Taller 1 - Simulación e identificación de procesos estocásticos

Series Cronológicas 2024

Marzo 2024

1. Simulación de procesos estocásticos

1.1. Random Walk sin drift

```r
# Función arima.sim
RW <- arima.sim(model = list(order = c(0, 1, 0)), n = 1500)

# Otra forma:
# Simulamos un Ruido Blanco
# RB <- rnorm (1500, 0, 1)
# head(RB)

# Obtenemos un Random Walk como la suma acumulada de shocks
# RW <- cumsum(RB)
# head(RW)

# Convertimos el proceso a formato ts
# RW <- ts(RW)
```

```r
# Graficamos el proceso
autoplot(RW) +
  geom_hline(yintercept = 0, col = "red") +
  labs(x = "Tiempo",
       y = "Valor")
```

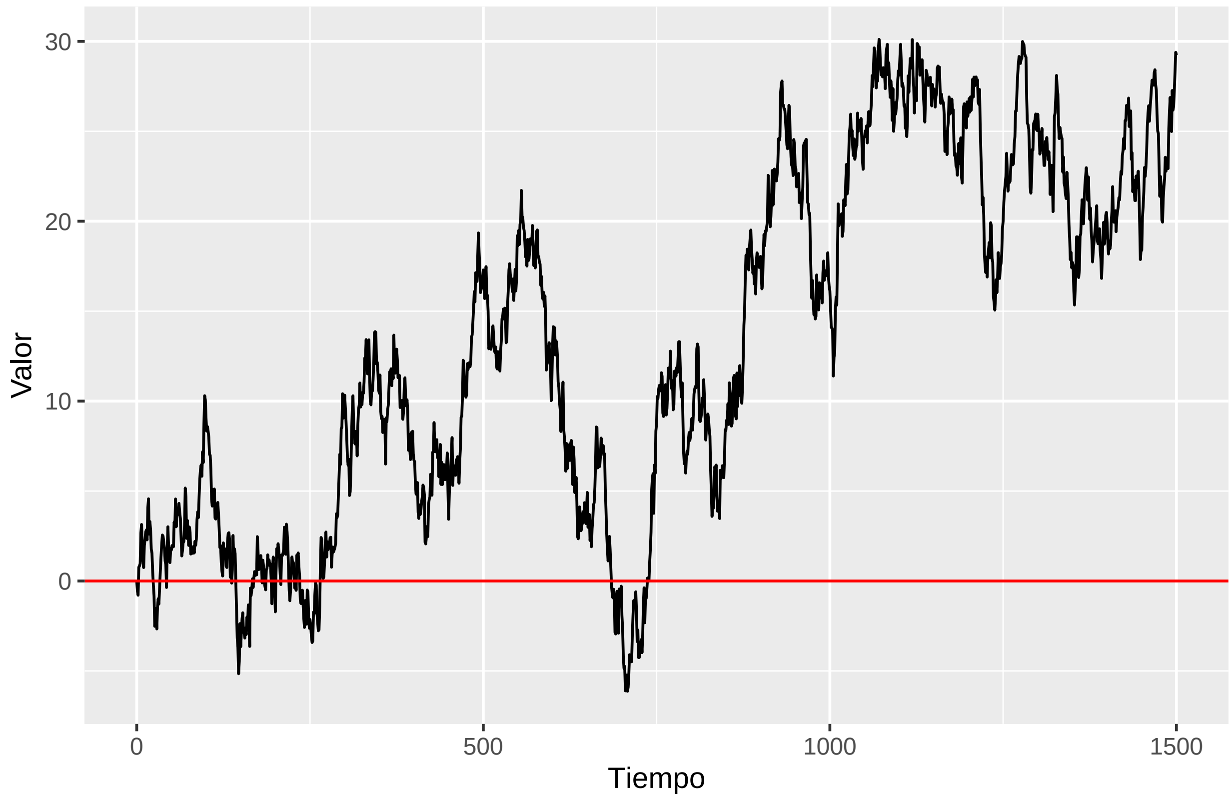


Figura 1: Simulación de un Random Walk.

```
# FAC
ggAcf(RW, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")
```

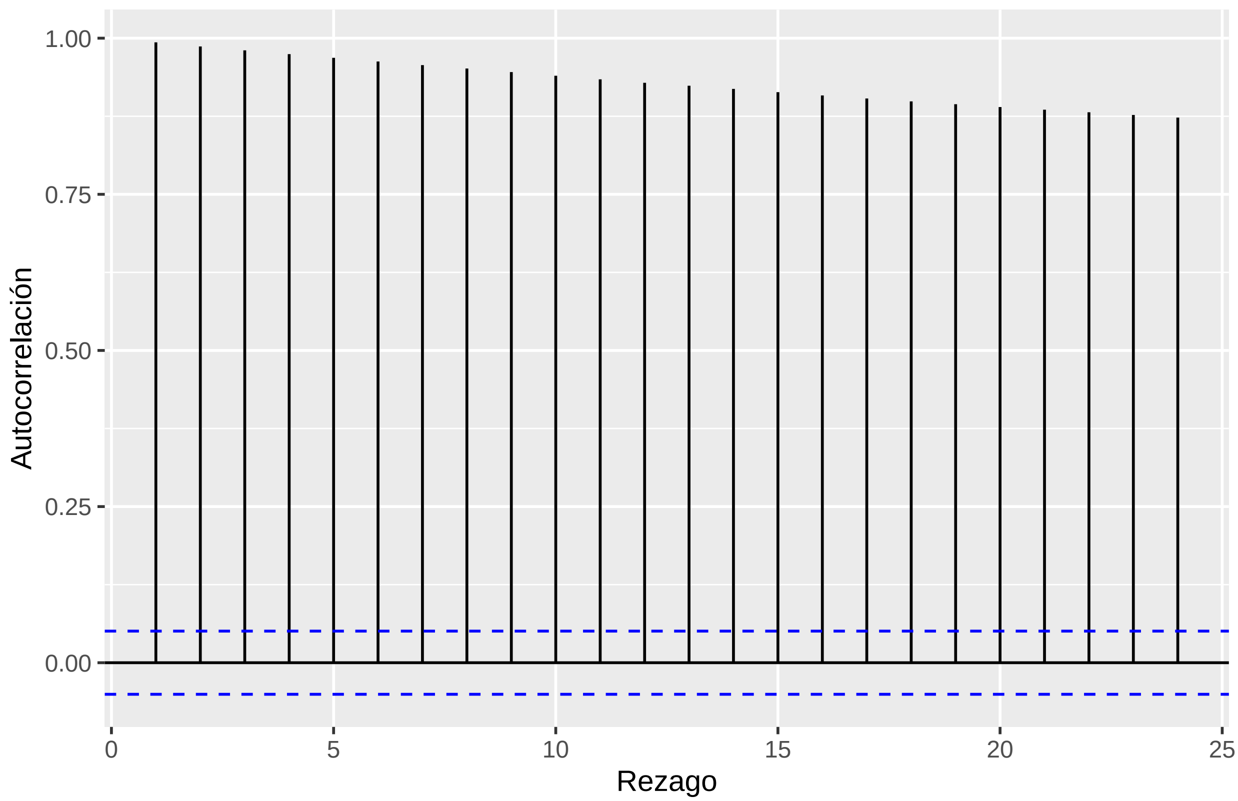


Figura 2: Función de Autocorrelación para un Random Walk.

1.2. Random Walk con drift

```
# Función arima.sim
RW_drift <- arima.sim(model = list(order = c(0, 1, 0)), n = 1500, mean = 0.2)

# Otra forma:
# Simulamos un Ruido Blanco
# RB <- rnorm (1500, 0, 1)
# head(RB)

# Obtenemos un Random Walk como la suma acumulada de shocks más el drift de 0.2
# delta <- 0.2
# RW_drift <- RB + delta
# RW_drift <- cumsum(RW_drift)
# head(RW_drift)

# Convertimos el proceso a formato ts
# RW_drift <- ts(RW_drift)
```

```
# Graficamos el proceso
autoplot(RW_drift) +
```

```r
  labs(x = "Tiempo",
       y = "Valor")
```

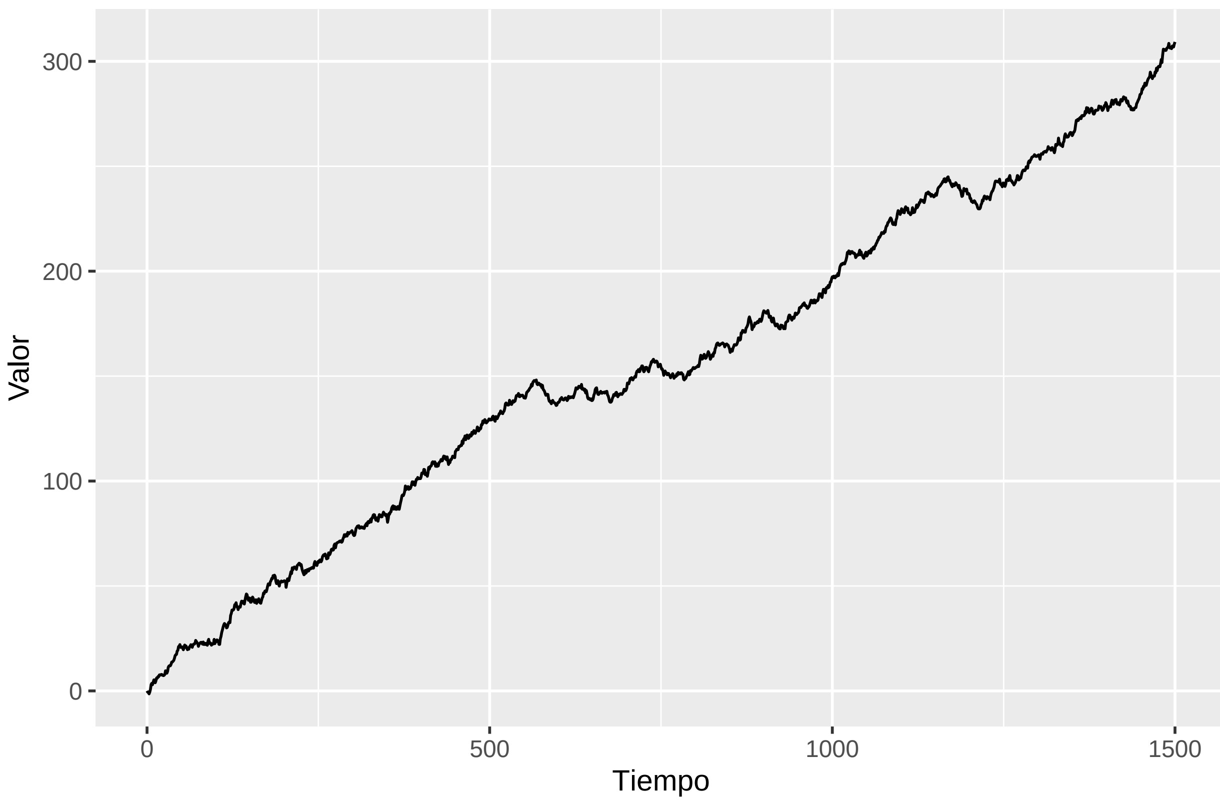


Figura 3: Simulación de un Random Walk con drift.

```r
# FAC
ggAcf(RW_drift, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")
```

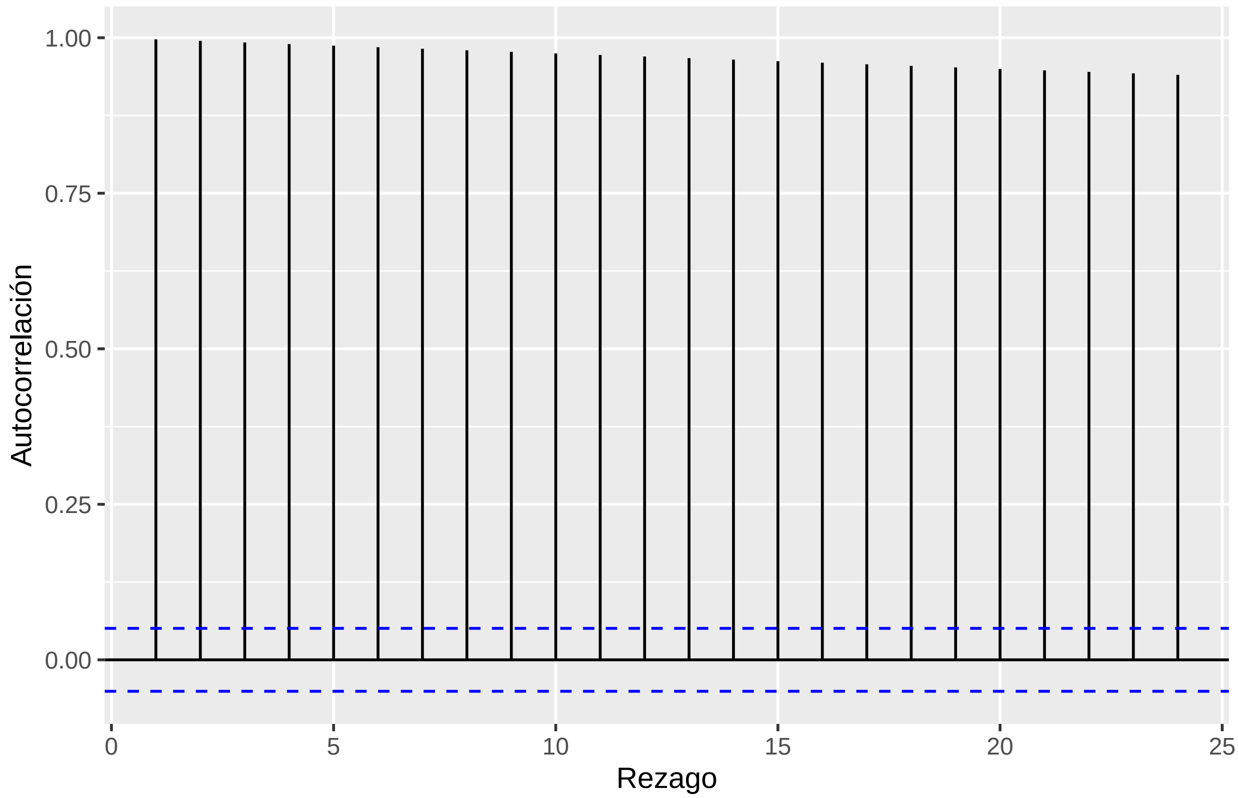


Figura 4: Función de Autocorrelación para un Random Walk con drift.

1.3. Proceso AR(2)

1.3.1. Polinomio característico: dos raíces reales

```
# Simulamos un proceso AR(2) con phi1 = 0,2 y phi2 = 0,6
simula_ar2_1 <- arima.sim(n = 1500, list(ar = c(0.2, 0.6)))

# Chequeamos estacionariedad
# La función Mod() obtiene el módulo y polyroot() las raíces del polinomio
polyroot(c(1, -0.2, -0.6)) # Dos raíces imaginarias
```

```
## [1]  1.135042+0i -1.468375+0i
```

```
Mod(polyroot(c(1, -0.2, -0.6))) # Módulo de las raíces fuera del círculo unitario
```

```
## [1] 1.135042 1.468375
```

```
# Graficamos el proceso
autoplot(simula_ar2_1) +
  geom_hline(yintercept = 0, col = "red") +
```

```
  labs(x = "Tiempo",
       y = "Valor")
```

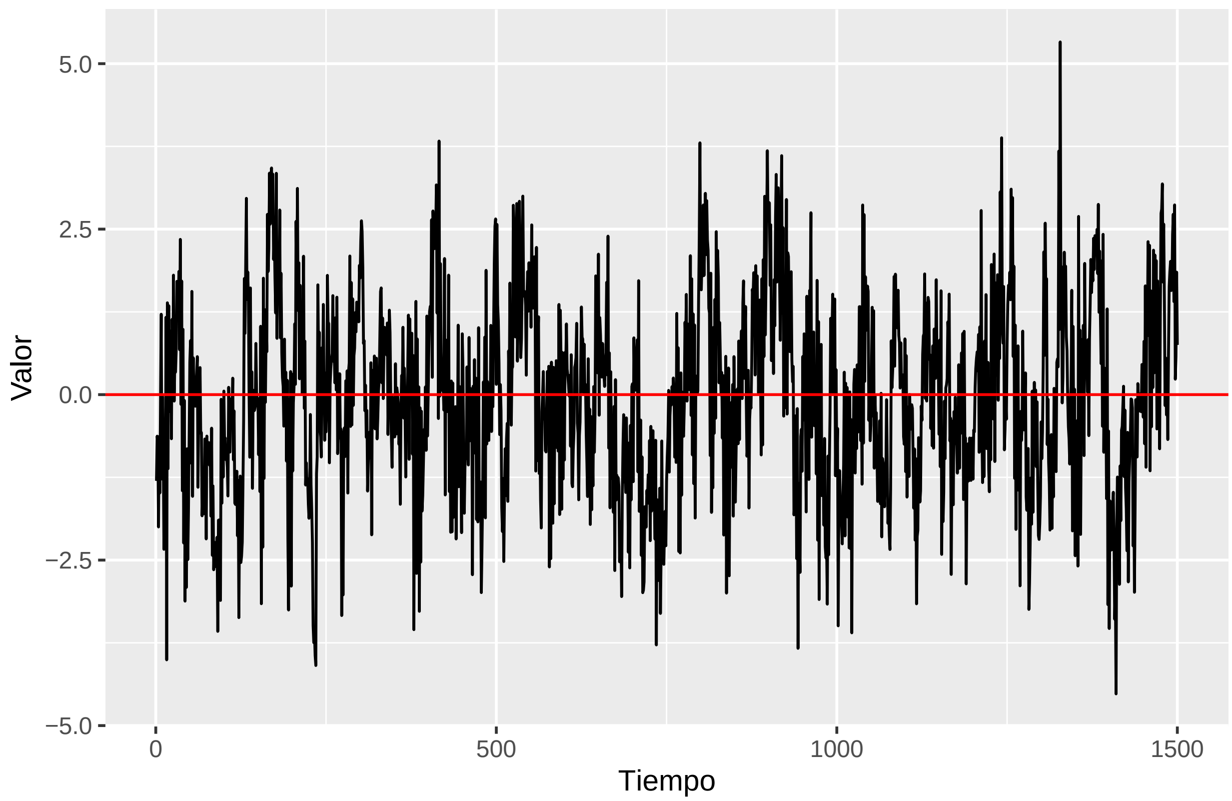
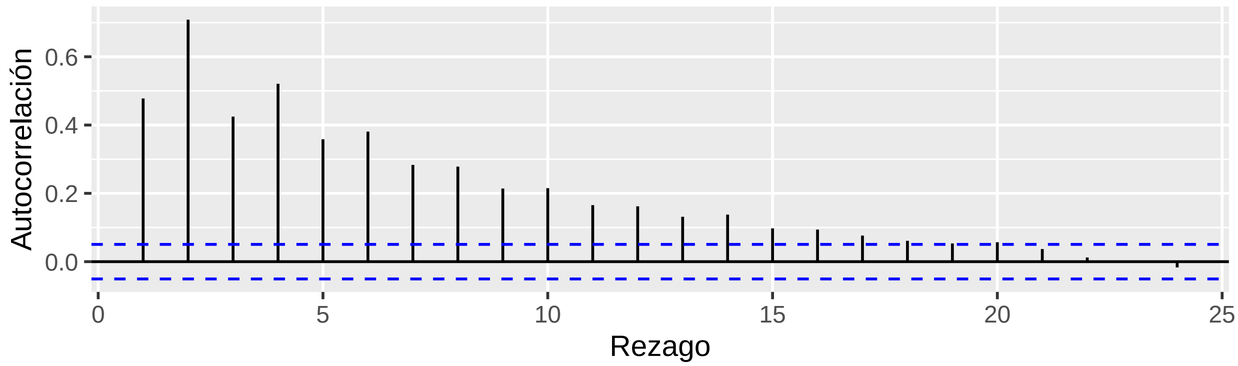


Figura 5: Simulación de un proceso AR(2) con coeficientes 0,2 y 0,6.

```
# FAC
acf_simula_ar2_1 <- ggAcf(simula_ar2_1, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
pacf_simula_ar2_1 <- ggAcf(simula_ar2_1, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(acf_simula_ar2_1, pacf_simula_ar2_1)
```



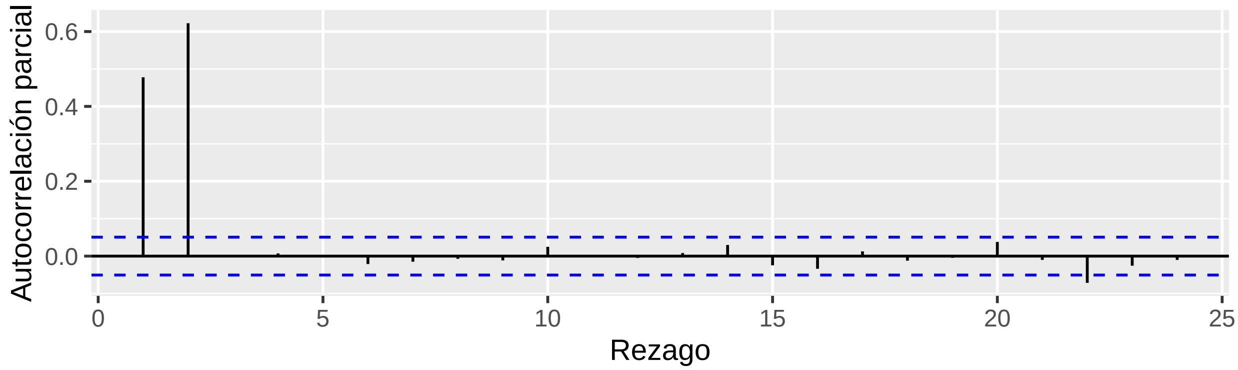


Figura 6: Funciones de Autocorrelación y Autocorrelación Parcial para un proceso AR(2) con coeficientes 0,2 y 0,6.

```r
# Obtenemos las autocorrelaciones teóricas
ar2_1_acf_teorico <- ARMAacf(ar = c(0.2, 0.6), lag.max = 14, pacf = FALSE)
ar2_1_pacf_teorico <- ARMAacf(ar = c(0.2, 0.6), lag.max = 14, pacf = TRUE)

# Obtenemos las autocorrelaciones estimadas
ar2_1_acf_est <- ggAcf(simula_ar2_1, plot = FALSE, lag.max = 14, type = "correlation")
ar2_1_pacf_est <- ggAcf(simula_ar2_1, plot = FALSE, lag.max = 14, type = "partial")

# Ordenamos los datos de la FAC para poder graficarlos
ar2_1_teorico_est_acf <- data.frame(rezago = 0:14, "Teórica" = ar2_1_acf_teorico, "Estimación" = ar2_1_a
ar2_1_teorico_est_acf <- ar2_1_teorico_est_acf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FAC teórica y estimada
ar2_1_acf <- ggplot(ar2_1_teorico_est_acf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación",
       fill = "FAC")

# Ordenamos los datos de la FACP para poder graficarlos
```

```r
ar2_1_teorico_est_pacf <- data.frame(rezago = 1:14, "Teórica" = ar2_1_pacf_teorico, "Estimación" = ar2_
ar2_1_teorico_est_pacf <- ar2_1_teorico_est_pacf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FACP teórica y estimada
ar2_1_pacf <- ggplot(ar2_1_teorico_est_pacf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       fill = "FACP")

grid.arrange(ar2_1_acf, ar2_1_pacf)
```

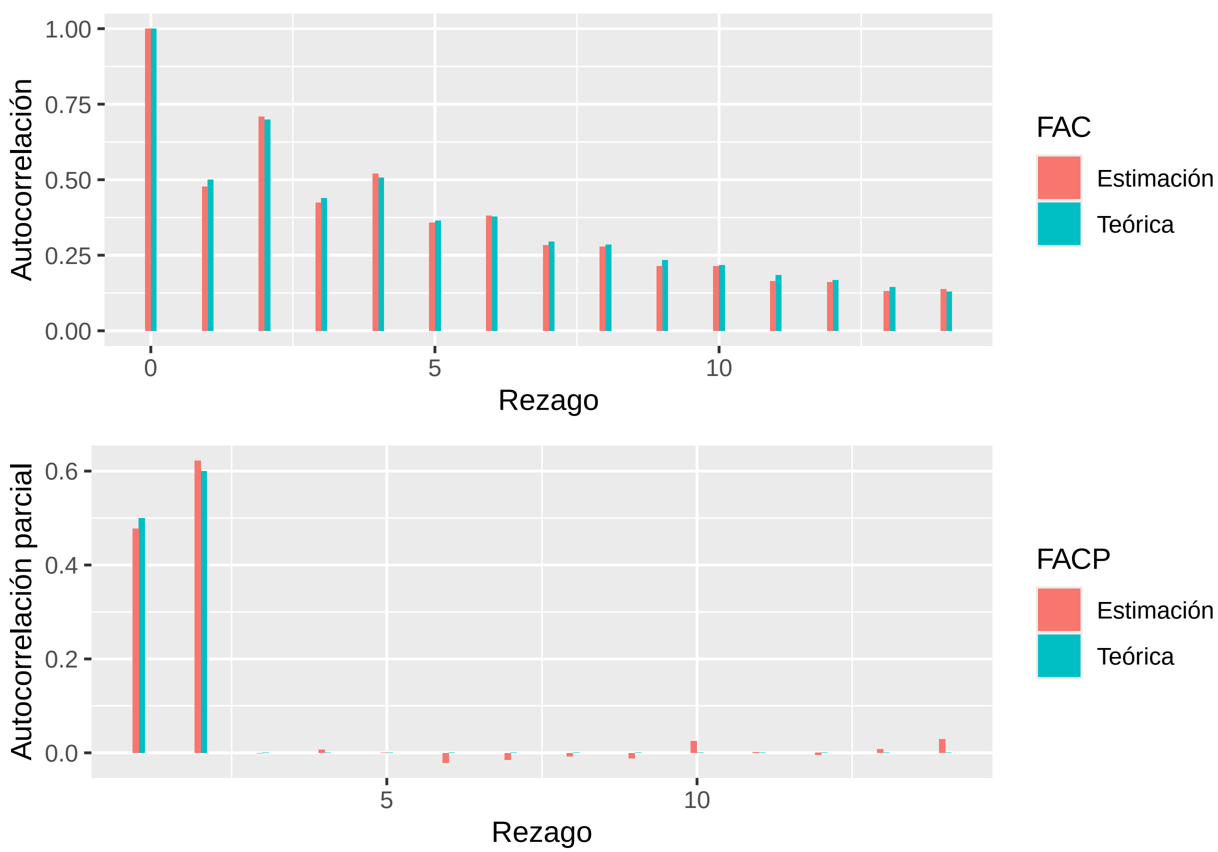


Figura 7: Funciones de Autocorrelación y Autocorrelación Parcial teóricas y estimadas para 1500 observaciones de un proceso AR(2) con coeficientes 0,2 y 0,6.

1.3.2. Polinomio característico: dos raíces imaginarias

```r
# Simulamos un proceso AR(2) con phi1 = 0,7 y phi2 = -0,5
simula_ar2_2 <- arima.sim(n = 1500, list(ar = c(0.7, -0.5)))

# Chequeamos estacionariedad
```

```r
# La función Mod() obtiene el módulo y polyroot() las raíces del polinomio
polyroot(c(1, -0.7, 0.5)) # Dos raíces imaginarias
```

```
## [1] 0.7+1.228821i 0.7-1.228821i
```

```r
Mod(polyroot(c(1, -0.7, 0.5))) # Módulo de las raíces fuera del círculo unitario
```

```
## [1] 1.414214 1.414214
```

```r
# Graficamos el proceso
autoplot(simula_ar2_2) +
  geom_hline(yintercept = 0, col = "red") +
  labs(x = "Tiempo",
       y = "Valor")
```

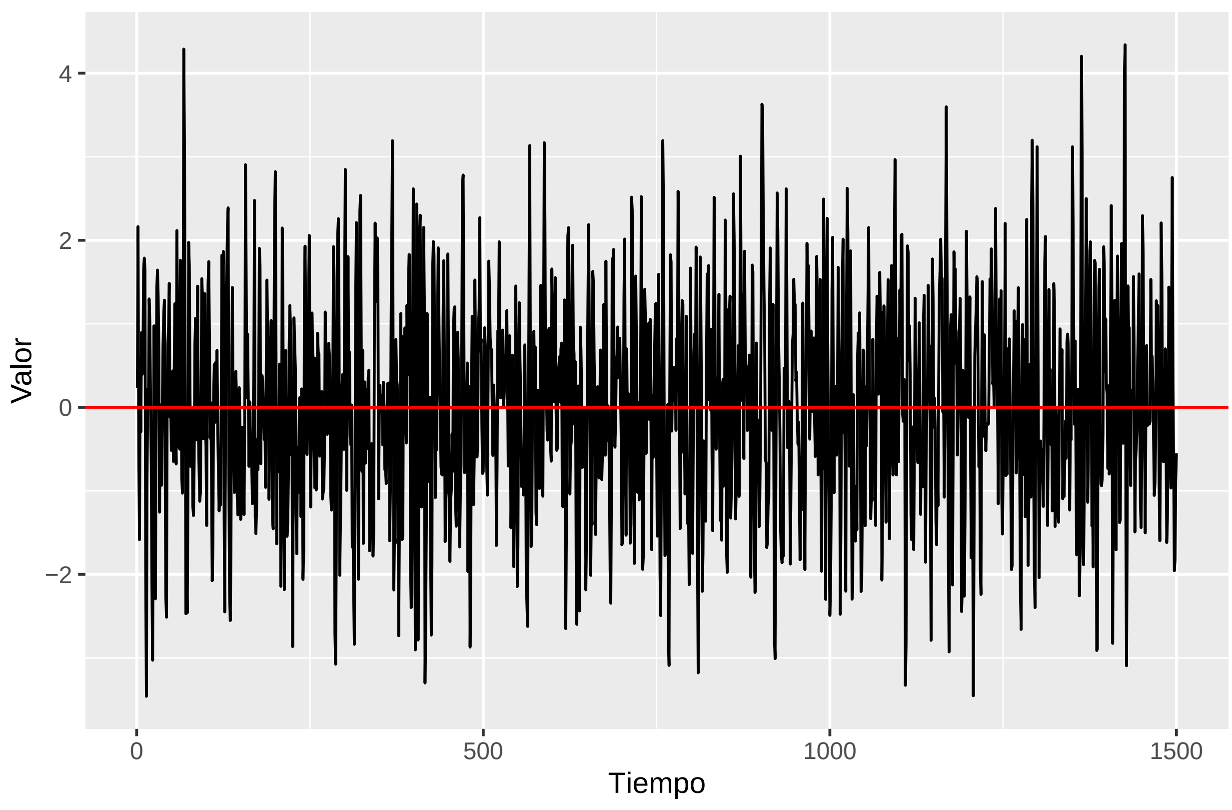


Figura 8: Simulación de un proceso AR(2) con coeficientes -0,7 y 0,5.

```r
# FAC
acf_simula_ar2_2 <- ggAcf(simula_ar2_2, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")
```

```r
# FACP
pacf_simula_ar2_2 <- ggAcf(simula_ar2_2, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(acf_simula_ar2_2, pacf_simula_ar2_2)
```

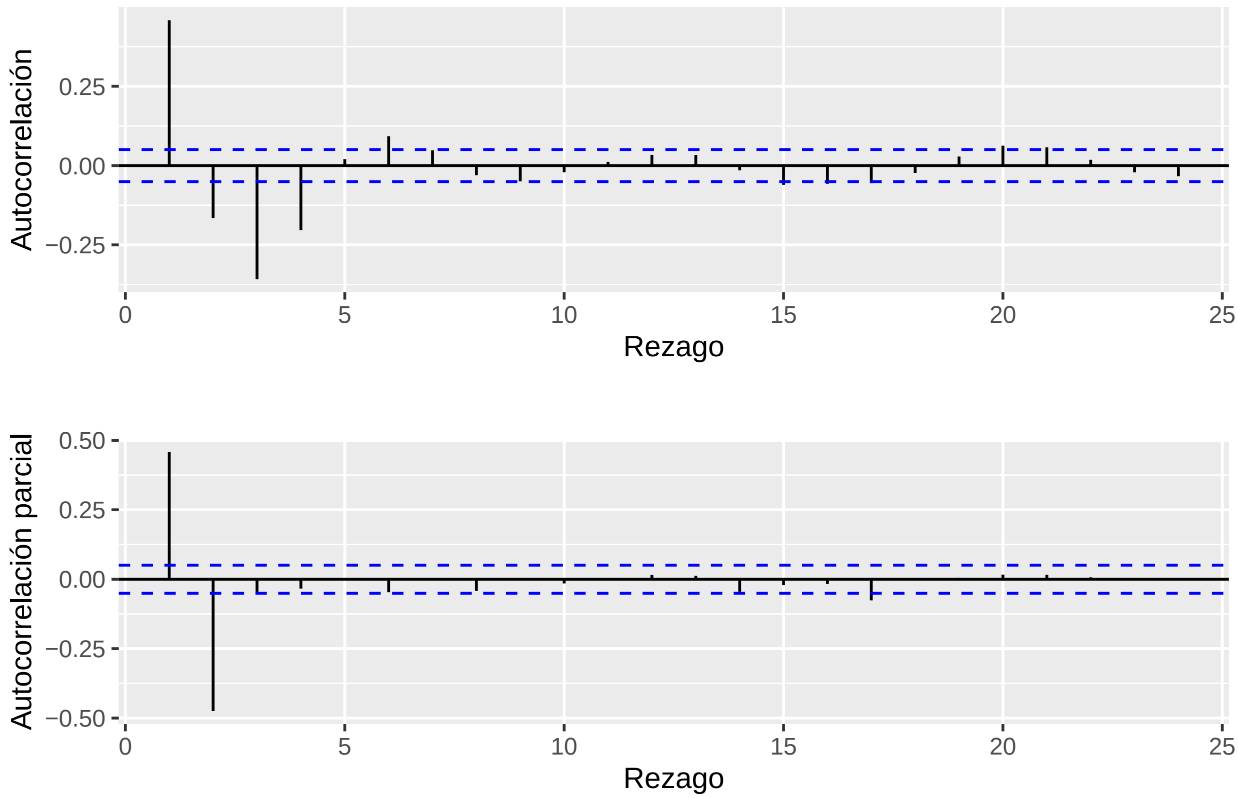


Figura 9: Funciones de Autocorrelación y Autocorrelación Parcial para un proceso AR(2) con coeficientes 0,7 y -0,5.

```r
# Obtenemos las autocorrelaciones teóricas
ar2_2_acf_teorico <- ARMAacf(ar = c(0.7, -0.5), lag.max = 14, pacf = FALSE)
ar2_2_pacf_teorico <- ARMAacf(ar = c(0.7, -0.5), lag.max = 14, pacf = TRUE)

# Obtenemos las autocorrelaciones estimadas
ar2_2_acf_est <- ggAcf(simula_ar2_2, plot = FALSE, lag.max = 14, type = "correlation")
ar2_2_pacf_est <- ggAcf(simula_ar2_2, plot = FALSE, lag.max = 14, type = "partial")

# Ordenamos los datos de la FAC para poder graficarlos
ar2_2_teorico_est_acf <- data.frame(rezago = 0:14, "Teórica" = ar2_2_acf_teorico, "Estimación" = ar2_2_a
ar2_2_teorico_est_acf <- ar2_2_teorico_est_acf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")
```

```r
# Graficamos las FAC teórica y estimada
ar2_2_acf <- ggplot(ar2_2_teorico_est_acf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación",
       fill = "FAC")

# Ordenamos los datos de la FACP para poder graficarlos
ar2_2_teorico_est_pacf <- data.frame(rezago = 1:14, "Teórica" = ar2_2_pacf_teorico, "Estimación" = ar2_2
ar2_2_teorico_est_pacf <- ar2_2_teorico_est_pacf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FACP teórica y estimada
ar2_2_pacf <- ggplot(ar2_2_teorico_est_pacf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       fill = "FACP")

grid.arrange(ar2_2_acf, ar2_2_pacf)
```

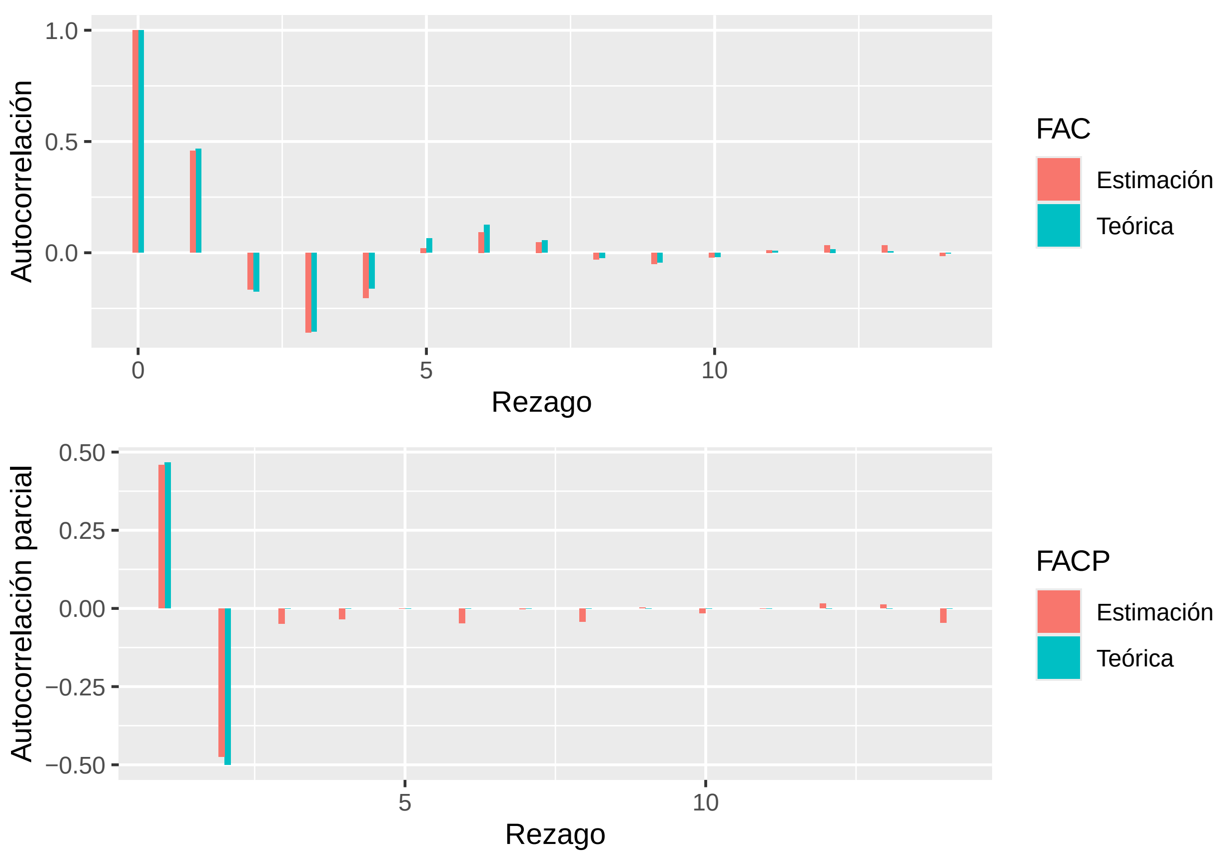


Figura 10: Funciones de Autocorrelación y Autocorrelación Parcial teóricas y estimadas para 1500 observaciones de un proceso AR(2) con coeficientes 0,7 y -0,5.

1.4. Procesos MA(2)

```
# Simulamos un proceso MA(2) con theta1 = 1,2 y theta2 = -0,7
# En la función arima.sim(), el signo del coeficiente está invertido
simula_ma2 <- arima.sim(list(ma = c(-1.2, 0.7)), n = 1500)
```

```
# Graficamos el proceso
autoplot(simula_ma2) +
  geom_hline(yintercept = 0, col = "red") +
  labs(x = "Tiempo",
       y = "Valor")
```

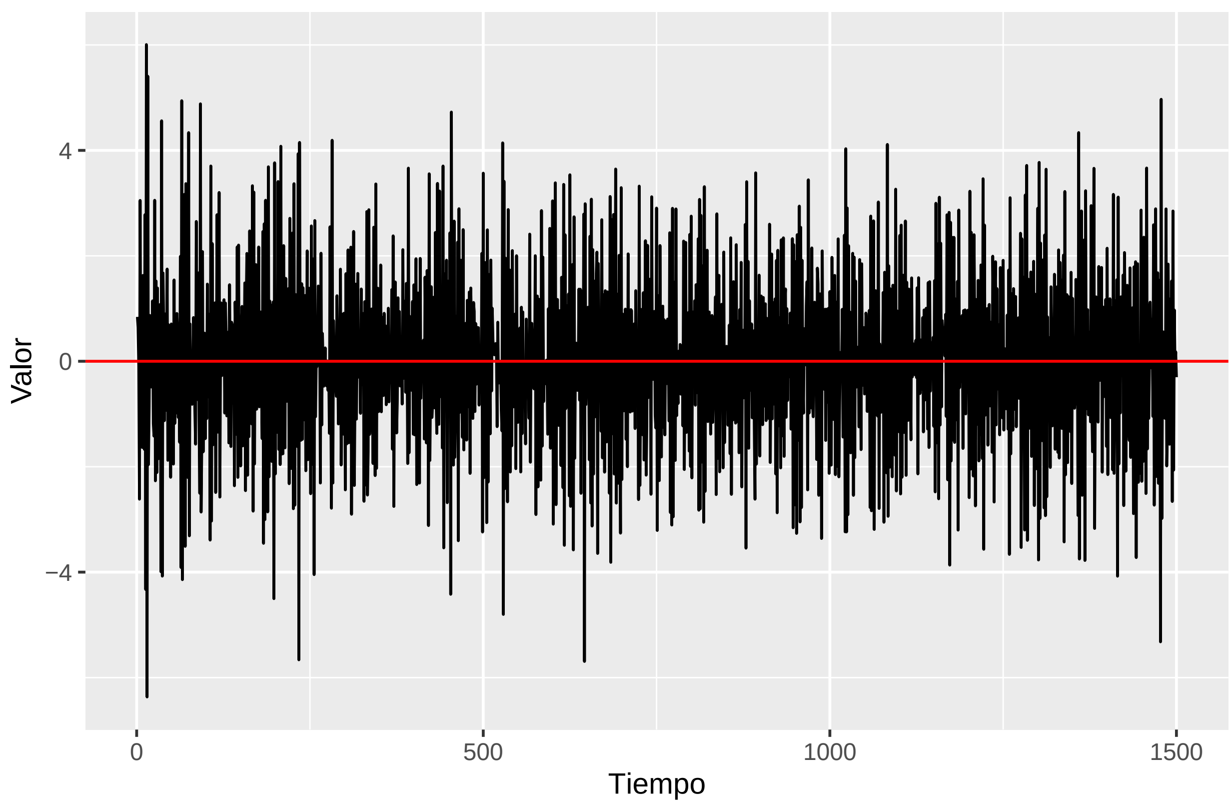


Figura 11: Simulación de un proceso MA(2) con coeficientes 1,2 y -0,7.

```
# FAC
acf_simula_ma2 <- ggAcf(simula_ma2, lag.max = 24, type = "correlation") +
  labs(x = "Rezago",
       y = "Autocorrelación",
       title = "")

# FACP
pacf_simula_ma2 <- ggAcf(simula_ma2, lag.max = 24, type = "partial") +
  labs(x = "Rezago",
```

```
       y = "Autocorrelación parcial",
       title = "")

grid.arrange(acf_simula_ma2, pacf_simula_ma2)
```

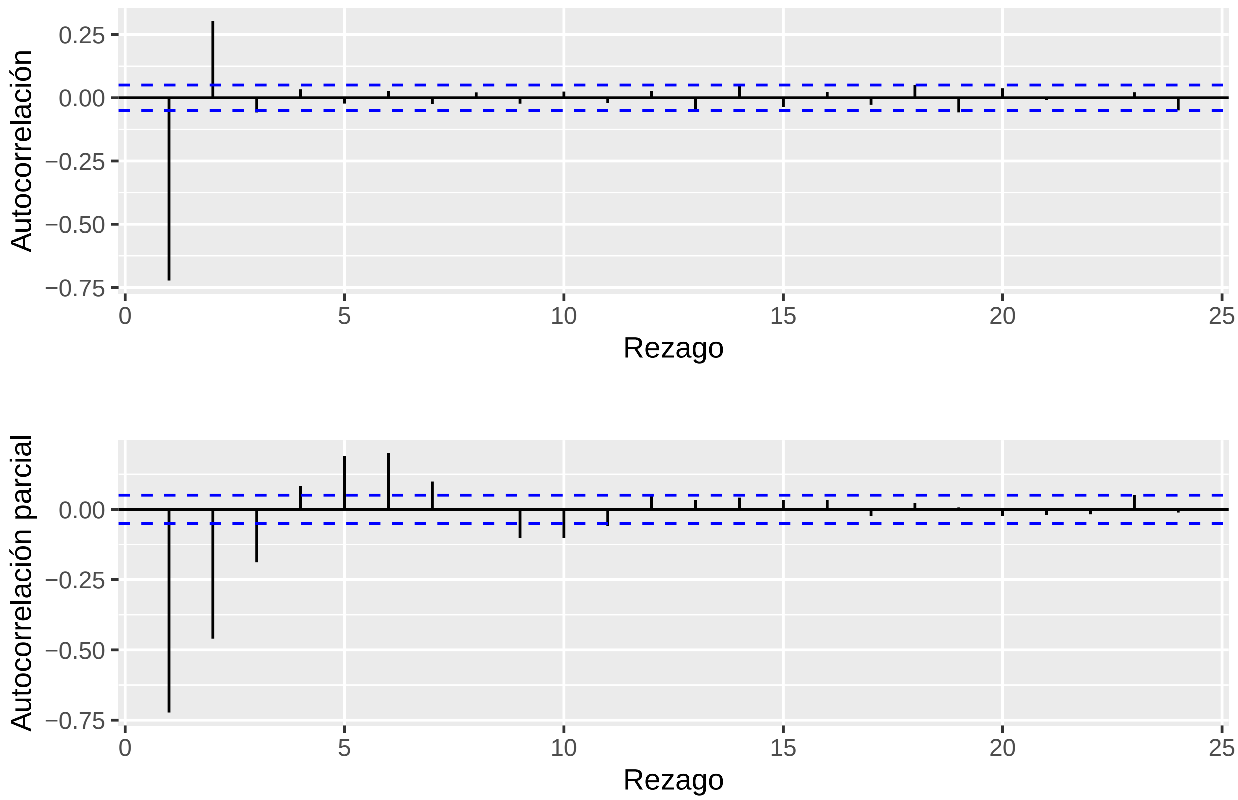


Figura 12: Funciones de Autocorrelación y Autocorrelación Parcial para un proceso MA(2) con coeficientes 1,2 y -0,7.

```
# Obtenemos las autocorrelaciones teóricas
ma2_acf_teorico <- ARMAacf(ma = c(-1.2, 0.7), lag.max = 14, pacf = FALSE)
ma2_pacf_teorico <- ARMAacf(ma = c(-1.2, 0.7), lag.max = 14, pacf = TRUE)

# Obtenemos las autocorrelaciones estimadas
ma2_acf_est <- ggAcf(simula_ma2, plot = FALSE, lag.max = 14, type = "correlation")
ma2_pacf_est <- ggAcf(simula_ma2, plot = FALSE, lag.max = 14, type = "partial")

# Ordenamos los datos de la FAC para poder graficarlos
ma2_teorico_est_acf <- data.frame(rezago = 0:14, "Teórica" = ma2_acf_teorico, "Estimación" = ma2_acf_es
ma2_teorico_est_acf <- ma2_teorico_est_acf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FAC teórica y estimada
ma2_acf <- ggplot(ma2_teorico_est_acf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
```

```
            position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación",
       fill = "FAC")

# Ordenamos los datos de la FACP para poder graficarlos
ma2_teorico_est_pacf <- data.frame(rezago = 1:14, "Teórica" = ma2_pacf_teorico, "Estimación" = ma2_pacf_
ma2_teorico_est_pacf <- ma2_teorico_est_pacf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FACP teórica y estimada
ma2_pacf <- ggplot(ma2_teorico_est_pacf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
            position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación parcial",
       fill = "FACP")

grid.arrange(ma2_acf, ma2_pacf)
```

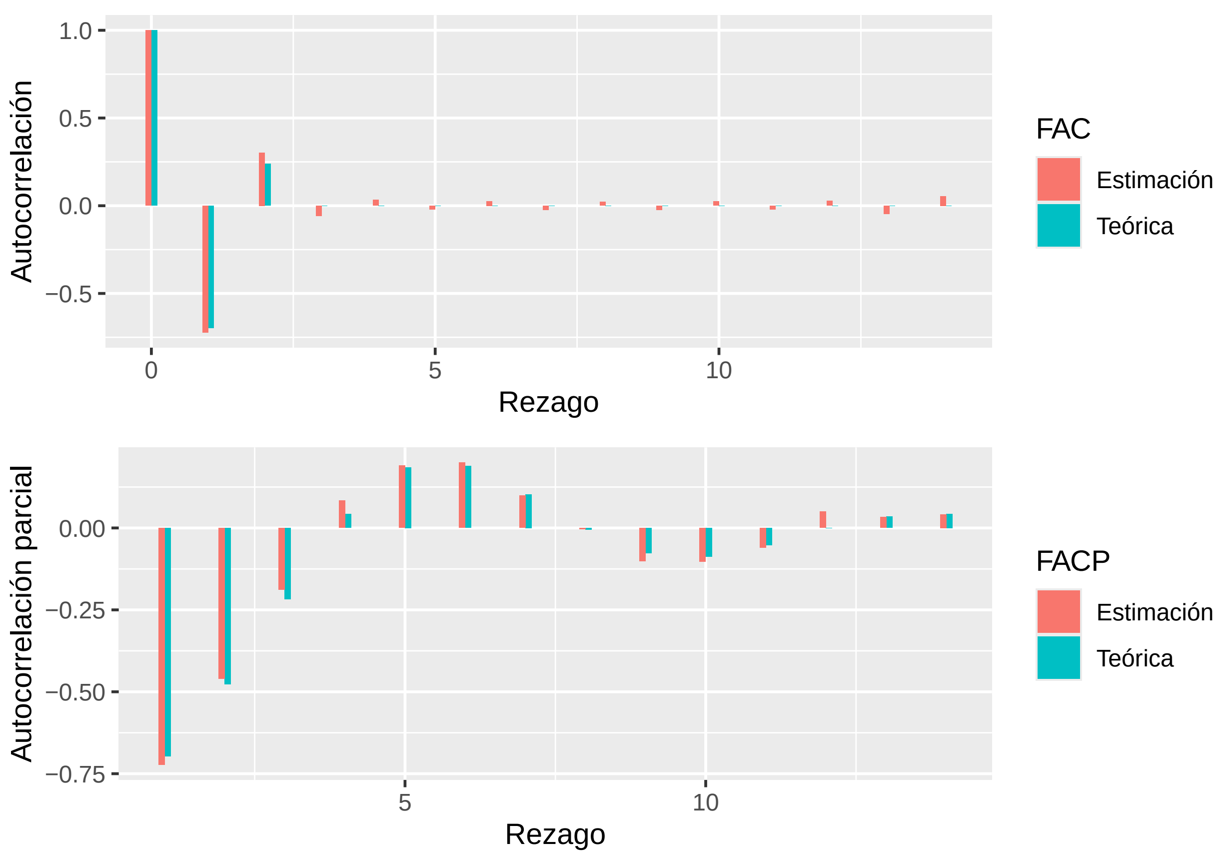


Figura 13: Funciones de Autocorrelación y Autocorrelación Parcial teóricas y estimadas para 1500 observaciones de un proceso MA(2) con coeficientes 1,2 y -0,7.

2. ARMA(p,q) como MA de orden infinito

```r
# Expresamos un proceso ARMA(2,1) como un MA(inf)
# En la función arima.sim(), el signo del coeficiente está invertido
ma_inf <- ARMAtoMA(ar = c(0.8, -0.15), ma = -0.3, lag.max = 40)
ma_inf <- data.frame(Tiempo = 1:40, Coeficientes = ma_inf)
head(ma_inf)
```

```
##   Tiempo Coeficientes
## 1      1     0.500000
## 2      2     0.250000
## 3      3     0.125000
## 4      4     0.062500
## 5      5     0.031250
## 6      6     0.015625
```

```r
# Graficamos los primeros 40 coeficientes del proceso expresado como un MA(inf)

ggplot(ma_inf) +
  geom_line(aes(x = Tiempo, y = Coeficientes))
```

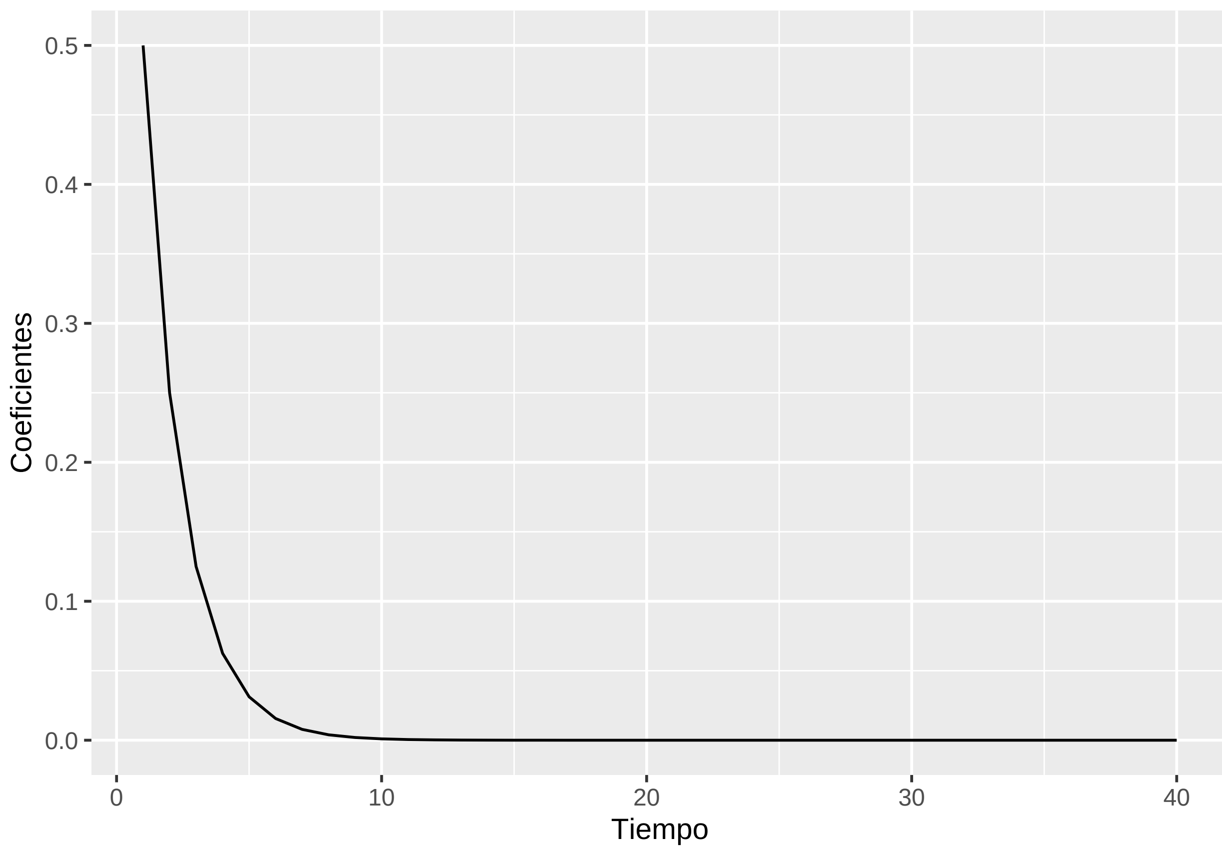


Figura 14: Primeros 40 coeficientes de la representación como un proceso MA de orden infinito de un proceso ARMA(1,2).