Ajuste de curvas

Fabiola Vázquez

20 de octubre de 2020

---

1. Introducción

Este estudio se realiza con el software R versión 4.0.2 [5], en un cuaderno de Jupyter [3]. Los conjuntos de datos con los que se trabaja fueron, en mayor parte, generados con alguna función en particular. Como un ejemplo real, se consideran los datos obtenidos del INEGI [1], de la población de México en ciertos años.

2. Análisis

El objetivo es transformar ciertos conjuntos de datos a una realación lineal. Para ello se hace uso de la transformada de Tukey [4], como se muestra en la ecuación 1. En base a esto, se implementa la función `lambda`, la cual está definida en el código 1. Esta función toma dos conjuntos `x` y `y`, y retorna el valor de λ que maximiza la correlación entre ellos. En dicha función, el conjunto de datos transformados es `y`.

$$\tilde{x}_\lambda = \begin{cases} x^\lambda, & \text{si } \lambda > 0, \\ \log x, & \text{si } \lambda = 0, \\ -x^\lambda, & \text{si } \lambda < 0. \end{cases} \tag{1}$$

```
lambda <- function (x,y){
      if (min(y) < 0){
         y<-y+abs(min(y))+0.01
     }

     cc <- numeric()
     lambda <- seq(-10, 10, .01)

     for (i in lambda){
         if (i == 0)
             cc <- c(cc, cor(x, log(y)))
         else if (i > 0)
             cc <- c(cc, cor(x, y**i))
         else
             cc <- c(cc, cor(x, -(y**i)))
     }
```

```
    return (lambda[which.max(cc)])
}
```

Código 1: Función `lambda`.

Se automatiza el proceso de ajustar una curva a los conjuntos de datos con los que se trabaja, haciendo uso de la función `ajuste_curva`, descrita en el código 2.

```
ajuste_curva <- function(x,y){

    if (min(y) < 0){
        y<-y+abs(min(y))+0.01
    }

    lam <- lambda(x,y)
    if (abs(lam) < 0.1){
        fit <- lm(log(y)~x)
        y2 <- x*fit$coefficients[2]+fit$coefficients[1]
        plot(x,y,col=rgb(0.4,0.4,0.8,0.6),pch=16 , cex=1.3, las=1,font.lab=1, cex.
    lab=1.7, xlab="", ylab="")
        lines(x, exp(y2), col =rgb(0,0,0,0.9), pch=16, lwd = 2)
    }
    else  if(lam >= 0.1){
        fit <- lm (y**lam~x)
        y2 <- x*fit$coefficients[2]+fit$coefficients[1]
        plot(x,y,col=rgb(0.4,0.4,0.8,0.6),pch=16 , cex=1.3, las=1,font.lab=1, cex.
    lab=1.7,xlab="", ylab="")
        points(x,(y2^(1/lam)), col=rgb(0,0,0,0.5), pch=16, lwd = 2)
    }
        else{
        fit <- lm (y**lam~x)
        y2  <- -x*fit$coefficients[2]-fit$coefficients[1]
        plot(y~x,col=rgb(0.4,0.4,0.8,0.6),pch=16 , cex=1.3, las=1,font.lab=1, cex.
    lab=1.7,xlab="", ylab="")
        points(x, y2^(1/lam), col=rgb(0,0,0,0.5), pch=16, lwd=2)
        }
}
```

Código 2: Función `ajuste_curva`.

La figura 1 contiene cuatro ejemplos de funciones a las cuales se les realiza un ajuste de curva con la función mostrada en el código 2.

La primera función considerada es $y_1 = 4x^2 + r_1$, donde $r_1 \sim N(300, 30000)$ es un ruido añadido para ver que tan preciso es el proceso de ajustarle una curva. Haciendo uso de la función `ajuste_curva` se obtiene la figura 1a. Considerando la función $y_2 = 4\sqrt{x} + 10 + r_2$, donde $r_2 \sim N(1, 3)$, se obtiene la figura 1b. Si se supone que la relación es del tipo $y_3 = 0.07 * \exp(x) + r_3$, donde $r_3 \sim N(0, 0.2)$, se obtiene la figura 1c.

Tabla 1: Fragmento de datos generados.

(a)

x	$y = 4x^2 + r_1$	$y = 4\sqrt{x} + 10 + r_2$
355	193132.87	88.00
134	301924.31	67.48
356	116678.83	53.53
253	1034167.41	37.23
470	155104.76	31.39
123	515293.85	14.45

(b)

x	$y = 0.07\exp(x) + r_3$
0.00	-0.04
0.05	0.11
0.10	-0.00
0.15	0.15
0.20	-0.06
0.25	0.24

Tabla 2: Población total de México por año.

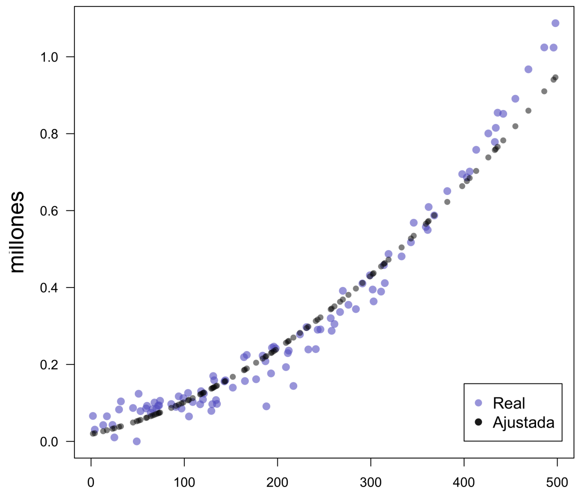
Año	Población
1910	15,160,369
1921	14,334,780
1930	16,552,722
1940	19,653,552
1950	25,791,017
1960	34,923,129
1970	48,225,238
1980	66,846,833
1990	81,249,645
1995	91,158,290
2000	97,483,412
2005	103,263,388
2010	112,336,538
2015	119,938,473

2.1. Población mexicana

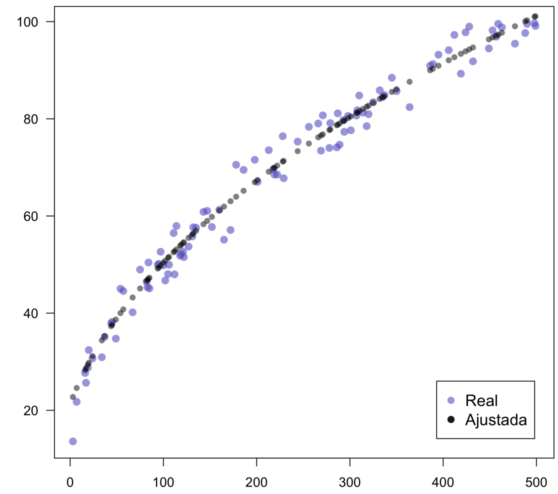
Se considera la población por año en México, cuyos datos se obtuvieron del INEGI [1]. Estos se muestran en el cuadro 2. Usando el mismo procedimiento que en la sección precedente, se ajusta una curva a la curva de crecimiento poblacional de México. El resultado de este ajuste se observa en la figura 1d.

2.2. Aproximación de polinomios

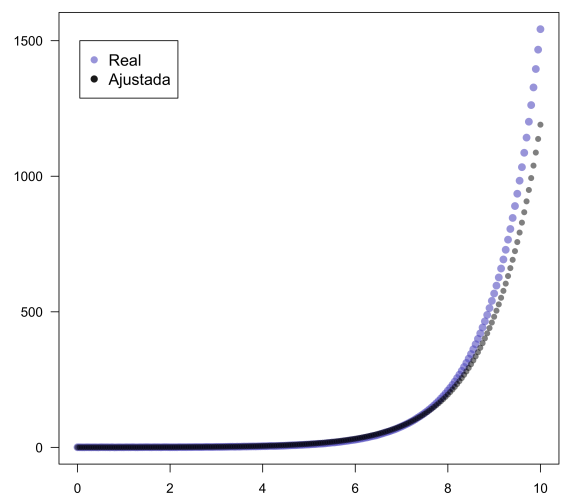
Se consideran los polinomios $y_1 = 0.1x^3 - 0.5x^2 - x + 10 + r_3$ y $y_2 = 0.1x^5 + 0.5x^3 - 0.4x^2 - 0.3x + 20 + r_4$. Un fragmento de los datos generados con estas funciones se muestra en el cuadro 3. Para ajustar curvas a este tipo de funciones [2] se trabaja con la función `lm(y ∼ w)`, donde `w` es la suma de las diferentes potencias de `x` con las que cuenta el polinomio a ajustar. Por ejemplo, en el primer caso, la función utilizada es `lm(y ∼ x + I(x∧2) + I(x∧3))` ya que es un polinomio de grado tres. La figura 2a muestra, en línea roja, la curva ajustada al polinomio y_1 y la figura 2b, la curva ajustada al polinomio y_2.



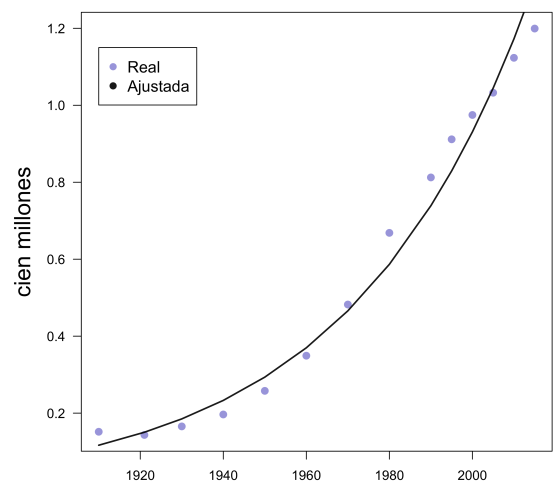
(a) Ajuste de una curva cuadrática.



(b) Ajuste de una curva con radicales.



(c) Ajuste de una curva exponencial.



(d) Ajuste de datos obtenidos del INEGI.

Figura 1: Ajuste de curvas con la función `ajuste_curva`.

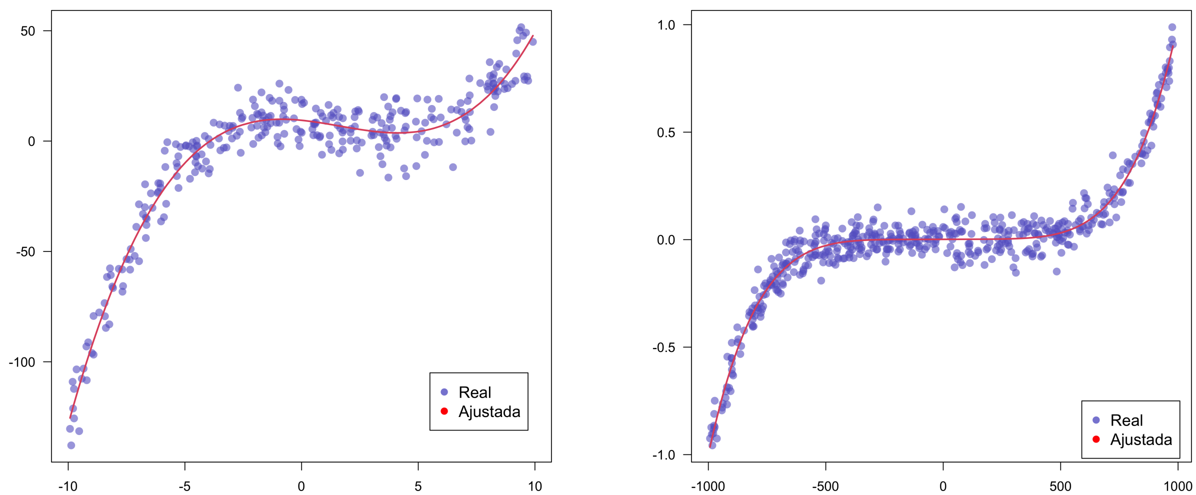
Tabla 3: Fragmento de datos generados por polinomios.

(a)

x	$y = 0.1x^3 - 0.5x^2 - x + 10 + r_3$
-8.05	-64.49
-5.07	-12.48
-5.24	-8.56
-8.28	-74.01
4.26	5.58
-8.25	-55.56

(b)

x	$y = 0.1x^5 + 0.5x^3 - 0.4x^2 - 0.3x + 20 + r_4$
-413.88	3,820,341,696,610.39
-673.99	-15,187,467,243,469.90
817.58	42,608,724,664,634.36
949.80	70,229,446,578,344.88
445.92	1,198,984,909,305.33
118.82	5,554,388,743,064.81



(a) Ajuste de una curva polinomial de grado dos.

(b) Ajuste de una curva polinomial de grado cinco.

Figura 2: Ajuste de curvas con la función `ajuste_curva`.

2.3. Multivariada

Se considera una función de tres variables independientes $y = x_1 + .4\log(x_2) + 4x_3$. Haciendo uso de la función `lm`, se obtienen los siguientes resultados.

```
Call:
lm(formula = y ~ x1 + log(x2) + x3)

Coefficients:
(Intercept)          x1        log(x2)          x3
  8.402e-13    1.000e+00    4.000e-01    4.000e+00
```

Referencias

[1] Instituto Nacional de Estadística y Geografía. Población total. https://www.inegi.org.mx/temas/estructura/.

[2] R graph gallery. Scatterplot with polynomial curve fitting. https://www.r-graph-gallery.com/44-polynomial-curve-fitting.html#:~:text=First%20of%20all%2C%20a%20scatterplot,plot%20it%20with%20line()%20.

[3] Thomas Kluyver, Benjamin Ragan-Kelley, Pérez, et al. Jupyter notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing*, page 87. IOS Press, 2016.

[4] David M. Lane. *Introduction to Statistics*. Online edition. http://onlinestatbook.com/Online_Statistics_Education.pdf.

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.