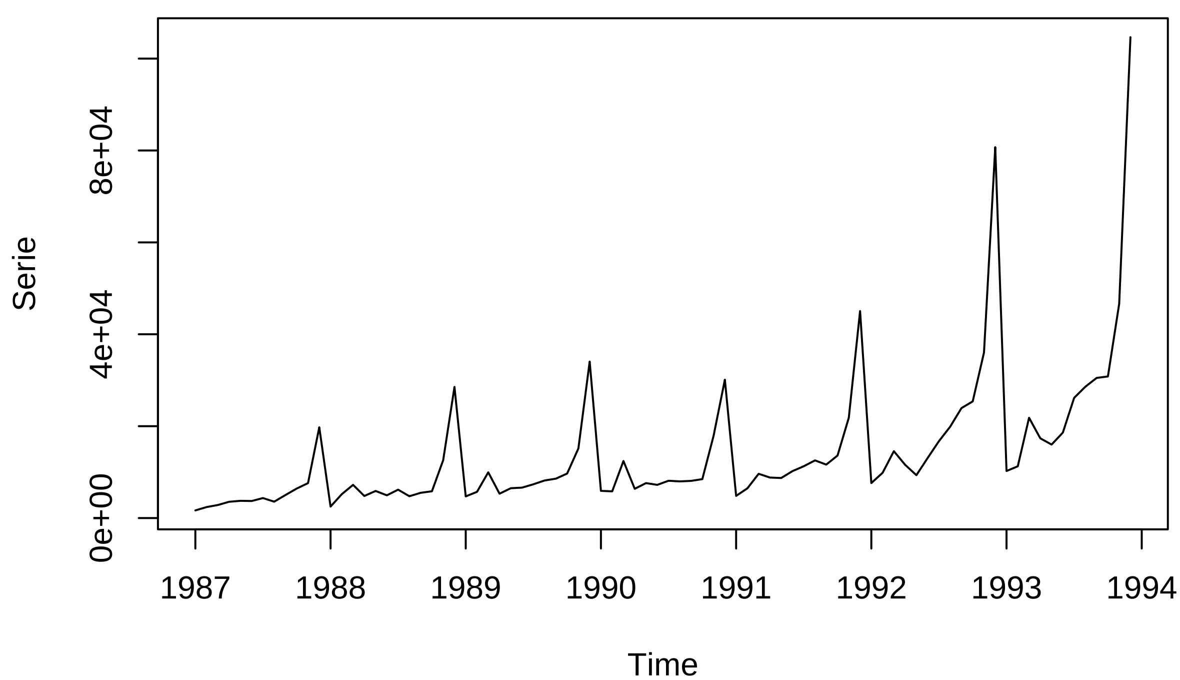
Tarea 1

Cuéllar, E. Tapia, J. Maciel, J. Saldaña, R. Miranda, G

15/Oct/2021

1.- Grafique los datos. Describa lo que observe de su información (varianza contante o no constante, tendencia, ciclos estacionales, periodicidad de los ciclos).

Gráfica de la serie de tiempo



Varianza

Podemos observar una varianza creciente conforme pasa el tiempo, teniendo primero un ligero aumento de 1987 a 1990 de manera lineal, despúes parece que se mantene constante de 1990 a 1991 para posteriormente crecer demasiado de 1991 a 1994 Es decir, no es constante la varianza. Lo que hace que se dispare es la temporada alta vacacional (Al parecer, en noviembre y diciembre, ya que analizamos una playa en Australia, donde el verano comienza en diciembre). Podemos comprobarlo con un bp test:

```r
# Ho: (Homocedasticidad)  vs  H1: La varianza no es constate (Heterocedasticidad)
t1 = seq(1987+0/12, 1993+11/12, by = 1 / 12)
bptest(Serie ~t1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Serie ~ t1
## BP = 5.9528, df = 1, p-value = 0.01469
```

Inclusive podemos ver de una vez que no es estacionaria tanto con el test de Dickey-Fuller como con el de Phillips:

```r
# Ho: La serie no es estacionaria vs. H1: La serie es estacionaria
adf.test(Serie)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Serie
## Dickey-Fuller = -2.0809, Lag order = 4, p-value = 0.5427
## alternative hypothesis: stationary
```

```r
#Por lo tanto:

# Ho: La serie es estacionaria vs. H1: La serie no es estacionaria
kpss.test(Serie)
```

```
## Warning in kpss.test(Serie): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  Serie
## KPSS Level = 1.3089, Truncation lag parameter = 3, p-value = 0.01
```

Tendencia

La tendencia parece ser en general creciente, repitiendo casi el mismo patrón que la varianza: Es creciente de manera ligera (lineal, con pendiente pequeña) de 1987 a 1990, para depués decrecer un poco de 1990 a 1991, sin embargo crece de manera cuadrática, al parecer, de 1991 a 1994.

Ciclos estacionales

Dado que estamos analizando una base de datos de ventas mensuales de una tienda de souvenirs en una playa en Asutralia, hace todo el sentido del mundo que tenga un ciclo ya que tanto en las ventas como visitas a sitios vacacionales hay una fuerte dependencia en los meses del año. Esto lo confirmamos con la gráfica, donde se observa un ciclo estacional bastante claro.

Periodicidad de los ciclos

Complementando el comentario del punto anterior, en la gráfica observamos que el ciclo es anual. De enero a febrero (o los primeros meses del año)parece crecer ligeramente, depués baja un poco para crecer de manera ligera nuevamente, pero al llegar lo que parece ser noviembre y diciembre (o los últimos meses del año)crece exageradamente; posteriormente baja de diciembre a enero y se repite el ciclo.

2.-Si la base presenta datos faltantes NA. Use algún método de imputación de la paquetería imputeTS.

No hay ningún NA, como podemos observar

```
## [1] 0
```

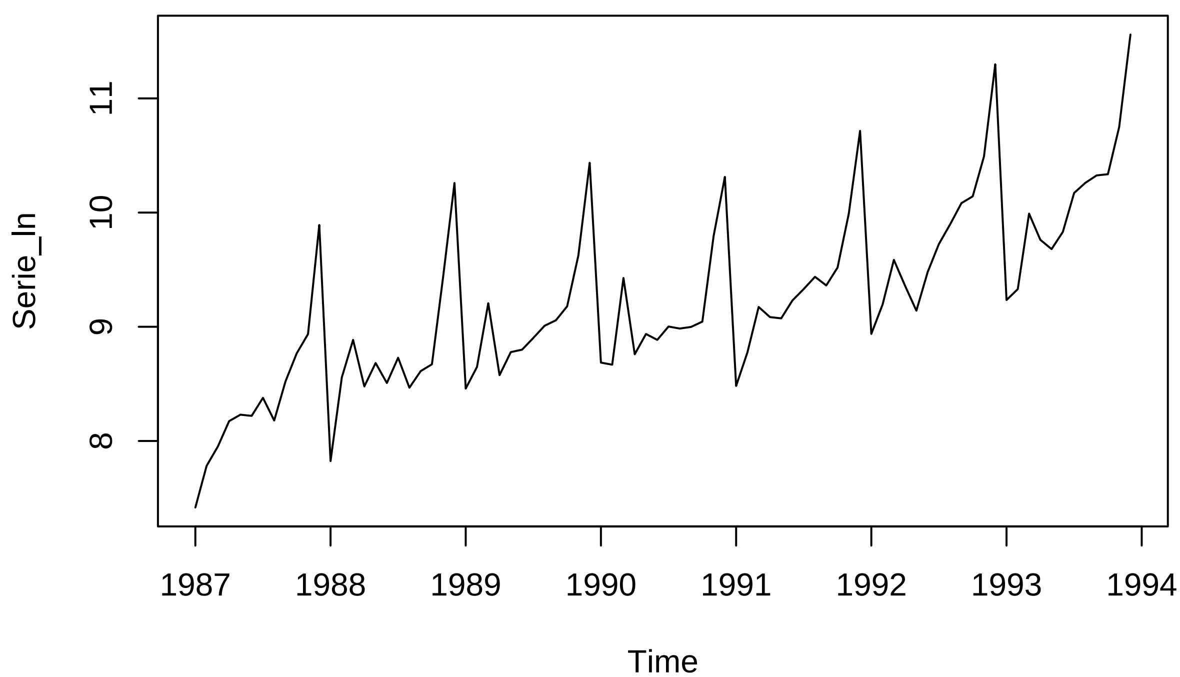
Por lo que no es necesario aplicar ningún método de imputación

3.- Use distintos métodos de descomposición de las series para obtener sus componentes (tendencia y ciclos estacionales), en específico use los siguientes:

(a) Ajuste de curvas (modelos deterministas o de regresión). Realice un pronóstico de 3 años futuros.

Primero realizareos una transformación con el logarítmo para estabilizar la varianza

```
Serie_ln<-(log(Serie))
ts.plot(Serie_ln)
```



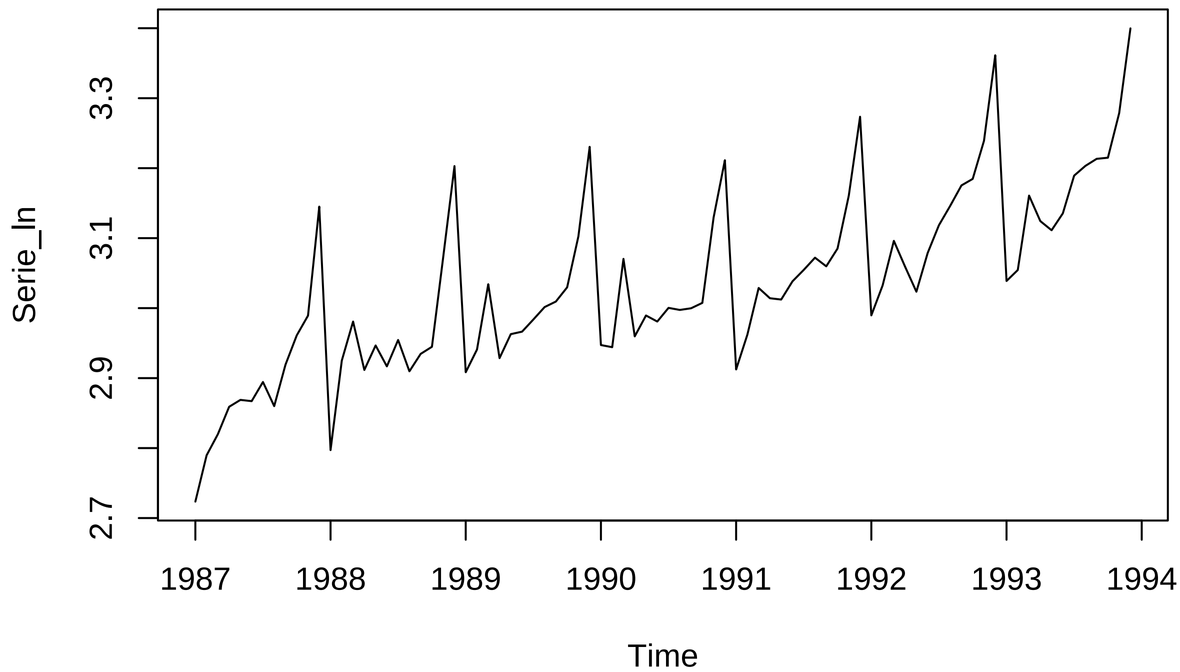
Al parecer, la varianza ya se estabilizó considerablemente (De manera gráfica). ¡Comprobémoslo con un bp test!

```
bptest(Serie_ln~t1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Serie_ln ~ t1
## BP = 0.3213, df = 1, p-value = 0.5708
```

¡Se pudo estabilizar! Pero, si tomamos la raíz, quizás podamos estabilizarla un poco más. comprobémoslo:

```
Serie_ln<-sqrt(log(Serie))
ts.plot(Serie_ln)
```




```r
bptest(Serie_ln~t1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Serie_ln ~ t1
## BP = 0.042518, df = 1, p-value = 0.8366
```

Nos quedaremos entonces con la raíz del logarítmo, puesto que presenta un p-value mayor.

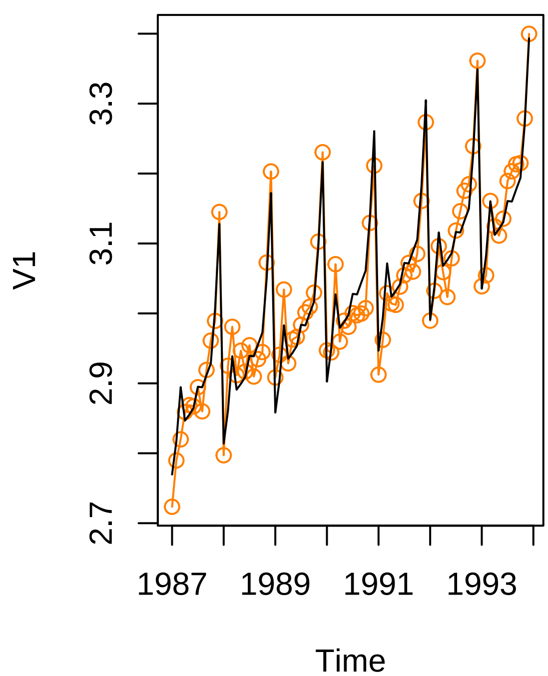
Como la tendencia parece seguir un comportamiento de un polinomio de grado 2, intentaremos ajustarle una curva con dicho polinomio

```r
M <- factor(cycle(Serie_ln))
t = time(Serie_ln)-1987
regresion_1= lm(Serie_ln ~ 0+ t+M,  na.action=NULL)
summary(regresion_1)
```

```
##
## Call:
## lm(formula = Serie_ln ~ 0 + t + M, na.action = NULL)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.074594 -0.019946  0.001211  0.017548  0.063774
##
## Coefficients:
##     Estimate Std. Error t value Pr(>|t|)
```

```
## t   0.044265   0.001669   26.52   <2e-16 ***
## M1  2.769652   0.012602  219.78   <2e-16 ***
## M2  2.813350   0.012658  222.26   <2e-16 ***
## M3  2.887112   0.012715  227.06   <2e-16 ***
## M4  2.835616   0.012774  221.99   <2e-16 ***
## M5  2.840288   0.012833  221.32   <2e-16 ***
## M6  2.846410   0.012894  220.75   <2e-16 ***
## M7  2.872990   0.012956  221.74   <2e-16 ***
## M8  2.868526   0.013020  220.32   <2e-16 ***
## M9  2.882184   0.013084  220.28   <2e-16 ***
## M10 2.895120   0.013150  220.16   <2e-16 ***
## M11 2.969272   0.013217  224.66   <2e-16 ***
## M12 3.087268   0.013284  232.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0306 on 71 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 6.363e+04 on 13 and 71 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,2))
plot(Serie_ln, type="o",col='darkorange1')
lines(fitted(regresion_1), col='black')
```



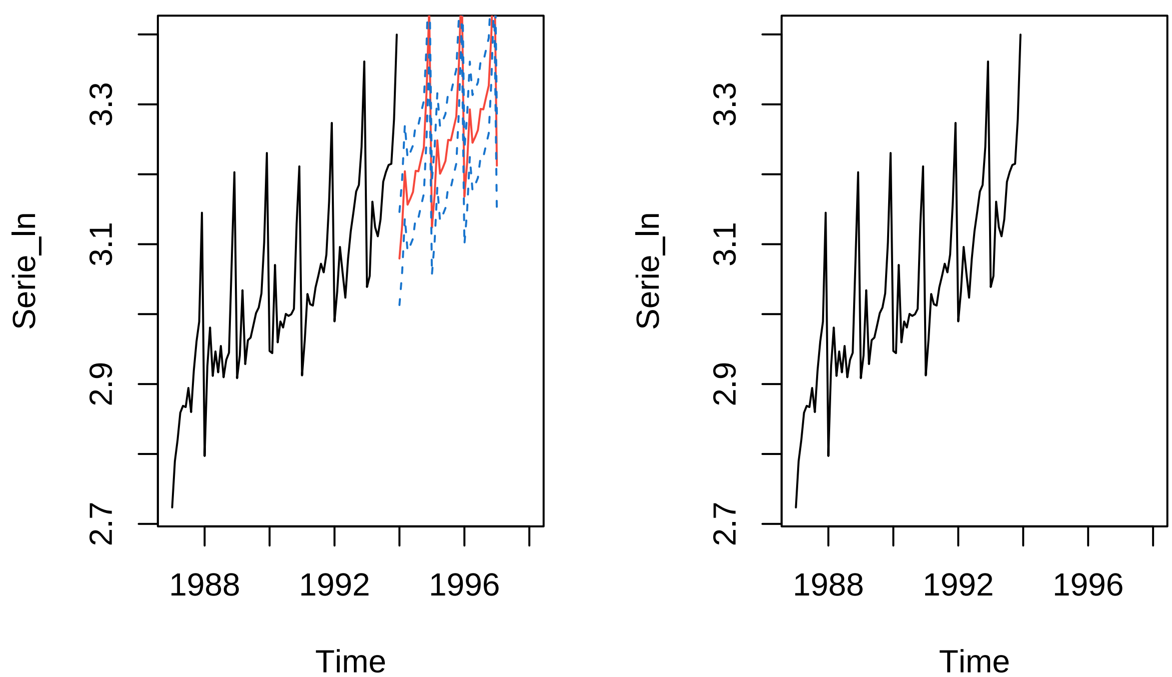
Observamos que todos los valores,tienen un p-value menor a 0.05. Entonces, quitando el efecto de ciclos estacionales, sigue una tendencia lineal.

$\mathbb{E}[X_t] = 0.26872t + M$

Donde M es la parte del ciclo estacional.

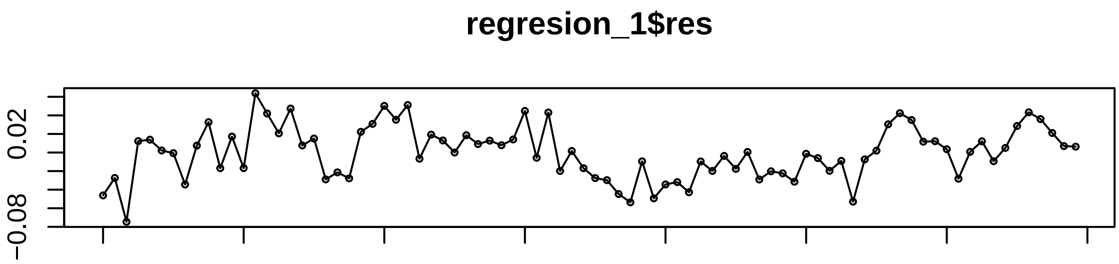
Ajustamos:

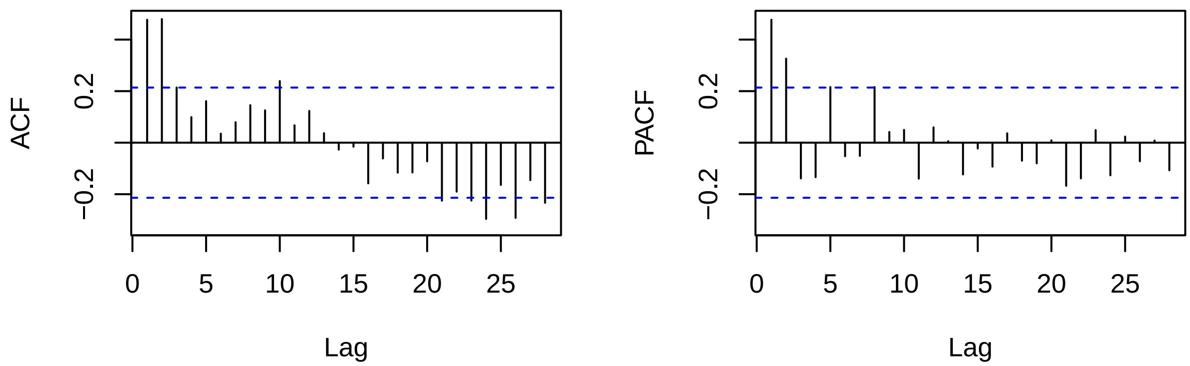
```
tnew = 7 + seq(0,3,length.out=37)
Mnew = factor(c((1:12),(1:12),(1:12),1))
pred1<- predict(regresion_1, newdata=list(t=tnew, M=Mnew), interval="prediction")
par(mfrow=c(1,2))
ts.plot(Serie_ln, xlim=c(1987,1998))
lines(1987+tnew,(pred1[,1]), lty=1,col=2)
lines(1987+tnew,(pred1[,2]), lty=2,col=4)
lines(1987+tnew,(pred1[,3]), lty=2,col=4)
ts.plot(Serie_ln, xlim=c(1987,1998))
lines(1987+tnew,exp(pred1[,1]**2), lty=1,col=2)
lines(1987+tnew,exp(pred1[,2]**2), lty=2,col=4)
lines(1987+tnew,exp(pred1[,3]**2), lty=2,col=4)
```



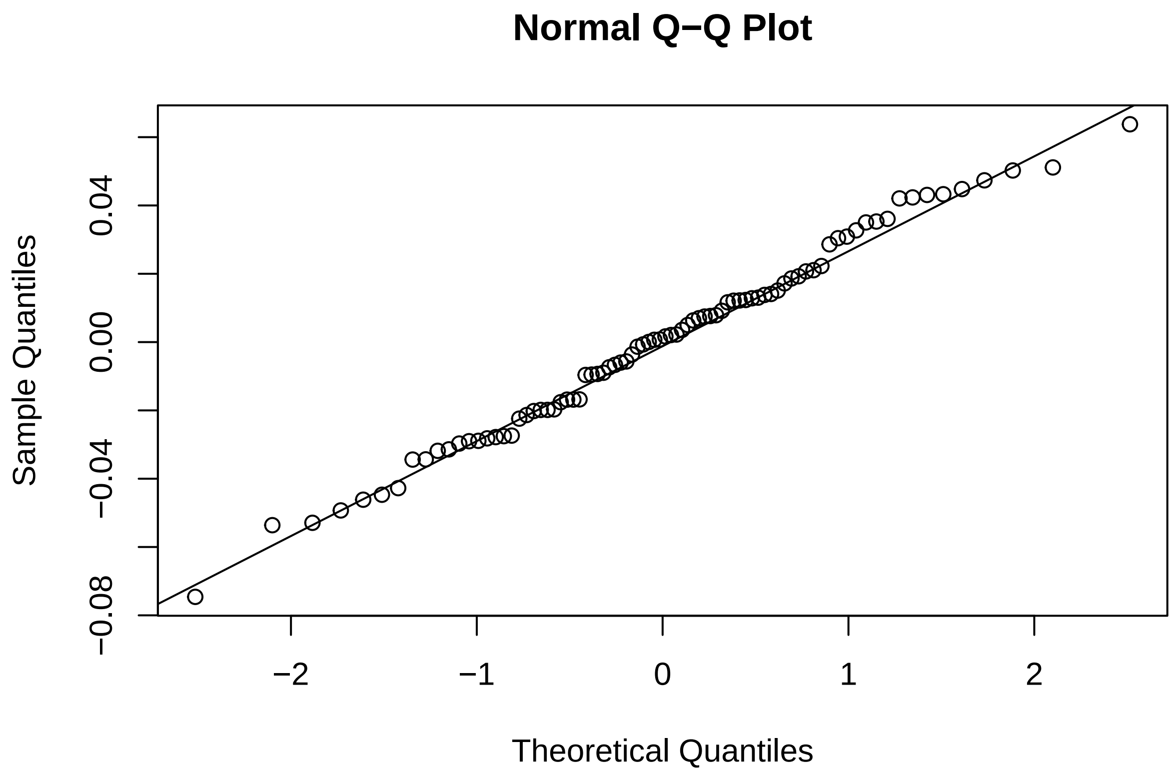
Ahora comprobemos los supuestos de regresión

```
tsdisplay(regresion_1$res)
```






```
qqnorm(regresion_1$res)
qqline(regresion_1$res)
```




```r
ad.test(regresion_1$res)
```

```
##
##  Anderson-Darling normality test
##
## data:  regresion_1$res
## A = 0.23236, p-value = 0.7937
```

```r
bptest(regresion_1)
```

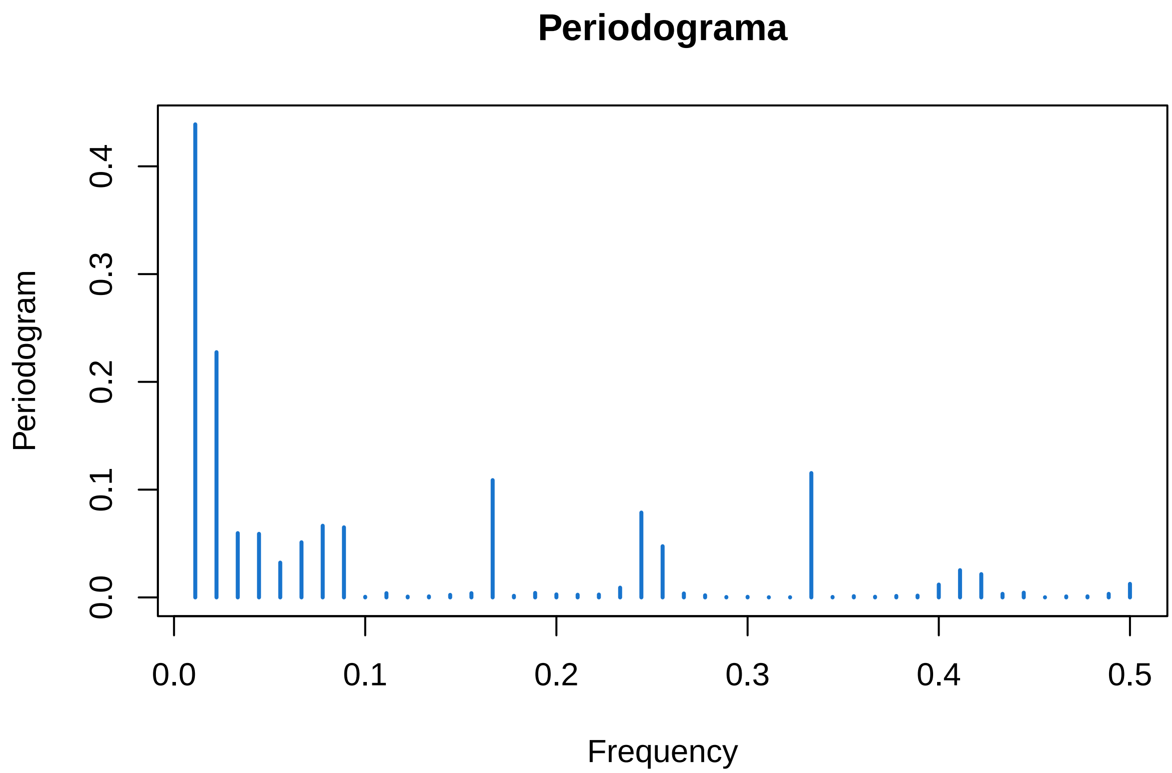
```
##
##  studentized Breusch-Pagan test
##
## data:  regresion_1
## BP = 24.664, df = 12, p-value = 0.0165
```

Pasamos los tests de Normalidad y homocedasticidad.

(b) Filtros lineales o suavizamientos exponenciales. Realice un pronóstico de 3 años futuros.

Vamos a descomponer usando filtros lineales. Con Holt Winters trabajaremos el suavizamiento exponencial y el pronóstico.

```r
#Realizamos mediante filtros lineales

#Veamos la tendencia y los ciclos
Xt = Serie_ln
p = periodogram(Xt, main="Periodograma", col=4) # Obtenemos el periodograma
```




```
names(p)
```

```
##  [1] "freq"      "spec"      "coh"       "phase"     "kernel"    "df"
##  [7] "bandwidth" "n.used"    "orig.n"    "series"    "snames"    "method"
## [13] "taper"     "pad"       "detrend"   "demean"
```

```
# Ordenamos de mayor a menor las estimaciones del periodograma.
spec = sort(p$spec, decreasing = TRUE)
(spec = spec[1:7]) # Nos quedamos con los coeficientes de mayor frecuencia.
```

```
## [1] 0.43894447 0.22754984 0.11538547 0.10882734 0.07875511 0.06651026 0.06502999
```

```
i = match(spec, p$spec) # Buscamos sus indices en el periodograma.
d = p$freq # Vemos las frecuencias del periodograma.
d = d[i] # Nos quedamos con las frecuencias que nos interesan.

cbind(spec,d,i)#
```

```
##            spec          d  i
## [1,] 0.43894447 0.01111111  1
## [2,] 0.22754984 0.02222222  2
## [3,] 0.11538547 0.33333333 30
## [4,] 0.10882734 0.16666667 15
## [5,] 0.07875511 0.24444444 22
## [6,] 0.06651026 0.07777778  7
## [7,] 0.06502999 0.08888889  8
```

```r
d = 1 / d # Obtenemos los parametros para utilizar en promedios moviles.
d = floor(d) #
(d = sort(d))
```

```
## [1]  2  4  6 11 12 45 90
```

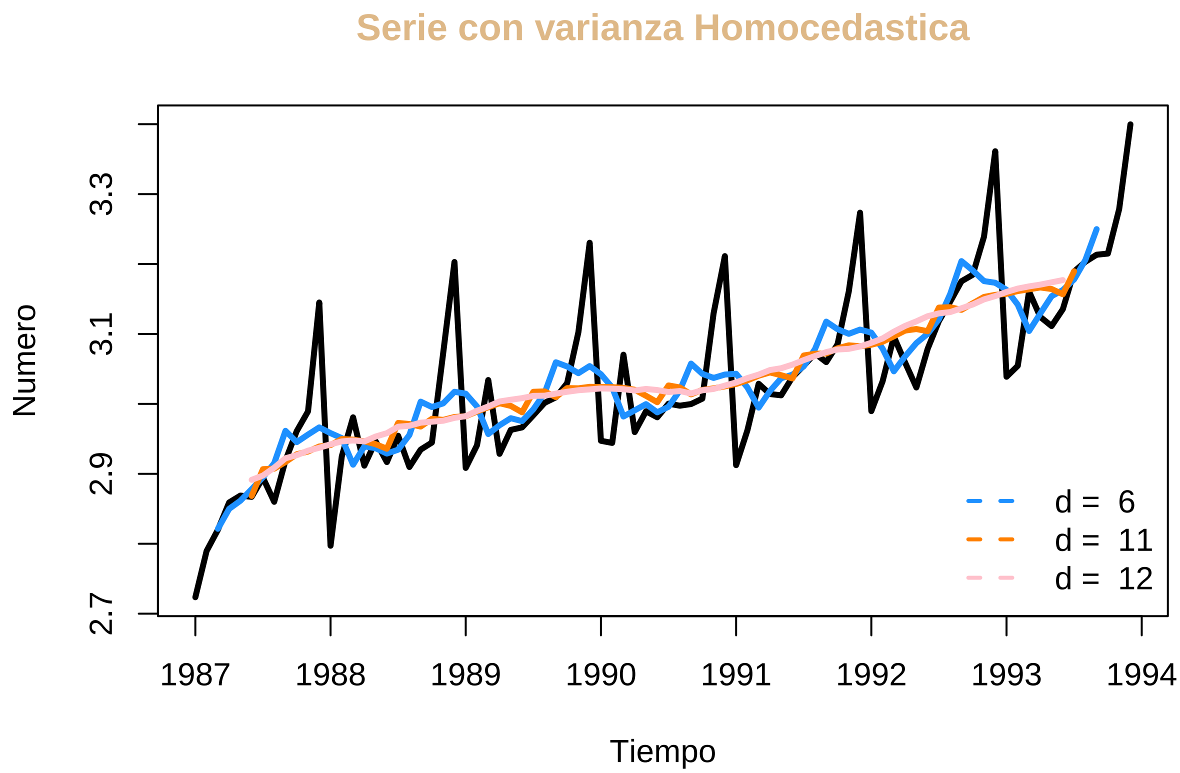
```r
# Quitamos los periodos mas grandes
d = d[-length(d)]
d = d[-length(d)]
# Quitamos los periodos mas chicos
d = d[-1]
d = d[-1]
d #Posibles periodos del ciclo
```

```
## [1]  6 11 12
```

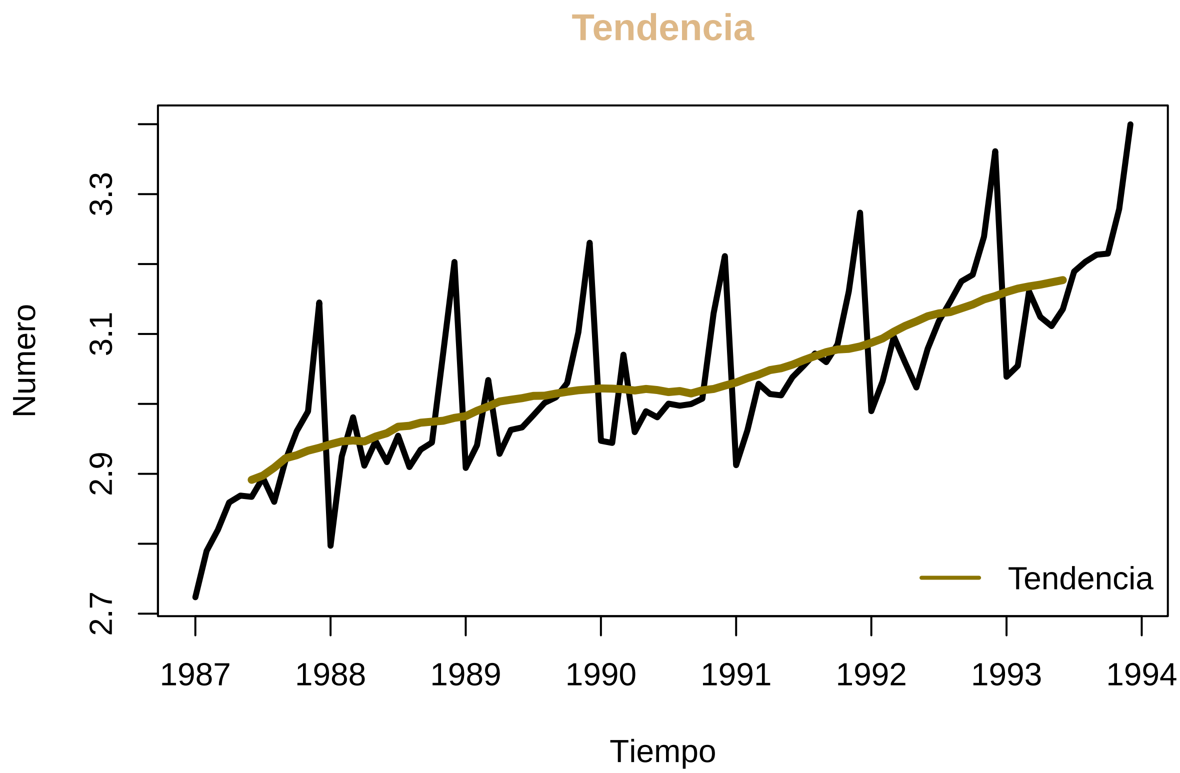
```r
#Realizamos la grafica:
col = c("dodgerblue1", "darkorange1", "pink")
plot(Serie_ln, lwd = 3, xlab = "Tiempo", col = "gray0",
     main = "Serie con varianza Homocedastica",
     ylab = "Numero", col.main = "burlywood")
library(dplyr)
```

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

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

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

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

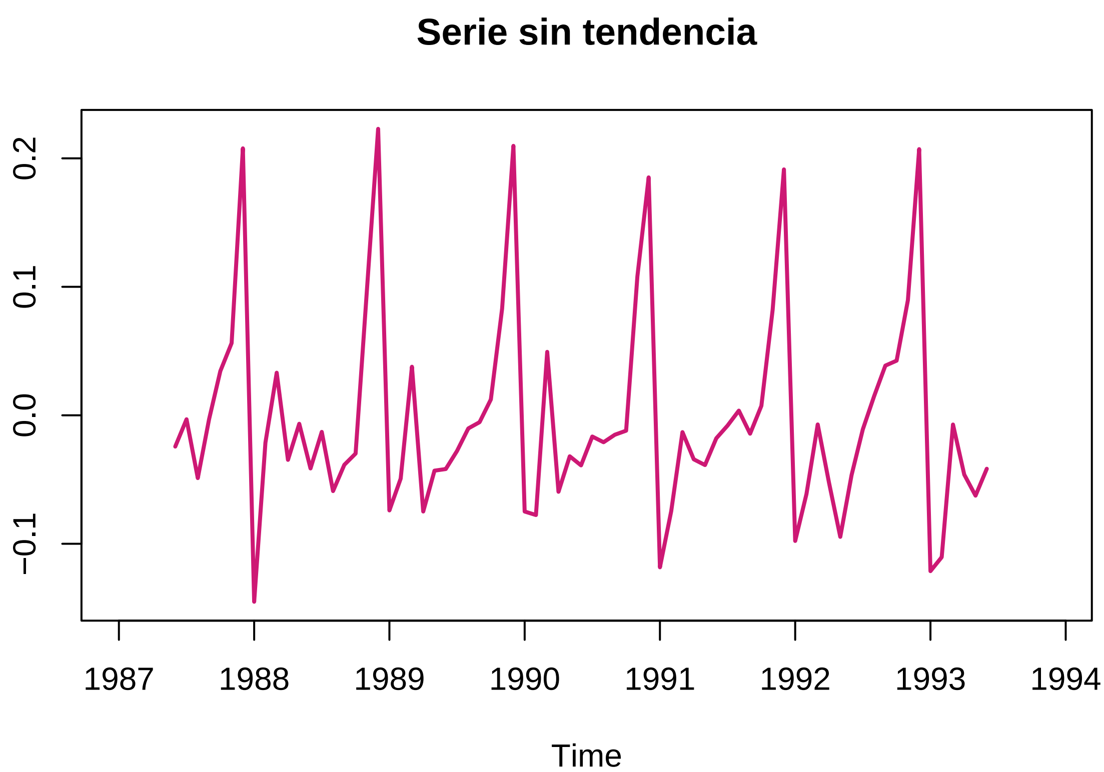
```r
t1 = seq(1987+0/12, 1993+11/12, by = 1 / 12)
for (i in 1:3) {
  lines(t1, stats::filter(Serie_ln, rep(1 / d[i], d[i])), col = col[i],
        lwd = 3)
}
legend("bottomright", col = col, lty = 2, lwd = 2, bty = "n",
       legend = c(paste("d = ", d[1]), paste("d = ", d[2]),
                  paste("d = ", d[3])), cex = 1)
```




```r
# Notemos que podemos aproximar la tendencia con d = 12, ya que esta nos
# muestra un mayor ajuste.

tendencia = stats::filter(Serie_ln, rep(1 / 12, 12))
plot(Serie_ln, lwd = 3, xlab = "Tiempo", col = "black",
     main = "Tendencia",
     ylab = "Numero", col.main = "burlywood")
lines(tendencia, col = "gold4", lwd = 4)
legend("bottomright", col = "gold4", lty = 1, lwd = 2, bty = "n",
       legend = "Tendencia", cex = 1)
```




```r
# Quitamos la tendencia
# Solo trabajamos con la serie cuya varianza es cte.

datosSinTendencia = Serie_ln - tendencia # Serie sin tendencia
plot(datosSinTendencia, main="Serie sin tendencia", lwd=2, ylab="", col=14)
```



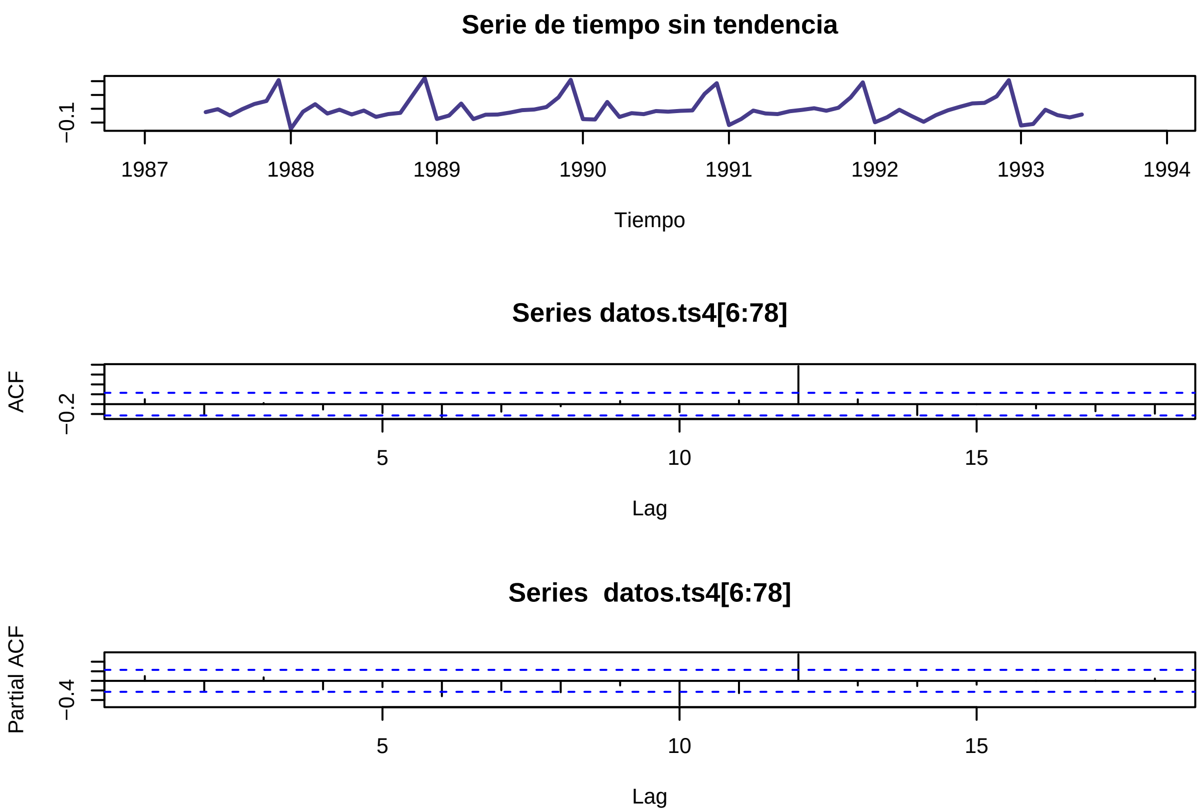

```
# Convertimos datosSinTendencia en objeto TS, dado que hicimos promedios moviles

start(Serie_ln)
```

```
## [1] 1987    1
```

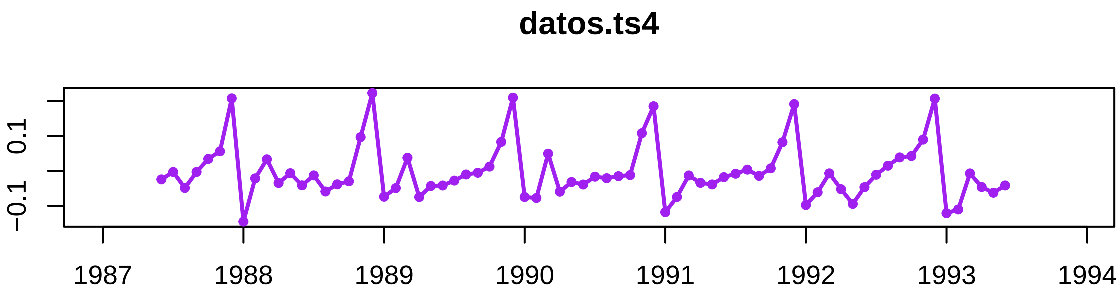
```
end(Serie_ln)
```

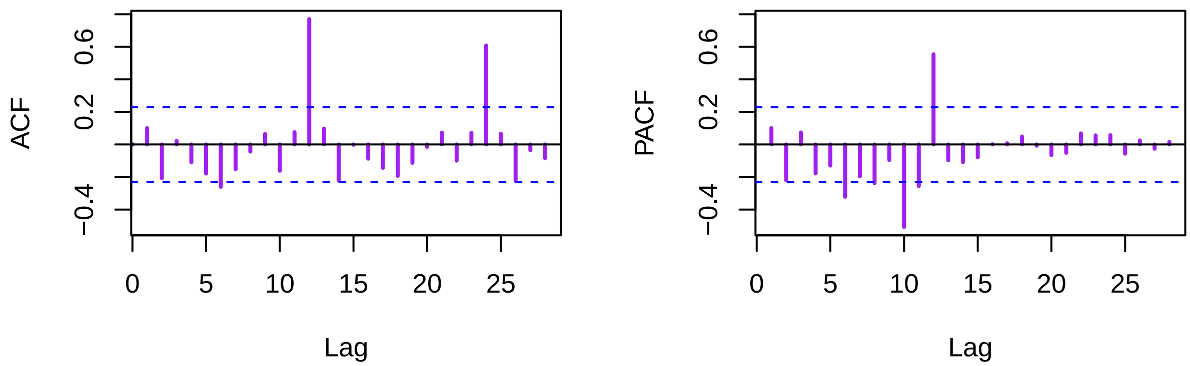
```
## [1] 1993   12
```

```
datos.ts4=ts(datosSinTendencia, frequency = 12, start=c(1987,01),end=c(1993,12))

View(datos.ts4)
par(mfrow = c(3,1))
plot(datos.ts4, col = "slateblue4", lwd = 2, ylab = " ", type = "l",
     main = "Serie de tiempo sin tendencia", xlab = "Tiempo")

acf(datos.ts4[6:78])
pacf(datos.ts4[6:78])
```



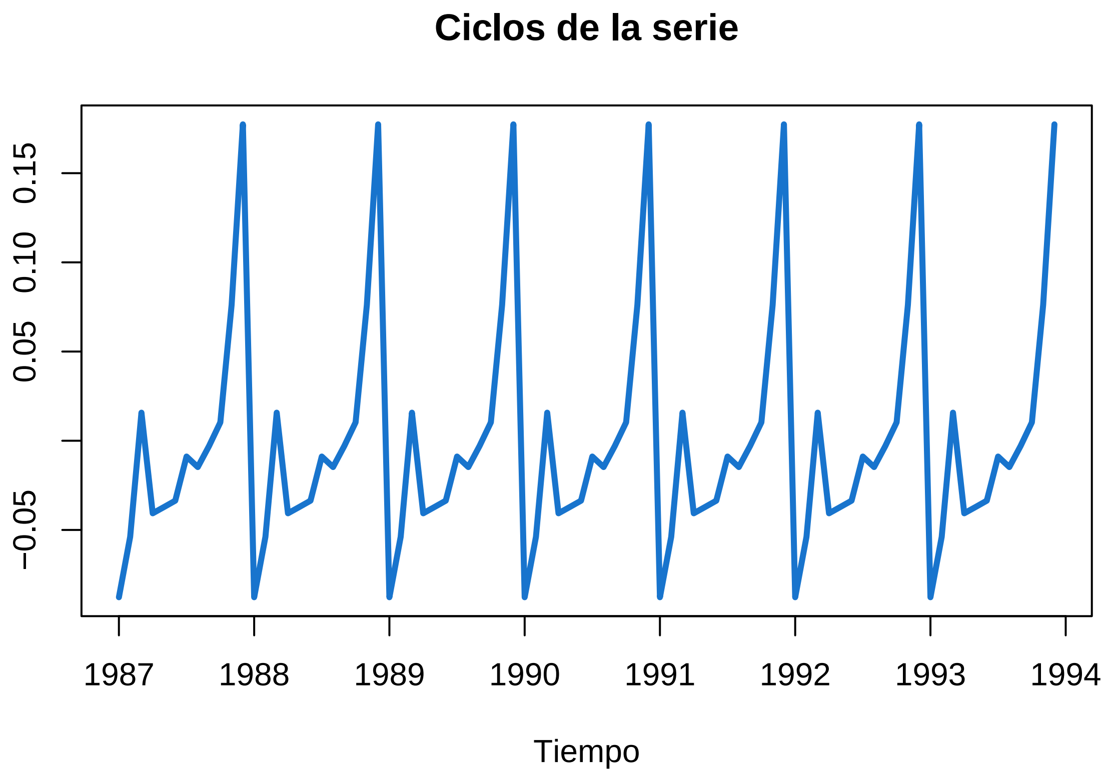

```
par(mfrow = c(1,1))
tsdisplay(datos.ts4, col="purple", lwd=2)
```



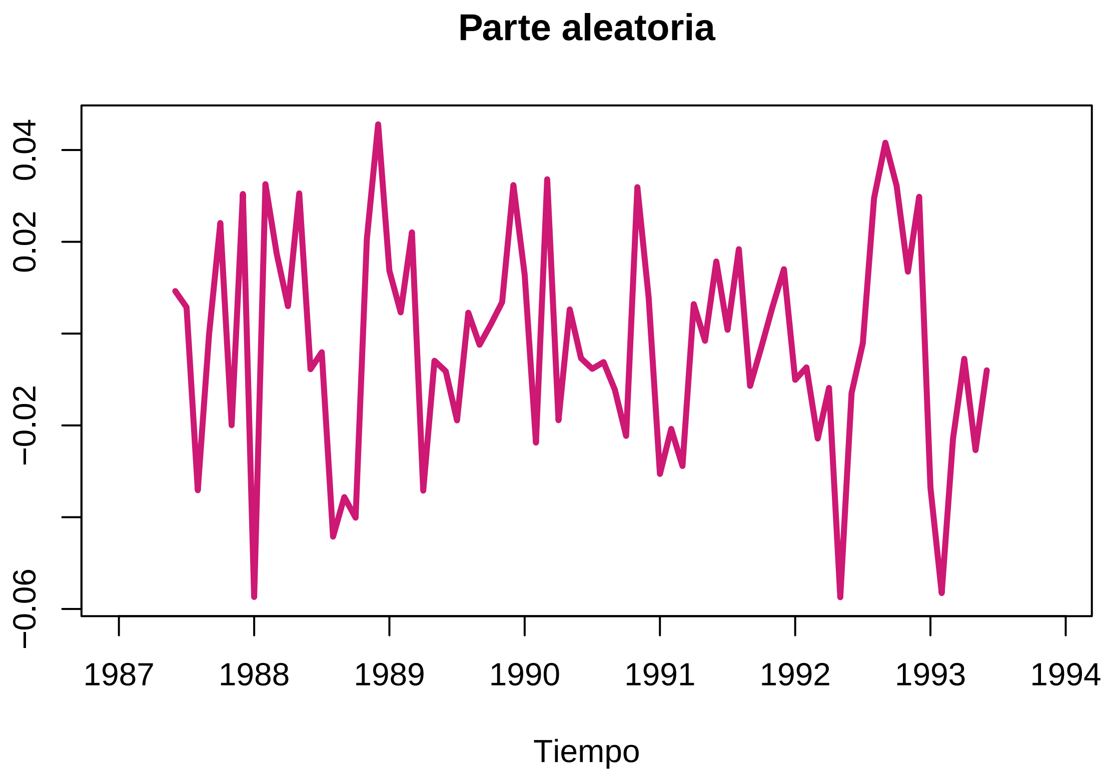



```
# Tenemos problemas de ciclos y muy marcados
```

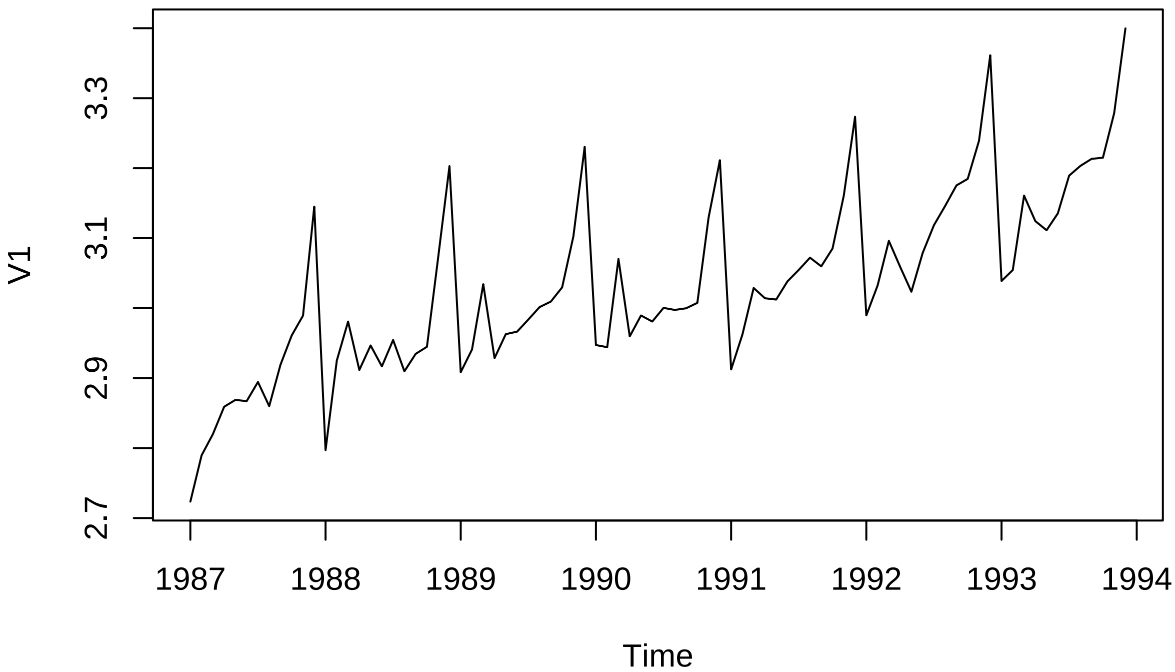
```
# Ahora, estimaremos la parte estacional. Tenemos que d = 12.
# n = length(Serie_ln) = 84, tenemos 72 (por los NA), then 72 / 12 = 6 ciclos.
# Creamos un ciclo promedio que estime la parte estacional,
# usando la serie sin tendencia.
d = 12
k = length(datos.ts4) / d # Numero de ciclos de la serie sin tendencia
w = rep(0, 12)
# Para el resto de los meses
for (i in 1:12)
  w[i] = sum(datos.ts4[d * (0:(k-1)) + i], na.rm = TRUE) / k

# Ahora, ajustamos el ciclo obtenido
ciclo  = w - mean(w)
ciclo = ts(rep(ciclo, times = k), start = start(Serie_ln),
           frequency = frequency(Serie_ln))
par(mfrow = c(1, 1))
plot(ciclo, col =20, lwd = 3, ylab = " ", xlab = "Tiempo",
     main = "Ciclos de la serie")# Es el ciclo de la serie
```



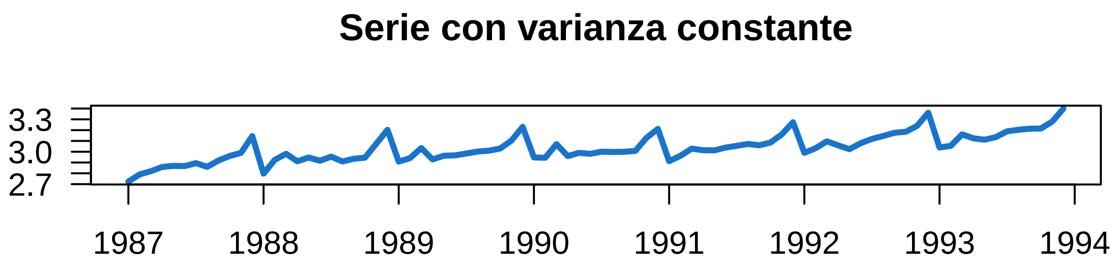

```
# Ciclos anuales

# Calculamos la parte aleatoria
parte_aleatoria = datos.ts4 - ciclo
plot(parte_aleatoria, main = "Parte aleatoria",
     col =30, lwd = 3, xlab = "Tiempo", ylab = "")
```

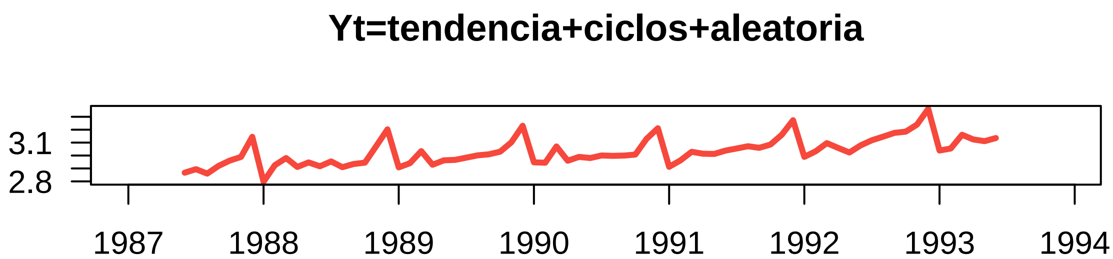



```
plot(Serie_ln)
```

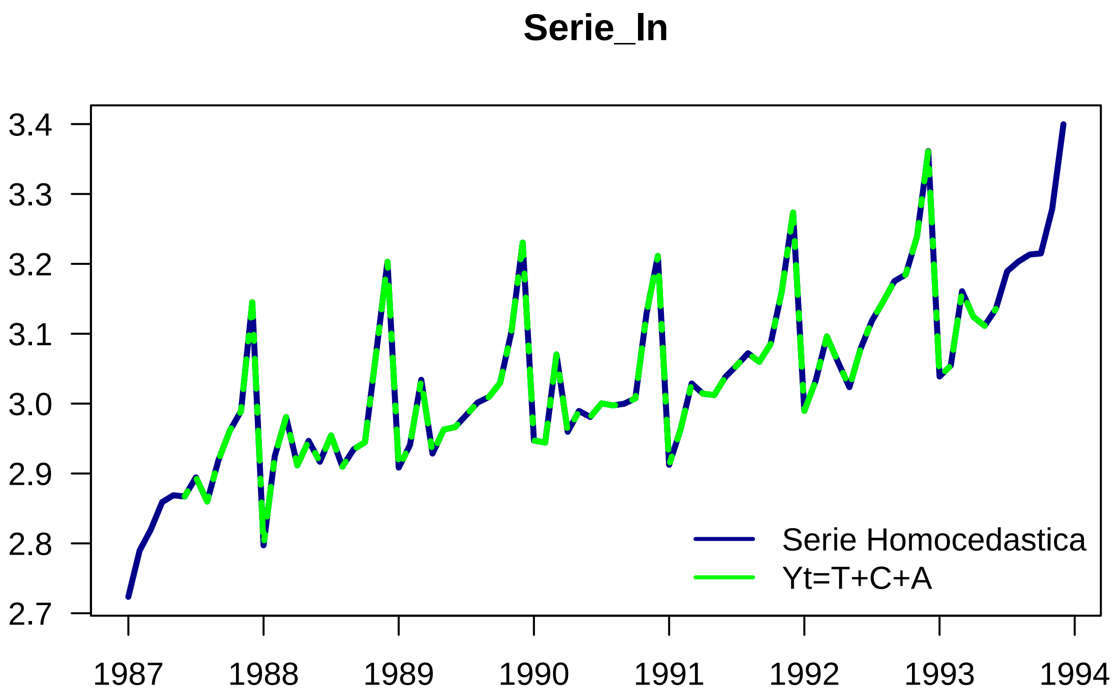



```r
# Con esto, ya tenemos nuestras series

componentes = tendencia + ciclo+parte_aleatoria
componentes = ts(componentes, start = start(Serie_ln), frequency = 12)
par(mfrow = c(2,1))
plot(Serie_ln, col=28, las=1, main="Serie con varianza constante", lwd=3, xlab="",ylab="")
plot(componentes, col = 18, lwd = 3, las=1, main="Yt=tendencia+ciclos+aleatoria", xlab="",ylab="")
```






```
par(mfrow = c(1,1))
plot(Serie_ln,col="darkblue", las=1, lwd=3,main="Serie_ln", ylab="",xlab="")
invisible(lines(componentes, type="l", lwd=3, col="green",lty=6))
legend("bottomright", col = c("darkblue","green"), lty = 1, lwd = 2, bty = "n",
       legend = c("Serie Homocedastica","Yt=T+C+A"), cex = 1)
```

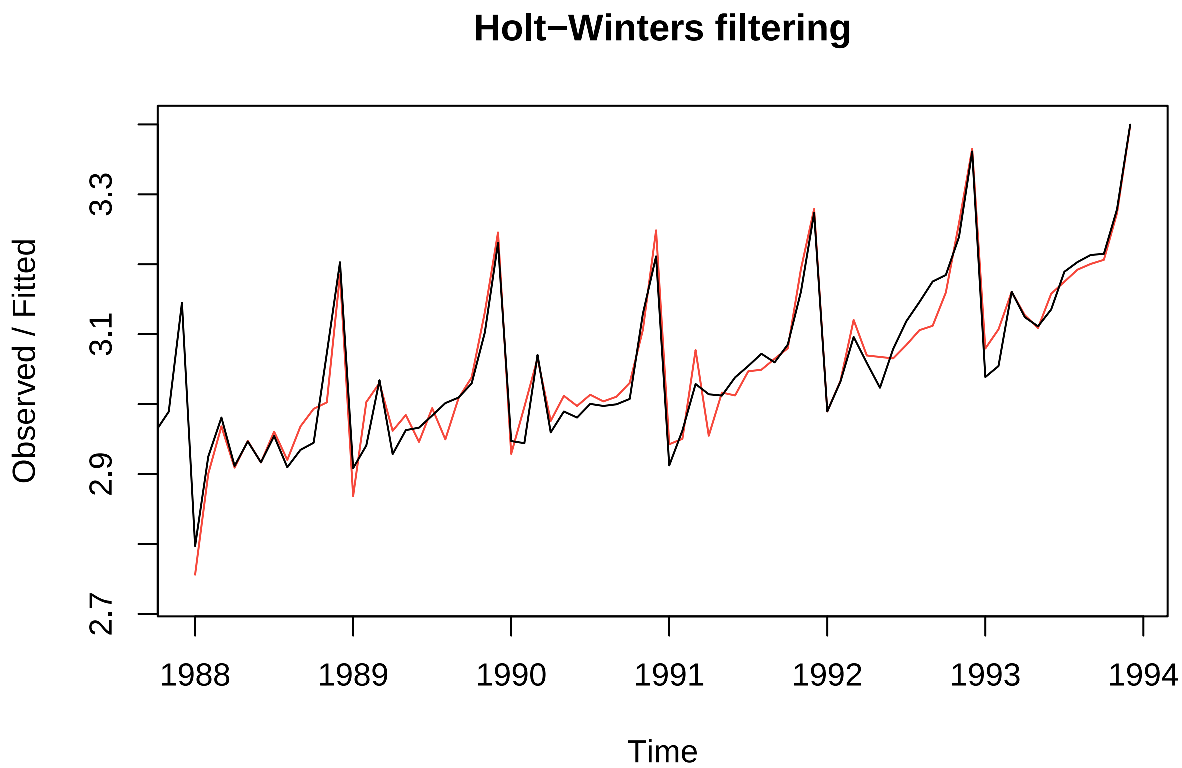


Veamos que ya la logramos descomponer.

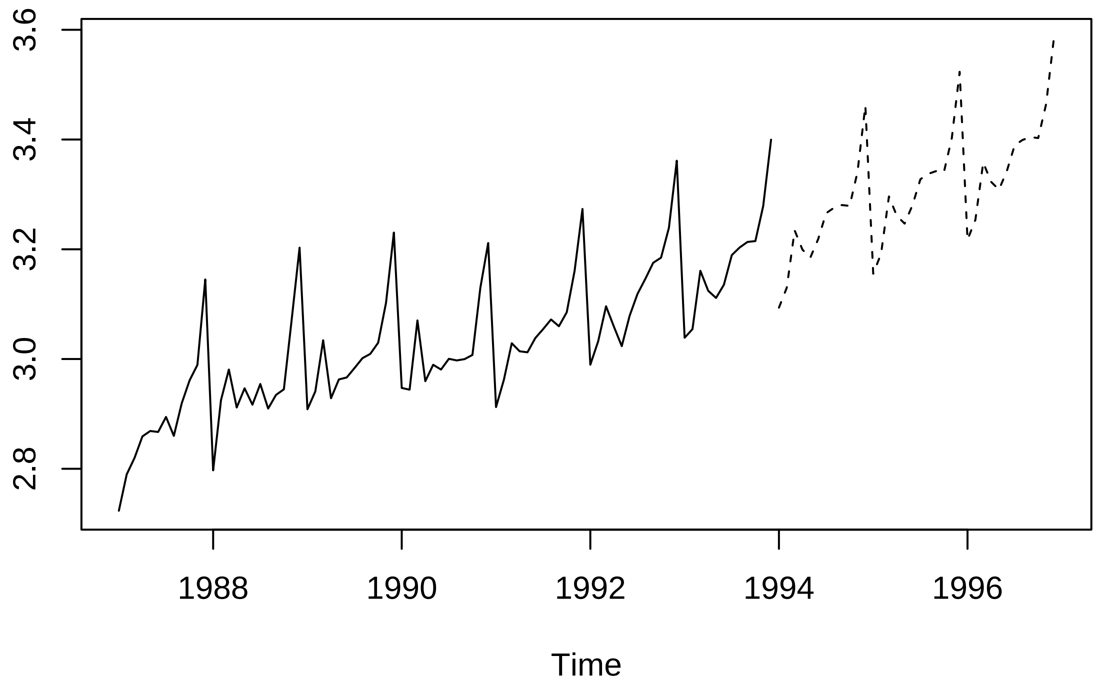
Suavizamiento exponencial

Para esta parte usaremos el método de Holt-Winter que pertenece a los métodos de suavizamiento exponencial. Usamos aditivo, debido que usamos el logaritmo.

```
xt.hw = HoltWinters(Serie_ln, seasonal="additive")
plot(xt.hw)
```



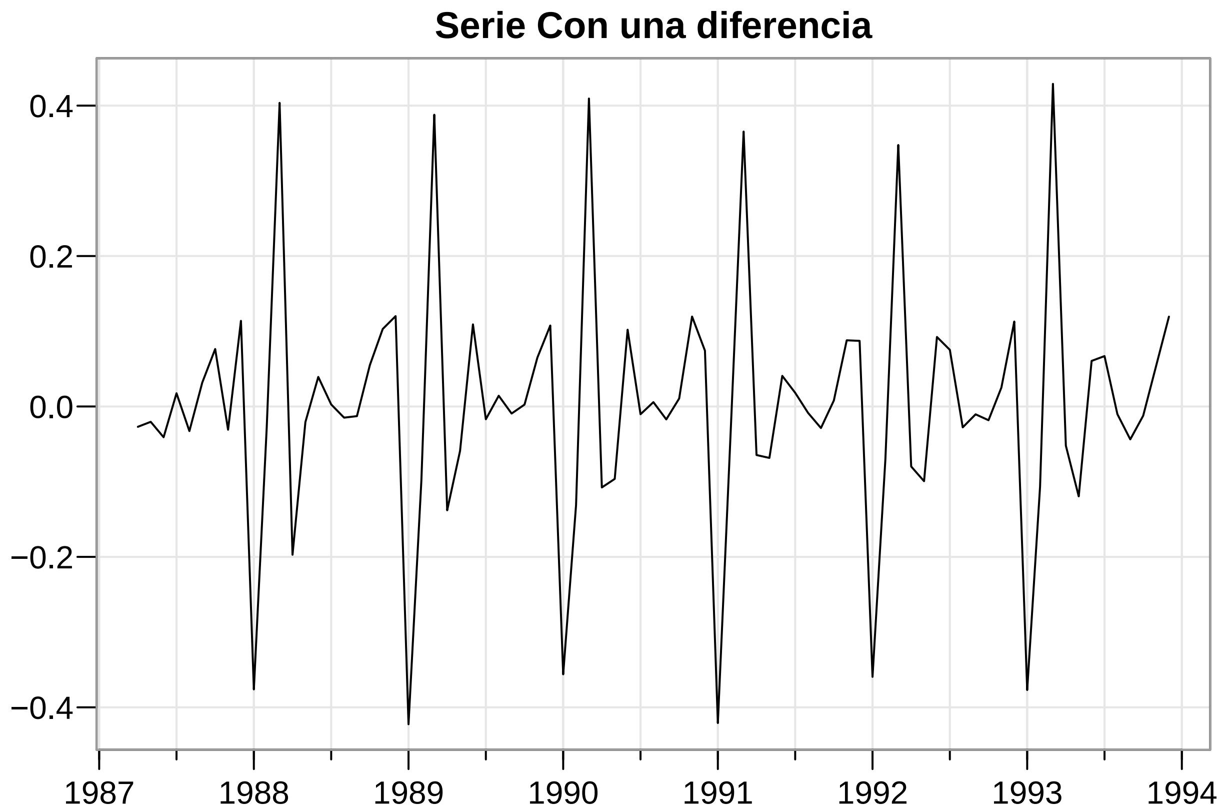

```
xt.predict = predict(xt.hw, n.ahead=3*12)
ts.plot(Serie_ln, xt.predict, lty=1:3)
```



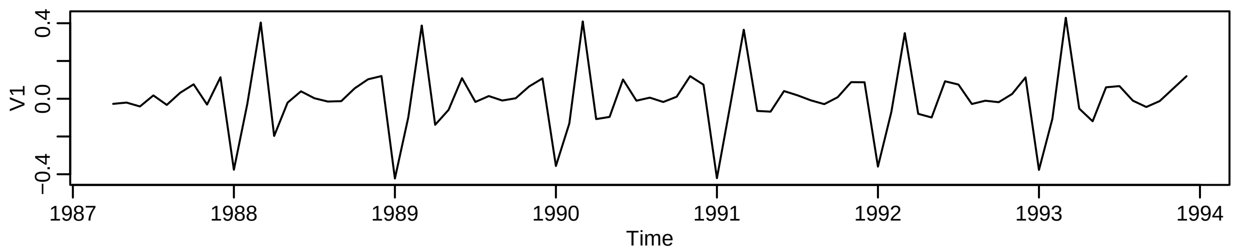
(c) Diferencias.

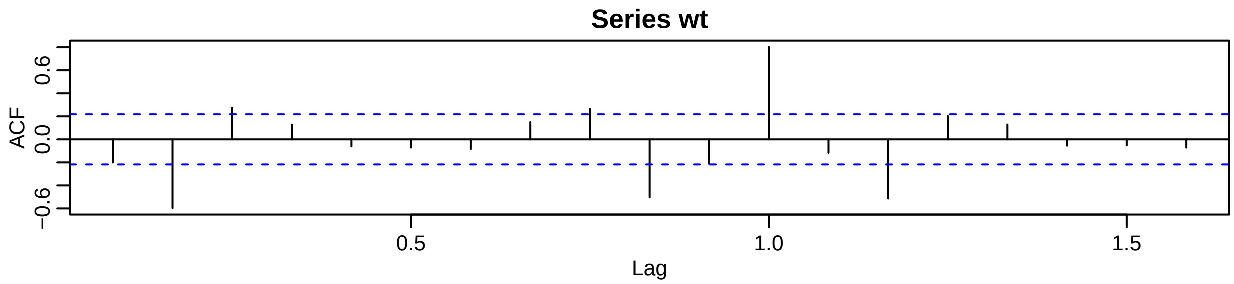
```
yt= Serie_ln
tsplot(Serie_ln, main="Serie original", ylab="", xlab="", las=1)
```

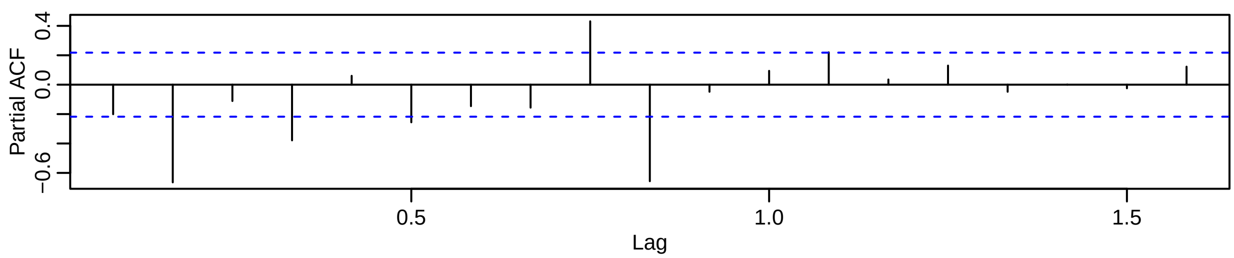

```
wt= diff(diff(yt),2)
tsplot(wt, main="Serie Con una diferencia", ylab="", xlab="", las=1)
```




```
par(mfrow=c(3,1))
plot(wt)
acf(wt)
pacf(wt)
```








```
kpss.test(wt)
```

```
## Warning in kpss.test(wt): p-value greater than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  wt
## KPSS Level = 0.059987, Truncation lag parameter = 3, p-value = 0.1
```

```
adf.test(wt)
```

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

##
##  Augmented Dickey-Fuller Test
##
## data:  wt
## Dickey-Fuller = -5.902, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Hacemos la diferencia con lag igual a 2, ya que en el inciso b) el periodo era de 2. 4.- Describa brevemente en qué consisten los métodos de suavizado exponencial (exponential smoothing) para las series de tiempo y el método de Holt Winters.

En las notas del curso se nos describe de manera breve y concisa cómo es que funcionan:

La selección del método se basa generalmente en el reconocimiento de la tendencia y estacionalidad, así como en la forma en que estos entran en el método de suavizamiento, como aditiva o multiplicativa. Generalmente se usa el promedio para pronosticar si todos los pronósticos futuros son iguales a un promedio simple de los datos obervados, puede ser sensato asignar mayor peso a las obser vaciones más recientes que a las del pasado más distante. En palabras más simples podemos definir lo de la siguiente manera: “Son básicamente promedios ponderados de observaciones pasadas, con los pesos decayendo exponencialmente a medida que las observaciones”envejecen“(…)”.

Por otro lado, el método de Holt Winters habla de la forma del componente para el método aditivo y el método multiplicativo.

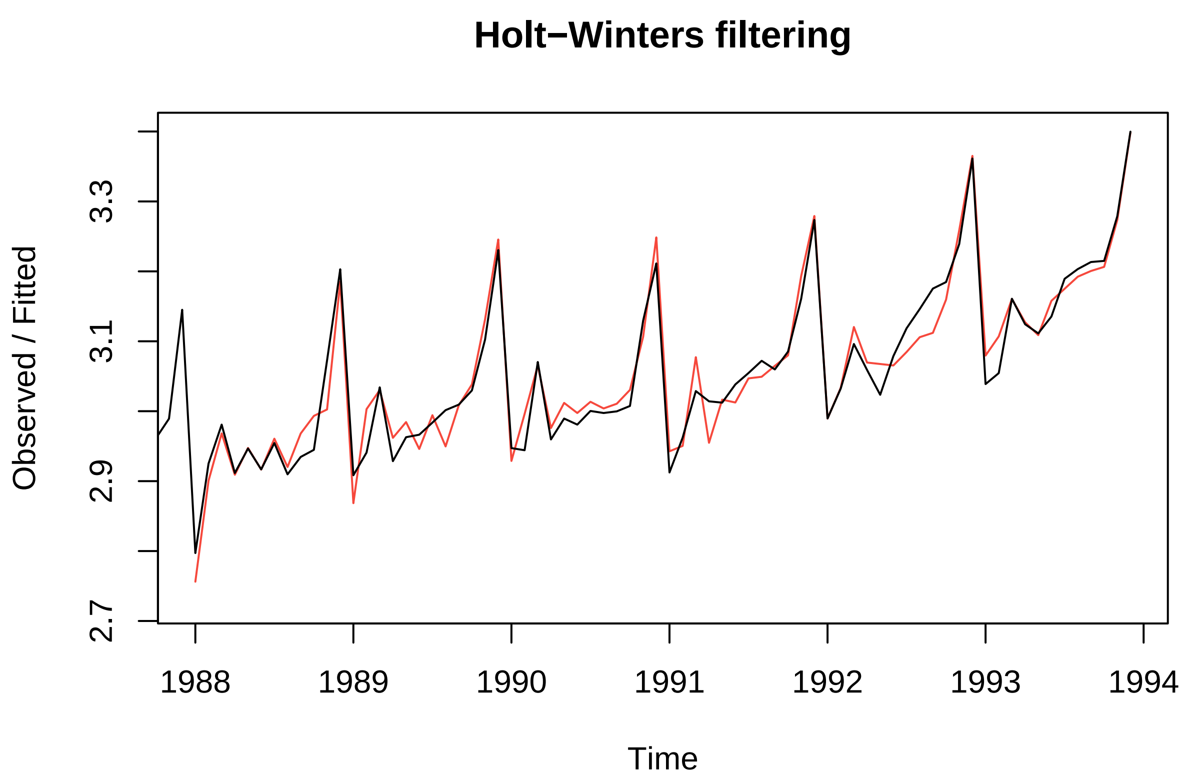
Más especificamente, el método de Holt Winters amplía el suavizado simple exponencial para permitir además el pronóstico de datos con tendencia y capturar la estacionalidad, además la ecuación estacional muestra un promedio ponderado entre el índice estacional actual y el índice estacional pero un año atrás.

El método multiplicativo es similar al aditivo. El método de multiplicativo de Holt-Winters también calcula valores suavizados simple exponencialmente para el nivel, tendencia y ajuste estacional para la previsión. Este método multiplica la previsión con tendencia por la estacionalidad, lo que produce la previsión de multiplicativo de Holt-Winters.

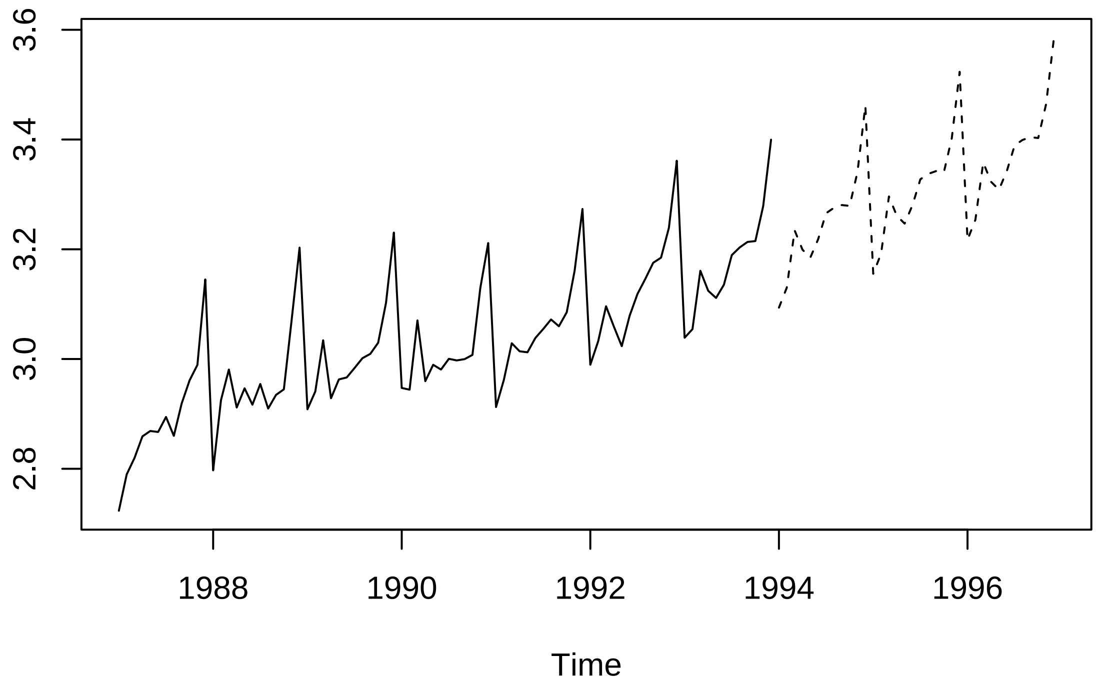
5.- Use el método de Holt Winters para el ajuste de la curva y predicción de los datos de 3 años futuros.

Esto se hizo en el 3b), con el siguiente código:

```
xt.hw = HoltWinters(Serie_ln, seasonal="additive")
plot(xt.hw)
```




```
xt.predict = predict(xt.hw, n.ahead=3*12)
ts.plot(Serie_ln, xt.predict, lty=1:3)
```


3.6
3.4
3.2
3.0
2.8
1988
1990
1992
1994
1996
Time