

Air Pollution Analysis – Port Talbot Hourly AQ

Load Required Libraries

```
library(tidyverse)
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2     3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr       1.0.2
## --- Conflicts ----- tidyverse_conflicts() ---
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)
library(ggplot2)
# TASK 1: IMPORT DATA + RENAME HEADERS + DISPLAY DATAFRAME (5 marks)
# Import CSV (must be in same folder as this script)
PortTalbotAQ <- read.csv("/Users/mac/Desktop/ASP/Anand/Port_Talbot_Hourly_AQ.csv",
                        header = TRUE,
                        check.names = FALSE) # keeps original column names

# Fix empty / missing header names (""")
names(PortTalbotAQ)[names(PortTalbotAQ)==" " | is.na(names(PortTalbotAQ))] <-
  paste0("Unnamed_", seq(sum(names(PortTalbotAQ)==" " | is.na(names(PortTalbotAQ))))))

# Remove redundant "Status" columns and rename pollutant variables to shorter readable names
PortTalbotAQ <- PortTalbotAQ %>%
  select(-starts_with("Status")) %>%
  rename(
    Ozone = "Ozone",
    NO = "Nitric oxide",
    NO2 = "Nitrogen dioxide",
    NOx = "Nitrogen oxides as nitrogen dioxide",
    SO2 = "Sulphur dioxide",
    CO = "Carbon monoxide",
    PM10 = "PM10 particulate matter (Hourly measured)",
    PM25 = "PM2.5 particulate matter (Hourly measured)",
    WindDirection = "Modelled Wind Direction",
    WindSpeed = "Modelled Wind Speed",
    Temp = "Modelled Temperature"
  ) %>%
  mutate(across(
```

```

.col = c(Ozone, NO, NO2, NOx, SO2, CO, PM10, PM25, WindSpeed, Temp),
.fns = ~ as.numeric(na_if(.x, "No data")) # convert "No data" to NA
))

# Create datetime field (combining Date + Time)
PortTalbotAQ$DateTime <- dmy_hms(paste(PortTalbotAQ$Date, PortTalbotAQ$Time))

# Display dataframe (as required in Task 1)
print(head(PortTalbotAQ))

##      Date      Time      Ozone      NO      NO2      NOx      SO2      CO      PM10
## 1 01-01-2020 01:00:00 32.39686 0.65250 12.26557 13.26605 0.44178 0.070977 25.1
## 2 01-01-2020 02:00:00 27.07500      NA      NA      NA 0.70684 0.086749 30.6
## 3 01-01-2020 03:00:00 34.72518 0.46059 12.30813 13.01435 0.92773 0.106465 27.6
## 4 01-01-2020 04:00:00 39.21551 0.40301  7.24397  7.86191 0.66267 0.073934 23.6
## 5 01-01-2020 05:00:00 36.12217 0.20151  7.33085  7.63982 0.39760 0.065062 22.9
## 6 01-01-2020 06:00:00 33.37808 0.40301  8.88742  9.50536 0.46387 0.065062 20.9
##      PM25      WindDirection      WindSpeed      Temp      DateTime
## 1 21.1      96.7      4.6      3.0 2020-01-01 01:00:00
## 2 25.0     101.2      5.3      3.1 2020-01-01 02:00:00
## 3 25.9      98.9      5.1      3.1 2020-01-01 03:00:00
## 4 19.8     103.5      4.8      3.2 2020-01-01 04:00:00
## 5 20.2      97.4      4.4      3.4 2020-01-01 05:00:00
## 6 15.9       90      4.3      3.6 2020-01-01 06:00:00

# TASK 2: CALCULATE PERCENTAGE OF INVALID (MISSING) DATA PER POLLUTANT (8 marks)
# Calculate percentage of missing values for each pollutant
invalidData <- PortTalbotAQ %>%
  summarise(
    across(c(Ozone, NO2, SO2, PM10, PM25), ~ mean(is.na(.x)) * 100)
  )

cat("\n===== Task 2: Percentage of Invalid / Missing Data =====\n")

##
## ===== Task 2: Percentage of Invalid / Missing Data =====
print(invalidData)

##      Ozone      NO2      SO2      PM10      PM25
## 1 1.98467 3.979606 2.395292 6.538005 6.86194

# Calculate missing value COUNT instead of percentage (for plotting)
missingCounts <- PortTalbotAQ %>%
  summarise(across(c(Ozone, NO2, SO2, PM10, PM25), ~ sum(is.na(.x)))) %>%
  pivot_longer(cols = everything(), names_to = "Pollutant", values_to = "MissingCount")

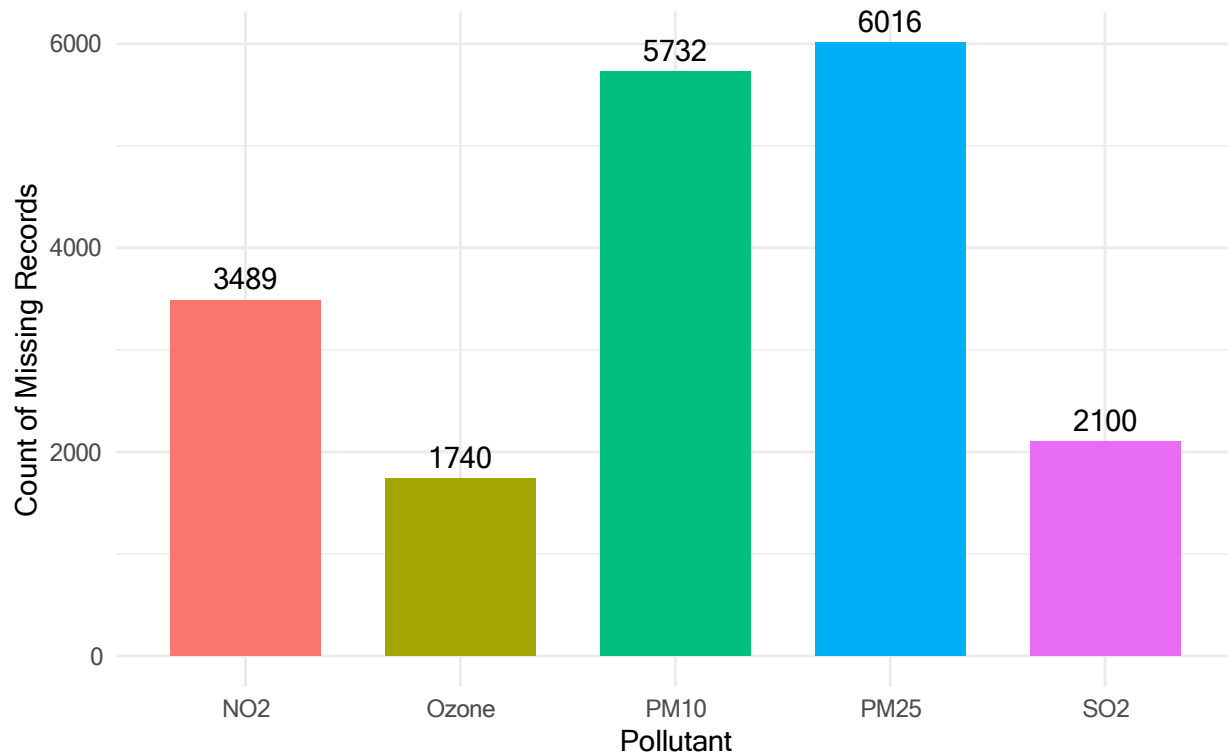
# Plot missing value counts
ggplot(missingCounts, aes(x = Pollutant, y = MissingCount, fill = Pollutant)) +
  geom_col(width = 0.7) +
  geom_text(aes(label = MissingCount), vjust = -0.5, size = 4) +
  labs(
    title = "Missing Value Count per Pollutant",
    subtitle = "Shows how many records were missing for each pollutant",
    x = "Pollutant", y = "Count of Missing Records"
  ) +

```

```
theme_minimal() +
theme(legend.position = "none")
```

Missing Value Count per Pollutant

Shows how many records were missing for each pollutant



```
# TASK 3: FUNCTION displayPollutantGraph() + GRAPH (15 marks)
# This function plots pollutant concentration measured at NOON, on the 1st day of each month.
displayPollutantGraph <- function(pollutant, startMonth, startYear, endMonth, endYear) {

  data_filtered <- PortTalbotAQ %>%
    filter(
      hour(DateTime) == 12, day(DateTime) == 1, # Noon, 1st day each month
      month(DateTime) >= startMonth, year(DateTime) >= startYear,
      month(DateTime) <= endMonth, year(DateTime) <= endYear
    )

  # Annotate highest and lowest points
  max_val <- data_filtered %>% filter(.data[[pollutant]] == max(.data[[pollutant]], na.rm = TRUE))
  min_val <- data_filtered %>% filter(.data[[pollutant]] == min(.data[[pollutant]], na.rm = TRUE))

  ggplot(data_filtered, aes(x = DateTime, y = .data[[pollutant]])) +
    geom_line(linewidth = 1.2, colour = "#0072B2") +
    geom_point(color = "#0072B2") +
    geom_label(data = max_val, aes(label = paste("Peak:", round(.data[[pollutant]], 1))),
      fill = "red", color = "white", size = 3) +
    geom_label(data = min_val, aes(label = paste("Low:", round(.data[[pollutant]], 1))),
      fill = "yellow", size = 3) +
    labs(
      title = paste("Trend of", pollutant),
```

```

    subtitle = "Measured on the 1st day of each month at 12:00",
    x = "Date", y = paste(pollutant, "(µg/m³)")
  ) +
  theme_minimal()
}

```

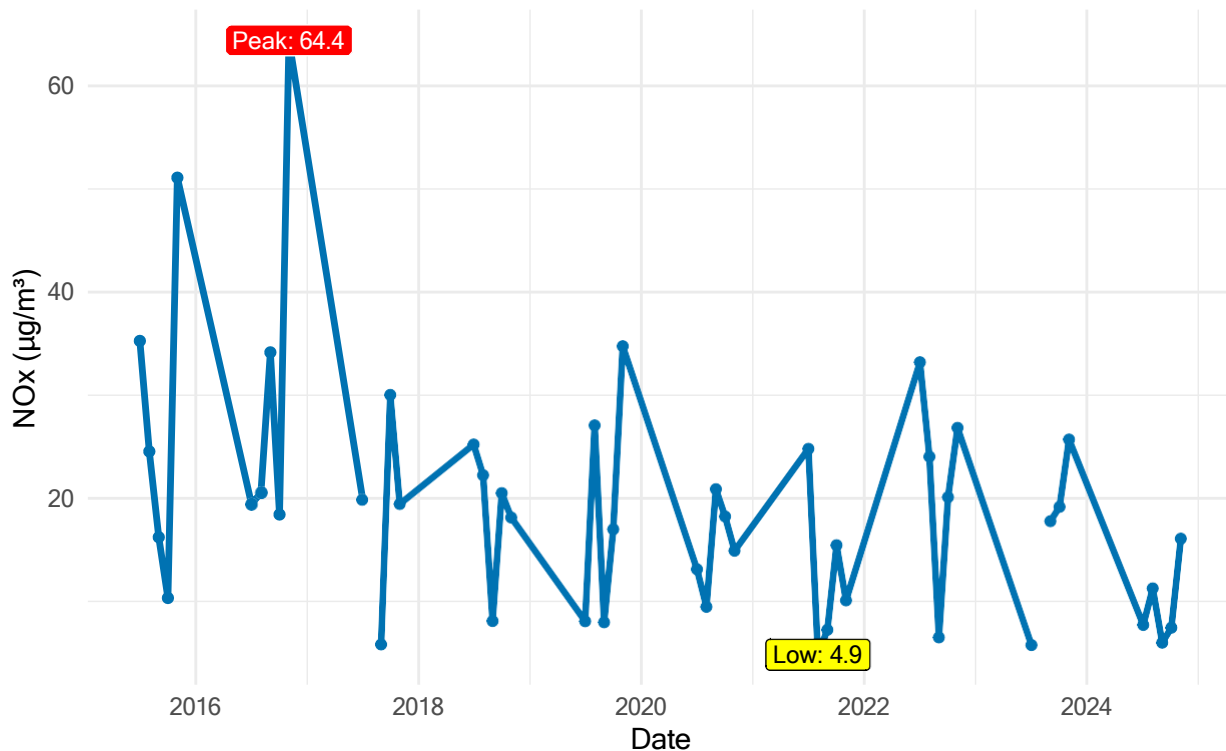
Required output for Task 3:

```
displayPollutantGraph("NOx", 7, 2015, 11, 2024)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Trend of NOx

Measured on the 1st day of each month at 12:00



TASK 4: FUNCTION comparePollutants() + MULTI-POLLUTANT GRAPH (8 marks)

```
comparePollutants <- function(pollutants) {
```

```
  data_filtered <- PortTalbotAQ %>% filter(hour(DateTime) == 12, day(DateTime) == 1)
```

```
  ggplot(data_filtered, aes(x = DateTime)) +
```

```
    lapply(pollutants, function(p) {
```

```
      geom_line(aes(y = .data[[p]], colour = p), linewidth = 1.2)
```

```
    }) +
```

```
    labs(
```

```
      title = "Comparison of Pollutant Levels Across Time",
```

```
      subtitle = "Measured on the 1st of every month at noon",
```

```
      y = "Pollutant Reading (µg/m³)", x = "Date",
```

```
      colour = "Pollutant"
```

```
    ) +
```

```

  annotate("text", x = min(data_filtered$DateTime),
    y = max(data_filtered$NO2, na.rm = TRUE),
    label = "NO2 & PM10 show highest peaks",
    size = 4, hjust = 0) +
  theme_minimal()
}

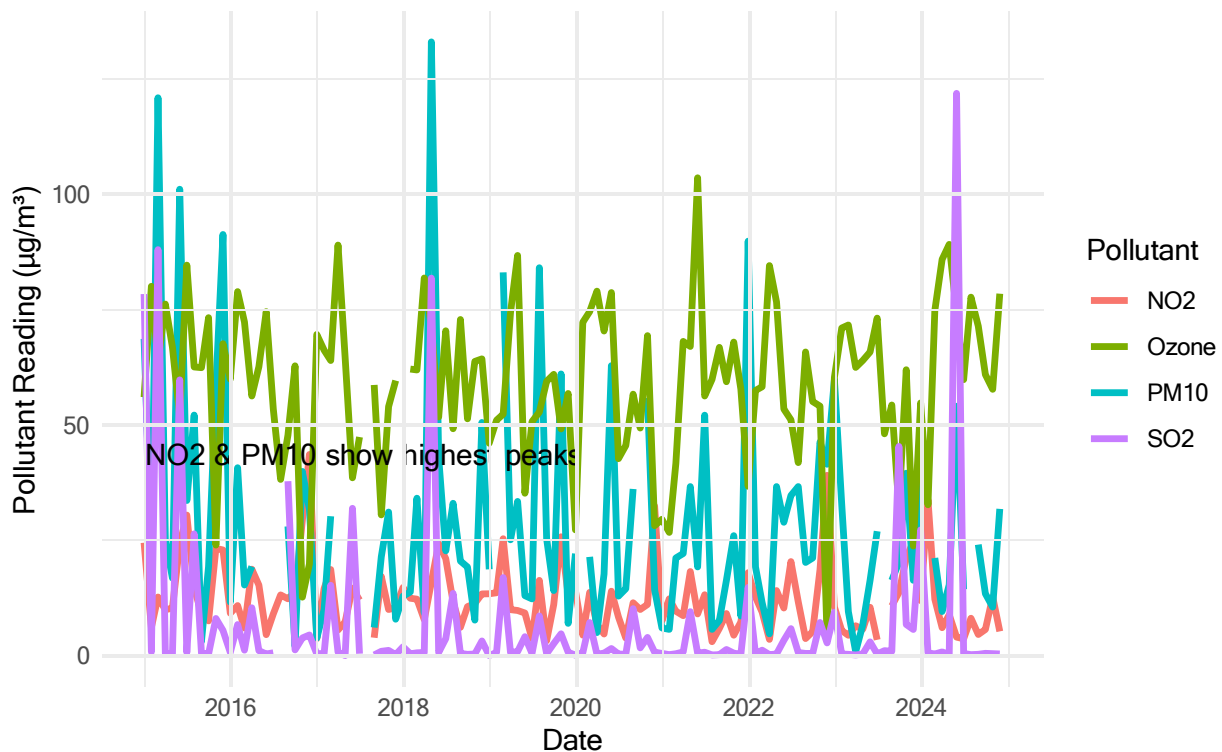
```

Required output for Task 4:

```
comparePollutants(c("NO2", "PM10", "Ozone", "SO2"))
```

Comparison of Pollutant Levels Across Time

Measured on the 1st of every month at noon



TASK 5: FUNCTION O3NOCorrelation() + SCATTERPLOT + CORRELATION (9 marks)

```
O3NOCorrelation <- function() {
```

```
  clean <- PortTalbotAQ %>% drop_na(Ozone, NO2)
```

```
  correlation <- cor(clean$Ozone, clean$NO2)
```

```
  ggplot(clean, aes(Ozone, NO2)) +
```

```
    geom_point(alpha = 0.4, colour = "darkgreen") +
```

```
    geom_smooth(method = "lm", se = FALSE, colour = "red") +
```

```
    annotate("text",
```

```
      x = max(clean$Ozone),
```

```
      y = min(clean$NO2),
```

```
      hjust = 1, label = paste("Correlation =", round(correlation, 3)),
```

```
      size = 4, colour = "red") +
```

```
  labs(
```

```
    title = "Correlation Between Ozone and Nitrogen Dioxide",
```

```

    subtitle = "Scatterplot with regression line",
    x = "Ozone (µg/m³)", y = "NO2 (µg/m³)"
  ) +
  theme_minimal()
}

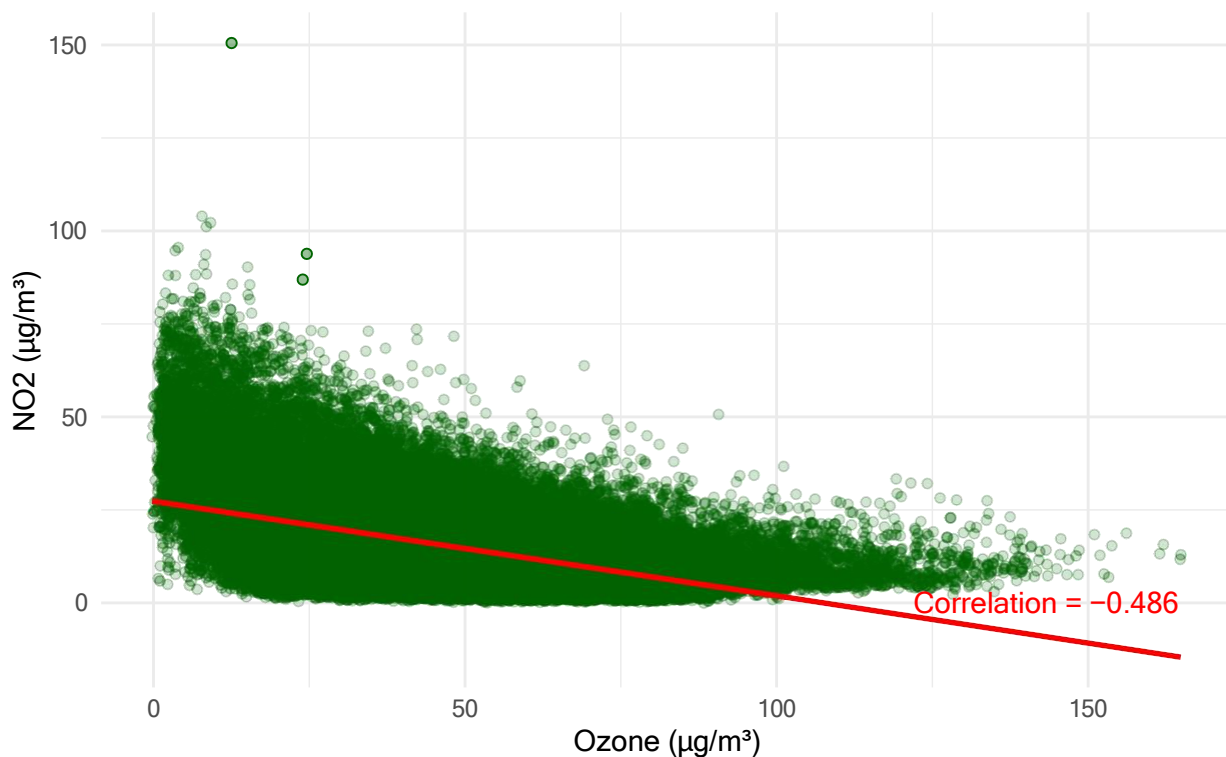
```

Required output for Task 5:
O3NOCorrelation()

```
## `geom_smooth()` using formula = 'y ~ x'
```

Correlation Between Ozone and Nitrogen Dioxide

Scatterplot with regression line



TASK 6: T-TEST – EFFECT OF WIND SPEED ON OZONE LEVELS (10 marks)

```
cat("\n==== Task 6: T-Test – Effect of Wind Speed on Ozone ==== \n")
```

```
##
```

```
## ===== Task 6: T-Test – Effect of Wind Speed on Ozone =====
```

Hypotheses (written in comments for submission)

H0 (Null Hypothesis): Mean ozone levels are the same regardless of wind speed.

H1 (Alternative Hypothesis): Ozone levels differ when WindSpeed < 5 compared to >= 5.

```
group_lowWind <- PortTalbotAQ$Ozone[PortTalbotAQ$WindSpeed < 5]
```

```
group_highWind <- PortTalbotAQ$Ozone[PortTalbotAQ$WindSpeed >= 5]
```

```
print(t.test(group_lowWind, group_highWind))
```

```
##
```

```

## Welch Two Sample t-test
##
## data: group_lowWind and group_highWind
## t = -86.531, df = 80001, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -11.84095 -11.31642
## sample estimates:
## mean of x mean of y
## 47.15779 58.73648

# Decision:
# If p-value < 0.05 → Reject H0 → Wind speed has significant effect on ozone levels.

# TASK 7: ANOVA + POST HOC TEST (Tukey) (20 marks)
cat("\n===== Task 7: ANOVA – NO2 Levels at 17:00 in Different Years =====\n")

##
## ===== Task 7: ANOVA – NO2 Levels at 17:00 in Different Years =====

# Hypotheses (written in comments for submission)
# H0: Mean NO2 concentration is the same across the three periods.
# H1: At least one year significantly differs.

anova_data <- PortTalbotAQ %>%
  filter(hour(DateTime) == 17,
         month(DateTime) %in% c(11, 12),
         year(DateTime) %in% c(2016, 2018, 2020)) %>%
  mutate(period = factor(year(DateTime))) %>%
  drop_na(NO2)

anova_result <- aov(NO2 ~ period, data = anova_data)
print(summary(anova_result))

##              Df Sum Sq Mean Sq F value    Pr(>F)
## period          2   6866    3433   15.35 7.2e-07 ***
## Residuals      175  39133     224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# If ANOVA p-value < 0.05 → Perform post-hoc to determine which year differs
cat("\n--- Performing Post-Hoc Tukey Test (because ANOVA indicates difference) ---\n")

##
## --- Performing Post-Hoc Tukey Test (because ANOVA indicates difference) ---

print(TukeyHSD(anova_result))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = NO2 ~ period, data = anova_data)
##
## $period
##              diff              lwr              upr              p adj
## 2018-2016 -12.7239363 -19.204890 -6.242983 0.0000202

```

```
## 2020-2016 -13.5170197 -19.997973 -7.036066 0.0000057
## 2020-2018 -0.7930834 -7.301211 5.715044 0.9553023
# Export cleaned dataset for record keeping
write.csv(PortTalbotAQ, "Clean_PortTalbotAQ.csv", row.names = FALSE)
```