EJEMPLOS DE REGRESIÓN CON SIMULACIONES

```
> n<- 10 ## tamaño de la muestra
> beta0<- 10; beta1<- 3 ## coeficientes de regresión
> sigma<- 10 ## desviación típica de los residuales
> x1<- round(runif(n, 0, 5), 1)
> ## generación del modelo
> ## residuales= N(0, sigma)
> y<- beta0 + beta1*x1 + round(rnorm(n, 0, sigma),1)
> ## Menú Estadísticos + Ajuste de modelos * regresión lineal
> reg<- lm(y ~ x1)## estima el modelo de regresión
> summary(reg) ## muestra los resultados del análisis
Call:
lm(formula = y ~ x1)
Residuals:
    Min      1Q  Median      3Q     Max
-19.620  -7.770   1.893   5.547  18.860
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.127      7.533   1.212    0.260
x1             3.533      2.940   1.202    0.264

Residual standard error: 12.21 on 8 degrees of freedom
Multiple R-squared:  0.1529,  Adjusted R-squared:  0.04706
F-statistic: 1.444 on 1 and 8 DF,  p-value: 0.2638
```

Conclusiones

- R2= 0.1529. x1 explica el 15,29% de la variabilidad de y.
- beta1=3.533. La relación, si existe, es creciente
- Al contrastar la hipótesis H0:beta1=0 se obtiene un p-valor de 0.264, es decir, no existe una evidencia clara en contra de que x1 e y sean linealmente independientes

Repetir el proceso con n=100

```
> n<- 100 ## tamaño de la muestra
> x1<- round(runif(n, 0, 5), 1)
> ## generación del modelo
> ## residuales= N(0, sigma)
> y<- beta0 + beta1*x1 + round(rnorm(n, 0, sigma))
> reg<- lm(y ~ x1)## estima el modelo de regresión
> summary(reg) ## muestra los resultados del análisis
Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  10.7098     2.1261   5.037 0.00000215 ***
x1            2.9660     0.7147   4.150 0.00007090 ***

Residual standard error: 9.814 on 98 degrees of freedom
Multiple R-squared:  0.1495,  Adjusted R-squared:  0.1408
F-statistic: 17.22 on 1 and 98 DF,  p-value: 0.0000709
```

Conclusiones

- R2= 0.1495. Muy parecido al anterior 0.1529.
- beta1=2.966 menor que el beta1 anterior=3.533.
- H0:beta1=0. p-valor= 0.00007090 implica el rechazo de beta1=0.
- x1 e y no son linealmente independientes

El tamaño de la muestra juega un papel muy importante en el resultado de los contrastes de hipótesis. Resultados parecidos, pero basados en tamaños de muestra muy diferentes, pueden dar lugar a conclusiones diferentes.

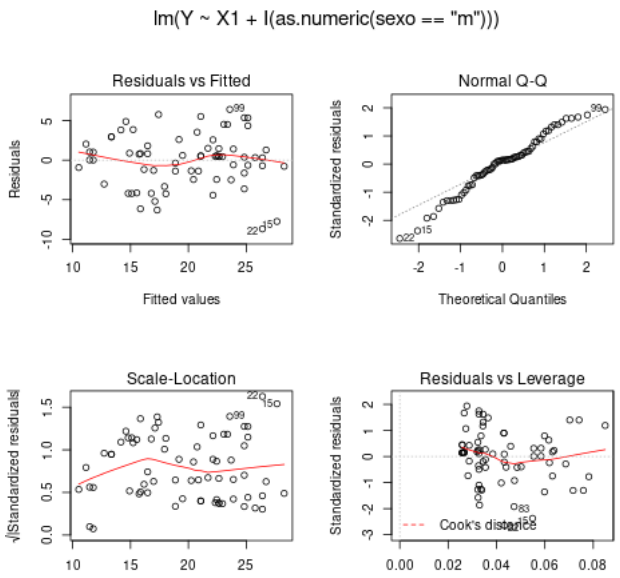
Repetir el proceso con distintas sigmas. ¿Cómo afectarán a la regresión?

Comprobar que al disminuir sigma, aumenta el valor de R2 y se rechaza con mayor facilidad la hipótesis de independencia entre x1 e y.

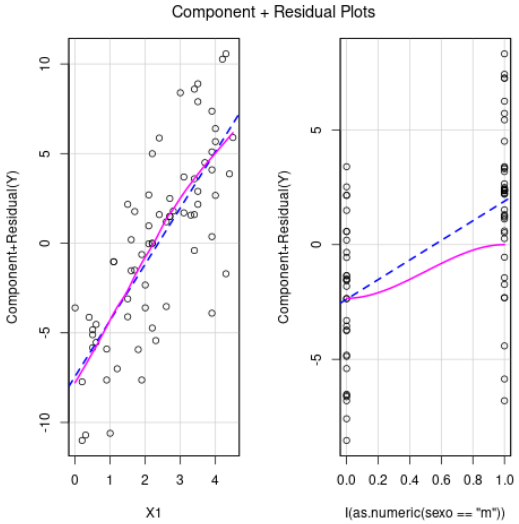
Regresión con variables cualitativas

```
>  ## obtener muestras al azar de una población finita
>  c(1:7) ## población: 1,2,3,..., 7
[1] 1 2 3 4 5 6 7
>  sample(c(1:7), 6,replace=TRUE ) ## muestreo con reemplazamiento
[1] 4 2 5 3 5 1
>  sample(c(1:7), 6, replace=FALSE) ## muestreo sin reemplazamiento
[1] 7 3 4 2 5 1
>  ## sample(c("h", "m"), n, replace=TRUE) ## n observaciones con "h", "m"

>  n<- 100
>  beta0<- 10; beta1<- 3 ; beta2<- 5; sigma<- 4 ## parámetros
>  X1<- round(runif(n, 0, 5), 1)
>  sexo<- sample(c("h", "m"), n, replace=TRUE); sexo  ## genera la variable
sexo. h=hombre; m= mujer
  [1] "h" "m" "m" "m" "m" "h" "m" "m" "m" "m" "m" "m" "h" "h" "h" "m" "h"
..................
 [97] "m" "m" "m" "h"

>  ## Modelo lineal Y = b0 + b1 * X1 + b2 * sexo + residual
>  Y<- beta0 + beta1*X1 + beta2*as.numeric(sexo=="m")+round(rnorm(n, 0, sigma))
>  Y<- beta0 + beta1*X1 + beta2*(sexo=="m")+round(rnorm(n, 0, sigma))
>  (sexo=="m") ## vector de tipo lógico
  [1] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  FALSE
.......................
 [97]  TRUE  TRUE  TRUE FALSE

>  beta2 * (sexo=="m") ## ## vector lógico transformado en numérico.
>  beta2 * FALSE = beta2 * 0 = 0; beta2 * TRUE = beta2 * 1 = beta2
  [1] 0 5 5 5 5 0 5 5 5 5 5 0 0 0 5 0 0 0 0 0 5 5 0 0 0 0 5  0 5 0 5 0 5 5 5 0 0
.......................
 [97] 5 5 5 0

> ## Crear un fichero de datos de R
>  EJ<- data.frame(X1, sexo, Y)
>  head(EJ)  ## muestra los 6 primeros casos de EJ
   X1 sexo    Y
1 0.6    h 11.8
2 0.9    m 15.7
3 4.0    m 28.0
4 1.2    m 20.6
5 1.2    m 14.6
6 3.3    h 18.9
>  ## Menú: poner EJ como conjunto de datos activo
>   summary(EJ)  ## describir las variables del fichero
       X1       sexo         Y
 Min.   :0.00   h:44   Min.   : 9.40
 1st Qu.:1.35   m:56   1st Qu.:15.53
 Median :2.25          Median :20.05
 Mean   :2.37          Mean   :20.00
 3rd Qu.:3.50          3rd Qu.:24.15
 Max.   :5.00          Max.   :38.50

> ## Nivel de significación elegido: alfa= 0.05
>  reg2<- lm( Y ~ X1+sexo, data=EJ) ## regresión con sexo como factor.
> ##  beta0<- 10; beta1<- 3 ; beta2<- 5; sigma<- 4 ## parámetros
>  summary(reg2)
Coefficients:
            Estimate Std. Error t value     Pr(>|t|)
(Intercept)  10.0923     0.8271  12.202      < 2e-16 ***
X1            3.1035     0.2804  11.067      < 2e-16 ***
sexo[T.m]     4.5577     0.7497   6.079 0.0000000238 ***
```

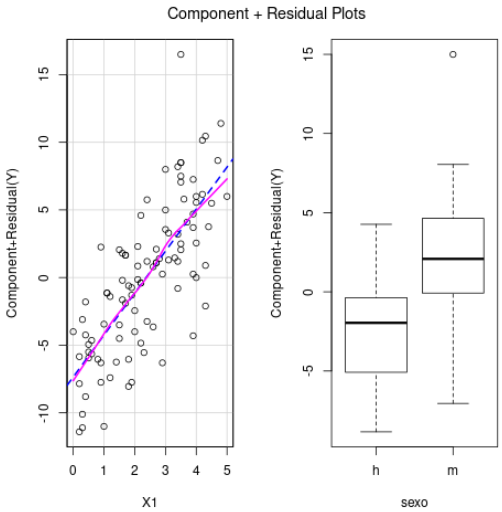
```
---
Residual standard error: 3.692 on 97 degrees of freedom
Multiple R-squared:  0.6488,  Adjusted R-squared:  0.6415
F-statistic: 89.59 on 2 and 97 DF,  p-value: < 2.2e-16

> ## codificación numérica de sexo. hombre=0; mujer=1
> as.numeric(sexo=="m"); sexo  ## son equivalentes.
  [1] 0 1 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1
.............................
 [97] 1 1 1 0
> sexo
  [1] "h" "m" "m" "m" "m" "h" "m" "m" "m" "m" "m" "m" "h" "h" "h" "m" "h"
.............................
 [97] "m" "m" "m" "h"

>  reg3<- lm(Y~X1+I(as.numeric(sexo=="m")),data=EJ) ## X1, sexo: numéricas
> ## I( g(X)) permite usar g(X) en lugar de X en la regresión. Ejemplos:
> ## I(X^2) usa X  al cuadrado en lugar de X
> ## I(as.numeric(sexo=="m")) en sexo cambia "h" por 0; "m" por 1.

> ## comparación de los dos modelos
> reg2
lm(formula = Y ~ X1 + sexo, data = EJ)
Coefficients:
(Intercept)           X1    sexo[T.m]
     10.092        3.104        4.558
> reg3
lm(formula = Y ~ X1 + I(as.numeric(sexo == "m")), data = EJ)
Coefficients:
(Intercept)           X1  I(as.numeric(sexo == "m"))
     10.092        3.104        4.558
> ## Los resultados son idénticos en las dos regresiones.

> ## En regresión una variable cualitativa V con categorías (A,B,C) pasa a
V1,V2.
> ##  V     V1    V2
> ##  A     1     0
> ##  B     0     1
> ##  C     0     0
> ## Una variable cualitativa V con k categorías se recodifica en k-1 variables

> ## Entrenamiento + Validación
> entrena<- sample(c(1:n), round(0.70 *n))
> valida<- setdiff(c(1:n), entrena) ## casos que no pertenecen a entrena
> reg4<- lm( Y ~ X1+ I(as.numeric(sexo=="m")), data=EJ, subset=entrena)
> summary(reg4)
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                 9.9017     0.9284  10.665 4.43e-16 ***
X1                          3.1341     0.3240   9.673 2.42e-14 ***
I(as.numeric(sexo == "m"))  4.2726     0.8165   5.233 1.80e-06 ***
Residual standard error: 3.361 on 67 degrees of freedom
Multiple R-squared:  0.6725,  Adjusted R-squared:  0.6627
F-statistic: 68.79 on 2 and 67 DF,  p-value: < 2.2e-16

> ## Análisis de los residuales
# Menú Modelos + Graficas
> ## Gráfica 1. (1)+Gráficas básicas de diagnóstico
> oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
> plot(reg4)
> par(oldpar)
> ## Gráfica 2. (2) Gráfica componentes + residuos  (modelo reg4)
> crPlots(reg4, smooth=list(span=0.5))
> ## Gráfica 2. Gráfica componentes + residuos  (modelo reg2)
> crPlots(reg2, smooth=list(span=0.5))
```



Gráfica 1.



Gráfica 2.(reg4)

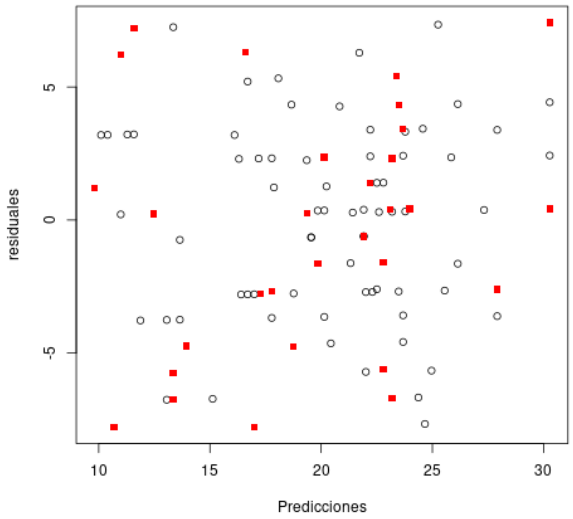


.

Gráfica 3 (reg2)

> ## Todas las gráficas sugieren un modelo lineal.

```
> ## Comparación de entrenamiento y validación
> Y_pred<- predict(reg4, EJ) ## predicciones
> Y_res<- EJ$Y - Y_pred  ## residuales
> ## gráfica de predicciones frente a residuales. Entrena+valida
> limx<- c(min(Y_pred), max(Y_pred))
> limy<- c(min(Y_res), max(Y_res))
> plot(Y_pred[entrena], Y_res[entrena], xlim=limx, ylim=limy,
xlab="Predicciones", ylab="residuales")
> points(Y_pred[valida], Y_res[valida], col="red", pch=15)
```




```
> reg4
lm(formula=Y~X1+I(as.numeric(sexo == "m")),data= EJ, subset = entrena)
Coefficients:
(Intercept)           X1  I(as.numeric(sexo == "m"))
   9.902          3.134                       4.273

> lm( Y ~ X1+ I(as.numeric(sexo=="m")), data=EJ, subset=valida)
lm(formula=Y~X1+I(as.numeric(sexo == "m")),data = EJ, subset = valida)
Coefficients:
(Intercept)           X1  I(as.numeric(sexo == "m"))
  10.487          3.059                       5.190


> ## Interpretación de los resultados:
> ## Los coeficientes de regresión son significativamente diferente de cero.
> ## A igual sexo, aumentar una unidad X1 significa un incremento  de 3.1 en Y.
> ## A igual X1, las mujeres tienen, en Y, unos 5 puntos más que los hombres.
> ## Los resultados de entrenamiento y validación son parecidos
> ## La desviación típica de los residuales es 2.4 aproximadamente.
> ## X1 + sexo explican cerca del 70% de la variabilidad de Y
```

```
### EJEMPLO DE REGRESIÓN LOGÍSTICA ###
## Fichero de datos:  https://stats.idre.ucla.edu/stat/data/binary.csv"
  ## gre (Puntuación en el examen de acceso)
  ## gpa (Puntuación media del grado)
  ## rank (prestigio del centro de origen): 1-4. Más rango menor prestigio
  ## admit (admisión en el centro de postgrado): 0= no; 1= si
     ## Analizar la relación de las  variables gre, gpa, rank con admit
     ## Construir un modelo para predecir admit
> DB<- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

> summary(DB)
     admit             gre             gpa             rank
 Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
 Median :0.0000   Median :580.0   Median :3.395   Median :2.000
 Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
 Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000

> DB$admit<- factor(DB$admit, labels=c("no", "si"))
> summary(DB)
 admit         gre             gpa             rank
 no:273   Min.   :220.0   Min.   :2.260   Min.   :1.000
 si:127   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
          Median :580.0   Median :3.395   Median :2.000
          Mean   :587.7   Mean   :3.390   Mean   :2.485
          3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
          Max.   :800.0   Max.   :4.000   Max.   :4.000

> numSummary(DB[, c("gre", "gpa", "rank")], groups=DB$admit)
Variable: gre
       mean       sd IQR  0% 25% 50% 75% 100%   n
no 573.1868 115.8302 160 220 500 580 660  800 273
si 618.8976 108.8849 140 300 540 620 680  800 127
## test de igualdad de medias
> t.test(gre~admit, alternative='two.sided', conf.level=.95, var.equal=FALSE,
data=DB)
      Welch Two Sample t-test
data:  gre by admit
t = -3.8292, df = 260.18, p-value = 0.0001611
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: -69.21683 -22.20482
 ## Diferencias importantes

Variable: gpa
       mean        sd   IQR   0%  25%  50%   75% 100%   n
no 3.343700 0.3771330 0.530 2.26 3.08 3.34 3.610    4 273
si 3.489213 0.3701771 0.535 2.42 3.22 3.54 3.755    4 127
## test de igualdad de medias
> t.test(gpa~admit, alternative='two.sided', conf.level=.95, var.equal=FALSE,
data=DB)
      Welch Two Sample t-test
data:  gpa by admit
t = -3.6379, df = 250.05, p-value = 0.0003339
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:  -0.2242921 -0.0667338
  ## Diferencia algo menor

Variable: rank
       mean        sd IQR 0% 25% 50% 75% 100%   n
no 2.641026 0.9171978   1  1   2   3   3    4 273
si 2.149606 0.9178887   2  1   1   2   3    4 127
  ## la diferencia entre los rangos medios es de media desv. típica
```

```
## Test chi-cuadrado deindependencia entre admit y rank (rank como cualitativa)
> T<- table(DB$admit, DB$rank) ## tabla de contingencia
> chisq.test(T)
      Chi-squared test= 25.242, df = 3, p-value = 0.00001374
> round(prop.table(T,2),2)

        1    2    3    4
  no 0.46 0.64 0.77 0.82
  si 0.54 0.36 0.23 0.18
## Diferencia importante entre las proporciones de admitidos según el rango.
## 54% en 1; 36% en 2; 21% en 3; 18% en 4.

> ## estimación del modelo. Entrenamiento + validación
> n1<- dim(DB)[1]; n1  ## numero de casos del fichero
[1] 400
> entrena1<- sample(1:n1, round(0.70*n1)); length(entrena1)
[1] 280
> valida1<- setdiff(c(1:n1), entrena1); length(valida1)
[1] 120
> Menú Estadísticos * Ajuste de modelos + Modelo lineal generalizado
> GLM.1 <- glm(admit ~ gpa + gre + rank, family=binomial(logit), data=DB,
subset=entrena1)
> summary(GLM.1)
Call:
glm(formula = admit ~ gpa + gre + rank, family = binomial(logit),
    data = DB, subset = entrena1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5112  -0.9033  -0.6322   1.1720   2.1320

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.527025   1.334447  -2.643 0.008216 **
gpa          0.795670   0.374276   2.126 0.033512 *
gre          0.002312   0.001226   1.886 0.059317 .
rank        -0.532935   0.153143  -3.480 0.000501 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 353.13  on 279  degrees of freedom
Residual deviance: 324.89  on 276  degrees of freedom
AIC: 332.89
Number of Fisher Scoring iterations: 3

> exp(coef(GLM.1))  # Exponentiated coefficients ("odds ratios")
(Intercept)         gpa         gre        rank
 0.02939221  2.21592447  1.00231460  0.58688000

> ## A mayor "gpa" y "gre" mayor probabilidad de ser admitido.
> ## A mayor "rank" menor probabilidad de ser admitido
> ## El coeficiente de "gre" no es significativamente distinto de cero.
## rge tenía diferencias significativas según admit.
## Este comportamiento se explica porque "rge" se relaciona con "gpa2 y "rank"
## Parte de la aportación de "rge" a "admit" se encuentra en "gpa2 y "rank"

> ## Elección de la probabilidad de corte
> DB$pred<- predict(GLM.1, DB, type="response") ## predicciones
> numSummary(DB$pred[entrena1], groups=DB$admit[entrena1])
Statistic
Group   mean     sd    IQR     0%    25%    50%    75%   100%
   no 0.2931 0.1357 0.2057 0.0649 0.1796 0.2787 0.3853 0.6808
   si 0.3912 0.1477 0.1934 0.1030 0.2908 0.3827 0.4842 0.7256
```

```
> head(DB$pred)
        1         2         3         4         5         6
0.1932270 0.3240387 0.7472865 0.1405450 0.0897638 0.3776522

> Boxplot(pred ~ admit, data=DB, id=list(method="y"))
[1] "70"  "294"
> ## Criterio para la admisión: p>0.35

> DB$grupo<- factor(DB$pred>0.35, labels=c("p.no", "p.si"))
> table(DB$grupo)
p.no p.si
 230  170
> ## Estimación de la proporción de aciertos y fallos
> ## Matriz de confusión con entrena1
> T.E<- table(DB$admit[entrena1], DB$grupo[entrena1])
> prop.table(T.E, 1)
          p.no      p.si
  no 0.6931217 0.3068783
  si 0.4065934 0.5934066
 ## La proporción media de aciertos está en torno al 65%
> ## Matriz de confusión con valida1
> T.V<- table(DB$admit[valida1], DB$grupo[valida1])
> prop.table(T.V, 1)
          p.no      p.si
  no 0.6071429 0.3928571
  si 0.3055556 0.6944444
 ## La media de aciertos se mantiene en torno al 65%
> ## No se aprecian indicios de sobreajuste con el modelo estimado

> ## Selección automática del modelo
> Menú Modelos + Selección del modelo
> modelo1 <- stepwise(GLM.1, direction='forward/backward', criterion='BIC')
Direction:  forward/backward
Criterion:  BIC
Start:  AIC=358.76
admit ~ 1
       Df Deviance    AIC
+ rank  1   337.30 348.57
+ gre   1   342.44 353.71
+ gpa   1   343.35 354.62
<none>      353.13 358.76

Step:  AIC=348.57
admit ~ rank

       Df Deviance    AIC
+ gpa   1   328.51 345.42
+ gre   1   329.53 346.44
<none>      337.30 348.57
- rank  1   353.13 358.76

Step:  AIC=345.42
admit ~ rank + gpa

       Df Deviance    AIC
<none>      328.51 345.42
+ gre   1   324.89 347.43
- gpa   1   337.30 348.57
- rank  1   343.35 354.62

> summary(modelo1)

Call:
glm(formula = admit ~ rank + gpa, family = binomial(logit), data = DB,
```

```
    subset = entrena1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4707  -0.8996  -0.6781   1.1454   2.1445

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.8794     1.2778  -2.253 0.024231 *
rank         -0.5632     0.1518  -3.711 0.000206 ***
gpa           1.0275     0.3550   2.894 0.003803 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 353.13  on 279  degrees of freedom
Residual deviance: 328.51  on 277  degrees of freedom
AIC: 334.51
Number of Fisher Scoring iterations: 4

> DB$pred1<- predict(modelo1, DB, type="response") ## predicciones
> numSummary(DB$pred1[entrena1], groups=DB$admit[entrena1])
Statistic
Group   mean     sd    IQR     0%    25%    50%    75%   100%
   no 0.2967 0.1249 0.1901 0.0702 0.1971 0.2739 0.3872 0.6609
   si 0.3838 0.1443 0.2243 0.1003 0.2822 0.3794 0.5065 0.6609
> ## El criterio para la admisión se mantiene: p>0.35

> DB$grupo1<- factor(DB$pred1>0.35, labels=c("p.no", "p.si"))

> ## Matriz de confusión con entrena1
> T.E1<- table(DB$admit[entrena1], DB$grupo1[entrena1])
> prop.table(T.E1, 1)
          p.no      p.si
  no 0.6560847 0.3439153
  si 0.4065934 0.5934066

> ## Matriz de confusión con valida1
> T.V1<- table(DB$admit[valida1], DB$grupo1[valida1])
> prop.table(T.V1, 1)
          p.no      p.si
  no 0.6071429 0.3928571
  si 0.3888889 0.6111111

> ## Qué modelo se elige?

> ## Predicciones para nuevos datos
> ND <- with(DB, data.frame(gre= mean(gre), gpa = mean(gpa), rank= 1:4))
> predict(GLM.1, ND, type="response")
        1         2         3         4
0.4989996 0.3689005 0.2554276 0.1675898
## la previsión es que los casos 1, 2 sean admitidos y los otros no.

>  Qué ocurre si usamos rank como factor?
```

```
## generación de modelos con relaciones no lineales
## Probar con distintos modelos y parámetros
n<- 60
## parámetros del modelo de regresión
beta0<- 10;
beta1<- 0;
beta1.1<- .5
beta2<- -1
beta3<- -80
beta4<- 0
sigma<-300 ## desviación típica de los residuales
X1<- 5 + seq(1, 80, length=n)
X2<- round(0.1*X1 + rnorm(n, 0, 7),1)
X3<- rbinom(n, 30, .5)
X4<- round(rnorm(n, 50, 10) ,1)

Y<- beta0+beta1*X1+beta1.1*X1^2 -
beta2*X2+beta3*X3+beta4*X4+round(rnorm(n,0,sigma),1)
D<- data.frame(X1,X2,X3,X4,Y)
head(D)
numSummary(D)
## matriz de correlaciones
cor(D, use="pair") ## use="pair" o use="complete": manejo de datos perdidos
cor(D[,c("X1","X2","X3","X4","Y")], use="complete")

summary(D)
## matriz de gráficas.
## Menú Gráficas + Matriz diagramaas de dispersión
scatterplotMatriX(~X1+X2+X3+X4+Y, regLine=FALSE, smooth=FALSE,
diagonal=list(method="densitY"), data=D)

## Matriz de gráficas de las X con Y. Pprogramando la sintaxis
par(mfrow= c(2,2)) ## organiza las gráficas como una matriz 2*2
for (j in 1:4) {
  titulo<- paste(names(D)[j], " ~ ", ylab=names(D)[5]) ## cadena con el titulo
  plot( D[,j], D[,5], xlab=names(D)[j], ylab=names(D)[5], main= titulo)}
par(mfrow= c(1,1)) ## vuelve a una sola gráfica

## Entrenamiento + Validación
entrena<- sample(c(1:n), round(0.70 *n))
valida<- setdiff(c(1:n), entrena) ## casos que no pertenecen a entrena

reg<- lm(Y ~ X1+X2+X3+X4, data=D, subset=entrena)
summary(reg)
## selección automática de las variables del modelo
modelo<- stepwise(reg, direction='forward/backward', criterion='BIC')
## stepwise(reg, direction='forward/backward', criterion='AIC')
names(modelo)
summary(modelo)

oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(modelo)
par(oldpar)
crPlots(modelo, smooth=list(span=0.5))
```