DATOS MACROECONOMICOS DE FILIPINAS

By: Ael-Dev

Diccionario de variables:

- date: Representa la fecha en la que se registraron los datos.
- real_gdp_growth: Indica el crecimiento real del Producto Interno Bruto (PIB).
- psei: Corresponde al índice compuesto de la Bolsa de Filipinas.
- bsp_rrp: Refiere a la tasa de interés de referencia del Banco Central de Filipinas.
- unem: Representa la tasa de desempleo en Filipinas.

Leer dataset

```
library(readr)

data <- read_csv('https://raw.githubusercontent.com/ecabestadistica/series-temporales-multivariantes/master/2.%20Casos%20de%20estudio%20en%20R/C.%20Datos%20macroecon%C3%B3micos%20de%20Filipinas/SampleVAR.csv',
                 col_types = cols(
                  #                date = col_double(),
                                  real_gdp_growth = col_double(),
                                  psei = col_double(),
                                  bsp_rrp = col_double(),
                                  unem = col_double()
                                  )
                 )
class(data)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
head(data)
```

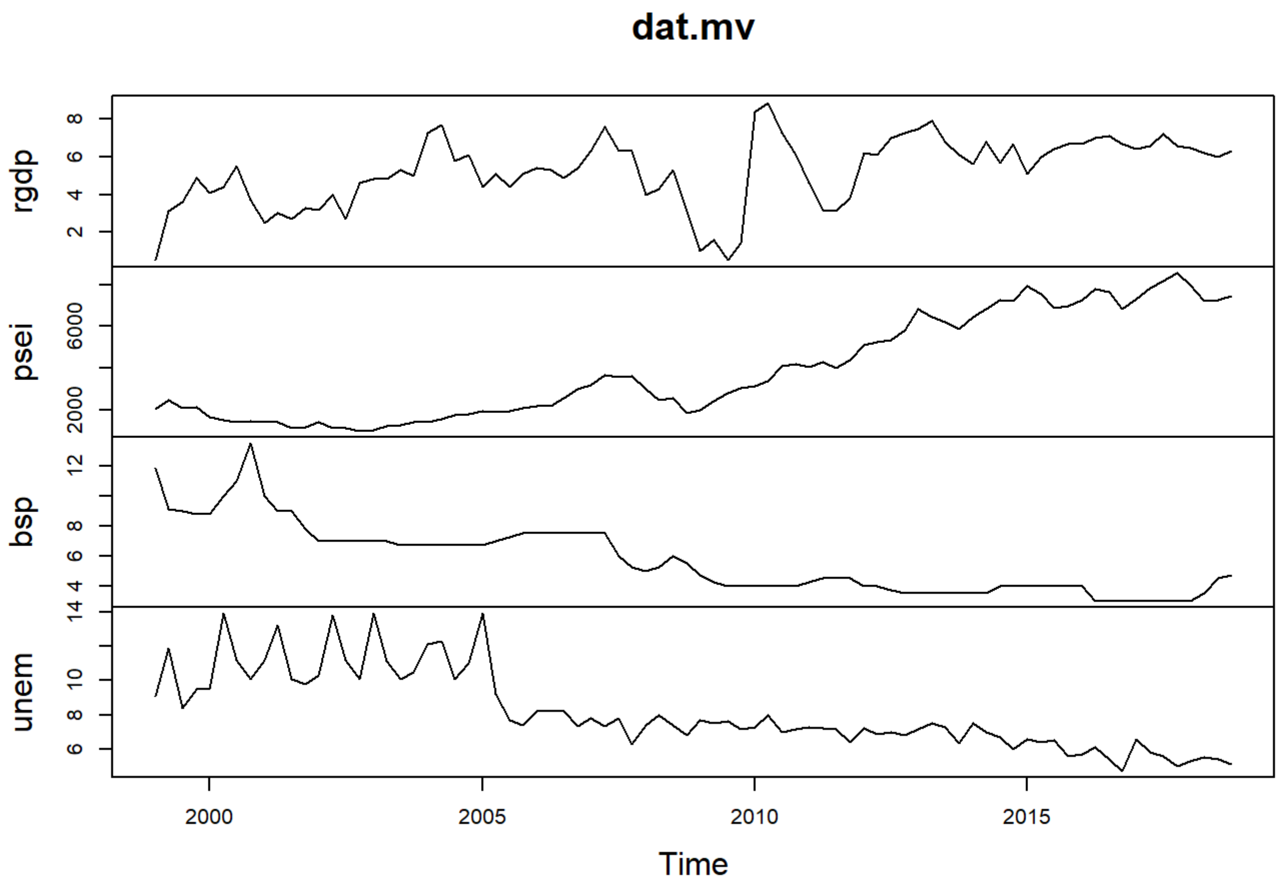
date <chr>	real_gdp_growth <dbl>	psei <dbl>	bsp_rrp <dbl>	unem <dbl>
3/31/99	0.5	2028.21	11.875	9.1
6/30/99	3.1	2486.96	9.125	11.9
9/30/99	3.6	2096.20	9.000	8.4
12/31/99	4.9	2142.97	8.750	9.5
3/31/00	4.1	1681.72	8.750	9.5
6/30/00	4.4	1533.99	10.000	13.9

6 rows

Análisis exploratorio

```
#library(tseries)

# Convertir a objeto ts las dos series
rgdp <- ts(data$real_gdp_growth, start = c(1999,1), frequency = 4)
psei <- ts(data$psei, start = c(1999,1), frequency = 4)
bsp <- ts(data$bsp_rrp, start = c(1999,1), frequency = 4)
unem <- ts(data$unem, start = c(1999,1), frequency = 4)
```

```
# Gráfico con plot:
dat.mv <- cbind(rgdp, psei, bsp, unem) # union
plot(dat.mv )
```



Dividir la serie en conjunto de entrenamiento y de prueba

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
n_obs=10
end=dim(dat.mv)[1]
X_train = dat.mv [1:(end-n_obs),]
X_test = dat.mv [(end-n_obs+1):end,]
dim(X_test)
```

```
## [1] 10  4
```

Prueba de estacionariedad

```
# Check if the 'zoo' package is attached
if ("zoo" %in% search()) {
  # Detach the 'zoo' package
  detach("package:zoo", unload = TRUE)
} else {
  message("Package 'zoo' is not currently attached.")
}
```

```
## Package 'zoo' is not currently attached.
```

```
# Load the 'zoo' package
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
apply(X_train, 2, adf.test) #2 aplicar por columnas
```

```
## $rgdp
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -3.8477, Lag order = 4, p-value = 0.02167
## alternative hypothesis: stationary
##
##
## $psei
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -2.0502, Lag order = 4, p-value = 0.5548
## alternative hypothesis: stationary
##
##
## $bsp
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -2.568, Lag order = 4, p-value = 0.3442
## alternative hypothesis: stationary
##
##
## $unem
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -2.7993, Lag order = 4, p-value = 0.2502
## alternative hypothesis: stationary
```

De los resultados obtenidos, se concluye que la mayoria de las series no son estacionarias, por lo que se sugiere diferenciar.

```
# Diferenciamos
library(MTS)
```

```
## Warning: package 'MTS' was built under R version 4.3.3
```

```
stnry = diffM(X_train)
```

```
# Volviendo a realizar el test:
apply(stnry, 2, adf.test)
```

```
## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value

## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value

## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value

## Warning in FUN(newX[, i], ...): p-value smaller than printed p-value
```

```
## $rgdp
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -5.5754, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $psei
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -4.1583, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $bsp
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -4.8257, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $unem
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -4.1159, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Se observa que todas las series son estacionarias.

MODELO VAR

```
# Identificación del orden del modelo
library(vars)
```

```
## Warning: package 'vars' was built under R version 4.3.3
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: strucchange
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
##
## Attaching package: 'vars'
```

```
## The following object is masked from 'package:MTS':
##
##     VAR
```

```
VARselect(stnry, type = "none", lag.max = 10)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $criteria
##                  1           2           3           4            5
## AIC(n)    11.10654    11.10972    11.11831    11.34442     11.51018
## HQ(n)     11.32647    11.54958    11.77810    12.22413     12.60982
## SC(n)     11.66994    12.23652    12.80851    13.59802     14.32718
## FPE(n) 66660.86239 67268.12567 68962.77048 89328.62356 111509.18430
##                   6            7            8            9           10
## AIC(n)     11.69179     11.96411     11.75616     11.75425     11.70457
## HQ(n)      13.01136     13.50360     13.51559     13.73361     13.90385
## SC(n)      15.07219     15.90791     16.26336     16.82485     17.33857
## FPE(n) 146023.73996 218639.30076 214757.44191 281088.02100 393785.02976
```

Basado en el criterio AIC, la sugerencia que brinda es que se debe considerar 1 retraso(1 orden).

```
# Creando el modelo
var.a <- vars::VAR(stnry,

                   lag.max = 10,

                   ic = "AIC",

                   type = "none")

summary(var.a)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: rgdp, psei, bsp, unem
## Deterministic variables: none
## Sample size: 68
## Log Likelihood: -795.837
## Roots of the characteristic polynomial:
## 0.3569 0.2497 0.1101 0.1101
## Call:
## vars::VAR(y = stnry, type = "none", lag.max = 10, ic = "AIC")
##
##
## Estimation results for equation rgdp:
## =====================================
## rgdp = rgdp.l1 + psei.l1 + bsp.l1 + unem.l1
##
##           Estimate Std. Error t value Pr(>|t|)
## rgdp.l1  0.0076459  0.1219374   0.063    0.950
## psei.l1  0.0005013  0.0004935   1.016    0.314
## bsp.l1  -0.1181970  0.2321487  -0.509    0.612
## unem.l1  0.0070694  0.1106894   0.064    0.949
##
##
## Residual standard error: 1.387 on 64 degrees of freedom
## Multiple R-Squared: 0.0225,  Adjusted R-squared: -0.03859
## F-statistic: 0.3684 on 4 and 64 DF,  p-value: 0.8303
##
##
## Estimation results for equation psei:
## =====================================
## psei = rgdp.l1 + psei.l1 + bsp.l1 + unem.l1
##
##         Estimate Std. Error t value Pr(>|t|)
## rgdp.l1  19.7364    30.5161   0.647    0.520
## psei.l1   0.2014     0.1235   1.631    0.108
## bsp.l1    7.8640    58.0975   0.135    0.893
## unem.l1 -21.4857    27.7011  -0.776    0.441
##
##
## Residual standard error: 347.1 on 64 degrees of freedom
## Multiple R-Squared: 0.05748, Adjusted R-squared: -0.001426
## F-statistic: 0.9758 on 4 and 64 DF,  p-value: 0.4271
##
##
## Estimation results for equation bsp:
## ====================================
## bsp = rgdp.l1 + psei.l1 + bsp.l1 + unem.l1
##
##           Estimate Std. Error t value Pr(>|t|)
## rgdp.l1  0.0742417  0.0593787   1.250    0.216
## psei.l1 -0.0002900  0.0002403  -1.206    0.232
## bsp.l1   0.0699962  0.1130473   0.619    0.538
## unem.l1  0.0468925  0.0539014   0.870    0.388
##
```

```
##
## Residual standard error: 0.6754 on 64 degrees of freedom
## Multiple R-Squared: 0.05491, Adjusted R-squared: -0.00416
## F-statistic: 0.9296 on 4 and 64 DF,  p-value: 0.4525
##
##
## Estimation results for equation unem:
## =====================================
## unem = rgdp.l1 + psei.l1 + bsp.l1 + unem.l1
##
##           Estimate Std. Error t value Pr(>|t|)
## rgdp.l1  0.0125843  0.1281911   0.098  0.92211
## psei.l1 -0.0003079  0.0005188  -0.593  0.55501
## bsp.l1  -0.1457935  0.2440546  -0.597  0.55236
## unem.l1 -0.3516801  0.1163662  -3.022  0.00361 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.458 on 64 degrees of freedom
## Multiple R-Squared: 0.1305,  Adjusted R-squared: 0.07617
## F-statistic: 2.402 on 4 and 64 DF,  p-value: 0.05894
##
##
##
## Covariance matrix of residuals:
##          rgdp       psei      bsp     unem
## rgdp  1.92366     41.959  0.05858  0.07537
## psei 41.95901 116576.717 -1.87439  8.03790
## bsp   0.05858     -1.874  0.45180 -0.10665
## unem  0.07537      8.038 -0.10665  2.11601
##
## Correlation matrix of residuals:
##         rgdp      psei       bsp     unem
## rgdp 1.00000  0.088604  0.062837  0.03736
## psei 0.08860  1.000000 -0.008167  0.01618
## bsp  0.06284 -0.008167  1.000000 -0.10907
## unem 0.03736  0.016184 -0.109073  1.00000
```

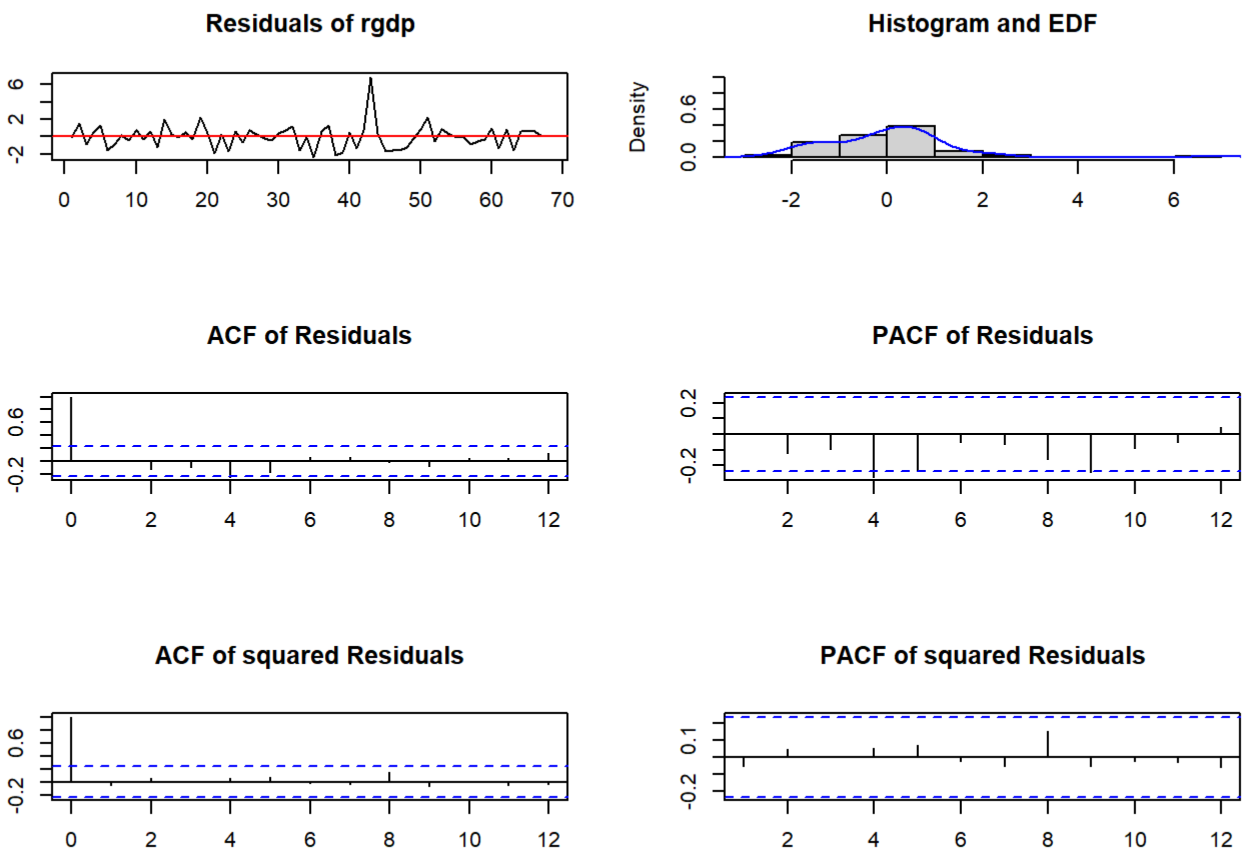
Se observa que las ecuaciones consideran 1 retraso de cada serie.

Diagnosis del modelo (Portmanteau test para objetos var)

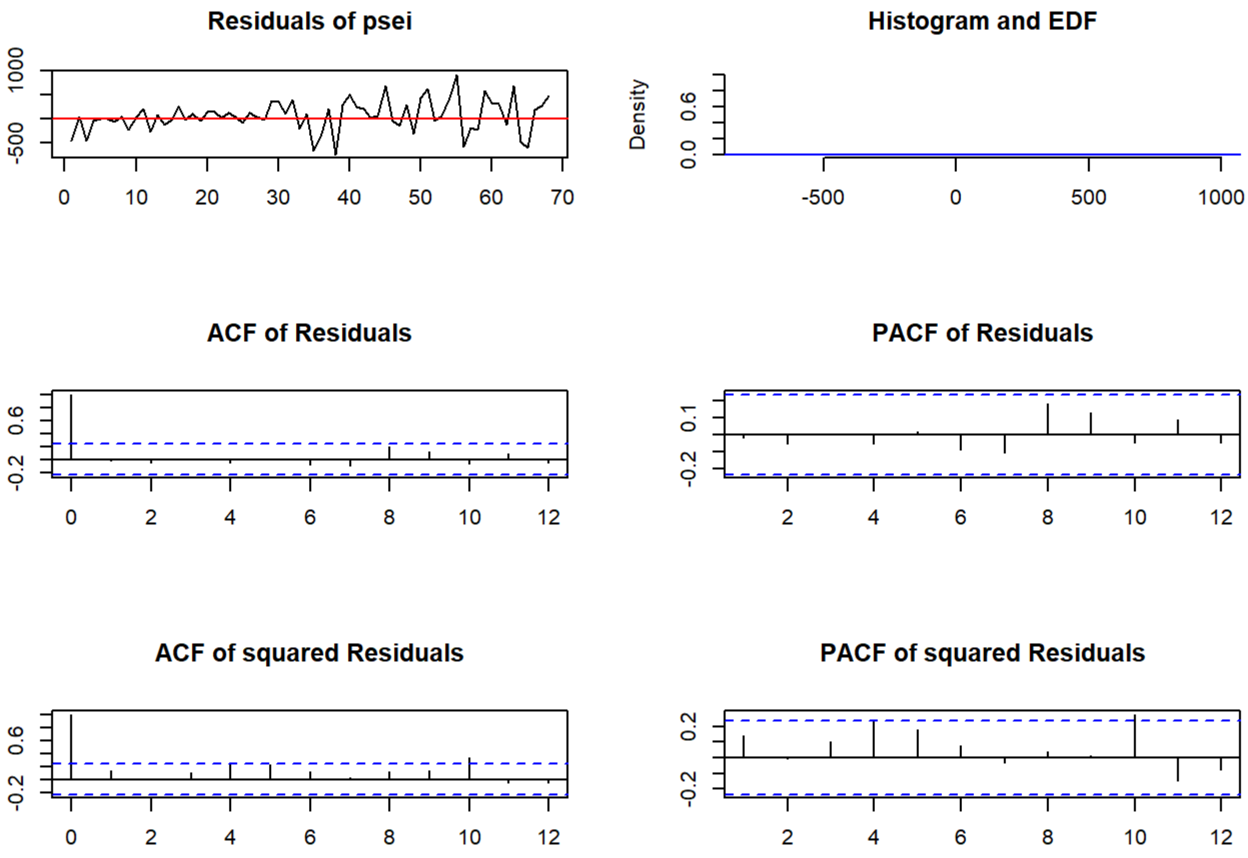
```
mv.serial=serial.test(var.a)
mv.serial
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var.a
## Chi-squared = 218.99, df = 240, p-value = 0.8309
```

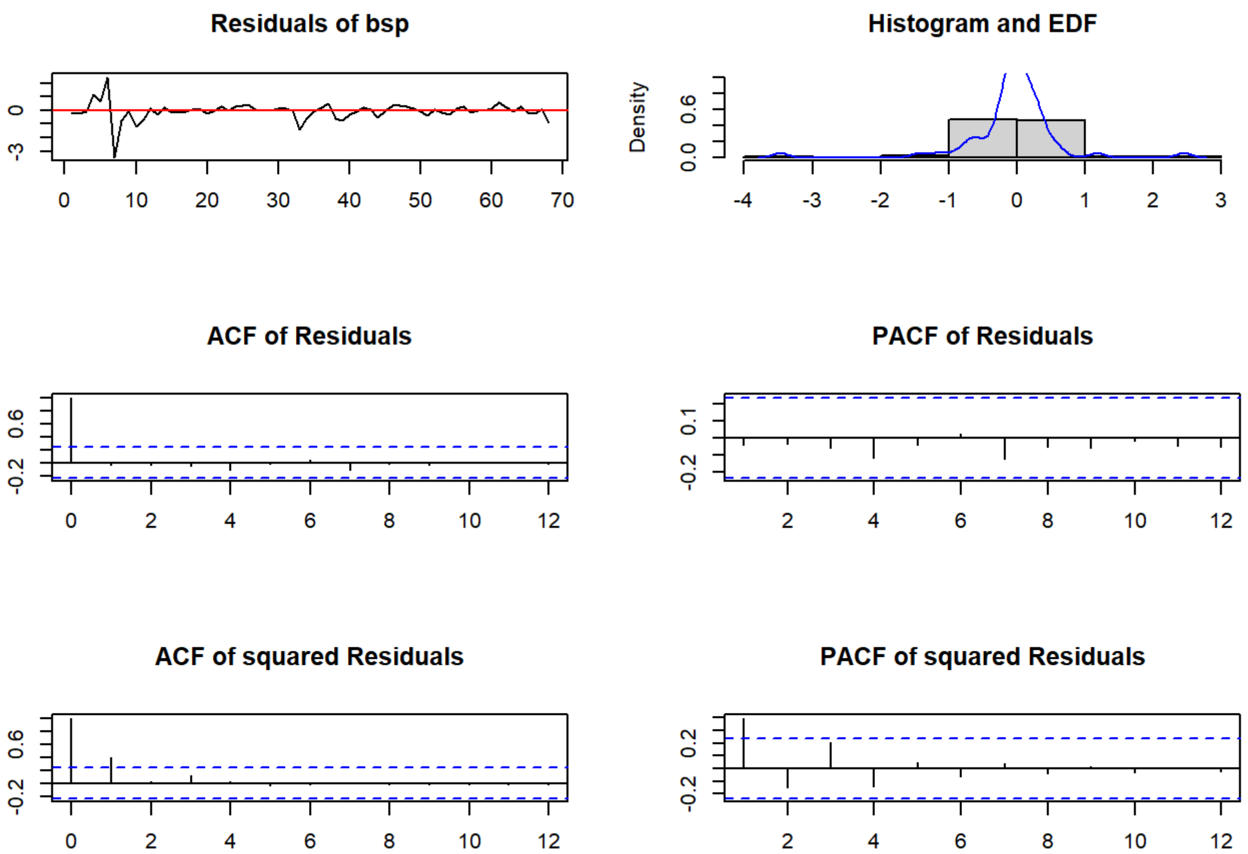
```
plot(mv.serial, names = "rgdp")
```



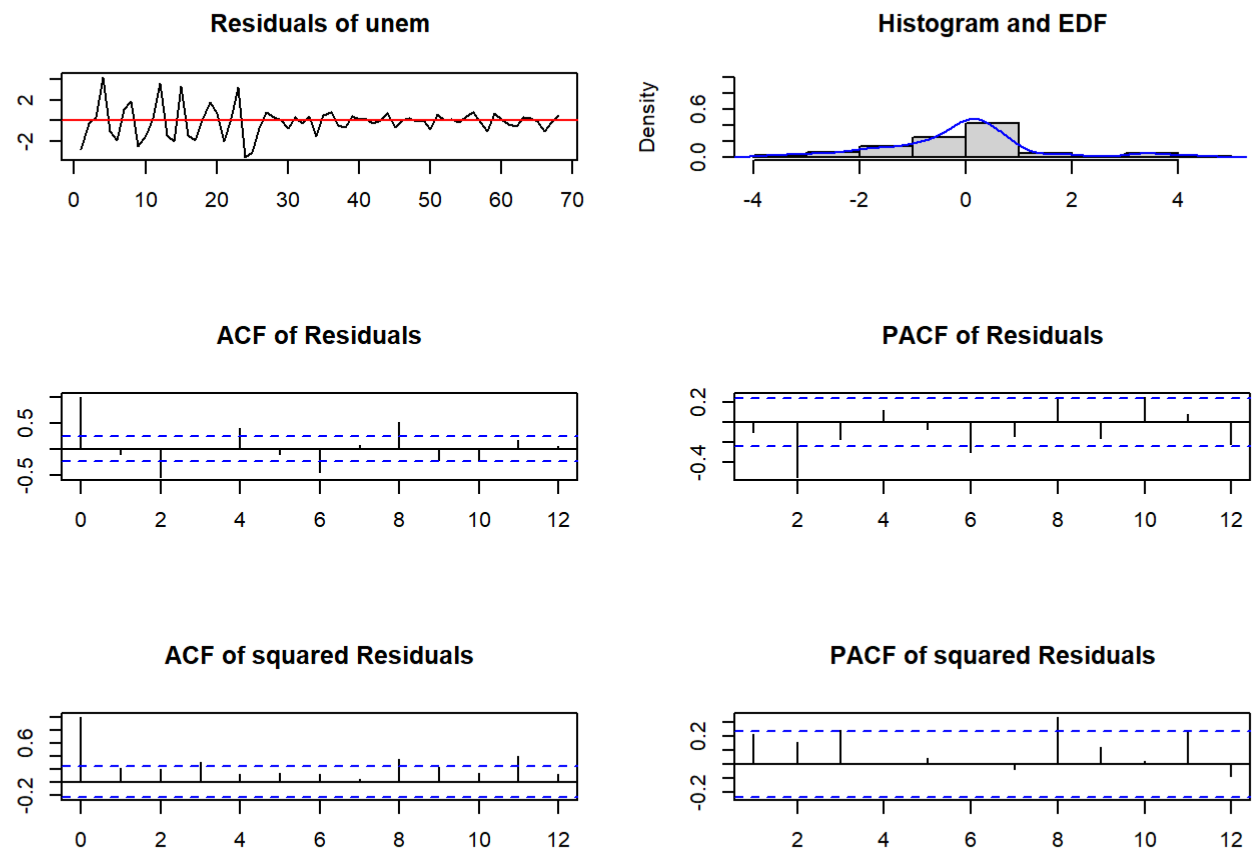

```
plot(mv.serial, names = "psei")
```




```
plot(mv.serial, names = "bsp")
```



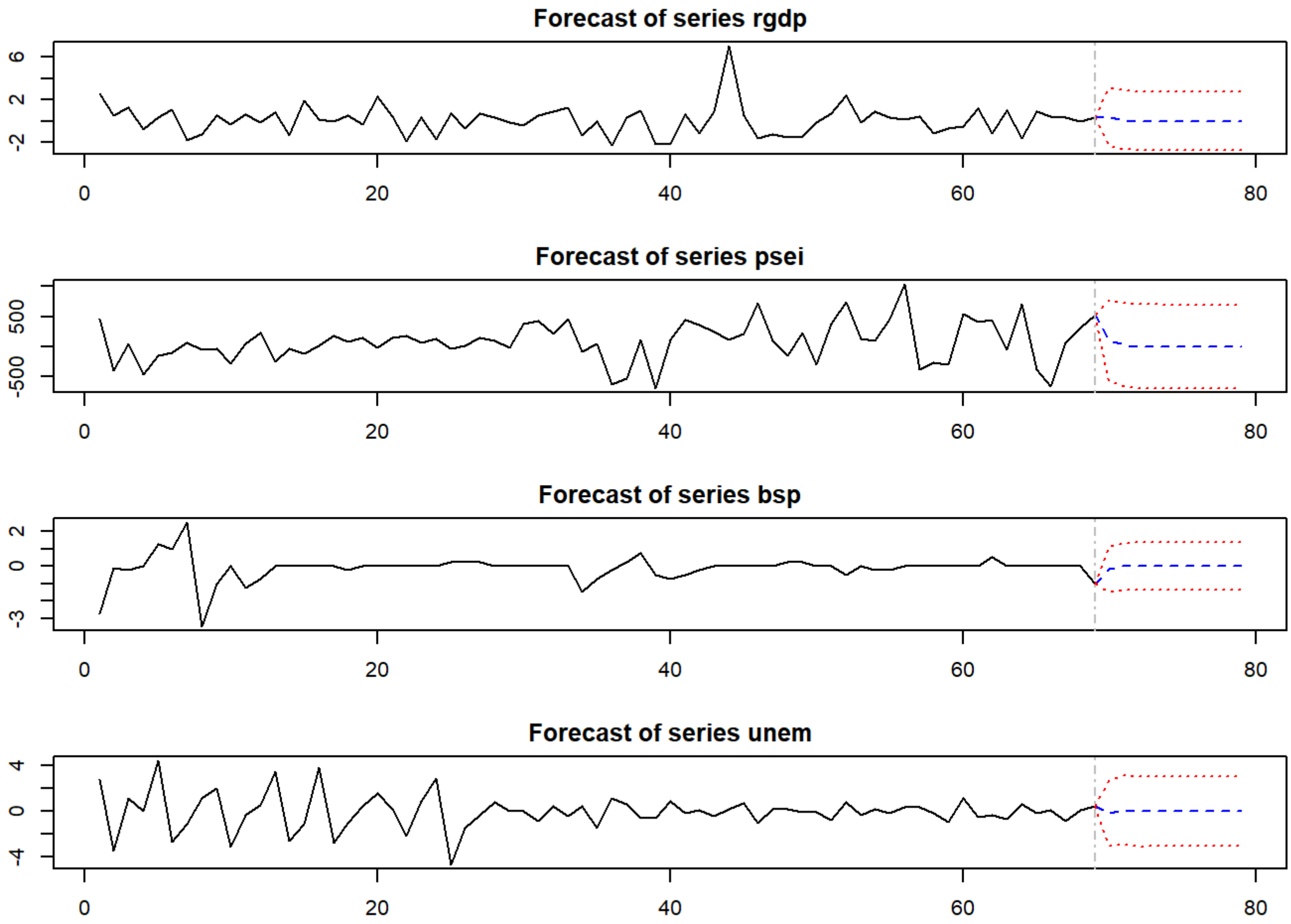
```
plot(mv.serial, names = "unem")
```



El p-value > 0.05, por lo tanto, cumple la diagnosis del modelo

Forecasting usando el modelo VAR (Hallando los pronósticos)

```
# Redefinir los márgenes de la figura
par(mar = c(3, 3, 2, 1))
fcast = predict(var.a, n.ahead = 10)
plot(fcast)
```




```
## Forecast gold
rgdp_pred = fcast$fcst[1]; # recuperar la columna correspondiente a las preds.
rgdp_pred
```

```
## $rgdp
##                fcst     lower    upper       CI
##  [1,]  3.910082e-01 -2.327399 3.109415 2.718407
##  [2,]  7.224536e-02 -2.672874 2.817364 2.745119
##  [3,]  1.784180e-02 -2.729385 2.765069 2.747227
##  [4,]  3.143057e-03 -2.744221 2.750507 2.747364
##  [5,]  1.317020e-03 -2.746056 2.748690 2.747373
##  [6,]  1.577920e-04 -2.747216 2.747532 2.747374
##  [7,]  1.002931e-04 -2.747274 2.747475 2.747374
##  [8,]  3.070562e-06 -2.747371 2.747377 2.747374
##  [9,]  8.599543e-06 -2.747366 2.747383 2.747374
## [10,] -6.458244e-07 -2.747375 2.747374 2.747374
```

```
# Extrayendo la columna de pronósticos (CASO RGDP)
x = rgdp_pred$rgdp[,1];
x
```

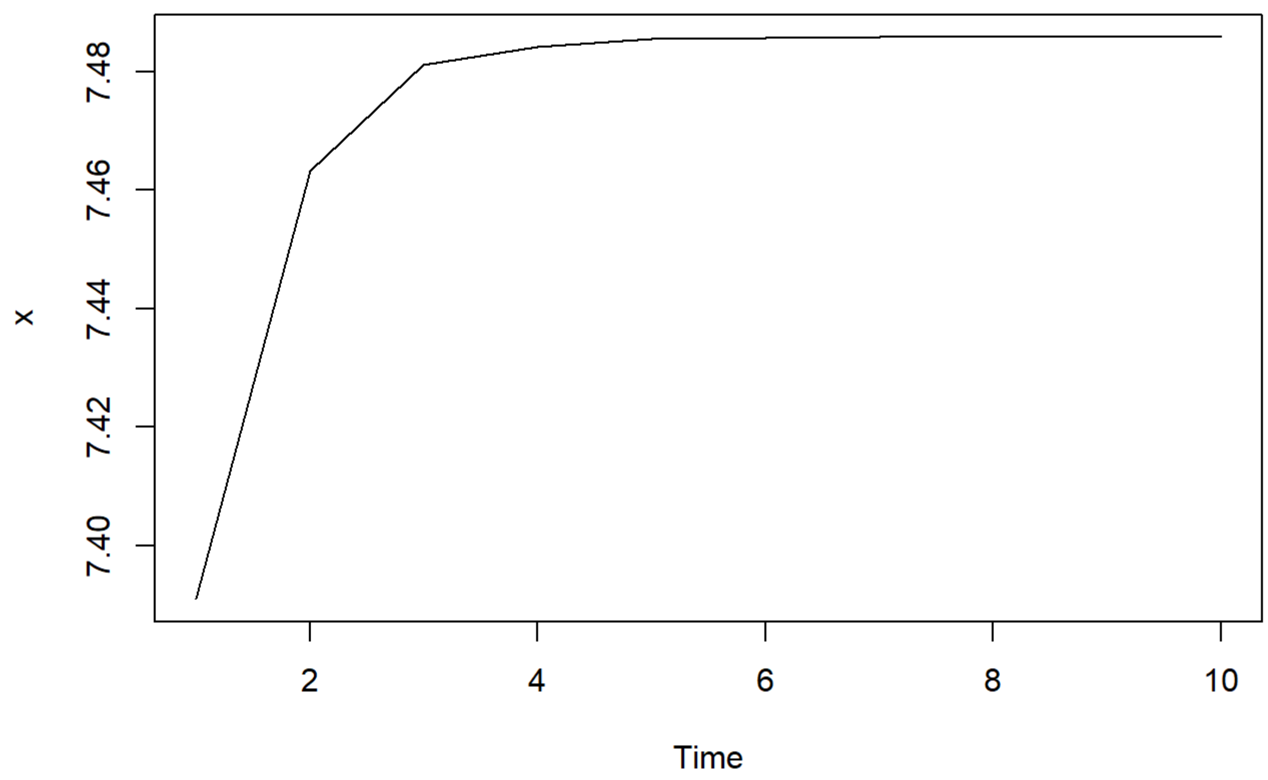
```
##  [1]  3.910082e-01  7.224536e-02  1.784180e-02  3.143057e-03  1.317020e-03
##  [6]  1.577920e-04  1.002931e-04  3.070562e-06  8.599543e-06 -6.458244e-07
```

```
# Mostrar los ultimos valores RGDP
tail(X_train)
```

```
##       rgdp    psei bsp unem
## [65,]  5.1 7940.49   4  6.6
## [66,]  6.0 7564.50   4  6.4
## [67,]  6.4 6893.98   4  6.5
## [68,]  6.7 6952.08   4  5.6
## [69,]  6.7 7262.30   4  5.7
## [70,]  7.0 7796.25   3  6.1
```

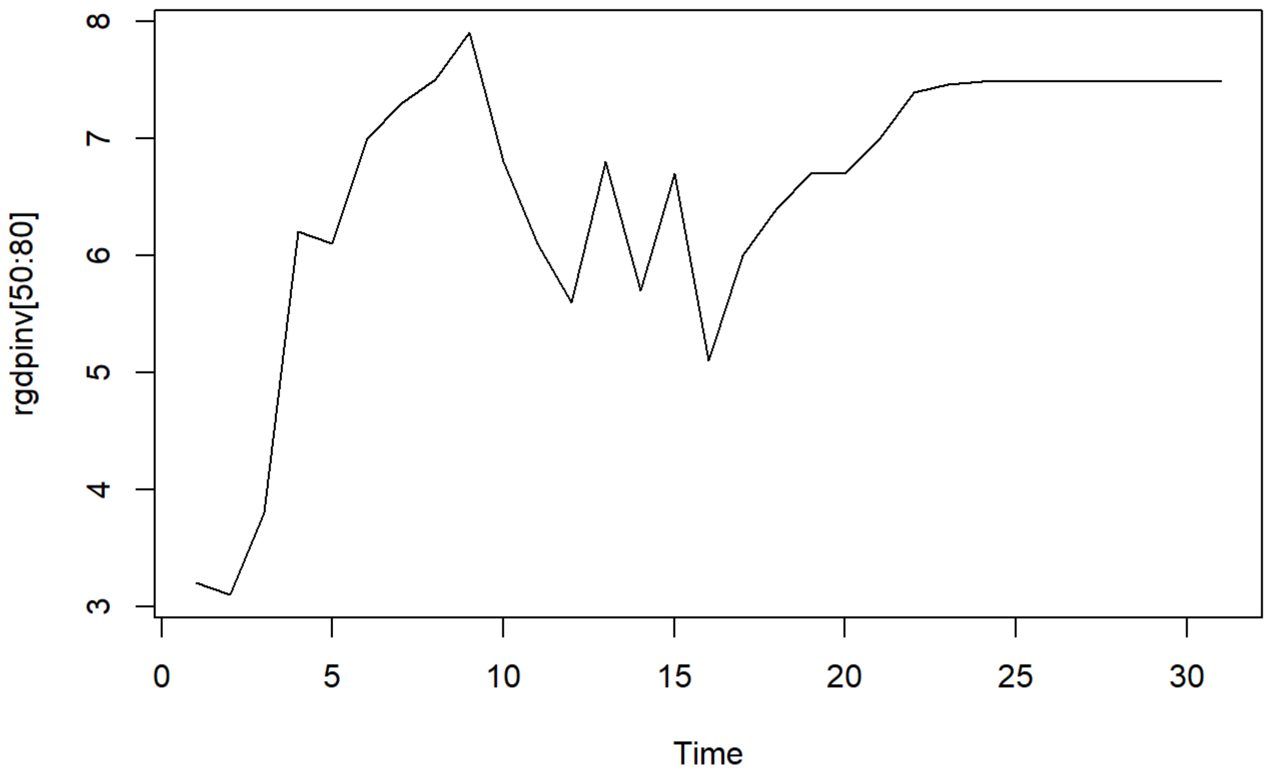
se observa que el ultimo valor de rgdp es 7.

```
# Invirtiendo la diferenciación sobre rgdp
x = cumsum(x) + 7

plot.ts(x)
```



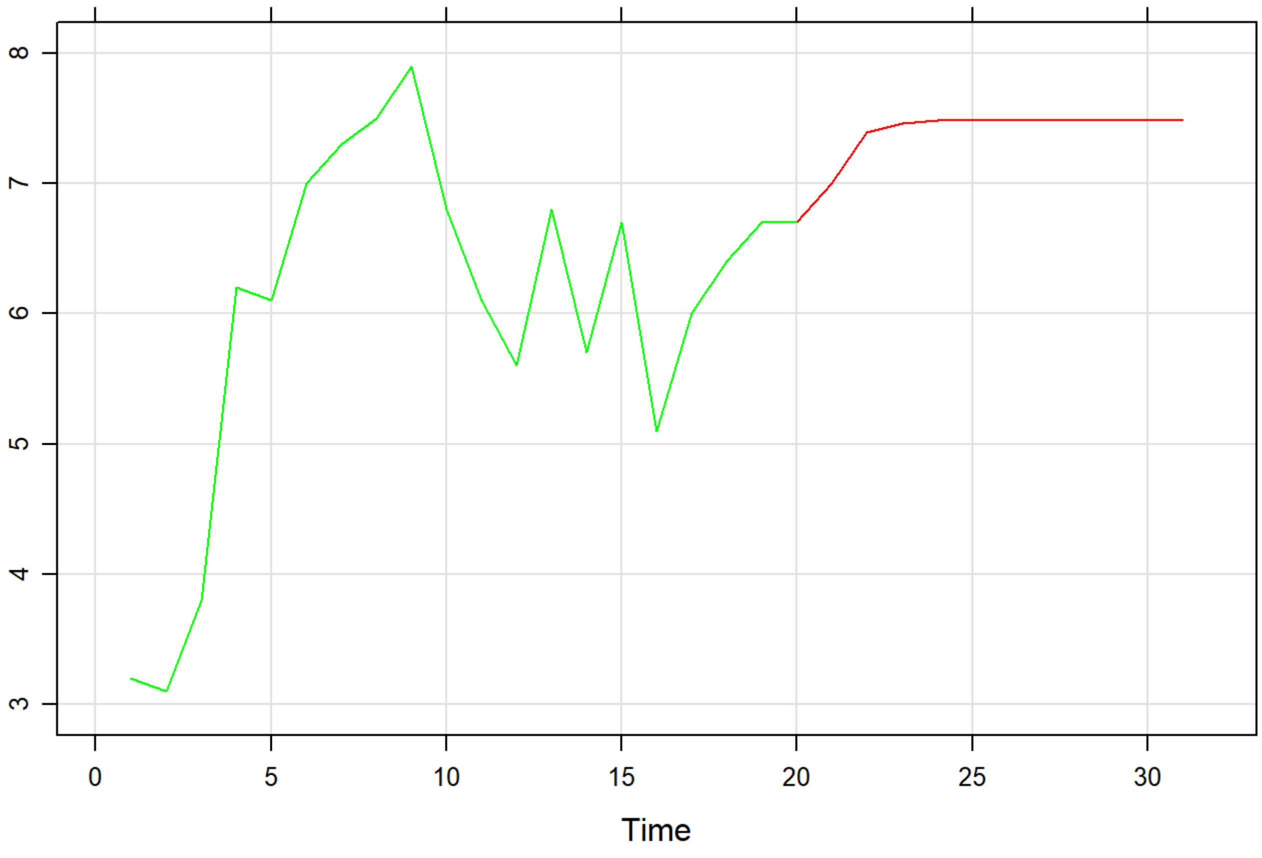

```
# Mostrar los datos reales en la escala original
# Combinando los datos reales y la predicción en una sola serie de tiempo
rgdpinv =ts(c(X_train[,1], x),
            start = c(1999,1), frequency = 4)

# Dibujando todo
plot(rgdpinv)
```


```
plot.ts(rgdpinv[50:80])
```



```r
# Plot avanzado con separación visual entre lo real y lo pronosticado
library(lattice)
library(grid)
library(zoo)

# Objeto zoo
xx = zoo(rgdpinv[50:80])
```

```r
xyplot(xx, grid=TRUE, panel = function(xx, y, ...){

  panel.xyplot(xx, y, col="red", ...)

  grid.clip(unit(20, "native"), just=c("right"))

  panel.xyplot(xx, y, col="green", ...) })
```




```r
### Evaluacion del modelo "rgdp"
rmse=sqrt(mean((X_test[,1]-x)^2))
rmse
```

```
## [1] 0.9860489
```

El valor de RMSE (Root Mean Square Error o Raíz del Error Cuadrático Medio) de 0.9860489 para los datos macroeconómicos del RGDP (Producto Interno Bruto Real) de Filipinas es relativamente bajo, lo que sugiere que el modelo de predicción tiene un buen ajuste a los datos históricos. Un RMSE bajo indica que las diferencias entre los valores predichos por el modelo y los valores reales observados son, en promedio, pequeñas.

En el contexto de los datos macroeconómicos, un RMSE cercano a 1 podría significar que el modelo es capaz de predecir el RGDP con un margen de error cercano a una unidad de la medida utilizada. Esto es particularmente útil para los formuladores de políticas y analistas económicos, ya que proporciona una herramienta confiable para la planificación y el análisis económico.

```
## Forecast psei
psei_pred = fcast$fcst[2]; # recuperar la columna correspondiente a las preds.
psei_pred
```

```
## $psei
##                fcst     lower    upper       CI
##  [1,] 97.0044602116 -583.3039 777.3128 680.3084
##  [2,] 29.1501752300 -670.4675 728.7679 699.6177
##  [3,]  5.9308657335 -694.9333 706.7951 700.8642
##  [4,]  2.0727769824 -698.8765 703.0220 700.9492
##  [5,]  0.3049998929 -700.6512 701.2612 700.9562
##  [6,]  0.1505737825 -700.8062 701.1074 700.9568
##  [7,]  0.0109951562 -700.9459 700.9678 700.9569
##  [8,]  0.0122518379 -700.9446 700.9691 700.9569
##  [9,] -0.0003330458 -700.9572 700.9565 700.9569
## [10,]  0.0011270942 -700.9557 700.9580 700.9569
```

```
# Extrayendo la columna de pronósticos (CASO PSEI)
x = psei_pred$psei[,1];
x
```

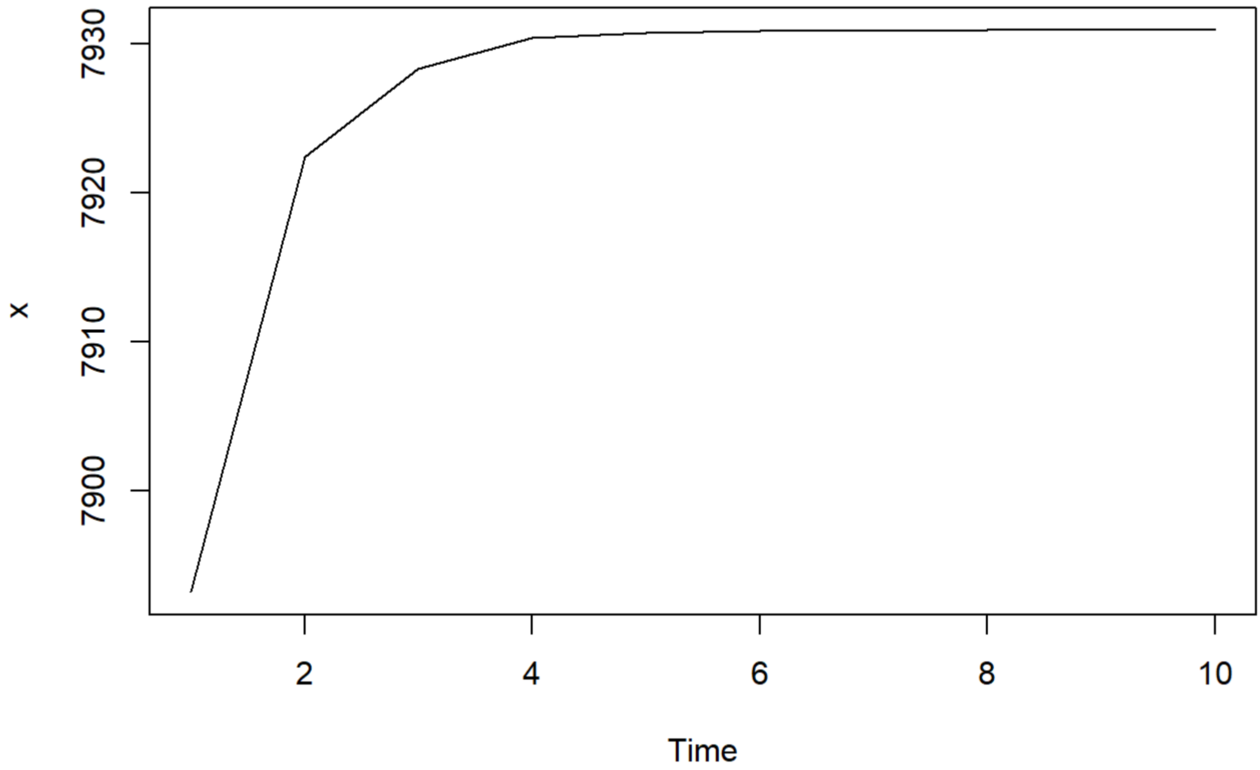
```
##  [1] 97.0044602116 29.1501752300  5.9308657335  2.0727769824  0.3049998929
##  [6]  0.1505737825  0.0109951562  0.0122518379 -0.0003330458  0.0011270942
```

```
# Mostrar los ultimos valores PSEI
tail(X_train)
```

```
##       rgdp    psei bsp unem
## [65,]  5.1 7940.49   4  6.6
## [66,]  6.0 7564.50   4  6.4
## [67,]  6.4 6893.98   4  6.5
## [68,]  6.7 6952.08   4  5.6
## [69,]  6.7 7262.30   4  5.7
## [70,]  7.0 7796.25   3  6.1
```

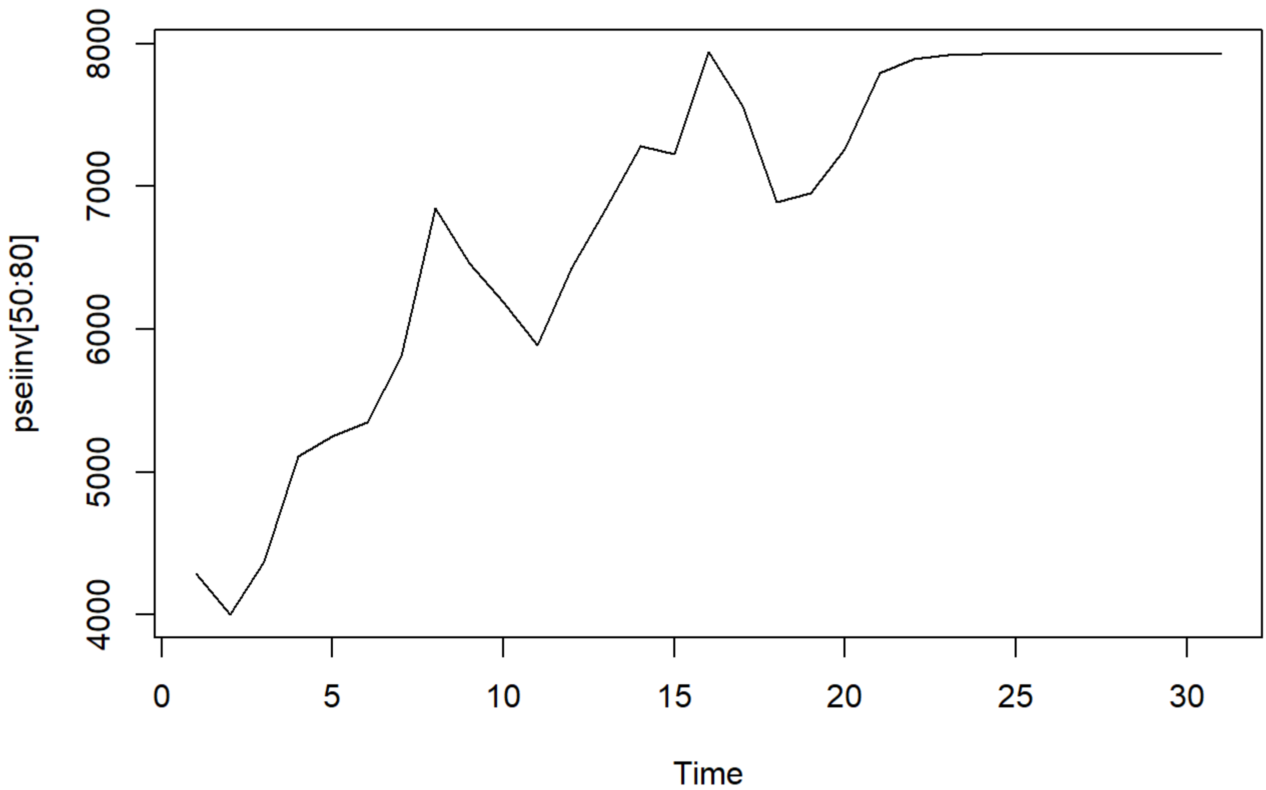
se observa que el ultimo valor de rgdp es 7796.25

```
# Invirtiendo la diferenciación sobre psei
x = cumsum(x) + 7796.25

plot.ts(x)
```



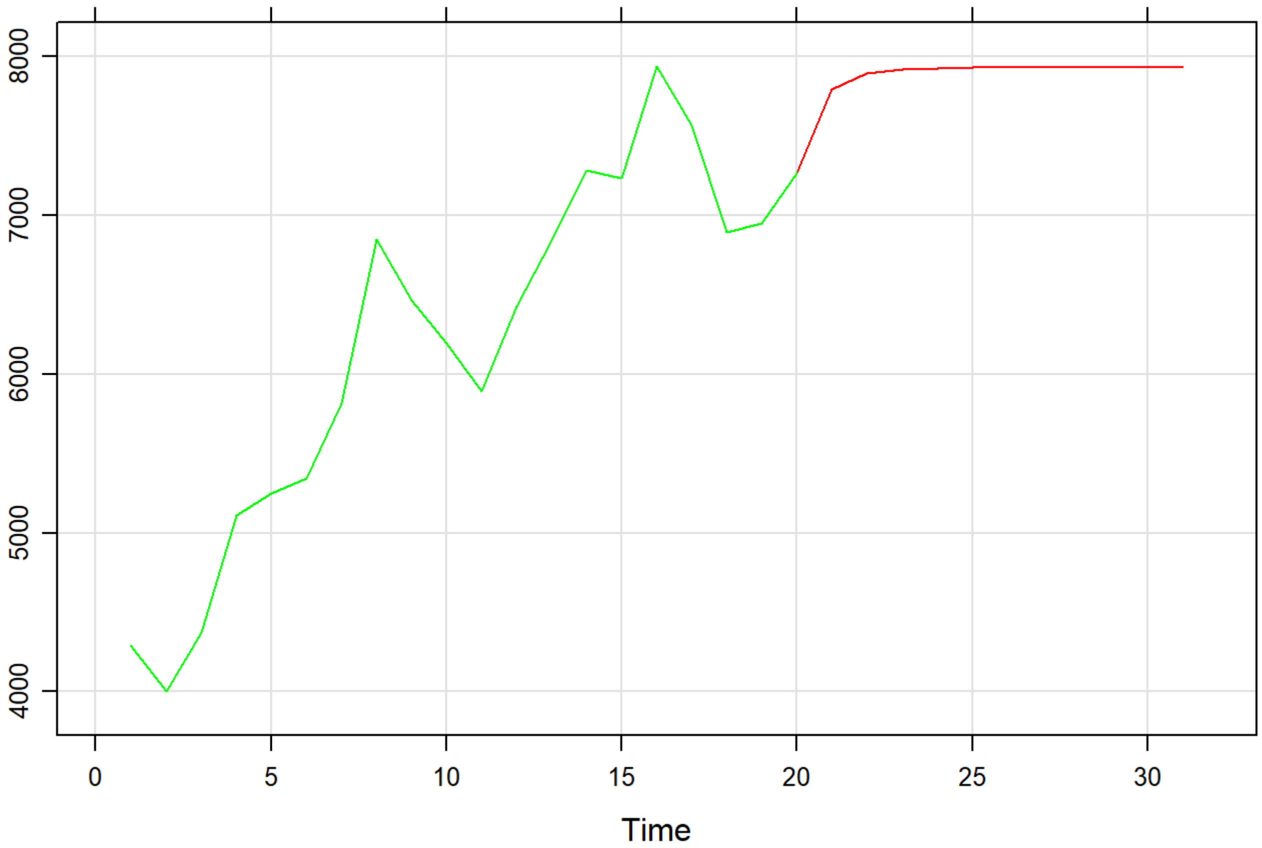

```
# Mostrar los datos reales en la escala original
# Combinando los datos reales y la predicción en una sola serie de tiempo
pseiinv =ts(c(X_train[,2], x),
            start = c(1999,1), frequency = 4)

# Dibujando todo
plot(pseiinv)
```


```
plot.ts(pseiinv[50:80])
```



```r
# Plot avanzado con separación visual entre lo real y lo pronosticado
library(lattice)
library(grid)
library(zoo)

# Objeto zoo
xx = zoo(pseiinv[50:80])
```

```r
xyplot(xx, grid=TRUE, panel = function(xx, y, ...){

  panel.xyplot(xx, y, col="red", ...)

  grid.clip(unit(20, "native"), just=c("right"))

  panel.xyplot(xx, y, col="green", ...) })
```




```r
### Evaluacion del modelo "rgdp"
rmse=sqrt(mean((X_test[,2]-x)^2))
rmse
```

```
## [1] 571.7547
```

El resultado sugiere que el modelo tiene una precisión aceptable para las predicciones.