R para Economia

Lucas Mendes

24/03/2020

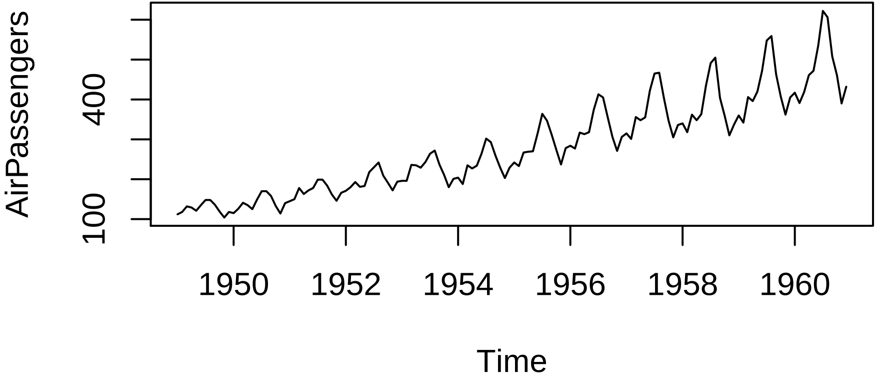
Séries Temporais

Séries Temporais

Na ultima aula trabalhamos com dados cross section, ou seja, dados que estavam na mesma unidade de tempo.

Agora trabalharemos com dados que variam de acordo com o periodo (Dia,Semana,Mês,Ano)

Séries de tempo



Padrões de uma serie de tempo

Tendência

A Tendência pode ser analisada como um crescimento ou decrescimento de longo prazo.

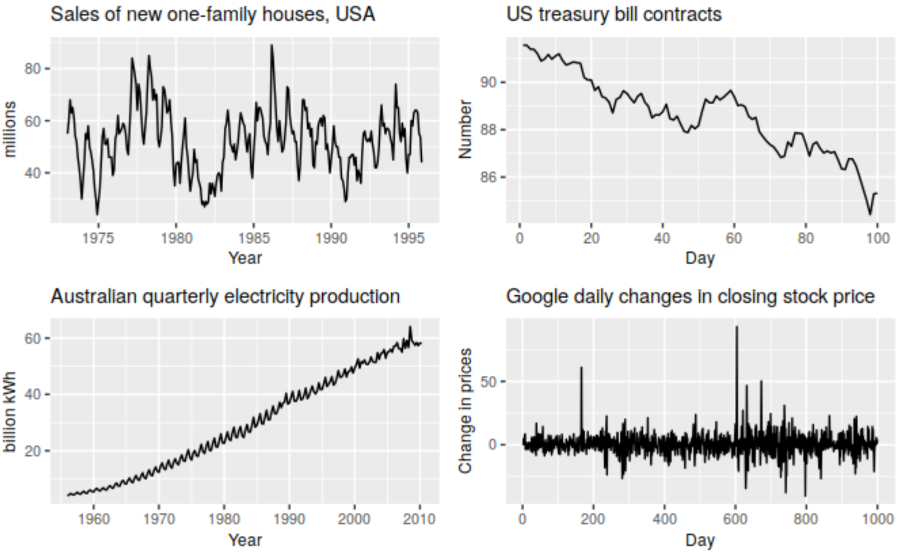
Sazonalidade

A Sazonalidade pode ser identificada como um padrão que ocorre frequentemente em algum periodo do tempo, como dia, mês ou ano.

Ciclos

Ciclos são tendencias de alta e queda que não possuem uma frequencia bem definida. Elas são costumeiramente geradas por causas econômicas ou chamados de ciclos de negócios.

Exemplos


Sales of new one-family houses, USA
millions
80
60
40
1975
1980
1985
1990
1995
Year
US treasury bill contracts
Number
90
88
86
0
20
40
60
80
100
Day
Australian quarterly electricity production
billion kWh
60
40
20
1960
1970
1980
1990
2000
2010
Year
Google daily changes in closing stock price
Change in prices
50
0
0
200
400
600
800
1000
Day

Decomposição de uma série temporal

Decomposição de uma série temporal

Uma série temporal tem dois tipos de decomposição, aditiva e multiplicativa.

Aditiva

$y_t = S_t + T_t + R_t$

Multiplicativa

$y_t = S_t x T_t x R_t$

Explicando

S_t = Componente Sazonal
T_t = Componente Ciclo_Tendencia
R_t = Componente Restante (aletório)

Decomposição de uma série temporal

Qual a diferença entre os dois?Além do óbvio

- Na série com decomposição aditiva, a magnitude da variação sazonal ou ciclo_tendencia não varia com o tempo

Decomposição de uma série temporal

Qual a diferença entre os dois?Além do óbvio

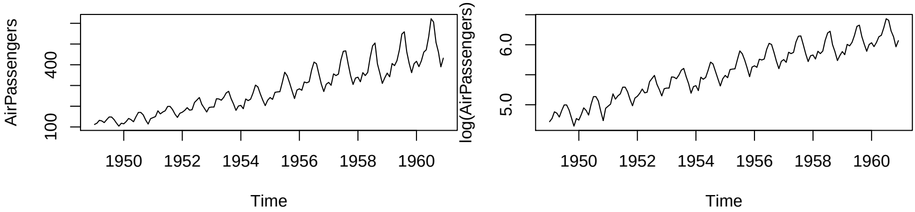
- Na série com decomposição aditiva, a magnitude da variação sazonal ou ciclo_tendencia não varia com o tempo
- Já o contrário ocorre na decomposição multiplicativa

Decomposição de uma série temporal

Qual a diferença entre os dois?Além do óbvio

- Na série com decomposição aditiva, a magnitude da variação sazonal ou ciclo_tendencia não varia com o tempo
- Já o contrário ocorre na decomposição multiplicativa
- Vendo graficamente

Decomposição de uma série temporal



Decomposição de uma série temporal

Lembrando que

$y_t = S_t x T_t x R_t = log(y_t) = log(S_t) + log(T_t) + log(R_t)$

Decomposição de uma série temporal

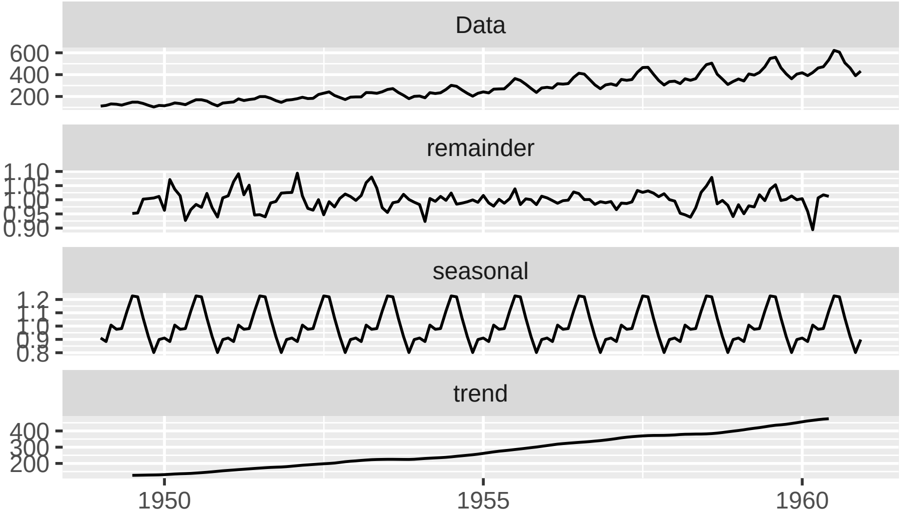
Decomposição clássica

Para demonstrar o método de decomposição clássica, podemos usar o código a seguir

```
library(ggfortify)
AirPassengers %>% # Base de dados
  decompose(type="multiplicative") %>% # Decomposição
  autoplot()  # Grafico
```

Decomposição de uma série temporal

Decomposição clássica



Simple exponential smoothing

Simple exponential smoothing

Esse método de previsão atribui um peso para as observações passadas para inferir o futuro.

Esse peso na qual chamamos de α varia entre 0 a 1 e decai exponencialmente com o n° de observações.

Ele é útil especialmente quando temos dados sem um tendência ou sazonalidade evidentes

$$y_{t+1|t} = \alpha y_t + \alpha(1-\alpha)y + \alpha(1-\alpha)^2 y + ...$$

Simple exponential smoothing

Representação por componentes

Forecast Equaton $y_{t+h|t} = l_t$

Smoothing equation $l_t = \alpha y_t + (1-\alpha) l_{t-1}$

Exemplo

Para usar o modelo, temos que recorrer a função `ses()` do pacote `forecast`

```
library(forecast)
prev <- AirPassengers %>%
  ses(h = 12 # Periodos de previsão
      )
```

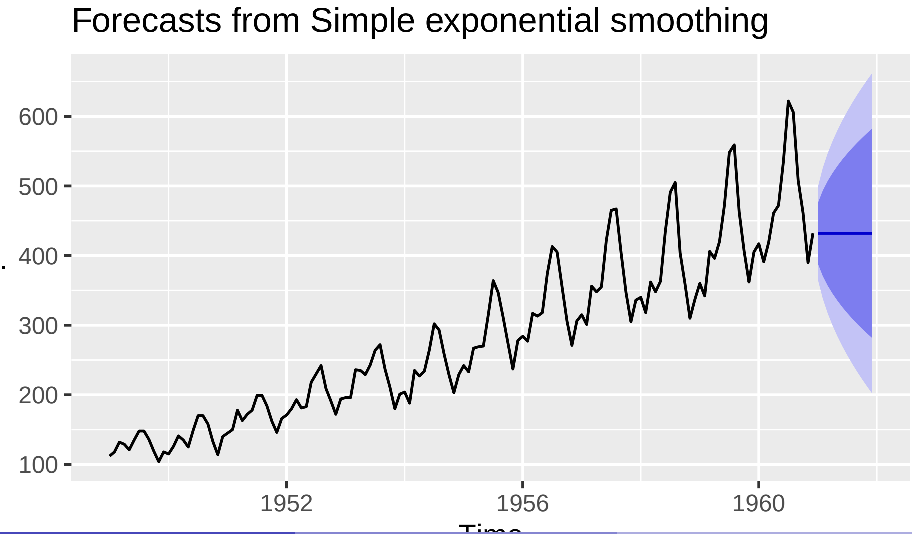
Exemplo

```r
prev
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1961       431.9958 388.6410 475.3506 365.6904 498.3012
## Feb 1961       431.9958 370.6859 493.3057 338.2304 525.7612
## Mar 1961       431.9958 356.9081 507.0835 317.1591 546.8325
## Apr 1961       431.9958 345.2927 518.6989 299.3949 564.5967
## May 1961       431.9958 335.0593 528.9323 283.7442 580.2474
## Jun 1961       431.9958 325.8075 538.1841 269.5948 594.3968
## Jul 1961       431.9958 317.2996 546.6920 256.5831 607.4085
## Aug 1961       431.9958 309.3806 554.6110 244.4721 619.5195
## Sep 1961       431.9958 301.9430 562.0486 233.0971 630.8945
## Oct 1961       431.9958 294.9082 569.0834 222.3384 641.6532
## Nov 1961       431.9958 288.2173 575.7743 212.1055 651.8861
## Dec 1961       431.9958 281.8241 582.1675 202.3281 661.6635
```

Plotando

```
autoplot(prev)
```



Holt's linear trend method

Holt's linear trend method

Esse método é uma extensão do antigo, possibilitando a previsão de séries com tendência.

Forecast Equation: $y_{t+h|t} = l_t + hb_t$

Level Equation: $l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + bt - 1)$

Trend Equation: $b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$

Podemos usar esse método usando a função `holt()` do pacote `forecast`

Holt's linear trend method

```r
prev <- holt(AirPassengers,h = 12)
```

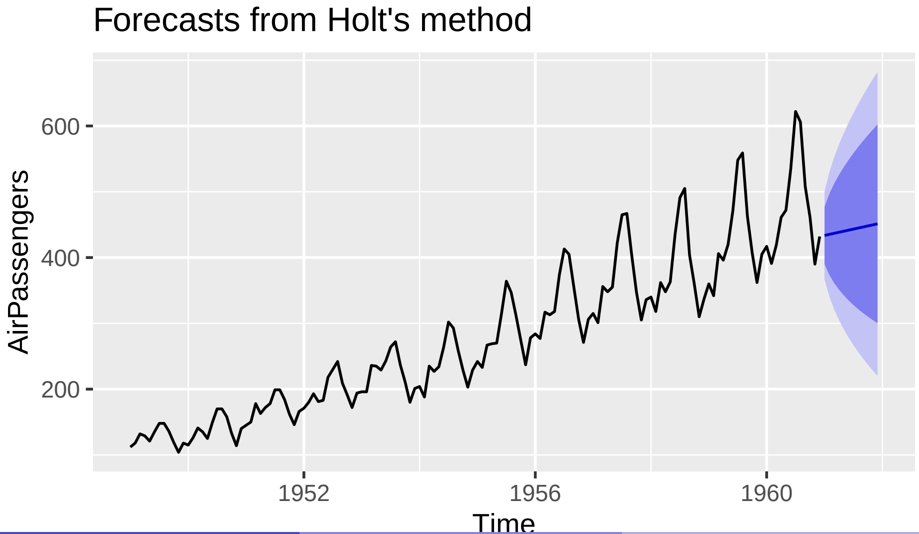
Holt's linear trend method

```
prev
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1961       433.6004 390.0116 477.1892 366.9370 500.2638
## Feb 1961       435.2049 373.5609 496.8488 340.9286 529.4811
## Mar 1961       436.8093 361.3087 512.3099 321.3411 552.2775
## Apr 1961       438.4138 351.2296 525.5980 305.0770 571.7505
## May 1961       440.0182 342.5389 537.4975 290.9365 589.0999
## Jun 1961       441.6227 334.8345 548.4109 278.3042 604.9411
## Jul 1961       443.2271 327.8772 558.5770 266.8146 619.6396
## Aug 1961       444.8316 321.5113 568.1518 256.2296 633.4335
## Sep 1961       446.4360 315.6288 577.2432 246.3837 646.4883
## Oct 1961       448.0405 310.1508 585.9301 237.1565 658.9244
## Nov 1961       449.6449 305.0179 594.2719 228.4571 670.8327
## Dec 1961       451.2494 300.1840 602.3147 220.2148 682.2839
```

Holt's linear trend method

```
autoplot(prev)
```



Holt-Winters' seasonal method

Holt-Winters' seasonal method

Uma segunda extensão do modelo, agora para podermos prever um modelo com sazonalidade.

Lembrando que temos duas especificações para sazonalidade

- Aditiva
- Multiplicativa

Para usar o modelo no R, chamamos a função `hw()` do pacote `forecast`

Holt-Winters' seasonal method additive

Forecast Equation $y_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)}$

Level Equation $l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$

Trend Equation $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$

Seasonal Equation $s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$

Holt-Winters' seasonal method multiplicative

Forecast Equation $y_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)}$

Level Equation $l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1})$

Trend Equation $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$

Seasonal Equation $s_t = \gamma\frac{y_t}{(l_{t-1}+b_{t-1})} + (1-\gamma)s_{t-m}$

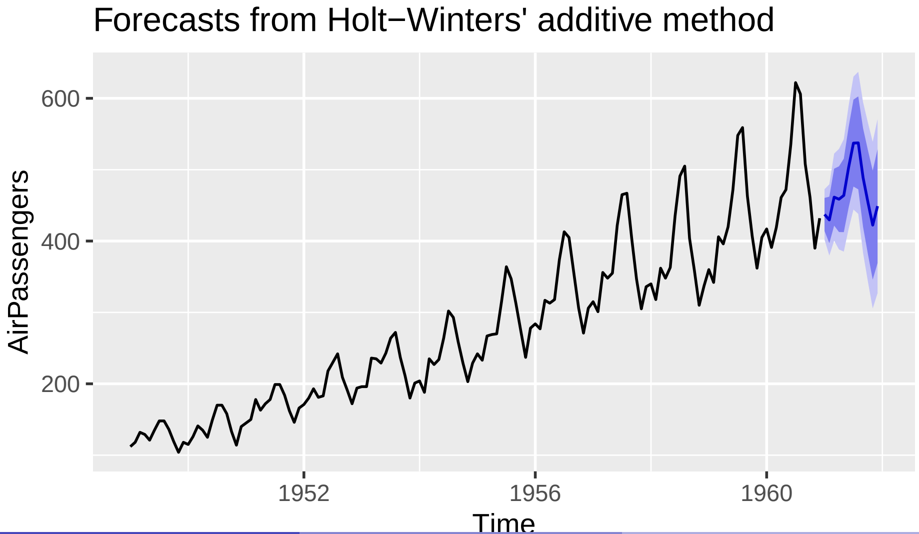
Holt-Winters' seasonal method

```r
prev_add <- hw(AirPassengers,
               seasonal = "additive",
               h = 12)

prev_mult <- hw(AirPassengers,
                seasonal = "multiplicative",
                h = 12)
```

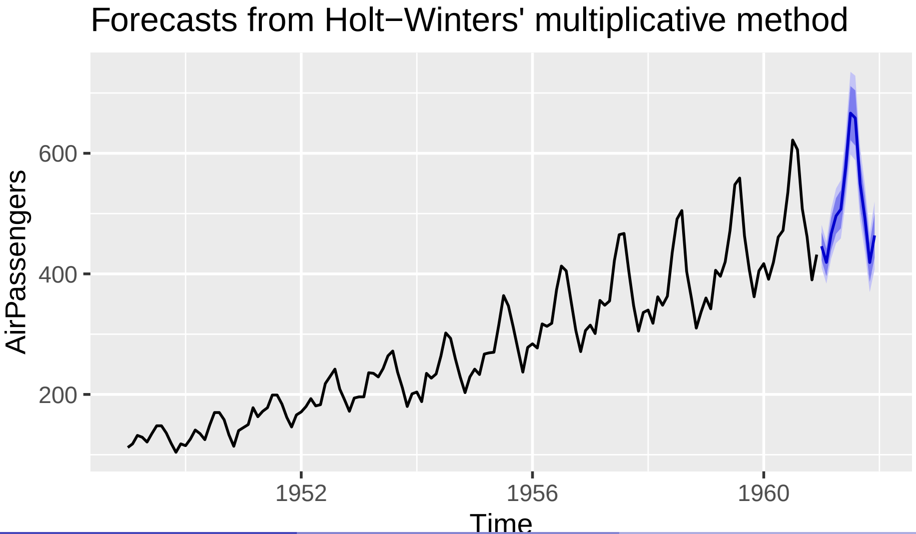
Holt-Winters' seasonal method

```
autoplot(prev_add)
```



Holt-Winters’ seasonal method

```
autoplot(prev_mult)
```



Exercicios

ARIMA

ARIMA

Modelos ARIMA providenciam outras aproximações para a previsão de séries temporais.

Enquanto os modelos de suavização exponencial baseavam - se na descrição de têndencia e sazonalidade dos dados, o modelo ARIMA se baseia na autocorrelação dos dados

Estacionariedade

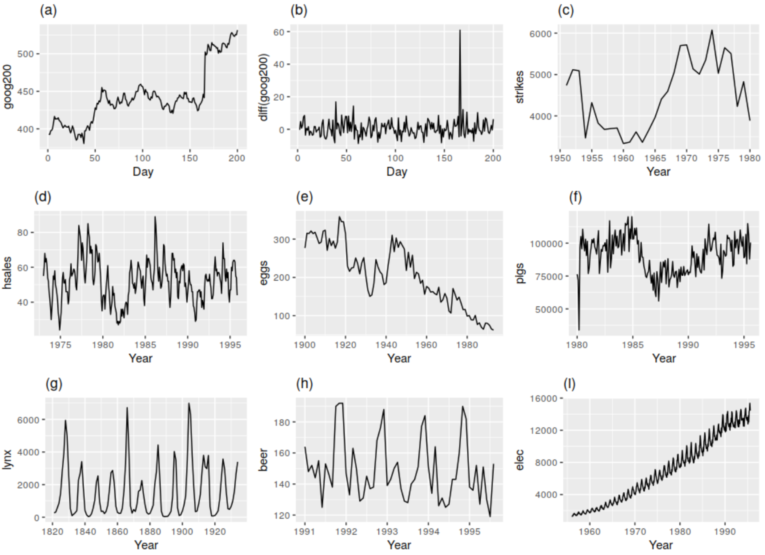
Antes de começar a falar do ARIMA, precisamos saber o conceito de estacionariedade.

Uma série temporal é dita estacionária quando suas propriedades não variam no tempo.

Exemplificando fracamente, quando sua média e variancia são constantes em toda a série.

Isso não ocorre quando presenciamos tendências e/ou sazonalidade em uma série.

Estacionariedade



Estacionariedade

Estacionarias (B,G)

Diferenciação

Para lidar com o problema da Estacionariedade, podemos usar o conceito de **diferenciação**

Ela é calculada através da diferença entre os pontos consecutivos de uma série, estabilizando assim a média da mesma.

1° Diferença

$$y'_t = y_t - y_{t-1}$$

2° Diferença

$$y''_t = y'_t - y'_{t-1} = y_t - 2y_{t-1} + y_{t-2}$$

Diferença Sazonal

$$y'_t = y_t - y_{t-m}$$

Teste de estacionariedade

Há alguns testes que nos informam se uma série é estacionária. Os mais conhecidos são o ADF e o KPSS.

Nós iremos testar se a serie de passagens é estacionária com a função `ur.kpss` do pacote `urca`.

A hipótese nula é que a série é estacionária

```
library(urca)

AirPassengers %>% ur.kpss() %>% summary()
```

Teste de estacionariedade

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.7395
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

O valor de test-statistic = 2.7395. O valor crítico há 1% é de 0.739

Ou seja, podemos rejeitar que ela é **estacionária**

Teste de estacionariedade

Teremos então de diferenciar a série e testar novamente. Podemos diferenciar a série usando a função diff()

```
AirPassengers %>% diff() %>% ur.kpss() %>% summary()
```

Teste de estacionariedade

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0146
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Como o valor de teste é menor do que o valor crítico a 1%, não rejeitamos que a 1° diferença é estacionária.

Modelos Autorregresivos

Modelos auto regressivos são parecidos com os modelos de regressão múltipla. Porém as variaveis exógenas agora são os valores defasados da variável endógena.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

Nós nos referimos a esses modelos como um AR(p), onde p é a ordem da defasagem

Modelos de médias móveis

Em vez de usar valores passados da variável de previsão em uma regressão, um modelo de média móvel usa erros de previsão passados em um modelo semelhante a regressão.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

Nós nos referimos a isso como um MA (q), um modelo de média móvel de ordem q. Obviamente, não observamos os valores de ε, portanto, não é realmente uma regressão no sentido usual.

ARIMA Especificação

Se combinarmos diferenciação com autoregressão e um modelo de média móvel, obteremos um modelo ARIMA não sazonal. O modelo completo não sazonal pode ser escrito como

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

ARIMA Especificando

Nos chamamos esse modelo de ARIMA(p,d,q) onde

p = parte auto regressiva d = n° de diferenciações q = parte média movel

Estimando um ARIMA

Especificar um arima por conta prórpia requer prática, o que por enquanto está fora do escopo desse curso.

Porém, o R traz a possibilidade de você estimar o melhor modelo possivel com a função `auto.arima`

Gerando o modelo

```
fit <- auto.arima(AirPassengers)
summary(fit)
```

Gerando o modelo

```
## Series: AirPassengers
## ARIMA(2,1,1)(0,1,0)[12]
##
## Coefficients:
##          ar1     ar2      ma1
##       0.5960  0.2143  -0.9819
## s.e.  0.0888  0.0880   0.0292
##
## sigma^2 estimated as 132.3:  log likelihood=-504.92
## AIC=1017.85   AICc=1018.17   BIC=1029.35
##
## Training set error measures:
##                    ME     RMSE     MAE       MPE     MAPE
## Training set 1.342299 10.84619 7.86754 0.4206976 2.800458 (
```

Gerando Previsão

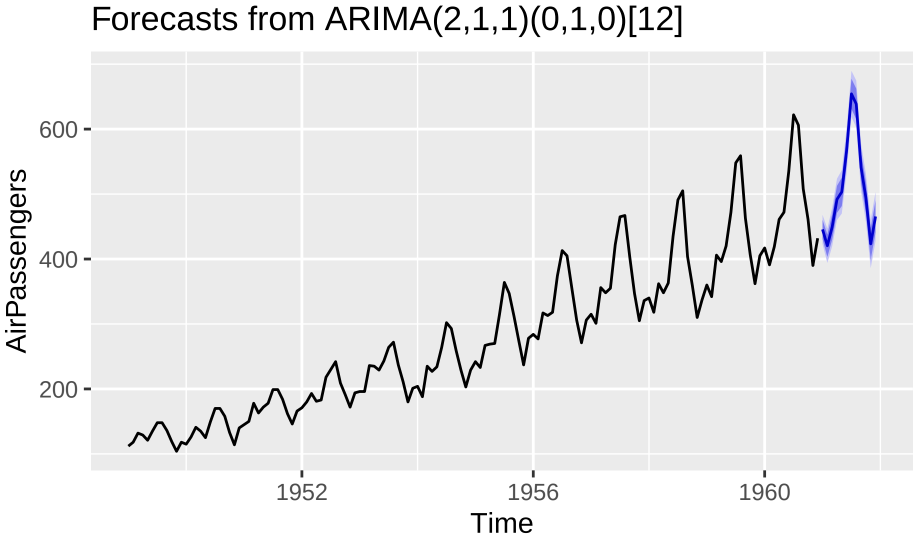
```
prev <- forecast(fit,12)
```

Gerando Previsão

Plotando

```
autoplot(prev)
```

Plotando



Forecasts from ARIMA(2,1,1)(0,1,0)[12]
AirPassengers
600
400
200
1952
1956
1960
Time

Exercicios