Tarea 2 Propiedades de los modelos ARMA

Cuéllar, Eduardo, García Jesús, Miranda Areli, Ramirez José, Saldaña Ricardo

10/26/2021

1.- Considere el proceso $MA(2)$:

$$X_t = Z_t - 0.4Z_{t-1} - 1.2Z_{t-2}$$

donde Z_t es un ruido blanco Gaussiano.

(a) Calcule σ_X^2 suponiendo que $\sigma_Z^2 = 1$.
(b) Encuentre la expresión general para la función de autocorrelación ρ_k.
(c) Grafique ρ_k (correlograma ACF), para $k = 0, 1, 2, ..., 10$.
(d) Encuentre la expresión general para la funció de autocorrelación parcial ϕ_{kk}.
(e) Grafique ϕ_{kk} (correlograma PACF), para $k = 0, 1, 2, ..., 10$.
(f) En R simule el proceso X_t para un tamaño de muestra n, grafique la serie de tiempo y los correlogramas ACF y PACF. Compare los correlogramas simulados con los del proceso original.

```
#Cargamos librerías
library(ggplot2)
```

Respueta:

a) $Var(X_t) = Var(Z_t - 0.4_{t-1}, -1.2Z_{t-2})$ .... (1) Como $Z_k \perp Z_j \ \forall \ k \neq j$

Podemos expresar a (1) de la siguiente manera:

$$\begin{aligned} &= Var(Z_t) + Var(-0.4Z_{t-1}) + Var(-1.2Z_{t-2}) \\ &= Var(Z_t) + (-0.4)^2Var(Z_{t-1}) + (-1.2)^2Var(Z_{t-2}) ....(2) \end{aligned}$$

Como Z_t son $v.a.i.i.d.$, con $\mathbb{E}[Z_t] = 0$ y $Var(Z_t) = 1$

$$\begin{aligned} (2) &= 1 + (.16)(1) + (1.44)(1) \\ &= 1 + .16 + 1.44 \\ &= 2.6Cov(Z_t - 1.4Z_{t-1} - 1.2Z_{t-2}, Z_{t+k} - 0.4Z_{t-1+k} - Z_{t-2+k}) \end{aligned}$$

b) Veamos la autocovarianza:

$$\begin{aligned} \gamma(k) &= Cov(X_t, X_{t+k}) \\ &= Cov(Z_t - 0.4Z_{t-1} - 1.2Z_{t-2}, Z_{t+k} - 0.4Z_{t-1+k} - 1.2Z_{t-2+k}) \\ &= Cov(Z_t, Z_t + k) - 0.4Cov(Z_t, Z_{t+k-1}) - 1.2Cov(Z_t, Z_{t+k-2}) \\ &-0.4Cov(Z_{t-1}, Z_{t+k}) + .16Cov(Z_{t-1}, Z_{t+k-1}) + .48Cov(Z_{t-1}, Z_{t+k-2}) \\ &-1.2Cov(Z_{t-2}, Z_{t+k}) + .48Cov(Z_{t-2}, Z_{t+k_1}) + 1.44Cov(Z_{t-2}, Z_{t+k-2}) \end{aligned}$$

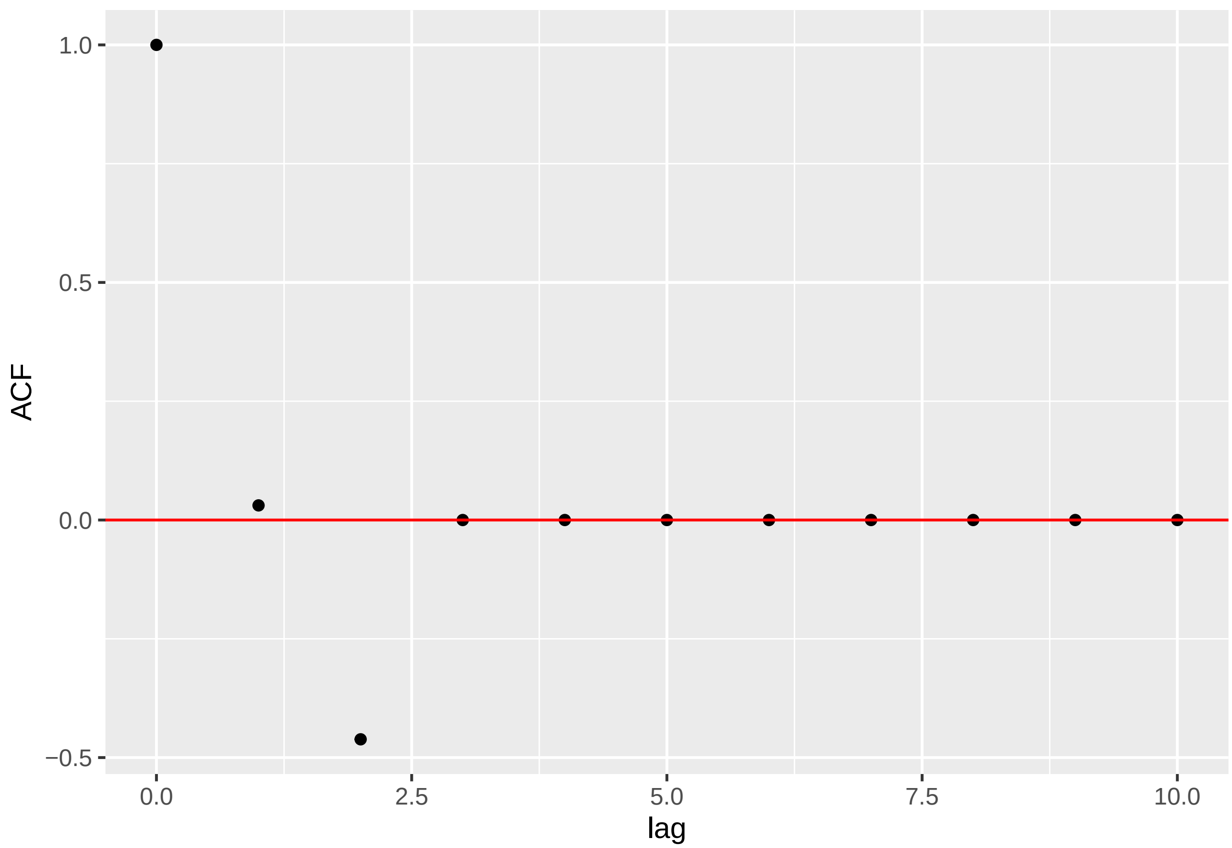
$$

Gráfica:

```r
acf_coefs_ej1=c(1,2/65,-6/13)
#llenemos de 0 los faltantes
for (i in (length(acf_coefs_ej1)+1):11){
  acf_coefs_ej1[i]=0
}
ACF_1<-data.frame('ACF'=acf_coefs_ej1,lag=0:10)
ACF_1
```

```
##            ACF lag
## 1   1.00000000   0
## 2   0.03076923   1
## 3  -0.46153846   2
## 4   0.00000000   3
## 5   0.00000000   4
## 6   0.00000000   5
## 7   0.00000000   6
## 8   0.00000000   7
## 9   0.00000000   8
## 10  0.00000000   9
## 11  0.00000000  10
```

```r
ggplot(ACF_1,aes(x=lag,y=ACF))+geom_point()+geom_abline(intercept = 0,slope=0,color='red')
```



Automatizaremos la obtención de los coeficientes del PACF para el ejercicio 1:

```r
#x será el vector con los coeficientes de autocorrelación
coefs_pacf<-function(p,k){
```

```
  if(k==0){
    return(1)
  }
  if(length(p)<k+1){
    for(i in length(p):k){
      p[i+1]=0
    }
  }
  A<-matrix(nrow=k,ncol = k)
  for (j in 1:k){
    for (i in 1:k){
       A[i,j]=p[abs(i-j)+1]
    }
  }
  B<-A
  for (i in 1:k){
    B[i,k]=p[i+1]
  }
  return(det(B)/det(A))
}
```

¿Cómo funciona? Bien, en clase, en la página 27 de las notas, podemos observar que ϕ_{kk}, que es el coeficiente de autocorrelación parcial para un lag de k se puede calcular usando Cramer. Observamos el patrón de que en la matriz que 'va en el denominador', iba el coeficiente ρ_i donde i era el valor absoluto de la diferencia entre el número de columna y renglón, por ello es que llenamos la matriz como $A[i, j] = p[abs(i - j) + 1]$. En la matriz 'numerador', únicamente es cambiar el último renglón por los ρ_j donde j es el número de renglón, siendo ambas matrices de dimensión $k * k$

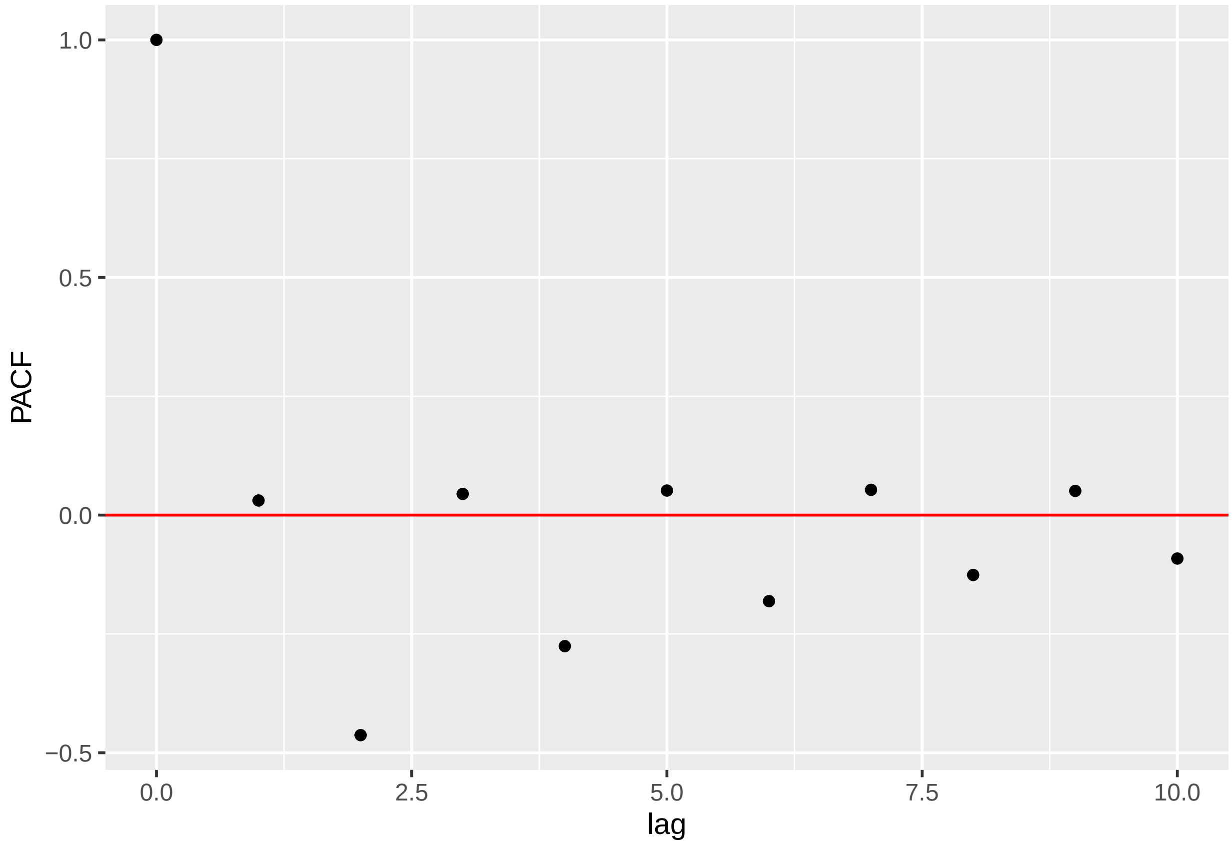
Ahora solo aplicamos la fórmula:

```
pacf_ej1=c()
for (i in 1:11){
  pacf_ej1[i]<-coefs_pacf(acf_coefs_ej1,i-1)
}
PACF_1=data.frame(PACF=pacf_ej1,lag=c(0:10))
PACF_1
```

```
##           PACF lag
## 1   1.00000000   0
## 2   0.03076923   1
## 3  -0.46292348   2
## 4   0.04461263   3
## 5  -0.27566723   4
## 6   0.05170474   5
## 7  -0.18101857   6
## 8   0.05318550   7
## 9  -0.12591880   8
## 10  0.05090694   9
## 11 -0.09137468  10
```

Graficamos

```r
ggplot(PACF_1,aes(x=lag,y=PACF))+geom_point()+geom_abline(intercept = 0,slope=0,color='red')
```



##Ejercicio 2

Programamos la función recursiva para los coeficientes de correlación

```r
#phi es un vector con los coeficientes del modelo AR(p)
#p es un vector con los acf para lag=1 y 2
coefs_acf<-function(phi,p,k){
  for (i in length(p):k+1){
    p[i]=phi[1]*p[i-1]+phi[2]*p[i-2]
  }
  return(p)
}
```

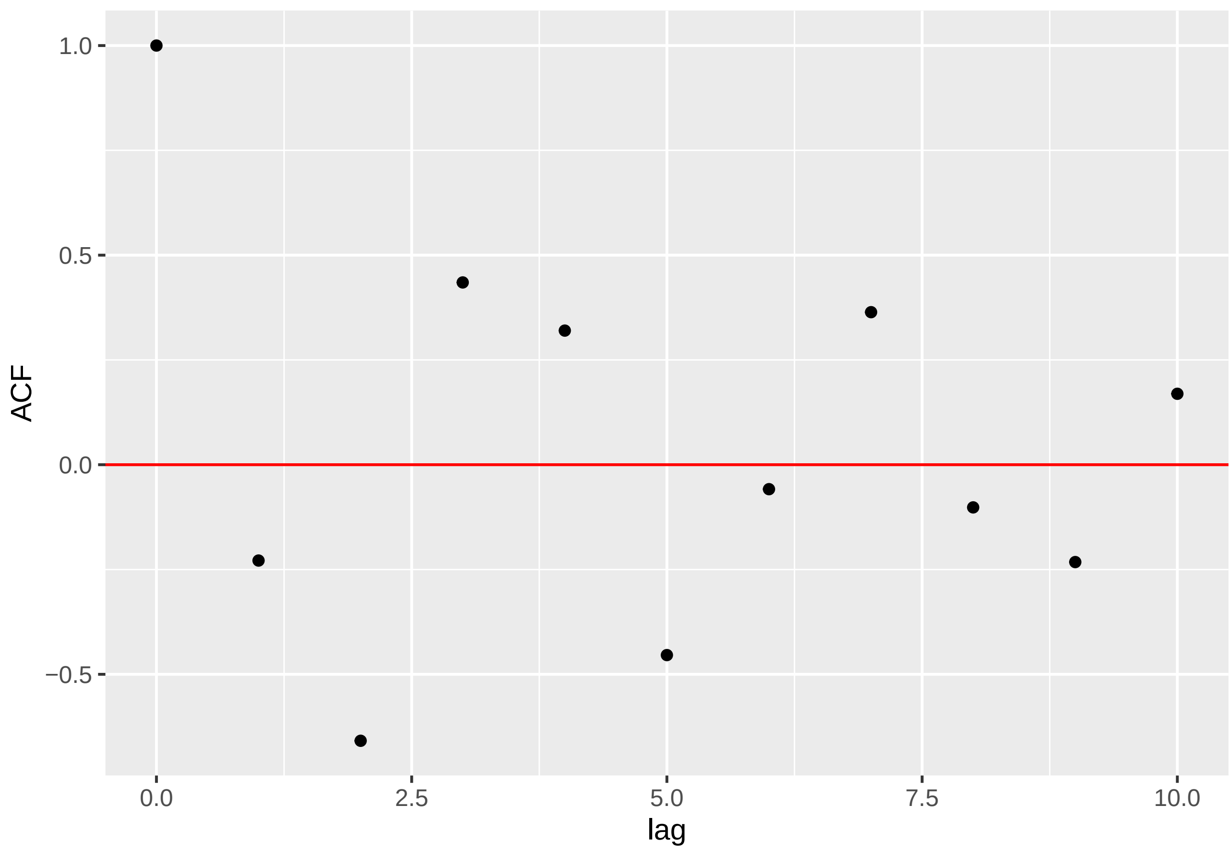
Aplicamos:

```r
acf_coefs_ej2<-c(1,-8/35,-461/700)
phi_ej2<-c(-0.4,-0.75)
acf_coefs_ej2<-coefs_acf(phi_ej2,acf_coefs_ej2,10)
ACF_2=data.frame(ACF=acf_coefs_ej2,lag=c(0:10))
ACF_2
```

```
##             ACF lag
## 1   1.00000000   0
## 2  -0.22857143   1
## 3  -0.65857143   2
## 4   0.43485714   3
```

```
## 5   0.31998571   4
## 6  -0.45413714   5
## 7  -0.05833443   6
## 8   0.36393663   7
## 9  -0.10182383   8
## 10 -0.23222294   9
## 11  0.16925705  10
```

```
ggplot(ACF_2,aes(x=lag,y=ACF))+geom_point()+geom_abline(intercept = 0,slope=0,color='red')
```



Entonces:

```
Var_Xt<-1/(1-sum(phi_ej2*(acf_coefs_ej2[2:3])))
Var_Xt
```

```
## [1] 2.411714
```

Sabemos que en un modelo AR(p), solo los primeros p coeficientes del PACF son distintos de 0, es decir, en este caso los primeros 2, para lag=1 y 2. Aplicamos la función escrita anteriormente:

```
pacf_ej2=c()
for (i in 1:3){
  pacf_ej2[i]<-coefs_pacf(acf_coefs_ej2,i-1)
}
for (i in 4:11){
  pacf_ej2[i]<-0
}
PACF_2=data.frame(PACF=pacf_ej2,lag=c(0:10))
PACF_2
```

```
##          PACF lag
## 1   1.0000000   0
## 2  -0.2285714   1
## 3  -0.7500000   2
## 4   0.0000000   3
## 5   0.0000000   4
## 6   0.0000000   5
## 7   0.0000000   6
## 8   0.0000000   7
## 9   0.0000000   8
## 10  0.0000000   9
## 11  0.0000000  10
```

Graficamos

```
ggplot(PACF_2,aes(x=lag,y=PACF))+geom_point()+geom_abline(intercept = 0,slope=0,color='red')
```