

Calibracion de sensores de bajo costo

Lenna Pedroso

2025-11-24

1. Marco teórico

La exposición a material particulado fino, con un diámetro aerodinámico menor o igual a 2,5 m (PM_{2,5}), puede provocar millones de muertes anuales, ya que estas partículas al ser inhaladas, pueden penetrar a través del sistema respiratorio y acumularse en diferentes partes del cuerpo (Li et al., 2020). Entre los riesgos asociados a su exposición se destacan: aumento en la probabilidad de desarrollar cáncer de pulmón, mortalidad prematura y enfermedades respiratorias (Borchers-Arriagada et al., 2024).

La Región Metropolitana de Santiago de Chile se encuentra entre las ciudades más contaminadas de Sudamérica, superando el límite máximo permisible referido a la concentración anual de PM_{2,5} establecido para Chile (20 g/m³) (Peralta et al., 2022). Esta situación se debe principalmente a su ubicación geográfica, cuyas características topográficas y condiciones meteorológicas restringen la dispersión de contaminantes favoreciendo su acumulación principalmente durante el invierno (Menares et al., 2021).

El control de la calidad del aire atmosférico desempeña un papel crucial en la prevención de la contaminación ambiental en la región. El Sistema de Información Nacional de Calidad del Aire (SINCA), en Santiago de Chile, presenta sensores remotos que estima las concentraciones de PM_{2,5}. Estos sensores están distribuidos en un número limitado de sitios, (10 estaciones de monitoreo aproximadamente), por lo que ofrecen una resolución espacio-temporal limitada al aportar datos restringidos al área donde se encuentran estas estaciones.

Otra de las limitaciones es su elevado costo y su incompatibilidad, en países en vías de desarrollo, debido a su tamaño y gastos de mantenimiento. Además, requieren la incorporación de personal capacitado y calificado para su operación (Tarazona Alvarado et al., 2024). Por otra parte, los modelos convencionales de distribución de estos sensores de monitoreo están sujetos a grandes incertidumbres debido a la falta de perfiles de emisión precisos de todas las fuentes contribuyentes y al número limitado de mediciones en los sitios receptores. Como consecuencia, gran parte de la población permanece expuesta a fuentes locales no capturada por el SINCA, subestimándose la exposición real y, por ende, sus implicaciones en la salud humana.

La tecnología de detección de bajo costo es una alternativa viable de monitoreo de la calidad del aire debido a que proporcionan información sobre la variación temporal y espacial de la contaminación por PM_{2,5}. Además, brindan información detallada sobre el grado de exposición de la población en la ciudad. Estos sensores se pueden utilizar para caracterizar variaciones de concentración de PM_{2,5} e identificar puntos críticos de contaminación a escala local, incluso en condiciones extremas de contaminación del aire. Por otra parte, proporcionan información sobre la dinámica de los patrones de contaminación y permiten analizar el impacto de las fuentes locales, contribuyendo a la conceptualización de una gestión ambiental más encaminada a reducir la exposición humana a estos contaminantes.

Una de las principales limitaciones de estos sensores es su baja presión en la determinación de la concentración del PM_{2,5} ya que son muy sensibles a variaciones de temperatura y humedad. Además, tienden a sobreestimar los valores de concentración por lo que la calibración de estos prototipos constituye un paso esencial para garantizar un monitoreo complementario preciso en la ciudad.

Objetivo general

El objetivo de esta investigación es realizar la calibración de sensores de bajo costo (Plantower) en la comuna Las Condes obteniendo un error de medición por debajo de 20 g/m³.

2. Metodología

Se realizó un monitoreo durante el periodo del 4 de junio al 23 de agosto de 2025. Los sensores de bajo costo se colocaron en la estación de referencia (SINCA) ubicada en la comuna Las Condes perteneciente a la ciudad de Santiago de Chile. Se colocaron 5 sensores de bajo costo para evaluar su capacidad de medir con precisión las concentraciones de PM_{2,5}.

2.1. Tendencia temporal

Primeramente, se analizará la tendencia temporal de las concentraciones de cinco sensores de la tecnología Plantower (P1-P5) y se compararán con los datos de referencia de la estación regulatoria (SINCA) para determinar si existe similitud en el comportamiento de ambos tipos de tecnología.

```
# Cargar librerias -----
library(dplyr)
library(tidyr)
library(rlang)
#Cargar datos -----
Condes_Plantower_SINCA_long3 <- read.csv2(
```

```

"C:/Semestre 2-2025/Visualizacion/Página web/Pagina web/Datos/Condes_Plantower_SINCA_long3
header = TRUE
)

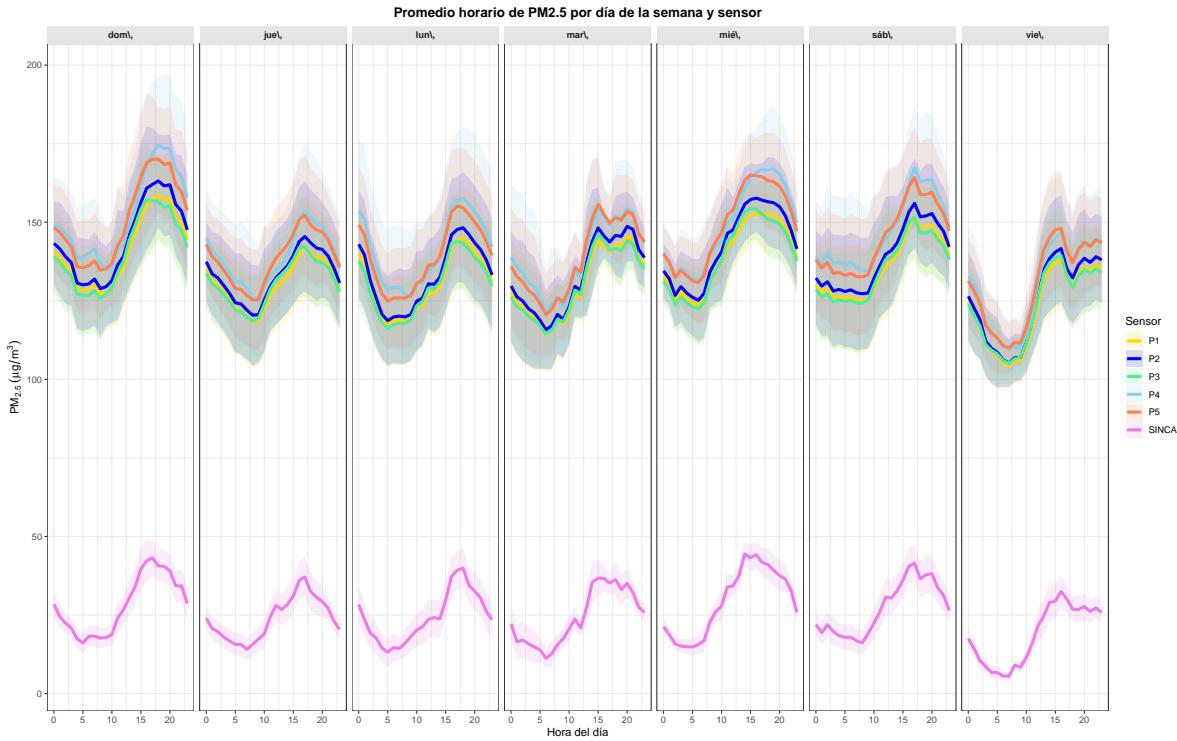
# Funcion para calcular las concentraciones semanales
Condes_summary_pm25 <- function(datos, tiempo, pm_col = "value") {
  datos %>%
    group_by(across(all_of(tiempo)), sensor) %>% # La funcion fue realizada para cada sensor
    summarise(
      n      = sum(!is.na(.data[[pm_col]])), # Para determinar los datos válidos
      mean   = mean(.data[[pm_col]], na.rm = TRUE),
      sd     = sd(.data[[pm_col]], na.rm = TRUE),
      se     = ifelse(n > 1, sd / sqrt(n), NA_real_),
      ymin   = mean - 1.96 * se,
      ymax   = mean + 1.96 * se,
      .groups = "drop"
    )
}
# Resumenes por dia de la hora y por dia de la semana
hour_wday <- Condes_summary_pm25(Condes_Plantower_SINCA_long3, c("wday", "hour"), "value_PM25")
hour <- Condes_summary_pm25(Condes_Plantower_SINCA_long3, "hour", "value_PM25")
wday <- Condes_summary_pm25(Condes_Plantower_SINCA_long3, "wday", "value_PM25")

# Cargar librerias -----
library(ggplot2)
# Grafico de variacion de PM25 por horas y dias de la semana

ggplot(hour_wday, aes(x = hour, y = mean, color = sensor, fill = sensor, group = sensor)) +
  geom_ribbon(aes(ymin = ymin, ymax = ymax), alpha = 0.15, colour = NA) +
  geom_line(linewidth = 1.4) +
  facet_wrap(vars(wday), nrow = 1) +
  labs(
    x = "Hora del día",
    y = expression(PM[2.5]~(mu*g/m^3)),
    color = "Sensor",
    fill = "Sensor",
    title = "Promedio horario de PM2.5 por día de la semana y sensor"
  ) +
  scale_color_manual(
    values = c(
      "P1" = "gold",
      "P2" = "blue",

```

```
"P3" = "#4EEE94",
"P4" = "#87CEEB",
"P5" = "coral",
"SINCA" = "#EE7AE9"
)
) +
scale_fill_manual(
values = c(
"P1" = "gold",
"P2" = "blue",
"P3" = "#4EEE94",
"P4" = "#87CEEB",
"P5" = "coral",
"SINCA" = "#EE7AE9"
)
) +
theme_bw() +
theme(
legend.position = "right",
plot.title = element_text(hjust = 0.5, face = "bold"),
strip.background = element_rect(fill = "grey90", color = NA),
strip.text = element_text(face = "bold"),
axis.text.x = element_text(angle = 0, hjust = 0.5)
)
```



Al analizar el gráfico se observa que existe una marcada diferencia en el comportamiento de los sensores de bajo costo en comparación con los valores de la estación de referencia (SINCA). En este caso los sensores de bajo costo tienden a sobreestimar las concentraciones de material particulado demostrándose la importancia de su calibración para solucionar estas limitaciones.

2.2. Regresión lineal

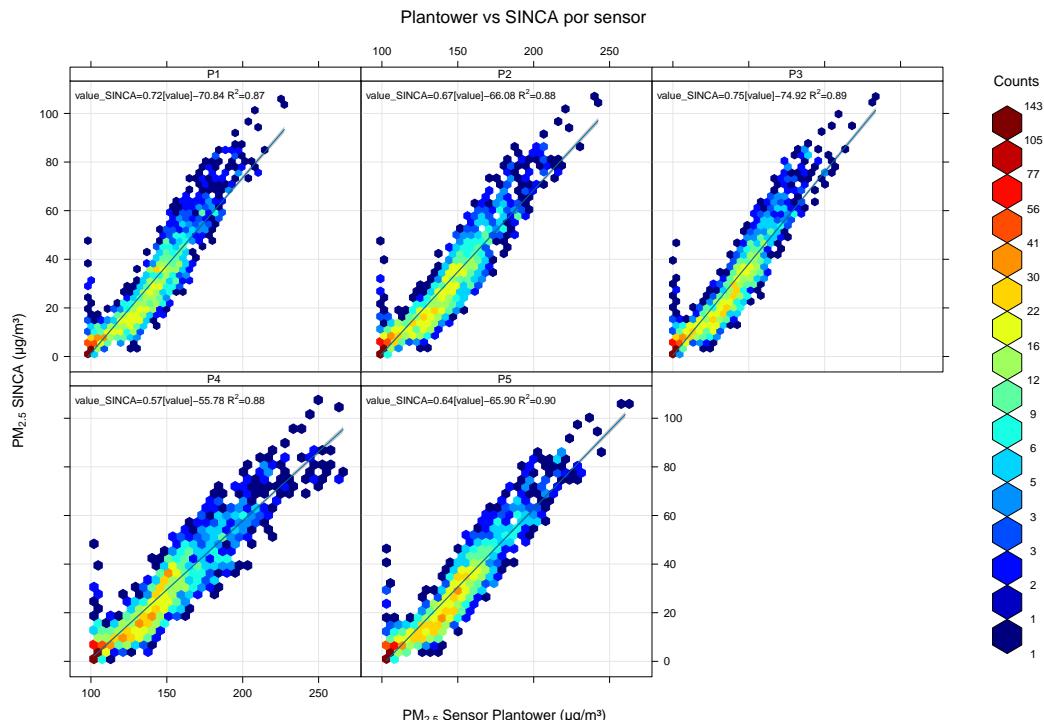
Para obtener la ecuación de calibración para cada sensor de bajo costo se aplicó un modelo de regresión lineal mediante `#scaterplot#`. Este grafico es específico para analizar regresiones lineales correspondientes a contaminantes atmosféricos ya que permite observar la frecuencia de episodios de alta concentración en la ciudad. Los resultados se muestran en la siguiente figura.

```
# Cargar librerias -----
library(openair)
#Cargar datos -----
Condes_Plantower_SINCA_long_2 <- read.csv2(
  "C:/Semestre 2-2025/Visualizacion/Página web/Página web/Datos/Condes_P_SINCA_long.csv",
  header = TRUE
)
```

```

# Modelo de regresión lineal-----
scatterPlot(
  mydata = Condes_Plantower_SINCA_long_2,
  x      = "value",
  y      = "value_SINCA",
  method = "hexbin",
  col    = "jet",
  linear = TRUE,
  ci     = TRUE,
  fit    = TRUE,
  type   = "sensor",
  main   = "Plantower vs SINCA por sensor",
  xlab   = "PM2.5 Sensor Plantower (µg/m3)",
  ylab   = "PM2.5 SINCA (µg/m3)",
  fontsize = 14
)

```



En la figura se demuestra que a pesar de existir un defase en el comportamiento de estos sensores un modelo de regresión lineal puede explicar con un buen ajuste ($R^2=0.87-0.9$) las concentraciones de material particulado. Además, los valores de alta concentración presentaban bajas frecuencias de aparición, destruyéndose que estos episodios no fueron comunes en el

periodo estudiado.

2.3 Análisis de los errores del modelo

Uno de los aspectos a considerar para el uso de los sensores de bajo costo como complemento de los monitores regulatorios (SINCA) es el análisis de error. En este contexto, se determinó que los errores absoluto, así como las métricas específicas de error (MB, MGE) deberían estar por debajo de 20 g/m³. Los resultados se muestran en la siguiente figura.

```
#Cargar datos -----
errores_completo <- read.csv2( "C:/Semestre 2-2025/Visualizacion/Página web/Página web/Datos"
  header = TRUE
)

# Cargar librerías -----
library(ggplot2)
library(dplyr)

# Seleccionar datos -----

errores_no_calibrado <- errores_completo %>% filter(Clasificacion == "No calibrados")

# Grafico de violín para errores no calibrados -----

violin_nocalibrado <- ggplot(errores_no_calibrado ,
  aes(x = sensor, y = error_abs, fill = sensor)) +
  geom_violin(trim = FALSE, alpha = 0.35, linewidth = 1.1) + # No recorta extremo y especifica
  geom_boxplot(width = 0.10, alpha = 0.8, linewidth = 1.0, #Añadir box plot dentro del violín
    outlier.size = 1.5) + # Especifica el tamaño de los puntos outlier
  geom_hline(yintercept = 20, linetype = "dashed", # Añade una linea horizontal en y=20
    color = "firebrick4", linewidth = 1) + # Selecciona el color de la linea
  scale_y_continuous(limits = c(0, NA)) + # Establece que el valor mínimo es 0 y sin máx
  scale_fill_manual(values = c(
    "PM25_P1" = "firebrick1",
    "PM25_P2" = "firebrick2",
    "PM25_P3" = "firebrick3",
    "PM25_P4" = "firebrick4",
    "PM25_P5" = "#5A0F0F"
  )) +
  theme_bw(base_size = 16) + # Tema blanco y negro con tamaño de texto 16
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5), #Título en negrita y centrado
```

```

    axis.text.x = element_text(angle = 45, hjust = 1) # Inclina el eje para evitar solapamiento
) +
labs(title = "Error absoluto - No calibrados",
      x = "sensor", y = "Error absoluto (µg/m³)",
      fill = "sensor")

# Grafico de violín para errores calibrados -----
# Seleccionar datos -----
errores_cal <- errores_completo %>% filter(Clasificacion == "Calibrados")

# Grafico de violín para errores no calibrados ----

violin_calibrado <- ggplot(errores_cal,
                           aes(x = sensor, y = error_abs, fill = sensor)) +
  geom_violin(trim = FALSE, alpha = 0.35, linewidth = 1.1) + # No recorta extremo y especifica el ancho
  geom_boxplot(width = 0.1, alpha = 0.8, linewidth = 1.0, #Añadir box plot dentro del violín
               outlier.size = 1.5) + # Especifica el tamaño de los puntos outlier
  geom_hline(yintercept = 20, linetype = "dashed", # Añade una linea horizontal en y=20
             color = "firebrick4", linewidth = 1) + # Selecciona el color de la linea
  scale_y_continuous(limits = c(0, NA)) + # Establece que el valor minimo es 0 y sin maximo
  scale_fill_manual(values = c( # Seleccion de colores para cada grafico de violín
    "PM25_P1" = "firebrick1",
    "PM25_P2" = "firebrick2",
    "PM25_P3" = "firebrick3",
    "PM25_P4" = "firebrick4",
    "PM25_P5" = "#5A0F0F"
  )) +
  theme_bw(base_size = 16) + # Tema blanco y negro con tamño de texto 16
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5), #Título en negrita y centrado
    axis.text.x = element_text(angle = 45, hjust = 1) # Inclina el eje para evitar solapamiento
  ) +
  labs(title = "Error absoluto - Calibrados",
        x = "sensor", y = "Error absoluto (µg/m³)",
        fill = "sensor")

#Cargar datos -----
Errores_antes_despues<- read.csv2( "C:/Semestre 2-2025/Visualizacion/Página web/Pagina web/Data/Errores_antes_despues.csv",
                                     header = TRUE
)
#Grafico de error antes y -----

```

```

barras_MGE <- ggplot(Errores_anter_despues,
                      aes(x = Sensor, y = MGE, fill = Clasificación)) +
  geom_bar(stat = "identity", color = "black", # añade un borde negro a cada grafica
           position = position_dodge(width = 0.9)) +
  geom_errorbar(aes(ymin = MGE - SD_MGE, ymax = MGE + SD_MGE),# añade la sd
                width = 0.2, linewidth = 0.8,
                position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 20, # Linea en 20 discontinua
             linetype = "dashed",
             color = "firebrick4",
             linewidth = 1) +    #grosor
  scale_fill_manual(values = c(    # Para especificar los colores a emplear
    "No calibrados" = "#1E90FF",
    "Calibrados"     = "firebrick1"
  )) +
  theme_bw(base_size = 16) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),# Título en negrita y centrado.
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.title = element_blank()
  ) +
  labs(title = "MGE antes / después de calibrar",
       x = "Sensor", y = "MGE (µg/m³)")
# Gráfico de barras MB -----

```



```

barras_MB <- ggplot(Errores_anter_despues,
                      aes(x = Sensor, y = MB, fill = Clasificación)) +
  geom_bar(stat = "identity", color = "black", # añade un borde negro a cada grafica
           position = position_dodge(width = 0.9)) +
  geom_errorbar(aes(ymin = MB - SD_MB, ymax = MB + SD_MB),# añade los valores de desviación
                width = 0.2, linewidth = 0.8,
                position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 20, # Linea en 20 discontinua
             linetype = "dashed",
             color = "firebrick4",
             linewidth = 1) +
  scale_fill_manual(values = c(
    "No calibrados" = "#1E90FF",
    "Calibrados"     = "firebrick1"
  )) +
  theme_bw(base_size = 16) +    # Tema blanco y negro con tamaño de texto 16

```

```

theme(
  plot.title = element_text(face = "bold", hjust = 0.5),# Título en negrita y centrado.
  axis.text.x = element_text(angle = 45, hjust = 1),
  legend.title = element_blank()
) +
  labs(title = "MB antes / después de calibrar",
       x = "Sensor", y = "MB ( $\mu\text{g}/\text{m}^3$ )")

```

```

# Cargar librerias -----
library(patchwork)

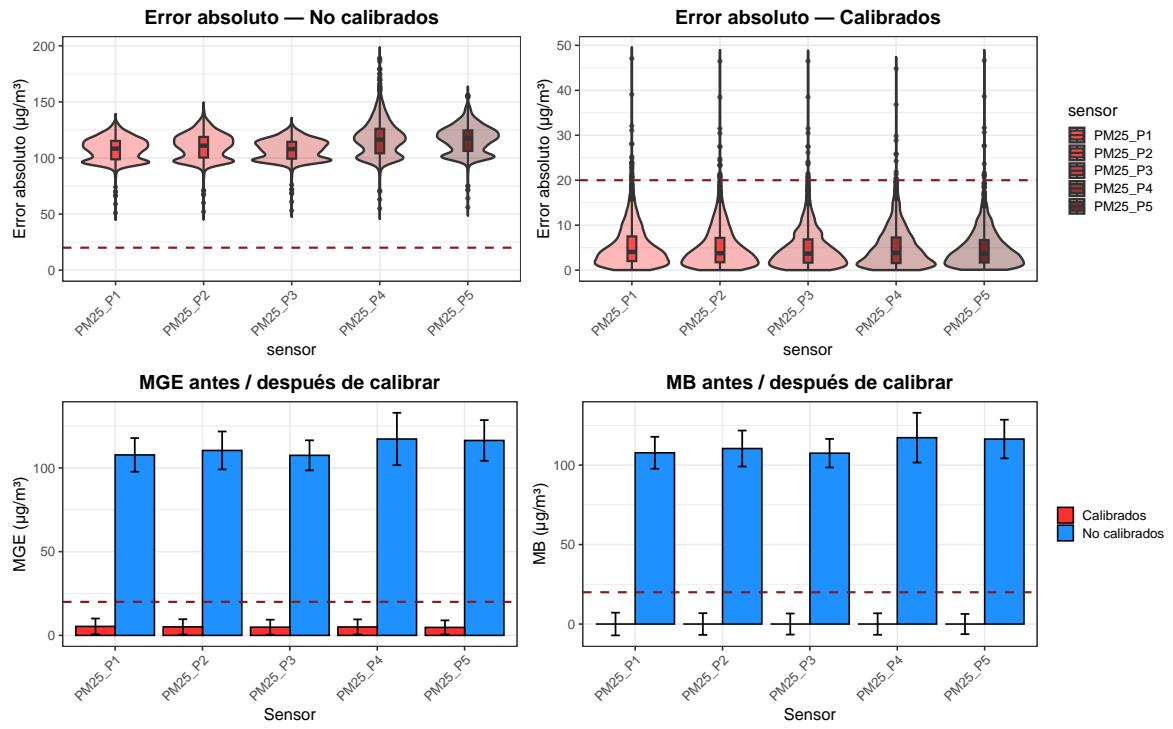
# Especificar una leyenda para los violines -----
figura_violines <- (violin_nocalibrado | violin_calibrado) + # Especifica el orden en la cu
  plot_layout(guides = "collect") & # Una leyenda
  theme(legend.position = "right")    # leyendas a la derecha

# Especificar una leyenda para los graficos de barras -----
figura_barras <- (barras_MGE | barras_MB) +
  plot_layout(guides = "collect") &      # Una sola leyenda para las barras
  theme(legend.position = "right")        # leyendas a la derecha

# Unir figuras y especificar orden -----
figura_final <- figura_violines / figura_barras

# Ver figura -----
figura_final

```



La figura se evidencia que al aplicarle un modelo de regresión lineal a los sendores de bajo costo todos los valores de errores medios estuvieron por debajo de 20 $\mu\text{g}/\text{m}^3$.

Conclusiones

1. Los sensores de bajo costo tienden a sobreestimar la concentración de PM_{2,5}.
2. Un modelo de regresión lineal puede representar el comportamiento de las concentraciones de PM_{2,5} con un ajuste de ($R^2=0,87-0,9$), permitiendo obtener valores de error de estimación del modelo por debajo de 20 $\mu\text{g}/\text{m}^3$.