Datos Macroeconónicos de EEUU

By: Ael-Dev

Descripción de los Datos

Los datos macroeconómicos de Estados Unidos proporcionan información crucial sobre variables como el Producto Interno Bruto (GDP), tasas de interés, desempleo, entre otros indicadores económicos clave.

Importar paquetes en R

```
# lectura del dataset
library(readr)

# test ADF de estacionariedad
library(tseries)
```

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

```
# plot
library(ggplot2)
library(ggfortify)
```

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

```
## 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
```

Cargando los datos

```
data <- read_csv('https://raw.githubusercontent.com/ecabestadistica/series-temporales-multiva
riantes/master/2.%20Casos%20de%20estudio%20en%20R/A.%20Datos%20macroecon%C3%B3micos%20de%20US
A/blanchQua.csv',
                 col_types = cols(
                                  dates = col_double(),
                                  GDP = col_double(),
                                  U = col_double()
                                  )
                 )
head(data)
```

dates <dbl>	GDP <dbl>	U <dbl>
1948.02	0.85443952	-0.14374997
1948.03	0.01210090	-0.06746175
1948.04	-0.08505847	-0.02450694
1949.01	-2.07658590	0.78511467
1949.02	-1.49186810	1.96140290
1949.03	-0.14913989	2.77102440

6 rows

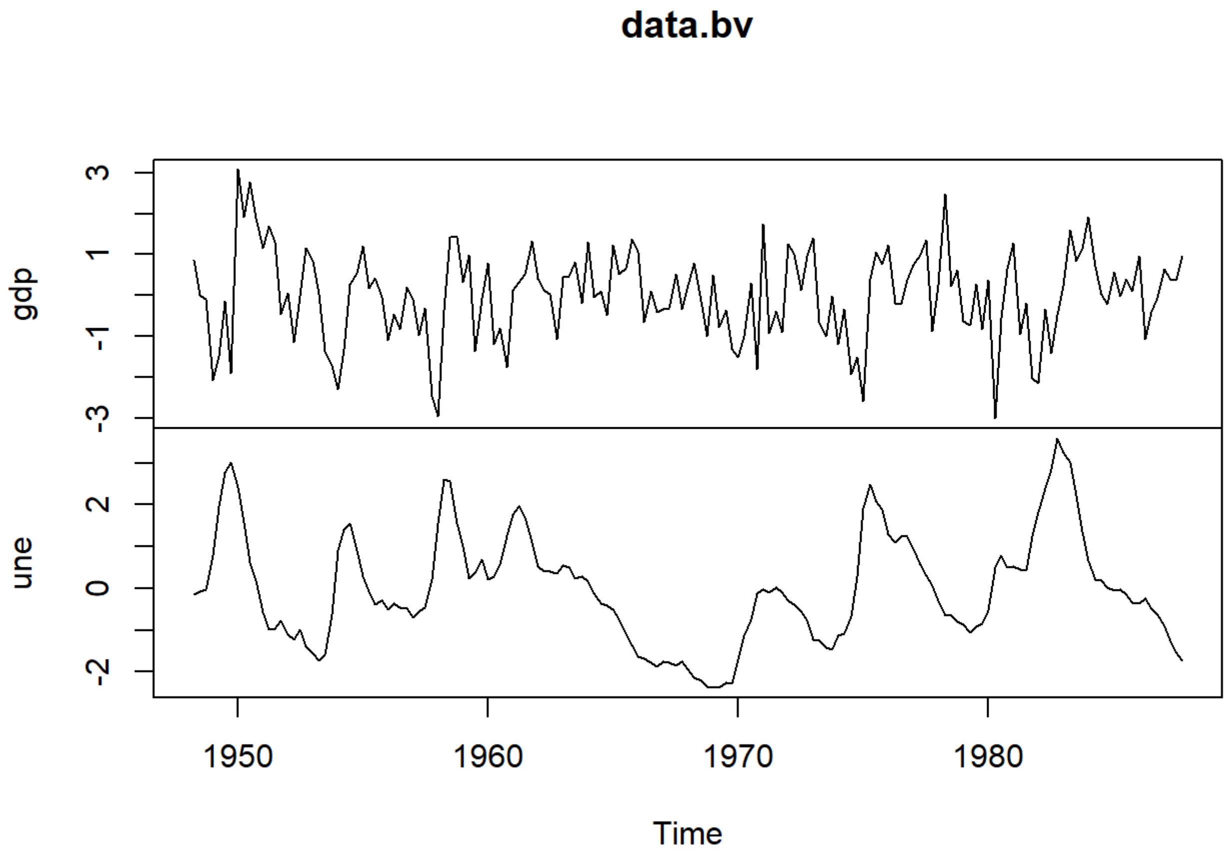
```
class(data)
```

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

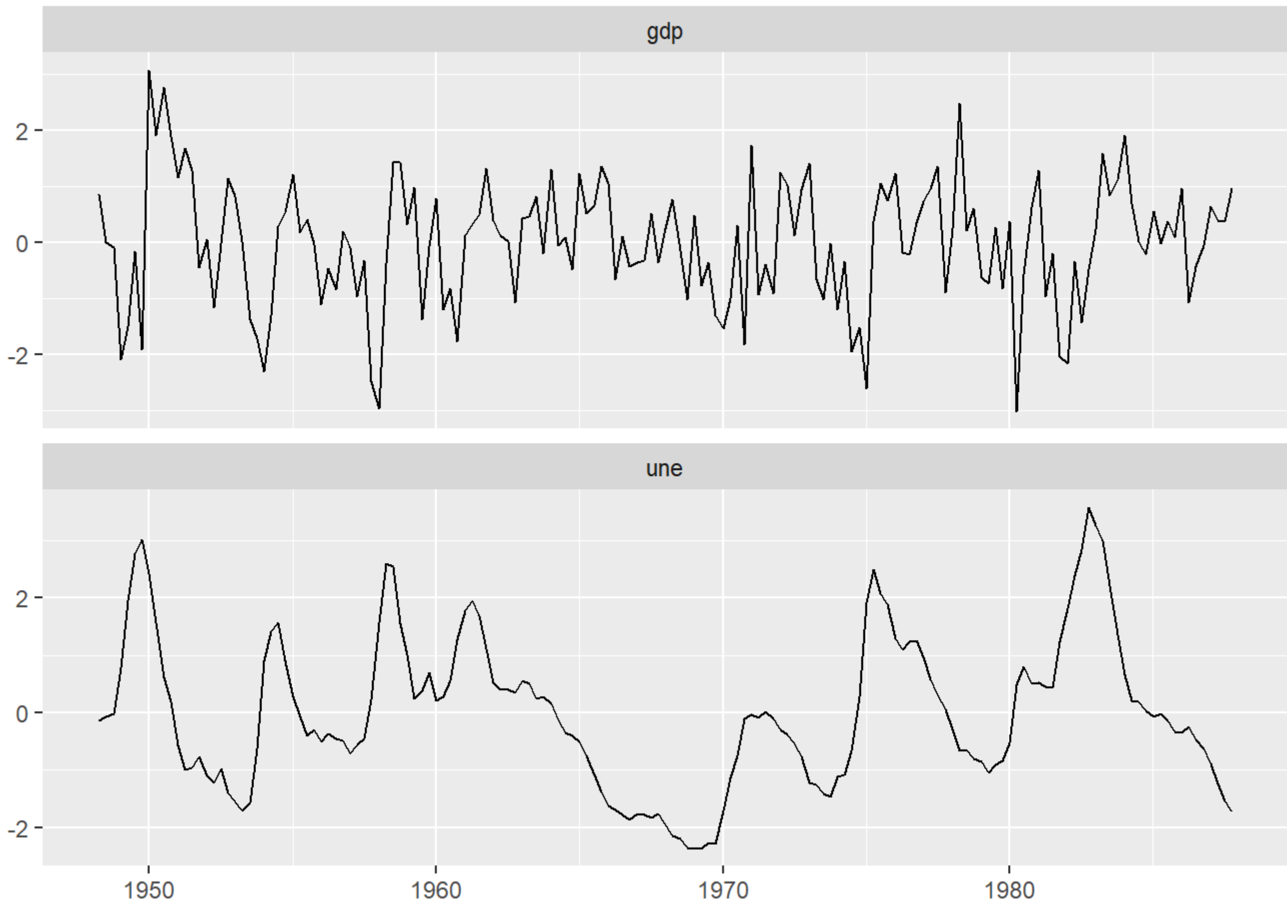
Análisis exploratorio de los datos

```
# Convertir a objeto ts
gdp <- ts(data$GDP, start = c(1948, 2), freq = 4) # 4 trimestres
une <- ts(data$U, start = c(1948, 2), freq = 4)
```

```
# Gráfico con plot:
data.bv <- cbind(gdp, une)
plot(data.bv)
```




```
library(ggplot2)
#install.packages("ggfortify")
library(ggfortify)
autoplot(data.bv )
```




```
## Dividir la serie en conjunto de entrenamiento y de prueba
library(dplyr)

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

```
## [1] 10  2
```

```
####### Prueba de estacionariedad
apply(X_train, 2, adf.test) #2 para especificar que lo queremos aplicar por columnas
```

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

```
## $gdp
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -4.7992, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
##
## $une
##
##  Augmented Dickey-Fuller Test
##
## data:  newX[, i]
## Dickey-Fuller = -3.4885, Lag order = 5, p-value = 0.0459
## alternative hypothesis: stationary
```

Todos los p-values son < 0.05, => no hay que diferenciar las series

Creando el modelo VAR

```r
# 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: zoo
```

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

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

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

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

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

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

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      1      2
##
## $criteria
##                  1           2           3           4           5           6
## AIC(n) -2.82819854 -2.90791028 -2.89783899 -2.86623464 -2.82935387 -2.82623785
## HQ(n)  -2.79388222 -2.83927764 -2.79489003 -2.72896935 -2.65777227 -2.62033992
## SC(n)  -2.74375325 -2.73901970 -2.64450312 -2.52845346 -2.40712741 -2.31956609
## FPE(n)  0.05911949  0.05459142  0.05514817  0.05692732  0.05908041  0.05928636
##                  7           8          9          10
## AIC(n) -2.78060501 -2.73652219 -2.7052103 -2.72286531
## HQ(n)  -2.54039077 -2.46199163 -2.3963634 -2.37970211
## SC(n)  -2.18948796 -2.06095985 -1.9452027 -1.87841238
## FPE(n)  0.06208603  0.06492826  0.0670521  0.06595099
```

Seleccionaremos el criterio AIC, el cual nos indica un VAR de orden 2

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

                   lag.max = 10,

                   ic = "AIC",

                   type = "const")

summary(var.a)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gdp, une
## Deterministic variables: const
## Sample size: 147
## Log Likelihood: -202.613
## Roots of the characteristic polynomial:
## 0.8335 0.8335 0.1612 0.04178
## Call:
## vars::VAR(y = X_train, type = "const", lag.max = 10, ic = "AIC")
##
##
## Estimation results for equation gdp:
## ====================================
## gdp = gdp.l1 + une.l1 + gdp.l2 + une.l2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## gdp.l1  0.08324    0.11169   0.745   0.4573
## une.l1 -0.54573    0.32008  -1.705   0.0904 .
## gdp.l2  0.05625    0.09471   0.594   0.5535
## une.l2  0.84113    0.32662   2.575   0.0110 *
## const  -0.03378    0.08092  -0.418   0.6769
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.9794 on 142 degrees of freedom
## Multiple R-Squared: 0.2753,  Adjusted R-squared: 0.2549
## F-statistic: 13.49 on 4 and 142 DF,  p-value: 2.417e-09
##
##
## Estimation results for equation une:
## ====================================
## une = gdp.l1 + une.l1 + gdp.l2 + une.l2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## gdp.l1 -0.12717    0.03730  -3.409 0.000849 ***
## une.l1  1.30855    0.10691  12.240  < 2e-16 ***
## gdp.l2 -0.03194    0.03163  -1.010 0.314311
## une.l2 -0.39444    0.10909  -3.616 0.000415 ***
## const   0.00129    0.02703   0.048 0.961997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3271 on 142 degrees of freedom
## Multiple R-Squared: 0.9427,  Adjusted R-squared: 0.9411
## F-statistic: 583.8 on 4 and 142 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##         gdp     une
## gdp  0.9592 -0.2116
```

```
## une -0.2116  0.1070
##
## Correlation matrix of residuals:
##         gdp     une
## gdp  1.0000 -0.6606
## une -0.6606  1.0000
```

Causalidad de Granger

```
# verificar si estas 2 series estan correlacionadas
causality(var.a, cause = c("gdp"))
```

```
## $Granger
##
##  Granger causality H0: gdp do not Granger-cause une
##
## data:  VAR object var.a
## F-Test = 5.8894, df1 = 2, df2 = 284, p-value = 0.003118
##
##
## $Instant
##
##  H0: No instantaneous causality between: gdp and une
##
## data:  VAR object var.a
## Chi-squared = 44.656, df = 1, p-value = 2.349e-11
```

```
causality(var.a, cause = c("une"))
```

```
## $Granger
##
##  Granger causality H0: une do not Granger-cause gdp
##
## data:  VAR object var.a
## F-Test = 12.411, df1 = 2, df2 = 284, p-value = 6.803e-06
##
##
## $Instant
##
##  H0: No instantaneous causality between: une and gdp
##
## data:  VAR object var.a
## Chi-squared = 44.656, df = 1, p-value = 2.349e-11
```

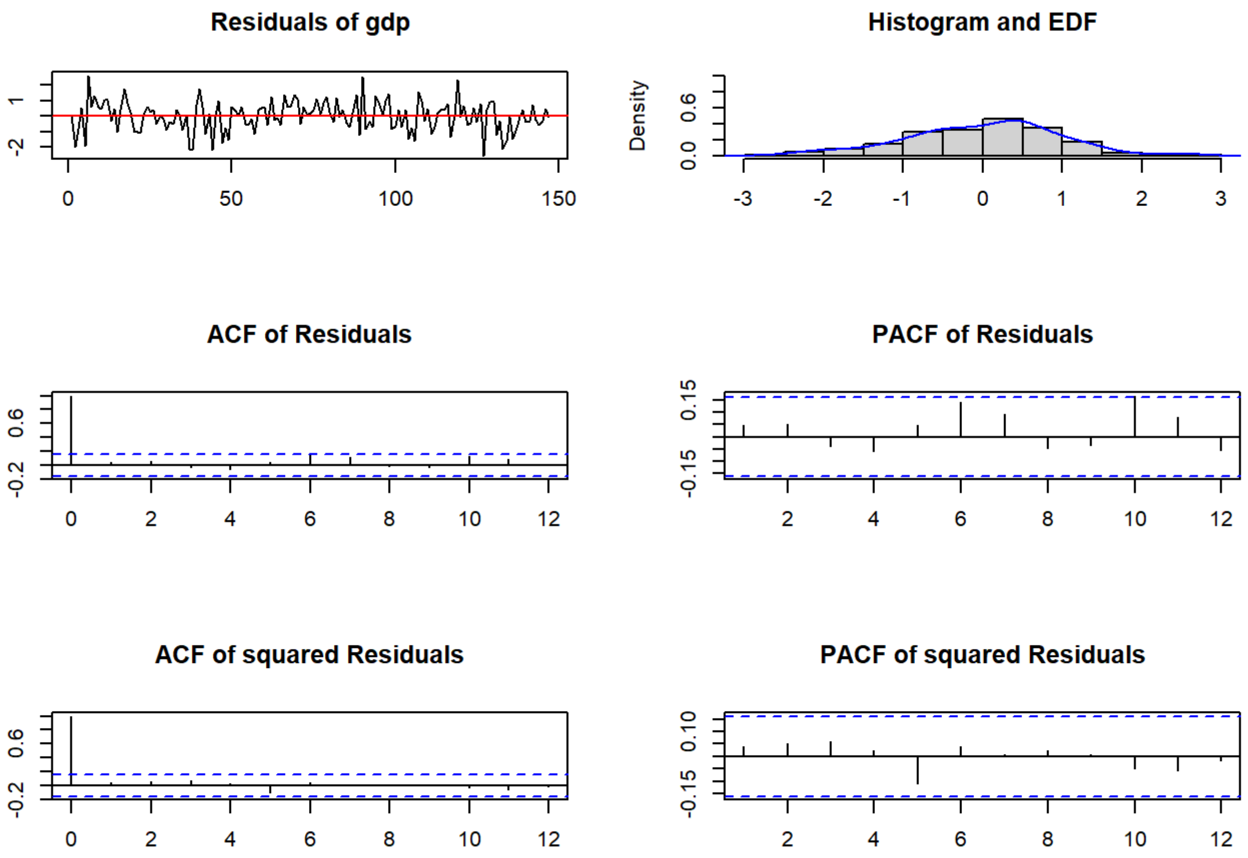
dado que los p-values <0.05, => hay presencia de una causalidad bidireccional

```
## Diagnosis del modelo (Portmanteau test para objetos var) o analisis de sus residuos, donde
la funcion permite realizar un test de hipotesis de los residuos.

# Deseamos obtener un p-valor > 0.05.

# Posibles soluciones si es < 0.05:
# a) Cambiar el orden del modelo.
# b) Cambiar el tipo de modelo.
# c) Añadir otro paso de diferenciación o transformar con logaritmos.

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

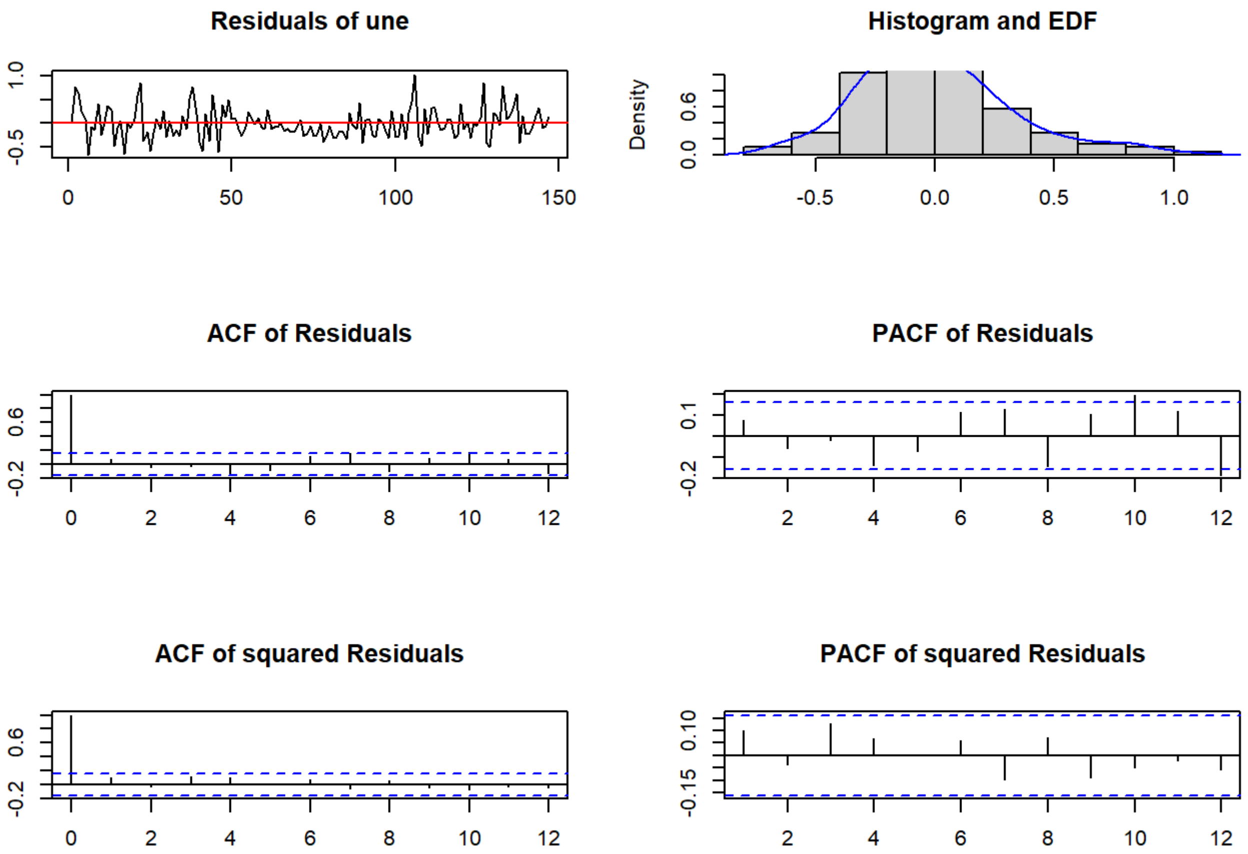
```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var.a
## Chi-squared = 57.677, df = 56, p-value = 0.413
```

en este caso el p-value > 0.5 por lo que cumple lo que se buscaba.

```
# ploteando la diagnosis
plot(bv.serial, names = "gdp")
```



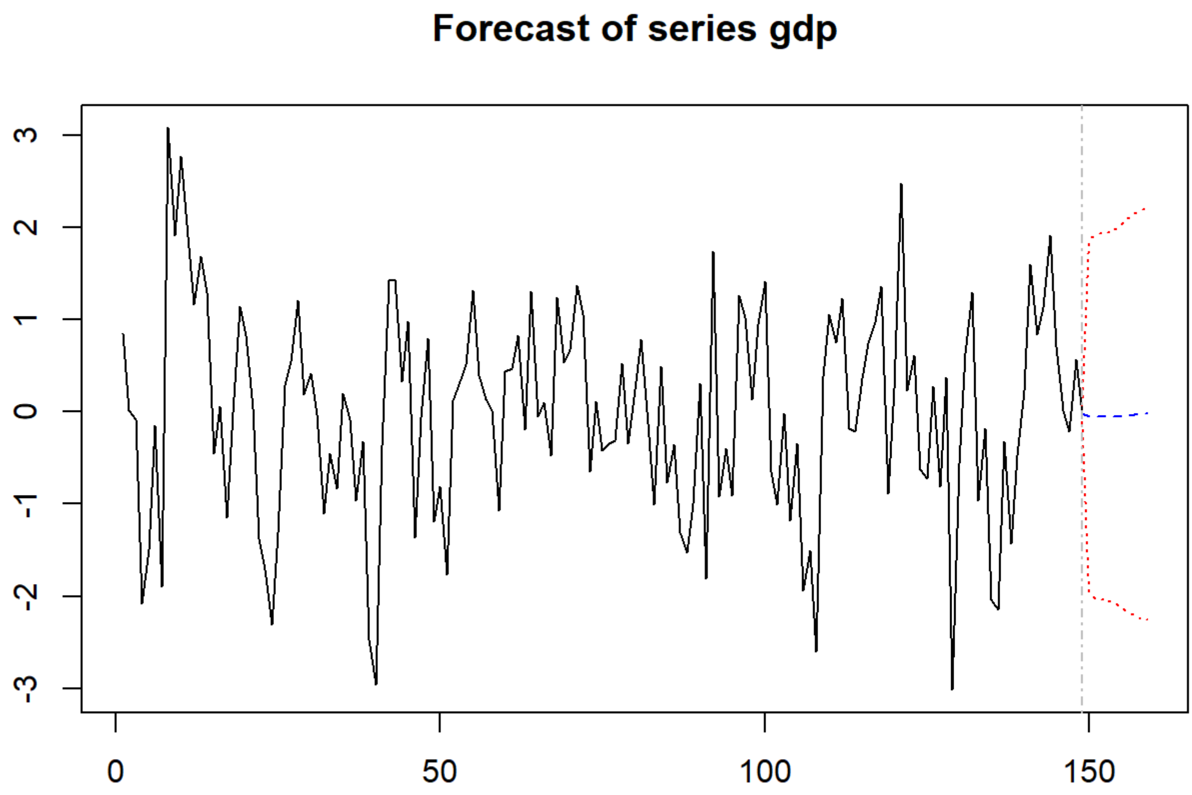

```
plot(bv.serial, names = "une")
```



Forecasting usando el modelo VAR (Hallando los pronósticos)

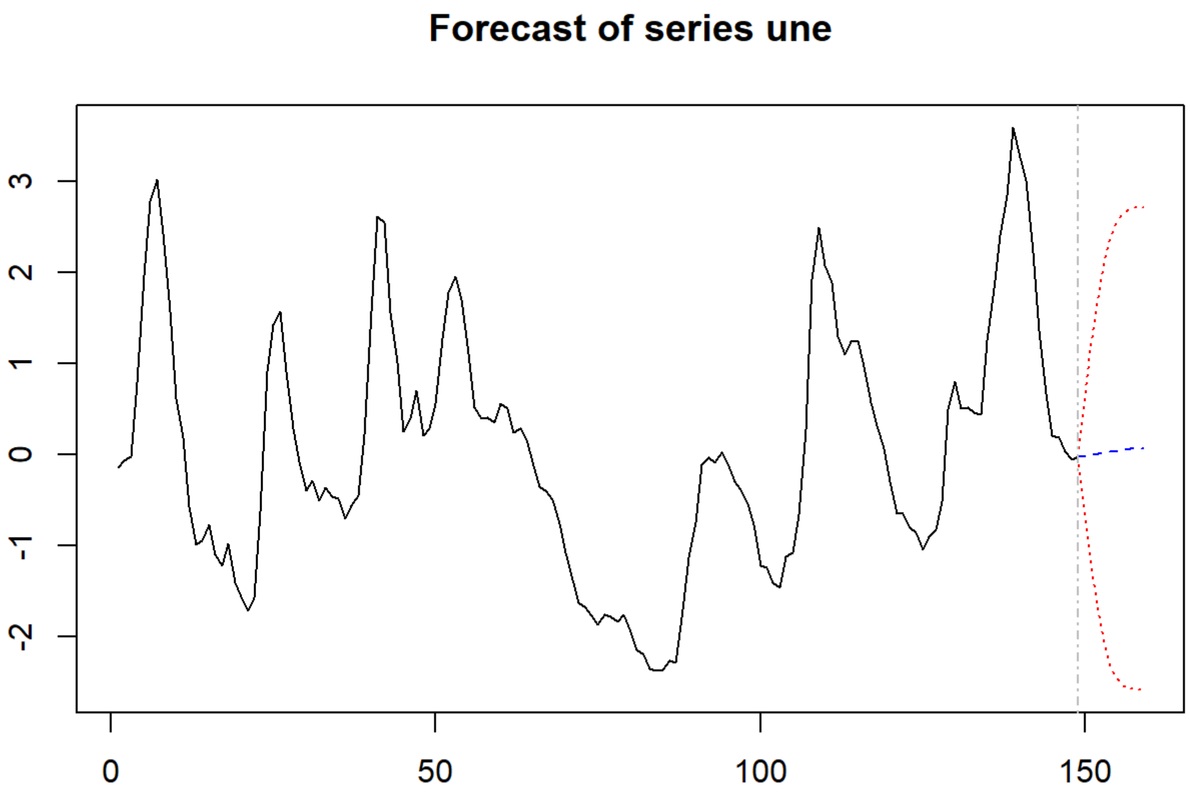
Respecto a GDP

```
predictions <- predict(var.a, n.ahead = 10, ci = 0.95)
plot(predictions, names = "gdp")
```

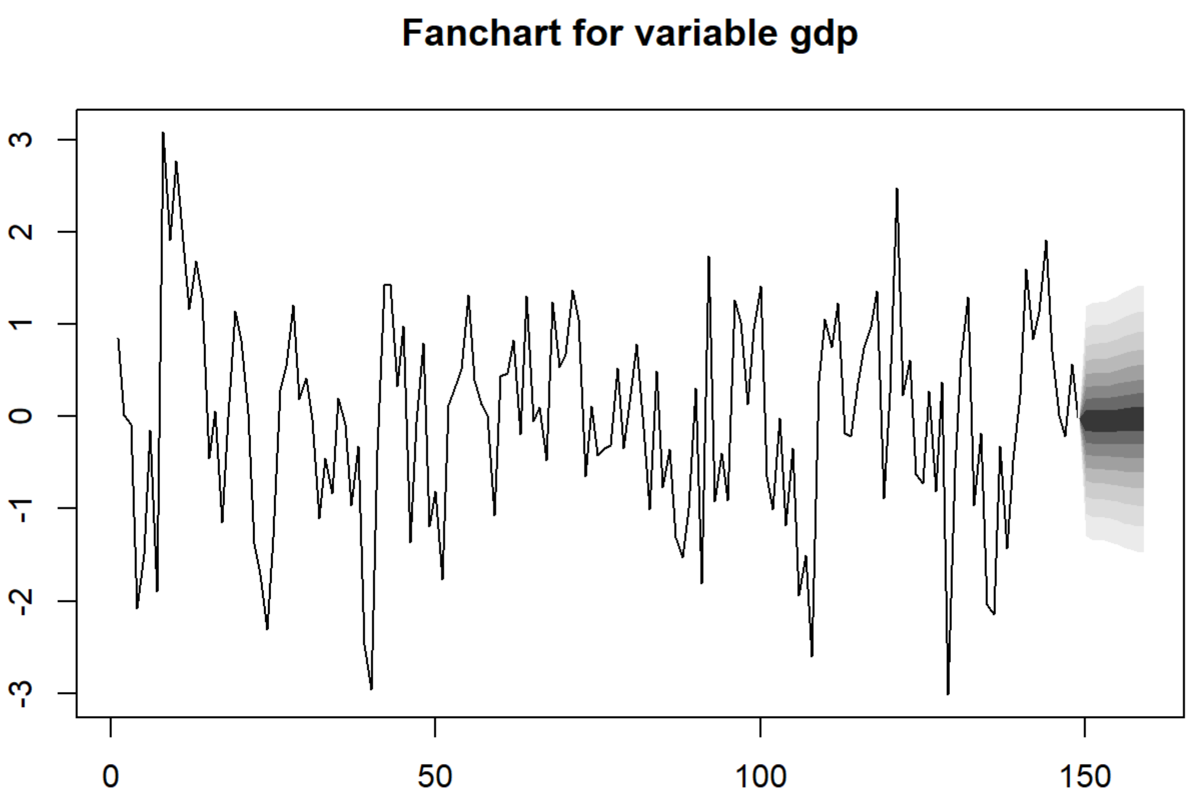


Respecto a une

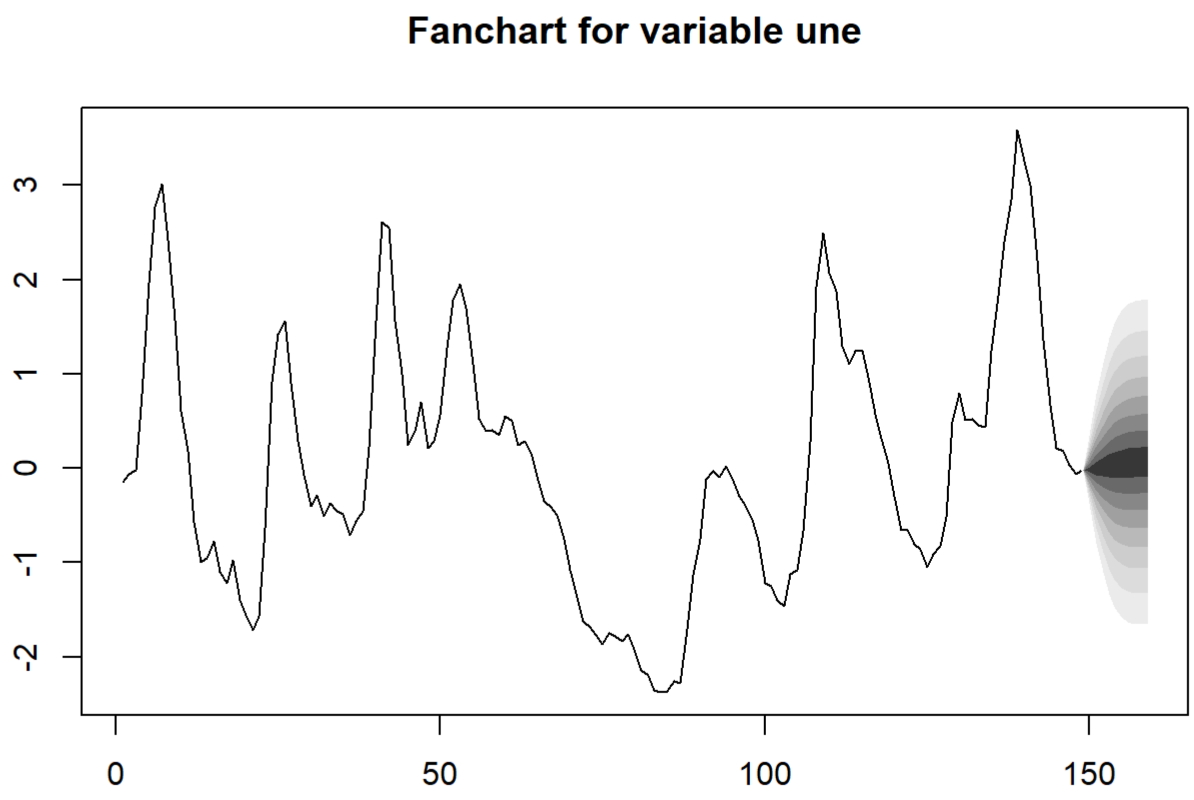
```
predictions <- predict(var.a, n.ahead = 10, ci = 0.95)
plot(predictions, names = "une")
```




```
# OTRA FORMA DE GRAFICAR
fanchart(predictions, names = "gdp")
```



```
fanchart(predictions, names = "une")
```



Evaluación del modelo

```
pred=predictions$fcst
rmse=sqrt(mean((X_test[,1]-pred$gdp)^2))
cat('RMSE gdp: ', rmse)
```

```
## RMSE gdp:  1.845125
```

```
rmse=sqrt(mean((X_test[,2]-pred$une)^2))
cat('RMSE une: ', rmse)
```

```
## RMSE une:  2.340479
```

se observa que se obtiene resultados aceptables teniendo en cuenta que se observa 2 datos en el pasado

Reentrenando el modelo con todos los datos

```
# Identificación del orden del modelo
VARselect(data.bv, type = "none", lag.max = 10)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
##
## $criteria
##                  1           2           3           4           5           6
## AIC(n) -2.88055345 -2.97522838 -2.96572378 -2.93483896 -2.89661887 -2.89965712
## HQ(n)  -2.84778959 -2.90970066 -2.86743221 -2.80378352 -2.73279957 -2.70307397
## SC(n)  -2.79991059 -2.81394267 -2.72379522 -2.61226754 -2.49340460 -2.41580000
## FPE(n)  0.05610388  0.05103709  0.05152765  0.05315026  0.05523189  0.05508059
##                  7          8           9          10
## AIC(n) -2.85864313 -2.8191397 -2.78690253 -2.80496021
## HQ(n)  -2.62929613 -2.5570289 -2.49202780 -2.47732162
## SC(n)  -2.29414316 -2.1739969 -2.06111684 -1.99853167
## FPE(n)  0.05741032  0.0597565  0.06175805  0.06070675
```

```
# Creando el modelo
var.a <- vars::VAR(data.bv,
                   lag.max = 10,
                   ic = "AIC",
                   type = "const")

summary(var.a)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gdp, une
## Deterministic variables: const
## Sample size: 157
## Log Likelihood: -211.593
## Roots of the characteristic polynomial:
## 0.8361 0.8361 0.1014 0.1014
## Call:
## vars::VAR(y = data.bv, type = "const", lag.max = 10, ic = "AIC")
##
##
## Estimation results for equation gdp:
## ====================================
## gdp = gdp.l1 + une.l1 + gdp.l2 + une.l2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## gdp.l1  0.06905    0.10782   0.640  0.52289
## une.l1 -0.61530    0.30723  -2.003  0.04699 *
## gdp.l2  0.04741    0.09098   0.521  0.60309
## une.l2  0.90193    0.31426   2.870  0.00469 **
## const  -0.01570    0.07684  -0.204  0.83833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.9613 on 152 degrees of freedom
## Multiple R-Squared: 0.2687,  Adjusted R-squared: 0.2495
## F-statistic: 13.97 on 4 and 152 DF,  p-value: 1e-09
##
##
## Estimation results for equation une:
## ====================================
## une = gdp.l1 + une.l1 + gdp.l2 + une.l2 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## gdp.l1 -0.119156   0.035975  -3.312 0.001157 **
## une.l1  1.333482   0.102510  13.008  < 2e-16 ***
## gdp.l2 -0.029883   0.030358  -0.984 0.326497
## une.l2 -0.416951   0.104853  -3.977 0.000108 ***
## const  -0.007501   0.025639  -0.293 0.770243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3207 on 152 degrees of freedom
## Multiple R-Squared: 0.9429,  Adjusted R-squared: 0.9414
## F-statistic: 628.1 on 4 and 152 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##         gdp     une
## gdp  0.9240 -0.2022
```

```
## une -0.2022  0.1029
##
## Correlation matrix of residuals:
##         gdp     une
## gdp  1.0000 -0.6558
## une -0.6558  1.0000
```

```
# Causalidad de Granger
causality(var.a, cause = c("gdp"))
```

```
## $Granger
##
##  Granger causality H0: gdp do not Granger-cause une
##
## data:  VAR object var.a
## F-Test = 5.5723, df1 = 2, df2 = 304, p-value = 0.0042
##
##
## $Instant
##
##  H0: No instantaneous causality between: gdp and une
##
## data:  VAR object var.a
## Chi-squared = 47.214, df = 1, p-value = 6.364e-12
```

```
causality(var.a, cause = c("une"))
```

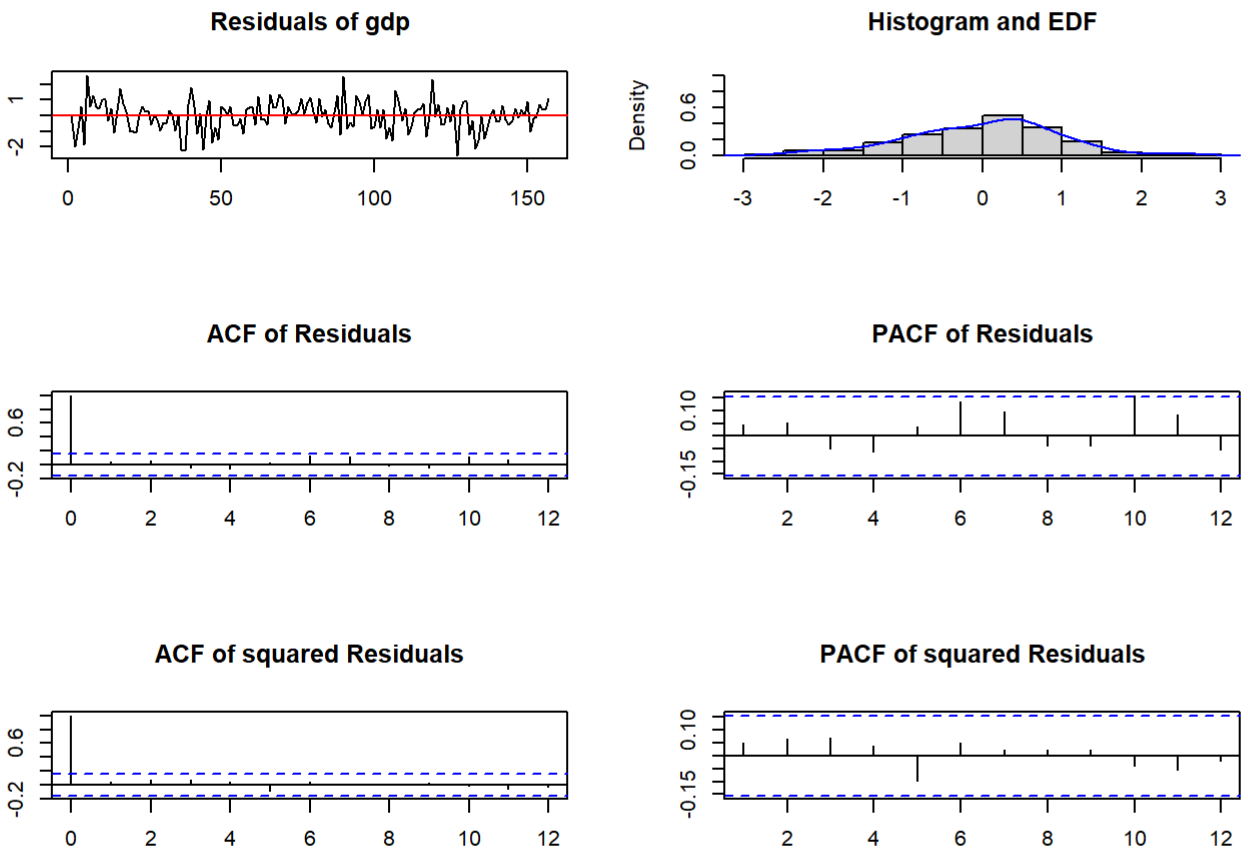
```
## $Granger
##
##  Granger causality H0: une do not Granger-cause gdp
##
## data:  VAR object var.a
## F-Test = 12.821, df1 = 2, df2 = 304, p-value = 4.511e-06
##
##
## $Instant
##
##  H0: No instantaneous causality between: une and gdp
##
## data:  VAR object var.a
## Chi-squared = 47.214, df = 1, p-value = 6.364e-12
```

```
## Diagnosis del modelo (Portmanteau test para objetos var)

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

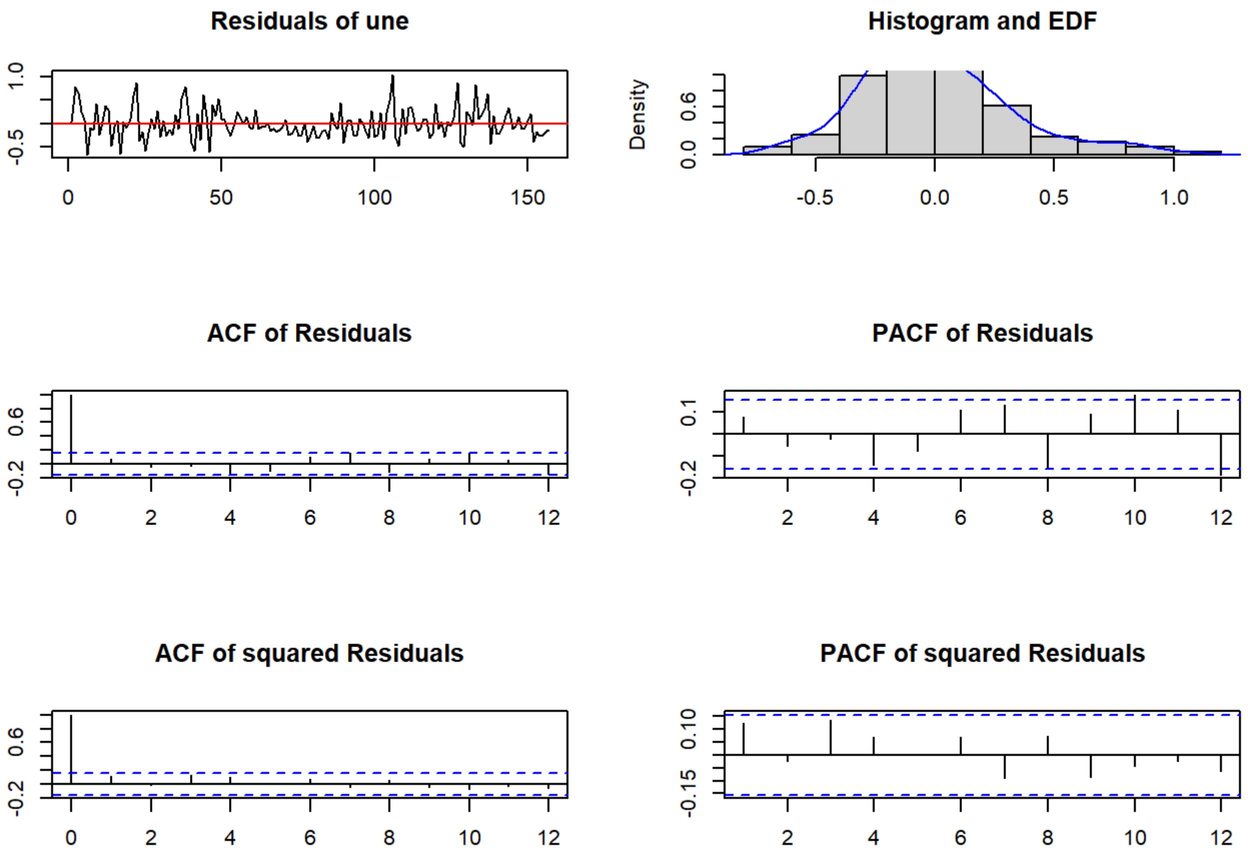
```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var.a
## Chi-squared = 59.717, df = 56, p-value = 0.3422
```

pasa la prueba

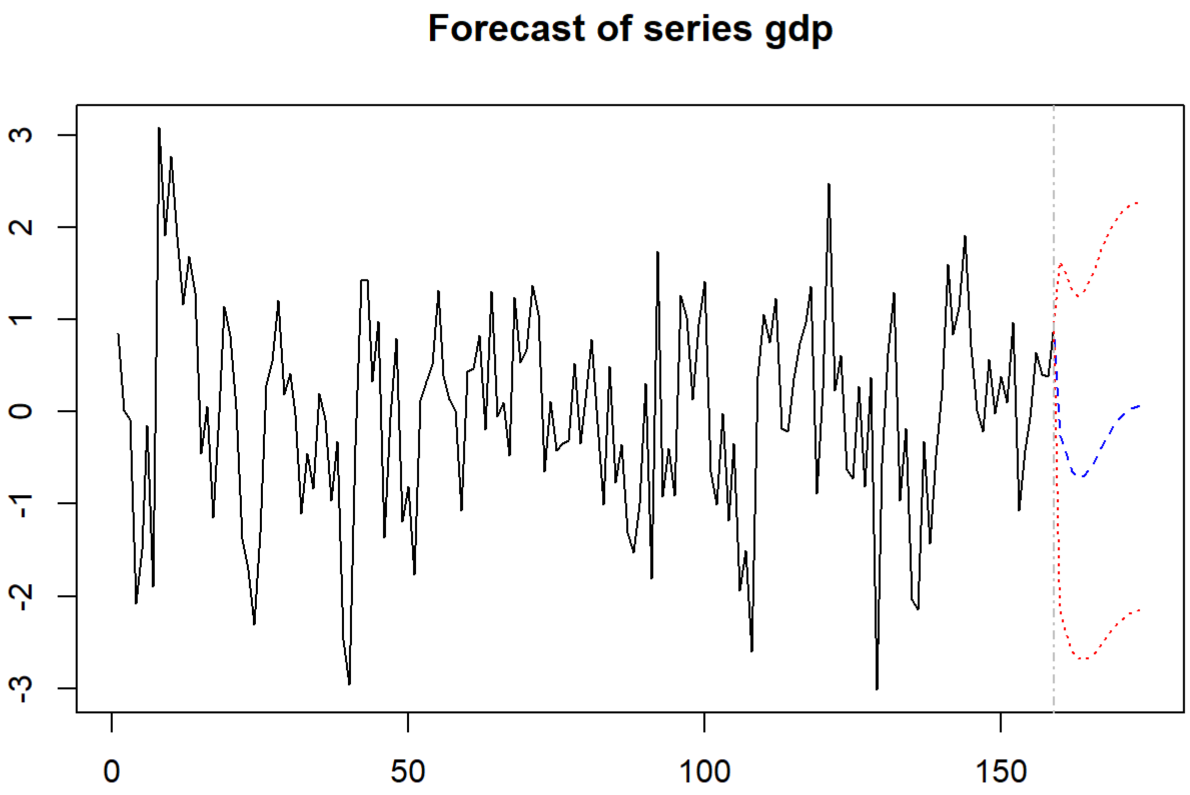
```
# graficando la diagnosis
plot(bv.serial, names = "gdp")
```



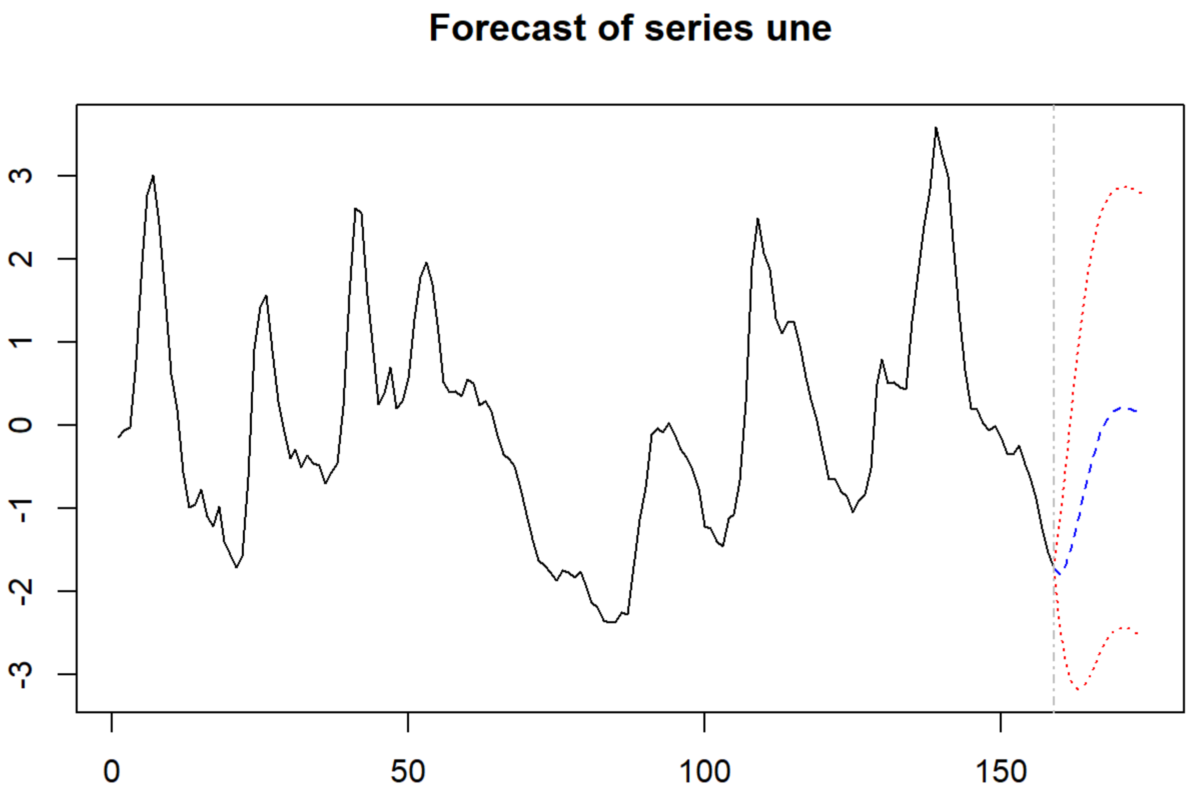

```
plot(bv.serial, names = "une")
```



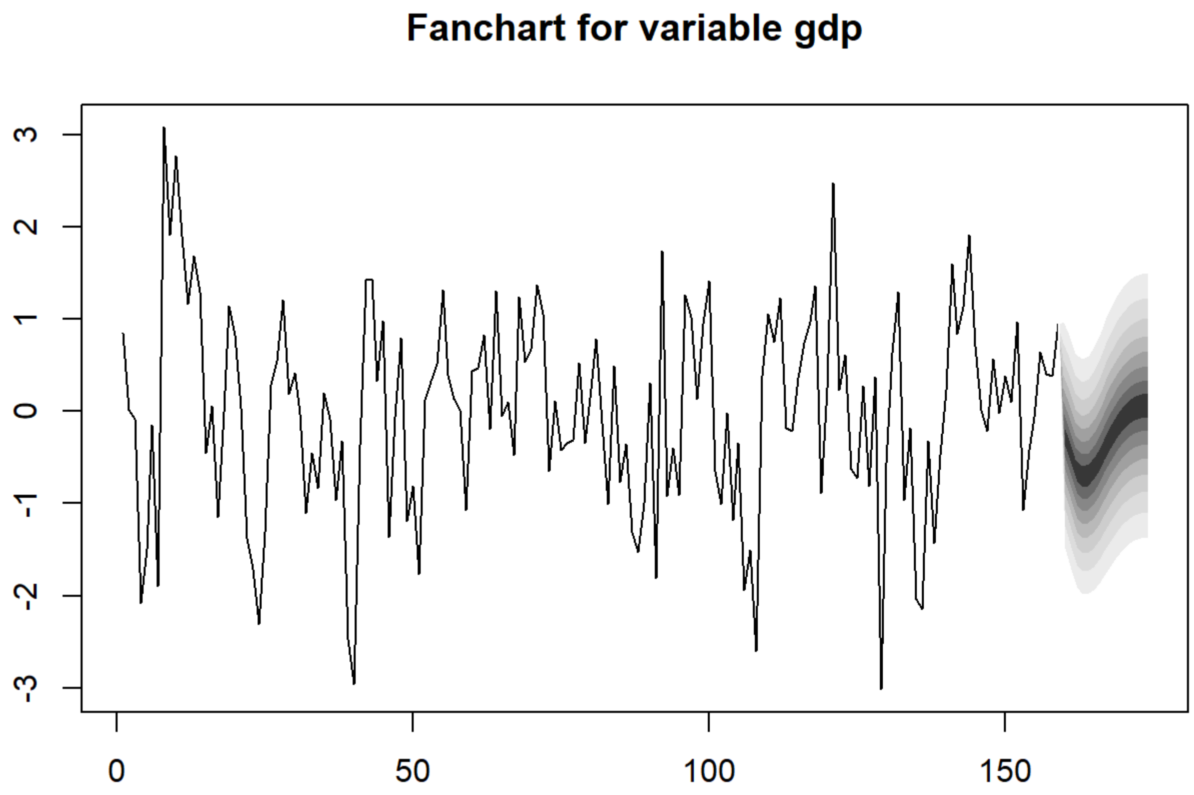

```
# Forecasting usando el modelo VAR (Hallando los pronósticos)
predictions <- predict(var.a, n.ahead = 15, ci = 0.95)
plot(predictions, names = "gdp")
```



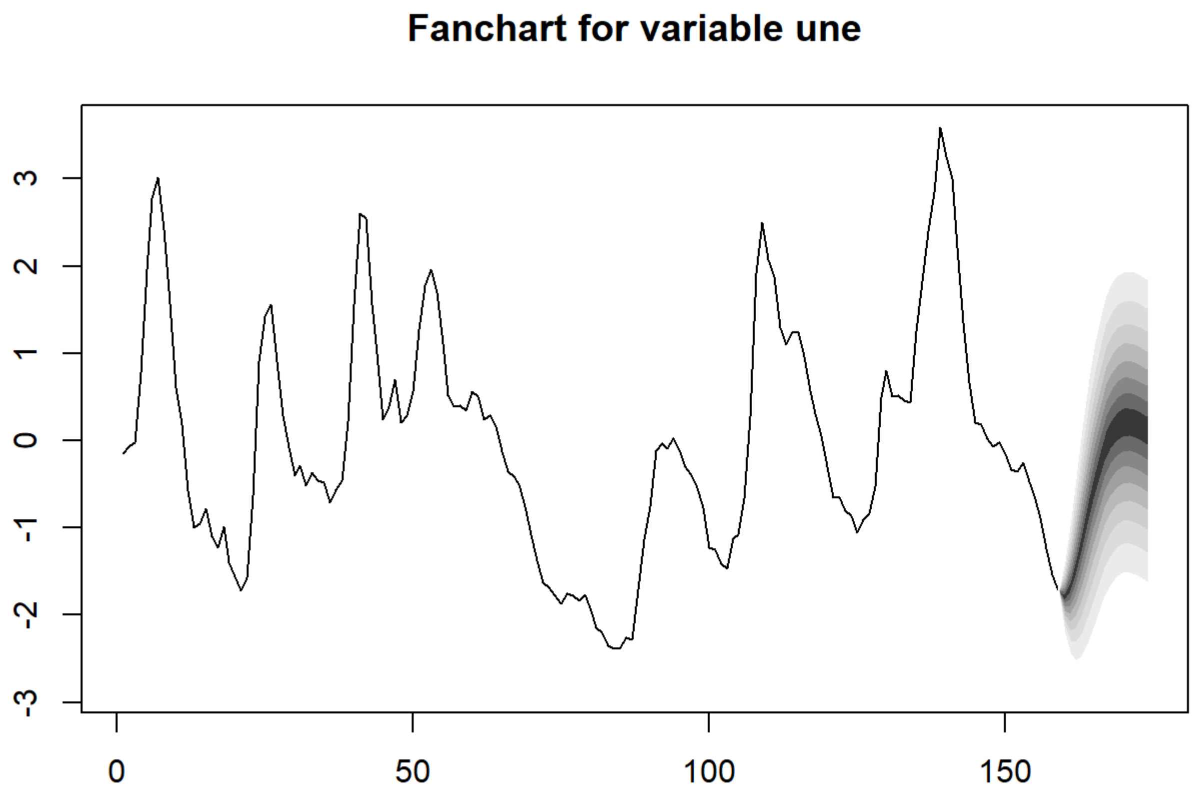

```
predictions <- predict(var.a, n.ahead = 15, ci = 0.95)
plot(predictions, names = "une")
```



```
# Otro gráfico
fanchart(predictions, names = "gdp")
```

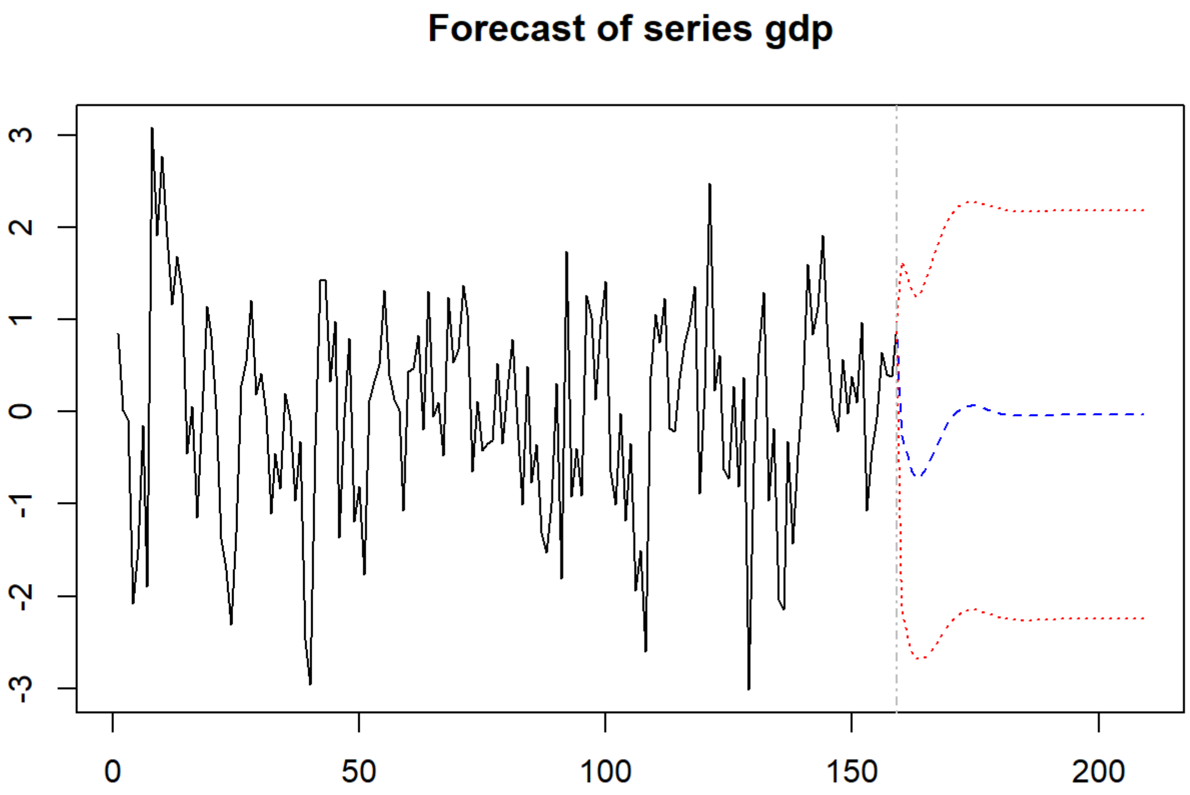



```
fanchart(predictions, names = "une")
```

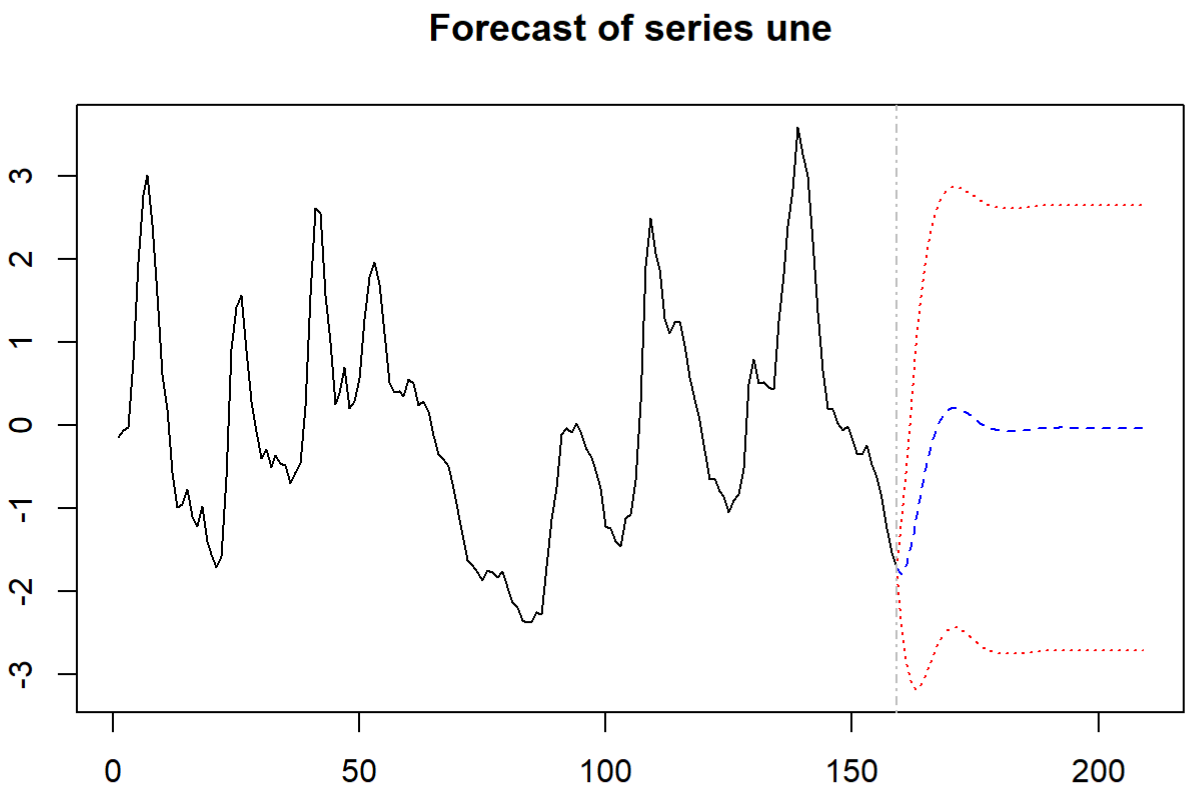


Más predicciones hacia el futuro

```
# "n.ahead = 50" (perdemos precisión)
predictions <- predict(var.a, n.ahead = 50, ci = 0.95)
plot(predictions, names = "gdp")
```

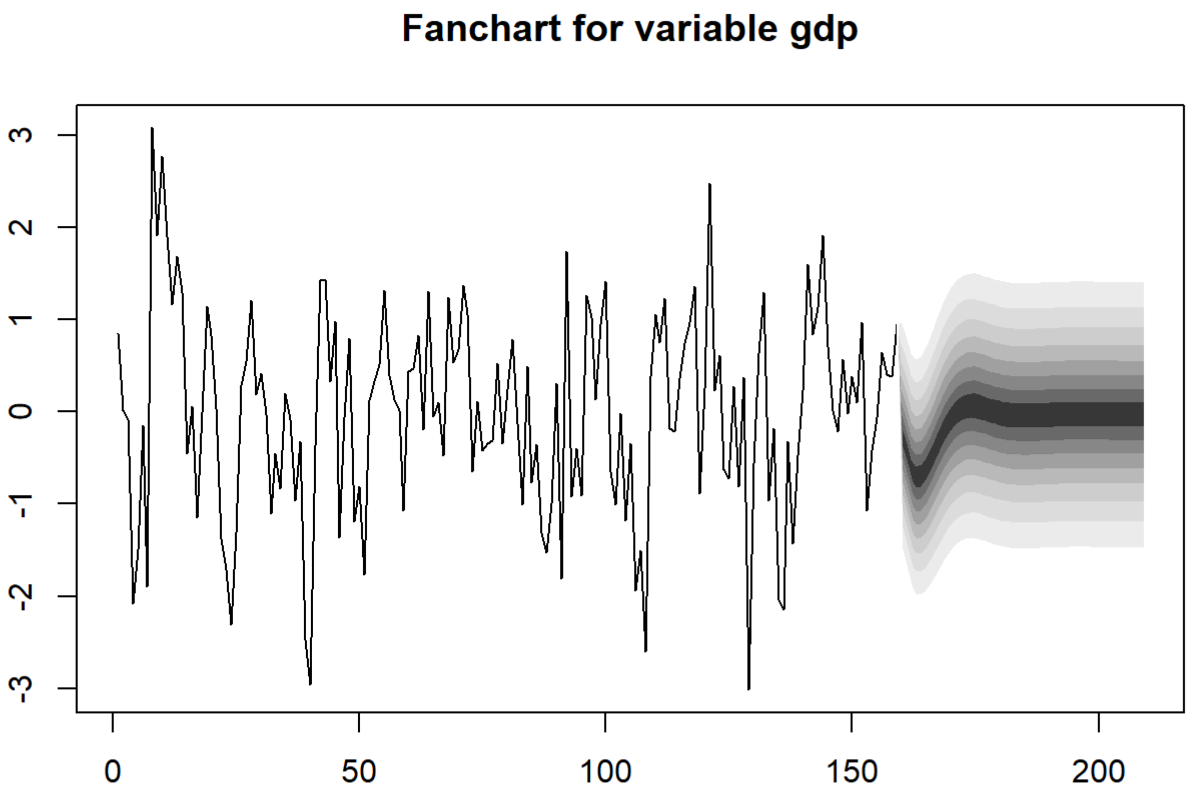



```
predictions <- predict(var.a, n.ahead = 50, ci = 0.95)
plot(predictions, names = "une")
```

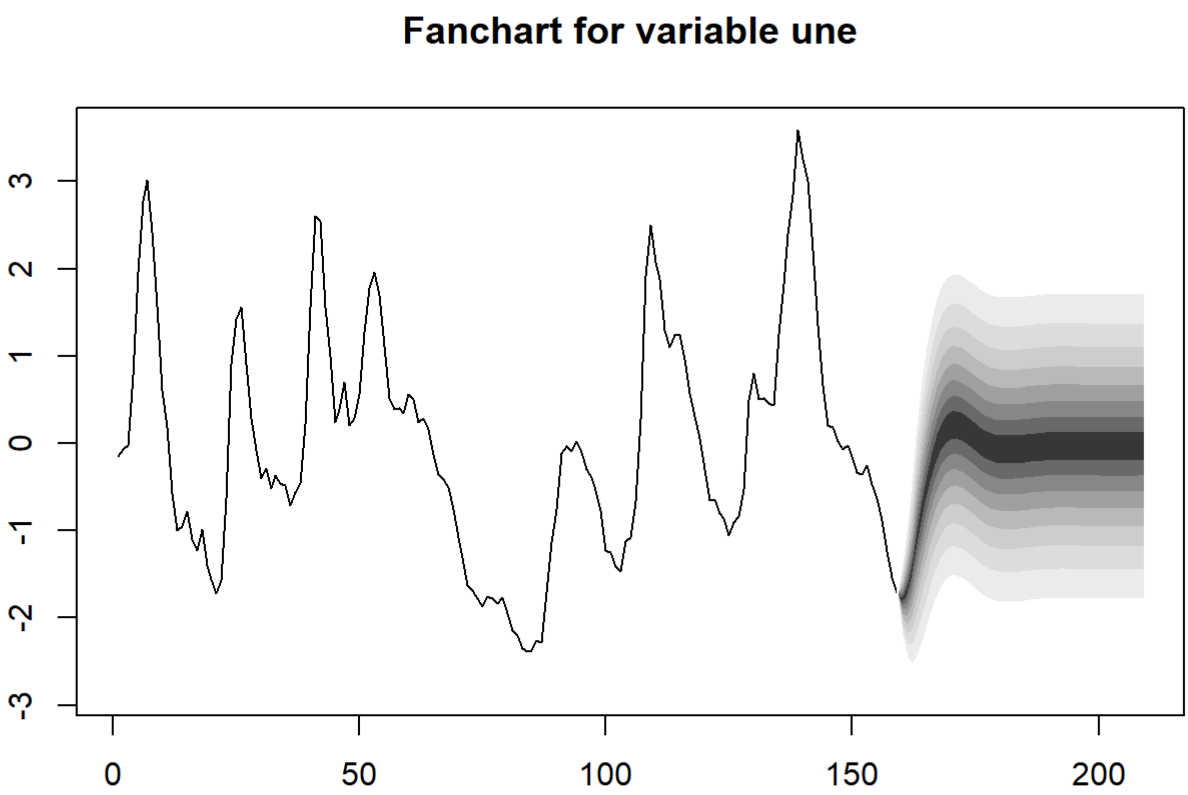


Se observa que se pierde precisión(capacidad predictiva)

```
# Otro gráfico
fanchart(predictions, names = "gdp")
```

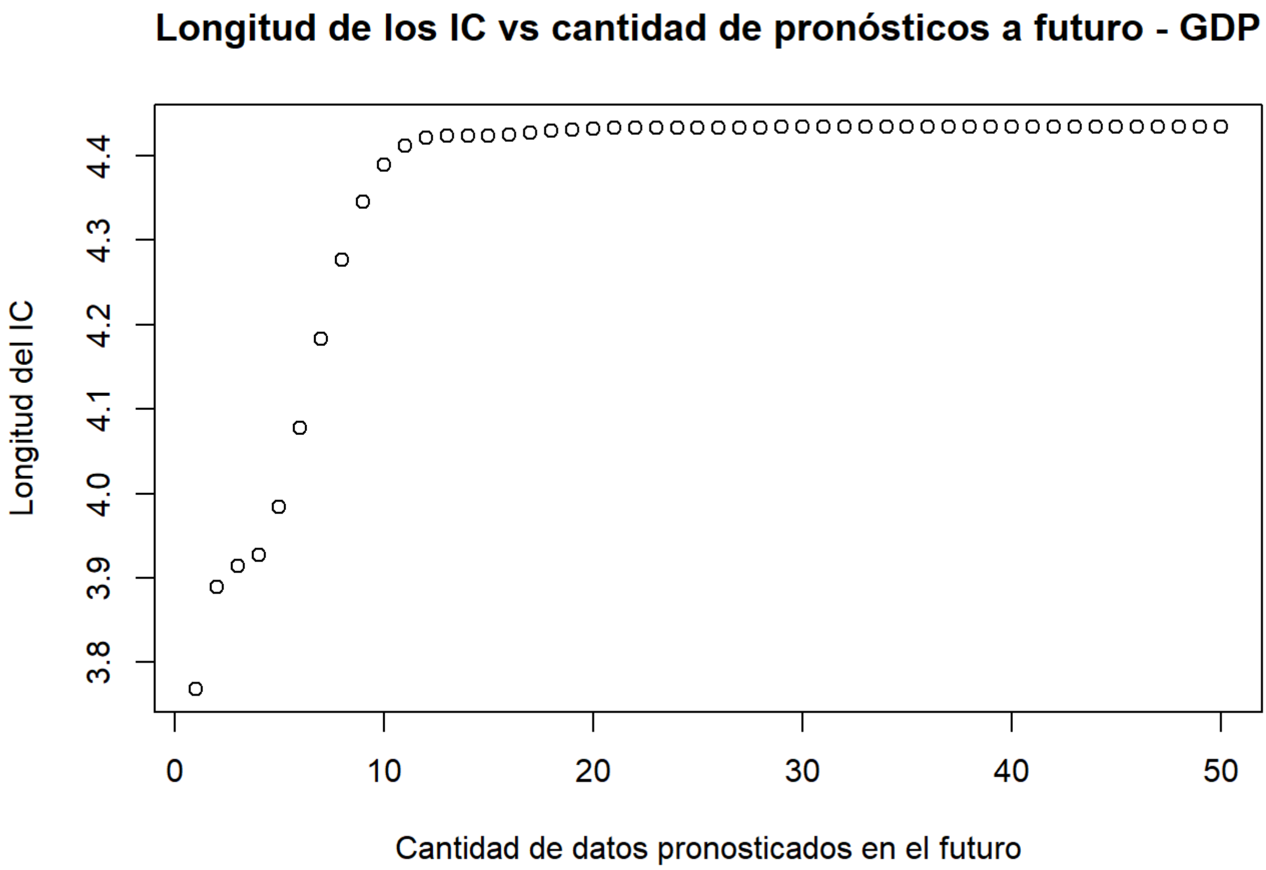



```
fanchart(predictions, names = "une")
```



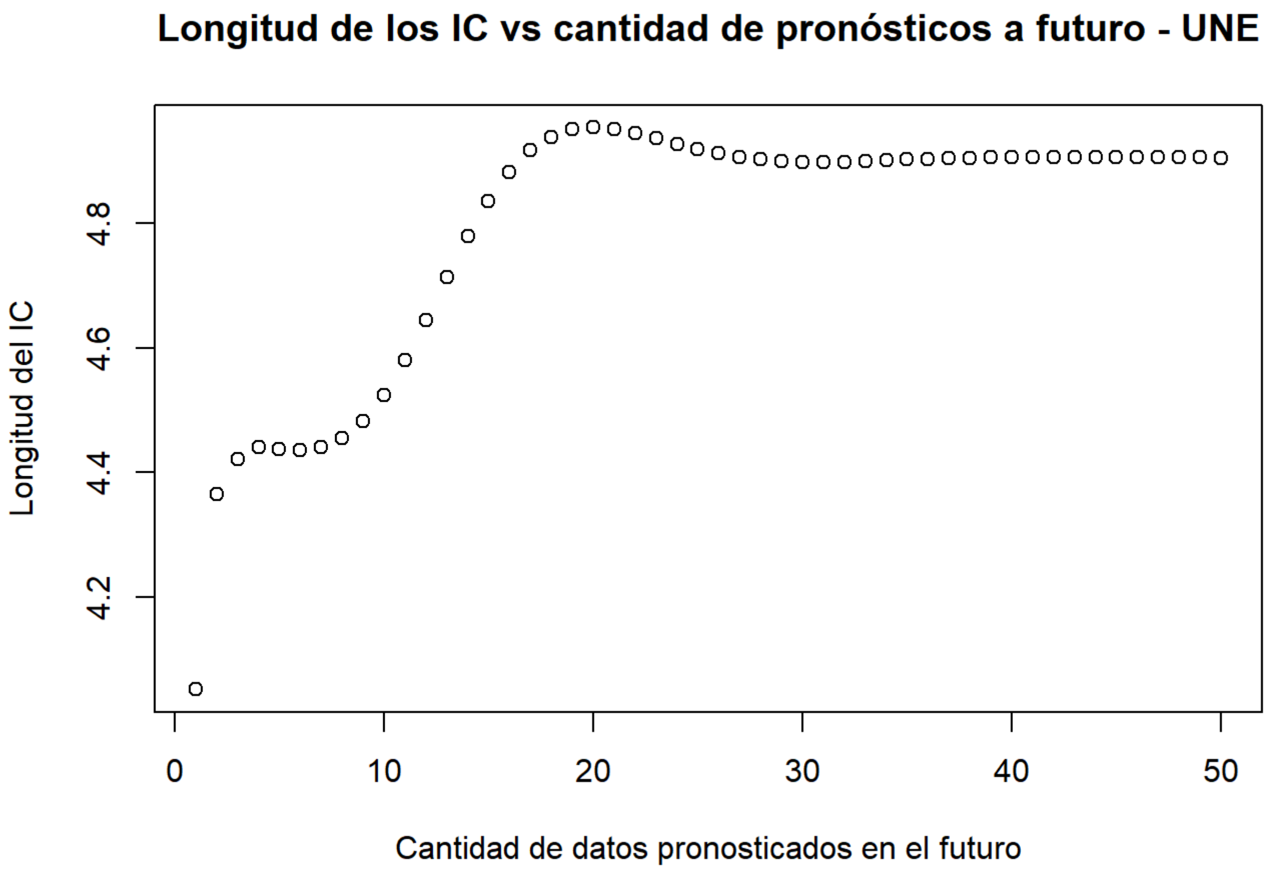
Calcular el tamaño de los intervalos de confianza

```
# preferentemente un 95% de confianza
diff_IC_gdp=predictions$fcst$gdp[,3]-predictions$fcst$gdp[,2] # retorna los intervalos de con
fianza
plot(diff_IC_gdp, main="Longitud de los IC vs cantidad de pronósticos a futuro - GDP", xlab
='Cantidad de datos pronosticados en el futuro', ylab='Longitud del IC')
```



El gráfico indica que no se debería pronosticar más de 12 datos al futuro con este modelo.

```
diff_IC_une=predictions$fcst$gdp[,3]-predictions$fcst$une[,2]
plot(diff_IC_une, main="Longitud de los IC vs cantidad de pronósticos a futuro - UNE", xlab
='Cantidad de datos pronosticados en el futuro', ylab='Longitud del IC')
```



El

gráfico indica que no se debería pronosticar más de 12 o 16 datos al futuro con este modelo.

De los gráficos se observa que cuantos más datos al futuro pronosticamos, mayor error se obtiene, perdiendo la precision.