

# Series Temporales y Predicción

## Práctica 2

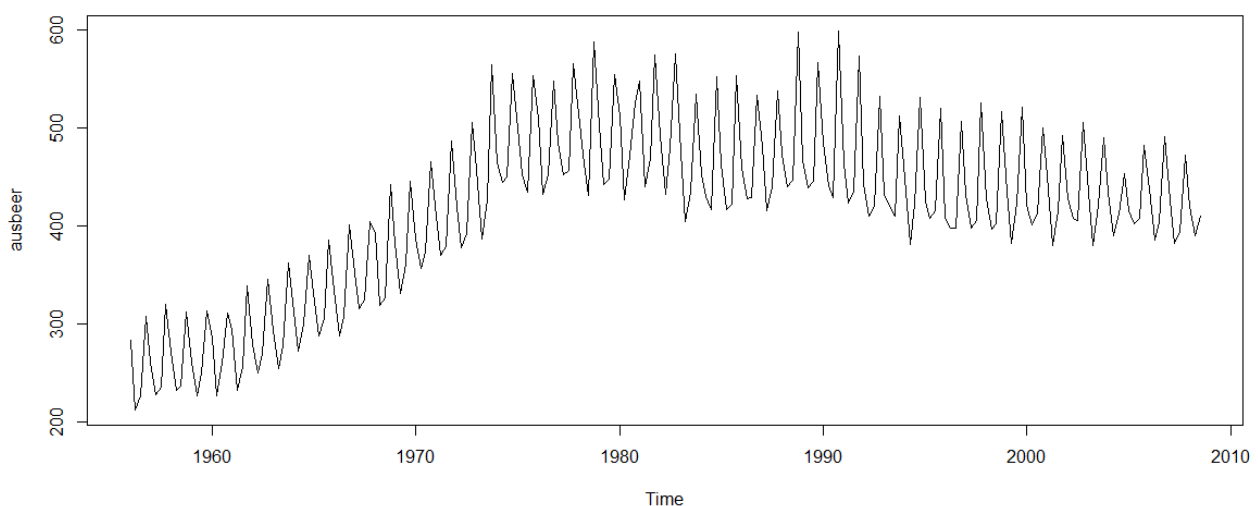
### Extracción de la tendencia y la estacionalidad con R

#### Solución orientativa a los ejercicios propuestos

##### Práctica 1.1

Ejecuta el siguiente código y pega a continuación la imagen de cómo viene representada la base de datos y el gráfico resultante.

```
install.packages("fpp");  
library(fpp)  
data(ausbeer)  
head(ausbeer)  
tail(ausbeer)  
plot.ts(ausbeer)
```



```
> head(ausbeer)  
      Qtr1 Qtr2 Qtr3 Qtr4  
1956  284  213  227  308  
1957  262  228  
> tail(ausbeer)  
      Qtr1 Qtr2 Qtr3 Qtr4  
2007     383  394  473  
2008  420  390  410
```

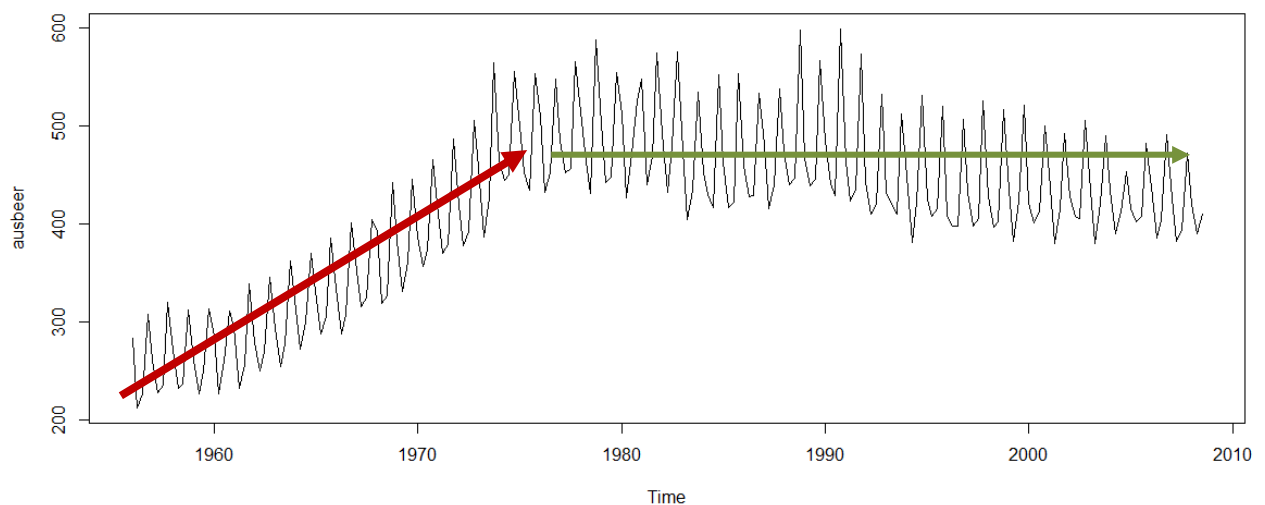
Se pide:

- ¿Qué periodicidad presentan los datos?  
Los datos tienen una periodicidad trimestral. La serie va desde el primer trimestre de 1956 al tercer trimestre de 2008.
- ¿La serie presenta estacionalidad?

Sí. La serie presenta un ciclo trimestral que se repite año tras año. Los picos anuales más altos se dan en el cuarto trimestre, año tras año.

iii. ¿Dirías que existe una tendencia creciente o decreciente en el tiempo?

La serie presenta una tendencia creciente los primeros años hasta llegar a un cierto nivel a partir del cual la serie mantiene cierta estabilidad, ni crece ni decrece.



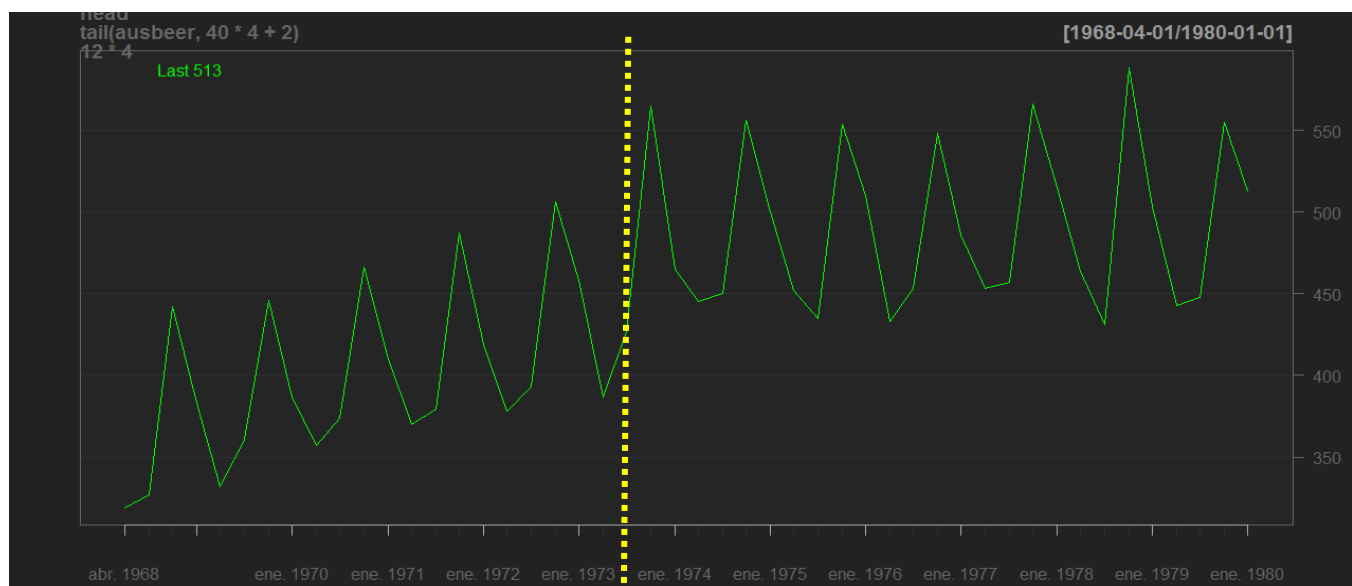
iv. ¿De qué año a qué año observas un cambio de tendencia?

Hay muchos caminos para responder a esta pregunta.

En el gráfico de la serie observamos que el cambio se produce entre 1970 y 1980. Si realizamos un zoom sobre estos años, visualizaremos mejor el año de cambio.

```
library(quantmod)
```

```
ggseasonplot(head(tail(ausbeer, 40*4+2),15*4))
```



El cambio de tendencia se produce a partir de 1974.

## Práctica 1.2

Para realizar el modelo de regresión en R de la serie `ausbeerrec` que hemos creado en el anterior apartado, se pide:

i. Pasar la serie a vector: `ausbeerrec_num`

- ii. Crear un vector de la misma longitud con los valores (1,2,...) que será nuestro vector tiempo.
- iii. Construid on R y pegad a continuación el modelo de regresión donde la variable ausbeerrec\_num es la variable explicada y t es la variable explicativa. Comentad los resultados.
- iv. Construid el gráfico de la serie añadiendo la recta de regresión obtenida en el anterior apartado. ¿Qué observas?
- v. Construid el gráfico de los residuos. ¿Qué observas?

### Solución orientativa

```
/* Apartado i.
ausbeerrec_num = unclass(ausbeerrec)
head(ausbeerrec_num)

/* Apartado ii.
t= (1:length(ausbeerrec_num))
t

/* Apartado iii.
ausbeerrec.lm=lm(ausbeerrec_num~t)
summary(ausbeerrec.lm)

/* Apartado iv.
beta=ausbeerrec.lm$coefficients
plot(t,ausbeerrec_num);
lines(t,beta[1]+beta[2]*t,col="orchid2",lwd=2)

/* Apartado v.
plot(t,ausbeerrec.lm$residuals,main="Noise process (residuals)");
abline(h=0)
```

i.

```
> ausbeerrec_num = unclass(ausbeerrec)
> head(ausbeerrec_num)
[1] 236 320 272 233 237 313
```

ii.

```
> t= (1:length(ausbeerrec_num))
> t
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[29] 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
[57] 57 58 59 60 61 62 63 64
```

iii.

```
Call:
lm(formula = ausbeerrec_num ~ t)

Residuals:
    Min       1Q   Median       3Q      Max
-54.90 -32.61 -11.37  33.87  85.70

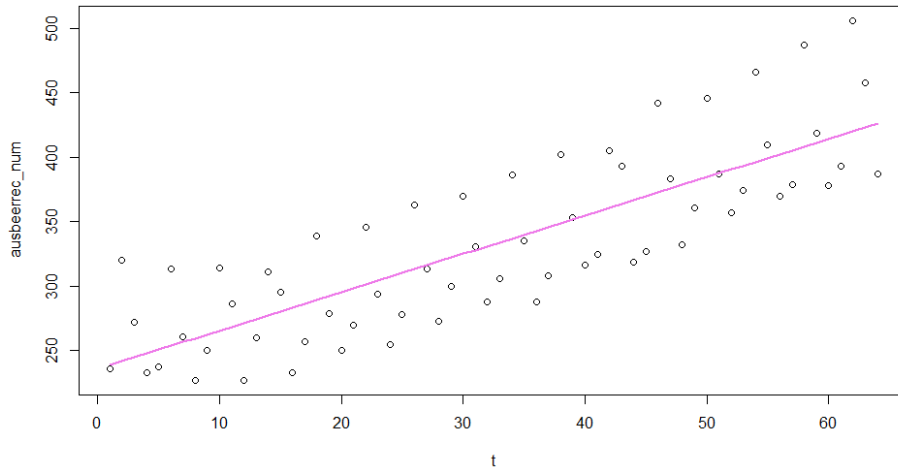
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  235.7336    10.1403   23.25  < 2e-16 ***
t              2.9769      0.2713   10.97 3.63e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40.09 on 62 degrees of freedom
Multiple R-squared:  0.6602,    Adjusted R-squared:  0.6547
F-statistic: 120.4 on 1 and 62 DF,  p-value: 3.629e-16
```

La recta de regresión obtenida es:  $Y_t = 235,7336 + 2,9769 * t + R_t$  donde  $R_t$  es el ruido.  
El ajuste es bueno puesto que:

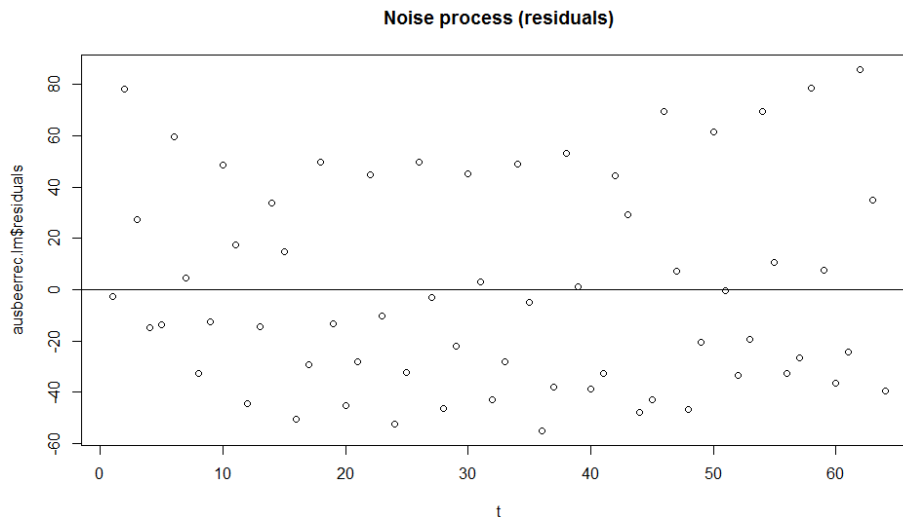
- 1 la constante y el coeficiente que acompaña a la variable  $t$  son significativos puesto que ambos p-valores son menores de 0,01.
- 2 Y el coeficiente de determinación de la recta es 66%; lo que supone que el tiempo explica un 66% de la variación de la serie ausbeerrec.

iv.



La recta de regresión si refleja la tendencia de la serie, pero se observa una pauta en los puntos alrededor de ella. Los puntos que están por encima de la recta, la gran mayoría se mantienen la distancia de separación a la recta ajustada parece mantenerse constante.

v.



Aunque los puntos estan dispersos, al igual que obserbávamos en la anterior gráfica, los residuos de la regresión parecen seguir un patrón que la regresión lineal no detecta: la estacionalidad.

## Práctica 1.3

Probar de suavizar la serie ausbeerrec con la función MA utilizando el orden 2, 4 y 10.

Se pide:

- i. Dad los resultados de los primeros registros de la serie y de la MA en orden 3 y replica el cálculo manualmente del primer registro obtenido en la MA(3).

- ii. Agregad todas las medias móviles, MA(2), MA(4) y MA(10) en el mismo gráfico con la serie original. ¿Qué orden crees que ajusta mejor?

Para realizar el ejercicio puedes ayudarte con el siguiente código:

---

```
trend_beer = ma(ausbeerrec, order = 4, centre = T)
plot(as.ts(ausbeerrec), col="navy",lwd=2)
lines(trend_beer, col="gold",lwd=3)
```

---

*Solución orientativa*

i.

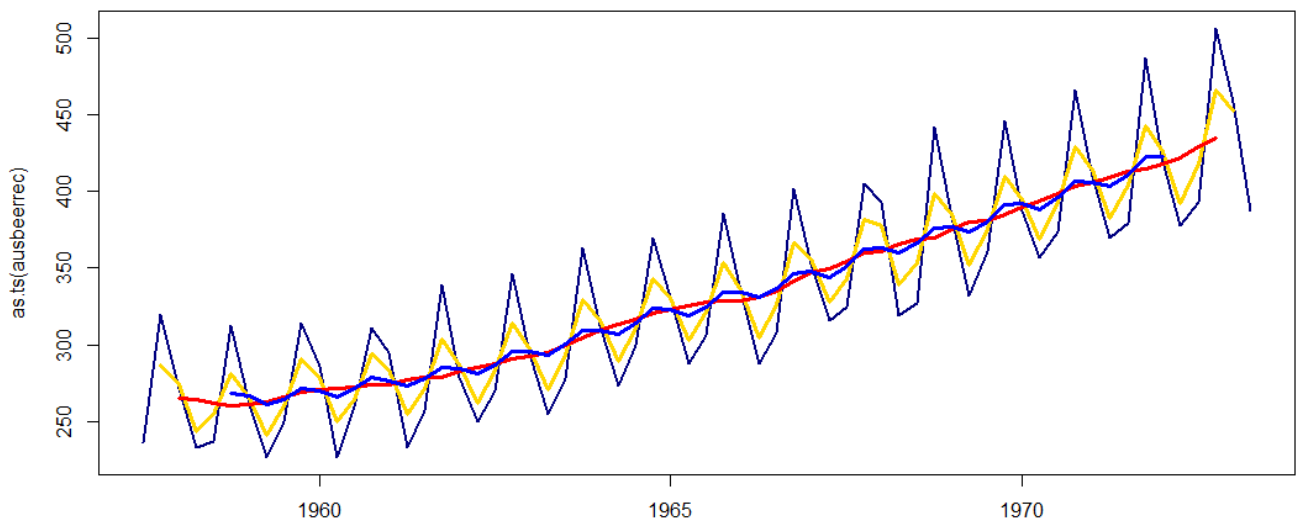
$$276 = 1/3 \cdot (236 + 320 + 272)$$

```
> trend_beer = ma(ausbeerrec, order = 3, centre = T)
> head(trend_beer)
      Qtr1      Qtr2      Qtr3      Qtr4
1957      NA 276.0000
1958 275.0000 247.3333 261.0000 270.3333
> ausbeerrec
      Qtr1 Qtr2 Qtr3 Qtr4
1957      236 320
1958 272 233 237 313
1959 261 227 250 314
1960 286 227 260 311
1961 295 233 257 339
1962 279 250 270 346
1963 294 255 278 363
1964 313 273 300 370
1965 331 288 306 386
1966 335 288 308 402
1967 353 316 325 405
1968 393 319 327 442
1969 383 332 361 446
1970 387 357 374 466
1971 410 370 379 487
1972 419 378 393 506
1973 458 387
```

ii.

---

```
trend_beer_2 = ma(ausbeerrec, order = 2, centre = T)
trend_beer_4 = ma(ausbeerrec, order = 4, centre = T)
trend_beer_10 = ma(ausbeerrec, order = 10, centre = T)
plot(as.ts(ausbeerrec), col="navy",lwd=2)
lines(trend_beer_2, col="gold",lwd=3)
lines(trend_beer_4, col="red",lwd=3)
lines(trend_beer_10, col="blue",lwd=3)
```



La media móvil de orden 4 (línea roja) es la que mejor suaviza la serie, coincidiendo el orden que mejor ajusta con el orden de la estacionalidad.

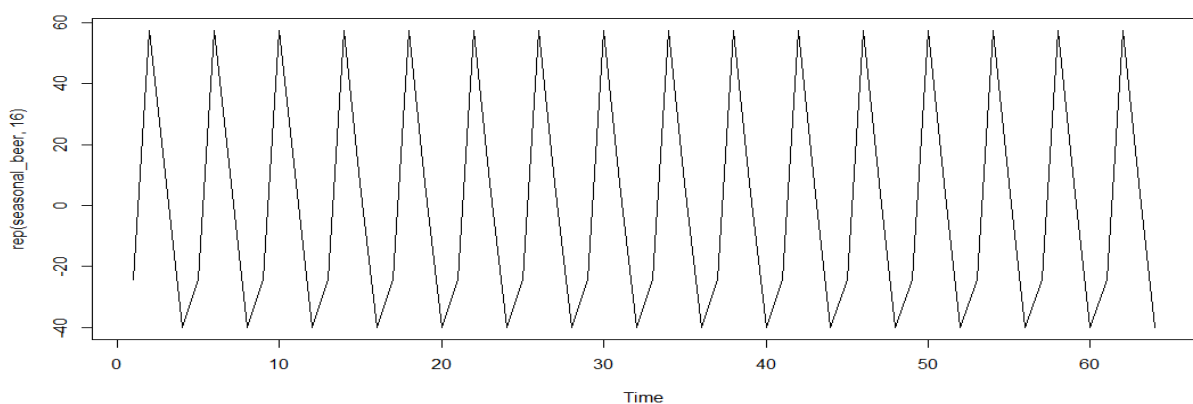
## Práctica 1.4

Para la serie `detrend_beer`, dad para cada trimestre la componente estacional como el valor promedio de cada trimestre. ¿En cuál de los cuatro trimestres el componente estacional es más alto? ¿Y el más bajo? ¿Tiene sentido con lo observado en la gráfica de la serie? Razonad vuestra respuesta.

### Solución orientativa

Ejecutamos el siguiente código y obtenemos las siguientes componentes estacionales:

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



```
> seasonal_beer = colMeans(m_beer, na.rm = T)
> seasonal_beer
[1] -24.51667  57.38333  7.17500 -40.01667
```

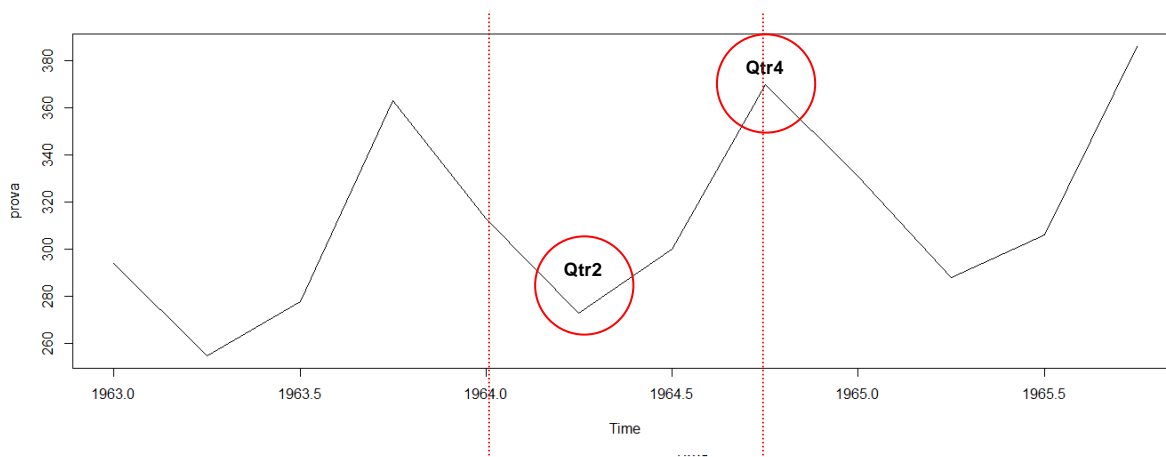
Ojo! Las columnas han cambiado con el código utilizado:

```
> m_beer = t(matrix(data = detrend_beer, nrow = 4))
> m_beer
      [,1] [,2] [,3] [,4]
[1,]    NA    NA 6.625 -31.625
[2,] -25.375 52.750 -0.125 -35.875
[3,] -16.125 44.750 15.500 -44.375
[4,] -12.125 37.000 20.625 -44.500
[5,] -22.000 59.875 -3.875 -35.375
[6,] -18.125 55.375 1.750 -40.375
[7,] -21.875 58.500 3.500 -40.125
[8,] -16.250 49.625 8.000 -37.750
[9,] -22.250 57.250 6.000 -43.250
[10,] -27.500 60.750 6.125 -33.375
[11,] -29.750 44.875 32.250 -46.625
[12,] -42.000 72.625 7.750 -48.000
[13,] -20.000 61.375 -2.375 -36.500
[14,] -24.875 62.625 4.375 -38.875
[15,] -33.625 72.250 1.500 -43.625
[16,] -35.875 71.125    NA    NA
> detrend_beer
      Qtr1  Qtr2  Qtr3  Qtr4
1957      NA      NA      NA      NA
1958  6.625 -31.625 -25.375  52.750
1959 -0.125 -35.875 -16.125  44.750
1960 15.500 -44.375 -12.125  37.000
```

Las componentes estacionales son:

Qtr1: 7,175  
Qtr2: -40,01667  
Qtr3: -24,51667  
Qtr4: 57,38333

El componente estacional más elevado es el del cuarto trimestre (57.38) y el más bajo es el segundo trimestre (-40.02) correspondiéndose con lo observado en la gráfica donde los picos más altos siempre se observan en el cuarto trimestre y los picos más bajos en el segundo trimestre.



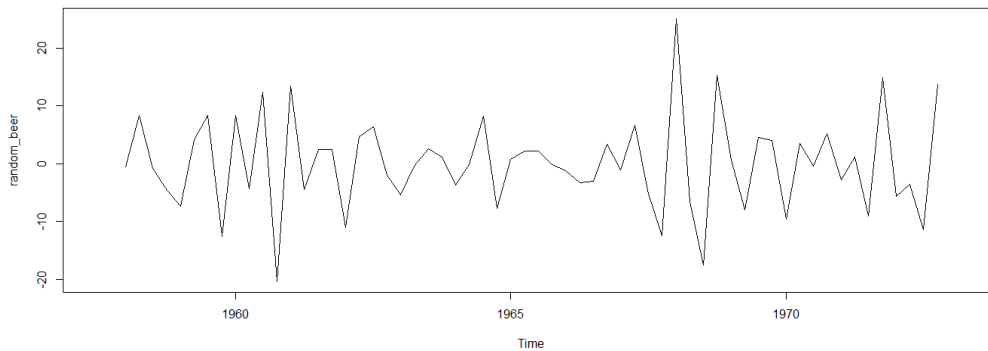
## Práctica 1.5

Para la serie [ausbeerrec](#), dibuja el ruido aleatorio que aparece al descomponer la serie en su media móvil de orden 4 ([trend\\_beer\\_4](#)) y las componentes estacionales ([seasonal\\_beer](#)) obtenidas en los ejercicios prácticos anteriores.

Comprobad manualmente que la primera componente del primer trimestre de la serie random beer se obtiene mediante la fórmula:

```
random_beer = ausbeerrec - trend_beer_4 - seasonal_beer
```

```
random_beer = ausbeerrec - trend_beer_4 - seasonal_beer  
plot.ts(random_beer)
```



```
> head(ausbeerrec)
      Qtr1 Qtr2 Qtr3 Qtr4
1957      236  320
1958 272  233  237  313
> head(trend_beer_4)
      Qtr1      Qtr2      Qtr3      Qtr4
1957      NA      NA
1958 265.375 264.625 262.375 260.250
> head(seasonal_beer)
[1] -24.51667  57.38333  7.17500 -40.01667
> head(random_beer)
      Qtr1      Qtr2      Qtr3      Qtr4
1957      NA      NA
1958 -0.550000  8.391667 -0.858333 -4.633333
```

$$272 = 265,375 + 7,175 - 0,55$$

## Práctica 1.6

Utilizad la función `decompose` para descomponer la serie `ausbeerrec`. Dad los valores de los componentes estacionales que calcula la función. ¿Son los mismos que hemos calculado en el anterior ejercicio práctico?

*Solución orientativa*

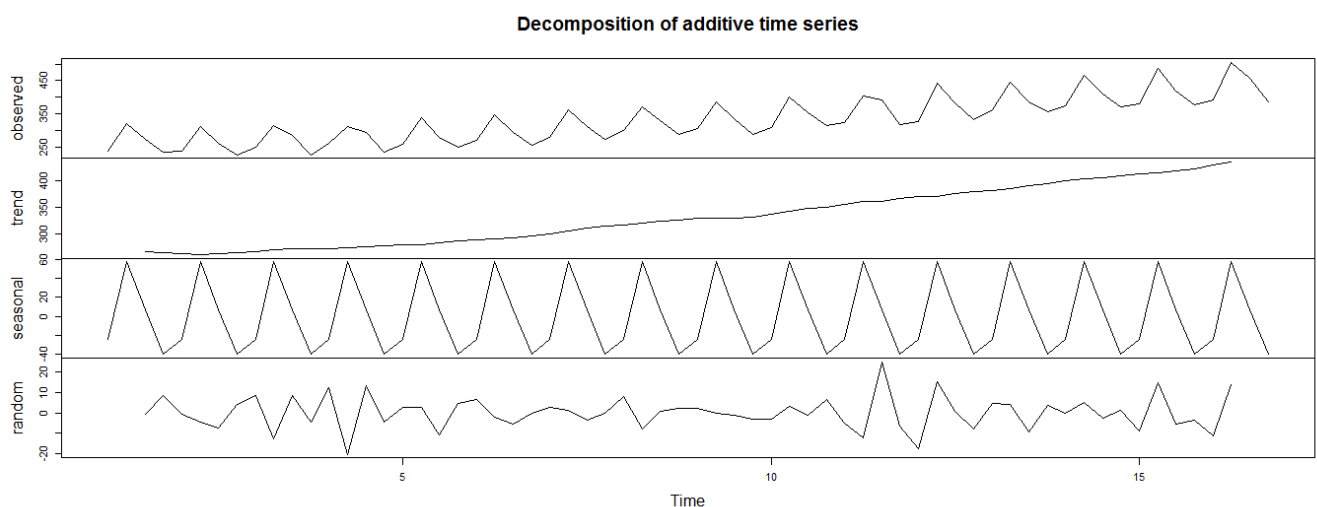
```
# Usando la funcion decompose: (frequency = 4 obs/año)
```



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

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

```
head(decompose_beer$seasonal)
head(seasonal_beer)
head(decompose_beer$trend)
head(trend_beer_4)
```



Las componentes estacionales obtenidas son ligeramente parecidas a las obtenidas en el anterior ejercicio. La media móvil está calculada de idéntico modo.

```
> head(decompose_beer$seasonal)
      Qtr1      Qtr2      Qtr3      Qtr4
1 -24.52292  57.37708   7.16875 -40.02292
2 -24.52292  57.37708
> head(seasonal_beer)
[1] -24.51667  57.38333   7.17500 -40.01667
> head(decompose_beer$trend)
      Qtr1      Qtr2      Qtr3      Qtr4
1      NA      NA 265.375 264.625
2 262.375 260.250
> head(trend_beer_4)
      Qtr1      Qtr2      Qtr3      Qtr4
1957      NA      NA      NA      NA
1958 265.375 264.625 262.375 260.250
> |
```

## Práctica 1.7

Utilizad la función `stl` para descomponer la serie `ausbeerrec`. Buscad la ayuda de la función en `help` para conocer la descomposición. Dad los valores de los componentes estacionales que calcula la función. ¿Son los mismos que hemos calculado en el anterior ejercicio práctico?

*Solución orientativa*

### Descomposicion usando la funcion stl (ver el help!)

```
ts_beer = ts(beerrec, frequency = 4)
stl_beer = stl(ts_beer, "periodic")
names(stl_beer)
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.ts(seasonal_stl_beer)
plot(trend_stl_beer)
plot(random_stl_beer)
plot(stl_beer)
```

← → ↺ <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/stl> RDocumentation  [R Enterprise Training](#) [R package](#) [Leaderboard](#) [Sign in](#)

# stl

From [stats v3.6.2](#)  
by [R-core](#) [R-core@R-project.org](#) 99.99th Percentile

## Seasonal Decomposition Of Time Series By Loess

Decompose a time series into seasonal, trend and irregular components using [loess](#), acronym STL.

[r-statistics.co](#) by Selva Prabhakaran

### Tutorial

[R Tutorial](#)

### ggplot2

[ggplot2 Short Tutorial](#)  
[ggplot2 Tutorial 1 - Intro](#)  
[ggplot2 Tutorial 2 - Theme](#)  
[ggplot2 Tutorial 3 - Masterlist](#)  
[ggplot2 Quickref](#)

### Foundations

[Linear Regression](#)  
[Statistical Tests](#)  
[Missing Value Treatment](#)  
[Outlier Analysis](#)  
[Feature Selection](#)

Loess Regression is the most common method used to smoothen a volatile time series. It is a non-parametric methods where least squares regression is performed in localized subsets, which makes it a suitable candidate for smoothing any numerical vector.

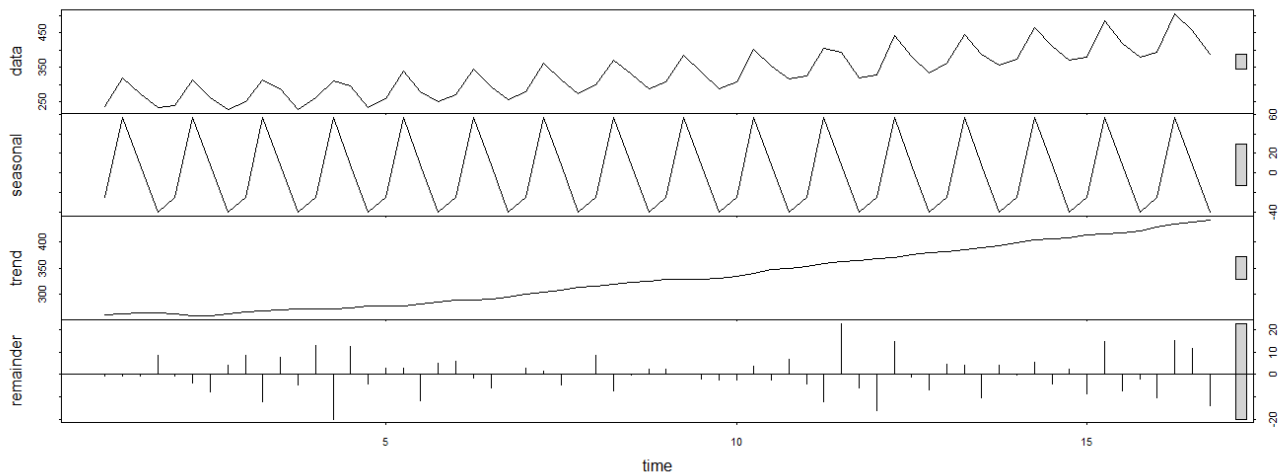
## Introduction

Loess short for Local Regression is a non-parametric approach that fits multiple regressions in local neighborhood. This can be particularly resourceful, if you know that your  $X$  variables are bound within a range.

Loess regression can be applied using the `loess()` on a numerical vector to smoothen it and to predict the  $Y$  locally (i.e, within the trained values of  $X$ s). The size of the neighborhood can be controlled using the `span` argument, which ranges between 0 to 1. It controls the degree of smoothing. So, the greater the value of `span`, more smooth is the fitted curve.

The predictor variable can just be indices from 1 to number of observations in the absence of explanatory variables. If other explanatory variables are available, they can be used as well (maximum of 4).

Ad



```
> head(seasonal_stl_beer)
      Qtr1      Qtr2      Qtr3      Qtr4
1 -24.964744  57.179168   8.404196 -40.618622
2 -24.964744  57.179168
> head(seasonal_beer)
[1] -24.51667  57.38333   7.17500 -40.01667
> |
```

Las componentes estacionales obtenidas son ligeramente parecidas a las obtenidas en el anterior ejercicio.

## Práctica 2.1

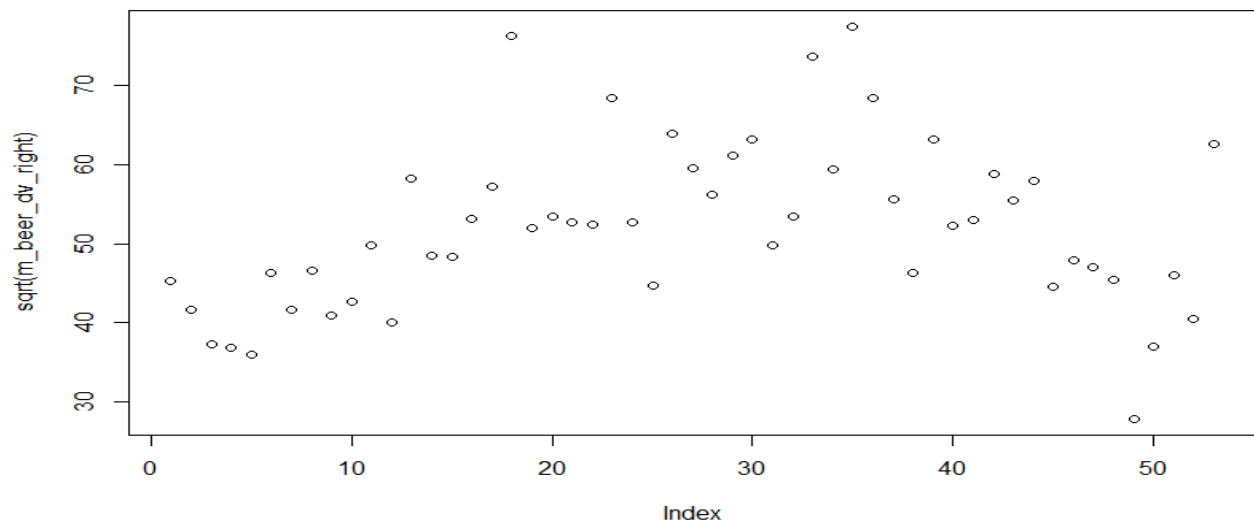
Calculad las desviaciones típicas de los cuatro trimestres para cada año de las series *ausbeer* y *AirPassengers* (trabajada en la práctica 1) y dibuja las dos series de desviaciones. Si la serie anual de las desviaciones típicas se mantiene estable se deduce una descomposición aditiva. Si la serie anual crece o decrece se deduce una descomposición multiplicativa. ¿Qué corresponde a cada una de nuestras dos series?

Si ejecutáis el siguiente código se obtiene la gráfica de las desviaciones típicas de la serie *ausbeer* donde los puntos no parecen tener una tendencia ni claramente creciente ni decreciente; por lo que una descomposición aditiva parece adecuada para esta serie.

---

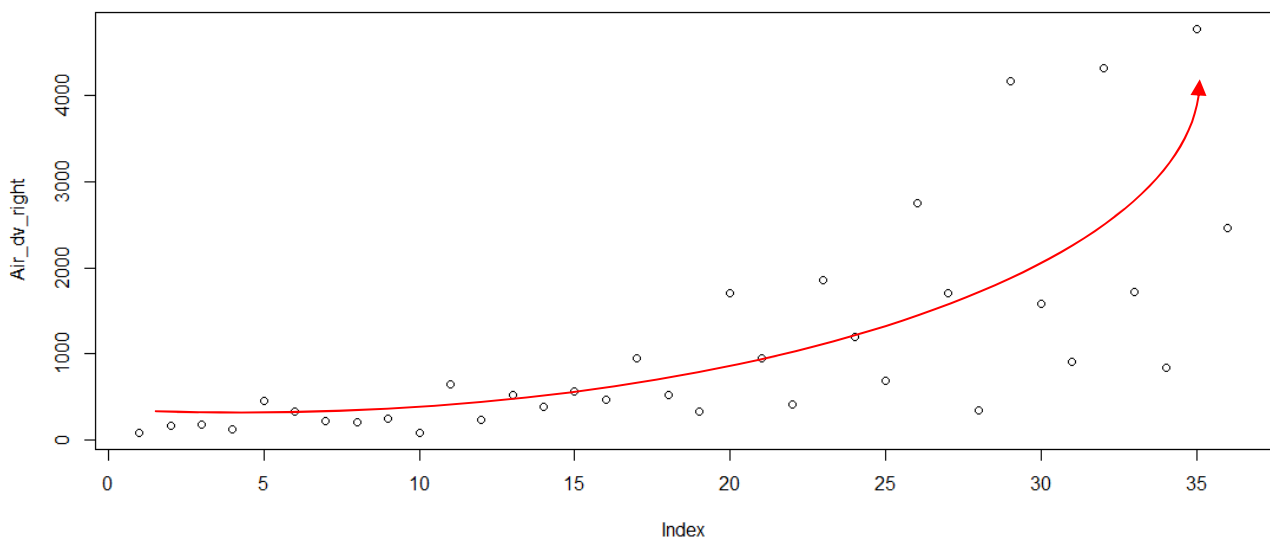
```
library(fpp)
data(ausbeer)
m_beer_dv = matrix(data = ausbeer, nrow = 4)
m_beer_dv_right = (colSums((m_beer_dv*m_beer_dv), na.rm = T)- (colMeans(m_beer_dv,
na.rm = T)^2)*4)/3
plot(sqrt(m_beer_dv_right))
```

---



Si ejecutáis el siguiente código se obtiene la gráfica de las desviaciones típicas de la serie AirPassengers donde los puntos presentan una tendencia claramente creciente; por lo que una descomposición multiplicativa parece adecuada para esta serie.

```
data(AirPassengers)
m_Air_dv = matrix(data = AirPassengers, nrow = 4)
m_Air_dv_right = (colSums((m_Air_dv*m_Air_dv), na.rm = T)- (colMeans(m_Air_dv, na.rm = T)^2)*4)/3
plot(sqrt(m_Air_dv_right))
```



## Práctica 2.2

Realizad los mismos pasos que los realizados con la serie ausbeerc para la serie AirPassengers teniendo en cuenta que la descomposición es multiplicativa.

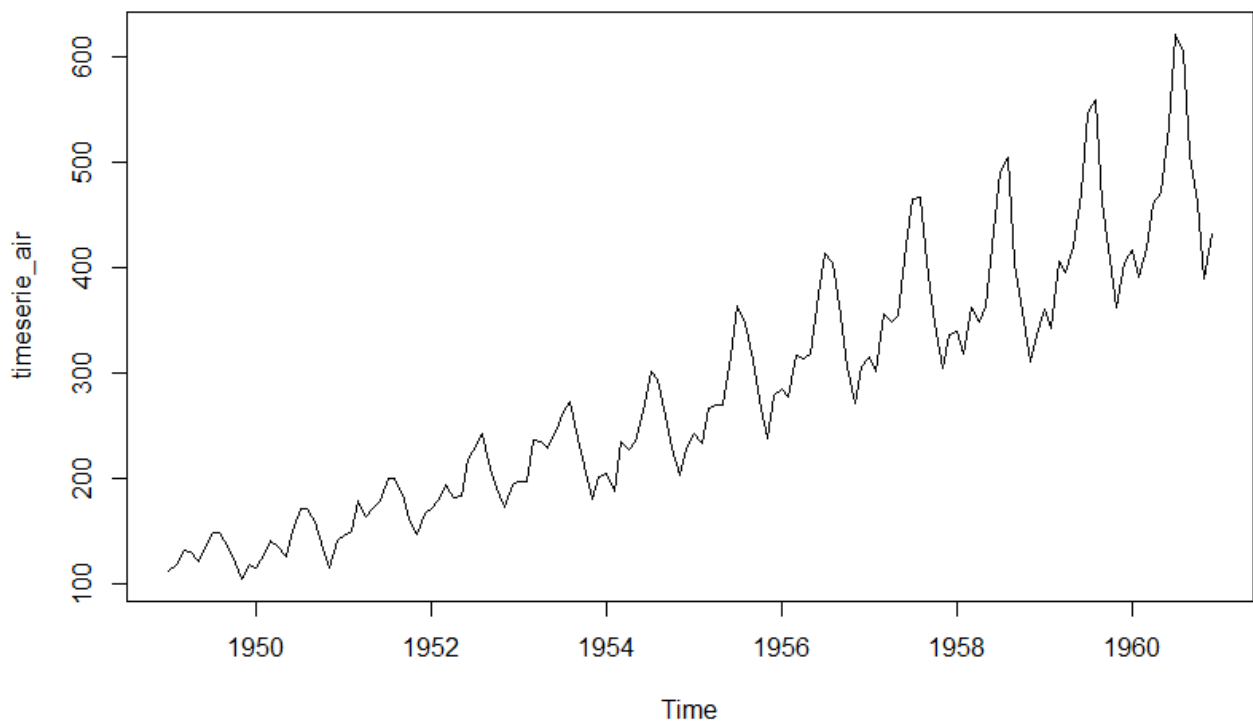
*Solución orientativa*

**Paso 1** Obtener y mirar los datos

---

```
data(AirPassengers)
timeserie_air = AirPassengers
plot.ts(timeserie_air)
```

---



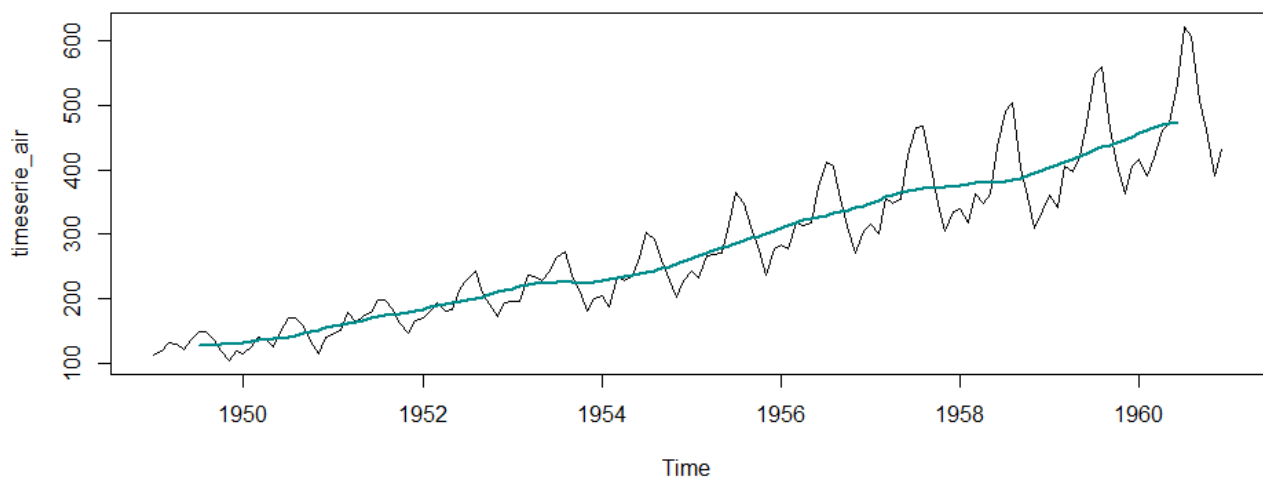
## Paso 2 Detectar la tendencia (Trend)

El proceso es idéntico que para el modelo aditivo. La estacionalidad es anual, y como en este caso tenemos observaciones mensuales, elegimos una ventana de tamaño 12 para el promedio móvil.

---

```
trend_air = ma(timeserie_air, order = 12, centre = T)
plot.ts(timeserie_air)
lines(trend_air,col="darkcyan",lwd=2)
plot.ts(trend_air)
```

---



## Paso 3 Quitar la tendencia (Trend)

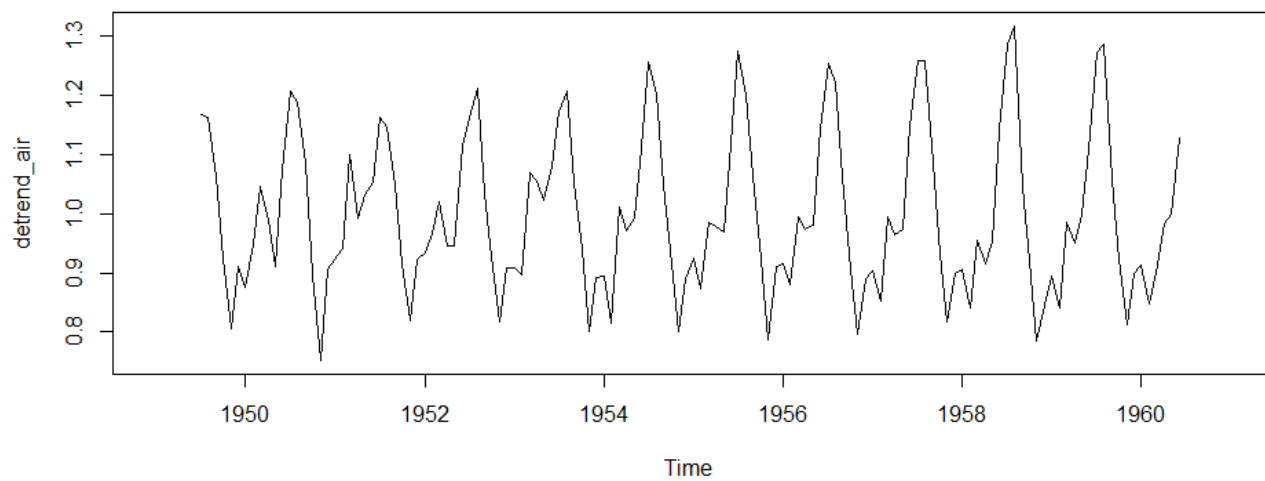
Al quitar la tendencia de la serie, en este caso, dividiéndola, obtendremos una nueva serie que muestra más claramente la componente estacional.

Si ejecutamos el siguiente código obtenemos la nueva serie sin tendencia:

---

```
detrend_air = timeserie_air / trend_air
plot.ts(detrend_air)
```

---



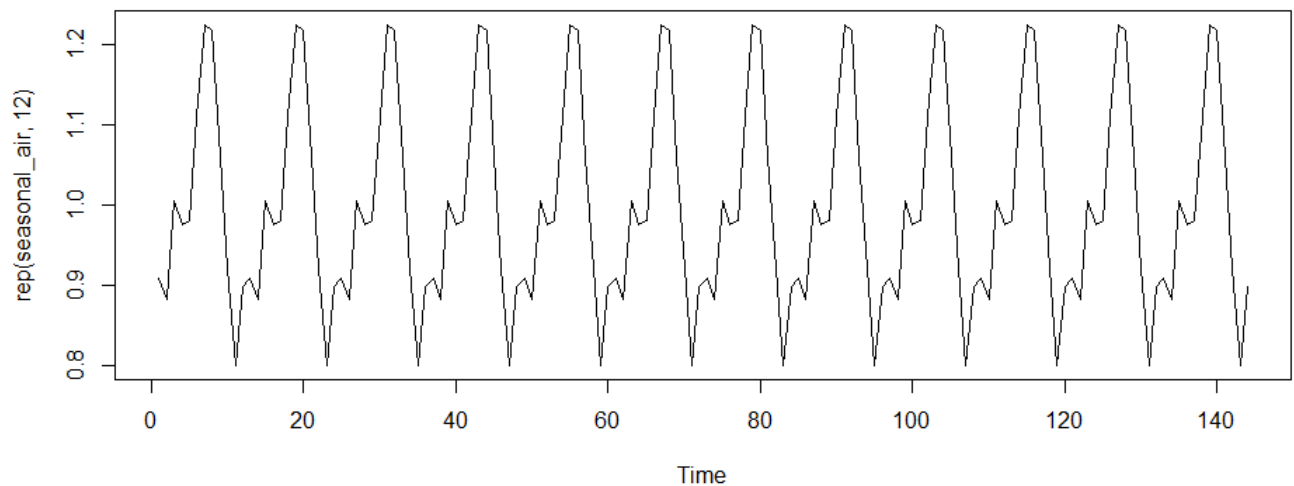
#### Paso 4 Promediar la estacionalidad (Seasonal)

En este caso usaremos 12 columnas en la matriz, una por cada mes.

---

```
m_air = t(matrix(data = detrend_air, nrow = 12))
seasonal_air = colMeans(m_air, na.rm = T)
plot.ts(rep(seasonal_air, 12))
```

---



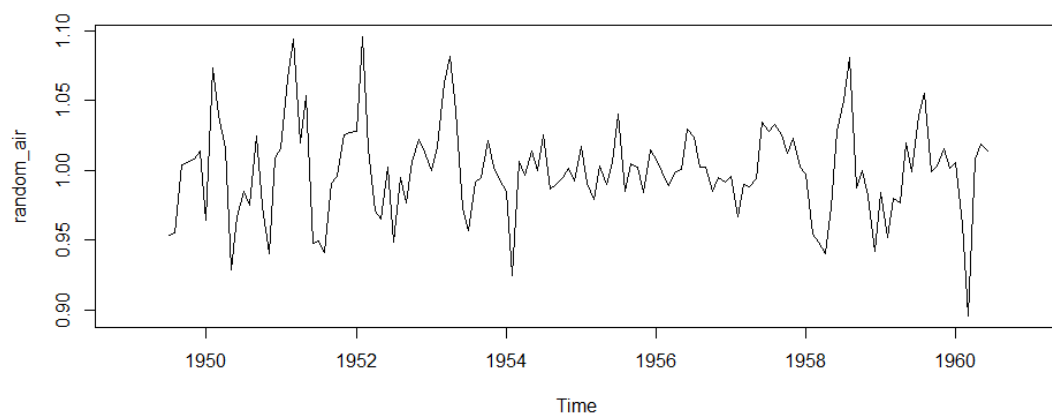
#### Paso 5 Examinar el ruido restante

Quitar a la serie original la tendencia y la estacionalidad, en este caso, dividiendo, para obtener el ruido.

---

```
random_air = timeserie_air / (trend_air * seasonal_air)
plot.ts(random_air)
```

---



También podemos quitar solamente la estacionalidad, y en ese caso obtenemos la llamada serie ajustada por estacionalidad, observar que en este caso, en 1960 se observa una anomalía que no era detectable a ojo a partir de los datos originales.

---

```
sadjust_air = timeserie_air / seasonal_air  
plot.ts(sadjust_air, col = "orchid", lwd = 2)
```

---

