PSET 6: Time Series Simulation

Juan Ramón Hernández G. Moises Barranco Juárez

Table of contents

Instructions

1. Read each exercise carefully and write the corresponding R code.
2. Provide brief comments explaining each part of the code.
3. Plot the generated time series and their autocorrelation functions (ACF and PACF).
4. Interpret the obtained results.
5. Submit your solutions in a .R file BEFORE 16:10 March 17 2025

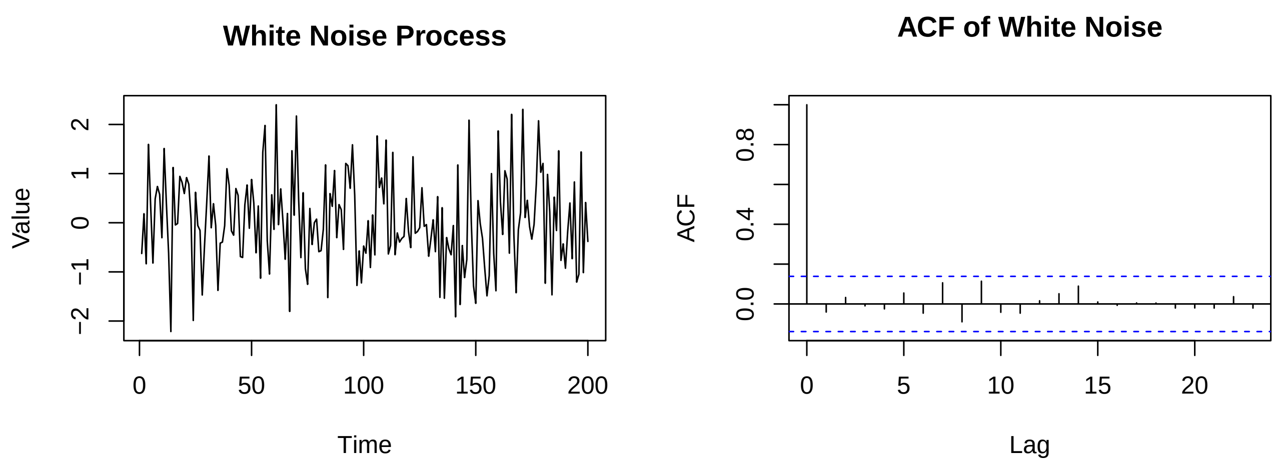
Exercise 1: White Noise Simulation

Generate a Gaussian **white noise** time series with mean 0 and variance 1.

- What is the empirical mean and variance of the simulated series?
- Plot the time series and its autocorrelation function (ACF).

Hint: Use `rnorm()`.

```
set.seed(1)
T <- 200
wn <- rnorm(T, mean = 0, sd = 1)

par(mfrow = c(1, 2))
# Plot
plot(wn, type = "l",
     main = "White Noise Process", ylab = "Value", xlab = "Time")

# ACF
acf(wn, main = "ACF of White Noise")
```



Exercise 2: Simulation of AR(p) Processes

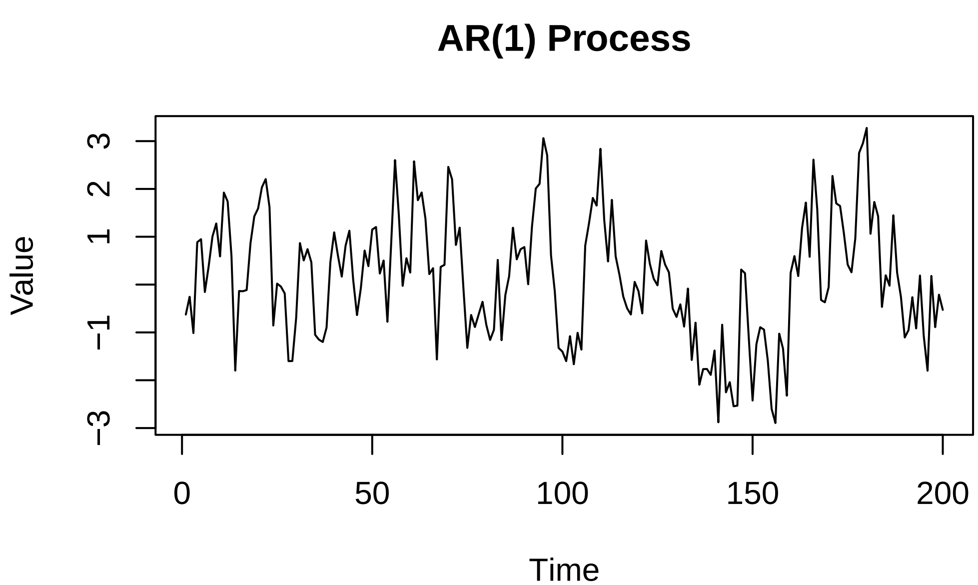
Part A: Simulate an AR(1) process given by:

$$y_t = 0.7y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(0, 1)$$

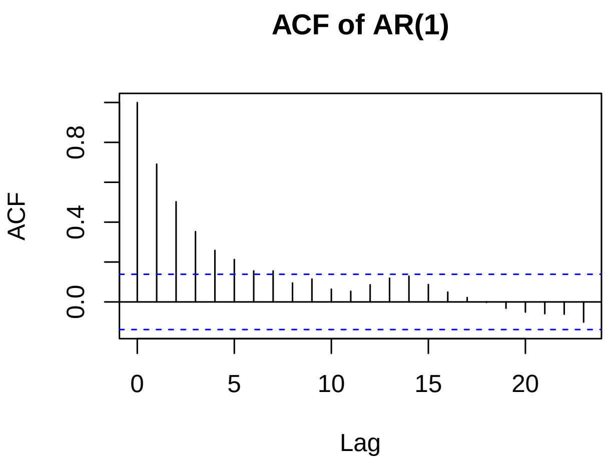
- Generate a time series of length T = 200.
- Manually simulate the AR process using for-loops and recursion.
- Plot the generated series and compare it with white noise.
- Plot the ACF and PACF. What do you observe in terms of temporal dependence?

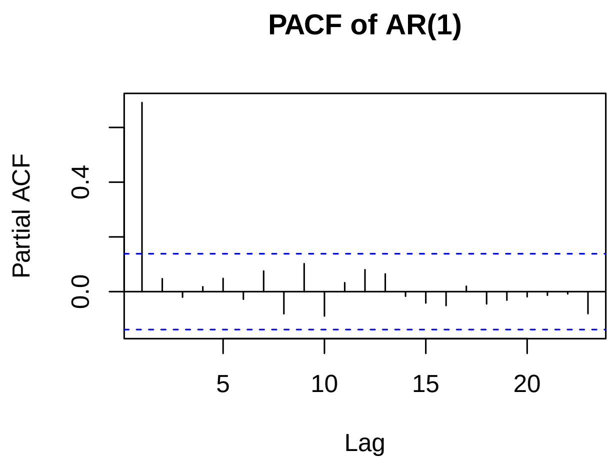
Hint: Use `for (t in 2:T) {y_ar1[t] <- 0.01 * y_ar1[t-1] + rnorm(1)}`

```r
# ar1 <- arima.sim(model = list(ar = 0.7), n = T)
set.seed(1)
y_ar1 <- numeric(T)
y_ar1[1] <- rnorm(1)

for (t in 2:T) {
  y_ar1[t] <- 0.7 * y_ar1[t-1] + rnorm(1)
}

# Plot
plot(y_ar1, type = "l",
     main = "AR(1) Process", ylab = "Value", xlab = "Time")
```



```r
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_ar1, main = "ACF of AR(1)")
pacf(y_ar1, main = "PACF of AR(1)")
```



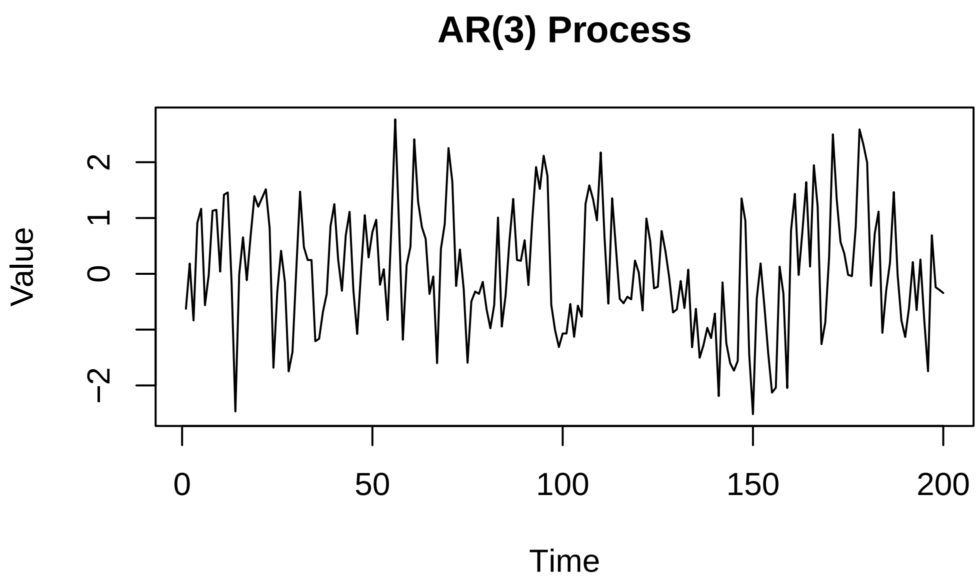


Part B: Simulate an AR(3) process with the following equation:

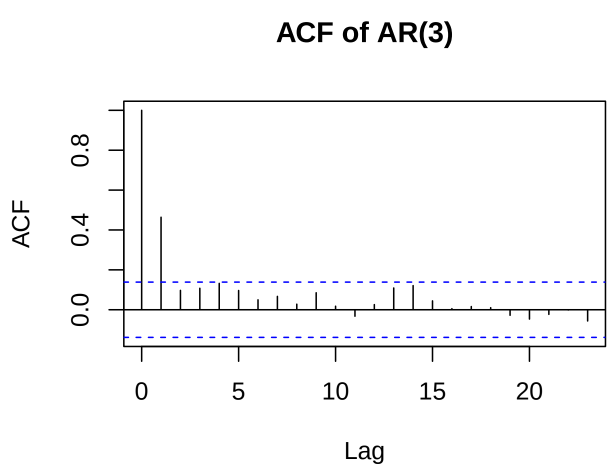
$$y_t = 0.6y_{t-1} - 0.3y_{t-2} + 0.2y_{t-3} + \varepsilon_t, \quad \varepsilon_t \sim \mathrm{WN}(0, 1)$$

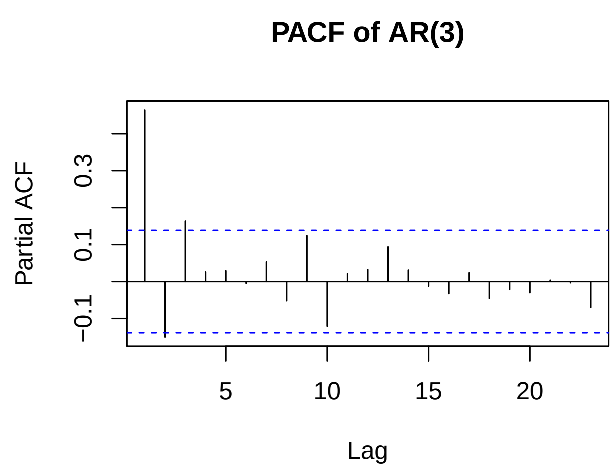
- Generate the series for T = 200.
- Plot the series and compare its behavior with the AR(1) process.
- Analyze the ACF and PACF. How does the autocorrelation structure change?

```r
# ar3 <- arima.sim(model = list(ar = c(0.6, -0.3, 0.2)), n = T)
set.seed(1)
y_ar3 <- numeric(T)
y_ar3[1:3] <- rnorm(3)

for (t in 4:T) {
  y_ar3[t] <- 0.6 * y_ar3[t-1] - 0.3 * y_ar3[t-2] + 0.2 * y_ar3[t-3] + rnorm(1)
}

# Plot
plot(y_ar3, type = "l",
     main = "AR(3) Process", ylab = "Value", xlab = "Time")
```




```r
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_ar3,  main = "ACF of AR(3)")
pacf(y_ar3, main = "PACF of AR(3)")
```





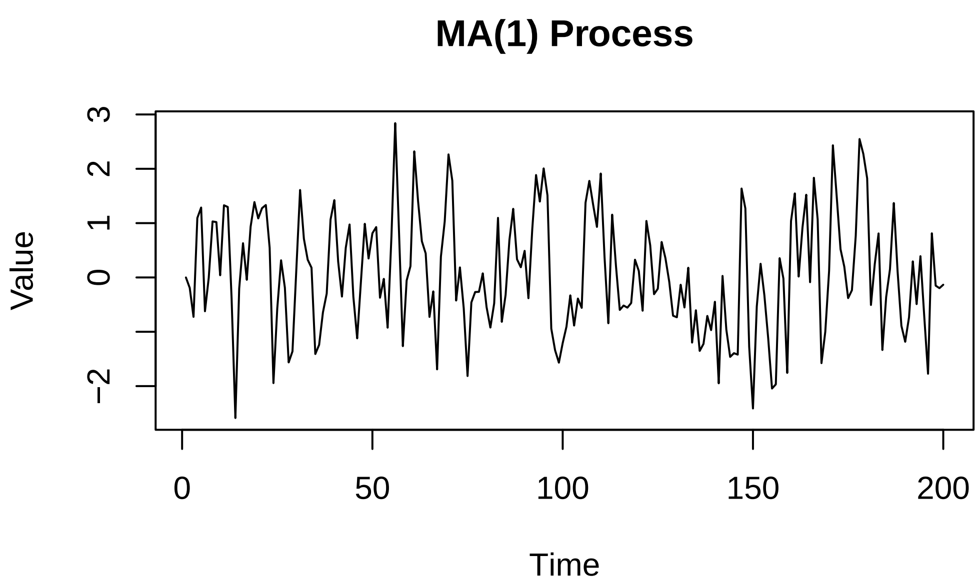
Exercise 3: Simulation of MA(q) Processes

Part A: Simulate an MA(1) process:

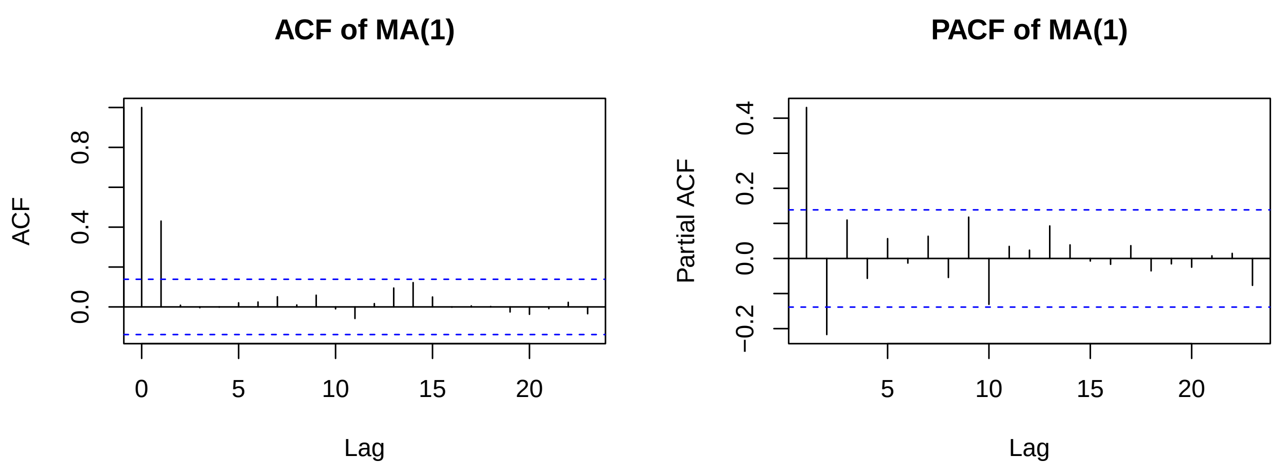
$$y_t = \varepsilon_t + 0.6\varepsilon_{t-1}, \quad \varepsilon_t \sim \mathrm{WN}(0, 1)$$

- Generate a series of length T = 200.
- Manually simulate the MA process using for-loops and recursion.
- Plot the series and compare it with the AR(1) process.
- Analyze the ACF and PACF.

```r
# ma1 <- arima.sim(model = list(ma = 0.6), n = T)
set.seed(1)
y_ma1 <- numeric(T)
e <- rnorm(T)

for (t in 2:T) {
  y_ma1[t] <- e[t] + 0.6 * e[t-1]
}

# Plot
plot(y_ma1, type = "l",
     main = "MA(1) Process", ylab = "Value", xlab = "Time")
```



```
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_ma1,  main = "ACF of MA(1)")
pacf(y_ma1, main = "PACF of MA(1)")
```

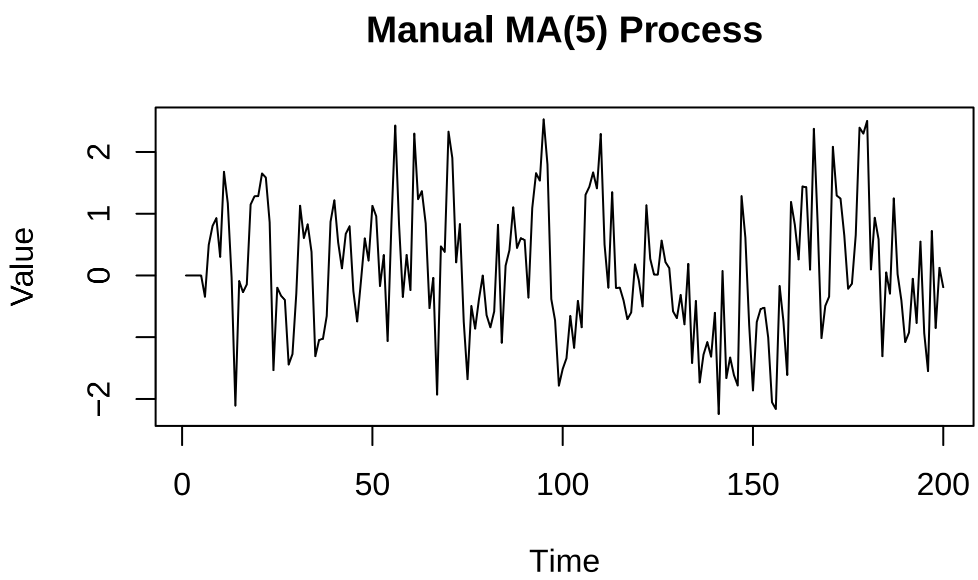


Part B: Simulate an MA(5) process:

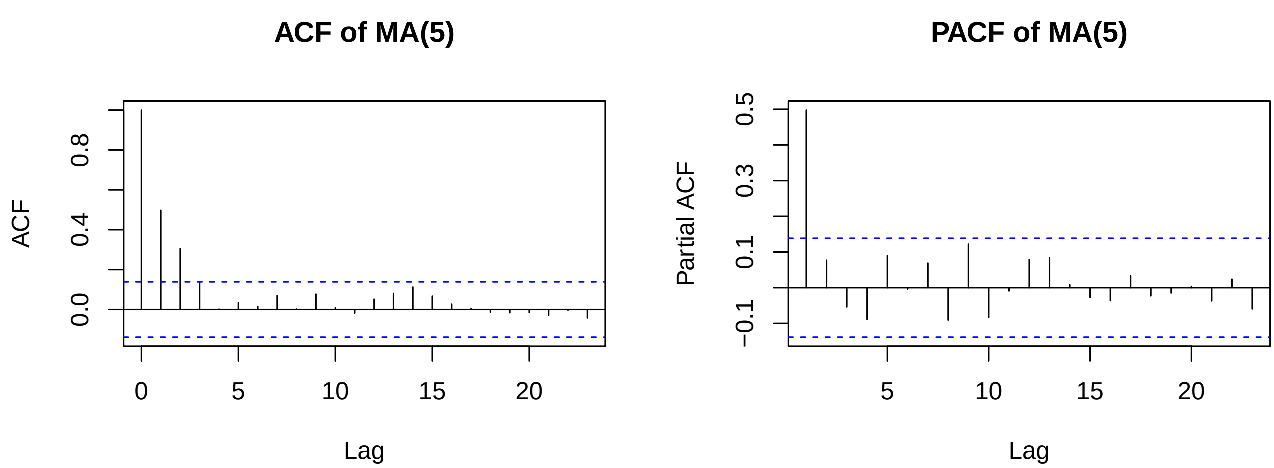
$$y_t = \varepsilon_t + 0.5\varepsilon_{t-1} + 0.3\varepsilon_{t-2} + 0.2\varepsilon_{t-3} + 0.1\varepsilon_{t-4} - 0.1\varepsilon_{t-5}$$

- Generate the series for T = 200.
- Plot the series and compare its behavior with the MA(1) process.
- Analyze the ACF and PACF. How many significant lags appear in the ACF?

```
set.seed(1)
y_ma5 <- numeric(T)
e <- rnorm(T)

for (t in 6:T) {
  y_ma5[t] <- e[t] + 0.5 * e[t-1] + 0.3 * e[t-2] + 0.2 * e[t-3]
  + 0.1 * e[t-4] - 0.1 * e[t-5]
}

# Plot
plot(y_ma5, type = "l",
     main = "Manual MA(5) Process", ylab = "Value", xlab = "Time")
```




```r
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_ma5, main = "ACF of MA(5)")
pacf(y_ma5, main = "PACF of MA(5)")
```



Exercise 4: Simulation of ARMA(p,q) Processes

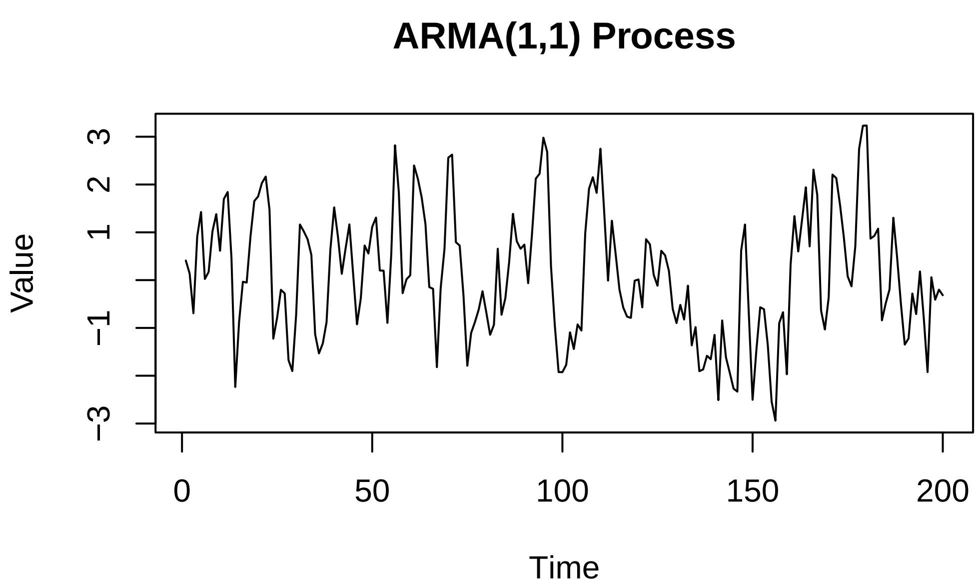
Part A: Simulate an ARMA(1,1) process:

$$y_t = 0.5y_{t-1} + 0.4\varepsilon_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{WN}(0,1)$$

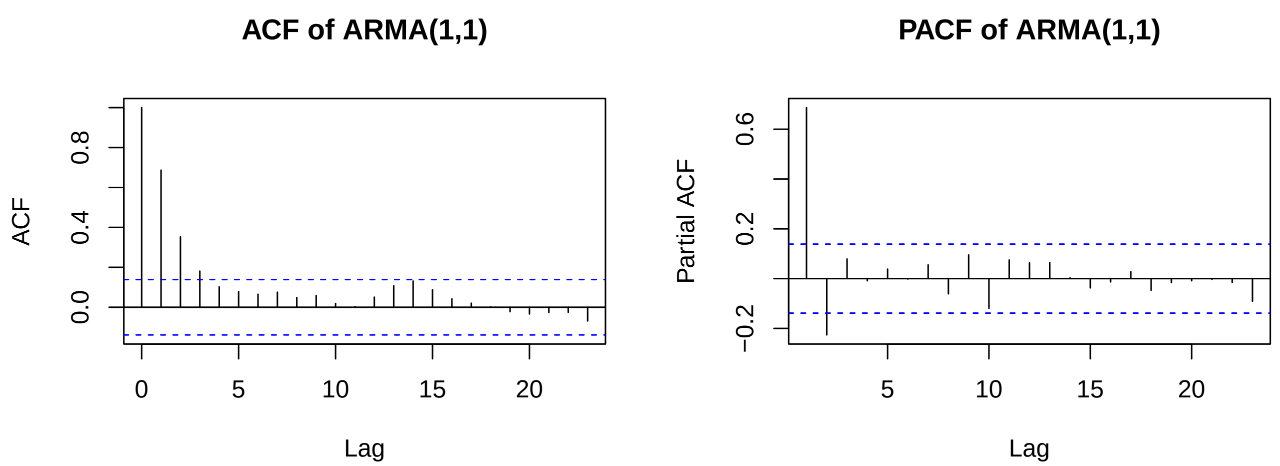
- Generate the series for T = 200.
- Manually simulate the ARMA process using for-loops and recursion.
- Plot the series and its autocorrelation functions (ACF and PACF).
- Compare the graphs with those of the AR(1) and MA(1) processes.

Hint: Use `for (t in 2:T) {y_arma11[t] <- 0.1 * y_arma11[t-1] + 0.1 * e[t-1] + e[t]}`

```
#arma11 <- arima.sim(model = list(ar = 0.5, ma = 0.4), n = T)
set.seed(1)
y_arma11 <- numeric(T)
e <- rnorm(T)
y_arma11[1] <- rnorm(1)

for (t in 2:T) {
  y_arma11[t] <- 0.5 * y_arma11[t-1] + 0.4 * e[t-1] + e[t]
}

# Plot
plot(y_arma11, type = "l",
     main = "ARMA(1,1) Process", ylab = "Value", xlab = "Time")
```



```r
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_arma11, main = "ACF of ARMA(1,1)")
pacf(y_arma11, main = "PACF of ARMA(1,1)")
```

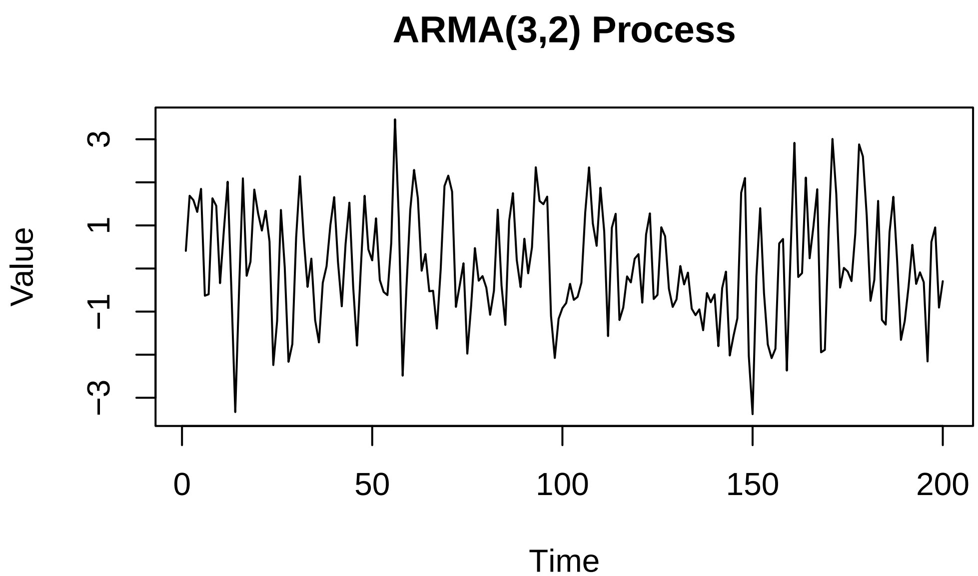


Part B: Simulate an ARMA(3,2) process:

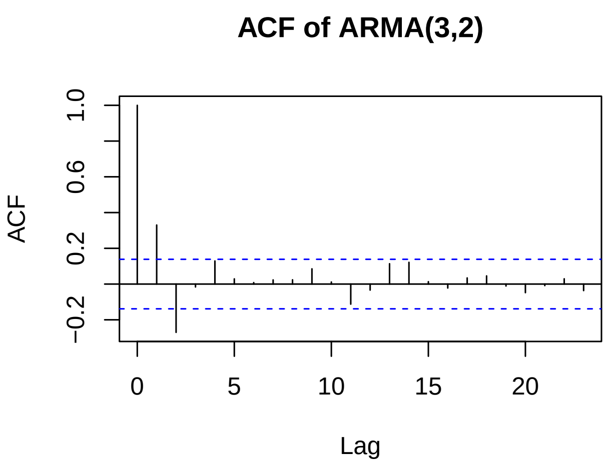
$$y_t = 0.4y_{t-1} - 0.3y_{t-2} + 0.2y_{t-3} + 0.5\varepsilon_{t-1} - 0.4\varepsilon_{t-2} + \varepsilon_t$$

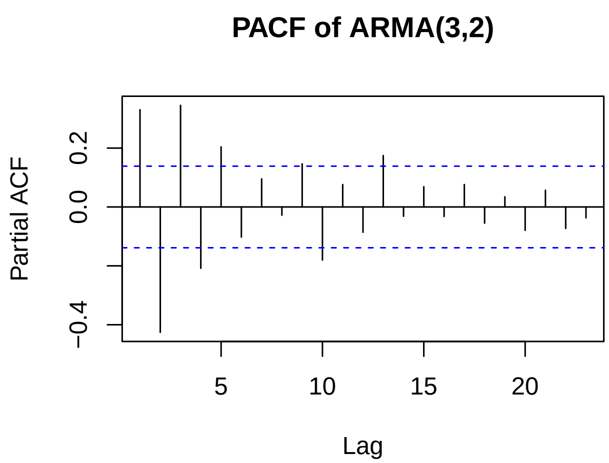
- Generate the series for T = 200.
- Plot the series and compare its behavior with the ARMA(1,1) process.
- Analyze the ACF and PACF. What patterns do you identify?

```r
#arma32 <- arima.sim(model = list(
#  ar = c(0.4, -0.3, 0.2),
#  ma = c(0.5, -0.4)),
#  n = T)
set.seed(1)
y_arma32 <- numeric(T)
e <- rnorm(T)
y_arma32[1:3] <- rnorm(3)

for (t in 4:T) {
  y_arma32[t] <- 0.4 * y_arma32[t-1] - 0.3 * y_arma32[t-2] +
    0.2 * y_arma32[t-3] + 0.5 * e[t-1] - 0.4 * e[t-2] + e[t]
}

# Plot
plot(y_arma32,
     type = "l", main = "ARMA(3,2) Process", ylab = "Value", xlab = "Time")
```




```r
par(mfrow = c(1, 2))
# ACF and PACF
acf(y_arma32, main = "ACF of ARMA(3,2)")
pacf(y_arma32, main = "PACF of ARMA(3,2)")
```





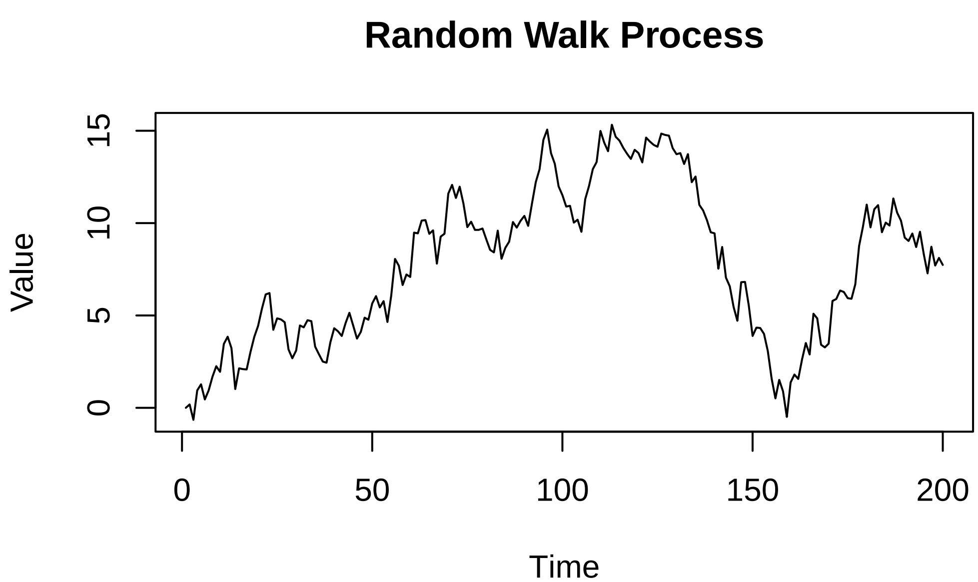
Exercise 5: Exploring Unit Roots

Simulate a **random walk** (without drift):

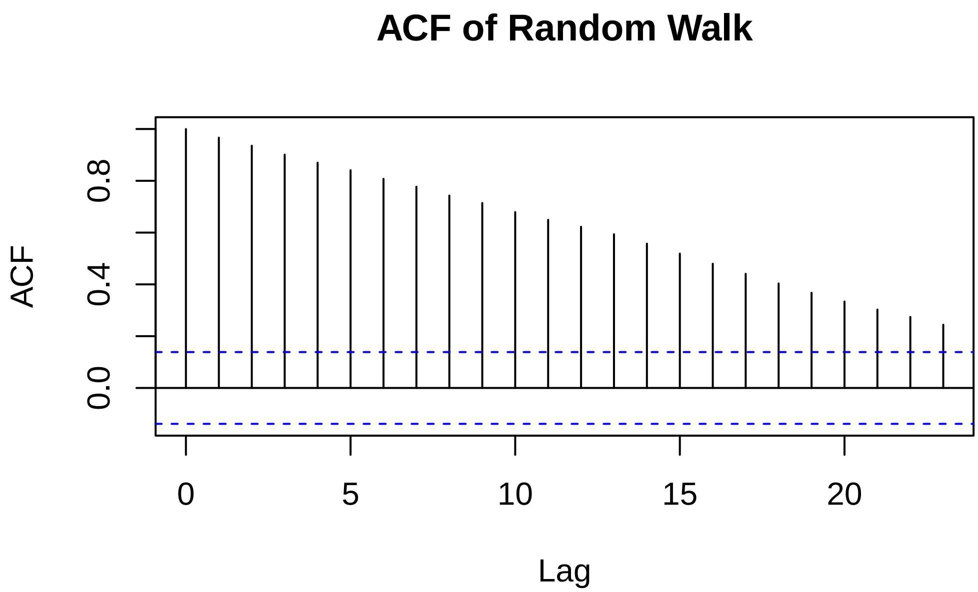
$$y_t = y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathrm{WN}(0,1)$$

- Generate a series of T = 200.
- Plot the series and compare its behavior with previous processes.
- Plot the ACF and explain why it differs from stationary processes.

```
# rw <- cumsum(rnorm(500))
set.seed(1)
rw <- numeric(T)
e <- rnorm(T)

for (t in 2:T) {
  rw[t] <- rw[t-1] + e[t]
}
# Plot
plot(rw, type = "l",
     main = "Random Walk Process", ylab = "Value", xlab = "Time")
```



```r
# ACF
acf(rw, main = "ACF of Random Walk")

# Dickey-Fuller test
library(urca)
```




```r
test <- ur.df(rw, type = "none", lags = 0)
summary(test)
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-2.20290 -0.56617 -0.01931  0.64375  2.42750

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
z.lag.1 -0.003654   0.007523  -0.486    0.628
```

```
Residual standard error: 0.9305 on 198 degrees of freedom
Multiple R-squared:  0.00119,   Adjusted R-squared:  -0.003854
F-statistic: 0.236 on 1 and 198 DF,  p-value: 0.6277


Value of test-statistic is: -0.4858

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Results

- White Noise: Mean close to 0 and variance of 1. Its ACF shows values close to 0 at all lags.
- AR(1): The series shows time dependence, with a slowly decaying ACF and a PACF with a cut-off at the first lag.
- AR(3): It shows oscillations in the series, with a more complex pattern in the ACF and breaks in the PACF in the first three lags.
- MA(1) and MA(5): ACF has significant values up to lag 1 in MA(1) and up to lag 5 in MA(5), while PACF gradually declines.
- ARMA(1,1) and ARMA(3,2): The combination of AR and MA generates a mixed pattern in ACF and PACF.
- Random Walk: ACF does not decay to 0, indicating non-stationarity. The ADF test confirms the presence of a unit root.

Proceso	FAC	FACp
Ruido Blanco	Sin picos, $\rho_s = 0$, $\forall s = 1, 2.$	Sin picos, $\theta_s = 0$, $\forall\, s = 1, 2, \ldots$
$AR(1)\ \phi_1 > 0$	Decrecimiento exponencial; $\rho_k = \phi^k$.	$\phi_{11} = \rho_1$; $\phi_k = 0$ para $k = 2, 3, \ldots$
$AR(1)\ \phi_1 < 0$	Decrecimiento exponencial; oscilatorio.	$\phi_{11} = \rho_1$; $\phi_k = 0$ para $k = 2, 3, \ldots$
$AR(p)$	Decrecimiento ($\rightarrow 0$) a partir del último parámetro (p) Antes, puede variar.	Picos hasta el último rezago válido (p), cero después.
$MA(q) : \theta > 0$	Pico en todos los rezagos hasta q; cero en los demás.	Decrecimiento geométrico a partir del rezago q.
$ARMA(p, q)$	Picos hasta $\max(p, q)$; decrecimiento exponencial después.	Decrecimiento geométrico a partir del rezago $\max(p, q)$.