Taller 1 - Simulación y autocorrelaciones para un proceso AR(1)

Adaptado del taller 2022 de Federico Molina

Series Cronológicas 2024

Marzo 2024

1. Simulación de un procesos AR(1)

Para simular un proceso autorregresivo de orden 1, alcanza con definir un valor inicial, y_0, y simular una serie de Ruidos Blancos ε_t con $t = 1, \ldots, T$. Se llega entonces a que $Y_1 = \phi Y_0 + \varepsilon_1$, $Y_2 = \phi Y_1 + \varepsilon_2$, y así sucesivamente.

```
# Sea un valor inicial nulo: y_0 = 0
# Simulamos una serie de Ruidos Blancos con media 0 y desvío 1
epsilon <- ts(rnorm(1500, 0, 1))

# Simulamos un proceso autorregresivo con phi = 0,8
phi <- 0.8
y <- rep(0, 1500)
y[1] <- epsilon[1]
for (t in 2:length(y)) {
y[t] <- phi*y[t-1] + epsilon[t]}
y <- ts(y)

head(y)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1] -0.5604756 -0.6785580  1.0158619  0.8831979  0.8358461  2.3837418
```

```
# Forma no "manual"
# y <- arima.sim(list(ar = 0.8), n = 1500)
```

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

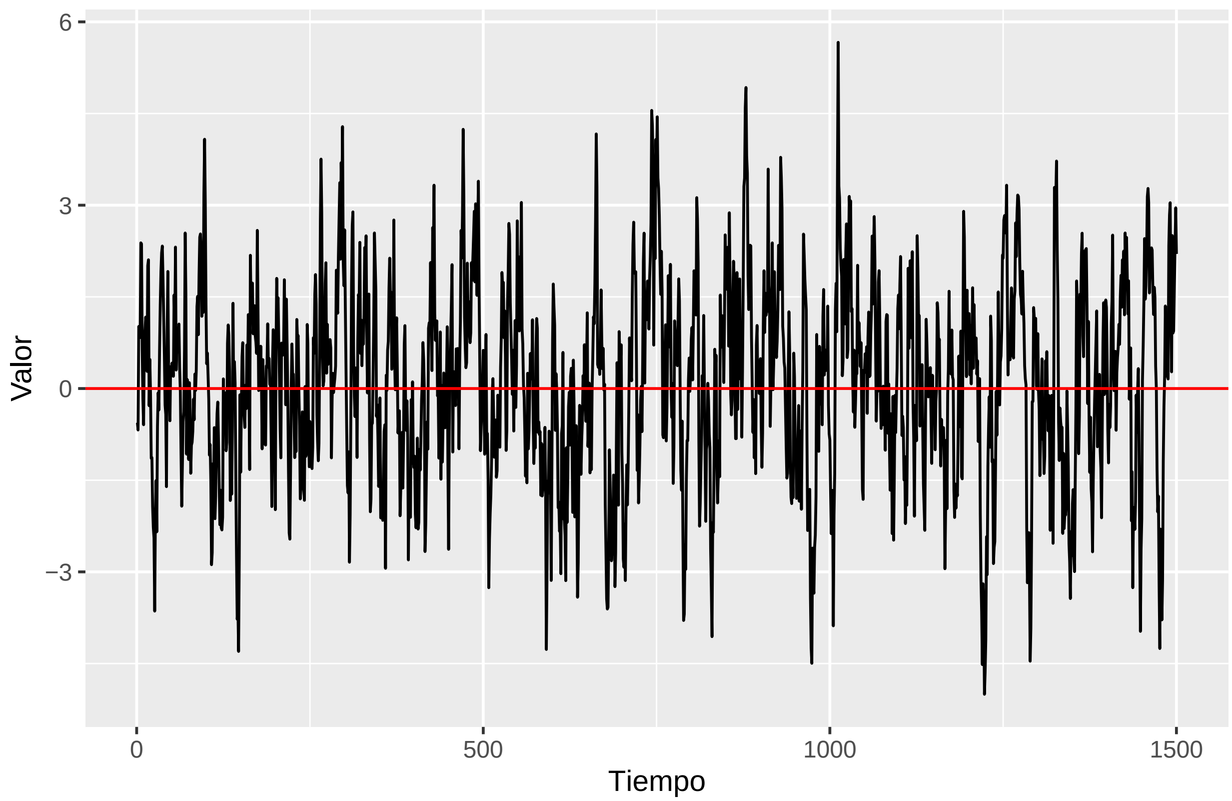

Figura 1: Simulación de un AR(1) con coeficiente igual a 0,8.

2. Función de Autocorrelación

2.1. Autocorrelaciones teóricas

Las autocorrelaciones teóricas de un proceso estocástico Y_t, $\rho_j, j = 1, 2, \ldots, k$, se calculan a partir de sus autocovarianzas, γ_j, (normalizadas mediante su varianza), de forma que:

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{Cov(Y_t, Y_{t-k})}{Var(Y_t)}$$

Un proceso AR(1) se define como:

$$Y_t = \phi Y_{t-1} + \varepsilon_t \Rightarrow (1 - \phi L)Y_t = \varepsilon_t$$

siendo ε_t un Ruido Blanco.

Si el proceso es estacionario ($|\phi| < 1$):

$$Y_t = \frac{\varepsilon_t}{(1 - \phi L)}$$

En este caso, se tiene que:

$$E(Y_t) = E\left(\frac{\varepsilon_t}{(1-\phi L)}\right) = 0$$

$$\gamma_0 = Var(Y_t) = E(Y_t^2) = E\left(\left(\sum_{t=1}^{\infty}\phi^j\varepsilon_{t-j}\right)^2\right) = E((\varepsilon_t + \phi\varepsilon_{t-1} + \phi^2\varepsilon_{t-2} + \ldots)(\varepsilon_t + \phi\varepsilon_{t-1} + \phi^2\varepsilon_{t-2} + \ldots))$$

$$= E(\varepsilon_t^2)+\phi^2E(\varepsilon_{t-1}^2)+\phi^4E(\varepsilon_{t-2}^2)+\ldots = \sigma_\varepsilon^2+\phi^2\sigma_\varepsilon^2+\phi^4\sigma_\varepsilon^2+\ldots = \sigma_\varepsilon^2(1+\phi^2+\phi^4+\ldots) = \sigma_\varepsilon^2\sum_{j=0}^{\infty}(\phi^2)^j = \sigma_\varepsilon^2\frac{1}{1-\phi^2}$$

$$\gamma_k = Cov(Y_t, Y_{t-k}) = E(Y_tY_{t-k}) = E\left(\left(\sum_{j=0}^{\infty}\varepsilon_{t-j}\right)\left(\sum_{j=0}^{\infty}\varepsilon_{t-k-j}\right)\right)$$

$$= E((\varepsilon_t + \phi\varepsilon_{t-1} + \phi^2\varepsilon_{t-2} + \ldots)(\varepsilon_{t-k} + \phi\varepsilon_{t-k-1} + \phi^2\varepsilon_{t-k-2} + \ldots) = \phi^kE(\varepsilon_{t-k}^2) + \phi^{k+2}E(\varepsilon_{t-k-1}^2) + \ldots$$

$$= \phi^k\sigma_\varepsilon^2 + \phi^{k+2}\sigma_\varepsilon^2 + \ldots = \sigma_\varepsilon^2(\phi^k + \phi^{k+2} + \ldots) = \sigma_\varepsilon^2\phi^k(1 + \phi^2 + \ldots) = \sigma_\varepsilon^2\phi^k\sum_{j=0}^{\infty}(\phi^2)^j = \sigma_\varepsilon^2\frac{\phi^k}{1-\phi^2} = \phi^k\gamma_0$$

De esta manera, para $k > 0$, se obtiene que $\rho_k = \phi^k$.

```
# Obtenemos la FAC teórica para los primero 20 rezagos
lag_max <- 20
phi <- 0.8
rho_teorica <- rep(1, lag_max + 1)
for (k in 2:length(rho_teorica)) {
rho_teorica[k] <- phi^(k-1)}
rho_teorica
```

```
##  [1] 1.00000000 0.80000000 0.64000000 0.51200000 0.40960000 0.32768000
##  [7] 0.26214400 0.20971520 0.16777216 0.13421773 0.10737418 0.08589935
## [13] 0.06871948 0.05497558 0.04398047 0.03518437 0.02814750 0.02251800
## [19] 0.01801440 0.01441152 0.01152922
```

```
# Forma no "manual"
# rho_teorica <- ARMAacf(ar = 0.8, lag.max = 20, pacf = FALSE)
```

```
# Graficamos la Función de Autocorrelación teórica
rho_teorica <- data.frame(rho = rho_teorica)

ggplot(rho_teorica) +
  geom_col(aes(x = 0:20, y = rho), width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación")
```

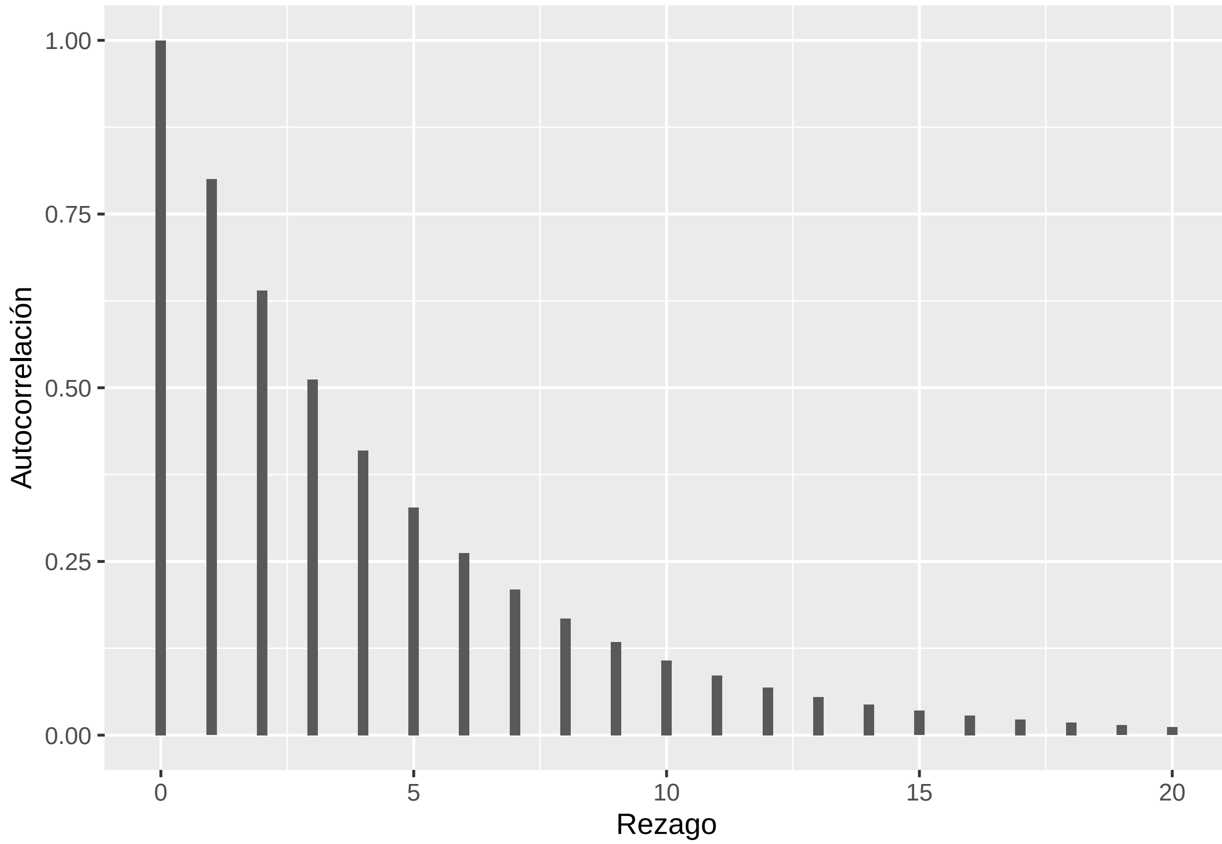

Figura 2: Función de Autocorrelación teórica de un AR(1) con coeficiente igual a 0,8.

2.2. Autocorrelaciones estimadas

Las autocorrelaciones estimadas para una serie de tiempo concreta, $\hat{\rho}_j, j = 1, 2, \ldots, k$, se obtienen mediante las correspondientes autocorrelaciones muestrales:

$$\hat{\rho}_k = \frac{\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

```
# Obtenemos las autocorrelaciones estimadas para 20 rezagos

lag_max <- 20 # Cantidad de rezagos
rho_est <- rep(0, lag_max) # Vector donde almacenamos las autocorrelaciones
suma_den <- sum((y-mean(y))^2) # Suma de Cuadrados Totales

# Calculamos las autocorrelaciones estimadas
for (k in 1:length(rho_est)) {
  suma_num <- 0 # Obtenemos el numerador de la fórmula para cada rho
  for (t in (k+1):length(y)) {
    suma_num <- suma_num + sum((y[t]-mean(y))*(y[t-k]-mean(y)))
  }
  rho_est[k] <- suma_num/suma_den # Calculamos las autocorrelaciones
  }
```

```r
rho_est <- c(1,rho_est) # Concatenamos un 1 correspondiente a rho_0
rho_est
```

```
##  [1]  1.000000000  0.785385749  0.616250504  0.488115867  0.383584569
##  [6]  0.300677463  0.231395059  0.177440481  0.135821911  0.096898732
## [11]  0.064427093  0.037624996  0.014085195  0.018864788  0.004426855
## [16] -0.011698899 -0.032536741 -0.041776113 -0.048331049 -0.047864701
## [21] -0.052522212
```

```r
# Forma no "manual"
# rho_est <- acf(y, lag.max = 20, type = "correlation", plot = FALSE)
# rho_est <- rho_est$acf
```

```r
# Graficamos la Función de Autocorrelación estimada
rho_est <- data.frame(rho = rho_est)

ggplot(rho_est) +
  geom_col(aes(x = 0:20, y = rho), width = 0.2, fill = "red") +
  labs(x = "Rezago",
       y = "Autocorrelación")
```

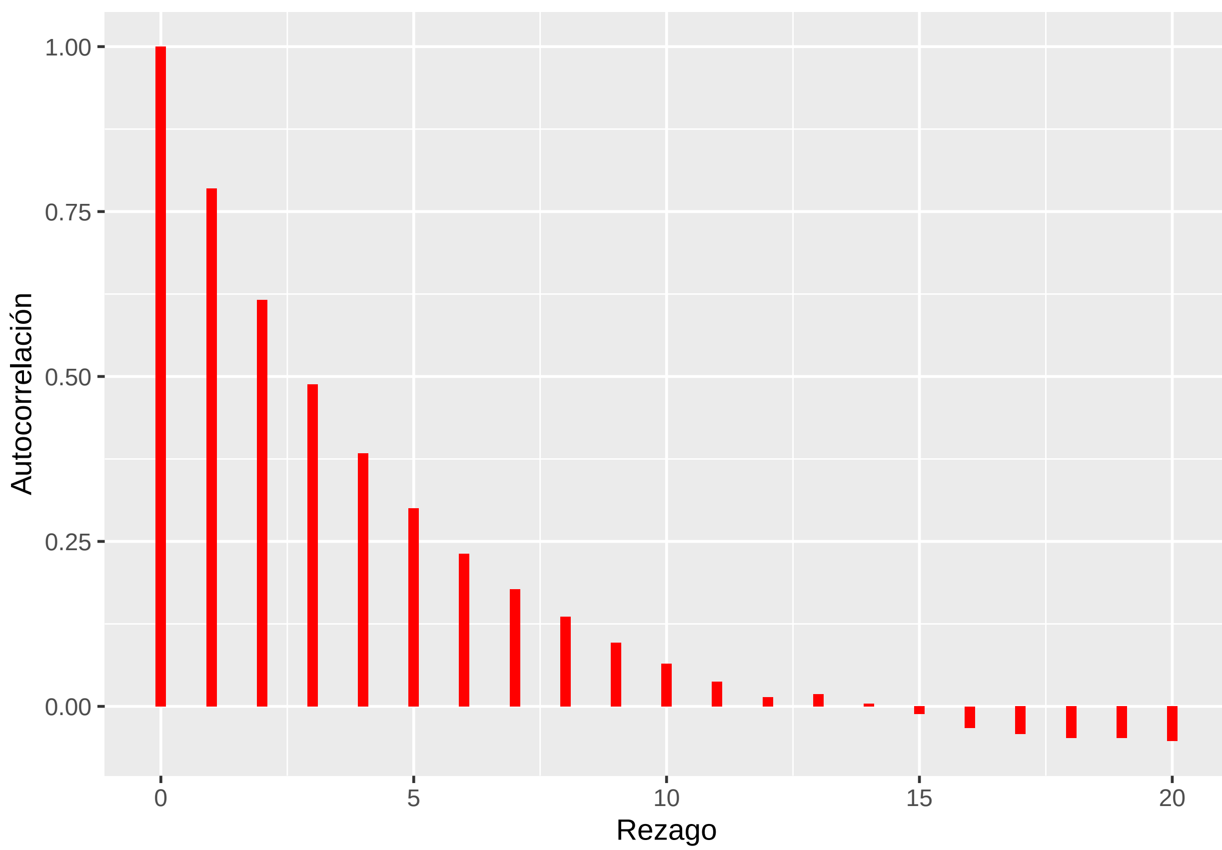

Figura 3: Función de Autocorrelación estimada de un AR(1) con coeficiente igual a 0,8.

```r
# Ordenamos los datos de la FAC para poder graficarlos
ar1_acf <- data.frame(rezago = 0:20, "Teórica" = rho_teorica$rho, "Estimación" = rho_est$rho)
ar1_acf <- ar1_acf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FAC teórica y estimada
ggplot(ar1_acf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación",
       fill = "FAC")
```

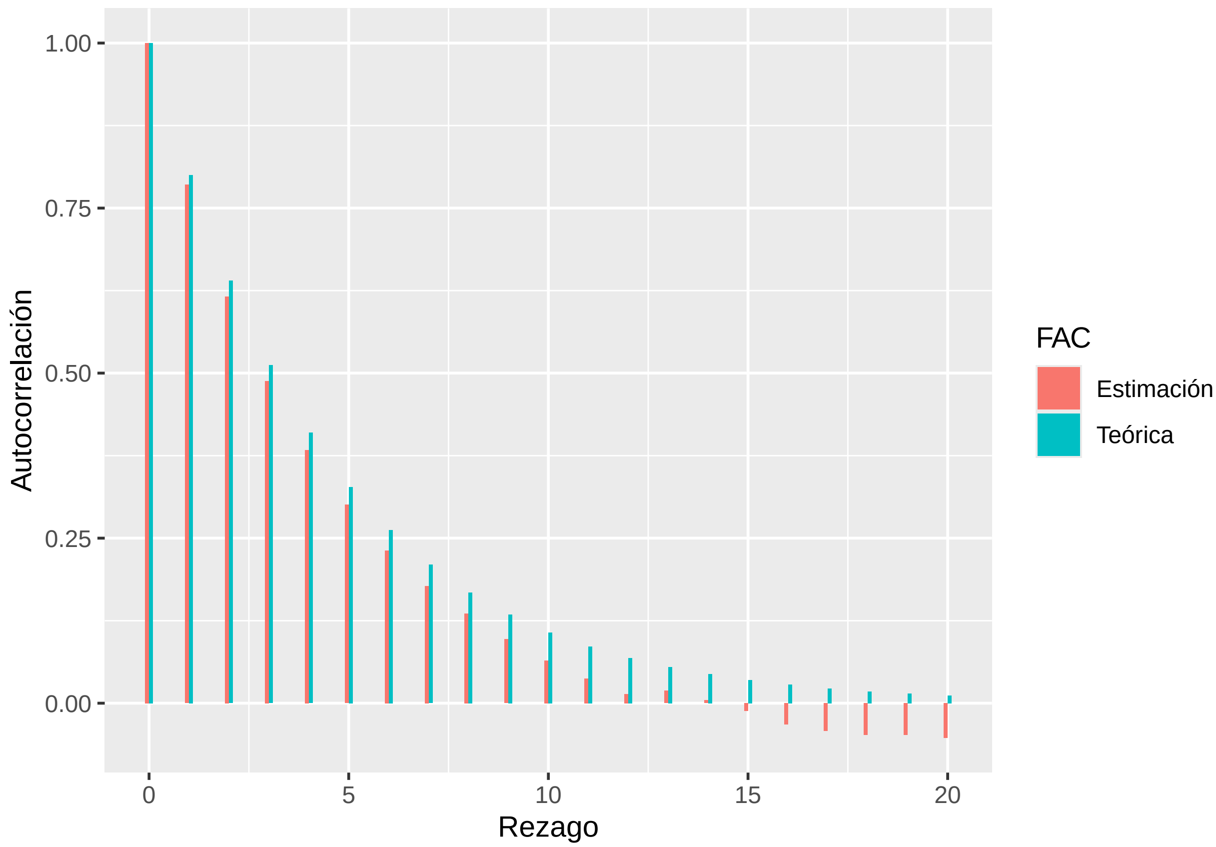

Figura 4: Función de Autocorrelación teórica y estimada para 1500 observaciones de un proceso AR(1) con coeficiente positivo.

3. Función de Autocorrelación Parcial

Las autocorrelaciones parciales, α_k, pueden interpretarse como el vínculo existente entre t y $t-k$, una vez "depuradas" todas las autocorrelaciones intermedias. Equivalen al k-ésimo coeficiente en una regresión lineal de Y_t sobre los primeros k rezagos. De esta manera, sea el modelo lineal:

$$Y_t = \phi_{k1} Y_{t-1} + \phi_{k2} Y_{t-2} + \ldots + \phi_{kk} Y_{t-k} + \varepsilon_t$$

La autocorrelación parcial de orden k será

$$\alpha_k = \phi_{kk}$$

. En el caso de un AR(1), α_1 será igual a ϕ.

```r
# Para un AR(1) con phi = 0,8, alpha_1 = 0.8
lag_max <- 20 # Cantidad de rezagos
alpha_teorica <- rep(0, lag_max) # Vector donde almacenamos las autocorrelaciones
alpha_teorica[1] <- 0.8 # La FACP teórica valdrá cero para todos los rezagos excepto el primero

# Forma no "manual"
# alpha_teorica <- ARMAacf(ar = 0.8, lag.max = 20, pacf = TRUE)
```

```r
# Graficamos la FACP teórica
alpha_teorica <- data.frame(alpha = alpha_teorica)

ggplot(alpha_teorica) +
  geom_col(aes(x = 1:20, y = alpha), width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación")
```

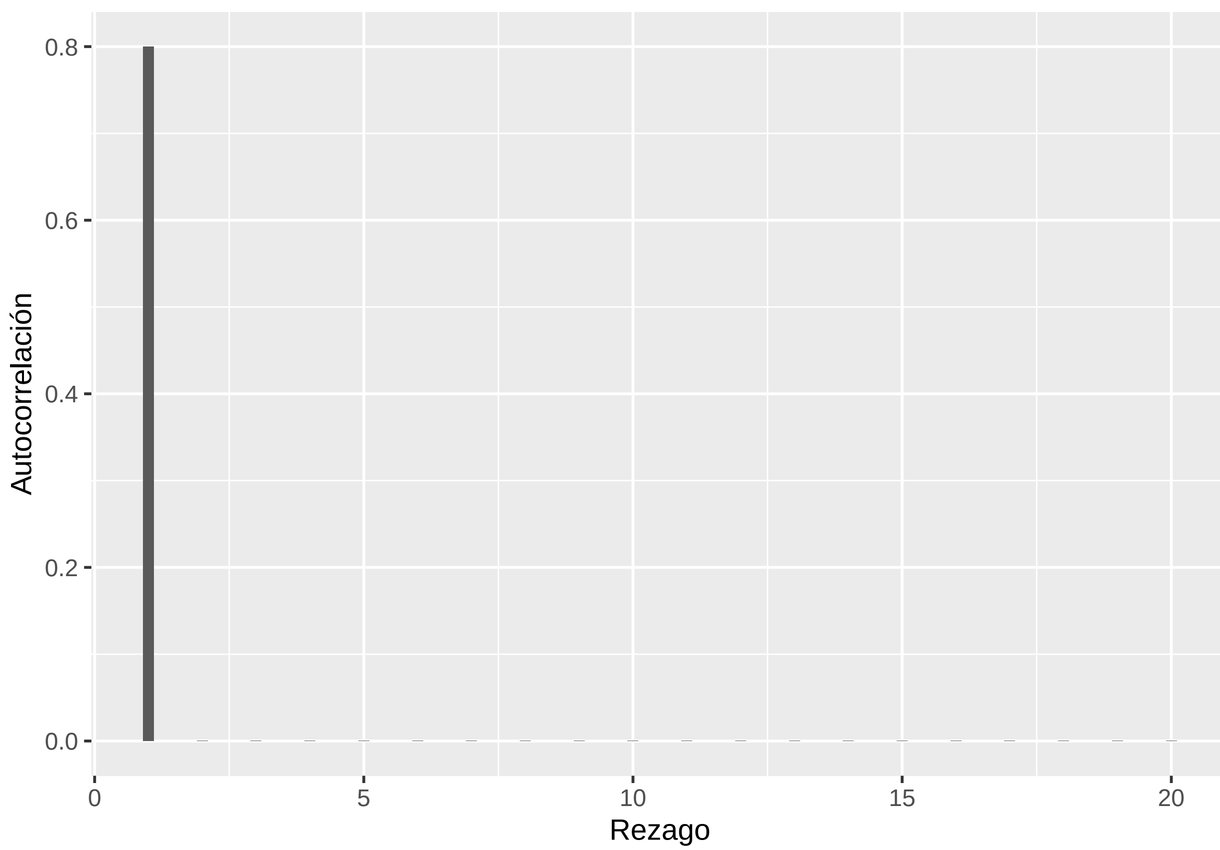

Figura 5: Función de Autocorrelación Parcial teórica de un AR(1) con coeficiente igual a 0,8.

Para una serie de tiempo, la función de autocorrelación parcial equivaldrá a las estimaciones MCO de estos coeficientes.

```r
# Obtenemos las autocorrelaciones parciales estimadas para 20 rezagos

lag_max <- 20 # Cantidad de rezagos

# Generamos un dataframe con la serie y sus rezagos
datos_y <- data.frame(y)

for (i in 1:lag_max) {
  datos_y <- cbind(datos_y, rep(NA, length(y)))
  for (j in (i+1):length(y)) {
    datos_y[j,i+1] <- datos_y[j-1,i]
  }
}

# Cambiamos el nombre de las columnas
colnames(datos_y) <- c("y", paste('lag', 1:lag_max, sep = "_"))

# Nos quedamos sin las filas con NAs
datos_y <- datos_y[lag_max + 1:nrow(datos_y),]

# Verificamos que los datos hayan quedado bien
head(datos_y)
```

```
##            y       lag_1       lag_2       lag_3       lag_4       lag_5
## 21 -1.141062 -0.09154766  0.47655468 -0.28100153  2.10701954  2.01146132
## 22 -1.130824 -1.14106184 -0.09154766  0.47655468 -0.28100153  2.10701954
## 23 -1.930664 -1.13082438 -1.14106184 -0.09154766  0.47655468 -0.28100153
## 24 -2.273422 -1.93066396 -1.13082438 -1.14106184 -0.09154766  0.47655468
## 25 -2.443777 -2.27342239 -1.93066396 -1.13082438 -1.14106184 -0.09154766
## 26 -3.641715 -2.44377718 -2.27342239 -1.93066396 -1.13082438 -1.14106184
##          lag_6      lag_7      lag_8     lag_9    lag_10     lag_11     lag_12
## 21  0.28068523  1.0456580  1.1687191 0.9599345 0.7501508 -0.5924137 -0.1834396
## 22  2.01146132  0.2806852  1.0456580 1.1687191 0.9599345  0.7501508 -0.5924137
## 23  2.10701954  2.0114613  0.2806852 1.0456580 1.1687191  0.9599345  0.7501508
## 24 -0.28100153  2.1070195  2.0114613 0.2806852 1.0456580  1.1687191  0.9599345
## 25  0.47655468 -0.2810015  2.1070195 2.0114613 0.2806852  1.0456580  1.1687191
## 26 -0.09154766  0.4765547 -0.2810015 2.1070195 2.0114613  0.2806852  1.0456580
##         lag_13     lag_14     lag_15     lag_16     lag_17    lag_18     lag_19
## 21  0.6292665  2.3679097  2.3837418  0.8358461  0.8831979 1.0158619 -0.6785580
## 22 -0.1834396  0.6292665  2.3679097  2.3837418  0.8358461 0.8831979  1.0158619
## 23 -0.5924137 -0.1834396  0.6292665  2.3679097  2.3837418 0.8358461  0.8831979
## 24  0.7501508 -0.5924137 -0.1834396  0.6292665  2.3679097 2.3837418  0.8358461
## 25  0.9599345  0.7501508 -0.5924137 -0.1834396  0.6292665 2.3679097  2.3837418
## 26  1.1687191  0.9599345  0.7501508 -0.5924137 -0.1834396 0.6292665  2.3679097
##         lag_20
## 21 -0.5604756
## 22 -0.6785580
## 23  1.0158619
## 24  0.8831979
## 25  0.8358461
## 26  2.3837418
```

```r
# Creamos un vector donde almacenamos las autocorrelaciones
alpha_est <- rep(0, lag_max)

# Para cada autocorrelación parcial, utilizamos el conjunto de datos sin NAs y estimamos una regresión
for (i in 1:lag_max) {
  alpha_est[i] <- lm(y ~ . - 1, data = datos_y[, 1:(i+1)])$coef[i]
}

# Corroboramos que las correlaciones hayan quedado bien calculadas
head(alpha_est)
```

```
## [1]  0.7882811113  0.0008217117  0.0093213188 -0.0064491613 -0.0034450466
## [6] -0.0107474142
```

```r
# Forma no "manual"
# alpha_est <- acf(y, lag.max = 20, type = "partial", plot = FALSE)
# alpha_est <- alpha_est$acf
```

```r
# Graficamos la FACP estimada
alpha_est <- data.frame(alpha = alpha_est)

ggplot(alpha_est) +
  geom_col(aes(x = 1:20, y = alpha), width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación")
```

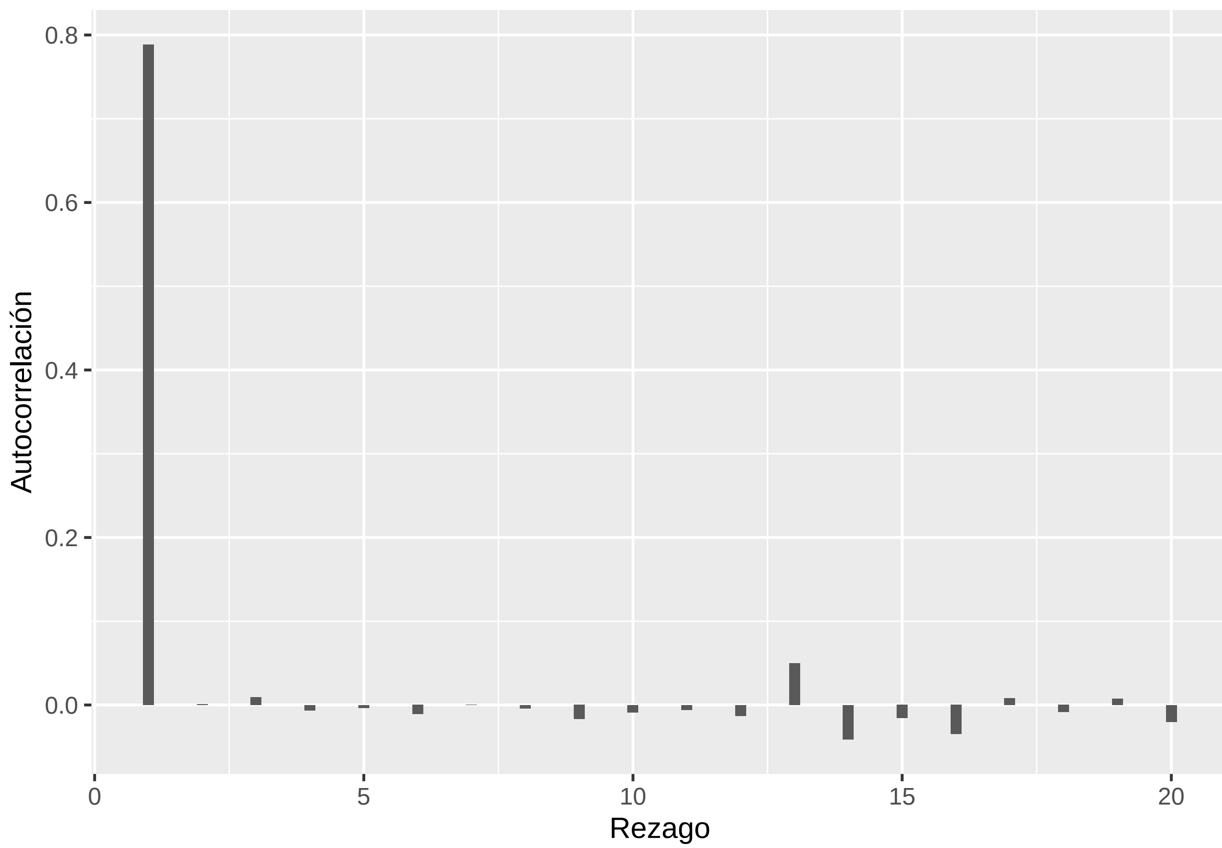

Figura 6: Función de Autocorrelación Parcial estimada de un AR(1) con coeficiente igual a 0,8.

```
# Ordenamos los datos de la FACP para poder graficarlos
ar1_pacf <- data.frame(rezago = 1:20, "Teórica" = alpha_teorica$alpha, "Estimación" = alpha_est$alpha)
ar1_pacf <- ar1_pacf %>%
  pivot_longer(cols = c("Teórica", "Estimación"), values_to = "valores")

# Graficamos las FACP teórica y estimada
ggplot(ar1_pacf) +
  geom_col(aes(x= rezago, y = valores, fill = name),
           position = "dodge", width = 0.2) +
  labs(x = "Rezago",
       y = "Autocorrelación",
       fill = "FACP")
```

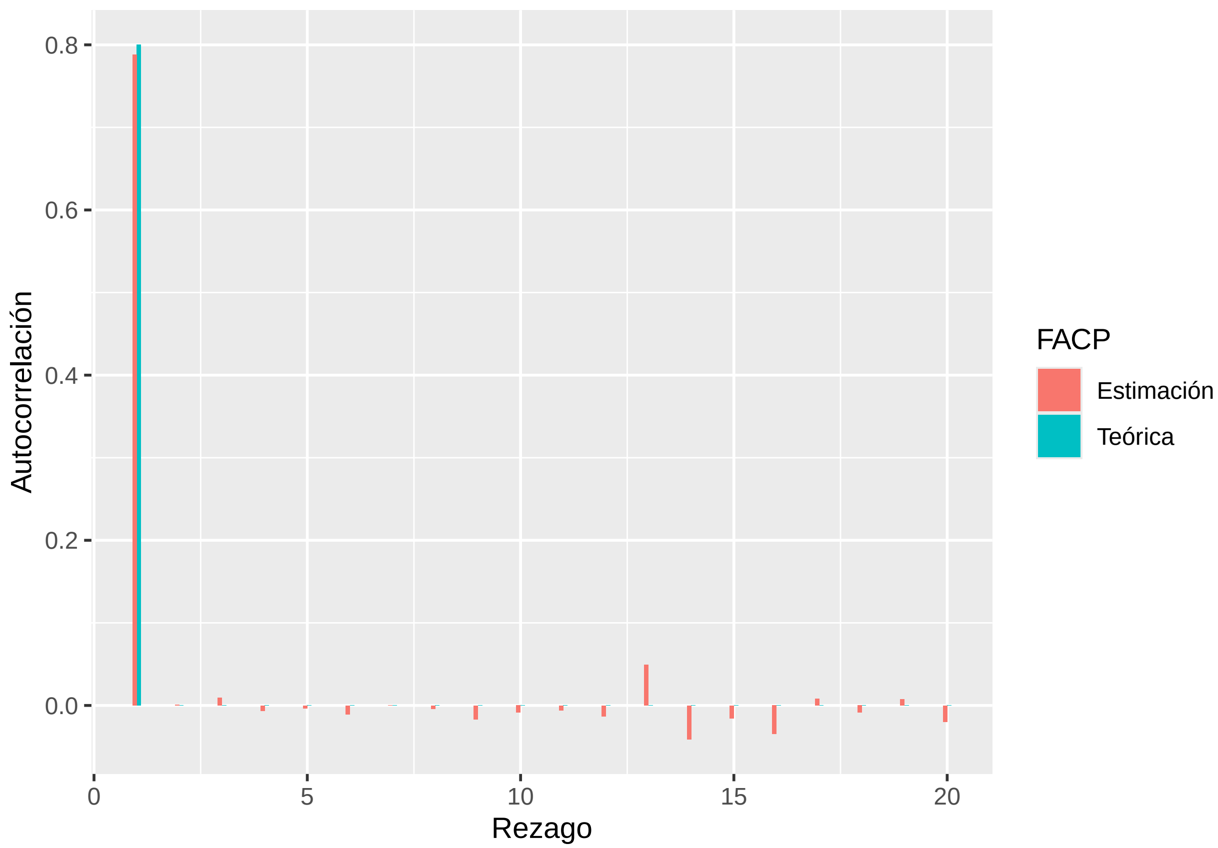

Figura 7: Función de Autocorrelación Parcial teórica y estimada para 1500 observaciones de un proceso AR(1) con coeficiente positivo.