FEIR 40: Modelos de Regresión

Apuntes del curso FEIR3, curso 2014/15 actualizados. `Última actualización:` martes 02 abril 2019, 18:56:26

María Elvira Ferre Jaén

Índice

1. Introducción

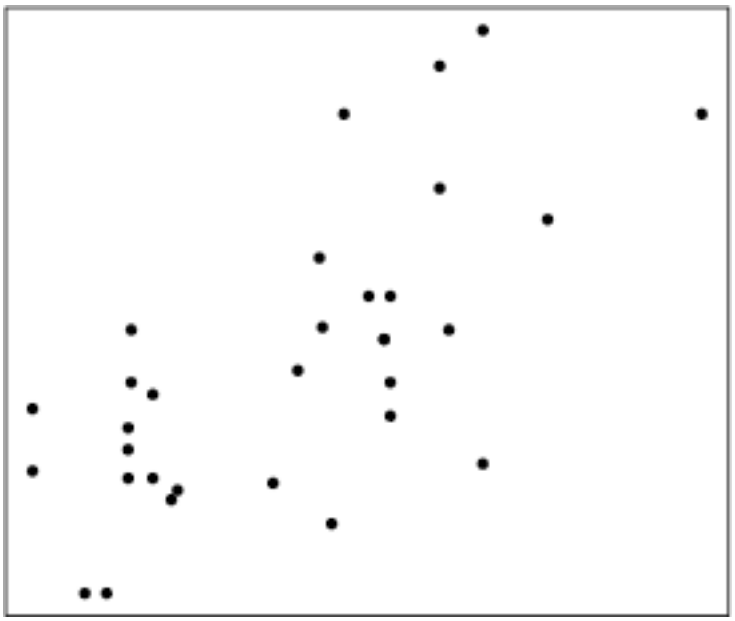
Como referencia bibliográfica básica para el desarrollo de este capítulo hemos utilizado el libro A. Field, Miles, & Field (2012), aunque también nos hemos servido de numerosos documentos que iremos referenciando a lo largo del texto.

1.1. Aproximación no formal al modelo de regresión lineal

El análisis de regresión lineal es una técnica estadística utilizada para estudiar la relación entre variables. A menudo resulta de interés conocer el efecto que una o varias variables pueden causar sobre otra, e incluso predecir en mayor o menor grado valores de una variable a partir de otra. Por ejemplo, supongamos que queremos estudiar si la altura de los padres influye significativamente en la de los hijos.

La regresión es el conjunto de técnicas usadas para explorar y cuantificar la relación de dependencia entre una variable cuantitativa llamada *variable dependiente o respuesta* y una o más variables independientes llamadas *variables predictoras*.

El primer paso para determinar si puede existir o no dependencia/relación entre variables es representando gráficamente los pares de valores observados mediante una nube de puntos, lo que se conoce como *diagrama de dispersión* (SPSS, 2007).



Una vez representados los datos y tras detectar que entre dos o más variables existe una relación el siguiente paso sería intentar modelizar dicha relación.

La modelización estadística más sencilla para expresar la variable dependiente a través de sus variables predictoras es mediante una ecuación lineal de la forma $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_k$.

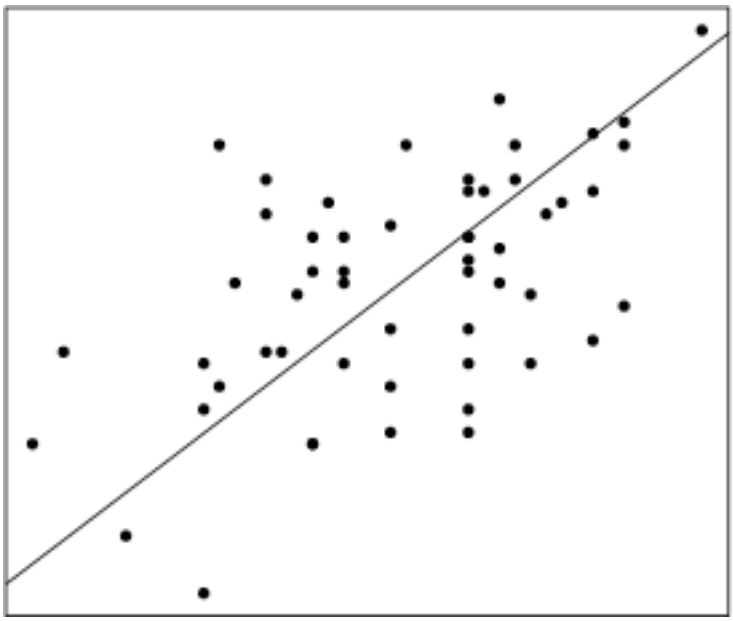
El caso más simple para una única variable sería una recta $Y = mx + n$ y recibirá el nombre de *regresión lineal simple*. Cuando $k > 1$ la llamaremos *regresión múltiple*.

Así, el proceso consistiría en ajustar la recta a nuestro conjunto de datos y crear una expresión matemática que permita predecir, de forma aproximada, el valor de la variable dependiente en un individuo cuando se conoce el valor de una variable predictora (regresión simple) o varias variables predictoras (regresión múltiple)

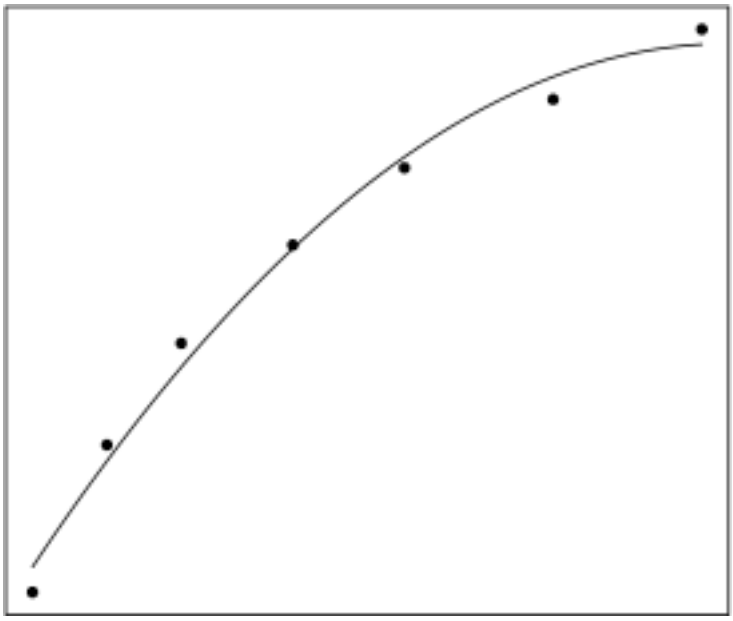
en ese mismo individuo. A la ecuación que representa esta relación se le llama **modelo de regresión** (Pérez, 2014).

Podemos considerar varias formas de estimar los parámetros de la ecuación del modelo de regresión. Sin embargo, nos centraremos en *el método de mínimos cuadrados* por ser el de más amplia aceptación, aunque existan también otros métodos como el de máxima verosimilitud.

Una vez creado el modelo de regresión, lo primero que debemos **analizar es su utilidad explicando los datos** que queremos relacionar. Así por ejemplo, la recta del siguiente gráfico describe, aproximadamente, la relación lineal entre las variables. (Sánchez, 2011)



En cambio, los datos del gráfico siguiente no se puede explicar mediante una la ecuación lineal.



Aunque sirve para hacernos una idea, no es suficiente con ver gráficamente que se trata de un modelo útil, sino que debemos **comprobar que el modelo de regresión cumple unos ciertos supuestos "matemáticos"**, que nos hablan de la bondad y calidad del modelo para nuestros fines.

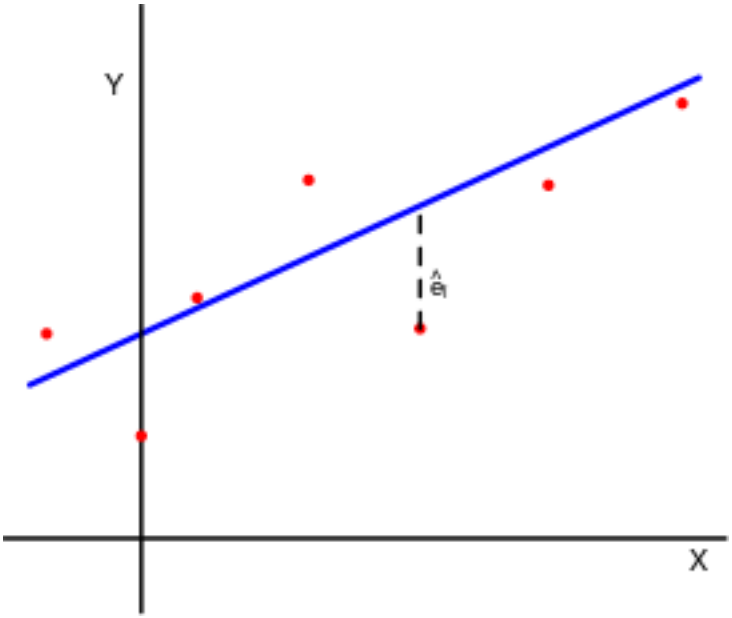
Que la recta se ajuste a los datos no significa que el modelo sea correcto, depende del uso que queramos darle. Si sólo pretendemos hallar la relación entre dos variables, con calcular la recta de mínimos cuadrados es suficiente, esa recta describe la relación entre las variables, otra cosa es que los datos tenga una buena relación lineal. Podría ser que los datos tuvieran muy mala relación lineal y la recta seguiría existiendo. En cambio si pretendemos describir la estructura general de los datos, o inferir/predecir con la recta de regresión debemos comprobar que se verifican unas reglas ya establecidas y aceptadas que aseguran que nuestro modelo es bueno.

Con tal fin existen una serie de procedimientos de diagnostico que nos informaran sobre la estabilidad e idoneidad del modelo de regresión. Los **supuestos** que tendremos que comprobar son

- En el modelo de regresión: linealidad
- En los residuos:
  - normalidad
  - varianza constante
  - valores atípicos

Por otro lado, para cada conjunto de datos existen varias rectas con las que podríamos resumir la tendencia general de los mismos. Necesitamos encontrar la recta del mejor ajuste, aquella que da lugar a la menor diferencia entre los datos originales y los estimados por la recta.

Para buscar esta recta utilizaremos el **criterio de mínimos cuadrados**, método con el que calculamos la recta que minimiza la suma de *los residuos*, esto es, las distancias verticales entre cada punto y la recta.



El objetivo que hay tras este método es que los residuos sean pequeños, lo que matemáticamente se traduce en que tengan media cero y en que bailen lo menos posible, es decir, en una σ^2 pequeña. De aquí es de donde surgen todos los supuestos que se le exigen al modelo de regresión lineal.

Uno de los resultados que obtenemos al aplicar el método de los mínimos cuadrados es que el coeficiente m, que cuantifica la relación entre la x y la y en nuestra ecuación, es en realidad el coeficiente de correlación de Pearson. Por ello, antes crear el modelo de regresión tenemos que analizar si este coeficiente es significativamente distinto de cero y en caso de serlo plantearemos el modelo de regresión lineal.

1.2. Correlación lineal

Un análisis de correlación nos permite cuantificar el grado de asociación lineal entre variables continuas, indica la fuerza y dirección de la relación lineal entre dos o más variables. Cuando exista dicha relación se podrá proceder a la obtención del modelo de regresión (simple o múltiple) que veremos posteriormente (Pérez, 2014).

Existen *diferentes tipos de correlación*, la correlación simple, la correlación múltiple y la correlación parcial. Utilizaremos la correlación simple cuando contemos con una sola variable predictora para explicar una respuesta, y los coeficientes de correlación parcial y múltiple cuando tengamos varios predictores.

1.2.1. Correlación lineal simple

Utilizamos la correlación lineal simple para estudiar el grado de variación conjunta entre dos o más variables. Queremos detectar si la variación de una de las variables tiene conexión con la variación de la otra, esperamos que si una variable de desvía de la media, la otra variable se desvíe de la media de manera similar.

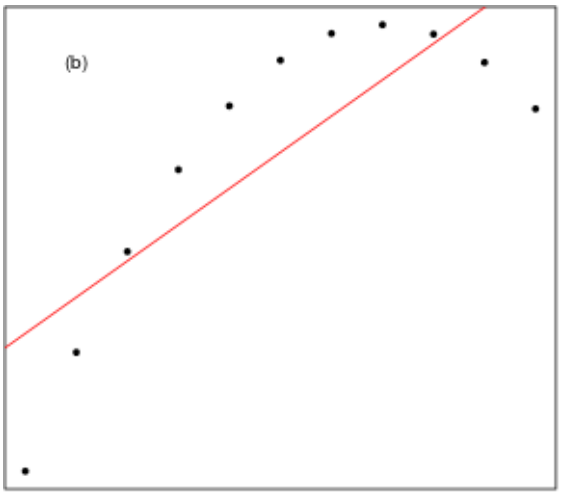
Una *relación lineal positiva* entre dos variables indica que los valores de las dos variables varían de forma parecida: los sujetos que puntúan alto en una variable tienden a puntuar alto en la otra y los que puntúan bajo en la primera tienden a puntuar bajo en la segunda, existe una relación directa entre ambas variables.

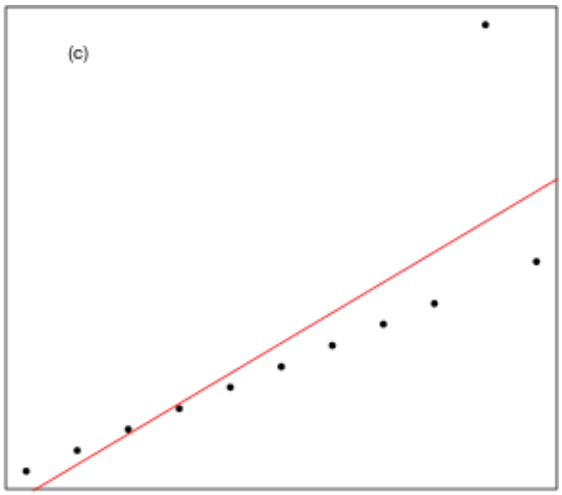
Una *relación lineal negativa* significa que los valores de las dos variables tienen una relación inversa: valores pequeños de una variable van asociados ahora a valores grandes de la otra y, equivalentemente, valores grandes de una se asocian a valores pequeños de la otra.

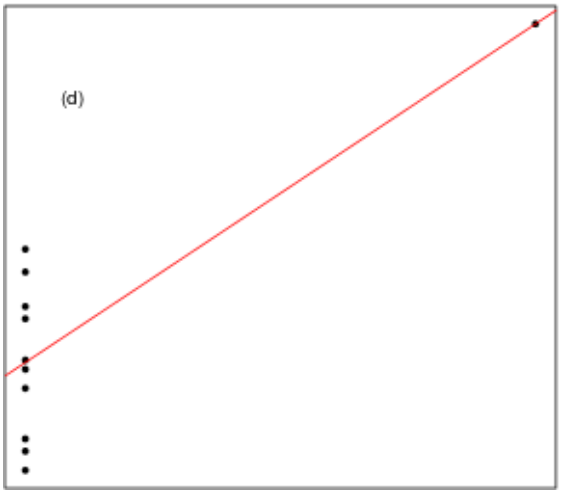
La forma más directa e intuitiva de formarnos una primera impresión sobre el tipo de relación existente entre dos variables es a través de un *diagrama de dispersión*. Se trata de un gráfico en el que una de las variables, X, se coloca en el eje de abscisas, la otra, Y, en el de ordenadas y los pares (x_i, y_i) se representan como una *nube de puntos*. La forma de la nube de puntos nos informa sobre el tipo de relación existente entre las variables.

Una regla fundamental es que cuanta mayor correlación haya entre dos variables en la representación bidimensional, más próximos a la recta estarán los valores.

Veamos un ejemplo: en el siguiente gráfico mostramos cuatro diagramas de dispersión que reflejan cuatro tipos de relación diferentes (Ferrari & Head, 2010).







Para todos estos conjuntos de datos la recta de regresión es la misma

$$\hat{y} = 3 + 0{,}5 \times x$$

con los coeficientes significativos con un nivel de significación $< 0{,}01$, y además todos tienen la misma

$R^2 = 0{,}67$ y $\hat{\sigma} = 1{,}24$.

Sin embargo, solamente podemos escribir mediante un modelo lineal los datos del gráfico (a). El gráfico (b) muestra un conjunto de datos es claramente no lineal y sería mejor ajustarlo mediante una función cuadrática.

El gráfico (c) muestra un conjunto de datos que tiene un punto que distorsiona los coeficientes de la recta ajustada. Por último, el gráfico muestra un conjunto de datos totalmente inapropiado para un ajuste lineal, la recta ajustada está determinada esencialmente por la observación extrema (Ali S. Hadi, 2006).

Tras haber realizado una representación de los datos, una buena manera de cuantificar la relación a entre dos variables es mediante la **covarianza**

$$r = Cov(X, Y) = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{N - 1},$$

donde N es el número de observaciones.

Sin embargo, la covarianza no es una medida útil para comparar rectas de regresión de variables distintas, o comparar el grado de asociación lineal entre distintos pares de variables, ya que depende de las escalas de medida de las variables. La solución está en estandarizarla y es de aquí de donde surgen llamados *coeficientes de correlación*.

1.2.1.1. Coeficientes de correlación

El más importante de los coeficientes de correlación es el *Coeficiente de Pearson*, que explicaremos en mayor profundidad, pero también están la *Rho de Spearman* y la *Tau de Kendall*. Veamos sus **propiedades generales**:

- Todos los coeficientes varían entre -1 y 1.
- Si el coeficiente de correlación es -1 existe correlación negativa, es decir, a medida que una variable aumenta, la otra disminuye. Cuando el coeficiente es 1 hay correlación positiva, cuando aumenta una variable, también aumenta la otra.
- Un valor cercano o igual a cero indica poca o nula relación lineal entre las variables.
- Se utilizan como una medida de la fuerza de asociación: valores ±0,1 representan pequeñas asociación, ±0,3 asociación mediana, ±0,5 asociación moderada, ±0,7 gran asociación y ±0,9 asociación muy alta.

Las principales **diferencias entre los coeficientes** son:

- La correlación de *Pearson* funciona bien con variables cuantitativas y que sigan bien la distribución normal.
- La correlación de *Spearman* se utiliza para datos ordinales o de intervalo que **no** satisfacen la condición de normalidad. (usualmente tiene valores muy parecidos a la de Pearson).
- La correlación de *Kendall* es una medida no paramétrica para el estudio de la correlación. Debemos utilizar este coeficiente en vez de la de Spearman cuando tengamos un conjunto de datos pequeño y muchas puntuaciones estén en el mismo nivel.

1.2.1.2. Coeficiente de Pearson

El coeficiente de correlación lineal de *Pearson (r)* viene definido como

$$r = \frac{Cov(X,Y)Sd(Y)}{Sd(X)} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

y se trata de la correlación entre las variables X e Y estandarizada.

Para que el coeficientes de correlación de Pearson sea una medida precisa de la relación lineal entre dos variables exige que las variables sean cuantitativas y que las dos variables se distribuyan normalmente, aunque podemos hacer una excepción si sólo una de las variables es normal y la otra es categórica con dos categorías. Si los datos no son normales o cuantitativos entonces se debe usar otro tipo de coeficientes como el de Spearman o el de Kendall.

Las *principales características* de este coeficiente son:

1. Medida de asociación lineal libre de escala
2. Valores comprendidos entre -1 y 1
3. Invariante a transformaciones lineales de las variables.

Su *interpretación* es la siguiente:

- Si r = 0 (asociación lineal nula) no existe relación entre las variables.
- Si r = 1 o -1 (asociación lineal perfecta).
- Cuando r > 0 (correlación positiva) existe una relación directa entre las variables
- Cuando r < 0 (correlación negativa) existe una relación inversa entre las variables.

El coeficiente hay que interpretarlo en magnitud, es decir, tomar su valor absoluto. Esto significa que cuanto más cerca estemos de los extremos (± 1) más relación existe entre las variables. Por eso, una correlación con valor $r = -0{,}9$ es más fuerte que una con $r = 0{,}7$, pues 0,9 es más grande que 0,7 aunque sea negativa.

Por último queda ver que la correlación entre las variables es *significativa*, es un valor fiable que no cambiaría mucho en otra muestra tomada en las mismas condiciones.

Una *correlación* será *significativa* si su p-valor es inferior a 0,05, de lo contrario supondremos que $r = 0$.

Según esto podemos decir que una $r = 0{,}8$ con un p-valor de 0,26 es en realidad una correlación más baja que una $r = 0{,}4$ con $p = 0{,}001$, ya que al no ser significativa la $r = 0{,}8$ no es una medida fiable, puede ser un efecto del azar del muestreo. De la misma forma que en esta muestra hemos calculado una $r = 0{,}8$ en otra muestra tomada en las mismas condiciones podríamos obtener $r = -0{,}8$. Debido a ello, y ante la duda, es mejor afirmar que no hay relación, que r es igual a 0. Para el caso de la correlación $r = 0{,}4$, aunque no se trata de una gran correlación, sí que es fiable (Pérez, 2014).

1.2.1.3. Coeficiente de Spearman

El coeficiente de correlación de Spearman es el mismo que el coeficiente de Pearson pero tras transformar las puntuaciones originales a rangos.

El coeficiente de Sperman puede utilizarse como una alternativa a Pearson cuando las variables son ordinales y/o no se incumple el supuesto de normalidad.

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)},$$

donde d es la distancia entre los rangos (X menos Y) y n es el número de datos.

1.2.1.4. Tau de Kendall

Es un coeficiente de correlación no paramétrico que se basa en el concepto de inversión, no-inversión y empate. Se calcula a partir de los desórdenes entre los rangos, su fórmula es la siguiente

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)},$$

donde C es el número de pares concordantes, aquellos en los que el rango de la segunda variable es mayor que el rango de la primera variable, y D el número de pares discordantes, cuando el rango de la segunda es igual o menor que el rango de la variable primera.

Podemos utilizarlo, al igual que en el caso de Spearman, cuando las variables no alcanzan el nivel de medida de intervalo y no podemos suponer que la distribución poblacional conjunta de las variables sea normal.

1.2.2. La correlación simple en R

Para el cálculo del coeficiente de correlación vamos a utilizar la función `cor( )`, que tiene la forma general

`cor( x,y use = "string", method = "tipo de correlación" )`, donde:

- `x`: variable numérica o un dataframe.
- `y`: otra variable numérica (si `x` es un dataframe no hay que especificarla).
- `use`: especifica el tratamiento para los datos perdidos.
  - `use = all.obs`: se asume que no existen valores perdidos, si existiera alguno produciría un error
  - `use = everything`: cualquier correlación que envuelva una variable con valores perdidos se tratará como *missing*
  - `use = complete.obs`: sólo se ejecutan los casos que están completos para todas las variables
  - `use = pairwise.complete.obs`: correlación entre pares de variables que se ejecuta para los casos que estén completos para estas dos variables.
- `method`: especifica el tipo de correlación. Podemos elegir entre `"pearson"` (por defecto), `"kendall"`, o `"spearman"`).

Ejemplo: *Calculamos la correlación entre las variables "Horsepower" y "Weight" del archivo Cars93*

```
library( MASS )
data( Cars93 )
df <- data.frame( Cars93 )
cor( df$Horsepower, df$Weight, method = "pearson" )
```

```
## [1] 0.7387975
```

1.2.2.1. Correlación significativa

No resulta suficiente la estimación puntual del coeficiente de correlación. Para asegurar la existencia de relación entre las variables dependiente y predictora debemos realizar un **test para estudiar la significación estadística**.

Enfrentaremos la hipótesis nula ($H_0 : r = 0$, no relación) frente a la hipótesis alternativa ($H_1 : r \neq 0$ existe relación) mediante la función `cor.test()` que toma la siguiente forma:

`cor.test( x, y, alternative = " ", method = " " )` donde

- `x` e `y` son las variables a estudiar
- `alternative` será "two.side", "less".º "greater"
- `method` especificaremos el tipo de correlación (pearson, spearman o kendall).

```
cor.test( df$Horsepower, df$Weight, method = "pearson" )
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$Horsepower and df$Weight
## t = 10.458, df = 91, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6298867 0.8192147
## sample estimates:
##       cor
## 0.7387975
```

Por defecto selecciona el método de Pearson. Fijándonos en el p-valor podemos asegurar la existencia de correlación entre las variables. Además este test estima el valor de la correlación y nos da un intervalo de confianza para dicho valor.

En el caso de querer calcular el coeficiente de correlación simple entre **varias variables de un archivo** no tenemos porque hacerlo dos a dos, podemos crear una matriz de correlaciones:

```
newdf <- data.frame( df$Price, df$Weight, df$RPM, df$Horsepower )
cor( newdf, use = "everything", method = "pearson" )
```

```
##                   df.Price  df.Weight       df.RPM df.Horsepower
## df.Price       1.000000000  0.6471790 -0.004954931    0.78821758
## df.Weight      0.647179005  1.0000000 -0.427931473    0.73879752
## df.RPM        -0.004954931 -0.4279315  1.000000000    0.03668821
## df.Horsepower  0.788217578  0.7387975  0.036688212    1.00000000
```

Además de las correlaciones queremos también los p-valores pero la función `cor.test` no funciona con matrices así que utilizamos una *nueva función*:

```
library( "psych" )
corr.test( newdf, use = "complete", method = "pearson" )
```

```
## Call:corr.test(x = newdf, use = "complete", method = "pearson")
## Correlation matrix
##               df.Price df.Weight df.RPM df.Horsepower
## df.Price          1.00      0.65   0.00          0.79
## df.Weight         0.65      1.00  -0.43          0.74
## df.RPM            0.00     -0.43   1.00          0.04
## df.Horsepower     0.79      0.74   0.04          1.00
## Sample Size
## [1] 93
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##               df.Price df.Weight df.RPM df.Horsepower
## df.Price          0.00         0   1.00             0
## df.Weight         0.00         0   0.00             0
## df.RPM            0.96         0   0.00             1
## df.Horsepower     0.00         0   0.73             0
##
```

```
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Analizando la salida vemos que se obtienen las mismas correlaciones que con la función `cor( )`, aunque aproximadas, y que los p-valores muy bajos ($p < 0{,}05$) han sido aproximados a 0, así que todas las correlaciones son significativas.

Observación: El procedimiento para hacer una correlación de Spearman o Kendall es el mismo que para una correlación de Pearson excepto que tenemos que especificar que queremos otra correlación, que se realiza mediante el `method = "spearman"` o `method = "kendall"` para `cor( )`, `cor.test( )`y `corr.test( )`.

1.2.3. Correlación parcial

La *correlación parcial* es una correlación entre dos variables en la que el efecto de otras variables auxiliares se mantiene contante, se busca la relación entre dos variables mientras 'se controla' el efecto de una o más variables adicionales.

Esta medida surge ya que en ocasiones las variables continuas con las que pretendemos predecir una respuesta no son totalmente independientes entre sí lo provoca que las variables compartan y solapen información a la hora de explicar la respuesta.

Por ejemplo, si queremos estudiar la relación entre las variables "inteligencia"ƨ "rendimiento escolar" tendremos que tener en cuenta terceras variables como el "número de horas de estudio", el "nivel educativo de los padres".

La correlación parcial se trata, por tanto, de un coeficiente de correlación que nos da una idea sobre la relación lineal existente entre dos variables pero ajustada a los efectos lineales que sobre las mismas puedan tener otra o más variables que intervengan. Utilizaremos la función `pcor( )` incluida en el paquete `ppcor`. Su forma general es:

```
pcor( var1 , var2 , control1 , control2,..., method = " " )
```

- `var1` y `var2` son las variables a ser correladas.
- `control1` , `control2` y las siguientes posibles son las variables con las que controlamos la correlación.
- `method = c( "pearson", "kendall", "spearman" )`, que por defecto empleará `spearman`.

Vamos a **calcular la correlación parcial** entre `Price` y `Weight` controlando el efecto de la variable `Length`.

```
library( "ppcor" )
pcor.test( df$Price, df$Weight, df$Length )
```

```
##    estimate      p.value statistic  n gp  Method
## 1 0.4718103 2.058463e-06   5.07654 93  1 pearson
```

tenemos que

- `estimate` es el coeficiente de correlación parcial entre las dos variables.
- `p.value` es el p-valor del test.
- `statistic` es el valor del estadístico del test.
- `n` es el número de muestras.
- `gn` es el número de variables.
- `method` es el método de correlación empleado (spearman, pearson o kendall).

Si calculamos la correlación simple entre las variables `Price` y `Weight`:

```
cor( df$Price, df$Weight )
```

```
## [1] 0.647179
```

observamos que tiene un valor diferente a la correlación parcial controlada por `df$Length`. Por tanto, las variables `Price` y `Weight` están influenciadas por `Length` ya que al controlar su efecto la correlación se reduce de 0,647 a 0,47.

1.2.4. Otras consideraciones

1.2.4.1. Causalidad

Debemos tener precaución a la hora de interpretar los coeficientes de correlación ya que estos no nos indican la dirección de *causalidad* de las variables, no nos dicen nada sobre qué variable causa que la otra varie.

Aunque es intuitivo pensar que ver anuncios nos provoque comprar más paquetes de galletas, no hay razón estadística por la que comprar paquetes de galletas no nos pueda provocar ver más anuncios. Pese a que la última conclusión tiene menos sentido, el coeficiente de correlación no nos dice que no puede ser cierta, para un matemático la dirección no importa.

Por otro lado existe *el problema de la tercera variable*. Este nos dice que no podemos asumir causalidad entre dos variables porque podría haber otras variables afectando a los resultados.

1.2.4.2. Tamaño del efecto

Recordemos que $(\hat{Y}_i - \bar{Y}) = \hat{\beta}_1(X_i - \bar{X})$ y que $\hat{\beta}_1 = r = \frac{Cov(X,Y)Sd(Y)}{Sd(X)}$ así que

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} = \hat{\beta}_1{}^2 \frac{\sum_{i=1}^{n}(\hat{X}_i - \bar{X})}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} = Cor(Y,X)^2 = r^2.$$

Entonces, aunque no podemos hacer conclusiones directas sobre la causalidad de una correlación, para dos variables sí podemos elevar el coeficiente de correlación al cuadrado y utilizarlo como una medida de la cantidad de variabilidad que una variable comparte con la otra. Es lo que se conoce como **coeficiente de determinación,** R^2, y es una medida tremendamente útil de la importancia de un efecto.

Para calcular este coeficiente, R^2, podemos elevar al cuadrado tanto el coeficiente de Pearson, r, como el coeficiente de Spearman r_s, ya que este usa la misma ecuación que Pearson. Lo único que debemos tener en cuenta es que el resultante R_S^2 hay que interpretarlo como la proporción de varianza en las categorías que las dos variables comparten.

El coeficiente de Kendall, sin embargo, no es numéricamente similar a r o r_s por lo que τ^2 no nos dice nada sobre la proporción de varianza compartida por las dos variables.

Calculamos el coeficiente de determinación para el conjunto de datos `newdf` anterior:

```
cor( newdf, use = "everything" ) ^ 2
```

```
##                   df.Price df.Weight       df.RPM df.Horsepower
## df.Price      1.000000e+00 0.4188407 2.455135e-05   0.621286950
## df.Weight     4.188407e-01 1.0000000 1.831253e-01   0.545821769
## df.RPM        2.455135e-05 0.1831253 1.000000e+00   0.001346025
## df.Horsepower 6.212870e-01 0.5458218 1.346025e-03   1.000000000
```

Se observa que el tamaño del efecto de `EngineSize` sobre `Weight`es muy elevado, así como para `Lenght` y `Weight`, siendo sin embargo muy bajo el efecto de `Lenght` sobre `Price`. Si queremos expresar estos valores en porcentajes basta multiplicar por 100.

1.2.4.3. Comunicar los coeficientes de correlación

Sólo hay que decir cómo de grande es y qué valor de significación tiene. La forma de reportar los coeficientes sería

- Existe una relación significativa entre `var1` y `var2`, $r = 0{,}78$, $p < 0{,}05$.
- `Var1` está significativamente correlacionada con `var2`, $r_s = 0{,}57$, y con `var3`, $r_s = 0{,}50$; la `var2` está también correlacionada con `var3`, $r_s = 0{,}83$ (todas $p < 0{,}01$).
- `Var2` está significativamente relacionada con `var1`, $\tau = -0{,}45$, $p < 0{,}01$.

1.2.5. Ejemplo de los tractores

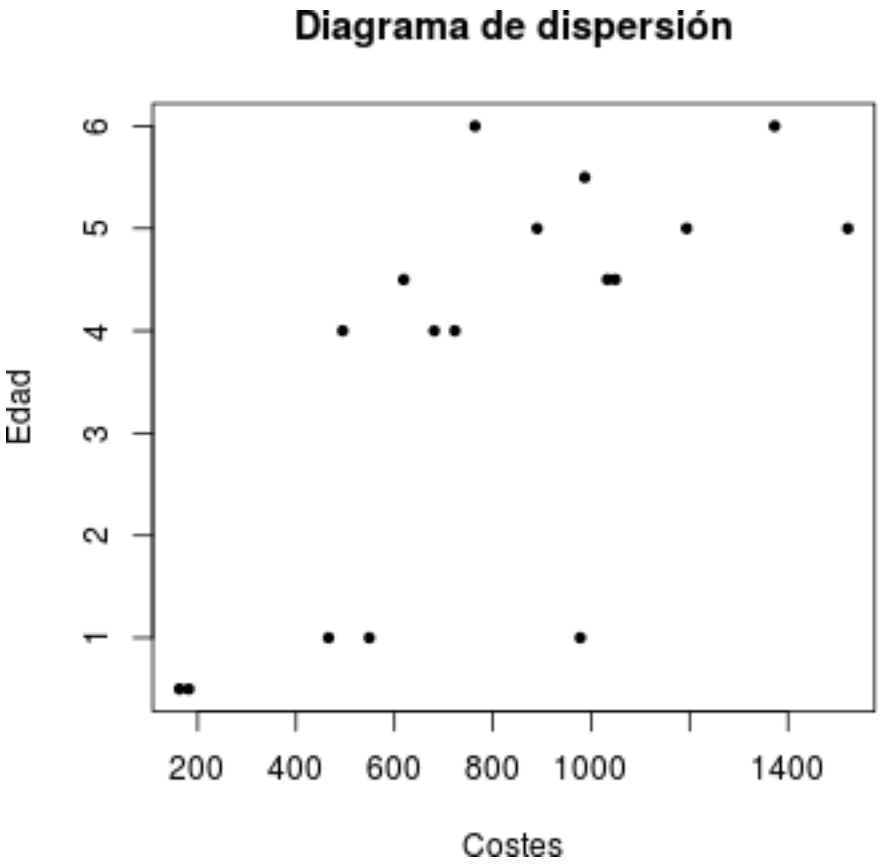
Supongamos que una empresa de tractores que pretende saber qué le es más conveniente, si renovar su flota de tractores, seguir manteniendo la que tienen o cambiar solo una parte. Utilizamos el conjunto de datos tractores.rda para intentar relacionar los costes de manutención de tractores con la edad de éstos.

Comenzamos calculando la correlación entre edad y costes, y realizamos el correspondiente gráfico de dispersión

```
load( "files/40A-tractores.rda" )
cor.test( tractores$costes, tractores$edad )
```

```
##
##  Pearson's product-moment correlation
##
## data:  tractores$costes and tractores$edad
## t = 3.6992, df = 15, p-value = 0.002143
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3144325 0.8793971
## sample estimates:
##       cor
## 0.6906927
```

```
plot( tractores$costes, tractores$edad, pch = 20, xlab = "Costes",
      ylab = "Edad", main = "Diagrama de dispersión" )
```

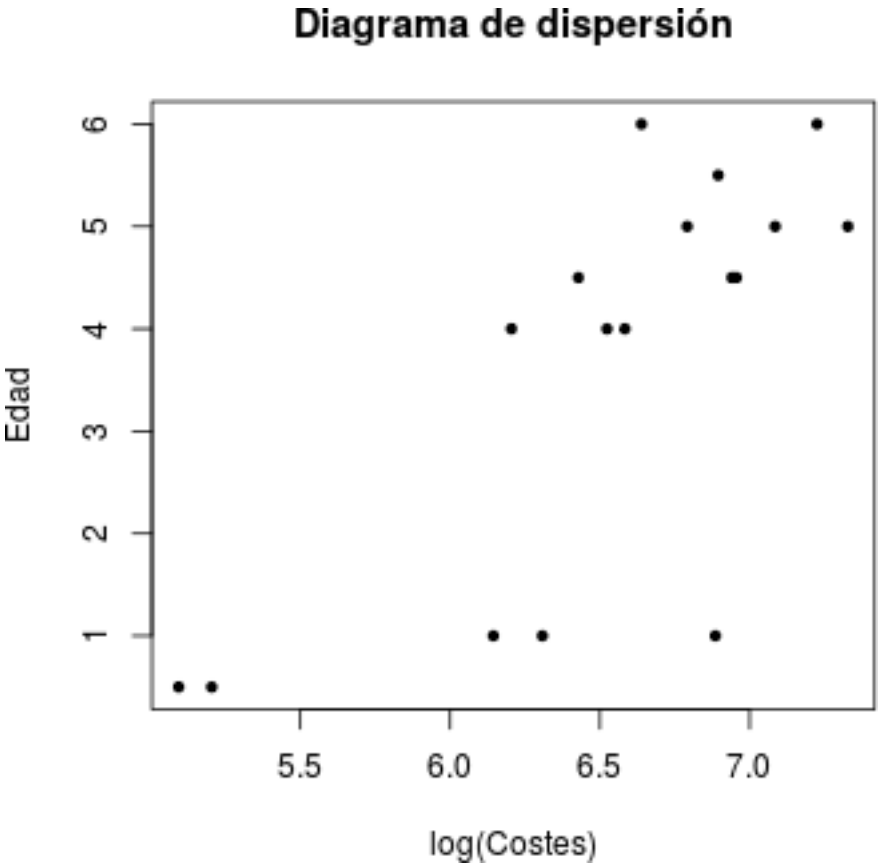


Como existe mucha diferencia en las escalas de medida aplicamos la función logaritmo, `log( )`, a los datos ya que es la que más puede reducir estos valores. Creamos una nueva variable que sea el logaritmo de los costes y realizamos de nuevo el análisis de correlación

```
tractores$logcostes <- log( tractores$costes )
cor.test(  tractores$logcostes, tractores$edad )
```

```
##
##  Pearson's product-moment correlation
##
## data:  tractores$logcostes and tractores$edad
## t = 4.2027, df = 15, p-value = 0.0007687
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3939673 0.8984522
## sample estimates:
##       cor
## 0.7353647
```

```
plot( tractores$logcostes, tractores$edad ,pch = 20, xlab="log(Costes)", ylab = "Edad",
      main = "Diagrama de dispersión" )
```



Como vemos la correlación ahora es más elevada y los puntos están menos dispersos en el plano.

Una vez detectada una relación significativa entre dos o más variables, el siguiente paso es intentar crear una fórmula matemática que formalice esa relación y que permita calcular pronósticos de una variable a partir de una o varias variables evaluadas en un individuo concreto. Este proceso se conoce como *regresión* y es el que estudiaremos en los siguientes apartados.

2. Regresión lineal simple

Para el desarrollo de los siguientes tres apartados nos hemos servido esencialmente de Sánchez (2011).

2.1. Introducción

El caso de modelo de regresión más sencillo es la construcción de una recta que modelice la relación que hay entre la variable respuesta, Y, y la variable predictora X. El modelo tiene la forma

$$Y = \beta_0 + \beta_1 X + e,$$

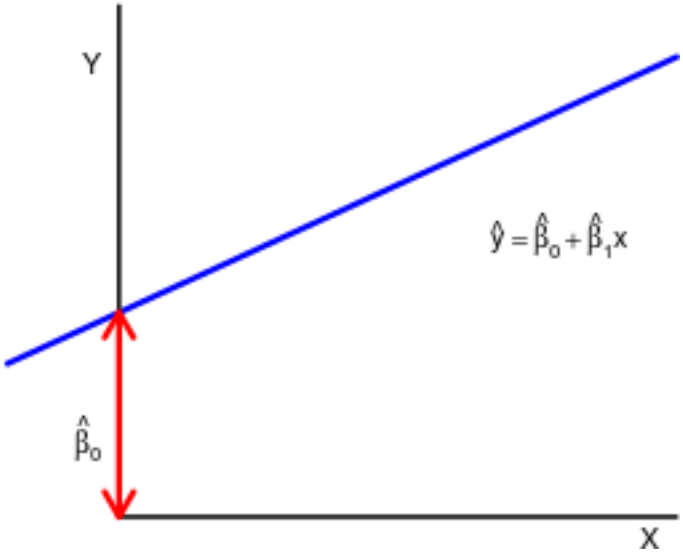
donde β_0 y β_1 se conocen como *coeficientes de regresión* y son, respectivamente, la ordenada en el origen (punto de corte con el eje Y) y la pendiente de la recta del modelo de regresión.

En la ecuación e es el error aleatorio, representa la diferencia entre el valor ajustado por la recta y el valor real. Refleja la ausencia de dependencia perfecta entre las variables, la relación está sujeta a incertidumbre.

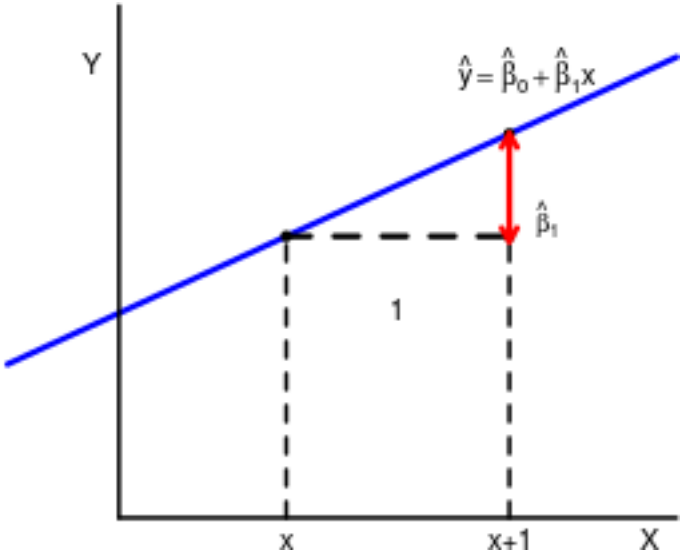
Por ejemplo, en el consumo de gasolina de un vehículo, Y, influyen la velocidad X y una serie de factores como el efecto conductor, el tipo de carretera, las condiciones ambientales, etc. Todos estos elementos quedarían englobados en el error e.

Los coeficientes de regresión se pueden interpretar como:

- β_0 el valor medio de la variable dependiente cuando la predictora es cero.



- β_1 el efecto medio (positivo o negativo) sobre la variable dependiente al aumentar en una unidad el valor de la predictora X.



Una recta que tiene una pendiente con valor positivo describe una relación positiva, mientras que una recta con una pendiente negativa describe una relación negativa. Entonces tenemos básicamente que la pendiente (β_1) nos da la apariencia del modelo (su forma) y la ordenada en el origen (β_0) nos dice dónde se sitúa el modelo en el plano.

2.2. Estructura del modelo de regresión simple

El modelo de regresión lineal simple tiene la siguiente estructura

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

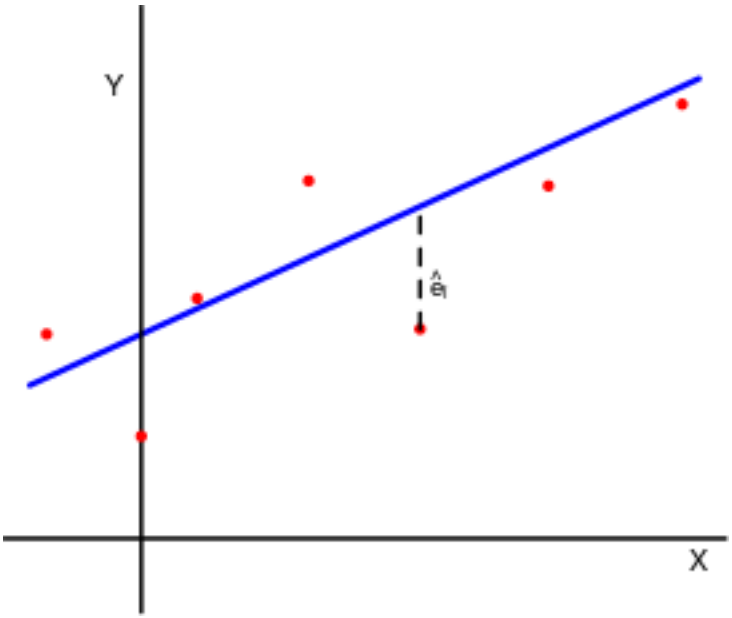
para $i = 1, ..., n$. Vamos a estudiarlo más detenidamente.

Supongamos que hemos ajustado una recta de regresión a un conjunto de datos, y sea (x_i, y_i) un punto cualquiera de la nube. Entonces y_i se puede descomponer como

$$y_i = f(y_i) + e_i = \hat{y}_i + e_i,$$

donde $\hat{y}_i$ es el valor ajustado a la recta del valore observado y_i, y e_i es el error que cometemos y al que llamaremos **residuo**.

Una vez calculado el modelo, el valor de $\hat{y}$ queda determinado para cada x_i, pero el valor $e_i = y_i - \hat{y}_i$ no queda determinado, puede haber dos observaciones con el mismo x_i y distinto e_i. En este razonamiento se basará la hipótesis de independencia de los residuos.



2.3. Supuestos del modelo

Para cada x_i, valor fijo de X, se cumple la ecuación $y_i = \beta_0 + \beta_1 x_i + e_i$, donde β_0 y β_1 son constantes desconocidas. Las hipótesis básicas del modelo son:

1. Incorrelación de los residuos $Corr(e_i, e_j) = 0$. Cualquier par de errores e_i y e_j son independientes.
2. Media cero de los residuos $E(e_i) = 0$.
3. Varianza constante de los residuos $Var(e_i) = \sigma^2$.
4. Normalidad de los residuos $e_i \sim N(0, \sigma^2)$.

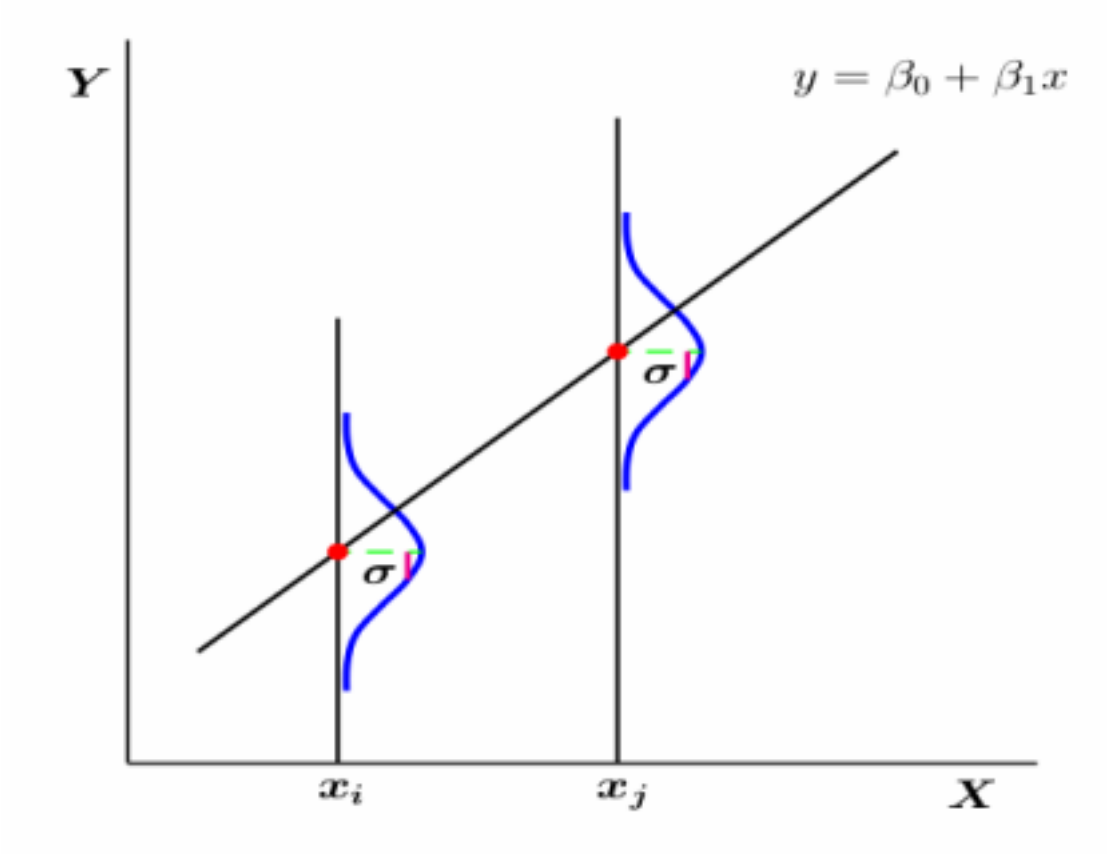


Figura 1: supuest

Como consecuencia:

- Cada valor x_i de la variable aleatoria X tiene distribución

$$(Y \mid X = x_i) \approx N(\beta_0 + \beta_1 x_i, \sigma^2).$$

- Las observaciones y_i de la variable Y son independientes.

Gráficamente, si las hipótesis del modelo son ciertas tenemos

2.3.1. Estimación de la recta de regresión. Método de mínimos cuadrados

Si nos encontrásemos en la situación ideal de que todos los puntos del diagrama de dispersión se encontraran en una línea recta no tendríamos que preocuparnos por encontrar la recta que mejor resume los puntos del diagrama, simplemente uniendo los puntos entre sí la obtendríamos.

Sin embargo si nos situamos en una situación más realista, en una nube de puntos es posible trazar muchas rectas diferentes, aunque obviamente, no todas ellas se ajustarán igualmente bien a la nube (SPSS, 2007). Se trata entonces de estimar la recta que el mejor represente el conjunto total de puntos.

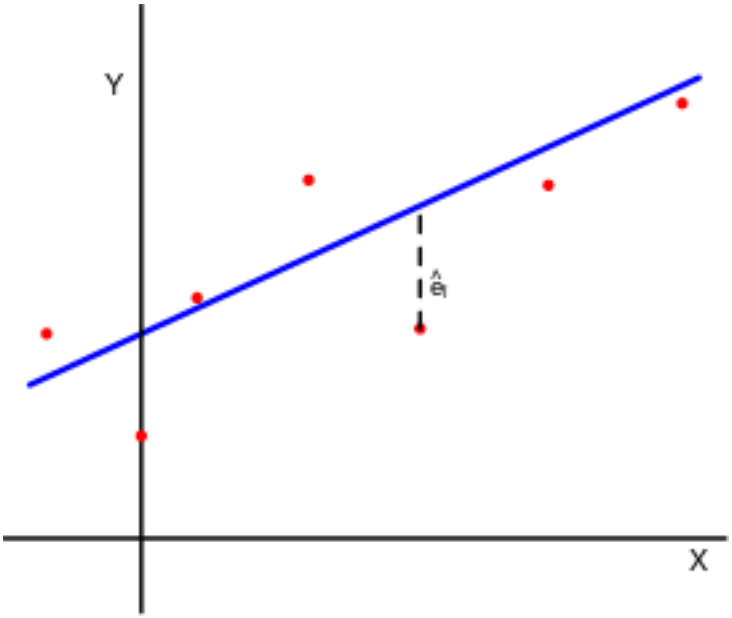
El procedimiento va a consistir en estimar los coeficientes de regresión β_0 y β_1 para obtener la recta

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

donde $\hat{y}$ denota el valor ajustado por la recta para el valor observado x.

Para estimar la ecuación de la recta de regresión podemos a utilizar el **criterio de mínimos cuadrados**, pues es el más empleado usualmente. Vamos a estudiarlo.

Siempre que ajustamos cualquier recta a un conjunto de datos existen pequeñas diferencias entre los valores estimados por la recta y los valores reales observados, así cada valor del modelo ajustado lleva asociado su error aleatorio $e_i = y_i - \hat{y}_i$.



Se nos podría ocurrir sumar todos los residuos para obtener así una estimación del error total, sin embargo, al sumar diferencias positivas y negativas estas tienden a cancelarse unas con otras. Para solucionar este problema decidimos elevar al cuadrado las diferencias antes de sumarlas (Ferrari & Head, 2010).

Por tanto, con el criterio de mínimos cuadrados estimamos los coeficientes de regresión, β_0 y β_1, haciendo mínima la suma de los cuadrados de los residuos, $SS_E = \sum_{i=1}^{n} e_i^2$.

$$SS_E = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \beta_0 - b_1 x_i)^2$$

Esto significa que, de todas las rectas posibles, existe una y sólo una que consigue que las distancias verticales entre cada punto y la recta sean mínimas (SPSS, 2007).

Las diferencias al cuadrado resultantes son un indicador de la capacidad de la recta ajustándose a los datos; si las diferencias al cuadrado son grandes la recta no es representativa de los datos, mientras que si son pequeñas la recta sí es representativa.

2.3.1.1. Consecuencias del criterio de mínimos cuadrados

- $\hat{\beta}_1 = r = \frac{Cov(X,Y)Sd(Y)}{Sd(X)}$.
- $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$.
- La suma de los residuos es cero
- La media de los valores observados Y_i coincide con la media de los valores ajustados $\overline{Y_i}$.
- La recta de regresión pasa por el punto $(\bar{x}, \bar{y})$.

- Los valores $\hat{\beta}_0$ y $\hat{\beta}_1$ son estimadores de β_0 y β_1.
- Las estimaciones de la respuesta para un valor $X = x$ se obtiene como

$$\overline{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

2.4. Ejemplo. Ajuste del modelo y proceso inferencial

Vamos a desarrollar esta sección mediante un ejemplo aplicado:

El presidente de personal de una multinacional está buscando si existe relación entre el salario de un trabajador y su porcentaje de absentismo. Éste dividió el intervalo de salarios en categorías y muestreó aleatoriamente a un grupo de trabajadores para determinar número de días que habían faltado en los últimos 3 años. ¿Es posible establecer un modelo que relacione la categoría y las ausencias?

2.4.1. Ajuste del modelo en R

Vamos a establecer el modelo que relaciona `Ausencias` con `Categoria`, pero antes de esto estudiaremos la normalidad de los datos y calcularemos la correlación entre categoría y ausencias, realizando además el correspondiente gráfico de dispersión.

```
datos <- read.table( "files/40A-william.csv", sep = ";", head = TRUE )
```

Empezamos estudiando la normalidad de la variable explicativa

```
shapiro.test( datos$Categoria )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$Categoria
## W = 0.93514, p-value = 0.2937
```

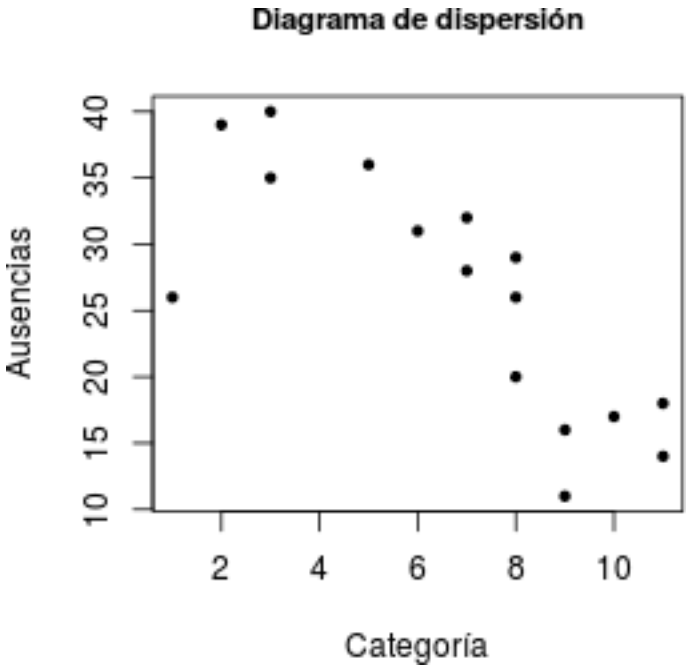
visto que los datos son normales, realizamos el análisis de correlación

```
cor.test( datos$Categoria, datos$Ausencias )
```

```
##
##  Pearson's product-moment correlation
##
## data:  datos$Categoria and datos$Ausencias
## t = -4.7432, df = 14, p-value = 0.0003144
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9219973 -0.4738285
## sample estimates:
##       cor
## -0.7851244
```

y representamos los puntos

```
plot( datos$Categoria, datos$Ausencias, pch = 20,
      xlab = "Categoría",  ylab = "Ausencias",
      main = "Diagrama de dispersión",cex.main = 0.95)
```



La correlación entre ambas variables es significativa con un p-valor menor a 0.05 y se trata de una relación inversa y alta (-0.7851), según crece la categoría disminuyen las ausencias.

Una vez visto que existe relación entre las variables pasamos a realizar el **ajuste del modelo**. Para ello usamos la función `lm( )` que toma la forma

```
lm( dependiente ~ predictora(s), data = dataFrame, na.action = .ªcción" )
```

donde `na.action` es opcional, puede ser útil si tenemos valores perdidos.

Creamos el objeto `modelAu` que contiene todos los resultados del ajuste.

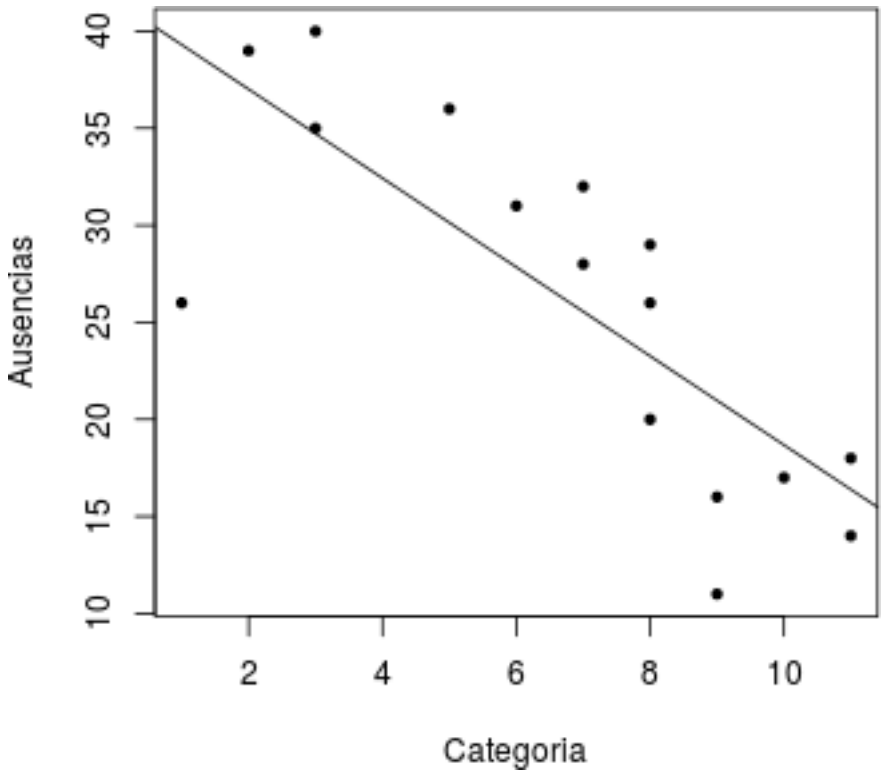
```
modelAu <- lm( Ausencias ~ Categoria, data = datos )
summary( modelAu )
```

```
##
## Call:
## lm(formula = Ausencias ~ Categoria, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.304  -2.603   1.802   3.687   6.448
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.5956     3.5795  11.621 1.41e-08 ***
## Categoria    -2.2919     0.4832  -4.743 0.000314 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.898 on 14 degrees of freedom
## Multiple R-squared:  0.6164, Adjusted R-squared:  0.589
## F-statistic:  22.5 on 1 and 14 DF,  p-value: 0.0003144
```

La parte 'Residuals' nos da la diferencia entre los valores experimentales y ajustados por el modelo. Las estimaciones de los coeficientes del modelo se proporcionan junto con el sus desviaciones estándar ('error estándar'), un t-valor y la probabilidad de la hipótesis nula de que los coeficientes tengan valor de cero. En este caso, por ejemplo, hay evidencia de que ambos coeficientes son significativamente diferentes de cero.

En la parte inferior de la tabla se encuentra la desviación sobre la recta regresión (error estándar s_r o residual), el coeficiente de correlación y el resultado del *test* F sobre la hipótesis nula de que los $\frac{MS_{reg}}{MS_{res}}$ es 1.

```
plot( datos$Categoria, datos$Ausencias, pch = 20,
      xlab = "Categoria", ylab = "Ausencias" )
abline( modelAu )
```



En primer lugar deseamos obtener los estimadores puntuales, errores estándar y p-valores asociados con cada coeficiente

```
summary( modelAu )$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 41.595638  3.5794561 11.620659 1.411089e-08
## Categoria   -2.291946  0.4832032 -4.743235 3.144361e-04
```

El resultado del ajuste es

```
(3.5795) (0.4832) Ausencias = 41.5956-2.2919 * Categoria
```

donde los valores entre paréntesis indican los errores estándar de cada coeficiente. Además, puesto que los p-valores asociados son inferiores a `0.05`, podemos concluir que:

1. En este caso no tiene sentido analizar el valor de la constante para `Categoría = 0`, ya que no pertenecería a la empresa, de ahí que el valor de Ausencias para `Categoría = 0` sea de `41.5956`, mayor que cualquiera de los datos de nuestro conjunto.
2. Existen evidencias estadísticas suficientes para considerar que hay una relación lineal entre Categoría y Ausencias. Dicha relación es negativa cuando aumenta la categoría laboral del individuo disminuyen las

ausencias. Además vemos que por cada grado que aumenta la categoría del trabajador, disminuyen las ausencias en `2,29` días por año.

3. El error estándar residual estimado (s) es de `5.898`. Este valor es muy importante, es un medidor de la calidad (precisión) del modelo. Además nos vamos a basar en él para calcular los intervalos de confianza para los coeficientes del modelo. Se calcula haciendo la raíz cuadrada de la media de la suma de cuadrados de los residuos (MS_R).

2.4.1.1. Intervalos de Confianza

Los intervalos de confianza (IC) complementan la información que proporcionan los contraste de hipótesis a la hora de expresar el grado de incertidumbre en nuestras estimaciones.

Obtenemos los correspondientes intervalos de confianza para cada parámetro del modelo con nivel significación al 95 %

```
confint( modelAu, level = 0.95 )
```

```
##                 2.5 %    97.5 %
## (Intercept) 33.918468 49.272807
## Categoria   -3.328314 -1.255579
```

como el intervalo no contiene al cero, podemos rechazar la hipótesis nula de que $H_0 : \beta_0 = \beta_1 = 0$.

Interpretamos los intervalos: con una probabilidad del 95 %, la ordenada en el origen del modelo, β_0, se encuentra en el intervalo `(33.92, 49.27)`, mientras que el efecto asociado con la *Categoria* se encuentra en el intervalo `(-3.32, -1.26)`.

2.5. Bondad de ajuste

Una vez realizado el ajuste, debemos verificar la eficiencia del modelo a la hora de explicar la variable dependiente, ya que aunque la recta sea la mejor disponible, ésta puede seguir siendo un ajuste terrible de los datos.

Las medidas fundamentales son el *error residual estimado*, el *test F* para la bondad de ajuste de la tabla ANOVA y el *coeficiente de determinación* R^2. Iremos explicándolas una a una pero antes vamos a hablar de la variabilidad del modelo de regresión.

La variabilidad del ajuste se puede descomponer como

`Variación total= variación explicada modelo + variación residual`, es decir,

$SS_T = SS_M + SS_R$, donde

- $SS_T = \sum(y - \bar{y})^2$ es la cantidad total de variabilidad existente al aplicar el modelo más básico, el modelo nulo (la media).
- $SS_R = \sum(y - \hat{y})^2$ representa el grado de imprecisión cuando se ha ajustado el mejor modelo a los datos.
- $SS_M = SS_T - SS_R$ muestra cómo mejora la predicción al usar el modelo de regresión en vez predecir con la media. Es la reducción de la imprecisión al ajustar el modelo de regresión a los datos.

Si SS_M es grande entonces el modelo de regresión es muy diferente de la media, lo que significa que se ha hecho una gran mejora a la hora de predecir la variable dependiente.

2.5.1. Coeficiente de determinación, R^2

El *coeficiente de determinación* que representa la proporción de mejora causada por el modelo, es decir, la proporción de variabilidad de la variable dependiente (Y) explicada por el modelo (SS_M), relativa a toda la variabilidad existente en el modelo (SS_T). Se puede escribir como

$$R^2 = \frac{SS_M}{SS_T}.$$

Para la regresión lineal simple, R^2 se corresponde con el cuadrado de la correlación entre Y y X.

Una variante de esta medida es la R^2 *ajustada* que se utiliza para la regresión múltiple, pues tiene en cuenta el número de grados de libertad. Vemos cómo se define.

Utilizando la fórmula de la variación total tenemos la siguiente igualdad

$$R^2 = \frac{SS_M}{SS_T} = 1 - \frac{SS_R}{SS_T}$$

y a partir de ella se define la R_a^2 dividiendo por los grados de libertad la introducción de variables innecesarias en el modelo

$$R_a^2 = 1 - \frac{SS_R/df_R}{SS_T/df_T}$$

Al añadir al modelo una variable que no aporte nada el df_R diminuye, por lo que el cociente $\frac{SS_R}{df_R}$ crecerá, haciendolo también $\frac{SS_R/df_R}{SS_T/df_T}$. Esto implica por tanto que el valor de la R_a^2 sea cada vez más pequeño.

Mientras que R^2 nos dice cuánta varianza de Y representa el modelo de regresión, la R_a^2 cuantifica la varianza de Y que representaría el modelo si este hubiera sido obtenido de la población donde hemos tomado la muestra. Si los valores de R^2 y R_a^2 están próximos significa que el modelo de regresión es bueno.

Estas medidas toman valores entre 0 y 1, y cuanto más se aproximen a 1 mejor será el ajuste, y por lo tanto, mayor la fiabilidad de las predicciones que con él realicemos.

Observación: ni R^2 ni R_a^2 son una indicación directa de la eficacia del modelo en la predicción de nuevas observaciones.

2.5.2. *Test F*

La última medida de ajuste que vamos a estudiar es el `test F`, una medida de cuánto ha mejorado el modelo prediciendo la variable dependiente con respecto al nivel de inexactitud del modelo. Se define como

$$F = \frac{MS_M}{MS_R},$$

donde MS son las medias de las sumas de cuadrados. Se definen como las sumas de cuadrados entre sus grados de libertad. Así tenemos

$$MS_M = \frac{SS_M}{df_M}$$

$$MS_R = \frac{SS_R}{df_R}$$

Un buen modelo debe tener un valor F grande (mayor que 1) ya que el numerador, la mejora en la predicción del modelo, será mayor que denominador, la diferencia entre el modelo y los datos observados.

Otra medida importante que se obtiene a partir de la suma de cuadrados de los residuos es el error estándar que se define como

$$SE_R = \sqrt{MS_R}.$$

Vamos a aplicar todo esto en `R` continuando con el ejemplo anterior.

2.5.3. Tabla ANOVA

Volvemos al ejemplo de las categorías y las ausencias. Obtenemos la correspondiente tabla ANOVA donde vemos la descomposición de la variabilidad del modelo

```
anova( modelAu )
```

```
## Analysis of Variance Table
##
## Response: Ausencias
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Categoria  1 782.70  782.70  22.498 0.0003144 ***
## Residuals 14 487.05   34.79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observamos que la variabilidad explicada por el modelo, `SSM=782.70`, es superior a la que queda por explicar (residuos), `SSR=487.05` y el estadístico `F=22.5`, mayor que 1. Además, volviendo a ver el resumen del modelo

```
## F-statistic:  22.5 on 1 and 14 DF,  p-value: 0.0003144
```

tenemos que el p-valor asociado con el estadístico F es inferior a `0.05`.

La conclusión es que hay evidencias suficientes para poder rechazar la hipótesis nula, $F = 1$ y por tanto, resulta posible establecer un modelo de regresión lineal para explicar el comportamiento de las ausencias en función de la categoría del empleado.

2.5.3.1. Coeficiente de determinación

En el `modelAu` el valor de R^2 es `Multiple R-squared: 0.6164`, alrededor del 62 % de la variabilidad de *Ausencias* es explicada por la recta ajustada.

2.6. Análisis de los parámetros del modelo

El test ANOVA significativo nos dice si el modelo tiene, en general, un grado de predicción significativamente bueno para la variable resultado, pero no nos dice nada sobre la contribución individual del modelo. Para encontrar los parámetros del modelo y su significación tenemos que volver a la parte `Coefficients` en el resumen del modelo.

```
summary( modelAu )$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 41.595638  3.5794561 11.620659 1.411089e-08
## Categoria   -2.291946  0.4832032 -4.743235 3.144361e-04
```

Observando la tabla vemos que $\beta_0 = 41{,}6$ (`intercept`) que podemos interpretar como que si no hubiera categorías ($X = 0$) el modelo predice que en la empresa habría un 41.6 % de ausencias, aunque en este caso no tiene sentido.

Por otro lado, β_1 es la pendiente de la recta y representa el cambio en la variable dependiente (ausencias) asociado al cambio de una unidad en la variable predictora. Si nuestra variable predictora incrementa una unidad, nuestro modelo predice que las ausencias se reducirán en 2,3, pues en este caso $\beta_1 = -2{,}2919$. Por tanto, la ecuación del modelo queda $Y = 41{,}6 - 2{,}3X$.

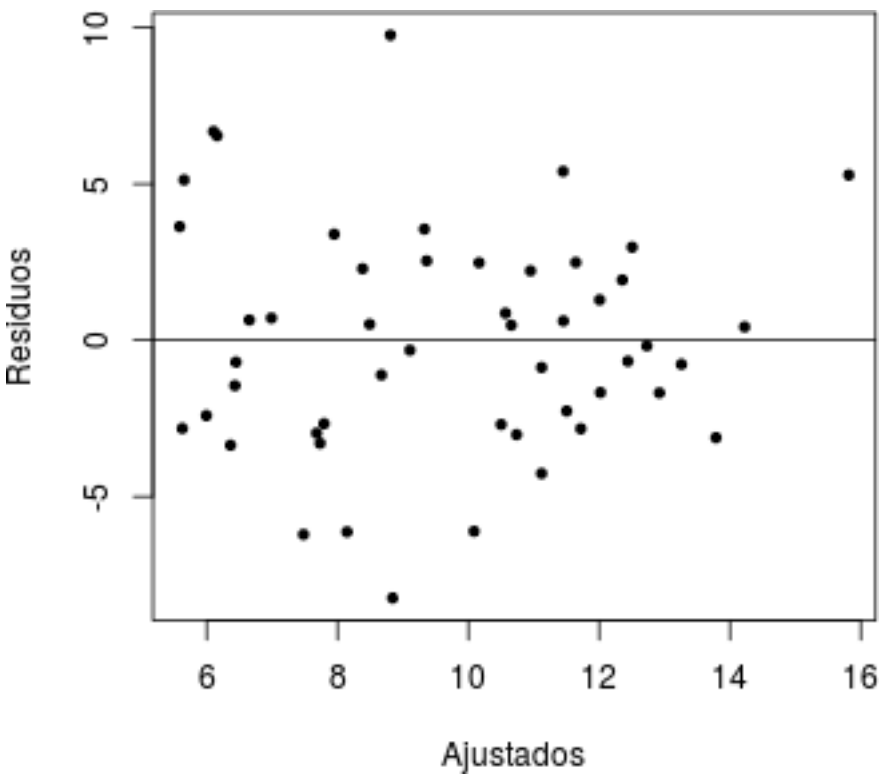
2.7. Diagnóstico del modelo

En este apartado hemos hecho uso tanto de J.Faraway (2009) como de Sánchez (2011) para el desarrollo del mismo.

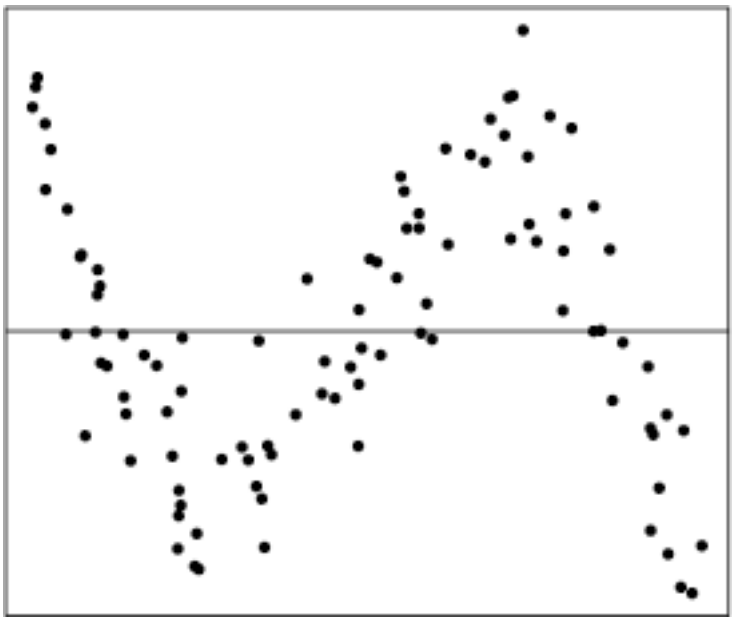
Una vez que tenemos el modelo ajustado procedemos con su diagnóstico, que se realiza a través del análisis de los residuos, e_i.

- Las hipótesis de linealidad, homocedasticidad e independencia se contrastan a través de un análisis gráfico que enfrenta los valores de los residuos, e_i, con los valores ajustados $\hat{x}_i$.
- Las hipótesis de media cero, varianza constante, incorrelación y normalidad la comprobamos analíticamente.

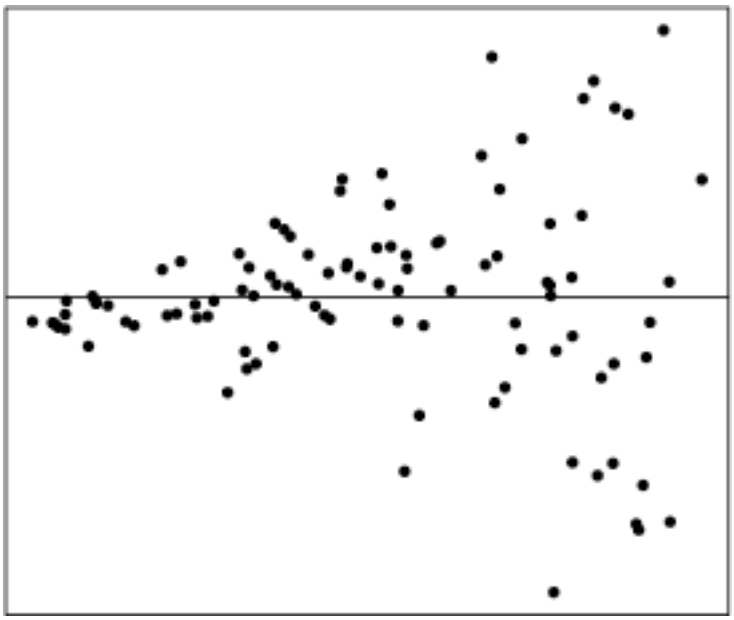
Comenzaremos con **el análisis gráfico**. Los residuos deberían formar una nube de puntos sin estructura y con, aproximadamente, la misma variabilidad por todas las zonas como se muestra en el gráfico.



En los siguientes gráficos no se cumplen las hipótesis. Los residuos de esta primera gráfica muestran una estructura que sugiere una relación no lineal entre las variables



y los de la siguiente sugieren la *ausencia de homocedasticidad.*



Continuamos ahora realizando el **diagnóstico analítico**. El primer paso es obtener los residuos, valores ajustados y estadísticos del modelo analizado para poder así estudiar si se cumplen los supuestos del mismo.

Obtención de residuos, valores ajustados y estadísticos necesarios

Para ello, añadimos los correspondientes resultados a nuestros datos a través del siguiente código:

```
datos$fitted.modelAu    <- fitted( modelAu )
datos$residuals.modelAu <- residuals( modelAu )
datos$rstudent.modelAu  <- rstudent( modelAu )
```

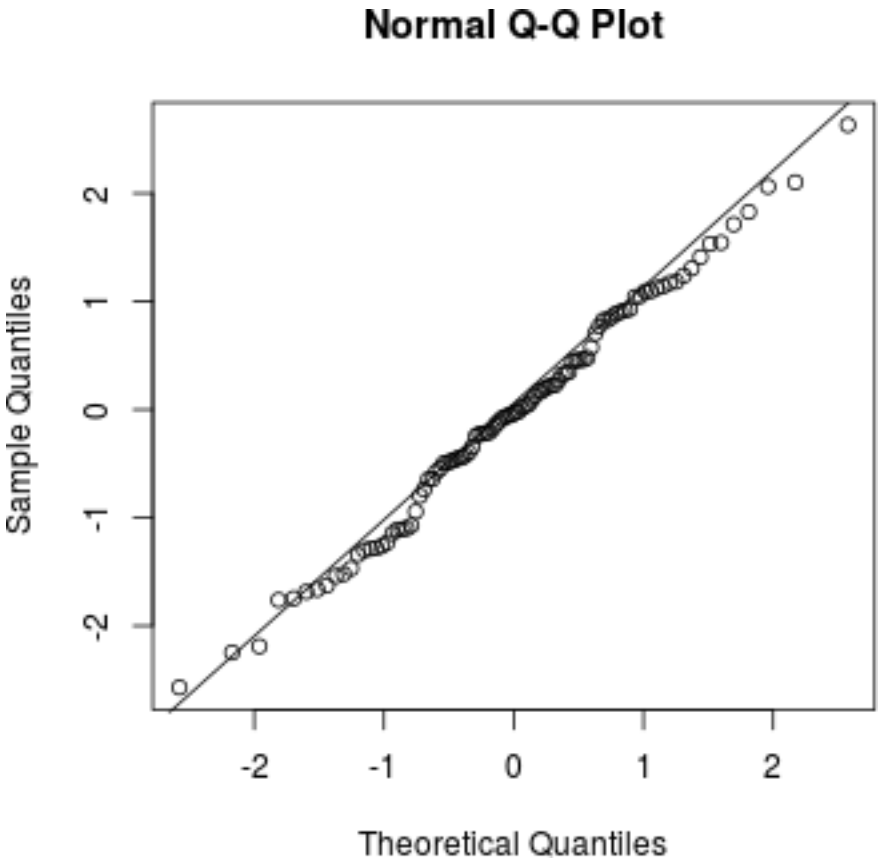
El resultado es la creación de las siguientes variables:

- `fitted.modelAu`: valores ajustados (valores de la variable respuesta) para las observaciones originales de la predictora.
- `residuals.modelAu`: residuos del modelo, esto es, diferencia entre valor observado de la respuesta y valor ajustado por el modelo.
- `rstudent.modelAu`: residuos estudentizados del modelo ajustado.
- `obsNumber`: número de la observación en el orden en que has sido recogidas.

Vamos a utilizar todas estas variables para estudiar si nuestro modelo cumple las hipótesis.

2.7.1. Test de normalidad (test de Kolmogorov-Smirnov)

Empezamos el análisis con un gráfico `qqplot`, que enfrenta los valores reales a los valores que obtendríamos si la distribución fuera normal. Si los datos reales se distribuyen normalmente, estos tendrán la misma distribución que los valores esperados y en el gráfico `qqplot` obtendremos una linea recta en la diagonal

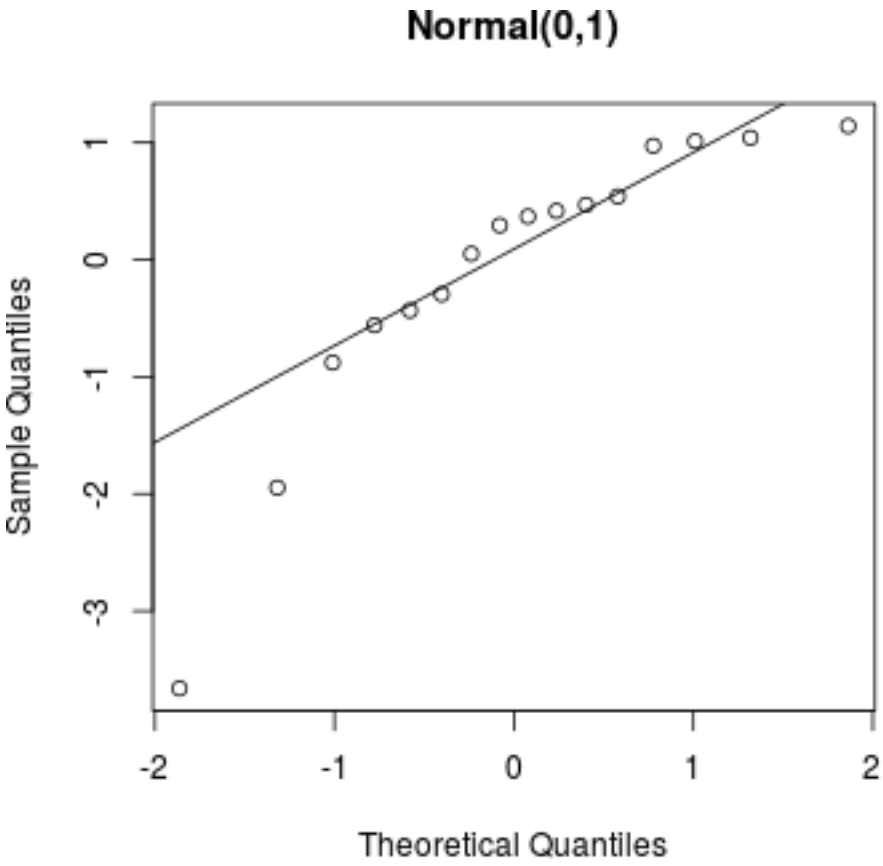


Analizamos nuestros residuos

```
shapiro.test( datos$rstudent.modelAu )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$rstudent.modelAu
## W = 0.82712, p-value = 0.006388
```

```
qqnorm( datos$rstudent.modelAu, main = "Normal(0,1)" )
qqline( datos$rstudent.modelAu )
```



Tenemos problemas con la condición de normalidad de los errores ya que obtenemos un p-valor para el contraste de `0.0063`, inferior a `0.05`. Como en el gráfico `qqplot` los puntos no se sitúan en la diagonal, efectivamente vemos que los datos no son normales.

2.7.2. Homogeneidad de varianzas

```
library( lmtest )
bptest( modelAu )
```

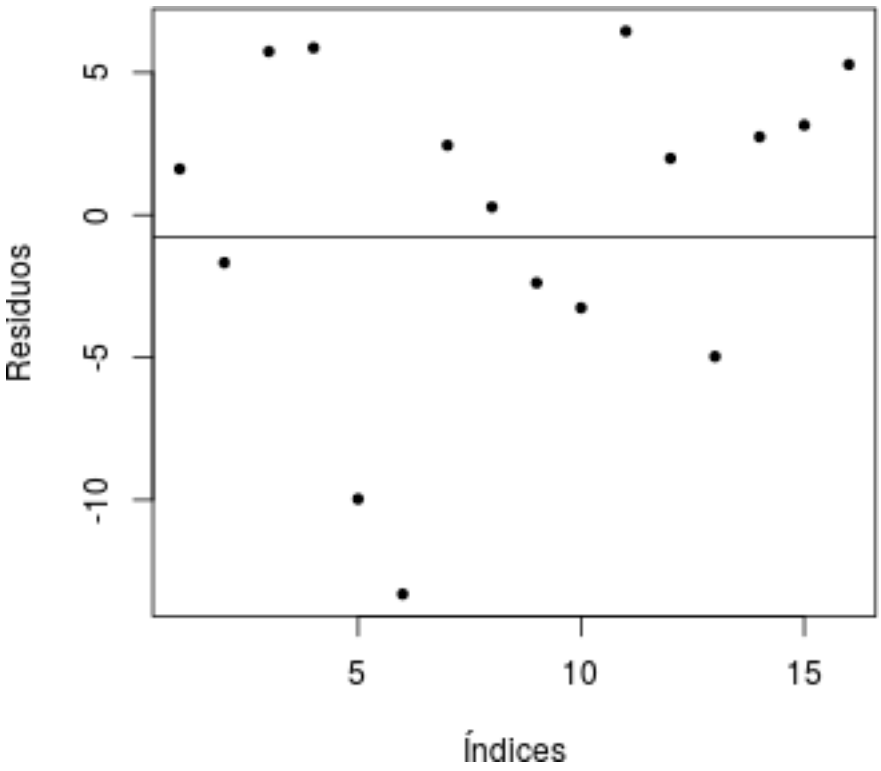
```
##
##  studentized Breusch-Pagan test
##
## data:  modelAu
## BP = 2.1918, df = 1, p-value = 0.1387
```

Existe homogeneidad pues la significación es mayor de `0.05`, la varianza es constante a lo largo de la muestra.

2.7.3. Autocorrelación (test de Durbin-Watson)

Hemos asumido que los residuos son incorrelados, vamos a comprobarlo.

```
plot( datos$residuals.modelAu, pch = 20, ylab = "Residuos", xlab = "Índices" )
abline( h = cor( datos$Ausencias, datos$Categoria ) )
```



Si hubiera una correlación seria, veríamos picos más largos de residuos por encima y por debajo de la línea de correlación. A menos que estos efectos sean fuertes, puede ser difícil de detectar la autocorrelación, por ello realizamos el *contraste de Durbin-Watson*.

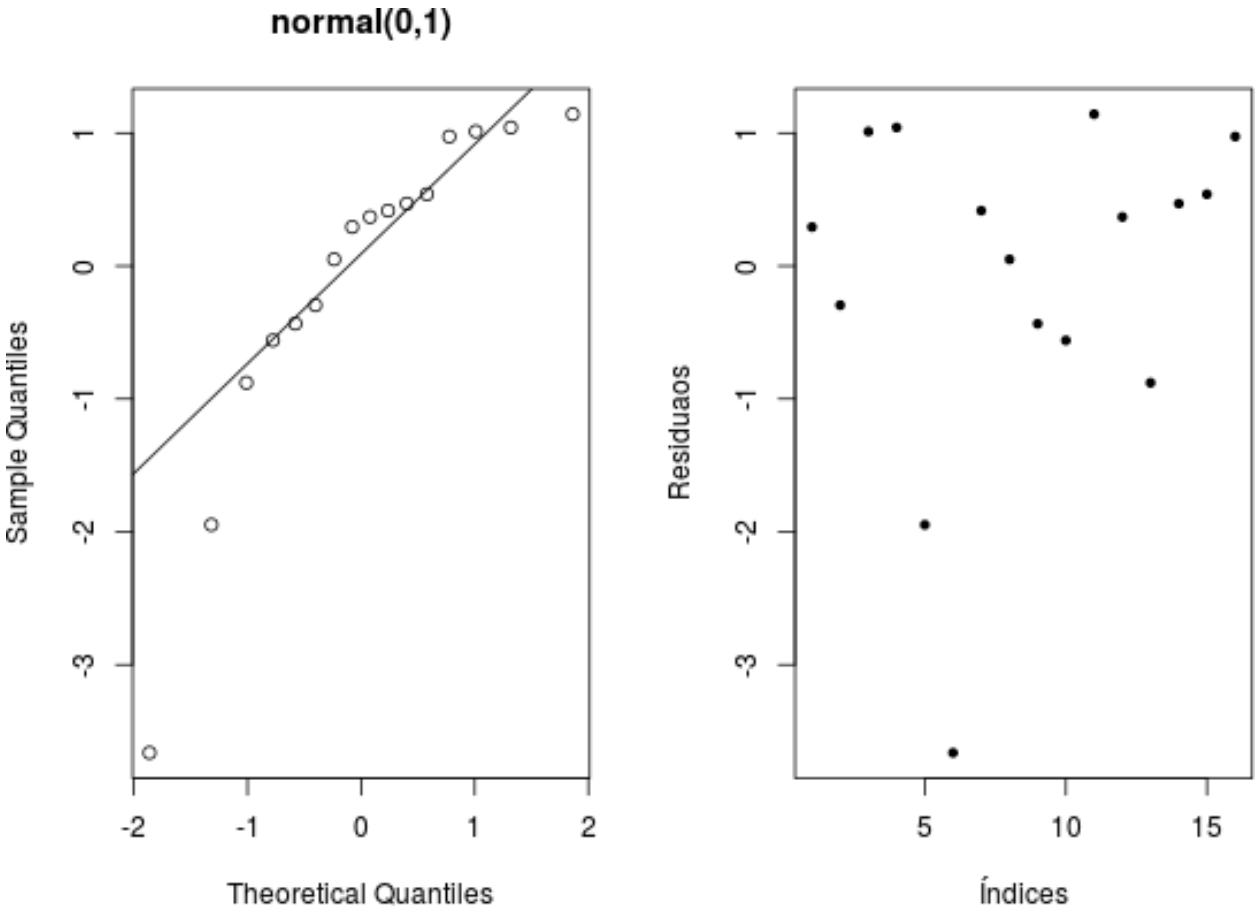
```
dwtest( Ausencias ~ Categoria, alternative = "two.sided", data = datos )
```

```
##
##  Durbin-Watson test
##
## data:  Ausencias ~ Categoria
## DW = 1.6732, p-value = 0.4935
## alternative hypothesis: true autocorrelation is not 0
```

En el contraste de autocorrelación también aceptamos la hipótesis nula de que no existe correlación entre los residuos con un p-valor superior a `0.05`.

Una vez comprobado el resto de supuestos del modelo, vamos a intentar solucionar el problema de normalidad. Lo primero que hacemos es representar de nuevo los datos en un QQ-plot y un diagrama de dispersión para detectar posibles perturbaciones.

```
par( mfrow = c( 1, 2 ) )
qqnorm( datos$rstudent.modelAu, main = "normal(0,1)" )
qqline( datos$rstudent.modelAu )
plot(   datos$rstudent.modelAu, pch = 20, ylab = "Residuaos", xlab = "Índices"  )
```



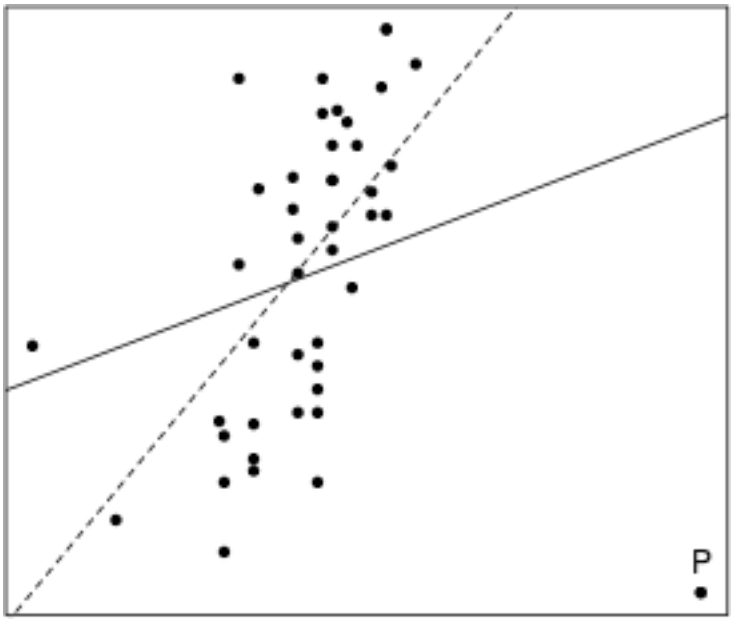
Si observamos de nuevo el gráfico vemos que hay un punto que está totalmente fuera de lugar, lo que parece en principio un valor atípico. Vamos a realizar un test de valores atípicos (Bonferroni).

2.7.4. Valores atípicos

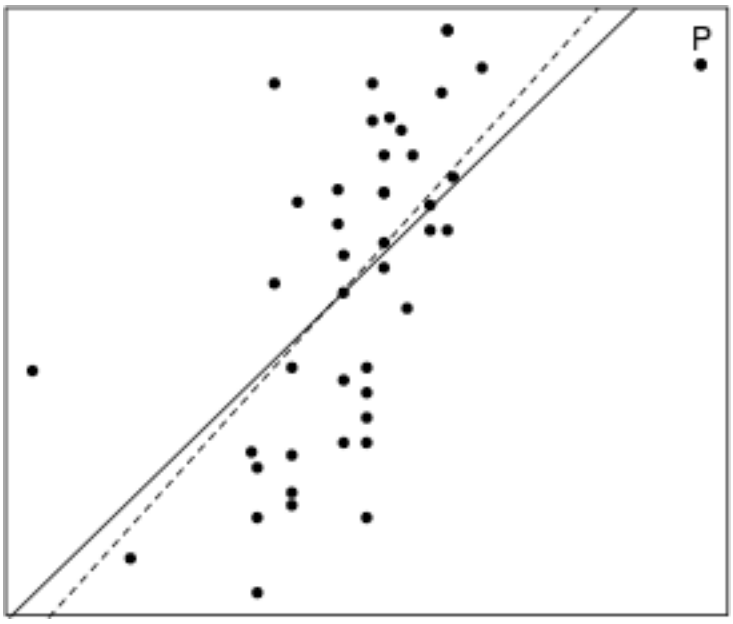
Un valor atípico es aquel que difiere sustancialmente de la tendencia general de los datos. Estos valores atípicos pueden perjudicar el modelo ya que afectan a los coeficientes de regresión estimados. Veamos gráficamente cómo pueden influir a la recta de regresión (Sánchez, 2011).

En los gráficos la linea discontinua representa la recta de regresión calculada sin considerar el punto P.

Para este primer gráfico tenemos que el punto P sí es influyente pues modifica sustancialmente la recta de regresión.



mientras que en el segundo gráfico el punto P apenas influye en el modelo.



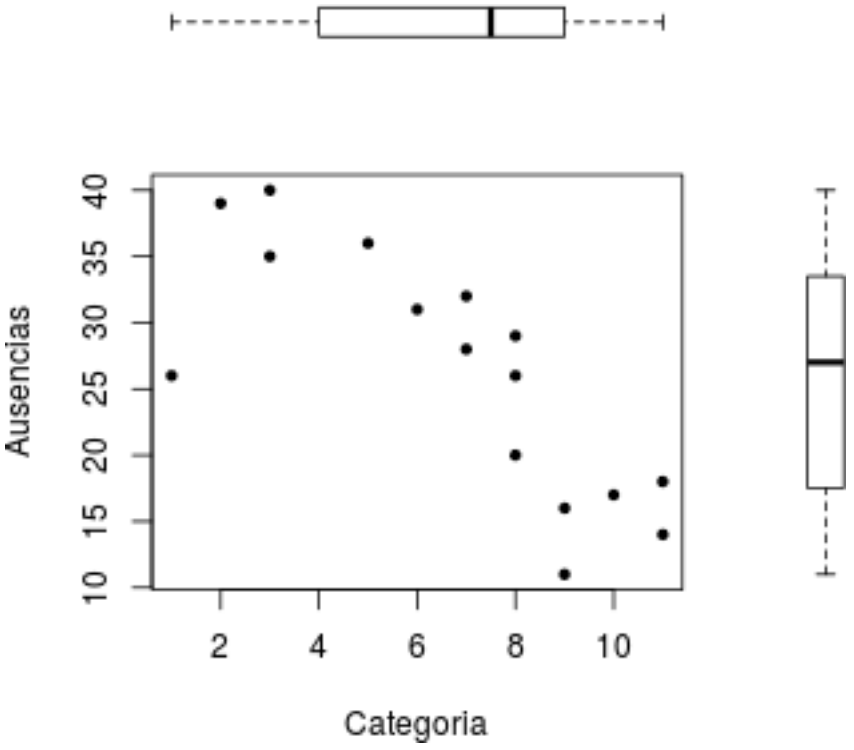
En el caso de observar valores atípicos los pasos a seguir son:

1. Descartar que sea un error.
2. Analizar si es un caso influyente.
3. En caso de ser influyente calcular las rectas de regresión incluyéndolo y excluyéndolo, y elegir la que mejor se adapte al problema y a las observaciones futuras.

Para el estudio de los valores atípico vamos a usar los **residuos estandarizados**, los residuos divididos por una estimación de su error estándar. Existen unas reglas generales:

1) Residuos estandarizados con un valor absoluto mayor de `3.29` (redondearemos a `3`) son causa de preocupación ya que es improbable que en una muestra media un valor tan grande ocurra por azar.
2) Si más del `1%` de los valores muestrales tienen residuos estandarizados con un valor absoluto mayor de `2.58` (podemos decir `2.5`) hay evidencias de que el nivel de error en nuestro modelo es inaceptable (ajuste pobre del modelo a los datos).
3) Si más del `5%` de los casos tienen residuos estandarizados con un valor absoluto mayor de `1.96` (usamos `2` por conveniencia) entonces vuelven a haber indicios de que el modelo es una pobre representación de los datos reales.

Vamos a hacer un **estudio de valores atípicos de nuestro modelo**. Empezamos con un gráfico en el que representamos el diagrama de puntos y un *boxplot* para cada una de las variables.



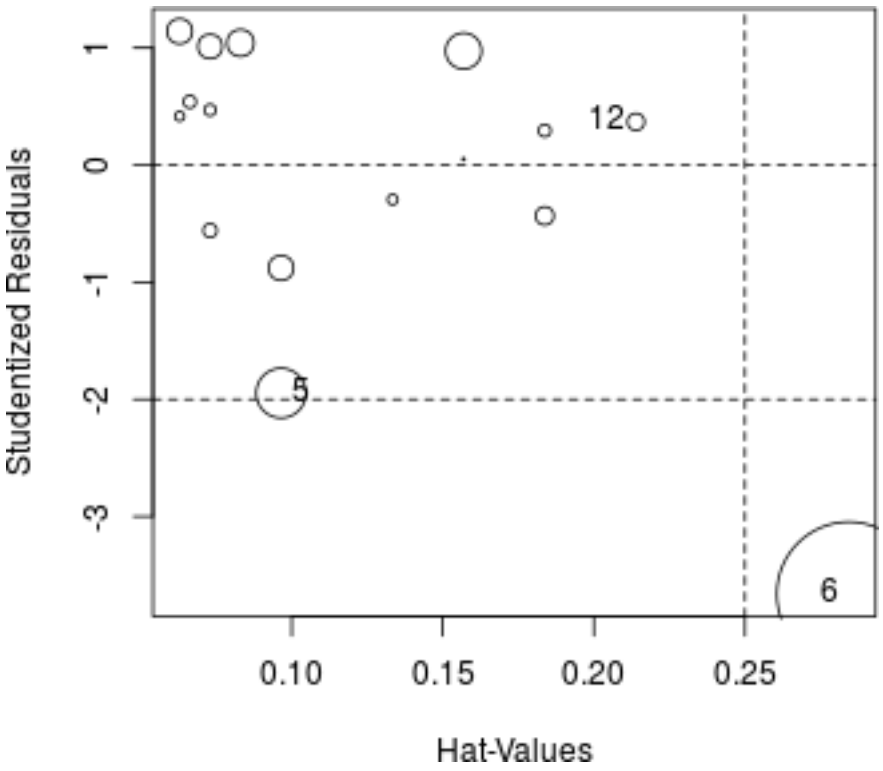
En la variable `Categoria` vemos que la mediana no está centrada en la media, los datos no son uniformes. Con la variable `Ausencias` ocurre lo mismo. Lo bueno es que en ninguno de los dos casos se aprecian valores atípicos (Kabacoff, 2014).

Contunuamos con un análiis más analítico:

```
library( car )
outlierTest( modelAu, cutoff = 0.05, n.max = 10, order = TRUE )
```

```
##    rstudent unadjusted p-value Bonferonni p
## 6 -3.662286          0.0028692     0.045908
```

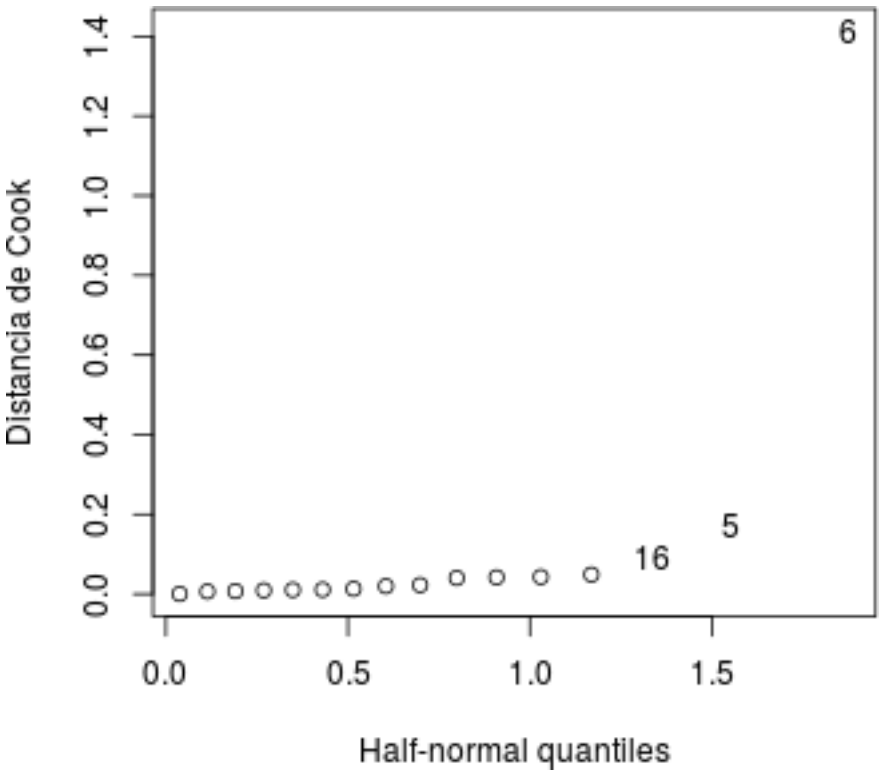
```
influencePlot( modelAu, id.n = 2 )
```




```
##       StudRes        Hat      CookD
## 5  -1.9471535 0.09647651 0.16876931
## 6  -3.6622863 0.28439597 1.41268847
## 12  0.3682807 0.21392617 0.01967006
```

El test y el gráfico nos indican que la observación número 6 es un valor atípico. Las observaciones 5 y 12 que vemos en el gráfico son medidas influyentes para ver si llegan a ser atípicos dibujamos el gráfico de las distancias de *Cook* (J.Faraway, 2009).

```
cook <- cooks.distance( modelAu )
labels <- rownames( datos )
library( faraway )
halfnorm( cook, 3, labs = labels, ylab = "Distancia de Cook" )
```



Se confirma que el valor 6 es un atípico, mientras que los puntos 16y 5 no lo son por ser su distancia de Cook menor que 1.

Aunque nunca es recomendable suprimir datos salvo estar seguros de que ha sido una mala medición o cualquier otro tipo de error, en este caso y en vista de lo obtenido, decidimos eliminar dicho dato.

```
datos <- datos[ -c( 6 ), ]
head( datos )
```

```
##   Categoria Ausencias fitted.modelAu residuals.modelAu rstudent.modelAu
## 1        11        18       16.38423          1.615772        0.2931414
## 2        10        17       18.67617         -1.676174       -0.2951496
## 3         8        29       23.26007          5.739933        1.0115842
## 4         5        36       30.13591          5.864094        1.0413881
## 5         9        11       20.96812         -9.968121       -1.9471535
## 7         7        28       25.55201          2.447987        0.4158870
```

NOTA: Cuidado con la eliminación de datos. ¡El diagnóstico del modelo es para fines predictivos! Para obtener un buen modelo aunque sin fines predictivos, únicamente debemos evitar el problema de la multicolinealidad.

Tras eliminar el valor atípico de la base de datos volvemos a realizar el test de *Shapiro-Wilk* para comprobar si se cumple ahora la condición de normalidad

```
shapiro.test( datos$rstudent.modelAu )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$rstudent.modelAu
## W = 0.91325, p-value = 0.1519
```

como este nuevo p-valor es mayor que 0.05 ahora sí existe normalidad en los datos. Una vez solucionados los problemas de diagnóstico pasamos a la fase de predicción.

2.8. Predicción

Tenemos un modelo de regresión con la capacidad de relacionar la variable predictora y la variable dependiente. Podemos utilizarlo ahora para predecir eventos futuros de la variable dependiente a través de nuevos valores de la variable predictora.

Para ello debe verificarse alguna de las siguientes condiciones

- el valor de la predictora está dentro del rango de la variable original.
- si el valor de la predictora está fuera del rango de la original, debemos asegurar que los valores futuros mantendrán el modelo lineal propuesto.

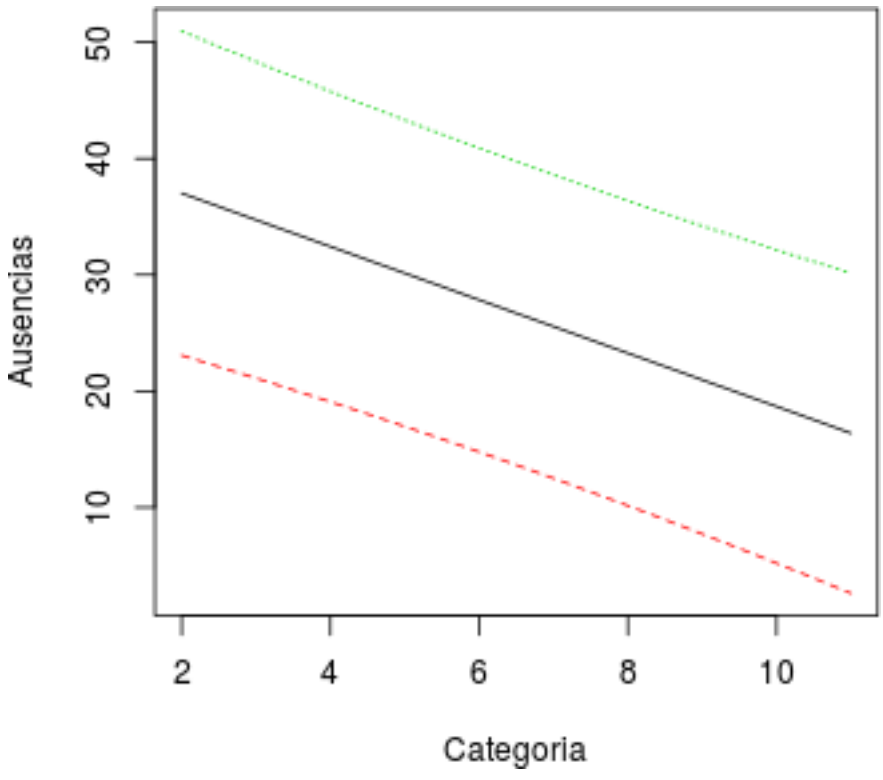
2.8.1. Predicción de nuevas observaciones

```
x0      <- seq( min( datos$Categoria ), max( datos$Categoria ), length = 15 )
dfp     <- data.frame( Categoria = x0 )
pred.ip <- predict( modelAu, dfp, interval = "prediction",
                    se.fit = TRUE, data = datos )
head( pred.ip$fit )
```

```
##        fit      lwr      upr
## 1 37.01174 23.07366 50.94983
## 2 35.53835 21.82140 49.25530
## 3 34.06496 20.53991 47.59000
## 4 32.59156 19.22793 45.95520
## 5 31.11817 17.88433 44.35201
## 6 29.64477 16.50819 42.78136
```

Dibujamos las bandas de predicción, que reflejan la incertidumbre sobre futuras observaciones:

```
matplot( x0, pred.ip$fit, type = "l", xlab = "Categoria", ylab = "Ausencias" )
```



Supongamos que no tuviéramos los datos en la escala original de la variable dependiente, sino que los hemos transformado mediante alguna función. En ese caso, para obtener las predicciones originales basta con deshacer la correspondiente transformación. Si hubiésemos transformado, por ejemplo, los datos originales mediante `log( )`, el código para obtener las predicciones sería

```
newpred <- exp( pred.ip$fit )
head( newpred )
```

2.8.2. Intervalos de confianza para los predictores

Dado un nuevo conjunto de predictores, x_0, debemos evaluar la incertidumbre en esta predicción. Para tomar decisiones racionales necesitamos algo más que puntos estimados. Si la predicción tiene intervalo de confianza ancho entonces entonces los resultados estarán lejos de la estimación puntual.

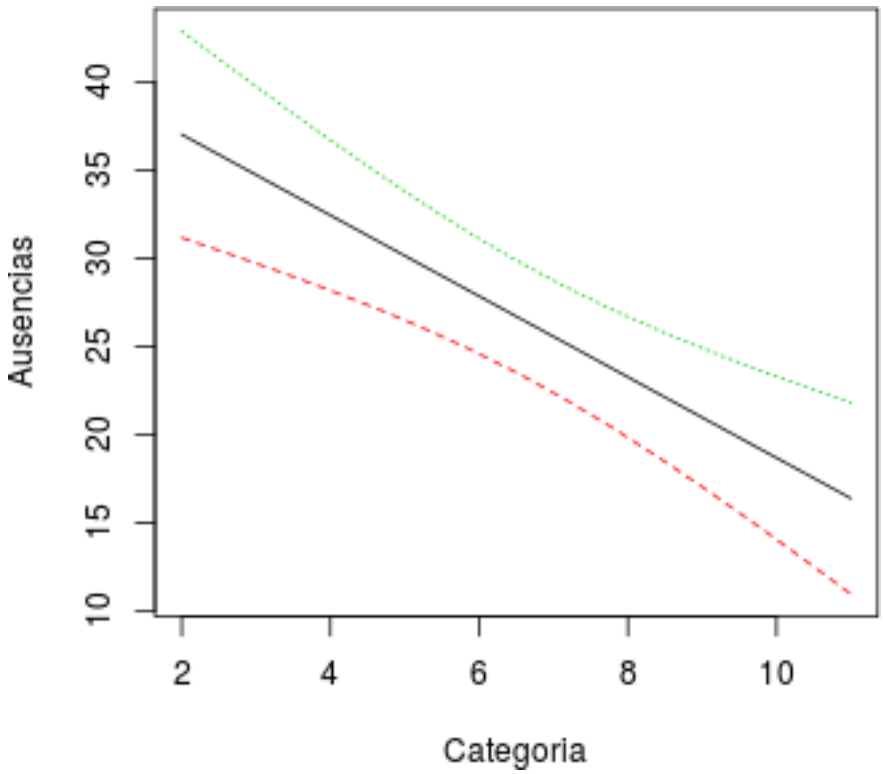
Las bandas de confianza reflejan la incertidumbre en la línea de regresión (lo bien que la línea está calculada).

```
pred.ic <- predict( modelAu, dfp, interval = "confidence",
                    se.fit = TRUE, data = datos )
head( pred.ic$fit )
```

```
##        fit      lwr      upr
## 1 37.01174 31.16063 42.86286
## 2 35.53835 30.23552 40.84118
## 3 34.06496 29.28037 38.84954
## 4 32.59156 28.28434 36.89878
## 5 31.11817 27.23232 35.00402
## 6 29.64477 26.10426 33.18529
```

Dibujamos las bandas de confianza, que además reflejan la incertidumbre sobre futuras observaciones:

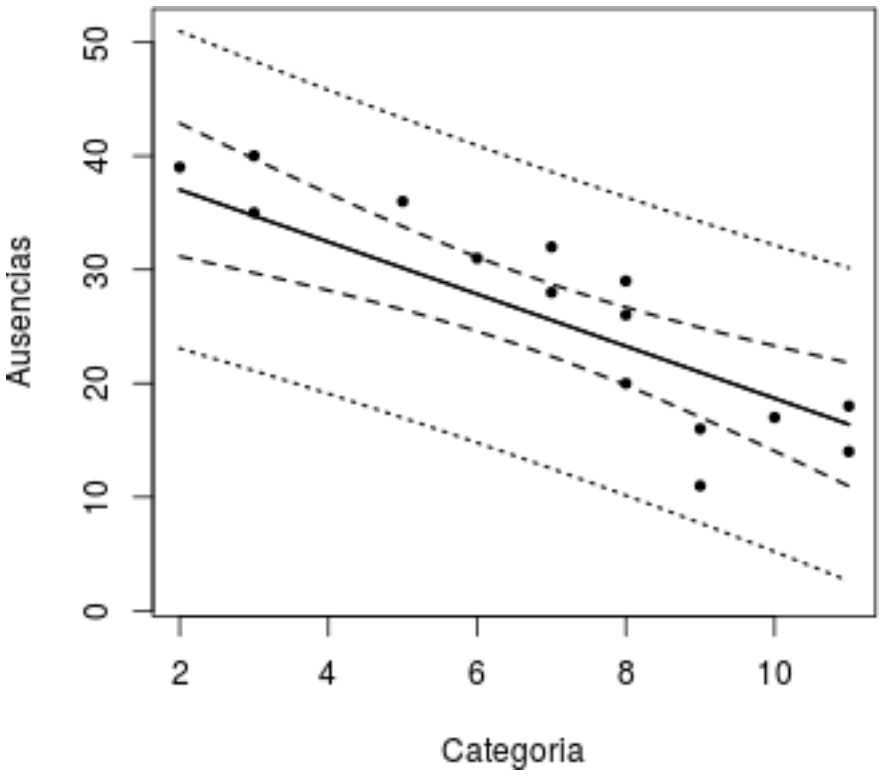
```
library( graphics )
matplot( x0, pred.ic$fit, type = "l", xlab = "Categoria", ylab = "Ausencias" )
```



Por último podemos hacer un gráfico con la nube de puntos y los dos bandas, la de confianza y la de predicción (Ferrari & Head, 2010).

```
plot( datos$Categoria, datos$Ausencias, pch = 20,
      ylim = range ( datos$Categoria, pred.ip, na.rm = TRUE ),
      xlab = "Categoria", ylab = "Ausencias" )

# Añadimos las bandas
matlines ( dfp$Categoria, pred.ic$fit, lty = c( 1 ,2 ,2 ), lwd = 1.5, col = 1 )
matlines ( dfp$Categoria, pred.ip$fit, lty = c( 1 ,3 ,3 ), lwd = 1.5, col = 1 )
```



2.9. Resumen de código en R

```
#Leer los datos de un fichero .csv
df <- read.table( "files/40A-file.csv", sep = ";", head = TRUE )

# Correlación
## Gráfico de dispersión (nube de puntos)
plot( df$var1, df$var2 )

## Normalidad de las variables explicativas
shapiro.test( df$var2 )

## Calculamos la correlación entre las varibales a estudiar
cor( df$var1, df$var2 )

### Además de calcularla vemos su significación con un test
cor.test( df$var1, df$var2, method = "pearson" )

## Calculamos la correlación de una matriz de variables
ndf <- data.frame( df$var1, df$var2, df$var3, df$var4 )
cor( ndf, use = "everything", method = "pearson" )

## Coeficiente de determinación (R^2)
cor( ndf, use = "everything" ) ^ 2

## Hacemos el test de correlaciones para la matriz (reg. multiple)
library( "psych" )
corr.test( ndf, use = "complete", method = "pearson" )
```

```r
# Modelo de regresión simple

## Creamos el modelo de regresión
model <- lm( var1 ~ var2, data = df )

## Representamos gráficamente el ajuste
plot( df$var1, df$var2, xlab = "var1", ylab = "var2" )
abline( model )

## Resumen del modelo
summary( model )

## Estudiamos los coeficientes del modelo
summary( model )$coefficients

### Intervalos de confianza para los coeficientes
confint( model, level = 0.95 )

## tabla ANOVA (ajuste del modelo)
anova( model )


# Diagnóstico del modelo (comprobar supuestos)

## Obtención de los residuos

df$fitted.model   <- fitted( model )
dfresiduals.model <- residuals( model )
df$rstudent.model <- rstudent(  model )

### Normalidad
shapiro.test( df$rstudent.model )
qqnorm( df$rstudent.model, main = "Normal(0,1)" )
qqline( df$rstudent.model )

### Homogeneidad de varianzas
library( lmtest )
bptest( model )

### Autocorrelación
plot( df$residuals.model, ylab = "Residuaos", xlab = "Índices" )
abline( h = cor( df$var1, df$var2 ) )
dwtest( var1 ~ var2, alternative = "two.sided", data = df )

### Valores atípicos
library( car )
outlierTest( model, cutoff = 0.05, n.max = 10, order = TRUE )

## Predicción
x0   <- seq( min( df$var2 ), max( df$var2 ), length = 15 )
pred <- predict( model, data.frame( var2 = x0 ), interval = "prediction",
                 se.fit = TRUE, data = df )
```

```r
head( pred )


### Intervalo de confianzar para los predictores
ic <- predict( model, data.frame( var2 = x0 ), interval = "confidence",
               se.fit = TRUE, data = df )
head( ic )

#### banda de confianza
#library( graphics )
matplot( x0, ic$fit, type = "l", xlab = "var2", ylab = "var1" )
```

3. Regresión lineal múltiple

3.1. Introducción

En la regresión lineal simple predecíamos la variable resultado Y a partir de los valores de X, usando la ecuación de una linea recta. Con los valores que habíamos ido obteniendo de X e Y calculábamos los parámetros de la ecuación ajustando el modelo a los datos mediante el método de mínimos cuadrados. La regresión múltiple es una extensión lógica de esto a situaciones en las que hay más de una variable predictora. La nueva ecuación será

$$Y_i = (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_n X_{ni}) + e_i.$$

Básicamente se trata de la misma ecuación que para la regresión simple excepto porque hemos incluido predictores extra. Cada predictor tienen asociado su propio coeficiente y predecimos la variable dependiente a partir de una combinación de todas las variables más un residuo, e_i, la diferencia entre el valor ajustado y observado de Y en la i-ésima observación.

Los coeficientes de regresión se pueden interpretar como:

- β_i el efecto medio (positivo o negativo) sobre la variable dependiente al aumentar en una unidad el valor de la predictora $X_i, i = 1, \cdots, k$.
- β_0 el valor medio de la variable dependiente cuando las predictoras son cero.

3.2. Ejemplo de un modelo de regresión lineal múltiple

Para entender el modelo de regresión lineal múltiple vamos a usar un ejemplo de una empresa de fabricación y reparto de pizzas. Utilizaremos la base de datos pizza.rda.

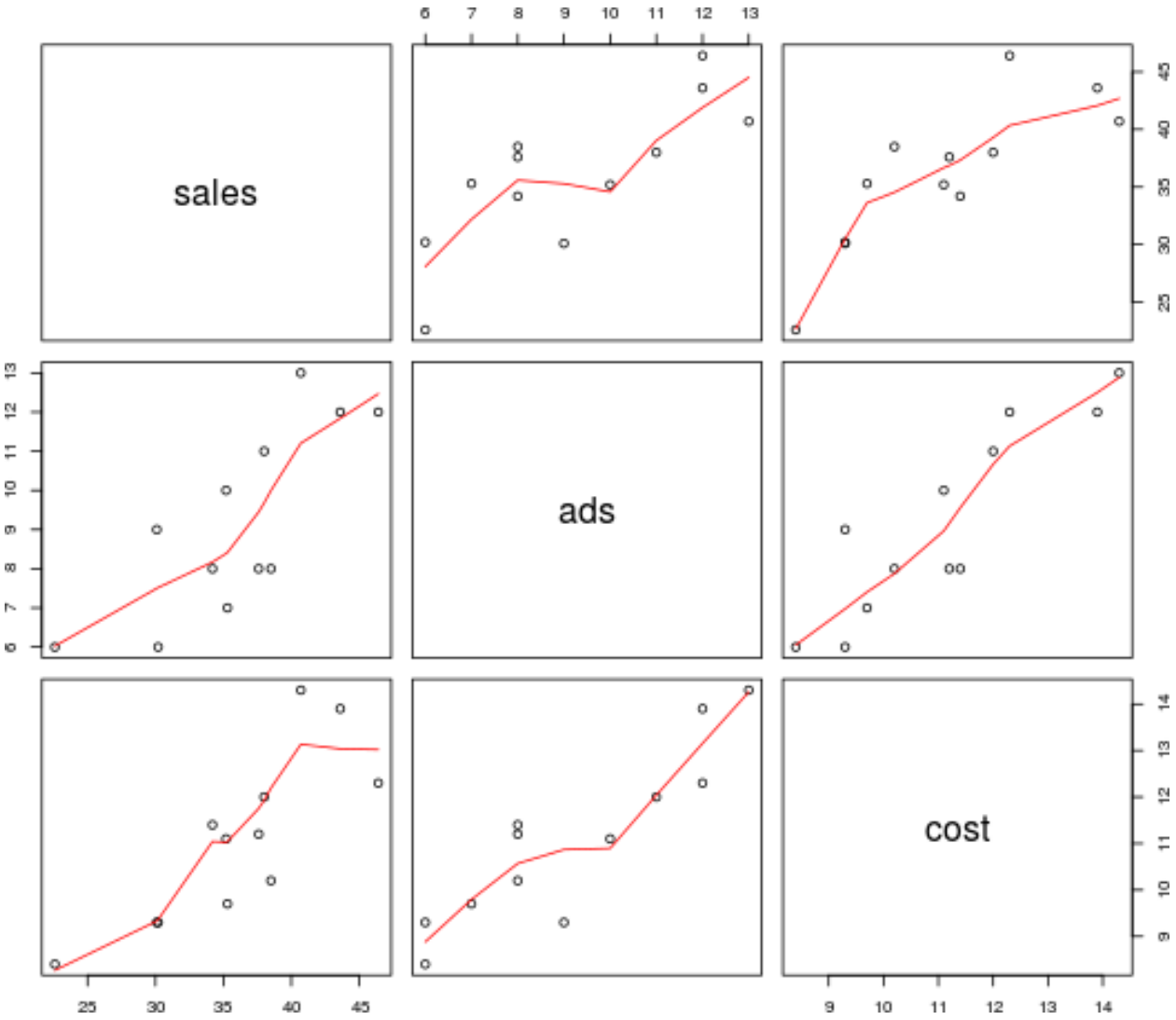
Planteamos el modelo $sales \sim ads + cost$ que tendrá ecuación es

$$sales = \beta_0 + \beta_1 ads + \beta_2 cost + e.$$

3.2.1. Análisis de correlación

Comenzamos representando los datos en una nube de puntos múltiple, donde vemos la relación entre cada par de variables.

```
load( "files/40A-pizza.rda" )
pairs( pizza, panel = panel.smooth )
```




```
cor( pizza, use = "everything", method = "pearson" )
```

```
##           sales       ads      cost
## sales 1.0000000 0.7808328 0.8204250
## ads   0.7808328 1.0000000 0.8949125
## cost  0.8204250 0.8949125 1.0000000
```

vemos que todas las variables tiene una correlación elevada.

3.2.2. Ajuste del modelo

```
modelPizza1 <- lm( sales ~ ads + cost, data = pizza )
summary( modelPizza1 )
```

```
##
## Call:
## lm(formula = sales ~ ads + cost, data = pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -5.6981 -1.8223 -0.6656  2.4470  6.0123
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5836     8.5422   0.771    0.461
## ads           0.6247     1.1203   0.558    0.591
## cost          2.1389     1.4701   1.455    0.180
##
## Residual standard error: 3.989 on 9 degrees of freedom
## Multiple R-squared:  0.684,  Adjusted R-squared:  0.6138
## F-statistic: 9.741 on 2 and 9 DF,  p-value: 0.005604
```

El error típico residual es `3.99`, la $R^2 = 0{,}684$, aunque para el modelo múltiple es mejor fijarnos en su valor ajustado $R^2_a = 0{,}614$. Esto que significa que la recta de regresión explica el 61 % de la variabilidad del modelo. Además, $F = 9{,}74$ con una significación $p < 0{,}05$, lo que nos dice que nuestro modelo de regresión resulta significativamente mejor que el modelo básico.

3.3. Comparación de modelos

Pretendemos seleccionar el "mejor" subconjunto de predictores por varias razones

1. Explicar los datos de la manera más simple. Debemos eliminar predictores redundantes.
2. Predictores innecesarios añade ruido a las estimaciones.
3. La causa de la multicolinealidad es tener demasiadas variables tratando de hacer el mismo trabajo. Eliminar el exceso de predictores ayuda a la interpretación del modelo.
4. Si vamos a utilizar el modelo para la predicción, podemos ahorrar tiempo y/o dinero al no medir predictores redundantes.

Puesto que tenemos dos variables explicativas disponemos de tres modelos posibles

$$modelo1 : sales \sim ads + cost$$

$$modelo2 : sales \sim ads$$

$$modelo3 : sales \sim cost$$

Vamos a ajustar cada uno de los modelos

```
modelPizza2 <- lm( sales ~ ads, data = pizza )
summary( modelPizza2 )
```

```
##
## Call:
## lm(formula = sales ~ ads, data = pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8364 -2.7568  0.6804  3.8346  4.8971
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.9369     4.9818   3.400  0.00677 **
## ads           2.0832     0.5271   3.952  0.00272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.206 on 10 degrees of freedom
## Multiple R-squared:  0.6097, Adjusted R-squared:  0.5707
## F-statistic: 15.62 on 1 and 10 DF,  p-value: 0.00272
```

```
modelPizza3 <- lm( sales ~ cost, data = pizza )
summary( modelPizza3 )
```

```
##
## Call:
## lm(formula = sales ~ cost, data = pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7016 -1.3227 -0.6647  1.7577  6.8957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.173      7.109   0.587  0.57023
## cost           2.873      0.633   4.538  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.849 on 10 degrees of freedom
## Multiple R-squared:  0.6731, Adjusted R-squared:  0.6404
## F-statistic: 20.59 on 1 and 10 DF,  p-value: 0.001079
```

Para evitar la elección subjetiva del mejor modelo, podemos comparar todos los modelos mediante una tabla ANOVA conjunta para cada par de modelos. Hay que tener en cuenta que para poder comparar modelos estos deben estar encajados, es decir, que uno de ellos contenga al otro más otro conjunto de variables explicativas.

```
anova( modelPizza3, modelPizza1 )
```

```
## Analysis of Variance Table
##
## Model 1: sales ~ cost
## Model 2: sales ~ ads + cost
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     10 148.15
## 2      9 143.20  1    4.9472 0.3109 0.5907
```

```
anova( modelPizza3, modelPizza2 )
```

```
## Analysis of Variance Table
##
## Model 1: sales ~ cost
## Model 2: sales ~ ads
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     10 148.15
## 2     10 176.88  0   -28.731
```

Comparando ambas tablas anova deducimos que el modelo que mejor se ajusta a los datos es el `modelo3` pues reduce el error estándar.

Para este conjunto de datos, al tener sólo dos variables explicativas, aún lo podemos realizar "a mano" comparando los modelos de dos en dos. Pero cuando tenemos más variables este proceso se vuelve muy tedioso por lo que mejor hacerlo automáticamente con los *métodos paso a paso*.

3.4. Selección del "mejor" modelo

Existen distintos métodos a la hora de construir un modelo complejo de regresión con varios predictores

- El *método jerárquico* en el que se seleccionan los predictores basándose en un trabajo anterior y el investigador decide en qué orden introducir las variables predictoras al modelo.
- El *método de entrada forzada* en el que todas las variables entran a la fuerza en el modelo simultáneamente.
- Los *métodos paso a paso* que se basan en un criterio matemático para decidir el orden en que los predictores entran en el modelo.

Nosotros vamos a utilizar en R los métodos paso a paso, pero antes de verlos vamos a introducir una nueva medida de ajuste.

3.4.1. Criterio de información de Akaike (AIC)

El problema de utilizar R^2 para comparar modelos es que al añadir nuevas variables al modelo, esta medida siempre crece. Si estamos decidiendo cual de todos los modelos ajusta mejor a los datos, el modelo con más predictores siempre será el mejor ajustando. Para evitar esto se utiliza el `AIC`, una medida de ajuste que penaliza el modelo por tener más variables. Viene definido por

$$AIC = n \times \log \frac{SS_R}{n} + 2k,$$

donde n es el número de casos en el modelo, SS_R es la suma de cuadrados de los residuos del modelo y k es el número de variables predictoras.

El único problema es que no existen directrices sobre este criterio, sólo que si el `AIC` es mayor, el modelo es peor; y si el `AIC` es peor, el ajuste es mejor.

3.4.2. Metodos *paso a paso*

En R accedemos a estos métodos utilizando el comando `step( modelo, direction= )`, donde las direcciones pueden ser:

- `forward`: el modelo inicial contiene solo la constante β_0 y a partir de ahí el ordenador busca la variable predictora (dentro de las disponibles) que mejor predice la variable dependiente. Si este predictor mejora la habilidad del modelo para predecir la variable respuesta, ésta permanece en el modelo y se busca otra variable predictora. Para la segunda variable se usa como criterio de selección coger aquella que tenga la mayor correlación parcial con la respuesta. R tiene que decidir cuándo parar de añadir predictores al modelo, y para hacerlo se basa en el criterio de AIC.
- `blackward`: este método es el opuesto al anterior, R empieza con todas las variables predictoras en el modelo y estudia si el AIC disminuye cuando eliminamos del modelo alguna de las variables.
- `both`: empieza del mismo modo que el método `forward` salvo que cada vez que una variable predictora es añadida a la ecuación, se realiza un test de extracción del predictor menos útil.

El método más preferible es `backward` debido al *efecto represor* que ocurre cuando una variable predictora tiene influencia pero sólo si otra de las variables se mantiene constante. Al usar métodos *paso a paso* es aconsejable después hacer una validación cruzada, método que estudiaremos más adelante.

3.4.2.1. Métodos paso a paso en R

Vamos a desarollar estos métodos con el ejemplo `bebidas.csv`. En él se pretende explicar las muertes por cirrosis según la bebida que consuman los pacientes ((SCG), 2013).

```
dfbeb <- read.table( "files/40A-bebidas.csv", sep = ";", head = TRUE )
str( dfbeb )
```

```
## 'data.frame':    46 obs. of  6 variables:
##  $ caseid   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ cirrosis : num  41.2 31.7 39.4 57.5 74.8 59.8 54.3 47.9 77.2 56.6 ...
##  $ poblacion: int  44 43 48 52 71 44 57 34 70 54 ...
##  $ cerveza  : num  33.2 33.8 40.6 39.2 45.5 37.5 44.2 31.9 45.6 45.9 ...
##  $ vino     : int  5 4 3 7 11 9 6 3 12 7 ...
##  $ licorDuro: int  30 41 38 48 53 65 73 32 56 57 ...
```

```
# Eliminamos la variable "caseid" del conjunto pues no nos interesa.
dfbeb <- dfbeb[ , 2:6 ]
```

El archivo recoge los datos de muerte por cirrosis, el tamaño de la población, el consumo de cerveza, el consumo de vino y el consumo de licores duros. Echamos un primer vistazo a los datos

```
summary( dfbeb )
```

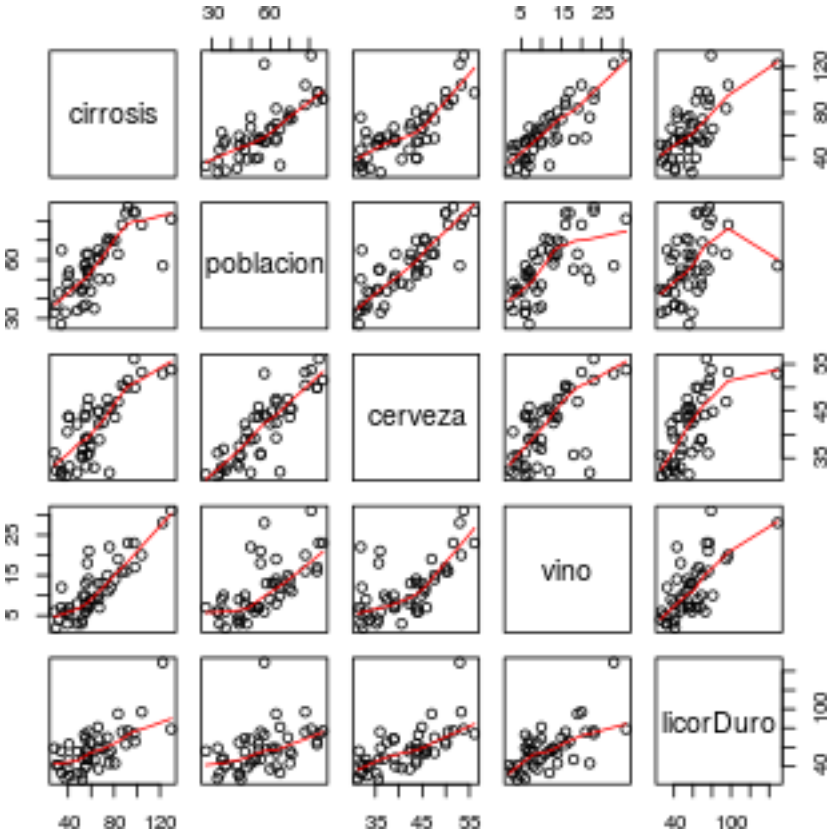
```
##     cirrosis        poblacion        cerveza          vino
##  Min.   : 28.00   Min.   :27.00   Min.   :31.20   Min.   : 2.00
##  1st Qu.: 48.90   1st Qu.:44.25   1st Qu.:35.62   1st Qu.: 6.25
##  Median : 57.65   Median :55.00   Median :42.25   Median :10.00
##  Mean   : 63.49   Mean   :56.26   Mean   :41.48   Mean   :11.59
##  3rd Qu.: 75.70   3rd Qu.:65.00   3rd Qu.:45.83   3rd Qu.:15.75
##  Max.   :129.90   Max.   :87.00   Max.   :56.10   Max.   :31.00
##    licorDuro
##  Min.   : 26.00
##  1st Qu.: 41.50
##  Median : 56.00
```

```
##  Mean   : 57.50
##  3rd Qu.: 68.75
##  Max.   :149.00
```

En todas las variables explicativas los valores de la media y la mediana son muy cercanos, lo cual es muy bueno.

Correlación

```
pairs( dfbeb, panel = panel.smooth )
```




```
cor( dfbeb, use = "everything", method = "pearson" )
```

```
##            cirrosis poblacion   cerveza      vino licorDuro
## cirrosis  1.0000000 0.7490740 0.7827244 0.8446112 0.6819694
## poblacion 0.7490740 1.0000000 0.8432812 0.6786230 0.4402957
## cerveza   0.7827244 0.8432812 1.0000000 0.6398407 0.6863643
## vino      0.8446112 0.6786230 0.6398407 1.0000000 0.6759206
## licorDuro 0.6819694 0.4402957 0.6863643 0.6759206 1.0000000
```

Como vemos en la tabla cirrosis está muy correlacionada con todas las variables explicativas y entre ellas también existe bastante correlación.

Pasamos a definir el ___modelo general_ con todas las variables.

```
modelCir <- lm( cirrosis ~ poblacion + cerveza + vino + licorDuro, data = dfbeb )
summary( modelCir )
```

```
##
## Call:
```

```
## lm(formula = cirrosis ~ poblacion + cerveza + vino + licorDuro,
##     data = dfbeb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8723  -6.7803   0.1507   7.3252  16.4419
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.96310   11.40035  -1.225   0.2276
## poblacion     0.09829    0.24407   0.403   0.6893
## cerveza       1.14838    0.58300   1.970   0.0556 .
## vino          1.85786    0.40096   4.634 3.61e-05 ***
## licorDuro     0.04817    0.13336   0.361   0.7198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.61 on 41 degrees of freedom
## Multiple R-squared:  0.8136, Adjusted R-squared:  0.7954
## F-statistic: 44.75 on 4 and 41 DF,  p-value: 1.951e-14
```

Analizamos el resumen de este primer modelo. Vemos que la mediana de los residuos es cercana a 0, lo cual es muy bueno pues queremos que los residuos tengan media cero.

Observando los coeficientes vemos que según el *estadístico t* sólo son significativas las variables `vino` y `cerveza`, ahora aplicaremos el método de selección de modelos para ver si eliminamos alguna variable.

Aún así, el modelo con todas las variables tiene un error estándar de `10.46` y una $R^2 = 0{,}8136$, aunque para el modelo múltiple es mejor fijarnos en su valor ajustado $R^2_a = 0{,}7954$. Esto quiere decir la recta de regresión explica el 79 % de la variabilidad del modelo.

Por otro lado, que el estadístico F sea alto también es bueno, la variabilidad explicada por el modelo es mayor que la que se queda sin explicar. Así $F = 44{,}75$ con una significación $p < 0{,}05$ quiere decir que nuestro modelo de regresión resulta significativamente mejor que el modelo básico. Veamos ahora si podemos mejorar el ajuste.

Selección del modelo

Vamos a aplicar los tres métodos a nuestros modelos para cómo funciona cada uno de ellos. Comenzamos con el método más recomendable, la *eliminación hacia atrás (`"backward"`)*.

```
step( modelCir, direction = "backward" )
```

```
## Start:  AIC=221.95
## cirrosis ~ poblacion + cerveza + vino + licorDuro
##
##             Df Sum of Sq    RSS    AIC
## - licorDuro  1     14.67 4625.8 220.09
## - poblacion  1     18.24 4629.3 220.13
## <none>                   4611.1 221.95
## - cerveza    1    436.38 5047.5 224.11
## - vino       1   2414.63 7025.7 239.32
##
## Step:  AIC=220.09
## cirrosis ~ poblacion + cerveza + vino
##
##             Df Sum of Sq    RSS    AIC
## - poblacion  1       6.3 4632.1 218.16
```

```
## <none>                    4625.8 220.09
## - cerveza    1    1046.8 5672.6 227.48
## - vino       1    4278.9 8904.7 248.22
##
## Step:  AIC=218.16
## cirrosis ~ cerveza + vino
##
##           Df Sum of Sq    RSS    AIC
## <none>                 4632.1 218.16
## - cerveza  1    2459.6 7091.7 235.75
## - vino     1    4951.3 9583.4 249.60

##
## Call:
## lm(formula = cirrosis ~ cerveza + vino, data = dfbeb)
##
## Coefficients:
## (Intercept)      cerveza         vino
##     -16.001        1.366        1.972
```

El proceso comienza con el modelo completo y con un `AIC` global de `221.95`. En el primer paso se considera la eliminación de todas las variables explicativas y se calcula el `AIC` relativo a dicha eliminación. `R` selecciona la variable `licorDuro` (variable que quedan por encima de `<none>`), ya que su eliminación proporciona un `AIC` más pequeño. El `AIC` resultante tras este paso y con el que compararemos en el siguiente es `220.09`.

Se considera ahora la posible eliminación de alguna de las tres variables restantes y se saca del modelo la variable `población` quedándonos con un `AIC` de `218.16`.

Por último se considera la posibilidad de suprimir alguna de las dos variables restantes, sin embargo, vemos que el proceso considera que estadísticamente resulta mejor que permanezcan en el modelo ya que al eliminarlas el `AIC` aumenta, como mínimo, hasta `235.75`.

Utilizamos ahora el método de *dos direcciones* cambiando el comando a

```
step( modelCir, direction = "both" )
```

```
## Start:  AIC=221.95
## cirrosis ~ poblacion + cerveza + vino + licorDuro
##
##             Df Sum of Sq    RSS    AIC
## - licorDuro  1     14.67 4625.8 220.09
## - poblacion  1     18.24 4629.3 220.13
## <none>                   4611.1 221.95
## - cerveza    1    436.38 5047.5 224.11
## - vino       1   2414.63 7025.7 239.32
##
## Step:  AIC=220.09
## cirrosis ~ poblacion + cerveza + vino
##
##             Df Sum of Sq    RSS    AIC
## - poblacion  1       6.3 4632.1 218.16
## <none>                   4625.8 220.09
## + licorDuro  1      14.7 4611.1 221.95
## - cerveza    1    1046.8 5672.6 227.48
## - vino       1    4278.9 8904.7 248.22
##
## Step:  AIC=218.16
```

```
## cirrosis ~ cerveza + vino
##
##             Df Sum of Sq    RSS    AIC
## <none>                   4632.1 218.16
## + poblacion  1       6.3 4625.8 220.09
## + licorDuro  1       2.7 4629.3 220.13
## - cerveza    1    2459.6 7091.7 235.75
## - vino       1    4951.3 9583.4 249.60
##
## Call:
## lm(formula = cirrosis ~ cerveza + vino, data = dfbeb)
##
## Coefficients:
## (Intercept)      cerveza         vino
##     -16.001        1.366        1.972
```

Partimos de un `AIC=221.95` y en el primer paso se elimina la variable `licorDuro`, reduciéndose a un `220.09`. En el siguiente paso además de la eliminación del resto de las variables se considera la entrada de nuevo de la variable, aunque se opta por suprimir `población` reduciendo el `AIC`a `218.16`. En el último paso se compara entre la posibilidad de recuperar alguna de las variables eliminadas o suprimir alguna más. Se decide no hacer nada, ni eliminar más ni meter las antiguas, quedando el modelo con `cervezay vino`.

Veamos por último *la selección hacia delante (`forward`)*. Debemos partir del modelo más sencillo, sólo con la constante, e indicar cuales son las posibles variables explicativas

```
mdlCir0 <- lm( cirrosis ~ 1 , data = dfbeb )
step( mdlCir0, direction = "forward", ~ poblacion + cerveza + vino + licorDuro )
```

```
## Start:  AIC=291.23
## cirrosis ~ 1
##
##             Df Sum of Sq     RSS    AIC
## + vino       1     17650  7091.7 235.75
## + cerveza    1     15158  9583.4 249.60
## + poblacion  1     13883 10858.7 255.35
## + licorDuro  1     11507 13234.6 264.45
## <none>                   24741.3 291.23
##
## Step:  AIC=235.75
## cirrosis ~ vino
##
##             Df Sum of Sq    RSS    AIC
## + cerveza    1   2459.58 4632.1 218.16
## + poblacion  1   1419.04 5672.6 227.48
## + licorDuro  1    562.06 6529.6 233.95
## <none>                   7091.7 235.75
##
## Step:  AIC=218.16
## cirrosis ~ vino + cerveza
##
##             Df Sum of Sq    RSS    AIC
## <none>                   4632.1 218.16
## + poblacion  1    6.2931 4625.8 220.09
## + licorDuro  1    2.7287 4629.3 220.13
##
```

```
## Call:
## lm(formula = cirrosis ~ vino + cerveza, data = dfbeb)
##
## Coefficients:
## (Intercept)         vino      cerveza
##     -16.001        1.972        1.366
```

Es el mismo procedimiento que para el método hacia atrás pero aquí se parte del modelo sin variables explicativas y se considera en cada paso la posible inclusión de una nueva variable (los signos ahora son `+`). La primera variable que se añade al modelo es `vino` seguida de `cerveza` pues la inclusión de alguna de las otras incrementa el `AIC`.

En todos los métodos nos hemos quedamos con el mismo modelo final. La última parte del método muestra los coeficientes del modelo con el que nos quedamos finalmente, que es

```
modelCirf <- lm( cirrosis ~ cerveza + vino, data = dfbeb )
summary( modelCirf )
```

```
##
## Call:
## lm(formula = cirrosis ~ cerveza + vino, data = dfbeb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8158  -6.8539   0.0599   7.2160  16.3714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.0008    10.1530  -1.576    0.122
## cerveza       1.3656     0.2858   4.778 2.08e-05 ***
## vino          1.9723     0.2909   6.780 2.69e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.38 on 43 degrees of freedom
## Multiple R-squared:  0.8128, Adjusted R-squared:  0.8041
## F-statistic: 93.34 on 2 and 43 DF,  p-value: 2.268e-16
```

En este modelo final la mediana de los residuos es prácticamente cero, lo que va a significar que los residuos van a tener una media muy cercana a 0. Vemos que las dos variables `cerveza` y `vino` son significativas. Tenemos un error estándar de `10.38`, y un $R_a^2 = 0{,}8041$ lo que significa que el modelo explica un 80 % de la variabilidad de los datos. Finalmente vemos que el *test* F es significativo ($p < 0{,}01$) con un valor elevado, lo cual nos indica que el modelo se ajusta significativamente a los datos.

```
anova( modelCirf )
```

```
## Analysis of Variance Table
##
## Response: cirrosis
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## cerveza    1 15158.0 15158.0 140.713 3.787e-15 ***
## vino       1  4951.3  4951.3  45.963 2.685e-08 ***
## Residuals 43  4632.1   107.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La tabla anova nos confirma que las variables explicativas de nuestro modelo son significativas, y vemos que la suma de cuadrados explicada por el modelo es mucho mayor que la suma de cuadrados de los residuos, por

tanto podemos afirmar que $R^2 \neq 0$.

Aplicar el modelo

Podemos utilizar la parte `Coefficients` proporciona el resumen del modelo para analizar individualmente la contribución de cada variable predictora la explicación de la dependiente.

Definimos el modelo reemplazando los b-valores en la ecuación inicial y obtenemos el modelo

$$cirrosis = -16{,}001 + 1{,}366 \times cerveza + 1{,}972 \times vino$$

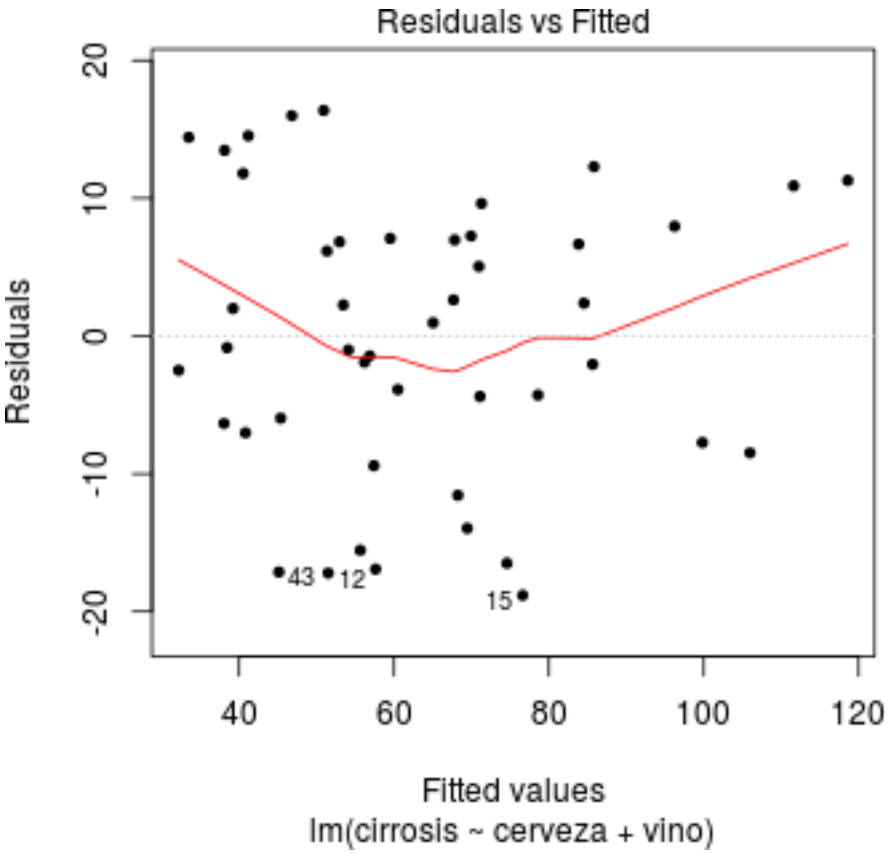
3.5. Diagnóstico del modelo

Para este apartado nos hemos apoyado fundamentalmente en el libro J.Faraway (2009).

Al haber generado el modelo basándonos en una muestra nos tenemos que preguntar si el modelo se ajusta bien a los datos observados o está influenciado por un pequeño número de casos, y por otro lado si el modelo se puede generalizar a otras muestras. Es un error pensar que porque un modelo se ajuste bien a los datos observados entonces podemos tomar conclusiones más allá de nuestra muestra.

Para poder generalizar un modelo de regresión debemos comprobar los supuestos del modelo, y una vez seguros de que se cumplen, para comprobar si el modelo se puede generalizar utilizaremos la *validación cruzada*. Empezamos analizando gráficamente los supuestos

```
plot( modelCirf, which = 1, pch = 20 )
```



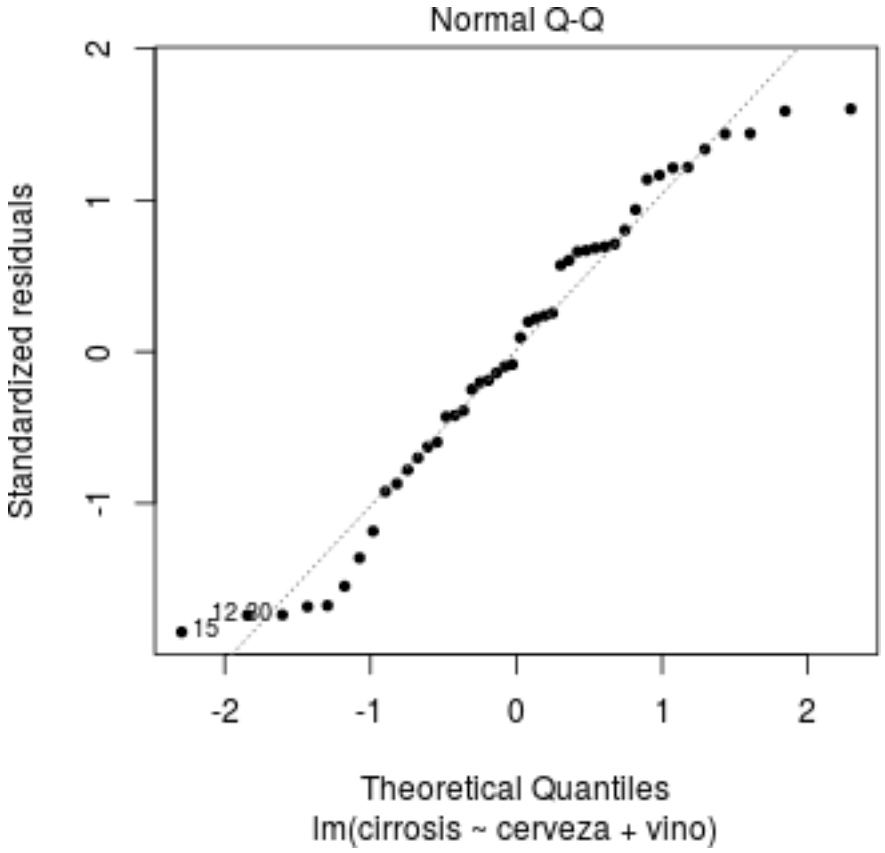
Este primer gráfico enfrenta los errores residuales frente a sus valores ajustados. El residuos deben estar distribuidos al azar alrededor de la línea horizontal que representa un error residual de cero; es decir, no debe haber una tendencia clara en la distribución de puntos. Una tendencia en la variabilidad de los residuos sugiere que la varianza está relacionada con la media, violando el supuesto de varianza constante.

Si el gráfico tiene forma de embudo, es decir, si los puntos parecen estar más o menos extendidos a lo largo del gráfico, entonces lo más probable es que exista heterocedastididad en los datos. En este caso los datos parecen exhibir una ligera tendencia con un incremento de la varianza en los extremos.

Si hubiera algún tipo de curva en la gráfica entonces se ha violado el supuesto de linealidad. Y si los datos parecen seguir un patrón y además están más extendidos por en algunos puntos de la gráfica que en otros entonces probablemente se incumplan los supuestos de homogeneidad de varianza y linealidad.

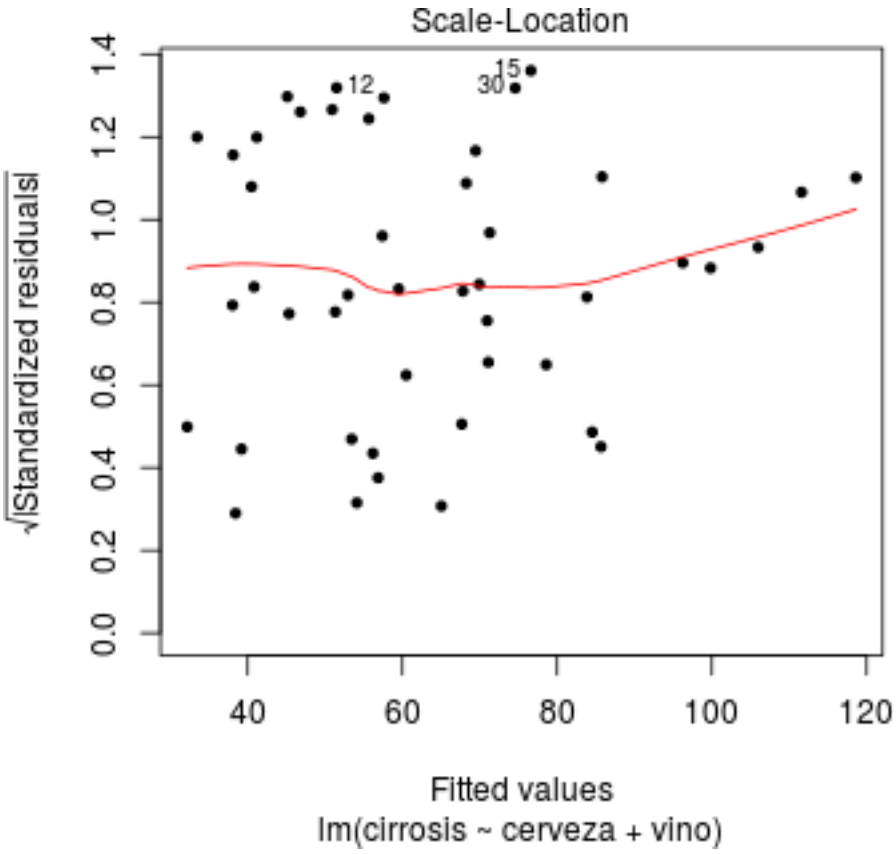
En general, parace que en nuestro modelo no se violan ninguno de los supuestos.

```
plot( modelCirf, which = 2, pch = 20 )
```



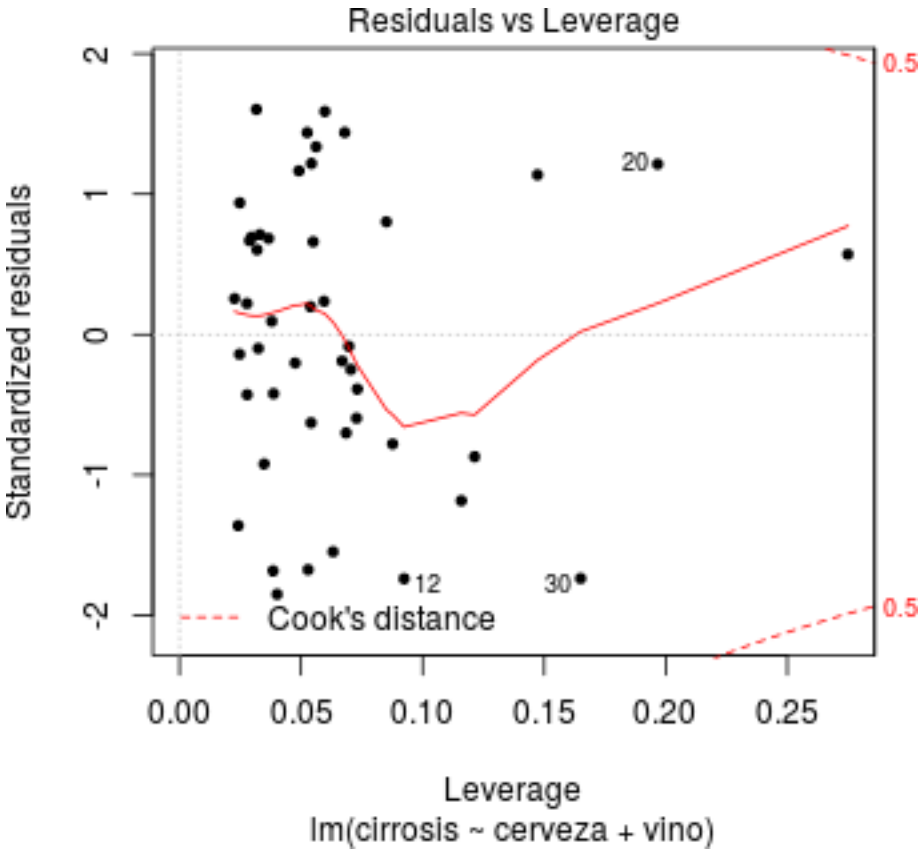
En este gráfico los residuos tipificados se trazan contra los cuantiles de una distribución normal estándar. Si los residuos se distribuyen normalmente los datos se deben situar a lo largo de la línea. En este caso, los datos no hacen parecen tener una distribución normal.

```
plot( modelCirf, which = 3, pch = 20 )
```



El tercero es el gráfico *escala-ubicación* en el que los residuos están estandarizados por sus desviaciones estándar estimadas. Esta gráfica se utiliza para detectar si la difusión de los residuos es constante en el rango de valores ajustados. Una vez más, se aprecia una tendencia muy leve en los datos de tal manera que los valores altos muestran una mayor variación.

```
plot( modelCirf, which = 5, pch = 20 )
```



Finalmente el cuarto gráfico muestra el valor *leverage* de cada punto, la medida de su importancia en la determinación del modelo de regresión. Están representados los datos que ejercen mayor influencia.

Superponen al diagrama de puntos *leverage* las curvas de nivel para la distancia de Cook, que es otra medida de la importancia de cada observación a la regresión. Si la línea de distancia Cooks abarca a algún punto de datos, significa que el análisis puede ser muy sensible a ese punto y quizá sea conveniente repetir el análisis excluyendo los datos. Distancias pequeñas significan que la eliminación de la observación tiene poco efecto sobre los resultados de la regresión y distancias mayores a 1 son sospechosas, sugieren la presencia de un posible valor atípico o de un modelo pobre.

Pasamos ahora a estudiar el modelo analíticamente, para ello obtenemos los residuos, los valores ajustados y estadísticos del modelo mediante el siguiente código:

```
dfbeb$fitted.modelCirf    <- fitted( modelCirf )
dfbeb$residuals.modelCirf <- residuals( modelCirf )
dfbeb$rstudent.modelCirf  <- rstudent( modelCirf )
```

3.5.1. Normalidad

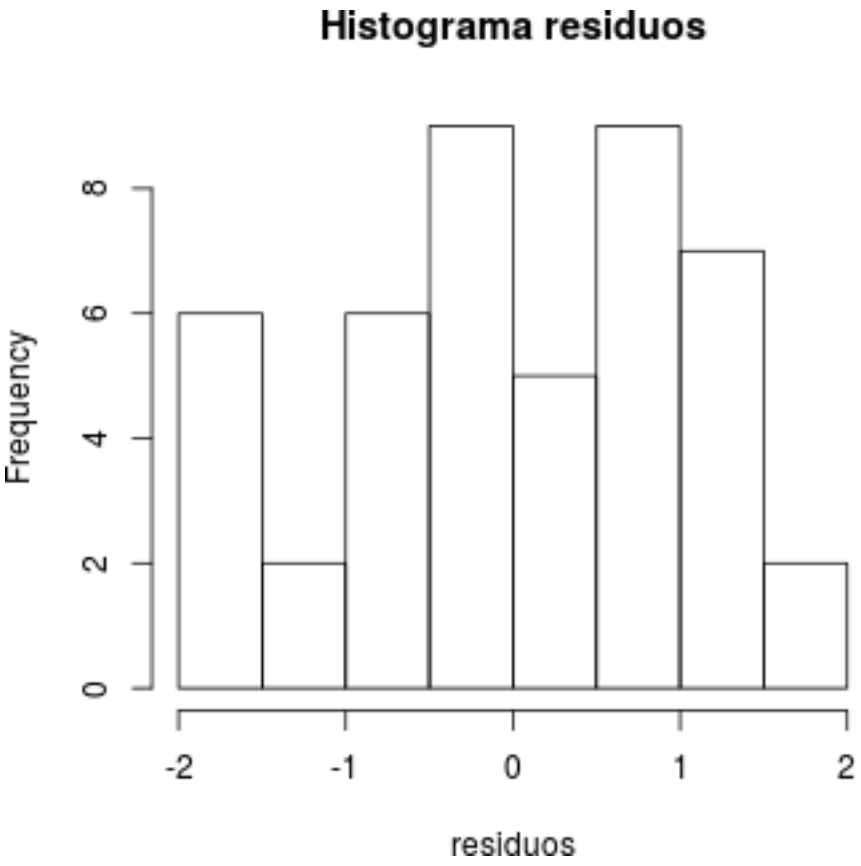
En el gráfico *Q-Q plot* que vimos antes sugería falta de normalidad en los datos. Lo comprobamos

```
ks.test( dfbeb$rstudent.modelCirf, "pnorm" )
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  dfbeb$rstudent.modelCirf
```

```
## D = 0.10577, p-value = 0.6434
## alternative hypothesis: two-sided
```

```
hist( dfbeb$rstudent.modelCirf, xlab = "residuos", main = "Histograma residuos"  )
```




```
# densidad
```

El p-valor para el contraste de normalidad es mayor que `0.05` (`p = 0.6434`) y además el histograma se parece a una distribución normal (curva en forma campana) entonces no hay problemas de normalidad.

3.5.2. Homogeneidad de varianzas

```
bptest( modelCirf, studentize = FALSE, data = dfbeb )
```

```
##
##  Breusch-Pagan test
##
## data:  modelCirf
## BP = 0.66652, df = 2, p-value = 0.7166
```

Significación `p = 0.7166`, mayor de `0.05`, por lo que podemos decir que la varianza es constante a lo largo de la muestra.

3.5.3. Autocorrelación

```
dwtest( modelCirf, alternative = "two.sided", data = dfbeb )
```

```
##
##  Durbin-Watson test
##
## data:  modelCirf
## DW = 2.5152, p-value = 0.07225
## alternative hypothesis: true autocorrelation is not 0
```

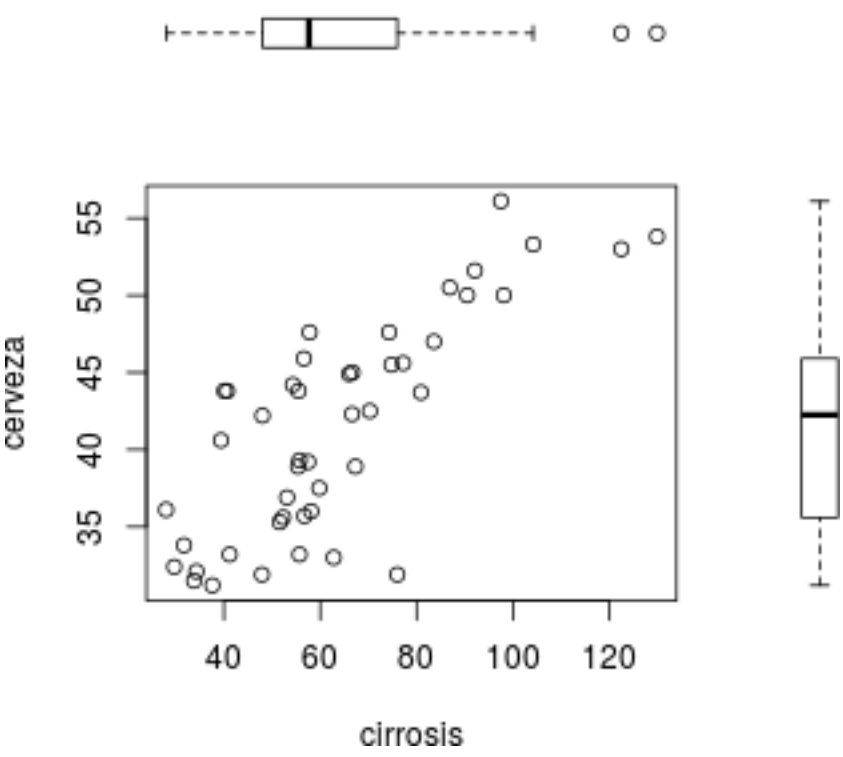
Aceptamos la hipótesis nula de que no existe correlación entre los residuos con un p-valor superior a `0.05`.

3.5.4. Casos atípicos y residuos

Podemos encontrar los valores atípicos observando grandes diferencias entre los datos muestrales y los datos ajustados por el modelo, es decir, estudiando los **residuos**.

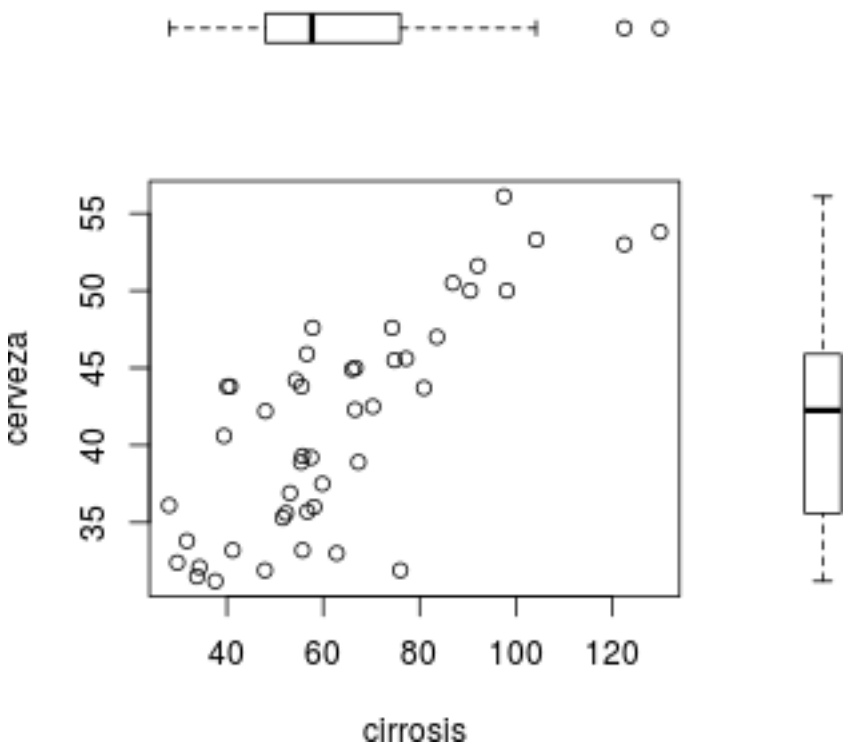
Si el modelo se ajusta bien a los datos muestrales entonces todos los residuos serán pequeños, mientras que si el ajuste del modelo es pobre los residuos serán grandes. Además, si algún caso sobresale por tener un gran residuo este podría ser entonces un valor atípico.

Vamos a analizar si existen valores atípicos en nuestro ejemplo. En el primer gráfico enfrentamos `cirrosis` con `cerveza` y en el segundo `cirrosis` con `vino`.



Para este primer gráfico se observan dos posibles valores atípicos.

Estudiamos el gráfico para las otras dos variables



Se observan los mismos candidatos a valores atípicos. Hacemos el test de Bonferroni para comprobarlo.

```
outlierTest( modelCirf )
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 15 -1.906236           0.063478           NA
```

Obtenemos que el valor 15 es un atípico.

3.6. Análisis de la influencia.

Con este análisis pretendemos ver si hay alguna observación que es demasiado influyente sobre los coeficientes del modelo, nos ayuda a determinar si el modelo de regresión es estable a lo largo de la muestra o si está perjudicado por unos pocos casos influyentes.

Utilizamos la función `influence.measures` que nos proporciona todas las medidas de influencia. Explicamos, a partir de los resultados de aplicar la función, cada una de las medidas:

```
infl <- influence.measures( modelCirf )
summary( infl )
```

```
## Potentially influential observations of
##   lm(formula = cirrosis ~ cerveza + vino, data = dfbeb) :
##
##    dfb.1_ dfb.crvz dfb.vino dffit cov.r   cook.d hat
## 20 -0.10  -0.01     0.45     0.60  1.20    0.12   0.20_*
## 38  0.26  -0.30     0.31     0.35  1.45_*  0.04   0.27_*
```

Analizamos la tabla resumen:

- la primera columna indica el índice de las observaciones potencialmente influyentes.

- las columnas que comienzan con `dfb` proporcionan las observaciones potencialmente influyentes sobre cada uno de los coeficientes del modelo.
- la columna `dffits` identifica las observaciones influyentes según el estadístico *DFFITS*.
- la columna `cov.r` muestra las observaciones potencialmente influyentes según el estadístico *COVRATIO*.
- la columna `cook.d` proporciona la *distancia de Cook* .
- la última columna presenta las observaciones que pueden resultar influyentes según los `leverages`.

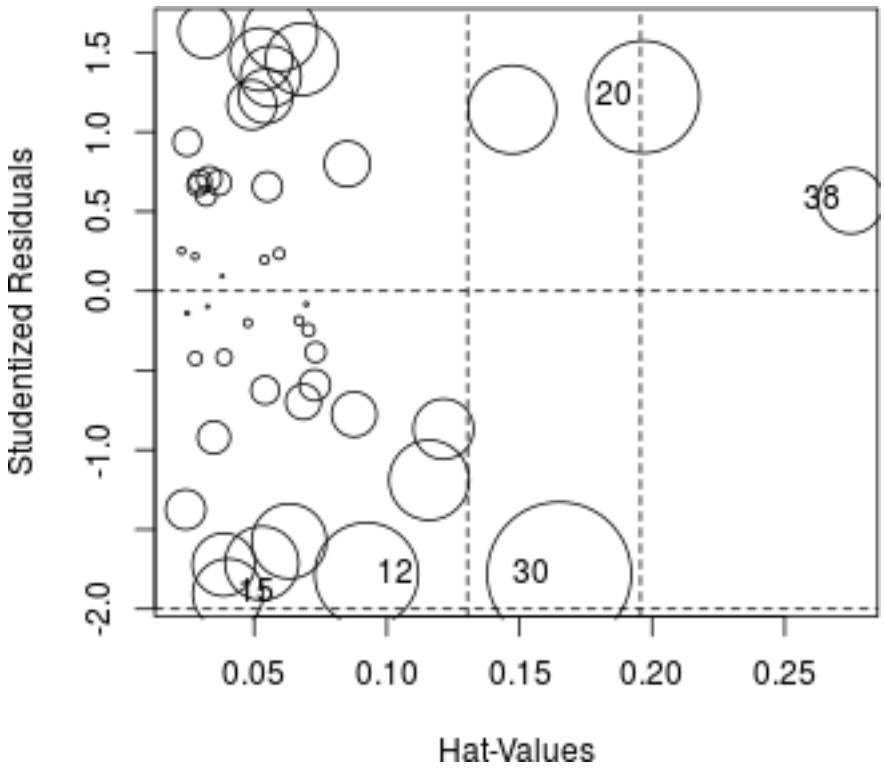
En cada columna el asterisco señala si realmente la observación puede ser influyente. En este caso tenemos que la observación `38` resulta influyente con el estadístico `cov.r`, y las `38` y la `20` para los 'leverages.

Analizamos un poco más estas medidas:

- Los **leverages (hat)** varían entre `0` (indicando que el caso no tiene influencia en absoluto) y `1` (indicando que ese caso tiene influencia completa sobre el modelo). Si ninguno de los casos ejerce excesiva influencia sobre el modelo entonces esperaremos que todos los valores 'leverage' estén entorno al valor medio `((k+1)/n)`, donde `k` es el número de predictores y `n` el número de participantes. Buscamos valores el doble o triple que `((k+1)/n)` para considerarlos como influyentes.
- Para **la distancia de Cook** se considera que valores mayores que `1` pueden ser causa de preocupación. Si un caso es un valor atípico pero su distancia de Cook es menor que 1, entonces no existe necesidad real de eliminar este dato ya que realmente no tiene un gran efecto sobre el modelo de regresión.

Lo estudiamos gráficamente. En el primer gráfico se muestra mediante círculos de distinto tamaño la influencia que cada punto ejerce sobre el modelo y en el segundo están representadas en orden ascendente las distancias de Cooks.

```
influencePlot( modelCirf, id.n = 2 )
```

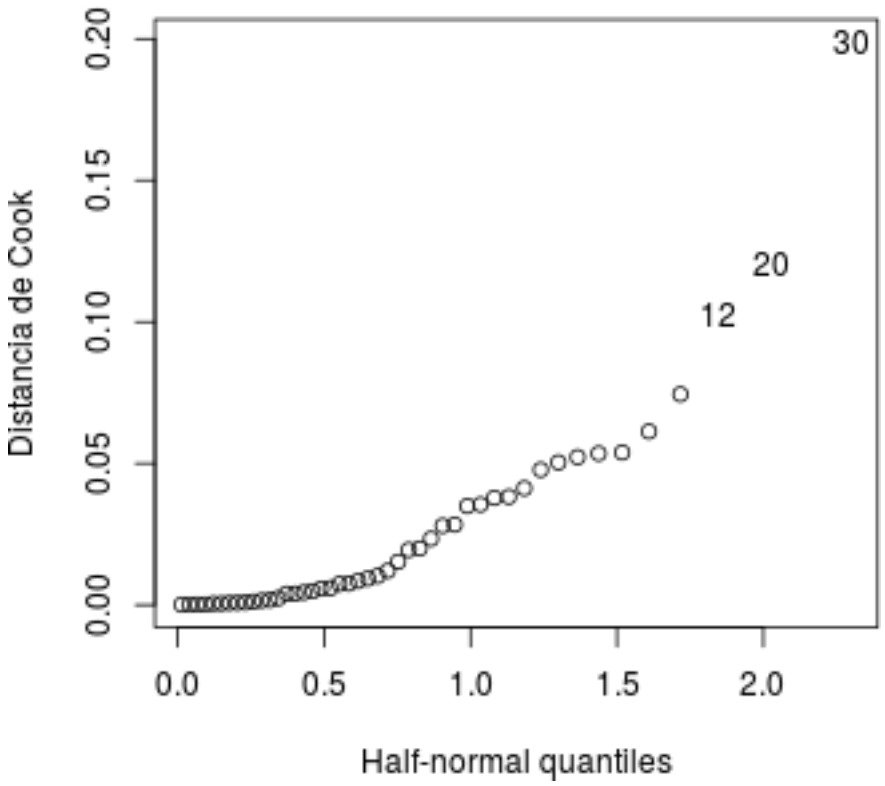



```
##       StudRes        Hat      CookD
## 12 -1.7834216 0.09235642 0.10267272
## 15 -1.9062358 0.04015442 0.04774704
```

```
## 20  1.2203502 0.19668424 0.12017592
## 30 -1.7810384 0.16500136 0.19889565
## 38  0.5664808 0.27494848 0.04121404
```

En este primer gráfico vemos que las medidas más influyentes son la 30, la 20 y la 12. Vemos el gráfico de las distancias de Cook.

```
cook   <- cooks.distance( modelCirf )
labels <- rownames( dfbeb )
halfnorm( cook, 3, labs = labels, ylab = "Distancia de Cook" )
```



En este gráfico volvemos a obtener que los puntos más influyentes son el 30, el 20 y el 12, pero como en ningún caso esta distancia es mayor que 1, pues para el valor más elevado es 0.2, podemos afirmar que ninguno de ellos es un caso atípico y no es necesario eliminarlos del modelo.

La forma habitual de proceder es eliminar dichas observaciones del modelo y comenzar de nuevo todo el proceso, sin embargo como el modelo cumple todas las hipótesis, eliminar dichas observaciones podría provocar que el nuevo modelo fuera incorrecto y tuviéramos que volver al modelo anterior.

Hay que tener en cuenta que los límites marcados para identificar una observación como influyente son aproximados, y por tanto deben ser tomados como orientación, a salvo que el valor obtenido sea exageradamente llamativo.

3.7. Validación cruzada

Al utilizar métodos paso a paso es recomendable hacer una validación cruzada de nuestro modelo para evaluar su eficacia prediciendo la variable dependiente en una muestra diferente. Evaluar la precisión de un modelo a través de diferentes muestras es lo que se conoce como *validación cruzada*.

Para poder generalizar un modelo este debe ser capaz de predecir con precisión la misma variable dependiente del mismo conjunto de predictores en un grupo diferente de gente. Si aplicamos el modelo a una muestra diferente y su poder predictivo se reduce severamente, entonces no es generalizable.

El método usual es calcular además de la R^2 su valor ajustado, pues es un indicador de la pérdida de poder predictivo. Mientras R^2 nos dice cuánta varianza de Y representa el modelo de regresión, la R_a^2 cuantifica la varianza de Y que representaría el modelo si este hubiera sido obtenido de la población donde hemos tomado la muestra. Si los valores de R^2 y R_a^2 están próximos significa que el modelo de regresión es bueno.

Sin embargo, esta medida ha sido criticada porque no dice nada sobre la efectividad del modelo de regresión si se aplica a un conjunto de datos totalmente distinto. Una alternativa sería *partir* los datos y cruzarlos, es decir, hacer una división aleatoria del conjunto de datos (p. ej un 80 %-20 %), calcular la ecuación de regresión en ambos conjuntos y comparar los modelos resultantes. Comparando los valores de R^2 y los b-valores en las dos muestras podemos saber la bondad del modelo original.

Para realizar la validación cruzada en `R` usamos la función `cv.lm( datos, modelo, m)`, donde `m`es el número de subconjuntos en los que asignamos los datos al azar. Cada subconjunto se retira del modelo,sucesivamente, mientras que los datos restantes se utiliza para volver a ajustar el modelo de regresión y predecir en las observaciones eliminados.

```
library( DAAG )
cv.lm( dfbeb, modelCirf, m = 2 )
```

3.8. Predicción

Para calcular las ecuaciones de predicción procedemos de forma similar al caso de regresión lineal simple, la única diferencia es que hay que dar valores predictivos para todas las variables que aparezcan en el modelo.

```
#Definiendo un intervalo para la vble vino.
x0   <- seq( min( dfbeb$vino ), max( dfbeb$vino ), length = length( dfbeb$vino ) )
dbp  <- data.frame( poblacion = 56, cerveza = 41, vino = x0, licorDuro = 58 )
pred <- predict( modelCirf, dbp, interval = "prediction", se.fit = TRUE, data = dfbeb )
head( pred$fit )
```

```
##        fit      lwr      upr
## 1 43.93498 22.08611 65.78385
## 2 45.20602 23.44836 66.96368
## 3 46.47705 24.80439 68.14972
## 4 47.74809 26.15414 69.34204
## 5 49.01913 27.49753 70.54072
## 6 50.29016 28.83451 71.74582
```

3.9. Diagnósticos de colinealidad (multicolinealidad)

Si en un modelo de regresión lineal múltiple alguna variable predictora es combinación lineal de otras de las variables del modelo, entonces el modelo es irresoluble, debido a que en ese caso la matriz $X'X$ es singular, es decir, su determinante es cero y no se puede invertir.

Una variable X_1 es combinación lineal de $X_2, \ldots, X_i$ con $i > 2$, si dichas variables están relacionadas por la expresión $X_1 = \beta_1 + \beta_2 X_2 + ... + \beta_i X_i$, siendo los β_i constantes. En tal caso el coeficiente de correlación múltiple también será `1`.

Por tanto, la *multicolinealidad* existe si hay una fuerte correlación entre dos o más variables predictoras del modelo, es decir, cuando alguno de los coeficientes de correlación simple o múltiple entre algunas de las variables independientes es `1`. Si existe una colinealidad perfecta entre predictores es imposible obtener

estimadores únicos para los coeficientes de regresión ya que hay un número infinito de coeficientes que funcionarían igual de bien.

En la práctica esta colinealidad exacta raras veces ocurre, pero sí surge con cierta frecuencia la llamada *casi-colinealidad*, cuando alguna variable es "casi" combinación lineal de otra u otras. Dicho de otro modo, algunos coeficientes de correlación simple o múltiple entre las variables independientes están cercanos a 1, aunque no llegan a dicho valor.

En ese caso la matriz $X'X$ es casi-singular, es decir, su determinante no es cero pero es muy pequeño. Como para invertir una matriz hay que dividir por su determinante surgen problemas de *precisión* en la estimación de los coeficientes, ya que los algoritmos de inversión de matrices pierden precisión al tener que dividir por un número muy pequeño, siendo además *inestables*.

Hay varias formas de **detectar este problema:**

- **Observar los estadísticos estimados**: cuando la prueba muestra que el modelo es globalmente significativo, es decir, que los coeficientes estimados son estadísticamente diferentes de cero, pero se encuentran unos valores estimados bajos que demuestran que los coeficientes no son significativos.
- **Observar la matriz de correlación entre parejas de regresores**: si este coeficiente es mayor a 0.8 entonces la multicolinealidad es un problema grave. Sin embargo, esta condición se puede considerar suficiente pero no necesaria, la multicolinealidad puede existir a pesar de que las correlaciones sean comparativamente bajas (es decir, inferiores a 0.5).
- **Regresiones auxiliares**: dado que la multicolinealidad surge por la relación lineal entre variables explicativas, se pueden estimar regresiones entre las variables explicativas y adoptar la *regla práctica de Klien*. Este sugiere que si el modelo obtenido en la regresión auxiliar es mayor que el global obtenido con todos los regresores, hay un serio problema de multicolinealidad.
- **Estimar el Factor de Inflación de Varianza (FIV)**: indica si el predictor tiene una fuerte relación lineal con otro predictor y es el que vamos a calcular con `R`. Aunque no existen reglas generales se tienen los siguientes criterios:
  - Un `VIF > 10` es causa de preocupación.
  - Si `VIF` es sustancialmente mayor que `1` entonces la regresión puede verse perjudicada.
  - `Tolerancia = 1/VIF` debajo de `0.1` indica un problema serio.
  - `Tolerancia` debajo de `0.2` indica un problema potencial.

Si **identificamos multicolinealidad** no hay mucho que podamos hacer, la solución no es fácil:

- Podemos intentar eliminar la variable menos necesaria implicada en la colinealidad, a riesgo de obtener un modelo menos válido. Sin embargo, un problema común es no saber qué variable debemos omitir. Cualquiera de las variables problemáticas puede ser omitida, no hay fundamentos estadísticos para suprimir una variable en vez de otra.
- Se recomienda que si eliminamos una variable predictora, ésta se reemplace por otra igualmente importante que no tenga una colinealidad tan fuerte.
- Se puede intentar cambiar la escala de medida de la variable en conflicto (es decir, transformarla). Sin embargo estas transformaciones hacen al modelo muy dependiente de los datos actuales, invalidando su capacidad predictiva.
- También se puede recurrir a aumentar la muestra para así aumentar la información en el modelo y ver si la multicolinealidad puede disminuir, aunque no siempre será posible.
- La última posibilidad, aunque más compleja cuando hay varios predictores, es hacer un análisis factorial y usar las puntuaciones del factor resultante como predictor.

Supongamos que estamos en el ejemplo de `dfbeb` y le realizamos un test de multicolinealidad al `modelCirf`:

```
vif( modelCirf )
```

```
##  cerveza     vino
```

```
## 1.693182 1.693182
```

```
sqrt( vif( modelCirf ) ) > 2
```

```
## cerveza    vino
##   FALSE   FALSE
```

Nuestro modelo no presenta problemas de multicolinealidad.

3.10. Resumen de código en R

```
#Leer los datos de un fichero .csv
df <- read.table( "files/40A-file.csv", sep = ";", head = TRUE )

### Primera aproximación a los datos
str( df )
summary( df )

# Correlación
# Gráfico de dispersión multivariante
pairs( df, panel = panel.smooth )

# Matriz de correlación
cor( df, use = "everything", method = "pearson" )
corr.test( df, use = "complete", method = "pearson" )

## Correlación parcial (si fuera necesario)
library( "ppcor" )
pcor.test( df$var1, df$var2, df$var3 )


# Modelo de regresión múltiple

## Creamos el modelo de regresión
modelo <- lm( var1 ~ var2 + var3 + ... , data = df )
summary(modelo) # analizamos el modelo inicial

## Comparación de modelos (encajados)
anova( model3, model1 )
anova( model3, model2 )

## Selección del modelo mediante los métodos paso a paso
### Método hacia atrás
step( modelo, direction = "backward" )

### Método de dos sentidos
step( modelo, direction = "both" )

### Método hacia delante
mdlCir0 <- lm( var1 ~ 1 , data = df )
step( mdlCir0, direction = "forward",  ~ var1 + var2 + var3 + var4 )
modelo <- lm( var1 ~ var2 + var3, data = df )


# Análisis del modelo final
```

```r
summary( modelo )
anova ( modelo )

## Diagnósitco del modelo


# Gráficamente
plot( modelo, which = 1 )
plot( modelo, which = 2 )
plot( modelo, which = 3 )
plot( modelo, which = 5 )

## Contrastes
### Obtenemos los residuos del modelo y valores ajustados
df$fitted.modelo    <- fitted( modelo )
df$residuals.modelo <- residuals( modelo )
df$rstudent.modelo  <- rstudent( modelo )

### Normalidad
ks.test( df$rstudent.modelo, "pnorm" )
hist( df$rstudent.modelo, xlab = "residuos", main = "histograma residuos"  )

### Homogeneidad de varianzas
library( lmtest )
bptest( modelo, studentize = FALSE, data = df )

### Autocorrelación
dwtest( modelo, alternative = "two.sided", data = df )

### Valores atípicos
library( car )
outlierTest( modelo )

### Análisis de la influencia
#### Tabla con las medidas de influencia
infl <- influence.measures( modelo )
summary( infl )

#### Gráfico medidas influyentes
influencePlot( modelo, id.n = 2 )

#### Gráfico de las distancias de Cook
cook   <- cooks.distance( modelo )
labels <- rownames( df )
library( faraway )
halfnorm( cook, 3, labs = labels, ylab = "Distancia de Cook" )

## validación cruzada
library( DAAG )
cv.lm( df, modelo, m = 2 )

# Predicción.
## Valores concretos de cada vble
predict( modelo, data.frame( var1 = 39, var = 62, var3 = 18 ),
```

```
        interval = "prediction", data = df )


# Poniendo un intervalo para una de las vbles.
x0   <- seq( min( df$var2 ), max( df$var2 ), length = length( df$var2 ) )
pred <- predict( modelo, data.frame( var2 = x0 ),
                 interval = "prediction", data = df )
head( pred )


# Multicolinealidad
library( car )
vif( modelo )
sqrt( vif( modelo ) ) > 2
```

3.11. Predictores categóricos. Variables *dummy*

Uno de los supuestos de la regresión lineal es que las variables del modelo deben ser continuas o categóricas con solo dos categorías. En el caso de variables con más de dos categorías usaremos lo que se conoce como **variables dummy**, variables ficticias, simuladas.

Esta codificación es una manera de representar varios grupos de personas pero usando sólo unos y ceros. El proceso consiste crear varias variables siguiendo estos pasos:

1. Contar el número de grupos que queremos re-codificar y restarle 1.
2. Crear tantas nuevas variables como el valor obtenido en 1. Estas serán las variables *dummy*.
3. Elegir uno de los grupos como el *grupo de referencia*, es decir, el grupo contra el que se van a comparar todos los demás grupos. Normalmente se toma el grupo control o aquel que representa a la mayoría de la población.
4. Elegido el grupo referencia fijamos el valor 0 a ese grupo en todas las variables dummy.
5. Para la primera variable dummy asignamos el valor 1 al primer grupo que queramos comparar contra el grupo referencia. Al resto de grupos le damos el valor 0.
6. En la segunda variable dummy damos el valor 1 al segundo grupo que queramos cotejar y 0 al resto de grupos.
7. Repetimos el proceso hasta acabar con todas las variables dummy.

Veamos cómo hacer una **codificación dummy en R**. Para ello utilizamos el conjunto de datos `40A-festival.csv`, archivo que contiene los niveles de higiene de los asistentes a un famoso festival murciano y una variable que mide el cambio en la higiene durante sus tres días de duración.

Los individuos están clasificados en cuatro grupos según sus estilos musicales, estos son indie, metal, pop y sin estilo predominante. Queremos estudiar los cambios de higiene para cada uno de ellos a lo largo del festival.

```
dffest <- read.table( "files/40A-festival.csv", sep = ";", head = TRUE )
head( dffest )
```

```
##   ticknumber     musica dia1 dia2 dia3 cambio
## 1       2111      metal 2.65 1.35 1.61  -1.04
## 2       2229        pop 0.97 1.41 0.29  -0.68
## 3       2338 sin estilo 0.84   NA   NA     NA
## 4       2384        pop 3.03   NA   NA     NA
## 5       2401 sin estilo 0.88 0.08   NA     NA
## 6       2405        pop 0.85   NA   NA     NA
```

Observamos que al contener texto, `R` ha convertido la variable `musica` en un factor de 4 niveles ordenados de 1 a 4.

```
str( dffest )
```

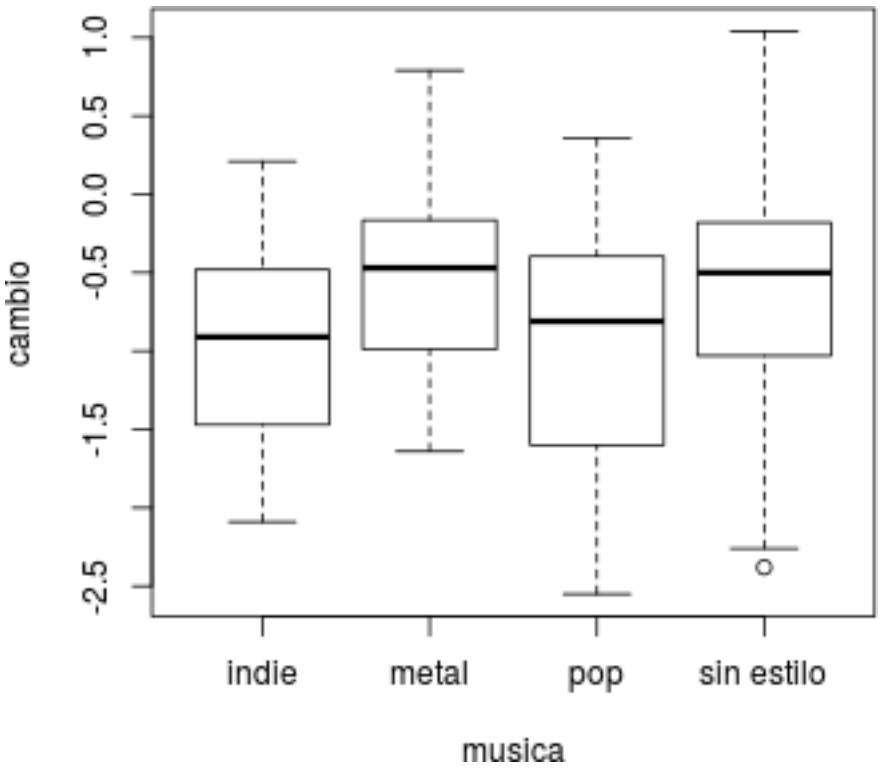
```
## 'data.frame':    810 obs. of  6 variables:
##  $ ticknumber: int  2111 2229 2338 2384 2401 2405 2467 2478 2490 2504 ...
##  $ musica    : Factor w/ 4 levels "indie","metal",..: 2 3 4 3 4 3 1 1 3 4 ...
##  $ dia1      : num  2.65 0.97 0.84 3.03 0.88 0.85 1.56 3.02 2.29 1.11 ...
##  $ dia2      : num  1.35 1.41 NA NA 0.08 NA NA NA NA 0.44 ...
##  $ dia3      : num  1.61 0.29 NA NA NA NA NA NA NA 0.55 ...
##  $ cambio    : num  -1.04 -0.68 NA NA NA NA NA NA NA -0.56 ...
```

```
levels( dffest$musica )
```

```
## [1] "indie"      "metal"      "pop"        "sin estilo"
```

Empezamos con un gráfico para hacernos una idea de cómo afecta las preferencias musicales de los asistentes a sus cambios en la higiene durante el desarrollo del festival.

```
plot( cambio ~ musica, data = dffest )
```



Creamos las variables *dummy*. Lo podemos hacer automáticamente mediante el comando `contr.treatment(numero de grupos, base = número del grupo referencia)`, donde en nuestro caso tenemos cuatro grupos y el grupo de referencia es el último, *sin estilo*.

```
contrasts( dffest$musica ) <- contr.treatment( 4, base = 4 )
```

```
## attr(,"contrasts")
##            1 2 3
## indie      1 0 0
## metal      0 1 0
## pop        0 0 1
## sin estilo 0 0 0
## Levels: indie metal pop sin estilo
```

Es preferible hacer este proceso de forma manual ya que tenemos control sobre la codificación y podemos poner nombres significativos a las variables. Tomamos la categoría *sin estilo* como grupo referencia

```
Indie_dum <- c( 1, 0, 0, 0 )
Metal_dum <- c( 0, 1, 0, 0 )
Pop_dum <- c( 0, 0, 1, 0 )
contrasts(dffest$musica) <- cbind( Indie_dum, Metal_dum, Pop_dum )
```

Si inspeccionamos la variable `dffest$musica` vemos que se obtiene el mismo resultado

```
## attr(,"contrasts")
##            indie_dum metal_dum pop_dum
## indie              1         0       0
## metal              0         1       0
## pop                0         0       1
## sin estilo         0         0       0
## Levels: indie metal pop sin estilo
```

Una vez creadas las variables *dummy* se ejecuta el modelo de regresión de la misma manera que para cualquier otro tipo de regresión

```
modelFesti <- lm( cambio ~ musica, data = dffest )
summary( modelFesti )
```

```
##
## Call:
## lm(formula = cambio ~ musica, data = dffest)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82569 -0.50489  0.05593  0.42430  1.59431
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.55431    0.09036  -6.134 1.15e-08 ***
## musicaIndie_dum -0.40998    0.20492  -2.001   0.0477 *
## musicaMetal_dum  0.02838    0.16033   0.177   0.8598
## musicaPop_dum   -0.41152    0.16703  -2.464   0.0152 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6882 on 119 degrees of freedom
##   (687 observations deleted due to missingness)
## Multiple R-squared:  0.07617,    Adjusted R-squared:  0.05288
## F-statistic:  3.27 on 3 and 119 DF,  p-value: 0.02369
```

El coeficiente R^2 nos dice que con las variables *dummy* podemos explicar el 7.6 % de la variabilidad en el cambio de higiene del individuo según sea su afiliación musical, y el estadístico F que esta varianza es significativa. Pasamos a examinar los *coeficientes* del modelo.

Recordemos que los valores *beta* muestran el cambio en la variable respuesta provocado por el cambio de una unidad en el predictor. En este caso el cambio del predictor es de 0 a 1 y como el grupo referencia es siempre cero, los valores beta realmente nos proporcionan la diferencia relativa entre cada grupo y el grupo elegido como referencia. Así, el valor de la variable `Indide_dum` indica la diferencia en el cambio de higiene de una persona sin afiliación musical comparada con una persona a la que le gusta la música indie.

El estadístico t contrasta si estas diferencias son cero. Si es significante quiere decir que el grupo codificado con 1 es significativamente diferente del grupo de referencia. Para esta primera variable el *t-test* es significativo

y el valor *beta* negativo por lo que podemos decir que la higiene empeora de una persona sin afiliación musical a una indie.

Para la segunda variable, `metal_dum`, obtenemos un valor positivo para beta, sin embargo no es significativo por lo que podríamos decir que el cambio en la higiene a lo largo de los tres días del festival es el mismo para una persona sin afiliación musical que para una que le gusta el metal.

Volver al índice del curso

Referencias y bibliografía

Ali S. Hadi, S. C. &. (2006). *Linear models with r* (4th edition.). John Wiley & Sons. Retrieved from http://samples.sainsburysebooks.co.uk/9780470055458_sample_381725.pdf

Ferrari, D., & Head, T. (2010). Regression in r. part i: Simple linear regression. UCLA Department of Statistics Statistical Consulting Center. Retrieved October 13, 2014, from http://scc.stat.ucla.edu/page_attachments/0000/0139/reg_1.pdf

Field, A., Miles, J., & Field, Z. (2012). *Discovering statistics using r* (1st edition.). Sage Publications Ltd.

J.Faraway, J. (2009). *Linear models with r* (1st edition.). Taylor & Francis e-Library. Retrieved from http://home.ufam.edu.br/jcardoso/PPGMAT537/Linear%20Models%20with%20R.pdf

Kabacoff, R. (2014). Creating a figure arrangement with fine control. Retrieved October 13, 2014, from http://www.statmethods.net/advgraphs/layout.html

Pérez, J. L. (2014). La estadística: Una orqueta hecha instrumento. Retrieved October 13, 2014, from http://estadisticaorquestainstrumento.wordpress.com/

Sánchez, J. G. P. (2011). Regresión lineal simple. Universidad Politécnica de Madrid. Retrieved October 13, 2014, from http://ocw.upm.es/estadistica-e-investigacion-operativa/introduccion-a-la-estadistica-basica-el-diseno-de-experimer contenidos/Material-de-clase/Regresion.pdf

(SCG), S. S. C. G. (2013). Multiple linear regression (r). San Diego State University. Retrieved October 13, 2014, from http://scg.sdsu.edu/mlr-r/

SPSS. (2007). Análisis de regresión lineal: El procedimiento regresión lineal. IBM SPSS Statistics. Retrieved October 13, 2014, from http://pendientedemigracion.ucm.es/info/socivmyt/paginas/D_departamento/materiales/analisis_datosyMultivariable/18reglin_SPSS.pdf