

Air Quality Analysis Report

Author: Malik Haroon

Module: Advanced Statistical Programming

Module code: ACCA7022

1. Introduction

The report analyzed the UCI Air Quality data through R. The main one was to explore the data, visualize important trends and relationships, and quantify the correlations between pollutants and to perform statistical tests comparing the levels of pollutants in various environmental conditions.

2. Exploratory Visualisation

There were three specific visualizations that were developed to examine various aspects of the data:

1. **Time Serial Plot:** The most suitable plot to use in order to see the trend of Carbon Monoxide (CO) and Benzene (C₆H₆) was a line plot. This is a clear indication of changes in the concentration of pollutants, where both seem to have a common trend implying the existence of a common source or mechanism of formation. The plot is directly related to the Time Trends picture that is given in the results.

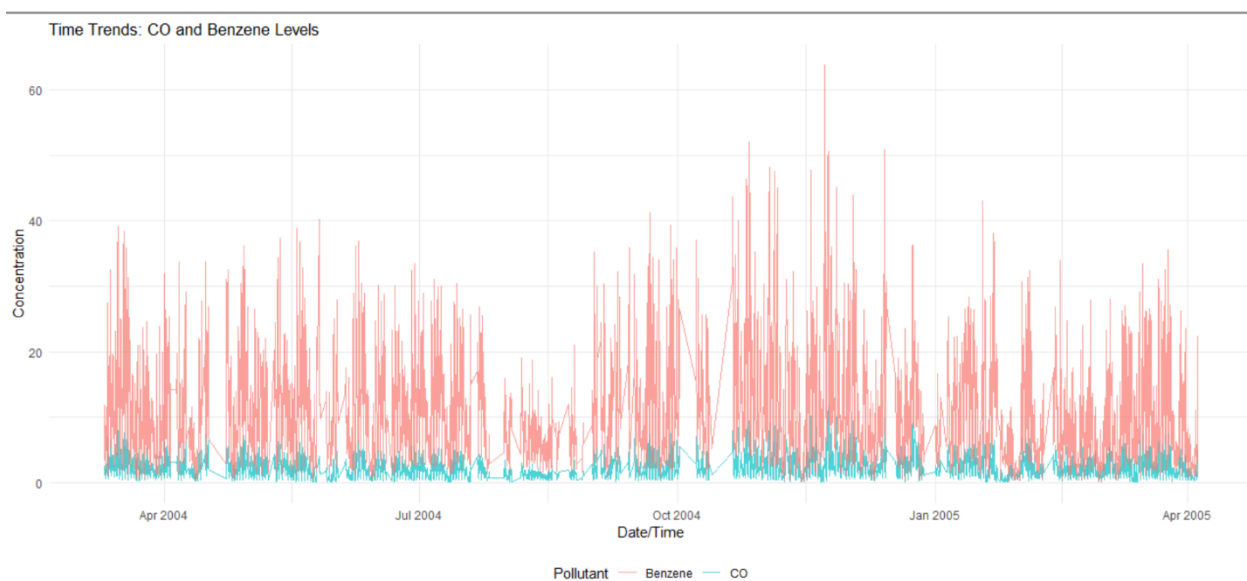


Figure 1: Time Trends: CO and Benzene Levels

2. **Scatter Plot with Regression:** A scatter plot was the most evident one to investigate the correlation between temperature and CO levels. They were useful in identifying the overall direction of the regression by adding a regression line (`geom_smooth`) and identifying a third dimension of analysis by colouring the points with relative humidity. Not only did this indicate the relationship between temperature and CO, but indicated that the relationship could be altered by humidity.

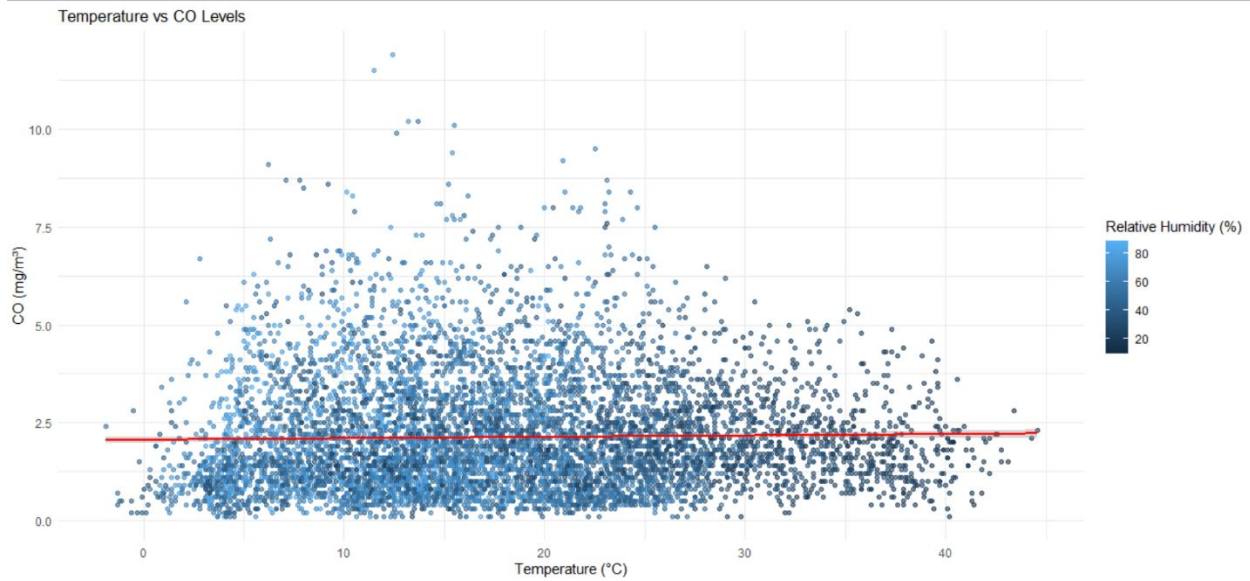


Figure 2: Temperature vs CO Levels

3. **Boxplot:** Boxplots were useful when analyzing the CO level at different humidity levels. As shows in Figure 3, 4 and 5 they present readily median values, spread, and outliers, and comparisons of groups are readily done. My results are presented in the form of a summary statistics table which is an immediate reflection of the information visualized by the boxplots, the five-number summary of each group.

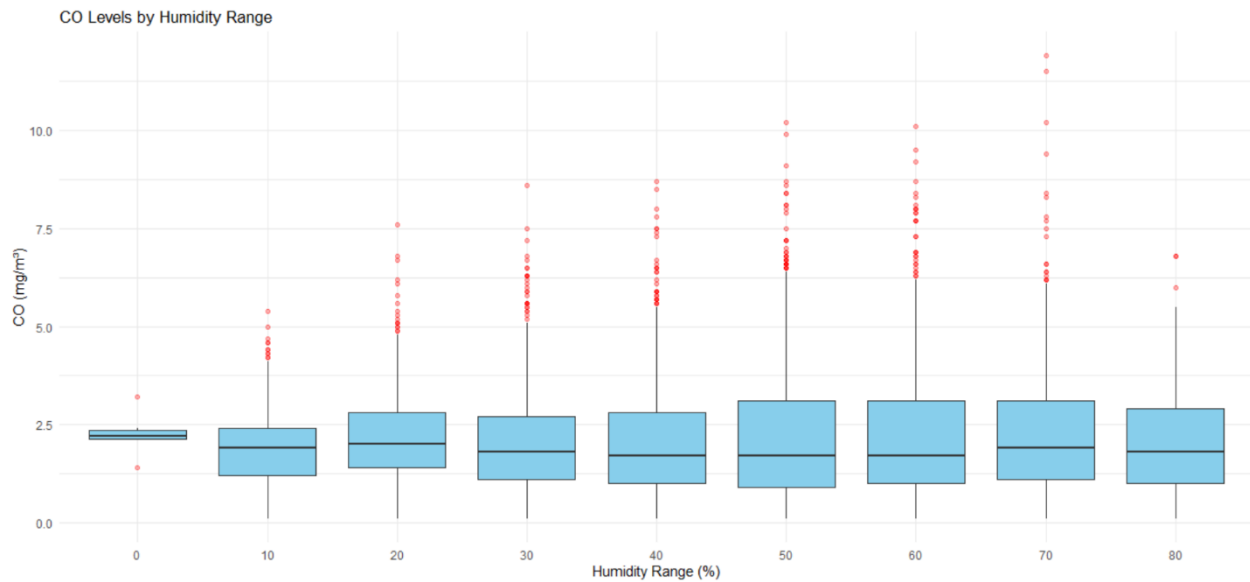


Figure 3: CO Levels by Humidity Range

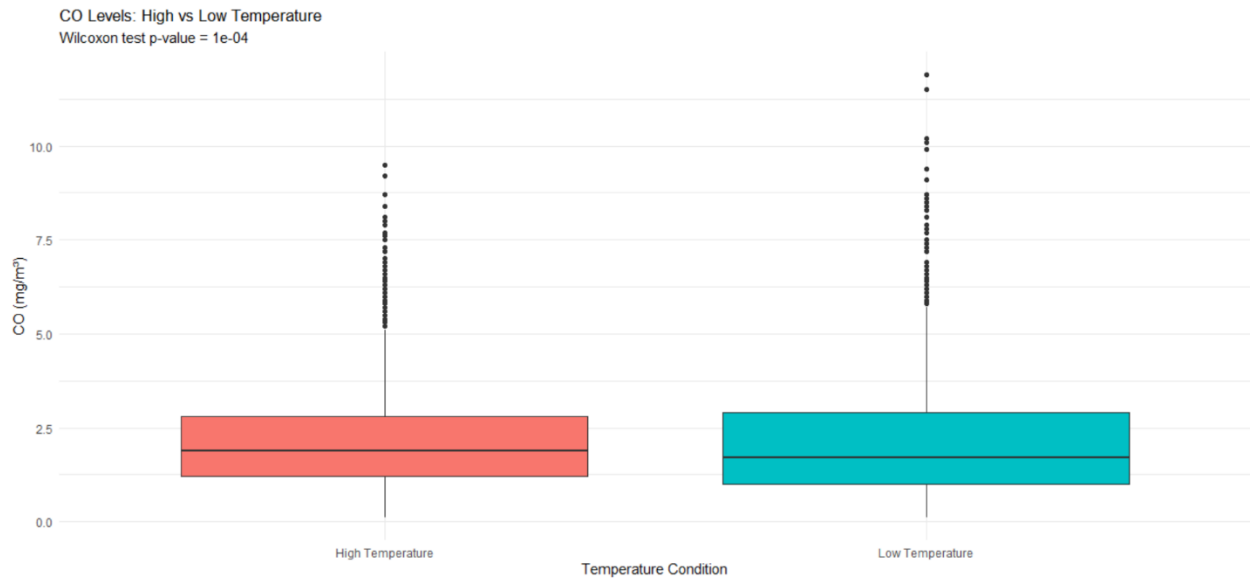


Figure 4: CO Levels: High vs Low Temperature

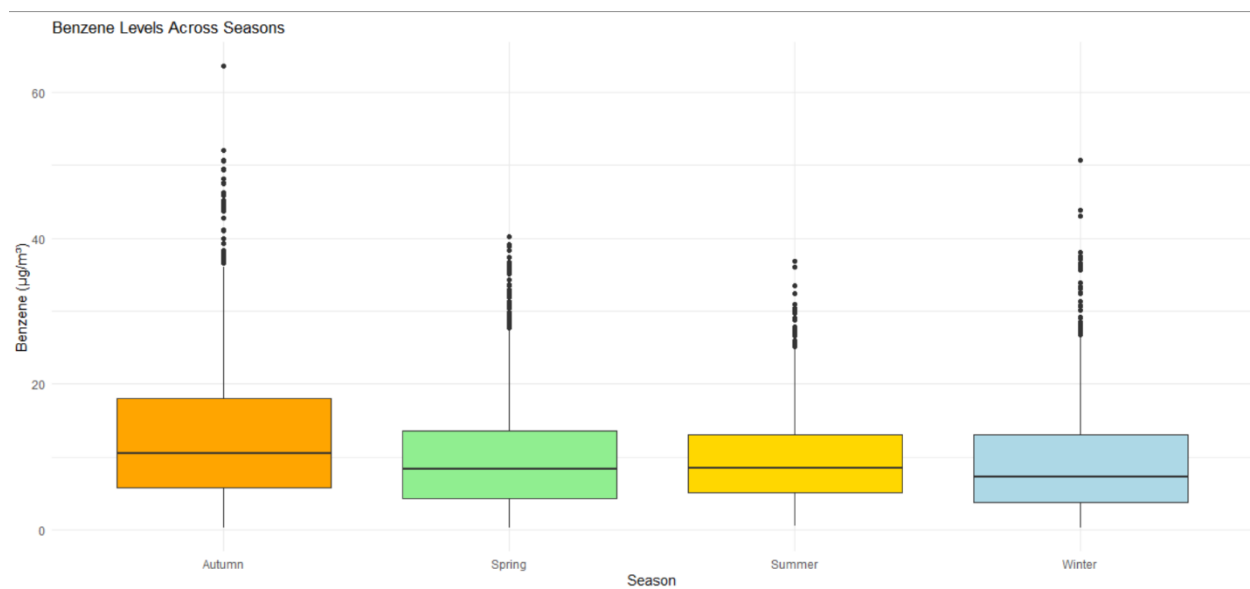


Figure 5: Benzene Levels Across Seasons

3. Challenges and Things Learned

1. **Issues with Data Import:** The data was stored in an unusual format -semicolon separators, comma use as a decimal point and -200 indicating the absence of a value. At first, I could not find right imports before I took my time in reading the documentation.
2. **Missing Values:** It was hard to deal with -200 values and whether to convert them to NA or not and whether or not to drop or keep incomplete observations. I realised that missing

data manipulation should reflect the situation of the analysis- time series analysis was the right choice to remove rows, but other types of analysis have other ideas.

3. **Statistical Test Choice:** I was going to use a t-test at first to compare the levels of CO in terms of temperatures. Nevertheless, upon examination of normality tests, i.e. Shapiro Wilk tests, it found the data to be non-normal.
4. **Code Resilience:** Fallback code Handwriting fallback code in case of missing columns (such as wind direction) made me conscious of the possibility of inconsistent data. This is one of the defensive programming methods that make the running of analyses to be smooth even when the data is not perfect.

4. Improvements for Future Work

In case of repeating this analysis, the first improvement is to begin with, use of time series decomposition to formally decompose trend, seasonal and random factors instead of using a simple visual examination. Second, to create predictive models to determine the level of pollutants using machine learning depending on the conditions in the environment.

5. Generative AI

I applied AI support to solve code bugs, learn how to use the parameters of a certain function and to clarify the clarity of my explanations. Every code was checked and comprehended prior to inclusion.

6. Annexure – R Code Snap Shots

```
=== TASK 1: DATA OVERVIEW ===
> cat("\nColumn Names:\n")

Column Names:
> print(names(AirQuality))
 [1] "Date"      "Time"      "CO.GT."    "PT08.S1.CO."
 [5] "NMHC.GT." "C6H6.GT."  "PT08.S2.NMHC." "NOx.GT."
 [9] "PT08.S3.NOx." "NO2.GT."   "PT08.S4.NO2." "PT08.S5.O3."
[13] "T"         "RH"        "AH"        "DateTime"
>
> cat("\nData Types:\n")

Data Types:
> print(str(AirQuality))
'data.frame':   9471 obs. of  16 variables:
 $ Date       : chr  "10/03/2004" "10/03/2004" "10/03/2004" "10/03/2004" ...
 $ Time       : chr  "18.00.00" "19.00.00" "20.00.00" "21.00.00" ...
 $ CO.GT.     : num  2.6 2 2.2 2.2 1.6 1.2 1.2 1 0.9 0.6 ...
 $ PT08.S1.CO.: int  1360 1292 1402 1376 1272 1197 1185 1136 1094 1010 ...
 $ NMHC.GT.   : int  150 112 88 80 51 38 31 31 24 19 ...
 $ C6H6.GT.   : num  11.9 9.4 9 9.2 6.5 4.7 3.6 3.3 2.3 1.7 ...
```

```

> cat("Rows:", nrow(AirQuality), "\n")
Rows: 9471
> cat("Columns:", ncol(AirQuality), "\n")
Columns: 16
>
> # Basic descriptive statistics for numeric columns
> cat("\nDescriptive Statistics:\n")

Descriptive Statistics:
> # Select only numeric columns for summary
> numeric_cols <- AirQuality[, apply(AirQuality, is.numeric)]
> print(summary(numeric_cols))
      CO.GT.      PT08.S1.CO.      NMHC.GT.      C6H6.GT.
Min.   : 0.100   Min.   : 647   Min.   : 7.0   Min.   : 0.10
1st Qu.: 1.100   1st Qu.: 937   1st Qu.: 67.0   1st Qu.: 4.40
Median : 1.800   Median :1063   Median : 150.0   Median : 8.20
Mean    : 2.153   Mean    :1100   Mean    : 218.8   Mean    :10.08
3rd Qu.: 2.900   3rd Qu.:1231   3rd Qu.: 297.0   3rd Qu.:14.00
Max.    :11.900   Max.    :2040   Max.    :1189.0   Max.    :63.70
NA's    :1797    NA's    :480    NA's    :8557    NA's    :480
PT08.S2.NMHC.   NOx.GT.      PT08.S3.NOx.      NO2.GT.
Min.   : 383.0   Min.   : 2.0   Min.   : 322.0   Min.   : 2.0
1st Qu.: 734.5   1st Qu.: 98.0   1st Qu.: 658.0   1st Qu.: 78.0
Median : 909.0   Median : 180.0   Median : 806.0   Median :109.0
Mean    : 939.2   Mean    : 246.9   Mean    : 835.5   Mean    :113.1

#####
# TASK 2: Exploratory Visualization
#####

cat("\n== TASK 2: EXPLORATORY VISUALIZATION ==\n")

== TASK 2: EXPLORATORY VISUALIZATION ==

# Data preprocessing for time series analysis
# Remove the last two columns (X and X.1) which are all NA
AirQuality <- AirQuality[, 1:15]

# Create proper datetime column
AirQuality$DateTime <- as.POSIXct(paste(AirQuality$Date, AirQuality$Time),
                                   format = "%d/%m/%Y %H.%M.%S")

# Remove rows with missing datetime or pollutant values
# First, convert -200 values (used as missing value indicator) to NA
AirQuality[AirQuality == -200] <- NA

# Create clean dataset
AirQuality_clean <- AirQuality %>%
  filter(!is.na(DateTime)) %>%
  filter(!is.na(CO.GT.), !is.na(C6H6.GT.), !is.na(T)) %>%

```

```

=== TASK 3: CORRELATION ANALYSIS ===
>
> # Select pollutants for correlation analysis
> pollutants <- AirQuality_clean %>%
+   select(CO.GT., PT08.S1.CO., C6H6.GT., PT08.S2.NMHC., NOx.GT., PT08.S3.NOx.)
>
> # Calculate correlation matrix
> cor_matrix <- cor(pollutants, use = "complete.obs")
>
> # Display correlation matrix
> cat("\nCorrelation Matrix:\n")

Correlation Matrix:
> print(round(cor_matrix, 3))
      CO.GT. PT08.S1.CO. C6H6.GT. PT08.S2.NMHC. NOx.GT. PT08.S3.NOx.
CO.GT.    1.000    0.877    0.930    0.914    0.786    -0.701
PT08.S1.CO. 0.877    1.000    0.877    0.886    0.708    -0.763
C6H6.GT.    0.930    0.877    1.000    0.983    0.718    -0.726
PT08.S2.NMHC. 0.914    0.886    0.983    1.000    0.705    -0.782
NOx.GT.     0.786    0.708    0.718    0.705    1.000    -0.662
PT08.S3.NOx. -0.701   -0.763   -0.726   -0.782   -0.662    1.000

=== TASK 4: HYPOTHESIS TESTING (TWO-GROUP COMPARISON) ===
>
> # Compare CO levels based on temperature (high vs low)
> # Create temperature categories (median split)
> median_temp <- median(AirQuality_clean$T, na.rm = TRUE)
> AirQuality_clean$Temp_Category <- ifelse(AirQuality_clean$T > median_temp,
+                                           "High Temperature", "Low Temperature")
>
> # Extract CO levels for each group
> co_low_temp <- AirQuality_clean$CO.GT.[AirQuality_clean$Temp_Category == "Low"]
> co_high_temp <- AirQuality_clean$CO.GT.[AirQuality_clean$Temp_Category == "High"]
>
> # Remove NA values
> co_low_temp <- co_low_temp[!is.na(co_low_temp)]
> co_high_temp <- co_high_temp[!is.na(co_high_temp)]
>
> # Check normality assumption (using sample due to large dataset)
> set.seed(123) # for reproducibility
> shapiro_low <- shapiro.test(sample(co_low_temp, min(500, length(co_low_temp))))
> shapiro_high <- shapiro.test(sample(co_high_temp, min(500, length(co_high_temp))))
>
> cat("\nNormality Test Results:\n")

```

Post-hoc Test (Tukey HSD):

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = C6H6.GT. ~ Season, data = seasonal_data)

\$Season		diff	lwr	upr	p adj
Spring-Autumn	-2.94249958	-3.5488067	-2.33619247	0.0000000	
Summer-Autumn	-3.03809123	-3.7012731	-2.37490940	0.0000000	
Winter-Autumn	-3.58979264	-4.2364460	-2.94313925	0.0000000	
Summer-Spring	-0.09559165	-0.7056057	0.51442236	0.9779224	
Winter-Spring	-0.64729306	-1.2392961	-0.05529005	0.0256144	
Winter-Summer	-0.55170141	-1.2018317	0.09842887	0.1287047	

```
>
> # Check normality of residuals
> residuals <- residuals(anova_result)
> shapiro_resid <- shapiro.test(sample(residuals, min(500, length(residuals))))
> cat("\nNormality of Residuals (Shapiro-Wilk test):\n")
```

Normality of Residuals (Shapiro-Wilk test):
