EDA Reto

2025-08-22

```
library(readxl)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(janitor)
```

```
##
## Adjuntando el paquete: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
##
## Adjuntando el paquete: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
path <- file.choose()  # abre el explorador de archivos


# Ver hojas disponibles (por si necesitas elegir)
excel_sheets(path)
```

```
## [1] "Param_horarios_Estaciones"
```

```
# Leer la primera hoja (o pon el nombre exacto de la hoja)
df <- read_excel(
  path,
  sheet = 1,                          # o "2023", "2024", etc.
  na = c("", "NA", "-99", "-999", "-9999"),
  guess_max = 100000                  # mejora la detección de tipos en archivos grandes
) |>
  clean_names()
```

```
# valores únicos de la primera fila
vals <- as.character(unlist(df[1, ], use.names = FALSE))
vals <- trimws(vals)

unicos <- sort(unique(vals[!is.na(vals) & vals != ""]))
cat(paste(unicos, collapse = "\n"))
```

```
## CO
## NO
## NO2
## NOX
## O3
## PM10
## PM2.5
## PRS
## RAINF
## RH
## SO2
## SR
## TOUT
## WDR
## WDV
## WSR
```

```
library(dplyr)
library(tidyr)
library(stringr)
library(lubridate)
library(janitor)

# contaminantes válidos
CONT_OK <- c("CO","NO","NO2","NOX","O3","PM10","PM2.5","PRS","RAINF","RH","SO2","SR","TOUT","WD
R","WDV","WSR")

date_col <- intersect(c("date"), names(df))
stopifnot(length(date_col) >= 1)
date_col <- date_col[1]

# diccionario col -> contaminante leyendo la PRIMERA fila
lab_primera <- toupper(trimws(as.character(df[1, ])))
dicc <- tibble(col = names(df), etiqueta = lab_primera) |>
  filter(etiqueta %in% CONT_OK)

# columnas de estación (todas menos la fecha)
cols_est <- setdiff(dicc$col, date_col)

# quito la fila 1 que traía etiquetas, y me quedo con fecha + estaciones
dat <- df[-1, c(date_col, cols_est)]

# normalizo la fecha (varios formatos posibles; también números de Excel)
parse_dt <- function(x){
  if (is.numeric(x)) return(as.POSIXct(as.numeric(x)*86400, origin="1899-12-30", tz="America/Mon
terrey"))
  x <- as.character(x)
  out <- suppressWarnings(ymd_hms(x, tz="America/Monterrey"))
  out[is.na(out)] <- suppressWarnings(ymd_hm(x, tz="America/Monterrey"))[is.na(out)]
  out[is.na(out)] <- suppressWarnings(ymd(x, tz="America/Monterrey"))[is.na(out)]
  out
}
dat[[date_col]] <- parse_dt(dat[[date_col]])

# paso a largo, asigno estación y contaminante por columna
df_long <- dat |>
  pivot_longer(-all_of(date_col), names_to = "col", values_to = "valor_raw") |>
  mutate(
    contaminante = dicc$etiqueta[match(col, dicc$col)],
    col_norm = tolower(col) |> str_replace_all("\\s+", "_"),
    estacion = toupper(str_remove(col_norm, "_\\d+$")),
    valor = suppressWarnings(as.numeric(valor_raw))
  ) |>
  select(date = all_of(date_col), estacion, contaminante, valor)

# resumen rápido
cat("Observaciones:", nrow(df_long),
    "| Contaminantes:", n_distinct(df_long$contaminante), "\n")
```

```
## Observaciones: 3093233 | Contaminantes: 16
```

```
eda <- df_long |>
  group_by(estacion, contaminante) |>
  mutate(
    med = median(valor, na.rm = TRUE),
    mad1 = mad(valor, constant = 1.4826, na.rm = TRUE),
    zrob = (valor - med)/mad1,
    outlier = !is.na(zrob) & abs(zrob) > 3
  ) |>
  ungroup()

# tabla de outliers por contaminante y por estación
out_por_cont <- eda |> summarise(n = n(), outs = sum(outlier, na.rm=TRUE),
                                 pct = 100*outs/n, .by = contaminante) |>
  arrange(desc(pct))
out_por_est  <- eda |> summarise(n = n(), outs = sum(outlier, na.rm=TRUE),
                                 pct = 100*outs/n, .by = estacion) |>
  arrange(desc(pct))

print(out_por_cont)
```

```
## # A tibble: 16 × 4
##    contaminante      n  outs      pct
##    <chr>         <int> <int>    <dbl>
##  1 SR           208065 81749 39.3
##  2 NO           208065 35424 17.0
##  3 WDR           97097 11973 12.3
##  4 WDV          110968 13614 12.3
##  5 NOX          208065 21523 10.3
##  6 SO2          208065 15363  7.38
##  7 NO2          208065 10944  5.26
##  8 PM10         208065  8924  4.29
##  9 PM2.5        180323  5174  2.87
## 10 O3           208065  4933  2.37
## 11 CO           208065  3495  1.68
## 12 RAINF        208065  3431  1.65
## 13 PRS          208065  3027  1.45
## 14 WSR          208065  2222  1.07
## 15 TOUT         208065   503  0.242
## 16 RH           208065    12  0.00577
```

```
print(head(out_por_est, 20))
```

```
## # A tibble: 20 × 4
##    estacion          n  outs   pct
##    <chr>         <int> <int> <dbl>
##  1 SURESTE_3_2   13871  1451 10.5
##  2 NOROESTE_3   180323 15708  8.71
##  3 NORESTE_3_2   13871  1181  8.51
##  4 SUROESTE     208065 17683  8.50
##  5 NORTE_2      194194 15280  7.87
##  6 NORESTE_3    180323 13886  7.70
##  7 SURESTE      208065 15832  7.61
##  8 NORESTE      208065 15590  7.49
##  9 SURESTE2     208065 15499  7.45
## 10 SURESTE_3    194194 14200  7.31
## 11 CENTRO       208065 15030  7.22
## 12 SUR          208065 14555  7.00
## 13 NOROESTE     208065 14475  6.96
## 14 NOROESTE_2_2  13871   939  6.77
## 15 SUROESTE2    208065 13561  6.52
## 16 NOROESTE_2   194194 12044  6.20
## 17 NORTE        208065 12640  6.08
## 18 NORESTE2     208065 11807  5.67
## 19 NOROESTE_3_2  13871   650  4.69
## 20 NORTE_2_2     13871   300  2.16
```

Qué muestra esta tabla

Es un resumen de outliers por contaminante. Para cada contaminante reporto:

n: cuántas observaciones hay.

outs: cuántas quedaron marcadas como atípicas con el criterio |z-score robusto| > 3 (calculado por estación–contaminante usando mediana y MAD).

pct: el porcentaje de outliers respecto a n.

Cómo interpretar los porcentajes

SR aprox 39%. SR (radiación solar) tiene distribución “mixta”: de noche aprox 0 y de día sube mucho. Si mezclo día+noche en el mismo umbral, la MAD queda pequeña (muchos ceros) y los valores diurnos “saltan” como atípicos. No es que 39% esté mal: hay que evaluar SR separando día/noche (o por hora).

WDR/WDV aprox 12–13%. Son direcciones de viento (variables angulares). Un z-score lineal no aplica (0° y 360° son el mismo punto). Se deben tratar como ángulos o convertir el viento a componentes u = WSR·cos(θ) y v = WSR·sin(θ) y evaluar ahí los outliers.

NO, NOX, SO2 aprox 7–17%. Gases con picos de episodio (tráfico, estabilidad). Es normal ver colas pesadas y porcentajes más altos que en una normal ideal.

PM10 / PM2.5 / O3 aprox 2–5%. Rango esperado; indica menos extremos (o que ya se suavizan al promediar).

```r
library(dplyr)
library(tidyr)

# dónde hay duplicados
dups <- df_long |>
  summarise(n = dplyr::n(), .by = c(date, estacion, contaminante)) |>
  filter(n > 1L)

# colapso duplicados: promedio por (date, estacion, contaminante)
df_long_dedup <- df_long |>
  mutate(valor = suppressWarnings(as.numeric(valor))) |>
  summarise(valor = mean(valor, na.rm = TRUE),
            .by = c(date, estacion, contaminante))
```

```r
wind_uv <- df_long_dedup |>
  filter(contaminante %in% c("WDV","WSR")) |>
  # paso a ancho ya sin duplicados
  pivot_wider(names_from = contaminante, values_from = valor) |>
  # coerción segura + normalización angular
  mutate(
    WDV = suppressWarnings(as.numeric(WDV)),
    WSR = suppressWarnings(as.numeric(WSR)),
    WDV = (WDV %% 360)
  ) |>
  mutate(
    u = WSR * cos(pi * WDV / 180),
    v = WSR * sin(pi * WDV / 180)
  ) |>
  select(date, estacion, u, v)

# lo regreso a largo para juntarlo con el resto
wind_uv_long <- wind_uv |>
  pivot_longer(c(u, v), names_to = "contaminante", values_to = "valor")
```

```
# base para outliers: todo menos WDR/WDV; en su lugar uso u/v
base_out <- df_long_dedup |>
  filter(!contaminante %in% c("WDR","WDV")) |>
  bind_rows(wind_uv_long)

# z-score robusto por estación-contaminante;
# para SR separo día/noche para no inflar outliers
library(lubridate)

df_tag <- base_out |>
  mutate(hora = hour(date),
         grupo = ifelse(contaminante == "SR" & hora %in% 6:18, "SR_DIA",
                 ifelse(contaminante == "SR", "SR_NOCHE", "REG"))) |>
  group_by(estacion, contaminante, grupo) |>
  mutate(
    med  = median(valor, na.rm = TRUE),
    mad1 = mad(valor, constant = 1.4826, na.rm = TRUE),
    zrob = (valor - med)/mad1,
    outlier = is.finite(zrob) & abs(zrob) > 3
  ) |>
  ungroup()

resumen_out <- df_tag |>
  filter(!contaminante %in% c("u","v")) |>
  summarise(n = dplyr::n(),
            outs = sum(outlier, na.rm = TRUE),
            pct = 100*outs/n,
            .by = contaminante) |>
  arrange(desc(pct))

print(resumen_out)
```

```
## # A tibble: 14 × 4
##    contaminante      n  outs      pct
##    <chr>         <int> <int>    <dbl>
##  1 SR           199395 47323 23.7
##  2 NO           199395 33786 16.9
##  3 NOX          199395 20535 10.3
##  4 SO2          199395 15025  7.54
##  5 NO2          199395 10781  5.41
##  6 PM10         199395  8591  4.31
##  7 PM2.5        172809  4938  2.86
##  8 O3           199395  4495  2.25
##  9 CO           199395  3344  1.68
## 10 PRS          199395  2879  1.44
## 11 WSR          199395  2041  1.02
## 12 TOUT         199395   479  0.240
## 13 RH           199395     8  0.00401
## 14 RAINF        199395     0  0
```

Viento en componentes. Las direcciones WDR/WDV son angulares (0° ≡ 360°), así que un z-score lineal no es válido. Convertí el viento a u = WSR·cos(θ) y v = WSR·sin(θ) (con θ en grados, WSR en m/s) y usé u/v para el etiquetado de outliers. También forcé WDV/WSR a numérico y normalicé WDV a [0,360).

SR por día y noche. La radiación solar (SR) es ~0 de noche y alta de día. Si mezclo ambos, la MAD queda pequeña y "todo el día" parece atípico. Por eso separé SR en dos grupos (día 06–18 h y noche) y a cada grupo le apliqué su propio umbral.

Criterio robusto y por estación. Marqué outliers con $|z\text{rob}|>3$, donde $z\text{rob}=(x-\text{mediana})/\text{MAD}$, calculado por estación y contaminante (y por franja en SR). No eliminé datos; solo los etiqueté.

Resultado después de las correcciones (aprox.)

SR ~ 23.7% (bajó desde ~39% al separar día/noche).

NO ~ 16.9%, NOX ~ 10.3%, SO2 ~ 7.5%, NO2 ~ 5.4%: gases con picos de episodio; es esperable ver colas pesadas.

PM10 ~ 4.3%, PM2.5 ~ 2.9%, O3 ~ 2.3%, CO ~ 1.7%, PRS ~ 1.4%: rangos razonables para este método.

El n de PM2.5 es menor porque no todas las estaciones reportan esa variable en todo el periodo.

Cómo usarlo en el análisis

Mantengo los valores originales y la bandera de outlier para hacer análisis con y sin outliers (sensibilidad).

Para comparar estaciones (topografía), trabajaremos con promedios diarios por estación y contaminante, exigiendo cobertura ≥ 75% de horas por día.

En multivariado (PCA/MANOVA por estación) usaré contaminantes en escala z y, para viento, u/v en vez de ángulos crudos.

Esta estrategia evita falsos positivos por estructura del dato (ángulos, día/noche) y deja los outliers como señal física real cuando corresponde (episodios).

```
library(readr)
# Discretización con intención: episodios p90 por contaminante
episodios <- df_long_dedup |>
  summarise(thres_p90 = quantile(valor, 0.90, na.rm = TRUE),
            .by = contaminante)

freq_episodios <- df_long_dedup |>
  left_join(episodios, by = "contaminante") |>
  mutate(ep90 = valor >= thres_p90) |>
  summarise(pct_episodios = mean(ep90, na.rm = TRUE) * 100,
            .by = c(estacion, contaminante)) |>
  arrange(contaminante, desc(pct_episodios))

# Agregados diarios con cobertura ≥ 75% y escalado z para multivariado
diario <- df_long_dedup |>
  mutate(fecha = as.Date(date)) |>
  summarise(
    n_ok  = sum(!is.na(valor)),
    media = if_else(n_ok >= 18, mean(valor, na.rm = TRUE), NA_real_),
    .by = c(estacion, fecha, contaminante)
  )

X_day <- diario |>
  select(estacion, fecha, contaminante, media) |>
  pivot_wider(names_from = contaminante, values_from = media)

cont_cols <- intersect(c("CO","NO","NO2","NOX","O3","PM10","PM2.5","SO2"),
                       names(X_day))

X_day_z <- X_day |>
  mutate(across(all_of(cont_cols), ~ as.numeric(scale(.x)),
                .names = "{.col}_z"))

# Atributos derivados útiles (lluvia diaria y % horas en calma)
rain_day <- df_long_dedup |>
  filter(contaminante == "RAINF") |>
  mutate(fecha = as.Date(date)) |>
  summarise(rain_mm = sum(as.numeric(valor), na.rm = TRUE),
            .by = c(estacion, fecha))

calma_day <- df_long_dedup |>
  filter(contaminante == "WSR") |>
  mutate(WSR = as.numeric(valor), fecha = as.Date(date)) |>
  summarise(pct_calma = mean(WSR < 1, na.rm = TRUE) * 100,
            .by = c(estacion, fecha))

X_features <- X_day |>
  left_join(rain_day, by = c("estacion","fecha")) |>
  left_join(calma_day, by = c("estacion","fecha"))

# 4) Reformateos finales y guardados
df_horario         <- df_long_dedup
```

```
df_diario_ancho    <- X_day
df_diario_features <- X_features |>
  mutate(across(all_of(cont_cols), ~ as.numeric(scale(.x)),
                .names = "{.col}_z"))

write_csv(df_horario,         "aire_mty_horario_long_dedup.csv")
write_csv(df_diario_ancho,    "aire_mty_diario_ancho.csv")
write_csv(df_diario_features, "aire_mty_diario_features.csv")
write_csv(freq_episodios,     "aire_mty_freq_episodios_p90.csv")

# chequeos rápidos para el informe
cat("Estaciones:", n_distinct(df_long_dedup$estacion),
    "| Contaminantes:", n_distinct(df_long_dedup$contaminante),
    "| Obs horarias:", nrow(df_long_dedup), "\n")
```

```
## Estaciones: 20 | Contaminantes: 16 | Obs horarias: 2964339
```

```
faltantes <- df_diario_ancho |>
  summarise(across(-c(estacion, fecha), ~ mean(is.na(.))*100))
print(faltantes)
```

```
## # A tibble: 1 × 16
##      CO    NO   NO2   NOX    O3  PM10 PM2.5   PRS RAINF    RH   SO2    SR  TOUT
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  29.0  26.9  26.9  26.9  26.7  26.4  42.6  26.0  25.7  31.4  28.1  26.9  26.0
## # ℹ 3 more variables: WSR <dbl>, WDV <dbl>, WDR <dbl>
```

```
dplyr::glimpse(df_horario)
```

```
## Rows: 2,964,339
## Columns: 4
## $ date         <dttm> 2022-12-31 18:00:00, 2022-12-31 18:00:00, 2022-12-31 18:…
## $ estacion     <chr> "SURESTE", "SURESTE", "SURESTE", "SURESTE", "SURESTE", "S…
## $ contaminante <chr> "CO", "NO", "NO2", "NOX", "O3", "PM10", "PM2.5", "PRS", "…
## $ valor        <dbl> 2.3700, 54.5000, 32.6000, 87.1000, 3.0000, 110.0000, 68.0…
```

```
dplyr::count(df_horario, estacion, contaminante) |> head()
```

estacion <chr>	contaminante <chr>	n <int>
CENTRO	CO	13293
CENTRO	NO	13293
CENTRO	NO2	13293
CENTRO	NOX	13293

estacion <chr>	contaminante <chr>	n <int>
CENTRO	O3	13293
CENTRO	PM10	13293

6 rows

```
names(df_diario_ancho)
```

```
##  [1] "estacion" "fecha"    "CO"       "NO"       "NO2"      "NOX"
##  [7] "O3"       "PM10"     "PM2.5"    "PRS"      "RAINF"    "RH"
## [13] "SO2"      "SR"       "TOUT"     "WSR"      "WDV"      "WDR"
```

```
head(freq_episodios)
```

estacion <chr>	contaminante <chr>	pct_episodios <dbl>
NORESTE	CO	35.32089
NORESTE2	CO	27.68602
SUROESTE2	CO	14.91725
CENTRO	CO	13.47721
SURESTE	CO	12.32227
NOROESTE_2	CO	10.20331

6 rows

Formato de la base y qué verifiqué

Dejamos la base en formato largo con cuatro columnas: date (hora), estación, contaminante y valor. El dataset resultante tiene 2,964,339 filas, y la lista de contaminantes es la esperada (CO, NO, NO_2, NOx, O_3, PM10, PM2.5, PRS, RAINF, RH, SO_2, SR, TOUT, WSR, WDV/WDR). Con esto confirmo que el reetiquetado y la reestructuración quedaron correctos para análisis por estación y por contaminante.

Cobertura por estación–contaminante

Antes de comparar estaciones, revisamos la cobertura (número de horas con dato por estación y contaminante). Por ejemplo, en CENTRO obtuvimos alrededor de 13,293 registros por contaminante; esa magnitud de n nos indica que hay información suficiente para calcular promedios diarios y comparar estaciones con un criterio homogéneo. Esta verificación es importante porque, si una estación tuviera muy pocas horas válidas, su comparación podría sesgarse.

Frecuencia de episodios altos (p90)

Para medir “acumulamiento” desde otra perspectiva, calculamos la frecuencia de episodios: porcentaje de horas que están por arriba del percentil 90 (p90) global de cada contaminante. En CO, el ranking muestra a NORESTE ≈ 35.3% y NORESTE2 ≈ 27.7% como las estaciones con mayor proporción de horas altas, seguidas por SUROESTE2 (~14.9%), CENTRO (~13.5%), SURESTE (~12.3%) y NOROESTE_2 (~10.2%). Esta señal es consistente con acumulamiento local (topografía + configuración de fuentes) en el eje noreste: no es un pico aislado, sino una frecuencia sostenida de valores altos. Este mismo análisis lo aplicaremos a NO/NO_2/NOx y PM2.5/PM10; si las mismas estaciones aparecen arriba en varios contaminantes, la evidencia de efecto topográfico se fortalece. Para O_3, anticipo patrones distintos por su formación secundaria.

Cómo usamos esta información en el análisis

Con la base larga y la cobertura revisada, construimos promedios diarios por estación–contaminante (exigiendo ≥75% de horas por día) y trabajo en dos planos:

- Comparación univariada por contaminante (ANOVA/Kruskal) para detectar diferencias sistemáticas entre estaciones.
- Enfoque multivariado (PCA/MANOVA) con contaminantes estandarizados para ver si las estaciones forman clusters coherentes. En paralelo, reportamos la frecuencia de episodios p90 por estación como indicador complementario de acumulamiento. Esta combinación nos da una lectura robusta y alineada con el objetivo: evaluar si la topografía de Monterrey se asocia con diferencias persistentes en los niveles de contaminantes entre estaciones.