Examen RMT

J. Antonio García Ramirez

Octobre 12, 2018

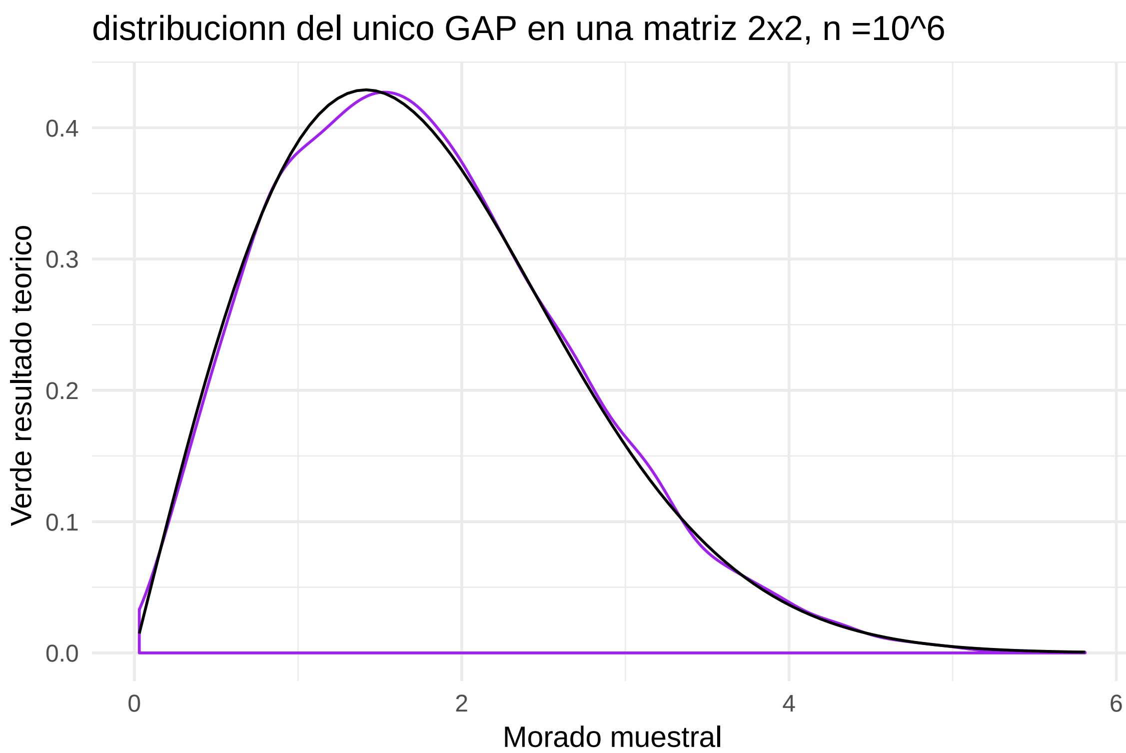
Ejercicio 4

Simule un ensamble GOE con 10^4 iteraciones de matrices simetricas H_s de 2×2

El resultado teorico fue parte de la tarea extra pasada y es $\frac{s}{2}e^{-s^2/4}$

Y lo simulo

```
set.seed(0)
GOE.2.2 <- function(i)
{
  m <- matrix(rnorm(4), ncol= 2 , nrow=2)
  m <- (m + t(m))/2
  valores <- eigen(m)$values
  valores <- sort(valores)
  s <- diff(valores)
  return(s)
}
n <- 10**4
simulacion <- mapply(1:n, FUN=GOE.2.2)
df <- data.frame(simulacion = simulacion)
teorico <- function(x) { (x/2)*exp(-x^2/4)}
library(ggplot2)
ggplot(data=df, aes(x = simulacion)) +geom_density(aes(colour=I('purple'))) +
  theme_minimal() + stat_function(fun = teorico) +xlab('Morado muestral')+
  ylab('Verde resultado teorico ') + ggtitle('distribucionn del unico GAP en una matriz 2x2, n =10^6')
```



Y los resultados del test de Kolmogorov, despues de simular la distribución por el metodo de la función inversa es :

```
teorica <- runif(n, 0,1)
teorica <- (-4*log(1-teorica))**.5
ks.test(simulacion, teorica)
```

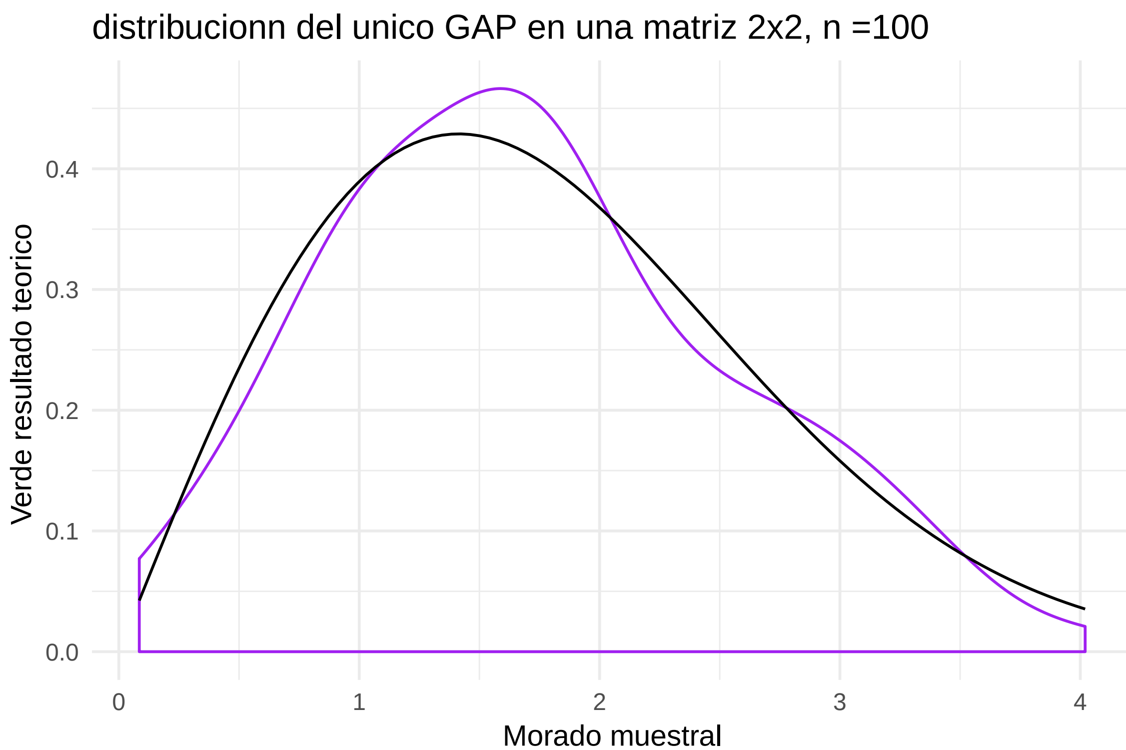
```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  simulacion and teorica
## D = 0.0158, p-value = 0.1647
## alternative hypothesis: two-sided
```

Con lo cual concordamos en que las distribuciones son iguales, pero era de esperar pues este caso es sencillo... y es un resultado exacto.

También lo simule para un n menor

```
set.seed(0)
GOE.2.2 <- function(i)
{
  m <- matrix(rnorm(4), ncol= 2 , nrow=2)
  m <- (m + t(m))/2
  valores <- eigen(m)$values
  valores <- sort(valores)
  s <- diff(valores)
  return(s)
}
n <- 100
simulacion <- mapply(1:n, FUN=GOE.2.2)
```

```r
df <- data.frame(simulacion = simulacion)
teorico <- function(x) { (x/2)*exp(-x^2/4)}
library(ggplot2)
ggplot(data=df, aes(x = simulacion)) +geom_density(aes(colour=I('purple'))) +
  theme_minimal() + stat_function(fun = teorico) +xlab('Morado muestral')+
  ylab('Verde resultado teorico ') + ggtitle('distribucionn del unico GAP en una matriz 2x2, n =100')
```



Y los resultados del test de Kolmogorov, despues de simular la distribución por el metodo de la función inversa es :

```r
teorica <- runif(n, 0,1)
teorica <- (-4*log(1-teorica))**.5
ks.test(simulacion, teorica)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  simulacion and teorica
## D = 0.11, p-value = 0.5806
## alternative hypothesis: two-sided
```

Ejercicio 3

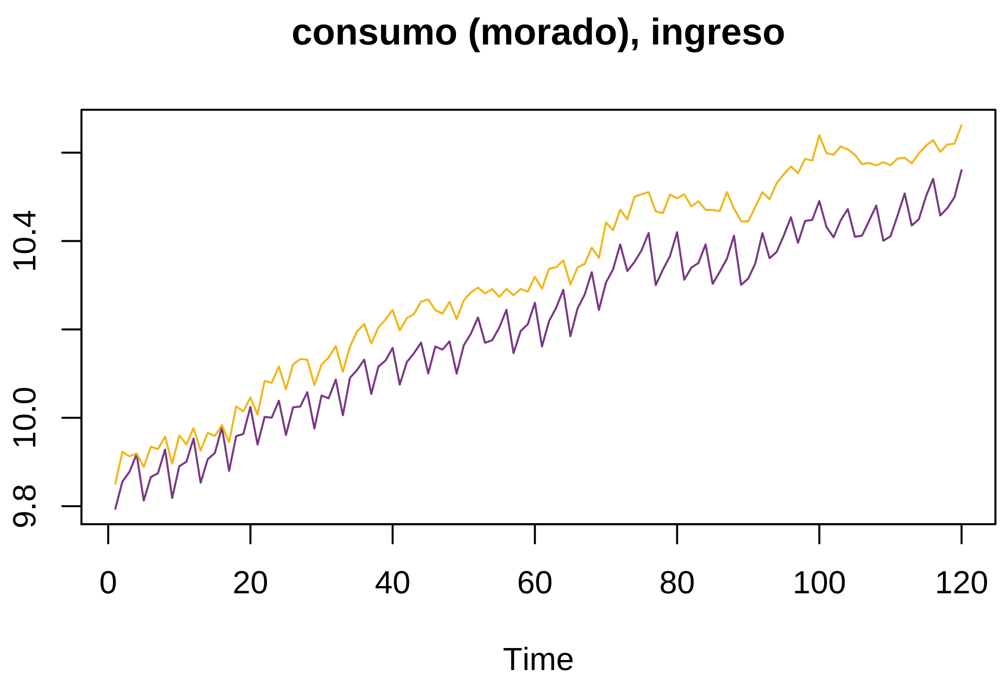
Cargamos las series de tiempo

```r
library(forecast)
library(vars)
library(urca)
UK <- data(UKconinc)
```

a) Visualizamos las series originales

El test aumentado de Dickey-Fuller señala en ambas series la presencia de raices unitarias :< y por lo tanto pueden existir cambios estructurales.

```r
library(tseries)
#class(UKconinc)
datos <- UKconinc
ts.plot(datos, main='consumo (morado), ingreso',
        col=c('#783884', '#F1B61A') )
```




```r
adf.test(datos$conl)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  datos$conl
## Dickey-Fuller = -2.3099, Lag order = 4, p-value = 0.4479
## alternative hypothesis: stationary
```

```r
adf.test(datos$incl)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  datos$incl
## Dickey-Fuller = -2.3324, Lag order = 4, p-value = 0.4385
## alternative hypothesis: stationary
```

```r
adf.test.custom(datos$conl, option='both')
```

```
## Lag optimo
```

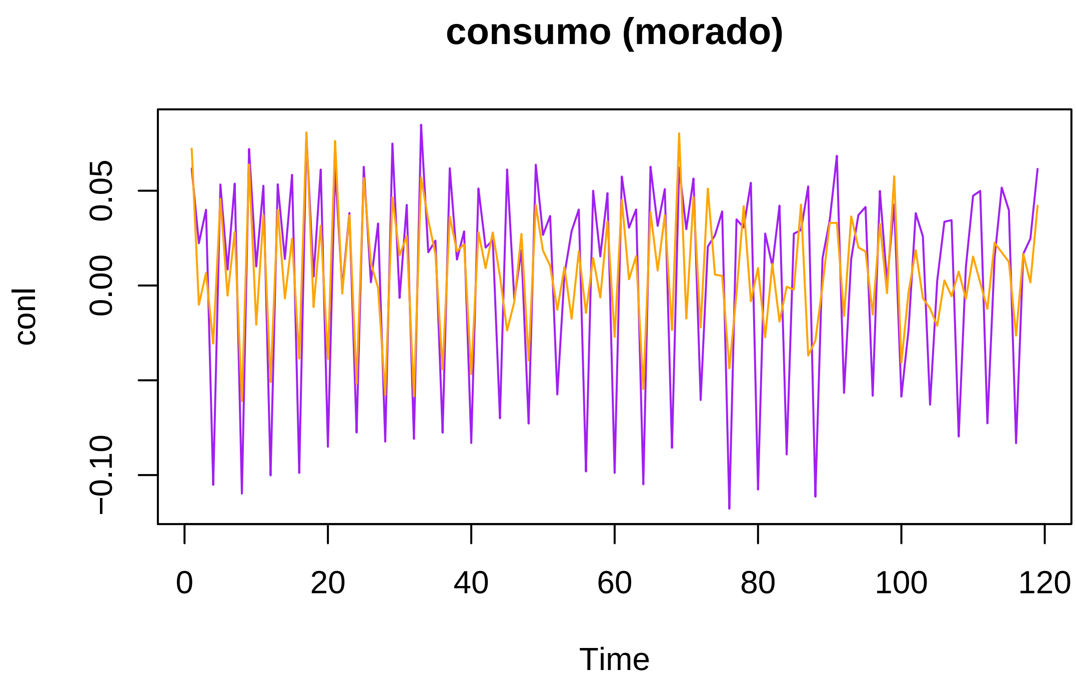
```
##          4
```

```r
adf.test.custom(datos$incl, option='both')
```

```
## Lag optimo
##          3
```

Como los resultados de diferencias cada serie dos veces, con lags de uno y posteriormente 4 se pueden representar por modelos MA tenemos evidencia para concluir que el orden de integración del consumo y el ingreso es 2.

```r
lag <- 4
conl <- diff(datos$conl)
ts.plot( conl , main='consumo (morado)', col='purple')
incl <- diff(datos$incl)
points(incl , main='ingreso', col='orange', type='l')
```




```r
adf.test( conl)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  conl
## Dickey-Fuller = -4.1175, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test( incl)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  incl
```

```
## Dickey-Fuller = -4.1429, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test.custom(conl, option='none')
```

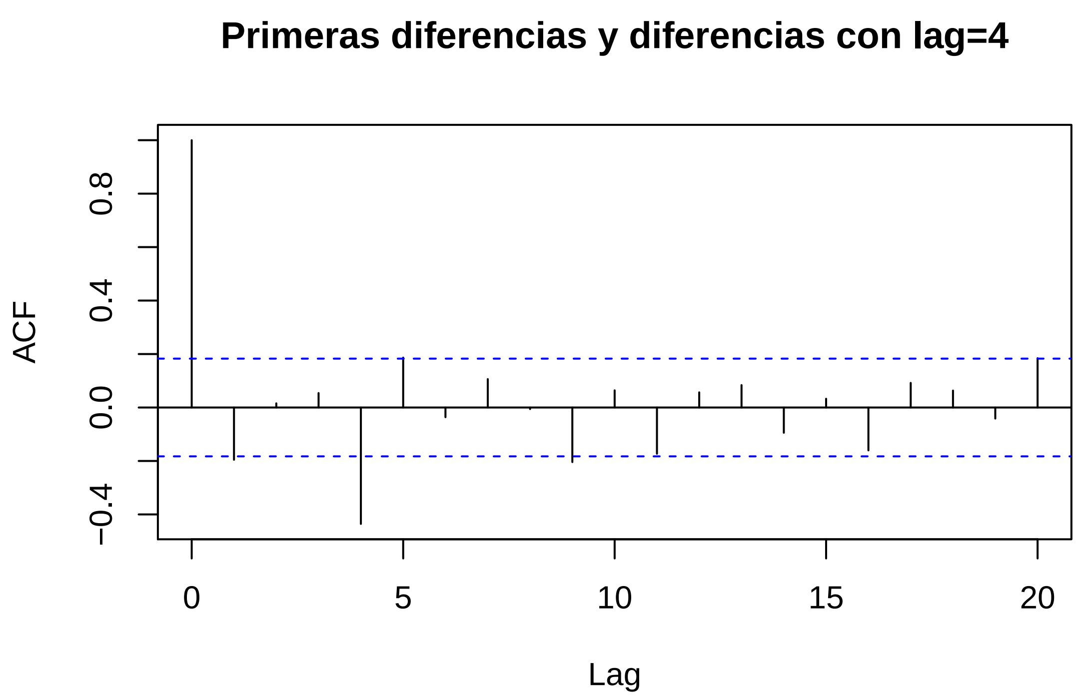
```
## Lag optimo
##          4
```

```r
adf.test.custom(incl, option='none')
```

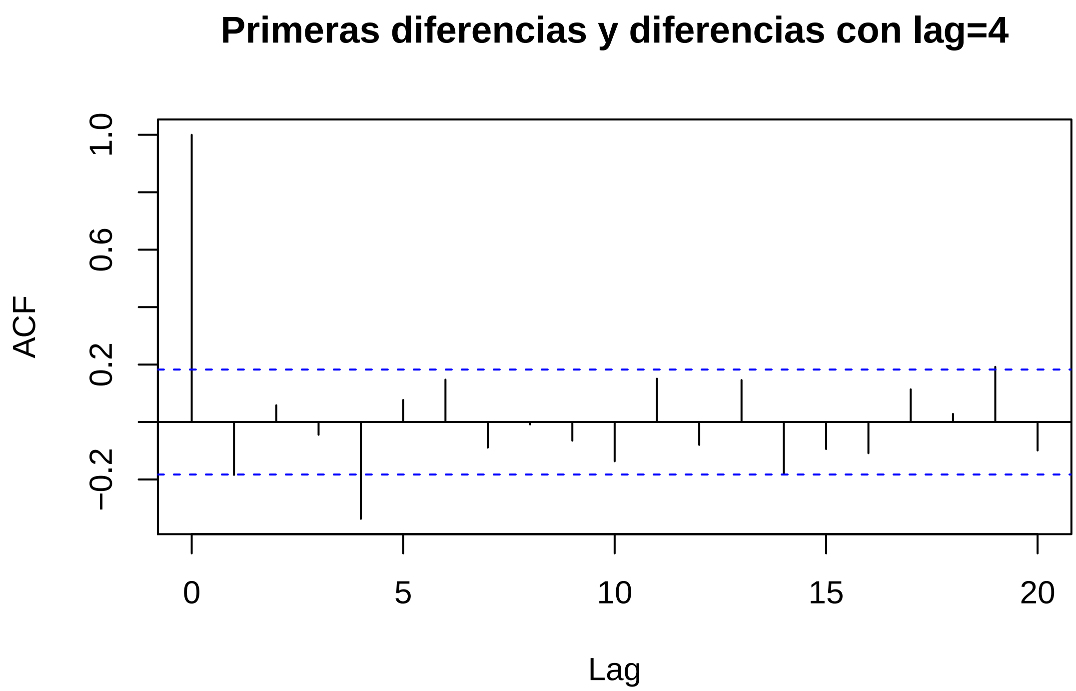
```
## Lag optimo
##          4
```

Como podemos ver las dos series diferenciadas parecen no tener más de una raiz unitaria por lo que tenemos elementos para afirmar que las series tienen el mismo orden de integración despues de diferenciar nuevamente pero con un lag de 4 que coincide con el lag de nuestra prueba implementada en la tarea 2

```r
conl <- diff(conl, lag=lag)
incl <- diff(incl, lag=lag)
acf(conl, main='Primeras diferencias y diferencias con lag=4')
```




```r
acf(incl, main='Primeras diferencias y diferencias con lag=4')
```



Estimamos los modelos, con fines dídacticos

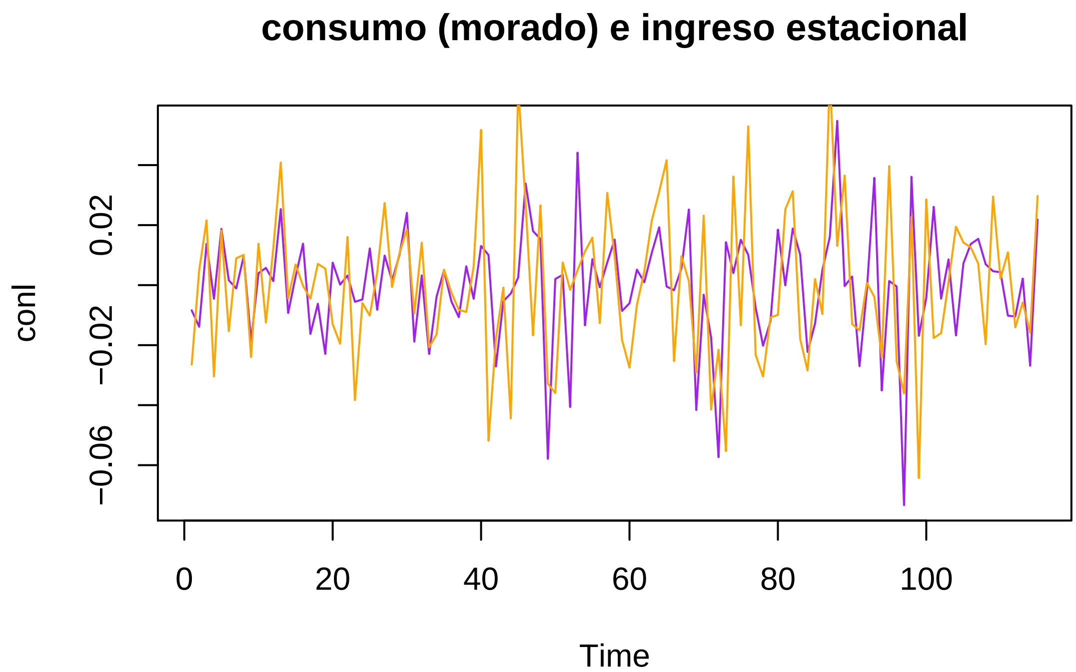
```
auto.arima(conl, stationary=TRUE ) # consumo de orden I(2)
```

```
## Series: conl
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##           ma1
##       -0.2034
## s.e.   0.0955
##
## sigma^2 estimated as 0.0003489:  log likelihood=295.05
## AIC=-586.09   AICc=-585.99   BIC=-580.61
```

```
auto.arima(incl, stationary=TRUE) # consumo de orden I(2)
```

```
## Series: incl
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 0.0006036:  log likelihood=263.05
## AIC=-524.09   AICc=-524.06   BIC=-521.35
```

```
ts.plot( conl , main='consumo (morado) e ingreso estacional',
         col='purple')
points(incl , main='', col='orange', type='l')
```



b)

Decidi no incluir la tendencia ni el drift en la selección del número de rezagos pues las series ya estan estacionalizadas y parecen tener media cercana a cero, adémas elegé el criterio de $HQ(n)$ (en un principio para tener un modelo parsimonioso) sin embargo déspues de una busqueda exhaustiva el criterio AIC logro un mejor ajuste sobre los demás modelos (considerando para ello el R^2 ajustado el cual nunca sobre paso a 0.5.

```
datos.estacionales <- cbind(incl, conl)
apply(datos.estacionales, 2, mean)
```

```
##          incl          conl
## -3.531304e-05  8.956522e-06
```

```
p <- VARselect(datos.estacionales, type = "none")
p$selection #rezagos sugeridos
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      4      1     10
```

```
p <- p$selection['AIC(n)']
p
```

```
## AIC(n)
##     10
```

c) Como la implementación de la funcion "VAR" de R estima por OLS la utilice. Los resultados se muestran en la siguiente tabla:

En vista de muchos coeficientes no son significativos, y que de manera conjunta tampo lo son esperamos que la varianza del proceso se incremente con el tiempo, es decir que no sea estacional a largo plazo.

La mayoría de los coeficientes con este modelo no son sifnificativos, la significancia conjunta no se logra ni individualmente, la bondad de ajuste es mala al medirla con el R cuadrado ajustado, sobre todo porque las regresiones presentan heterosedasticidad es decir que los errores de las regresiones para ambas series estan correlacionados por 0.5

```
v.ar <- VAR(datos.estacionales, p = p)
summary(v.ar)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: incl, conl
## Deterministic variables: const
## Sample size: 105
## Log Likelihood: 585.116
## Roots of the characteristic polynomial:
## 0.9299 0.9299 0.9222 0.9222 0.9095 0.9095 0.9044 0.9044 0.894 0.894 0.8906 0.8906 0.8755 0.8755 0.784
## Call:
## VAR(y = datos.estacionales, p = p)
##
##
## Estimation results for equation incl:
## =====================================
## incl = incl.l1 + conl.l1 + incl.l2 + conl.l2 + incl.l3 + conl.l3 + incl.l4 + conl.l4 + incl.l5 + conl
##
##           Estimate Std. Error t value Pr(>|t|)
## incl.l1  -0.4571737  0.1230996  -3.714 0.000366 ***
## conl.l1   0.3225780  0.1582744   2.038 0.044687 *
## incl.l2  -0.2800458  0.1412872  -1.982 0.050737 .
## conl.l2   0.7318607  0.1717357   4.262 5.28e-05 ***
## incl.l3  -0.4571910  0.1387430  -3.295 0.001442 **
## conl.l3   0.3196599  0.1890714   1.691 0.094605 .
## incl.l4  -0.8164786  0.1471363  -5.549 3.27e-07 ***
## conl.l4   0.4547278  0.1901914   2.391 0.019042 *
## incl.l5  -0.5267058  0.1598658  -3.295 0.001444 **
## conl.l5   0.6251356  0.2037301   3.068 0.002896 **
## incl.l6  -0.1673255  0.1549435  -1.080 0.283272
## conl.l6   0.8072973  0.2080191   3.881 0.000206 ***
## incl.l7  -0.5107066  0.1399925  -3.648 0.000457 ***
## conl.l7   0.6093564  0.1902535   3.203 0.001923 **
## incl.l8  -0.5957791  0.1429412  -4.168 7.43e-05 ***
## conl.l8   0.5652390  0.1906102   2.965 0.003934 **
## incl.l9  -0.2298314  0.1413438  -1.626 0.107686
## conl.l9   0.3353410  0.1848640   1.814 0.073250 .
## incl.l10 -0.0619446  0.1230785  -0.503 0.616075
## conl.l10  0.3552302  0.1550081   2.292 0.024426 *
## const    -0.0003139  0.0019862  -0.158 0.874799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.02028 on 84 degrees of freedom
## Multiple R-Squared: 0.4749,  Adjusted R-squared: 0.3499
## F-statistic: 3.799 on 20 and 84 DF,  p-value: 8.536e-06
##
##
```

```
## Estimation results for equation conl:
## =====================================
## conl = incl.l1 + conl.l1 + incl.l2 + conl.l2 + incl.l3 + conl.l3 + incl.l4 + conl.l4 + incl.l5 + conl
##
##            Estimate Std. Error t value Pr(>|t|)
## incl.l1   0.1943091  0.0961824   2.020  0.04655 *
## conl.l1  -0.3548901  0.1236657  -2.870  0.00520 **
## incl.l2   0.0032492  0.1103930   0.029  0.97659
## conl.l2  -0.1327161  0.1341836  -0.989  0.32547
## incl.l3   0.0090392  0.1084051   0.083  0.93375
## conl.l3  -0.0499717  0.1477286  -0.338  0.73601
## incl.l4  -0.2301941  0.1149631  -2.002  0.04848 *
## conl.l4  -0.2808323  0.1486037  -1.890  0.06223 .
## incl.l5  -0.1905913  0.1249092  -1.526  0.13081
## conl.l5   0.0633983  0.1591819   0.398  0.69144
## incl.l6  -0.1720573  0.1210632  -1.421  0.15895
## conl.l6   0.2665423  0.1625331   1.640  0.10476
## incl.l7  -0.3025867  0.1093814  -2.766  0.00697 **
## conl.l7   0.3666759  0.1486522   2.467  0.01567 *
## incl.l8  -0.3188036  0.1116853  -2.854  0.00543 **
## conl.l8   0.1469080  0.1489309   0.986  0.32676
## incl.l9  -0.3220248  0.1104372  -2.916  0.00455 **
## conl.l9   0.1734806  0.1444412   1.201  0.23311
## incl.l10 -0.1798744  0.0961659  -1.870  0.06490 .
## conl.l10  0.2307165  0.1211137   1.905  0.06021 .
## const    -0.0003204  0.0015519  -0.206  0.83691
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01584 on 84 degrees of freedom
## Multiple R-Squared: 0.4744,  Adjusted R-squared: 0.3493
## F-statistic: 3.791 on 20 and 84 DF,  p-value: 8.826e-06
##
##
##
## Covariance matrix of residuals:
##           incl      conl
## incl 0.0004112 0.0001607
## conl 0.0001607 0.0002510
##
## Correlation matrix of residuals:
##        incl   conl
## incl 1.0000 0.5003
## conl 0.5003 1.0000
```

d)

Sobre los residuos encontramos que distan de tener una distribución normal bivariada como se puede ver en la segunda imagen adémas podriamos decir que con una confianza de .9 los residuos son descorrelacionados en cada componente sin embargo con una confianza excesiva los residuos presentan el fenómeno Arch, es decir que la varianza de los errores se mueve con el tiempo.

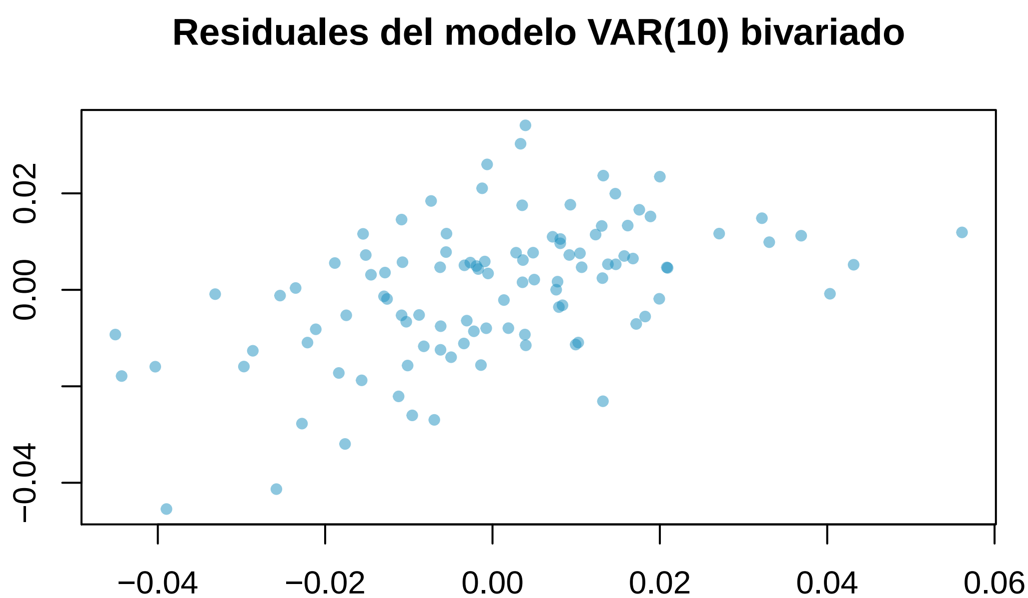
```
residuos <- residuals(v.ar)
serial.test(v.ar) #test de Portmanteau H0: Ausencia correlacion serial
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object v.ar
## Chi-squared = 35.845, df = 24, p-value = 0.05682
```

```r
arch.test(v.ar) #test de arch H0: Prosencia de Arch y heteroscesdicidad
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object v.ar
## Chi-squared = 43.144, df = 45, p-value = 0.5509
```

```r
plot(residuos, col=rgb( 29/255, 145/255, 192/255, alpha=0.5),
     main='Residuales del modelo VAR(10) bivariado', pch=20,
     xlab='',
     ylab='')
```



e) *Eliminando las 5 observaciones finales, vuelve a estimar el VAR realiza el pronóstico 5 pasos adelante, estima el porcentaje de error de pronostico para ambas variables y comenta*

En vista de que eliminamos las últimas 5 observaciones considere prudente volver a estimar el número de rezagos, en vista de que no es computacionalmente más costoso que OLS y por el supuesto de ergodicidad no considere adecuado realizar nuevamente las pruebas de raicez unitarias.

Se encontro que el mejor ajuste se logra también con un modelo $VAR(10)$ y los coeficientes R^2 ajustados son muy parecidos.

```r
train <- datos.estacionales[1:(dim(datos.estacionales)[1]-5), ]
test <- datos.estacionales[(dim(datos.estacionales)[1]-4):dim(datos.estacionales)[1], ]
p <- VARselect(train, type = "none")
```

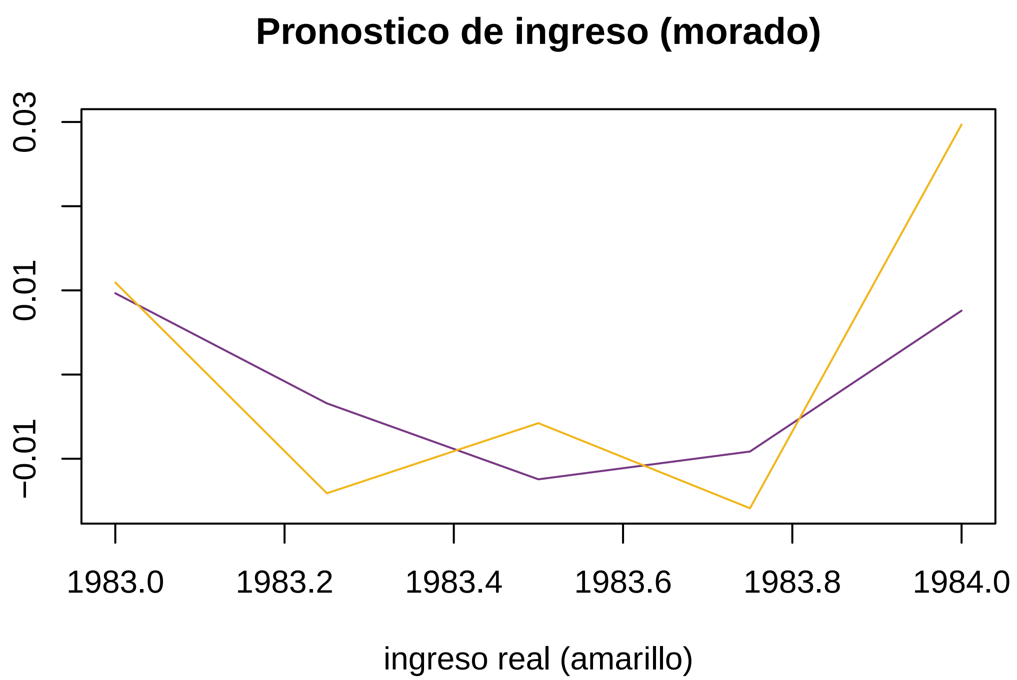
```r
p$selection #rezagos sugeridos iguales
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      4      1     10
```

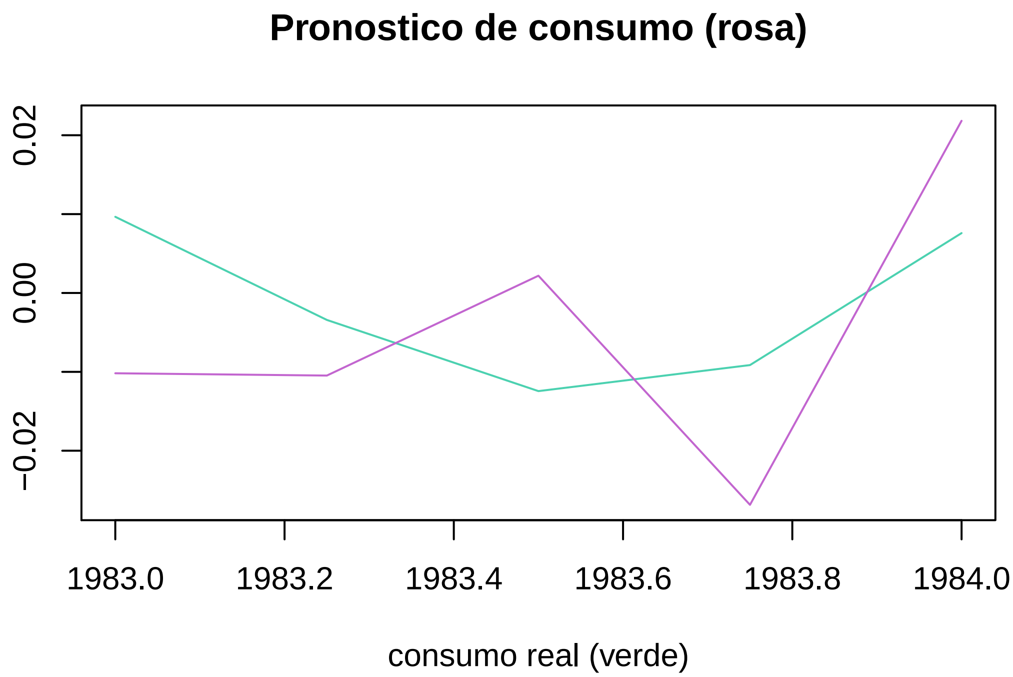
```r
p <- p$selection['AIC(n)']
#p
v.ar <- VAR(train, p = p)
#summary(v.ar)
fore <- predict(v.ar, n.ahead = 5)
```

Decidí utilizar el SMAPE pues es la manera en que medire el porcentaje del error pronósticado en el proyecto final y tiene una interpretación sencilla como de porcentaje de error entre 0 y 100.

```r
pronostico.incl <- ts(fore$fcst$incl[, 'fcst'], end=1984, frequency = 4)
incl.real <- ts(test[,'incl'], end=1984, frequency = 4 )
ts.plot(pronostico.incl, incl.real, xlab='ingreso real (amarillo)',
        main='Pronostico de ingreso (morado)',
        col=c('#783884', '#F1B61A') )
```




```r
pronostico.conl <- ts(fore$fcst$conl[, 'fcst'], end=1984, frequency = 4)
conl.real <- ts(test[,'conl'], end=1984, frequency = 4 )
ts.plot(pronostico.incl, conl.real, xlab='consumo real (verde)',
        main='Pronostico de consumo (rosa)',
        col=c('#4BD1B0', '#C265CF') )
```




```
###cuentas
pronostico.puntual.1 <- ts(fore$fcst$incl[, 'fcst'],  end=1984, frequency = 4)
pronostico.puntual.2 <- ts(fore$fcst$conl[, 'fcst'],  end=1984, frequency = 4)
pronostico.puntual <- ts.intersect(pronostico.puntual.1, pronostico.puntual.2)
pronostico.puntual <- as.data.frame(pronostico.puntual)
SMAPE <- (100/(dim(test)[1]))*
  apply( abs(pronostico.puntual - test)/ (abs(test) + abs(pronostico.puntual)), 2, sum)
```

Al no considerar las últimas cinco observaciones y pronosticarlas con el modelo tenemos las dos anteriores gráficas, mientras que el error promedio de pronóstico es de 38.01% para la primer variable y 67.14% para la segunda, lo cual es un error demasiado alto y siguiere la idea de buscar un modelo $AR(P)$ más apropiado.

f) *¿Están cointegradas las dos variables? Si es asi concluye que sí, estima un modelo de correción de errores e interpreta lo obtenido.*

Como en a) verifique que las series de los logaritmos de consumo e ingresos son no estacionales pero ambas tienen nivel de inetegración 2 estas pueden estar cointegradas. Para verificarlo usare la función de la tarea anterior que implementa las ideas de Mackinnon sobre

```
Mackinnon <- function(x, y, confianza=0.95, type=c("none", "trend", "both"))
{
  df <- as.data.frame(cbind(x,y))
  n <- length(x)
  type <- match.arg(type)
  res <- lm(x~., data=df)$residuals #Generamos una regresion entre ambas series de tiempo y obtenemos s
  res.z <- diff(res)
  res.lag <- res[2:length(res)]
  df2 <- as.data.frame(cbind(res.z,res.lag))
  temp=0
  if (confianza==0.99) {temp=1}
  if (confianza==0.95) {temp=2}
```

```r
if (confianza==0.9) {temp=3}
if (type=="none")
{
  pos.root <- lm(res.z~.-1, data=df2) #calculamos una regresion para un modelo AR(1) con las diferenc
  t_value <- summary(pos.root)$coef[, 3] #calculamos el valor critico
  print(paste("t_value:",round(t_value,2)))
  tabla <- data.frame(cbind(size=c(0.99,0.95,0.9),
                          binf=c(-2.56574, -1.94100,-1.61682),
                          beta1=c(-2.2358, -0.2686,0.2656),
                          beta2=c(-3.627,-3.365,-2.714),
                          beta3=c(0, 31.223,25.364)))
  val.crit <- tabla[temp,2] +
    (tabla[temp,3]/n) + (tabla[temp,4]/(n^2)) + (tabla[temp,5]/(n^3))
  if (t_value<val.crit)
  { #Comparamos el estadistico contra las tablas
    print("Series no cointegradas")
  }
  else{print("Series cointegradas")}
}
  if (type=="trend")
  {
    pos.root <- lm(res.z~., data=df2) #calculamos una regresion para un modelo AR(1) con las diferenc
    t_value <- summary(pos.root)$coef[, 3][2] #calculamos el valor critico
    print(paste("t_value:",round(t_value,2)))
    tabla <- data.frame(cbind(size=c(0.99,0.95,0.9),
                          binf=c(-3.43035,-2.86154,-2.56677),
                          beta1=c(-6.5393,-2.8903,-1.5384),
                          beta2=c(-16.786,-4.234,-2.809),
                          beta3=c(-79.433,-40.040,0)))
  val.crit <- tabla[temp,2]+
    (tabla[temp,3]/n) + (tabla[temp,4]/(n^2)) + (tabla[temp,5]/(n^3))
  if (t_value<val.crit)
  { #Comparamos el estadistico contra las tablas
    print("Series no cointegradas")
  }
  else{print("Series cointegradas")}
}
if (type=="both")
{
  df2$t <- seq(1,(n-1),1)
  pos.root <- lm(res.z~., data=df2) #calculamos una regresion para un modelo AR(1) con las diferencia
  t_value <- summary(pos.root)$coef[, 3][2] #calculamos el valor critico
  print(paste("t_value:",round(t_value,2)))
  tabla <- data.frame(cbind(size=c(0.99,0.95,0.9),
                          binf=c(-3.95877,-3.41049,-3.12705),
                          beta1=c(-6.5393,-2.8903,-1.5384),
                          beta2=c(-16.786,-4.234,-2.809),
                          beta3=c(-79.433,-40.040,0)))
  val.crit <- tabla[temp,2]+ (tabla[temp,3]/n) + (tabla[temp,4]/(n^2)) + (tabla[temp,5]/(n^3))
  if (t_value<val.crit)
  { #Comparamos el estadistico contra las tablas
    print("Series no cointegradas")
  }
```

```
    else{print("Series cointegradas")}
  }
}
```

Y la probare sobre las series originales no estacionales con una confianza de 0.95. La implementación que realice dice que ambas series son cointegradas

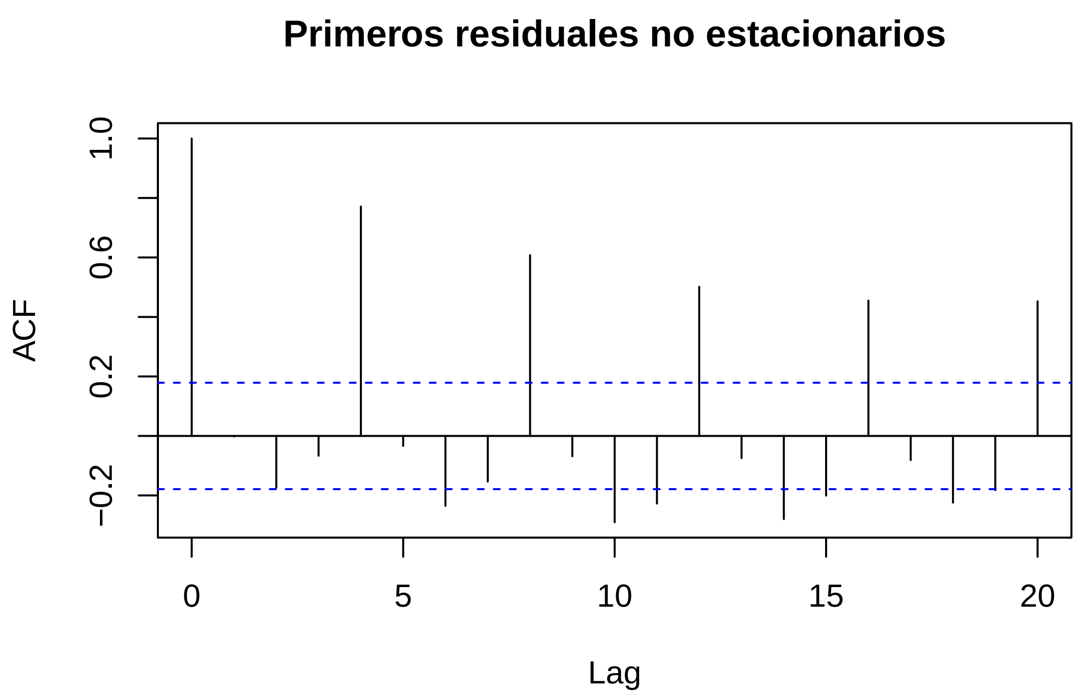
```
Mackinnon(datos$incl, datos$conl)
```

```
## [1] "t_value: 10.97"
## [1] "Series cointegradas"
```

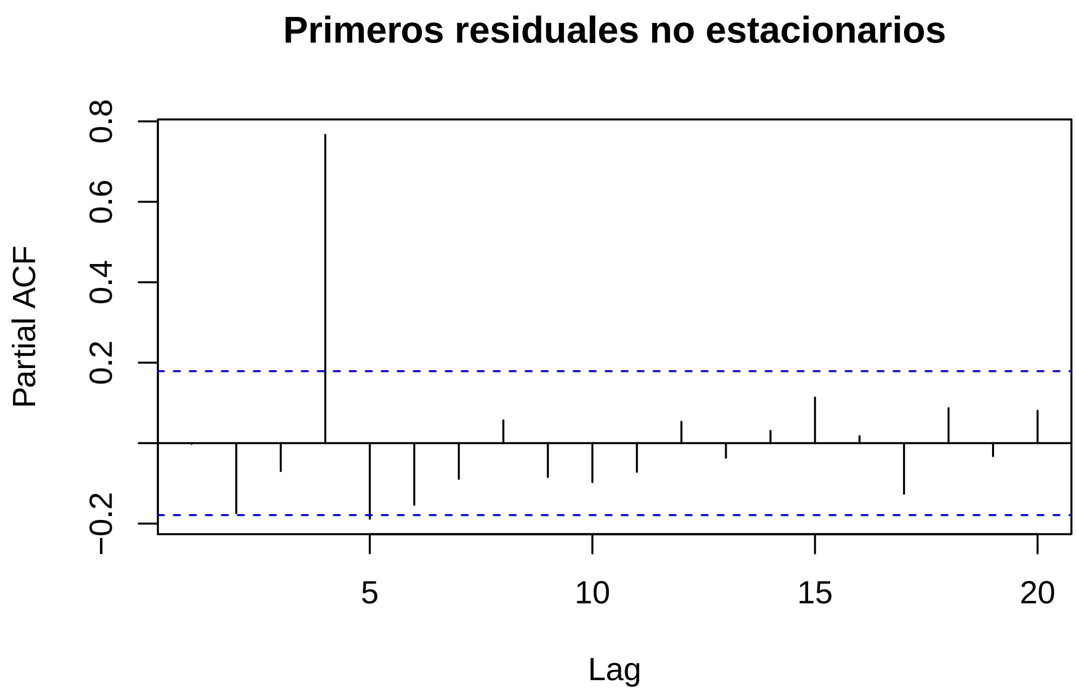
Ahora procederemos a estimar el modelo de correción de error.

De nueva cuenta vamos a regresar el ingreso sobre el consumo con intercepto, que como vimos en el inciso a) tienen comportamientos parecidos y revizamos con las ACF y PACF que los residuales no son estacionarios y es más distan de ser normales.

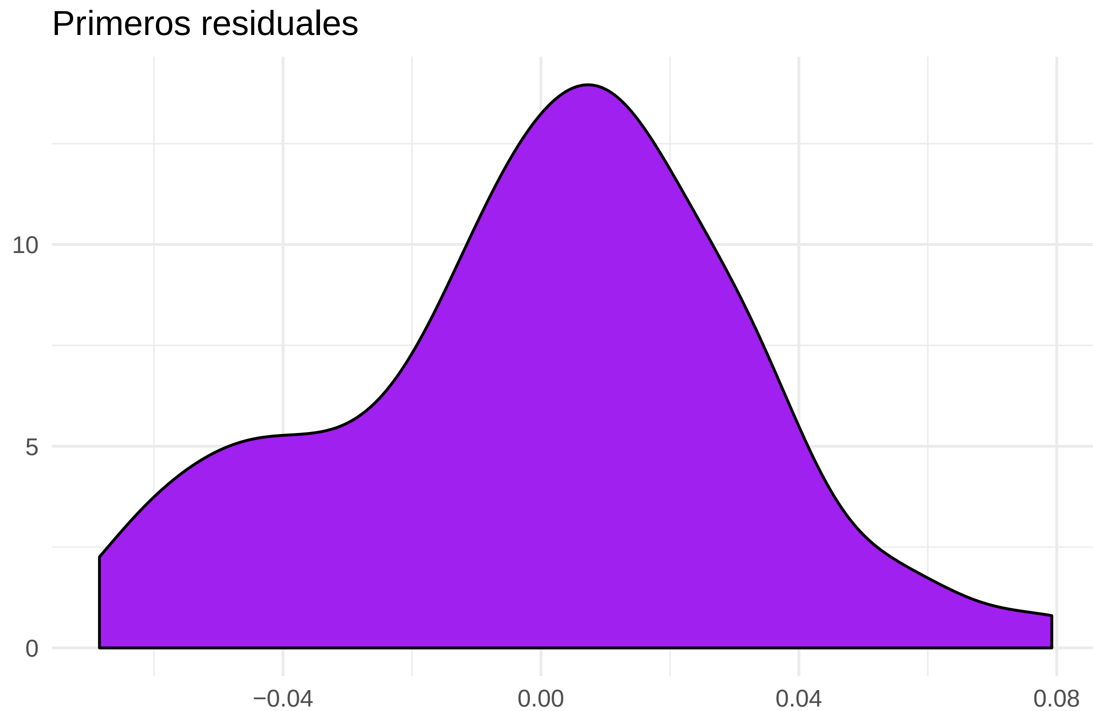
```
d <- datos
ingreso <- d$incl
consumo <- d$conl
modelo.MCE <- lm(ingreso~consumo )
residuales.1 <- modelo.MCE$residuals
acf(residuales.1, main='Primeros residuales no estacionarios')
```




```
pacf(residuales.1, main='Primeros residuales no estacionarios')
```

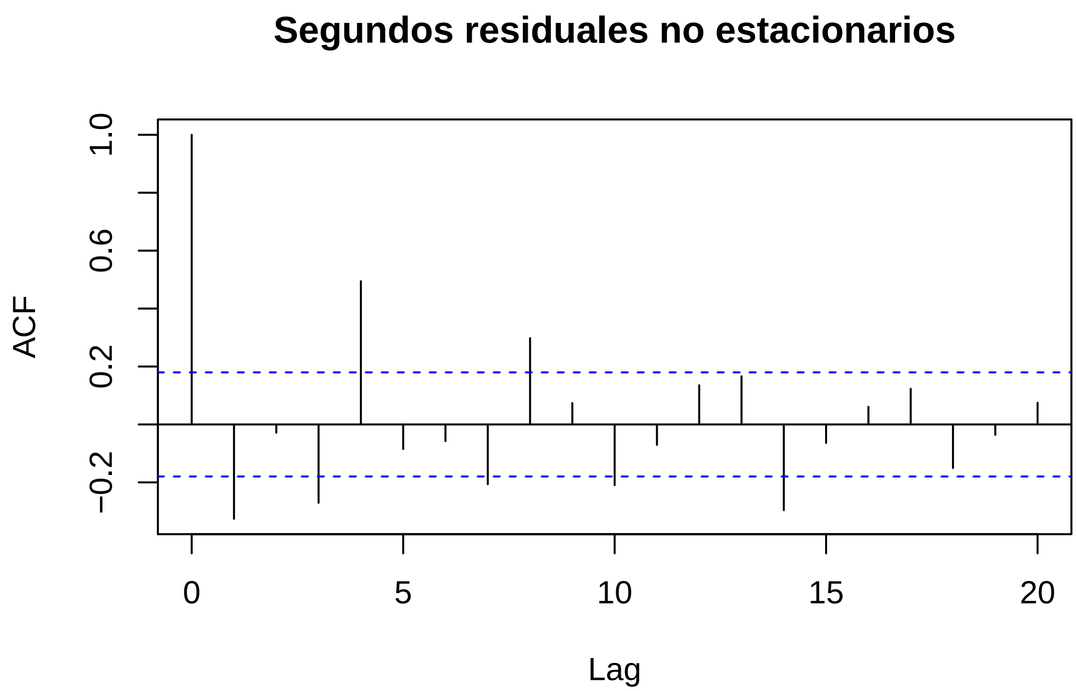



```r
library(ggplot2)
r <- data.frame(e1 = residuales.1)
d <- cbind(d, r )
ggplot(data=d, aes(x=e1 )) +
  geom_density(aes(y=..density.., fill=I('purple'))) +theme_minimal() +xlab('')+
  ylab('')+ggtitle('Primeros residuales')
```

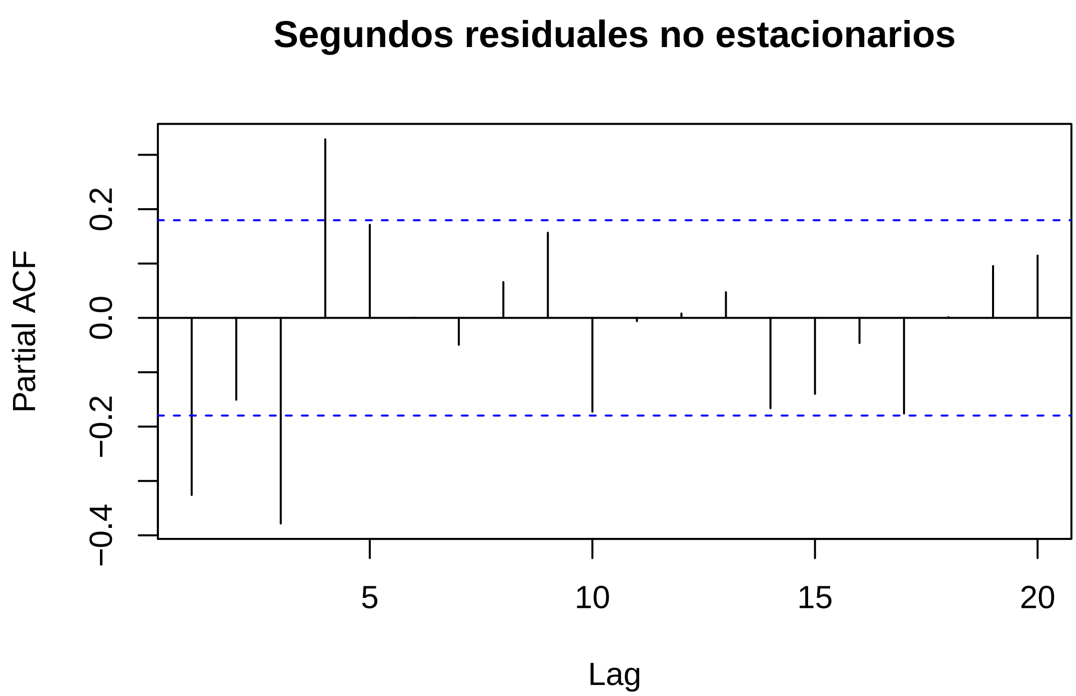


Realizamos la siguiente regresión

```
modelo.MCE.2 <- lm(diff(ingreso)~ diff(consumo) + e1[1:(dim(datos)[1]-1)],
                   data=d  )
residuales.2 <- modelo.MCE.2$residuals
acf(residuales.2, main='Segundos residuales no estacionarios')
```




```r
pacf(residuales.2, main='Segundos residuales no estacionarios')
```

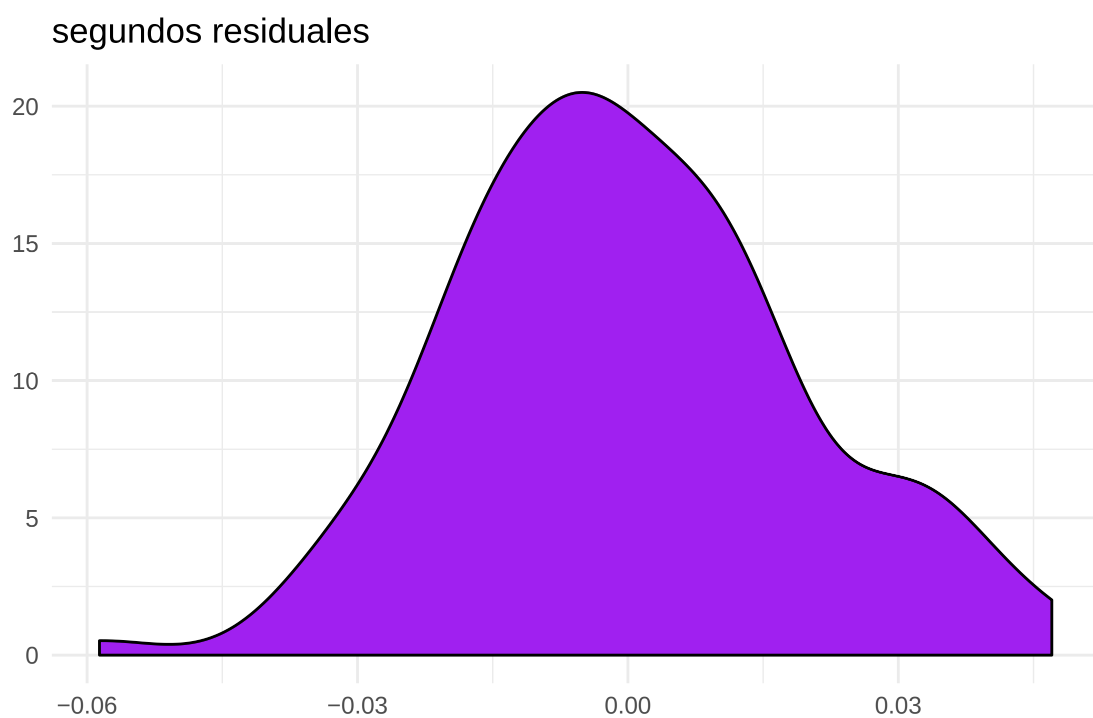



```r
summary(modelo.MCE.2)
```

```
##
```

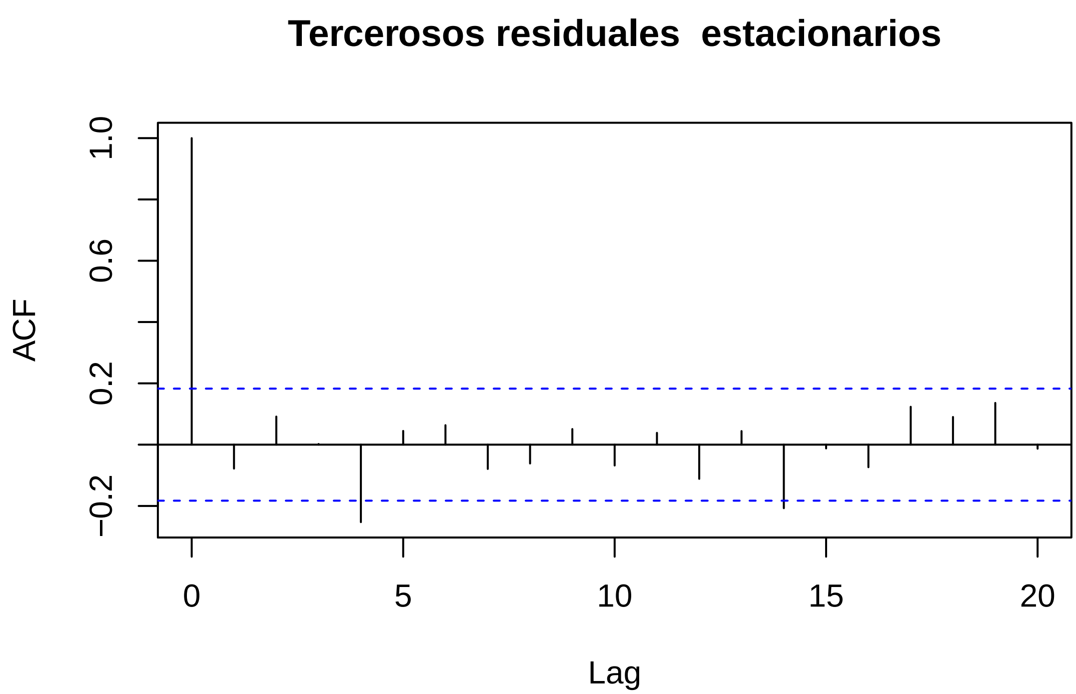
```
## Call:
## lm(formula = diff(ingreso) ~ diff(consumo) + e1[1:(dim(datos)[1] -
##     1)], data = d)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.058627 -0.012742 -0.002143  0.012199  0.047030
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.003414   0.001808   1.888   0.0615 .
## diff(consumo)              0.548533   0.041771  13.132  < 2e-16 ***
## e1[1:(dim(datos)[1] - 1)] -0.345590   0.074724  -4.625 9.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01953 on 116 degrees of freedom
## Multiple R-squared:  0.6281, Adjusted R-squared:  0.6217
## F-statistic: 97.97 on 2 and 116 DF,  p-value: < 2.2e-16
```

```r
library(ggplot2)
r <- data.frame(e2 = c(residuales.2, NA))
d <- cbind(d, e2 <- c(r$e2) )
ggplot(data=d, aes(x=e2 )) +
  geom_density(aes(y=..density.., fill=I('purple'))) +theme_minimal() +xlab('')+
  ylab('')+ggtitle('segundos residuales')
```

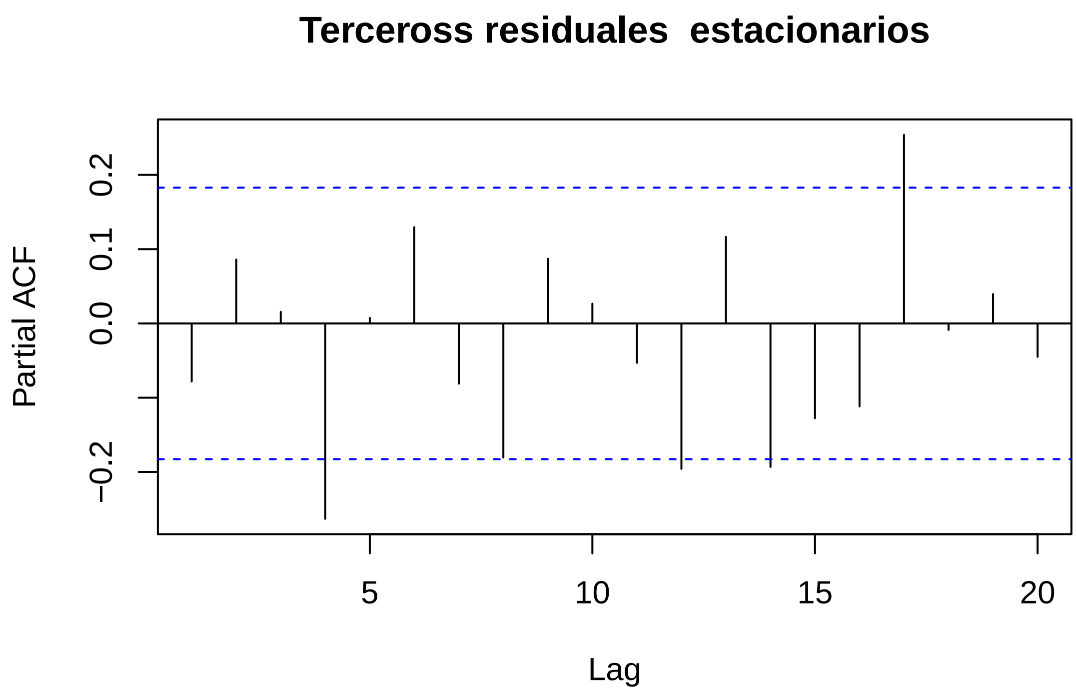


Hasta aqui tenemos que los residuales se parecen más a ruido blanco adémas de tener un coeficiente negativo en los terminos de las partes moviles y repetimos una vez más:

```r
modelo.MCE.3 <- lm(diff(diff(ingreso), 4)~ diff(diff(consumo), 4) +
                                        diff(e1[1:(dim(datos)[1]-1)], 4),
                   data=d  )
residuales.3 <- modelo.MCE.3$residuals
acf(residuales.3, main='Tercerosos residuales  estacionarios')
```




```r
pacf(residuales.3, main='Terceross residuales  estacionarios')
```

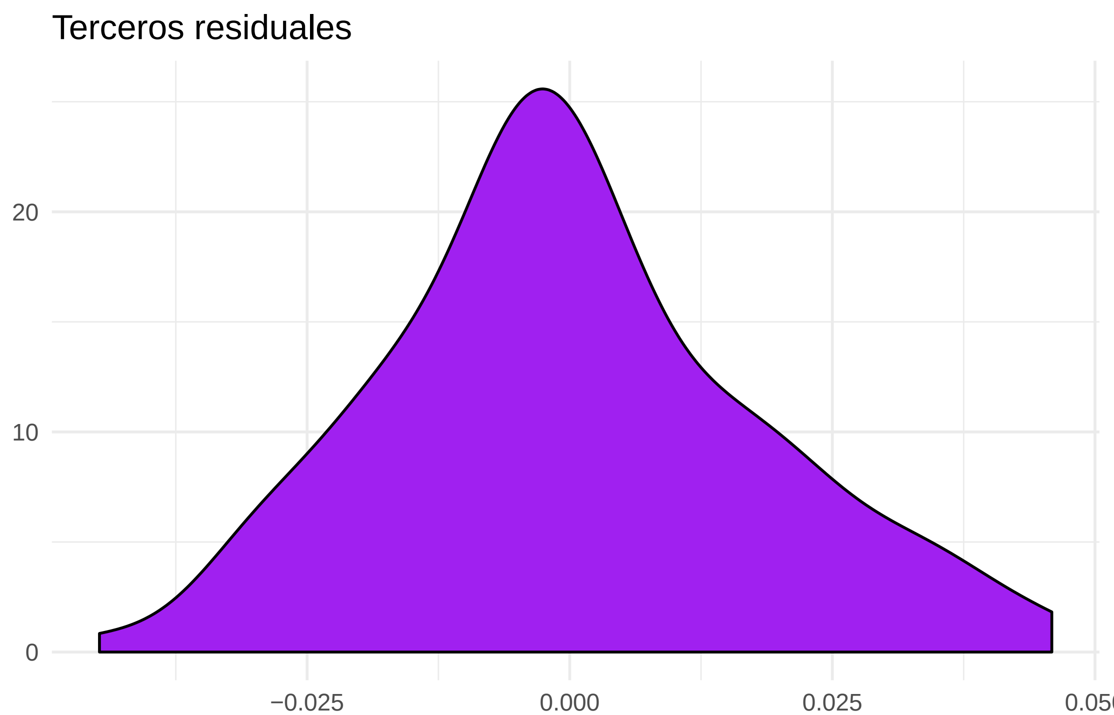



```r
summary(modelo.MCE.3)
```

```
##
## Call:
## lm(formula = diff(diff(ingreso), 4) ~ diff(diff(consumo), 4) +
##     diff(e1[1:(dim(datos)[1] - 1)], 4), data = d)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.044764 -0.011207 -0.002457  0.011369  0.045891
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        8.674e-05  1.683e-03   0.052    0.959
## diff(diff(consumo), 4)             7.015e-01  9.088e-02   7.718 5.33e-12
## diff(e1[1:(dim(datos)[1] - 1)], 4) -6.883e-01  8.624e-02  -7.982 1.37e-12
##
## (Intercept)
## diff(diff(consumo), 4)             ***
## diff(e1[1:(dim(datos)[1] - 1)], 4) ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01804 on 112 degrees of freedom
## Multiple R-squared:  0.4746, Adjusted R-squared:  0.4652
## F-statistic: 50.59 on 2 and 112 DF,  p-value: 2.221e-16
```

```r
library(ggplot2)
r <- data.frame(e3 = c(residuales.3,NA,NA,NA, NA, NA))
d <- cbind(d, e3=r$e3 )
```

```r
ggplot(data=d, aes(x=e3 )) +
  geom_density(aes(y=..density.., fill=I('purple'))) +theme_minimal() +xlab('')+
  ylab('')+ggtitle('Terceros residuales')
```



Hasta aqui que los coeficientes del termino MA siempre fueron negativos y que las sries presentan relaciones a largo plazo.

Descompondremos la matriz de varianzas para poder informarnos acerca del número de 'direcciones a largo placo' o bien que podemos interpretar como una variable latente.

```r
C <- cov(datos)
eigen <- eigen(C)
(var.expli <- eigen$values /sum(eigen$values))
```

```
## [1] 0.995394482 0.004605518
```