Relatório Técnico Interativo — Qualidade do Ar (RMC)

Equipe Qualidade do Ar RMC

2025-11-15

Table of contents

1 Visão Geral

Este relatório apresenta a caracterização da qualidade do ar e meteorologia em municípios selecionados da Região Metropolitana de Curitiba (RMC), com base em:

- Sensores GM-5000 (gases + PM)
- Sensores PurpleAir (PM2.5)
- Dados meteorológicos (vento, temperatura, umidade, precipitação, pressão, radiação)
- Cálculo e representação de AQI / diretrizes da OMS

Conteúdos principais: 1. Fontes e estrutura dos dados 2. Séries e climatologia meteorológica 3. Integração com dashboard Shiny 4. Padrões temporais hora × dia da semana 5. Influência meteorológica (polar plots / rosas) 6. Mapas de sensores e concentrações médias 7. Estatísticas resumidas (`aqStats`) 8. Anexos (sessão, tabelas WHO / AQI)

O dashboard interativo está disponível também como capítulo (iframe) e/ou separado via shinyapps.io.

2 Dados e Métodos

```
library(tidyverse)
library(DT)
```

2.1 Objetos carregados

```
ls()
```

```
character(0)
```

2.1.1 Meteorologia (horária)

```
if (exists("meteo_hour")) dplyr::glimpse(meteo_hour)
```

2.1.2 Qualidade do ar — diário (GM-5000)

```
if (exists("Datafinal")) dplyr::glimpse(Datafinal)
```

2.1.3 Qualidade do ar — horário (GM-5000)

```
if (exists("air_quality_data_ugm3")) dplyr::glimpse(air_quality_data_ugm3)
```

2.1.4 PurpleAir (PM2.5 diário corrigido)

```r
if (exists("purpleair")) dplyr::glimpse(purpleair)
```

2.1.5 Tabelas auxiliares (AQI e precauções)

```r
if (exists("Poltab")) head(Poltab)
if (exists("AQItab")) head(AQItab)
```

2.2 Diretrizes OMS

```r
if (exists("who_limit_all")) who_limit_all
```

2.3 Nota sobre horários e TZ

Todas as datas são normalizadas para "America/Sao_Paulo". Manter consistência é essencial para agregações diárias e cálculo de médias móveis (ex.: O3 8h).

3 Caracterização Meteorológica

```
library(openair)
library(ggplot2)
library(scales)
```

3.0.1 Séries horárias

```
if (exists("meteo_hour")) {
  meteo_hour %>%
    filter(Cidade %in% c("Rio Branco do Sul", "Almirante Tamandaré")) %>%
    openair::timePlot(pollutant = c("press","temp","rad","umid","prec","ws"),
                      y.relation = "free")
}
```

3.0.2 Rosa de vento (velocidade como variável)

```
if (exists("meteo_hour")) {
  meteo_hour %>%
    filter(Cidade %in% c("Rio Branco do Sul","Almirante Tamandaré")) %>%
    openair::pollutionRose(pollutant = "ws", type = "Cidade",
                           hemisphere = "southern",
                           key.footer = "Velocidade do vento (m/s)")
}
```

3.0.3 Sazonalidade (vento)

```
if (exists("meteo_hour")) {
  meteo_hour %>%
    filter(Cidade %in% c("Rio Branco do Sul","Almirante Tamandaré")) %>%
```

```
    openair::cutData(hemisphere = "southern", type = "season") %>%
    openair::pollutionRose(pollutant = "ws", type = c("season","Cidade"),
                           hemisphere = "southern",
                           key.footer = "Velocidade do vento (m/s)")
}
```

3.0.4 Climatologia diária simplificada

```
if (exists("meteo_hour")) {
  meteo_day <- meteo_hour %>%
    mutate(date = lubridate::floor_date(date,"day")) %>%
    group_by(Cidade, date) %>%
    summarise(temp = mean(temp, na.rm=TRUE),
              umid = mean(umid, na.rm=TRUE),
              prec = sum(prec, na.rm=TRUE),
              .groups="drop")
  ggplot(meteo_day, aes(x=date)) +
    geom_col(aes(y=prec*10, fill="Precipitação"), alpha=0.4) +
    geom_line(aes(y=temp, color="Temperatura"), linewidth=0.6) +
    geom_line(aes(y=umid, color="Umidade Relativa"), linewidth=0.6) +
    scale_y_continuous(sec.axis = sec_axis(~./10, name="Precipitação (mm)")) +
    facet_wrap(.~Cidade, scales="free_y") +
    scale_fill_manual(values="darkgreen") +
    scale_color_manual(values=c("Temperatura"="red","Umidade Relativa"="blue")) +
    theme_bw() +
    labs(y="Temp (°C) / UR (%)", x="Data", fill="", color="", title="Climatologia diária (Ter
    scale_x_date(labels = scales::label_date_short(), breaks="1 month")
}
```

4 Dashboard Shiny Integrado

O dashboard interativo faz cálculo de AQI, exibe mapas leaflet, séries temporais e correlações.

4.0.1 Versão pública (shinyapps.io)

Dashboard em produção

4.0.2 Iframe

4.0.3 Código-Fonte

O arquivo `app.R` incluído no repositório pode ser executado localmente após carregar dependências:

```
shiny::runApp()
```

Principais pontos: - Carregamento global dos dados - Uso de `pollutionRose`, `aqStats`, `polarMap` - Escalas de cores do AQI com `colorBin` - Evita repetição de IDs (corrigido) - Integra PurpleAir + GM-5000 num único objeto combinado

5 Padrões Temporais da Qualidade do Ar

```
library(dplyr)
library(ggplot2)
library(lubridate)
library(ggh4x)
library(reshape2)
```

5.0.1 Padrão hora × dia da semana (GM-5000 horário)

```
if (exists("air_quality_data_ugm3")) {
  gm_hour <- air_quality_data_ugm3 %>%
    mutate(hr = hour(date),
           wday = wday(date, label = TRUE)) %>%
    pivot_longer(-c(Cidade,date,hr,wday), names_to="Poluente", values_to="Valor") %>%
    group_by(Cidade, Poluente, wday, hr) %>%
    summarise(Media = mean(Valor, na.rm=TRUE), .groups="drop")

  ggplot(gm_hour, aes(hr, Media, colour=wday)) +
    geom_line(linewidth=0.4) +
    facet_nested(Poluente ~ Cidade, scales="free_y") +
    scale_x_continuous(breaks=seq(0,24,4)) +
    theme_classic() +
    labs(x="Hora", y="Média horária", colour="Dia", title="Padrões hora × dia da semana (GM-5
}
```

5.0.2 Padrão hora × dia da semana (PurpleAir)

```
if (exists("purpleair")) {
  pa_hour <- purpleair %>%
    mutate(hr = hour(Date),
           wday = wday(Date, label = TRUE)) %>%
```

```
    group_by(Cidade, sensor_id, wday, hr) %>%
    summarise(PM25 = mean(PM2.5, na.rm=TRUE), .groups="drop")

  ggplot(pa_hour, aes(hr, PM25, colour=wday)) +
    geom_line(linewidth=0.4) +
    facet_nested(. ~ Cidade + sensor_id, scales="free_y") +
    scale_x_continuous(breaks=seq(0,24,4)) +
    theme_classic() +
    labs(x="Hora", y="PM2.5 (µg/m³)", colour="Dia", title="Padrões hora × dia da semana (Pur
}
```

5.0.3 Comparação com Limites OMS (diário GM-5000)

```
if (exists("Datafinal")) {
  df_day <- Datafinal %>%
    select(Cidade, Date, SO2, NO2, PM2.5, PM10, O3, CO) %>%
    pivot_longer(-c(Cidade, Date), names_to="Poluente", values_to="Valor")

  df_mean <- df_day %>%
    group_by(Cidade, Poluente) %>%
    summarise(Media = mean(Valor, na.rm=TRUE), .groups="drop")

  ggplot(df_day, aes(Date, Valor)) +
    stat_summary(fun="mean", geom="col", fill="grey70") +
    geom_hline(data=df_mean, aes(yintercept=Media, linetype="média período"), colour="black"
    facet_wrap(~Poluente, scales="free_y") +
    scale_x_date(breaks="1 month", labels=scales::label_date_short()) +
    theme_bw() +
    labs(x="Data", y="Concentração", linetype="", title="Médias diárias por poluente") +
    theme(legend.position="bottom")
}
```

6 Meteorologia e Potenciais Fontes (Polar / Vento)

```
library(openair)
```

6.0.1 Polar plots (exemplo NO2)

```
if (exists("air_quality_data_ugm3") && exists("meteo_hour")) {
  joined <- air_quality_data_ugm3 %>%
    left_join(meteo_hour %>% select(date, ws, wd, Cidade), by="date")

  joined %>%
    filter(Cidade == "Rio Branco do Sul") %>%
    openair::polarPlot(pollutant="NO2", ws="ws", wd="wd", main="NO2 vs. vento - RBS")
}
```

6.0.2 `pollutionRose` aplicado a PM2.5 sintetizado

```
if (exists("purpleair") && exists("meteo_hour")) {
  pa_join <- purpleair %>%
    left_join(meteo_hour %>% select(date, ws, wd, Cidade), by=c("Date"="date"))

  pa_join %>%
    openair::pollutionRose(pollutant="PM2.5", type="Cidade", hemisphere="southern",
                           key.footer="PM2.5 (µg/m³)")
}
```

7 Mapas Interativos

```
library(leaflet)
```

Caso os dados de localização estejam no objeto `localizacao`, geramos mapa interativo dos sensores.

```
if (exists("localizacao")) {
  df_map <- localizacao %>%
    mutate(popup = paste0("<b>", Local, "</b><br/>Cidade: ", Cidade,
                          "<br/>Tipo: ", Tipo))
  pal <- colorFactor(RColorBrewer::brewer.pal(5,"Set2"), df_map$Cidade)

  leaflet(df_map) %>%
    addTiles() %>%
    addCircleMarkers(~Long, ~Lat,
                     color=~pal(Cidade),
                     radius=6,
                     stroke=TRUE, weight=1,
                     fillOpacity=0.8,
                     popup=~popup) %>%
    addLegend("bottomright", pal=pal, values=~Cidade, title="Cidade")
}
```

8 Estatísticas Resumidas

```
library(openair)
library(DT)
```

8.0.1 aqStats (GM-5000)

```
if (exists("air_quality_data_ugm3")) {
  stats_gm <- air_quality_data_ugm3 %>%
    aqStats(type="Cidade", pollutant=c("SO2","NO2","O3","CO","PM2.5","PM10"))
  DT::datatable(stats_gm)
}
```

8.0.2 aqStats (PurpleAir PM2.5)

```
if (exists("purpleair")) {
  stats_pa <- purpleair %>%
    select(Cidade, Date, PM2.5) %>%
    rename(date = Date) %>%
    aqStats(type="Cidade", pollutant="PM2.5")
  DT::datatable(stats_pa)
}
```

9 Anexos

9.0.1 Tabela AQI

```
if (exists("AQItab")) DT::datatable(AQItab)
```

9.0.2 Poltab (Precauções)

```
if (exists("Poltab")) DT::datatable(Poltab)
```

9.0.3 Sessão

```
sessionInfo()
```

```
R version 4.4.3 (2025-02-28 ucrt)
Platform: x86_64-w64-mingw32/x64
Running under: Windows 11 x64 (build 26100)

Matrix products: default


locale:
[1] LC_COLLATE=Portuguese_Brazil.utf8  LC_CTYPE=Portuguese_Brazil.utf8
[3] LC_MONETARY=Portuguese_Brazil.utf8 LC_NUMERIC=C
[5] LC_TIME=Portuguese_Brazil.utf8

time zone: America/Sao_Paulo
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

```
loaded via a namespace (and not attached):
 [1] compiler_4.4.3    fastmap_1.2.0     cli_3.6.5         tools_4.4.3
 [5] htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10       codetools_0.2-20
 [9] rmarkdown_2.29    knitr_1.50        jsonlite_2.0.0    xfun_0.53
[13] digest_0.6.37     rlang_1.1.6       evaluate_1.0.5
```