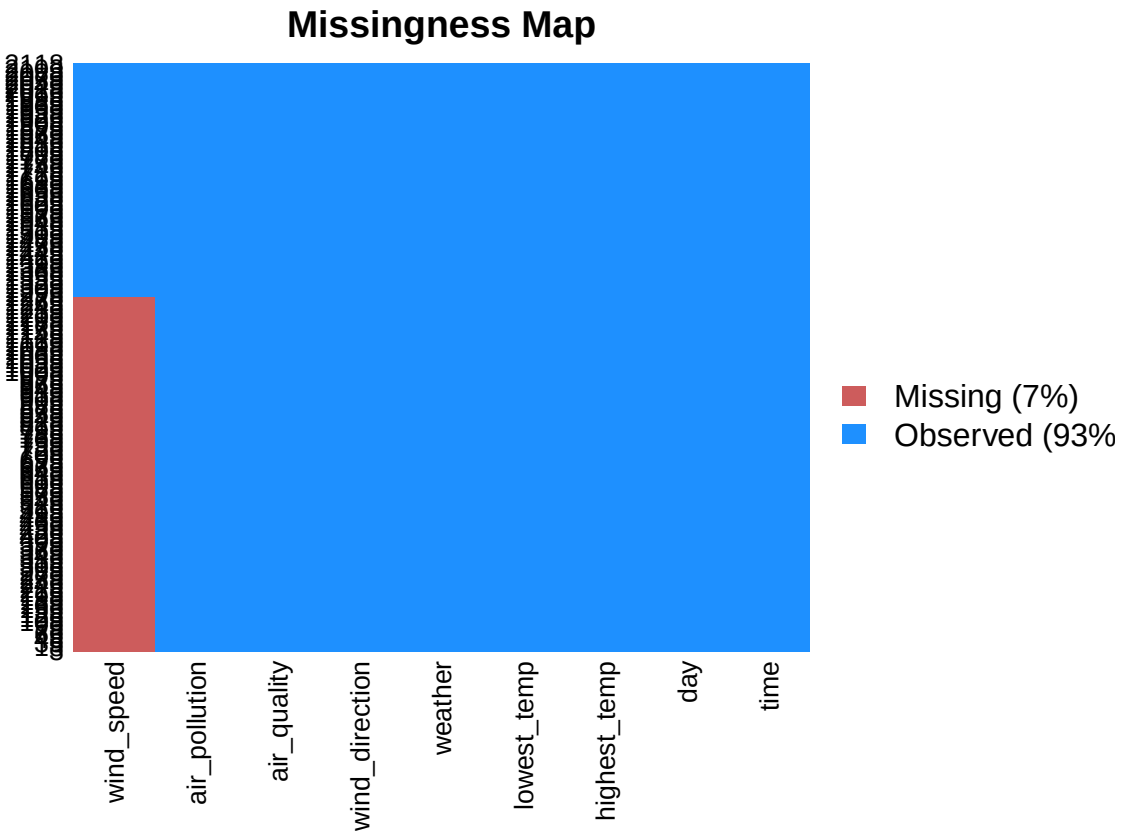
Weather_regression

Chi Ting Low

7/2/2018

```
library(readxl)
library(Amelia)
library(dplyr)
library(psych)
library(tidyverse)
library(caret)

#read data
data <- read_xlsx("weather.xlsx", na = "NA")

#rename header
colnames(data) <- c("time", "day", "highest_temp", "lowest_temp", "weather", "wind_direction", "wind_sp

#recoding header
data$day <- recode(data$day, "  " = "Sunday", "  " = "Monday", "  " = "Tuesday", "  " = "Wednesday", "  "
data$wind_speed <- recode(data$wind_speed, "1-2 " = "Light Breeze", "3-4 " = "Gentle Breeze", "4-5 " = "M
data$air_pollution <- recode(data$air_pollution, "   " = "Very Poor", "   " = "Moderate", " " = "Very Gooc
data$weather <- recode(data$weather, " ~ " = "raining", " " = "cloudy", " ~ " = "raining", " ~ " = "rai
data$wind_direction <- recode(data$wind_direction, ' ' = "East", " ~ " = "North East", " " = "North East"

#remove ℃ in data
data$highest_temp = unlist(strsplit(data$highest_temp, split = '℃', fixed = TRUE))
data$lowest_temp = unlist(strsplit(data$lowest_temp, split = '℃', fixed = TRUE))

#checking missing data
missmap(data)
```

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

## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'arguments'.

## Warning: Unknown or uninitialised column: 'imputations'.
```



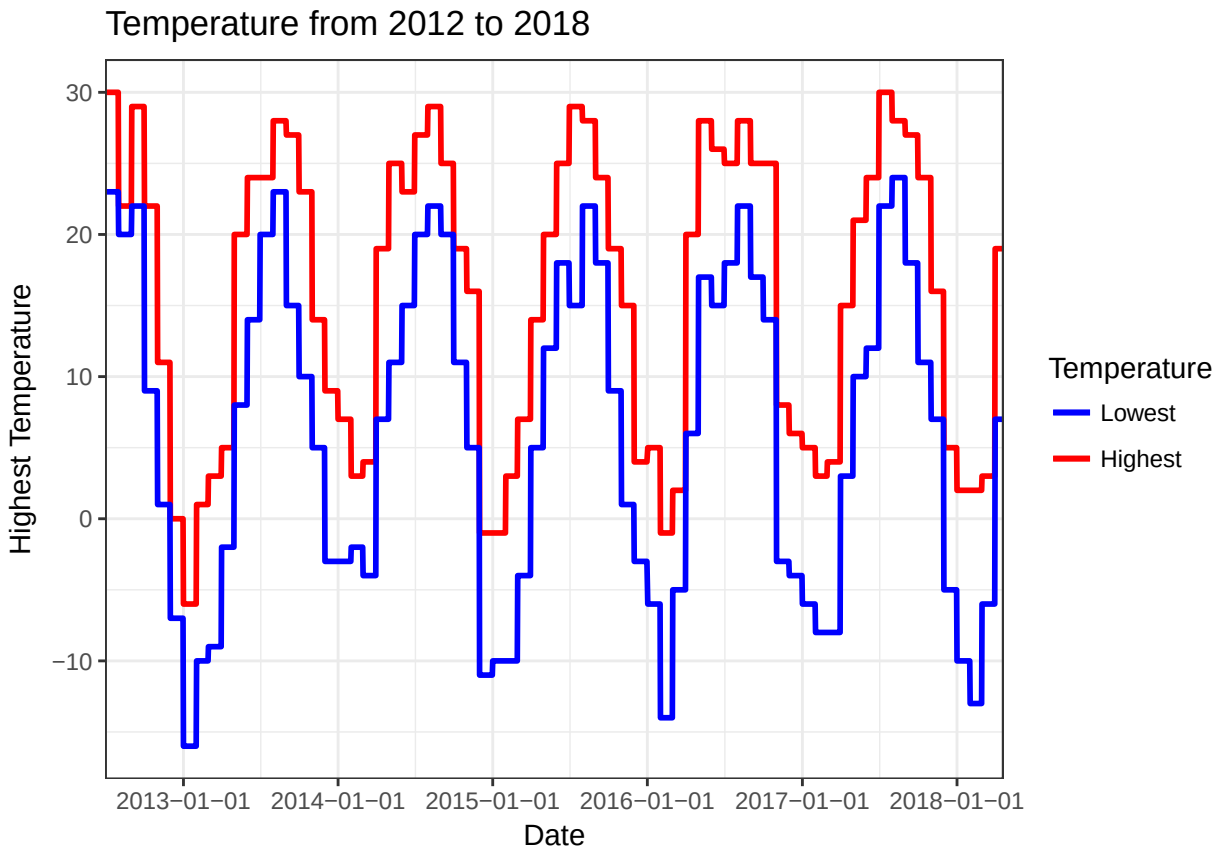

```
#recode into the relevant data type

data$day <- as.factor(data$day)
data$weather <- as.factor(data$weather)
data$wind_direction <- as.factor(data$wind_direction)
data$wind_speed <- as.factor(data$wind_speed)
data$air_pollution <- as.factor(data$air_pollution)
data$highest_temp <- as.numeric(data$highest_temp)
data$lowest_temp <- as.numeric(data$lowest_temp)

str(data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2118 obs. of  9 variables:
##  $ time          : POSIXct, format: "2018-04-01" "2018-04-02" ...
##  $ day           : Factor w/ 7 levels "Friday","Monday",..: 4 2 6 7 5 1 3 4 2 6 ...
##  $ highest_temp  : num  19 19 19 19 19 19 19 19 19 19 ...
##  $ lowest_temp   : num  7 7 7 7 7 7 7 7 7 7 ...
##  $ weather       : Factor w/ 4 levels "cloudy","fog",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ wind_direction: Factor w/ 9 levels "East","Inconsistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ wind_speed    : Factor w/ 5 levels "Breeze","Gentle Breeze",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ air_quality   : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ air_pollution : Factor w/ 6 levels "Good","Light",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
#exploratory analysis
##put into descending order
##cheking temperature across the year
data[ order(data$time , decreasing = TRUE ),]
```

```
## # A tibble: 2,118 x 9
##    time                day       highest_temp lowest_temp weather
##    <dttm>              <fct>            <dbl>       <dbl> <fct>
##  1 2018-04-20 00:00:00 Friday              19           7 sunny
##  2 2018-04-19 00:00:00 Thursday            19           7 sunny
##  3 2018-04-18 00:00:00 Wednesday           19           7 sunny
##  4 2018-04-17 00:00:00 Tuesday             19           7 sunny
##  5 2018-04-16 00:00:00 Monday              19           7 sunny
##  6 2018-04-15 00:00:00 Sunday              19           7 sunny
##  7 2018-04-14 00:00:00 Saturday            19           7 sunny
##  8 2018-04-13 00:00:00 Friday              19           7 sunny
##  9 2018-04-12 00:00:00 Thursday            19           7 sunny
## 10 2018-04-11 00:00:00 Wednesday           19           7 sunny
## # ... with 2,108 more rows, and 4 more variables: wind_direction <fct>,
## #   wind_speed <fct>, air_quality <dbl>, air_pollution <fct>
```

```
ggplot(data, aes(x = time)) +
  geom_line(aes(y = highest_temp, color = "red"),size = 1) +
  geom_line(aes(y = lowest_temp, color = "blue"), size = 1) +
  scale_x_datetime(date_labels = ("%Y-%m-%d"), expand = c(0,0), date_breaks = ("year")) +
  labs(title = "Temperature from 2012 to 2018", x = "Date", y = "Highest Temperature",  color = "Tempera
  scale_color_manual(labels = c("Lowest", "Highest"), values = c("blue", "red")) +
  theme_bw()
```

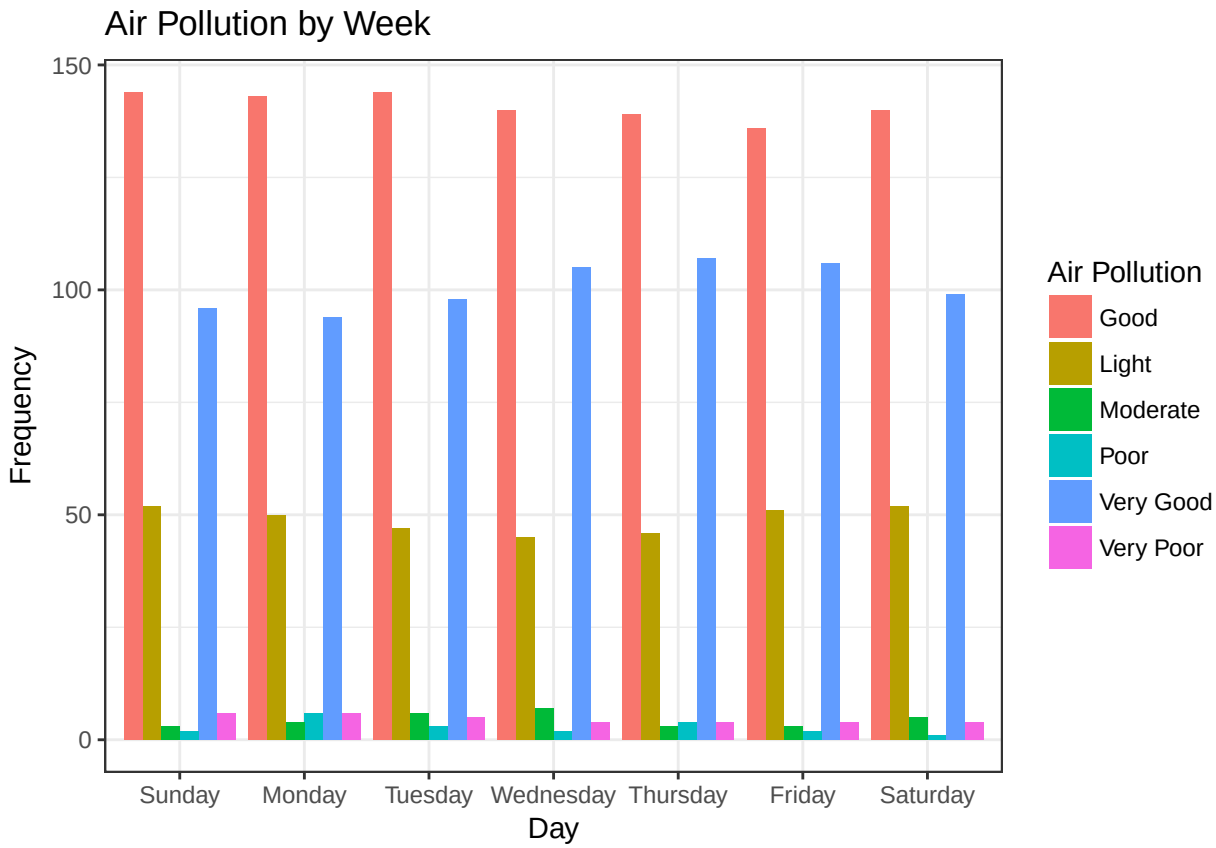



```
#checking air pollution by day
df <- table(data$day, data$air_pollution)
df <- as.data.frame(df)
colnames(df) <- c("day", "air_pollution", "Freq")
```

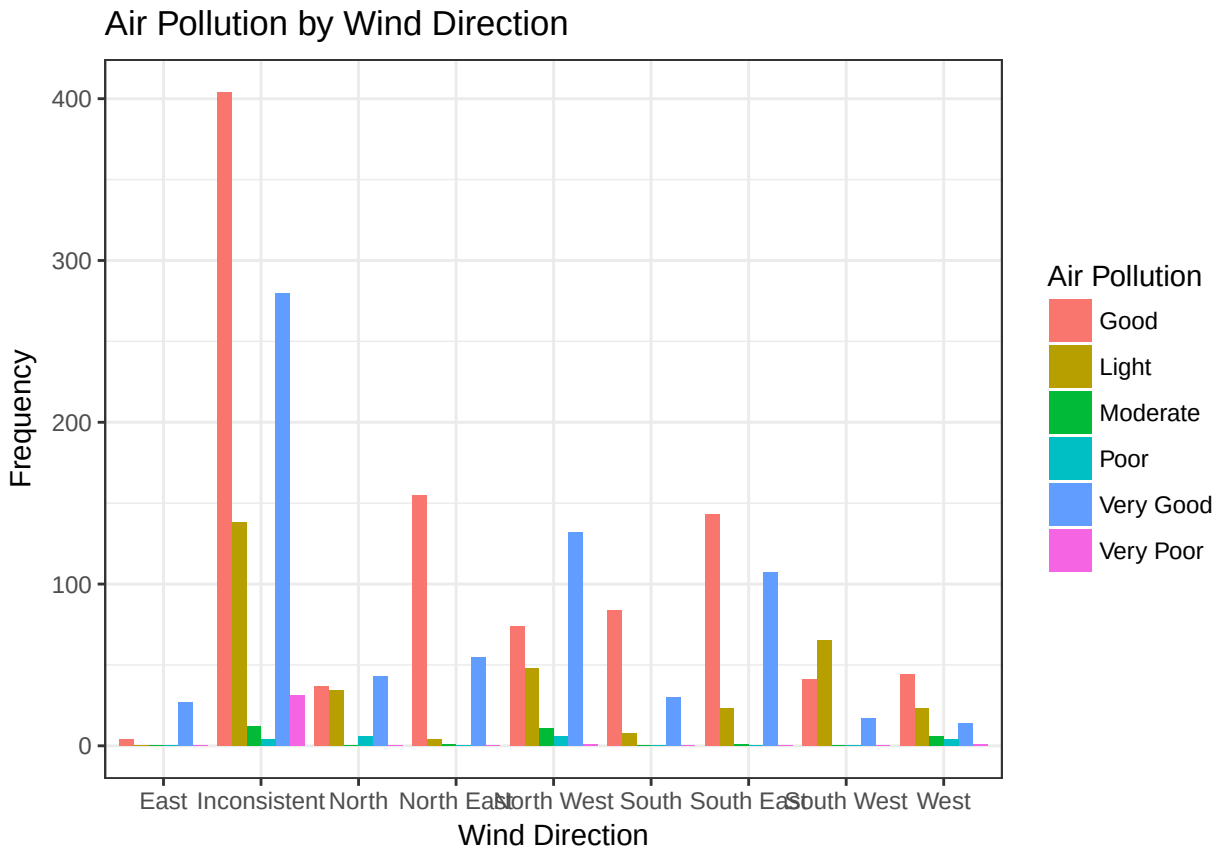
```r
df$day <- factor(df$day, levels = c("Sunday", "Monday","Tuesday", "Wednesday", "Thursday", "Friday", "Sa
df[order(df$day), ]
```

```
##          day air_pollution Freq
## 4     Sunday          Good  144
## 11    Sunday         Light   52
## 18    Sunday      Moderate    3
## 25    Sunday          Poor    2
## 32    Sunday     Very Good   96
## 39    Sunday     Very Poor    6
## 2     Monday          Good  143
## 9     Monday         Light   50
## 16    Monday      Moderate    4
## 23    Monday          Poor    6
## 30    Monday     Very Good   94
## 37    Monday     Very Poor    6
## 6    Tuesday          Good  144
## 13   Tuesday         Light   47
## 20   Tuesday      Moderate    6
## 27   Tuesday          Poor    3
## 34   Tuesday     Very Good   98
## 41   Tuesday     Very Poor    5
## 7  Wednesday          Good  140
## 14 Wednesday         Light   45
## 21 Wednesday      Moderate    7
## 28 Wednesday          Poor    2
## 35 Wednesday     Very Good  105
## 42 Wednesday     Very Poor    4
## 5   Thursday          Good  139
## 12  Thursday         Light   46
## 19  Thursday      Moderate    3
## 26  Thursday          Poor    4
## 33  Thursday     Very Good  107
## 40  Thursday     Very Poor    4
## 1     Friday          Good  136
## 8     Friday         Light   51
## 15    Friday      Moderate    3
## 22    Friday          Poor    2
## 29    Friday     Very Good  106
## 36    Friday     Very Poor    4
## 3   Saturday          Good  140
## 10  Saturday         Light   52
## 17  Saturday      Moderate    5
## 24  Saturday          Poor    1
## 31  Saturday     Very Good   99
## 38  Saturday     Very Poor    4
```

```r
ggplot(df, aes(x = day, y = Freq, fill = air_pollution)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Air Pollution by Week", x = "Day", y = "Frequency") +
  guides(fill = guide_legend(title = "Air Pollution")) +
  theme_bw()
```




```r
#checking wind direction and air pollution
df1 <- table(data$wind_direction, data$air_pollution)
df1 <- as.data.frame(df1)
colnames(df1) <- c("wind_direction", "air_pollution", "Freq")

ggplot(df1, aes(x = wind_direction, y = Freq, fill = air_pollution)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Air Pollution by Wind Direction", x = "Wind Direction", y = "Frequency") +
  guides(fill = guide_legend(title = "Air Pollution")) +
  theme_bw()
```




```
set.seed(7)
# prepare training scheme
#remove missing value and time
names(data)
```

```
## [1] "time"           "day"            "highest_temp"   "lowest_temp"
## [5] "weather"        "wind_direction" "wind_speed"     "air_quality"
## [9] "air_pollution"
```

```
data.na <- data[,c(-1,-2,-7)]

#making regression model
summary(lm(air_quality ~ ., data = data.na))
```

```
##
## Call:
## lm(formula = air_quality ~ ., data = data.na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.011  -8.166   0.461   7.394  54.037
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   67.6027     2.4124  28.023  < 2e-16 ***
## highest_temp                   0.8345     0.1266   6.590 5.55e-11 ***
## lowest_temp                   -0.6122     0.1247  -4.908 9.92e-07 ***
## weatherfog                    10.8395     1.5948   6.797 1.39e-11 ***
```

```
## weatherraining               0.2849     0.9120   0.312   0.7547
## weathersunny                -0.7393     0.6662  -1.110   0.2672
## wind_directionInconsistent  -3.9981     2.3670  -1.689   0.0914 .
## wind_directionNorth         -4.8414     2.5528  -1.896   0.0580 .
## wind_directionNorth East    -9.6480     2.4730  -3.901 9.87e-05 ***
## wind_directionNorth West    -2.2824     2.3728  -0.962   0.3362
## wind_directionSouth        -11.5778     2.4538  -4.718 2.54e-06 ***
## wind_directionSouth East    -0.6261     2.3106  -0.271   0.7864
## wind_directionSouth West   -13.3301     2.5730  -5.181 2.42e-07 ***
## wind_directionWest           3.5661     2.5721   1.386   0.1658
## air_pollutionLight          42.4944     0.7894  53.829  < 2e-16 ***
## air_pollutionModerate       99.3532     2.2037  45.085  < 2e-16 ***
## air_pollutionPoor          168.4139     2.7025  62.319  < 2e-16 ***
## air_pollutionVery Good     -32.1513     0.6108 -52.635  < 2e-16 ***
## air_pollutionVery Poor     294.2936     2.4736 118.974  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.65 on 2099 degrees of freedom
## Multiple R-squared:  0.9483, Adjusted R-squared:  0.9478
## F-statistic:  2138 on 18 and 2099 DF,  p-value: < 2.2e-16
```

```r
anova(lm(air_quality ~ ., data = data.na))
```

```
## Analysis of Variance Table
##
## Response: air_quality
##                  Df  Sum Sq Mean Sq  F value    Pr(>F)
## highest_temp      1   39146   39146  288.226 < 2.2e-16 ***
## lowest_temp       1    4639    4639   34.155 5.886e-09 ***
## weather           3  815378  271793 2001.165 < 2.2e-16 ***
## wind_direction    8  180166   22521  165.817 < 2.2e-16 ***
## air_pollution     5 4186714  837343 6165.220 < 2.2e-16 ***
## Residuals      2099  285080     136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```