

# Atividade 7

Carlos Souza

2023-06-04

```
set.seed(2023)
# Criando uma funcao simulada
simulate_ARMA <- function(p, q) {

  ar_coefs <- rnorm(p)
  ma_coefs <- rnorm(q)

  n <- 50 # observacoes
  series <- numeric(n)

  # ruído branco
  series[1:p] <- rnorm(p)

  for (i in (p + 1):n) {
    # termo AR
    ar_term <- sum(ar_coefs * series[(i - 1):(i - p)])

    # termo MA
    ma_term <- sum(ma_coefs * rnorm(q))

    # valor atual da série
    series[i] <- ar_term + ma_term
  }

  # grafico da serie simulada
  plot(series, type = "l", main = "Serie ARMA Simulada", xlab = "Tempo", ylab = "Valor")

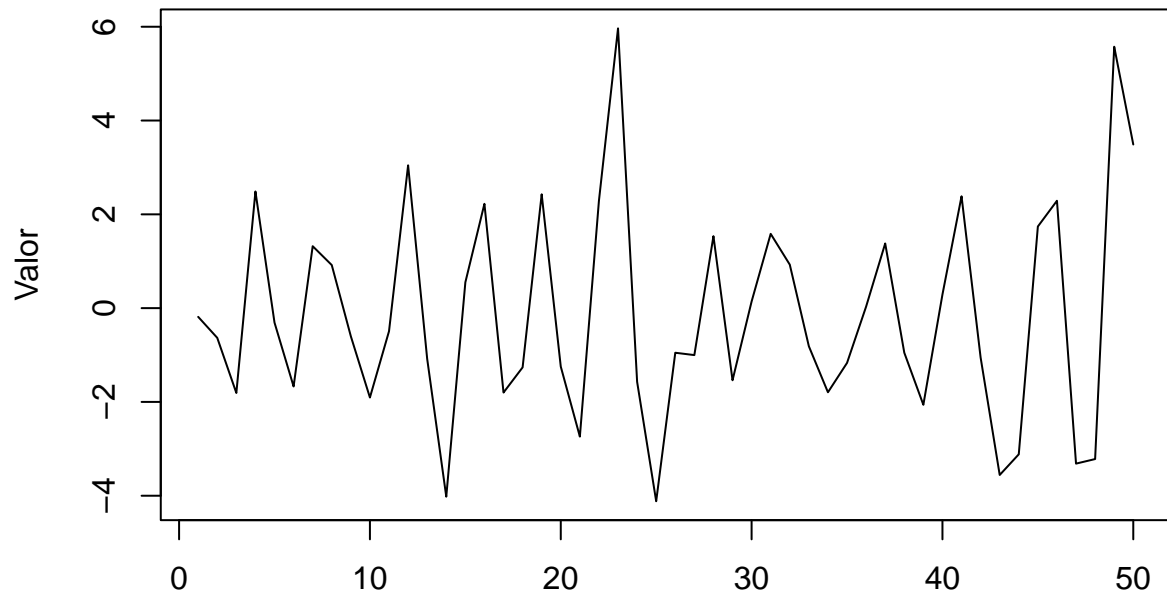
  # histograma
  hist(series, main = "Histograma ", xlab = "Valor", ylab = "Frequencia", prob = TRUE)
  lines(density(series), col = "red")

  # Boxplot
  boxplot(series, main = "Boxplot da Serie")

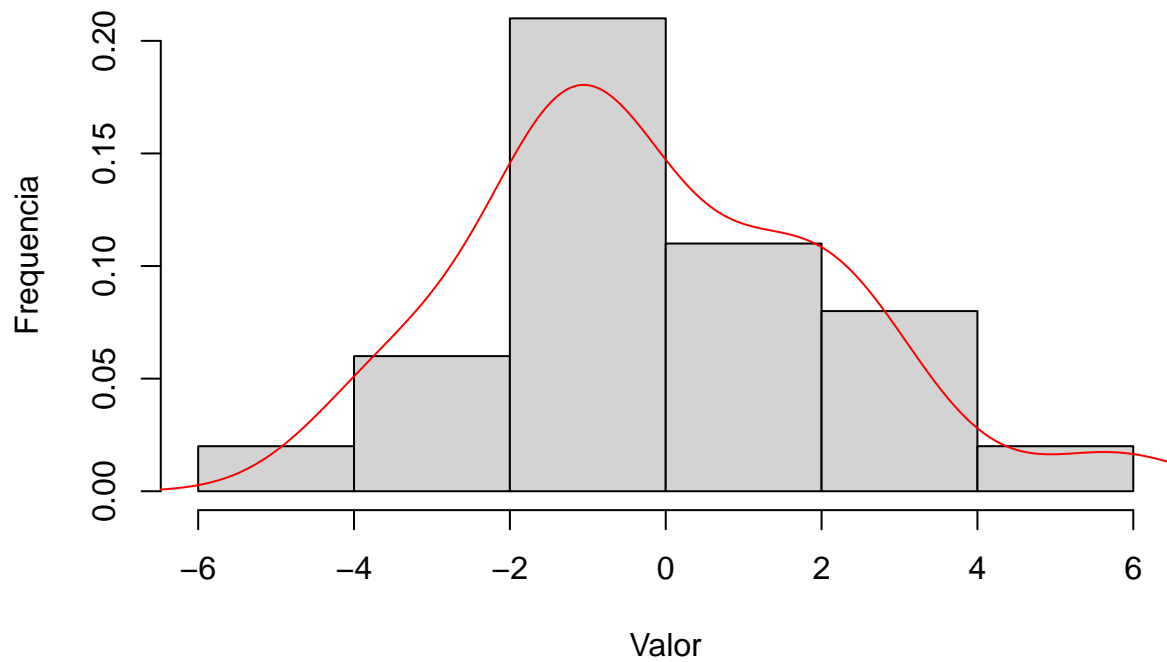
  # Gráfico da função de autocorrelação
  acf(series, main = "Funcao de Autocorrelacao")
}

# serie simulada para o modelo ARMA(p, q)
simulate_ARMA(2, 1)
```

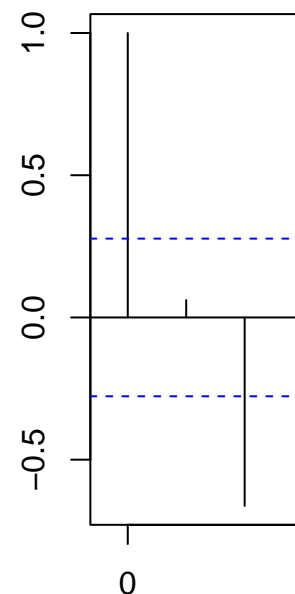
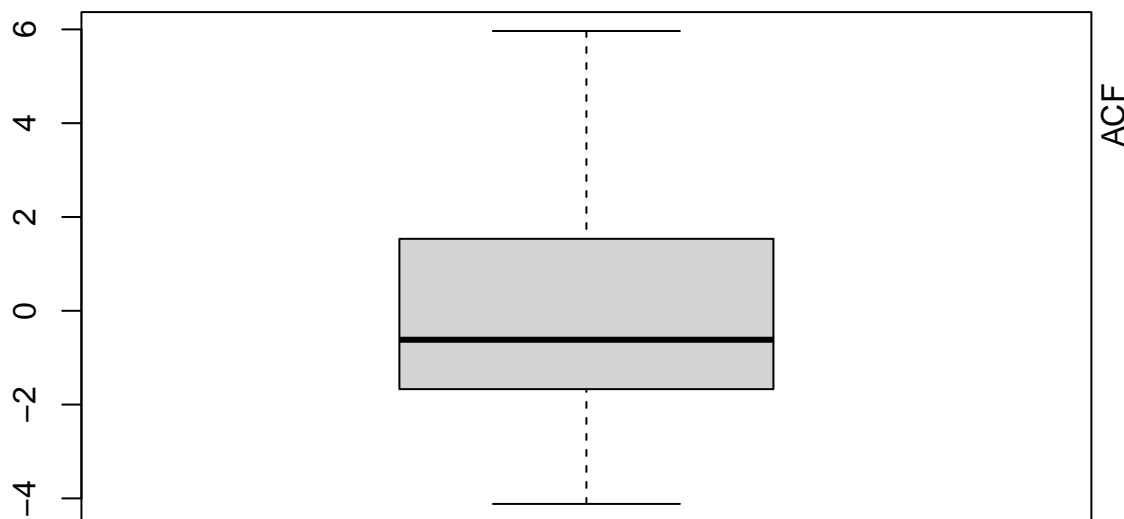
### Serie ARMA Simulada



### Histograma



## Boxplot da Serie



```
par(mfrow = c(1, 2))
# Criando uma função para estimar o ACF
estimate_ACF <- function(series) {
  # Estimar a ACF
  acf_result <- acf(series, main = "Funcao de Autocorrelacao (ACF)")

  # Calculando o limite de significancia superior e inferior
  n <- length(series)
  alpha <- 0.05
  limit <- qnorm(1 - alpha/2) / sqrt(n)

  # Plote do ACF com a regioao de decisao
  plot(acf_result$acf, type = "h", main = "Funcao de Autocorrelacao (ACF)")
  abline(h = limit, col = "red", lty = 2)
  abline(h = -limit, col = "red", lty = 2)
}

# # Criando uma função para estimar o ACFP
estimate_ACFP <- function(series) {
  pacf_result <- pacf(series, main = "Funcao ACFP")

  n <- length(series)
  alpha <- 0.05
  limit <- qnorm(1 - alpha/2) / sqrt(n)

  # Plote do ACFP com a regioao de decisao
  plot(pacf_result$acf, type = "h", main = "Funcao ACFP")
  abline(h = limit, col = "red", lty = 2)
  abline(h = -limit, col = "red", lty = 2)
}
```

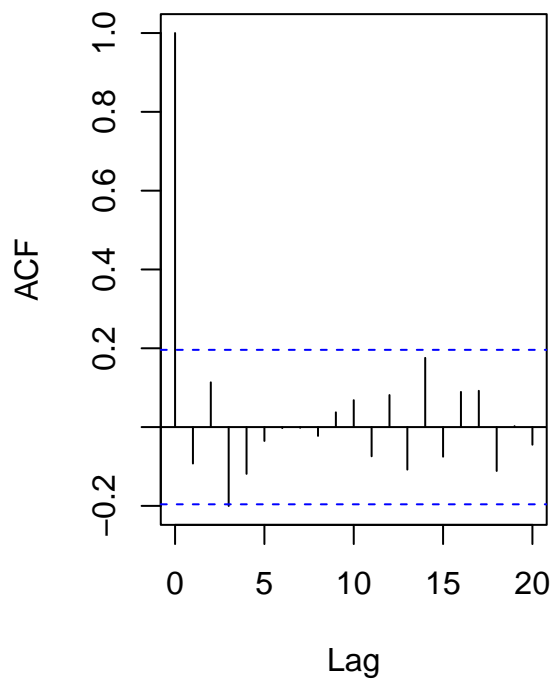
```

}

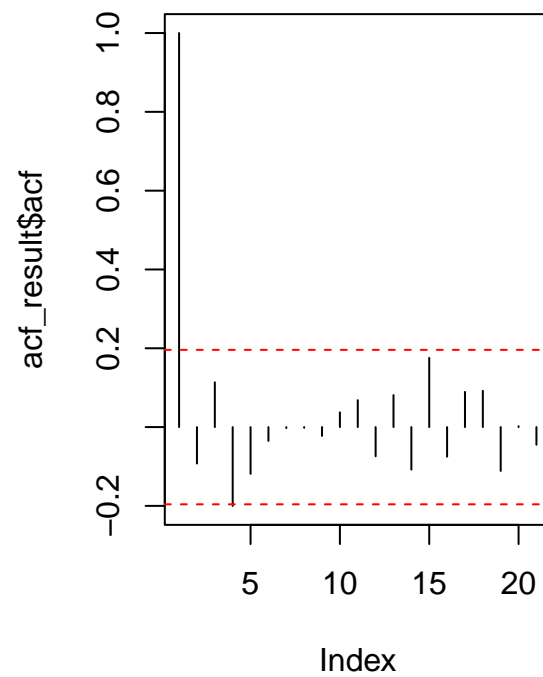
# Serie simulada
series <- rnorm(100)
estimate_ACF(series)

```

**Funcao de Autocorrelacao (ACF)**

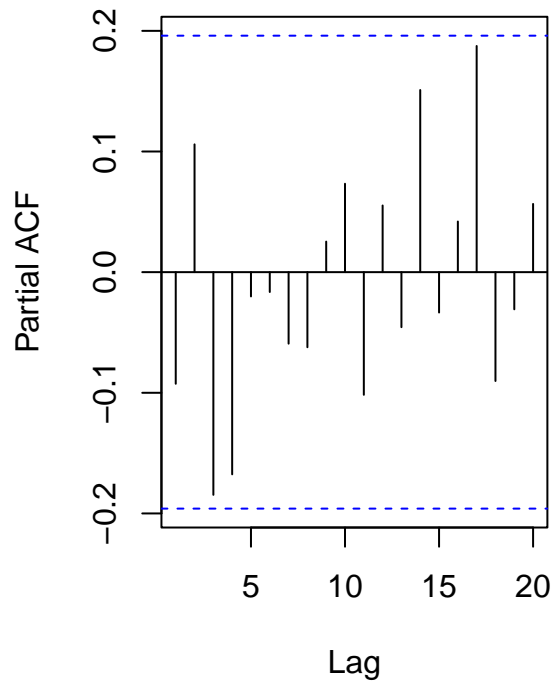


**Funcao de Autocorrelacao (ACF)**

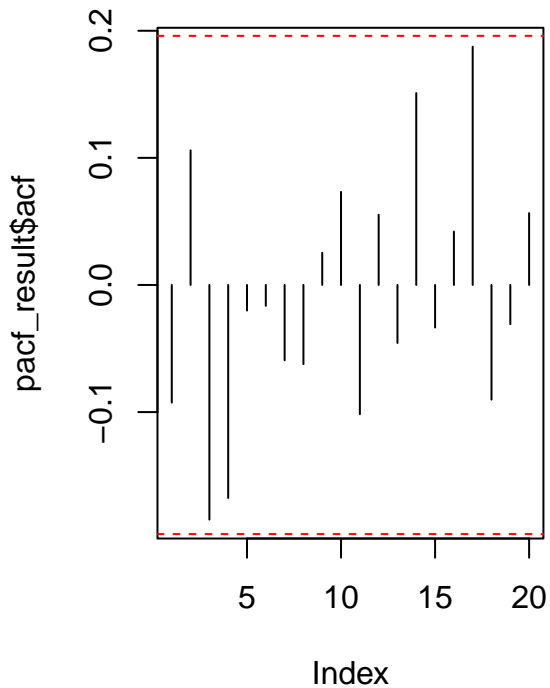


```
estimate_ACFP(series)
```

**Funcao ACFP**



**Funcao ACFP**



```
# Criando uma função para estimar os parâmetros de um modelo AR(p).
# Usando os estimadores de Yule-Walker
yw <- function(estimativa, p) {
  # Verificando se o numero de autocorrelacoes estimadas e suficiente
  num_acf <- length(estimativa)
  if (num_acf <= p) {
    stop("0 numero de autocorrelacoes estimadas e insuficiente para estimar o modelo AR(p)")
  }

  # matriz de autocovariancia
  acv_matrix <- toeplitz(estimativa[1:p])

  # vetor de autocorrelacoes
  acr_vector <- estimativa[2:(p + 1)]

  # Estimando os parametros usando os estimadores de Yule-Walker
  coefs <- solve(acv_matrix) %*% acr_vector

  return(coefs)
}

# funcao de autocorrelacao estimada
estimativa <- c(.6, .1, .2, .9, .5)
p <- 4 # qtd de estimativas

# Estimando os parametros do modelo AR(p)
parametro <- yw(estimativa, p)
cat("Parametros estimados do modelo AR(p):", parametro, "\n")
```

```
## Parametros estimados do modelo AR(p): 0.890625 0.234375 1.296875 -0.796875
```