Resumen de modelos y de instrucciones con *R*

Antonio Francisco Roldán López de Hierro[1]

Supongamos que disponemos de los datos emparejados de una variable bidimensional (X, Y):

Variable X (independiente)	x_1	x_2	x_3		x_n
Variable Y (dependiente)	y_1	y_2	y_3	$\ldots$	y_n

En las variables `x` e `y` introducimos los datos de la muestra conjunta anterior y definimos la variable `n` como el número de datos de la muestra.

```
x <- c(x1, x2, ..., xn)
y <- c(y1, y2, ..., yn)
n <- length(x)
```

[1]Profesor de la Universidad de Granada - http://www.ugr.es/~aroldan

El modelo lineal

El modelo lineal $Y = a + bX$

Este modelo se obtiene escribiendo directamente `lm(y~x)` (`lm` significa *linear model*).

Con las siguientes instrucciones se puede obtener la covarianza, los coeficientes de correlación y de determinación, y la varianza residual.

Las últimas instrucciones permiten dibujar la nube de puntos y la recta de regresión de `y` sobre `x`.

```
n <- length(x);
modelo.lineal <- lm( y ~ x );

## Coeficiente "a" del modelo LINEAL  Y = a+b*X;
a <- modelo.lineal$coefficients[[1]];a;
## Coeficiente "b" del modelo LINEAL  Y = a+b*X;
b <- modelo.lineal$coefficients[[2]];b;

cov(x,y);                        ## Covarianza;
cor(x,y);                        ## Coeficiente de correlación lineal;
cor(x,y)^2;                      ## Coeficiente de determinación lineal;
sum((y - (a+b*x))^2)/n;          ## Varianza residual;

plot(x,y, col="blue");                    ## Nube de puntos;
abline(a, b, col="red", lwd="3");         ## Recta de regresión;
##
```

El modelo lineal que pasa por el origen

El modelo lineal que pasa por el origen $Y = bX$

Este modelo se obtiene ejecutando el siguiente código:

```
n <- length(x);
modelo.lineal.simple <- lm( y ~ x-1 );

## Coeficiente "b" del modelo LINEAL SIMPLE  Y = b*X;
b <- modelo.lineal.simple$coefficients[[1]];b;

cov(x-1,y);                   ## Covarianza;
cor(x-1,y);                   ## Coeficiente de correlación;
cor(x-1,y)^2;                 ## Coeficiente de determinación;
sum((y - (b*x))^2)/n;         ## Varianza residual;

plot(x,y, col="blue");        ## Nube de puntos;
curve(b*x, col="red", add=TRUE, lwd="3");
##
```

El modelo exponencial

El modelo exponencial $Y = a \cdot b^X$

Hacemos la transformación:

$$Y = a \cdot b^X \quad \Leftrightarrow \quad \log(Y) = \log\left(a \cdot b^X\right) = \log(a) + \log(b^X) = \log(a) + \log(b) \cdot X$$

de donde

$$Y' = A + B \cdot X, \quad \text{siendo} \quad \begin{cases} Y' = \log(Y) \\ A = \log(a) \\ B = \log(b) \end{cases}$$

Por ello,

$$\begin{cases} a = \mathrm{e}^A = \exp(A) \\ b = \mathrm{e}^B = \exp(B) \end{cases}$$

y programamos:

```
n <- length(x);
log.y <- log(y);
modelo.exponencial <- lm( log.y ~ x );

## Coeficiente "a" del modelo EXPONENCIAL  Y = a*(b^X);
a <- exp(modelo.exponencial$coefficients[[1]]);a;
## Coeficiente "b" del modelo EXPONENCIAL  Y = a*(b^X);
b <- exp(modelo.exponencial$coefficients[[2]]);b;

cov(x,log.y);                   ## Covarianza;
cor(x,log.y);                   ## Coeficiente de correlación;
cor(x,log.y)^2;                 ## Coeficiente de determinación;
sum((y - (a*(b^x)))^2)/n;       ## Varianza residual;

plot(x,y, col="blue");          ## Nube de puntos;
curve( a*(b^x), col="red", add=TRUE, lwd="3");
##
```

El modelo potencial

El modelo potencial $Y = a \cdot X^b$

Hacemos la transformación:

$$Y = a \cdot X^b \quad \Leftrightarrow \quad \log(Y) = \log\left(a \cdot X^b\right) = \log(a) + \log(X^b) = \log(a) + b \cdot \log(X)$$

de donde

$$Y' = A + b \cdot X', \quad \text{siendo} \quad \begin{cases} Y' = \log(Y) \\ A = \log(a) \\ X' = \log(X) \end{cases}$$

Por ello,

$$a = \mathrm{e}^A = \exp(A).$$

y programamos:

```
n <- length(x);
log.x <- log(x);
log.y <- log(y);
modelo.potencial <- lm( log.y ~ log.x );

## Coeficiente "a" del modelo POTENCIAL  Y = a*(X^b);
a <- exp(modelo.potencial$coefficients[[1]]);a;
## Coeficiente "b" del modelo POTENCIAL  Y = a*(X^b);
b <- modelo.potencial$coefficients[[2]];b;

cov(log.x,log.y);                  ## Covarianza;
cor(log.x,log.y);                  ## Coeficiente de correlación;
cor(log.x,log.y)^2;                ## Coeficiente de determinación;
sum((y - (a*(x^b)))^2)/n;          ## Varianza residual;

plot(x,y, col="blue");             ## Nube de puntos;
curve( a*(x^b), col="red", add=TRUE, lwd="3");
##
```

El modelo hiperbólico

El modelo hiperbólico $Y = a + \dfrac{b}{X}$

Hacemos la transformación:

$$Y = a + \frac{b}{X} = a + b \cdot \frac{1}{X}$$

de donde

$$Y = a + b \cdot X', \quad \text{siendo} \quad X' = 1/X$$

Programamos:

```
n <- length(x);
inv.x <- 1/x;
modelo.hiperbolico <- lm( y ~ inv.x );

## Coeficiente "a" del modelo HIPERBÓLICO  Y = a + b/X;
a <- modelo.hiperbolico$coefficients[[1]];a;
## Coeficiente "b" del modelo HIPERBÓLICO  Y = a + b/X;
b <- modelo.hiperbolico$coefficients[[2]];b;

cov(inv.x,y);                    ## Covarianza;
cor(inv.x,y);                    ## Coeficiente de correlación;
cor(inv.x,y)^2;                  ## Coeficiente de determinación;
sum((y - (a+b/x))^2)/n;          ## Varianza residual;

plot(x,y, col="blue");           ## Nube de puntos;
curve( a+b/x, col="red", add=TRUE, lwd="3");
##;
```

El modelo parabólico

El modelo parabólico $Y = a + bX + cX^2$

Programamos:

```
n <- length(x);
modelo.parabolico <- lm( y ~ x + I(x^2) );

## Coeficiente "a" del modelo PARABÓLICO general;
a <- modelo.parabolico$coefficients[[1]];a;
## Coeficiente "b" del modelo PARABÓLICO general;
b <- modelo.parabolico$coefficients[[2]];b;
## Coeficiente "c" del modelo PARABÓLICO general;
c <- modelo.parabolico$coefficients[[3]];c;

mp.VR <- sum((y - (a+b*x+c*x^2))^2)/n;
mp.VR;                                    ## Varianza residual;
var.y <- sum((y-mean(y))^2)/n;
var.y;                                    ## Varianza de la variable Y;
1-mp.VR/var.y;                            ## Coef. determinación;

plot(x,y, col="blue");                    ## Nube de puntos;
curve( a+b*x+c*x^2, col="red", add=TRUE, lwd="3");
##
```

El modelo parabólico simple

El modelo parabólico simple $Y = a + cX^2$

Programamos:

```
n <- length(x);
modelo.parabolico.simple <- lm( y ~ I(x^2) );

## Coeficiente "a" del modelo PARABÓLICO simple;
a <- modelo.parabolico.simple$coefficients[[1]];a;
## Coeficiente "c" del modelo PARABÓLICO simple;
c <- modelo.parabolico.simple$coefficients[[2]];c;

mps.VR <- sum((y - (a+c*x^2))^2)/n;
mps.VR;                                    ## Varianza residual;
var.y <- sum((y-mean(y))^2)/n;
var.y;                                     ## Varianza de la variable Y;
1-mps.VR/var.y;                            ## Coef. determinación;

plot(x,y, col="blue");                     ## Nube de puntos;
curve( a+c*x^2, col="red", add=TRUE, lwd="3");
##
```

El modelo cúbico

El modelo cúbico $Y = a + bX + cX^2 + dX^3$

Programamos:

```
n <- length(x);
modelo.cubico <- lm( y ~ x + I(x^2) + I(x^3) );

## Coeficiente "a" del modelo CÚBICO;
a <- modelo.cubico$coefficients[[1]];a;
## Coeficiente "b" del modelo CÚBICO;
b <- modelo.cubico$coefficients[[2]];b;
## Coeficiente "c" del modelo CÚBICO;
c <- modelo.cubico$coefficients[[3]];c;
## Coeficiente "d" del modelo CÚBICO;
d <- modelo.cubico$coefficients[[4]];d;

mc.VR <- sum((y - (a+b*x+c*x^2+d*x^3))^2)/n;
mc.VR;                                  ## Varianza residual;
var.y <- sum((y-mean(y))^2)/n;
var.y;                                  ## Varianza de la variable Y;
1-mc.VR/var.y;                          ## Coef. determinación;

plot(x,y, col="blue");                  ## Nube de puntos;
curve( a+b*x+c*x^2+d*x^3, col="red", add=TRUE, lwd="3");
##
```