Código Tesis

Omar Díaz Landa

2019-03-12

Contents

Chapter 1

Prerequisitos

En esta sección encontraremos todos los paquetes necesarios para replicar la ejecución:

```
library(bookdown)
library(readxl)
library(ggplot2)
library(tidyverse)
library(plotly)
library(urca)
library(forecast)
library(lubridate)
library(vars)
library(texreg)
library(ggpubr)
```

Función para generar varias graficas en una sola

```
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
```

```
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                    ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

Funcion para graficar las series de tiempo

```
grafica_serie <- function(base_in, eje_y, titulo, titulo_y) {

  theme_set( theme_gray())

  tmp_gf <- ggplot(data=base_in, aes_string(x='year', y=eje_y)) +
                    geom_line(size = 1) +
                    ggtitle(titulo) +
                    ylab(titulo_y) +
                    scale_x_continuous("Año",  breaks = base_in$year) +
                    theme(plot.title = element_text( size=14, face="bold.italic",hjust=0.5),
                          axis.text.x = element_text(angle = 90, vjust = 0.5),
                          axis.title.y = element_text( size=10, face="bold"))

return(tmp_gf)
}

graf_latex <- function(grafica1, grafica2, nombre){

  ruta_dir <- '/Users/omardiaz/Google Drive/Tesis_Lic/V2/Tesis Latex/images'
  ruta_def <- paste(ruta_dir,nombre,sep="/")

  ggarrange(grafica1, grafica2, ncol = 2, nrow = 1) +
    ggsave(ruta_def, width = 15, height = 6)

  unlink(nombre)

}

graf_latex1 <- function(grafica1, nombre){
```

```r
  ruta_dir <- '/Users/omardiaz/Google Drive/Tesis_Lic/V2/Tesis Latex/images'
  ruta_def <- paste(ruta_dir,nombre,sep="/")

  ggarrange(grafica1, ncol = 1, nrow = 1) +
    ggsave(ruta_def, width = 15, height = 6)

  unlink(nombre)

}
```

```r
extract.summary.lm <- function (model, include.rsquared = TRUE, include.adjrs = TRUE,
                                include.nobs = TRUE, include.fstatistic = FALSE, include.rmse = TRUE,
                                ...)
{
  s <- model;
  names <- rownames(s$coef)
  co <- s$coef[, 1]
  se <- s$coef[, 2]
  pval <- s$coef[, 4]
  rs <- s$r.squared
  adj <- s$adj.r.squared
  n <- length(s$residuals)
  gof <- numeric()
  gof.names <- character()
  gof.decimal <- logical()
  if (include.rsquared == TRUE) {
    gof <- c(gof, rs)
    gof.names <- c(gof.names, "R$^2$")
    gof.decimal <- c(gof.decimal, TRUE)
  }
  if (include.adjrs == TRUE) {
    gof <- c(gof, adj)
    gof.names <- c(gof.names, "Adj. R$^2$")
    gof.decimal <- c(gof.decimal, TRUE)
  }
  if (include.nobs == TRUE) {
    gof <- c(gof, n)
    gof.names <- c(gof.names, "Num. obs.")
    gof.decimal <- c(gof.decimal, FALSE)
  }
  if (include.fstatistic == TRUE) {
    fstat <- s$fstatistic[[1]]
    gof <- c(gof, fstat)
    gof.names <- c(gof.names, "F statistic")
    gof.decimal <- c(gof.decimal, TRUE)
  }
  if (include.rmse == TRUE && !is.null(s$sigma[[1]])) {
    rmse <- s$sigma[[1]]
    gof <- c(gof, rmse)
    gof.names <- c(gof.names, "RMSE")
    gof.decimal <- c(gof.decimal, TRUE)
  }
  tr <- createTexreg(coef.names = names, coef = co, se = se,
```

```
                      pvalues = pval, gof.names = gof.names, gof = gof, gof.decimal = gof.decimal)
  return(tr)
}

setMethod("extract",  signature = 'summary.lm', definition = extract.summary.lm)
```

```
## in method for 'extract' with signature '"summary.lm"': no definition for class "summary.lm"
```

Chapter 2

Lectura de datos

Los datos se han obtenido a través e tres fuentes distintas, la API del banco mundial, la ley de ingresos de la federación y el presupuesto de egresos de la federación:

Las series obtenidas vía le banco mundial son:

Code	Indicator Name
SP.POP.TOTL	Population, total
SL.UEM.TOTL.NE.ZS	Unemployment, total (% of total labor force) (national estimate)
NY.GDP.MKTP.CN	GDP (current LCU)

Para más detalle acerca de la descripción y fuentes de estas series, se pueden consultar en el apartado de 14.

Las serie obtenida de la ley de ingresos de la federación corresponde al importe en pesos de la recaudación impositiva.

La series obtenidas a través del presupuesto de egresos de la federación son el gasto en salud pública y el gasto en educación pública. Tanto la recaudación impositiva, como el gasto en educación y salud pública serán representados como porcentaje del PIB a precios corrientes.

Extraemos de las series anteriormente mencionadas, todos los datos disponibles de manera anual para México a partir de 1991 hasta 2016. Esto es debido a que se trata del horizonte de tiempo con mayor cantidad de datos informados.

```r
series_db <- read_excel("Datos/Series Tesis Recolección.xlsx", sheet = "Datos Finales")
series_db
```

```
## # A tibble: 26 x 9
##    country  year Poblacion_Total Desempleo_Total GDP_corriente
##    <chr>   <dbl>           <dbl>           <dbl>         <dbl>
##  1 Mexico   1991        87071512            3.05  949148000000
##  2 Mexico   1992        88828310            3.1  1125334000000
##  3 Mexico   1993        90600453            3.21 1560093286000
##  4 Mexico   1994        92349147            4.25 1781422460000
##  5 Mexico   1995        94045579            6.89 2311458453000
##  6 Mexico   1996        95687452            5.25 3123167939000
##  7 Mexico   1997        97281739            4.06 3962524166000
##  8 Mexico   1998        98821456            3.57 4810123454000
##  9 Mexico   1999       100300579            2.49 5738466369000
## 10 Mexico   2000       101719673            2.56 6693683014000
```

```
## # ... with 16 more rows, and 4 more variables: GDP_constante <dbl>,
## #   Recaudacion_Impositiva_PorcGDP <dbl>, Gasto_Educacion_PorcGDP <dbl>,
## #   Gasto_Salud_PorcGDP <dbl>
```

Utilizaremos el logaritmo como función estabilizadora de varianza para el PIB a precios corrientes, aprecios constantes y para la población total.

```
series_db$log_GDP_corriente <- log(series_db$GDP_corriente)
series_db$log_GDP_constante <- log(series_db$GDP_constante)
series_db$log_Poblacion_Total <- log(series_db$Poblacion_Total)
series_db
```

```
## # A tibble: 26 x 12
##    country  year Poblacion_Total Desempleo_Total GDP_corriente
##    <chr>   <dbl>           <dbl>           <dbl>         <dbl>
##  1 Mexico   1991        87071512            3.05  949148000000
##  2 Mexico   1992        88828310            3.1  1125334000000
##  3 Mexico   1993        90600453            3.21 1560093286000
##  4 Mexico   1994        92349147            4.25 1781422460000
##  5 Mexico   1995        94045579            6.89 2311458453000
##  6 Mexico   1996        95687452            5.25 3123167939000
##  7 Mexico   1997        97281739            4.06 3962524166000
##  8 Mexico   1998        98821456            3.57 4810123454000
##  9 Mexico   1999       100300579            2.49 5738466369000
## 10 Mexico   2000       101719673            2.56 6693683014000
## # ... with 16 more rows, and 7 more variables: GDP_constante <dbl>,
## #   Recaudacion_Impositiva_PorcGDP <dbl>, Gasto_Educacion_PorcGDP <dbl>,
## #   Gasto_Salud_PorcGDP <dbl>, log_GDP_corriente <dbl>,
## #   log_GDP_constante <dbl>, log_Poblacion_Total <dbl>
```

2.1 Graficamos las series de datos

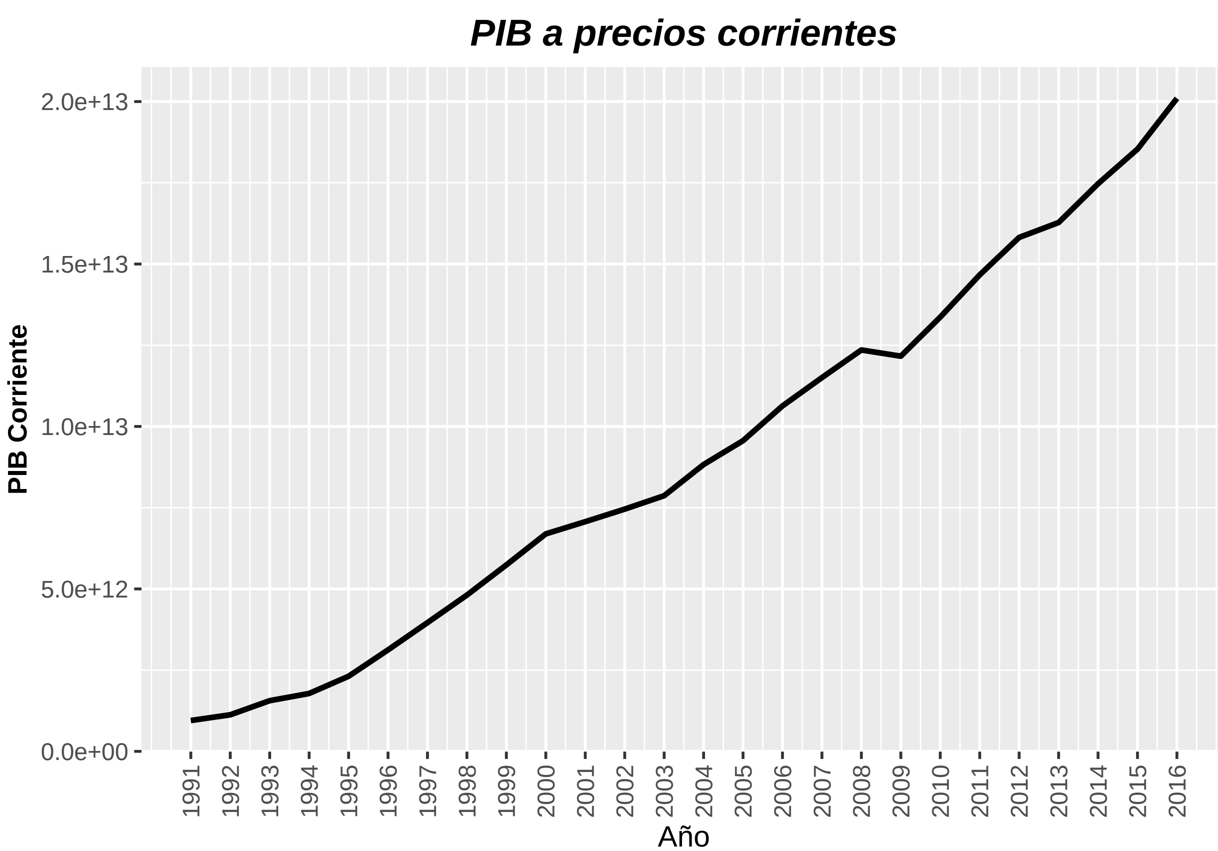
2.1.1 Producto Interno Bruto

Es muy importante antes de realizar cualquier análisis, primero realizar un análisis exploratorio, que en el caso de series de tiempo se reduce a realizar gráficas de las mismas:

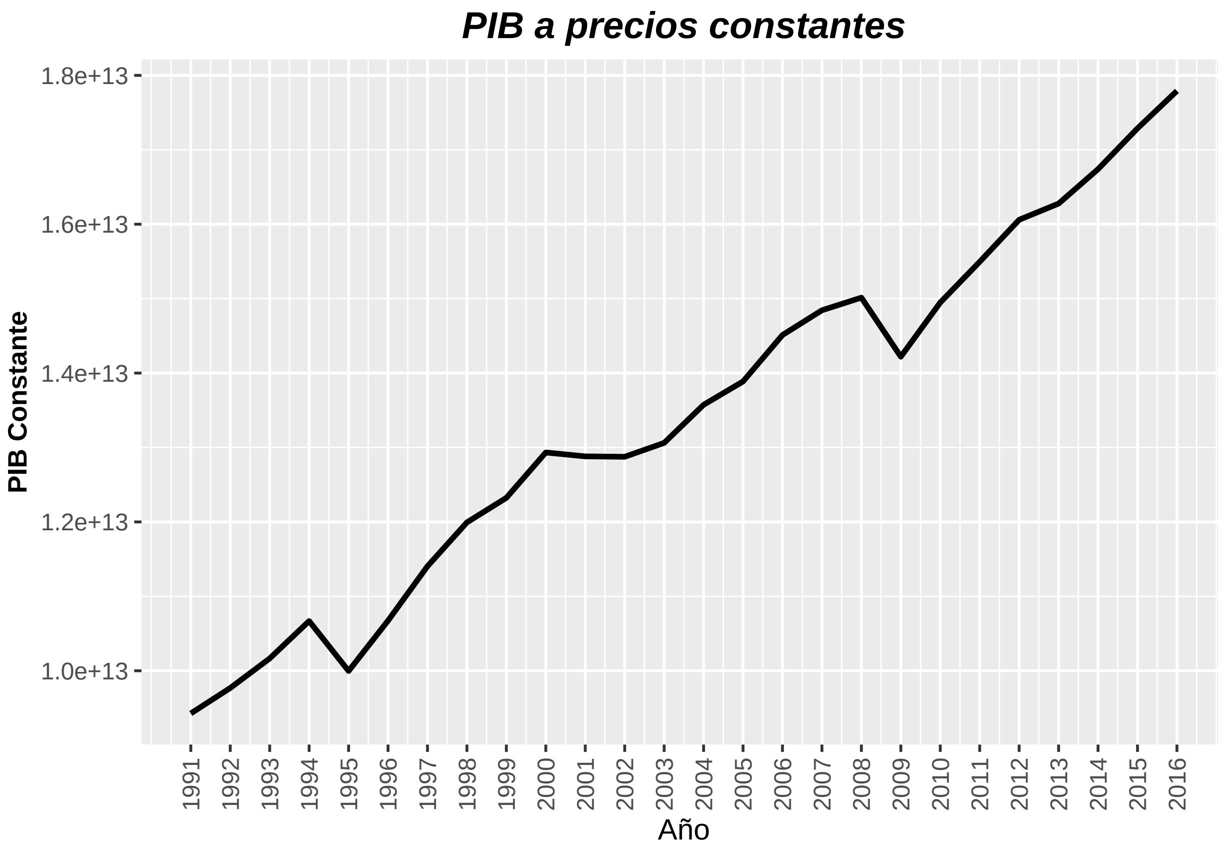
```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'GDP_corriente',
                     titulo = 'PIB a precios corrientes',
                     titulo_y = 'PIB Corriente')

gf1_1 <- grafica_serie(base_in = series_db,
                     eje_y = 'GDP_constante',
                     titulo = 'PIB a precios constantes',
                     titulo_y = 'PIB Constante')

graf_latex(grafica1 = gf1,
          grafica2 = gf1_1,
          nombre = 'PIB_normal.pdf')

gf2 <- grafica_serie(base_in = series_db,
                     eje_y = 'log_GDP_corriente',
                     titulo = 'Logaritmo del PIB a precios corrientes',
```

```
                      titulo_y = 'log(PIB Corriente)')

gf2_2 <- grafica_serie(base_in = series_db,
                      eje_y = 'log_GDP_constante',
                      titulo = 'Logaritmo del PIB a precios constantes',
                      titulo_y = 'log(PIB Constante)')


graf_latex(grafica1 = gf2,
          grafica2 = gf2_2,
          nombre = 'log_PIB.pdf')
```

```
gf1
```



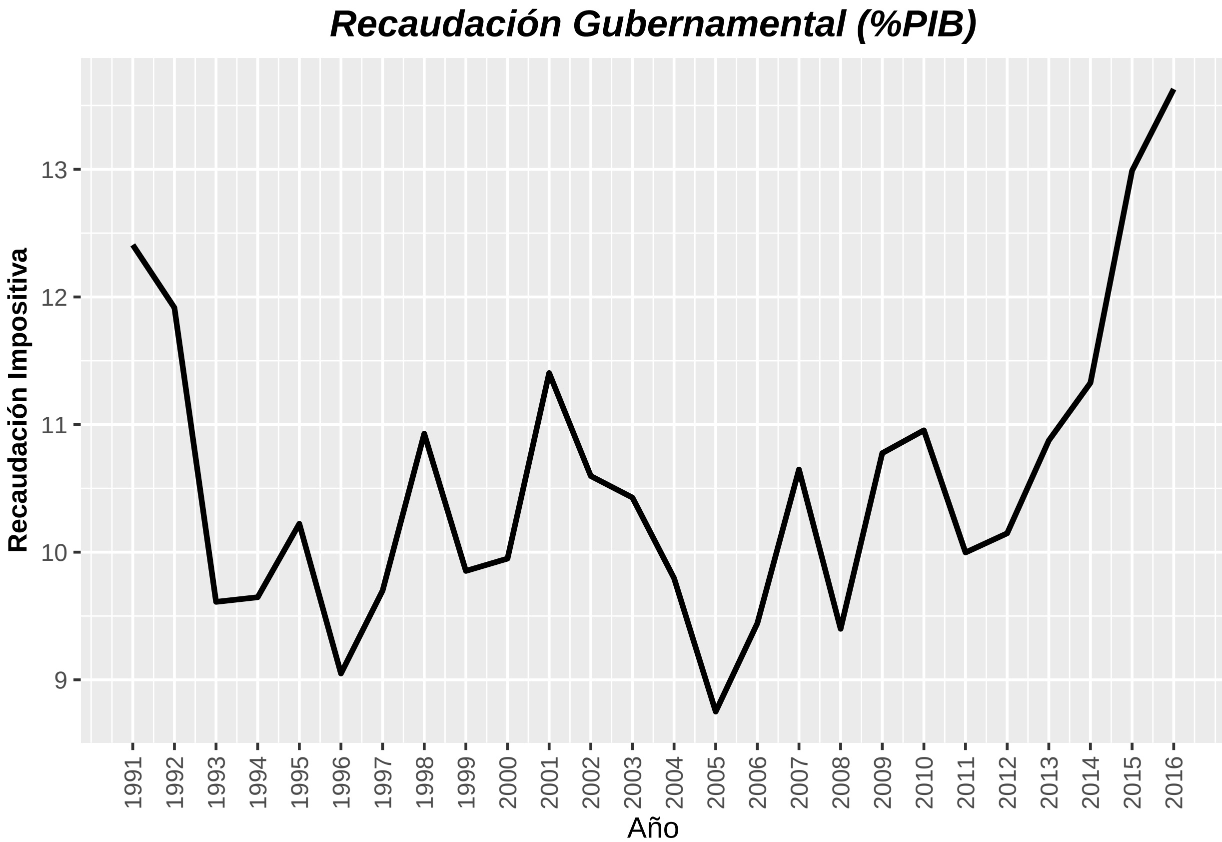

```
gf1_1
```



2.1.2 Recaudación Fiscal

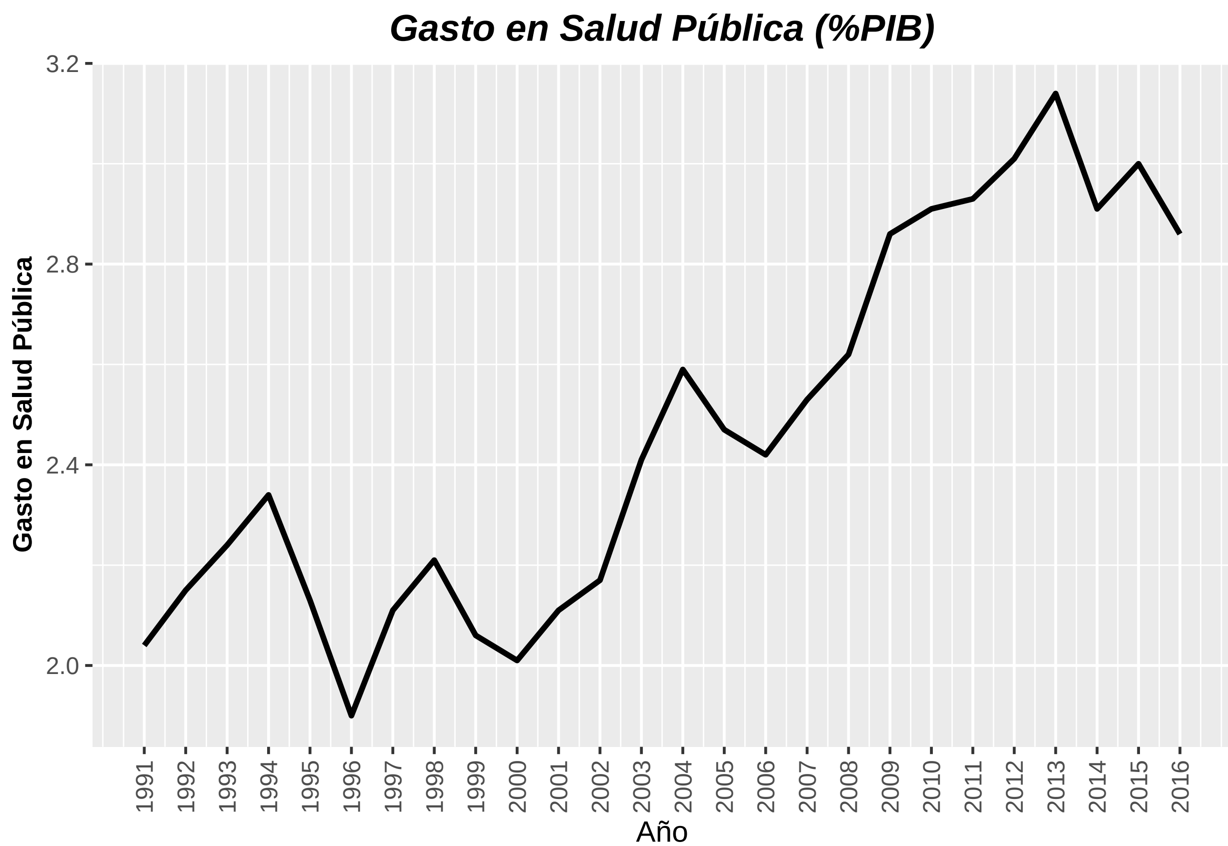
A continuación observamos la serie de la recaudación impositiva como porcentaje del producto interno bruto:

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Recaudacion_Impositiva_PorcGDP',
                     titulo = 'Recaudación Gubernamental (%PIB)',
                     titulo_y = 'Recaudación Impositiva')

graf_latex1(grafica1 = gf1,
          nombre = 'Recaudacion.pdf')

gf1
```



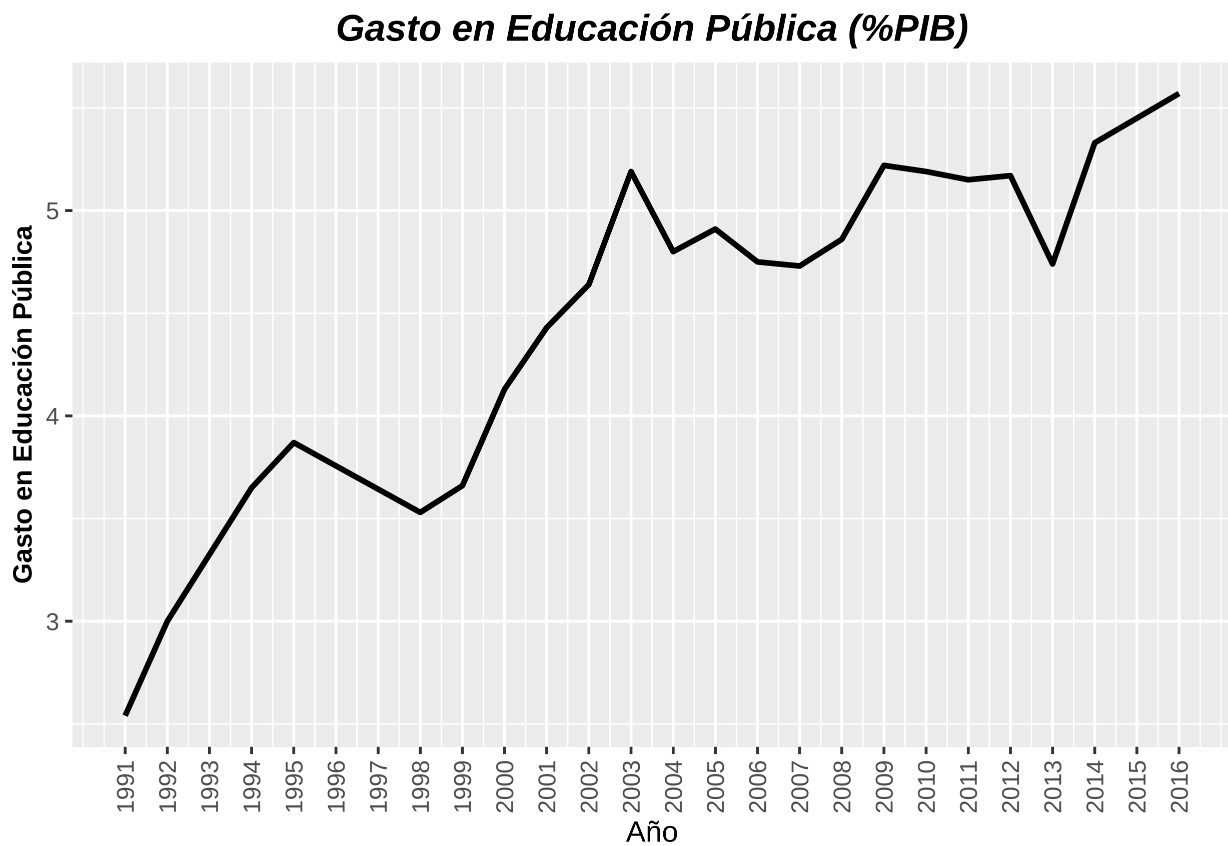
2.1.3 Gasto en Salud Pública

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Gasto_Salud_PorcGDP',
                     titulo = 'Gasto en Salud Pública (%PIB)',
                     titulo_y = 'Gasto en Salud Pública')

graf_latex1(grafica1 = gf1,
          nombre = 'Gasto_Salud.pdf')

gf1
```



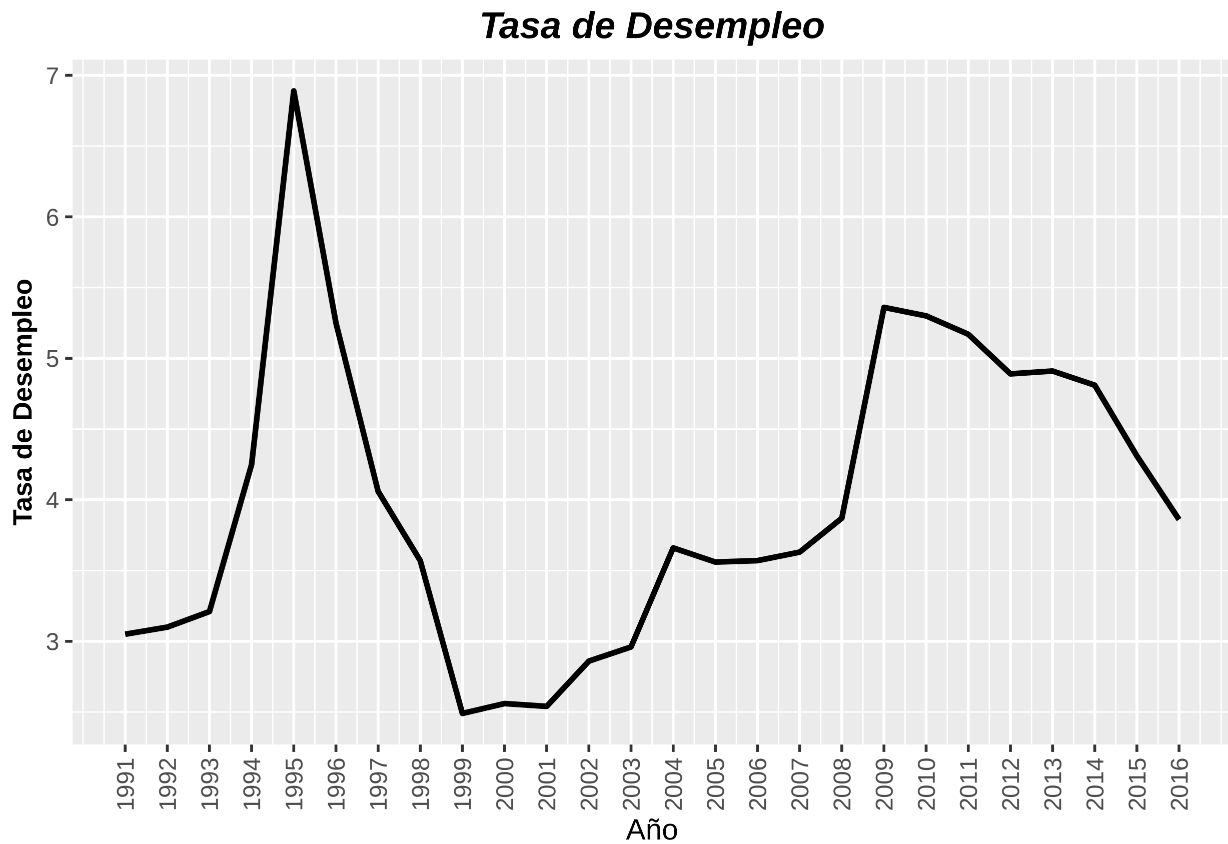
2.1.4 Gasto en Educación Pública

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Gasto_Educacion_PorcGDP',
                     titulo = 'Gasto en Educación Pública (%PIB)',
                     titulo_y = 'Gasto en Educación Pública')

graf_latex1(grafica1 = gf1,
          nombre = 'Gasto_Educacion.pdf')

gf1
```



Tasa de Desempleo

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Desempleo_Total',
                     titulo = 'Tasa de Desempleo',
                     titulo_y = 'Tasa de Desempleo')

graf_latex1(grafica1 = gf1,
          nombre = 'Tasa_Desempleo.pdf')

gf1
```


Tasa de Desempleo
Tasa de Desempleo
7
6
5
4
3
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Año

Chapter 3

Orden de Integración (Teoría)

En el modelo

$$y_t = a_1 y_{t-1} + \epsilon_t$$

Al restar por y_{t-1} en ambos lados de la ecuación, podemos llegar a la siguiente expresión equivalente:

$$\nabla y_t = \gamma y_{t-1} + \epsilon_t$$

donde $\gamma = a_1 - 1$. Por lo tanto, realizar la prueba de hipótesis $a_1 = 1$ es equivalente a probar que $\gamma = 0$. Dickey y Fuller consideran tres tipos de ecuaciones de regresión que pueden ser utilizadas para probar la presencia de raíces unitarias:

1) $\nabla y_t = \gamma y_{t-1} + \epsilon_t$
2) $\nabla y_t = a_0 + \gamma y_{t-1} + \epsilon_t$
3) $\nabla y_t = a_0 + \gamma y_{t-1} + a_2 t + \epsilon_t$

El primero de ellos es un modelo puro de caminata aleatoria, el segundo agrega un intercepto o drift y el tercero incluye tanto un drift como una tendencia lineal temporal.

El parámetro de interés en todas estas ecuaciones es γ; Si $\gamma = 0$ entonces la serie y_t contiene una raíz unitaria. La prueba se reduce en estimar las ecuaciones anteriormente mencionadas via OLS, de tal manera que se obtenga el valor estimado del parametro γ y su error estandar asociado. Dicho estadistico t se debe comparar con sus correspondientes valores de acuerdo con las tablas creadas por Dicckey-Fuller.

Esta metodología se sigue para cualquiera de las tres ecuaciones enunciadas anteriormente, sin embargo, el detalle es que cambian los valores criticos de Dickey-Fuller dependiendo la ecuacion utilizada para estimar el valor del parametro γ. Dickey-Fuller concluyeron que los valores criticos para $\gamma = 0$ dependen del tipo de regresión y el tamaño de la muestra.

Los estadísticos τ_1, τ_2, τ_3 son los estadísticos apropiados para usar en las ecuaciones (1), (2) y (3) respectivamente. Los valores reportados en las tablas de Dickey-Fuller permiten al investigador determinar si aceptar o rechazar la hipótesis nula $\gamma = 0$ unicamente.

Adicionalmente Dickey and Fuller crearon tres F-estadisticos adicionales llamados ϕ_1, ϕ_2, ϕ_3 para probar pruebas de hipotesis conjuntas:

- El estadistico ϕ_1 .- Emplea la ecuacion numero 2 para probar la hipotesis nula conjunta $\gamma = a_0 = 0$
- El estadistico ϕ_2 .- Emplea la ecuacion numero 3 para probar la hipotesis nula conjunta $\gamma = a_0 = a_2 = 0$
- El estadistico ϕ_3 .- Emplea la ecuacion numero 3 para probar la hipotesis nula conjunta $\gamma = a_2 = 0$

En donde los tres estadisticos se construyen igual que cualquier prueba F:

$$\phi_i = \frac{[RSS_{restricted} - RSS_{unrestricted}]r}{RSS_{unrestricted}(T-k)}$$

donde RSS son las sumas al cuadrado de los residuales del modelo restringido vs el no restringido, r es el numero de restricciones, T el numero de observaciones usbales y k el numero de parametros estimados en el modelo sin restricciones.

Por lo tanto, la hipotesis nula es que los datos fueron generados por el modelo con restricciones y la alternativa es que los datos fueron generados por el modelo sin restricciones. Si la restriccion no se cumple, la suma de los cuadrados de ambos modelos deberian ser muy similares y por lo tanto ϕ_i será pequeña. Asi pues, si el valor de ϕ es grande quiere decir que se rechaza la hipotesis nula.

http://stats.stackexchange.com/questions/24072/interpreting-rs-ur-df-dickey-fuller-unit-root-test-results

Usaremos la prueba ADF sobre la serie del consumo en UK usando datos trimestrales del perido 1966:Q4-1991:Q2. La serie del consumo esta ajustada estacionalmente a precios de 1985 y expresadas en su logaritmo natural.

Como un primer paso, una regresión con una constante y tendencia temporal será estimada, se agregarán 3 lags a la estimación para evitar la presencia de autocorrelaciones y autocorrelaciones parciales y asegurar un proceso esférico del error. Incluir el cuarto lag resultaba ser no significativo, mientras que incluir solo dos lags no eran suficientes para alcanzar errores serialmente no correlacionados.

```
data("Raotbl3")
attach(Raotbl3)
```

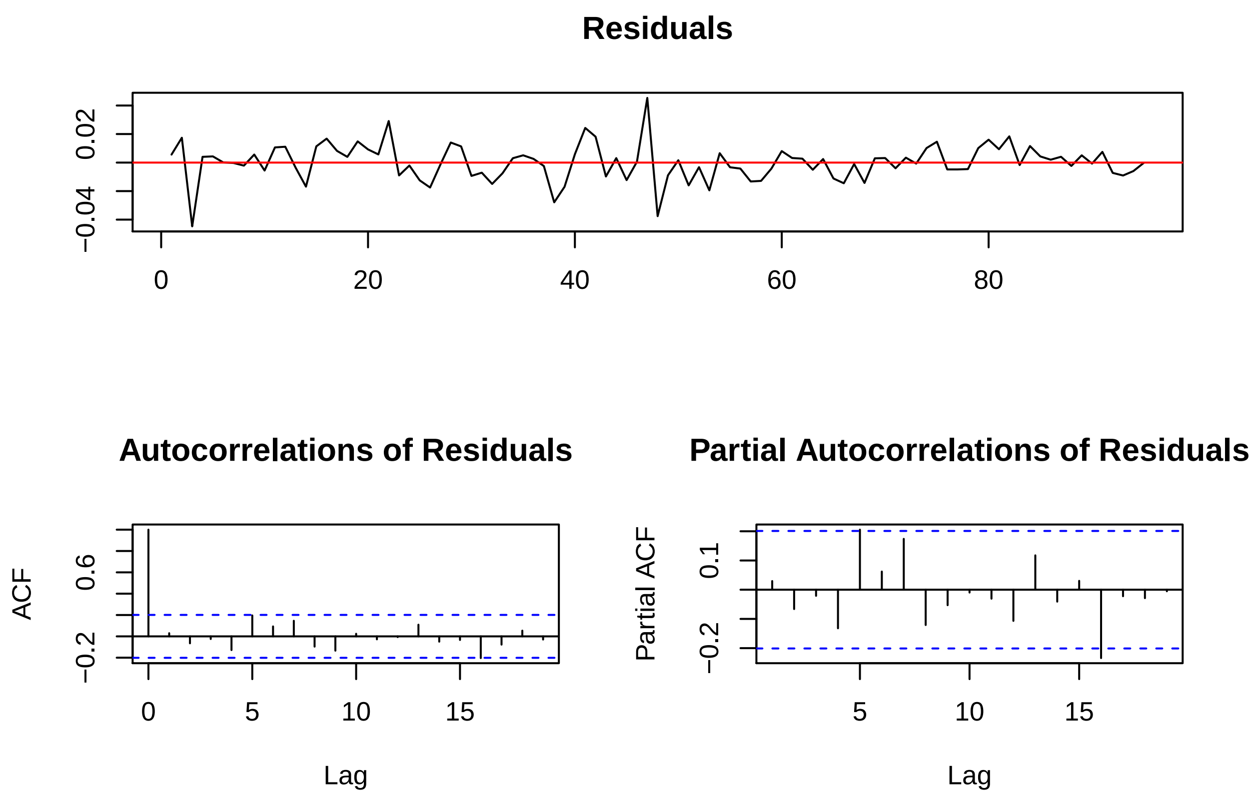
```
## The following objects are masked from Raotbl3 (pos = 8):
##
##     dd682, dd792, dd883, lc, li, lw
```

```
lc <- ts(lc, start=c(1966,4), end=c(1991,2),frequency=4)
lc.ct <- ur.df(lc,lags=3,type='trend')
summary(lc.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.044714 -0.006525  0.000129  0.006225  0.045353
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7976591  0.3547775   2.248   0.0270 *
## z.lag.1     -0.0758706  0.0338880  -2.239   0.0277 *
## tt           0.0004915  0.0002159   2.277   0.0252 *
## z.diff.lag1 -0.1063957  0.1006744  -1.057   0.2934
## z.diff.lag2  0.2011373  0.1012373   1.987   0.0500 .
```

```
## z.diff.lag3  0.2998586  0.1020548   2.938   0.0042 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01307 on 89 degrees of freedom
## Multiple R-squared:  0.1472, Adjusted R-squared:  0.09924
## F-statistic: 3.071 on 5 and 89 DF,  p-value: 0.01325
##
##
## Value of test-statistic is: -2.2389 3.7382 2.5972
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
plot(lc.ct)
```



Ahora bien la hipótesis ϕ_3 es probada bajo una usual prueba F, es decir, $\phi_3 = (a_0, \gamma, a_2) = (a_0, 0, 0)$. Esto es, se han colocado restricciones igual a cero a la tendencia temporal y el lag del la variable.

El valor del estadístico es

```
lc.ct@teststat
```

```
##                tau3     phi2     phi3
## statistic -2.238865 3.738151 2.597211
```

Debemos recordar que se deben consular los valores críticos propuestas por Dickey and Fuller. Los valores críticos para una muestra de tamaño 100 y niveles de significancia del 10%,5% y 1% se muestran a continuación

```
lc.ct@cval
```

```
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Por lo tanto, la hipótesis nula no puede ser rechazada, lo cual implica que la serie contiene una raíz unitaria. Esto puede ser reiterado con el estadístico τ_3 con valor de -2.24 y para la variable *z.lag.1*. Los valore críticos relevantes que debemos utilizar ahora son los de Fuller[1976], los cuales se muestran para una muestra de tamaño 100.

Luego entonces, la presencia de una raíz unitaria no puede rechazada. El siguiente paso, es probar si la serie es una caminata aleatoria con o sin drift (constante). El estadístico relevante es ϕ_2 ($\gamma = a_0 = a_2 = 0$) el cual tiene un valor de 3.74, con valores críticos de 4.16, 4.88 y 6.50 para niveles de significancia de 10%, 5% y 1% respectivamente. La conclusión es entonces que la serie se comporta como una caminata aleatoria pura.

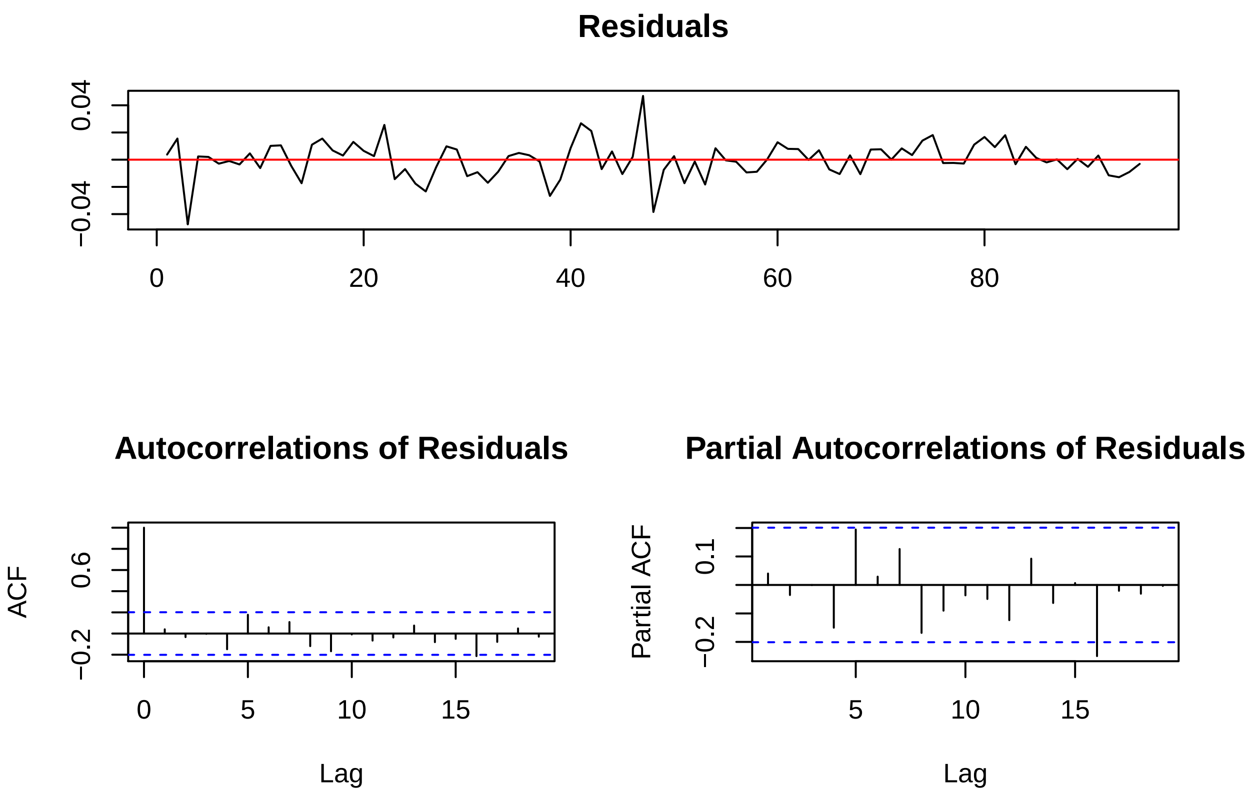
Uno procede entonces a estimar la ecuación $\nabla y_t = a_0 + \gamma y_{t-1} + \epsilon_t$ basado en los resultados obtenidos por la prueba ϕ_3. Los resultados se muestran a continuación:

```
lc.co <- ur.df(lc,lags=3,type='drift')
summary(lc.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.047547 -0.007071  0.000265  0.007731  0.046880
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0123237  0.0851358   0.145   0.8852
## z.lag.1     -0.0007356  0.0079043  -0.093   0.9261
## z.diff.lag1 -0.1433015  0.1016454  -1.410   0.1620
## z.diff.lag2  0.1615256  0.1020242   1.583   0.1169
## z.diff.lag3  0.2585280  0.1027364   2.516   0.0136 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01337 on 90 degrees of freedom
## Multiple R-squared:  0.09747,    Adjusted R-squared:  0.05735
## F-statistic:  2.43 on 4 and 90 DF,  p-value: 0.05335
##
##
## Value of test-statistic is: -0.0931 2.8806
##
## Critical values for test statistics:
```

```
##        1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

```
plot(lc.co)
```



Con el fin de completar la prueba, ahora se prueba si en este modelo un término constante hace falta. Las pruebas se muestran a continuación

```
lc.co@teststat
```

```
##                   tau2     phi1
## statistic -0.09306748 2.880589
```

```
lc.co@cval
```

```
##        1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

El valor del estadístico ϕ_1 que prueba $\gamma = a_0 = 0$ es 2.88, el cual resulta ser no significativo comparado con los valores críticos mostrados.

Por lo tanto, se puede concluir que la serie contiene un raíz unitaria pero no contiene ni tendencia temporal ni tendencia constante en el proceso generador de los datos.

Finalmente, se probará si diferenciando la serie una vez es suficiente para alcanzar estacionariedad. La prueba se logra utilizando como insumo para la regresión a la serie diferenciada una vez.

```
lc2 <- diff(lc)
lc2.ct <- ur.df(lc2,type="trend",lags=3)
summary(lc2.ct)
```

```
##
```

```
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.045039 -0.007870  0.000013  0.007807  0.046403
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.864e-03  3.051e-03   1.266   0.2087
## z.lag.1     -8.826e-01  2.013e-01  -4.385  3.2e-05 ***
## tt           3.186e-05  5.112e-05   0.623   0.5348
## z.diff.lag1 -2.253e-01  1.873e-01  -1.203   0.2321
## z.diff.lag2 -4.668e-02  1.600e-01  -0.292   0.7711
## z.diff.lag3  1.775e-01  1.057e-01   1.679   0.0967 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01329 on 88 degrees of freedom
## Multiple R-squared:  0.6147, Adjusted R-squared:  0.5929
## F-statistic: 28.08 on 5 and 88 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -4.3853 6.4477 9.6164
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

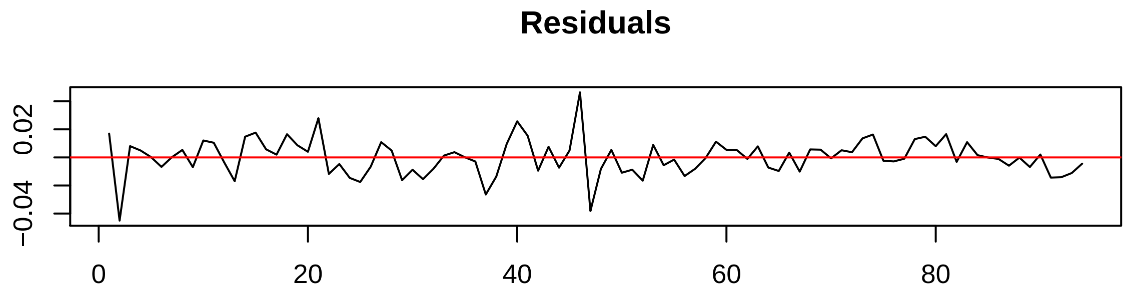
```
lc2.ct@teststat
```

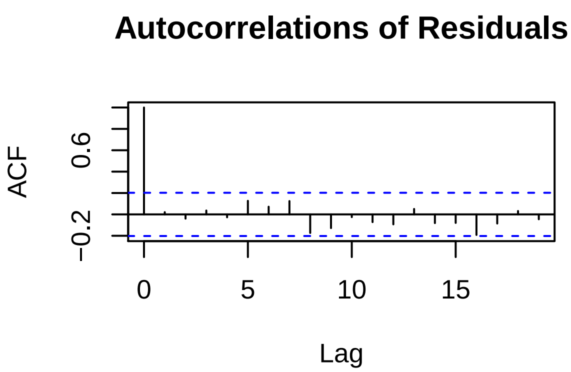
```
##                tau3     phi2     phi3
## statistic -4.385326 6.447681 9.616431
```

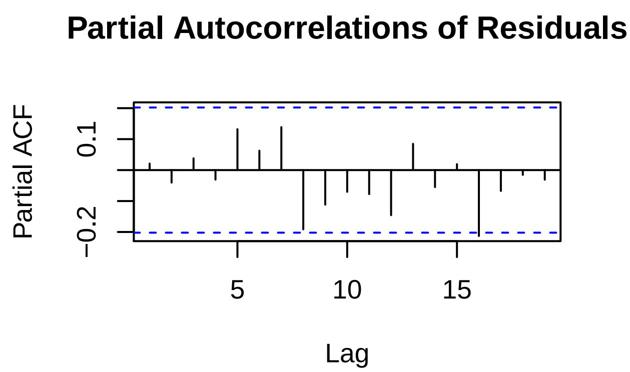
```
lc2.ct@cval
```

```
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

```
plot(lc2.ct)
```







La hipótesis de que el consumo es $I(2)$ puede ser descartado rápidamente dado el estadístico t con valor de -4.39.

Debe notarse que las pruebas de Dickey-Fuller asumen que los errores son independientes y tienen varianza constante. Esto genera 4 importantes problemas relacionados con el hecho de que no conocemos el verdadero proceso generador de los datos:

1) El verdadero proceso generador de los datos puede contener componentes autorregresivas como componentes de promedios móviles
2) No se puede estimar correctamente γ y su error estándar amenos que todos los términos autorregresivos sean incluidos en la ecuación a estimar (seleccionar el lag length apropiado)
3) El hecho de que la preuba de Dickey-Fuller considera únicamente una raíz unitaria, por lo que llevar una serie de un orden de integración mayor, requeire de diferenciarla tantas veces como sea necesario
4) No tenemos certeza de si incluir un intercepto o una tendencia temporal a la ecuación

El primero de los puntos se resuelve fácilmente ya que un modelo MA invertible (si sus raíces caen fuera del círculo unitario) puede ser expresado en un modelo autorregresivo con lags infinitos. Afortunadamente, Said y Dickey (1984) demuestran que un proceso ARIMA(p,1,q) puede ser correctamente aproximado por un modelo ARIMA(n,1,0) autorregresivo de orden $T^{1/3}$.

El segundo punto es muy importante ya que incluir demasidos lags reduce el poder de las pruebas estadísticas para rechazar la hipótesis nula de que existe raíz unitara, ya que un mayor número de lags necesita un mayor número de parámetros a estimar y una pérdida en grados de libertad. Los grados de libertad se reducen ya que el número de parámetros a estimar aumenta y por que el número de observaciones utilizables se reduce (perdemos una por cada lag). Por otro lado, definir pocos lags provoca que no capturemos apropiadamente el error process, por lo que γ y su error estandar no estarán bien estimados. Para solucionar este tema se sugeire empezar por un número suficientemente grande de lags e ir reduciendo hasta que el i-ésimo lag sea estadísticamente significativo de acuerdo a las preubas t. Una vez que el lag ha sido determinado, se procede a realizar un diagnóstico, graficar los residuales es el diagnóstico más importante. No debe haber evidencia de cambios estructurales ni correlación serial.

El tercer punto se puede atacar de manera secuencial tomando como input la serie diferenciada, y el proceso se repite hasta que se alcance la estacionariedad.

El cuarto punto

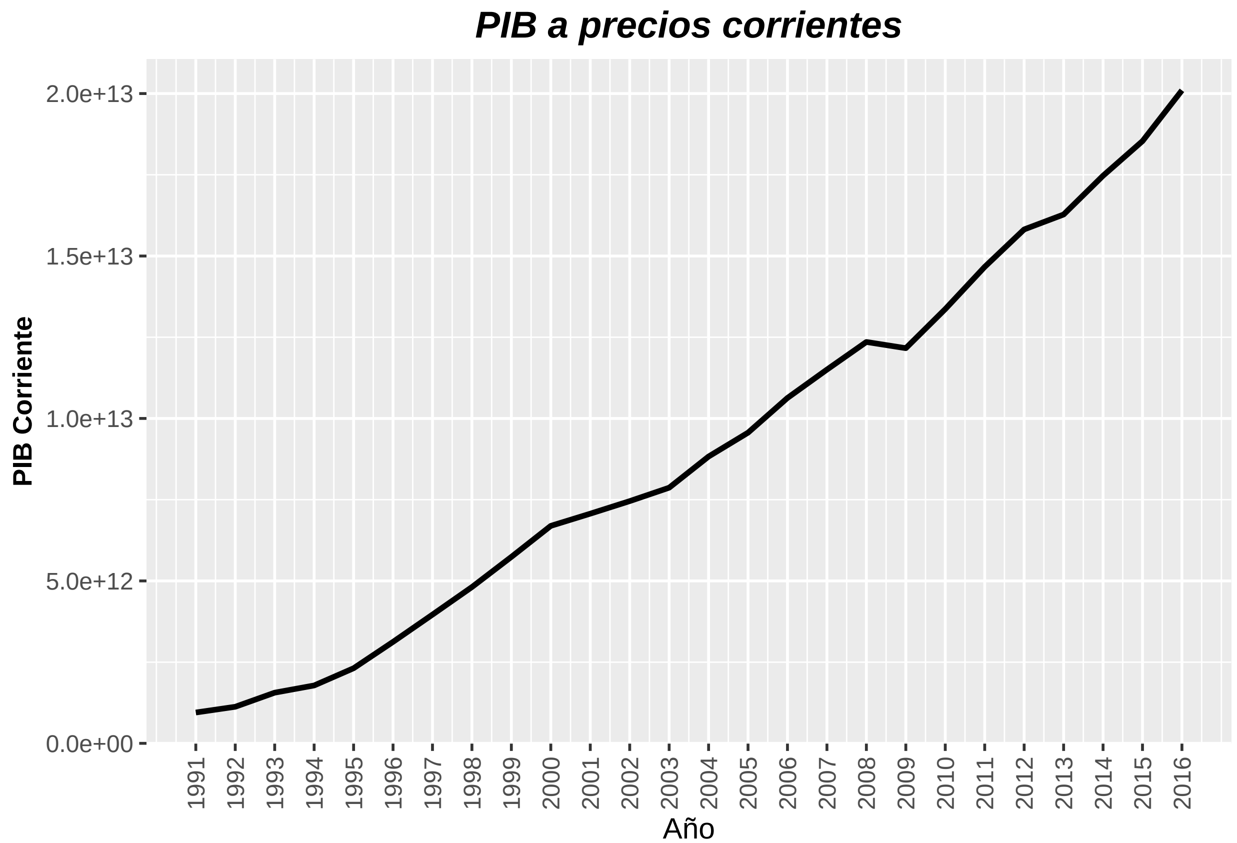
https://bookdown.org/ccolonescu/RPoE4/vec-and-var-models.html

Chapter 4

Orden Integración PIB a precios corrientes

La serie de PIB a precios corrientes parece ser una serie con un orden de integración, esto será corroborado con las pruebas de hipótesis presentadas por Dickey & Fuller, tal y como ya se ha presentado en la sección previa.

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'GDP_corriente',
                     titulo = 'PIB a precios corrientes',
                     titulo_y = 'PIB Corriente')
gf1
```

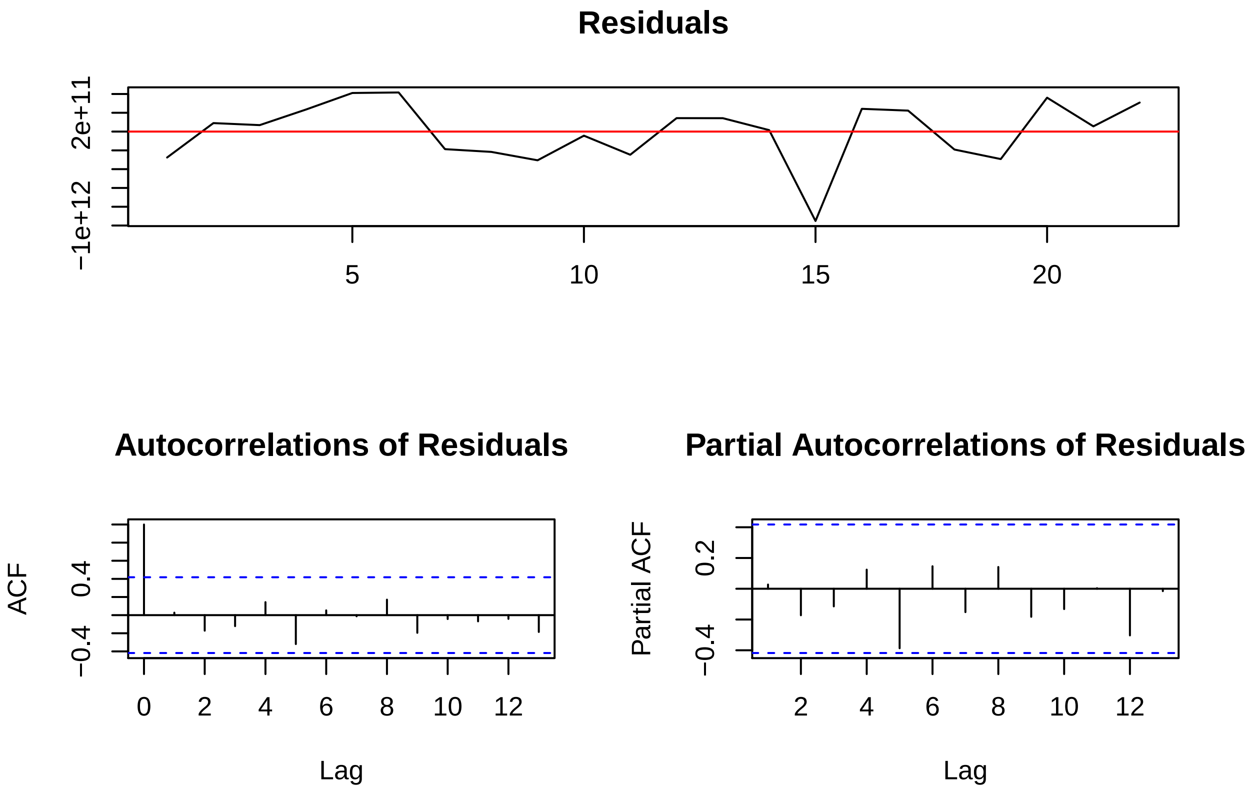


En primer lugar necesitamos realizar la lectura de los datos como un objeto *ts* y posteriormente una regresión con una constante y tendencia temporal será estimada, en este caso en particular, no será necesario agregar lags a la estimación ya que no hay evidencia de presencia de autocorrelaciones y autocorrelaciones parciales diferentes de cero, por lo que podemos asegurar un proceso esférico del error. Incluir desde uno hasta el cuarto lag resultaban ser no significativos.

```
pib_corr <- ts(series_db$GDP_corriente,start = 1991, end = 2016, frequency = 1)
pib_corr.ct <- ur.df(pib_corr,lags=3,type='trend')
summary(pib_corr.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.516e+11 -2.101e+11  6.205e+10  2.318e+11  4.160e+11
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.996e+11  6.242e+11   1.441    0.169
## z.lag.1      4.345e-02  2.574e-01   0.169    0.868
## tt           3.994e+09  1.925e+11   0.021    0.984
## z.diff.lag1 -1.065e-01  2.791e-01  -0.381    0.708
## z.diff.lag2 -1.885e-01  2.615e-01  -0.721    0.481
## z.diff.lag3 -4.611e-01  2.384e-01  -1.934    0.071 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.648e+11 on 16 degrees of freedom
## Multiple R-squared:  0.3232, Adjusted R-squared:  0.1117
## F-statistic: 1.528 on 5 and 16 DF,  p-value: 0.2364
##
##
## Value of test-statistic is: 0.1688 5.7828 2.9826
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```
plot(pib_corr.ct)
```



Ahora bien la hipótesis ϕ_3 es probada bajo una usual prueba F, es decir, $\phi_3 = (a_0, \gamma, a_2) = (a_0, 0, 0)$. Esto es, se han colocado restricciones igual a cero a la tendencia temporal y el lag del la variable.

El valor del estadístico es

```
pib_corr.ct@teststat
```

```
##                tau3    phi2     phi3
## statistic 0.1688183 5.782813 2.982571
```

Debemos recordar que se deben consular los valores críticos propuestas por Dickey and Fuller. Los valores críticos para una muestra de tamaño 100 y niveles de significancia del 10%,5% y 1% se muestran a continuación

```
pib_corr.ct@cval
```

```
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

Por lo tanto, la hipótesis nula no puede ser rechazada, lo cual implica que la serie contiene una raíz unitaria. Esto puede ser reiterado con el estadístico τ_3 y para la variable *z.lag.1*. Los valores críticos relevantes que debemos utilizar ahora son los de Fuller[1976], los cuales se muestran para una muestra de tamaño 100.

Luego entonces, la presencia de una raíz unitaria no puede rechazada. El siguiente paso, es probar si la serie es una caminata aleatoria con o sin drift (constante).

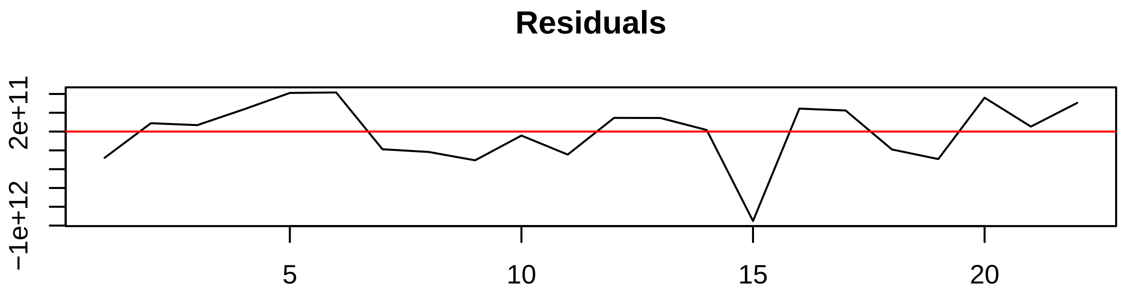
El estadístico relevante es ϕ_2 ($\gamma = a_0 = a_2 = 0$) el cual tiene un valor de 3.738072, con valores críticos de 7.02 5.13 4.31 para niveles de significancia de 1%, 5% y 10% respectivamente. La conclusión es entonces que la serie no se comporta como una caminata aleatoria pura, ya que tiene una constante incluida en el proceso, así como un orden de integración.

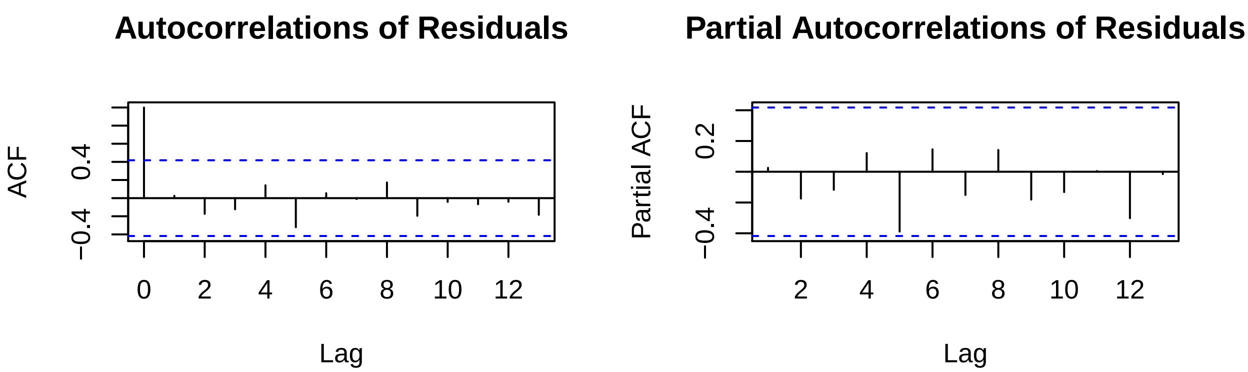
Uno procede entonces a estimar la ecuación $\nabla y_t = a_0 + \gamma y_{t-1} + \epsilon_t$ basado en los resultados obtenidos por la prueba ϕ_3. Los resultados se muestran a continuación:

```
pib_corr.co <- ur.df(pib_corr,lags=3,type='drift')
summary(pib_corr.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.511e+11 -2.104e+11  6.060e+10  2.324e+11  4.157e+11
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.113e+11  2.589e+11   3.519  0.00263 **
## z.lag.1      4.877e-02  1.937e-02   2.517  0.02214 *
## z.diff.lag1 -1.095e-01  2.292e-01  -0.478  0.63879
## z.diff.lag2 -1.911e-01  2.243e-01  -0.852  0.40615
## z.diff.lag3 -4.623e-01  2.246e-01  -2.058  0.05527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.539e+11 on 17 degrees of freedom
## Multiple R-squared:  0.3232, Adjusted R-squared:  0.164
## F-statistic:  2.03 on 4 and 17 DF,  p-value: 0.1357
##
##
## Value of test-statistic is: 2.5174 9.2159
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

```
plot(pib_corr.co)
```





Con el fin de completar la prueba, ahora se valida si en este modelo un término constante hace falta. Las pruebas se muestran a continuación:

```
pib_corr.co@teststat
```

```
##                tau2     phi1
## statistic 2.517406 9.215881
```

```
pib_corr.co@cval
```

```
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

El valor del estadístico ϕ_1 que prueba $\gamma = a_0 = 0$ resulta ser significativo comparado con los valores críticos mostrados.

Por lo tanto, se puede concluir que la serie contiene un raíz unitaria y una tendencia constante en el proceso generador de los datos.

Finalmente, se probará si diferenciando la serie una vez es suficiente para alcanzar estacionariedad. La prueba se logra utilizando como insumo para la regresión a la serie diferenciada una vez.

```
pib_corr2 <- diff(pib_corr)
pib_corr2.ct <- ur.df(pib_corr2,type="none",lags=1)
summary(pib_corr2.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.034e+12 -8.491e+10  7.674e+10  3.365e+11  9.722e+11
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.01871    0.12262  -0.153   0.8802
## z.diff.lag -0.40253    0.21583  -1.865   0.0762 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.633e+11 on 21 degrees of freedom
## Multiple R-squared:  0.1665, Adjusted R-squared:  0.08708
## F-statistic: 2.097 on 2 and 21 DF,  p-value: 0.1478
##
##
## Value of test-statistic is: -0.1526
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

```r
pib_corr2 <- diff(pib_corr,differences = 2)
pib_corr2.ct <- ur.df(pib_corr2,type="none",lags=0)
summary(pib_corr2.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.051e+12 -1.053e+11  6.029e+10  3.251e+11  9.639e+11
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1  -1.4139     0.1981  -7.137 3.72e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.529e+11 on 22 degrees of freedom
## Multiple R-squared:  0.6984, Adjusted R-squared:  0.6847
## F-statistic: 50.94 on 1 and 22 DF,  p-value: 3.72e-07
```

```
##
##
## Value of test-statistic is: -7.137
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

```
pib_corr2.ct@teststat
```

```
##                tau1
## statistic -7.136979
```

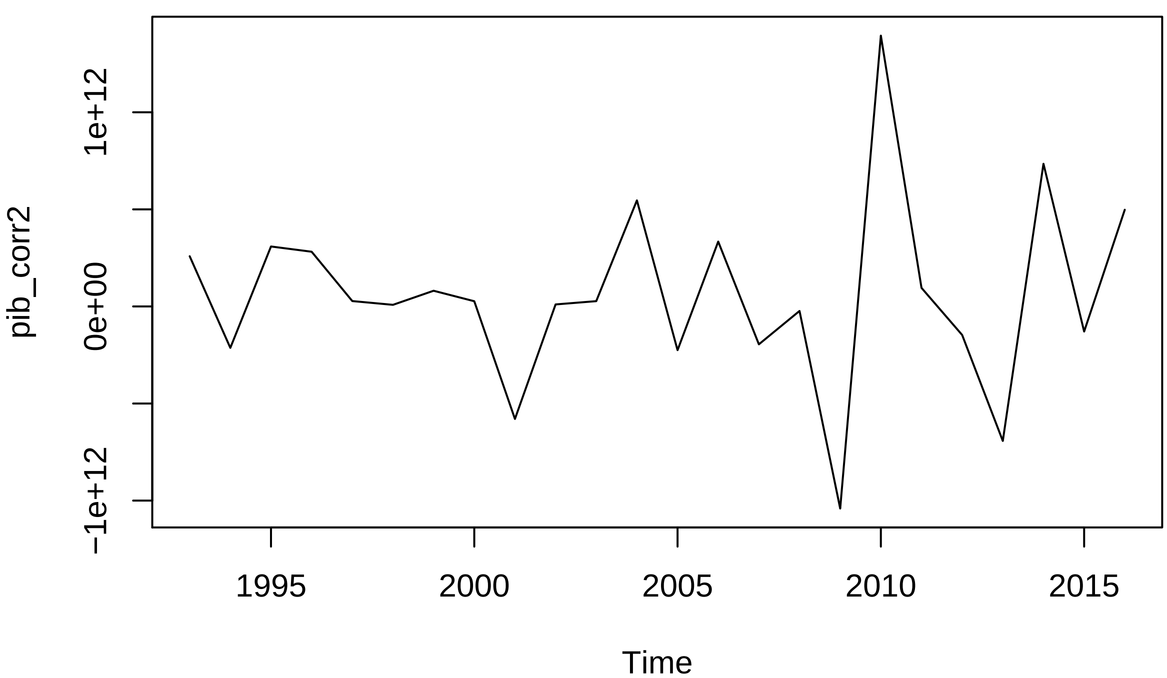
```
pib_corr2.ct@cval
```

```
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

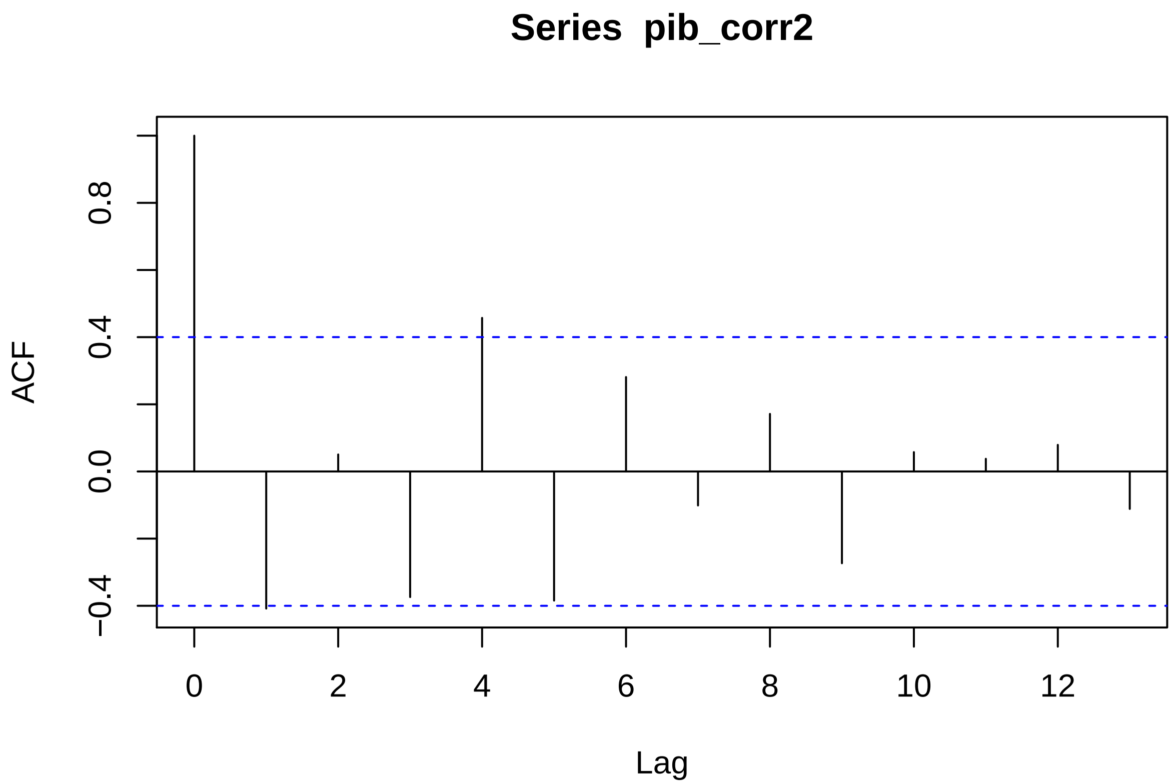
La hipótesis de que el PIB a precios corrientes es $I(2)$ se comprueba rápidamente dado el estadístico t con valor de -2.681622. ya que diferenciando solo una vez la serie no fue suficiente para hacer estacionaria la serie.

Por lo tanto. La serie del PIB a precios corrientes es $I(2)$ y de la forma $\nabla y_t = a_0 + \gamma y_{t-1} + \epsilon_t$.

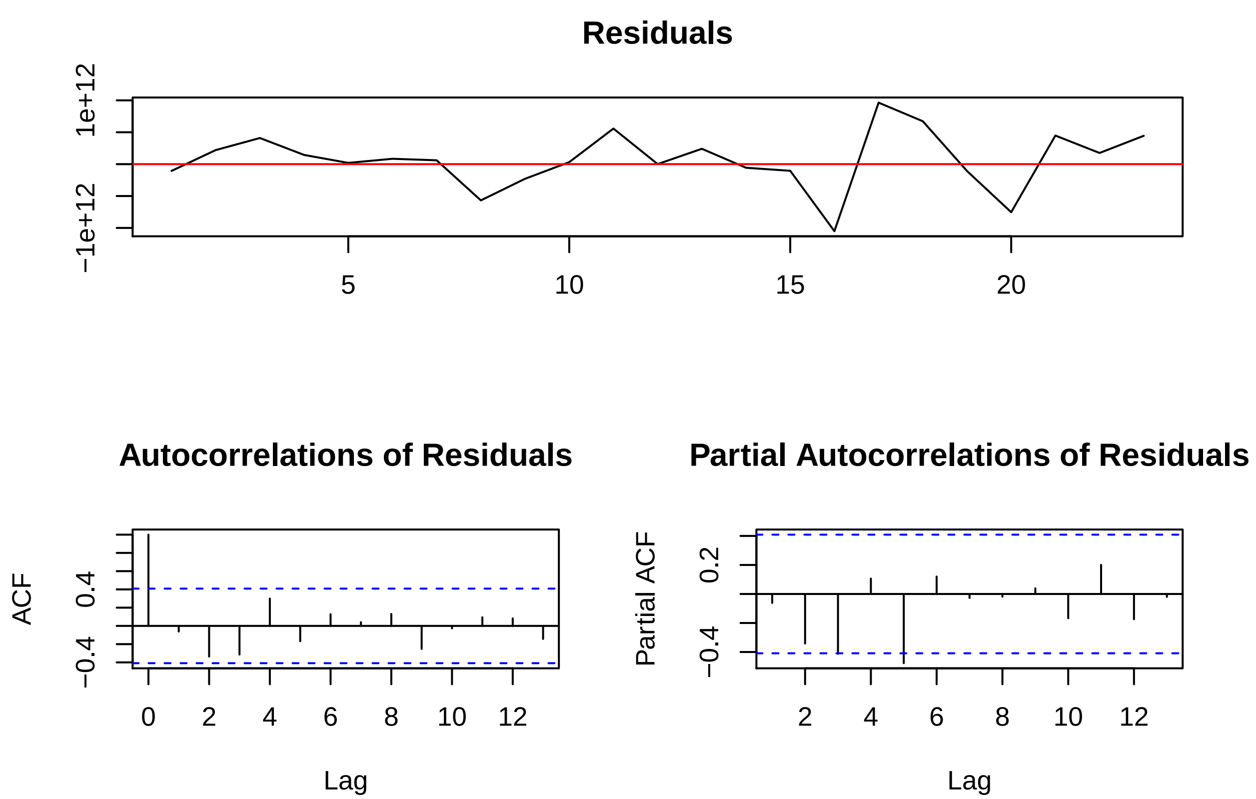
```
plot(pib_corr2)
```



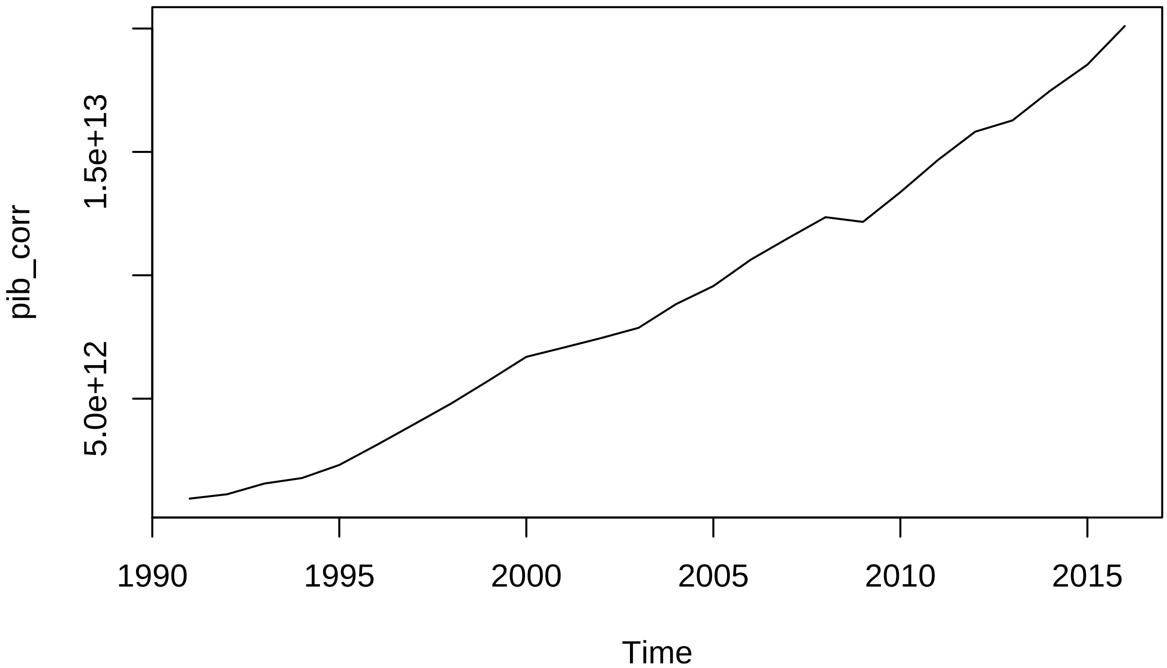

```
acf(pib_corr2)
```



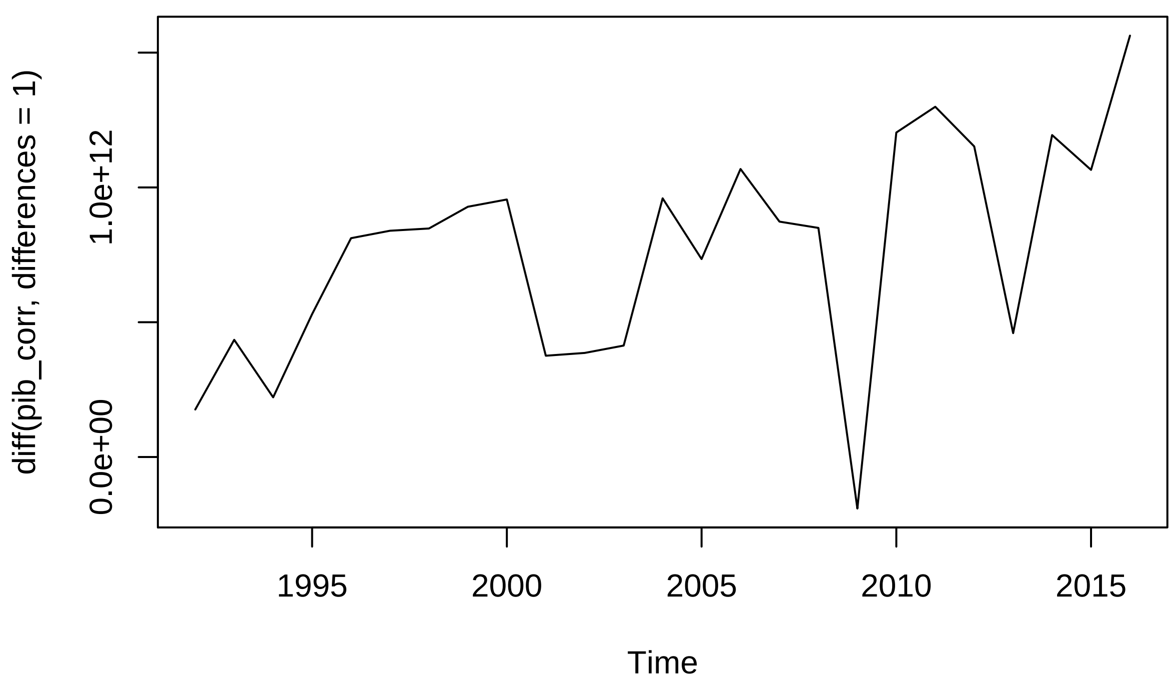

```
plot(pib_corr2.ct)
```



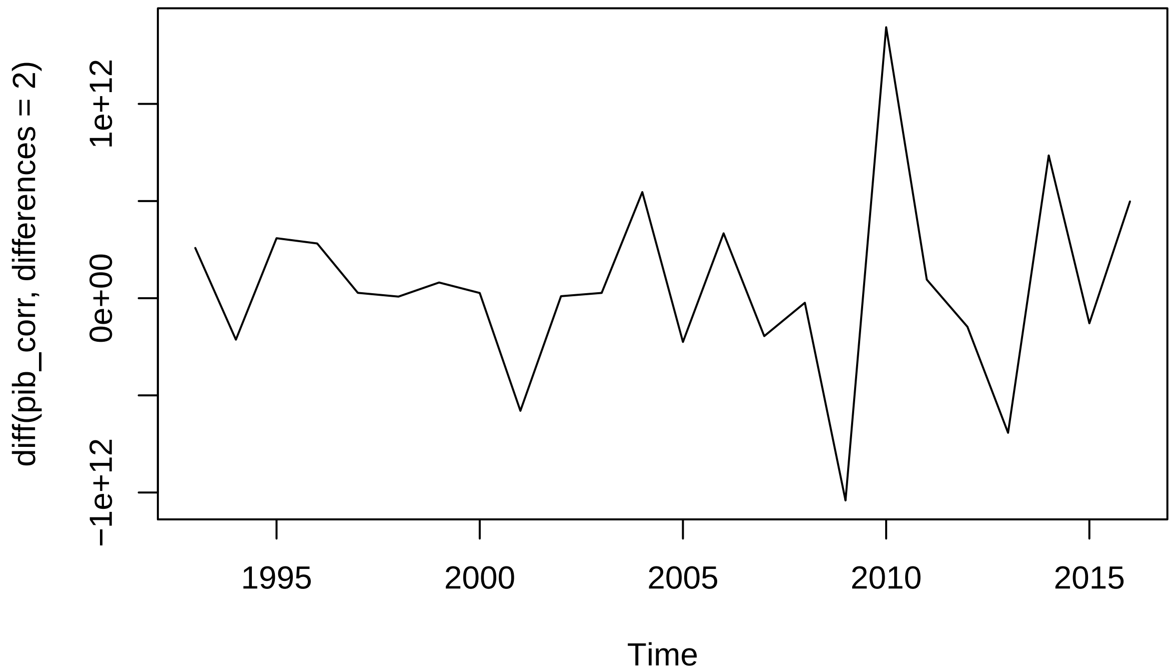
```r
plot(pib_corr)
```




```r
plot(diff(pib_corr,differences = 1))
```




```r
plot(diff(pib_corr,differences = 2))
```




```
var(pib_corr)
```

```
## [1] 3.387077e+25
```

```
var(diff(pib_corr, differences = 1))
```

```
## [1] 1.666134e+23
```

```
var(diff(pib_corr, differences = 2))
```

```
## [1] 2.345595e+23
```

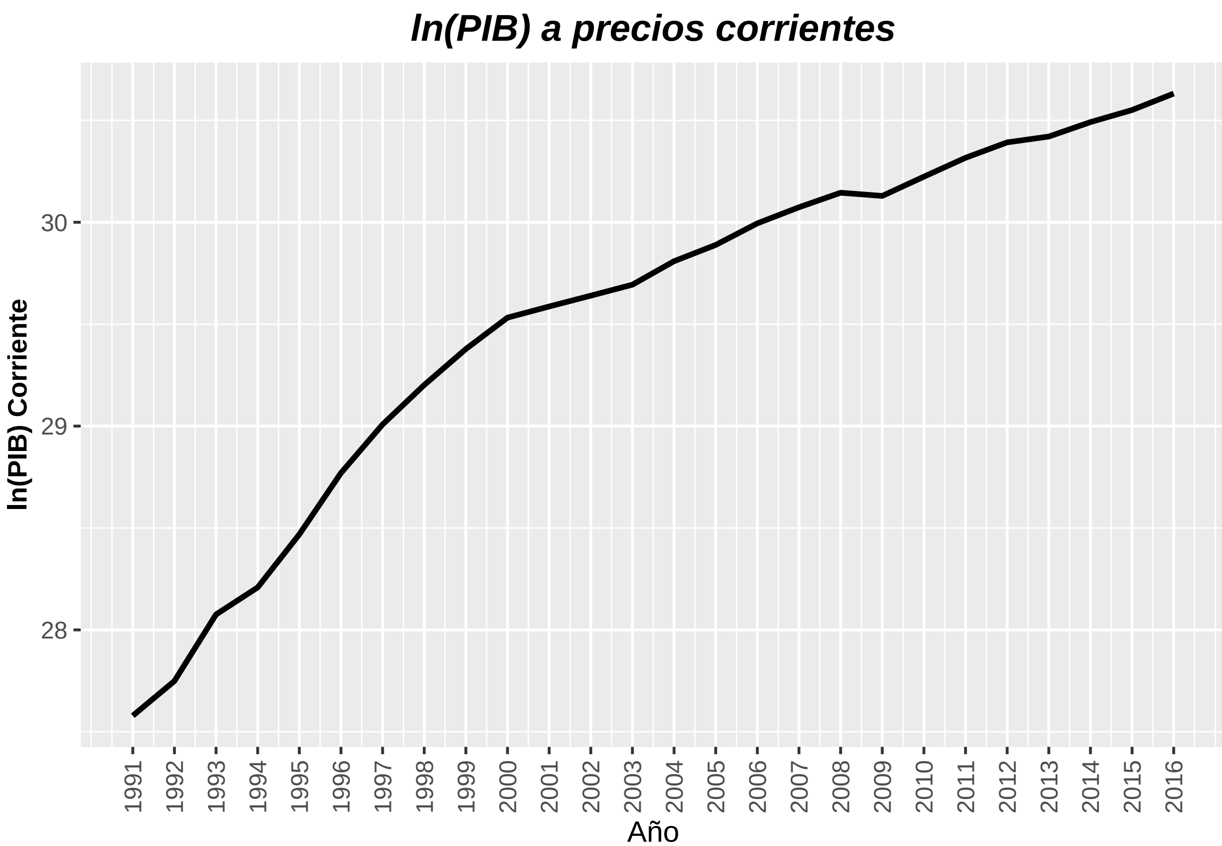
```
var(diff(pib_corr, differences = 3))
```

```
## [1] 6.799588e+23
```

Chapter 5

Orden de Integración ln(PIB) a precios corrientes

```r
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'log_GDP_corriente',
                     titulo = 'ln(PIB) a precios corrientes',
                     titulo_y = 'ln(PIB) Corriente')
gf1
```

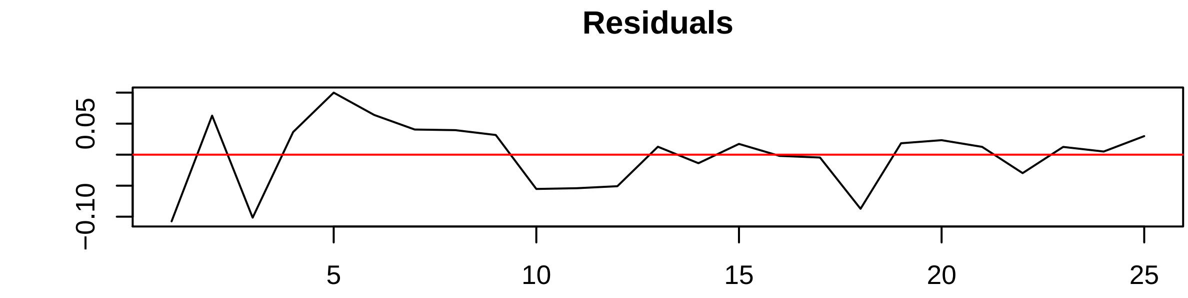


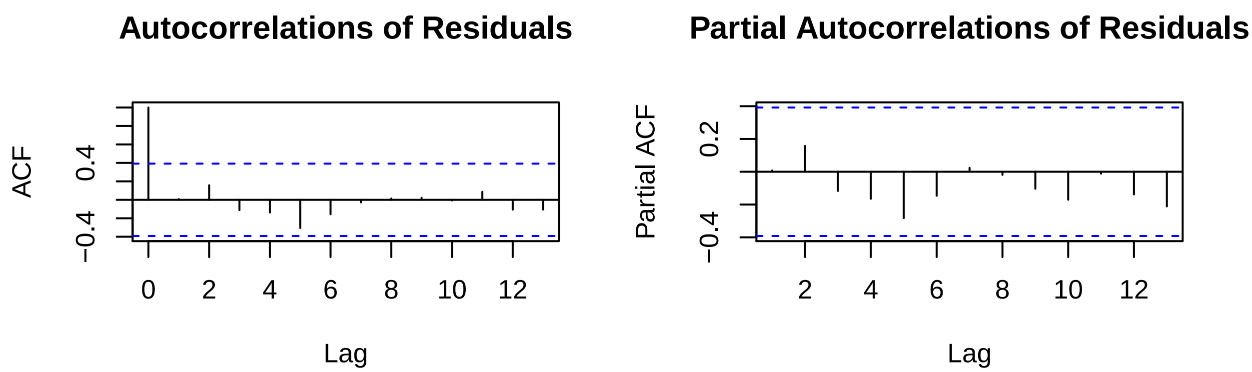

```r
log_GDP_corr <- ts(series_db$log_GDP_corriente,start = 1991, end = 2016, frequency = 1)
log_GDP_corr.ct <- ur.df(log_GDP_corr,lags=0,type='trend')
```

```r
summary(log_GDP_corr.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.10752 -0.02976  0.01256  0.03166  0.09997
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.000017   1.190524   2.520   0.0195 *
## z.lag.1     -0.098808   0.042526  -2.323   0.0298 *
## tt           0.002790   0.005143   0.542   0.5930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05449 on 22 degrees of freedom
## Multiple R-squared:  0.6335, Adjusted R-squared:  0.6002
## F-statistic: 19.01 on 2 and 22 DF,  p-value: 1.603e-05
##
##
## Value of test-statistic is: -2.3235 54.527 19.0133
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```r
plot(log_GDP_corr.ct)
```





El valor del estadístico-t para la hipótesis nula de $\gamma = 0$ es de -2.3235. El valor crítico de τ a univel de significancia del 5% reportado en las tablas de Dickey-Fuller es de -3.50, por lo que no es posible rechazar la hipótesis nula de la existencia de una raíz unitaria dada la presencia del término constante (drift) y la tendencia temporal (trend).

Recordemos que el poder de la prueba puede verse reducido debido a la presencia de términos drift/trend innecesarios, por lo que probaremos si la presencia del término temporal es necesaria dada una raíz unitaria. Para ello utilizaremos el estadístico ϕ_3 que prueba la hipótesis conjunta $a_2 = \gamma = 0$. Derivado de los resultados mostrados en las tablas anteriores, se puede rechazar la hipótesis nula, por lo que se tiene una raíz unitaria y también un término temporal.

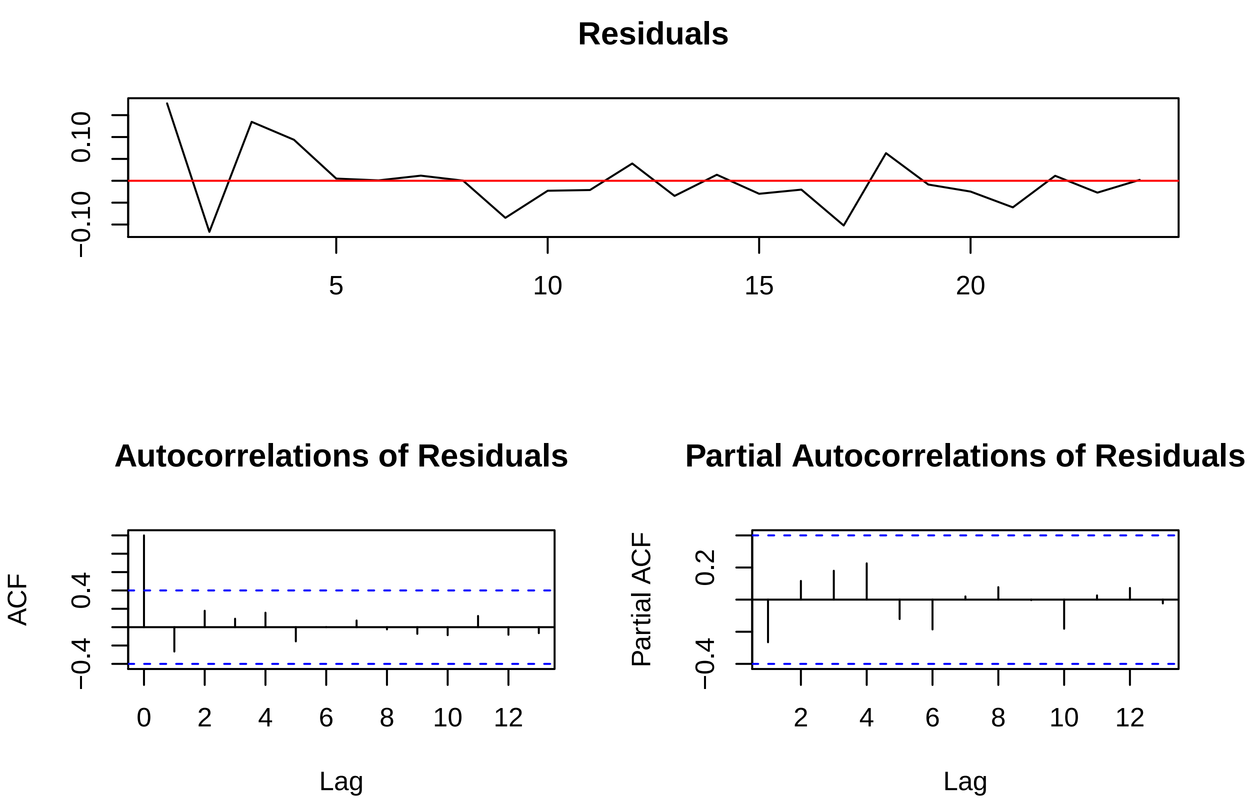
Probaremos si la serie al ser diferenciada una vez más, alcanza la estacionaredad:

```r
d_log_GDP_corr <- diff(log_GDP_corr)
d_log_GDP_corr.ct <- ur.df(d_log_GDP_corr,type="drift",lags=0)
summary(d_log_GDP_corr.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.116748 -0.027785 -0.004129  0.012247  0.176952
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04117    0.02466   1.669   0.1093
## z.lag.1     -0.36251    0.16375  -2.214   0.0375 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06879 on 22 degrees of freedom
## Multiple R-squared:  0.1822, Adjusted R-squared:  0.145
## F-statistic: 4.901 on 1 and 22 DF,  p-value: 0.03752
##
##
## Value of test-statistic is: -2.2138 2.4855
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.75 -3.00 -2.63
## phi1  7.88  5.18  4.12
```

```
plot(d_log_GDP_corr.ct)
```



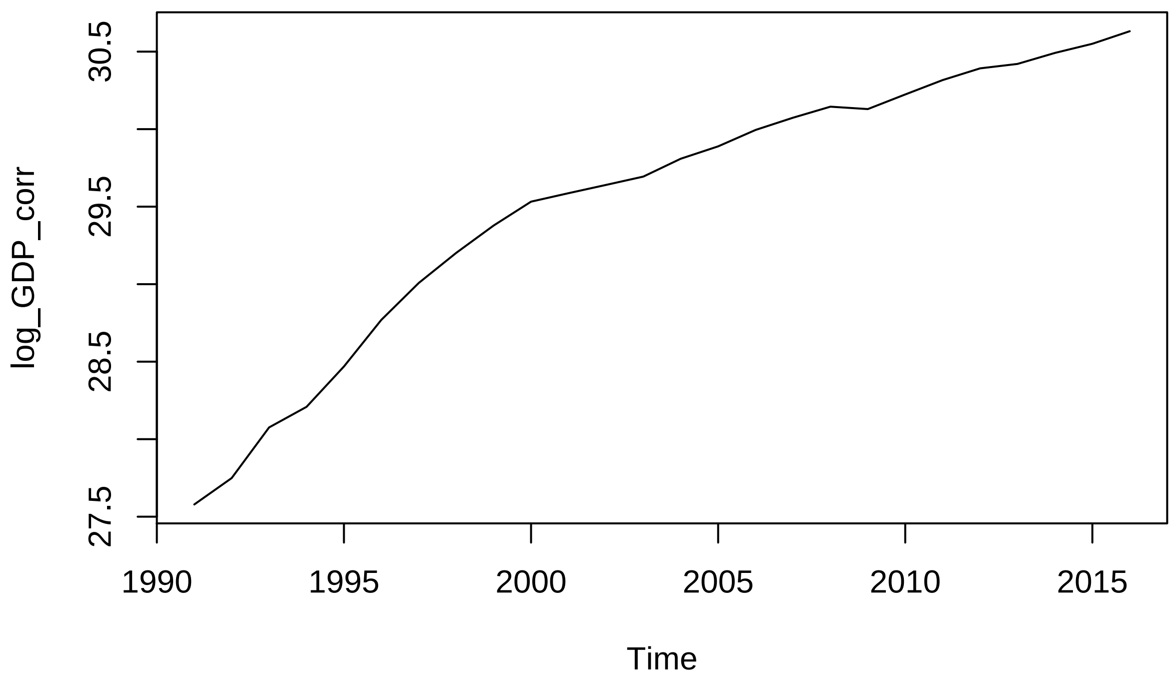

```
d_log_GDP_corr <- diff(log_GDP_corr,differences = 2)
d_log_GDP_corr.ct <- ur.df(d_log_GDP_corr,type="none",lags=0)
summary(d_log_GDP_corr.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
```

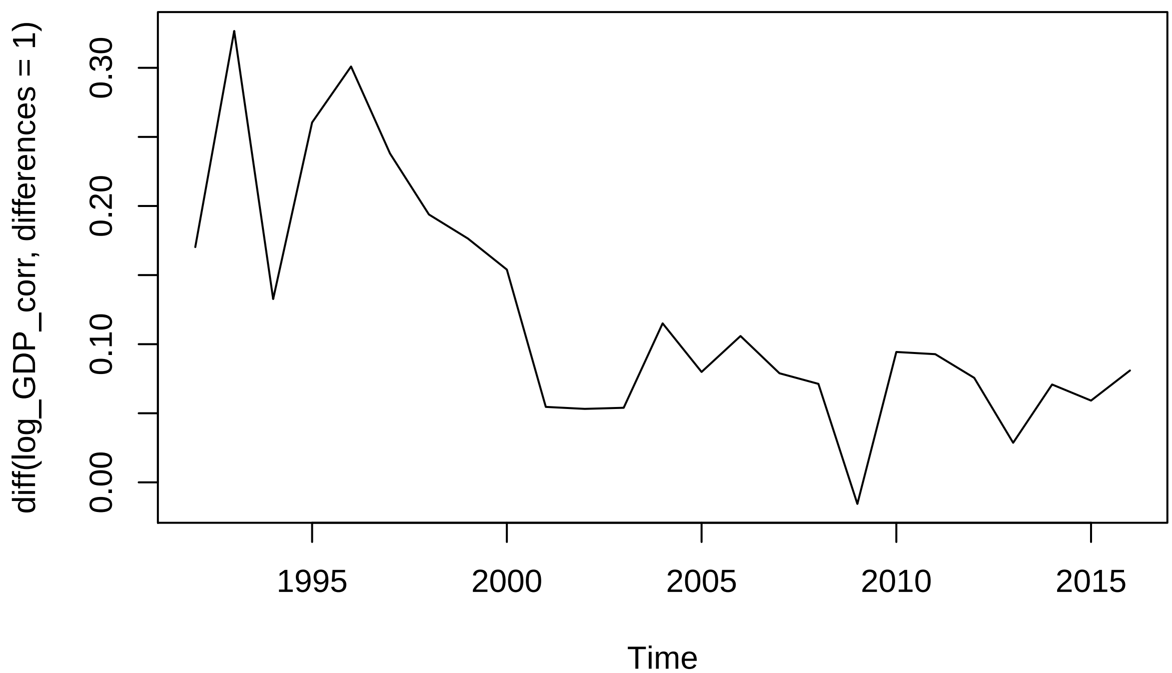
```
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11940 -0.04622 -0.01456  0.01799  0.10147
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1  -1.4770     0.1629  -9.067 6.95e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05809 on 22 degrees of freedom
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.7793
## F-statistic: 82.21 on 1 and 22 DF,  p-value: 6.954e-09
##
##
## Value of test-statistic is: -9.0669
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

Por lo que la serie al ser diferenciada dos veces alcanza estacionariedad. Por lo tanto, la serie del logaritmo natural del PIB a precios corrientes es $I(2)$ y es de la forma $\nabla y_t = a_0 + \gamma y_{t-1} + a_2 t + \epsilon_t$.

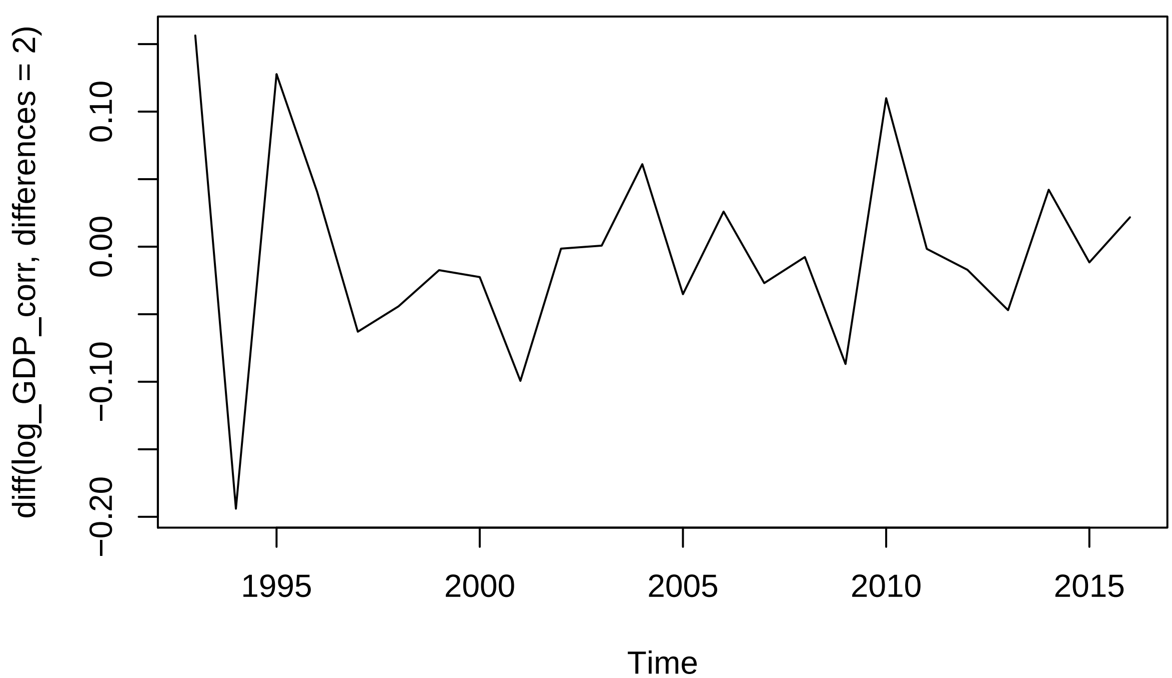
```
plot(log_GDP_corr)
```



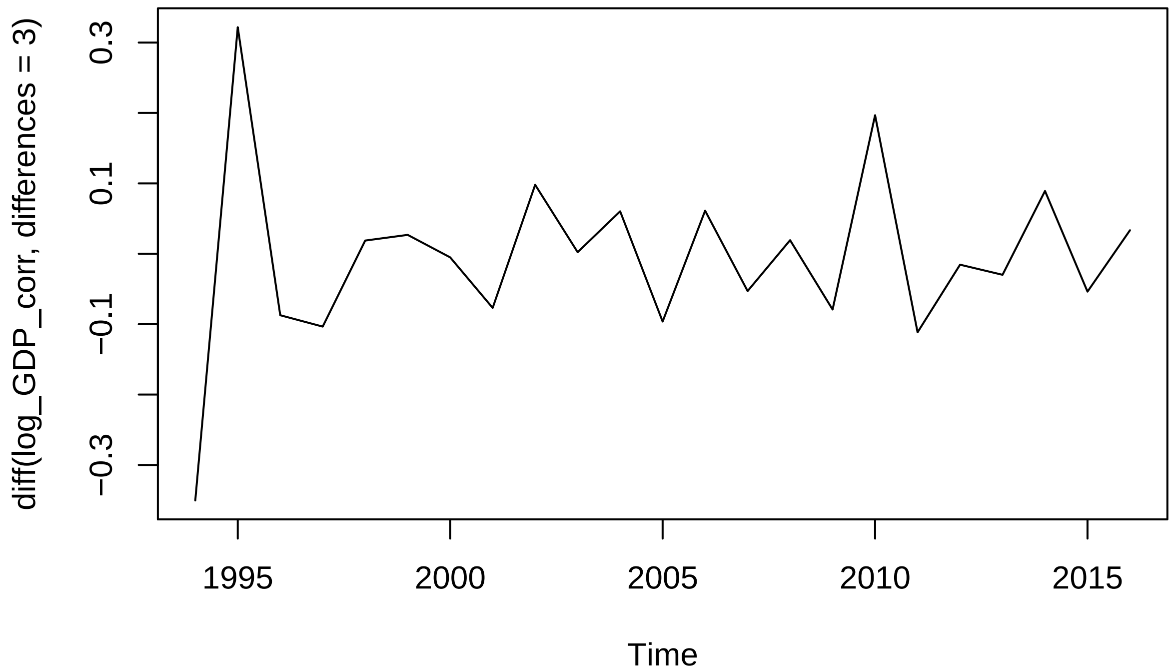
```
plot(diff(log_GDP_corr, differences = 1))
```




```
plot(diff(log_GDP_corr, differences = 2))
```




```
plot(diff(log_GDP_corr, differences = 3))
```




```
mean(log_GDP_corr)
```

```
## [1] 29.53691
```

```
mean(diff(log_GDP_corr,differences = 1))
```

```
## [1] 0.1221156
```

```
mean(diff(log_GDP_corr,differences = 2))
```

```
## [1] -0.003721427
```

```
mean(diff(log_GDP_corr,differences = 3))
```

```
## [1] -0.00585276
```

```
var(log_GDP_corr)
```

```
## [1] 0.8104397
```

```
var(diff(log_GDP_corr, differences = 1))
```

```
## [1] 0.007426465
```

```
var(diff(log_GDP_corr, differences = 2))
```

```
## [1] 0.00553448
```

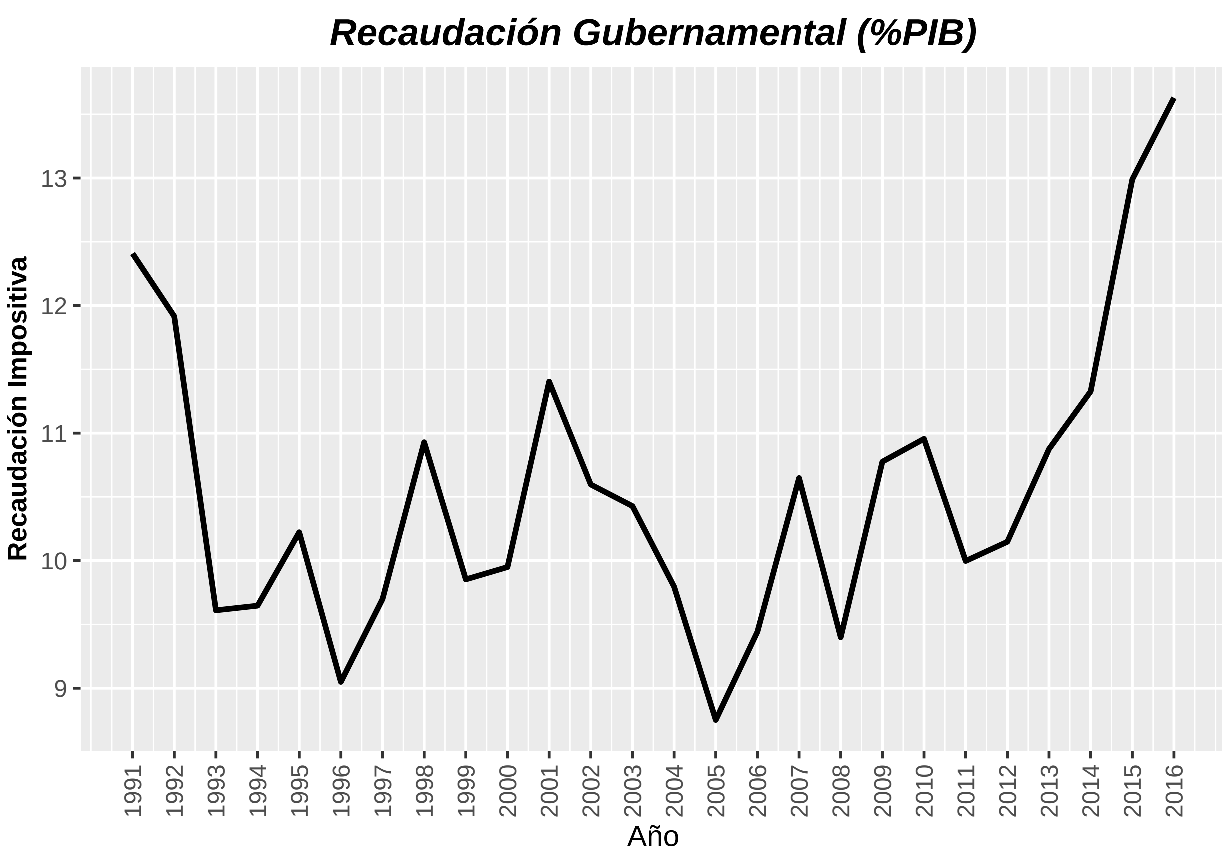
```
var(diff(log_GDP_corr, differences = 3))
```

```
## [1] 0.01594711
```

Chapter 6

Orden Integración de la serie Recaudación Impositiva

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Recaudacion_Impositiva_PorcGDP',
                     titulo = 'Recaudación Gubernamental (%PIB)',
                     titulo_y = 'Recaudación Impositiva')
gf1
```

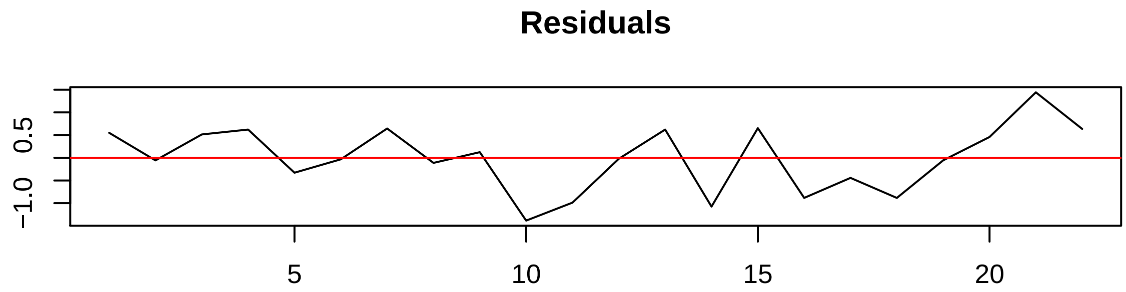


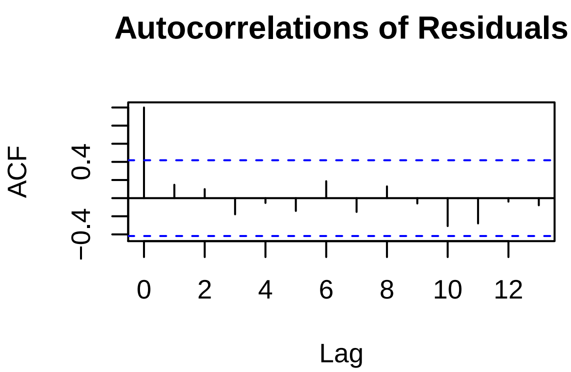

```
rec_imp <- ts(series_db$Recaudacion_Impositiva_PorcGDP,start = 1991, end = 2016, frequency = 1)
rec_imp.ct <- ur.df(rec_imp,lags=3,type='trend')
```

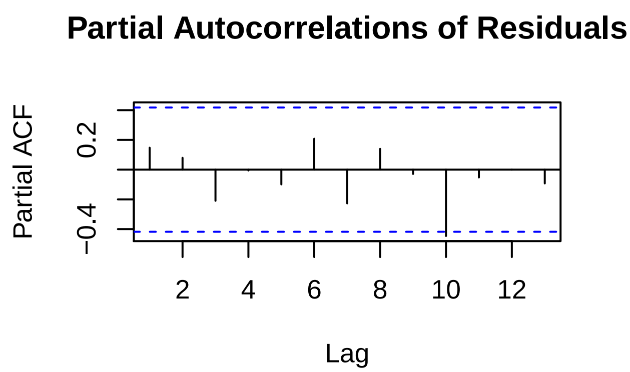
```r
summary(rec_imp.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38382 -0.41429 -0.02738  0.60396  1.44221
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.21915    4.15990   1.014   0.3256
## z.lag.1     -0.45355    0.41719  -1.087   0.2931
## tt           0.04417    0.03320   1.330   0.2021
## z.diff.lag1  0.17628    0.41533   0.424   0.6769
## z.diff.lag2 -0.09357    0.27355  -0.342   0.7368
## z.diff.lag3  0.44170    0.22932   1.926   0.0721 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8226 on 16 degrees of freedom
## Multiple R-squared:  0.4191, Adjusted R-squared:  0.2376
## F-statistic: 2.309 on 5 and 16 DF,  p-value: 0.09268
##
##
## Value of test-statistic is: -1.0871 1.3035 1.1716
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```r
plot(rec_imp.ct)
```







El valor del estadístico-t para la hipótesis nula de $\gamma = 0$ y el valor crítico de τ a univel de significancia del 5% reportado en las tablas de Dickey-Fuller demuestran que no es posible rechazar la hipótesis nula de la existencia de una raíz unitaria dada la presencia del término constante (drift) y la tendencia temporal (trend).

Recordemos que el poder de la prueba puede verse reducido debido a la presencia de términos drift/trend innecesarios, por lo que probaremos si la presencia del término temporal es necesaria dada una raíz unitaria. Para ello utilizaremos el estadístico ϕ_3 que prueba la hipótesis conjunta $a_2 = \gamma = 0$. Derivado de los resultados mostrados en las tablas anteriores, se puede aceptar la hipótesis nula, por lo que se tiene una raíz unitaria sin la presencia de un término temporal.

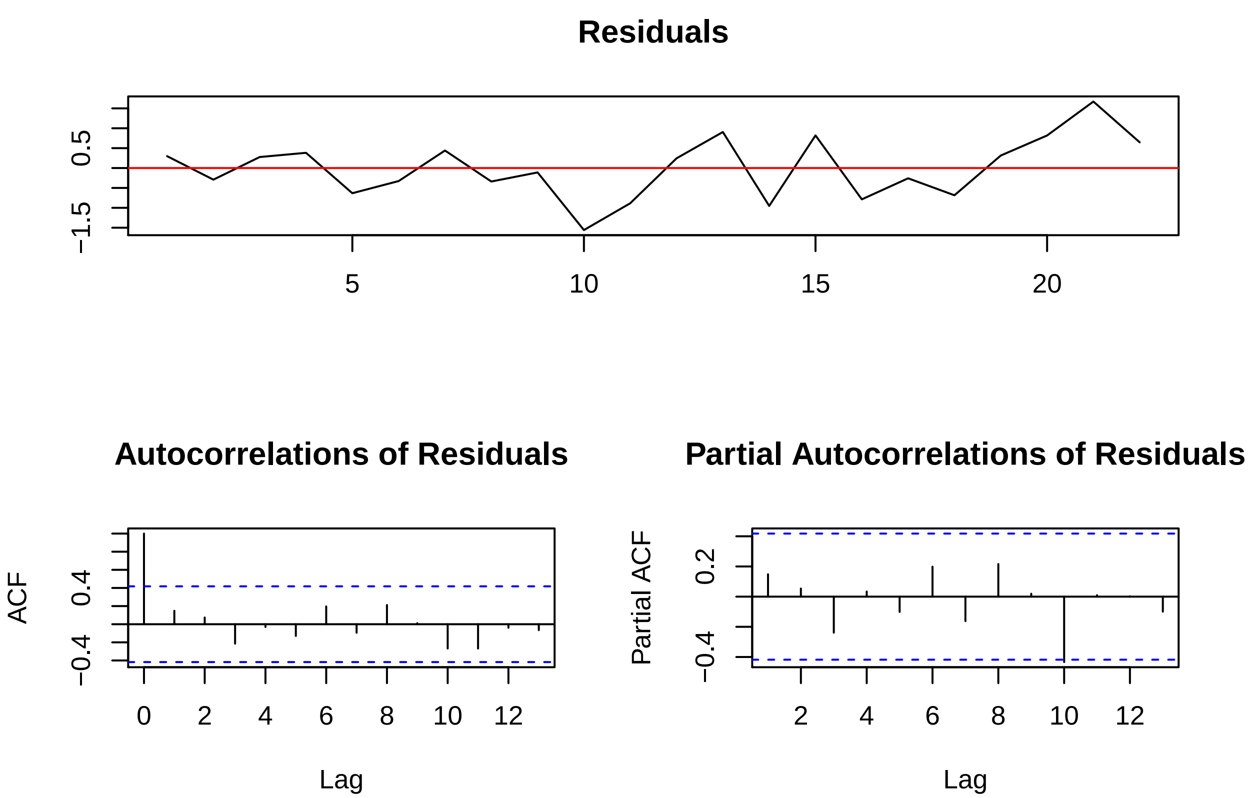
Probaremos si la serie requiere del término constante:

```
rec_imp.co <- ur.df(rec_imp,type='drift',lags=3)
summary(rec_imp.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55849 -0.55907  0.06725  0.42599  1.67112
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.32944    4.19768   0.793   0.4386
## z.lag.1     -0.30434    0.41083  -0.741   0.4689
## z.diff.lag1  0.16072    0.42446   0.379   0.7096
## z.diff.lag2 -0.05717    0.27827  -0.205   0.8397
## z.diff.lag3  0.51692    0.22722   2.275   0.0361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8411 on 17 degrees of freedom
## Multiple R-squared:  0.3549, Adjusted R-squared:  0.2031
## F-statistic: 2.338 on 4 and 17 DF,  p-value: 0.09679
##
##
## Value of test-statistic is: -0.7408 1.0241
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

```
plot(rec_imp.co)
```



Derivado de los resultados anteriores se puede observar que la serie tampoco requiere la presencia de un término constante, por lo que haremos la prueba sin este término

```
rec_imp.n <- ur.df(rec_imp,type='none',lags=3)
summary(rec_imp.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6705 -0.4756 -0.0684  0.4774  1.5735
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1      0.02120    0.01761   1.204   0.2443
## z.diff.lag1 -0.12894    0.21409  -0.602   0.5545
## z.diff.lag2 -0.21662    0.19041  -1.138   0.2702
## z.diff.lag3  0.42687    0.19478   2.192   0.0418 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8323 on 18 degrees of freedom
## Multiple R-squared:  0.3559, Adjusted R-squared:  0.2128
## F-statistic: 2.487 on 4 and 18 DF,  p-value: 0.08019
##
##
## Value of test-statistic is: 1.2038
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

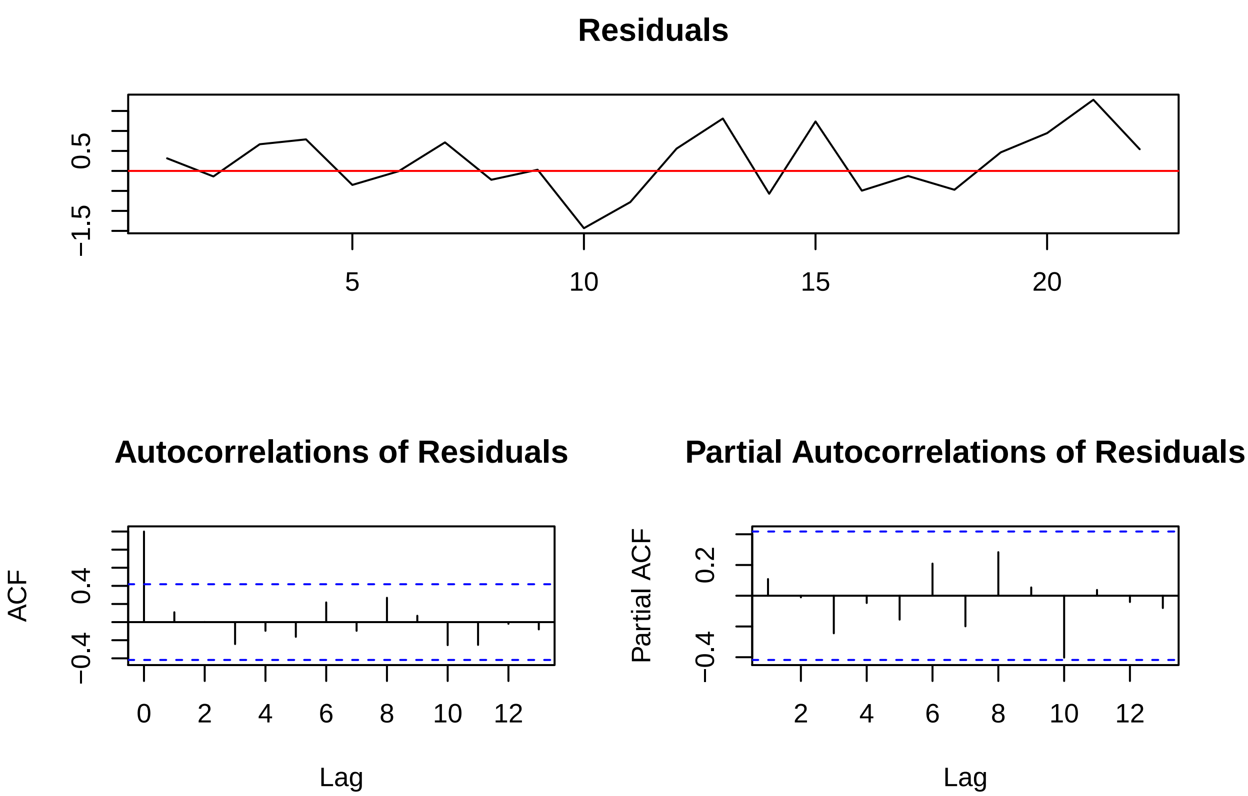
Probaremos si la serie al ser diferenciada una vez más, alcanza la estacionaredad:

```r
d.rec_imp <- diff(rec_imp,differences = 1)
d.rec_imp.n <- ur.df(d.rec_imp,type="none",lags=2)
summary(d.rec_imp.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4326 -0.3179  0.1727  0.7023  1.7802
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -0.8506     0.4389  -1.938   0.0676 .
## z.diff.lag1  -0.2222     0.3093  -0.719   0.4811
## z.diff.lag2  -0.4279     0.1971  -2.171   0.0428 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8421 on 19 degrees of freedom
## Multiple R-squared:  0.6974, Adjusted R-squared:  0.6496
## F-statistic: 14.59 on 3 and 19 DF,  p-value: 3.605e-05
##
##
## Value of test-statistic is: -1.9381
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

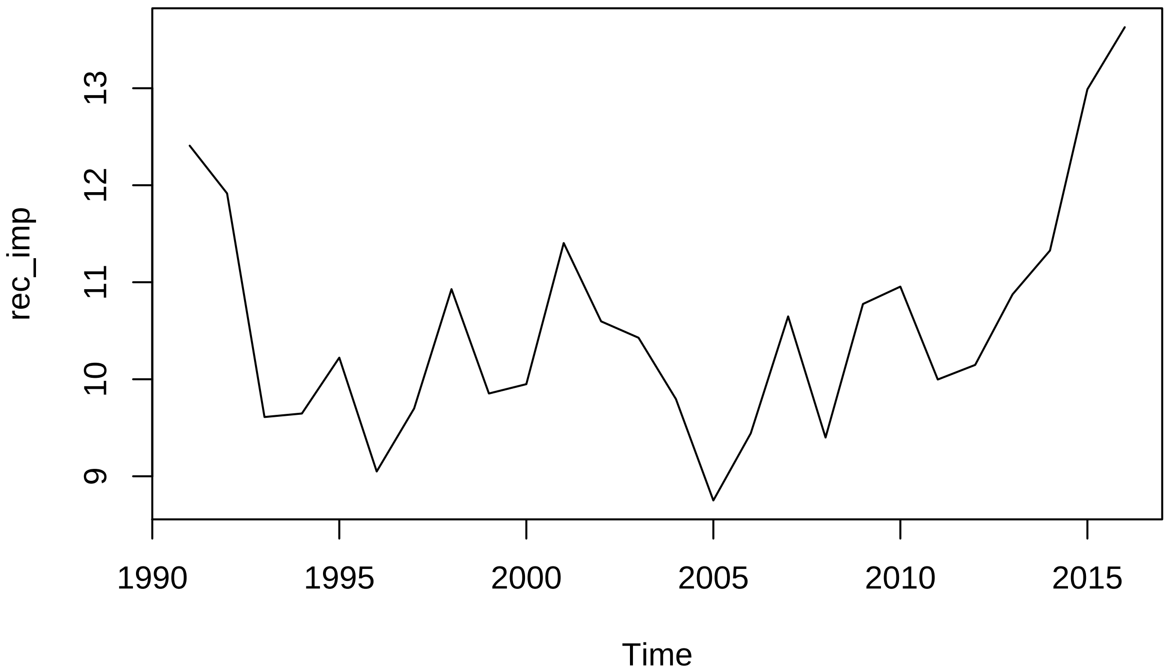
```r
plot(d.rec_imp.n)
```



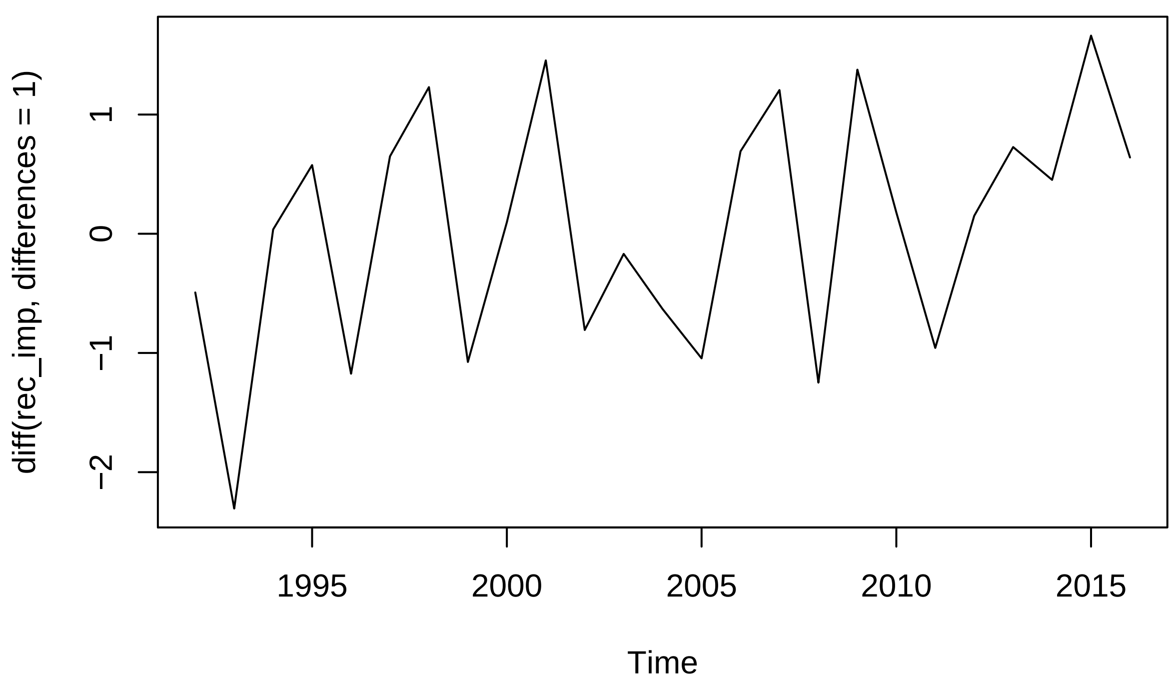
Por lo que la serie al ser diferenciada una vez, alcanza estacionariedad.

Por lo tanto. La serie de recaudación impositiva es $I(1)$ y de la forma $\nabla y_t = \gamma y_{t-1} + \epsilon_t$.

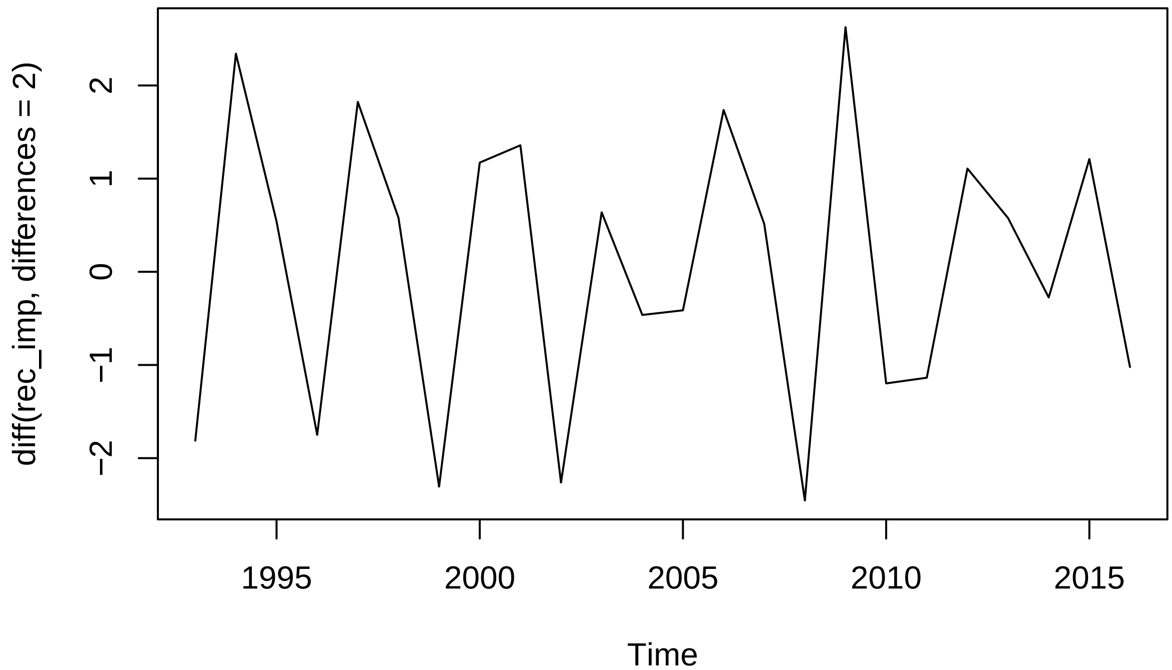
```r
plot(rec_imp)
```



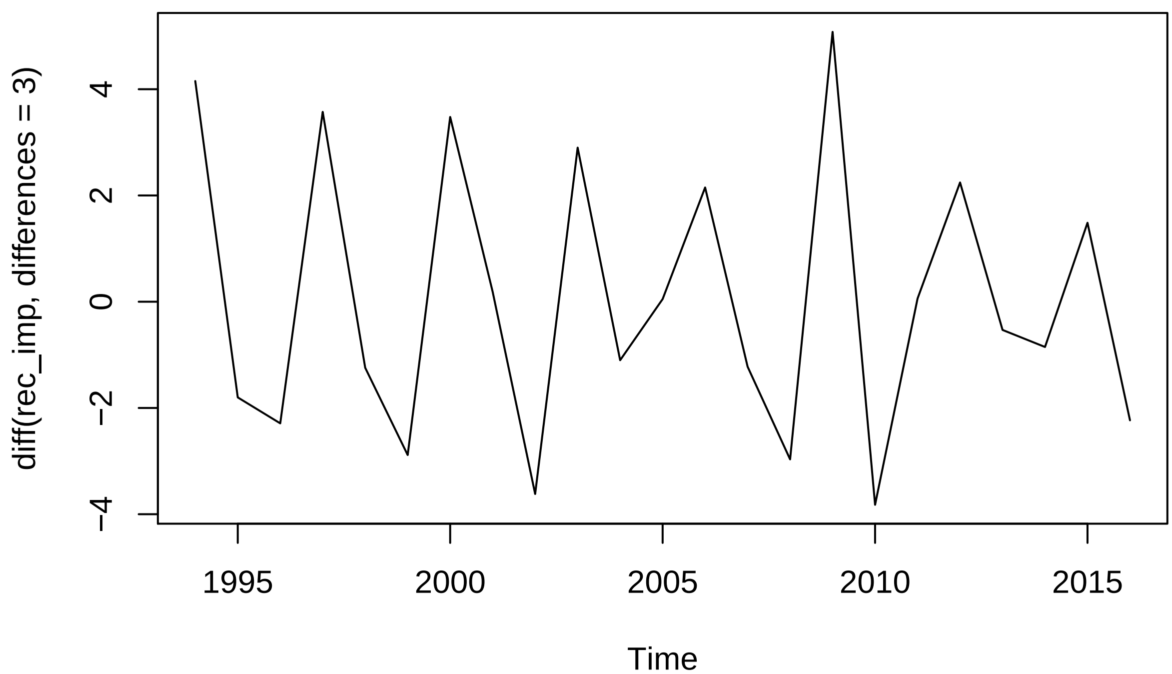

```
plot(diff(rec_imp, differences = 1))
```




```
plot(diff(rec_imp, differences = 2))
```




```
plot(diff(rec_imp, differences = 3))
```




```
var(rec_imp)
```

```
## [1] 1.400806
```

```
var(diff(rec_imp, differences = 1))
```

```
## [1] 1.035755
```

```
var(diff(rec_imp, differences = 2))
```

```
## [1] 2.314733
```

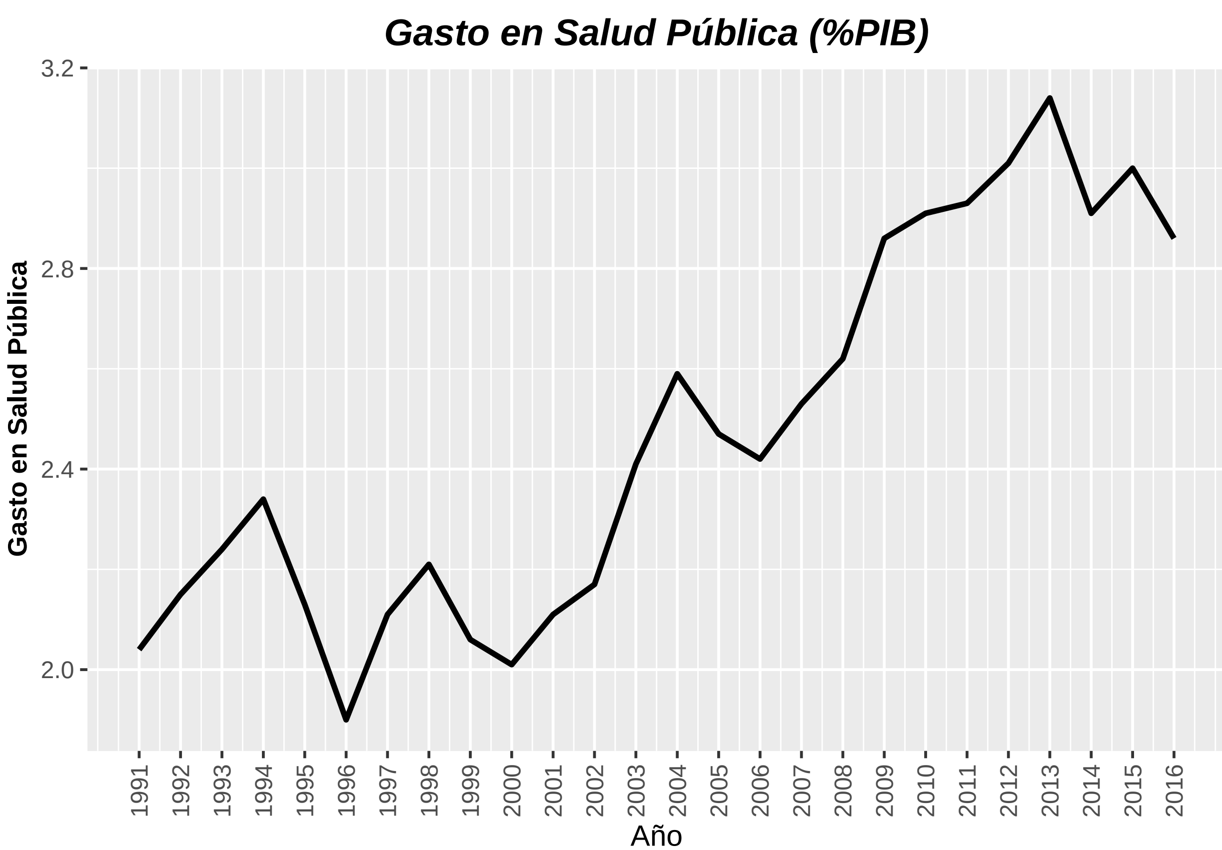
```r
var(diff(rec_imp, differences = 3))
```

```
## [1] 6.899994
```

Chapter 7

Orden Integración de la serie Gasto en Salud

```r
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Gasto_Salud_PorcGDP',
                     titulo = 'Gasto en Salud Pública (%PIB)',
                     titulo_y = 'Gasto en Salud Pública')
gf1
```

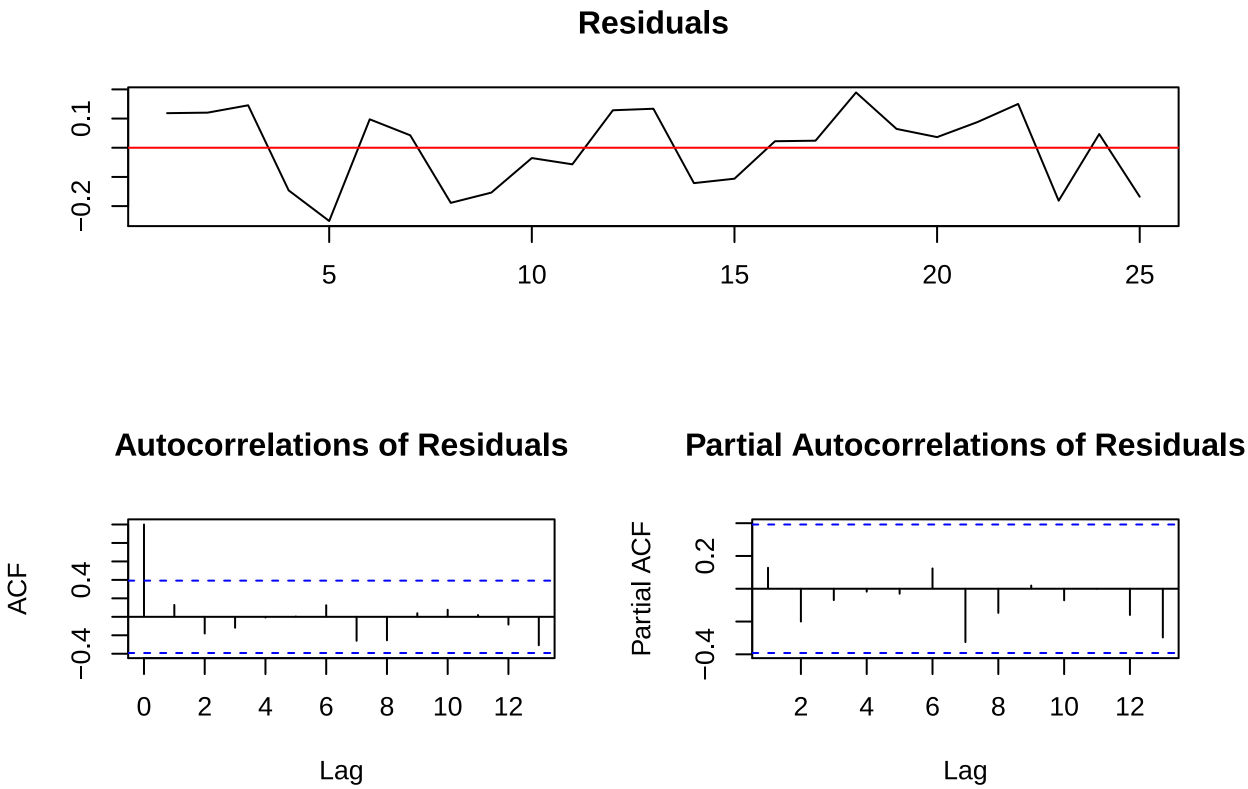



```r
salud <- ts(series_db$Gasto_Salud_PorcGDP,start = 1991, end = 2016, frequency = 1)
salud.ct <- ur.df(salud,lags=0,type='trend')
```

```r
summary(salud.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2510 -0.1212  0.0364  0.1184  0.1894
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.657304   0.304718   2.157   0.0422 *
## z.lag.1     -0.333622   0.161126  -2.071   0.0503 .
## tt           0.014857   0.008245   1.802   0.0853 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1357 on 22 degrees of freedom
## Multiple R-squared:  0.1633, Adjusted R-squared:  0.08728
## F-statistic: 2.147 on 2 and 22 DF,  p-value: 0.1406
##
##
## Value of test-statistic is: -2.0706 1.9182 2.1475
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```r
plot(salud.ct)
```



El valor del estadístico-t para la hipótesis nula de $\gamma = 0$ y el valor crítico de τ a univel de significancia del 5% reportado en las tablas de Dickey-Fuller demuestran que no es posible rechazar la hipótesis nula de la existencia de una raíz unitaria dada la presencia del término constante (drift) y la tendencia temporal (trend).

Recordemos que el poder de la prueba puede verse reducido debido a la presencia de términos drift/trend innecesarios, por lo que probaremos si la presencia del término temporal es necesaria dada una raíz unitaria. Para ello utilizaremos el estadístico ϕ_3 que prueba la hipótesis conjunta $a_2 = \gamma = 0$. Derivado de los resultados mostrados en las tablas anteriores, se puede aceptar la hipótesis nula, por lo que se tiene una raíz unitaria sin la presencia de un término temporal.

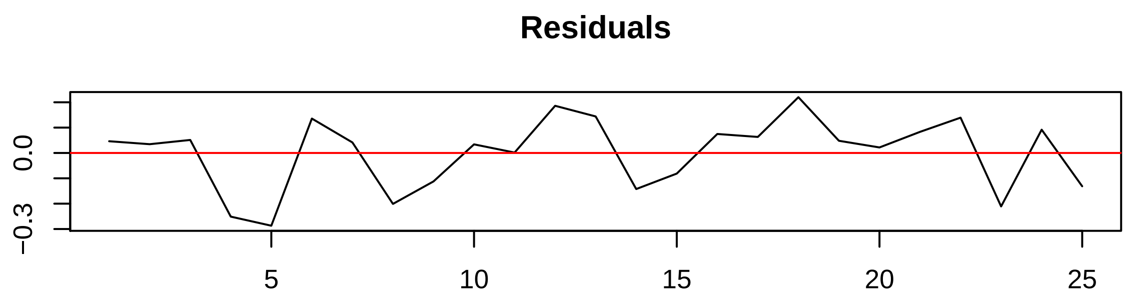
Probaremos si la serie requiere del término constante:

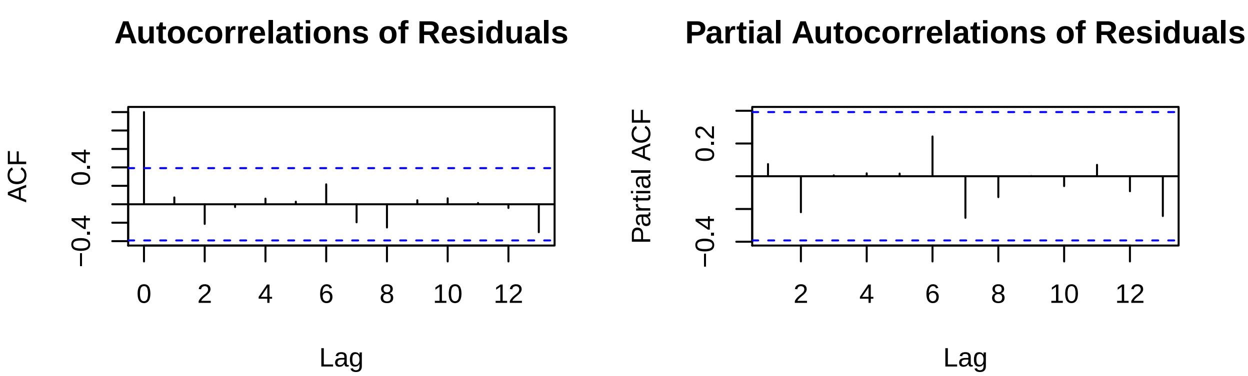
```r
salud.co <- ur.df(salud,type='drift',lags=0)
summary(salud.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28696 -0.11223  0.04153  0.08329  0.21994
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21738    0.19105   1.138    0.267
## z.lag.1     -0.07532    0.07709  -0.977    0.339
##
## Residual standard error: 0.1422 on 23 degrees of freedom
## Multiple R-squared:  0.03985,    Adjusted R-squared:  -0.001896
## F-statistic: 0.9546 on 1 and 23 DF,  p-value: 0.3387
##
##
## Value of test-statistic is: -0.977 1.1422
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

```r
plot(salud.co)
```





Derivado de los resultados anteriores se puede observar que la serie tampoco requiere la presencia de un término constante, por lo que haremos la prueba sin este término

```r
salud.n <- ur.df(salud,type='none',lags=0)
summary(salud.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
```

```
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26585 -0.07820  0.05677  0.08237  0.21522
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1  0.01142    0.01155   0.989    0.333
##
## Residual standard error: 0.1431 on 24 degrees of freedom
## Multiple R-squared:  0.03914,    Adjusted R-squared:  -0.000892
## F-statistic: 0.9777 on 1 and 24 DF,  p-value: 0.3326
##
##
## Value of test-statistic is: 0.9888
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

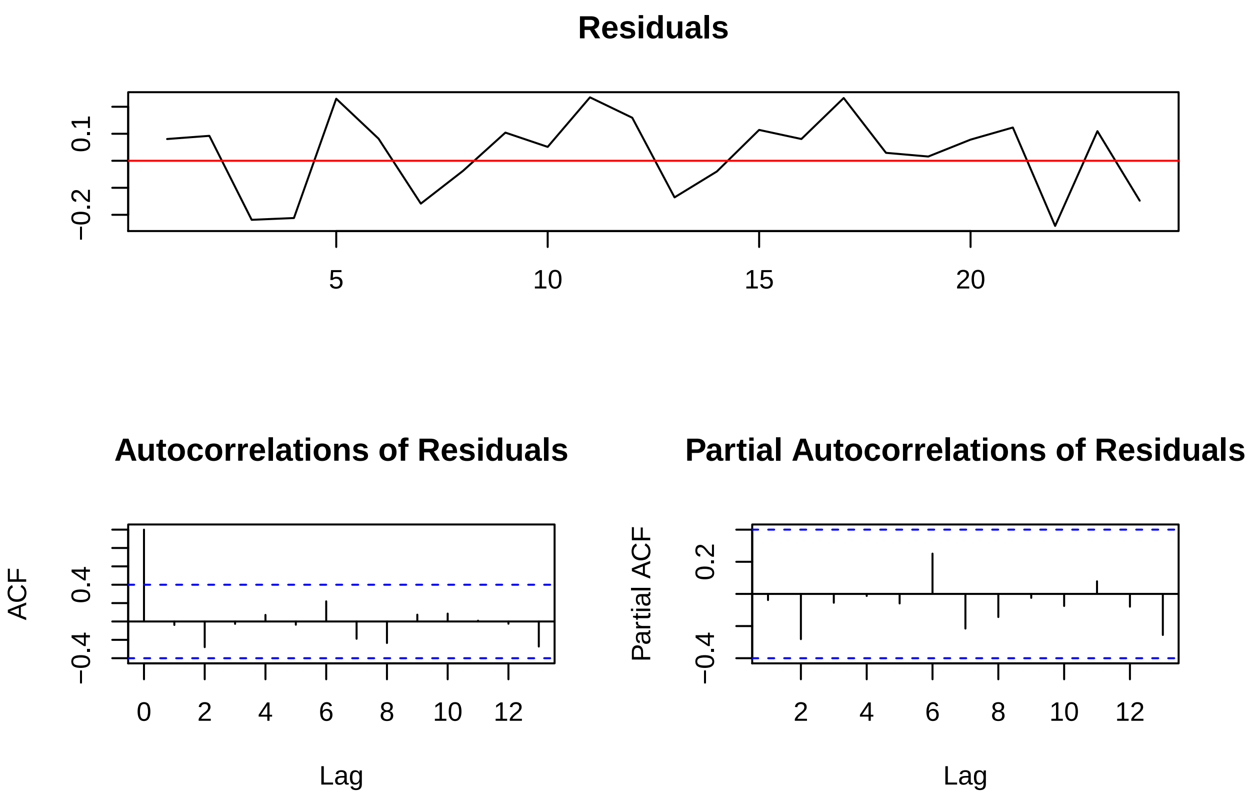
Probaremos si la serie al ser diferenciada una vez, alcanza la estacionaredad:

```r
d.salud <- diff(salud,differences = 1)
d.salud.n <- ur.df(d.salud,type="none",lags=0)
summary(d.salud.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24116 -0.06364  0.07942  0.11087  0.23485
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1  -0.9142     0.2093  -4.367 0.000225 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1468 on 23 degrees of freedom
## Multiple R-squared:  0.4533, Adjusted R-squared:  0.4295
## F-statistic: 19.07 on 1 and 23 DF,  p-value: 0.0002255
```

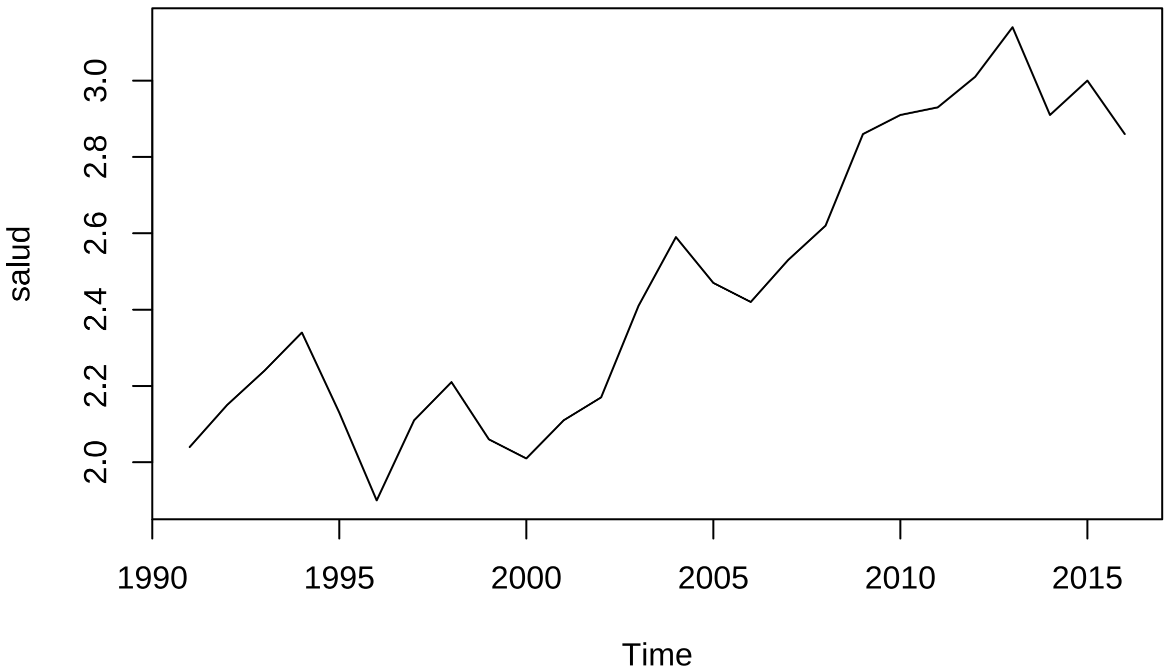
```
##
##
## Value of test-statistic is: -4.3671
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

```
plot(d.salud.n)
```

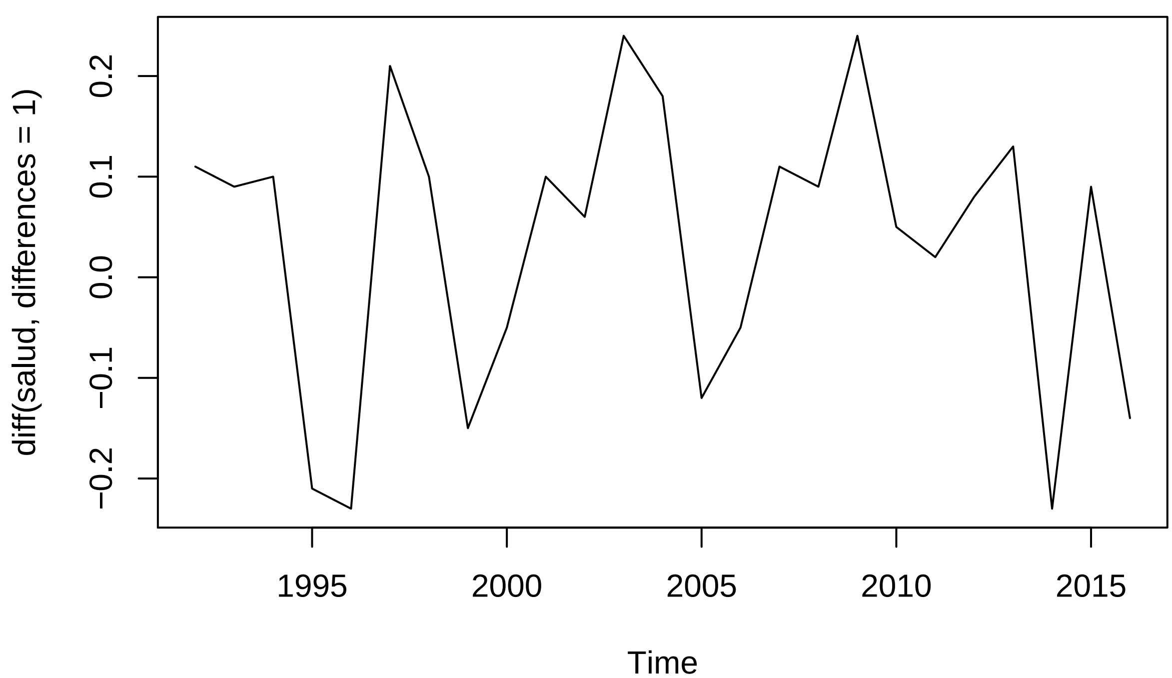


Por lo tanto. La serie de gasto en salud es $I(1)$ y de la forma $\nabla y_t = \gamma y_{t-1} + \epsilon_t$.

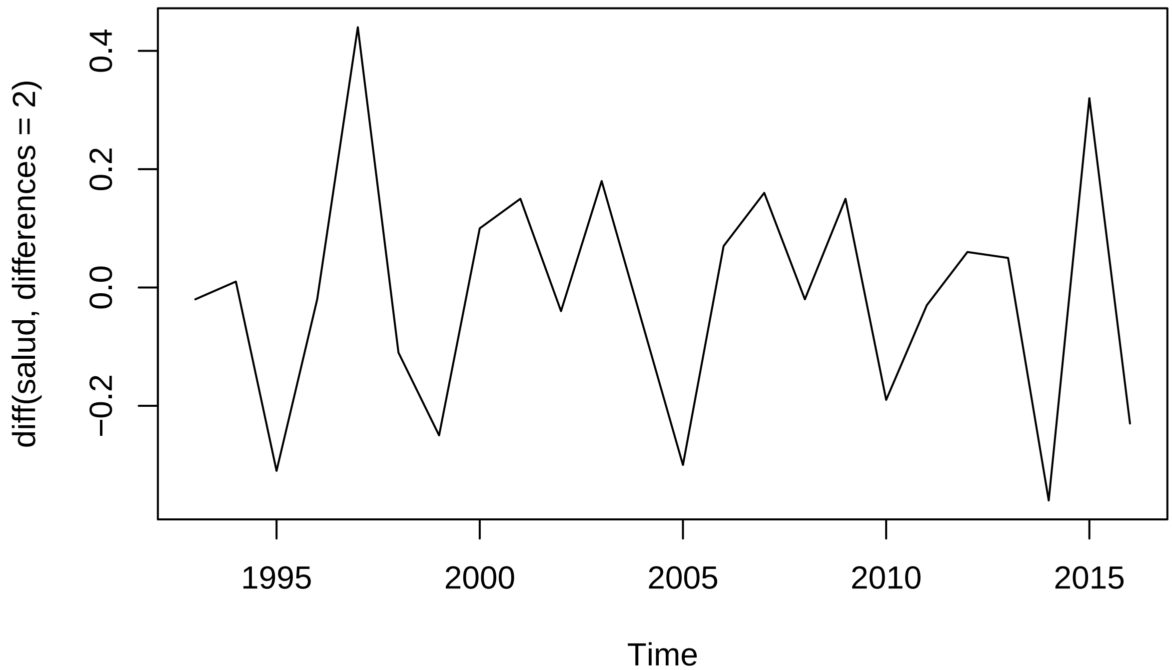
```
plot(salud)
```



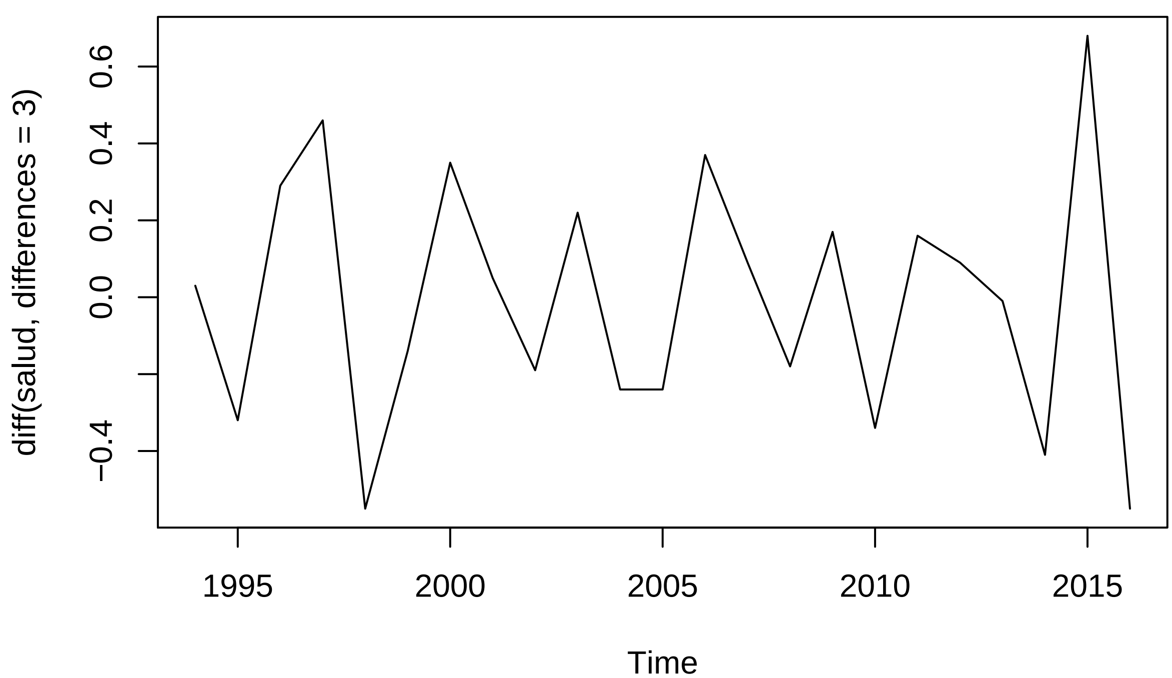

```r
plot(diff(salud, differences = 1))
```



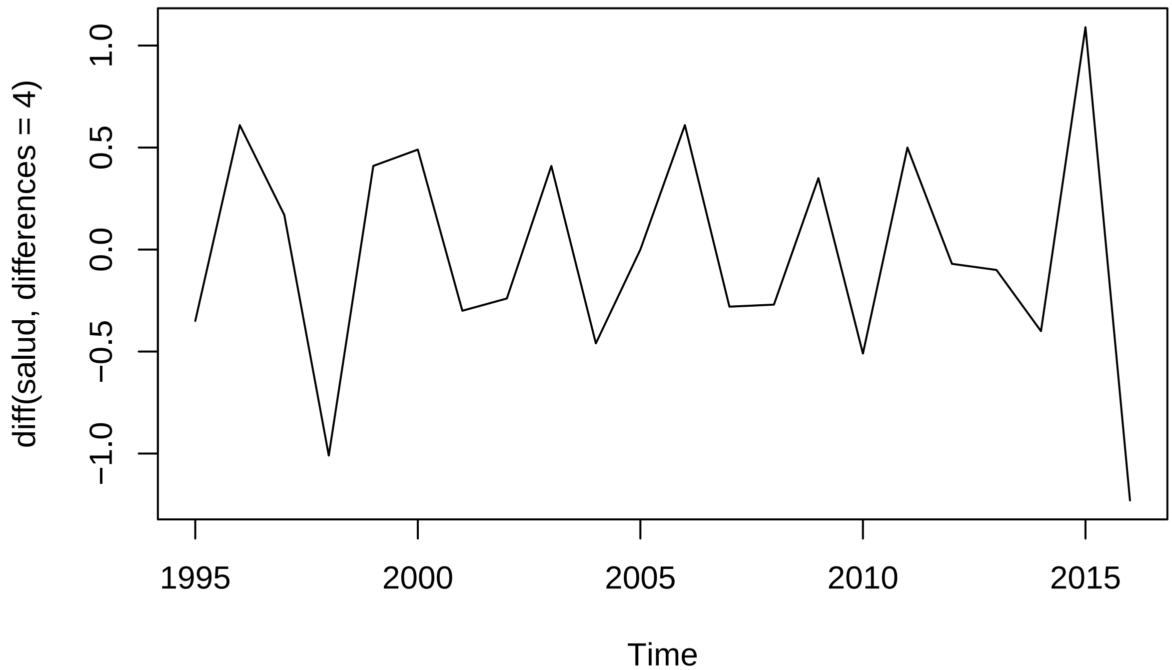

```r
plot(diff(salud, differences = 2))
```




```r
plot(diff(salud, differences = 3))
```




```r
plot(diff(salud, differences = 4))
```




```r
var(salud)
```

```
## [1] 0.1425915
```

```r
var(diff(salud, differences = 1))
```

```
## [1] 0.02018767
```

```r
var(diff(salud, differences = 2))
```

```
## [1] 0.03930851
```

```r
var(diff(salud, differences = 3))
```

```
## [1] 0.1060265
```

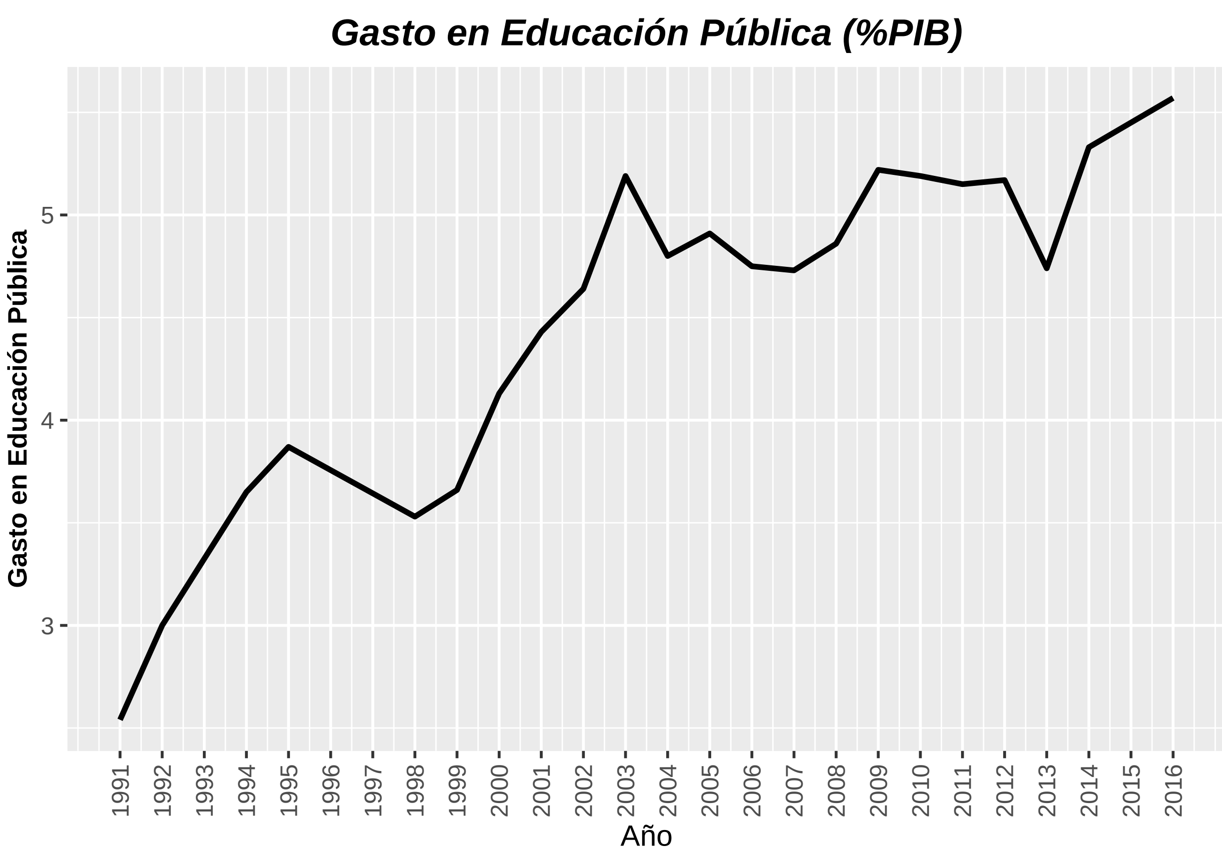
```r
var(diff(salud, differences = 4))
```

```
## [1] 0.3093195
```

Chapter 8

Orden Integración de la serie Gasto en Educación

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'Gasto_Educacion_PorcGDP',
                     titulo = 'Gasto en Educación Pública (%PIB)',
                     titulo_y = 'Gasto en Educación Pública')
gf1
```

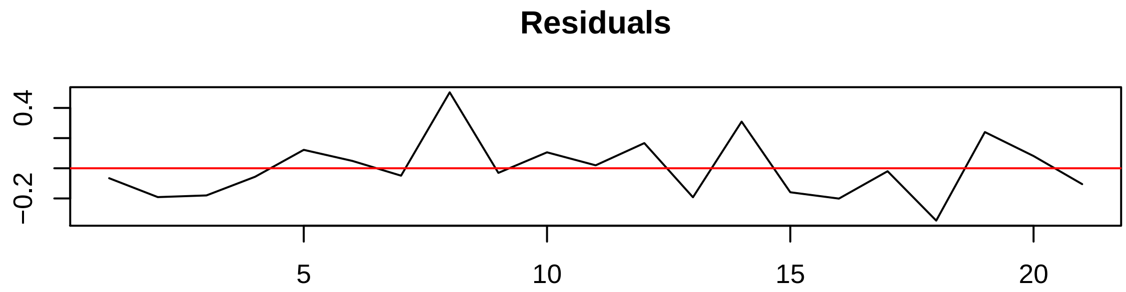


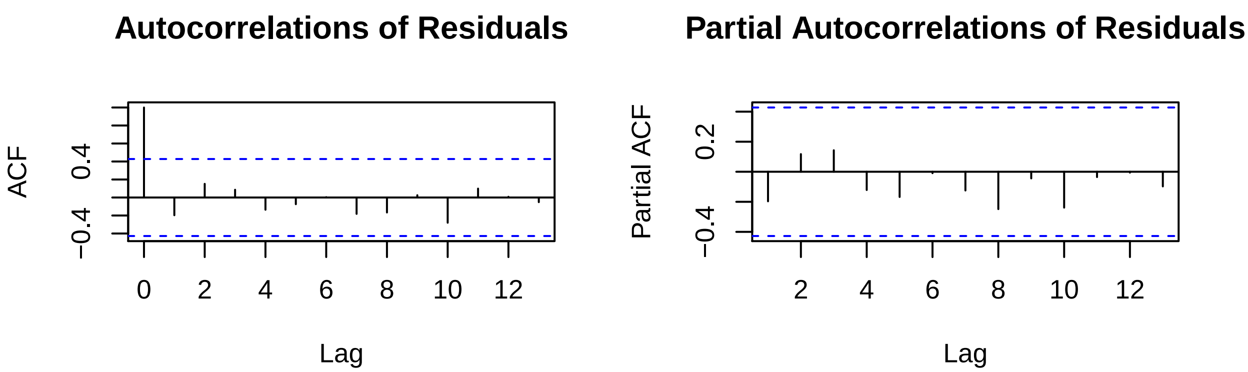

```
edu <- ts(series_db$Gasto_Educacion_PorcGDP,start = 1991, end = 2016, frequency = 1)
edu.ct <- ur.df(edu,lags=4,type='trend')
```

```r
summary(edu.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3471 -0.1591 -0.0305  0.1057  0.5037
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.502036   0.986763   1.522    0.150
## z.lag.1     -0.400896   0.337211  -1.189    0.254
## tt           0.031642   0.036124   0.876    0.396
## z.diff.lag1 -0.001856   0.284202  -0.007    0.995
## z.diff.lag2  0.250190   0.252359   0.991    0.338
## z.diff.lag3  0.045158   0.293993   0.154    0.880
## z.diff.lag4 -0.547433   0.271068  -2.020    0.063 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2368 on 14 degrees of freedom
## Multiple R-squared:  0.4685, Adjusted R-squared:  0.2408
## F-statistic: 2.057 on 6 and 14 DF,  p-value: 0.1249
##
##
## Value of test-statistic is: -1.1889 3.404 1.236
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```r
plot(edu.ct)
```





El valor del estadístico-t para la hipótesis nula de $\gamma = 0$ y el valor crítico de τ a univel de significancia del 5% reportado en las tablas de Dickey-Fuller demuestran que no es posible rechazar la hipótesis nula de la existencia de una raíz unitaria dada la presencia del término constante (drift) y la tendencia temporal (trend).

Recordemos que el poder de la prueba puede verse reducido debido a la presencia de términos drift/trend innecesarios, por lo que probaremos si la presencia del término temporal es necesaria dada una raíz unitaria. Para ello utilizaremos el estadístico ϕ_3 que prueba la hipótesis conjunta $a_2 = \gamma = 0$. Derivado de los resultados mostrados en las tablas anteriores, se puede aceptar la hipótesis nula, por lo que se tiene una raíz unitaria sin la presencia de un término temporal.

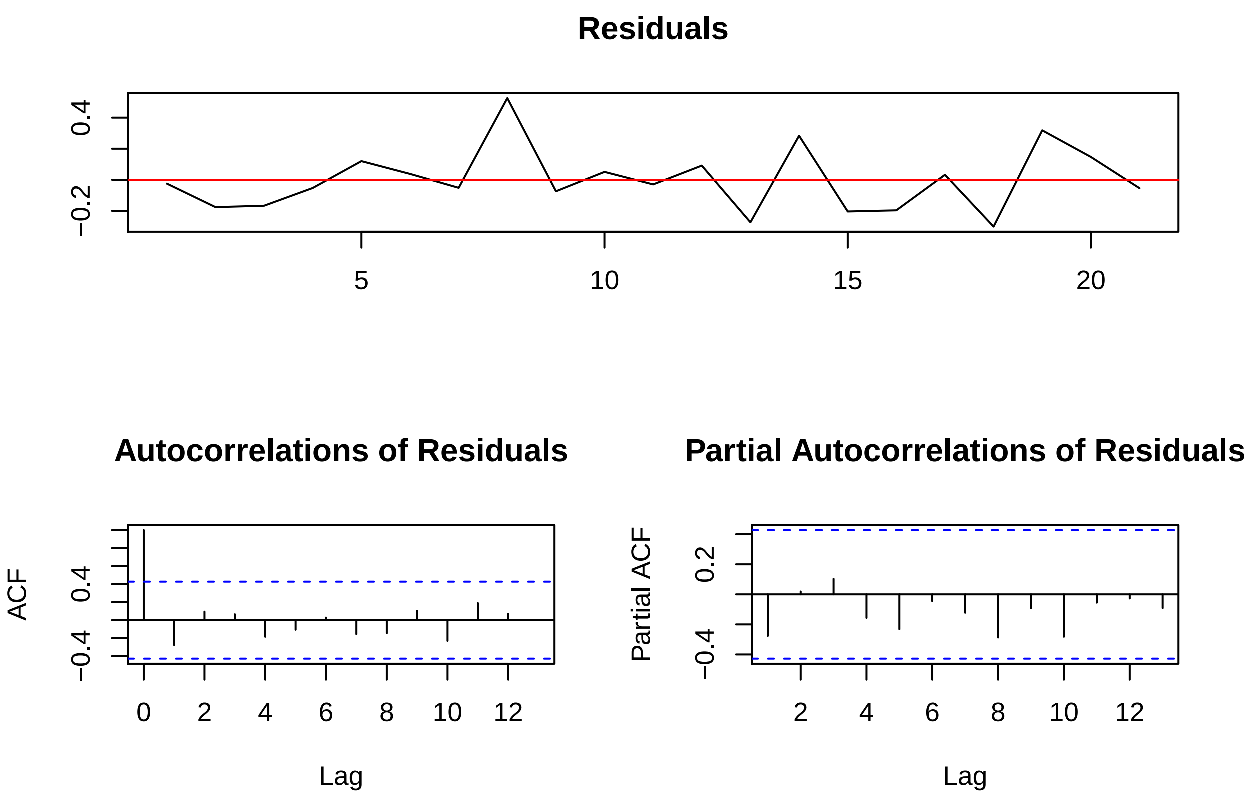
Probaremos si la serie requiere del término constante:

```
edu.co <- ur.df(edu,type='drift',lags=4)
summary(edu.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30135 -0.16691 -0.02998  0.09135  0.52575
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.71779    0.41159   1.744  0.10162
## z.lag.1     -0.11595    0.08811  -1.316  0.20795
## z.diff.lag1 -0.17614    0.20135  -0.875  0.39548
## z.diff.lag2  0.10972    0.19333   0.568  0.57876
## z.diff.lag3 -0.13544    0.20794  -0.651  0.52467
## z.diff.lag4 -0.67535    0.22659  -2.981  0.00933 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2349 on 15 degrees of freedom
## Multiple R-squared:  0.4394, Adjusted R-squared:  0.2525
## F-statistic: 2.351 on 5 and 15 DF,  p-value: 0.09148
##
##
## Value of test-statistic is: -1.3159 4.7969
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

```
plot(edu.co)
```



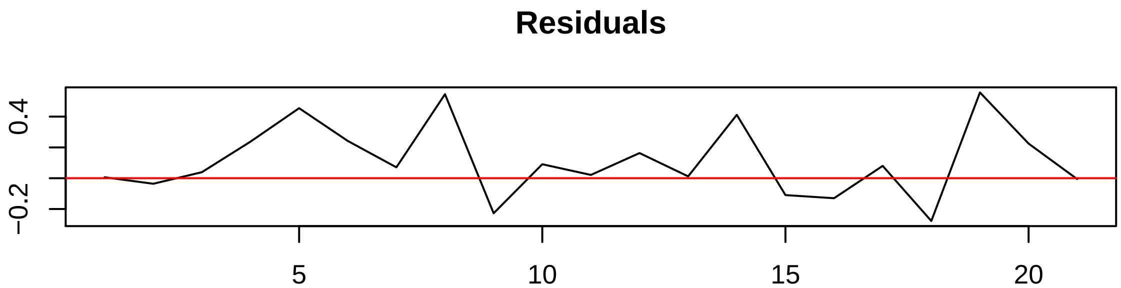
Derivado de los resultados anteriores se puede observar que la serie requiere la presencia de un término constante, por lo que veremos si alcanzamos estacionariedad diferenciando la serie

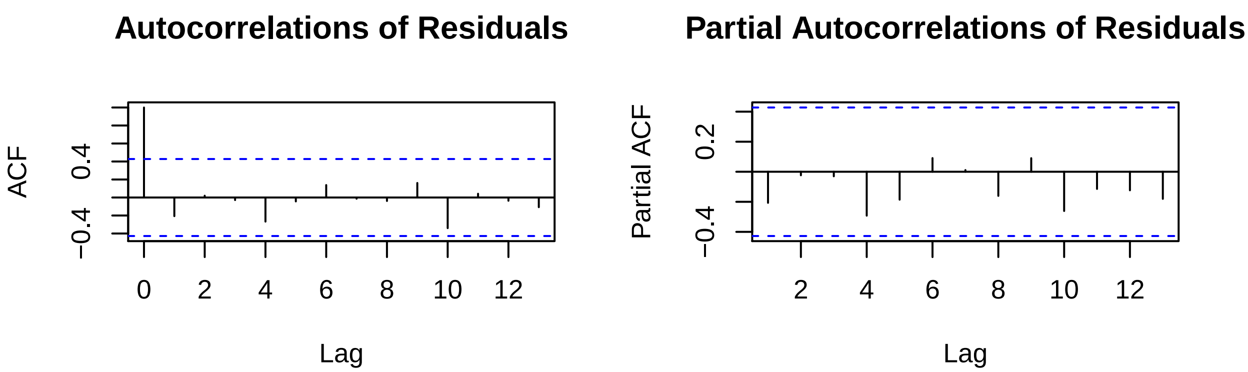
```
d.edu <- diff(edu,differences = 1)
d.edu.n <- ur.df(d.edu,type="none",lags=3)
```

```r
summary(d.edu.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27766 -0.00578  0.07066  0.23769  0.55677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -1.2161     0.3864  -3.147  0.00588 **
## z.diff.lag1   0.1779     0.3331   0.534  0.60023
## z.diff.lag2   0.4056     0.2913   1.393  0.18170
## z.diff.lag3   0.3965     0.2493   1.591  0.13012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2826 on 17 degrees of freedom
## Multiple R-squared:  0.596,  Adjusted R-squared:  0.5009
## F-statistic:  6.27 on 4 and 17 DF,  p-value: 0.002736
##
##
## Value of test-statistic is: -3.1472
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

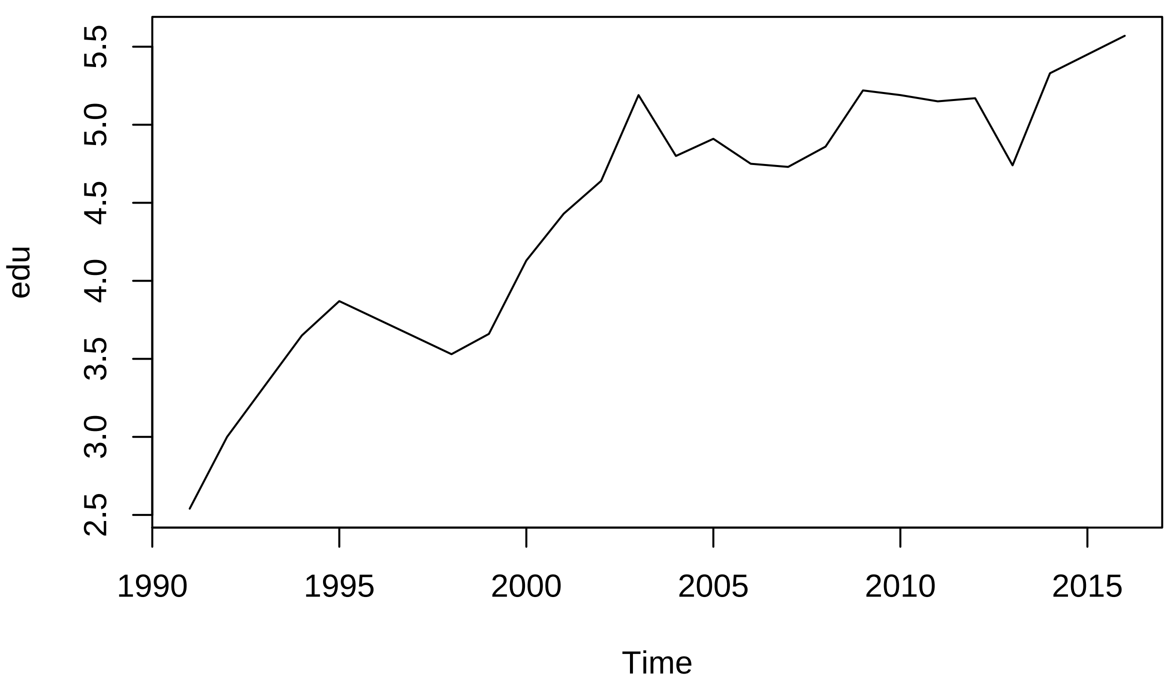
```r
plot(d.edu.n)
```



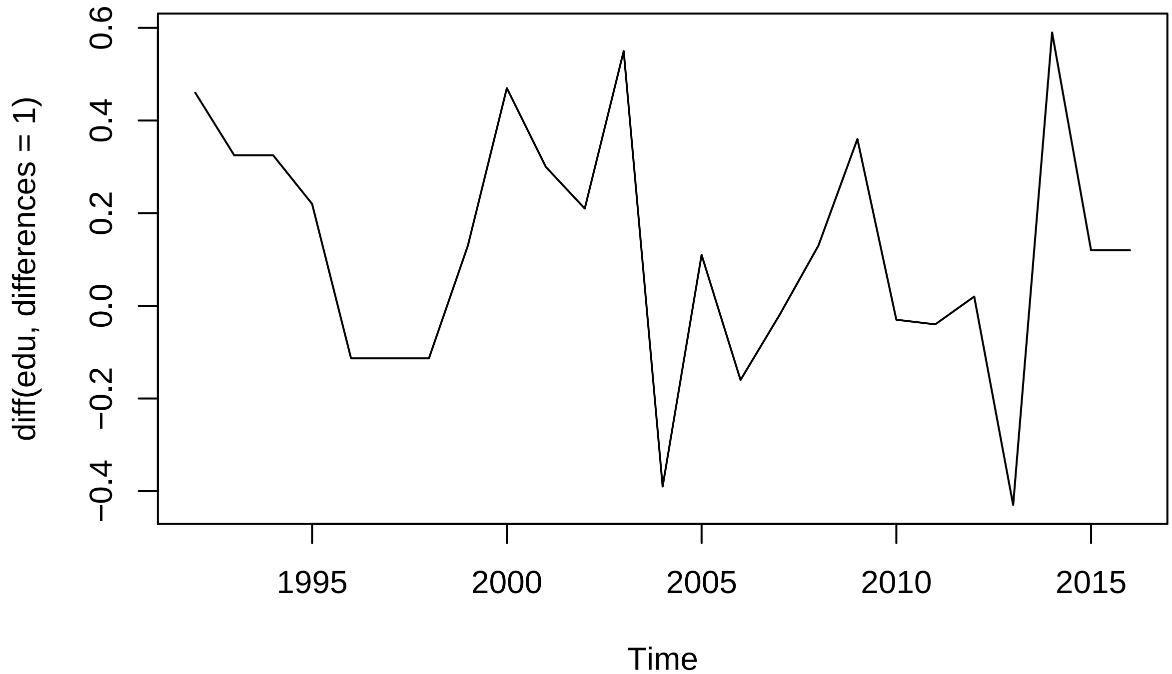


Por lo tanto. La serie de gasto en educación es $I(1)$ y de la forma $\nabla y_t = \alpha_0 + \gamma y_{t-1} + \epsilon_t$.

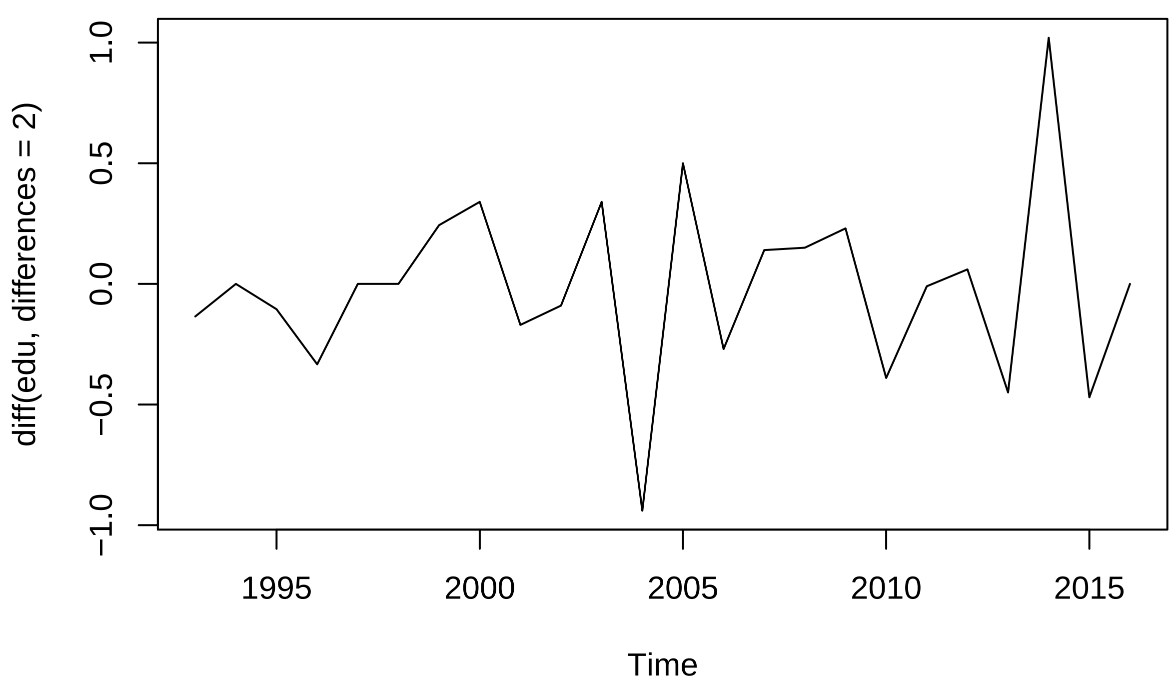
```r
plot(edu)
```



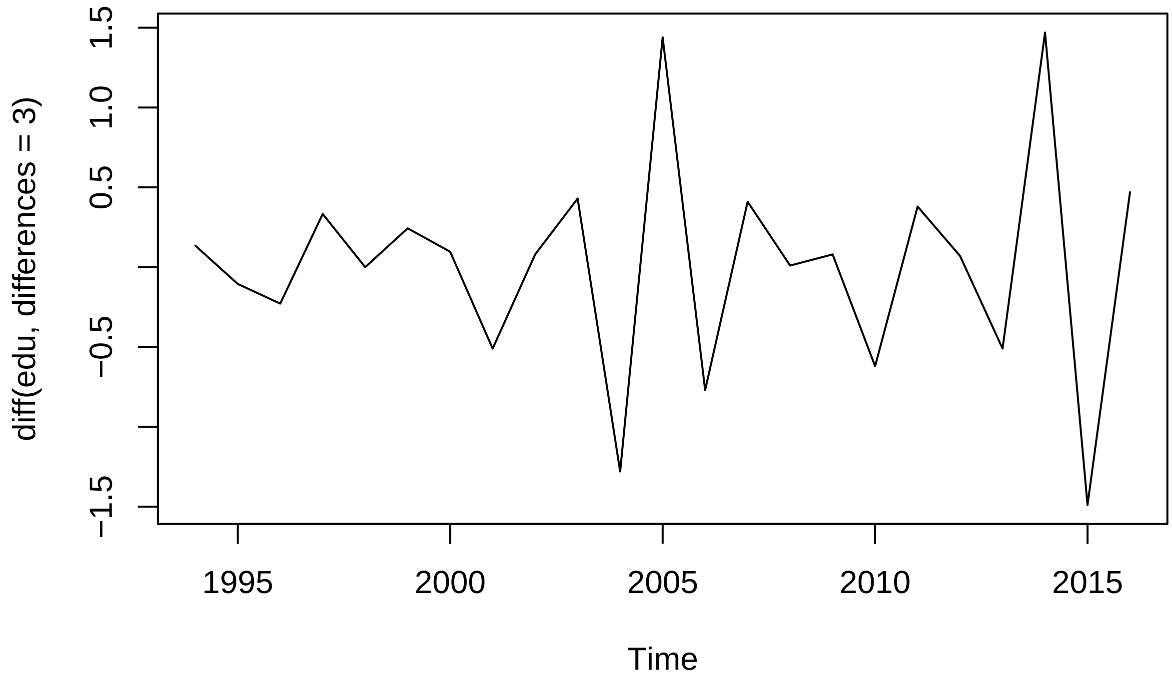

```r
plot(diff(edu, differences = 1))
```




```r
plot(diff(edu, differences = 2))
```




```r
plot(diff(edu, differences = 3))
```




```
var(edu)
```

```
## [1] 0.6882052
```

```
var(diff(edu, differences = 1))
```

```
## [1] 0.07159781
```

```
var(diff(edu, differences = 2))
```

```
## [1] 0.1471328
```

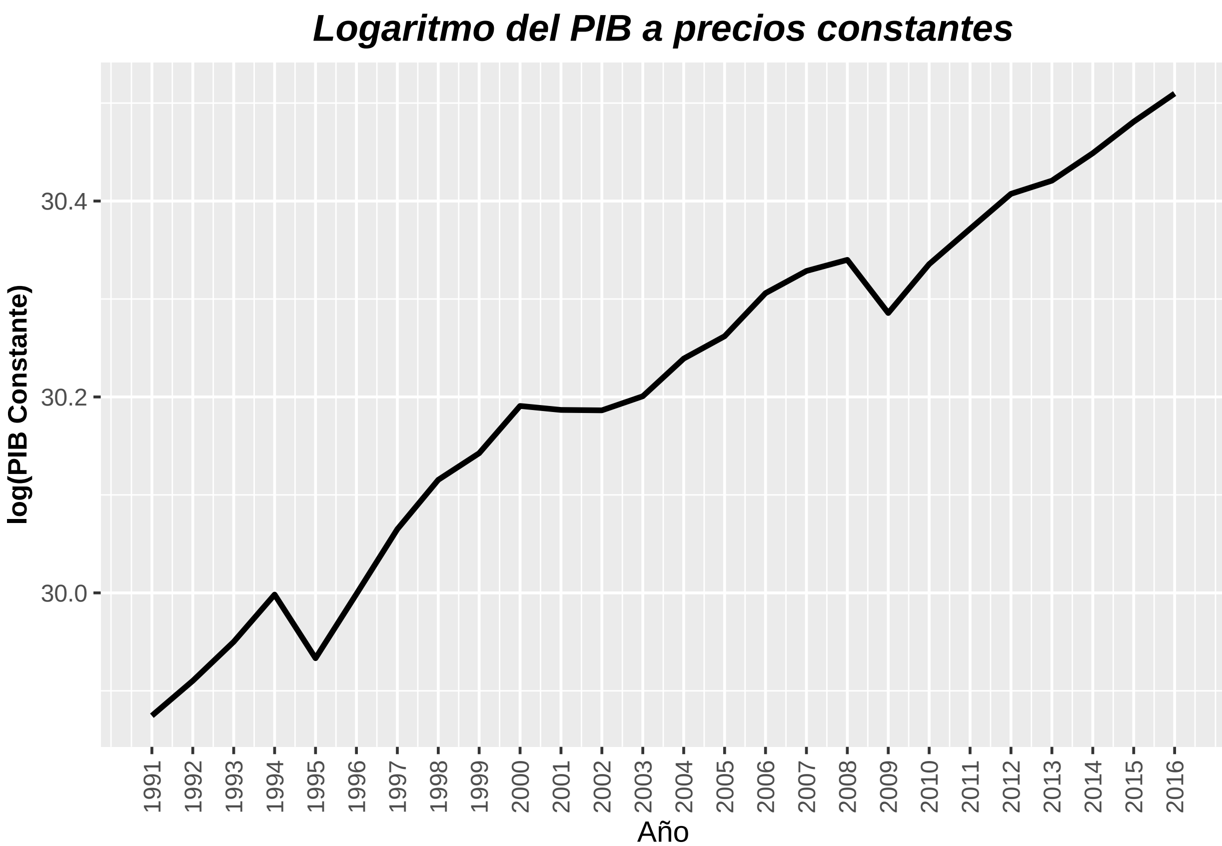
```
var(diff(edu, differences = 3))
```

```
## [1] 0.4812209
```

Chapter 9

Orden Integración de la serie ln(PIB) a precios ctes

```
gf1 <- grafica_serie(base_in = series_db,
                     eje_y = 'log_GDP_constante',
                     titulo = 'Logaritmo del PIB a precios constantes',
                     titulo_y = 'log(PIB Constante)')
gf1
```

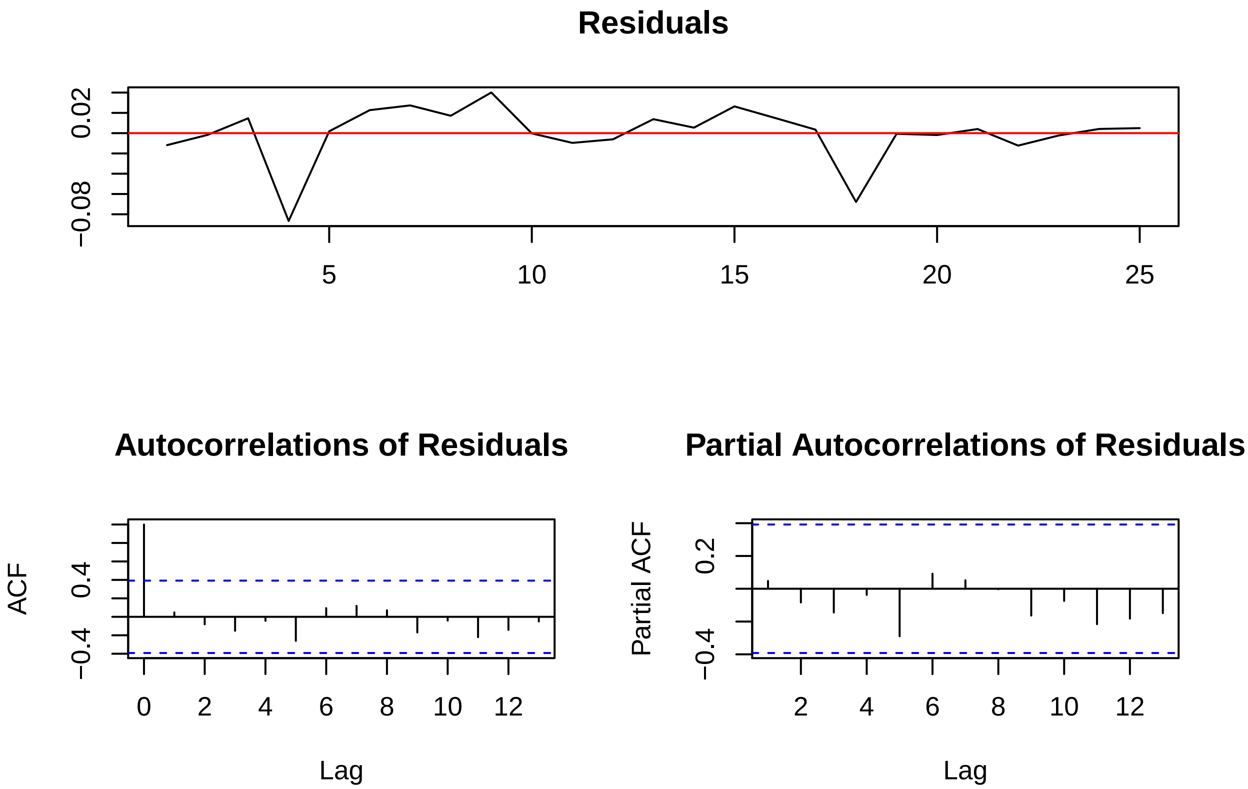



```
ln_pib_cte <- ts(series_db$log_GDP_constante,start = 1991, end = 2016, frequency = 1)
ln_pib_cte.ct <- ur.df(ln_pib_cte,lags=0,type='trend')
```

```r
summary(ln_pib_cte.ct)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.086613 -0.002315  0.003448  0.014649  0.040110
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.295480   5.198222   2.750   0.0117 *
## z.lag.1     -0.477299   0.173928  -2.744   0.0118 *
## tt           0.011074   0.004245   2.609   0.0160 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02791 on 22 degrees of freedom
## Multiple R-squared:  0.261,  Adjusted R-squared:  0.1938
## F-statistic: 3.885 on 2 and 22 DF,  p-value: 0.03589
##
##
## Value of test-statistic is: -2.7442 9.4963 3.8854
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```r
plot(ln_pib_cte.ct)
```



El valor del estadístico-t para la hipótesis nula de $\gamma = 0$ y el valor crítico de τ a univel de significancia del 5% reportado en las tablas de Dickey-Fuller demuestran que no es posible rechazar la hipótesis nula de la existencia de una raíz unitaria dada la presencia del término constante (drift) y la tendencia temporal (trend).

Recordemos que el poder de la prueba puede verse reducido debido a la presencia de términos drift/trend innecesarios, por lo que probaremos si la presencia del término temporal es necesaria dada una raíz unitaria. Para ello utilizaremos el estadístico ϕ_3 que prueba la hipótesis conjunta $a_2 = \gamma = 0$. Derivado de los resultados mostrados en las tablas anteriores, se puede aceptar la hipótesis nula, por lo que se tiene una raíz unitaria sin la presencia de un término temporal.

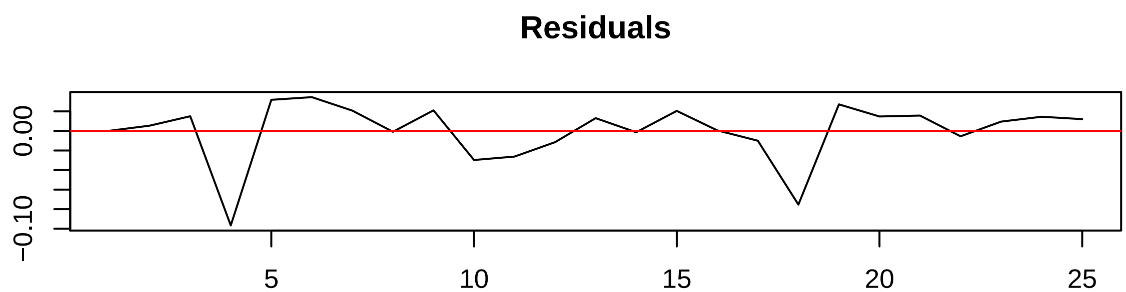
Probaremos si la serie requiere del término constante:

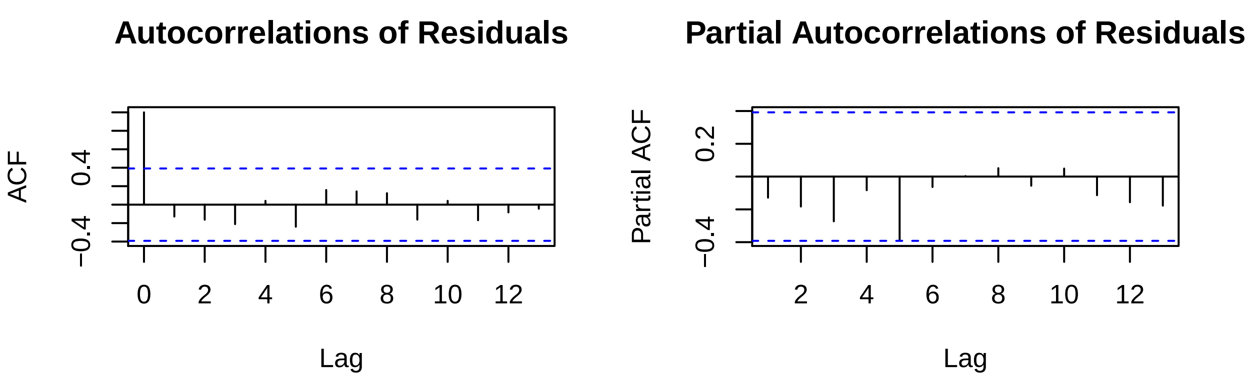
```
ln_pib_cte.co <- ur.df(ln_pib_cte,type='drift',lags=0)
summary(ln_pib_cte.co)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.096643 -0.005473  0.009552  0.015738  0.034579
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.96603    1.07194   0.901    0.377
## z.lag.1     -0.03115    0.03550  -0.878    0.389
##
## Residual standard error: 0.03123 on 23 degrees of freedom
## Multiple R-squared:  0.03239,    Adjusted R-squared:  -0.009675
## F-statistic:  0.77 on 1 and 23 DF,  p-value: 0.3893
##
##
## Value of test-statistic is: -0.8775 8.656
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

```
plot(ln_pib_cte.co)
```





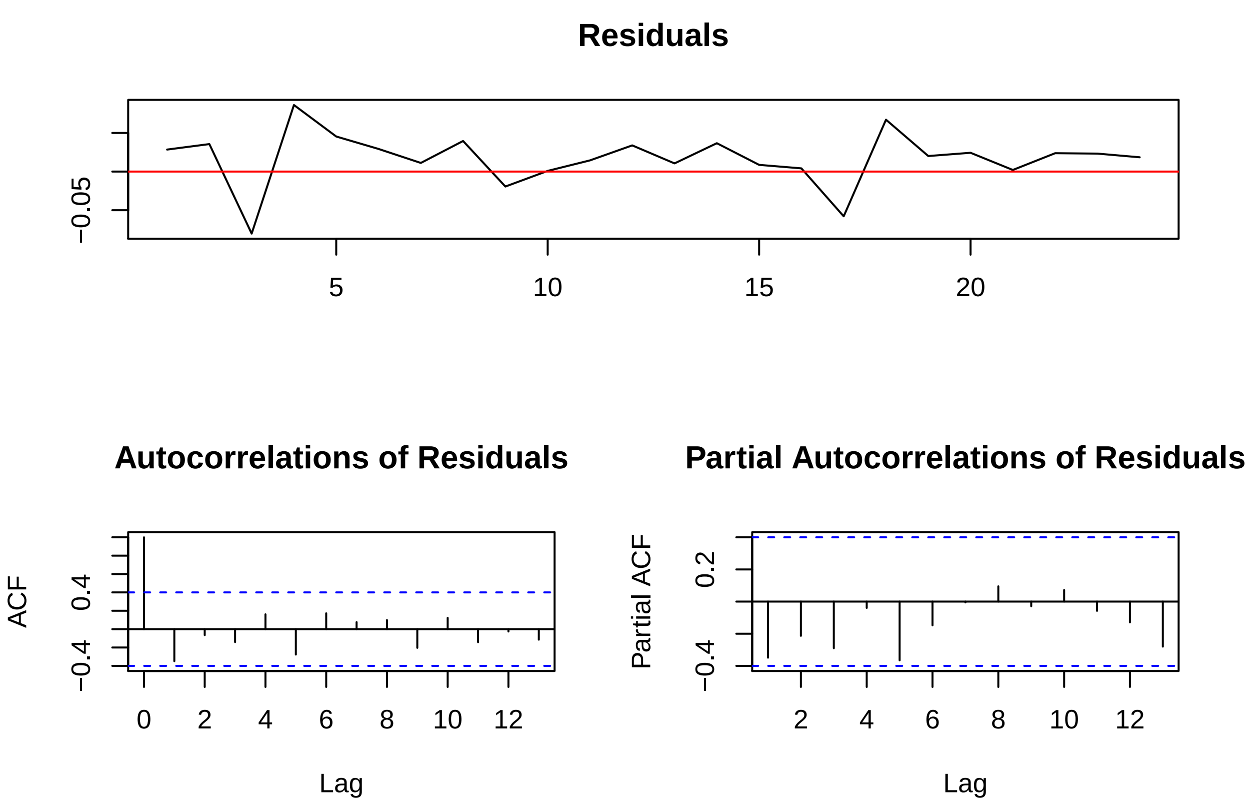
Derivado de los resultados anteriores se puede observar que la serie requiere la presencia de un término constante, por lo que veremos si alcanzamos estacionariedad diferenciando la serie

```
d.ln_pib_cte <- diff(ln_pib_cte,differences = 1)
d.ln_pib_cte.n <- ur.df(d.ln_pib_cte,type="none",lags=0)
summary(d.ln_pib_cte.n)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
```

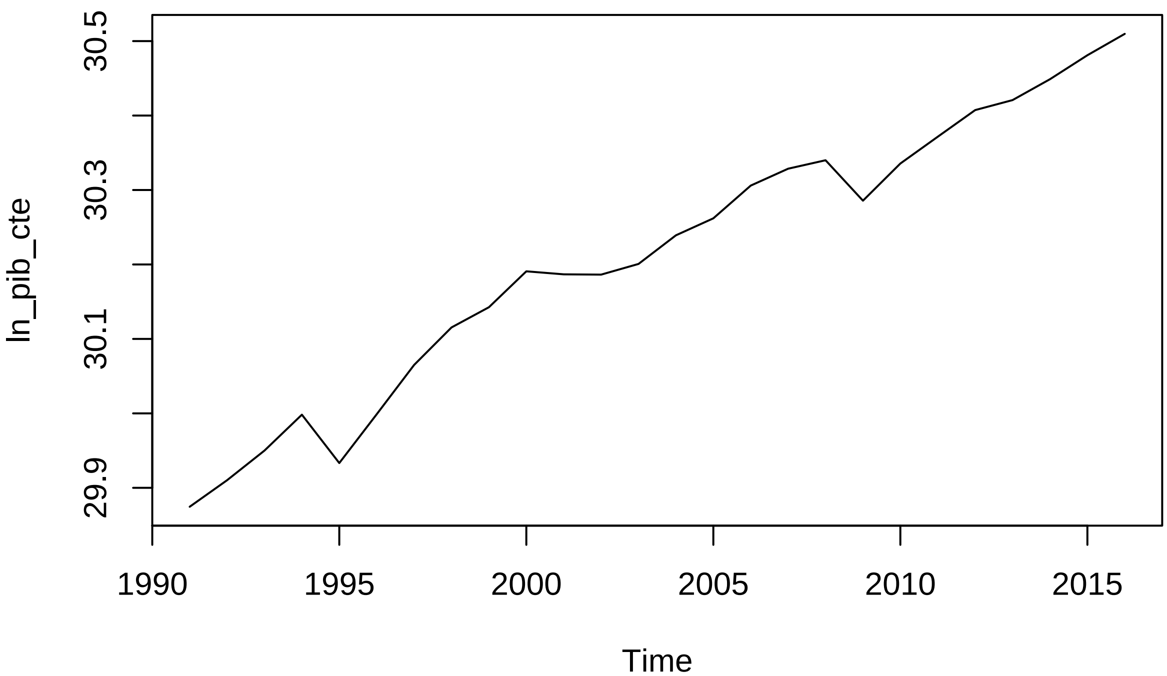
```
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.080294  0.007563  0.021698  0.034319  0.086172
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1  -0.6824     0.1964  -3.474  0.00205 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03855 on 23 degrees of freedom
## Multiple R-squared:  0.3441, Adjusted R-squared:  0.3156
## F-statistic: 12.07 on 1 and 23 DF,  p-value: 0.002054
##
##
## Value of test-statistic is: -3.4739
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.66 -1.95  -1.6
```

```r
plot(d.ln_pib_cte.n)
```

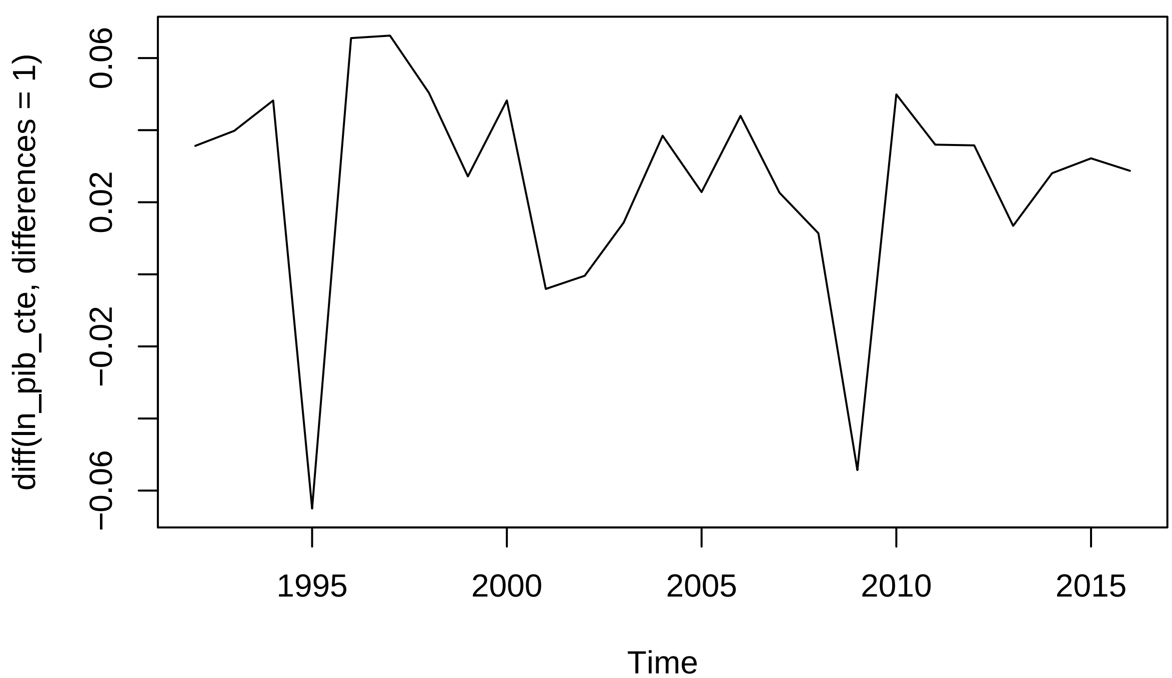


Por lo tanto. La serie de ln(PIB) a precios cte es $I(1)$ y de la forma $\nabla y_t = \alpha_0 + \gamma y_{t-1} + \epsilon_t$.

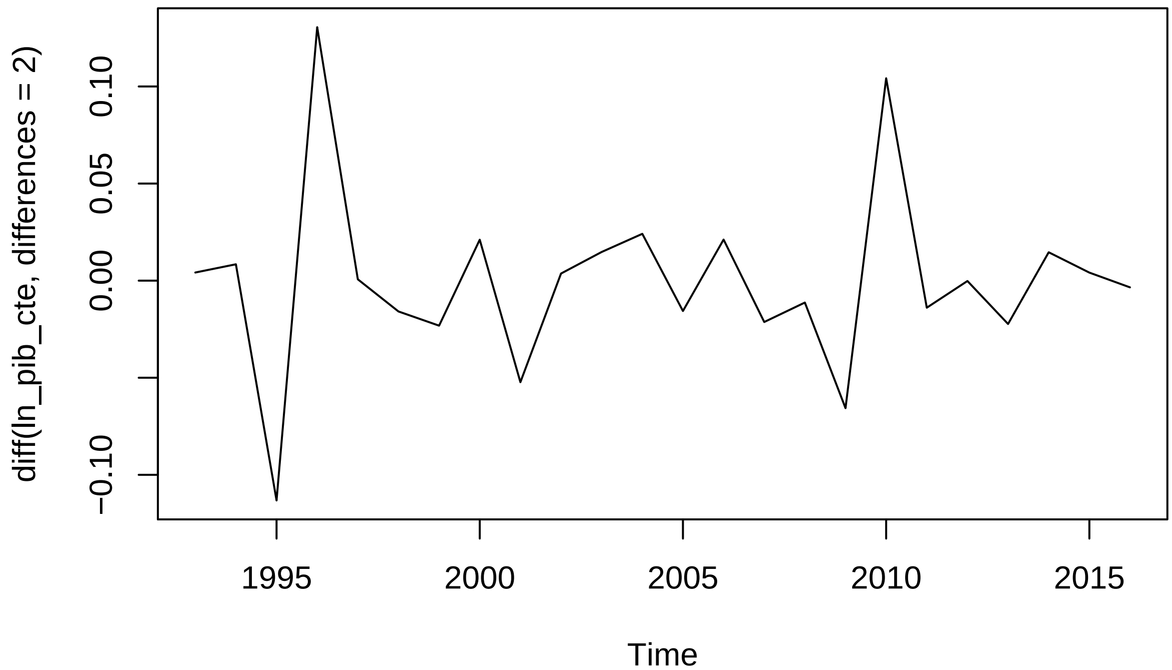
```
plot(ln_pib_cte)
```



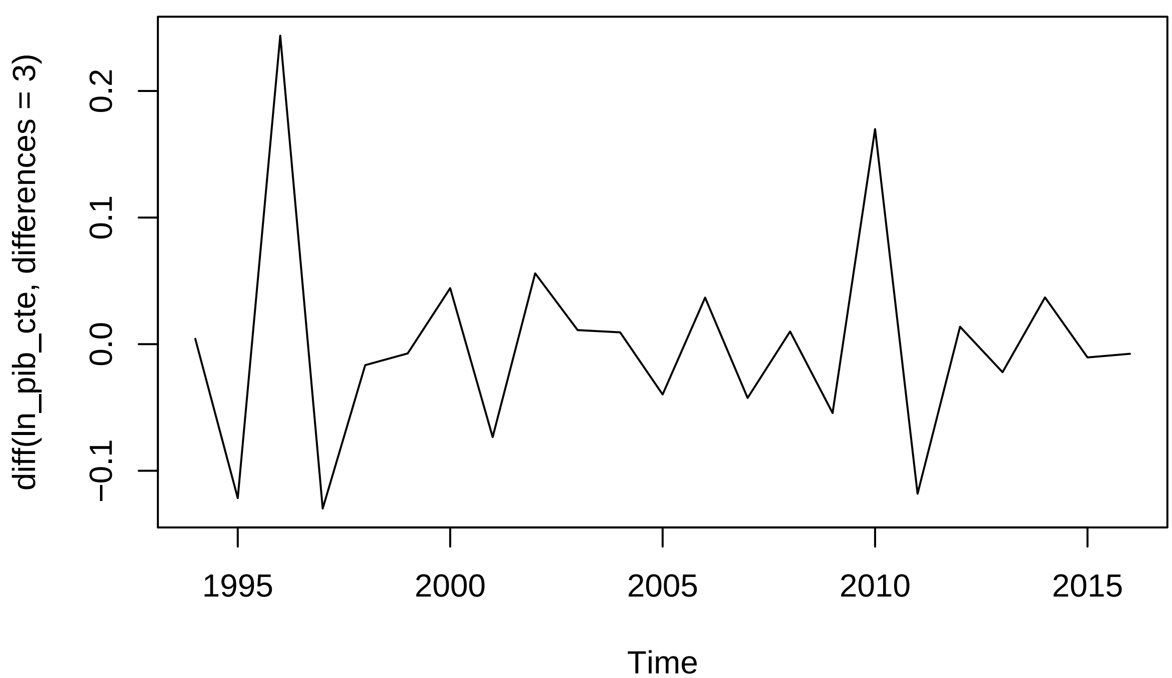

```
plot(diff(ln_pib_cte, differences = 1))
```



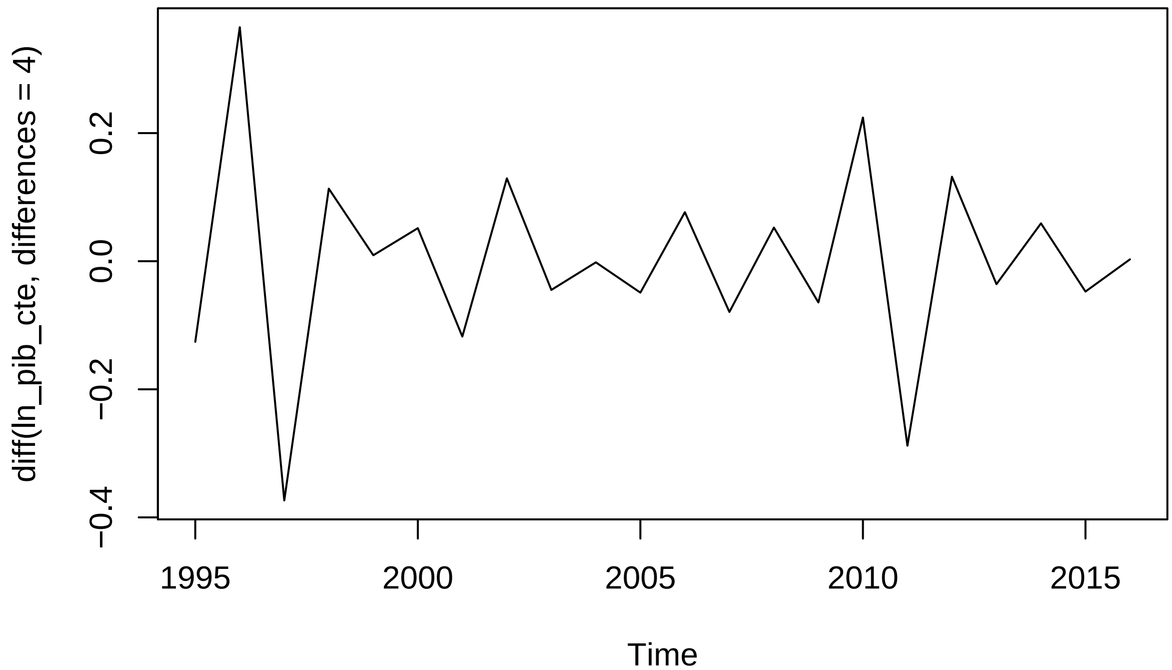

```
plot(diff(ln_pib_cte, differences = 2))
```




```
plot(diff(ln_pib_cte, differences = 3))
```




```
plot(diff(ln_pib_cte, differences = 4))
```




```
var(ln_pib_cte)
```

```
## [1] 0.03468178
```

```
var(diff(ln_pib_cte, differences = 1))
```

```
## [1] 0.0009661952
```

```
var(diff(ln_pib_cte, differences = 2))
```

```
## [1] 0.002265285
```

```
var(diff(ln_pib_cte, differences = 3))
```

```
## [1] 0.007041242
```

```
var(diff(ln_pib_cte, differences = 4))
```

```
## [1] 0.02457956
```

Chapter 10

Cointegración (Teoría)

Se probará la hipótesis $H_1(r) : \Pi = \alpha\beta'$ es decir, si Π es de rango reducido. para ello utilizaremos la prueba del estadístico de la traza y del máximo eigenvalor.

```
library(urca)
data(UKpppuip)
names(UKpppuip)
```

```
## [1] "p1"     "p2"     "e12"    "i1"     "i2"     "doilp0" "doilp1"
```

```
attach(UKpppuip)
```

```
## The following objects are masked from UKpppuip (pos = 8):
##
##     doilp0, doilp1, e12, i1, i2, p1, p2
```

```
dat1 <- cbind(p1,p2,e12,i1,i2)
dat2 <- cbind(doilp0,doilp1)
args('ca.jo')
```

```
## function (x, type = c("eigen", "trace"), ecdet = c("none", "const",
##     "trend"), K = 2, spec = c("longrun", "transitory"), season = NULL,
##     dumvar = NULL)
## NULL
```

```
H1 <- ca.jo(dat1, type='trace',K=2,season=4,dumvar=dat2)

H1.trace <- summary(ca.jo(dat1,type='trace',K=2,season=4,dumvar=dat2))
H1.eigen <- summary(ca.jo(dat1,type='eigen',K=2,season=4,dumvar=dat2))
```

Considerando el estadístico del máximo eigenvalor, la hipótesis de no cointegración no puede ser rechazada, tal y como se observa en la siguiente tabla:

```
H1.eigen
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend
##
```

```
## Eigenvalues (lambda):
## [1] 0.40672818 0.28538240 0.25415335 0.10230406 0.08287097
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 4 |  5.19  6.50  8.18 11.65
## r <= 3 |  6.48 12.91 14.90 19.19
## r <= 2 | 17.59 18.90 21.07 25.75
## r <= 1 | 20.16 24.78 27.14 32.14
## r = 0  | 31.33 30.84 33.32 38.78
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##             p1.l2      p2.l2    e12.l2      i1.l2      i2.l2
## p1.l2   1.0000000   1.000000  1.000000  1.0000000  1.0000000
## p2.l2  -0.9086265  -1.143047 -1.272628 -2.4001444 -1.4528820
## e12.l2 -0.9321133  -3.363042  1.113631  1.1221619 -0.4805235
## i1.l2  -3.3746393  35.243576  2.746828 -0.4088865  2.2775510
## i2.l2  -1.8906210 -32.917370 -2.835714  2.9863624  0.7628011
##
## Weights W:
## (This is the loading matrix)
##
##             p1.l2         p2.l2       e12.l2        i1.l2       i2.l2
## p1.d  -0.06816507  0.0011795779 -0.002790218  0.001373599 -0.01333013
## p2.d  -0.01773477  0.0001220008 -0.014159241  0.013178503  0.00755575
## e12.d  0.10065321 -0.0001432122 -0.055628059 -0.035400025 -0.04707585
## i1.d   0.03434737 -0.0041631581 -0.010363374  0.012309982 -0.02394672
## i2.d   0.05766426  0.0082830953  0.004821036  0.026984801 -0.01006765
```

Sin embargo, el estadístico de la traza indica que existen 2 relaciones de cointegración.

```
H1.trace
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , with linear trend
##
## Eigenvalues (lambda):
## [1] 0.40672818 0.28538240 0.25415335 0.10230406 0.08287097
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 4 |  5.19  6.50  8.18 11.65
## r <= 3 | 11.67 15.66 17.95 23.52
## r <= 2 | 29.26 28.71 31.52 37.22
## r <= 1 | 49.42 45.23 48.28 55.43
## r = 0  | 80.75 66.49 70.60 78.87
##
```

```
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##              p1.l2      p2.l2     e12.l2      i1.l2      i2.l2
## p1.l2   1.0000000   1.000000  1.000000  1.0000000  1.0000000
## p2.l2  -0.9086265  -1.143047 -1.272628 -2.4001444 -1.4528820
## e12.l2 -0.9321133  -3.363042  1.113631  1.1221619 -0.4805235
## i1.l2  -3.3746393  35.243576  2.746828 -0.4088865  2.2775510
## i2.l2  -1.8906210 -32.917370 -2.835714  2.9863624  0.7628011
##
## Weights W:
## (This is the loading matrix)
##
##              p1.l2         p2.l2       e12.l2        i1.l2       i2.l2
## p1.d  -0.06816507  0.0011795779 -0.002790218  0.001373599 -0.01333013
## p2.d  -0.01773477  0.0001220008 -0.014159241  0.013178503  0.00755575
## e12.d  0.10065321 -0.0001432122 -0.055628059 -0.035400025 -0.04707585
## i1.d   0.03434737 -0.0041631581 -0.010363374  0.012309982 -0.02394672
## i2.d   0.05766426  0.0082830953  0.004821036  0.026984801 -0.01006765
```

Peor aún, existe la posiblidad de una tercera combinación lineal estacionaria debido a la cercanía entre los igenvectores 2 y 3.

```
H1@lambda
```

```
## [1] 0.40672818 0.28538240 0.25415335 0.10230406 0.08287097
```

Para generar una decisión final acerca del orden de integración, debemos observar varios aspectos como las matrices $\hat{\alpha}$ y $\hat{\beta}$ así como las relaciones de cointegración $\hat{\beta}'y_t$ y aquellas que son corregidas por las influecnias de corto plazo $\hat{\beta}'R_{1t}$. Para obtener las tablas similares a las reportadas en el paper de Johansen y Joselius, las matrices fueron normalizadas respectivamente:

```
beta <- H1@V
beta[,2] <- beta[,2]/beta[4,2]
beta[,3] <- beta[,3]/beta[4,3]
beta
```

```
##              p1.l2       p2.l2     e12.l2      i1.l2      i2.l2
## p1.l2   1.0000000  0.02837397  0.3640563  1.0000000  1.0000000
## p2.l2  -0.9086265 -0.03243276 -0.4633082 -2.4001444 -1.4528820
## e12.l2 -0.9321133 -0.09542285  0.4054245  1.1221619 -0.4805235
## i1.l2  -3.3746393  1.00000000  1.0000000 -0.4088865  2.2775510
## i2.l2  -1.8906210 -0.93399632 -1.0323596  2.9863624  0.7628011
```

Y la matriz alpha es la siguiente

```
alpha <- H1@PI%*%solve(t(beta))
alpha
```

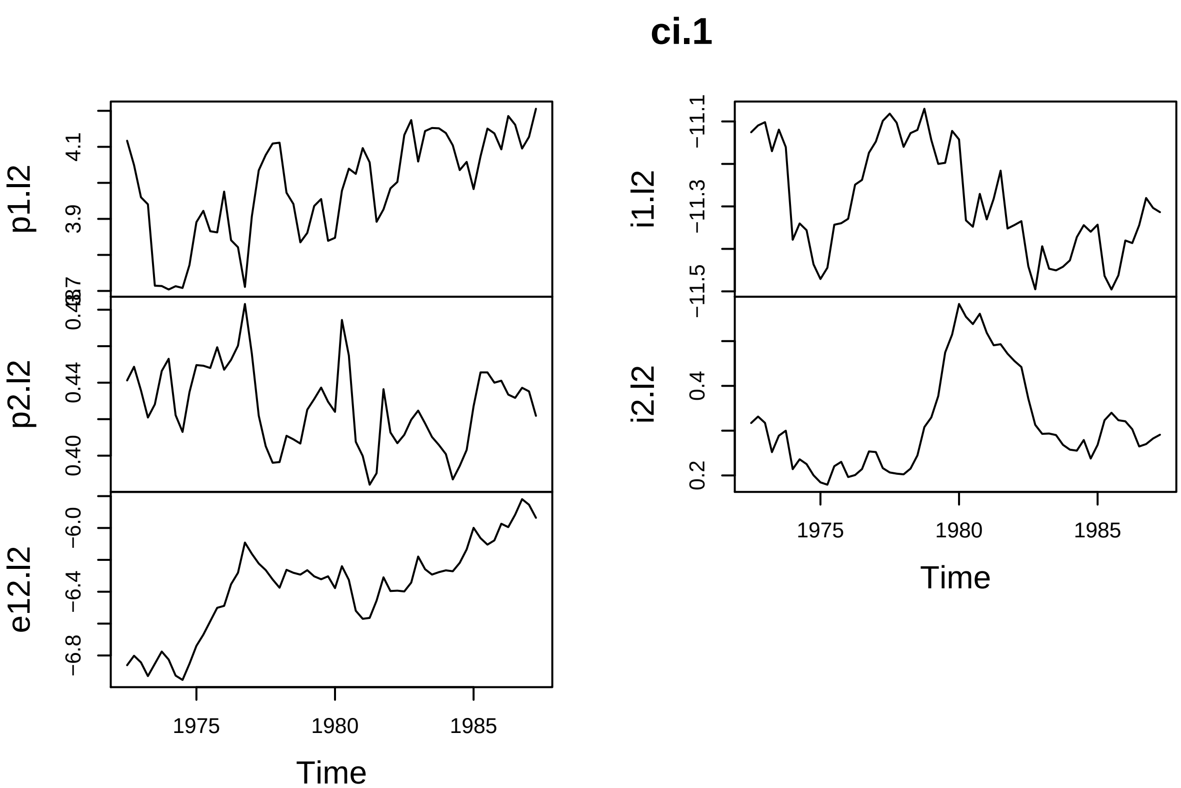
```
##              p1.l2        p2.l2       e12.l2        i1.l2       i2.l2
## p1.d  -0.06816507  0.041572544 -0.007664249  0.001373599 -0.01333013
## p2.d  -0.01773477  0.004299743 -0.038893000  0.013178503  0.00755575
## e12.d  0.10065321 -0.005047310 -0.152800708 -0.035400025 -0.04707585
## i1.d   0.03434737 -0.146724578 -0.028466405  0.012309982 -0.02394672
## i2.d   0.05766426  0.291925899  0.013242558  0.026984801 -0.01006765
```

Se puede observar que los valores de $\hat{\alpha}_{i.2}$ para $i = 1, 2, 3$ son cercanos a cero para el segundo vector de cointegración, por lo que la pequeña estimación del eigenvalor λ_2 se puede atribuir a que estos valores se encuentran justo en la frontera de la no estacionariedad impactando la prueba de hipótesis.

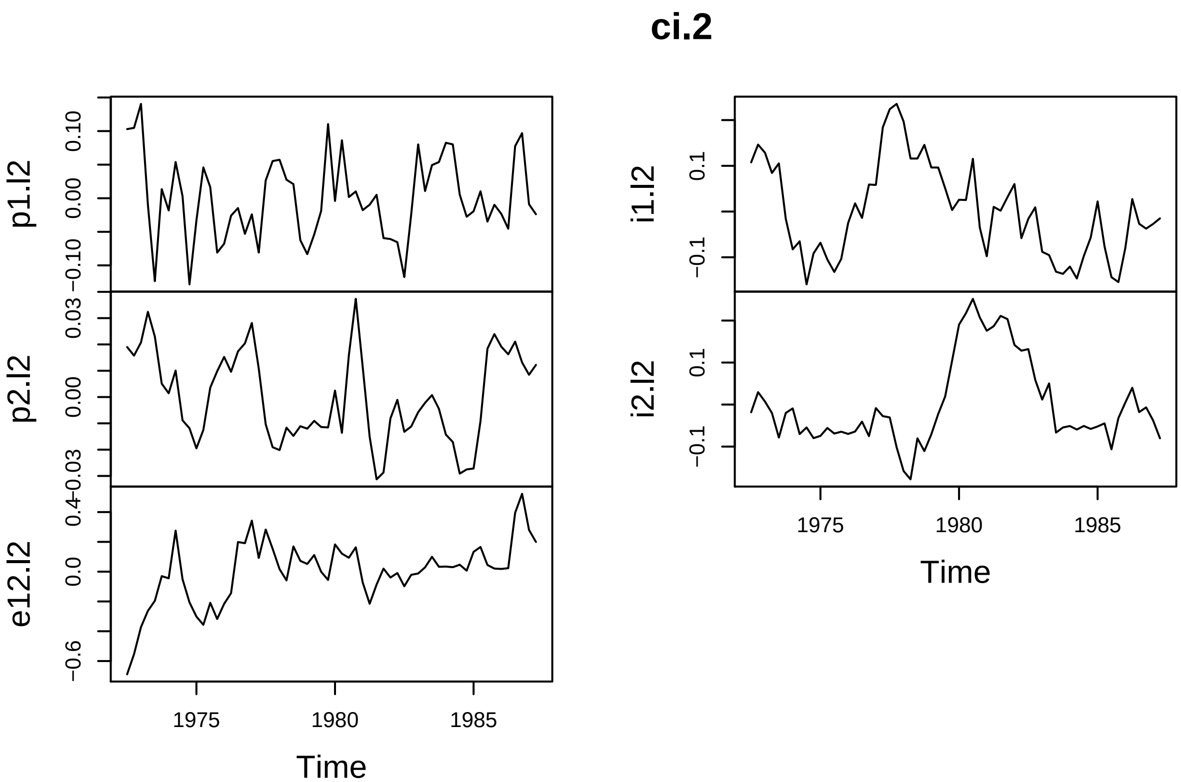
Además Johanses y Joselius investigaron de manera gráfica las relaciones de cointegración, ya que si en efecto existen dos relaciones de cointegración ($r = 2$) entonces estas deberían observarse como un proceso estacionario. Sin embargo, debido las influencias de corto plazo que influyen en el proceso de estimación, los autores decidieron también analizarlas.

```
beta1 <- cbind(beta[,1:2], H1@V[,3:5])
ci.1 <- ts((H1@x%*%beta1)[-c(1,2),],start=c(1972,3),end=c(1987,2),frequency = 4)
ci.2 <- ts((H1@RK%*%beta1),start=c(1972,3),end=c(1987,2),frequency = 4)
```

```
plot(ci.1)
```




```
plot(ci.2)
```



Basado en las pruebas estadísticas, los elementos de la matriz $\hat{\alpha}$ y la tendencia de las relaciones de cointegración, Johansen y Joselius decidieron mantener la hipótesis de que el grado de cointegración es $r = 2$

```
set.seed(12345)
e1 <- rnorm(250, 0, 0.5)
e2 <- rnorm(250, 0, 0.5)
e3 <- rnorm(250, 0, 0.5)

u1.ar1 <- arima.sim(model = list(ar=0.75),
                    innov = e1, n=250)
u2.ar1 <- arima.sim(model = list(ar=0.3),
                    innov = e2, n=250)
y3 <- cumsum(e3)
y1 <- 0.8 * y3 + u1.ar1
y2 <- -0.3 * y3 + u2.ar1

y.mat <- data.frame(y1,y2,y3)

vecm <- ca.jo(y.mat)

jo.results <- summary(vecm)

vecm.r2 <- cajorls(vecm, r=2)
class(jo.results)
```

```
## [1] "sumurca"
## attr(,"package")
## [1] "urca"
```

```r
slotNames(jo.results)
```

```
##  [1] "classname" "test.name" "testreg"   "teststat"  "cval"
##  [6] "bpoint"    "signif"    "model"     "type"      "auxstat"
## [11] "lag"       "H"         "A"         "lambda"    "pval"
## [16] "V"         "W"         "P"
```

```r
vecm.r2$beta
```

```
##              ect1      ect2
## y1.l2  1.0000000 0.0000000
## y2.l2  0.0000000 1.0000000
## y3.l2 -0.7328534 0.2951962
```

```r
vecm.r2$rlm
```

```
##
## Call:
## lm(formula = substitute(form1), data = data.mat)
##
## Coefficients:
##           y1.d       y2.d       y3.d
## ect1      -0.331293   0.064612   0.012682
## ect2       0.094473  -0.709385  -0.009165
## constant   0.168371  -0.027019   0.025255
## y1.dl1    -0.227677   0.027012   0.068158
## y2.dl1     0.144452  -0.715607   0.040487
## y3.dl1     0.123467  -0.290828  -0.075251
```

```r
summary(vecm.r2$rlm)
```

```
## Response y1.d :
##
## Call:
## lm(formula = y1.d ~ ect1 + ect2 + constant + y1.dl1 + y2.dl1 +
##     y3.dl1 - 1, data = data.mat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4471 -0.4862 -0.0256  0.4474  2.4896
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## ect1     -0.33129    0.06319  -5.243 3.44e-07 ***
## ect2      0.09447    0.10950   0.863 0.389101
## constant  0.16837    0.05047   3.336 0.000982 ***
## y1.dl1   -0.22768    0.08080  -2.818 0.005234 **
## y2.dl1    0.14445    0.09129   1.582 0.114870
## y3.dl1    0.12347    0.10530   1.173 0.242135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6587 on 242 degrees of freedom
## Multiple R-squared:  0.1165, Adjusted R-squared:  0.0946
## F-statistic: 5.318 on 6 and 242 DF,  p-value: 3.57e-05
##
```

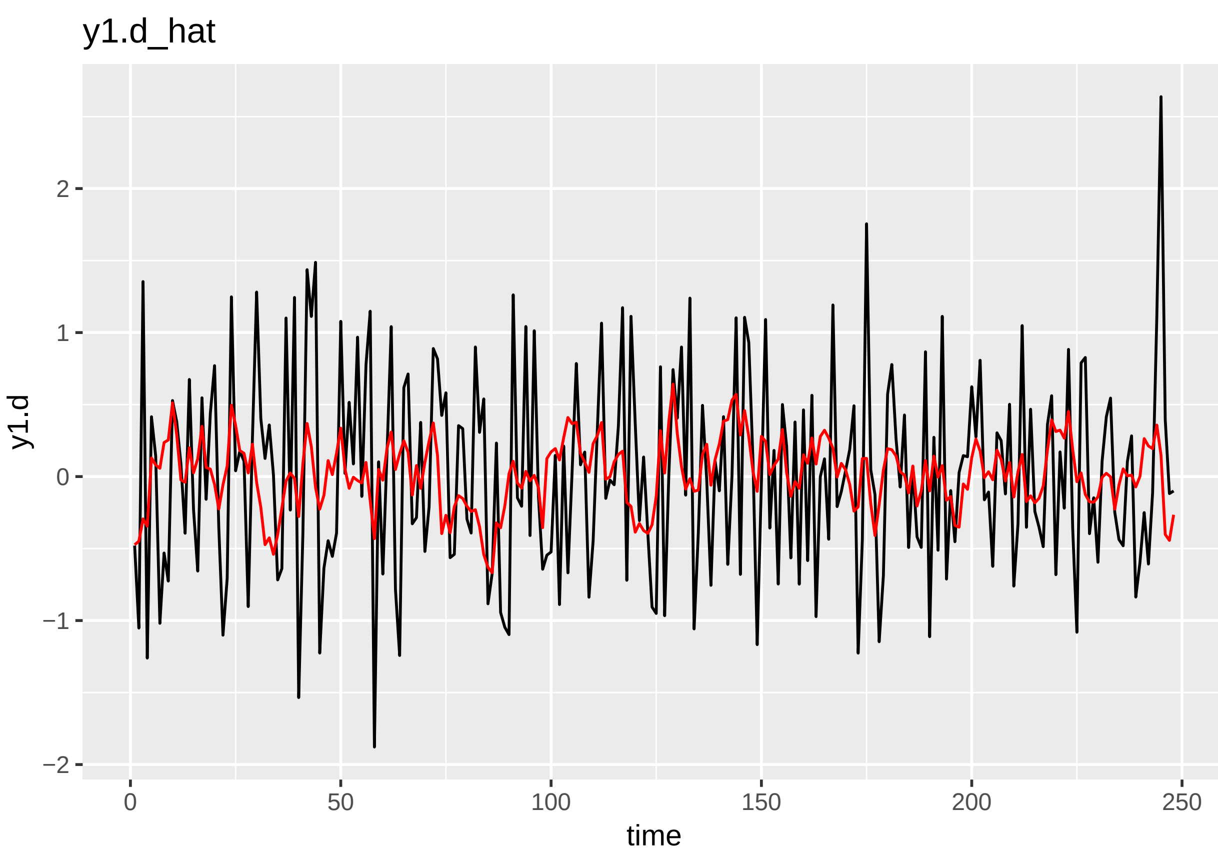
```
##
## Response y2.d :
##
## Call:
## lm(formula = y2.d ~ ect1 + ect2 + constant + y1.dl1 + y2.dl1 +
##     y3.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47003 -0.29237 -0.01239  0.33676  1.53785
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## ect1      0.06461    0.04723   1.368 0.172543
## ect2     -0.70938    0.08184  -8.668 6.41e-16 ***
## constant -0.02702    0.03772  -0.716 0.474462
## y1.dl1    0.02701    0.06039   0.447 0.655043
## y2.dl1   -0.71561    0.06823 -10.489  < 2e-16 ***
## y3.dl1   -0.29083    0.07870  -3.695 0.000271 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4923 on 242 degrees of freedom
## Multiple R-squared:  0.3326, Adjusted R-squared:  0.3161
## F-statistic:  20.1 on 6 and 242 DF,  p-value: < 2.2e-16
##
##
## Response y3.d :
##
## Call:
## lm(formula = y3.d ~ ect1 + ect2 + constant + y1.dl1 + y2.dl1 +
##     y3.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40655 -0.30401 -0.01374  0.32107  1.69270
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## ect1      0.012682   0.050764   0.250    0.803
## ect2     -0.009165   0.087966  -0.104    0.917
## constant  0.025255   0.040542   0.623    0.534
## y1.dl1    0.068158   0.064910   1.050    0.295
## y2.dl1    0.040487   0.073338   0.552    0.581
## y3.dl1   -0.075251   0.084594  -0.890    0.375
##
## Residual standard error: 0.5292 on 242 degrees of freedom
## Multiple R-squared:  0.01173,    Adjusted R-squared:  -0.01277
## F-statistic: 0.4788 on 6 and 242 DF,  p-value: 0.8239
```

Donde la beta del objeto vecm.r2 se obtiene de la siguiente manera:

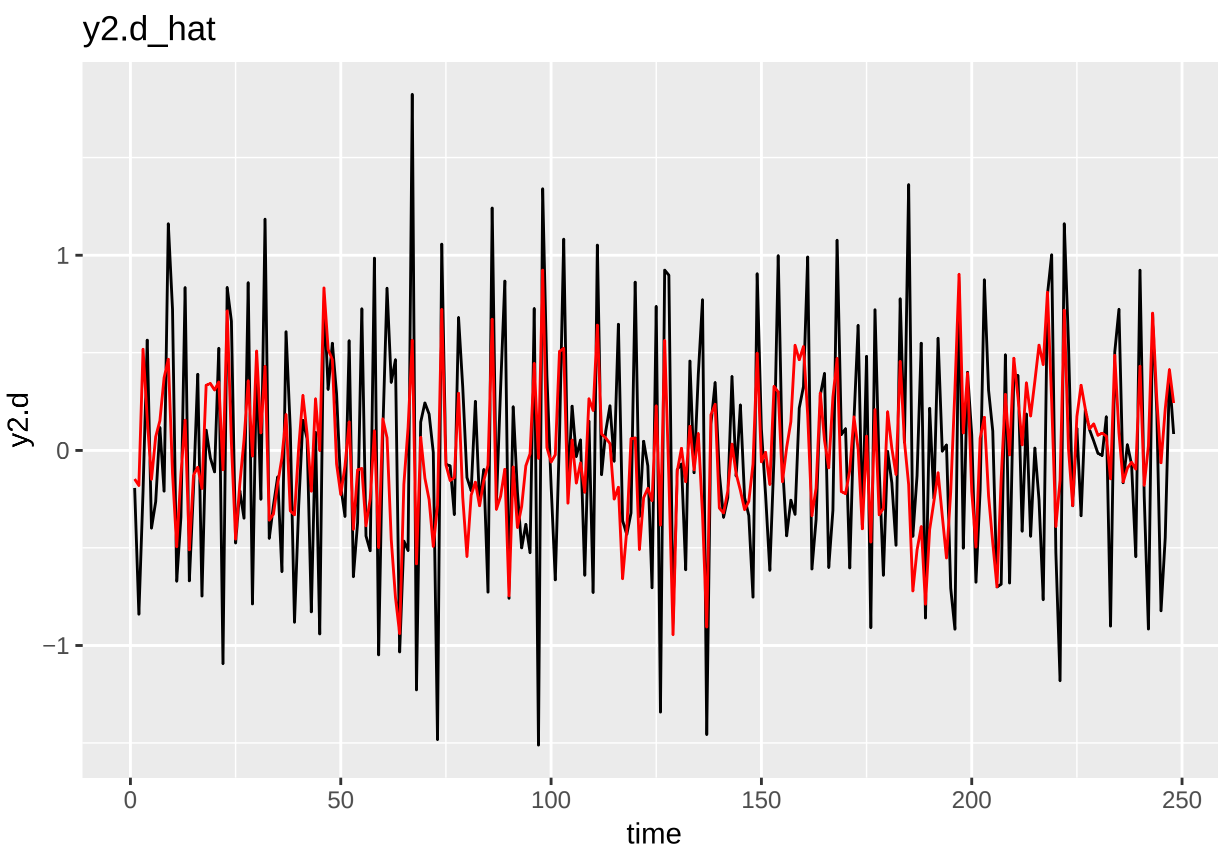
```
tmp_beta <- vecm@V[,1:2]

pre_s <- diag(2)
```

```r
st <- cbind(pre_s, c(0,0))


beta_c <- tmp_beta%*%solve(st%*%tmp_beta)
```

```r
# y1.d_hat


hat <- as.data.frame(vecm.r2$rlm$fitted.values[,1])
colnames(hat) <- c('y1.d_hat')

obs <- as.data.frame(vecm.r2$rlm$model$`z@Z0`[,1])
colnames(obs) <- c('y1.d')

ajuste <- cbind(obs,hat)


ajuste <- ajuste %>%
  mutate(time=row_number())


ggplot(ajuste)+
  geom_line(aes(time,y1.d))+
  geom_line(aes(time,y1.d_hat), color='red')+
  ggtitle('y1.d_hat')
```

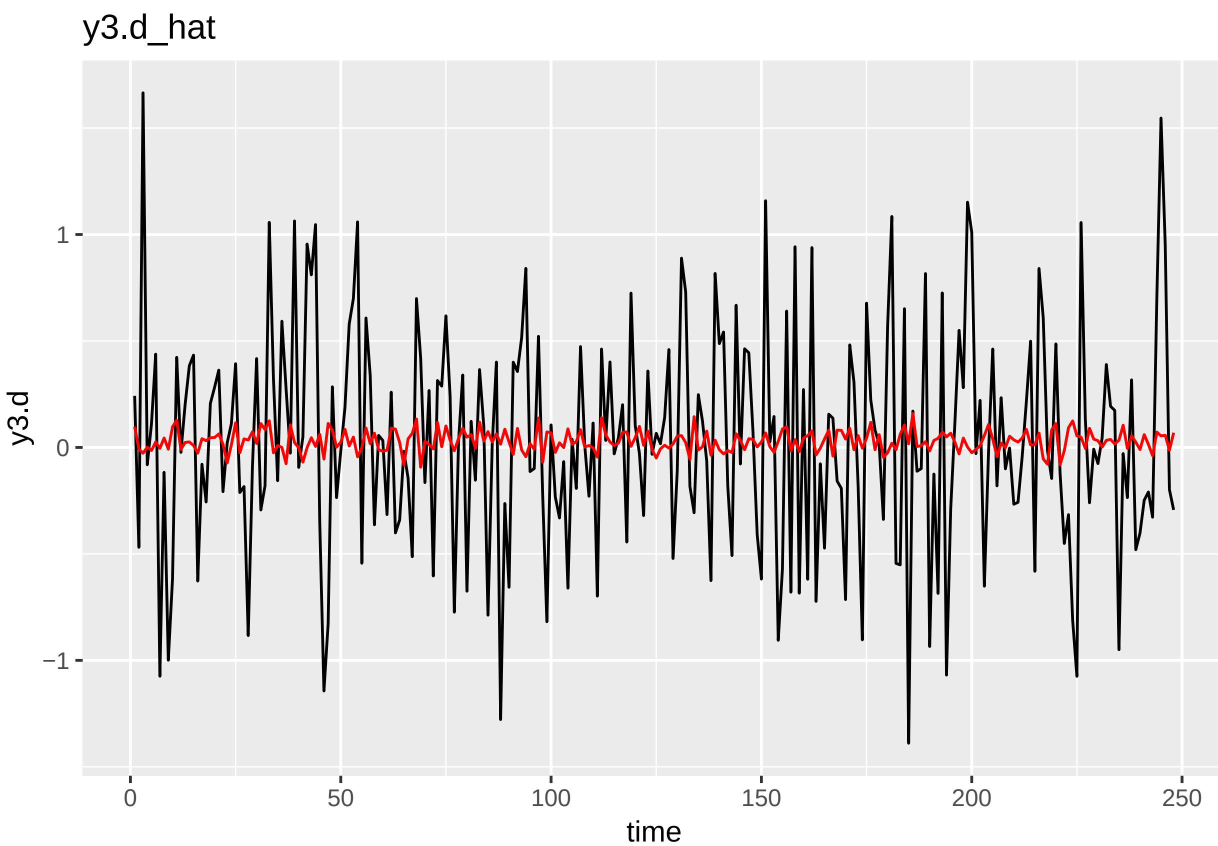


```r
# y2.d_hat


hat <- as.data.frame(vecm.r2$rlm$fitted.values[,2])
colnames(hat) <- c('y2.d_hat')

obs <- as.data.frame(vecm.r2$rlm$model$`z@Z0`[,2])
colnames(obs) <- c('y2.d')

ajuste <- cbind(obs,hat)


ajuste <- ajuste %>%
  mutate(time=row_number())


ggplot(ajuste)+
  geom_line(aes(time,y2.d))+
  geom_line(aes(time,y2.d_hat), color='red')+
  ggtitle('y2.d_hat')
```




```r
# y3.d_hat

hat <- as.data.frame(vecm.r2$rlm$fitted.values[,3])
colnames(hat) <- c('y3.d_hat')

obs <- as.data.frame(vecm.r2$rlm$model$`z@Z0`[,3])
```

```
colnames(obs) <- c('y3.d')

ajuste <- cbind(obs,hat)


ajuste <- ajuste %>%
  mutate(time=row_number())


ggplot(ajuste)+
  geom_line(aes(time,y3.d))+
  geom_line(aes(time,y3.d_hat), color='red')+
  ggtitle('y3.d_hat')
```



Analicemos los objetos output de un ca.jo

Z0: Object of class "matrix": The matrix of the differenced series.

```
#vecm
head(vecm@Z0)
```

```
##            y1.d       y2.d        y3.d
## [1,] -0.4794131 -0.1922064  0.24235792
## [2,] -1.0522604 -0.8403053 -0.46898616
## [3,]  1.3541820 -0.2094376  1.66536665
## [4,] -1.2606834  0.5650305 -0.08147273
## [5,]  0.4155807 -0.3992869  0.11022789
## [6,]  0.1428918 -0.2635745  0.43810541
```

```r
head(diff(vecm@x[,1]))[-1]
```

```
## [1] -0.4794131 -1.0522604  1.3541820 -1.2606834  0.4155807
```

Z1: Object of class “matrix”: The regressor matrix, except for the lagged variables in levels.

```r
head(vecm@Z1)
```

```
##      constant    y1.dl1     y2.dl1      y3.dl1
## [1,]        1 -1.3386618  0.8801562 -1.23346932
## [2,]        1 -0.4794131 -0.1922064  0.24235792
## [3,]        1 -1.0522604 -0.8403053 -0.46898616
## [4,]        1  1.3541820 -0.2094376  1.66536665
## [5,]        1 -1.2606834  0.5650305 -0.08147273
## [6,]        1  0.4155807 -0.3992869  0.11022789
```

ZK: Object of class “matrix”: The matrix of the lagged variables in levels.

```r
head(vecm@ZK)
```

```
##           y1.l2       y2.l2      y3.l2
## [1,]  2.2583477  0.19939792 -0.7101619
## [2,]  0.9196859  1.07955410 -1.9436313
## [3,]  0.4402729  0.88734767 -1.7012733
## [4,] -0.6119875  0.04704239 -2.1702595
## [5,]  0.7421945 -0.16239519 -0.5048928
## [6,] -0.5184889  0.40263534 -0.5863656
```

```r
head(vecm@x[,1])
```

```
## [1]  2.2583477  0.9196859  0.4402729 -0.6119875  0.7421945 -0.5184889
```

GAMMA: Object of class “matrix”: The coeffecient matrix of Z1.

```r
vecm@GAMMA
```

```
##         constant      y1.dl1      y2.dl1     y3.dl1
## y1.d  0.23923393 -0.22926059  0.14319319  0.1114252
## y2.d -0.05551169  0.02764891 -0.71510124 -0.2859867
## y3.d  0.11515655  0.06614925  0.03889071 -0.0905275
```

R0: Object of class “matrix”: The matrix of residuals from the regressions in differences.

```r
head(vecm@R0)
```

```
##         R0.y1.d     R0.y2.d     R0.y3.d
## [1,] -0.6904635 -0.08037954  0.16940277
## [2,] -1.0705514 -0.87294513 -0.44518760
## [3,]  1.3567145 -0.59827432  1.70442968
## [4,] -1.1485490  0.78386750 -0.07149617
## [5,]  0.2704663 -0.22409425  0.12573125
## [6,]  0.1881747 -0.37170924  0.40727514
```

Estos se obtienen de la siguiente manera para y1.d

```r
obs <- vecm@Z0[,1]
tmp <- lm(obs ~ vecm@Z1)
head(tmp$residuals)
```

```
##          1          2          3          4          5          6
## -0.6904635 -1.0705514  1.3567145 -1.1485490  0.2704663  0.1881747
```

RK: Object of class "matrix": The matrix of residuals from the regression in lagged levels.

```
head(vecm@RK)
```

```
##         RK.y1.l2  RK.y2.l2  RK.y3.l2
## [1,] -0.7880642 1.4228006 -3.990389
## [2,] -1.7167073 1.7503912 -4.438872
## [3,] -2.5056460 1.2466576 -4.578191
## [4,] -2.3680375 0.6996849 -4.002224
## [5,] -2.2307412 0.8807273 -3.096962
## [6,] -2.7678351 0.9845930 -3.265588
```

```
obs <- vecm@ZK[,1]
tmp <- lm(obs ~ vecm@Z1)
head(tmp$residuals)
```

```
##          1          2          3          4          5          6
## -0.7880642 -1.7167073 -2.5056460 -2.3680375 -2.2307412 -2.7678351
```

Codigo para obtener los p-values de la relacion de cointegración:

```
library("vars")
data("Canada")
summary(Canada)
```

```
##        e              prod             rw              U
##  Min.   :928.6   Min.   :401.3   Min.   :386.1   Min.   : 6.700
##  1st Qu.:935.4   1st Qu.:404.8   1st Qu.:423.9   1st Qu.: 7.782
##  Median :946.0   Median :406.5   Median :444.4   Median : 9.450
##  Mean   :944.3   Mean   :407.8   Mean   :440.8   Mean   : 9.321
##  3rd Qu.:950.0   3rd Qu.:410.7   3rd Qu.:461.1   3rd Qu.:10.607
##  Max.   :961.8   Max.   :418.0   Max.   :470.0   Max.   :12.770
```

```
###################################################
### VECM
###################################################
vecm.p3 <- summary(ca.jo(Canada, type = "trace", ecdet = "trend", K = 3, spec = "transitory"))
vecm.p2 <- summary(ca.jo(Canada, type = "trace", ecdet = "trend", K = 2, spec = "transitory"))


###################################################
### VECM r = 1
###################################################
vecm <- ca.jo(Canada[, c("rw", "prod", "e", "U")], type = "trace", ecdet = "trend", K = 3, spec = "tran
vecm.r1 <- cajorls(vecm, r = 1)
##
## Calculation of t-values for alpha and beta
##
alpha <- coef(vecm.r1$rlm)[1, ]
names(alpha) <- c("rw", "prod", "e", "U")
alpha
```

```
##           rw         prod            e            U
## -0.084814510 -0.011994081 -0.015606039 -0.008659911
```

```
beta <- vecm.r1$beta
beta
```

```
##                 ect1
## rw.l1     1.00000000
## prod.l1   0.54487553
## e.l1     -0.01299605
## U.l1      1.72657188
## trend.l1 -0.70918872
```

```
resids <- resid(vecm.r1$rlm)
N <- nrow(resids)
sigma <- crossprod(resids) / N
## t-stats for alpha (calculated by hand)
alpha.se <- sqrt(solve(crossprod(cbind(vecm@ZK %*% beta, vecm@Z1)))[1, 1] * diag(sigma))
names(alpha.se) <-  c("rw", "prod", "e", "U")
alpha.t <- alpha / alpha.se
alpha.t
```

```
##         rw       prod          e          U
## -5.7117416 -0.9186147 -2.1579440 -1.4868989
```

```
## Differ slightly from coef(summary(vecm.r1$rlm))
## due to degrees of freedom adjustment
coef(summary(vecm.r1$rlm))
```

```
## Response rw.d :
##              Estimate  Std. Error    t value     Pr(>|t|)
## ect1     -0.08481451  0.01586043 -5.3475545 1.032919e-06
## constant 55.46912512 10.27534117  5.3982758 8.457630e-07
## rw.dl1   -0.01208216  0.10704961 -0.1128651 9.104560e-01
## prod.dl1 -0.07449278  0.12959959 -0.5747917 5.672491e-01
## e.dl1    -0.63408419  0.32375145 -1.9585524 5.409412e-02
## U.dl1     0.06313697  0.39823872  0.1585405 8.744810e-01
## rw.dl2   -0.15738805  0.10657625 -1.4767647 1.441606e-01
## prod.dl2 -0.25194030  0.13324238 -1.8908420 6.272600e-02
## e.dl2     0.08119694  0.34085520  0.2382153 8.124002e-01
## U.dl2    -0.23000852  0.40783183 -0.5639788 5.745457e-01
##
## Response prod.d :
##               Estimate Std. Error     t value    Pr(>|t|)
## ect1     -0.011994081 0.01394591 -0.86004279 0.392660259
## constant  8.274808112 9.03500121  0.91586132 0.362839920
## rw.dl1    0.004706801 0.09412762  0.05000446 0.960259241
## prod.dl1  0.234441189 0.11395558  2.05730322 0.043331256
## e.dl1    -0.246543876 0.28467130 -0.86606509 0.389371704
## U.dl1    -0.979868038 0.35016719 -2.79828628 0.006608993
## rw.dl2   -0.190264297 0.09371139 -2.03032194 0.046071053
## prod.dl2 -0.029520300 0.11715865 -0.25196859 0.801793287
## e.dl2    -0.580472750 0.29971045 -1.93677850 0.056752354
## U.dl2    -0.128100838 0.35860231 -0.35722257 0.721984808
##
## Response e.d :
##               Estimate  Std. Error     t value     Pr(>|t|)
## ect1     -0.015606039 0.007724419 -2.02035108 4.712059e-02
## constant 10.331308141 5.004343822  2.06446809 4.262776e-02
## rw.dl1   -0.078491214 0.052135793 -1.50551490 1.366279e-01
## prod.dl1  0.200953060 0.063118190  3.18375830 2.158925e-03
```

```
## e.dl1     0.821557783 0.157674917  5.21045326 1.766876e-06
## U.dl1     0.003379404 0.193952049  0.01742391 9.861473e-01
## rw.dl2   -0.095834953 0.051905254 -1.84634398 6.901044e-02
## prod.dl2  0.048272523 0.064892318  0.74388655 4.593999e-01
## e.dl2    -0.459693071 0.166004862 -2.76915427 7.165319e-03
## U.dl2    -0.103414812 0.198624129 -0.52065584 6.042265e-01
##
## Response U.d :
##               Estimate  Std. Error     t value     Pr(>|t|)
## ect1     -0.008659911 0.006220787 -1.39209252 1.682397e-01
## constant  5.687831831 4.030200359  1.41130250 1.625226e-01
## rw.dl1    0.017262536 0.041987062  0.41113942 6.822087e-01
## prod.dl1 -0.138916466 0.050831629 -2.73287454 7.918064e-03
## e.dl1    -0.646846115 0.126981984 -5.09399914 2.775841e-06
## U.dl1    -0.191125426 0.156197425 -1.22361445 2.251431e-01
## rw.dl2    0.080354366 0.041801399  1.92228891 5.858215e-02
## prod.dl2 -0.002908953 0.052260406 -0.05566266 9.557669e-01
## e.dl2    -0.019741041 0.133690425 -0.14766234 8.830278e-01
## U.dl2    -0.262685288 0.159960039 -1.64219319 1.049722e-01
```

```
## t-stats for beta
beta.se <- sqrt(diag(kronecker(solve(crossprod(vecm@RK[, -1])),
                               solve(t(alpha) %*% solve(sigma) %*% alpha))))
beta.t <- c(NA, beta[-1] / beta.se)
names(beta.t) <- rownames(vecm.r1$beta)
beta.t
```

```
##       rw.l1     prod.l1        e.l1        U.l1    trend.l1
##          NA  0.90044324 -0.01917236  1.19328037 -2.56940604
```

Chapter 11

Cointegración (Análisis)

En esta sección haremos la prueba de cointegración multivaraida para las series no estacionarias mostradas en las secciones previas. Es decir, las series que entrarán al modelo VECM son:

- ln(PIB constante), $I(1)$
- Gasto en Educación, $I(1)$
- Gasto en salud, $I(1)$
- Recaudación Impositiva, $I(1)$

Las series utilizadas tienen una profundidad anual histórica desde 1991-2016.

Nos interesa conocer desde un punto de vista totalmente estadístico si existe algún tipo de relación de largo plazo entre la recaudación impositiva y el desarrollo de la nación en términos de indicadores básicos tales como el desempleo, gasto en salud y PIB a precios constantes.

Para ello, necesitamos estructurar la información para ser leída por la función `ca.jo`.

Definimos las series a utilizar, dado que todas las series son integradas de orden 1 $I(1)$ no hay necesidad de diferenciar las series, pues buscamos una o más combinaciones lineales de ellas que sean $I(0)$.

```
ln_pib_cte <-  series_db$log_GDP_constante
edu <- series_db$Gasto_Educacion_PorcGDP
salud <- series_db$Gasto_Salud_PorcGDP
rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP
```

Posteriormente, tomaremos todas las series que ya son integradas de orden 1 $I(1)$ y eliminaremos el primer registro para que todas las series tengan la misma longitud

```
#ln_pib_cte <-  series_db$log_GDP_constante[-1]
#edu <- series_db$Gasto_Educacion_PorcGDP[-1]
#salud <- series_db$Gasto_Salud_PorcGDP[-1]
#rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP[-1]


model.data <- cbind(ln_pib_cte,edu,salud,rec_imp)
```

Como un modelo preliminar utilizaremos la siguiente especificiación:

$\nabla y_t = \Gamma_1 \nabla y_{t-1} + \cdots + \Gamma_{k-1} \nabla y_{t-k+1} + \Pi y_{t-1} + \mu + \epsilon_t$

donde el vector y_t contiene a los elementos (ln__pib__cte,edu,salud,rec__imp)' y el vector μ es un vector de constantes. El proceso del error de 4 dimensiones ϵ_t se asume i.i.d $N(0, \Sigma)$ para $t = 1, \cdots, T$. Cabe mencionar que estaremos utilizando la especificación que mide los efectos transitorios para el VECM. Es decir, $\Gamma_i = -(\Pi_{i+1}, \cdots, \Pi_k)$ con $i = 1, \cdots, k-1$ y $\Pi = -(I - \Pi_1 - \cdots - \Pi_k)$.

11.1 Determinación del Rango de Cointegración

Debido a que las inferencias sobre el espacio de cointegración generado por las series dependen de si existen o no tendencias lineales en los datos, se puede argumentar mediante un análisis visual y razonamiento lógico que la series como el GDP, desempleo y salud, contienen una tendencia lineal y es completamente lógica, por lo que el vector μ puede ser estimado sin imponer ningun tipo de restricción sobre él.

La hipótesis $H_1(r) : \Pi = \alpha\beta'$ (i.e. Π es de rango reducido) es probada utilizando el estadístico de la traza y del máximo eigenvalor.

```
M1 <- ca.jo(model.data,spec='transitory', type='eigen',K=2)
#summary(M2)



M1.trace <- summary(ca.jo(model.data,spec='transitory',type='trace',K=2))
cbind(test=M1.trace@teststat,M1.trace@cval)
```

```
##                test 10pct  5pct  1pct
## r <= 3 |  0.2680195  6.50  8.18 11.65
## r <= 2 |  8.4860040 15.66 17.95 23.52
## r <= 1 | 26.7400746 28.71 31.52 37.22
## r = 0  | 61.7048157 45.23 48.28 55.43
```

```
M1.eigen <- summary(ca.jo(model.data,spec='transitory',type='eigen',K=2))

cbind(test=M1.eigen@teststat,M1.eigen@cval)
```

```
##                test 10pct  5pct  1pct
## r <= 3 |  0.2680195  6.50  8.18 11.65
## r <= 2 |  8.2179845 12.91 14.90 19.19
## r <= 1 | 18.2540706 18.90 21.07 25.75
## r = 0  | 34.9647411 24.78 27.14 32.14
```

En los dos outputs anteriores, los resultados de las dos pruebas son mostrados. Si consideramos por un lado el estadístico del máximo eigenvalor ($H_0 : rank(\Pi) = r$ vs $H_a : rank(\Pi) = r + 1$), la hipótesis de no cointegración puede ser rechazada a un nivel de 5% de confianza, mientras que la hipótesis de que existen 1 relaciones de cointegración vs 2 no puede ser rechazada, por lo tanto, existen 1 relaciones de cointegración. Por otro lado, el estadístico de la traza ($H_0 : rank(\Pi) \leq r$) indica un espacio de cointegración de $r = 1$ también. Consideremos también si esta conclsión podría ser errónea debido a la posible cercanía de los eigenvalores:

```
M1@lambda
```

```
## [1] 0.76703434 0.53260767 0.28994725 0.01110535
```

Como es posible observar, dos de los cuatro eigen valores están relativamente cerca entre ellos. Para determinar correctamente el orden de integración, Johansen y Juselius investigaron sobe las matrices $\hat{\beta}$ y $\hat{\alpha}$ así como las relaciones de cointegración estimadas $\hat{\beta}_i{}' y_{t-1}$ y aquellas relaciones corregidas por las influencias de corto plazo $\hat{\beta}_i{}' R_{1t}$. Para ello, proponen los siguientes pasos a seguir:

1) Estimar el modelo de corrección de error
2) Determinar el rango de Π

3) utilizar los r vectores de cointegración más significativos para formar β'
4) Seleccionar α tal que $\Pi = \alpha\beta'$

Observemos la matriz de eigen vectores beta

```
beta.matrix <- M1@V
alpha.matrix <- M1@PI%*%solve(t(beta.matrix))
beta.matrix
```

```
##               ln_pib_cte.l1      edu.l1   salud.l1 rec_imp.l1
## ln_pib_cte.l1     1.0000000  1.00000000  1.0000000  1.0000000
## edu.l1           -0.2734779 -0.12722367 -0.8329062 -0.4709689
## salud.l1         -0.3187240 -0.14485981  1.3555847 -0.7250595
## rec_imp.l1        0.2359534 -0.02050015 -0.2056044 -0.4166424
```

En la matriz β se pueden observar los vectores de cointegración, de tal manera que la primera columna corresponde al vector de cointegración asociado con el eigenvalor más grande. Debe notarse que los vectores de cointegración están todos normalizados a la primera variable, por lo que la matriz de velocidad de ajuste $\hat{\alpha}$ se ajusta de acuerdo a esta matriz normalizada:

```
alpha.matrix
```

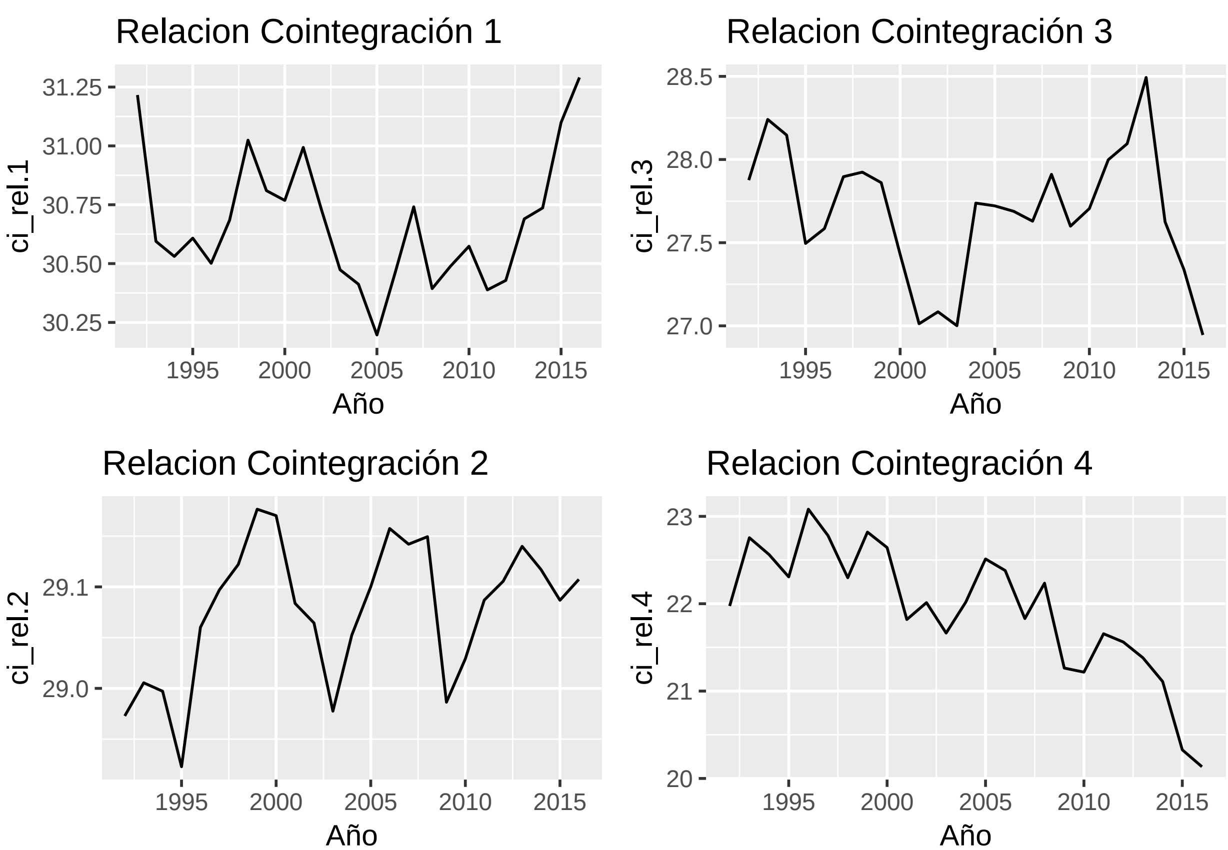
```
##              ln_pib_cte.l1     edu.l1    salud.l1   rec_imp.l1
## ln_pib_cte.d    0.02575836 -0.1945384 -0.01292212 -0.004630415
## edu.d           0.81182331  1.8215325  0.14885276  0.008440650
## salud.d        -0.10602593  0.7318614 -0.18202612 -0.004282401
## rec_imp.d      -2.42505459  8.1784032  0.50872868 -0.058362273
```

Al observar la matriz $\hat{\alpha}$ se puede concluir que las velocidades de ajuste de las relaciones de cointegración parecen ser distintas de cero, esto significa que las relaciones de cointegración juegan un papel importante en la dinámica de corto plazo de cada una de las series, aunque aún falta validar que todas las relaciones de cointegración son estadísticamente significativas en cada serie.

Además podemos analizar la relación de cointegración de manera visual:

```
ci.1 <- ts((M1@x%*%beta.matrix)[-c(1),],
           start = 1992,
           end = 2016,
           frequency = 1)

ci.rel <- as.data.frame(ci.1) %>%
            mutate(date=time(ci.1))

ci.rel <- ci.rel[,c(5,1:4)]
colnames(ci.rel) <- c('Año','ci_rel.1','ci_rel.2','ci_rel.3','ci_rel.4')

ci.rel.df <- as.data.frame(ci.rel)
ci.rel.df$Año <- as.Date(ci.rel.df$Año)

p1 <-  ggplot(ci.rel.df[,c(1,2)],
             aes(x=Año,y=ci_rel.1))+
             geom_line()+
             ggtitle('Relacion Cointegración 1')


p2 <-  ggplot(ci.rel.df[,c(1,3)],
             aes(x=Año,y=ci_rel.2))+
             geom_line()+
```

```
            ggtitle('Relacion Cointegración 2')

p3 <-  ggplot(ci.rel.df[,c(1,4)],
            aes(x=Año,y=ci_rel.3))+
            geom_line()+
            ggtitle('Relacion Cointegración 3')

p4 <-  ggplot(ci.rel.df[,c(1,5)],
            aes(x=Año,y=ci_rel.4))+
            geom_line()+
            ggtitle('Relacion Cointegración 4')



multiplot(p1,p2,p3,p4,cols=2)
```

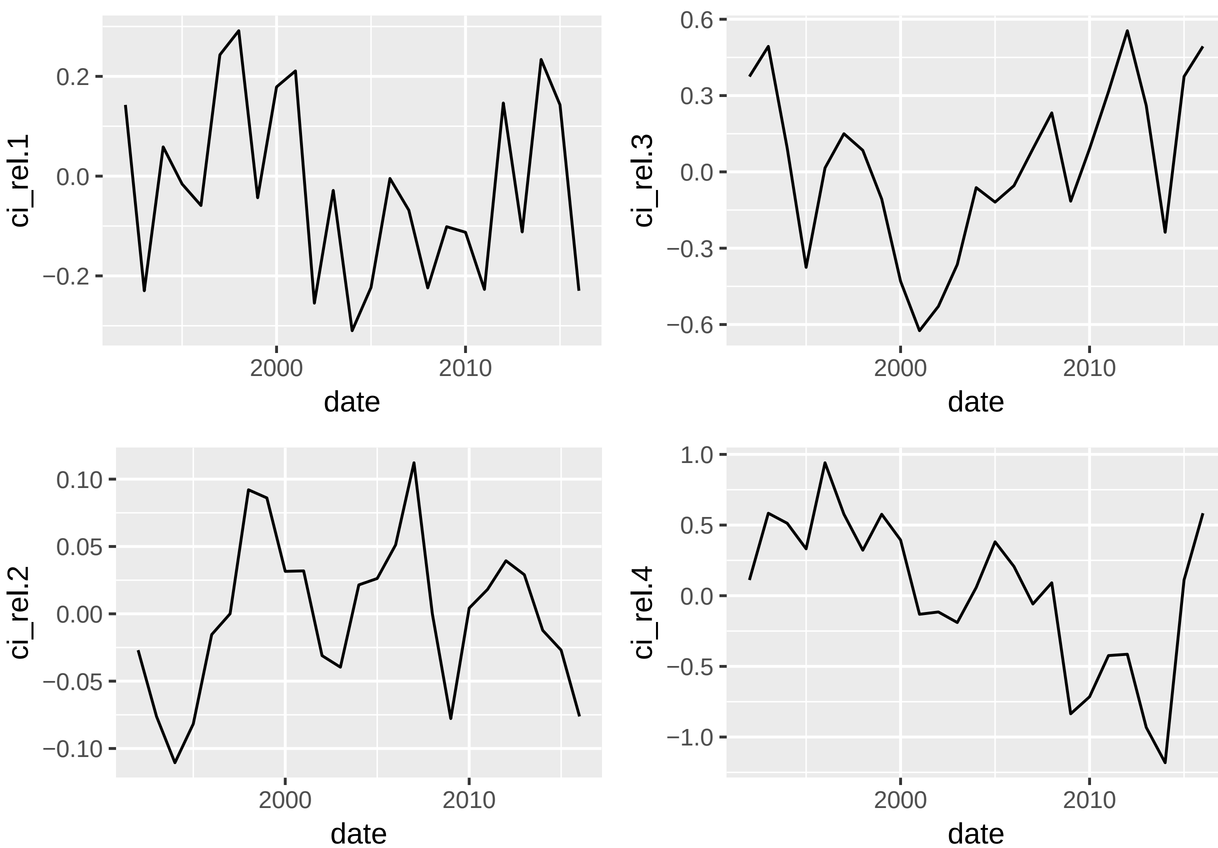


Debido a que el rango fue de $r = 1$ la primera relación de cointegración debería comportarse como proceso estacionario. Sin embargo, debido a influencias de corto plazo que interactuan en el proceso de ajuste del modelo las gráficas anteriores podrían verse afectadas. Por esta razón, también se analizan las trayectorias del ajuste $\hat{\beta}' R_{1t}$ donde R_{1t} son los residuales obtenidos al hacer la regresión de y_{t-1} explicado por ∇y_{t-1} que toman en cuenta la dinámica de corto plazo.

```
ci_sr.1 <- ts((M1@RK%*%beta.matrix)[-c(1),],
           start = 1992,
           end = 2016,
           frequency = 1)
```

```
ci_sr.rel <- as.data.frame(ci_sr.1) %>%
           mutate(date=time(ci_sr.1))

ci_sr.rel <- ci_sr.rel[,c(5,1:4)]
colnames(ci_sr.rel) <- c('date','ci_rel.1','ci_rel.2','ci_rel.3','ci_rel.4')

p1 <- ggplot(as.data.frame(ci_sr.rel[,c(1,2)]),
            aes(date,ci_rel.1))+
            geom_line()

p2 <- ggplot(as.data.frame(ci_sr.rel[,c(1,3)]),
            aes(date,ci_rel.2))+
            geom_line()

p3 <- ggplot(as.data.frame(ci_sr.rel[,c(1,4)]),
            aes(date,ci_rel.3))+
            geom_line()
p4 <- ggplot(as.data.frame(ci_sr.rel[,c(1,5)]),
            aes(date,ci_rel.4))+
            geom_line()

multiplot(p1,p2,p3,p4,cols=2)
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Basado en los resultados de las pruebas, los elementos de la matriz $\hat{\alpha}$ y la forma de las trayectorias de las relaciones de cointegración se puede conluir que existe únicamente 1 relación de cointegración.

Chapter 12

Estimación del modelo

El siguiente paso consiste en observar los valores estimados para cada relación de cointegración y determinar si todas ellas influyen en el comportamiento de corto plazo de cada una de las series en el análisis.

Johansen [1995] propone restringir β' de tal manera que la primera parte de la matriz sea una matriz identidad. Es decir, $\beta' = [I_r : \beta_1']$ donde β_1' tiene dimensión $((k-r) \times r)$. Esto se obtiene al normalizar el espacio de cointegración de la siguiente manera:

$$\beta_c = \beta(S'\beta)^{-1}$$

donde $S' = (I_r, 0)$.

```
#traemos las series
ln_pib_cte <-  series_db$log_GDP_constante
edu <- series_db$Gasto_Educacion_PorcGDP
salud <- series_db$Gasto_Salud_PorcGDP
rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP

# generamos la matriz X
model.data <- cbind(ln_pib_cte,edu,salud,rec_imp)

# estimamos el rango de cointegracion de PI con ca.jo
M1 <- ca.jo(model.data,spec='transitory', type='eigen',K=2)
```

12.1 Modelo 1: VECM con K=2 (lags), y r=1 (relaciones de cointegración)

Desde los objetos obtenidos por el ajuste del modelo, se puede obtener β_c de la siguiente manera

```
k_aux <- 4
rel_coint_aux <- 1

tmp_beta <-  M1@V[,1:rel_coint_aux]
st <- cbind(diag(rel_coint_aux),matrix(0,rel_coint_aux,(k_aux-rel_coint_aux)))
beta_c <- tmp_beta%*%solve(st%*%tmp_beta)
beta_c
```

```
##            [,1]
## [1,]  1.0000000
## [2,] -0.2734779
## [3,] -0.3187240
## [4,]  0.2359534
```

De tal manera que los vectores de cointegración tienen una interpretación mucho más sencilla.

Este resultado puede ser validado con lo que se obtiene a partir de la función cajorls

```
M1.jorls <- cajorls(M1, r = rel_coint_aux)
M1.jorls$beta
```

```
##                      ect1
## ln_pib_cte.l1  1.0000000
## edu.l1        -0.2734779
## salud.l1      -0.3187240
## rec_imp.l1     0.2359534
```

Ahora bien, Una vez que se han definido las relaciones de cointegración restringida veamos cómo es el ajuste del modelo a partir de estas definiciones:

```
summary(M1.jorls$rlm)
```

```
## Response ln_pib_cte.d :
##
## Call:
## lm(formula = ln_pib_cte.d ~ ect1 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.078343 -0.009848  0.003730  0.015520  0.040400
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1            0.025758   0.033287   0.774    0.449
## constant       -0.751983   1.017088  -0.739    0.469
## ln_pib_cte.dl1 -0.248623   0.247119  -1.006    0.328
## edu.dl1        -0.030300   0.028502  -1.063    0.302
## salud.dl1      -0.060255   0.050947  -1.183    0.252
## rec_imp.dl1    -0.004260   0.008287  -0.514    0.613
##
## Residual standard error: 0.03313 on 18 degrees of freedom
## Multiple R-squared:  0.4807, Adjusted R-squared:  0.3077
## F-statistic: 2.777 on 6 and 18 DF,  p-value: 0.04316
##
##
## Response edu.d :
##
## Call:
## lm(formula = edu.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34754 -0.12005 -0.01668  0.11324  0.35722
```

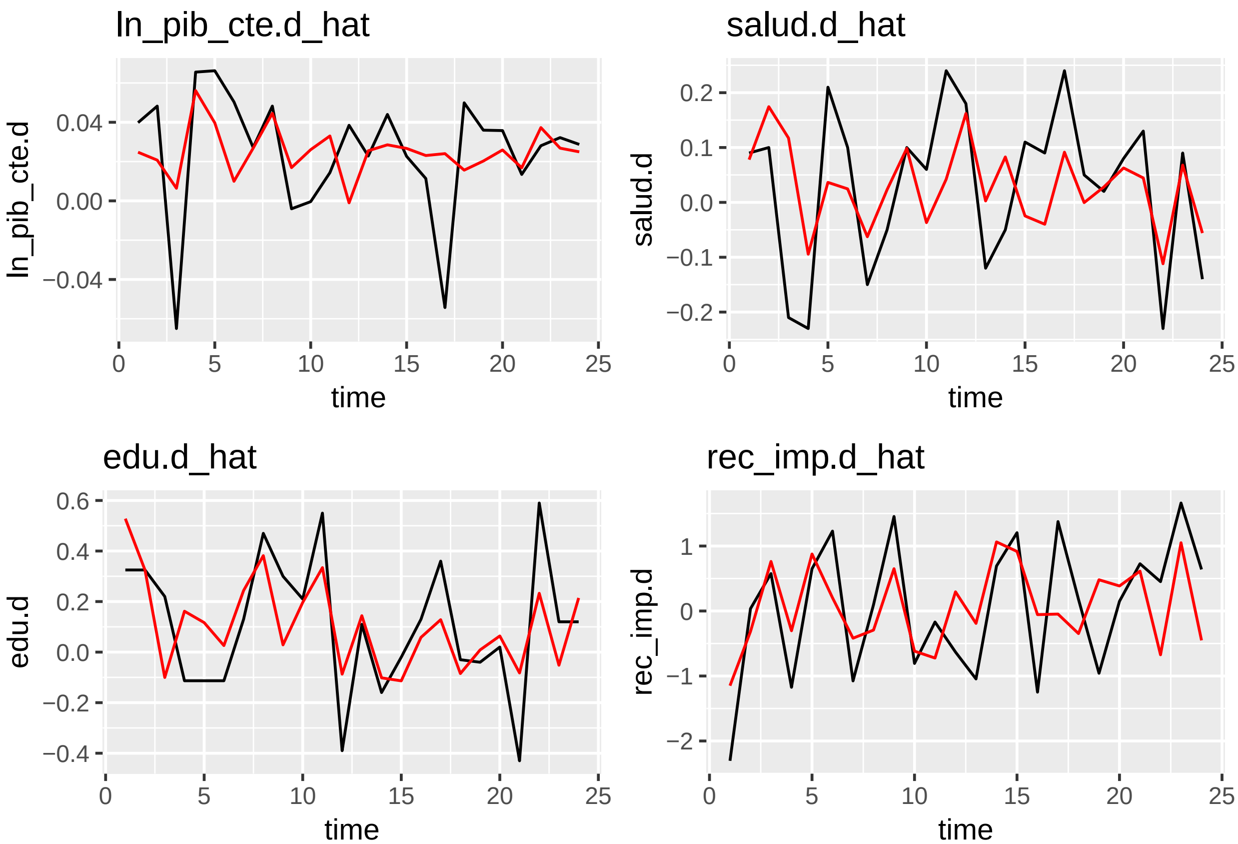
```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             0.81182    0.22350   3.632  0.00191 **
## constant       -24.67609    6.82904  -3.613  0.00199 **
## ln_pib_cte.dl1  -2.43462    1.65923  -1.467  0.15954
## edu.dl1         -0.29248    0.19137  -1.528  0.14380
## salud.dl1        0.08351    0.34207   0.244  0.80990
## rec_imp.dl1     -0.15076    0.05564  -2.710  0.01436 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2225 on 18 degrees of freedom
## Multiple R-squared:  0.5246, Adjusted R-squared:  0.3662
## F-statistic: 3.311 on 6 and 18 DF,  p-value: 0.02245
##
##
## Response salud.d :
##
## Call:
## lm(formula = salud.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32725 -0.08489  0.01468  0.08836  0.19771
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## ect1           -0.10603    0.14013  -0.757   0.4591
## constant        3.22199    4.28164   0.753   0.4615
## ln_pib_cte.dl1  1.04846    1.04030   1.008   0.3269
## edu.dl1         0.20965    0.11998   1.747   0.0976 .
## salud.dl1       0.14725    0.21447   0.687   0.5011
## rec_imp.dl1    -0.03174    0.03489  -0.910   0.3750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1395 on 18 degrees of freedom
## Multiple R-squared:  0.2986, Adjusted R-squared:  0.06485
## F-statistic: 1.277 on 6 and 18 DF,  p-value: 0.3163
##
##
## Response rec_imp.d :
##
## Call:
## lm(formula = rec_imp.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43866 -0.70667 -0.03661  0.56896  1.42373
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             -2.4251     0.9305  -2.606   0.0179 *
## constant         73.9915    28.4304   2.603   0.0180 *
## ln_pib_cte.dl1   14.0049     6.9076   2.027   0.0577 .
## edu.dl1           0.9886     0.7967   1.241   0.2306
## salud.dl1        -2.0040     1.4241  -1.407   0.1764
## rec_imp.dl1       0.3522     0.2316   1.520   0.1458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9262 on 18 degrees of freedom
## Multiple R-squared:  0.3743, Adjusted R-squared:  0.1657
## F-statistic: 1.794 on 6 and 18 DF,  p-value: 0.1569
```

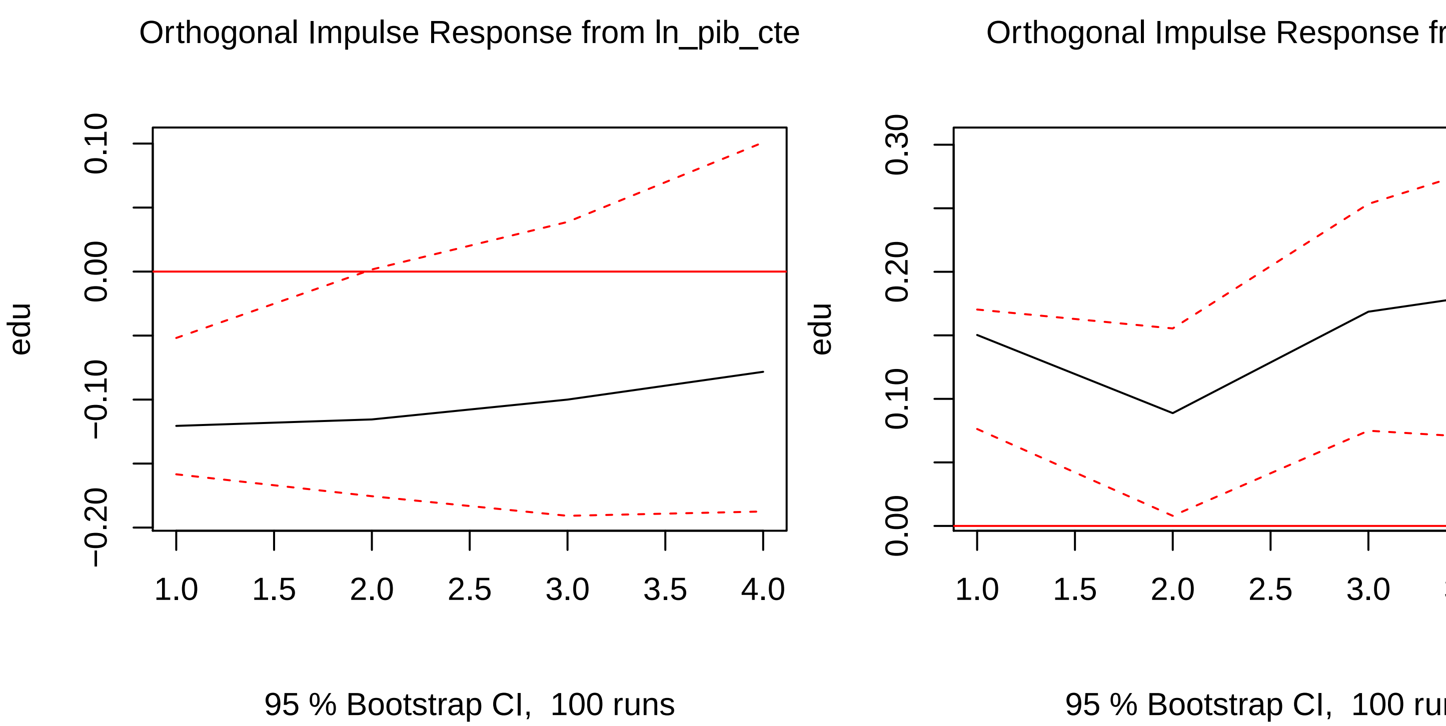
```r
mysum <- summary(M1.jorls$rlm)
screenreg(list(mysum[[1]], mysum[[2]],mysum[[3]],mysum[[4]]))
```

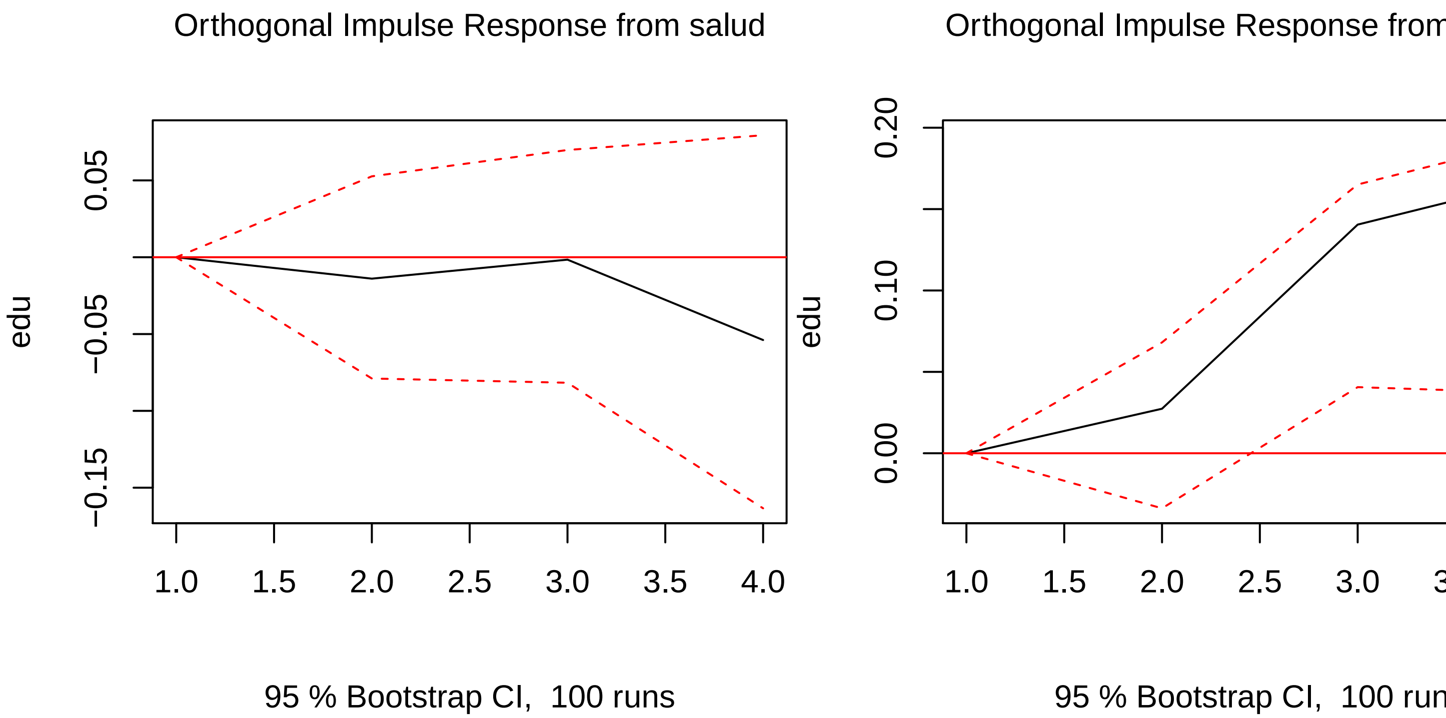
```
##
## =====================================================
##                  Model 1  Model 2    Model 3  Model 4
## -----------------------------------------------------
## ect1              0.03      0.81 **  -0.11     -2.43 *
##                  (0.03)    (0.22)    (0.14)    (0.93)
## constant         -0.75    -24.68 **   3.22     73.99 *
##                  (1.02)    (6.83)    (4.28)   (28.43)
## ln_pib_cte.dl1   -0.25     -2.43      1.05     14.00
##                  (0.25)    (1.66)    (1.04)    (6.91)
## edu.dl1          -0.03     -0.29      0.21      0.99
##                  (0.03)    (0.19)    (0.12)    (0.80)
## salud.dl1        -0.06      0.08      0.15     -2.00
##                  (0.05)    (0.34)    (0.21)    (1.42)
## rec_imp.dl1      -0.00     -0.15 *   -0.03      0.35
##                  (0.01)    (0.06)    (0.03)    (0.23)
## -----------------------------------------------------
## R^2               0.48      0.52      0.30      0.37
## Adj. R^2          0.31      0.37      0.06      0.17
## Num. obs.        24        24        24        24
## RMSE              0.03      0.22      0.14      0.93
## =====================================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

Finalmente obsevemos nuestro modelo ajustado vs los datos recopilados:

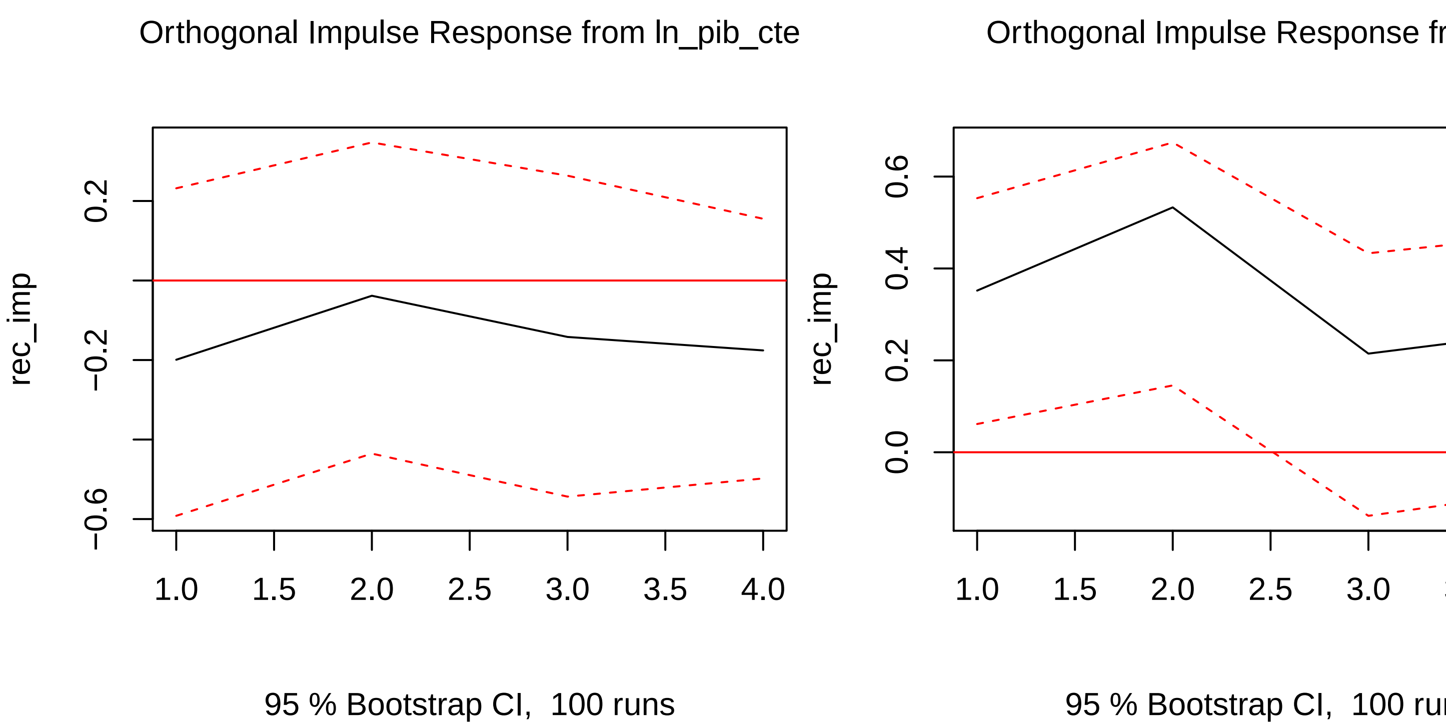



```r
plot(irf(vec2var(M1,r=rel_coint_aux),n.ahead=3,response='edu',boot=TRUE))
```






```r
plot(irf(vec2var(M1,r=rel_coint_aux),n.ahead=3,response='rec_imp',boot=TRUE))
```




```
BIC(vec2var(M1,r=rel_coint_aux))
```

```
## [1] 4.412841
```

Interpretar la relación de cointegración como ceteris paribus de gastos vs ceteris paribus de income (pib y recauación).

Esta relación de cointegración podemos analizar con dos bandas de desviación estándar y despues ver en què punto la relación queda fuera de las bandas y ver los datos históricos para ver qué hicieron para que la relación de cointegración regresara al equilibrio

Tenemos que explicar que el PIB es un alma libre, porque realmente las series que podemos controlar son la recaudación y los gastos (salud y educaciòn). El producto interno bruto no puede ser impactado por las series de gasto en salud, educación ni recaudación impositiva, pero si al revés.

Explicar que la serie de salud no tiene ningún efecto ante la ruptura del equilibrio. Esto hace sentido porque rara vez se toma una medida política incrementando el gasto en salud para corregir el pib o la recaudación.. Hace sentido que no tenga efectos

En donde sí que tenemos que meterle a la interpretación es en el caso de la educación, pues el signo de alpha es positivo, esto quiere decir que si suponemos que existe un efecto positivo de la relación de equilibiro (que el pib y la recaudación aumentan respecto al gasto) entonces veremos una corrección en el corto plazo de la educación de manera positiva, aunque la velocidad de ajuste es relativamente lenta. Por otro lado si la relación de cointegración se rompa negativamente, entonces veríamos una contracción de la eduación

La recaudación impositiva también tiene la relación de cointegración significativa en su ecuación. Esta tiene una alpha negativa, lo cual quiere decir que si el Pib crece bastante, veremos reflejado relativamente rápido una reducción en la recaudación impositiva. Por otro lado, si el gasto es el que provoca la salida del equilibrio de la relación de cointegración se verá casi inmediato un incremento en la recaudación impositiva para contrarrestar el efecto. Algo muy utilizado en el gobierno mexicano y en casi todos los gobiernos.

Debemos de ver esto representado en las funciones de impulso respuesta. Aunque debemos investigar si el efecto aleatorio es positivo.

El profesor propone documentar tambien el escenario 4, pues las relaciones de cointegración sobre educación son muy interesantes, en particular deberíamos entrar a detalle con la relación número dos de cointegración, pues es el efecto del gasto en educación en fucnión del gasto en salud y la recaudación. Es decir como se ve la recaudación vs el gasto del gobierno. Esto está muy interesante aunque deberíamos ser muy cuidadosos con la estimación del modelo pues se tienen demasiados paràmetros estiamdos y muy pocos datos. Esto nos podría generar un sobre ajuste.

Finalmente, el profe propone que pongamos una tablita con los AIC por ecuación del primer modelo vs el último modelo porque así podremos darnos cuenta del ajuste real de los datos. También sugiere sacar a mano el BIC pues no sabemos cómo toma los datos útiles la función de BIC sobre un moelo VAR.

12.2 Modelo 2: VECM con K=2 (lags), y r=2 (relaciones de cointegración)

```
#traemos las series
ln_pib_cte <-  series_db$log_GDP_constante
edu <- series_db$Gasto_Educacion_PorcGDP
salud <- series_db$Gasto_Salud_PorcGDP
rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP

# generamos la matriz X
model.data <- cbind(ln_pib_cte,edu,salud,rec_imp)

# estimamos el rango de cointegracion de PI con ca.jo
M1 <- ca.jo(model.data,spec='transitory', type='eigen',K=2)
```

Desde los objetos obtenidos por el ajuste del modelo, se puede obtener β_c de la siguiente manera

```
k_aux <- 4
rel_coint_aux <- 2

tmp_beta <-  M1@V[,1:rel_coint_aux]
st <- cbind(diag(rel_coint_aux),matrix(0,rel_coint_aux,(k_aux-rel_coint_aux)))
beta_c <- tmp_beta%*%solve(st%*%tmp_beta)
beta_c
```

```
##                         [,1]      [,2]
## ln_pib_cte.l1  1.000000e+00  0.000000
## edu.l1        -2.775558e-17  1.000000
## salud.l1       6.381212e-03  1.188781
## rec_imp.l1    -2.435841e-01 -1.753478
```

De tal manera que los vectores de cointegración tienen una interpretación mucho más sencilla.

Este resultado puede ser validado con lo que se obtiene a partir de la función cajorls

```
M1.jorls <- cajorls(M1, r = rel_coint_aux)
M1.jorls$beta
```

```
##                         ect1      ect2
## ln_pib_cte.l1  1.000000e+00  0.000000
## edu.l1        -2.775558e-17  1.000000
## salud.l1       6.381212e-03  1.188781
## rec_imp.l1    -2.435841e-01 -1.753478
```

Ahora bien, Una vez que se han definido las relaciones de cointegración restringida veamos cómo es el ajuste del modelo a partir de estas definiciones:

```
summary(M1.jorls$rlm)
```

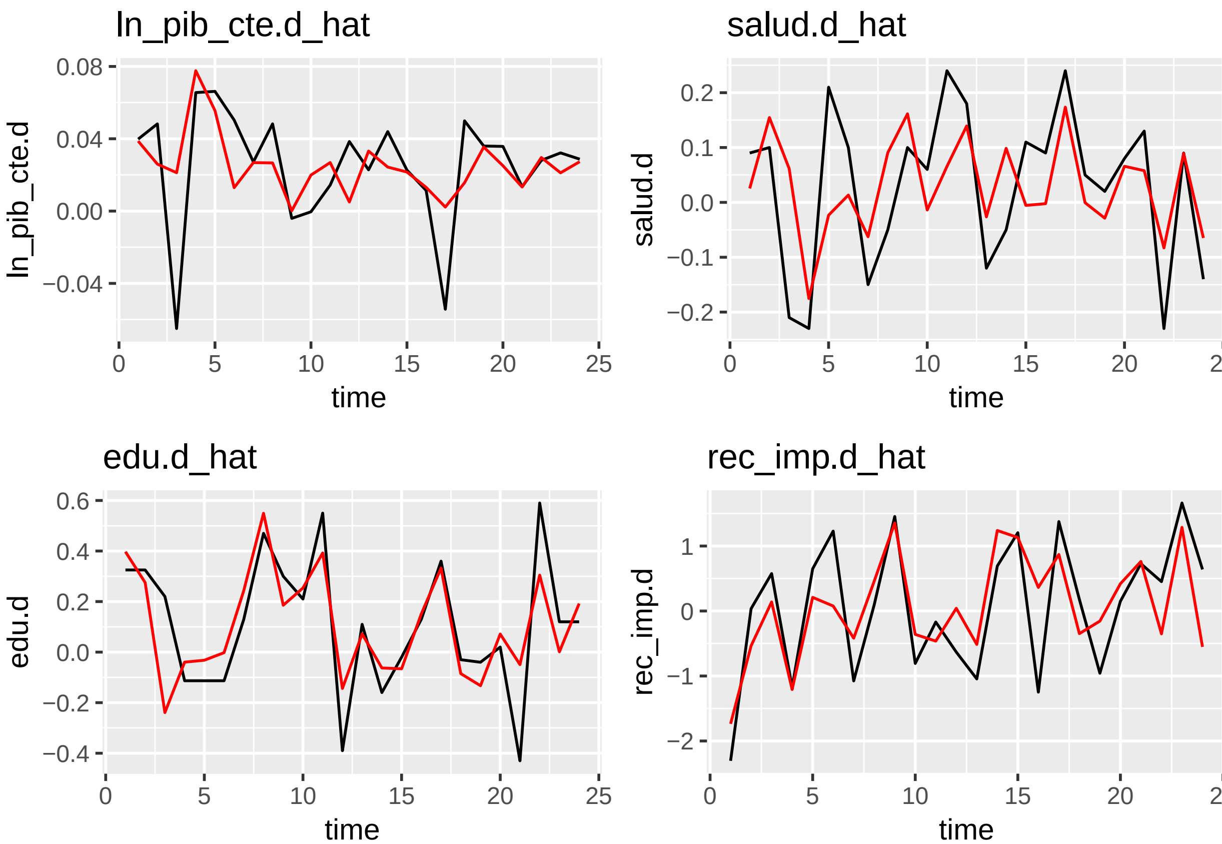
```
## Response ln_pib_cte.d :
##
## Call:
## lm(formula = ln_pib_cte.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.086214 -0.005774  0.000822  0.013168  0.037405
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1           -0.168780   0.118160  -1.428    0.171
## ect2            0.017706   0.016872   1.049    0.309
## constant        4.902612   3.447436   1.422    0.173
## ln_pib_cte.dl1 -0.084688   0.253742  -0.334    0.743
## edu.dl1        -0.043199   0.028125  -1.536    0.143
## salud.dl1      -0.085585   0.050647  -1.690    0.109
## rec_imp.dl1    -0.001347   0.008060  -0.167    0.869
##
## Residual standard error: 0.0315 on 17 degrees of freedom
## Multiple R-squared:  0.5569, Adjusted R-squared:  0.3744
## F-statistic: 3.052 on 7 and 17 DF,  p-value: 0.02846
##
##
## Response edu.d :
##
## Call:
## lm(formula = edu.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38058 -0.07983 -0.03273  0.06450  0.45910
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             2.63336    0.72614   3.626 0.002085 **
## ect2            -0.45376    0.10368  -4.376 0.000412 ***
## constant       -77.62209   21.18595  -3.664 0.001923 **
## ln_pib_cte.dl1  -3.96960    1.55935  -2.546 0.020895 *
## edu.dl1         -0.17170    0.17284  -0.993 0.334450
## salud.dl1        0.32067    0.31125   1.030 0.317311
## rec_imp.dl1     -0.17804    0.04953  -3.594 0.002235 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1936 on 17 degrees of freedom
## Multiple R-squared:  0.6601, Adjusted R-squared:  0.5202
## F-statistic: 4.717 on 7 and 17 DF,  p-value: 0.004238
```

```
##
##
## Response salud.d :
##
## Call:
## lm(formula = salud.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27149 -0.07816  0.02743  0.07264  0.23336
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             0.62584    0.50595   1.237   0.2329
## ect2            -0.06411    0.07224  -0.887   0.3872
## constant       -18.05083   14.76156  -1.223   0.2381
## ln_pib_cte.dl1   0.43173    1.08650   0.397   0.6960
## edu.dl1          0.25818    0.12043   2.144   0.0468 *
## salud.dl1        0.24254    0.21687   1.118   0.2790
## rec_imp.dl1     -0.04270    0.03451  -1.237   0.2328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1349 on 17 degrees of freedom
## Multiple R-squared:  0.3807, Adjusted R-squared:  0.1258
## F-statistic: 1.493 on 7 and 17 DF,  p-value: 0.2351
##
##
## Response rec_imp.d :
##
## Call:
## lm(formula = rec_imp.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61135 -0.53487  0.05402  0.45646  1.19171
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1              5.7533     2.9231   1.968   0.0656 .
## ect2             -0.3773     0.4174  -0.904   0.3787
## constant       -163.7280    85.2841  -1.920   0.0718 .
## ln_pib_cte.dl1    7.1131     6.2772   1.133   0.2729
## edu.dl1           1.5308     0.6958   2.200   0.0419 *
## salud.dl1        -0.9392     1.2529  -0.750   0.4637
## rec_imp.dl1       0.2297     0.1994   1.152   0.2652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7792 on 17 degrees of freedom
## Multiple R-squared:  0.5817, Adjusted R-squared:  0.4095
## F-statistic: 3.378 on 7 and 17 DF,  p-value: 0.01903
```

```
mysum <- summary(M1.jorls$rlm)
screenreg(list(mysum[[1]], mysum[[2]],mysum[[3]],mysum[[4]]))
```

```
##
## ===========================================================
##                  Model 1  Model 2     Model 3   Model 4
## -----------------------------------------------------------
## ect1             -0.17      2.63 **     0.63      5.75
##                  (0.12)    (0.73)      (0.51)    (2.92)
## ect2              0.02     -0.45 ***   -0.06     -0.38
##                  (0.02)    (0.10)      (0.07)    (0.42)
## constant          4.90    -77.62 **   -18.05   -163.73
##                  (3.45)   (21.19)     (14.76)   (85.28)
## ln_pib_cte.dl1   -0.08     -3.97 *      0.43      7.11
##                  (0.25)    (1.56)      (1.09)    (6.28)
## edu.dl1          -0.04     -0.17        0.26 *    1.53 *
##                  (0.03)    (0.17)      (0.12)    (0.70)
## salud.dl1        -0.09      0.32        0.24     -0.94
##                  (0.05)    (0.31)      (0.22)    (1.25)
## rec_imp.dl1      -0.00     -0.18 **    -0.04      0.23
##                  (0.01)    (0.05)      (0.03)    (0.20)
## -----------------------------------------------------------
## R^2               0.56      0.66        0.38      0.58
## Adj. R^2          0.37      0.52        0.13      0.41
## Num. obs.        24        24          24        24
## RMSE              0.03      0.19        0.13      0.78
## ===========================================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

Finalmente obsevemos nuestro modelo ajustado vs los datos recopilados:




```r
vecm.level <- vec2var(M1,r=rel_coint_aux)
arch.test(vecm.level)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 190, df = 500, p-value = 1
```

```r
normality.test(vecm.level)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 20.104, df = 8, p-value = 0.009948
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 10.138, df = 4, p-value = 0.03816
##
##
## $Kurtosis
```

```
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 9.966, df = 4, p-value = 0.04101
```

```
serial.test(vecm.level)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 174.4, df = 228, p-value = 0.9966
```

```
predict(vecm.level)
```

```
## $ln_pib_cte
##           fcst    lower    upper         CI
##  [1,] 30.55934 30.50739 30.61130 0.05195443
##  [2,] 30.57767 30.51072 30.64462 0.06694729
##  [3,] 30.59004 30.51119 30.66889 0.07885251
##  [4,] 30.61790 30.53273 30.70308 0.08517283
##  [5,] 30.64494 30.55536 30.73452 0.08958148
##  [6,] 30.67394 30.57885 30.76903 0.09509408
##  [7,] 30.70390 30.60335 30.80444 0.10054392
##  [8,] 30.72937 30.62278 30.83596 0.10658961
##  [9,] 30.75334 30.64022 30.86646 0.11311650
## [10,] 30.77627 30.65702 30.89552 0.11925199
##
## $edu
##           fcst    lower    upper        CI
##  [1,] 6.077113 5.757831 6.396395 0.3192819
##  [2,] 6.532336 6.162907 6.901765 0.3694286
##  [3,] 6.795815 6.332980 7.258650 0.4628350
##  [4,] 7.014157 6.395996 7.632317 0.6181605
##  [5,] 7.072200 6.291781 7.852618 0.7804186
##  [6,] 7.082083 6.147802 8.016364 0.9342809
##  [7,] 7.134247 6.091060 8.177434 1.0431870
##  [8,] 7.212057 6.094409 8.329704 1.1176475
##  [9,] 7.344362 6.170599 8.518125 1.1737630
## [10,] 7.514848 6.295759 8.733938 1.2190897
##
## $salud
##           fcst    lower    upper        CI
##  [1,] 2.776238 2.553774 2.998701 0.2224634
##  [2,] 2.920687 2.564271 3.277103 0.3564157
##  [3,] 3.008082 2.542615 3.473549 0.4654673
##  [4,] 3.050788 2.507351 3.594225 0.5434369
##  [5,] 3.110439 2.508689 3.712189 0.6017496
##  [6,] 3.122135 2.469348 3.774922 0.6527868
##  [7,] 3.129627 2.431256 3.827998 0.6983710
##  [8,] 3.149231 2.408864 3.889597 0.7403664
##  [9,] 3.166564 2.384987 3.948141 0.7815772
## [10,] 3.196444 2.374667 4.018221 0.8217770
##
```

```
## $rec_imp
##           fcst    lower    upper       CI
##  [1,] 12.73788 11.45261 14.02315 1.285271
##  [2,] 12.97129 11.37561 14.56698 1.595683
##  [3,] 12.64827 10.92842 14.36812 1.719846
##  [4,] 12.08990 10.21225 13.96755 1.877649
##  [5,] 12.03398 10.11163 13.95633 1.922351
##  [6,] 11.97677 10.05006 13.90349 1.926714
##  [7,] 12.10515 10.17624 14.03406 1.928910
##  [8,] 12.40626 10.47264 14.33988 1.933624
##  [9,] 12.62933 10.69245 14.56621 1.936876
## [10,] 12.82522 10.88297 14.76748 1.942256
```

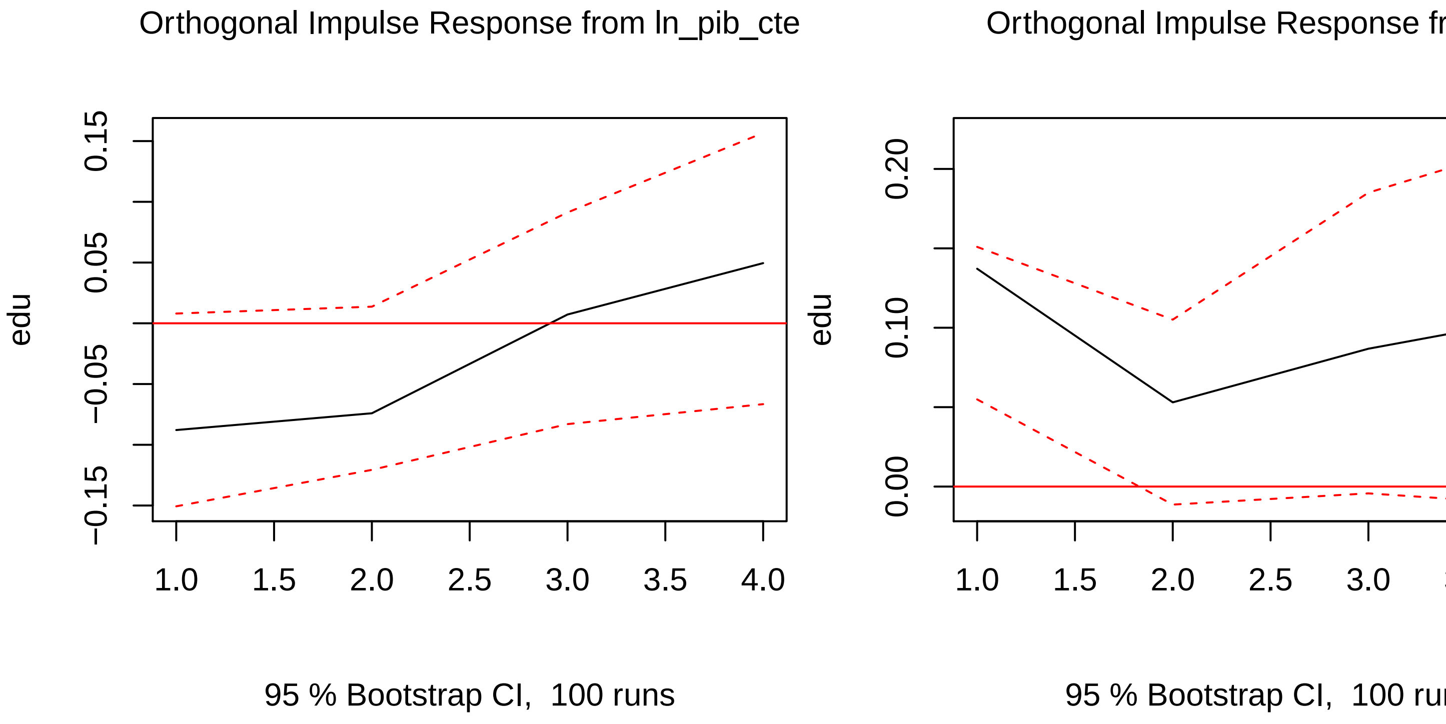
```r
irf(vecm.level,boot=FALSE)
```

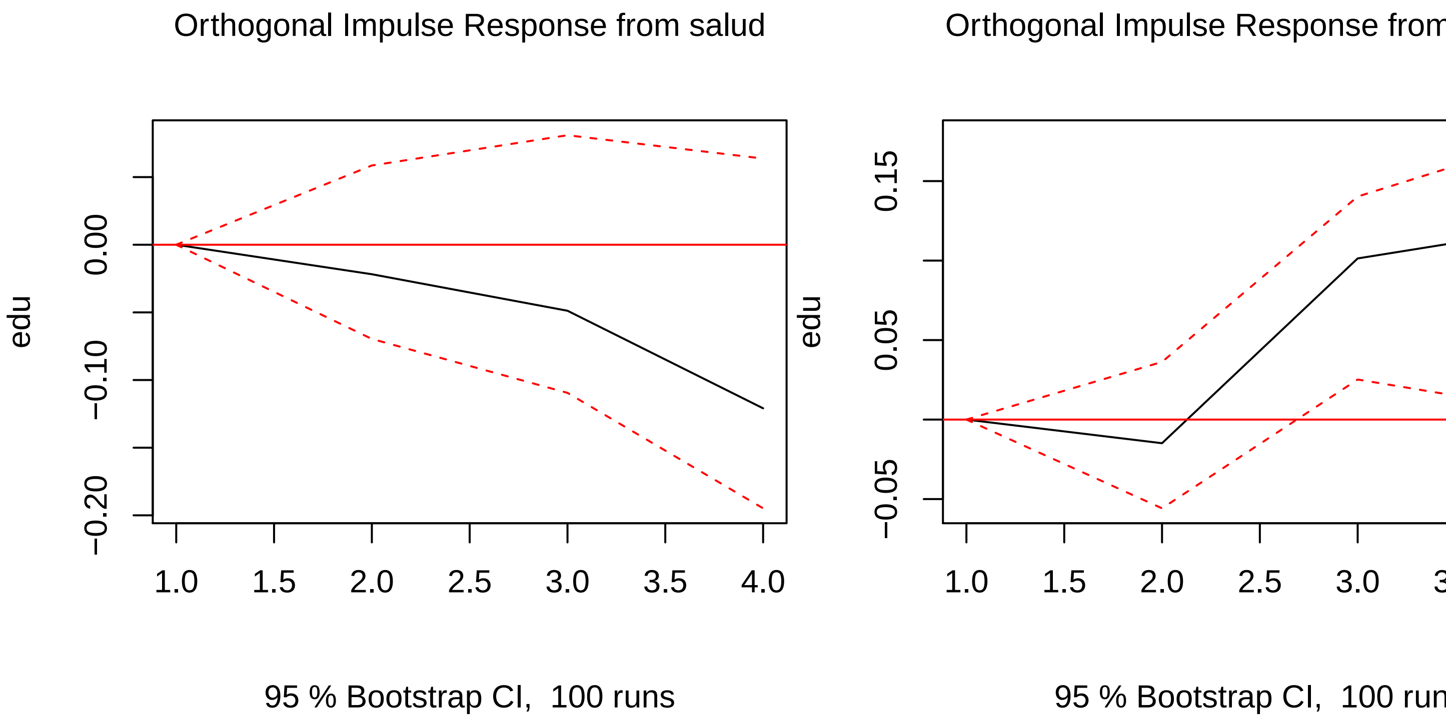
```
## 
## Impulse response coefficients
## $ln_pib_cte
##        ln_pib_cte          edu      salud      rec_imp
##  [1,] 0.02650785 -0.087870552 0.03119667 -0.024312688
##  [2,] 0.01977019 -0.074052969 0.04950254  0.185641495
##  [3,] 0.01610376  0.007232141 0.05175272  0.173857822
##  [4,] 0.01291972  0.049538716 0.07112342  0.207373014
##  [5,] 0.01186309  0.061511538 0.07541191  0.111921878
##  [6,] 0.01291634  0.063195901 0.07671714  0.027963439
##  [7,] 0.01350821  0.043648508 0.07887976  0.015708690
##  [8,] 0.01436027  0.024495389 0.07446609  0.002292907
##  [9,] 0.01503154  0.013512040 0.07133805  0.021112885
## [10,] 0.01503096  0.007805182 0.06977369  0.052721406
## [11,] 0.01489427  0.010363860 0.06832830  0.068505909
## 
## $edu
##          ln_pib_cte        edu       salud       rec_imp
##  [1,] 0.0000000000 0.13717073 -0.03155182  0.1986786261
##  [2,] 0.0003055544 0.05301416 -0.02674002  0.2981871446
##  [3,] 0.0067239401 0.08678498 -0.06462684 -0.0379783453
##  [4,] 0.0071454399 0.10945428 -0.04314167  0.0783861847
##  [5,] 0.0047943376 0.09384652 -0.03943127  0.0822260391
##  [6,] 0.0062375797 0.10969458 -0.04719915 -0.0134149215
##  [7,] 0.0060360972 0.10503034 -0.03946851  0.0302987376
##  [8,] 0.0058917722 0.09486340 -0.04206944  0.0134442591
##  [9,] 0.0065688497 0.09587811 -0.04456224  0.0009084525
## [10,] 0.0064059752 0.09167027 -0.04293135  0.0260372065
## [11,] 0.0063747058 0.09081330 -0.04457499  0.0233630402
## 
## $salud
##           ln_pib_cte         edu      salud      rec_imp
##  [1,]  0.0000000000  0.00000000 0.10447185  0.029500992
##  [2,] -0.0065975448 -0.02179995 0.11982458 -0.126689736
##  [3,] -0.0043553753 -0.04877702 0.11542497 -0.242635645
##  [4,] -0.0031129077 -0.12088983 0.11508697 -0.165159098
##  [5,] -0.0011573818 -0.16630051 0.09771601 -0.146097067
##  [6,] -0.0001684599 -0.17641559 0.09053670 -0.057074147
##  [7,] -0.0011821703 -0.17326702 0.08976055  0.032203159
##  [8,] -0.0020376260 -0.15236310 0.08873314  0.048831693
```

```
##  [9,] -0.0028928177 -0.13091912 0.09293202  0.052625197
## [10,] -0.0035553118 -0.11919965 0.09672086  0.031843772
## [11,] -0.0035600483 -0.11457887 0.09842410 -0.002132192
##
## $rec_imp
##         ln_pib_cte         edu       salud      rec_imp
##  [1,] 0.000000000  0.00000000  0.00000000  0.623770656
##  [2,] 0.005438855 -0.01486241 -0.05159602  0.305564320
##  [3,] 0.011328809  0.10135897 -0.05615848  0.128930383
##  [4,] 0.006496407  0.12107961 -0.01768759  0.267124292
##  [5,] 0.005958048  0.13721303 -0.02436241  0.059898784
##  [6,] 0.007697202  0.14673085 -0.01890699 -0.012342925
##  [7,] 0.007566109  0.11447091 -0.01402216 -0.001278395
##  [8,] 0.008994716  0.09524006 -0.02323532 -0.046579823
##  [9,] 0.009789247  0.08341227 -0.02481857 -0.007763804
## [10,] 0.009566978  0.07457358 -0.02668871  0.031000116
## [11,] 0.009560181  0.07955572 -0.02928219  0.041511038
```

```
#fevd(vecm.level)
```

```
plot(irf(vecm.level,n.ahead=3,response='edu',boot=TRUE))
```






```
BIC(vec2var(M1,r=rel_coint_aux))
```

```
## [1] -13.84123
```

12.3 Modelo 3: VECM con K=3 (lags), y r=1 (relaciones de cointegración)

```
#traemos las series
ln_pib_cte <-  series_db$log_GDP_constante
edu <- series_db$Gasto_Educacion_PorcGDP
salud <- series_db$Gasto_Salud_PorcGDP
rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP

# generamos la matriz X
model.data <- cbind(ln_pib_cte,edu,salud,rec_imp)

# estimamos el rango de cointegracion de PI con ca.jo
M1 <- ca.jo(model.data,spec='transitory', type='eigen',K=3)
```

Desde los objetos obtenidos por el ajuste del modelo, se puede obtener β_c de la siguiente manera

```
k_aux <- 4
rel_coint_aux <- 1

tmp_beta <-  M1@V[,1:rel_coint_aux]
st <- cbind(diag(rel_coint_aux),matrix(0,rel_coint_aux,(k_aux-rel_coint_aux)))
beta_c <- tmp_beta%*%solve(st%*%tmp_beta)
beta_c
```

```
##             [,1]
```

```
## [1,]  1.0000000
## [2,] -0.2551987
## [3,] -0.2733728
## [4,]  0.2757764
```

De tal manera que los vectores de cointegración tienen una interpretación mucho más sencilla.

Este resultado puede ser validado con lo que se obtiene a partir de la función cajorls

```
M1.jorls <- cajorls(M1, r = rel_coint_aux)
M1.jorls$beta
```

```
##                     ect1
## ln_pib_cte.l1  1.0000000
## edu.l1        -0.2551987
## salud.l1      -0.2733728
## rec_imp.l1     0.2757764
```

Ahora bien, Una vez que se han definido las relaciones de cointegración restringida veamos cómo es el ajuste del modelo a partir de estas definiciones:

```
summary(M1.jorls$rlm)
```

```
## Response ln_pib_cte.d :
##
## Call:
## lm(formula = ln_pib_cte.d ~ ect1 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 +
##     salud.dl2 + rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.064620 -0.008924  0.005533  0.017995  0.037844
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## ect1            0.11332    0.08801   1.288    0.220
## constant       -3.46748    2.72848  -1.271    0.226
## ln_pib_cte.dl1 -0.64815    0.38279  -1.693    0.114
## edu.dl1        -0.07366    0.04420  -1.667    0.120
## salud.dl1      -0.04525    0.06392  -0.708    0.491
## rec_imp.dl1    -0.02058    0.01685  -1.221    0.244
## ln_pib_cte.dl2 -0.55179    0.39058  -1.413    0.181
## edu.dl2        -0.05473    0.04474  -1.223    0.243
## salud.dl2      -0.02391    0.06393  -0.374    0.714
## rec_imp.dl2    -0.01915    0.01732  -1.105    0.289
##
## Residual standard error: 0.03544 on 13 degrees of freedom
## Multiple R-squared:  0.5524, Adjusted R-squared:  0.208
## F-statistic: 1.604 on 10 and 13 DF,  p-value: 0.2096
##
##
## Response edu.d :
##
## Call:
## lm(formula = edu.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 + salud.dl2 +
```

```
##     rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39058 -0.11580 -0.00759  0.14918  0.32439
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             1.31537    0.61076   2.154   0.0506 .
## constant       -40.80566   18.93476  -2.155   0.0505 .
## ln_pib_cte.dl1  -3.35748    2.65644  -1.264   0.2285
## edu.dl1         -0.41775    0.30674  -1.362   0.1964
## salud.dl1        0.09742    0.44356   0.220   0.8296
## rec_imp.dl1     -0.26018    0.11695  -2.225   0.0444 *
## ln_pib_cte.dl2  -0.78458    2.71053  -0.289   0.7768
## edu.dl2         -0.22663    0.31051  -0.730   0.4784
## salud.dl2        0.17681    0.44364   0.399   0.6967
## rec_imp.dl2     -0.08504    0.12022  -0.707   0.4918
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2459 on 13 degrees of freedom
## Multiple R-squared:  0.5554, Adjusted R-squared:  0.2134
## F-statistic: 1.624 on 10 and 13 DF,  p-value: 0.2038
##
##
## Response salud.d :
##
## Call:
## lm(formula = salud.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 + salud.dl2 +
##     rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.125937 -0.044893 -0.007654  0.043395  0.129886
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1           -0.424615   0.223279  -1.902  0.07960 .
## constant       13.269233   6.922098   1.917  0.07749 .
## ln_pib_cte.dl1  0.783238   0.971131   0.807  0.43445
## edu.dl1         0.333203   0.112135   2.971  0.01082 *
## salud.dl1       0.531420   0.162156   3.277  0.00601 **
## rec_imp.dl1    -0.001809   0.042755  -0.042  0.96690
## ln_pib_cte.dl2 -0.604075   0.990906  -0.610  0.55262
## edu.dl2        -0.094375   0.113516  -0.831  0.42077
## salud.dl2      -0.663533   0.162183  -4.091  0.00127 **
## rec_imp.dl2     0.114249   0.043951   2.599  0.02203 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0899 on 13 degrees of freedom
## Multiple R-squared:  0.7861, Adjusted R-squared:  0.6215
```

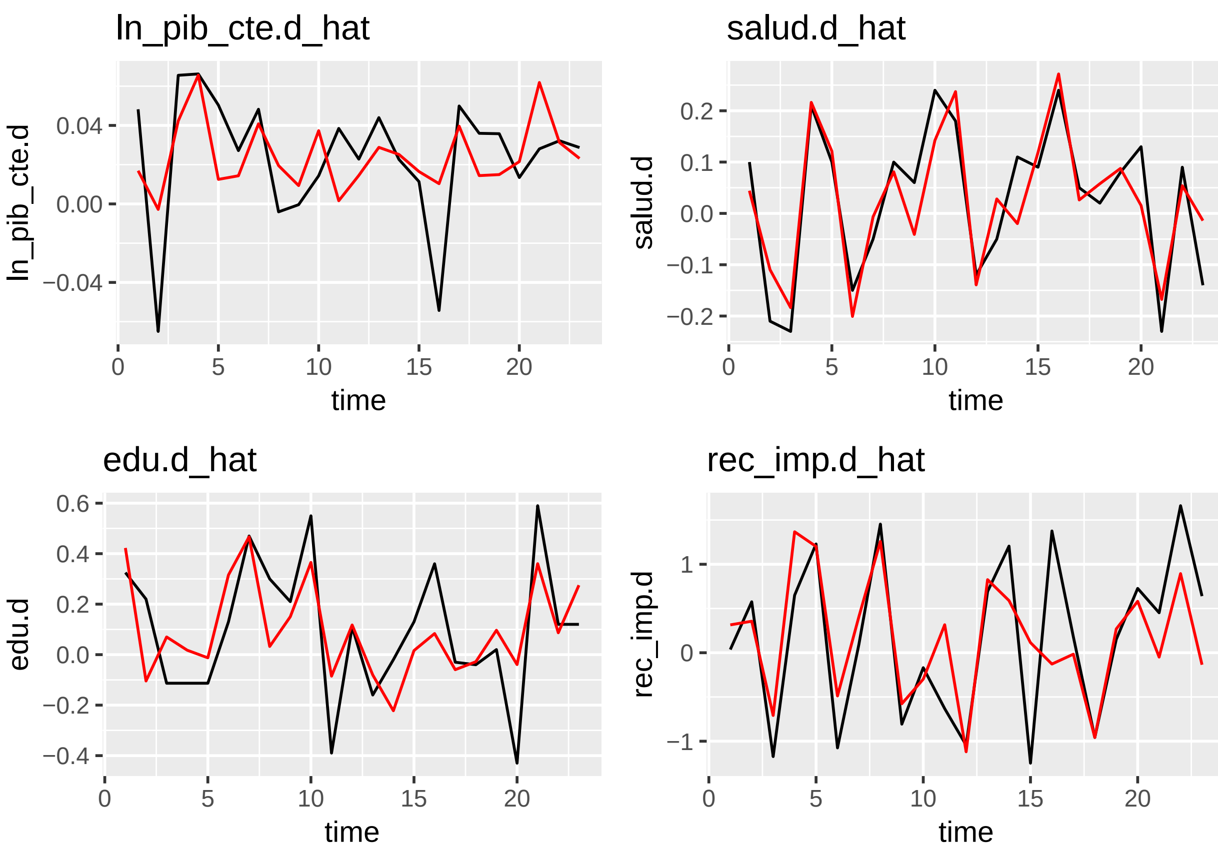
```
## F-statistic: 4.777 on 10 and 13 DF,  p-value: 0.005201
##
##
## Response rec_imp.d :
##
## Call:
## lm(formula = rec_imp.d ~ ect1 + constant + ln_pib_cte.dl1 + edu.dl1 +
##     salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 + salud.dl2 +
##     rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36265 -0.29583  0.02853  0.20818  1.50462
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1             1.41481    1.98995   0.711   0.4897
## constant       -43.43218   61.69260  -0.704   0.4938
## ln_pib_cte.dl1  -1.01710    8.65512  -0.118   0.9082
## edu.dl1         -0.04207    0.99940  -0.042   0.9671
## salud.dl1       -0.43089    1.44520  -0.298   0.7703
## rec_imp.dl1     -0.43743    0.38105  -1.148   0.2717
## ln_pib_cte.dl2  -7.83374    8.83136  -0.887   0.3912
## edu.dl2         -1.93246    1.01171  -1.910   0.0784 .
## salud.dl2       -2.66402    1.44544  -1.843   0.0882 .
## rec_imp.dl2     -0.48020    0.39171  -1.226   0.2420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8013 on 13 degrees of freedom
## Multiple R-squared:  0.5689, Adjusted R-squared:  0.2374
## F-statistic: 1.716 on 10 and 13 DF,  p-value: 0.179
```

```r
mysum <- summary(M1.jorls$rlm)
screenreg(list(mysum[[1]], mysum[[2]],mysum[[3]],mysum[[4]]))
```

```
##
## ====================================================
##                 Model 1  Model 2   Model 3   Model 4
## ----------------------------------------------------
## ect1             0.11      1.32    -0.42       1.41
##                 (0.09)    (0.61)   (0.22)     (1.99)
## constant        -3.47    -40.81    13.27     -43.43
##                 (2.73)   (18.93)   (6.92)    (61.69)
## ln_pib_cte.dl1  -0.65     -3.36     0.78      -1.02
##                 (0.38)    (2.66)   (0.97)     (8.66)
## edu.dl1         -0.07     -0.42     0.33 *    -0.04
##                 (0.04)    (0.31)   (0.11)     (1.00)
## salud.dl1       -0.05      0.10     0.53 **   -0.43
##                 (0.06)    (0.44)   (0.16)     (1.45)
## rec_imp.dl1     -0.02     -0.26 *  -0.00      -0.44
##                 (0.02)    (0.12)   (0.04)     (0.38)
## ln_pib_cte.dl2  -0.55     -0.78    -0.60      -7.83
##                 (0.39)    (2.71)   (0.99)     (8.83)
## edu.dl2         -0.05     -0.23    -0.09      -1.93
```

```
##                   (0.04)     (0.31)    (0.11)     (1.01)
## salud.dl2         -0.02       0.18     -0.66 **   -2.66
##                   (0.06)     (0.44)    (0.16)     (1.45)
## rec_imp.dl2       -0.02      -0.09      0.11 *    -0.48
##                   (0.02)     (0.12)    (0.04)     (0.39)
## ----------------------------------------------------
## R^2                0.55       0.56      0.79       0.57
## Adj. R^2           0.21       0.21      0.62       0.24
## Num. obs.         23         23        23         23
## RMSE               0.04       0.25      0.09       0.80
## ====================================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

Finalmente obsevemos nuestro modelo ajustado vs los datos recopilados:




```
vecm.level <- vec2var(M1,r=rel_coint_aux)
arch.test(vecm.level)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 180, df = 500, p-value = 1
```

```
normality.test(vecm.level)
```

```
## $JB
##
##  JB-Test (multivariate)
```

```
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 29.45, df = 8, p-value = 0.0002643
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 15.118, df = 4, p-value = 0.004463
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 14.333, df = 4, p-value = 0.006306
```

```
serial.test(vecm.level)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 156.46, df = 212, p-value = 0.9984
```

```
predict(vecm.level)
```

```
## $ln_pib_cte
##           fcst    lower    upper         CI
##  [1,] 30.58335 30.53114 30.63557 0.05221794
##  [2,] 30.61140 30.53703 30.68577 0.07436864
##  [3,] 30.63445 30.54432 30.72458 0.09013325
##  [4,] 30.68663 30.57995 30.79330 0.10667838
##  [5,] 30.71560 30.58991 30.84130 0.12569521
##  [6,] 30.74623 30.60488 30.88758 0.14134924
##  [7,] 30.79402 30.63881 30.94924 0.15521395
##  [8,] 30.82699 30.65668 30.99730 0.17030986
##  [9,] 30.86241 30.67755 31.04727 0.18486294
## [10,] 30.90642 30.70694 31.10590 0.19948138
##
## $edu
##            fcst    lower     upper        CI
##  [1,]  6.437152 6.074777  6.799528 0.3623757
##  [2,]  6.960571 6.509653  7.411489 0.4509185
##  [3,]  7.666197 7.069040  8.263354 0.5971568
##  [4,]  8.576224 7.778922  9.373526 0.7973023
##  [5,]  9.192893 8.105446 10.280339 1.0874467
##  [6,]  9.936738 8.481743 11.391733 1.4549949
##  [7,] 10.747608 8.893253 12.601962 1.8543543
##  [8,] 11.386826 9.098617 13.675035 2.2882091
##  [9,] 12.164780 9.418884 14.910675 2.7458956
## [10,] 12.980324 9.749561 16.211086 3.2307623
```

```
##
## $salud
##            fcst      lower    upper        CI
##  [1,] 2.600278  2.4678017 2.732754 0.1324760
##  [2,] 2.613927  2.3189618 2.908892 0.2949651
##  [3,] 2.426350  2.0241454 2.828555 0.4022047
##  [4,] 2.263039  1.7509631 2.775115 0.5120761
##  [5,] 2.188472  1.5614863 2.815458 0.6269858
##  [6,] 1.962251  1.2142373 2.710264 0.7480133
##  [7,] 1.803997  0.9093393 2.698654 0.8946573
##  [8,] 1.695141  0.6400899 2.750193 1.0550516
##  [9,] 1.487589  0.2688420 2.706337 1.2187474
## [10,] 1.335938 -0.0585061 2.730382 1.3944440
##
## $rec_imp
##            fcst    lower    upper       CI
##  [1,] 13.78677 12.60609 14.96745 1.180680
##  [2,] 15.21376 13.62497 16.80255 1.588788
##  [3,] 15.18568 13.17595 17.19540 2.009725
##  [4,] 15.40113 12.88466 17.91761 2.516476
##  [5,] 16.27567 13.38746 19.16388 2.888207
##  [6,] 16.24326 13.00738 19.47914 3.235885
##  [7,] 16.71124 13.14407 20.27842 3.567174
##  [8,] 17.51022 13.61587 21.40458 3.894357
##  [9,] 17.64983 13.40617 21.89349 4.243659
## [10,] 18.17329 13.57046 22.77613 4.602834
```

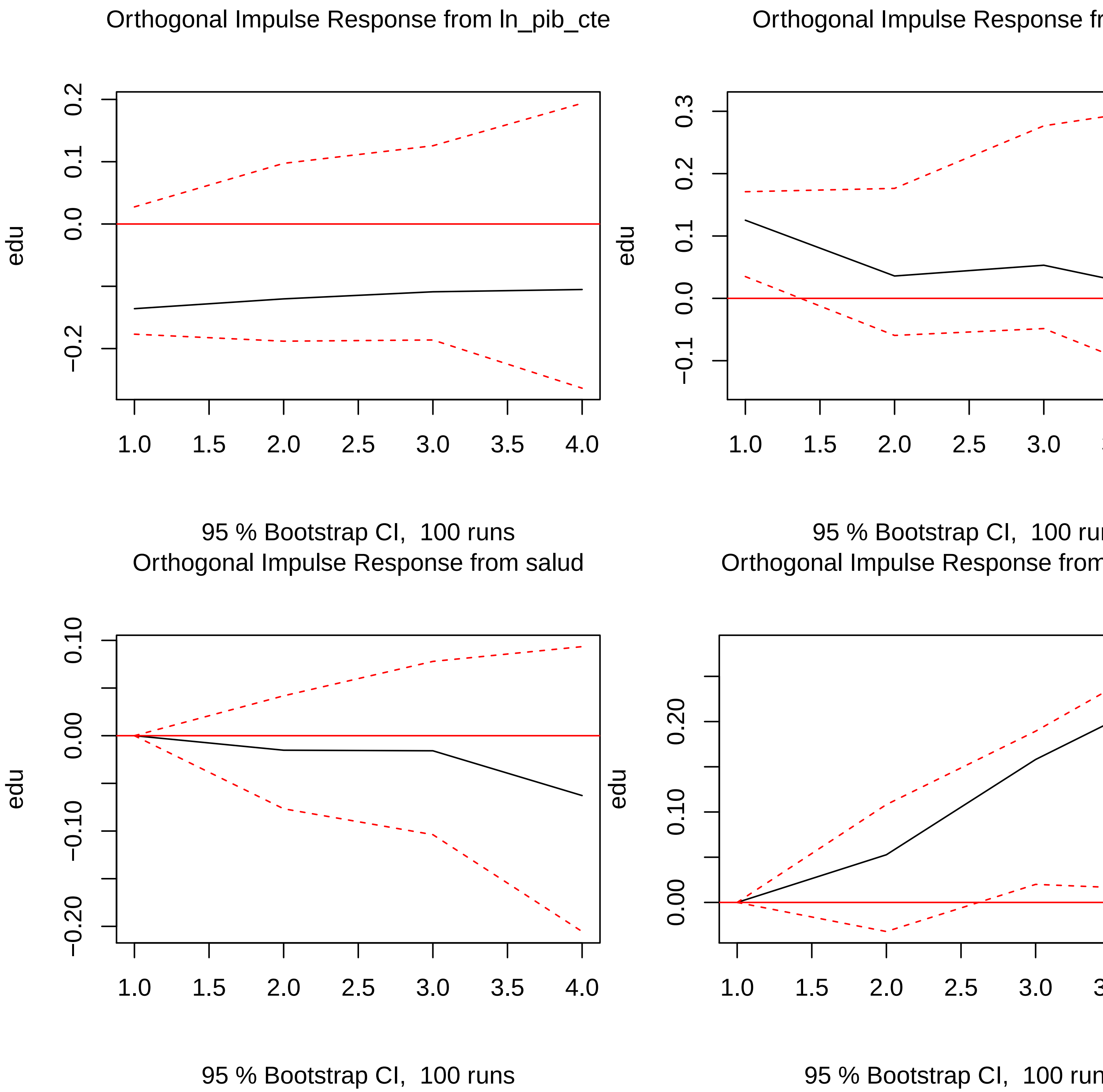
```r
irf(vecm.level,boot=FALSE)
```

```
##
## Impulse response coefficients
## $ln_pib_cte
##        ln_pib_cte         edu        salud      rec_imp
##  [1,] 0.02664230 -0.13594153  0.002081061 -0.309374702
##  [2,] 0.02287819 -0.12020449 -0.010256918 -0.231058708
##  [3,] 0.02088415 -0.10882041 -0.051524699 -0.070237923
##  [4,] 0.02482186 -0.10514992 -0.072007649 -0.065556615
##  [5,] 0.02725043 -0.07504652 -0.055323516  0.038750530
##  [6,] 0.02676706 -0.03424351 -0.053974890  0.098743927
##  [7,] 0.02617682  0.02213995 -0.076515637  0.008597935
##  [8,] 0.02709918  0.06572744 -0.086210416  0.010495818
##  [9,] 0.02736409  0.08273848 -0.092047726  0.062121119
## [10,] 0.02794316  0.11211844 -0.108001249  0.043208632
## [11,] 0.02920904  0.14541112 -0.115508171  0.070025815
##
## $edu
##          ln_pib_cte           edu         salud       rec_imp
##  [1,]  0.000000000  0.125314888 0.0002362181  0.048998045
##  [2,] -0.012350011  0.035863086 0.0498975235 -0.004028229
##  [3,] -0.010823186  0.053118072 0.0459144352 -0.303524114
##  [4,] -0.008872427  0.002889556 0.0794005026 -0.180980751
##  [5,] -0.015635979 -0.098484267 0.0821305692 -0.254679119
##  [6,] -0.013293562 -0.118059198 0.0753059659 -0.396441583
##  [7,] -0.012331911 -0.195240197 0.1186850791 -0.216034202
##  [8,] -0.016468805 -0.267217700 0.1283661063 -0.282787234
```

```
##  [9,] -0.015300740 -0.276585445 0.1309905543 -0.397905268
## [10,] -0.016076966 -0.338164201 0.1639538446 -0.299406929
## [11,] -0.019025478 -0.395986095 0.1697261583 -0.376956915
##
## $salud
##         ln_pib_cte          edu      salud     rec_imp
##  [1,]  0.000000000  0.00000000 0.06755856  0.02442764
##  [2,] -0.004889509 -0.01520577 0.10839800 -0.03196692
##  [3,] -0.007842645 -0.01581954 0.09602453 -0.26681587
##  [4,] -0.008601252 -0.06277397 0.10220964 -0.31638079
##  [5,] -0.009668003 -0.14863580 0.11832958 -0.27703730
##  [6,] -0.009522250 -0.20717379 0.12667782 -0.30636217
##  [7,] -0.009516317 -0.25463063 0.14649570 -0.28957203
##  [8,] -0.011084043 -0.30490373 0.16509784 -0.29081296
##  [9,] -0.012023706 -0.34076485 0.17513500 -0.34668773
## [10,] -0.012649793 -0.38125970 0.19019746 -0.36061559
## [11,] -0.013914638 -0.42962185 0.20351514 -0.37642441
##
## $rec_imp
##        ln_pib_cte         edu       salud   rec_imp
##  [1,] 0.000000000 0.00000000  0.00000000 0.5139789
##  [2,] 0.005484920 0.05271944 -0.06111591 0.4896892
##  [3,] 0.007764521 0.15806614 -0.07401895 0.4754629
##  [4,] 0.008879186 0.24009973 -0.06491104 0.6781820
##  [5,] 0.008355053 0.32393039 -0.10131533 0.6163515
##  [6,] 0.010214906 0.43038071 -0.13670636 0.5417531
##  [7,] 0.011937431 0.49050523 -0.14596428 0.6753540
##  [8,] 0.012279216 0.54696430 -0.17389582 0.6861915
##  [9,] 0.014767167 0.63270617 -0.20146884 0.6764659
## [10,] 0.016226304 0.69432404 -0.21165816 0.7781959
## [11,] 0.016525320 0.75746269 -0.23534003 0.7847116
```

```
#fevd(vecm.level)
```

```
plot(irf(vecm.level,n.ahead=3,response='edu',boot=TRUE))
```




```r
BIC(vec2var(M1,r=rel_coint_aux))
```

```
## [1] 7.254725
```

12.4 Modelo 4: VECM con K=3 (lags), y r=2 (relaciones de cointegración)

```
#traemos las series
ln_pib_cte <-  series_db$log_GDP_constante
edu <- series_db$Gasto_Educacion_PorcGDP
salud <- series_db$Gasto_Salud_PorcGDP
rec_imp <- series_db$Recaudacion_Impositiva_PorcGDP

# generamos la matriz X
model.data <- cbind(ln_pib_cte,edu,salud,rec_imp)

# estimamos el rango de cointegracion de PI con ca.jo
M1 <- ca.jo(model.data,spec='transitory', type='eigen',K=3)
```

Desde los objetos obtenidos por el ajuste del modelo, se puede obtener β_c de la siguiente manera

```
k_aux <- 4
rel_coint_aux <- 2

tmp_beta <-  M1@V[,1:rel_coint_aux]
st <- cbind(diag(rel_coint_aux),matrix(0,rel_coint_aux,(k_aux-rel_coint_aux)))
beta_c <- tmp_beta%*%solve(st%*%tmp_beta)
beta_c
```

```
##                       [,1]      [,2]
## ln_pib_cte.l1  1.0000000  0.000000
## edu.l1         0.0000000  1.000000
## salud.l1       0.5133019  3.082597
## rec_imp.l1    -0.8837869 -4.543767
```

De tal manera que los vectores de cointegración tienen una interpretación mucho más sencilla.

Este resultado puede ser validado con lo que se obtiene a partir de la función cajorls

```
M1.jorls <- cajorls(M1, r = rel_coint_aux)
M1.jorls$beta
```

```
##                       ect1      ect2
## ln_pib_cte.l1  1.0000000  0.000000
## edu.l1         0.0000000  1.000000
## salud.l1       0.5133019  3.082597
## rec_imp.l1    -0.8837869 -4.543767
```

Ahora bien, Una vez que se han definido las relaciones de cointegración restringida veamos cómo es el ajuste del modelo a partir de estas definiciones:

```
summary(M1.jorls$rlm)
```

```
## Response ln_pib_cte.d :
##
## Call:
## lm(formula = ln_pib_cte.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 +
##     salud.dl2 + rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.056835 -0.008283  0.002389  0.012079  0.027955
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## ect1           -0.31206    0.15041  -2.075  0.06019 .
## ect2            0.04601    0.02929   1.571  0.14217
## constant        8.66010    4.36656   1.983  0.07070 .
## ln_pib_cte.dl1 -0.39278    0.30495  -1.288  0.22203
## edu.dl1        -0.13193    0.03863  -3.416  0.00512 **
## salud.dl1      -0.05962    0.04932  -1.209  0.24997
## rec_imp.dl1    -0.03547    0.01378  -2.575  0.02432 *
## ln_pib_cte.dl2 -0.33674    0.30767  -1.094  0.29523
## edu.dl2        -0.12503    0.04092  -3.055  0.00999 **
## salud.dl2      -0.05833    0.05030  -1.160  0.26879
## rec_imp.dl2    -0.02284    0.01336  -1.709  0.11308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02723 on 12 degrees of freedom
## Multiple R-squared:  0.7561, Adjusted R-squared:  0.5325
## F-statistic: 3.382 on 11 and 12 DF,  p-value: 0.02341
##
##
## Response edu.d :
##
## Call:
## lm(formula = edu.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 +
##     salud.dl2 + rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41729 -0.04614 -0.00449  0.08467  0.28656
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## ect1            4.333e+00  1.024e+00   4.232 0.001164 **
## ect2           -8.673e-01  1.994e-01  -4.350 0.000946 ***
## constant       -1.268e+02  2.973e+01  -4.267 0.001094 **
## ln_pib_cte.dl1 -5.169e+00  2.076e+00  -2.490 0.028436 *
## edu.dl1        -4.358e-03  2.630e-01  -0.017 0.987050
## salud.dl1       1.993e-01  3.357e-01   0.594 0.563700
## rec_imp.dl1    -1.545e-01  9.378e-02  -1.648 0.125282
## ln_pib_cte.dl2 -2.310e+00  2.095e+00  -1.103 0.291660
## edu.dl2         2.721e-01  2.786e-01   0.977 0.347998
## salud.dl2       4.210e-01  3.424e-01   1.229 0.242533
## rec_imp.dl2    -5.885e-02  9.096e-02  -0.647 0.529827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1853 on 12 degrees of freedom
## Multiple R-squared:  0.7669, Adjusted R-squared:  0.5532
## F-statistic: 3.589 on 11 and 12 DF,  p-value: 0.01879
```

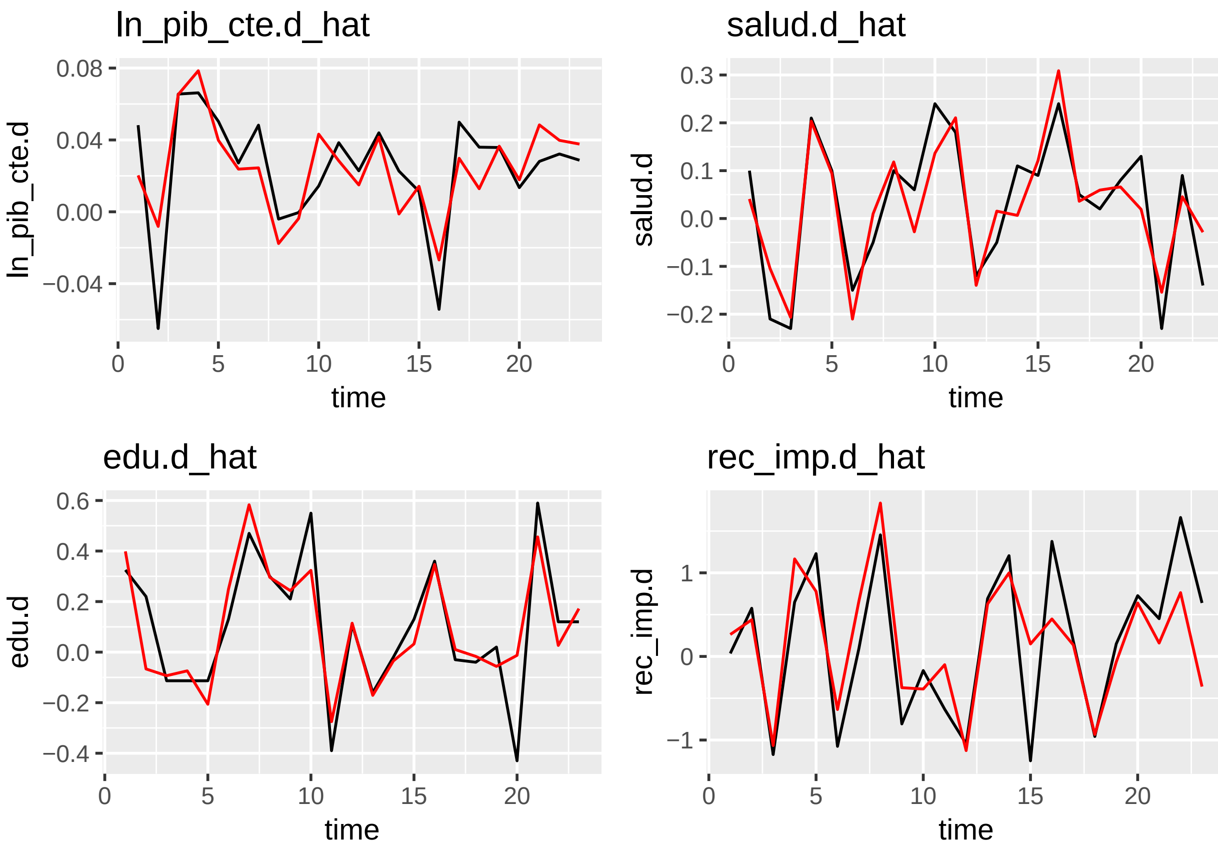
```
##
##
## Response salud.d :
##
## Call:
## lm(formula = salud.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 +
##     salud.dl2 + rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.111421 -0.049610  0.006008  0.051836  0.110819
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## ect1            0.0007667  0.4983438   0.002  0.99880
## ect2            0.0334274  0.0970448   0.344  0.73647
## constant        1.1415918 14.4673519   0.079  0.93841
## ln_pib_cte.dl1  0.5278656  1.0103731   0.522  0.61087
## edu.dl1         0.3914729  0.1279734   3.059  0.00992 **
## salud.dl1       0.5457862  0.1633934   3.340  0.00588 **
## rec_imp.dl1     0.0130817  0.0456404   0.287  0.77929
## ln_pib_cte.dl2 -0.8191312  1.0193844  -0.804  0.43728
## edu.dl2        -0.0240736  0.1355908  -0.178  0.86204
## salud.dl2      -0.6291192  0.1666646  -3.775  0.00265 **
## rec_imp.dl2     0.1179407  0.0442675   2.664  0.02063 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09021 on 12 degrees of freedom
## Multiple R-squared:  0.8012, Adjusted R-squared:  0.619
## F-statistic: 4.397 on 11 and 12 DF,  p-value: 0.008467
##
##
## Response rec_imp.d :
##
## Call:
## lm(formula = rec_imp.d ~ ect1 + ect2 + constant + ln_pib_cte.dl1 +
##     edu.dl1 + salud.dl1 + rec_imp.dl1 + ln_pib_cte.dl2 + edu.dl2 +
##     salud.dl2 + rec_imp.dl2 - 1, data = data.mat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.39772 -0.40729  0.06374  0.21882  1.00065
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## ect1              8.0191     4.0832   1.964   0.0731 .
## ect2             -1.5245     0.7951  -1.917   0.0793 .
## constant       -231.7213   118.5377  -1.955   0.0743 .
## ln_pib_cte.dl1   -4.9819     8.2785  -0.602   0.5585
## edu.dl1           0.8626     1.0485   0.823   0.4267
## salud.dl1        -0.2079     1.3388  -0.155   0.8792
## rec_imp.dl1      -0.2062     0.3740  -0.552   0.5914
```

```
## ln_pib_cte.dl2  -11.1726     8.3523  -1.338   0.2058
## edu.dl2          -0.8410     1.1110  -0.757   0.4637
## salud.dl2        -2.1297     1.3656  -1.560   0.1448
## rec_imp.dl2      -0.4229     0.3627  -1.166   0.2663
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7391 on 12 degrees of freedom
## Multiple R-squared:  0.6614, Adjusted R-squared:  0.3511
## F-statistic: 2.131 on 11 and 12 DF,  p-value: 0.1047
```

```
mysum <- summary(M1.jorls$rlm)
screenreg(list(mysum[[1]], mysum[[2]],mysum[[3]],mysum[[4]]))
```

```
##
## ==========================================================
##                 Model 1   Model 2      Model 3    Model 4
## ----------------------------------------------------------
## ect1            -0.31       4.33 **     0.00        8.02
##                 (0.15)     (1.02)      (0.50)      (4.08)
## ect2             0.05      -0.87 ***    0.03       -1.52
##                 (0.03)     (0.20)      (0.10)      (0.80)
## constant         8.66    -126.84 **     1.14     -231.72
##                 (4.37)    (29.73)     (14.47)    (118.54)
## ln_pib_cte.dl1  -0.39      -5.17 *      0.53       -4.98
##                 (0.30)     (2.08)      (1.01)      (8.28)
## edu.dl1         -0.13 **   -0.00        0.39 **     0.86
##                 (0.04)     (0.26)      (0.13)      (1.05)
## salud.dl1       -0.06       0.20        0.55 **    -0.21
##                 (0.05)     (0.34)      (0.16)      (1.34)
## rec_imp.dl1     -0.04 *    -0.15        0.01       -0.21
##                 (0.01)     (0.09)      (0.05)      (0.37)
## ln_pib_cte.dl2  -0.34      -2.31       -0.82      -11.17
##                 (0.31)     (2.09)      (1.02)      (8.35)
## edu.dl2         -0.13 **    0.27       -0.02       -0.84
##                 (0.04)     (0.28)      (0.14)      (1.11)
## salud.dl2       -0.06       0.42       -0.63 **    -2.13
##                 (0.05)     (0.34)      (0.17)      (1.37)
## rec_imp.dl2     -0.02      -0.06        0.12 *     -0.42
##                 (0.01)     (0.09)      (0.04)      (0.36)
## ----------------------------------------------------------
## R^2              0.76       0.77        0.80        0.66
## Adj. R^2         0.53       0.55        0.62        0.35
## Num. obs.       23         23          23          23
## RMSE             0.03       0.19        0.09        0.74
## ==========================================================
## *** p < 0.001, ** p < 0.01, * p < 0.05
```

Finalmente obsevemos nuestro modelo ajustado vs los datos recopilados:




```r
vecm.level <- vec2var(M1,r=rel_coint_aux)
arch.test(vecm.level)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 180, df = 500, p-value = 1
```

```r
normality.test(vecm.level)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 56.884, df = 8, p-value = 1.897e-09
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 22.777, df = 4, p-value = 0.0001403
##
##
## $Kurtosis
```

```
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 34.107, df = 4, p-value = 7.084e-07
```

```
serial.test(vecm.level)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object vecm.level
## Chi-squared = 152.91, df = 212, p-value = 0.9992
```

```
predict(vecm.level)
```

```
## $ln_pib_cte
##           fcst    lower    upper         CI
##  [1,] 30.65166 30.61311 30.69020 0.03854448
##  [2,] 30.73744 30.68323 30.79165 0.05420936
##  [3,] 30.80853 30.72688 30.89018 0.08164980
##  [4,] 30.91224 30.79847 31.02602 0.11377775
##  [5,] 31.01563 30.86893 31.16234 0.14670673
##  [6,] 31.13453 30.94791 31.32116 0.18662691
##  [7,] 31.29073 31.05513 31.52633 0.23560235
##  [8,] 31.44581 31.14882 31.74279 0.29698669
##  [9,] 31.61533 31.24318 31.98748 0.37214990
## [10,] 31.81768 31.35942 32.27593 0.45825545
##
## $edu
##            fcst     lower     upper        CI
##  [1,]  5.952567  5.690167  6.214967 0.2624001
##  [2,]  6.251284  5.978034  6.524535 0.2732505
##  [3,]  6.694731  6.402774  6.986688 0.2919571
##  [4,]  7.468030  7.103180  7.832879 0.3648497
##  [5,]  8.091748  7.540673  8.642822 0.5510743
##  [6,]  8.796180  7.953029  9.639331 0.8431512
##  [7,]  9.648249  8.474239 10.822260 1.1740102
##  [8,] 10.482563  8.949807 12.015318 1.5327552
##  [9,] 11.423143  9.478370 13.367917 1.9447738
## [10,] 12.562490 10.153953 14.971026 2.4085363
##
## $salud
##             fcst      lower     upper        CI
##  [1,]  2.5319722  2.4042662 2.6596783 0.1277060
##  [2,]  2.3873472  2.0966550 2.6780393 0.2906922
##  [3,]  1.9894310  1.5787632 2.4000988 0.4106678
##  [4,]  1.5495305  1.0037610 2.0952999 0.5457695
##  [5,]  1.2621912  0.5034159 2.0209665 0.7587753
##  [6,]  0.8028485 -0.1996667 1.8053636 1.0025151
##  [7,]  0.2226202 -1.0380177 1.4832582 1.2606379
##  [8,] -0.4046979 -1.9772512 1.1678554 1.5725533
##  [9,] -1.2003118 -3.1520752 0.7514515 1.9517634
## [10,] -2.0650463 -4.4600761 0.3299835 2.3950298
##
```

```
## $rec_imp
##           fcst    lower    upper       CI
##  [1,] 12.72629 11.67993 13.77264 1.046355
##  [2,] 13.69513 12.46288 14.92739 1.232254
##  [3,] 14.17579 12.69055 15.66103 1.485240
##  [4,] 14.57248 12.68543 16.45954 1.887053
##  [5,] 15.75117 13.50975 17.99259 2.241423
##  [6,] 15.94917 13.36523 18.53311 2.583943
##  [7,] 15.96701 12.97786 18.95617 2.989155
##  [8,] 16.81759 13.52664 20.10855 3.290959
##  [9,] 17.49025 13.93233 21.04818 3.557921
## [10,] 18.31136 14.38909 22.23363 3.922271
```

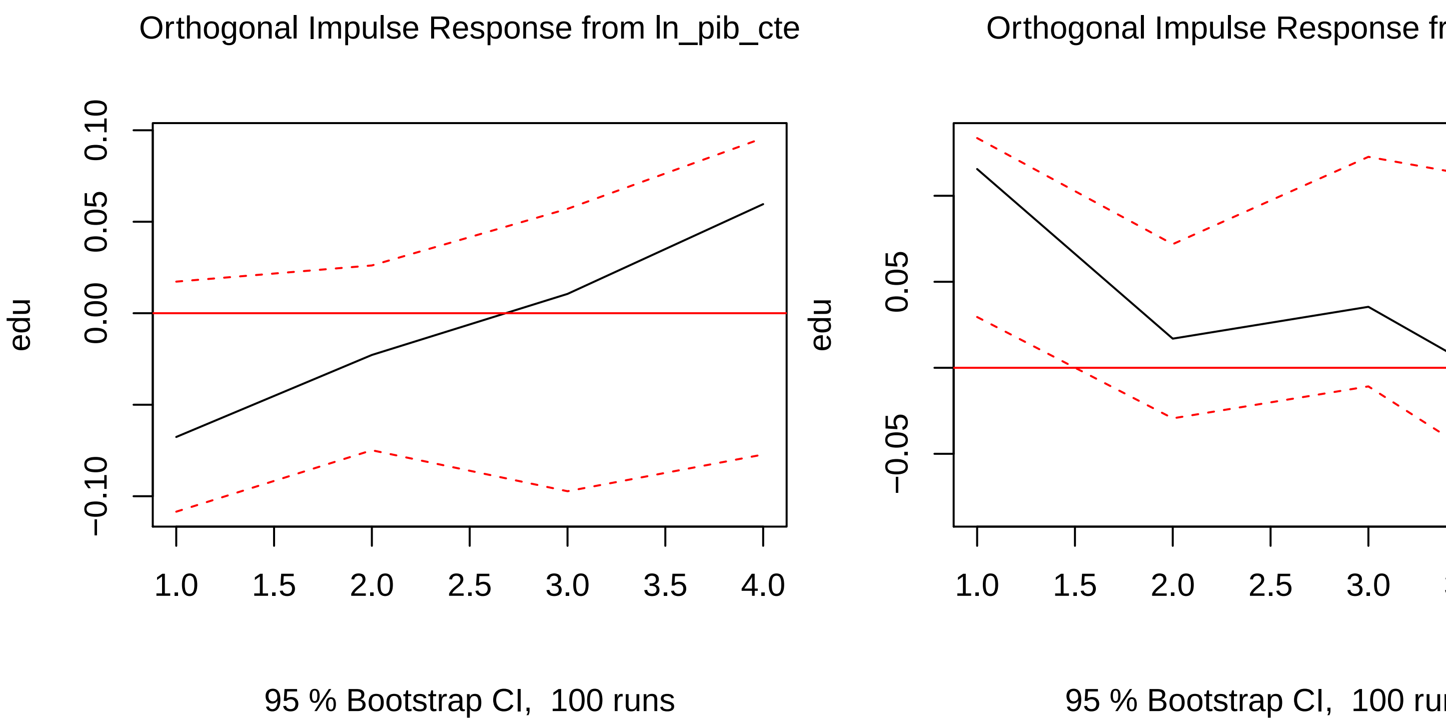
```r
irf(vecm.level,boot=FALSE)
```

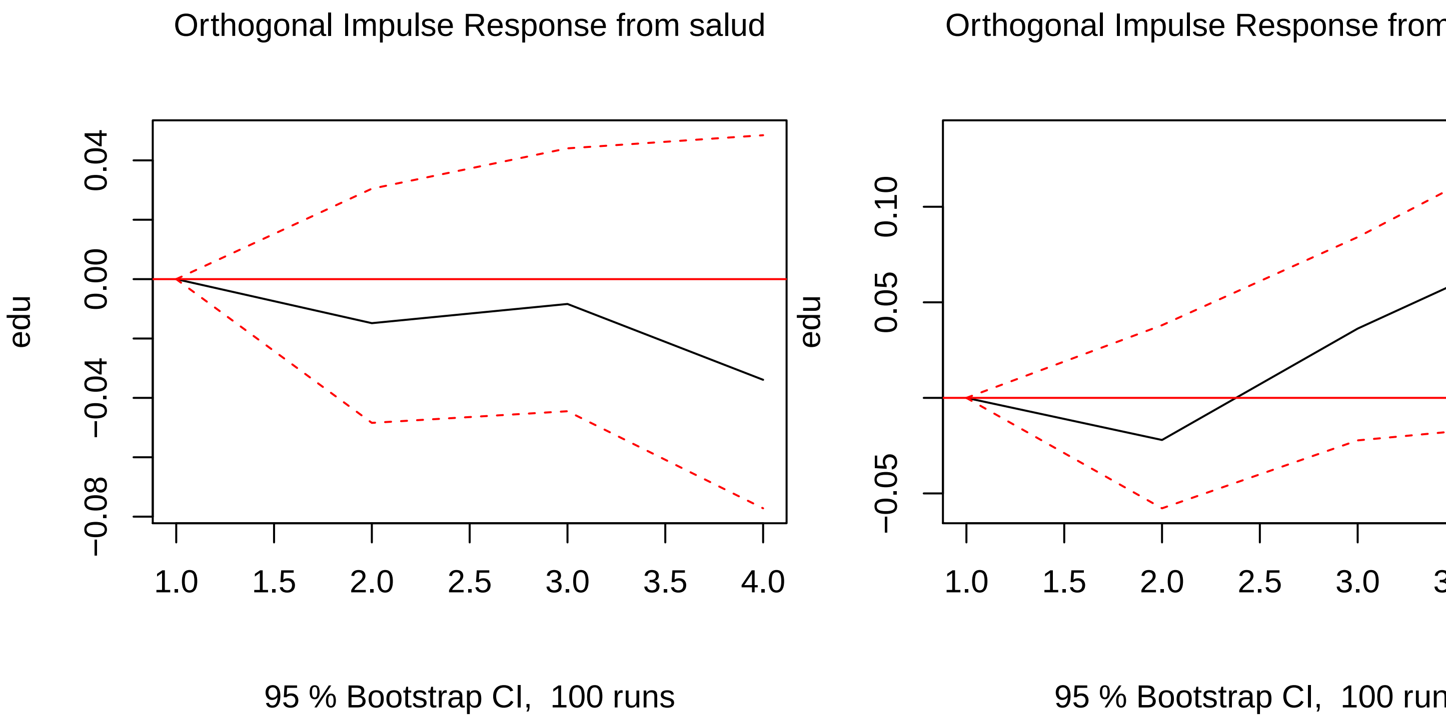
```
##
## Impulse response coefficients
## $ln_pib_cte
##         ln_pib_cte          edu      salud      rec_imp
##  [1,]  0.019665915 -0.067621848 0.01924698 -0.164073922
##  [2,]  0.004987483 -0.022810768 0.03629142 -0.014647077
##  [3,] -0.001266895  0.010560782 0.01663196 -0.015724262
##  [4,] -0.007542540  0.059610301 0.03796728  0.041916649
##  [5,] -0.008563506  0.005717826 0.08233467  0.003086728
##  [6,] -0.007269197 -0.019204112 0.08997103 -0.206365290
##  [7,] -0.006167527 -0.061440771 0.09211014 -0.262402508
##  [8,] -0.012247377 -0.096082335 0.09412637 -0.181800170
##  [9,] -0.016657566 -0.118631753 0.10592125 -0.133533343
## [10,] -0.021057324 -0.131840575 0.14323682 -0.077800842
## [11,] -0.026586636 -0.153937579 0.18152875 -0.102727279
##
## $edu
##         ln_pib_cte         edu       salud     rec_imp
##  [1,]  0.000000000  0.11554723 -0.01076395  0.01313726
##  [2,] -0.008677824  0.01695083  0.02951149 -0.05964126
##  [3,] -0.007232773  0.03545908  0.02561387 -0.27769284
##  [4,] -0.002820246 -0.02918105  0.05039765 -0.18070533
##  [5,] -0.011979173 -0.07850716  0.04912751 -0.19332255
##  [6,] -0.012088293 -0.07968758  0.04985533 -0.25011906
##  [7,] -0.017800494 -0.09571792  0.09763363 -0.05157325
##  [8,] -0.025974625 -0.12741414  0.12473210 -0.10862066
##  [9,] -0.028859403 -0.12370039  0.14567050 -0.26687425
## [10,] -0.034575666 -0.16802731  0.19026744 -0.22158197
## [11,] -0.043061727 -0.22479755  0.21310922 -0.29501369
##
## $salud
##         ln_pib_cte          edu      salud     rec_imp
##  [1,]  0.000000000  0.000000000 0.06131206 -0.01139146
##  [2,] -0.005135646 -0.014829485 0.10270615 -0.05570572
##  [3,] -0.010053731 -0.008364414 0.09490898 -0.23765786
##  [4,] -0.014718601 -0.033908309 0.10729118 -0.25307568
##  [5,] -0.020068377 -0.094987805 0.13726730 -0.19649183
##  [6,] -0.023043167 -0.147978574 0.16344636 -0.25131092
##  [7,] -0.027484443 -0.186526490 0.19865328 -0.26714126
##  [8,] -0.036311149 -0.223500603 0.23605198 -0.24610289
```

```
##  [9,] -0.046019550 -0.254893585 0.27261871 -0.28099290
## [10,] -0.055187713 -0.299110975 0.32429142 -0.31327173
## [11,] -0.065219083 -0.365141602 0.38280706 -0.36346418
##
## $rec_imp
##        ln_pib_cte         edu      salud   rec_imp
##  [1,] 0.00000000  0.00000000  0.0000000 0.5077286
##  [2,] 0.01586447 -0.02203174 -0.0708190 0.3215490
##  [3,] 0.02855658  0.03624250 -0.1093815 0.2124226
##  [4,] 0.03678278  0.08311668 -0.1347014 0.5042615
##  [5,] 0.04016589  0.17083033 -0.2104784 0.5521372
##  [6,] 0.05228791  0.27819120 -0.2728708 0.5118057
##  [7,] 0.06536654  0.35497321 -0.3075557 0.6670783
##  [8,] 0.07979532  0.42114120 -0.3871802 0.6228804
##  [9,] 0.09931970  0.52786266 -0.4910633 0.5549350
## [10,] 0.11802035  0.62487884 -0.5828324 0.7457755
## [11,] 0.13663790  0.73839863 -0.6968783 0.8720783
```

```
#fevd(vecm.level)
```

```
plot(irf(vecm.level,n.ahead=3,response='edu',boot=TRUE))
```






```
BIC(vec2var(M1,r=rel_coint_aux))
```

```
## [1] -15.47004
```

Chapter 13

Interpolación Serie Educación

Veamos la serie original

```
library(imputeTS)
library(forecast)
serie_edu_unfill <- read_excel("Datos/Series Tesis Recolección.xlsx",sheet ="Gasto en Educación", na =

serie_edu_unfill <- serie_edu_unfill[1:26,]
serie_edu_unfill
```

```
## # A tibble: 26 x 2
##    Fecha Gasto_Educacion_PorcGDP
##    <dbl>                   <dbl>
##  1  1989                    2.27
##  2  1990                    2.31
##  3  1991                    2.54
##  4  1992                    3
##  5  1993                   NA
##  6  1994                    3.65
##  7  1995                    3.87
##  8  1996                   NA
##  9  1997                   NA
## 10  1998                    3.53
## # ... with 16 more rows
```

```
aux <- as.ts(serie_edu_unfill$Gasto_Educacion_PorcGDP)
statsNA(aux)
```

```
## [1] "Length of time series:"
## [1] 26
## [1] "-------------------------"
## [1] "Number of Missing Values:"
## [1] 3
## [1] "-------------------------"
## [1] "Percentage of Missing Values:"
## [1] "11.5%"
## [1] "-------------------------"
## [1] "Stats for Bins"
## [1] "  Bin 1 (7 values from 1 to 7) :      1 NAs (14.3%)"
## [1] "  Bin 2 (7 values from 8 to 14) :      2 NAs (28.6%)"
```

```
## [1] "  Bin 3 (7 values from 15 to 21) :      0 NAs (0%)"
## [1] "  Bin 4 (5 values from 22 to 26) :      0 NAs (0%)"
## [1] "-------------------------"
## [1] "Longest NA gap (series of consecutive NAs)"
## [1] "2 in a row"
## [1] "-------------------------"
## [1] "Most frequent gap size (series of consecutive NA series)"
## [1] "2 NA in a row (occuring 1 times)"
## [1] "-------------------------"
## [1] "Gap size accounting for most NAs"
## [1] "2 NA in a row (occuring 1 times, making up for overall 2 NAs)"
## [1] "-------------------------"
## [1] "Overview NA series"
## [1] "  1 NA in a row: 1 times"
## [1] "  2 NA in a row: 1 times"
```

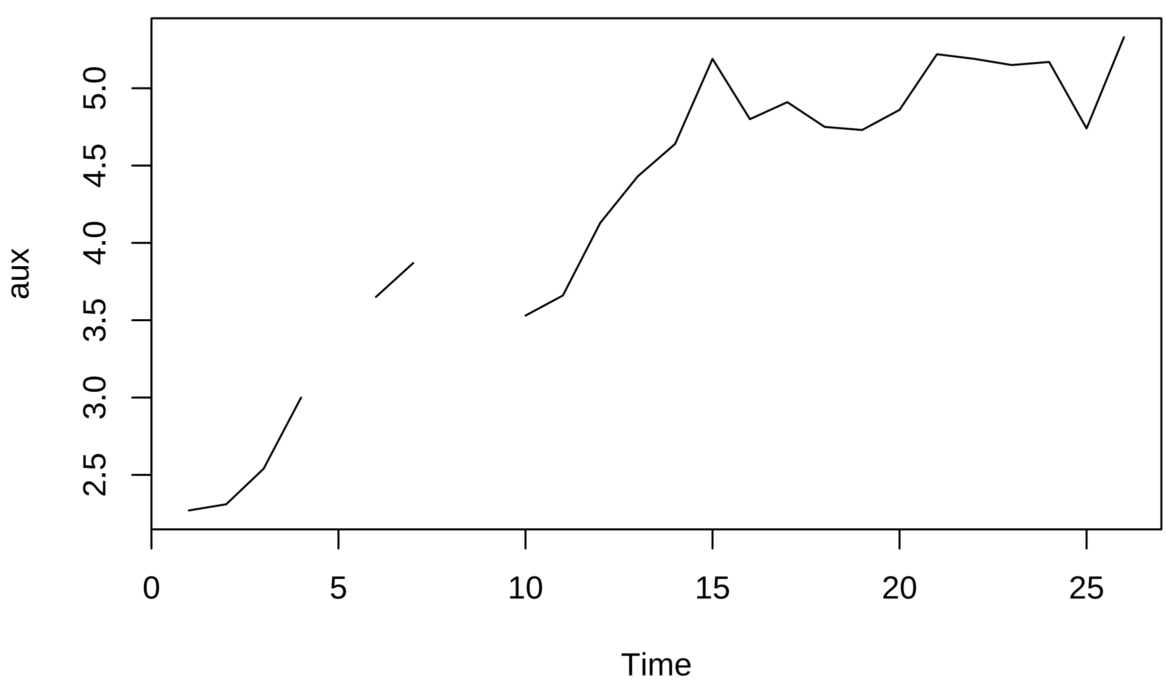
interpolamos los vacios entre las series

```r
aux2 <- na.interp(aux)
aux2
```

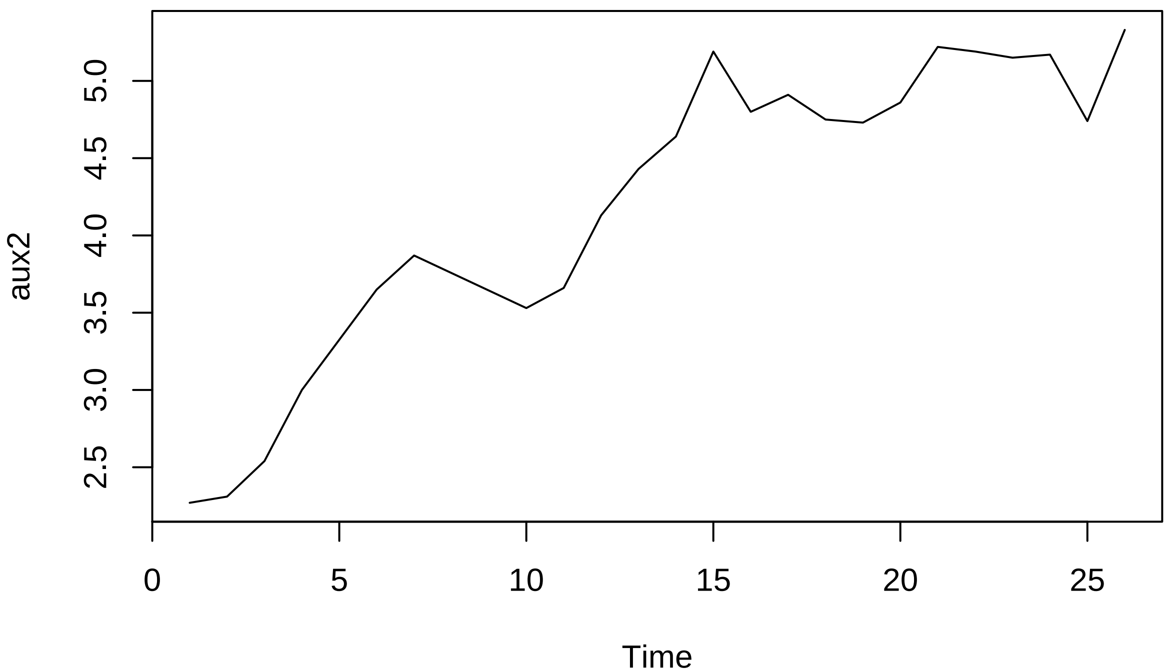
```
## Time Series:
## Start = 1
## End = 26
## Frequency = 1
##  [1] 2.270000 2.310000 2.540000 3.000000 3.325000 3.650000 3.870000
##  [8] 3.756667 3.643333 3.530000 3.660000 4.130000 4.430000 4.640000
## [15] 5.190000 4.800000 4.910000 4.750000 4.730000 4.860000 5.220000
## [22] 5.190000 5.150000 5.170000 4.740000 5.330000
```

graficamos las series

```r
plot(aux)
```



```r
plot(aux2)
```

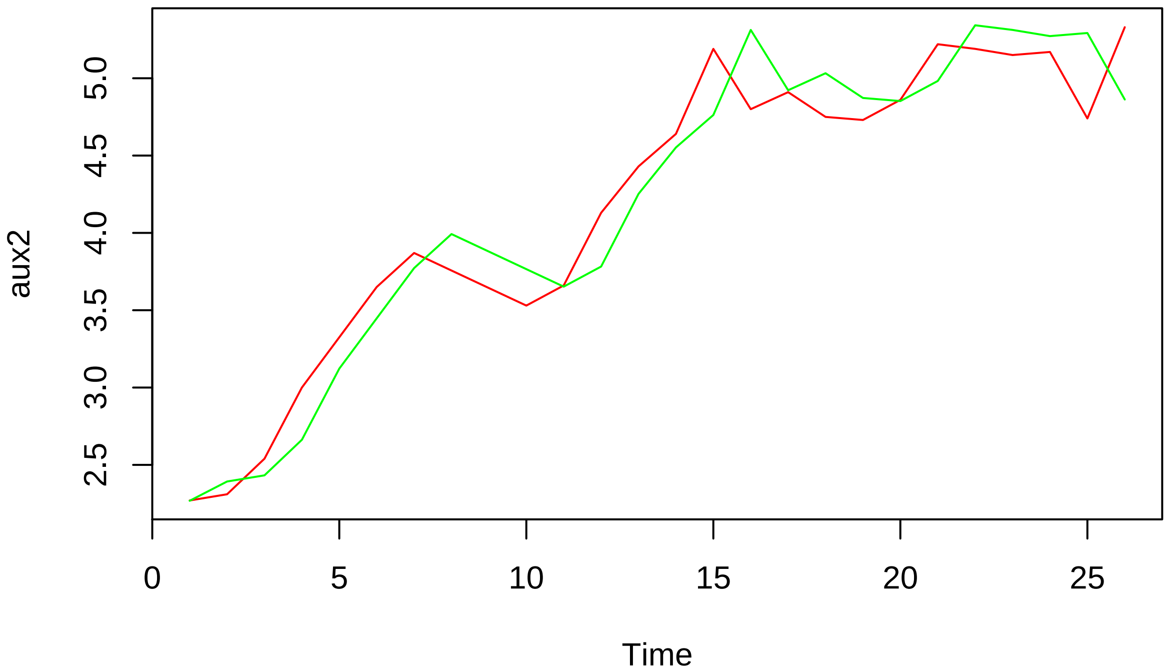


Ajustamos un modelo arima para pronosticar los siguietnes dos puntos

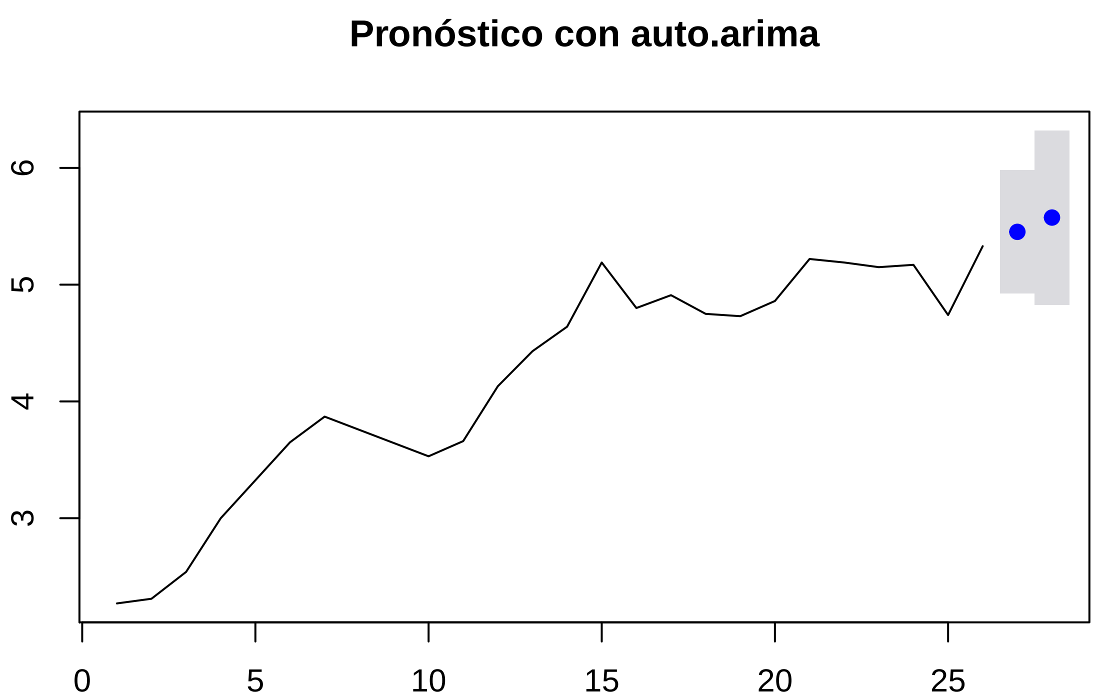
```r
modelo<-auto.arima(aux2)
summary(modelo)
```

```
## Series: aux2
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.1224
## s.e.  0.0527
##
## sigma^2 estimated as 0.07236:  log likelihood=-2.14
## AIC=8.28   AICc=8.82   BIC=10.71
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 8.259996e-05 0.2584528 0.2084518 0.1304937 4.945747 0.8862748
##                      ACF1
## Training set -0.002651446
```

```r
plot(aux2,type="l",col="red")
lines(as.ts(modelo$fitted),col="green")
```




```
pronostico<- forecast(modelo,2,level=95)

plot(pronostico,main="Pronóstico con auto.arima")
```




```
matriz.pronosticos <-data.frame(pronostico$mean,pronostico$lower,pronostico$upper)
matriz.pronosticos
```

```
##   pronostico.mean     X95.   X95..1
## 1          5.4524 4.925158 5.979642
## 2          5.5748 4.829167 6.320433
```

regresamos el prónostico al dataframe

13.1 version 2 del pronositico

```
mean(aux2)
```

```
## [1] 4.184423
```

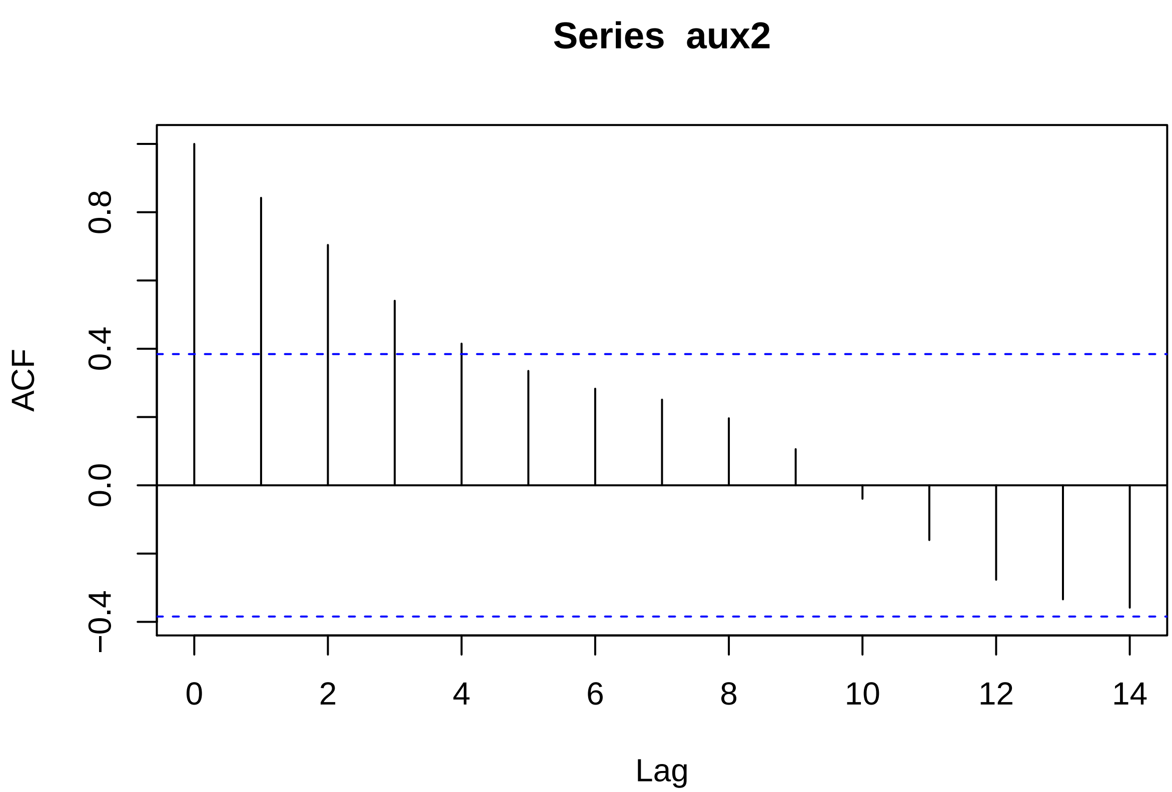
```
mean(diff(aux2))
```

```
## [1] 0.1224
```

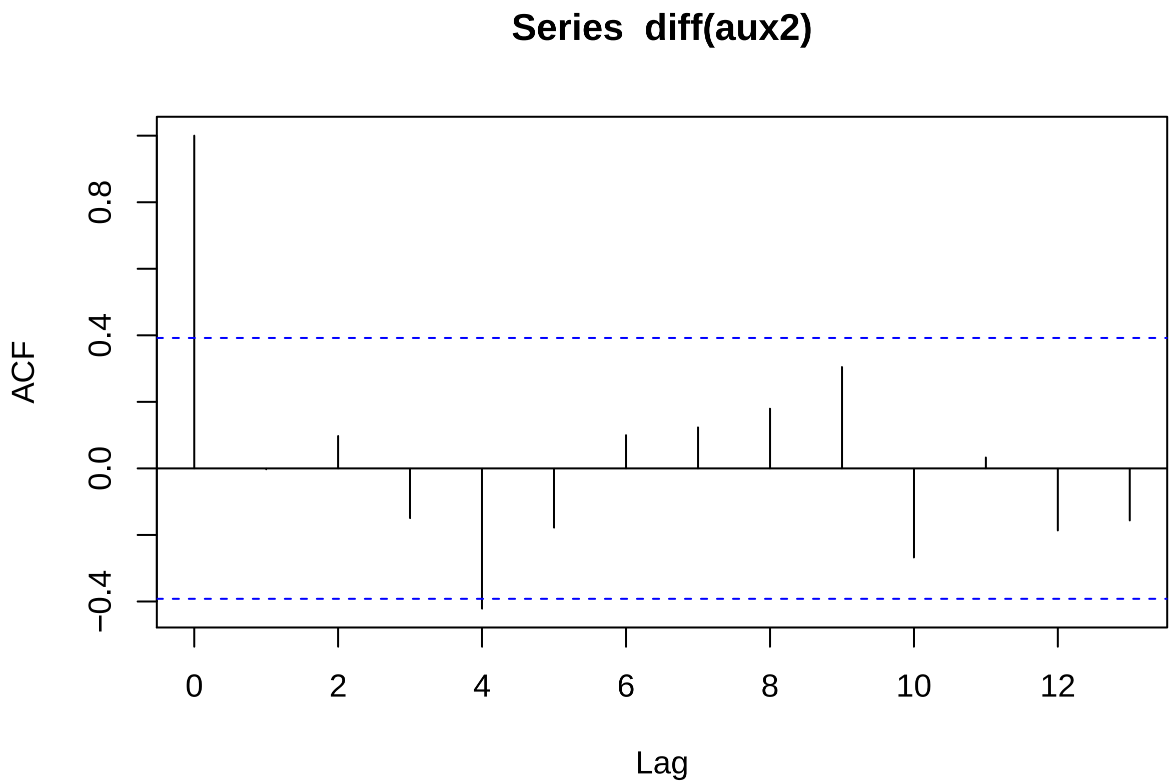
```
mean(diff(diff(aux2)))
```

```
## [1] 0.02291667
```

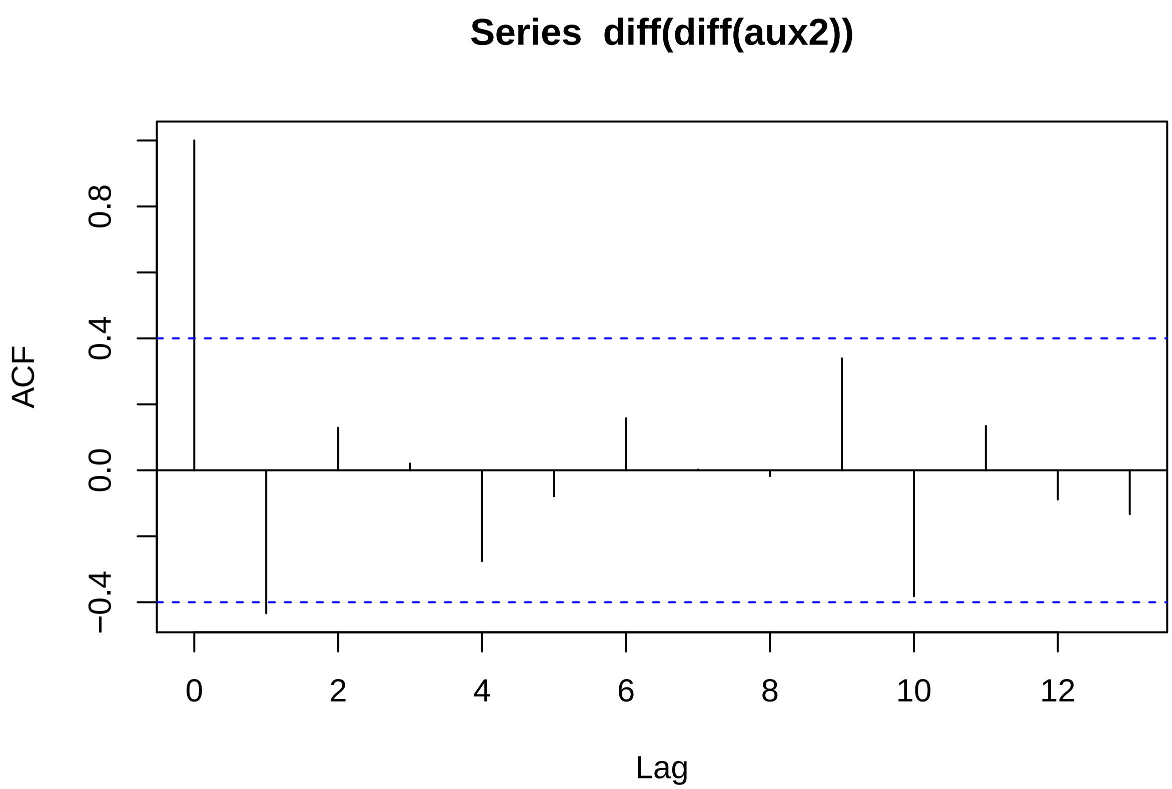
```
acf(aux2)
```




```
acf(diff(aux2))
```




```r
acf(diff(diff(aux2)))
```




```r
var(aux2)
```

```
## [1] 0.8982895
```

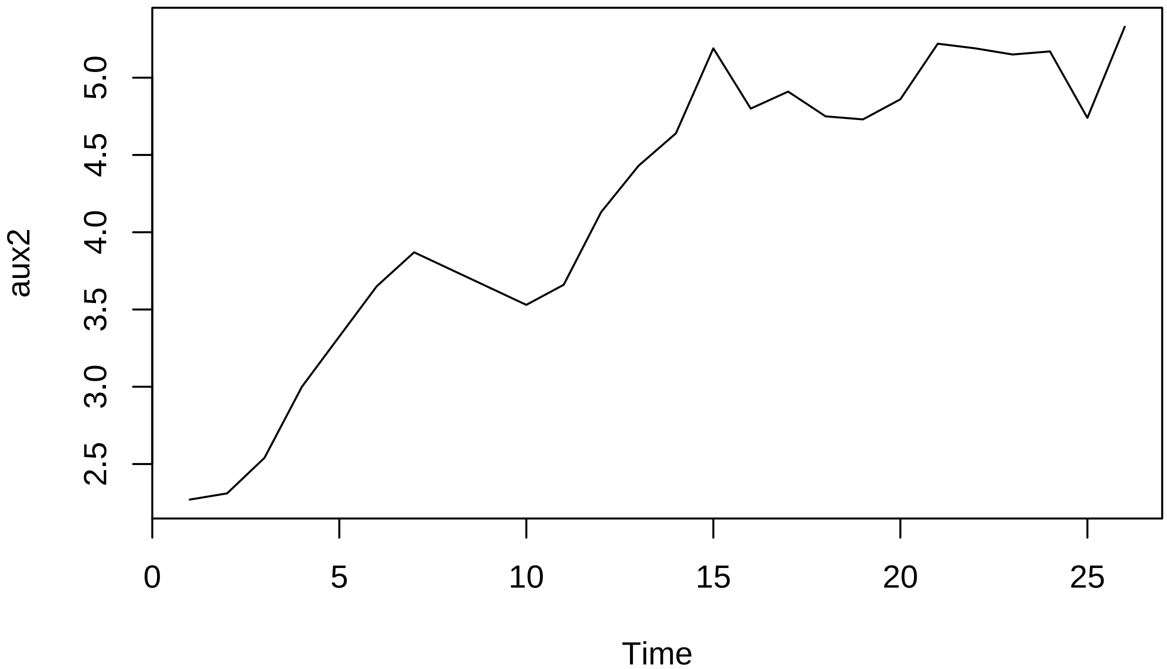
```
var(diff(aux2))
```

```
## [1] 0.07236414
```

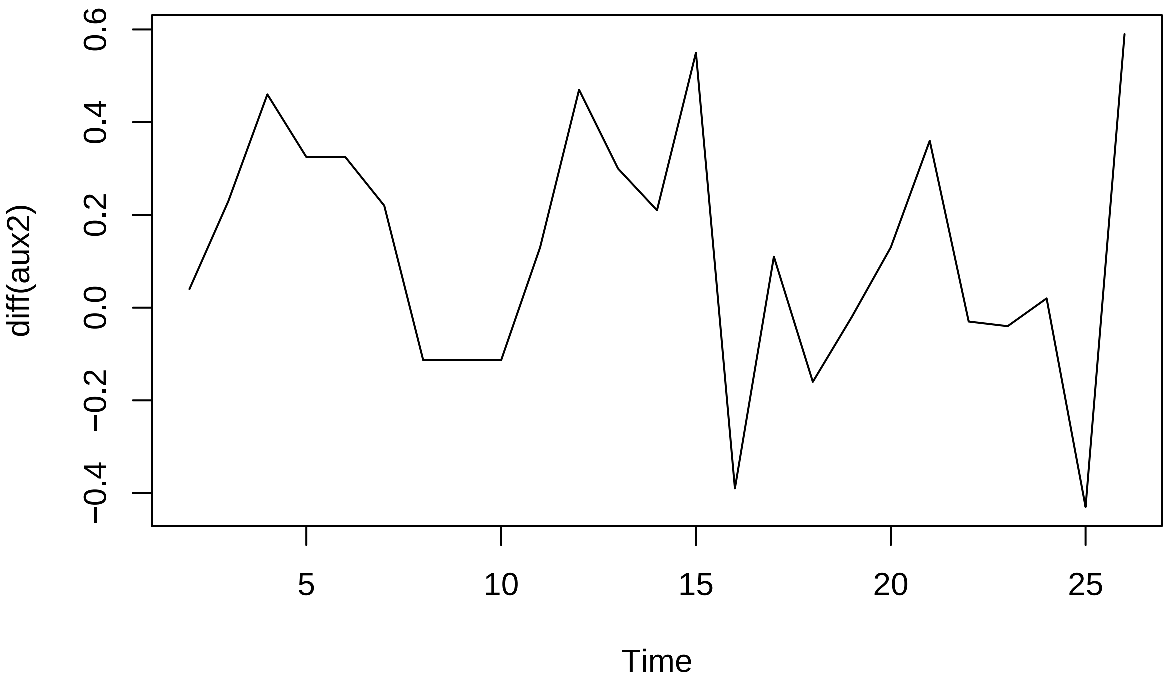
```
var(diff(diff(aux2)))
```

```
## [1] 0.1410595
```

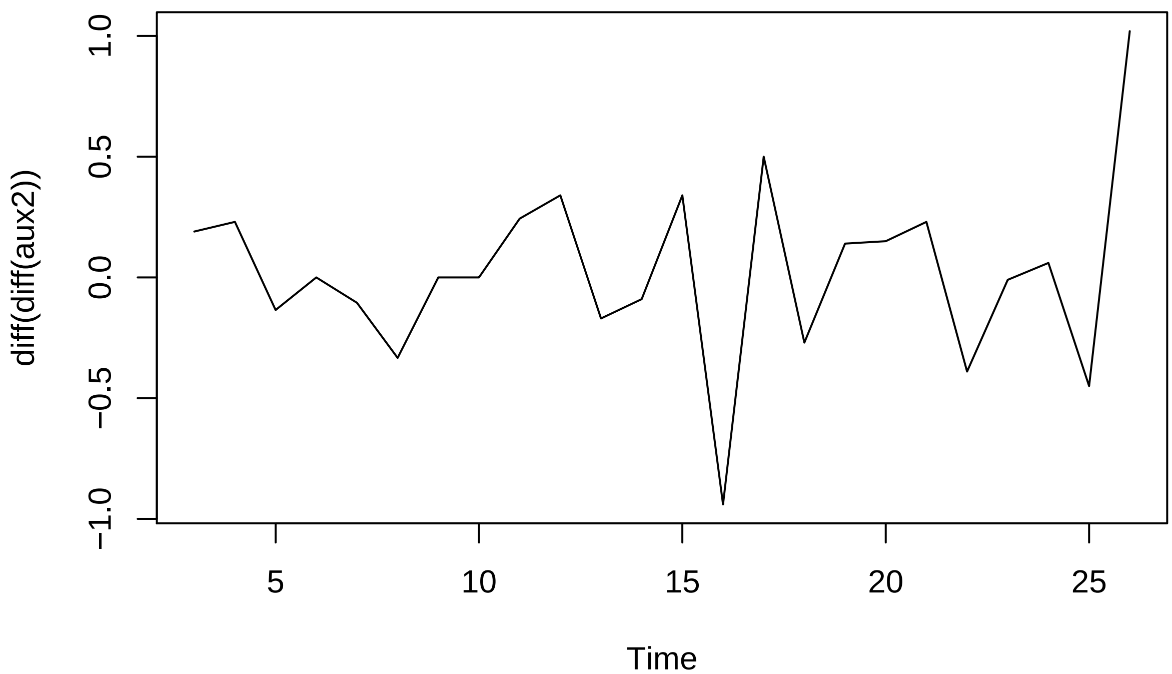
```
plot(aux2)
```




```
plot(diff(aux2))
```

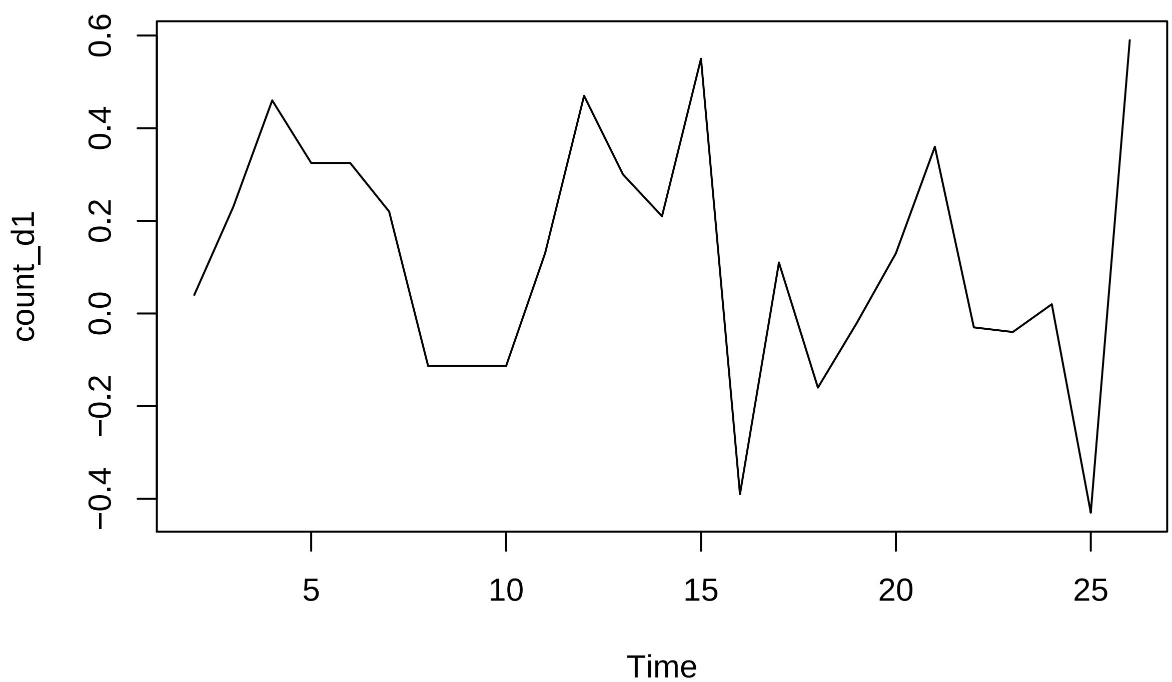


```
plot(diff(diff(aux2)))
```



la serie parece ser I(1).

```
library('tseries')
count_d1 = diff(aux2, differences = 1)
plot(count_d1)
```




```
adf.test(count_d1, alternative = "stationary")
```

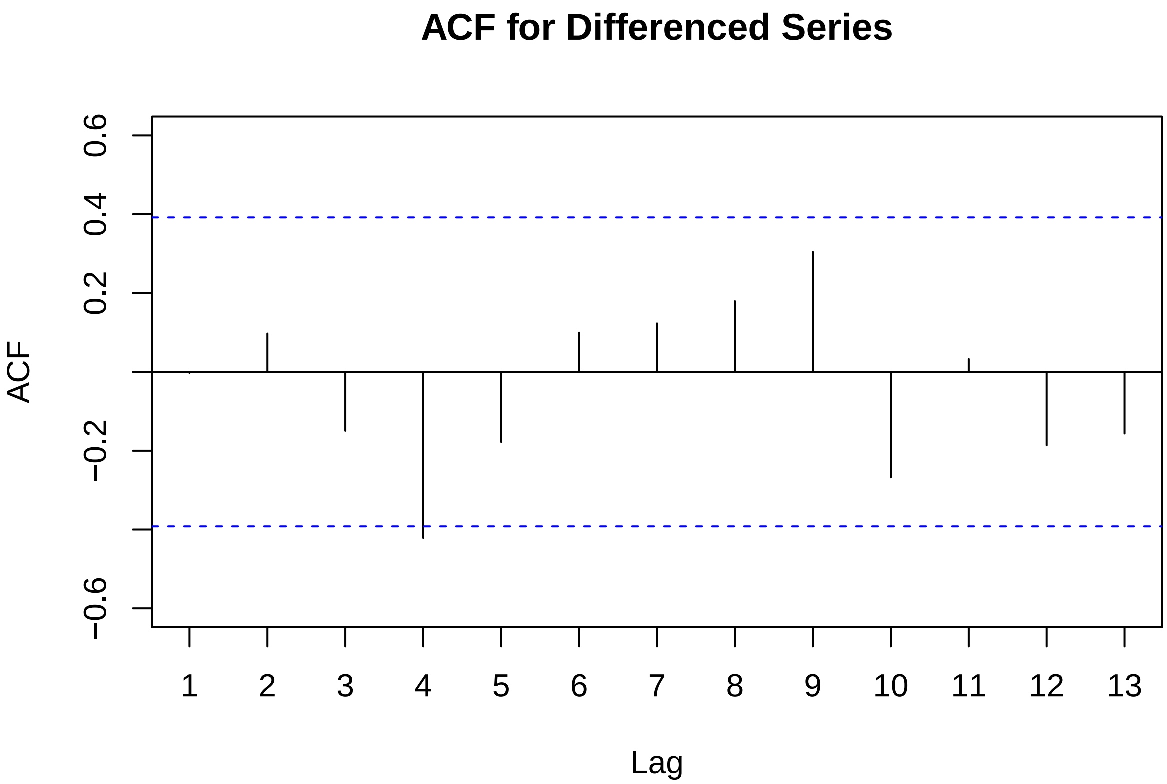
```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  count_d1
## Dickey-Fuller = -2.8183, Lag order = 2, p-value = 0.2606
## alternative hypothesis: stationary
```

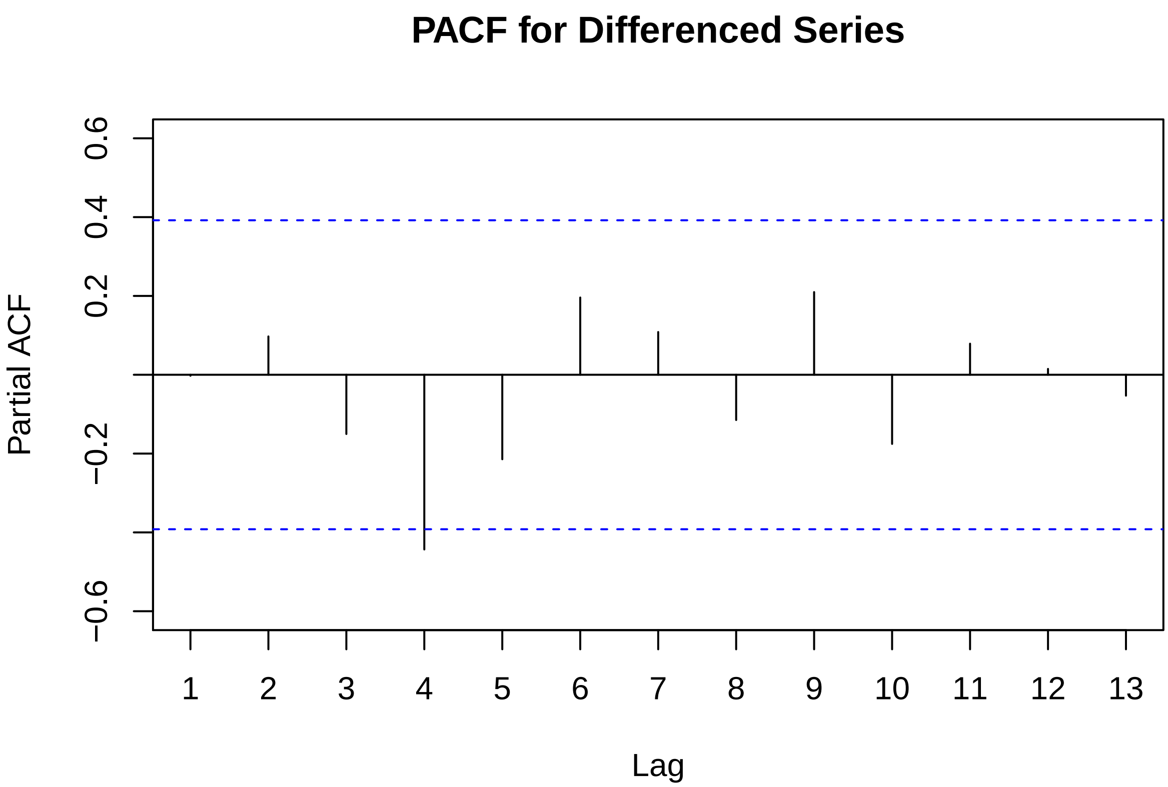
```r
summary(ur.df(count_d1,lags=0,type='drift'))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55599 -0.18430  0.00385  0.19965  0.46316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.12603    0.06187   2.037 0.053855 .
## z.lag.1     -1.00190    0.22826  -4.389 0.000233 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2804 on 22 degrees of freedom
## Multiple R-squared:  0.4669, Adjusted R-squared:  0.4426
## F-statistic: 19.27 on 1 and 22 DF,  p-value: 0.0002331
##
##
## Value of test-statistic is: -4.3893 9.7131
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.75 -3.00 -2.63
## phi1  7.88  5.18  4.12
```

```r
Acf(count_d1, main='ACF for Differenced Series')
```




```r
Pacf(count_d1, main='PACF for Differenced Series')
```

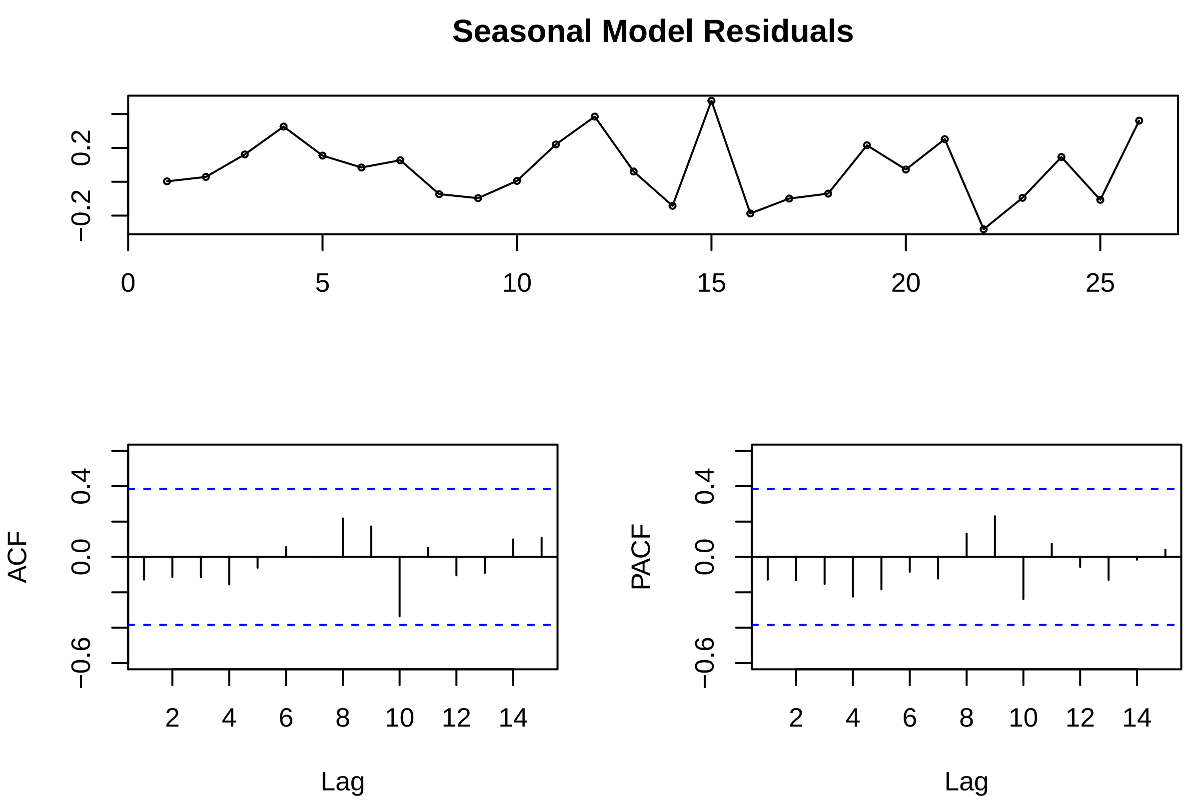



```r
fit2 = arima(aux2, order=c(4,1,4))
```

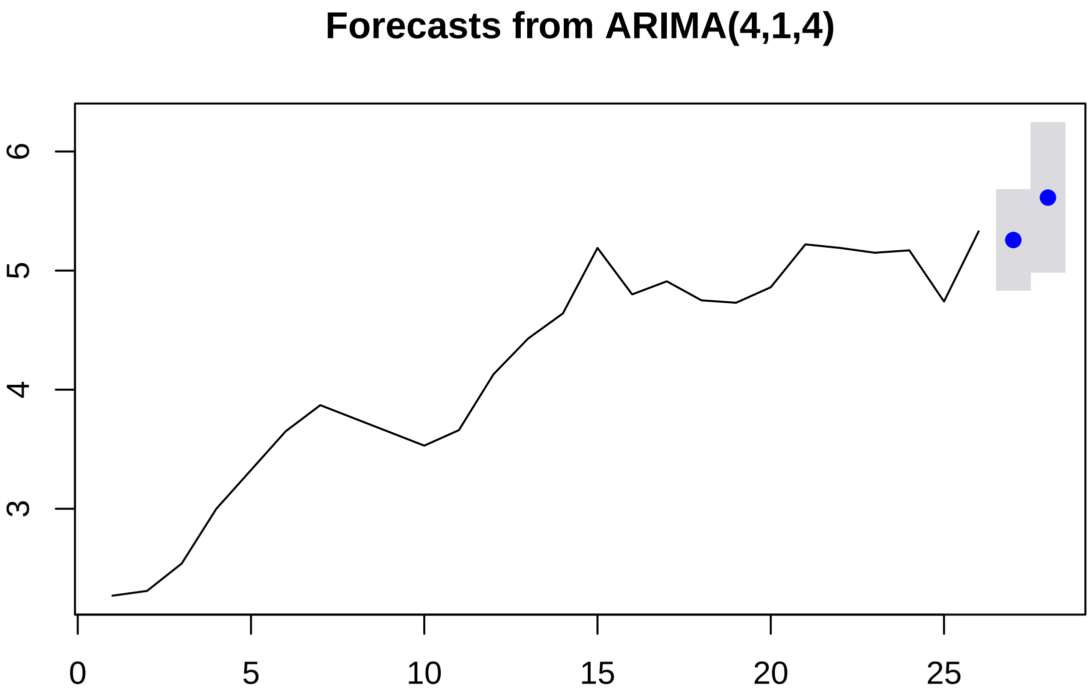
```r
fit2
```

```
##
## Call:
## arima(x = aux2, order = c(4, 1, 4))
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ma1     ma2     ma3      ma4
##       -0.3674  0.2608  -0.2477  -0.1629  0.4902  0.5539  0.7730  -0.2907
## s.e.   0.6030  0.4189   0.2670   0.2572  0.5967  0.6155  0.6205   0.6338
##
## sigma^2 estimated as 0.04236:  log likelihood = 0.84,  aic = 16.32
```

```r
tsdisplay(residuals(fit2), lag.max=15, main='Seasonal Model Residuals')
```




```r
fcast <- forecast(fit2, h=2,level=95)
plot(fcast)
```




```
matriz.pronosticos <-data.frame(fcast$mean,fcast$lower,fcast$upper)
matriz.pronosticos
```

```
##   fcast.mean     X95.    X95..1
## 1   5.256603 4.831780 5.681426
## 2   5.612957 4.982007 6.243908
```

Chapter 14

Anexos

Anexos

14.1 Descripción de las series

Code	Indicator Name
SP.POP.TOTL	Population, total
SL.UEM.TOTL.NE.ZS	Unemployment, total (% of total labor force) (national estimate)
NY.GDP.MKTP.CN	GDP (current LCU)

Code	Indicator Name	Long definition	Source
SP.POP.TOTL	Population, total	Total population is based on the de facto definition of population, which counts all residents regardless of legal status or citizenship. The values shown are midyear estimates.	(1) United Nations Population Division. World Population Prospects, (2) Census reports and other statistical publications from national statistical offices, (3) Eurostat: Demographic Statistics, (4) United Nations Statistical Division. Population and Vital Statistics Report (various years), (5) U.S. Census Bureau: International Database, and (6) Secretariat of the Pacific Community: Statistics and Demog-

Code	Indicator Name	Long definition	Source
SL.UEM.TOTL.NE.ZS	Unemployment, total (% of total labor force) (national estimate)	Unemployment refers to the share of the labor force that is without work but available for and seeking employment. Definitions of labor force and unemployment differ by country.	International Labour Organization, ILOSTAT database. Data retrieved in March 2017.
SH.XPD.PUBL.ZS	Health expenditure, public (% of GDP)	Public health expenditure consists of recurrent and capital spending from government (central and local) budgets, external borrowings and grants (including donations from international agencies and nongovernmental organizations), and social (or compulsory) health insurance funds.	World Health Organization Global Health Expenditure database (see http://apps.who.int/nha/database for the most recent updates).
NY.GDP.MKTP.CN	GDP (current LCU)	GDP at purchaser's prices is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. It is calculated without making deductions for depreciation of fabricated assets or for depletion and degradation of natural resources. Data are in current local currency.	World Bank national accounts data, and OECD National Accounts data files.

14.2 Bibliografia

http://otexts.org/fpp2/intro.html

http://www.pfaffikus.de/files/conf/user/useR2008.pdf

http://www.mexicomaxico.org/Voto/PIBMex.htm https://www.mexicoevalua.org/wp-content/uploads/2016/05/MEX_EVA-INHOUS-GASTO_SALUD-LOW.pdf

14.3 Dudas