

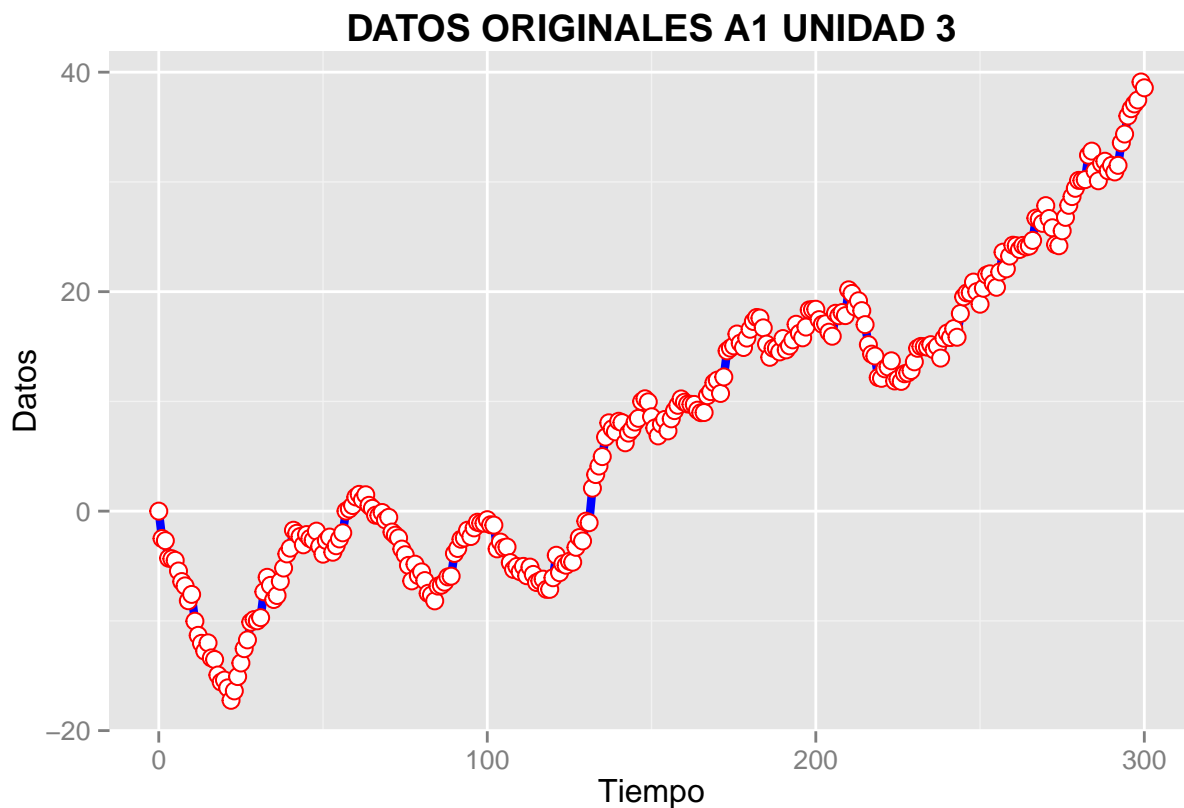
Actividad 1 Unidad 3

Carlos Antonio Lara Verduzco

Friday, September 04, 2015

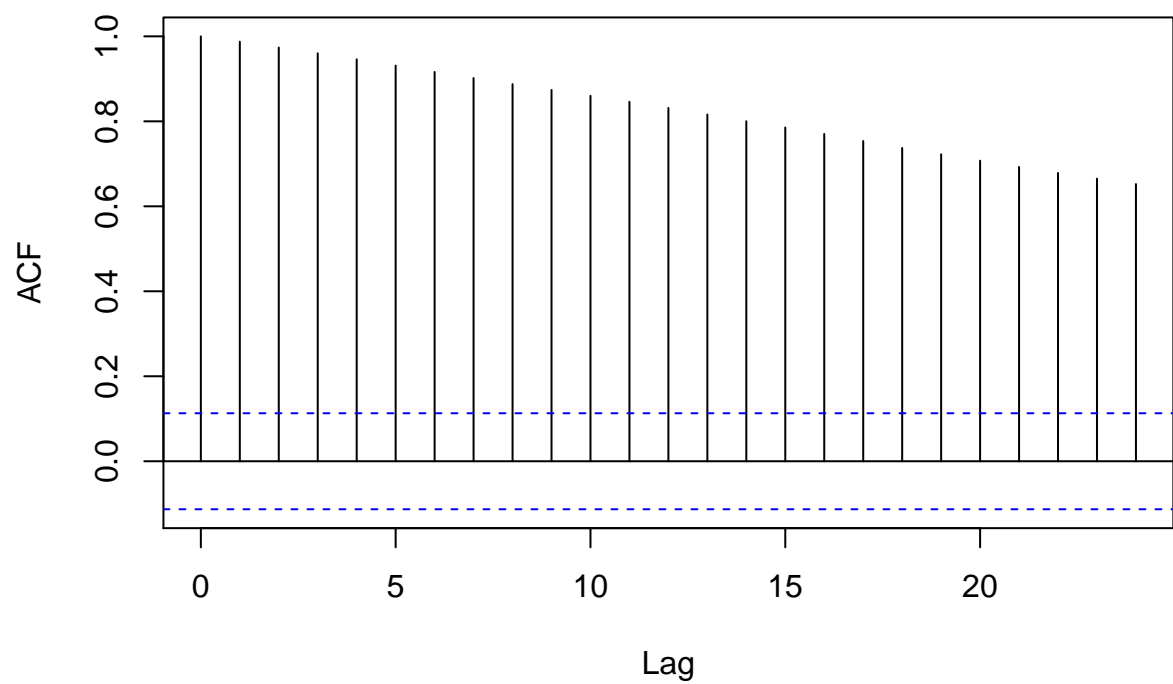
Los datos que bienen en el archivo “Datos y presentación de un modelo” los introducimos en R y al graficarlos obtenemos:

```
setwd("~/Estadistica 3")
Datos <- read.table("datos.txt", sep="\t")
Tiempo<- 0:300
library("ggplot2")
Dat<- data.frame(Tiempo,Datos)
library(plyr)
Dat<- rename(Dat, c("V1"="Datos"))
bp<- ggplot(data=Dat, aes(x=Tiempo, y=Datos, group=1)) +
  geom_line(colour="blue", linetype="solid", size=1.5) +
  geom_point(colour="red", size=3, shape=21, fill="white")
bp + ggtitle("DATOS ORIGINALES A1 UNIDAD 3") +
  theme(plot.title = element_text(lineheight=.8, face="bold"))
```

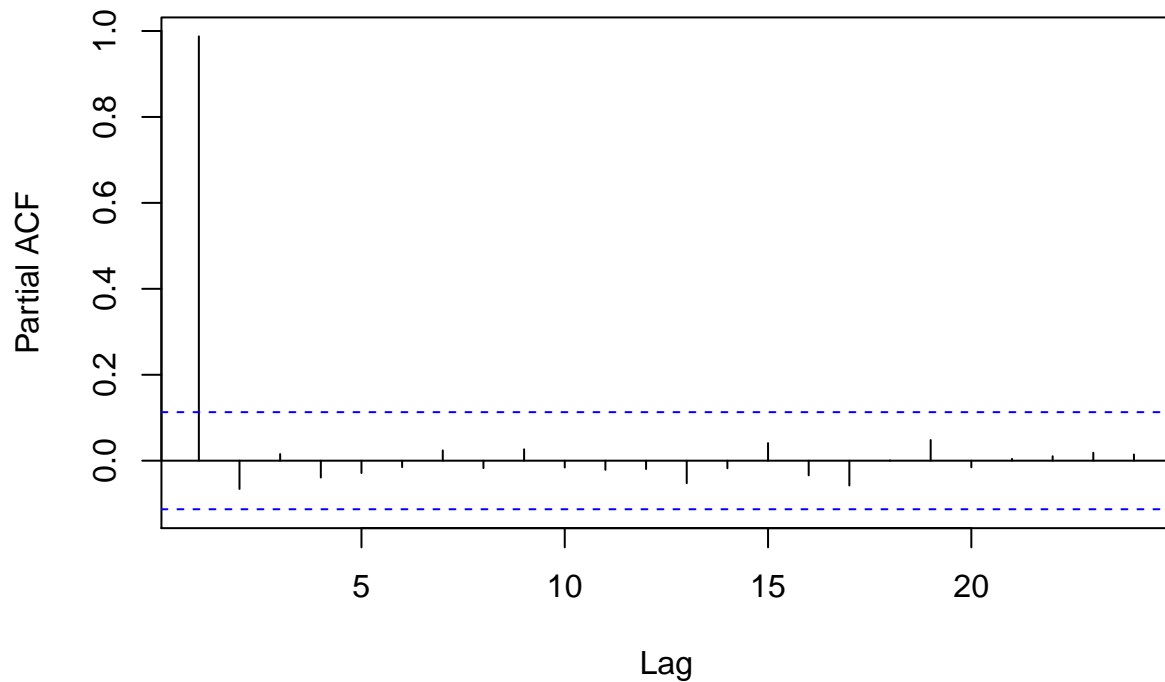


Los correlogramas correspondientes a *ACF* y *PACF* a los datos proporcionados en la actividad son:

AFC_Autocorrelación



PAFC Autocorrelación Parcial



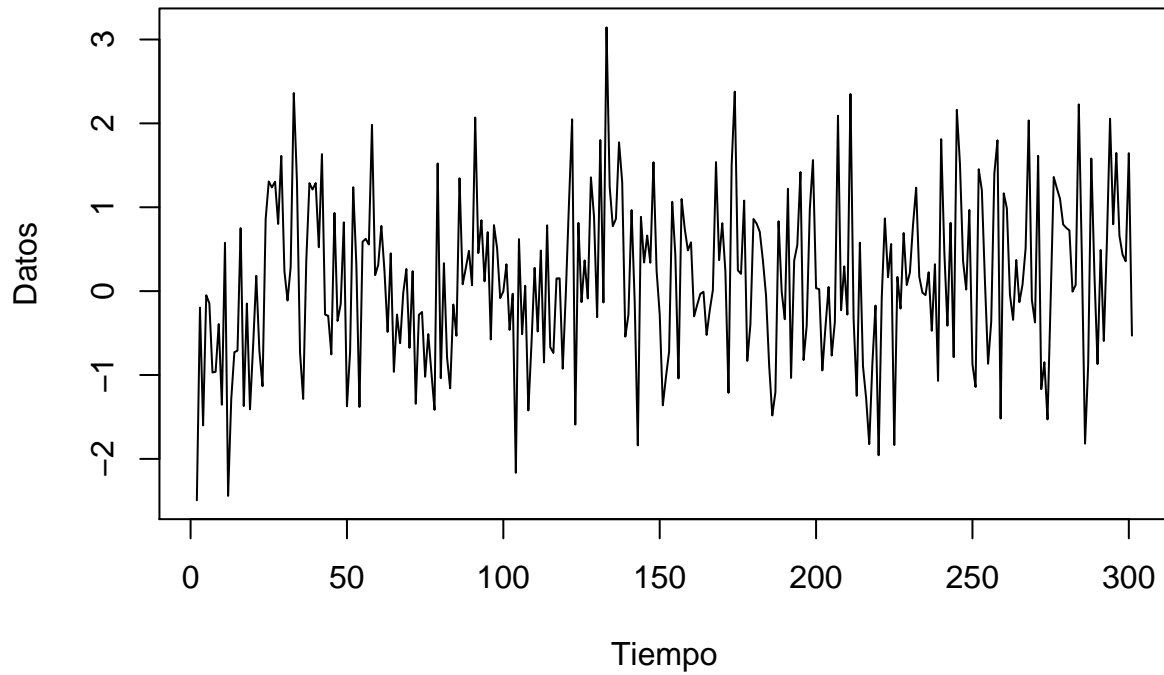
En el presente caso de esta base de datos, la serie de tiempo no es estacionaria, vemos que la variabilidad no es muy estable pues esta se incrementa al principio y luego disminuye considerablemente y presenta, en general, una tendencia positiva. Se aprecia también un cambio en la media con el tiempo por la tendencia positiva de los datos. Por ende, podemos considerar primeramente realizar un ajuste quizá logarítmico, es decir utilizar ajuste tipo Box-Cox. Sin embargo, la serie presenta datos negativos y no podremos calcular λ o $\ln X_t$ para algunos valores de la variable aleatoria. En este caso, es mejor considerar el método de diferencias o datos diferenciados $Y_t = \nabla X_t$ o

$$\nabla X_t = X_t - X_{t-1}$$

Con el software R esto es posible con la función “diff()” y el resultado al graficar los nuevos datos es:

```
x1<- Dat[,2]
x <- ts(x1)
Dat_Dif<-diff(x,1)
ts.plot(Dat_Dif, main = "Datos Ajustados por Diferencial (1)", xlab="Tiempo", ylab="Datos")
```

Datos Ajustados por Diferencial (1)

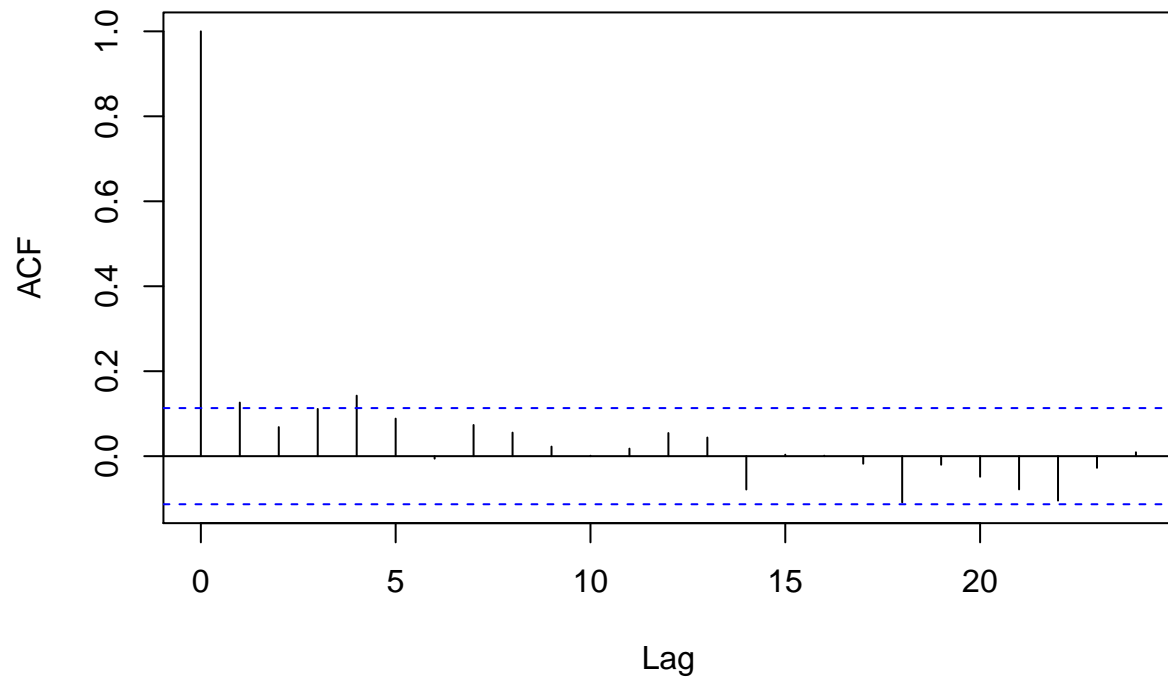


Nótese ahora se tiene un comportamiento estacionario, de hecho, los datos se distribuyen alrededor de su media a lo largo del tiempo y dicha media es:

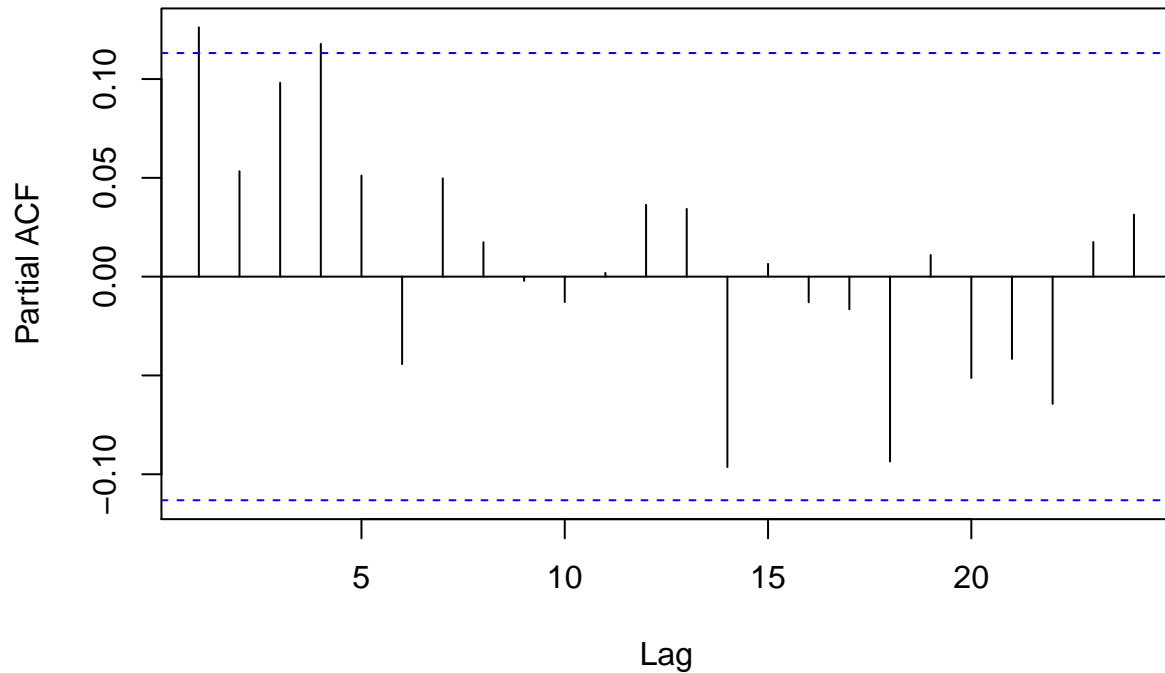
```
## [1] 0.1285916
```

Podemos ahora, presentar los correlogramas de los nuevos datos:

AFC Autocorrelación



PAFC Autocorrelación Parcial



En el modelo ajustado, los correlogramas muestran muchos picos no nulos. Vemos en ambos que se hacen completamente nulos después de Lag 4 aproximadamente. Podemos entonces proponer algún modelo $MA(4)$ o $AR(4)$ que modele los datos. Para ver cual se ajusta mejor, utilizamos en R el Criterio Akaike (AIC); el modelo con un coeficiente menor AIC será el más adecuado, esto es:

```
AIC(arima(Dat_Dif,order = c(0,0,4)),arima(Dat_Dif,order = c(4,0,0)))
```

```
##                                df      AIC
## arima(Dat_Dif, order = c(0, 0, 4))  6 837.1672
## arima(Dat_Dif, order = c(4, 0, 0))  6 835.2060
```

Véase como un $AR(4)$ parece ser mejor, sin embargo un $MA(4)$ no está lejos según el criterio empleado. Podemos entonces proponer un $AR(4)$ Ajustando el modelo a los datos se tiene: Primero para saber qué modelo $AR(p)$ es mejor para nuestros datos, usamos la función $ar()$, esto nos dirá tanto el orden como los coeficientes del modelo mediante sus MLE . El resultado es:

```
ar(Dat_Dif)
```

```
##
## Call:
## ar(x = Dat_Dif)
##
## Coefficients:
##      1      2      3      4
## 0.1026 0.0367 0.0846 0.1178
```

```
##  
## Order selected 4  sigma^2 estimated as  0.9273
```

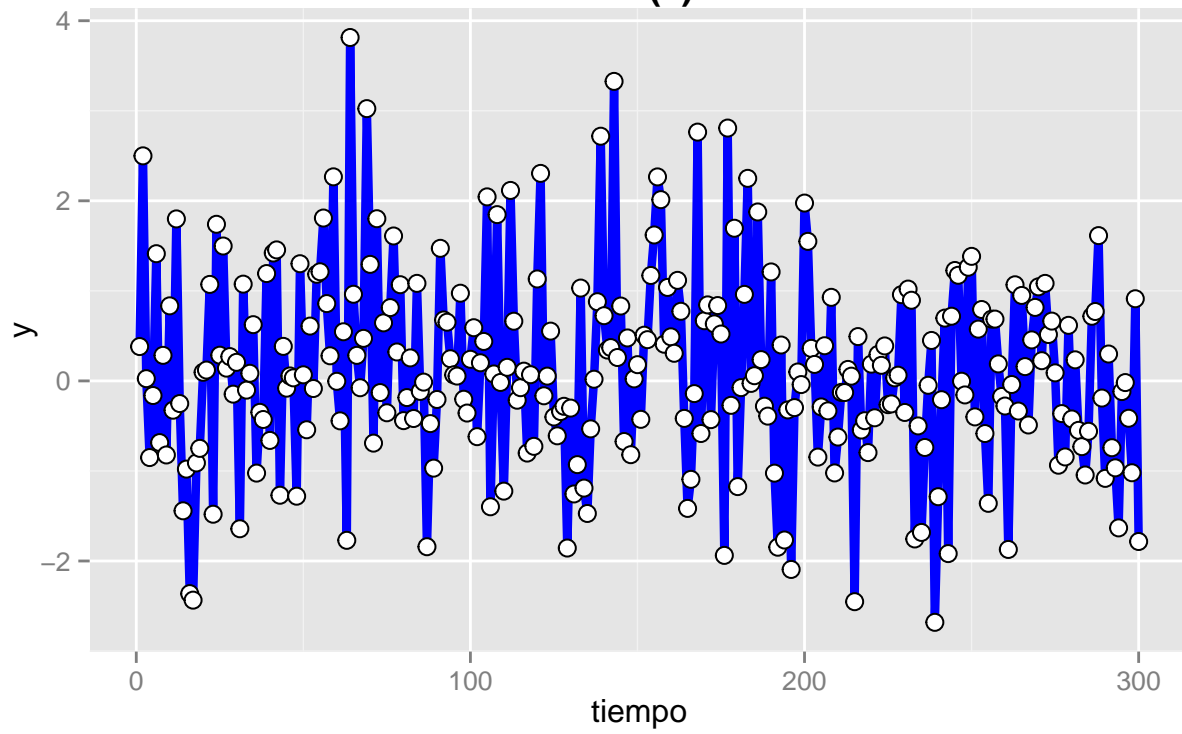
Vemos como nuestra predicción a partir de los correlogramas y el criterio AIC concuerdan con lo arrojado por `ar()` y vemos como incluso $\sigma^2 = 1$ con lo que el modelo para los datos sería:

$$X_t = 0.1026X_{t-1} + 0.0367X_{t-2} + 0.0846X_{t-3} + 0.1178X_{t-4} + \epsilon_t$$

Donde ϵ_t es ruido blanco donde $\mu = 0$ y $\sigma^2 = 1$. Ahora, podemos hacer una simulación con el modelo obtenido para comparar los datos generados con los originales y ver si el comportamiento es similar. La gráfica de la simulación es:

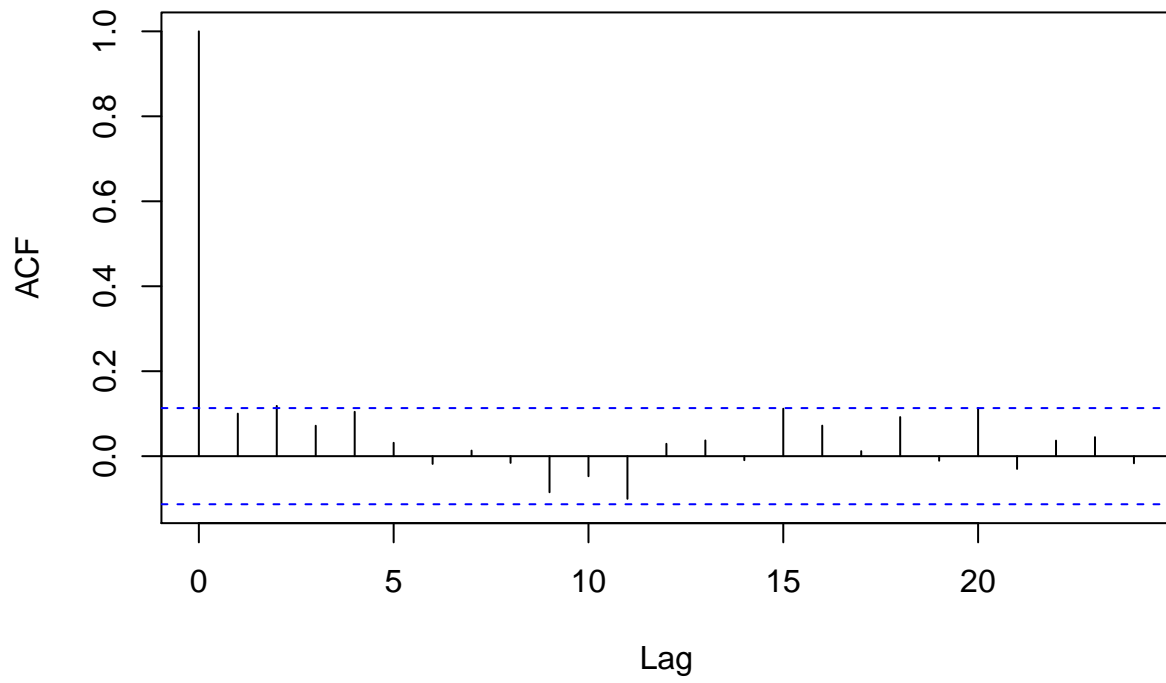
```
phi1<-0.1026  
phi2<-0.036  
phi3<-0.0846  
phi4<-0.1178  
n<-300  
  
mburn <- 300  
  
m <- n + mburn  
  
y <- {1:m}  
tiempo <- {1:n}  
  
# generar ruido blanco  
eps <- rnorm(m)  
  
y[1] <- eps[1]  
y[2] <- eps[2]  
y[3] <- eps[3]  
y[4] <- eps[4]  
  
# generar proceso AR(1) en vec y[]  
for(i in {5:m}){  
  
  y[i] <- phi1*y[i-1] +phi2*y[i-2]+phi3*y[i-3]+phi4*y[i-4] + eps[i]}  
  
# tirar las primeras mburn observaciones  
y <- y[(mburn+1):m]  
dat<- data.frame(tiempo,y)  
  
# graficar la serie de tiempo  
bp<- ggplot(data=dat, aes(x=tiempo, y=y, group=1)) +  
  geom_line(colour="blue", linetype="solid", size=1.5) +  
  geom_point(colour="black", size=3, shape=21, fill="white")  
  
bp + ggtitle("METODO AUTOREGRESIVO\n AR(4)") +  
  theme(plot.title = element_text(lineheight=.8, face="bold"))
```

METODO AUTOREGRESIVO AR(4)

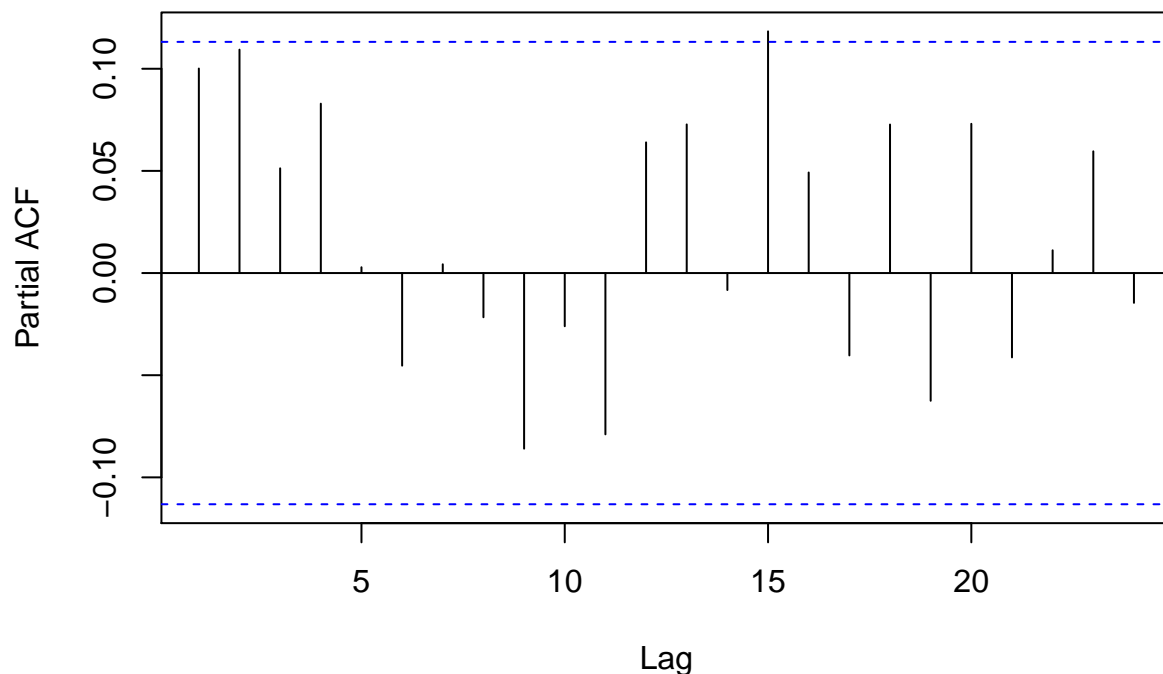


fin de la funcion

AFC Autocorrelación



PAFC Autocorrelación Parcial



En conclusión, hemos hecho un ajuste de los datos que teníamos originalmente de manera que la serie de tiempo fuera estacionaria. Para ello utilizamos datos diferenciados $Y_t = \nabla X_t$ con $\nabla X_t = X_t - X_{t-1}$ después graficamos y calculamos los correlogramas de los nuevos datos. De ellos nos servimos para poder estimar el tipo de modelo tipo $ARMA(p, q)$ que mejor se ajustara a la nueva serie estacionaria utilizando el software R. Concluimos de nuestro análisis que el modelo que representa nuestros datos es un $AR(4)$ y hemos hecho una simulación con éste donde nos damos cuenta que los datos generados y sus correlogramas presentan un comportamiento muy similar, lo cual nos dice, que efectivamente hemos hecho una buena estimación a partir de la transformación que hicimos con los datos originales.