Tema 02.01: Análisis clásicos de series temporales con R

@umh1465: Análisis estadístico de series económicas

Xavi Barber

Centro de Investigación Operativa
Universidad Miguel Hernández de Elche

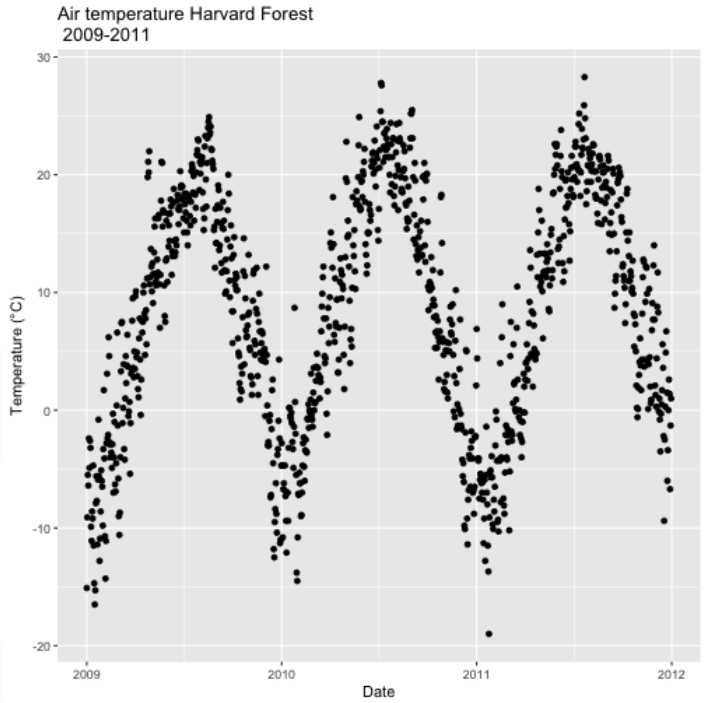
2018-02-22

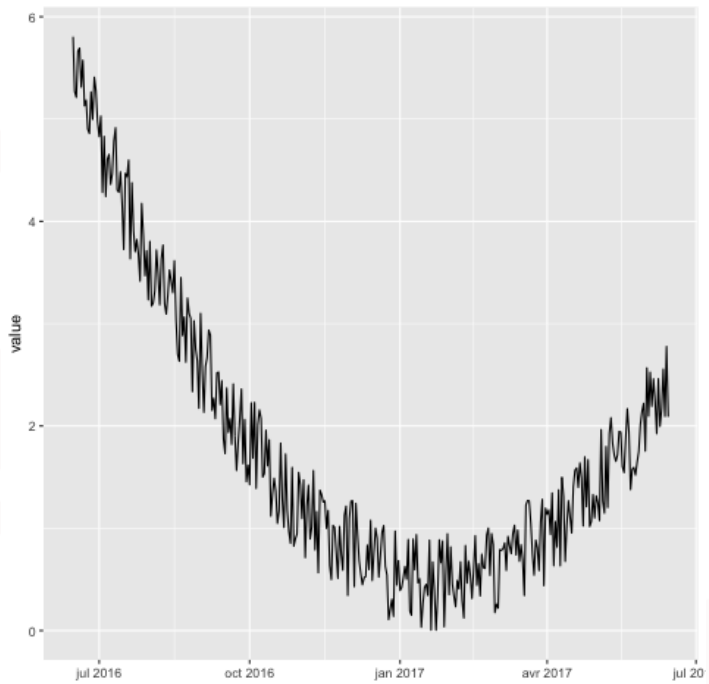
¿QUÉ ES UNA SERIE TEMPORAL?

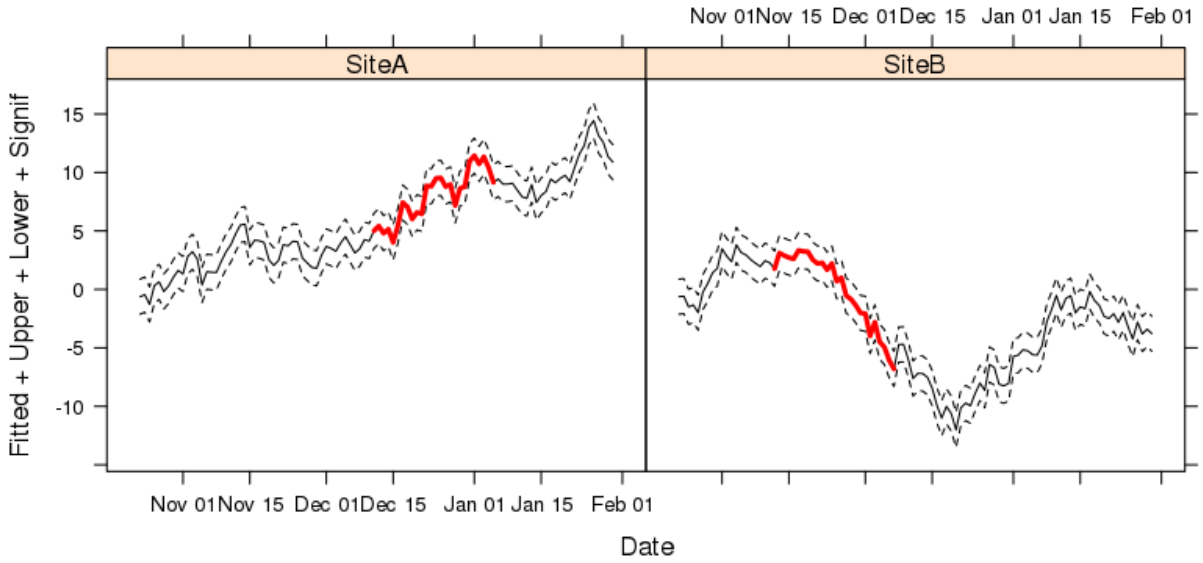
Objetivo del análisis de series temporales

Con el análisis de series temporales se pretende extraer las regularidades que se observan en el comportamiento pasado de la variable, es decir, obtener el mecanismo que la genera, para tener un mejor conocimiento de la misma en el tiempo. Además, bajo el supuesto de que las condiciones estructurales que conforman la serie objeto de estudio permanecen constantes, también se trata de predecir el comportamiento futuro.

- Explicar la evolución de una variable a lo largo del tiempo
- Prever sus valores futuros


Air temperature Harvard Forest
2009-2011
Temperature (°C)
30
20
10
0
-10
-20
2009
2010
2011
2012
Date

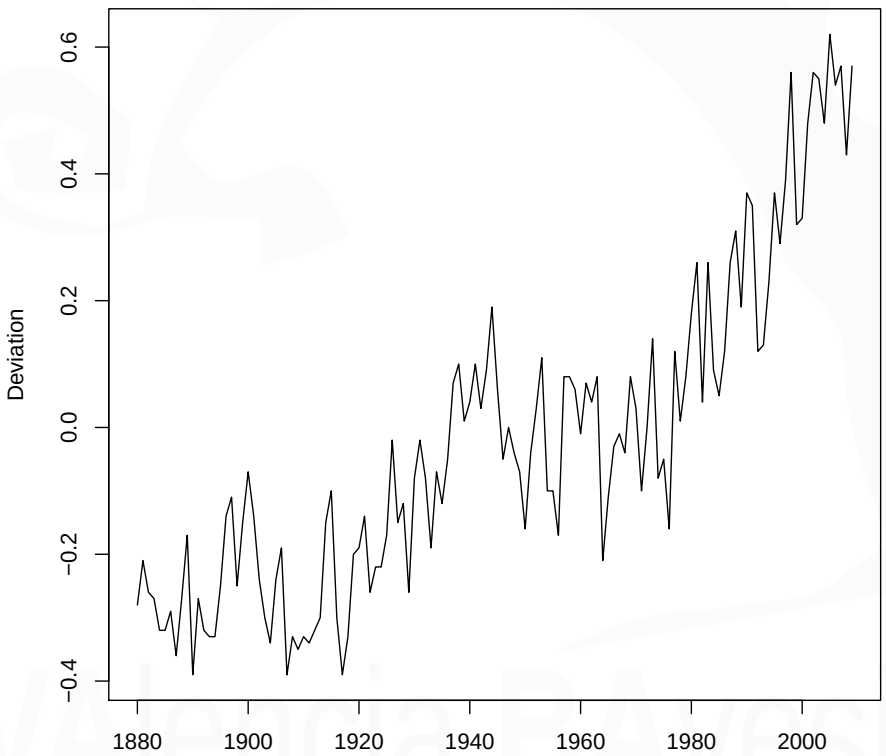

value
6
4
2
0
jul 2016
oct 2016
jan 2017
avr 2017
jul 20


Nov 01Nov 15 Dec 01Dec 15 Jan 01 Jan 15 Feb 01
SiteA
SiteB
Fitted + Upper + Lower + Signif
15
10
5
0
-5
-10
Nov 01Nov 15 Dec 01Dec 15 Jan 01 Jan 15 Feb 01
Date

ANÁLISIS PRELIMINAR DE UNA SERIE

Tendencia

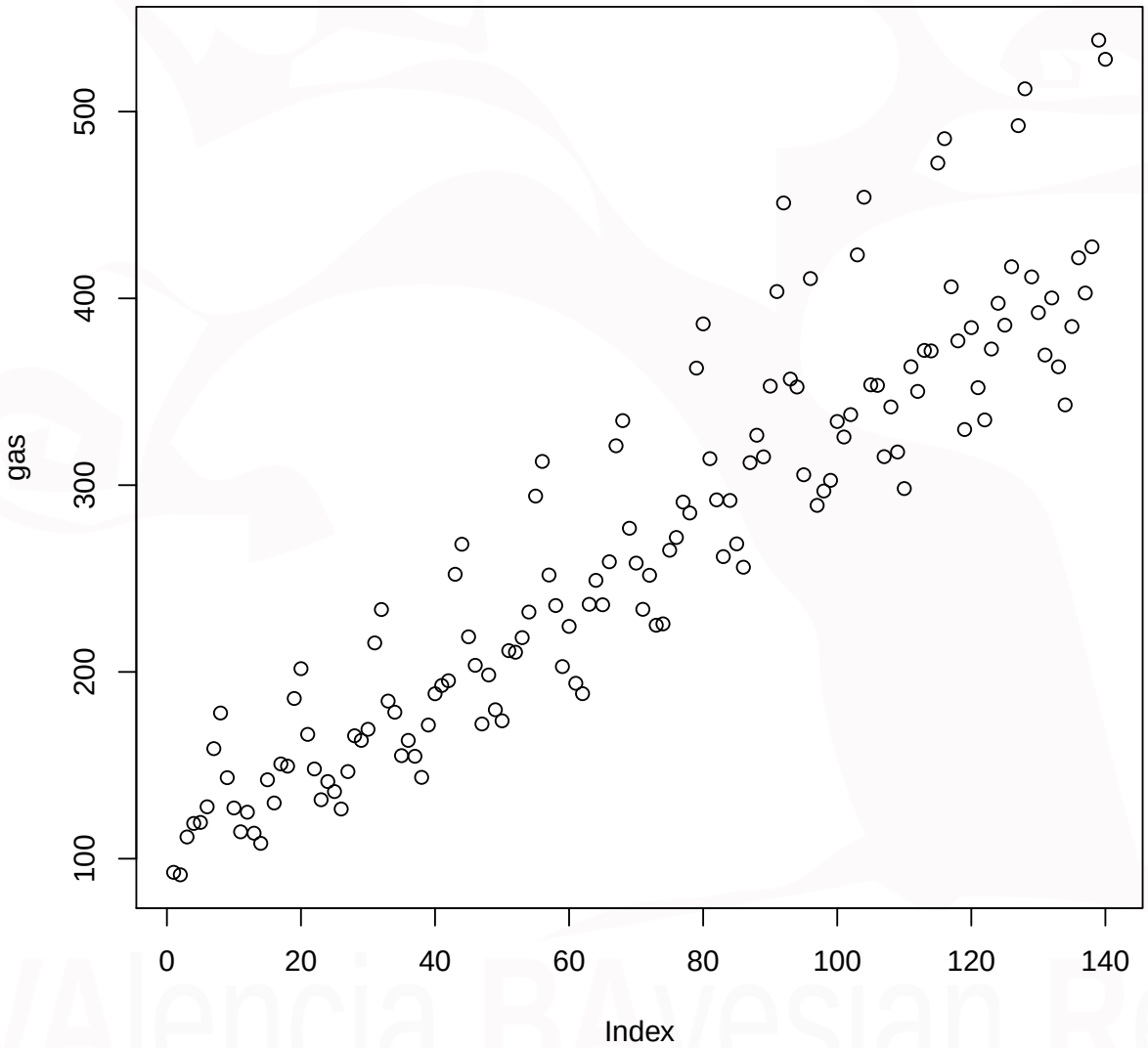
La forma más sencilla de comenzar el análisis de una serie temporal es me-diante su representación gráfica. El gráfico que se emplea para representar las series temporales es el gráfico de secuencia. Los gráficos de secuencia son diagramas de líneas en los cuales el tiempo se representa en el eje de abscisas (x), y la variable cuya evolución en el tiempo estudiamos en el eje de ordenadas (y).



Tendencia y nivel de la serie

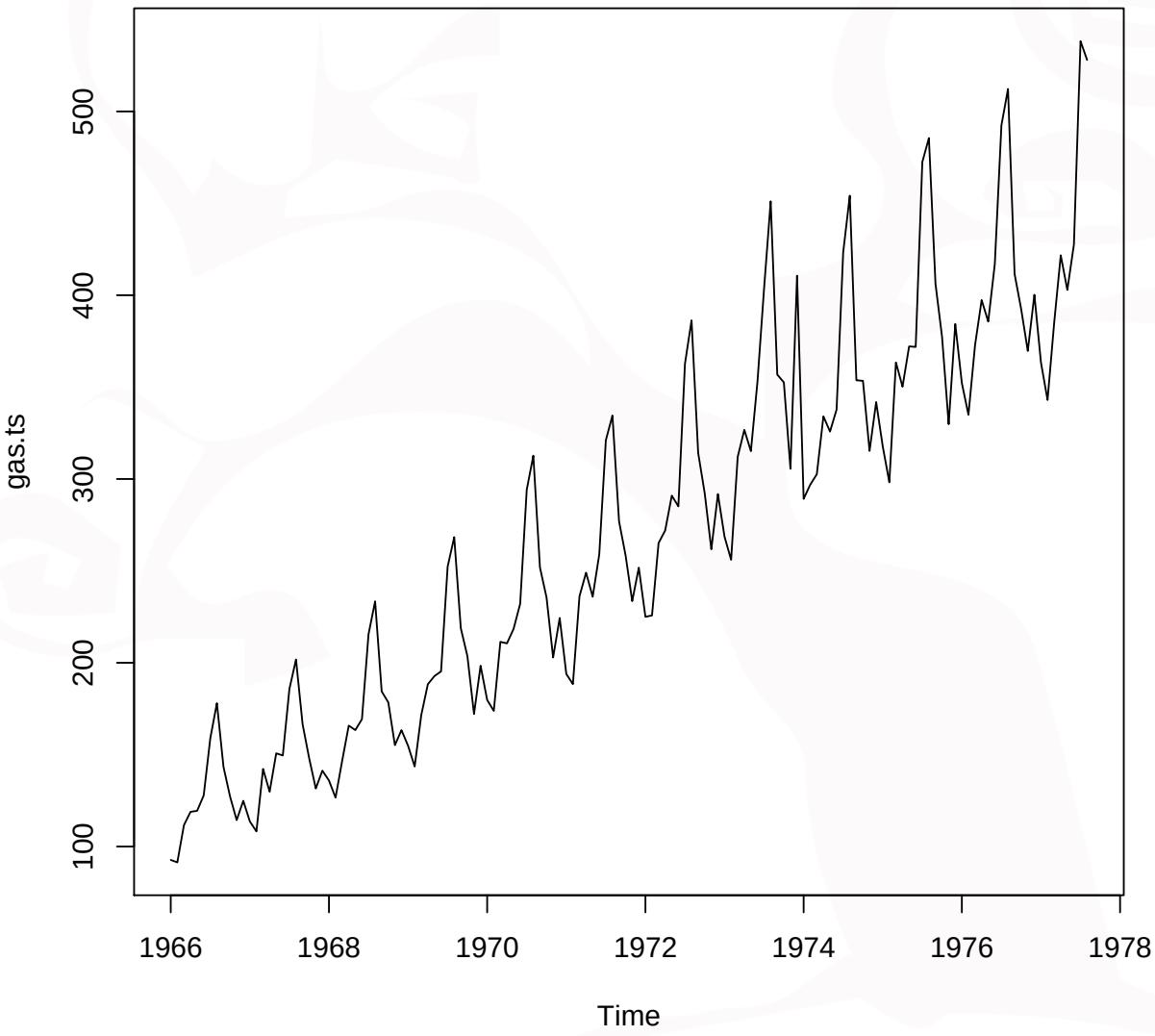
El nivel de una serie es una medida local de tendencia central, como por ejemplo la media, de cada periodo de tiempo que consideremos. Cuando trabaja-mos con un calendario (tiempo representado en días, meses o años), no es recomendable establecer periodos de tiempo antinaturales para estudiar esta característica.

```
gas = scan("http://verso.mat.uam.es/~joser.berrendero/datos/gas6677.dat")
plot(gas)
```

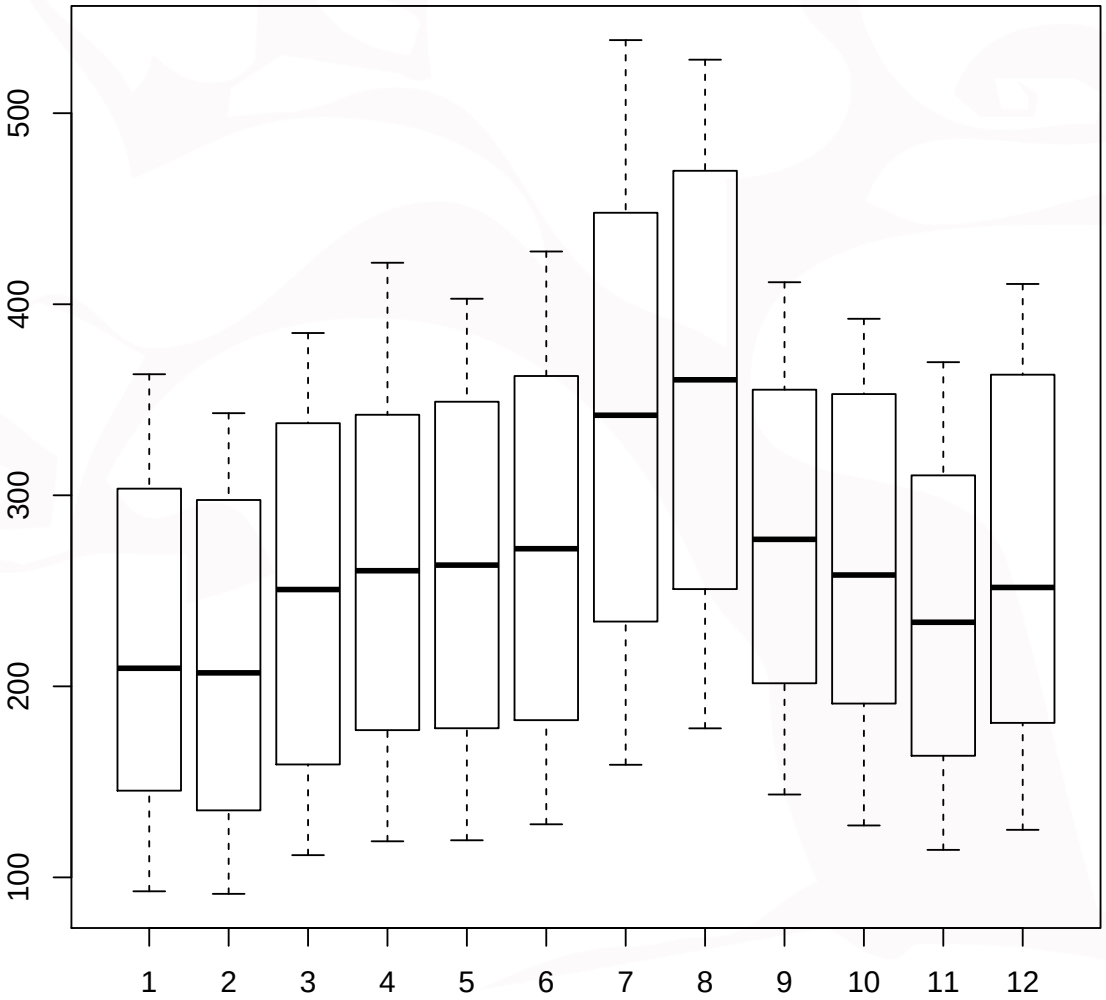


```
gas.ts = ts(gas, start = c(1966, 1), frequency = 12)
```

```
##          Jan     Feb     Mar     Apr     May     Jun     Jul     Aug
## 1966  92.718  91.380 111.643 118.888 119.432 127.796 158.943 178.013
## 1967 113.661 108.224 142.256 129.835 150.735 149.554 185.792 201.758
## 1968 135.951 126.615 146.647 165.822 163.365 169.294 215.538 233.427
## 1969 154.844 143.552 171.573 188.322 192.756 195.296 252.288 268.379
## 1970 179.759 173.821 211.387 210.551 218.371 232.057 294.173 312.700
## 1971 193.916 188.375 236.187 249.037 235.957 258.980 321.085 334.562
## 1972 225.010 225.742 265.159 271.986 290.953 285.108 362.687 386.347
## 1973 268.578 256.063 312.041 326.741 315.157 353.016 403.662 451.098
## 1974 289.186 296.881 302.589 334.091 325.790 337.782 423.297 454.172
## 1975 317.760 298.188 363.429 350.203 372.149 371.877 472.458 485.517
## 1976 352.200 334.938 372.891 397.388 385.657 416.961 492.480 512.209
## 1977 363.367 342.979 384.936 421.718 402.877 427.615 538.254 528.007
##          Sep     Oct     Nov     Dec
## 1966 143.385 127.179 114.403 124.900
## 1967 166.565 148.048 131.581 141.315
## 1968 184.402 178.432 155.179 163.355
## 1969 218.810 203.545 172.148 198.381
## 1970 251.891 235.560 202.876 224.383
## 1971 276.932 258.269 233.532 251.755
## 1972 314.205 292.124 261.740 291.810
## 1973 356.811 352.566 305.580 410.614
## 1974 353.727 353.413 315.272 341.902
## 1975 406.223 377.262 329.794 384.350
## 1976 411.514 392.380 369.671 400.243
## 1977
```


gas.ts
100
200
300
400
500
1966
1968
1970
1972
1974
1976
1978
Time

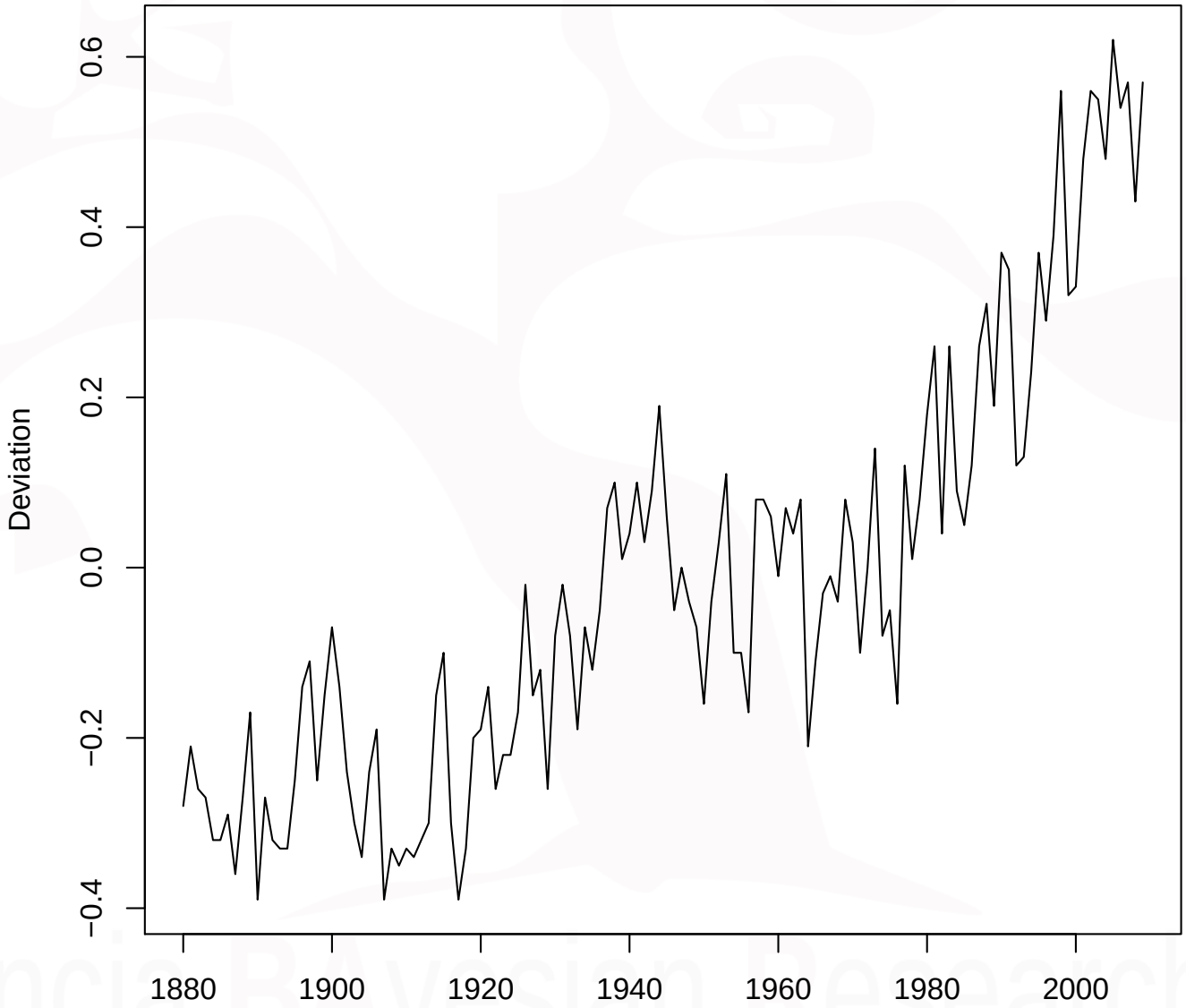
```
boxplot(gas.ts ~ cycle(gas.ts))
```

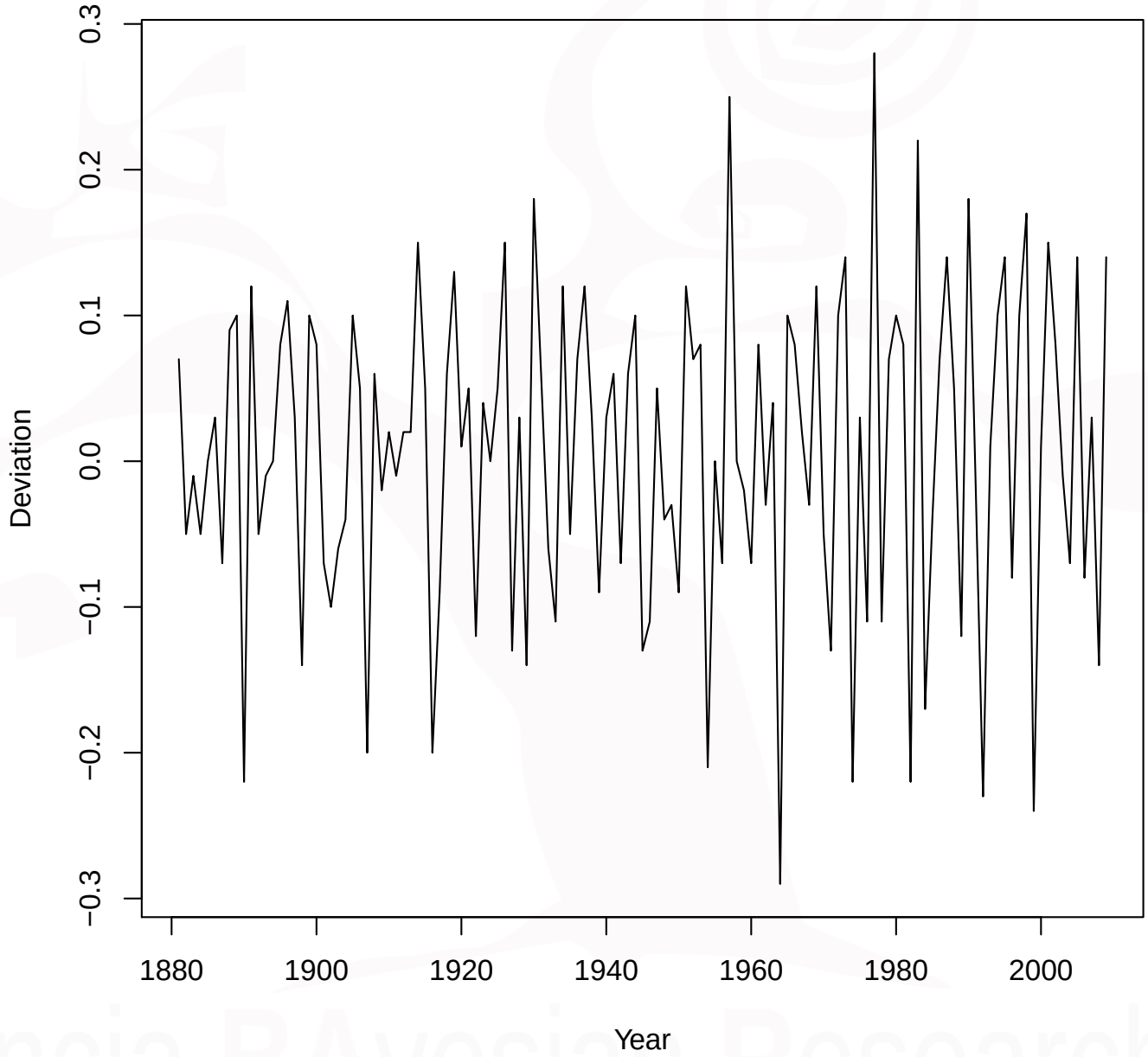


Estacionariedad

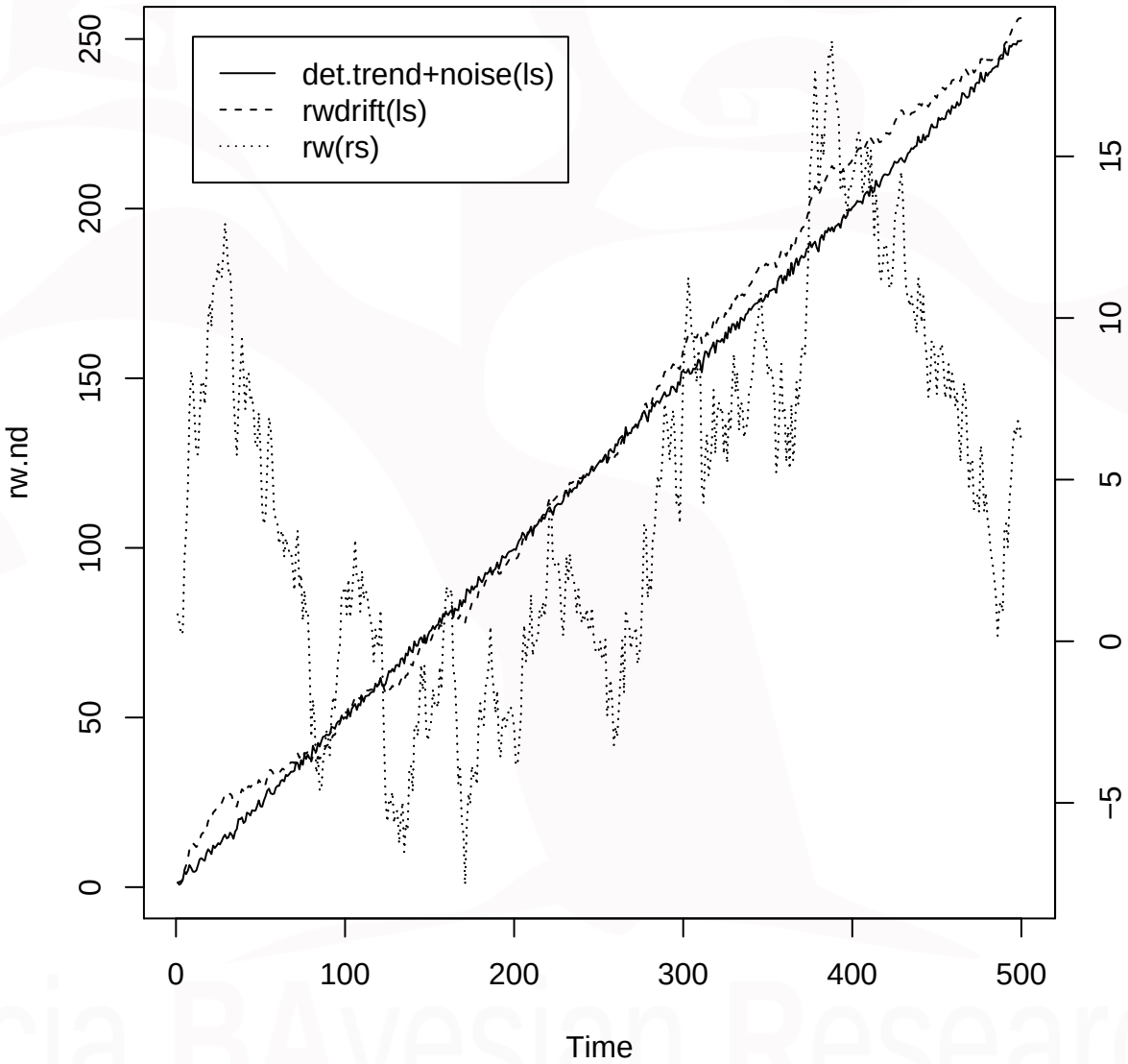
- **Series Estacionaria**: Estacionarias en la media y la varianza (frecuentes en el mundo físico, pero no en el social o económico).
  - Una serie es estacional cuando podemos observar en ella un patrón sistemático que se repite periódicamente (cada año, cada mes, etc., dependiendo de las unidades de tiempo en que vengan recogidos los datos).
- **Serie no-estacionaria**: Su variabilidad y/o su media cambian en el tiempo.
  - El cambio en la varianza implica que la dispersión (variabilidad) no es constante en el tiempo.
  - El cambio en la media implica tendencia (a crecer o decrecer), la serie no oscila alrededor de un valor constante. Fenómenos sociales.

¿Cómo detectamos si una serie es o no estacionaria en varianza?




Deviation
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
1880
1900
1920
1940
1960
1980
2000
Year

¿Cómo detectamos si una serie es o no estacionaria en media?

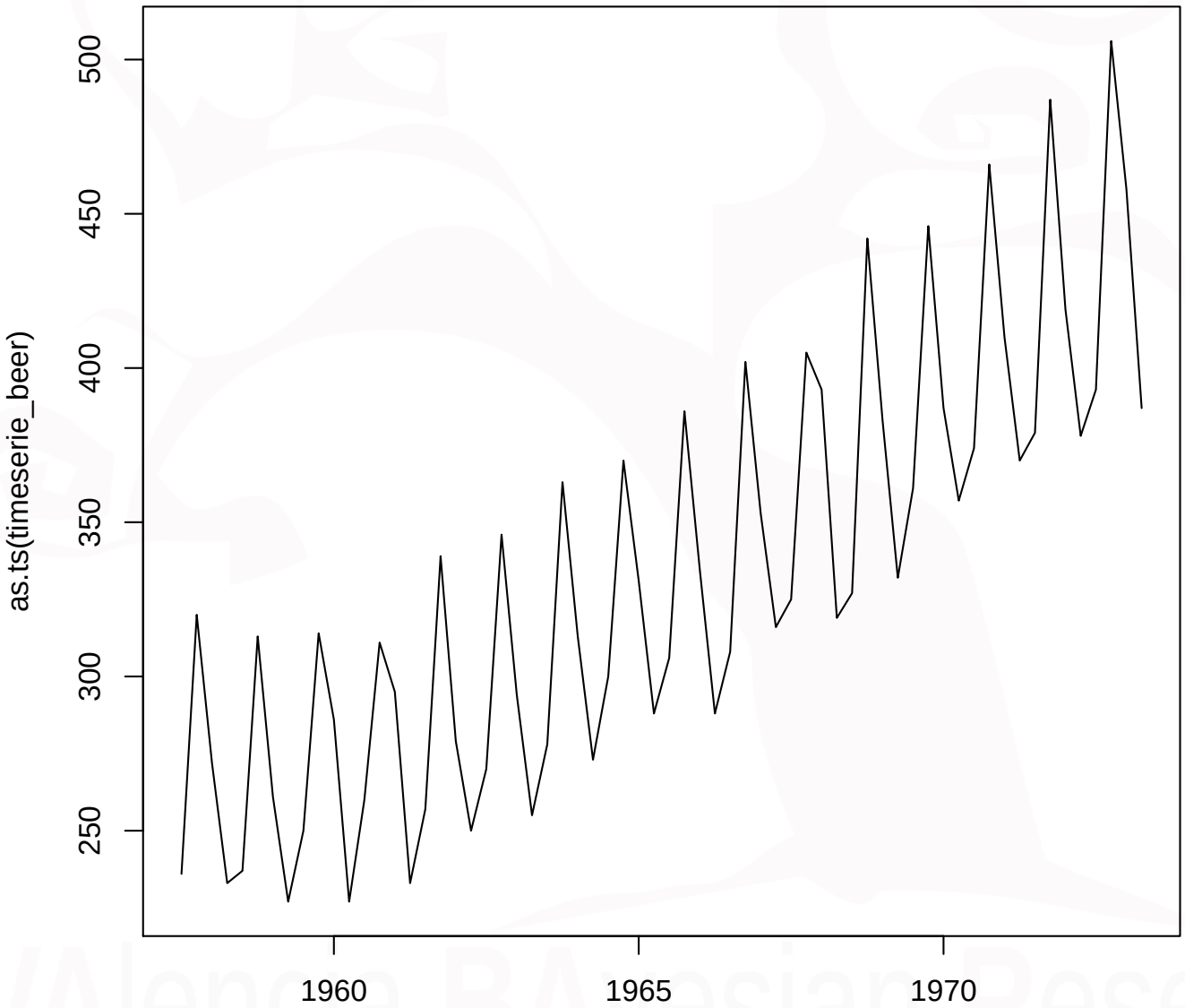


Estacionalidad

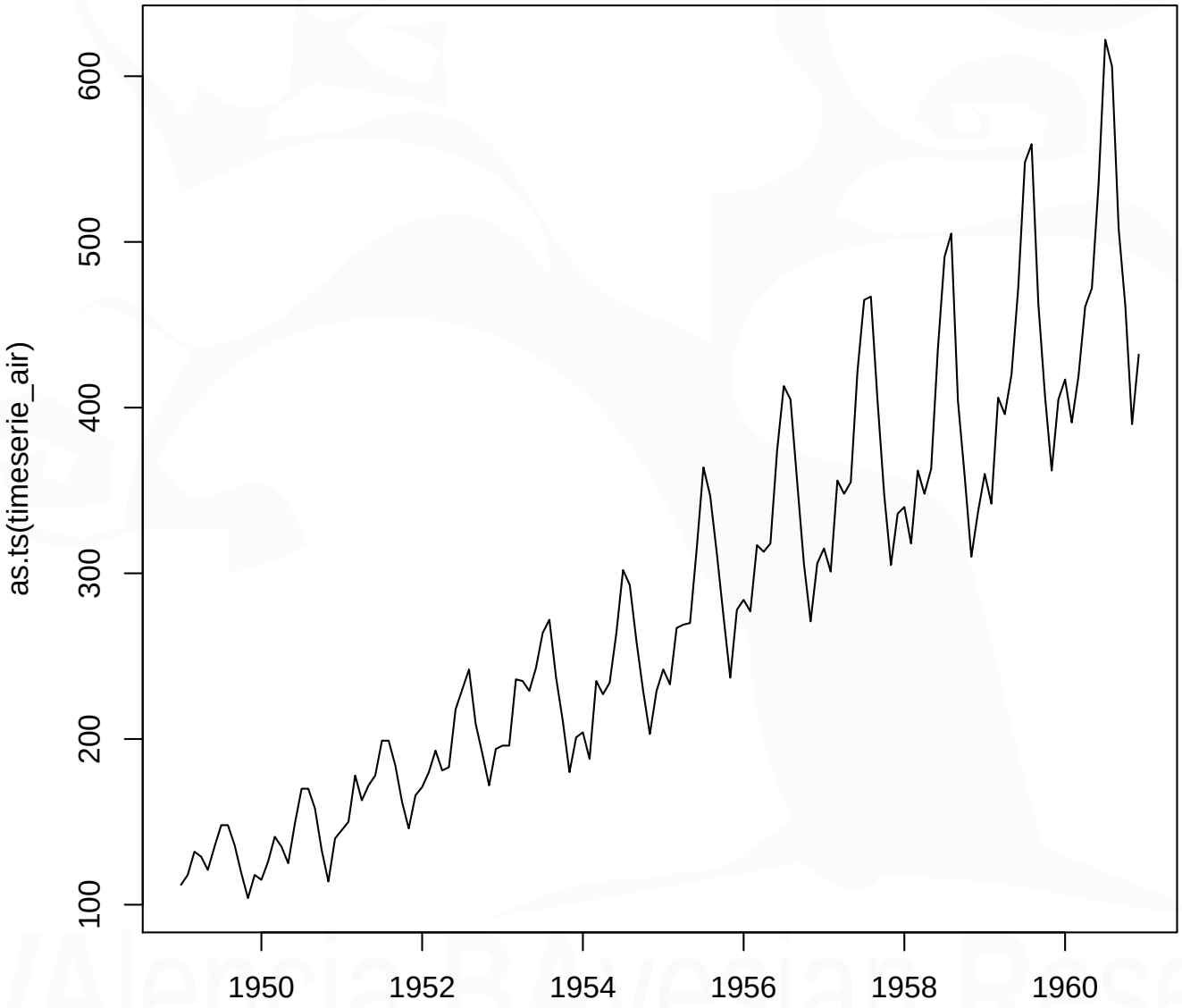
Una serie es estacional cuando podemos observar en ella un patrón sistemático que se repite periódicamente (cada año, cada mes, etc., dependiendo de las unidades de tiempo en que vengan recogidos los datos).

Existen muchos ejemplos de series con comportamiento estacional. El hecho de que las vacaciones laborales se concentren en los meses de verano condi-ciona los valores de muchas series. Un claro ejemplo es el de las series rela-cionadas con el turismo, tales como número mensual de pernoctaciones hotele-ras, número de viajeros en avión registrado por meses, etc.

ADITIVA



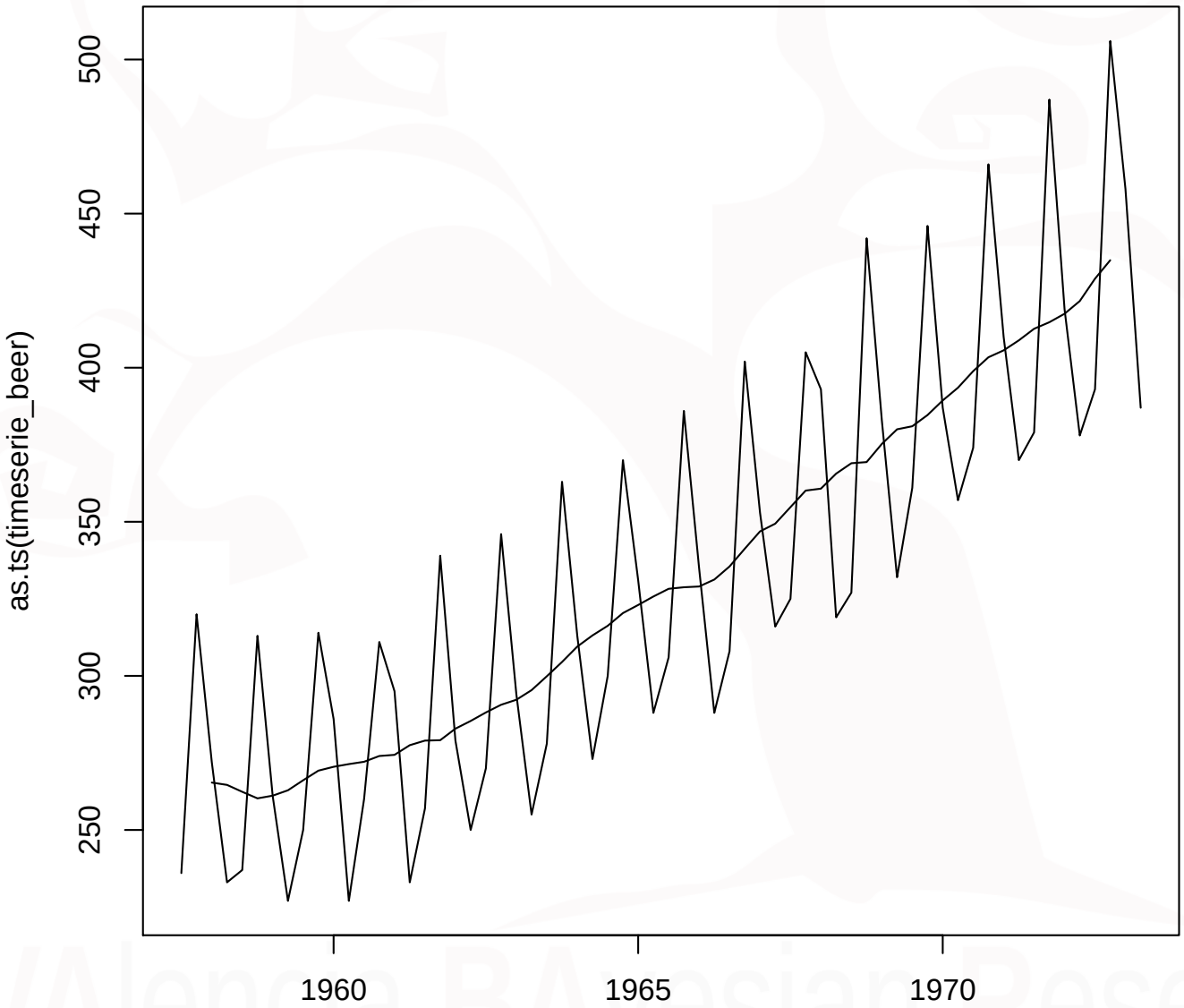
Multiplicativa



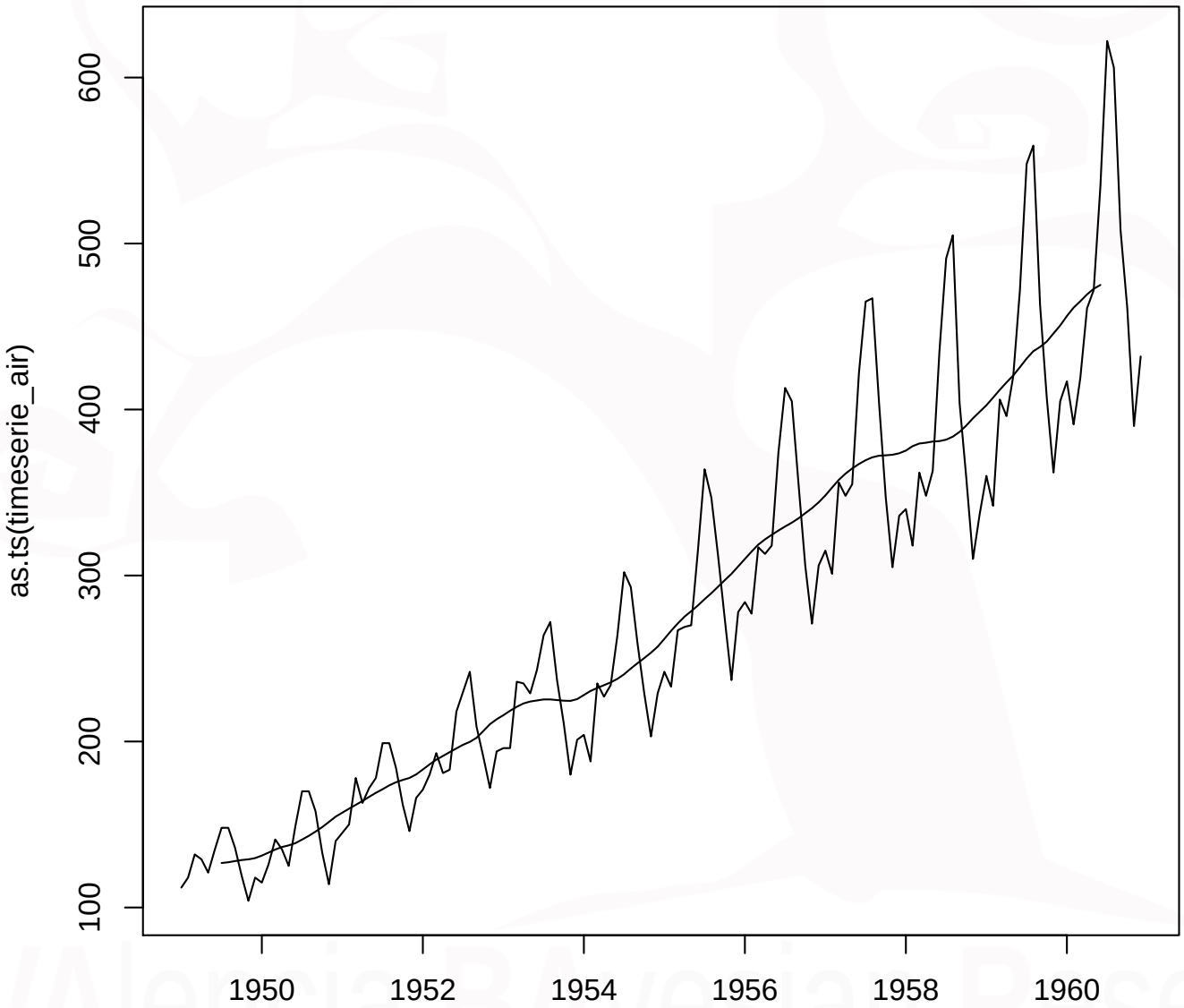
Tendencia

Si la variabilidad de una serie no depende del nivel significa que los componentes de la serie se combinan de forma aditiva, es decir, el incremento debido a la estacionalidad siempre es el mismo aunque exista tendencia creciente o decreciente. Si la variabilidad y el nivel dependen entre sí los elementos de la serie se combinan de forma multiplicativa. Esto quiere decir que el incremento debi-do a la estacionalidad aumenta o disminuye conforme la tendencia crece o decrece.

ADITIVA



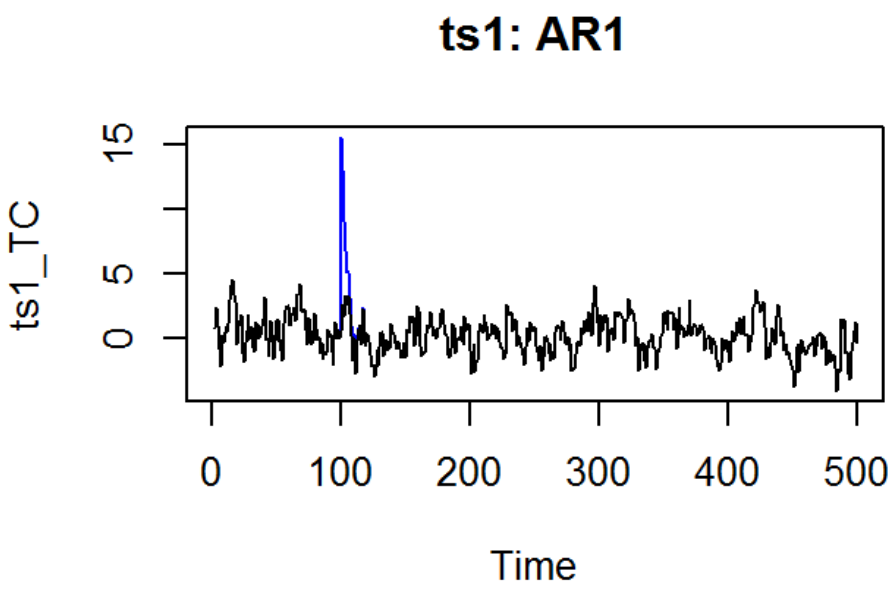
MULTIPLICATIVA

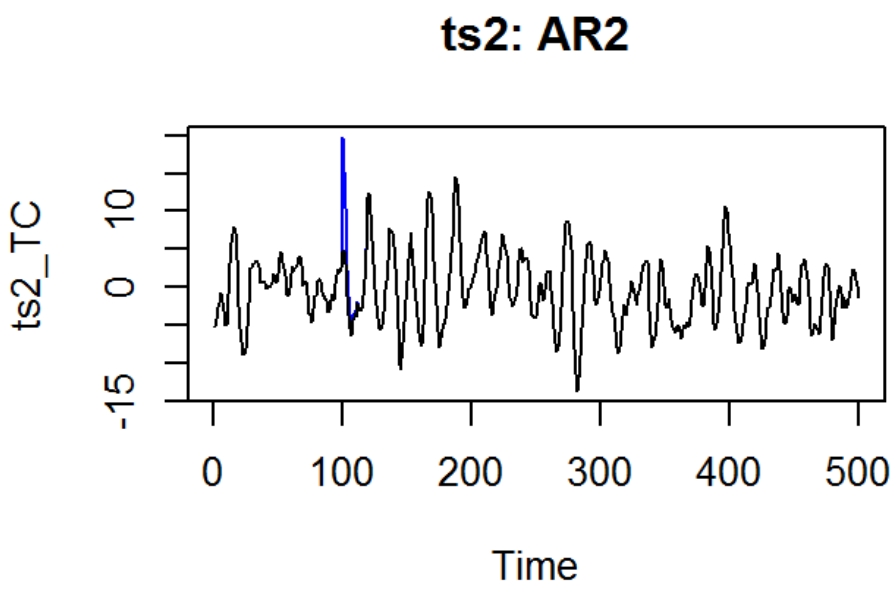


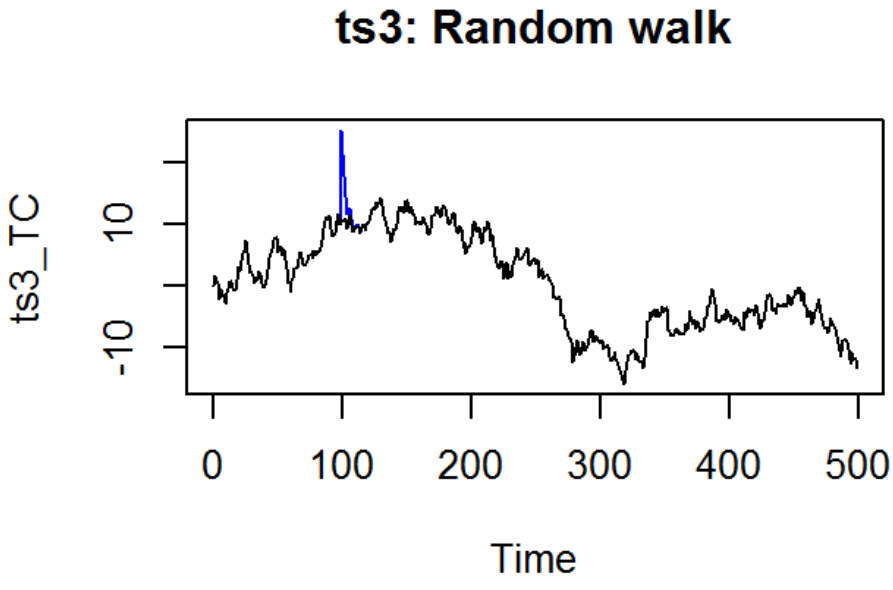
Comportamientos anómalos

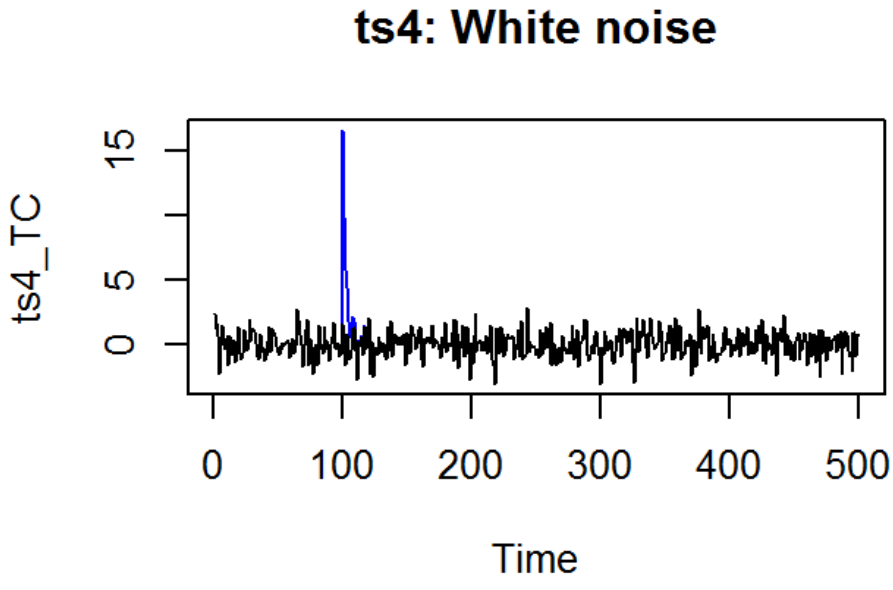
Otro aspecto a estudiar en el análisis preliminar de una serie temporal es el de los comportamientos extraños.

Si una serie temporal tiene valores perdidos (en un determinado momento no se han recogido datos) o valores raros, no podemos ignorarlos. Los comporta-mientos anómalos pueden ser de tres tipos: cambios de tendencia, subidas bruscas de nivel o aparición de valores extraños.


ts1: AR1
ts1_TC
15
5
0
0
100
200
300
400
500
Time


ts2: AR2
ts2_TC
10
0
-15
0
100
200
300
400
500
Time


ts3: Random walk
ts3_TC
10
-10
0
100
200
300
400
500
Time


ts4: White noise
ts4_TC
15
5
0
0
100
200
300
400
500
Time

MÉTODOS CLÁSICOS

Introducción

Desde una perspectiva teórica el enfoque clásico de análisis de series temporales considera que el comportamiento de una variable en el tiempo es el resulta-do de la integración de cuatro componentes fundamentales (aunque no siem-pre aparecen todos):

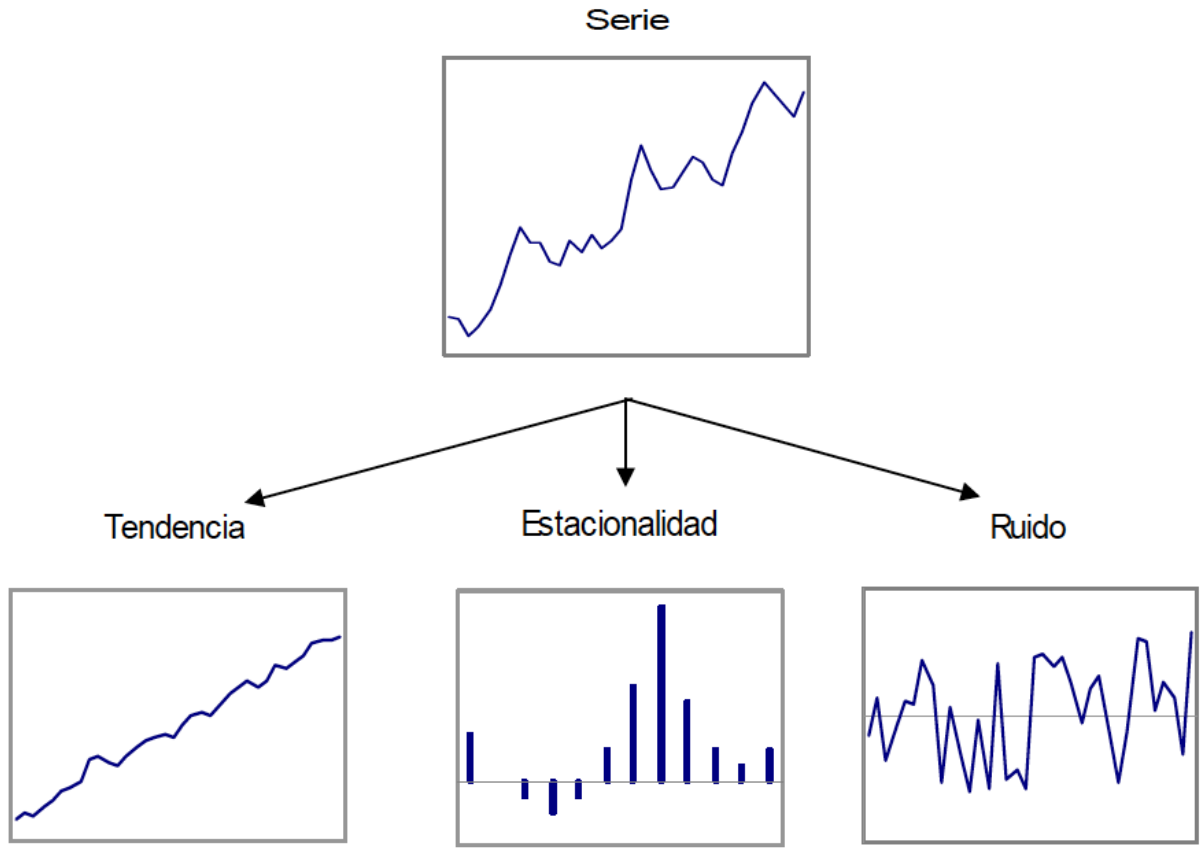
- tendencia (T_t),
- ciclo (C_t),
- componente estacional (S_t) y
- componente irregular o ruido (E_t).

De esta forma con los métodos clásicos una serie temporal X_t es una función de estos cuatro componentes.

$$X_t = f(C_t, T_t, S_t, E_t)$$

Métodos de descomposición estacional y ajuste de tendencia

Los métodos de descomposición estacional son eminentemente descriptivos. Tratan de separar la serie en subseries correspondientes a la tendencia-ciclo4, la estacionalidad y el ruido (componente aleatorio).

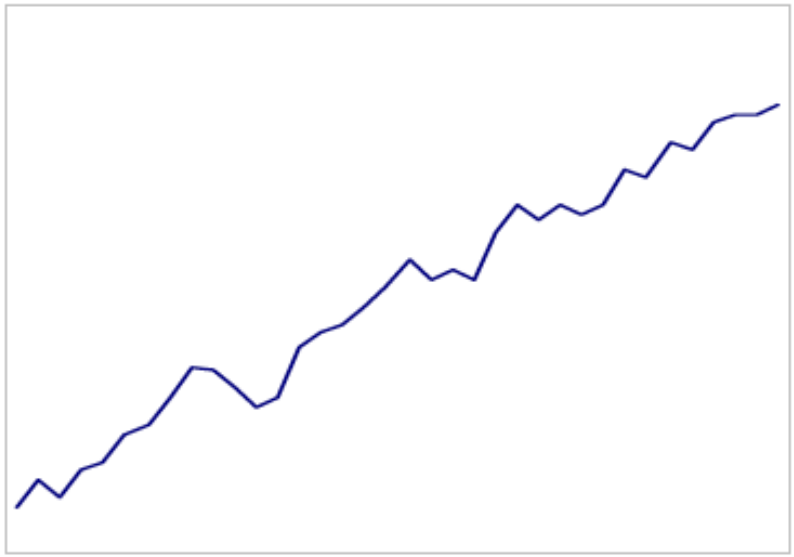


En ocasiones tendencia y estacionalidad se enmascaran, a veces una tenden-cia marcada puede no dejarnos ver la estacionalidad, y viceversa. Los métodos de descomposición estacional separan tendencia, estacionalidad y ruido, pero no predicen. Para predecir es necesario combinarlos con métodos de ajuste de tendencia.

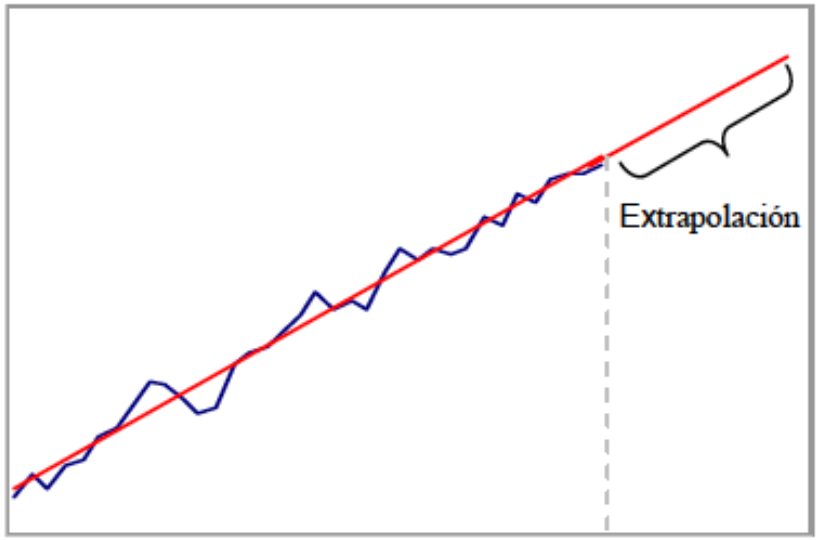
A la hora de predecir consideramos la estacionalidad constante periodo a pe-riodo y el ruido cero. El ruido es aleatorio, impredecible, y tiene media cero, de manera que la mejor previsión que podemos hacer de él es cero.

De esta forma para pronosticar realizamos un ajuste de tendencia con el fin de obtener un modelo extrapolable, y le añadimos la estacionalidad.

Tendencia



Tendencia con ajuste



El primer paso a seguir a la hora de descomponer una serie es determinar cómo se combinan sus componentes. Las combinaciones aditiva y multiplicativa son las más habituales

Las combinaciones aditiva y multiplicativa son las más habituales5. Tal y como hemos visto e una serie temporal X_t es una función que depende de cuatro componentes:

- **Componentes aditivas**: $X_t = C_t + T_t + S_t + E_t$
- **Componentes multiplicativas**: $X_t = C_t \times T_t \times S_t \times E_t$

Modelos Aditivos

En primer lugar eliminamos el ruido y la estacionalidad.

- El ruido se elimina sustituyendo cada observación por una media de lo ocurrido anteriormente (media móvil anterior).
- La estacionalidad realizando un proceso de media móvil centrada.
  - Este último procedimiento suaviza cada observación tomando la media de igual número de valores anteriores y posteriores a la misma.
  - El orden de la media móvil centrada, es decir, el número total de observaciones que generará cada media móvil centrada, habitualmente es igual al periodo de la serie.

Una vez eliminados la estacionalidad y el ruido obtenemos una serie que únicamente está formada por la tendencia y el ciclo: $M_t = C_t + T_t$.

A continuación, eliminando la tendencia y el ciclo de la serie de partida, conseguimos una serie integrada solo por la estacionalidad y el ruido:

$$X_t - M_t = E_t + S_t$$

- Para estimar el factor estacional a partir de esta última serie dia de todas las observaciones disponibles de cada unidad de periodo (por ejemplo, cada mes de un año).

De esta forma la serie del factor estacional estará formada por los valores de estas medias, es decir, será una serie que repetirá constantemente los mismos valores en cada unidad de periodo. Haciendo

$$X_t - M_t - S_t = E_t$$

obtenemos una serie con el error, con el ruido.

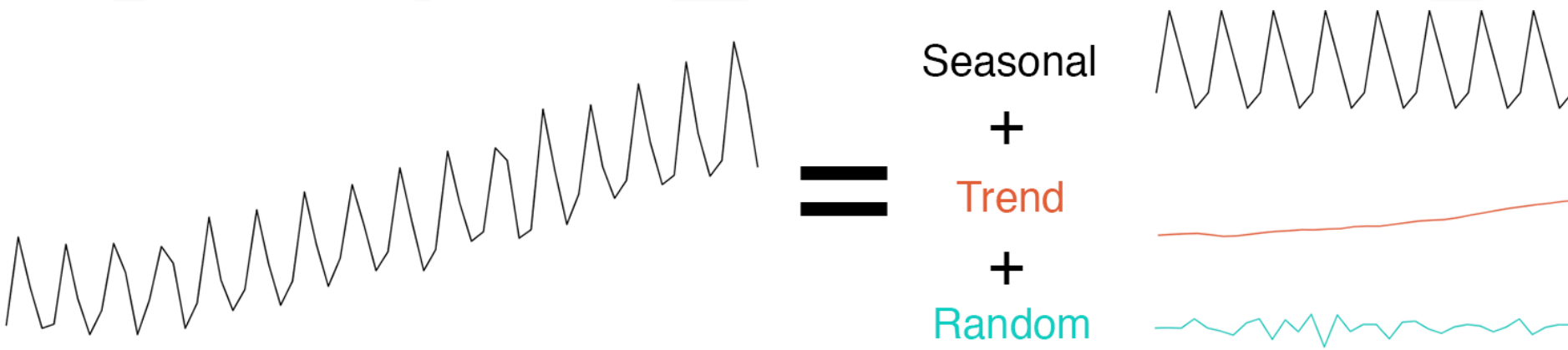
Modelo Multiplicativo

Encaso de que los componentes de la serie se combinen de forma multiplicativa el proceso es equivalente. En primer mediante medias móviles obtenemos una serie formada únicamente por la tendencia y el ciclo: $M_t = C_t \times T_t$.

A continuación haciendo $\frac{X_t}{M_t} = E_t \times S_t$ conseguimos una serie formada solo por la estacionalidad y el ruido.

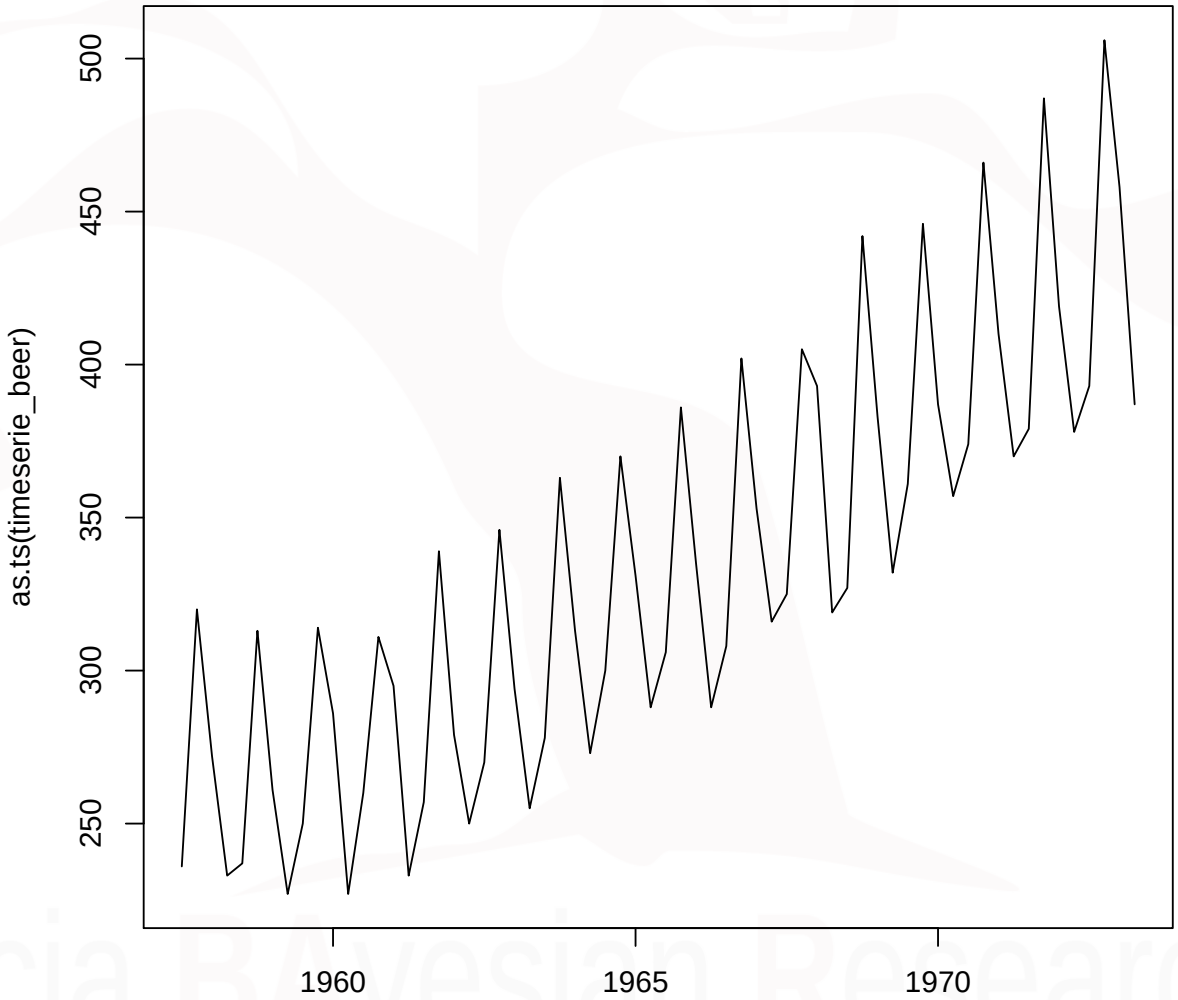
Finalmente, tras estimar la componente estacional haciendo $\frac{X_t}{M_t \times S_t} = E_t$ obtenemos una serie con el error.

Descompiniendo en R



Serie Aditiva

```
plot(as.ts(timeserie_beer))
```



Detectando la TENDENCIA

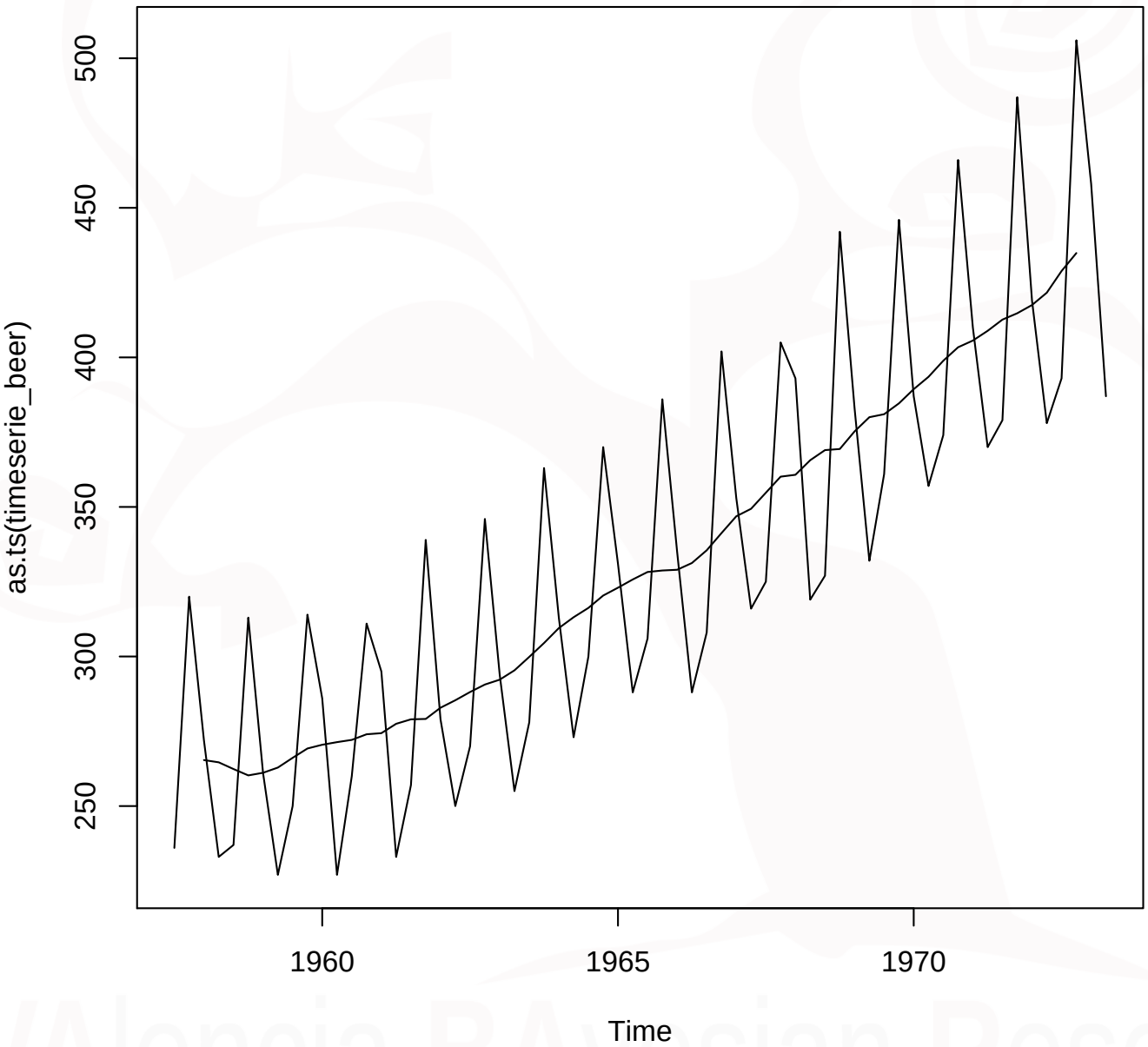
Primero miramos los datos:

```
head(ausbeer, 16)
```

	Q1	Q2	Q3	Q4
1956	284	213	227	308
1957	262	228	236	320
1958	272	233	237	313
1959	261	227	250	314

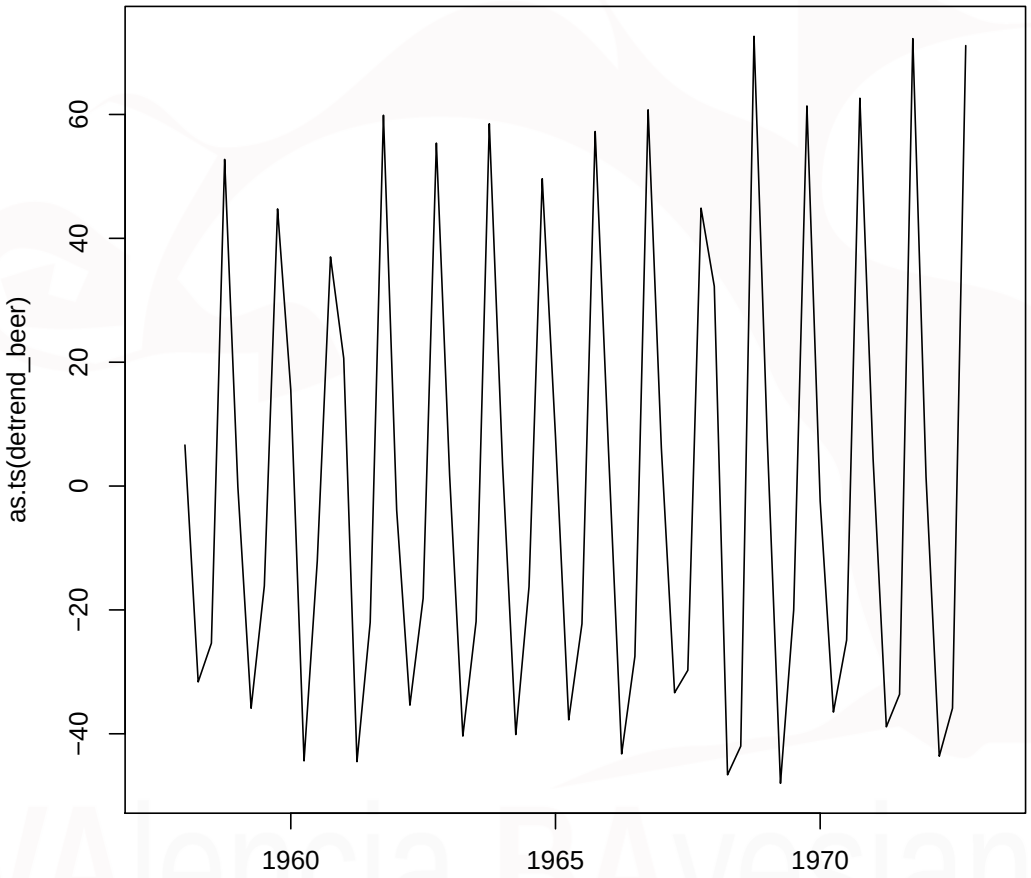
Queda claro que el periodo de estos datos es un año, descompuesto es 4 trimestres Q_1, Q_2, Q_3 y Q_4.

```
library(forecast)
trend_beer = ma(timeserie_beer, order = 4, centre = T)
plot(as.ts(timeserie_beer))
lines(trend_beer)
plot(as.ts(trend_beer))
```


as.ts(timeserie_beer)
250
300
350
400
450
500
1960
1965
1970
Time

Una vez calculada esa tendencia *trend_beer*, podemos eliminar dicha tendencia de la serie:

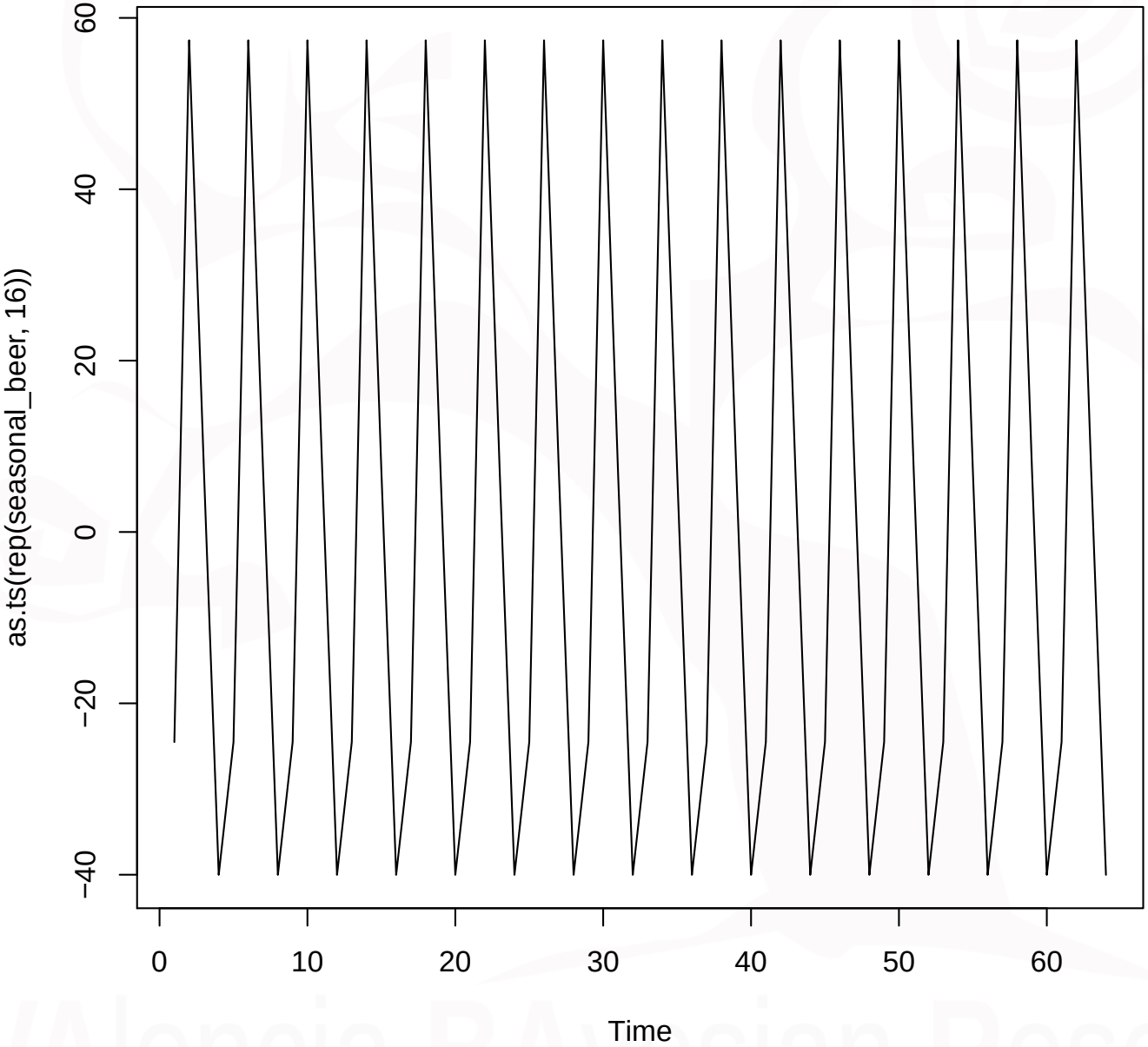
```
detrend_beer = timeserie_beer - trend_beer
plot(as.ts(detrend_beer))
```



Detectando la ESTACIONALIDAD

El siguiente paso es el estudio de la estacionalidad, esta podría ser anual, podría ser cada 5 años... para ello vamos a graficarla:

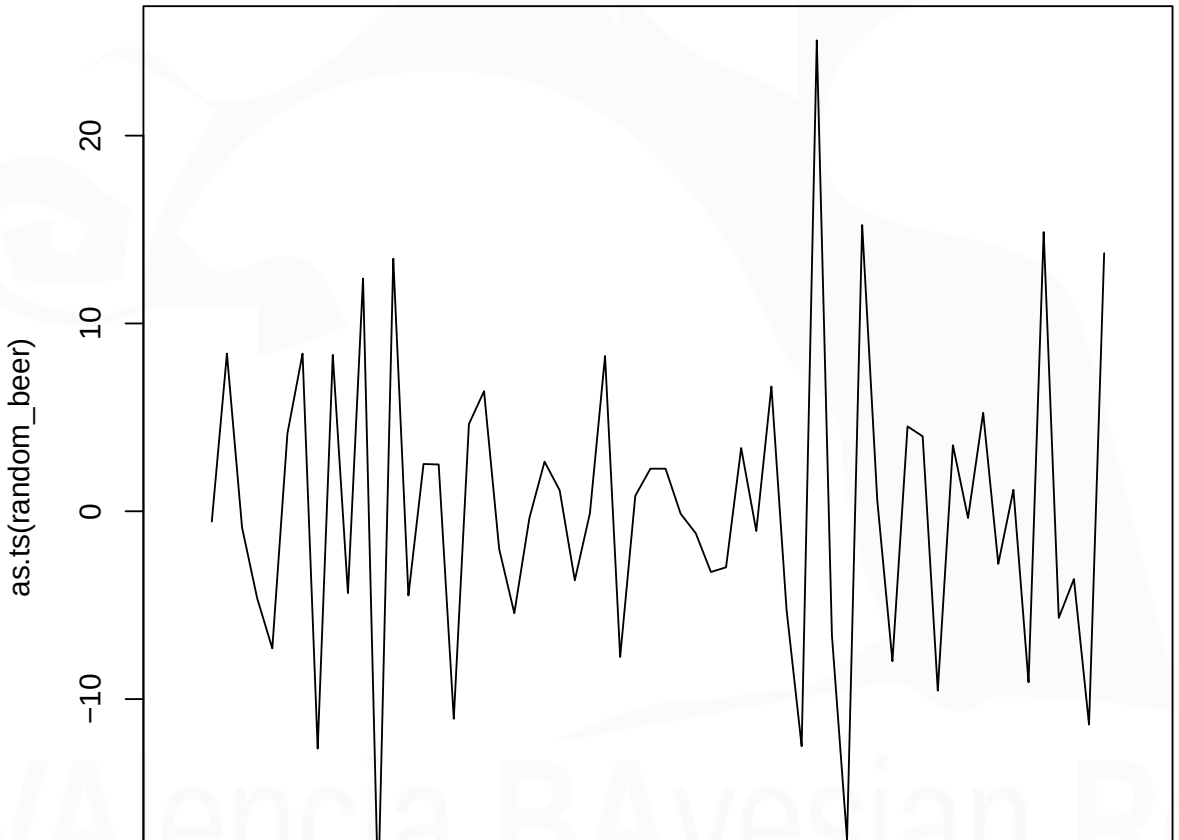
```
m_beer = t(matrix(data = detrend_beer, nrow = 4))
seasonal_beer = colMeans(m_beer, na.rm = T)
plot(as.ts(rep(seasonal_beer, 16)))
```


as.ts(rep(seasonal_beer, 16))
60
40
20
0
−20
−40
0
10
20
30
40
50
60
Time

Generando el "Ruido Blanco"

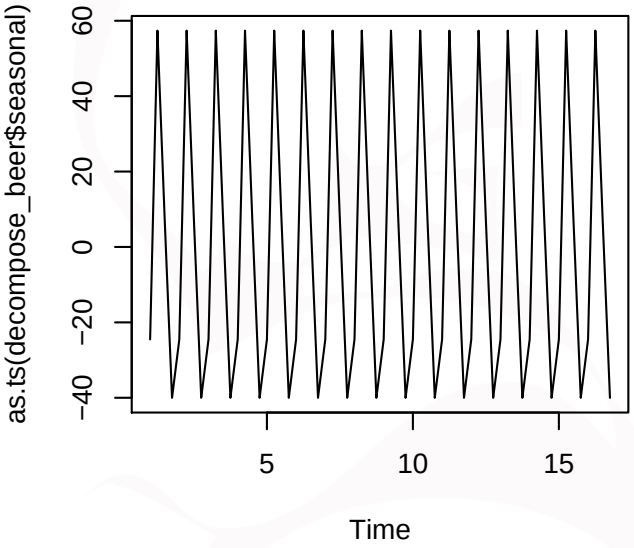
El último paso será obtener el "proceso sin"random noise" (¿ruido aleatorio?)

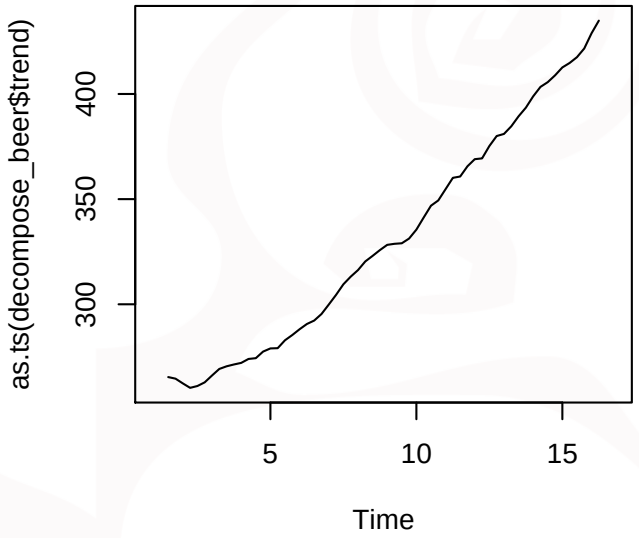
```
random_beer = timeserie_beer - trend_beer - seasonal_beer
plot(as.ts(random_beer))
```

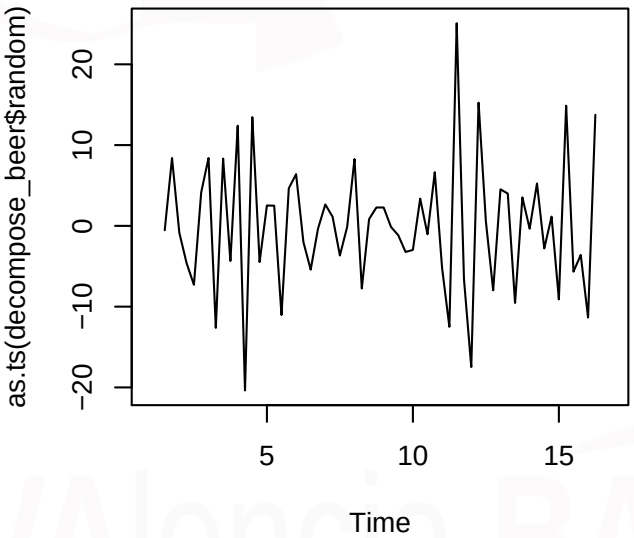


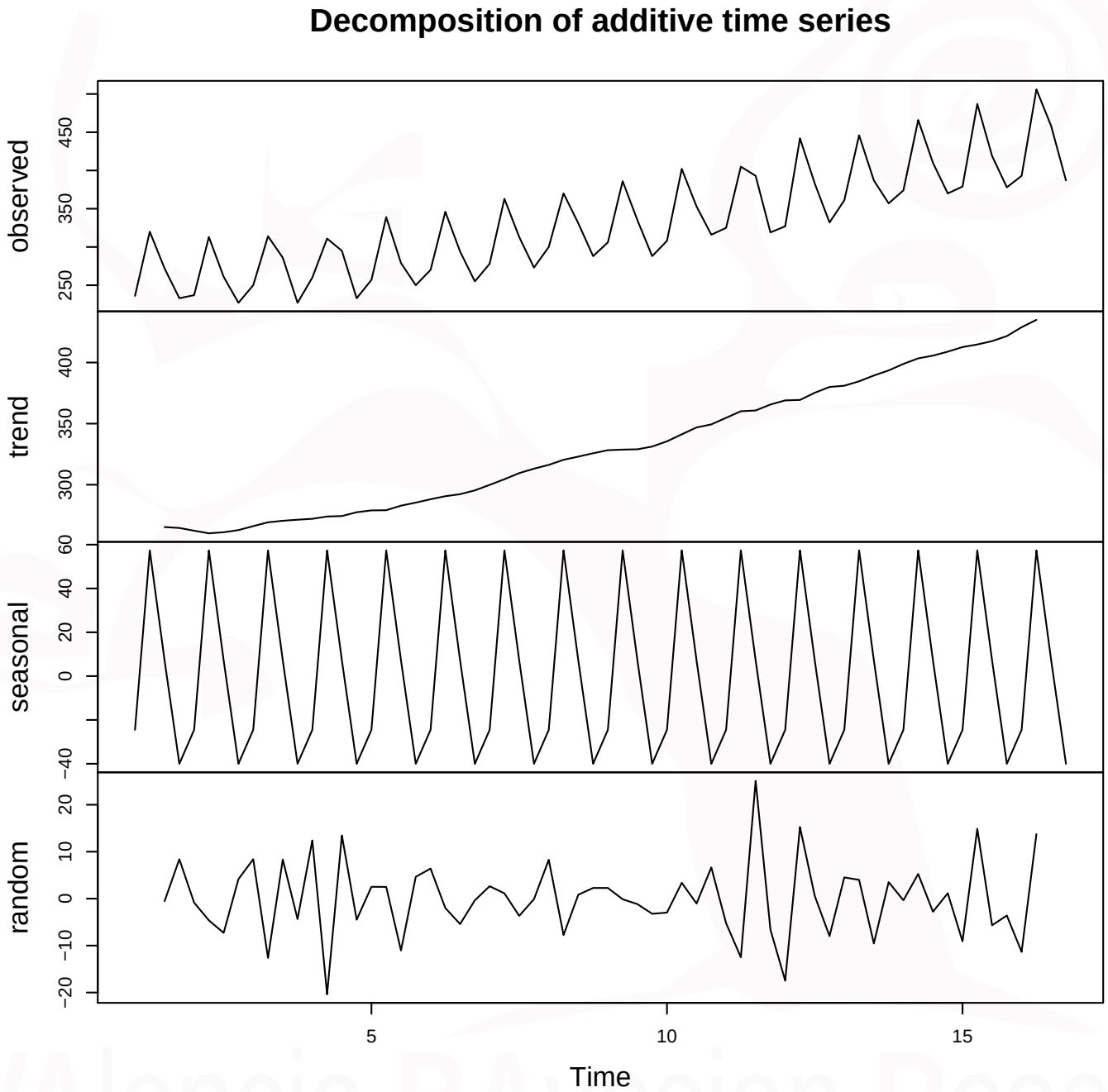
DECOMPOSE( ) y STL() Con el fin de agilizar todo esto utilizaremos el comando ***decompose***:

```
ts_beer = ts(timeserie_beer, frequency = 4)
decompose_beer = decompose(ts_beer, "additive")

plot(as.ts(decompose_beer$seasonal))
plot(as.ts(decompose_beer$trend))
plot(as.ts(decompose_beer$random))
plot(decompose_beer)
```


as.ts(decompose_beer$seasonal)
60
40
20
0
−20
−40
5
10
15
Time

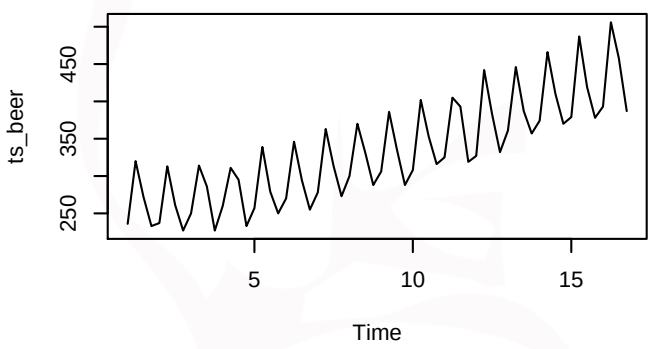

as.ts(decompose_beer$trend)
400
350
300
5
10
15
Time

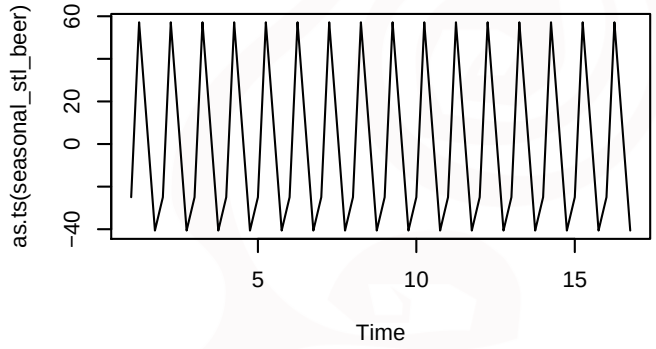

as.ts(decompose_beer$random)
20
10
0
−10
−20
5
10
15
Time

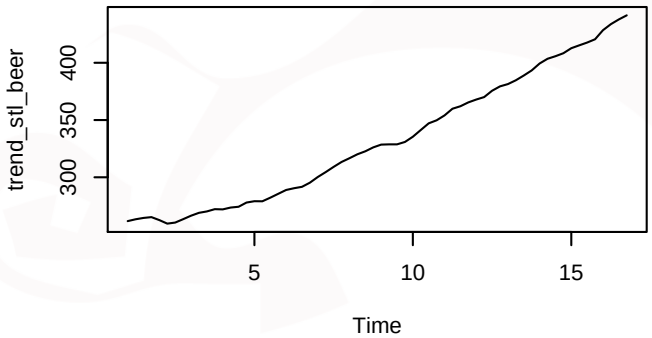

Decomposition of additive time series
observed
trend
seasonal
random
250
350
450
300
350
400
0
20
40
60
−40
−20
−10
0
10
20
5
10
15
Time

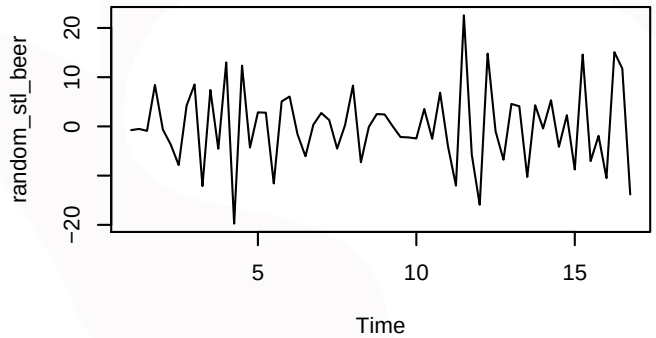
Veamos ahora el resultado si utilizamos el comando ***slt***:

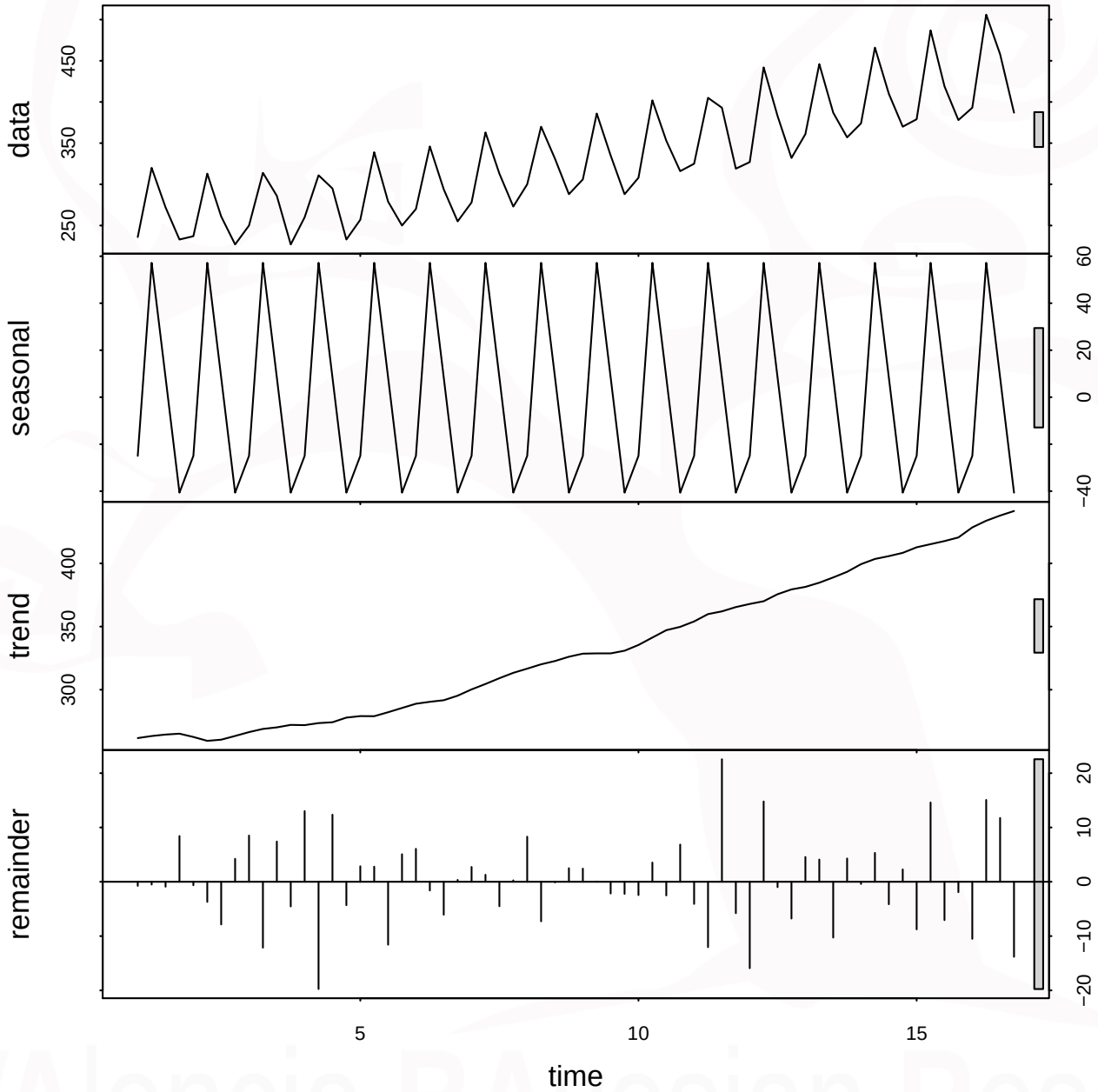
```
ts_beer = ts(timeserie_beer, frequency = 4)
stl_beer = stl(ts_beer, "periodic")
seasonal_stl_beer <- stl_beer$time.series[, 1]
trend_stl_beer <- stl_beer$time.series[, 2]
random_stl_beer <- stl_beer$time.series[, 3]

plot(ts_beer)
plot(as.ts(seasonal_stl_beer))
plot(trend_stl_beer)
plot(random_stl_beer)
plot(stl_beer)
```


ts_beer
250
350
450
5
10
15
Time


as.ts(seasonal_stl_beer)
−40
0
20
60
5
10
15
Time


trend_stl_beer
300
350
400
5
10
15
Time


random_stl_beer
−20
0
10
20
5
10
15
Time


data
seasonal
trend
remainder
250
350
450
−40
0
20
40
60
300
350
400
−20
−10
0
10
20
5
10
15
time

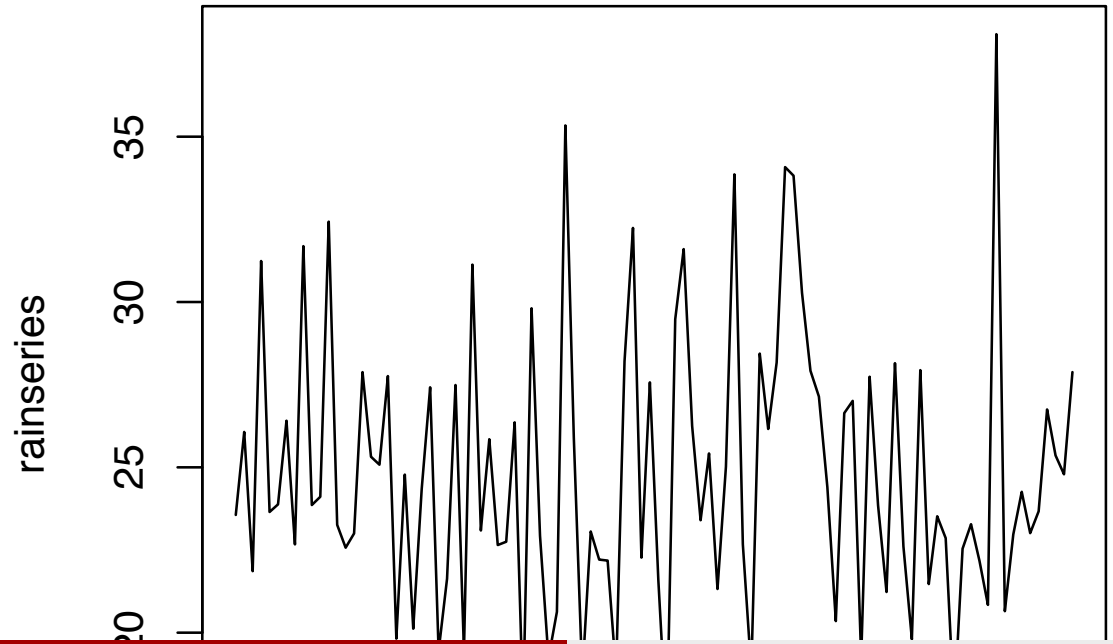
Suavizado Exponencial

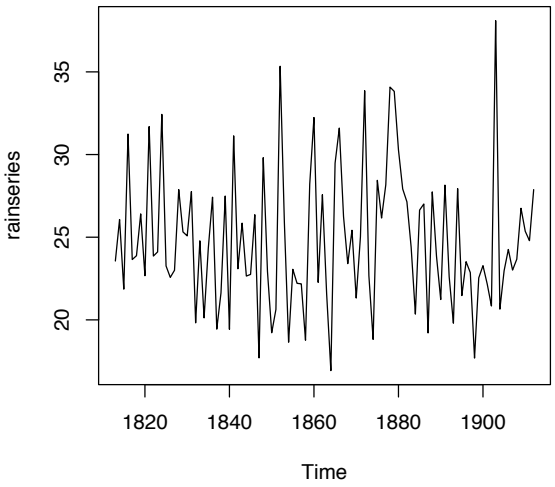
El **suavizado exponencial** puede utilizarse para predecir a corto plazo en las series temporales.

- Los métodos de suavizado o alisado son técnicas de tipo predictivo más que descriptivo. Resultan más adecuados para pronosticar, y proporcionan previ-siones razonables para horizontes de predicción inmediatos. Además los resultados que se obtienen con ellos son satisfactorios, incluso cuando no se dispo-ne de un gran número de datos históricos.
- A diferencia de los métodos de descomposición estacional, para aplicar los de suavizado no es necesario que la serie presente comportamiento estacional. Dentro de estos últimos existen modelos para series no afectadas por tenden-cia ni estacionalidad, para series con tendencia y para series con tendencia y estacionalidad.

Simple Exponential Smoothing

```
rain <- scan("http://robjhyndman.com/tsdldata/hurst/precip1.dat",
    skip = 1)
rainseries <- ts(rain, start = c(1813))
plot.ts(rainseries)
```





Todos los modelos que se van a exponer a continuación proporcionan una mis-ma previsión para todo el horizonte de previsión. Por esta razón se exige que se apliquen a series sin tendencia ni estacionalidad. Se emplean para realizar predicciones a corto plazo, puesto que a largo plazo lo que se haría es actuali-zar la serie con la nueva información y efectuar de nuevo una predicción a corto plazo.

Utilizaremos la funcion ***HoltWinters()***:

HoltWinters(x, alpha = NULL, beta = NULL, gamma = NULL, seasonal = c("additive", "multiplicative"), start.periods = 2, l.start = NULL, b.start = NULL, s.start = NULL, optim.start = c(alpha = 0.3, beta = 0.1, gamma = 0.1), optim.control = list())

x	An object of class ts
alpha	alpha parameter of Holt-Winters Filter.
beta	beta parameter of Holt-Winters Filter. **If set to FALSE, the function will do exponential smoothing**.
gamma	gamma parameter used for the seasonal component. **If set to FALSE, an non-seasonal model is fitted**.
seasonal	Character string to select an "additive" (the default) or "multiplicative" seasonal model. The first few characters are sufficient. (Only takes effect if gamma is non-zero).
start.periods	Start periods used in the autodetection of start values. Must be at least 2.
l.start	Start value for level ($a[0]$).
b.start	Start value for trend ($b[0]$).
s.start	Vector of start values for the seasonal component ($s_1[0] \ldots s_p[0]$)
optim.start	Vector with named components alpha, beta, and gamma containing the starting values for the optimizer. Only the values needed must be specified. Ignored in the one-parameter case.
optim.control	Optional list with additional control parameters passed to optim if this is used. Ignored in the one-parameter case.

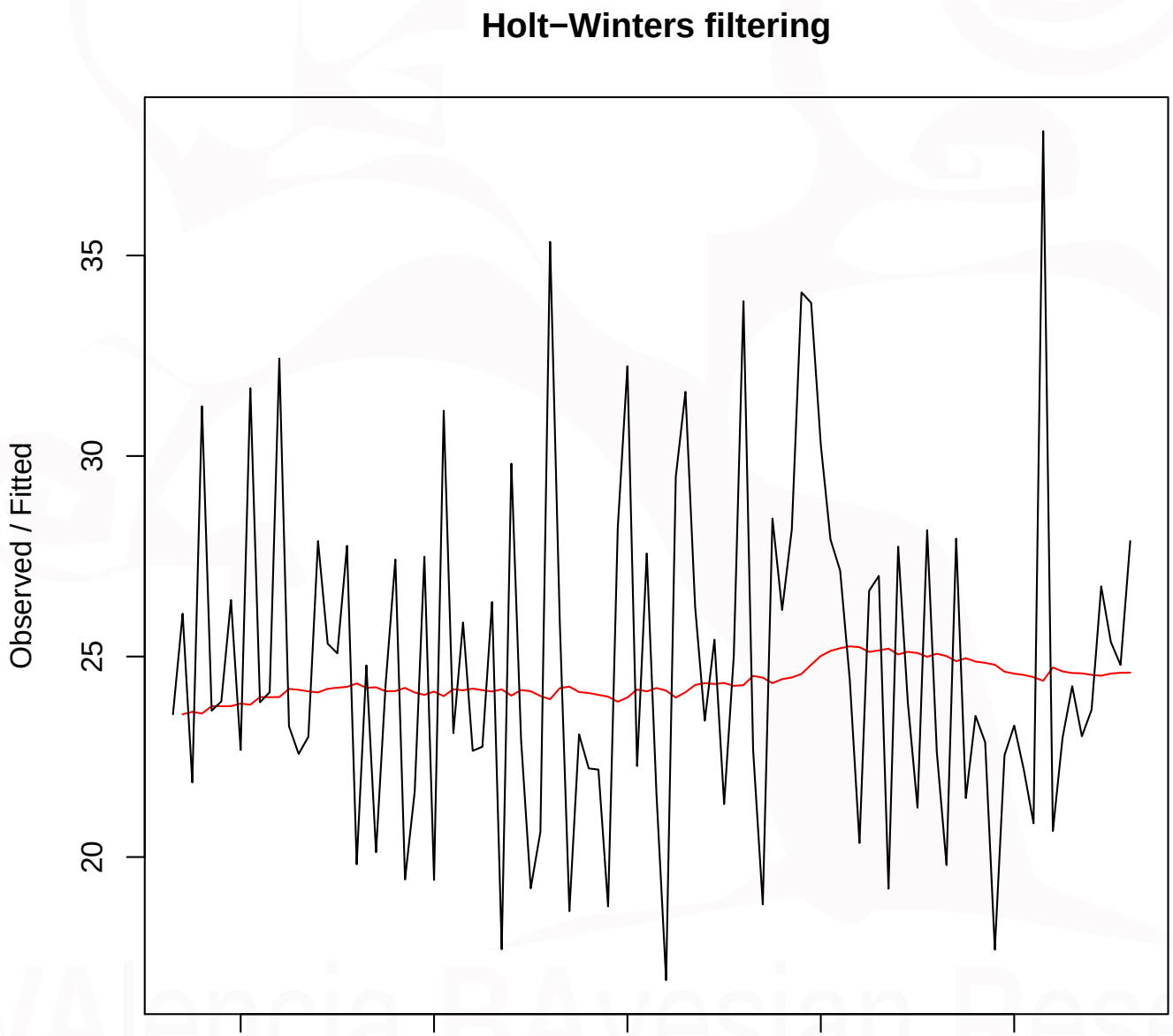
```
rainseriesforecasts <- HoltWinters(rainseries, beta = FALSE,
    gamma = FALSE)
rainseriesforecasts
```

```
## Holt-Winters exponential smoothing without trend and without seas
##
## Call:
## HoltWinters(x = rainseries, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.02412151
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 24.67819
```

```
rainseriesforecasts$fitted
```

	xhat	level
1814	23.56000	23.56000
1815	23.62054	23.62054
1816	23.57808	23.57808
1817	23.76290	23.76290
1818	23.76017	23.76017
1819	23.76306	23.76306
1820	23.82691	23.82691
1821	23.79900	23.79900
1822	23.98935	23.98935
1823	23.98623	23.98623
1824	23.98921	23.98921
1825	24.19282	24.19282
1826	24.17032	24.17032
1827	24.13171	24.13171
1828	24.10442	24.10442
1829	24.19549	24.19549
1830	24.22261	24.22261
1831	24.24329	24.24329
1832	24.32812	24.32812
1833	24.21938	24.21938
1834	24.23290	24.23290
1835	24.13369	24.13369
1836	24.13867	24.13867
1837	24.21782	24.21782
1838	24.10257	24.10257
1839	24.04293	24.04293

```
plot(rainseriesforecasts)
```



La medida para saber el grado de exactitud de las predicciones será la Suma de Cuadrados de los Errores (SSE).

```
rainseriesforecasts$SSE
```

```
## [1] 1828.855
```

Una forma de mejorar la capacidad predictiva de este método es indicarle el valor inicial.

```
HoltWinters(rainseries, beta = FALSE, gamma = FALSE, l.start = 23.56)
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = rainseries, beta = FALSE, gamma = FALSE, l.start = 23.56)
##
## Smoothing parameters:
##  alpha: 0.02412151
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 24.67819
```

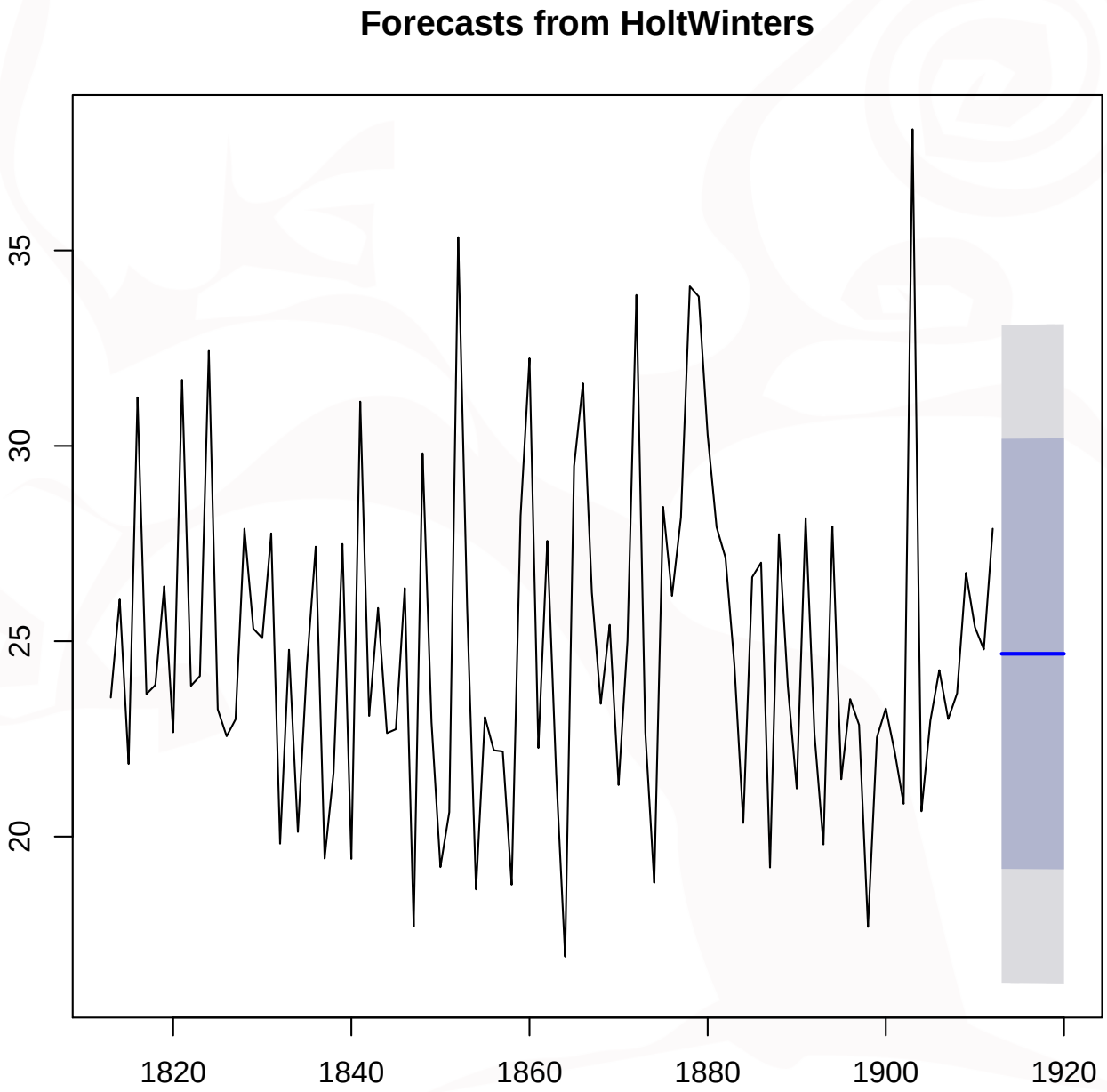
Existen diferentes paquetes para realizar la predicción de *HoltWinters* con *R*.

```
library("forecast")
rainseriesforecasts2 <- forecast(rainseriesforecasts, h = 8)
```

¿qué es *h=8*?

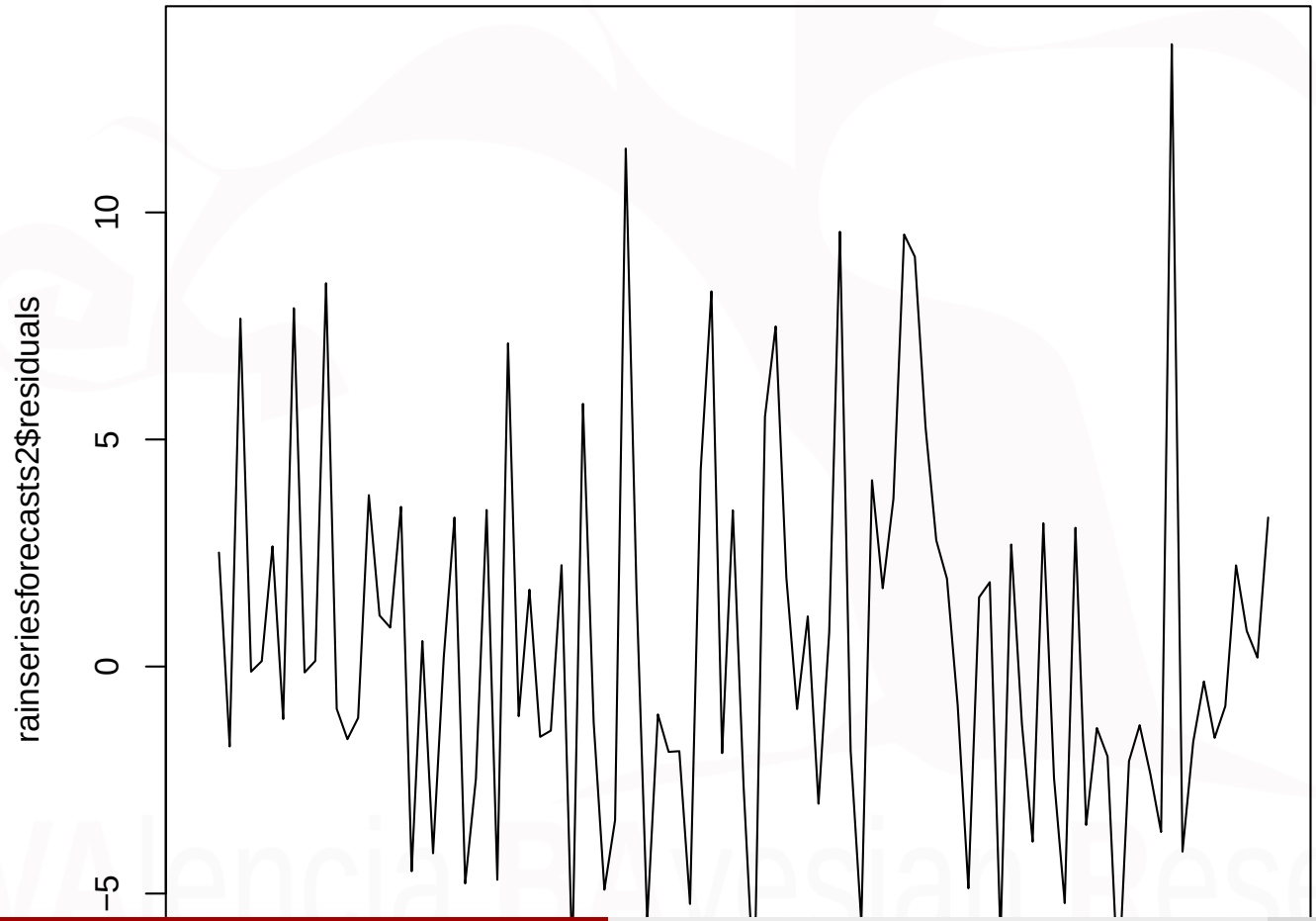
el perido del que queremos que nos haga la predicción

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1913	24.67819	19.17493	30.18145	16.26169	33.09470
1914	24.67819	19.17333	30.18305	16.25924	33.09715
1915	24.67819	19.17173	30.18465	16.25679	33.09960
1916	24.67819	19.17013	30.18625	16.25434	33.10204
1917	24.67819	19.16853	30.18785	16.25190	33.10449
1918	24.67819	19.16694	30.18945	16.24945	33.10694
1919	24.67819	19.16534	30.19105	16.24701	33.10938
1920	24.67819	19.16374	30.19265	16.24456	33.11182


Forecasts from HoltWinters
35
30
25
20
1820
1840
1860
1880
1900
1920

Y lo más importante es que los residuos después del suavizado exponencial sean un "ruido blanco":

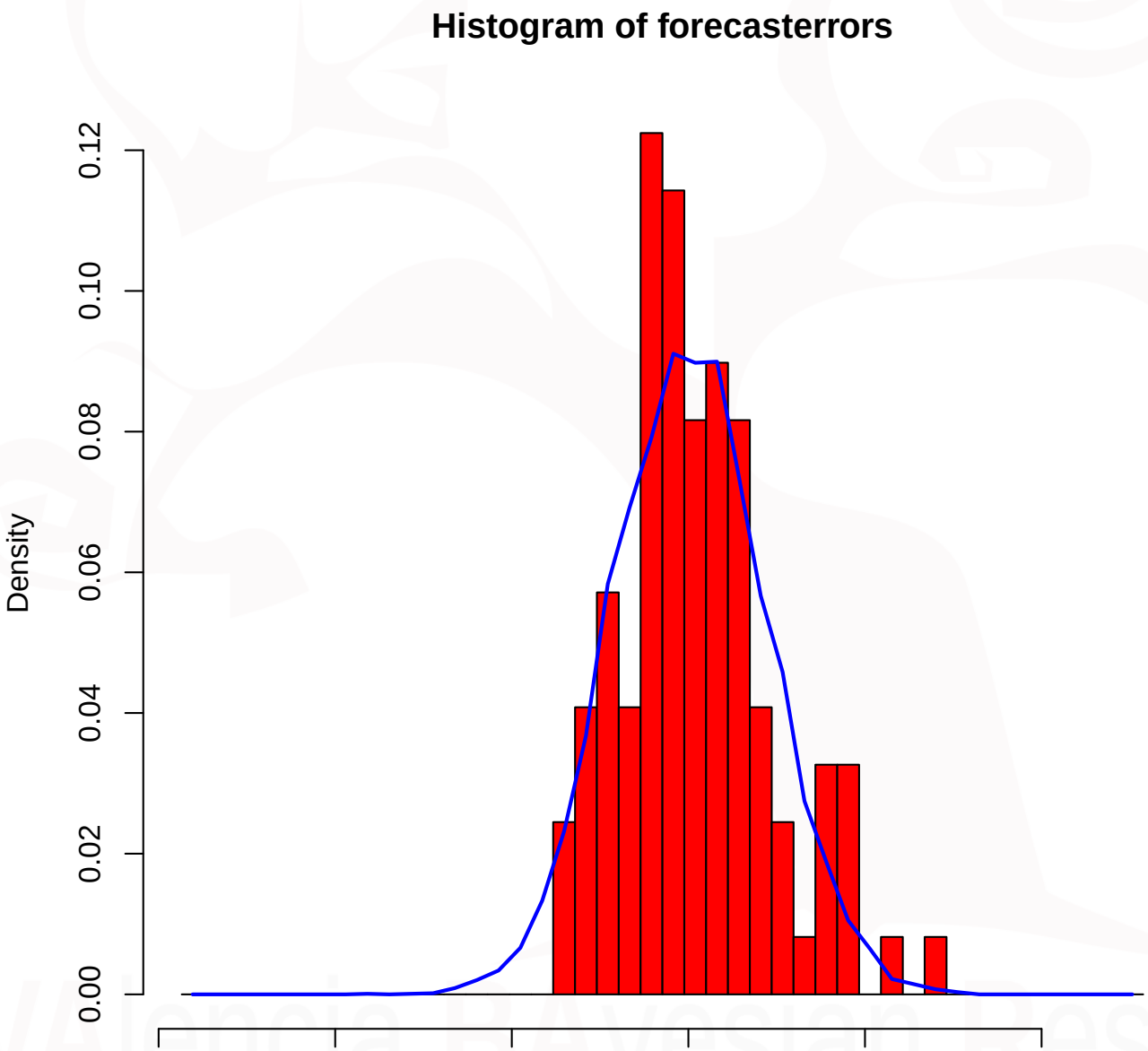
```
plot(rainseriesforecasts2$residuals)
```



Para estudiar el error que cometemos en las predicciones utilizaremos la siguiente función:

```
plotForecastErrors <- function(forecasterrors) {
    # make a histogram of the forecast errors:
    mybinsize <- IQR(forecasterrors, na.rm = T)/4
    mysd <- sd(forecasterrors, na.rm = T)
    mymin <- min(forecasterrors, na.rm = T) - mysd * 5
    mymax <- max(forecasterrors, na.rm = T) + mysd * 3
    # generate normally distributed data with mean 0 and standard
    # deviation mysd
    mynorm <- rnorm(10000, mean = 0, sd = mysd)
    mymin2 <- min(mynorm, na.rm = T)
    mymax2 <- max(mynorm, na.rm = T)
    if (mymin2 < mymin) {
        mymin <- mymin2
    }
    if (mymax2 > mymax) {
        mymax <- mymax2
    }
    # make a red histogram of the forecast errors, with the
    # normally distributed data overlaid:
    mybins <- seq(mymin, mymax, mybinsize)
    hist(forecasterrors, col = "red", freq = FALSE, breaks = mybins)
    # freq=FALSE ensures the area under the histogram = 1
    # generate normally distributed data with mean 0 and standard
    # deviation mysd
    myhist <- hist(mynorm, plot = FALSE, breaks = mybins)
    # plot the normal curve as a blue line on top of the
    # histogram of forecast errors:
    points(myhist$mids, myhist$density, type = "l", col = "blue",
        lwd = 2)
}
```

```
plotForecastErrors(rainseriesforecasts2$residuals)
```



Holt's Exponential Smoothing

Si tenemos una serie **con TENDENCIA** pero **sin estacionalidad** este es el método adecuado.

```
skirts <- scan("http://robjhyndman.com/tsdldata/roberts/skirts.dat",
    skip = 5)
skirtsseries <- ts(skirts, start = c(1866))
plot.ts(skirtsseries)
```


skirtsseries
600
700
800
900
1000
1870
1880
1890
1900
1910
Time

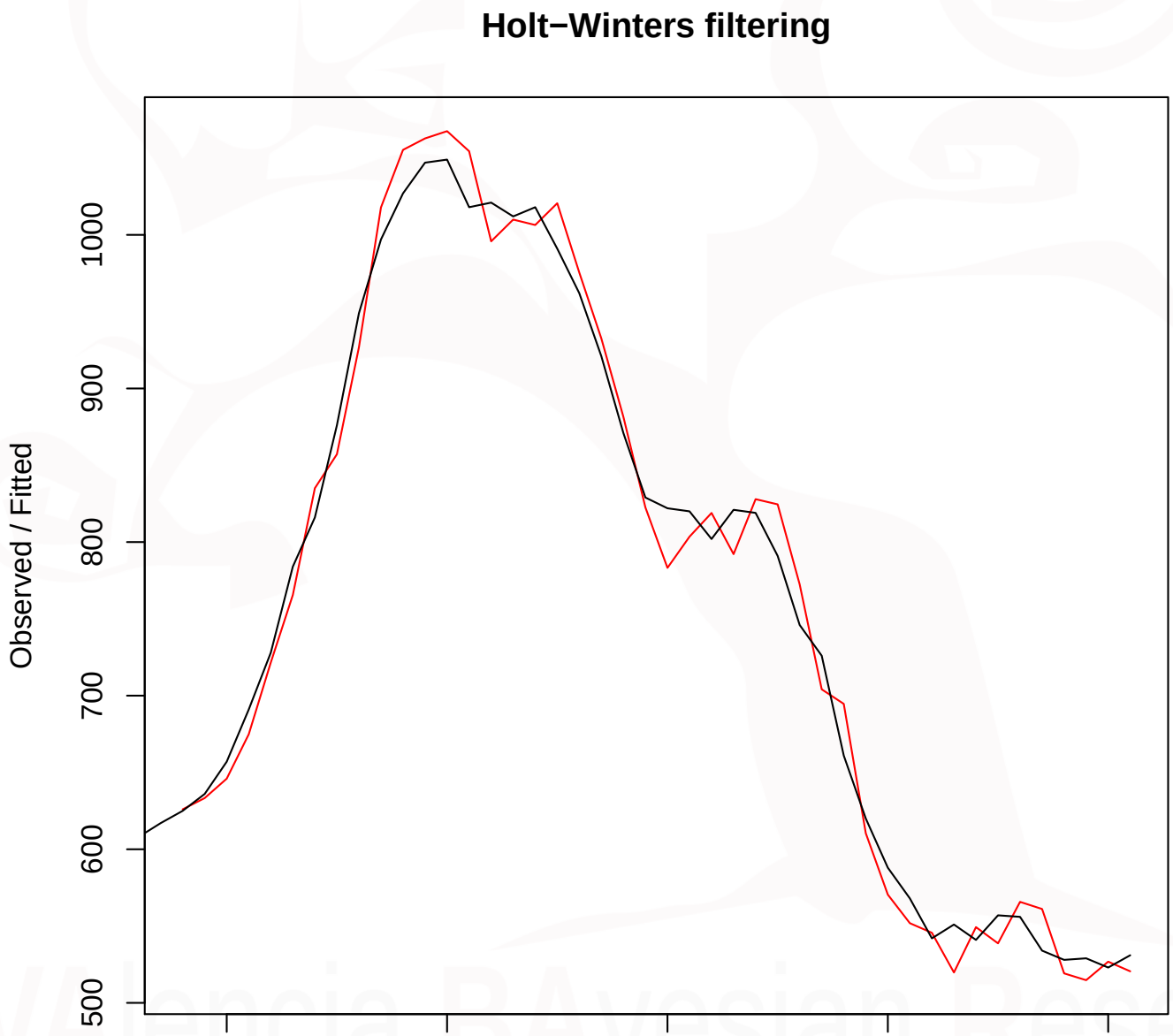
```
skirtsseriesforecasts <- HoltWinters(skirtsseries, gamma = FALSE)
skirtsseriesforecasts
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = skirtsseries, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.8383481
##  beta : 1
##  gamma: FALSE
##
## Coefficients:
##          [,1]
## a 529.308585
## b   5.690464
```

```
skirtsseriesforecasts$SSE
```

```
## [1] 16954.18
```

```
plot(skirtsseriesforecasts)
```



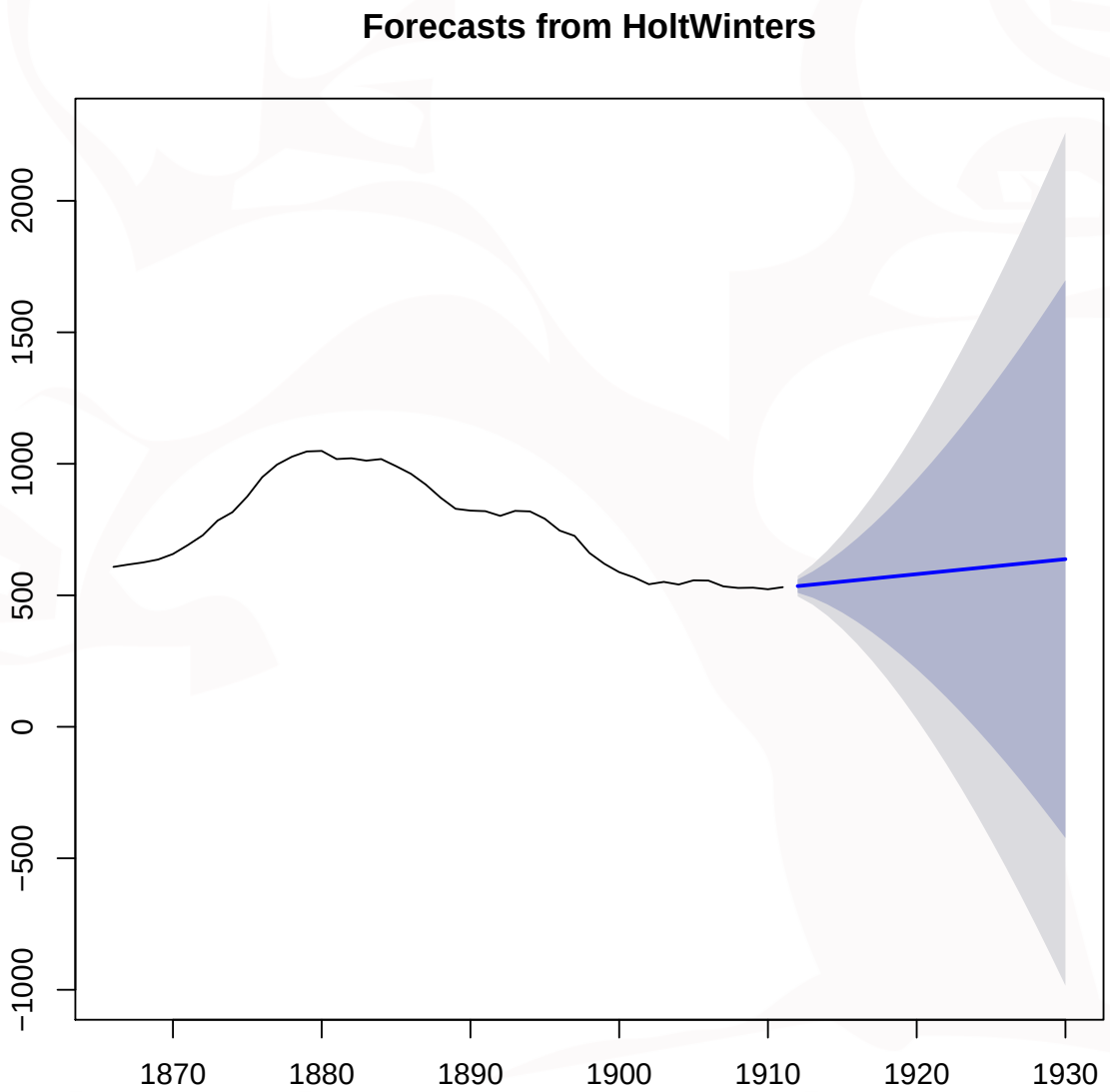
```
HoltWinters(skirtsseries, gamma = FALSE, l.start = 608, b.start = 9)
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = skirtsseries, gamma = FALSE, l.start = 608, b.start = 9)
##
## Smoothing parameters:
##  alpha: 0.8346775
##  beta : 1
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 529.278637
## b   5.670129
```

```
skirtsseriesforecasts2 <- forecast(skirtsseriesforecasts, h = 19)
```

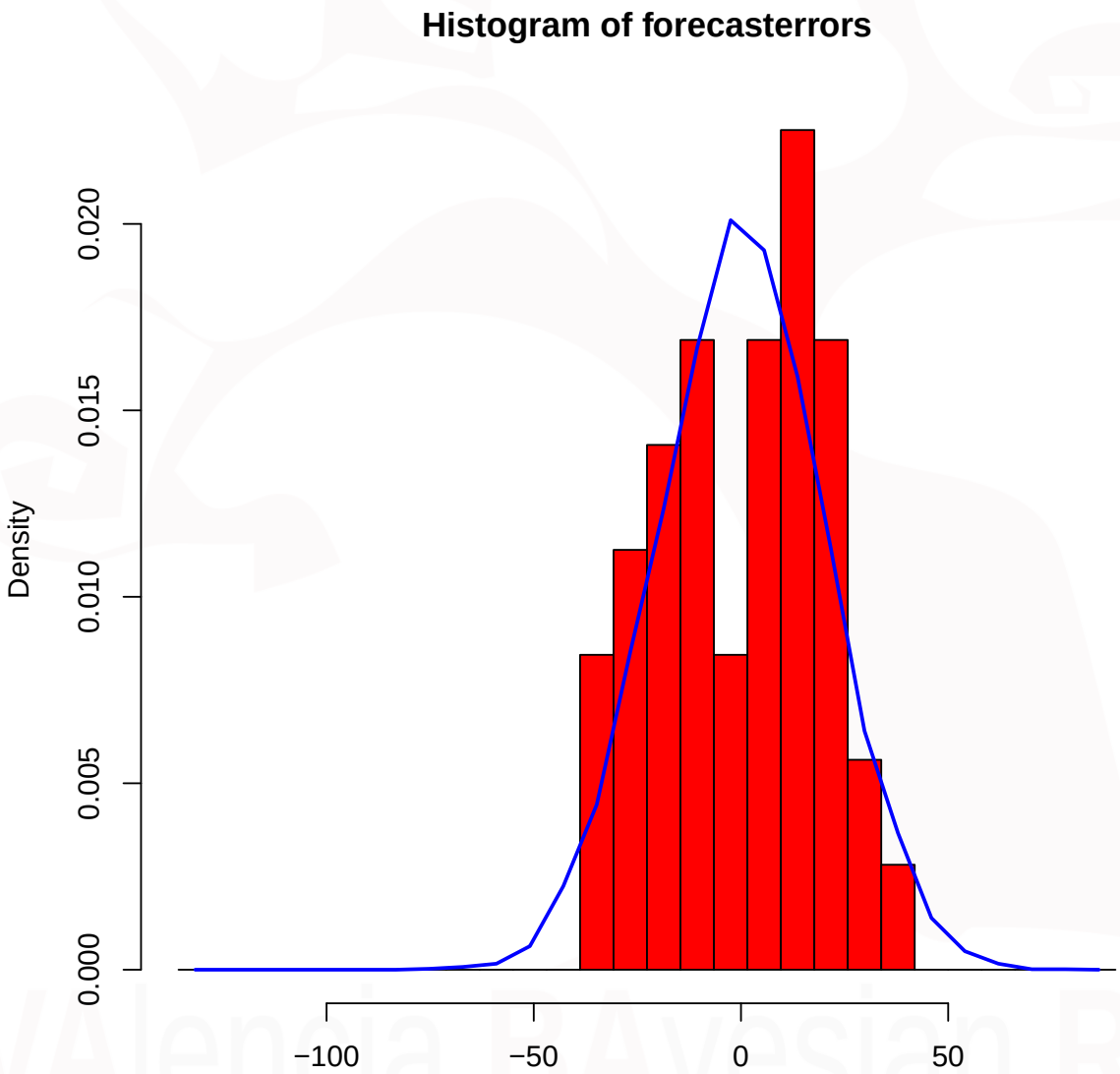
	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1912	534.9990	509.55210	560.4460	496.08130	573.9168
1913	540.6895	491.01052	590.3685	464.71204	616.6670
1914	546.3800	465.36129	627.3987	422.47258	670.2874
1915	552.0704	434.40205	669.7388	372.11216	732.0287
1916	557.7609	398.94120	716.5806	314.86713	800.6547
1917	563.4514	359.47147	767.4313	251.49103	875.4117
1918	569.1418	316.34076	821.9429	182.51596	955.7677
1919	574.8323	269.81480	879.8498	108.34829	1041.3163
1920	580.5228	220.10648	940.9391	29.31362	1131.7319
1921	586.2132	167.39191	1005.0345	-54.31870	1226.7452
1922	591.9037	111.82029	1071.9871	-142.32052	1326.1279
1923	597.5942	53.52019	1141.6681	-234.49517	1429.6835
1924	603.2846	-7.39593	1213.9652	-330.67069	1537.2399
1925	608.9751	-70.82861	1288.7788	-430.69495	1648.6451
1926	614.6655	-136.68903	1366.0201	-534.43211	1763.7632
1927	620.3560	-204.89720	1445.6092	-641.75986	1882.4719
1928	626.0465	-275.38060	1527.4736	-752.56728	2004.6602
1929	631.7369	-348.07309	1611.5470	-866.75319	2130.2271
1930	637.4274	-422.91396	1697.7688	-984.22478	2259.0796

```
plot(skirtsseriesforecasts2)
```



y los errores de la predicción:

```
plotForecastErrors(skirtsseriesforecasts2$residuals)
```

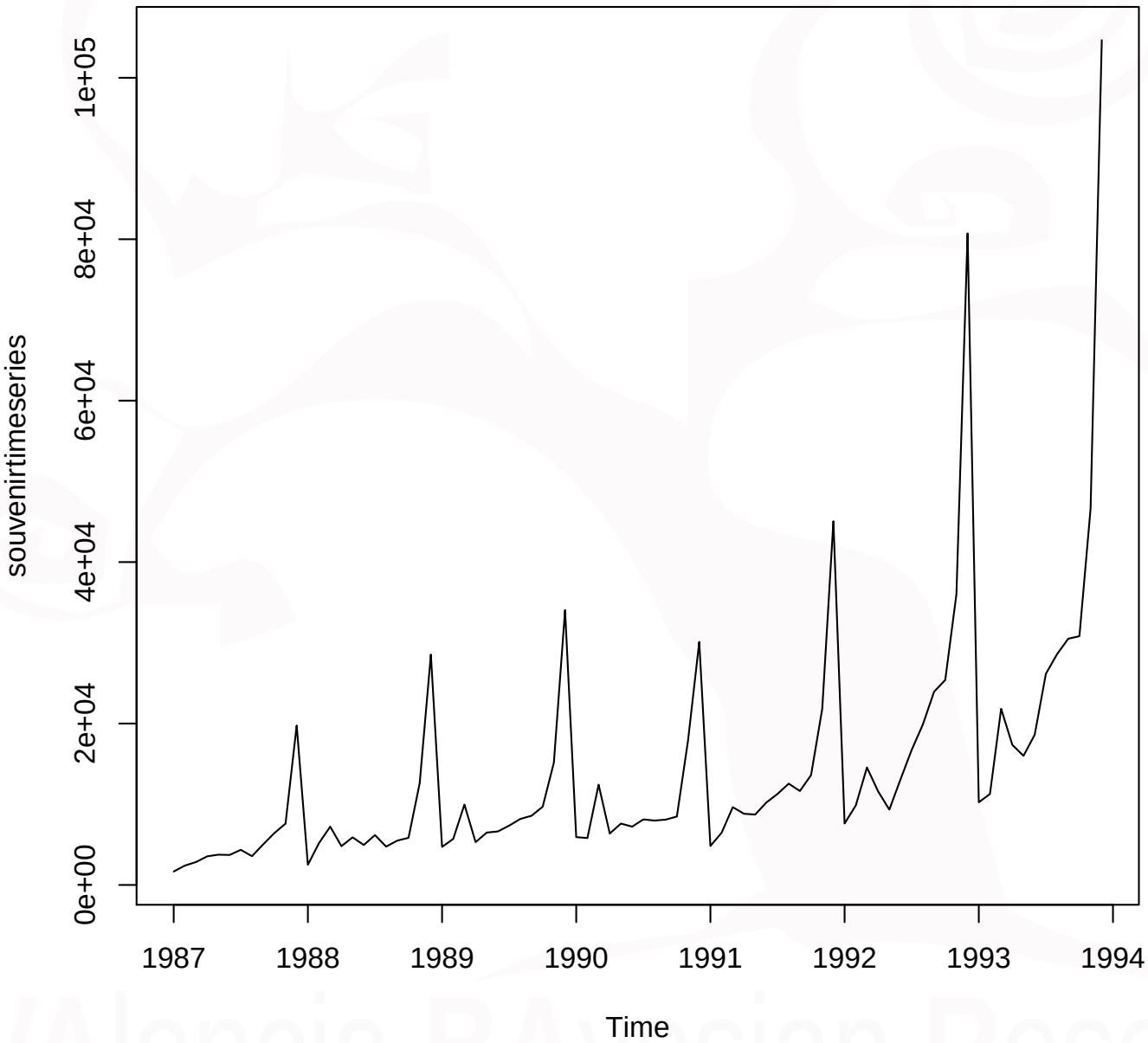


Holt-Winters Exponential Smoothing

Cuando tenemos **Tendencia** a la vez que **Estacionalidad**

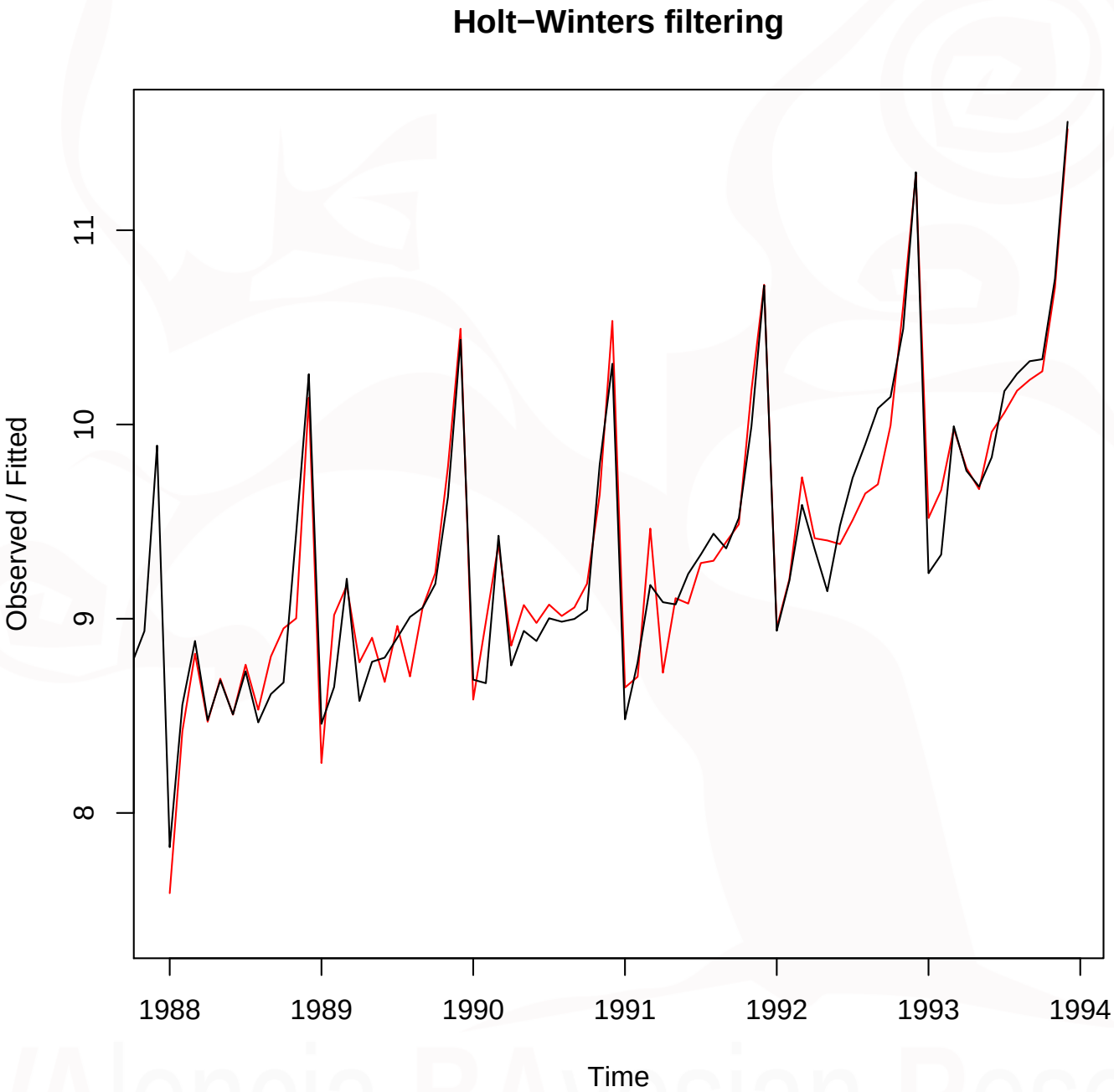
```
souvenir <- scan("http://robjhyndman.com/tsdldata/data/fancy.dat")
souvenirtimeseries <- ts(souvenir, frequency = 12, start = c(1987,
    1))
```

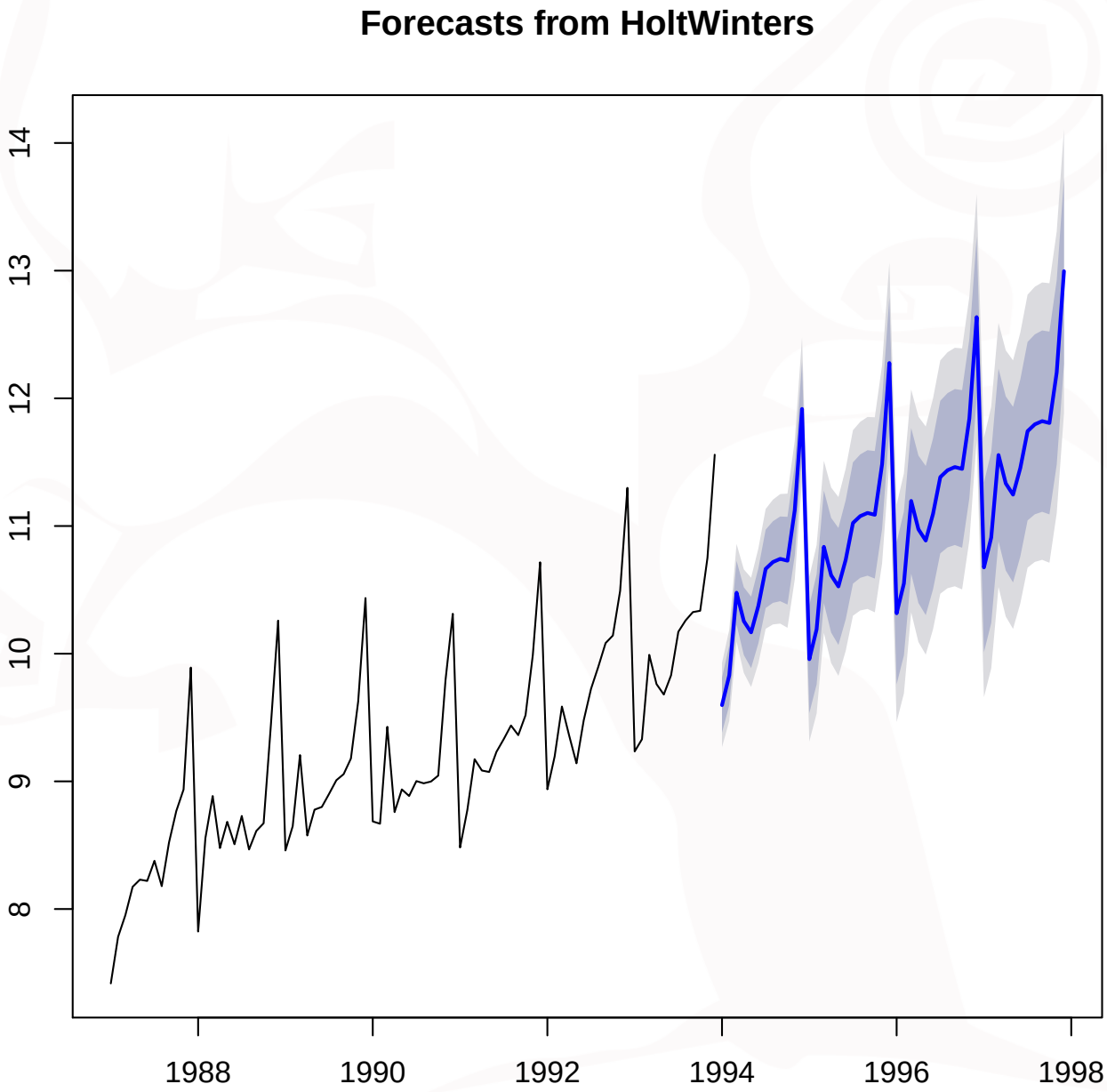
x
1664.81
2397.53
2840.71
3547.29
3752.96
3714.74

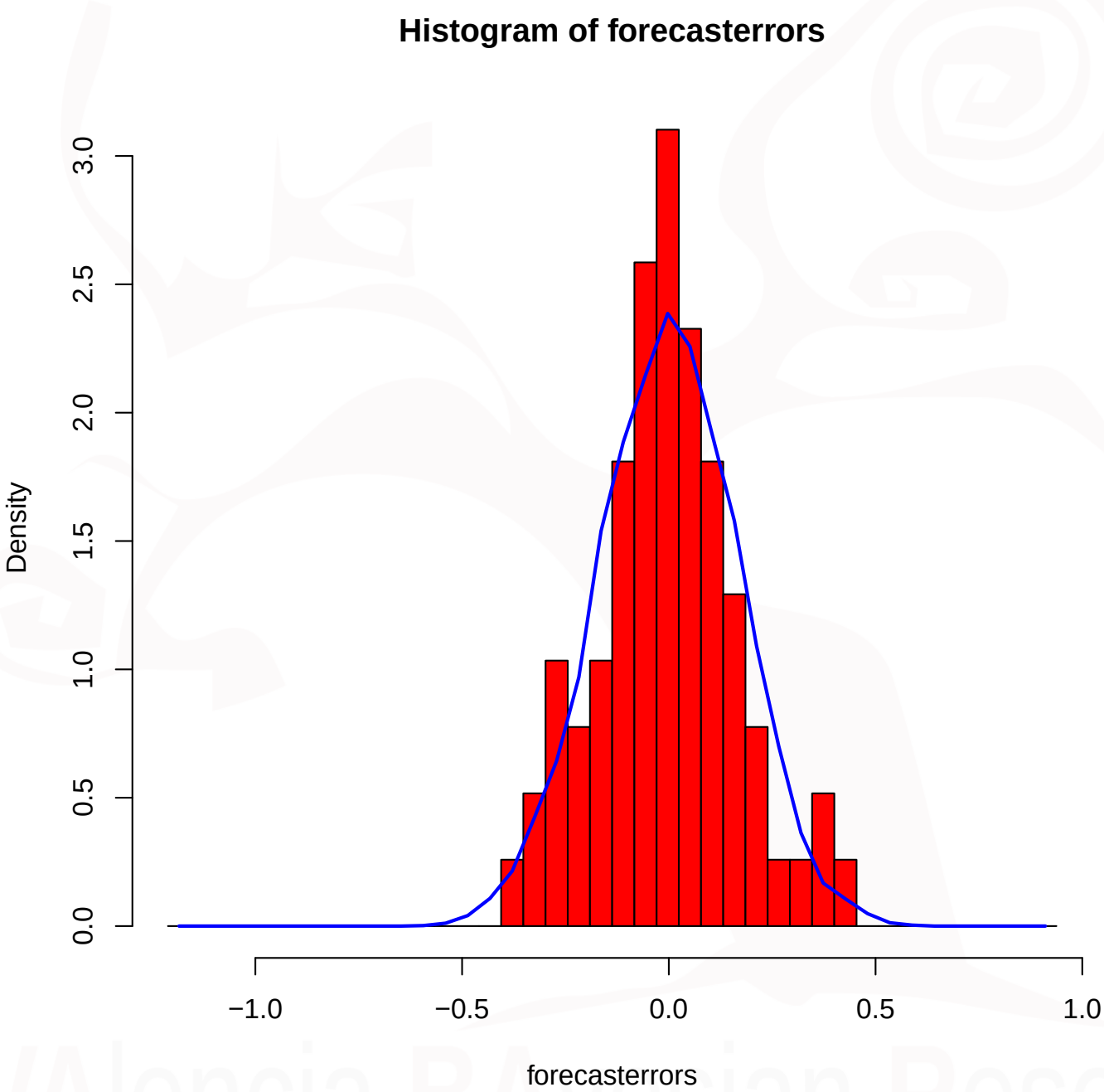

souvenirtimeseries
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
1987
1988
1989
1990
1991
1992
1993
1994
Time

```
logsouvenirtimeseries <- log(souvenirtimeseries)
souvenirtimeseriesforecasts <- HoltWinters(logsouvenirtimeseries)
souvenirtimeseriesforecasts
souvenirtimeseriesforecasts$SSE
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = logsouvenirtimeseries)
##
## Smoothing parameters:
##  alpha: 0.413418
##  beta : 0
##  gamma: 0.9561275
##
## Coefficients:
##             [,1]
## a   10.37661961
## b    0.02996319
## s1  -0.80952063
## s2  -0.60576477
## s3   0.01103238
## s4  -0.24160551
## s5  -0.35933517
## s6  -0.18076683
## s7   0.07788605
## s8   0.10147055
## s9   0.09649353
## s10  0.05197826
```


Holt−Winters filtering
Observed / Fitted
8
9
10
11
1988
1989
1990
1991
1992
1993
1994
Time


Forecasts from HoltWinters
14
13
12
11
10
9
8
1988
1990
1992
1994
1996
1998


Histogram of forecasterrors
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−1.0
−0.5
0.0
0.5
1.0
forecasterrors