Air Dataset

Lakshita

2025-11-10

Contents

1 Loading Libraries

```r
library(ggplot2)
library(maps)
library(ggrepel)
library(tidyverse)
library(dplyr)
library(tidyr)
library(corrplot)
library(gridExtra)
library(plotly)
library(factoextra)
library(psych)
library(GGally)
```

2 Loading data

```r
air=read.csv('indian_weather_data.csv')
head(air)
```

```
##        city    lat    lon temperature weather_code  sunrise   sunset moonrise
## 1 New Delhi 28.600 77.200          21          143 07:05 AM 05:26 PM 01:04 AM
## 2    Mumbai 18.975 72.826          30          122 07:03 AM 06:03 PM 01:20 AM
## 3   Kolkata 22.570 88.370          21          143 06:07 AM 04:54 PM 12:16 AM
## 4   Chennai 13.083 80.283          26          143 06:22 AM 05:44 PM 12:48 AM
## 5 Bengaluru 12.983 77.583          24          113 06:32 AM 05:55 PM 12:59 AM
## 6 Hyderabad 17.375 78.474          26          143 06:37 AM 05:44 PM 12:56 AM
##    moonset      co   no2  o3   so2  pm2_5   pm10 wind_speed wind_degree
## 1 01:06 PM 1411.85 23.95 264 76.65 137.25 140.05          4          34
## 2 01:29 PM  644.85 25.55 209 31.15  46.65  47.05         18         300
## 3 12:23 PM  457.85  1.95 214 12.95  44.55  47.25          8           3
## 4 01:00 PM  275.85  2.05 135  7.55  28.75  35.15         19          31
## 5 01:11 PM  243.85  3.85 152 10.75  20.95  26.35          9          76
## 6 01:06 PM  291.85  0.85 174 11.65  28.85  31.45         10          81
##   wind_dir pressure precip humidity cloudcover feelslike uv_index visibility
## 1       NE     1017      0       53         50        21        0          1
## 2      WNW     1011      0       35          0        32        0          4
## 3        N     1014      0       73          0        21        0          3
## 4      NNE     1012      0       65         25        28        0          5
## 5      ENE     1015      0       25          0        24        0         10
## 6        E     1016      0       32          0        26        0          4
```

```r
# making numerical df
num_df=air[c("lat","lon","temperature","weather_code","co","no2","o3","so2","pm2_5","pm10","wind_speed"
num_df=data.frame(num_df)

# making categorical df
cat_df=air[c("city","sunrise","sunset","moonrise","moonset","wind_dir")]
```

3 Data Handling

```
colSums(is.na(air))
```

```
##         city          lat          lon  temperature weather_code      sunrise
##            0            0            0            0            0            0
##       sunset     moonrise      moonset           co          no2           o3
##            0            0            0            0            0            0
##          so2        pm2_5         pm10   wind_speed  wind_degree     wind_dir
##            0            0            0            0            0            0
##     pressure       precip     humidity   cloudcover    feelslike     uv_index
##            0            0            0            0            0            0
##   visibility
##            0
```

4 PCA: Principal Component Analysis

```
pca_fit <- prcomp(num_df, scale=TRUE)
pca_summary<-summary(pca_fit)

importance_matrix<-pca_summary$importance

# Convert to data frame
pca_df <- as.data.frame(importance_matrix)
pca_df
```

```
##                             PC1      PC2      PC3      PC4      PC5       PC6
## Standard deviation     2.556806 1.743488 1.588943 1.216242 1.148401 0.9812938
## Proportion of Variance 0.363180 0.168870 0.140260 0.082180 0.073270 0.0535000
## Cumulative Proportion  0.363180 0.532060 0.672320 0.754500 0.827770 0.8812600
##                              PC7       PC8       PC9      PC10      PC11
## Standard deviation     0.8594431 0.6286084 0.5348331 0.4571326 0.3796192
## Proportion of Variance 0.0410400 0.0219500 0.0158900 0.0116100 0.0080100
## Cumulative Proportion  0.9223000 0.9442500 0.9601400 0.9717500 0.9797600
##                             PC12      PC13      PC14      PC15      PC16
## Standard deviation     0.3410012 0.3294213 0.2505385 0.1999257 0.1600532
## Proportion of Variance 0.0064600 0.0060300 0.0034900 0.0022200 0.0014200
## Cumulative Proportion  0.9862200 0.9922500 0.9957400 0.9979600 0.9993800
##                             PC17       PC18
## Standard deviation     0.1045533 0.01545629
## Proportion of Variance 0.0006100 0.00001000
## Cumulative Proportion  0.9999900 1.00000000
```

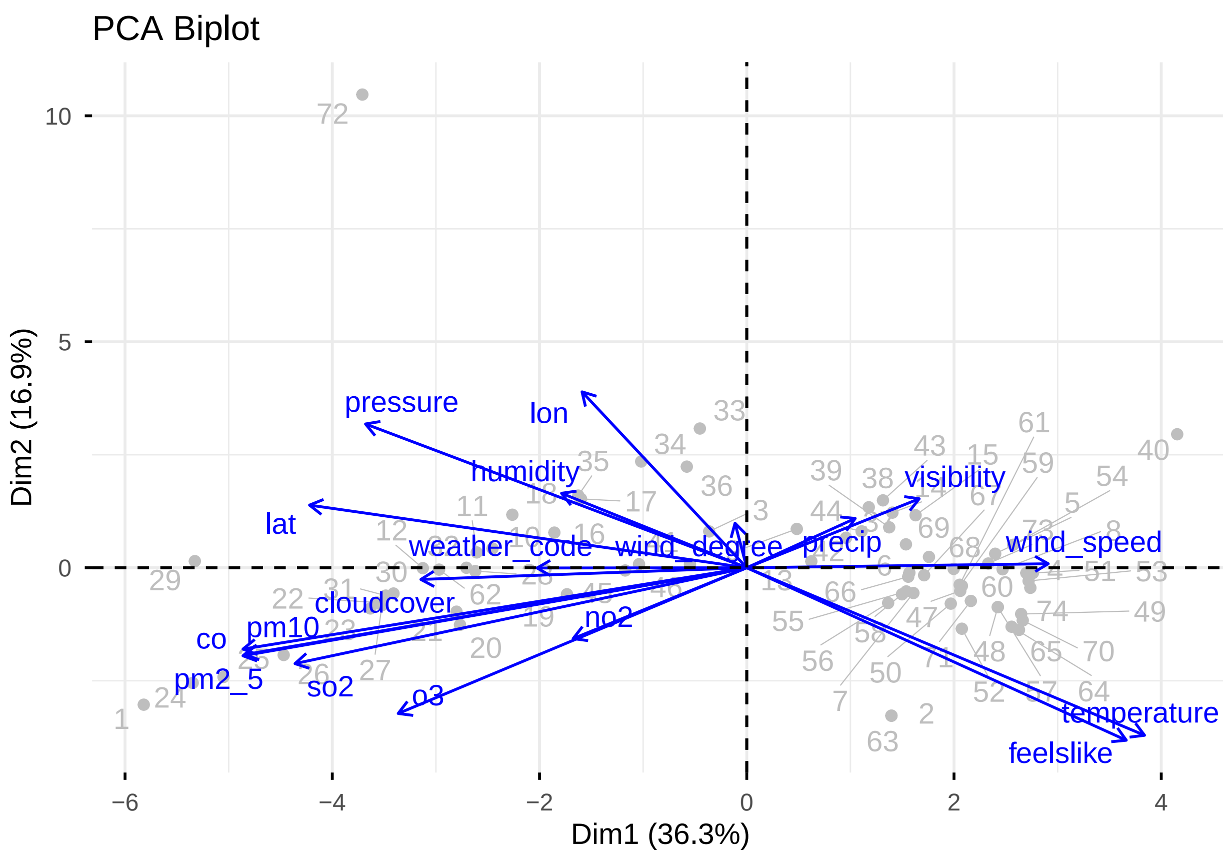
```
# Rotating the factors
rotation_df<-as.data.frame(pca_fit$rotation[,0:5])
rotation_df
```

```
##                         PC1           PC2         PC3         PC4         PC5
```

```
## lat          -0.301868872  0.1453971495  0.28720575 -0.04956144  0.05510171
## lon          -0.113729065  0.4083336324 -0.27112770 -0.02599798 -0.17747386
## temperature   0.274617258 -0.3888775279 -0.05857732  0.03761798 -0.01312017
## weather_code -0.144746718 -0.0006020218 -0.44531317  0.23438962  0.03755406
## co           -0.347793001 -0.1887966325 -0.04063375  0.02551994 -0.13137380
## no2          -0.119853939 -0.1644456273 -0.07546516 -0.54981440 -0.38058460
## o3           -0.240639055 -0.3383079733 -0.01290288  0.00254050  0.23094580
## so2          -0.311915100 -0.2225324396 -0.01515916 -0.16638917 -0.11444140
## pm2_5        -0.347825059 -0.2043726542 -0.03959826  0.04178356 -0.05548955
## pm10         -0.346411936 -0.2014739719 -0.04542474  0.04574484 -0.05018419
## wind_speed    0.208128297  0.0092724937 -0.17182039 -0.53401774 -0.21653369
## wind_degree  -0.008120785  0.1033177727 -0.03723924 -0.44244894  0.21845127
## pressure     -0.263220706  0.3337530033  0.08896988 -0.07950896  0.14769723
## precip        0.074879478  0.1144982689 -0.34734273  0.29757846 -0.49927881
## humidity     -0.127881700  0.1728986878 -0.49705138 -0.06621550  0.09813160
## cloudcover   -0.225001040 -0.0271496149  0.22535800  0.14860329 -0.43482550
## feelslike     0.262003201 -0.4004191405 -0.11003143  0.04748753 -0.06499603
## visibility    0.119028243  0.1601554371  0.40183188  0.03877666 -0.40139908
```

```r
fviz_pca_biplot(pca_fit,
                repel = TRUE,  # Prevents text overlapping
                col.var = "blue",     # Variables color
                col.ind = "gray",     # Individuals color
                title = "PCA Biplot")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



5 Factor Analysis

5.1 KMO and Bartlett's Test of Sphericity

```
# making a correlation matrix
correlation_matrix<-cor(num_df)

#KMO Test
kmo_result <- KMO(correlation_matrix)
print(kmo_result)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = correlation_matrix)
## Overall MSA =  0.69
## MSA for each item =
##          lat          lon  temperature weather_code           co          no2
##         0.85         0.54         0.72         0.65         0.83         0.35
##           o3          so2        pm2_5         pm10   wind_speed  wind_degree
##         0.82         0.78         0.75         0.74         0.56         0.34
##     pressure       precip     humidity   cloudcover    feelslike   visibility
##         0.79         0.35         0.56         0.81         0.69         0.49
```

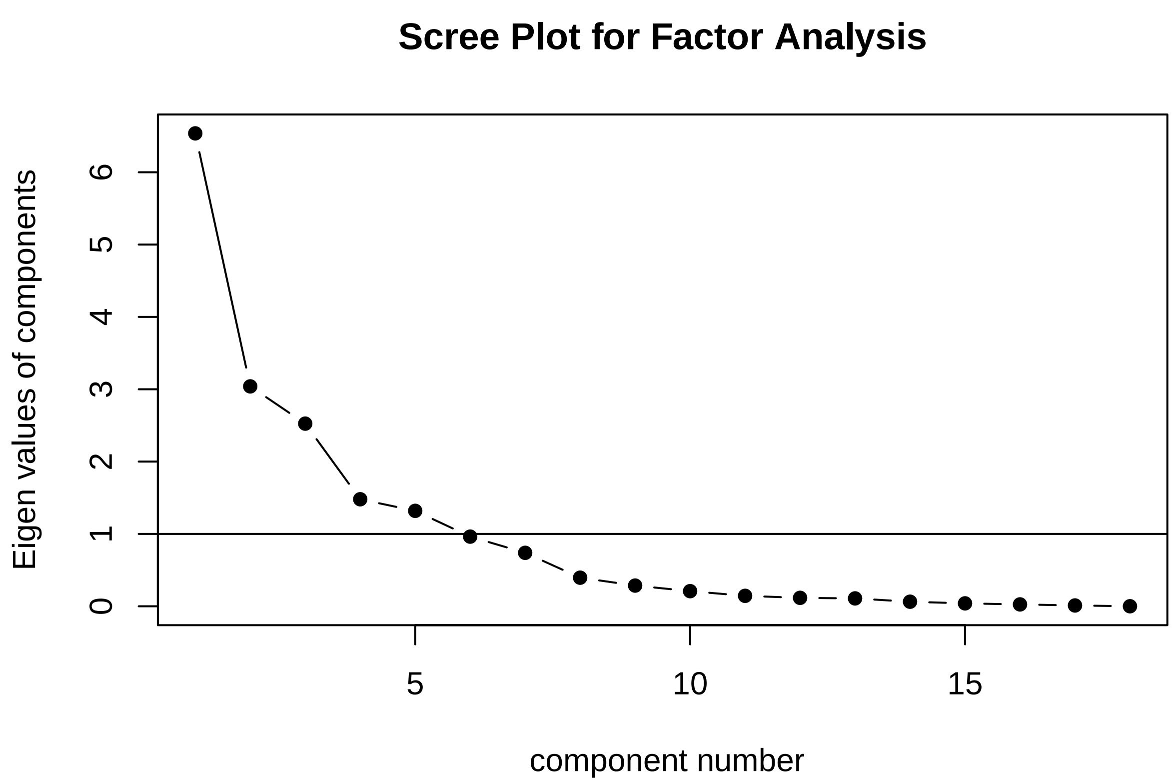
```r
# Bartlett's Test of Sphericity
cortest.bartlett(correlation_matrix, n = nrow(num_df))
```

```
## $chisq
## [1] 1873.867
##
## $p.value
## [1] 1.449223e-293
##
## $df
## [1] 153
```

1. **KMO Test :** we can observe the overall MSA value is greater than 0.69 indicating that correlation matrix is not identity matrix.
2. **Bartlett's Test of Sphericity :** We can Observe that the p-value is 1.449223e-293 <0.05 indicating that the data is suitable for factor analysis

5.2 Deciding number of factors

```r
scree(num_df, factors = FALSE, pc = TRUE,
      main = "Scree Plot for Factor Analysis")
```



```r
fa<- fa(num_df, nfactors=5,rotate="varimax",scores="regression")
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```r
fa
```

```
## Factor Analysis using method =  minres
## Call: fa(r = num_df, nfactors = 5, rotate = "varimax", scores = "regression")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                MR1   MR2   MR3   MR5   MR4    h2      u2 com
## lat           0.48  0.70 -0.21 -0.23 -0.12 0.840  0.1599 2.3
## lon          -0.10  0.61  0.32  0.46  0.16 0.720  0.2796 2.7
## temperature  -0.20 -0.97 -0.06 -0.08  0.03 0.985  0.0150 1.1
## weather_code  0.24  0.08  0.63  0.27 -0.10 0.543  0.4571 1.8
## co            0.93  0.21  0.15  0.02 -0.03 0.927  0.0731 1.2
## no2           0.45 -0.02 -0.03  0.00  0.66 0.632  0.3683 1.8
## o3            0.73 -0.10  0.23 -0.29 -0.13 0.693  0.3071 1.7
## so2           0.86  0.14  0.10 -0.09  0.16 0.810  0.1903 1.2
## pm2_5         0.94  0.18  0.18 -0.01 -0.09 0.950  0.0502 1.2
## pm10          0.93  0.18  0.19 -0.01 -0.10 0.936  0.0644 1.2
## wind_speed   -0.39 -0.27  0.00  0.09  0.78 0.833  0.1668 1.8
## wind_degree  -0.08  0.14  0.07 -0.08  0.20 0.074  0.9257 2.9
## pressure      0.19  0.87  0.10 -0.13 -0.01 0.812  0.1880 1.2
## precip       -0.12 -0.14  0.14  0.96 -0.07 0.986  0.0140 1.1
## humidity      0.05  0.30  0.76  0.30  0.17 0.785  0.2154 1.8
## cloudcover    0.56  0.28 -0.35  0.16 -0.12 0.561  0.4392 2.5
## feelslike    -0.14 -0.99 -0.01  0.01  0.04 1.005 -0.0049 1.0
## visibility   -0.24  0.10 -0.79  0.19 -0.04 0.739  0.2611 1.4
##
##                        MR1  MR2  MR3  MR5  MR4
## SS loadings           5.01 3.97 2.07 1.55 1.23
## Proportion Var        0.28 0.22 0.11 0.09 0.07
## Cumulative Var        0.28 0.50 0.61 0.70 0.77
## Proportion Explained  0.36 0.29 0.15 0.11 0.09
## Cumulative Proportion 0.36 0.65 0.80 0.91 1.00
##
## Mean item complexity =  1.6
## Test of the hypothesis that 5 factors are sufficient.
##
## df null model =  153  with the objective function =  28.32 with Chi Square =  1873.87
## df of  the model are 73  and the objective function was  10.23
##
## The root mean square of the residuals (RMSR) is  0.04
## The df corrected root mean square of the residuals is  0.06
##
## The harmonic n.obs is  74 with the empirical chi square  38.28  with prob <  1
## The total n.obs was  74  with Likelihood Chi Square =  642.77  with prob <  4.8e-92
```

```
##
## Tucker Lewis Index of factoring reliability =  0.266
## RMSEA index =  0.324  and the 90 % confidence intervals are  0.304 0.35
## BIC =  328.58
## Fit based upon off diagonal values = 0.99
```

Factors are:

1. Factor 1: positive effect: co, o3, so2, pm2_5,pm10 negative effect: lon, temprature
2. Factor 2: positive effect: lat,lon,pressure negative effect: temperature,feelslike
3. Factor 3: positive effect: wether_code,humidity negative effect: visibility
4. Factor 4: positive effect: no2,wind_speed
5. Factor 5: positive effect: precip

- Factor Analysis is used for identifying the **underlying structure of the data**. It helps in reducing variable and these obtained factors can be effectively used for EDA.
- The First Factor can be named as pollutants, Second Factor as geographic_cond, Third Factor is weather_cond , Fourth can be named as ozone since higher wind speed decreases no2 concentration and fifth can be named precipitation

6 Data Engineering

6.1 Adding FA Score / Factors for EDA

```r
# Storing FA Scores as df
fa_scores<-as.data.frame(fa$scores)

# renaming column names
colnames(fa_scores)=c("pollutants","geographic_cond","weather_cond","ozone","precipitation")

# concatenating 2 dfs air and fa_scores
df<-cbind(air,fa_scores)
head(df)
```

```
##        city    lat    lon temperature weather_code  sunrise   sunset moonrise
## 1 New Delhi 28.600 77.200          21          143 07:05 AM 05:26 PM 01:04 AM
## 2    Mumbai 18.975 72.826          30          122 07:03 AM 06:03 PM 01:20 AM
## 3   Kolkata 22.570 88.370          21          143 06:07 AM 04:54 PM 12:16 AM
## 4   Chennai 13.083 80.283          26          143 06:22 AM 05:44 PM 12:48 AM
## 5 Bengaluru 12.983 77.583          24          113 06:32 AM 05:55 PM 12:59 AM
## 6 Hyderabad 17.375 78.474          26          143 06:37 AM 05:44 PM 12:56 AM
##    moonset      co   no2  o3   so2  pm2_5   pm10 wind_speed wind_degree
## 1 01:06 PM 1411.85 23.95 264 76.65 137.25 140.05          4          34
## 2 01:29 PM  644.85 25.55 209 31.15  46.65  47.05         18         300
## 3 12:23 PM  457.85  1.95 214 12.95  44.55  47.25          8           3
## 4 01:00 PM  275.85  2.05 135  7.55  28.75  35.15         19          31
## 5 01:11 PM  243.85  3.85 152 10.75  20.95  26.35          9          76
```

```
## 6 01:06 PM  291.85  0.85 174 11.65  28.85  31.45         10          81
##   wind_dir pressure precip humidity cloudcover feelslike uv_index visibility
## 1       NE     1017      0       53         50        21        0          1
## 2      WNW     1011      0       35          0        32        0          4
## 3        N     1014      0       73          0        21        0          3
## 4      NNE     1012      0       65         25        28        0          5
## 5      ENE     1015      0       25          0        24        0         10
## 6        E     1016      0       32          0        26        0          4
##   pollutants geographic_cond weather_cond       ozone precipitation
## 1  2.2654204     -0.01456634   1.14973930  0.05673288    0.81763956
## 2  0.6594620     -1.67846417   0.07958311 -0.14244472    2.80381480
## 3 -0.5014364      0.03410579   1.85784828 -0.81907616   -0.07196903
## 4 -1.0885686     -0.88240122   1.44971972 -0.15516614    1.78635517
## 5 -1.4185979     -0.07237518   0.09356560  0.01079488   -0.58437966
## 6 -0.7058790     -0.53332118   0.76170481 -0.41674365   -0.08529779
```

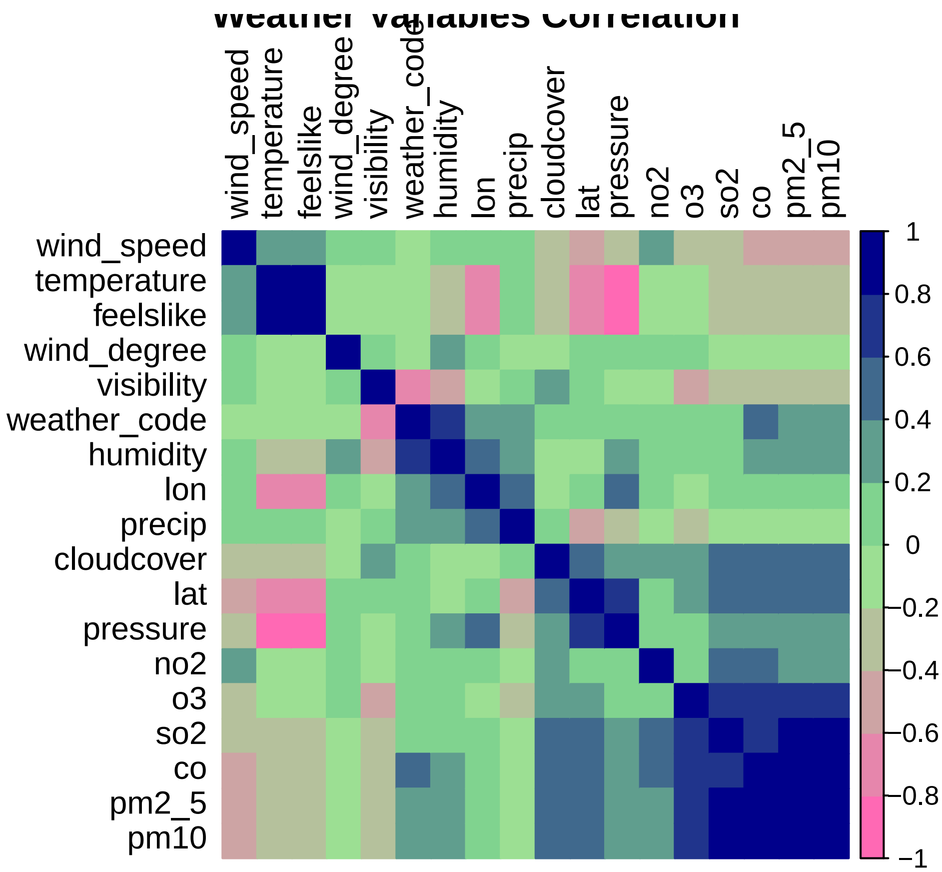
6.2 Converting Weather code and visibility to categorical variables

```r
df<-df %>%
  mutate(weather_cat=case_when(
    weather_code==113~"sunny",
    weather_code==122~"partly cloudly",
    weather_code==143~"mist",
    weather_code==116~"moderate rain",
    weather_code==119~"showers",
    weather_code==248~"fog",
    weather_code==176~"moderate rain",
  )) %>%
  mutate(visibility_cat=case_when(
    visibility<1~"very poor",
    between(visibility,1,3) ~"poor",
    between(visibility,3,5) ~"moderate",
    visibility>5 ~"good"
  )) %>%
  mutate(parts_of_India = case_when(
    between(lat, 28, 37.6) & between(lon, 68.7, 97.25) ~ "North",
    between(lat, 15, 28) & between(lon, 68, 78) ~ "West",
    between(lat, 20, 28) & between(lon, 83, 97.25) ~ "East",
    lat < 20 & between(lon, 74, 84) ~ "South",
    between(lat, 18, 26) & between(lon, 74, 85) ~ "Central",
    between(lat, 22, 28) & between(lon, 89, 97.25) ~ "Northeast",
    TRUE ~ "Other Region"
  ))
```

7 Exploratory Data Analysis

7.1 Correlation among variables

```r
# Calculate correlation with pairwise complete observations
weather_cor <- cor(num_df,
                   use = "pairwise.complete.obs")

c_color<- colorRampPalette(c("hotpink", "lightgreen","darkblue"))

corrplot(weather_cor,
         method = "color",
         title = "Weather Variables Correlation",
         order="hclust",
         col=c_color(10),
         tl.col="black"
         )
```

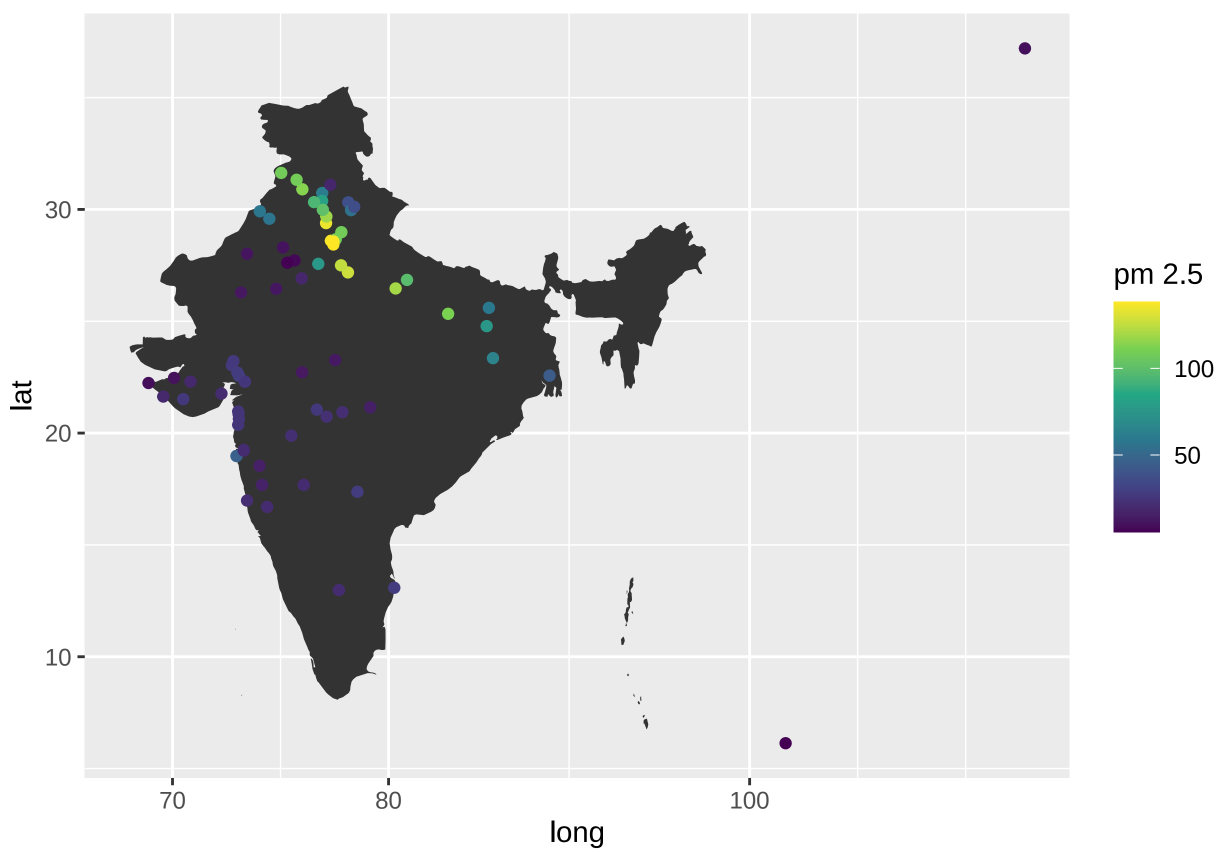


* Positive Correlation: pollutants are highly correlated such as pm2_5, pm_10, co, so2 , no2 is moderately correlated, feelslike and temperature

- Moderate Correlation (positive and negative): We can see moderate correlation between co2 and weather code,longitude with temperature and feelslike,latitude with temprature and feelslike
- Negative Correlation:pressure with feelslike and temprature

7.2 Temprature according to latitude and longitude values

```
world<-map_data("world")

#getting the map for india
india<-subset(world,region=="India")


ggplot()+
   geom_polygon(data=india,
                aes(x=long,y=lat,group=group))+
    geom_point(data=df,
             aes(x=lon,y=lat,color=pm2_5))+
  scale_color_continuous(
    type = "viridis",  # Or "gradient"
    name = "pm 2.5"
  )+
  scale_x_log10()
```

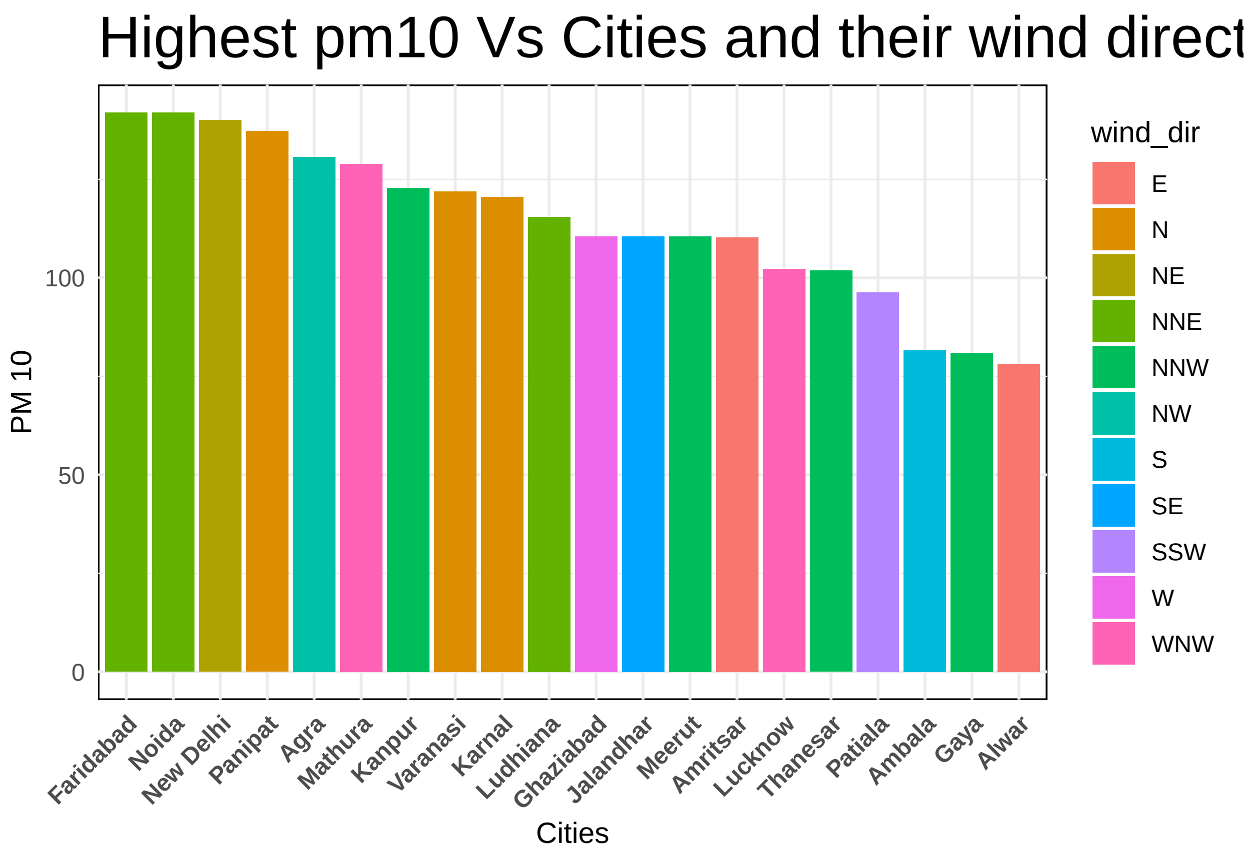



```
# setting theme for all plots
set_theme(theme_minimal()+
            theme(
                plot.title=
                  element_text(
```

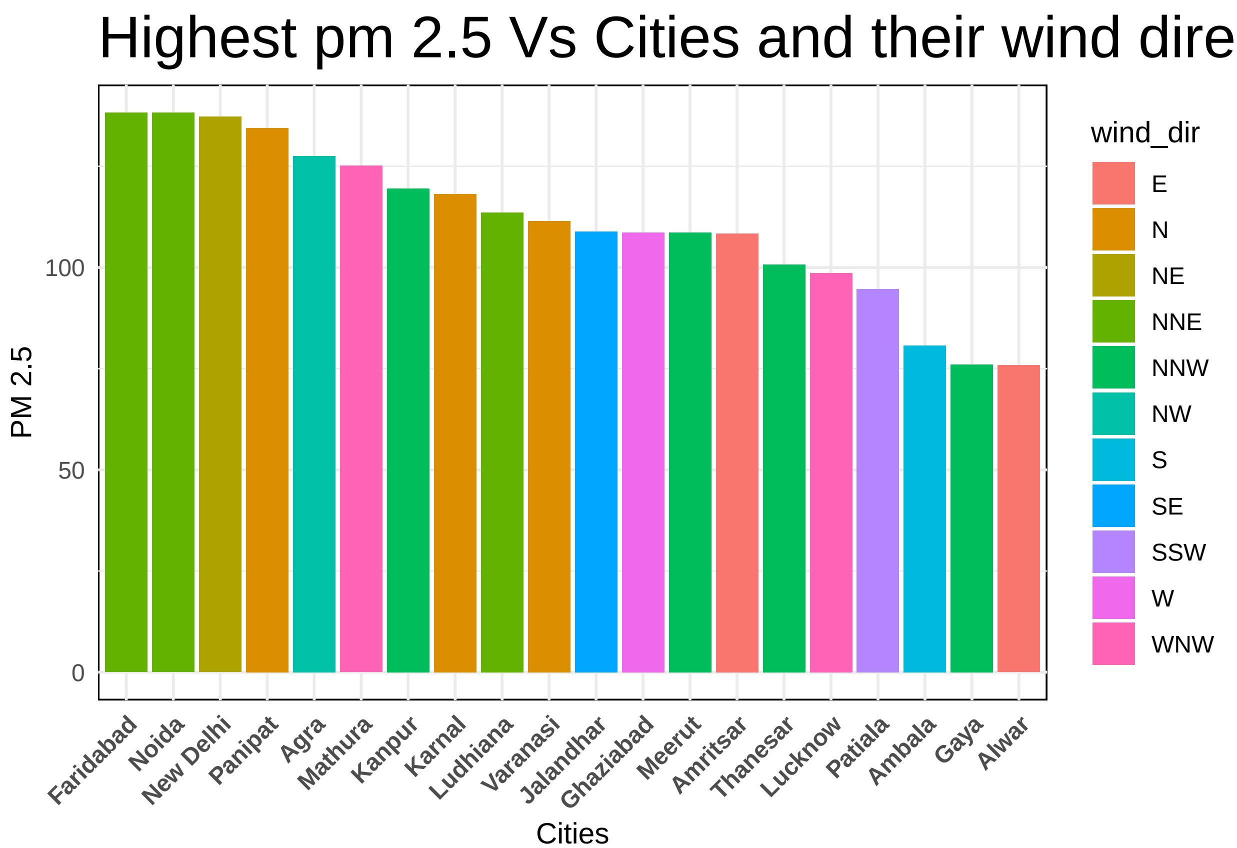
```
                    size=rel(2)),

                panel.background =
                  element_rect(color="black"),

              ))
```

7.3 Top 20 cities with worst air quality pm10 and pm2

```
# pm10
df %>%
  arrange(desc(pm10)) %>%
  select(city,pm10,wind_dir) %>%
  slice_head(n=20) %>%
  ggplot(
    aes(x=reorder(city,-pm10),y=pm10,fill=wind_dir)
  )+
  labs(
    title="Highest pm10 Vs Cities and their wind direction",
    x="Cities",
    y="PM 10"
  ) +
  geom_bar(stat="identity")+
  theme(
      axis.text.x =
                  element_text(angle=45,
                               hjust=1,
                              face="bold")
  )
```

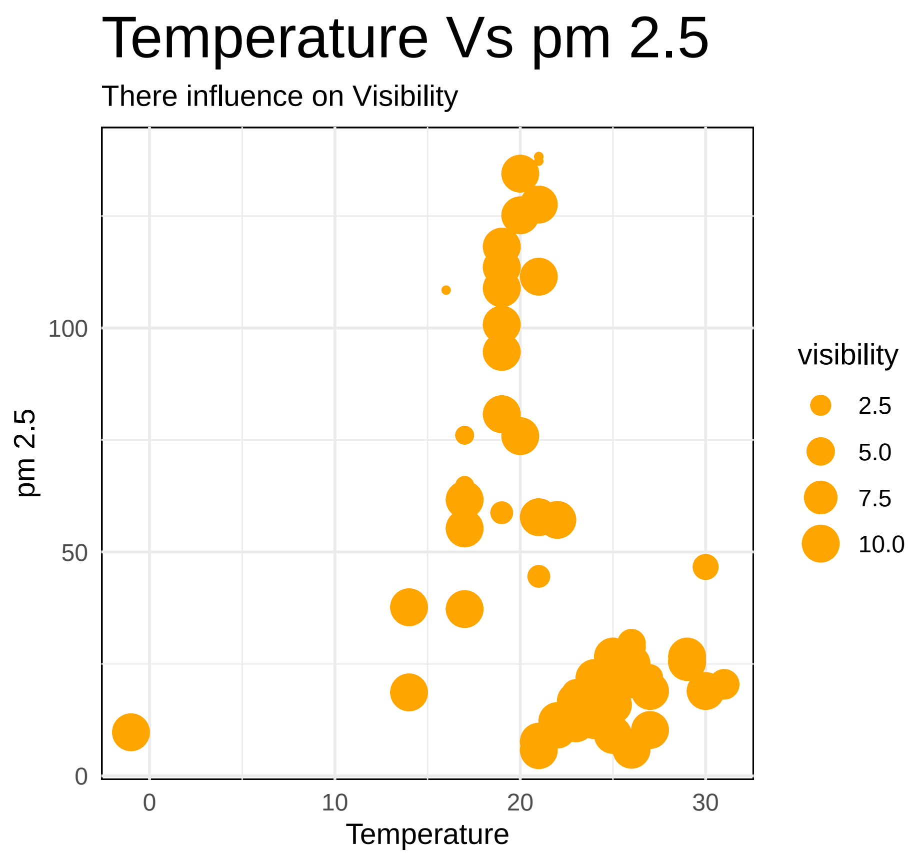



```r
df %>%
  arrange(desc(pm2_5)) %>%
  select(city,pm2_5,wind_dir) %>%
  slice_head(n=20) %>%
  ggplot(
    aes(x=reorder(city,-pm2_5),y=pm2_5,fill=wind_dir)
  )+
  geom_bar(stat="identity")+
  labs(
    title="Highest pm 2.5 Vs Cities and their wind direction",
    x="Cities",
    y="PM 2.5"
  )+
  theme(
      axis.text.x =
                  element_text(angle=45,
                               hjust=1,
                              face="bold")
  )
```



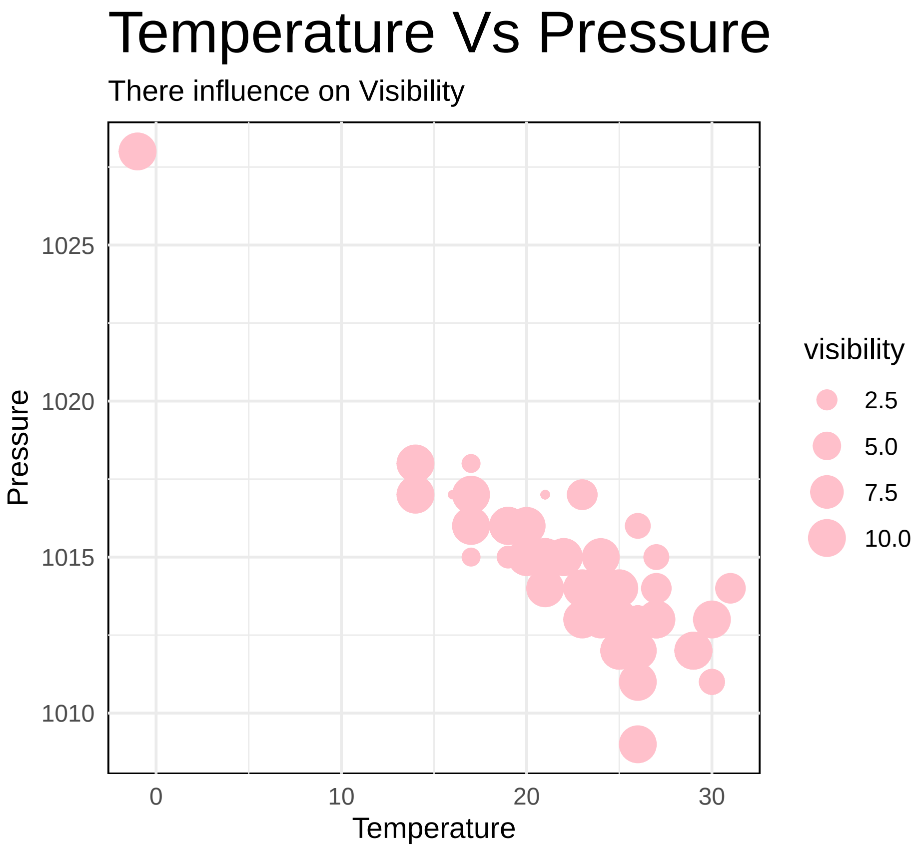
#One-on-one relationships between two continuous variables ## Temperature vs PM2.5 levels

```
# temperature Vs pm2.5 levels
plot1<-df %>%
  ggplot(aes(temperature,pm2_5,size=visibility))+
  geom_point(color="orange")+
  labs(
    title="Temperature Vs pm 2.5",
    subtitle="There influence on Visibility",
    x="Temperature",
    y="pm 2.5"
  )+
  theme(
    aspect.ratio = 1
  )
plot1
```



7.4 Temptrature Vs Pressure

```
# temperature Vs pm2.5 levels
plot2<-df %>%
  ggplot(aes(temperature,pressure,size=visibility))+
  geom_point(color="pink")+
  labs(
    title="Temperature Vs Pressure",
    subtitle="There influence on Visibility",
    x="Temperature",
    y="Pressure"
  )+
  theme(
    aspect.ratio = 1
  )

plot2
```



Observing 3 variables Wind speed , Temprature and pm 2.5 concentration

```
#fig <- plot_ly(df, x = ~wind_speed, y = ~temperature, z = ~pm2_5, color = ~city)
#fig
```

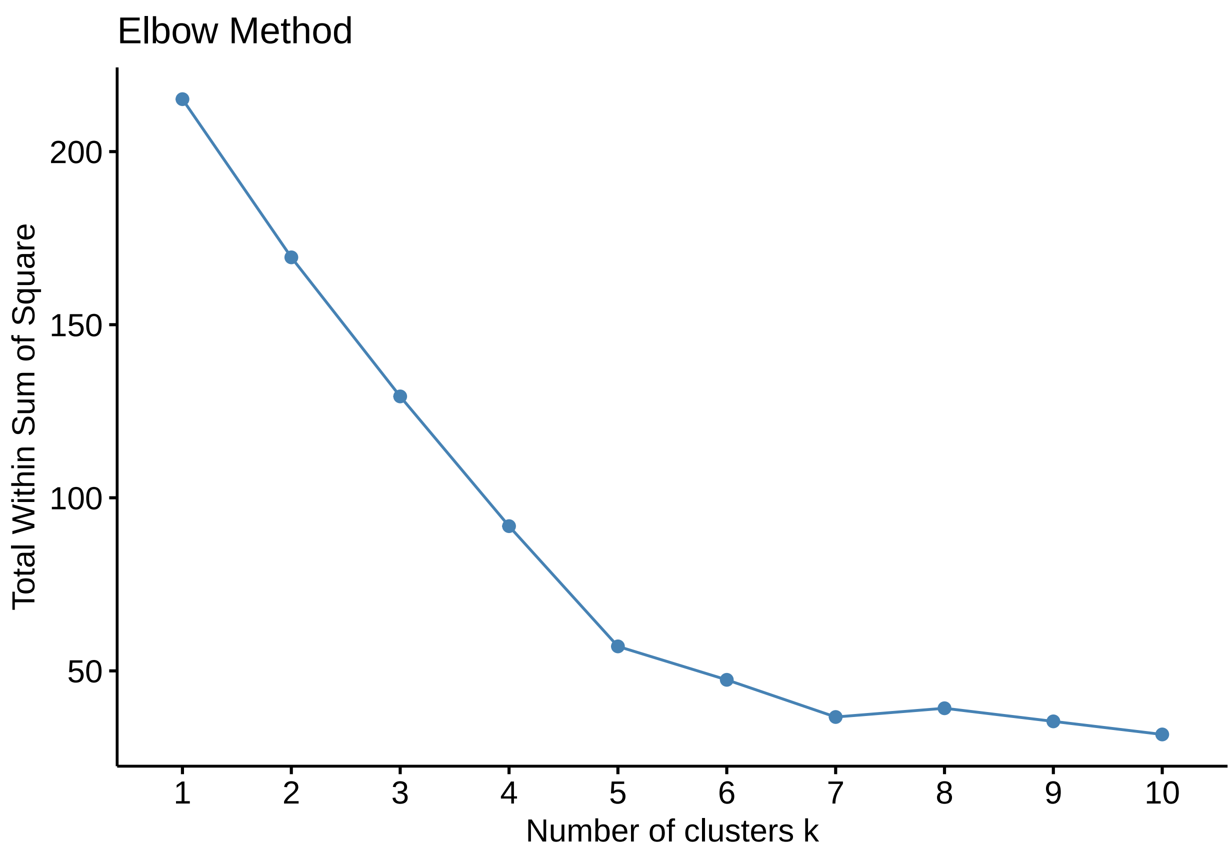
8 Clustering based on pollutants

8.1 Making dataset for clustering

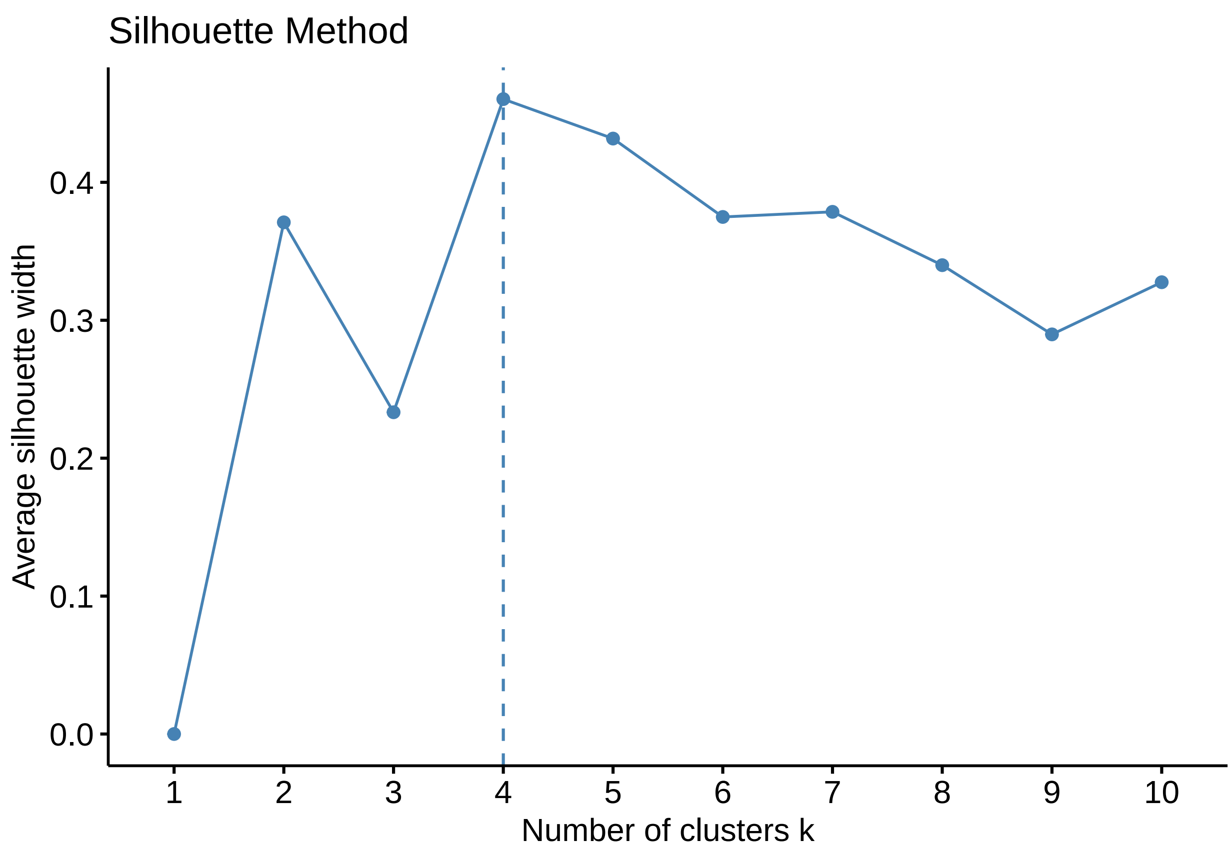
```
# Data for clustering
k_data<-df %>%
  select("city","pollutants","weather_cond","geographic_cond")
```

8.2 Selecting number of clusters

```
fviz_nbclust(k_data[,c("pollutants","weather_cond","geographic_cond")], kmeans, method = "wss") +
  ggtitle("Elbow Method")
```




```
fviz_nbclust(k_data[,c("pollutants","weather_cond","geographic_cond")], kmeans, method = "silhouette") +
  ggtitle("Silhouette Method")
```



Performing K-means clustering

```r
# Perform k-means clustering (e.g., 4 clusters)
set.seed(123)
kmeans_result <- kmeans(k_data[,c("pollutants","weather_cond","geographic_cond")], centers = 4, nstart =
print(kmeans_result)
```

```
## K-means clustering with 4 clusters of sizes 3, 13, 46, 12
##
## Cluster means:
##   pollutants weather_cond geographic_cond
## 1 -1.2598530   -0.1369784       3.3428137
## 2  1.3139388   -0.9437692       0.3596657
## 3 -0.5321342   -0.1300377      -0.4335428
## 4  0.9313775    1.5551392       0.4365730
##
## Clustering vector:
##  [1] 4 3 4 3 3 3 3 3 3 2 4 4 3 3 3 4 4 4 3 2 2 4 4 4 4 2 2 2 4 2 2 2 1 1 3 3 3 3
## [39] 3 3 2 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 1 3 3
##
## Within cluster sum of squares by cluster:
## [1]  9.656581  6.247595 44.417537 16.614590
##  (between_SS / total_SS =  64.2 %)
##
## Available components:
##
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

8.3 Storing Clusters as factor

```r
k_data$cluster <- as.factor(kmeans_result$cluster)

k_data %>%
  mutate(cluster_name = case_when(
    cluster == 1 ~ "Extremely Clean, Very Warm",
    cluster== 2 ~ "Outlier City",
    cluster== 3 ~ "Clean, Bad Weather",
    cluster== 4 ~ "Very Polluted, Bad Weather",
    cluster== 5 ~ "Polluted, Good Weather",
    TRUE ~ "Unknown"
  ))
```

```
##            city  pollutants weather_cond geographic_cond cluster
## 1     New Delhi  2.26542045   1.14973930    -0.014566338       4
## 2        Mumbai  0.65946204   0.07958311    -1.678464169       3
## 3       Kolkata -0.50143636   1.85784828     0.034105792       4
## 4       Chennai -1.08856858   1.44971972    -0.882401217       3
## 5     Bengaluru -1.41859791   0.09356560    -0.072375182       3
## 6     Hyderabad -0.70587902   0.76170481    -0.533321183       3
## 7     Ahmedabad -0.22039040   0.08647039    -0.199522710       3
## 8          Pune -0.72680901  -0.47723834    -0.022275075       3
## 9        Jaipur -0.67121486   0.54159772    -0.162219276       3
## 10   Chandigarh  0.33462381  -0.89573481     1.184933236       2
## 11      Lucknow  0.56578380   1.71054907     0.475497778       4
## 12       Kanpur  1.16697096   1.65321460     0.456999291       4
## 13       Nagpur -0.43357047   0.46689526    -0.583885400       3
## 14       Indore -0.90383472   0.84862006    -0.364412393       3
## 15       Bhopal -0.60178871   0.40379245    -0.320215428       3
## 16        Patna  0.07161826   1.83336578     0.276206414       4
## 17       Ranchi -0.44130525   1.14047631     1.495417088       4
## 18         Gaya -0.35098109   2.41696052     0.968987935       4
## 19     Varanasi  0.38573868  -0.19359786    -0.223358158       3
## 20         Agra  1.94924467  -0.93161685    -0.109004565       2
## 21      Mathura  1.81885499  -0.89947691     0.194886537       2
## 22       Meerut  1.47948994   1.29524497     0.028071376       4
## 23    Ghaziabad  1.62499017   1.31462610     0.015084704       4
## 24        Noida  2.18276001   1.17916351     0.009809733       4
## 25    Faridabad  1.83083431   1.08918102     0.224382393       4
## 26      Panipat  2.21197958  -1.12352087     0.135341014       2
## 27       Karnal  1.73293325  -0.94026083     0.525762231       2
## 28       Ambala  1.13508060  -0.70246219     0.341130074       2
## 29     Amritsar  1.28238453   2.02130047     1.268879378       4
## 30     Ludhiana  1.78447267  -0.71437929     0.355747038       2
## 31    Jalandhar  1.62812195  -0.80504303     0.363384454       2
## 32      Patiala  0.94895716  -0.73634264     0.480126320       2
## 33       Shimla -1.43227015  -0.27717240     2.152548011       1
## 34     Dehradun -0.61474604  -0.17365132     2.092699182       1
```

```
## 35      Haridwar -0.39243473  -0.38613070     1.086826825       3
## 36     Rishikesh -0.73958738  -0.02513954     1.021270260       3
## 37       Jodhpur -0.72529489  -0.93410411    -0.142860476       3
## 38       Udaipur -0.72877293  -1.40472487     0.603662593       3
## 39         Ajmer -0.66994196  -0.95096502     0.220211147       3
## 40    Kota Bharu -1.04684167   1.22716272    -1.151914630       3
## 41         Alwar  0.48790117  -0.90898821     0.560222610       2
## 42       Bikaner -0.91097494  -1.36783317     0.247425707       3
## 43         Sikar -0.99361843  -1.13638370     0.500500199       3
## 44         Churu -0.54235445  -1.51725493     0.378214718       3
## 45 Sri Ganganagar  1.13154820  -1.43270787     0.176375347       2
## 46   Hanumangarh  0.62705819  -1.49444296     0.116217600       2
## 47         Surat -0.34469871  -0.31694794    -0.680984311       3
## 48      Vadodara -0.47446837  -0.60853938    -1.104097462       3
## 49        Rajkot -0.63436810  -0.99334686    -0.857485917       3
## 50     Bhavnagar -0.48116891   0.21468876    -0.377836887       3
## 51      Jamnagar -0.85794209  -0.74568992    -0.528638856       3
## 52      Junagadh -0.34826280  -0.67261460    -1.117142475       3
## 53     Porbandar -0.64638698  -0.25136666    -0.744836562       3
## 54        Dwarka -1.07089775  -0.03175381    -0.286079808       3
## 55   Gandhinagar -0.33444884   0.18629497    -0.291794272       3
## 56         Anand -0.31677483   0.27256961    -0.273544614       3
## 57      Vadodara -0.47446837  -0.60853938    -1.104097462       3
## 58        Nadiad -0.30272225   0.25526577    -0.301202444       3
## 59        Valsad -0.23575717  -0.28494555    -0.617159117       3
## 60          Vapi -0.23588332  -0.26965938    -0.609907897       3
## 61       Navsari -0.26606092  -0.26377951    -0.621489737       3
## 62      Thanesar  1.29042819  -0.68402286     0.350531953       2
## 63        Mumbai  0.65946204   0.07958311    -1.678464169       3
## 64        Kalyan -0.83487793   0.60626639    -0.913155280       3
## 65    Ulhasnagar -0.70128389   0.60494434    -1.199729032       3
## 66    Aurangabad -0.72161538   0.95000922    -0.744028214       3
## 67       Jalgaon -0.19377784  -0.67310653    -0.368007446       3
## 68         Akola -0.26750307  -0.72041991    -0.264097923       3
## 69      Amravati -0.40314100  -0.32363604    -0.562984057       3
## 70       Solapur -0.57522647   0.12962318    -1.430998309       3
## 71      Kolhapur -0.54873826   0.16607256    -0.505526300       3
## 72        Sangli -1.73254291   0.03988858     5.783193802       1
## 73        Satara -0.49054433  -0.38965758     0.001760754       3
## 74     Ratnagiri -0.90134519   0.14121007    -0.482328740       3
##                cluster_name
## 1  Very Polluted, Bad Weather
## 2          Clean, Bad Weather
## 3  Very Polluted, Bad Weather
## 4          Clean, Bad Weather
## 5          Clean, Bad Weather
## 6          Clean, Bad Weather
## 7          Clean, Bad Weather
## 8          Clean, Bad Weather
## 9          Clean, Bad Weather
## 10               Outlier City
## 11 Very Polluted, Bad Weather
## 12 Very Polluted, Bad Weather
## 13         Clean, Bad Weather
```

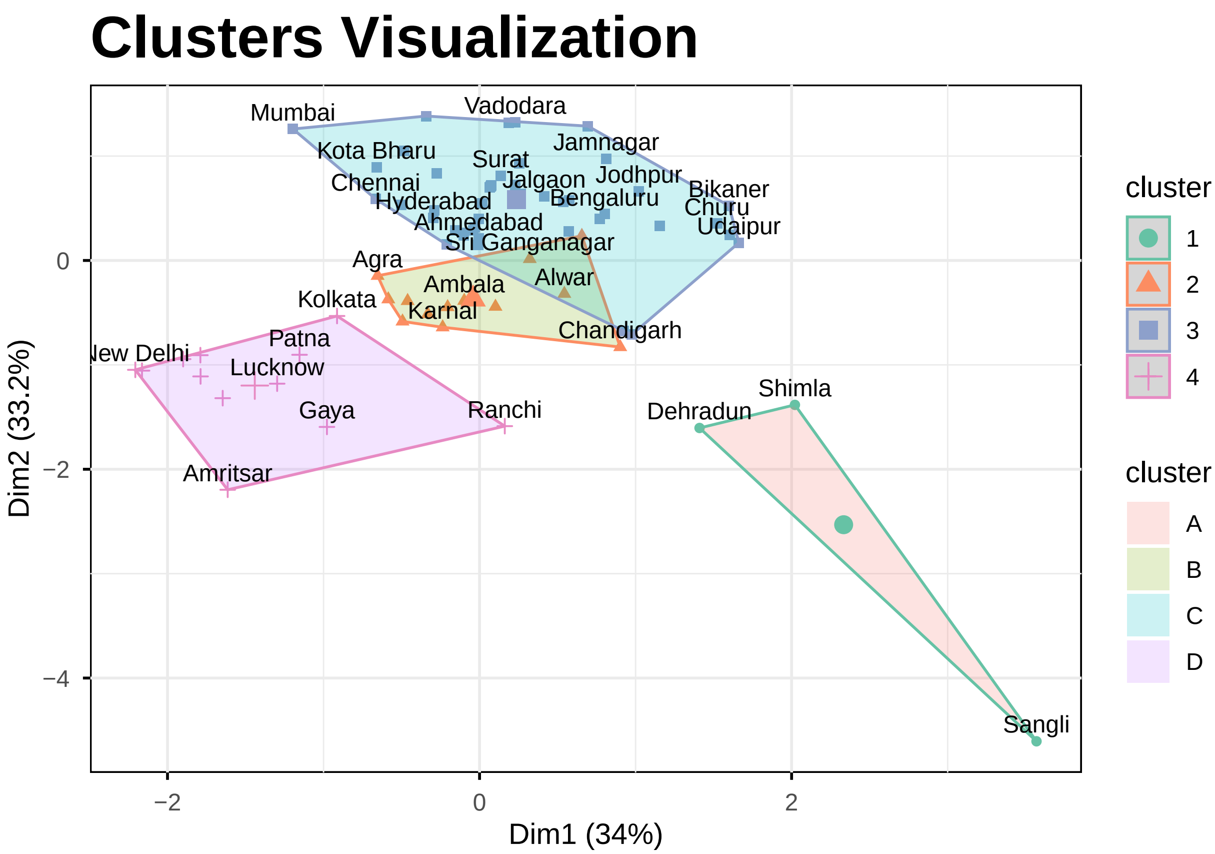
```
## 14         Clean, Bad Weather
## 15         Clean, Bad Weather
## 16 Very Polluted, Bad Weather
## 17 Very Polluted, Bad Weather
## 18 Very Polluted, Bad Weather
## 19         Clean, Bad Weather
## 20               Outlier City
## 21               Outlier City
## 22 Very Polluted, Bad Weather
## 23 Very Polluted, Bad Weather
## 24 Very Polluted, Bad Weather
## 25 Very Polluted, Bad Weather
## 26               Outlier City
## 27               Outlier City
## 28               Outlier City
## 29 Very Polluted, Bad Weather
## 30               Outlier City
## 31               Outlier City
## 32               Outlier City
## 33 Extremely Clean, Very Warm
## 34 Extremely Clean, Very Warm
## 35         Clean, Bad Weather
## 36         Clean, Bad Weather
## 37         Clean, Bad Weather
## 38         Clean, Bad Weather
## 39         Clean, Bad Weather
## 40         Clean, Bad Weather
## 41               Outlier City
## 42         Clean, Bad Weather
## 43         Clean, Bad Weather
## 44         Clean, Bad Weather
## 45               Outlier City
## 46               Outlier City
## 47         Clean, Bad Weather
## 48         Clean, Bad Weather
## 49         Clean, Bad Weather
## 50         Clean, Bad Weather
## 51         Clean, Bad Weather
## 52         Clean, Bad Weather
## 53         Clean, Bad Weather
## 54         Clean, Bad Weather
## 55         Clean, Bad Weather
## 56         Clean, Bad Weather
## 57         Clean, Bad Weather
## 58         Clean, Bad Weather
## 59         Clean, Bad Weather
## 60         Clean, Bad Weather
## 61         Clean, Bad Weather
## 62               Outlier City
## 63         Clean, Bad Weather
## 64         Clean, Bad Weather
## 65         Clean, Bad Weather
## 66         Clean, Bad Weather
## 67         Clean, Bad Weather
```

```
## 68         Clean, Bad Weather
## 69         Clean, Bad Weather
## 70         Clean, Bad Weather
## 71         Clean, Bad Weather
## 72 Extremely Clean, Very Warm
## 73         Clean, Bad Weather
## 74         Clean, Bad Weather
```

8.4 Visualizing clusters

```r
city_names<-k_data$city
fviz_cluster(kmeans_result, data = k_data[,c("pollutants","weather_cond","geographic_cond")],
             palette = "Set2", ggtheme = theme_minimal(),
             geom = "point") +
  geom_text(aes(label = city_names),
            check_overlap = TRUE,
            size = 3,
            vjust = -0.5)+
 labs(
   title="Clusters Visualization"
 )+
  theme(
    plot.title =
      element_text(face = "bold",
                   size=rel(2)),
    panel.background =
                  element_rect(color="black")
  )+
  scale_fill_discrete(labels = c("A", "B", "C","D","E"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```


Clusters Visualization
Mumbai
Vadodara
Jamnagar
Kota Bharu
Surat
Chennai
Jalgaon
Jodhpur
Bikaner
Hyderabad
Bengaluru
Churu
Ahmedabad
Udaipur
Sri Ganganagar
Agra
Ambala
Alwar
Kolkata
Karnal
Chandigarh
Patna
New Delhi
Lucknow
Shimla
Gaya
Ranchi
Dehradun
Amritsar
Sangli
0
−2
−4
−2
0
2
Dim1 (34%)
Dim2 (33.2%)
cluster
1
2
3
4
cluster
A
B
C
D