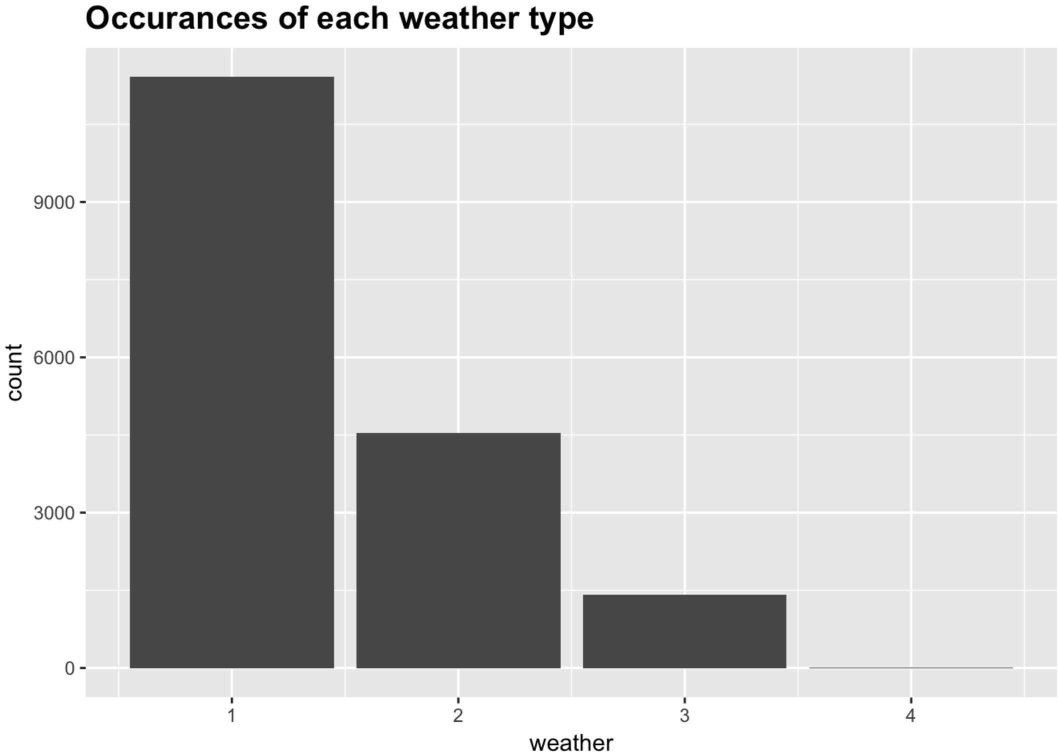
While the expectation seems to hold true for the registered group (i.e., rideships don't vary too much by season) except in the winter when the number of rideships decrease in general, there seems to be no big differene in rideships for working days and non-working days for the casual groups too for each season. A further hypothesis test should be conducted to see if the working day and season categories are independent from one another for the casual users.

4.3. How does the weather affect rideship?

Riding a bike is different from many other modes of transporation, as the rider is usually fully exposed to the environment during the ride. As such is the case, weather conditions, including the actual weather situation, temperature, humidity, and wind, are all important factors that may influence the number of bikes used (or borrwed in this case).

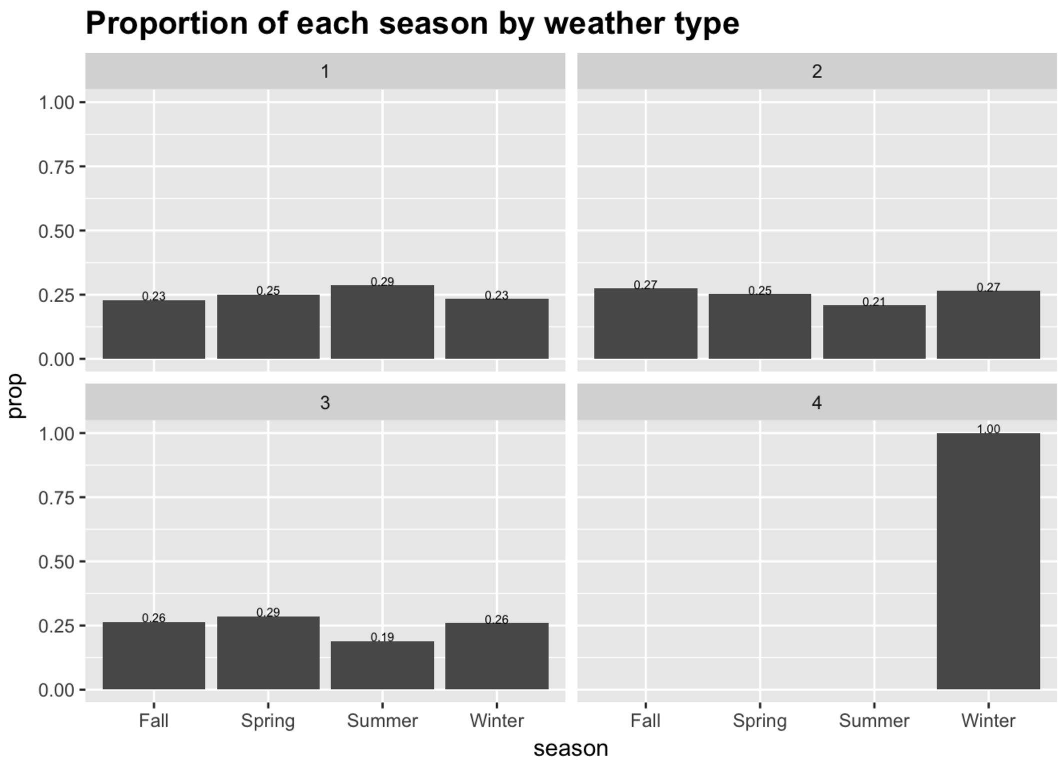
What weather was the most common in the dataset?

```
ggplot(bs, aes(x = weather))+
  geom_bar() +
  ggtitle("Occurances of each weather type") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



While milder weathers are the most common, the harshest weathers including a snow storm or thundertorm are far less common in the dataset. Are these weather patterns evenly seen in all seasons?

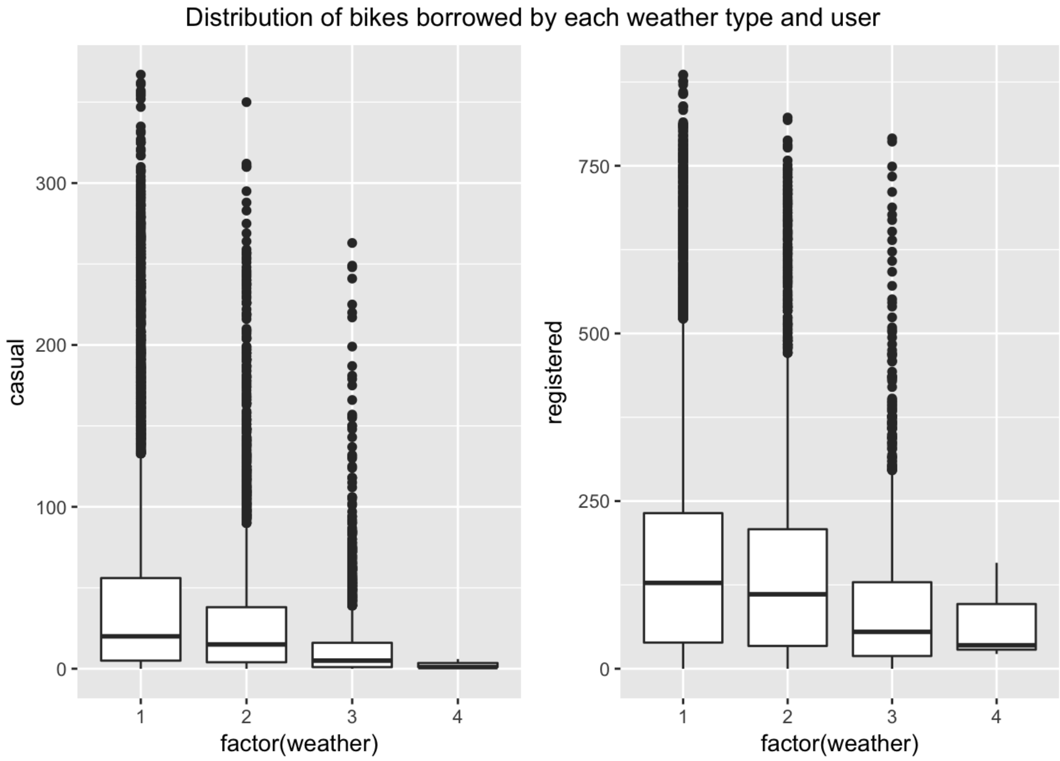
```
ggplot(bs, aes(x = season,
               y = ..prop.., group = 1)) +
  stat_count(show.legend = F) +
  facet_wrap(~factor(weather)) +
  geom_text(stat = 'count',
            aes(label = sprintf("%0.2f",
                                round(..prop.., digits = 2))),
            vjust = 0,
            size = 2) +

  ggtitle("Proportion of each season by weather type") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



We can see that the harshest weather only occur in Winter, and though the number of harsh weather (4) is small, this may have an impact on the average number of daily ridership for Winter.

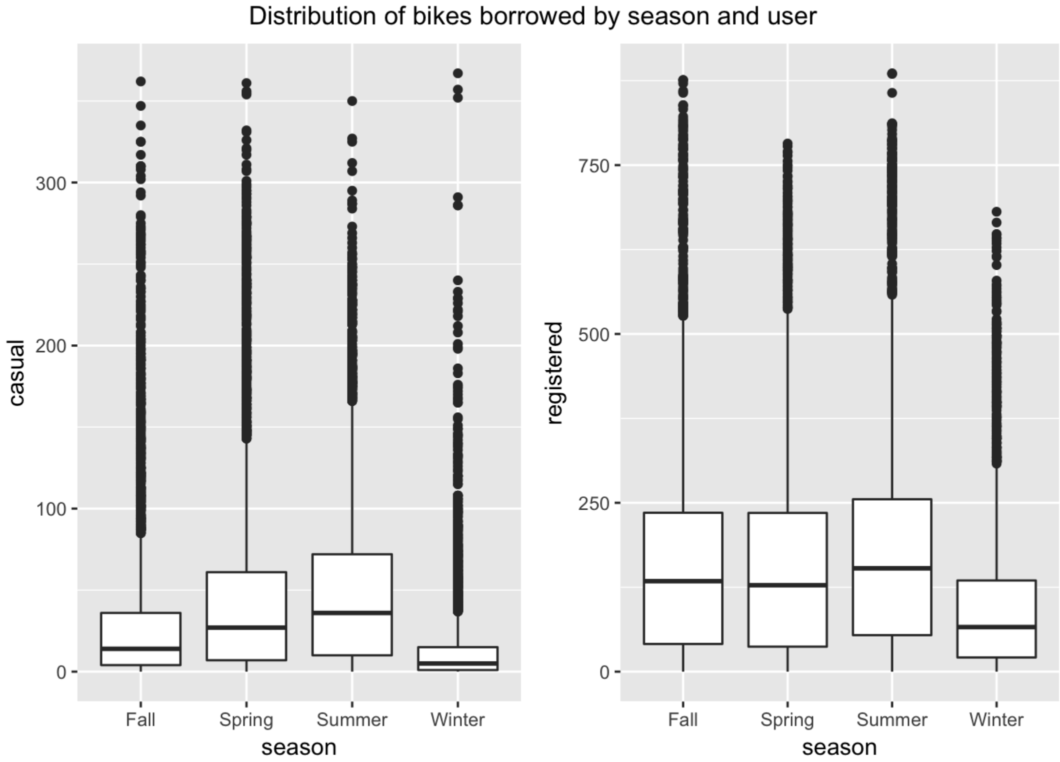
Would each of these weather have an impact on the number of bikes borrowed by each user?

```
grid.arrange(
  ggplot(bs, aes(y = casual, x = factor(weather))) +
    geom_boxplot(),

  ggplot(bs, aes(y = registered, x = factor(weather))) +
    geom_boxplot(),

  ncol = 2,
  top = "Distribution of bikes borrowed by each weather type and user"
)
```



Number of total bike rented decreases as the weather gets harsher, which is predictable. It is however interesting to see that there are outliers in all weather conditions (except the harshest 4), meaning that there are significant number of people who use bikes regardless of some weather changes. Would we see similar patterns for each season?

```
grid.arrange(
  ggplot(bs, aes(y = casual, x = season)) +
    geom_boxplot(),

  ggplot(bs, aes(y = registered, x = season)) +
    geom_boxplot(),

  ncol = 2,
  top = "Distribution of bikes borrowed by season and user"
)
```



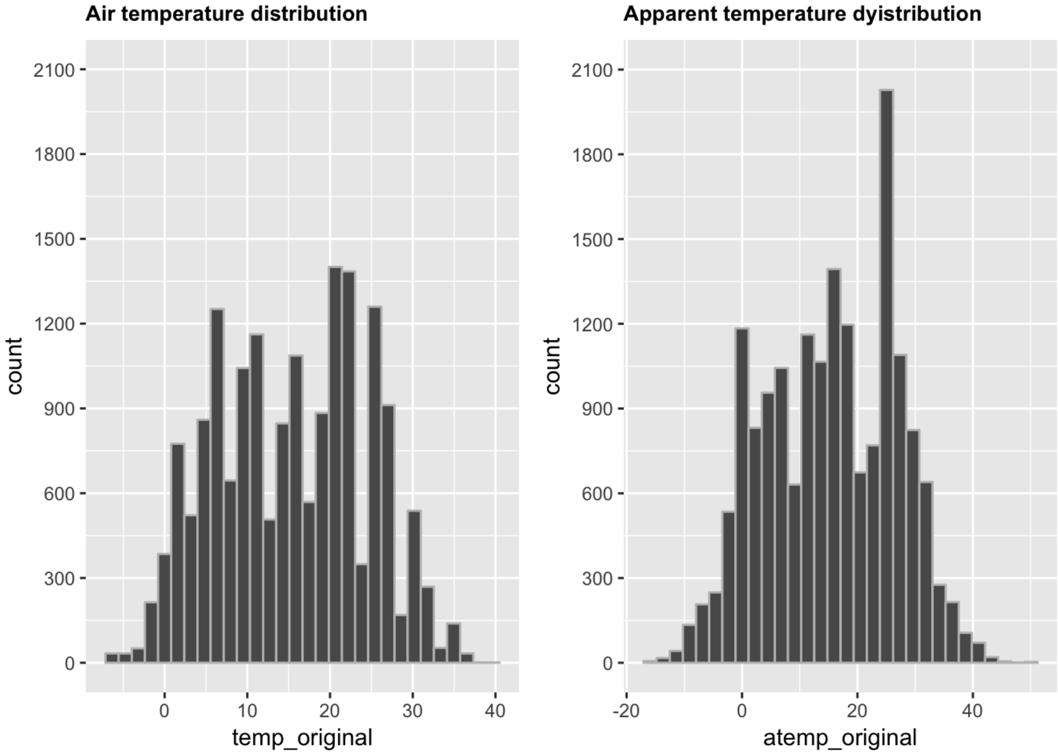
Unsurprisingly, average daily rideship decreases during winter, possibly due to factor such as weather condition or temperature. It is interesting to see that there are outliers in the top in all seasons, meaning that there are some people who ride bikes regardless of the season. These people may be riding bikes not for leisure, but for transportation means. Furthermore, this observation further confirms that registered users are less impacted by weather factors such as season and weather situation, as they use bikes for transporation means.

While seasons and weather situations are important to look at, those are aggregated data of different days with different temperatures, humidity, and windspeed. Let's look at the specific weather conditions.

There are two types of temperature given in this datset - the normal air temperature, and the apparent temperature, which is the temperature actually perceived by humans, accounting for other weather conditions such as humidity and windspeed. A natural question therefore would be whether or not air temperature and apparent temperature similar.

```
grid.arrange(
  ggplot(bs, aes(temp_original)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Air temperature distribution") +
    scale_y_continuous(breaks = seq(0,2100,300), limits = c(0,2100)) +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ggplot(bs, aes(atemp_original)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Apparent temperature dyistribution") +
    scale_y_continuous(breaks = seq(0,2100,300), limits = c(0,2100)) +
    theme(plot.title = element_text(size = 10, face = "bold")),

  ncol = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The overall shape seems to be similar except for the peark around 27 degrees in atemp_original. Since apparent temperature is affected by not only temperature but also other weather conditions such as humidity and wind speed, this may explain why the distribution is not exactly equal to one another.

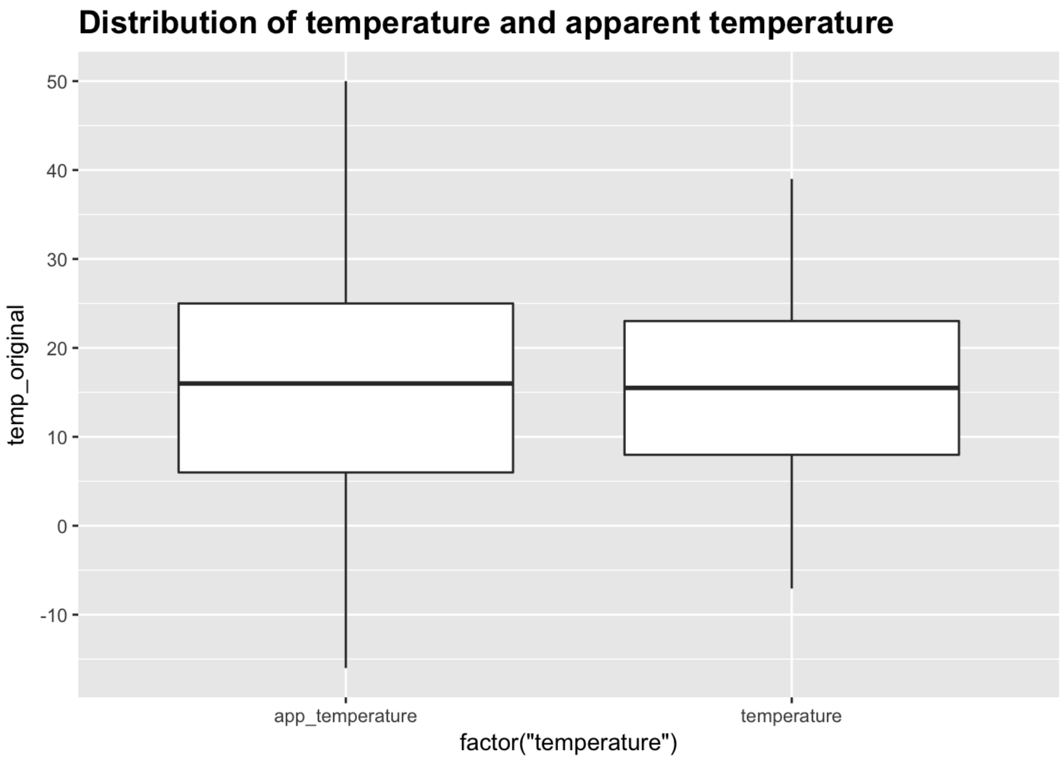
```
summary(bs$atemp_original)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -16.000   5.998  15.997  15.401  24.999  50.000
```

```
summary(bs$temp_original)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -7.06    7.98   15.50   15.36   23.02   39.00
```

```
ggplot(bs) +
  geom_boxplot(aes(x = factor('temperature'),y = temp_original)) +
  geom_boxplot(aes(x = factor('app_temperature'),y = atemp_original)) +
  scale_y_continuous(breaks = seq(-20,50,10)) +
  ggtitle("Distribution of temperature and apparent temperature") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```

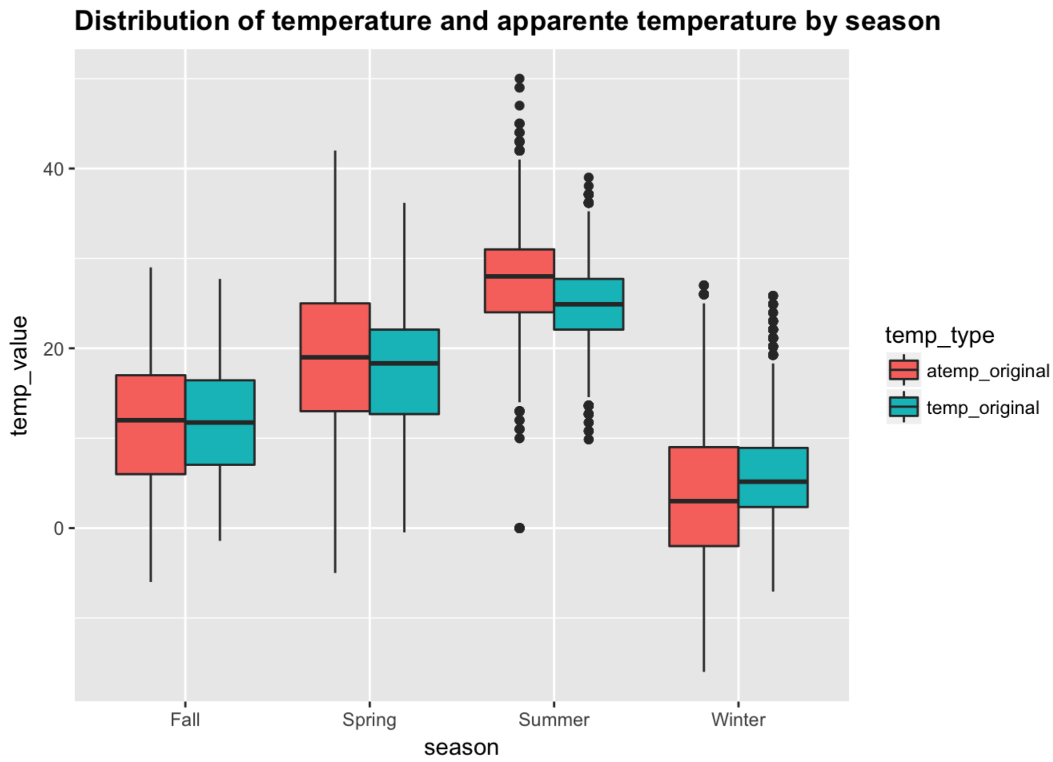


It also seems that the apparent temperature is more sparsely distributed than temperature, as shown by the larger IQR box height and the min and max whisker. The median, however, is more or less smiliar around 15 ~ 16 degrees. Let's look at this further by each season, where temperature should differ significantly.

```
# data needs to be gathered to achieve this

bs_melt <- bs %>%
  select(temp_original, atemp_original,season) %>%
  gather(key = temp_type, value = temp_value, -season)

head(bs_melt)
```

```
##   season     temp_type temp_value
## 1 Winter temp_original       3.28
## 2 Winter temp_original       2.34
## 3 Winter temp_original       2.34
## 4 Winter temp_original       3.28
## 5 Winter temp_original       3.28
## 6 Winter temp_original       3.28
```

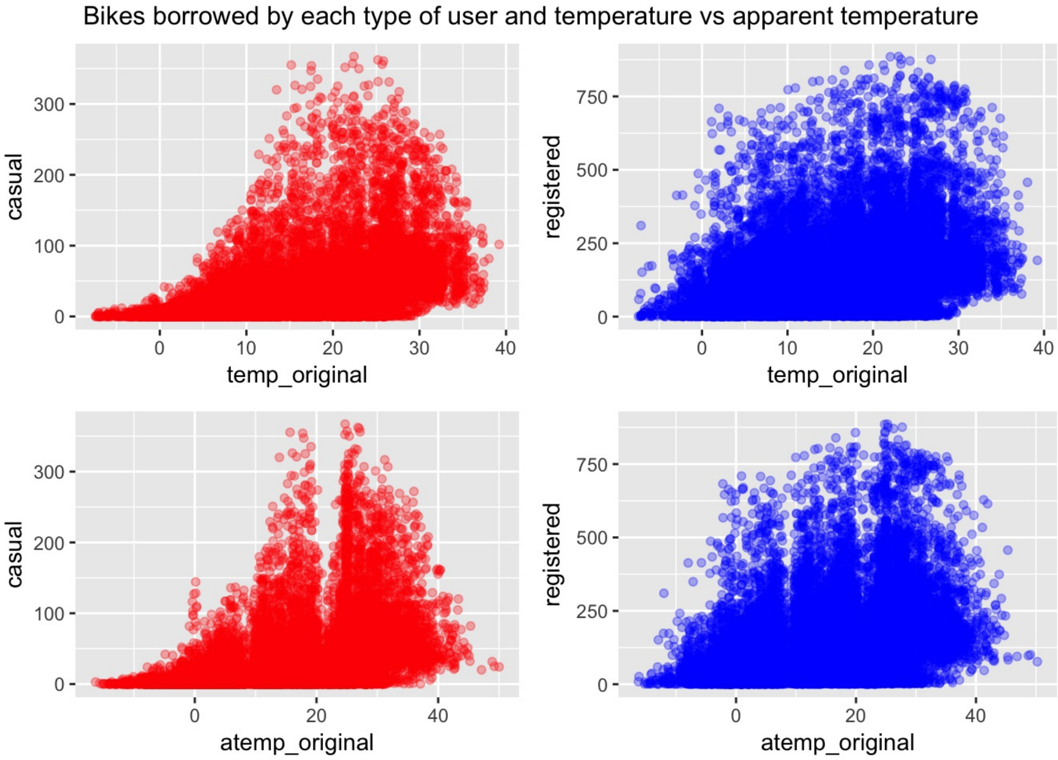
```
# boxplot comparing temperature for each season
ggplot(bs_melt, aes(x = season, y = temp_value, fill = temp_type)) +
  geom_boxplot() +
  ggtitle("Distribution of temperature and apparente temperature by season") +
  theme(plot.title = element_text(size = 13, face = "bold"))
```



It is interesting to see that the apparent temperature is often more extreme than the actual air temperature. For example, the apparent temperature is higher than the actual air temperature during summer, while it is on average lower than the actual temperature during winter.

The important question is then, does the ridership differ by each temperature for each users?

```
# does the ridership differ for each type of user by temperature?
grid.arrange(
  ggplot(bs, aes(x = temp_original)) +
  geom_point(aes(y = casual),
        color = I('red'),
        alpha = 0.3, position = 'jitter'),

 ggplot(bs, aes(x = temp_original)) +
  geom_point(aes(y = registered),
        color = I('blue'),
        alpha = 0.3, position = 'jitter'),

 ggplot(bs, aes(x = atemp_original)) +
  geom_point(aes(y = casual),
        color = I('red'),
        alpha = 0.3, position = 'jitter'),

 ggplot(bs, aes(x = atemp_original)) +
  geom_point(aes(y = registered),
        color = I('blue'),
        alpha = 0.3, position = 'jitter'),

 ncol = 2,

 top = "Bikes borrowed by each type of user and temperature vs apparent temperature"
)
```



Interestingly, there doesn't seem to be a big pattern, other than the fact that there seems to be some divison in different points of apparent temperature for both the casual (around 20 degrees) and the registered (around 10 and 20 degrees) users.

Let's look at humidity and windspeed next. How are humidity and windspeed distributed in the dataset?

```
summary(bs$hum_original)
```

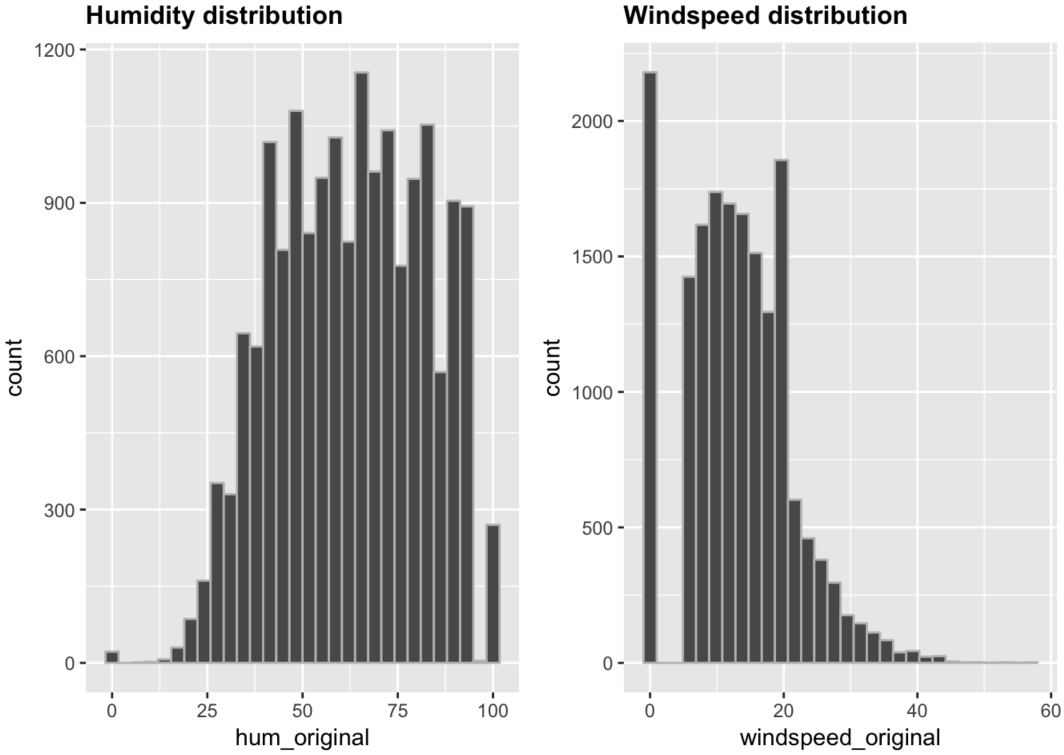
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   48.00   63.00   62.72   78.00  100.00
```

```
summary(bs$windspeed_original)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   7.002  12.998  12.737  16.998  56.997
```

```
grid.arrange(
  ggplot(bs, aes(hum_original)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Humidity distribution") +
    theme(plot.title = element_text(size = 12, face = "bold")),

  ggplot(bs, aes(windspeed_original)) +
    geom_histogram(color = I('gray')) +
    ggtitle("Windspeed distribution") +
    theme(plot.title = element_text(size = 12, face = "bold")),

  ncol = 2
)
```

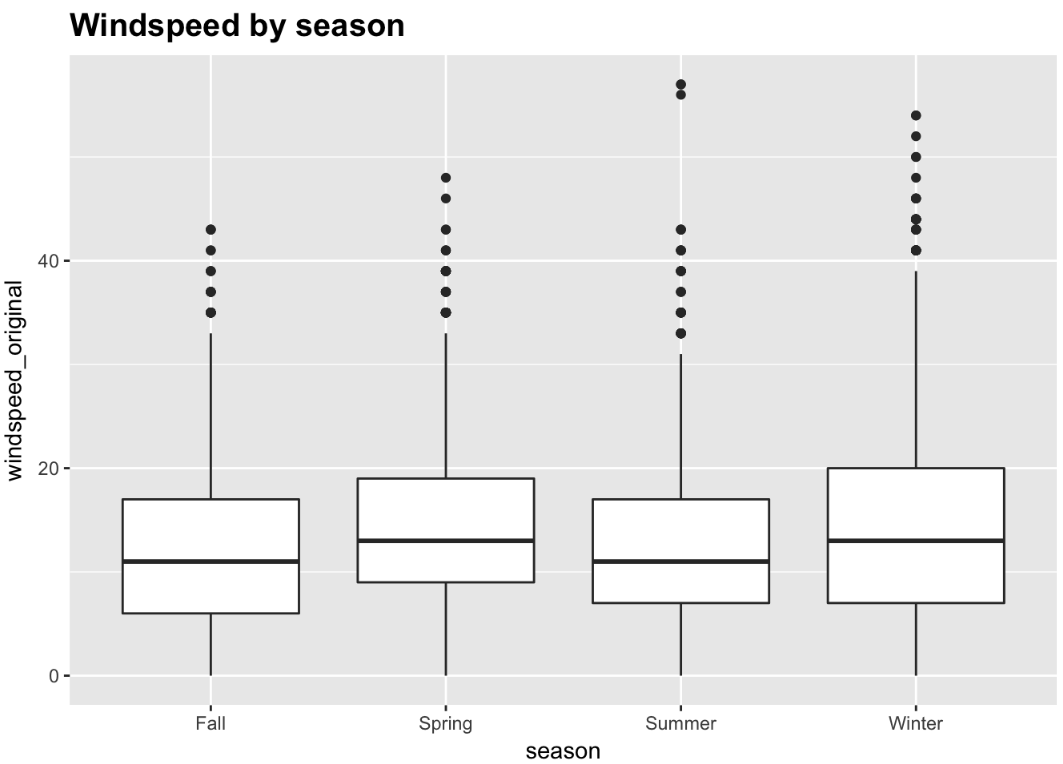
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



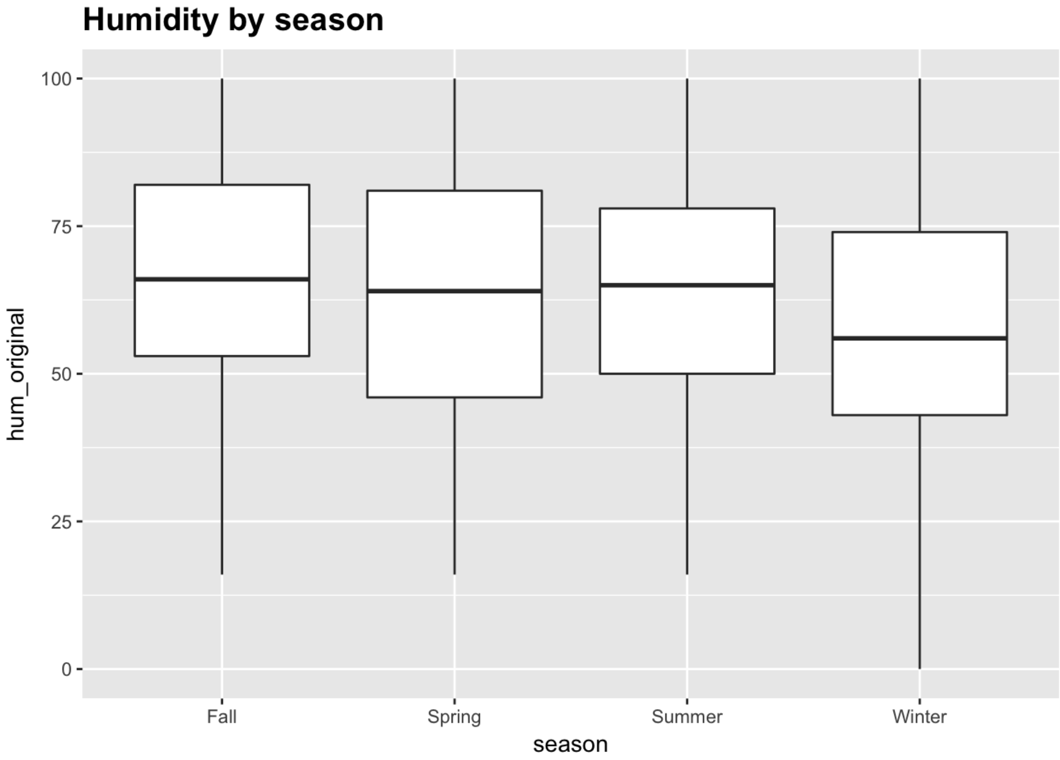
Both the humidity and windspeed seemed to be somewhat skewed, especially the windspeed. According to the Beaufort wind force scale classication (https://en.wikipedia.org/wiki/Beaufort_scale), wind speed between 20 to 28 is considered a moderate breeze, between 29 to 38 fresh breeze, between 39 to 49 strong breeze, and between 50 to 61 high wind (moderate gale). This means that most of the days, the windspeed was lower than a moderate breeze, and that there were few days with very strong winds that may affect bike ridership.

Does the windspeed and humidity differ by each season as do the temperature?

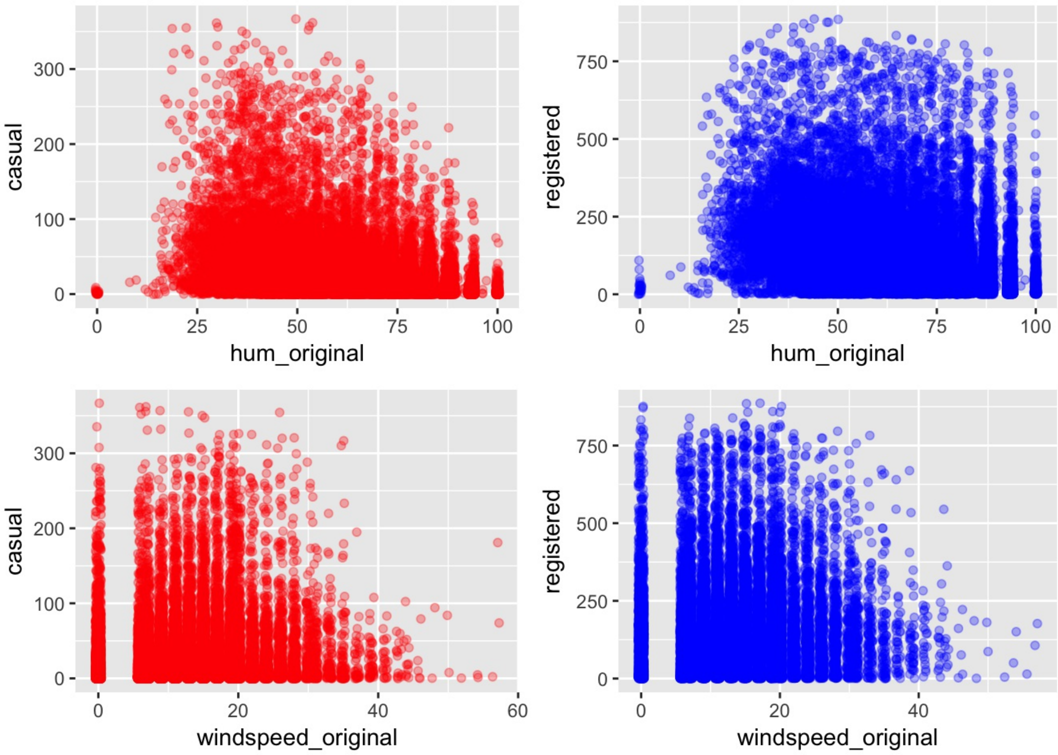
```
# windspeed for each season
ggplot(bs, aes(x = season, y = windspeed_original)) +
  geom_boxplot() +
  ggtitle("Windspeed by season") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```



```r
# humidity for each season
ggplot(bs, aes(x = season, y = hum_original)) +
  geom_boxplot() +
  ggtitle("Humidity by season") +
  theme(plot.title = element_text(size = 15, face = "bold"))
```




```r
# humidity and windspeed?
grid.arrange(
    ggplot(bs, aes(x = hum_original)) +
    geom_point(aes(y = casual),
            color = I('red'),
            alpha = 0.3, position = 'jitter'),

  ggplot(bs, aes(x = hum_original)) +
    geom_point(aes(y = registered),
            color = I('blue'),
            alpha = 0.3, position = 'jitter'),

  ggplot(bs, aes(x = windspeed_original)) +
    geom_point(aes(y = casual),
            color = I('red'),
            alpha = 0.3, position = 'jitter'),

  ggplot(bs, aes(x = windspeed_original)) +
    geom_point(aes(y = registered),
            color = I('blue'),
            alpha = 0.3, position = 'jitter'),

  ncol = 2
)
```



conclusion: Variable analysis conclusion

While weather conditions all have impact on ridership in general, it is also true that the level of impact differs for each type of user. In the statistical test, it would be interesting to further this observation and see if the differences are significant.

5. Statistical tests

5.1 Is there a significant difference between the actual air temperature and the apparent temperature perceived by humans in terms of number of rideships?

```
# two-tailed, 2 independent variables t-test, 95% confidence level
t.test(bs$temp_original, bs$atemp_original, alternative = 'two.sided', paired = T, mu = 0)
```

```
##
##  Paired t-test
##
## data:  bs$temp_original and bs$atemp_original
## t = -2.0204, df = 17378, p-value = 0.04335
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.084242583 -0.001277067
## sample estimates:
## mean of the differences
##             -0.04275983
```

The p-value is 0.043, which means that there is enough evidence to prove that the air temperature and the temperature perceived by humans differ significantly.

5.2 Does season and workingday have an effect on rideships for each type of users? If so, do the two effects interact?

```
summary(aov(data = bs, casual ~ season + factor(workingday) + season:factor(workingday)))
```

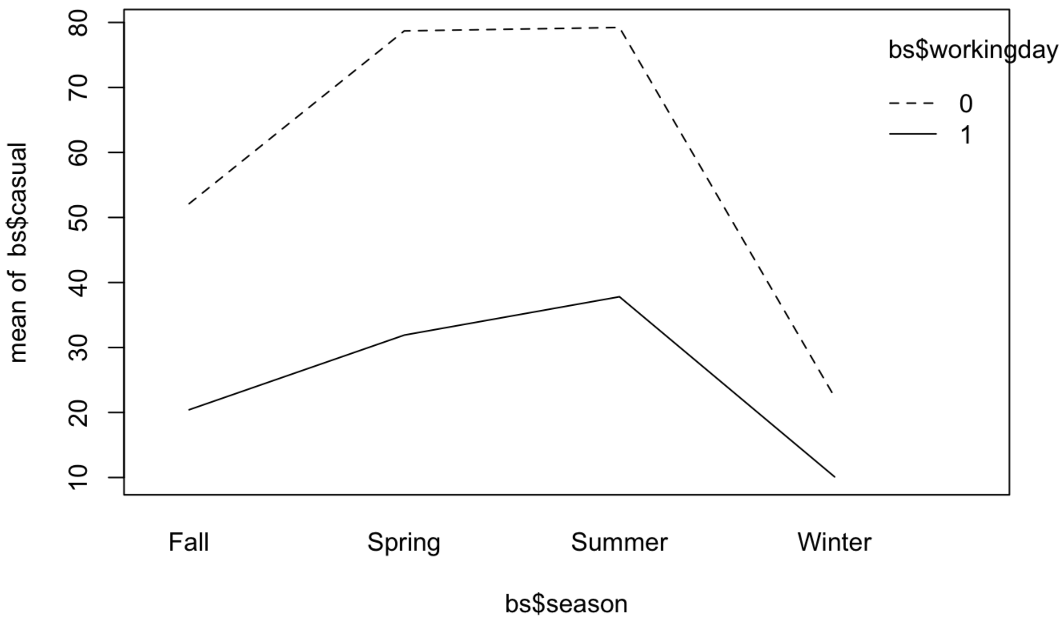
```
##                            Df   Sum Sq Mean Sq F value Pr(>F)
## season                      3  3490647 1163549   594.2 <2e-16 ***
## factor(workingday)          1  4086243 4086243  2086.9 <2e-16 ***
## season:factor(workingday)   3   655924  218641   111.7 <2e-16 ***
## Residuals               17371 34012861    1958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(data = bs, registered ~ season + factor(workingday) + season:factor(workingday)))
```

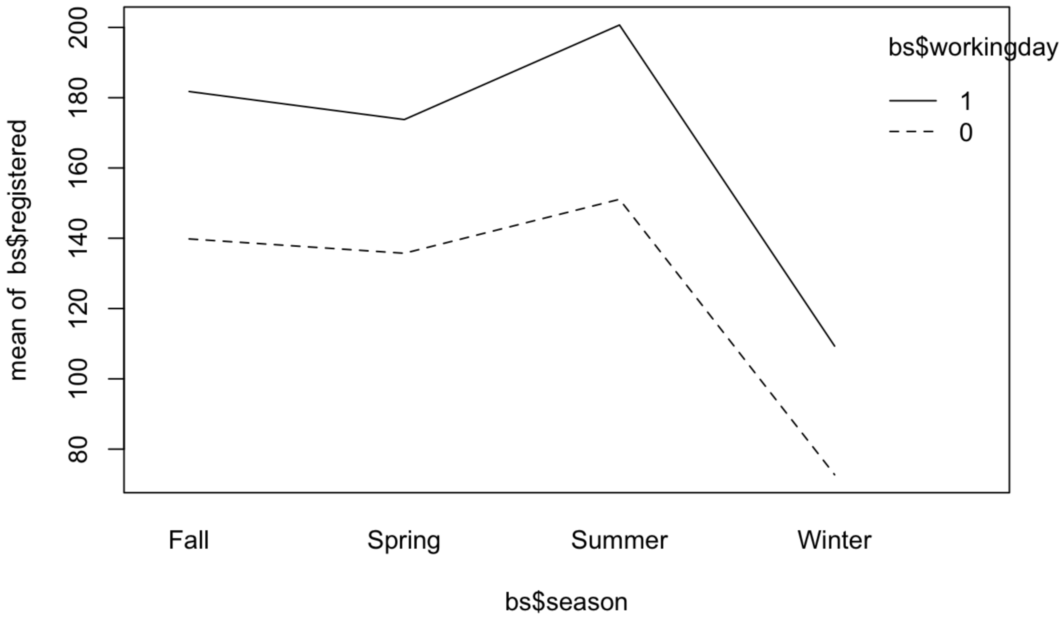
```
##                            Df    Sum Sq Mean Sq F value Pr(>F)
## season                      3  19542008 6514003 304.194 <2e-16 ***
## factor(workingday)          1   6492104 6492104 303.171 <2e-16 ***
## season:factor(workingday)   3     96220   32073   1.498  0.213
## Residuals               17371 371982758   21414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This confirms the hypothesis formed during the multivariate analysis. While season and workingday both have an impact on the number of bikes rented for both types of users, the interaction effect between the two variables don't exist for registered users, while it does for the causal users. Again, this is because registered users, who mostly use the bikes for commute purposes, are less impacted by seasonal factors than are casual users, half of whom ride bikes for leisure. We can also confirm the interaction effect visually:

```
interaction.plot(bs$season, bs$workingday, bs$casual)
```




```
interaction.plot(bs$season, bs$workingday, bs$registered)
```



From the interaction plot, it is clear that season has an impact on ridership for casual users. More specifically, the number of bikes rented for non-working days drop more drastically in Winter than it does for working days. On the contrast, the interaction plot for the registered users show that the patterns of total number of bikes are similar (if not the same), regardless of the season.

5.3 Does the time of the day has a significant impact on ridership?

H0: No difference in ridership with time of the day H1: There is some difference For this purpose we create a categorical variable from hr - hr_cat: 1. Late Night 2. Early Morning 3. Afternoon 4. Evening/Night

```
bs= bs %>%
  mutate(hr_cat=ifelse(hr>=0 & hr<=5,"1",
                    ifelse(hr>=6 & hr <=11,2,
                        ifelse(hr>=12 & hr<=17,3,4))))
a1=aov(data = bs, total_bikes ~ hr_cat)
summary(a1)
```

```
##                Df    Sum Sq  Mean Sq F value Pr(>F)
## hr_cat          3 172231550 57410517    2497 <2e-16 ***
## Residuals   17375 399530041    22995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

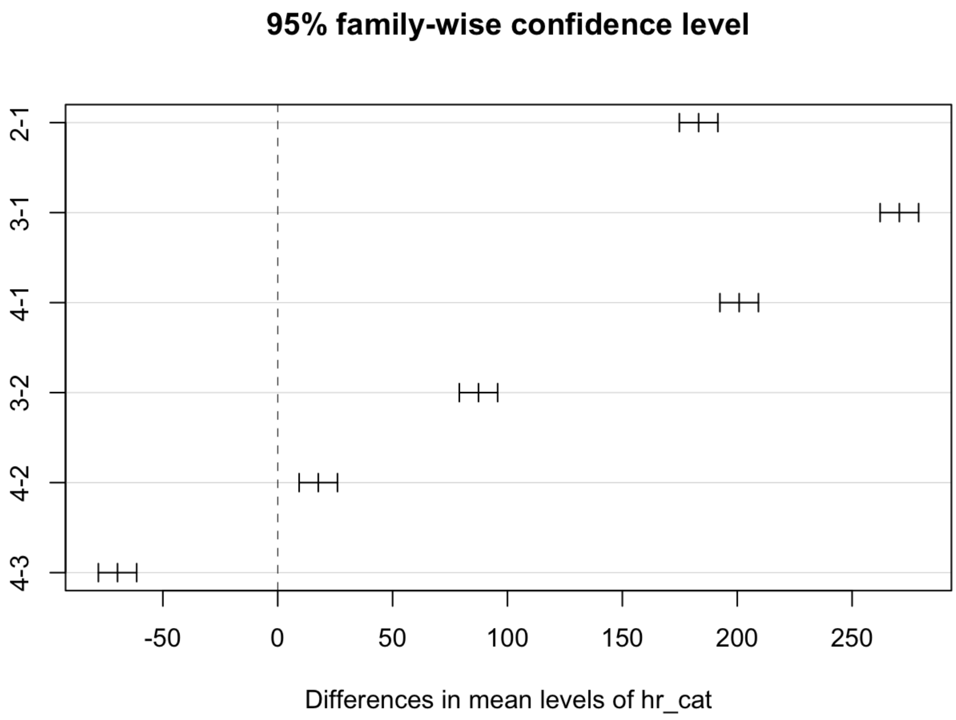
Based on the above result (p-value<0.05) with 95% confidence, we reject H0 and conclude that there is some difference.

Now lets find out when the ridership is the highest.

```
a2=TukeyHSD(a1)
print(a2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = total_bikes ~ hr_cat, data = bs)
##
## $hr_cat
##          diff        lwr       upr p adj
## 2-1 183.19213 174.806839 191.57742 0e+00
## 3-1 270.57533 262.197157 278.95350 0e+00
## 4-1 200.84900 192.467509 209.23048 0e+00
## 3-2  87.38320  79.045944  95.72045 0e+00
## 4-2  17.65687   9.316279  25.99745 4e-07
## 4-3 -69.72633 -78.059760 -61.39290 0e+00
```

```
plot(a2)
```



Looking at the above chart we can rank the demand. $ Ridership in Afternoon> Evening/ Night > Early Morning> Late Night$ 1. Ridership is the highest in Afternoon 2. Ridership is the lowest in Late Night

5.4 Does ridership depend on type of day?

Lets define three types of days: 1. Holiday 2. Working day 3. Weekend

H0: No difference in ridership with type of day H1: There is some difference

```
bs= bs %>%
  mutate(typeofday = ifelse(holiday==0 & workingday==0,"Weekend",
                         ifelse(holiday ==1, "Holiday", "Working Day" )))
b1 = aov(data=bs, total_bikes ~ typeofday )
summary(b1)
```

```
##                Df    Sum Sq Mean Sq F value   Pr(>F)
## typeofday       2    855393  427697   13.02 2.24e-06 ***
## Residuals   17376 570906198   32856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

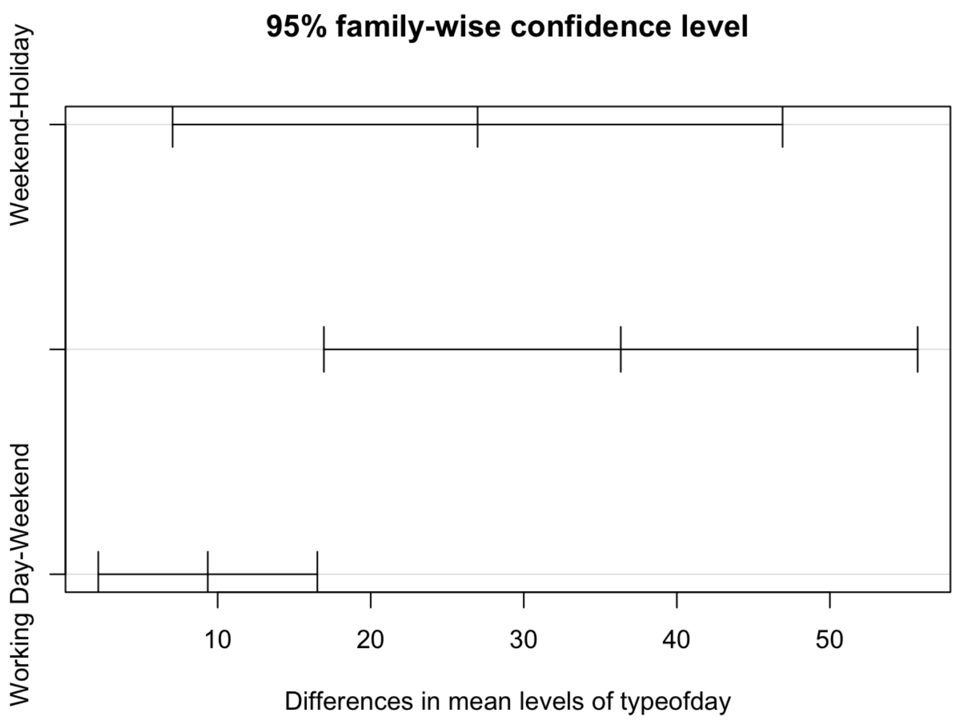
Since the p-value is less than .05 we can reject H0 and conclude that the rideship is different during different types of days.

Now again lets find out on which type of days the ridership is higher.

```
b2= TukeyHSD(b1)
print(b2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = total_bikes ~ typeofday, data = bs)
##
## $typeofday
##                          diff       lwr      upr     p adj
## Weekend-Holiday      26.98201  7.056793 46.90724 0.0042972
## Working Day-Holiday 36.33775 16.941175 55.73433 0.0000338
## Working Day-Weekend  9.35574  2.199345 16.51213 0.0061887
```

```
plot(b2)
```



Conclusion: Taking a look at the above table and the plot we can conclude that - RidershiponWorkingDay > RidershiponWeekend > RidershiponHoliday

5.5 Does rideship depend on the weather conditions?

Lets see if the type f weather has an impact on ridership.

H0: Type of weather has no impact on riderwhip H1: There is some impact

```
c1 = aov(data=bs, total_bikes ~ factor(weather, levels = c(1,2,3)))
summary (c1)
```

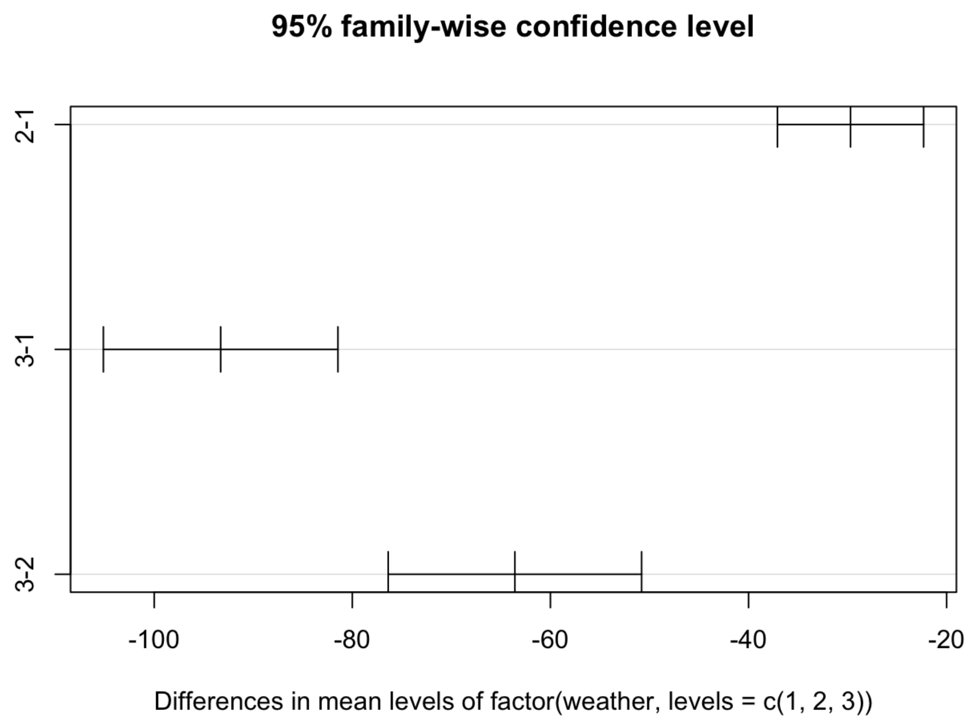
```
##                                  Df    Sum Sq Mean Sq F value
## factor(weather, levels = c(1, 2, 3))     2  12245259 6122629   190.1
## Residuals                          17373 559464416   32203
##                                  Pr(>F)
## factor(weather, levels = c(1, 2, 3)) <2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness
```

Since the p-value is <0.05 we cab conclude that the ridership depends on the weather. Now lets find out how the weather affects the ridership.

```
c2 = TukeyHSD(c1)
print(c2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = total_bikes ~ factor(weather, levels = c(1, 2, 3)), data = bs)
##
## $`factor(weather, levels = c(1, 2, 3))`
##          diff        lwr       upr p adj
## 2-1 -29.70378  -37.08188 -22.32567     0
## 3-1 -93.28999 -105.12979 -81.45019     0
## 3-2 -63.58621  -76.37738 -50.79504     0
```

```
plot(c2)
```



Looking at the above data we can conclude that: $Ridership in Clear weather > Mist > Light Snow/rain $

6. Regression analysis

We can use the variables we have explored to predict how many bikes will be borrowed in a given hour and day?

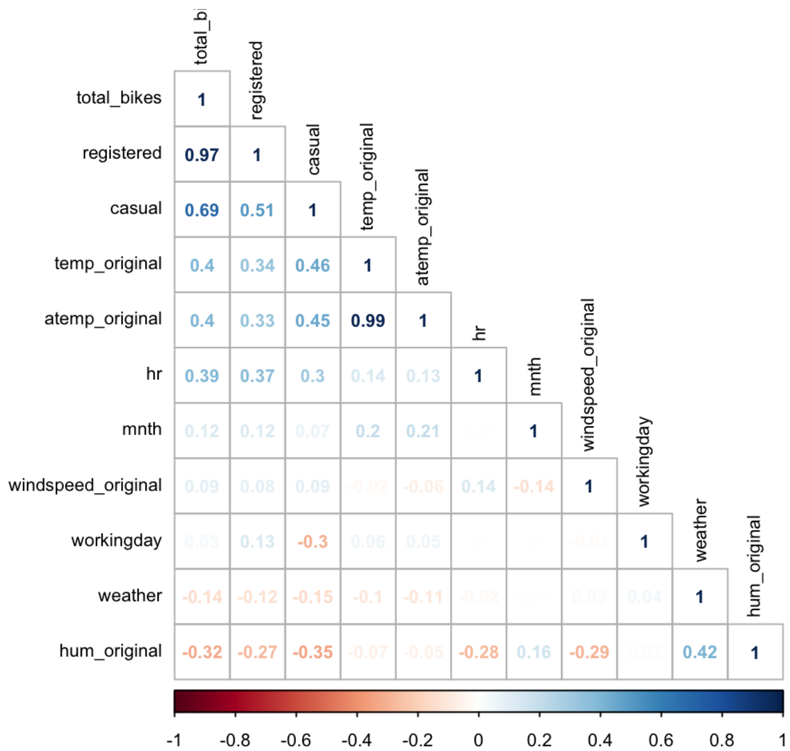
6.1 Checking linear relation between independent and dependent variables

we can run a correlation matrix for some of the numerical variables, to check beforehand if any of them are highly correlated.

```
aux= bs %>%
  dplyr::select(-date,-season, -yr, -holiday, -weekday, -temp, -atemp, -hum, -windspeed, -hr_cat, -typeofday)
head(aux)
```

```
##   mnth hr workingday weather casual registered total_bikes temp_original
## 1    1  0          0       1      3         13          16          3.28
## 2    1  1          0       1      8         32          40          2.34
## 3    1  2          0       1      5         27          32          2.34
## 4    1  3          0       1      3         10          13          3.28
## 5    1  4          0       1      0          1           1          3.28
## 6    1  5          0       2      0          1           1          3.28
##   atemp_original hum_original windspeed_original
## 1         3.0014           81             0.0000
## 2         1.9982           80             0.0000
## 3         1.9982           80             0.0000
## 4         3.0014           75             0.0000
## 5         3.0014           75             0.0000
## 6         1.0016           75             6.0032
```

```
caux=cor(aux)
corrplot(caux, method="number", order="FPC", type="lower",tl.col="black", tl.cex=0.7, number.cex=0.7, cl.cex=0.7)
```



Based on the matrix only, we don't see huge problems of corrleation between potential independent variables.

6.2 Selecting variables for the model

We can use the step-wise method to choose the most significant variables for our linear regression model.

```
# filter out unwanted
bs2 = bs%>%
  filter(yr == 1) %>% # to reduce the impact of time correlation from different years data, we only select one of the two years.
  dplyr::select(-registered,-date,-yr,-mnth, -weekday, -hr, -temp, -atemp, -hum, -windspeed, -typeofday, -temp_original)

null= lm(data=bs2, total_bikes ~ 1)  # empty model
full = lm(data=bs2, total_bikes ~ .) # full model

step = stepAIC(null, scope=list(lower=null, upper=full), direction = "forward")
```

```
## Start:  AIC=93313.45
## total_bikes ~ 1
##
##                      Df Sum of Sq       RSS   AIC
## + casual              1 174268667 206872475 87978
## + hr_cat              3 135923085 245218057 89468
## + atemp_original      1  59887825 321253317 91822
## + hum_original        1  45119783 336021359 92215
## + season              3  23524192 357616949 92763
```

```
## + weather             1   8387641 372753501 93121
## + windspeed_original  1   4709159 376431983 93207
## + workingday          1    800215 380340926 93297
## + holiday             1    650612 380490530 93301
## <none>                            381141142 93313
##
## Step:  AIC=87978.39
## total_bikes ~ casual
##
##                      Df Sum of Sq       RSS   AIC
## + hr_cat              3  41813670 165058804 86012
## + workingday          1  25665967 181206508 86823
## + atemp_original      1   3653492 203218983 87825
## + hum_original        1   3607111 203265363 87827
## + season              3   3502143 203370332 87835
## + holiday             1   1073453 205799022 87935
## + weather             1    815969 206056506 87946
## + windspeed_original  1    457062 206415413 87961
## <none>                            206872475 87978
##
## Step:  AIC=86012.24
## total_bikes ~ casual + hr_cat
##
##                      Df Sum of Sq       RSS   AIC
## + workingday          1  18817091 146241713 84957
## + season              3   5459740 159599064 85724
## + atemp_original      1   4918193 160140612 85750
## + weather             1   1638882 163419923 85927
## + holiday             1   1022867 164035937 85960
## + hum_original        1    766317 164292488 85974
## <none>                            165058804 86012
## + windspeed_original  1     13147 165045657 86014
##
## Step:  AIC=84957.07
## total_bikes ~ casual + hr_cat + workingday
##
##                      Df Sum of Sq       RSS   AIC
## + season              3   3648820 142592893 84742
## + atemp_original      1   1419291 144822423 84874
## + weather             1   1183977 145057737 84888
## + hum_original        1    337662 145904052 84939
## <none>                            146241713 84957
## + holiday             1     23761 146217952 84958
## + windspeed_original  1      3583 146238130 84959
##
## Step:  AIC=84742.39
## total_bikes ~ casual + hr_cat + workingday + season
##
##                      Df Sum of Sq       RSS   AIC
## + weather             1   1207058 141385835 84670
## + hum_original        1    815254 141777640 84694
## + atemp_original      1    664501 141928392 84704
## + windspeed_original  1    131417 142461476 84736
## <none>                            142592893 84742
## + holiday             1     26492 142566401 84743
##
## Step:  AIC=84670.14
## total_bikes ~ casual + hr_cat + workingday + season + weather
##
##                      Df Sum of Sq       RSS   AIC
## + atemp_original      1    638059 140747777 84633
## + hum_original        1    203724 141182111 84660
## + windspeed_original  1    132392 141253443 84664
## <none>                            141385835 84670
## + holiday             1     10846 141374989 84671
##
## Step:  AIC=84632.64
## total_bikes ~ casual + hr_cat + workingday + season + weather +
##     atemp_original
##
##                      Df Sum of Sq       RSS   AIC
## + hum_original        1    277488 140470288 84617
## + windspeed_original  1    189585 140558192 84623
```

```
## <none>                          140747777 84633
## + holiday             1      4630 140743147 84634
##
## Step:  AIC=84617.4
## total_bikes ~ casual + hr_cat + workingday + season + weather +
##     atemp_original + hum_original
##
##                      Df Sum of Sq       RSS   AIC
## + windspeed_original  1     92252 140378037 84614
## <none>                          140470288 84617
## + holiday             1      5606 140464683 84619
##
## Step:  AIC=84613.66
## total_bikes ~ casual + hr_cat + workingday + season + weather +
##     atemp_original + hum_original + windspeed_original
##
##           Df Sum of Sq       RSS   AIC
## <none>                 140378037 84614
## + holiday  1    5480.6 140372556 84615
```

```
step$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## total_bikes ~ 1
##
## Final Model:
## total_bikes ~ casual + hr_cat + workingday + season + weather +
##     atemp_original + hum_original + windspeed_original
##
##
##                   Step Df     Deviance Resid. Df Resid. Dev      AIC
## 1                                           8733  381141142 93313.45
## 2             + casual  1 174268667.03      8732  206872475 87978.39
## 3             + hr_cat  3  41813670.33      8729  165058804 86012.24
## 4         + workingday  1  18817091.01      8728  146241713 84957.07
## 5             + season  3   3648820.18      8725  142592893 84742.39
## 6            + weather  1   1207057.80      8724  141385835 84670.14
## 7     + atemp_original  1    638058.75      8723  140747777 84632.64
## 8       + hum_original  1    277488.24      8722  140470288 84617.40
## 9 + windspeed_original  1     92251.65      8721  140378037 84613.66
```

```
summary(step)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather + atemp_original + hum_original + windspeed_original,
##     data = bs2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -321.66  -78.30  -31.09   46.83  562.06
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.51077    8.37597   0.777 0.436994
## casual               2.12520    0.03587  59.244  < 2e-16 ***
## hr_cat2            165.33051    3.98597  41.478  < 2e-16 ***
## hr_cat3            138.91892    5.03166  27.609  < 2e-16 ***
## hr_cat4            160.93281    4.13600  38.910  < 2e-16 ***
## workingday          96.63213    3.24670  29.763  < 2e-16 ***
## seasonSpring       -46.23849    4.21666 -10.966  < 2e-16 ***
## seasonSummer       -34.08453    5.10929  -6.671 2.69e-11 ***
## seasonWinter       -50.17654    4.13721 -12.128  < 2e-16 ***
## weather            -14.99283    2.51325  -5.966 2.53e-09 ***
## atemp_original       1.49597    0.22028   6.791 1.18e-11 ***
## hum_original        -0.32672    0.09766  -3.345 0.000825 ***
## windspeed_original   0.43363    0.18113   2.394 0.016688 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.9 on 8721 degrees of freedom
## Multiple R-squared:  0.6317, Adjusted R-squared:  0.6312
## F-statistic:  1246 on 12 and 8721 DF,  p-value: < 2.2e-16
```

```r
# define linear models from the best set of variables
lm1 <- lm(data = bs2, total_bikes ~ casual + hr_cat + workingday + season + weather +
    atemp_original + hum_original + windspeed_original)
lm2 <- lm(data = bs2, total_bikes ~ casual + hr_cat + workingday + season + weather +
    atemp_original + hum_original)
lm3 <- lm(data = bs2, total_bikes ~ casual + hr_cat + workingday + season + weather)
lm4 <- lm(data = bs2, total_bikes ~ casual + hr_cat + workingday + season)
```

```r
summary(lm1)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather + atemp_original + hum_original + windspeed_original,
##     data = bs2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -321.66  -78.30  -31.09   46.83  562.06
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.51077    8.37597   0.777 0.436994
## casual               2.12520    0.03587  59.244  < 2e-16 ***
## hr_cat2            165.33051    3.98597  41.478  < 2e-16 ***
## hr_cat3            138.91892    5.03166  27.609  < 2e-16 ***
## hr_cat4            160.93281    4.13600  38.910  < 2e-16 ***
## workingday          96.63213    3.24670  29.763  < 2e-16 ***
## seasonSpring       -46.23849    4.21666 -10.966  < 2e-16 ***
## seasonSummer       -34.08453    5.10929  -6.671 2.69e-11 ***
## seasonWinter       -50.17654    4.13721 -12.128  < 2e-16 ***
## weather            -14.99283    2.51325  -5.966 2.53e-09 ***
## atemp_original       1.49597    0.22028   6.791 1.18e-11 ***
## hum_original        -0.32672    0.09766  -3.345 0.000825 ***
## windspeed_original   0.43363    0.18113   2.394 0.016688 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.9 on 8721 degrees of freedom
## Multiple R-squared:  0.6317, Adjusted R-squared:  0.6312
## F-statistic:  1246 on 12 and 8721 DF,  p-value: < 2.2e-16
```

```
summary(lm2)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather + atemp_original + hum_original, data = bs2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -320.86  -78.45  -31.38   46.68  562.63
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     14.58332    7.66930   1.902   0.0573 .
## casual           2.12405    0.03588  59.202  < 2e-16 ***
## hr_cat2        165.71852    3.98375  41.599  < 2e-16 ***
## hr_cat3        140.01800    5.01203  27.936  < 2e-16 ***
## hr_cat4        161.47124    4.13100  39.088  < 2e-16 ***
## workingday      96.36760    3.24570  29.691  < 2e-16 ***
## seasonSpring   -45.37604    4.20238 -10.798  < 2e-16 ***
## seasonSummer   -33.69490    5.10808  -6.596 4.46e-11 ***
## seasonWinter   -49.55123    4.13007 -11.998  < 2e-16 ***
## weather        -14.22069    2.49314  -5.704 1.21e-08 ***
## atemp_original   1.46176    0.21987   6.648 3.15e-11 ***
## hum_original    -0.39024    0.09401  -4.151 3.34e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.9 on 8722 degrees of freedom
## Multiple R-squared:  0.6314, Adjusted R-squared:  0.631
## F-statistic:  1359 on 11 and 8722 DF,  p-value: < 2.2e-16
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather, data = bs2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -333.28  -78.62  -30.73   47.74  555.32
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.42369    5.40185   0.449  0.65367
## casual         2.22890    0.03318  67.185  < 2e-16 ***
## hr_cat2      165.63816    3.98729  41.542  < 2e-16 ***
## hr_cat3      148.43248    4.77937  31.057  < 2e-16 ***
## hr_cat4      167.12569    4.03757  41.393  < 2e-16 ***
## workingday   101.66205    3.17741  31.995  < 2e-16 ***
## seasonSpring -34.84433    3.94349  -8.836  < 2e-16 ***
## seasonSummer -12.42659    3.93245  -3.160  0.00158 **
## seasonWinter -54.66456    3.94998 -13.839  < 2e-16 ***
## weather      -19.17683    2.22207  -8.630  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.3 on 8724 degrees of freedom
## Multiple R-squared:  0.629,  Adjusted R-squared:  0.6287
## F-statistic:  1644 on 9 and 8724 DF,  p-value: < 2.2e-16
```

```
summary(lm4)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season,
##     data = bs2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -332.10  -76.87  -33.12   47.44  564.46
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -26.54181    4.25042  -6.245 4.45e-10 ***
## casual         2.27492    0.03288  69.185  < 2e-16 ***
## hr_cat2      163.73967    3.99795  40.956  < 2e-16 ***
## hr_cat3      144.82637    4.78107  30.292  < 2e-16 ***
## hr_cat4      165.95859    4.05226  40.955  < 2e-16 ***
## workingday   102.83364    3.18785  32.258  < 2e-16 ***
## seasonSpring -34.63479    3.95998  -8.746  < 2e-16 ***
## seasonSummer -10.59073    3.94319  -2.686  0.00725 **
## seasonWinter -53.66788    3.96488 -13.536  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.8 on 8725 degrees of freedom
## Multiple R-squared:  0.6259, Adjusted R-squared:  0.6255
## F-statistic:  1825 on 8 and 8725 DF,  p-value: < 2.2e-16
```

We can now use these models to test the assumptions of linear regression. We can run the gvlma function to quickly test the assumptions of each of the linear regression models.

```
# run assumptions on the models
gvlma(lm1)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather + atemp_original + hum_original + windspeed_original,
##     data = bs2)
##
## Coefficients:
##        (Intercept)              casual             hr_cat2
##             6.5108              2.1252            165.3305
##            hr_cat3             hr_cat4          workingday
##           138.9189            160.9328             96.6321
##       seasonSpring        seasonSummer        seasonWinter
##           -46.2385            -34.0845            -50.1765
##            weather      atemp_original        hum_original
##           -14.9928              1.4960             -0.3267
## windspeed_original
##             0.4336
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = lm1)
##
##                        Value  p-value                   Decision
## Global Stat         6356.459 0.000000 Assumptions NOT satisfied!
## Skewness            3516.616 0.000000 Assumptions NOT satisfied!
## Kurtosis            2706.115 0.000000 Assumptions NOT satisfied!
## Link Function          7.028 0.008023 Assumptions NOT satisfied!
## Heteroscedasticity   126.700 0.000000 Assumptions NOT satisfied!
```

```
gvlma(lm2)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather + atemp_original + hum_original, data = bs2)
##
## Coefficients:
##    (Intercept)          casual         hr_cat2         hr_cat3
##        14.5833          2.1241        165.7185        140.0180
##        hr_cat4      workingday    seasonSpring    seasonSummer
##       161.4712         96.3676        -45.3760        -33.6949
##   seasonWinter         weather  atemp_original    hum_original
##       -49.5512        -14.2207          1.4618         -0.3902
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = lm2)
##
##                        Value  p-value                   Decision
## Global Stat         6363.521 0.000000 Assumptions NOT satisfied!
## Skewness            3522.217 0.000000 Assumptions NOT satisfied!
## Kurtosis            2706.778 0.000000 Assumptions NOT satisfied!
## Link Function          7.562 0.005962 Assumptions NOT satisfied!
## Heteroscedasticity   126.964 0.000000 Assumptions NOT satisfied!
```

```
gvlma(lm3)
```

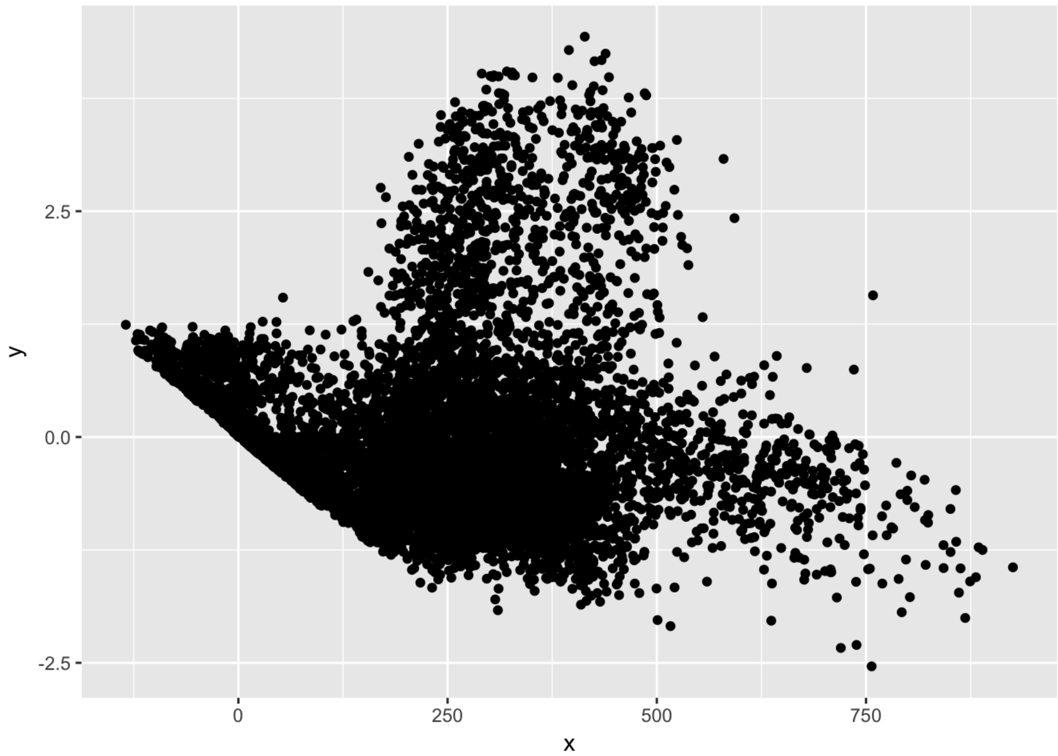
```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season +
##     weather, data = bs2)
##
## Coefficients:
##  (Intercept)        casual       hr_cat2       hr_cat3       hr_cat4
##        2.424         2.229       165.638       148.432       167.126
##   workingday  seasonSpring  seasonSummer  seasonWinter       weather
##      101.662       -34.844       -12.427       -54.665       -19.177
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = lm3)
##
##                       Value   p-value                   Decision
## Global Stat        6346.86 0.000e+00 Assumptions NOT satisfied!
## Skewness           3517.26 0.000e+00 Assumptions NOT satisfied!
## Kurtosis           2691.34 0.000e+00 Assumptions NOT satisfied!
## Link Function        16.46 4.977e-05 Assumptions NOT satisfied!
## Heteroscedasticity  121.80 0.000e+00 Assumptions NOT satisfied!
```

```
gvlma(lm4)
```

```
##
## Call:
## lm(formula = total_bikes ~ casual + hr_cat + workingday + season,
##     data = bs2)
##
## Coefficients:
##  (Intercept)        casual       hr_cat2       hr_cat3       hr_cat4
##      -26.542         2.275       163.740       144.826       165.959
##   workingday  seasonSpring  seasonSummer  seasonWinter
##      102.834       -34.635       -10.591       -53.668
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = lm4)
##
##                       Value   p-value                   Decision
## Global Stat        6325.10 0.000e+00 Assumptions NOT satisfied!
## Skewness           3495.91 0.000e+00 Assumptions NOT satisfied!
## Kurtosis           2681.33 0.000e+00 Assumptions NOT satisfied!
## Link Function        29.45 5.738e-08 Assumptions NOT satisfied!
## Heteroscedasticity  118.41 0.000e+00 Assumptions NOT satisfied!
```

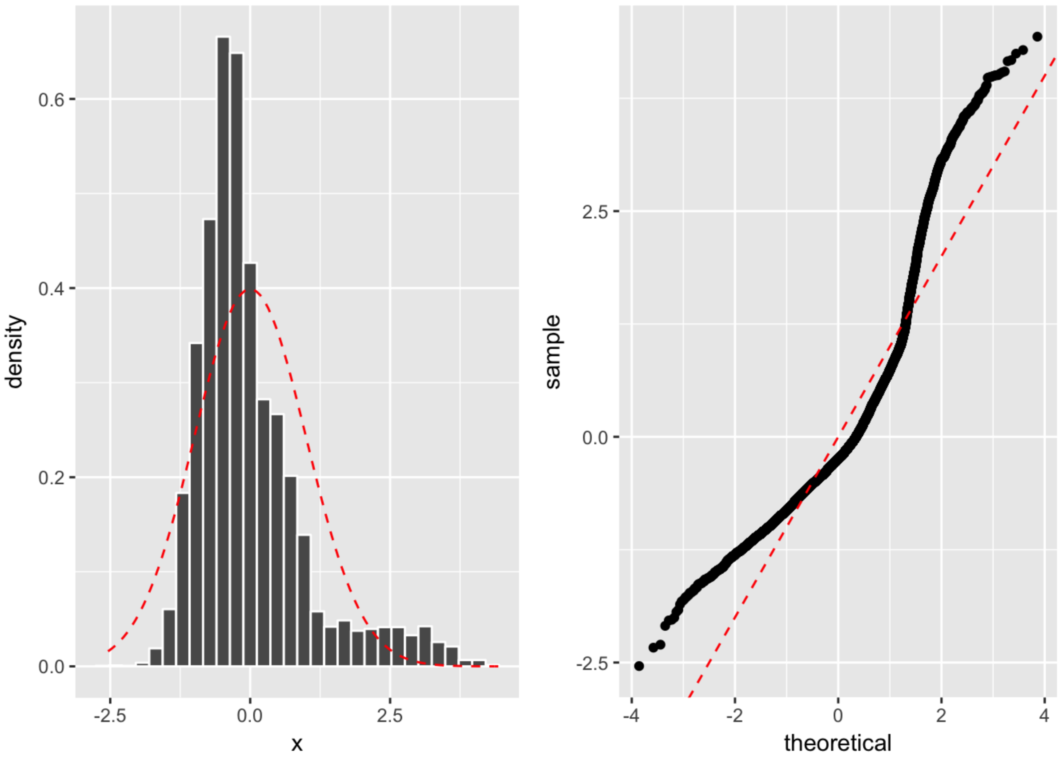
We can see that for all of the models the homoscedasticity assumption is not held. We can choose one of the linear model manually verify important assumptions of linear regression.

```
# check homoscedasticity
lm1_df <- data.frame(x = rstandard(lm1))

ggplot(data.frame(x = predict(lm1), y = rstandard(lm1))) +
  geom_point(aes(x, y ))
```




```r
# check normality of residuals (histogram)
grid.arrange(
  ggplot(lm1_df, aes(x = x)) +
    geom_histogram(color = I('white'), aes(y = ..density..)) +
    stat_function(fun = dnorm, args = list(mean = mean(lm1_df$x),
                                            sd = sd(lm1_df$x)),
                  color = I('red'),
                  linetype = 2),

  ggplot(lm1_df, aes(sample = scale(x))) +
    stat_qq() +
    geom_abline(slope = 1, intercept = 0, color = I('red'), linetype = 2),

  ncol = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



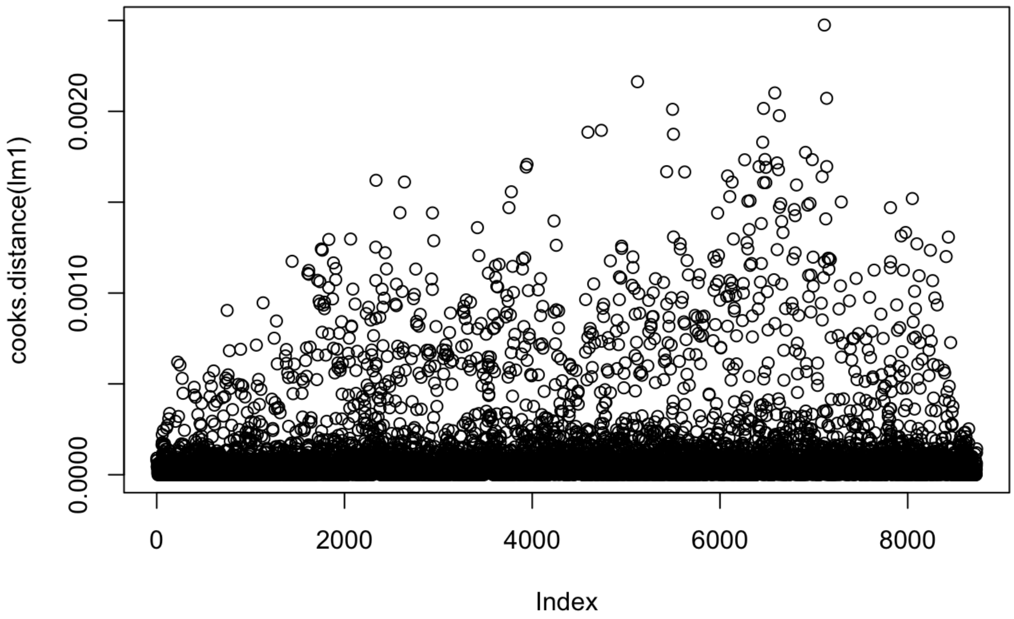

```
# check collinearlity of variables
vif(lm1)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## casual             2.265863  1        1.505278
## hr_cat             1.805699  3        1.103505
## workingday         1.241059  1        1.114028
## season             2.787683  3        1.186335
## weather            1.336359  1        1.156010
## atemp_original     3.182344  1        1.783913
## hum_original       1.832492  1        1.353696
## windspeed_original 1.179365  1        1.085986
```

```
# check for outliers
outlierTest(lm1)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferonni p
## 6320 4.437051          9.232e-06     0.080632
```

```
# check for influential points that may have an impact on our analysis
plot(cooks.distance(lm1))
```




```
# autocorrelation of errors
durbinWatsonTest(lm1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.5805379     0.8386622       0
##  Alternative hypothesis: rho != 0
```

We can see that while there are no significant influencers that can have an effect on the regression model (influencer test), and the variables do not show signs of collinearity (vif test), the residuals don't follow a perfect normal distribution, and the residuals do not hold the assumption of homoscedasity. We can also further check whether our residuals are really heteroscedastic by using the Goldfeld-Quandt test. The null hypothesis is that the residuas are homoscedastic, and the alternative is that they are not. Getting a gq statstic that is more than 0.05 (or the alpha level) will indicate that the model truly has a heteroscedasticity problem.

```
# test for hereostedcity
gqtest(data = bs2, total_bikes ~ casual + hr_cat + workingday + season + weather +
    atemp_original + hum_original + windspeed_original)
```

```
##
##  Goldfeld-Quandt test
##
## data:  total_bikes ~ casual + hr_cat + workingday + season + weather +     atemp_original + hum_original + windspeed_original
## GQ = 1.4149, df1 = 4354, df2 = 4354, p-value < 2.2e-16
## alternative hypothesis: variance increases from segment 1 to 2
```

Since the GQ value is 1.41, which is higher than alpha value 0.05, we can reject the null and say that there isa true heteroscedasticty roblem with our model.

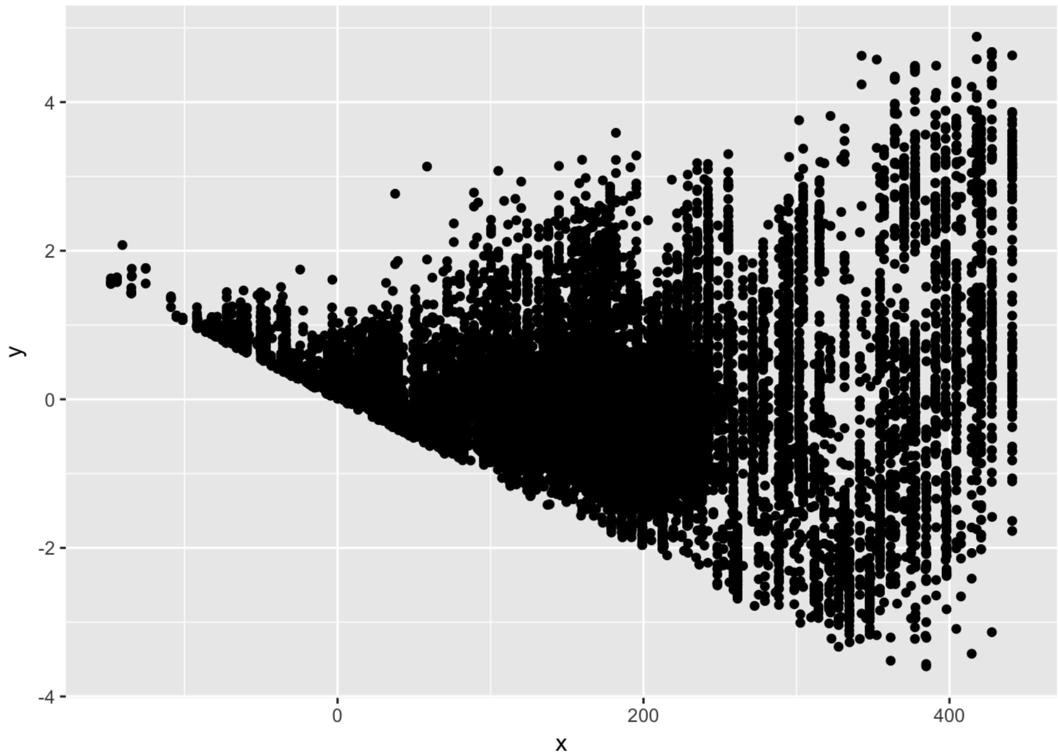
Dual model for each type of users

We were looking for ways to improve our initial model, and based on our variable analysis and statstical tests, we decided that it may yield better models with stronger predictive powers when we divide the prediction model for each type of users, casual and registered. This was because there were some differences in bike usage pattern between the two types of users, and thus separating them would make more sense.

```
# model for registered users
lmr <- lm(data = bs, registered ~ season + factor(hr) + factor(workingday) + factor(weather))
summary(lmr)
```

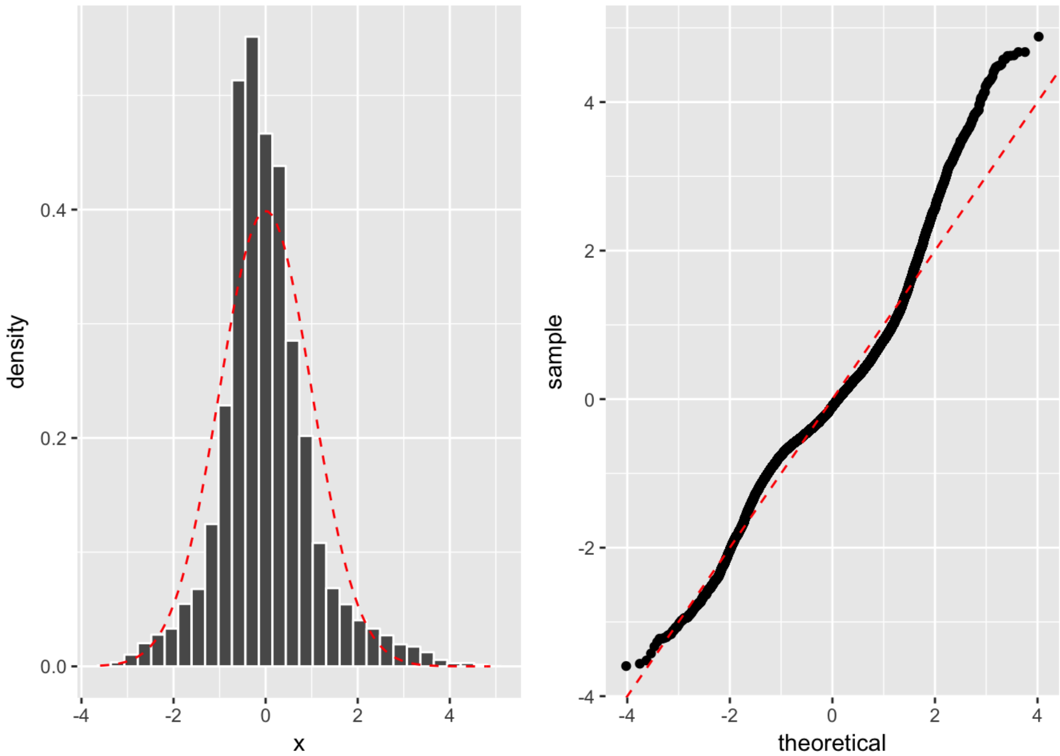
```
##
## Call:
## lm(formula = registered ~ season + factor(hr) + factor(workingday) +
##     factor(weather), data = bs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -344.78  -52.83   -9.16   43.91  468.20
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           39.421      3.967   9.938  < 2e-16 ***
## seasonSpring          -7.003      2.067  -3.388 0.000706 ***
## seasonSummer          13.330      2.064   6.459 1.08e-10 ***
## seasonWinter         -73.064      2.086 -35.018  < 2e-16 ***
## factor(hr)1          -16.519      5.043  -3.276 0.001055 **
## factor(hr)2          -25.620      5.058  -5.065 4.12e-07 ***
## factor(hr)3          -35.342      5.091  -6.942 4.00e-12 ***
## factor(hr)4          -39.274      5.091  -7.714 1.28e-14 ***
## factor(hr)5          -25.926      5.055  -5.129 2.94e-07 ***
## factor(hr)6           29.030      5.041   5.759 8.61e-09 ***
## factor(hr)7          159.746      5.038  31.709  < 2e-16 ***
## factor(hr)8          295.098      5.038  58.575  < 2e-16 ***
## factor(hr)9          146.042      5.039  28.984  < 2e-16 ***
## factor(hr)10          84.458      5.038  16.765  < 2e-16 ***
## factor(hr)11         105.303      5.038  20.902  < 2e-16 ***
## factor(hr)12         142.541      5.036  28.303  < 2e-16 ***
## factor(hr)13         138.730      5.034  27.560  < 2e-16 ***
## factor(hr)14         122.815      5.033  24.400  < 2e-16 ***
## factor(hr)15         133.989      5.034  26.619  < 2e-16 ***
## factor(hr)16         195.873      5.033  38.917  < 2e-16 ***
## factor(hr)17         345.356      5.033  68.619  < 2e-16 ***
## factor(hr)18         322.138      5.036  63.963  < 2e-16 ***
## factor(hr)19         219.379      5.035  43.567  < 2e-16 ***
## factor(hr)20         146.992      5.036  29.186  < 2e-16 ***
## factor(hr)21         100.443      5.035  19.947  < 2e-16 ***
## factor(hr)22          65.368      5.035  12.982  < 2e-16 ***
## factor(hr)23          30.248      5.035   6.007 1.93e-09 ***
## factor(workingday)1   42.911      1.567  27.379  < 2e-16 ***
## factor(weather)2     -13.165      1.698  -7.753 9.48e-15 ***
## factor(weather)3     -75.236      2.710 -27.759  < 2e-16 ***
## factor(weather)4     -90.462     55.479  -1.631 0.103000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96 on 17348 degrees of freedom
## Multiple R-squared:  0.5984, Adjusted R-squared:  0.5977
## F-statistic: 861.8 on 30 and 17348 DF,  p-value: < 2.2e-16
```

```
# check homoscedasticity
lmr_df <- data.frame(x = rstandard(lmr))

ggplot(data.frame(x = predict(lmr), y = rstandard(lmr))) +
  geom_point(aes(x, y ))
```




```r
# check normality of residuals (histogram)
grid.arrange(
  ggplot(lmr_df, aes(x = x)) +
    geom_histogram(color = I('white'), aes(y = ..density..)) +
    stat_function(fun = dnorm, args = list(mean = mean(lmr_df$x),
                                           sd = sd(lmr_df$x)),
                  color = I('red'),
                  linetype = 2),

  ggplot(lmr_df, aes(sample = scale(x))) +
    stat_qq() +
    geom_abline(slope = 1, intercept = 0, color = I('red'), linetype = 2),

  ncol = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



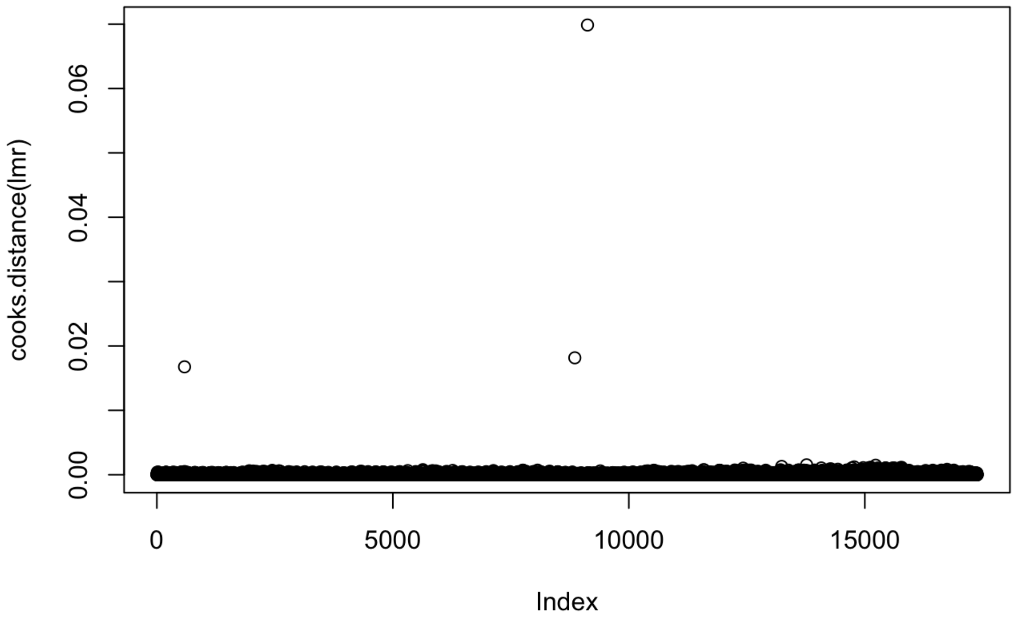

```
# check collinearlity of variables
vif(lmr)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## season              1.011542  3        1.001915
## factor(hr)          1.011563 23        1.000250
## factor(workingday) 1.003463  1        1.001730
## factor(weather)     1.023418  3        1.003865
```

```
# check for outliers
outlierTest(lmr)
```

```
##       rstudent unadjusted p-value Bonferonni p
## 14774  4.88432         1.0471e-06     0.018198
```

```
# check for influential points that may have an impact on our analysis
plot(cooks.distance(lmr))
```



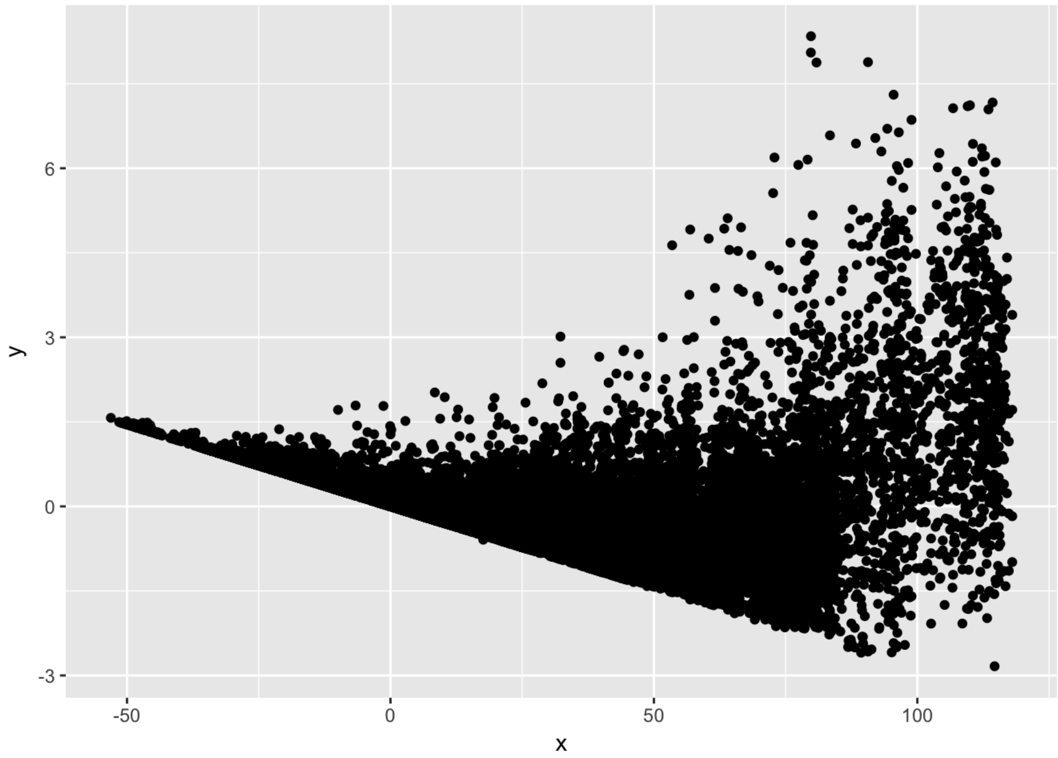

```
# autocorrelation of errors
durbinWatsonTest(lmr)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7595005     0.4809854       0
##  Alternative hypothesis: rho != 0
```

```
# model for casual users
lmc <- lm(data =bs, casual ~ factor(hr) + factor(weather) + season + windspeed_original + factor(workingday))
summary(lmc)
```

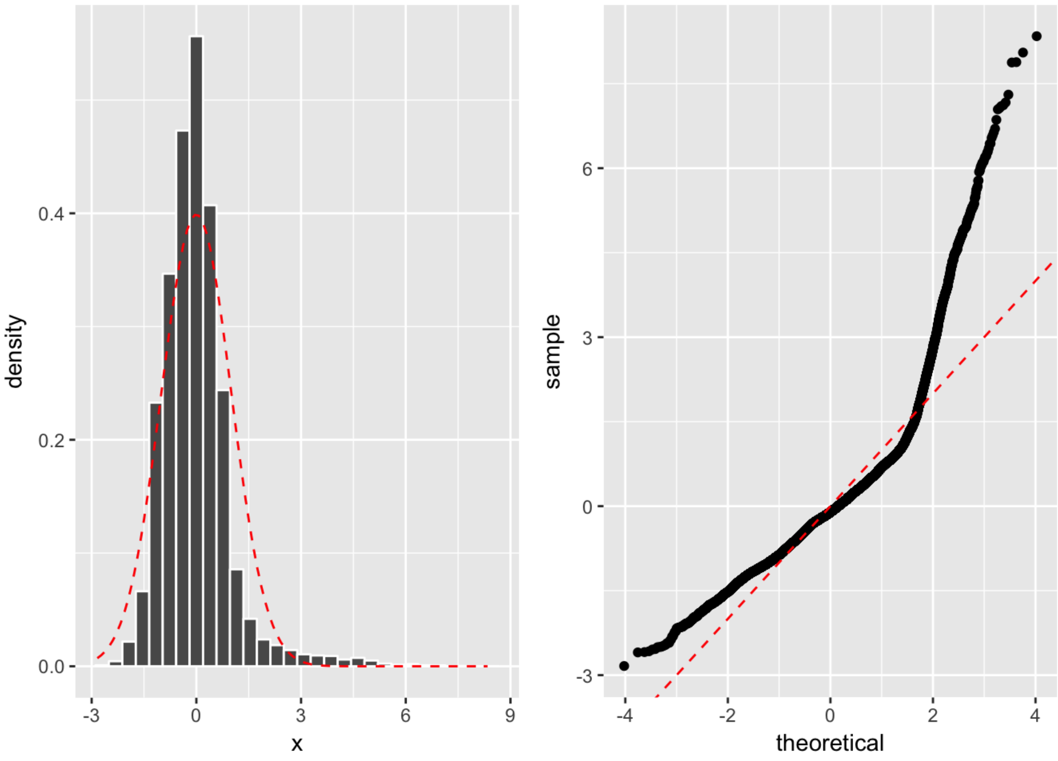
```
##
## Call:
## lm(formula = casual ~ factor(hr) + factor(weather) + season +
##     windspeed_original + factor(workingday), data = bs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -97.665 -20.767  -3.697  13.778 287.187
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           32.43514    1.46064  22.206  < 2e-16 ***
## factor(hr)1           -3.65339    1.80984  -2.019  0.04354 *
## factor(hr)2           -5.63074    1.81540  -3.102  0.00193 **
## factor(hr)3           -8.27011    1.82732  -4.526 6.06e-06 ***
## factor(hr)4           -9.67499    1.82723  -5.295 1.21e-07 ***
## factor(hr)5           -8.67466    1.81430  -4.781 1.76e-06 ***
## factor(hr)6           -5.60021    1.80923  -3.095  0.00197 **
## factor(hr)7            1.70475    1.80813   0.943  0.34578
## factor(hr)8           12.24361    1.80848   6.770 1.33e-11 ***
## factor(hr)9           21.62562    1.80970  11.950  < 2e-16 ***
## factor(hr)10          37.14592    1.81046  20.517  < 2e-16 ***
## factor(hr)11          50.11207    1.81116  27.668  < 2e-16 ***
## factor(hr)12          59.19356    1.81168  32.673  < 2e-16 ***
## factor(hr)13          63.14818    1.81162  34.857  < 2e-16 ***
## factor(hr)14          66.46054    1.81345  36.649  < 2e-16 ***
## factor(hr)15          65.81275    1.81394  36.282  < 2e-16 ***
## factor(hr)16          64.56096    1.81409  35.589  < 2e-16 ***
## factor(hr)17          65.19209    1.81318  35.955  < 2e-16 ***
## factor(hr)18          51.91131    1.81288  28.635  < 2e-16 ***
## factor(hr)19          39.11110    1.81041  21.603  < 2e-16 ***
## factor(hr)20          26.49976    1.80901  14.649  < 2e-16 ***
## factor(hr)21          18.23881    1.80774  10.089  < 2e-16 ***
## factor(hr)22          12.17715    1.80742   6.737 1.66e-11 ***
## factor(hr)23           5.43341    1.80726   3.006  0.00265 **
## factor(weather)2      -7.19881    0.60993 -11.803  < 2e-16 ***
## factor(weather)3     -21.47235    0.97435 -22.038  < 2e-16 ***
## factor(weather)4     -26.59480   19.91199  -1.336  0.18169
## seasonSpring          16.24507    0.74533  21.796  < 2e-16 ***
## seasonSummer          19.11433    0.74075  25.804  < 2e-16 ***
## seasonWinter         -17.18321    0.75520 -22.753  < 2e-16 ***
## windspeed_original    -0.14194    0.03344  -4.244 2.21e-05 ***
## factor(workingday)1  -32.48518    0.56255 -57.746  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.45 on 17347 degrees of freedom
## Multiple R-squared:  0.5126, Adjusted R-squared:  0.5117
## F-statistic: 588.4 on 31 and 17347 DF,  p-value: < 2.2e-16
```

```r
# check homoscedasticity
lmc_df <- data.frame(x = rstandard(lmc))

ggplot(data.frame(x = predict(lmc), y = rstandard(lmc))) +
  geom_point(aes(x, y ))
```




```r
# check normality of residuals (histogram)
grid.arrange(
  ggplot(lmc_df, aes(x = x)) +
    geom_histogram(color = I('white'), aes(y = ..density..)) +
    stat_function(fun = dnorm, args = list(mean = mean(lmc_df$x),
                                           sd = sd(lmc_df$x)),
                  color = I('red'),
                  linetype = 2),

  ggplot(lmc_df, aes(sample = scale(x))) +
    stat_qq() +
    geom_abline(slope = 1, intercept = 0, color = I('red'), linetype = 2),

  ncol = 2
)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



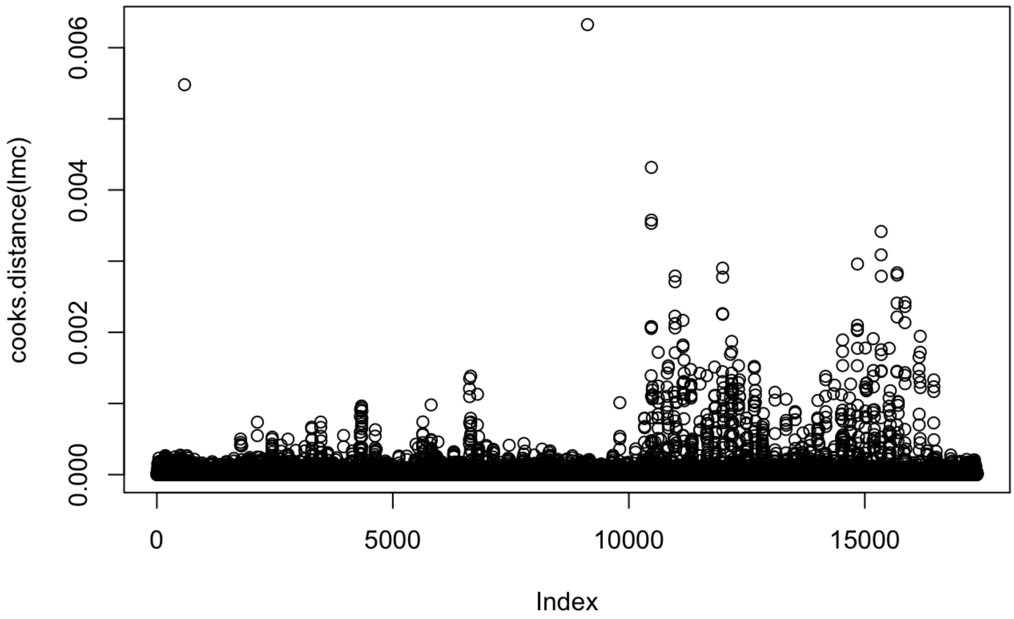

```r
# check collinearlity of variables
vif(lmc)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## factor(hr)          1.077878 23        1.001632
## factor(weather)     1.029465  3        1.004852
## season              1.038770  3        1.006360
## windspeed_original 1.100099  1        1.048856
## factor(workingday) 1.003580  1        1.001788
```

```r
# check for outliers
outlierTest(lmc)
```

```
##       rstudent unadjusted p-value Bonferonni p
## 10478 8.360202         6.7299e-17   1.1696e-12
## 10477 8.067838         7.6189e-16   1.3241e-11
## 15344 7.897018         3.0262e-15   5.2593e-11
## 10476 7.890678         3.1834e-15   5.5325e-11
## 15685 7.316911         2.6483e-13   4.6024e-09
## 11986 7.177503         7.3867e-13   1.2837e-08
## 10978 7.127709         1.0607e-12   1.8433e-08
## 10981 7.111379         1.1937e-12   2.0745e-08
## 14844 7.075882         1.5417e-12   2.6794e-08
## 11987 7.054587         1.7965e-12   3.1221e-08
```

```r
# check for influential points that may have an impact on our analysis
plot(cooks.distance(lmc))
```




```
# autocorrelation of errors
durbinWatsonTest(lmc)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.8857809     0.2284004       0
##  Alternative hypothesis: rho != 0
```

The R2 values are around 0.5 for both models, which means that about half of the information of the predictions are explained by other things other than the variables. Furthermore, the residuals, are not homoscedastic, meaning that somehow the residuals are worse for certain types of observations. Thus, even when separated, the models don't explain the number of bikes rented completely.

7. Regression model validation

Now let's validate our 2 user models by using k-folds cross validation.

create training set and testing results for casual model

```
trainingFold1 = createDataPartition(bs$casual, p = 0.8)
training1 = bs[trainingFold1$Resample1, ]
testing1  = bs[-trainingFold1$Resample1, ]


trainMethod = trainControl(method="cv", number=5, returnData =TRUE, returnResamp = "all")
model_casual = train(data=training1, casual ~ factor(hr) + factor(weather) + season + windspeed_original + factor(workingday), method = 'lm')
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading
```

```
summary(model_casual)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -89.469 -20.811  -3.579  13.816 286.757
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             32.81512    1.63537  20.066  < 2e-16 ***
## `factor(hr)1`           -4.33155    2.01378  -2.151 0.031497 *
## `factor(hr)2`           -5.83874    2.03397  -2.871 0.004103 **
## `factor(hr)3`           -8.87002    2.03148  -4.366 1.27e-05 ***
## `factor(hr)4`          -10.03819    2.03798  -4.926 8.51e-07 ***
## `factor(hr)5`           -9.28077    2.02497  -4.583 4.62e-06 ***
## `factor(hr)6`           -5.84334    2.03876  -2.866 0.004161 **
## `factor(hr)7`            0.57100    2.00137   0.285 0.775416
## `factor(hr)8`           12.17885    2.01251   6.052 1.47e-09 ***
## `factor(hr)9`           20.56329    2.01551  10.203  < 2e-16 ***
## `factor(hr)10`          36.54202    2.02824  18.017  < 2e-16 ***
## `factor(hr)11`          51.09415    2.02132  25.278  < 2e-16 ***
## `factor(hr)12`          57.82189    2.03322  28.439  < 2e-16 ***
## `factor(hr)13`          63.50282    2.02947  31.290  < 2e-16 ***
## `factor(hr)14`          66.50125    2.01560  32.993  < 2e-16 ***
## `factor(hr)15`          65.60599    2.01089  32.625  < 2e-16 ***
## `factor(hr)16`          64.58776    2.03656  31.714  < 2e-16 ***
## `factor(hr)17`          64.23472    2.03286  31.598  < 2e-16 ***
## `factor(hr)18`          51.01490    2.01311  25.341  < 2e-16 ***
## `factor(hr)19`          37.41676    2.02845  18.446  < 2e-16 ***
## `factor(hr)20`          26.26593    2.01315  13.047  < 2e-16 ***
## `factor(hr)21`          18.04512    2.01034   8.976  < 2e-16 ***
## `factor(hr)22`          11.95446    2.01748   5.925 3.19e-09 ***
## `factor(hr)23`           5.50100    2.00116   2.749 0.005987 **
## `factor(weather)2`      -6.88966    0.68127 -10.113  < 2e-16 ***
## `factor(weather)3`     -21.34174    1.07918 -19.776  < 2e-16 ***
## `factor(weather)4`     -26.58351   19.88001  -1.337 0.181180
## seasonSpring            16.08865    0.83336  19.306  < 2e-16 ***
## seasonSummer            18.66586    0.82543  22.613  < 2e-16 ***
## seasonWinter           -17.16027    0.84456 -20.318  < 2e-16 ***
## windspeed_original      -0.14224    0.03756  -3.787 0.000153 ***
## `factor(workingday)1`  -32.32637    0.62885 -51.406  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.39 on 13873 degrees of freedom
## Multiple R-squared:  0.5117, Adjusted R-squared:  0.5106
## F-statistic: 468.9 on 31 and 13873 DF,  p-value: < 2.2e-16
```

create training set and testing results for registered model

```
trainingFold2 = createDataPartition(bs$registered, p = 0.8)
training2 = bs[trainingFold2$Resample1, ]
testing2  = bs[-trainingFold2$Resample1, ]

trainMethod = trainControl(method="cv", number=5, returnData =TRUE, returnResamp = "all")
model_registered = train(data=training2, registered ~ season + factor(hr) + factor(workingday) + factor(weather), method = 'lm')
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
## fit may be misleading
```

```
summary(model_registered)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -341.80  -52.89   -9.46   43.74  474.13
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             38.133      4.390   8.687  < 2e-16 ***
## seasonSpring            -6.570      2.304  -2.851  0.00437 **
## seasonSummer            14.520      2.307   6.295 3.17e-10 ***
## seasonWinter           -73.037      2.329 -31.364  < 2e-16 ***
## `factor(hr)1`          -16.913      5.615  -3.012  0.00260 **
## `factor(hr)2`          -27.490      5.612  -4.899 9.76e-07 ***
## `factor(hr)3`          -36.198      5.647  -6.410 1.50e-10 ***
## `factor(hr)4`          -39.322      5.673  -6.932 4.34e-12 ***
## `factor(hr)5`          -26.072      5.619  -4.640 3.52e-06 ***
## `factor(hr)6`           30.236      5.607   5.392 7.06e-08 ***
## `factor(hr)7`          160.656      5.600  28.686  < 2e-16 ***
## `factor(hr)8`          290.191      5.564  52.151  < 2e-16 ***
## `factor(hr)9`          146.081      5.608  26.048  < 2e-16 ***
## `factor(hr)10`          85.672      5.630  15.217  < 2e-16 ***
## `factor(hr)11`         105.193      5.598  18.791  < 2e-16 ***
## `factor(hr)12`         143.746      5.622  25.568  < 2e-16 ***
## `factor(hr)13`         139.522      5.599  24.918  < 2e-16 ***
## `factor(hr)14`         124.242      5.606  22.160  < 2e-16 ***
## `factor(hr)15`         135.117      5.602  24.119  < 2e-16 ***
## `factor(hr)16`         195.203      5.601  34.850  < 2e-16 ***
## `factor(hr)17`         343.669      5.597  61.406  < 2e-16 ***
## `factor(hr)18`         315.973      5.603  56.390  < 2e-16 ***
## `factor(hr)19`         215.494      5.550  38.827  < 2e-16 ***
## `factor(hr)20`         149.539      5.559  26.902  < 2e-16 ***
## `factor(hr)21`         101.574      5.581  18.201  < 2e-16 ***
## `factor(hr)22`          65.365      5.612  11.648  < 2e-16 ***
## `factor(hr)23`          30.888      5.630   5.487 4.17e-08 ***
## `factor(workingday)1`   43.248      1.750  24.719  < 2e-16 ***
## `factor(weather)2`     -10.890      1.903  -5.722 1.08e-08 ***
## `factor(weather)3`     -75.927      2.999 -25.316  < 2e-16 ***
## `factor(weather)4`     -87.015     55.446  -1.569  0.11658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95.91 on 13874 degrees of freedom
## Multiple R-squared:  0.5947, Adjusted R-squared:  0.5938
## F-statistic: 678.5 on 30 and 13874 DF,  p-value: < 2.2e-16
```

Again, the r2 for both models are still at 51~ 59%, meaning that we may need more variables that may explain the total number of bikes better.

8. Conclusion

While we have had some interesting insights for our dataset, we weren't able to build a sastisfying model despite our efforts:

- Different models: we have created various models playing with different variables to predict the total number of bikes borrowed. We also created two separate models for different types of users, as the users have different usage patterns.
- Numerous tests: we have tested the models against numerous tests to see if the assumptions of linear regression held and what actions we could further do to improve the regression model.
- Playing with variables: we tried to create new variables (hr_cat) to better explain the model. We tried to remove errors coming from autocorrelated errors due to timeseries data by limiting the dataset to 1 year.

We believe that there are various ways in which this prediction model can be improved: * Try other predictive algorithms: this dataset may not be appropriate for linear regression, and thus that may be the reason why the predictive results were not satisfying. * More data on customers: each of the row in this dataset is actually an aggregate of customers by each hour. This means that there are lack of customer related data, especially on an individual level. It would be nice to have more information on individual customers that may help improve the prediction power (e.g. age, gender, nationality etc.), as these personal variables may also have an impact on the usage pattern. Currently, the only thing we know about the customers themselves are whether they are registered or casual customers, and to assume all of them would borrow bieks for similar reasons and patterns would be a big mistake.