

```

setwd("")
# Identificação e Estimação de Modelos ARMA(p,q) Simulados
# Período de Simulação: t = 1,...,200
# 1) Processo AR(1):  $Y_a(t) = 0,7Y_a(t-1) + e(t)$ 
# Características:  $Y_a(1) = 0$ ;  $e(t) \sim N(0,1/4)$ 
# 2) Processo MA(2):  $Y_b(t) = e(t) + 0,4e(t-1) - 0,5e(t-2)$ 
# Características:  $e(t) \sim N(0,1/4)$ 
# 3) Processo ARMA(2,1):  $Y_c(t) = 0,6Y_c(t-1) - 0,5Y_c(t-2) + e(t) + 0,7e(t-1)$ 
# Características: yc <- arima.sim(n=200, list(ar=c(0.6,-0.5),
ma=c(0.7), sd=sqrt(0.5)))
#
=====
=====
rm(list=ls())
graphics.off()
# Carregando Pacotes
#
=====
=====
require(TSA) # a chamada a este pacote é para poder usar
# o argumento drop.lag.0=T dentro da função acf.
library("forecast")
# Lendo dados: variaveis Ya Yb Yc Yd
#
=====
=====
D <- read.table("series_simuladas.txt", header=TRUE, sep = "")
Dts <- ts(D, start = 1, frequency = 1)
t <- Dts[,1]
ya <- Dts[,2]
yb <- Dts[,3]
yc <- Dts[,4]

# I D E N T I F I C A Ç Ã O

# SÉRIE YA: Calculando a FAC e a FACP
#
=====
=====
ts.plot(ya, main = "Série YA")
par(mfrow = c(1,2))
acf(ya, drop.lag.0 = T, main = "Série YA: FAC")
pacf(ya, main = "Série YA: FACP")

# SÉRIE YB: Calculando a FAC e a FACP
#
=====
=====
ts.plot(yb, main = "Série YB")
par(mfrow = c(1,2))
acf(yb, drop.lag.0 = T, main = "Série YB: FAC")
pacf(yb, main = "Série YB: FACP")

# SÉRIE YC: Calculando a FAC e a FACP
#

```

```

=====
=====
ts.plot(yc, main = "Série YC")
par(mfrow = c(1,2))
acf(yc, drop.lag.0 = T, main = "Série YC: FAC")
pacf(yc, main = "Série YC: FACP")

#   E S T I M A Ç Ã O / D I A G N Ó S T I C O
t.test <- function(modelo_arima){
  # estatística t
  coef <- modelo_arima$coef
  se <- sqrt(diag(modelo_arima$var.coef))
  t <- abs(coef/se)
  # Teste t
  ok <- t > qt(0.975, length(modelo_arima$x) -
sum(modelo_arima$arma[c(1,2,3,4,6,7)]))
  resul <- data.frame(Coef = coef, sd = se, t = t, rej_H0 = ok)
  return(resul)
}

# SÉRIE YA: Modelo AR(1)
#
=====
=====
fit.ya <- Arima(ya, order = c(1,0,0), method = "ML")
fit.ya
t.test(fit.ya)
fit.ya <- Arima(ya, order = c(1,0,0), include.mean = FALSE, method =
"ML")
t.test(fit.ya)
ya_chap <- ya - fit.ya$residuals # computando Yb-chapéu
par(mfrow = c(1,1))
plot(ya, main = "Série YA: Real x Ajustado")
lines(ya_chap, col = 'red')
legend('topleft', legend=c("série real", "série
ajustada"),col=c("black","red"),
      lty=c(1,5), cex=0.8,
      lwd =c(1,2))
qlb.ya <- Box.test(fit.ya$residuals, lag = 18, type="Ljung")
qlb.ya
require(normtest)
jb.norm.test(fit.ya$residuals)

# SÉRIE YB: Modelo MA(2)
#
=====
=====
fit.yb <- Arima(yb, order = c(0,0,2), method = "ML")
fit.yb # resultados da estimação
t.test(fit.yb)
yb_chap <- yb - fit.yb$residuals # computando Yb-chapéu
plot(yb, main = "Série YB: Real x Ajustado")
lines(yb_chap, col = 'red')
legend('topleft', legend=c("série real", "série
ajustada"),col=c("black","red"),
      lty=c(1,5), cex=0.8,
      lwd =c(1,2))

```

```

qlb.yb <- Box.test(fit.yb$residuals, lag = 18, type="Ljung")
qlb.yb
jb.norm.test(fit.yb$residuals)

# SÉRIE YC: Modelo ARMA(2,1)
#
=====
=====
fit.yc1 <- Arima(yb, order = c(2,0,3), method = "ML")
fit.yc1 # resultados da estimação
t.test(fit.yc1)
fit.yc2 <- Arima(yb, order = c(2,0,2), method = "ML")
t.test(fit.yc2)
fit.yc3 <- Arima(yb, order = c(2,0,1), method = "ML")
t.test(fit.yc3)
fit.yc4 <- Arima(yb, order = c(1,0,3), method = "ML")
t.test(fit.yc4)

fit.yc3 <- Arima(yb, order = c(2,0,1), method = "ML")
t.test(fit.yc3)
yb_chap <- yb - fit.yc3$residuals # computando Yb-chapéu
plot(yb, main = "Série YC: Real x Ajustado")
lines(yb_chap, col = 'red')
legend('topleft', legend=c("série real", "série
ajustada"), col=c("black", "red"),
      lty=c(1,5), cex=0.8,
      lwd =c(1,2))
qlb.yc3 <- Box.test(fit.yc3$residuals, lag = 18, type="Ljung")
jb.norm.test(fit.yc3$residuals)

#   P R E V I S Ã O
# SÉRIE YA: Previsão
#
=====
=====
prev.ya <- forecast(object = fit.ya, h=12, level = 0.95)
prev.ya
plot(prev.ya)

# SÉRIE YB: Previsão
#
=====
=====
prev.yb <- forecast(object = fit.yb, h=12, level = 0.95)
prev.yb
plot(prev.yb)

# SÉRIE YC: Previsão
#
=====
=====
prev.yc3 <- forecast(object = fit.yc3, h=12, level = 0.95)
prev.yc3
plot(prev.yc3)

# ===== FIM

```

