

Análise de series temporais

Disciplina: Métodos Estatísticos de Previsão

Rafael Sebastião Arocho e Márcio Antonio Vieira

19/11/2025

Tabela 1: [“resumo dos dados tbl1”, “resumo dos dados tbl2”]

Characteristic	N	Overall N = 247¹	1 N = 48¹	2 N = 99¹
MA type	247 247	456,822 (412,509, 518,911) 332,876 (326,076, 338,709)	432,801 (425,751, 469,921)	
house		100 (40%)	0 (0%)	49 (49%)
unit		147 (60%)	48 (100%)	50 (51%)

¹Median (Q1, Q3); n (%)

Characteristic	N	Overall N = 100¹	4 N = 51¹	5 N = 49¹
MA type	100 100	771,248 (636,635, 824,491) 636,687 (599,614, 745,430)	807,826 (775,199, 952,321)	
house		100 (100%)	51 (100%)	49 (100%)

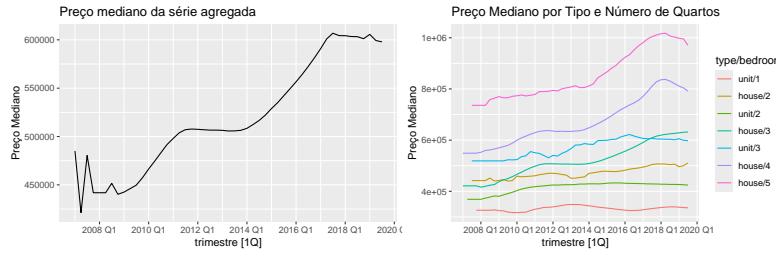
¹Median (Q1, Q3); n (%)

Visão Geral dos Dados

Para esta análise, utilizamos o dataset ***House Property Sales Time Series***.

- **Fonte dos dados:** [Kaggle](#).
- **Período:** 2007 a 2019
- **Frequência:** Trimestral ($n = 347$ observações)
- **Variável Resposta (Y_t):**
 - *MA*: Preço (\$) mediano de casas e unidades habitacionais

Visualização da Série



(a) serie completa
distribuições das series

Observamos que será necessário separar as series em tipo de moradia e número de quartos para simplificar o ajuste.

Parece haver tendência de crescimento em praticamente todas as séries decompostas.

Vamos utilizar diferenciação de nível um e dois para ter outro olhar sobre as séries.

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_line()`).
```

Séries d=1

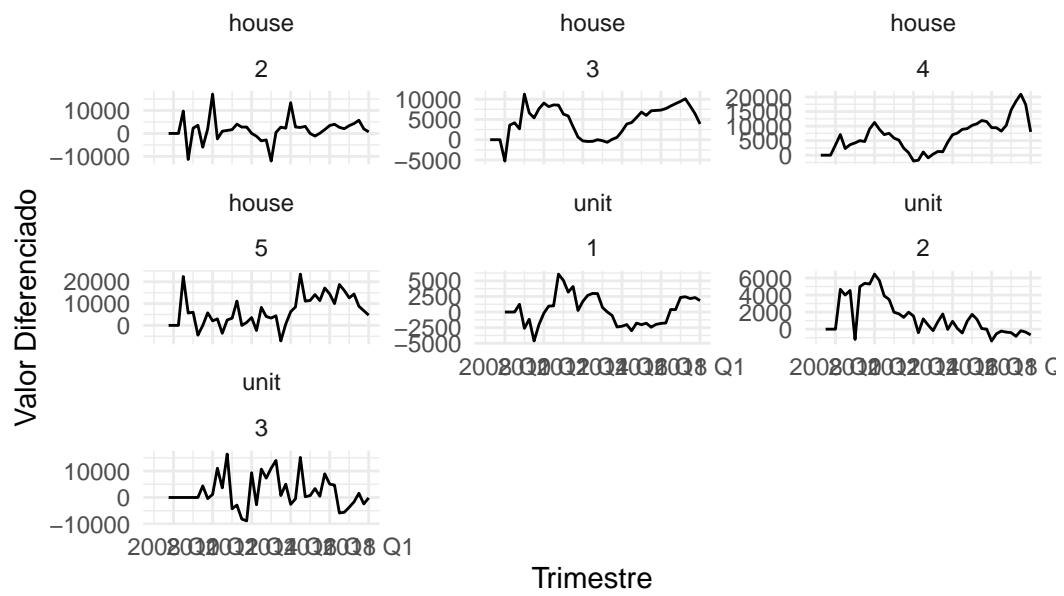


Figura 3: diferenciação de primeiro nível

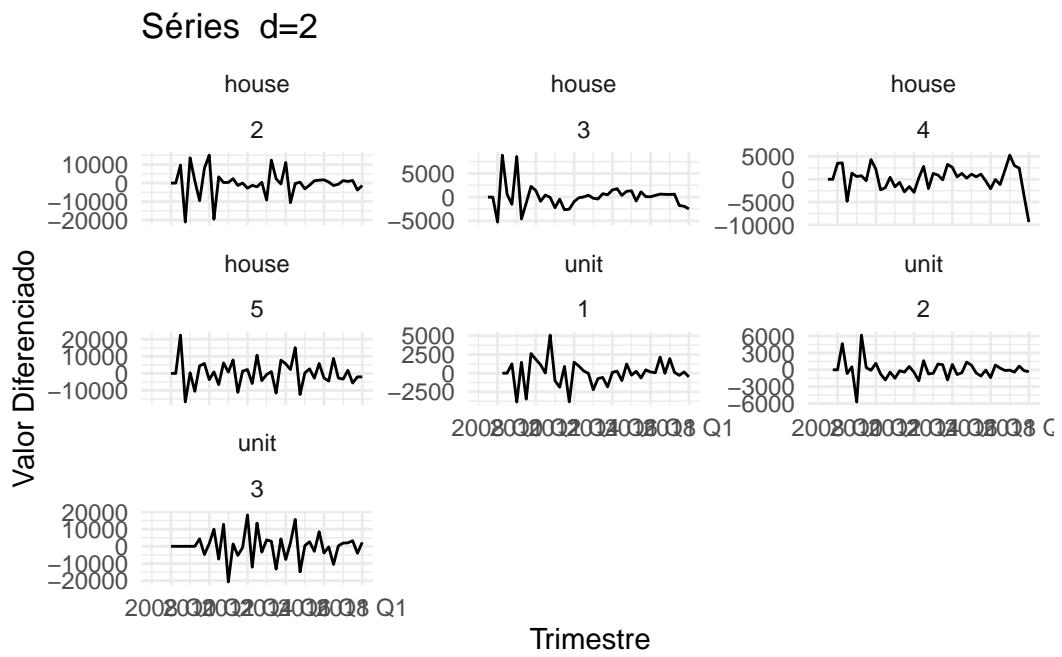


Figura 4: diferenciação de segundo nível

comportamentos diferenciados

Observando as séries com as diferenciações aplicadas vamos seguir a análise com os preços de diferentes combinações de tipo de domicílio e número de quartos.

```
## Ajuste de modelos  
  
type=house x bedrooms=5
```

Para a série de type: *house* e 5 quartos testamos diferenciação para tornar a serie estacionária.

```
an_1 <- plot_acf_pacf(data=treino, type="house", bedrooms=5, difference=1)  
an_1
```

```
[[1]]
```

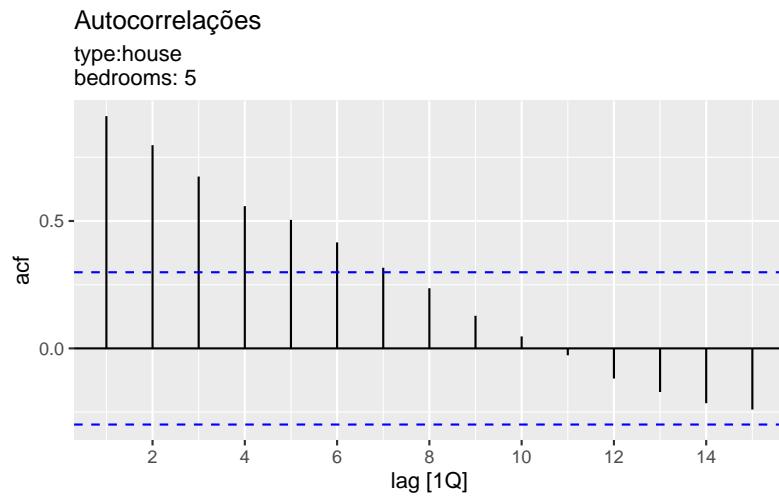


Figura 5: ACF

```
[[2]]
```

Autocorrelações parciais

type:house
bedrooms: 5

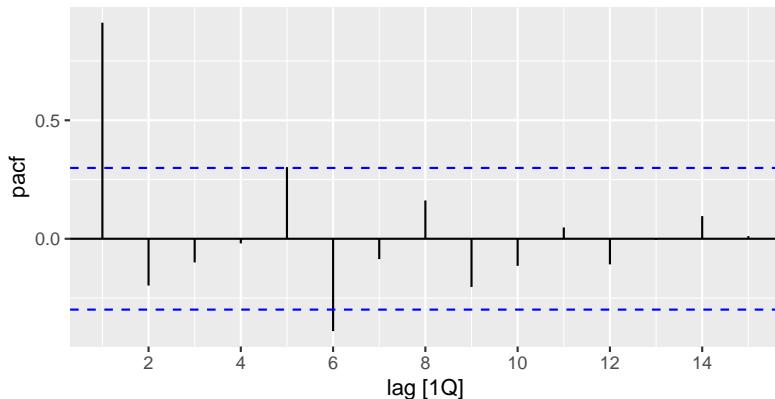


Figura 6: PACF

```
an_2 <- plot_acf_pacf(data=treino, type="house", bedrooms=5, difference=2)
an_2
```

[[1]]

Autocorrelações

type:house
bedrooms: 5

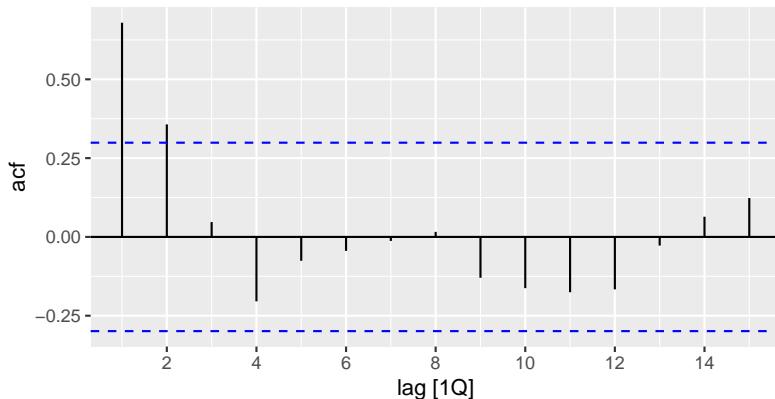


Figura 7: ACF

```
[[2]]
```

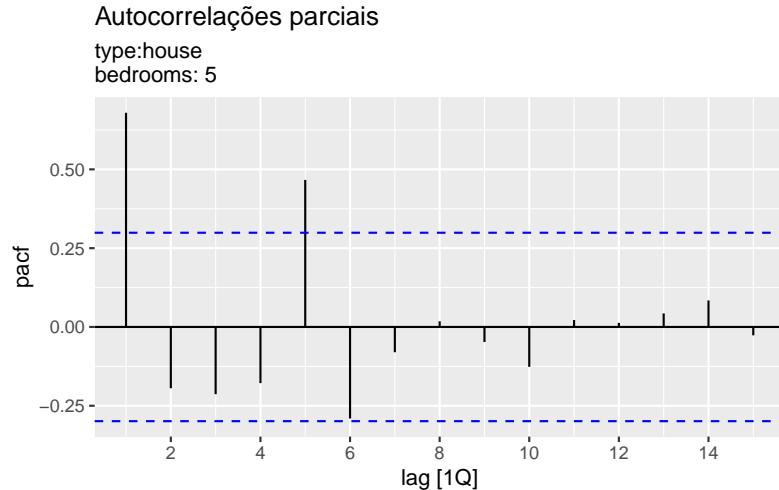


Figura 8: PACF

Parecemos ter um comportamento de ondas senóides no gráfico de ACF e um comportamento irregular no PACF que pode indicar sazonalidade. Vamos ajustar de inicio um modelo ARIMA(2, 1, 0).

Após o ajuste inicial faremos a sobrefixação dos modelos para identificar o melhor modelo através da avaliação das métricas de AIC

```
# A tibble: 3 x 8
  type  bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>        <chr>     <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(2, 1~ ar1    0.356     0.148     2.40  0.0208
2 house      5 ARIMA(MA ~ pdq(2, 1~ ar2    0.216     0.149     1.45  0.154 
3 house      5 ARIMA(MA ~ pdq(2, 1~ cons~ 2707.     907.      2.98  0.00472 

# A tibble: 2 x 8
  type  bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>        <chr>     <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(1, 1~ ar1    0.454     0.136     3.34  1.79e-3
2 house      5 ARIMA(MA ~ pdq(1, 1~ cons~ 3544.     949.      3.74  5.59e-4
```

```

# A tibble: 3 x 8
  type bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>          <chr>    <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(1, 1~ ar1     0.811     0.189     4.30 1.00e-4
2 house      5 ARIMA(MA ~ pdq(1, 1~ ma1    -0.485     0.311    -1.56 1.26e-1
3 house      5 ARIMA(MA ~ pdq(1, 1~ cons~ 1192.      452.      2.64 1.17e-2

# A tibble: 1 x 8
  type bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>          <chr>    <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(1, 2~ ar1    -0.425     0.138    -3.07 0.00383

# A tibble: 2 x 8
  type bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>          <chr>    <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(2, 2~ ar1    -0.485     0.153    -3.17 0.00289
2 house      5 ARIMA(MA ~ pdq(2, 2~ ar2    -0.136     0.151    -0.898 0.374

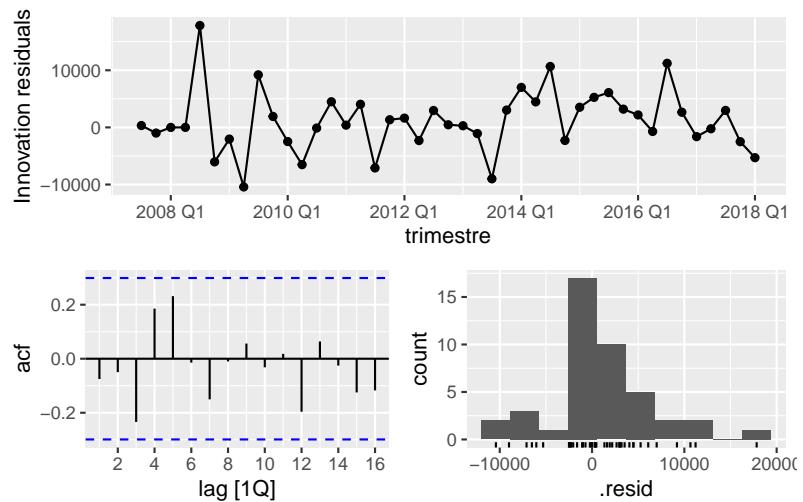
# A tibble: 2 x 8
  type bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>          <chr>    <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(0, 2~ ma1    -0.450     0.155    -2.91 0.00584
2 house      5 ARIMA(MA ~ pdq(0, 2~ sma1   -0.756     0.217    -3.49 0.00118

# A tibble: 3 x 8
  type bedrooms .model          term  estimate std.error statistic p.value
  <chr>    <int> <chr>          <chr>    <dbl>     <dbl>      <dbl>    <dbl>
1 house      5 ARIMA(MA ~ pdq(1, 2~ ar1    1.36e-4    0.295  0.000462 1.000
2 house      5 ARIMA(MA ~ pdq(1, 2~ ma1    -4.50e-1    0.278 -1.62    0.113
3 house      5 ARIMA(MA ~ pdq(1, 2~ sma1   -7.57e-1    0.222 -3.41    0.00148

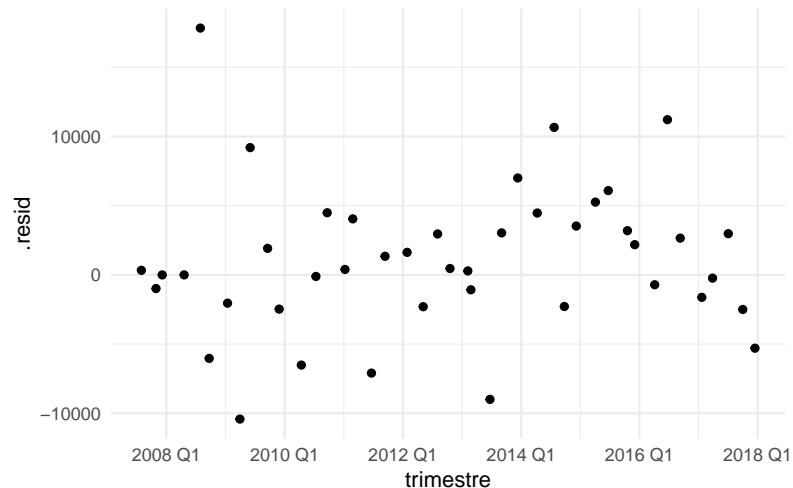
```

Ajustados os modelos daremos sequência na análise com o modelo ARIMA(0,2,1)(0,0,1)[4] por ter apresentado significância nos parâmetros e o menor dos valores de AIC

```
ggttime::gg_tsresiduals(modelo_arima_6)
```



```
residuals(modelo_arima_6) |>
  ggplot(mapping=aes(x=trimestre, y=.resid)) +
  geom_jitter() +
  theme_minimal()
```



```
shapiro.test(residuals(modelo_arima_6)$ .resid)
```

Shapiro-Wilk normality test

```
data: residuals(modelo_arima_6)$resid  
W = 0.96361, p-value = 0.1877
```

```
Box.test(residuals(modelo_arima_6)$resid, lag=12)
```

Box-Pierce test

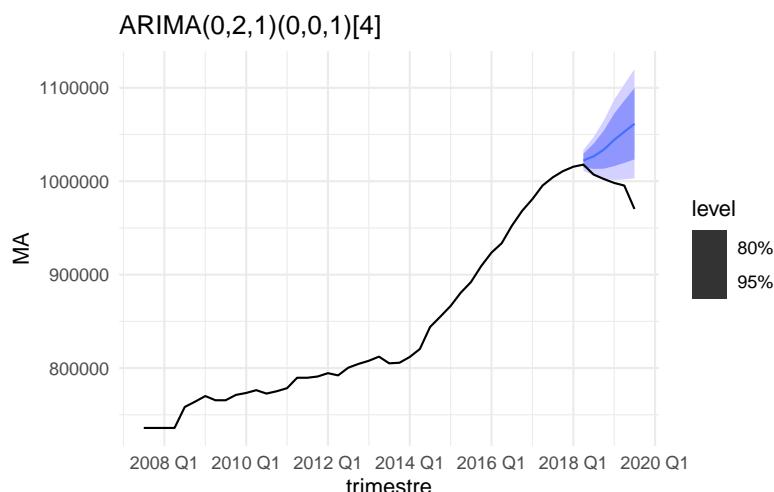
```
data: residuals(modelo_arima_6)$resid  
X-squared = 9.3267, df = 12, p-value = 0.6748
```

Os testes nos mostram que os resíduos apresentam normalidade e o teste de box-pierce não rejeita a hipótese de independencia, o que é um indicador que o ajuste está adequado.

Vamos fazer uma previsão alguns passos a frente

```
forecast(modelo_arima_6, h=6) |>  
  autoplot(bind_rows(treino_casa_5, teste_casa_5)) +  
  labs(title = "ARIMA(0,2,1)(0,0,1)[4]") +  
  theme_minimal()
```

```
`mutate_if()` ignored the following grouping variables:  
* Columns `type`, `bedrooms`
```



Modelo de alisamento exponencial

Seguindo a mesma lógica dos ajustes anteriores faremos o ajuste de modelo de alisamento exponencial. Por não termos identificado sazonalidade utilizaremos alisamento exponencial de Holt-Winters aditivo por haver tendência e sazonalidade na série

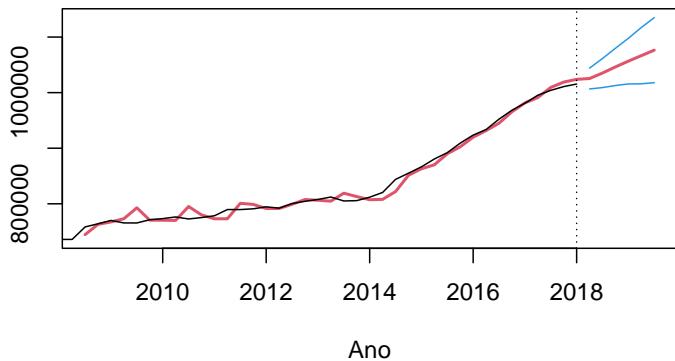
```
ma <- ts(treino_casa_5$MA, start = c(2007, 3), frequency = 4)

AEH <- HoltWinters(ma, alpha = NULL, beta = NULL, gamma = TRUE, seasonal = "additive")
# Calculo das previsões 6 passos a frente e os intervalos de previsão
previsao = predict(AEH,
                     n.ahead=6,
                     prediction.interval = TRUE,
                     level = 0.95,
                     interval="prediction")
previsao
```

	fit	upr	lwr
2018 Q2	1025381	1044152	1006609
2018 Q3	1035293	1061462	1009125
2018 Q4	1046315	1079847	1012783
2019 Q1	1056702	1097765	1015638
2019 Q2	1066553	1117241	1015866
2019 Q3	1076466	1134950	1017982

```
# Constrói o gráfico com ajuste, previsões e intervalos de previsão
plot(AEH, previsao, lwd=2, col="black", xlab="Ano", ylab=NA)
```

Holt–Winters filtering



Faremos agora a comparação dos dois modelos utilizando erro quadrático médio

```
# As previsões pontuais são a primeira coluna da matriz de previsão:  
previsoes_pontuais_hw <- previsao[, "fit"]  
previsoes_pontuais_arima <- forecast(modelo_arima_6, h=6)$mean  
valores_reais <- teste_casa_5$MA  
  
# Cálculo do Erro Quadrático Médio (MSE)  
erros_hw <- previsoes_pontuais_hw - valores_reais  
erros_arima <- previsoes_pontuais_arima - valores_reais  
MSE_hw <- mean(erros_hw^2)  
MSE_arima <- mean(erros_arima^2)  
cat("\nPrevisões Pontuais HW:\n")
```

Previsões Pontuais HW:

```
print(previsoes_pontuais_hw)
```

	Qtr1	Qtr2	Qtr3	Qtr4
2018	1025381	1035293	1046315	
2019	1056702	1066553	1076466	

```
cat("\nPrevisões Pontuais ARIMA:\n")
```

Previsões Pontuais ARIMA:

```
print(previsoes_pontuais_arima)
```

```
[1] 1022220 1026582 1033842 1044259 1052874 1061490
```

```
cat("\nValores Reais (Base de Teste):\n")
```

Valores Reais (Base de Teste):

```
print(valores_reais)
```

```
[1] 1017752 1007114 1002323 998136 995363 970268
```

```
cat("\nErro Quadrático Médio (MSE) para os 6 passos HW:\n")
```

Erro Quadrático Médio (MSE) para os 6 passos HW:

```
print(MSE_hw)
```

```
[1] 3760607450
```

```
cat("\nErro Quadrático Médio (MSE) para os 6 passos ARIMA:\n")
```

Erro Quadrático Médio (MSE) para os 6 passos ARIMA:

```
print(MSE_arima)
```

```
[1] 2524795264
```

Observamos um EQM menor no ajuste com o modelo ARIMA, então podemos concluir que se trata do melhor modelo para ajustar a série de preços medianos de casas com 5 quartos.