uc3m | Universidad Carlos III de Madrid

Curso OpenCourseWare

Aprendizaje del Software Estadístico R: un entorno para simulación y computación estadística

Alberto Muñoz García

---

13. Análisis de regresión. Casos univariante y multivariante

Introducción

El análisis de regresión se utiliza para explicar una determinada variable, digamos Y, en función de una variable X, o bien en función de varias variables X1, X2, ..., Xk. En el primer caso se tratará de regresión univariante, y en el segundo caso, de regresión multivariante. El modelo de explicación en ambos casos es lineal, esto es, se asume que la dependencia entre Y y las variable explicativa X adopta la forma:

Y = a + b X + error

O, en el caso multivariante:

Y = a + b1 X1 + b2 X2 + ... + bk Xk.+ error

El término de error aparece porque cada vez que observamos una X, no siempre observaremos la misma Y. Por ejemplo, si X es la estatura de una persona, e Y el peso, cada vez que observemos una estatura, no siempre obtendremos el mismo peso en Y.

Los que hayáis estudiado estadística, conoceréis el modelo perfectamente. Los que no, simplemente debéis saber que el modelo es útil para predecir relaciones lineales, y para un estudio más profundo, cualquier libro de estadística con un capítulo de regresión sirve. En caso de querer leer un buen libro on-line con muchos ejemplos programados en R, podéis acudir a:

http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf(Proyecto R, Julian Faraway, Bath University, Reino Unido).

Regresión lineal simple

A continuación vamos a poner un ejemplo de regresión simple, realizado en R. El fichero sueldos.txt contiene la antigüedad (en años) y el sueldo (en millones de pesetas) de una serie de directivos de una empresa. Se trata de determinar si existe relación entre el sueldo de un directivo de dicha empresa y su antigüedad en la misma, y si es así, de estimarla.

En primer lugar, cargamos los datos y ponemos nombres a las variables:

> sueldos = scan("c:\\sueldos.txt") # variara segun donde guardeis

los datos

Read 40 items

> sueldos = structure(sueldos,dim=c(2,20),byrow=T)

> sueldos = t(sueldos)

> colnames(sueldos) = c("antiguedad","sueldo")

> sueldos

antiguedad sueldo

[1,] 0 4.61

[2,] 2 6.97

[3,] 2 6.36

[4,] 2 6.61

[5,] 1 3.61

[6,] 5 10.15

[7,] 0 4.00

[8,] 3 8.63

[9,] 3 9.34

[10,] 0 3.86

[11,] 5 12.62

[12,] 4 9.42

[13,] 3 7.63

[14,] 4 9.97

[15,] 2 6.33

[16,] 0 3.19

[17,] 1 5.62

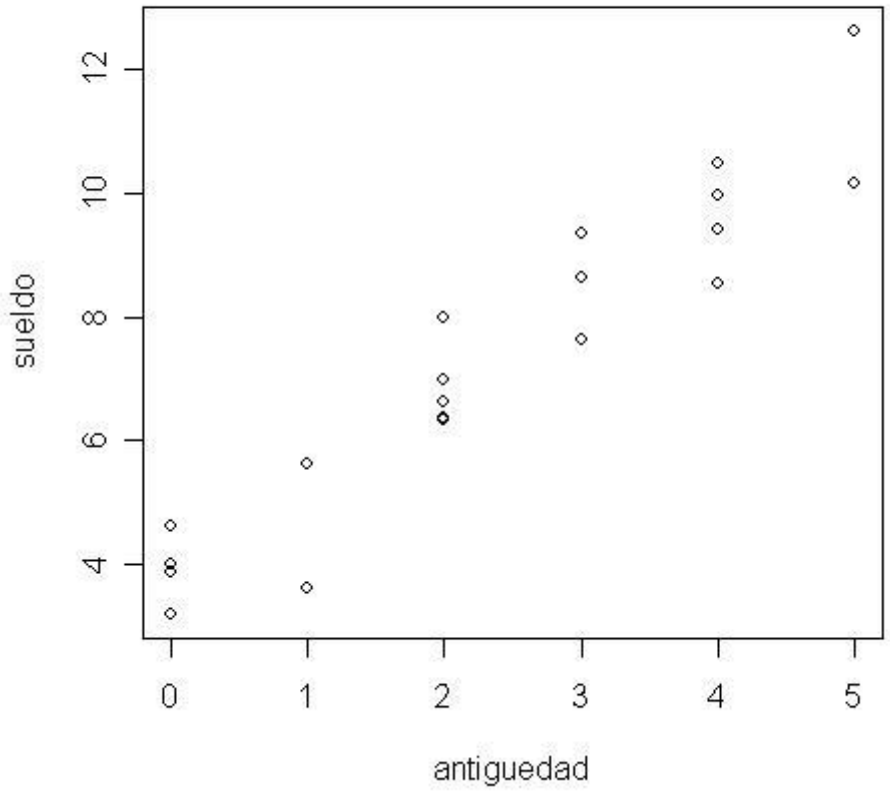
[18,] 2 7.98

[19,] 4 10.49

[20,] 4 8.54

El primer paso será observar el aspecto de los datos, para ver si es correcto intentar ajustar una recta a los mismos:

> plot(sueldos)



Una vez hecho eso, para ajustar la recta, se usa el comando lm. lm significa "linear model", y su uso es muy sencillo:

Convertimos sueldos a data frame para usar con comodidad los nombres de las variables:

> sueldos = data.frame(sueldos) # Convierte la matriz sueldos a data frame

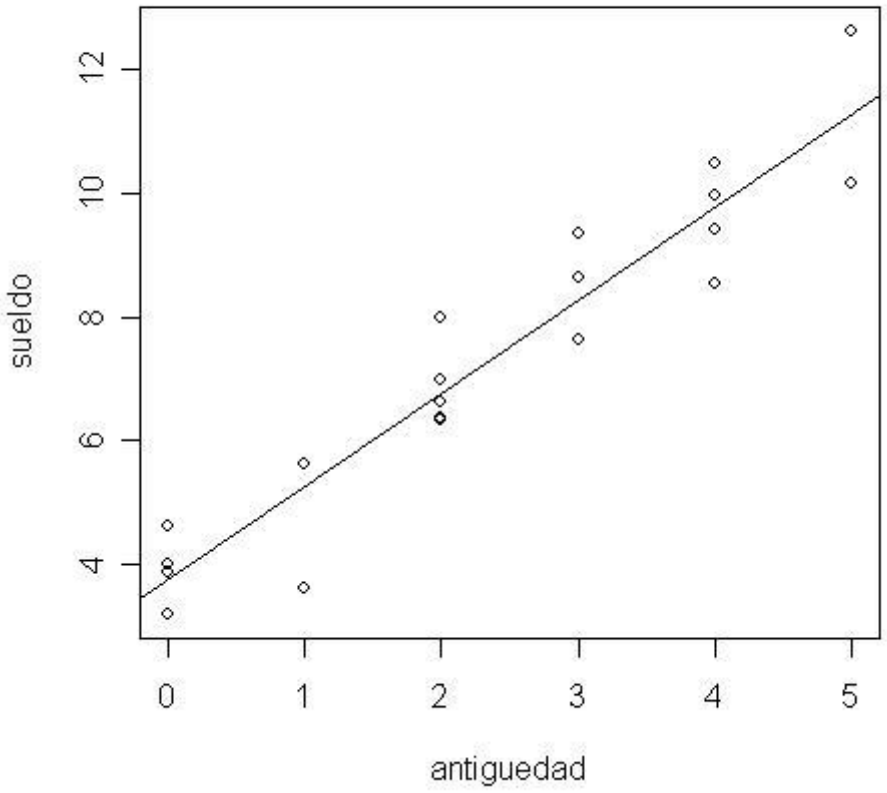
> attach(sueldos) # Permite que las variables de

"sueldos" se puedan llamar directamente

> sueldos.lm = lm(sueldo~antiguedad) # Realiza la regresión

Si queremos dibujar la recta de regresión haremos:

> abline(sueldos.lm)



Si queremos ver qué se guarda exactamente en el objeto de regresión sueldos.lm :

> attributes(sueldos.lm)

$names

[1] "coefficients" "residuals" "effects" "rank"

[5] "fitted.values" "assign" "qr" "df.residual"

[9] "xlevels" "call" "terms" "model"

$class

[1] "lm"

Los que hayan estudiado regresión con anterioridad reconocerán varios de los nombres de la estructura anterior. Para explicar los más importantes, procederemos a visualizar un sumario del proceso de regresión:

> summary(sueldos.lm)

Call:

lm(formula = sueldo ~ antiguedad)

Residuals:

Min 1Q Median 3Q Max

-1.6538 -0.4717 0.1455 0.4444 1.3334

Coefficients:

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

(Intercept) 3.7581 0.3324 11.31 1.31e-09 *

antiguedad 1.5057 0.1164 12.93 1.50e-10 *

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

Residual standard error: 0.8441 on 18 degrees of freedom

Multiple R-Squared: 0.9028, Adjusted R-squared: 0.8974

F-statistic: 167.2 on 1 and 18 degrees of freedom, p-value: 1.502e-010

Lo primero es localizar la ecuación de la recta de regresión. Esta viene dada por los coeficientes, que en este caso son 3.7581 y 1.5057 , lo que quiere decir que la recta de regresión viene dada por :

Sueldo = 1.5057 + 3.7581 * Antiguedad

Naturalmente para cada dato concreto se comete un error. Dichos errores son los residuos. Si los queremos ver para los datos de nuestro problema, escribiremos:

> sueldos.lm$residuals

1 2 3 4 5 6 7

0.8518934 0.2004948 -0.4095052 -0.1595052 -1.6538059 -1.1366032 0.2418934

8 9 10 11 12 13 14

0.3547954 1.0647954 0.1018934 1.3333968 -0.3609039 -0.6452046 0.1890961

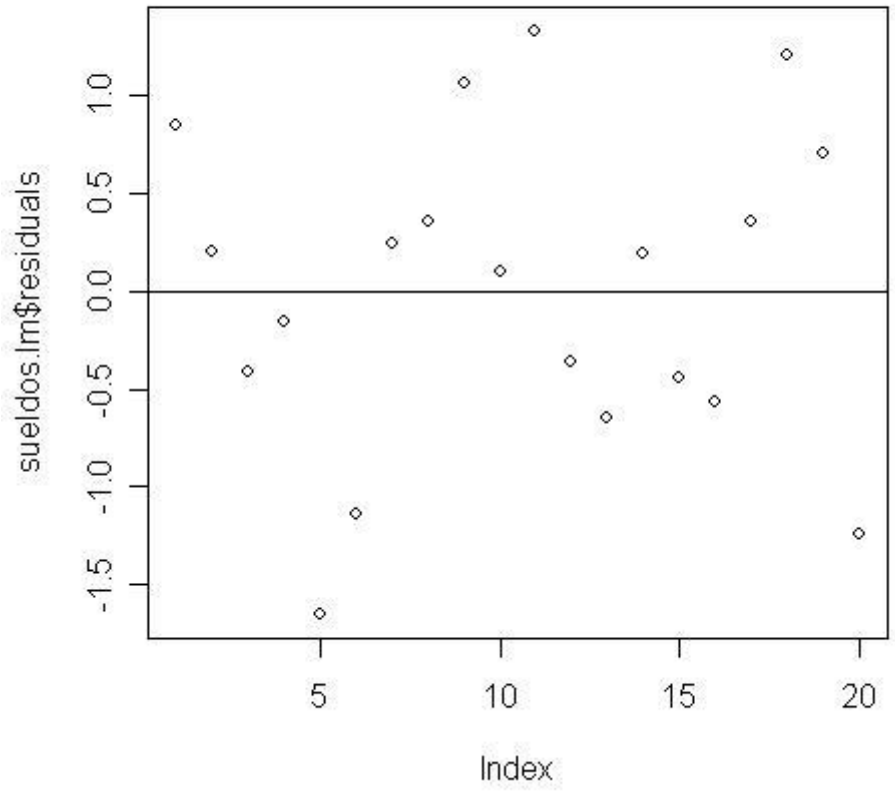
15 16 17 18 19 20

-0.4395052 -0.5681066 0.3561941 1.2104948 0.7090961 -1.2409039

Si los queremos dibujar:

> plot(sueldos.lm$residuals)

> abline(h=0)



Dibujamos una línea horizontal en y=0 porque los residuos cumplen la propiedad de estar centrados alrededor de dicha línea. Si en el gráfico observamos algún dato que se aleja mucho por arriba o por abajo, eso quiere decir que para ese dato, el modelo de regresión no predijo bien, dado que su residuo es elevado. Eso quiere decir que tal dato no es bien explicado por el modelo, y así tenemos una forma de detectar tal situación. Por ejemplo, algo así podría pasar para el sueldo del director, tal vez.

Los valores predichos para los datos observados son los fitted values:

> sueldos.lm$fitted.values

1 2 3 4 5 6 7 8

3.758107 6.769505 6.769505 6.769505 5.263806 11.286603 3.758107 8.275205

9 10 11 12 13 14 15 16

8.275205 3.758107 11.286603 9.780904 8.275205 9.780904 6.769505 3.758107

17 18 19 20

5.263806 6.769505 9.780904 9.780904

La lista anterior muestra el sueldo predicho para cada individuo por el modelo de regresión ajustado.

Hay otras características de interés en el modelo, de naturaleza estadística. Por ejemplo, el R-Squared mide la variabilidad de los datos explicada por el modelo. En el ejemplo anterior es aproximadamente 0.90 lo que quiere decir que más del 90% de la variabilidad de los datos fue recogida por el modelo: esto es, es un buen modelo.

En cuanto al uso del modelo para predecir, si quisiéramos predecir el sueldo de un directivo recién entrado en la empresa (0 años de antigüedad), escribiríamos:

> predict.lm(sueldos.lm,data.frame(antiguedad=0))

[1] 3.758107

Luego la predicción es 3.75 millones de pesetas. Si queremos predecir los sueldos esperables de directivos que lleven año y medio, 2 y 3.5 años, escribiríamos:

> predict.lm(sueldos.lm,data.frame(antiguedad=c(1.5,2,3.5)))

1 2 3

6.016656 6.769505 9.028054

Regresión lineal múltiple

Funciona de similar manera al modelo de regresión lineal simple, con la diferencia de que lo que se estima es un plano de regresión.

Vamos a cargar unos datos de R sobre coches:

> data(mtcars)

> attach(mtcars)

Mostramos los primeros registros:

> mtcars[1:5,]

mpg cyl disp hp drat   wt qsec vs am gear carb

Mazda RX4   21.0   6 160 110 3.90 2.620 16.46 0 1   4   4

Mazda RX4 Wag   21.0   6 160 110 3.90 2.875 17.02 0 1   4   4

Datsun 710   22.8   4 108 93 3.85 2.320 18.61 1 1   4   1

Hornet 4 Drive   21.4   6 258 110 3.08 3.215 19.44 1 0   3   1

Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02 0 0   3   2

Vamos a explicar el consumo (mpg) en función de la potencia (hp) y del peso (wt):

> cars.lm = lm(mpg~hp+wt)

Observemos que mpg es la variable que se explica, y el signo más (+) indica sólo yuxtaposición, esto es, que las variables que explican con hp y wt:

Observemos el resultado:

> summary(cars.lm)

Call:

lm(formula = mpg ~ hp + wt)

Residuals:

Min   1Q Median   3Q   Max

-3.941 -1.600 -0.182 1.050 5.854

Coefficients:

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

(Intercept) 37.22727 1.59879 23.285 < 2e-16 *

**hp -0.03177 0.00903 -3.519 0.00145 **

wt -3.87783 0.63273 -6.129 1.12e-06 *

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

Residual standard error: 2.593 on 29 degrees of freedom

Multiple R-Squared: 0.8268, Adjusted R-squared: 0.8148

F-statistic: 69.21 on 2 and 29 degrees of freedom, p-value: 9.109e-012

Por tanto, el modelo es:

millas recorridas por galón = 37.22 - 0.03 potencia - 3.87 peso

Esto es, cuanto más potente es el coche, menos millas recorre (de ahí el signo negativo de su coeficiente), y cuanto más pesa, menos millas recorre. El R-squared es del 82% , lo que quiere decir que esas dos variables explican bastante bien el consumo.

Podemos dibujar los residuos para ver si hay algún coche que se comporta de modo muy distinto a los demás:

> plot(cars.lm$residuals)

> abline(h=0)

Si queremos predecir las millas recorridas por galón por un coche con 150 caballos y peso 2.5:

> predict.lm(cars.lm,data.frame(hp=150,wt=2.5))

[1] 22.76675