Data Cleaning, Preprocessing & Visualization - R Assignment

Group 9

Nurul Alam - 101443564
Ananth Satish Embrandiri - 101416006
Denny Paul - 101411032
Anuj Yadav - 101419066
Abhishek Singh Bisht - 101420857
Saptarshi Kundu - 101365961

2023-02-11

Delhi Air Quality Index

About Dataset

This dataset contains air quality data from the national capital of Delhi, India. It includes information on air pollution levels, including particulate matter (PM2.5 and PM10) levels, nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon dioxide (CO2), ozone (O3), and other pollutants. The data was collected from monitoring stations located in various areas of Delhi between November 25, 2020, and January 24, 2023. This dataset is a valuable resource for researchers and policymakers to better understand air quality in Delhi and its impacts on public health.

Import Dataset

```
aqi = read.csv("delhi_aqi.csv")
```

Q1 : Print the structure of the dataset

```
str(aqi)
```

```
## 'data.frame':    18776 obs. of  9 variables:
##  $ date : chr  "2020-11-25 01:00:00" "2020-11-25 02:00:00" "2020-11-25 03:00:00" "2020-11-25 04:00:00
##  $ co   : num  2617 3632 4539 4539 4379 ...
##  $ no   : num  2.18 23.25 52.75 50.96 42.92 ...
##  $ no2  : num  70.6 89.1 100.1 111 117.9 ...
##  $ o3   : num  13.59 0.33 1.11 6.44 17.17 ...
##  $ so2  : num  38.6 54.4 68.7 78.2 87.7 ...
##  $ pm2_5: num  365 421 464 455 448 ...
##  $ pm10 : num  412 486 542 534 529 ...
##  $ nh3  : num  28.6 41 49.1 48.1 46.6 ...
```

Q2 : List the variables in your dataset

```
colnames(aqi)
```

```
## [1] "date"  "co"    "no"    "no2"   "o3"    "so2"   "pm2_5" "pm10"  "nh3"
```

Q3 : Print the top 15 rows of your dataset

```r
head(aqi, n=15)
```

```
##                    date      co    no    no2     o3    so2  pm2_5   pm10   nh3
## 1  2020-11-25 01:00:00 2616.88  2.18  70.60  13.59  38.62 364.61 411.73 28.63
## 2  2020-11-25 02:00:00 3631.59 23.25  89.11   0.33  54.36 420.96 486.21 41.04
## 3  2020-11-25 03:00:00 4539.49 52.75 100.08   1.11  68.67 463.68 541.95 49.14
## 4  2020-11-25 04:00:00 4539.49 50.96 111.04   6.44  78.20 454.81 534.00 48.13
## 5  2020-11-25 05:00:00 4379.27 42.92 117.90  17.17  87.74 448.14 529.19 46.61
## 6  2020-11-25 06:00:00 3898.62 28.39 117.90  40.05 101.09 437.25 511.79 42.05
## 7  2020-11-25 07:00:00 1949.31 14.53 105.56  83.69 185.01 312.76 349.20 12.79
## 8  2020-11-25 08:00:00 1508.71 11.62 112.41  87.98 217.44 275.53 303.47  6.59
## 9  2020-11-25 09:00:00 1361.85  7.04 109.67  95.84 213.62 263.51 289.86  6.02
## 10 2020-11-25 10:00:00 1602.17  3.10  93.22 104.43 152.59 271.25 302.27 12.16
## 11 2020-11-25 11:00:00 2136.23  1.27  94.59  86.55 103.95 284.51 324.34 21.28
## 12 2020-11-25 12:00:00 2590.18  0.19 109.67  50.78  82.02 287.83 336.00 27.87
## 13 2020-11-25 13:00:00 3017.43  0.60 120.64  19.67  69.62 295.37 354.19 35.47
## 14 2020-11-25 14:00:00 3471.37  6.65 117.90   3.00  65.80 325.89 402.37 46.10
## 15 2020-11-25 15:00:00 3898.62 16.09 105.56   0.25  63.90 363.16 456.38 50.66
```

Q4 : Write a user defined function using any of the variables from the dataset

```r
library(purrr)
```

```r
get_floor <- function(x) {
  x = floor(x)
}

aqi$pm_2_5_floor <- map_dbl(aqi$pm2_5, get_floor)
head(aqi, n=5)
```

```
##                   date      co    no    no2    o3   so2  pm2_5   pm10   nh3
## 1 2020-11-25 01:00:00 2616.88  2.18  70.60 13.59 38.62 364.61 411.73 28.63
## 2 2020-11-25 02:00:00 3631.59 23.25  89.11  0.33 54.36 420.96 486.21 41.04
## 3 2020-11-25 03:00:00 4539.49 52.75 100.08  1.11 68.67 463.68 541.95 49.14
## 4 2020-11-25 04:00:00 4539.49 50.96 111.04  6.44 78.20 454.81 534.00 48.13
## 5 2020-11-25 05:00:00 4379.27 42.92 117.90 17.17 87.74 448.14 529.19 46.61
##   pm_2_5_floor
## 1          364
## 2          420
## 3          463
## 4          454
## 5          448
```

Q5 : Use data manipulation techniques and filter rows based on any logical criteria that exist in your dataset

```
library(dplyr)
```

```
aqi_no_filter = aqi %>% filter(no > 0.0)
dim(aqi_no_filter)
```

```
## [1] 16470    10
```

Q6 : Identify the dependent & independent variables and use reshaping techniques and create a new data frame by joining those variables from your dataset

```
subframe1 = aqi[, c(1, 2, 8)]
subframe1$total_co_pm10 = subframe1$co + subframe1$pm10
subframe1 = subframe1[, -c(2, 3)]
aqi = merge(aqi, subframe1, by="date")
```

Q7 : Remove missing values in your dataset

```
dim(aqi)
```

```
## [1] 18776    11
```

```
colSums(is.na(aqi))
```

```
##          date            co            no           no2            o3 
##             0             0             0             0             0 
##           so2         pm2_5          pm10           nh3  pm_2_5_floor 
##             0             0             0             0             0 
## total_co_pm10 
##             0
```

```
aqi = na.omit(aqi)
dim(aqi)
```

```
## [1] 18776    11
```

Q8 : Identify and remove duplicated data in your dataset

```
dim(aqi)
```

```
## [1] 18776    11
```

```
sum(duplicated(aqi))
```

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

```
aqi = unique(aqi)
dim(aqi)
```

```
## [1] 18776    11
```

Q9 : Reorder multiple rows in descending order

```
aqi = aqi %>% arrange(desc(date))
```

Q10 : Rename some of the column names in your dataset

```
colnames(aqi)
```

```
##  [1] "date"          "co"            "no"            "no2"
##  [5] "o3"            "so2"           "pm2_5"         "pm10"
##  [9] "nh3"           "pm_2_5_floor"  "total_co_pm10"
```

```
names(aqi)[names(aqi) == "date"] = "Date"
names(aqi)[names(aqi) == "co"] = "CO"
names(aqi)[names(aqi) == "no"] = "NO"
colnames(aqi)
```

```
##  [1] "Date"          "CO"            "NO"            "no2"
##  [5] "o3"            "so2"           "pm2_5"         "pm10"
##  [9] "nh3"           "pm_2_5_floor"  "total_co_pm10"
```

Q11 : Add new variables in your data frame by using a mathematical function (for e.g. – multiply an existing column by 2 and add it as a new variable to your data frame)

```
aqi$total_pm = aqi$pm2_5 + aqi$pm10
aqi = aqi %>% mutate(pm2_5_10 = pm2_5 * 10)
```

Q12 : Create a training set using random number generator engine.

```
dim(aqi)
```

```
## [1] 18776    13
```

```
set.seed(42)
aqi_train = aqi %>% sample_frac(0.8, replace = FALSE)
dim(aqi_train)
```

```
## [1] 15021    13
```

Q13 : Print the summary statistics of your dataset

```
summary(aqi)
```

```
##      Date                 CO                NO              no2        
##   Length:18776       Min.   :  260.4   Min.   :  0.00   Min.   :  4.28  
##   Class :character   1st Qu.: 1068.1   1st Qu.:  0.68   1st Qu.: 33.93  
##   Mode  :character   Median : 1842.5   Median :  5.25   Median : 54.15  
##                      Mean   : 2929.2   Mean   : 33.66   Mean   : 66.22  
##                      3rd Qu.: 3685.0   3rd Qu.: 35.76   3rd Qu.: 83.63  
##                      Max.   :21148.7   Max.   :500.68   Max.   :460.62  
##        o3             so2             pm2_5             pm10        
##   Min.   :  0.00   Min.   :  5.25   Min.   :  11.83   Min.   :  15.07  
##   1st Qu.:  0.34   1st Qu.: 34.81   1st Qu.:  84.44   1st Qu.: 118.80  
##   Median : 27.18   Median : 52.93   Median : 157.44   Median : 209.71  
##   Mean   : 60.35   Mean   : 66.69   Mean   : 238.13   Mean   : 300.09  
##   3rd Qu.: 92.98   3rd Qu.: 82.02   3rd Qu.: 313.00   3rd Qu.: 387.96  
##   Max.   :801.09   Max.   :579.83   Max.   :1708.09   Max.   :1969.93  
##       nh3          pm_2_5_floor    total_co_pm10        total_pm      
##   Min.   :  0.00   Min.   :  11.0   Min.   :  352.1   Min.   :  27.66  
##   1st Qu.:  9.63   1st Qu.:  84.0   1st Qu.: 1187.5   1st Qu.: 207.12  
##   Median : 17.48   Median : 157.0   Median : 2035.0   Median : 365.03  
##   Mean   : 25.11   Mean   : 237.6   Mean   : 3229.3   Mean   : 538.22  
##   3rd Qu.: 30.40   3rd Qu.: 313.0   3rd Qu.: 4076.1   3rd Qu.: 697.48  
##   Max.   :287.77   Max.   :1708.0   Max.   :22985.2   Max.   :3678.02  
##     pm2_5_10      
##   Min.   :  118.3  
##   1st Qu.:  844.4  
##   Median : 1574.5  
##   Mean   : 2381.3  
##   3rd Qu.: 3130.0  
##   Max.   :17080.9  
```

Q14 : Use any of the numerical variables from the dataset and perform the following statistical functions

- **Mean**
- **Median**
- **Mode**
- **Range**

```
mean(aqi$o3)
```

```
## [1] 60.34624
```

```
median(aqi$o3)
```

```
## [1] 27.18
```

```
#Calculating mode
counts = table(aqi$o3)
max_count <- max(counts)
mode_indices <- which(counts == max_count)
mode_values <- names(counts)[mode_indices]
mode_values <- as.numeric(mode_values)
print(mode_values)
```

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

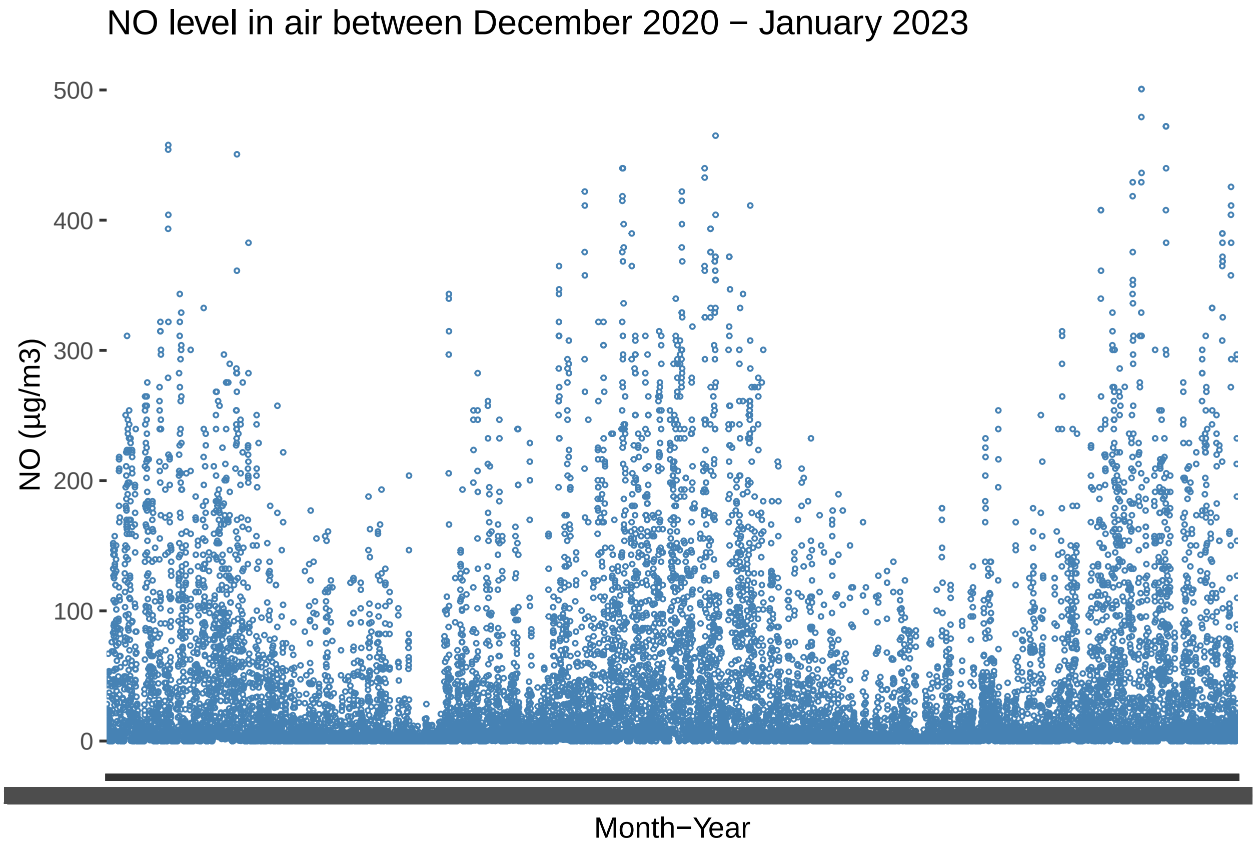
```
range(aqi$o3)
```

```
## [1]   0.00 801.09
```

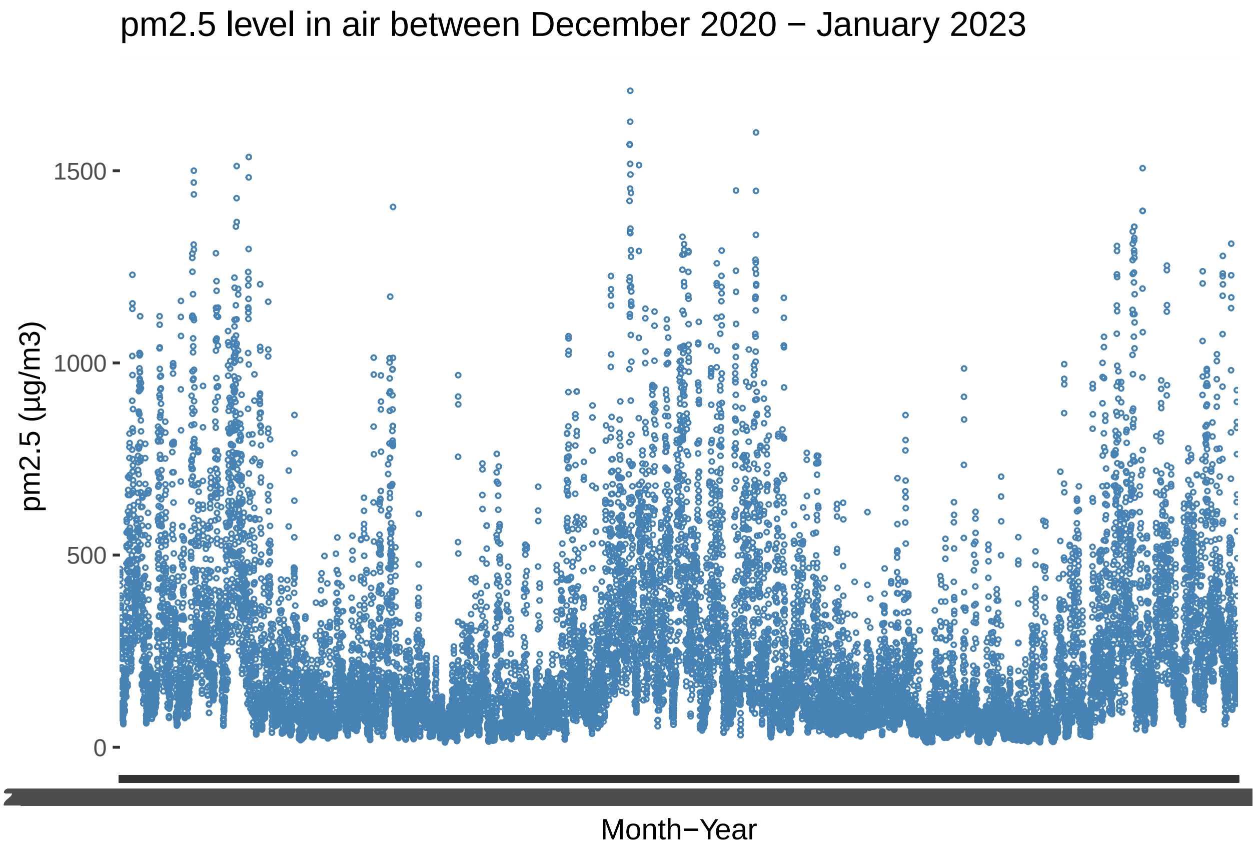
Q15 : Plot a scatter plot for any 2 variables in your dataset

```
library(ggplot2)
```

```
ggplot(data = aqi,aes(y = NO, x = Date))+geom_point(stat='identity', size = 0.5,
        color = "steelblue", shape=21)+labs(x = "Month-Year", y = "NO (µg/m3)",
        title = "NO level in air between December 2020 - January 2023")
```

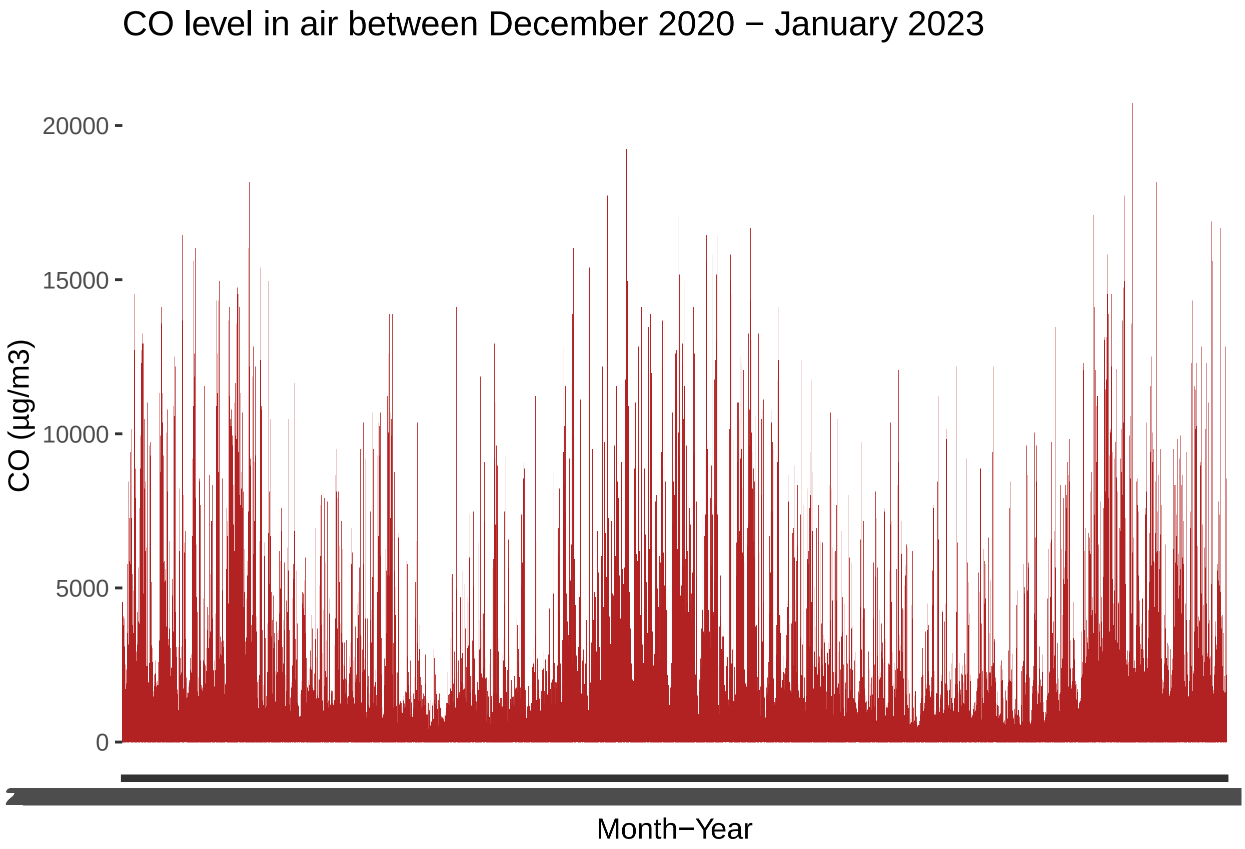


```
ggplot(data = aqi,aes(y = pm2_5, x = Date))+geom_point(stat='identity',
        size = 0.5, color = "steelblue", shape=21)+labs(x = "Month-Year",
        y = "pm2.5 (µg/m3)",
        title = "pm2.5 level in air between December 2020 - January 2023")
```

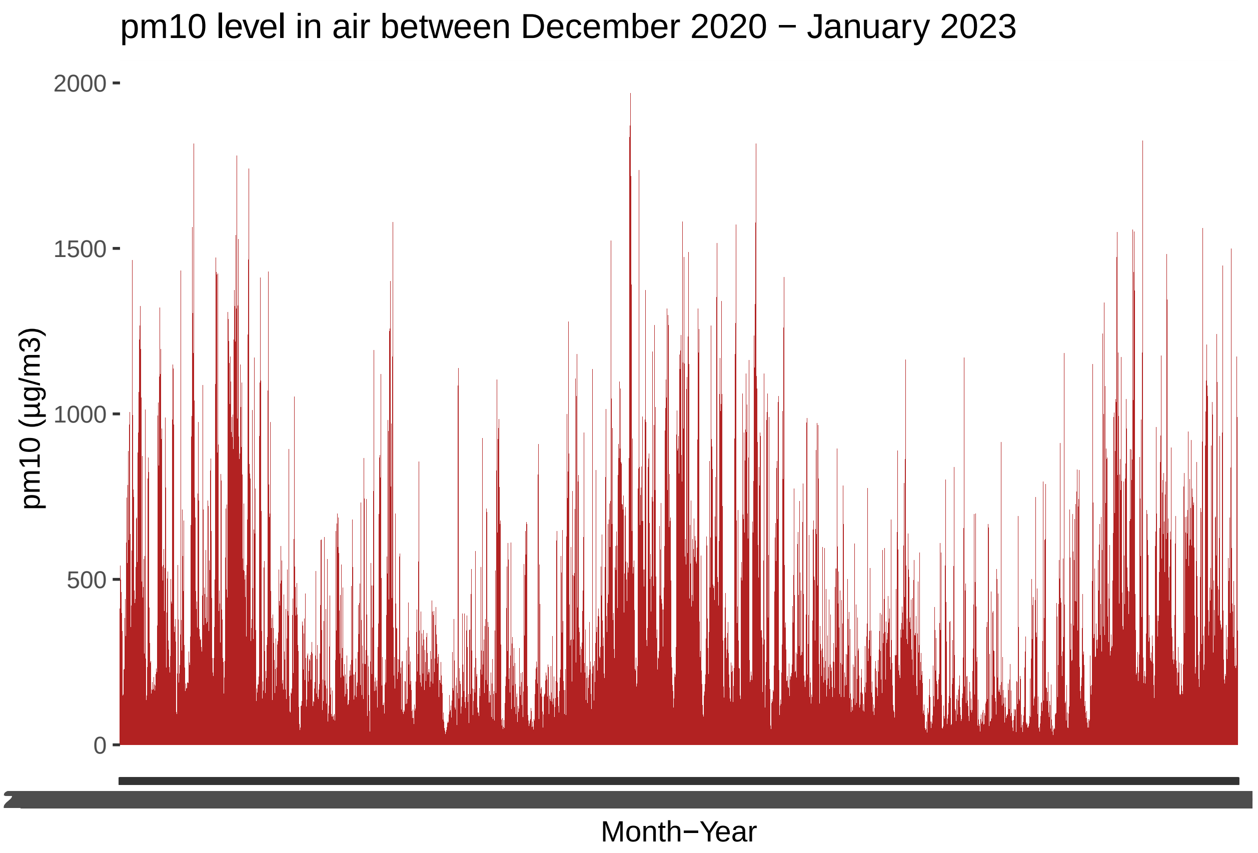


Q16 : Plot a bar plot for any 2 variables in your dataset

```
ggplot(data = aqi,aes(y = CO, x = Date))+geom_bar(stat='identity',
        fill = "firebrick")+labs(x = "Month-Year", y = "CO (µg/m3)",
        title = "CO level in air between December 2020 - January 2023")
```



```
ggplot(data = aqi,aes(y = pm10, x = Date))+geom_bar(stat='identity',
        fill = "firebrick")+labs(x = "Month-Year", y = "pm10 (µg/m3)",
        title = "pm10 level in air between December 2020 - January 2023")
```



Q17 : Find the correlation between any 2 variables by applying least square linear regression model

```
Y = aqi[, "NO"]
X = aqi[, "CO"]
co_no_corr = cor(Y,X, method="pearson")
co_no_corr
```

```
## [1] 0.9141286
```