Script R - Parte 3 - Thiago Menezes

1) Importação do conjunto de dados “dataset_st_cenarios” no RStudio:

2) Conversão dos dados para em um objeto tsibble:

```
install.packages("dplyr")
install.packages("tsibble")
install.packages("fabletools")
install.packages("ggplot2")

library(dplyr)
library(tsibble)
library(fabletools)
library(ggplot2)

dataset_st <- dataset_st_cenarios |>
  mutate(mes = yearmonth(mes)) |>
  as_tsibble(index = mes)

dataset_st
```

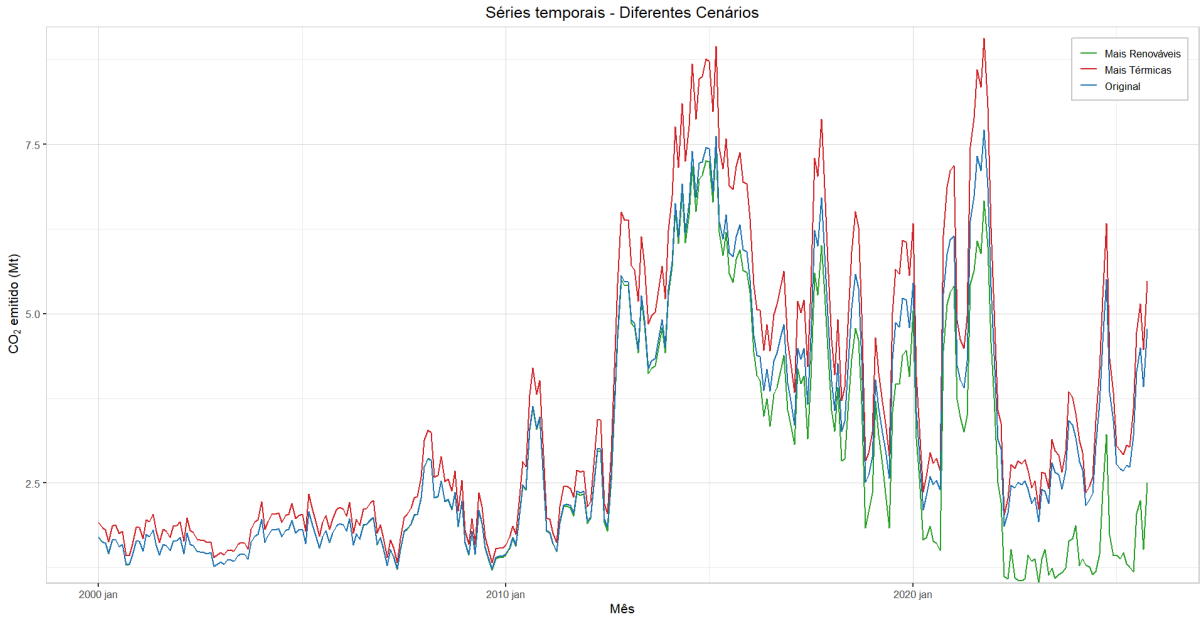
```
> dataset_st
# A tsibble: 310 x 4 [1M]
        mes soma1 soma2r soma3t
      <mth> <dbl>  <dbl>  <dbl>
 1 2000 jan  1.70   1.70   1.92
 2 2000 fev  1.64   1.63   1.84
 3 2000 mar  1.61   1.61   1.81
 4 2000 abr  1.46   1.46   1.63
 5 2000 mai  1.66   1.66   1.87
 6 2000 jun  1.67   1.67   1.88
 7 2000 jul  1.56   1.56   1.75
 8 2000 ago  1.59   1.59   1.79
 9 2000 set  1.30   1.29   1.43
10 2000 out  1.30   1.30   1.43
# i 300 more rows
# i Use `print(n = ...)` to see more rows
```

3) Geração do gráfico da emissão de CO2 por cenários:

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(grid)   # para unit()

# ------------------------------------------------
# Paleta "tab10" do Matplotlib (igual Script Parte 2)
# ------------------------------------------------
tab10 <- c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
```

```
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
  "#bcbd22", "#17becf"
)

# ------------------------------------------------
# Tema estilo Python / Matplotlib (igual Script Parte 2)
# ------------------------------------------------
theme_python <- function(base_size = 14) {
  theme_bw(base_size = base_size) +
    theme(
      panel.background = element_rect(fill = "white"),
      panel.grid.major = element_line(colour = "grey85", linewidth = 0.4),
      panel.grid.minor = element_line(colour = "grey92", linewidth = 0.2),
      panel.border     = element_rect(colour = "grey80", fill = NA),
      plot.title       = element_text(hjust = 0.5),
      legend.title     = element_blank()
    )
}

# Define tema global (como no Script Parte 2)
theme_set(theme_python())

# ------------------------------------------------
# Séries temporais dos cenários (Original / Renováveis / Térmicas)
# ------------------------------------------------
dataset_st %>%
  select(mes, soma1, soma2r, soma3t) %>%
  pivot_longer(-mes, names_to = "grupo", values_to = "valor") %>%
  mutate(
    grupo = recode(
      grupo,
      soma1  = "Original",
      soma2r = "Mais Renováveis",
      soma3t = "Mais Térmicas"
    )
  ) %>%
  ggplot(aes(x = mes, y = valor, colour = grupo)) +
  geom_line(linewidth = 0.9) +
  # Azul, verde e vermelho da paleta tab10
  scale_colour_manual(values = c(
    "Original"        = tab10[1],  # azul
    "Mais Renováveis" = tab10[3],  # verde
    "Mais Térmicas"   = tab10[4]   # vermelho
  )) +
  labs(
    y = expression("CO"[2] * " emitido (Mt)"),
    x = "Mês",
    title = "Séries temporais - Diferentes Cenários"
```

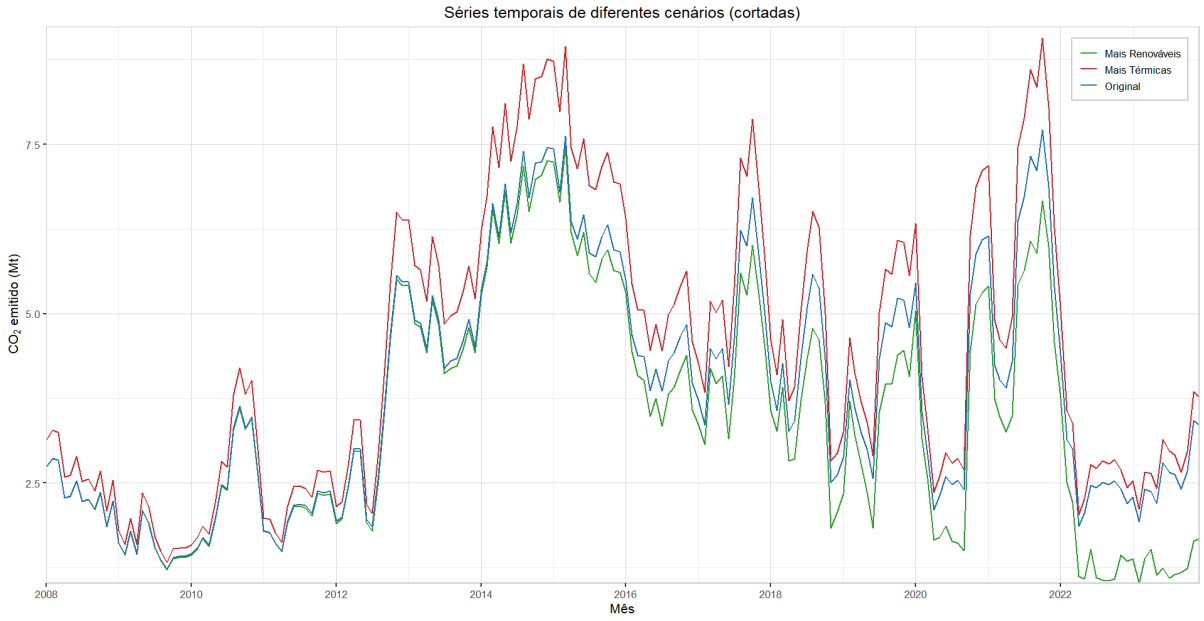
```
) +
# tira “folga” extra no eixo Y, como nos outros gráficos
scale_y_continuous(expand = expansion(mult = c(0, 0.02))) +
# legenda no canto superior direito, dentro do gráfico, estilo Python
theme(
  legend.position      = c(0.99, 0.98),
  legend.justification = c(1, 1),
  legend.background    = element_rect(fill = "white", colour = "grey80"),
  legend.key.width     = unit(1.5, "lines")
)
```



4) Corte

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)
library(tsibble)
library(grid)   # para unit()

# --- manter apenas TREINO (até 2023 Dez) ---
train_only <- dataset_st %>%
  filter(mes <= yearmonth("2023 Dec"))

# --- aplicar cortes ANTES de plotar ---
train_only_cut <- train_only %>%
  mutate(
    soma1  = if_else(mes < yearmonth("2008 Jan"), NA_real_, soma1),
    soma2r = if_else(mes < yearmonth("2008 Jan"), NA_real_, soma2r),
```

```
    soma3t = if_else(mes < yearmonth("2008 Jan"), NA_real_, soma3t)
  )

# --- GRÁFICO (somente treinamento) ---
train_only_cut %>%
  filter(mes >= yearmonth("2005 Jan")) %>%   # começa no ano desejado
  select(mes, soma1, soma2r, soma3t) %>%
  pivot_longer(-mes, names_to = "grupo", values_to = "valor") %>%
  mutate(
    grupo = recode(
      grupo,
      soma1  = "Original",
      soma2r = "Mais Renováveis",
      soma3t = "Mais Térmicas"
    ),
    date = as.Date(mes)
  ) %>%
  ggplot(aes(x = date, y = valor, colour = grupo)) +
  geom_line(linewidth = 0.9, na.rm = TRUE) +
  scale_colour_manual(values = c(
    "Original"        = tab10[1],  # azul
    "Mais Renováveis" = tab10[3],  # verde
    "Mais Térmicas"   = tab10[4]   # vermelho
  )) +
  labs(
    title = "Séries temporais de diferentes cenários (cortadas)",
    x = "Mês",
    y = expression("CO"[2] * " emitido (Mt)")
  ) +
  scale_x_date(
    limits = c(
      as.Date(yearmonth("2008 Jan")),
      max(as.Date(train_only_cut$mes))
    ),
    expand = c(0, 0),
    date_breaks = "2 years",
    date_labels = "%Y"
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.02))) +
  theme(
    legend.position      = c(0.99, 0.98),
    legend.justification = c(1, 1),
    legend.background    = element_rect(fill = "white", colour = "grey80"),
    legend.key.width     = unit(1.5, "lines"),
    legend.title         = element_blank()
  )
```



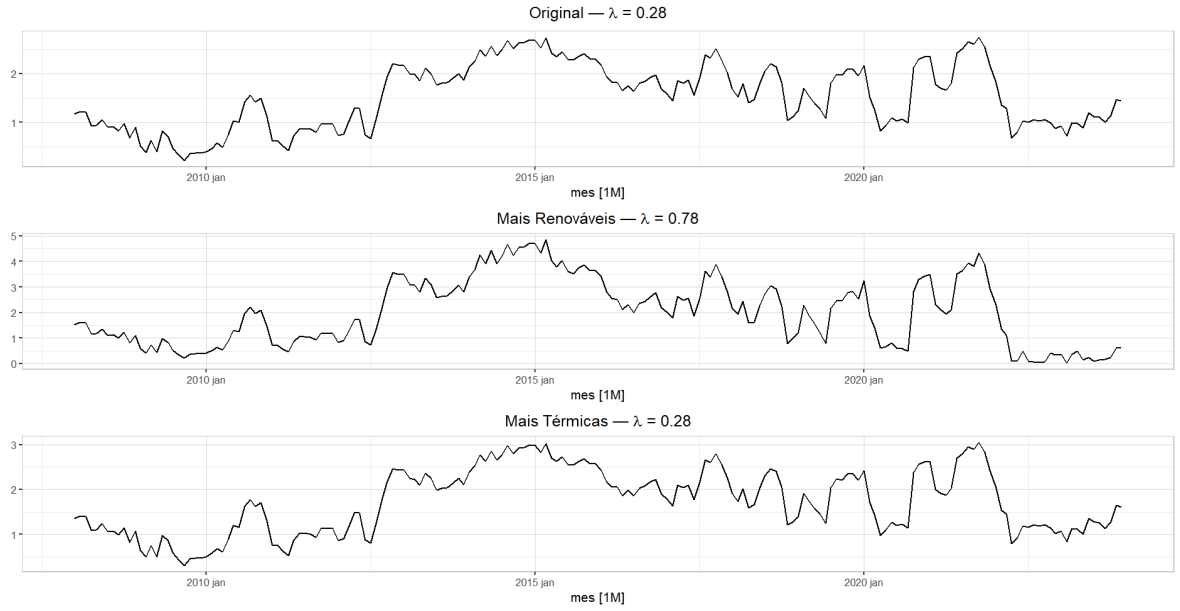
5) Separação dos dados em conjuntos de treinamento e teste:

```
# --- Treino: 2008–2023 ---
soma1_treino_2008_2023 <- dataset_st |>
  select(mes, soma1) |>
  filter_index("2008 Jan" ~ "2023 Dec")

# --- Teste: 2024–2025 ---
soma1_teste_2024_25 <- dataset_st |>
  select(mes, soma1) |>
  filter_index("2024 Jan" ~ "2025 Oct")

soma2r_treino_2008_2023 <- dataset_st |>
  select(mes, soma2r) |>
  filter_index("2008 Jan" ~ "2023 Dec")

soma2r_teste_2024_25 <- dataset_st |>
  select(mes, soma2r) |>
  filter_index("2024 Jan" ~ "2025 Oct")

soma3t_treino_2008_2023 <- dataset_st |>
  select(mes, soma3t) |>
  filter_index("2008 Jan" ~ "2023 Dec")

soma3t_treinno_2024_25 <- dataset_st |>
  select(mes, soma3t) |>
  filter_index("2024 Jan" ~ "2025 Oct")
```

6) Transformação Box-Cox

```
library(ggplot2)
library(fabletools)
library(feasts)
library(latex2exp)
library(patchwork)

# aqui assumo que você já tem:
# tab10 <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
#            "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
#            "#bcbd22", "#17becf")
# theme_python() definido e, se quiser, theme_set(theme_python())

# -------------------------
# Original
# -------------------------
lambda_soma1 <- soma1_treino_2008_2023 |>
  features(soma1, features = guerrero) |>
  pull(lambda_guerrero)

p_soma1 <- soma1_treino_2008_2023 |>
  autoplot(box_cox(soma1, lambda_soma1)) +
  labs(
    y = "",
    title = latex2exp::TeX(paste0(
      "Original — $\\lambda$ = ", round(lambda_soma1, 2)))
  ) +
  theme_python(14)

# -------------------------
# Mais Renováveis
# -------------------------
lambda_soma2r <- soma2r_treino_2008_2023 |>
  features(soma2r, features = guerrero) |>
  pull(lambda_guerrero)

p_soma2r <- soma2r_treino_2008_2023 |>
  autoplot(box_cox(soma2r, lambda_soma2r)) +
  labs(
    y = "",
    title = latex2exp::TeX(paste0(
      "Mais Renováveis — $\\lambda$ = ", round(lambda_soma2r, 2)))
  ) +
  theme_python(14)

# -------------------------
# Mais Térmicas
# -------------------------
```

```
lambda_soma3t <- soma3t_treino_2008_2023 |>
  features(soma3t, features = guerrero) |>
  pull(lambda_guerrero)

p_soma3t <- soma3t_treino_2008_2023 |>
  autoplot(box_cox(soma3t, lambda_soma3t)) +
  labs(
    y = "",
    title = latex2exp::TeX(paste0(
      "Mais Térmicas — $\\lambda$ = ", round(lambda_soma3t, 2)))
  ) +
  theme_python(14)

# -------------------------
# 3 gráficos empilhados (1 por linha)
# -------------------------
p_soma1 /
p_soma2r /
p_soma3t
```



7) Ajuste do modelo ARIMA AUTOMÁTICO com Box–Cox para cada grupo

```
# ORIGINAL
soma1_bc <- soma1_treino_2008_2023 %>%
  mutate(co2_bc = box_cox(soma1, lambda_soma1))

# MAIS RENOVAVEIS
soma2r_bc <- soma2r_treino_2008_2023 %>%
  mutate(co2_bc = box_cox(soma2r, lambda_soma2r))
```

```
# MAIS TERMICAS
soma3t_bc <-soma3t_treino_2008_2023 %>%
  mutate(co2_bc = box_cox(soma3t, lambda_soma3t))

library(fable)
library(fabletools)

# ORIGINAL
fit_soma1_arima <- soma1_bc %>%
  model(
    arima = ARIMA(co2_bc, stepwise = FALSE, approx = FALSE)
  )

# MAIS RENOVAVEIS
fit_soma2r_arima <- soma2r_bc %>%
  model(
    arima = ARIMA(co2_bc, stepwise = FALSE, approx = FALSE)
  )

# MAIS TERMICAS
fit_soma3t_arima <- soma3t_bc %>%
  model(
    arima = ARIMA(co2_bc, stepwise = FALSE, approx = FALSE)
  )


report(fit_soma1_arima)
report(fit_soma2r_arima)
report(fit_soma3t_arima)
```

```
> report(fit_soma1_arima)
Series: co2_bc
Model: ARIMA(0,1,4)(2,0,0)[12]

Coefficients:
         ma1     ma2      ma3      ma4    sar1    sar2
      0.1101  0.0181  -0.1025  -0.3527  0.1027  0.1159
s.e.  0.0708  0.0748   0.0766   0.0864  0.0735  0.0757

sigma^2 estimated as 0.05228:  log likelihood=13.36
AIC=-12.73   AICc=-12.12   BIC=10.04
> report(fit_soma2r_arima)
Series: co2_bc
Model: ARIMA(0,1,4)(1,0,0)[12]

Coefficients:
         ma1     ma2      ma3      ma4    sar1
      0.0883  0.0448  -0.0684  -0.2986  0.1150
```

```
s.e.  0.0716  0.0755   0.0780   0.0851  0.0729

sigma^2 estimated as 0.2054:  log likelihood=-117.6
AIC=247.2   AICc=247.65   BIC=266.71
> report(fit_soma3t_arima)
Series: co2_bc
Model: ARIMA(0,1,4)(2,0,0)[12]

Coefficients:
         ma1     ma2      ma3      ma4    sar1    sar2
      0.1143  0.0126  -0.1033  -0.3536  0.0985  0.1151
s.e.  0.0708  0.0747   0.0766   0.0864  0.0736  0.0758

sigma^2 estimated as 0.0623:  log likelihood=-3.38
AIC=20.76   AICc=21.37   BIC=43.52
```

8) BOOTSTRAP - ARIMA AUTOMÁTICO

```
sims_soma1 <- generate(
  fit_soma1_arima,
  h = 22,
  times = 500,
  bootstrap = TRUE   # garante variabilidade realista
)

sims_soma2r <- generate(
  fit_soma2r_arima,
  h = 22,
  times = 500,
  bootstrap = TRUE
)


sims_soma3t <- generate(
  fit_soma3t_arima,
  h = 22,
  times = 500,
  bootstrap = TRUE
)
```

9) INVERTENDO BOX-COX - ARIMA AUTOMÁTICO

```
inv_boxcox <- function(x, lambda){
  if(lambda == 0) return(exp(x))
  (lambda * x + 1)^(1/lambda)
}
```

```
sims_soma1_real <- sims_soma1 %>%
  mutate(value = inv_boxcox(.sim, lambda_soma1))


sims_soma2r_real <- sims_soma2r %>%
  mutate(value = inv_boxcox(.sim, lambda_soma2r))

sims_soma3t_real <- sims_soma3t %>%
  mutate(value = inv_boxcox(.sim, lambda_soma3t))
```

10) Ajuste do ARIMA (2,0,0)(0,1,1) - TRAVADO

```
library(fable)
library(fabletools)
library(dplyr)

# ------------------------------------------------
# 1) MODELOS ARIMA TRAVADOS EM CADA CENÁRIO
#    (usando a série JÁ transformada: co2_bc)
# ------------------------------------------------

# ORIGINAL  -> ARIMA(2,0,0)(0,1,1)[12]
fit_soma1_arima_travado <- soma1_bc %>%
  model(
    arima = ARIMA(co2_bc ~ 0 + pdq(2, 0, 0) + PDQ(0, 1, 1))
  )

# MAIS RENOVÁVEIS -> ARIMA(2,0,0)(0,1,1)[12]
fit_soma2r_arima_travado <- soma2r_bc %>%
  model(
    arima = ARIMA(co2_bc ~ 0 + pdq(2, 0, 0) + PDQ(0, 1, 1))
  )

# MAIS TÉRMICAS -> ARIMA(2,0,0)(0,1,1)[12]
fit_soma3t_arima_travado <- soma3t_bc %>%
  model(
    arima = ARIMA(co2_bc ~ 0 + pdq(2, 0, 0) + PDQ(0, 1, 1))
  )

# (opcional) ver o resumo
report(fit_soma1_arima_travado)
report(fit_soma2r_arima_travado)
report(fit_soma3t_arima_travado)
```

```
> report(fit_soma1_arima_travado)
Series: co2_bc
```

```
Model: ARIMA(2,0,0)(0,1,1)[12]

Coefficients:
         ar1      ar2     sma1
      1.1105  -0.1818  -0.8407
s.e.  0.0728   0.0733   0.0718

sigma^2 estimated as 0.05389:  log likelihood=0.85
AIC=6.3   AICc=6.53   BIC=19.07
> report(fit_soma2r_arima_travado)
Series: co2_bc
Model: ARIMA(2,0,0)(0,1,1)[12]

Coefficients:
         ar1      ar2     sma1
      1.0937  -0.1571  -0.8995
s.e.  0.0732   0.0738   0.0931

sigma^2 estimated as 0.2108:  log likelihood=-124.23
AIC=256.46   AICc=256.68   BIC=269.23
> report(fit_soma3t_arima_travado)
Series: co2_bc
Model: ARIMA(2,0,0)(0,1,1)[12]

Coefficients:
         ar1      ar2     sma1
      1.1103  -0.1822  -0.8389
s.e.  0.0728   0.0733   0.0718

sigma^2 estimated as 0.0645:  log likelihood=-15.26
AIC=38.53   AICc=38.76   BIC=51.3
```

11) BOOTSTRAP DO ARIMA (2,0,0)(0,1,1)

```
# ------------------------------------------------
# 2) GERAR 500 TRAJETÓRIAS FUTURAS (BOOTSTRAP)
# ------------------------------------------------

h <- 22  # horizonte

sims_soma1_trav <- generate(
  fit_soma1_arima_travado,
  h = h,
  times = 500,
  bootstrap = TRUE
)

sims_soma2r_trav <- generate(
  fit_soma2r_arima_travado,
```

```
  h = h,
  times = 500,
  bootstrap = TRUE
)

sims_soma3t_trav <- generate(
  fit_soma3t_arima_travado,
  h = h,
  times = 500,
  bootstrap = TRUE
)
```

12) INVERTENDO BOX-COX - ARIMA (2,0,0)(0,1,1)

```
# ------------------------------------------------
# 3) VOLTAR DA ESCALA BOX–COX PARA A ESCALA ORIGINAL
# ------------------------------------------------

inv_boxcox <- function(x, lambda){
  if (lambda == 0) return(exp(x))
  (lambda * x + 1)^(1 / lambda)
}

sims_soma1_trav_real <- sims_soma1_trav %>%
  mutate(value = inv_boxcox(.sim, lambda_soma1))

sims_soma2r_trav_real <- sims_soma2r_trav %>%
  mutate(value = inv_boxcox(.sim, lambda_soma2r))

sims_soma3t_trav_real <- sims_soma3t_trav %>%
  mutate(value = inv_boxcox(.sim, lambda_soma3t))
```

13) MÉTRICAS

```
library(dplyr)
library(fable)
library(fabletools)
library(tibble)

# ------------------------
# 1) Funções auxiliares
# ------------------------

inv_boxcox <- function(x, lambda){
  if (lambda == 0) return(exp(x))
  (lambda * x + 1)^(1 / lambda)
}
```

```
RMSE <- function(real, prev) sqrt(mean((real - prev)^2, na.rm = TRUE))
MAE  <- function(real, prev) mean(abs(real - prev), na.rm = TRUE)
MAPE <- function(real, prev) mean(abs((real - prev)/real), na.rm = TRUE) * 100

calc_metricas <- function(real, prev, cenario, modelo){
  tibble(
    Cenario = cenario,
    Modelo  = modelo,
    RMSE    = RMSE(real, prev),
    MAE     = MAE(real, prev),
    MAPE    = MAPE(real, prev)
  )
}

# ------------------------
# 2) PREVISÕES: ARIMA AUTOMÁTICO
# ------------------------

# Original
fc_soma1_auto <- forecast(
  fit_soma1_arima,
  new_data = soma1_teste_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma1))

# Mais Renováveis
fc_soma2r_auto <- forecast(
  fit_soma2r_arima,
  new_data = soma2r_teste_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma2r))

# Mais Térmicas
fc_soma3t_auto <- forecast(
  fit_soma3t_arima,
  new_data = soma3t_treinno_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma3t))



# ------------------------
# 3) PREVISÕES: ARIMA TRAVADO 200011
# ------------------------
```

```
# Original
fc_soma1_trav <- forecast(
  fit_soma1_arima_travado,
  new_data = soma1_teste_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma1))

# Mais Renováveis
fc_soma2r_trav <- forecast(
  fit_soma2r_arima_travado,
  new_data = soma2r_teste_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma2r))

# Mais Térmicas
fc_soma3t_trav <- forecast(
  fit_soma3t_arima_travado,
  new_data = soma3t_treinno_2024_25,
  bias_adjust = TRUE
) %>%
  mutate(.mean = inv_boxcox(.mean, lambda_soma3t))



# ------------------------
# 4) MÉTRICAS PARA CADA COMBINAÇÃO
# ------------------------

# Original
met_orig_auto <- calc_metricas(
  real   = soma1_teste_2024_25$soma1,
  prev   = fc_soma1_auto$.mean,
  cenario = "Original",
  modelo  = "ARIMA auto"
)

met_orig_trav <- calc_metricas(
  real   = soma1_teste_2024_25$soma1,
  prev   = fc_soma1_trav$.mean,
  cenario = "Original",
  modelo  = "ARIMA(2,0,0)(0,1,1)"
)

# Mais Renováveis
met_ren_auto <- calc_metricas(
  real   = soma2r_teste_2024_25$soma2r,
  prev   = fc_soma2r_auto$.mean,
```

```
  cenario = "Mais Renováveis",
  modelo  = "ARIMA auto"
)

met_ren_trav <- calc_metricas(
  real   = soma2r_teste_2024_25$soma2r,
  prev   = fc_soma2r_trav$.mean,
  cenario = "Mais Renováveis",
  modelo  = "ARIMA(2,0,0)(0,1,1)"
)

# Mais Térmicas
met_term_auto <- calc_metricas(
  real   = soma3t_treinno_2024_25$soma3t,
  prev   = fc_soma3t_auto$.mean,
  cenario = "Mais Térmicas",
  modelo  = "ARIMA auto"
)

met_term_trav <- calc_metricas(
  real   = soma3t_treinno_2024_25$soma3t,
  prev   = fc_soma3t_trav$.mean,
  cenario = "Mais Térmicas",
  modelo  = "ARIMA(2,0,0)(0,1,1)"
)

# ------------------------
# 5) TABELA FINAL DE MÉTRICAS
# ------------------------

metricas_modelos <- bind_rows(
  met_orig_auto,
  met_orig_trav,
  met_ren_auto,
  met_ren_trav,
  met_term_auto,
  met_term_trav
)

metricas_modelos
```

```
> metricas_modelos
# A tibble: 6 × 5
  Cenario         Modelo                RMSE   MAE  MAPE
  <chr>           <chr>                <dbl> <dbl> <dbl>
1 Original        ARIMA auto           1.07  0.737  18.4
2 Original        ARIMA(2,0,0)(0,1,1)  0.580 0.447  13.1
3 Mais Renováveis ARIMA auto           0.528 0.381  21.1
```

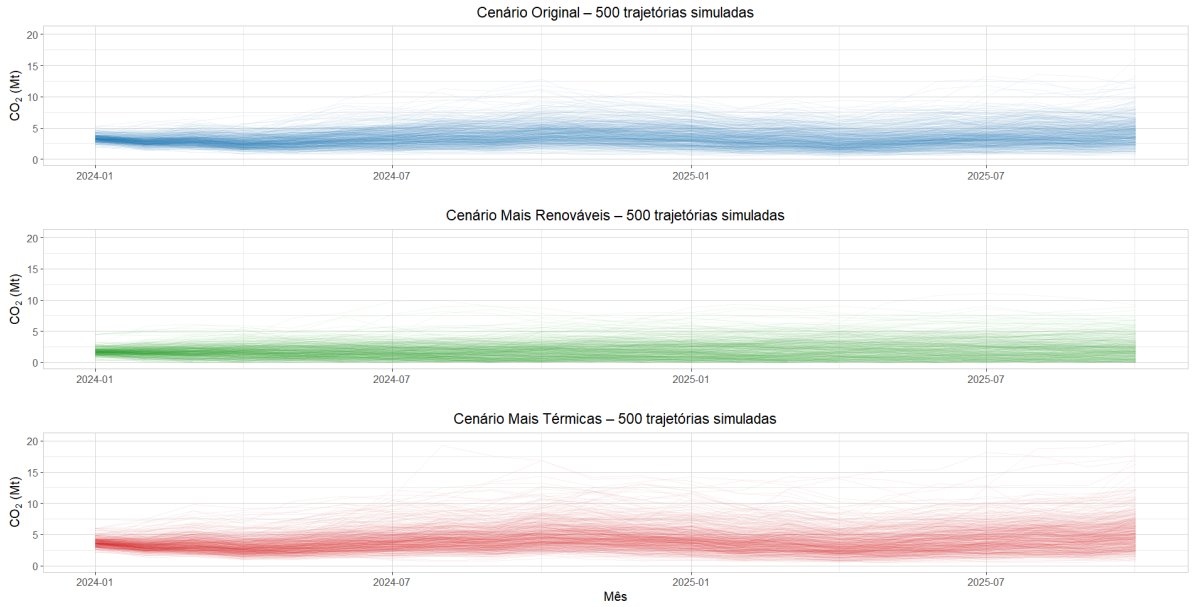
```
4 Mais Renováveis ARIMA(2,0,0)(0,1,1) 0.762 0.660  44.8
5 Mais Térmicas   ARIMA auto           1.28  0.884  19.6
6 Mais Térmicas   ARIMA(2,0,0)(0,1,1) 0.697 0.548  14.6
```

14) GRÁFICO DE TRAJETÓRIAS FUTURAS - MELHORES ARIMA

```
library(dplyr)
library(ggplot2)
library(patchwork)

# ================================================
# Paleta tab10 (Matplotlib)
# ================================================
tab10 <- c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
  "#bcbd22", "#17becf"
)

# ================================================
# Tema estilo Python / Matplotlib
# ================================================
theme_python <- function(base_size = 14) {
  theme_bw(base_size = base_size) +
    theme(
      panel.background = element_rect(fill = "white"),
      panel.grid.major = element_line(colour = "grey85", linewidth = 0.4),
      panel.grid.minor = element_line(colour = "grey92", linewidth = 0.2),
      panel.border     = element_rect(colour = "grey80", fill = NA),
      plot.title       = element_text(hjust = 0.5),
      legend.position  = "none"
    )
}

# ================================================
# Transformar índices para Date
# ================================================
soma1_hist <- soma1_treino_2008_2023 %>% mutate(date = as.Date(mes))
soma2r_hist <- soma2r_treino_2008_2023 %>% mutate(date = as.Date(mes))
soma3t_hist <- soma3t_treino_2008_2023 %>% mutate(date = as.Date(mes))

sims_soma1_plot <- sims_soma1_trav_real %>% mutate(date = as.Date(mes))
sims_soma2r_plot <- sims_soma2r_real %>% mutate(date = as.Date(mes))
sims_soma3t_plot <- sims_soma3t_trav_real %>% mutate(date = as.Date(mes))

# ================================================
# Limites comuns Y
```

```
# ================================================
y_min <- min(
  sims_soma1_plot$value,
  sims_soma2r_plot$value,
  sims_soma3t_plot$value,
  na.rm = TRUE
)

y_max <- max(
  sims_soma1_plot$value,
  sims_soma2r_plot$value,
  sims_soma3t_plot$value,
  na.rm = TRUE
)

# ================================================
# GRÁFICOS
# ================================================

# --- ORIGINAL ---
g1 <- ggplot(sims_soma1_plot, aes(x = date, y = value, group = .rep)) +
  geom_line(alpha = 0.05, colour = tab10[1]) +   # azul matplotlib
  coord_cartesian(ylim = c(y_min, y_max)) +
  labs(
    title = "Cenário Original – 500 trajetórias simuladas",
    x = "",
    y = expression("CO"[2] * " (Mt)")
  ) +
  theme_python(13)

# --- MAIS RENOVÁVEIS ---
g2 <- ggplot(sims_soma2r_plot, aes(x = date, y = value, group = .rep)) +
  geom_line(alpha = 0.05, colour = tab10[3]) +   # verde matplotlib
  coord_cartesian(ylim = c(y_min, y_max)) +
  labs(
    title = "Cenário Mais Renováveis – 500 trajetórias simuladas",
    x = "",
    y = expression("CO"[2] * " (Mt)")
  ) +
  theme_python(13)

# --- MAIS TÉRMICAS ---
g3 <- ggplot(sims_soma3t_plot, aes(x = date, y = value, group = .rep)) +
  geom_line(alpha = 0.05, colour = tab10[4]) +   # vermelho matplotlib
  coord_cartesian(ylim = c(y_min, y_max)) +
  labs(
    title = "Cenário Mais Térmicas – 500 trajetórias simuladas",
    x = "Mês",
```

```
    y = expression("CO"[2] * " (Mt)")
  ) +
  theme_python(13)

# ================================================
# SAÍDA FINAL (3 gráficos empilhados)
# ================================================
g1 / g2 / g3
```



15) ALGUMAS ESTATÍSTICAS DESCRITIVAS

```
library(dplyr)

# ----------------------------------------
# Função simples para calcular estatísticas
# ----------------------------------------
resumo_simples <- function(sim_tbl, nome){
  sim_tbl %>%
    as_tibble() %>%                    # garante tibble
    mutate(date = as.Date(mes)) %>%
    filter(date >= as.Date("2024-01-01")) %>%   # apenas futuro
    group_by(date) %>%
    summarise(
      q05     = quantile(value, 0.05, na.rm = TRUE),
      mediana = quantile(value, 0.50, na.rm = TRUE),
      q95     = quantile(value, 0.95, na.rm = TRUE),
      media   = mean(value, na.rm = TRUE),
      .groups = "drop"
    ) %>%
```

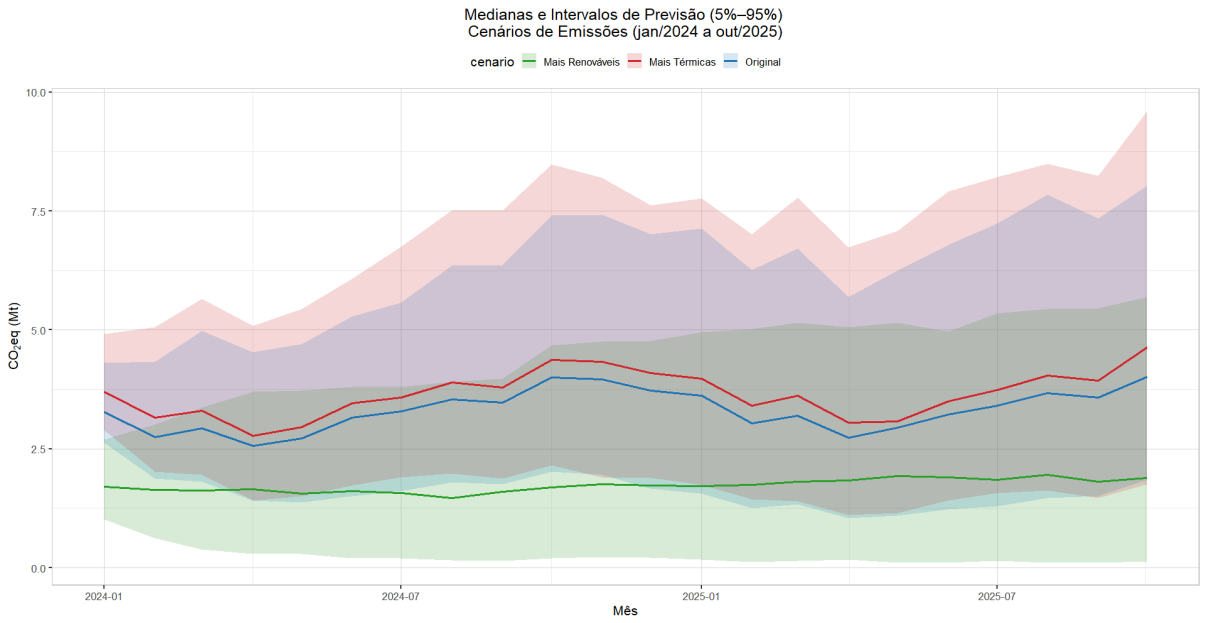
```
    summarise(
      media_2024_25   = mean(media),
      mediana_2024_25 = mean(mediana),
      IC5_2024_25     = mean(q05),
      IC95_2024_25    = mean(q95),
      .groups = "drop"
    ) %>%
    mutate(Cenario = nome)
}

# ----------------------------------------
# Aplicar aos 3 modelos
# ----------------------------------------

tab_orig <- resumo_simples(sims_soma1_trav_real,  "Original")
tab_ren  <- resumo_simples(sims_soma2r_real,      "Mais Renováveis")
tab_term <- resumo_simples(sims_soma3t_trav_real, "Mais Térmicas")

# ----------------------------------------
# Tabela final com os 3 modelos
# ----------------------------------------

tabela_resumo <- bind_rows(tab_orig, tab_ren, tab_term)

tabela_resumo
```

```
> tabela_resumo
# A tibble: 3 × 5
  media_2024_25 mediana_2024_25 IC5_2024_25 IC95_2024_25 Cenario
          <dbl>           <dbl>       <dbl>        <dbl> <chr>
1          3.57            3.31       1.60          6.33 Original
2          1.96            1.73       0.238         4.47 Mais Renováveis
3          4.19            3.85       1.83          7.76 Mais Térmicas
```

16) Gráficos comparando as medianas

```
library(dplyr)
library(ggplot2)

# Função: extrair bandas (IC 5% e 95% + mediana)
extrair_bandas <- function(sim_tbl, nome_cenario) {
  sim_tbl %>%
    as_tibble() %>%
    mutate(date = as.Date(mes)) %>%
    filter(date >= as.Date("2024-01-01")) %>%
    group_by(date) %>%
    summarise(
```

```
      q05     = quantile(value, 0.05, na.rm = TRUE),
      mediana = quantile(value, 0.50, na.rm = TRUE),
      q95     = quantile(value, 0.95, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    mutate(cenario = nome_cenario)
}

# Bandas para cada cenário
band1  <- extrair_bandas(sims_soma1_trav_real,  "Original")
band2r <- extrair_bandas(sims_soma2r_real,      "Mais Renováveis")
band3t <- extrair_bandas(sims_soma3t_trav_real, "Mais Térmicas")

band_all <- bind_rows(band1, band2r, band3t)

# Gráfico com mediana + intervalo de confiança 5–95% (estilo Python)
ggplot(band_all, aes(x = date, colour = cenario, fill = cenario)) +
  geom_ribbon(aes(ymin = q05, ymax = q95),
              alpha = 0.18, colour = NA) +
  geom_line(aes(y = mediana), linewidth = 1.2) +
  scale_colour_manual(values = c(
    "Original"        = tab10[1],  # azul
    "Mais Renováveis" = tab10[3],  # verde
    "Mais Térmicas"   = tab10[4]   # vermelho
  )) +
  scale_fill_manual(values = c(
    "Original"        = tab10[1],
    "Mais Renováveis" = tab10[3],
    "Mais Térmicas"   = tab10[4]
  )) +
  labs(
    title = "Medianas e Intervalos de Previsão (5%–95%) \nCenários de Emissões (jan/2024 a
out/2025)",
    x = "Mês",
    y = expression("CO"[2] * "eq (Mt)")
  ) +
  theme_python(14) +
  theme(
    legend.position      = "top",
    legend.justification = "center"
  )
```



17) Métricas RMSE / MAE / MAPE usando a Mediana e Média do bootstrap

```
library(dplyr)
library(tsibble)

# Resumo do bootstrap por mês (cenário Original)
res_boot_orig <- sims_soma1_trav_real %>%
  index_by(mes) %>%   # mes é o índice do tsibble
  summarise(
    sim_median = median(value, na.rm = TRUE),
    sim_mean   = mean(value,   na.rm = TRUE),
    sim_q05    = quantile(value, 0.05, na.rm = TRUE),
    sim_q95    = quantile(value, 0.95, na.rm = TRUE)
  ) %>%
  left_join(
    soma1_teste_2024_25 %>%
      as_tibble() %>%
      select(mes, obs = soma1),
    by = "mes"
  )


res_boot_orig
```

```
> res_boot_orig
# A tsibble: 22 x 6 [1M]
        mes sim_median sim_mean sim_q05 sim_q95   obs
      <mth>      <dbl>    <dbl>   <dbl>   <dbl> <dbl>
 1 2024 jan       3.28     3.34    2.62    4.29  3.17
 2 2024 fev       2.83     2.90    1.86    4.12  2.82
 3 2024 mar       2.99     3.15    1.76    4.95  2.69
 4 2024 abr       2.60     2.77    1.42    4.64  2.17
 5 2024 mai       2.76     2.94    1.44    4.89  2.26
 6 2024 jun       3.10     3.31    1.54    5.51  2.35
 7 2024 jul       3.26     3.49    1.65    5.98  3.05
 8 2024 ago       3.55     3.79    1.80    6.74  3.68
 9 2024 set       3.47     3.75    1.73    6.71  4.58
10 2024 out       3.93     4.25    2.04    7.60  5.50
# i 12 more rows
# i Use `print(n = ...)` to see more rows
```

```
rmse <- function(y, yhat) sqrt(mean((y - yhat)^2, na.rm = TRUE))
mae  <- function(y, yhat) mean(abs(y - yhat), na.rm = TRUE)
mape <- function(y, yhat) mean(abs((y - yhat) / y), na.rm = TRUE) * 100

metricas_boot_median <- tibble::tibble(
  modelo = "bootstrap_mediana_cenario_original",
  RMSE   = rmse(res_boot_orig$obs, res_boot_orig$sim_median),
  MAE    = mae(res_boot_orig$obs, res_boot_orig$sim_median),
  MAPE   = mape(res_boot_orig$obs, res_boot_orig$sim_median)
)

metricas_boot_mean <- tibble::tibble(
  modelo = "bootstrap_media_cenario_original",
  RMSE   = rmse(res_boot_orig$obs, res_boot_orig$sim_mean),
  MAE    = mae(res_boot_orig$obs, res_boot_orig$sim_mean),
  MAPE   = mape(res_boot_orig$obs, res_boot_orig$sim_mean)
)


metricas_bootstrap <- dplyr::bind_rows(
  metricas_boot_median,
  metricas_boot_mean
)

metricas_bootstrap
```

```
> metricas_bootstrap
# A tibble: 2 x 4
  modelo                              RMSE   MAE  MAPE
  <chr>                              <dbl> <dbl> <dbl>
1 bootstrap_mediana_cenario_original 0.571 0.418  12.1
2 bootstrap_media_cenario_original   0.590 0.504  16.2
>
```

##A MEDIANA do bootstrap performa MELHOR que o ARIMA travado

18) Coverage + largura do intervalo (qualidade da distribuição)

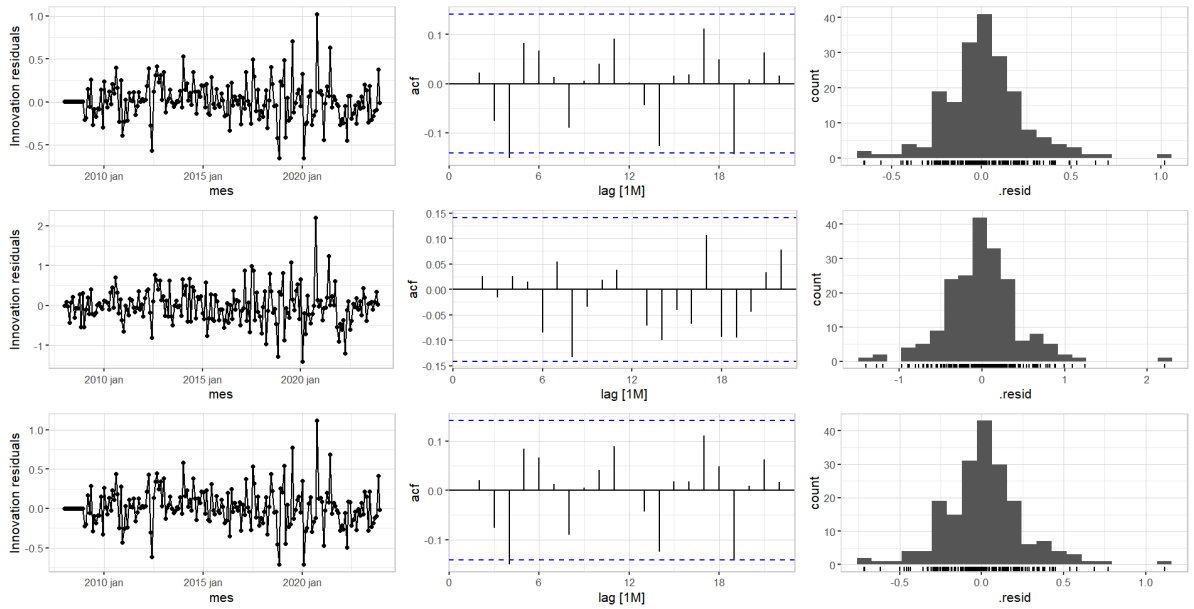
```
coverage_5_95 <- mean(
  res_boot_orig$obs >= res_boot_orig$sim_q05 &
  res_boot_orig$obs <= res_boot_orig$sim_q95,
  na.rm = TRUE
)

largura_media_IC <- mean(
  res_boot_orig$sim_q95 - res_boot_orig$sim_q05,
  na.rm = TRUE
)

qualidade_distribuicao <- tibble::tibble(
  cobertura_5_95   = coverage_5_95,
  largura_media_IC = largura_media_IC
)

qualidade_distribuicao
```

```
> qualidade_distribuicao
# A tibble: 1 x 2
  cobertura_5_95 largura_media_IC
           <dbl>            <dbl>
1              1             4.70
>
```

19) Diagnóstico dos Resíduos e teste Ljung-Box:

```
library(fable)
library(fabletools)
library(feasts)
library(ggplot2)
library(patchwork)

plot_residuos_modelo <- function(fit, titulo){
  wrap_plots(
    fit %>% gg_tsresiduals()
  ) + plot_annotation(title = titulo)
}

# Original – ARIMA(2,0,0)(0,1,1)
g1 <- plot_residuos_modelo(
  fit_soma1_arima_travado,
  "Original – ARIMA(2,0,0)(0,1,1)"
)

# Mais Renováveis – ARIMA(0,1,4)(1,0,0)
g2 <- plot_residuos_modelo(
  fit_soma2r_arima,
```

```
  "Mais Renováveis – ARIMA(0,1,4)(1,0,0)"
)

# Mais Térmicas – ARIMA(2,0,0)(0,1,1)
g3 <- plot_residuos_modelo(
  fit_soma3t_arima_travado,
  "Mais Térmicas – ARIMA(2,0,0)(0,1,1)"
)

# ---- Unir tudo em 3 linhas ----
grafico_final <- g1 / g2 / g3

grafico_final
```




```
library(fable)
library(fabletools)
library(feasts)

# ---- ORIGINAL – ARIMA(2,0,0)(0,1,1) ----
# p = 2, q = 0, P = 0, Q = 1  → dof = 3
lb1 <- fit_soma1_arima_travado %>%
  augment() %>%
  features(.innov, ljung_box, lag = 24, dof = 3)

# ---- MAIS RENOVÁVEIS – ARIMA(0,1,4)(1,0,0) ----
# p = 0, q = 4, P = 1, Q = 0  → dof = 5
lb2 <- fit_soma2r_arima %>%
  augment() %>%
  features(.innov, ljung_box, lag = 24, dof = 5)
```

```
# ---- MAIS TÉRMICAS – ARIMA(2,0,0)(0,1,1) ----
# p = 2, q = 0, P = 0, Q = 1  → dof = 3
lb3 <- fit_soma3t_arima_travado %>%
  augment() %>%
  features(.innov, ljung_box, lag = 24, dof = 3)

lb1
lb2
lb3
```

```
> lb1
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     26.4     0.193
> lb2
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     19.7     0.414
> lb3
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     26.0     0.208
```