



# Previsão de Séries Temporais com o {fable}

Paula Maçaira

Semana da Estatística 2022 - UFJF

## Quem sou eu

- ▶ Bacharelado em Estatística (ENCE, 2013)
- ▶ Mestrado em Eng. Elétrica (PUC-Rio, 2015)
- ▶ Doutorado em Eng. de Produção (PUC-Rio, 2018)
- ▶ Pós-doutorado em Eng. de Produção (PUC-Rio, 2019)
- ▶ Professora Adjunta desde 2019 (DEI, PUC-Rio)

We're data-driven!



## Me encontre em

- ▶ @paula\_macaira
- ▶ [github.com/paulamacaira](https://github.com/paulamacaira)
- ▶ [sites.google.com/view/paulamacaira](https://sites.google.com/view/paulamacaira)
- ▶ paulamacaira@puc-rio.br



## Premissas

- ▶ Presumo que você conheça minimamente a **linguagem R** e o meio ambiente **RStudio**
- ▶ Suponho que você seja familiarizado com conceitos como **séries temporais, média, desvio padrão, quantis, regressão, distribuição normal, probabilidade** etc.

# Falaremos sobre...

1. o que é um tsibble
2. alguns gráficos de ST
3. benchmarks para previsão
4. análise dos resíduos
5. medidas de acurácia
6. métodos de Suavização Exponencial
7. modelos de Box & Jenkins



Let's Go

## Pacotes necessários

*# Manipulação de dados*

```
library(tidyverse)
```

*# Manipulação de Séries Temporais*

```
library(tsibble)
```

*# Funções de Previsão*

```
library(fable)
```

*# Gráficos e Estatísticas de Séries Temporais*

```
library(feasts)
```

*# Séries Temporais Tidy*

```
library(tsibbledata)
```

*# Todos os itens acima e mais*

```
library(fpp3)
```

## O que é um tsibble

Um tsibble permite o armazenamento e manipulação de múltiplas séries temporais em R

Ele contém:

- ▶ um index (informação de tempo);
- ▶ variáveis medidas; e
- ▶ variáveis chave (identificadores únicos opcionais para cada série)

## Alguns exemplos

### global\_economy

```
## # A tsibble: 15,150 x 9 [1Y]
## # Key:      Country [263]
## #       Country     Code   Year     GDP Growth    CPI Imports Exports Population
## #       <fct>      <fct> <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## # 1 Afghanistan AFG     1960 537777811.    NA    NA    7.02    4.13    8996351
## # 2 Afghanistan AFG     1961 548888896.    NA    NA    8.10    4.45    9166764
## # 3 Afghanistan AFG     1962 546666678.    NA    NA    9.35    4.88    9345868
## # 4 Afghanistan AFG     1963 751111191.    NA    NA   16.9    9.17    9533954
## # 5 Afghanistan AFG     1964 800000044.    NA    NA   18.1    8.89    9731361
## # 6 Afghanistan AFG     1965 1006666638.   NA    NA   21.4    11.3    9938414
## # 7 Afghanistan AFG     1966 1399999967.   NA    NA   18.6    8.57    10152331
## # 8 Afghanistan AFG     1967 1673333418.   NA    NA   14.2    6.77    10372630
## # 9 Afghanistan AFG     1968 1373333367.   NA    NA   15.2    8.90    10604346
## # 10 Afghanistan AFG    1969 1408888922.   NA    NA   15.0    10.1    10854428
## # ... with 15,140 more rows
## # i Use `print(n = ...)` to see more rows
```

# Alguns exemplos

## tourism

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:      Region, State, Purpose [304]
##   Quarter Region     State       Purpose Trips
##   <qtr>    <chr>    <chr>       <chr>   <dbl>
## 1 1998 Q1 Adelaide South Australia Business 135.
## 2 1998 Q2 Adelaide South Australia Business 110.
## 3 1998 Q3 Adelaide South Australia Business 166.
## 4 1998 Q4 Adelaide South Australia Business 127.
## 5 1999 Q1 Adelaide South Australia Business 137.
## 6 1999 Q2 Adelaide South Australia Business 200.
## 7 1999 Q3 Adelaide South Australia Business 169.
## 8 1999 Q4 Adelaide South Australia Business 134.
## 9 2000 Q1 Adelaide South Australia Business 154.
## 10 2000 Q2 Adelaide South Australia Business 169.
## # ... with 24,310 more rows
## # i Use `print(n = ...)` to see more rows
```

# Convertendo um data frame/tibble em tsibble

```
consumo = readr::read_csv2("https://raw.githubusercontent.com/paulamacaира/Semana-da-Estatistica-2022-UFJF/main/Minicurso-fable/consumo_energia_2022.csv")
consumo

## # A tibble: 224 x 6
##   Data      Norte Nordeste Sudeste   Sul `Centro-Oeste`
##   <date>    <dbl>   <dbl>   <dbl> <dbl>       <dbl>
## 1 2004-01-01 1574.   4457.  14930.  4510.      1507.
## 2 2004-02-01 1507.   4140.  14715.  4690.      1456.
## 3 2004-03-01 1574.   4373.  14904.  4899.      1535.
## 4 2004-04-01 1599.   4462.  15501.  4873.      1611.
## 5 2004-05-01 1625.   4424.  14907.  4723.      1566.
## 6 2004-06-01 1641.   4409.  14899.  4557.      1577.
## 7 2004-07-01 1665.   4376.  15094.  4599.      1612.
## 8 2004-08-01 1730.   4439.  15190.  4653.      1646.
## 9 2004-09-01 1687.   4570.  15594.  4654.      1727.
## 10 2004-10-01 1724.   4658.  15840.  4591.      1743.
## # ... with 214 more rows
## # i Use `print(n = ...)` to see more rows
```

O link do arquivo no GitHub é:

"[https://raw.githubusercontent.com/paulamacaира/Semana-da-Estatistica-2022-UFJF/main/Minicurso-fable/consumo\\_energia\\_2022.csv](https://raw.githubusercontent.com/paulamacaира/Semana-da-Estatistica-2022-UFJF/main/Minicurso-fable/consumo_energia_2022.csv)"

## Convertendo para tsibble

```
consumo_tsibble = consumo %>%
  mutate(Data = yearmonth(Data)) %>%
  pivot_longer(-Data, names_to = "Regiao",
              values_to = "Consumo") %>%
  as_tsibble(index = Data, key = Regiao)
consumo_tsibble
```

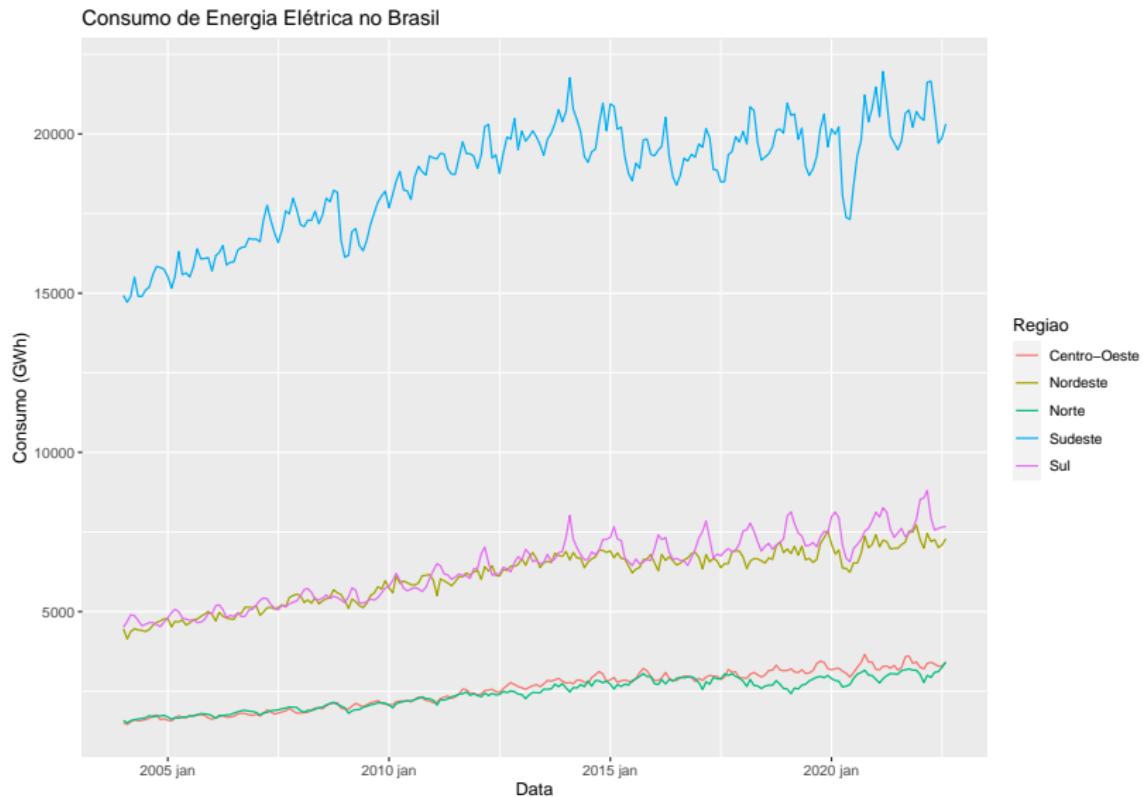
## Convertendo para tsibble

```
## # A tsibble: 1,120 x 3 [1M]
## # Key:      Regiao [5]
##       Data  Regiao     Consumo
##       <mth> <chr>     <dbl>
## 1 2004 jan Centro-Oeste 1507.
## 2 2004 fev Centro-Oeste 1456.
## 3 2004 mar Centro-Oeste 1535.
## 4 2004 abr Centro-Oeste 1611.
## 5 2004 mai Centro-Oeste 1566.
## 6 2004 jun Centro-Oeste 1577.
## 7 2004 jul Centro-Oeste 1612.
## 8 2004 ago Centro-Oeste 1646.
## 9 2004 set Centro-Oeste 1727.
## 10 2004 out Centro-Oeste 1743.
## # ... with 1,110 more rows
## # i Use `print(n = ...)` to see more rows
```

# Alguns gráficos de Séries Temporais

```
consumo_tsibble %>%
  autoplot(Consumo) +
  labs(x="Data",y="Consumo (GWh)",
       title="Consumo de Energia Elétrica no Brasil")
```

# Alguns gráficos de Séries Temporais

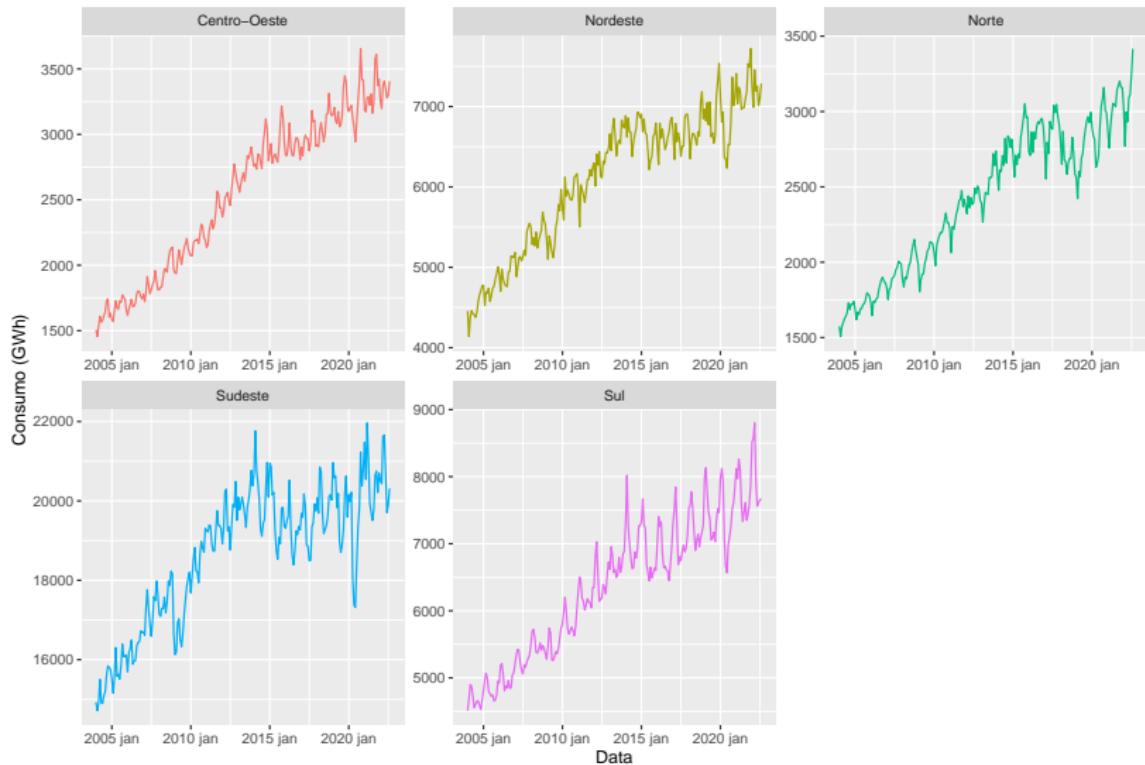


# Alguns gráficos de Séries Temporais

```
consumo_tsibble %>%
  autoplot(Consumo) +
  facet_wrap(~Regiao, scales = "free") +
  labs(x="Data",y="Consumo (GWh)",
       title="Consumo de Energia Elétrica no Brasil") +
  theme(legend.position = "none")
```

# Alguns gráficos de Séries Temporais

Consumo de Energia Elétrica no Brasil



## Componentes de uma Série Temporal

- ▶ Tendência: quando há um aumento ou diminuição de longo prazo nos dados
- ▶ Sazonalidade: quando uma série é influenciada por fatores sazonais (por exemplo, o trimestre de o ano, o mês ou o dia da semana)
- ▶ Ciclo: quando a série apresenta padrões que não estão fixos no tempo

Sazonalidade vs Ciclo: O momento de picos e depressões é previsível com dados sazonais, mas imprevisível a longo prazo com dados cíclicos

- ▶ Podemos usar a função de autocorrelação (seu gráfico) para visualizar essas componentes

# Função de autocorrelação (ou ACF)

- ▶ Correlação do instante  $t$  com  $t - k$

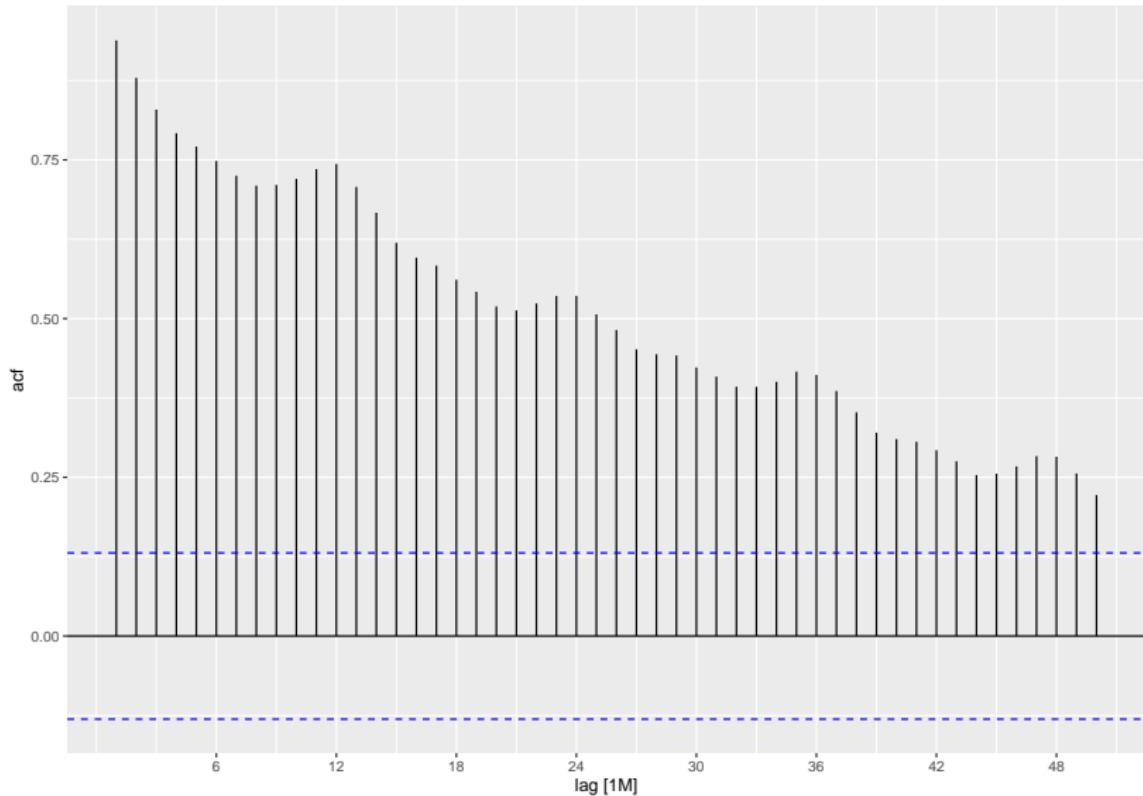
```
consumo_tsibble %>%
  filter(Regiao == "Sudeste") %>%
  ACF(Consumo, lag_max = 50)
```

```
## # A tsibble: 50 x 3 [1M]
## # Key:      Regiao [1]
##   Regiao    lag    acf
##   <chr>    <lag> <dbl>
## 1 Sudeste  1M  0.938
## 2 Sudeste  2M  0.879
## 3 Sudeste  3M  0.829
## 4 Sudeste  4M  0.792
## 5 Sudeste  5M  0.771
## 6 Sudeste  6M  0.748
## 7 Sudeste  7M  0.725
## 8 Sudeste  8M  0.709
## 9 Sudeste  9M  0.710
## 10 Sudeste 10M  0.720
## # ... with 40 more rows
## # i Use `print(n = ...)` to see more rows
```

## Função de autocorrelação (ou ACF)

```
consumo_tsibble %>%
  filter(Regiao == "Sudeste") %>%
  ACF(Consumo, lag_max = 50) %>%
  autoplot()
```

# Função de autocorrelação (ou ACF)



## Componentes da ST na ACF

- ▶ Presença de tendência: as autocorrelações para pequenos lags tendem a ser grandes e positivas
- ▶ Presença de sazonalidade: as autocorrelações serão maiores em lags múltiplos da frequência sazonal
- ▶ Tendência + Sazonalidade: combinação desses dois efeitos

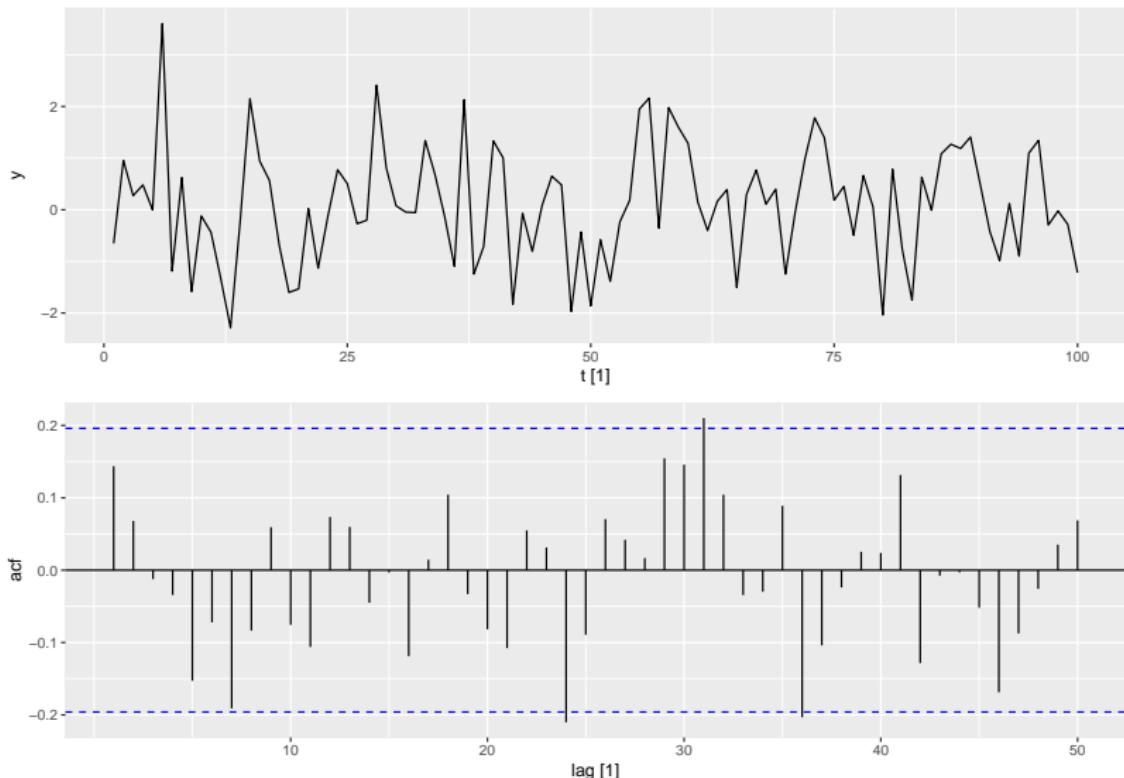
## Ruído Branco

- ▶ Dados não estão correlacionados ao longo do tempo, possuem média zero e variância constante (teoricamente, também exigimos independência)

```
rb = tsibble(t = seq(100), y = rnorm(100), index = t)
rb %>% autoplot(y) -> p1
rb %>% ACF(y, lag_max = 50) %>% autoplot() -> p2
```

# Ruído Branco

```
library(patchwork); p1 / p2
```



## Decomposição de uma Série Temporal

- ▶ Aditiva:  $Y_t = T_t + S_t + R_t$
- ▶ Multiplicativa:  $Y_t = T_t \times S_t \times R_t$

Onde,

- ▶  $Y_t$ : Série Temporal
- ▶  $T_t$ : Componente de Tendência e Ciclo
- ▶  $S_t$ : Componente Sazonal
- ▶  $R_t$ : Resto (Remainder)

## Decomposição STL

- ▶ Seasonal and Trend decomposition using Loess (STL)
  - ▶ versátil e robusta para outliers (se desejável)
  - ▶ formulação apenas aditiva
  - ▶ usar transformação Box-Cox para outras formulações

# Decomposição STL

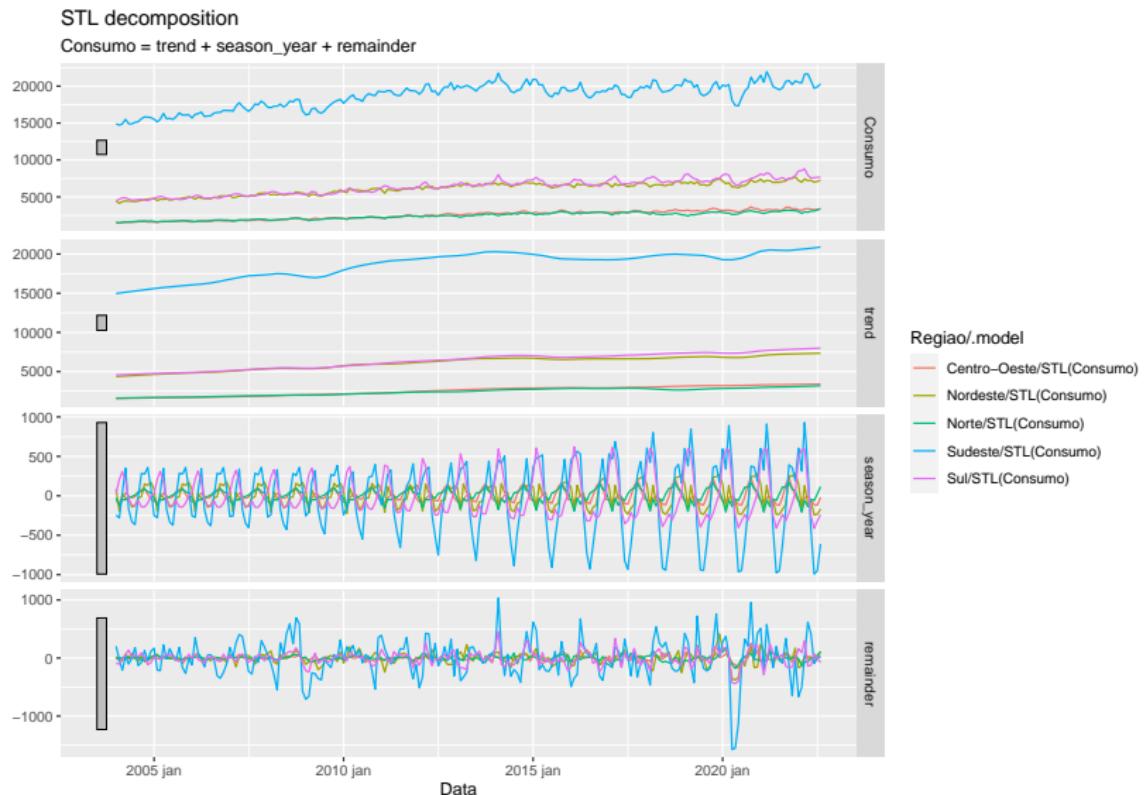
```
consumo_tsibble %>%
  model(STL(Consumo)) %>%
  components()
```

```
## # A dable: 1,120 x 8 [1M]
## # Key:      Regiao, .model [5]
## # :
##   Consumo = trend + season_year + remainder
##     Regiao      .model          Data Consumo trend season_year remain~1 seaso~2
##     <chr>      <chr>        <mth>    <dbl> <dbl>       <dbl>    <dbl>    <dbl>
## 1 Centro-Oeste STL(Consumo) 2004 jan  1507. 1552.     -68.8     23.6  1576.
## 2 Centro-Oeste STL(Consumo) 2004 fev  1456. 1561.     -84.3     -20.8  1540.
## 3 Centro-Oeste STL(Consumo) 2004 mar  1535. 1570.     -27.3     -7.89  1562.
## 4 Centro-Oeste STL(Consumo) 2004 abr  1611. 1579.     36.2     -4.43  1574.
## 5 Centro-Oeste STL(Consumo) 2004 mai  1566. 1588.    -0.888    -20.9  1567.
## 6 Centro-Oeste STL(Consumo) 2004 jun  1577. 1596.     -27.8      8.75  1605.
## 7 Centro-Oeste STL(Consumo) 2004 jul  1612. 1605.     -10.9     17.7  1623.
## 8 Centro-Oeste STL(Consumo) 2004 ago  1646. 1614.     34.2     -2.18  1611.
## 9 Centro-Oeste STL(Consumo) 2004 set  1727. 1622.     81.0     23.7  1646.
## 10 Centro-Oeste STL(Consumo) 2004 out 1743. 1631.     88.9     23.4  1655.
## # ... with 1,110 more rows, and abbreviated variable names 1: remainder,
## #   2: season_adjust
## # i Use `print(n = ...)` to see more rows
```

## Decomposição STL

```
consumo_tsibble %>%
  model(STL(Consumo)) %>%
  components() %>%
  autoplot()
```

# Decomposição STL



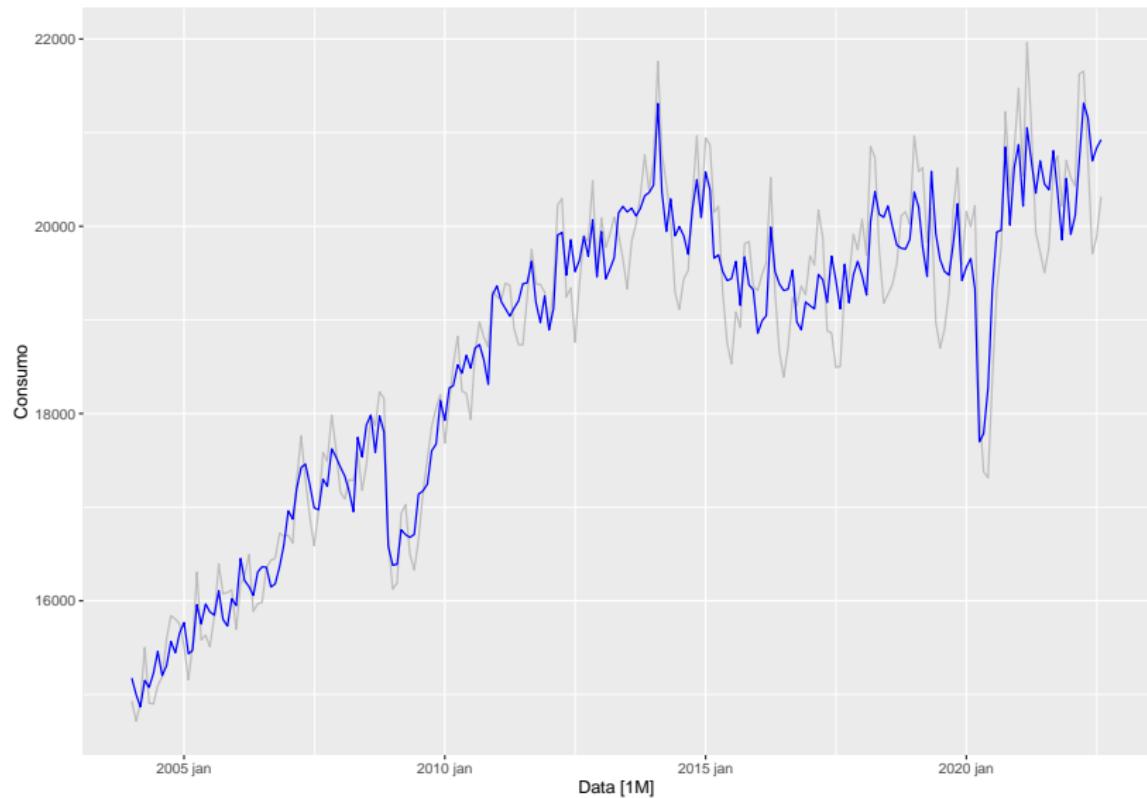
## Ajuste Sazonal

- ▶ Aditiva:  $Y_t - S_t = T_t + R_t$
- ▶ Multiplicativa:  $Y_t/S_t = T_t \times R_t$

```
dcmp = consumo_tsibble %>% filter(Regiao == "Sudeste") %>%
  model(STL(Consumo)) %>% components()

consumo_tsibble %>% filter(Regiao == "Sudeste") %>%
  autoplot(Consumo, col = "gray") +
  autolayer(dcmp, season_adjust, color = "blue")
```

# Ajuste Sazonal



## Tendência e Sazonalidade

$$\text{Força Tendência} = \max(0, 1 - \text{Var}(Rt) / \text{Var}(Tt + Rt))$$

$$\text{Força Sazonal} = \max(0, 1 - \text{Var}(Rt) / \text{Var}(St + Rt))$$

# Tendência e Sazonalidade

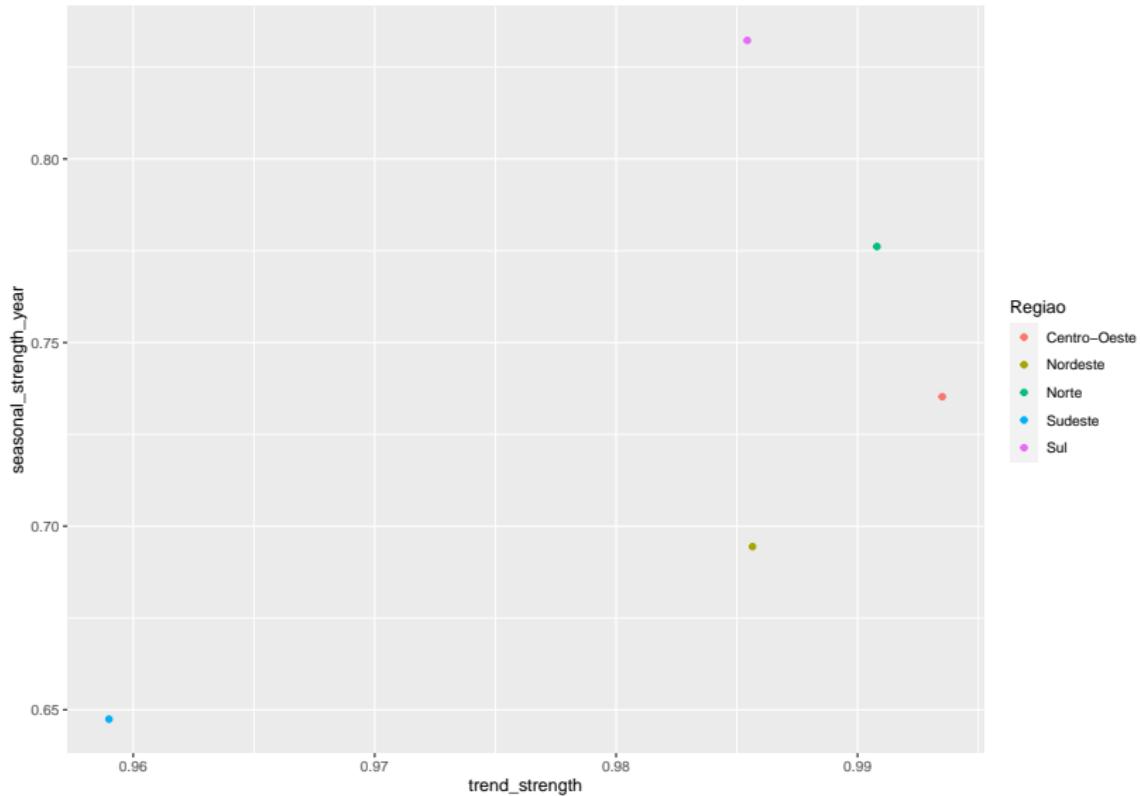
```
consumo_tsibble %>%
  features(Consumo, feat_stl)

## # A tibble: 5 x 10
##   Regiao trend~1 seaso~2 seaso~3 seaso~4 spiki~5 linea~6 curva~7 stl_e~8 stl_e~9
##   <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Centr~    0.994    0.735     10      7  3.32e2   8655.   -828.    0.207   0.269
## 2 Norde~    0.986    0.694      0       6  9.14e3  11902.  -2852.    0.313   0.296
## 3 Norte     0.991    0.776     10      2  3.58e2   6695.   -939.    0.389   0.406
## 4 Sudes~    0.959    0.647      3       6  1.54e6  21426.  -8621.    0.386   0.396
## 5 Sul       0.985    0.832      2       6  1.77e4  14702.  -1709.    0.381   0.461
## # ... with abbreviated variable names 1: trend_strength,
## #   2: seasonal_strength_year, 3: seasonal_peak_year, 4: seasonal_trough_year,
## #   5: spikiness, 6: linearity, 7: curvature, 8: stl_e_acf1, 9: stl_e_acf10
```

## Tendência e Sazonalidade

```
consumo_tsibble %>%
  features(Consumo, feat_stl) %>%
  ggplot(aes(x = trend_strength,
              y = seasonal_strength_year,
              col = Regiao)) +
  geom_point()
```

# Tendência e Sazonalidade



## Previsão estatística

- ▶  $y_{T+h}$ : “coisa” a ser prevista
- ▶  $y_1, \dots, y_T$ : o que conhecemos/histórico
- ▶  $\hat{y}_{T+h|T} = E[y_{T+h}|y_1, \dots, y_T]$ : previsão pontual

## Métodos benchmark

- ▶ MEAN( $y$ ): previsões são iguais a média histórica
- ▶ NAIIVE( $y$ ): previsões são iguais ao último valor observado
- ▶ SNAIVE( $y \sim \text{lag}(m)$ ): previsões iguais ao último valor do mesmo período
- ▶ RW( $y \sim \text{drift}()$ ): previsões iguais ao último valor mais variação média

## Função model()

- A função model() treina o modelo nos dados

```
consumo_fit = consumo_tsibble %>%
  model(Seasonal_naive = SNAIVE(Consumo) ,
        Naive = NAIVE(Consumo) ,
        Drift = RW(Consumo ~ drift()) ,
        Mean = MEAN(Consumo))
consumo_fit
```

## Função model()

```
## # A mable: 1 x 5
## # Key:      Regiao [1]
##   Regiao Seasonal_naive    Naive          Drift     Mean
##   <chr>       <model> <model>       <model> <model>
## 1 Sudeste      <SNAIVE> <NAIVE> <RW w/ drift> <MEAN>
```

- ▶ o mable é uma tabela com os modelos ajustados

## Função forecast()

- ▶ Para produzir as previsões use a função forecast()

```
consumo_fc = consumo_fit %>%
  forecast(h = 12)
consumo_fc
```

# Função forecast()

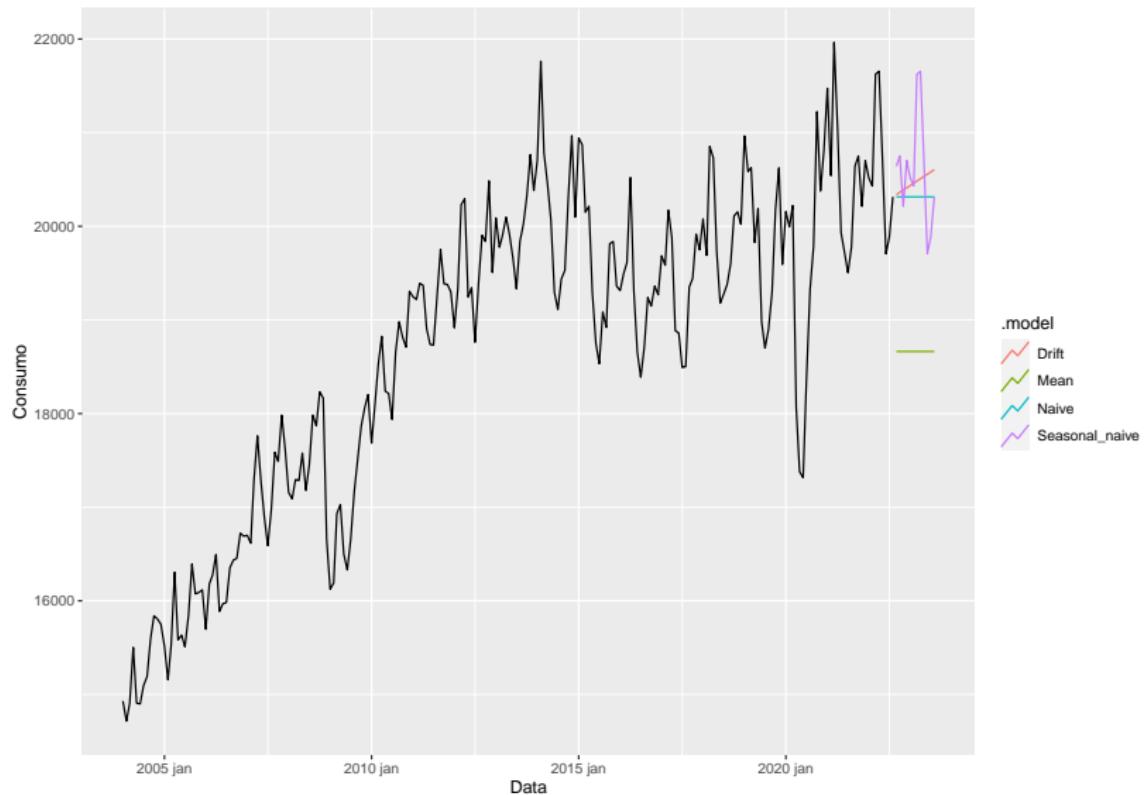
```
## # A fable: 48 x 5 [1M]
## # Key:     Regiao, .model [4]
##   Regiao   .model          Data      Consumo  .mean
##   <chr>    <chr>        <mth>     <dist>   <dbl>
## 1 Sudeste Seasonal_naive 2022 set N(20642, 707823) 20642.
## 2 Sudeste Seasonal_naive 2022 out N(20751, 707823) 20751.
## 3 Sudeste Seasonal_naive 2022 nov N(20213, 707823) 20213.
## 4 Sudeste Seasonal_naive 2022 dez N(20702, 707823) 20702.
## 5 Sudeste Seasonal_naive 2023 jan N(20518, 707823) 20518.
## 6 Sudeste Seasonal_naive 2023 fev N(20430, 707823) 20430.
## 7 Sudeste Seasonal_naive 2023 mar N(21624, 707823) 21624.
## 8 Sudeste Seasonal_naive 2023 abr N(21656, 707823) 21656.
## 9 Sudeste Seasonal_naive 2023 mai N(20731, 707823) 20731.
## 10 Sudeste Seasonal_naive 2023 jun N(19707, 707823) 19707.
## # ... with 38 more rows
## # i Use `print(n = ...)` to see more rows
```

- ▶ o fable é uma tabela de previsão com previsões pontuais e distribuições

## Plotando as previsões

```
consumo_fc %>%  
  autoplot(consumo_tsibble, level = NULL)
```

# Plotando as previsões



## Resíduos

$$\blacktriangleright e_t = y_t - \hat{y}_{t|t-1}$$

Premissas:

- ▶  $e_t$  não são correlacionados, caso sejam: ficaram informações nos resíduos que deveriam estar no modelo
- ▶  $e_t$  possui média zero, caso não seja então as previsões são viesadas

# Resíduos

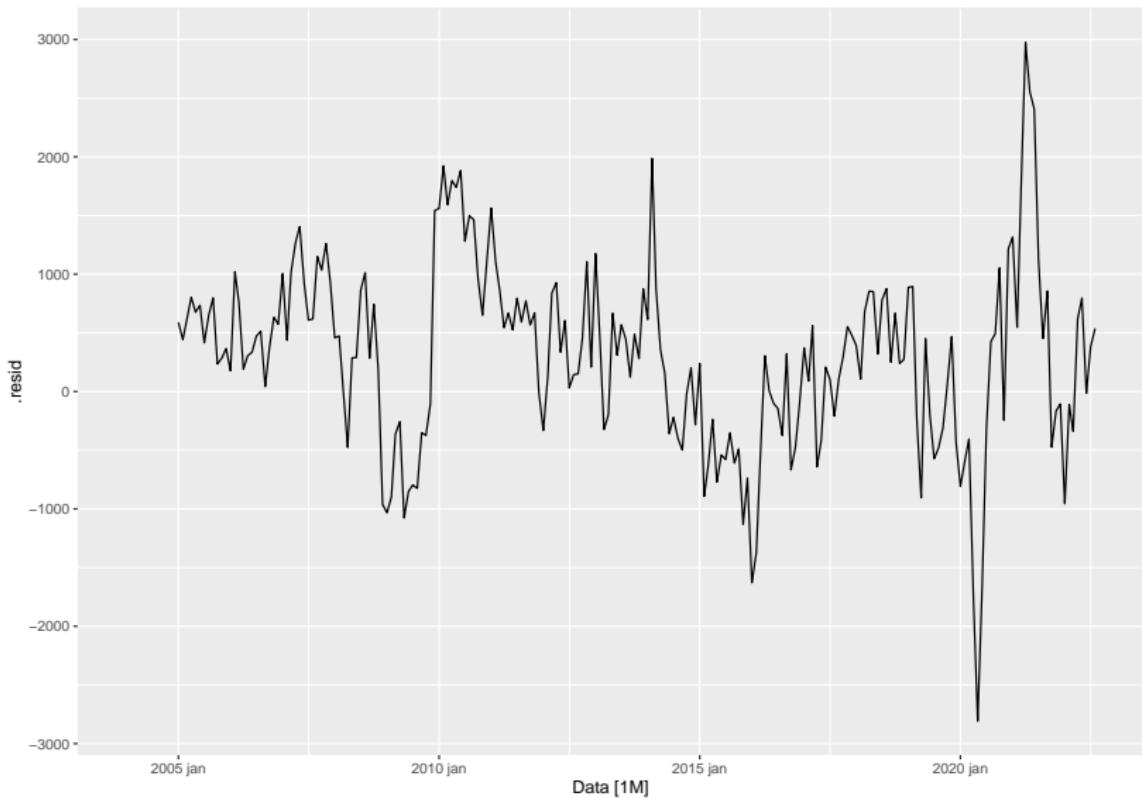
```
augment(consumo_fit)
```

```
## # A tsibble: 896 x 7 [1M]
## # Key:      Regiao, .model [4]
##   Regiao   .model      Data Consumo .fitted .resid .innov
##   <chr>    <chr>      <mth>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Sudeste Seasonal_naive 2004 jan  14930.     NA     NA     NA
## 2 Sudeste Seasonal_naive 2004 fev  14715.     NA     NA     NA
## 3 Sudeste Seasonal_naive 2004 mar  14904.     NA     NA     NA
## 4 Sudeste Seasonal_naive 2004 abr  15501.     NA     NA     NA
## 5 Sudeste Seasonal_naive 2004 mai  14907.     NA     NA     NA
## 6 Sudeste Seasonal_naive 2004 jun  14899.     NA     NA     NA
## 7 Sudeste Seasonal_naive 2004 jul  15094.     NA     NA     NA
## 8 Sudeste Seasonal_naive 2004 ago  15190.     NA     NA     NA
## 9 Sudeste Seasonal_naive 2004 set  15594.     NA     NA     NA
## 10 Sudeste Seasonal_naive 2004 out  15840.    NA     NA     NA
## # ... with 886 more rows
## # i Use `print(n = ...)` to see more rows
```

## Resíduos

```
augment(consumo_fit) %>%
  filter(.model == "Seasonal_naive") %>%
  autoplot(.resid)
```

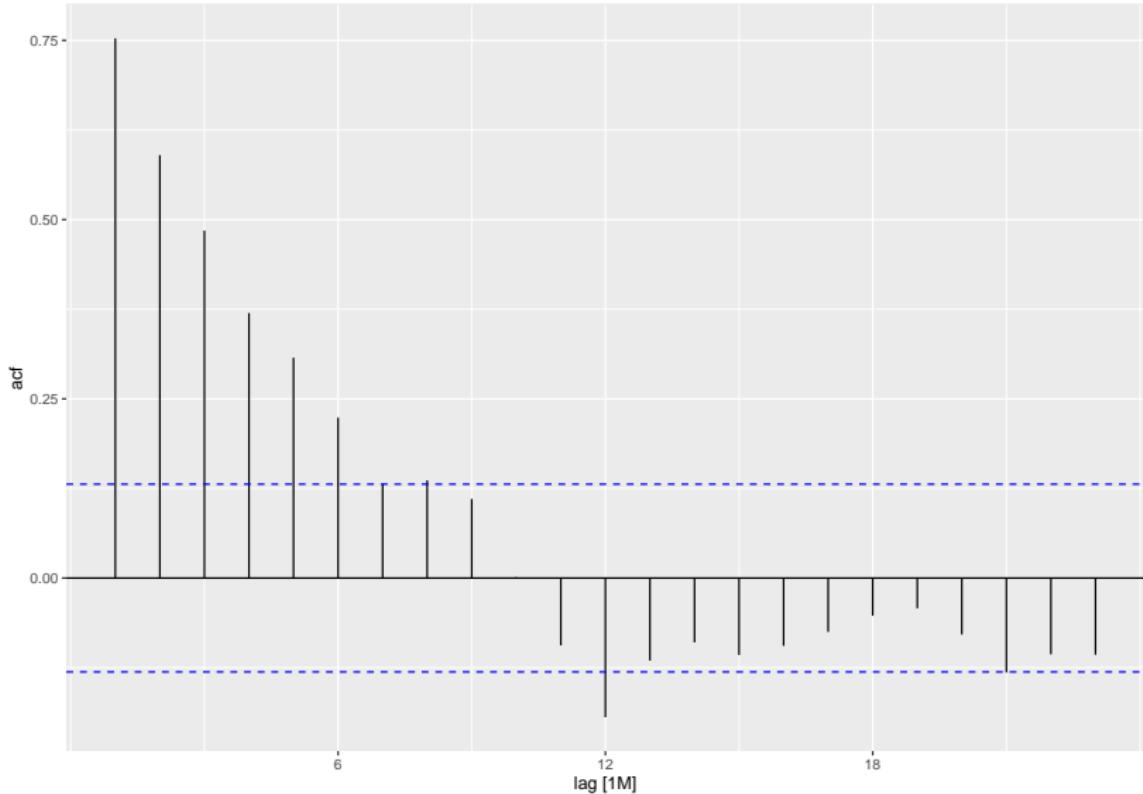
# Resíduos



## Resíduos (e a ACF)

```
augment(consumo_fit) %>%
  filter(.model == "Seasonal_naive") %>%
  ACF(.resid) %>%
  autoplot()
```

## Resíduos (e a ACF)



## Resíduos (Teste de Ljung-Box)

- ▶ Não queremos rejeitar  $H_0$ : os resíduos são iid

```
augment(consumo_fit) %>%
  features(.resid, ljung_box)
```

```
## # A tibble: 4 x 4
##   Regiao   .model      lb_stat lb_pvalue
##   <chr>    <chr>        <dbl>     <dbl>
## 1 Sudeste Drift       0.240     0.624
## 2 Sudeste Mean        200.      0
## 3 Sudeste Naive       0.240     0.624
## 4 Sudeste Seasonal_naive 122.     0
```

## Medidas de acurácia

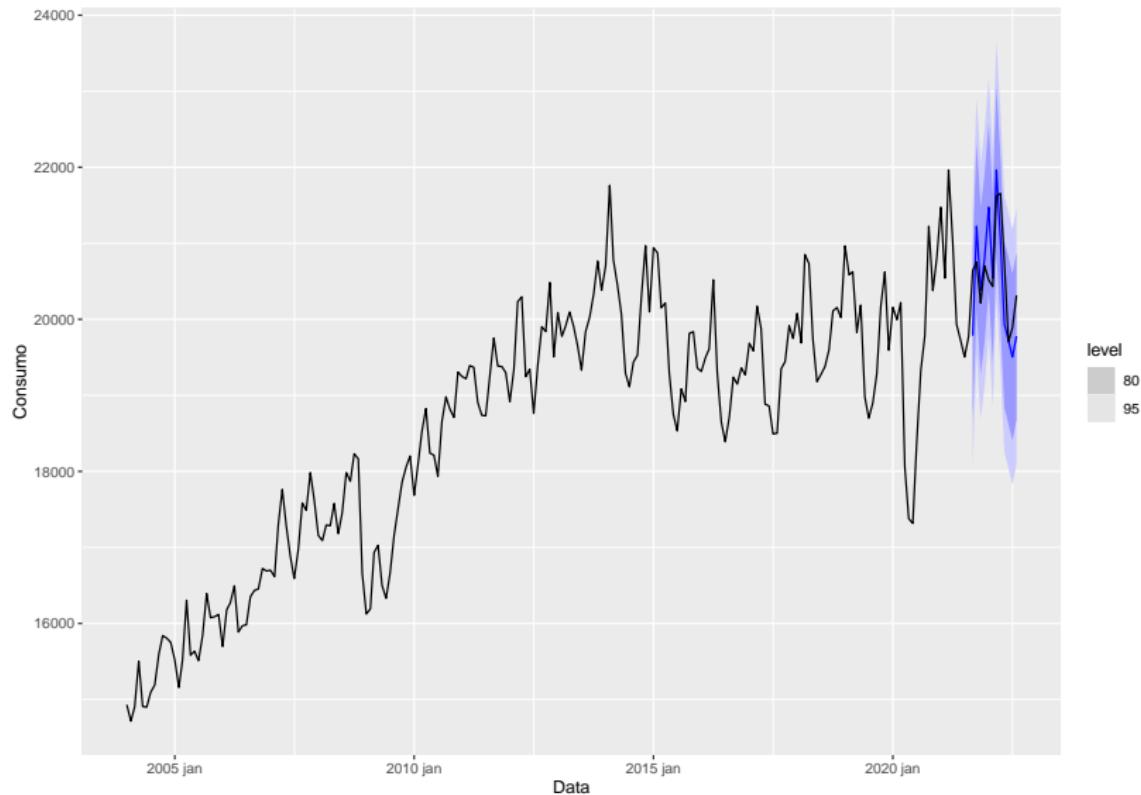
- ▶ Primeiro vamos selecionar apenas a região Sudeste e o modelo “Seasonal\_naive” para que a visualização fique mais fácil

```
consumo_fit = consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
  model(Seasonal_naive = SNAIVE(Consumo))

consumo_fc = consumo_fit %>% forecast(h = 12)

consumo_fc %>%
  autoplot(consumo_tsibble)
```

# Medidas de acurácia



## Medidas de acurácia

- ▶  $e_t = y_t - \hat{y}_{t|t-1}$
- ▶  $MAE = mean(|e_{T+h}|)$
- ▶  $MSE = mean(e_{T+h}^2)$
- ▶  $MAPE = 100 \times mean(|e_{T+h}|/|y_{T+h}|)$
- ▶  $RMSE = sqrt(mean(e_{T+h}^2))$
- ▶ MAE, MSE e RMSE são dependentes da escala dos dados, enquanto o MAPE é independente da escala, porém só é sensível a valores positivos

# Medidas de acurácia

```
accuracy(consumo_fc, consumo_tsibble)
```

```
## # A tibble: 1 x 11
##   .model      Regiao .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Seasonal_naive Sudeste Test    85.2 539.  446. 0.412  2.15  0.657 0.630 0.126
```

# Modelos ETS

- ▶ ETS é uma taxonomia para modelos de Suavização Exponencial
- ▶ E(Error) T(Trend) S(Season)
  - ▶ Erro: aditivo (A) ou multiplicativo (M)
  - ▶ Tendência: nenhuma (N), aditiva (A), multiplicativa (M) ou amortecida (Ad ou Md)
  - ▶ Sazonalidade: nenhuma (N), aditiva (A) ou multiplicativa (M)

# Modelos ETS

| Trend Component         | Seasonal Component |             |                  |
|-------------------------|--------------------|-------------|------------------|
|                         | N                  | A           | M                |
|                         | (None)             | (Additive)  | (Multiplicative) |
| N (None)                | (N,N)              | (N,A)       | (N,M)            |
| A (Additive)            | (A,N)              | (A,A)       | (A,M)            |
| $A_d$ (Additive damped) | ( $A_d$ ,N)        | ( $A_d$ ,A) | ( $A_d$ ,M)      |

| Short hand  | Method                              |
|-------------|-------------------------------------|
| (N,N)       | Simple exponential smoothing        |
| (A,N)       | Holt's linear method                |
| ( $A_d$ ,N) | Additive damped trend method        |
| (A,A)       | Additive Holt-Winters' method       |
| (A,M)       | Multiplicative Holt-Winters' method |
| ( $A_d$ ,M) | Holt-Winters' damped method         |

## Ajustando o ETS

```
consumo_ets = consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
model(ets = ETS(Consumo))
```

## Olhando o modelo ajustado

```
consumo_ets
```

```
## # A mable: 1 x 2
## # Key:      Regiao [1]
##   Regiao          ets
##   <chr>          <model>
## 1 Sudeste <ETS(M,Ad,M)>
```

# Olhando o modelo ajustado com mais detalhes

```
report(consumo_ets)
```

```
## Series: Consumo
## Model: ETS(M,Ad,M)
##   Smoothing parameters:
##     alpha = 0.7918555
##     beta  = 0.000150039
##     gamma = 0.0001159213
##     phi   = 0.9792373
##
##   Initial states:
##     l[0]      b[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]
## 14849.65 66.68413 1.005455 1.021454 1.013903 1.001061 0.9837097 0.9633066
##     s[-6]      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## 0.9675771 0.9848672 1.020219 1.022489 1.00687 1.009088
##
##   sigma^2:  6e-04
##
##       AIC      AICc      BIC
## 3728.900 3732.444 3789.318
```

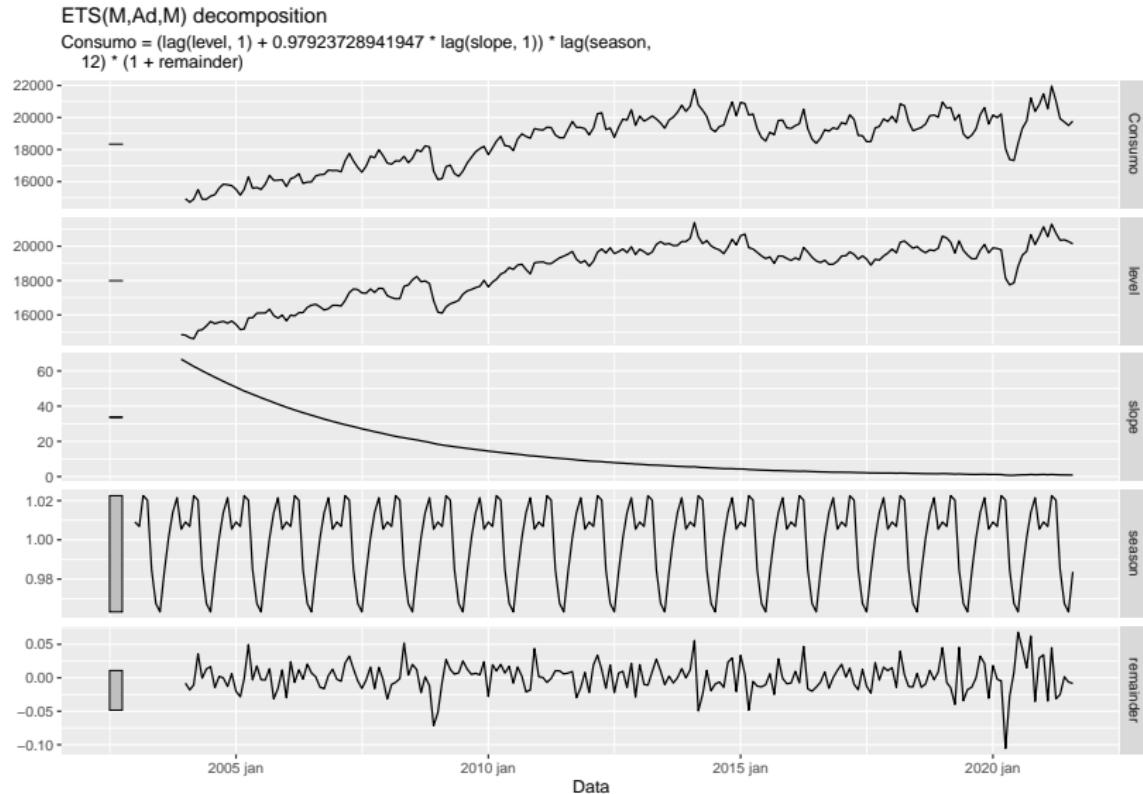
# Observando as componentes do modelo ajustado

```
components(consumo_ets)
```

```
## # A dable: 224 x 8 [1M]
## # Key:      Regiao, .model [1]
## # :
## #   Consumo = (lag(level, 1) + 0.97923728941947 * lag(slope, 1)) *
## #   lag(season, 12) * (1 + remainder)
## #   Regiao  .model      Data Consumo level slope season remainder
## #   <chr>    <chr>    <mth>  <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 Sudeste ets 2003 jan     NA     NA     NA  1.01     NA
## 2 Sudeste ets 2003 fev     NA     NA     NA  1.01     NA
## 3 Sudeste ets 2003 mar     NA     NA     NA  1.02     NA
## 4 Sudeste ets 2003 abr     NA     NA     NA  1.02     NA
## 5 Sudeste ets 2003 mai     NA     NA     NA  0.985    NA
## 6 Sudeste ets 2003 jun     NA     NA     NA  0.968    NA
## 7 Sudeste ets 2003 jul     NA     NA     NA  0.963    NA
## 8 Sudeste ets 2003 ago     NA     NA     NA  0.984    NA
## 9 Sudeste ets 2003 set     NA     NA     NA  1.00     NA
## 10 Sudeste ets 2003 out    NA     NA     NA  1.01     NA
## # ... with 214 more rows
## # i Use `print(n = ...)` to see more rows
```

# Plotando as componentes

```
components(consumo_ets) %>% autoplot()
```

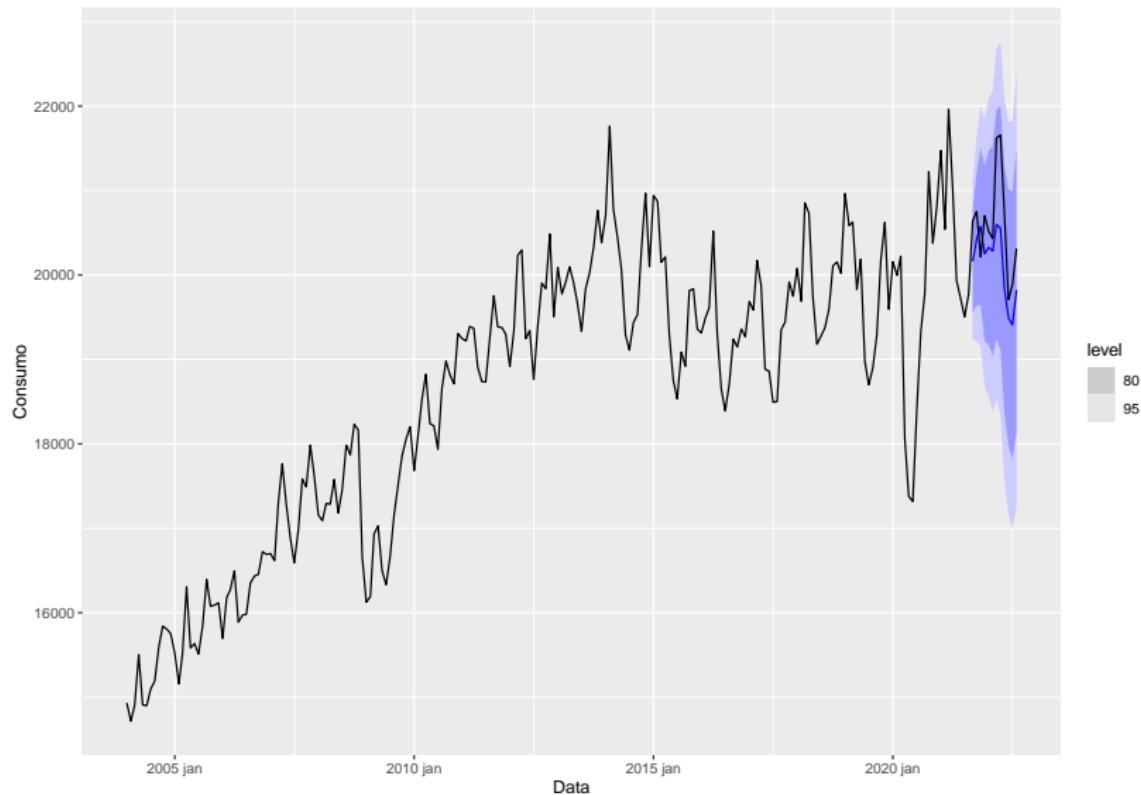


## Realizando a previsão e plotando

```
consumo_ets_fc = consumo_ets %>% forecast(h = 12)

consumo_ets_fc %>%
  autoplot(consumo_tsibble)
```

# Realizando a previsão e plotando



## Modificando o modelo ajustado “na mão”

```
consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
  model(ets = ETS(Consumo ~ trend("M"))) %>%
  forecast(h = 12) %>%
  autoplot(consumo_tsibble)
```

## Simulação via Bootstrap

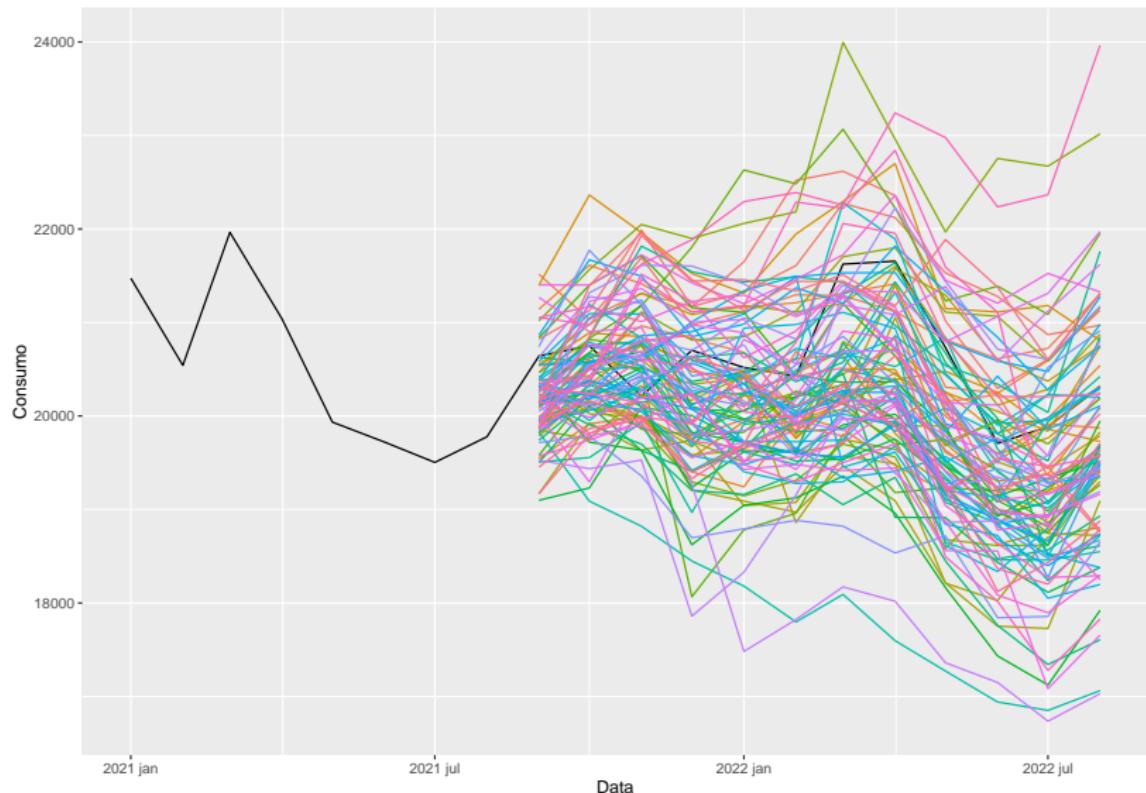
```
sim = consumo_ets %>%
  generate(h = 12, times = 100, bootstrap = TRUE)
sim
```

```
## # A tsibble: 1,200 x 5 [1M]
## # Key:      Regiao, .model, .rep [100]
##   Regiao   .model     Data .rep     .sim
##   <chr>    <chr>     <mth> <chr>   <dbl>
## 1 Sudeste  ets      2021 set  1     20821.
## 2 Sudeste  ets      2021 out  1     21128.
## 3 Sudeste  ets      2021 nov  1     21989.
## 4 Sudeste  ets      2021 dez  1     21168.
## 5 Sudeste  ets      2022 jan  1     21648.
## 6 Sudeste  ets      2022 fev  1     22521.
## 7 Sudeste  ets      2022 mar  1     22617.
## 8 Sudeste  ets      2022 abr  1     22357.
## 9 Sudeste  ets      2022 mai  1     21016.
## 10 Sudeste ets     2022 jun  1     20294.
## # ... with 1,190 more rows
## # i Use `print(n = ...)` to see more rows
```

## Visualização as simulações

```
consumo_tsibble %>%
  filter(year(Data) >= 2021,
        Regiao == "Sudeste") %>%
  ggplot(aes(x = Data)) +
  geom_line(aes(y = Consumo)) +
  geom_line(aes(y = .sim, colour = as.factor(.rep)),
            data = sim) +
  guides(col = FALSE)
```

# Visualização as simulações



## Modelos ARIMA (ou Box & Jenkins)

- ▶ AR: autoregressivo (observações defasadas como input)
- ▶ I: integrado (diferenciação para tornar a série estacionária)
- ▶ MA: média móvel (erros defasados como input)

## Ajustando o ARIMA

```
consumo_arima = consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
model(arima = ARIMA(Consumo))
```

## Olhando o modelo ajustado com mais detalhes

```
report(consumo_arima)

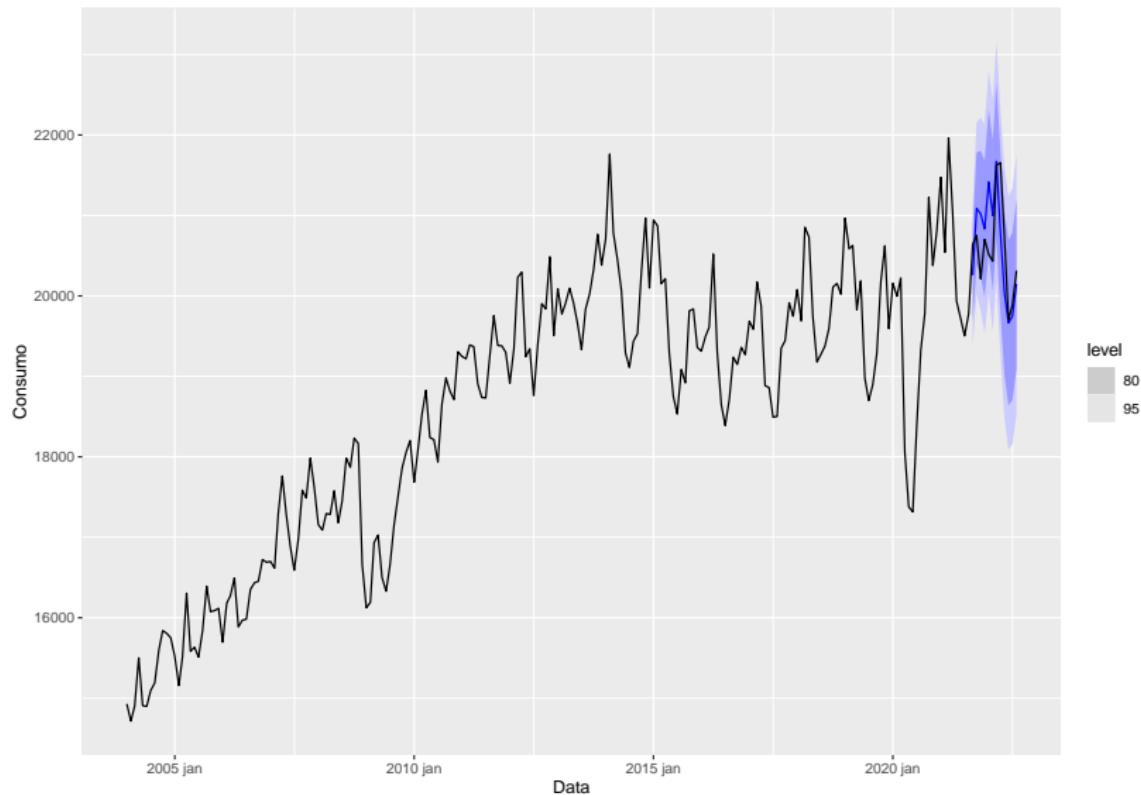
## Series: Consumo
## Model: ARIMA(1,0,1)(0,1,1)[12] w/ drift
##
## Coefficients:
##             ar1      ma1     sma1  constant
##             0.9063 -0.2124 -0.6648   27.1876
## s.e.    0.0373  0.0849  0.0679   8.9059
##
## sigma^2 estimated as 198225: log likelihood=-1505.42
## AIC=3020.84    AICc=3021.14    BIC=3037.33
```

## Realizando a previsão e plotando

```
consumo_arima_fc = consumo_arima %>% forecast(h = 12)

consumo_arima_fc %>%
  autoplot(consumo_tsibble)
```

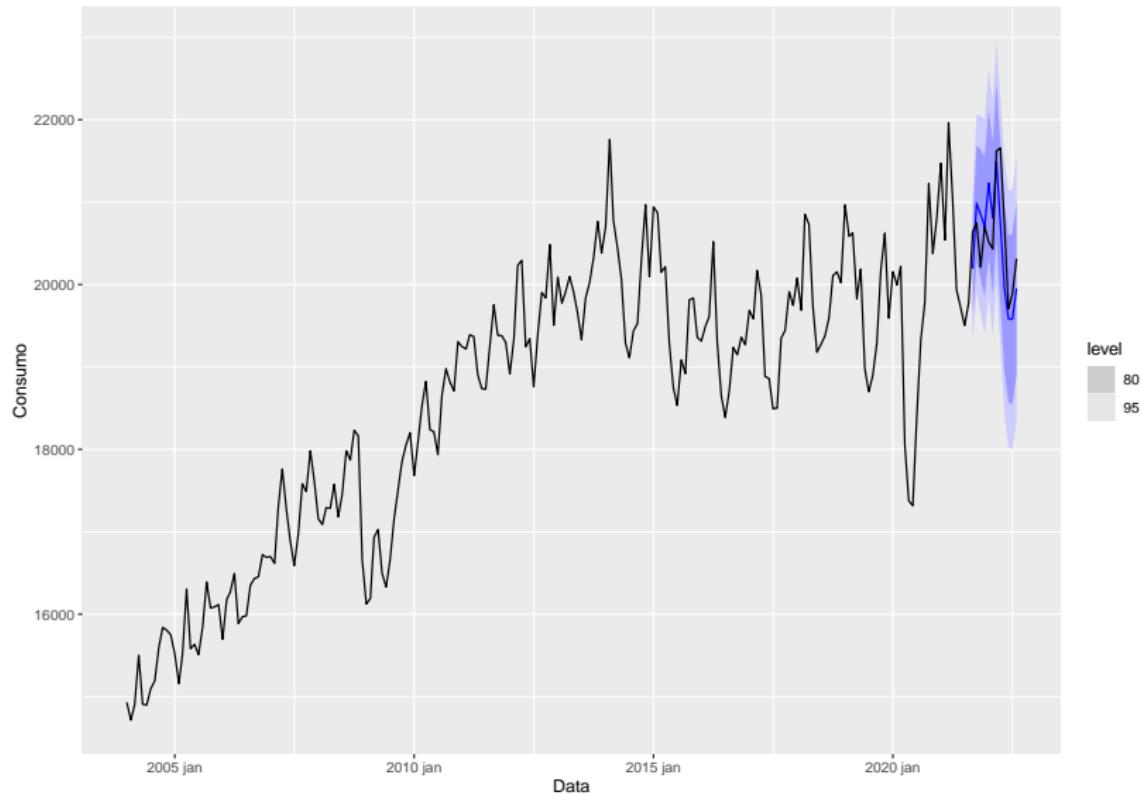
# Realizando a previsão e plotando



## Modificando o modelo ajustado “na mão”

```
consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
model(arima = ARIMA(Consumo ~ pdq(1,1,2)+PDQ(1,1,1))) %>%
forecast(h = 12) %>%
autoplot(consumo_tsibble)
```

## Modificando o modelo ajustado “na mão”



## Comparando a acurácia dos modelos ajustados

```
consumo_previsees = consumo_tsibble %>%
  filter(Data <= yearmonth("2021-08-01"),
         Regiao == "Sudeste") %>%
model(ets = ETS(Consumo),
      arima = ARIMA(Consumo),
      seasonal_naive = SNAIVE(Consumo)) %>%
forecast(h = 12)

accuracy(consumo_previsees, consumo_tsibble)
```

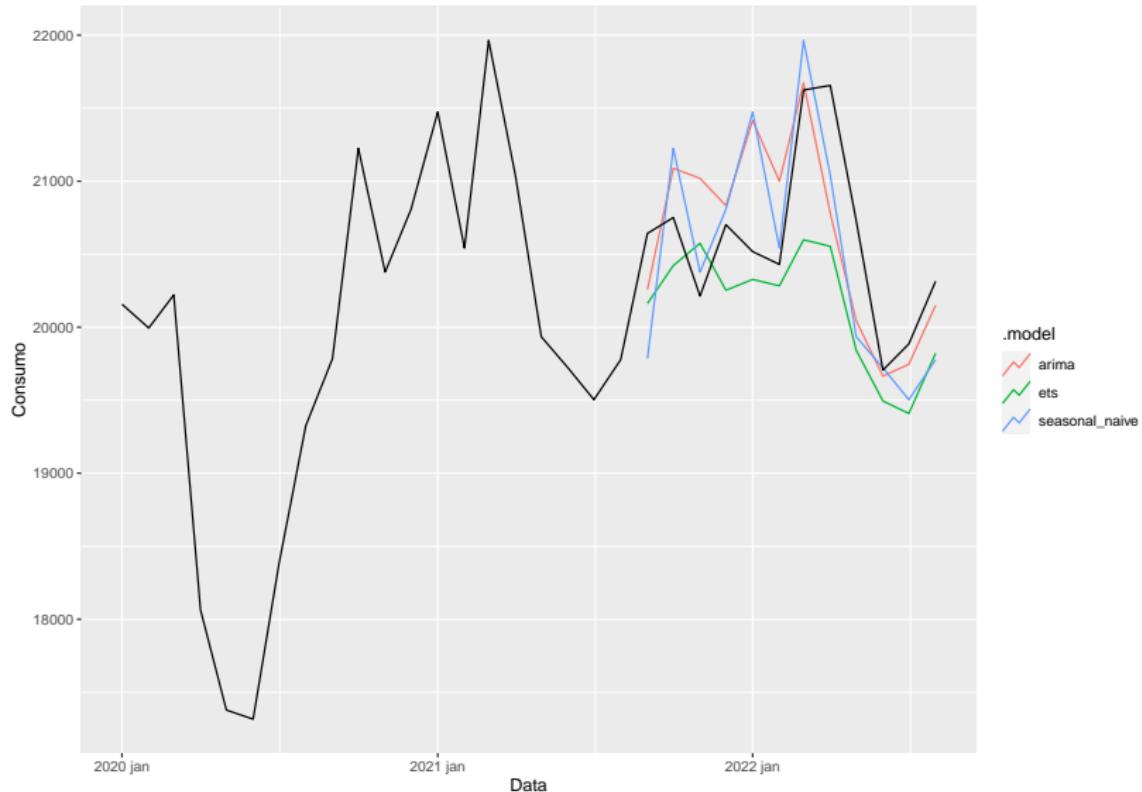
# Comparando a acurácia dos modelos ajustados

```
## # A tibble: 3 x 11
##   .model      Regiao .type     ME   RMSE    MAE     MPE    MAPE    MASE RMSSE   ACF1
##   <chr>      <chr>  <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>
## 1 arima       Sudeste Test  -41.5  528.  424. -0.224  2.05  0.624  0.617  0.443
## 2 ets         Sudeste Test   453.  599.  513.  2.16   2.46  0.755  0.699  0.300
## 3 seasonal_naive Sudeste Test   85.2  539.  446.  0.412  2.15  0.657  0.630  0.126
```

## Visualizando as previsões dos modelos ajustados

```
consumo_previsees %>%
  autoplot(consumo_tsibble %>%
            filter(year(Data) >= 2020) ,
           level = NULL)
```

# Visualizando as previsões dos modelos ajustados



## Não pare por aqui!

- ▶ O pacote `{fable}` tem mais a oferecer, como Regressão Dinâmica e Modelos Hierárquicos
- ▶ Além disso, a parte exploratória, features etc. é gigantesca, explore!

# Inscrições abertas para Mestrado e Doutorado na PUC-Rio

## PROCESSO SELETIVO 2023.1

DEI  
DEPARTAMENTO  
DE ENGENHARIA  
INDUSTRIAL



MESTRADO & DOUTORADO  
ENG. DE PRODUÇÃO

INSCRIÇÕES

ABERTAS!

ATÉ

13/11



Mais informações em:  
<http://www.ind.puc-rio.br/ensino/pos-graduacao-academica/inicio/>

Para nos conhecer melhor acesse:  
<https://vimeo.com/537856245>

{OPERAÇÕES E NEGÓCIOS  
EM ENGENHARIA}

{PESQUISA OPERACIONAL}