Clase 1 - Macroeconomía

Jorge Sarcos González

22 de agosto de 2024

```
################################# MACROECONOMETRICS: INTRODUCTION
############### Mg. Pablo Villacampa Portuguez
############### Session 01: Introduction to time series and first steps in RStudio

### I. PASOS INICIALES:

# Antes de nada, limpiamos el workspace por si hubiera alguna información cargada
rm(list = ls())

# Limpiamos la consola
cat("\014")
```

```
# Otra opción es utilizar solamente setwd y asignar el nombre de la carpeta donde están los d
atos.
#  setwd("C:/Users/pvillacampa/Desktop/RStudio/Datos") #no olvidar las comillas y el cambio d
el slash.

#Paquete para importar archivos Excel
#istall.packages("readxl") #Leerá archivos excel
library(readxl)
```

```
### II. LEEMOS LAS SERIES DE TIEMPO

# Leemos la serie de tiempo, desde un archivo csv - Leemos el archivo PBI
library(readr)
pbi <- read_csv("C:/Users/Home/Downloads/pbi.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 228 Columns: 1
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## num (1): PBI
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(pbi)
dim(pbi)
```

```
## [1] 228   1
```

```
summary(pbi)
```

```
##       PBI
##  Min.   :  87.0
##  1st Qu.: 994.2
##  Median :1202.5
##  Mean   :1196.2
##  3rd Qu.:1439.5
##  Max.   :2087.0
```

```
# Leemos ahora la serie de tiempo desde un archivo xlsx
library(readxl)
base01 <- read_excel("C:/Users/Home/Downloads/Libro1 (2).xlsx",
    sheet = "datos")
View(base01)
```

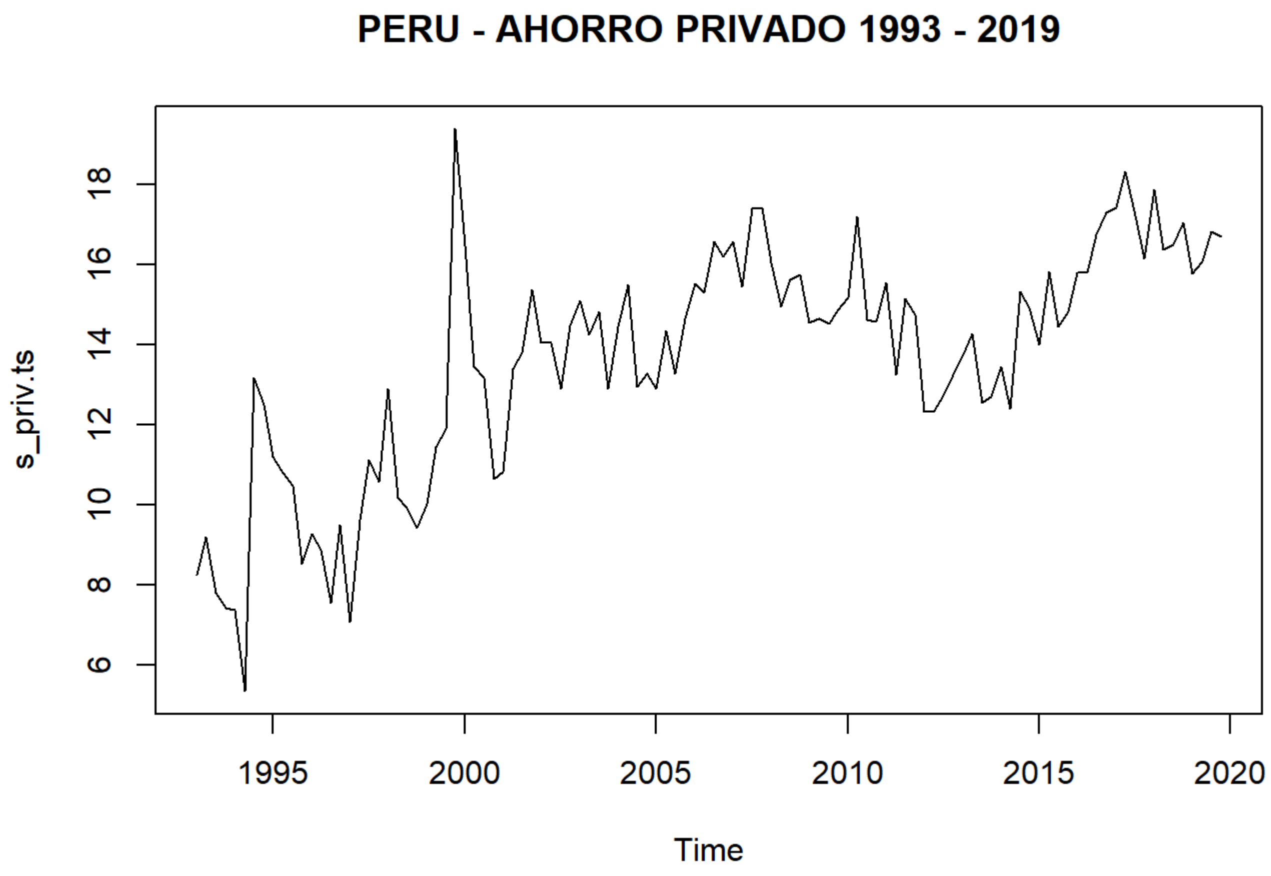
```
summary(base01)
```

```
##      s_priv           i_priv            tot             ipm       
##  Min.   : 5.351   Min.   : 8.736   Min.   :3.971   Min.   :4.109  
##  1st Qu.:12.339   1st Qu.:13.802   1st Qu.:4.075   1st Qu.:4.216  
##  Median :14.318   Median :16.550   Median :4.375   Median :4.520  
##  Mean   :13.615   Mean   :16.190   Mean   :4.337   Mean   :4.497  
##  3rd Qu.:15.661   3rd Qu.:18.616   3rd Qu.:4.573   3rd Qu.:4.763  
##  Max.   :19.382   Max.   :25.675   Max.   :4.739   Max.   :4.867  
##       ipx             cca              s_pub              i_pub      
##  Min.   :3.485   Min.   :-9.125   Min.   :-0.1753   Min.   :2.653  
##  1st Qu.:3.678   1st Qu.:-5.129   1st Qu.: 2.3540   1st Qu.:3.527  
##  Median :4.364   Median :-2.775   Median : 3.4349   Median :4.552  
##  Mean   :4.229   Mean   :-3.035   Mean   : 3.9653   Mean   :4.440  
##  3rd Qu.:4.739   3rd Qu.:-1.260   3rd Qu.: 6.0050   3rd Qu.:5.106  
##  Max.   :4.998   Max.   : 5.163   Max.   : 9.0197   Max.   :6.297  
```

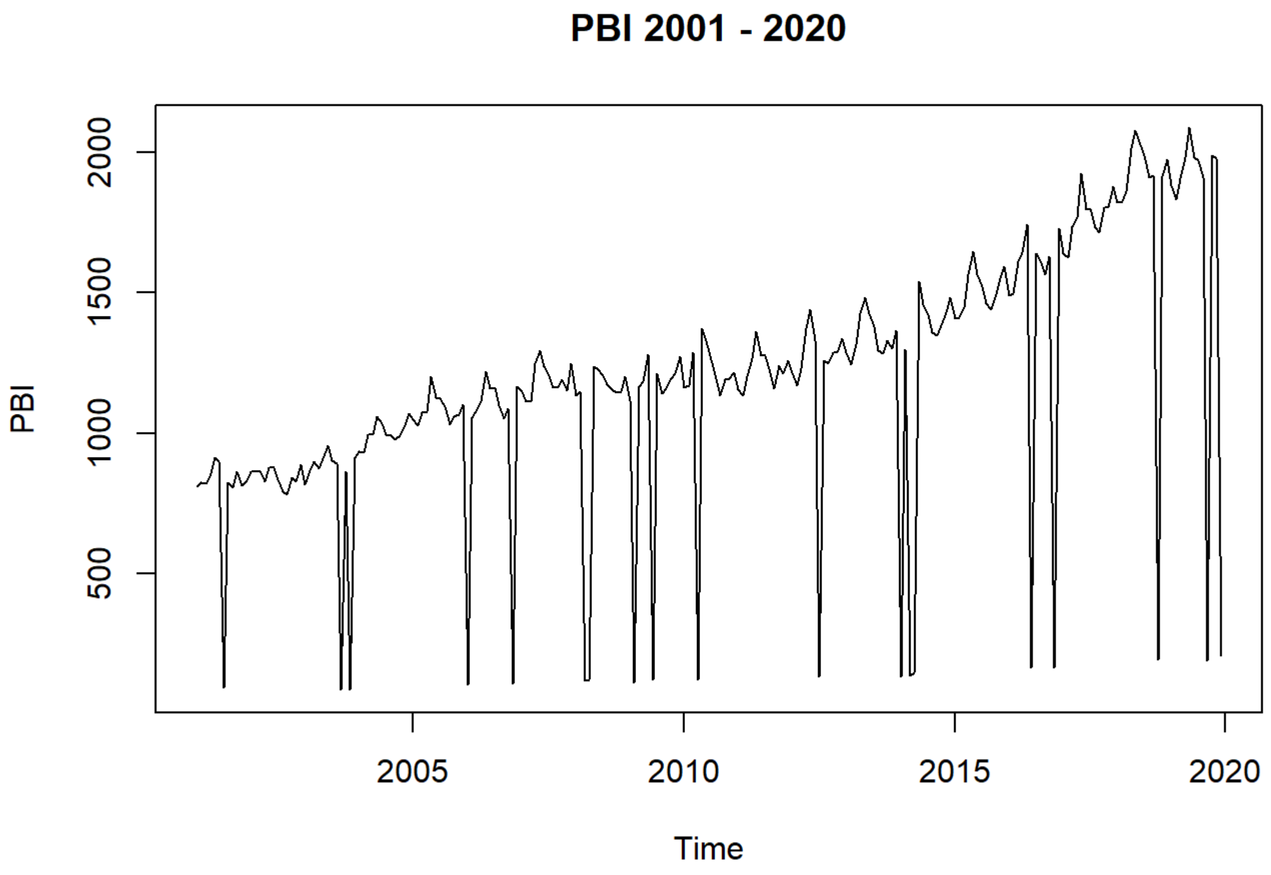
```
# Creacion de una serie de tiempo para el PBI
pbi.ts<-ts(pbi, start=c(2001,1), end=c(2019,12), frequency = 12)

pbi_2.ts<-ts(pbi, start=c(2001,1), frequency = 12)

# Creacion de una serie de tiempo para las variables del archivo base01
attach(base01)
s_priv.ts=ts(s_priv, start = c(1993,1), end = c(2019,4),frequency = 4)
i_priv.ts=ts(i_priv, start = c(1993,1), end = c(2019,4),frequency = 4)
i_pub.ts=ts(i_pub, start = c(1993,1), end = c(2019,4),frequency = 4)
tot.ts=ts(tot, start = c(1993,1), end = c(2019,4),frequency = 4)
ipm.ts=ts(ipm, start = c(1993,1), end = c(2019,4),frequency = 4)
detach(base01)

# Borrador ante consulta en clase.
s_priv_02.ts=ts(s_priv.ts, start = c(1993,1),frequency = 12)
```

```
options(bitmapType = "cairo")
```

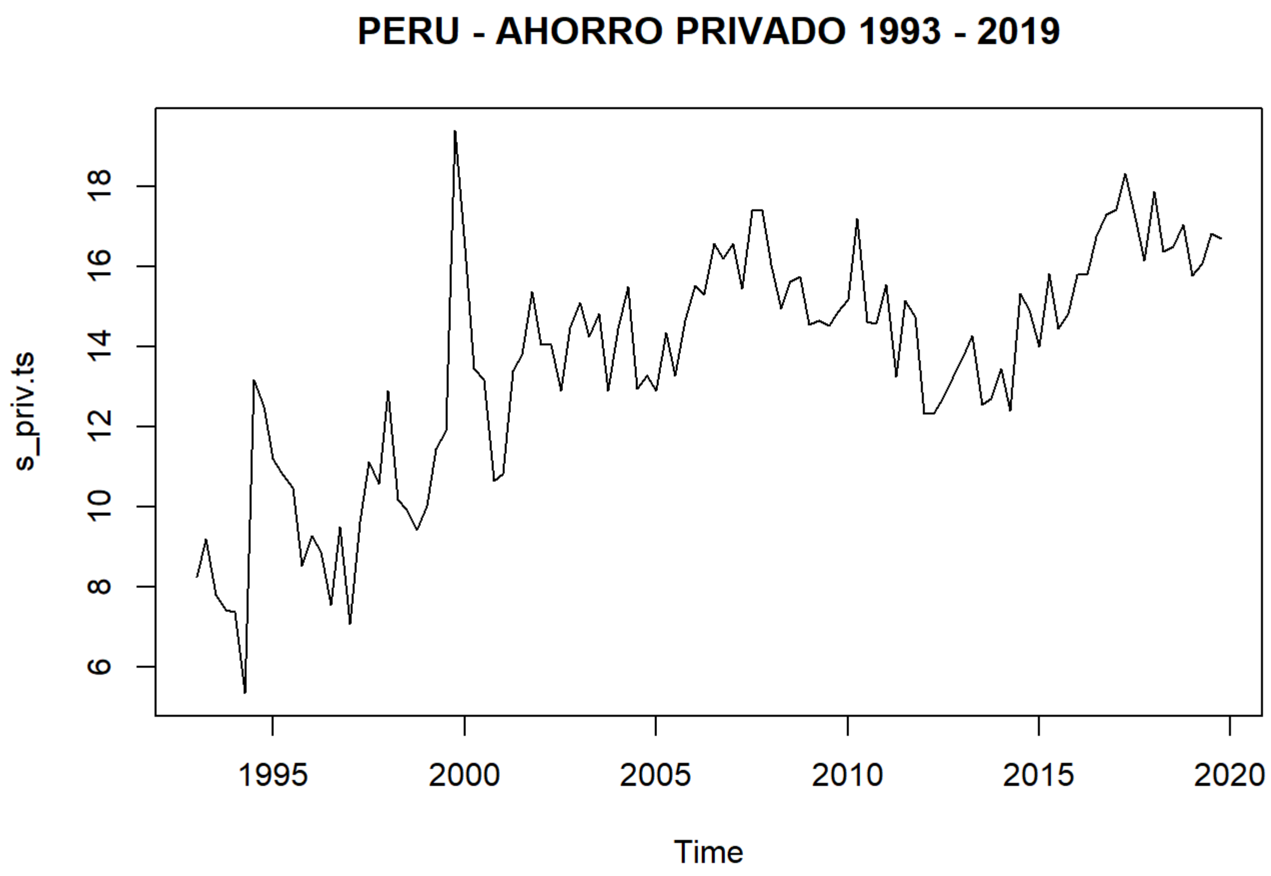
```
# Graficamos la serie de tiempo
plot(s_priv.ts, main="PERU - AHORRO PRIVADO 1993 - 2019")
```



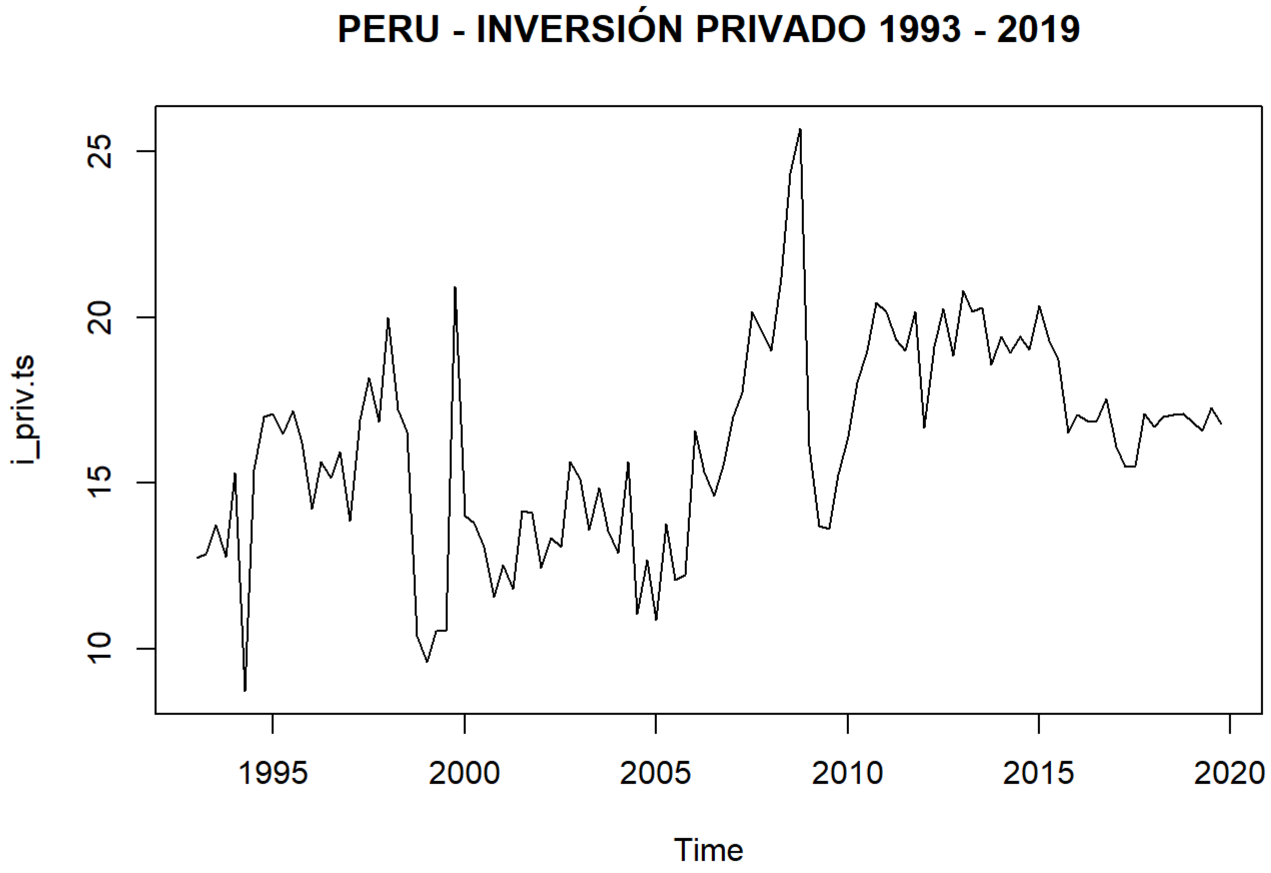
```
plot(pbi.ts, main = "PBI 2001 - 2020")
```



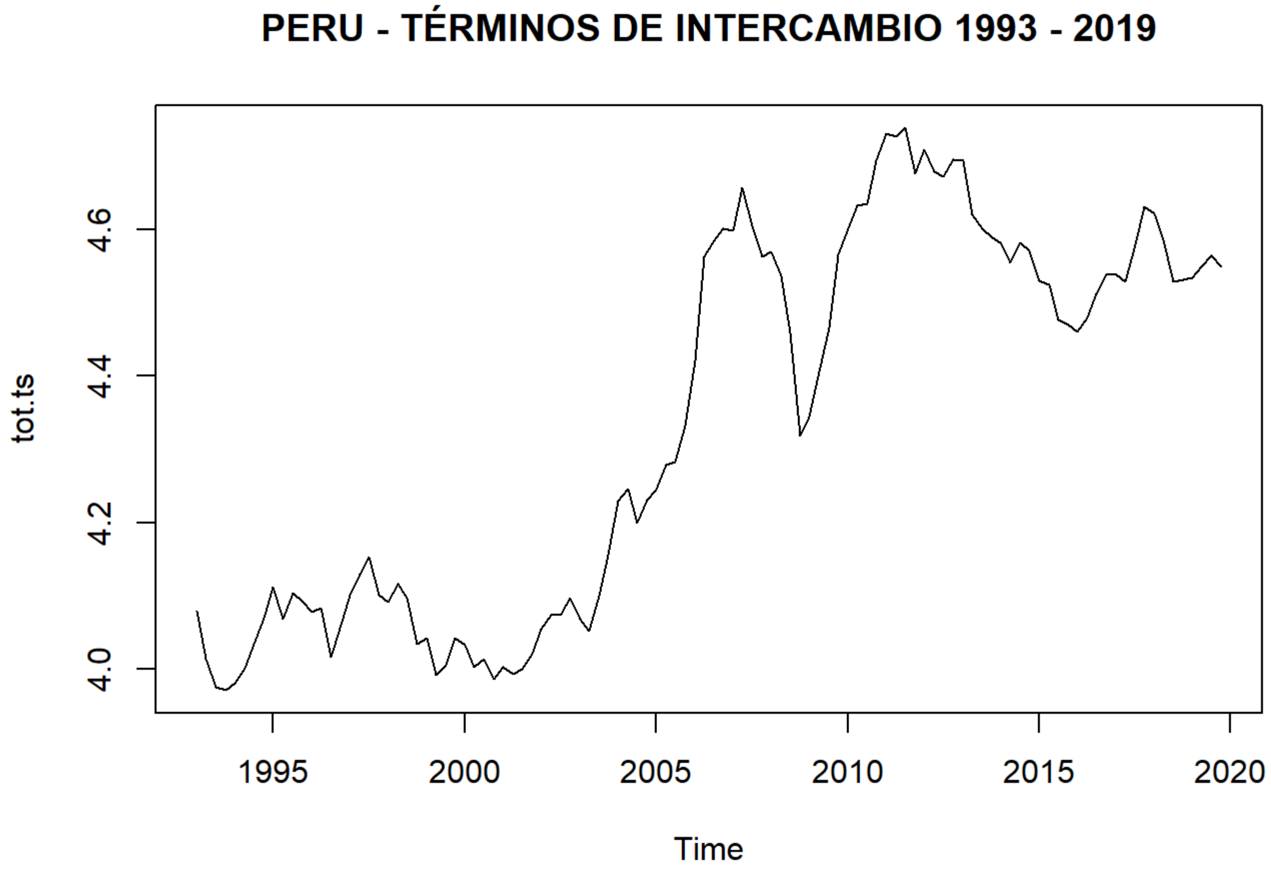
```
plot(s_priv.ts, main="PERU - AHORRO PRIVADO 1993 - 2019")
```



```
plot(i_priv.ts, main="PERU - INVERSIÓN PRIVADO 1993 - 2019")
```

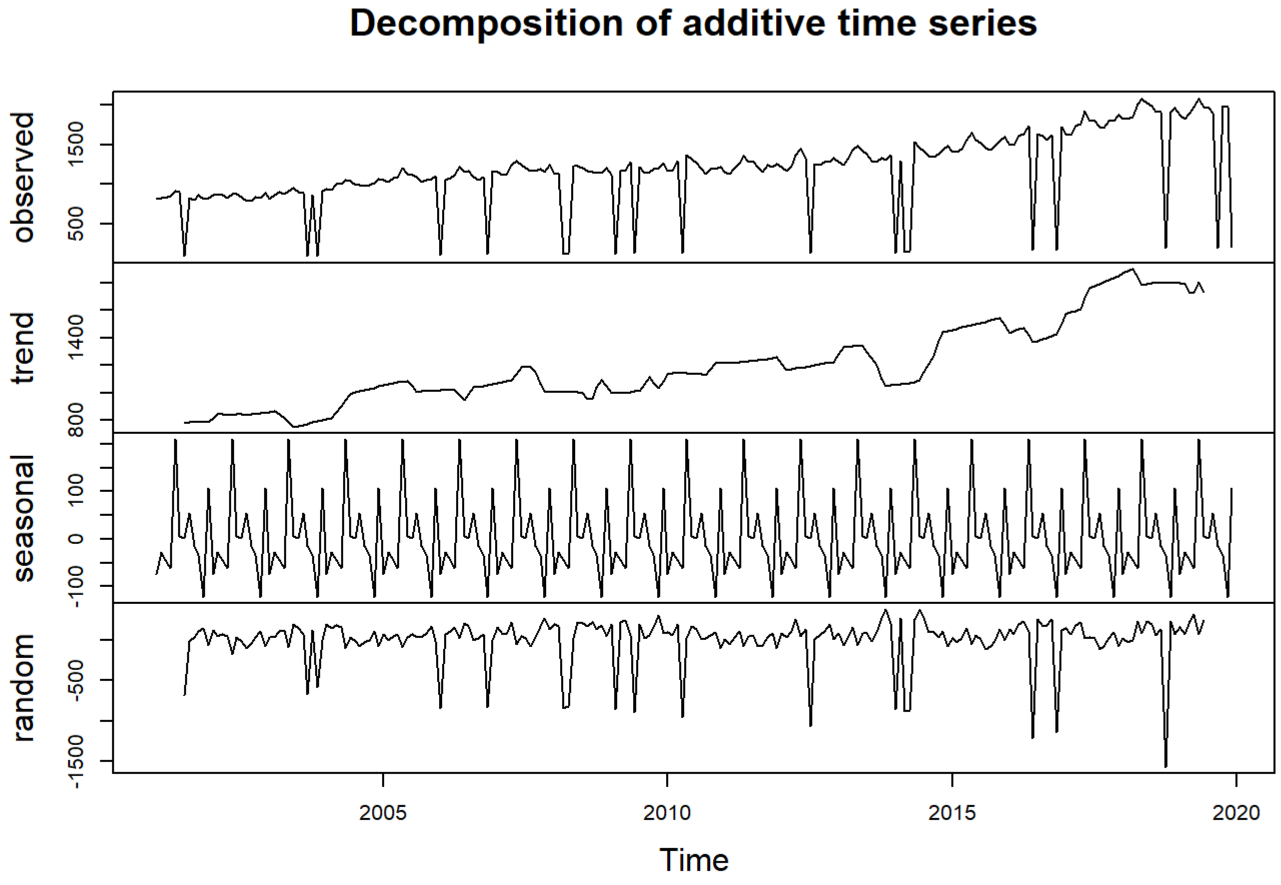

PERU - INVERSIÓN PRIVADO 1993 - 2019

```
plot(tot.ts, main="PERU - TÉRMINOS DE INTERCAMBIO 1993 - 2019")
```

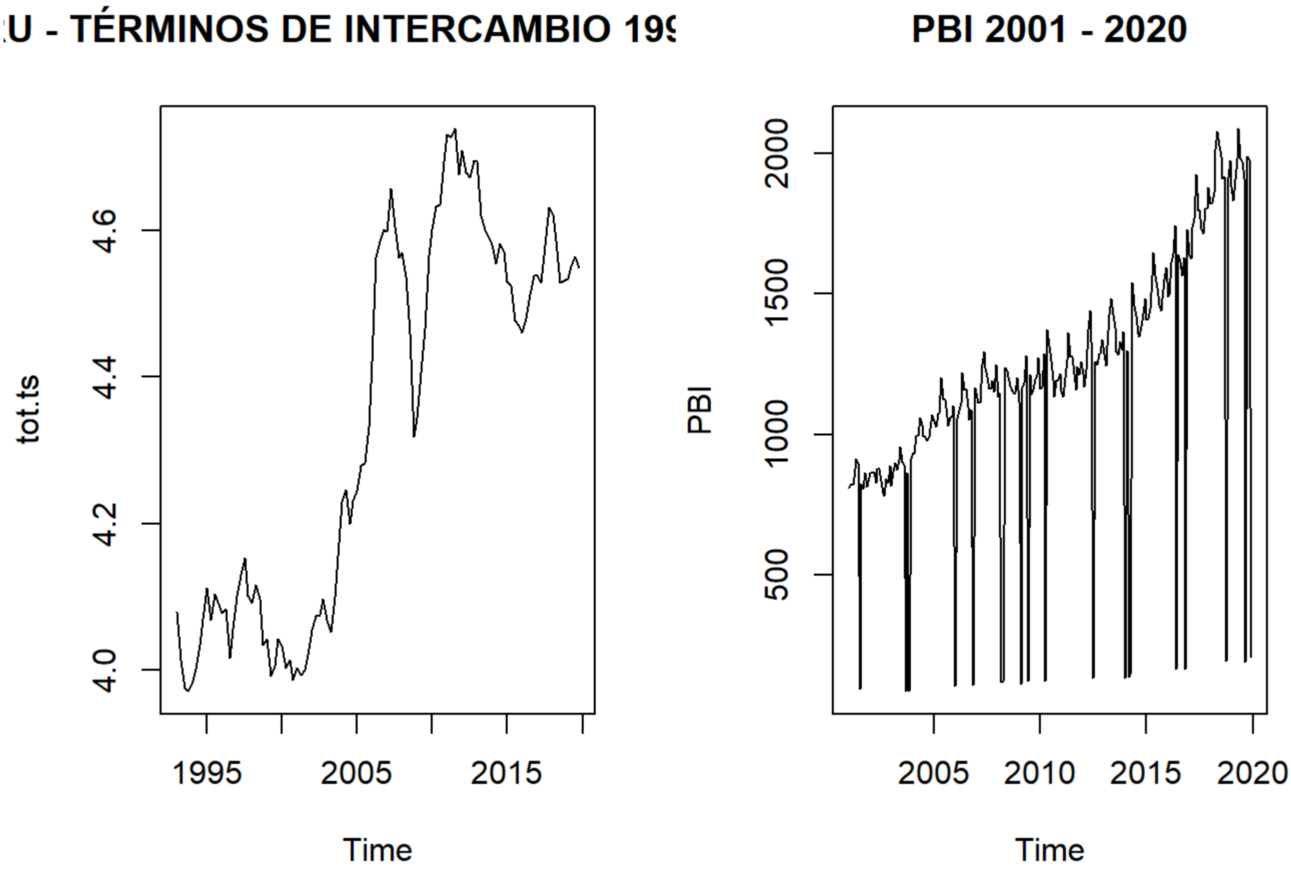

PERU - TÉRMINOS DE INTERCAMBIO 1993 - 2019

```
par(mfrow=c(1,1))
```

```
### III. ANÁLISIS INICIAL DE LAS SERIES TEMPORALES
# Descomponemos de manera inicial la serie: comando decompose
pbi.ts.desc=decompose(pbi.ts,type = c("additive"))
plot(pbi.ts.desc)
```

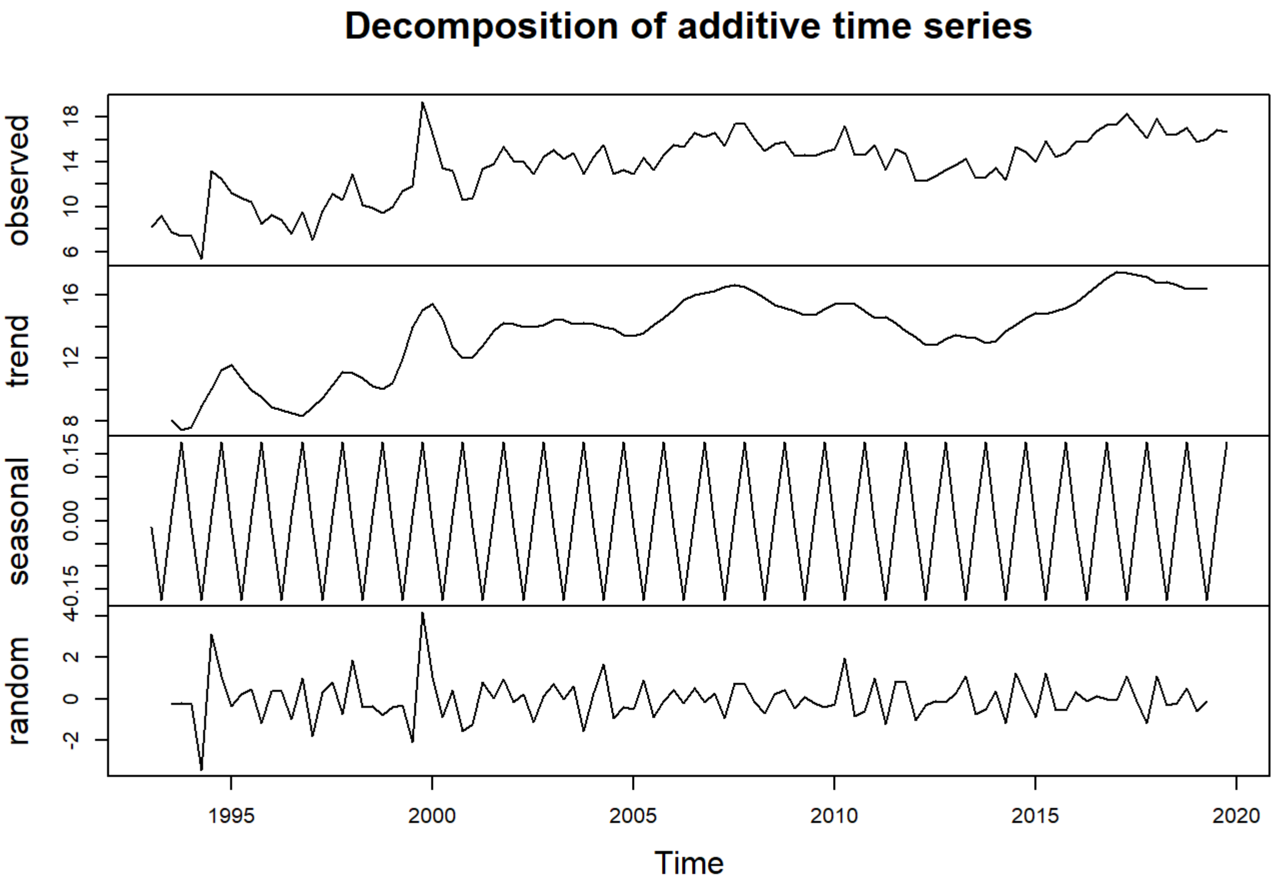


```
par(mfrow=c(1,2))
plot(tot.ts, main="PERU - TÉRMINOS DE INTERCAMBIO 1993 - 2019")
plot(pbi.ts, main="PBI 2001 - 2020")
```

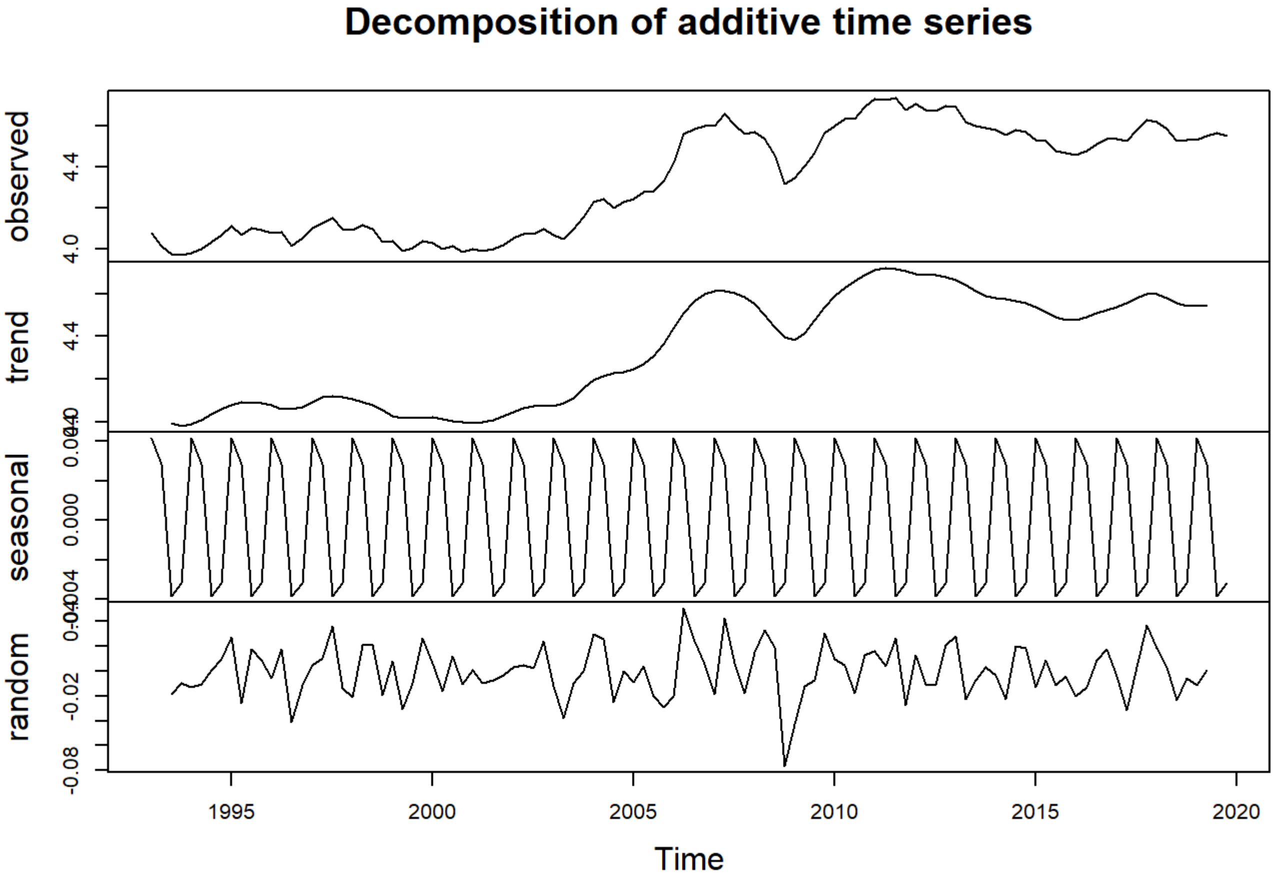


```
par(mfrow=c(1,1))
```

```
s_priv.ts.desc=decompose(s_priv.ts)
plot(s_priv.ts.desc)
```

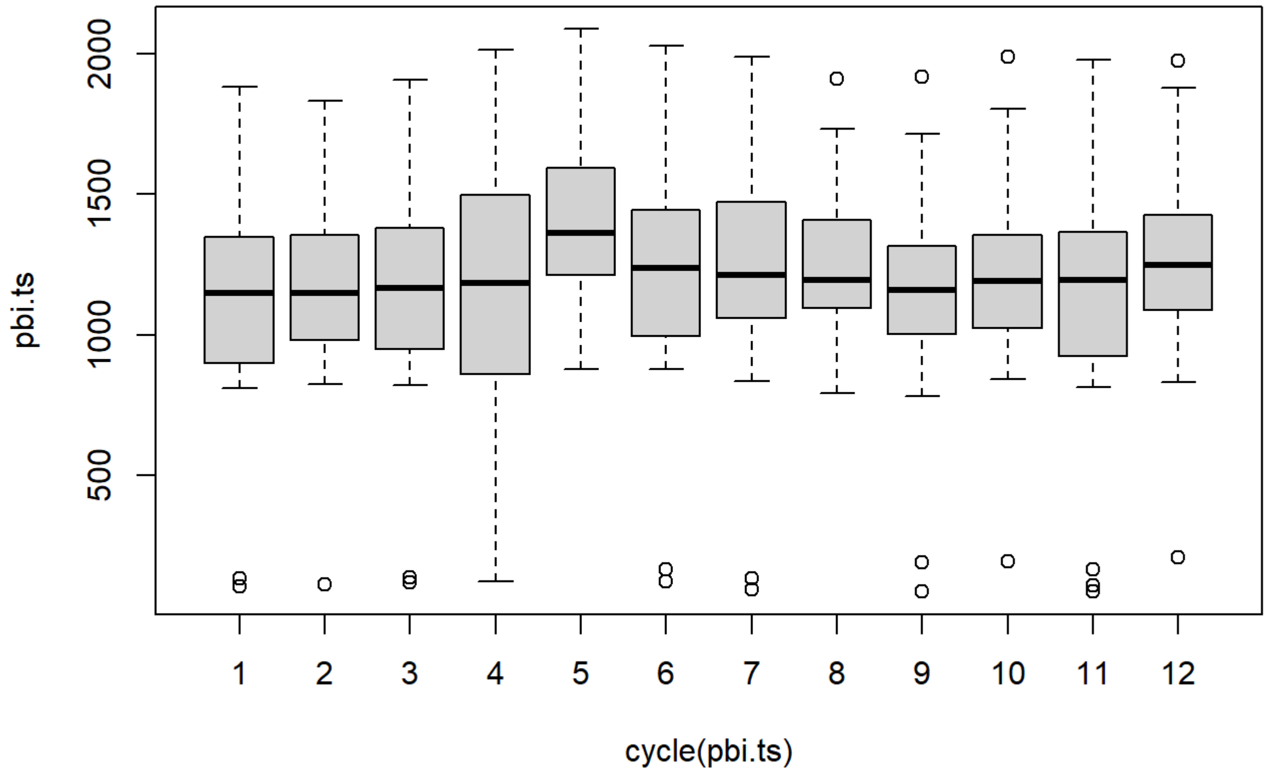


```
tot.ts.desc=decompose(tot.ts)
plot(tot.ts.desc)
```



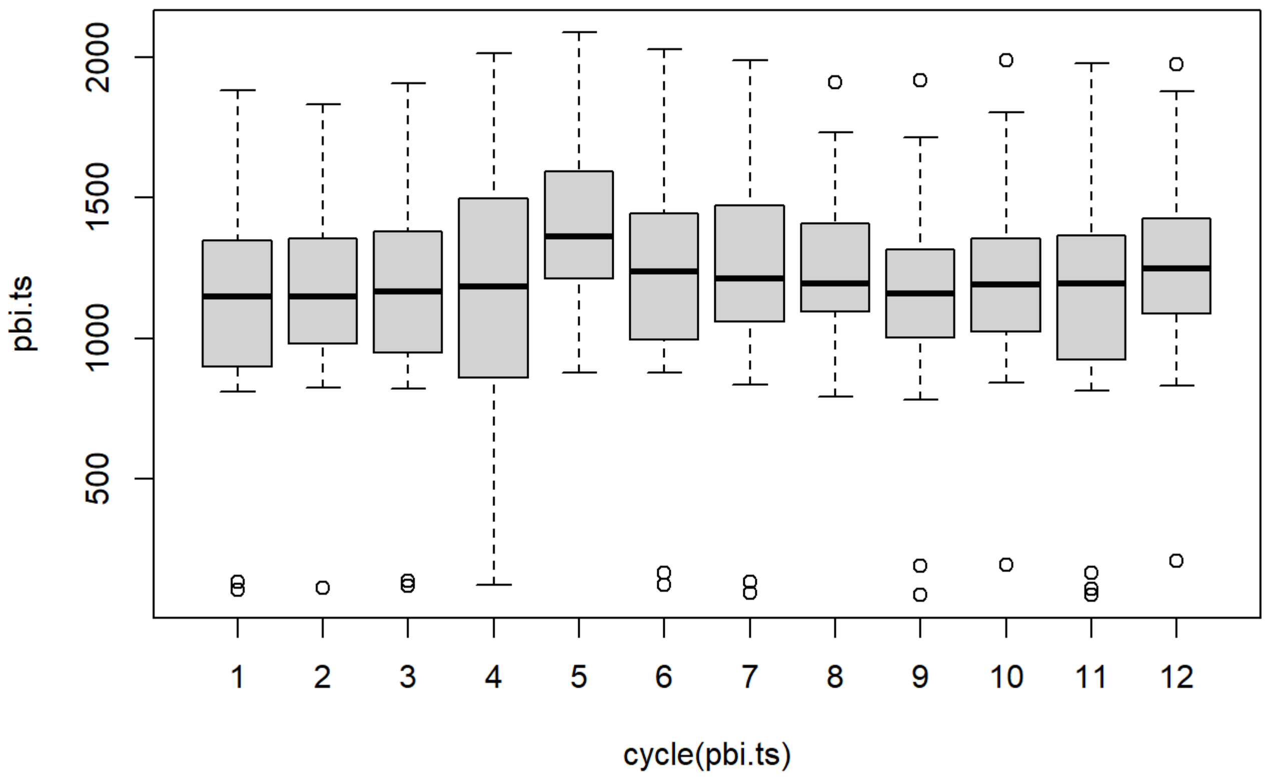
#Realizamos un gráfico de caja y usamos el comando ciclo para ver la estacionalidad y su influencia mensual.

```
boxplot((pbi.ts~cycle(pbi.ts)))
```



```
#Realizamos un resumen estadístico de las cajas
```

```
summary<-boxplot(pbi.ts~cycle(pbi.ts))$stats
```

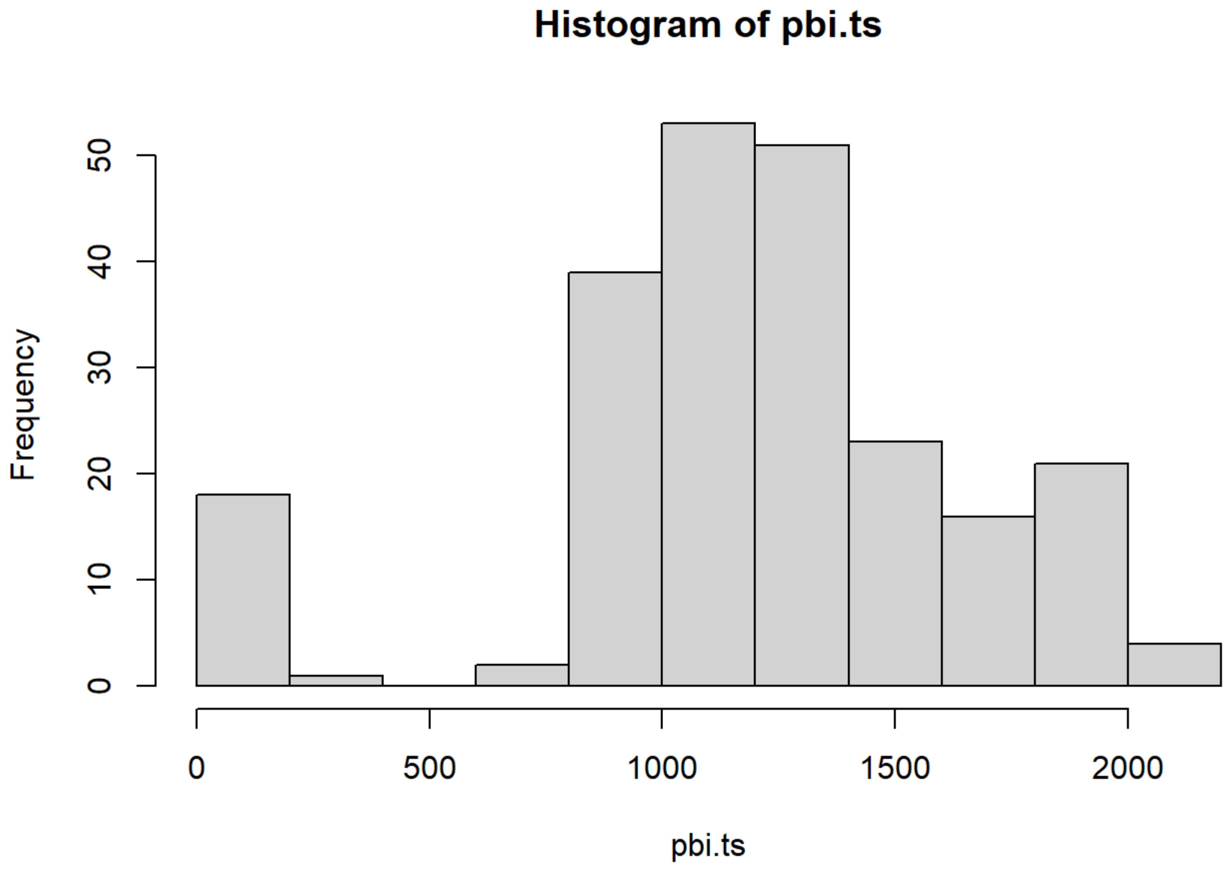


```
colnames(summary)<-c("Enero","Febrero", "MArzo", "Abril", "Mayo","Junio","Julio","Agosto","Setiembre","Octubre","Noviembre","Diciembre")
```

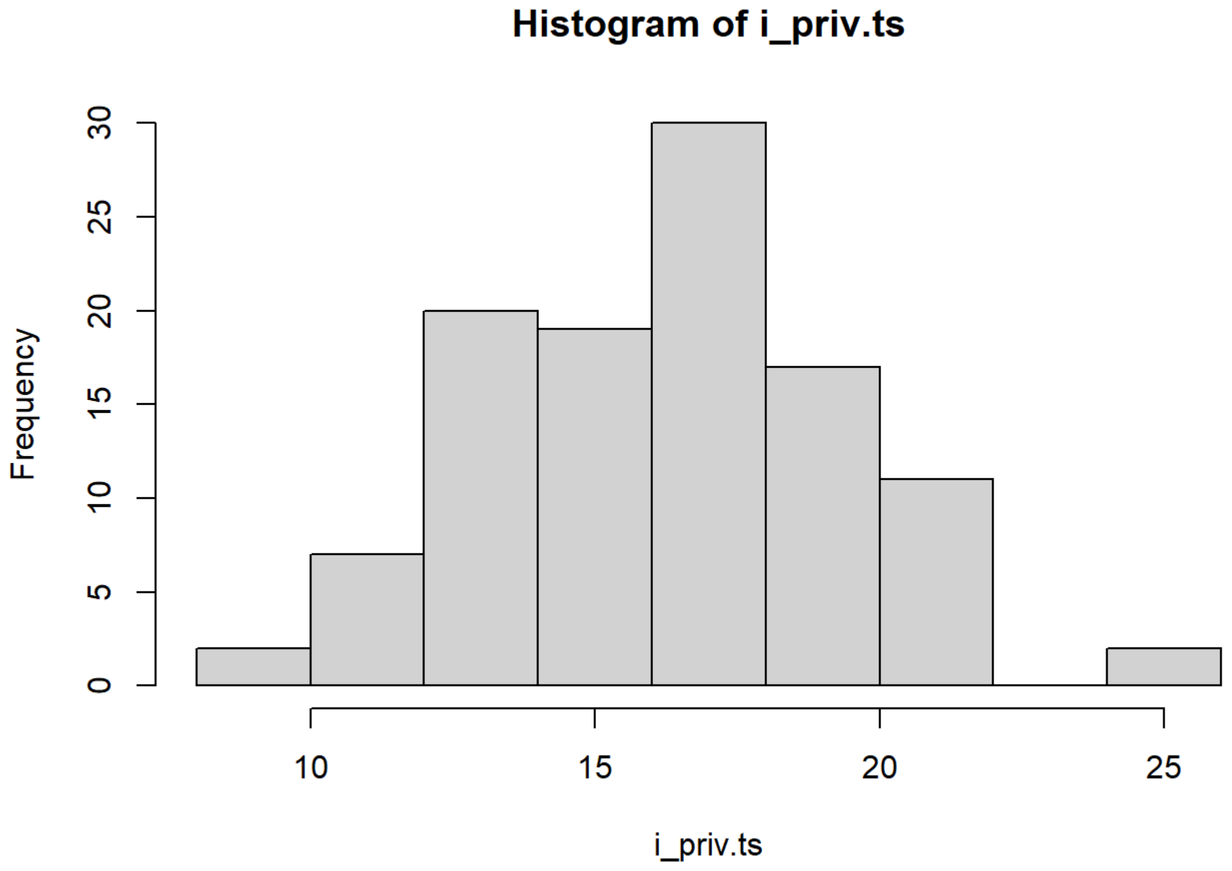
```
rownames(summary)<-c("Min","First Quartile","Median","Third Quartile ","Max")
summary
```

```
##                   Enero Febrero MArzo Abril   Mayo  Junio  Julio Agosto
## Min               809.0   824.0   821   123  878.0  879.0  835.0    791
## First Quartile    900.5   980.0   949   860 1212.0  995.5 1058.5   1096
## Median           1149.0  1148.0  1168  1185 1361.0 1238.0 1214.0   1196
## Third Quartile   1347.0  1355.5  1381  1496 1594.5 1442.5 1473.5   1410
## Max              1881.0  1832.0  1908  2015 2087.0 2029.0 1989.0   1732
##                  Setiembre Octubre Noviembre Diciembre
## Min                    783     842       813     831.0
## First Quartile        1004    1025       924    1086.5
## Median                1158    1193      1194    1247.0
## Third Quartile        1317    1356      1365    1424.5
## Max                   1716    1804      1979    1879.0
```

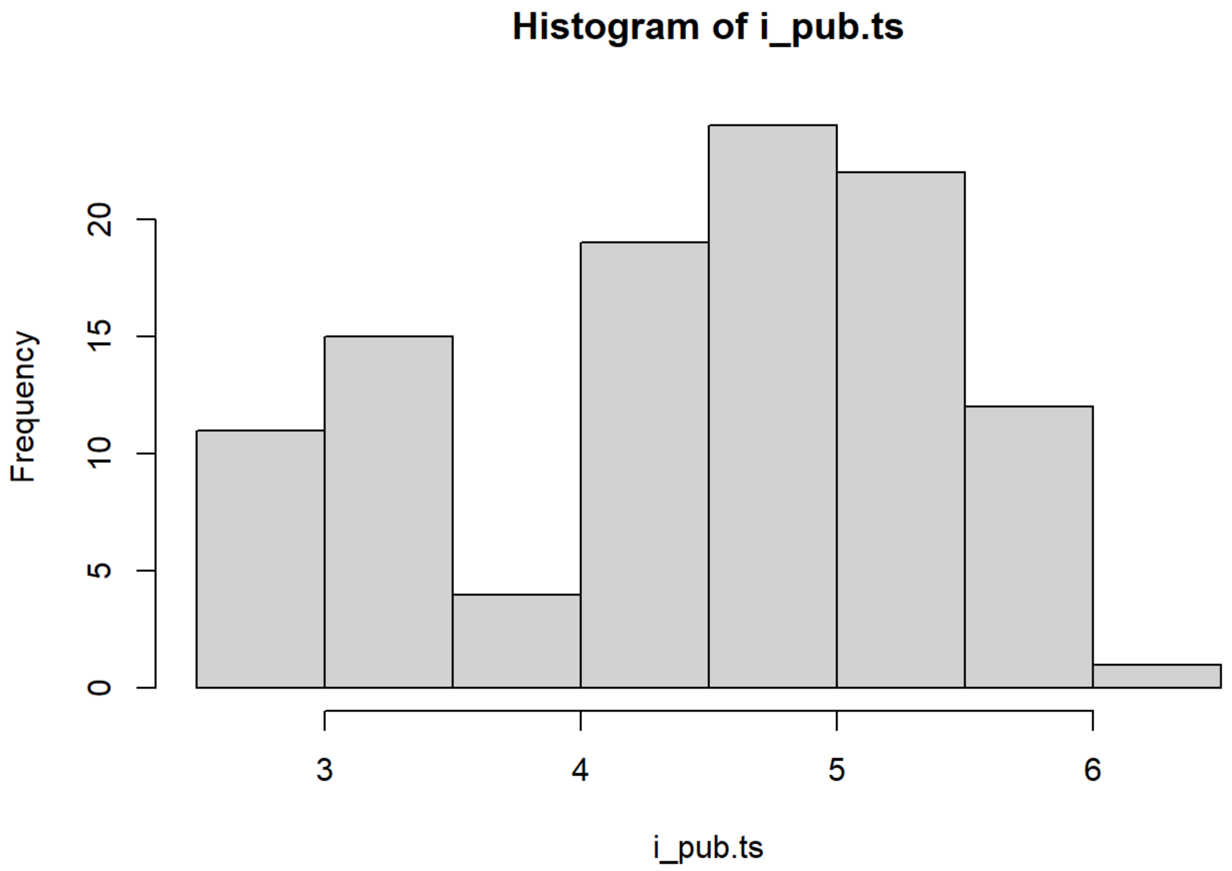
```
#Histograma
hist(pbi.ts)
```



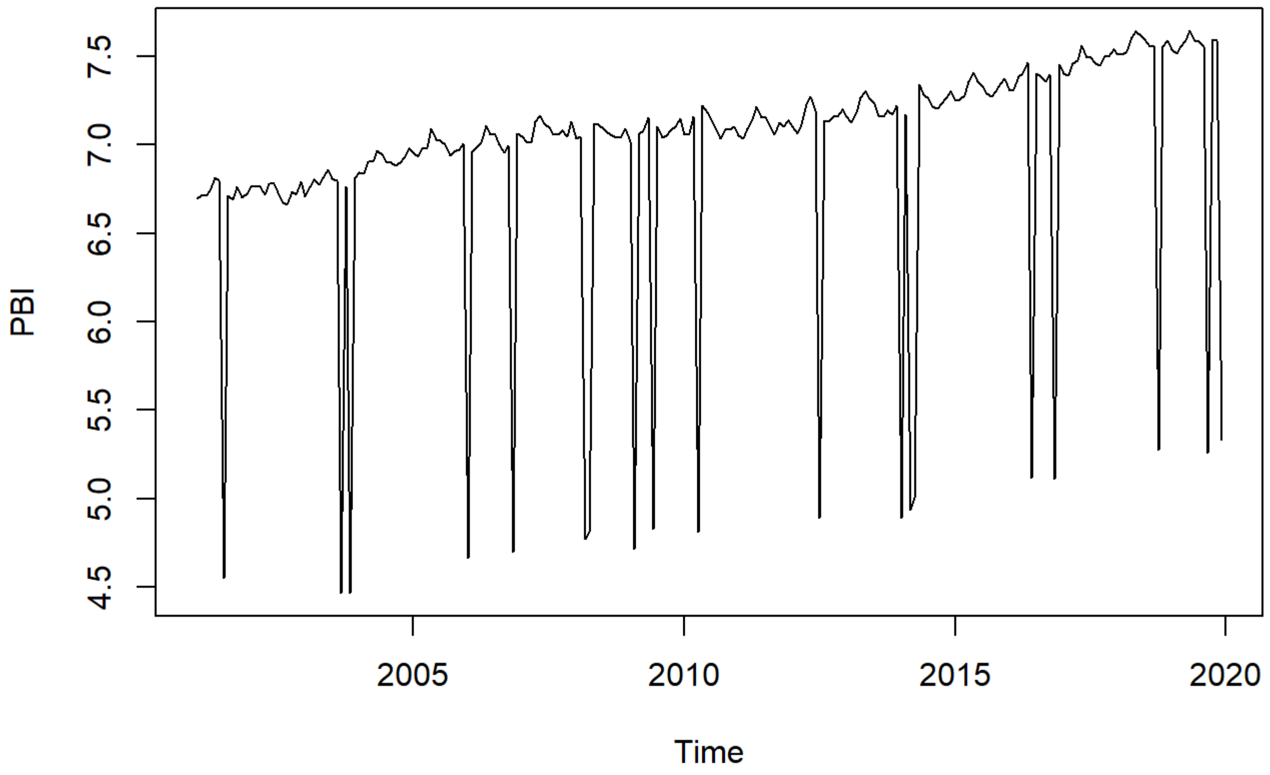
```
hist(i_priv.ts)
```



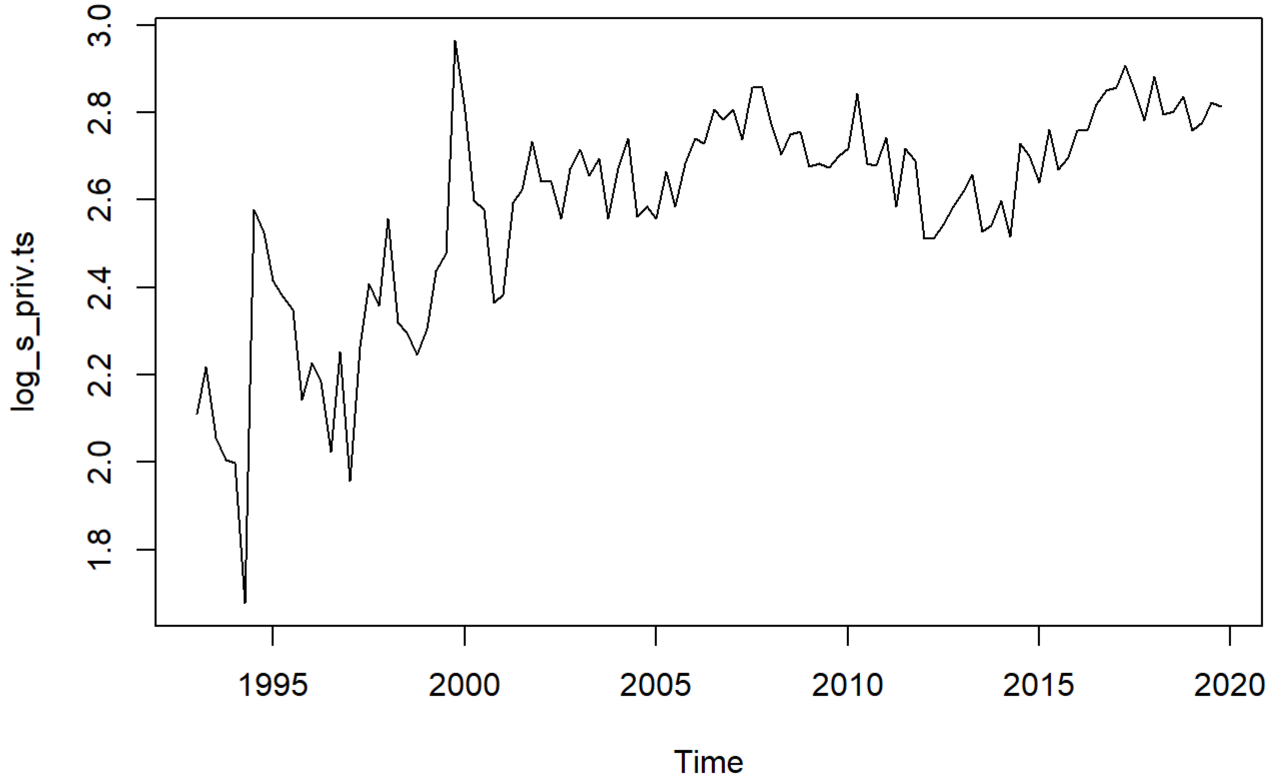
```
hist(i_pub.ts)
```



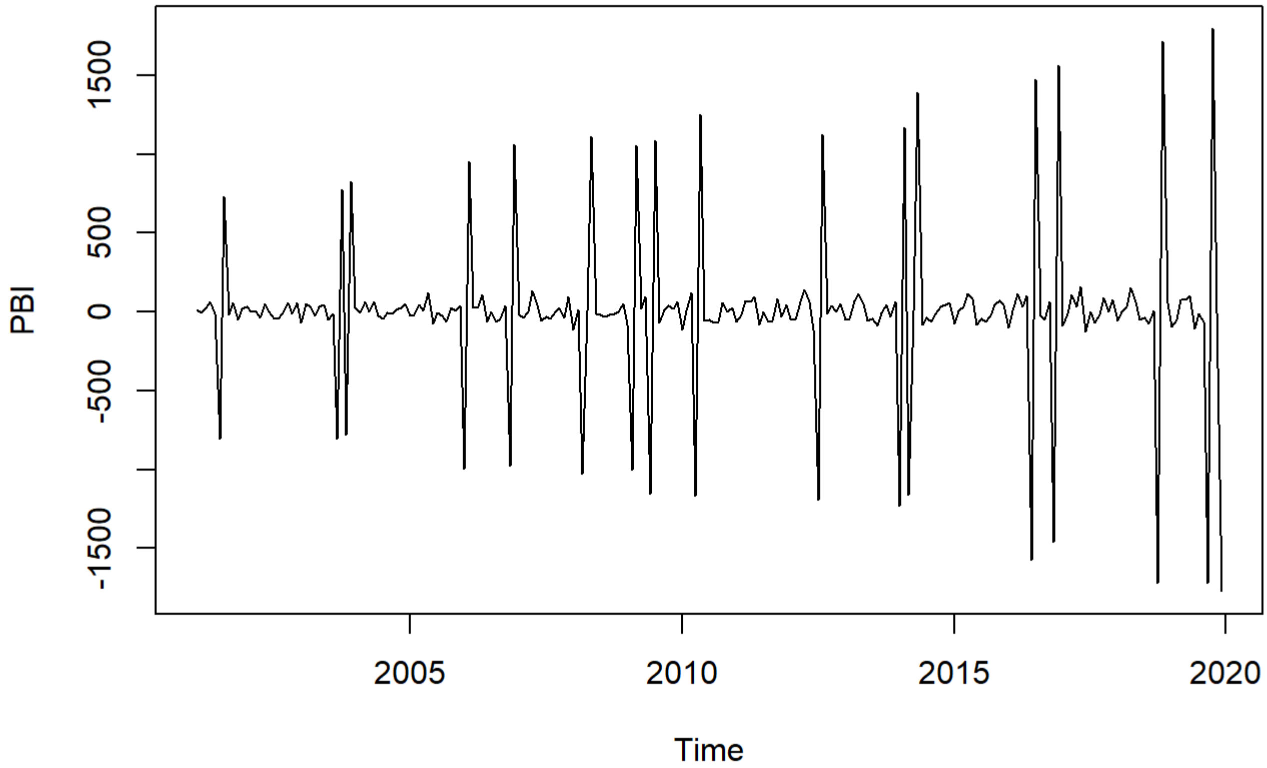
```
# Convertimos la serie a logaritmos
log_pbi.ts<-log(pbi.ts)
plot(log_pbi.ts)
```



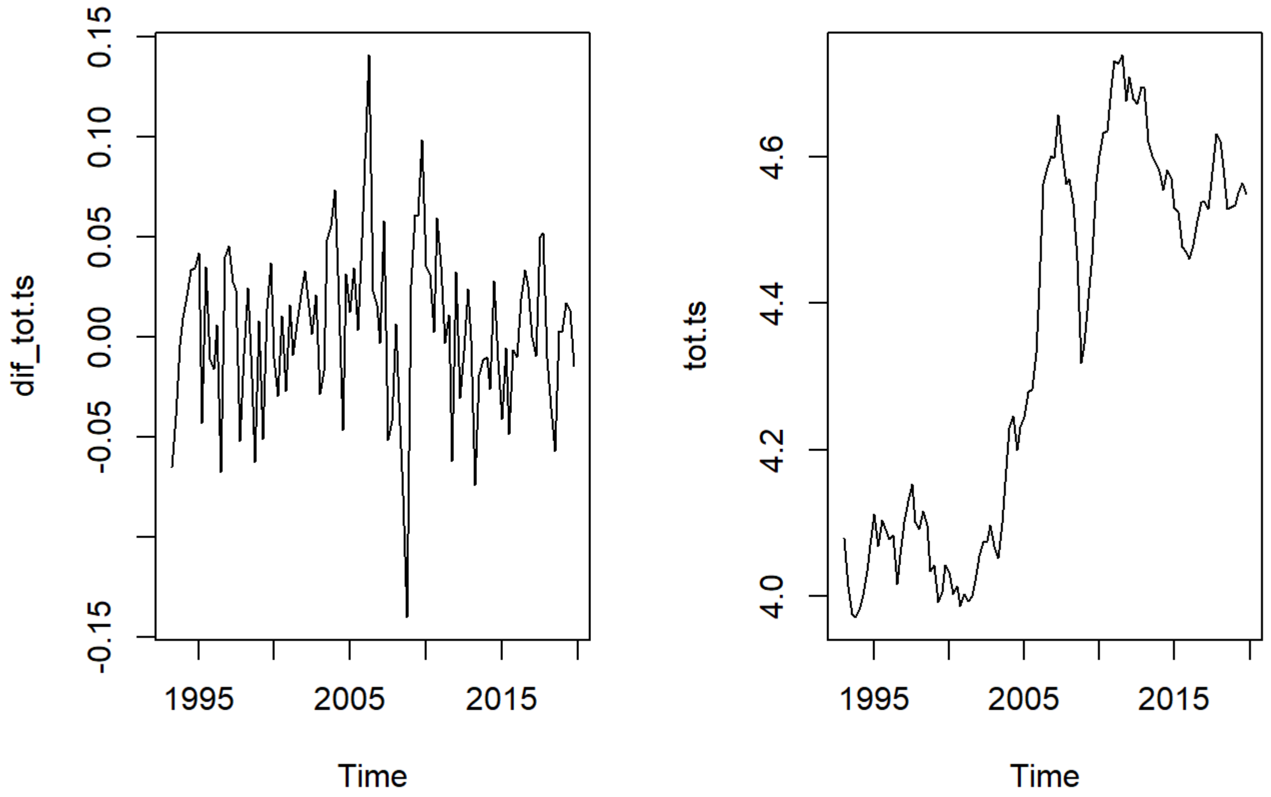
```
log_s_priv.ts<-log(s_priv.ts) # Conversión a logaritmos del ahorro privado
plot(log_s_priv.ts)
```



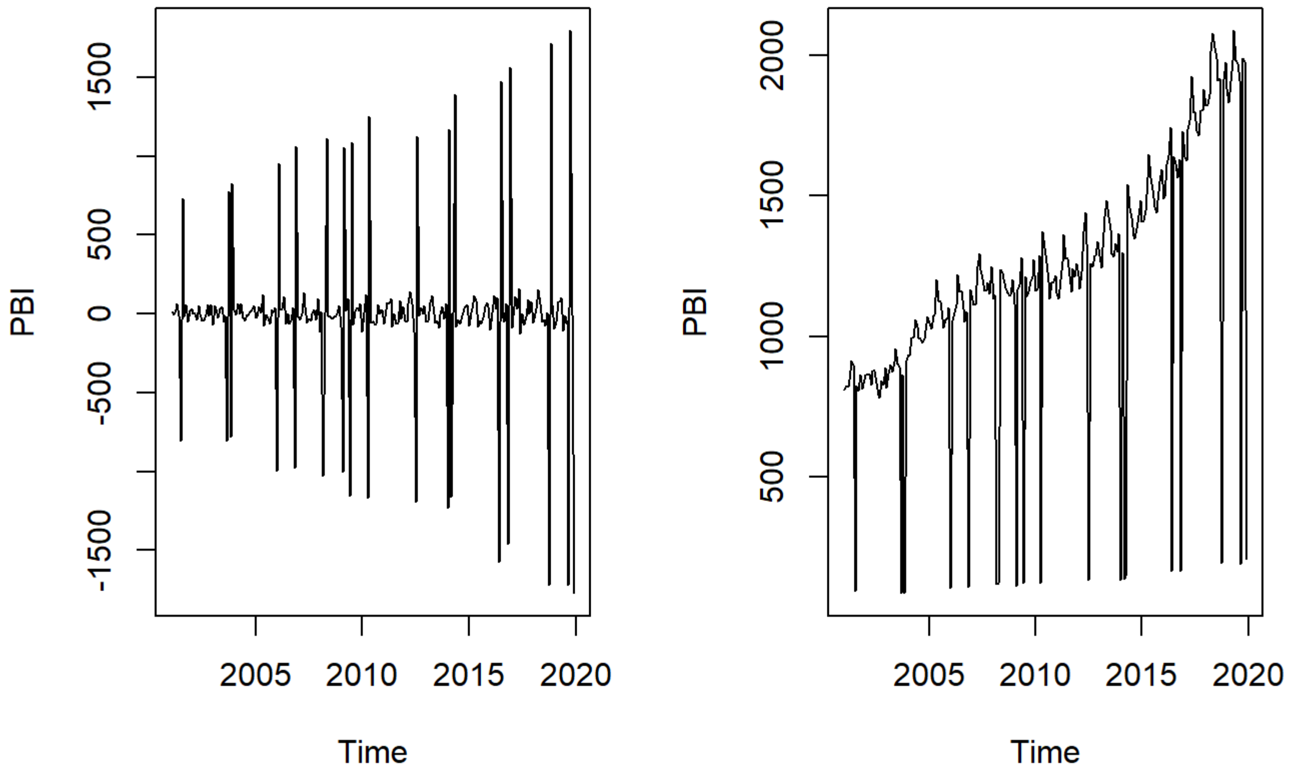
```
# Diferenciamos la serie: Eliminación de la tendencia
dif_pbi.ts=diff(pbi.ts)
plot(dif_pbi.ts)
```



```
dif_tot.ts=diff(tot.ts)
par(mfrow=c(1,2))
plot(dif_tot.ts)
plot(tot.ts)
```

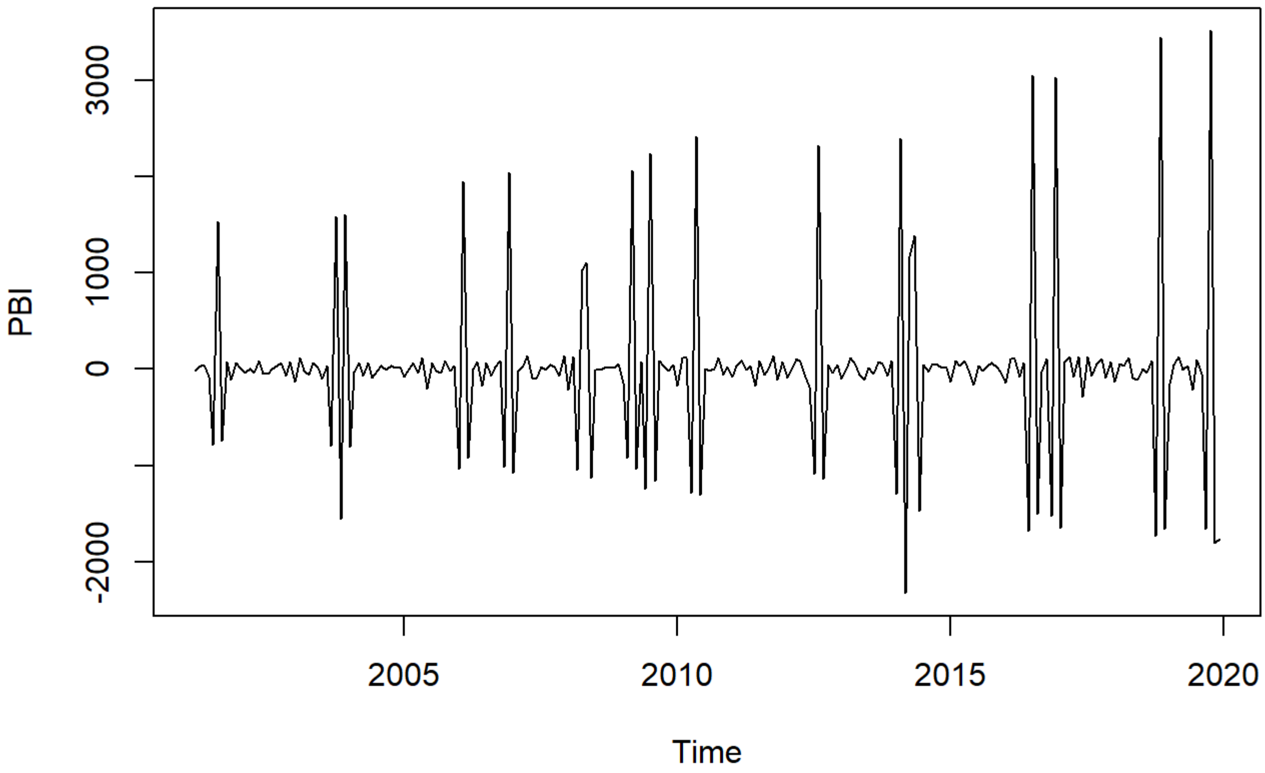


```
par(mfrow=c(1,2))
plot(dif_pbi.ts)
plot(pbi.ts)
```



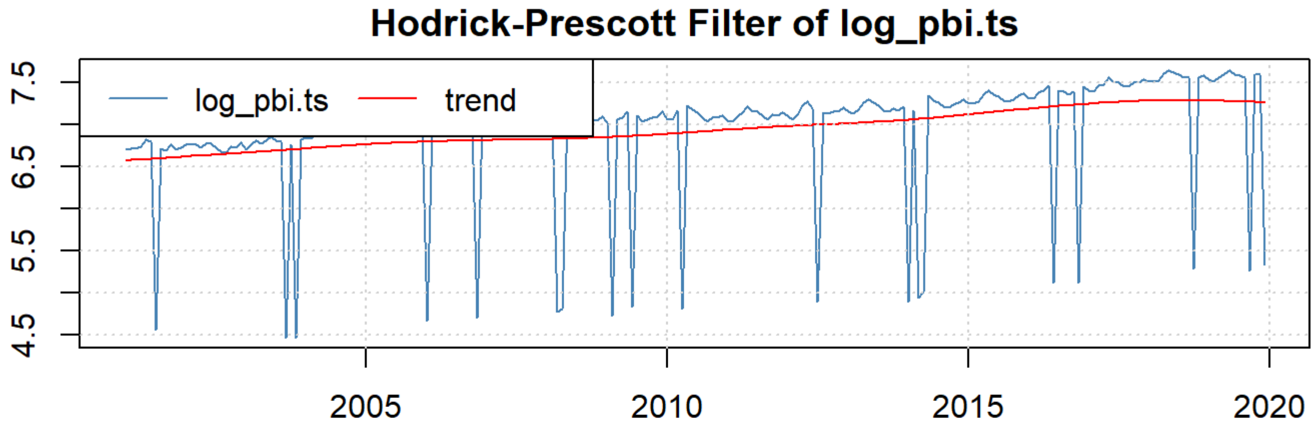
```
par(mfrow=c(1,1))
```

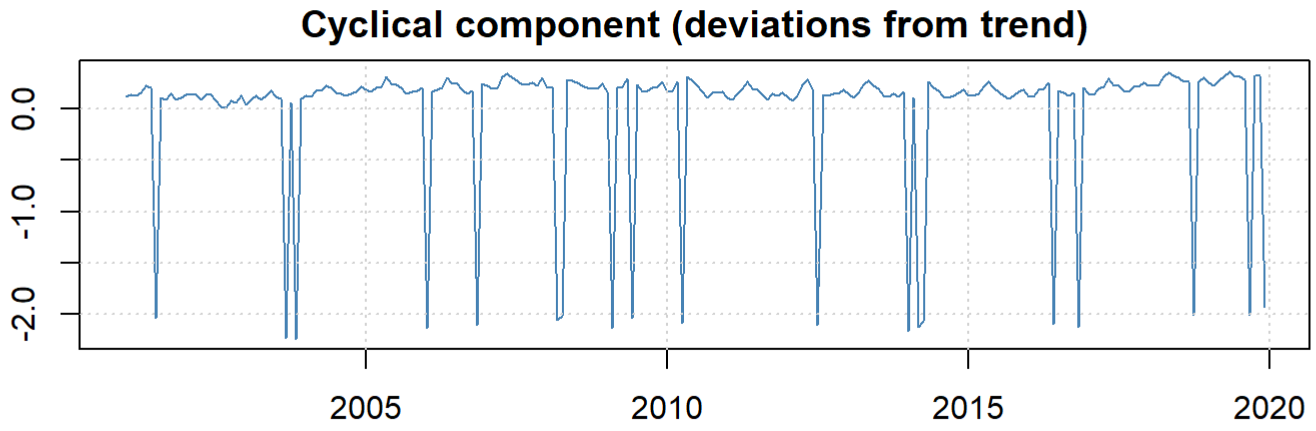
```
# Diferenciamos la serie: Eliminación de estacionalidad
dif_12_pbi.ts<-diff(dif_pbi.ts, lags=12)
plot(dif_12_pbi.ts)
```



```
# Filtro de Hodrick Prescott

library(mFilter)
```

```
lambda=14400
pbi.hp<-hpfilter(log_pbi.ts, type="lambda")
plot(pbi.hp)
```


Hodrick-Prescott Filter of log_pbi.ts
log_pbi.ts
trend
7.5
6.5
5.5
4.5
2005
2010
2015
2020


Cyclical component (deviations from trend)
0.0
-1.0
-2.0
2005
2010
2015
2020

```
print(pbi.hp)
```

```
##
## Title:
##  Hodrick-Prescott Filter
##
## Call:
##  hpfilter(x = log_pbi.ts, type = "lambda")
##
## Method:
##  hpfilter
##
## Filter Type:
##  lambda
##
## Series:
##  log_pbi.ts
##
##          log_pbi.ts Trend    Cycle
## Jan 2001      6.696 6.573  0.12293
## Feb 2001      6.714 6.577  0.13739
## Mar 2001      6.711 6.581  0.12983
## Apr 2001      6.741 6.585  0.15592
## May 2001      6.816 6.589  0.22711
## Jun 2001      6.800 6.592  0.20771
## Jul 2001      4.554 6.596 -2.04253
## Aug 2001      6.713 6.600  0.11258
## Sep 2001      6.692 6.604  0.08772
## Oct 2001      6.762 6.608  0.15322
## Nov 2001      6.701 6.612  0.08838
## Dec 2001      6.723 6.616  0.10628
## Jan 2002      6.763 6.620  0.14240
## Feb 2002      6.763 6.624  0.13843
## Mar 2002      6.766 6.628  0.13794
## Apr 2002      6.720 6.632  0.08803
## May 2002      6.778 6.636  0.14153
## Jun 2002      6.779 6.640  0.13877
## Jul 2002      6.727 6.644  0.08353
## Aug 2002      6.673 6.648  0.02552
## Sep 2002      6.663 6.652  0.01150
## Oct 2002      6.736 6.655  0.08030
## Nov 2002      6.720 6.659  0.06090
## Dec 2002      6.789 6.663  0.12581
## Jan 2003      6.704 6.667  0.03741
## Feb 2003      6.765 6.671  0.09419
## Mar 2003      6.804 6.675  0.12879
## Apr 2003      6.773 6.679  0.09449
## May 2003      6.812 6.682  0.12985
## Jun 2003      6.862 6.686  0.17528
## Jul 2003      6.807 6.690  0.11642
## Aug 2003      6.795 6.694  0.10014
## Sep 2003      4.466 6.699 -2.23263
## Oct 2003      6.763 6.703  0.06002
## Nov 2003      4.466 6.707 -2.24104
## Dec 2003      6.817 6.711  0.10548
## Jan 2004      6.842 6.716  0.12601
## Feb 2004      6.837 6.720  0.11736
## Mar 2004      6.905 6.724  0.18041
## Apr 2004      6.905 6.729  0.17606
## May 2004      6.965 6.733  0.23208
## Jun 2004      6.943 6.737  0.20586
## Jul 2004      6.903 6.741  0.16128
## Aug 2004      6.899 6.746  0.15315
## Sep 2004      6.883 6.750  0.13388
## Oct 2004      6.900 6.753  0.14623
## Nov 2004      6.927 6.757  0.16928
## Dec 2004      6.976 6.761  0.21538
## Jan 2005      6.955 6.765  0.19013
## Feb 2005      6.935 6.768  0.16745
## Mar 2005      6.979 6.771  0.20796
## Apr 2005      6.984 6.774  0.20949
## May 2005      7.093 6.777  0.31530
## Jun 2005      7.028 6.780  0.24809
## Jul 2005      7.023 6.783  0.24007
## Aug 2005      6.999 6.785  0.21317
## Sep 2005      6.939 6.788  0.15149
## Oct 2005      6.964 6.790  0.17408
## Nov 2005      6.971 6.792  0.17850
## Dec 2005      7.005 6.794  0.21058
## Jan 2006      4.663 6.796 -2.13284
## Feb 2006      6.963 6.798  0.16500
## Mar 2006      6.987 6.800  0.18748
## Apr 2006      7.015 6.802  0.21307
## May 2006      7.107 6.803  0.30403
## Jun 2006      7.055 6.805  0.25035
## Jul 2006      7.058 6.806  0.25144
## Aug 2006      7.000 6.808  0.19246
## Sep 2006      6.958 6.809  0.14922
## Oct 2006      6.993 6.811  0.18248
## Nov 2006      4.700 6.812 -2.11130
## Dec 2006      7.063 6.813  0.25004
## Jan 2007      7.047 6.814  0.23246
## Feb 2007      7.016 6.815  0.20039
## Mar 2007      7.015 6.816  0.19839
## Apr 2007      7.128 6.817  0.31102
## May 2007      7.165 6.818  0.34700
## Jun 2007      7.121 6.819  0.30178
## Jul 2007      7.095 6.820  0.27464
## Aug 2007      7.060 6.821  0.23825
## Sep 2007      7.059 6.822  0.23646
## Oct 2007      7.083 6.823  0.25932
## Nov 2007      7.048 6.824  0.22419
## Dec 2007      7.128 6.825  0.30330
## Jan 2008      7.033 6.826  0.20637
## Feb 2008      7.046 6.827  0.21838
## Mar 2008      4.771 6.829 -2.05796
## Apr 2008      4.812 6.830 -2.01783
## May 2008      7.120 6.832  0.28891
## Jun 2008      7.112 6.833  0.27834
## Jul 2008      7.094 6.835  0.25929
## Aug 2008      7.068 6.837  0.23134
## Sep 2008      7.051 6.839  0.21214
## Oct 2008      7.043 6.841  0.20217
## Nov 2008      7.046 6.843  0.20251
## Dec 2008      7.092 6.846  0.24607
## Jan 2009      7.018 6.848  0.17019
## Feb 2009      4.718 6.851 -2.13240
## Mar 2009      7.063 6.854  0.20930
## Apr 2009      7.077 6.857  0.22076
## May 2009      7.154 6.860  0.29395
## Jun 2009      4.828 6.863 -2.03485
## Jul 2009      7.102 6.867  0.23509
## Aug 2009      7.041 6.870  0.17041
## Sep 2009      7.054 6.874  0.18068
## Oct 2009      7.086 6.878  0.20839
## Nov 2009      7.100 6.881  0.21869
## Dec 2009      7.150 6.885  0.26469
## Jan 2010      7.059 6.889  0.16958
## Feb 2010      7.064 6.893  0.17073
## Mar 2010      7.162 6.897  0.26440
## Apr 2010      4.812 6.901 -2.08913
## May 2010      7.224 6.905  0.31858
## Jun 2010      7.183 6.910  0.27352
## Jul 2010      7.142 6.914  0.22829
## Aug 2010      7.087 6.918  0.16885
## Sep 2010      7.033 6.922  0.11062
## Oct 2010      7.084 6.926  0.15813
## Nov 2010      7.085 6.930  0.15493
## Dec 2010      7.104 6.934  0.17001
## Jan 2011      7.053 6.938  0.11466
## Feb 2011      7.034 6.942  0.09247
## Mar 2011      7.093 6.946  0.14686
## Apr 2011      7.144 6.949  0.19498
## May 2011      7.216 6.953  0.26292
## Jun 2011      7.152 6.957  0.19567
## Jul 2011      7.153 6.960  0.19299
## Aug 2011      7.105 6.963  0.14152
## Sep 2011      7.055 6.967  0.08857
## Oct 2011      7.124 6.970  0.15452
## Nov 2011      7.102 6.973  0.12856
## Dec 2011      7.137 6.976  0.16108
## Jan 2012      7.104 6.979  0.12492
## Feb 2012      7.064 6.982  0.08171
## Mar 2012      7.116 6.985  0.13126
## Apr 2012      7.225 6.988  0.23745
## May 2012      7.273 6.991  0.28219
## Jun 2012      7.192 6.994  0.19841
## Jul 2012      4.890 6.997 -2.10629
## Aug 2012      7.138 7.000  0.13854
## Sep 2012      7.129 7.002  0.12685
## Oct 2012      7.159 7.005  0.15392
## Nov 2012      7.160 7.008  0.15174
## Dec 2012      7.199 7.011  0.18762
## Jan 2013      7.160 7.014  0.14574
## Feb 2013      7.126 7.017  0.10871
## Mar 2013      7.182 7.020  0.16112
## Apr 2013      7.264 7.024  0.24042
## May 2013      7.301 7.027  0.27433
## Jun 2013      7.263 7.030  0.23254
## Jul 2013      7.231 7.033  0.19712
## Aug 2013      7.165 7.037  0.12782
## Sep 2013      7.159 7.040  0.11805
## Oct 2013      7.192 7.044  0.14802
## Nov 2013      7.171 7.048  0.12287
## Dec 2013      7.219 7.052  0.16688
## Jan 2014      4.890 7.056 -2.16589
## Feb 2014      7.169 7.061  0.10870
## Mar 2014      4.934 7.065 -2.13081
## Apr 2014      5.004 7.070 -2.06619
## May 2014      7.340 7.075  0.26499
## Jun 2014      7.286 7.080  0.20506
## Jul 2014      7.261 7.086  0.17538
## Aug 2014      7.215 7.091  0.12312
## Sep 2014      7.207 7.097  0.11009
## Oct 2014      7.232 7.103  0.12923
## Nov 2014      7.265 7.109  0.15614
## Dec 2014      7.302 7.114  0.18804
## Jan 2015      7.249 7.120  0.12885
## Feb 2015      7.253 7.126  0.12648
## Mar 2015      7.277 7.132  0.14503
## Apr 2015      7.355 7.138  0.21687
## May 2015      7.407 7.144  0.26330
## Jun 2015      7.356 7.150  0.20640
## Jul 2015      7.328 7.156  0.17275
## Aug 2015      7.287 7.161  0.12543
## Sep 2015      7.272 7.167  0.10456
## Oct 2015      7.301 7.173  0.12837
## Nov 2015      7.347 7.178  0.16831
## Dec 2015      7.373 7.184  0.18954
## Jan 2016      7.308 7.189  0.11863
## Feb 2016      7.311 7.195  0.11664
## Mar 2016      7.385 7.200  0.18478
## Apr 2016      7.404 7.205  0.19927
## May 2016      7.463 7.210  0.25325
## Jun 2016      5.118 7.215 -2.09715
## Jul 2016      7.403 7.220  0.18295
## Aug 2016      7.388 7.225  0.16271
## Sep 2016      7.356 7.230  0.12647
## Oct 2016      7.395 7.235  0.16059
## Nov 2016      5.112 7.239 -2.12714
## Dec 2016      7.455 7.244  0.21109
## Jan 2017      7.402 7.248  0.15384
## Feb 2017      7.393 7.252  0.14042
## Mar 2017      7.459 7.256  0.20248
## Apr 2017      7.477 7.260  0.21688
## May 2017      7.562 7.264  0.29833
## Jun 2017      7.494 7.267  0.22715
## Jul 2017      7.495 7.270  0.22449
## Aug 2017      7.457 7.273  0.18355
## Sep 2017      7.448 7.276  0.17154
## Oct 2017      7.498 7.279  0.21908
## Nov 2017      7.498 7.281  0.21743
## Dec 2017      7.538 7.283  0.25568
## Jan 2018      7.508 7.284  0.22376
## Feb 2018      7.510 7.286  0.22401
## Mar 2018      7.528 7.287  0.24081
## Apr 2018      7.608 7.288  0.32054
## May 2018      7.639 7.288  0.35027
## Jun 2018      7.615 7.289  0.32656
## Jul 2018      7.595 7.289  0.30658
## Aug 2018      7.556 7.289  0.26727
## Sep 2018      7.560 7.288  0.27132
## Oct 2018      5.278 7.288 -2.00952
## Nov 2018      7.556 7.287  0.26958
## Dec 2018      7.587 7.286  0.30144
## Jan 2019      7.540 7.285  0.25485
## Feb 2019      7.513 7.283  0.22978
## Mar 2019      7.554 7.282  0.27193
## Apr 2019      7.592 7.280  0.31163
## May 2019      7.643 7.278  0.36505
## Jun 2019      7.591 7.277  0.31484
## Jul 2019      7.586 7.274  0.31180
## Aug 2019      7.554 7.272  0.28195
## Sep 2019      5.257 7.270 -2.01272
## Oct 2019      7.594 7.268  0.32638
## Nov 2019      7.590 7.266  0.32458
## Dec 2019      5.333 7.264 -1.93080
```

```
# Realizamos el modelo MCO
modelo1<-lm(i_priv~i_pub+ipx+ipm, data = base01 )
summary(modelo1)
```

```
##
## Call:
## lm(formula = i_priv ~ i_pub + ipx + ipm, data = base01)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2207 -1.2647 -0.1623  1.3261  8.6455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.0785    13.5947  -0.521   0.6037
## i_pub         0.4653     0.2752   1.691   0.0938 .
## ipx           1.9638     3.0148   0.651   0.5162
## ipm           2.8683     5.8733   0.488   0.6263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.37 on 104 degrees of freedom
## Multiple R-squared:  0.4312, Adjusted R-squared:  0.4148
## F-statistic: 26.28 on 3 and 104 DF,  p-value: 9.84e-13
```

```
# Outliers

library(tsoutliers)
```

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

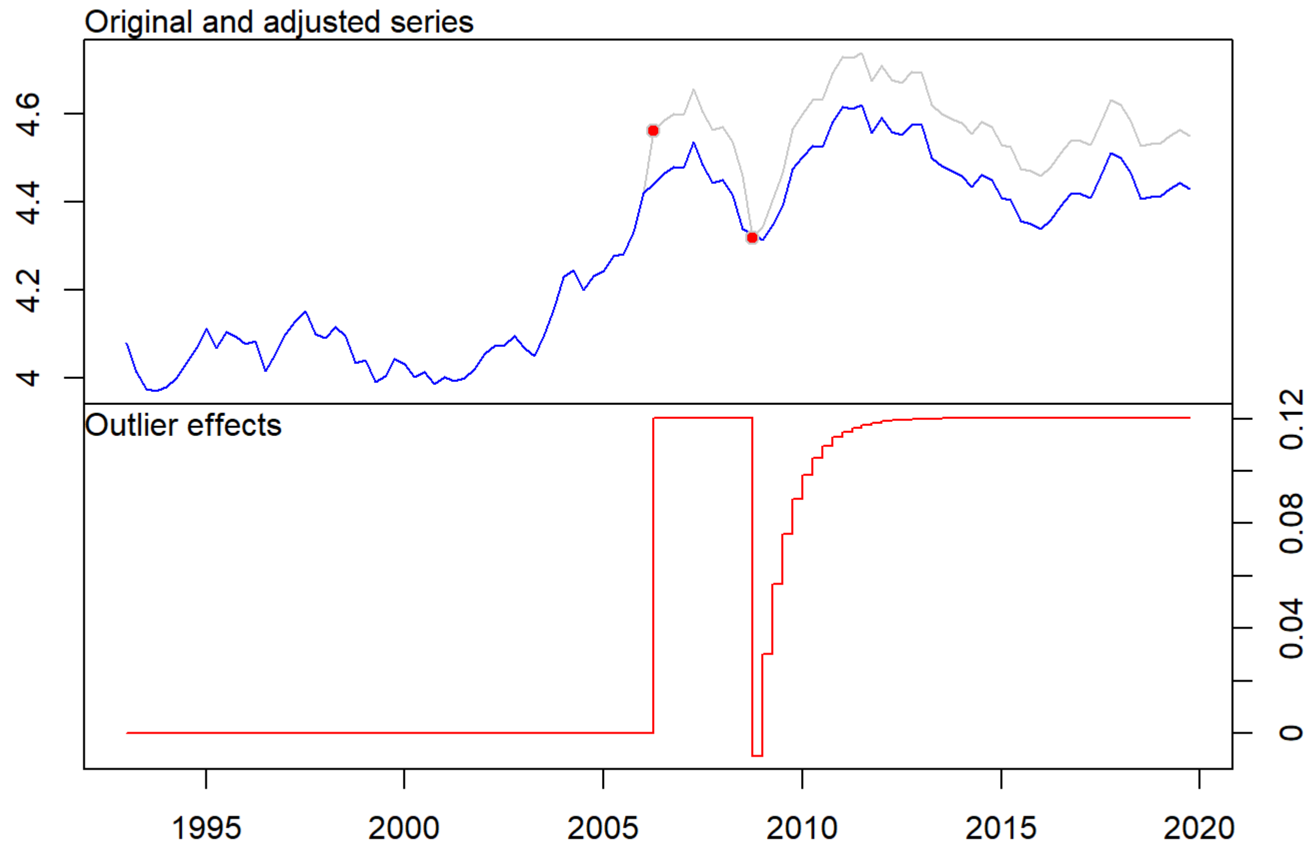
```
out_tot<-tso(tot.ts, types=c('AO','LS','TC'), maxit.iloop=15)
out_tot
```

```
## Series: tot.ts 
## Regression with ARIMA(2,0,0) errors 
## 
## Coefficients:
##          ar1      ar2  intercept    LS54     TC64
##       1.2283  -0.2485     4.2741  0.1202  -0.1289
## s.e.  0.1039   0.1041     0.1261  0.0350   0.0312
## 
## sigma^2 = 0.001195:  log likelihood = 210.94
## AIC=-409.88   AICc=-409.05   BIC=-393.79
## 
## Outliers:
##   type ind    time coefhat  tstat
## 1   LS  54 2006:02  0.1202  3.438
## 2   TC  64 2008:04 -0.1289 -4.130
```

```
out_tot$time
```

```
## [1] 2006.25 2008.75
```

```
plot(out_tot)
```



```
out_pbi<-tso(log_pbi.ts, types=c('AO','LS','TC'), maxit.iloop=15)
```

```
## Warning in locate.outliers.oloop(y = y, fit = fit, types = types, cval = cval,
## : stopped when 'maxit.oloop = 4' was reached
```

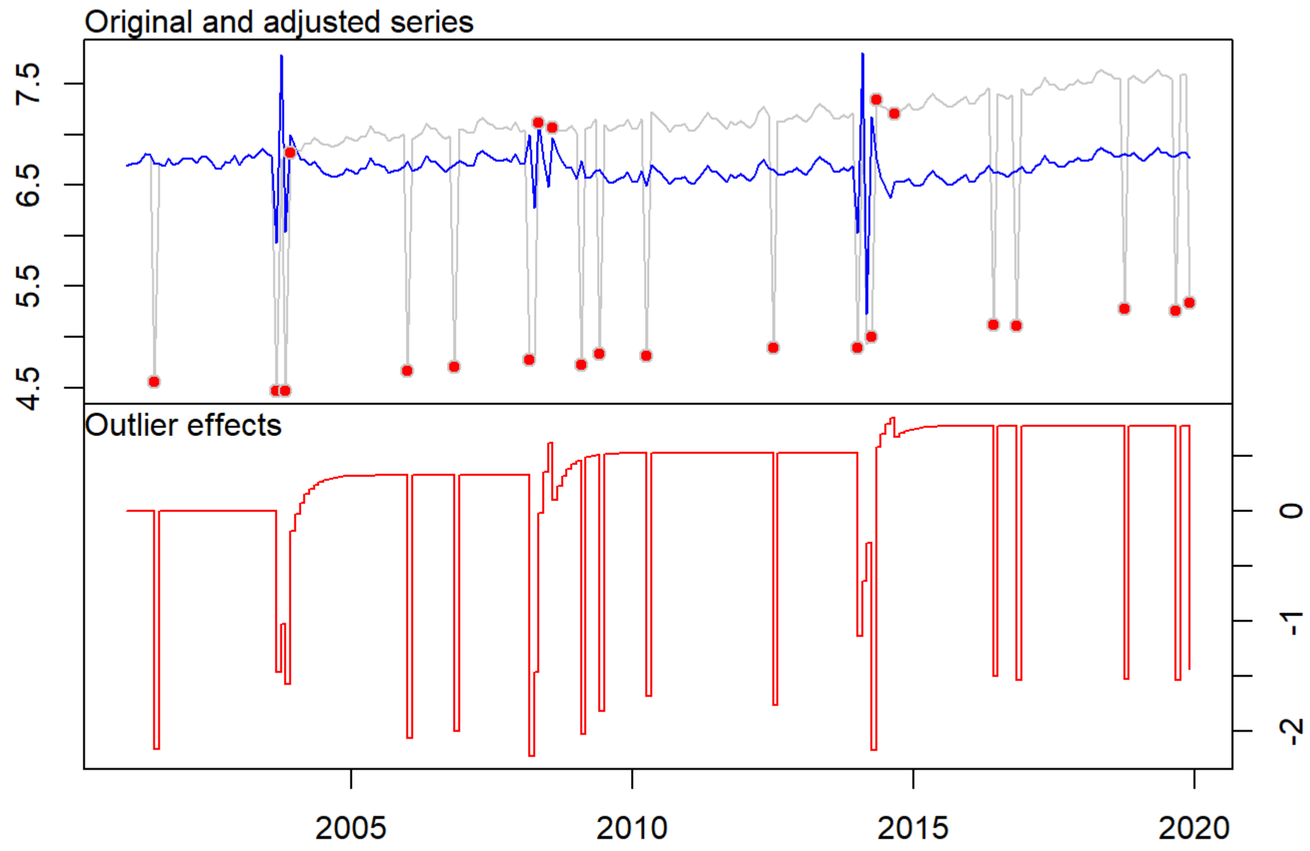
```
out_pbi
```

```
## Series: log_pbi.ts
## Regression with ARIMA(3,0,1)(0,0,2)[12] errors
##
## Coefficients:
##            ar1     ar2     ar3      ma1    sma1    sma2  intercept      AO7
##        -0.0404  0.5427  0.3909  -0.6402  0.2167  0.1496     6.6938  -2.1587
## s.e.    0.0880  0.0526  0.0696   0.0827  0.0693  0.0678     0.0620   0.1037
##           TC33     AO35    LS36     AO61     AO71     TC87    LS89     LS92
##        -1.4667  -0.8540  0.3214  -2.3864  -2.3252  -2.5458  0.9016  -0.6994
## s.e.    0.0797   0.1185  0.0506   0.1026   0.1016   0.0735  0.0559   0.0612
##           AO98    AO102    AO112    AO139    TC157    AO160   LS161    LS165
##        -2.4996  -2.3310  -2.2014  -2.2874  -1.6604  -2.1271  0.4555  -0.2091
## s.e.    0.1044   0.1075   0.1016   0.1015   0.0767   0.1071  0.0600   0.0511
##          AO186    AO191    AO214    AO225    AO228
##        -2.2719  -2.3055  -2.2990  -2.3028  -2.2087
## s.e.    0.1027   0.1012   0.1018   0.1177   0.1523
##
## sigma^2 = 0.02235:  log likelihood = 123.82
## AIC=-187.64   AICc=-178.2   BIC=-84.76
##
## Outliers:
##    type ind    time coefhat   tstat
## 1    AO   7 2001:07 -2.1587 -20.808
## 2    TC  33 2003:09 -1.4667 -18.412
## 3    AO  35 2003:11 -0.8540  -7.208
## 4    LS  36 2003:12  0.3214   6.350
## 5    AO  61 2006:01 -2.3864 -23.265
## 6    AO  71 2006:11 -2.3252 -22.882
## 7    TC  87 2008:03 -2.5458 -34.642
## 8    LS  89 2008:05  0.9016  16.118
## 9    LS  92 2008:08 -0.6994 -11.434
## 10   AO  98 2009:02 -2.4996 -23.946
## 11   AO 102 2009:06 -2.3310 -21.685
## 12   AO 112 2010:04 -2.2014 -21.673
## 13   AO 139 2012:07 -2.2874 -22.542
## 14   TC 157 2014:01 -1.6604 -21.647
## 15   AO 160 2014:04 -2.1271 -19.857
## 16   LS 161 2014:05  0.4555   7.590
## 17   LS 165 2014:09 -0.2091  -4.090
## 18   AO 186 2016:06 -2.2719 -22.116
## 19   AO 191 2016:11 -2.3055 -22.777
## 20   AO 214 2018:10 -2.2990 -22.588
## 21   AO 225 2019:09 -2.3028 -19.569
## 22   AO 228 2019:12 -2.2087 -14.498
```

```
out_pbi$time
```

```
##  [1] 2001.500 2003.667 2003.833 2003.917 2006.000 2006.833 2008.167 2008.333
##  [9] 2008.583 2009.083 2009.417 2010.250 2012.500 2014.000 2014.250 2014.333
## [17] 2014.667 2016.417 2016.833 2018.750 2019.667 2019.917
```

```
plot(out_pbi)
```


Original and adjusted series
7.5
6.5
5.5
4.5
Outlier effects
0
-1
-2
2005
2010
2015
2020