Modelos para datos temporales y espaciales. Trabajo temas 1 a 4.

Jerónimo Carranza Carranza

16 de agosto de 2017

Contents

1 Introducción

1.1 Alumno

Jerónimo Carranza Carranza

1.2 Serie temporal

Precipitación total mensual en la Estación Meteorológica de Sevilla - Tablada (Código: 5790) en el periodo 1857 - 2004

Los datos se han descargado de la página *Descargas REDIAM* (http://descargasrediam.cica.es/repo/s/RUR?path=%2F) y corresponde concretamente a las series mensuales de la AEMET que se ubican en la siguiente posición en el árbol de directorios del repositorio:

http://descargasrediam.cica.es/repo/s/RUR?path=%2F04_RECURSOS_NATURALES%2F03_CLIMA%2F01_REDES_DE_OBSERVACION%2F02_DATOS%2F01_AEMET%2FDATOS_MENSUALES_AEMET

Se utiliza el fichero DATOS.DBF como origen de datos. Se filtra (CODIGO=5790) y se seleccionan las variables AGNO, MES y PREFINAL.

2 Lectura de datos

```r
#library('foreign')
#datos = read.dbf('DATOS.DBF')
#datos = datos[datos$CODIGO == '5790',2:4]
#save(datos,file = 'datos.RData')

load('datos.RData')
dim(datos)
```

```
## [1] 1602    3
```

3 Resumen de datos

```r
str(datos)
```

```
## 'data.frame':    1602 obs. of  3 variables:
##  $ AGNO    : num  1871 1871 1871 1871 1871 ...
##  $ MES     : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ PREFINAL: num  26.2 35.8 59.9 9.9 40 10.6 0 0 14.3 25 ...
```

```r
summary(datos)
```

```
##       AGNO           MES            PREFINAL
##  Min.   :1871   Min.   : 1.000   Min.   :  0.00
##  1st Qu.:1904   1st Qu.: 3.000   1st Qu.:  3.70
##  Median :1937   Median : 6.000   Median : 28.00
##  Mean   :1937   Mean   : 6.489   Mean   : 47.44
##  3rd Qu.:1971   3rd Qu.: 9.000   3rd Qu.: 68.95
```

```
##  Max.   :2004   Max.   :12.000   Max.   :395.20
##                                  NA's   :8
```

```r
datos[is.na(datos$PREFINAL)==TRUE,]
```

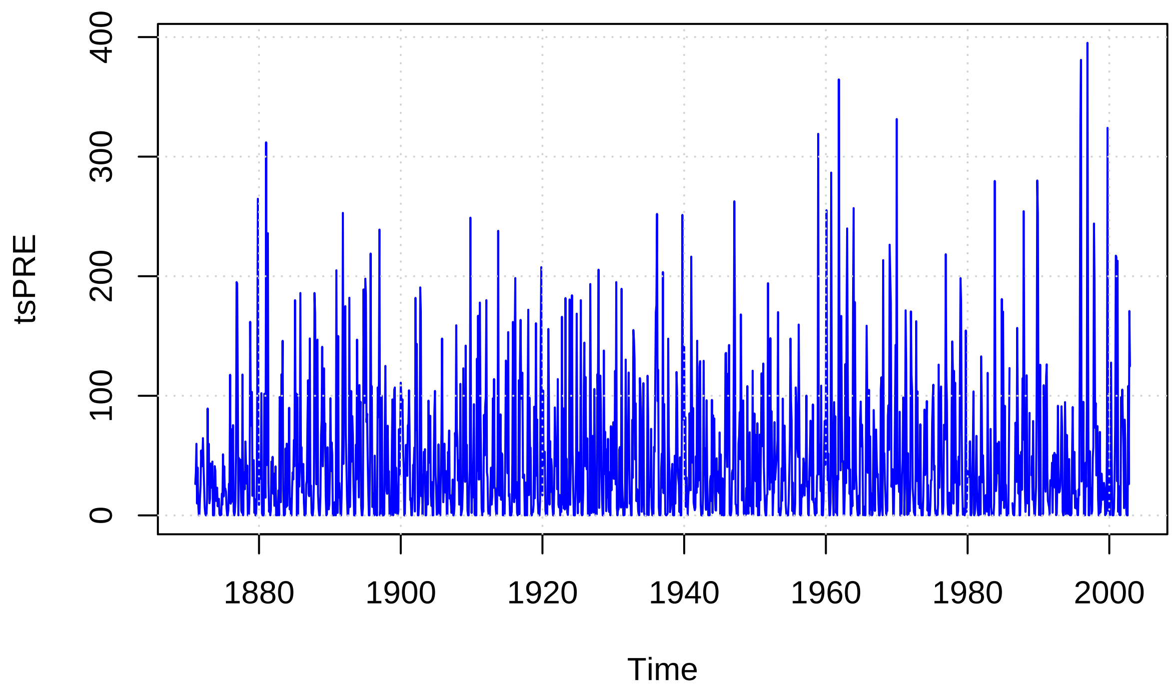
```
##        AGNO MES PREFINAL
## 426905 1986   1       NA
## 426906 1986   2       NA
## 426907 1986   3       NA
## 426908 1986   4       NA
## 426936 1988   8       NA
## 426939 1988  11       NA
## 426940 1988  12       NA
## 426974 1991  10       NA
```

4 Creación de serie temporal

```r
# Se incluyen años completos hasta 2002, no incluye 2003 destinado a contraste de predicción
tsPRE = ts(datos$PREFINAL, frequency = 12, start = c(1871,1), end = c(2002,12))
```

5 Representación gráfica

```r
plot.ts(tsPRE,col=4)
grid()
```

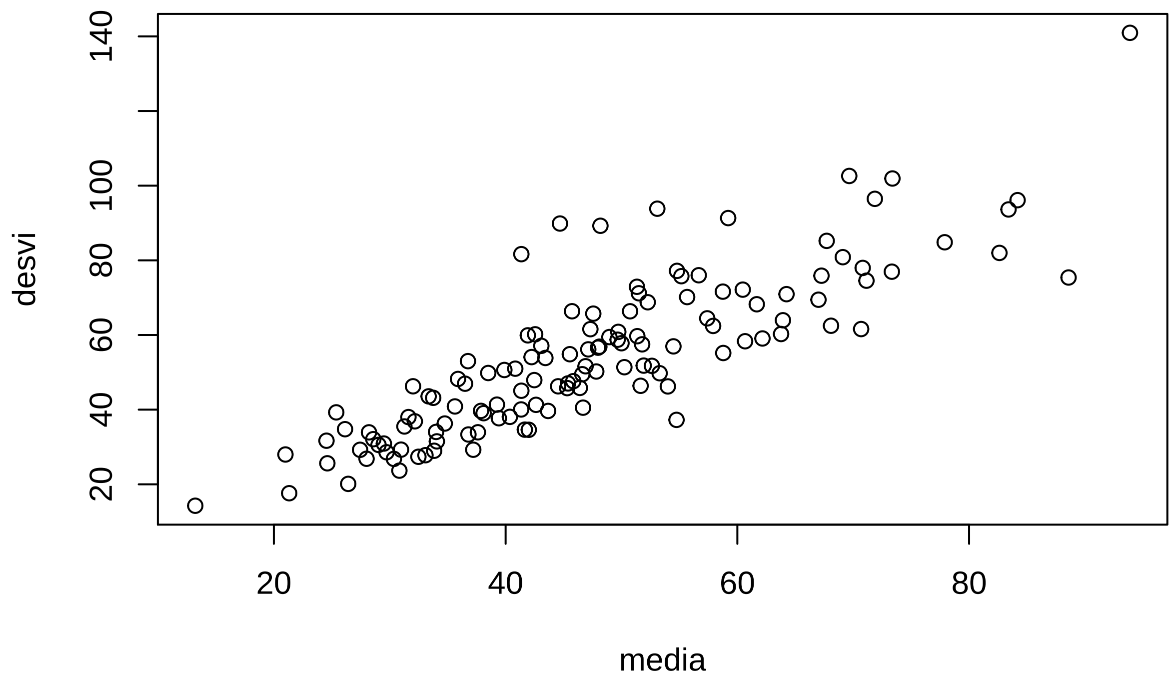


6 Imputación de datos faltantes

```
# Librería específica para este fin
library('imputeTS')

# Imputación por interpolación lineal previa descomposición estacional
tsPREi = na.seasplit(tsPRE, algorithm = "interpolation")
```

7 Homogeneidad de varianza

```
media = c(rep(0,132))
desvi = c(rep(0,132))
Anual = matrix(tsPRE, nr=12, byrow=F)
for (i in 1:132){
  media[i] = mean(Anual[,i])
  desvi[i] = sd(Anual[,i])
}
plot(media,desvi)
```



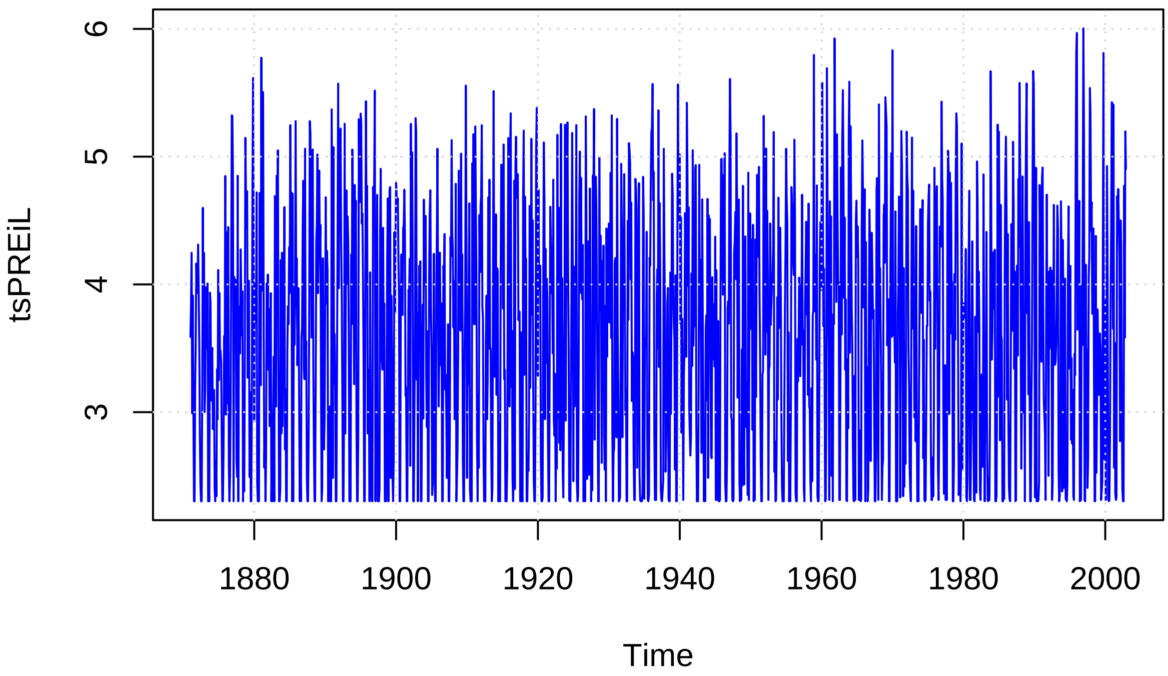
8 Transformación (logarítmica)

```
lmedia = log(media)
ldesvi = log(desvi)
(regre = lm(ldesvi~lmedia))
```

```
##
## Call:
## lm(formula = ldesvi ~ lmedia)
##
## Coefficients:
## (Intercept)       lmedia
##    -0.08049      1.05024
```

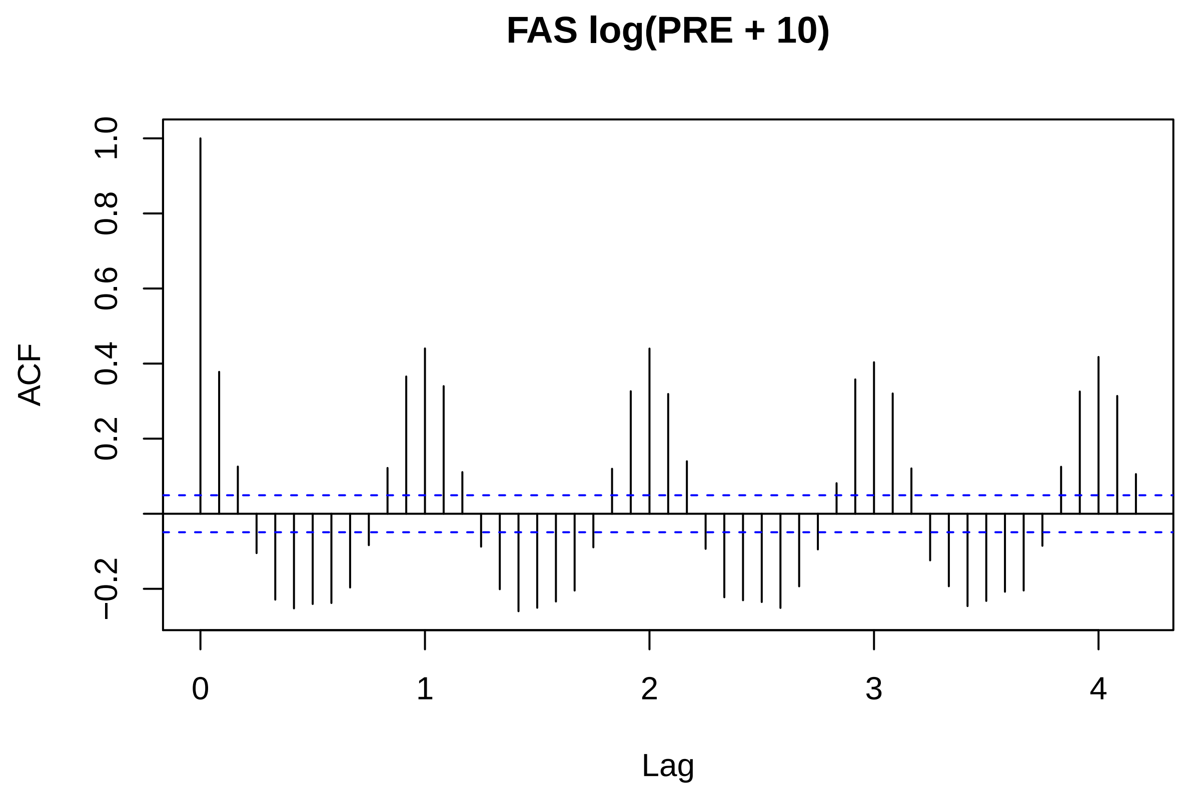
lambda = 1 - alfa = 1 - (1.05024) = -0.05 ~ 0 -> transformación logarítmica

```
# Dado que existen observaciones cero se añade 10mm a todas las observaciones en la transformación
tsPREiL = log(tsPREi + 10)
plot.ts(tsPREiL, col=4)
grid()
```

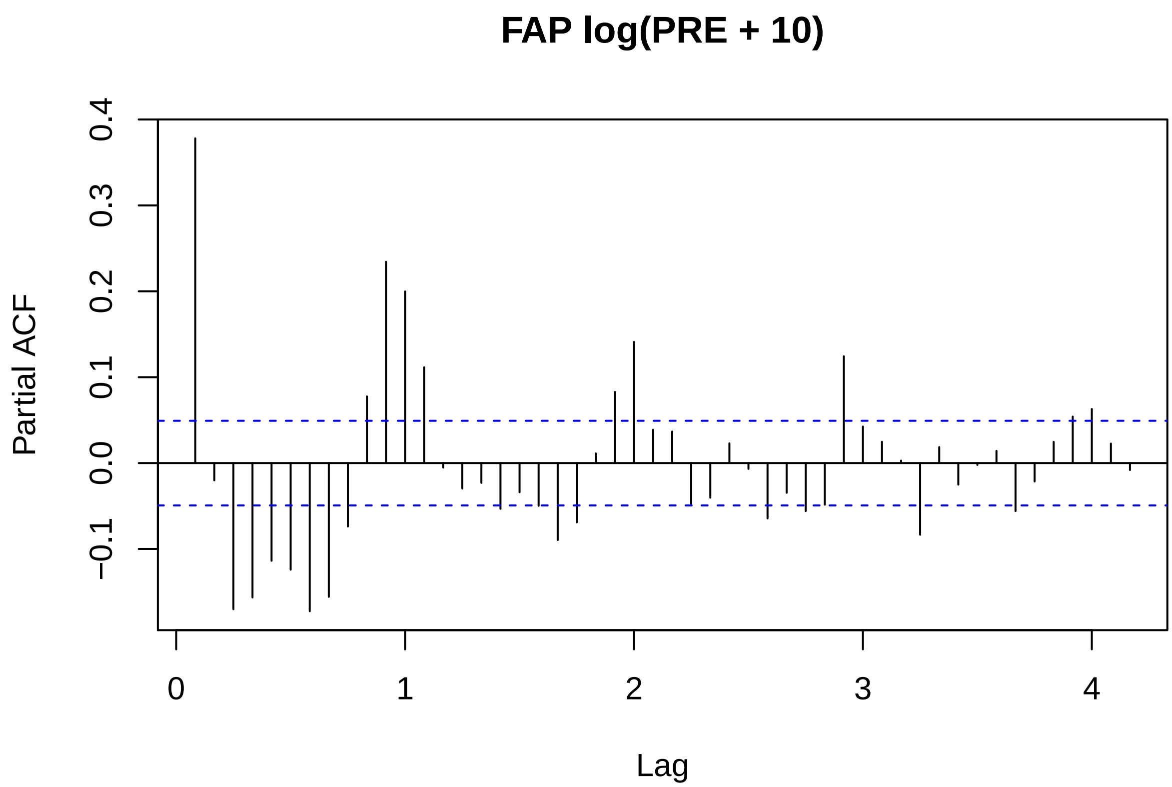


9 Estacionariedad de media

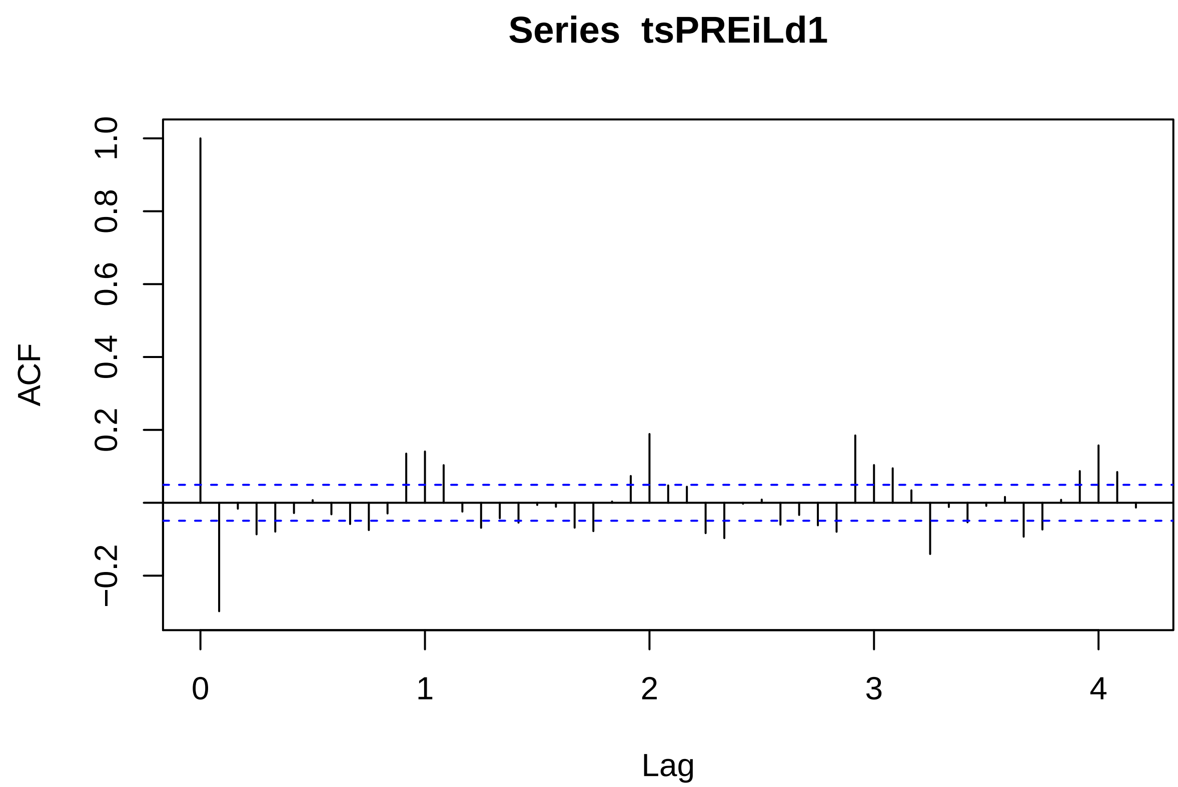
```r
acf(tsPREiL, main="FAS log(PRE + 10)", lag.max = 50)
```



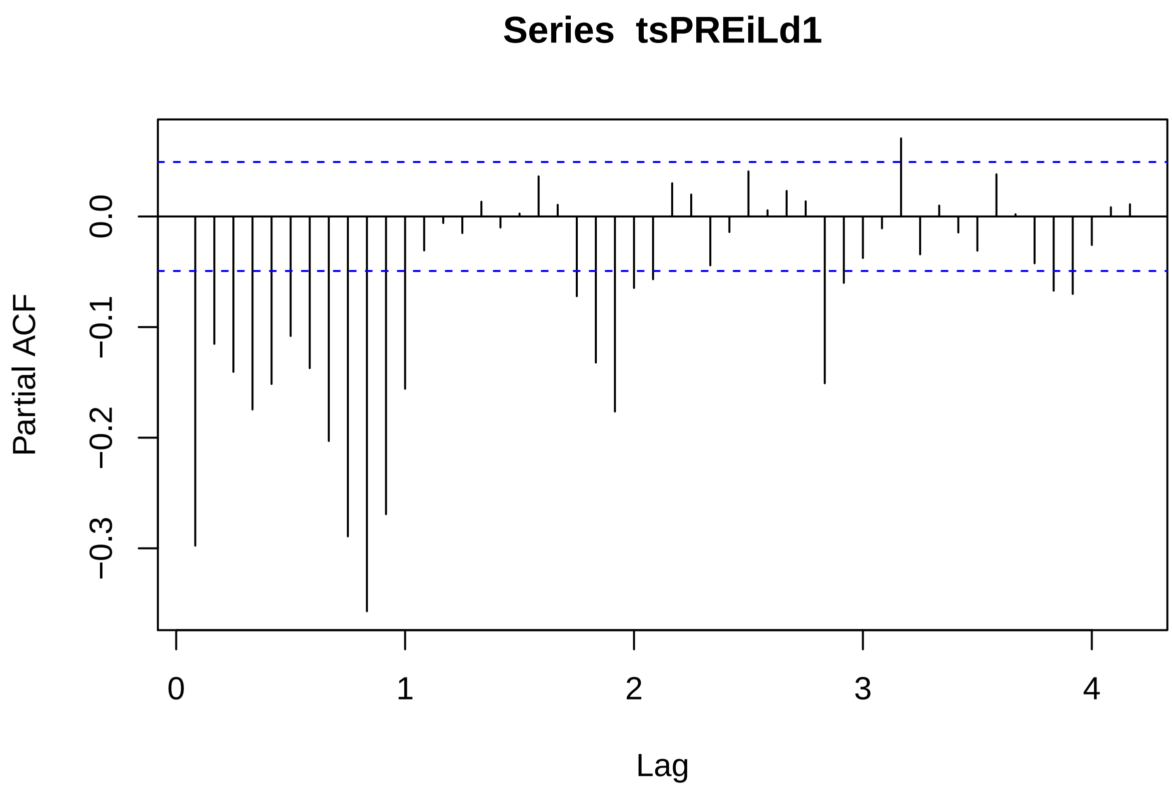

```
pacf(tsPREiL, main="FAP log(PRE + 10)", lag.max = 50)
```



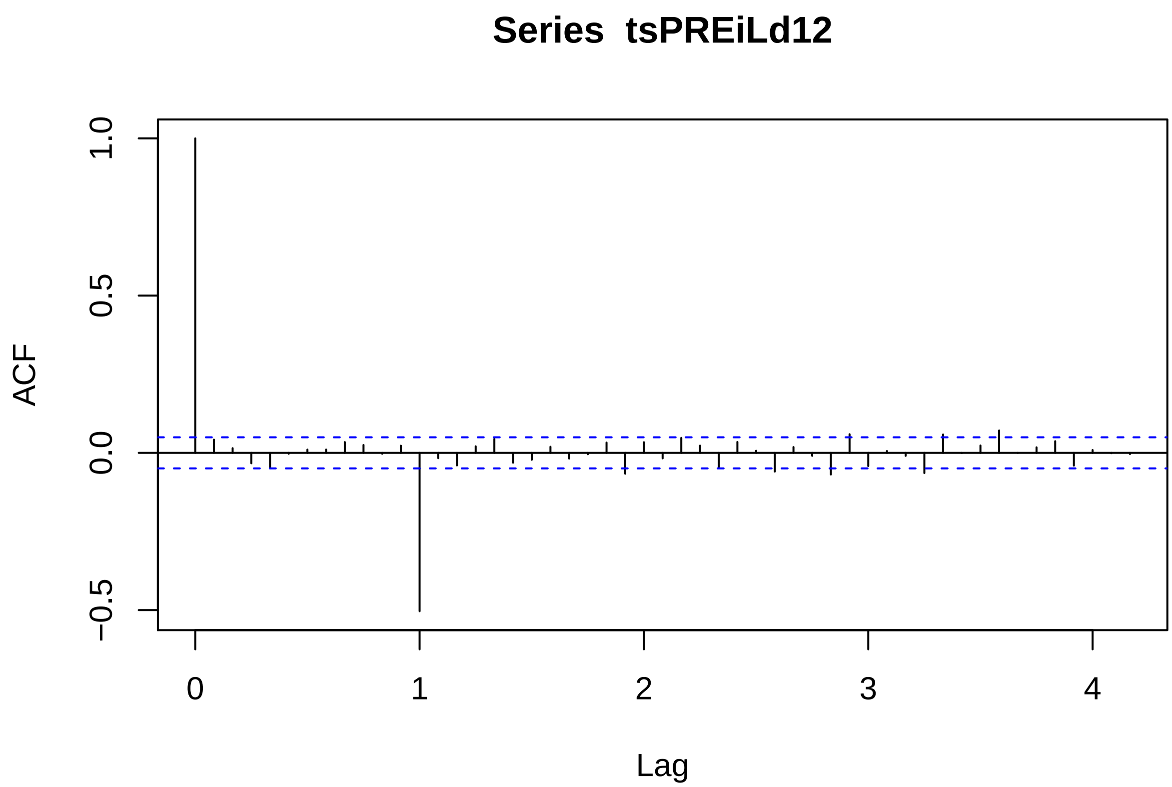

```r
tsPREiLd1 = diff(tsPREiL, lag = 1, differences = 1)
acf(tsPREiLd1, lag.max = 50)
```



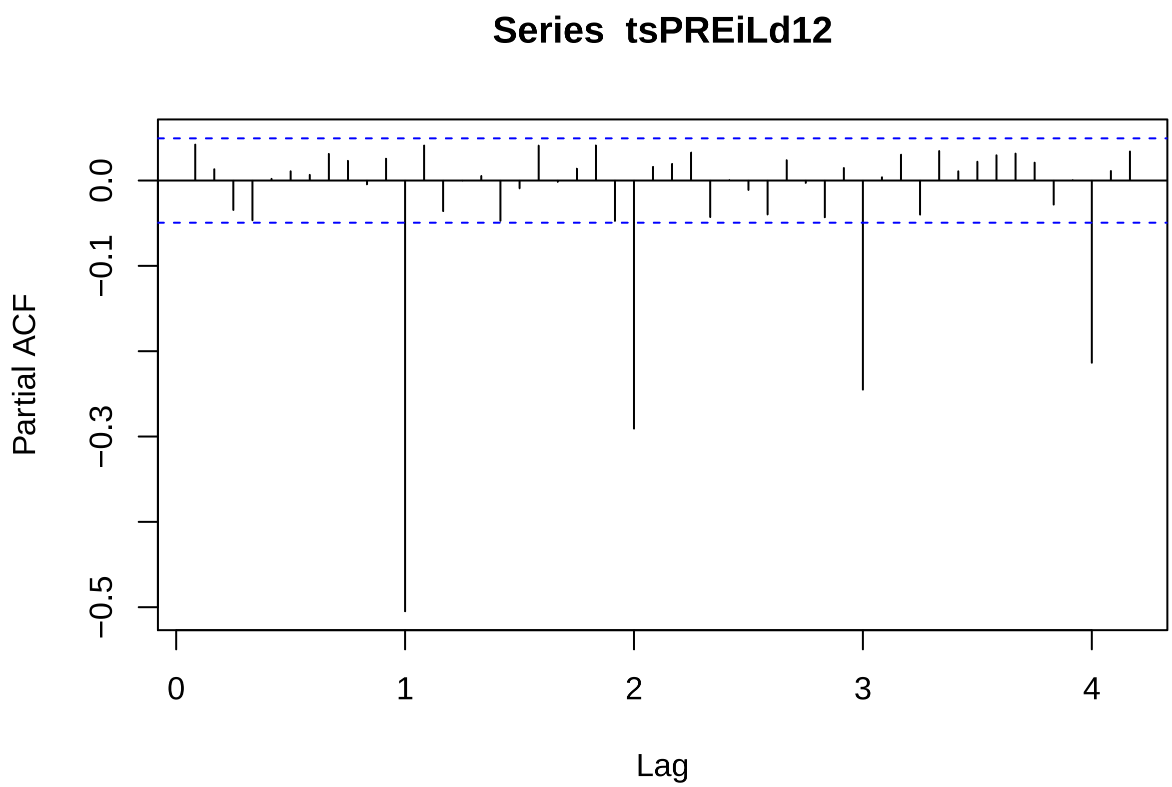

```
pacf(tsPREiLd1, lag.max = 50)
```



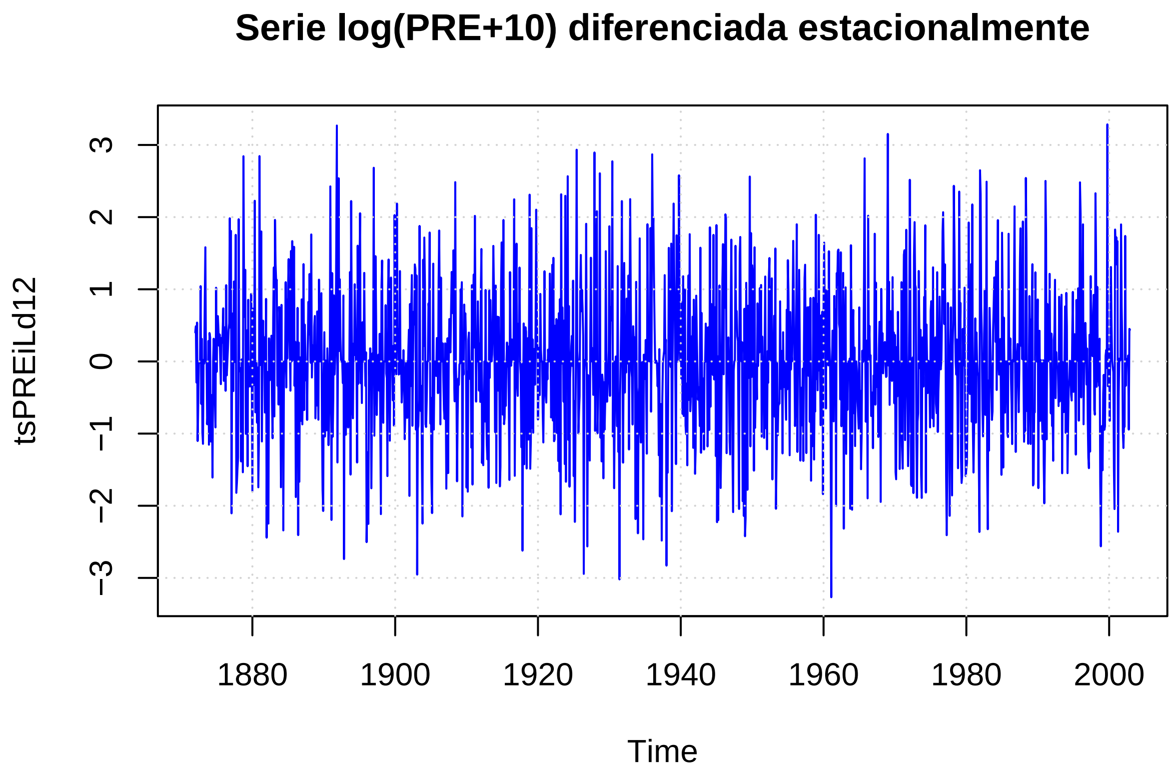

```
tsPREiLd12 = diff(tsPREiL, lag = 12, differences = 1)
acf(tsPREiLd12, lag.max = 50)
```



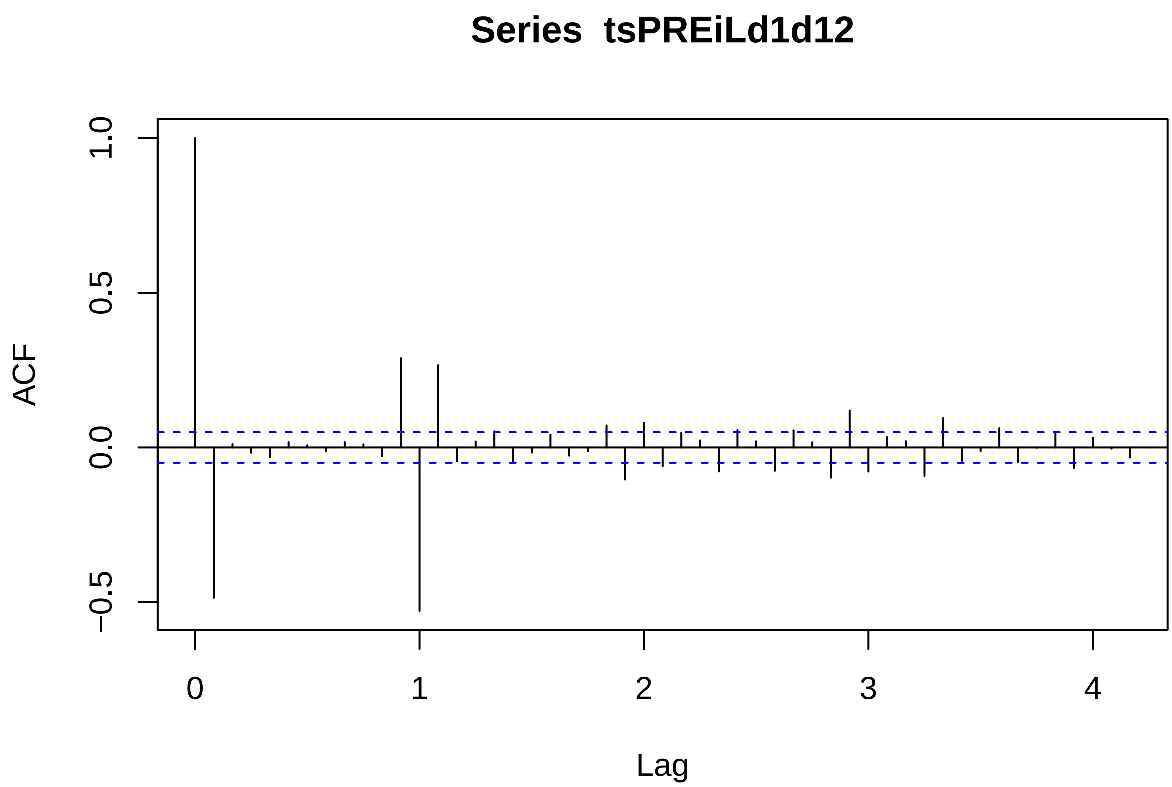

```
pacf(tsPREiLd12, lag.max = 50)
```



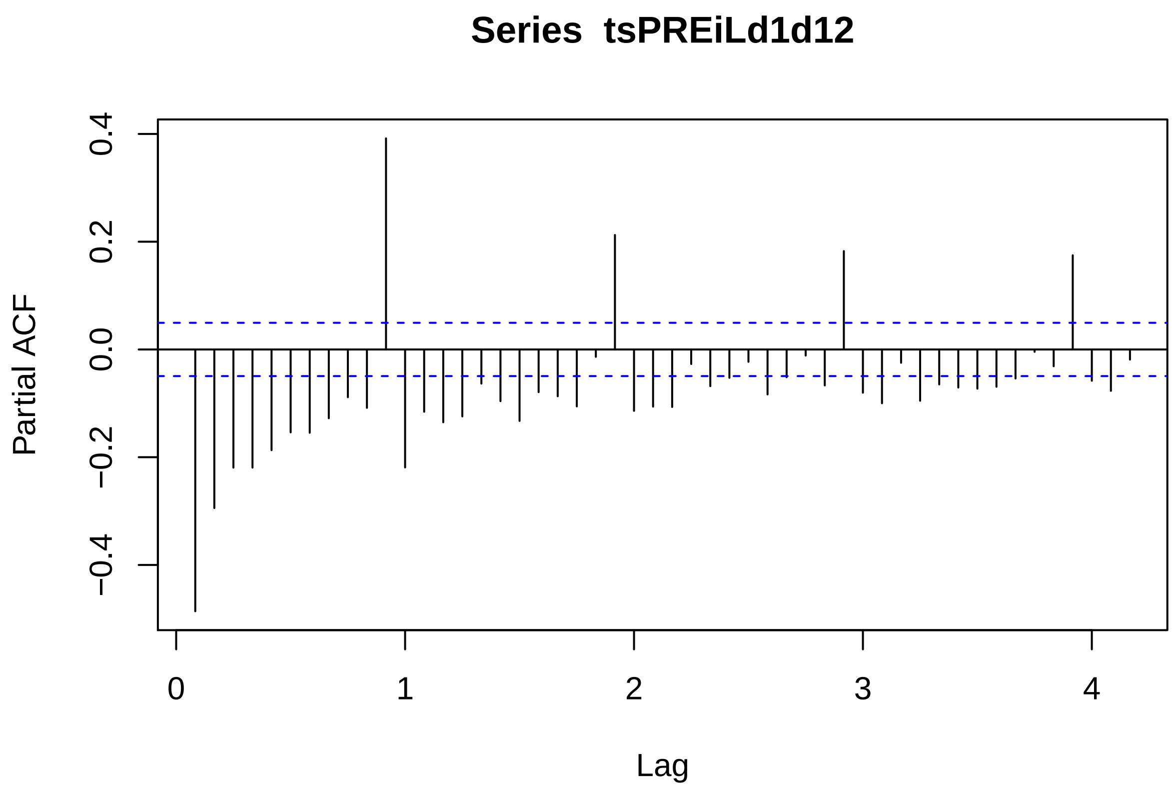

```
plot.ts(tsPREiLd12, col=4, main="Serie log(PRE+10) diferenciada estacionalmente")
grid()
```



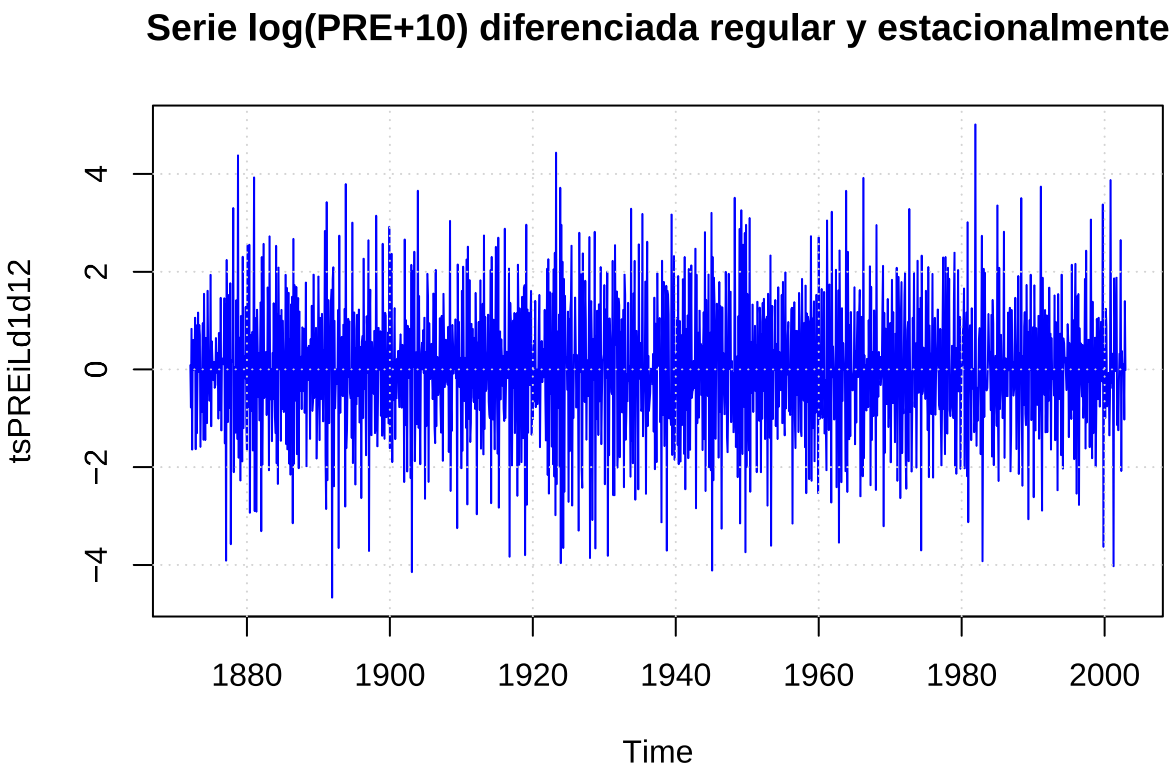

```r
tsPREiLd1d12 = diff(tsPREiLd1, lag = 12, differences = 1)
acf(tsPREiLd1d12, lag.max = 50)
```



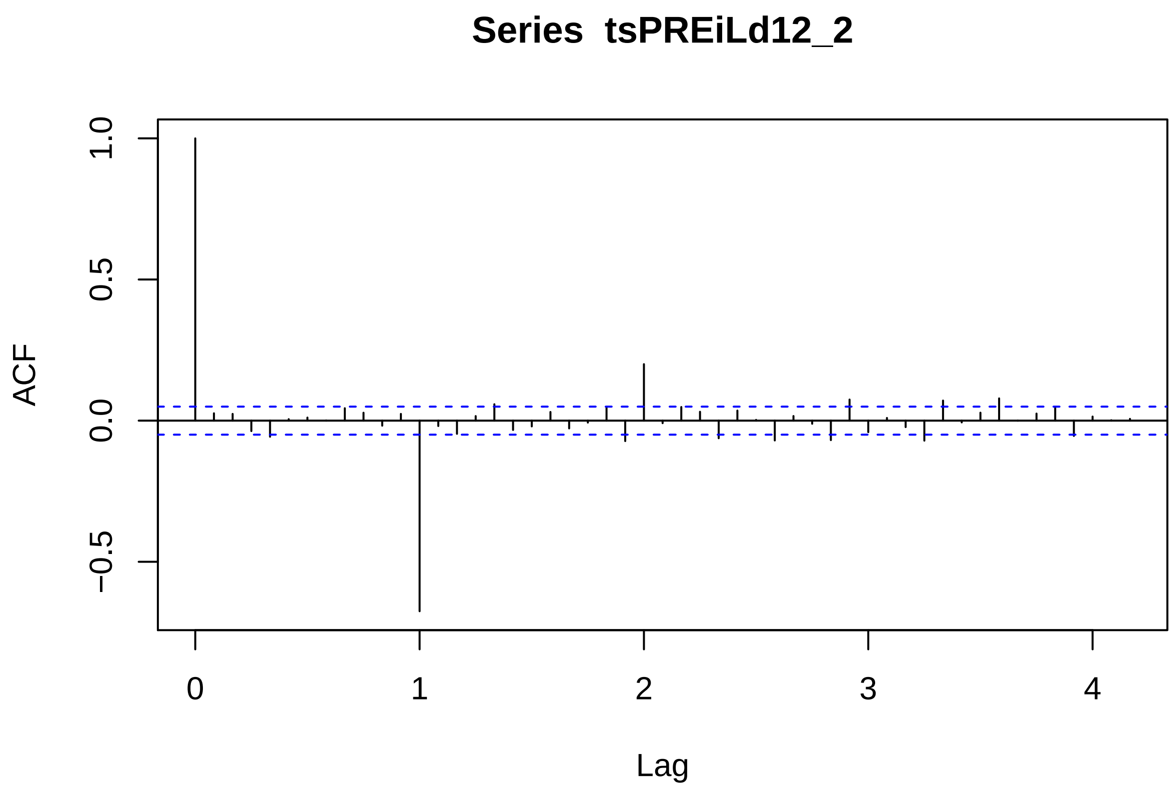

```
pacf(tsPREiLd1d12, lag.max = 50)
```



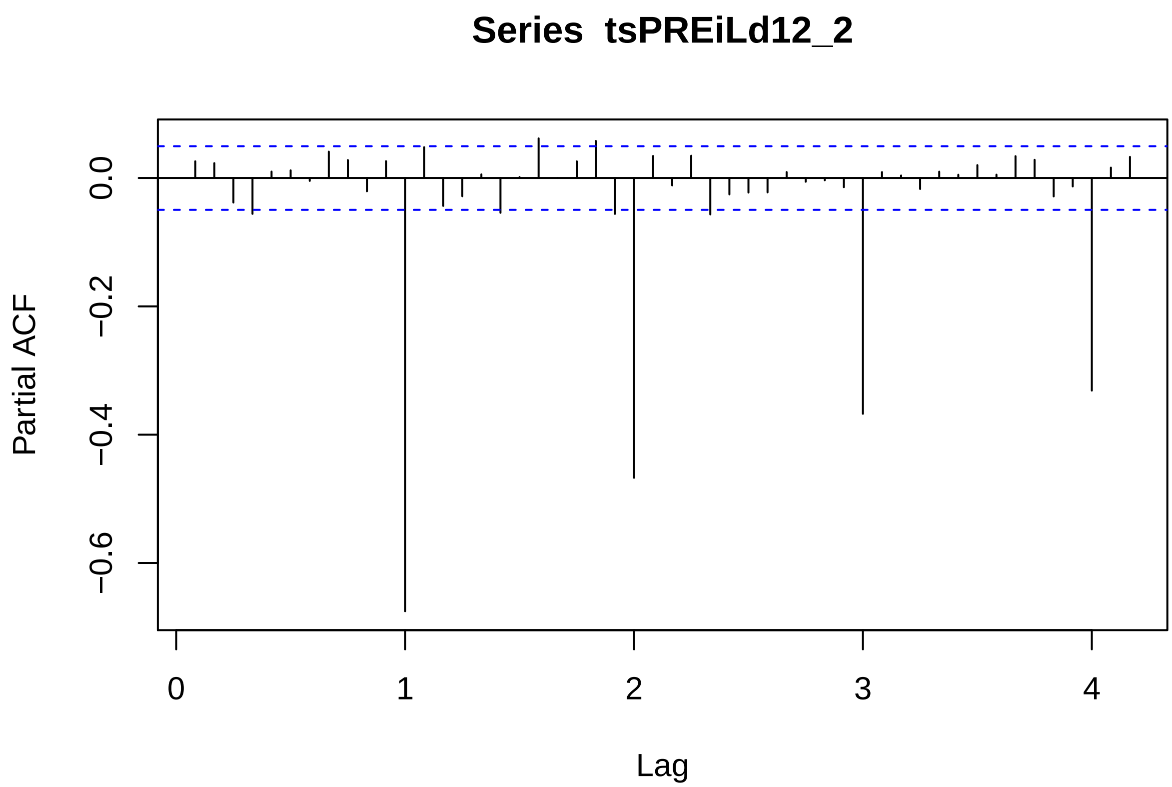

```r
plot.ts(tsPREiLd1d12, col=4, main="Serie log(PRE+10) diferenciada regular y estacionalmente")
grid()
```



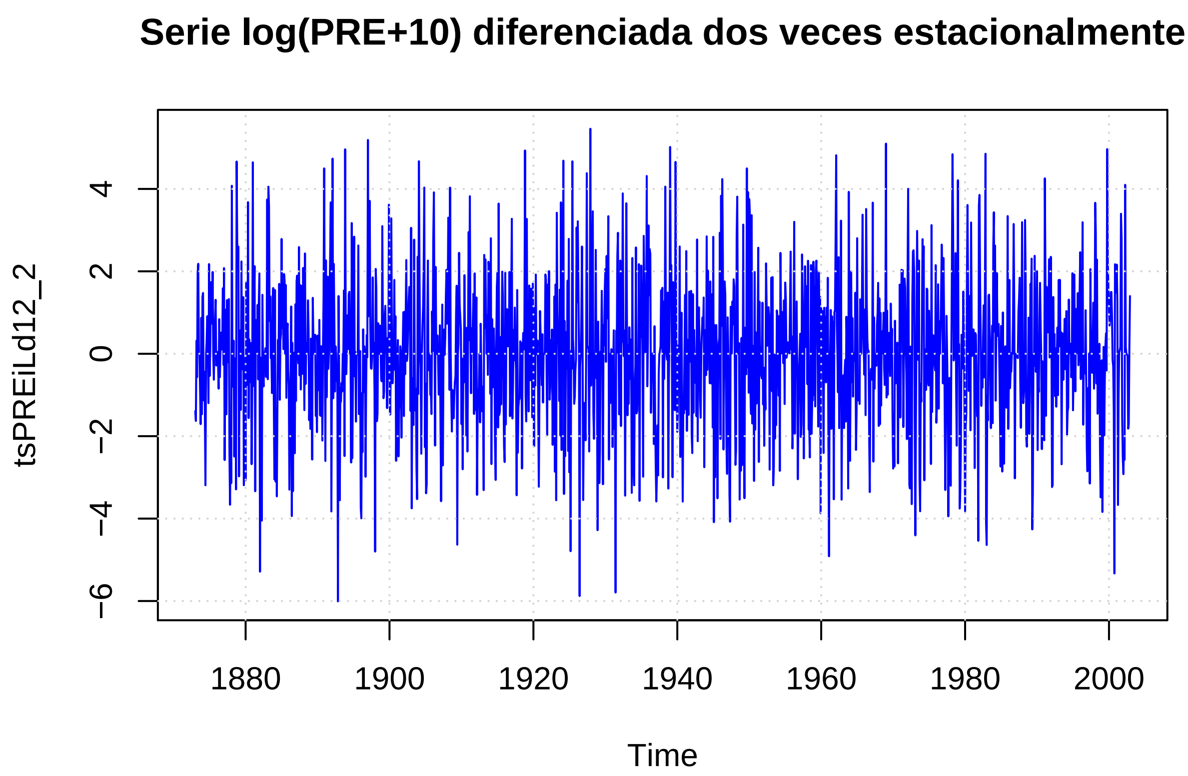

```r
tsPREiLd12_2 = diff(tsPREiL, lag = 12, differences = 2)
acf(tsPREiLd12_2, lag.max = 50)
```




```
pacf(tsPREiLd12_2, lag.max = 50)
```




```r
plot.ts(tsPREiLd12_2, col=4, main="Serie log(PRE+10) diferenciada dos veces estacionalmente")
grid()
```




```r
# Contraste estacionariedad
library('tseries')
```

```
##
## Attaching package: 'tseries'
## The following object is masked from 'package:imputeTS':
##
##     na.remove
```

```r
adf.test(tsPREiLd1)
```

```
## Warning in adf.test(tsPREiLd1): p-value smaller than printed p-value
##
##  Augmented Dickey-Fuller Test
##
## data:  tsPREiLd1
## Dickey-Fuller = -30.486, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(tsPREiLd12)
```

```
## Warning in adf.test(tsPREiLd12): p-value smaller than printed p-value
##
##  Augmented Dickey-Fuller Test
##
## data:  tsPREiLd12
## Dickey-Fuller = -18.858, Lag order = 11, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

```r
adf.test(tsPREiLd1d12)
```

```
## Warning in adf.test(tsPREiLd1d12): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  tsPREiLd1d12
## Dickey-Fuller = -15.242, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

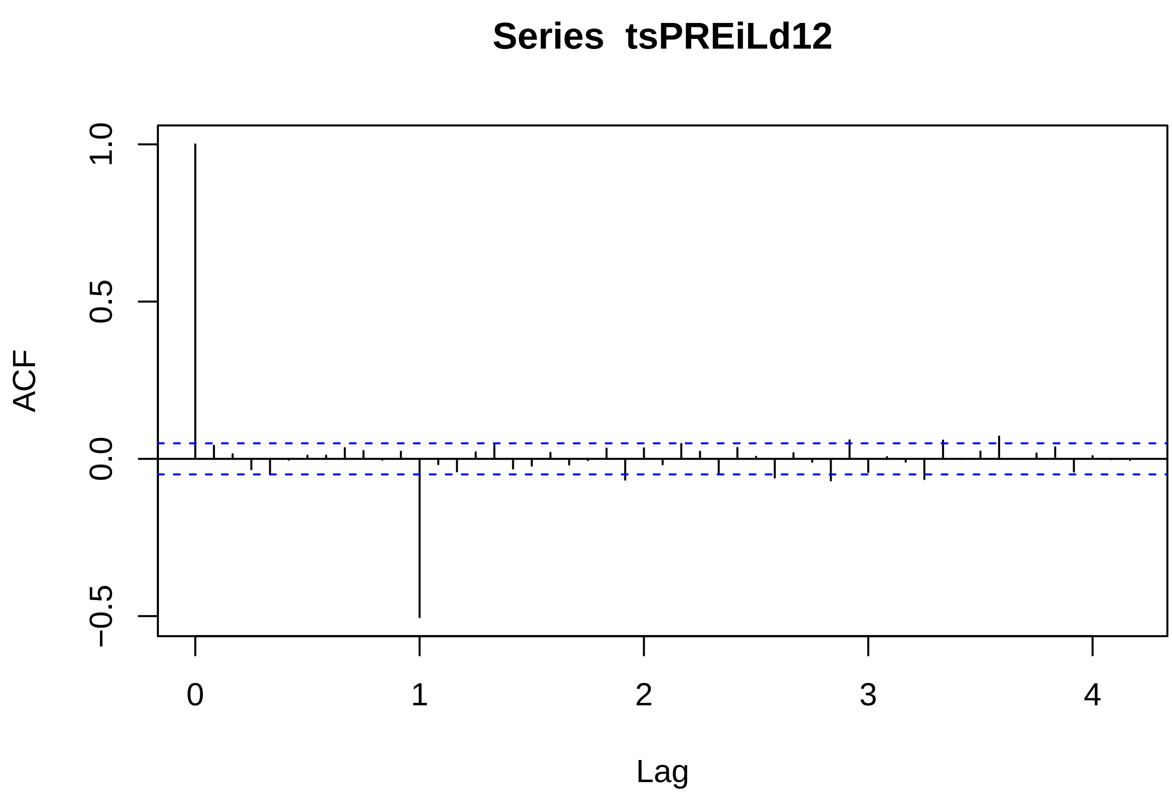
```r
adf.test(tsPREiLd12_2)
```

```
## Warning in adf.test(tsPREiLd12_2): p-value smaller than printed p-value
```

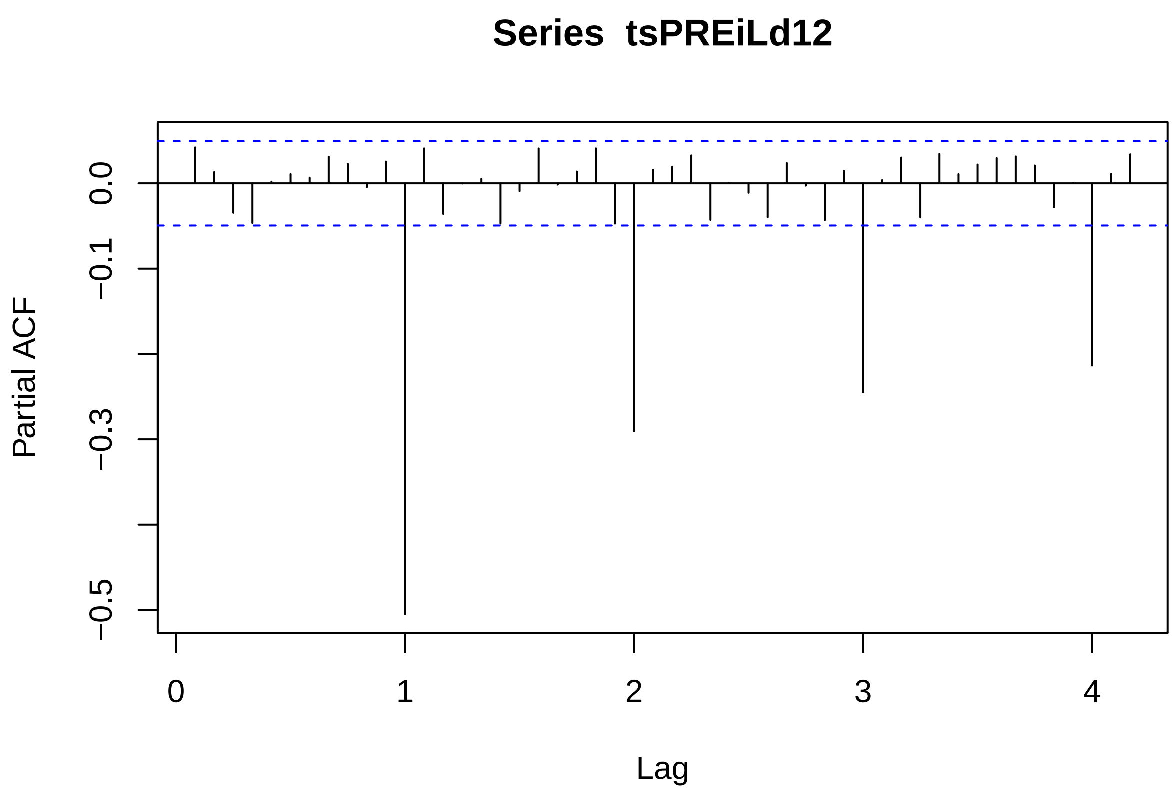
```
##
##  Augmented Dickey-Fuller Test
##
## data:  tsPREiLd12_2
## Dickey-Fuller = -24.85, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

10 Estructura ARIMA

```r
acf(tsPREiLd12, lag.max = 50)
```



```r
pacf(tsPREiLd12, lag.max = 50)
```




```r
# Exploramos modelos ARIMA(0:2,0,0)(0:2,0,0:1)[12] para tsPREiLd12

arimaCheck = function(serie, p, d, q, P, D, Q){
  arimafit = arima(serie, order=c(p,d,q), seasonal = list(order=c(P,D,Q), period = 12))
  cat('ARIMA (',p,d,q,')(',P,D,Q,')', '->', 'AIC = ', arimafit$aic, '\n')
}

for (p in 0:2){
  for (d in 0:0){
    for (q in 0:0){
      for (P in 0:2){
        for (D in 0:0){
          for (Q in 0:1){
            arimaCheck(tsPREiLd12, p,d,q, P,D,Q)
            }
          }
        }
    }
  }
}
```

```
## ARIMA ( 0 0 0 )( 0 0 0 ) -> AIC =  4539.811
## ARIMA ( 0 0 0 )( 0 0 1 ) -> AIC =  3534.134
## ARIMA ( 0 0 0 )( 1 0 0 ) -> AIC =  4081.831
```

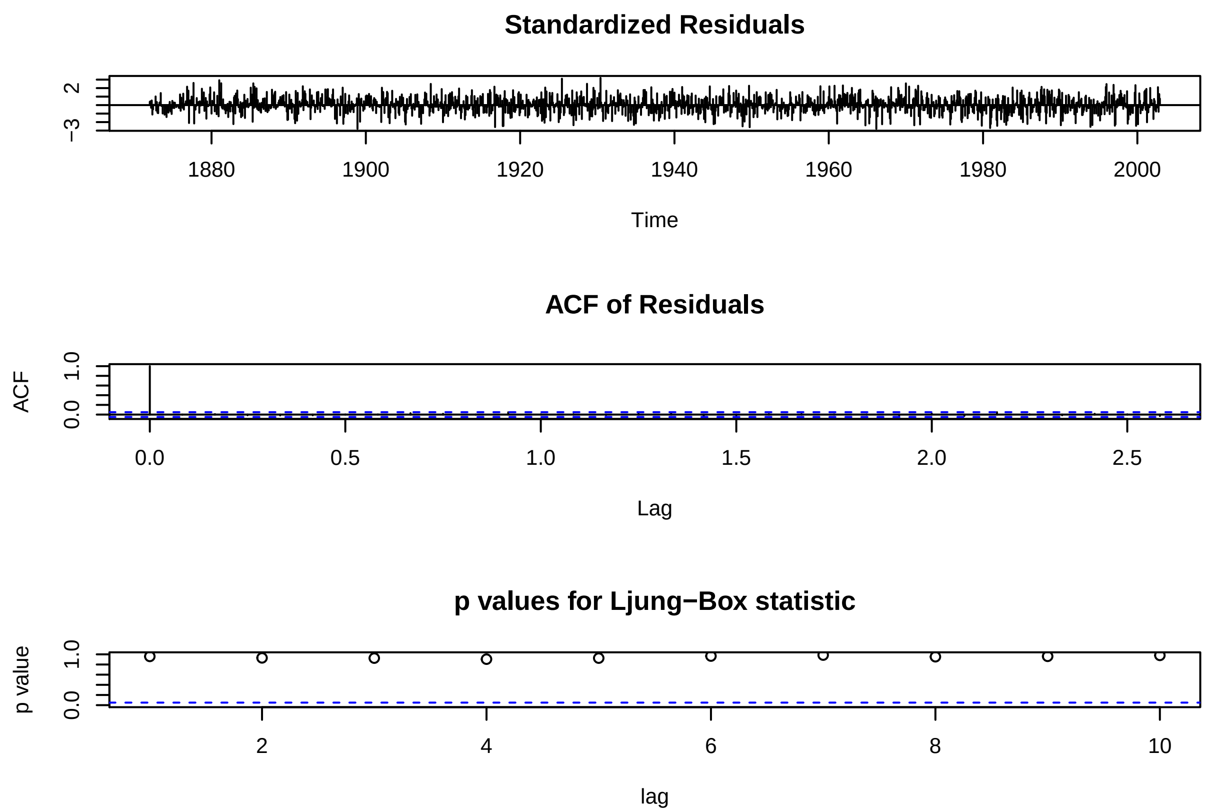
```
## ARIMA ( 0 0 0 )( 1 0 1 ) -> AIC =  3535.783
## ARIMA ( 0 0 0 )( 2 0 0 ) -> AIC =  3942.987
## ARIMA ( 0 0 0 )( 2 0 1 ) -> AIC =  3536.902
## ARIMA ( 1 0 0 )( 0 0 0 ) -> AIC =  4539.03
## ARIMA ( 1 0 0 )( 0 0 1 ) -> AIC =  3528.777
## ARIMA ( 1 0 0 )( 1 0 0 ) -> AIC =  4074.925
## ARIMA ( 1 0 0 )( 1 0 1 ) -> AIC =  3530.581
## ARIMA ( 1 0 0 )( 2 0 0 ) -> AIC =  3938.18
## ARIMA ( 1 0 0 )( 2 0 1 ) -> AIC =  3531.469
## ARIMA ( 2 0 0 )( 0 0 0 ) -> AIC =  4540.756
## ARIMA ( 2 0 0 )( 0 0 1 ) -> AIC =  3530.651
## ARIMA ( 2 0 0 )( 1 0 0 ) -> AIC =  4076.784
## ARIMA ( 2 0 0 )( 1 0 1 ) -> AIC =  3532.453
## ARIMA ( 2 0 0 )( 2 0 0 ) -> AIC =  3940.014
## ARIMA ( 2 0 0 )( 2 0 1 ) -> AIC =  3533.383
```

11 Estimación y diagnóstico

```r
(fit1 = arima(tsPREiLd12, order = c(1,0,0), seasonal = list(order = c(0,0,1), period = 12)))
```

```
##
## Call:
## arima(x = tsPREiLd12, order = c(1, 0, 0), seasonal = list(order = c(0, 0, 1),
##     period = 12))
##
## Coefficients:
##          ar1     sma1  intercept
##       0.0683  -0.9870      4e-04
## s.e.  0.0252   0.0116      6e-04
##
## sigma^2 estimated as 0.5348:  log likelihood = -1760.39,  aic = 3528.78
```

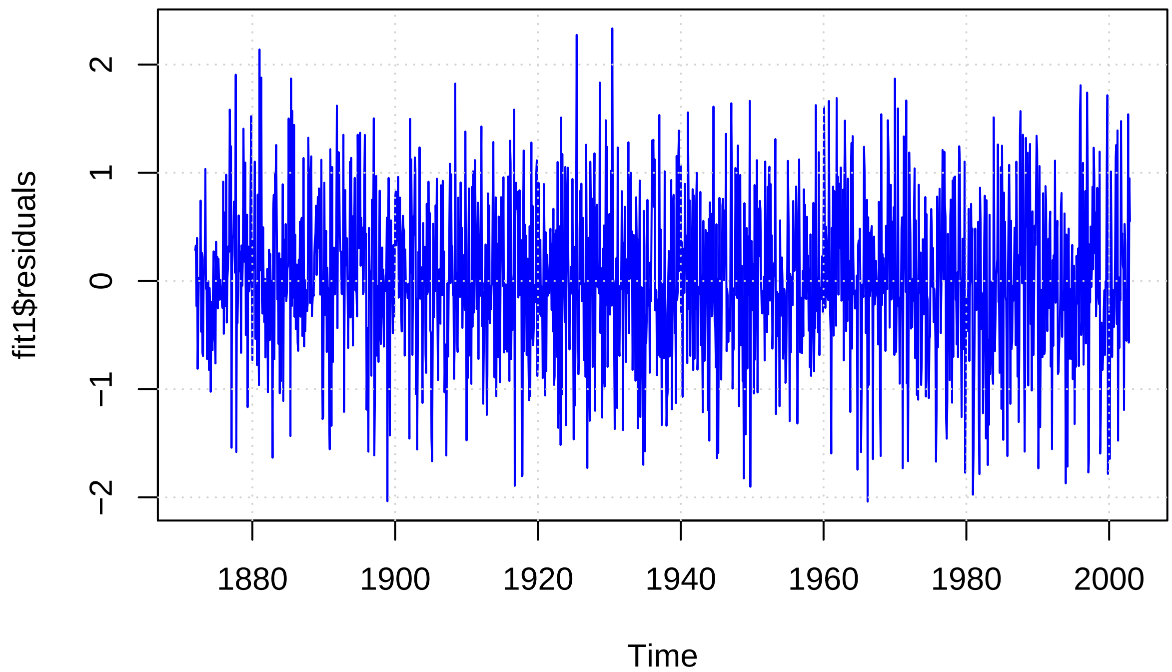
```r
tsdiag(fit1)
```




```
Box.test(fit1$residuals, lag=1, type = "Ljung")
```

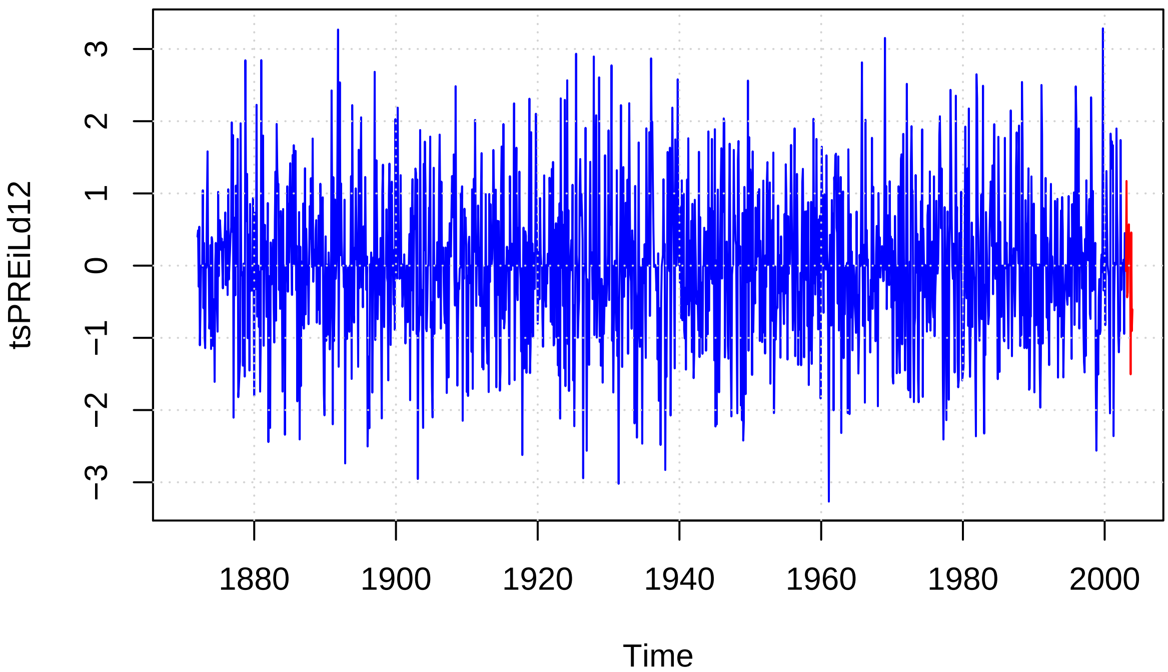
```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 0.0026316, df = 1, p-value = 0.9591
```

```
plot(fit1$residuals, col=4)
grid()
```



12 Predicción

```
plot(tsPREiLd12,xlim=c(1871, 2003), col=4)
fit1.pred = predict(fit1,n.ahead = 12)
lines(fit1.pred$pred,col=2)
grid()
```



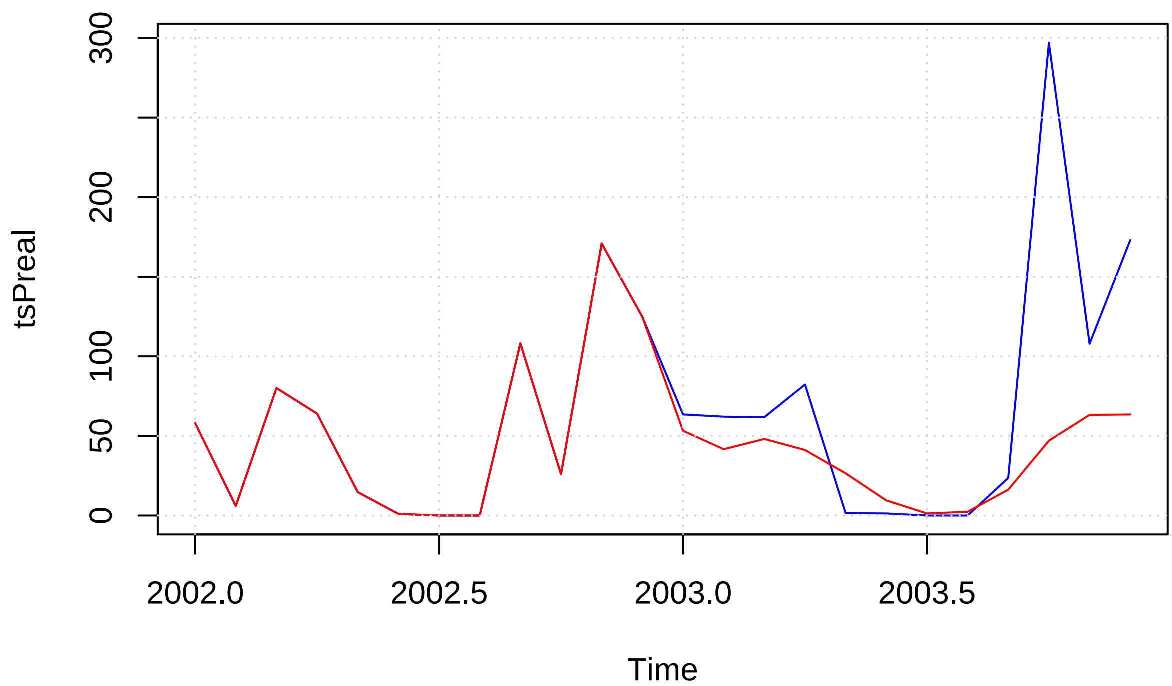

```r
tsPpred = exp(diffinv(fit1.pred$pred,lag=12, differences = 1, xi= tail(tsPREiL, 12)))-10

(tsPpred = ts(round(tsPpred,2), freq=12, start = c(2002,1), end = c(2003,12)))
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2002  58.10   6.00  80.10  64.00  14.70   1.00   0.00   0.00 108.20  26.00
## 2003  53.25  41.65  48.08  41.15  26.52   9.49   1.31   2.35  16.23  47.01
##         Nov    Dec
## 2002 170.90 124.80
## 2003  63.24  63.48
```

```r
(tsPreal = ts(tail(datos$PREFINAL, 30), freq=12, start = c(2002,1), end = c(2003,12)))
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 2002  58.1   6.0  80.1  64.0  14.7   1.0   0.0   0.0 108.2  26.0 170.9
## 2003  63.5  62.1  61.8  82.3   1.5   1.3   0.0   0.0  23.6 297.1 107.9
##        Dec
## 2002 124.8
## 2003 173.0
```

```r
plot.ts(tsPreal,col=4)
lines(tsPpred,col=2)
grid()
```



La librería forecast tiene la función auto.arima que permite la obtención del mejor modelo ARIMA de acuerdo a los criterios de información AIC, AICc o BIC y las restricciones que se quieran establecer en relación a los parámetros.

Se aplica a la serie ya diferenciada estacionalmente (tsPREiLd12) y se incluye traza de modelos probados.

```
library(forecast)
(fit2=auto.arima(tsPREiLd12,trace=TRUE,ic="aic"))
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2)(1,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with non-zero mean : 4539.811
##  ARIMA(1,0,0)(1,0,0)[12] with non-zero mean : 4079.677
##  ARIMA(0,0,1)(0,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)            with zero mean     : 4537.818
##  ARIMA(1,0,0)            with non-zero mean : 4539.876
##  ARIMA(1,0,0)(2,0,0)[12] with non-zero mean : 3947.476
##  ARIMA(1,0,0)(2,0,1)[12] with non-zero mean : Inf
##  ARIMA(0,0,0)(2,0,0)[12] with non-zero mean : 3951.277
##  ARIMA(2,0,0)(2,0,0)[12] with non-zero mean : 3950.244
##  ARIMA(1,0,1)(2,0,0)[12] with non-zero mean : 3949.358
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : 3943.951
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : 3942.418
##  ARIMA(2,0,1)(1,0,0)[12] with zero mean     : 4081.381
##  ARIMA(2,0,1)(2,0,1)[12] with zero mean     : Inf
```

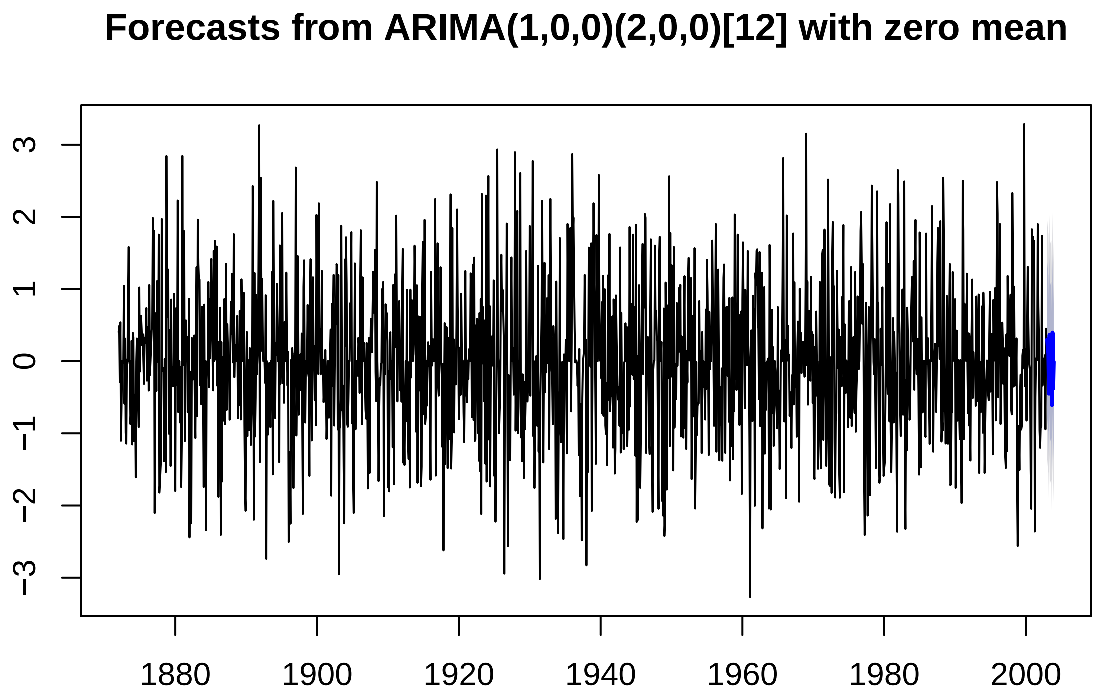
```
##  ARIMA(1,0,1)(2,0,0)[12] with zero mean     : 3947.422
##  ARIMA(3,0,1)(2,0,0)[12] with zero mean     : 3950.219
##  ARIMA(2,0,0)(2,0,0)[12] with zero mean     : 3948.311
##  ARIMA(2,0,2)(2,0,0)[12] with zero mean     : 3944.199
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 3945.54
##  ARIMA(3,0,2)(2,0,0)[12] with zero mean     : 3952.138
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,1)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(2,0,1)(2,0,0)[12] with non-zero mean : Inf
##  ARIMA(2,0,2)(2,0,0)[12] with zero mean     : Inf
##  ARIMA(1,0,0)(2,0,0)[12] with zero mean     : 3936.223
##
##  Best model: ARIMA(1,0,0)(2,0,0)[12] with zero mean
```

```
## Series: tsPREiLd12
## ARIMA(1,0,0)(2,0,0)[12] with zero mean
##
## Coefficients:
##          ar1     sar1     sar2
##       0.0660  -0.6539  -0.2913
## s.e.  0.0254   0.0241   0.0242
##
## sigma^2 estimated as 0.7113:  log likelihood=-1964.11
## AIC=3936.22   AICc=3936.25   BIC=3957.66
```

El modelo resultante seleccionado es distinto del obtenido anteriormente, con un AIC mayor (-> peor).

Vemos su predicción.

```r
plot(forecast(fit2,h=12))
```




```
forecast(fit2,h=12)
```

```
##          Point Forecast       Lo 80     Hi 80     Lo 95    Hi 95
## Jan 2003    0.300456349 -0.7803570 1.3812697 -1.352505 1.953418
## Feb 2003    0.186538695 -0.8966236 1.2697009 -1.470015 1.843092
## Mar 2003    0.106971164 -0.9762013 1.1901436 -1.549598 1.763540
## Apr 2003   -0.449633522 -1.5328060 0.6335390 -2.106203 1.206936
## May 2003    0.365294087 -0.7178784 1.4484666 -1.291275 2.021863
## Jun 2003   -0.035426209 -1.1185987 1.0477463 -1.691996 1.621143
## Jul 2003    0.003608698 -1.0795638 1.0867812 -1.652961 1.660178
## Aug 2003    0.013618268 -1.0695542 1.0967908 -1.642951 1.670188
## Sep 2003   -0.606771064 -1.6899436 0.4764014 -2.263340 1.049798
## Oct 2003    0.395743993 -0.6874285 1.4789165 -1.260825 2.052313
## Nov 2003   -0.378556885 -1.4617294 0.7046156 -2.035126 1.278013
## Dec 2003   -0.007175302 -1.0903478 1.0759972 -1.663745 1.649394
```

Comprobamos su predicción:

```
fit2.pred = predict(fit2,n.ahead = 12)
tsPpred2 = exp(diffinv(fit2.pred$pred,lag=12, differences = 1, xi= tail(tsPREiL, 12)))-10

(tsPpred2 = ts(round(tsPpred2,2), freq=12, start = c(2002,1), end = c(2003,12)))
```

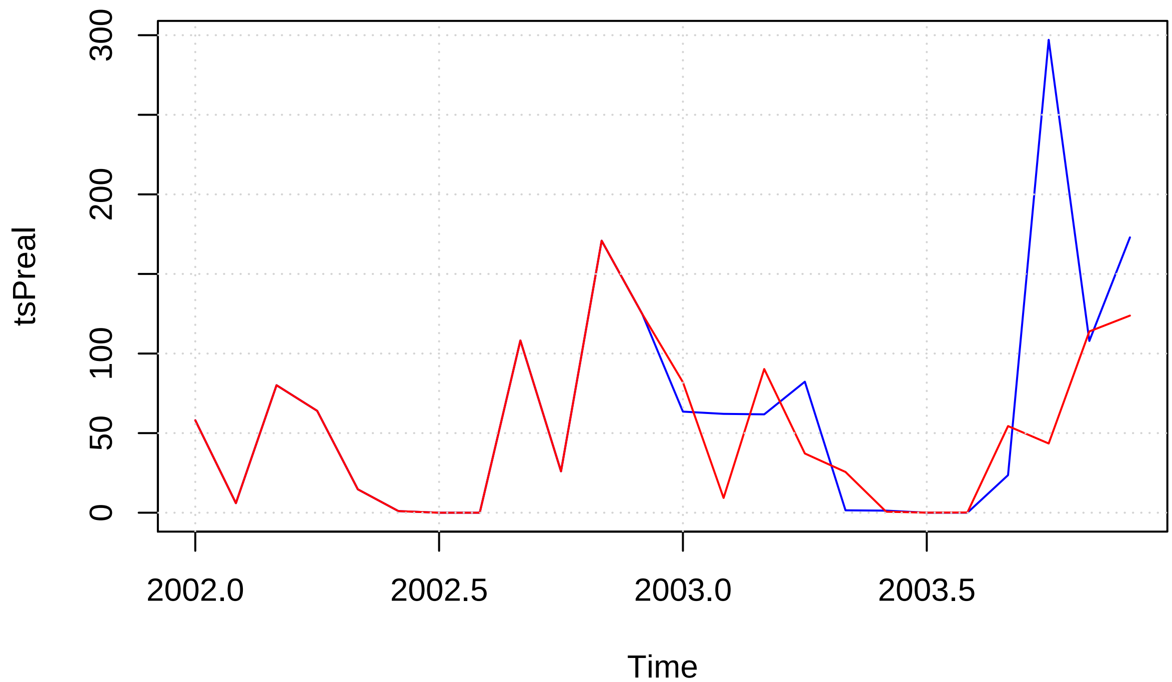
```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2002  58.10   6.00  80.10  64.00  14.70   1.00   0.00   0.00 108.20  26.00
## 2003  81.97   9.28  90.27  37.20  25.59   0.62   0.04   0.14  54.43  43.48
##         Nov    Dec
## 2002 170.90 124.80
```

```
## 2003 113.89 123.84
```

```
(tsPreal = ts(tail(datos$PREFINAL, 30), freq=12, start = c(2002,1), end = c(2003,12)))
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 2002  58.1   6.0  80.1  64.0  14.7   1.0   0.0   0.0 108.2  26.0 170.9
## 2003  63.5  62.1  61.8  82.3   1.5   1.3   0.0   0.0  23.6 297.1 107.9
##        Dec
## 2002 124.8
## 2003 173.0
```

```
plot.ts(tsPreal,col=4)
lines(tsPpred2,col=2)
grid()
```



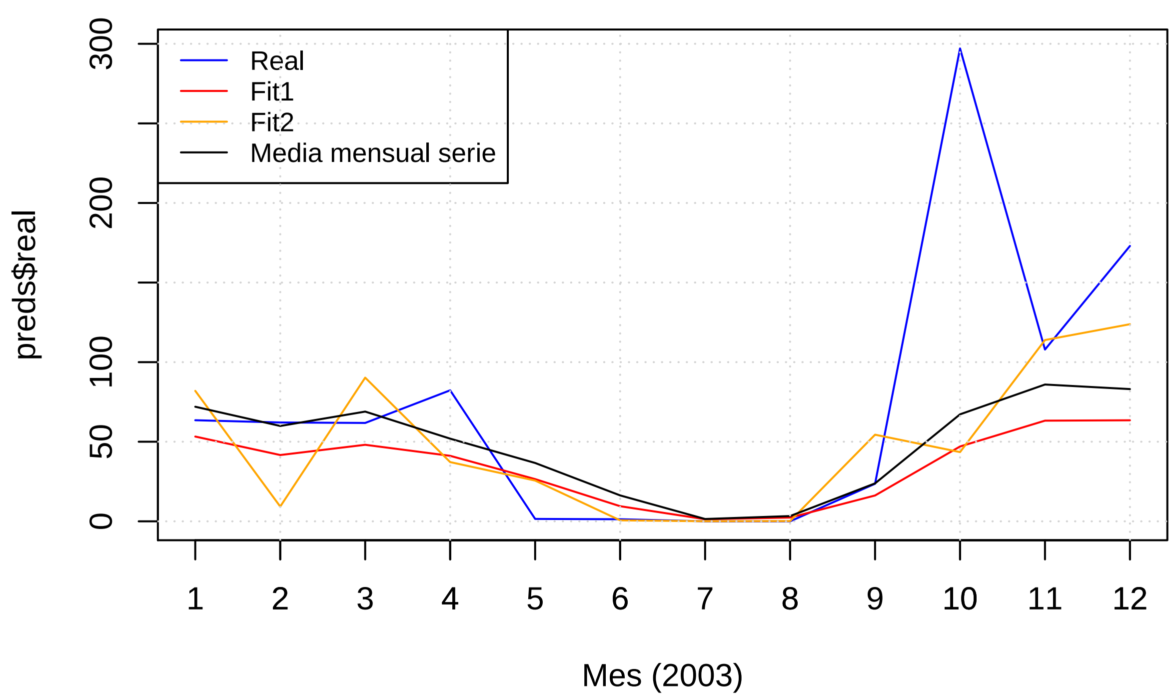
Las comparamos:

```
(
preds = data.frame('real'=as.vector(tail(tsPreal,12)),
                   'pred1'=as.vector(tail(tsPpred,12)),
                   'pred2'=as.vector(tail(tsPpred2,12)),
                   'media'=as.vector(tapply(tsPREi, cycle(tsPREi), mean)))
)
```

```
##    real pred1  pred2     media
## 1  63.5 53.25  81.97 71.993182
## 2  62.1 41.65   9.28 59.910985
## 3  61.8 48.08  90.27 68.936742
## 4  82.3 41.15  37.20 51.923106
```

```
## 5    1.5 26.52  25.59 36.644697
## 6    1.3  9.49   0.62 16.283333
## 7    0.0  1.31   0.04  1.456818
## 8    0.0  2.35   0.14  3.300379
## 9   23.6 16.23  54.43 23.838636
## 10 297.1 47.01  43.48 67.217424
## 11 107.9 63.24 113.89 85.943561
## 12 173.0 63.48 123.84 83.077273
```

```r
plot(preds$real,col=4, type='l', xlab = 'Mes (2003)')
axis(side=1, at=c(1:12))
lines(preds$pred1,col='red')
lines(preds$pred2,col='orange')
lines(preds$media,col=1)
legend("topleft",legend=c("Real", "Fit1", "Fit2","Media mensual serie"),
       col=c("blue","red", "orange","black"), lty=1, cex=0.8)
grid()
```




```r
(ECM1 = mean((preds$real-preds$pred1)**2))
```

```
## [1] 6641.134
```

```r
(ECM2 = mean((preds$real-preds$pred2)**2))
```

```
## [1] 6190.22
```

Comparando las predicciones de fit1 y fit2, vemos que el segundo modelo proporciona una mejor predicción, aunque, como vimos, su AIC es mayor.