FOPS

Tommaso Strada

2022-11

Contents

Introduction

The goal of this project is to briefly describe the dataset and do a linear regression model to predict the number of rented bikes in a given time frame.

Link al dataset: [https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset] We use a daily aggregated dataset, which includes 731 records over 2011 and 2012.

There are 16 variables:

instant: record index

dteday : date

season : season (1:springer, 2:summer, 3:fall, 4:winter)

yr : year (0: 2011, 1:2012)

mnth : month (1 to 12)

hr : hour (0 to 23)

holiday : weather day is holiday or not (extracted from http://dchr.dc.gov/page/holiday-schedule)

weekday : day of the week

workingday : if day is neither weekend nor holiday is 1, otherwise is 0.

weathersit : - 1: Clear, Few clouds, Partly cloudy, Partly cloudy - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog.

temp : Normalized temperature in Celsius. The values are divided to 41 (max)

atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)

hum: Normalized humidity. The values are divided to 100 (max)

windspeed: Normalized wind speed. The values are divided to 67 (max)

casual: count of casual users

registered: count of registered users

cnt: count of total rental bikes including both casual and registered

'cnt' is the target variable.

Requirements

For this R project different packages are required.

- install.packages("ggplot2")
- install.packages("ggpubr")
- install.packages("GGally")
- install.packages("ggpairs")
- install.packages("wesanderson")
- install.packages("ggcorrplot")
- install.packages("moments")
- install.packages("olsrr")

Import libraries

```
library(ggplot2)
library(ggpubr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(RColorBrewer)
library(wesanderson)
library(car)
```

```
## Caricamento del pacchetto richiesto: carData
```

```
library(stringr)
library(moments)
library(olsrr)
```

```
##
## Caricamento pacchetto: 'olsrr'
```

```
## Il seguente oggetto è mascherato da 'package:datasets':
##
##     rivers
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(dplyr)
```

```
##
## Caricamento pacchetto: 'dplyr'
```

```
## Il seguente oggetto è mascherato da 'package:car':
##
##     recode
```

```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag
```

```
## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
```

```
##
## Caricamento pacchetto: 'purrr'

## Il seguente oggetto è mascherato da 'package:car':
##
##     some
```

Data Exploration

Set our working directory:

```
setwd("C:/Users/Tommi/OneDrive/Desktop/Data Science/I ANNO/Primo semestre/F.Probability and Statistic/")
getwd()
```

```
## [1] "C:/Users/Tommi/OneDrive/Desktop/Data Science/I ANNO/Primo semestre/F.Probability and Statistic"
```

Upload the dataset:

```
dt <- read.table('day.csv', sep = ',', header = TRUE)
```

Check if there are all data and variables expressed in the documentation.

```
View(dt)

##  Dataset Shape
dim(dt)
```

```
## [1] 731  16
```

The dataset contains 731 rows and 16 columns.

Check variables types.

```
str(dt)
```

```
## 'data.frame':    731 obs. of  16 variables:
##  $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : chr  "2011-01-01" "2011-01-02" "2011-01-03" "2011-01-04" ...
##  $ 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 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
```

```
##  $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
##  $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

There are 3 kind of variables: the most frequent variable is integer type. It's a numeric sub-category which implies integer numbers only. 'dteday' is a character type that's equal to a string. The other variables are numeric type. We noticed that some variables are not passed correctly. Indeed some of them, such as 'season', must be passed as dummy variables inside of integer.

Preprocessing

In this phase we transform some variables from integers to string characters to be more interpretable. Then we tranform them to dummy variables.

season

```
dt$season <- str_replace_all(dt$season, "1", "springer")
dt$season <- str_replace_all(dt$season, "2", "summer")
dt$season <- str_replace_all(dt$season, "3", "fall")
dt$season <- str_replace_all(dt$season, "4", "winter")

dt$season <- as.factor(dt$season)
```

weathersit

```
dt$weathersit <- str_replace_all(dt$weathersit, "1", "Good")
dt$weathersit <- str_replace_all(dt$weathersit, "2", "Fair")
dt$weathersit <- str_replace_all(dt$weathersit, "3", "Bad")
dt$weathersit <- str_replace_all(dt$weathersit, "4", "Very_bad")

dt$weathersit <- as.factor(dt$weathersit)
```

workingday

```
dt$workingday <- str_replace_all(dt$workingday, "1", "Workday")
dt$workingday <- str_replace_all(dt$workingday, "0", "Holiday")

dt$workingday <- as.factor(dt$workingday)
```

mnth

```
dt$mnth <- str_replace_all(dt$mnth, "10", "Oct")
dt$mnth <- str_replace_all(dt$mnth, "11", "Nov")
dt$mnth <- str_replace_all(dt$mnth, "12", "Dec")
dt$mnth <- str_replace_all(dt$mnth, "1", "Gen")
```

```r
dt$mnth <- str_replace_all(dt$mnth, "2", "Feb")
dt$mnth <- str_replace_all(dt$mnth, "3", "Mar")
dt$mnth <- str_replace_all(dt$mnth, "4", "Apr")
dt$mnth <- str_replace_all(dt$mnth, "5", "May")
dt$mnth <- str_replace_all(dt$mnth, "6", "Jun")
dt$mnth <- str_replace_all(dt$mnth, "7", "Jul")
dt$mnth <- str_replace_all(dt$mnth, "8", "Aug")
dt$mnth <- str_replace_all(dt$mnth, "9", "Sep")

dt$mnth <- factor(dt$mnth , levels=c("Gen", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oc
```

weekday

```r
dt$weekday <- str_replace_all(dt$weekday, "0", "Mon")
dt$weekday <- str_replace_all(dt$weekday, "1", "Tue")
dt$weekday <- str_replace_all(dt$weekday, "2", "Wen")
dt$weekday <- str_replace_all(dt$weekday, "3", "Thu")
dt$weekday <- str_replace_all(dt$weekday, "4", "Fri")
dt$weekday <- str_replace_all(dt$weekday, "5", "Sat")
dt$weekday <- str_replace_all(dt$weekday, "6", "Sun")

dt$weekday <- factor(dt$weekday , levels=c("Mon", "Tue", "Wen", "Thu", "Fri", "Sat", "Sun"))
```

yr

```r
dt$yr <- sapply(dt$yr, function(x) {
        if (x == 0) {'2011'}
        else {'2012'}})
```

Print dataset head and tail

```r
head(dt)
```

```
##   instant     dteday   season   yr mnth holiday weekday workingday weathersit
## 1       1 2011-01-01 springer 2011  Gen       0     Sun    Holiday       Fair
## 2       2 2011-01-02 springer 2011  Gen       0     Mon    Holiday       Fair
## 3       3 2011-01-03 springer 2011  Gen       0     Tue    Workday       Good
## 4       4 2011-01-04 springer 2011  Gen       0     Wen    Workday       Good
## 5       5 2011-01-05 springer 2011  Gen       0     Thu    Workday       Good
## 6       6 2011-01-06 springer 2011  Gen       0     Fri    Workday       Good
##       temp    atemp      hum windspeed casual registered  cnt
## 1 0.344167 0.363625 0.805833 0.1604460    331        654  985
## 2 0.363478 0.353739 0.696087 0.2485390    131        670  801
## 3 0.196364 0.189405 0.437273 0.2483090    120       1229 1349
## 4 0.200000 0.212122 0.590435 0.1602960    108       1454 1562
## 5 0.226957 0.229270 0.436957 0.1869000     82       1518 1600
## 6 0.204348 0.233209 0.518261 0.0895652     88       1518 1606
```

```r
tail(dt)
```

```
##     instant     dteday   season   yr mnth holiday weekday workingday weathersit
```

```
## 726     726 2012-12-26 springer 2012  Dec       0     Thu    Workday        Bad
## 727     727 2012-12-27 springer 2012  Dec       0     Fri    Workday       Fair
## 728     728 2012-12-28 springer 2012  Dec       0     Sat    Workday       Fair
## 729     729 2012-12-29 springer 2012  Dec       0     Sun    Holiday       Fair
## 730     730 2012-12-30 springer 2012  Dec       0     Mon    Holiday       Good
## 731     731 2012-12-31 springer 2012  Dec       0     Tue    Workday       Fair
##         temp    atemp      hum windspeed casual registered  cnt
## 726 0.243333 0.220333 0.823333  0.316546      9        432  441
## 727 0.254167 0.226642 0.652917  0.350133    247       1867 2114
## 728 0.253333 0.255046 0.590000  0.155471    644       2451 3095
## 729 0.253333 0.242400 0.752917  0.124383    159       1182 1341
## 730 0.255833 0.231700 0.483333  0.350754    364       1432 1796
## 731 0.215833 0.223487 0.577500  0.154846    439       2290 2729
```

We remove 'holiday' column because 'workingday' column just contains the same informations. For the same reason we also remove 'dteday' column which contents are expressed by 'yr', 'mnth' and 'weekday' columns. In the end we remove the index column.

```r
df <- subset(dt, select = -c(instant, dteday, holiday) )
```

Check missing values

```r
df[rowSums(is.na(df)) > 0, ]
```

```
##  [1] season     yr         mnth       weekday    workingday weathersit
##  [7] temp       atemp      hum        windspeed  casual     registered
## [13] cnt
## <0 righe> (o 0-length row.names)
```

As we can see there aren't Null or missing values.

Descroptive Analysis

Now we start a explorative analysis of all variable in the dataset.

```r
summary(df)
```

```
##       season          yr                  mnth      weekday     workingday
##  fall    :188   Length:731         Gen    : 62   Mon:105   Holiday:231
##  springer:181   Class :character   Mar    : 62   Tue:105   Workday:500
##  summer  :184   Mode  :character   May    : 62   Wen:104
##  winter  :178                      Jul    : 62   Thu:104
##                                    Aug    : 62   Fri:104
##                                    Oct    : 62   Sat:104
##                                    (Other):359   Sun:105
##  weathersit      temp              atemp              hum
##  Bad : 21    Min.   :0.05913   Min.   :0.07907   Min.   :0.0000
##  Fair:247    1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200
```

```
##  Good:463   Median :0.49833   Median :0.48673   Median :0.6267
##             Mean   :0.49538   Mean   :0.47435   Mean   :0.6279
##             3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302
##             Max.   :0.86167   Max.   :0.84090   Max.   :0.9725
##
##    windspeed           casual         registered        cnt
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
##
```

From the summary we have some information about variables distributions. We noticed that in 'weathersit' column the most frequent class is 'Good', followed by 'Fair'. We have only 21 'Bad' weather oblservation. There aren't 'Very Bad' oberservations. We also noticed that 'casual' variable has the maximum value bigger than its third quartile. Here mean and median aren't equal. It means that distribution is not symmetrical and there are outliers.
The others variables don't show problems.

Variables

Now we try to understand variables distribution and their relationship with target variable. We use different plots and some hypothesis test to verify our assumptions.

For quantitative variables we plots histograms and boxplots for understanding their distribution and searching outliers. Then we use scatterplots for understanding their correlation with target variable. We test the normality distribution thanks to Shapiro-Wilk normality test.

The histograms numbers of bins are calculated thanks to Sturges formula:

$(numbers\, of\, bins) = 1 + log_2 n$

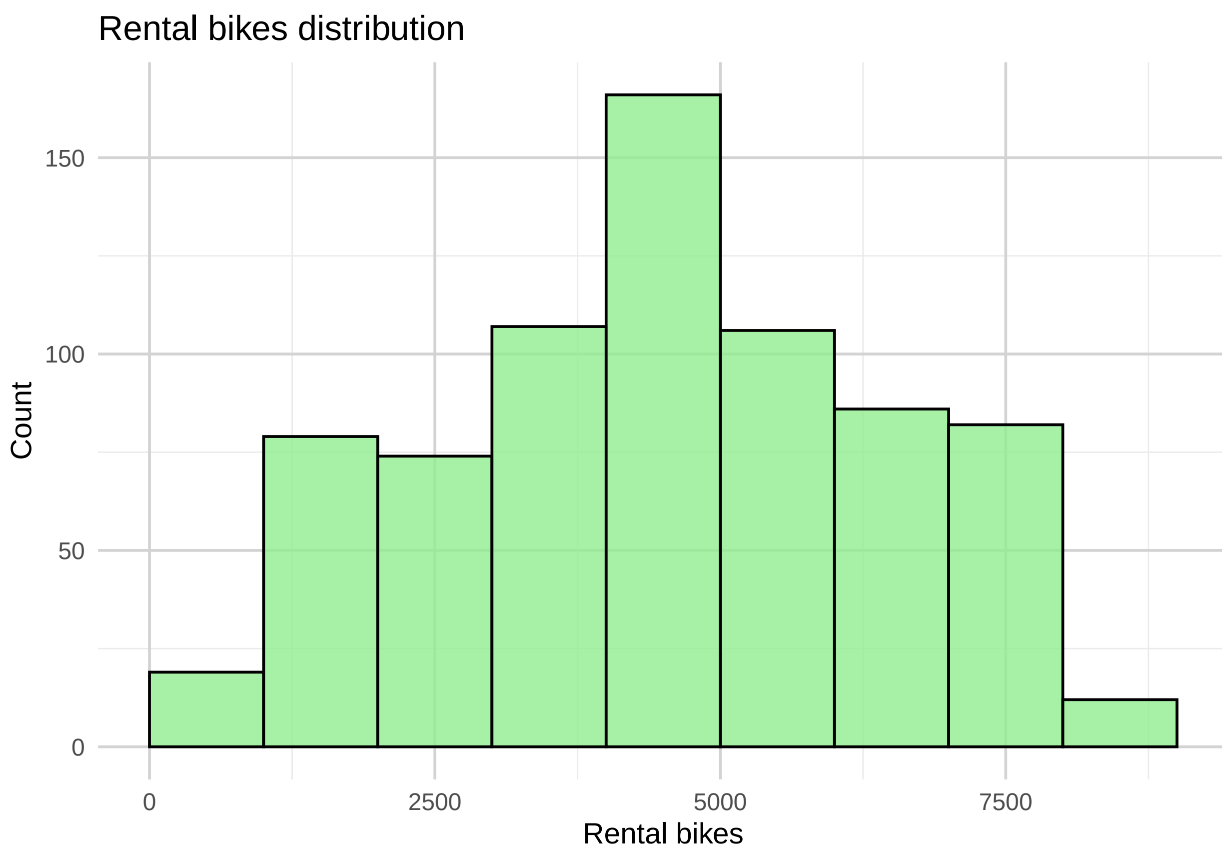
For qualitative variables we plots barcharts for distribution analysis and conditional boxplots for understanding their correlation with target variable.

Target variable

cnt

```r
brx_cnt <- pretty(range(df$cnt),
              n = nclass.Sturges(df$cnt), min.n = 1)

ggplot(df, aes(cnt)) + geom_histogram(color = "black", fill = 'lightgreen', alpha = 0.8, breaks = brx_cn
  labs(x = "Rental bikes", y = "Count", title = "Rental bikes distribution") +
  theme_minimal() + theme(panel.grid.major = element_line(color = "lightgrey"))
```



The target variable distribution is similar to a Normal distribution.

Quantitative variables

temp

```
brx_temp <- pretty(range(df$temp),
                   n = nclass.Sturges(df$temp), min.n = 1)
# Histogram
Temp1 <- ggplot(df) +
  geom_histogram(aes(x=temp), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_temp) +
  geom_vline(aes(xintercept=mean(temp)), color="white", linetype="dashed", size=1) +
  labs(title="",x="temp", y="Count") +
  theme_classic()
```

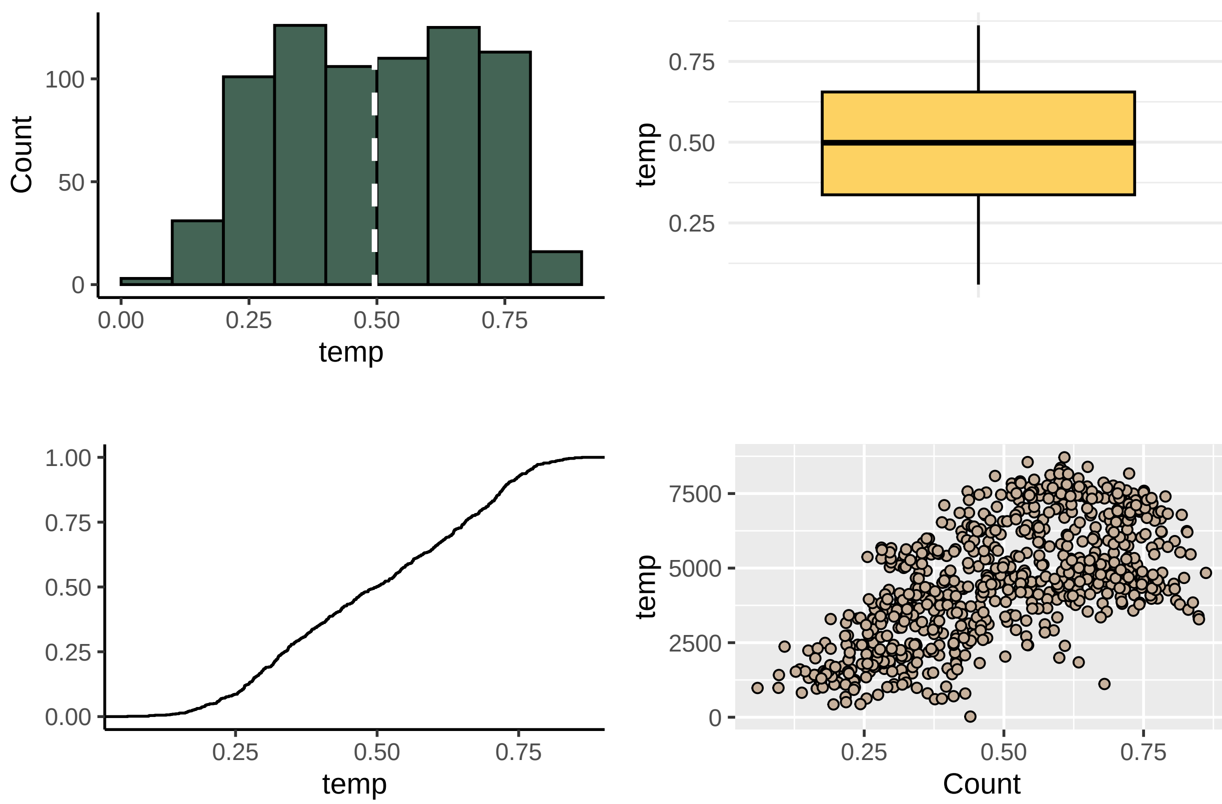
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```

```
# BoxPlot
Temp2 <- ggplot(df, aes(x = "", y=temp)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "temp")
  theme_minimal()

# ECDF
```

```
Temp3 <- ggplot(df, aes(temp)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="temp") +
  theme_classic()

# Scatterplot
Temp4 <- ggplot(df) +
  geom_point(aes(x=temp, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "temp", x="Count")
ggarrange(Temp1, Temp2, Temp3, Temp4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that 'temp' distribution is not a normal distribution. The scatterplot shows a positive linear correlation with target variable. We check the hypothesis of normal distribution thanks to Shapiro-Wilk normality test.

```
shapiro.test(df$temp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$temp
## W = 0.96591, p-value = 5.146e-12
```

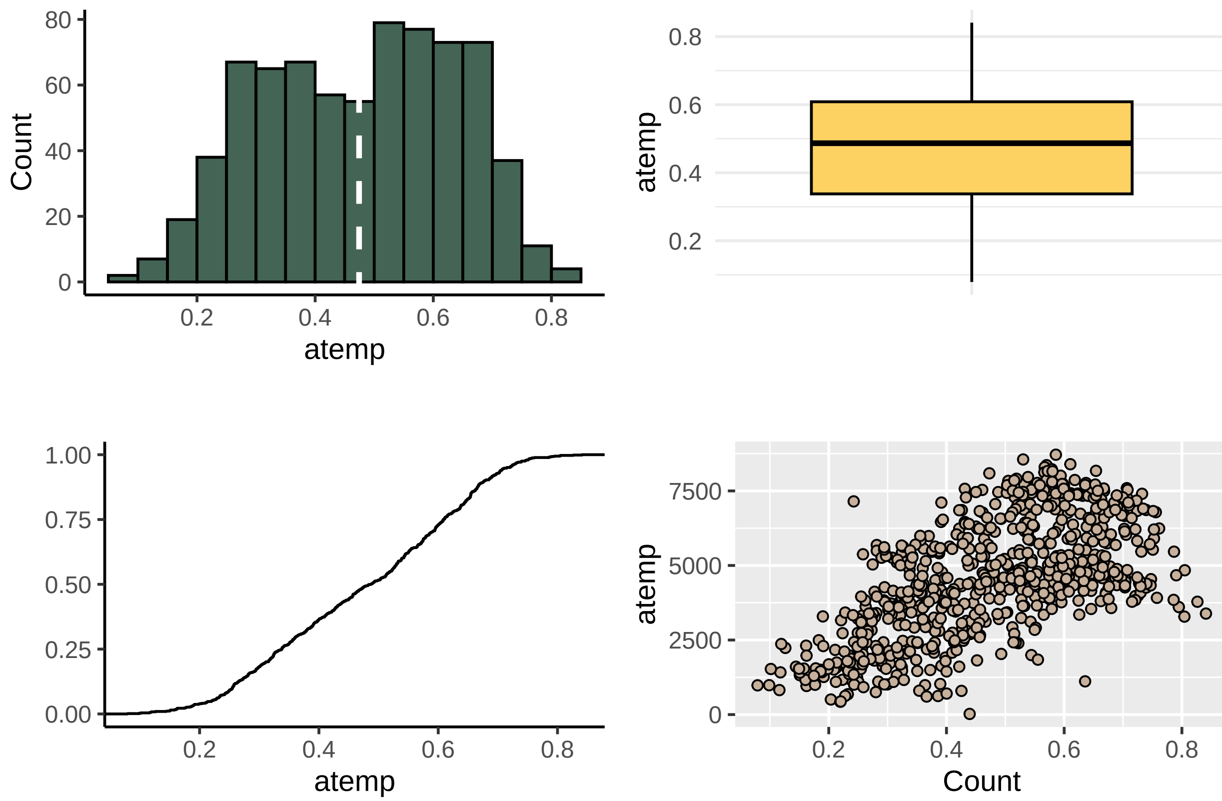
The Shapiro-Wilk normality test shows a very low p-value (5.013e-12). With this p-value we reject null hypothesis. We can't assume that 'temp' variable distribution is normal.

atemp

```
brx_at <- pretty(range(df$atemp),
              n = nclass.Sturges(df$atemp), min.n = 1)
# Histogram
Atemp1 <- ggplot(df) +
  geom_histogram(aes(x=atemp), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_at) +
  geom_vline(aes(xintercept=mean(atemp)), color="white", linetype="dashed", size=1) +
  labs(title="",x="atemp", y="Count") +
  theme_classic()
# BoxPlot
Atemp2 <- ggplot(df, aes(x = "", y=atemp)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "atemp"
  theme_minimal()

# ECDF
Atemp3 <- ggplot(df, aes(atemp)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="atemp") +
  theme_classic()

# Scatterplot
Atemp4 <- ggplot(df) +
  geom_point(aes(x=atemp, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "atemp", x="Count")
ggarrange(Atemp1, Atemp2, Atemp3, Atemp4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that 'atemp' distribution is not a normal distribution. The scatterplot shows a positive linear correlation with target variable. There aren't outliers. We check the hypothesis of normal distribution thanks to Shapiro-Wilk normality test.

```
shapiro.test(df$atemp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$atemp
## W = 0.97384, p-value = 3.744e-10
```

The Shapiro-Wilk normality test shows a very low p-value (3.744e-10). With this p-value we reject null hypothesis. We can't assume that 'atemp' variable distribution is normal.

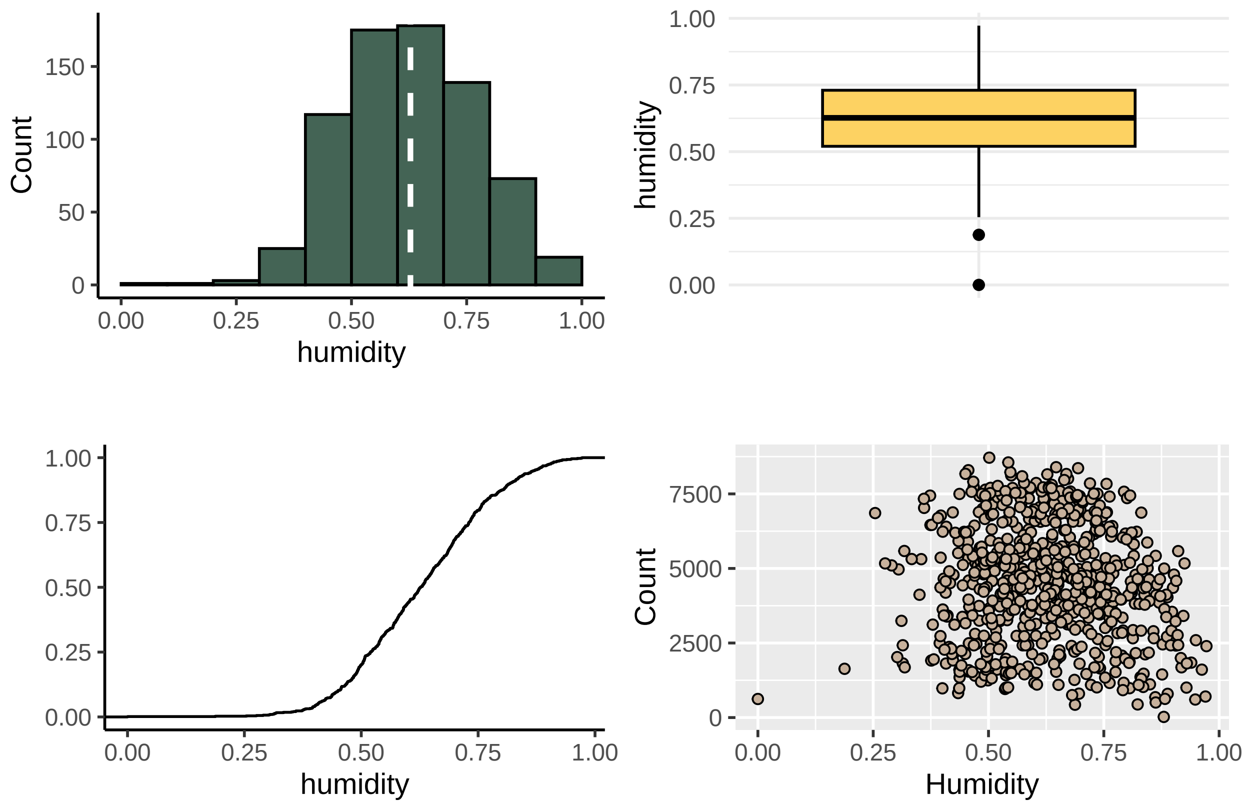
hum

```
brx_hum <- pretty(range(df$hum),
              n = nclass.Sturges(df$hum), min.n = 1)
# Histogram
Hum1 <- ggplot(df) +
  geom_histogram(aes(x=hum), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_hum) +
  geom_vline(aes(xintercept=mean(hum)), color="white", linetype="dashed", size=1) +
  labs(title="",x="humidity", y="Count") +
  theme_classic()
# BoxPlot
```

```
Hum2 <- ggplot(df, aes(x = "", y=hum)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "humidi
  theme_minimal()

# ECDF
Hum3 <- ggplot(df, aes(hum)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="humidity") +
  theme_classic()

# Scatterplot
Hum4 <- ggplot(df) +
  geom_point(aes(x=hum, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "Count", x="Humidity")
ggarrange(Hum1, Hum2, Hum3, Hum4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that ‘atemp’ distribution is a normal distribution. The scatterplot shows a linear negative correlation with target variable. There are 2 outliers beneath the second quartile. We check the hypothesis of normal distribution thanks to Shapiro-Wilk normality test.

```
shapiro.test(df$hum)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  df$hum
## W = 0.99335, p-value = 0.002481
```

The Shapiro-Wilk normality test shows a p-value of 0.002481. With this p-value we reject null hypothesis. We can't assume that 'hum' variable distribution is normal.

Now we analyse the outliers

```
boxplot.stats(df$hum)
```

```
## $stats
## [1] 0.2541670 0.5200000 0.6266670 0.7302085 0.9725000
##
## $n
## [1] 731
##
## $conf
## [1] 0.6143827 0.6389513
##
## $out
## [1] 0.187917 0.000000
```

The first value next to the second quartile is equal to 25% of humidity. It's a realistic value. We don't drop it.

```
ordered_Hum <- df[order(df$hum), ]
head(ordered_Hum)
```

```
##       season   yr mnth weekday workingday weathersit     temp    atemp      hum
## 69  springer 2011  Mar     Fri    Workday        Bad 0.389091 0.385668 0.000000
## 50  springer 2011  Feb     Sun    Holiday       Good 0.399167 0.391404 0.187917
## 463   summer 2012  Apr     Sun    Holiday       Good 0.437500 0.426129 0.254167
## 464   summer 2012  Apr     Mon    Holiday       Good 0.500000 0.492425 0.275833
## 452   summer 2012  Mar     Wen    Workday       Good 0.323333 0.315654 0.290000
## 87    summer 2011  Mar     Tue    Workday       Good 0.264348 0.257574 0.302174
##     windspeed casual registered  cnt
## 69   0.261877     46        577  623
## 50   0.507463    532       1103 1635
## 463  0.274871   3252       3605 6857
## 464  0.232596   2230       2939 5169
## 452  0.187192    531       4571 5102
## 87   0.212204    222       1806 2028
```

The furthest outlier is equal to 0% of humidity. Following scientific literature this value is impossibile in the earth atmosphere. [link] https://www.wunderground.com/cat6/world-record-low-humidity-116f-036-humidity-iran For this reason we drop this row.

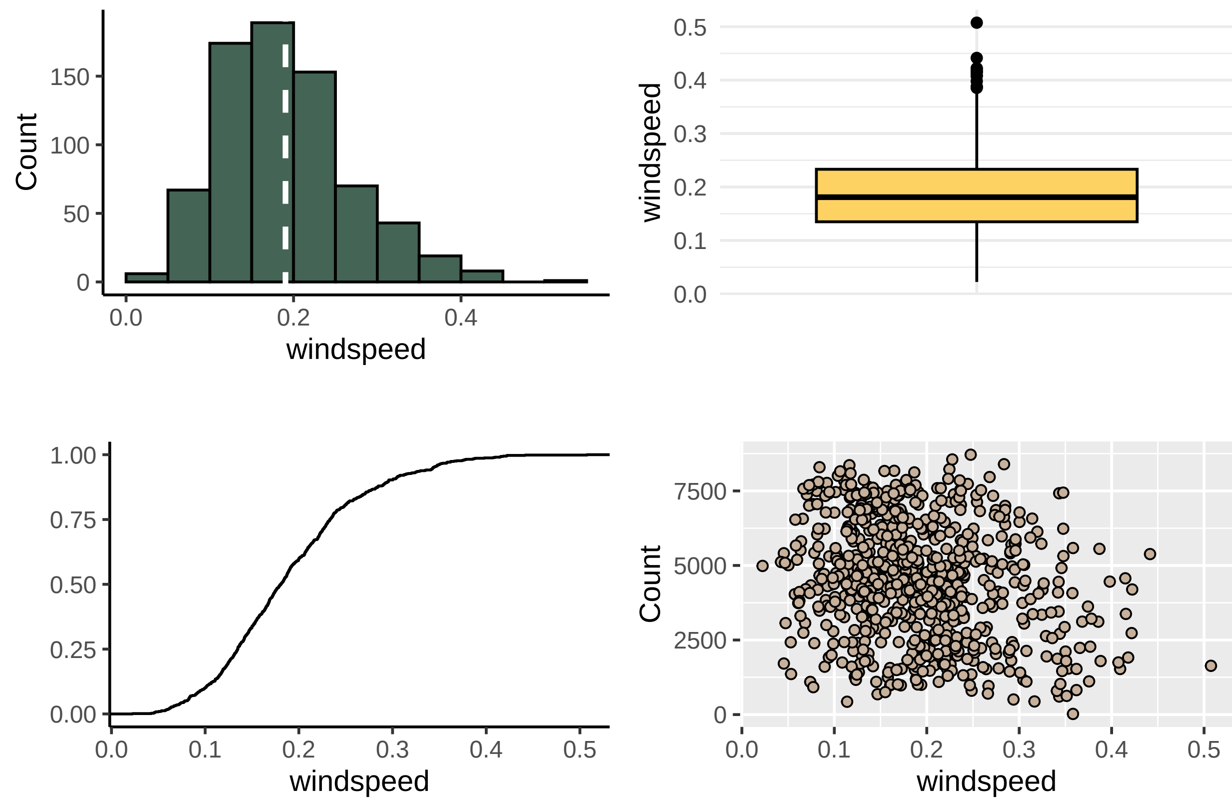
```
df <- df[-c(69), ]
```

windspeed

```
brx_ws <- pretty(range(df$windspeed),
              n = nclass.Sturges(df$windspeed), min.n = 1)
# Histogram
WS1 <- ggplot(df) +
  geom_histogram(aes(x=windspeed), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_ws)
  geom_vline(aes(xintercept=mean(windspeed)), color="white", linetype="dashed", size=1) +
  labs(title="",x="windspeed", y="Count") +
  theme_classic()
# BoxPlot
WS2 <- ggplot(df, aes(x = "", y=windspeed)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "windsp
  theme_minimal()

# ECDF
WS3 <- ggplot(df, aes(windspeed)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="windspeed") +
  theme_classic()

# Scatterplot
WS4 <- ggplot(df) +
  geom_point(aes(x=windspeed, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "Count", x="windspeed")
ggarrange(WS1, WS2, WS3, WS4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that ‘windspeed’ distribution isn’t a normal distribution. The scatterplot shows a linear negative correlation with target variable. There are several outliers above the third quartile. We check the hypothesis of normal distribution thanks to Shapiro-Wilk normality test.

```
shapiro.test(df$windspeed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$windspeed
## W = 0.97094, p-value = 7.311e-11
```

The Shapiro-Wilk normality test shows very low p-value (7.311e-11). With this p-value we reject null hypothesis. We can’t assume that ‘windspeed’ variable distribution is normal.

The reason why there isn’t a normal in distribution is the presence of a positive skewness, as we can see from the histogram.

We check positive skewness thanks to the Coefficient of Skewness.

```
skewness(df$windspeed)
```

```
## [1] 0.6793886
```

The Coefficient of Skewness is near to 1. It means that there is a positive skewness.

Now we analise the outliers

```
boxplot.stats(df$windspeed)
```

```
## $stats
## [1] 0.0223917 0.1349500 0.1809710 0.2332080 0.3781080
##
## $n
## [1] 730
##
## $conf
## [1] 0.175225 0.186717
##
## $out
##  [1] 0.417908 0.507463 0.385571 0.388067 0.422275 0.415429 0.409212 0.421642
##  [9] 0.441563 0.414800 0.386821 0.398008 0.407346
```

```
ordered_WindS <- df[order(df$windspeed), ]
tail(ordered_WindS)
```

```
##       season   yr mnth weekday workingday weathersit     temp    atemp      hum
## 383 springer 2012  Gen     Thu    Workday       Good 0.303333 0.275254 0.443333
## 45  springer 2011  Feb     Tue    Workday       Good 0.415000 0.398350 0.375833
## 421 springer 2012  Feb     Sun    Holiday       Good 0.290833 0.255675 0.395833
## 293   winter 2011  Oct     Fri    Workday       Good 0.475833 0.466525 0.636250
## 433 springer 2012  Mar     Fri    Workday       Good 0.527500 0.524604 0.567500
## 50  springer 2011  Feb     Sun    Holiday       Good 0.399167 0.391404 0.187917
##     windspeed casual registered  cnt
## 383  0.415429    109       3267 3376
## 45   0.417908    208       1705 1913
## 421  0.421642    317       2415 2732
## 293  0.422275    471       3724 4195
## 433  0.441563    486       4896 5382
## 50   0.507463    532       1103 1635
```

We know that 'windspeed' is a normalized variable. It's calculated dividing each value for the maximum value avaible (67). We dont't know the unit measure. For understandind if an outlier is a realistic value we split the problem into two cases: if the unit measure is km/h, the maximum value will be equal to 33.5 km/h; on the other side if the unit measure is mph, the maximum value will be equal to 54 km/h. In both cases, there are possibile values. For this reason we don't drop those rows.

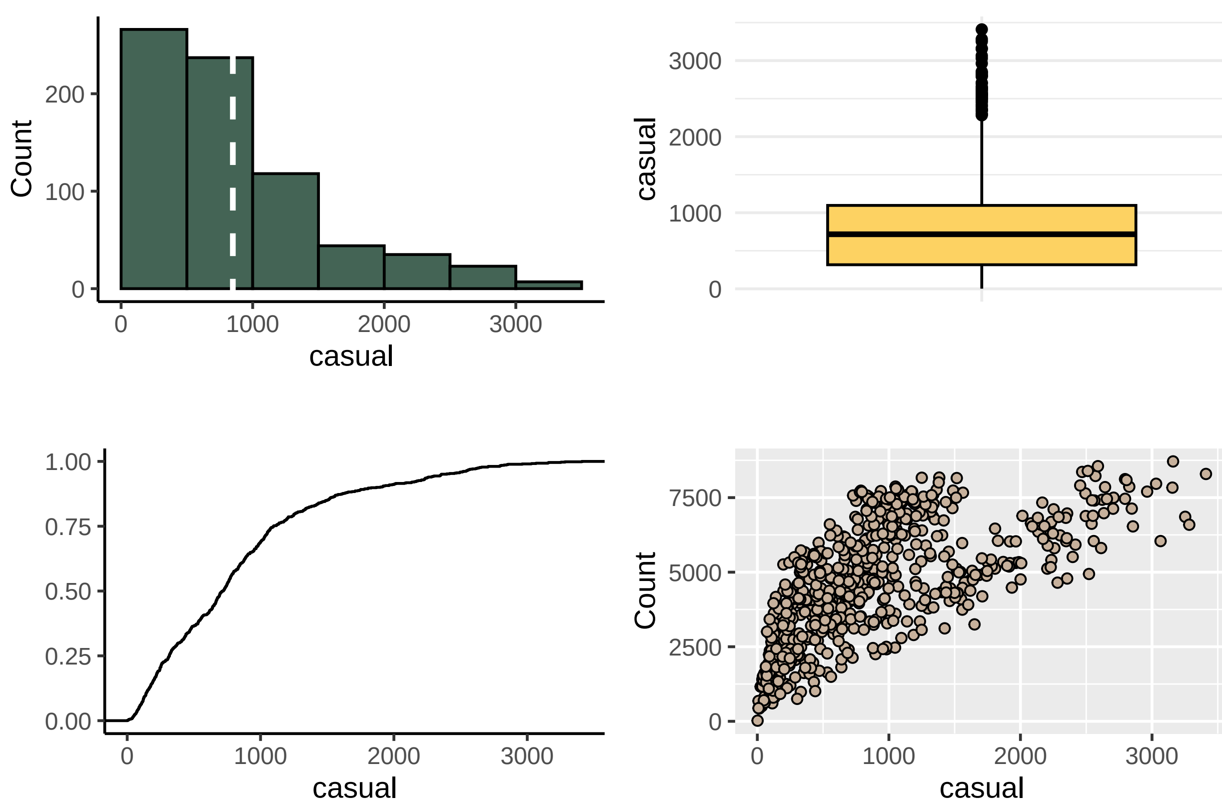
casual

```
brx_cas <- pretty(range(df$casual),
              n = nclass.Sturges(df$casual), min.n = 1)
# Histogram
Cas1 <- ggplot(df) +
  geom_histogram(aes(x=casual), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_cas) +
  geom_vline(aes(xintercept=mean(casual)), color="white", linetype="dashed", size=1) +
  labs(title="",x="casual", y="Count") +
  theme_classic()
# BoxPlot
Cas2 <- ggplot(df, aes(x = "", y=casual)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "casual
```

```
  theme_minimal()

# ECDF
Cas3 <- ggplot(df, aes(casual)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="casual") +
  theme_classic()

# Scatterplot
Cas4 <- ggplot(df) +
  geom_point(aes(x=casual, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "Count", x="casual")
ggarrange(Cas1, Cas2, Cas3, Cas4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that 'windspeed' distribution isn't a normal distribution. The scatterplot shows a linear positive correlation with target variable. There are several outliers above the third quartile. We check the hypothesis of normal distribution thanks to Shapiro-Wilk normality test.

```
shapiro.test(df$casual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$casual
```

```
## W = 0.88503, p-value < 2.2e-16
```

The Shapiro-Wilk normality test shows very low p-value (2.2e-16). With this p-value we reject null hypothesis. As we said We can't assume that 'windspeed' variable distribution is normal.

The reason why there isn't a normal in distribution is the presence of a positive skewness, as we can see from the histogram.

We check positive skewness thanks to the Coefficient of Skewness.

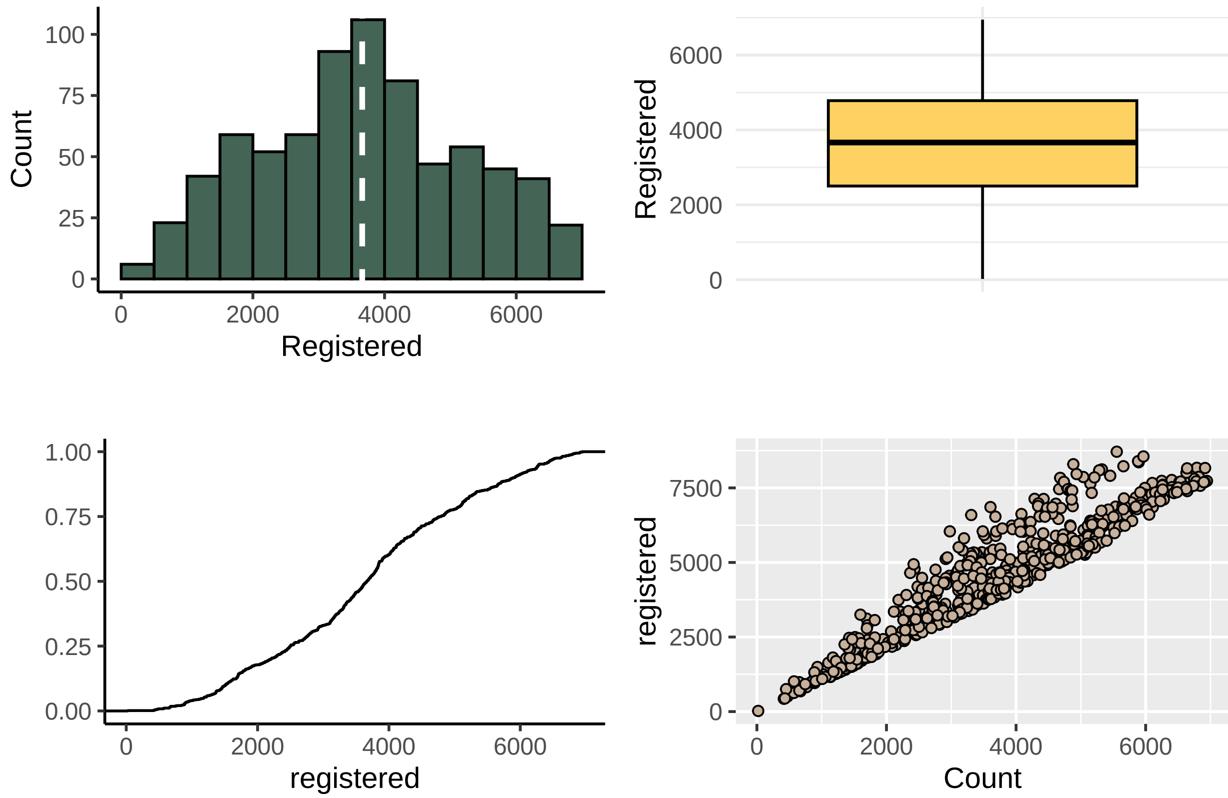
```
skewness(df$casual)
```

```
## [1] 1.263929
```

As we noticed there is a high positive skewness.

registered

```
brx_reg <- pretty(range(df$registered),
              n = nclass.Sturges(df$registered), min.n = 1)
# Histogram
Reg1 <- ggplot(df) +
  geom_histogram(aes(x=registered), fill = wes_palette("Chevalier1")[1], color="black", breaks = brx_reg
  geom_vline(aes(xintercept=mean(registered)), color="white", linetype="dashed", size=1) +
  labs(title="",x="Registered", y="Count") +
  theme_classic()
# Box Plot
Reg2 <- ggplot(df, aes(x = "", y=registered)) +
  geom_boxplot(fill=wes_palette("Chevalier1")[2], color="black") + labs(title = "", x = "", y = "Regist
  theme_minimal()
# ECDF
Reg3 <- ggplot(df, aes(registered)) +
  stat_ecdf(geom="step") +
  labs(title="", y = "", x="registered") +
  theme_classic()
# Scatterplot
Reg4 <- ggplot(df) +
  geom_point(aes(x=registered, y=cnt), shape=21, fill=wes_palette("Chevalier1")[4], color="black") +
  labs(title="", y = "registered", x="Count")
ggarrange(Reg1, Reg2, Reg3, Reg4,
          ncol = 2,
          nrow = 2)
```



From histogram and ECDF we assume that ‘registered’ distribution is a normal distribution. The scatterplot shows a linear positive correlation with target variable. There aren’t outliers.

We check skewness thanks to the Coefficient of Skewness.

```
skewness(df$registered)
```

```
## [1] 0.04637257
```

The coefficient is near to 0. We can assume that ‘registered’ variable has a symmetrical distribution.

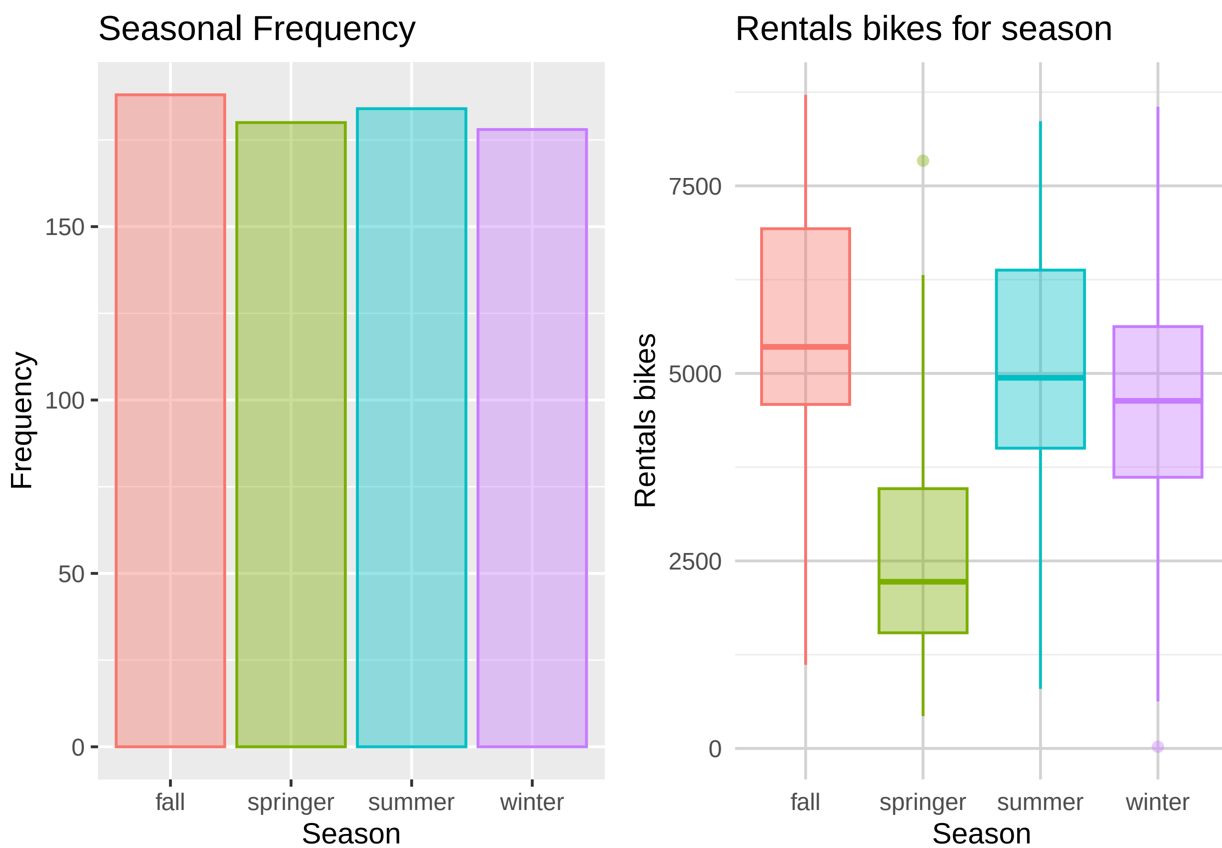
Qualitative variables

season

```
Season1 <- ggplot(df, aes(season)) + geom_bar(aes(color = season, fill = season), alpha = 0.4) +
  labs(x = "Season", y = "Frequency", title = "Seasonal Frequency") + theme(legend.position = "none")


Season2 <- ggplot(df, aes(season, cnt)) + geom_boxplot(aes(color = season, fill = season), alpha = 0.4)
  labs(x = "Season", y = "Rentals bikes", title = "Rentals bikes for season") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey"

ggarrange(Season1, Season2,
          ncol = 2,
          nrow = 1)
```



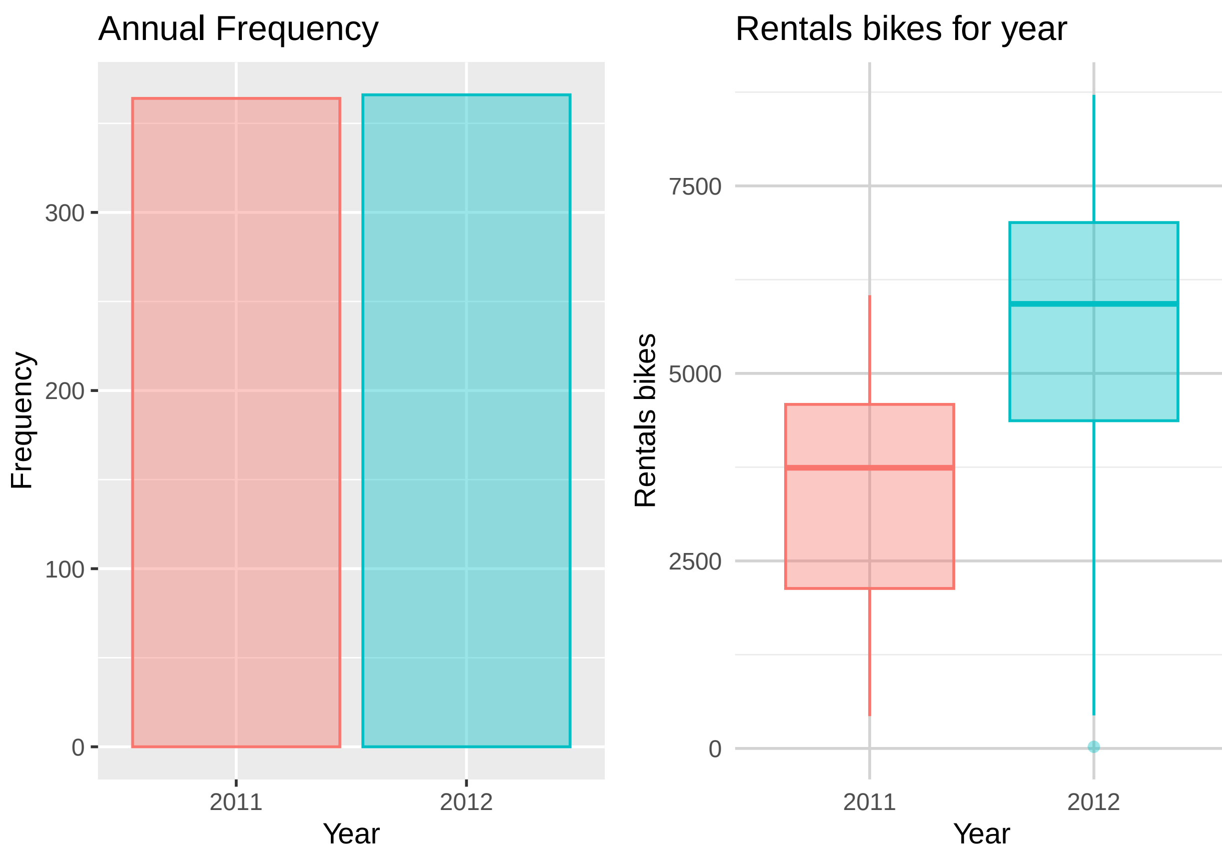
There are more rentals bikes in fall than in the other season. On the other side, during the springer there is the lowest number of rentals bikes. There are 2 outliers in springer and winter season.

yr

```
Year1 <- ggplot(df, aes(yr)) + geom_bar(aes(color = yr, fill = yr), alpha = 0.4) +
  labs(x = "Year", y = "Frequency", title = "Annual Frequency") + theme(legend.position = "none")


Year2 <- ggplot(df, aes(yr, cnt)) + geom_boxplot(aes(color = yr, fill = yr), alpha = 0.4) +
  labs(x = "Year", y = "Rentals bikes", title = "Rentals bikes for year") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey"

ggarrange(Year1, Year2,
          ncol = 2,
          nrow = 1)
```



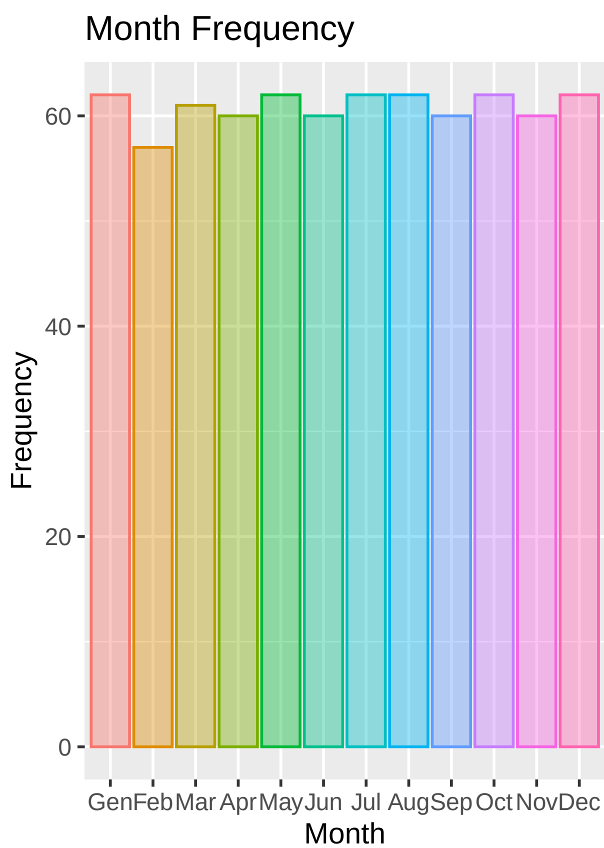
The the number of rentals bikes increase over the years. In 2012 there are more or less the double of rentals.

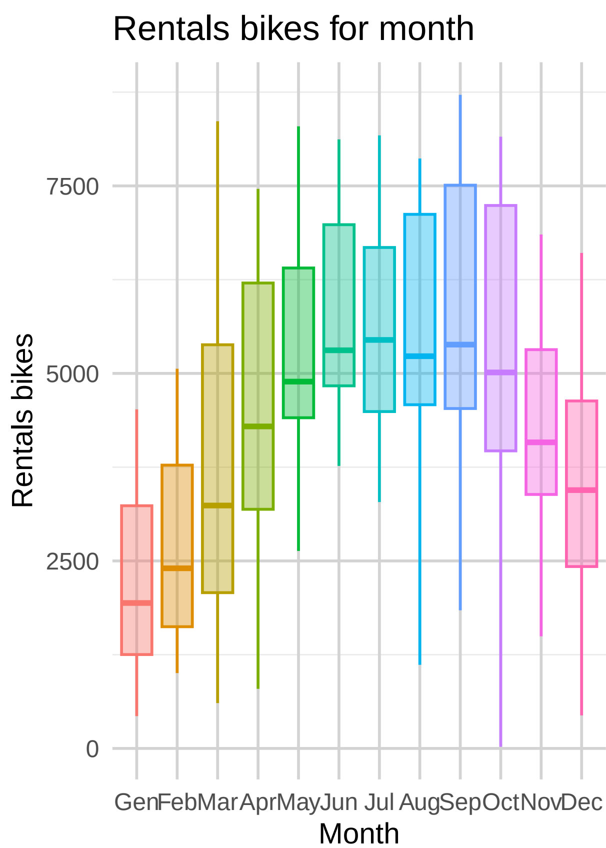
mnth

```
Month1 <- ggplot(df, aes(mnth)) + geom_bar(aes(color = mnth, fill = mnth), alpha = 0.4) +
  labs(x = "Month", y = "Frequency", title = "Month Frequency") + theme(legend.position = "none")


Month2 <- ggplot(df, aes(mnth, cnt)) + geom_boxplot(aes(color = mnth, fill = mnth), alpha = 0.4) +
  labs(x = "Month", y = "Rentals bikes", title = "Rentals bikes for month") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey")

ggarrange(Month1, Month2,
          ncol = 2,
          nrow = 1)
```





From the contiditional boxplots we assume that the rentals bikes depends on the month. From June to September there are the highest values. The lowest values are in the winter months.

We check this assumption with an hypothesis test:

$$H_0 : \mu_i = \mu_k \;\; for \; \forall \, i, k \; in \, \{Gen, Feb, ..., Dec\}$$

$$H_1 : at\, least\, one\, equivalence\, in\, H_0\, is\, not\, true$$

```
an <- aov(cnt~as.factor(mnth), data = df)
summary(an)
```

```
##                  Df    Sum Sq  Mean Sq F value Pr(>F)
## as.factor(mnth)  11 1.065e+09 96789231   41.87 <2e-16 ***
## Residuals       718 1.660e+09  2311655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

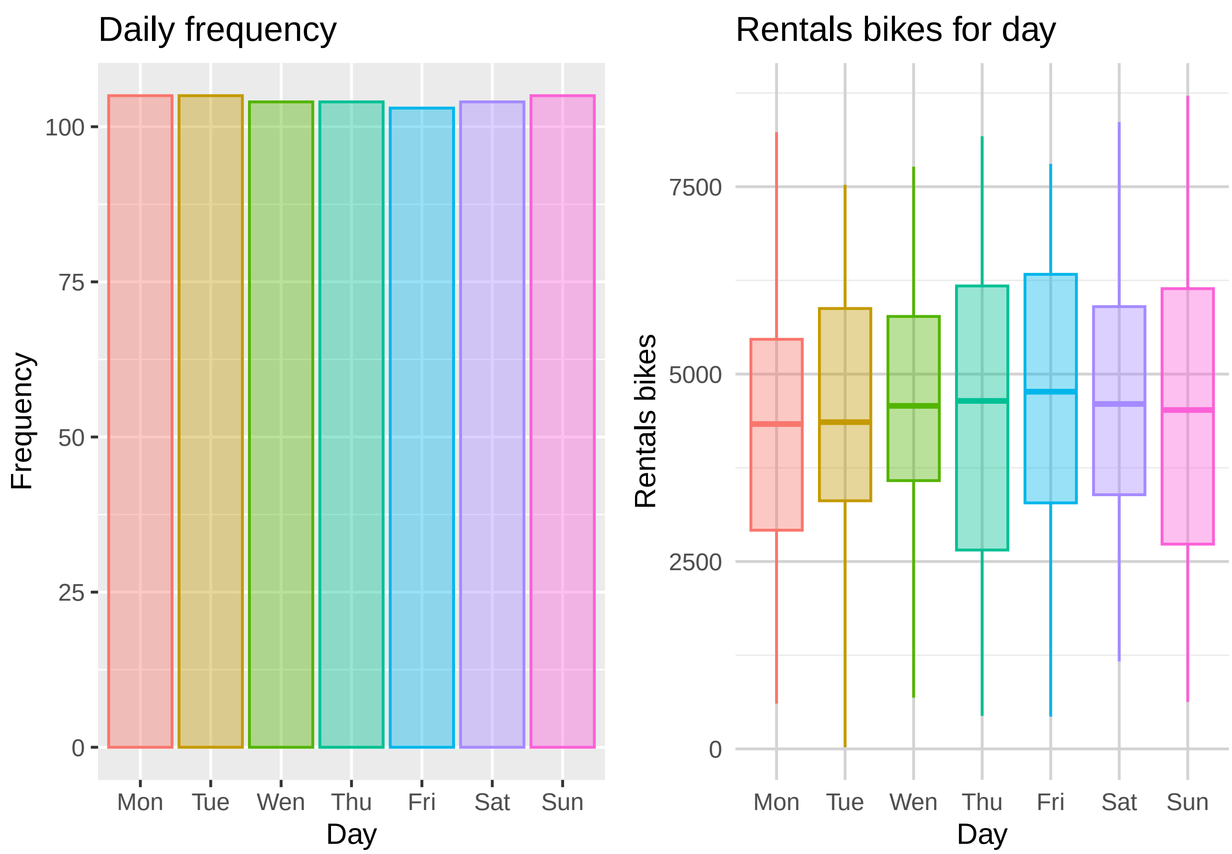
We reject the null hypothesis. So month averages are not the equal to each other. We assume that the target variable is dependent on mean to 'mnth' variable.

weekday

```
WD1 <- ggplot(df, aes(weekday)) + geom_bar(aes(color = weekday, fill = weekday), alpha = 0.4) +
  labs(x = "Day", y = "Frequency", title = "Daily frequency") + theme(legend.position = "none")
```

```r
WD2 <- ggplot(df, aes(weekday, cnt)) + geom_boxplot(aes(color = weekday, fill = weekday), alpha = 0.4) +
  labs(x = "Day", y = "Rentals bikes", title = "Rentals bikes for day") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey"

ggarrange(WD1, WD2,
          ncol = 2,
          nrow = 1)
```



From the contiditional boxplots we assume that the rentals bikes don' t depend on the day. Indeed the daily means are kind of equal.

We check this assumption with an hypothesis test:

$$H_0 : \mu_i = \mu_k \;\; for \; \vee \, i, k \; in \, \{Mon, Tue, ..., Sun\}$$

$$H_1 : at\, least\, one\, equivalence\, in\, H_0\, is\, not\, true$$

```r
an <- aov(cnt~as.factor(weekday), data = df)
summary(an)
```

```
##                     Df    Sum Sq Mean Sq F value Pr(>F)
## as.factor(weekday)   6 1.909e+07 3181391    0.85  0.531
## Residuals          723 2.705e+09 3741856
```

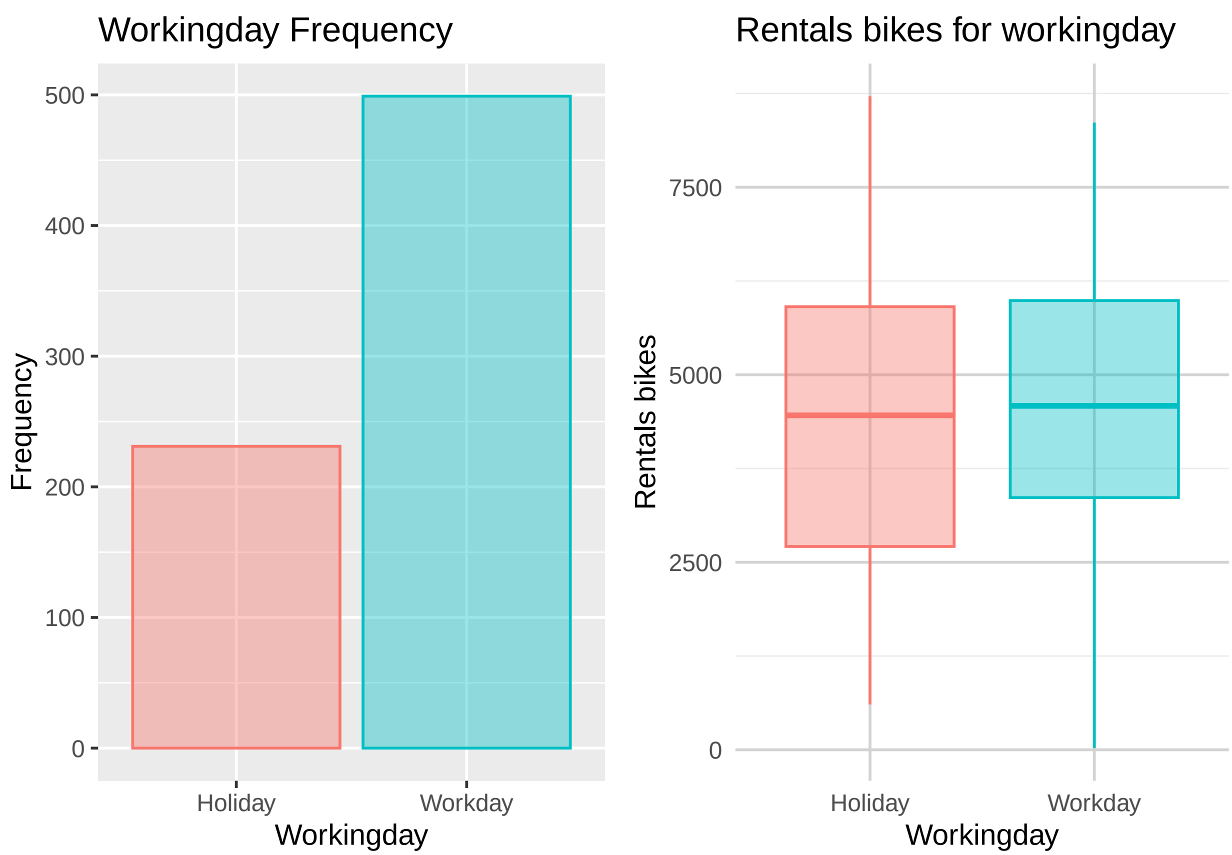
With a pvalue of 0.531, we can't reject the null hypothesis. So daily means are equal to each other. We assume that the target variable is independent on mean to 'weekday' variable. So we don't use this variable in the linear regression model.

workingday

```
WRD1 <- ggplot(df, aes(workingday)) + geom_bar(aes(color = workingday, fill = workingday), alpha = 0.4)
  labs(x = "Workingday", y = "Frequency", title = "Workingday Frequency") + theme(legend.position = "no


WRD2 <- ggplot(df, aes(workingday, cnt)) + geom_boxplot(aes(color = workingday, fill = workingday), alpl
  labs(x = "Workingday", y = "Rentals bikes", title = "Rentals bikes for workingday") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey"

ggarrange(WRD1, WRD2,
          ncol = 2,
          nrow = 1)
```



From the contiditional boxplots we assume that the rentals bikes don't depend on the workingday variable. Indeed the means of both classes are kind of the same.

We check this assumption with an hypothesis test:

$$H_0 : \mu_w = \mu_h$$

$$H_1 : The\, equivalence\, in\, H_0\, is\, not\, true$$

```
an <- aov(cnt~as.factor(workingday), data = df)
summary(an)
```

```
##                          Df    Sum Sq   Mean Sq F value Pr(>F)
```

```
## as.factor(workingday)   1 1.089e+07 10888005   2.921 0.0879 .
## Residuals             728 2.714e+09  3727420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

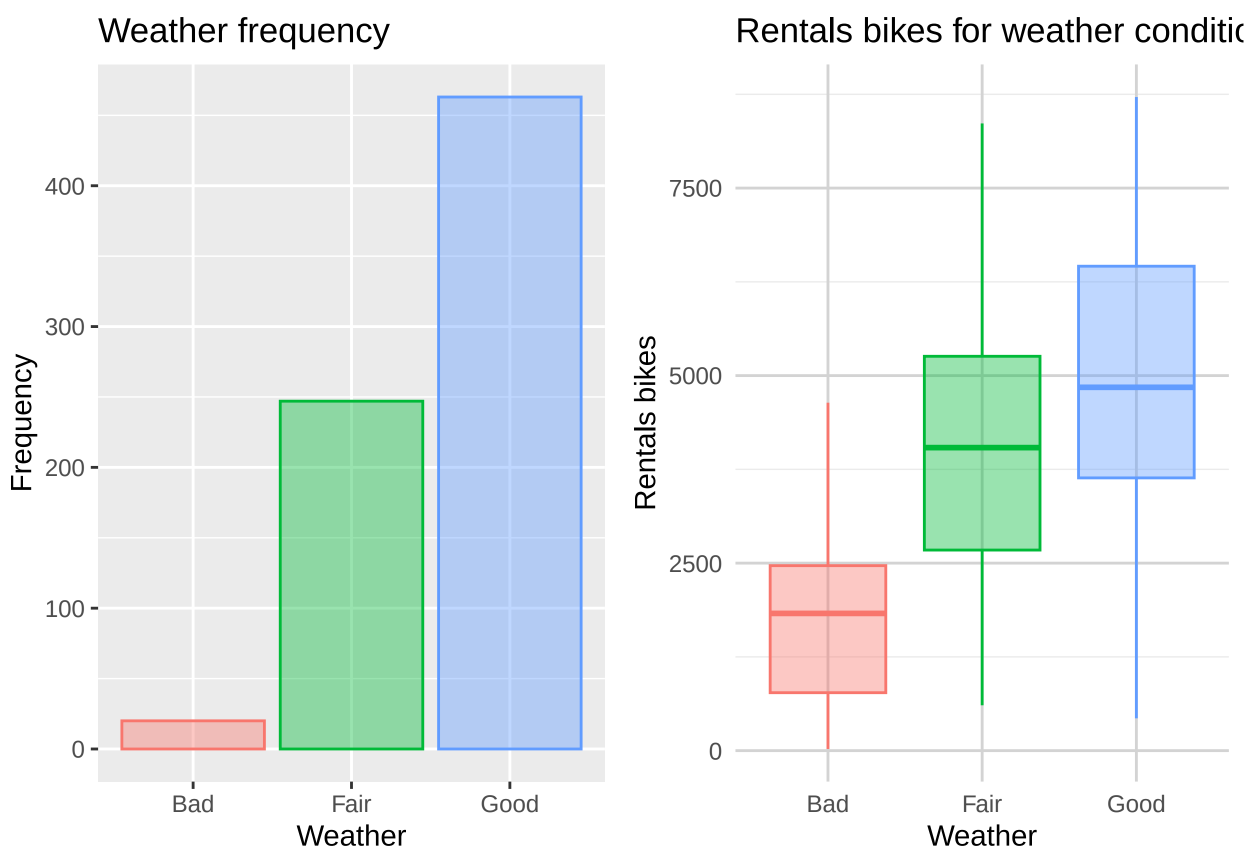
Using a confidence of 95%, we can't reject the null hypothesis. So workingday means are equal to each other. We assume that the target variable is independent on mean to 'workingday' variable. So we don't use this variable in the linear regression model.

weathersit

```
WS1 <- ggplot(df, aes(weathersit)) + geom_bar(aes(color = weathersit, fill = weathersit), alpha = 0.4) +
  labs(x = "Weather", y = "Frequency", title = "Weather frequency") + theme(legend.position = "none")


WS2 <- ggplot(df, aes(weathersit, cnt)) + geom_boxplot(aes(color = weathersit, fill = weathersit), alpha
  labs(x = "Weather", y = "Rentals bikes", title = "Rentals bikes for weather condition") +
  theme_minimal() + theme(legend.position = "none", panel.grid.major = element_line(color = "lightgrey")

ggarrange(WS1, WS2,
          ncol = 2,
          nrow = 1)
```



The number of rentals bikes depend on weather conditions. Indeed it increases with better weather conditions.

We check this assumption with an hypothesis test:

$$H_0 : \mu_b = \mu_F = \mu_G$$

$$H_1 : At\, least\, one\, equivalence\, in\, H_0\, is\, not\, true$$

```r
an <- aov(cnt~as.factor(weathersit), data = df)
summary(an)
```

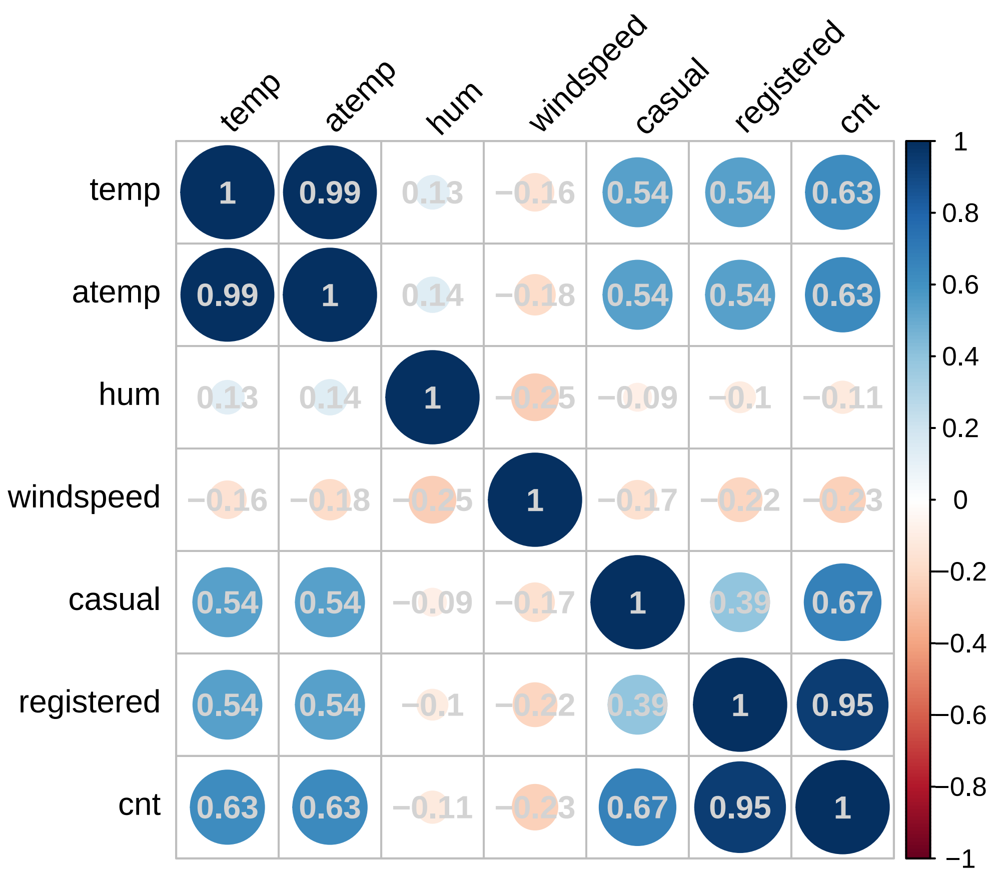
```
##                         Df    Sum Sq   Mean Sq F value Pr(>F)
## as.factor(weathersit)    2 2.580e+08 129010898   38.03 <2e-16 ***
## Residuals              727 2.466e+09   3392611
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a pvalue of 2e-16 we reject the null hypothesis. The target variable is dependent on means to 'weathersit' variable.

Heatmap

We check possible high correlation between variables. We use a heatmap.

```r
corrplot(cor(df[, -c(1:6)]),
         addCoef.col = "lightgrey",
         tl.col="black", tl.srt=45)
```



We can observe a high correlation between 'registered' and 'casual' variable related to the target variable. We'll not use those two variables because they will not be very useful in the modeling process as these are just number of users and unlikely to be the factor that directly causes rise in the number of bike rentals. There also is autocorrelation between 'atemp' and 'temp' variable. We won't use them in the regression model.

Linear Regression

Now we build a linear regression model with the backward process. We build a first model including all variables, except for 'atemp', 'casual', 'registered', 'weekday' and 'workingday'. Then we'll drop variables that aren't significant and we compare models using Test F.

```
df1 <- subset(df, select = -c(atemp, casual, registered, weekday, workingday))

View(df1)
```

Build the model

```
cnt.lm_all <- lm(cnt ~ ., data=df1)
summary(cnt.lm_all)
```

```
##
## Call:
## lm(formula = cnt ~ ., data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3895.3  -362.8    90.6   480.8  3017.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1001.31     421.06   2.378 0.017666 *
## seasonspringer  -848.00     218.18  -3.887 0.000111 ***
## seasonsummer      36.13     189.11   0.191 0.848557
## seasonwinter     766.71     195.47   3.922 9.62e-05 ***
## yr2012          2007.10      59.65  33.647  < 2e-16 ***
## mnthFeb          149.98     146.89   1.021 0.307582
## mnthMar          596.64     169.13   3.528 0.000446 ***
## mnthApr          458.23     252.63   1.814 0.070127 .
## mnthMay          735.70     272.96   2.695 0.007201 **
## mnthJun          498.63     286.54   1.740 0.082263 .
## mnthJul          -33.32     318.74  -0.105 0.916763
## mnthAug          404.04     306.83   1.317 0.188329
## mnthSep          966.57     269.70   3.584 0.000362 ***
## mnthOct          486.90     246.48   1.975 0.048608 *
## mnthNov         -147.01     235.20  -0.625 0.532135
## mnthDec          -86.58     186.03  -0.465 0.641790
## weathersitFair  1378.15     195.02   7.067 3.80e-12 ***
## weathersitGood  1784.19     211.75   8.426  < 2e-16 ***
## temp            4586.32     419.13  10.943  < 2e-16 ***
## hum            -1772.25     312.00  -5.680 1.96e-08 ***
```

```
## windspeed      -3029.20     416.96  -7.265 9.86e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 786.5 on 709 degrees of freedom
## Multiple R-squared:  0.839,  Adjusted R-squared:  0.8345
## F-statistic: 184.7 on 20 and 709 DF,  p-value: < 2.2e-16
```

The first model with 20 variables have high level of R^2 and R^2 *Adjusted* and its F-statistic is significant, so there is at least one coefficient significantly different from zero. We noticed that some coefficients are not significant. We drop that variables which have p-value over 0.30. Then we compare the models.

Dummy

Now we exclude the irrelevant dummy levels that we observed before. We build two new variables that aggregate all irrelevant levels in one.

```
df1$reduced_season <- df1$season
df1$reduced_season[which(df1$reduced_season == 'summer')] <- 'fall'

df1$reduced_mnth <- df1$mnth
df1$reduced_mnth[which(df1$mnth == 'Feb')] <- 'Gen'
df1$reduced_mnth[which(df1$mnth == 'Jul')] <- 'Gen'
df1$reduced_mnth[which(df1$mnth == 'Nov')] <- 'Gen'
df1$reduced_mnth[which(df1$mnth == 'Dec')] <- 'Gen'

df2 <-  subset(df1, select = -c(season, mnth))


# Build a new model
cnt.lm_1 <- lm(cnt ~ ., data=df2)
```

Now we compare the two models using a Test F. It works with ANOVA function.

```
anova(cnt.lm_all, cnt.lm_1, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: cnt ~ season + yr + mnth + weathersit + temp + hum + windspeed
## Model 2: cnt ~ yr + weathersit + temp + hum + windspeed + reduced_season +
##     reduced_mnth
##   Res.Df       RSS Df Sum of Sq      F Pr(>F)
## 1    709 438624607
## 2    714 440108082 -5  -1483475 0.4796 0.7916
```

The pvalue is not significant. We can't reject the null hypothesis. So we can't say that the second model coefficients are significantly different from zero. We can consider the second model goog as much as the first one.

```r
summary(cnt.lm_1)
```

```
##
## Call:
## lm(formula = cnt ~ ., data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3898.7  -365.2    94.0   493.0  2987.6
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1003.37     382.01   2.627  0.00881 **
## yr2012                  2007.02      59.15  33.931  < 2e-16 ***
## weathersitFair          1381.86     194.34   7.110 2.81e-12 ***
## weathersitGood          1783.17     210.85   8.457  < 2e-16 ***
## temp                    4563.47     299.81  15.221  < 2e-16 ***
## hum                    -1794.61     304.81  -5.888 6.03e-09 ***
## windspeed              -3033.51     414.31  -7.322 6.61e-13 ***
## reduced_seasonspringer  -792.58     140.34  -5.647 2.35e-08 ***
## reduced_seasonwinter     675.23     131.17   5.148 3.41e-07 ***
## reduced_mnthMar          594.62     115.03   5.169 3.06e-07 ***
## reduced_mnthApr          516.57     142.85   3.616  0.00032 ***
## reduced_mnthMay          798.71     134.17   5.953 4.13e-09 ***
## reduced_mnthJun          549.92     133.52   4.118 4.26e-05 ***
## reduced_mnthAug          432.79     132.98   3.254  0.00119 **
## reduced_mnthSep         1018.80     124.37   8.191 1.19e-15 ***
## reduced_mnthOct          602.74     128.77   4.681 3.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 785.1 on 714 degrees of freedom
## Multiple R-squared:  0.8385, Adjusted R-squared:  0.8351
## F-statistic: 247.1 on 15 and 714 DF,  p-value: < 2.2e-16
```

The model coefficients are all significant. It keeps high value of R^2 and R^2 *Adjusted*

Multicollinearity of indipendent variables: VIF

We evaluate the collinearity between variables using VIF index.

$$VIF_i = 1/(1 - R_i^2)$$

VIF measures how much the variance of an estimated regression coefficient is increased because of collinearity. The values higher than 10 represent multicollinearity of indipendent variables

```r
ols_vif_tol(cnt.lm_1)
```

```
##                Variables  Tolerance       VIF
## 1                 yr2012 0.96538563  1.035855
## 2         weathersitFair 0.09986475 10.013543
```

```
## 3          weathersitGood 0.08187266 12.214089
## 4                    temp 0.28048964  3.565194
## 5                     hum 0.46027849  2.172598
## 6               windspeed 0.82000818  1.219500
## 7  reduced_seasonspringer 0.23076414  4.333429
## 8    reduced_seasonwinter 0.26617709  3.756897
## 9         reduced_mnthMar 0.83323957  1.200135
## 10        reduced_mnthApr 0.54854239  1.823013
## 11        reduced_mnthMay 0.60350901  1.656976
## 12        reduced_mnthJun 0.62782197  1.592808
## 13        reduced_mnthAug 0.61437471  1.627671
## 14        reduced_mnthSep 0.72360178  1.381976
## 15        reduced_mnthOct 0.65519175  1.526271
```

'weatherstiGood' and 'weatherstiFair' variables have VIF higher than 10. There is collinearity. Now we drop 'weatherstiGood' and build another model.

```
df1$reduced_weathersit <- df1$weathersit
df1$reduced_weathersit[which(df1$reduced_weathersit == 'Good')] <- 'Bad'

df3 <-  subset(df1, select = -c(season, mnth, weathersit))

# Build a new model
cnt.lm_2 <- lm(cnt ~ ., data=df3)
summary(cnt.lm_2)
```

```
##
## Call:
## lm(formula = cnt ~ ., data = df3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4726.3  -379.1   105.1   521.0  2872.0
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3338.87     276.66  12.069  < 2e-16 ***
## yr2012                   1996.81      61.98  32.215  < 2e-16 ***
## temp                     4960.99     310.36  15.985  < 2e-16 ***
## hum                     -3062.21     278.19 -11.008  < 2e-16 ***
## windspeed               -3894.79     420.93  -9.253  < 2e-16 ***
## reduced_seasonspringer  -662.41     146.21  -4.530 6.90e-06 ***
## reduced_seasonwinter     779.00     136.88   5.691 1.84e-08 ***
## reduced_mnthMar          626.90     120.51   5.202 2.58e-07 ***
## reduced_mnthApr          579.46     149.52   3.875 0.000116 ***
## reduced_mnthMay          941.13     139.52   6.745 3.15e-11 ***
## reduced_mnthJun          559.68     139.95   3.999 7.02e-05 ***
## reduced_mnthAug          481.53     139.25   3.458 0.000577 ***
## reduced_mnthSep         1054.59     130.29   8.094 2.48e-15 ***
## reduced_mnthOct          540.62     134.75   4.012 6.66e-05 ***
## reduced_weathersitFair  -141.31      76.52  -1.847 0.065206 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 822.9 on 715 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8188
## F-statistic: 236.3 on 14 and 715 DF,  p-value: < 2.2e-16
```

```
ols_vif_tol(cnt.lm_2)
```

```
##                  Variables Tolerance      VIF
## 1                   yr2012 0.9657882 1.035424
## 2                     temp 0.2875580 3.477560
## 3                      hum 0.6070796 1.647230
## 4                windspeed 0.8727424 1.145813
## 5  reduced_seasonspringer 0.2335736 4.281306
## 6    reduced_seasonwinter 0.2685269 3.724022
## 7         reduced_mnthMar 0.8341584 1.198813
## 8         reduced_mnthApr 0.5500329 1.818073
## 9         reduced_mnthMay 0.6131692 1.630871
## 10        reduced_mnthJun 0.6278689 1.592689
## 11        reduced_mnthAug 0.6155311 1.624613
## 12        reduced_mnthSep 0.7244402 1.380376
## 13        reduced_mnthOct 0.6573307 1.521304
## 14 reduced_weathersitFair 0.7076874 1.413053
```

In the cnt.lm_2 model there isn't collinearity anymore. Now 'reduced_weatherFair' is not significant.

Now we compare the two models using a Test F. It works with ANOVA function.

```
anova(cnt.lm_1, cnt.lm_2, test = 'F')
```

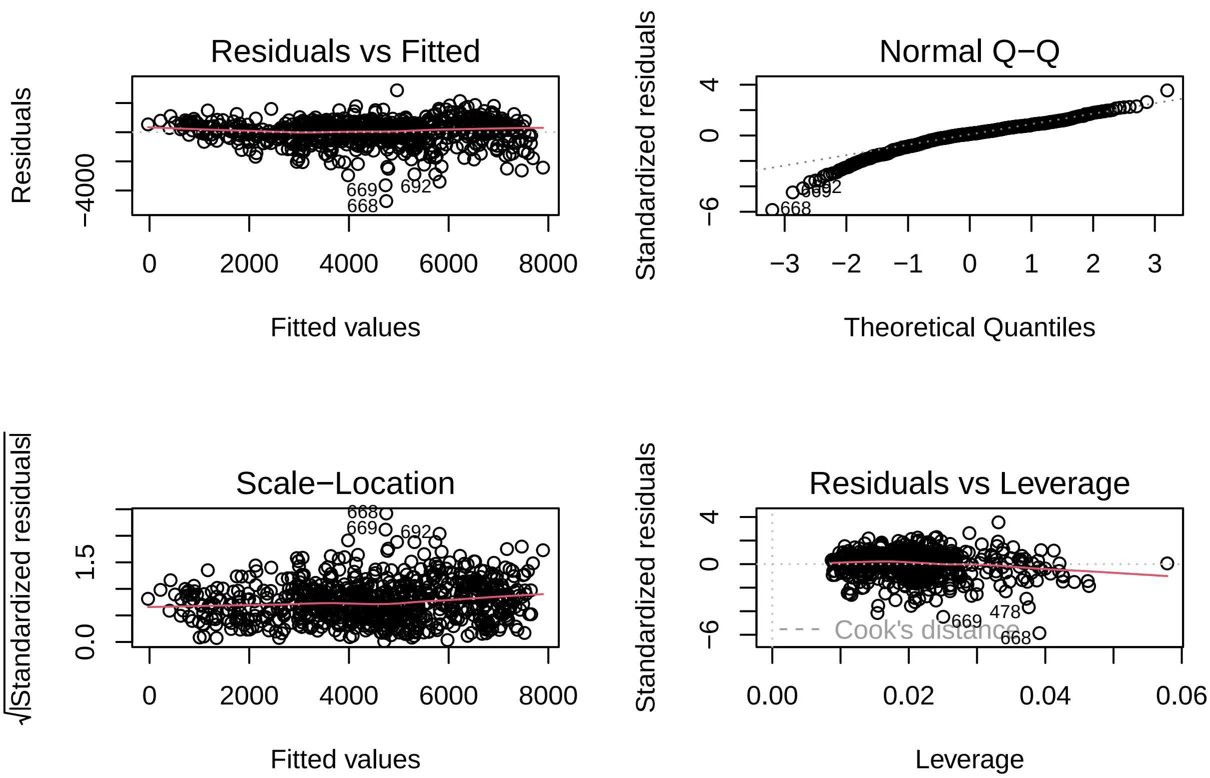
```
## Analysis of Variance Table
##
## Model 1: cnt ~ yr + weathersit + temp + hum + windspeed + reduced_season +
##     reduced_mnth
## Model 2: cnt ~ yr + temp + hum + windspeed + reduced_season + reduced_mnth +
##     reduced_weathersit
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1    714 440108082
## 2    715 484193249 -1 -44085168 71.521 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is significant. We must reject the null hypothesis. So we can't say that the cnt.lm_2 model is good as much as the cnt.lm_1 model. We must keep cnt.lm_1.

Residuals analysis

Now we evaluate the cnt.lm_1 model and we check if it satisfies the Classic linear regression models assumptions. We don't modify the model.

```r
par(mfrow = c(2, 2))
plot(cnt.lm_2)
```



Residual vs Fitted The residuals distribution shape doesn't suggest the presence of heteroscedasticity. From this plot we can also observe the presence of linearity.

Now we check with the White test the presence of residuals heteroscedasticity.

White Test

```r
white.test<-function(lmod){
    u2<-lmod$residuals^2
    y<-lmod$fitted
    R2u<-summary(lm(u2~y+I(y^2)))$r.squared
    LM<-length(y)*R2u
    p.val<-1-pchisq(LM,2)
    data.frame("Test Statistic"=LM, "P"=p.val)
}

white.test(cnt.lm_1)
```

```
##   Test.Statistic          P
## 1       8.806523 0.01223736
```

With a confidence of 95% we reject the null hypothesis. The model has homoskedastic residuals.

Q-Q Plot The Q-Q plot is useful to understand the presence of normality in the model shape. In our case the residuals are distributed along the red line, so we could assume the presence of normality in distribution. There is a heavy left tail that is very different from the other residuals.

Now we use the Shapiro-Wilks normality Test to control the presence of normality in residuals distribution. The null hypothesis is:

$$H_0 : our\, residuals\, are\, normally\, distributed$$

Test di Shapiro-Wilks

```
shapiro.test(cnt.lm_1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  cnt.lm_1$residuals
## W = 0.95138, p-value = 8.83e-15
```

We reject the null hypothesis. We can't assume that residuals are normally distributed

Scale-Location

This plot is useful to verify the homoskedasticity of residuals. Indeed, when the red line is horizontal the null hypothesis is satisfied. We verified this assumption in the chunk above.

Residuals vs Leverage

We use this plot to verify the presence of outliers. There are many bordelines values.

As we can see there aren't leverage points. A leverage point is a value in [0,1] range. In this case the most part of residuals are in [-3,3] range.

There are many values that could be problematic. For example we see that the 668 and 669 obervations can be considered as outliers. So we could try to drop them and fit another model. The elimination of outliers could change residuals distribution to a normal one.

It will be useful make a deeper analysis of outliers.

Residuals autocorrelation: Durbin-Watson test

Now we use the Durbin-Watson test to check if the residuals are correlated each other. This index is in a [0,4] range. Typically if a value is near 2, there will not be correlation. On the otherside a value near to 0 means that there is a positive correlation between residuals.

```
durbinWatsonTest(cnt.lm_1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1        0.413243        1.1731       0
##  Alternative hypothesis: rho != 0
```

In this case there is a positive correlation between residuals. A possibile reason of this result is that the dataset is composed by time aggregated values. This kind of data need specific corrections or use specific models based on temporal analysis.

Conclusion

This is the end of our analysis. We explored the dataset and all its variables.

We build a model for forecasting the rentals bikes and we discovered that 'weekday', 'workingday' and 'atemp' variables are irrelevant. The model has a great measure of R^2. Unfortunately it doesn't satisfy the assumptions of residuals normality and autocorrelation. More analysis are required.

We could build another model using based on temporal analysis and compare it to linear regression model to verify the useless of some variables.

In the end, it could be a great opportunity analyse outliers and strange values together with a domain expert.