

Selección de Modelos

Cuando trabajamos en forma teórica asumimos que tenemos un modelo $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$ y podemos obtener estimadores, test, intervalos de confianza y propiedades de optimalidad.

Sin embargo, en la práctica tenemos una muestra $(Y_1, \mathbf{x}_1), \dots, (Y_n, \mathbf{x}_n)$, de manera que la matriz de diseño \mathbf{X} tiene como filas a \mathbf{x}_i 's y lo primero que debemos determinar es cuáles de las columnas de \mathbf{X} debemos usar.

Un principio general para elegir un modelo es que sea **parsimonioso**, donde parsimonia se refiere a que conjugue simpleza con buen ajuste. La idea es hacer las cosas tan simples como sea posible, pero tampoco no tan simples....

Supongamos que realizamos la regresión entre \mathbf{Y} y $\mathbf{X} \in \mathbb{R}^{n \times k}$ ajustando del modelo

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

Cuando consideramos la matriz \mathbf{X} podemos obtener $2^k - 1$ modelos posibles relacionados con ella, llamemos $2^{\mathbf{X}}$ a este conjunto.

Si además considerásemos las posibles transformaciones de \mathbf{Y} y de cada una de las columnas de las covariables, este conjunto aumentaría considerablemente.

Existen algoritmos rápidos para computar todos estos ajustes y son especialmente útiles cuando p es grande, pero son necesarios **métodos de comparación** para elegir los mejores y debe tenerse en cuenta que el orden en que entran las variables al modelo puede afectar los resultados.

Análisis Exploratorio de Datos

El análisis exploratorio de datos nos puede guiar dándonos un primer esbozo. A tal fin podemos realizar gráficos de:

- Y vs. cada covariable
- Y vs. transformaciones de cada covariable
- transformaciones de Y vs. cada covariable
- residuos parciales

Recordemos que habíamos visto el coeficiente de regresión múltiple y coeficiente de regresión múltiple ajustado para evaluar la bondad del ajuste. Recordemos su definición:

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

$$R_{adj}^2 = 1 - (1 - R^2) \frac{n}{n - p}$$

Sin embargo, estos son sólo algunos de los métodos de comparación de modelos de uso frecuente.

Criterios de Selección de Modelos

- Criterios basados en el Error de Predicción
- Criterio C_p de Mallows
- Criterios de Información: AIC (Criterio de Akaike), BIC (Criterio de Información de Bayes), etc.
- Regularización

Balance entre Sesgo y Varianza

Volviendo al planteo inicial, supongamos que realizamos la regresión entre \mathbf{Y} y $\mathbf{X} \in \mathbb{R}^{n \times k}$ y que

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

donde $\beta_s = 0$ para un subconjunto de β_s de $\boldsymbol{\beta}$.

El **modelo verdadero** incluye solamente las columnas de \mathbf{X} para las cuales $\beta_s \neq 0$.

Llamaremos **modelo correcto** a un modelo verdadero con algunas columnas de \mathbf{X} extras.

Llamaremos **modelo incorrecto** a un modelo que no incluye todas las columnas del modelo verdadero.

Un criterio que parece razonables es elegir de acuerdo al error de predicción del modelo.

Cada uno de los modelos $M \in 2^{\mathbf{X}}$ conducirán a predicciones

$$\hat{\mathbf{Y}}(M) = \mathbf{P}_M \mathbf{Y}$$

Si observásemos nuevas respuestas independientes para el mismo diseño \mathbf{X} , el Error de Predicción para el modelo M puede calcularse como

$$\|\mathbf{Y}_+ - \hat{\mathbf{Y}}(M)\|^2 = \|\mathbf{Y}_+ - \mathbf{P}_M \mathbf{Y}\|^2$$

Sin embargo, esta norma es una variable aleatoria. Por lo tanto, un criterio posible es elegir el modelo M^* de acuerdo con el menor Error de Predicción Esperado (EPE), es decir

$$\min_{M \in 2^{\mathbf{X}}} \frac{1}{n} E \|\mathbf{Y}_+ - \mathbf{P}_M \mathbf{Y}\|^2 = \min_{M \in 2^{\mathbf{X}}} EPE$$

Supongamos que $\ddot{\mathbf{X}} \in \mathbb{R}^{n \times q}$ es el modelo verdadero y sea $\check{\mathbf{X}} \in \mathbb{R}^{n \times p}$ el modelo ajustado. Ambas $\ddot{\mathbf{X}}$ y $\check{\mathbf{X}}$ son construidas a partir de \mathbf{X} .

Como la verdadera relación entre \mathbf{Y} y \mathbf{X} es a través de $\check{\mathbf{X}}$, entonces

$$\mathbf{Y} = \check{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \boldsymbol{\eta} + \boldsymbol{\epsilon}$$

Las predicciones usando el modelo M asociado a $\check{\mathbf{X}}$ serán

$$\hat{\mathbf{Y}} = (\check{\mathbf{X}}'\check{\mathbf{X}})^{-1}\check{\mathbf{X}}'\mathbf{Y} = \check{\mathbf{P}}\mathbf{Y}$$

Si observásemos nuevas respuestas \mathbf{Y}_+ que corresponden a la misma matriz de diseño \mathbf{X} , como antes, tendríamos

$$\mathbf{Y}_+ = \check{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon}_+ = \boldsymbol{\eta} + \boldsymbol{\epsilon}_+$$

por lo tanto el Error de Predicción correspondiente al modelo M será:

$$\begin{aligned}\mathbf{Y}_+ - \hat{\mathbf{Y}} &= \boldsymbol{\eta} + \boldsymbol{\epsilon}_+ - \check{\mathbf{P}}(\boldsymbol{\eta} + \boldsymbol{\epsilon}) \\ &= (\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \boldsymbol{\epsilon}_+ - \check{\mathbf{P}}\boldsymbol{\epsilon}\end{aligned}$$

En consecuencia:

$$\begin{aligned}\|\mathbf{Y}_+ - \hat{\mathbf{Y}}\|^2 &= ((\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \boldsymbol{\epsilon}_+ - \check{\mathbf{P}}\boldsymbol{\epsilon})' ((\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \boldsymbol{\epsilon}_+ - \check{\mathbf{P}}\boldsymbol{\epsilon}) \\ &= \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\epsilon} + 0 + \boldsymbol{\epsilon}'_+(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} \\ &\quad + \boldsymbol{\epsilon}'_+\boldsymbol{\epsilon}_+ + \boldsymbol{\epsilon}'_+\check{\mathbf{P}}\boldsymbol{\epsilon} + 0 - \boldsymbol{\epsilon}'\check{\mathbf{P}}\boldsymbol{\epsilon}_+ + \boldsymbol{\epsilon}'\check{\mathbf{P}}\boldsymbol{\epsilon}\end{aligned}$$

y tomando esperanza obtenemos:

$$\begin{aligned}E\|\mathbf{Y}_+ - \hat{\mathbf{Y}}\|^2 &= \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + tr(\Sigma_{\boldsymbol{\epsilon}_+}) + E(\boldsymbol{\epsilon}'\check{\mathbf{P}}\boldsymbol{\epsilon}) \\ &= \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \sigma^2 n + \sigma^2 tr(\check{\mathbf{P}}) \\ &= \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + \sigma^2 n + \sigma^2 rg(\check{\mathbf{P}})\end{aligned}$$

De esta forma resulta:

$$EPE = \begin{cases} (1 + \frac{q}{n})\sigma^2 & \text{modelo verdadero} \\ (1 + \frac{p}{n})\sigma^2 & \text{modelo correcto} \\ \frac{1}{n}\boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + (1 + \frac{p}{n})\sigma^2 & \text{modelo incorrecto} \end{cases}$$

De tal forma que si elegimos un modelo correcto incrementamos la varianza, pero si elegimos un modelo incorrecto introducimos un sesgo.

Convalidación Cruzada (CV)

Obviamente EPE no podemos calcularlo, por lo tanto tendremos que estimarlo.

Si n es grande se pueden dividir los datos en dos: una parte para ajustar (training sample) y la otra para estimar el error de predicción (validation sample):

$$\begin{cases} \mathbf{X}^* \mathbf{Y}^* & \text{para ajustar el modelo} \\ \mathbf{X}^o \mathbf{Y}^o & \text{para estimar a } EPE \end{cases}$$

de manera que

$$E\widehat{PE} = \frac{1}{n^o} \|\mathbf{Y}^o - \mathbf{X}^o \widehat{\boldsymbol{\beta}}\|^2$$

Muchas veces ocurre que n no es lo suficientemente grande como para dividir la muestra en dos y por esa razón se usa CV basado en el método leave-one-out en el que se saca una observación por vez y se predice con el resto de las $n - 1$ observaciones:

$$E\widehat{PE}_{CV} = \frac{1}{n} \sum_{j=1}^n (y_j - \mathbf{x}_j \widehat{\boldsymbol{\beta}}_{(j)})^2$$

donde $\widehat{\boldsymbol{\beta}}_{(j)}$ se computa sin la observación j .

En base a la relación entre $\widehat{\boldsymbol{\beta}}$ y $\widehat{\boldsymbol{\beta}}_{(j)}$ tenemos que

$$E\widehat{P}E_{CV} = \frac{1}{n} \sum_{j=1}^n \frac{(y_j - \mathbf{x}_j \widehat{\boldsymbol{\beta}})^2}{1 - p_{jj}}$$

La idea es elegir las variables de manera de minimizar el $E\widehat{P}E_{CV}$.

C_p de Mallows

Notemos que si

$$E\|\mathbf{Y}_+ - \widehat{\mathbf{Y}}\|^2 = \boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta} + (n + p)\sigma^2$$

entonces

$$\frac{E\|\mathbf{Y}_+ - \widehat{\mathbf{Y}}\|^2}{\sigma^2} - (n + p) = \frac{\boldsymbol{\eta}'(\mathbf{I} - \check{\mathbf{P}})\boldsymbol{\eta}}{\sigma^2}$$

Mallows propone una medida cercana

$$C_p = \frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2}{s^2} + 2p - n$$

Notemos que $C_p \simeq p$ cuando el modelo es bueno. Un problema de este método es que necesita estimar a σ^2 y generalmente se hace usando el s^2 basado en las k covariables (es decir suponiendo sesgo pequeño).

Métodos Automáticos de Selección de Variables

Podemos dividirlos entre aquellos procedimientos de búsqueda que escogen el mejor entre todos los modelos posibles y aquellos que eligen iterativamente, en forma automática.

Búsqueda de todos los subconjuntos posibles

Este método consiste en evaluar todos los modelos posibles que se pueden construir en un conjunto dado de variables independientes.

Es particularmente útil cuando el número de variables no es demasiado grande. En general, uno puede forzar la presencia de ciertas variables y eso reduce el tamaño de la búsqueda. Uno puede imponer el criterio de selección R^2 , R^2_a y C_p .

Si bien el C_p parece el más razonable debemos tener en cuenta que asume que el modelo con todas las variables no tiene sesgo. Además, si bien se basa en los errores de predicción no tiene en cuenta que pasaría con futuras observaciones

En R contamos con Leaps.

Consideremos los datos de cemento.

Recordemos que la respuesta y ($y.hald$) es la temperatura de la mezcla de cemento y las 4 covariables ($x.hald$) son:

x1: tricalcium aluminate

x2: tricalcium silicate

x3: tetracalcium alumino ferrite

x4: dicalcium silicate.

Recordemos $\text{corr}(x1,x3) = -0.824$ y $\text{corr}(x2,x4) = -0.975$.

```
library(leaps)
```

```
library(wle)
```

```
data(hald)
```

```
hald
```

```
> cor(x.hald)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1.0000000	0.2285795	-0.8241338	-0.2454451
[2,]	0.2285795	1.0000000	-0.1392424	-0.9729550
[3,]	-0.8241338	-0.1392424	1.0000000	0.0295370
[4,]	-0.2454451	-0.9729550	0.0295370	1.0000000

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	78.5	7	26	6	60
[2,]	74.3	1	29	15	52
[3,]	104.3	11	56	8	20
[4,]	87.6	11	31	8	47
[5,]	95.9	7	52	6	33
[6,]	109.2	11	55	9	22
[7,]	102.7	3	71	17	6
[8,]	72.5	1	31	22	44
[9,]	93.1	2	54	18	22
[10,]	115.9	21	47	4	26
[11,]	83.8	1	40	23	34
[12,]	113.3	11	66	9	12
[13,]	109.4	10	68	8	12

all-subsets regression

```
leaps(x=x.hald, y=y.hald, method=c("Cp", "adjr2", "r2"))
```

```
leaps(x=x.hald, y=y.hald, method=c("Cp", "adjr2", "r2"))
```

```
$which
```

```

      1      2      3      4
1 FALSE FALSE FALSE TRUE
1 FALSE TRUE FALSE FALSE
1 TRUE FALSE FALSE FALSE
1 FALSE FALSE TRUE FALSE
2 TRUE TRUE FALSE FALSE
2 TRUE FALSE FALSE TRUE
2 FALSE FALSE TRUE TRUE
2 FALSE TRUE TRUE FALSE
2 FALSE TRUE FALSE TRUE
2 TRUE FALSE TRUE FALSE
3 TRUE TRUE FALSE TRUE
3 TRUE TRUE TRUE FALSE
3 TRUE FALSE TRUE TRUE
3 FALSE TRUE TRUE TRUE
4 TRUE TRUE TRUE TRUE

```

```
$label
```

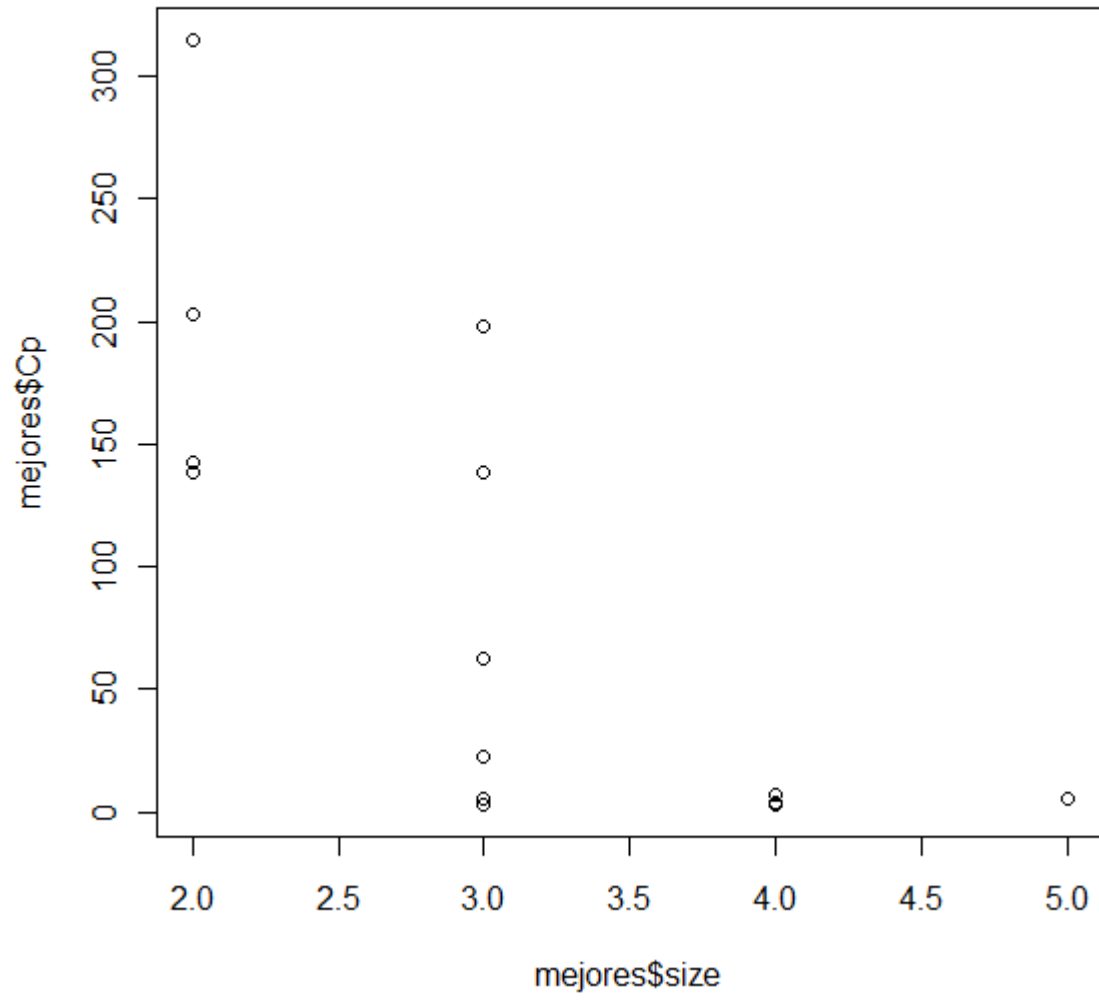
```
[1] "(Intercept)" "1"          "2"          "3"          "4"
```

```
$size
```

```
[1] 2 2 2 2 3 3 3 3 3 3 4 4 4 4 5
```

```
$Cp
```

```
[1] 138.730833 142.486407 202.548769 315.154284 2.678242 5.495851
[7] 22.373112 62.437716 138.225920 198.094653 3.018233 3.041280
[13] 3.496824 7.337474 5.000000
```

```
> cbind(leap.cem$size,leap.cem$Cp)
```

```
      [,1]      [,2]
 [1,] 2 138.730833
 [2,] 2 142.486407
 [3,] 2 202.548769
 [4,] 2 315.154284
 [5,] 3  2.678242
 [6,] 3  5.495851
 [7,] 3 22.373112
 [8,] 3 62.437716
 [9,] 3 138.225920
[10,] 3 198.094653
[11,] 4  3.018233
[12,] 4  3.041280
[13,] 4  3.496824
[14,] 4  7.337474
[15,] 5  5.000000
```

```
leaps(x=x.hald, y=y.hald, method=c("r2"))
```

```
$which
```

```

      1      2      3      4
1 FALSE FALSE FALSE TRUE
1 FALSE TRUE FALSE FALSE
1 TRUE FALSE FALSE FALSE
1 FALSE FALSE TRUE FALSE
2 TRUE TRUE FALSE FALSE
2 TRUE FALSE FALSE TRUE
2 FALSE FALSE TRUE TRUE
2 FALSE TRUE TRUE FALSE
2 FALSE TRUE FALSE TRUE
2 TRUE FALSE TRUE FALSE
3 TRUE TRUE FALSE TRUE
3 TRUE TRUE TRUE FALSE
3 TRUE FALSE TRUE TRUE
3 FALSE TRUE TRUE TRUE
4 TRUE TRUE TRUE TRUE

```

```
$label
```

```
[1] "(Intercept)" "1"          "2"          "3"          "4"
```

```
$size
```

```
[1] 2 2 2 2 3 3 3 3 3 3 4 4 4 4 5
```

```
$r2
```

```

[1] 0.6745420 0.6662683 0.5339480 0.2858727 0.9786784 0.9724710 0.9352896
[8] 0.8470254 0.6800604 0.5481667 0.9823355 0.9822847 0.9812811 0.9728200
[15] 0.9823756

```

```
leaps(x=x.hald, y=y.hald, method=c("adjr2"))
```

```
$which
```

```

      1      2      3      4
1 FALSE FALSE FALSE TRUE
1 FALSE TRUE FALSE FALSE
1 TRUE FALSE FALSE FALSE
1 FALSE FALSE TRUE FALSE
2 TRUE TRUE FALSE FALSE
2 TRUE FALSE FALSE TRUE
2 FALSE FALSE TRUE TRUE
2 FALSE TRUE TRUE FALSE
2 FALSE TRUE FALSE TRUE
2 TRUE FALSE TRUE FALSE
3 TRUE TRUE FALSE TRUE
3 TRUE TRUE TRUE FALSE
3 TRUE FALSE TRUE TRUE
3 FALSE TRUE TRUE TRUE
4 TRUE TRUE TRUE TRUE

```

```
$label
```

```
[1] "(Intercept)" "1"          "2"          "3"          "4"
```

```
$size
```

```
[1] 2 2 2 2 3 3 3 3 3 3 4 4 4 4 5
```

```
$adjr2
```

```

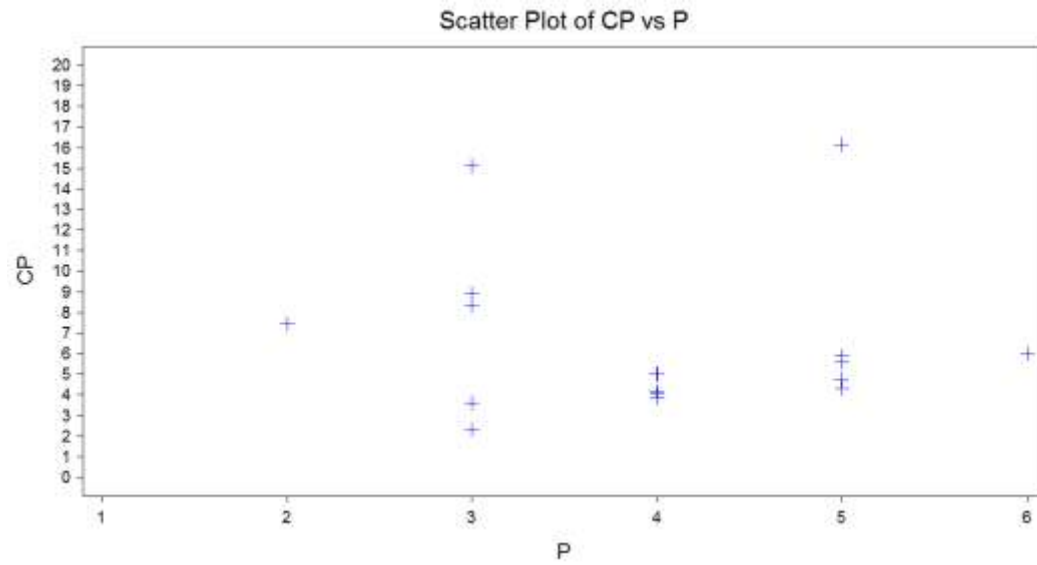
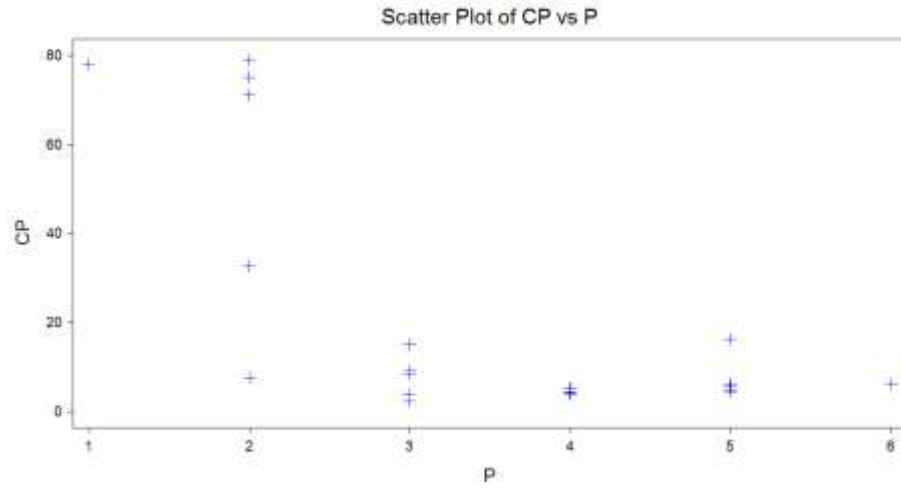
[1] 0.6449549 0.6359290 0.4915797 0.2209521 0.9744140 0.9669653 0.9223476
[8] 0.8164305 0.6160725 0.4578001 0.9764473 0.9763796 0.9750415 0.9637599
[15] 0.9735634

```

Datos de Biomasa

UNFORCED INDEPENDENT VARIABLES: (A)K (B)NA (C)PH (D)SAL (E)ZN

P	CP	ADJUSTED R SQUARE	R SQUARE	RESID SS	MODEL VARIABLES
---	---	---	---	---	-----
1	77.9	0.0000	0.0000	1.917E+07	INTERCEPT ONLY
2	7.4	0.5900	0.5994	7680575	C
2	32.7	0.3757	0.3899	1.169E+07	E
2	70.9	0.0525	0.0740	1.775E+07	B
2	74.8	0.0198	0.0421	1.836E+07	A
2	78.6	-0.0124	0.0106	1.897E+07	D
3	2.3	0.6422	0.6584	6548174	B C
3	3.6	0.6308	0.6476	6755845	A C
3	8.3	0.5896	0.6083	7509642	C E
3	8.9	0.5845	0.6034	7603247	C D
3	15.1	0.5313	0.5526	8576766	D E
4	3.8	0.6378	0.6625	6471149	B C E
4	4.0	0.6355	0.6604	6511089	A B C
4	4.2	0.6341	0.6590	6536396	B C D
4	5.0	0.6268	0.6522	6667664	A C D
4	5.0	0.6267	0.6521	6669300	A C E
5	4.3	0.6424	0.6749	6232954	A C D E
5	4.7	0.6389	0.6718	6292475	B C D E
5	5.6	0.6306	0.6642	6438038	A B C E
5	5.9	0.6279	0.6617	6485307	A B C D
5	16.1	0.5351	0.5773	8102649	A B D E
6	6.0	0.6360	0.6773	6186048	A B C D E



Procedimientos Stepwise

Existen tradicionalmente tres versiones: **Forward**, **Backward** y la combinación de ambos que es la **Stepwise**.

Podríamos decir que hay tantas implementaciones de este método como programas, por lo que es necesario leer detalladamente la descripción del programa que estamos utilizando.

Describiremos la implementación de `mle.stepwise` de `wle`.

Forward:

Este procedimiento no incluye inicialmente ninguna covariable, salvo la intercept, y va agregando las variables una a una de acuerdo con la que tiene mayor F parcial en los sucesivos modelos evaluados y superior al valor $F.in$.

Backard:

Este procedimiento incluye inicialmente todas las covariables y las va eliminando de a una a medida que el valor del F parcial sea inferior al valor $F.out$.

Stepwise:

Es una combinación de los dos anteriores y tiene en cuenta tanto el valor $F.in$ como el $F.out$.

Stepwise Regression: veamos un ejemplo de Forward

```
library(wle)
data(hald)
result <- mle.stepwise(y.hald~x.hald)
summary(result)
```

```
Forward selection procedure
```

```
F.in: 4
```

```
Last 3 iterations:
```

```
(Intercept) x.hald1 x.hald2 x.hald3 x.hald4
[1,]          1         0         0         0         1  22.800
[2,]          1         1         0         0         1 108.200
[3,]          1         1         1         0         1   5.026
```

```
> summary(lm(y.hald~x.hald[,1]))
```

```
Coefficients:
```

```
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  81.4793     4.9273   16.54 4.07e-09 ***
x.hald[, 1]   1.8687     0.5264    3.55 0.00455 **
```

```
Residual standard error: 10.73 on 11 degrees of freedom
Multiple R-squared:  0.5339,    Adjusted R-squared:  0.4916
F-statistic:  12.6 on 1 and 11 DF,  p-value: 0.004552
```



```
> summary(lm(y.hald~x.hald[,2]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	57.4237	8.4906	6.763	3.1e-05	***
x.hald[, 2]	0.7891	0.1684	4.686	0.000665	***

```
Residual standard error: 9.077 on 11 degrees of freedom  
Multiple R-squared: 0.6663, Adjusted R-squared: 0.6359  
F-statistic: 21.96 on 1 and 11 DF, p-value: 0.0006648
```

```
> summary(lm(y.hald~x.hald[,3]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	110.2027	7.9478	13.866	2.6e-08	***
x.hald[, 3]	-1.2558	0.5984	-2.098	0.0598	.

```
Residual standard error: 13.28 on 11 degrees of freedom  
Multiple R-squared: 0.2859, Adjusted R-squared: 0.221  
F-statistic: 4.403 on 1 and 11 DF, p-value: 0.05976
```

```
> summary(lm(y.hald~x.hald[,4]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	117.5679	5.2622	22.342	1.62e-10	***
x.hald[, 4]	-0.7382	0.1546	-4.775	0.000576	***

```
Residual standard error: 8.964 on 11 degrees of freedom  
Multiple R-squared: 0.6745, Adjusted R-squared: 0.645  
F-statistic: 22.8 on 1 and 11 DF, p-value: 0.0005762
```

```
salida.41<-lm(y.hald~ x.hald[,4]+x.hald[,1])
anova(salida.41)
```

Analysis of Variance Table

Response: y.hald

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 4]	1	1831.90	1831.90	245.03	2.319e-08	***
x.hald[, 1]	1	809.10	809.10	108.22	1.105e-06	***
Residuals	10	74.76	7.48			

```
salida.43<-lm(y.hald~ x.hald[,4]+x.hald[,3])
anova(salida.43)
```

Analysis of Variance Table

Response: y.hald

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 4]	1	1831.90	1831.90	104.240	1.314e-06	***
x.hald[, 3]	1	708.13	708.13	40.295	8.375e-05	***
Residuals	10	175.74	17.57			

```
salida.42<-lm(y.hald~ x.hald[,4]+x.hald[,2])
anova(salida.42)
```

Analysis of Variance Table

Response: y.hald

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 4]	1	1831.90	1831.90	21.0834	0.0009927	***
x.hald[, 2]	1	14.99	14.99	0.1725	0.6866842	
Residuals	10	868.88	86.89			

```
salida.412<-lm(y.hald~ x.hald[,4]+x.hald[,1]++x.hald[,2])
anova(salida.412)
```

Response: y.hald

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 4]	1	1831.90	1831.90	343.6758	1.771e-08	***
x.hald[, 1]	1	809.10	809.10	151.7934	6.150e-07	***
x.hald[, 2]	1	26.79	26.79	5.0259	0.05169	.
Residuals	9	47.97	5.33			

```
salida.413<-lm(y.hald~ x.hald[,4]+x.hald[,1]++x.hald[,3])
anova(salida.413)
```

Response: y.hald

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 4]	1	1831.90	1831.90	324.3179	2.285e-08	***
x.hald[, 1]	1	809.10	809.10	143.2435	7.875e-07	***
x.hald[, 3]	1	23.93	23.93	4.2358	0.06969	.
Residuals	9	50.84	5.65			

```
> summary(lm(y.hald~ x.hald[,1]+ x.hald[,2]+x.hald[,4]))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	71.6483	14.1424	5.066	0.000675	***
x.hald[, 1]	1.4519	0.1170	12.410	5.78e-07	***
x.hald[, 2]	0.4161	0.1856	2.242	0.051687	.
x.hald[, 4]	-0.2365	0.1733	-1.365	0.205395	

Residual standard error: 2.309 on 9 degrees of freedom

Multiple R-squared: 0.9823, Adjusted R-squared: 0.9764

F-statistic: 166.8 on 3 and 9 DF, p-value: 3.323e-08

```
resultb <- mle.stepwise(y.hald~x.hald,type="Backward")
summary(resultb)
```

Backward selection procedure

F.out: 4

Last 2 iterations:

	(Intercept)	x.hald1	x.hald2	x.hald3	x.hald4	
[1,]	1	1	1	0	1	0.01823
[2,]	1	1	1	0	0	1.86300

```
summary(lm(y.hald~ x.hald[,1]+ x.hald[,2]+ x.hald[,3]+x.hald[,4]))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	62.4054	70.0710	0.891	0.3991
x.hald[, 1]	1.5511	0.7448	2.083	0.0708 .
x.hald[, 2]	0.5102	0.7238	0.705	0.5009
x.hald[, 3]	0.1019	0.7547	0.135	0.8959 (0.135*0.135=0.018225)
x.hald[, 4]	-0.1441	0.7091	-0.203	0.8441

Residual standard error: 2.446 on 8 degrees of freedom

Multiple R-squared: 0.9824, Adjusted R-squared: 0.9736

F-statistic: 111.5 on 4 and 8 DF, p-value: 4.756e-07

```
anova(lm(y.hald~ x.hald[,1]+ x.hald[,2]+ x.hald[,4]))
```

```
Response: y.hald
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 1]	1	1450.08	1450.08	272.0439	4.934e-08	***
x.hald[, 2]	1	1207.78	1207.78	226.5879	1.094e-07	***
x.hald[, 4]	1	9.93	9.93	1.8633	0.2054	
Residuals	9	47.97	5.33			

```
anova(lm(y.hald~ x.hald[,1]+ x.hald[,4]+ x.hald[,2]))
```

```
Response: y.hald
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 1]	1	1450.08	1450.08	272.0439	4.934e-08	***
x.hald[, 4]	1	1190.92	1190.92	223.4253	1.163e-07	***
x.hald[, 2]	1	26.79	26.79	5.0259	0.05169	.
Residuals	9	47.97	5.33			

```
anova(lm(y.hald~ x.hald[,2]+ x.hald[,4]+ x.hald[,1]))
```

```
Response: y.hald
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 2]	1	1809.43	1809.43	339.460	1.870e-08	***
x.hald[, 4]	1	37.46	37.46	7.027	0.02644	*
x.hald[, 1]	1	820.91	820.91	154.008	5.781e-07	***
Residuals	9	47.97	5.33			

```
anova(lm(y.hald~ x.hald[,1]+ x.hald[,2]))
```

```
Response: y.hald
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 1]	1	1450.1	1450.08	250.43	2.088e-08	***
x.hald[, 2]	1	1207.8	1207.78	208.58	5.029e-08	***
Residuals	10	57.9	5.79			

```
anova(lm(y.hald~ x.hald[,2]+ x.hald[,1]))
```

```
Response: y.hald
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
x.hald[, 2]	1	1809.43	1809.43	312.48	7.149e-09	***
x.hald[, 1]	1	848.43	848.43	146.52	2.692e-07	***
Residuals	10	57.90	5.79			

```
results <- mle.stepwise(y.hald~x.hald,type="Stepwise")
summary(results)

mle.stepwise(formula = y.hald ~ x.hald, type = "Stepwise")
```

Stepwise selection procedure

```
F.in: 4
F.out: 4
```

Last 4 iterations:

	(Intercept)	x.hald1	x.hald2	x.hald3	x.hald4	
[1,]	1	0	0	0	1	22.800
[2,]	1	1	0	0	1	108.200
[3,]	1	1	1	0	1	5.026
[4,]	1	1	1	0	0	1.863

```
> summary(lm(y.hald~x.hald[,4]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	117.5679	5.2622	22.342	1.62e-10	***
x.hald[, 4]	-0.7382	0.1546	-4.775	0.000576	***

```
Residual standard error: 8.964 on 11 degrees of freedom
Multiple R-squared: 0.6745, Adjusted R-squared: 0.645
F-statistic: 22.8 on 1 and 11 DF, p-value: 0.0005762
```

```
> summary(lm(y.hald~ x.hald[,1]+x.hald[,4]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	103.09738	2.12398	48.54	3.32e-13	***
x.hald[, 1]	1.43996	0.13842	10.40	1.11e-06	***
x.hald[, 4]	-0.61395	0.04864	-12.62	1.81e-07	***

```
Residual standard error: 2.734 on 10 degrees of freedom
Multiple R-squared: 0.9725, Adjusted R-squared: 0.967
F-statistic: 176.6 on 2 and 10 DF, p-value: 1.581e-08
```

```
> summary(lm(y.hald~ x.hald[,1]+ x.hald[,2]+x.hald[,4]))
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	71.6483	14.1424	5.066	0.000675	***
x.hald[, 1]	1.4519	0.1170	12.410	5.78e-07	***
x.hald[, 2]	0.4161	0.1856	2.242	0.051687	.
x.hald[, 4]	-0.2365	0.1733	-1.365	0.205395	

```
Residual standard error: 2.309 on 9 degrees of freedom
Multiple R-squared: 0.9823, Adjusted R-squared: 0.9764
F-statistic: 166.8 on 3 and 9 DF, p-value: 3.323e-08
```


Forward selection procedure

F.in: 4

Last 3 iterations:

	(Intercept)	x.hald1	x.hald2	x.hald3	x.hald4	
[1,]	1	0	0	0	1	22.800
[2,]	1	1	0	0	1	108.200
[3,]	1	1	1	0	1	5.026

Backward selection procedure

F.out: 4

Last 2 iterations:

	(Intercept)	x.hald1	x.hald2	x.hald3	x.hald4	
[1,]	1	1	1	0	1	0.01823
[2,]	1	1	1	0	0	1.86300

Stepwise selection procedure

F.in: 4

F.out: 4

Last 4 iterations:

	(Intercept)	x.hald1	x.hald2	x.hald3	x.hald4	
[1,]	1	0	0	0	1	22.800
[2,]	1	1	0	0	1	108.200
[3,]	1	1	1	0	1	5.026
[4,]	1	1	1	0	0	1.863