Xi'an Weather Multiple Linear Regression

Chi Ting Low

7/14/2018

Introduction

In recent year, cities in China have encouter the problem of smug or fog. This increase the city's pollution and many concern about the influence of the smug or fog. The case study shown here is one of the city in China Xi'an. Due to it's location, Xi'an has a temperate climate that is influenced by the East Asian monsoon with less wind and rain. Xi'an also a city with heavy industry complex and it is less environmental friendly. The aims of the current study is to understand what influence the air quality. This project is also aimed to study which factor contribute to air quality index.

```
#loading packaged
library(readxl)
library(dplyr)
library(tidyverse)

source("https://raw.githubusercontent.com/iascchen/VisHealth/master/R/calendarHeat.R")

#read data
weather_xian <- read_xlsx("2016-2017   .xlsx", na = "NA")

#checking missing values
anyNA(weather_xian)
```

```
## [1] FALSE
```

Date Preprocessing

Prior the analysis, we have to recode the variable name into correct values. Due to some unique character that is not identifiable by R. Therefore, these Chinese chracter is recode into english. In addition, we also remove special character in the temperature. Once the processes are completed, the data is tranform into correct data type.

```
#rename column
colnames(weather_xian) <- c('Date', "Days","Higest_temperature",
                            "Lowest_temperature", "Weather",
                            "Wind_direction", "Wind_speed",
                            "Air_quality_index", "Air_quality")

#remove character in highest and lowest temperature
weather_xian$Higest_temperature = unlist(strsplit(weather_xian$Higest_temperature,
                                                  split = "℃", fixed = TRUE))
weather_xian$Lowest_temperature = unlist(strsplit(weather_xian$Lowest_temperature,
                                                  split = "℃", fixed = TRUE))

#recoding data
weather_xian$Days <- recode(weather_xian$Days, "  " = "Sunday",
                            "  " = "Monday", "  " = "Tuesday",
```

```
                              "  " = "Wednesday", "  " = "Thursday",
                              "  " = 'Friday', "  " = "Saturday" )

weather_xian$Wind_speed <- recode(weather_xian$Wind_speed, '0' = 'Calm',
                                  '1-2 ' = 'Light Breeze',
                                  '3-4 ' = 'Moderate Wind',
                                  '4-5 ' = 'Strong Wind')

weather_xian$Wind_direction <- recode(weather_xian$Wind_direction, ' ' = 'East',
                                      ' ' = 'North East',
                                      ' ' = "South East", ' ' = 'North',
                                      ' ' = 'South', '  ' = 'Unpredicted',
                                      ' ' = 'West', ' ' = 'North West',
                                      ' ' = 'South West')

weather_xian$Air_quality <- recode(weather_xian$Air_quality,
                                   '   ' = 'Serious pollution',
                                   '   ' = 'Moderately pollution',
                                   ' ' = 'Excellent', ' ' = 'Good',
                                   '   ' = 'Mild pollution',
                                   '   ' = 'Severe pollution')

weather_xian$Weather <- recode(weather_xian$Weather, '   ~ ' = 'raining',
                               ' ' = 'raining', ' ~ ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'raining',
                               ' ' = 'cloudy', ' ~   ' = 'cloudy',
                               ' ~ ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~ ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~   ' = 'cloudy', ' ~ ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'raining',
                               '   ~ ' = 'raining', '   ~ ' = 'raining',
                               '   ~ ' = 'raining', ' ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'raining',
                               ' ~   ' = 'raining', ' ~ ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'snowing',
                               ' ~ ' = 'snowing', ' ' = 'sunny',
                               ' ~ ' = 'sunny', ' ~ ' = 'sunny',
                               ' ~ ' = 'sunny', ' ~ ' = 'raining',
                               ' ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~ ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~   ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~ ' = 'cloudy', ' ~ ' = 'cloudy',
                               ' ~ ' = 'cloudy', ' ~   ' = 'cloudy',
                               ' ' = 'raining', ' ~ ' = 'raining',
                               ' ~ ' = 'raining', ' ~ ' = 'raining',
                               '   ' = 'raining', '   ~ ' = 'raining',
                               '   ~ ' = 'raining', ' ' = 'fog',
                               ' ~ ' = 'fog', ' ~ ' = 'fog', ' ~ ' = 'fog' )

#convert into right data type
weather_xian$Days <- as.factor(weather_xian$Days)
weather_xian$Higest_temperature <- as.numeric(weather_xian$Higest_temperature)
```

```
weather_xian$Lowest_temperature <- as.numeric(weather_xian$Lowest_temperature)
weather_xian$Air_quality_index <- as.numeric(weather_xian$Air_quality_index)
weather_xian$Weather <- as.factor(weather_xian$Weather)
weather_xian$Wind_direction <- as.factor(weather_xian$Wind_direction)
weather_xian$Wind_speed <- as.factor(weather_xian$Wind_speed)
weather_xian$Air_quality <- as.factor(weather_xian$Air_quality)

str(weather_xian)
```

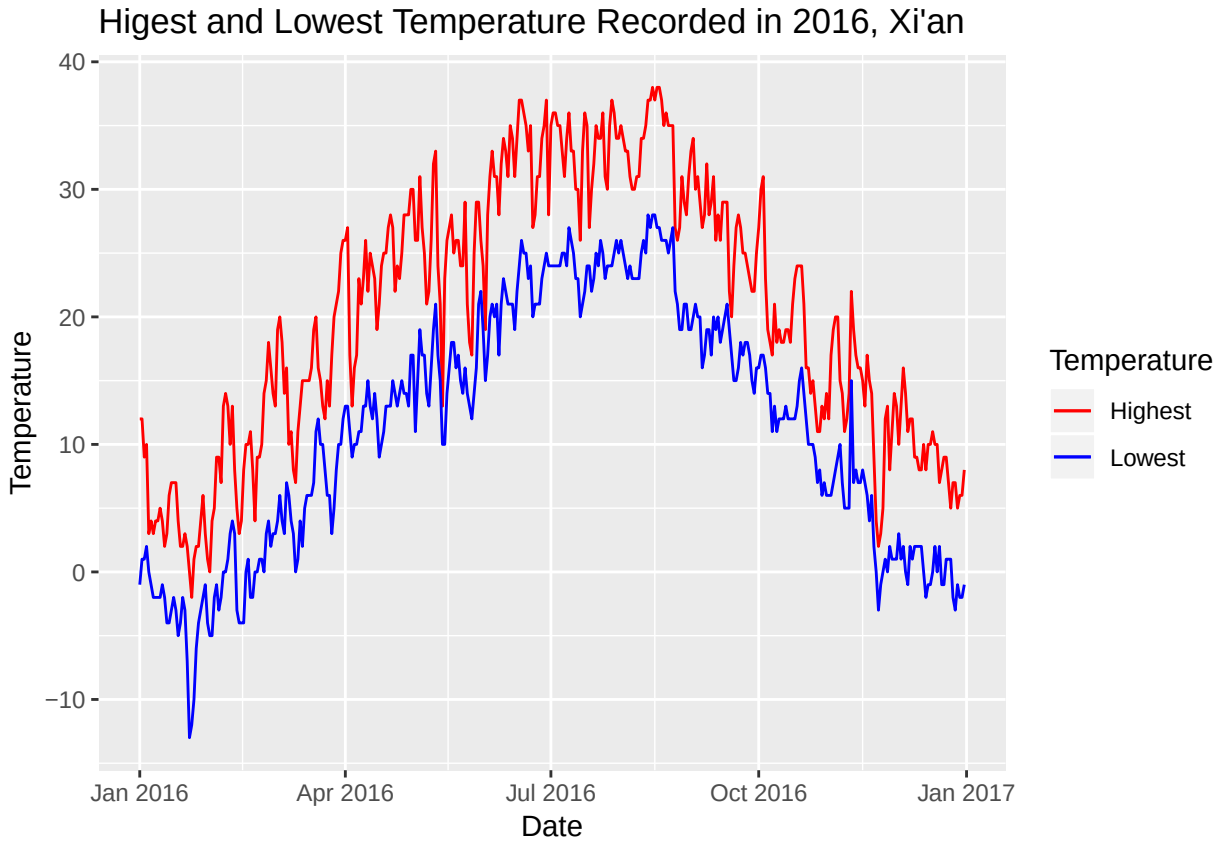
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    712 obs. of  9 variables:
##  $ Date              : POSIXct, format: "2017-12-01" "2017-12-02" ...
##  $ Days              : Factor w/ 7 levels "Friday","Monday",..: 1 3 4 2 6 7 5 1 3 4 ...
##  $ Higest_temperature: num  7 10 13 9 8 8 8 5 8 9 ...
##  $ Lowest_temperature: num  -4 -1 -2 -3 -4 -3 -3 -4 -3 -5 ...
##  $ Weather           : Factor w/ 5 levels "cloudy","fog",..: 5 1 5 1 5 1 1 1 5 5 ...
##  $ Wind_direction    : Factor w/ 9 levels "East","North",..: 2 7 5 3 3 7 7 2 7 7 ...
##  $ Wind_speed        : Factor w/ 4 levels "Calm","Light Breeze",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Air_quality_index : num  160 229 197 132 107 123 133 62 101 96 ...
##  $ Air_quality       : Factor w/ 6 levels "Excellent","Good",..: 4 6 4 3 3 3 3 2 3 2 ...
```

Date exploration

Below are the plot of the highest and lowest temperature recored on the day in 2016 and 2017. It shows that the highest temperature are recorded from Apri to August. However, for the highest air quality index are occurded during December to next year January.

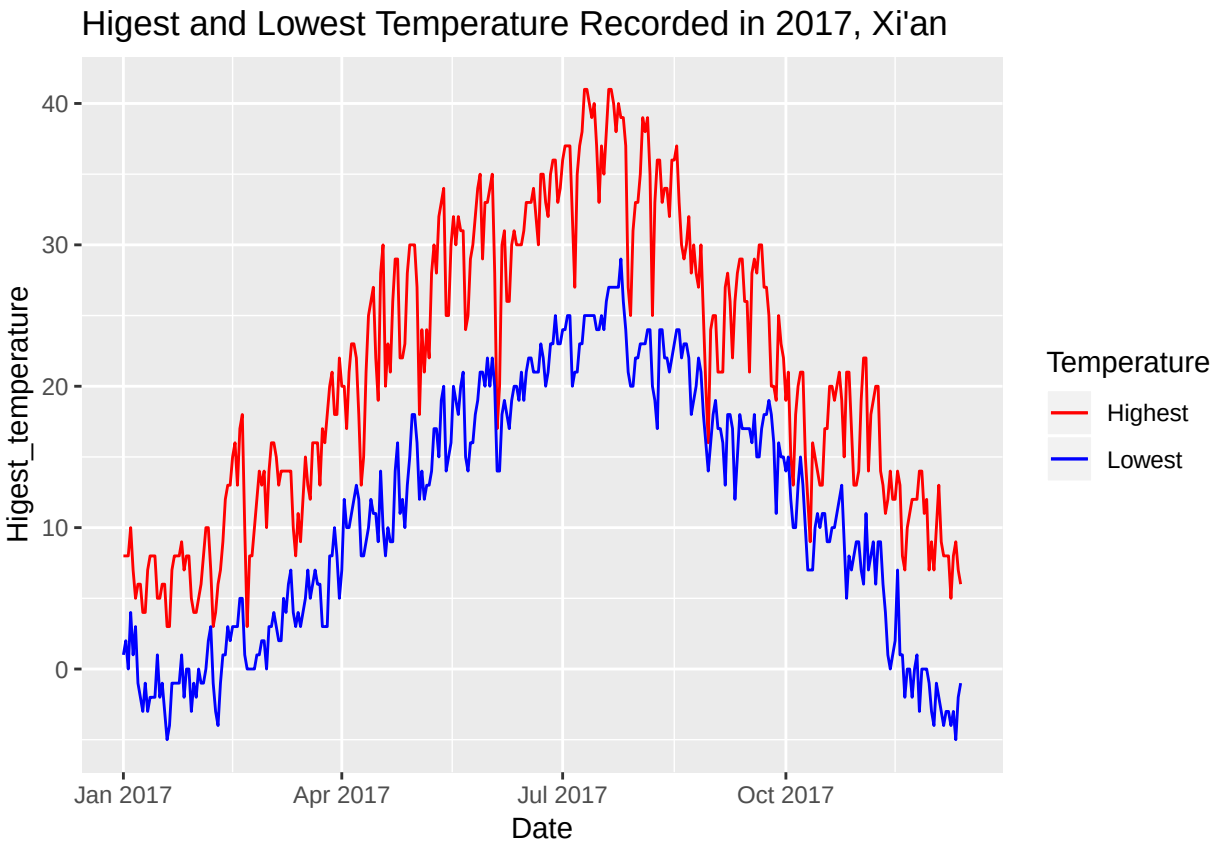
```
weather_2016 = weather_xian[weather_xian$Date < "2017-01-01",]
weather_2016$Date <- as.Date(weather_2016$Date)

weather_2016 %>%
  ggplot() +
  geom_line(aes(x = Date,
                y = Higest_temperature,
                colour = 'blue')) +
  geom_line(aes(x = Date,
                y = Lowest_temperature,
                colour = 'red')) +
  labs(title = "Higest and Lowest Temperature Recorded in 2016, Xi'an",
       x = "Date",
       y = 'Temperature',
       color = "Temperature" ) +
  scale_color_manual(labels = c("Highest", "Lowest"),
                     values = c("red", "blue"))
```

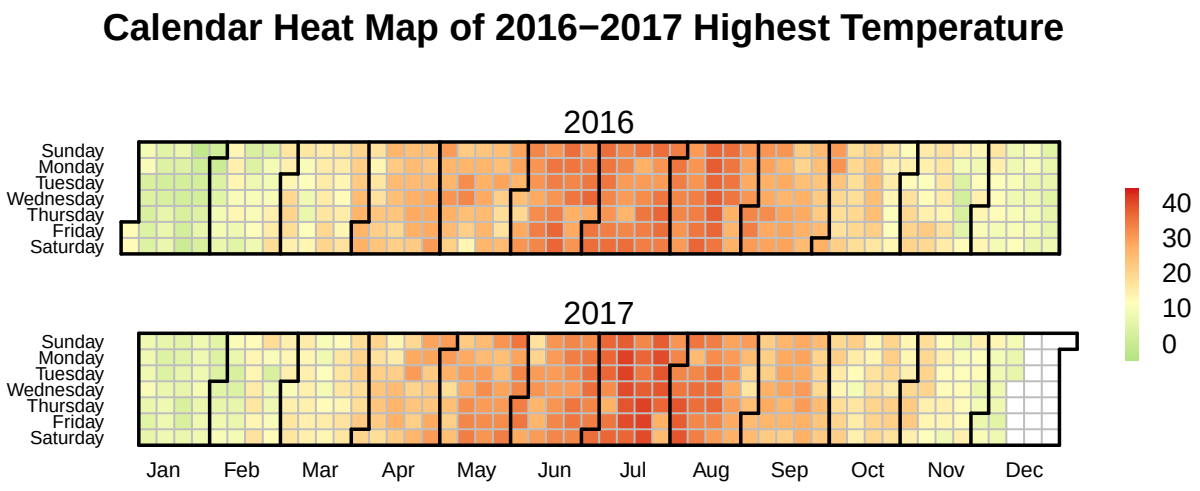



```
weather_2017 = weather_xian[weather_xian$Date >= "2017-01-01",]
weather_2017$Date <- as.Date(weather_2017$Date)

weather_2017 %>%
  ggplot() +
  geom_line(aes(x = Date,
                y = Higest_temperature,
                colour = 'blue')) +
  geom_line(aes(x = Date,
                y = Lowest_temperature,
                colour = 'red')) +
  labs(title = "Higest and Lowest Temperature Recorded in 2017, Xi'an",
       color = "Temperature" ) +
  scale_color_manual(labels = c("Highest", "Lowest"),
                     values = c("red", "blue"))
```

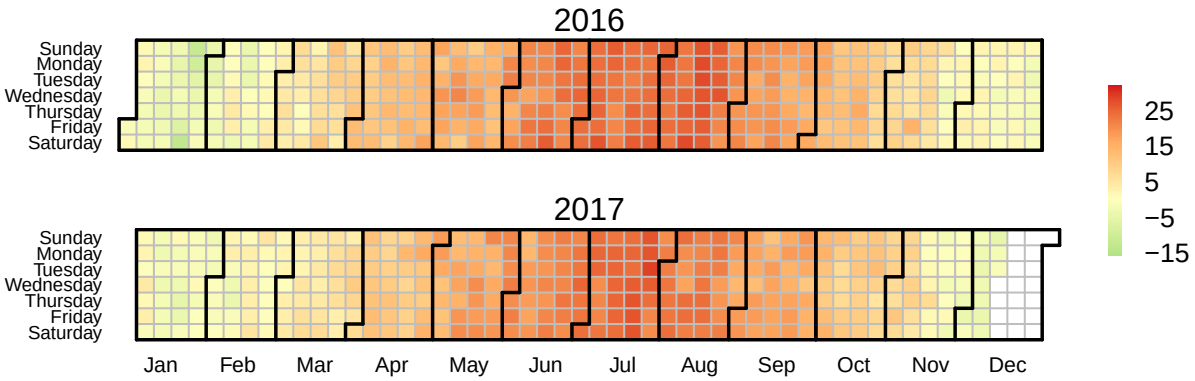



```
calendarHeat(dates = weather_xian$Date, values = weather_xian$Higest_temperature, color = 'g2r', varname
```

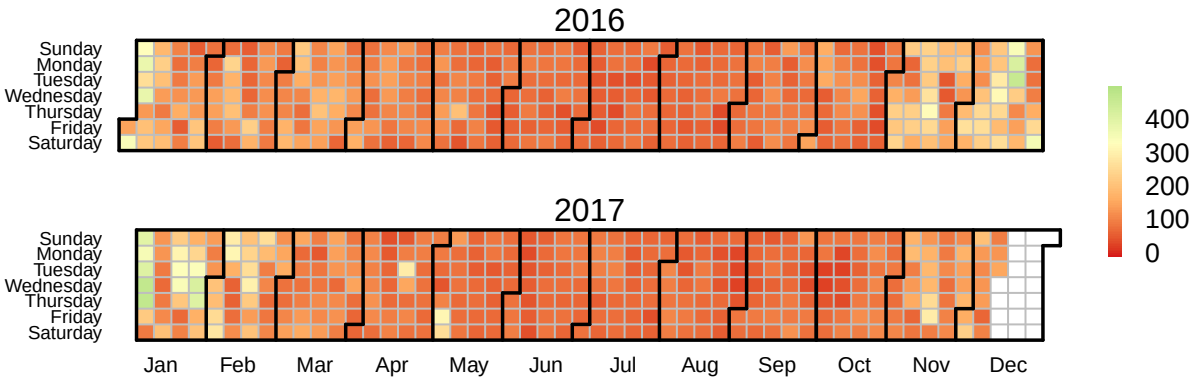



```
calendarHeat(dates = weather_xian$Date, values = weather_xian$Lowest_temperature, color = 'g2r', varname
```

```
## Warning in if (class(dates) == "character" | class(dates) == "factor") {:
## the condition has length > 1 and only the first element will be used
```




```
calendarHeat(dates = weather_xian$Date, values = weather_xian$Air_quality_index, varname = '2016-2017 A
```

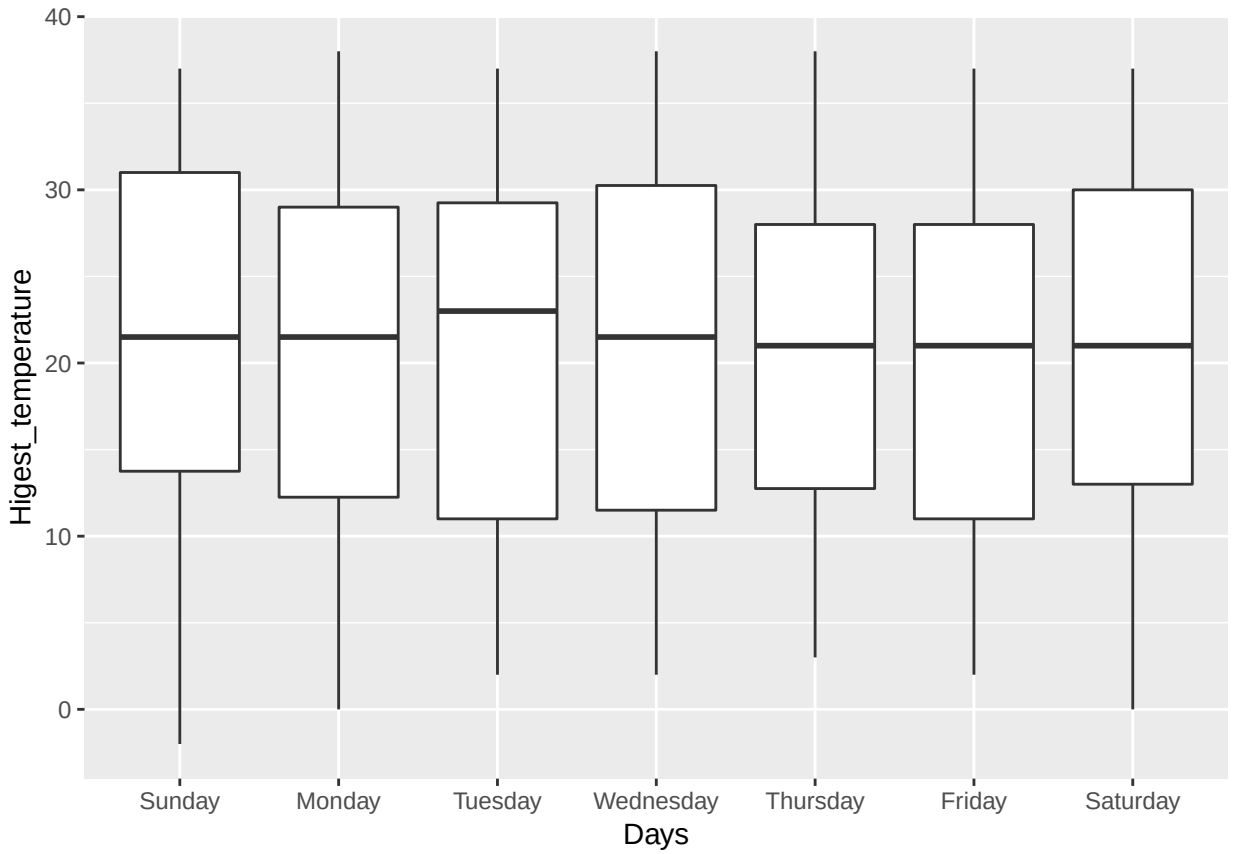


Below are the boxplot of the temperature from 2016 to 2017.

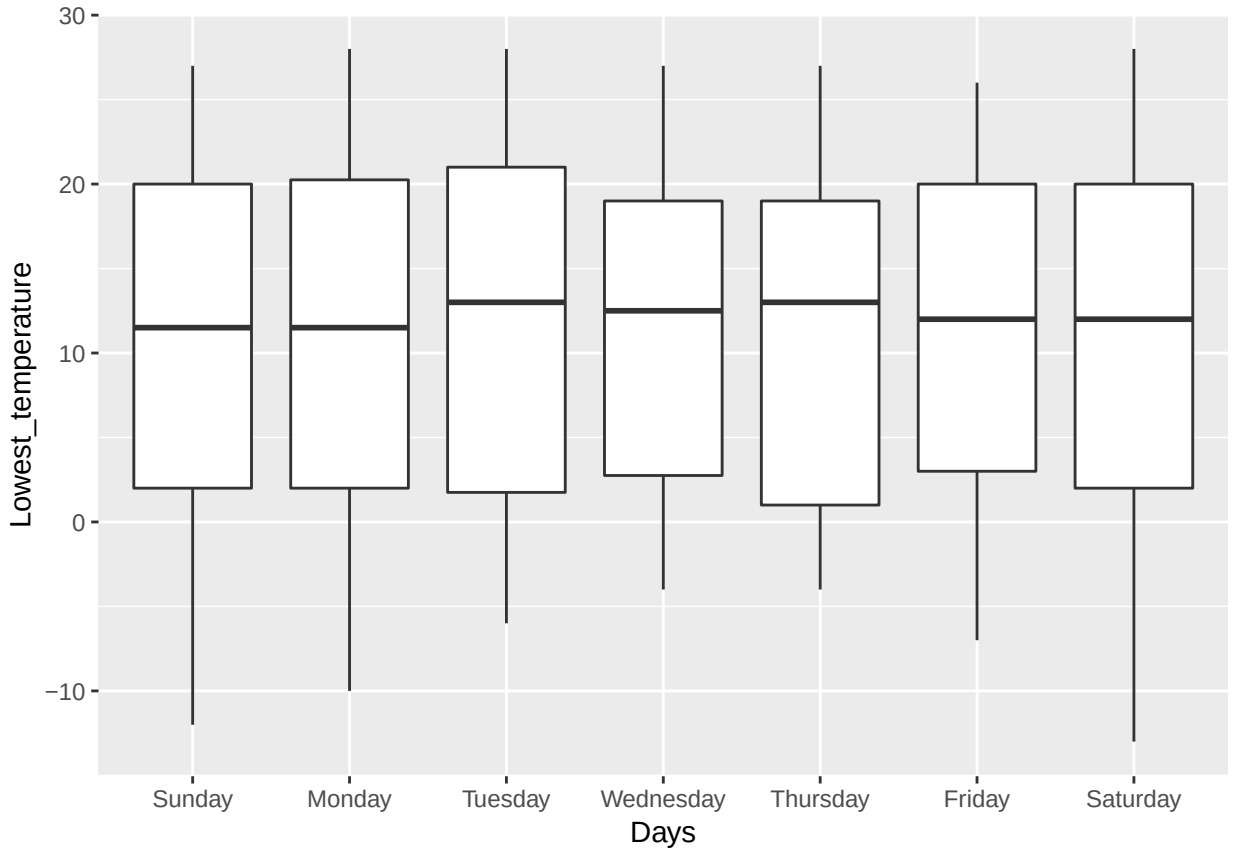
```
weather_2016$Days <- factor(weather_2016$Days, levels = c("Sunday", "Monday","Tuesday", "Wednesday", "Th
weather_2016[order(weather_2016$Days), ]
```

```
## # A tibble: 366 x 9
##    Date       Days   Higest_temperature Lowest_temperature Weather
##    <date>     <fct>               <dbl>              <dbl> <fct>
##  1 2016-12-04 Sunday                 16                  2 sunny
##  2 2016-12-11 Sunday                  8                  2 fog
##  3 2016-12-18 Sunday                 10                  2 fog
##  4 2016-12-25 Sunday                  5                  1 raining
##  5 2016-11-06 Sunday                 15                 10 raining
##  6 2016-11-13 Sunday                 17                  8 cloudy
##  7 2016-11-20 Sunday                 14                  6 fog
##  8 2016-11-27 Sunday                 13                  0 sunny
##  9 2016-10-02 Sunday                 30                 17 sunny
## 10 2016-10-09 Sunday                 18                 11 raining
## # ... with 356 more rows, and 4 more variables: Wind_direction <fct>,
## #   Wind_speed <fct>, Air_quality_index <dbl>, Air_quality <fct>
```

```
weather_2016 %>%
  ggplot() +
  geom_boxplot(aes(x = Days, y = Higest_temperature))
```



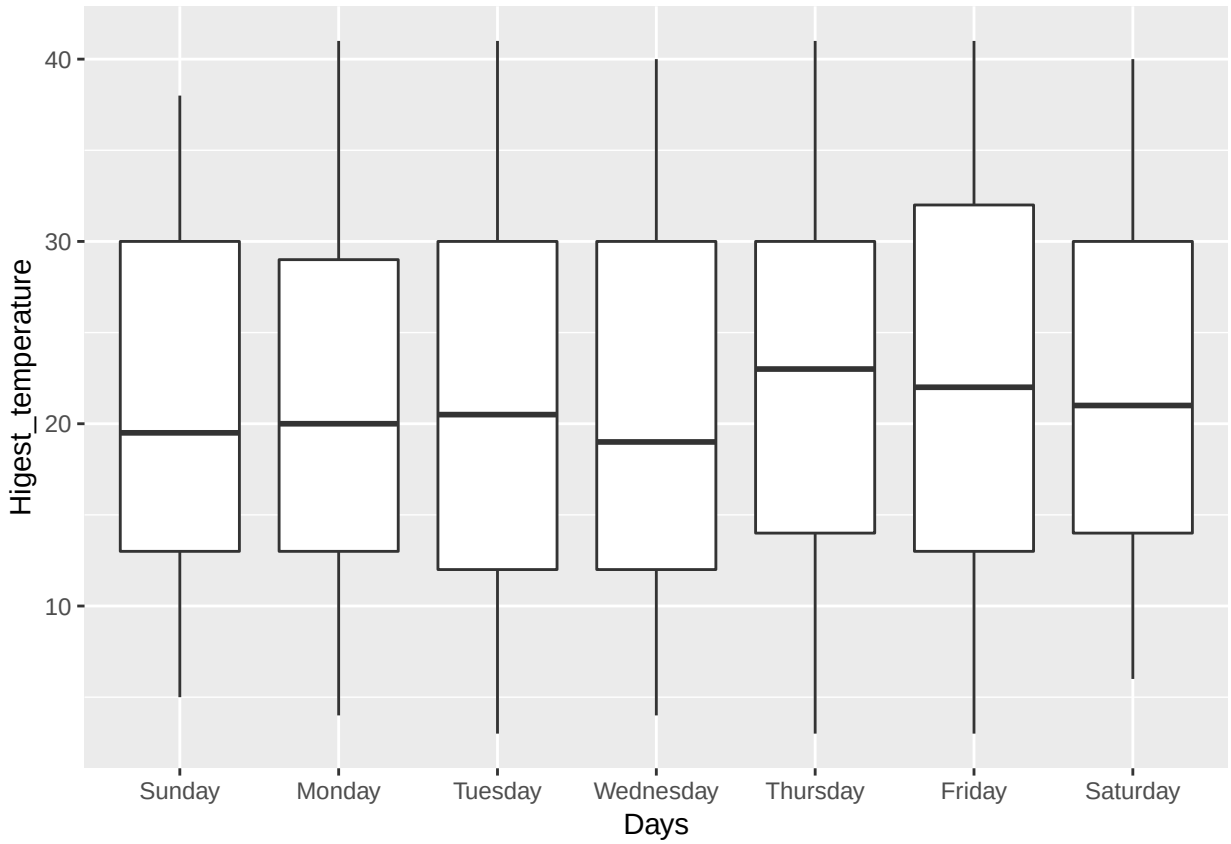

```
weather_2016 %>%
  ggplot() +
  geom_boxplot(aes(x = Days, y = Lowest_temperature))
```



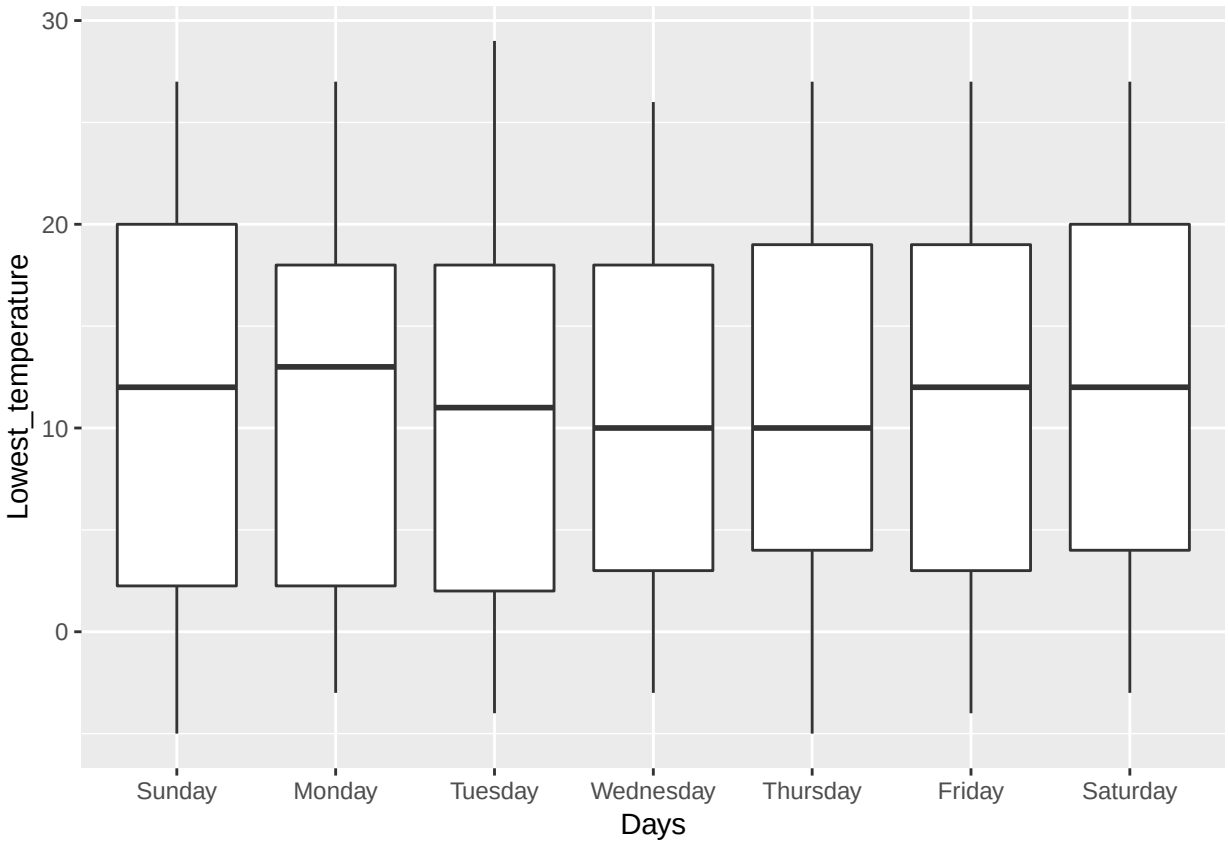

```
weather_2017$Days <- factor(weather_2017$Days, levels = c("Sunday", "Monday","Tuesday", "Wednesday", "T
weather_2017[order(weather_2017$Days), ]
```

```
## # A tibble: 346 x 9
##    Date       Days   Higest_temperature Lowest_temperature Weather
##    <date>     <fct>               <dbl>              <dbl> <fct>
##  1 2017-12-03 Sunday                 13                 -2 sunny
##  2 2017-12-10 Sunday                  9                 -5 sunny
##  3 2017-11-05 Sunday                 18                  8 sunny
##  4 2017-11-12 Sunday                 12                  1 raining
##  5 2017-11-19 Sunday                  7                 -2 raining
##  6 2017-11-26 Sunday                 14                  0 sunny
##  7 2017-10-01 Sunday                 19                 14 raining
##  8 2017-10-08 Sunday                 21                 13 raining
##  9 2017-10-15 Sunday                 13                 10 raining
## 10 2017-10-22 Sunday                 20                 11 cloudy
## # ... with 336 more rows, and 4 more variables: Wind_direction <fct>,
## #   Wind_speed <fct>, Air_quality_index <dbl>, Air_quality <fct>
```

```
weather_2017 %>%
  ggplot() +
  geom_boxplot(aes(x = Days, y = Higest_temperature))
```

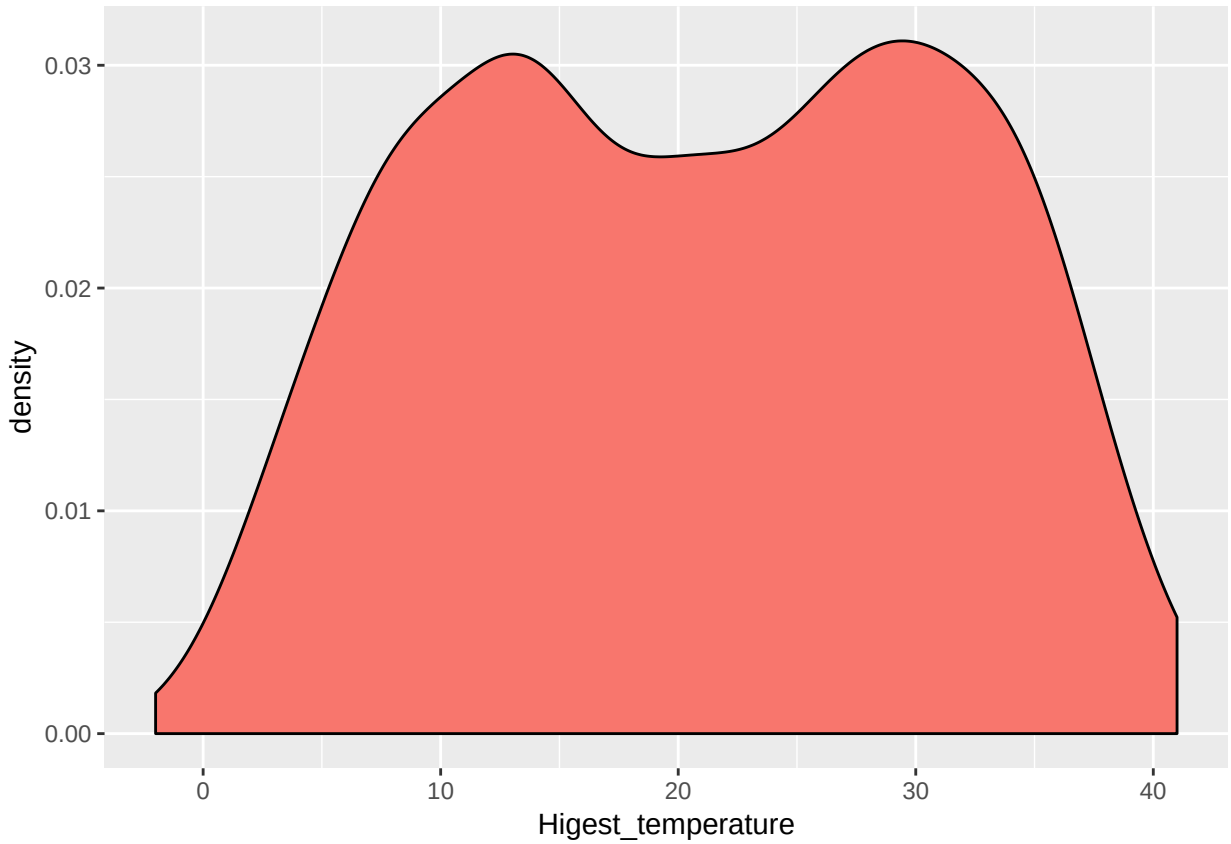



```
weather_2017 %>%
  ggplot() +
  geom_boxplot(aes(x = Days, y = Lowest_temperature))
```

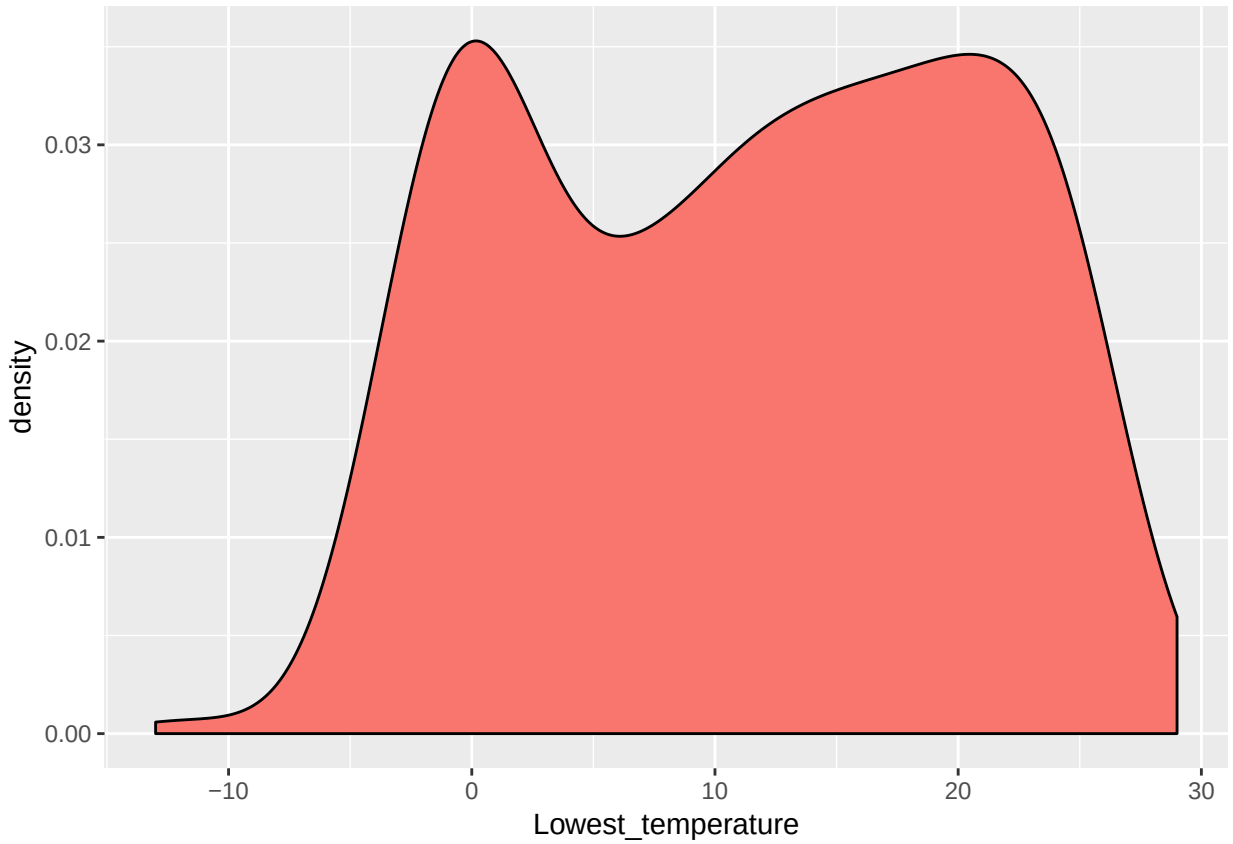


As shown in the density plot, it shows that the temperature are not normally distributed. Additionally, there is a heavy skew in the air quality index.

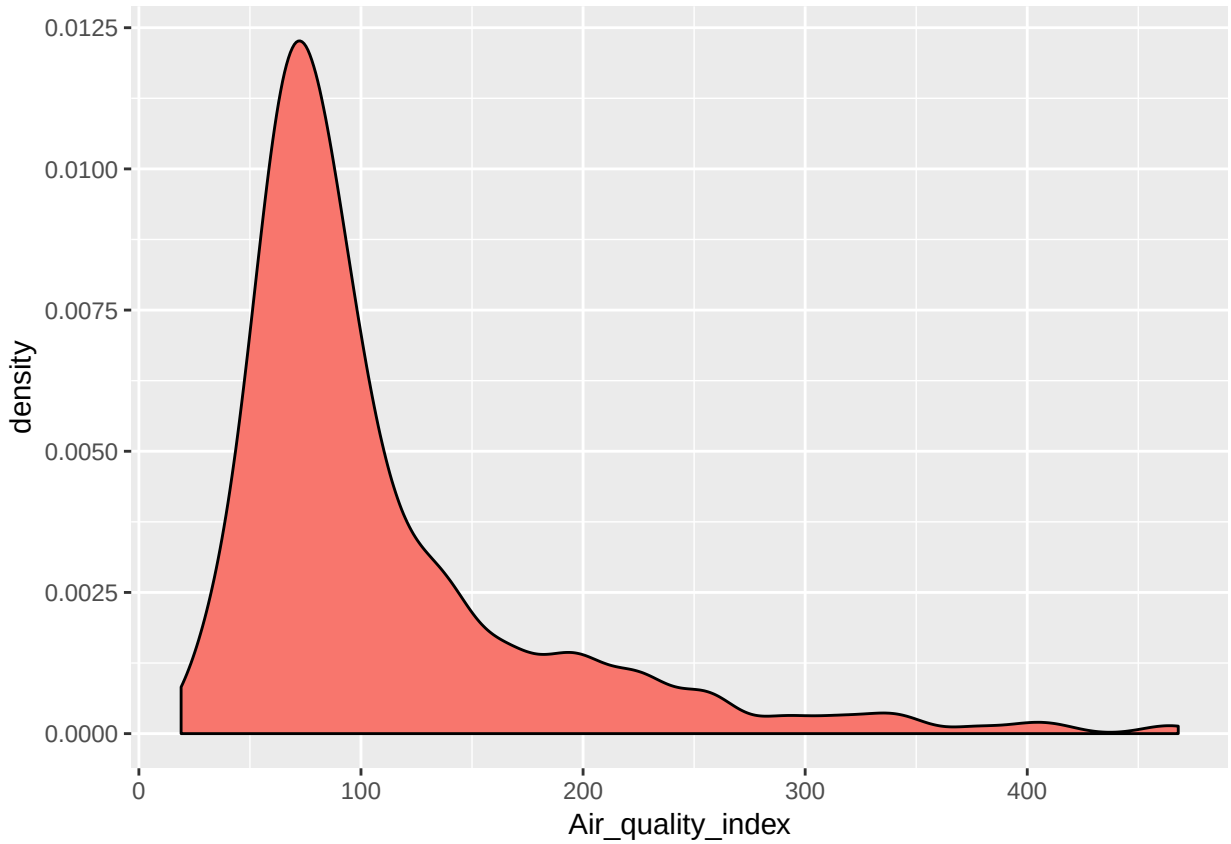
```
weather_xian %>%
  ggplot() +
  geom_density(aes(Higest_temperature, fill = 'red')) +
  theme(legend.position="none")
```




```
weather_xian %>%
  ggplot() +
  geom_density(aes(Lowest_temperature, fill = 'red')) +
  theme(legend.position="none")
```




```
weather_xian %>%
  ggplot() +
  geom_density(aes(Air_quality_index, fill = 'red')) +
  theme(legend.position="none")
```



Data Analysis

To predict the air quality index, we are using highest temperature, lowest temperature, weather of the day, wind direction and wind speed to analyze. It shows that the lowest temperature, foggy weather and raining day may contribute to the air quality index.

```
attach(weather_xian)
names(weather_xian)
```

```
## [1] "Date"                "Days"                "Higest_temperature"
## [4] "Lowest_temperature" "Weather"             "Wind_direction"
## [7] "Wind_speed"          "Air_quality_index"   "Air_quality"
```

```
model <- lm(Air_quality_index ~ Higest_temperature + Lowest_temperature + Weather + Wind_direction + Wi
summary(model)
```

```
##
## Call:
## lm(formula = Air_quality_index ~ Higest_temperature + Lowest_temperature +
##     Weather + Wind_direction + Wind_speed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -145.98  -30.20   -7.32   16.23  329.26
##
## Coefficients:
```

```
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                145.5536     9.4475  15.407  < 2e-16 ***
## Higest_temperature           0.2065     0.8343   0.247 0.804604
## Lowest_temperature          -3.5592     0.9438  -3.771 0.000176 ***
## Weatherfog                 131.1120    10.0907  12.993  < 2e-16 ***
## Weatherraining             -17.8976     6.1864  -2.893 0.003935 **
## Weathersnowing              -8.1848    32.2640  -0.254 0.799817
## Weathersunny                -6.5507     5.5338  -1.184 0.236914
## Wind_directionNorth         -3.0524    11.4780  -0.266 0.790365
## Wind_directionNorth East   -10.4255     5.3797  -1.938 0.053037 .
## Wind_directionNorth West   -65.3895    21.6973  -3.014 0.002675 **
## Wind_directionSouth        -12.1298    17.1166  -0.709 0.478773
## Wind_directionSouth East    -7.1556    17.7307  -0.404 0.686654
## Wind_directionSouth West   -18.0935     7.8126  -2.316 0.020851 *
## Wind_directionUnpredicted    1.7254    31.9301   0.054 0.956921
## Wind_directionWest         -29.8112    14.6640  -2.033 0.042437 *
## Wind_speedLight Breeze      11.1989     5.3259   2.103 0.035849 *
## Wind_speedModerate Wind     -2.1370    10.9133  -0.196 0.844811
## Wind_speedStrong Wind        6.0592    33.8670   0.179 0.858059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.36 on 694 degrees of freedom
## Multiple R-squared:  0.4365, Adjusted R-squared:  0.4227
## F-statistic: 31.62 on 17 and 694 DF,  p-value: < 2.2e-16
```

Reference

Dataset https://mp.weixin.qq.com/s?__biz=MzA5MjEyMTYwMg==&mid=2650241195&idx=1&sn=
ccfddc373857dc7b5a8a94b7847b7ba1&chksm=887227c6bf05aed0f8af502356a811f216ec04284edb96385eb299716d77a65bf98a10
scene=21#wechat_redirect