



## Previsão de Séries Temporais com o {fable}

Paula Maçaira

SBPO 2023 - São José dos Campos/SP



# Paula Maçaira

Professora de Estatística

<b>Atuação</b>	<b>Conquistas</b>
Métodos de <i>Statistical Learning</i>	Bolsista de Produtividade em Pesquisa (CNPq)
Programação Estatística (R/Python)	Jovem Cientista do Nosso Estado
Séries Temporais	Jovem Cientista Mulher (FAPERJ)
Previsão ( <i>forecasting</i> )	Melhor Tese em Engenharia de Produção 2019 (ABREPO)

# TRAJETÓRIA



2008-2012

2013-2015

2015-2018

2019-Presente

## Graduação

Estatística  
ENCE

## Mestrado

Engenharia Elétrica  
Métodos de Apoio à Decisão  
PUC-Rio

## Doutorado

Engenharia de Produção  
Gerência da Produção  
PUC-Rio

## Prof. Universitária

Quadro Principal  
Engenharia Industrial  
PUC-Rio



paulamacaira@puc-rio.br



github.com/paulamacaira



profpaulamacaira

## 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...

Dia 1:

1. o que é um tsibble
2. alguns gráficos de ST
3. benchmarks para previsão

Dia 2:

4. análise dos resíduos
5. medidas de acurácia
6. métodos de Suavização Exponencial
7. modelos de Box & Jenkins



Let's Go

Dia 1

## 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>    <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
## # i 15,140 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.
## # i 24,310 more rows
```

## Convertendo um data frame/tibble em tsibble

```
consumo = readr::read_csv2("https://raw.githubusercontent.com/paulamacaira/SBPO2023_Minicurso_fable/main/consumo_energia_2022completo.csv")
consumo

## # A tibble: 228 x 6
##   Data     Norte Nordeste Sudeste   Sul `Centro-Oeste`
##   <chr>    <dbl>   <dbl>   <dbl> <dbl>   <dbl>
## 1 01/01/2004 1574.   4457.   14930.  4510.   1507.
## 2 01/02/2004 1507.   4140.   14715.  4690.   1456.
## 3 01/03/2004 1574.   4373.   14904.  4899.   1535.
## 4 01/04/2004 1599.   4462.   15501.  4873.   1611.
## 5 01/05/2004 1625.   4424.   14907.  4723.   1566.
## 6 01/06/2004 1641.   4409.   14899.  4557.   1577.
## 7 01/07/2004 1665.   4376.   15094.  4599.   1612.
## 8 01/08/2004 1730.   4439.   15190.  4653.   1646.
## 9 01/09/2004 1687.   4570.   15594.  4654.   1727.
## 10 01/10/2004 1724.   4658.   15840.  4591.   1743.
## # i 218 more rows
```

O link do arquivo no GitHub é:

"[https://raw.githubusercontent.com/paulamacaira/SBPO2023\\_Minicurso\\_fable/main/consumo\\_energia\\_2022completo.csv](https://raw.githubusercontent.com/paulamacaira/SBPO2023_Minicurso_fable/main/consumo_energia_2022completo.csv)"

## Convertendo para tsibble

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

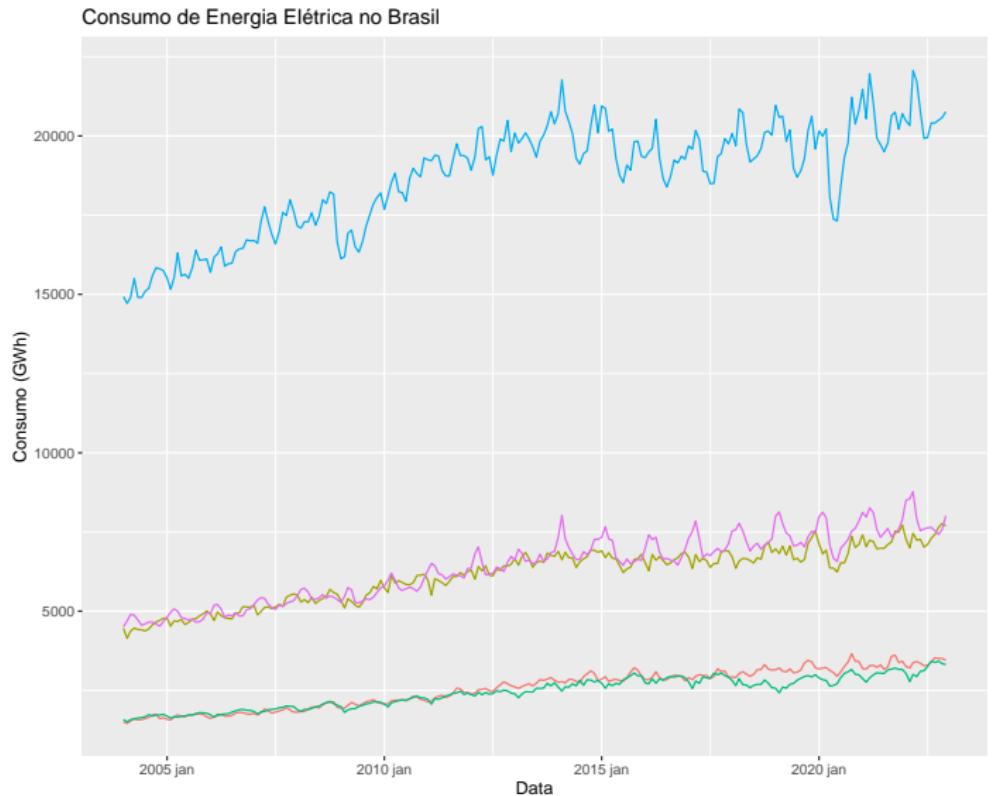
## Convertendo para tsibble

```
## # A tsibble: 1,140 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.
## # i 1,130 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

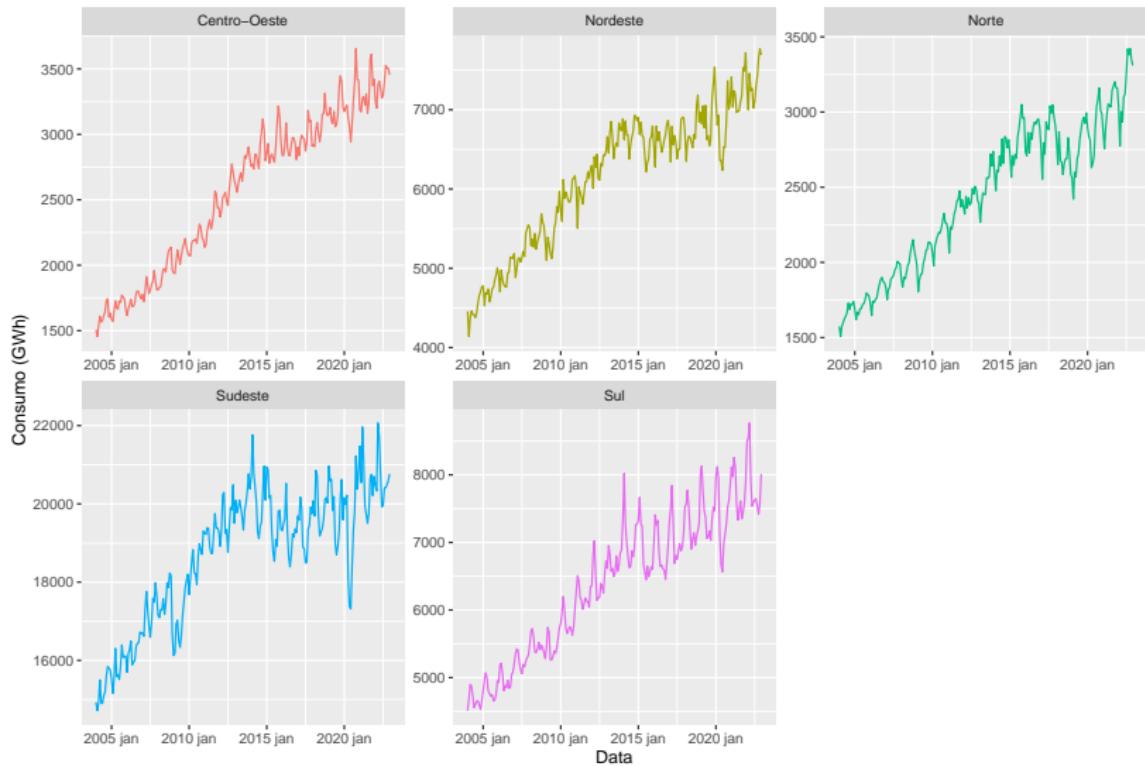


# Alguns gráficos de Séries Temporais

```
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  $t - k$

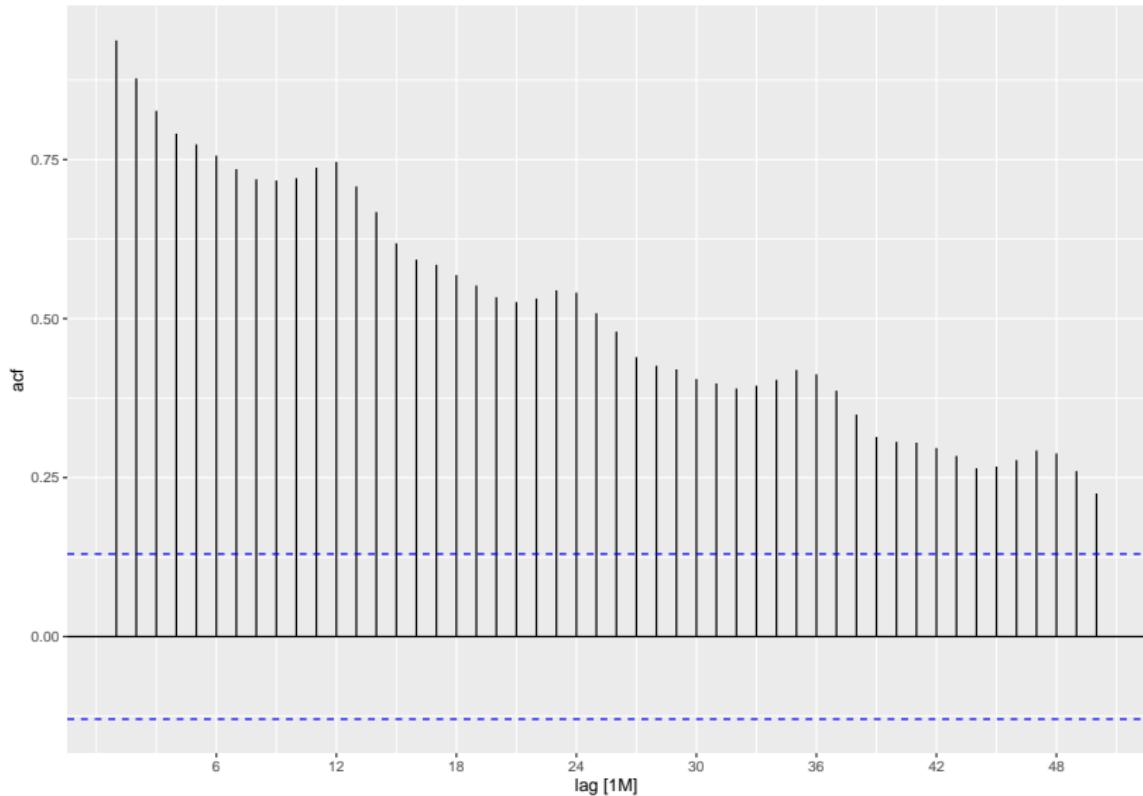
```
consumo_tsibble %>%  
  filter(Regiao == "Sudeste") %>%  
  ACF(Consumo, lag_max = 50)
```

```
## # A tsibble: 50 x 3 [1M]  
## # Key:      Regiao [1]  
##   Regiao     lag    acf  
##   <chr>    <cf_lag> <dbl>  
## 1 Sudeste    1M 0.937  
## 2 Sudeste    2M 0.878  
## 3 Sudeste    3M 0.827  
## 4 Sudeste    4M 0.791  
## 5 Sudeste    5M 0.774  
## 6 Sudeste    6M 0.756  
## 7 Sudeste    7M 0.735  
## 8 Sudeste    8M 0.719  
## 9 Sudeste    9M 0.717  
## 10 Sudeste   10M 0.721  
## # i 40 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

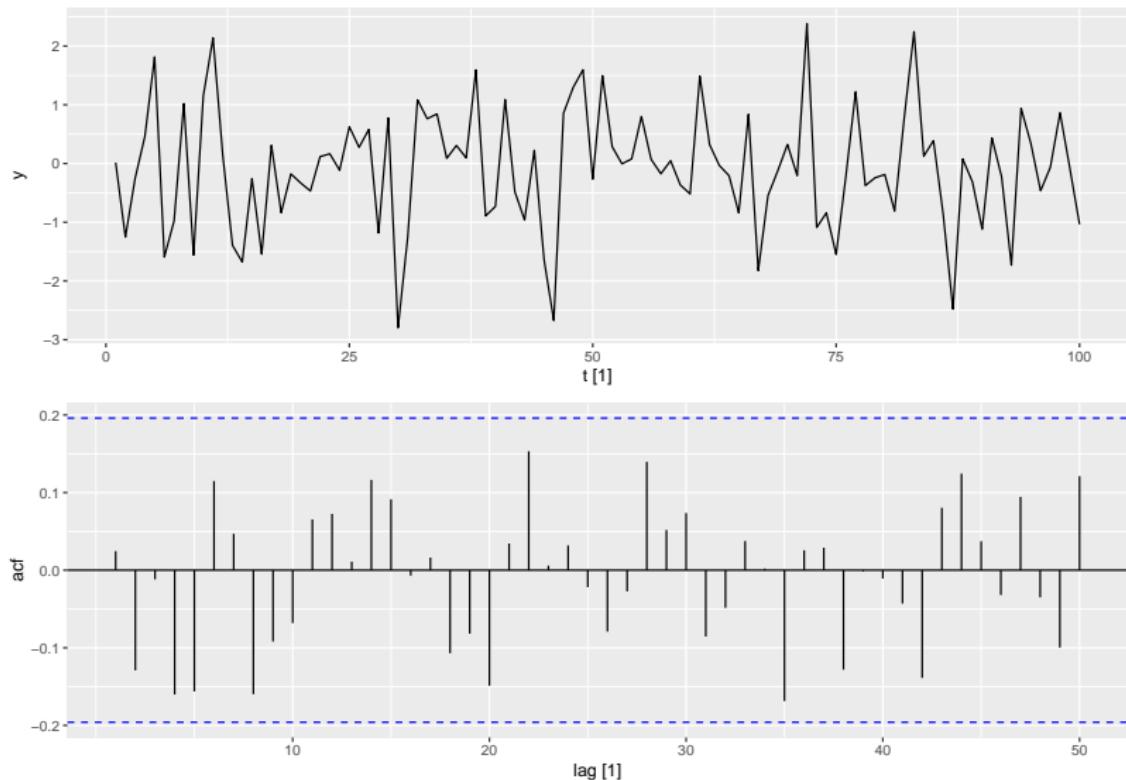
## Ruído Branco

- ▶ 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

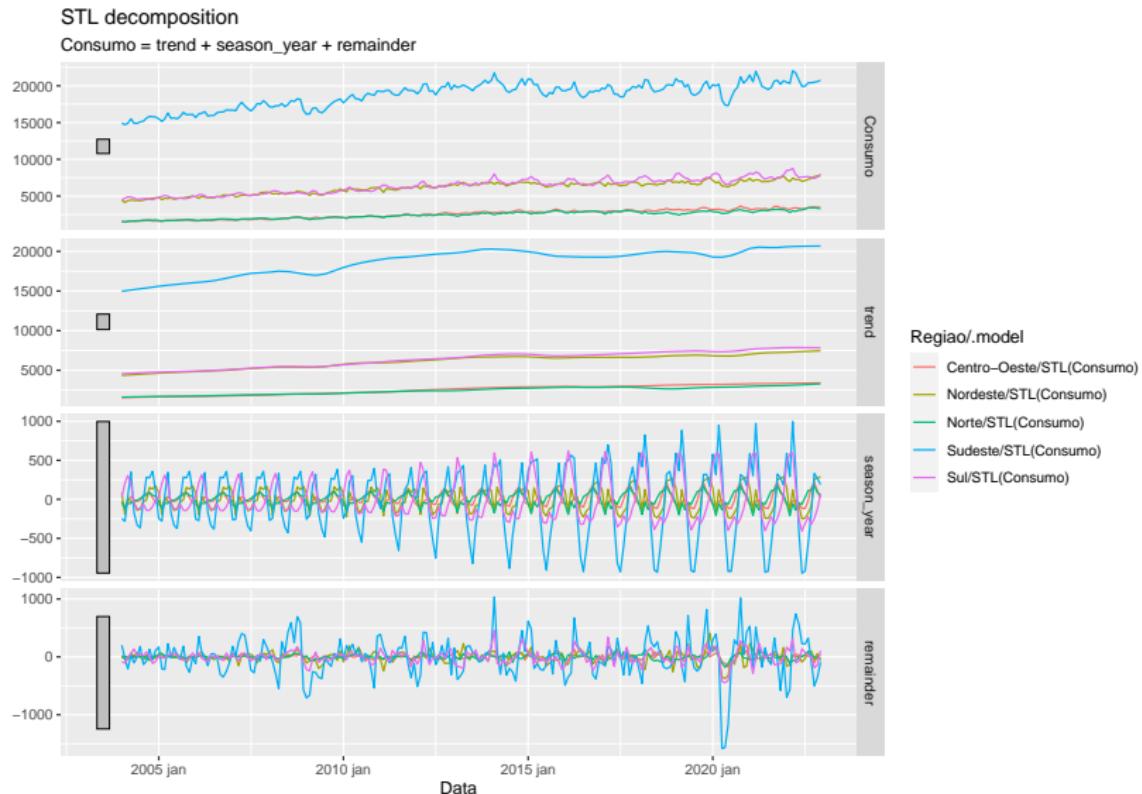
```
consumo_tsibble %>%  
  model(STL(Consumo)) %>%  
  components()
```

```
## # A dable: 1,140 x 8 [1M]  
## # Key:      Regiao, .model [5]  
## # :  
## Consumo = trend + season_year + remainder  
##   Regiao      .model     Data Consumo trend season_year remainder season_adjust  
##   <chr>       <chr>    <mth>  <dbl>  <dbl>      <dbl>    <dbl>      <dbl>  
## 1 Centro-Oes~ STL(C- 2004 jan  1507. 1552.    -68.8    23.6    1576.  
## 2 Centro-Oes~ STL(C- 2004 fev  1456. 1561.    -84.3    -20.8    1540.  
## 3 Centro-Oes~ STL(C- 2004 mar  1535. 1570.    -27.3    -7.89    1562.  
## 4 Centro-Oes~ STL(C- 2004 abr  1611. 1579.    36.2     -4.43    1574.  
## 5 Centro-Oes~ STL(C- 2004 mai  1566. 1588.   -0.888   -20.9    1567.  
## 6 Centro-Oes~ STL(C- 2004 jun  1577. 1596.    -27.8     8.75    1605.  
## 7 Centro-Oes~ STL(C- 2004 jul  1612. 1605.    -10.9     17.7    1623.  
## 8 Centro-Oes~ STL(C- 2004 ago  1646. 1614.    34.2     -2.18    1611.  
## 9 Centro-Oes~ STL(C- 2004 set  1727. 1622.    81.0     23.7    1646.  
## 10 Centro-Oes~ STL(C- 2004 out 1743. 1631.    88.9     23.4    1655.  
## # i 1,130 more rows
```

# Decomposição STL

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

# Decomposição STL



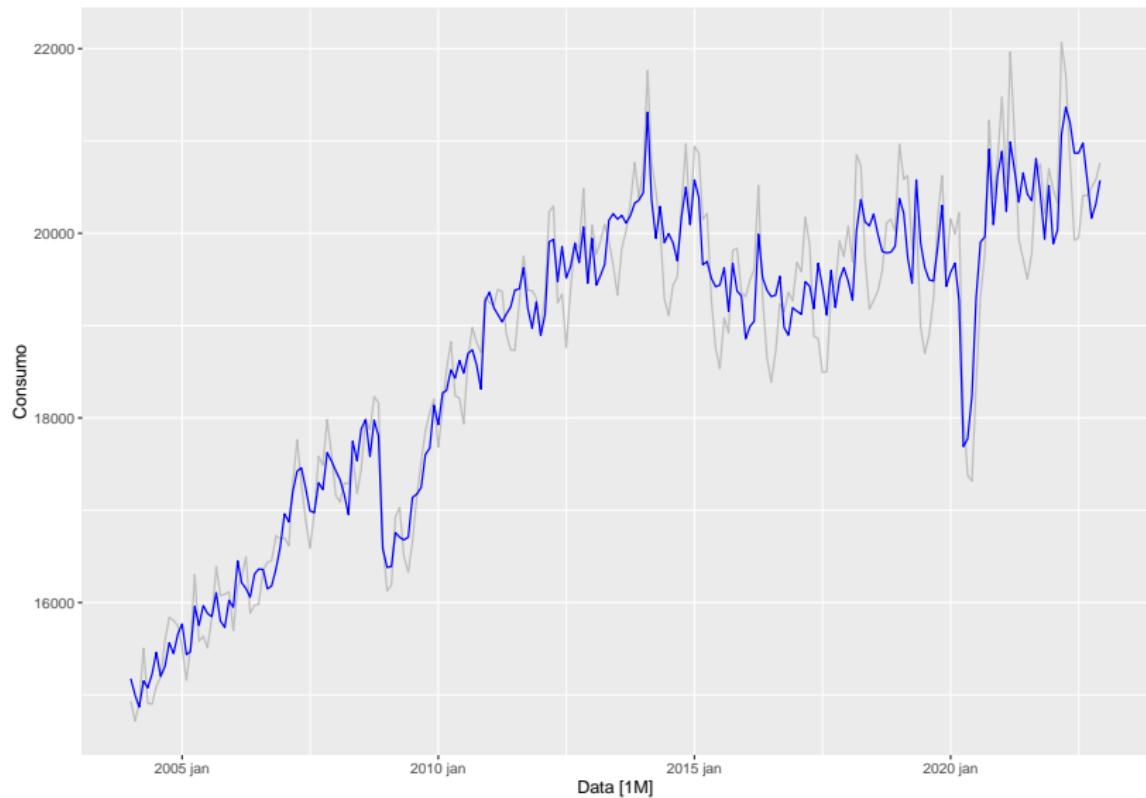
# Ajuste Sazonal

- ▶ 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 = "grey") +
  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 %>%
```

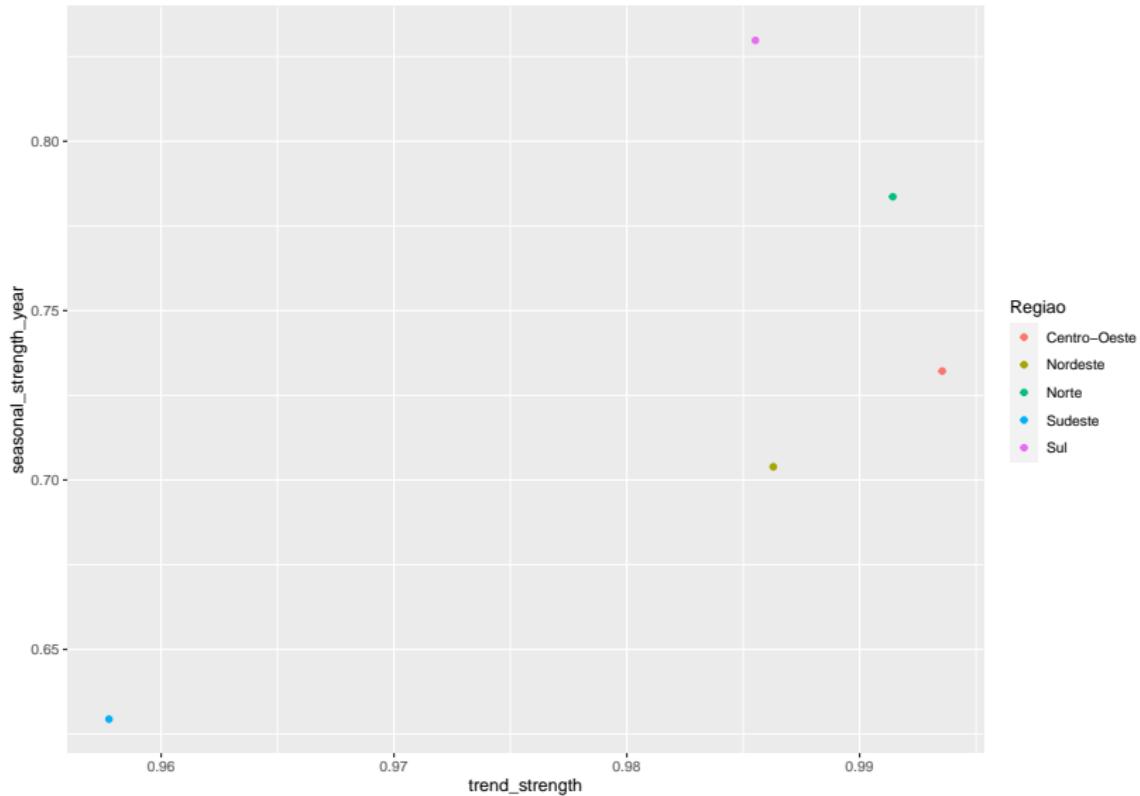
```
  features(Consumo, feat_stl)
```

```
## # A tibble: 5 x 10
##   Regiao      trend_strength seasonal_strength_year seasonal_peak_year
##   <chr>          <dbl>                  <dbl>                  <dbl>
## 1 Centro-Oeste    0.994                 0.732                  10
## 2 Nordeste        0.986                 0.704                   0
## 3 Norte           0.991                 0.784                  10
## 4 Sudeste         0.958                 0.629                   3
## 5 Sul             0.986                 0.830                   2
## # i 6 more variables: seasonal_trough_year <dbl>, spikiness <dbl>,
## #   linearity <dbl>, curvature <dbl>, stl_e_acf1 <dbl>, stl_e_acf10 <dbl>
```

## 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()

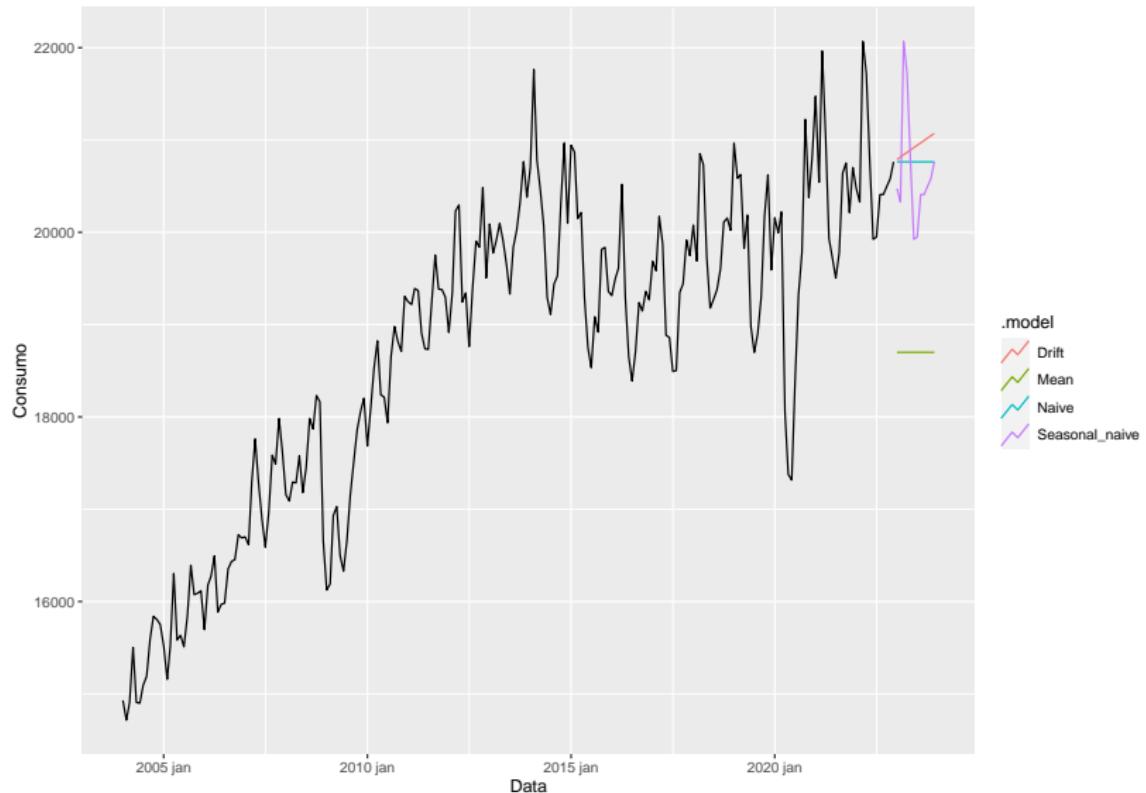
```
## # A fable: 48 x 5 [1M]
## # Key:     Regiao, .model [4]
##   Regiao  .model      Data       Consumo  .mean
##   <chr>    <chr>      <mth>      <dist>   <dbl>
## 1 Sudeste Seasonal_naive 2023 jan N(20474, 7e+05) 20474.
## 2 Sudeste Seasonal_naive 2023 fev N(20328, 7e+05) 20328.
## 3 Sudeste Seasonal_naive 2023 mar N(22068, 7e+05) 22068.
## 4 Sudeste Seasonal_naive 2023 abr N(21718, 7e+05) 21718.
## 5 Sudeste Seasonal_naive 2023 mai N(20789, 7e+05) 20789.
## 6 Sudeste Seasonal_naive 2023 jun N(19925, 7e+05) 19925.
## 7 Sudeste Seasonal_naive 2023 jul N(19949, 7e+05) 19949.
## 8 Sudeste Seasonal_naive 2023 ago N(20409, 7e+05) 20409.
## 9 Sudeste Seasonal_naive 2023 set N(20410, 7e+05) 20410.
## 10 Sudeste Seasonal_naive 2023 out N(20496, 7e+05) 20496.
## # i 38 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



Dia 2

## 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

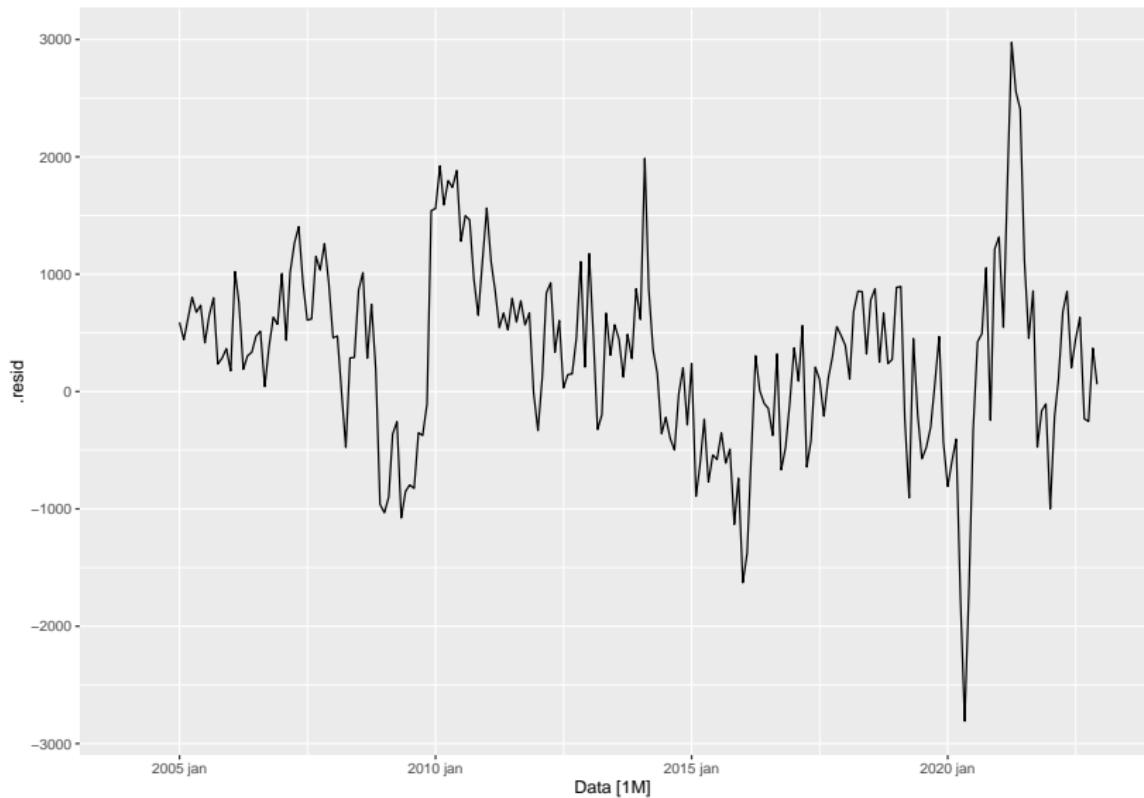
```
augment(consumo_fit)
```

```
## # A tsibble: 912 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
## # i 902 more rows
```

# Resíduos

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

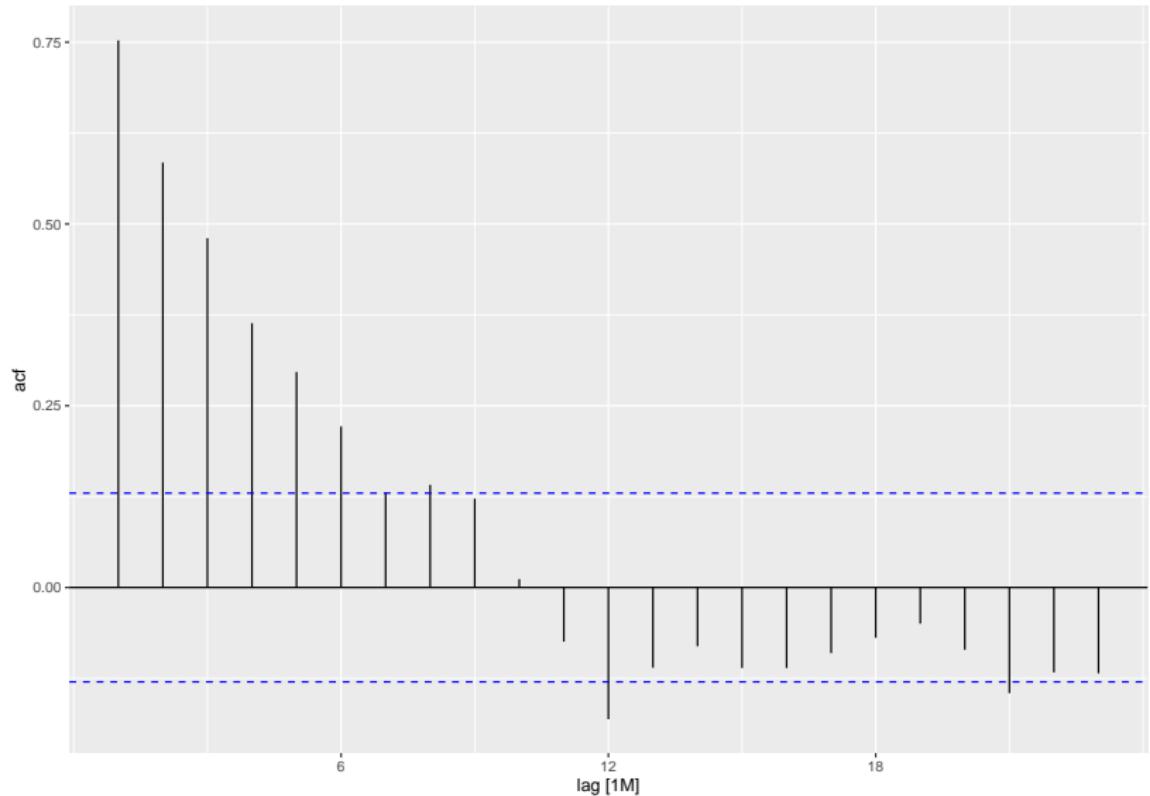
# Resíduos



## 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       0.356     0.551
## 2 Sudeste Mean        203.      0
## 3 Sudeste Naive       0.356     0.551
## 4 Sudeste Seasonal_naive 124.     0
```

## Medidas de acurácia

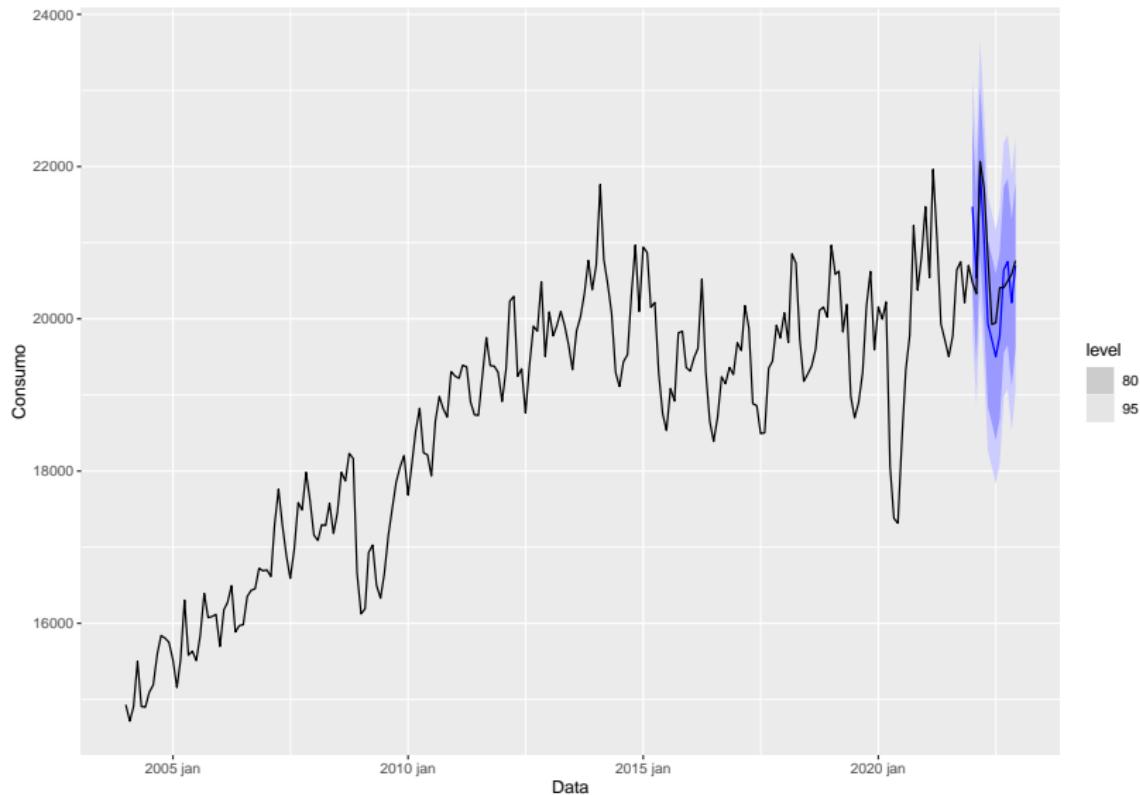
- ▶ Primeiro vamos selecionar apenas a região Sudeste e o modelo “Seasonal\_naive” para que a visualização fique mais fácil

```
consumo_fit = consumo_tsibble %>%
  filter(Data <= yearmonth("2021-12-01"),
         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   137.  512.  421.  0.652  2.04  0.624  0.602  0.293
```

# 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("2021-12-01"),
         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,A,M)>
```

## Olhando o modelo ajustado com mais detalhes

```
report(consumo_ets)
```

```
## Series: Consumo
## Model: ETS(M,A,M)
##   Smoothing parameters:
##     alpha = 0.7700755
##     beta  = 0.0007507238
##     gamma = 0.1620534
##
##   Initial states:
##     l[0]      b[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]
## 14845.59 55.98986 1.021179 1.029193 1.021805 1.014869 0.9916968 0.9693771
##     s[-6]      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
## 0.9663719 0.9772143 1.01423 0.9951997 0.9940676 1.004797
##
##   sigma^2:  6e-04
##
##          AIC      AICc      BIC
## 3803.779 3806.870 3861.159
```

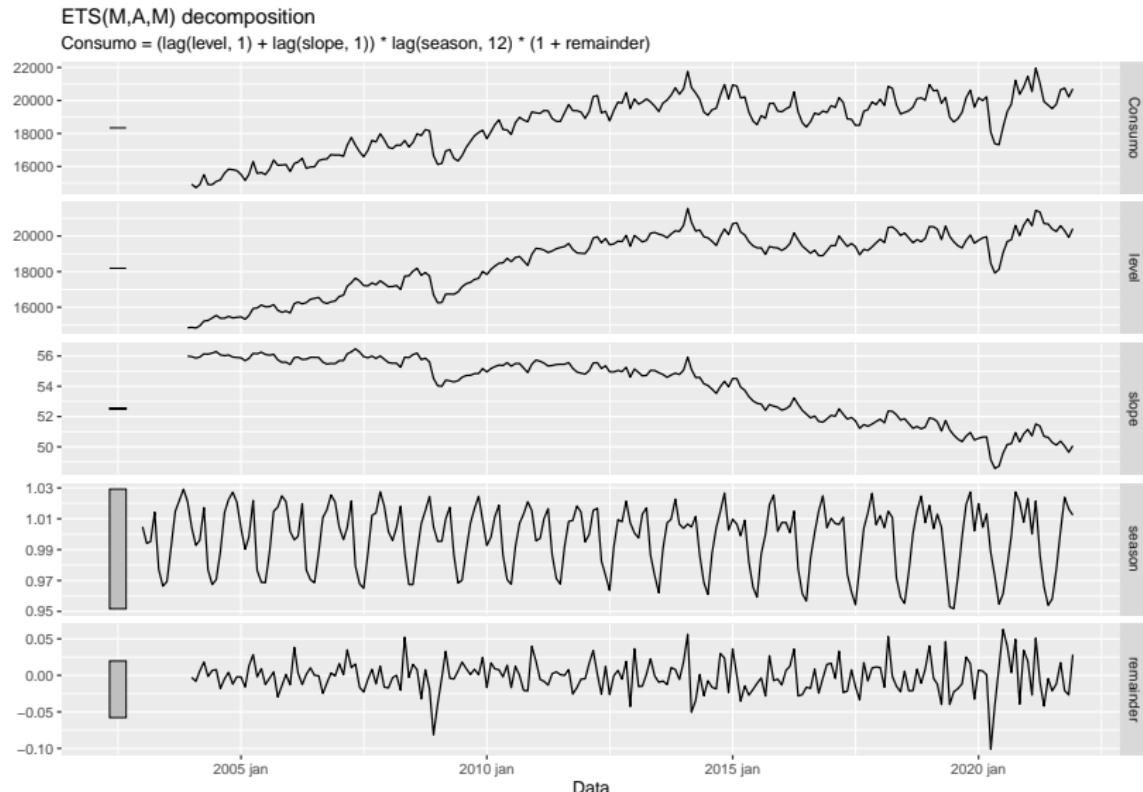
# Observando as componentes do modelo ajustado

```
components(consumo_ets)
```

```
## # A dable: 228 x 8 [1M]
## # Key:      Regiao, .model [1]
## # :          Consumo = (lag(level, 1) + 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  1.00      NA
## 2 Sudeste ets  2003 fev      NA     NA     NA  0.994     NA
## 3 Sudeste ets  2003 mar      NA     NA     NA  0.995     NA
## 4 Sudeste ets  2003 abr      NA     NA     NA  1.01      NA
## 5 Sudeste ets  2003 mai      NA     NA     NA  0.977     NA
## 6 Sudeste ets  2003 jun      NA     NA     NA  0.966     NA
## 7 Sudeste ets  2003 jul      NA     NA     NA  0.969     NA
## 8 Sudeste ets  2003 ago      NA     NA     NA  0.992     NA
## 9 Sudeste ets  2003 set      NA     NA     NA  1.01      NA
## 10 Sudeste ets 2003 out     NA     NA     NA  1.02      NA
## # i 218 more rows
```

# Plotando as componentes

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

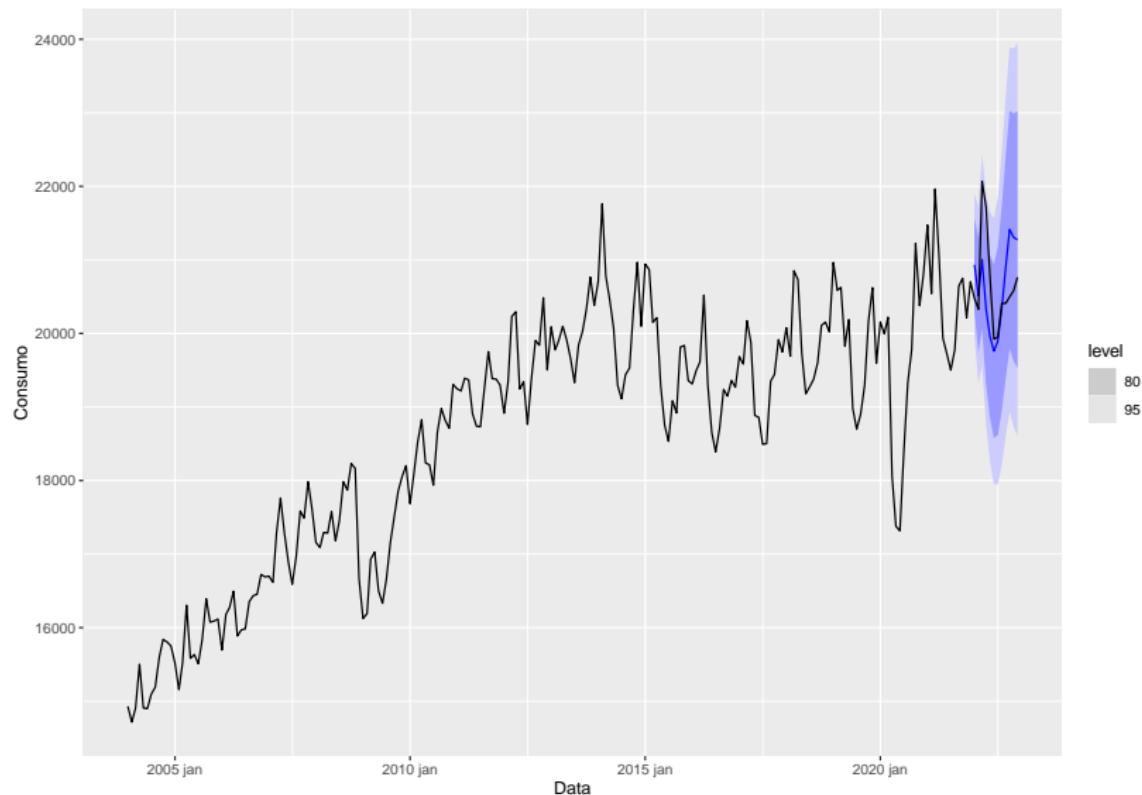


## Realizando a previsão e plotando

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

## Simulação via Bootstrap

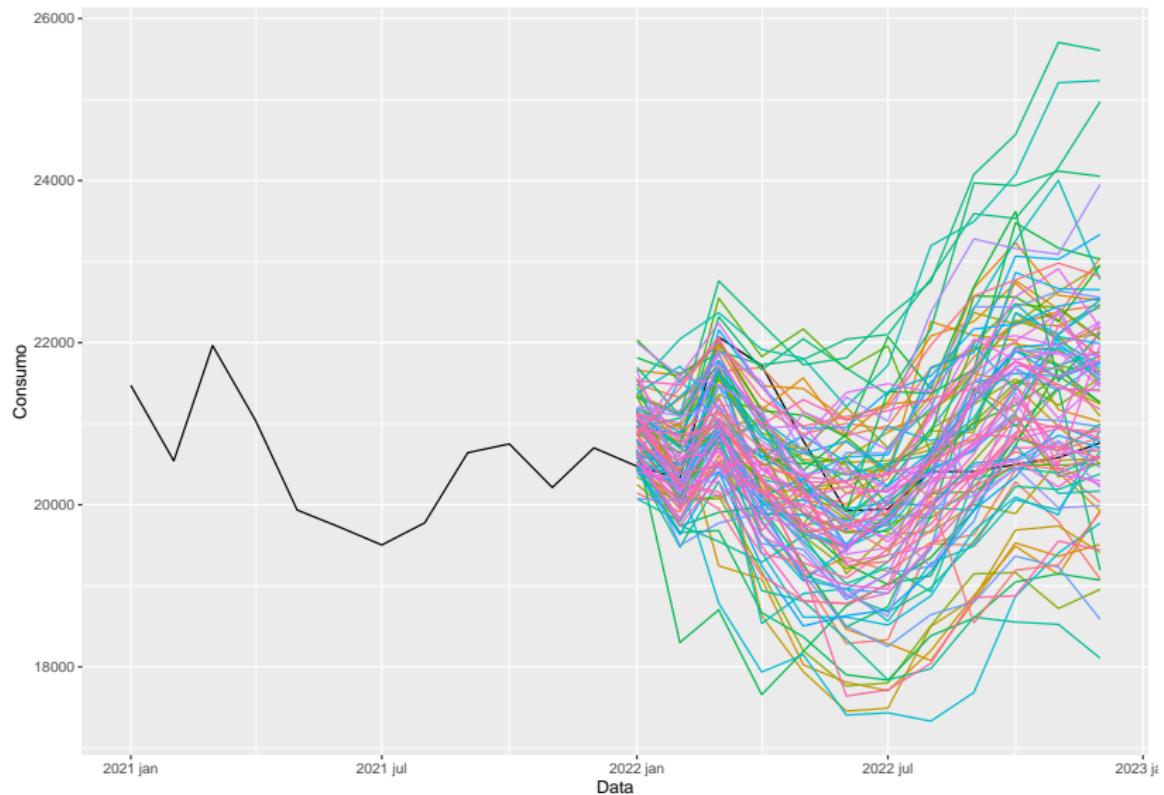
```
sim = consumo_ets %>%
  generate(h = 12, times = 100, bootstrap = TRUE)
sim

## # A tsibble: 1,200 x 6 [1M]
## # Key:      Regiao, .model, .rep [100]
##   Regiao  .model .rep      Data    .innov    .sim
##   <chr>   <chr>  <chr>    <mth>    <dbl>    <dbl>
## 1 Sudeste ets     1 2022 jan -0.0378  20144.
## 2 Sudeste ets     1 2022 fev -0.000441 19911.
## 3 Sudeste ets     1 2022 mar  0.0183  20760.
## 4 Sudeste ets     1 2022 abr -0.00399 19935.
## 5 Sudeste ets     1 2022 mai -0.0253  19095.
## 6 Sudeste ets     1 2022 jun -0.0383  18288.
## 7 Sudeste ets     1 2022 jul -0.0132  18336.
## 8 Sudeste ets     1 2022 ago  0.0379  19522.
## 9 Sudeste ets     1 2022 set -0.0200  19500.
## 10 Sudeste ets    1 2022 out  0.0107  20283.
## # i 1,190 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("2021-12-01"),
         Regiao == "Sudeste") %>%
model(arima = ARIMA(Consumo))
```

## Olhando o modelo ajustado com mais detalhes

```
report(consumo_arima)

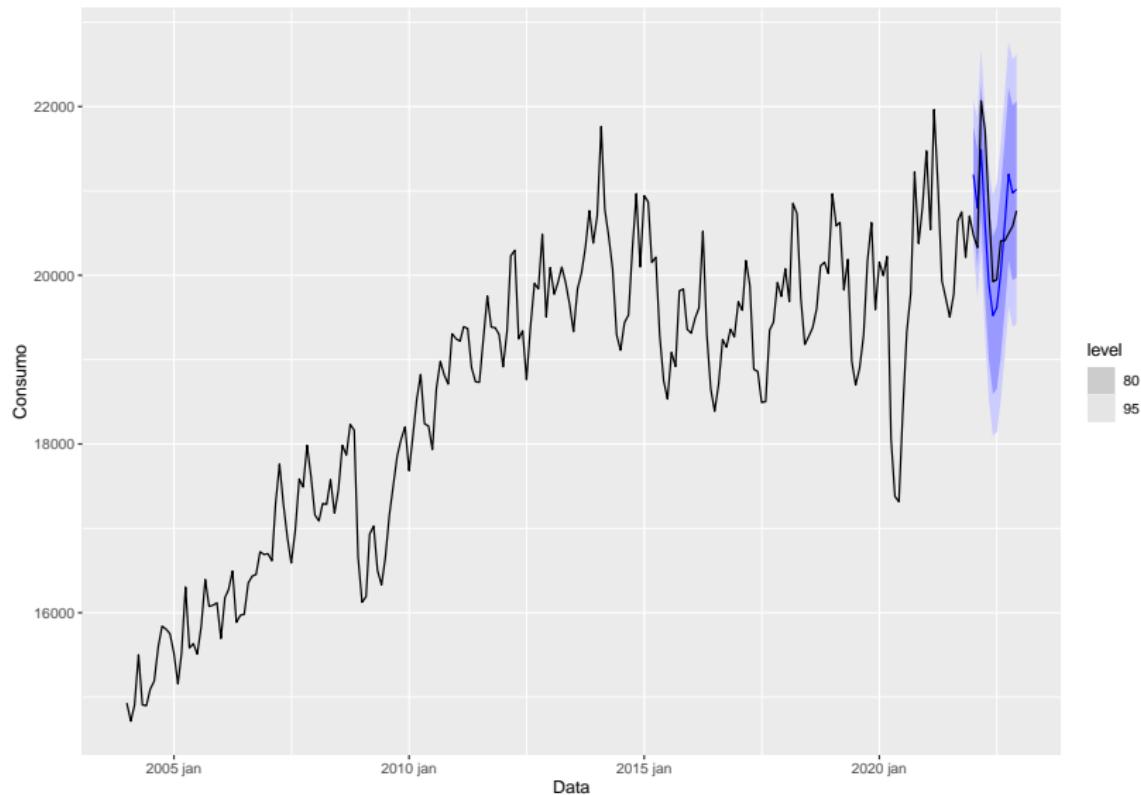
## Series: Consumo
## Model: ARIMA(1,0,1)(0,1,1)[12] w/ drift
##
## Coefficients:
##             ar1      ma1     sma1  constant
##             0.9063 -0.2207 -0.6654   26.6154
## s.e.    0.0374  0.0849  0.0677   8.7272
##
## sigma^2 estimated as 199873: log likelihood=-1536.33
## AIC=3082.67   AICc=3082.97   BIC=3099.26
```

## Realizando a previsão e plotando

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

consumo_arima_fc %>%
  autoplot(consumo_tsibble)
```

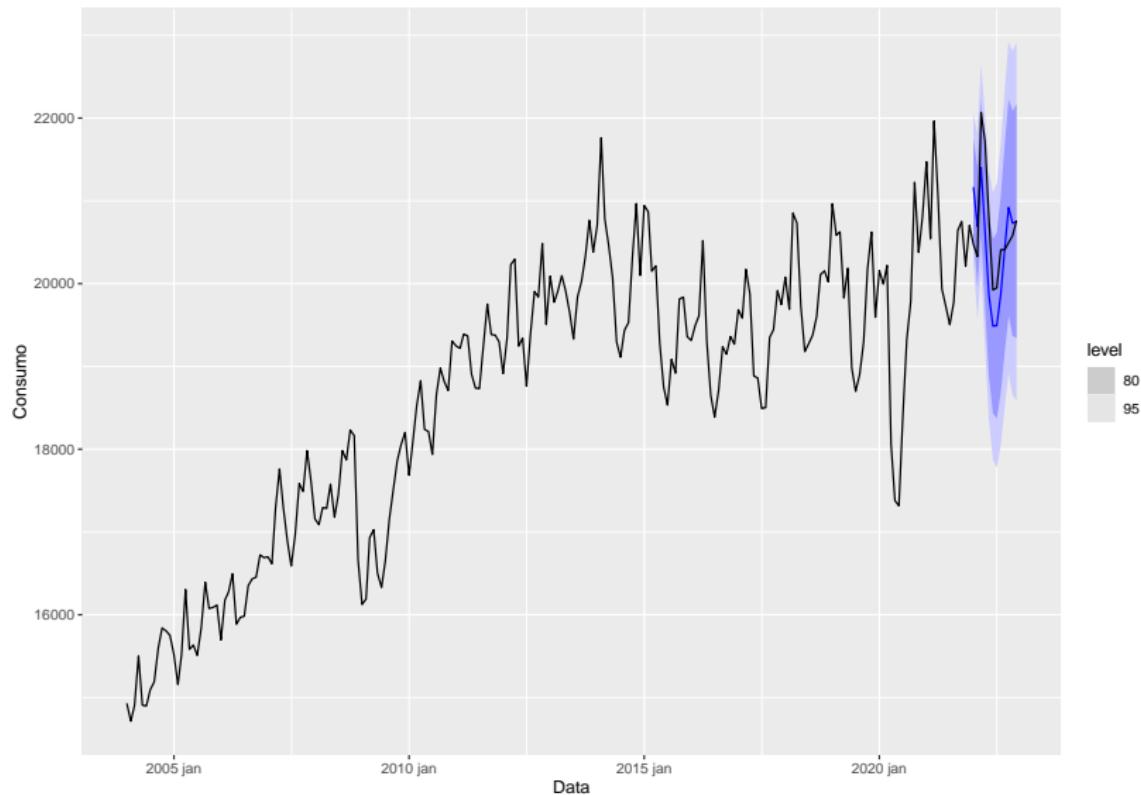
# Realizando a previsão e plotando



## Modificando o modelo ajustado “na mão”

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

accuracy(consumo_previsees, consumo_tsibble)
```

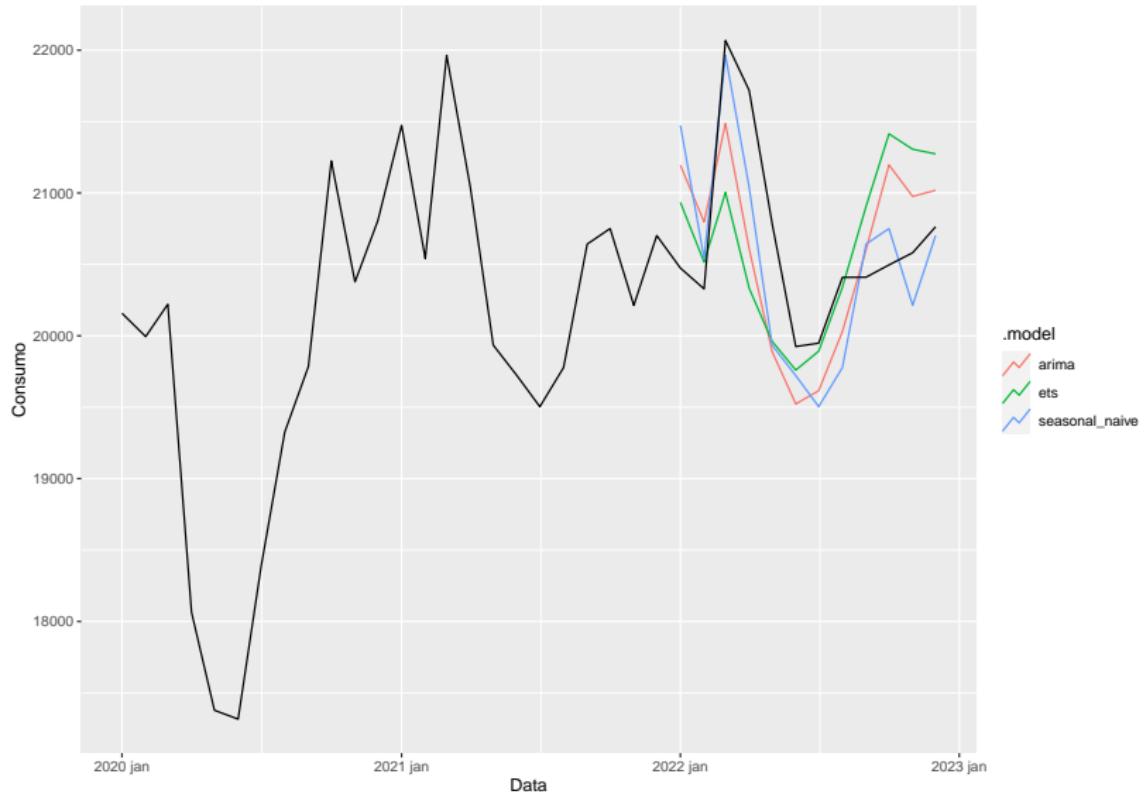
# Comparando a acurácia dos modelos ajustados

```
## # 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  80.9  597.  536.  0.356  2.58  0.796  0.702  0.620
## 2 ets         Sudeste Test  22.6  701.  572.  0.0479 2.72  0.849  0.824  0.694
## 3 seasonal_naive Sudeste Test 137.   512.  421.  0.652  2.04  0.624  0.602  0.293
```

## 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!

# PROCESSO SELETIVO

# SELETIVO

Mestrado e Doutorado

EM ENGENHARIA DE PRODUÇÃO



## ÁREAS DE CONCENTRAÇÃO

Operações e Negócios em Engenharia

Pesquisa Operacional

Gestão de Operações  
Gestão de Negócios

Algoritmos e Otimização  
Métodos estatísticos e Analytics

INSCRIÇÕES

12 NOV



Programa de Pós-Graduação em Engenharia de Produção

