

# Introducción a la Econometría Financiera

Javier Sánchez García  
Salvador Cruz Rambaud

## Introducción a la Econometría Financiera

texto:

Javier Sánchez García  
Salvador Cruz Rambaud

Textos Docentes n.º 181

edición:

Editorial Universidad de Almería, 2024

[editorial@ual.es](mailto:editorial@ual.es)

[www.ual.es/editorial](http://www.ual.es/editorial)

Telf/Fax: 950 015459

α

ISBN: 978-84-1351-351-5



Esta obra se edita bajo una licencia Creative Commons  
CC BY-NC-SA (Atribución-NoComercial-Compartirigual) 4.0 Internacional



En este libro puede volver al índice  
pulsando el pie de la página

# Tabla de contenidos

<b>Prefacio</b>	<b>5</b>
<b>1 Introducción a las series temporales</b>	<b>6</b>
1.1 Gráfica de series temporales . . . . .	7
1.2 Aspectos avanzados de las series temporales . . . . .	10
1.3 Tipos de gráficos para analizar series temporales . . . . .	13
1.3.1 Gráfico estacional . . . . .	14
1.3.2 Correlación entre series . . . . .	15
<b>2 Series temporales estacionarias</b>	<b>19</b>
2.1 Tendencias estocásticas . . . . .	19
2.1.1 Tratamiento de una tendencia estocástica . . . . .	20
2.1.2 Diferenciación estacional . . . . .	22
2.2 Estacionareidad . . . . .	23
2.2.1 Estacionareidad en varianza . . . . .	24
2.2.2 Autocorrelación . . . . .	27
2.2.3 Contrastes de estacionareidad . . . . .	29
<b>3 Análisis de regresión</b>	<b>30</b>
3.1 Regresión simple . . . . .	31
3.2 Regresión espuria . . . . .	37
3.3 Regresión múltiple . . . . .	39
3.3.1 Interpretación de coeficientes . . . . .	43
3.3.2 Evaluación de un modelo econométrico . . . . .	44
3.4 Predicción con el modelo de regresión . . . . .	47
3.4.1 Dummies estacionales . . . . .	50
3.4.2 Predicción de escenarios . . . . .	51
<b>4 Modelos econométricos basados en procesos estocásticos</b>	<b>54</b>
4.1 Proceso autorregresivo . . . . .	54
4.2 Proceso de medias móviles . . . . .	57
4.3 Modelo ARIMA . . . . .	58
4.3.1 Diagnóstico del modelo . . . . .	59
4.3.2 Componente estacional . . . . .	61
4.4 Proceso de ruido blanco . . . . .	62
4.5 Paseo aleatorio . . . . .	63

4.6	Predicción con el modelo ARIMA . . . . .	65
4.6.1	Evaluación de las predicciones . . . . .	67
<b>5</b>	<b>Modelizando volatilidad</b>	<b>70</b>
5.1	Modelo GARCH . . . . .	70
5.2	t-GARCH . . . . .	77
5.3	Modelos GARCH multivariantes . . . . .	81
<b>6</b>	<b>Cointegración y shocks</b>	<b>88</b>
6.1	Cointegración . . . . .	88
6.2	Modelo ADL . . . . .	92
6.3	El modelo VAR . . . . .	93
6.3.1	Modelos VECM . . . . .	100

## Prefacio

La econometría financiera es una disciplina que a una la utilización de modelos estadísticos y matemáticos con la programación computacional y el estudio riguroso de instrumentos financieros. Como subdisciplina de las finanzas cuantitativas, emplea un trilingüismo como enfoque natural, alternando lenguaje común con código y matemáticas. Este es precisamente el enfoque empleado en este manual, cuyo objetivo es el de dar un tratamiento riguroso a la par que pragmático de la disciplina. Para ello, se alterna el español con la formulación matemática y el código en el lenguaje de programación en estadística computacional R. Para el correcto abordaje de este libro, se precisan conocimientos básicos de inferencia estadística, matemática aplicada, y finanzas. Conocimientos elementales de programación son un añadido importante pero no un requisito indispensable.

El *leit motif* de este libro es claramente el estudio del análisis temporales. La causa de esto es clara: gran parte de los instrumentos financieros operan diariamente en los mercados financieros, por lo que el estudio riguroso de las series temporales es por tanto una de las habilidades más demandadas en los trabajos de índole financiera en la industria y academia.

De esta forma, el primer capítulo del libro comienza ofreciendo una introducción a las series temporales. El segundo capítulo introduce aspectos fundamentales para la modelización econométrica de series temporales financieras, tales como las tendencias o los supuestos de estacionariedad. Una vez cubiertas las necesidades básicas indispensables en los dos capítulos anteriores, el tercer capítulo expone los aspectos fundamentales del análisis de regresión de series temporales.

Los tres primeros capítulos pueden pensarse como una introducción básica a la econometría financiera, mientras que los tres últimos son de corte más avanzado. En particular, el capítulo 4 muestra como emplear procesos estocásticos para modelizar series temporales, con a menudo óptimos resultados en ámbitos como la predicción del valor de instrumentos financieros. El quinto capítulo, por otra parte, introduce al lector al análisis cuantitativo del riesgo financiero mediante modelos para la volatilidad. Finalmente, el sexto capítulo muestra como modelizar shocks de diversos tipos, además de conceptos avanzados de comovimiento de series financieras tales como la cointegración.

A lo largo del libro, el contenido teórico se intercala con ejemplos prácticos para el refuerzo de la comprensión de los contenidos. Además, muchos capítulos cuentan con ejercicios a resolver por el lector, de forma que se asegure el correcto aprendizaje de los conocimientos.

# 1 Introducción a las series temporales

Los datos de series temporales permiten responder preguntas de inferencia y predicción en relación a una variable cuyo valor no varía entre sujetos, sino que varía conforme pasa el tiempo. Esto implica trabajar con variables del tipo

$$X_t : X \in \mathbb{R}, t = 1, \dots, n$$

Es decir, la variable  $X$  toma un valor de los números reales para cada periodo  $t$ , que varía desde el primero (1) hasta el último ( $n$ ).

Para trabajar con R en series temporales, se usa la función `ts()`

```
x = data.frame(date=c(2000,2001,2002,2003,2004), z=c(4, 7, 21, 54, 64))
y = ts(x)
class(x)
```

```
[1] "data.frame"
```

```
class(y)
```

```
[1] "mts"      "ts"      "matrix"
```

```
head(y)
```

```
      date z
[1,] 2000 4
[2,] 2001 7
[3,] 2002 21
[4,] 2003 54
[5,] 2004 64
```

Si los datos son anuales, R solo necesita la fecha de inicio.

```
y=ts(c(4, 7, 21, 54, 64), start = 2000)
head(y)
```

```
[1] 4 7 21 54 64
```

En caso de que las observaciones de la variable se produzcan en una frecuencia distinta a la anual, basta con añadir el comando `frequency`

```
x=rnorm(1200, mean=50, sd=20)
y=ts(x, start=2000, frequency=12)
summary(y)
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-8.978  35.662  50.334  49.638  62.902 116.484
```

De forma que frecuencias menores a la anual se especifican teniendo en cuenta el número de veces ( $k$ ) que hay que dividir el año, o  $k$ -ésimos de año. Si la frecuencia es mensual, el año se divide en 12 meses ( $k = 12$ ), si es trimestral, el año se divide en 4 trimestres ( $k = 4$ ), y así sucesivamente.

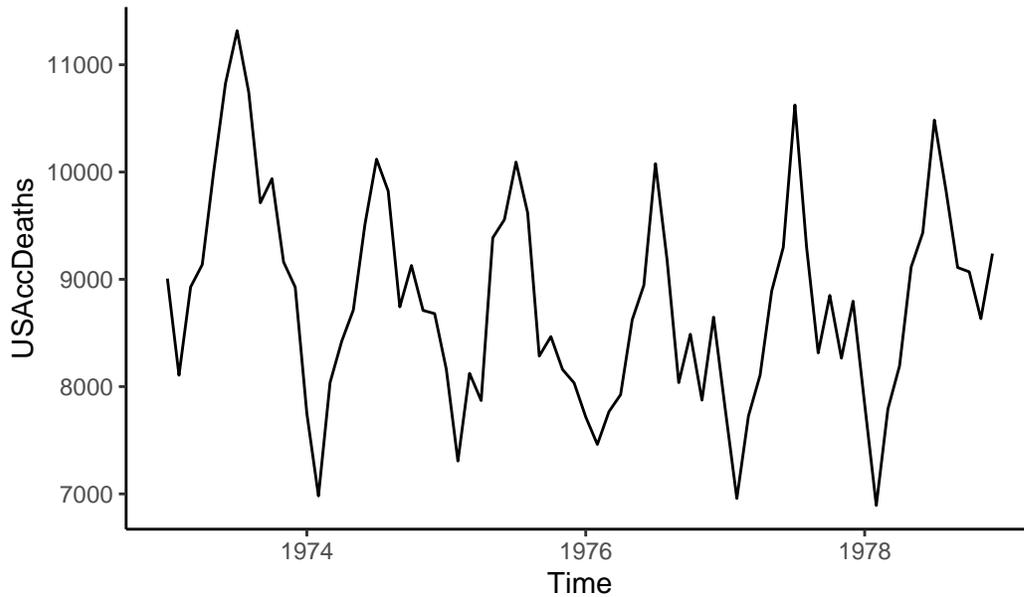
Frecuencia	$k$
Anual	1
Semestral	2
Trimestral	3
Cuatrimestral	4
Mensual	12
Semanal	52
Diaria	365

## 1.1 Gráfica de series temporales

El dicho popular “*una imagen vale más que mil palabras*” es perfectamente aplicable al caso de las variables de tipo serie temporal, donde el visualizar tablas de datos es poco informativo. La opción más útil en estos casos es empezar el análisis con un gráfico de serie temporal, que tenga el valor de la variable en el eje y, y el paso del tiempo en el eje x.

La librería `forecast` es especialmente útil para este tipo de gráficos. Si la variable es de series temporales, el comando `autoplot` producirá un gráfico de series temporales.

```
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
data("USAccDeaths")
autoplot(USAccDeaths) + theme_classic()
```



El gráfico de series temporales arroja mucha más información que sus estadísticos descriptivos. Por ejemplo, es fácilmente interpretable que el pico de muertos por accidentes se produce en el primer año de la serie, que sigue a una caída drástica en el número de muertes por accidentes.

Además, es fácilmente percible el hecho de que, a mitad de año, el número de muertes por accidentes aumenta sustancialmente, para después caer a finales del año y principios del siguiente. A mediados del año la estación del año es verano, fecha en la que los trabajadores suelen tomar vacaciones y aumentan actividades como el baño en playas y piscinas que pueden dar lugar a ahogamientos. También aumentan viajes por carretera que pueden dar lugar a accidentes. A finales del año y principios del siguiente, es invierno, fecha en la que se reducen considerablemente tanto el baño como el desplazamiento. Esto es lo que se denomina como *patrón estacional* o simplemente *estacionalidad*, haciendo evidente que el comportamiento de la serie depende de la estación del año en que se encuentre. Otros ejemplos claros de patrones estacionales tienen que ver con las temperaturas o la lluvia.

En el paquete `Forecast`, además de en otros paquetes como `Ecdat`, se pueden encontrar muchos conjuntos de datos disponibles para el análisis. Sin embargo, el primero se caracteriza por contener variables de series temporales, como por ejemplo `EuStockMarkets`.

```
data("EuStockMarkets")
head(EuStockMarkets)
```

Time Series:

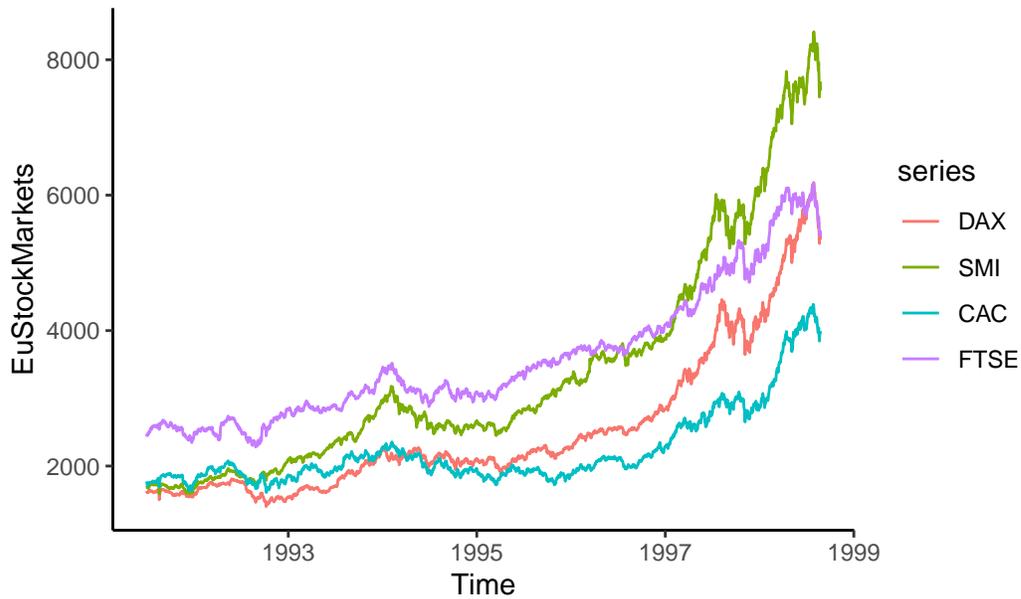
Start = c(1991, 130)

End = c(1991, 135)

Frequency = 260

	DAX	SMI	CAC	FTSE
1991.496	1628.75	1678.1	1772.8	2443.6
1991.500	1613.63	1688.5	1750.5	2460.2
1991.504	1606.51	1678.6	1718.0	2448.2
1991.508	1621.04	1684.1	1708.1	2470.4
1991.512	1618.16	1686.6	1723.1	2484.7
1991.515	1610.61	1671.6	1714.3	2466.8

```
autoplot(EuStockMarkets) + theme_classic()
```



Como se puede comprobar, si se tienen varias series temporales apiladas en un mismo conjunto de datos, autoplot representa cada una por separado.

Este tipo de gráficos es muy útil pues permite obtener mucha información de manera compacta. Por ejemplo:

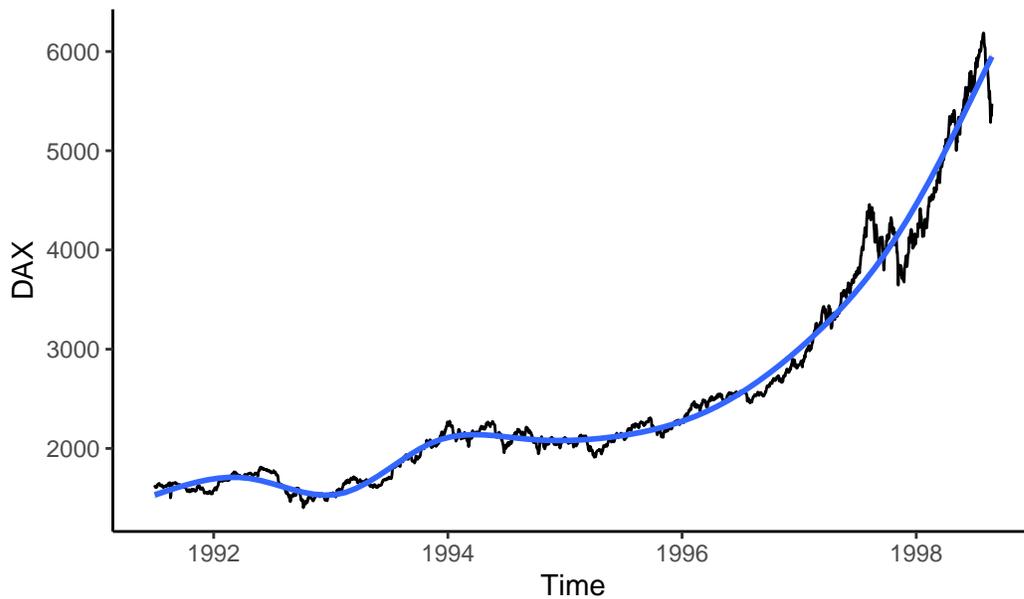
- El índice bursátil de acciones alemán (DAX) superó al francés (CAC) en 1994.
- Suiza (SMI) pasó de tener el penúltimo índice bursátil a tener el primero en valor alrededor de 1997.
- El crecimiento del índice bursátil inglés (FTSE) presenta una tendencia de tipo lineal y cierto declive, mientras que los índices suizo, alemán y francés presentan tendencias exponenciales de crecimiento. Esto podría implicar que los inversores interpretan que los últimos países experimentarán un gran crecimiento económico y, por tanto, prefieren depositar sus fondos en estos índices, mientras que esperan que el crecimiento económico de Inglaterra se estanque.
- Suiza (SMI) y Alemania (DAX) presentan patrones muy similares de crecimiento desde finales de 1997. Podría decirse que *co-mueven*.

## 1.2 Aspectos avanzados de las series temporales

Además del componente estacional, aquel componente repetitivo que afecta a la serie temporal de forma fija y conocida, existen otro tipo de componentes fundamentales de las series temporales.

La *tendencia* de una serie es su comportamiento a largo plazo. Se caracteriza por un crecimiento o decrecimiento lineal o no lineal. Se denota con  $T$ . En el contexto del modelo de regresión lineal, es posible capturar esta tendencia incorporando el parámetro  $T$ , el cual es igual a 1 en la primera observación, a 2, en la segunda, etc., con la respectiva forma funcional que corresponda (lineal, polinómica, exponencial, etc.).

```
library(ggplot2, quietly=TRUE)
forecast::autoplot(EuStockMarkets[,1]) + geom_smooth() + ylab("DAX") +
  theme_classic()
```



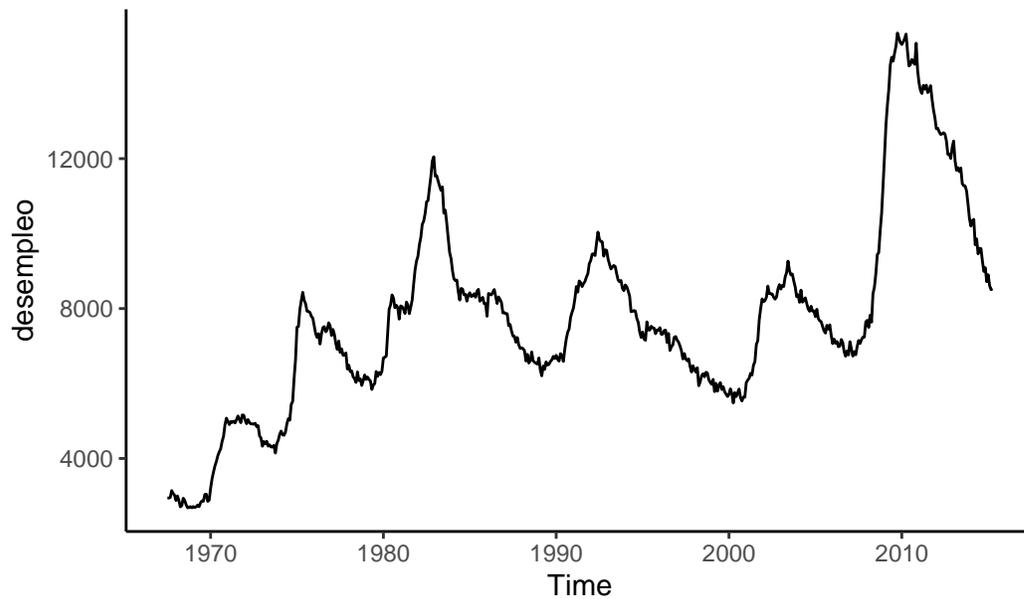
Se aprecia como la serie presenta una tendencia creciente al comienzo, una decreciente desde 1992 a 1993, una creciente desde 1993 a 1994, una relativamente estable entre 1994 y 1996, y un crecimiento exponencial desde 1996 hacia adelante.

El *ciclo* de una serie ocurre cuando esta presenta oscilaciones crecientes y decrecientes que no tienen una frecuencia fija, es decir, que no son de carácter estacional. A menudo suceden en series temporales económicas y están relacionadas con lo que se conoce como “ciclo económico”. Existe una popular descomposición, denominada la descomposición de Hodrick-Prescott, por la cual una serie se puede representar como

$$Y = T + C + \epsilon$$

siendo  $Y$  la serie,  $T$  la tendencia,  $C$  el ciclo y  $\epsilon$  el término error.

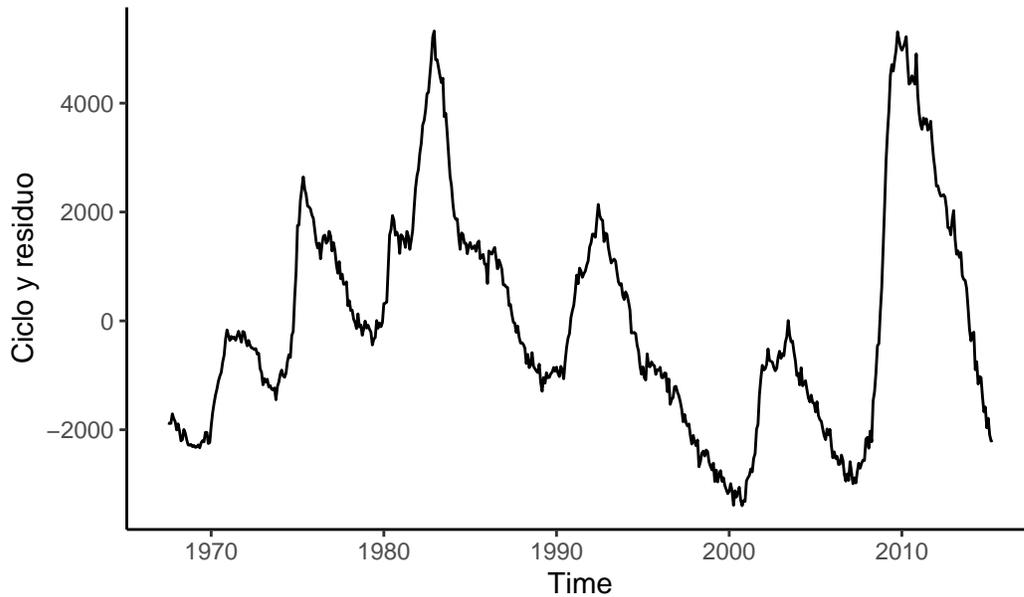
```
data(economics)
desempleo <- ts(economics$unemploy, economics$date, start=c(1967,7),
               end=c(2015,4), frequency=12)
autoplot(desempleo) + theme_classic()
```



Parece que la serie incorpora un componente cíclico bastante claro. Está claro que a los periodos de aumento de desempleados le siguen periodos de disminución en consonancia con las circunstancias económicas EEUU. No obstante, estos intervalos no siguen un patrón regular como en el caso del componente estacional. Por lo tanto, hablamos de ciclos, y no de estaciones del año.

Por otra parte, parece que el número de desempleados sigue una tendencia creciente a largo plazo. Esto es lógico si se tiene en cuenta el crecimiento poblacional (a más población, más desempleados y empleados). Vamos a capturar la tendencia a largo plazo con un modelo de regresión, y a representar el residuo, que corresponderá a los términos  $C$  y  $\epsilon$  de la descomposición HP.

```
modelo <- lm(unemploy ~ date, data = economics)
residuos <- residuals(modelo)
serie <- ts(residuos, economics$date, start=c(1967,7), end=c(2015,4),
           frequency=12)
forecast::autoplot(serie) + ylab("Ciclo y residuo") + theme_classic()
```



Se aprecia claramente el componente cíclico de la serie de desempleados. El número de desempleados aumenta sustancialmente en:

- 1970-1980: Crisis del petróleo.
- 1980-1985: Segunda crisis del petróleo e inflación.
- 1990-1993: Recesión global de los primeros años de la década de 1990.
- 2001-2003: Burbuja financiera dot.com.
- 2007-2010: Crisis financiera o gran recesión.

lo que se corresponde con los periodos de recesión, crisis o contracción del ciclo económico. Por contra, el número de desempleados desciende con fuerza en los periodos anteriores y posteriores a los periodos de crisis, en lo que se conoce como periodos de expansión.

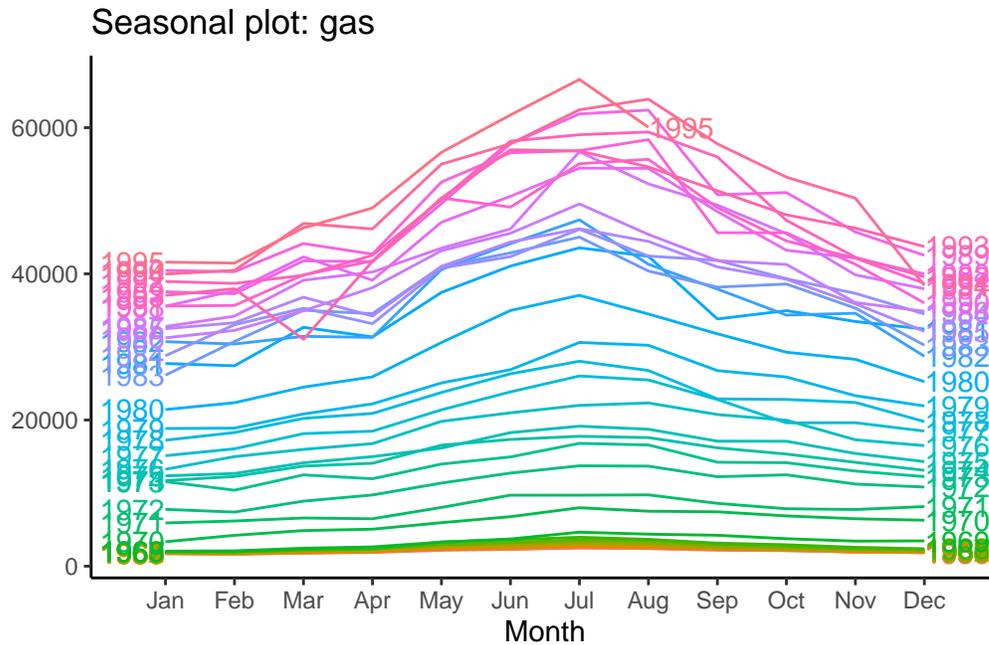
### 1.3 Tipos de gráficos para analizar series temporales

Existen algunos gráficos útiles para analizar más en profundidad las propiedades de las series temporales.

### 1.3.1 Gráfico estacional

Un gráfico estacional permite visualizar claramente el patrón estacional de la serie:

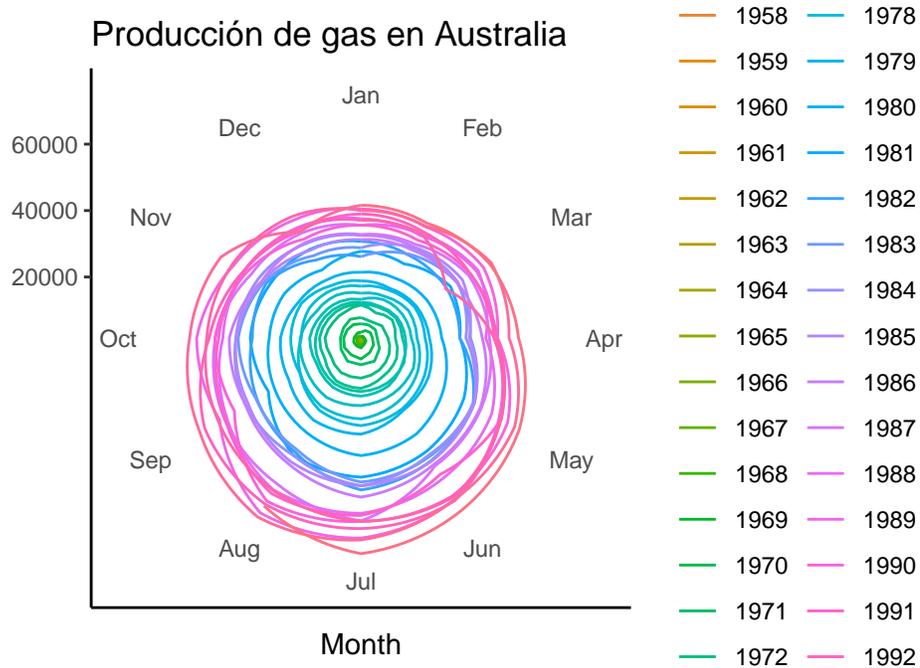
```
library(forecast, quietly=TRUE)
data(gas)
forecast::ggseasonplot(gas, year.labels=TRUE, year.labels.left=TRUE) +
  theme_classic()
```



Se observa que la producción de gas en Australia ha aumentado considerablemente con los años (tendencia creciente). En cuanto al componente estacional, se observa que la producción crece considerablemente en los meses de calor, con máximos en verano, y disminuye en los meses de frío, con mínimos en invierno.

Si se quiere obtener mayor comparativa entre meses, el comando 'polar=TRUE' es bastante útil.

```
library(ggplot2, quietly=TRUE)
forecast::ggseasonplot(gas, polar=TRUE) +
  ggtitle("Producción de gas en Australia") + theme_classic()
```

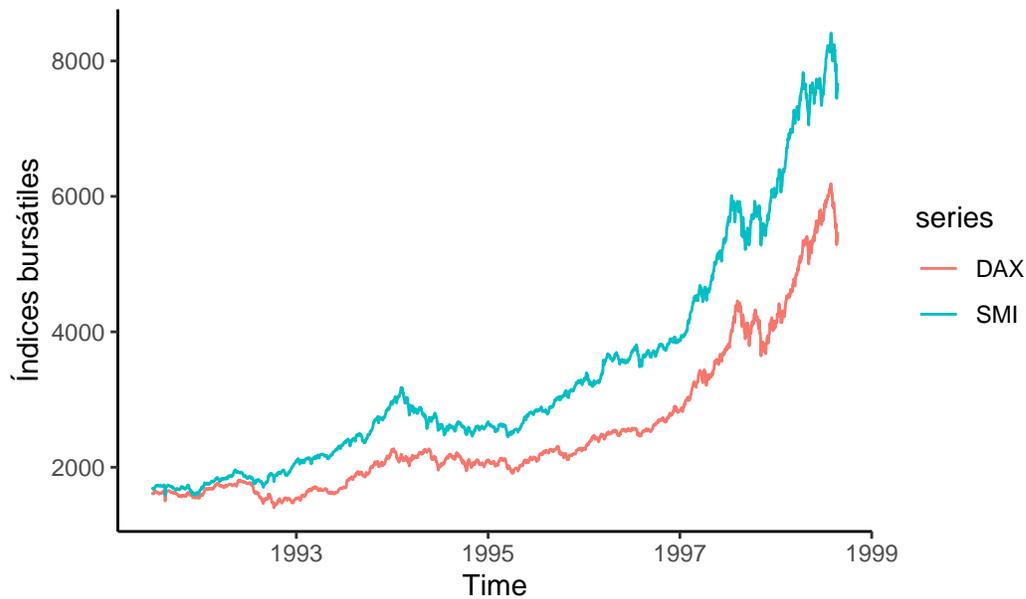


Claramente, los meses líderes en producción son julio y agosto, y los de menor producción, diciembre y enero.

### 1.3.2 Correlación entre series

Además de analizar los aspectos individuales de una serie temporal, se pueden calcular correlaciones entre series para medir el grado de relación entre dos o más series.

```
data("EuStockMarkets")
autoplot(EuStockMarkets[,c(1:2)]) + ylab("Índices bursátiles") +
  theme_classic()
```



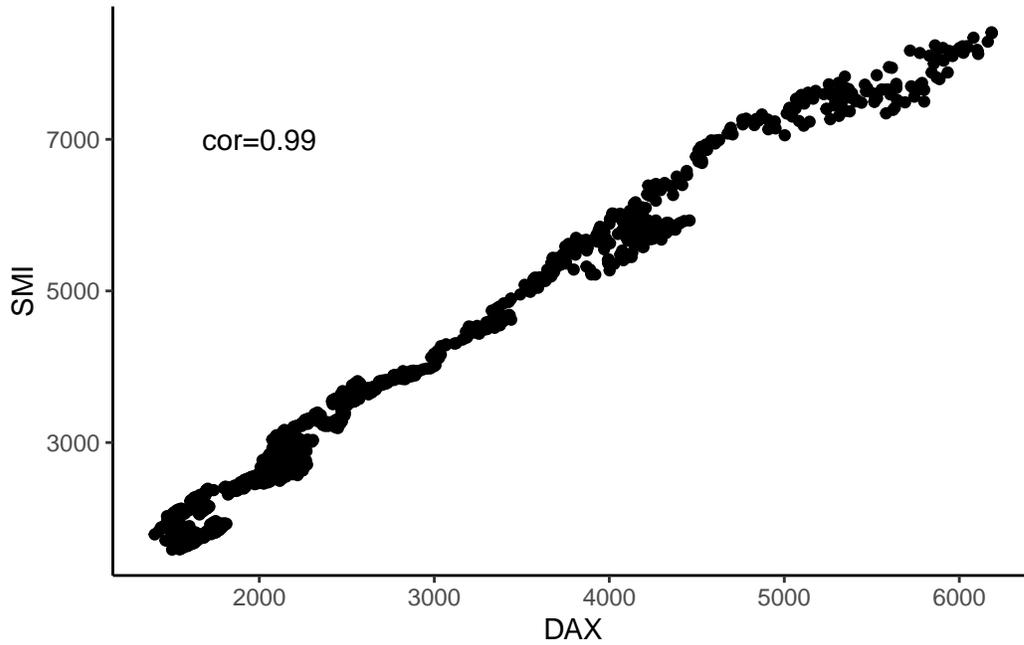
A priori, parece que ambos índices bursátiles guardarán una fuerte correlación. Para obtener una métrica formal de dicha correlación, es útil calcular el coeficiente de correlación entre las dos series:

$$\rho(X_t, Y_t) = \frac{\sigma(X_t Y_t)}{\sigma(X_t)\sigma(Y_t)} = \frac{Cov(X_t, Y_t)}{\sqrt{Var(X_t)Var(Y_t)}}$$

```
EUdf <- as.data.frame(EuStockMarkets)
cor(EUdf$DAX, EUdf$SMI)
```

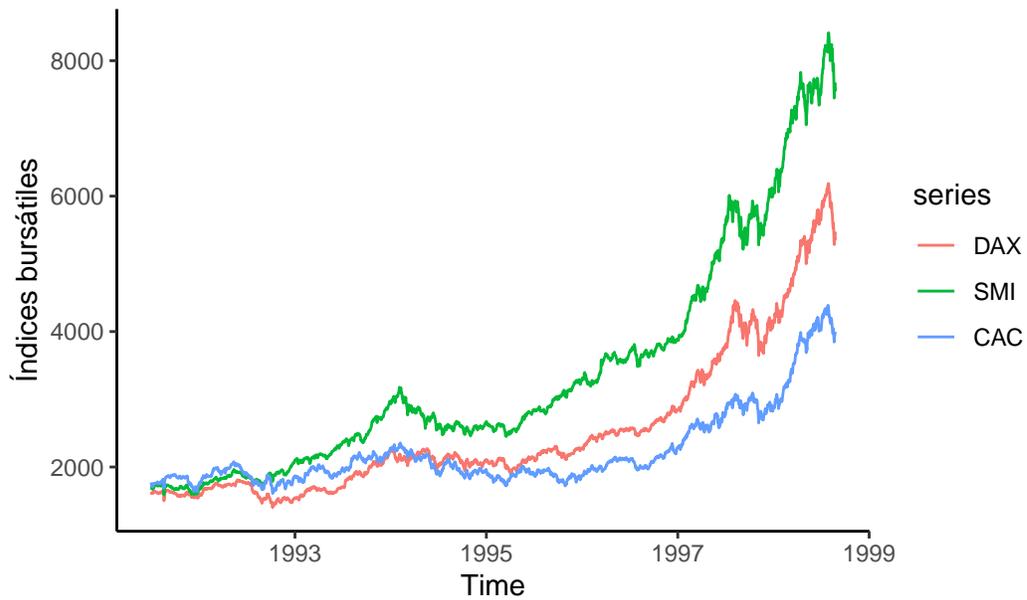
```
[1] 0.9911539
```

```
ggplot(data=EUdf, aes(x=DAX, y=SMI)) +
  geom_point() +
  annotate("text", label="cor=0.99", x=2000, y=7000, size=4, color="black")+
  theme_classic()
```



La correlación es casi perfecta. Veamos entre tres índices bursátiles a la vez

```
autoplot(EuStockMarkets[,c(1:3)]) + ylab("Índices bursátiles") +  
  theme_classic()
```

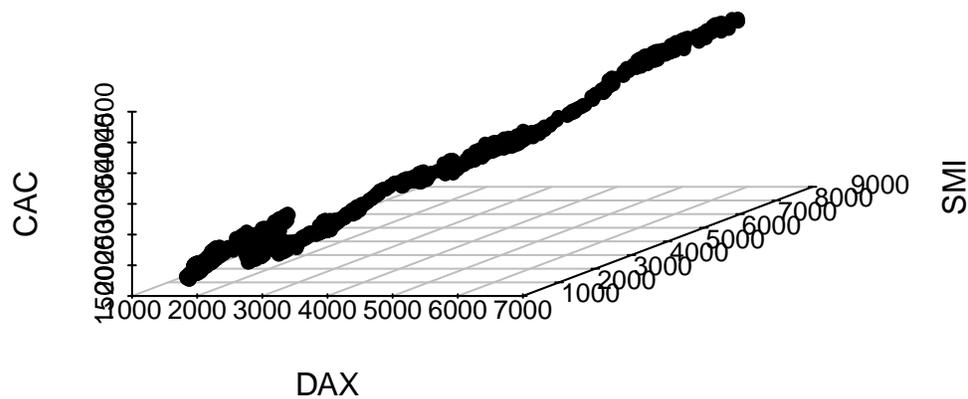


A priori debería de haber una correlación muy alta entre los tres.

```
attach(EUdf)
library(scatterplot3d, quietly=TRUE)
```

Warning: package 'scatterplot3d' was built under R version 4.2.3

```
scatterplot3d(DAX, SMI, CAC, grid=TRUE, box=FALSE, color = "black",
              pch = 16)
```



Efectivamente, hay una alta correlación, lo cual se interpreta de la variación lineal creciente de la nube de puntos.

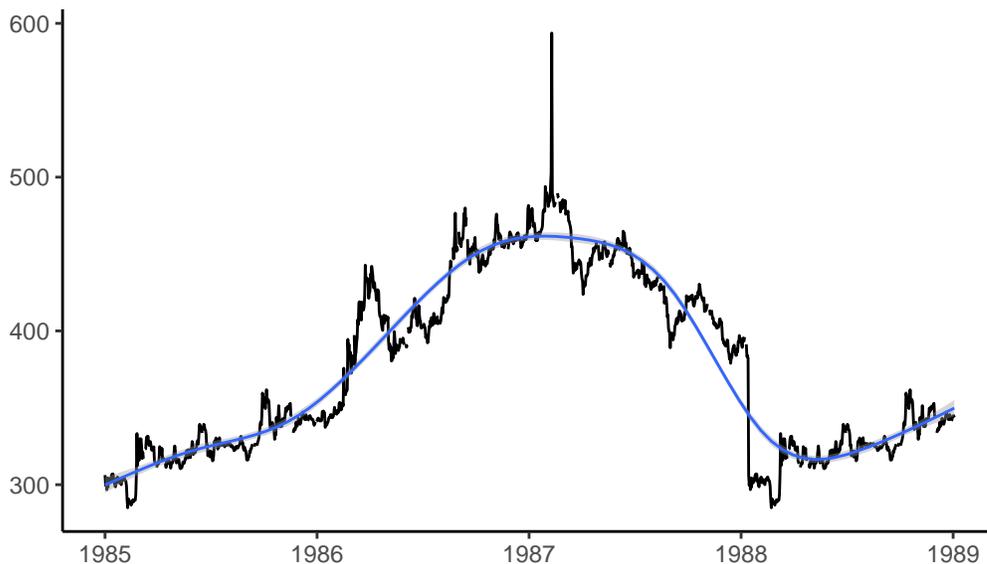
**Ejercicio:** Cargue y analice la estacionalidad de la serie AirPassengers.

## 2 Series temporales estacionarias

### 2.1 Tendencias estocásticas

Además del componente estacional, y de las tendencias que se pueden ajustar como una función del paso del tiempo, existen otro tipo de tendencias. Una *tendencia estocástica* define la trayectoria a largo plazo de una serie, pero de forma aleatoria.

```
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
data("gold")
gold1 <- ts(gold, start=c(1985,1), end=c(1989,03), frequency = 365)
autoplot(gold1) + geom_smooth(lwd=0.5) + xlab("") + ylab("") + theme_classic()
```



En la figura anterior, la tendencia ajustada con `geom_smooth()` no captura completamente el comportamiento de la serie, que es mucho más errático que la función del tiempo. Por ejemplo, entre 1987, el precio sube y baja de forma mucha mas continua de lo que ajusta la línea. Esto es

debido a que la línea constituye una *tendencia determinística* (que crece o decrece dependiendo de una función conocida), mientras que la serie temporal se ajusta a una *tendencia estocástica* (que crece o decrece sujeta a la aleatoriedad).

De manera que, siendo  $x_t, y_t$  variables de series temporales

$$x_t = f(t)$$

constituye una serie dependiente de tendencia determinística, y

$$y_t = f(t) + \epsilon_t$$

constituye una serie dependiente de una tendencia estocástica. El componente  $\epsilon_t$  es aleatorio o impredecible, y hace que la serie se desvíe de forma aleatoria de su trayectoria predecible.

### 2.1.1 Tratamiento de una tendencia estocástica

Para eliminar una tendencia estocástica, basta con realizar primeras diferencias de la serie temporal. Para ello, primero hay que retardar la variable.

```
y = c(2, 3.5, 6, 8, 9, 10)
y_t = ts(y, start=2000)
y_t1 = lag(y_t, 1)
y_t
```

```
Time Series:
Start = 2000
End = 2005
Frequency = 1
[1] 2.0 3.5 6.0 8.0 9.0 10.0
```

```
y_t1
```

```
Time Series:
Start = 1999
End = 2004
Frequency = 1
[1] 2.0 3.5 6.0 8.0 9.0 10.0
```

$t$	$y_t$	$y_{t-1}$
2000	2	3.5
2001	3.5	6
2002	6	8

En el año 2001,  $y_t=3.5$ , luego  $y_{t-1}$  tendrá dicho valor en el año 2000 (un periodo antes). Toda la variable se retarda un periodo.

Las primeras diferencias de la serie  $y_t$ , denotadas como  $\Delta y_t$ , se calculan:

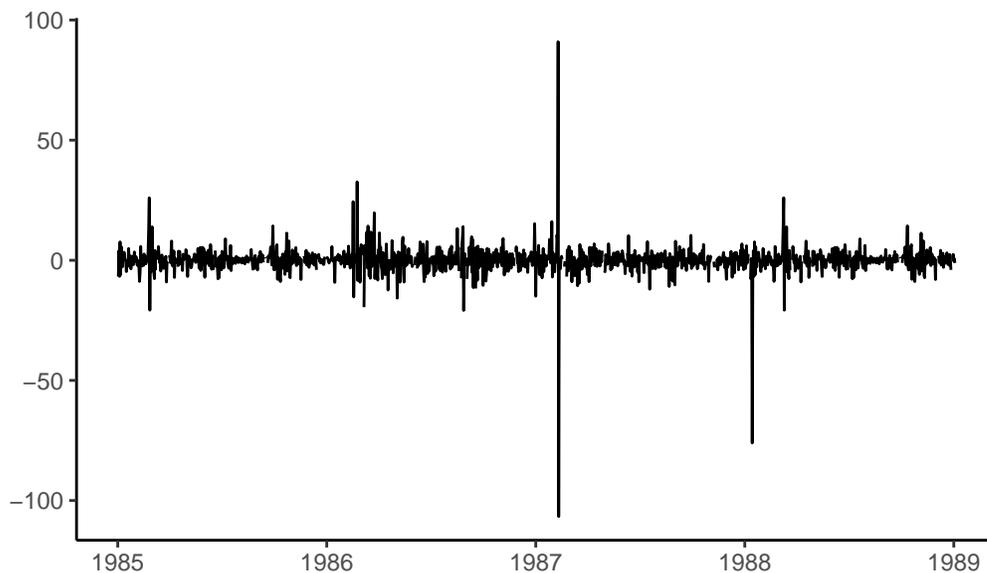
$$\Delta y_t = y_t - y_{t-1}$$

En caso de que la tendencia no se elimine, este proceso se puede iterar al orden  $k$ , donde la diferenciación toma la forma:

$$\Delta^k y_t = y_t - y_{t-k}$$

En el caso del ejemplo anterior:

```
goldt1 <- diff(gold1, 1)
autoplot(goldt1) + xlab("") + ylab("") + theme_classic()
```



Donde lo que se representa ahora es la variación de la serie. Este proceso elimina la tendencia estocástica, puesto que la media ahora es constante alrededor de 0 (lo que también se denomina *reversión hacia la media* (0 en este caso)).

## 2.1.2 Diferenciación estacional

Este proceso también se puede aplicar para eliminar la tendencia estacional. Si la frecuencia de los datos es  $s$ , entonces se puede diferenciar en el orden  $s$  para eliminar el componente estacional.

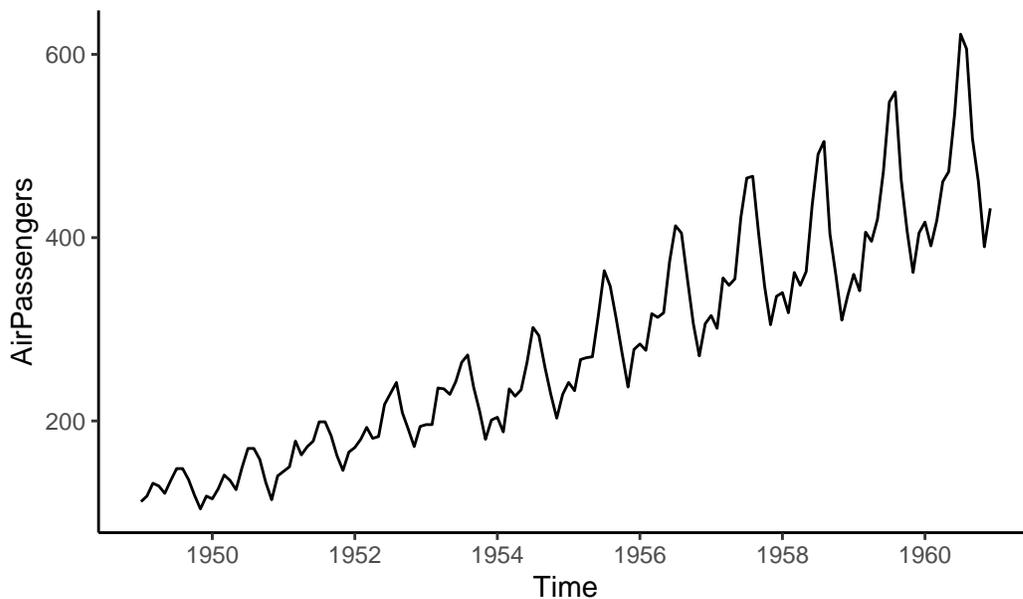
Es decir, la serie:

$$\Delta^s y_t = y_t - y_{t-s}$$

estará libre del componente estacional.

Observe la siguiente serie:

```
data("AirPassengers")
forecast::autoplot(AirPassengers) + theme_classic()
```



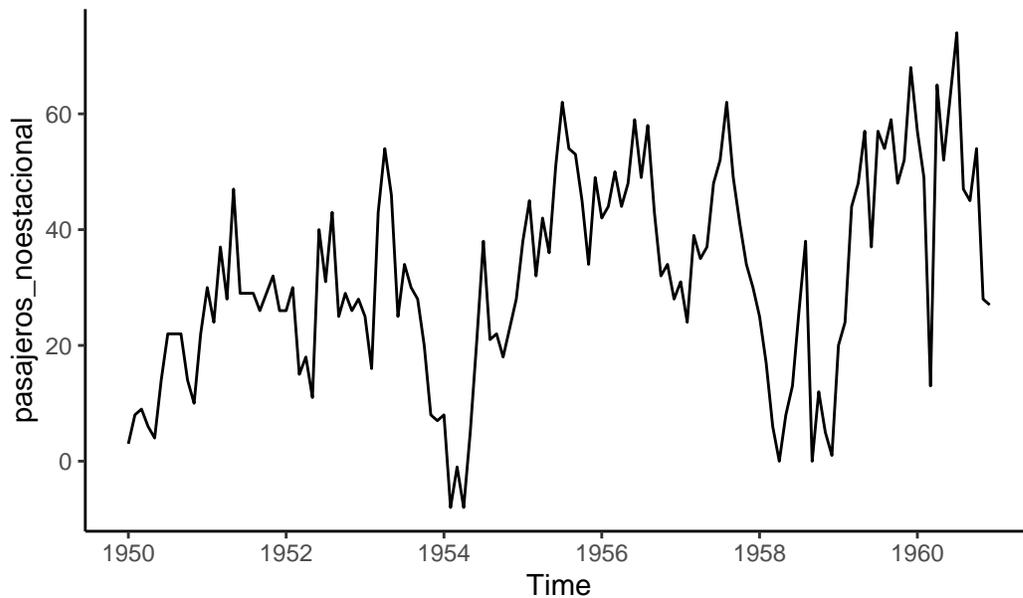
Claramente se tienen más pasajeros en los meses de vacaciones.

La serie presenta una frecuencia mensual, luego al aplicar diferencias en el doceavo retardo, es decir:

$$\Delta^{12}y_t = y_t - y_{t-12}$$

se obtiene:

```
pasajeros_noestacional <- diff(AirPassengers, 12)
forecast::autoplot(pasajeros_noestacional) + theme_classic()
```



Como alternativa, se pueden emplear  $1, \dots, s$  variables ficticias para capturar el componente estacional, y después restar dicho efecto a la serie, obteniendo así una serie desestacionalizada.

## 2.2 Estacionariedad

Para realizar un análisis de regresión con series temporales y poder extraer inferencias, primero es necesario que las series temporales sean estacionarias. Para ello, se tienen que dar tres condiciones:

$$E(y_t) = \mu \tag{2.1}$$

$$\sigma^2(y_t) = \sigma^2 \quad (2.2)$$

$$\lim_{p \rightarrow \infty} E(y_t y_{t-p}) = 0 \quad (2.3)$$

que conocemos como estacionareidad en media, varianza y autocorrelación.

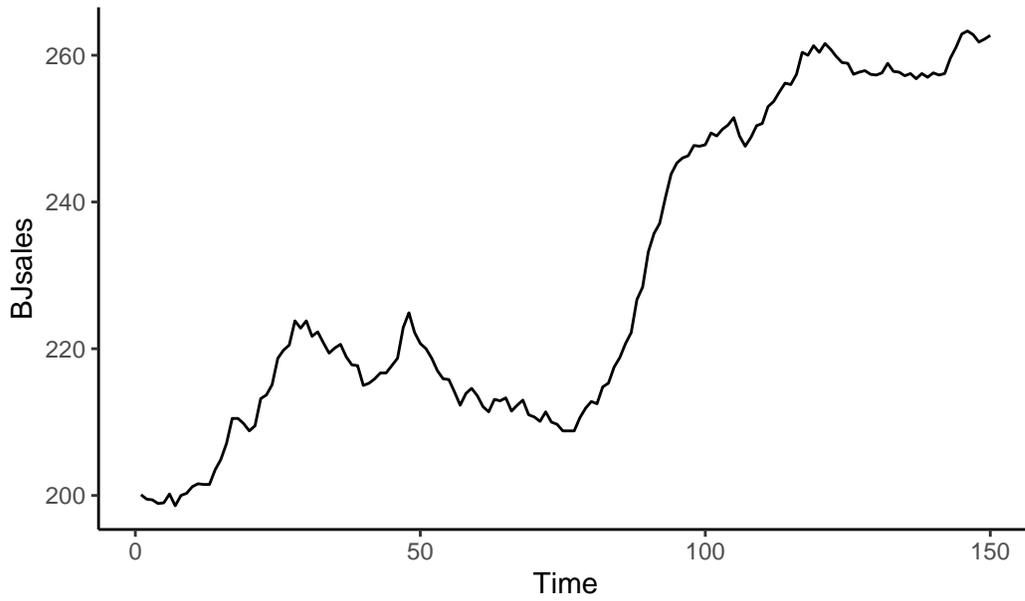
- La primera condición implica que la serie tiene una media (cualquiera) constante (que no depende o varía con el tiempo).
- La segunda condición implica que la serie tiene una varianza (cualquiera) constante (también invariable en el tiempo).
- La tercera condición implica que la correlación entre la serie y sus valores pasados desaparece conforme pasa el tiempo.

Como hemos visto, la primera condición se consigue con la diferenciación, o bien la inclusión de una tendencia determinística. Veamos que pasa con la de la segunda y tercera condición.

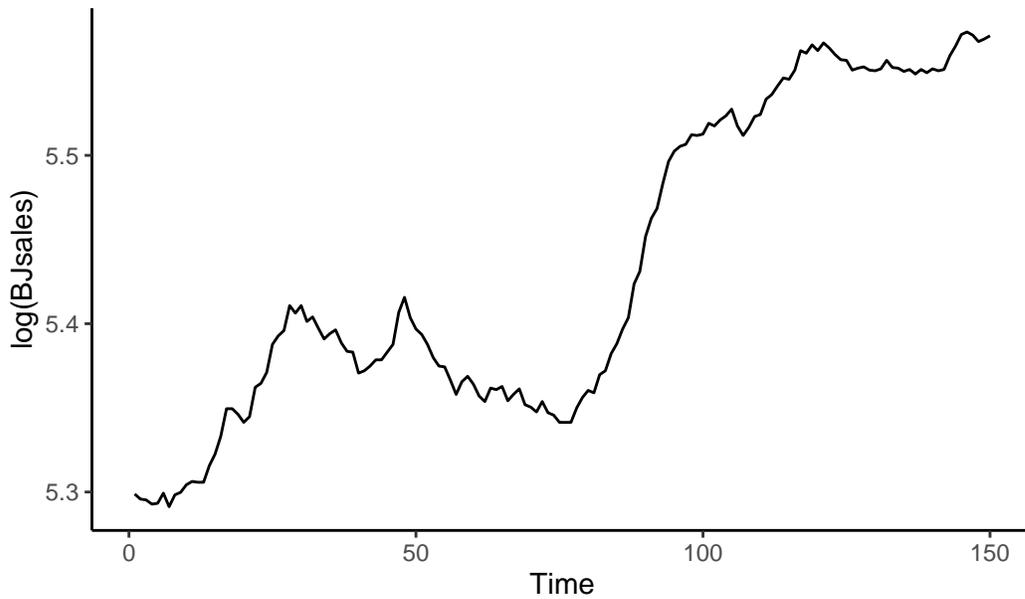
### 2.2.1 Estacionareidad en varianza

Si una serie no es estacionaria en varianza, se puede conseguir que lo sea mediante la aplicación de logaritmos naturales.

```
data("BJsales")
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
autoplot(BJsales) + theme_classic()
```



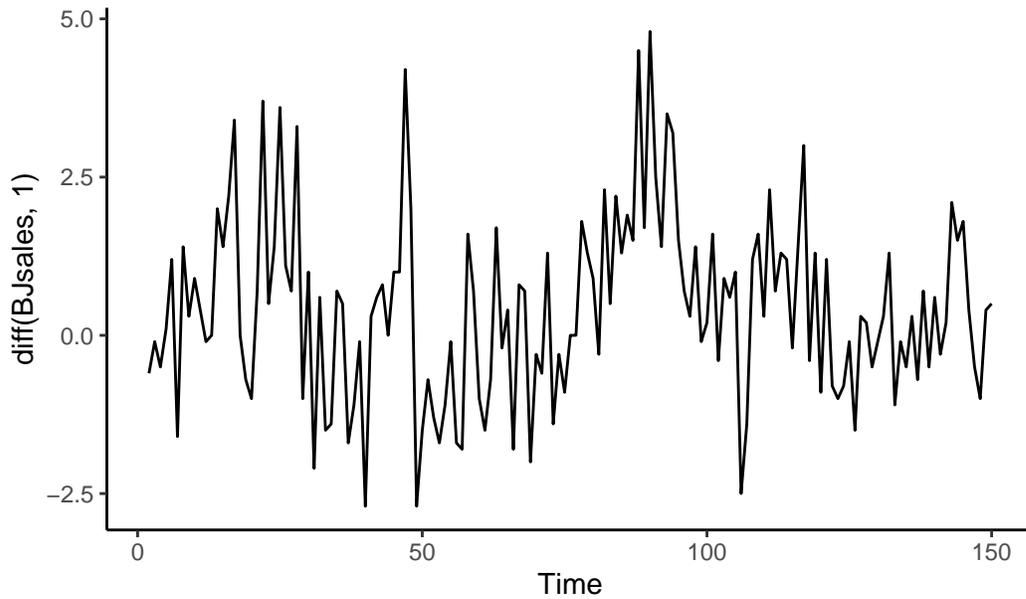
```
autoplot(log(BJsales)) + theme_classic()
```



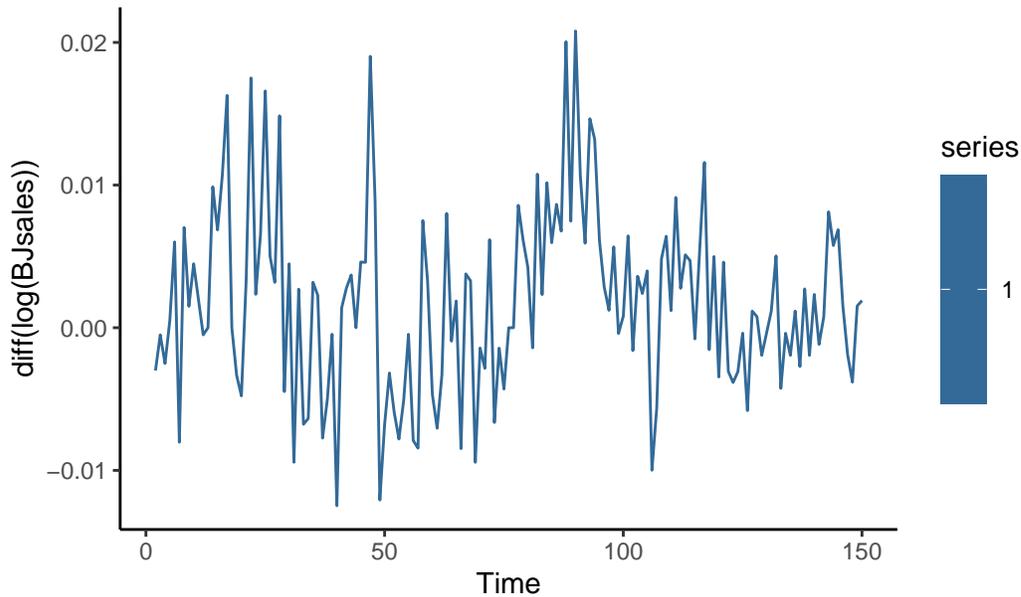
La clave de la interpretación de las figuras anteriores está en la escala del eje Y. Cuando se representa la serie, esta varía desde 200 a 260. Al aplicar logaritmos naturales, la serie varía

desde entre 5.3 y 5.6, con lo que la varianza será mucho menor. El logaritmo suaviza la varianza de la serie. Sin embargo, la serie no tiene una media constante (no es estacionaria en media).

```
library(forecast, quietly = TRUE)
autoplot(diff(BJsales,1)) + theme_classic()
```



```
autoplot(diff(log(BJsales)),1) + theme_classic()
```



Al aplicar primeras diferencias, se eliminan las tendencias y la media es constante. Si *antes* se aplican logaritmos, la serie será estacionaria en media y varianza, como se aprecia al analizar el eje de abscisas.

### 2.2.2 Autocorrelación

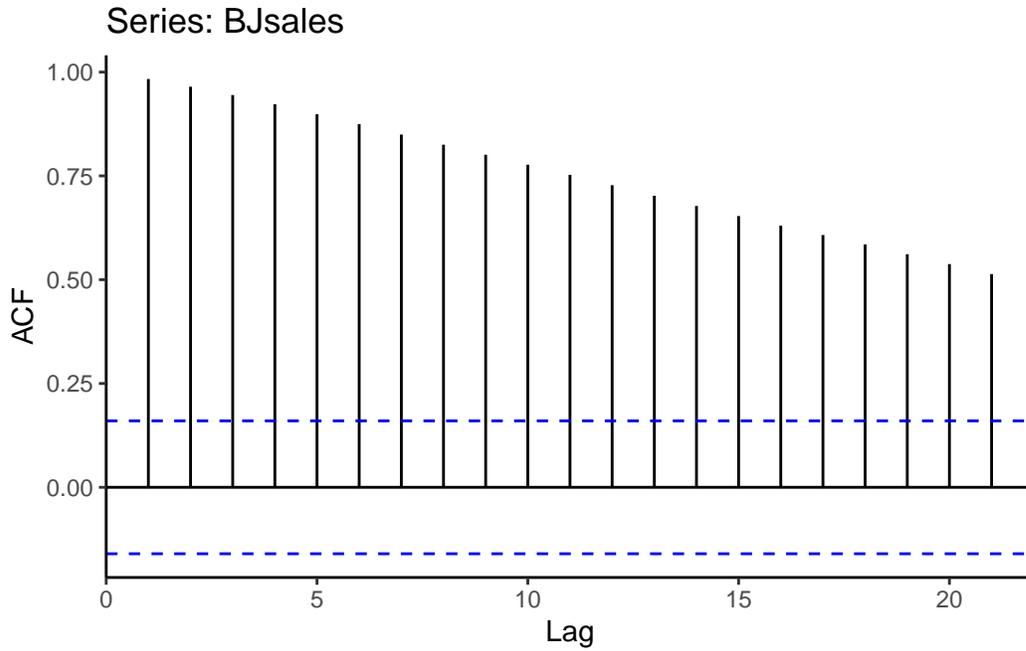
Para medir el grado de autocorrelación de orden  $r$  una serie, se emplea el coeficiente de autocorrelación ( $\rho_r$ ).

$$\rho_r = \frac{\sum_{t=r+1}^n (y_t - \bar{y})(y_{t-r} - \bar{y})}{\sum_{t=1}^n (y_t - \bar{y})^2}$$

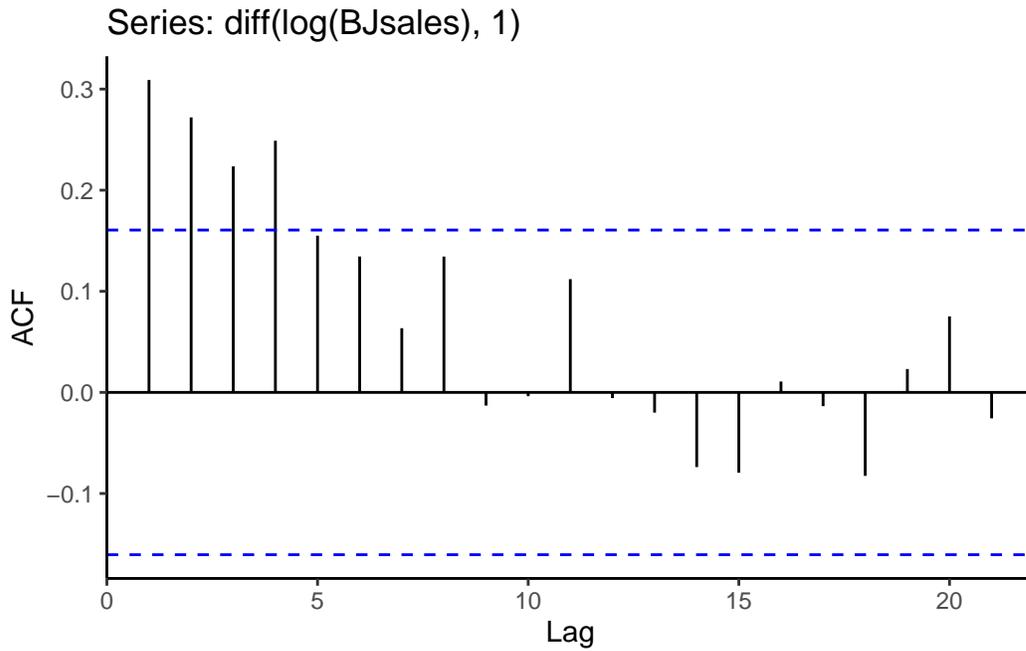
siendo  $n$  el horizonte o tamaño de la serie temporal.

Normalmente, este coeficiente se representa para los  $r$  retardos en un gráfico de autocorrelación o *correlograma*:

```
ggAcf(BJsales) + theme_classic()
```



```
ggAcf(diff(log(BJsales),1)) + theme_classic()
```



donde salirse de las bandas azules indica que el retardo es estadísticamente significativo a un

nivel de confianza estadística del 95%. Es evidente que la serie log-diferenciada está mucho menos autocorrelacionada conforme avanza el tiempo.

## 2.2.3 Contrastes de estacionareidad

Aunque un análisis visual siempre arroja información sobre si una serie es o no estacionaria, existen test estadísticos que permiten contrastarlo formalmente. Es el caso del contraste KPSS.

### 2.2.3.1 Contraste KPSS

- Hipótesis nula  $H_0$  : La serie es estacionaria.
- Hipótesis nula  $H_1$  : La serie no es estacionaria.

```
library(tseries, quietly=TRUE)
kpss.test(BJsales.lead)
```

KPSS Test for Level Stationarity

```
data: BJsales.lead
KPSS Level = 2.556, Truncation lag parameter = 4, p-value = 0.01
```

```
kpss.test(diff(log(BJsales),1))
```

KPSS Test for Level Stationarity

```
data: diff(log(BJsales), 1)
KPSS Level = 0.12482, Truncation lag parameter = 4, p-value = 0.1
```

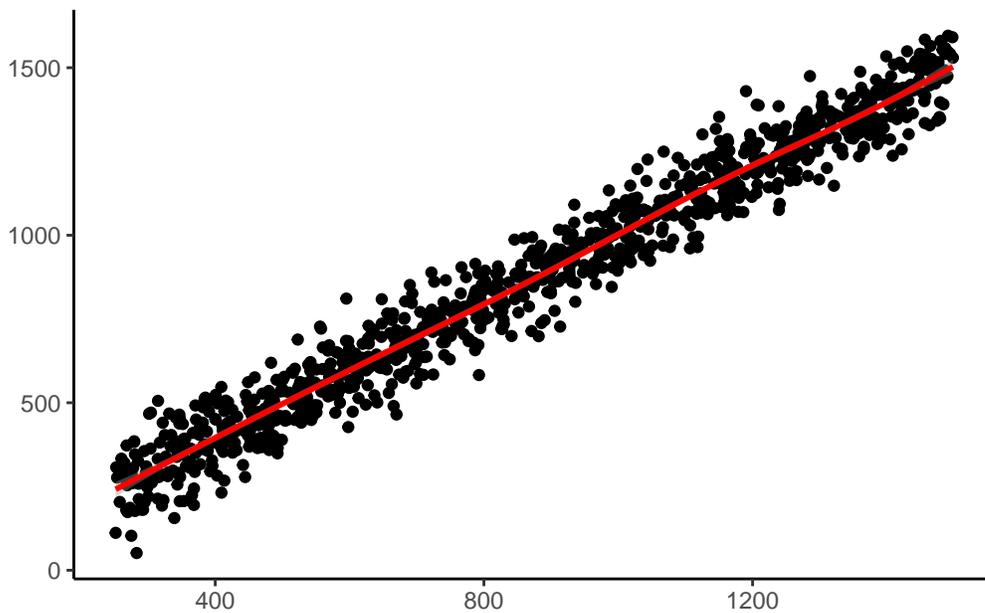
Tras log-diferenciar, el contraste KPSS permite aceptar la nula a cualquier nivel de significación estadística usual (p-valor por encima de 0.1).

**Ejercicio:** Cargue, inspeccione y analice la estacionareidad en media, varianza y autocorrelación de la serie LakeHuron. Si no es estacionaria, tránsformela para que lo sea. Contraste formalmente la estacionareidad de la serie original, y de la serie transformada.

### 3 Análisis de regresión

La siguiente gráfica muestra un ajuste a unos datos través de un modelo de regresión lineal:

```
N <- 1000
X = runif(N, 0, 25)
alpha = 250
beta = 50
mu = alpha + beta*X
sigma = 75
y=rnorm(n=N, mu, sigma)
library(ggplot2, quietly=TRUE)
datasim <- data.frame(mu,y)
ggplot(data = datasim, aes(x=mu, y=y)) + geom_point() +
  geom_smooth(col="red") + theme_classic() +
  xlab("") + ylab("")
```



### 3.1 Regresión simple

El análisis de regresión permite realizar inferencias y predicciones sobre una o varias series temporales. Se basa en el modelo de regresión lineal:

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t$$

donde las variables poseen el sufijo  $t$  ya que son series temporales. No varían por sujeto (sección cruzada; sufijo  $i$ ), sino que la variación es temporal, y corresponde al mismo sujeto ( $i$  es fija).

El intercepto o  $\beta_0$  representa el valor predcido para  $Y_t$  cuando  $X_t = 0$ . Por otra parte, la pendiente  $\beta_1$  consiste en el cambio medio en  $Y_t$  cuando  $X_t$  aumenta en una unidad. A esto se le denomina *efecto marginal* de  $X_t$  sobre  $Y_t$ .

```
library(forecast, quietly=TRUE)
library(Ecdat, quietly=TRUE)
data("Longley")
Longley
```

Time Series:

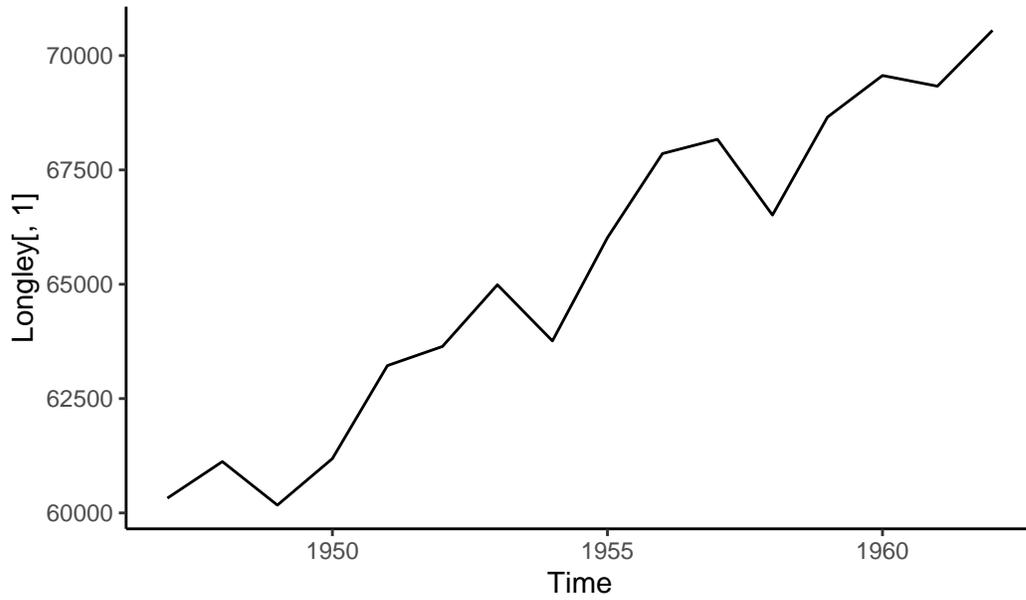
Start = 1947

End = 1962

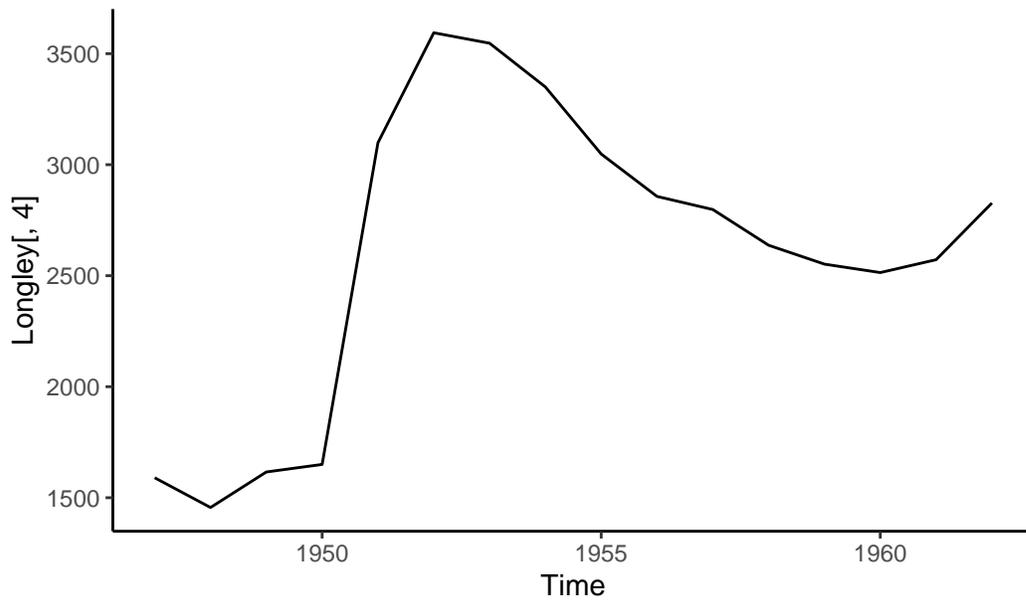
Frequency = 1

	employ	price	gnp	armed
1947	60323	83.0	234289	1590
1948	61122	88.5	259426	1456
1949	60171	88.2	258054	1616
1950	61187	89.5	284599	1650
1951	63221	96.2	328975	3099
1952	63639	98.1	346999	3594
1953	64989	99.0	365385	3547
1954	63761	100.0	363112	3350
1955	66019	101.2	397469	3048
1956	67857	104.6	419180	2857
1957	68169	108.4	442769	2798
1958	66513	110.8	444546	2637
1959	68655	112.6	482704	2552
1960	69564	114.2	502601	2514
1961	69331	115.7	518173	2572
1962	70551	116.9	554894	2827

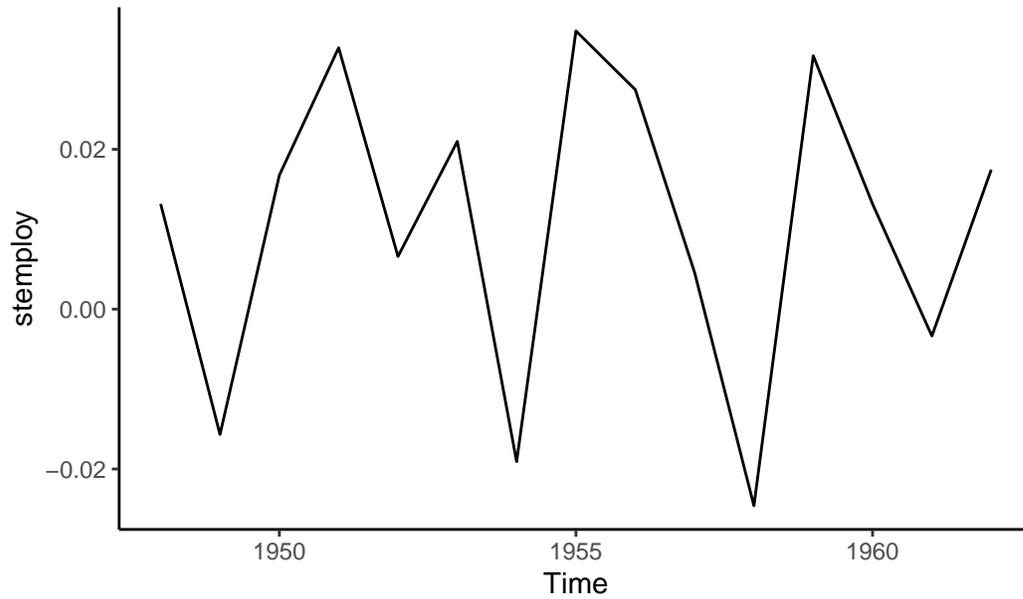
```
forecast::autoplot(Longley[,1]) + theme_classic()
```



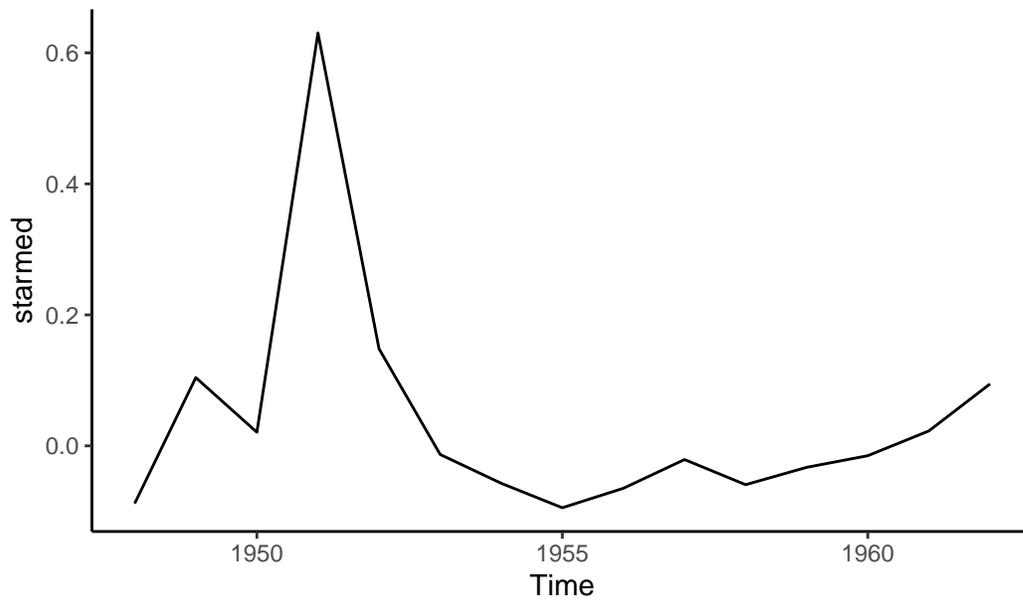
```
forecast::autoplot(Longley[,4]) + theme_classic()
```



```
stemploy <- diff(log(Longley[,1]))  
starmed <- diff(log(Longley[,4]))  
forecast::autoplot(stemploy) + theme_classic()
```



```
forecast::autoplot(starmed) + theme_classic()
```



```
library(tseries, quietly=TRUE)
kpss.test(starmed)
```

KPSS Test for Level Stationarity

```
data: starmed
KPSS Level = 0.1619, Truncation lag parameter = 2, p-value = 0.1
```

```
kpss.test(stemploy)
```

KPSS Test for Level Stationarity

```
data: stemploy
KPSS Level = 0.095943, Truncation lag parameter = 2, p-value = 0.1
```

```
modelo1 <- lm(starmed ~ stemploy)
summary(modelo1)
```

Call:

```
lm(formula = starmed ~ stemploy)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.18373	-0.09455	-0.03384	0.02740	0.54544

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.01655	0.05372	0.308	0.763
stemploy	2.08901	2.54152	0.822	0.426

Residual standard error: 0.1809 on 13 degrees of freedom

Multiple R-squared: 0.0494, Adjusted R-squared: -0.02372

F-statistic: 0.6756 on 1 and 13 DF, p-value: 0.4259

Evalando los resultados del modelo:

- El coeficiente  $\hat{\beta}_1$  es de 2.08, con lo que se espera que conforme aumente el empleo en una unidad, el número de alistados en las fuerzas armadas aumente en 2 unidades aproximadamente. Este resultado es contrario a lo esperado, ya que uno de los motivos que puede llevar a un ciudadano a escoger un trabajo de riesgo como es el ejército es precisamente un alto desempleo, con lo cual se esperaría una relación inversa.
- Sin embargo, el  $p$ -valor del coeficiente es de 0.43 aproximadamente. Con lo cual, el parámetro no es estadísticamente significativo, y el efecto marginal inexistente.
- A el mismo resultado se llega si se hace un contraste de hipótesis individual mediante el estadístico  $t = \beta/\sigma(\beta)$ , ya que  $2.08/2.54=0.82$ , muy inferior al valor tabulado de  $t_{15-1,0.025} = 2.14$ , como se puede comprobar justo debajo. De manera que el contraste de hipótesis queda:
  - $H_0$ : El parámetro no es estadísticamente significativo (si el valor tabulado es superior al empírico).
  - $H_1$ : El parámetro es estadísticamente significativo (si sucede al contrario).

```
qt(0.025, 14, lower.tail = F)
```

```
[1] 2.144787
```

- Estadístico  $R^2$ . Indica que el modelo explica los datos en aproximadamente un 5%. Sin embargo, el estadístico  $R^2$  que penaliza la inclusión adicional de una variable aleatoria es negativo, con lo que se puede concluir un ajuste nulo.

- Estadístico  $F$  de significación global del modelo. Las hipótesis son:
  - $H_0$  : El modelo no es globalmente significativo.
  - $H_1$  : El modelo es globalmente significativo.

al tener un  $p$ -valor de 0.42, se acepta la nula, y el modelo no es globalmente significativo, en concordancia con el contraste de hipótesis individual para el parámetro  $\hat{\beta}_1$ .

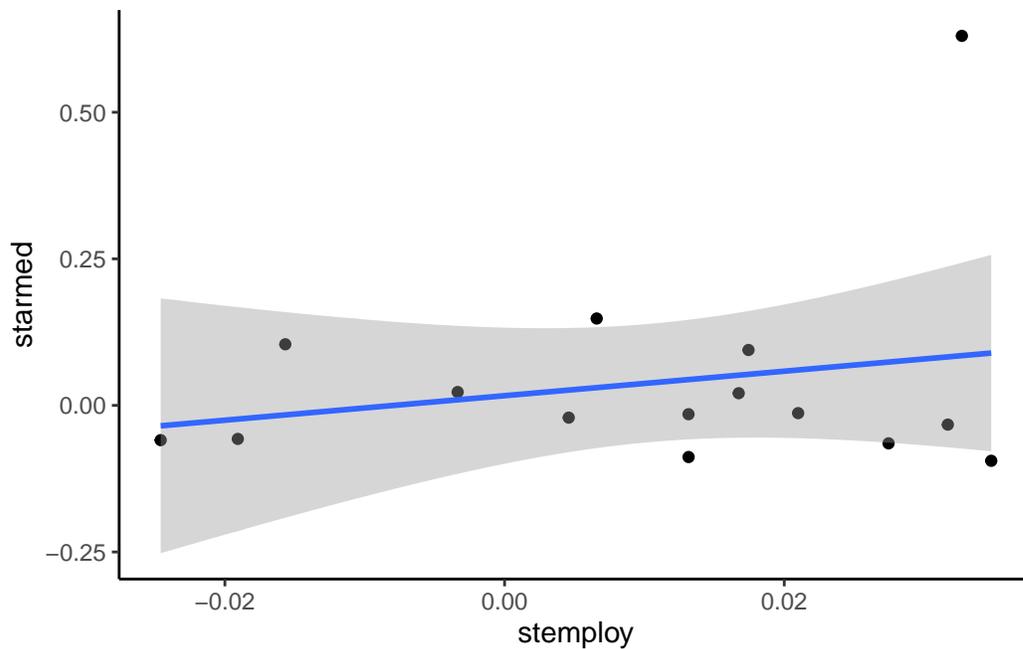
De manera que se puede inferir que el empleo no explica el número de alistados en EEUU, al menos durante el periodo considerado (al ser la muestra pequeña, y, por tanto, no representativa de toda la historia de EEUU). Veamos si el análisis gráfico lo corrobora:

```
conjuntodatos <- data.frame(stemploy, starmed)
ggplot(conjuntodatos, aes(x=stemploy, y=starmed)) +
  geom_point() +
  geom_smooth(method = "lm" ) + theme_classic()
```

Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

`geom\_smooth()` using formula = 'y ~ x'



Efectivamente, el que aumente el empleo apenas hace que aumente el valor de los alistados en la armada. Tanto la nube de puntos como la recta de regresión son prácticamente planas.

## 3.2 Regresión espuria

Veamos ahora que hubiese sucedido si, en lugar de asegurar una transformación estacionaria de las variables, se hubiesen considerado las variables en *niveles*.

```
kpss.test(Longley[,1])
```

KPSS Test for Level Stationarity

```
data: Longley[, 1]
KPSS Level = 0.62594, Truncation lag parameter = 2, p-value = 0.02028
```

```
kpss.test(Longley[,4])
```

Warning in kpss.test(Longley[, 4]): p-value greater than printed p-value

KPSS Test for Level Stationarity

```
data: Longley[, 4]
KPSS Level = 0.22033, Truncation lag parameter = 2, p-value = 0.1
```

```
modelo2 <- lm(armed ~ employ, data=Longley)
summary(modelo2)
```

Call:

```
lm(formula = armed ~ employ, data = Longley)
```

Residuals:

Min	1Q	Median	3Q	Max
-770.5	-489.2	-166.0	453.8	1139.4

Coefficients:

Estimate	Std. Error	t value	Pr(> t )
----------	------------	---------	----------

```
(Intercept) -3.312e+03  3.080e+03  -1.075  0.3004
employ       9.062e-02  4.710e-02   1.924  0.0749 .
```

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

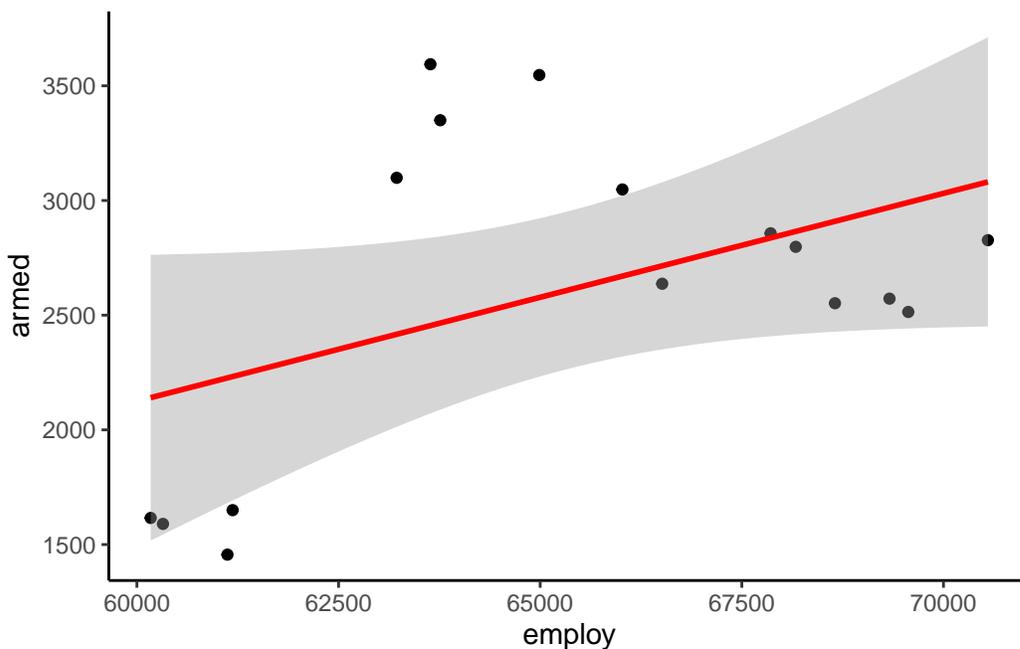
Residual standard error: 640.6 on 14 degrees of freedom

Multiple R-squared: 0.2091, Adjusted R-squared: 0.1526

F-statistic: 3.702 on 1 and 14 DF, p-value: 0.07492

El modelo arroja un efecto marginal positivo de 0.09 estadísticamente significativo a un nivel de confianza del 90%. Además, el ajuste de los datos es de  $R^2 = 21\%$ . El coeficiente ajustado da un valor del 15%. Se podría inferir que existe una relación directa significativa, lo cual es contraintuitivo, además de falso, como se ha visto en el apartado anterior. A esto se le denomina *regresión espuria*.

```
library(dplyr, quietly=TRUE)
Longley %>% as.data.frame() %>% ggplot(aes(x=employ, y=armed)) +
  geom_point() +
  geom_smooth(method="lm", col="red") + theme_classic()
```



Por el análisis gráfico del modelo también parece que hay relación, cuando no la hay. Es por ello que siempre hay que asegurar que las variables de series temporales sean estacionarias antes de realizar un análisis de regresión. De lo contrario, se puede incurrir en falsas conclusiones.

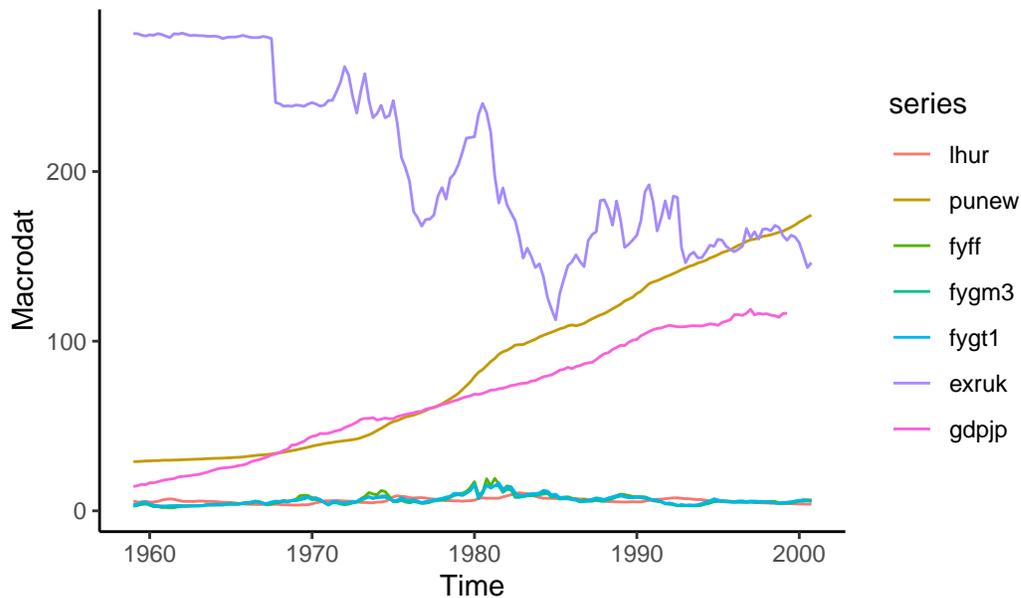
### 3.3 Regresión múltiple

El modelo de regresión lineal se puede extender a  $k$  variables aleatorias independientes:

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \dots + \beta_k X_{k,t} + \epsilon_t$$

donde se mantienen las propiedades del modelo de regresión lineal simple o univariante. La recta de regresión es ahora un hiperplano.

```
library(Ecdat, quietly=TRUE)
library(dplyr, quietly=TRUE)
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
library(tseries, quietly=TRUE)
data(Macrodat)
autoplot(Macrodat) + theme_classic()
```



```
lista1 <- lapply(Macrodat[,c(1,5,6)], diff)
lhur1 <- lista1$lhur
kpss.test(lhur1)
```

Warning in kpss.test(lhur1): p-value greater than printed p-value

KPSS Test for Level Stationarity

```
data: lhur1
KPSS Level = 0.088716, Truncation lag parameter = 4, p-value = 0.1
```

```
fygt11 <- lista1$fygt1
kpss.test(fygt11)
```

Warning in kpss.test(fygt11): p-value greater than printed p-value

KPSS Test for Level Stationarity

```
data: fygt11
KPSS Level = 0.082756, Truncation lag parameter = 4, p-value = 0.1
```

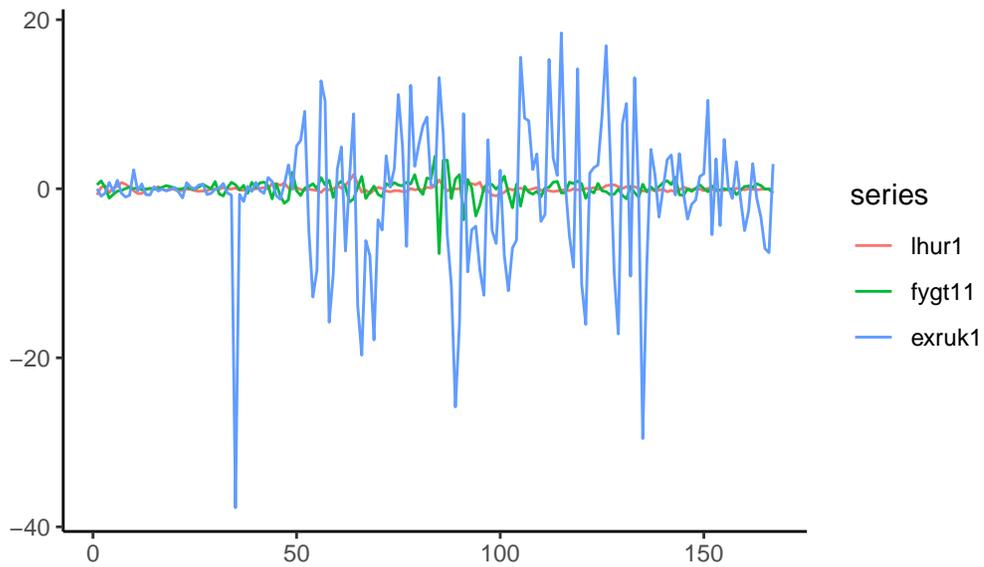
```
exruk1 <- lista1$exruk
kpss.test(exruk1)
```

Warning in kpss.test(exruk1): p-value greater than printed p-value

KPSS Test for Level Stationarity

```
data: exruk1
KPSS Level = 0.047765, Truncation lag parameter = 4, p-value = 0.1
```

```
dat <- data.frame(lhur1, fygt11, exruk1)
dat1 <- as.ts(dat, start = c(1959,1), end=c(2000,4))
autoplot(dat1) + xlab("") + ylab("") + theme_classic()
```



```
modelom1 <- lm(exruk1 ~ fygt11 + lhur1)
summary(modelom1)
```

Call:

```
lm(formula = exruk1 ~ fygt11 + lhur1)
```

Residuals:

Min	1Q	Median	3Q	Max
-36.481	-3.713	0.852	4.072	18.741

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.7891	0.6161	-1.281	0.2021
fygt11	-1.1128	0.6005	-1.853	0.0656 .
lhur1	0.5274	2.0121	0.262	0.7936

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.957 on 164 degrees of freedom

Multiple R-squared: 0.02692, Adjusted R-squared: 0.01506

F-statistic: 2.269 on 2 and 164 DF, p-value: 0.1067

El tipo de interés a 1 año del bono afecta negativamente al tipo de cambio dolar-libra. El efecto marginal es de  $-\$1.11$ , y el coeficiente es significativo al 90% de significación. Al ser un modelo con dos variables independientes y una dependiente, la representación gráfica es en tres dimensiones.

```
x <- fyg11
y <- lhur1
z <- exruk1

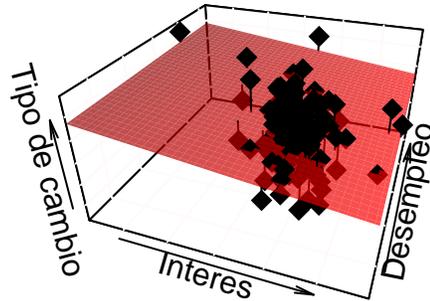
fit <- lm(z ~ x + y)

grid.lines = 40
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)

fitpoints <- predict(fit)

library(plot3D, quietly=TRUE)

scatter3D(x, y, z, pch = 18, cex = 1.5, colvar=FALSE, col="black",
          theta = 20, phi = 30, bty="u",
          col.panel = "white", expand = 0.5, col.grid = "snow",
          xlab = "Interes", ylab = "Desempleo", zlab = "Tipo de cambio",
          surf = list(x = x.pred, y = y.pred, z = z.pred,
                     facets = TRUE, col=ramp.col (col = c("red1","red4"),
                                                    n = 100, alpha=0.7), fit =
                     fitpoints),main = "")
```



Conforme aumenta el tipo de interés, disminuye el tipo de cambio.

### 3.3.1 Interpretación de coeficientes

Cuando se log-diferencia una serie, la escala de las variables, y, por tanto, de los coeficientes, varía. Si se tiene una variable aleatoria  $Y_t$ , se toman logaritmos  $\log(Y_t)$ , y se diferencia, se tiene:

$$\Delta \log(Y_t) = \log(Y_t) - \log(Y_{t-1}) = \log\left(\frac{Y_t}{Y_{t-1}}\right) \approx \frac{Y_t - Y_{t-1}}{Y_{t-1}}$$

es decir, se está trabajando con tasas de crecimiento o decrecimiento porcentuales o proporcionales. Por tanto, en el modelo de regresión lineal:

$$\Delta \log(Y_t) = \beta_0 + \beta_1 \Delta \log(X_{1,t}) + \beta_2 \Delta \log(X_{2,t}) + \dots + \beta_k \Delta \log(X_{k,t}) + \epsilon_t$$

los coeficientes  $\beta_1, \beta_2, \dots, \beta_k$  miden el efecto marginal en elasticidades, es decir, cuanto variará en porcentaje  $Y_t$  si la variable  $X_{1,t}, X_{2,t}, \dots, X_{k,t}$  aumenta un 1%.

### 3.3.2 Evaluación de un modelo econométrico

Los residuos del modelo son la diferencia entre los valores observados en los datos y aquellos estimados por la recta de regresión para cada observación. Por tanto, si la observación 1 tiene valor 3.5, es decir,  $Y_1 = 3.5$ , y el valor estimado para esa observación de la recta de regresión es de 3.8, es decir,  $\hat{Y}_1 = 3.8$ , el residuo para  $t = 1$  será:

$$e_1 = Y_1 - \hat{Y}_1 = 3.5 - 3.8 = -0.3$$

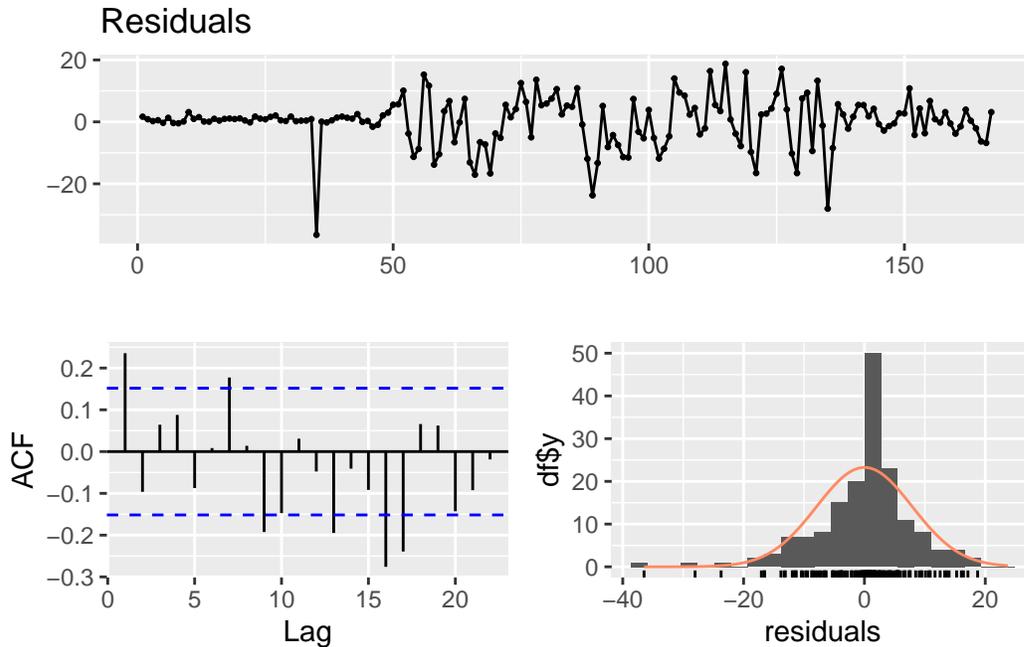
y así sucesivamente para las  $t = 1, \dots, T$  observaciones de las que se disponga.

Para que un modelo de regresión de series temporales esté correctamente especificado, ha de cumplir una serie de propiedades:

1.  $\hat{e}_t = 0$  . \*La media de los residuos ha de ser cero.\* Esto implica que, aunque el modelo se desvía de los datos en cada observación, estas desviaciones se compensan, y, en general, el modelo no se equivoca sistemáticamente.
2.  $E(\hat{e}_t \hat{e}_{t-p}) = 0, \forall 1, \dots, p$ . \*Los residuos no deben de sufrir de autocorrelación serial.\* Esto dificulta la inferencia (aunque se pueden emplear estadísticos robustos a la autocorrelación), además de indicar que existen retardos de variables incluidas y no incluidas (tanto dependientes como independiente) que son importantes para explicar el modelo.
3.  $\sum_{t=1}^T X_{k,t} e_t = 0 \forall k$  . \*Las variables independientes no deben de estar correlacionadas con los residuos.\* A esta condición también se le llama exogeneidad.

Finalmente, conviene que los residuos sigan una distribución normal. Aunque esto no es completamente necesario, la inferencia estadística se basa en la distribución normal, con lo cual que los residuos lo sean la facilita

```
checkresiduals(modelom1)
```



Breusch-Godfrey test for serial correlation of order up to 10

```
data: Residuals
LM test = 26.997, df = 10, p-value = 0.002607
```

Por un lado, la media de los residuos parece 0. Sin embargo, parece haber dos periodos muy claros diferenciados por la varianza de los residuos, que es mucho menor para el primero que para el segundo, a lo que se le llama *\*heterocedasticidad\**. Por tanto, el modelo se ajusta bien al primer periodo, pero no al segundo. Habría que estimar dos modelos distintos para los dos periodos, o bien tener en cuenta este *\*cambio estructural\** en el modelo.

Por otra parte, parece que los residuos están autocorrelacionados. Esto se corrobora por el test de Breusch-Godfrey, cuya hipótesis nula es no autocorrelación residual. Probablemente hay variables y retardos de las mismas que precisan ser consideradas en el modelo.

Finalmente, hay una presencia clara de datos anómalos para los cuales los residuos son muy altos, que probablemente vengan por periodos de crisis económica para los cuales el modelo no puede predecir variaciones bruscas de los datos.

```
dat[,"Residuals"] = as.numeric(residuals(modelom1))
grafica1 <- ggplot(dat, aes(x=fygt11, y=Residuals)) + geom_point() + geom_smooth(method="lm")
grafica2 <- ggplot(dat, aes(x=lhur1, y=Residuals)) + geom_point() + geom_smooth(method="lm")
```

```
grafica3 <- ggplot(dat, aes(x=exruk1, y=Residuals))+ geom_point() + geom_smooth(method="lm")
gridExtra::grid.arrange(grafica1, grafica2, grafica3, nrow=1)
```

Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

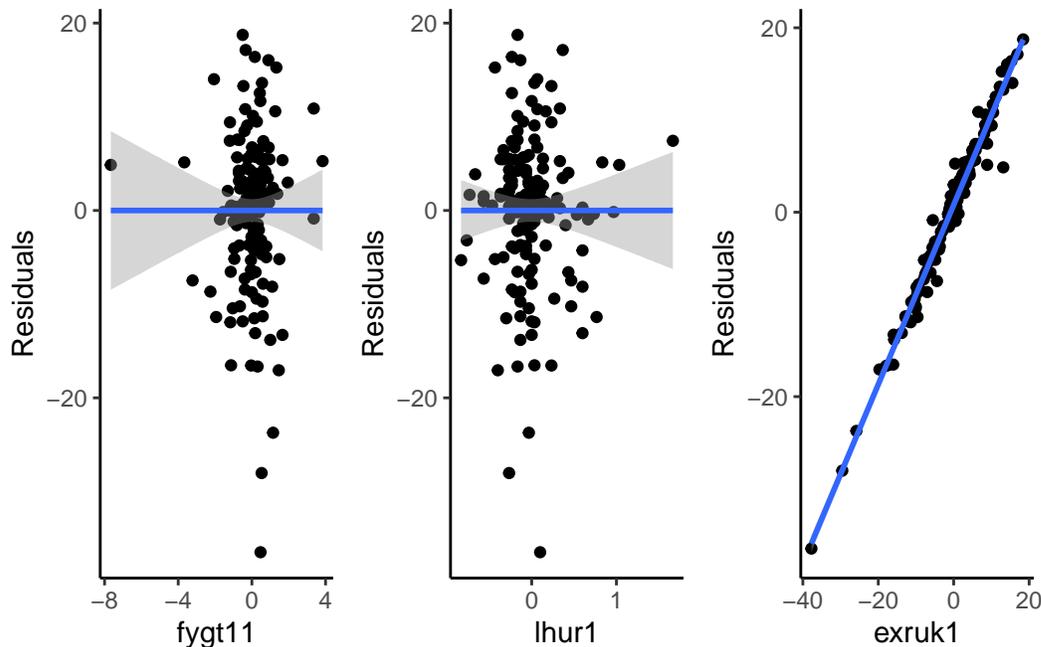
```
`geom_smooth()` using formula = 'y ~ x'
```

Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

```
`geom_smooth()` using formula = 'y ~ x'
```

Don't know how to automatically pick scale for object of type <ts>. Defaulting to continuous.

```
`geom_smooth()` using formula = 'y ~ x'
```



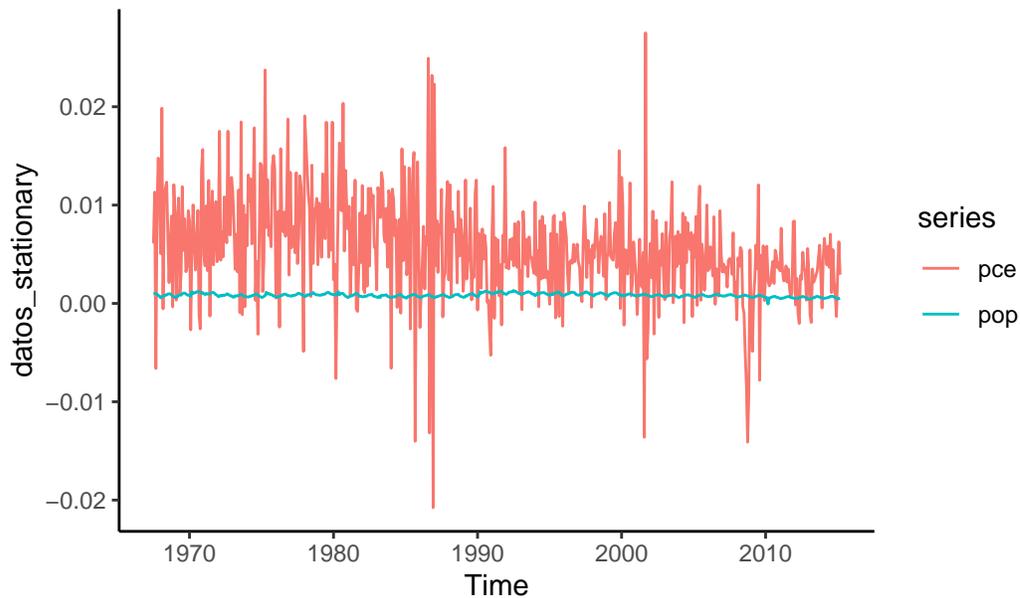
Parece que no hay relación entre las variables independientes y los residuos, con lo cual se cumple el requisito. Sin embargo, hay una alta correlación lineal entre la dependiente y los residuos, lo cual indica que hay otras variables no incluidas en el modelo que explican la variable dependiente.

**Ejercicio:** Cargue e inspeccione la serie temporal `ice.river` del paquete `tseries`. Analice el caudal medio diario del río Vatnsdalsa a través de la precipitación y temperaturas en Hveravellir. Interprete los resultados. Evalúe el modelo.

### 3.4 Predicción con el modelo de regresión

Además de realizar inferencias sobre las variables que causan a otra, el modelo de regresión se puede emplear para realizar predicciones.

```
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
library(dplyr, quietly=TRUE)
library(tseries, quietly=TRUE)
data(economics)
datos <- economics[,2:3]
datos_stationary = datos %>%
  lapply(log) %>%
  lapply(diff) %>%
  as.data.frame() %>%
  ts(start=c(start=c(1967,7), end=c(2015, 03)),
     deltat = 1/12)
autoplot(datos_stationary) + theme_classic()
```



```
lapply(datos_stationary, kpss.test) #
```

```
$pce
```

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 4.4404, Truncation lag parameter = 6, p-value = 0.01

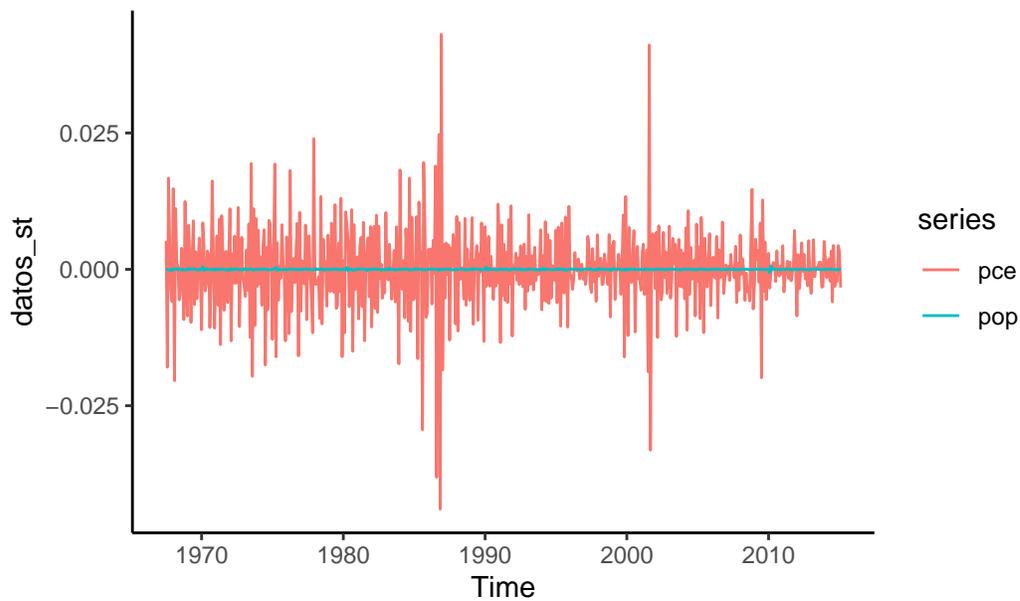
\$pop

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 1.5535, Truncation lag parameter = 6, p-value = 0.01

```
datos_st <- datos_stationary %>%
  lapply(diff) %>%
  as.data.frame() %>%
  ts(start=c(start=c(1967,7), end=c(2015, 02)), deltat = 1/12)
autoplot(datos_st) + theme_classic()
```



```
lapply(datos_st, kpss.test)
```

\$pce

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 0.0063759, Truncation lag parameter = 6, p-value = 0.1

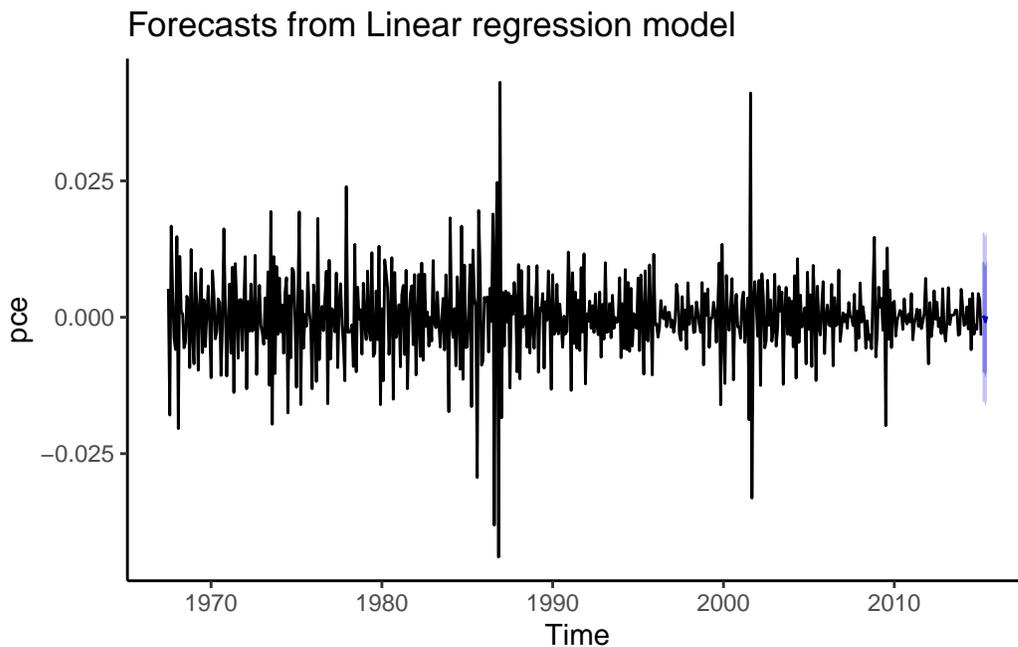
\$pop

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 0.012085, Truncation lag parameter = 6, p-value = 0.1

```
modelots <- tslm(pce ~ pop, data=datos_st)
nuevosdatos = data.frame(pop=c(1.406604e-05,
                               -2.109815e-05, 1.196598e-04, -2.567329e-05))
pred <- forecast(modelots, newdata=nuevosdatos)
autoplot(pred) + theme_classic()
```



Aunque, conforme se llevan a cabo transformaciones de los datos, tanto las predicciones como los coeficientes son cada vez menos interpretables.

### 3.4.1 Dummies estacionales

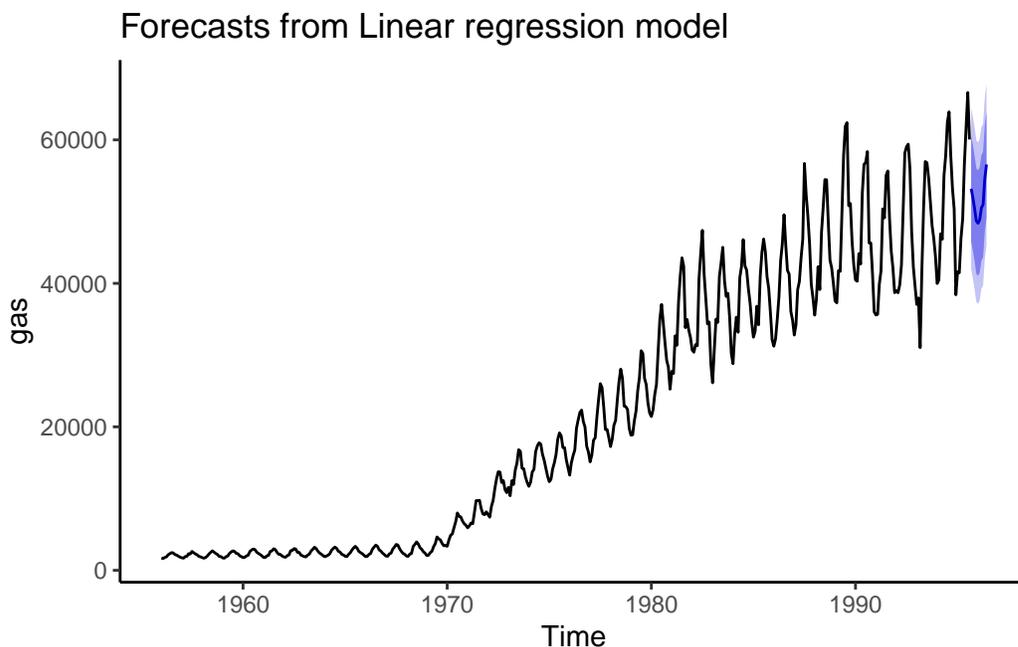
Para evitar tener que hacer diferenciación estacional, se pueden aplicar variables *ficticias* o *dummies*.

$$Y_t = \beta_0 + \sum_{i=1}^k \beta_i X_{i,t} + \sum_{j=1}^s \alpha_j d_{j,t} + \epsilon_t$$

donde cada variable  $d$  es una variable que toma valor 1 si es el periodo del año al que corresponde y 0 si no. La clave está en añadir una variable para cada estación del año  $1, \dots, s$  menos una, para evitar una combinación lineal perfecta de los predictores (trampa de las variables ficticias). Los coeficientes de las variables recogerán el efecto del componente estacional. Si a la serie original, se le sustrae este efecto, se tiene lo que se conoce como una serie *desestacionalizada*.

Si una serie posee una tendencia y un componente estacional, pero no un componente estocástico, es posible hacerla estacionaria mediante la inclusión de dummies estacionales, y una función de la tendencia.

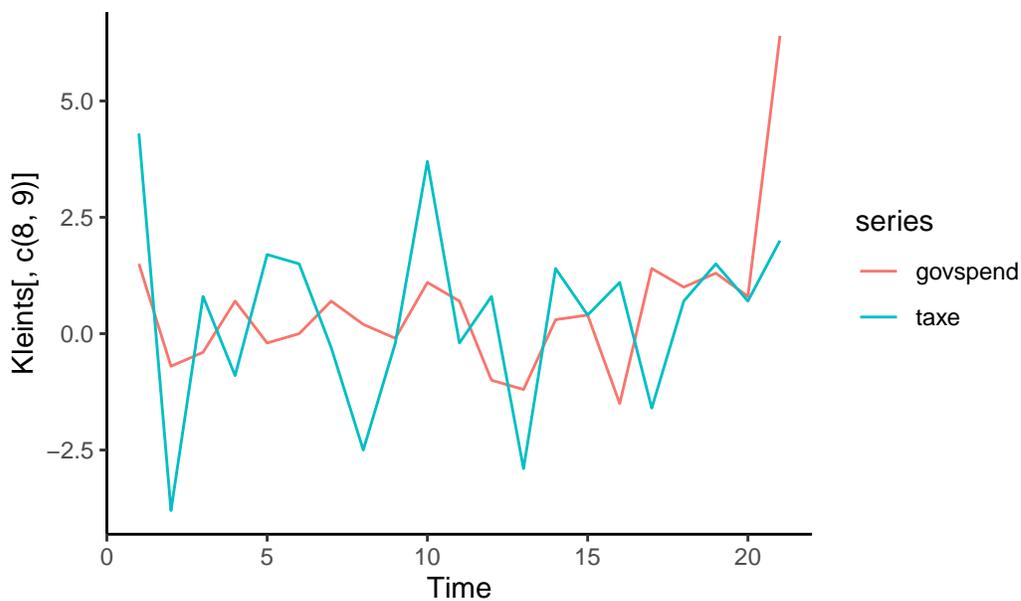
```
data("gas")
modelogas <- tslm(gas ~ trend + season,
                  data=gas)
pred.gas <- forecast(modelogas)
autoplot(pred.gas) + theme_classic()
```



### 3.4.2 Predicción de escenarios

También es posible analizar que pasaría en distintos escenarios incluyendo datos inventados para las variables independientes.

```
library(Ecdat, quietly=TRUE)
data("Klein")
Kleints <- as.ts(Klein, start=1920, end=1940)
Kleints <- lapply(Kleints, diff) %>%
  as.data.frame() %>% as.ts(start=1920, end=1939)
autoplot(Kleints[,c(8,9)]) + theme_classic()
```



```
kpss.test(diff(Kleints[,8]))
```

KPSS Test for Level Stationarity

```
data: diff(Kleints[, 8])
```

```
KPSS Level = 0.42086, Truncation lag parameter = 2, p-value = 0.06817
```

```
kpss.test(diff(Kleints[,9]))
```

KPSS Test for Level Stationarity

data: diff(Kleints[, 9])

KPSS Level = 0.14562, Truncation lag parameter = 2, p-value = 0.1

```
gobmod <- tslm(govspend ~ taxe, data=Kleints)
gobmod
```

Call:

```
tslm(formula = govspend ~ taxe, data = Kleints)
```

Coefficients:

```
(Intercept)      taxe
    0.4336      0.2798
```

```
h = 2
```

```
Kleints
```

Time Series:

Start = 1

End = 21

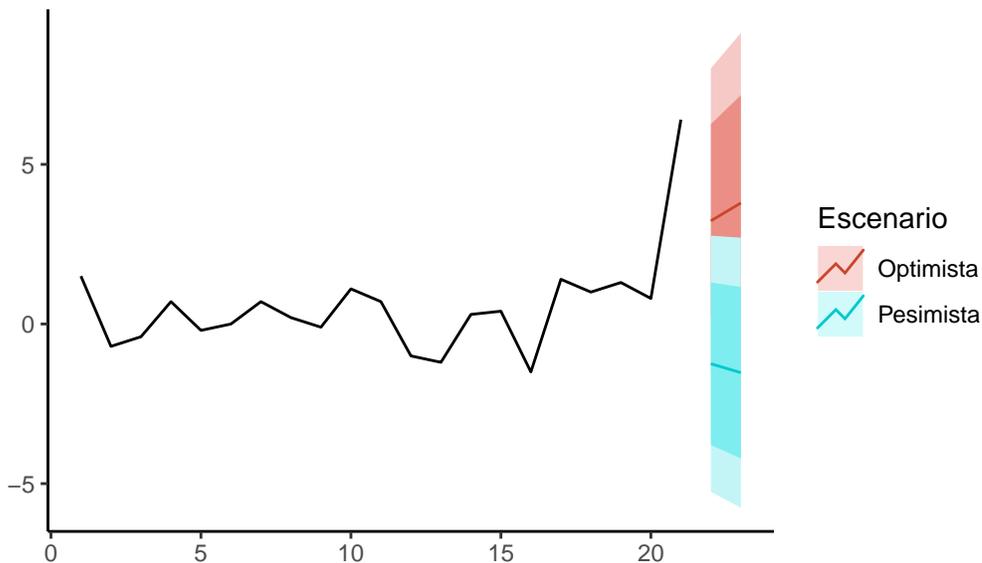
Frequency = 1

	cons	profit	privwage	inv	lcap	gnp	pubwage	govspend	taxe
1	2.1	-0.3	-3.3	-2.9	2.7	0.7	0.5	1.5	4.3
2	3.1	4.5	3.8	2.1	-0.2	4.5	0.2	-0.7	-3.8
3	4.2	1.5	4.8	3.3	1.9	7.1	0.0	-0.4	0.8
4	1.4	1.0	-0.2	-2.2	5.2	-0.1	0.2	0.7	-0.9
5	2.0	0.7	1.5	2.1	3.0	3.9	0.1	-0.2	1.7
6	2.5	-0.5	2.0	0.5	5.1	3.0	0.1	0.0	1.5
7	1.1	0.2	0.5	-1.4	5.6	0.4	0.3	0.7	-0.3
8	1.1	1.3	1.3	-1.2	4.2	0.1	0.1	0.2	-2.5
9	0.5	0.6	2.1	2.1	3.0	2.5	0.3	-0.1	-0.2
10	-2.8	-6.1	-3.4	-4.1	5.1	-5.8	0.2	1.1	3.7
11	-4.1	-4.2	-3.4	-4.4	1.0	-7.8	0.6	0.7	-0.2
12	-5.3	-4.4	-5.5	-2.8	-3.4	-9.1	0.5	-1.0	0.8
13	0.9	4.2	-0.5	1.1	-6.2	0.8	0.3	-1.2	-2.9

## Introducción a la Econometría Financiera

14	2.2	1.1	2.1	2.1	-5.1	4.6	0.4	0.3	1.4
15	2.6	1.7	2.6	1.7	-3.0	4.7	0.1	0.4	0.4
16	6.4	3.6	3.6	3.4	-1.3	8.3	1.3	-1.5	1.1
17	1.0	-0.3	4.2	-0.1	2.1	2.3	-0.7	1.4	-1.6
18	-1.2	-2.0	-2.8	-3.9	2.0	-4.1	1.0	1.0	0.7
19	4.1	3.7	3.4	3.2	-1.9	8.6	0.1	1.3	1.5
20	3.4	2.1	3.4	2.0	1.3	6.2	0.2	0.8	0.7
21	4.7	2.4	8.3	1.6	3.3	12.7	0.5	6.4	2.0

```
valores.opt <- data.frame(taxe=c(10, 12))
valores.pes <- data.frame(taxe=c(-6, -7))
pred.opt <- forecast::forecast(gobmod, newdata=valores.opt)
pred.pes <- forecast::forecast(gobmod, newdata=valores.pes)
autoplot(Kleints[,8]) + xlab("") + ylab("") + autolayer(pred.opt, PI=T, series="Optimista")
```



Dando valores positivos de la variable recaudación impositiva, se obtiene una predicción mayor y viceversa.

**Ejercicio:** Con solo una línea de código, prediga el comportamiento de la serie `AirPassengers` a través de su tendencia y dummies estacionales. ¿Es una predicción razonable?

## 4 Modelos econométricos basados en procesos estocásticos

Como se ha visto anteriormente, las series temporales están sujetas a cierto grado de aleatoriedad o incertidumbre (tendencia estocástica), y a cierto componente cierto que determina su comportamiento a largo plazo (tendencia determinística). Formalmente, una variable aleatoria que evoluciona en el tiempo y soporta efectos o tendencias aleatorias es un **proceso estocástico**.

Existen multitud de fórmulas para describir el comportamiento de procesos estocásticos, siendo algunas de ellas tremendamente útiles para modelizar el comportamiento empírico de las series temporales. De hecho, el poder modelizar una serie temporal mediante un proceso estocástico permite predecir sin necesidad de conocer la estructura o los factores que influyen en la serie, sino simplemente describiendo el comportamiento de la misma hasta la fecha.

### 4.1 Proceso autorregresivo

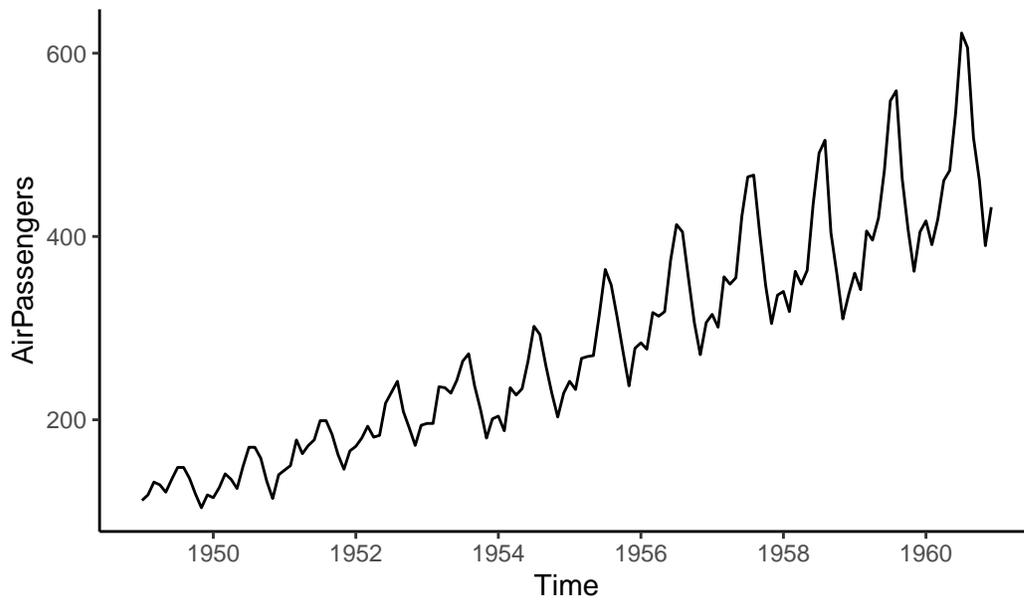
Un proceso autorregresivo consiste, tal y como el nombre indica, en un proceso estocástico que se regresa sobre si mismo más un término error.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + u_t = c + \sum_{i=1}^p \phi_i Y_{t-i} + u_t$$

siendo  $Y_t$  el proceso estocástico,  $\phi_0, \phi_1, \dots, \phi_p$  los coeficientes del término independiente y el proceso retardado  $p$  veces respectivamente, y  $u_t$  el término error. Al modelo autorregresivo de orden (máximo retardo)  $p$  se le conoce como  $AR(p)$ .

Para estimar un modelo autorregresivo, la serie debe de ser estacionaria previamente. Además, el comportamiento de muchísimas series temporales agregadas, especialmente las económicas, suele aproximarse bien con un modelo  $AR(1)$  o  $AR(2)$ .

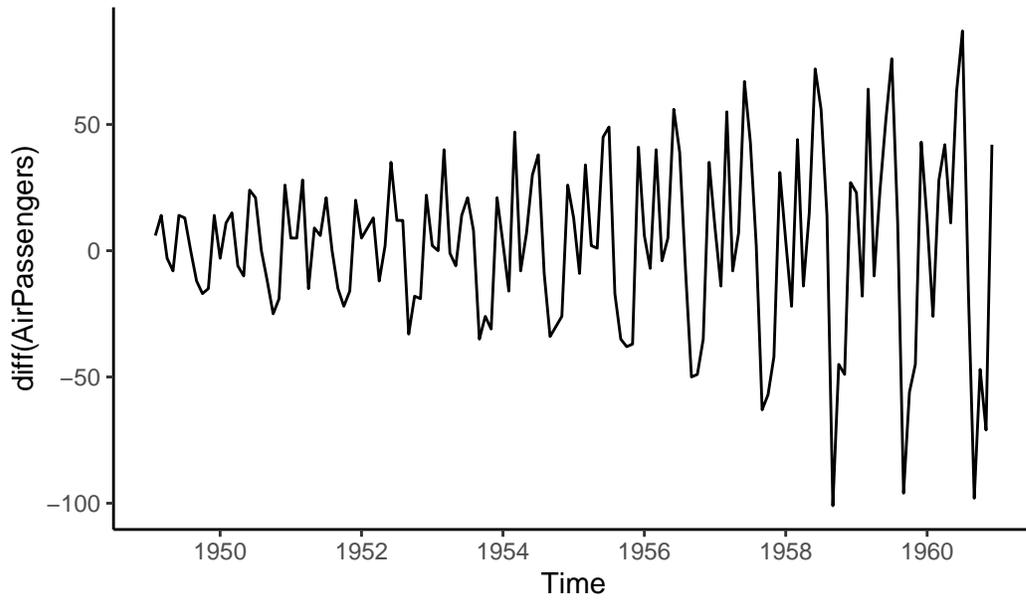
```
data("AirPassengers")
library(forecast, quietly=TRUE)
autoplot(AirPassengers) + ggplot2::theme_classic()
```



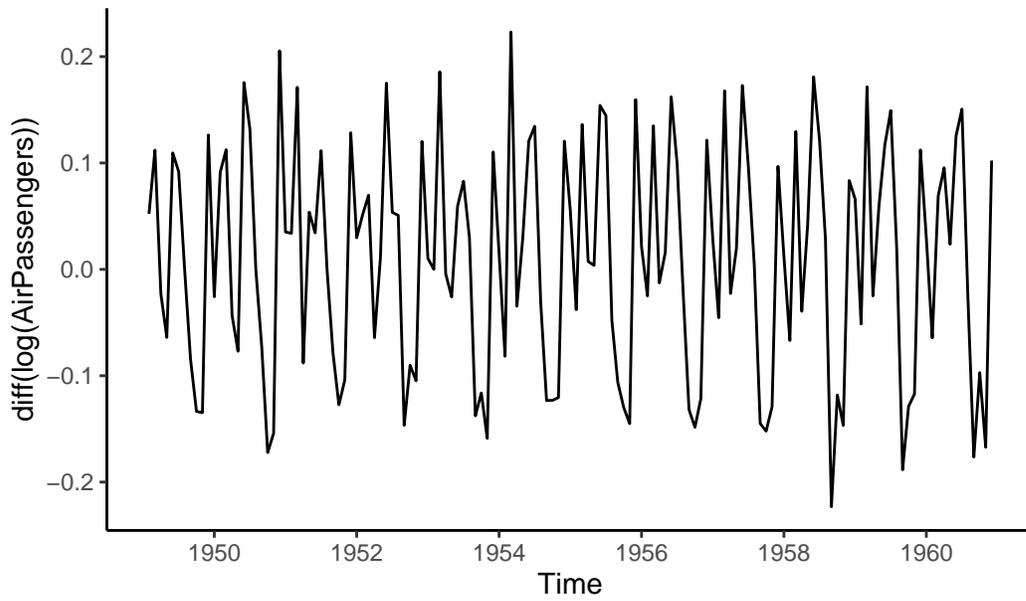
```
ndiffs(AirPassengers, alpha=0.05, test=c("adf"),  
       type=c("level"))
```

```
[1] 1
```

```
autoplot(diff(AirPassengers)) + ggplot2::theme_classic()
```



```
autoplot(diff(log(AirPassengers))) + ggplot2::theme_classic()
```



```
AR.2 <- Arima(diff(log(AirPassengers)), order = c(2,0,0))
summary(AR.2)
```

```
Series: diff(log(AirPassengers))
ARIMA(2,0,0) with non-zero mean
```

Coefficients:

	ar1	ar2	mean
	0.2359	-0.1725	0.0096
s.e.	0.0826	0.0833	0.0092

```
sigma^2 = 0.01073: log likelihood = 122.8
AIC=-237.6 AICc=-237.31 BIC=-225.75
```

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	9.490067e-05	0.1024837	0.08866769	NaN	Inf	2.568306	-0.01487092

## 4.2 Proceso de medias móviles

Un proceso de medias móviles basa la regresión de un proceso estocástico en el término error y sus retardos:

$$Y_t = c + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \dots + \theta_q u_{t-q} = c + u_t + \sum_{j=1}^q \theta_j u_{t-j}$$

donde los coeficientes  $\theta_1, \theta_2, \dots, \theta_q$  miden el efecto de los retardos del término error  $u_t, u_{t-1}, u_{t-2}, \dots, u_{t-q}$  sobre el proceso  $Y_t$ .

Al igual que el proceso autorregresivo, es necesario que la serie sea estacionaria para modelizarla como un proceso de medias móviles de orden  $q$  o MA( $q$ ).

```
MA.2 <- Arima(diff(log(AirPassengers)), order = c(0,0,2))
summary(MA.2)
```

```
Series: diff(log(AirPassengers))
ARIMA(0,0,2) with non-zero mean
```

Coefficients:

	ma1	ma2	mean
--	-----	-----	------

```

      0.2019  -0.3409  0.0096
s.e.  0.1196   0.1879  0.0073

```

```

sigma^2 = 0.0105:  log likelihood = 124.19
AIC=-240.38  AICc=-240.09  BIC=-228.53

```

Training set error measures:

```

              ME      RMSE      MAE  MPE  MAPE      MASE      ACF1
Training set 0.0001935616 0.1014078 0.08817457 NaN  Inf  2.554023 0.0358125

```

### 4.3 Modelo ARIMA

A menudo el mejor modelo para una serie temporal consiste en una combinación de un modelo AR de orden  $p$  y un modelo MA de orden  $q$ , es decir, un modelo ARMA( $p, q$ ). Además, se le añade que la serie tiene que ser estacionaria, lo cual se consigue diferenciando (cuyo inverso es la integración). Si una serie se hace estacionaria diferenciando  $d$  veces, se le denomina integrada de orden  $d$  o  $I(d)$ .

La combinación de los tres componentes da lugar al modelo ARIMA( $p, d, q$ ), que agrupa todos los componentes de los procesos estocásticos vistos hasta el momento.

$$Y_t = c \sum_{i=1}^p \phi_i Y_{t-i} + u_t + \sum_{j=1}^q \theta_j u_{t-j}$$

```

ARIMA.1 <- Arima(log(AirPassengers), order=c(2,1,2))
summary(ARIMA.1)

```

```

Series: log(AirPassengers)
ARIMA(2,1,2)

```

Coefficients:

```

      ar1      ar2      ma1      ma2
      0.2432  0.2677 -0.0867 -0.6813
s.e.  0.1674  0.1596  0.1315  0.1211

```

```

sigma^2 = 0.009891:  log likelihood = 128.89
AIC=-247.78  AICc=-247.34  BIC=-232.97

```

Training set error measures:

```

              ME      RMSE      MAE      MPE      MAPE      MASE
Training set 0.02028092 0.09771061 0.08380607 0.3511798 1.516783 0.6923716

```

ACF1

Training set 0.005928279

### 4.3.1 Diagnóstico del modelo

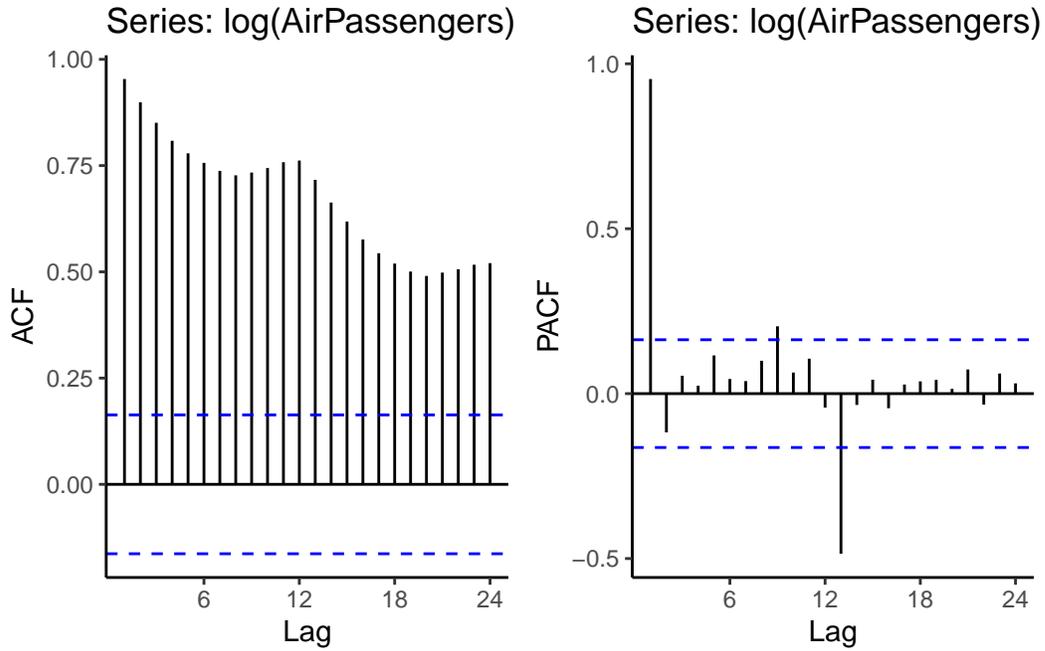
Para conocer el modelo adecuado para una serie temporal, es posible realizar un diagnóstico previo basado en el correlograma, y el **correlograma parcial**. El correlograma parcial extrae el efecto del resto de retardos que no se estén considerando  $1, 2, \dots, l - 1$  en el momento para obtener el efecto marginal del retardo  $l$  sobre el valor contemporáneo de la variable, es decir, la correlación neta entre  $y_t$  y  $y_{t-l}$ .

Existen dos patrones reversos que determinan el tipo de modelo a elegir:

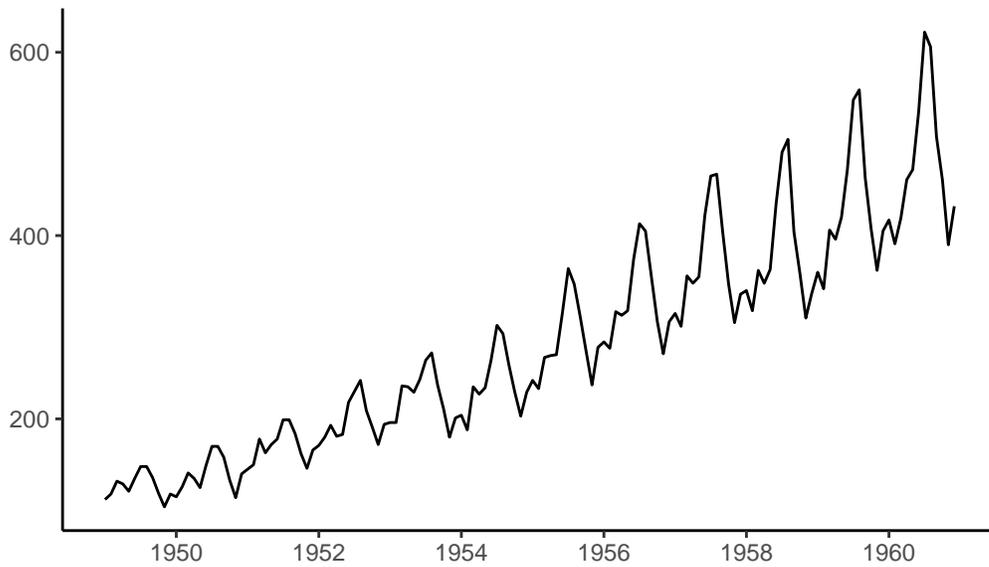
1. Un correlograma que disminuye exponencialmente y uno o varios retardos extremadamente significativos al principio del correlograma parcial. En ese caso se tiene un proceso tipo AR con orden  $p$  hasta el último retardo significativo.
2. Un correlograma parcial que disminuye exponencialmente y uno o varios retardos extremadamente significativos al principio del correlograma. En ese caso se tiene un proceso tipo MA con orden  $q$  hasta el último retardo significativo.

En el caso de tener una mezcla de los dos, el diagnóstico visual se complica, y lo mejor es recurrir a los estadísticos de ajuste y predicción del modelo.

```
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
correlograma <- ggAcf(log(AirPassengers)) + theme_classic()
correlograma.parcial <- ggPacf(log(AirPassengers)) +
  theme_classic()
gridExtra::grid.arrange(correlograma, correlograma.parcial, nrow=1)
```



```
autoplot(AirPassengers) + xlab("") + ylab("") +  
  theme_classic()
```



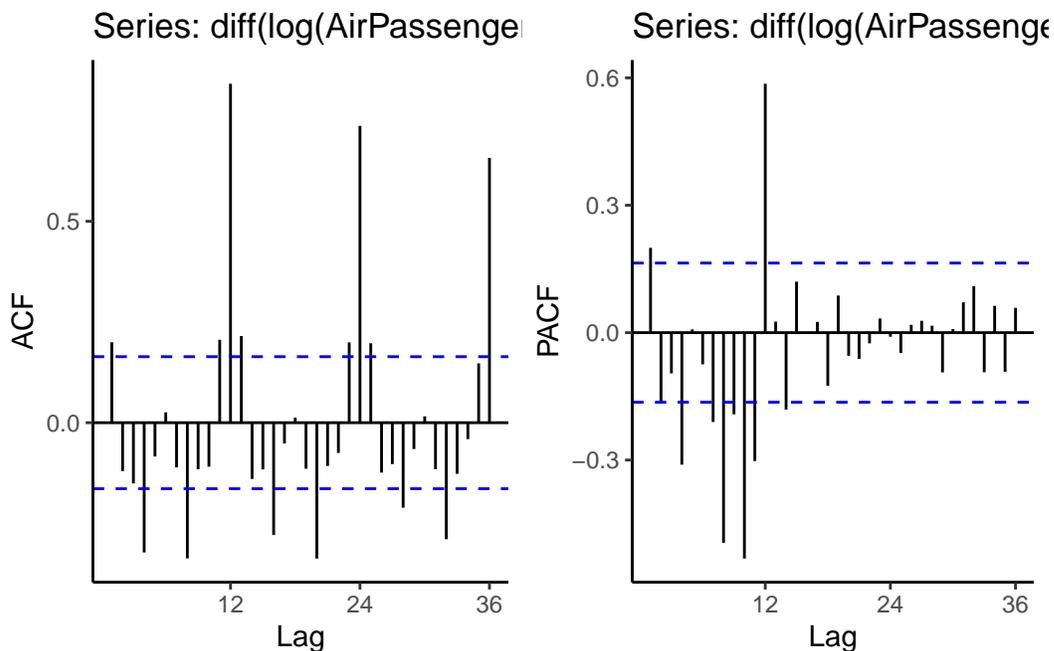
En este caso, el correlograma (ACF) decrece exponencialmente, y hay un retardo significativo de los primeros del correlograma parcial. Además, para que la serie sea estacionaria hay que tomar  $d = 1$  diferencias. De este modo, se tiene que el modelo ARIMA(0,1,1) ajustaría bien la serie.

### 4.3.2 Componente estacional

El modelo ARIMA puede incorporar componentes estacionales para adaptarse a series con este tipo de patrón. Al igual que la diferenciación estacional, denotada por  $D$ , también se pueden incluir términos autorregresivos de y medias móviles que se aplican sobre el periodo estacional, con orden  $P$  y  $Q$ . Siendo  $a$  la fracción del año correspondiente, el modelo es entonces ARIMA  $(p, d, q)(P, D, Q)_a$ .

El diagnóstico del modelo para los componentes AR y MA de la dimensión estacional es equivalente al de la dimensión no estacional, pero ahora los retardos se repetirán en la fracción de año con la que se trabaje.

```
correlograma.est <- ggAcf(diff(log(AirPassengers)), lag.max = 36) +
  theme_classic()
correlograma.parcial.est <- ggPacf(diff(log(AirPassengers)), lag.max = 36) +
  theme_classic()
gridExtra::grid.arrange(correlograma.est, correlograma.parcial.est,
  nrow=1)
```



Cada 12 retardos, en el correlograma existe un retardo significativo, pero el valor va disminuyendo geométricamente. En el correlograma parcial, hay un retardo significativo en el orden 12, y ninguno en los órdenes 24 y 36. Esto indica un componente estacional de tipo autorregresivo y orden  $P = 1$ .

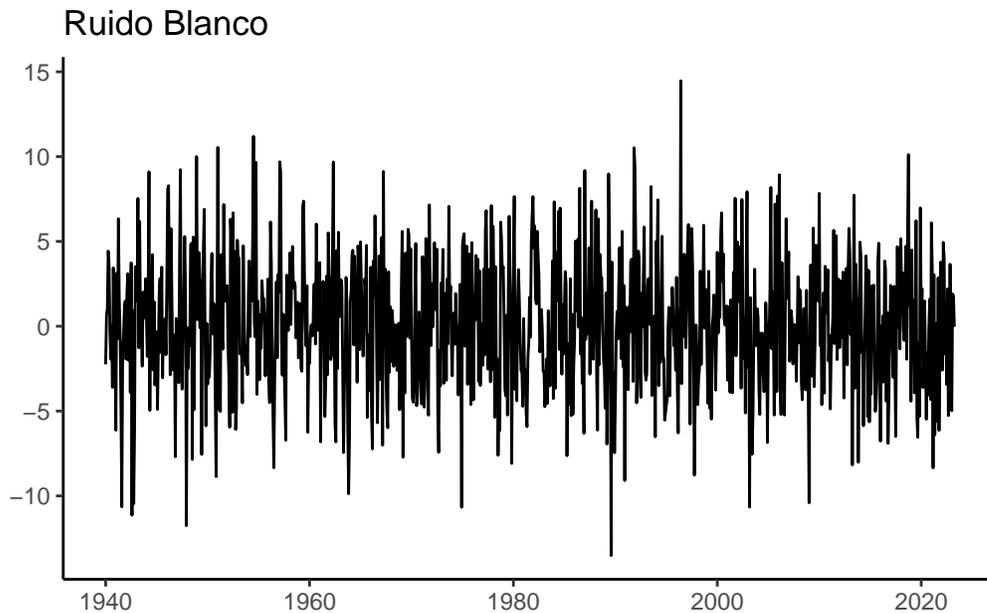
## 4.4 Proceso de ruido blanco

Un caso específico del modelo ARIMA es cuando su orden es  $(0,0,0)$ . Entonces, el proceso es aleatorio por definición, y se conoce como ruido blanco:

$$Y_t = u_t$$

El proceso de ruido blanco es estacionario por definición, ya que su media es cero, su varianza es constante, y no existe correlación entre ninguno de sus retardos.

```
library(ggplot2, quietly=TRUE)
x=rnorm(1000, mean=0, sd=4)
x2 = ts(x, start=c(1940,1), frequency = 12)
autoplot(x2) + xlab("") + ylab("") +
  ggtitle("Ruido Blanco") + theme_classic()
```

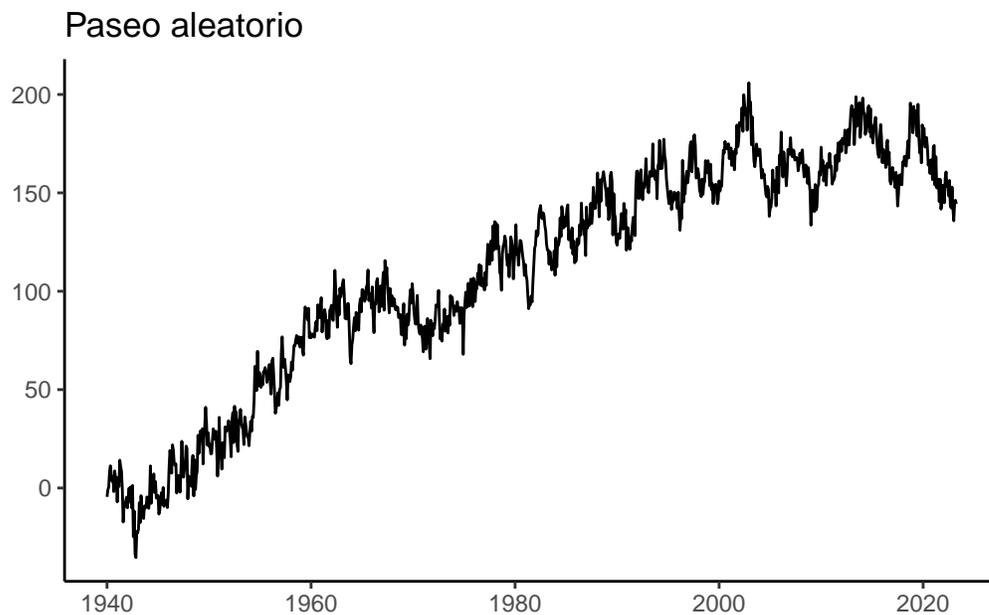


## 4.5 Paseo aleatorio

Otro caso especial es el modelo ARIMA(0,1,0), que se denomina paseo aleatorio. Es estacionario al tomar primeras diferencias de la serie. Su forma es:

$$Y_t = Y_{t-1} + u_t$$

```
u = cumsum(x)
y = x + u
y2 = ts(y, start=c(1940,1), frequency = 12)
autoplot(y2) + xlab("") + ylab("") +
  ggtitle("Paseo aleatorio") + theme_classic()
```



Si, además, el proceso posee una constante, entonces se le denomina paseo aleatorio con deriva.

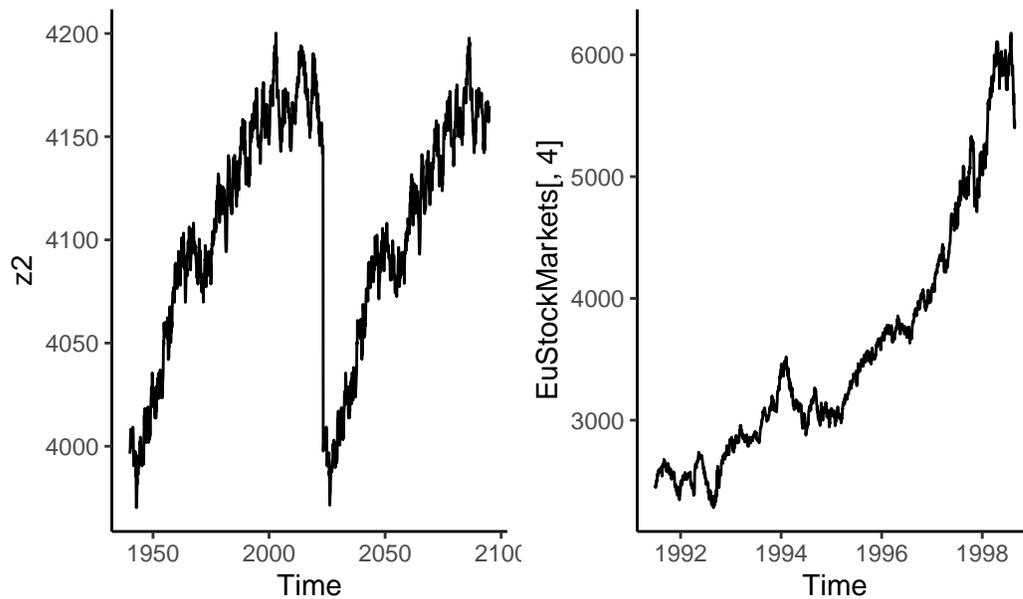
$$Y_t = c + Y_{t-1} + u_t$$

```
w = rnorm(1860)
m <- cumsum(x)
z = 4000 + w + m
z2 = ts(z, start=c(1940,1), frequency = 12)
autoplot(z2) + xlab("") + ylab("") +
  ggtitle("Paseo aleatorio con deriva") + theme_classic()
```



Cuyo comportamiento no difiere demasiado de una serie temporal financiera, como la de de un índice bursátil.

```
data("EuStockMarkets")
gridExtra::grid.arrange(autoplot(z2) + theme_classic(), autoplot(EuStockMarkets[,4]) + theme...
```



## 4.6 Predicción con el modelo ARIMA

Los modelos ARIMA son una potente herramienta para la predicción debido a que permiten predecir una serie temporal con solo su propia trayectoria. Es decir, no necesitan de muestreo u obtención de datos adicionales para conocer los factores estructurales que afectan a la serie.

Para obtener predicciones con un modelo ARIMA, basta con sustituir los valores obtenidos en cada periodo de forma recursiva para el horizonte  $h = 1, 2, \dots, H$ .

$$Y_{t+h} = c + \phi_1 Y_{t+h-1} + \phi_2 Y_{t+h-2} + \dots + \phi_p Y_{t+h-p} + u_t + \theta_1 u_{t+h-1} + \theta_2 u_{t+h-2} + \dots + \theta_q u_{t+h-q}$$

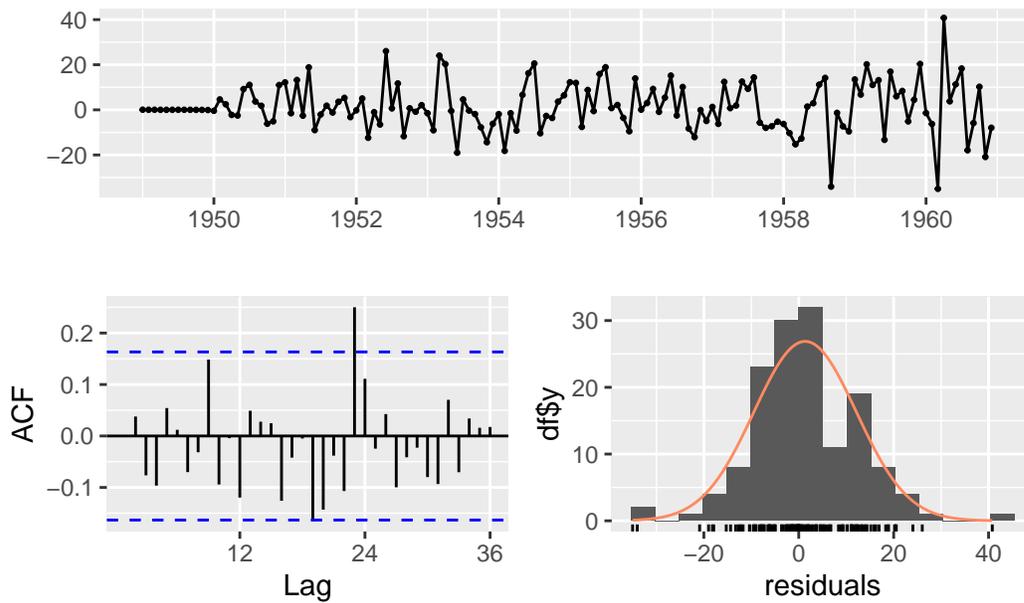
De forma que para el  $h = 1$

$$Y_{t+1} = c + \phi_1 Y_t + \phi_2 Y_{t-1} + \dots + \phi_p Y_{t+1-p} + u_t + \theta_1 u_t + \theta_2 u_{t-1} + \dots + \theta_q u_{t+1-q}$$

y así sucesivamente.

```
data("AirPassengers")
library(forecast, quietly=TRUE)
modeloauto <- auto.arima(AirPassengers)
checkresiduals(modeloauto)
```

Residuals from ARIMA(2,1,1)(0,1,0)[12]

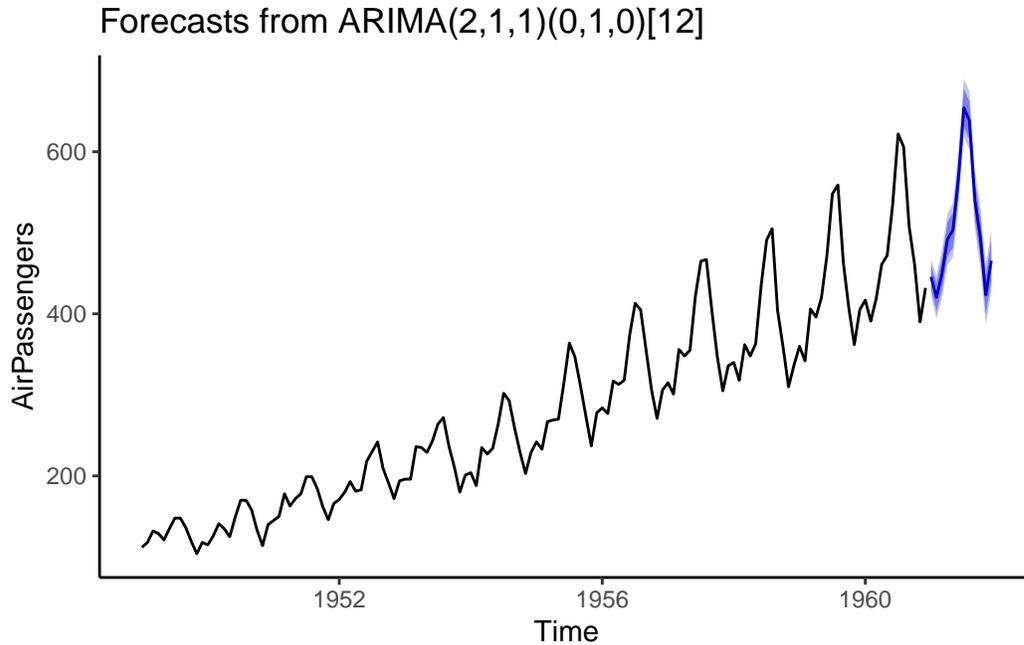


Ljung-Box test

```
data: Residuals from ARIMA(2,1,1)(0,1,0)[12]
Q* = 37.784, df = 21, p-value = 0.01366
```

```
Model df: 3. Total lags used: 24
```

```
autoplot(forecast(modeloauto, h=12)) +
  ggplot2::theme_classic()
```



#### 4.6.1 Evaluación de las predicciones

Para evaluar la calidad de las predicciones de un modelo, hay algunos estadísticos de referencia.

- Error Medio (EM). La media de la desviación entre las observaciones  $Y_t$  y las predicciones del modelo  $\hat{Y}_t$

$$\frac{\sum_{t=1}^T Y_t - \hat{Y}_t}{T}$$

- Raíz del Error Cuadrático Medio (RECM). Igual que el EM pero estandarizado por unidad de medida

$$\sqrt{\frac{\sum_{t=1}^T (Y_t - \hat{Y}_t)^2}{T}}$$

- Error Medio Absoluto (EMA). Igual que el EM pero con estandarización por signos.

$$\frac{\sum_{t=1}^T |Y_t - \hat{Y}_t|}{T}$$

```
summary(modeloauto)
```

```
Series: AirPassengers
ARIMA(2,1,1)(0,1,0) [12]
```

Coefficients:

```
      ar1      ar2      ma1
      0.5960  0.2143 -0.9819
s.e.  0.0888  0.0880  0.0292
```

```
sigma^2 = 132.3: log likelihood = -504.92
AIC=1017.85  AICc=1018.17  BIC=1029.35
```

Training set error measures:

```
              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set  1.3423  10.84619  7.86754  0.420698  2.800458  0.245628 -0.00124847
```

```
modelonoauto <- Arima(AirPassengers, order=c(0,1,1), seasonal=c(1,1,0))
summary(modelonoauto)
```

```
Series: AirPassengers
ARIMA(0,1,1)(1,1,0) [12]
```

Coefficients:

```
      ma1      sar1
      -0.3096 -0.1459
s.e.  0.0885  0.0974
```

```
sigma^2 = 136.8: log likelihood = -507.21
AIC=1020.43  AICc=1020.61  BIC=1029.05
```

Training set error measures:

```
              ME      RMSE      MAE      MPE      MAPE      MASE
Training set  0.3454169  11.07003  8.108353  0.04835617  2.871605  0.2531463
              ACF1
Training set -0.003449102
```

Sin embargo, lo ideal es dejar algunas observaciones fuera de la estimación del modelo como conjunto de prueba, y medir los estadísticos para ver como se ajusta el modelo a una situación real. Medir los estadísticos dentro del mismo conjunto que se usa para la estimación del modelo

puede resultar erróneo, ya que el modelo puede ajustar bien los datos, pero no predecir bien situaciones nuevas.

**Ejercicio: Estime el mejor modelo ARIMA que considere para los datos “gas”, y prediga con un horizonte de 12 periodos.**

## 5 Modelizando volatilidad

Hasta ahora, los modelos presentados modelizan el comportamiento de una variable mediante una combinación lineal de parámetros, variables y término error:

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \dots + \beta_k X_{k,t} + u_t$$

Sin embargo, muchas series temporales, como las financieras, presentan un fenómeno conocido como **clustering de volatilidad**. Esto implica que las series temporales tienen variabilidad cambiante con periodos de alta y baja volatilidad caracterizados por su persistencia.

### 5.1 Modelo GARCH

Esta volatilidad cambiante se puede modelizar a través de la varianza del término error de una serie temporal:

$$Y_t = \beta_0 + \dots + e_t$$

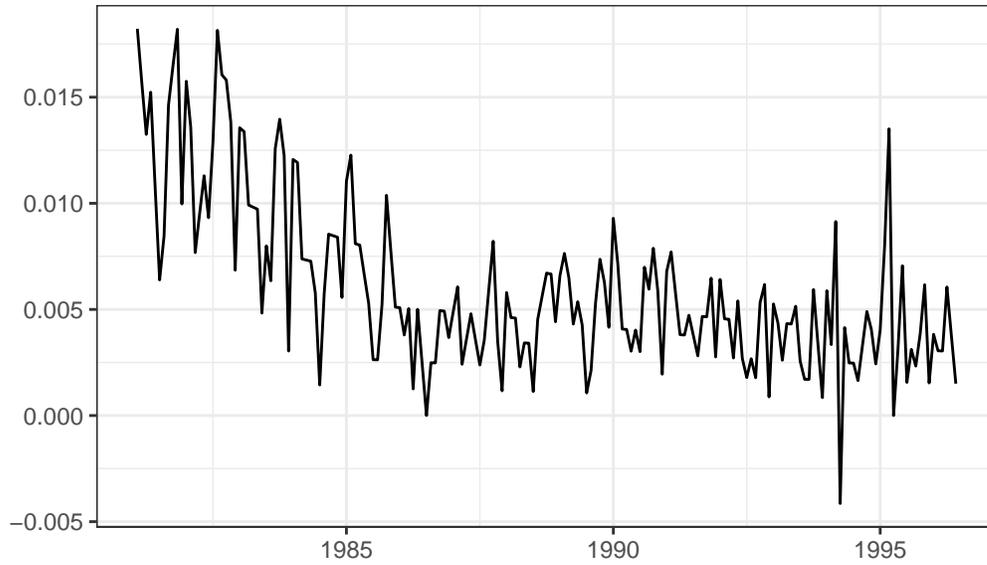
$$e_t \sim \mathcal{N}(0, v_t)$$

$$v_t = \alpha_0 + \alpha_1 e_{t-1}^2 + \alpha_2 e_{t-2}^2 + \dots + \alpha_p e_{t-p}^2 + \beta_1 v_{t-1} + \beta_2 v_{t-2} + \dots + \beta_q v_{t-q}$$

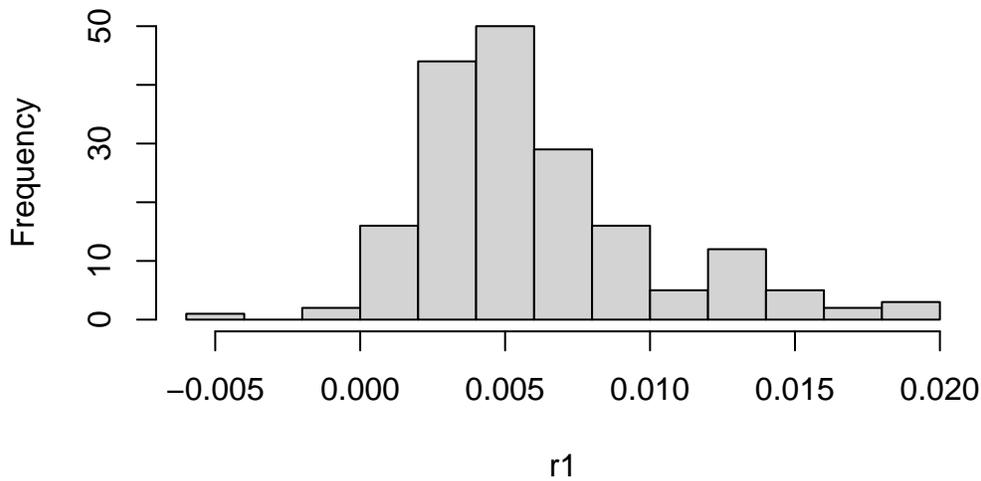
donde ahora la media del término error es constante y cero, pero la varianza es variable y depende de los errores al cuadrado retardados  $p$  periodos (término autorregresivo) y de las perturbaciones aleatorias retardadas  $q$  periodos (término de medias móviles).

Sin embargo, para que este modelo tenga sentido, lo primero es testar que la serie presente efectos ARCH en sus errores. A menudo, series con distribuciones de probabilidad de colas pesadas y mucha variabilidad suelen presentarlos.

```
library(forecast, quietly = T)
library(Ecdat, quietly = T)
library(ggplot2, quietly=T)
data("PPP")
r1 <- diff(PPP[,1])
autoplot(r1) + xlab("") + ylab("") + theme_bw()
```



```
hist(r1, main="")
```



Sin embargo, como siempre un test estadístico formal es necesario para contrastarlo de forma fiable. El test prueba si los parámetros  $\alpha_1, \alpha_2, \dots, \alpha_j$  de la regresión:

$$e_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + \alpha_2 e_{t-2}^2 + \dots + \alpha_j e_{t-j}^2 + \epsilon_t$$

son conjuntamente significativos mediante un estadístico multiplicador de Lagrange (LM), con hipótesis nula  $\alpha_1, \alpha_2, \dots, \alpha_j = 0$  e hipótesis alternativa de lo contrario.

```
library(FinTS, quietly = T)
ArchTest(r1) #Por defecto, j=12.
```

ARCH LM-test; Null hypothesis: no ARCH effects

```
data: r1
Chi-squared = 111.23, df = 12, p-value < 2.2e-16
```

Para modelizar la serie, la elección del orden ( $p, q$ ) de los parámetros se puede hacer por prueba error, analizando cuales de los mismos son estadísticamente significativos. Sin embargo, una opción estandar suele ser el modelo GARCH(1,1).

```
library(rugarch, quietly = T)
especificacion <- ugarchspec(
  variance.model=list(model="sGARCH",
                      garchOrder=c(1,1)),
  mean.model=list(armaOrder=c(1,0)),
  distribution.model="std")
ajuste <- ugarchfit(spec=especificacion, data=r1)
ajuste
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(1,0,0)
Distribution   : std
```

Optimal Parameters

```
-----
      Estimate Std. Error t value Pr(>|t|)
mu      0.004984  0.000439 11.34215 0.000000
ar1     0.538940  0.079915  6.74390 0.000000
omega   0.000000  0.000002  0.16057 0.872432
alpha1  0.075840  0.058460  1.29729 0.194531
beta1   0.889464  0.059334 14.99082 0.000000
shape   8.160328  4.027879  2.02596 0.042769
```

Robust Standard Errors:

```
      Estimate Std. Error t value Pr(>|t|)
mu      0.004984  0.001877  2.655733 0.007914
ar1     0.538940  0.414437  1.300415 0.193459
omega   0.000000  0.000026  0.009621 0.992324
alpha1  0.075840  0.725487  0.104537 0.916744
beta1   0.889464  0.777960  1.143328 0.252902
shape   8.160328 16.195748  0.503856 0.614362
```

LogLikelihood : 821.4546

Information Criteria

```
-----
Akaike      -8.8157
Bayes      -8.7113
Shibata    -8.8177
Hannan-Quinn -8.7734
```

Weighted Ljung-Box Test on Standardized Residuals

```
-----
                        statistic p-value
Lag[1]                  1.440 0.23010
Lag[2*(p+q)+(p+q)-1] [2] 1.464 0.45773
Lag[4*(p+q)+(p+q)-1] [5] 5.232 0.08831
d.o.f=1
H0 : No serial correlation
```

Weighted Ljung-Box Test on Standardized Squared Residuals

```
-----
                        statistic p-value
Lag[1]                  2.267 0.1322
Lag[2*(p+q)+(p+q)-1] [5] 3.020 0.4034
Lag[4*(p+q)+(p+q)-1] [9] 3.755 0.6305
d.o.f=2
```

Weighted ARCH LM Tests

```
-----
                        Statistic Shape Scale P-Value
ARCH Lag[3]    0.01748 0.500 2.000 0.8948
ARCH Lag[5]    1.07364 1.440 1.667 0.7112
ARCH Lag[7]    1.37046 2.315 1.543 0.8469
```

Nyblom stability test

```
-----
Joint Statistic: 53.6379
Individual Statistics:
mu      0.81013
ar1     2.24817
omega   19.88440
alpha1  0.10444
beta1   0.09910
shape   0.06491
```

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic:            1.49 1.68 2.12  
 Individual Statistic:    0.35 0.47 0.75

Sign Bias Test

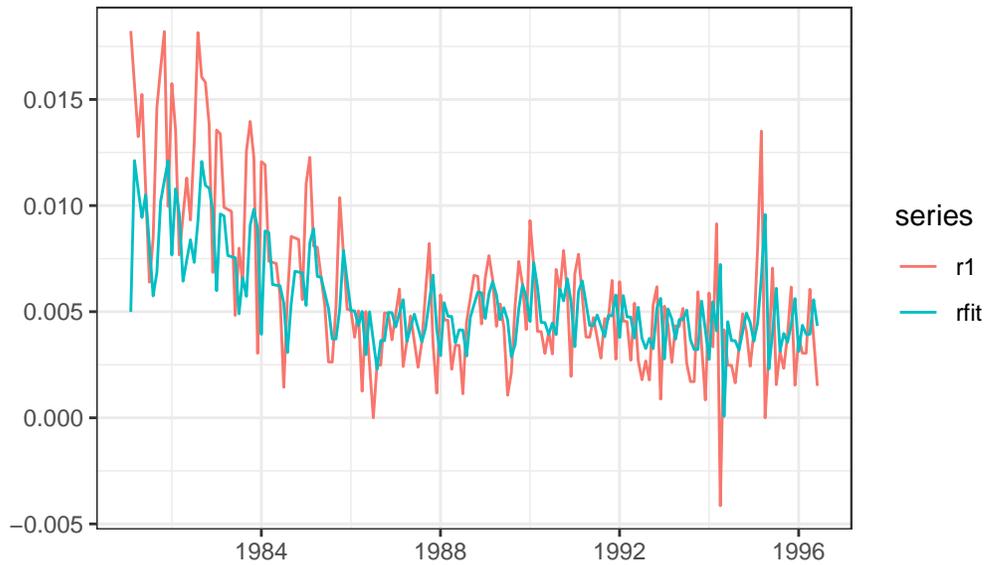
```
-----
                t-value  prob sig
Sign Bias      0.5919 0.5546
Negative Sign Bias 0.3373 0.7363
Positive Sign Bias 2.2463 0.0259 **
Joint Effect   5.5338 0.1366
```

Adjusted Pearson Goodness-of-Fit Test:

```
-----
group statistic p-value(g-1)
1    20    16.19    0.64462
2    30    31.16    0.35782
3    40    49.59    0.11910
4    50    67.70    0.03947
```

Elapsed time : 0.1023111

```
rfit <- ts(ajuste@fit$fitted.values,
           frequency = 12, start=c(1981,2))
estimacion <- ts.union(r1, rfit)
autoplot(estimacion) + xlab("") + ylab("") +
  theme_bw()
```



Una vez estimado el modelo, se pueden realizar predicciones.

```
submuestra = ugarchfit(data = r1,
                      spec = especificacion, out.sample=4)
prediccion = ugarchforecast(submuestra, n.ahead=4)
prediccion
```

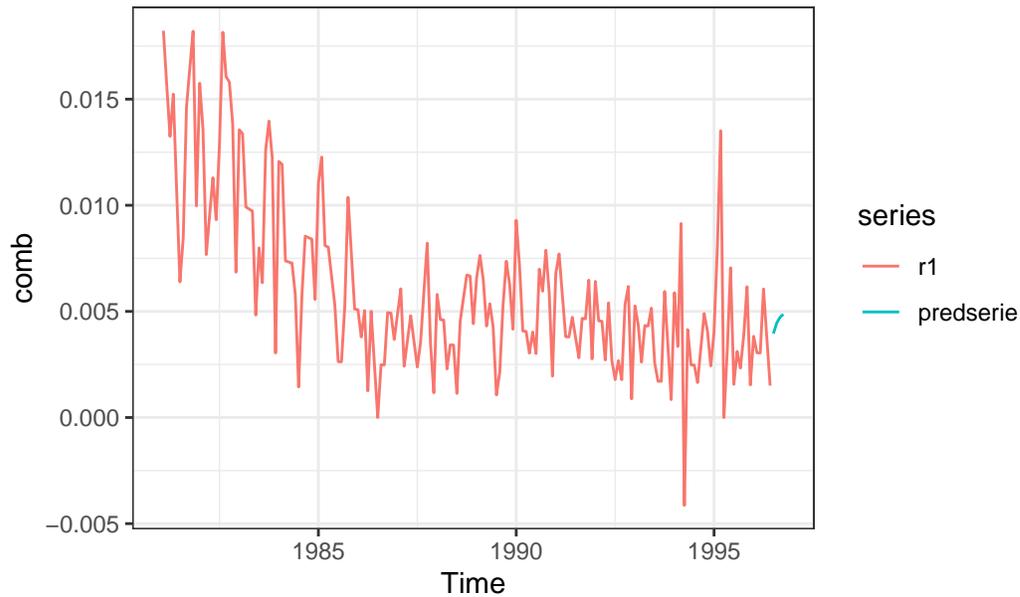
```
*-----*
*      GARCH Model Forecast      *
*-----*
Model: sGARCH
Horizon: 4
Roll Steps: 0
Out of Sample: 4

0-roll forecast [T0=feb. 1996]:
      Series      Sigma
T+1 0.003954 0.003128
T+2 0.004448 0.003117
T+3 0.004718 0.003106
T+4 0.004866 0.003095
```

```

predserie <- ts(prediccion@forecast$seriesFor,
               start=c(1996,07), frequency = 12)
comb <- ts.union(r1,predserie)
autoplot(comb) + theme_bw()

```



## 5.2 t-GARCH

Desde la proposición de la versión original, se han estudiado numerosas variantes del modelo GARCH para adaptarse a las necesidades del análisis o investigación.

Una de las variantes más conocidas es la suposición de que los errores siguen una distribución t de student en lugar de una distribución normal. La distribución t de student tiene colas más pesadas que la normal, con lo cual, asigna más probabilidad a sucesos extremos. Esto permite ajustar mejor series que son más tendentes de lo normal a situarse en esos valores, como las financieras. Este modelo se denomina **t-GARCH**.

$$Y_t = \beta_0 + \dots + e_t$$

$$e_t \sim t(0, v_t)$$

$$v_t = \alpha_0 + \sum_{i=1}^p \alpha_i e_{t-i}^2 + \sum_{j=1}^q \beta_j v_{t-j}$$

```
library(rugarch, quietly = T)
library(forecast, quietly=T)
library(ggplot2, quietly=T)
tgarchspec <- ugarchspec(
  variance.model=list(model="sGARCH",
                      garchOrder=c(1,1),
                      submodel="TGARCH"),
  mean.model=list(armaOrder=c(1,0)),
  distribution.model="std")
data("dji30ret")
datostgarch <- ts(dji30ret$AA, start=c(1987,03),
                 frequency=365)
ajustetgarch <- ugarchfit(spec=tgarchspec,
                          data=datostgarch)
ajustetgarch
```

```
*-----*
*          GARCH Model Fit          *
*-----*
```

Conditional Variance Dynamics

```
-----
GARCH Model : sGARCH(1,1)
Mean Model  : ARFIMA(1,0,0)
Distribution : std
```

Optimal Parameters

```
-----
      Estimate Std. Error t value Pr(>|t|)
mu      0.000348   0.000238   1.4624 0.143619
ar1     0.047709   0.013474   3.5409 0.000399
omega   0.000003   0.000002   1.2581 0.208366
alpha1  0.044389   0.009370   4.7372 0.000002
beta1   0.949991   0.011147  85.2235 0.000000
shape   6.924101   0.476246  14.5389 0.000000
```

Robust Standard Errors:

## Introducción a la Econometría Financiera

	Estimate	Std. Error	t value	Pr(> t )
mu	0.000348	0.000237	1.46797	0.142112
ar1	0.047709	0.014519	3.28611	0.001016
omega	0.000003	0.000011	0.26822	0.788526
alpha1	0.044389	0.046098	0.96293	0.335584
beta1	0.949991	0.053982	17.59826	0.000000
shape	6.924101	1.600889	4.32516	0.000015

LogLikelihood : 13987.92

### Information Criteria

-----

Akaike	-5.0650
Bayes	-5.0578
Shibata	-5.0650
Hannan-Quinn	-5.0625

### Weighted Ljung-Box Test on Standardized Residuals

-----

	statistic	p-value
Lag[1]	2.327	0.127151
Lag[2*(p+q)+(p+q)-1] [2]	4.335	0.002034
Lag[4*(p+q)+(p+q)-1] [5]	8.779	0.003640

d.o.f=1  
H0 : No serial correlation

### Weighted Ljung-Box Test on Standardized Squared Residuals

-----

	statistic	p-value
Lag[1]	0.678	0.4103
Lag[2*(p+q)+(p+q)-1] [5]	1.631	0.7077
Lag[4*(p+q)+(p+q)-1] [9]	2.776	0.7953

d.o.f=2

### Weighted ARCH LM Tests

-----

	Statistic	Shape	Scale	P-Value
ARCH Lag[3]	0.3353	0.500	2.000	0.5625
ARCH Lag[5]	1.1947	1.440	1.667	0.6761
ARCH Lag[7]	1.8742	2.315	1.543	0.7439

Nyblom stability test

-----  
 Joint Statistic: 88.403

Individual Statistics:

mu 0.09014  
 ar1 0.82669  
 omega 15.95110  
 alpha1 1.07076  
 beta1 0.85877  
 shape 0.58268

Asymptotic Critical Values (10% 5% 1%)

Joint Statistic: 1.49 1.68 2.12

Individual Statistic: 0.35 0.47 0.75

Sign Bias Test

-----  

	t-value	prob	sig
Sign Bias	1.135	0.25622	
Negative Sign Bias	2.012	0.04429	**
Positive Sign Bias	1.035	0.30056	
Joint Effect	5.328	0.14930	

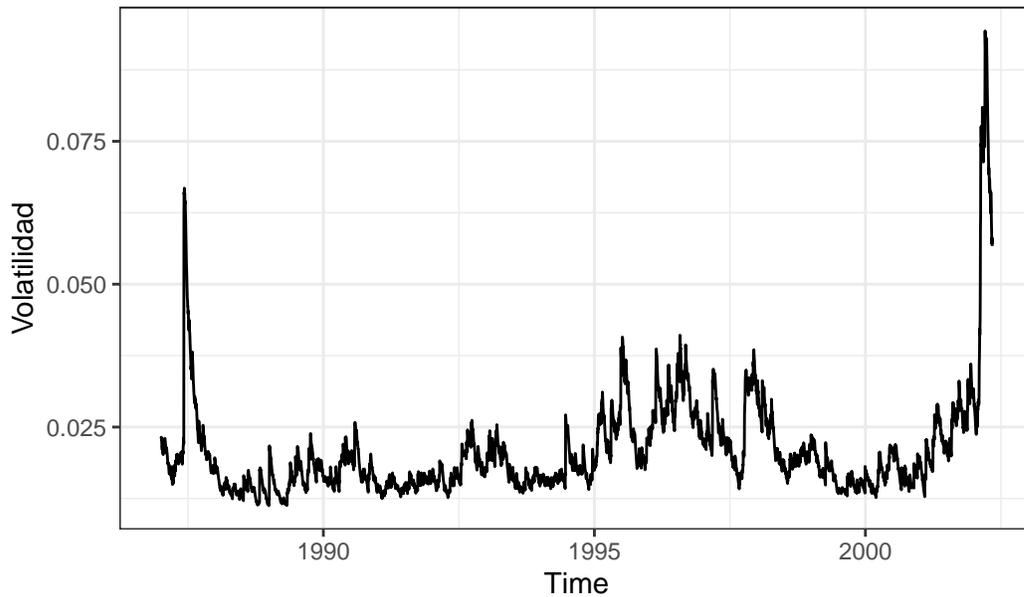
Adjusted Pearson Goodness-of-Fit Test:

-----  

group	statistic	p-value(g-1)	
1	20	29.51	0.05844
2	30	47.46	0.01669
3	40	50.40	0.10435
4	50	68.76	0.03270

Elapsed time : 1.002222

```
volatilidad <- ts(ajustetgarch@fit$sigma, start =c(1987, 03), frequency = 360)
autoplot(volatilidad) + ylab("Volatilidad") +
  theme_bw()
```



Este tipo de variaciones abre la puerta a un universo de modelos univariantes donde varía la distribución de probabilidad, el tipo de estimación, etc., pero donde solo se trabaja con la propia serie temporal. Algunos de estos modelos son el EGARCH, AGARCH, etc.

### 5.3 Modelos GARCH multivariantes

El modelo GARCH puede ampliarse para recoger el efecto que otras variables o series temporales puedan tener sobre la varianza de la perturbación aleatoria de la serie objetivo. En este caso, se tiene:

$$Y_t = \beta_0 + \dots + e_t$$

$$e_t \sim \mathcal{D}(0, V_t)$$

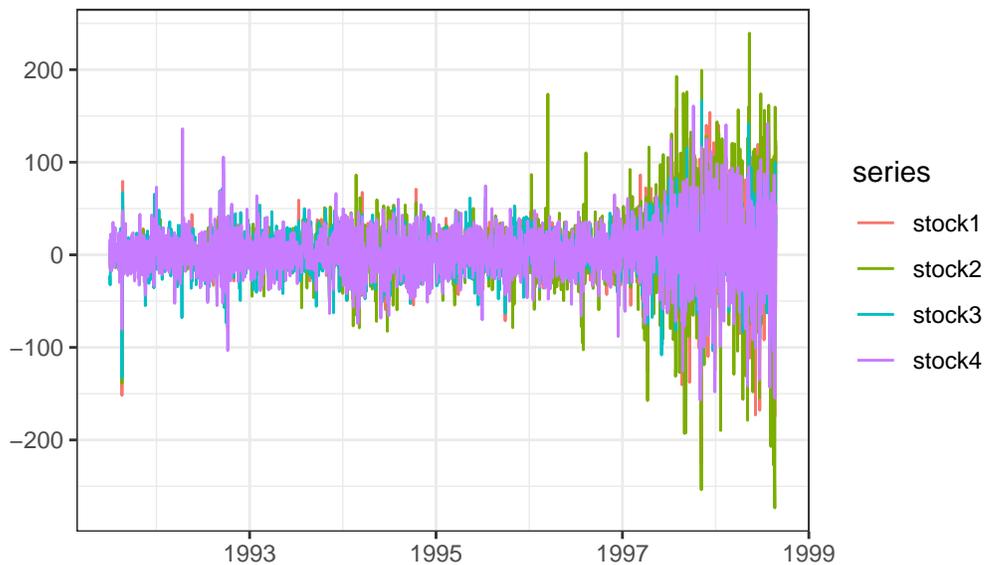
donde  $\mathcal{D}$  es una distribución de probabilidad cualquiera, habitualmente la normal o t de Student. Al modelizar la matriz de varianzas-covarianzas de la perturbación aleatoria del modelo de esta forma, es posible capturar la variación de la volatilidad entre las distintas series temporales.

Al igual que en el caso univariante, este marco abre la puerta a un universo de modelos para distintas formas de  $V_t$ . El más utilizado de ellos es el DCC-GARCH, que parte de la idea de descomponer la matriz de varianzas-covarianzas en:

$$V_t = D_t R_t D_t$$

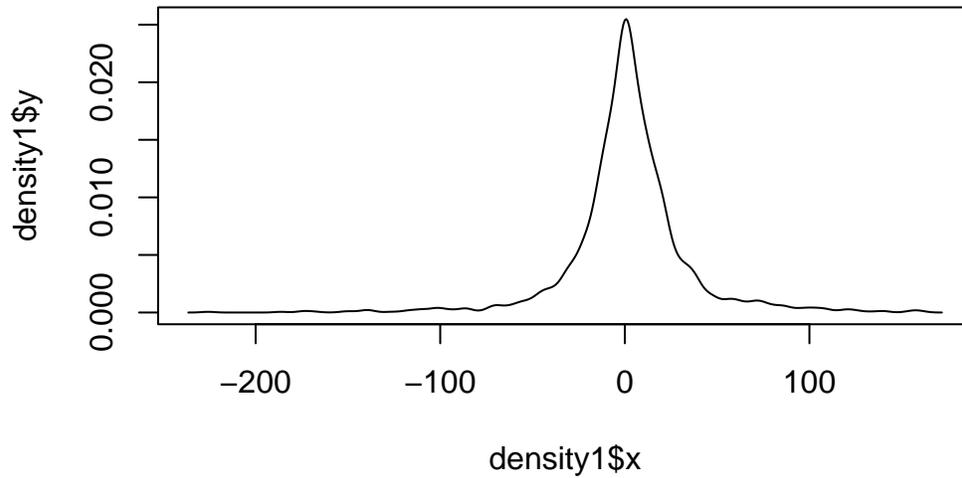
donde  $D_t$  es una matriz de desviaciones típicas, y  $R_t$  es una matriz de correlaciones, ambas de las perturbaciones aleatorias de las distintas series temporales.

```
data("EuStockMarkets")
stock1 <- diff(EuStockMarkets[,1])
stock2 <- diff(EuStockMarkets[,2])
stock3 <- diff(EuStockMarkets[,3])
stock4 <- diff(EuStockMarkets[,4])
stocks <- ts.union(stock1, stock2, stock3, stock4)
autoplot(stocks) + xlab("") + ylab("") +
  theme_bw()
```

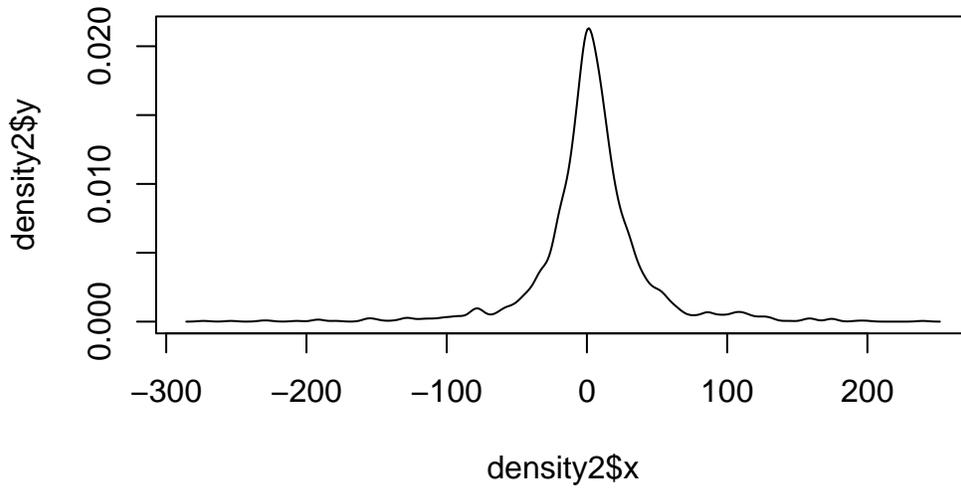


```
library(rugarch, quietly=T)
library(rmgarch, quietly=T)
mgarch1 <- ugarchspec(mean.model=list(armaOrder=c(0,0), include.mean=F),
  variance.model = list(model="sGARCH",
```

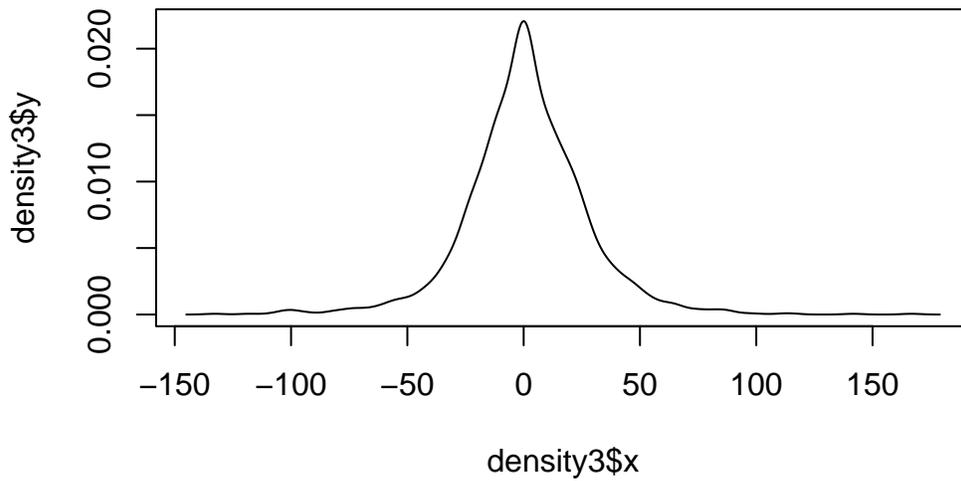
```
garchOrder=c(1,1)),  
distribution.model="std")  
density1 <- density(stocks[,1])  
plot(density1$x, density1$y, type="l")
```



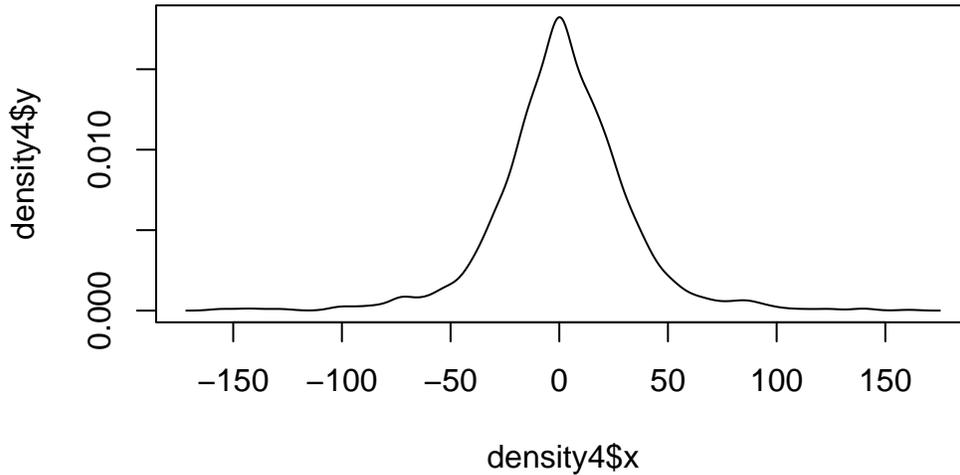
```
density2 <- density(stocks[,2])  
plot(density2$x, density2$y, type="l")
```



```
density3 <- density(stocks[,3])  
plot(density3$x, density3$y, type="l")
```



```
density4 <- density(stocks[,4])
plot(density4$x, density4$y, type="l")
```



```
mgarch2 <- dccspec(uspec = multispec(replicate(4, mgarch1)),
  dccOrder = c(1,1),distribution = "mvt")
mgarchestimado <- dccfit(mgarch2, data=stocks)
mgarchestimado
```

```
*-----*
*           DCC GARCH Fit           *
*-----*
```

```
Distribution      : mvt
Model             : DCC(1,1)
No. Parameters    : 25
[VAR GARCH DCC UncQ] : [0+16+3+6]
No. Series        : 4
No. Obs.          : 1859
Log-Likelihood    : -32294.54
Av.Log-Likelihood : -17.37
```

Optimal Parameters

	Estimate	Std. Error	t value	Pr(> t )
[stock1].omega	3.767383	2.553540	1.4754	0.140117
[stock1].alpha1	0.085510	0.018802	4.5478	0.000005
[stock1].beta1	0.913490	0.022265	41.0280	0.000000
[stock1].shape	6.144515	0.923601	6.6528	0.000000
[stock2].omega	7.916024	3.987486	1.9852	0.047120
[stock2].alpha1	0.109077	0.027248	4.0032	0.000063
[stock2].beta1	0.889923	0.029269	30.4050	0.000000
[stock2].shape	5.923411	0.889703	6.6577	0.000000
[stock3].omega	2.032128	1.232564	1.6487	0.099209
[stock3].alpha1	0.035330	0.003300	10.7047	0.000000
[stock3].beta1	0.963376	0.001255	767.3413	0.000000
[stock3].shape	8.371992	1.713368	4.8863	0.000001
[stock4].omega	2.147005	1.482623	1.4481	0.147586
[stock4].alpha1	0.036623	0.002966	12.3466	0.000000
[stock4].beta1	0.962377	0.001284	749.3365	0.000000
[stock4].shape	9.566493	2.024745	4.7248	0.000002
[Joint]dcca1	0.027964	0.004322	6.4696	0.000000
[Joint]dccb1	0.933977	0.012950	72.1212	0.000000
[Joint]mshape	7.890250	0.660005	11.9548	0.000000

Information Criteria

Akaike	34.771
Bayes	34.845
Shibata	34.771
Hannan-Quinn	34.798

Elapsed time : 5.696136

```
mgarchcov <- (mgarchestimado@mfit$H[1:1,2:2,])
mgarchcov <- ts(mgarchcov, start=c(1991, 131),
               end = c(1998, 169), frequency=260)
autoplot(mgarchcov) + xlab("") + ylab("") + theme_bw()
```



```
prediccionmgarch <- dccforecast(mgarchestimado, n.ahead=20)
```

**Ejercicio:** Cargue, analice y estime un modelo GARCH para la transformación que considere de la variable “tcm1y” del del conjunto de datos tcm del paquete tseries. Extraiga la volatilidad estimada de la serie.

## 6 Cointegración y shocks

### 6.1 Cointegración

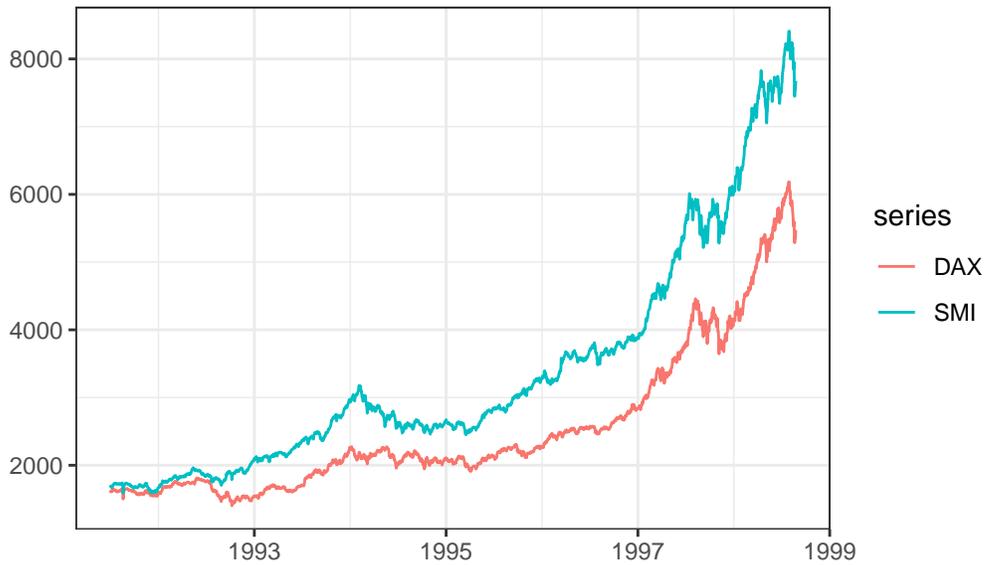
Hasta ahora se ha visto que una serie es estacionaria si es  $I(0)$ , estacionaria tras realizar primeras diferencias si es  $I(1)$ , y así sucesivamente hasta  $I(d)$ .

Sin embargo, también es posible que la diferencia entre dos series sea estacionaria:

$$Z_t = Y_t - \alpha X_t = I(0)$$

Este concepto se denomina **cointegración**, término que denota el que la diferenciación de dos series sea estacionaria. A efectos prácticos, esto implica que las series **comueven** a largo plazo ya que comparten una tendencia estocástica común. Esto implica que se mueven a la par con desviaciones temporales a corto plazo.

```
data("EuStockMarkets")
library(forecast, quietly=TRUE)
library(ggplot2, quietly=TRUE)
autoplot(EuStockMarkets[,1:2]) + xlab("") +
  ylab("") + theme_bw()
```



Es bastante evidente que cuando una serie aumenta, la otra también lo hace y viceversa.

```
library(tseries)
library(tidyverse)
lapply(EuStockMarkets[,1:2], kps.test)
```

\$DAX

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 15.401, Truncation lag parameter = 8, p-value = 0.01

\$SMI

KPSS Test for Level Stationarity

data: X[[i]]

KPSS Level = 16.457, Truncation lag parameter = 8, p-value = 0.01

Ninguna de las dos series estacionaria. Sin embargo, una combinación lineal de las variables puede serlo.

```
reg <- lm(EuStockMarkets[,1] ~ EuStockMarkets[,2],
          data=EuStockMarkets)
summary(reg)
```

Call:

```
lm(formula = EuStockMarkets[, 1] ~ EuStockMarkets[, 2], data = EuStockMarkets)
```

Residuals:

Min	1Q	Median	3Q	Max
-285.75	-106.88	-20.15	104.20	603.45

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.478e+02	7.558e+00	46.02	<2e-16 ***
EuStockMarkets[, 2]	6.465e-01	2.008e-03	321.91	<2e-16 ***

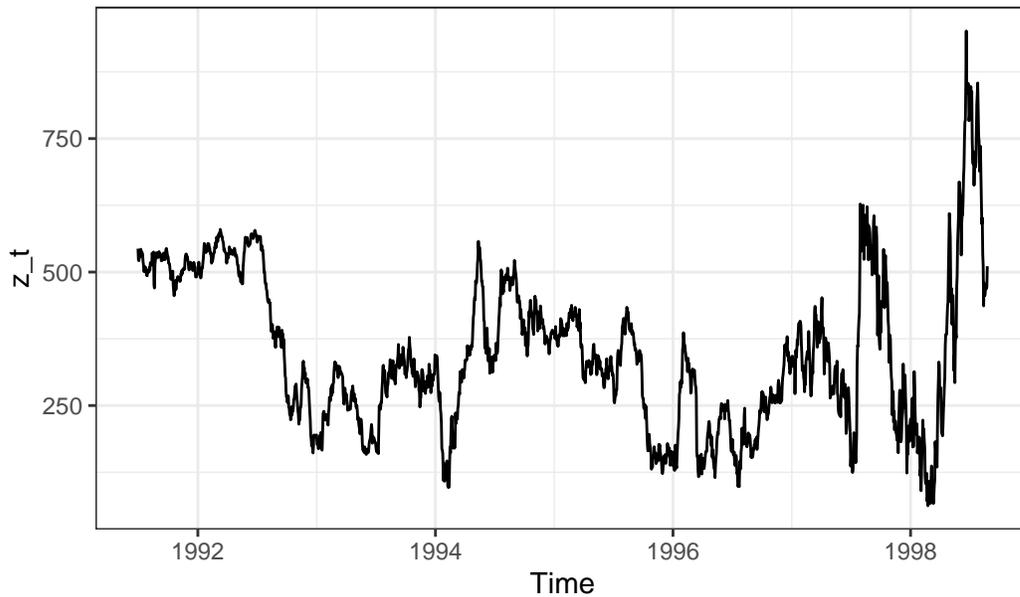
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 144 on 1858 degrees of freedom

Multiple R-squared: 0.9824, Adjusted R-squared: 0.9824

F-statistic: 1.036e+05 on 1 and 1858 DF, p-value: < 2.2e-16

```
z_t = EuStockMarkets[,1] - 6.465e-01*EuStockMarkets[,2]
autoplot(z_t) + theme_bw()
```



```
kpss.test(z_t)
```

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

```
      KPSS Test for Level Stationarity
```

```
data: z_t
```

```
KPSS Level = 2.0542, Truncation lag parameter = 8, p-value = 0.01
```

```
adf.test(z_t)
```

```
Augmented Dickey-Fuller Test
```

```
data: z_t
```

```
Dickey-Fuller = -3.3581, Lag order = 12, p-value = 0.06065
```

```
alternative hypothesis: stationary
```

## 6.2 Modelo ADL

El modelo autorregresivo de rezagos distribuidos o ADL proporciona información sobre el efecto de una serie temporal en el movimiento de otra, al incorporar su propia persistencia mediante autorregresiones.

$$Y_t = c + \sum_{i=1}^p \alpha_i Y_{t-i} + \sum_{j=0}^q \beta_j X_{t-j} + e_t$$

En el caso de que dos series estén cointegradas, no es necesario diferenciarlas.

```
y_t <- EuStockMarkets[,1]
x_t <- EuStockMarkets[,2]
x_t1 <- stats::lag(x_t, 1)
x_t2 <- stats::lag(x_t,2)
w_t = ts.union(x_t, x_t1, x_t2)
w_t = w_t[1:1860,]
adl <- Arima(y_t, order=c(2,0,0),xreg=w_t)
summary(adl)
```

Series: y\_t

Regression with ARIMA(2,0,0) errors

Coefficients:

	ar1	ar2	intercept	x_t	x_t1	x_t2
	1.0061	-0.0086	778.6044	-0.0270	-0.0670	0.6159
s.e.	0.0233	0.0233	186.7688	0.0128	0.0128	0.0128

sigma<sup>2</sup> = 458.4: log likelihood = -8328.79

AIC=16671.59 AICc=16671.65 BIC=16710.28

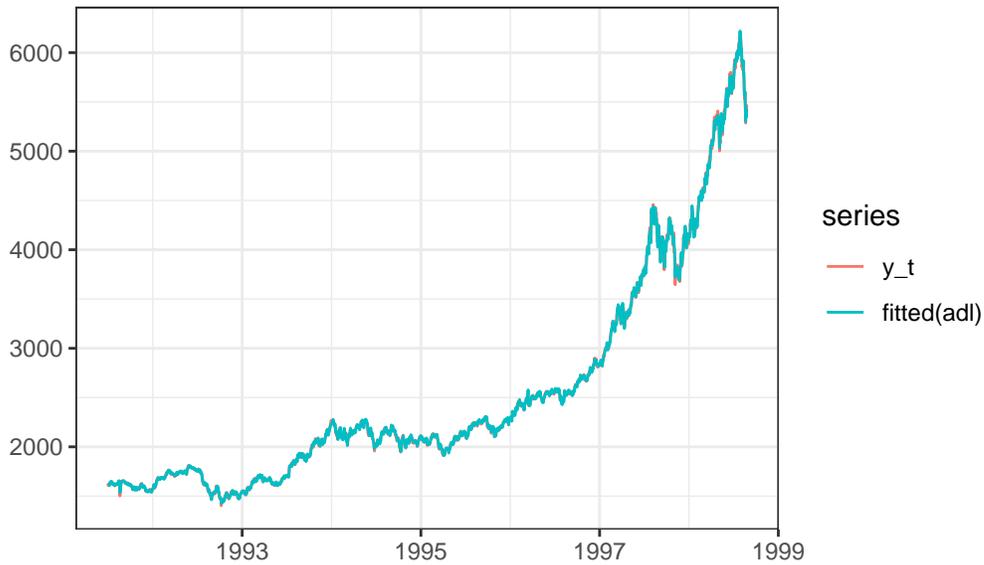
Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
Training set	0.3595568	21.37653	14.18575	-0.003301388	0.5421216	0.02581027

ACF1

Training set 5.368272e-05

```
tsframe <- ts.union(y_t, fitted(adl))
autoplot(tsframe) + xlab("") + ylab("") +
  theme_bw()
```



El modelo ADL es extensible a cualquier número de predictores.

**Ejercicio:** Empleando el conjunto de datos `ice.river` del paquete `tseries`, analice la potencial cointegración que pueda existir entre la temperatura y el caudal del río Jokulsa Eystri river.

### 6.3 El modelo VAR

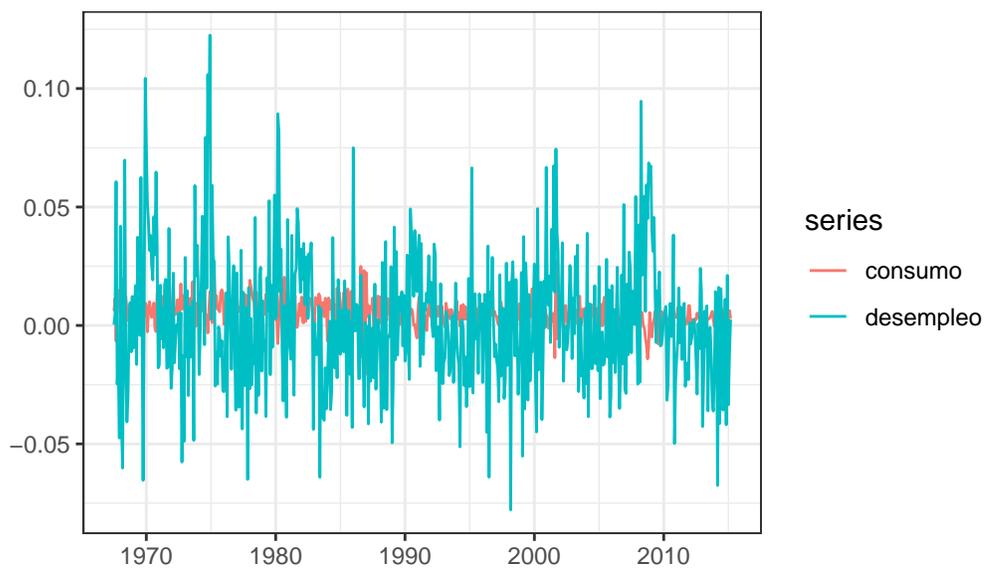
Existen numerosas ocasiones en las cuales las variables guardan relaciones bidireccionales entre sí. Es decir, no solo  $Y$  afecta a  $X$ , sino que  $X$  también afecta a  $Y$ . Los modelos empleados hasta ahora, no son capaces de capturar dicho comportamiento.

Sin embargo, el modelo de Vectores Autorregresivos o VAR, si que lo permite:

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \alpha_2 X_{t-1} + \epsilon_t X_t = \beta_0 + \beta_1 X_{t-1} + \beta_2 Y_{t-1} + u_t$$

donde se ha escogido un número de variables  $k = 2$  y de retardos  $p = 1$  por sencillez, pero siendo el modelo generalizable a cualquier conjunto  $k = 1, \dots, K$  de variables aleatorias y  $p = 1, \dots, P$  retardos.

```
library(ggplot2, quietly=T)
library(forecast, quietly=T)
data("economics")
consumo = economics$pce %>% log %>%
  diff %>% ts(start=c(1967, 07), frequency=12)
desempleo = economics$unemploy %>% log %>%
  diff %>% ts(start=c(1967, 07), frequency=12)
datos <- ts.union(consumo, desempleo)
autoplot(datos) + xlab("") + ylab("") +
  theme_bw()
```



```
library(vars, quietly=T)
VARselect(datos, lag.max = 13, type=c("const"))
```

\$selection

AIC(n)	HQ(n)	SC(n)	FPE(n)
12	6	3	12

\$criteria

	1	2	3	4	5
AIC(n)	-1.764123e+01	-1.770928e+01	-1.776039e+01	-1.777839e+01	-1.780951e+01
HQ(n)	-1.762313e+01	-1.767910e+01	-1.771814e+01	-1.772407e+01	-1.774312e+01

Introducción a la Econometría Financiera

SC(n)	-1.759486e+01	-1.763200e+01	-1.765219e+01	-1.763928e+01	-1.763949e+01
FPE(n)	2.180266e-08	2.036843e-08	1.935365e-08	1.900834e-08	1.842598e-08
	6	7	8	9	10
AIC(n)	-1.782655e+01	-1.782352e+01	-1.784221e+01	-1.784945e+01	-1.785167e+01
HQ(n)	-1.774809e+01	-1.773298e+01	-1.773961e+01	-1.773478e+01	-1.772492e+01
SC(n)	-1.762561e+01	-1.759166e+01	-1.757945e+01	-1.755577e+01	-1.752707e+01
FPE(n)	1.811488e-08	1.817007e-08	1.783369e-08	1.770536e-08	1.766650e-08
	11	12	13		
AIC(n)	-1.785775e+01	-1.786676e+01	-1.786633e+01		
HQ(n)	-1.771894e+01	-1.771587e+01	-1.770337e+01		
SC(n)	-1.750224e+01	-1.748034e+01	-1.744900e+01		
FPE(n)	1.755968e-08	1.740272e-08	1.741063e-08		

```
modeloVAR <- VAR(datos, p=12, type=c("const"))
summary(modeloVAR)
```

VAR Estimation Results:

=====

Endogenous variables: consumo, desempleo

Deterministic variables: const

Sample size: 561

Log Likelihood: 3469.588

Roots of the characteristic polynomial:

0.9744 0.876 0.876 0.8687 0.8687 0.8497 0.8497 0.8473 0.8473 0.8385 0.8385 0.8334 0.8334 0.8334 0.8334 0.8334 0.8334 0.8334 0.8334 0.8334

Call:

VAR(y = datos, p = 12, type = c("const"))

Estimation results for equation consumo:

=====

consumo = consumo.l1 + desempleo.l1 + consumo.l2 + desempleo.l2 + consumo.l3 + desempleo.l3 + consumo.l4 + desempleo.l4

	Estimate	Std. Error	t value	Pr(> t )	
consumo.l1	-0.1917412	0.0432314	-4.435	1.12e-05	***
desempleo.l1	-0.0150701	0.0085397	-1.765	0.07818	.
consumo.l2	-0.0055244	0.0445757	-0.124	0.90142	
desempleo.l2	-0.0076812	0.0084468	-0.909	0.36357	
consumo.l3	0.0641227	0.0447336	1.433	0.15232	
desempleo.l3	-0.0122587	0.0084621	-1.449	0.14802	
consumo.l4	0.0626403	0.0445335	1.407	0.16013	
desempleo.l4	-0.0027378	0.0084704	-0.323	0.74666	

Introducción a la Econometría Financiera

consumo.l5	0.0792834	0.0439719	1.803	0.07194	.
desempleo.l5	0.0102157	0.0084667	1.207	0.22813	.
consumo.l6	0.0997613	0.0439380	2.271	0.02357	*
desempleo.l6	0.0014219	0.0085310	0.167	0.86769	.
consumo.l7	0.0475148	0.0437779	1.085	0.27825	.
desempleo.l7	0.0216564	0.0085378	2.537	0.01148	*
consumo.l8	0.1403817	0.0436131	3.219	0.00137	**
desempleo.l8	0.0164071	0.0085382	1.922	0.05518	.
consumo.l9	0.1412484	0.0442244	3.194	0.00149	**
desempleo.l9	0.0051179	0.0084911	0.603	0.54694	.
consumo.l10	0.0876676	0.0443292	1.978	0.04848	*
desempleo.l10	-0.0052164	0.0083954	-0.621	0.53464	.
consumo.l11	0.1336436	0.0446833	2.991	0.00291	**
desempleo.l11	0.0044546	0.0082578	0.539	0.58980	.
consumo.l12	0.0582583	0.0438372	1.329	0.18442	.
desempleo.l12	0.0105982	0.0081193	1.305	0.19234	.
const	0.0014107	0.0005789	2.437	0.01514	*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.005018 on 536 degrees of freedom

Multiple R-Squared: 0.1635, Adjusted R-squared: 0.126

F-statistic: 4.365 on 24 and 536 DF, p-value: 7.645e-11

Estimation results for equation desempleo:

=====

desempleo = consumo.l1 + desempleo.l1 + consumo.l2 + desempleo.l2 + consumo.l3 + desempleo.l3 + consumo.l4 + desempleo.l4 + consumo.l5 + desempleo.l5 + consumo.l6 + desempleo.l6

	Estimate	Std. Error	t value	Pr(> t )	
consumo.l1	-1.0139801	0.2179173	-4.653	4.13e-06	***
desempleo.l1	-0.0418458	0.0430464	-0.972	0.331435	.
consumo.l2	-0.7896301	0.2246937	-3.514	0.000478	***
desempleo.l2	0.1177857	0.0425778	2.766	0.005864	**
consumo.l3	-0.4793588	0.2254893	-2.126	0.033971	*
desempleo.l3	0.1175444	0.0426549	2.756	0.006056	**
consumo.l4	-0.2097369	0.2244811	-0.934	0.350560	.
desempleo.l4	0.1105201	0.0426971	2.588	0.009902	**
consumo.l5	0.3617449	0.2216500	1.632	0.103255	.
desempleo.l5	0.1270671	0.0426783	2.977	0.003039	**
consumo.l6	0.1628471	0.2214791	0.735	0.462496	.
desempleo.l6	0.1151596	0.0430026	2.678	0.007634	**

```

consumo.l7      0.3460996  0.2206723  1.568 0.117381
desempleo.l7   0.0596034  0.0430364  1.385 0.166643
consumo.l8     0.5643426  0.2198412  2.567 0.010527 *
desempleo.l8   0.0746248  0.0430387  1.734 0.083511 .
consumo.l9     0.3576799  0.2229229  1.605 0.109193
desempleo.l9   0.0731534  0.0428014  1.709 0.088005 .
consumo.l10    0.4932116  0.2234512  2.207 0.027720 *
desempleo.l10 -0.0241842  0.0423191 -0.571 0.567919
consumo.l11    0.3282575  0.2252362  1.457 0.145594
desempleo.l11  0.0232047  0.0416252  0.557 0.577441
consumo.l12   -0.0022385  0.2209711 -0.010 0.991921
desempleo.l12 -0.1289334  0.0409270 -3.150 0.001722 **
const         -0.0001761  0.0029182 -0.060 0.951911
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 0.0253 on 536 degrees of freedom  
Multiple R-Squared: 0.2166, Adjusted R-squared: 0.1815  
F-statistic: 6.176 on 24 and 536 DF, p-value: < 2.2e-16

Covariance matrix of residuals:

	consumo	desempleo
consumo	2.518e-05	-1.283e-05
desempleo	-1.283e-05	6.399e-04

Correlation matrix of residuals:

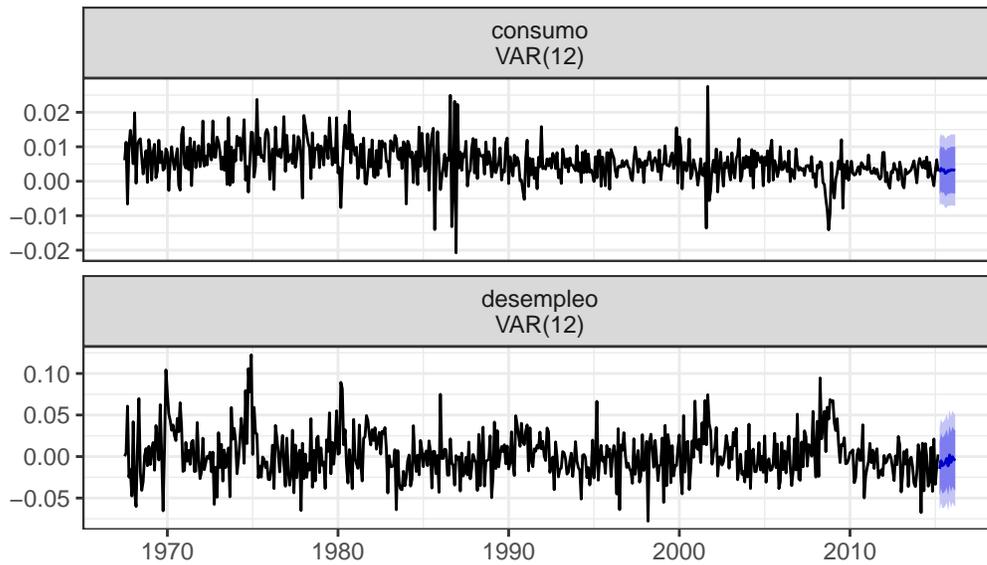
	consumo	desempleo
consumo	1.000	-0.101
desempleo	-0.101	1.000

Una vez estimado un modelo VAR, se pueden hacer predicciones. Habitualmente, debido a las relaciones de endogeneidad entre variables que permite, el modelo VAR supera a la mayoría de los modelos en la predicción.

```

predicciones <- predict(modeloVAR, h=12)
forecast(modeloVAR, h=12) %>% autoplot() +
  xlab("") + theme_bw()

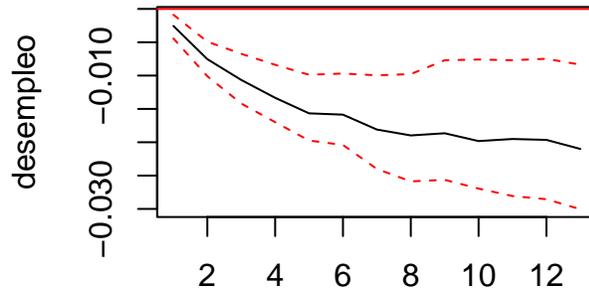
```



Además, se puede realizar un análisis estructural del modelo. En primer lugar, se pueden obtener las funciones de respuesta al impulso, que miden cuanto responderá una variable respuesta ante una innovación o shock de una desviación típica en la variable de impulso.

```
impulsosconsumo <- irf(modeloVAR, impulse = "consumo", response = "desempleo", n.ahead=12, cumulative=TRUE, boot = TRUE, runs=100)
plot(impulsosconsumo)
```

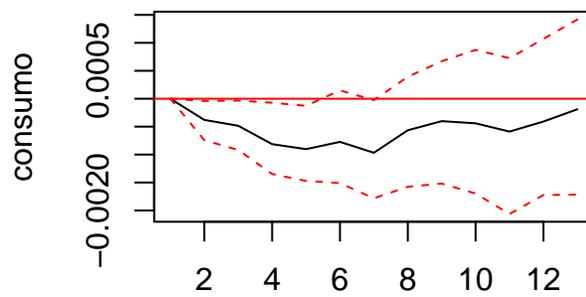
Orthogonal Impulse Response from consumo (cumulative)



95 % Bootstrap CI, 100 runs

```
impulsosdesempleo <- irf(modeloVAR, impulse = "desempleo", response = "consumo", n.ahead=12,  
                          cumulative=TRUE, boot = TRUE, runs=100)  
plot(impulsosdesempleo)
```

Orthogonal Impulse Response from desempleo (cumulative)



95 % Bootstrap CI, 100 runs

También se puede ejecutar una descomposición de la varianza predicha, con lo que se obtiene una idea de cuanto afecta la varianza del error de una variable en la otra.

```
descomposicion <- fevd(modeloVAR, n.ahead=24)
descomposicion$consumo
```

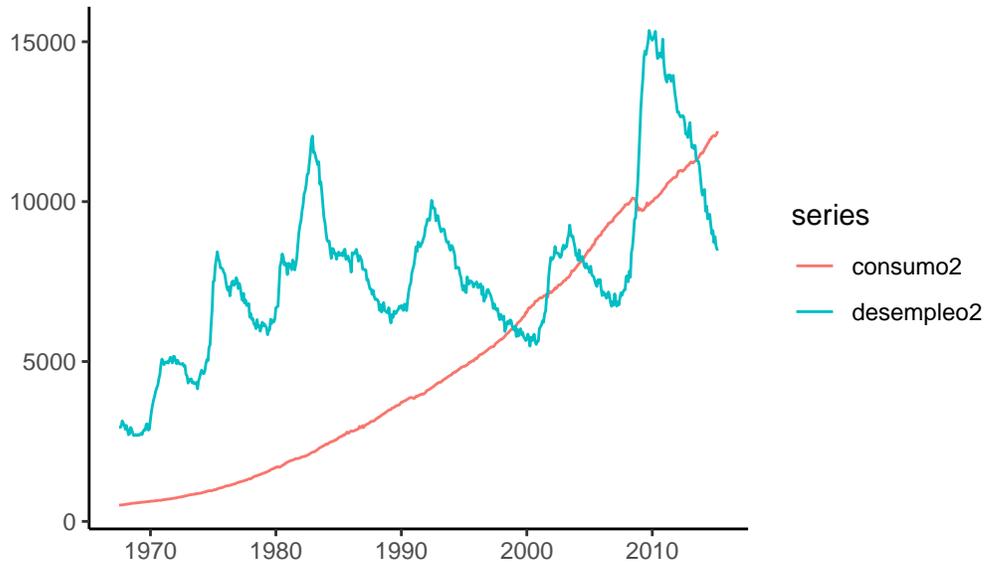
```
      consumo  desempleo
[1,] 1.0000000 0.000000000
[2,] 0.9945059 0.005494142
[3,] 0.9941029 0.005897102
[4,] 0.9900707 0.009929277
[5,] 0.9898099 0.010190130
[6,] 0.9892396 0.010760440
[7,] 0.9879176 0.012082419
[8,] 0.9820339 0.017966140
[9,] 0.9813073 0.018692669
[10,] 0.9813763 0.018623718
[11,] 0.9806586 0.019341387
[12,] 0.9798168 0.020183192
[13,] 0.9781804 0.021819564
[14,] 0.9778629 0.022137053
[15,] 0.9775448 0.022455199
[16,] 0.9766376 0.023362361
[17,] 0.9750619 0.024938050
[18,] 0.9745766 0.025423434
[19,] 0.9734312 0.026568814
[20,] 0.9730773 0.026922664
[21,] 0.9723008 0.027699218
[22,] 0.9716707 0.028329259
[23,] 0.9706824 0.029317616
[24,] 0.9699543 0.030045723
```

### 6.3.1 Modelos VECM

Si dos series temporales están cointegradas, en el largo plazo tenderán a coincidir. Sin embargo, en el corto plazo, existirán desviaciones. El modelo de Vector de Corrección de Error ,VECM por sus siglas en inglés, explota este hecho, incluyendo coeficientes a largo plazo, y desviaciones a corto plazo.

$$Y_t = \alpha_0 + \alpha_1(Y_{t-1} - \alpha_2 X_{t-1}) + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta X_{t-1} + \epsilon_t X_t = \beta_0 + \beta_1(X_{t-1} - \beta_2 Y_{t-1}) + \delta_1 \Delta X_{t-1} + \delta_2 \Delta Y_{t-1} + u_t$$

```
consumo2 = economics$pce %>% ts(start=c(1967, 07), frequency=12)
desempleo2 = economics$unemploy %>%
  ts(start=c(1967, 07), frequency=12)
datos2 <- ts.union(consumo2, desempleo2)
autoplot(datos2) + xlab("") + ylab("") +
  theme_classic()
```



```
test1 <- ur.df(consumo2)
summary(test1)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

```
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-173.072	-4.184	3.631	13.383	172.732

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
z.lag.1	0.0035417	0.0002348	15.084	<2e-16 ***
z.diff.lag	0.0204848	0.0420327	0.487	0.626

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.53 on 570 degrees of freedom

Multiple R-squared: 0.4185, Adjusted R-squared: 0.4164

F-statistic: 205.1 on 2 and 570 DF, p-value: < 2.2e-16

Value of test-statistic is: 15.0842

Critical values for test statistics:

	1pct	5pct	10pct
tau1	-2.58	-1.95	-1.62

```
test2 <- ur.df(diff(consumo2))
summary(test2)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-122.508	-3.863	5.044	17.788	205.334

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
z.lag.1	-0.33426	0.04099	-8.154	2.26e-15 ***

```
z.diff.lag -0.40846    0.03847 -10.617 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 27.61 on 569 degrees of freedom
Multiple R-squared:  0.402, Adjusted R-squared:  0.3999
F-statistic: 191.2 on 2 and 569 DF,  p-value: < 2.2e-16
```

Value of test-statistic is: -8.1539

```
Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

```
test3 <- ur.df(desempleo2)
summary(test3)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

```
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

```
Residuals:
      Min       1Q   Median       3Q      Max
-840.67 -126.91  -0.81  125.19  789.87
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
z.lag.1    0.0002688   0.0010842   0.248   0.804
z.diff.lag 0.1833904   0.0412105   4.450 1.03e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 212.8 on 570 degrees of freedom
Multiple R-squared:  0.0339,    Adjusted R-squared:  0.03051
```

F-statistic: 10 on 2 and 570 DF, p-value: 5.382e-05

Value of test-statistic is: 0.2479

Critical values for test statistics:

1pct 5pct 10pct  
tau1 -2.58 -1.95 -1.62

```
test4 <- ur.df(diff(desempleo2))
summary(test4)
```

```
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####
```

Test regression none

Call:

```
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

Residuals:

Min	1Q	Median	3Q	Max
-792.17	-118.18	-0.02	123.88	695.86

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
z.lag.1	-0.59759	0.05161	-11.580	< 2e-16 ***
z.diff.lag	-0.26782	0.04040	-6.629	7.85e-11 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 205.2 on 569 degrees of freedom

Multiple R-squared: 0.4505, Adjusted R-squared: 0.4486

F-statistic: 233.3 on 2 and 569 DF, p-value: < 2.2e-16

Value of test-statistic is: -11.5795

Critical values for test statistics:

```
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

```
johansen <- ca.jo(datos2, type="eigen", K=2)
summary(johansen)
```

```
#####
# Johansen-Procedure #
#####
```

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

```
Eigenvalues (lambda):
[1] 0.107137493 0.005512336
```

Values of teststatistic and critical values of test:

```
      test 10pct  5pct  1pct
r <= 1 |  3.16  6.50  8.18 11.65
r = 0  | 64.82 12.91 14.90 19.19
```

Eigenvectors, normalised to first column:  
(These are the cointegration relations)

```
      consumo2.12 desempleo2.12
consumo2.12      1.0000000      1.000000
desempleo2.12 -0.1984701     -2.602472
```

Weights W:  
(This is the loading matrix)

```
      consumo2.12 desempleo2.12
consumo2.d      0.002732568 -6.749625e-05
desempleo2.d    0.002403782  2.819766e-03
```

```
modeloVECM <- cajorls(johansen, r=1)
modeloVECM
```

```
$rlm
```

Call:

```
lm(formula = substitute(form1), data = data.mat)
```

Coefficients:

	consumo2.d	desempleo2.d
ect1	0.002733	0.002404
constant	12.507717	34.467272
consumo2.dl1	-0.037815	-1.672301
desempleo2.dl1	-0.020429	0.158024

\$beta

	ect1
consumo2.l2	1.0000000
desempleo2.l2	-0.1984701