Maestría en Analítica e Inteligencia de Negocios

Clase: Series de tiempo y pronóstico

PhD. St. Orlando Joaqui-Barandica

Universidad del Valle

2021

Contenido

Contenido

Tendencia

Tendencia lineal:

Es común que los datos de series temporales tengan tendencia

$$x_t = t \tag{1}$$

- $t = 1, 2, ..., T$
- Fuerte suposición de que la tendencia continuará

Variables Dummy

Si una variable categórica toma solo dos valores (por ejemplo, "Sí" o "No"), entonces se puede construir una variable numérica equivalente tomando el valor 1 en caso afirmativo y 0 en caso negativo.

Esto se llama una variable ficticia o **variable dummy**.

	A	B
1	Yes	1
2	Yes	1
3	No	0
4	Yes	1
5	No	0
6	No	0
7	Yes	1
8	Yes	1
9	No	0
10	No	0
11	No	0
12	No	0
13	Yes	1
14	No	0
		21

Variables Dummy

Si hay más de dos categorías, la variable puede codificarse utilizando varias variables ficticias (una menos que el número total de categorías).

	A	B	C	D	E
1	Monday	1	0	0	0
2	Tuesday	0	1	0	0
3	Wednesday	0	0	1	0
4	Thursday	0	0	0	1
5	Friday	0	0	0	0
6	Monday	1	0	0	0
7	Tuesday	0	1	0	0
8	Wednesday	0	0	1	0
9	Thursday	0	0	0	1
10	Friday	0	0	0	0
11	Monday	1	0	0	0
12	Tuesday	0	1	0	0
13	Wednesday	0	0	1	0
14	Thursday	0	0	0	1
15	Friday	0	0	0	0

Cuidado con las variables Dummy

- ¡Usar un variable dummy para cada categoría da demasiadas variables dummy!
- La regresión será entonces singular e inestimable.
- Omita la constante u omita la dummy para una categoría.
- Los coeficientes de las dummies son relativos a la categoría omitida.

Uso de variables dummy

Dummies estacionales

- Para datos trimestrales: use 3 dummies
- Para datos mensuales: use 11 dummies
- Para datos diarios: use 6 dummies
- ¿Qué hacer con los datos semanales?

Outliers

- Si hay un valor atípico, puede usar una variable ficticia (tomando el valor 1 para esa observación y 0 en otro lugar) para eliminar su efecto.

Producción de cerveza

```
beer2 <- window(ausbeer, start=1992)
autoplot(beer2) + xlab("Year") + ylab("Megalitres")
```

Producción de cerveza

```
beer2 <- window(ausbeer, start=1992)
autoplot(beer2) + xlab("Year") + ylab("Megalitres")
```

Producción de cerveza

Queremos pronosticar el valor de la futura producción de cerveza. Podemos modelar estos datos usando un modelo de regresión con una tendencia lineal y variables ficticias trimestrales, donde 1 si está en el trimestre y 0 de lo contrario.

Modelo de regresión

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t$$

- $d_{i,t} = 1$ si t es trimestre i y 0 en otro caso

Producción de cerveza

Queremos pronosticar el valor de la futura producción de cerveza. Podemos modelar estos datos usando un modelo de regresión con una tendencia lineal y variables ficticias trimestrales, donde 1 si está en el trimestre y 0 de lo contrario.

Modelo de regresión

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t$$

- $d_{i,t} = 1$ si t es trimestre i y 0 en otro caso

Tenga en cuenta que *trend* y *season* no son objetos en el espacio de trabajo de R; se crean automáticamente tslm()cuando se especifican de la siguiente manera.

Producción de cerveza

```
fit.beer <- tslm(beer ~ trend + season)
summary(fit.beer)
```

Producción de cerveza

```
fit.beer <- tslm(beer ~ trend + season)
summary(fit.beer)
```

```
##
## Call:
## tslm(formula = beer ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.903  -7.599  -0.459   7.991  21.789
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 441.80044    3.73353 118.333  < 2e-16 ***
## trend        -0.34027    0.06657  -5.111 2.73e-06 ***
## season2     -34.65973    3.96832  -8.734 9.10e-13 ***
## season3     -17.82164    4.02249  -4.430 3.45e-05 ***
## season4      72.79641    4.02305  18.095  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9199
## F-statistic: 210.7 on 4 and 69 DF,  p-value: < 2.2e-16
```

Producción de cerveza

Interpretación

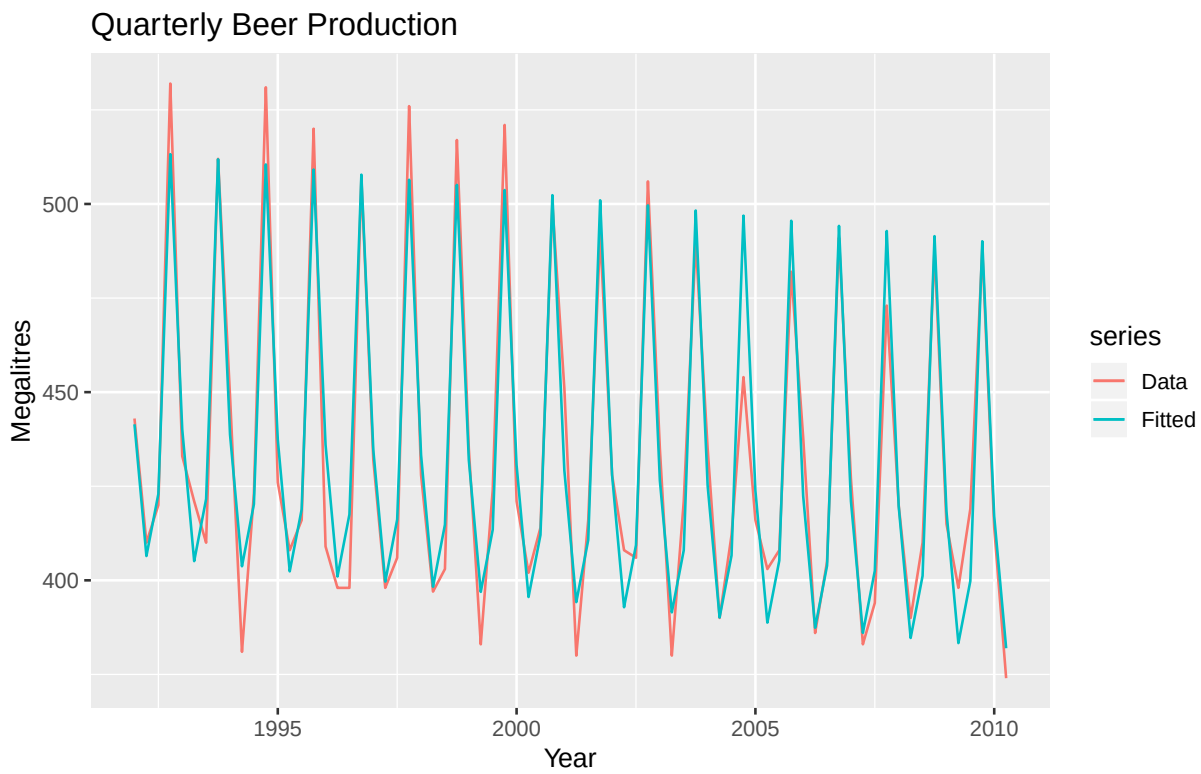
1. Hay una tendencia promedio a la baja de -0.34 megalitros por trimestre.
2. En promedio, el segundo trimestre tiene una producción de 34.7 megalitros menor que el primer trimestre
3. El tercer trimestre tiene una producción de 17.8 megalitros menor que el primer trimestre
4. El cuarto trimestre tiene una producción de 72.8 megalitros mayor que el primer trimestre.

Producción de cerveza

```
autoplot(beer2, series="Data") +
  autolayer(fitted(fit.beer), series="Fitted") +
  xlab("Year") + ylab("Megalitres") +
  ggtitle("Quarterly Beer Production")
```

Producción de cerveza

```
autoplot(beer2, series="Data") +
  autolayer(fitted(fit.beer), series="Fitted") +
  xlab("Year") + ylab("Megalitres") +
  ggtitle("Quarterly Beer Production")
```

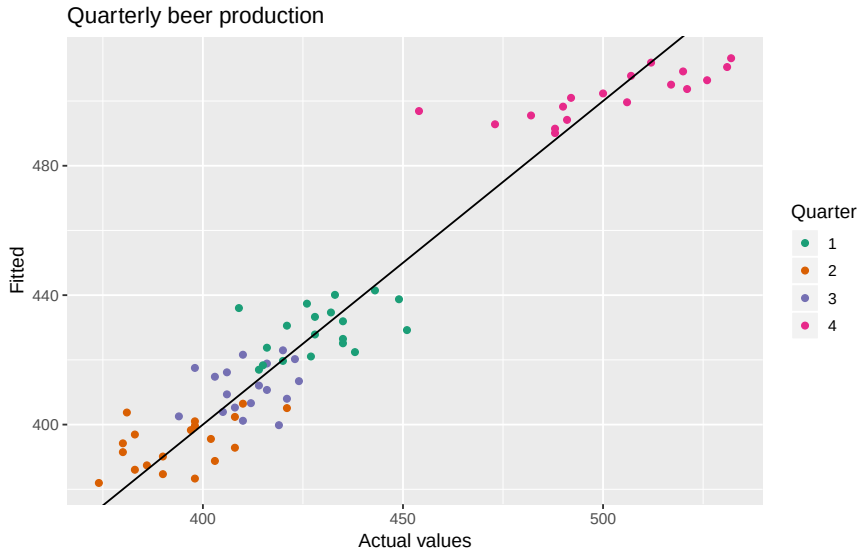


Producción de cerveza

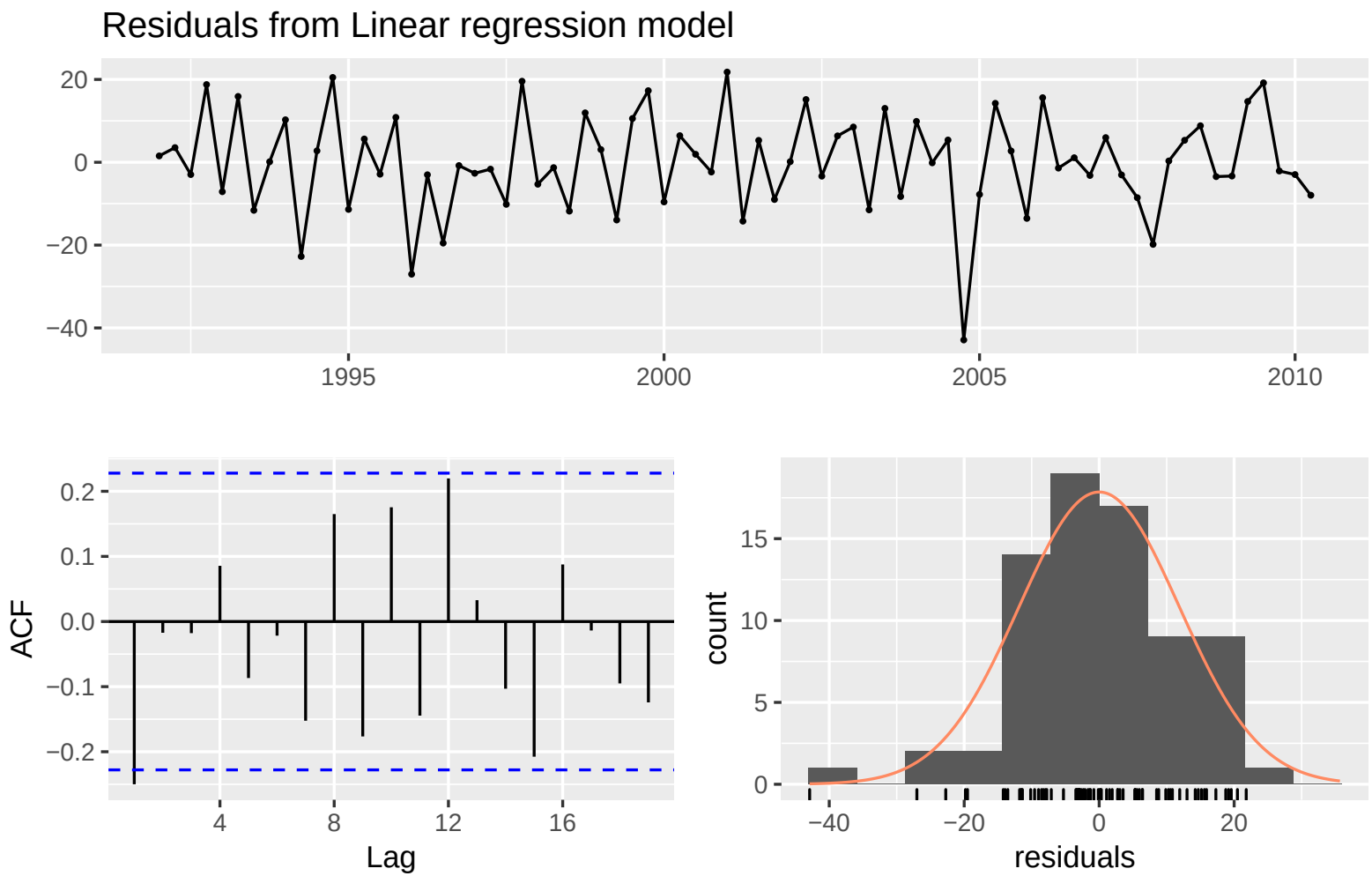
```
cbind(Data=beer2, Fitted=fitted(fit.beer)) %>%
  as.data.frame() %>%
  ggplot(aes(x = Data, y = Fitted,
             colour = as.factor(cycle(beer2)))) +
    geom_point() +
    ylab("Fitted") + xlab("Actual values") +
    ggtitle("Quarterly beer production") +
    scale_colour_brewer(palette="Dark2", name="Quarter") +
    geom_abline(intercept=0, slope=1)
```

Producción de cerveza

```
cbind(Data=beer2, Fitted=fitted(fit.beer)) %>%
  as.data.frame() %>%
  ggplot(aes(x = Data, y = Fitted,
             colour = as.factor(cycle(beer2)))) +
    geom_point() +
    ylab("Fitted") + xlab("Actual values") +
    ggtitle("Quarterly beer production") +
    scale_colour_brewer(palette="Dark2", name="Quarter") +
    geom_abline(intercept=0, slope=1)
```



Producción de cerveza



Variables de intervención

A menudo es necesario modelar intervenciones que pueden haber afectado la variable a pronosticar. Por ejemplo, la actividad de la competencia, el gasto publicitario, la acción industrial, etc., pueden tener un efecto.

Variables de intervención

A menudo es necesario modelar intervenciones que pueden haber afectado la variable a pronosticar. Por ejemplo, la actividad de la competencia, el gasto publicitario, la acción industrial, etc., pueden tener un efecto.

Picos

- Equivalente a una variable ficticia para manejar un valor atípico. Esta es una variable ficticia que toma el valor uno en el período de la intervención y cero en otro lugar.

Variables de intervención

A menudo es necesario modelar intervenciones que pueden haber afectado la variable a pronosticar. Por ejemplo, la actividad de la competencia, el gasto publicitario, la acción industrial, etc., pueden tener un efecto.

Picos

- Equivalente a una variable ficticia para manejar un valor atípico. Esta es una variable ficticia que toma el valor uno en el período de la intervención y cero en otro lugar.

Pasos

- Si una intervención causa un cambio de nivel. La variable toma el valor 0 antes de la intervención y 1 después.

Variables de intervención

A menudo es necesario modelar intervenciones que pueden haber afectado la variable a pronosticar. Por ejemplo, la actividad de la competencia, el gasto publicitario, la acción industrial, etc., pueden tener un efecto.

Picos

- Equivalente a una variable ficticia para manejar un valor atípico. Esta es una variable ficticia que toma el valor uno en el período de la intervención y cero en otro lugar.

Pasos

- Si una intervención causa un cambio de nivel. La variable toma el valor 0 antes de la intervención y 1 después.

Cambio de pendiente

- Aquí la intervención se maneja utilizando una tendencia lineal por partes. Las variables toman los valores 0 antes de la intervención y los valores $\{1, 2, 3, ...\}$ después

Contenido

Pronósticos con regresión

Recuerde que las predicciones de y se pueden obtener usando:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t}$$

El cual comprende los coeficientes estimados e ignora el error en la ecuación de regresión

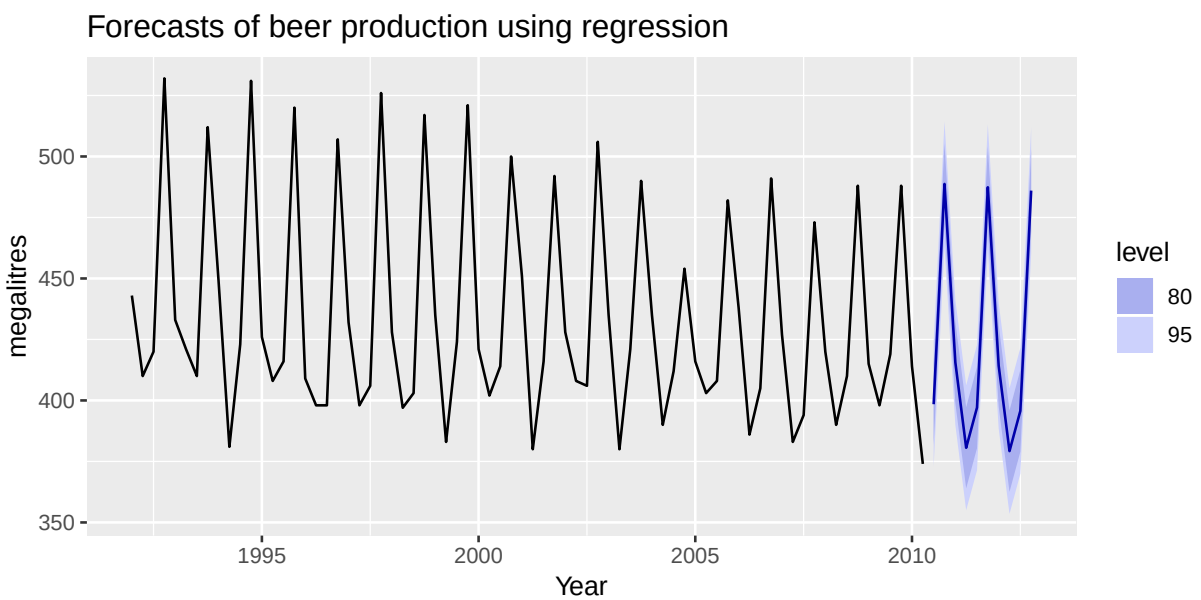
Nos interesa pronosticar valores futuros.

Pronósticos

```
beer2 <- window(ausbeer, start=1992)
fit.beer <- tslm(beer2 ~ trend + season)
fcast <- forecast(fit.beer)
autoplot(fcast) +
  ggtitle("Forecasts of beer production using regression") +
  xlab("Year") + ylab("megalitres")
```

Pronósticos

```
beer2 <- window(ausbeer, start=1992)
fit.beer <- tslm(beer2 ~ trend + season)
fcast <- forecast(fit.beer)
autoplot(fcast) +
  ggtitle("Forecasts of beer production using regression") +
  xlab("Year") + ylab("megalitres")
```



Pronósticos basados en escenarios

El pronosticador asume posibles escenarios para las variables predictoras que son de interés.

Por ejemplo, un creador de políticas de los Estados Unidos puede estar interesado en comparar el cambio pronosticado en el consumo cuando hay un crecimiento constante de 1 % y 0.5 % respectivamente para ingresos y ahorros sin cambios en la tasa de empleo,

versus

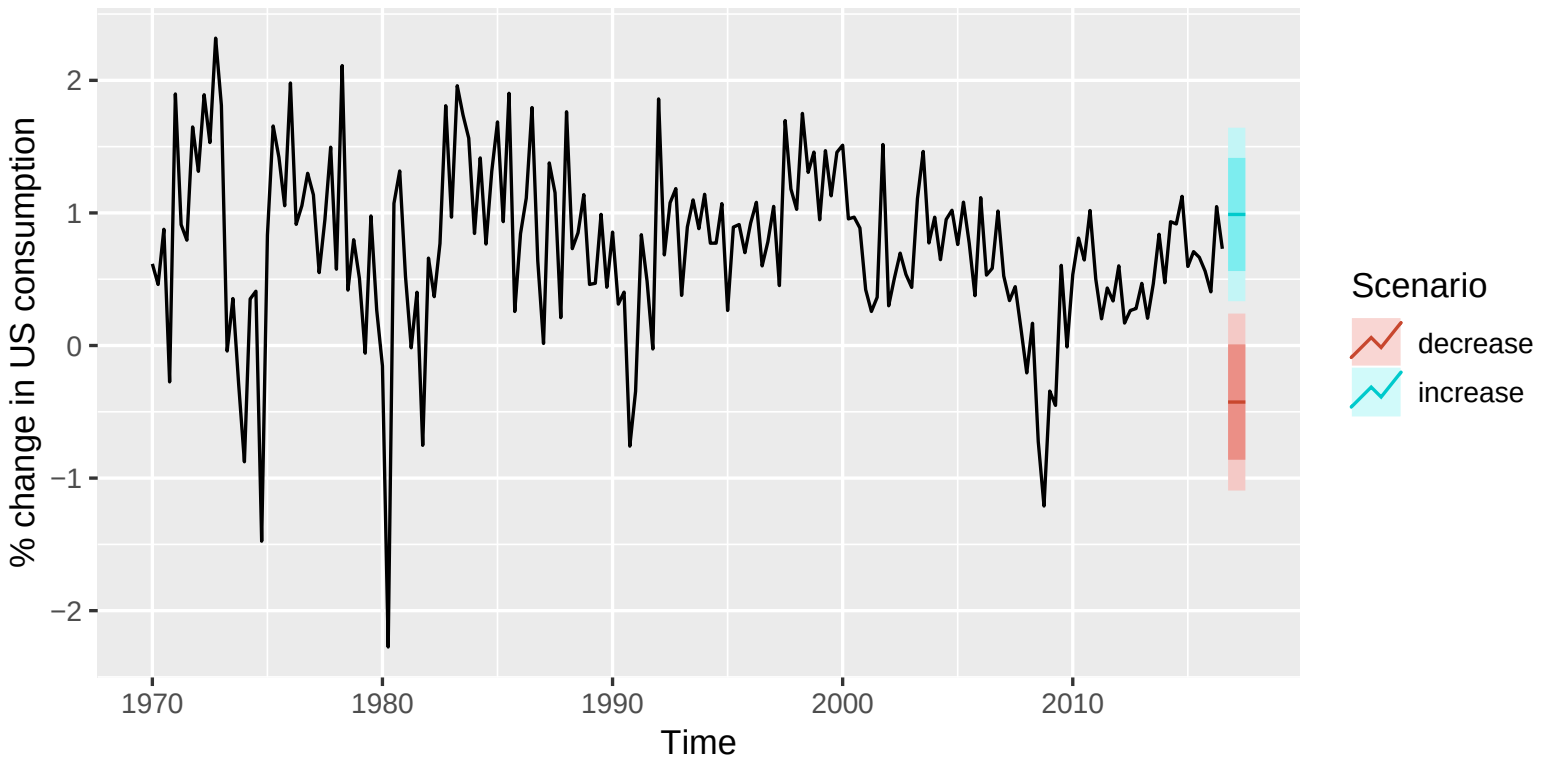
una disminución respectiva de 1 % y 0.5 %, para cada uno de los cuatro trimestres siguientes al final de la muestra.

Pronósticos basados en escenarios

```
fit.consBest <- tslm(
  Consumption ~ Income + Savings + Unemployment,
  data = uschange)
h <- 4
newdata <- data.frame(
    Income = c(1, 1, 1, 1),
    Savings = c(0.5, 0.5, 0.5, 0.5),
    Unemployment = c(0, 0, 0, 0))
fcast.up <- forecast(fit.consBest, newdata = newdata)
newdata <- data.frame(
    Income = rep(-1, h),
    Savings = rep(-0.5, h),
    Unemployment = rep(0, h))
fcast.down <- forecast(fit.consBest, newdata = newdata)
```

```
autoplot(uschange[, 1]) +
  ylab("% change in US consumption") +
  autolayer(fcast.up, PI = TRUE, series = "increase") +
  autolayer(fcast.down, PI = TRUE, series = "decrease") +
  guides(colour = guide_legend(title = "Scenario"))
```

Pronósticos basados en escenarios



Contenido

Regresión no lineal

Hay muchos casos en los que una forma funcional no lineal es más adecuada. Para simplificar las cosas, suponemos que solo tenemos un predictor x.

Regresión no lineal

Hay muchos casos en los que una forma funcional no lineal es más adecuada. Para simplificar las cosas, suponemos que solo tenemos un predictor x.

La forma más sencilla de modelar una relación no lineal es transformar la variable de pronóstico y y/o la variable predictora x antes de estimar un modelo de regresión. Si bien esto proporciona una forma funcional no lineal, el modelo sigue siendo lineal en los parámetros.

Regresión no lineal

Hay muchos casos en los que una forma funcional no lineal es más adecuada. Para simplificar las cosas, suponemos que solo tenemos un predictor x.

La forma más sencilla de modelar una relación no lineal es transformar la variable de pronóstico y y/o la variable predictora x antes de estimar un modelo de regresión. Si bien esto proporciona una forma funcional no lineal, el modelo sigue siendo lineal en los parámetros.

Una forma funcional específica: **log - log**

$$log(y) = \beta_0 + \beta_1 log(x) + \varepsilon \tag{2}$$

- La pendiente β_1 puede interpretarse como una elasticidad: β_1 es el cambio porcentual promedio en y resultante de un 1 % aumento en x.

Regresión no lineal

Hay casos en los que la simple transformación de los datos no será adecuada y puede ser necesaria una especificación más general. Entonces el modelo que usamos es:

$$y = f(x) + \varepsilon \tag{3}$$

dónde f es una función no lineal.

Regresión no lineal

Hay casos en los que la simple transformación de los datos no será adecuada y puede ser necesaria una especificación más general. Entonces el modelo que usamos es:

$$y = f(x) + \varepsilon \tag{3}$$

dónde f es una función no lineal.

Una de las especificaciones más simples es hacer f lineal por partes. Es decir, introducimos puntos donde la pendiente de f puede cambiar. Estos puntos se llaman nudos . Esto se puede lograr dejando $x_{1,t} = t$ e introduciendo $x_{2,t}$ tal que:

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geqslant \tau \end{cases} \tag{4}$$

Regresión no lineal

Las relaciones lineales por partes construidas de esta manera son un caso especial de splines de regresión . En general, se obtiene una **spline de regresión lineal** usando:

$$x_1 = t \qquad x_2 = (t - \tau_1) \quad \ldots \quad x_k = (t - \tau_{k-1}) \tag{5}$$

dónde $\tau_1, \tau_2, \ldots, \tau_{k-1}$ son los nudos (los puntos en los que la línea puede doblarse).

Regresión no lineal

Las relaciones lineales por partes construidas de esta manera son un caso especial de splines de regresión . En general, se obtiene una **spline de regresión lineal** usando:

$$x_1 = t \qquad x_2 = (t - \tau_1) \quad \ldots \quad x_k = (t - \tau_{k-1}) \tag{5}$$

dónde $\tau_1, \tau_2, \ldots, \tau_{k-1}$ son los nudos (los puntos en los que la línea puede doblarse).

La selección del número de nudos $(k - 1)$ y dónde deben colocarse puede ser difícil y algo arbitrario. Algunos algoritmos de selección automática de nudos están disponibles en algunos programas, pero aún no se usan ampliamente.

Regresión no lineal

Se puede obtener un resultado más uniforme utilizando cubos por partes en lugar de líneas por partes. Estos están limitados a ser continuos (se unen) y suaves (para que no haya cambios repentinos de dirección, como vemos con splines lineales por partes).

En general, una **spline de regresión cúbica** se escribe como:

$$x_1 = t \qquad x_2 = t^2 \qquad x_3 = t^3 \qquad x_4 = (t - \tau) \quad \ldots \quad x_k = (t - \tau_{k-1}) \quad (6)$$

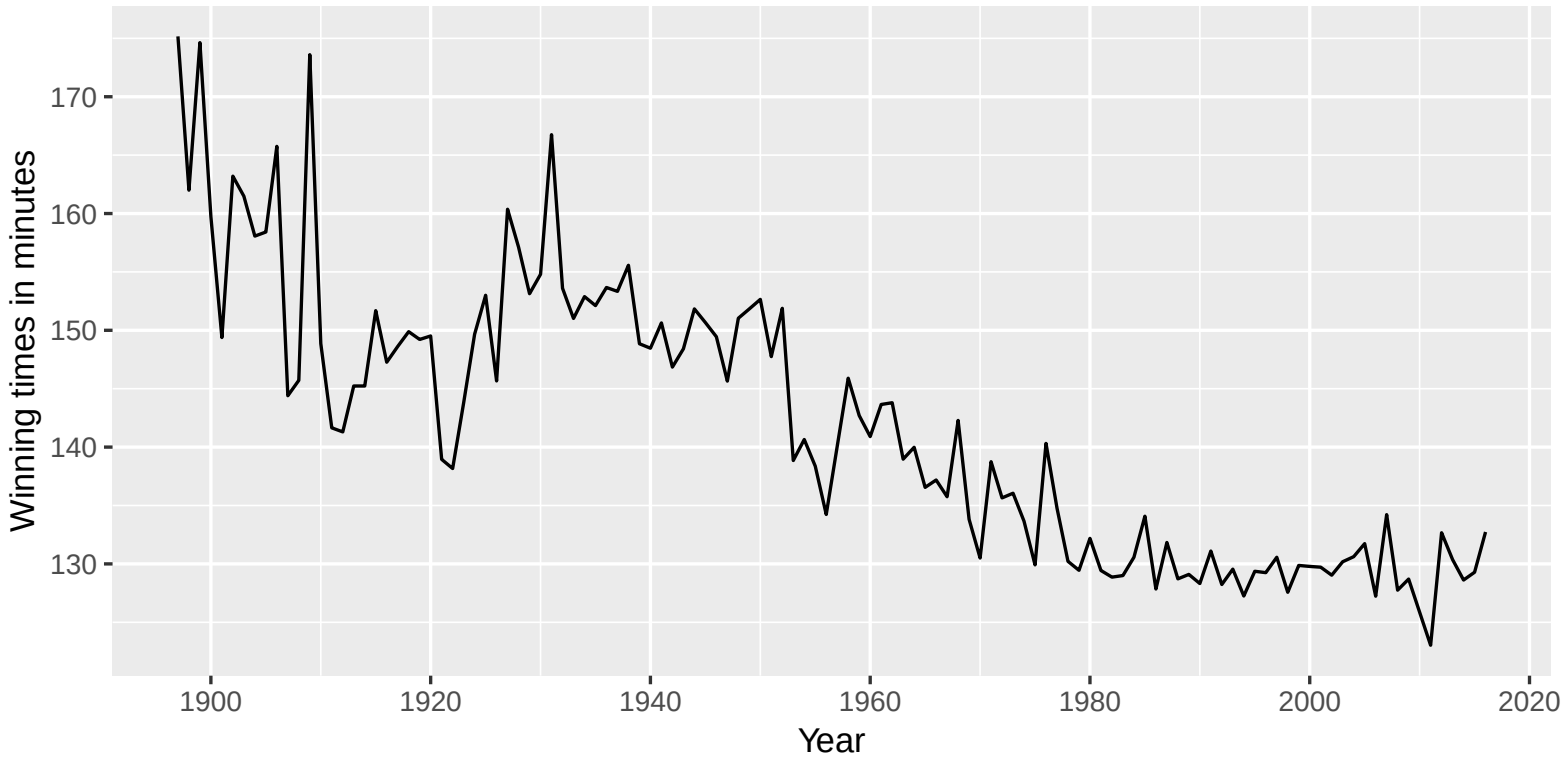
Las splines cúbicas generalmente se ajustan mejor a los datos. Sin embargo, los pronósticos de y pueden llegar a volverse poco confiables cuando x está fuera del rango de los datos históricos.

Ejemplo: Tiempos maratón de Boston

```
autoplot(marathon) +
xlab("Year") + ylab("Winning times in minutes")
```

Ejemplo: Tiempos maratón de Boston

```
autoplot(marathon) +
xlab("Year") + ylab("Winning times in minutes")
```

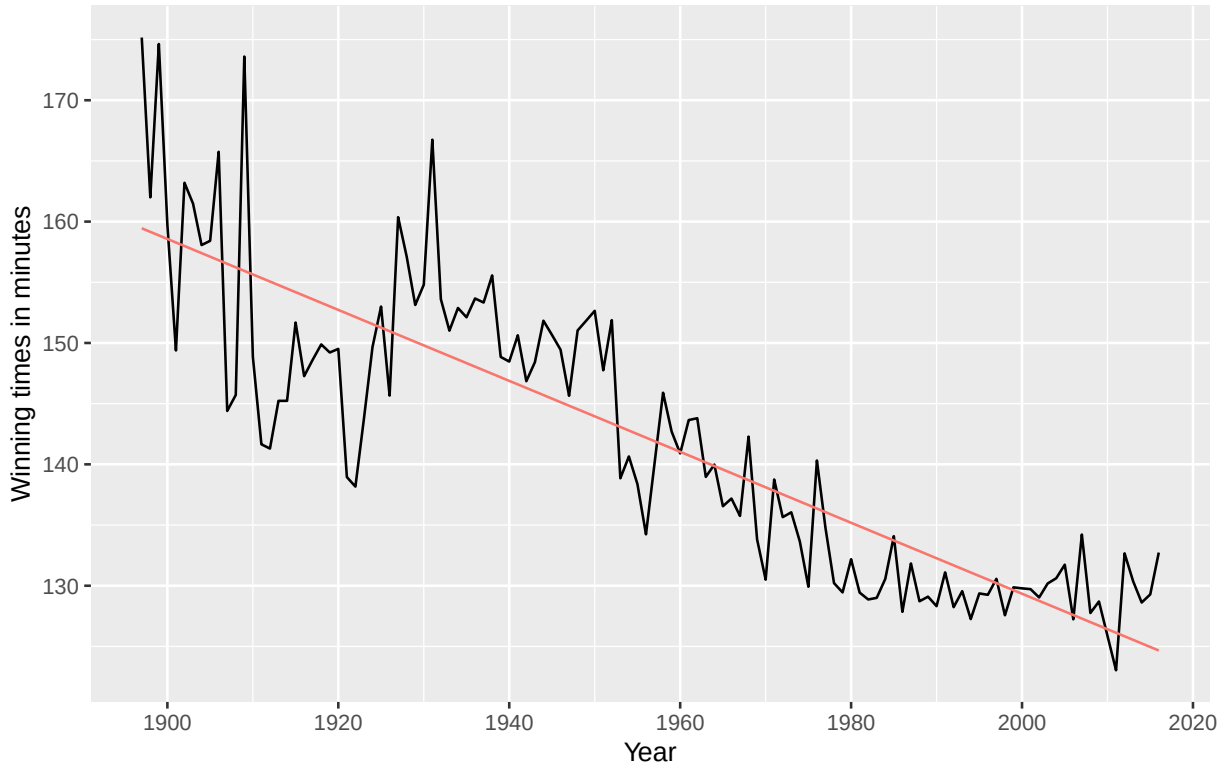


Ejemplo: Tiempos maratón de Boston

```
fit.lin <- tslm(marathon ~ trend)
autoplot(marathon) +
  autolayer(fitted(fit.lin), series = "Linear")+
  xlab("Year") + ylab("Winning times in minutes")
```

Ejemplo: Tiempos maratón de Boston

```
fit.lin <- tslm(marathon ~ trend)
autoplot(marathon) +
  autolayer(fitted(fit.lin), series = "Linear")+
  xlab("Year") + ylab("Winning times in minutes")
```

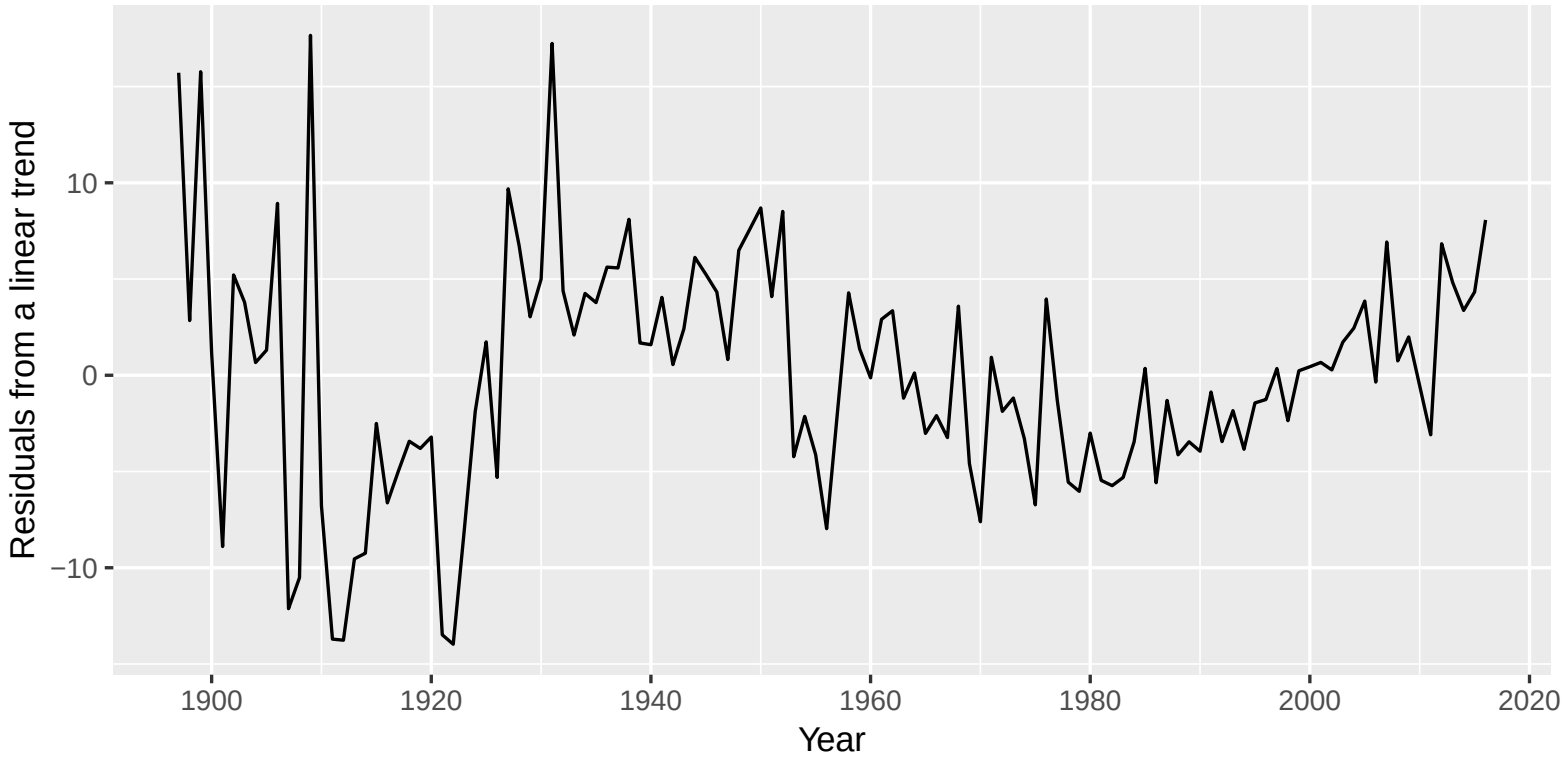


Ejemplo: Tiempos maratón de Boston

```
autoplot(residuals(fit.lin)) +
xlab("Year") + ylab("Residuals from a linear trend")
```

Ejemplo: Tiempos maratón de Boston

```
autoplot(residuals(fit.lin)) +
xlab("Year") + ylab("Residuals from a linear trend")
```



Ejemplo: Tiempos maratón de Boston

Interpretación:

Datos: los tiempos ganadores del maratón de Boston (en minutos) desde que comenzó en 1897.

- La serie temporal muestra una tendencia general a la baja a medida que los tiempos ganadores han ido mejorando a lo largo de los años.
- El gráfico de residuos muestra los residuos del ajuste de una tendencia lineal a los datos.
- El gráfico muestra un patrón no lineal obvio que no ha sido capturado por la tendencia lineal.
- También hay algo de heterocedasticidad, con una variación decreciente con el tiempo.

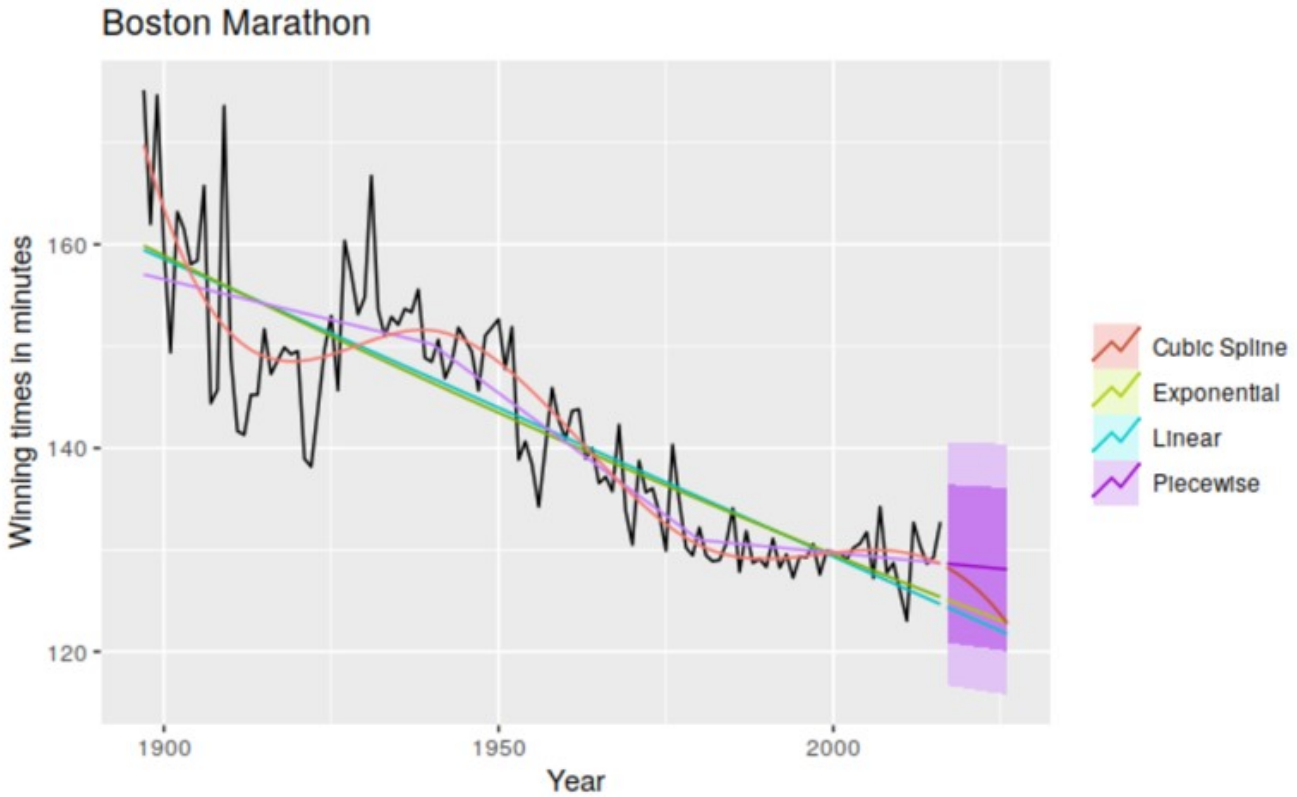
Ejemplo: Tiempos maratón de Boston

```
h <- 10
fit.lin <- tslm(marathon ~ trend)
fcasts.lin <- forecast(fit.lin, h = h)
fit.exp <- tslm(marathon ~ trend, lambda = 0) #Tend. Exp.
fcasts.exp <- forecast(fit.exp, h = h)

t <- time(marathon)
t.break1 <- 1940
t.break2 <- 1980
tb1 <- ts(pmax(0, t - t.break1), start = 1897)
tb2 <- ts(pmax(0, t - t.break2), start = 1897)

fit.pw <- tslm(marathon ~ t + tb1 + tb2)
t.new <- t[length(t)] + seq(h)
tb1.new <- tb1[length(tb1)] + seq(h)
tb2.new <- tb2[length(tb2)] + seq(h)

newdata <- cbind(t=t.new, tb1=tb1.new, tb2=tb2.new) %>%
  as.data.frame()
fcasts.pw <- forecast(fit.pw, newdata = newdata)

fit.spline <- tslm(marathon ~ t + I(t^2) + I(t^3) +
  I(tb1^3) + I(tb2^3))
fcasts.spl <- forecast(fit.spline, newdata = newdata)
```

Ejemplo: Tiempos maratón de Boston

```
autoplot(marathon) +
  autolayer(fitted(fit.lin), series = "Linear") +
  autolayer(fitted(fit.exp), series = "Exponential") +
  autolayer(fitted(fit.pw), series = "Piecewise") +
  autolayer(fitted(fit.spline), series = "Cubic Spline") +
  autolayer(fcasts.pw, series="Piecewise") +
  autolayer(fcasts.lin, series="Linear", PI=FALSE) +
  autolayer(fcasts.exp, series="Exponential", PI=FALSE) +
  autolayer(fcasts.spl, series="Cubic Spline", PI=FALSE) +
  xlab("Year") + ylab("Winning times in minutes") +
  ggtitle("Boston Marathon") +
  guides(colour = guide_legend(title = " "))
```

Ejemplo: Tiempos maratón de Boston



Los mejores pronósticos parecen provenir de la tendencia lineal por partes, mientras que la spline cúbica brinda el mejor ajuste a los datos históricos, pero los pronósticos son deficientes.

Ejemplo: Tiempos maratón de Boston

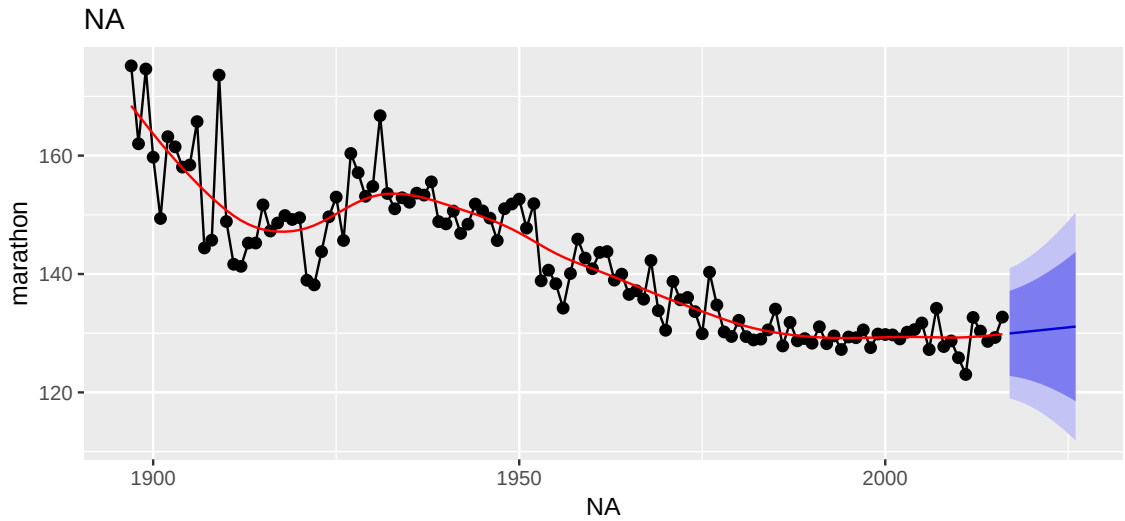
Existe una formulación alternativa de splines cúbicas **(llamadas splines de suavizado cúbico natural)** que impone algunas restricciones, por lo que la función de spline es lineal al final, lo que generalmente ofrece pronósticos mucho mejores sin comprometer el ajuste.

```
marathon %>%
  splinef(lambda=0) %>%
  autoplot()

#transformación logaritmica (lambda=0) para manejar la heterocedasticidad.
```

Ejemplo: Tiempos maratón de Boston

Existe una formulación alternativa de splines cúbicas **(llamadas splines de suavizado cúbico natural)** que impone algunas restricciones, por lo que la función de spline es lineal al final, lo que generalmente ofrece pronósticos mucho mejores sin comprometer el ajuste.

```
marathon %>%
  splinef(lambda=0) %>%
  autoplot()

#transformación logaritmica (lambda=0) para manejar la heterocedasticidad.
```



Ejemplo: Tiempos maratón de Boston

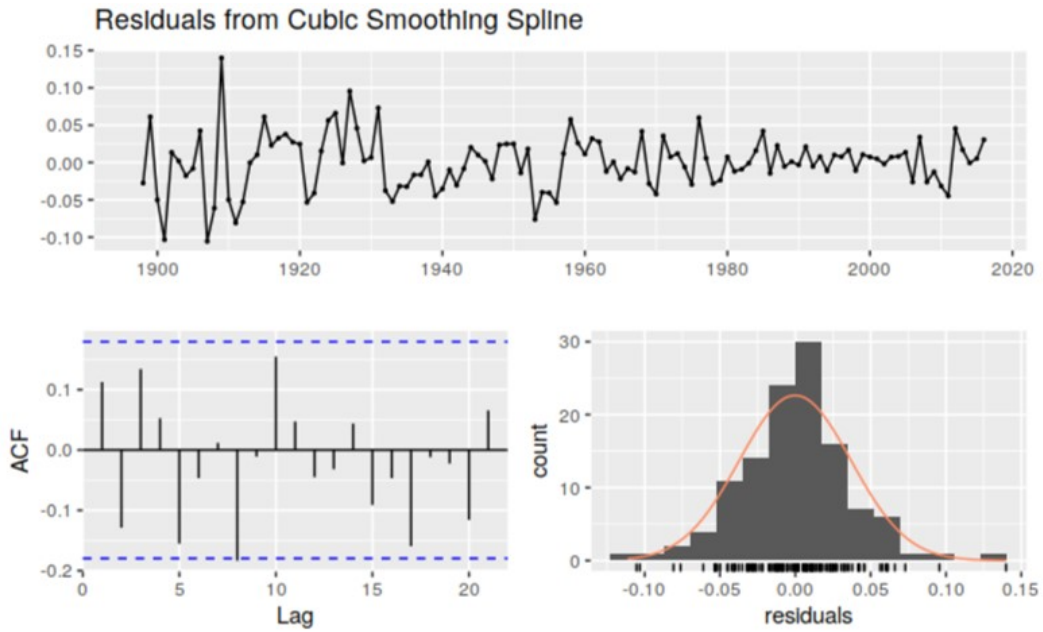
Los residuos muestran que este modelo ha capturado bien la tendencia, aunque queda algo de heterocedasticidad. El amplio intervalo de predicción en los pronósticos refleja la volatilidad observada en los tiempos ganadores históricos.

```
marathon %>%
  splinef(lambda=0) %>%
  checkresiduals()
```

Ejemplo: Tiempos maratón de Boston

Los residuos muestran que este modelo ha capturado bien la tendencia, aunque queda algo de heterocedasticidad. El amplio intervalo de predicción en los pronósticos refleja la volatilidad observada en los tiempos ganadores históricos.

```
marathon %>%
  splinef(lambda=0) %>%
  checkresiduals()
```



Preguntas?

Gracias!! ,

Jr.

orlando.joaqui@correounivalle.edu.co