Contents

REGRESIÓN

Axustando o modelo

O comando para facer regresión en R é

lm(Y~X1+X2+X3)

Sendo Y, X1, X2, X3 vectores contendo os valores das variables que usaremos na análise

Así usando os datos do leite tense

```
lm(leite~IMPEXAC+NMIEMB+EDADSP+TIPHOGAR8,data=gastan)
```

```
##
## Call:
## lm(formula = leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8, data = gastan)
##
## Coefficients:
## (Intercept)      IMPEXAC       NMIEMB       EDADSP   TIPHOGAR82
##  -3.554e+02    3.978e-04    7.261e+01    5.733e+00    1.736e+02
##  TIPHOGAR83   TIPHOGAR84
##   7.948e+01   -1.610e+01
```

data=gastan é unha maneira de indicar ao comando que algúns nomes de variables corresponden ao data.frame ***gastan***. Equivalen a escribir *lm(leite~gastan*$IMPEXAC$ + *gastan*$NMIEMB$+*gastan*$EDADSP$ + *gastan*$TIPHOGAR8$*)*, pero esta última maneira, ainda que funciona igualmente, complica os cálculos máis adiante, cando se pretenda facer predición coa función *predict*.

As expresións do tipo ***Y~X1+x2+x3*** denóminanse fórmulas en R, e son maneiras de representar formulas matemáticas dentro dos comandos de R. Neste caso representan $\boldsymbol{Y=b0+b1 \cdot X1+b2 \cdot X2+b3 \cdot X3}$. Exemplos de outras posibilidade pódense ver nesta web:

pdf:formulas en R: https://www.u-cursos.cl/ingenieria/2011/2/MA5501/1/material_docente/bajar?id_material=402148

Así usando unha expresión apropiada como fórmula podemos colocar un data.frame na parte explicativa:

lm(Y~.,varX) sendo varX un data.frame que contén todas as variables explicativas.

```
lm(leite~.,gastan[c(15,7,13,12)])
```

```
##
## Call:
## lm(formula = leite ~ ., data = gastan[c(15, 7, 13, 12)])
##
## Coefficients:
## (Intercept)      IMPEXAC       NMIEMB       EDADSP   TIPHOGAR82
##  -3.554e+02    3.978e-04    7.261e+01    5.733e+00    1.736e+02
##  TIPHOGAR83   TIPHOGAR84
##   7.948e+01   -1.610e+01
```

Correlación

Correlación simple

O comando é:

cor()

Que se pode aplicar a dous vectores numéricos ou a un data.frame ou unha matriz con 2 ou máis columnas. Nese caso o resultado é unha matriz de correlacións:

```
cor(leite,gastan[15])
```

```
##        IMPEXAC
## [1,] 0.1778043
```

```
auxi=data.frame(leite,gastan[c(15,7,13)]) #TIPHOGAR8 non se inclue por ser factor
cor(auxi)
```

```
##              leite    IMPEXAC     NMIEMB      EDADSP
## leite   1.00000000  0.1778043  0.2583703  0.03078582
## IMPEXAC 0.17780432  1.0000000  0.4067172 -0.16118401
## NMIEMB  0.25837026  0.4067172  1.0000000 -0.44132494
## EDADSP  0.03078582 -0.1611840 -0.4413249  1.00000000
```

```
#pódese redondear para mellorar a visualización
round(cor(auxi),2)
```

```
##         leite IMPEXAC NMIEMB EDADSP
## leite    1.00    0.18   0.26   0.03
## IMPEXAC  0.18    1.00   0.41  -0.16
## NMIEMB   0.26    0.41   1.00  -0.44
## EDADSP   0.03   -0.16  -0.44   1.00
```

Para contrastar a significación deste coeficiente pódese usar o comando *cor.test()*:

```
#só se usa con vectores non con unha matriz
cor.test(leite,gastan$IMPEXAC)
```

```
##
##  Pearson's product-moment correlation
##
## data:  leite and gastan$IMPEXAC
## t = 1.6461, df = 83, p-value = 0.1035
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0367109  0.3766553
## sample estimates:
##       cor
## 0.1778043
```

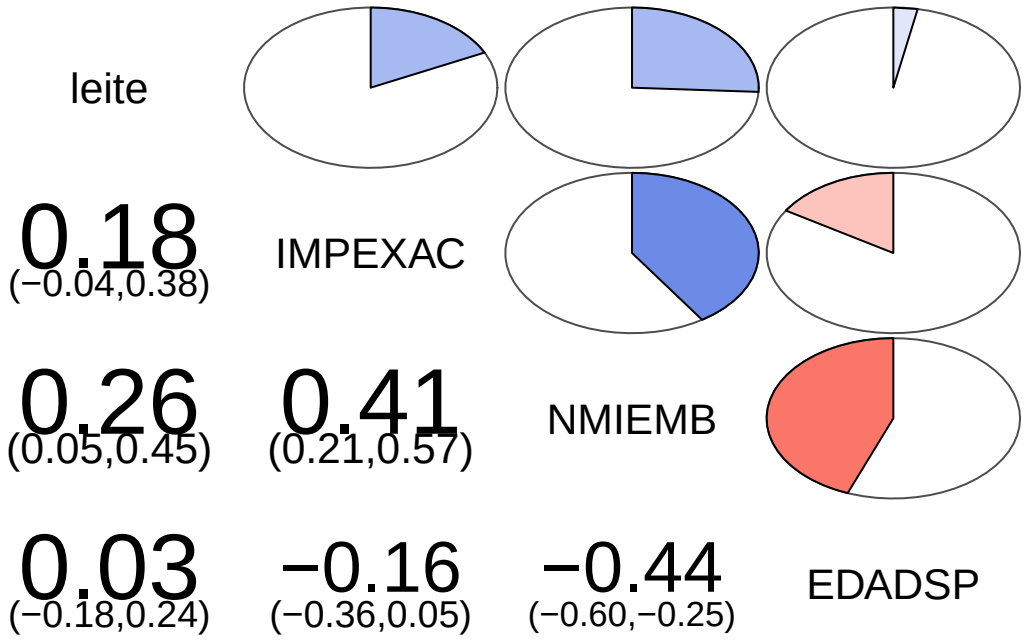
Unha versión dese comando que permite facer os contrastes de correlación por matrices está no paquete ***psych***, *e denomínase* corr.test*

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

```
## Call:corr.test(x = auxi, use = "complete", method = "pearson")
## Correlation matrix
##         leite IMPEXAC NMIEMB EDADSP
## leite    1.00    0.18   0.26   0.03
## IMPEXAC  0.18    1.00   0.41  -0.16
## NMIEMB   0.26    0.41   1.00  -0.44
## EDADSP   0.03   -0.16  -0.44   1.00
## Sample Size
## [1] 85
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         leite IMPEXAC NMIEMB EDADSP
## leite    0.00    0.31   0.07   0.78
## IMPEXAC  0.10    0.00   0.00   0.31
## NMIEMB   0.02    0.00   0.00   0.00
## EDADSP   0.78    0.14   0.00   0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Unha versión gráfica (*correlograma*) pode realizarse cos paquetes ***corrgram*** ou ***corrplot***

```
library(corrgram)
corrgram(auxi, lower.panel=panel.conf, upper.panel=panel.pie )
```



Correlación parcial

Para calcular a correlación parcial mediante un comando é necesario instalar o paquete ***ppcor***, xa que non ven no R básico

Despois sería simplemente cargar o paquete e usar o comando *pcor()*

```
#cargar o paquete
library(ppcor)
```

```
## Loading required package: MASS
```

```
#O comando pcor calcula unha matriz de correlacións parciais:
pcor(auxi)
```

```
## $estimate
##              leite     IMPEXAC     NMIEMB       EDADSP
## leite   1.00000000 0.079809628  0.2562175  0.165813422
## IMPEXAC 0.07980963 1.000000000  0.3446607  0.008722968
## NMIEMB  0.25621746 0.344660726  1.0000000 -0.439770873
## EDADSP  0.16581342 0.008722968 -0.4397709  1.000000000
##
## $p.value
##              leite     IMPEXAC       NMIEMB       EDADSP
## leite   0.00000000 0.473239222 1.938598e-02 1.341036e-01
## IMPEXAC 0.47323922 0.000000000 1.419006e-03 9.376162e-01
## NMIEMB  0.01938598 0.001419006 0.000000e+00 3.185567e-05
## EDADSP  0.13410365 0.937616241 3.185567e-05 0.000000e+00
##
## $statistic
##             leite   IMPEXAC    NMIEMB     EDADSP
## leite   0.0000000 0.7205852  2.385590  1.5132688
## IMPEXAC 0.7205852 0.0000000  3.304417  0.0785097
## NMIEMB  2.3855901 3.3044170  0.000000 -4.4069633
```

```
## EDADSP  1.5132688 0.0785097 -4.406963  0.0000000
##
## $n
## [1] 85
##
## $gp
## [1] 2
##
## $method
## [1] "pearson"
```

```
#é o primeiro dos resultados producidos por ese comando.
#para obter unicamente a matriz de correrlacións parciais, sen outros resultados pódese facer:
pcor(auxi)$estimate
```

```
##              leite     IMPEXAC     NMIEMB       EDADSP
## leite   1.00000000 0.079809628  0.2562175  0.165813422
## IMPEXAC 0.07980963 1.000000000  0.3446607  0.008722968
## NMIEMB  0.25621746 0.344660726  1.0000000 -0.439770873
## EDADSP  0.16581342 0.008722968 -0.4397709  1.000000000
```

Para os coeficientes de determinación múltiple e múltiple axustado é necesario manexar *summary(lm())*, o que se verá a continuación

summary.lm

A información que proporciona a saida do comando *lm()* é escasa, polo que e necesario aplicarlle o comando xenérico *summary()* para ver máis información:

```
summary(lm(leite~gastan$IMPEXAC+gastan$NMIEMB+gastan$EDADSP+gastan$TIPHOGAR8))
```

```
##
## Call:
## lm(formula = leite ~ gastan$IMPEXAC + gastan$NMIEMB + gastan$EDADSP +
##     gastan$TIPHOGAR8)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -258.04 -108.43  -28.81   63.59  562.65
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.554e+02  1.899e+02  -1.871  0.06508 .
## gastan$IMPEXAC     3.978e-04  1.087e-03   0.366  0.71550
## gastan$NMIEMB      7.261e+01  2.649e+01   2.741  0.00758 **
## gastan$EDADSP      5.733e+00  2.386e+00   2.403  0.01863 *
## gastan$TIPHOGAR82  1.736e+02  8.086e+01   2.147  0.03489 *
## gastan$TIPHOGAR83  7.948e+01  9.747e+01   0.816  0.41727
## gastan$TIPHOGAR84 -1.610e+01  7.158e+01  -0.225  0.82263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 174 on 78 degrees of freedom
## Multiple R-squared:  0.1596, Adjusted R-squared:  0.09491
## F-statistic: 2.468 on 6 and 78 DF,  p-value: 0.0308
```

Deste xeito aparecen non só os parámetros estimados, senón que aparecen os seus erros típicos, os seus estatísticos t e os p-valores asociados, a desviación típica dos erros (*Residual standard error*), os R^2 múltiple e múltiple axustado, e o resultado do contraste F co seu p-valor.

Ademais, como calquera cousa en R, os resultados proporcionados por *summary(lm())* son un obxecto, e nel gardan máis información e resultados ca os que se ven na súa saida.

Podemos gardar o resultado nun obxecto *regresion* e analizar algúns dos elementos que o compoñen:

```
regresion=summary(lm(leite~IMPEXAC+NMIEMB+EDADSP+TIPHOGAR8,data=gastan))
#vemos a súa saida
regresion
```

```
##
## Call:
## lm(formula = leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8, data = gastan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -258.04 -108.43  -28.81   63.59  562.65
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.554e+02  1.899e+02  -1.871  0.06508 .
## IMPEXAC      3.978e-04  1.087e-03   0.366  0.71550
## NMIEMB       7.261e+01  2.649e+01   2.741  0.00758 **
## EDADSP       5.733e+00  2.386e+00   2.403  0.01863 *
## TIPHOGAR82   1.736e+02  8.086e+01   2.147  0.03489 *
## TIPHOGAR83   7.948e+01  9.747e+01   0.816  0.41727
## TIPHOGAR84  -1.610e+01  7.158e+01  -0.225  0.82263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 174 on 78 degrees of freedom
## Multiple R-squared:  0.1596, Adjusted R-squared:  0.09491
## F-statistic: 2.468 on 6 and 78 DF,  p-value: 0.0308
```

```
#como tipo de obxecto será?:
mode(regresion)
```

```
## [1] "list"
```

```
#Ao ser unha lista saberemos que é unha colección de elementos de diferentes tipos.
#Ademais de ser unha lista tamén é de clase "summary.lm"
class(regresion)
```

```
## [1] "summary.lm"
```

```r
#o que indica o tipo de estructura que presenta.
#Pódese analizar con máis detalle esa estructura. Por exemplo, cantos elementos ten?
length(regresion)
```

```
## [1] 11
```

```r
#como se chaman?
names(regresion)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
#ou tamén
attributes(regresion)
```

```
## $names
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
##
## $class
## [1] "summary.lm"
```

```r
#cal é a súa estructura?
str(regresion)
```

```
## List of 11
##  $ call         : language lm(formula = leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8, data = gastan
##  $ terms        :Classes 'terms', 'formula' length 3 leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8
##   .. ..- attr(*, "variables")= language list(leite, IMPEXAC, NMIEMB, EDADSP, TIPHOGAR8)
##   .. ..- attr(*, "factors")= int [1:5, 1:4] 0 1 0 0 0 0 0 1 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:5] "leite" "IMPEXAC" "NMIEMB" "EDADSP" ...
##   .. .. .. ..$ : chr [1:4] "IMPEXAC" "NMIEMB" "EDADSP" "TIPHOGAR8"
##   .. ..- attr(*, "term.labels")= chr [1:4] "IMPEXAC" "NMIEMB" "EDADSP" "TIPHOGAR8"
##   .. ..- attr(*, "order")= int [1:4] 1 1 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(leite, IMPEXAC, NMIEMB, EDADSP, TIPHOGAR8)
##   .. ..- attr(*, "dataClasses")= Named chr [1:5] "numeric" "numeric" "numeric" "numeric" ...
##   .. .. ..- attr(*, "names")= chr [1:5] "leite" "IMPEXAC" "NMIEMB" "EDADSP" ...
##  $ residuals    : Named num [1:85] -77.6 -110.2 -102.1 -241.3 -60.3 ...
##   ..- attr(*, "names")= chr [1:85] "1" "3" "4" "5" ...
##  $ coefficients : num [1:7, 1:4] -3.55e+02 3.98e-04 7.26e+01 5.73 1.74e+02 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "(Intercept)" "IMPEXAC" "NMIEMB" "EDADSP" ...
##   .. ..$ : chr [1:4] "Estimate" "Std. Error" "t value" "Pr(>|t|)"
##  $ aliased      : Named logi [1:7] FALSE FALSE FALSE FALSE FALSE FALSE ...
##   ..- attr(*, "names")= chr [1:7] "(Intercept)" "IMPEXAC" "NMIEMB" "EDADSP" ...
```

```
##  $ sigma        : num 174
##  $ df           : int [1:3] 7 78 7
##  $ r.squared    : num 0.16
##  $ adj.r.squared: num 0.0949
##  $ fstatistic   : Named num [1:3] 2.47 6 78
##   ..- attr(*, "names")= chr [1:3] "value" "numdf" "dendf"
##  $ cov.unscaled : num [1:7, 1:7] 1.19 4.38e-07 -7.26e-02 -1.44e-02 -3.59e-01 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "(Intercept)" "IMPEXAC" "NMIEMB" "EDADSP" ...
##   .. ..$ : chr [1:7] "(Intercept)" "IMPEXAC" "NMIEMB" "EDADSP" ...
##  - attr(*, "class")= chr "summary.lm"
```

```
#A estructura é algo complexa, pero mirando os nomes podemos informarnos na axuda de R do que é cada un
```

Por exemplo:

- *call*: É a fórmula empregada para producir o obxecto *regresión*:

```
#como regresión é unha lista podemos ver o que hai detras de cada nome usando un $ como separador
regresion$call
```

```
## lm(formula = leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8, data = gastan)
```

- *residuos*: residuos da regresión

```
#por cuestion de espacio só se sacan os 6 primeiros
regresion$residuals
```

```
##          1          3          4          5          6          7
##  -77.55970 -110.16932 -102.06534 -241.30170  -60.29696   20.77428
```

- *coefficients*: coeficientes da regresión

```
#por cuestion de espacio só se sacan os 6 primeiros
regresion$coefficients
```

.....

Dado que regresión é unha lista tamén podemos obter os seus elementos polo seu número de orde:

```
regresion[[1]]
```

```
## lm(formula = leite ~ IMPEXAC + NMIEMB + EDADSP + TIPHOGAR8, data = gastan)
```

é o primeiro elemento, que se corresponde con *regresion$call*

Se en lugar de gardar *summary(lm)* (ou *summary.lm* a forma correcta de escribilo na axuda de R) gardásemos un obxecto *lm* cambiarían alguns dos seus elementos:

```
regresion0=lm(leite~IMPEXAC+NMIEMB+EDADSP+TIPHOGAR8,data=gastan)

attributes(regresion0)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "contrasts"     "xlevels"       "call"          "terms"
## [13] "model"
##
## $class
## [1] "lm"
```

Funcións de R asociadas á regresión

- fitted(regresion0): valores axustados, equivale a lm1$fitted.values
- resid(regresion0): residuos, equivale a regresion0$residuals
- coef(regresion0) coeficientes, equivale a regresion0$coefficients
- confint(regresion0): intervalos de confianza para os coeficientes
- vcov(regresion0): taboa de varianza-covarianza
- logLik(regresion0): log likelihood
- AIC(regresion0): criterio de información de Akaike
- anova(regresion0): significatividade da influencia das variables

predición

A función *predict* permite facer predición para diferentes tipos de modelos, entre eles as regresións.

> predict(obxecto co modelo, obxecto cos valores a predicir)

Cando se pon unicamente o obxecto co modelo a predición faise para todos os valores usados na súa estimación:

```
predict(regresion0)
```

```
##         1         3         4         5         6         7         8
## 224.76143 112.83381 433.90212 242.81124 171.66615 178.44511 189.97850
##         9        10        12        14        15        16        18
## 244.56627 168.68910 132.98868 225.18840 205.30855 241.13593 240.43239
##        19        20        21        22        23        24        25
## 179.11987 170.63534 208.28511 214.50761 236.35244 272.20322 222.49497
##        26        27        28        30        31        32        33
## 292.53998 236.83689 291.21321 186.50364 239.62225 189.44045 190.66112
##        34        35        36        37        38        39        40
## 173.02380 261.09591 279.86598 451.52422  93.65288 266.03571 300.42029
##        41        42        43        44        45        46        47
## 138.96064 131.69478 229.75912 229.05550 233.31376 158.19279 241.95323
##        48        49        50        51        52        53        54
## 236.52069 359.32257 338.53926 175.87353 217.78730 277.51113 240.90639
##        55        56        57        59        60        61        62
## 189.27122 269.66118 167.94687 202.61621 239.30242 215.78762 131.69744
```

```
##        64        65        67        68        69        70        71
## 266.70878 275.45947  95.57185 157.25839 248.60174 170.79900 121.57462
##        72        73        75        76        77        78        79
## 375.46224 168.78934 236.93018 192.86108 292.11063 255.48259 220.36040
##        80        81        82        84        86        88        89
## 106.27020 381.03760 422.73370 192.68645 271.84618 101.13328 172.68823
##        90        91        92        93        97        98        99
## 189.86297 185.44023 245.40326 104.46547 259.98721 188.43727 165.45606
##       100
## 195.40356
```

Para predicir novos valores é necesaro incluir un obxecto novo cunha estructura semellante ao empregado na regresión, pero incluindo os valores para os que se queira predicir.

```
#crear unha estructura semellante á usada na regresión:
str(gastan[c(15,7,13,12)])
```

```
## 'data.frame':    85 obs. of  4 variables:
##  $ IMPEXAC  : int  26796 16500 35880 14520 14124 32472 32640 10020 15552 7440 ...
##  $ NMIEMB   : int  2 2 6 2 1 3 3 1 2 1 ...
##  $ EDADSP   : int  74 58 62 78 48 39 41 61 65 72 ...
##  $ TIPHOGAR8: Factor w/ 4 levels "1","2","3","4": 1 4 4 1 2 3 3 2 1 1 ...
```

```
#cos valores que se queiran predicir
novo=data.frame(IMPEXAC=20000,NMIEMB=5,EDADSP=45,TIPHOGAR8="1")
#vexamos:
novo
```

```
##   IMPEXAC NMIEMB EDADSP TIPHOGAR8
## 1   20000      5     45         1
```

```
#a predicion será:
predict(regresion0,novo)
```

```
##        1
## 273.6123
```

Con máis datos:

```
#crear unha estructura semellante á usada na regresión:
#cos valores que se queiran predicir
novo=data.frame(IMPEXAC=c(20000,12000,40000),NMIEMB=c(5,6,4),EDADSP=c(45,41,36)
                ,TIPHOGAR8=factor(c("1","3","4")))
predict(regresion0,novo)
```

```
##        1        2        3
## 273.6123 399.5875 141.2627
```

Hai que ter coidado con que se manteñan os tipos de dato de cada columna.

Exemplos de regresión

Nesta parte vaise usar R para replicar algúns dos exemplos que aparecen no libro **Introductory Econometrics: A Modern Approach**, 5th Edition, de Jeffrey M. Wooldridge. En particular os exemplos dos temas 2 a 7 que empregan os datos *wage1*, que podedes atopar en formato de datos para Stata nos enderezos

- https://ideas.repec.org/p/boc/bocins/wage1.html
- https://ideas.repec.org/p/boc/bocins/wage2.html

Para ler datos en formato de Stata é necesario ter cargada (e instalada) o paquete *foreign*, e empregar a súa función *read.dta()*. Previamente sería conveniente ter colocados os ficheiros no cartafol de traballo:

```
library(foreign)
wage1=read.dta("datos/wage1.dta") #debedes adaptar aos vosos datos
wage2=read.dta("datos/wage2.dta")
```

Tema 2

- **Exemplo 2.4**: *regresión simple - (Wage and Education)*

Para unha poboación de xente traballando en 1976, considera a variable $\boldsymbol{y}$ o salario (*wage*) en dólares por hora, e $\boldsymbol{x}$ a educación (*educ*), indicada polos anos de escolarización. O tamaño da mostra é n=526.

Con eses datos facemos a regresión simple:

```
lm(wage~educ,data=wage1)
```

```
##
## Call:
## lm(formula = wage ~ educ, data = wage1)
##
## Coefficients:
## (Intercept)         educ
##     -0.9049       0.5414
```

Obtense a regresión $\widehat{wage} = -0.9 + 0.5414\ educ$

Para unha persoa **con 8 anos de escolaridade** o salario predito é:

```
novo=data.frame(educ=c(8))
predict(lm(wage~educ,data=wage1),novo)
```

```
##        1
## 3.426022
```

Ou sexa 3.43 dólares por hora.

Cada ano máis de escolaridade incrementarase o salario en:

```
coeficientes=coefficients(lm(wage~educ,data=wage1))
coeficientes[2]
```

```
##      educ
## 0.5413593
```

Ou sexa 0.5414 dolares por hora. Pol tanto 4 anos produciron un incremento de:

```
4*coeficientes[2]
```

```
##     educ
## 2.165437
```

que son 13.704 dólares por hora.

- **Exemplo 2.7**: *regresión simple - (Wage and Education)*

O salario medio da mostra é 5.9, e o nivel medio de educación 12.56.

Se sustituimos a educación media na recta de regresión obténse o salario medio:

$-0.9049 + 0.5414 \cdot 12.56 = 5.9$

O código empregado é:

```
sal.medio=mean(wage1$educ)
round(sal.medio,2)
round(mean(wage1$educ),2)

novo=data.frame(educ=c(sal.medio))
round(predict(lm(wage~educ,data=wage1),novo),2)
```

- **Exemplo 2.10**: *regresión simple e correlación - (A Log Wage Equation)*

Repetindo o modelo anterior pero para o logaritmo dos salarios:

```
lm(log(wage)~educ,data=wage1)
```

```
##
## Call:
## lm(formula = log(wage) ~ educ, data = wage1)
##
## Coefficients:
## (Intercept)         educ
##     0.58377      0.08274
```

A regresión é agora $\widehat{log(wage)} = -0.9 + 0.5414\ educ$

Ademais:

```
auxi=lm(log(wage)~educ,data=wage1)
length(auxi$model$educ) #nº de elementos da variable educ incluida no modelo
```

```
## [1] 526
```

```
#agora sacar do summary.lm o valor que corresponde a R cadrado
#redondeado a tres decimais
round(summary(auxi)$r.squared,3)
```

```
## [1] 0.186
```

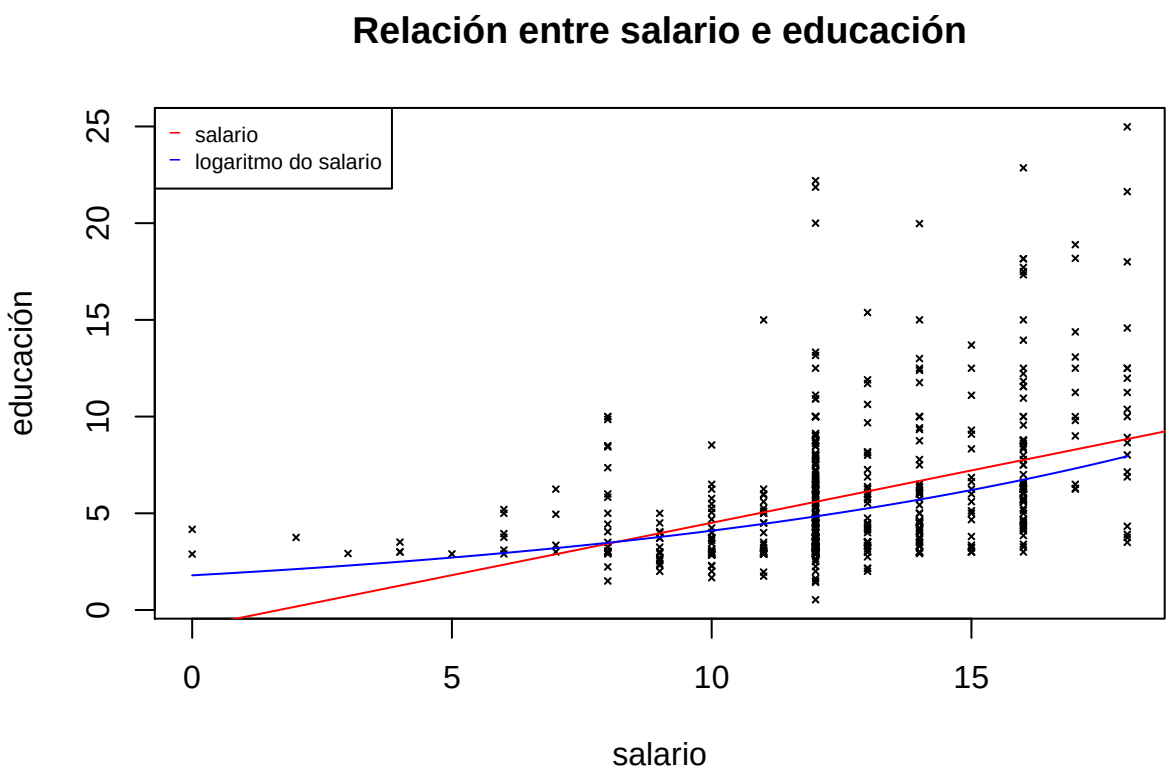
N = 526 e $R^2 = 0.186$

- **Unhas gráficas**

```
plot(wage1$educ,wage1$wage,main="Relación entre salario e educación",xlab="salario",
     ylab="educación",pch=4,cex=0.4)
abline(lm(wage~educ,data=wage1),col="red")

X=seq(min(wage1$educ),max(wage1$educ),by = 0.01)
Y=predict(auxi,data.frame(educ=X))
lines(X,exp(Y),col="blue")

legend("topleft",
legend=c("salario", "logaritmo do salario"),
cex=.7,pch=45,
col=c("red","blue"),
ncol=1)
```



“‘

Tema 3

- **Exemplo 3.2**: *regresión múltiple - (Hourly Wage Equation)*

Neste exemplo explícase o logaritmo do salario *log(wage)* mediante os anos de educación (*educ*), anos de experiencia no mercado de traballo (*exper*) e anos co actual empregador (*tenure*).

A ecuación obtida é:

Obtense a regresión $\widehat{log(wage)} = 0.284 + 0.092\ educ + 0.0041\ exper + 0.022\ tenure$

Neste caso pode dicirse que un incremento dun ano de escolaridade tradúcese nun aumento do 9.2% no salario, e igual para as demais variables.

```
round(100*coeficientes[2],1) #educ
```

```
## educ
##  9.2
```

```
round(100*coeficientes[3],2)#exper
```

```
## exper
##  0.41
```

```
round(100*coeficientes[4],1)#tenure
```

```
## tenure
##    2.2
```

- **Exemplos 3.4 e 3.5**: *regresión múltiple e R^2 - (adaptados para salarios)*

Os coeficientes R^2 poden obterse aplicando *summary()* ao comando *lm()*, por exemplo, para explicar o *logaritmo dos salarios* mediante *educación* e *experiencia*:

```
summary(lm(log(wage)~educ+exper,data=wage1))
```

```
##
## Call:
## lm(formula = log(wage) ~ educ + exper, data = wage1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05800 -0.30136 -0.04539  0.30601  1.44425
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.216854   0.108595   1.997   0.0464 *
## educ        0.097936   0.007622  12.848  < 2e-16 ***
## exper       0.010347   0.001555   6.653 7.24e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4614 on 523 degrees of freedom
## Multiple R-squared:  0.2493, Adjusted R-squared:  0.2465
## F-statistic: 86.86 on 2 and 523 DF,  p-value: < 2.2e-16
```

Aparte de cuestións de inferencia, tamén proporciona o R^2 e o R^2 axustado. Podemos saber os seus valores aí, pero tamén podemos separalos do conxunto, lembrando simplemente que o resultado de *summary.lm()* é unha lista, na cal os valores dos R^2 terán cadanseu nome: *r.squared* e *adj.r.squared*

```
#por comodidade, creamos un obxecto
auxi=summary(lm(log(wage)~educ+exper,data=wage1))
#R2
auxi$r.squared
```

```
## [1] 0.2493434
```

```
#R2 axustado
auxi$adj.r.squared
```

```
## [1] 0.2464728
```

Estimamos agora o modelo con diferente número de variables e vemos o efecto nos R^2. Con educación e experiencia como variables explicativas tense:

$\widehat{log(wage)} = 0.217 + 0.098\ educ + 0.0103\ exper$; ademais $R^2 = 0.249$ e o $R^2 axustado = 0.246$

Se engadimos *tenure* no modelo obtense:

```
#por comodidade, creamos un obxecto
auxi=summary(lm(log(wage)~educ+exper+tenure,data=wage1))
#R2
auxi$r.squared
```

```
## [1] 0.3160133
```

```
#R2 axustado
auxi$adj.r.squared
```

```
## [1] 0.3120824
```

e polo tanto:

$\widehat{log(wage)} = 0.284 + 0.092\ educ + 0.0041\ exper + 0.022\ tenure$;

con $R^2 = 0.316$ e $R^2 axustado = 0.312$

Tema 4

- **Exemplo 4.1**: *erros estandar - (Hourly Wage Equation)*

```
auxi=summary(lm(log(wage)~educ+exper+tenure,data=wage1))
auxi$coefficients  #en summary.lm() 'coefficientes inclue máis cousas'
```

```
##                Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 0.284359555 0.104190378  2.729230 6.562462e-03
## educ        0.092028987 0.007329923 12.555246 8.824204e-32
## exper       0.004121109 0.001723277  2.391437 1.713562e-02
## tenure      0.022067217 0.003093649  7.133070 3.294408e-12
```

$$\widehat{log(wage)} = \underset{(0.104)}{0.284} + \underset{(0.007)}{0.092}\ educ + \underset{(0.0017)}{0.0041}\ exper + \underset{(0.104)}{0.022}\ tenure;$$

- *adaptado doutros exemplos deste tema*

```
#En particular podemos obter os coeficientes como unha columna de auxi$coefficients
auxi$coefficients[,1]#auxi$coefficients ten estructura de matriz
```

```
## (Intercept)        educ       exper      tenure
## 0.284359555 0.092028987 0.004121109 0.022067217
```

```
#Os erros estandar dos coeficientes serían a segunda columna
auxi$coefficients[,2]#auxi$coefficients ten estructura de matriz
```

```
## (Intercept)        educ       exper      tenure
## 0.104190378 0.007329923 0.001723277 0.003093649
```

Tamén se poden obter os intervalos de confianza:

```
auxi0=lm(log(wage)~educ+exper+tenure,data=wage1)
confint(auxi0)
```

```
##                    2.5 %     97.5 %
## (Intercept) 0.0796755842 0.48904353
## educ        0.0776292137 0.10642876
## exper       0.0007356983 0.00750652
## tenure      0.0159896850 0.02814475
```

```
#pódese especificar un único intervalo
confint(auxi0,"educ")
```

```
##           2.5 %    97.5 %
## educ 0.07762921 0.1064288
```

```
#Ou diferentes niveis de confianza (1-nivel de significación)
confint(auxi0,"educ",level = 0.99)
```

```
##           0.5 %    99.5 %
## educ 0.07307908 0.1109789
```

Tema 7

- **Exemplo 7.1**: *variable "dummy" - (Hourly Wage Equation)*

Introdúcese sexo como variable explicativa dicotómica, que nos datos aparece como variable *female*:

```
summary(lm(wage~female+educ+exper+tenure,data=wage1))
```

```
##
## Call:
## lm(formula = wage ~ female + educ + exper + tenure, data = wage1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7675 -1.8080 -0.4229  1.0467 14.0075
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.56794    0.72455  -2.164   0.0309 *
## female      -1.81085    0.26483  -6.838 2.26e-11 ***
## educ         0.57150    0.04934  11.584  < 2e-16 ***
## exper        0.02540    0.01157   2.195   0.0286 *
## tenure       0.14101    0.02116   6.663 6.83e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.958 on 521 degrees of freedom
## Multiple R-squared:  0.3635, Adjusted R-squared:  0.3587
## F-statistic:  74.4 on 4 and 521 DF,  p-value: < 2.2e-16
```

```
summary(lm(wage~female,data=wage1))
```

```
##
## Call:
## lm(formula = wage ~ female, data = wage1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5995 -1.8495 -0.9877  1.4260 17.8805
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.0995     0.2100  33.806  < 2e-16 ***
## female       -2.5118     0.3034  -8.279 1.04e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.476 on 524 degrees of freedom
## Multiple R-squared:  0.1157, Adjusted R-squared:  0.114
## F-statistic: 68.54 on 1 and 524 DF,  p-value: 1.042e-15
```

- **Exemplo 7.5**: *variable "dummy" e regresión polinómica - (Hourly Wage Equation)*

O mesmo ca o exemplo anterior pero explicando o logaritmo do salario, e empregando tamén como explicativas os cadrados de *exper* e *tenure*:

```
summary(lm(log(wage)~female+educ+exper+I(exper^2)+tenure+I(tenure^2),data=wage1))
```

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

```
## lm(formula = log(wage) ~ female + educ + exper + I(exper^2) +
##     tenure + I(tenure^2), data = wage1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83160 -0.25658 -0.02126  0.25500  1.13370
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4166910  0.0989279   4.212 2.98e-05 ***
## female      -0.2965110  0.0358054  -8.281 1.04e-15 ***
## educ         0.0801966  0.0067573  11.868  < 2e-16 ***
## exper        0.0294324  0.0049752   5.916 6.00e-09 ***
## I(exper^2)  -0.0005827  0.0001073  -5.431 8.65e-08 ***
## tenure       0.0317139  0.0068452   4.633 4.56e-06 ***
## I(tenure^2) -0.0005852  0.0002347  -2.493    0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3998 on 519 degrees of freedom
## Multiple R-squared:  0.4408, Adjusted R-squared:  0.4343
## F-statistic: 68.18 on 6 and 519 DF,  p-value: < 2.2e-16
```

Para facer regresión polinómica débese incluir a expresión dentro de *I()* para que R o identifique como un elemento máis do modelo.

O modelo dinos agora que $\widehat{log(wage_F)} - \widehat{log(wage_M)} = -0.296511$

Polo tanto exponenciando e extraendo 1:

$$\frac{\widehat{log(wage_F)} - \widehat{log(wage_M)}}{\widehat{log(wage_M)}} = exp(-0.296511) - 1 \approx -0.2565925$$

- **Exemplo 7.6**: *variables "dummy" e regresión polinómica - (Hourly Wage Equation)*

Neste modelo estima diferencias de salarios entre grupos: homes (casados e solteiros), mulleres (casadas e solteiras).

Para facelo selecciona un grupo base, os homes solteiros, e estima a diferencia nos demais grupos con respecto a eles. Crea para esas estimacións as seguinte variables: marrmale (vale 1 para os homes casados), marrfem (vale 1 para as mulleres casadas), e singfem (vale 1 para as mulleres solteiras).

Os seguintes códigos, deste exemplo e o seguinte, proceden de https://econometricswithr.wordpress.com/wooldridge-2013/wooldridge-chapter-7/

```
# Example 7.6
  # Generate the subgroup dummies
#os seguintes comandos fan o seguinte:
#aplican unha condición: wage1$female==0 & wage1$married==1
#que produce un vector lóxico (TRUE e FALSE)
#as.numeric() transforma eses TRUE FALSE en 1 e 0, para traballar cun vector numérico
marrmale<-as.numeric(wage1$female==0 & wage1$married==1)
marrfem<-as.numeric(wage1$female==1 & wage1$married==1)
singfem<-as.numeric(wage1$female==1 & wage1$married==0)
lm.7.6.1<-lm(lwage ~ marrmale + marrfem + singfem + educ + exper +
```

```
            expersq + tenure + tenursq, data=wage1)
summary(lm.7.6.1)
```

```
##
## Call:
## lm(formula = lwage ~ marrmale + marrfem + singfem + educ + exper +
##     expersq + tenure + tenursq, data = wage1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89697 -0.24060 -0.02689  0.23144  1.09197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3213780  0.1000090   3.213 0.001393 **
## marrmale     0.2126756  0.0553572   3.842 0.000137 ***
## marrfem     -0.1982676  0.0578355  -3.428 0.000656 ***
## singfem     -0.1103502  0.0557421  -1.980 0.048272 *
## educ         0.0789103  0.0066945  11.787  < 2e-16 ***
## exper        0.0268006  0.0052428   5.112 4.50e-07 ***
## expersq     -0.0005352  0.0001104  -4.847 1.66e-06 ***
## tenure       0.0290875  0.0067620   4.302 2.03e-05 ***
## tenursq     -0.0005331  0.0002312  -2.306 0.021531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3933 on 517 degrees of freedom
## Multiple R-squared:  0.4609, Adjusted R-squared:  0.4525
## F-statistic: 55.25 on 8 and 517 DF,  p-value: < 2.2e-16
```

Neste mesmo exemplo cambia o grupo base, seleccionando mulleres casadas como nova referencia:

```
singmale<-as.numeric(wage1$female==0)*as.numeric(wage1$married==0)
lm.7.6.2<-lm(lwage ~ marrmale + singmale + singfem + educ + exper +
            expersq + tenure + tenursq, data=wage1)
summary(lm.7.6.2)
```

```
##
## Call:
## lm(formula = lwage ~ marrmale + singmale + singfem + educ + exper +
##     expersq + tenure + tenursq, data = wage1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89697 -0.24060 -0.02689  0.23144  1.09197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1231104  0.1057937   1.164 0.245089
## marrmale     0.4109433  0.0457709   8.978  < 2e-16 ***
## singmale     0.1982676  0.0578355   3.428 0.000656 ***
## singfem      0.0879174  0.0523481   1.679 0.093664 .
```

```
## educ         0.0789103  0.0066945  11.787  < 2e-16 ***
## exper        0.0268006  0.0052428   5.112 4.50e-07 ***
## expersq     -0.0005352  0.0001104  -4.847 1.66e-06 ***
## tenure       0.0290875  0.0067620   4.302 2.03e-05 ***
## tenursq     -0.0005331  0.0002312  -2.306 0.021531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3933 on 517 degrees of freedom
## Multiple R-squared:  0.4609, Adjusted R-squared:  0.4525
## F-statistic: 55.25 on 8 and 517 DF,  p-value: < 2.2e-16
```

Unha colección máis completa destes exemplos resoltos con R aparecen no web: **R-Econometrics**, cunha sección adicada a resover a maioría dos exemplos que aparecen en Wooldridge (2013), e aos que se pode acceder en: https://econometricswithr.wordpress.com/wooldridge-2013