# Risk Assessment of Arboviruses in Brazilian State Capitals

2025-08-22

## Content:

## 1. Gathering Data

```r
dengue_full <- read_csv(here("dengue/Dengue_Provaveis_2014_2024.csv")) %>%
  mutate(epidemiologic_week = paste0(year, "w", week))
```

```
## New names:
## Rows: 3444 Columns: 11
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (3): week, municipality, Years.Range dbl (8): ...1, ibge, year, population,
## deaths, likely_cases, cases_pcap, dea...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
# Epidemiological week starting on the first week of the winter
epiweek40 <- read_dta(
  here("dengue/epidemiological_week_startsweek40.dta"))
```

## 2. Preparing Data

### Grouping by municipality and epidemiological week

```
 ## Grouping by municipality and epidemiological week
dengue_df <- dengue_full %>%
  # Correctly group by municipality, year, and week first
  group_by(municipality, year, week) %>%
  summarise(
    cases = sum(likely_cases, na.rm = TRUE),
    population = sum(population, na.rm = TRUE),
    deaths = sum(deaths, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    incidence = ifelse(population > 0, cases / (population / 100000), 0),
    wy = paste0(year, "w", as.integer(week))
  ) %>%
  full_join(epiweek40, by = "wy") %>%
  # IMPORTANT FIX: Group by municipality BEFORE arranging and filling NAs
  group_by(municipality) %>%
  arrange(year_week40, epidemiologic_week_startsweek40) %>%
  mutate(
    # Use across() to apply na.locf cleanly to several columns
    across(c(year, cases, population, deaths, incidence), ~ zoo::na.locf(., na.rm = FALSE)),
    yw = as.character(paste0(year_week40,"-", epidemiologic_week_startsweek40)),
    epi_label = paste0(year_week40, "w", epidemiologic_week_startsweek40)
  ) %>%
  # Ungroup at the end to have a clean, final dataframe
  ungroup()
```

## 3. Risk Assessment

### Function for Risk Assessment

```
perform_risk_analysis <- function(df, analysis_year = 2024) {
  # Defines a function that takes a dataframe 'df' and an 'analysis_year'.

  tryCatch({
    # If an error occurs, the code will jump to the 'error' section at the end.

    # === Part 1: Identify Epidemic Years based on Risk Score ===
    # The goal of this part is to analyze all historical data to determine which years behaved like epi

    # Create a preliminary endemic channel using ALL historical data.
    preliminary_channel <- df %>%
      # 1. Select only data from years before the analysis_year.
      filter(year_week40 < analysis_year) %>%
      # 2. Group data by the epidemiological week number.
      group_by(week2 = epidemiologic_week_startsweek40) %>%
      # 3. For each week, calculate the historical average and standard deviation of incidence.
      summarise(avg_incidence = mean(incidence, na.rm = TRUE), sd_incidence = sd(incidence, na.rm = TRU
      # 4. Calculate the upper alert threshold (95% confidence interval).
      mutate(high = avg_incidence + (1.96 * sd_incidence))
```

```r
# Check if there is enough historical data to proceed.
if(nrow(preliminary_channel) == 0) return(NULL)

# Assess all historical years against the preliminary channel to find epidemic years.
risk_all_years <- df %>%
  # 1. Select only historical data.
  filter(year_week40 < analysis_year) %>%
  # 2. Add the calculated avg_incidence and high threshold to each row based on its week.
  left_join(preliminary_channel, by = c("epidemiologic_week_startsweek40" = "week2")) %>%
  # 3. Group the data by year to analyze each year independently.
  group_by(year_week40) %>%
  # 4. Ensure weeks are sorted chronologically within each year.
  arrange(epidemiologic_week_startsweek40, .by_group = TRUE) %>%
  # 5. Create several new columns (triggers) based on different risk conditions.
  mutate(
    # Trigger: Is incidence above average but below the high threshold? (1=yes, 0=no)
    increasedincidence = ifelse(incidence > avg_incidence & incidence < high, 1, 0),
    # Trigger: Has 'increasedincidence' been true for 4 straight weeks?
    increasedincidence4weeks = zoo::rollapply(increasedincidence, 4, function(x) as.integer(all(x ==
    # Trigger: Is incidence above the high threshold?
    increasedincidencehigh = ifelse(incidence > high, 1, 0),
    # Trigger: Has 'increasedincidencehigh' been true for 4 straight weeks?
    increasedincidencehigh4weeks = zoo::rollapply(increasedincidencehigh, 4, function(x) as.integer
    # Trigger: Did deaths increase from the previous week?
    increaseddeath = ifelse(deaths > lag(deaths), 1, 0),
    # Trigger: Has 'increaseddeath' been true for 4 straight weeks?
    increaseddeath4w = zoo::rollapply(increaseddeath, 4, function(x) as.integer(all(x == 1)), fill =
    # Calculate log of incidence to check for exponential growth.
    ln_incidence = ifelse(incidence > 0, log(incidence), 0),
    # Calculate the week-over-week change in log(incidence).
    exponentialgrowth = ln_incidence - dplyr::lag(ln_incidence),
    # Trigger: Is growth exponential AND is incidence above the high threshold?
    increasedexpgrowth = ifelse(exponentialgrowth > 0 & incidence > high, 1, 0),
    # Trigger: Has 'increaseddeath' been true for 5 straight weeks?
    increaseddeath5w = zoo::rollapply(increaseddeath, 5, function(x) as.integer(all(x == 1)), fill =
  ) %>%
  # Replace any NAs created by the rolling functions (at the start of the series) with 0.
  mutate(across(c(increasedincidence4weeks, increasedincidencehigh4weeks, increaseddeath4w, increase
  # Assign risk levels based on the triggers.
  mutate(
    # Risk Level 1: Sustained increase in cases.
    risk_level1 = if_else(increasedincidence4weeks == 1, 1L, 0L),
    # Risk Level 2: High alert for cases or sustained increase in deaths.
    risk_level2 = if_else(increasedincidencehigh4weeks == 1 | increaseddeath4w == 1, 2L, 0L),
    # Risk Level 3: Epidemic level based on exponential growth or prolonged increase in deaths.
    risk_level3 = if_else(increasedexpgrowth == 1 | increaseddeath5w == 1, 3L, 0L),
    # The final risk assessment for a week is the highest level triggered.
    riskassessment = pmax(risk_level1, risk_level2, risk_level3, na.rm = TRUE)
  )

# From the historical assessment, create a list of years that hit risk level 3.
epidemic_years <- risk_all_years %>% filter(riskassessment >= 3) %>% distinct(year_week40) %>% pull
# Create a list of all available historical years.
```

```
all_hist_years <- unique(df$year_week40[df$year_week40 < analysis_year])
# Create the list of non-epidemic years by removing epidemic years from all historical years.
non_epidemic_years <- all_hist_years[!all_hist_years %in% epidemic_years]
# Select only the 5 most recent non-epidemic years for the final baseline.
non_epidemic_years <- non_epidemic_years %>%
  sort() %>%
  tail(5)

# Fallback condition:  Skips the municipality if there are not enough non-epidemic years to build a
if (length(non_epidemic_years) < 2) {

  message(paste("Skipping municipality:",
                unique(df$municipality),
                "| Reason: Fewer than 2 non-epidemic years found."))

  return(NULL)
}

# === Part 2: Build Final Control Diagram and Assess the Current Year ===
# The goal of this part is to use the clean "non-epidemic" years to build a
# final, reliable baseline and assess the current analysis_year against it.

# Isolate the incidence and death data for the current analysis_year.
incidence_current_year <- df %>% filter(year_week40 == analysis_year) %>% transmute(incidence_curre

# Build the final, clean endemic channel.
control_data <- df %>%
  # 1. IMPORTANT: Filter for only the selected non-epidemic years.
  filter(year_week40 %in% non_epidemic_years) %>%
  # 2. Group by week.
  group_by(week2 = epidemiologic_week_startsweek40) %>%
  # 3. Recalculate the average and standard deviation using this cleaner data.
  summarise(avg_incidence = mean(incidence, na.rm = TRUE), sd_incidence = sd(incidence, na.rm = TRU
  # 4. Create the final upper and lower thresholds for the control diagram.
  mutate(high = avg_incidence + 1.96*sd_incidence, low = pmax(0, avg_incidence - 1.96*sd_incidence)
  # 5. Join the final channel data with the current year's data.
  left_join(incidence_current_year, by = "week2")

# Perform the final risk assessment for the analysis_year.
risk_final <- control_data %>%
  arrange(week2) %>%
  mutate(
    increasedincidence = ifelse(incidence_current > avg_incidence & incidence_current < high, 1, 0)
    increasedincidence4weeks = zoo::rollapply(increasedincidence, 4, function(x) as.integer(all(x==
    increasedincidencehigh = ifelse(incidence_current > high, 1, 0),
    increasedincidencehigh4weeks = zoo::rollapply(increasedincidencehigh, 4, function(x)as.integer(
    increaseddeath = ifelse(deaths > lag(deaths), 1, 0),
    increaseddeath4w = zoo::rollapply(increaseddeath, 4, function(x) as.integer(all(x==1)), fill=NA
    ln_incidence = ifelse(incidence_current > 0, log(incidence_current), 0),
    exponentialgrowth = ln_incidence - dplyr::lag(ln_incidence),
    increasedexpgrowth = ifelse(exponentialgrowth > 0 & incidence_current > high, 1, 0),
    increaseddeath5w = zoo::rollapply(increaseddeath, 5, function(x) as.integer(all(x==1)), fill=NA
  ) %>%
      mutate(across(c(increasedincidence4weeks, increasedincidencehigh4weeks, increaseddeath4w, increase
      mutate(
        risk_level1 = if_else(increasedincidence4weeks == 1, 1L, 0L),
        risk_level2 = if_else(increasedincidencehigh4weeks == 1 | increaseddeath4w == 1, 2L, 0L),
        risk_level3 = if_else(increasedexpgrowth == 1 | increaseddeath5w == 1, 3L, 0L),
        riskassessment = pmax(risk_level1, risk_level2, risk_level3, na.rm = TRUE)
      )

    # Return the results as a list containing two objects.
    return(list(risk_final = risk_final, epidemic_years = epidemic_years))

  }, error = function(e) {
    # If any error occurred in the 'try' block, this code will run.
    message(paste("Could not process municipality:", unique(df$municipality), "| Error:", e$message))
    return(NULL)
  })
}
```

### Running risk assessment

```
# Nest the data, creating a list-column with a dataframe for each municipality
nested_data <- dengue_df %>%
  group_by(municipality) %>%
  nest()

# Apply function to each nested dataframe
results <- nested_data %>%
  mutate(risk_analysis_results = map(data, perform_risk_analysis))

# Unnest the results for the 2024 risk assessment
risk_2024 <- results %>%
  # Safely extract the 'risk_final' tibble from the list
  mutate(risk_data = map(risk_analysis_results, ~ .x$risk_final)) %>%
  select(municipality, risk_data) %>%
  unnest(cols = c(risk_data))
```

### Table of epidemic years per municipality

```
# Helper function to safely extract epidemic years from the results list
extract_epidemic_years <- function(model_output) {
  if (!is.null(model_output) && "epidemic_years" %in% names(model_output)) {
    if (length(model_output$epidemic_years) > 0) {
      return(tibble(epidemic_year = as.character(model_output$epidemic_years)))
    }
  }
  return(tibble(epidemic_year = character(0)))
}

# Extract and unnest the epidemic years data
```

```r
epidemic_years_data <- results %>%
  mutate(epidemic_years = map(risk_analysis_results, extract_epidemic_years)) %>%
  select(municipality, epidemic_years) %>%
  unnest(cols = c(epidemic_years))

# Create a summary table
epidemic_summary_table <- epidemic_years_data %>%
  group_by(municipality) %>%
  summarise(
    epidemic_years_list = paste(sort(unique(epidemic_year)), collapse = ", "),
    .groups = "drop"
  ) %>%
  rename(
    `Municipality` = municipality,
    `Epidemic Years` = epidemic_years_list
  )

# Table
knitr::kable(epidemic_summary_table, caption = "Epidemic Years of Dengue by Municipality")
```

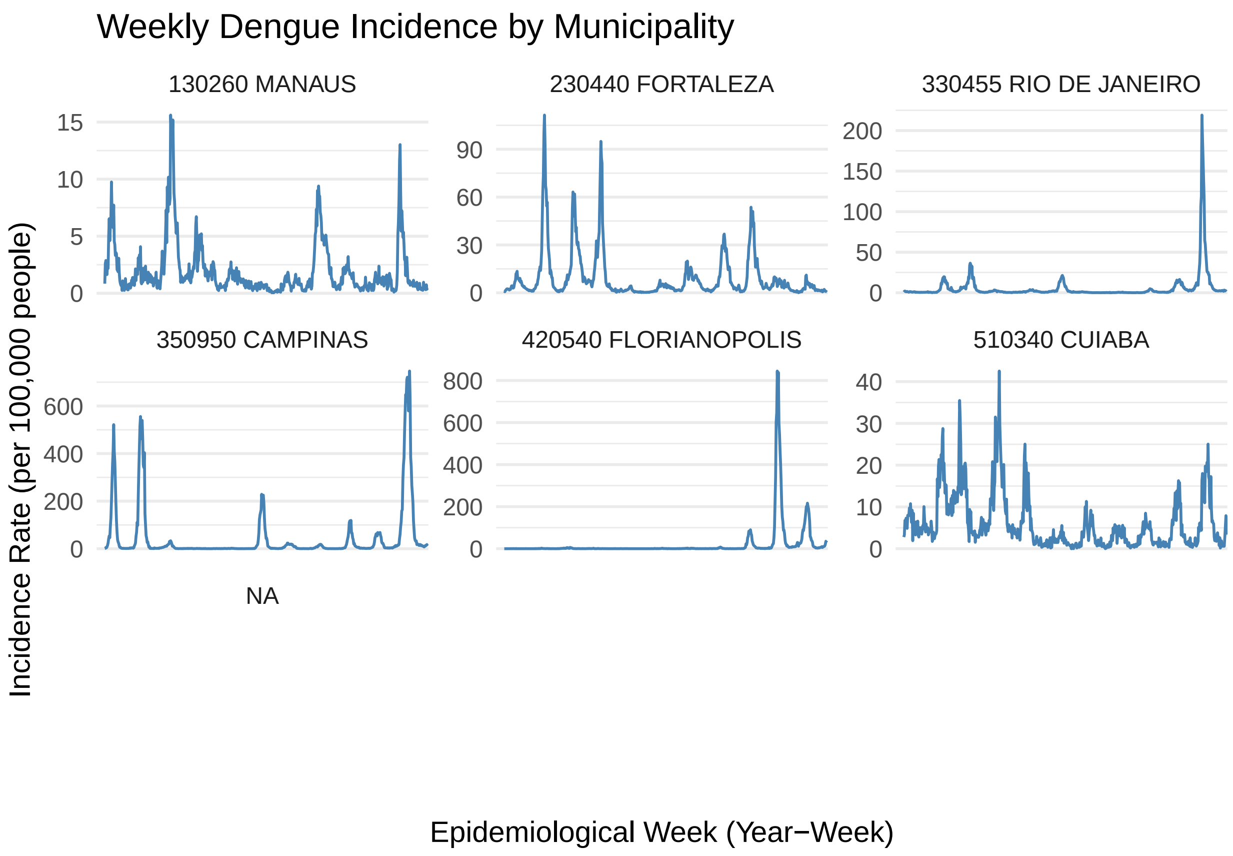
Tabela 1: Epidemic Years of Dengue by Municipality

| Municipality | Epidemic Years |
|---|---|
| 130260 MANAUS | 2016, 2017 |
| 230440 FORTALEZA | 2015, 2016, 2017 |
| 330455 RIO DE JANEIRO | 2015, 2016, 2023 |
| 350950 CAMPINAS | 2014, 2015, 2016, 2019, 2022, 2023 |
| 420540 FLORIANOPOLIS | 2023 |
| 510340 CUIABA | 2015, 2016, 2017 |

# 4. Descriptive graphs of incidence and risk

## Plotting Incidence per Municipality per Epidemiological Week

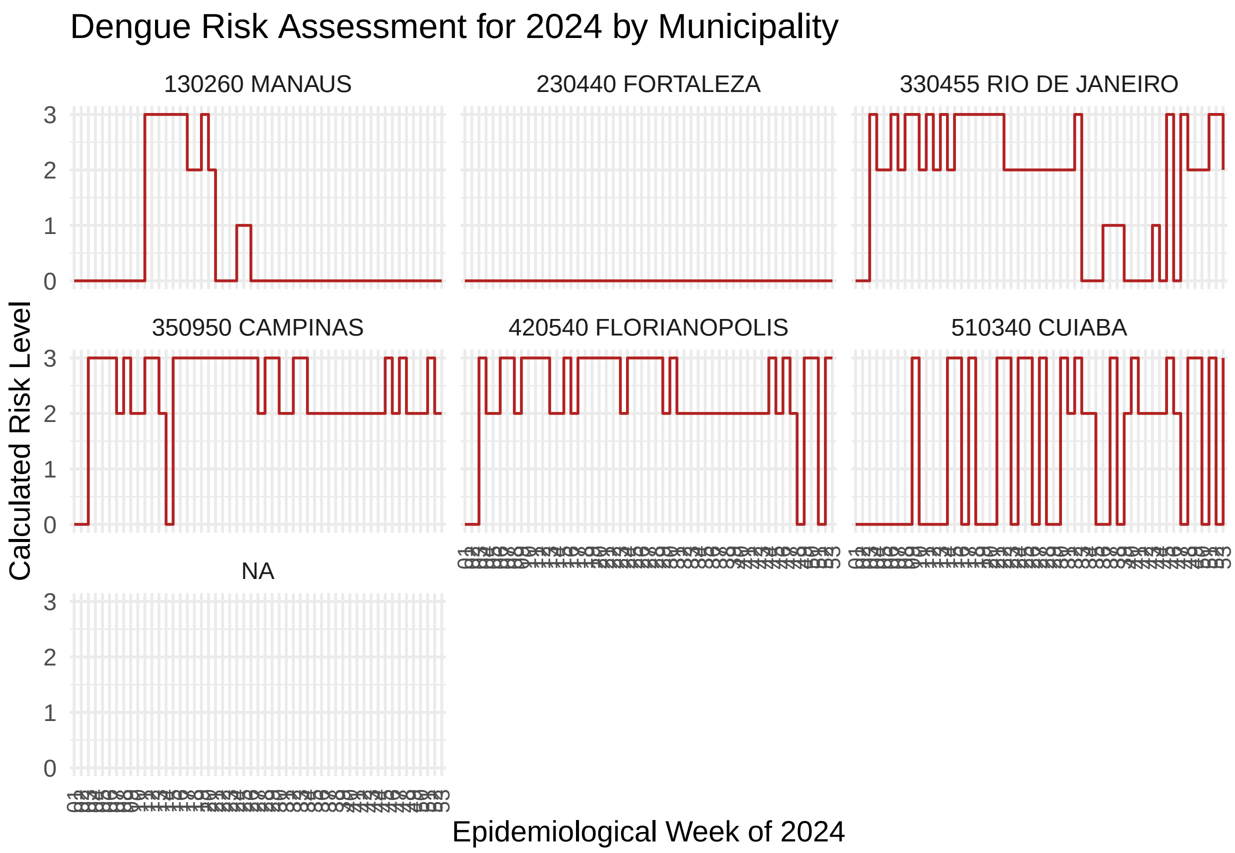
```r
ggplot(dengue_df, aes(x = yw, y = incidence, group = 1)) +
  geom_line(color = "steelblue") +
  facet_wrap(~ municipality, scales = "free_y") +
  labs(
    title = "Weekly Dengue Incidence by Municipality",
    x = "Epidemiological Week (Year-Week)",
    y = "Incidence Rate (per 100,000 people)"
  ) +
  scale_x_discrete(breaks = levels(dengue_df$yw)[seq(1, length(levels(dengue_df$yw)))]) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 8))
```



### Plotting Risk per Municipality per Epidemiological Week

```r
ggplot(risk_2024, aes(x = week2, y = riskassessment, group = 1)) +
  # geom_step is great for visualizing discrete changes in risk level
  geom_step(color = "firebrick") +
  # Create a separate plot for each municipality
  facet_wrap(~ municipality) +
  labs(
    title = "Dengue Risk Assessment for 2024 by Municipality",
    x = "Epidemiological Week of 2024",
    y = "Calculated Risk Level"
  ) +
  scale_y_continuous(breaks = 0:3, limits = c(0, 3)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 8))
```

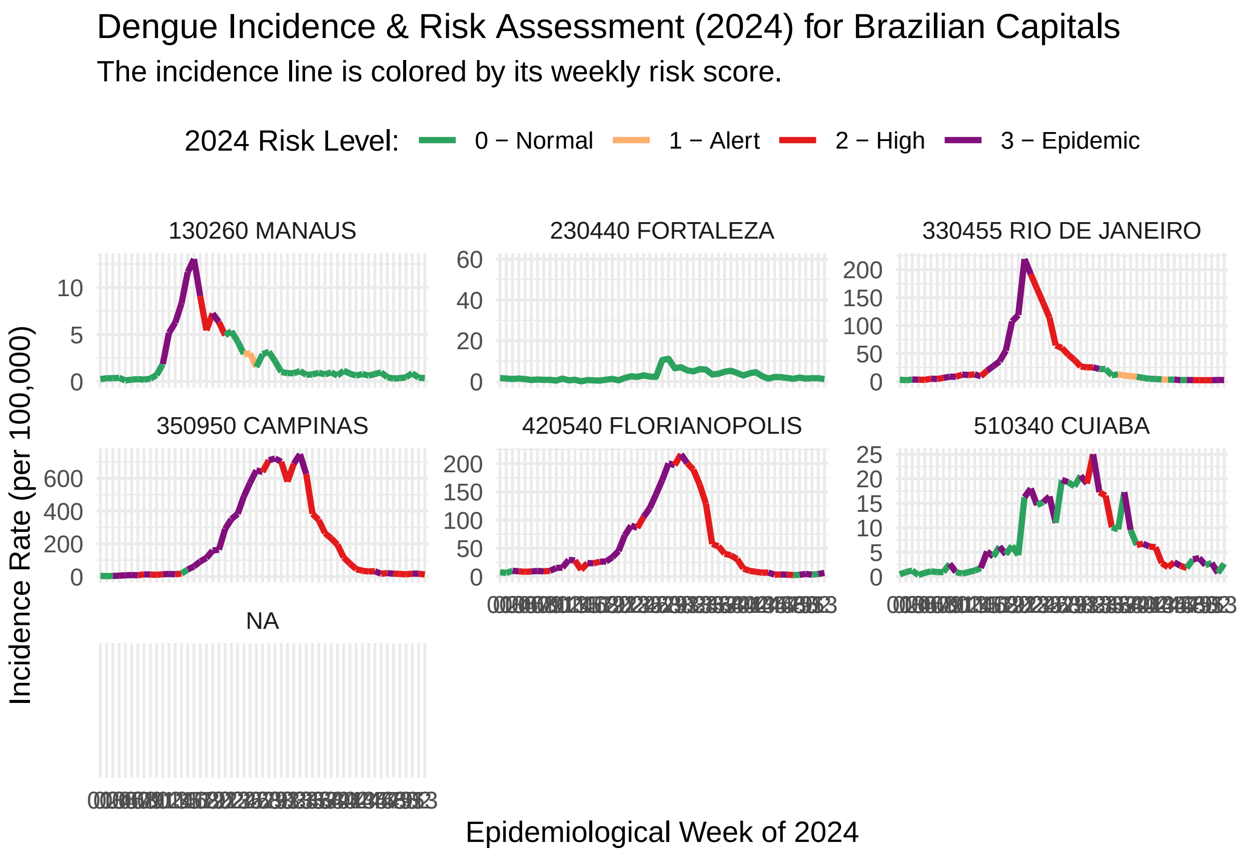
```
## `geom_path()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



### Combining the graphs

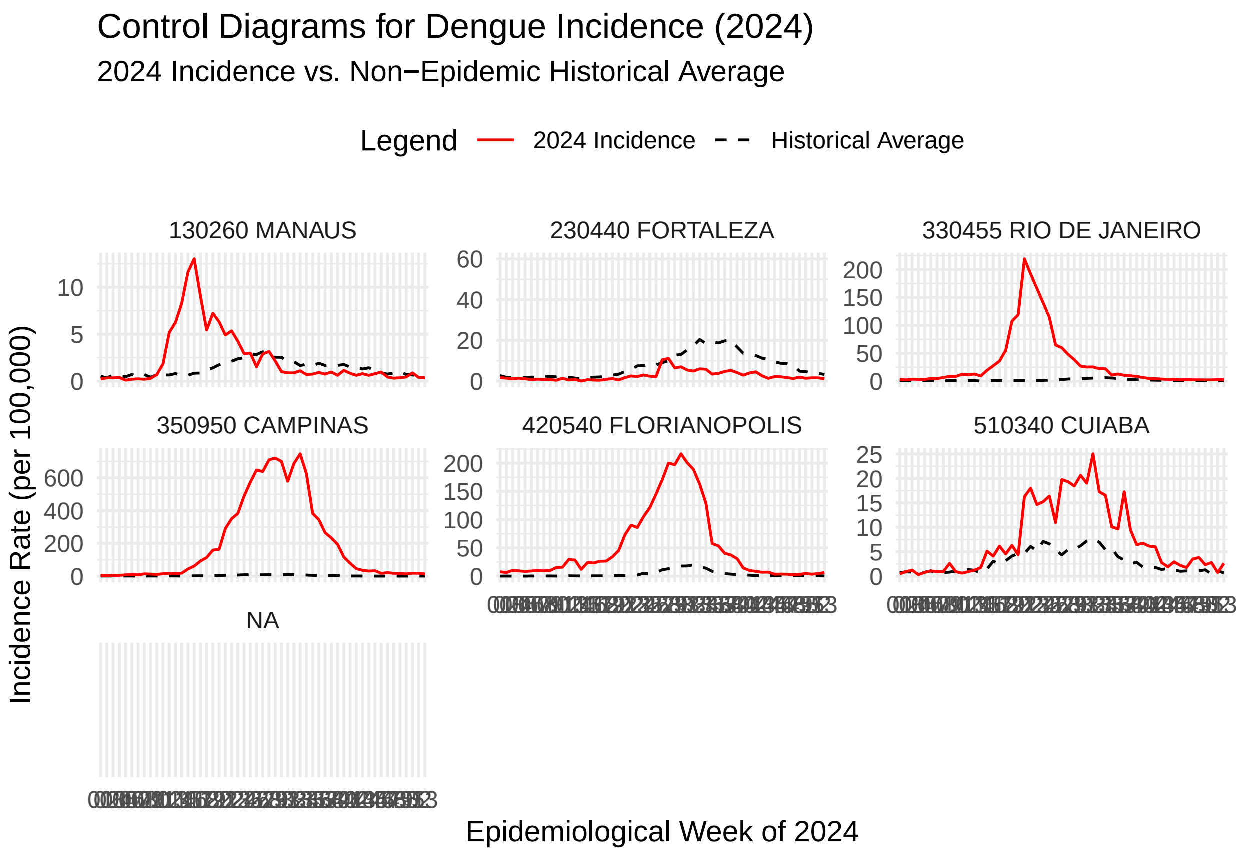
```r
risk_colors <- c(
  "0" = "#2ca25f", # Green
  "1" = "#fdae6b", # Orange
  "2" = "#e31a1c", # Red
  "3" = "#810f7c"  # Purple for the highest risk
)

ggplot(risk_2024, aes(x = week2)) +

  # The historical endemic channel (grey ribbon and dashed line)
  geom_ribbon(aes(ymin = low, ymax = high), fill = "grey80", alpha = 0.6) +
  geom_line(aes(y = avg_incidence), linetype = "dashed", color = "black") +

  # The 2024 incidence line, with color mapped to risk level
  geom_line(aes(y = incidence_current, color = as.factor(riskassessment), group = 1), size = 1.1) +

  # Creating a grid of plots
  facet_wrap(~ municipality, scales = "free_y") +

  # Apply custom color scale and labels
  scale_color_manual(
    name = "2024 Risk Level:",
    values = risk_colors,
    labels = c("0 - Normal", "1 - Alert", "2 - High", "3 - Epidemic")
  ) +

  # Update titles and labels for the combined plot
  labs(
    title = "Dengue Incidence & Risk Assessment (2024) for Brazilian Capitals",
    subtitle = "The incidence line is colored by its weekly risk score.",
    y = "Incidence Rate (per 100,000)",
    x = "Epidemiological Week of 2024"
  ) +
  theme_minimal() +
  theme(legend.position = "top")
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



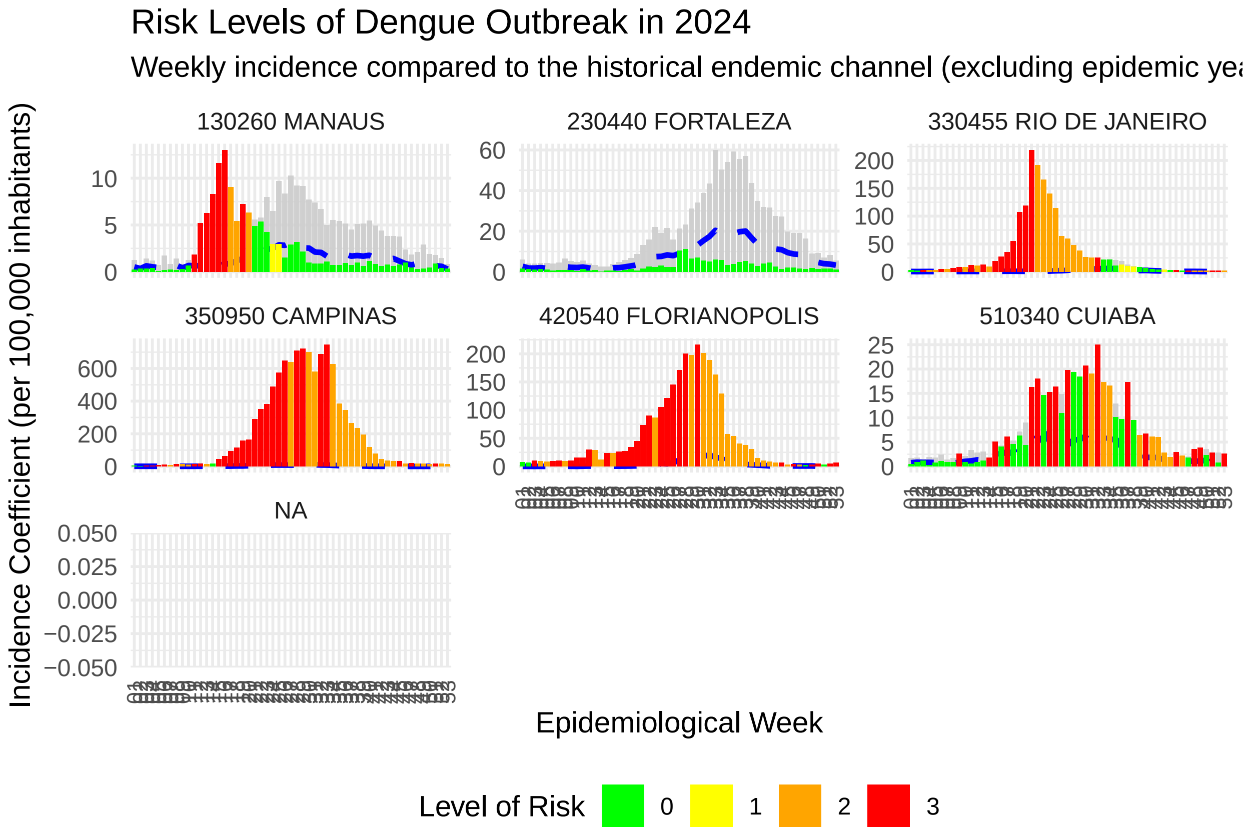
## 5. Control Diagrams (for 2024, without epidemic years)

```r
ggplot(risk_2024, aes(x = week2)) +

  # The historical endemic channel (grey ribbon)
  geom_ribbon(aes(ymin = low, ymax = high), fill = "lightblue", alpha = 0.8) +

  # The historical average line
  geom_line(aes(y = avg_incidence, linetype = "Historical Average", group = 1), color = "black") +

  # The 2024 incidence line
  geom_line(aes(y = incidence_current, linetype = "2024 Incidence", group = 1), color = "red") +

  # Create a grid of plots for each municipality
  facet_wrap(~ municipality, scales = "free_y") +

  # Customize the legend for clarity
  scale_linetype_manual(
    name = "Legend",
    values = c("Historical Average" = "dashed", "2024 Incidence" = "solid")
  ) +

  # Add titles and labels
  labs(
    title = "Control Diagrams for Dengue Incidence (2024)",
    subtitle = "2024 Incidence vs. Non-Epidemic Historical Average",
    y = "Incidence Rate (per 100,000)",
    x = "Epidemiological Week of 2024"
  ) +
  theme_minimal() +
  theme(legend.position = "top")
```



## 6. Risk Levels of Dengue in 2024

```
# Define the colors for each risk level
risk_colors <- c(
  "0" = "green", # Green
  "1" = "yellow", # Orange
  "2" = "orange", # Red
  "3" = "red"   # Purple for the highest risk
)

# Create the plot using a bar chart style
ggplot(risk_2024, aes(x = week2)) +
  # 1. Gray bars representing the upper limit of the endemic channel
  geom_col(aes(y = high), fill = "grey80", alpha = 0.9) +

  # 2. Blue dashed line for the historical average
  geom_line(aes(y = avg_incidence, group = 1), color = "blue", linetype = "dashed", size = 1) +

  # 3. Stacked, colored bars for the 2024 incidence, with the color based on risk level
  geom_col(aes(y = incidence_current, fill = as.factor(riskassessment))) +

  # Create a separate plot for each municipality
  facet_wrap(~ municipality, scales = "free_y") +

  # Apply the custom color scale
  scale_fill_manual(
    name = "Level of Risk",
    values = risk_colors
  ) +

  # Add titles and labels
  labs(
    title = "Risk Levels of Dengue Outbreak in 2024",
    subtitle = "Weekly incidence compared to the historical endemic channel (excluding epidemic years)"
    y = "Incidence Coefficient (per 100,000 inhabitants)",
    x = "Epidemiological Week"
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 90, vjust = 0.5, size = 8)
  )
```


Risk Levels of Dengue Outbreak in 2024

Weekly incidence compared to the historical endemic channel (excluding epidemic yea