Vaamini Mathimaran - 231179486
Ap23210@qmul.ac.uk

Practical Technique for Data Science – Coursework 1

Part 1- Delhi Data set using R

Exploring the Data

I downloaded the file and converted it into a dataframe using 'read.csv'. I then used 'summary' to describe the dataset. For the date column, it provided the length, class, and mode, while for the pollutant columns, it displayed the column name, minimum value, first quartile, median, mean, third quartile, and maximum value. I also used 'class' to confirm the data type, which was a 'data.frame', and 'str' to examine the structure of the dataset. The output showed that the dataset contained 18,776 observations of 9 variables. It also indicated that the date column was of character type, while the rest (pollutants) were numerical.

Additionally, I checked the number of rows using 'nrow', which returned 18,776, and the number of columns using 'ncol', which confirmed there were 9. Finally, I checked for missing values using 'any(is.na(dt))', which returned FALSE, indicating that there were no missing values. This ensures that data manipulation and formatting will not be affected.

Histogram

I created histograms for the numerical variables in the dataset. To do this, I first created an empty list and then used a 'for' loop to iterate through the columns. To ensure that only numerical columns were used, I applied an 'if' statement to check their data type. I adjusted the number of bins using the Freedman-Diaconis method.

For each histogram, I set the title to the corresponding pollutant and labeled the x-axis accordingly, ensuring that the unit of measurement was also included since all pollutants share the same unit. I then overlaid some histograms to allow for direct comparisons. From the 'summary' function, I observed that NO, NO_2, O_3, SO_2, and NH_3 had similar ranges. However, by analyzing the histograms, I was able to further categorize them into two groups: NH_3, NO_2, and SO_2 in one group, and NO and O_3 in another. Additionally, both the summary statistics and the histogram shapes indicated that $PM_{2.5}$ and PM_{10} had similar distributions, so I overlaid their histograms for a clearer comparison.

Boxplot

I created boxplots for the numerical variables in a similar manner to the histograms, using a 'for' loop to iterate through the columns and an 'if' statement to filter for numerical data. I ensured that the y-axis was labeled with the respective pollutant name and unit.

The boxplots revealed that all pollutants had a significant number of outliers, with NO having the highest number of outliers.

Correlation

To analyze the correlation between pollutants, I first created a new dataset by removing the non-numeric date column. I then used the 'cor' function with the Pearson method to compute a correlation matrix. Although this provided correlation values, it was difficult to interpret at a glance. To visualize the relationships more effectively, I generated a heatmap using 'corrplot', adjusting the color palette so that positive correlations appeared in green, neutral values in white, and negative correlations in pink.

I also created scatter plots for pollutant pairs with correlation values greater than 0.9 or lower than -0.9. I used 'ggplot2' to plot the points and added a linear trend line for comparison.

Table

I generated a table displaying key statistical metrics for the numerical variables, including the mean, median, mode, standard deviation, and coefficient of variation. While R provides built-in functions for most of these metrics, it does not have a direct mode function. To calculate the mode, I wrote a custom function that identified unique values, counted their frequencies, and returned the most frequently occurring value.

To improve readability, I formatted the table using the 'kable' function, making it easier to interpret.

Sampling

For sampling, I selected $PM_{2.5}$ as the pollutant of interest. I first calculated its mean and then applied bootstrapping using a custom function. I plotted the bootstrap distribution as a histogram and recorded the summary statistics. Additionally, I created a Q-Q plot to assess normality.

Next, I calculated the confidence intervals for the mean at 68% and 95% using 'boot.ci'. To compare these values, I also calculated the confidence intervals for the mean using the t-distribution. The results were very similar, so I added the 95% confidence interval from 'boot.ci' as a reference line on the histogram.

I then repeated the same process for standard deviation, applying bootstrapping, generating a histogram and Q-Q plot, and recording the results. I calculated confidence intervals at 68% and 95% using 'boot.ci' and compared them with those obtained from the chi-squared distribution. Once again, the confidence intervals were very similar. Finally, I added the 95% confidence interval line to the histogram for visualization.

Part 2 – London Dataset using Python

Building the Dataset

My dataset for London was divided into two CSV files, so I saved both as datasets and converted the dates into a datetime object. I then removed any missing values and combined both datasets using 'concat'.

To create a rectangular dataset where each row represents the hourly average measurement of each pollutant, I used 'pivot_table', which transforms the data into a pivot table while keeping it in a dataframe format. I set 'ReadingDateTime' as the index and used the different pollutant species as columns. After that, I dropped any remaining missing values and reset the index. Additionally, I created a dataset containing only numerical values using 'select_dtype'. To gain a better understanding of the data, I used the 'describe' function.

Histogram

To create histograms of the numerical quantities recorded in the dataset, I created two lists—one containing the names of the columns with only numerical values and another empty list to store bin widths. I then used a 'for' loop to iterate through the columns and applied the Freedman-Diaconis method to determine the number of bins, storing the results in the list.

Next, I plotted the histogram for each pollutant, ensuring that the number of bins varied based on the Freedman-Diaconis method. I also set the title and x-axis label according to the pollutant. To analyze similarities, I overlaid some histograms and identified two distinct groups:

- Group 1: FINE, PM1, PM10, and PM2.5
- Group 2: NO, NO2, and TSP

Boxplot

I created boxplots for the numerical quantities in the dataset using the same approach as the histograms—a 'for' loop iterating through all numerical columns. This time, I used the 'boxplot' function and modified the titles to include the pollutant names.

The boxplots revealed that all pollutants contained many outliers, with NO having the most extreme outliers once again.

Correlation

I calculated the correlation between numerical variables by computing a Pearson correlation matrix using the 'corr' function with 'pearson' as the method.

To visualize the correlations, I created a heatmap using the 'heatmap' function from Seaborn and added a title. Instead of computing scatter plots for each pollutant individually, I used the 'pairplot' function, which automatically generates pairwise scatter plots between all numerical variables.

Yearly Comparison

The dataset contained data from 2022 to 2024, so I created separate datasets for each year by filtering using the 'loc' function.

To compare the data effectively, I determined that the best approach was to create histograms for each pollutant while overlaying the distributions for each year. To accomplish this, I reused the histogram code from earlier but modified it to include the 'hist' function two additional times to plot all three years together.

A/B Testing

I conducted an A/B test between FINE and PM2.5 because their Pearson correlation was high, and their histograms appeared similar.

To perform the test, I first determined the length of available data for FINE and PM2.5 in 2022 and 2024. Then, I created separate datasets for FINE (including only 2022 and 2024 data) and PM2.5, resetting their indexes. I merged them into a single dataframe using 'concat'.

Next, I wrote a function for a permutation test and created two empty lists to store values. I ran the test 1,000 times and generated a histogram using the results.

Finally, I estimated the p-value, which was 0.003, indicating that the observed similarity between FINE and PM2.5 was likely coincidental rather than causational. To further validate this, I performed a two-sample t-test, which resulted in a t-statistic of 2.92 and a p-value of 0.0035. Since this is less than 0.05, we reject the null hypothesis, suggesting that despite their visual similarities in the histogram and correlation analysis, their means are significantly different.