R-PRESENTATION

TOPIC - CHAPTER-3

STARTING OUT : WORKING WITH OBJECTS

GROUP 3

JASLEEN KAUR	**2018PSC1059**
PREETI PALAK KAUR	**2018PSC1060**
KESHAV MADHAV TRIVEDI	**2018PSC1022**
SAKSHAM KUMAR	**2018PSC1024**
ANANT JAWLA	**2018PSC1102**
HRISHABH SINGH	**2018PSC1029**
MD. DANISH KHAN	**2018PSC1017**
ARPIT AGARWAL	**2018PSC1062**
KARTAR KAUR	**2018PSC1057**
ANKIT SINGH	**2018PSC1068**
HAMAAD MASROOR	**2018PSC1111**

What in this chapter??

- Different forms of DATA objects : Vectors, Data Frames, Matrices
- Data manipulation.
- Selecting, Sorting, Rearranging and Displaying data object.
- Constructing data objects and converting it from one form to another.
- Some basics function of *DPLYR* package.

VECTORS

These are one dimensional objects

consider an example: Student Marks

```
s_marks <- c(87,96,44,80,68,90,55,NA,99,89,64,70,68,81,66,83,58,90,85,NA)
s_marks
```

```
##  [1] 87 96 44 80 68 90 55 NA 99 89 64 70 68 81 66 83 58 90 85 NA
```

```
length(s_marks)
```

```
#to check the length of the vector i.e, total number of elements in the
 data
```

```
## [1] 20
```

```
s_marks[5]
```

```
 #to display any particular element
```

```
## [1] 68
```

```
s_marks[3:13]
```

```
#to display the subset on vector by giving indexes you want to display
```

```
##  [1] 44 80 68 90 55 NA 99 89 64 70 68
```

```
#we can also perform this on some logic functions by giving the appropr
iate logical operators like >, <, == ,etc.
s_marks[s_marks>75]
```

```
##  [1] 87 96 80 90 NA 99 89 81 83 90 85 NA
```

```
s_marks[(length(s_marks)-10):length(s_marks)]
```

```
 #last ten items of the vector will be displayed
```

```
##  [1] 89 64 70 68 81 66 83 58 90 85 NA
```

```
max(s_marks)
```

```
# to get largest item
```

```
## [1] NA
```

```
max(s_marks, na.rm = TRUE)
```

```
#to get the largest value by removing NA value
```

```
##[1] 99
```

```
which(s_marks == max(s_marks, na.rm = TRUE))

#if we want to know which student have the highest marks

## [1] 9

which(s_marks == "40")

#which student have marks=40

## integer(0)

 #because 40 is not present in the vector
```

We can also manipulate the values to get them in any interval For this purpose we use seq() function

```
s_marks[seq(1,length(s_marks),5)]

## [1] 87 90 64 83

#SYNTAX : data[seq(start,end,interval)]
```

In this example we have taken interval as 5 so it will print the first element an every 5 value.

THESE FUNCTIONS ALSO WORKS ON CHARACTER VALUES Let's take the Example of city names

```
cities <- c("Delhi", "Mumbai", "Chandigarh", "Lucknow", "Bangalore", "K
olkata", "Patna", "Chennai","Ahmedabad", "Kanpur")
cities

##  [1] "Delhi"      "Mumbai"     "Chandigarh" "Lucknow"    "Bangalore"

##  [6] "Kolkata"    "Patna"      "Chennai"    "Ahmedabad"  "Kanpur"

length(cities)

## [1] 10

cities[2]

## [1] "Mumbai"

cities[-3:-6]

#display the vector except elements present at indexes 3 to 6

## [1] "Delhi"     "Mumbai"    "Patna"     "Chennai"   "Ahmedabad" "Kan
pur"

which(cities == max(cities))

#display the index of the element having maximum characters
```

```
## [1] 7
```

```
which(cities == "Kolkata")
```

```
## [1] 6
```

SORTING AND REARRANGING

```
sort(s_marks, na.last = TRUE)
```

```
##  [1] 44 55 58 64 66 68 68 70 80 81 83 85 87 89 90 90 96 99 NA NA
```

```
sort(s_marks, na.last = FALSE, decreasing = TRUE)
```

```
##  [1] NA NA 99 96 90 90 89 87 85 83 81 80 70 68 68 66 64 58 55 44
```

```
sort(cities)
```

```
##  [1] "Ahmedabad"  "Bangalore"  "Chandigarh" "Chennai"    "Delhi"
##  [6] "Kanpur"     "Kolkata"    "Lucknow"    "Mumbai"     "Patna"
```

```
sort(cities, decreasing = TRUE)
```

```
##  [1] "Patna"      "Mumbai"     "Lucknow"    "Kolkata"    "Kanpur"
##  [6] "Delhi"      "Chennai"    "Chandigarh" "Bangalore"  "Ahmedabad"
```

```
order(s_marks, na.last = TRUE)
```

```
##  [1]  3  7 17 11 15  5 13 12  4 14 16 19  1 10  6 18  2  9  8 20
```

NOW WE'LL PERFORM THAT SAME FUNCTION ON A DATA FRAME

DATA FRAME AND MATRIX **are two dimensional data objects.**

DATA - POLLUTION IN DELHI

```
library(readr)
poll_delhi <- read.csv("pollution_data_delhiXX.csv" , header = TRUE)
```

```
poll_delhi
```

```
##           Date  PM2.5   PM10   CO   SO2    O3 AQI   AQI_Bucket
## 1   01-01-2019 287.34 461.02 2.54 17.77 39.58 475       Severe
## 2   01-02-2019 331.20 515.72 2.78 18.27 43.30 501       Severe
## 3   01-03-2019 355.40 519.34 2.55 15.12 39.77 537       Severe
## 4   01-04-2019 246.46 388.79 2.01 13.22 28.75 432       Severe
## 5   01-05-2019 262.53 411.49 2.23 15.59 36.33 440       Severe
## 6   01-06-2019 174.26 254.66 1.49 13.65 31.48 371    Very Poor
## 7   01-07-2019 181.74 283.88 1.57 15.49 27.11 331    Very Poor
## 8   01-08-2019 156.26 242.02 1.31 14.67 27.37 340    Very Poor
## 9   01-09-2019 143.66 238.94 1.30 15.09 30.90 321    Very Poor
## 10  01-10-2019 165.95 278.87 1.75 16.81 30.92 317    Very Poor
```

```
## 11  01-11-2019 278.95 432.20 2.90 20.03 35.61 401       Severe
## 12  01-12-2019 311.03 465.89 3.11 17.81 39.78 482       Severe
## 13   1/13/2019 272.45 403.32 2.36 17.83 37.53 464       Severe
## 14   1/14/2019  94.44 173.94 1.04 15.05 31.33 320    Very Poor
## 15   1/15/2019 134.57 229.10 1.37 17.09 29.85 248         Poor
## 16   1/16/2019 271.96 424.07 2.58 20.35 33.75 383    Very Poor
## 17   1/17/2019 358.91 519.49 3.13 19.63 39.76 515       Severe
## 18   1/18/2019 237.22 364.61 2.06 14.70 30.35 437       Severe
## 19   1/19/2019 262.68 402.21 2.35 18.07 34.75 411       Severe
## 20   1/20/2019 314.59 478.43 3.66 20.98 45.03 480       Severe
## 21   1/21/2019 175.49 289.40 1.83 16.82 32.77 413       Severe
## 22   1/22/2019  53.25  72.61 1.01 11.87 27.01 174     Moderate
## 23   1/23/2019 135.68 211.71 1.62 14.17 26.51 186     Moderate
## 24   1/24/2019 130.59 218.93 1.44 15.29 30.57 328    Very Poor
## 25   1/25/2019  81.20 136.43 1.01 14.62 30.01 188     Moderate
## 26   1/26/2019 101.61 169.32 0.98 14.48 35.41 231         Poor
## 27   1/27/2019 136.47 217.45 1.10 15.89 33.34 275         Poor
## 28   1/28/2019 126.48 212.42 1.06 15.76 33.60 301    Very Poor
## 29   1/29/2019 142.00 235.85 1.24 18.56 33.83 297         Poor
## 30   1/30/2019 206.14 334.21 1.76 24.75 35.84 342    Very Poor
## 31   1/31/2019 204.25 335.07 1.90 20.48 37.52 397    Very Poor
## 32  02-01-2019 178.68 265.65 1.34 17.83 33.12 329    Very Poor
## 33  02-02-2019 177.32 266.30 1.24 11.55 30.88 338    Very Poor
## 34  02-03-2019 131.10 209.41 1.32 11.70 29.44 323    Very Poor
## 35  02-04-2019 144.62 248.49 1.81 14.06 33.42 301    Very Poor
## 36  02-05-2019 256.33 408.32 2.95 22.38 46.33 398    Very Poor
## 37  02-06-2019 177.25 270.92 1.64 15.59 34.40 380    Very Poor
## 38  02-07-2019  65.65 107.83 0.84 11.12 30.25 233         Poor
## 39  02-08-2019  72.29 156.60 0.94 12.66 30.59 136     Moderate
## 40  02-09-2019  82.50 165.94 0.96 13.25 33.55 169     Moderate
## 41  02-10-2019 143.31 255.47 1.36 16.41 33.03 277         Poor
## 42  02-11-2019 173.61 304.98 1.78 17.52 37.03 332    Very Poor
## 43  02-12-2019 208.64 357.36 2.09 19.93 43.29 357    Very Poor
## 44   2/13/2019 198.26 336.99 1.94 18.61 43.36 372    Very Poor
## 45   2/14/2019 153.73 237.08 1.42 14.53 32.61 350    Very Poor
## 46   2/15/2019 123.46 194.76 1.22 15.38 28.15 275         Poor
## 47   2/16/2019 112.34 181.64 1.13 12.92 26.45 288         Poor
## 48   2/17/2019  99.27 180.93 1.09 15.14 34.81 247         Poor
## 49   2/18/2019 109.11 215.05 1.18 16.78 33.69 278         Poor
## 50   2/19/2019 126.12 229.75 1.56 15.93 36.66 232         Poor
## 51   2/20/2019 102.30 185.40 1.40 17.16 23.68 305    Very Poor
## 52   2/21/2019  73.33 178.07 1.37 20.10 34.41 158     Moderate
## 53   2/22/2019  73.90 147.86 1.07 13.94 28.45 191     Moderate
## 54   2/23/2019  56.51 115.07 0.85 15.66 37.26 116     Moderate
## 55   2/24/2019  89.34 161.89 1.03 18.99 38.02 169     Moderate
## 56   2/25/2019  92.36 172.59 1.24 18.07 42.49 221         Poor
## 57   2/26/2019  59.97 120.30 0.95 16.88 32.19 144     Moderate
## 58   2/27/2019  47.18  91.52 0.79 12.52 31.38 112     Moderate
## 59   2/28/2019  96.38 180.46 1.25 16.98 35.95 198     Moderate

… Total 547 Rows
```

```r
length(poll_delhi)

#here it will give the number of columns in our data frame
```

```
## [1] 8
```

```r
poll_delhi[length(poll_delhi)-3:length(poll_delhi)]
```

```
##       SO2   CO   PM10  PM2.5       Date
## 1   17.77 2.54 461.02 287.34 01-01-2019
## 2   18.27 2.78 515.72 331.20 01-02-2019
## 3   15.12 2.55 519.34 355.40 01-03-2019
## 4   13.22 2.01 388.79 246.46 01-04-2019
## 5   15.59 2.23 411.49 262.53 01-05-2019
## 6   13.65 1.49 254.66 174.26 01-06-2019
## 7   15.49 1.57 283.88 181.74 01-07-2019
## 8   14.67 1.31 242.02 156.26 01-08-2019
## 9   15.09 1.30 238.94 143.66 01-09-2019
## 10  16.81 1.75 278.87 165.95 01-10-2019
## 11  20.03 2.90 432.20 278.95 01-11-2019
## 12  17.81 3.11 465.89 311.03 01-12-2019
## 13  17.83 2.36 403.32 272.45  1/13/2019
## 14  15.05 1.04 173.94  94.44  1/14/2019
## 15  17.09 1.37 229.10 134.57  1/15/2019
## 16  20.35 2.58 424.07 271.96  1/16/2019
## 17  19.63 3.13 519.49 358.91  1/17/2019
## 18  14.70 2.06 364.61 237.22  1/18/2019
## 19  18.07 2.35 402.21 262.68  1/19/2019
## 20  20.98 3.66 478.43 314.59  1/20/2019
## 21  16.82 1.83 289.40 175.49  1/21/2019
## 22  11.87 1.01  72.61  53.25  1/22/2019
## 23  14.17 1.62 211.71 135.68  1/23/2019
## 24  15.29 1.44 218.93 130.59  1/24/2019
## 25  14.62 1.01 136.43  81.20  1/25/2019
## 26  14.48 0.98 169.32 101.61  1/26/2019
## 27  15.89 1.10 217.45 136.47  1/27/2019
## 28  15.76 1.06 212.42 126.48  1/28/2019
## 29  18.56 1.24 235.85 142.00  1/29/2019
## 30  24.75 1.76 334.21 206.14  1/30/2019
## 31  20.48 1.90 335.07 204.25  1/31/2019
## 32  17.83 1.34 265.65 178.68 02-01-2019
## 33  11.55 1.24 266.30 177.32 02-02-2019
## 34  11.70 1.32 209.41 131.10 02-03-2019
## 35  14.06 1.81 248.49 144.62 02-04-2019
## 36  22.38 2.95 408.32 256.33 02-05-2019
## 37  15.59 1.64 270.92 177.25 02-06-2019
## 38  11.12 0.84 107.83  65.65 02-07-2019
## 39  12.66 0.94 156.60  72.29 02-08-2019
## 40  13.25 0.96 165.94  82.50 02-09-2019
## 41  16.41 1.36 255.47 143.31 02-10-2019  … Total 547 Rows
```

```r
max(poll_delhi$AQI)
```

```
## [1] 659  #highest number in column named AQI.
```

```r
which(poll_delhi == max(poll_delhi$AQI))
```

```
## [1] 3589 #index of the  highest number in the column mentioned above.
```

```r
poll_delhi[seq(1,length(poll_delhi),2)]
```

```
##           Date   PM10   SO2 AQI
## 1   01-01-2019 461.02 17.77 475
## 2   01-02-2019 515.72 18.27 501
## 3   01-03-2019 519.34 15.12 537
## 4   01-04-2019 388.79 13.22 432
## 5   01-05-2019 411.49 15.59 440
## 6   01-06-2019 254.66 13.65 371
## 7   01-07-2019 283.88 15.49 331
## 8   01-08-2019 242.02 14.67 340
## 9   01-09-2019 238.94 15.09 321
## 10  01-10-2019 278.87 16.81 317
## 11  01-11-2019 432.20 20.03 401
## 12  01-12-2019 465.89 17.81 482
## 13   1/13/2019 403.32 17.83 464
## 14   1/14/2019 173.94 15.05 320
## 15   1/15/2019 229.10 17.09 248
## 16   1/16/2019 424.07 20.35 383
## 17   1/17/2019 519.49 19.63 515
## 18   1/18/2019 364.61 14.70 437
## 19   1/19/2019 402.21 18.07 411
## 20   1/20/2019 478.43 20.98 480
## 21   1/21/2019 289.40 16.82 413
## 22   1/22/2019  72.61 11.87 174
## 23   1/23/2019 211.71 14.17 186
## 24   1/24/2019 218.93 15.29 328
## 25   1/25/2019 136.43 14.62 188
## 26   1/26/2019 169.32 14.48 231
## 27   1/27/2019 217.45 15.89 275
## 28   1/28/2019 212.42 15.76 301
## 29   1/29/2019 235.85 18.56 297
## 30   1/30/2019 334.21 24.75 342
## 31   1/31/2019 335.07 20.48 397
## 32  02-01-2019 265.65 17.83 329
## 33  02-02-2019 266.30 11.55 338
## 34  02-03-2019 209.41 11.70 323
## 35  02-04-2019 248.49 14.06 301
## 36  02-05-2019 408.32 22.38 398
## 37  02-06-2019 270.92 15.59 380
## 38  02-07-2019 107.83 11.12 233
```

```r
sort(poll_delhi$PM10) #to sort elements of a particular column.
```

```
##   [1]  19.51  30.45  34.15  37.10  38.50  40.43  45.04  45.48  46.35  47.75
##  [11]  51.03  51.91  53.02  53.19  55.75  56.18  56.28  56.71  57.31  58.36
##  [21]  58.60  60.59  61.06  61.26  61.47  62.19  62.49  62.61  62.69  63.51
##  [31]  64.12  64.17  64.66  65.65  65.84  66.66  66.98  67.01  67.15  67.50
##  [41]  67.73  67.82  68.03  68.59  69.36  69.49  72.61  72.61  72.98  73.86
##  [51]  75.81  76.32  76.67  77.01  77.48  77.64  77.76  78.08  79.04  79.22
##  [61]  80.22  80.35  80.36  80.47  80.49  81.44  81.97  82.81  83.02  83.22
##  [71]  83.26  83.33  83.66  84.53  85.26  85.45  85.49  85.68  85.83  85.89
##  [81]  86.13  86.31  87.05  87.27  87.50  87.80  88.15  88.35  89.47  89.58
##  [91]  90.29  90.44  91.17  91.52  91.55  91.79  91.88  91.98  92.41  92.58
## [101]  92.70  93.49  94.02  94.40  94.90  95.63  95.71  96.13  96.22  96.29
## [111]  96.88  97.32  98.32  99.98 100.29 100.55 100.62 101.79 101.93 102.53
## [121] 103.01 103.27 103.62 104.96 105.16 105.55 105.68 106.47 106.51 107.36
## [131] 107.83 107.99 107.99 108.22 108.73 110.39 112.08 112.57 112.93 113.25
## [141] 113.92 114.01 114.88 114.91 115.07 115.63 115.82 115.91 116.56 117.10
## [151] 117.66 117.84 118.10 118.65 119.66 119.87 119.98 120.23 120.30 120.82
## [161] 123.23 123.44 124.24 124.26 124.28 126.12 126.43 126.53 126.90 127.82
## [171] 128.10 128.36 128.60 129.33 129.40 129.46 129.47 130.49 130.88 130.99
## [181] 132.06 132.61 133.11 133.47 133.75 134.17 134.36 134.46 135.40 135.84
## [191] 136.43 136.58 136.77 137.08 137.61 137.67 137.81 138.46 138.56 140.15
## [201] 140.22 140.23 140.60 142.76 143.11 143.56 143.79 143.97 144.35 144.37
## [211] 144.47 144.70 145.05 145.85 147.86 148.67 148.79 149.10 149.12 149.82
## [221] 150.28 150.34 150.39 151.57 151.80 151.81 152.39 152.70 152.98 153.96
## [231] 155.18 155.39 155.41 155.90 155.94 156.29 156.60 156.93 158.30 159.20
## [241] 159.36 160.15 160.91 160.96 161.36 161.76 161.89 162.18 162.41 162.56
## [251] 162.83 163.67 163.77 163.80 165.50 165.94 168.11 168.32 168.43 169.16
## [261] 169.32 169.61 170.14 170.33 171.58 172.28 172.59 172.68 173.94 174.33
## [271] 174.78 174.84 175.93 176.22 177.16 177.53 178.07 178.21 179.08 179.30
## [281] 179.78 180.46 180.84 180.93 181.26 181.64 181.67 182.20 182.54 182.89
## [291] 183.77 184.01 184.12 184.81 185.40 186.49 186.75 187.24 187.59 187.62
## [301] 187.93 188.16 188.21 188.73 189.32 189.69 190.11 190.98 191.04 191.17
## [311] 192.47 192.68 193.76 193.79 194.58 194.76 195.15 196.87 197.17 197.90
## [321] 198.24 199.28 200.79 200.90 201.19 201.27 202.01 202.05 202.75 203.14
## [331] 203.36 203.39 203.51 203.54 203.77 204.19 205.43 205.43 205.56 206.04
## [341] 207.19 208.65 208.72 209.02 209.41 209.42 209.49 211.47 211.71 212.42
## [351] 212.70 213.26 213.47 213.67 214.88 215.05 215.18 217.00 217.45 217.74
## [361] 218.13 218.38 218.72 218.93 219.06 219.79 223.08 223.69 224.02 224.14
## [371] 225.50 225.67 225.78 227.07 229.08 229.10 229.55 229.75 230.00 230.50
## [381] 230.98 231.19 233.25 233.49 235.00 235.85 236.45 237.08 237.61 237.81
## [391] 238.17 238.91 238.94 238.99 239.15 239.47 240.19 240.57 240.88 241.56
## [401] 242.02 243.43 244.21 245.20 245.62 246.05 247.47 247.53 247.62 248.49
## [411] 249.97 253.44 253.82 254.66 255.31 255.41 255.42 255.47 257.27 257.94
## [421] 258.28 258.62 260.26 262.58 265.12 265.65 266.30 268.58 270.03 270.77
## [431] 270.87 270.92 271.27 274.39 274.68 274.83 277.28 277.92 278.71 278.87
## [441] 279.78 280.23 280.99 281.53 282.83 282.84 283.88 284.13 285.26 285.73
## [451] 286.39 289.05 289.40 290.60 292.44 292.56 293.00 294.27 295.05 295.47
## [461] 295.84 296.44 296.98 296.98 298.35 299.85 300.70 301.82 303.71 304.98
## [471] 306.12 308.65 309.09 311.58 316.40 320.44 321.68 323.97 325.36 327.22
## [481] 329.77 330.91 332.60 334.21 335.07 336.99 337.04 337.77 338.18 340.56
## [491] 346.51 348.44 348.45 351.28 352.26 353.94 355.16 355.22 357.36 362.10
## [501] 363.30 363.70 363.77 364.61 371.45 378.89 380.54 385.13 386.16 388.79
## [511] 401.03 402.21 403.32 408.32 411.49 412.59 414.70 415.23 416.79 420.35
```

```
## [521] 424.07 426.12 432.20 439.31 440.77 449.06 452.65 454.35 461.02 465.89
## [531] 467.74 478.43 483.87 486.35 487.77 500.04 500.15 503.28 515.72 519.34
## [541] 519.49 529.59 541.19 545.66 576.12 584.15 706.58
```

```
sort(poll_delhi$PM10,decreasing = TRUE)
```

```
##   [1] 706.58 584.15 576.12 545.66 541.19 529.59 519.49 519.34 515.72 503.28
##  [11] 500.15 500.04 487.77 486.35 483.87 478.43 467.74 465.89 461.02 454.35
##  [21] 452.65 449.06 440.77 439.31 432.20 426.12 424.07 420.35 416.79 415.23
##  [31] 414.70 412.59 411.49 408.32 403.32 402.21 401.03 388.79 386.16 385.13
##  [41] 380.54 378.89 371.45 364.61 363.77 363.70 363.30 362.10 357.36 355.22
##  [51] 355.16 353.94 352.26 351.28 348.45 348.44 346.51 340.56 338.18 337.77
##  [61] 337.04 336.99 335.07 334.21 332.60 330.91 329.77 327.22 325.36 323.97
##  [71] 321.68 320.44 316.40 311.58 309.09 308.65 306.12 304.98 303.71 301.82
##  [81] 300.70 299.85 298.35 296.98 296.98 296.44 295.84 295.47 295.05 294.27
##  [91] 293.00 292.56 292.44 290.60 289.40 289.05 286.39 285.73 285.26 284.13
## [101] 283.88 282.84 282.83 281.53 280.99 280.23 279.78 278.87 278.71 277.92
## [111] 277.28 274.83 274.68 274.39 271.27 270.92 270.87 270.77 270.03 268.58
## [121] 266.30 265.65 265.12 262.58 260.26 258.62 258.28 257.94 257.27 255.47
## [131] 255.42 255.41 255.31 254.66 253.82 253.44 249.97 248.49 247.62 247.53
## [141] 247.47 246.05 245.62 245.20 244.21 243.43 242.02 241.56 240.88 240.57
## [151] 240.19 239.47 239.15 238.99 238.94 238.91 238.17 237.81 237.61 237.08
## [161] 236.45 235.85 235.00 233.49 233.25 231.19 230.98 230.50 230.00 229.75
## [171] 229.55 229.10 229.08 227.07 225.78 225.67 225.50 224.14 224.02 223.69
## [181] 223.08 219.79 219.06 218.93 218.72 218.38 218.13 217.74 217.45 217.00
## [191] 215.18 215.05 214.88 213.67 213.47 213.26 212.70 212.42 211.71 211.47
## [201] 209.49 209.42 209.41 209.02 208.72 208.65 207.19 206.04 205.56 205.43
## [211] 205.43 204.19 203.77 203.54 203.51 203.39 203.36 203.14 202.75 202.05
## [221] 202.01 201.27 201.19 200.90 200.79 199.28 198.24 197.90 197.17 196.87
## [231] 195.15 194.76 194.58 193.79 193.76 192.68 192.47 191.17 191.04 190.98
## [241] 190.11 189.69 189.32 188.73 188.21 188.16 187.93 187.62 187.59 187.24
## [251] 186.75 186.49 185.40 184.81 184.12 184.01 183.77 182.89 182.54 182.20
## [261] 181.67 181.64 181.26 180.93 180.84 180.46 179.78 179.30 179.08 178.21
## [271] 178.07 177.53 177.16 176.22 175.93 174.84 174.78 174.33 173.94 172.68
## [281] 172.59 172.28 171.58 170.33 170.14 169.61 169.32 169.16 168.43 168.32
## [291] 168.11 165.94 165.50 163.80 163.77 163.67 162.83 162.56 162.41 162.18
## [301] 161.89 161.76 161.36 160.96 160.91 160.15 159.36 159.20 158.30 156.93
## [311] 156.60 156.29 155.94 155.90 155.41 155.39 155.18 153.96 152.98 152.70
## [321] 152.39 151.81 151.80 151.57 150.39 150.34 150.28 149.82 149.12 149.10
## [331] 148.79 148.67 147.86 145.85 145.05 144.70 144.47 144.37 144.35 143.97
## [341] 143.79 143.56 143.11 142.76 140.60 140.23 140.22 140.15 138.56 138.46
## [351] 137.81 137.67 137.61 137.08 136.77 136.58 136.43 135.84 135.40 134.46
## [361] 134.36 134.17 133.75 133.47 133.11 132.61 132.06 130.99 130.88 130.49
## [371] 129.47 129.46 129.40 129.33 128.60 128.36 128.10 127.82 126.90 126.53
## [381] 126.43 126.12 124.28 124.26 124.24 123.44 123.23 120.82 120.30 120.23
## [391] 119.98 119.87 119.66 118.65 118.10 117.84 117.66 117.10 116.56 115.91
## [401] 115.82 115.63 115.07 114.91 114.88 114.01 113.92 113.25 112.93 112.57
## [411] 112.08 110.39 108.73 108.22 107.99 107.99 107.83 107.36 106.51 106.47
## [421] 105.68 105.55 105.16 104.96 103.62 103.27 103.01 102.53 101.93 101.79
## [431] 100.62 100.55 100.29  99.98  98.32  97.32  96.88  96.29  96.22  96.13
## [441]  95.71  95.63  94.90  94.40  94.02  93.49  92.70  92.58  92.41  91.98
## [451]  91.88  91.79  91.55  91.52  91.17  90.44  90.29  89.58  89.47  88.35
## [461]  88.15  87.80  87.50  87.27  87.05  86.31  86.13  85.89  85.83  85.68
## [471]  85.49  85.45  85.26  84.53  83.66  83.33  83.26  83.22  83.02  82.81
## [481]  81.97  81.44  80.49  80.47  80.36  80.35  80.22  79.22  79.04  78.08
```

```
## [491]  77.76  77.64  77.48  77.01  76.67  76.32  75.81  73.86  72.98  72.61
## [501]  72.61  69.49  69.36  68.59  68.03  67.82  67.73  67.50  67.15  67.01
## [511]  66.98  66.66  65.84  65.65  64.66  64.17  64.12  63.51  62.69  62.61
## [521]  62.49  62.19  61.47  61.26  61.06  60.59  58.60  58.36  57.31  56.71
## [531]  56.28  56.18  55.75  53.19  53.02  51.91  51.03  47.75  46.35  45.48
## [541]  45.04  40.43  38.50  37.10  34.15  30.45  19.51
```

```
order(poll_delhi$PM10)
```

```
##   [1] 229 230 453 228 431 517 208 452 209 222 272 224 483 516 454 225 227 270
##  [19] 218 226 332 490 458 207 221 262 107 541 459 206 540 210 455 265 457 515
##  [37] 271 456 223 430 198 237 450 238 257 269  22 258 219 475 451 465 212 251
##  [55] 539 460 542 199 464 197 220 215 484 518 523 244 273 211 489 256 250 266
##  [73] 214 169 439 432 417 175 481 213 231 268 522 477 187 185 216 493 249 473
##  [91] 241 248 524  58 108 188 426 547 274 534 468 497 463 498 277 538 232 275
## [109] 267 264 461 261 519 186 333 448 535 245 501 474 240 525 449 217  62 202
## [127] 242 263 138 488  38 239 533 278 537 496 543 436 466 476 184 520 462 485
## [145]  54 486 425 491 281 482 144 201 467 145 203  81 176 532  57 494 243 418
## [163] 440 347 134 174 252 500 495 479 331 348 190 530  71 233 259 234 171 478
## [181] 374 255 350  83 434 394 528 447  64 260  25 146 469 168 388 196 280 480
## [199]  82 279 433 351  75 235 487 246 521 276  74 441 253 470 420 373  53 526
## [217] 189 123 393 236 531  84 247 349 502 389 158 529 382 204 437 109 506 177
## [235] 544 527  39 205 200 499 334 442 435 167 424 513  55  68 511 443 139 379
## [253] 507  69 321  40 104  65 395  85  26 492  91 396 155 410  56 147  14 254
## [271] 411 292 103 372 173  90  52 328 380 510 409  59  63  48  60  47 322 419
## [289]  61 282 148  76 546 446  51  86 106 503 105 159 384 172 165 428 182 164
## [307] 512 471 135 472 335 504 110 429 141  46 444 124  80 381 383  67 445 398
## [325] 397 142  77 356 149 293 284 291 427 375 423 101  70 402 536  93 310 505
## [343] 156 438  34 285 509  92  23  28  66 295 508 329 352  49 283 102  27  73
## [361] 323 136 140  24 111 153 170 416 421 327  99 385 330  78 336  15 514  50
## [379] 143 313 178  72 183 161 294  29 157  45  79 412 404 414   9  89 179 287
## [397] 357 390 163 337   8 399 116 290 376 150 371 100 405  35 125 312 403   6
## [415] 112 401 286  41  88 370 359 422  87 114 160  32  33 296 181  94 298  37
## [433] 180 115 166 119 113 338 191  10 289  98 195  95 122 413   7 120 288 118
## [451] 415 131  21 299 406 311 320 194 408 324 126 314 309 387 152 360 154 117
## [469]  96  42 407 545 391 343 137 130 300 121 377 369 378 392 326  30  31  44
## [487] 368 151 365 400 297 341 386 133 192 355 342  97  43 340 344 193 325  18
## [505] 361 127 315 353 308   4 358  19  13  36   5 346 129 128 339 301  16 362
## [523]  11 367 302 354 345 303   1  12 306  20 366 364 162 304 363 132   2   3
## [541]  17 316 317 318 319 305 307
```

```
rank(poll_delhi$PM2.5,ties.method = 'first')
```

```
##   [1] 526 537 539 510 515 472 481 459 445 463 523 530 521 344 429 518 540 505
##  [19] 516 532 475 163 430 423 299 368 431 417 441 494 493 480 478 425 447 513
##  [37] 477 239 266 307 443 470 496 487 457 410 393 360 385 416 372 270 275 188
##  [55] 325 335 214 143 353 341 386 225 379 251 310 377 334 269 276 304 193 323
##  [73] 388 280 248 340 362 374 326 392  89 180 164 236 283 293 399 394 371 207
##  [91] 208 327 315 404 414 354 456 300 320 370 257 222 184 154 173 209  15  94
## [109] 263 330 348 387 375 317 319 305 267 285 312 350 389 289 156 292 367 403
## [127] 420 444 433 347 369 479 466 200 314 282 337 132 303 342 290 232 336 130
## [145] 129 175 234 241 279 316 322 349 273 243 174 272 259 172 255 365 278 401
## [163] 212 159 166 301 118 107  43 298 122 262 247 119  25 165 244 238 221 199
## [181] 237 192 264 104  84 145  65  93 147 121 253 287 284 228 227 116  50  51
## [199]  56 226 139 151 216 268 229  17  45  24   9  20  41  42  66  71  90 140
## [217] 157  40  53  33   2   6  23   8  13  59  35   4   1   3  91 114 141 131
## [235] 150 218  60  22 110 117  63 128 191  73 138 260 271  62  79  70  54 187
## [253] 233 296 196  61  28  78 189 210  76  14 109  96  26  68  80  44  18   5
## [271]  16   7  38  64 106 197 124  99 205 182 134 288 363 357 373 426 398 428
## [289] 448 407 329 286 356 396 339 408 469 421 450 488 529 527 524 531 545 534
```

```
## [307] 547 520 484 366 482 458 438 485 512 544 543 541 546 471 352 384 418 474
## [325] 498 491 422 383 409 395 240  47 167 355 411 439 437 462 519 508 503 506
## [343] 490 504 528 525 345 294 346 321 338 446 514 533 511 449 467 517 464 486
## [361] 507 522 535 538 509 542 536 502 500 465 461 405 400 324 424 454 495 499
## [379] 397 390 436 382 413 412 442 501 489 277 297 432 492 497 376 291 332 333
## [397] 391 406 435 483 452 402 453 451 455 473 476 468 351 274 309 427 460 415
## [415] 434 359 162 230 378 258 419 440 381 311 215 123 361 318 308  49  11 112
## [433] 265 198 313 177 190 249  97 219 224 254 252 331 328 306 295 169 181  87
## [451] 136  29  10  32  72  75  55  34  48 111 155 213  58  39  83 195 161  92
## [469] 217 201 203 256  30  37  77 160  86 202 171 242  57 108  31 115 176 148
## [487] 220  69  52  27 186 204  82 153 168 133 127 125 250 185 144 302 358 364
## [505] 380 245 183 343 281 178 223 179 126  95  21  19  12 113 137 152 246 120
## [523]  74 105 142 235 206 170 231 158 261 146 103  67  85 194  81 100  98  36
## [541]  46  88 149 101 211 135 102
```

```
rank(poll_delhi$AQI_Bucket,ties.method = 'average')
```

```
##   [1] 447.0 447.0 447.0 447.0 447.0 507.0 507.0 507.0 507.0 507.0 447.0 447.0
##  [13] 447.0 507.0 292.5 507.0 447.0 447.0 447.0 447.0 447.0 113.0 113.0 507.0
##  [25] 113.0 292.5 292.5 507.0 292.5 507.0 507.0 507.0 507.0 507.0 507.0 507.0
##  [37] 507.0 292.5 113.0 113.0 292.5 507.0 507.0 507.0 507.0 292.5 292.5 292.5
##  [49] 292.5 292.5 507.0 113.0 113.0 113.0 113.0 292.5 113.0 113.0 113.0 292.5
##  [61] 292.5 113.0 292.5 113.0 113.0 292.5 292.5 113.0 113.0 113.0 113.0 113.0
##  [73] 292.5 113.0 113.0 113.0 292.5 292.5 292.5 292.5 113.0 113.0 113.0 113.0
##  [85] 113.0 113.0 292.5 292.5 292.5 113.0 113.0 113.0 292.5 292.5 292.5 292.5
##  [97] 507.0 292.5 292.5 292.5 113.0 292.5 113.0 113.0 113.0 292.5 113.0 113.0
## [109] 113.0 113.0 292.5 292.5 292.5 292.5 292.5 292.5 292.5 292.5 292.5 292.5
## [121] 292.5 292.5 292.5 113.0 292.5 292.5 507.0 507.0 507.0 507.0 292.5 507.0
## [133] 507.0 292.5 113.0 113.0 292.5 113.0 113.0 292.5 292.5 113.0 292.5 113.0
## [145] 113.0 113.0 113.0 113.0 113.0 292.5 292.5 292.5 292.5 292.5 292.5 113.0
## [157] 292.5 113.0 113.0 292.5 292.5 507.0 292.5 113.0 113.0 292.5 292.5 113.0
## [169] 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 292.5 292.5
## [181] 292.5 113.0 292.5 113.0 113.0 113.0 393.5 113.0 113.0 113.0 113.0 292.5
## [193] 292.5 507.0 292.5 113.0 113.0 393.5 393.5 113.0 113.0 113.0 113.0 113.0
## [205] 113.0 393.5 393.5 393.5 393.5 393.5 393.5 393.5 393.5 393.5 113.0 113.0
## [217] 113.0 393.5 113.0 393.5 393.5 393.5 393.5 393.5 393.5 393.5 393.5 393.5
## [229] 393.5 393.5 393.5 113.0 113.0 113.0 113.0 113.0 113.0 393.5 113.0 113.0
## [241] 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 393.5 393.5 393.5 113.0
## [253] 113.0 113.0 113.0 113.0 393.5 393.5 113.0 113.0 113.0 393.5 393.5 113.0
## [265] 393.5 393.5 113.0 393.5 393.5 393.5 393.5 393.5 393.5 393.5 113.0 113.0
## [277] 113.0 113.0 113.0 113.0 113.0 113.0 292.5 292.5 292.5 292.5 292.5 292.5
## [289] 507.0 292.5 292.5 113.0 292.5 292.5 292.5 292.5 507.0 507.0 507.0 507.0
## [301] 447.0 447.0 447.0 447.0 447.0 447.0 447.0 447.0 507.0 292.5 292.5 507.0
## [313] 507.0 507.0 507.0 447.0 447.0 447.0 447.0 447.0 292.5 292.5 292.5 507.0
## [325] 507.0 507.0 507.0 292.5 292.5 292.5 113.0 113.0 393.5 113.0 292.5 292.5
## [337] 507.0 507.0 507.0 447.0 507.0 447.0 507.0 507.0 447.0 447.0 507.0 113.0
## [349] 292.5 292.5 113.0 292.5 507.0 447.0 447.0 507.0 507.0 507.0 507.0 507.0
## [361] 507.0 447.0 447.0 447.0 447.0 447.0 447.0 447.0 507.0 507.0 507.0 292.5
## [373] 292.5 292.5 292.5 507.0 507.0 507.0 507.0 292.5 292.5 292.5 292.5 292.5
## [385] 292.5 507.0 507.0 292.5 113.0 292.5 507.0 507.0 507.0 292.5 113.0 292.5
## [397] 292.5 292.5 292.5 507.0 507.0 292.5 507.0 507.0 507.0 507.0 507.0 507.0
## [409] 292.5 113.0 113.0 292.5 507.0 292.5 292.5 292.5 113.0 113.0 292.5 113.0
## [421] 292.5 507.0 292.5 292.5 113.0 113.0 113.0 292.5 113.0 113.0 393.5 393.5
## [433] 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0
## [445] 292.5 292.5 113.0 113.0 113.0 393.5 393.5 393.5 393.5 393.5 393.5 393.5
## [457] 393.5 393.5 393.5 393.5 113.0 113.0 393.5 393.5 393.5 113.0 113.0 113.0
## [469] 113.0 113.0 113.0 113.0 113.0 113.0 393.5 113.0 113.0 113.0 113.0 113.0
## [481] 113.0 113.0 393.5 113.0 113.0 113.0 113.0 113.0 113.0 393.5 113.0 113.0
## [493] 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 292.5 292.5
## [505] 292.5 113.0 113.0 113.0 292.5 113.0 113.0 113.0 113.0 113.0 113.0 393.5
## [517] 393.5 393.5 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0
```

```
## [529] 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 113.0 393.5
## [541] 393.5 393.5 113.0 113.0 113.0 292.5 113.0
```

```
which(poll_delhi$AQI_Bucket == "Satisfactory")
```

```
#this will display that in data, which values are “satisfactory” in AQI_Bucket column
```

```
##  [1] 187 198 199 206 207 208 209 210 211 212 213 214 218 220 221 222 223 224 225
## [20] 226 227 228 229 230 231 238 249 250 251 257 258 262 263 265 266 268 269 270
## [39] 271 272 273 274 333 431 432 450 451 452 453 454 455 456 457 458 459 460 463
## [58] 464 465 475 483 490 516 517 518 540 541 542
```

```
poll_delhi$AQI < 400 | poll_delhi=='Poor'
```

```
##         Date PM2.5  PM10    CO   SO2    O3   AQI AQI_Bucket
##   [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##   [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##   [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##   [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##   [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##   [6,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##   [7,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##   [8,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##   [9,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [10,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [14,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [15,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [16,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [17,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [18,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [19,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [20,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [21,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE      FALSE
##  [22,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [23,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [24,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [25,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [26,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [27,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [28,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [29,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [30,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [31,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [32,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [33,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [34,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [35,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [36,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [37,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [38,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [39,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [40,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
##  [41,]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE       TRUE
```

SELECTING PARTS OF DATA FRAME

SYNTAX : data[row,column]

```
poll_delhi[10,2]

#to display item from 10th row and 2nd column

## [1] 165.95

poll_delhi[3,1:4]

 #it will display items present at 3rd row of columns 1 to 4

##         Date PM2.5   PM10   CO
## 3 01-03-2019 355.4 519.34 2.55

poll_delhi[c(1,5,7,9),]

#items present at rows 1,5,7,9 of each column will be displayed

##         Date  PM2.5   PM10   CO   SO2    O3 AQI AQI_Bucket
## 1 01-01-2019 287.34 461.02 2.54 17.77 39.58 475     Severe
## 5 01-05-2019 262.53 411.49 2.23 15.59 36.33 440     Severe
## 7 01-07-2019 181.74 283.88 1.57 15.49 27.11 331  Very Poor
## 9 01-09-2019 143.66 238.94 1.30 15.09 30.90 321  Very Poor

poll_delhi[c(1,4,7,3),3]

## [1] 461.02 388.79 283.88 519.34

#items present at rows 1,4,7,3 of 3rd column will be displayed

poll_delhi[c(1,5,8,3),"PM2.5" ]

 #display the 1st,5th,8th,3rd elements present in the  column named PM2.
5

## [1] 287.34 262.53 156.26 355.40

str(poll_delhi)

#str() function is used to display the structure of the data

## 'data.frame':    547 obs. of  8 variables:
##  $ Date      : chr  "01-01-2019" "01-02-2019" "01-03-2019" "01-04-2019"
 ...
##  $ PM2.5     : num  287 331 355 246 263 ...
##  $ PM10      : num  461 516 519 389 411 ...
##  $ CO        : num  2.54 2.78 2.55 2.01 2.23 1.49 1.57 1.31 1.3 1.75 ...
##  $ SO2       : num  17.8 18.3 15.1 13.2 15.6 ...
##  $ O3        : num  39.6 43.3 39.8 28.8 36.3 ...
##  $ AQI       : int  475 501 537 432 440 371 331 340 321 317 ...
##  $ AQI_Bucket: chr  "Severe" "Severe" "Severe" "Severe" ...
```

```r
attach(poll_delhi)
search()
```

#attach() function is used to access the variables present in the data framework without calling th e data frame.

```
##  [1] ".GlobalEnv"        "poll_delhi"        "package:readr"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "package:methods"
## [10] "Autoloads"         "package:base"
```

```r
with(poll_delhi,PM10)
```

#in this example we don't need to mention the data frame name with the variabl e/column.

```
##   [1] 461.02 515.72 519.34 388.79 411.49 254.66 283.88 242.02 238.94 278.87
##  [11] 432.20 465.89 403.32 173.94 229.10 424.07 519.49 364.61 402.21 478.43
##  [21] 289.40  72.61 211.71 218.93 136.43 169.32 217.45 212.42 235.85 334.21
##  [31] 335.07 265.65 266.30 209.41 248.49 408.32 270.92 107.83 156.60 165.94
##  [41] 255.47 304.98 357.36 336.99 237.08 194.76 181.64 180.93 215.05 229.75
##  [51] 185.40 178.07 147.86 115.07 161.89 172.59 120.30  91.52 180.46 181.26
##  [61] 182.54 105.16 180.84 135.40 168.32 212.70 199.28 162.18 163.80 205.43
##  [71] 129.40 231.19 217.74 144.35 140.60 184.01 202.01 227.07 237.61 197.17
##  [81] 119.87 138.56 133.47 150.34 169.16 186.49 260.26 257.27 238.99 177.53
##  [91] 170.14 211.47 206.04 270.77 281.53 303.71 355.22 280.23 225.50 247.53
## [101] 204.19 217.00 175.93 168.11 187.59 186.75  62.49  91.55 155.39 193.76
## [111] 219.06 255.31 277.28 262.58 274.39 244.21 301.82 285.73 274.83 284.13
## [121] 323.97 282.83 149.10 196.87 249.97 295.84 378.89 415.23 414.70 320.44
## [131] 289.05 503.28 351.28 124.28 191.04 218.38 316.40 106.51 162.83 218.72
## [141] 194.58 201.27 230.00 117.66 118.65 136.58 172.68 183.77 202.75 246.05
## [151] 337.77 298.35 219.79 300.70 171.58 208.72 236.45 152.39 187.62 265.12
## [161] 233.49 487.77 240.88 189.69 188.21 274.68 160.96 137.08  84.53 223.08
## [171] 130.88 188.16 177.16 126.12  85.68 119.98 155.90 230.98 239.15 271.27
## [181] 270.03 189.32 233.25 113.92  87.80  99.98  87.50  91.79 148.79 128.60
## [191] 278.71 352.26 363.70 294.27 280.99 137.67  79.22  67.73  78.08 158.30
## [201] 117.84 105.55 119.66 153.96 156.93  63.51  61.26  45.04  46.35  64.17
## [211]  82.81  76.67  85.89  83.66  80.35  88.15 104.96  57.31  72.98  80.22
## [221]  61.47  47.75  67.15  51.91  56.18  58.36  56.28  37.10  19.51  30.45
## [231]  86.13  95.71 129.46 130.49 142.76 149.82  67.82  68.59 107.99 103.01
## [241]  90.29 105.68 123.23  81.44 101.79 143.56 150.39  90.44  89.47  83.26
## [251]  77.01 126.43 144.47 174.33 132.61  83.22  69.36  72.61 129.47 135.84
## [261]  97.32  62.19 106.47  96.29  65.65  83.33  96.22  86.31  69.49  56.71
## [271]  66.98  51.03  81.97  92.41  96.13 143.97  94.90 108.22 140.15 137.81
## [281] 116.56 182.89 215.18 203.36 209.42 255.42 239.47 285.26 279.78 245.20
## [291] 203.39 174.84 203.14 235.00 213.26 268.58 346.51 270.87 290.60 321.68
## [301] 420.35 440.77 454.35 500.04 584.15 467.74 706.58 386.16 296.98 207.19
## [311] 292.56 253.44 230.50 296.44 380.54 529.59 541.19 545.66 576.12 293.00
## [321] 165.50 181.67 218.13 295.47 363.77 332.60 224.14 178.21 213.67 225.78
## [331] 128.10  58.60 100.29 159.36 192.47 229.08 241.56 277.92 416.79 362.10
## [341] 348.44 355.16 311.58 363.30 452.65 412.59 124.26 128.36 151.57 133.11
## [351] 140.23 214.88 385.13 449.06 353.94 202.05 240.19 401.03 258.28 299.85
## [361] 371.45 426.12 500.15 486.35 338.18 483.87 439.31 337.04 327.22 257.94
## [371] 247.47 176.22 145.85 132.06 203.54 245.62 325.36 329.77 163.67 179.08
## [381] 197.90 152.98 198.24 187.93 225.67 348.45 296.98 137.61 151.81 240.57
```

```
## [391] 309.09 330.91 149.12 134.17 168.43 170.33 201.19 200.90 243.43 340.56
## [401] 255.41 205.43 253.82 238.17 247.62 292.44 306.12 295.05 179.78 172.28
## [411] 174.78 237.81 282.84 238.91 286.39 223.69  85.49 123.44 182.20 145.05
## [421] 224.02 258.62 203.77 161.36 115.82  91.88 203.51 188.73 193.79  67.50
## [431]  38.50  85.45 140.22 133.75 160.91 112.57 155.18 209.02  85.26 124.24
## [441] 144.37 160.15 162.56 195.15 200.79 184.81 134.46 100.55 103.62  68.03
## [451]  75.81  45.48  34.15  55.75  64.66  67.01  65.84  61.06  62.69  77.64
## [461]  96.88 114.88  94.02  79.04  76.32 112.93 118.10  92.70 136.77 144.70
## [471] 190.98 191.17  89.58 102.53  73.86 113.25  87.27 130.99 127.82 138.46
## [481]  85.83 117.10  53.02  80.36 114.91 115.63 143.11 107.36  83.02  60.59
## [491] 115.91 169.61  88.35 120.82 126.90 110.39  93.49  94.40 159.20 126.53
## [501] 101.93 151.80 187.24 192.68 208.65 155.41 163.77 213.47 209.49 179.30
## [511] 162.41 190.11 161.76 229.55  66.66  53.19  40.43  80.47  98.32 114.01
## [521] 143.79  87.05  80.49  91.17 103.27 148.67 156.29 134.36 152.70 129.33
## [531] 150.28 120.23 107.99  92.58 100.62 205.56 108.73  95.63  77.48  64.12
## [541]  62.61  77.76 112.08 155.94 308.65 184.12  91.98
```

```
head(poll_delhi)

#head() function is used to display first few lines/rows of the data ob
ject.
```

```
##         Date  PM2.5   PM10   CO   SO2    O3 AQI AQI_Bucket
## 1 01-01-2019 287.34 461.02 2.54 17.77 39.58 475     Severe
## 2 01-02-2019 331.20 515.72 2.78 18.27 43.30 501     Severe
## 3 01-03-2019 355.40 519.34 2.55 15.12 39.77 537     Severe
## 4 01-04-2019 246.46 388.79 2.01 13.22 28.75 432     Severe
## 5 01-05-2019 262.53 411.49 2.23 15.59 36.33 440     Severe
## 6 01-06-2019 174.26 254.66 1.49 13.65 31.48 371  Very Poor
```

```
tail(poll_delhi)

#tail() function is used to display last few line/rows of the data obje
cts.
```

```
##          Date PM2.5   PM10   CO   SO2    O3 AQI   AQI_Bucket
## 542 6/25/2020 38.37  77.76 1.02 11.28 25.92  86 Satisfactory
## 543 6/26/2020 50.01 112.08 1.01 11.66 24.82 104     Moderate
## 544 6/27/2020 39.80 155.94 0.87 10.38 18.88 112     Moderate
## 545 6/28/2020 59.52 308.65 0.94 10.70 18.05 196     Moderate
## 546 6/29/2020 44.86 184.12 0.88 11.58 26.61 233         Poor
## 547 6/30/2020 39.80  91.98 0.84 10.51 37.29 114     Moderate
```

```
head(poll_delhi,n=3)

#We can also display desired number of rows from head as well as tail.
```

```
##         Date  PM2.5   PM10   CO   SO2    O3 AQI AQI_Bucket
## 1 01-01-2019 287.34 461.02 2.54 17.77 39.58 475     Severe
```

```
## 2 01-02-2019 331.20 515.72 2.78 18.27 43.30 501     Severe
## 3 01-03-2019 355.40 519.34 2.55 15.12 39.77 537     Severe
```

```
summary(poll_delhi)
```

```
##      Date               PM2.5              PM10              CO        
##  Length:547         Min.   : 10.24   Min.   : 19.51   Min.   :0.520  
##  Class :character   1st Qu.: 46.02   1st Qu.:112.33   1st Qu.:0.930  
##  Mode  :character   Median : 73.84   Median :176.22   Median :1.130  
##                     Mean   : 99.17   Mean   :195.80   Mean   :1.267  
##                     3rd Qu.:123.56   3rd Qu.:249.23   3rd Qu.:1.420  
##                     Max.   :582.28   Max.   :706.58   Max.   :3.660  
##       SO2               O3              AQI         AQI_Bucket       
##  Min.   : 6.67   Min.   :16.51   Min.   : 51.0   Length:547        
##  1st Qu.:10.42   1st Qu.:30.45   1st Qu.:123.5   Class :character  
##  Median :13.25   Median :37.78   Median :191.0   Mode  :character  
##  Mean   :14.00   Mean   :39.53   Mean   :215.5                     
##  3rd Qu.:16.82   3rd Qu.:47.66   3rd Qu.:285.0                     
##  Max.   :33.50   Max.   :76.32   Max.   :659.0
```

```
names(poll_delhi)
```

```
#this function shows the name of the columns by default.
```

```
## [1] "Date"       "PM2.5"      "PM10"       "CO"         "SO2"       
## [6] "O3"         "AQI"        "AQI_Bucket"
```

```
#if we want to display the names of rows, the function used if row.name
s()
```

```
row.names(poll_delhi)
```

```
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12" 
##  [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24" 
##  [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36" 
##  [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48" 
##  [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60" 
##  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72" 
##  [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84" 
##  [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96" 
##  [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
## [109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
## [145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156"
## [157] "157" "158" "159" "160" "161" "162" "163" "164" "165" "166" "167" "168"
## [169] "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
## [181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192"
## [193] "193" "194" "195" "196" "197" "198" "199" "200" "201" "202" "203" "204"
## [205] "205" "206" "207" "208" "209" "210" "211" "212" "213" "214" "215" "216"
## [217] "217" "218" "219" "220" "221" "222" "223" "224" "225" "226" "227" "228"
## [229] "229" "230" "231" "232" "233" "234" "235" "236" "237" "238" "239" "240"
## [241] "241" "242" "243" "244" "245" "246" "247" "248" "249" "250" "251" "252"
## [253] "253" "254" "255" "256" "257" "258" "259" "260" "261" "262" "263" "264"
## [265] "265" "266" "267" "268" "269" "270" "271" "272" "273" "274" "275" "276"
## [277] "277" "278" "279" "280" "281" "282" "283" "284" "285" "286" "287" "288"
## [289] "289" "290" "291" "292" "293" "294" "295" "296" "297" "298" "299" "300"
```

```
## [301] "301" "302" "303" "304" "305" "306" "307" "308" "309" "310" "311" "312"
## [313] "313" "314" "315" "316" "317" "318" "319" "320" "321" "322" "323" "324"
## [325] "325" "326" "327" "328" "329" "330" "331" "332" "333" "334" "335" "336"
## [337] "337" "338" "339" "340" "341" "342" "343" "344" "345" "346" "347" "348"
## [349] "349" "350" "351" "352" "353" "354" "355" "356" "357" "358" "359" "360"
## [361] "361" "362" "363" "364" "365" "366" "367" "368" "369" "370" "371" "372"
## [373] "373" "374" "375" "376" "377" "378" "379" "380" "381" "382" "383" "384"
## [385] "385" "386" "387" "388" "389" "390" "391" "392" "393" "394" "395" "396"
## [397] "397" "398" "399" "400" "401" "402" "403" "404" "405" "406" "407" "408"
## [409] "409" "410" "411" "412" "413" "414" "415" "416" "417" "418" "419" "420"
## [421] "421" "422" "423" "424" "425" "426" "427" "428" "429" "430" "431" "432"
## [433] "433" "434" "435" "436" "437" "438" "439" "440" "441" "442" "443" "444"
## [445] "445" "446" "447" "448" "449" "450" "451" "452" "453" "454" "455" "456"
## [457] "457" "458" "459" "460" "461" "462" "463" "464" "465" "466" "467" "468"
## [469] "469" "470" "471" "472" "473" "474" "475" "476" "477" "478" "479" "480"
## [481] "481" "482" "483" "484" "485" "486" "487" "488" "489" "490" "491" "492"
## [493] "493" "494" "495" "496" "497" "498" "499" "500" "501" "502" "503" "504"
## [505] "505" "506" "507" "508" "509" "510" "511" "512" "513" "514" "515" "516"
## [517] "517" "518" "519" "520" "521" "522" "523" "524" "525" "526" "527" "528"
## [529] "529" "530" "531" "532" "533" "534" "535" "536" "537" "538" "539" "540"
## [541] "541" "542" "543" "544" "545" "546" "547"
```

```
dimnames(poll_delhi)
```

```
#dimnames() display both the name of rows as well as columns.
```

```
## [[1]]
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"
##  [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"
##  [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"
##  [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"
##  [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
##  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"
##  [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"
##  [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"
##  [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
## [109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
## [145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156"
## [157] "157" "158" "159" "160" "161" "162" "163" "164" "165" "166" "167" "168"
## [169] "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
## [181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192"
## [193] "193" "194" "195" "196" "197" "198" "199" "200" "201" "202" "203" "204"
## [205] "205" "206" "207" "208" "209" "210" "211" "212" "213" "214" "215" "216"
## [217] "217" "218" "219" "220" "221" "222" "223" "224" "225" "226" "227" "228"
## [229] "229" "230" "231" "232" "233" "234" "235" "236" "237" "238" "239" "240"
## [241] "241" "242" "243" "244" "245" "246" "247" "248" "249" "250" "251" "252"
## [253] "253" "254" "255" "256" "257" "258" "259" "260" "261" "262" "263" "264"
## [265] "265" "266" "267" "268" "269" "270" "271" "272" "273" "274" "275" "276"
## [277] "277" "278" "279" "280" "281" "282" "283" "284" "285" "286" "287" "288"
## [289] "289" "290" "291" "292" "293" "294" "295" "296" "297" "298" "299" "300"
## [301] "301" "302" "303" "304" "305" "306" "307" "308" "309" "310" "311" "312"
## [313] "313" "314" "315" "316" "317" "318" "319" "320" "321" "322" "323" "324"
## [325] "325" "326" "327" "328" "329" "330" "331" "332" "333" "334" "335" "336"
## [337] "337" "338" "339" "340" "341" "342" "343" "344" "345" "346" "347" "348"
## [349] "349" "350" "351" "352" "353" "354" "355" "356" "357" "358" "359" "360"
## [361] "361" "362" "363" "364" "365" "366" "367" "368" "369" "370" "371" "372"
## [373] "373" "374" "375" "376" "377" "378" "379" "380" "381" "382" "383" "384"
## [385] "385" "386" "387" "388" "389" "390" "391" "392" "393" "394" "395" "396"
## [397] "397" "398" "399" "400" "401" "402" "403" "404" "405" "406" "407" "408"
```

```
## [409] "409" "410" "411" "412" "413" "414" "415" "416" "417" "418" "419" "420"
## [421] "421" "422" "423" "424" "425" "426" "427" "428" "429" "430" "431" "432"
## [433] "433" "434" "435" "436" "437" "438" "439" "440" "441" "442" "443" "444"
## [445] "445" "446" "447" "448" "449" "450" "451" "452" "453" "454" "455" "456"
## [457] "457" "458" "459" "460" "461" "462" "463" "464" "465" "466" "467" "468"
## [469] "469" "470" "471" "472" "473" "474" "475" "476" "477" "478" "479" "480"
## [481] "481" "482" "483" "484" "485" "486" "487" "488" "489" "490" "491" "492"
## [493] "493" "494" "495" "496" "497" "498" "499" "500" "501" "502" "503" "504"
## [505] "505" "506" "507" "508" "509" "510" "511" "512" "513" "514" "515" "516"
## [517] "517" "518" "519" "520" "521" "522" "523" "524" "525" "526" "527" "528"
## [529] "529" "530" "531" "532" "533" "534" "535" "536" "537" "538" "539" "540"
## [541] "541" "542" "543" "544" "545" "546" "547"
##
## [[2]]
## [1] "Date"       "PM2.5"      "PM10"       "CO"         "SO2"
## [6] "O3"         "AQI"        "AQI_Bucket"
```

```
poll_delhi.t=t(poll_delhi)
poll_delhi.t
```

```
#t() is transpose of the data, here we are treating data in matrix for
so we can interchange the rows and columns, this function is called the
 transpose.
```

```
##            [,1]         [,2]         [,3]         [,4]         [,5]
## Date       "01-01-2019" "01-02-2019" "01-03-2019" "01-04-2019" "01-05-2019"
## PM2.5      "287.34"     "331.20"     "355.40"     "246.46"     "262.53"
## PM10       "461.02"     "515.72"     "519.34"     "388.79"     "411.49"
## CO         "2.54"       "2.78"       "2.55"       "2.01"       "2.23"
## SO2        "17.77"      "18.27"      "15.12"      "13.22"      "15.59"
## O3         "39.58"      "43.30"      "39.77"      "28.75"      "36.33"
## AQI        "475"        "501"        "537"        "432"        "440"
## AQI_Bucket "Severe"     "Severe"     "Severe"     "Severe"     "Severe"
##            [,6]         [,7]         [,8]         [,9]         [,10]
## Date       "01-06-2019" "01-07-2019" "01-08-2019" "01-09-2019" "01-10-2019"
## PM2.5      "174.26"     "181.74"     "156.26"     "143.66"     "165.95"
## PM10       "254.66"     "283.88"     "242.02"     "238.94"     "278.87"
## CO         "1.49"       "1.57"       "1.31"       "1.30"       "1.75"
## SO2        "13.65"      "15.49"      "14.67"      "15.09"      "16.81"
## O3         "31.48"      "27.11"      "27.37"      "30.90"      "30.92"
## AQI        "371"        "331"        "340"        "321"        "317"
## AQI_Bucket "Very Poor"  "Very Poor"  "Very Poor"  "Very Poor"  "Very Poor"
##            [,11]        [,12]        [,13]       [,14]       [,15]
## Date       "01-11-2019" "01-12-2019" "1/13/2019" "1/14/2019" "1/15/2019"
## PM2.5      "278.95"     "311.03"     "272.45"    " 94.44"    "134.57"
## PM10       "432.20"     "465.89"     "403.32"    "173.94"    "229.10"
## CO         "2.90"       "3.11"       "2.36"      "1.04"      "1.37"
## SO2        "20.03"      "17.81"      "17.83"     "15.05"     "17.09"
## O3         "35.61"      "39.78"      "37.53"     "31.33"     "29.85"
## AQI        "401"        "482"        "464"       "320"       "248"
## AQI_Bucket "Severe"     "Severe"     "Severe"    "Very Poor" "Poor"
##            [,16]       [,17]       [,18]       [,19]       [,20]
## Date       "1/16/2019" "1/17/2019" "1/18/2019" "1/19/2019" "1/20/2019"
## PM2.5      "271.96"    "358.91"    "237.22"    "262.68"    "314.59"
## PM10       "424.07"    "519.49"    "364.61"    "402.21"    "478.43"
## CO         "2.58"      "3.13"      "2.06"      "2.35"      "3.66"
```

```
## SO2         "20.35"     "19.63"     "14.70"     "18.07"     "20.98"
## O3          "33.75"     "39.76"     "30.35"     "34.75"     "45.03"
## AQI         "383"       "515"       "437"       "411"       "480"
## AQI_Bucket "Very Poor" "Severe"    "Severe"    "Severe"    "Severe"
##             [,21]       [,22]       [,23]       [,24]       [,25]
## Date        "1/21/2019" "1/22/2019" "1/23/2019" "1/24/2019" "1/25/2019"
## PM2.5       "175.49"    " 53.25"    "135.68"    "130.59"    " 81.20"
## PM10        "289.40"    " 72.61"    "211.71"    "218.93"    "136.43"
```

```
library(readr)
pollution <- read_csv("delhi_19-20.csv")

## Parsed with column specification:
## cols(
##   Date = col_character(),
##   PM2.5 = col_double(),
##   PM10 = col_double(),
##   CO = col_double(),
##   SO2 = col_double(),
##   O3 = col_double(),
##   AQI = col_double(),
##   AQI_Bucket = col_character()
## )

#we are here making a variable pollution and storing the data from the
government site.
pollution

## # A tibble: 547 x 8
##    Date      PM2.5  PM10    CO   SO2    O3   AQI AQI_Bucket
##    <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
##  1 1/1/2019   287.  461.  2.54  17.8  39.6   475 Severe
##  2 1/2/2019   331.  516.  2.78  18.3  43.3   501 Severe
##  3 1/3/2019   355.  519.  2.55  15.1  39.8   537 Severe
##  4 1/4/2019   246.  389.  2.01  13.2  28.8   432 Severe
##  5 1/5/2019   263.  411.  2.23  15.6  36.3   440 Severe
##  6 1/6/2019   174.  255.  1.49  13.6  31.5   371 Very Poor
##  7 1/7/2019   182.  284.  1.57  15.5  27.1   331 Very Poor
##  8 1/8/2019   156.  242.  1.31  14.7  27.4   340 Very Poor
##  9 1/9/2019   144.  239.  1.3   15.1  30.9   321 Very Poor
## 10 1/10/2019  166.  279.  1.75  16.8  30.9   317 Very Poor
## # ... with 537 more rows

str(pollution)

## tibble [547 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date      : chr [1:547] "1/1/2019" "1/2/2019" "1/3/2019" "1/4/2019" ...
##  $ PM2.5     : num [1:547] 287 331 355 246 263 ...
##  $ PM10      : num [1:547] 461 516 519 389 411 ...
##  $ CO        : num [1:547] 2.54 2.78 2.55 2.01 2.23 1.49 1.57 1.31 1.3 1.75 ...
##  $ SO2       : num [1:547] 17.8 18.3 15.1 13.2 15.6 ...
##  $ O3        : num [1:547] 39.6 43.3 39.8 28.8 36.3 ...
##  $ AQI       : num [1:547] 475 501 537 432 440 371 331 340 321 317 ...
```

```
##  $ AQI_Bucket: chr [1:547] "Severe" "Severe" "Severe" "Severe" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   PM2.5 = col_double(),
##   ..   PM10 = col_double(),
##   ..   CO = col_double(),
##   ..   SO2 = col_double(),
##   ..   O3 = col_double(),
##   ..   AQI = col_double(),
##   ..   AQI_Bucket = col_character()
##   .. )
```

#checking the format of data stored in our variable .In this case the data stored in pollution is a data frame.

DATA FRAME AS A LIST

```
pollution.list=list(pollution$Date,pollution$PM2.5,pollution$PM10,pollution$CO,pollution$SO2,pollution$O3,pollution$AQI,pollution$AQI_Bucket)
```

#making the given data from government site in a form of list with entries of different columns as elements of list

```
pollution.list
```

```
## [[1]]
##   [1] "1/1/2019"   "1/2/2019"   "1/3/2019"   "1/4/2019"   "1/5/2019"
##   [6] "1/6/2019"   "1/7/2019"   "1/8/2019"   "1/9/2019"   "1/10/2019"
##  [11] "1/11/2019"  "1/12/2019"  "1/13/2019"  "1/14/2019"  "1/15/2019"
##  [16] "1/16/2019"  "1/17/2019"  "1/18/2019"  "1/19/2019"  "1/20/2019"
##  [21] "1/21/2019"  "1/22/2019"  "1/23/2019"  "1/24/2019"  "1/25/2019"
##  [26] "1/26/2019"  "1/27/2019"  "1/28/2019"  "1/29/2019"  "1/30/2019"
##  [31] "1/31/2019"  "2/1/2019"   "2/2/2019"   "2/3/2019"   "2/4/2019"
##  [36] "2/5/2019"   "2/6/2019"   "2/7/2019"   "2/8/2019"   "2/9/2019"
##  [41] "2/10/2019"  "2/11/2019"  "2/12/2019"  "2/13/2019"  "2/14/2019"
##  [46] "2/15/2019"  "2/16/2019"  "2/17/2019"  "2/18/2019"  "2/19/2019"
##  [51] "2/20/2019"  "2/21/2019"  "2/22/2019"  "2/23/2019"  "2/24/2019"
##  [56] "2/25/2019"  "2/26/2019"  "2/27/2019"  "2/28/2019"  "3/1/2019"
##  [61] "3/2/2019"   "3/3/2019"   "3/4/2019"   "3/5/2019"   "3/6/2019"
##  [66] "3/7/2019"   "3/8/2019"   "3/9/2019"   "3/10/2019"  "3/11/2019"
##  [71] "3/12/2019"  "3/13/2019"  "3/14/2019"  "3/15/2019"  "3/16/2019"
##  [76] "3/17/2019"  "3/18/2019"  "3/19/2019"  "3/20/2019"  "3/21/2019"
##  [81] "3/22/2019"  "3/23/2019"  "3/24/2019"  "3/25/2019"  "3/26/2019"
##  [86] "3/27/2019"  "3/28/2019"  "3/29/2019"  "3/30/2019"  "3/31/2019"
##  [91] "4/1/2019"   "4/2/2019"   "4/3/2019"   "4/4/2019"   "4/5/2019"
##  [96] "4/6/2019"   "4/7/2019"   "4/8/2019"   "4/9/2019"   "4/10/2019"
## [101] "4/11/2019"  "4/12/2019"  "4/13/2019"  "4/14/2019"  "4/15/2019"
## [106] "4/16/2019"  "4/17/2019"  "4/18/2019"  "4/19/2019"  "4/20/2019"
## [111] "4/21/2019"  "4/22/2019"  "4/23/2019"  "4/24/2019"  "4/25/2019"
## [116] "4/26/2019"  "4/27/2019"  "4/28/2019"  "4/29/2019"  "4/30/2019"
## [121] "5/1/2019"   "5/2/2019"   "5/3/2019"   "5/4/2019"   "5/5/2019"
## [126] "5/6/2019"   "5/7/2019"   "5/8/2019"   "5/9/2019"   "5/10/2019"
## [131] "5/11/2019"  "5/12/2019"  "5/13/2019"  "5/14/2019"  "5/15/2019"
## [136] "5/16/2019"  "5/17/2019"  "5/18/2019"  "5/19/2019"  "5/20/2019"
## [141] "5/21/2019"  "5/22/2019"  "5/23/2019"  "5/24/2019"  "5/25/2019"
## [146] "5/26/2019"  "5/27/2019"  "5/28/2019"  "5/29/2019"  "5/30/2019"
## [151] "5/31/2019"  "6/1/2019"   "6/2/2019"   "6/3/2019"   "6/4/2019"
```

```
## [156] "6/5/2019"   "6/6/2019"   "6/7/2019"   "6/8/2019"   "6/9/2019"
## [161] "6/10/2019"  "6/11/2019"  "6/12/2019"  "6/13/2019"  "6/14/2019"
## [166] "6/15/2019"  "6/16/2019"  "6/17/2019"  "6/18/2019"  "6/19/2019"
## [171] "6/20/2019"  "6/21/2019"  "6/22/2019"  "6/23/2019"  "6/24/2019"
## [176] "6/25/2019"  "6/26/2019"  "6/27/2019"  "6/28/2019"  "6/29/2019"
## [181] "6/30/2019"  "7/1/2019"   "7/2/2019"   "7/3/2019"   "7/4/2019"
## [186] "7/5/2019"   "7/6/2019"   "7/7/2019"   "7/8/2019"   "7/9/2019"
## [191] "7/10/2019"  "7/11/2019"  "7/12/2019"  "7/13/2019"  "7/14/2019"
## [196] "7/15/2019"  "7/16/2019"  "7/17/2019"  "7/18/2019"  "7/19/2019"
## [201] "7/20/2019"  "7/21/2019"  "7/22/2019"  "7/23/2019"  "7/24/2019"
#now we can ascess different parts of list using indexes for example as
 given below
pollution.list[2:3]

## [[1]]
##   [1] 287.34 331.20 355.40 246.46 262.53 174.26 181.74 156.26 143.66 165.95
##  [11] 278.95 311.03 272.45  94.44 134.57 271.96 358.91 237.22 262.68 314.59
##  [21] 175.49  53.25 135.68 130.59  81.20 101.61 136.47 126.48 142.00 206.14
##  [31] 204.25 178.68 177.32 131.10 144.62 256.33 177.25  65.65  72.29  82.50
##  [41] 143.31 173.61 208.64 198.26 153.73 123.46 112.34  99.27 109.11 126.12
##  [51] 102.30  73.33  73.90  56.51  89.34  92.36  59.97  47.18  96.38  93.83
##  [61] 109.60  62.92 105.93  68.86  83.45 104.61  92.07  73.31  73.98  82.17
##  [71]  57.28  89.17 111.20  74.88  68.64  93.61 100.32 103.77  89.82 112.29
##  [81]  38.43  55.62  53.28  65.00  75.82  78.37 116.55 112.67 102.27  58.96
##  [91]  59.08  90.10  85.02 119.71 125.44  96.60 152.91  81.30  87.74 101.94
## [101]  70.25  61.34  55.93  51.51  54.84  59.08  22.14  38.92  71.62  90.55
## [111]  95.24 110.16 104.17  85.70  86.41  82.37  73.06  76.44  84.04  95.72
## [121] 111.57  77.72  52.16  77.96 101.25 118.90 128.78 143.34 137.20  95.07
## [131] 101.83 178.22 170.81  58.22  84.81  75.45  92.53  44.28  81.99  93.97
## [141]  77.78  64.24  92.40  43.86  43.66  55.13  64.66  65.82  74.80  85.21
## [151]  88.57  95.40  73.36  66.94  54.95  73.35  70.61  54.81  69.75 101.05
## [161]  74.48 118.38  59.86  52.66  53.79  81.86  42.06  40.18  30.39  79.96
## [171]  42.76  71.15  68.27  42.06  26.29  53.65  67.36  65.30  61.21  58.08
## [181]  65.24  57.27  71.68  39.99  36.82  48.60  34.25  38.81  49.12  42.56
## [191]  69.21  76.99  76.35  63.18  63.16  42.04  31.51  32.06  32.92  63.02
## [201]  46.43  50.57  60.38  73.08  63.26  22.79  30.63  26.16  21.13  24.45
## [211]  30.10  30.22  34.26  35.06  38.54  46.69  52.20  30.04  32.63  27.69
## [221]  14.59  16.73  25.53  20.53  21.68  33.21  28.78  15.52  10.24  15.51
## [231]  38.55  41.74  46.69  44.16  50.49  60.71  33.49  25.06  40.68  42.04
## [241]  33.92  43.50  57.16  35.13  46.15  70.64  73.33  33.90  35.81  34.98
## [251]  32.83  56.31  64.48  79.60  57.83  33.59  26.65  35.52  56.60  59.14
## [261]  35.41  21.74  40.33  39.23  26.51  34.46  35.88  30.52  22.90  15.66
## [271]  22.52  16.91  29.42  33.92  40.16  57.99  43.07  39.48  58.75  55.88
## [281]  44.85  77.02 100.82  98.07 103.45 132.57 116.33 133.35 146.37 121.68
## [291]  90.41  76.77  98.05 115.16  93.51 122.12 173.56 129.90 148.73 198.81
## [301] 301.39 287.56 279.32 314.32 388.45 320.26 582.28 272.30 190.41 101.09
## [311] 182.47 154.14 140.80 192.35 252.48 384.34 383.36 371.42 423.52 174.03
## [321]  96.34 108.40 126.92 175.19 217.70 201.17 130.41 107.86 122.37 114.27
## [331]  65.70  31.03  53.79  96.96 123.66 141.12 140.07 163.24 271.99 241.69
## [341] 233.79 239.98 200.33 236.40 299.85 284.18  94.62  78.59  94.67  88.05
## [351]  93.07 144.00 261.16 316.09 249.87 146.81 171.47 262.69 166.71 196.97
## [361] 240.07 275.60 326.79 333.43 245.53 372.14 327.04 228.90 220.60 170.21
## [371] 159.64 120.31 117.98  89.26 130.82 151.10 207.23 218.98 115.62 111.57
## [381] 139.06 107.65 125.30 125.04 142.18 222.81 198.88  74.45  79.83 136.72
## [391] 201.20 209.97 104.45  77.95  91.56  92.04 112.13 120.86 138.50 188.42
## [401] 150.08 118.57 150.15 149.67 151.43 174.40 176.57 172.12  95.72  73.84
```

```
## [411]  83.07 132.63 158.69 125.53 138.46  98.84  53.11  63.62 105.71  70.59
## [421] 127.75 141.81 107.28  83.48  60.01  42.83  99.34  86.26  82.90  31.43
## [431]  21.32  40.96  72.10  58.04  84.35  55.19  56.92  68.72  39.41  60.99
## [441]  61.61  69.70  68.93  90.62  90.31  82.39  78.61  54.26  55.85  37.87
## [451]  45.69  26.89  21.23  27.59  35.08  35.31  32.88  28.51  31.36  40.79
## [461]  51.79  59.91  33.12  29.70  36.13  57.67  53.03  38.57  60.60  58.22
## [471]  58.31  69.80  26.93  29.09  35.49  52.86  37.17  58.25  54.71  65.91
## [481]  33.09  40.24  27.18  41.75  55.18  49.96  61.04  34.67  32.56  26.63
## [491]  56.24  58.44  36.02  51.09  54.18  44.30  43.26  43.19  68.80  56.17
give.names=c('date','PM2.5','PM10','CO','SO2','03','AQI','AQI_TYPE')
names(pollution.list)=give.names
pollution.list

## $date
##   [1] "1/1/2019"   "1/2/2019"   "1/3/2019"   "1/4/2019"   "1/5/2019"
##   [6] "1/6/2019"   "1/7/2019"   "1/8/2019"   "1/9/2019"   "1/10/2019"
##  [11] "1/11/2019"  "1/12/2019"  "1/13/2019"  "1/14/2019"  "1/15/2019"
##  [16] "1/16/2019"  "1/17/2019"  "1/18/2019"  "1/19/2019"  "1/20/2019"
##  [21] "1/21/2019"  "1/22/2019"  "1/23/2019"  "1/24/2019"  "1/25/2019"
##  [26] "1/26/2019"  "1/27/2019"  "1/28/2019"  "1/29/2019"  "1/30/2019"
##  [31] "1/31/2019"  "2/1/2019"   "2/2/2019"   "2/3/2019"   "2/4/2019"
##  [36] "2/5/2019"   "2/6/2019"   "2/7/2019"   "2/8/2019"   "2/9/2019"
##  [41] "2/10/2019"  "2/11/2019"  "2/12/2019"  "2/13/2019"  "2/14/2019"
##  [46] "2/15/2019"  "2/16/2019"  "2/17/2019"  "2/18/2019"  "2/19/2019"
##  [51] "2/20/2019"  "2/21/2019"  "2/22/2019"  "2/23/2019"  "2/24/2019"
##  [56] "2/25/2019"  "2/26/2019"  "2/27/2019"  "2/28/2019"  "3/1/2019"
##  [61] "3/2/2019"   "3/3/2019"   "3/4/2019"   "3/5/2019"   "3/6/2019"
##  [66] "3/7/2019"   "3/8/2019"   "3/9/2019"   "3/10/2019"  "3/11/2019"
##  [71] "3/12/2019"  "3/13/2019"  "3/14/2019"  "3/15/2019"  "3/16/2019"
##  [76] "3/17/2019"  "3/18/2019"  "3/19/2019"  "3/20/2019"  "3/21/2019"
##  [81] "3/22/2019"  "3/23/2019"  "3/24/2019"  "3/25/2019"  "3/26/2019"
##  [86] "3/27/2019"  "3/28/2019"  "3/29/2019"  "3/30/2019"  "3/31/2019"
##  [91] "4/1/2019"   "4/2/2019"   "4/3/2019"   "4/4/2019"   "4/5/2019"
##  [96] "4/6/2019"   "4/7/2019"   "4/8/2019"   "4/9/2019"   "4/10/2019"
## [101] "4/11/2019"  "4/12/2019"  "4/13/2019"  "4/14/2019"  "4/15/2019"
## [106] "4/16/2019"  "4/17/2019"  "4/18/2019"  "4/19/2019"  "4/20/2019"
## [111] "4/21/2019"  "4/22/2019"  "4/23/2019"  "4/24/2019"  "4/25/2019"
## [116] "4/26/2019"  "4/27/2019"  "4/28/2019"  "4/29/2019"  "4/30/2019"
## [121] "5/1/2019"   "5/2/2019"   "5/3/2019"   "5/4/2019"   "5/5/2019"
## [126] "5/6/2019"   "5/7/2019"   "5/8/2019"   "5/9/2019"   "5/10/2019"
## [131] "5/11/2019"  "5/12/2019"  "5/13/2019"  "5/14/2019"  "5/15/2019"
## [136] "5/16/2019"  "5/17/2019"  "5/18/2019"  "5/19/2019"  "5/20/2019"
## [141] "5/21/2019"  "5/22/2019"  "5/23/2019"  "5/24/2019"  "5/25/2019"
## [146] "5/26/2019"  "5/27/2019"  "5/28/2019"  "5/29/2019"  "5/30/2019"
## [151] "5/31/2019"  "6/1/2019"   "6/2/2019"   "6/3/2019"   "6/4/2019"
## [156] "6/5/2019"   "6/6/2019"   "6/7/2019"   "6/8/2019"   "6/9/2019"
## [161] "6/10/2019"  "6/11/2019"  "6/12/2019"  "6/13/2019"  "6/14/2019"
## [166] "6/15/2019"  "6/16/2019"  "6/17/2019"  "6/18/2019"  "6/19/2019"
## [171] "6/20/2019"  "6/21/2019"  "6/22/2019"  "6/23/2019"  "6/24/2019"
## [176] "6/25/2019"  "6/26/2019"  "6/27/2019"  "6/28/2019"  "6/29/2019"
## [181] "6/30/2019"  "7/1/2019"   "7/2/2019"   "7/3/2019"   "7/4/2019"
## [186] "7/5/2019"   "7/6/2019"   "7/7/2019"   "7/8/2019"   "7/9/2019"
## [191] "7/10/2019"  "7/11/2019"  "7/12/2019"  "7/13/2019"  "7/14/2019"
## [196] "7/15/2019"  "7/16/2019"  "7/17/2019"  "7/18/2019"  "7/19/2019"
## [201] "7/20/2019"  "7/21/2019"  "7/22/2019"  "7/23/2019"  "7/24/2019"
## [206] "7/25/2019"  "7/26/2019"  "7/27/2019"  "7/28/2019"  "7/29/2019"
## [211] "7/30/2019"  "7/31/2019"  "8/1/2019"   "8/2/2019"   "8/3/2019"
## [216] "8/4/2019"   "8/5/2019"   "8/6/2019"   "8/7/2019"   "8/8/2019"
## [221] "8/9/2019"   "8/10/2019"  "8/11/2019"  "8/12/2019"  "8/13/2019"
```

```
## [226] "8/14/2019"  "8/15/2019"  "8/16/2019"  "8/17/2019"  "8/18/2019"
## [231] "8/19/2019"  "8/20/2019"  "8/21/2019"  "8/22/2019"  "8/23/2019"
## [236] "8/24/2019"  "8/25/2019"  "8/26/2019"  "8/27/2019"  "8/28/2019"
## [241] "8/29/2019"  "8/30/2019"  "8/31/2019"  "9/1/2019"   "9/2/2019"
## [246] "9/3/2019"   "9/4/2019"   "9/5/2019"   "9/6/2019"   "9/7/2019"
## [251] "9/8/2019"   "9/9/2019"   "9/10/2019"  "9/11/2019"  "9/12/2019"
## [256] "9/13/2019"  "9/14/2019"  "9/15/2019"  "9/16/2019"  "9/17/2019"
## [261] "9/18/2019"  "9/19/2019"  "9/20/2019"  "9/21/2019"  "9/22/2019"
## [266] "9/23/2019"  "9/24/2019"  "9/25/2019"  "9/26/2019"  "9/27/2019"
## [271] "9/28/2019"  "9/29/2019"  "9/30/2019"  "10/1/2019"  "10/2/2019"
#giving name to each element of the list according to the data provided
 in list
```

CONSTRUCTING A MATRIX

```
#the given data is converted into matrix by binding the data columnwise
 i.e.. we will get the data in matrix where given input will be columns
 of matrix
pollution_mat1=cbind(pollution$Date,pollution$PM2.5,pollution$PM10,poll
ution$CO,pollution$SO2,pollution$O3,pollution$AQI,pollution$AQI_Bucket)
pollution_mat1
```

```
##        [,1]         [,2]     [,3]     [,4]   [,5]    [,6]    [,7]
##   [1,] "1/1/2019"   "287.34" "461.02" "2.54" "17.77" "39.58" "475"
##   [2,] "1/2/2019"   "331.2"  "515.72" "2.78" "18.27" "43.3"  "501"
##   [3,] "1/3/2019"   "355.4"  "519.34" "2.55" "15.12" "39.77" "537"
##   [4,] "1/4/2019"   "246.46" "388.79" "2.01" "13.22" "28.75" "432"
##   [5,] "1/5/2019"   "262.53" "411.49" "2.23" "15.59" "36.33" "440"
##   [6,] "1/6/2019"   "174.26" "254.66" "1.49" "13.65" "31.48" "371"
##   [7,] "1/7/2019"   "181.74" "283.88" "1.57" "15.49" "27.11" "331"
##   [8,] "1/8/2019"   "156.26" "242.02" "1.31" "14.67" "27.37" "340"
##   [9,] "1/9/2019"   "143.66" "238.94" "1.3"  "15.09" "30.9"  "321"
##  [10,] "1/10/2019"  "165.95" "278.87" "1.75" "16.81" "30.92" "317"
##  [11,] "1/11/2019"  "278.95" "432.2"  "2.9"  "20.03" "35.61" "401"
##  [12,] "1/12/2019"  "311.03" "465.89" "3.11" "17.81" "39.78" "482"
##  [13,] "1/13/2019"  "272.45" "403.32" "2.36" "17.83" "37.53" "464"
##  [14,] "1/14/2019"  "94.44"  "173.94" "1.04" "15.05" "31.33" "320"
##  [15,] "1/15/2019"  "134.57" "229.1"  "1.37" "17.09" "29.85" "248"
##  [16,] "1/16/2019"  "271.96" "424.07" "2.58" "20.35" "33.75" "383"
##  [17,] "1/17/2019"  "358.91" "519.49" "3.13" "19.63" "39.76" "515"
##  [18,] "1/18/2019"  "237.22" "364.61" "2.06" "14.7"  "30.35" "437"
##  [19,] "1/19/2019"  "262.68" "402.21" "2.35" "18.07" "34.75" "411"
##  [20,] "1/20/2019"  "314.59" "478.43" "3.66" "20.98" "45.03" "480"
##  [21,] "1/21/2019"  "175.49" "289.4"  "1.83" "16.82" "32.77" "413"
##  [22,] "1/22/2019"  "53.25"  "72.61"  "1.01" "11.87" "27.01" "174"
##  [23,] "1/23/2019"  "135.68" "211.71" "1.62" "14.17" "26.51" "186"
##  [24,] "1/24/2019"  "130.59" "218.93" "1.44" "15.29" "30.57" "328"
##  [25,] "1/25/2019"  "81.2"   "136.43" "1.01" "14.62" "30.01" "188"
##  [26,] "1/26/2019"  "101.61" "169.32" "0.98" "14.48" "35.41" "231"
```

```
##  [27,] "1/27/2019"  "136.47" "217.45" "1.1"  "15.89" "33.34" "275"
##  [28,] "1/28/2019"  "126.48" "212.42" "1.06" "15.76" "33.6"  "301"
##  [29,] "1/29/2019"  "142"    "235.85" "1.24" "18.56" "33.83" "297"
##  [30,] "1/30/2019"  "206.14" "334.21" "1.76" "24.75" "35.84" "342"
##  [31,] "1/31/2019"  "204.25" "335.07" "1.9"  "20.48" "37.52" "397"
##  [32,] "2/1/2019"   "178.68" "265.65" "1.34" "17.83" "33.12" "329"
##  [33,] "2/2/2019"   "177.32" "266.3"  "1.24" "11.55" "30.88" "338"
##  [34,] "2/3/2019"   "131.1"  "209.41" "1.32" "11.7"  "29.44" "323"
#similary we can construct that data in matrix Row wise.
#the given data is converted into matrix by binding the data rowwise i.
e.. we will get the data in matrix where given input will be rows of ma
trix
pollution_mat2=rbind(pollution$Date,pollution$PM2.5,pollution$PM10,poll
ution$CO,pollution$SO2,pollution$O3,pollution$AQI,pollution$AQI_Bucket,
dimnames(give.names))
pollution_mat2
```

```
##      [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] "1/1/2019" "1/2/2019" "1/3/2019" "1/4/2019" "1/5/2019" "1/6/2019"
## [2,] "287.34"   "331.2"    "355.4"    "246.46"   "262.53"   "174.26"
## [3,] "461.02"   "515.72"   "519.34"   "388.79"   "411.49"   "254.66"
## [4,] "2.54"     "2.78"     "2.55"     "2.01"     "2.23"     "1.49"
## [5,] "17.77"    "18.27"    "15.12"    "13.22"    "15.59"    "13.65"
## [6,] "39.58"    "43.3"     "39.77"    "28.75"    "36.33"    "31.48"
## [7,] "475"      "501"      "537"      "432"      "440"      "371"
## [8,] "Severe"   "Severe"   "Severe"   "Severe"   "Severe"   "Very Poor"
##      [,7]       [,8]       [,9]       [,10]      [,11]      [,12]
## [1,] "1/7/2019" "1/8/2019" "1/9/2019" "1/10/2019" "1/11/2019" "1/12/2019"
## [2,] "181.74"   "156.26"   "143.66"   "165.95"   "278.95"   "311.03"
## [3,] "283.88"   "242.02"   "238.94"   "278.87"   "432.2"    "465.89"
## [4,] "1.57"     "1.31"     "1.3"      "1.75"     "2.9"      "3.11"
## [5,] "15.49"    "14.67"    "15.09"    "16.81"    "20.03"    "17.81"
## [6,] "27.11"    "27.37"    "30.9"     "30.92"    "35.61"    "39.78"
## [7,] "331"      "340"      "321"      "317"      "401"      "482"
## [8,] "Very Poor" "Very Poor" "Very Poor" "Very Poor" "Severe"  "Severe"
##      [,13]      [,14]      [,15]      [,16]      [,17]      [,18]
## [1,] "1/13/2019" "1/14/2019" "1/15/2019" "1/16/2019" "1/17/2019" "1/18/2019"
## [2,] "272.45"   "94.44"    "134.57"   "271.96"   "358.91"   "237.22"
## [3,] "403.32"   "173.94"   "229.1"    "424.07"   "519.49"   "364.61"
## [4,] "2.36"     "1.04"     "1.37"     "2.58"     "3.13"     "2.06"
## [5,] "17.83"    "15.05"    "17.09"    "20.35"    "19.63"    "14.7"
## [6,] "37.53"    "31.33"    "29.85"    "33.75"    "39.76"    "30.35"
## [7,] "464"      "320"      "248"      "383"      "515"      "437"
## [8,] "Severe"   "Very Poor" "Poor"    "Very Poor" "Severe"   "Severe"
##      [,19]      [,20]      [,21]      [,22]      [,23]      [,24]
## [1,] "1/19/2019" "1/20/2019" "1/21/2019" "1/22/2019" "1/23/2019" "1/24/2019"
## [2,] "262.68"   "314.59"   "175.49"   "53.25"    "135.68"   "130.59"
## [3,] "402.21"   "478.43"   "289.4"    "72.61"    "211.71"   "218.93"
## [4,] "2.35"     "3.66"     "1.83"     "1.01"     "1.62"     "1.44"
## [5,] "18.07"    "20.98"    "16.82"    "11.87"    "14.17"    "15.29"
## [6,] "34.75"    "45.03"    "32.77"    "27.01"    "26.51"    "30.57"
## [7,] "411"      "480"      "413"      "174"      "186"      "328"
## [8,] "Severe"   "Severe"   "Severe"   "Moderate" "Moderate" "Very Poor"
##      [,25]      [,26]      [,27]      [,28]      [,29]      [,30]
## [1,] "1/25/2019" "1/26/2019" "1/27/2019" "1/28/2019" "1/29/2019" "1/30/2019"
## [2,] "81.2"     "101.61"   "136.47"   "126.48"   "142"      "206.14"
## [3,] "136.43"   "169.32"   "217.45"   "212.42"   "235.85"   "334.21"
## [4,] "1.01"     "0.98"     "1.1"      "1.06"     "1.24"     "1.76"
## [5,] "14.62"    "14.48"    "15.89"    "15.76"    "18.56"    "24.75"
## [6,] "30.01"    "35.41"    "33.34"    "33.6"     "33.83"    "35.84"
## [7,] "188"      "231"      "275"      "301"      "297"      "342"
## [8,] "Moderate" "Poor"     "Poor"     "Very Poor" "Poor"    "Very Poor"
##      [,31]      [,32]      [,33]      [,34]      [,35]      [,36]
## [1,] "1/31/2019" "2/1/2019" "2/2/2019" "2/3/2019" "2/4/2019" "2/5/2019"
## [2,] "204.25"   "178.68"   "177.32"   "131.1"    "144.62"   "256.33"
## [3,] "335.07"   "265.65"   "266.3"    "209.41"   "248.49"   "408.32"
```

```
## [4,] "1.9"       "1.34"      "1.24"      "1.32"      "1.81"      "2.95"
## [5,] "20.48"     "17.83"     "11.55"     "11.7"      "14.06"     "22.38"
## [6,] "37.52"     "33.12"     "30.88"     "29.44"     "33.42"     "46.33"
## [7,] "397"       "329"       "338"       "323"       "301"       "398"
## [8,] "Very Poor" "Very Poor" "Very Poor" "Very Poor" "Very Poor" "Very Poor"
##      [,37]       [,38]       [,39]       [,40]       [,41]       [,42]
## [1,] "2/6/2019"  "2/7/2019"  "2/8/2019"  "2/9/2019"  "2/10/2019" "2/11/2019"
## [2,] "177.25"    "65.65"     "72.29"     "82.5"      "143.31"    "173.61"
## [3,] "270.92"    "107.83"    "156.6"     "165.94"    "255.47"    "304.98"
## [4,] "1.64"      "0.84"      "0.94"      "0.96"      "1.36"      "1.78"
## [5,] "15.59"     "11.12"     "12.66"     "13.25"     "16.41"     "17.52"
```

```
#making a variable to store all the data in pollution in a single conca
tenated form using (c) command.
all.poll=c(pollution$Date,pollution$PM2.5,pollution$PM10,pollution$CO,p
ollution$SO2,pollution$O3,pollution$AQI,pollution$AQI_Bucket)
#making a matrix using the concatenated data with matrix commad and mak
ing 8 number of columns.
pollution_mat3=matrix(all.poll,ncol=8,dimnames = list(c(1:547),give.nam
es))
#here dimnames is used to give names to the rows and columns of matrix
pollution_mat3
```

```
##     date          PM2.5    PM10     CO     SO2     03      AQI   AQI_TYPE

## 1   "1/1/2019"   "287.34" "461.02" "2.54" "17.77" "39.58" "475" "Severe"

## 2   "1/2/2019"   "331.2"  "515.72" "2.78" "18.27" "43.3"  "501" "Severe"

## 3   "1/3/2019"   "355.4"  "519.34" "2.55" "15.12" "39.77" "537" "Severe"

## 4   "1/4/2019"   "246.46" "388.79" "2.01" "13.22" "28.75" "432" "Severe"

## 5   "1/5/2019"   "262.53" "411.49" "2.23" "15.59" "36.33" "440" "Severe"

## 6   "1/6/2019"   "174.26" "254.66" "1.49" "13.65" "31.48" "371" "Very Poor"

## 7   "1/7/2019"   "181.74" "283.88" "1.57" "15.49" "27.11" "331" "Very Poor"

## 8   "1/8/2019"   "156.26" "242.02" "1.31" "14.67" "27.37" "340" "Very Poor"

## 9   "1/9/2019"   "143.66" "238.94" "1.3"  "15.09" "30.9"  "321" "Very Poor"

## 10  "1/10/2019"  "165.95" "278.87" "1.75" "16.81" "30.92" "317" "Very Poor"

## 11  "1/11/2019"  "278.95" "432.2"  "2.9"  "20.03" "35.61" "401" "Severe"

## 12  "1/12/2019"  "311.03" "465.89" "3.11" "17.81" "39.78" "482" "Severe"

## 13  "1/13/2019"  "272.45" "403.32" "2.36" "17.83" "37.53" "464" "Severe"

## 14  "1/14/2019"  "94.44"  "173.94" "1.04" "15.05" "31.33" "320" "Very Poor"

## 15  "1/15/2019"  "134.57" "229.1"  "1.37" "17.09" "29.85" "248" "Poor"

## 16  "1/16/2019"  "271.96" "424.07" "2.58" "20.35" "33.75" "383" "Very Poor"
```

```
## 17  "1/17/2019"  "358.91" "519.49" "3.13" "19.63" "39.76" "515" "Severe"

## 18  "1/18/2019"  "237.22" "364.61" "2.06" "14.7"  "30.35" "437" "Severe"

## 19  "1/19/2019"  "262.68" "402.21" "2.35" "18.07" "34.75" "411" "Severe"

## 20  "1/20/2019"  "314.59" "478.43" "3.66" "20.98" "45.03" "480" "Severe"
   … Total 547 Rows
#Here is the data of as a matrix format of PM2.5 and PM10 with their re
spective AQI_bucket on different dates from year 2019
pollution_md1=cbind(pollution$Date,pollution$PM2.5,pollution$PM10,
pollution$AQI_Bucket)
pollution_md1

##        [,1]         [,2]     [,3]     [,4]
##   [1,] "1/1/2019"   "287.34" "461.02" "Severe"
##   [2,] "1/2/2019"   "331.2"  "515.72" "Severe"
##   [3,] "1/3/2019"   "355.4"  "519.34" "Severe"
##   [4,] "1/4/2019"   "246.46" "388.79" "Severe"
##   [5,] "1/5/2019"   "262.53" "411.49" "Severe"
##   [6,] "1/6/2019"   "174.26" "254.66" "Very Poor"
##   [7,] "1/7/2019"   "181.74" "283.88" "Very Poor"
##   [8,] "1/8/2019"   "156.26" "242.02" "Very Poor"
##   [9,] "1/9/2019"   "143.66" "238.94" "Very Poor"
##  [10,] "1/10/2019"  "165.95" "278.87" "Very Poor"
##  [11,] "1/11/2019"  "278.95" "432.2"  "Severe"
##  [12,] "1/12/2019"  "311.03" "465.89" "Severe"
##  [13,] "1/13/2019"  "272.45" "403.32" "Severe"
##  [14,] "1/14/2019"  "94.44"  "173.94" "Very Poor"
##  [15,] "1/15/2019"  "134.57" "229.1"  "Poor"
##  [16,] "1/16/2019"  "271.96" "424.07" "Very Poor"
##  [17,] "1/17/2019"  "358.91" "519.49" "Severe"
##  [18,] "1/18/2019"  "237.22" "364.61" "Severe"
##  [19,] "1/19/2019"  "262.68" "402.21" "Severe"
##  [20,] "1/20/2019"  "314.59" "478.43" "Severe"
##  [21,] "1/21/2019"  "175.49" "289.4"  "Severe"
##  [22,] "1/22/2019"  "53.25"  "72.61"  "Moderate"
##  [23,] "1/23/2019"  "135.68" "211.71" "Moderate"
##  [24,] "1/24/2019"  "130.59" "218.93" "Very Poor"
##  [25,] "1/25/2019"  "81.2"   "136.43" "Moderate"
##  [26,] "1/26/2019"  "101.61" "169.32" "Poor"
##  [27,] "1/27/2019"  "136.47" "217.45" "Poor"
##  [28,] "1/28/2019"  "126.48" "212.42" "Very Poor"
##  [29,] "1/29/2019"  "142"    "235.85" "Poor"
##  [30,] "1/30/2019"  "206.14" "334.21" "Very Poor"
##  [31,] "1/31/2019"  "204.25" "335.07" "Very Poor"
##  [32,] "2/1/2019"   "178.68" "265.65" "Very Poor"
##  [33,] "2/2/2019"   "177.32" "266.3"  "Very Poor"
##  [34,] "2/3/2019"   "131.1"  "209.41" "Very Poor"
##  [35,] "2/4/2019"   "144.62" "248.49" "Very Poor"
```

```
##  [36,] "2/5/2019"   "256.33" "408.32" "Very Poor"
##  [37,] "2/6/2019"   "177.25" "270.92" "Very Poor"
##  [38,] "2/7/2019"   "65.65"  "107.83" "Poor"
##  [39,] "2/8/2019"   "72.29"  "156.6"  "Moderate"
##  [40,] "2/9/2019"   "82.5"   "165.94" "Moderate"
##  [41,] "2/10/2019"  "143.31" "255.47" "Poor"
##  [42,] "2/11/2019"  "173.61" "304.98" "Very Poor"
##  [43,] "2/12/2019"  "208.64" "357.36" "Very Poor"
```

MATRIX TO DATA FRAME

NOW AGAIN CONVERTING THAT DATA IN DATA FRAME

We'll consider the same example As we all know, NEW DELHI saw a huge improvement in the air quality index in 2020 due the lockdown after march. thus the covid situation helped the people of delhi to live in good quality air which was one of the positive impact of the covid -19 in our lives. Here is representation of air quality index in year 2020 which is represented in data frame you can compare aqi of janurary and february to that of later.

```
date=pollution$Date[366:547]
aqi=pollution$AQI[366:547]
aqibucket=pollution$AQI_Bucket[366:547]
date
```

```
##   [1] "1/1/2020"  "1/2/2020"  "1/3/2020"  "1/4/2020"  "1/5/2020"  "1/6/
2020"
##   [7] "1/7/2020"  "1/8/2020"  "1/9/2020"  "1/10/2020" "1/11/2020" "1/12
/2020"
##  [13] "1/13/2020" "1/14/2020" "1/15/2020" "1/16/2020" "1/17/2020" "1/18
/2020"
##  [19] "1/19/2020" "1/20/2020" "1/21/2020" "1/22/2020" "1/23/2020" "1/24
/2020"
##  [25] "1/25/2020" "1/26/2020" "1/27/2020" "1/28/2020" "1/29/2020" "1/30
/2020"
##  [31] "1/31/2020" "2/1/2020"  "2/2/2020"  "2/3/2020"  "2/4/2020"  "2/5/
2020"
##  [37] "2/6/2020"  "2/7/2020"  "2/8/2020"  "2/9/2020"  "2/10/2020" "2/11
/2020"
##  [43] "2/12/2020" "2/13/2020" "2/14/2020" "2/15/2020" "2/16/2020" "2/17
/2020"
##  [49] "2/18/2020" "2/19/2020" "2/20/2020" "2/21/2020" "2/22/2020" "2/23
/2020"
##  [55] "2/24/2020" "2/25/2020" "2/26/2020" "2/27/2020" "2/28/2020" "2/29
/2020"
##  [61] "3/1/2020"  "3/2/2020"  "3/3/2020"  "3/4/2020"  "3/5/2020"  "3/6/
2020"
```

```
##  [67] "3/7/2020"  "3/8/2020"  "3/9/2020"  "3/10/2020" "3/11/2020" "3/12
/2020"
##  [73] "3/13/2020" "3/14/2020" "3/15/2020" "3/16/2020" "3/17/2020" "3/18
/2020"
##  [79] "3/19/2020" "3/20/2020" "3/21/2020" "3/22/2020" "3/23/2020" "3/24
/2020"
##  [85] "3/25/2020" "3/26/2020" "3/27/2020" "3/28/2020" "3/29/2020" "3/30
/2020"
##  [91] "3/31/2020" "4/1/2020"  "4/2/2020"  "4/3/2020"  "4/4/2020"  "4/5/
2020"
##  [97] "4/6/2020"  "4/7/2020"  "4/8/2020"  "4/9/2020"  "4/10/2020" "4/11
/2020"
## [103] "4/12/2020" "4/13/2020" "4/14/2020" "4/15/2020" "4/16/2020" "4/17
/2020"
## [109] "4/18/2020" "4/19/2020" "4/20/2020" "4/21/2020" "4/22/2020" "4/23
/2020"
## [115] "4/24/2020" "4/25/2020" "4/26/2020" "4/27/2020" "4/28/2020" "4/29
/2020"
## [121] "4/30/2020" "5/1/2020"  "5/2/2020"  "5/3/2020"  "5/4/2020"  "5/5/
2020"
## [127] "5/6/2020"  "5/7/2020"  "5/8/2020"  "5/9/2020"  "5/10/2020" "5/11
/2020"
## [133] "5/12/2020" "5/13/2020" "5/14/2020" "5/15/2020" "5/16/2020" "5/17
/2020"
## [139] "5/18/2020" "5/19/2020" "5/20/2020" "5/21/2020" "5/22/2020" "5/23
/2020"
## [145] "5/24/2020" "5/25/2020" "5/26/2020" "5/27/2020" "5/28/2020" "5/29
/2020"
## [151] "5/30/2020" "5/31/2020" "6/1/2020"  "6/2/2020"  "6/3/2020"  "6/4/
2020"
## [157] "6/5/2020"  "6/6/2020"  "6/7/2020"  "6/8/2020"  "6/9/2020"  "6/10
/2020"
## [163] "6/11/2020" "6/12/2020" "6/13/2020" "6/14/2020" "6/15/2020" "6/16
/2020"
## [169] "6/17/2020" "6/18/2020" "6/19/2020" "6/20/2020" "6/21/2020" "6/22
/2020"
## [175] "6/23/2020" "6/24/2020" "6/25/2020" "6/26/2020" "6/27/2020" "6/28
/2020"
## [181] "6/29/2020" "6/30/2020"
```

```
pollution.frame1=data.frame(date,aqi,aqibucket)
pollution.frame1
```

```
##          date aqi    aqibucket
## 1    1/1/2020 492       Severe
## 2    1/2/2020 485       Severe
## 3    1/3/2020 426       Severe
## 4    1/4/2020 366    Very Poor
## 5    1/5/2020 354    Very Poor
## 6    1/6/2020 335    Very Poor
## 7    1/7/2020 285         Poor
## 8    1/8/2020 281         Poor
## 9    1/9/2020 231         Poor
## 10  1/10/2020 255         Poor
```

```
## 11  1/11/2020 309    Very Poor
## 12  1/12/2020 352    Very Poor
## 13  1/13/2020 381    Very Poor
## 14  1/14/2020 314    Very Poor
## 15  1/15/2020 241         Poor
## 16  1/16/2020 286         Poor
## 17  1/17/2020 287         Poor
## 18  1/18/2020 262         Poor
## 19  1/19/2020 290         Poor
## 20  1/20/2020 282         Poor
## 21  1/21/2020 359    Very Poor
## 22  1/22/2020 378    Very Poor
## 23  1/23/2020 273         Poor
## 24  1/24/2020 155     Moderate
## 25  1/25/2020 251         Poor
## 26  1/26/2020 349    Very Poor
## 27  1/27/2020 365    Very Poor
## 28  1/28/2020 302    Very Poor
## 29  1/29/2020 218         Poor
## 30  1/30/2020 195     Moderate
## 31  1/31/2020 203         Poor
## 32   2/1/2020 244         Poor
## 33   2/2/2020 254         Poor
## 34   2/3/2020 297         Poor
## 35   2/4/2020 336    Very Poor
## 36   2/5/2020 344    Very Poor
## 37   2/6/2020 277         Poor
## 38   2/7/2020 307    Very Poor
## 39   2/8/2020 315    Very Poor
## 40   2/9/2020 310    Very Poor
## 41  2/10/2020 328    Very Poor
## 42  2/11/2020 347    Very Poor
## 43  2/12/2020 343    Very Poor
## 94   4/3/2020  84 Satisfactory
## 95   4/4/2020  90 Satisfactory
## 96   4/5/2020 103     Moderate
## 97   4/6/2020 134     Moderate
## 98   4/7/2020 100 Satisfactory
## 99   4/8/2020  91 Satisfactory
## 100  4/9/2020  89 Satisfactory
## 101 4/10/2020 111     Moderate
## 102 4/11/2020 126     Moderate
## 103 4/12/2020 102     Moderate
## 104 4/13/2020 119     Moderate
## 105 4/14/2020 134     Moderate
## 106 4/15/2020 152     Moderate
## 107 4/16/2020 172     Moderate
## 108 4/17/2020 123     Moderate
## 109 4/18/2020 102     Moderate
## 110 4/19/2020  95 Satisfactory
## 111 4/20/2020 115     Moderate
## 112 4/21/2020 102     Moderate
## 113 4/22/2020 122     Moderate
## 114 4/23/2020 129     Moderate
## 115 4/24/2020 143     Moderate
## 116 4/25/2020 121     Moderate
## 117 4/26/2020 119     Moderate

… Total 547 Rows
```

DATA FRAME TO MATRIX

Here is example of converting data frame into matrix and vice versa using (as) command. also converting a data frame into list.

```
pollution.mat=as.matrix(pollution)
pollution.mat
```

```
##         Date          PM2.5    PM10     CO     SO2     O3      AQI
##   [1,] "1/1/2019"   "287.34" "461.02" "2.54" "17.77" "39.58" "475"
##   [2,] "1/2/2019"   "331.20" "515.72" "2.78" "18.27" "43.30" "501"
##   [3,] "1/3/2019"   "355.40" "519.34" "2.55" "15.12" "39.77" "537"
##   [4,] "1/4/2019"   "246.46" "388.79" "2.01" "13.22" "28.75" "432"
##   [5,] "1/5/2019"   "262.53" "411.49" "2.23" "15.59" "36.33" "440"
##   [6,] "1/6/2019"   "174.26" "254.66" "1.49" "13.65" "31.48" "371"
##   [7,] "1/7/2019"   "181.74" "283.88" "1.57" "15.49" "27.11" "331"
##   [8,] "1/8/2019"   "156.26" "242.02" "1.31" "14.67" "27.37" "340"
##   [9,] "1/9/2019"   "143.66" "238.94" "1.30" "15.09" "30.90" "321"
##  [10,] "1/10/2019"  "165.95" "278.87" "1.75" "16.81" "30.92" "317"
##  [11,] "1/11/2019"  "278.95" "432.20" "2.90" "20.03" "35.61" "401"
##  [12,] "1/12/2019"  "311.03" "465.89" "3.11" "17.81" "39.78" "482"
##  [13,] "1/13/2019"  "272.45" "403.32" "2.36" "17.83" "37.53" "464"
##  [14,] "1/14/2019"  " 94.44" "173.94" "1.04" "15.05" "31.33" "320"
##  [15,] "1/15/2019"  "134.57" "229.10" "1.37" "17.09" "29.85" "248"
##  [16,] "1/16/2019"  "271.96" "424.07" "2.58" "20.35" "33.75" "383"
##  [17,] "1/17/2019"  "358.91" "519.49" "3.13" "19.63" "39.76" "515"
##  [18,] "1/18/2019"  "237.22" "364.61" "2.06" "14.70" "30.35" "437"
##  [19,] "1/19/2019"  "262.68" "402.21" "2.35" "18.07" "34.75" "411"
##  [20,] "1/20/2019"  "314.59" "478.43" "3.66" "20.98" "45.03" "480"
##  [21,] "1/21/2019"  "175.49" "289.40" "1.83" "16.82" "32.77" "413"
##  [22,] "1/22/2019"  " 53.25" " 72.61" "1.01" "11.87" "27.01" "174"
##  [23,] "1/23/2019"  "135.68" "211.71" "1.62" "14.17" "26.51" "186"
##  [24,] "1/24/2019"  "130.59" "218.93" "1.44" "15.29" "30.57" "328"
##  [25,] "1/25/2019"  " 81.20" "136.43" "1.01" "14.62" "30.01" "188"
##  [26,] "1/26/2019"  "101.61" "169.32" "0.98" "14.48" "35.41" "231"
##  [27,] "1/27/2019"  "136.47" "217.45" "1.10" "15.89" "33.34" "275"
##  [28,] "1/28/2019"  "126.48" "212.42" "1.06" "15.76" "33.60" "301"
##  [29,] "1/29/2019"  "142.00" "235.85" "1.24" "18.56" "33.83" "297"
##  [30,] "1/30/2019"  "206.14" "334.21" "1.76" "24.75" "35.84" "342"
##  [31,] "1/31/2019"  "204.25" "335.07" "1.90" "20.48" "37.52" "397"
##  [32,] "2/1/2019"   "178.68" "265.65" "1.34" "17.83" "33.12" "329"
##  [33,] "2/2/2019"   "177.32" "266.30" "1.24" "11.55" "30.88" "338"
##  [34,] "2/3/2019"   "131.10" "209.41" "1.32" "11.70" "29.44" "323"
##  [35,] "2/4/2019"   "144.62" "248.49" "1.81" "14.06" "33.42" "301"
##  [36,] "2/5/2019"   "256.33" "408.32" "2.95" "22.38" "46.33" "398"
##  [37,] "2/6/2019"   "177.25" "270.92" "1.64" "15.59" "34.40" "380"
##  [38,] "2/7/2019"   " 65.65" "107.83" "0.84" "11.12" "30.25" "233"
##  [39,] "2/8/2019"   " 72.29" "156.60" "0.94" "12.66" "30.59" "136"
```

```
##  [40,] "2/9/2019"   " 82.50" "165.94" "0.96" "13.25" "33.55" "169"
##  [41,] "2/10/2019"  "143.31" "255.47" "1.36" "16.41" "33.03" "277"
```

#Resulting matrix have columns as that of the given data frame.

MATRIX TO DATA FRAME

```
pollution.frame2=as.data.frame(pollution_mat1)
pollution.frame2
```

```
##             V1     V2     V3   V4    V5    V6  V7           V8
## 1     1/1/2019 287.34 461.02 2.54 17.77 39.58 475       Severe
## 2     1/2/2019  331.2 515.72 2.78 18.27  43.3 501       Severe
## 3     1/3/2019  355.4 519.34 2.55 15.12 39.77 537       Severe
## 4     1/4/2019 246.46 388.79 2.01 13.22 28.75 432       Severe
## 5     1/5/2019 262.53 411.49 2.23 15.59 36.33 440       Severe
## 6     1/6/2019 174.26 254.66 1.49 13.65 31.48 371    Very Poor
## 7     1/7/2019 181.74 283.88 1.57 15.49 27.11 331    Very Poor
## 8     1/8/2019 156.26 242.02 1.31 14.67 27.37 340    Very Poor
## 9     1/9/2019 143.66 238.94  1.3 15.09  30.9 321    Very Poor
## 10   1/10/2019 165.95 278.87 1.75 16.81 30.92 317    Very Poor
## 11   1/11/2019 278.95  432.2  2.9 20.03 35.61 401       Severe
## 12   1/12/2019 311.03 465.89 3.11 17.81 39.78 482       Severe
## 13   1/13/2019 272.45 403.32 2.36 17.83 37.53 464       Severe
## 14   1/14/2019  94.44 173.94 1.04 15.05 31.33 320    Very Poor
## 15   1/15/2019 134.57  229.1 1.37 17.09 29.85 248         Poor
## 16   1/16/2019 271.96 424.07 2.58 20.35 33.75 383    Very Poor
## 17   1/17/2019 358.91 519.49 3.13 19.63 39.76 515       Severe
## 18   1/18/2019 237.22 364.61 2.06  14.7 30.35 437       Severe
## 19   1/19/2019 262.68 402.21 2.35 18.07 34.75 411       Severe
## 20   1/20/2019 314.59 478.43 3.66 20.98 45.03 480       Severe
## 21   1/21/2019 175.49  289.4 1.83 16.82 32.77 413       Severe
## 22   1/22/2019  53.25  72.61 1.01 11.87 27.01 174     Moderate
## 23   1/23/2019 135.68 211.71 1.62 14.17 26.51 186     Moderate
## 24   1/24/2019 130.59 218.93 1.44 15.29 30.57 328    Very Poor
## 25   1/25/2019   81.2 136.43 1.01 14.62 30.01 188     Moderate
## 26   1/26/2019 101.61 169.32 0.98 14.48 35.41 231         Poor
## 27   1/27/2019 136.47 217.45  1.1 15.89 33.34 275         Poor
## 28   1/28/2019 126.48 212.42 1.06 15.76  33.6 301    Very Poor
## 29   1/29/2019    142 235.85 1.24 18.56 33.83 297         Poor
## 30   1/30/2019 206.14 334.21 1.76 24.75 35.84 342    Very Poor
## 31   1/31/2019 204.25 335.07  1.9 20.48 37.52 397    Very Poor
## 32    2/1/2019 178.68 265.65 1.34 17.83 33.12 329    Very Poor
## 33    2/2/2019 177.32  266.3 1.24 11.55 30.88 338    Very Poor
## 34    2/3/2019  131.1 209.41 1.32  11.7 29.44 323    Very Poor
## 35    2/4/2019 144.62 248.49 1.81 14.06 33.42 301    Very Poor
## 36    2/5/2019 256.33 408.32 2.95 22.38 46.33 398    Very Poor
## 37    2/6/2019 177.25 270.92 1.64 15.59  34.4 380    Very Poor
```

```
## 38    2/7/2019  65.65 107.83 0.84 11.12 30.25 233         Poor
## 39    2/8/2019  72.29  156.6 0.94 12.66 30.59 136     Moderate
## 40    2/9/2019   82.5 165.94 0.96 13.25 33.55 169     Moderate
## 41   2/10/2019 143.31 255.47 1.36 16.41 33.03 277         Poor
## 42   2/11/2019 173.61 304.98 1.78 17.52 37.03 332    Very Poor
## 43   2/12/2019 208.64 357.36 2.09 19.93 43.29 357    Very Poor
#the resulting data frame contains the data of columns matrix as variab
les instead of character in matrix.
```

DATA FRAME TO LIST

```
pollution.list1=as.list(pollution)
pollution.list1
```

```
## $Date
##   [1] "1/1/2019"   "1/2/2019"   "1/3/2019"   "1/4/2019"   "1/5/2019"
##   [6] "1/6/2019"   "1/7/2019"   "1/8/2019"   "1/9/2019"   "1/10/2019"
##  [11] "1/11/2019"  "1/12/2019"  "1/13/2019"  "1/14/2019"  "1/15/2019"
##  [16] "1/16/2019"  "1/17/2019"  "1/18/2019"  "1/19/2019"  "1/20/2019"
##  [21] "1/21/2019"  "1/22/2019"  "1/23/2019"  "1/24/2019"  "1/25/2019"
##  [26] "1/26/2019"  "1/27/2019"  "1/28/2019"  "1/29/2019"  "1/30/2019"
##  [31] "1/31/2019"  "2/1/2019"   "2/2/2019"   "2/3/2019"   "2/4/2019"
##  [36] "2/5/2019"   "2/6/2019"   "2/7/2019"   "2/8/2019"   "2/9/2019"
##  [41] "2/10/2019"  "2/11/2019"  "2/12/2019"  "2/13/2019"  "2/14/2019"
##  [46] "2/15/2019"  "2/16/2019"  "2/17/2019"  "2/18/2019"  "2/19/2019"
##  [51] "2/20/2019"  "2/21/2019"  "2/22/2019"  "2/23/2019"  "2/24/2019"
##  [56] "2/25/2019"  "2/26/2019"  "2/27/2019"  "2/28/2019"  "3/1/2019"
##  [61] "3/2/2019"   "3/3/2019"   "3/4/2019"   "3/5/2019"   "3/6/2019"
##  [66] "3/7/2019"   "3/8/2019"   "3/9/2019"   "3/10/2019"  "3/11/2019"
##  [71] "3/12/2019"  "3/13/2019"  "3/14/2019"  "3/15/2019"  "3/16/2019"
##  [76] "3/17/2019"  "3/18/2019"  "3/19/2019"  "3/20/2019"  "3/21/2019"
##  [81] "3/22/2019"  "3/23/2019"  "3/24/2019"  "3/25/2019"  "3/26/2019"
##  [86] "3/27/2019"  "3/28/2019"  "3/29/2019"  "3/30/2019"  "3/31/2019"
##  [91] "4/1/2019"   "4/2/2019"   "4/3/2019"   "4/4/2019"   "4/5/2019"
##  [96] "4/6/2019"   "4/7/2019"   "4/8/2019"   "4/9/2019"   "4/10/2019"
## [101] "4/11/2019"  "4/12/2019"  "4/13/2019"  "4/14/2019"  "4/15/2019"
## [106] "4/16/2019"  "4/17/2019"  "4/18/2019"  "4/19/2019"  "4/20/2019"
## [111] "4/21/2019"  "4/22/2019"  "4/23/2019"  "4/24/2019"  "4/25/2019"
## [116] "4/26/2019"  "4/27/2019"  "4/28/2019"  "4/29/2019"  "4/30/2019"
## [121] "5/1/2019"   "5/2/2019"   "5/3/2019"   "5/4/2019"   "5/5/2019"
## [126] "5/6/2019"   "5/7/2019"   "5/8/2019"   "5/9/2019"   "5/10/2019"
## [131] "5/11/2019"  "5/12/2019"  "5/13/2019"  "5/14/2019"  "5/15/2019"
## [136] "5/16/2019"  "5/17/2019"  "5/18/2019"  "5/19/2019"  "5/20/2019"
## [141] "5/21/2019"  "5/22/2019"  "5/23/2019"  "5/24/2019"  "5/25/2019"
## [146] "5/26/2019"  "5/27/2019"  "5/28/2019"  "5/29/2019"  "5/30/2019"
## [151] "5/31/2019"  "6/1/2019"   "6/2/2019"   "6/3/2019"   "6/4/2019"
## [156] "6/5/2019"   "6/6/2019"   "6/7/2019"   "6/8/2019"   "6/9/2019"
## [161] "6/10/2019"  "6/11/2019"  "6/12/2019"  "6/13/2019"  "6/14/2019"
## [166] "6/15/2019"  "6/16/2019"  "6/17/2019"  "6/18/2019"  "6/19/2019"
## [171] "6/20/2019"  "6/21/2019"  "6/22/2019"  "6/23/2019"  "6/24/2019"
## [176] "6/25/2019"  "6/26/2019"  "6/27/2019"  "6/28/2019"  "6/29/2019"
## [181] "6/30/2019"  "7/1/2019"   "7/2/2019"   "7/3/2019"   "7/4/2019"
## [186] "7/5/2019"   "7/6/2019"   "7/7/2019"   "7/8/2019"   "7/9/2019"
## [191] "7/10/2019"  "7/11/2019"  "7/12/2019"  "7/13/2019"  "7/14/2019"
## [196] "7/15/2019"  "7/16/2019"  "7/17/2019"  "7/18/2019"  "7/19/2019"
## [201] "7/20/2019"  "7/21/2019"  "7/22/2019"  "7/23/2019"  "7/24/2019"
## [206] "7/25/2019"  "7/26/2019"  "7/27/2019"  "7/28/2019"  "7/29/2019"
```

```
## [211] "7/30/2019"  "7/31/2019"  "8/1/2019"   "8/2/2019"   "8/3/2019"
## [216] "8/4/2019"   "8/5/2019"   "8/6/2019"   "8/7/2019"   "8/8/2019"
## [221] "8/9/2019"   "8/10/2019"  "8/11/2019"  "8/12/2019"  "8/13/2019"
## [226] "8/14/2019"  "8/15/2019"  "8/16/2019"  "8/17/2019"  "8/18/2019"
## [231] "8/19/2019"  "8/20/2019"  "8/21/2019"  "8/22/2019"  "8/23/2019"
## [236] "8/24/2019"  "8/25/2019"  "8/26/2019"  "8/27/2019"  "8/28/2019"
## [241] "8/29/2019"  "8/30/2019"  "8/31/2019"  "9/1/2019"   "9/2/2019"
## [246] "9/3/2019"   "9/4/2019"   "9/5/2019"   "9/6/2019"   "9/7/2019"
## [251] "9/8/2019"   "9/9/2019"   "9/10/2019"  "9/11/2019"  "9/12/2019"
## [256] "9/13/2019"  "9/14/2019"  "9/15/2019"  "9/16/2019"  "9/17/2019"
## [261] "9/18/2019"  "9/19/2019"  "9/20/2019"  "9/21/2019"  "9/22/2019"
##
## attr(,"spec")
## cols(
##   Date = col_character(),
##   PM2.5 = col_double(),
##   PM10 = col_double(),
##   CO = col_double(),
##   SO2 = col_double(),
##   O3 = col_double(),
##   AQI = col_double(),
##   AQI_Bucket = col_character()
## )

#the resulting list contains the column of data frame as elements
```

We use the pollution data for this activity, which you use to reorder and add to.

step 1:-Look at the data frame pollution and create an index using the values in the PM2.5column, with ties resolved by the PM10 column and look the index you just created

```
pollution_index = with(pollution, order(AQI,Date))
pollution_index
```

```
##   [1] 453 454 230 229 272 541 228 455 222 273 452 432 238 458 270 271 457 517
##  [19] 224 262 456 431 210 209 223 258 459 516 540 208 225 257 542 265 269 333
##  [37] 465 221 266 460 211 212 464 450 490 198 451 226 227 207 475 214 250 263
##  [55] 249 274 483 199 206 213 187 231 218 220 251 463 518 268 188 216 468 107
##  [73] 474 477 523 461 232 241 256 267 418 108 539 543 524 215 219 275 426 484
##  [91] 538 245 332 535 233 237 252 278 217 489 534 242 466 185 259  58 430 544
## [109] 525 264 277 497 186 547 239 476  54 491 197 261 175 248 469 482 169 240
## [127] 481 478 498 281 473 485 519 520 202 243 244 440 449 203 437 501 145 234
## [145]  83 467 533 168 522 279 448 496 488 436 486 479 515 189 532 235 439  82
## [163] 176 190 246 499 530 536 280 470 462 146  39  84 201 434 494 526 492 253
## [181] 276 487 493  81 495 528 200  75 480 260  57 425 521 531 441  71 507 236
## [199] 104 191 109 144 471 529  91 512 174 500 527 389 433 513 204  69 511  52
## [217] 537 255 177 443  85 184 105 147 442 282 205 247 331 254 142 165  72 139
## [235]  55 420  40 178 292 417 438  62 472  64 170  22 411 510 514 334 103 502
## [253] 148 171 138  86 351 172 348 196 124  74 159 173  23 410 435  25  65 164
## [271]  70 156  53 444 149 427 136 158  68  92 135 395 182 101 506 545  59  76
## [289] 447  90 110 508 429 350 102 503 509 123 183 396 349 446 167 106 141 179
## [307] 283  93 445 157 155 424 140  63 134 166 394  67 419  56  77  60 504 153
## [325] 116 180 111 428 150  61  66 143 161  26 374 125  50 114 154  38 546  79
## [343] 181 285 160 505  78  99 380 293 284 295 195 397 421 322  87 112  48  15
## [361] 296 117 390 323 115  89 416 294 412 398 375 119 335 113 120 137  94 383
## [379] 100 328 291 310 131 151 126 329 118  96 423  80 388 321 352  88  27  46
## [397] 122  41 402  49 409 163 373  73 385 286 330  95  98 372 287 121 381 382
```

```
## [415]  47 384 288 414 193 311  29 399 336 290 415 152 192  28  35 393 337 422
## [433]  51 403 376 405 127 194 338 379 299 324 404  97 162  10  14 289   9 313
## [451] 357 347 413  34  24 406  32   7 298  42 130 327 371 297 400  33 314   8
## [469] 360  30 408 401 312 407 391  45 356 377 132 370 300 353  43 386 392 369
## [487]   6 343 361  44 325 315 358 133 326 344 387  37 378 341  16 359 309 339
## [505] 128 129  31  36  11 342  19  21 301 365 302 340 362 368 345   4  18 320
## [523]   5 355 303 354 363 304  13 346 306   1 316  20  12 367 366   2 364  17
## [541] 317 308   3 318 305 319 307

#here, the pollution data is sorted by the increasing order of the AQI
but in case if the AQI is equal then it will sort according to date col
umn
```

step 2:-Now create a new data frame using the sort index you just made.

```
pollution.resort = pollution[pollution_index,]
pollution.resort
```

```
## # A tibble: 547 x 8
##    Date      PM2.5  PM10    CO   SO2    O3   AQI AQI_Bucket
##    <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
##  1 3/28/2020  21.2  34.2 0.54  10.3   25.1    51 Satisfactory
##  2 3/29/2020  27.6  55.8 0.52  10.4   35.8    64 Satisfactory
##  3 8/18/2019  15.5  30.4 0.98   8.57  18.0    66 Satisfactory
##  4 8/17/2019  10.2  19.5 0.81   7.67  16.5    67 Satisfactory
##  5 9/29/2019  16.9  51.0 0.74   8.85  22.8    68 Satisfactory
##  6 6/24/2020  30.9  62.6 0.89  11.0   24.8    71 Satisfactory
##  7 8/16/2019  15.5  37.1 0.87   8.61  23.6    73 Satisfactory
##  8 3/30/2020  35.1  64.7 0.580 12.5   40.3    74 Satisfactory
##  9 8/10/2019  16.7  47.8 0.87   8.47  25.4    75 Satisfactory
## 10 9/30/2019  29.4  82.0 0.92   8.22  26.7    75 Satisfactory
## # ... with 537 more rows

#here we have sorted the data according to air quality index and as obs
erved most of the satisfactory and moderate air quality index is in 202
0 and i.e.. because of lockdown.
```

step 3:-Select a different order for the columns by specifying them in the square brackets in a new order.

```
pollution.resort = pollution[pollution_index, c(2, 1)]
pollution.resort
```

```
## # A tibble: 547 x 2
##    PM2.5 Date
##    <dbl> <chr>
##  1  21.2 3/28/2020
##  2  27.6 3/29/2020
##  3  15.5 8/18/2019
##  4  10.2 8/17/2019
##  5  16.9 9/29/2019
##  6  30.9 6/24/2020
##  7  15.5 8/16/2019
##  8  35.1 3/30/2020
```

```
##  9  16.7 8/10/2019
## 10  29.4 9/30/2019
## # ... with 537 more rows
```

step 4:-Now create a new vector of values

```
p = c(1:547)
#here,we make a new column name p for adding to the original data.
```

step 5:-Finally, create a data frame that includes the original data plus the new vector you just created. Use the sort index from before.

```
pollution.resort = with(pollution, data.frame(AQI, Date, p)[pollution_index,])
pollution.resort
```

```
##     AQI       Date   p
## 453  51  3/28/2020 453
## 454  64  3/29/2020 454
## 230  66  8/18/2019 230
## 229  67  8/17/2019 229
## 272  68  9/29/2019 272
## 541  71  6/24/2020 541
## 228  73  8/16/2019 228
## 455  74  3/30/2020 455
## 222  75  8/10/2019 222
## 273  75  9/30/2019 273
## 452  76  3/27/2020 452
## 432  77   3/7/2020 432
## 238  77  8/26/2019 238
## 458  78   4/2/2020 458
## 270  78  9/27/2019 270
## 271  78  9/28/2019 271
## 457  80   4/1/2020 457
## 517  80  5/31/2020 517
## 224  80  8/12/2019 224
## 262  80  9/19/2019 262
## 456  81  3/31/2020 456
## 431  82   3/6/2020 431
## 210  82  7/29/2019 210
## 209  83  7/28/2019 209
## 223  83  8/11/2019 223
## 258  83  9/15/2019 258
## 459  84   4/3/2020 459
## 516  84  5/30/2020 516
## 540  85  6/23/2020 540
## 208  85  7/27/2019 208
## 225  85  8/13/2019 225
## 257  85  9/14/2019 257
## 542  86  6/25/2020 542
## 265  86  9/22/2019 265
## 269  87  9/26/2019 269
## 333  89 11/29/2019 333
## 465  89   4/9/2020 465
## 221  89   8/9/2019 221
## 266  89  9/23/2019 266
## 460  90   4/4/2020 460
## 211  90  7/30/2019 211
## 212  90  7/31/2019 212
```

~DPLYR~

Dplyr is a package that can be used to perform many functions in R. It process faster than base R functions. It is because dplyr functions were written in a computationally efficient manner. They are also more stable in the syntax and better supports data frames than vectors.

Some basic functions of dplyr are:

```
summarise(pol_delhi) #works same as summary() function in R
```

```
## # A tibble: 1 x 0
```

```
str(pol_delhi)
```

```
## tibble [548 x 10] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date      : chr [1:548] "1/1/2019" "1/2/2019" "1/3/2019" "1/4/2019" ...
##  $ PM2.5     : num [1:548] 287 331 355 246 263 ...
##  $ PM10      : num [1:548] 461 516 519 389 411 ...
##  $ NO        : num [1:548] 92 101.1 77.9 47.8 65 ...
##  $ NO2       : num [1:548] 69.5 75.1 66.2 57.5 60.4 ...
##  $ CO        : num [1:548] 2.54 2.78 2.55 2.01 2.23 1.49 1.57 1.31 1.3 1.75 ...
##  $ SO2       : num [1:548] 17.8 18.3 15.1 13.2 15.6 ...
##  $ O3        : num [1:548] 39.6 43.3 39.8 28.8 36.3 ...
##  $ AQI       : num [1:548] 475 501 537 432 440 371 331 340 321 317 ...
##  $ AQI_Bucket: chr [1:548] "Severe" "Severe" "Severe" "Severe" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   PM2.5 = col_double(),
##   ..   PM10 = col_double(),
##   ..   NO = col_double(),
##   ..   NO2 = col_double(),
##   ..   CO = col_double(),
##   ..   SO2 = col_double(),
##   ..   O3 = col_double(),
##   ..   AQI = col_double(),
##   ..   AQI_Bucket = col_character()
##   .. )
```

```
sample_n(pol_delhi,3)
```

#this function selects random rows from the data frame

```
## # A tibble: 3 x 10
##   Date      PM2.5  PM10    NO   NO2    CO   SO2    O3   AQI AQI_Bucket
##   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
## 1 3/5/2019   68.9  135. 18.8   40.0  1.01 14.3   36.3   173 Moderate
## 2 6/19/2020  57.6  206.  6.58  17.8  0.86 12.0   41.0   132 Moderate
## 3 9/17/2019  59.1  136. 14.1   38.2  1.12  8.64  41.2   143 Moderate
```

```
pol_delhi %>%
  filter(AQI_Bucket == "Poor")
```

```
#filter() command is used to display the subset data with matching logi
cal condition
```

```
## # A tibble: 134 x 10
##    Date      PM2.5  PM10    NO   NO2    CO   SO2    O3   AQI AQI_Bucket
##    <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
##  1 1/15/2019 135.   229.  36.3  52.4  1.37  17.1  29.8   248 Poor
##  2 1/26/2019 102.   169.  15.3  38.4  0.98  14.5  35.4   231 Poor
##  3 1/27/2019 136.   217.  19.8  45.3  1.1   15.9  33.3   275 Poor
##  4 1/29/2019 142    236.  33.5  51.5  1.24  18.6  33.8   297 Poor
##  5 2/7/2019   65.6  108.  12.0  30.8  0.84  11.1  30.2   233 Poor
##  6 2/10/2019 143.   255.  47.9  49.3  1.36  16.4  33.0   277 Poor
##  7 2/15/2019 123.   195.  25.1  44.3  1.22  15.4  28.2   275 Poor
##  8 2/16/2019 112.   182.  17.3  33.3  1.13  12.9  26.4   288 Poor
##  9 2/17/2019  99.3  181.  23.2  42.6  1.09  15.1  34.8   247 Poor
## 10 2/18/2019 109.   215.  33.0  56.1  1.18  16.8  33.7   278 Poor
## # ... with 124 more rows
```

```
pol_delhi %>%
select(Date, starts_with("PM"))
```

```
#select() function is used to display desired variables, like in the ex
ample we have taken another function start_with() so it will show the d
ata of the column whose column name have "PM" in it.
```

```
## # A tibble: 548 x 3
##    Date      PM2.5  PM10
##    <chr>     <dbl> <dbl>
##  1 1/1/2019   287.  461.
##  2 1/2/2019   331.  516.
##  3 1/3/2019   355.  519.
##  4 1/4/2019   246.  389.
##  5 1/5/2019   263.  411.
##  6 1/6/2019   174.  255.
##  7 1/7/2019   182.  284.
##  8 1/8/2019   156.  242.
##  9 1/9/2019   144.  239.
## 10 1/10/2019  166.  279.
## # ... with 538 more rows
```

```
pol_delhi %>%
mutate(Date, A = PM2.5/PM10) %>%
```

```
#add a new variable(columns) from already existing variables
  select(Date,PM2.5,PM10,A)
```

```
## # A tibble: 548 x 4
##    Date      PM2.5  PM10     A
##    <chr>     <dbl> <dbl> <dbl>
##  1 1/1/2019   287.  461. 0.623
##  2 1/2/2019   331.  516. 0.642
##  3 1/3/2019   355.  519. 0.684
```

```
##  4 1/4/2019   246.  389. 0.634
##  5 1/5/2019   263.  411. 0.638
##  6 1/6/2019   174.  255. 0.684
##  7 1/7/2019   182.  284. 0.640
##  8 1/8/2019   156.  242. 0.646
##  9 1/9/2019   144.  239. 0.601
## 10 1/10/2019  166.  279. 0.595
## # ... with 538 more rows
```

```
pol_delhi %>%
group_by(AQI_Bucket)%>%

summarise(
  n = n(),
  AQI = mean(AQI, na.rm = TRUE)
)%>%
  filter(n>1)

#group_by() function is basically used to do grouping task , like if we
 have some values repeating we can create a subset of group by giving s
ome logical condition. Doing this in basic R maybe a messy task.
```

```
## # A tibble: 5 x 3
##   AQI_Bucket       n   AQI
##   <chr>        <int> <dbl>
## 1 Moderate       226 144.
## 2 Poor           134 249.
## 3 Satisfactory    68  86.1
## 4 Severe          39 472.
## 5 Very Poor       81 344.
```

```
pol_delhi <- select(pol_delhi,-c(CO,SO2))
pol_delhi

# - (minus) sign is used to drop a variables/Column.
```

```
## # A tibble: 547 x 6
##    Date      PM2.5  PM10    O3   AQI AQI_Bucket
##    <chr>     <dbl> <dbl> <dbl> <dbl> <chr>
##  1 1/1/2019   287.  461.  39.6   475 Severe
##  2 1/2/2019   331.  516.  43.3   501 Severe
##  3 1/3/2019   355.  519.  39.8   537 Severe
##  4 1/4/2019   246.  389.  28.8   432 Severe
##  5 1/5/2019   263.  411.  36.3   440 Severe
##  6 1/6/2019   174.  255.  31.5   371 Very Poor
##  7 1/7/2019   182.  284.  27.1   331 Very Poor
##  8 1/8/2019   156.  242.  27.4   340 Very Poor
##  9 1/9/2019   144.  239.  30.9   321 Very Poor
## 10 1/10/2019  166.  279.  30.9   317 Very Poor
## # ... with 537 more rows
```

```r
pol_delhi <- select(pol_delhi,AQI,everything())
pol_delhi

#this will print selected Variables in the front and the reset after th
at.
```

```
## # A tibble: 547 x 6
##      AQI Date      PM2.5  PM10    O3 AQI_Bucket
##    <dbl> <chr>     <dbl> <dbl> <dbl> <chr>
##  1   475 1/1/2019   287.  461.  39.6 Severe
##  2   501 1/2/2019   331.  516.  43.3 Severe
##  3   537 1/3/2019   355.  519.  39.8 Severe
##  4   432 1/4/2019   246.  389.  28.8 Severe
##  5   440 1/5/2019   263.  411.  36.3 Severe
##  6   371 1/6/2019   174.  255.  31.5 Very Poor
##  7   331 1/7/2019   182.  284.  27.1 Very Poor
##  8   340 1/8/2019   156.  242.  27.4 Very Poor
##  9   321 1/9/2019   144.  239.  30.9 Very Poor
## 10   317 1/10/2019  166.  279.  30.9 Very Poor
## # ... with 537 more rows
```

```r
rename(pol_delhi,Air_Quality_Index = AQI)

#we can rename a Variable/Column by using rename() function
```

```
## # A tibble: 547 x 6
##    Air_Quality_Index Date      PM2.5  PM10    O3 AQI_Bucket
##                <dbl> <chr>     <dbl> <dbl> <dbl> <chr>
##  1               475 1/1/2019   287.  461.  39.6 Severe
##  2               501 1/2/2019   331.  516.  43.3 Severe
##  3               537 1/3/2019   355.  519.  39.8 Severe
##  4               432 1/4/2019   246.  389.  28.8 Severe
##  5               440 1/5/2019   263.  411.  36.3 Severe
##  6               371 1/6/2019   174.  255.  31.5 Very Poor
##  7               331 1/7/2019   182.  284.  27.1 Very Poor
##  8               340 1/8/2019   156.  242.  27.4 Very Poor
##  9               321 1/9/2019   144.  239.  30.9 Very Poor
## 10               317 1/10/2019  166.  279.  30.9 Very Poor
## # ... with 537 more rows
```

```r
summarize(pol_delhi,PM10_mean = mean(PM10),PM2.5 = median(PM2.5))

#here is a very interesting function that we can perform
#we can display the summary of any particular column/variable.
```

```
## # A tibble: 1 x 2
##   PM10_mean PM2.5
##       <dbl> <dbl>
## 1      196.  73.8
```

```r
pol_delhi%>%
  arrange(AQI,AQI_Bucket)%>%
  select(AQI,AQI_Bucket)
```

```
#this function is used to sort the data frame.
```

```
## # A tibble: 547 x 2
##      AQI AQI_Bucket
##    <dbl> <chr>
##  1    51 Satisfactory
##  2    64 Satisfactory
##  3    66 Satisfactory
##  4    67 Satisfactory
##  5    68 Satisfactory
##  6    71 Satisfactory
##  7    73 Satisfactory
##  8    74 Satisfactory
##  9    75 Satisfactory
## 10    75 Satisfactory
## # ... with 537 more rows
```

```
#arrange() function is similar to sort()
```

THANK YOU!!