



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
GRADUAÇÃO EM SISTEMAS DA INFORMAÇÃO
TÉCNICAS ESTATÍSTICAS DE PREDIÇÃO**

**GUILHERME FELIPE DAROSSO
RAQUEL ALVES PINTO
THIAGO COUTINHO DA SILVA**

**ANÁLISE TEMPORAL DA SAÚDE RESPIRATÓRIA E QUALIDADE DO AR EM
GRANDES METRÓPOLES**

**Florianópolis
2025**

1. INTRODUÇÃO E OBJETIVOS

A poluição atmosférica nos grandes centros urbanos tem se intensificado por diversos fatores nas últimas décadas. A exposição contínua a poluentes como o Material Particulado (PM_{2.5} e PM₁₀), Dióxido de Nitrogênio (NO₂) e Ozônio (O₃) está fortemente associada ao aumento de problemas respiratórios, impactando diretamente a saúde pública. Compreender essa relação ao longo do tempo é essencial para o desenvolvimento de políticas públicas, planejamento urbano e alocação adequada de recursos hospitalares.

O presente trabalho principal deste trabalho é realizar uma análise exploratória dos dados, identificando as possíveis sazonalidades, tendências ou ciclos, além de anomalias nos indicadores ambientais e de saúde. Em seguida será conduzida uma análise das séries temporais com foco na evolução dos níveis de poluição e das internações hospitalares.

Com isso, espera-se contribuir para a compreensão dos impactos da poluição atmosférica na saúde respiratória no contexto urbano do dataset, reforçando a importância do monitoramento contínuo da qualidade do ar como instrumento de prevenção e gestão em saúde pública.

2. MATERIAIS E MÉTODOS

O presente estudo utiliza o conjunto de dados *Global Air Quality and Respiratory Health Outcomes*, disponível na plataforma Kaggle, que reúne registros diários de poluentes atmosféricos e internações hospitalares em grandes centros urbanos. A análise foi centrada na cidade de Londres, com o objetivo de investigar a relação entre variáveis ambientais e o número de internações por doenças respiratórias.

Inicialmente, foi realizada uma análise exploratória das variáveis, com descrição estatística e cálculo do coeficiente de determinação (R^2) para medir a força da associação entre os poluentes (AQI, PM2.5, PM10, NO₂, O₃), temperatura, umidade e os registros de internações. Em seguida, a base foi tratada, convertendo datas para o tipo apropriado e transformando variáveis categóricas em fatores. Também foram verificados valores ausentes e identificados dias com internações muito acima da média.

Os dados filtrados para Londres foram transformados em uma série temporal diária e decompostos pelo método STL para análise das componentes sazonais e tendências. Com base nisso, foram ajustados dois modelos: um de regressão linear múltipla com variáveis ambientais e outro SARIMA, com enfoque exclusivamente temporal. Ambos foram avaliados com as métricas MAE e RMSE, comparando suas previsões com os dados reais. Essa abordagem permitiu observar o potencial preditivo de cada técnica e sua aplicabilidade na área de saúde pública.

3. DEFINIÇÃO DAS VARIÁVEIS

Variável dependente:

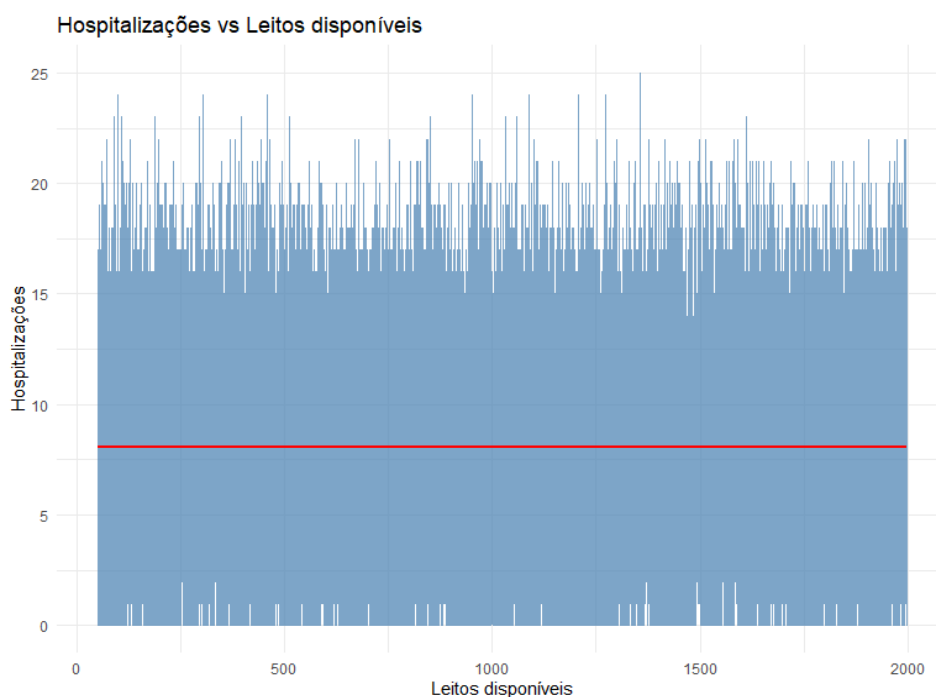
- *hospital_admissions*: Número de internações diárias em hospitais por problemas respiratórios;

Variáveis independentes:

- *population_density*: Classificação Urbana, Suburbana ou Rural;
- *hospital_capacity*: Número de leitos disponíveis nos hospitais locais;
- *humidity*: Umidade relativa média diária (%);
- *temperature*: Temperatura média diária (°C);
- *o3*: Concentração de ozônio (ppb);
- *no2*: Concentração de dióxido de nitrogênio (ppb);
- *pm10*: Concentração de PM10 ($\mu\text{g}/\text{m}^3$);
- *pm2_5*: Concentração de PM2.5 ($\mu\text{g}/\text{m}^3$);
- *aqi*: Valor do Índice de Qualidade do Ar;
- *date*: Data da observação (AAAA-MM-DD);
- *city*: Nome da cidade onde os dados foram registrados;

3.1 Apresentação das variáveis

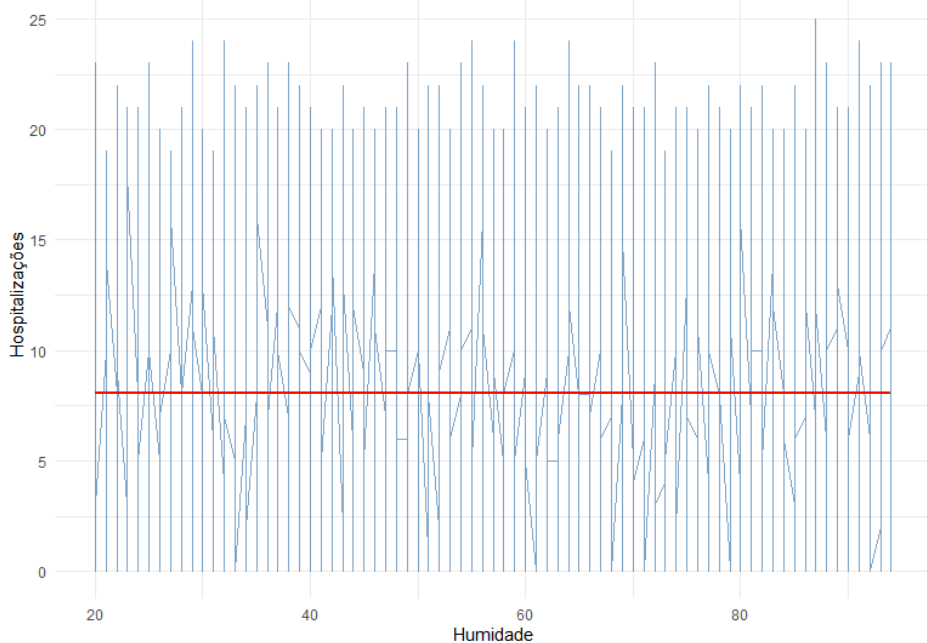
A fim de representar a importância de cada variável independente, foi realizada uma comparação individual entre elas e o número de internações diárias.



correlação: 0.001533149

R^2 : 2.350546e-06

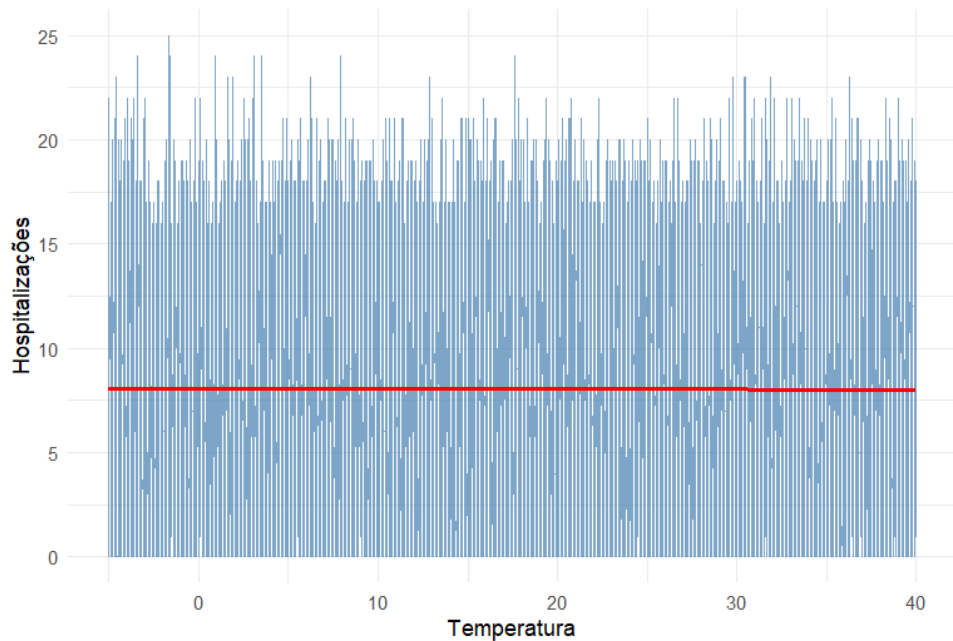
Hospitalizações vs Humidade



correlação: $9.882925e-06$

R^2 : $9.76722e-11$

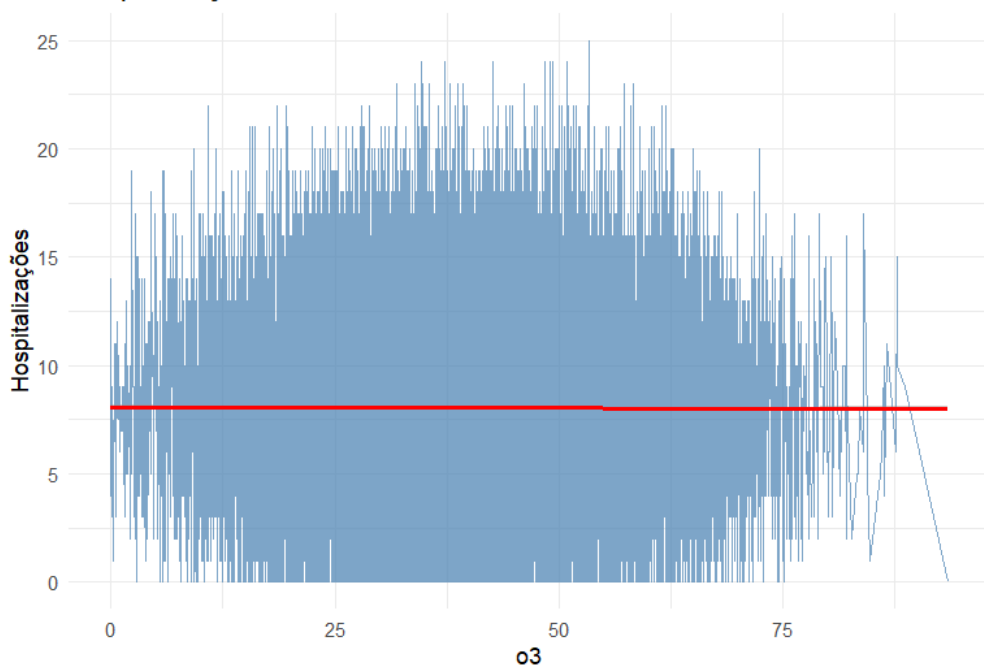
Hospitalizações vs Temperatura



Correlação: -0.003782829

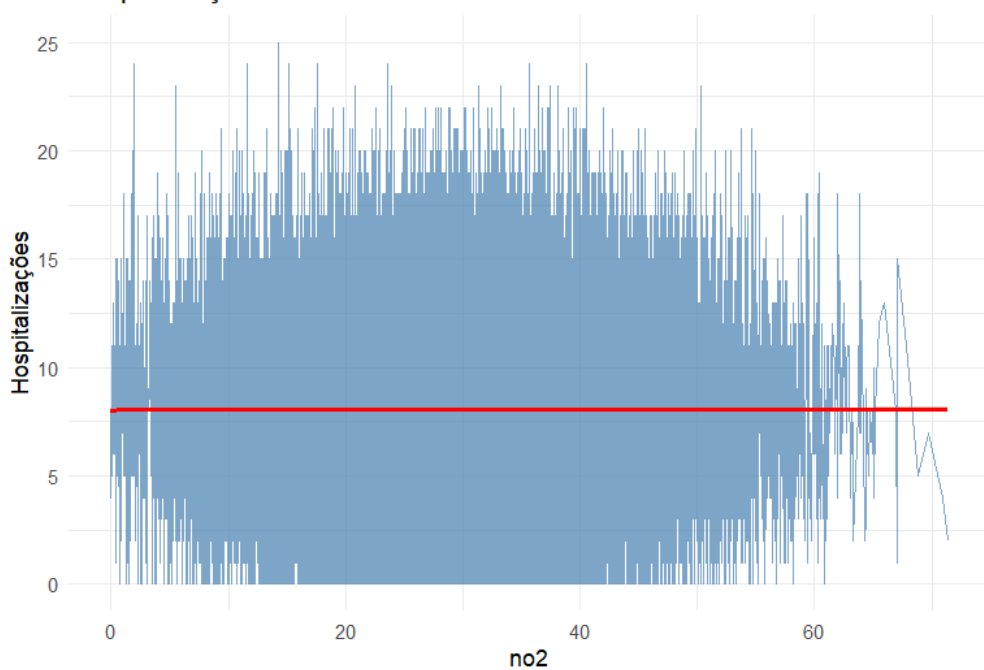
R^2 : $1.430979e-05$

Hospitalizações vs o3



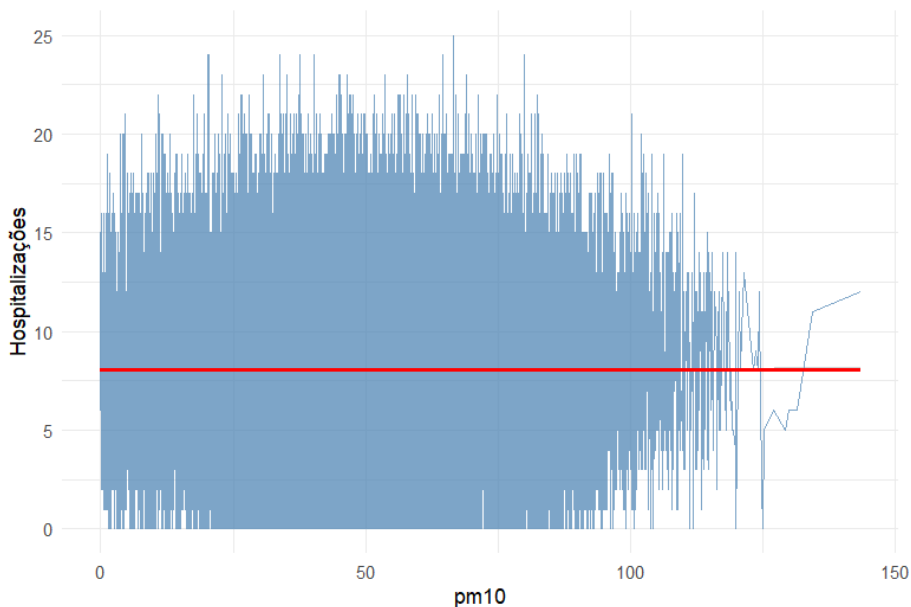
Correlação:
0.003031959
 R^2 : 9.192777e-06

Hospitalizações vs no2



Correlação:
0.001279014
 R^2 : 1.635877e-06

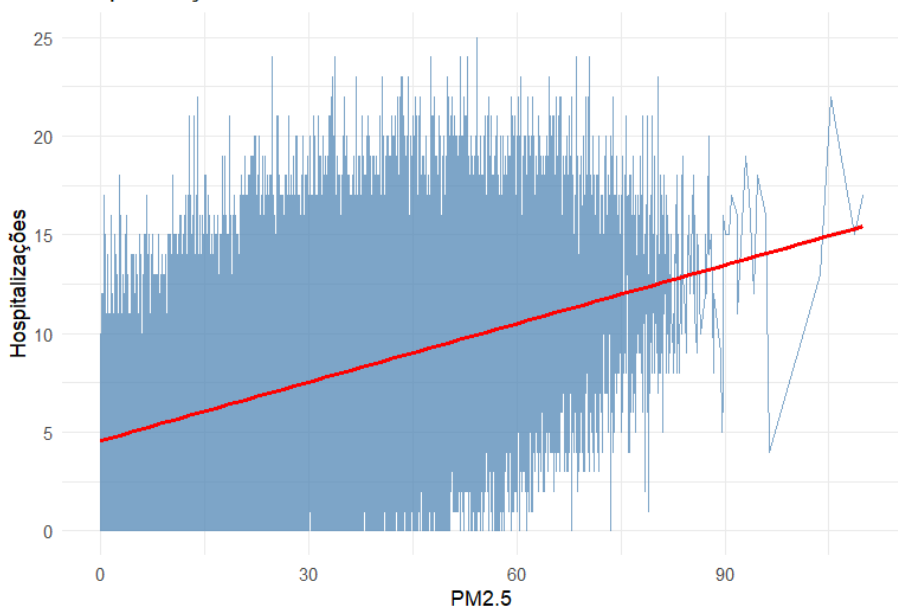
Hospitalizações vs pm10



Correlação: 0.0005374457

R²: 2.888479e-07

Hospitalizações vs PM2.5

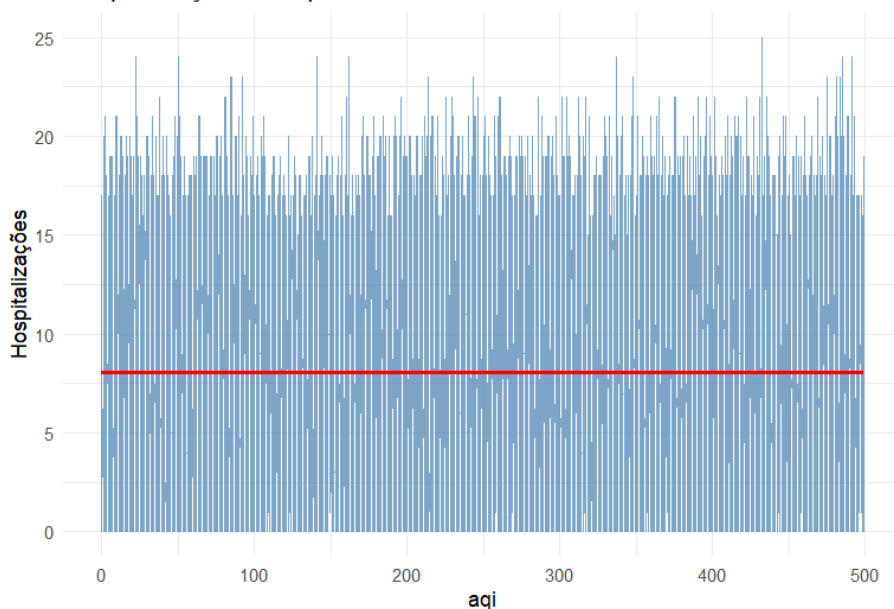


Correlação: 0.3923088

R² : 0.1539062

O gráfico mostra que há uma relação positiva entre PM2.5 e hospitalizações, mas essa associação é fraca: apenas 15% da variação nas internações é explicada pelos níveis de PM2.5.

Hospitalizações vs aqi



Correlação: -0.0003933545

R²: 1.547277e-07

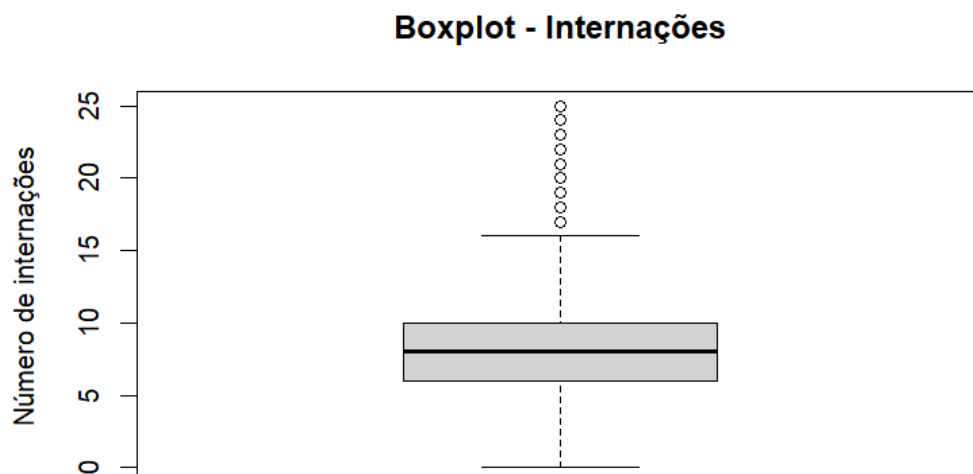
4. RESULTADOS E DISCUSSÕES

4.1 Preparação dos dados

Inicialmente, carregamos os dados no ambiente RStudio e seguimos com a preparação dos dados para análise. As funções `any(is.na())` e `sum(is.na())` indicaram que não existem valores vazios na base de dados, o que permitiu seguir com as análises sem necessidade de imputação ou remoção de dados.

Em seguida, a coluna `date` (de datas) foi convertida para o formato `Date`, a fim de permitir manipulações temporais adequadas. As variáveis qualitativas, como `city` (cidade) e `population_density` (densidade populacional), foram convertidas para o tipo fator (`factor`), permitindo que fossem tratadas corretamente em análises estatísticas e modelos.

Foi gerado um boxplot da variável `hospital_admissions` com o objetivo de identificar possíveis valores discrepantes (*outliers*). A partir do gráfico, observou-se uma quantidade significativa de dias com internações superiores a 16 casos. Utilizando o pacote `dplyr`, foi feita uma filtragem dessas observações, totalizando 1.373 registros considerados *outliers*.



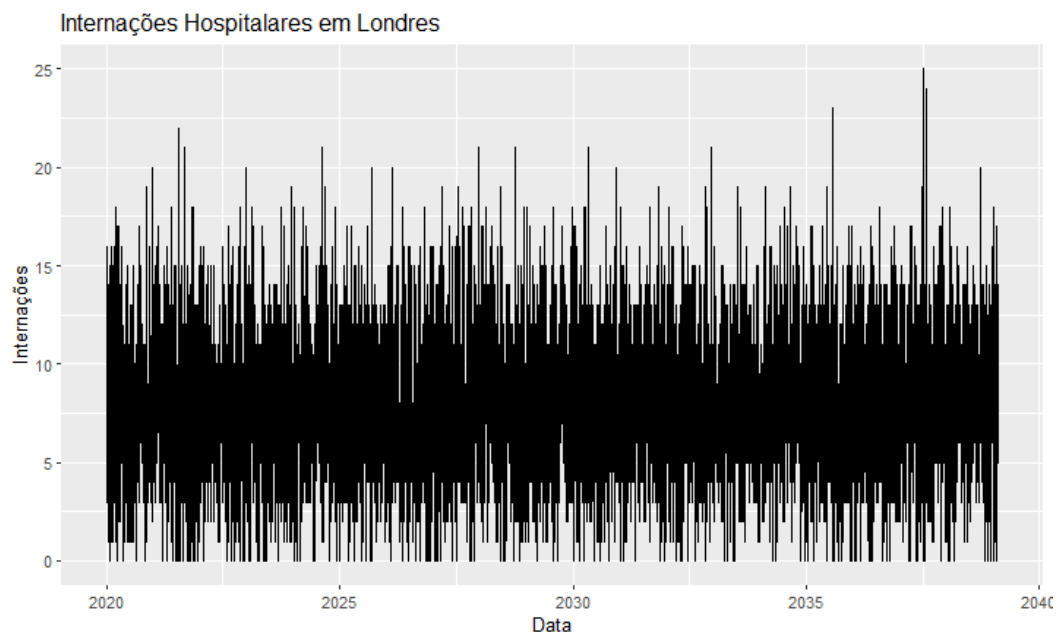
Ao investigar esses casos, verificou-se que estavam associados a datas futuras. Uma análise complementar revelou que a base de dados é simulada, construída para fins de aprendizado de máquina. Os picos de internação refletem cenários simulados baseados em eventos históricos e nas projeções de crescimento de problemas respiratórios em grandes centros urbanos. Por esse motivo, optou-se por manter os outliers na análise,

considerando que representam situações plausíveis e relevantes para o escopo da pesquisa.

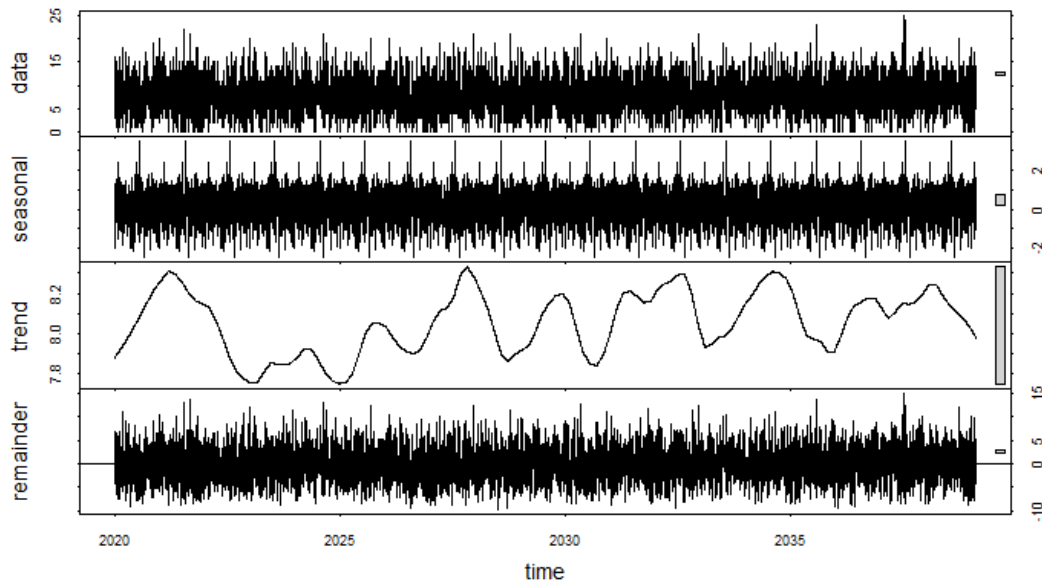
4.2 Análise das séries temporais

Feita a preparação dos dados, partimos para a análise das séries temporais, a fim de mensurar conceitos como a tendência, ciclos e sazonalidade das internações ao longo do tempo. Para tanto, criamos séries temporais distintas para cada cidade a fim de facilitar a exploração.

No gráfico abaixo, observamos a variação do número de internações ao longo dos dias(entre 2020 a 2039, aproximadamente) representado pelas linhas pretas. O gráfico possui alta variabilidade diária - alguns dias tem surto, outros têm quedas. É possível observar picos regulares onde as internações ultrapassam 15 ou 20 casos. Além de estacionariedade da série, já que o padrão não muda drasticamente ao longo dos anos.



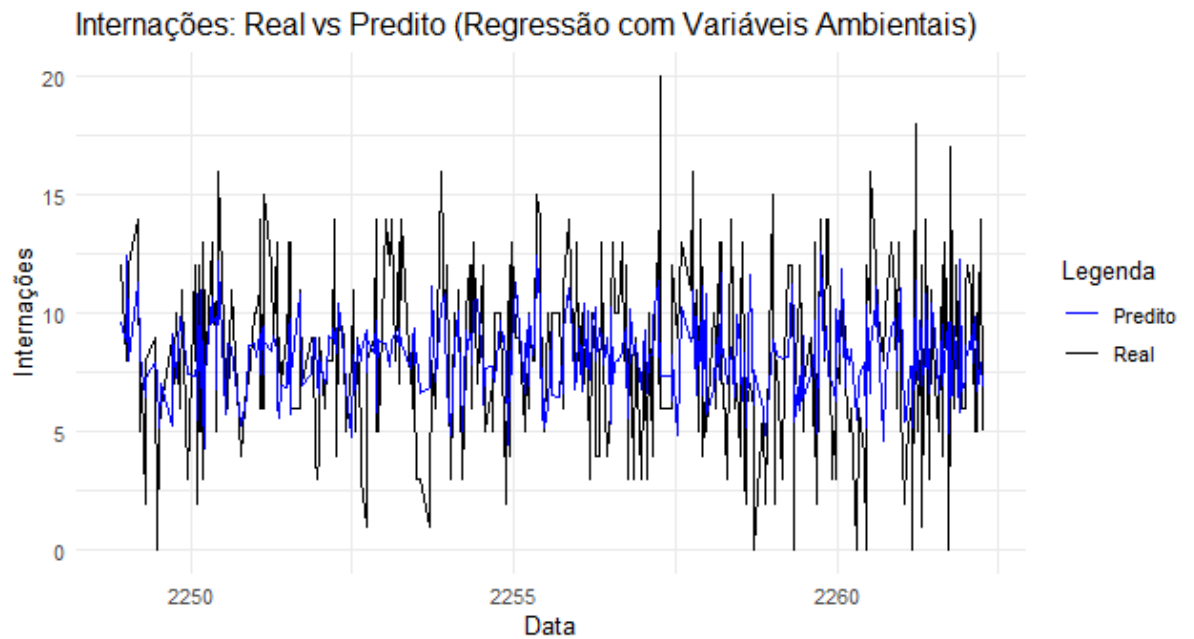
No entanto, o gráfico apresenta alto ruído, havendo a necessidade de aplicar técnicas de decomposição para visualizar tendências ou padrões sazonais com mais clareza.



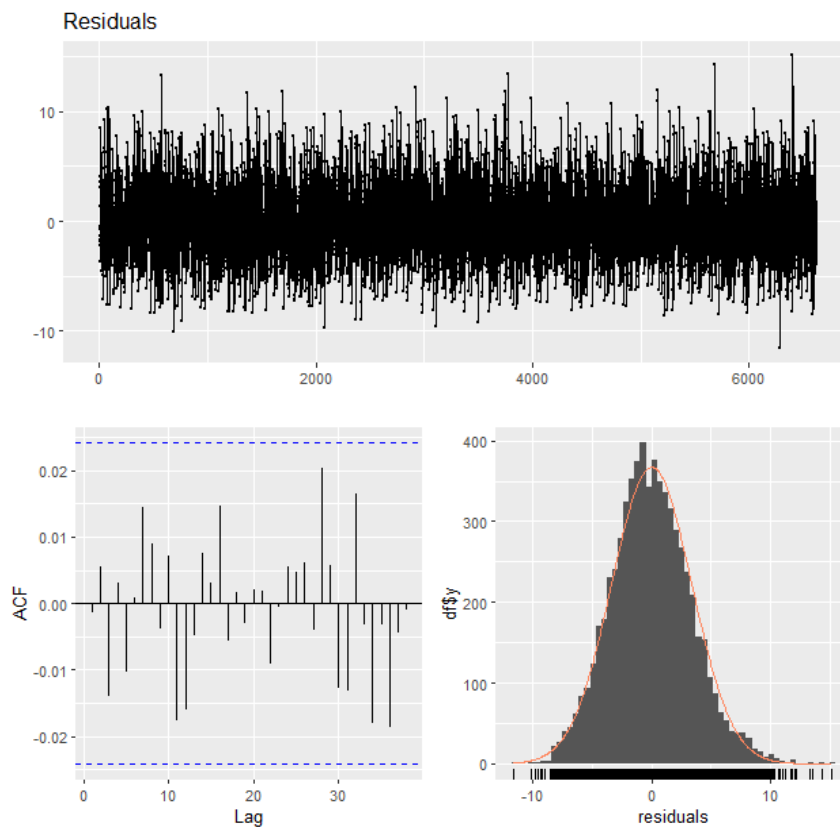
- O gráfico *data*(série original) apresenta alta variabilidade como já tínhamos notado;
- O componente sazonal *seasonal* indica uma oscilação periódica consistente, com padrão se repetindo ao longo do tempo, isto é, existe uma sazonalidade clara;
- O gráfico de tendência à longo prazo - *trend* - indica um comportamento cíclico, sem tendência à longo prazo;
- Por fim o gráfico *remainder*, apresenta os resíduos gerados após retirar a sazonalidade e tendência. Eles aparecem sem padrão claro, o que significa que os outros dois componentes já explicam bem a série;

4.3 Modelo de Previsão

Inicialmente criamos um modelo de previsão via modelo de regressão adicionando variáveis de sazonalidade: *date*, *year*, *month* e *weekday*. Só a sazonalidade não explica as variações de internação, indicando que fatores climáticos como o *aqi*, *temperature*, *humidity* e partículas poluentes tornam o modelo mais robusto como se observa abaixo:



Modelos alternativos como SARIMA (AutoRegressivos Integrados de Médias Móveis com Sazonalidade), apresentaram um RMSE igual à 3.61, superior ao RMSE da regressão múltipla igual à 3.19, além de uma análise visual inferior à apresentada acima. Um gráfico de resíduos do nosso modelo também apresentou bons resultados com valores aleatórios com média zero, isso indica que o modelo foi bem ajustado.



5. CONSIDERAÇÕES FINAIS

A análise conduzida demonstrou o potencial de modelos estatísticos na investigação da relação entre poluentes atmosféricos e internações hospitalares por doenças respiratórias. A partir da exploração do conjunto de dados voltado à qualidade do ar e desfechos de saúde em grandes centros urbanos, foi possível aplicar diferentes abordagens preditivas sobre a cidade de Londres, que serviu como estudo de caso.

A etapa exploratória inicial, com o cálculo do coeficiente de determinação (R^2), forneceu indícios de que variáveis ambientais como AQI, PM2.5, PM10, NO₂ e O₃, bem como temperatura e umidade, apresentam correlação com os níveis de internação. O modelo de regressão múltipla, alimentado por essas variáveis, mostrou-se capaz de capturar parte significativa da variação nos dados, apresentando erros preditivos relativamente baixos, conforme indicado pelas métricas MAE e RMSE.

A aplicação do modelo SARIMA, por outro lado, permitiu avaliar a série temporal de forma univariada, considerando apenas padrões internos, como tendência e sazonalidade. Embora sem o suporte direto das variáveis ambientais, o modelo SARIMA apresentou desempenho comparável em termos preditivos, evidenciando que a própria estrutura temporal da série carrega informações relevantes para a previsão das internações.

A comparação entre os dois modelos reforça a importância de considerar múltiplas abordagens na análise de dados em saúde pública, uma vez que diferentes modelos podem capturar aspectos complementares do fenômeno estudado. Além disso, destaca-se que a metodologia aplicada pode ser replicada para outras cidades presentes no conjunto de dados, oferecendo subsídios para decisões estratégicas e ações preventivas voltadas à melhoria da qualidade do ar e à redução dos impactos à saúde da população.

6. REFERÊNCIAS BIBLIOGRÁFICAS

RISHI R CARLONI. *Global Air Quality and Respiratory Health Outcomes Comprehensive Multi-City Data on AQ Metrics & Respiratory Hospitalizations*. [S.l.]: Kaggle, [2025]. Disponível em: <<https://www.kaggle.com/datasets/tfisthis/global-air-quality-and-respiratory-health-outcomes/data>>. Acesso em: 23 de junho, 2025.

7. ANEXOS

internacoes.R

2025-06-30

library(ggplot2)

Warning: pacote 'ggplot2' foi compilado no R versão 4.4.3

library(forecast)

Warning: pacote 'forecast' foi compilado no R versão 4.4.3

Registered S3 method overwritten by 'quantmod':

method from

as.zoo.data.frame zoo

library(dplyr)

##

Anexando pacote: 'dplyr'

Os seguintes objetos são mascarados por 'package:stats':

##

filter, lag

Os seguintes objetos são mascarados por 'package:base':

##

intersect, setdiff, setequal, union

library(readr)

1. Carregar e preparar os dados

data_frame <- read.csv("air_quality_health_dataset.csv")

Verificar se há valores ausentes

any(is.na(data_frame)) # Retorna TRUE se houver qualquer NA

[1] FALSE

sum(is.na(data_frame)) # Conta o total de valores NA

[1] 0

Converter coluna 'date' para o tipo Date

data_frame\$date <- as.Date(data_frame\$date, format = "%Y-%m-%d")

Transformar variáveis qualitativas em fatores

data_frame\$city <- as.factor(data_frame\$city)

```
data_frame$population_density <- as.factor(data_frame$population_density)
```

```
# 2. Análise exploratória dos dados
```

```
# Boxplot do número de internações
```

```
boxplot(data_frame$hospital_admissions,  
        main = "Boxplot - Internações",  
        ylab = "Número de internações")
```

```
# Visualizar casos com internações acima de 16 por dia
```

```
library(dplyr)
```

```
picos <- data_frame %>%
```

```
  filter(hospital_admissions > 16) %>%
```

```
  arrange(desc(hospital_admissions)) %>%
```

```
  select(date, hospital_admissions)
```

```
print(picos)
```

```
##      date hospital_admissions
```

```
## 1  2240-07-26             25
```

```
## 2  2060-08-19             24
```

```
## 3  2120-08-16             24
```

```
## 4  2141-02-18             24
```

```
## 5  2145-07-01             24
```

```
## 6  2186-01-20             24
```

```
## 7  2231-12-30             24
```

```
## 8  2241-03-04             24
```

```
## 9  2067-11-16             23
```

```
## 10 2078-07-27             23
```

```
##...
```

```
#-----|
```

```
# 3. Análise de correlação dos dados|
```

```
#-----|
```

```
# Análise 1 – hospital_admissions vs date
```

```
ggplot(data_frame, aes(x = date, y = hospital_admissions)) +
```

```
  geom_line(color = "steelblue", alpha = 0.7) +
```

```
  geom_smooth(method = "lm", col = "red") +
```

```
  labs(title = "Hospitalizações vs data", x = "Data", y = "Hospitalizações") +
```

```

theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
modelo_date <- lm(hospital_admissions ~ date, data = data_frame)
summary(modelo_date)$r.squared
## [1] 3.542533e-06
# Análise 2 – hospital_admissions vs aqi
ggplot(data_frame, aes(x = aqi, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs AQI", x = "AQI", y = "Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$aqi, use = "complete.obs")
## [1] -0.0003933545
modelo_aqi <- lm(hospital_admissions ~ aqi, data = data_frame)
summary(modelo_aqi)$r.squared
## [1] 1.547277e-07
# Análise 3 – hospital_admissions vs pm2_5
ggplot(data_frame, aes(x = pm2_5, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs PM2.5", x = "PM2.5", y = "Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$pm2_5, use =
"complete.obs")
## [1] 0.3923088
modelo_pm2_5 <- lm(hospital_admissions ~ pm2_5, data = data_frame)
summary(modelo_pm2_5)$r.squared
## [1] 0.1539062
# Análise 4 – hospital_admissions vs pm10
ggplot(data_frame, aes(x = pm10, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +

```



```

labs(title = "Hospitalizações vs PM10", x = "PM10", y = "Hospitalizações") +
theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$pm10, use =
"complete.obs")
## [1] 0.0005374457
modelo_pm10 <- lm(hospital_admissions ~ pm10, data = data_frame)
summary(modelo_pm10)$r.squared
## [1] 2.888479e-07
# Análise 5 – hospital_admissions vs no2
ggplot(data_frame, aes(x = no2, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs NO2", x = "NO2", y = "Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$no2, use = "complete.obs")
## [1] 0.001279014
modelo_no2 <- lm(hospital_admissions ~ no2, data = data_frame)
summary(modelo_no2)$r.squared
## [1] 1.635877e-06
# Análise 6 – hospital_admissions vs o3
ggplot(data_frame, aes(x = o3, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs O3", x = "O3", y = "Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$o3, use = "complete.obs")
## [1] -0.003031959
modelo_o3 <- lm(hospital_admissions ~ o3, data = data_frame)
summary(modelo_o3)$r.squared
## [1] 9.192777e-06
# Análise 7 – hospital_admissions vs hospital_capacity

```

```

ggplot(data_frame, aes(x = hospital_capacity, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs Leitos disponíveis", x = "Leitos disponíveis",
y = "Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$hospital_capacity, use =
"complete.obs")
## [1] 0.001533149
modelo_hospital_capacity <- lm(hospital_admissions ~ hospital_capacity, data
= data_frame)
summary(modelo_hospital_capacity)$r.squared
## [1] 2.350546e-06
# Análise 8 – hospital_admissions vs temperature
ggplot(data_frame, aes(x = temperature, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Hospitalizações vs Temperatura", x = "Temperatura", y =
"Hospitalizações") +
  theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$temperature, use =
"complete.obs")
## [1] -0.003782829
modelo_temperature <- lm(hospital_admissions ~ temperature, data =
data_frame)
summary(modelo_temperature)$r.squared
## [1] 1.430979e-05
# Análise 9 – hospital_admissions vs humidity
ggplot(data_frame, aes(x = humidity, y = hospital_admissions)) +
  geom_line(color = "steelblue", alpha = 0.7) +
  geom_smooth(method = "lm", col = "red") +

```

```

    labs(title = "Hospitalizações vs Umidade", x = "Umidade", y =
"Hospitalizações") +
    theme_minimal()
## `geom_smooth()` using formula = 'y ~ x'
cor(data_frame$hospital_admissions, data_frame$humidity, use =
"complete.obs")
## [1] 9.882925e-06
modelo_humidity <- lm(hospital_admissions ~ humidity, data = data_frame)
summary(modelo_humidity)$r.squared
## [1] 9.76722e-11
# 4. Filtrar dados para uma cidade específica (Londres)
london_data <- subset(data_frame, city == "London")

# Criar série temporal com frequência diária
london_ts <- ts(london_data$hospital_admissions, start = c(2020, 1),
frequency = 365)
# 5. Visualizar a série temporal
# Construir dataframe com datas correspondentes
london_df <- data.frame(
  date = seq.Date(from = as.Date("2020-01-01"), length.out = length(london_ts),
by = "day"),
  hospital_admissions = as.numeric(london_ts)
)

# Gráfico de linhas da série temporal
ggplot(london_df, aes(x = date, y = hospital_admissions)) +
  geom_line() +
  labs(title = "Internações Hospitalares em Londres", x = "Data", y =
"Internações")
# 6. Decomposição da série temporal
# Decomposição com STL (Seasonal-Trend decomposition using Loess)
decomp <- stl(london_ts, s.window = "periodic")
plot(decomp)
#-----

```

7. Criando modelos de previsão

#-----

Criação do dataframe temporal com as variáveis relevantes

```
london_df <- london_data %>%  
  mutate(  
    date = as.Date(date),  
    year = as.numeric(format(date, "%Y")),  
    month = as.factor(format(date, "%m")),  
    weekday = as.factor(weekdays(date))  
  ) %>%  
  select(  
    date,  
    total = hospital_admissions,  
    year, month, weekday,  
    aqi, pm2_5, pm10, no2, o3, temperature, humidity  
  )
```

8. Separar dados de treino e teste (últimos 365 dias para teste)

```
indice_teste <- tail(1:nrow(london_df), 365)
```

```
treino <- london_df[-indice_teste, ]
```

```
teste <- london_df[indice_teste, ]
```

9. Ajustar modelo de regressão múltipla com variáveis ambientais

```
modelo_multiplo <- lm(total ~ year + month + weekday + aqi + pm2_5 + pm10 +  
  no2 + o3 + temperature + humidity,  
  data = treino)
```

Verificar resíduos

```
checkresiduals(modelo_multiplo)
```

```
##
```

```
## Breusch-Godfrey test for serial correlation of order up to 29
```

```
##
```

```
## data: Residuals
```

```
## LM test = 14.919, df = 29, p-value = 0.9856
```

10. Prever valores no conjunto de teste

```
prev_mult <- predict(modelo_multiplo, newdata = teste)
```

```
real <- teste$total
```

Calcular erro absoluto

```
abs_error <- function(yreal, yprev) {
```

```
  abs(yreal - yprev)
```

```
}
```

```
erro <- abs_error(real, prev_mult)
```

11. Visualizar previsão vs valores reais

```
comparacao_df <- data.frame(
```

```
  date = teste$date,
```

```
  real = real,
```

```
  predito = prev_mult
```

```
)
```

```
ggplot(comparacao_df, aes(x = date)) +
```

```
  geom_line(aes(y = real, color = "Real")) +
```

```
  geom_line(aes(y = predito, color = "Predito")) +
```

```
  labs(title = "Internações: Real vs Predito (Regressão com Variáveis Ambientais)",
```

```
        x = "Data", y = "Internações") +
```

```
  scale_color_manual(name = "Legenda", values = c("Real" = "black",  
"Predito" = "blue")) +
```

```
  theme_minimal()
```

12. Métricas de erro

```
mae <- mean(erro)
```

```
rmse <- sqrt(mean(erro^2))
```

```
cat("MAE:", mae, "\nRMSE:", rmse, "\n")
```

```
## MAE: 2.593089
```

```
## RMSE: 3.196303
```

```
#-----
```

```
# MODELO SARIMA
```

```
#-----
```

```
library(forecast)
library(ggplot2)

# Criando a série temporal completa (supondo dados diários de hospitalizações)
london_ts <- ts(data_frame$hospital_admissions, start = c(2020, 1), frequency
= 365)

# Tamanho da série completa
n <- length(london_ts)

# Separando treino (tudo menos o último ano = 365 dias)
treino <- window(london_ts, end = c(2020 + (n - 365 - 1)/365))

# Separando teste (últimos 365 dias)
test_ts <- window(london_ts, start = c(2020 + (n - 365)/365))

# Ajustando o modelo SARIMA
modelo_sarima <- auto.arima(treino, seasonal = TRUE, stepwise = TRUE,
approximation = TRUE)

checkresiduals(modelo_sarima)
##
## Ljung-Box test
##
## data: Residuals from ARIMA(0,0,0) with non-zero mean
## Q* = 786.71, df = 730, p-value = 0.07151
##
## Model df: 0. Total lags used: 730
# Previsão para o período de teste
forecast_sarima <- forecast(modelo_sarima, h = length(test_ts))

# Avaliação
```

```
real_sarima <- as.numeric(test_ts)
prev_sarima <- as.numeric(forecast_sarima$mean)

mae_sarima <- mean(abs(real_sarima - prev_sarima))
rmse_sarima <- sqrt(mean((real_sarima - prev_sarima)^2))

# Exibindo resultados
cat("Modelo SARIMA:\n")
## Modelo SARIMA:
cat("MAE:", round(mae_sarima, 4), "\n")
## MAE: 3.1272
cat("RMSE:", round(rmse_sarima, 4), "\n")
## RMSE: 3.8537
```