

# SIMPLE AND MULTIPLE LINEAR REGRESSION

# TRABAJO EN EL LABORATORIO

(Rafael Alcalá)

---

## Bibliography:

Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani  
**An Introduction to Statistical Learning** with Applications in R  
Springer, 2013

## Chapter 03

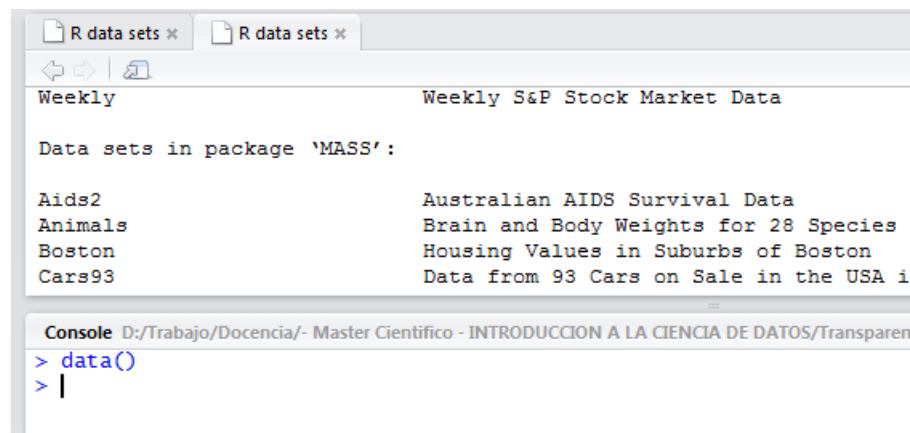
# Outline

- Regresión lineal simple
  - Visualización de los datos – Variables candidatas
  - Obtención de los modelos
  - Visualización de resultados
  - Predicción con nuevos valores y cálculo del RMSE o MSE
- Regresión lineal múltiple
  - Obtención de modelos añadiendo otra variable – Visualización de pares de variables
  - Obtención de los modelos – Todas las variables
  - Eliminar variables innecesarias
  - Interacciones
  - Predicción con nuevos valores y cálculo del RMSE o MSE
- Lectura de datos en formato KEEL
- TAREA – Conjunto California

# Uso de Datasets Pre-instalados

La función *data* nos informa sobre los conjuntos de datos previamente instalados:

`> data()`



```
R data sets x R data sets x
Weekly Weekly S&P Stock Market Data

Data sets in package 'MASS':

Aids2 Australian AIDS Survival Data
Animals Brain and Body Weights for 28 Species
Boston Housing Values in Suburbs of Boston
Cars93 Data from 93 Cars on Sale in the USA i

Console D:/Trabajo/Docencia/- Master Cientifico - INTRODUCCION A LA CIENCIA DE DATOS/Transparen
> data()
> |
```

- El paquete MASS junto con el paquete ISLR (*Data for An Introduction to Statistical Learning... BOOK*) proporcionan un buen número de conjuntos de ejemplo:

```
require(ISLR) #o install.packages("ISLR")
require(MASS)
```

- Usaremos el conjunto de datos *Boston* para esta sesión

# El Conjunto de Datos Boston

> ?Boston

> Boston #or fix(Boston) to check-edit

- The Boston data frame has 506 rows and 14 columns
- **crim** - per capita crime rate by town.
- **zn** - proportion of residential land zoned for lots over 25,000 sq.ft.
- **indus** - proportion of non-retail business acres per town.
- **chas** - Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
- **nox** - nitrogen oxides concentration (parts per 10 million).
- **rm** - average number of rooms per dwelling.
- **age** - proportion of owner-occupied units built prior to 1940.
- **dis** - weighted mean of distances to five Boston employment centres.
- **rad** - index of accessibility to radial highways.
- **tax** - full-value property-tax rate per \ \$10,000.
- **prratio** - pupil-teacher ratio by town.
- **black** -  $1000(Bk - 0.63)^2$ , where  $Bk$  is the proportion of blacks by town.
- **lstat** - lower status of the population (percent).
- **medv** - median value of owner-occupied homes in \ \$1000s.

Predict medv from others (Y=medv; Regressors=remaining attributes)

# Acceso/manejo de los datos

Boston\$lstat      Boston\$rm      Boston\$attribute

- Añadir el objeto al entorno:
  - > attach(Boston)
  - > lstat ## Acceso directo a los campos
- detach(Boston) elimina el objeto del entorno
- En esta presentación se considera por claridad que Boston no se ha añadido al entorno (el/la estudiante puede considerar añadirlo por comodidad)

# Visualización – Búsqueda de posibles relaciones

- Una variable respecto a la salida:  
    > `plot(medv~age,Boston)` #probando una a una
- Previsualización de todas las variables entre si o respecto a la salida,

```
temp <- Boston
plotY <- function (x,y) {
  plot(temp[,y]~temp[,x], xlab=paste(names(temp)[x]," X",x,sep=""),
  ylab=names(temp)[y])
}
par(mfrow=c(3,4)) #Si margin too large => (2,3)
x <- sapply(1:(dim(temp)[2]-1), plotY, dim(temp)[2])
par(mfrow=c(1,1))
```

# Visualización – Búsqueda de posibles relaciones

- Se puede afinar con las más relevantes,

```
par(mfrow=c(3,3)) #Si margin too large => (2,3)
x <- sapply(c(1, 5, 6, 7, 8, 10, 11, 12, 13), plotY, dim(temp)[2])
par(mfrow=c(1,1))
```

- Variables candidatas para el ajuste lineal simple (6 y 13):  
**rm** y **lstat**

**rm** - average number of rooms per dwelling.

**lstat** - lower status of the population (percent).

# Obtención de modelos lineales simples

- Obtención del modelo para **lstat**

```
fit1=lm(medv~lstat,data=Boston)
```

```
#or fit1=lm(Boston$medv~Boston$lstat)
```

```
fit1
```

Coefficients:

(Intercept) Boston\$lstat

34.55 -0.95

- Obtención del modelo para **rm**

```
fit2=lm(medv~rm,data=Boston)
```

```
fit2
```

Coefficients:

(Intercept) rm

-34.671 9.102



# Visualización de los resultados

## ➤ Modelo **lstat** (fit1)

```
summary(fit1)
par(mfrow=c(2,1))
plot(medv~lstat,Boston)
abline(fit1,col="red")
confint(fit1)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.168	-3.990	-1.318	2.034	24.500

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Int.)	34.55384	0.56263	61.41	<2e-16 ***
lstat	-0.95005	0.03873	-24.53	<2e-16 ***

Signif. codes:

0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: **6.216** on 504 degrees of freedom

Multiple R-squared: **0.5441**, Adjusted R-squared: 0.5432

F-statistic: 601.6 on 1 and 504 DF, p-value: < 2.2e-16

## ➤ Modelo **rm** (fit2)

```
summary(fit2)
plot(medv~rm,Boston)
abline(fit2,col="blue")
par(mfrow=c(1,1))
confint(fit2)
```

Residuals:

Min	1Q	Median	3Q	Max
-23.346	-2.547	0.090	2.986	39.433

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Int.)	-34.671	2.650	-13.08	<2e-16 ***
rm	9.102	0.419	21.72	<2e-16 ***

Signif. codes:

0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: **6.616** on 504 degrees of freedom

Multiple R-squared: **0.4835**, Adjusted R-squared: 0.4825

F-statistic: 471.8 on 1 and 504 DF, p-value: < 2.2e-16

Y el R2 ajustado. ¿Qué es?

# Acceso a la información del modelo

## Cálculo manual del error

- Centrémonos en el modelo fit1 - **lstat**

`names(fit1)`

```
[1] "coefficients" "residuals"    "effects"      "rank"         "fitted.values" "assign"
[7] "qr"           "df.residual"  "xlevels"      "call"         "terms"        "model"
```

- Cálculo manual de la raíz del ECM (RMSE)

`sqrt(sum(fit1$residuals^2)/length(fit1$residuals))`

6.203464 respecto al **6.216** que indicaba con `summary()`

`sqrt(sum(fit1$residuals^2)/(length(fit1$residuals)-2))`

6.21576 que redondeado coincide

- Ojo! Hay que tener cuidado con ésto para las comparativas (utiliza n-p en el denominador, pero en la práctica se usa directamente n para las comparativas)

# Predicción sobre nuevos datos

## Cálculo manual del error para conjuntos de test

- Predicción sobre nuevos valores

```
predict(fit1,data.frame(lstat=c(5,10,15)))  
29.80359 25.05335 20.30310
```

- Cálculo manual de la raíz del ECM (RMSE) para conjuntos de test (a modo de ejemplo se usa el propio conjunto inicial)

```
yprime=predict(fit1,data.frame(lstat=Boston$lstat))  
#o directamente #yprime=predict(fit1,Boston)  
sqrt(sum(abs(Boston$medv-yprime)^2)/length(yprime))
```

# Obtención de modelos lineales múltiples

- Obtención del modelo añadiendo variables

```
fit3=lm(medv~lstat+age,data=Boston)
summary(fit3)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	33.22276	0.73085	45.458	< 2e-16 ***
lstat	-1.03207	0.04819	-21.416	< 2e-16 ***
age	0.03454	0.01223	2.826	<u>0.00491</u> **

Residual standard error: **6.173** on 503 degrees of freedom  
 Multiple R-squared: **0.5513**, Adjusted R-squared: 0.5495  
 F-statistic: 309 on 2 and 503 DF, p-value: < 2.2e-16

Visualización de pares de variables por escala de grises:

```
temp <- Boston
plot(temp[, -dim(temp)[2]], pch=16, col=gray(1-(temp[, dim(temp)[2]]/max(temp[, dim(temp)[2]])))
```

- Probemos **lstat+rm**

```
fit4=lm(medv~lstat+rm,data=Boston)
summary(fit4)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-1.35827	3.17283	-0.428	0.669
lstat	-0.64236	0.04373	-14.689	<2e-16 ***
rm	5.09479	0.44447	11.463	<u>&lt;2e-16</u> ***

Residual standard error: **5.54** on 503 degrees of freedom  
 Multiple R-squared: **0.6386**, Adjusted R-squared: 0.6371  
 F-statistic: 444.3 on 2 and 503 DF, p-value: < 2.2e-16

# Obtención de modelos lineales múltiples

- Obtención del modelo con todas las variables

```
fit5=lm(medv~.,data=Boston)
```

```
summary(fit5)
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	3.646e+01	5.103e+00	7.144	3.28e-12	***
crim	-1.080e-01	3.286e-02	-3.287	0.001087	**
zn	4.642e-02	1.373e-02	3.382	0.000778	***
indus	2.056e-02	6.150e-02	0.334	<b>0.738288</b>	
chas	2.687e+00	8.616e-01	3.118	0.001925	**
nox	-1.777e+01	3.820e+00	-4.651	4.25e-06	***
rm	3.810e+00	4.179e-01	9.116	< 2e-16	***
age	6.922e-04	1.321e-02	0.052	<b>0.958229</b>	
dis	-1.476e+00	1.995e-01	-7.398	6.01e-13	***
rad	3.060e-01	6.635e-02	4.613	5.07e-06	***
tax	-1.233e-02	3.760e-03	-3.280	0.001112	**
ptratio	-9.527e-01	1.308e-01	-7.283	1.31e-12	***
black	9.312e-03	2.686e-03	3.467	0.000573	***
lstat	-5.248e-01	5.072e-02	-10.347	< 2e-16	***

Residual standard error: 4.745 on 492 degrees of freedom  
 Multiple R-squared: 0.7406, Adjusted R-squared: 0.7338  
 F-statistic: 108.1 on 13 and 492 DF, p-value: < 2.2e-16

- Ojo! tener en cuenta los distintos grados de libertad de modelos con distinto número de variables

# Obtención de modelos lineales múltiples

- Eliminamos las variables supuestamente no relevantes

```
fit6=lm(medv~.-age-indus,data=Boston)
summary(fit6)
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	36.341145	5.067492	7.171	2.73e-12	***
crim	-0.108413	0.032779	-3.307	<b>0.001010</b>	**
zn	0.045845	0.013523	3.390	0.000754	***
chas	2.718716	0.854240	3.183	<b>0.001551</b>	**
nox	-17.376023	3.535243	-4.915	1.21e-06	***
rm	3.801579	0.406316	9.356	< 2e-16	***
dis	-1.492711	0.185731	-8.037	6.84e-15	***
rad	0.299608	0.063402	4.726	3.00e-06	***
tax	-0.011778	0.003372	-3.493	0.000521	***
ptratio	-0.946525	0.129066	-7.334	9.24e-13	***
black	0.009291	0.002674	3.475	0.000557	***
lstat	-0.522553	0.047424	-11.019	< 2e-16	***

Residual standard error: 4.736 on 494 degrees of freedom  
 Multiple R-squared: 0.7406, Adjusted R-squared: **0.7348**  
 F-statistic: 128.2 on 11 and 494 DF, p-value: < 2.2e-16

- Mirando adjusted R2 podemos ver si el modelo mejora o empeora (se pueden aceptar pequeñas pérdidas)
- Mejora levemente respecto al anterior

# Obtención de modelos lineales múltiples

- Eliminamos más variables (**pero una a una!**)

```
fit7=lm(medv~.-age-indus-chas-crim,data=Boston)
summary(fit7)
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	35.459724	5.158054	6.875	1.87e-11	***
zn	0.041396	0.013737	3.013	<b>0.002715</b>	**
nox	-15.502932	3.583879	-4.326	1.84e-05	***
rm	3.879580	0.414180	9.367	< 2e-16	***
dis	-1.451648	0.187926	-7.725	6.26e-14	***
rad	0.252412	0.061778	4.086	5.12e-05	***
tax	-0.012360	0.003427	-3.606	0.000342	***
ptratio	-0.968703	0.131248	-7.381	6.69e-13	***
black	0.010842	0.002705	4.008	7.06e-05	***
lstat	-0.555124	0.047699	-11.638	< 2e-16	***

Residual standard error: 4.832 on 496 degrees of freedom  
 Multiple R-squared: 0.7289, Adjusted R-squared: **0.724**  
 F-statistic: 148.2 on 9 and 496 DF, p-value: < 2.2e-16

- Mirando adjusted R2 podemos ver si el modelo mejora o empeora (se pueden aceptar pequeñas pérdidas)
- Empeora levemente respecto al anterior (1% aprox.)

# Interacciones y no linealidad

- Interacciones (incluimos la función con la interacción)

```
attach(Boston)
fit8=lm(medv~lstat*rm,Boston)
summary(fit8)
plot(medv~lstat)
points(lstat,fitted(fit8),col="green",pch=20)
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -29.12452    3.34250  -8.713   <2e-16 ***
lstat         2.19398    0.20570  10.666   <2e-16 ***
rm           9.70126    0.50023  19.393   <2e-16 ***
lstat:rm     -0.48494    0.03459 -14.018   <2e-16 ***
Residual standard error: 4.701 on 502 degrees of freedom
Multiple R-squared:  0.7402,    Adjusted R-squared:  0.7387
F-statistic: 476.9 on 3 and 502 DF,  p-value: < 2.2e-16
```

- Directamente la interacción entre ambas da el mejor modelo obtenido hasta el momento



# Interacciones y no linealidad

- No linealidad (usamos la función `I` para que lo considere tal cual)

```
fit9=lm(medv~I(lstat^2),Boston)
```

```
summary(fit9)
```

```
plot(medv~lstat)
```

```
points(lstat,fitted(fit9),col="red",pch=20)
```

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept) 27.647392   0.429925   64.31  <2e-16 ***
I(lstat^2)  -0.024240   0.001359  -17.84  <2e-16 ***
Residual standard error: 7.207 on 504 degrees of freedom
Multiple R-squared:  0.3871,    Adjusted R-squared:  0.3859
F-statistic: 318.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

- No linealidad: **es necesario incluir los términos de menor grado!**

```
fit9=lm(medv~lstat +I(lstat^2),Boston)
```

```
summary(fit9)
```

```
plot(medv~lstat)
```

```
points(lstat,fitted(fit9),col="red",pch=20)
```

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.862007   0.872084   49.15  <2e-16 ***
lstat       -2.332821   0.123803  -18.84  <2e-16 ***
I(lstat^2)   0.043547   0.003745   11.63  <2e-16 ***
Residual standard error: 5.524 on 503 degrees of freedom
Multiple R-squared:  0.6407,    Adjusted R-squared:  0.6393
F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

# Interacciones y no linealidad

➤ Qué pasaría con?:

```
fitprueba=lm(medv~lstat +rm +l(lstat * rm) +l(lstat^2) +l(lstat^2 * rm),Boston)
summary(fitprueba)
plot(medv~lstat)
points(lstat,fitted(fitprueba),col="red",pch=20)
```

# Predicción sobre nuevos datos

## Cálculo manual del error para conjuntos de test

- Cálculo manual de la raíz del ECM (RMSE) para conjuntos de test (a modo de ejemplo se usa el propio conjunto inicial)

```
yprime=predict(fit8,Boston)
```

```
sqrt(sum(abs(Boston$medv-yprime)^2)/length(yprime))
```

# Lectura de datasets en formato KEEL

- Keel.ugr.es -> Keel-dataset -> Regression datasets (32)
- Descargar el dataset completo "California"
- Lectura:

```
xtra <- read.csv("california.dat", comment.char="@", header = FALSE)
#head(xtra)
```

#Asignación manual

```
names(xtra) <- c("Longitude", "Latitude", "HousingMedianAge",
  "TotalRooms", "TotalBedrooms", "Population", "Households",
  "MedianIncome", "MedianHouseValue")
```

#Asignación automática, facilita el acceso a los campos

```
n <- length(names(xtra)) - 1
names(xtra)[1:n] <- paste ("X", 1:n, sep="")
names(xtra)[n+1] <- "Y"
```

TAREA (para evaluación continua):  
Reproduzca el estudio para el conjunto de datos California y suba el/los fichero/s .R a PRADO en la actividad correspondiente

Keel.ugr.es -> Keel-dataset -> Regression datasets (32)