Clasificación supervisada con R

Diagnóstico de cáncer de mama por imagen

Los datos del fichero Wisconsin.RData proceden de un estudio sobre diagnóstico del cáncer de mama por imagen. Mediante una punción con aguja fina se extrae una muestra del tejido sospechoso de la paciente. La muestra se tiñe para resaltar los núcleos de las células y se determinan los límites exactos de los núcleos. Las variables consideradas corresponden a los valores medios de distintos aspectos de la forma de los núcleos de cada muestra.

El fichero contiene un `data.frame`, llamado `datos`, con 10 variables explicativas medidas en pacientes cuyos tumores fueron diagnosticados posteriormente como benignos o malignos. También contiene el vector `clases` que toma los valores 0 o 1 en función de si la correspondiente fila de `datos` corresponde a un tumor benigno o maligno respectivamente. Más información sobre los datos se puede encontrar en esta dirección.

Cargamos los paquetes que vamos a necesitar y los datos. Mediante el comando `head` vemos los datos de algunos de los primeros pacientes del fichero:

```
library(MASS)
library(class)
load(file = 'Wisconsin.Rdata')
head(datos)
```

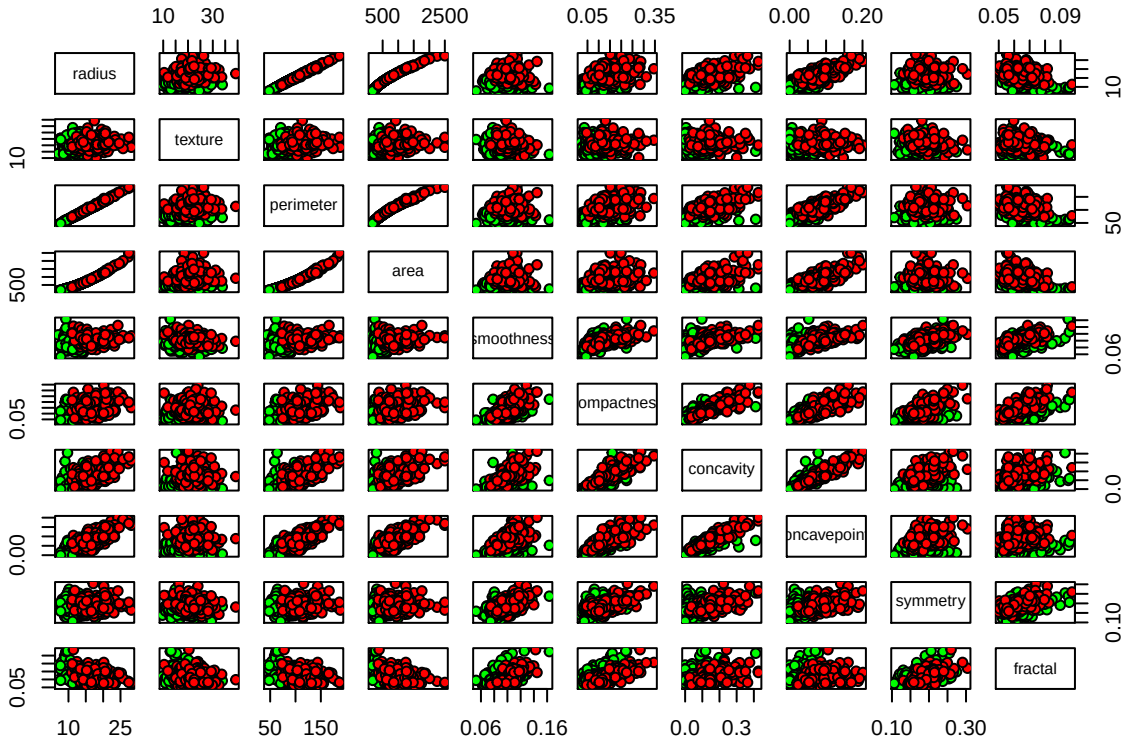
```
##    radius texture perimeter  area smoothness compactness concavity
## 20 13.540   14.36     87.46 566.3    0.09779     0.08129   0.06664
## 21 13.080   15.71     85.63 520.0    0.10750     0.12700   0.04568
## 22  9.504   12.44     60.34 273.9    0.10240     0.06492   0.02956
## 38 13.030   18.42     82.61 523.8    0.08983     0.03766   0.02562
## 47  8.196   16.84     51.71 201.9    0.08600     0.05943   0.01588
## 49 12.050   14.63     78.04 449.3    0.10310     0.09092   0.06592
##    concavepoints symmetry fractal
## 20      0.047810   0.1885 0.05766
## 21      0.031100   0.1967 0.06811
## 22      0.020760   0.1815 0.06905
## 38      0.029230   0.1467 0.05863
## 47      0.005917   0.1769 0.06503
## 49      0.027490   0.1675 0.06043
```

Primero vemos cuántas observaciones tenemos en total y en cada uno de los grupos. Posteriormente representamos la matriz de diagramas de dispersión de los datos y los diagramas de estrella de las observaciones:

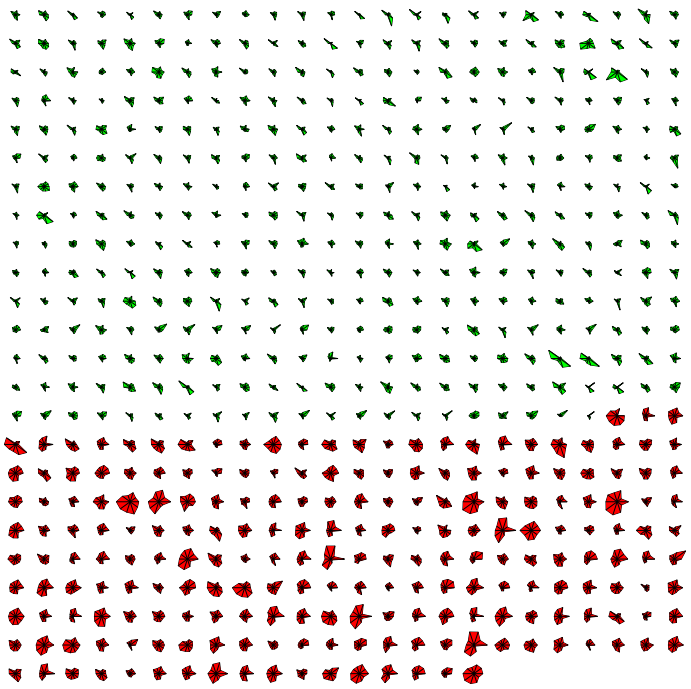
```
# Cuentan el numero de observaciones
n0 <- sum(clases == 0)
n1 <- sum(clases == 1)
n <- n0 + n1

# Para que los graficos queden mas bonitos (rojo = maligno, verde = benigno)
colores <- c(rep('green',n0),rep('red',n1))
pchn <- 21

# Numero de observaciones total y en cada clase
print(c(n, n0, n1))
```

```
## [1] 569 357 212
```

```
# Diagramas de dispersion
pairs(datos, pch = pchn, bg = colores)
```




```
# Diagrama de estrellas
stars(datos, col.stars=colores, labels=NULL)
```



Se dispone en total de 569 imágenes, de las cuáles 357 corresponden a tumores benignos y 212 a malignos. Vemos que algunas variables están estrechamente relacionadas (por ejemplo, el radio, el perímetro y el área). También se observa que en términos generales, los tumores malignos corresponden a valores más altos de las variables.

Regla de clasificación lineal de Fisher

El comando básico es `lda` del paquete `MASS`. Tiene tres argumentos principales: el primero es la matriz o *data frame* con las variables explicativas, el segundo es el vector que contiene la clase a la que pertenece cada observación, el tercero es el vector de probabilidades a priori de cada clase (por defecto, son las frecuencias relativas del vector que contiene las clases). Nosotros vamos a fijar probabilidades a priori iguales.

```
resultado.lda <- lda(datos, clases, prior = c(0.5, 0.5))
```

El objeto `resultado.lda` es una lista con todos los resultados del análisis. Veamos sus elementos más importantes. El vector `resultado.lda$means` contiene los vectores de medias $\bar{x}_0$ y $\bar{x}_1$ correspondientes a cada grupo:

```
resultado.lda$means
```

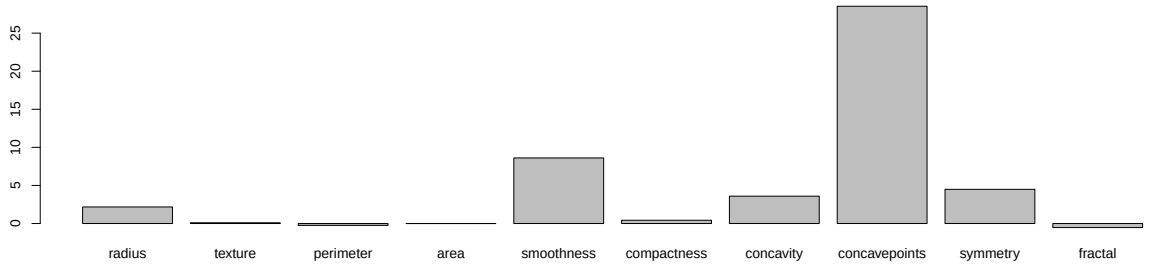
```
##     radius  texture perimeter     area smoothness compactness  concavity
## 0 12.14652 17.91476  78.07541 462.7902 0.09247765  0.08008462 0.04605762
## 1 17.46283 21.60491 115.36538 978.3764 0.10289849  0.14518778 0.16077472
##   concavepoints symmetry    fractal
## 0    0.02571741 0.174186 0.06286739
## 1    0.08799000 0.192909 0.06268009
```

Como habíamos observado en las representaciones gráficas, los valores medios del grupo de tumores malignos son superiores a los del grupo de benignos. El vector `resultado.lda$scaling` contiene los coeficientes de la función discriminante lineal de Fisher:

```
resultado.lda$scaling
```

```
##                       LD1
## radius          2.173832578
## texture         0.097479319
## perimeter      -0.243883158
## area           -0.004235635
## smoothness      8.610211091
## compactness     0.431476344
## concavity       3.592356858
## concavepoints  28.529778564
## symmetry        4.489073661
## fractal        -0.529214778
```

```
barplot(as.vector(resultado.lda$scaling), names.arg = names(datos))
```

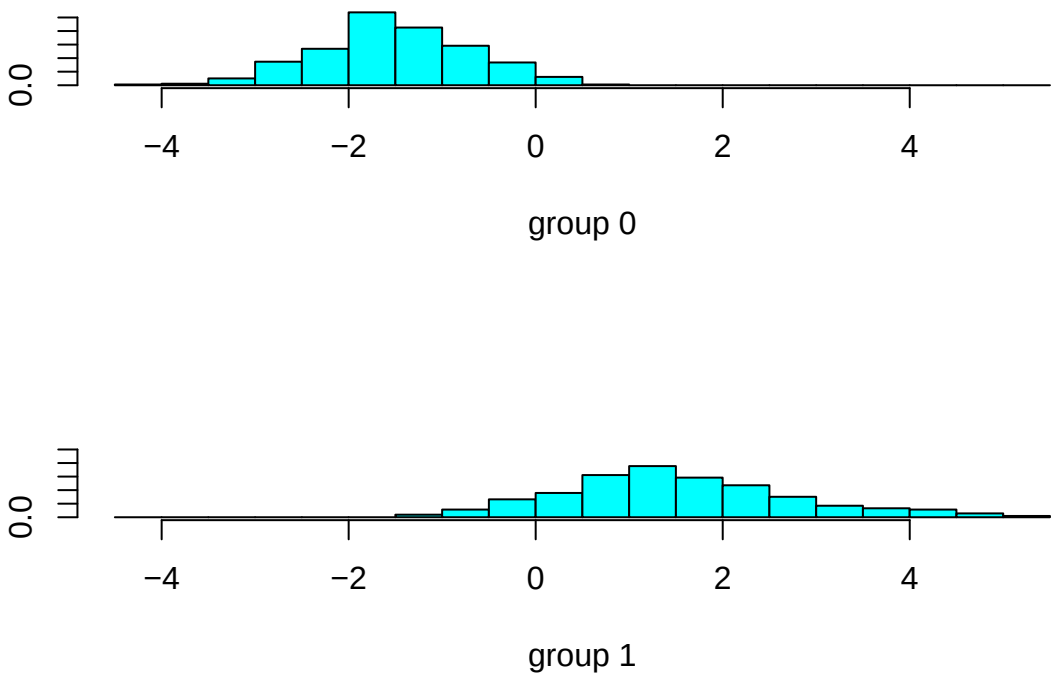


Si llamamos w al vector anterior, clasificamos una observación x en el grupo 1 siempre que

$$w^\top \left(x - \frac{\bar{x}_0 + \bar{x}_1}{2} \right) > 0,$$

y en el grupo 0 en caso contrario. El valor del término de la izquierda de la desigualdad es la puntuación discriminante de x. Al aplicar `plot` a `resultado.lda` obtenemos histogramas de las puntuaciones discriminantes estandarizadas de las observaciones de cada grupo:

```
plot(resultado.lda)
```



Vemos que, efectivamente, las puntuaciones en el grupo 0 tienden a ser negativas y en el grupo 1 tienden a ser positivas. Sin embargo, hay una zona de solapamiento de las puntuaciones discriminantes que llevará a un cierto porcentaje de errores de clasificación. Con el fin de calcular la tasa de error aparente, usamos el comando `predict` para clasificar los datos de la muestra y luego contamos la proporción de veces que nos hemos equivocado:

```
clases.pred <- predict(resultado.lda)$class
1 - sum(clases.pred == clases) / n
```

```
## [1] 0.05975395
```

Esto significa que nos hemos equivocado para aproximadamente un 5.97 % de las observaciones. Para calcular la tasa de error por validación cruzada, seleccionamos `CV=TRUE` en el comando `lda`, de la siguiente forma:

```
clases.pred.cv <- lda(datos, clases, prior=c(0.5, 0.5), CV=TRUE)$class
1 - sum(clases.pred.cv == clases) / n
```

```
## [1] 0.06326889
```

Si queremos clasificar nuevos vectores de observaciones, tenemos que crear previamente un *data frame* que los contenga y luego usar de nuevo el comando `predict`. El siguiente código genera aleatoriamente dos vectores de observaciones y los clasifica. También permite obtener las dos puntuaciones discriminantes correspondientes:

```
# Predicciones
x1 <- rnorm(10)
x2 <- rnorm(10)
nuevas.obs <- data.frame(rbind(x1, x2))
names(nuevas.obs) <- names(datos)
```

```
nuevas.obs
```

```
##       radius    texture perimeter      area  smoothness compactness
## x1 -1.146736  0.8006946 0.5144072 1.0443750 -0.01288357   0.5338722
## x2  1.017993 -0.8110074 2.1510126 0.3612358  0.11572068  -0.4605097
##    concavity concavepoints   symmetry   fractal
## x1 1.2288492     0.6750844 -1.0139854  1.625556
## x2 0.3207167    -0.3409700 -0.6407013 -1.302934
```

```
predict(resultado.lda, nuevas.obs)$class
```

```
## [1] 1 0
## Levels: 0 1
```

```
predict(resultado.lda, nuevas.obs)$x
```

```
##           LD1
## x1   4.694688
## x2 -19.499296
```

Ejercicio

Al mirar la función discriminante de Fisher, vemos que las variables `smoothness` y `concavepoints` son las que reciben una mayor ponderación. Repite los cálculos teniendo en cuenta únicamente estas dos variables. ¿Cuál es la función discriminante lineal de Fisher en este caso? ¿Cuáles son las nuevas tasas de error?

Clasificación cuadrática

El comando básico es `qda` del paquete `MASS`. Tiene dos argumentos principales: el primero es la matriz o *data frame* con las variables explicativas, el segundo es el vector que contiene la clase a la que pertenece cada observación. Al aplicar `predict` al resultado de este comando se obtiene una lista, cuyo elemento `class` nos da el grupo en el que clasificamos cada observación. Esto nos permite calcular la tasa de error aparente:

```
resultado.qda <- qda(datos, clases, prior=c(0.5, 0.5))
clases.qda <- predict(resultado.qda)$class
1-sum(clases.qda == clases) / n
```

```
## [1] 0.05799649
```

Si queremos obtener la tasa de error por validación cruzada, usamos el argumento `CV=TRUE` del comando `qda`, de la siguiente forma:

```
clase.qdacv <- qda(datos, clases, prior=c(0.5,0.5), CV=TRUE)$class
1 - sum(clase.qdacv == clases) / n
```

```
## [1] 0.06678383
```

Los resultados no mejoran (de hecho empeoran ligeramente) al compararlos con los obtenidos mediante la función lineal de Fisher, que además es más fácil de interpretar.

Para clasificar nuevas observaciones, se procede de forma análoga a como se hacía en clasificación lineal:

```
predict(resultado.qda, nuevas.obs)$class
```

```
## [1] 0 0
## Levels: 0 1
```

Ejercicio

Repite los cálculos utilizando únicamente las variables `smoothness` y `concavepoints`.

Regresión logística

En 1986, el transbordador espacial *Challenger* tuvo un accidente catastrófico debido a un incendio en una de las piezas de sus propulsores. Era la vez 25 en que se lanzaba un transbordador espacial. En todas las ocasiones anteriores se habían inspeccionado los propulsores de las naves, y en algunas de ellas se habían encontrando defectos. El fichero `challenger` contiene 23 observaciones de las siguientes variables: `defecto`, que toma los valores 1 y 0 en función de si se encontraron defectos o no en los propulsores; y `temp`, la temperatura (en grados Fahrenheit) en el momento del lanzamiento.

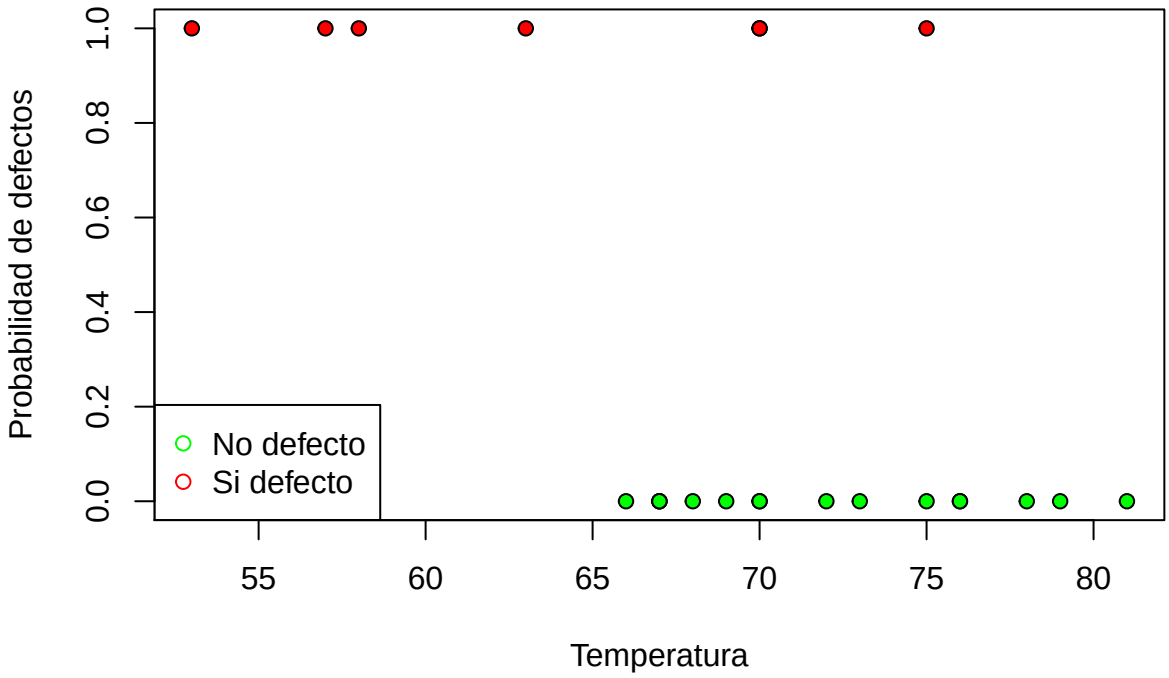
Primero leemos los datos y contamos las frecuencias de casos sin y con defectos:

```
challenger <- read.table('http://www.uam.es/joser.berrendero/datos/challenger.txt', header = TRUE)
table(challenger$defecto)
```

```
##
##  0  1
## 16  7
```

Una representación gráfica de los datos, puede obtenerse mediante:

```
colores <- NULL
colores[challenger$defecto==0] <- 'green'
colores[challenger$defecto==1] <- 'red'
plot(challenger$temp, challenger$defecto, pch = 21, bg = colores, xlab = 'Temperatura', ylab = 'Probabi
legend('bottomleft', c('No defecto', 'Si defecto'), pch = 21, col = c('green', 'red'))
```



Hemos usado los argumentos `pch` y `bg` para mejorar la apariencia del gráfico. También hemos usado el comando `legend` para incluir una leyenda explicativa.

Parece razonable, a la vista de los datos, pensar que la temperatura puede influir en la probabilidad de que los propulsores tengan defectos. En esta práctica, vamos a ajustar un modelo de regresión logística para estudiar la posible relación. Para ajustar el modelo se usa el comando `glm` (para modelos lineales generalizados) indicando que la respuesta es binomial mediante el argumento `family`:

```
reg <- glm(defecto ~ temp, data = challenger, family=binomial)
summary(reg)
```

```
##
## Call:
## glm(formula = defecto ~ temp, family = binomial, data = challenger)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0611  -0.7613  -0.3783   0.4524   2.2175
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## temp         -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

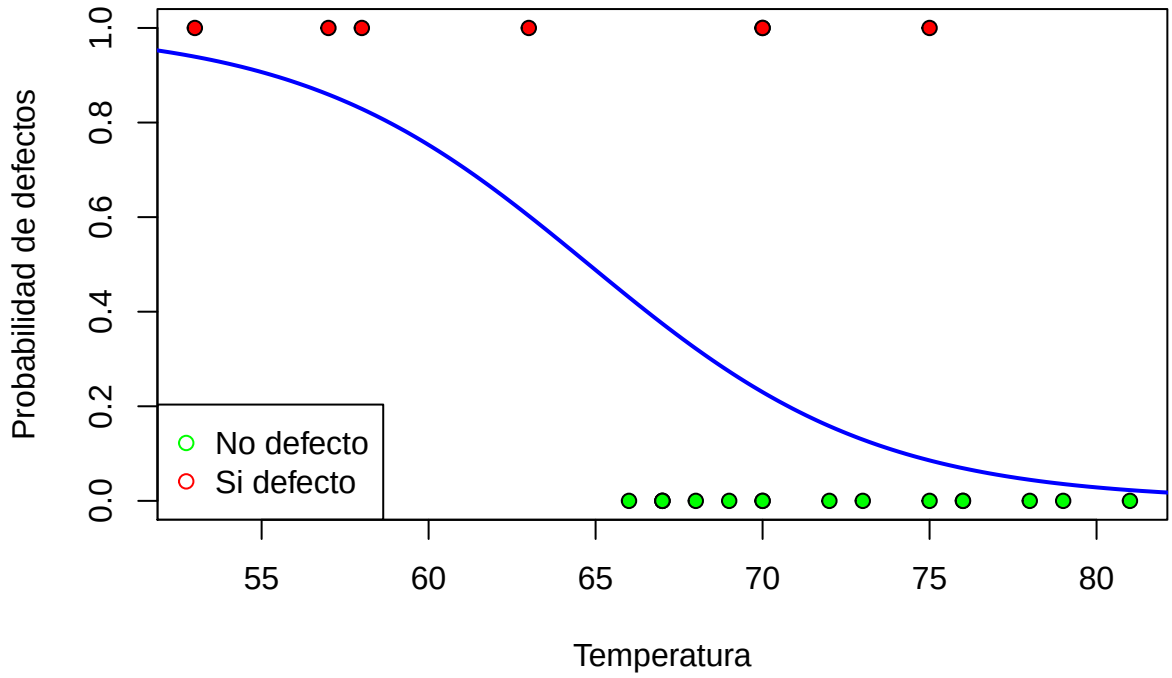
En el modelo de regresión logística la raíz de las desviaciones representa el papel de los residuos:

$$D_i = \mp\sqrt{-2[Y_i \log \hat{p}_i + (1 - Y_i) \log(1 - \hat{p}_i)]},$$

donde el signo coincide con el signo de $Y_i - \hat{p}_i$. En la salida anterior estas cantidades se denominan deviance residuals. Para calcular estos pseudo-residuos, podemos ejecutar `res = resid(reg)`.

Para representar gráficamente la función logística estimada, calculamos las probabilidades de fallo estimadas (usando el comando `predict`) para un vector adecuado de nuevas temperaturas (entre 50 y 85 grados):

```
datos <- data.frame(temp = seq(50, 85, 0.1))
probabilidades <- predict(reg, datos, type = 'response')  # por defecto calcularía log p_i/(1-p_i), par
plot(challenger$temp, challenger$defecto, pch = 21, bg = colores, xlab = 'Temperatura', ylab = 'Probabil
legend('bottomleft', c('No defecto', 'Si defecto'), pch = 21, col = c('green', 'red'))
lines(datos$temp, probabilidades, col = 'blue', lwd = 2)
```



Ejercicios

1. ¿Se puede afirmar a nivel $\alpha = 0.05$ que la temperatura influye en la probabilidad de que los propulsores tengan defectos? ¿Y a nivel $\alpha = 0.01$? Usa el test de Wald.
2. Interpreta el valor del coeficiente estimado para la variable temperatura: $\hat{\beta}_1 = -0.2322$.

3. ¿Para qué valores de la temperatura la probabilidad estimada de que se produzcan defectos es menor que 0.1?

4. ¿Para qué valores de la temperatura se predice que se van a producir defectos?