FINAL EXAM DANA-4830

Predict annual average air pollution concentrations of various cities in the U.S.

NAME: HARPREET KAUR
STUDENT ID: 100357359
SUBMITTED TO: MISS MONICA

OBJECTIVE

The main objective of air pollution data is to predict the annual average air pollution concentrations of various cities in the U.S. using predictors such as data about population density, urbanization, and road density, as well as satellite pollution data and chemical modelling data.

STAGE 1 – EXPLORING DATA

1. **Explore data, provide a descriptive analysis of the variables, and decide if you need to transform the variables to the right formats.**

The datasets provide the report of air pollution concentrations of various cities in the U.S. Moreover, the data set descriptions are as follows:

TABLE 1: DATASET DESCRIPTION		
Dataset	**Number of rows**	**Number of Columns**
Pm2.5_data	876	50
Total	876	50

In the dataset, there are 50 attributes/features with values for each of the 876 monitors (observations). **Table-1**

DESCRIPTIVE ANALYSIS OF VARIABLES

- Checking the structure of variables, found that there were 876 observations with 50 numbers of features**.** The value which is the most important column indicates the PM2.5 concentration level average of fine particles/volume of air for 876 gravimetric monitors for 2008.

```
spec_tbl_df [876 x 50] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ id                         : num [1:876] 1003 1027 1033 1049 1055 ...
 $ value                      : num [1:876] 9.6 10.8 11.2 11.7 12.4 ...
 $ fips                       : num [1:876] 1003 1027 1033 1049 1055 ...
 $ lat                        : num [1:876] 30.5 33.3 34.8 34.3 34 ...
 $ lon                        : num [1:876] -87.9 -85.8 -87.7 -86 -86 ...
 $ state                      : chr [1:876] "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ county                     : chr [1:876] "Baldwin" "Clay" "Colbert" "DeKalb" ...
 $ city                       : chr [1:876] "Fairhope" "Ashland" "Muscle Shoals" "Crossville" ...
 $ CMAQ                       : num [1:876] 8.1 9.77 9.4 8.53 9.24 ...
 $ zcta                       : num [1:876] 36532 36251 35660 35962 35901 ...
 $ zcta_area                  : num [1:876] 1.91e+08 3.74e+08 1.67e+07 2.04e+08 1.54e+08 ...
 $ zcta_pop                   : num [1:876] 27829 5103 9042 8300 20045 ...
 $ imp_a500                   : num [1:876] 0.0173 1.9697 19.173 5.782 16.4931 ...
 $ imp_a1000                  : num [1:876] 1.41 0.853 11.145 3.868 11.096 ...
 $ imp_a5000                  : num [1:876] 3.336 0.985 15.179 1.231 14.678 ...
 $ imp_a10000                 : num [1:876] 1.988 0.521 9.725 1.032 8.22 ...
 $ imp_a15000                 : num [1:876] 1.439 0.336 5.247 0.973 5.161 ...
 $ county_area                : num [1:876] 4.12e+09 1.56e+09 1.53e+09 2.01e+09 1.39e+09 ...
 $ county_pop                 : num [1:876] 182265 13932 54428 71109 104430 ...
 $ log_dist_to_prisec         : num [1:876] 4.65 7.22 5.76 3.72 5.26 ...
 $ log_pri_length_5000        : num [1:876] 8.52 8.52 8.52 8.52 9.07 ...
 $ log_pri_length_10000       : num [1:876] 9.21 9.21 9.27 10.41 11.14 ...
 $ log_pri_length_15000       : num [1:876] 9.63 9.62 9.66 11.17 11.59 ...
 $ log_pri_length_25000       : num [1:876] 11.3 10.1 10.2 11.9 12 ...
 $ log_prisec_length_500      : num [1:876] 7.3 6.21 8.61 7.31 8.74 ...
 $ log_prisec_length_1000     : num [1:876] 8.2 7.6 9.74 8.59 9.63 ...
 $ log_prisec_length_5000     : num [1:876] 10.8 10.2 11.8 10.2 11.7 ...
 $ log_prisec_length_10000    : num [1:876] 11.9 11.4 12.8 11.5 12.8 ...
 $ log_prisec_length_15000    : num [1:876] 12.2 12 13.3 12.4 13.2 ...
 $ log_prisec_length_25000    : num [1:876] 13.4 12.8 13.8 13.6 13.7 ...
```

- Checked missing data of all the variables and noted that there was no missing value in the dataset. So, we can proceed further.

	missing
id	0
value	0
fips	0
lat	0
lon	0
state	0
county	0
city	0
CMAQ	0
zcta	0
zcta_area	0
zcta_pop	0
imp_a500	0
imp_a1000	0
imp_a5000	0
imp_a10000	0
imp_a15000	0
county_area	0
county_pop	0
log_dist_to_prisec	0

	missing
log_pri_length_5000	0
log_pri_length_10000	0
log_pri_length_15000	0
log_pri_length_25000	0
log_prisec_length_500	0
log_prisec_length_1000	0
log_prisec_length_5000	0
log_prisec_length_10000	0
log_prisec_length_15000	0
log_prisec_length_25000	0
log_nei_2008_pm25_sum_10000	0
log_nei_2008_pm25_sum_15000	0
log_nei_2008_pm25_sum_25000	0
log_nei_2008_pm10_sum_10000	0
log_nei_2008_pm10_sum_15000	0
log_nei_2008_pm10_sum_25000	0
popdens_county	0
popdens_zcta	0
nohs	0
somehs	0

	missing
log_nei_2008_pm25_sum_25000	0
log_nei_2008_pm10_sum_10000	0
log_nei_2008_pm10_sum_15000	0
log_nei_2008_pm10_sum_25000	0
popdens_county	0
popdens_zcta	0
nohs	0
somehs	0
hs	0
somecollege	0
associate	0
bachelor	0
grad	0
pov	0
hs_orless	0
urc2013	0
urc2006	0
aod	0

- Noted that some columns are numeric and do not contains numerical values example id, flips, zcta thus converting them into the factor variable.

```
spec_tbl_df [876 x 50] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ id                    : Factor w/ 876 levels "1003.001","1027.0001",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ value                 : num [1:876] 9.6 10.8 11.2 11.7 12.4 ...
 $ fips                  : Factor w/ 569 levels "1003","1027",..: 1 2 3 4 5 6 7 7 7 7 ...
 $ lat                   : num [1:876] 30.5 33.3 34.8 34.3 34 ...
 $ lon                   : num [1:876] -87.9 -85.8 -87.7 -86 -86 ...
 $ state                 : chr [1:876] "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ county                : chr [1:876] "Baldwin" "Clay" "Colbert" "DeKalb" ...
 $ city                  : chr [1:876] "Fairhope" "Ashland" "Muscle Shoals" "Crossville" ...
 $ CMAQ                  : num [1:876] 8.1 9.77 9.4 8.53 9.24 ...
 $ zcta                  : Factor w/ 842 levels "1022","1103",..: 305 303 298 301 300 304 292 288 295 287 ...
```

```
-- Data Summary ------------------------
                           Values
Name                       df
Number of rows             876
Number of columns          50
_______________________
Column type frequency:
  character                3
  factor                   3
  numeric                  44
________________________
Group variables            None
```

- Used skim package to check the distribution of the numerical dataset.

skim_variabe	miss	complete_rate	mean	sd	p0	p25	p50	p75	p100	Histogram
value	0	1	10.81	2.58	3.02	9.27	11.15	12.37	23.16	
lat	0	1	38.48	4.62	25.47	35.03	39.30	41.66	48.40	
lon	0	1	-91.74	14.96	-124.18	-99.16	-87.47	-80.69	-68.04	
CMAQ	0	1	8.41	2.97	1.63	6.53	8.62	10.24	23.13	
zcta_area	0	1	183173481.91	542598878.48	15459.00	14204601.75	37653560.50	160041508.25	8164820625.00	
zcta_pop	0	1	24227.58	17772.16	0.00	9797.00	22014.00	35004.75	95397.00	
imp_a500	0	1	24.72	19.34	0.00	3.70	25.12	40.22	69.61	
imp_a1000	0	1	24.26	18.02	0.00	5.32	24.53	38.59	67.50	
imp_a5000	0	1	19.93	14.72	0.05	6.79	19.07	30.11	74.60	

imp_a10000	0	1	15.82	13.81	0.09	4.54	12.36	24.17	72.09	
imp_a15000	0	1	13.43	13.12	0.11	3.24	9.67	20.55	71.10	
county_area	0	1	3768701992.12	6212829553.56	33703512.00	1116536297.50	1690826566.50	2878192209.00	51947229509.00	
county_pop	0	1	687298.44	1293488.74	783.00	100948.00	280730.50	743159.00	9818605.00	
log_dist_to_prisec	0	1	6.19	1.41	-1.46	5.43	6.36	7.15	10.45	
log_pri_length_5000	0	1	9.82	1.08	8.52	8.52	10.05	10.73	12.05	
log_pri_length_10000	0	1	10.92	1.13	9.21	9.80	11.17	11.83	13.02	
log_pri_length_15000	0	1	11.50	1.15	9.62	10.87	11.72	12.40	13.59	
log_pri_length_25000	0	1	12.24	1.10	10.13	11.69	12.46	13.05	14.36	
log_prisec_length_500	0	1	6.99	0.95	6.21	6.21	6.21	7.82	9.40	
log_prisec_length_1000	0	1	8.56	0.79	7.60	7.60	8.66	9.20	10.47	
log_prisec_length_5000	0	1	11.28	0.78	8.52	10.91	11.42	11.83	12.78	
log_prisec_length_10000	0	1	12.41	0.73	9.21	11.99	12.53	12.94	13.85	
log_prisec_length_15000	0	1	13.03	0.72	9.62	12.59	13.13	13.57	14.41	
log_prisec_length_25000	0	1	13.82	0.70	10.13	13.38	13.92	14.35	15.23	
log_nei_2008_pm25_sum_10000	0	1	3.97	2.35	0.00	2.15	4.29	5.69	9.12	
log_nei_2008_pm25_sum_15000	0	1	4.72	2.25	0.00	3.47	5.00	6.35	9.42	
log_nei_2008_pm25_sum_25000	0	1	5.67	2.11	0.00	4.66	5.91	7.28	9.65	
log_nei_2008_pm10_sum_10000	0	1	4.35	2.32	0.00	2.69	4.62	6.07	9.34	
log_nei_2008_pm10_sum_15000	0	1	5.10	2.18	0.00	3.87	5.39	6.72	9.71	

log_nei_2008_pm10_sum_25000	0	1	6.07	2.01	0.00	5.10	6.37	7.52	9.88	
popdens_county	0	1	551.76	1711.51	0.26	40.77	156.67	510.81	26821.91	
popdens_zcta	0	1	1279.66	2757.49	0.00	101.15	610.35	1382.52	30418.84	
nohs	0	1	6.99	7.21	0.00	2.70	5.10	8.80	100.00	
somehs	0	1	10.17	6.20	0.00	5.90	9.40	13.90	72.20	
hs	0	1	30.32	11.40	0.00	23.80	30.75	36.10	100.00	
somecollege	0	1	21.58	8.60	0.00	17.50	21.30	24.70	100.00	
associate	0	1	7.13	4.01	0.00	4.90	7.10	8.80	71.40	
bachelor	0	1	14.90	9.71	0.00	8.80	12.95	19.23	100.00	
grad	0	1	8.91	8.65	0.00	3.90	6.70	11.00	100.00	
pov	0	1	14.95	11.33	0.00	6.50	12.10	21.23	65.90	
hs_orless	0	1	47.48	16.75	0.00	37.93	48.65	59.10	100.00	
urc2013	0	1	2.92	1.52	1.00	2.00	3.00	4.00	6.00	
urc2006	0	1	2.97	1.52	1.00	2.00	3.00	4.00	6.00	
aod	0	1	43.70	19.56	5.00	31.66	40.17	49.67	143.00	

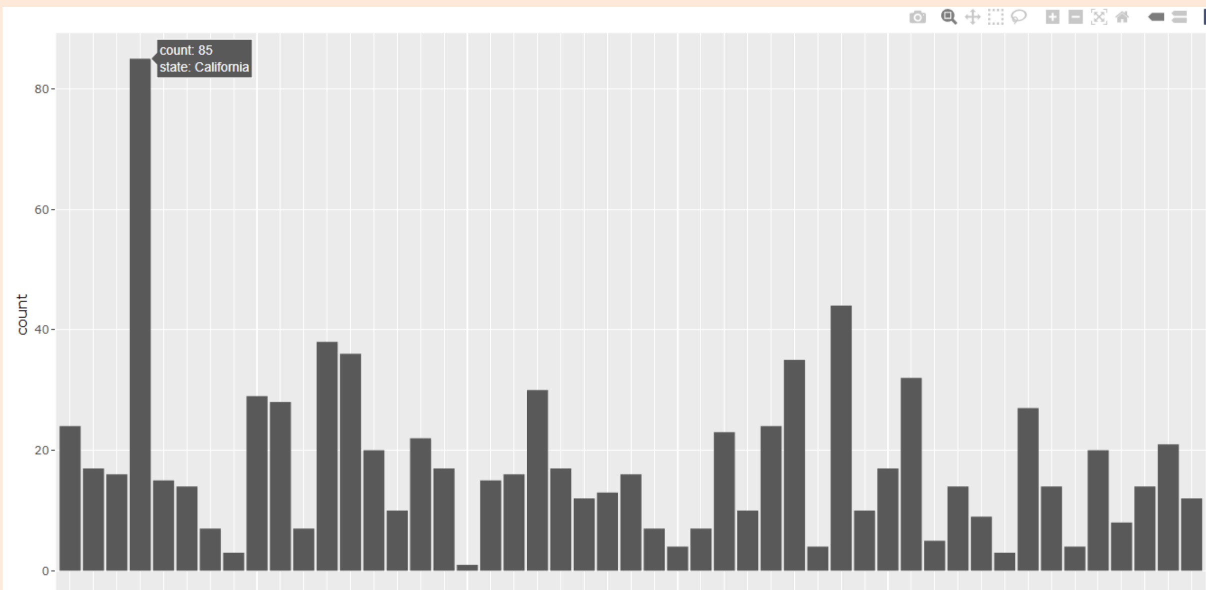
Summary:

- No missing value in the numerical dataset.
- **Mean** represent the average value of each column
- **Standard deviation explains about** spread or dispersion of our data points around the mean value
- **Quantile (p0 to p100)** tells about the minimum, maximum, range of the dataset
- **Histogram** tells about the distribution of the numerical values**.** To illustrate some of the **Right skewerd Histogram are (**urc2013, urc2006,somecollege, associate, zcta_area, zcta_pop**). Left skewed (**value, log_dist_to_prisec, log_prisec_length_5000, aod**). Normally distributed (**hs_orless, CMAQ, log_nei_2008_pm25_sum_25000**).**

- Used bar plot to check count for the categorical dataset.

variable	miss	complete_rate	min	max	empty	n_unique	whitespace
state	0	1	4	20	0	49	0
county	0	1	3	20	0	471	0
city	0	1	4	48	0	607	0

STATE:



CITY:

- Factor variables achieved from skim package

variable	miss	complete_rate	ordered	n_unique	top_counts
id	0	1	FALSE	876	100: 1, 102: 1, 103: 1, 104: 1
fips	0	1	FALSE	569	170: 12, 603: 10, 261: 9, 107: 8
zcta	0	1	FALSE	842	475: 3, 110: 2, 160: 2, 290: 2

2. How many states are there in the dataset?

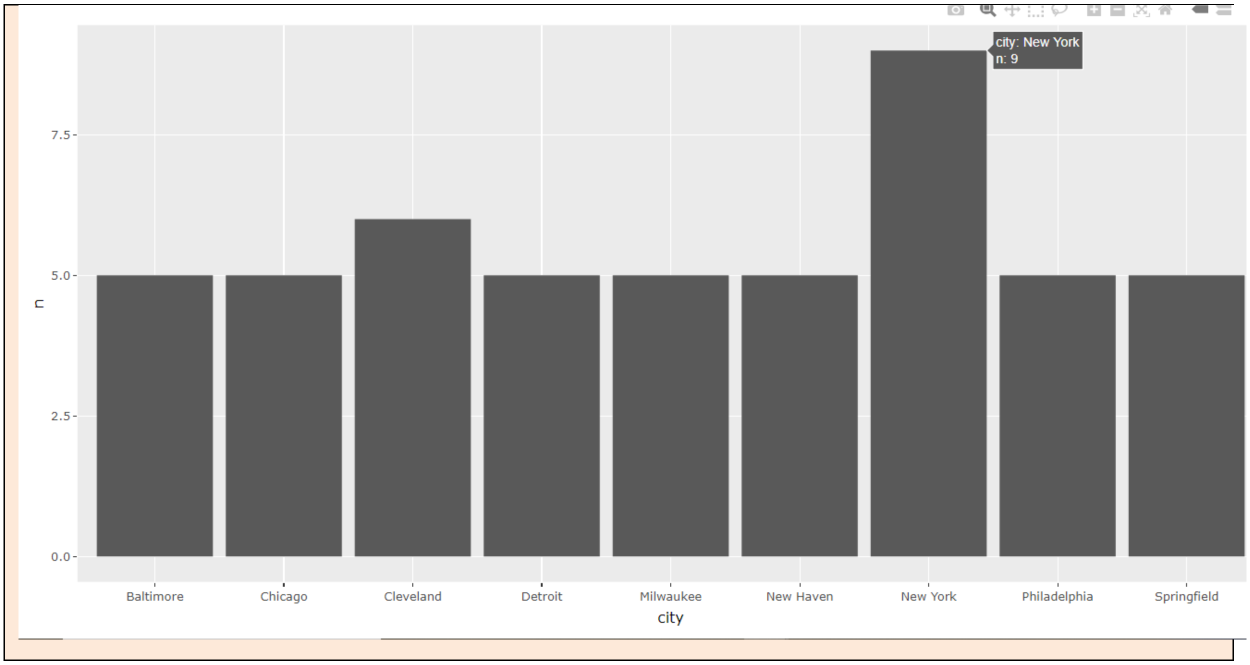
```
> df_state
 [1] "Alabama"             "Arizona"              "Arkansas"
 [4] "California"          "Colorado"             "Connecticut"
 [7] "Delaware"            "District Of Columbia" "Florida"
[10] "Georgia"             "Idaho"                "Illinois"
[13] "Indiana"             "Iowa"                 "Kansas"
[16] "Kentucky"            "Louisiana"            "Maine"
[19] "Maryland"            "Massachusetts"        "Michigan"
[22] "Minnesota"           "Mississippi"          "Missouri"
[25] "Montana"             "Nebraska"             "Nevada"
[28] "New Hampshire"       "New Jersey"           "New Mexico"
[31] "New York"            "North Carolina"       "North Dakota"
[34] "Ohio"                "Oklahoma"             "Oregon"
[37] "Pennsylvania"        "Rhode Island"         "South Carolina"
[40] "South Dakota"        "Tennessee"            "Texas"
[43] "Utah"                "Vermont"              "Virginia"
[46] "Washington"          "West Virginia"        "Wisconsin"
[49] "Wyoming"
```

There are 49 states in the dataset as shown above. **Using Annex-2.**

3. How many stations are there per city?

	city	n
1	Aberdeen	1
2	Akron	2
3	Albany	3
4	Albuquerque	2
5	Alexandria	1
6	Allen Park	1
7	Altamont	1
8	Alton	1
9	Amarillo	1
10	Anadarko	1
11	Anaheim	1
12	Anderson	1
13	Annandale	1
14	Apache Junction	1
15	Apple Valley	1
16	Appleton	1
17	Arden-Arcade	1
18	Arlington	1
19	Arnold	1
20	Asheville	1

	city	n
491	San Luis Obispo	1
492	Sandersville	1
493	Sanford	1
494	Santa Barbara	1
495	Santa Fe	1
496	Santa Maria	1
497	Santa Rosa	1
498	Sault Ste. Marie	2
499	Savannah	1
500	Schiller Park	1
501	Scottsbluff	1
502	Scottsdale	1
503	Scranton	1
504	Seaford	1
505	Searcy	1
506	Seattle	2
507	Seeley Lake	1
508	Seven Oaks	1
509	Shakopee	1
510	Sharonville	1



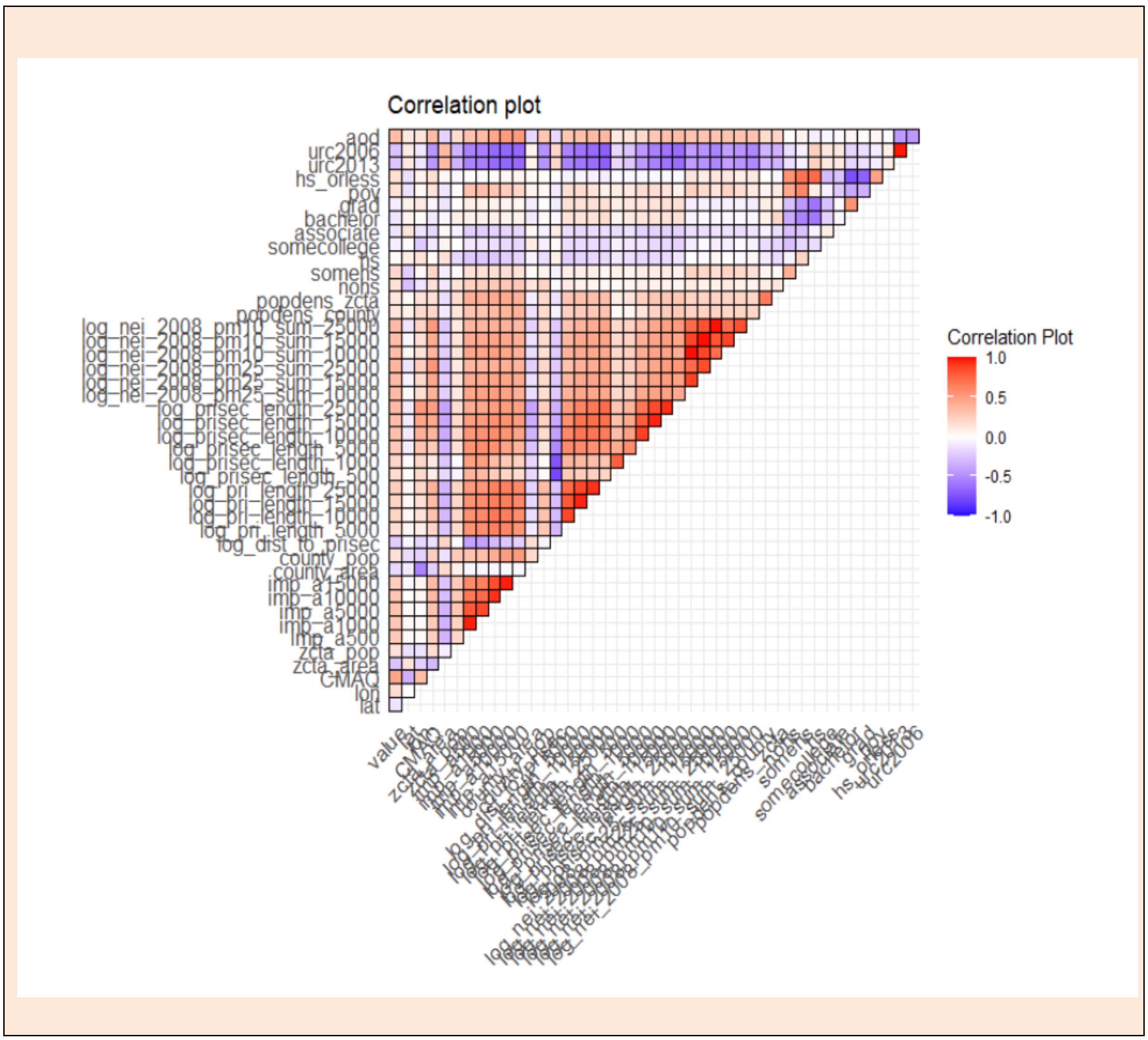
Using ggplotly library of R can check the number of stations per city. **Using Annex- 3**. To illustrate some of them

City	Count of stations
New York	9
Cleveland	6
Baltimore	5
Chicago	5
Detroit	5

4. **Conduct a correlation test and provide the results focusing on highly correlated pairs. In the correlation plot, we can see that if the correlation value is close to 1 (dark red color) then it means it has a strong correlation and if the correlation value is close to -1 (dark blue color) then it has a strong negative correlation.**

- Variables have strong correlation value as they are close to 1. To illustrate some of them are Log_pri_length_5000 and log_prisec_length_10000 shows correlation value of 87%, Log_pri_length_5000 and log_prisec_length_10000 shows correlation value of 81%, Log_pri_length_25000 and log_prisec_length_10000 shows correlation value of 86% these are the road density variables.

- imp_a500 and imp_a1000 have 97% , imp_a5000 and imp_a500 have 81% of strong correlation value. Where imp variables are development variables.
- log_pri_length_5000 and log_pri_length_1000 have 87% log_pri_length_5000, log_pri_length_15000 and log_pri_length_25000 have strong correlation. Whre log_pri are primary and secondary road length variables.
- log_nei_2008_pm25_sum_15000,log_nei_2008_pm25_sum_25000, log_nei_2008_pm10_sum_10000,log_nei_2008_pm10_sum_15000, log_nei_2008_pm25_sum_25000 have strong correlation. Where log_nei indicates emission variables.



STAGE 2 – PRE- PROCESSING THE DATASET WITH A VIEW TO DEVELOP MODELS

5. **Report on the size of both datasets. Check the proportion of cities for both datasets.**

   Split the data is to use the ratio of 2:3. And then checked the train and test dataset dimension.

```
<Analysis/Assess/Total>
<584/292/876>
> train<-training(split_df)
> dim(train)
[1] 584  50
> test<-testing(split_df)
> dim(test)
[1] 292  50
```

For the proportion of the dataset. **Used Annex- 5**

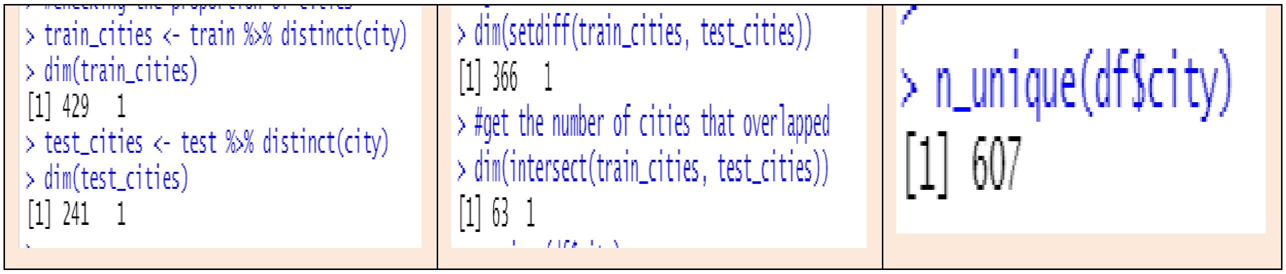
	city	n	prop
1	Akron	1	0.003424658
2	Albuquerque	1	0.003424658
3	Alton	1	0.003424658
4	Anaheim	1	0.003424658
5	Anderson	1	0.003424658
6	Annandale	1	0.003424658
7	Apache Junction	1	0.003424658
8	Apple Valley	1	0.003424658
9	Ashland	1	0.003424658
10	Atlantic City	1	0.003424658
11	Azusa	1	0.003424658
12	Bakersfield	1	0.003424658
13	Baltimore	1	0.003424658
14	Bay City	1	0.003424658
15	Bayport	1	0.003424658
16	Baytown	1	0.003424658
17	Belle Glade	1	0.003424658
18	Bend	1	0.003424658
19	Bensley	1	0.003424658
20	Birmingham	1	0.003424658

	city	n	prop
1	Aberdeen	1	0.001712329
2	Akron	1	0.001712329
3	Albany	3	0.005136986
4	Albuquerque	1	0.001712329
5	Alexandria	1	0.001712329
6	Allen Park	1	0.001712329
7	Altamont	1	0.001712329
8	Amarillo	1	0.001712329
9	Anadarko	1	0.001712329
10	Appleton	1	0.001712329
11	Arden-Arcade	1	0.001712329
12	Arlington	1	0.001712329
13	Arnold	1	0.001712329
14	Asheville	1	0.001712329
15	Ashland	1	0.001712329
16	Atascadero	1	0.001712329
17	Athens-Clarke County (Remainder)	1	0.001712329
18	Atlanta	2	0.003424658
19	Augusta-Richmond County (Remainder)	2	0.003424658
20	Aurora	1	0.001712329

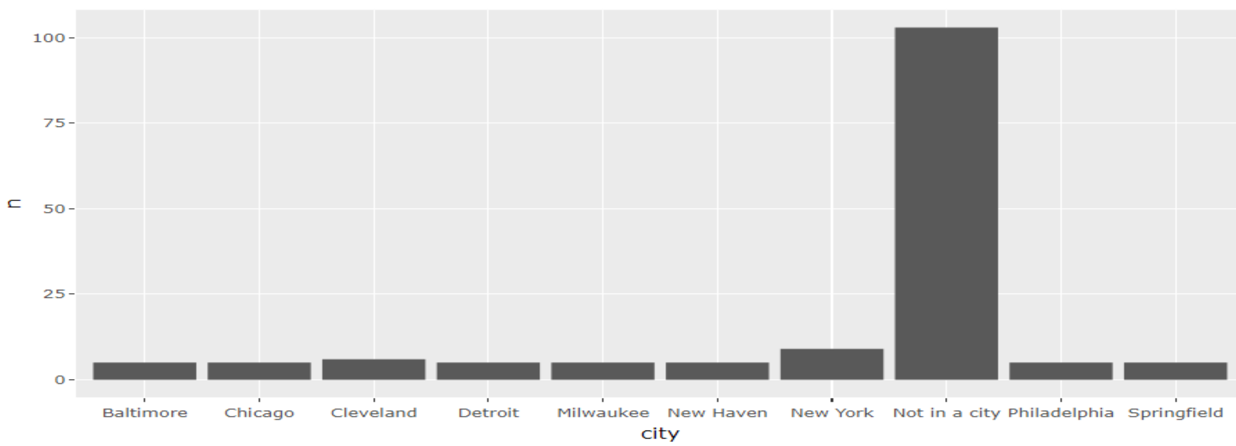
6. **Report if there is an issue with the datasets.**

   I found that there were about 366 cities that were different. Then I check intersections and how many cities are common in both and I found that 63 cities were common.

This shows that some cities were missing as the actual dataset has about 607 total distinct cities but the model shows only 366 cities. If used in prediction it will bias the result. **Annex-6**



7. **Propose a meaningful way to bin the types of cities.**



I used the case when the function in mutates of dplyr library. **Used Annex-7.**

Created a two-bin one not in the city and one in the city

Build a recipe or pipeline using the *recipes* package using the following steps on the training set:

- After doing binning of cities. Again used train and test method to split the dataset into 2:3 ratios to remove biasess from the model.
- **This dataset used value as its outcome variable as value is the most important continuous feature of the dataset. Then followed 8-15 steps as mentioned.**

8. **Given the 'value' variable the outcome role**
9. **Given the 'id' variable a role of the 'id variable'**
10. **Given the 'fips' variable a new role of the id variable that is 'county id' as it is also an id variable only**

11. **Created a dummy variable with one-hot encoding using the function step_dummy() in the recipe. Passed variables state, county, city, zcta to be converted into dummy variables using one-hot encoding.**
12. **Added step_normalize() for all predictors to normalize the data.**
13. **Have a lot of variables that have a high correlation. This issue can be solved using a recipe. Used step_corr() to remove variables with high correlation. Note exclude 'CMAQ' and 'aod' variables from correlation checking as we need these variables for prediction.**
14. **Next removed variables with nonzero variance. These are the variables that have very similar values. This can be done by adding step_nzv() in the recipe. This also excludes 'CMAQ' and 'aod' variables.**
15. **Used prep() and bake() to apply this recipe to the training and testing dataset and extract the transformed training and testing dataset. Used prep() for training dataset and retain= TRUE to extract the training dataset with bake().**

Note: Followed the same step from 8-15 on tests set and got 292 number of observation with 38 numbers of columns. Used Annex-8

GLIMPSE OF TRAIN DATASET AFTER APPLYING 8-15 STEP

```
> glimpse(preproc_train)
Rows: 584
Columns: 38
$ id                         <fct> 18003.0004, 55041.0007, 6065.1003, 39009.0003, 39061.8001, 24510.00~
$ fips                       <fct> 18003, 55041, 6065, 39009, 39061, 24510, 6061, 6065, 44003, 37111, ~
$ lat                        <dbl> 0.58882316, 1.54904648, -0.94755112, 0.23362069, 0.17739219, 0.2117~
$ lon                        <dbl> 0.42951200, 0.17907102, -1.75257100, 0.64522788, 0.47070057, 1.0050~
$ CMAQ                       <dbl> 0.62436324, -1.68207222, 0.96207702, -0.17358380, 1.09015207, 0.808~
$ zcta_area                  <dbl> -0.28622929, 0.27358825, -0.24623526, -0.10306628, -0.30418048, -0.~
$ zcta_pop                   <dbl> -0.19976846, -1.15119671, 1.05817876, -1.31892300, -1.01678287, -1.~
$ imp_a500                   <dbl> 0.24200521, -1.27986426, 0.31614745, -1.27986426, 1.41228325, -0.02~
$ imp_a15000                 <dbl> -0.05331877, -1.01475362, 0.75684362, -1.01754557, 0.60313914, 0.77~
$ county_area                <dbl> -0.342430521, -0.180011111, 2.639170133, -0.412409009, -0.456882807~
$ county_pop                 <dbl> -0.27809775, -0.54915429, 1.15879922, -0.50571553, 0.07209219, -0.0~
$ log_dist_to_prisec         <dbl> 0.29488010, 1.61107668, 0.88038866, 0.09145249, -0.32414257, 0.0513~
$ log_pri_length_5000        <dbl> -1.21698969, -1.21698969, 0.31071749, -1.21698969, 0.12970259, 0.45~
$ log_pri_length_25000       <dbl> 0.45913806, -1.91358030, 0.79623717, -1.94792448, 0.68414867, 1.289~
$ log_prisec_length_500      <dbl> -0.8042596, -0.8042596, -0.8042596, -0.8042596, 0.1920465, -0.80425~
$ log_prisec_length_1000     <dbl> 0.928627921, -1.182719540, -1.182719540, 0.353144673, -0.143654244,~
$ log_prisec_length_5000     <dbl> 0.26056031, -2.42210441, -1.47069503, -0.94131275, 0.53867297, 1.04~
$ log_prisec_length_10000    <dbl> 0.47139083, -1.35651373, -1.15057340, -1.01734984, 0.75600388, 1.65~
$ log_prisec_length_25000    <dbl> 0.2241867, -1.0114206, -0.5800285, -0.3673025, 1.1251682, 1.5413004~
$ log_nei_2008_pm10_sum_10000 <dbl> 0.24948049, -0.19987754, -0.12995847, -1.86988606, 0.67593452, 0.57~
$ log_nei_2008_pm10_sum_15000 <dbl> 0.11563799, -0.58233628, 0.14333043, -2.37106465, 0.52786902, 0.560~
$ log_nei_2008_pm10_sum_25000 <dbl> -0.121767991, -1.062726117, 0.399495219, -1.122062880, 0.326424589,~
$ popdens_county             <dbl> -0.20740018, -0.30849061, -0.25243537, -0.28577436, 0.06579996, 1.1~
$ popdens_zcta               <dbl> -0.04352436, -0.43677384, -0.11420670, -0.43763220, -0.05930112, 0.~
$ nohs                       <dbl> -0.356765228, -0.248611177, -0.437880766, -0.289168946, -0.65418886~
$ somehs                     <dbl> -0.52236149, 0.05997758, -0.64827264, 0.23310541, 0.07571648, -1.57~
$ hs                         <dbl> 0.13046097, 0.84997688, -1.02411108, 1.43562937, -0.01176892, -2.52~
$ somecollege                <dbl> 0.65283220, 0.30201273, 0.55098138, -0.16197432, 0.64151544, -2.425~
$ associate                  <dbl> 0.25455966, 0.06777774, 0.27790740, -0.93617508, 0.32460288, 15.010~
```

GLIMPSE OF TEST DATASET AFTER APPLYING 8-15 STEP

```
> glimpse(preproc_test)
Rows: 292
Columns: 38
$ id                        <fct> 1033.1002, 1055.001, 1069.0003, 1073.0023, 1073.1005, 10~
$ fips                      <fct> 1033, 1055, 1069, 1073, 1073, 1073, 1073, 1097, 1101, 11~
$ lat                       <dbl> -0.7728832, -0.9372959, -1.5320343, -1.0320050, -1.07970~
$ lon                       <dbl> 0.257320687, 0.369434553, 0.409990629, 0.313770612, 0.30~
$ CMAQ                      <dbl> 0.29998216, 0.24674242, 0.20709388, 0.57552868, 0.575528~
$ zcta_area                 <dbl> -0.28619719, -0.06873196, -0.05509092, -0.27002790, -0.0~
$ zcta_pop                  <dbl> -0.879542188, -0.269663654, 0.294153890, -0.881315896, -~
$ imp_a500                  <dbl> -0.2729472, -0.4136902, -0.2747190, 0.9170994, -1.190707~
$ imp_a15000                <dbl> -0.64497903, -0.65149607, -0.68340563, 0.27993882, -0.71~
$ county_area               <dbl> -0.37188088, -0.39811726, -0.37770611, -0.13575524, -0.1~
$ county_pop                <dbl> -0.51380669, -0.47463793, -0.47689631, -0.04063725, -0.0~
$ log_dist_to_prisec        <dbl> -0.33751301, -0.70345873, 0.65481464, 0.27951897, 0.2251~
$ log_pri_length_5000       <dbl> -1.2169897, -0.7031434, -1.2169897, 1.2520999, 1.0258320~
$ log_pri_length_25000      <dbl> -1.91968114, -0.23213427, -1.94792448, 0.65358646, 0.541~
$ log_prisec_length_500     <dbl> 1.7732562, 1.9116667, -0.8042596, -0.8042596, -0.8042596~
$ log_prisec_length_1000    <dbl> 1.56648535, 1.42781688, -1.18271954, 0.71693458, 0.55378~
$ log_prisec_length_5000    <dbl> 0.63219480, 0.57811543, 1.32022543, 1.29810472, -0.32049~
$ log_prisec_length_10000   <dbl> 0.59000435, 0.48880977, 0.80456945, 1.20199261, -0.40497~
$ log_prisec_length_25000   <dbl> -0.05308927, -0.19999493, 0.02926116, 0.91050048, -0.052~
$ log_nei_2008_pm10_sum_10000 <dbl> 1.01796555, 0.04496970, -1.46902770, 1.67300349, 0.62064~
$ log_nei_2008_pm10_sum_15000 <dbl> 0.71537341, -0.31566797, -0.67857324, 1.61236874, 0.7925~
$ log_nei_2008_pm10_sum_25000 <dbl> 0.52844464, -0.67681323, -1.13231066, 1.36225960, 0.9539~
$ popdens_county            <dbl> -0.2927645, -0.2731028, -0.2769199, -0.1975178, -0.19751~
$ popdens_zcta              <dbl> -0.27208923, -0.39980156, -0.38250454, -0.33623122, -0.4~
$ nohs                      <dbl> 0.04881246, -0.35676523, -0.15397638, 0.02177395, -0.573~
$ somehs                    <dbl> 0.90987784, 0.51640550, 0.24884431, 1.11448346, -0.53810~
$ hs                        <dbl> 0.03842986, -0.19583113, -0.02850185, 0.59061649, 0.0467~
$ somecollege               <dbl> -0.060123502, 0.879167340, -0.003539716, 0.234112183, 0.~
$ associate                 <dbl> 0.11447322, 0.69816673, 0.18451644, 0.04443000, 0.207864~
$ bachelor                  <dbl> 0.24668560, 0.50802690, 0.14989253, 0.90487850, 0.22~
```

Dimension of both dataset is

```
> dim(preproc_test)
[1] 292  38
> dim(preproc_train)
[1] 584  38
```

STAGE 3 – BUILDING MODELS FOR PREDICTION AND CLASSIFICATION

16. MODEL-1 LINEAR REGRESSION MODEL

In linear regression checked that value is the dependent which will predict the annual average air pollution concentrations of various US cities, whereas value was normalized using the recipe. **Annex-9.**

Relationship between Predictors and relationship model: In the model, the value of F =10.39 which is far greater than 1, shows that there can be a relationship between predictors and response variables (Value).

To check the significant relationship of each predictor: If the variables have p values less than 0.05, then it show that a variable have a significant relationship. And imp_a500, imp_a1500 0 and many more variables is less significant because the p-value for that is greater than 0.05. So, we can remove that variable in further findings while making a new model.

Model fit: Multiple R squared values show the variation by the model. R square value is 39% which shows that there is a 39% variation between dependent and predictors.

The residual standard error measures the deviation from the regression line which is about 2.046.

.

```
Call:
lm(formula = value ~ ., data = .)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.9656 -1.1692 -0.0339  1.1417 10.6338 

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                10.832814   0.084674 127.935  < 2e-16 ***
lat                         0.151756   0.105993   1.432   0.1528    
lon                         0.234825   0.148821   1.578   0.1152    
CMAQ                        0.744421   0.119983   6.204 1.08e-09 ***
zcta_area                  -0.216822   0.100831  -2.150   0.0320 *  
zcta_pop                    0.182828   0.096168   1.901   0.0578 .  
imp_a500                    0.096424   0.141272   0.683   0.4952    
imp_a15000                 -0.040458   0.153673  -0.263   0.7924    
county_area                -0.132194   0.111834  -1.182   0.2377    
county_pop                 -0.096712   0.118609  -0.815   0.4152    
log_dist_to_prisec          0.084679   0.161675   0.524   0.6007    
log_pri_length_5000        -0.214490   0.146453  -1.465   0.1436    
log_pri_length_25000       -0.059509   0.179908  -0.331   0.7409    
log_prisec_length_500       0.205020   0.168869   1.214   0.2252    
log_prisec_length_1000      0.089577   0.160359   0.559   0.5767    
log_prisec_length_5000      0.182296   0.230986   0.789   0.4303    
log_prisec_length_10000    -0.024576   0.299880  -0.082   0.9347    
log_prisec_length_25000     0.353718   0.247896   1.427   0.1542    
log_nei_2008_pm10_sum_10000 0.423886   0.185493   2.285   0.0227 *  
log_nei_2008_pm10_sum_15000 -0.051140  0.241512  -0.212   0.8324    
log_nei_2008_pm10_sum_25000 0.049259   0.196202   0.251   0.8019    
popdens_county              0.044704   0.122957   0.364   0.7163    
popdens_zcta               -0.006856   0.125452  -0.055   0.9564    
nohs                      -22.065112   8.715583  -2.532   0.0116 *  
```

```
> MAE_val =MAE(p1  ,preproc_test$value)
> MAE_val
[1] 1.491549
> RMSE_val = RMSE(p1  ,preproc_test$value)
> RMSE_val
[1] 2.048858
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.046 on 549 degrees of freedom
Multiple R-squared:  0.3915,	Adjusted R-squared:  0.3538 
F-statistic: 10.39 on 34 and 549 DF,  p-value: < 2.2e-16
```

Where the coefficients of the variables are as explained and provided in the table.

Where the mean absolute error is 1.49. This implies that, on average, the difference between our model's predictions and the true quality score is about 1.49.

Whereas, the Root Mean squared error is 2.04. Which tells about residuals how far they are from the regression line or best fit line.

17. MODEL-2 LINEAR DISCRIMINANT ANALYSIS

Using Annex- 10, Firstly created an AQI category from the value as provided in the link. And the variable has been linked with the recipe model for further analysis.

By using Annex-10, performed Linear Discriminant Analysis. Where there were certain variables 'id','fips','value','hs_orless','hs','nohs','log_prisec_length_10000','log_nei_2008_pm10_sum_15000' which were collinear removed from the dataset to perform the analysis.

- **Prior probabilities of groups:** it tells about the proportion of the training dataset. For example, there are 70% of the training observations have good air quality, and 29% of training observations show moderate Air quality.
- **Group means:** it tells about mean in Good and Moderate Air quality index.
- **LDA Equation:**
  LD1 = 0.072*lat + ….. -0.10*City not in a city

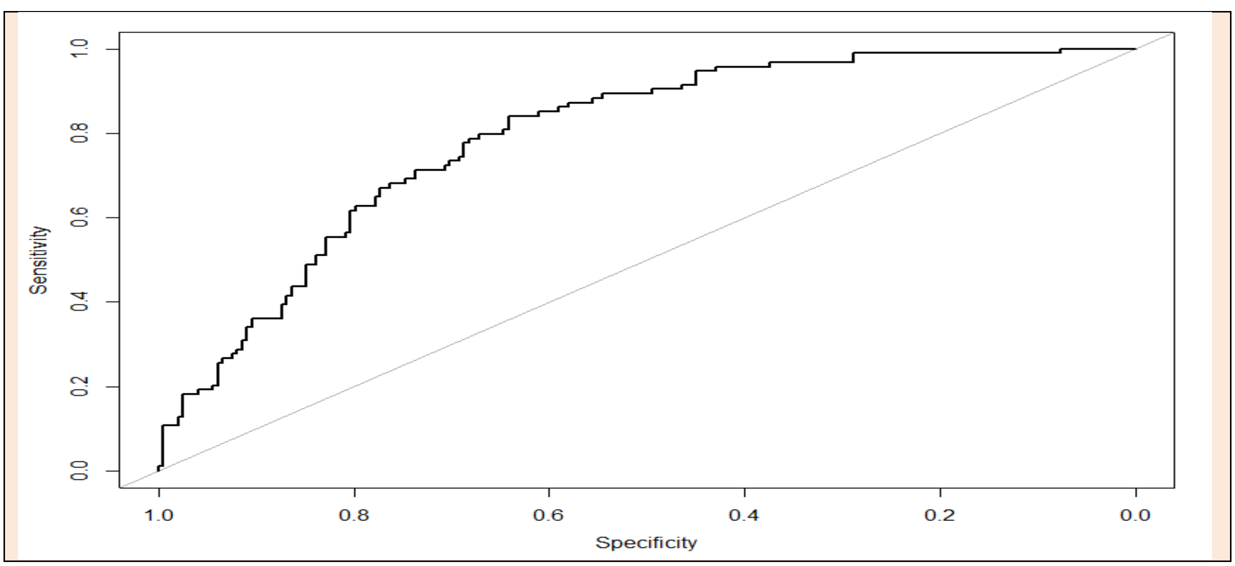
Moreover also did prediction on the test dataset and checked the confusion Matrix of the dataset and found that the accuracy rate of the model for predicting the annual average air pollution concentrations for US cities is 71%.

The ROC curve demonstrates the trade-off between sensitivity and specificity. Classifiers that produce curves closer to the top-left corner perform better. A random classifier is supposed to deliver points along the diagonal as a baseline. The closer the curve gets to the ROC space's 45-degree diagonal, the less accurate the test is.

```
Call:
lda(AQI_Category ~ ., data = .)

Prior probabilities of groups:
     Good 0 to 500 Moderate 51 to 100
         0.7089041          0.2910959

Group means:
                           lat         lon       CMAQ   zcta_area    zcta_pop   imp_a500
Good 0 to 500       0.03673946 -0.06133082 -0.1932884  0.06416932 -0.06128559 -0.1236668
Moderate 51 to 100 -0.08947138  0.14935857  0.4707141 -0.15627116  0.14924844  0.3011651
                   imp_a15000 county_area  county_pop log_dist_to_prisec
Good 0 to 500      -0.1153840  0.04145419 -0.09975457         0.05317261
Moderate 51 to 100  0.2809939 -0.10095315  0.24293172        -0.12949094
                   log_pri_length_5000 log_pri_length_25000 log_prisec_length_500
Good 0 to 500               -0.0692933           -0.1169961           -0.06827778
Moderate 51 to 100           0.1687496            0.2849199            0.16627648
                   log_prisec_length_1000 log_prisec_length_5000 log_prisec_length_25000
Good 0 to 500                 -0.08304003             -0.1306902              -0.1636623
Moderate 51 to 100             0.20222689              0.3182691               0.3985657
                   log_nei_2008_pm10_sum_10000 log_nei_2008_pm10_sum_25000 popdens_county
Good 0 to 500                       -0.2033626                  -0.1909967    -0.08117605
Moderate 51 to 100                   0.4952476                   0.4651330     0.19768754
                   popdens_zcta      somehs somecollege   associate    bachelor        grad
Good 0 to 500       -0.07214443 -0.07084722  0.03218051  0.03525163  0.02418304  0.02434455
Moderate 51 to 100   0.17569290  0.17253382 -0.07836900 -0.08584810 -0.05889280 -0.05928614
                           pov    urc2006        aod state_California city_Not.in.a.city
Good 0 to 500      -0.06697094  0.1366286 -0.1378930      -0.06601811         0.07698998
Moderate 51 to 100  0.16309394 -0.3327309  0.3358099       0.16077351        -0.18749324
```

```
                                          LD1
lat                               0.072087579
lon                               0.168191581
CMAQ                              0.429345747
zcta_area                         0.109792723
zcta_pop                         -0.010855897
imp_a500                          0.010762063
imp_a15000                       -0.225890113
county_area                       0.106964976
county_pop                        0.083163991
log_dist_to_prisec                0.435147246
log_pri_length_5000              -0.213359403
log_pri_length_25000             -0.150131241
log_prisec_length_500             0.359982086
log_prisec_length_1000            0.096981484
log_prisec_length_5000            0.209370965
log_prisec_length_25000           0.402661056
log_nei_2008_pm10_sum_10000       0.460325457
log_nei_2008_pm10_sum_25000       0.077717325
popdens_county                    0.157481000
popdens_zcta                     -0.030830155
somehs                           -0.053639905
somecollege                      -0.045481485
associate                         0.005266987
bachelor                         -0.034593243
grad                             -0.060341023
pov                               0.083392216
urc2006                           0.079158198
aod                               0.252338904
state_California                  0.467762498
city_Not.in.a.city               -0.101829183
```

```
> confusionMatrix(confusiontab)
Confusion Matrix and Statistics

                     Good 0 to 500 Moderate 51 to 100
  Good 0 to 500                174                 60
  Moderate 51 to 100            24                 34

               Accuracy : 0.7123
                 95% CI : (0.6567, 0.7636)
    No Information Rate : 0.6781
    P-Value [Acc > NIR] : 0.1163700

                  Kappa : 0.2674

 Mcnemar's Test P-Value : 0.0001341

            Sensitivity : 0.8788
            Specificity : 0.3617
         Pos Pred Value : 0.7436
         Neg Pred Value : 0.5862
             Prevalence : 0.6781
         Detection Rate : 0.5959
   Detection Prevalence : 0.8014
      Balanced Accuracy : 0.6202

       'Positive' Class : Good 0 to 500
```



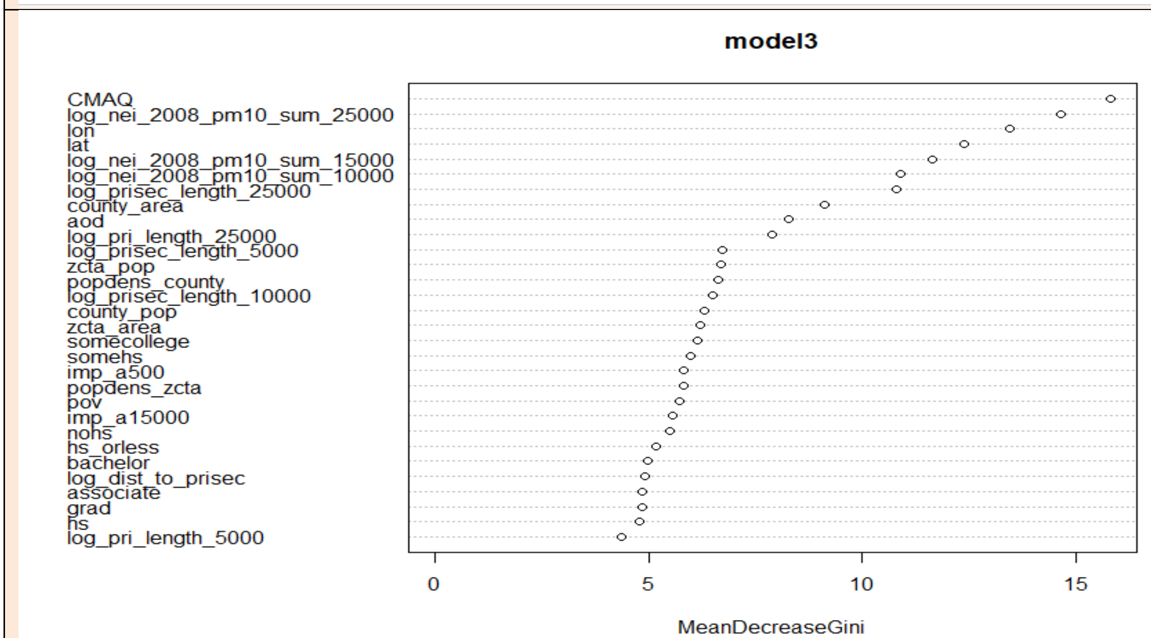
18. MODEL-3 RANDOM FOREST

Random Forest method classification technique is applied to the dataset when the response variable is categorical moreover, and also can reduce the variance by building a random forest model.

Out of box estimate error is 19.86%. And when checking the accuracy we found that the accuracy of the model is 79%.

VarImp() Plot was created to determine which variables are most significant in determining or predicting the annual average air pollution concentrations. **Using Annex-11**

```
> model3 <- preproc_train  %>% randomForest(AQI_Category~.,data=.)
> model3

Call:
 randomForest(formula = AQI_Category ~ ., data = .)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 19.86%
Confusion matrix:
                   Good 0 to 500 Moderate 51 to 100 class.error
Good 0 to 500                390                 24  0.05797101
Moderate 51 to 100            92                 78  0.54117647
>
```



```
Confusion Matrix and Statistics

                      Reference
Prediction            Good 0 to 500 Moderate 51 to 100
  Good 0 to 500                 190                 52
  Moderate 51 to 100              8                 42

               Accuracy : 0.7945
                 95% CI : (0.7436, 0.8394)
    No Information Rate : 0.6781
    P-Value [Acc > NIR] : 6.623e-06
```

Note:

In comparison to model 1 found the Accuracy rate of model-3 which is a random forest has an accuracy rate of 79%. This means the higher the accuracy of the model the better the model is for the prediction of the annual average air pollution concentrations.

Whereas, the Model-1 linear regression has about a 39% accuracy rate and the Model-2 Linear discriminant model has a 71% of accuracy rate.

After checking the RMSE of all the three models, we found that the RMSE of one model3 is lower than model1. As Root mean square criteria is that if the model has lower RMSE then it will give better performance. Hence Model3 gives better performance than Model2 and Model1.

19. Make conclusions related to pros and cons of predictive techniques and classifications.

PROS OF LINEAR DISCRIMINANT ANALYSIS

- It still beats some algorithms (logistic regression) when its assumptions are met

CONS OF LINEAR DISCRIMINANT ANALYSIS

- It is suitable when it contains two or more categories
- It requires normal distribution of the variables

PROS OF RANDOM FOREST

- It works well with the non-linear dataset
- It gives better accuracy than another classification algorithm

CONS OF RANDOM FOREST

- It is not used in linear methods
- It creates biasness when dealing with categorical variables**.**

PROS OF LINEAR REGRESSION

- It is fast than other models, not create complex calculations. Moreover works well on a large amount of dataset.
- It is the simplest linear equation that tells the relationship between dependent and independent variables.

CONS OF LINEAR REGRESSION

- The linear regression model is far too simplistic to account for real-world complexities.
- Outliers create an effect on output, as the best fit line tries to minimize mean squared error for the outliers as well.

Overall Pros:

All these modelling techniques are very helpful to know the in-depth analysis of any dataset. Moreover, it can also help in identifying the risks..

Overall Cons:

If the dataset is small, it will create a problem, as the algorithm fails to consider which variable is important.

20. Make conclusions on the *rsample* and *recipes* packages to answer the reason as to why these 2 packages are used.

Rsample Package:

- The package has been used to split the dataset into train and test. Where initial split is used to create a single binary split of the dataset into training and testing datasets.
- The initial split() function tells us what rows of data frame should be assigned to train and test. It does not split the dataset

Recipe Package

- It is used to prepare the dataset for data analysis.
- Because after splitting, we checked if the dataset can be used in the model.

- It is standardized for the sequence of steps used to pre-process the dataset.
- Creating a recipe specifies how to create a data frame of predictors - it specifies which variables to use and the pre-processing steps, but it does not execute these steps or create the data frame of predictors.

  To illustrate:

  Id is not introduced in the model because it contains county numbers and the number of particular monitoring stations assigned to them in the model will create noise.

21. Make recommendations for other experts in the data analytics field as to what lessons have been learnt to attain a high prediction rate.

Recommendation:

- The main thing is to explore the dataset before starting the analysis.

  To illustrate: which variables are significant for the dataset, details about the dataset, and also the most important thing which variable we are going to predict as it will be our output variable.
- Normalize the dataset, as most of the time; the dataset is skewed, which shows that the dataset has outliers.
- It is important to check the correlation also in the dataset. To check if there exists a correlation in any variable.

  To illustrate: If we are going to use linear regression then we can run into the problem of multi-collinearity, and we do not want redundant variables as they create noise in our dataset.
- Check the modelling techniques which suit the model best.

  To illustrate in this dataset applied three techniques linear regression, Linear Discriminant analysis. By checking the accuracy and RMSE found that random is the best method as its accuracy rate is high and has low RMSE from three models.

R-CODE

#LIBARIES

```
library(tidyverse)
library(readr)
library(plotly)
library(skimr)
library(dplyr)
library(ggcorrplot)
library("writexl")
library(tidymodels)
library(recipes)
library(vip)

df <- read_csv("pm25_data.csv")
View(df)
```

ANNEX-1:

```
#structure of data
str(df)

#checking missing value
missing <- sapply(df,function(x)sum(is.na(x)))
data_missing <- data.frame(missing)
View(data_missing)

#converting three variables into factor
df <- df %>% mutate(id = as.factor(id)) %>%
  mutate(fips = as.factor(fips)) %>%
  mutate(zcta = as.factor(zcta))
```

```
#checking again three variables
str(df)

df_skimmed <- skimr::skim(df)
df_skimmed1 <- data.frame(df_skimmed)
View(df_skimmed1)

skim(df)

#bar plot for categorical values
g <- ggplot(df, aes(x = state)) + geom_bar()
ggplotly(g)
p <- ggplot(df, aes(x = county)) + geom_bar()
ggplotly(p)
q <- ggplot(df, aes(x = city)) + geom_bar()
ggplotly(q)
```

#ANNEX-2

```
df_state <- unique(df$state)
df_state
```

#ANNEX-3

```
df_stations_per_city = df %>% count(city)
View(df_stations_per_city )

#can change slicer accordingly
g <- df_stations_per_city %>%
  arrange(desc(n)) %>%
  slice(2:10) %>%
  ggplot(., aes(x=city, y=n))+
```

```
  geom_bar(stat='identity')
ggplotly(g)
```

#ANNEX-4

```
corr <- round(cor(df %>%select_if(is.numeric),method = "pearson"), 2)
ggcorrplot(corr, method = "square", type = "upper", ggtheme = ggplot2::theme_minimal, title =
"Correlation plot",
          show.legend = TRUE, legend.title = "Correlation Plot", show.diag = FALSE,
          colors = c("blue", "white", "red"), outline.color = "Black",
          hc.order = FALSE, hc.method = "complete", lab = FALSE,
          lab_col = "black", lab_size = 4, p.mat = NULL, sig.level = 0.05,
          insig = "pch", pch = 4, pch.col = "black",
          pch.cex = 5, tl.cex = 12, tl.col = "black", tl.srt = 45,
          digits = 2)
df_cor <- data.frame(corr)
View(df_cor)
write_xlsx(df_cor,"df_cor.xlsx")
```

#<u>ANNEX-5</u>

```
#splitting the data

set.seed(123)
split_df <-initial_split(df, prop = 2/3)
split_df

train<-training(split_df)
dim(train)
test<-testing(split_df)
dim(test)

#checking the proportion of cities
```

```
train_cities <- train %>% distinct(city)
dim(train_cities)
test_cities <- test %>% distinct(city)
dim(test_cities)

a <- train %>%
  count(city) %>%
  mutate(prop = n/sum(n))
View(a)

b <- test %>%
  count(city) %>%
  mutate(prop = n/sum(n))
View(b)
```

#ANNEX-6

```
# different cities
dim(setdiff(train_cities, test_cities))

# cities that overlap
dim(intersect(train_cities, test_cities))

n_unique(df$city)
```

#ANNEX-7

```
g <- df_stations_per_city %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(., aes(x=city, y=n))+
  geom_bar(stat='identity')
ggplotly(g)
```

```
df_bin <- df %>%
  mutate(city = case_when(city == "Not in a city" ~ "Not in a city",
                          city != "Not in a city" ~ "In a city"))
str(df_bin)
```

#**ANNEX-8 TRAIN SET AND TEST**

```
set.seed(1234)
df_bin_split <-initial_split(df_bin, prop = 2/3)
df_bin_split

df_bin_train <-training(df_bin_split)
df_bin_test <-testing(df_bin_split)
View(df_bin_test)

library(recipes)

rec <-df_bin_train %>%
  recipe(value~.) %>%
  update_role(everything(), new_role = "predictor") %>%
  update_role(value, new_role = "outcome role") %>%
  update_role(id, new_role = "id variable") %>%
  update_role("fips", new_role = "county id") %>%
  step_dummy(state, county, city, zcta, one_hot = TRUE) %>%
  step_corr(all_numeric(), - CMAQ, - aod)%>%
  step_nzv(all_numeric(), - CMAQ, - aod) %>%
  step_normalize(all_predictors())

a <- prep(rec, retain = TRUE)
```

```
preproc_train <- bake(a, new_data = NULL)
glimpse(preproc_train)

preproc_test<- bake(a, new_data = df_bin_test)
glimpse(preproc_test)

dim(preproc_train)
dim(preproc_test)
```

#**<u>ANNEX-8 TEST SET</u>**

```
library(recipes)
rec_test <- df_bin_test %>%
  recipe(value~.) %>%
  update_role(everything(), new_role = "predictor") %>%
  update_role(value, new_role = "outcome role") %>%
  update_role(id, new_role = "id variable") %>%
  update_role("fips", new_role = "county id") %>%
  step_dummy(state, county, city, zcta, one_hot = TRUE) %>%
  step_corr(all_numeric(), - CMAQ, - aod)%>%
  step_nzv(all_numeric(), - CMAQ, - aod) %>%
  step_normalize(all_predictors())


a_test <- prep(rec_test, retain = TRUE)

#preproc_test <- bake(a_test, new_data = NULL)
#glimpse(preproc_test)

dim(preproc_test)
```

#ANNEX-9 LINEAR REGRESSION MODEL

```
model1 <- preproc_train %>% dplyr::select(-c('id','fips')) %>%
  lm(value~.,data=.)
summary(model1)

p1 <- predict(model1, preproc_test)
p1

MAE_val =MAE(p1  ,preproc_test$value)
MAE_val
RMSE_val = RMSE(p1  ,preproc_test$value)
RMSE_val
```

#ANNEX-10 LINEAR DISCRIMINANT MODEL

```
df_bin_train <- df_bin_train %>% mutate(AQI_Category= case_when(value >= 150.5 ~ 'Very
unhealthy 201 to 300',
                                                            value >= 55.5 ~ 'Unhealthy 151 to 200',
                                                            value >= 35.5 ~ 'Unhealthy for sensitive group 101 to
150',
                                                            value >= 12.1 ~ 'Moderate 51 to 100',
                                                            TRUE ~ 'Good 0 to 500'
))
df_bin_train = df_bin_train %>% mutate(AQI_Category= as.factor(AQI_Category))

df_bin_test <- df_bin_test %>% mutate(AQI_Category= case_when(value >= 150.5 ~ 'Very
unhealthy 201 to 300',
                                                          value >= 55.5 ~ 'Unhealthy 151 to 200',
                                                          value >= 35.5 ~ 'Unhealthy for sensitive group 101 to
150',
                                                          value >= 12.1 ~ 'Moderate 51 to 100',
                                                          TRUE ~ 'Good 0 to 500'
))
```

```
df_bin_test= df_bin_test %>% mutate(AQI_Category= as.factor(AQI_Category))

preproc_train$AQI_Category <- df_bin_train$AQI_Category
preproc_test$AQI_Category <- df_bin_test$AQI_Category


library("MASS")
model2 <- preproc_train %>% dplyr::select(-
c('id','fips','value','hs_orless','hs','nohs','log_prisec_length_10000',
                               'log_nei_2008_pm10_sum_15000')) %>%
  lda(AQI_Category~.,data=.)
model2

prob <- predict(model2,preproc_test,type="response")
prob
mean(prob$class ==preproc_test$AQI_Category)

confusiontab <- table(prob$class,preproc_test$AQI_Category)
sum(diag(confusiontab))/sum(confusiontab)
library(caret)
confusionMatrix(confusiontab)

library(pROC)
roc(preproc_test$AQI_Category ~ prob$x) %>% plot(asp = NA)
```

#ANNEX-11 RANNDOM FOREST MODEL

```
library(randomForest)
preproc_train$id <- NULL
preproc_train$fips <- NULL
preproc_train$value <- NULL
```

```
model3 <- preproc_train  %>% randomForest(AQI_Category~.,data=.)
model3

importance(model3)
varImpPlot(model3)

predValid <- predict(model3, preproc_test)
confusionMatrix(predValid,preproc_test$AQI_Category)
```

References:

1. Understand *rsample* package:
   https://www.rdocumentation.org/packages/rsample/versions/0.0.5/topics/initial_split

2. Understand *recipes* package:
https://www.rdocumentation.org/packages/recipes/versions/0.1.17/topics/recipe
https://recipes.tidymodels.org/reference/

3. Understand the usage of random forest:
https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest

THANK-YOU