Aplicaciones de modelos lineales generalizados

Los modelos lineales generalizados son una familia de modelos para el análisis estadístico. Incluye la regresión lineal y logística como casos especiales.

Un modelo lineal generalizado consiste de:

1. Un vector de datos $y = (y_1, \ldots, y_n)$
2. Predictores X y coeficientes β para construir un predictor lineal $X\beta$.
3. Una función de enlace g que da como resultado datos transformados

$$E(Y) = g^{-1}(X\beta)$$

   que son usados para modelar los datos.
4. Una distribución para los datos $p(y|X)$.
5. Posiblemente otros parámetros, como varianza, o puntos de corte, involucrados en los predictores, o bien, la función enlace o la distribución de los datos.

La regresión lineal predice directamente datos continuos y de un predictor lineal $X\beta = \beta_0 + X_1\beta_1 + \cdots + X_k\beta_k$.

`Otros modelos que vamos a ver son:`

1. El modelo *logístico-binomial* se utiliza en casos cuando los datos observados y_i representan el número de éxitos en n_i ensayos independientes. En este modelo la función liga es logit y la distribución de los datos es binomial. Al igual que con la regresión Poisson, el modelo binomial típicamente se puede mejorar agregando un parámetro de sobredispersión.

2. El modelo *probit* es igual que regresión logística pero se reemplaza la función liga por la *distribución normal acumulada*. Se puede pensar como usar la distribución normal en los errores estimados del modelo.

3. El *modelo Poisson* se utiliza para datos de *conteos*; es decir, donde cada dato observado y_i puede ser igual a $0, 1, 2, \ldots$. La función liga que se utiliza habitualmente g es logarítmica, de modo que $g(x) = \exp(x)$ transforma un predictor lineal continuo $X_i\beta$ en un y_i positivo. La distribución de datos es Poisson. A veces es buena idea agregar un parámetro a este modelo para capturar la **sobredispersión**, es decir, la variación en los datos más allá de la que captura el modelo.

Función glm()

glm(variable respuesta $\sim$ variable explicativa1 + ...+variable explicativa p, family = familia(link="función de enlace"), data = datos)

Familia Función de enlace por defecto:

binomial (link = "logit")

gaussian (link = "identity")

Gamma (link = "inverse")

inverse.gaussian (link = "1/mu^2")

poisson (link = "log")

quasi (link = "identity", variance = "constant")

quasibinomial (link = "logit")

quasipoisson (link = "log")

Regresión Binomial: Datos binarios

La regresión logística predice $P(y = 1)$ para datos binarios a partir de un predictor lineal transformado por la función logística inversa.

En el siguiente ejemplo se modela la probabilidad de fraude por impago (default = Y) en función del balance de la cuenta bancaria (balance = X).

Ejemplo 1: Interpretación de coeficientes de regresión

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ISLR)
datos <- Default

head(datos)
```

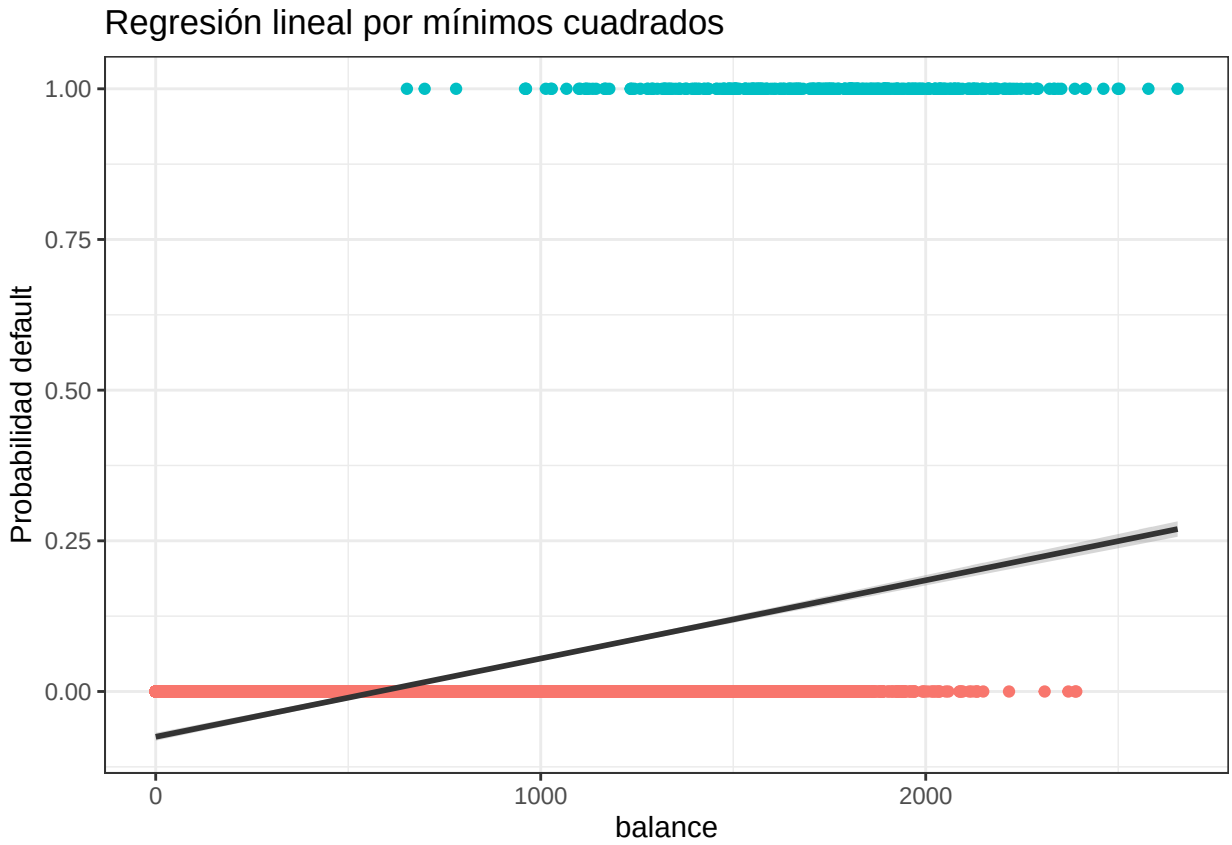
```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

```
# Se recodifican los niveles No, Yes a 0 y 1
datos <- datos %>%
  select(default, balance) %>%
  mutate(default = recode(default,
                          "No"  = 0,
                          "Yes" = 1))
head(datos)
```

```
##   default   balance
## 1       0  729.5265
## 2       0  817.1804
## 3       0 1073.5492
## 4       0  529.2506
## 5       0  785.6559
## 6       0  919.5885
```

```
# Modelo de regresión Lineal
modelo_lineal <- lm(default ~ balance, data = datos)

# Representación gráfica del modelo.
```

```
ggplot(data = datos, aes(x = balance, y = default)) +
  geom_point(aes(color = as.factor(default))) +
  geom_smooth(method = "lm", color = "gray20") +
  theme_bw()  +
  labs(title = "Regresión lineal por mínimos cuadrados",
       y = "Probabilidad default") +
  theme(legend.position = "none")
```




```
predict(object = modelo_lineal, newdata = data.frame(balance = 100))
```

```
##           1
## -0.06220474
```

Modelo mal planteado, se predice la probabilidad de default para alguien que tiene un balance de 100, el valor obtenido es menor que 0.

Modelo de regresión Logística binaria

La asociación entre X e Y es no lineal, el modelo de regresión logística asocia la variable explicativa X con la media de Y a través de una función de enlace.

$$E(y_i) = \pi$$

$$logit(\pi) = \beta_0 + \beta_1 x$$

$$p(Y = 1|X = x) = \pi = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Puede interpretarse como: la probabilidad de que la variable cualitativa Y adquiera el valor 1 (el nivel de referencia, codificado como 1, probabilidad de éxito), dado que el predictor X tiene el valor x.

A partir de dicha definición se tiene que:

$$logit(\pi) = \beta_0 + \beta_1 x.$$

Ajuste de un modelo logístico usando la función glm():

```
modelo_logistico <- glm(default ~ balance, data = datos, family = "binomial")

summary(modelo_logistico)
```

```
##
## Call:
## glm(formula = default ~ balance, family = "binomial", data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <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: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

```
modelo_logistico <- glm(default ~ balance, data = datos, family = "binomial")

summary(modelo_logistico)
```

```
##
## Call:
## glm(formula = default ~ balance, family = "binomial", data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <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: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

beta 0 → ## (Intercept) -1.065e+01 3.612e-01 -29.49 <2e-16 ***

beta 1 → ## balance 5.499e-03 2.204e-04 24.95 <2e-16 ***

H0: beta0 = 0 (no significativo)
H1: beta0 !=0 (significativo)
p-valor < 0.05 entonces rechazo H0, beta0 es significativo
H0: beta1 = 0 (no significativo)
H1: beta1 !=0 (significativo)
pvalor < 0.05 entonces rechazo H0, beta1 es significativo

“

```
names(modelo_logistico)
```

```
##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "call"              "formula"           "terms"
## [25] "data"              "offset"            "control"
## [28] "method"            "contrasts"         "xlevels"
```

```
vcov(modelo_logistico)
```

```
##               (Intercept)       balance
## (Intercept)  1.304346e-01 -7.817111e-05
## balance     -7.817111e-05  4.856301e-08
```

```
# Ajuste del modelo
pchisq(deviance(modelo_logistico),df.residual(modelo_logistico) ,lower=FALSE)
```

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

H0: Modelo actual (buen ajuste)

H1: Modelo saturado

Si p-valor < 0.05 rechazo Hipótesis nula

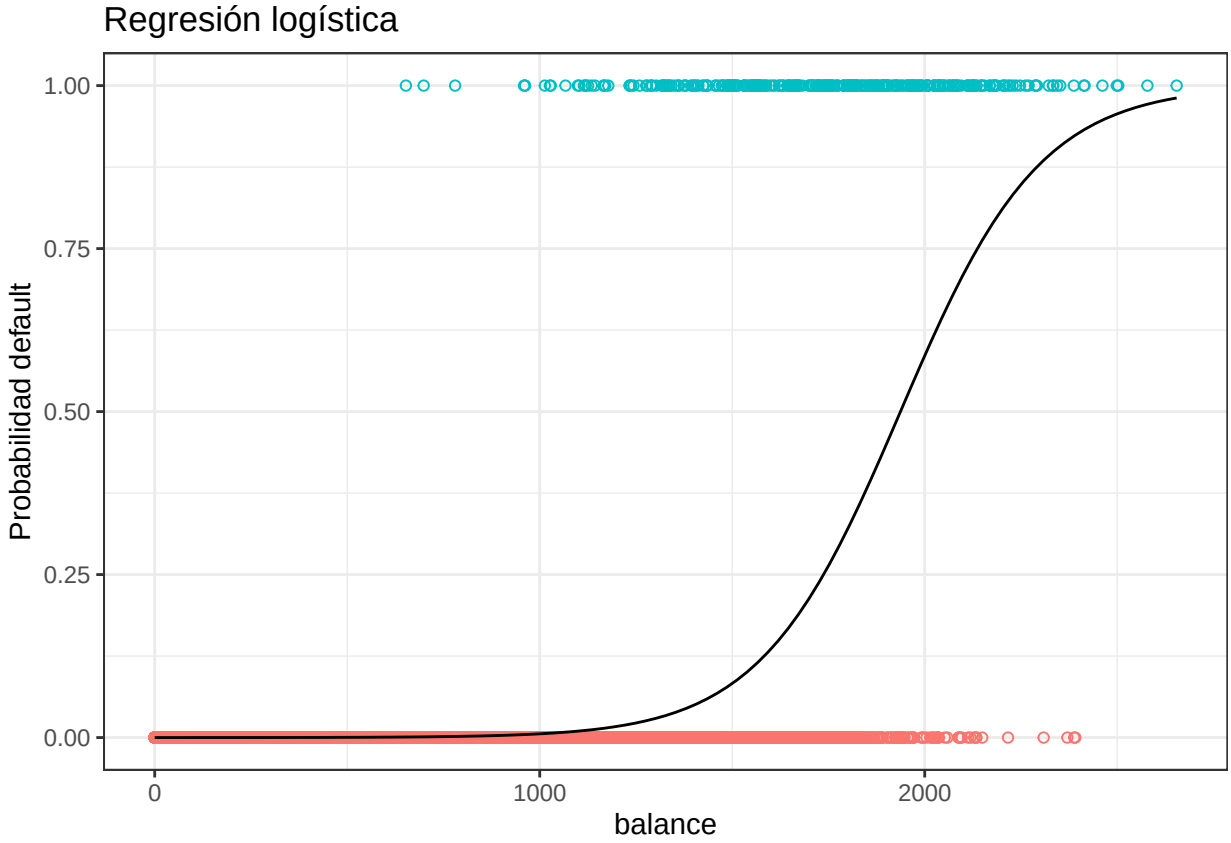
COmo p-valor $=1 > 0.05$, no rechazo la hipótesis nula, por lo tanto el modelo actual presenta buen ajuste.

```
# "Estimación"" de y para los valores observados:
ypred <- predict(modelo_logistico, type = "response")
```

```r
ypred[1:10]
```

```
##            1            2            3            4            5
## 1.305680e-03 2.112595e-03 8.594741e-03 4.344368e-04 1.776957e-03
##            6            7            8            9           10
## 3.704153e-03 2.211431e-03 2.016174e-03 1.383298e-02 2.366877e-05
```

```r
# Representación gráfica del modelo.
ggplot(data = datos, aes(x = balance, y = default)) +
  geom_point(aes(color = as.factor(default)), shape = 1) +
  stat_function(fun = function(x){predict(modelo_logistico,
                                          newdata = data.frame(balance = x),
                                          type = "response")}) +
  theme_bw() +
  labs(title = "Regresión logística",
       y = "Probabilidad default") +
  theme(legend.position = "none")
```



Sea $\pi = p(Y = 1|X = x)$, se define el ODD como:

$$ODD = \pi/(1 - \pi)$$

Suponga que la probabilidad de éxito es de 0.8, por lo que la probabilidad de fracaso es de 1 - 0.8 = 0.2. *Los ODDs (o razón de probabilidad) de éxitos se definen como el ratio entre la probabilidad de éxito y la probabilidad de fracaso p/q.

$$ODDS(exito) = probabilidad(éxito)/probabilidad(fracaso)$$

En este caso los ODDs de éxito son 0.8 / 0.2 = 4, lo que equivale a decir que se esperan 4 éxitos por cada fracaso. Nos define quién es más probable el éxito o el fracaso, según sus probabilidades respectivas.

La trasformación de probabilidades a ODDs es monótona, si la probabilidad aumenta también lo hacen los ODDs, y viceversa. El rango de valores que pueden tomar los ODDs es de [0,∞].

Dado que el valor de una probabilidad está acotado entre [0,1] se recurre a una trasformación logit (existen otras) que consiste en el logaritmo natural de los ODDs. Esto permite convertir el rango de probabilidad previamente limitado a [0,1] a $[-\infty, +\infty]$.

De esta forma,

$$LOG(ODDS) = ln(\frac{p(Y=1|X=x)}{1-p(Y=1|X=x)}) = \beta_0 + \beta_1 X = \eta,$$

en la regresión logística, tal como se ha descrito en la sección anterior, se modela la probabilidad de que la variable respuesta Y pertenezca al nivel de referencia 1 en función del valor que adquieran los predictores, mediante el uso de LOG of ODDs.

```
library(knitr)
p<- c(0.001,0.01, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.999,0.9999)
odds <- p/(1-p)
df <- data.frame(p ,odds , log(odds))

kable(df)
```

p	odds	log.odds.
0.0010	0.0010010	-6.9067548
0.0100	0.0101010	-4.5951199
0.2000	0.2500000	-1.3862944
0.3000	0.4285714	-0.8472979
0.4000	0.6666667	-0.4054651
0.5000	1.0000000	0.0000000
0.6000	1.5000000	0.4054651
0.7000	2.3333333	0.8472979
0.8000	4.0000000	1.3862944
0.9000	9.0000000	2.1972246
0.9990	999.0000000	6.9067548
0.9999	9999.0000000	9.2102404

Los ODDs y el logaritmo de ODDs cumplen que:

Si p(exito) = p(fracaso), entonces odds(exito) = 1

Si p(exito) < p(fracaso), entonces odds(exito) < 1

Si p(exito) > p(fracaso), entonces odds(exito) > 1

A diferencia de la probabilidad que no puede exceder el 1, los ODDs no tienen límite superior.

Si odds(exito) = 1, entonces logit(p) = 0

Si odds(exito) < 1, entonces logit(p) < 0

Si odds(exito) > 1, entonces logit(p) > 0

La transformación logit no existe para p = 0.

La razon de ODDS = $\frac{\frac{p(Y=1|X=x+1)}{1-p(Y=1|X=x+1)}}{\frac{p(Y=1|X=x)}{1-p(Y=1|X=x)}} = e^{\beta_1}$ indica el cambio en el logaritmo de ODDs de éxito debido al incremento de una unidad de X, es decir e^{β_1} es el cambio en el ODDS de éxito cuando el valor de la variable explicativa aumenta en una unidad.

En la literatura se sugiere interpretar el cociente de ODSS como el "aumento estimado" en la probabilidad de éxito asociado con un cambio unitario en el valor de la variable explicativa.

En general, el aumento estimado del cociente de ODDS, asociado con un cambio de d unidades en la variable predictora, es $e^{(d\beta^1)}$.

```
modelo_logistico$coefficients
```

```
##   (Intercept)       balance
## -10.651330614   0.005498917
```

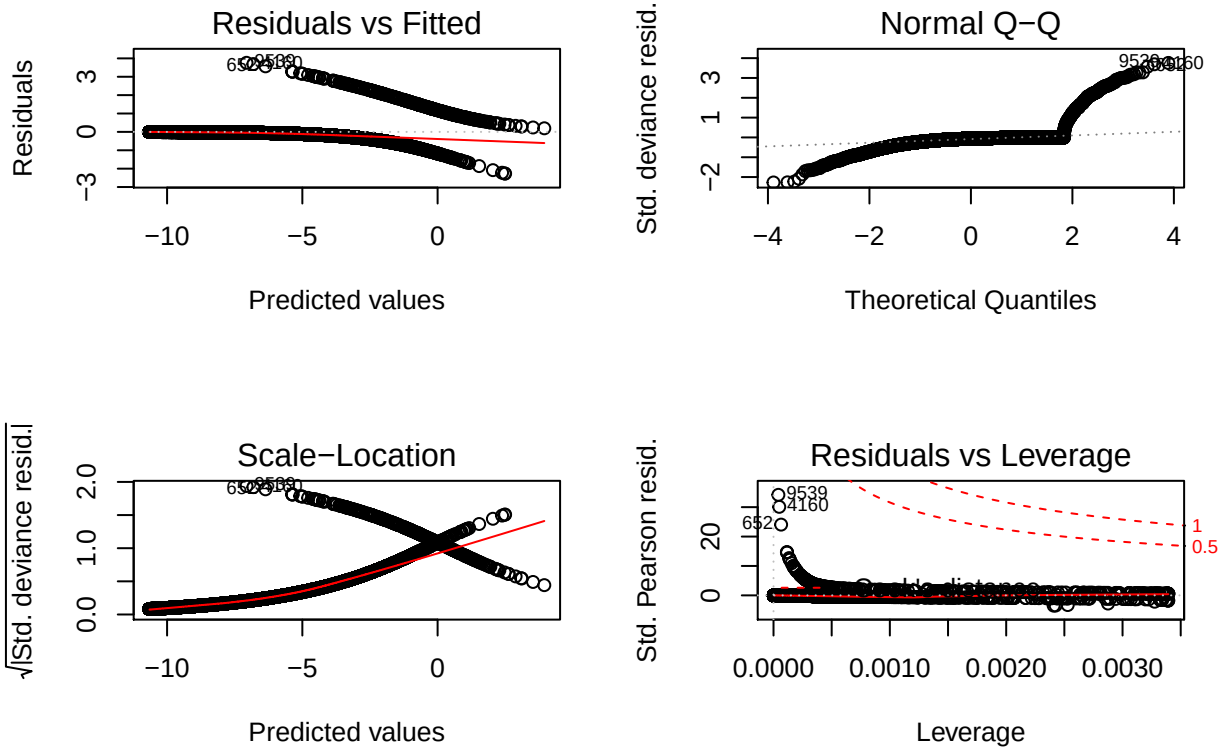
El ODDS ((p/(1-p))) de un cliente realizar un fraude cuando tiene un balance de x+1 es $e^{0.005498917} = 1.005514$ veces el ODDS de un cliente realizar un fraude cuando tiene un balance x. De forma equivalente, el ODDS de un cliente realizar un fraude se incrementa en 0.05% por cada u.m. adicional en su balance.

En la práctica "se asume" que por cada u.m adicional en el balance aumenta (1- 1.005514)% =0.05%, la probabilidad de un cliente realizar un fraude.

Entonces por cada 10 u.m. adicionales en el balance de un cliente, $e^{0.005498917*10} = 1.056529$, el ODDS del cliente realizar un fraude se incrementa en 5%.

Por cada 100 u.m. adicionales en el balance de un cliente, $e^{(100*0.005498917)} = 1.733065$, entonces el ODDS de un cliente realizar un fraude se incrementa en 73%.

```
op <- par(mfrow = c(2, 2))
plot(modelo_logistico)
```



Ejemplo 2: Interpretación de curvas ROC -

Datos binarios de sobrevivientes del Titanic

El conjunto de datos es una colección de datos sobre algunos de los pasajeros, y el objetivo es predecir la supervivencia (1 si el pasajero sobrevivió o 0 si no lo hizo) basándose en algunas características como la clase

de servicio, el sexo, la edad, etc. Como puede ver, vamos a utilizar variables categóricas y continuas.

Descripción de las variables:

pclass Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd) survival Survival (0 = No; 1 = Yes) name Name sex Sex age Age sibsp Number of Siblings/Spouses Aboard parch Number of Parents/Children Aboard ticket Ticket Number fare Passenger Fare cabin Cabin embarked Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton) boat Lifeboat body Body Identification Number home_dest Home/Destination

$$logit(\pi) = \beta_0 + \beta_1 x_1 + \ldots, \beta_p x_p$$

```
train <- read.csv('http://idaejin.github.io/courses/R/data/titanic_train.csv',header=TRUE,row.names=1)
 test <- read.csv('http://idaejin.github.io/courses/R/data/titanic_test.csv',header=TRUE,row.names=1)


model <- glm(Survived ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6064  -0.5954  -0.4254   0.6220   2.4165
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.137627   0.594998   8.635  < 2e-16 ***
## Pclass      -1.087156   0.151168  -7.192 6.40e-13 ***
## Sexmale     -2.756819   0.212026 -13.002  < 2e-16 ***
## Age         -0.037267   0.008195  -4.547 5.43e-06 ***
## SibSp       -0.292920   0.114642  -2.555   0.0106 *
## Parch       -0.116576   0.128127  -0.910   0.3629
## Fare         0.001528   0.002353   0.649   0.5160
## EmbarkedQ   -0.002656   0.400882  -0.007   0.9947
## EmbarkedS   -0.318786   0.252960  -1.260   0.2076
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  709.39  on 791  degrees of freedom
## AIC: 727.39
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
```

```
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                       799    1065.39
## Pclass    1   83.607       798     981.79 < 2.2e-16 ***
## Sex       1  240.014       797     741.77 < 2.2e-16 ***
## Age       1   17.495       796     724.28 2.881e-05 ***
## SibSp     1   10.842       795     713.43  0.000992 ***
## Parch     1    0.863       794     712.57  0.352873
## Fare      1    0.994       793     711.58  0.318717
## Embarked  2    2.187       791     709.39  0.334990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod1 <- glm(Survived ~ as.factor(Pclass), family=binomial, data=train)
summary(mod1)
```

```
##
## Call:
## glm(formula = Survived ~ as.factor(Pclass), family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3787  -0.7515  -0.7515   0.9887   1.6747
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.4616     0.1474   3.131  0.00174 **
## as.factor(Pclass)2  -0.5455     0.2138  -2.551  0.01074 *
## as.factor(Pclass)3  -1.5816     0.1844  -8.575  < 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: 1065.39  on 799  degrees of freedom
## Residual deviance:  979.94  on 797  degrees of freedom
## AIC: 985.94
##
## Number of Fisher Scoring iterations: 4
```

```
anova(mod1,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
```

```
##                   Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                799    1065.39
## as.factor(Pclass)  2   85.452       797     979.94 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod2 <- glm(Survived ~ Pclass + Sex  + Age + SibSp, family = binomial(logit), data = train)
summary(mod2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp, family = binomial(logit),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6595  -0.6125  -0.4247   0.6149   2.4302
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.05604    0.50130  10.086  < 2e-16 ***
## Pclass      -1.14391    0.12585  -9.089  < 2e-16 ***
## Sexmale     -2.75564    0.20471 -13.461  < 2e-16 ***
## Age         -0.03725    0.00812  -4.588 4.48e-06 ***
## SibSp       -0.33075    0.10892  -3.037  0.00239 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  713.43  on 795  degrees of freedom
## AIC: 723.43
##
## Number of Fisher Scoring iterations: 5
```

```
anova(mod2,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     799    1065.39
## Pclass  1   83.607       798     981.79 < 2.2e-16 ***
## Sex     1  240.014       797     741.77 < 2.2e-16 ***
## Age     1   17.495       796     724.28 2.881e-05 ***
## SibSp   1   10.842       795     713.43  0.000992 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod3 <-  glm(Survived ~ Pclass + Sex + Pclass:Sex + Age + SibSp, family = binomial(logit), data = train)
summary(mod3)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Pclass:Sex + Age + SibSp,
##     family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1993  -0.6265  -0.4770   0.4485   2.3093
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     7.606411   0.960804   7.917 2.44e-15 ***
## Pclass         -2.108360   0.316024  -6.672 2.53e-11 ***
## Sexmale        -5.887480   0.920417  -6.397 1.59e-10 ***
## Age            -0.038063   0.008498  -4.479 7.50e-06 ***
## SibSp          -0.310269   0.109370  -2.837 0.004556 **
## Pclass:Sexmale  1.254202   0.338241   3.708 0.000209 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1065.39  on 799  degrees of freedom
## Residual deviance:  695.66  on 794  degrees of freedom
## AIC: 707.66
##
## Number of Fisher Scoring iterations: 6
```

```r
anova(mod3,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                         799    1065.39
## Pclass      1   83.607       798     981.79 < 2.2e-16 ***
## Sex         1  240.014       797     741.77 < 2.2e-16 ***
## Age         1   17.495       796     724.28 2.881e-05 ***
## SibSp       1   10.842       795     713.43  0.000992 ***
## Pclass:Sex  1   17.779       794     695.66 2.481e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#clasificación
#Si  $P(Y=1|x) > 0.5$ entonces Y=1, caso contrario Y=0.
#Tener en cuenta que para algunas aplicaciones diferentes límites de #decisión podría ser una mejor opc
```

```
fitted.results <- predict(mod3,newdata=test,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results != test$Survived)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.8075"
```

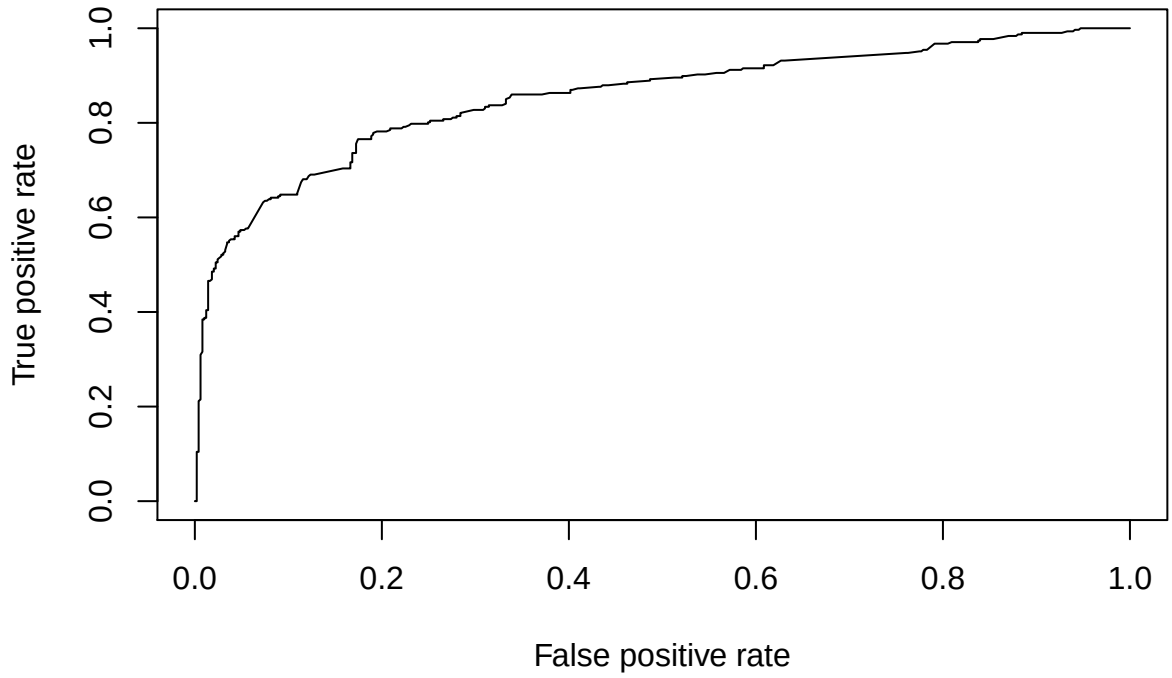
```
#Evaluar la capacidad predictiva

library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
p <- predict(mod3, newdata=subset(test,select=c(2,3,4,5,6,7,8)), type="response")
pr <- ROCR::prediction(p, test$Survived)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8543155
```

```
#El modelo tiene mejor desempeño si la curva ROC se aleja más de la diagonal principal
# El área bajo la curva ROC representa la probabilidad de
# clasificar correctamente a los individuos.
```

```r
# área ROC = 0.5 (Poder discriminante nulo)
# 0.7 < área ROC < 0.8 (Poder discriminante  Aceptable)
# 0.8 < área ROC < 0.9 (Poder discriminante Excelente)
#  área ROC > 0.9 (Poder discriminante Excepcionalmente buena)

library(InformationValue)
predicted <- predict(mod3, test, type="response")  # predicted scores
optCutOff <- optimalCutoff(test$Survived, predicted)[1]
optCutOff
```

```
## [1] 0.53401
```

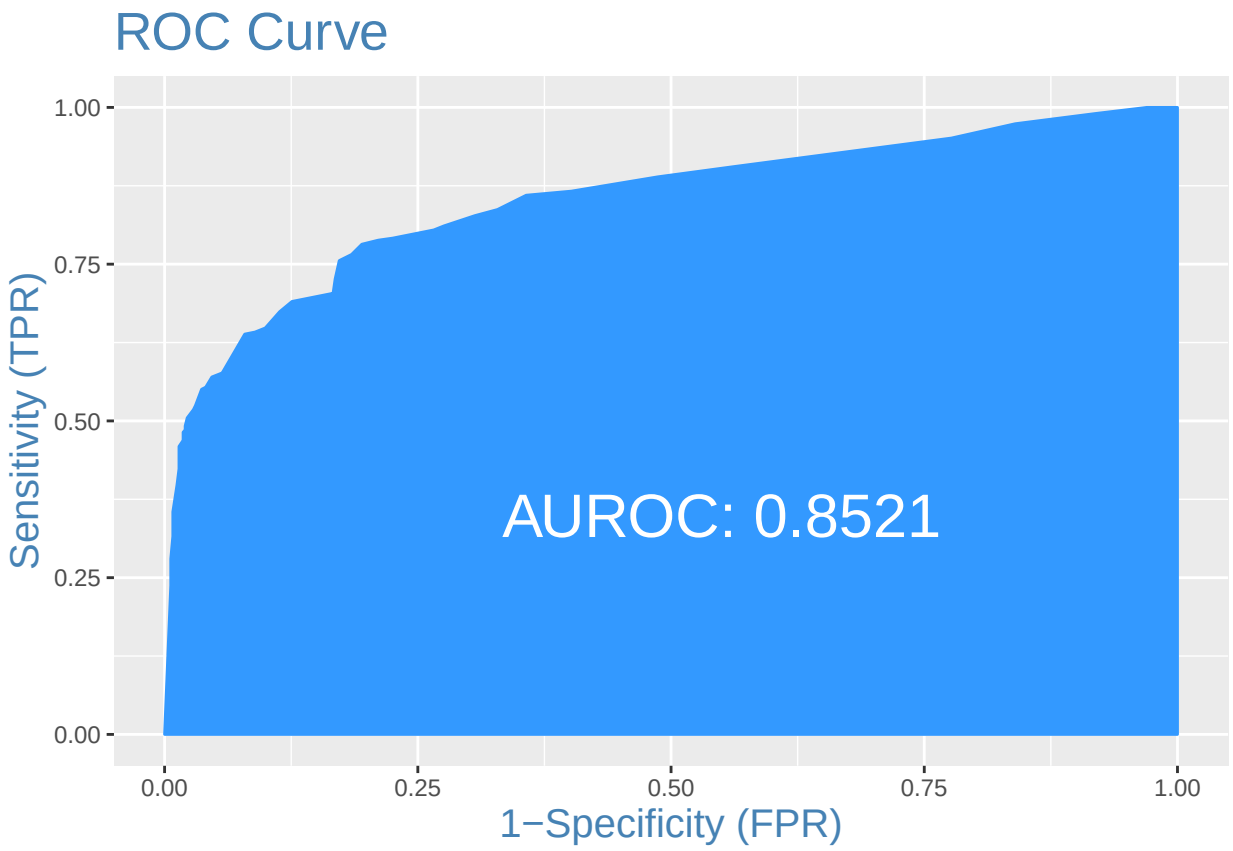
```r
misClassError(test$Survived, predicted, threshold = optCutOff)
```

```
## [1] 0.1862
```

```r
1-misClassError(test$Survived, predicted, threshold = optCutOff)
```

```
## [1] 0.8138
```

```r
plotROC(test$Survived, predicted)
```



Regresión Binomial: proporciones

Ejemplo: Orings

El 28 de enero de 1986, se anticipaba un lanzamiento de rutina para el Challenger transbordador espacial. Setenta y tres segundos durante el vuelo, ocurrió un desastre: el transbordador se rompió, matando a los tripulantes a bordo. Una investigación sobre la causa del desastre se centró en un sello crítico llamado O-ring, y se cree que el daño a estos o-rings durante el lanzamiento de un transbordador puede estar relacionado con la temperatura ambiente durante el lanzamiento. La siguiente tabla resume los datos de observación de los o-rings para 23 misiones de lanzamiento. La unidad de la temperatura es en Fahrenheit, Dañado representa el número de o-rings dañados, y No dañado representa el número de o-rings que no sufrieron daños.

$$Y_i \sim Binomial(n_i, \pi_i)$$

$$logit(\pi) = \beta_0 + \beta_1 x$$

```
library(faraway)
library(tidyverse)
library(visreg)
library(broom)

# datos
orings
```

```
##    temp damage
## 1    53      5
## 2    57      1
## 3    58      1
## 4    63      1
## 5    66      0
## 6    67      0
## 7    67      0
## 8    67      0
## 9    68      0
## 10   69      0
## 11   70      1
## 12   70      0
## 13   70      1
## 14   70      0
## 15   72      0
## 16   73      0
## 17   75      0
## 18   75      1
## 19   76      0
## 20   76      0
## 21   78      0
## 22   79      0
## 23   81      0
```

```
#?orings

as_tibble(orings) %>%
  mutate(
```

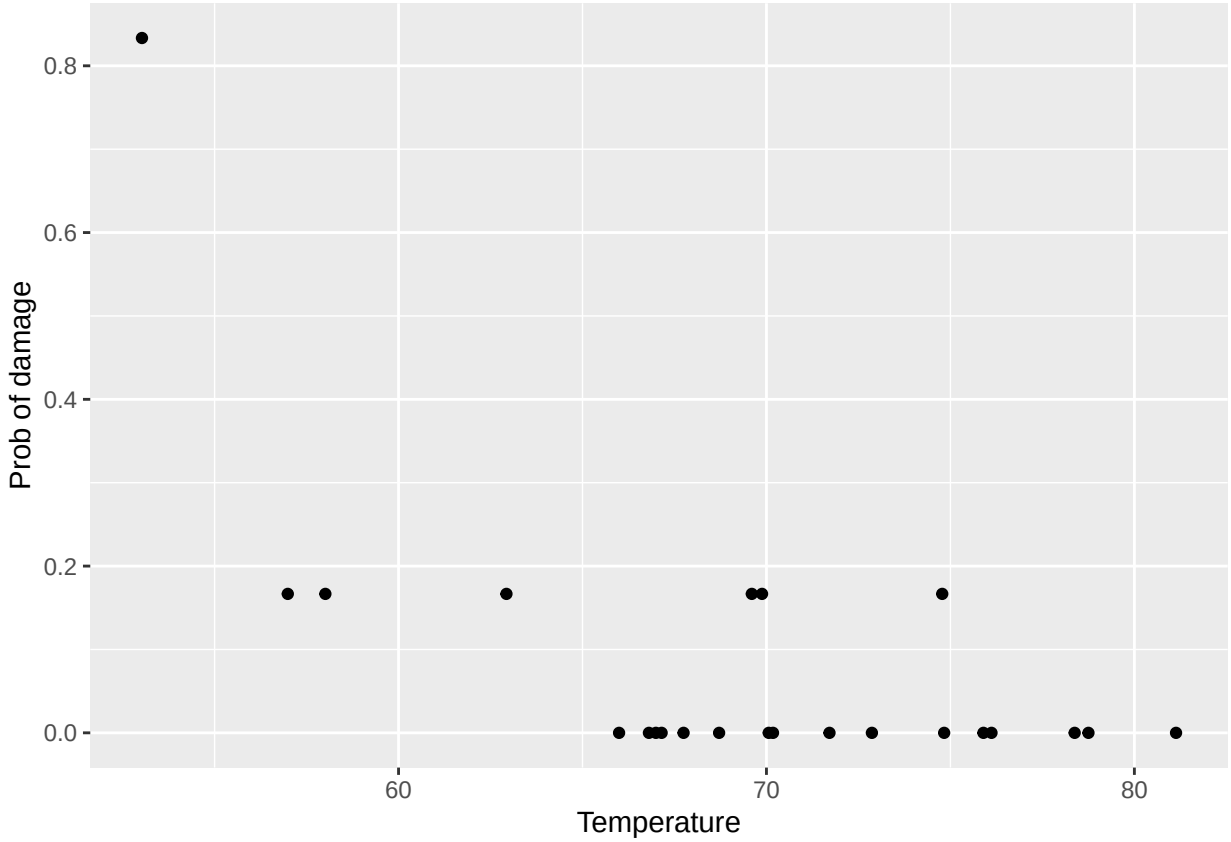
```
    undamaged=6-damage,
    pdamage = damage/6) ->
  orings

orings
```

```
## # A tibble: 23 x 4
##     temp damage undamaged pdamage
##    <dbl>  <dbl>     <dbl>   <dbl>
##  1    53      5         1   0.833
##  2    57      1         5   0.167
##  3    58      1         5   0.167
##  4    63      1         5   0.167
##  5    66      0         6   0
##  6    67      0         6   0
##  7    67      0         6   0
##  8    67      0         6   0
##  9    68      0         6   0
## 10    69      0         6   0
## # ... with 13 more rows
```

```
summary(orings)
```

```
##       temp           damage         undamaged        pdamage
##  Min.   :53.00   Min.   :0.0000   Min.   :1.000   Min.   :0.00000
##  1st Qu.:67.00   1st Qu.:0.0000   1st Qu.:5.000   1st Qu.:0.00000
##  Median :70.00   Median :0.0000   Median :6.000   Median :0.00000
##  Mean   :69.57   Mean   :0.4783   Mean   :5.522   Mean   :0.07971
##  3rd Qu.:75.00   3rd Qu.:1.0000   3rd Qu.:6.000   3rd Qu.:0.16667
##  Max.   :81.00   Max.   :5.0000   Max.   :6.000   Max.   :0.83333
```

```
ggplot(orings, aes(x=temp, y=pdamage)) +
  geom_jitter(height=0) +
  xlab("Temperature") + ylab("Prob of damage")
```




```
lmod <- glm(cbind(damage,undamaged) ~ temp,
            family=binomial, orings)
summary(lmod)
```

```
##
## Call:
## glm(formula = cbind(damage, undamaged) ~ temp, family = binomial,
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9529  -0.7345  -0.4393  -0.2079   1.9565
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 11.66299    3.29626   3.538 0.000403 ***
## temp        -0.21623    0.05318  -4.066 4.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 16.912  on 21  degrees of freedom
## AIC: 33.675
##
## Number of Fisher Scoring iterations: 6
```

```r
# Ajuste del modelo
pchisq(deviance(lmod),df.residual(lmod) ,lower=FALSE)
```

```
## [1] 0.7164099
```

```r
#p-valor>0.05, no rechazo H0:modelo actual buen ajuste.

drop1(lmod, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(damage, undamaged) ~ temp
##        Df Deviance    AIC    LRT  Pr(>Chi)
## <none>      16.912 33.675
## temp    1   38.898 53.660 21.985 2.747e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
exp(lmod$coefficients)
```

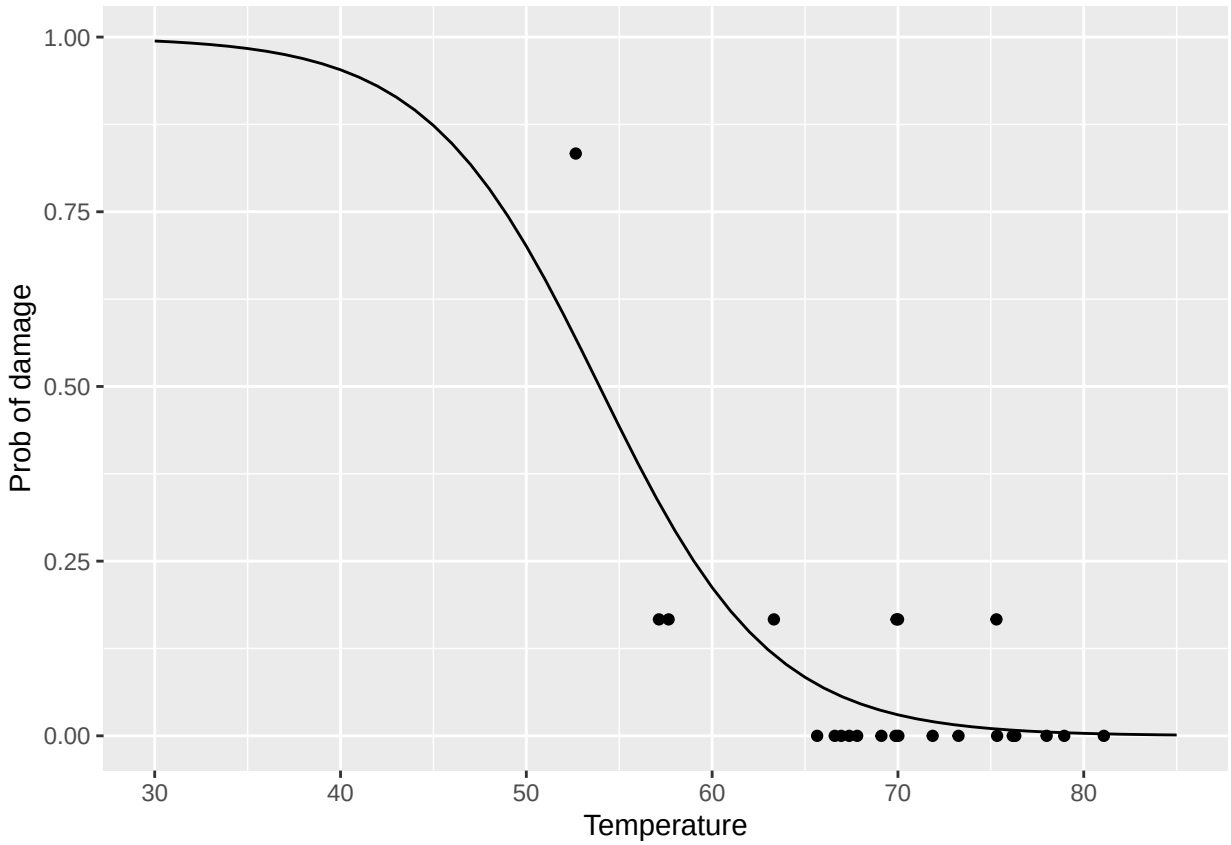
```
##  (Intercept)         temp
## 1.161909e+05 8.055471e-01
```

```r
confint(lmod)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept)  5.575195 18.737598
## temp        -0.332657 -0.120179
```

```r
newdf <- tibble(
  temp = 30:85,
  pred = predict(lmod,
            newdata=data.frame(temp=temp),
            type='response')
)

ggplot(orings, aes(x=temp, y=pdamage)) +
  geom_jitter(height=0) +
  xlab("Temperature") + ylab("Prob of damage") +
  geom_line(aes(x=temp, y=pred), data=newdf)
```



Interpretación del coeficiente de regresión: Por cada grado farenheit que se incremente la temperatura, la posibilidad de un o-ring dañado con temperatura x+1 es exp(-0.21) = 0.80 veces la posibilidad de un o-ring dañado con temperatura x. Es decir La posibilidad de un o-ring dañado se recuce el 20% por cada grado farenheit de temperatura adicional.

$$logit(\pi) = \beta_0 + \beta_1 x + \beta_2 x_1^2$$

```
# Try quadratic

lmod2 <- glm(cbind(damage,undamaged) ~ temp + I(temp^2),
             family=binomial, orings)
summary(lmod2)
```

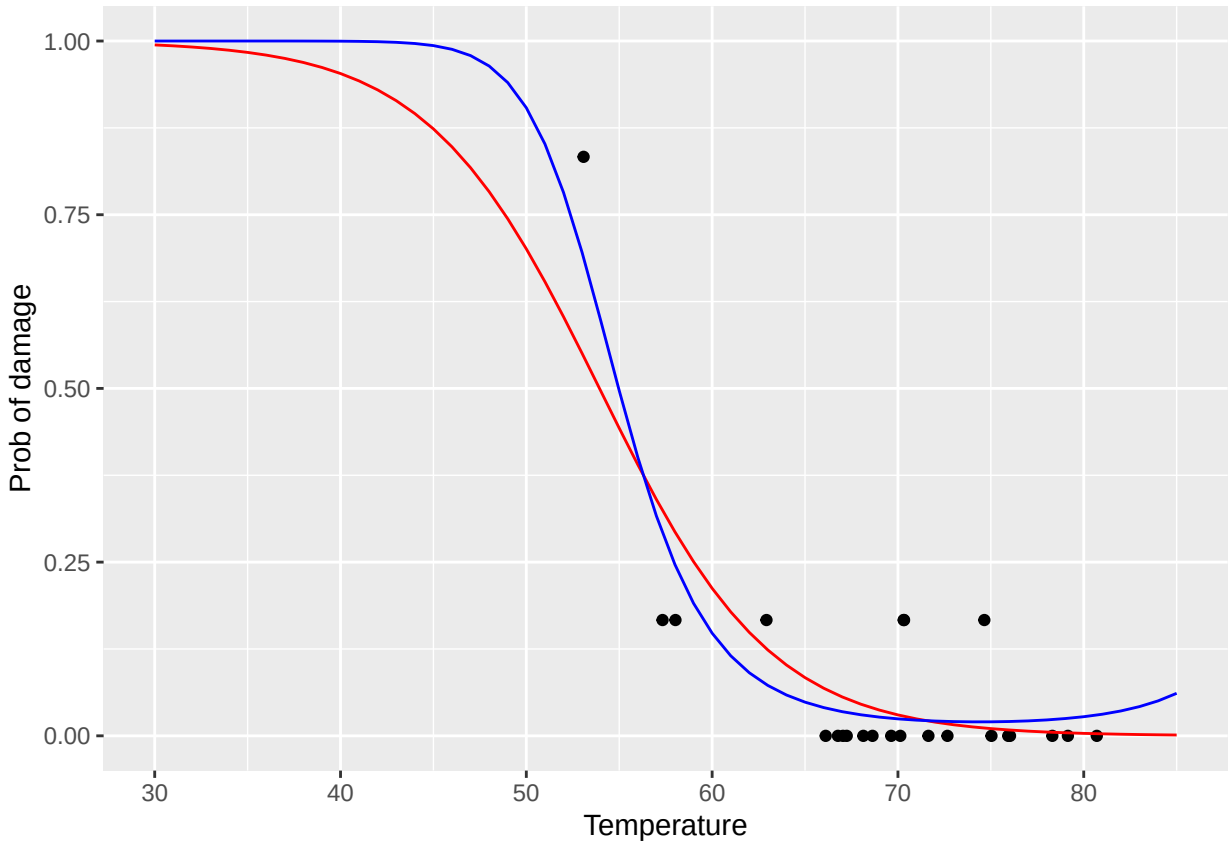
```
##
## Call:
## glm(formula = cbind(damage, undamaged) ~ temp + I(temp^2), family = binomial,
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8360  -0.6119  -0.5272  -0.4837   1.6137
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 53.09113   25.94876   2.046   0.0408 *
## temp        -1.53165    0.80792  -1.896   0.0580 .
## I(temp^2)    0.01029    0.00620   1.660   0.0969 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 14.526  on 20  degrees of freedom
## AIC: 33.288
##
## Number of Fisher Scoring iterations: 5
```

```
drop1(lmod2, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(damage, undamaged) ~ temp + I(temp^2)
##           Df Deviance    AIC    LRT Pr(>Chi)
## <none>         14.526 33.288
## temp       1   17.672 34.434 3.1458  0.07612 .
## I(temp^2)  1   16.912 33.675 2.3864  0.12240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
newdf2 <- tibble(
  temp = 30:85,
  pred = predict(lmod2,
        newdata=data.frame(temp=temp),
        type='response')
)

ggplot(orings, aes(x=temp, y=pdamage)) +
  geom_jitter(height=0) +
  xlab("Temperature") + ylab("Prob of damage") +
  geom_line(aes(x=temp, y=pred), col='red', data=newdf) +
  geom_line(aes(x=temp, y=pred), col='blue', data=newdf2)
```



Otras funciones de enlace

Regresión Probit

$$\Phi^{-1}(\pi) = \beta_0 + \beta_1 x$$

```
lmod3 <- glm(cbind(damage,undamaged) ~ temp,
             family=binomial(link="probit") ,orings)
summary(lmod3)
```

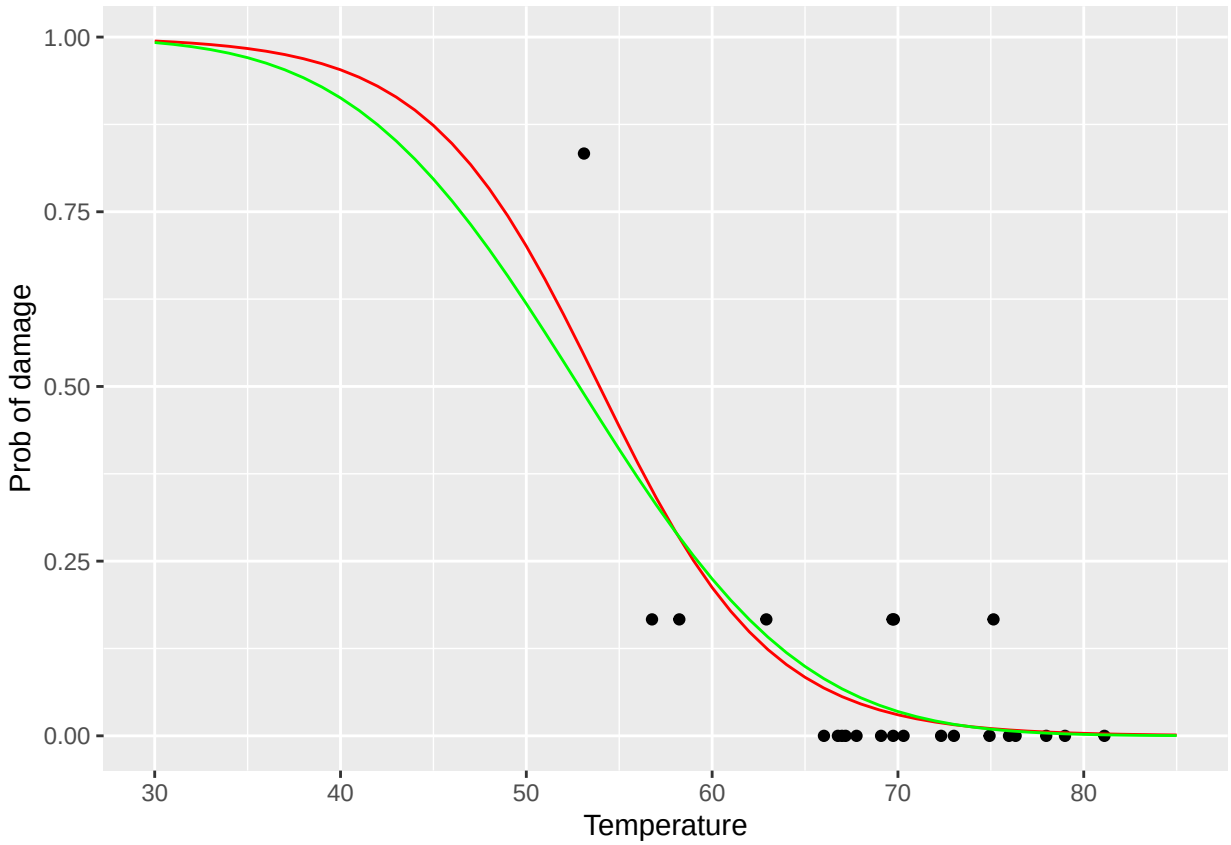
```
##
## Call:
## glm(formula = cbind(damage, undamaged) ~ temp, family = binomial(link = "probit"),
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0134  -0.7761  -0.4467  -0.1581   1.9983
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.59145    1.71055   3.269  0.00108 **
## temp        -0.10580    0.02656  -3.984 6.79e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
## Residual deviance: 18.131  on 21  degrees of freedom
## AIC: 34.893
##
## Number of Fisher Scoring iterations: 6
```

```
drop1(lmod3, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(damage, undamaged) ~ temp
##        Df Deviance    AIC    LRT  Pr(>Chi)
## <none>      18.131 34.893
## temp    1   38.898 53.660 20.767 5.187e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
newdf3 <- tibble(
  temp = 30:85,
  pred = predict(lmod3,
        newdata=data.frame(temp=temp),
        type='response')
)

ggplot(orings, aes(x=temp, y=pdamage)) +
  geom_jitter(height=0.0) +
  xlab("Temperature") + ylab("Prob of damage") +
  geom_line(aes(x=temp, y=pred), col='red', data=newdf) +
  geom_line(aes(x=temp, y=pred), col='green', data=newdf3)
```



$$\Phi^{-1}(\pi) = \beta_0 + \beta_1 x + \beta_2 x_1^2$$

```
lmod4 <- glm(cbind(damage,undamaged) ~ temp + I(temp^2),
             family=binomial(link="probit") ,orings)
summary(lmod4)
```

```
##
## Call:
## glm(formula = cbind(damage, undamaged) ~ temp + I(temp^2), family = binomial(link = "probit"),
##     data = orings)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9204  -0.6320  -0.5444  -0.4843   1.6414
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 28.139415  12.284691   2.291   0.0220 *
## temp        -0.812870   0.377946  -2.151   0.0315 *
## I(temp^2)    0.005468   0.002872   1.904   0.0569 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 38.898  on 22  degrees of freedom
```

```
## Residual deviance: 15.180  on 20  degrees of freedom
## AIC: 33.943
##
## Number of Fisher Scoring iterations: 6
```

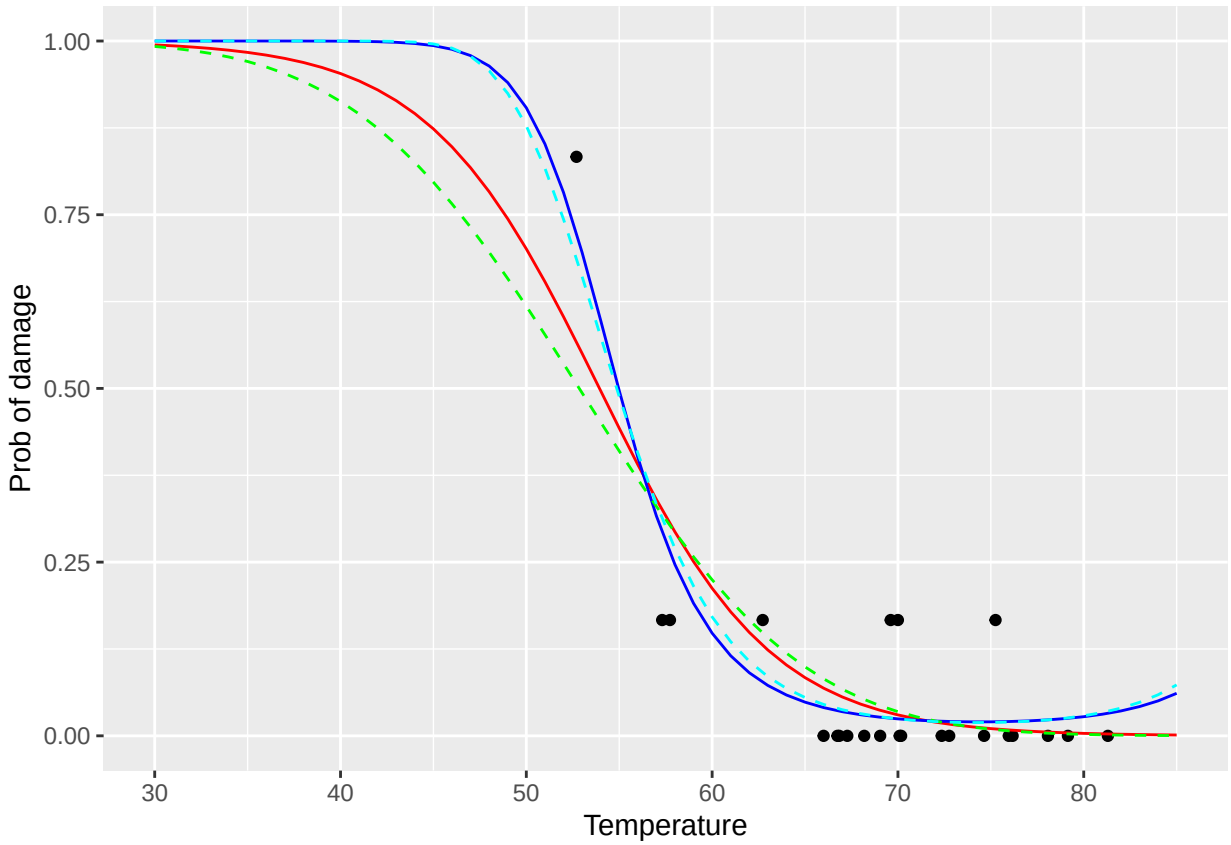
```
drop1(lmod4, test="Chisq")
```

```
## Single term deletions
##
## Model:
## cbind(damage, undamaged) ~ temp + I(temp^2)
##           Df Deviance    AIC    LRT Pr(>Chi)
## <none>         15.180 33.943
## temp       1   18.985 35.748 3.8050  0.05110 .
## I(temp^2)  1   18.131 34.893 2.9505  0.08585 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pchisq(deviance(lmod4),df.residual(lmod4) ,lower=FALSE)
```

```
## [1] 0.7660123
```

```
#anova(lmod3, test="Chisq") va agregando las variables explicativas de forma secuencial
# verificando...
# lmod4 <- glm(cbind(damage,undamaged) ~  I(temp^2),
#              family=binomial(link="probit") ,orings)
# anova(lmod4, test="Chisq")
#drop1(lmod4, test="Chisq")

newdf4 <- tibble(
  temp = 30:85,
  pred = predict(lmod4,
        newdata=data.frame(temp=temp),
        type='response')
)

ggplot(orings, aes(x=temp, y=pdamage)) +
  geom_jitter(height=0.0) +
  xlab("Temperature") + ylab("Prob of damage") +
  geom_line(aes(x=temp, y=pred), col='red', data=newdf) +
  geom_line(aes(x=temp, y=pred), col='blue', data=newdf2) +
  geom_line(aes(x=temp, y=pred),linetype = "dashed", col='green', data=newdf3)+
  geom_line(aes(x=temp, y=pred),linetype = "dashed", col='cyan', data=newdf4)
```


1.00
0.75
0.50
0.25
0.00
Prob of damage
30
40
50
60
70
80
Temperature