Departamento de Estadística e I.O.

Máster en Estadística Aplicada

UNIVERSIDAD
DE GRANADA

MODELOS DE RESPUESTA DISCRETA
APLICACIONES BIOSANITARIAS

Tema 2 de prácticas
Ajuste de regresión logística
con variables explicativas cuantitativas con R

Profesores

Ana María Aguilera del Pino

Manuel Escabias Machuca

Título original: Modelos de Respuesta Discreta. Aplicaciones Biosanitarias. Tema 2 de prácticas: Ajuste de regresión logística con variables explicativas cuantitativas con R

Contents

Chapter 1

Ajuste de regresión logística con variables explicativas cuantitativas con R

1.1 Introducción

En este documento se explican con cierto detalle algunas funciones de `R` de utilidad para un análisis de regresión logística simple.

Se asume en este tema que el lector está familiarizado con la filosofía y la práctica del trabajo con R, esto es, con el uso de los principales tipos de objetos de R. Así, se asume que el lector conoce:

- El símbolo $<-$ (operador para asignar).
- Objetos simples: numérico, carácter y lógico.
- Objeto vector tanto numérico, lógico como carácter y factor; su definición, asignación, acceso a sus elementos (`[]`) y las funciones de operaciones básicas como longitud (`length()`), ordenar (`sort()` y `order()`), operaciones matemáticas (+, -, *, /), funciones estadísticas (`mean()` y `var()`...)
- Objeto matriz; su definición, asignación, acceso a sus elementos (`[,]`) y las funciones de operaciones básicas (+, -, *, /, %*%)
- Objeto lista; su definición, asignación, acceso a sus elementos (`[[]]` y $) y las funciones de operaciones básicas
- Data Frame; su definición, asignación, acceso a sus elementos (`[,]` y $) y las funciones de operaciones básicas (`class(), nrow(), ncol(), dim(), head() tail()`)

El tema se divide en cuatro secciones bien diferenciadas:

- Ajuste de un modelo logit simple con datos agrupados.
- Ajuste de un modelo logit simple con datos no agrupados.
- Ajuste de un modelo logit múltiple.

En cada uno de los ajustes se abordarán los siguientes apartados:

- El formato de datos.
- Ajuste del modelo.
- Predicción e interpretación de parámetros.
- Bondad del ajuste.
- Significación de parámetros.
- Validación.
- Precisión del modelo logístico y clasificación de observaciones.

1.2 Ajuste de un modelo logit simple con datos agrupados

Recuérdese que el objetivo de la regresión logística simple es modelizar una variable respuesta binaria Y a partir de una variable explicativa X (en este caso cuantitativa).

1.2.1 Formato de datos

Para ilustrar el uso de las funciones necesarias en un análisis de regresión logística binaria en esta situación de datos agrupados con `R` se va a utilizar el siguiente ejemplo.

Ejemplo. Se sabe que la presión arterial de las personas es un factor determinante para padecer problemas coronarios. Se realizó un estudio en un conjunto de personas a las que se midió la presión arterial y se registró si habían padecido problemas coronarios. El número de personas con y sin problemas coronarios para cada nivel de presión arterial fue registrado

Presión	Cor=0	Cor=1	Presión	Cor=0	Cor=1
55	1	0	105	0	1
60	4	0	110	2	2
65	2	0	88	8	0
66	1	0	90	31	6
68	4	0	92	2	0
70	23	4	94	2	1
74	2	1	95	0	1
75	12	0	96	1	0
76	1	0	100	3	1
78	8	1	104	2	0
80	53	5	105	0	1
82	6	1	110	2	2
84	1	0	112	1	0

siendo Cor=0 y Cor=1 el número de sujetos sin y con problemas coronarios, respectivamente, para la persión arterial indicada.

La información de este conjunto de datos está en el formato que se denominará *datos agrupados* o en *tabla de frecuencias*, esto es, para cada observación de la variable explicativa (presión arterial) se tiene el número de éxitos (problemas coronarios) y el número de fracasos (no problemas coronarios) de la variable respuesta.

El formato de los datos en R. Para un ajuste de regresión logística simple con datos agrupados, los datos deben estar registrados en un *Data.Frame* con tres columnas, similarmente a como se muestran en el enunciado del ejercicio, cada columna con su nombre (preferiblemente que sea descriptivo del contenido de la columna).

Asumimos en este ejemplo que diponemos de un *Data.Frame* denominado `Presion` con las mismas tres columnas que se muestran en el enunciado y cuyos nombres son `Diastolica`, `Coronarios0` y `Coronarios1`:

```
##     Diastolica Coronarios0 Coronarios1
## 55          55           1           0
## 60          60           4           0
## 65          65           2           0
## 66          66           1           0
## 68          68           4           0
## 70          70          23           4
## 74          74           2           1
## 75          75          12           0
## 76          76           1           0
## 78          78           8           1
```

```
## 80          80          53          5
## 82          82           6          1
## 84          84           1          0
## 85          85           4          1
## 86          86           0          1
## 88          88           8          0
## 90          90          31          6
## 92          92           2          0
## 94          94           2          1
## 95          95           0          1
## 96          96           1          0
## 100        100           3          1
## 104        104           2          0
## 105        105           0          1
## 110        110           2          2
## 112        112           1          0
```

Este *Data.Frame* puede haber sido generado a partir de la lectura en `R` de un fichero de tipo *.csv* con las tres columnas indicadas anteriormente o bien generándolo mediante la sentencia `data.frame()` y los vectores descritos en el enunciado.

Antes de abordar el ajuste del modelo logit a este conjunto de datos, veamos brevemente algunas identificaciones de los aspectos teóricos del tema con los datos disponibles.

La formulación del modelo logístico binario con datos agrupados es del siguiente modo:

$$y_q \rightsquigarrow B(n_q, p(x_q)),\ p(x_q) = \frac{e^{\beta_0+\beta_1 x_q}}{1+e^{\beta_0+\beta_1 x_q}}.\ q = 1, \ldots, Q$$

donde

- Q es el número de observaciones diferentes de la variable explicativa X
- $p(x_q)$ es la probabilidad de que la respuesta tome el valor $Y = 1$ para una observación de la variable explicativa $X = x_q$
- β_0 y β_1 son los parámetros a estimar.
- y_q representa en número de éxitos de entre n_q observaciones de la respuesta Y cada una de las cuales tiene probabilidade éxito $p(x_q)$

Así, con los datos en este formato podemos realizar las siguientes identificaciones con los aspectos teóricos:

- $Q = 26$
- $N = n_1 + n_2 + \cdots + n_{26}$ =1+4+$\cdots$+1=200

$X = x_q$	$n_q - y_q$	y_q	n_q
x_1 =55	1	y_1 =0	n_1 =1
x_2 =60	4	y_2 =0	n_2 =4
x_3 =65	2	y_3 =0	n_3 =2
x_4 =66	1	y_4 =0	n_4 =1
x_5 =68	4	y_5 =0	n_5 =4
$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{22} =100	3	y_{22} =1	n_{22} =4
x_{23} =104	2	y_{23} =0	n_{23} =2
x_{24} =105	0	y_{24} =1	n_{24} =1
x_{25} =110	2	y_{25} =2	n_{25} =4
x_{26} =112	1	y_{26} =0	n_{26} =1
			N =200

1.2.2 Ajuste del modelo

El ajuste del modelo de regresión logística en `R` se hace a través de la función `glm()` que es una función para el ajuste de un modelo lineal generalizado. Los argumentos más importantes de esta función son `formula`, `family` y `data`.

- El argumento `formula` es ampliamente usado en la modelización con R. La sintaxis de `formula` tiene tres partes: el lado izquierdo, el símbolo $\sim$ y el lado derecho. En el lado izquierdo se especifica el nombre de la variable respuesta. El símbolo $\sim$ se usa como separador. El lado derecho de una fórmula es una expresión que incluye los nombres de las variables predictoras separadas por el símbolo `+`.
- El argumento `family` sirve para indicar el componente aleatorio del modelo lineal generalizado, así como la función de enlace (link) que se utilizará. Para el caso de regresión logística se utiliza `family=binomial`.
- El argumento `data` es para especificar el nombre del *Data.Frame* que contiene los datos del ajuste y que denominaremos aquí, *Data.Frame* **del ajuste**.

Para ajustar un modelo de regresión logística con `R` es conveniente asignar un nombre al ajuste. El resultado es que `R` crea lo que llamaremos

un objeto de tipo *glm* (o familiarmente un objeto de tipo regresión logística). Un objeto de tipo *glm* es una lista de R con diferentes elementos que se irán explicando a lo largo de este documento. Para nuestro *Data.Frame* del ajuste (denominado `Presion`) se tendría:

```
Ajuste.Presion<-glm(cbind(Coronarios1,Coronarios0)~Diastolica,
data=Presion,family = binomial)
```

De este modo se ha creado el objeto `Ajuste.Presion` que es de tipo *glm* y que tiene todos los elementos necesarios para hacer el estudio de nuestros datos mediante regresión logística.

El resumen del modelo ajustado se obtiene con la sentencia `summary()` aplicada al objeto de tipo *glm* que hemos denominado `Ajuste.Presion`:

```
summary(Ajuste.Presion)

##
## Call:
## glm(formula = cbind(Coronarios1, Coronarios0) ~ Diastolica, family = binom
##     data = Presion)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6594  -0.6881  -0.3080   0.4915   1.9642
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.05492    1.75777  -3.445 0.000572 ***
## Diastolica   0.04980    0.02049   2.430 0.015085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.330  on 25  degrees of freedom
## Residual deviance: 24.411  on 24  degrees of freedom
## AIC: 50.841
##
## Number of Fisher Scoring iterations: 4
```

En esta salida se pueden observar entre otros los siguientes elementos:

- Un resumen de los residuos de la deviance (se explicarán más adelante) mediante su valor mínimo (-1.65944), sus tres cuartiles (Q_1 =-0.68814, Q_2 =-0.30802, Q_3 =0.49154) y su valor máximo (1.96416).
- Estimación puntual del parámetro independiente: $\widehat{\beta}_0$ =-6.05492
- Error estándar de estimación del parámetro independiente: $\widehat{S.E}(\widehat{\beta}_0)$ =1.75777
- Estimación puntual del parámetro asociado a la variable explicativa: $\widehat{\beta}_1$ =0.0498
- Error estándar de estimación del parámetro asociado a la variable explicativa: $\widehat{S.E}(\widehat{\beta}_1)$ =0.02049

1.2.3 Predicción e interpretación de parámetros

Aunque la salida anterior no lo muestra, la estimación puntual de los parámetros del modelo permite obtener una estimación puntual de las probabilidades de éxito (probabilidad de padecer problemas coronarios) para los distintos valores de la variable explicativa, sin más que sustituir en la ecuación del modelo:

$$\widehat{p}(x_q) = \frac{e^{\widehat{\beta}_0+\widehat{\beta}_1 x_q}}{1+e^{\widehat{\beta}_0+\widehat{\beta}_1 x_q}}$$

`R` calcula estas probabilidades mediante la sentencia `fitted.values(Ajuste.Presion)` (recordemos que `Ajuste.Presion` es el nombre que le dimos al objeto tipo *glm* para realizar el ajuste).

```
fitted.values(Ajuste.Presion)

##         55         60         65         66         68         70
## 0.03503277 0.04449767 0.05637035 0.05907873 0.06486489 0.07117488
##         74         75         76         78         80         82
## 0.08552255 0.08949874 0.09364086 0.10244348 0.11197136 0.12226467
##         84         85         86         88         90         92
## 0.13336212 0.13922389 0.14530011 0.15811176 0.17182596 0.18646623
##         94         95         96        100        104        105
## 0.20204958 0.21019816 0.21858534 0.25450621 0.29410831 0.30455257
##        110        112
## 0.35969184 0.38293764
```

El resultado de esta sentencia es un vector con las probabilidades estimadas por el modelo. En la siguiente tabla se muestran estas probabilidades junto

con los valores observados de las variables y las ventajas tanto de respuesta 1 ($\frac{\widehat{p}(x_q)}{1-\widehat{p}(x_q)}$) como de respuesta 0 ($\frac{1-\widehat{p}(x_q)}{\widehat{p}(x_q)}$):

$X = x_q$	$n_q - y_q$	y_q	n_q	$\widehat{p}(x_q)$	$\frac{\widehat{p}(x_q)}{1-\widehat{p}(x_q)}$	$\frac{1-\widehat{p}(x_q)}{\widehat{p}(x_q)}$
55	1	0	1	**0.035**	**0.0363**	27.5447
60	4	0	4	0.0445	0.0466	21.4731
65	2	0	2	0.0564	0.0597	16.7398
66	1	0	1	0.0591	0.0628	15.9266
68	4	0	4	0.0649	0.0694	14.4167
...	...	...	...	...	...	...
100	3	1	4	0.2545	0.3414	2.9292
104	**2**	**0**	**2**	**0.2941**	**0.4166**	**2.4001**
105	**0**	**1**	**1**	**0.3046**	**0.4379**	**2.2835**
110	2	2	4	0.3597	0.5617	1.7802
112	1	0	1	**0.3829**	0.6206	1.6114

De estas probabilidades podemos sacar algunas interpretaciones sobre la persona que sustenta una familia:

- A las personas con presión arterial de 55 el modelo le asigna una probabilidad de padecer problemas coronarios de 0.035.
- A las personas con presión arterial de 112 el modelo le asigna una probabilidad de padecer problemas coronarios de 0.3829.
- A las personas con presión arterial de 104 el modelo le asigna:
  - una probabilidad de padecer problemas coronarios de 0.2941
  - una probabilidad de no padecer problemas coronarios de $1-\widehat{p}(104)$ =0.7059.
  - una ventaja de padecer problemas coronarios frente a no padecerlos de
    $$\frac{\widehat{p}(104)}{1-\widehat{p}(104)} = 0.4166,$$
    y por tanto resulta 0.4166 veces más (menos) probable padecer problemas coronarios que no padecerlos.
  - una ventaja de no padecer problemas coronarios frente a padecerlos de
    $$\frac{1-\widehat{p}(104)}{\widehat{p}(104)} = 2.4001,$$
    y por tanto resulta 2.4001 veces más probable no padecer problemas coronarios que padecerlos.

- Para las personas con presión arterial de 105 la ventaja de padecer problemas coronarios frente a no padecerlos es 0.4379 (0.3046/0.6954).
- Al aumentar en una unidad la presión arterial (de 104 a 105) la ventaja pasa de 0.4166 a 0.4379, esto es, se multiplica por 1.0511 veces (0.4379/0.4166)
- El cambio anterior coincide con la exponencial del parámetro asociado a la presión arterial $e^{\widehat{\beta}_1} = e^{0.0498}$ =1.0511
- Para una persona con presión arterial 0 el modelo le asignaría una ventaja de padecer problemas coronarios frente a no padecerlos $\frac{\widehat{p}(0)}{1-\widehat{p}(0)}$ y coincidiría con la exponencial del parámetro independiente $e^{\widehat{\beta}_0} = e^{-6.0549}$ =0.0023. Sin embargo en este ejemplo no tiene sentido hacer esta interpretación pues no puede haber nadie con presión arterial 0.

Además de con la función `fitted.values()`, también se pueden obtener diferentes valores predichos con la función `predict()`. Se trata de una función genérica de `R` que se utiliza para obtener predicciones de varios modelos. Los argumentos más importantes son `object` y `type` dónde se indica el objeto tipo *glm* del ajuste y el tipo de predicción a obtener:

- Para obtener las probabilidades predichas se utiliza (`type="response"`)

```
predict(Ajuste.Presion,type= "response")

##         55         60         65         66         68         70
## 0.03503277 0.04449767 0.05637035 0.05907873 0.06486489 0.07117488
##         74         75         76         78         80         82
## 0.08552255 0.08949874 0.09364086 0.10244348 0.11197136 0.12226467
##         84         85         86         88         90         92
## 0.13336212 0.13922389 0.14530011 0.15811176 0.17182596 0.18646623
##         94         95         96        100        104        105
## 0.20204958 0.21019816 0.21858534 0.25450621 0.29410831 0.30455257
##        110        112
## 0.35969184 0.38293764
```

que coincide con el resultado de `fitted.values()`

- Para obtener una estimación del predictor lineal dado por $\widehat{\beta}_0 + \widehat{\beta}_1 x_p$ se utiliza (`type="link"`)

```
predict(Ajuste.Presion,type = "link")

##         55         60         65         66         68         70
## -3.3158101 -3.0668003 -2.8177904 -2.7679885 -2.6683845 -2.5687806
##         74         75         76         78         80         82
## -2.3695727 -2.3197708 -2.2699688 -2.1703649 -2.0707609 -1.9711570
##         84         85         86         88         90         92
## -1.8715530 -1.8217511 -1.7719491 -1.6723452 -1.5727412 -1.4731373
##         94         95         96        100        104        105
## -1.3735334 -1.3237314 -1.2739294 -1.0747216 -0.8755137 -0.8257117
##        110        112
## -0.5767019 -0.4770980
```

El valor por defecto es para `type="link"`

En todos los casos el resultado es un vector de igual dimensión al número de filas del *Data.Frame* del modelo, esto es, el primer elemento del vector es la predicción para el valor de la variable explicativa de la primera fila del data.frame, el segundo elemento del vector es la predicción para el valor de la variable explicativa de la segunda fila del data.frame, y así sucesivamente.

Cuando en la función `predict()` se incluye la opción `se.fit=TRUE` además de los vectores anteriores se obtiene un segundo vector con los errores estándar de estimación. En este caso, la función `predict()` devuelve una lista con dos elementos: el vector de las estimaciones (llamado `$fit`) y el vector de los errores estándar (llamado `$se.fit`).

```
predict(Ajuste.Presion,type= "response",se.fit = T)

## $fit
##         55         60         65         66         68         70
## 0.03503277 0.04449767 0.05637035 0.05907873 0.06486489 0.07117488
##         74         75         76         78         80         82
## 0.08552255 0.08949874 0.09364086 0.10244348 0.11197136 0.12226467
##         84         85         86         88         90         92
## 0.13336212 0.13922389 0.14530011 0.15811176 0.17182596 0.18646623
##         94         95         96        100        104        105
## 0.20204958 0.21019816 0.21858534 0.25450621 0.29410831 0.30455257
##        110        112
## 0.35969184 0.38293764
##
## $se.fit
```

```
##         55         60         65         66         68         70
## 0.02209615 0.02371684 0.02472613 0.02483382 0.02494401 0.02491134
##         74         75         76         78         80         82
## 0.02446395 0.02429801 0.02412694 0.02382445 0.02370182 0.02396348
##         84         85         86         88         90         92
## 0.02485955 0.02562606 0.02664285 0.02950979 0.03356242 0.03881375
##         94         95         96        100        104        105
## 0.04521952 0.04883341 0.05270784 0.07058933 0.09168926 0.09736081
##        110        112
## 0.12723507 0.13952014
##
## $residual.scale
## [1] 1
```

```
predict(Ajuste.Presion,type= "link",se.fit = T)
```

```
## $fit
##         55         60         65         66         68         70
## -3.3158101 -3.0668003 -2.8177904 -2.7679885 -2.6683845 -2.5687806
##         74         75         76         78         80         82
## -2.3695727 -2.3197708 -2.2699688 -2.1703649 -2.0707609 -1.9711570
##         84         85         86         88         90         92
## -1.8715530 -1.8217511 -1.7719491 -1.6723452 -1.5727412 -1.4731373
##         94         95         96        100        104        105
## -1.3735334 -1.3237314 -1.2739294 -1.0747216 -0.8755137 -0.8257117
##        110        112
## -0.5767019 -0.4770980
##
## $se.fit
##        55        60        65        66        68        70        74
## 0.6536264 0.5578119 0.4648404 0.4467444 0.4112275 0.3768221 0.3128045
##        75        76        78        80        82        84        85
## 0.2981764 0.2842736 0.2591056 0.2383679 0.2232983 0.2150915 0.2138345
##        86        88        90        92        94        95        96
## 0.2145365 0.2216907 0.2358538 0.2558644 0.2804737 0.2941508 0.3085834
##       100       104       105       110       112
## 0.3720460 0.4416448 0.4596821 0.5524427 0.5904455
##
## $residual.scale
## [1] 1
```

Estos errores de estimación $\widehat{SE}(\widehat{p}(x_q))$ permitirían obtener intervalos de confianza para las predicciones. No se debe olvidar que las probabilidades $p(x_q)$ son también parámetros de los que se pueden obtener estimaciones puntuales, intervalos de confianza y hacer contrastes de hipótesis, para los cuales se necesita conocer su error estándar de estimación.

Otro parámetro adicional que se puede incluir en la función `predict()` es el parámetro `newdata`, un nuevo *Data.Frame* con observaciones de la variable explicativa para los que predecir la probabilidad de éxito. En este caso este Data.Frame debe tener al menos una columna con los valores de la variable explicativa y con el mismo nombre que tenía dicha variable en el *Data.Frame* del ajuste. El resultado en este caso será un vector de igual longitud que número de filas tiene el nuevo *Data.Frame*. A continuación se muestra un ejemplo de la predicción que daría el modelo para personas con presiones arteriales de 69, 98 o 107 (valores no observados en la muestra):

```
Nuevas.Presion<-data.frame(c(69,98,107))
names(Nuevas.Presion)<-"Diastolica"
Nuevas.Presion

##   Diastolica
## 1         69
## 2         98
## 3        107

predict(Ajuste.Presion,newdata = Nuevas.Presion,type= "response",se.fit = T)

## $fit
##          1          2          3
## 0.06795201 0.23607402 0.32604943
##
## $se.fit
##          1          2          3
## 0.02494541 0.06119488 0.10906388
##
## $residual.scale
## [1] 1

predict(Ajuste.Presion,newdata = Nuevas.Presion,type= "link",se.fit = T)

## $fit
##          1          2          3
```

```
## -2.6185826 -1.1743255 -0.7261078
##
## $se.fit
##         1         2         3
## 0.3938675 0.3393248 0.4963288
##
## $residual.scale
## [1] 1
```

1.2.4 Bondad del ajuste

El test de bondad de ajuste trata de determinar si el modelo logístico es o no adecudado para modelizar los datos, en definitiva si

$$H_0 \ : \ p(x_q) = \frac{e^{\beta_0+\beta_1 x_q}}{1+e^{\beta_0+\beta_1 x_q}}$$
$$H_1 \ : \ p(x_q) \neq \frac{e^{\beta_0+\beta_1 x_q}}{1+e^{\beta_0+\beta_1 x_q}}$$

Existen varias formas de resolver el test bondad del ajute de un modelo de regresión logística: estadístico chi-cuadrado de Pearson de bondad de ajuste, estadístico de Wilks de razón de verosimilitudes y el estadístico de Hosmer y Lemeshow. Los tres estadísticos tienen ciertas restricciones para ser aplicados, que deberían ser testadas por el usuario antes de su uso, puesto que ningún software garantiza que se cumplan estas restricciones.

El único estadístico de los indicados anteriormente que proporciona de manera automática la función `glm`, es el estadístico de Wilks de razón de verosimilitudes a través de la función `summary()`

```
summary(Ajuste.Presion)

##
## Call:
## glm(formula = cbind(Coronarios1, Coronarios0) ~ Diastolica, family = binomial,
##     data = Presion)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6594  -0.6881  -0.3080   0.4915   1.9642
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.05492    1.75777  -3.445 0.000572 ***
## Diastolica   0.04980    0.02049   2.430 0.015085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.330  on 25  degrees of freedom
## Residual deviance: 24.411  on 24  degrees of freedom
## AIC: 50.841
##
## Number of Fisher Scoring iterations: 4
```

En esta salida se muestran los siguientes elementos:

- Valor experimental del test de bondad de ajuste del modelo que sólo tiene el parámetro independiente:

$$H_0 : p(x_q) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$
$$H_1 : p(x_q) \neq \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$

  Su valor es 30.3302 y la salida del ajuste lo denomina `Null deviance`. Dado que la distribución del estadístico es Chi-Cuadrado y sus grados de libertad son 25, se puede calcular su p-valor como $P\{X > 30.3302\} = 0.2121612$ siendo X una variable Chi-cuadrado con 25 grados de libertad. Como se puede ver el test es no significativo y por tanto es preferible el modelo logit con únicamente el parámetro independiente al modelo saturado.

- Valor experimental del test de bondad de ajuste del modelo que tiene todos los parámetros del modelo

$$H_0 : p(x_q) = \frac{e^{\beta_0+\beta_1 x_q}}{1 + e^{\beta_0+\beta_1 x_q}}$$
$$H_1 : p(x_q) \neq \frac{e^{\beta_0+\beta_1 x_q}}{1 + e^{\beta_0+\beta_1 x_q}}$$

Su valor es 24.4115 y la salida del ajuste lo denomina `Residual deviance`. Dado que la distribución del estadístico es Chi-Cuadrado y sus grados de libertad son 24, se puede calcular su p-valor como $P\{X > 24.4115\} = 0.4382813$ siendo X una variable Chi-cuadrado con 24 grados de libertad. Como se puede ver el test es no significativo y por tanto es preferible el modelo logístico al saturado.

Para asegurarse que este estadístico sigue distribución Chi-Cuadrado hay que asegurarse de que las frecuencias esperadas $n_q\widehat{p}_q$ son mayores que 5 en el 80% de los casos. Para ello se pueden ver cuántas frecuencias esperadas son mayores que 5 con la sentencia:

```
(Presion$Coronarios0+Presion$Coronarios1)*fitted.values(Ajuste.Presion)
```

```
##         55         60         65         66         68         70
## 0.03503277 0.17799070 0.11274070 0.05907873 0.25945956 1.92172165
##         74         75         76         78         80         82
## 0.25656765 1.07398486 0.09364086 0.92199133 6.49433862 0.85585269
##         84         85         86         88         90         92
## 0.13336212 0.69611945 0.14530011 1.26489405 6.35756043 0.37293246
##         94         95         96        100        104        105
## 0.60614873 0.21019816 0.21858534 1.01802484 0.58821663 0.30455257
##        110        112
## 1.43876737 0.38293764
```

Se puede ver que en este caso sólo 2 de los 26 casos cumplen la condición y por tanto no se debería utilizar este estadístico para estudiar la bondad del ajuste.

La alternativa al estadístico de Wilks cuando no se verifica que el 80% de las frecunencias esperadas son mayores que 5, es el test de Hosmer y Lemeshow.

La función `glm()` no proporciona el estadístico del test de Hosmer y Lemeshow (tampoco el estadístico Chi-Cuadrado de Pearson). Se pueden encontrar algunas funciones de R que permiten el cálculo de estos estadísticos. También existen algunos paquetes de R que también incluyen esta facilidad como el paquete *ResourceSelection* que proporciona una función para el cálculo del Test de Hosmer y Lemeshow. Tanto cuando se utilicen paquetes de R como funciones compartidas por la comunidad científica, debemos asegurarnos de comprender bien el formato de entrada de datos y las salidas que proporcionan.

Los profesores de la asignatura han encontrado una función de R que permite calcular el estadístico y el p-valor del Test de Hosmer y Lemeshow.

Esta función estaba inicialmente diseñada para el caso de datos no agrupados (se verá en la siguiente sección) y ha sido modificada convenientemente para su uso también con datos agrupados.

```
-------------------------------------------------------------------------------
hosmerlem.test <- function(y, yhat, g=10, group=F)
{
  colnum<-ncol(y)
  if(group==F)
  {
    cutyhat1 = cut(yhat,breaks =unique(quantile(yhat, probs=seq(0,1, 1/g))), inclu
    obs = xtabs(cbind(1 - y[,colnum], y[,colnum]) ~ cutyhat1)
    expect = xtabs(cbind(1 - yhat, yhat) ~ cutyhat1)
  }
  else
  {
    y2<-c(rep(seq(0,0,length=nrow(y)),y[,colnum-1]),rep(seq(1,1,length=nrow(y)),y[
    yhat2<-c(rep(yhat,y[,colnum-1]),rep(yhat,y[,colnum]))
    cutyhat1 = cut(yhat2,breaks =unique(quantile(yhat2, probs=seq(0,1, 1/g))), incl
    obs = xtabs(cbind(1 - y2, y2) ~ cutyhat1)
    expect = xtabs(cbind(1 - yhat2, yhat2) ~ cutyhat1)
  }
  chisq.C = sum((obs - expect)^2/expect)
  P.C = 1 - pchisq(chisq.C, g - 2)
  res <- data.frame(c(chisq.C,P.C))
  colnames(res)<- c("Hosmer-Lemeshow Test")
  rownames(res)<- c("X-squared","p.value")
  return(res)
}
-------------------------------------------------------------------------------
```

Para el ejemplo que nos ocupa (datos agrupados) la función está diseñada para que los datos estén dispuestos en un *Data.Frame* en el que las dos últimas columnas tengan el número de facasos y el número de éxitos respectivamente para cada combinación de valores de la/s variable/s explicativa/s.

Para usar la función, los parámetros necesarios son:

- *Data.frame* del ajuste.
- Vector con las probabilidades predichas por el modelo (obtenido a partir de la función `fitted.values()`)
- Número de grupos a considerar (g=10 es el valor por defecto).

- Valor lógico indicando si se trata de datos agrupados ((TRUE)) o no agrupados (`FALSE`), éste último es el valor por defecto.

Antes de poder ejecutar la función es necesario que *compilar* la función abriendo el fichero que la contiene, con RStudio y pulsando la techal *Source*.

En nuestro ejemplo el test de bondad de ajuste de Hosmer y Lemeshow para 10 grupos ofrece el siguiente resultado:

```
hosmerlem.test(Presion,fitted.values(Ajuste.Presion),g=10,group=T)
```

```
##           Hosmer-Lemeshow Test
## X-squared            4.4532204
## p.value              0.8140948
```

Este resultado conduce a no rechazar la hipótesis nula del test de bondad del ajuste del modelo de regresión logística, y por tanto a aceptar que el modelo logístico se ajusta bien a los datos.

1.2.5 Significación de parámetros

Con el estudio de la significación de parámetros se pretende contrastar si

$$\begin{array}{ll} H_0 : \beta_0 = 0 & H_0 : \beta_1 = 0 \\ H_1 : \beta_0 \neq 0 & H_1 : \beta_1 \neq 0 \end{array}$$

El resumen del modelo ajustado obtenido con la sentencia `summary()` además las estimaciones puntuales y los errores estándar de los parámetros permite estudiar la significación de parámetros mediante el Test de Wald

```
summary(Ajuste.Presion)
```

```
##
## Call:
## glm(formula = cbind(Coronarios1, Coronarios0) ~ Diastolica, family = binomial,
##     data = Presion)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6594  -0.6881  -0.3080   0.4915   1.9642
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.05492    1.75777  -3.445 0.000572 ***
## Diastolica   0.04980    0.02049   2.430 0.015085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.330  on 25  degrees of freedom
## Residual deviance: 24.411  on 24  degrees of freedom
## AIC: 50.841
##
## Number of Fisher Scoring iterations: 4
```

En esta salida se pueden observar en la zona `Coefficients` los siguientes elementos:

- Valor experimental del test de significación del parámetro independiente ($H_0 : \beta_0 = 0$ vs $H_1 : \beta_0 \neq 0$) dado por

$$\frac{\widehat{\beta}_0}{\widehat{S.E}(\widehat{\beta}_0)}$$

  y cuyo valor es -3.445

- p-valor del test de significación del parámetro independiente ($H_0 : \beta_0 = 0$ vs $H_1 : \beta_0 \neq 0$) dado por $P\{|Z| > 3.445\} = 5.7178735 \times 10^{-4}$. El resultado del test muestra que el parámetro independiente es significativo y por tanto no nulo.

- Valor experimental del test de significación del parámetro asociado a la variable explicativa ($H_0 : \beta_1 = 0$ vs $H_1 : \beta_1 \neq 0$) dado por

$$\frac{\widehat{\beta}_1}{\widehat{S.E}(\widehat{\beta}_1)}$$

  y cuyo valor es 2.43

- p-valor del test de significación del parámetro asociado a la variable explicativa ($H_0 : \beta_1 = 0$ vs $H_1 : \beta_1 \neq 0$) dado por $P\{|Z| > 2.43\} = 0.0150852$. El resultado del test muestra que el parámetro asociado a la variable explicativa es significativo y por tanto no nulo.

Intervalos de confianza Una alternativa para analizar la significación de parámetros es a través de sus intervalos de confianza. La función `glm()` permite obtener los intervalos de confianza para los parámetros del modelo ajustado con ella. La función es `confint.default()` y se ejecuta para un objeto tipo *glm*, devolviendo por defecto el intervalo de confianza al 95% para los parámetros, aunque este nivel de confianza se puede modificar:

```
confint.default(Ajuste.Presion,level=0.95)

##                     2.5 %      97.5 %
## (Intercept) -9.500087629 -2.60974903
## Diastolica   0.009638615  0.08996532

confint.default(Ajuste.Presion,level=0.90)

##                      5 %        95 %
## (Intercept) -8.94619555 -3.16364111
## Diastolica   0.01609582  0.08350812
```

Recuérdese que la exponencial de los paráMeros del modelo permiten la interpretación en términos de ventajas (parámetro independiente β_0) y de cociente de ventajas (parámetro asociado a la variable explicativa β_1). La función genérica de `R exp()` permite por tanto obtener intervalos de confianza para dicha ventaja y dicho cociente de ventajas:

```
exp(confint.default(Ajuste.Presion,level=0.95))

##                     2.5 %   97.5 %
## (Intercept) 7.484527e-05 0.073553
## Diastolica  1.009685e+00 1.094136

exp(confint.default(Ajuste.Presion,level=0.90))

##                       5 %       95 %
## (Intercept) 0.0001302317 0.04227155
## Diastolica  1.0162260548 1.08709404
```

Recuérdese que si el intervalo de confianza a nivel $1-\alpha$ para la exponencial del parámetro β_1 contiene el valor 1, se podía concluir la no significación de parámetro β_1 y por tanto la independencia de la respuesta de la variable explicativa.

Test condicional de razón de verosimilitudes: comparación de modelos. La comparación de modelos es una alternativa al test de Wald para estudiar la significación de parámetros. Se basa en los test condicionales de razón de verosimilitud, esto es:

$$H_0 \ : \ p(x_q) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$$
$$H_1 \ : \ p(x_q) = \frac{e^{\beta_0+\beta_1 x_q}}{1+e^{\beta_0+\beta_1 x_q}}$$

El test trata de decidir si es mejor el modelo general (H_1) o el particular (H_0). Para realizar el contraste con `R` basta con ajustar los dos modelos y compararlos con la función genérica `anova()`

```
#MODELO GENERAL
Ajuste.Presion<-glm(cbind(Coronarios1,Coronarios0)~Diastolica,
data=Presion,family=binomial)
#MODELO PARTICULAR
Ajuste.Presion.0<-glm(cbind(Coronarios1,Coronarios0)~1,
data=Presion,family=binomial)
#COMPARACION
anova(Ajuste.Presion.0,Ajuste.Presion)

## Analysis of Deviance Table
##
## Model 1: cbind(Coronarios1, Coronarios0) ~ 1
## Model 2: cbind(Coronarios1, Coronarios0) ~ Diastolica
##   Resid. Df Resid. Dev Df Deviance
## 1        25     30.330
## 2        24     24.412  1   5.9188
```

El valor experimental del test aparece en la tabla con el nombre `Deviance` que utilizando una distribución Chi-cuadrado con un grado de libertad proporciona un p-valor 0.0149803 y que en este caso nos lleva a rehazar H_0 y por tanto a preferir el modelo con los dos parámetros: β_0 y β_1 (y en definitiva el que tiene la variable `Diastolica`) al que sólo tiene el parámetro β_0.

1.2.6 Validación

La validación de un modelo de regresión consiste en la identificación de observaciones para las que el modelo se ajusta pobremente. El principal

elemento para la validación de un modelo de regresión es el conjunto de residuos. Normalmente aquellas observaciones con residuos anormalmente grandes suelen ser observaciones pobremente ajustadas.

Existen fundamentalmente dos tipos de residuos en Regresión logística: residuos *de Pearson* y residuos *deviance* tanto para sus versiones normalizadas como no normalizadas.

La función `residuals()` aplicada a un objeto tipo *glm* permite obtener los distintos residuos:

- Los residuos de *pearson*

$$r_q = \frac{y_q - n_q\widehat{p}_q}{(n_q\widehat{p}_q(1-\widehat{p}_q))^{1/2}}$$

se obtienen con la opción `type="pearson"` .

```
residuals(Ajuste.Presion,type= "pearson")
```

```
##          55          60          65          66          68          70
## -0.19053773 -0.43160133 -0.34565240 -0.25057569 -0.52674165  1.55557694
##          74          75          76          78          80          82
##  1.53480783 -1.08607247 -0.32142714  0.08575285 -0.62225453  0.16631263
##          84          85          86          88          90          92
## -0.39228113  0.39256892  2.42534687 -1.22574422 -0.15582725 -0.67706022
##          94          95          96         100         104         105
##  0.56631060  1.93840545 -0.52889534 -0.02069047 -0.91285041  1.51112720
##         110         112
##  0.58472694 -0.78777011
```

- Los residuos *deviance*

$$d_q = \text{signo}(y_q - \widehat{m}_q)\left(2\left[y_q \ln\left(\frac{y_q}{\widehat{m}_q}\right) + (n_q - y_q)\ln\left(\frac{n_q - y_q}{n_q - \widehat{m}_q}\right)\right]\right)^{\frac{1}{2}}$$

se obtienen con la opción `type="deviance"`

```
residuals(Ajuste.Presion,type= "deviance")
```

```
##          55          60          65          66          68          70
## -0.26706232 -0.60344399 -0.48175310 -0.34898657 -0.73247120  1.37303384
##          74          75          76          78          80          82
```

```
##  1.20684517 -1.50007995 -0.44344030  0.08471917 -0.64551552  0.162566
##          84          85          86          88          90
## -0.53504031  0.37240617  1.96415579 -1.65943605 -0.15701584 -0.908554
##          94          95          96         100         104         1
##  0.53125756  1.76618493 -0.70235224 -0.02073106 -1.18032787  1.542019
##         110         112
##  0.57277908 -0.98263440
```

El resultado en ambos casos es un vector, de igual dimensión que número de filas contiene el *Data.Frame* del ajuste con los residuos.

Los Leverages (denotados por h_{qq} en apuntes) se obtienen mediante la función `hatvalues()`

```
hatvalues(Ajuste.Presion)
```

```
##          55          60          65          66          68          70
## 0.014444574 0.052924276 0.022989647 0.011095412 0.041034197 0.253472966
##          74          75          76          78          80          82
## 0.022958790 0.086946449 0.006859051 0.055560274 0.327700152 0.037458452
##          84          85          86          88          90          92
## 0.005347247 0.027399459 0.005716009 0.052337425 0.292889744 0.019862301
##          94          95          96         100         104         105
## 0.038048843 0.014364426 0.016264775 0.105049251 0.080987260 0.044754490
##         110         112
## 0.281156554 0.082377977
```

De nuevo el resultado es un vector, de igual dimensión que número de filas contiene el Data.Frame utilizado en el ajuste, con los leverages.

La función `rstandard()` aplicada a un objeto tipo *glm* permite obtener los distintos residuos estandarizados:

- Los residuos de *pearson*

$$r_q^s = \frac{r_q}{(1-h_{qq})^{1/2}}$$

se obtienen con la opción `type="pearson"`

```
rstandard(Ajuste.Presion,type= "pearson")
```

```
##          55          60          65          66          68          70
## -0.19192894 -0.44349673 -0.34969546 -0.25197749 -0.53789325  1.80039885
##          74          75          76          78          80          82
##  1.55273579 -1.13660792 -0.32253518  0.08823917 -0.75890351  0.16951787
##          84          85          86          88          90          92
## -0.39333417  0.39806011  2.43230838 -1.25913701 -0.18531041 -0.68388606
##          94          95          96         100         104         105
##  0.57740186  1.95247929 -0.53324971 -0.02187111 -0.95222340  1.54612115
##         110         112
##  0.68966144 -0.82237061
```

- Los residuos *deviance*

$$d_q^s = \frac{d_q}{(1-h_{qq})^{1/2}}$$

se obtienen con la opción `type="deviance"`

```
rstandard(Ajuste.Presion,type= "deviance")
```

```
##          55          60          65          66          68          70
## -0.26901227 -0.62007556 -0.48738811 -0.35093890 -0.74797829  1.58912651
##          74          75          76          78          80          82
##  1.22094223 -1.56987937 -0.44496896  0.08717553 -0.78727268  0.16569992
##          84          85          86          88          90          92
## -0.53647657  0.37761533  1.96979354 -1.70464385 -0.18672389 -0.91771421
##          94          95          96         100         104         105
##  0.54166230  1.77900836 -0.70813468 -0.02191402 -1.23123768  1.57772851
##         110         112
##  0.67556943 -1.02579375
```

El resultado en ambos casos es un vector, de igual dimensión que número de filas contiene el Data.Frame utilizado en el ajuste, con los residuos estandarizados definidos en la teoría.

Recordemos que los residuos estandarizados se pueden obtener dividiendo los residuos entre los leverages. El lector puede hacer las comprobaciones oportunas.

Para validar un modelo se analizan los residuos. Un residuo será significativo (distinto de cero) al nivel de significación $\alpha = 0.05$, cuando el valor absoluto del residuo ajustado sea mayor o igual que $z_{\alpha/2} = 1.96$.

Medidas de influencia. Las medidas de influencia pretenden determinar si algunas observaciones tienen una influencia anómala en las estimaciones de los parámetros. Existen funciones genéricas de `R` para obtener estas medidas en modelos lineales, sin embargo sólo algunas de ellas están disponibles para modelos logit. Una de las medidas de influencia adaptada para modelos logit es la distancias de Cook, denotadas por r_q^{s*} en apuntes y que es posible obtenerla con la función `cooks.distance()`

```
cooks.distance(Ajuste.Presion)
```

```
##           55           60           65           66           68
## 2.699446e-04 5.495675e-03 1.438743e-03 3.561907e-04 6.190205e-03
##           70           74           75           76           78
## 5.502925e-01 2.832704e-02 6.151017e-02 3.592339e-04 2.290251e-04
##           80           82           84           85           86
## 1.403643e-01 5.591539e-04 4.158648e-04 2.231900e-03 1.700551e-02
##           88           90           92           94           95
## 4.377988e-02 7.111917e-03 4.738926e-03 6.593482e-03 2.777888e-02
##           96          100          104          105          110
## 2.350721e-03 2.807407e-05 3.995240e-02 5.599879e-02 9.301560e-02
##          112
## 3.035655e-02
```

Devuelve un vector de dimensión igual al número de filas del data.frame del ajuste con las distancias de cook o medidas globales de la influencia que tiene cada observación en la estimación global de los parámetros.

1.2.7 Precisión del modelo logístico y clasificación de observaciones

Una vez estimado, el modelo logit se puede usar como test de diagnóstico para clasificar las categorías de la variable respuesta en función de los valores de la variable explicativa. Como se indica en la teoría del tema son varias las medidas que muestran la utilidad del modelo logístico ajustado para clasificar:

- Tabla de clasificación.
- Tasa de clasificaciones correctas (CCR de sus siglas en inglés).
- Tasa de verdaderos positivos o sensibilidad muestral.

- Tasa de verdaderos negativos o especificidad muestral.
- Curva ROC y su área.

`R` no posee funciones estándar para realizar estos cálculos, sin embargo su obtención es sencilla mediante operaciones con vectores. Además, para la obtención de la curva ROC y su área existen algunos paquetes de `R` que permiten su obtención.

Por la definición de estas medidas es más útil calcularlas para el caso de datos no agrupados que se abordará en la sección siguiente, por lo que se dejará la explicación detallada a ese momento.

Finalmente para el modelo logístico no existen el coeficiente de determinación sin embargo se han definido medidas de tipo R^2 a partir de las funciones de verosimilitud. Existe una función de `R` de nominada (deviance()) que para un objeto de tipo *glm* devuelve $-2L$, siendo L el máximo de la log-verosimilitud. Por tanto se puede utilizar esta función para calcular la aproximación de R^2 de Cox y Snell, así como la de Nagelkerke. Para ello bastaría con tomar las log-verosimilitudes del modelo nulo (sólo con el parámetro β_0) y del modelo ajustado y realizar los cálculos oportunos.

En nuestro caso se tendría que R^2_{CN} =0.0291603 y R^2_N =0.0291603

1.3 Ajuste de un modelo logit simple con datos no agrupados

A pesar de haber comenzado con la explicación del modelo de regresión logística para datos agrupados, lo más habitual es que los datos no estén en formato agrupado, sobre todo cuando la variable explicativa es cuantitativa continua, pues no es fácil disponer de unos pocos valores diferentes observados en muchas ocasiones cada uno.

En esta sección se mostrará el ajuste de regresión logística del mismo conjunto de datos anterior (personas con problemas coronarios en términos de la presión sanguínea) pero a partir de las observaciones no agrupadas, esto es, en el formato que se supone tenían cuando fueron observadas.

1.3.1 Formato de datos

Se dispone de un conjunto 200 personas a las que se registró la presión arterial y si habían padecido problemas coronarios (1=Si, 0=No). La siguiente tabla muestra las primeras y últimas de estas observaciones

Presión	Problemas coronarios
80	0
70	0
80	0
80	0
110	1
...	...
70	0
78	0
70	0
92	0
68	0

La información de este conjunto de datos está en el formato que se denominará *datos observados* (no agrupados), esto es, para cada observación de la variable explicativa (presión arterial) se tiene una observación de la variable respuesta binaria (1 si sufre problemas coronarios y 0 en caso contrario).

El formato de los datos en R. Para un ajuste de regresión logística simple con datos no agrupados, los datos deben estar registrados en un *Data.Frame* con dos columnas, cada columna con su nombre (preferiblemente que sea descriptivo del contenido de la columna).

Asumimos en este ejemplo que diponemos de un *Data.Frame* denominado `Presion.Orig` con las dos columnas de las variables cuyos nombres son `Diastolica`, y `Coronarios`, y que tienen como primeros y últimos valores los que se muestran a continuación:

```
##   Diastolica Coronarios
## 1         80          0
## 2         70          0
## 3         80          0
## 4         80          0
## 5        110          1
## 6         88          0
##     Diastolica Coronarios
## 195         85          0
## 196         70          0
## 197         78          0
## 198         70          0
## 199         92          0
## 200         68          0
```

Este *Data.Frame* puede haber sido generado a partir de la lectura en `R` de un fichero de tipo *.csv* con las dos columnas indicadas anteriormente.

Antes de abordar el ajuste del modelo logit a este conjunto de datos, veamos brevemente algunas identificaciones de los aspectos teóricos del tema con los datos disponibles.

La formulación del modelo logístico binario con datos no agrupados es del siguiente modo:

$$y_i \rightsquigarrow B(p(x_i)),\ p(x_i) = \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}.\ i = 1, \ldots, n$$

donde

- n es el número de observaciones de la variable explicativa X y de la respuesta Y
- $p(x_i)$ es la probabilidad de que la respuesta tome el valor $Y = 1$ para una observación de la variable explicativa $X = x_i$
- β_0 y β_1 son los parámetros a estimar.
- y_i representa la observación de la respuesta ($y_i = 1$) si se observó un éxito o ($y_i = 0$) si se observó un fracaso.
- La distribución de la respuesta es una Bernouilli de parámetro $p(x_i)$

1.3.2 Ajuste del modelo

Para ajustar un modelo de regresión logística con `R` con datos no agrupados en la fórmula de la función `glm()` no es necesario utilizar la expresión `cbind()` sino que hemos de poner el nombre de las variables. Para nuestro *Data.Frame* del ajuste (denominado `Presion.Orig`) se tendría:

```
Ajuste.Presion.Orig<-glm(Coronarios~Diastolica,data=Presion.Orig,
family=binomial)
```

De este modo se ha creado el objeto `Ajuste.Presion.Orig` que es de tipo *glm* y que tiene todos los elementos necesarios para hacer el estudio de nuestros datos mediante regresión logística.

El resumen del modelo ajustado se obtiene con la sentencia `summary()` aplicada al objeto de tipo *glm* que hemos denominado `Ajuste.Presion.Orig`:

```
summary(Ajuste.Presion.Orig)

##
## Call:
## glm(formula = Coronarios ~ Diastolica, family = binomial, data = Presion.O
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9826  -0.5867  -0.4873  -0.3843   2.2990
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.05492    1.75780  -3.445 0.000572 ***
## Diastolica   0.04980    0.02049   2.430 0.015087 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.55  on 199  degrees of freedom
## Residual deviance: 148.64  on 198  degrees of freedom
## AIC: 152.64
##
## Number of Fisher Scoring iterations: 5
```

Como se puede apreciar los resultado de este ajuste son los mismos que se vieron para el caso de datos agrupados, puesto que ambos conjuntos de datos son el mismo, sólo que en distinto formato. La única diferencia que se aprecia está en las cantidades `Null deviance`, `Residual deviance` y en las estadísticas de residuos. La razón está en que para el caso de datos agrupados, se utilizan 26 distibuciones binomiales para obtener la log-verosimilitud, mientras que ahora se utilizan 200 distribuciones de Bernouilli.

No entraremos por tanto a interpretar los parámetros en este punto pues coincide esta interpretación con las del ejemplo con datos agrupados.

1.3.3 Predicción e interpretación de parámetros

Al igual que en el caso de datos agrupados, la estimación puntual de los parámetros del modelo permite obtener una estimación puntual de las probabilidades de éxito (probabilidad de padecer Coronarios) para cada una de las observaciones,

sin más que sustituir en la ecuación del modelo:

$$\widehat{p}(x_i) = \frac{e^{\widehat{\beta}_0+\widehat{\beta}_1 x_i}}{1+e^{\widehat{\beta}_0+\widehat{\beta}_1 x_i}}$$

`R` calcula estas probabilidades mediante la sentencia `fitted.values(Ajuste.Presion.Orig)` (recordemos que `Ajuste.Presion.Orig` es el nombre que se dio al objeto tipo *glm* para realizar el ajuste).

```
fitted.values(Ajuste.Presion.Orig)
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##         1         2         3         4         5         6
## 0.11197136 0.07117488 0.11197136 0.11197136 0.35969184 0.15811176
##         7         8         9        10
## 0.20204958 0.08552255 0.11197136 0.11197136
##       191       192       193       194       195       196
## 0.11197136 0.21019816 0.25450621 0.11197136 0.13922389 0.07117488
##       197       198       199       200
## 0.10244348 0.07117488 0.18646623 0.06486489
```

El resultado de esta sentencia es un vector con las probabilidades estimadas por el modelo (sólo se muestran los 10 primeros y 10 últimos). En la siguiente tabla se muestran las primeras y últimas probabilidades junto con los valores observados de las variables y las ventajas tanto de respuesta 1 ($\frac{\widehat{p}(x_i)}{1-\widehat{p}(x_i)}$) como de respuesta 0 ($\frac{1-\widehat{p}(x_i)}{\widehat{p}(x_i)}$):

Observación	$X = x_i$	y_i	$\widehat{p}(x_i)$	$1-\widehat{p}(x_i)$	$\frac{\widehat{p}(x_i)}{1-\widehat{p}(x_i)}$	$\frac{1-\widehat{p}(x_i)}{\widehat{p}(x_i)}$
1	**80**	**0**	**0.112**	**0.888**	**0.1261**	**7.9309**
2	70	0	0.0712	0.9288	0.0766	13.0499
3	80	0	0.112	0.888	0.1261	7.9309
4	80	0	0.112	0.888	0.1261	7.9309
5	**110**	1	**0.3597**	**0.6403**	**0.5617**	1.7802
...	...	...	...	...	...	...
196	70	0	0.0712	0.9288	0.0766	13.0499
197	78	0	0.1024	0.8976	0.1141	8.7615
198	70	0	0.0712	0.9288	0.0766	13.0499
199	92	0	0.1865	0.8135	0.2292	4.3629
200	**68**	0	**0.0649**	**0.9351**	**0.0694**	14.4167

De estas probabilidades podemos sacar algunas interpretaciones sobre la persona que sustenta una familia:

- Para la primera persona de la muestra, que tenía presión arterial 80 y una situación de problemas coronarios y_1 =0, el modelo le asigna una probabilidad de padecer problemas coronarios de 0.112.
- Para la última persona de la muestra, que tenía presión arterial 68 y una situación de problemas coronarios y_{200} =0, el modelo le asigna una probabilidad de padecer problemas coronarios de 0.0649.
- Para la quinta persona de la muestra, que tenía presión arterial 110 y una situación de problemas coronarios y_5 =0, modelo le asigna:
  - una probabilidad de padecer problemas coronarios de 0.3597
  - una probabilidad de no padecer problemas coronarios de 0.6403.
  - una ventaja de padecer problemas coronarios frente a no padecerlos de 0.5617, y por tanto resulta 0.5617 veces más (menos) probable padecer problemas coronarios que no padecerlos.
  - una ventaja de no padecer problemas coronarios frente a padecerlos de 1.7802, y por tanto resulta 1.7802 veces más probable no padecer problemas coronarios que padecerlos.
- Obśervese qué ocurre con las ventajas al aumentar en 10 unidades la presión arterial (de 70 a 80, por ejemplo) y compárese con la exponencial del parámetro $10 \times \beta_1$.
- Obśervese qué ocurre con las ventajas al aumentar en dos unidades la presión arterial (de 68 a 70 por ejemplo) y compárese con la exponencial del parámetro $2 \times \beta_1$.

Todo lo explicado anteriormente sobre las funciones `fitted.values()` y `predict()` con sus distintas opciones : `type=`, `se.fit=TRUE` o `newdata`, es válido también para el caso no agrupado.

1.3.4 Bondad del ajuste

Cuando los datos del ajuste no están agrupados, la única alternativa es utilizar el Test de Hosmer y Lemeshow, puesto que para cada observación de la variable explicativa sólo hay una observación de la respuesta, con lo que no podemos esperar que se verifiquen las condiciones de los Test Chi-cuadrado y la salida `Residual.deviance` de la función `summary()` en este caso no muestra el valor experimental del test de razón de verosimilitudes.

En el caso de datos no agrupados la función de cálculo del test de Hosmer y Lemeshow está diseñada para que los datos estén dispuestos en un *Data.Frame*

en el que la última columna tenga las observaciones de la respuesta (ceros y unos) para cada observación de la/s variable/s explicativa/s.

Para usar la función, los parámetros necesarios son:

- *Data.frame* del ajuste.
- Vector con las probabilidades predichas por el modelo (obtenido a partir de la función `fitted.values()`)
- Número de grupos a considerar (g=10 es el valor por defecto).
- Valor lógico indicando si se trata de datos agrupados ((TRUE)) o no agrupados (`FALSE`), éste último es el valor por defecto.

En nuestro ejemplo el test de bondad de ajuste de Hosmer y Lemeshow para 10 grupos ofrece el siguiente resultado:

```
hosmerlem.test(Presion.Orig,fitted.values(Ajuste.Presion.Orig),g=10,group=F)
```

```
##           Hosmer-Lemeshow Test
## X-squared            4.4532205
## p.value              0.8140948
```

Este resultado conduce a no rechazar la hipótesis nula del test de bondad del ajuste del modelo de regresión logística, y por tanto a aceptar que el modelo logístico se ajusta bien a los datos. Además se puede observar que el resultado es el mismo que en el caso de datos agrupados por tratarse de la misma información.

1.3.5 Significación de parámetros

El estudio de la significación de parámetros no cambia nada con respecto al caso de datos agrupados, por lo que no se dará explicación alguna.

1.3.6 Validación

La validación del modelo en el caso de datos no agrupados se estudia del mismo modo que para los agrupados, a través de los residuos y los leverages, que se obtienen con las mismas funciones de `R` que se explicaron para la situación de datos agrupados. A continuación se muestran los resultados (sólo se muestran los 10 primeros y 10 últimos):

- Residuos de *pearson*

$$r_i = \frac{y_i - \widehat{p}_i}{(\widehat{p}_i(1-\widehat{p}_i))^{1/2}}$$

```
residuals(Ajuste.Presion.Orig,type= "pearson")
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##          1          2          3          4          5          6
## -0.3550913 -0.2768193 -0.3550913 -0.3550913  1.3342255 -0.4333660
##          7          8          9         10
## -0.5032004 -0.3058115 -0.3550913 -0.3550913
##        191        192        193        194        195        196
## -0.3550913  1.9384054 -0.5842883 -0.3550913 -0.4021720 -0.2768193
##        197        198        199        200
## -0.3378401 -0.2768193 -0.4787539 -0.2633708
```

- Residuos *deviance*

$$d_i = \text{signo}(y_i - \widehat{p}_i)\left(2\left[y_i \ln\left(\frac{y_i}{\widehat{p}_i}\right) + (1-y_i)\ln\left(\frac{1-y_i}{1-\widehat{p}_i}\right)\right]\right)^{\frac{1}{2}}$$

se obtienen con la opción `type="deviance"`

```
residuals(Ajuste.Presion.Orig,type= "deviance")
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##          1          2          3          4          5          6
## -0.4873423 -0.3842780 -0.4873423 -0.4873423  1.4300403 -0.5866992
##          7          8          9         10
## -0.6718762 -0.4228533 -0.4873423 -0.4873423
##        191        192        193        194        195        196
## -0.4873423  1.7661849 -0.7664313 -0.4873423 -0.5475780 -0.3842780
##        197        198        199        200
## -0.4649284 -0.3842780 -0.6424451 -0.3662356
```

- Leverages

```
hatvalues(Ajuste.Presion.Orig)
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##           1           2           3           4           5           6
## 0.005650050 0.009387658 0.005650050 0.005650050 0.070290784 0.006542169
##           7           8           9          10
## 0.012682942 0.007652870 0.005650050 0.005650050
##         191         192         193         194         195         196
## 0.005650050 0.014364443 0.026262601 0.005650050 0.005479917 0.009387658
##         197         198         199         200
## 0.006173394 0.009387658 0.009931123 0.010258210
```

- Residuos de *pearson* estandarizados

$$r_i^s = \frac{r_i}{(1-h_{ii})^{1/2}}$$

```
rstandard(Ajuste.Presion.Orig,type= "pearson")
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##          1          2          3          4          5          6
## -0.3560987 -0.2781279 -0.3560987 -0.3560987  1.3837437 -0.4347906
##          7          8          9         10
## -0.5064222 -0.3069884 -0.3560987 -0.3560987
##        191        192        193        194        195        196
## -0.3560987  1.9524793 -0.5921153 -0.3560987 -0.4032784 -0.2781279
##        197        198        199        200
## -0.3388878 -0.2781279 -0.4811490 -0.2647322
```

- Residuos *deviance* estandarizados

$$d_i^s = \frac{d_i}{(1-h_{ii})^{1/2}}$$

```
rstandard(Ajuste.Presion.Orig,type= "deviance")
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##          1          2          3          4          5          6
## -0.4887250 -0.3860946 -0.4887250 -0.4887250  1.4831146 -0.5886279
##          7          8          9         10
## -0.6761778 -0.4244807 -0.4887250 -0.4887250
##        191        192        193        194        195        196
## -0.4887250  1.7790084 -0.7766982 -0.4887250 -0.5490846 -0.3860946
##        197        198        199        200
## -0.4663701 -0.3860946 -0.6456591 -0.3681286
```

- Distancias de Cooks

```
cooks.distance(Ajuste.Presion.Orig)
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##            1            2            3            4            5
## 0.0003602664 0.0003665326 0.0003602664 0.0003602664 0.0723823313
##            6            7            8            9           10
## 0.0006224473 0.0016472472 0.0003633915 0.0003602664 0.0003602664
##          191          192          193          194          195
## 0.0003602664 0.0277789176 0.0047280101 0.0003602664 0.0004480644
##          196          197          198          199          200
## 0.0003665326 0.0003566936 0.0003665326 0.0011610800 0.0003631893
```

1.3.7 Precisión del modelo logístico y clasificación de observaciones

Como se indicó previamente las medidas de precisión y clasificación del modelo logístico son generalmente usadas en el caso de datos no agrupados, por lo que detallaremos en este apartado su obtención.

Tabla de clasificación

El modelo logit se puede usar como test de diagnóstico para clasificar las categorías de la variable respuesta en función de los valores de la variable explicativa mediante la tabla de clasificación, una tabla que en las filas contiene el estado real en la muestra y en las columnas la categoría predicha por el modelo.

	Clasificación	
Observación	Fracaso	éxito
Fracaso	Verdaderos Negativos (VN)	Falsos Positivos (FP)
éxito	Falsos Negativos (FN)	Verdaderos Positivos (VP)

La tabla permite calcular la tasa de clasificaciones correctas (CCR de sus siglas en inglés) como los individuos correctamente clasificados (VP+VN) entre el número total de individuos. Así mismo se puede obtener la tasa de verdaderos positivos o sensibilidad muestral (VP/(VP+FN)), y la tasa de verdaderos negativos o especificidad muestral (VN/(FP+VN)).

Para determinar cómo clasifica el modelo a cada observación, se elige un punto de corte, si la probabilidad predicha por el modelo para una observación es mayor que el punto de corte, se clasifica como éxito y si es menor como fracaso. Los valores VP, FN, FP y VN variarán según el punto de corte elegido. La tabla de clasificación y la tasa de clasificaciones correctas se puede obtener mediante el comando `table()` con el vector de observaciones de la respuesta y el vector de categorías predichas (para el punto de corte elegido) como parámetros de la función.

La función `ifelse(`*Condicion*,*valor.si*,*valor.no*`)` puede utilizarse para obtener el vector de categorías predichas:

```
Categoria.Pred<-ifelse(fitted.values(Ajuste.Presion.Orig)>=0.5,1,0)
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##   1   2   3   4   5   6   7   8   9  10
##   0   0   0   0   0   0   0   0   0   0
## 191 192 193 194 195 196 197 198 199 200
##   0   0   0   0   0   0   0   0   0   0
```

Y la tabla de clasificación:

```
table(Presion.Orig$Coronarios,Categoria.Pred)

##    Categoria.Pred
##       0
##   0 174
##   1  26
```

Obsérvese que en este caso, utilizando 0.5 como punto de corte, el modelo predice todas las observaciones como ceros (sin problemas coronarios)

En nuestro ejemplo la tabla de clasificación ha sido

	Clasificación	
Observación	Fracaso	éxito
Fracaso	VN=174	FP=0
éxito	FN=26	VP=0

Y las medidas:

- Tasa de clasificaciones correctas: CCR =87%
- Sensibilidad: 0%
- Especificidad: 100%

Cualquier otro punto de corte produciría otros modos de clasificación. Por ejemplo, si tomamos 0.13 que es la proporción de éxitos en la muestra, el estudio de la clasificación se modifica considerablemente:

```
Categoria.Pred.013<-ifelse(fitted.values(Ajuste.Presion.Orig)>=0.13,1,0)
```

(sólo se muestran los 10 primeros y 10 últimos)

```
##  1  2  3  4  5  6  7  8  9 10
##  0  0  0  0  1  1  1  0  0  0
## 191 192 193 194 195 196 197 198 199 200
##   0   1   1   0   1   0   0   0   1   0
```

Y la tabla de clasificación:

```
table(Presion.Orig$Coronarios,Categoria.Pred.013)

##    Categoria.Pred.013
##       0   1
##   0 117  57
##   1  12  14
```

En nuestro ejemplo la tabla de clasificación ha sido

	Clasificación	
Observación	Fracaso	éxito
Fracaso	VN=117	FP=57
éxito	FN=12	VP=14

Y las medidas:

- Tasa de clasificaciones correctas: CCR =65.5%
- Sensibilidad: 53.85%
- Especificidad: 67.24%

Curva ROC

La librería `pROC` permite analizar varias medidas relacionadas con la tabla de clasificación, y representarlas gráficamente. La función (roc()) de esta librería genera un objeto de tipo *curva roc* que contiene, entre otras cosas;

- Vector de puntos de corte para el cálculo de sesibilidades y especificidades
- Vector de sensibilidades para esos puntos de corte
- Vector de especificidades para esos puntos de corte

Las sensibilidades y especificidades para distintos puntos de corte son las que se utilizan para dibujar la *curva ROC* y el área bajo dicha curva que sirve como medida de precisión de las estimaciones del modelo logístico.

Para generar el objeto de tipo *curva roc* la función `roc()` el vector de la variable respuesta (de ceros y unos) y el vector de predicciones.

```
library(pROC)
CurvaROC<-roc(Presion.Orig$Coronarios,fitted.values(Ajuste.Presion.Orig))
CurvaROC

##
## Call:
## roc.default(response = Presion.Orig$Coronarios, predictor = fitted.values(Ajuste.
##
## Data: fitted.values(Ajuste.Presion.Orig) in 174 controls (Presion.Orig$Coronarios
## Area under the curve: 0.6293
```

Como se puede ver la única salida de la función es el área bajo la curva ROC, pero permite mostrar el resto de elementos:

```
CurvaROC$thresholds
```

```
##  [1]       -Inf 0.03976522 0.05043401 0.05772454 0.06197181 0.06801988
##  [7] 0.07834871 0.08751064 0.09156980 0.09804217 0.10720742 0.11711801
## [13] 0.12781340 0.13629301 0.14226200 0.15170593 0.16496886 0.17914609
## [19] 0.19425790 0.20612387 0.21439175 0.23654578 0.27430726 0.29933044
## [25] 0.33212220 0.37131474        Inf
```

```
CurvaROC$sensitivities
```

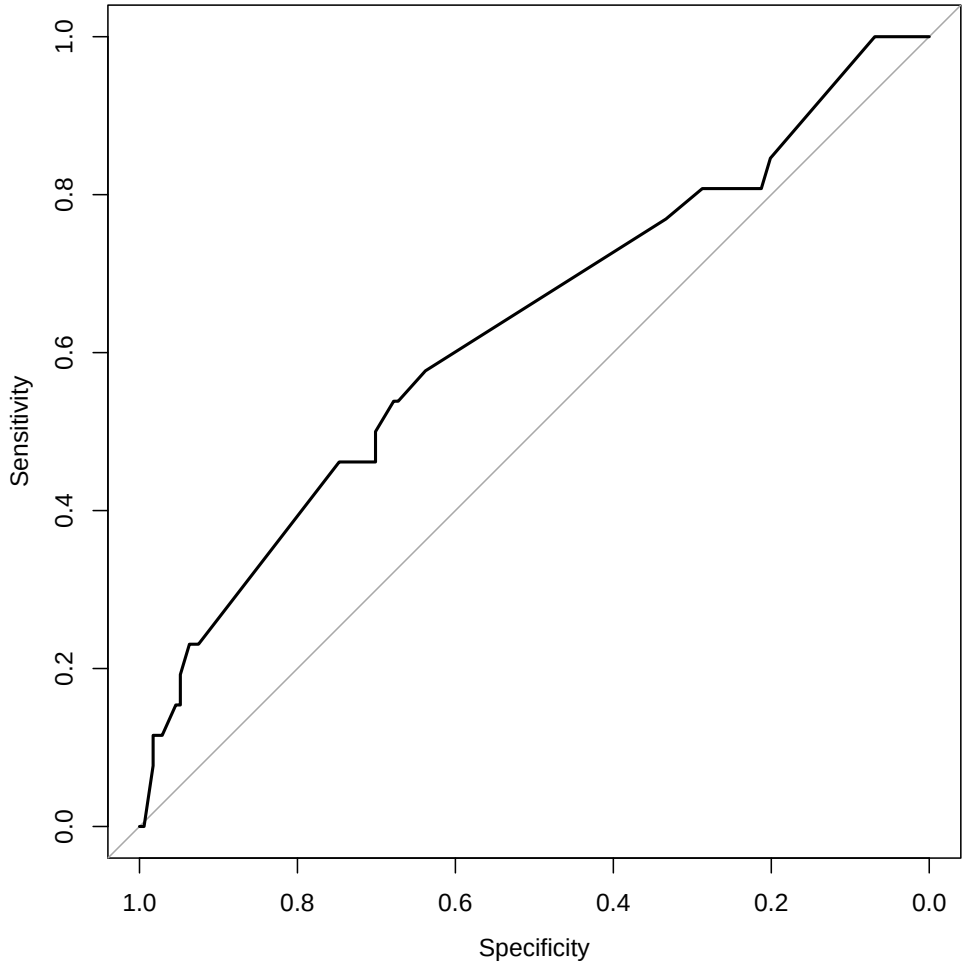
```
##  [1] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
##  [7] 0.84615385 0.80769231 0.80769231 0.80769231 0.76923077 0.57692308
## [13] 0.53846154 0.53846154 0.50000000 0.46153846 0.46153846 0.23076923
## [19] 0.23076923 0.19230769 0.15384615 0.15384615 0.11538462 0.11538462
## [25] 0.07692308 0.00000000 0.00000000
```

```
CurvaROC$specificities
```

```
##  [1] 0.000000000 0.005747126 0.028735632 0.040229885 0.045977011
##  [6] 0.068965517 0.201149425 0.212643678 0.281609195 0.287356322
## [11] 0.333333333 0.637931034 0.672413793 0.678160920 0.701149425
## [16] 0.701149425 0.747126437 0.925287356 0.936781609 0.948275862
## [21] 0.948275862 0.954022989 0.971264368 0.982758621 0.982758621
## [26] 0.994252874 1.000000000
```

Con la función `plot` aplicada a un objeto de tipo *curva roc* se obtiene la gráfica de la curva ROC.

```
plot(CurvaROC)
```



Adaptación para datos agrupados

Como se indicó en la primera parte de este documento, en el caso de datos agrupados la información disponible se encuentra en tres columnas:

- Observaciones únicas de la variable explicativa.
- Número de fracasos para cada valor único de la variable explicativa.
- Número de éxitos para cada valor único de la variable explicativa.

Resulta sencillo mediante sentencias de `R` pasar de un *Data.Frame* que contiene datos en formato agrupado (3 columnas) a un *Data.Frame* que contiene datos en formato no agrupado (2 columnas). Para ello se puede

utilizar la función `rep()` que crea un vector con valores repetidos tantas veces como se indique (ver ayudas de la sentencia).

Así supongamos que tenemos un *Data.Frame* denominado `Datos.Agrupados` que contiene datos en formato agrupado (3 columnas), entonces la siguiente sentencia permite expresar ese mismo conjunto de datos en formato no agrupado (2 columnas):

```
Datos.Orig.Explicativa<-c(rep(Datos.Agrupados[,1],Datos.Agrupados[,2]),
                          rep(Datos.Agrupados[,1],Datos.Agrupados[,3]))
Datos.Orig.Respuesta<-c(rep(seq(0,0,length=nrow(Datos.Agrupados)),
                            Datos.Agrupados[,2]),
                        rep(seq(1,1,length=nrow(Datos.Agrupados)),
                            Datos.Agrupados[,3]))
Datos.Orig<-data.frame(Datos.Orig.Explicativa,Datos.Orig.Respuesta)
names(Datos.Orig)<-c("Explicativa","Respuesta")
```

Si además se tiene el vector de probabilidades predichas obtenido con el ajuste de los datos agrupados denominado `Probabilidades`, se puede transformar en el vector de longitud original de la forma:

```
Probabilidades.Orig<-c(rep(Probabilidades,Datos.Agrupados[,2]),
                       rep(Probabilidades,Datos.Agrupados[,3]))
```

De este modo, con el *Data.Frame* `Datos.Orig` y el vector de probabilidades `Probabilidades.Orig` se pueden utilizar los mecanismos de cálculo de la CCR y la curva ROC. Como ocurre con cualquier procedimiento de `R` ésta no es la única manera de resolver esta cuestión sino que existen multitud de mecanismos. La que aquí se aporta es una de las posibles soluciones, que tiene en cuenta la posición correcta de cada uno de los elementos de los vectores involucrados.

1.4 Ajuste de un modelo logit múltiple.

1.4.1 Introducción

En este tema se aborda el ajuste de un modelo de regresión logística múltiple para un ejemplo clásico de regresión losgística. Dado que la mayoría de las cuestiones del ajuste se han explicado con detalle en el modelo simple, no se entreará en detalles en los aspectos allí tratados y se explicará con más de detalle las cuestiones que tienen que ver con la selección de variables por el método stepwise y los modelos con interacción.

En el ejemplo que se utilizará de ilustración se trata de explicar la probabilidad de padecer una enfermedad coronaria de una muestra de 200 hombres para los que se han observado además las siguientes siete variables explicativas:

1. Edad en años (E)
2. Presión sistólica en milímetros de mercurio (S)
3. Presión diastólica en milímetros de mercurio (D)
4. Colesterol en miligramos por DL (C)
5. Altura en pulgadas (A)
6. Peso en libras (P)
7. Problemas coronarios (1 si se tiene alguna incidencia en los 10 años previos y 0 en otro caso)

Los datos están disponibles en el fichero chapman.dat, que se trata de un fichero de texto plano, configurado en 8 columnas separadas por espacios en blanco y sin los nombres de las variables en dicho fichero. La primera columna contiene un indicador numérico que identifica cada caso.

La lectura de datos se lleva a cabo mediante las siguientes sentencias:

```
Chapman<-read.csv('CHAPMAN.DAT',header=F,sep='')
names(Chapman)<-c("Id","Edad","Sistolica","Diastolica",
"Colesterol","Altura","Peso","Coronarios")
```

Tras leer el conjunto de datos podemos mostrar las primera y últimas filas del *Data.Frame* resultante:

```
head(Chapman)

##   Id Edad Sistolica Diastolica Colesterol Altura Peso Coronarios
## 1  1   44       124         80        254     70  190          0
## 2  2   35       110         70        240     73  216          0
## 3  3   41       114         80        279     68  178          0
## 4  4   31       100         80        284     68  149          0
## 5  5   61       190        110        315     68  182          1
## 6  6   61       130         88        250     70  185          0

tail(Chapman)
```

```
##      Id Edad Sistolica Diastolica Colesterol Altura Peso Coronarios
## 195 195   29       120         85        187     68  181          0
## 196 196   29       110         70        238     72  143          0
## 197 197   49       112         78        283     64  149          0
## 198 198   49       100         70        264     70  166          0
## 199 199   50       128         92        264     70  176          0
## 200 200   31       105         68        193     67  141          0
```

como se podrá observar, los conjuntos de datos mostrados en los ajustes de los modelos logit simples (tanto para datos no agrupados como para datos agrupados) son una parte de este conjunto de datos, en concreto las variables *Diastolica* y *Coronarios.*

1.4.2 Selección variables mediante el método stepwise

La variable de respuesta será denotada por Y en las explicaciones teóricas y toma el valor $Y = 1$ cuando el individuo tiene algún problema coronario y el valor $Y = 0$ en otro caso. La notación para las variables explicativas es la indicada entre paréntesis en el enunciado del ejercicio.

El modelo con todas las variables explicativas es de la forma

$$\begin{aligned} L(e,s,d,c,a,p) &= \ln\left(\frac{p(e,s,d,c,a,p)}{1-p(e,s,d,c,a,p)}\right) \\ &= \beta_0 + \beta_E e + \beta_S s + \beta_D d + \beta_C c + \beta_A a + \beta_P p \end{aligned}$$

donde $p(e,s,d,c,a,p)$ es la probabilidad de padecer un infarto para los valores observados de las variables explicativas representados por la inicial de la variable con minúscula. Observemos que se ha cambiado la notación de los apuntes de teoría para favorecer la intuición en la interpretación de los parámetros cuyos subíndices están representados por las iniciales de las variables en lugar de números de orden como en los apuntes de teoría.

En el *Data.Frame* que contiene los datos, la variable respuesta es *Coronario* y las variables explicativas *Edad, Sistolica, Diastolica, Colesterol, Altura, Peso.* El ajuste con todas las variables se muestra a continuación:

```
Ajuste.all<-glm(Coronarios~Edad+Sistolica+Diastolica+Colesterol+Altura+Peso
,family=binomial,data=Chapman)
summary(Ajuste.all)

##
```

```
## Call:
## glm(formula = Coronarios ~ Edad + Sistolica + Diastolica + Colesterol +
##     Altura + Peso, family = binomial, data = Chapman)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1130  -0.5541  -0.3907  -0.2527   2.6811
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.517321   7.481215  -0.604   0.5460
## Edad         0.045900   0.023535   1.950   0.0511 .
## Sistolica    0.006856   0.020198   0.339   0.7343
## Diastolica  -0.006937   0.038352  -0.181   0.8565
## Colesterol   0.006306   0.003632   1.736   0.0825 .
## Altura      -0.074002   0.106214  -0.697   0.4860
## Peso         0.020142   0.009871   2.041   0.0413 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.55  on 199  degrees of freedom
## Residual deviance: 134.85  on 193  degrees of freedom
## AIC: 148.85
##
## Number of Fisher Scoring iterations: 5
```

La función de `R` para realizar la selección stepwise necesita del ajuste del modelo nulo o modelo que sólo tiene el parámetro independiente que se muestra a continuación:

```
Ajuste.0<-glm(Coronarios~1,family=binomial,data=Chapman)
summary(Ajuste.0)

##
## Call:
## glm(formula = Coronarios ~ 1, family = binomial, data = Chapman)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -0.5278  -0.5278  -0.5278  -0.5278   2.0200
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9010     0.2103  -9.041   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.55  on 199  degrees of freedom
## Residual deviance: 154.55  on 199  degrees of freedom
## AIC: 156.55
##
## Number of Fisher Scoring iterations: 4
```

Para realizar una selección stepwise para regresión logística con R se utiliza la función `step()` cuyos argumentos son más importantes son:

- `object`. Es un objeto de tipo *glm* que se utiliza como modelo inicial del proceso stepwise. Si el proceso se realiza hacia adelante, el modelo inicial es el que no tiene variables (sólo la constante).

- `scope`. Define el modelo inicial y final deseado mediante fórmulas de R.

- `direction`. Indica la dirección de la búsqueda en el proceso stepwise, siendo sus opciones `"forward"` para búsqueda hacia delante (introduciendo variables), `"backward"` para búsqueda hacia atrás (eliminando variables), o `"both"` para ambos sentidos (en cada paso inclusión y eliminación o viceversa). La opción por defecto es `"backward"`.

En nuestro ejemplo, la selección parte del modelo que tiene solo el parámetro independiente y se pretende utilizar el procedimiento stepwise en ambas direcciones para encontrar los mejores predictores. La sentencia para ello y la salida obtenida se muestran a continuación.

```
Ajuste.step<-step(Ajuste.0,scope=list(lower=Coronarios~1,
upper=Coronarios~Edad+Sistolica+Diastolica+Colesterol+Altura+Peso),
direction="both")
```

```
## Start:  AIC=156.55
## Coronarios ~ 1
##
##              Df Deviance    AIC
## + Edad        1   142.74 146.74
## + Colesterol  1   146.94 150.94
## + Sistolica   1   148.47 152.47
## + Diastolica  1   148.64 152.64
## + Peso        1   150.13 154.13
## <none>            154.56 156.56
## + Altura      1   153.84 157.84
##
## Step:  AIC=146.74
## Coronarios ~ Edad
##
##              Df Deviance    AIC
## + Peso        1   138.77 144.77
## + Colesterol  1   139.93 145.93
## <none>            142.74 146.74
## + Diastolica  1   141.42 147.42
## + Sistolica   1   141.87 147.87
## + Altura      1   142.74 148.74
## - Edad        1   154.56 156.56
##
## Step:  AIC=144.77
## Coronarios ~ Edad + Peso
##
##              Df Deviance    AIC
## + Colesterol  1   135.52 143.52
## <none>            138.77 144.77
## + Altura      1   138.05 146.05
## + Sistolica   1   138.43 146.43
## + Diastolica  1   138.50 146.50
## - Peso        1   142.74 146.74
## - Edad        1   150.13 154.13
##
## Step:  AIC=143.52
## Coronarios ~ Edad + Peso + Colesterol
##
##              Df Deviance    AIC
```

```
## <none>            135.52 143.52
## - Colesterol  1   138.77 144.77
## + Altura      1   134.98 144.98
## + Sistolica   1   135.35 145.35
## + Diastolica  1   135.45 145.45
## - Peso        1   139.93 145.93
## - Edad        1   142.30 148.30
```

El resultado es un objeto de tipo *glm* con el modelo ajustado que contiene las variables seleccionadas mediante el método stepwise.

Para explicar el proceso de selección stepwise trocearemos esta salida en los distintos pasos dados por el programa.

Recordemos que en el paso inicial del procedimiento stepwise se realizan sucesivamente los contrastes condicionales de razón de verosimilitud entre el modelo cte y cada uno de los modelos simples que resultan de la introducción de cada una de las variables explicativas. Por ejemplo, para el caso de la variable edad, las hipótesis del test son

$$\begin{aligned} H_0: &\quad L(e,s,d,c,a,p) = \beta_0 \quad (M_P) \\ H_1: &\quad L(e,s,d,c,a,p) = \beta_0 + \beta_E e \quad (M_G) \end{aligned}$$

donde el modelo cte es un caso particular ($M_P = cte$) del modelo más general obtenido al introducir la variable edad ($M_G = (cte, E)$). El estadístico RV condicional para este contraste $G^2(cte/(cte,E))$ tiene distribución chi-cuadrado con un grado de libertad, y su valor observado se obtiene mediante la diferencia de las *deviance* de ambos modelos $G^2_{Obs} = G^2(M_P = cte) - G^2(M_G = (cte, E))$.

De entre todas las variables para las que en el procedimiento anterior el contraste es significativo ($p-valor \leq \alpha_1$) se selecciona la asociada al mínimo p-valor (equivalentemente máximo valor del estadístico RV condicional).

La primera tabla de la salida de R del procedimiento stepwise correspondiente al paso inicial aparece a continuación.

```
--------------------------------------------------------------------------------
Start:  AIC=156.55
Coronarios ~ 1

             Df Deviance    AIC
+ Edad        1   142.74 146.74
+ Colesterol  1   146.94 150.94
+ Sistolica   1   148.47 152.47
```

```
+ Diastolica  1   148.64 152.64
+ Peso        1   150.13 154.13
<none>            154.56 156.56
+ Altura      1   153.84 157.84
-------------------------------------------------------------------------------
```

En esta salida se muestra inicialmente el *score* del criterio de información de Akaike (AIC) del modelo cte. A continuación se muestra una tabla donde aparecen cada una de las variables que es posible seleccionar para su inclusión en el modelo en este paso. Junto a cada variable aparecen los siguientes elementos:

- Grados de libertad (`Df`) de la distribución chi-cuadrado del estadístico RV condicional para el contraste que compara el modelo cte con el que resulta de añadir a éste la variable. En el procedimiento stepwise estos grados de libertad son siempre 1.
- La *deviance* del modelo que resulta de añadir al modelo cte, ésta variable. Recordemos que R denomina *deviance* de un modelo, al doble (con signo negativo) de la log-verosimilitud de dicho modelo.
- El AIC del modelo que resulta de añadir al modelo cte, ésta variable. El AIC para un modelo ajustado en R es el doble (con signo negativo) de la log-verosimilitud de dicho modelo más el doble del número de parámetros del modelo.

Además de las variables a seleccionar, aparece en esta tabla una fila denominada `<none>` que tiene la *deviance* y el AIC del modelo actual (en este paso del modelo cte).

Como se ha indicado previamente, las variables candidatas a entrar en el modelo son todas aquellas para las que el contraste es significativo ($p-valor \leq \alpha_1$). En la tabla, las variables para las que el constraste es significativo aparecen por encima de la fila `<none>` mientras que aquellas para las que el contraste es no significativo aparecen por debajo. Además recordemos que se selecciona la variable asociada al mínimo p-valor (equivalentemente máximo valor del estadístico RV condicional). Como el valor observado del estadístico de contraste es la diferencia entre la *deviance* del modelo cte y la del modelo que resulta de incluir cada variable, la variable a seleccionar será aquella con menor *deviance* (puesto que la del modelo cte es la misma para todos los contrastes de cada variable). Como se puede observar en la tabla de la salida de R, las variables cuyo contraste es significativo (están por encima de `<none>`) están ordenadas de menor a mayor valor de la *deviance*. Así, la primera variable que entra en el modelo es la de la primera fila de la tabla.

En este primer paso, la variable con menor *deviance* es la variable Edad (142.74) que será por tanto la de mayor valor observado del estadístico RV condicional para este contraste $G^2_{Obs}(cte/(cte, E)) = 154.56 - 142.74 = 11.82$, con p-valor 0.006. Este p-valor no lo muestra la salida de R, pero se puede obtener mediante `1-pchisq(11.82,1)`. Como este p-valor es menor que el nivel de significación usualmente fijado para la entrada de términos ($\alpha_1 = 0.1$), el test es significativo y la variable edad es la seleccionada para entrar en el modelo. Este hecho ya se sabía porque el programa había colocado a la variable Edad por encima de la fila `<none>`. Obsérvese que para Colesterol $G^2_{Obs}(cte/(cte, C)) = 154.56 - 146.94 = 7.62$, con p-valor `1-pchisq(7.62,1)`=0.00577 (significativo), para Sistólica $G^2_{Obs}(cte/(cte, S)) = 154.56-148.47 = 6.09$, con p-valor `1-pchisq(6.09,1)`=0.01359 (significativo), para Diastólica $G^2_{Obs}(cte/(cte, D)) = 154.56 - 148.64 = 5.92$, con p-valor `1-pchisq(5.92,1)`=0.01497 (significativo), para Peso $G^2_{Obs}(cte/(cte, P)) = 154.56-150.13 = 4.43$, con p-valor `1-pchisq(4.43,1)`=0.0353 (significativo), y que para Altura $G^2_{Obs}(cte/(cte, A)) = 154.56 - 153.84 = 0.72$, con p-valor `1-pchisq(0.72,1)`=0.3961 (no significatico).

Una vez seleccionada la variable que entra en el modelo, habría que elegir el término que se elimina. Para ello se realizan contrastes condicionales RV que tiene en la hipótesis alternativa el modelo de partida y en la hipótesis nula el modelo que resulta de eliminar de uno en uno cada uno de los términos del modelo de partida. Son candidatos a ser eliminados aquellos términos cuyo contraste sea no significativo (p-valor $> \alpha_2$). El nivel de significación fijado para los contrastes de eliminación de términos es $\alpha_2 = 0.15$.

En el Paso 0 el único término del modelo de partida es la cte que no es usual eliminarla. Es más, al considerar siempre modelos con término cte no nos plantearemos su eliminación.

Como consecuencia de todo lo expuesto, el modelo de partida en el Paso 1 es el que tiene como términos la cte y la variable edad (cte, e), es decir

$$L(e, s, d, c, a, p) = \beta_0 + \beta_E e.$$

Los resultados del ajuste del modelo de partida (cte y E aunque la constante no aparezca) y de la inclusión y eliminación de términos en este modelo figuran en la siguiente tabla:

```
--------------------------------------------------------------------------
Step:  AIC=146.74
Coronarios ~ Edad

            Df Deviance    AIC
+ Peso       1   138.77 144.77
```

```
+ Colesterol  1   139.93 145.93
<none>            142.74 146.74
+ Diastolica  1   141.42 147.42
+ Sistolica   1   141.87 147.87
+ Altura      1   142.74 148.74
- Edad        1   154.56 156.56
-------------------------------------------------------------------------
```

Obsérvese que en este caso aparecen las variables candidatas a entrar en el modelo con un signo `+` y las candidatas a ser eliminadas con un signo `-`. Al igual que en la tabla del paso 0 aparecen los valores de la *deviance* y el *score* de Akaike del modelo que resultaría de añadir cada variable al que tiene sólo la edad (y el término cte). Obsérvese que la *deviance* y el *score* de Akaike de la variable candidata a salir corresponden a los del modelo que tiene la constante que es el que quedaría si se eliminara la Edad. Obsérvese ahora que en la fila `<none>` están la *deviance* y el *score* de Akaike del modelo actual o de partida. Finalmente nótese que de las variables no incluidas en el modelo, las realmente candidatas a entrar (test significativo) están por encima de la fila `<none>` siendo la más candidata y que por tanto entrará efectivamente en el modelo la variable Peso.

Las variables consideradas para inclusión son aquellas que no están en el modelo (ps,pd, cole, altura y peso). Para estas cinco variables se realiza el test condicional entre el modelo de partida (cte,E) y el modelo resultante de su inclusión en éste. Los valores observados de los estadísticos RV condicionales y p-valores no los muestra R pero se calcularían de igual modo que se indicó en el paso 0. Los estadísticos de contraste para las variables candidatas a entrar serían $G^2[(cte,E)/(cte,E,P)] = 142.74-138.77 = 3.97$ y $(G^2[(cte,E)/(cte,E,C)] = 142.74-139.93 = 2.81$ y los p-valores `1-pchisq(3.97,1)`$= 0.0463$) y `1-pchisq(2.81,1)`$=$ 0.0937 respectivamente. Los p-valores asociados al resto de variables que no están en el modelo serían mayores que el nivel de significación de entrada $\alpha_1 =$ 0.1. El mínimo p-valor sería el del peso, por lo que la variable seleccionada para entrar es el peso.

En cada paso se haría la eliminación hacia atrás (*backward*) antes de la inclusión del Peso sobre el modelo de partida (cte,E). En principio, podrían ser eliminados el término cte o la variable edad. Sin embargo la variable edad que entró en el paso anterior no puede ser eliminada porque se han fijado para ello los niveles de significación de entrada ($\alpha_1 = 0.1$) y de salida ($\alpha_2 = 0.15$) verificando que $\alpha_1 < \alpha_2$. Por lo tanto el único término que podría eliminarse es la cte y estamos considerando modelos siempre con término constante, por lo que no se plantea su eliminación.

Como consecuencia el modelo de partida en el Paso 2 es (cte,E,P) formulado

como

$$L(e, s, d, c, a, p) = \beta_0 + \beta_E e + \beta_P p.$$

Los resultados de los contrastes condicionales para la inclusión de términos que aún no están en el modelo y/o eliminación de los que sí están figuran en la siguiente tabla:

```
-------------------------------------------------------------------------------
Step:  AIC=144.77
Coronarios ~ Edad + Peso

             Df Deviance    AIC
+ Colesterol  1   135.52 143.52
<none>            138.77 144.77
+ Altura      1   138.05 146.05
+ Sistolica   1   138.43 146.43
+ Diastolica  1   138.50 146.50
- Peso        1   142.74 146.74
- Edad        1   150.13 154.13
-------------------------------------------------------------------------------
```

El primer paso aquí sería la eliminación hacia atrás (*backward*) sobre el modelo de partida (cte,E,P). En principio, podrían ser eliminados la variable edad o el peso. Como ya se ha explicado anteriormente la variable peso que entró en el paso anterior no puede ser eliminada ya que $\alpha_1 < \alpha_2$. Por lo tanto el únicos término que podrían eliminarse es la variable edad. El estadísticos RV para el contraste condicional entre el modelo que resulta de eliminar cada uno de los términos actuales y el modelo de partida (cte,E,P) se pueden calcular a partir de la *deviance* de las variables que aparecen con un signo - delante. Para la Edad sería $G^2[(cte, P)/(cte, E, P)] = 150.13 - 138.77 = 11.36]$ y para el peso $G^2[(cte, E)/(cte, E, P)] = 142.74 - 138.77 = 3.97]$ con p-valores asociados `1-pchisq(11.36,1)`$= 0.0008$) y `1-pchisq(3.97,1)`$= 0.0463$ respectivamente ambos menores que α_2. Esto significa que ni la variable edad ni peso son eliminadas del modelo.

Las variables consideradas para inclusión en el PASO 2 son aquellas que no están en el modelo (ps,pd, cole y altura). Para estas cuatro variables se realiza el test condicional entre el modelo de partida (cte,E,P) y el modelo resultante de su inclusión en éste. Los valores de los estadísticos RV condicionales se pueden obtener de nuevo a partir de la *deviance* los modelos resultantes de incluir tales variables. En este caso sólo una de las variables está por encima de `<none>` y su p-valor asociado es menor que el nivel de significación de entrada $\alpha_1 = 0.1$, el de la variable colesterol ($G^2[(cte, E, P)/(cte, E, P, C)] =$

$138.77 - 135.52 = 3.25$ y p-valor `1-pchisq(3.25,1)`$= 0.0714$. Por lo tanto, la variable seleccionada para entrar es el colesterol.

Como consecuencia el modelo de partida en el Paso 3 es (cte,E,P,C) formulado como

$$L(e, s, d, c, a, p) = \beta_0 + \beta_E e + \beta_P p + \beta_C c.$$

Los resultados de los contrastes condicionales para la inclusión y eliminación de términos en el siguiente paso figuran en la siguiente tabla:

```
-----------------------------------------------------------------------------
Step:  AIC=143.52
Coronarios ~ Edad + Peso + Colesterol

             Df Deviance    AIC
<none>            135.52 143.52
- Colesterol  1   138.77 144.77
+ Altura      1   134.98 144.98
+ Sistolica   1   135.35 145.35
+ Diastolica  1   135.45 145.45
- Peso        1   139.93 145.93
- Edad        1   142.30 148.30
-----------------------------------------------------------------------------
```

Siguiendo los razonamientos anteriores ninguna variable es candidata a salir del modelo pues todas las que tienen signo `-` están por debajo de la fila `<none>` (se puede comprobar calculando los estadísticos de los contrastes y los p-valores asociados), y ninguna es candidata a entrar en el modelo pues a todas aquellas con signo `+` le ocurre lo mismo.

Las variables que no están en el modelo de partida son presión sistólica, presión diastólica y altura. En este caso los p-valores asociados a los contrastes condicionales para su inclusión son todos mayores que el nivel de significación. Por lo tanto ninguna de estas variables entra en el modelo.

Por otro lado, podrían considerarse para eliminación las variables que están en el modelo (edad, peso y colesterol). Pero los p-valores asociados a los contrastes condicionales entre el modelo que resulta de la eliminación de cada una y el modelo de partida son todos menores que el nivel de significación. Como consequencia ninguna variable es eliminada y el procedimiento de selección stepwise acaba en este paso.

Como consecuencia del procedimiento de selección stepwise se ha seleccionado el modelo que tiene como variables explicativas la edad, el peso y el colesterol

$$L(e, p, c) = \beta_0 + \beta_E e + \beta_P p + \beta_C c.$$

1.4.3 Significación e interpretación de los parámetros

Los resultados del ajuste del modelo figuran en la tabla de resultados de la salida de R siguiente.

```
summary(Ajuste.step)

##
## Call:
## glm(formula = Coronarios ~ Edad + Peso + Colesterol, family = binomial,
##     data = Chapman)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1049  -0.5541  -0.3777  -0.2510   2.7009
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.255892   2.071678  -4.468  7.9e-06 ***
## Edad         0.053004   0.020827   2.545   0.0109 *
## Peso         0.017539   0.008272   2.120   0.0340 *
## Colesterol   0.006518   0.003589   1.816   0.0693 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.55  on 199  degrees of freedom
## Residual deviance: 135.52  on 196  degrees of freedom
## AIC: 143.52
##
## Number of Fisher Scoring iterations: 5
```

Las estimaciones puntuales de los parámetros del modelo son

$$\begin{aligned} \hat{\beta}_0 &= -9.256 \\ \hat{\beta}_E &= 0.053 \\ \hat{\beta}_P &= 0.01754 \\ \hat{\beta}_C &= 0.006518 \end{aligned}$$

Para contrastar la significación estadística de cada uno de los parámetros del modelo usaremos el test estadístico Z de Wald obtenido como el cociente entre

el valor estimado del parámetro y su error estandar (COEF/SE). Fijando nivel de significación $\alpha = 0.05$ se tiene que rechazaremos la hipótesis nula de igualdad a cero del parámetro cuando $|Z| \geq z_{\alpha/2} = 1.96$. Como consecuencia son significativamente distintos de cero el parámetro cte y los parámetros asociados a la edad y el peso. El parámetro asociado al colesterol es cero puesto que $Z_{Obs} = 1.82 < 1.96$.

A continuación pasamos a interpretar la exponencial de cada uno de los parámetros estimados, construyendo, además, un intervalo de confianza del 95% para cada uno de los valores poblacionales asociados.

```
exp(confint.default(Ajuste.step))
```

```
##                    2.5 %      97.5 %
## (Intercept) 1.647409e-06 0.005541574
## Edad        1.012257e+00 1.098366844
## Peso        1.001327e+00 1.034327042
## Colesterol  9.994846e-01 1.013643663
```

Observemos que de estos cuatro intervalos, el único que contiene al uno es el de la exponencial del parámetro asociado al colesterol que es no significativo. Esto significa que tener un infarto es independiente del nivel de colesterol.

En este estudio no tiene sentido interpretar la exponencial de $\hat{\alpha}$ porque sería la ventaja a favor de tener un infarto para individuos con todas las variables nulas, y en este estudio carece de sentido hablar de peso, altura, edad, ... =0.

Para ver la relación entre la edad y el infarto, se estima el cociente de ventajas a favor de tener un infarto cuando se incrementa la edad. Observemos que se trata de un modelo sin interacción y como consecuencia esta relación no depende del peso ni del colesterol.

$$\hat{\theta}(\triangle E = 1/p, c) = \exp\left(\hat{\beta}_E\right) = 1.05.$$

Esto quiere decir que por cada aumneto de un año en la edad la ventaja de infarto se multiplica por 1.05. Si en lugar de incrementar la edad un año, la incrementamos 15 años se tiene

$$\hat{\theta}(\triangle E = 15/p, c) = \exp\left(\hat{\beta}_E\right) = 2.079,$$

que significa que por cada incremento de 15 años en la edad la ventaja a favor de tener un infarto se multiplica aproximadamente por dos. Un intervalo de confianza aproximado de nivel 95% para $\theta(\triangle E = 15/p, c)$ se obtiene elevando a 15 los extremos del I.C. para $\exp(\beta_E)$, y es de la forma $(1.161, 4.177)$.

Para ver la relación entre el peso y el infarto, se estima el cociente de ventajas a favor de tener un infarto cuando se incrementa el peso.

$$\hat{\theta}(\triangle P = 1/e, c) = \exp\left(\hat{\beta}_P\right) = 1.02.$$

Esto significa que por cada aumento de un kilo en el peso la ventaja de infarto se multiplica por 1.02. En este caso tendremos que aumentar el peso 35 kilos para que la ventaja de tener un infarto se duplique

$$\hat{\theta}(\triangle P = 35/p, c) = \exp\left(\hat{\beta}_E\right) = 1.02^{35} = 1.9999.$$

Un intervalo de confianza aproximado de nivel 95% para $\theta(\triangle E = 35/e, c)$ se obtiene elevando a 35 los extremos del I.C. para $\exp(\beta_P)$, y es de la forma $(1, 2.814)$.

1.4.4 Bondad del ajuste

Una vez estimado el modelo vamos a contrastar la bondad del ajuste. En este caso hay una única observación de la respuesta para cada combinación de valores observados de las variables explicativas (datos no agrupados). Esto quiere decir que la frecuencia esperada de infartos ($Y = 1$) coincide con la probabilidad estimada de infarto. Por lo tanto todas las frecuencias esperadas de infarto son menores que 5 y, como consecuencia, el test más adecuado para estudiar la bondad del ajuste es el Test de Hosmer-Lemeshow. Utilizando la función propuesta en explicaciones anteriores el resultado es

```
hosmerlem.test(Chapman,fitted.values(Ajuste.all),g=10,group=F)
```

```
##           Hosmer-Lemeshow Test
## X-squared           12.7928473
## p.value              0.1191786
```

Que al proporcional un p-valor mayor que $\alpha = 0.05$ nos permite afirmar que el modelo se ajusta globalmente bien a los datos.

1.4.5 Validación

La validación como siempre se lleva a cabo a partir de los residuos estandarizados y medidas de influencia.

```
rstandard(Ajuste.step,type="pearson")
rstandard(Ajuste.step,type="deviance")
cooks.distance(Ajuste.step)
```

Recordemos que los residuos son significativos (distintos de cero) al nivel de confianza $\alpha = 0.05$ cuando sus valores ajustados son en valor absoluto mayores o iguales que 1.96. Para identificar los casos con residuos significativos, podemos ordenar los residuos con la función (sort()):

El número de residuos significativos es 13 que de un total de 200 observaciones representa el 6.5% de las observaciones. Sin embargo, en todos ellos la influencia es muy pequeña y no significativa. Por lo tanto se puede concluir que el modelo se ajusta bien dato a dato.

1.4.6 Precisión del modelo logístico y clasificación de observaciones

Tabla de clasificación

Obtenemos la tabla de clasificación con 0.5 como punto de corte

```
Categoria.Pred<-ifelse(fitted.values(Ajuste.step)>=0.5,1,0)
Categoria.Pred
```

Y la tabla de clasificación:

```
table(Chapman$Coronarios,Categoria.Pred)
```

```
##    Categoria.Pred
##       0   1
##   0 174   0
##   1  24   2
```

En nuestro ejemplo la tabla de clasificación ha sido

	Clasificación	
Observación	Fracaso	éxito
Fracaso	VN=174	FP=0
éxito	FN=24	VP=2

Y las medidas:

- Tasa de clasificaciones correctas: CCR =88%

- Sensibilidad: 7.69%
- Especificidad: 100%

Cualquier otro punto de corte produciría otros modos de clasificación.

Curva ROC

Recoredemos que para generar el objeto de tipo *curva roc* la función `roc()` el vector de la variable respuesta (de ceros y unos) y el vector de predicciones.

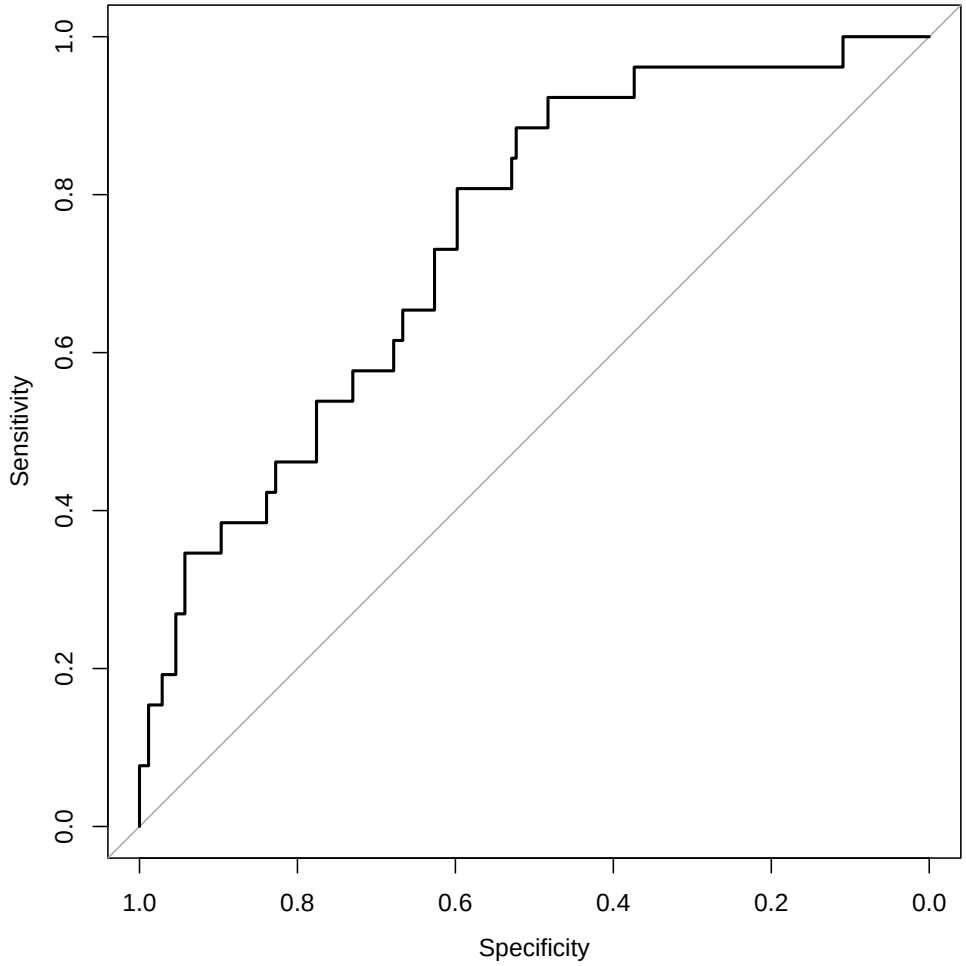
```
CurvaROC<-roc(Chapman$Coronarios,fitted.values(Ajuste.step))
CurvaROC

##
## Call:
## roc.default(response = Chapman$Coronarios, predictor = fitted.values(Ajust
##
## Data: fitted.values(Ajuste.step) in 174 controls (Chapman$Coronarios 0) <
## Area under the curve: 0.746
```

Recordemos que la única salida de la función es el área bajo la curva ROC, pero permite mostrar el resto de elementos: puntos de corte, sensibilidades y especificidades. Se omitirán aquí todas estas salidas por ser muy extensas y no aportar nada nuevo en la explicación.

```
CurvaROC$thresholds
CurvaROC$sensitivities
CurvaROC$specificities
```

Con la función `plot` aplicada a un objeto de tipo *curva roc* se obtiene la gráfica de la curva ROC.

```
plot(CurvaROC)
```



1.4.7 Interacción

Ahora vamos a plantearnos la posibilidad de introducir en el modelo alguna interacción para intentar reducir el número de residuos significativos y ver si la variable colesterol es significativa para explicar la respuesta.

La consideración de interacción se introduce en R en el argumento la `formula` mediante el separador `*` que le indica a la función `glm()` que considere los efectos principales y las interacciones. En nuestro caso la fórmula sería `Coronarios~Edad*Peso*Colesterol`

Para ver si hay alguna interacción significativa vamos a aplicar el método de selección paso a paso tomando como modelo de partida el modelo seleccionado anteriormente, que tiene como variables explicativas la edad, el peso y el colesterol. Los resultados de los contrastes RV condicionales para la selección

forward y la eliminación backward a partir de este modelo figuran en la siguiente tabla.

```
Ajuste.Step.Inter<-step(Ajuste.step,
scope=list(lower=Coronarios~Edad+Peso+Colesterol,
upper=Coronarios~Edad*Peso*Colesterol),direction="both")

## Start:  AIC=143.52
## Coronarios ~ Edad + Peso + Colesterol
##
##                    Df Deviance    AIC
## + Peso:Colesterol   1   132.39 142.39
## <none>                  135.52 143.52
## + Edad:Colesterol   1   134.71 144.71
## + Edad:Peso         1   135.45 145.45
##
## Step:  AIC=142.39
## Coronarios ~ Edad + Peso + Colesterol + Peso:Colesterol
##
##                    Df Deviance    AIC
## <none>                  132.39 142.39
## + Edad:Colesterol   1   131.37 143.37
## - Peso:Colesterol   1   135.52 143.52
## + Edad:Peso         1   132.37 144.37
```

Como el modelo de partida tiene tres variables explicativas, hay tres posibles interacciones de orden uno que podrían ser considerarlas para inclusión en el modelo: Edad*Peso, Edad*Colesterol y Peso*Colesterol.

Observemos en la tabla que de los contrastes RV condicionales que comparan el modelo de partida con el que resulta de introducir en él cada interacción, el único significativo con p-valor menor o igual que el nivel de significación de entrada es el de la interacción entre el peso y el colesterol (p*c) que, como consecuencia, entra en el modelo.

Con respecto a la eliminación backward, podrían considerarse para su eliminación las variables edad, peso, colesterol y la cte. Ninguno de los p-valores asociados a los contrastes RV condicionales para eliminar cada uno de estos términos en el modelo de partida, es mayor que el nivel de significación. Por lo tanto ninguna de estas variables sale del modelo. En cualquier caso si está en el modelo la interacción entre dos variables debe también estar el término individual de cada variable por separado para que el modelo resultante sea jerárquico.

Como consecuencia de todo lo expuesto, el modelo resultante del Paso 0 es el siguiente:

$$L(e, s, d) = \beta_0 + \beta_E e + \beta_P p + \beta_C c + \beta_{PC} p * c.$$

En primer lugar se realizan contrastes condicionales RV para comparar el modelo de partida con el que resulta de la inclusión de cada una de las dos interacciones que no están en el modelo (e*p y e*c). Ninguna de las dos interacciones candidatas a entrar en el modelo en este paso tienen p-valores menores o iguales que el nivel de significación de entrada. Por lo tanto ninguna de ellas entra en el modelo. Por otro lado, ninguno de los términos del modelo de partida puede salir excepto la cte. A pesar de ello la cte no será eliminada porque estamos considerando modelos con cte.

El modelo seleccionado mediante selección stepwise es

$$L(e, p, c) = \beta_0 + \beta_E e + \beta_P p + \beta_C c + \beta_{PC} p * c$$

```
summary(Ajuste.Step.Inter)

##
## Call:
## glm(formula = Coronarios ~ Edad + Peso + Colesterol + Peso:Colesterol,
##     family = binomial, data = Chapman)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4837  -0.5296  -0.3695  -0.2596   2.4478
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      6.8900451  9.7226839   0.709  0.47854
## Edad             0.0570503  0.0212679   2.682  0.00731 **
## Peso            -0.0793026  0.0583097  -1.360  0.17382
## Colesterol      -0.0503332  0.0340574  -1.478  0.13944
## Peso:Colesterol  0.0003369  0.0002012   1.675  0.09403 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.55  on 199  degrees of freedom
```

```
## Residual deviance: 132.39  on 195  degrees of freedom
## AIC: 142.39
##
## Number of Fisher Scoring iterations: 5
```

Parámetros estimados

$$\begin{aligned}\hat{\beta}_0 &= 6.890\\ \hat{\beta}_E &= 0.05705\\ \hat{\beta}_P &= -0.0793\\ \hat{\beta}_C &= -0.05033\\ \hat{\beta}_{PC} &= 0.0003369\end{aligned}$$

En la tabla del ajuste del modelo podemos ver que el modelo se ajusta globalmente bien. En lo que respecta a la significación estadística de los parámetros, sólo es significativo el parámetro asociado a la variable edad. En el resto el valor absoluto del estadístico Z de Wald es menor que 1.96 y por lo tanto se acepta que estos parámetros son nulos.

A continuación analizaremos la relación entre la enfermedad coronaria y la edad, el peso y el colesterol en base a los parámetros estimados. Veamos en primer lugar la relación entre la edad y el infarto.

$$\hat{\theta}(\triangle E = 1/p, c) = \exp(0.05705) = 1.06.$$

Por cada incremento de 10 años en la edad se tiene

$$\hat{\theta}(\triangle E = 10/p, c) = 1.06^{10} = 1.8 \simeq 2.$$

Un intervalo de confianza del 95% para $\theta(\triangle E = 10/p, c)$ es de la forma

$$(1.02^{10}, 1.10^{10}) = (1.219, 2.594).$$

Como consecuencia se puede concluir que por cada aumento de 10 años en la edad la ventaja a favor de tener un infarto se duplica.

Veamos ahora que como el modelo tiene interacción entre el peso y el colesterol, la relación entre el infarto y cualquiera de ellas depende de la otra. Para estudiar por ejemplo, la relación entre el peso y el infarto calculamos el siguiente cociente de ventajas:

$$\hat{\theta}(\triangle P = 1/e, c) = \exp(-0.0793 + 0.0003369c),$$

que vale 1 cuando $c = 235.3814$. Por lo tanto, para valores del colesterol superiores a 235.3814, ocurre que $\hat{\theta}(\triangle P = 1/e, c) > 1$. Esto significa que la probabilidad de tener infarto aumenta al aumentar el peso cuando el colesterol es superior a esta cantidad.

Para poder extrapolar esto a la población habría que contrastar la significación estadística de este cociente de ventajas en base a un intervalo de confianza para su valor poblacional.

En primer lugar calcularíamos un intervalo de confianza para $\ln[\theta(\triangle P = 1/e, c)]$

$$\ln[\hat{\theta}(\triangle P = 1/e, c)] \pm \hat{\sigma}\left(\ln[\hat{\theta}(\triangle P = 1/e, c)]\right) z_{\alpha/2}.$$

Posteriormente se obtendía un I.C. aproximado para $\theta(\triangle P = 1/e, c)$ tomando exponenciales en los extremos del intervalo para su logaritmo.

Para ello calcularíamos en primer lugar la varianza dada por

$$\begin{aligned}\hat{\sigma}^2\left(\ln\left(\hat{\theta}(\triangle P = 1/e, c)\right)\right) &= \hat{\sigma}^2(\hat{\beta}_P + \hat{\beta}_{PC}c) \\ &= \hat{\sigma}^2(\hat{\beta}_P) + c^2\hat{\sigma}^2(\hat{\beta}_{PC}) + 2 * c * \hat{Cov}(\hat{\beta}_P, \hat{\beta}_{PC}) \\ &= (0.583E-01)^2 + c^2(0.201E-03)^2 + 2 * c * (-1.1589E-05),\end{aligned}$$

que depende del nivel de colesterol c.

Por ejemplo, para una persona con nivel de colesterol c=300 se tiene que

$$\hat{\theta}(\triangle P = 1/e, c) = \exp(-0.0793 + 0.0003369 * 300) = \exp(0.02177) = 1.022.$$

La varianza estimada de $\ln(\hat{\theta}(\triangle P = 1/e, c))$ es $8.158E{-}05$. Como consecuencia, un intervalo de confianza de nivel 95% para $\ln(\theta(\triangle P = 1/e, c))$ es de la forma

$$0.02176 \pm 1.96 * \sqrt{8.158E-05} = (0.004057, 0.03946).$$

Tomando exponenciales en los extremos se obtiene un intervalo de confianza aproximado del 95% para $\theta(\triangle P = 1/e, c)$ dado por $(1.0041, 1.0402)$. Esto significa que este cociente de ventajas es significativamente distinto de uno.

Si aumentamos el peso en 35 libras se tiene que para los individuos con nivel de colesterol 300, la ventaja a favor de tener un infarto se duplica

$$\hat{\theta}(\triangle P = 35/e, c) = (1.022)^{35} = 2.141.$$

El intervalo de confianza para $\theta(\triangle P = 35/e, c)$ es $(1.0041^{35}, 1.0402^{35}) = (1.1540, 3.9727)$.