

# Análise Temporal do Índice de Preços ao Produtor (PPI) com Modelos ARIMA

Alison Cordeiro Sousa

## Introdução

Este estudo investiga a dinâmica temporal do Índice de Preços ao Produtor (PPI) utilizando modelos ARIMA. A análise começou verificando a estacionariedade da série através do teste Dickey-Fuller Aumentado, que indicou não-estacionariedade ( $p$ -valor=0.83), levando à aplicação de diferenciação para estabilizar a média. A série diferenciada mostrou-se estacionária ( $p$ -valor=0.08), permitindo a modelagem ARIMA. Foram testados diversos modelos combinando componentes autoregressivos (AR) e de médias móveis (MA), com e sem diferenciação, selecionando o melhor através do critério AIC. O modelo ARIMA(2,1,3) emergiu como o mais adequado, com AIC de 395.03, significativamente melhor que alternativas como ARIMA(1,1,1) ou modelos sem diferenciação.

## Resultados e Previsões

O modelo selecionado apresentou coeficientes significativos:  $AR_1=1.47$ ,  $AR_2=-0.61$ ,  $MA_1=-0.97$ ,  $MA_2=0.14$  e  $MA_3=0.31$ , com resíduos mostrando comportamento aleatório adequado (sem autocorrelação nos testes ACF/PACF). A previsão para 100 períodos à frente revela uma trajetória estável, com intervalos de confiança que se ampliam conforme aumenta o horizonte preditivo - comportamento esperado em séries temporais. A comparação entre os modelos testados demonstrou a importância da diferenciação (termo  $d=1$ ) e da combinação de componentes AR e MA para capturar adequadamente a dinâmica da série. Estes resultados fornecem ferramentas valiosas para antecipar movimentos futuros no índice de preços, com aplicações em planejamento econômico e formulação de políticas públicas.

---

```
# Pacotes necessários
library(tseries)
library(forecast)
library(ggplot2)

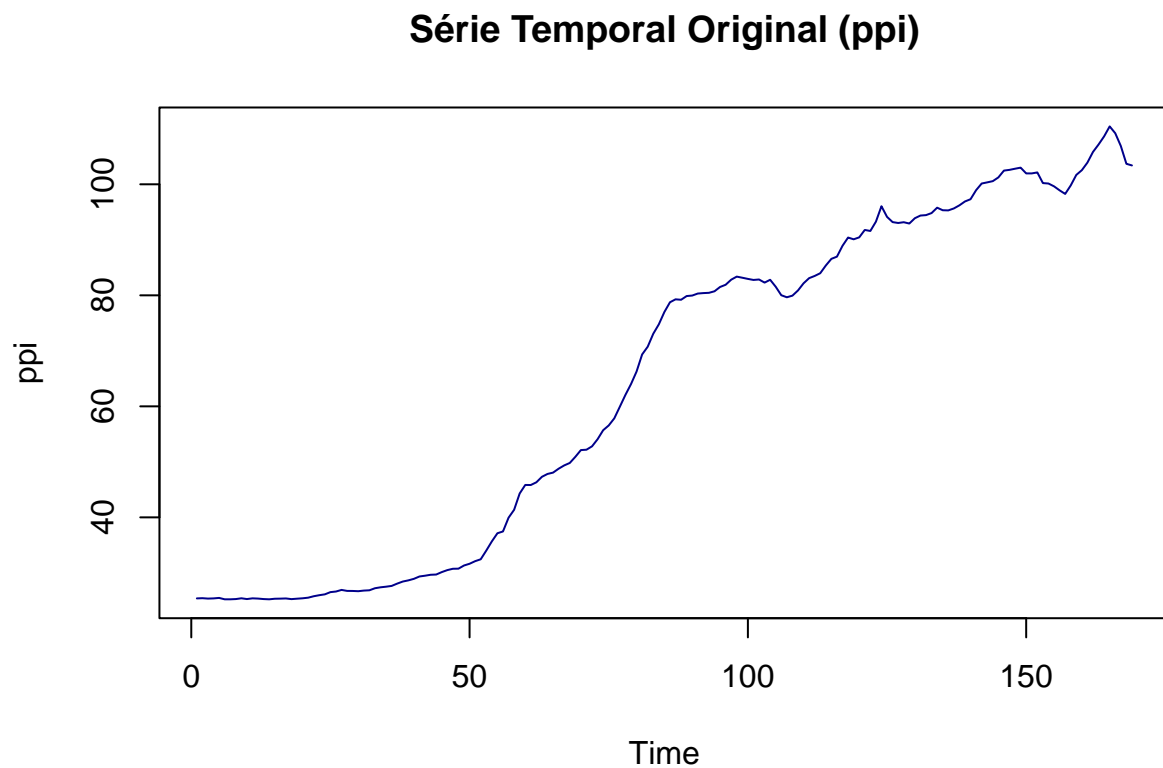
# Definir diretório de trabalho
setwd("C:/Users/PC GAMER/Downloads/data")

# Carregar os dados
df <- read.csv("timeseries.csv")
Y <- df$ppi
```

```
# Estatísticas descritivas e gráfico da série original  
summary(Y)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
##    25.24   29.66   77.00   64.68   93.27  110.43
```

```
plot.ts(Y, main = "Série Temporal Original (ppi)", ylab = "ppi", col = "darkblue")
```



```
# Teste Dickey-Fuller para estacionariedade  
adf_original <- adf.test(Y, alternative = "stationary")  
print(adf_original)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: Y  
## Dickey-Fuller = -1.3857, Lag order = 5, p-value = 0.8327  
## alternative hypothesis: stationary
```

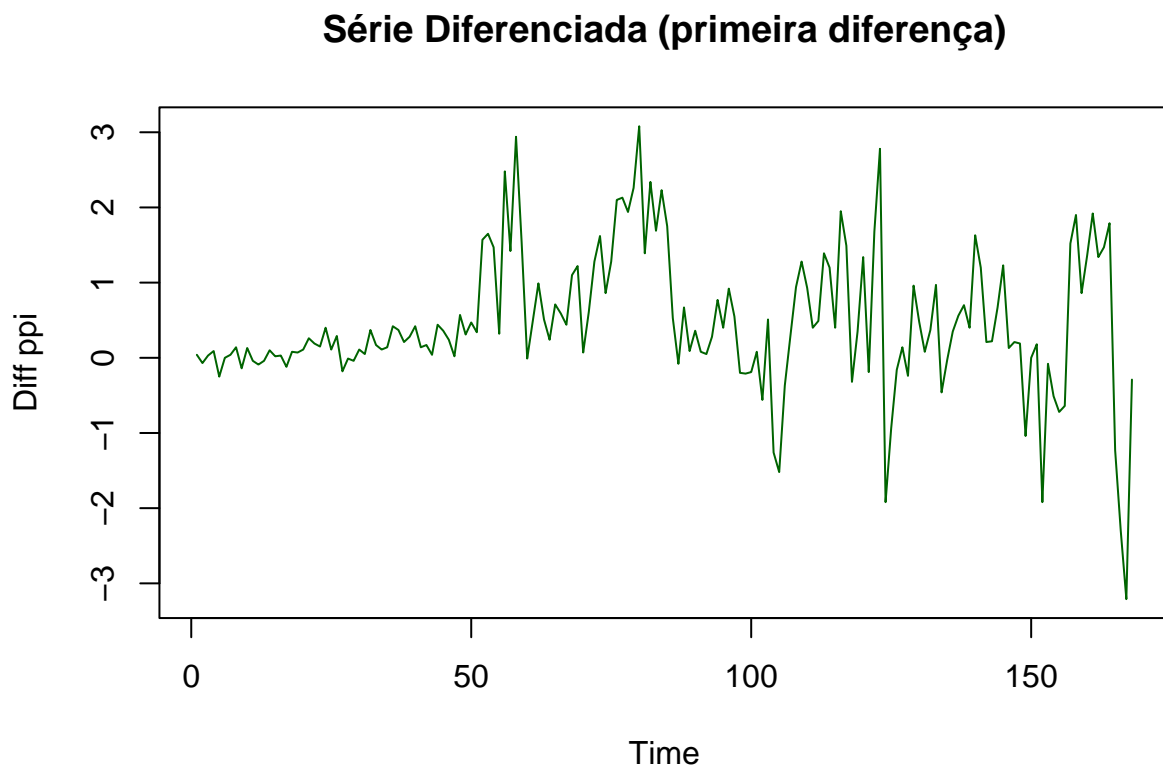
```
# Diferenciação para estacionarizar
```

```
dY <- diff(Y)
```

```
summary(dY)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.2100  0.0300  0.3000  0.4643  0.9625  3.0800
```

```
plot.ts(
  dY,
  main = "Série Diferenciada (primeira diferença)",
  ylab = "Diff ppi",
  col = "darkgreen"
)
```



```
adf_diff <- adf.test(dY, alternative = "stationary")
print(adf_diff)
```

```
##
## Augmented Dickey-Fuller Test
##
## data:  dY
```

```
## Dickey-Fuller = -3.2459, Lag order = 5, p-value = 0.08252
## alternative hypothesis: stationary
```

```
# Função para ajustar modelo ARIMA e retornar modelo, AIC e ordem
ajustar_arima <- function(order_vec, serie) {
  fit <- tryCatch(
    arima(serie, order = order_vec),
    error = function(e) NULL
  )
  if (is.null(fit)) return(NULL)
  list(model = fit, aic = AIC(fit), order = order_vec)
}

# Modelos ARIMA a testar
modelos_testar <- list(
  c(1, 0, 0),
  c(2, 0, 0),
  c(0, 0, 1),
  c(1, 0, 1),
  c(1, 1, 0),
  c(0, 1, 1),
  c(1, 1, 1),
  c(1, 1, 3),
  c(2, 1, 3)
)

# Ajustar modelos e filtrar os válidos
resultados <- lapply(modelos_testar, ajustar_arima, serie = Y)
resultados_validos <- Filter(Negate(is.null), resultados)

# Criar tabela resumo com ordem e AIC
aic_table <- do.call(
  rbind,
  lapply(resultados_validos, function(x) {
    data.frame(p = x$order[1], d = x$order[2], q = x$order[3], AIC = x$aic)
  })
)
print(aic_table[order(aic_table$AIC), ])
```

```
##   p d q      AIC
## 9 2 1 3 395.0309
## 8 1 1 3 395.3484
## 7 1 1 1 396.0575
## 5 1 1 0 400.1912
## 2 2 0 0 404.5373
## 6 0 1 1 426.7043
## 4 1 0 1 440.8180
```

```
## 1 1 0 0 502.4044
## 3 0 0 1 1410.9622
```

```
# Selecionar melhor modelo pelo menor AIC
indice_melhor <- which.min(sapply(resultados_validos, function(x) x$aic))
melhor_modelo <- resultados_validos[[indice_melhor]]

cat(
  "Melhor modelo ARIMA selecionado: (p,d,q) = (",
  melhor_modelo$order[1], ", ",
  melhor_modelo$order[2], ", ",
  melhor_modelo$order[3], ") com AIC = ",
  round(melhor_modelo$aic, 2), "\n\n"
)
```

```
## Melhor modelo ARIMA selecionado: (p,d,q) = ( 2 , 1 , 3 ) com AIC = 395.03
```

```
# Resumo do melhor modelo
summary(melhor_modelo$model)
```

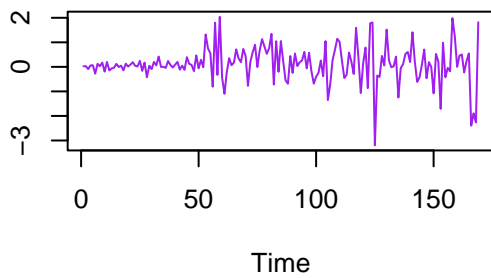
```
##
## Call:
## arima(x = serie, order = order_vec)
##
## Coefficients:
##          ar1          ar2          ma1          ma2          ma3
##      1.4672   -0.6053   -0.9747    0.1377    0.3095
## s.e.  0.1736    0.1600    0.1728    0.1317    0.0867
##
## sigma^2 estimated as 0.5688:  log likelihood = -191.52,  aic = 395.03
##
## Training set error measures:

##              ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN
```

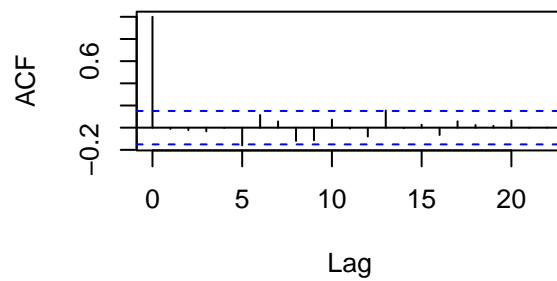
```
# Diagnóstico dos resíduos
par(mfrow = c(2, 2))
plot.ts(residuals(melhor_modelo$model),
  main = "Resíduos do modelo ARIMA",
  col = "purple"
)
acf(residuals(melhor_modelo$model), main = "ACF dos resíduos")
pacf(residuals(melhor_modelo$model), main = "PACF dos resíduos")
qqnorm(residuals(melhor_modelo$model))
qqline(residuals(melhor_modelo$model))
```

residuals(melhor\_modelo\$model)

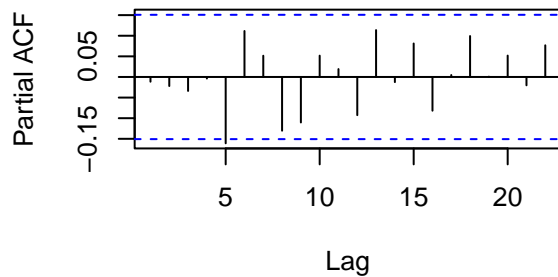
**Resíduos do modelo ARIMA**



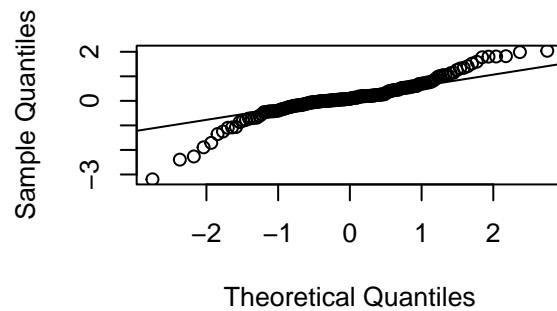
**ACF dos resíduos**



**PACF dos resíduos**



**Normal Q-Q Plot**



```
par(mfrow = c(1, 1))

# Previsão para 100 períodos
horizonte <- 100
prev <- predict(melhor_modelo$model, n.ahead = horizonte)

tempo_original <- 1:length(Y)
tempo_prev <- (length(Y) + 1):(length(Y) + horizonte)

df_ic <- data.frame(
  Tempo = tempo_prev,
  Inferior = prev$pred - 2 * prev$se,
  Superior = prev$pred + 2 * prev$se
)

# Gráfico final da série com previsão e intervalo de confiança
ggplot() +
  geom_line(
    data = data.frame(Tempo = tempo_original, Valor = Y),
    aes(x = Tempo, y = Valor),
    color = "darkblue"
  ) +
```

```

geom_line(
  data = data.frame(Tempo = tempo_prev, Valor = prev$pred),
  aes(x = Tempo, y = Valor),
  color = "red"
) +
geom_ribbon(
  data = df_ic,
  aes(x = Tempo, ymin = Inferior, ymax = Superior),
  alpha = 0.3,
  fill = "pink"
) +
labs(
  title = paste0(
    "Previsão ARIMA(",
    paste(melhor_modelo$order, collapse = ","),
    ") para ppi"
  ),
  x = "Tempo",
  y = "ppi"
) +
theme_minimal(base_size = 15) +
theme(plot.title = element_text(face = "bold"))

```

