Curso de Estadística básica para Data Scientists

Dae-Jin Lee < lee.daejin@gmail.com >

TEMA 7. Regresión para datos binarios y de conteo

Índice

Regresar a la página principal

1. Regresión Logística

Normalmente se utiliza una regresión logística cuando hay una variable de resultado dicotómica (como ganar o perder) y una variable predictora continua que está relacionada con la probabilidad o las probabilidades de la variable de resultado. También puede usarse con predictores categóricos y con múltiples predictores.

Si usamos una regresión lineal para modelar una variable dicotómica (como Y), el modelo resultante podría no restringir los Y's previstos dentro de 0 y 1. Además, otros supuestos de regresión lineal como la normalidad de errores pueden ser violados. Así que en su lugar, modelamos las probabilidades log del evento $\ln(\frac{p}{1-p})$ o logit, donde, p es la probabilidad del evento.

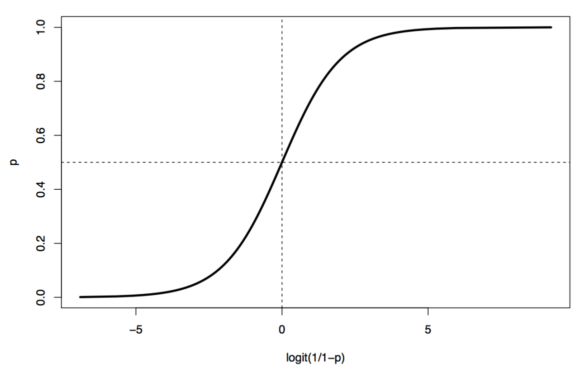
$$z_i = \ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

La ecuación anterior se puede modelar usando `glm()` colocando el argumento `family "binomial"`. Pero estamos más interesados en la probabilidad del evento, que en las probabilidades logarítmicas del evento. Por lo tanto, los valores predichos del modelo anterior, es decir, las probabilidades logarítmicas del evento, se pueden convertir en probabilidad de evento como sigue:

$$p_i = 1 - \frac{1}{1+\exp(z_i)}$$

Esta conversión se logra utilizando la función `plogis()`.

La función `logit` tiene la forma



1.1. ESR y proteínas de plasma

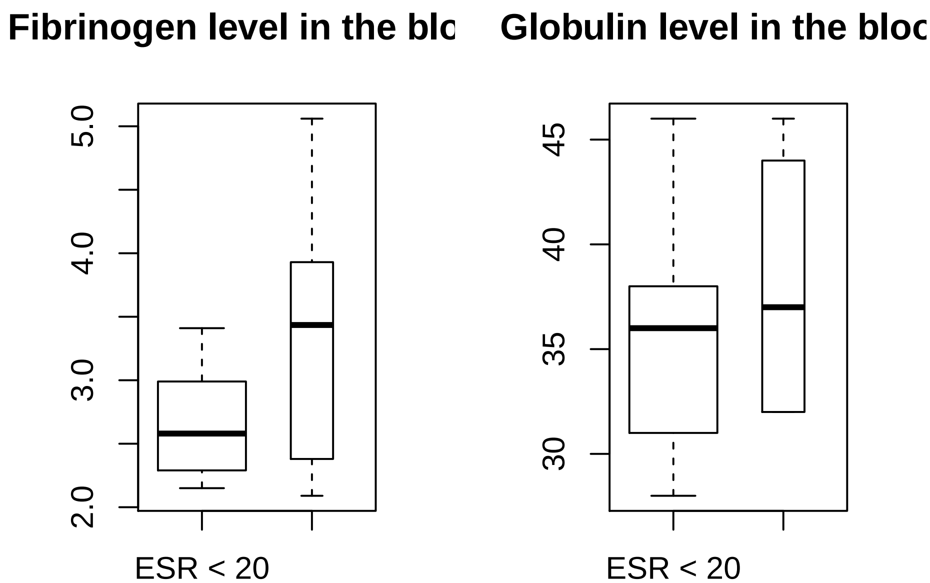
La velocidad de sedimentación de eritrocitos (ESR - *erythrocyte sedimentation rate*) es la velocidad a la que los glóbulos rojos (eritrocitos) se asientan fuera de suspensión en el plasma sanguíneo, cuando se miden en condiciones estándar. Si la ESR aumenta cuando el nivel de ciertas proteínas en el plasma sanguíneo se eleva en asociación con afecciones tales como enfermedades reumáticas, infecciones crónicas y enfermedades malignas, su determinación podría ser útil en el cribado de muestras de sangre tomadas de personas sospechosas de padecer una de las condiciones mencionados. El valor absoluto de la ESR no es de gran importancia; más bien, menos de 20mm/hr indica un individuo "sano". Para evaluar si la ESR es una herramienta de diagnóstico útil, la cuestión de interés es si existe alguna asociación entre la probabilidad de una lectura de ESR mayor de 20mm/hr y los niveles de las dos proteínas plasmáticas. Si no es así, la determinación de ESR no sería útil para fines de diagnóstico. Un marco de datos con 32 observaciones sobre las 3 variables siguientes.

- `fibrinogen` el nivel de fibrinógeno en la sangre.
- `globulin` el nivel de globulina en la sangre.
- `ESR` la velocidad de sedimentación de los eritrocitos, sea menor o mayor de 20 mm/hora.


```
data("plasma", package = "HSAUR")
head(plasma)
```

```
##   fibrinogen globulin      ESR
## 1       2.52       38 ESR < 20
## 2       2.56       31 ESR < 20
## 3       2.19       33 ESR < 20
## 4       2.18       31 ESR < 20
## 5       3.41       37 ESR < 20
## 6       2.46       36 ESR < 20
```

```
layout(matrix(1:2, ncol = 2))
boxplot(fibrinogen ~ ESR, data = plasma, varwidth = TRUE, main="Fibrinogen level in the bloo
boxplot(globulin ~ ESR, data = plasma, varwidth = TRUE, main="Globulin level in the blood")
```



La cuestión de interés es si existe alguna asociación entre la probabilidad de una lectura ESR superior a 20 mm / hr y los niveles de las dos proteínas plasmáticas. Si no es así, la determinación de ESR no sería útil para fines de diagnóstico.

Dado que la variable de respuesta es binaria, un modelo de regresión múltiple no es adecuado para un análisis de regresión.

Podemos escribir

$$\mathbb{P}\text{r}(y_i = 1) = \pi_i \qquad \mathbb{P}\text{r}(y_i = 0) = 1 - \pi_i$$

El modelo

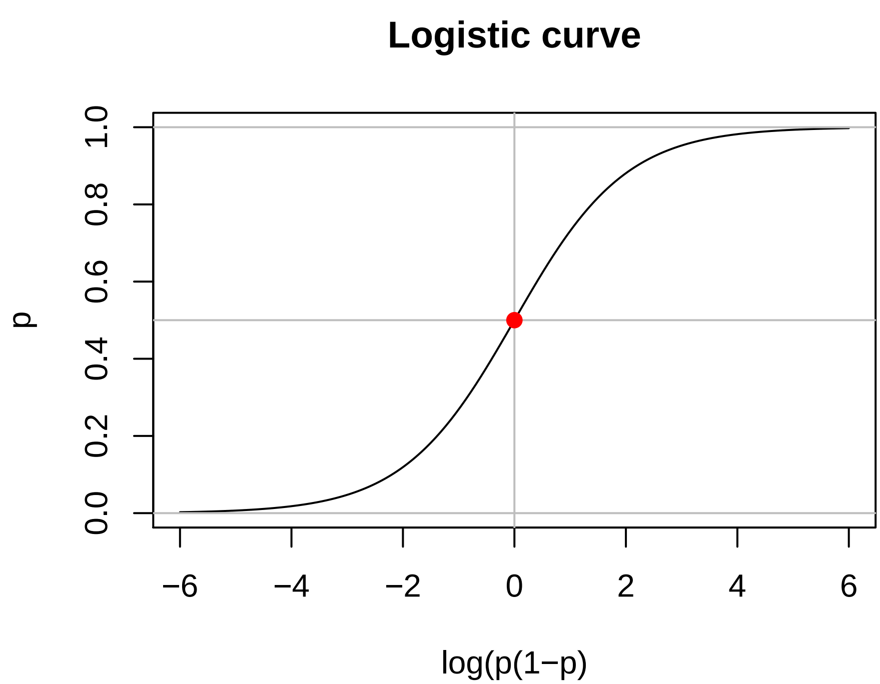
$$\text{logit}(\pi) = \text{logit}\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

El logit de una probabilidad es simplemente el log de las probabilidades de la respuesta tomando el valor una transformación logit o de p: $logit(p) = \log(p/1{-}p)$. Propiedades

- Si `odds(y=1) = 1`, entonces `logit(p) = 0`.
- Si `odds(y=1) < 1`, entonces `logit(p) < 0`.
- Si `odds(y=1) > 1`, entonces `logit(p) > 0`.

Cuando la respuesta es una variable binaria (dicotómica), y x es numérica, la regresión logística ajusta una curva logística a la relación entre x y y. Por lo tanto, la regresión logística es la regresión lineal en la transformación logit de y, donde y es la proporción (o probabilidad) de éxito en cada valor de x. Sin embargo, se evitar la tentación de hacer una regresión lineal, ya que ni la normalidad ni la suposición de homoscedasticidad se satisfacen.

```
x <- seq(-6,6,0.01)
logistic <- exp(x)/(1+exp(x))
plot(x,logistic,t='l',main="Logistic curve",ylab="p",xlab="log(p(1-p)")
abline(h=c(0,0.5,1),v=0,col="grey")
points(0,0.5,pch=19,col=2)
```



La regresión logística en `R` se ajusta mediante la función `glm`. En primer lugar, comenzamos con el modelo que incluye como variable explicativa `fibrinogen`

```
plasma_glm_1 <- glm(ESR ~ fibrinogen, data = plasma,family = binomial())
summary(plasma_glm_1)
```

```
##
## Call:
## glm(formula = ESR ~ fibrinogen, family = binomial(), data = plasma)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -0.9298  -0.5399  -0.4382  -0.3356   2.4794
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.8451     2.7703  -2.471   0.0135 *
## fibrinogen    1.8271     0.9009   2.028   0.0425 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.885  on 31  degrees of freedom
## Residual deviance: 24.840  on 30  degrees of freedom
## AIC: 28.84
##
## Number of Fisher Scoring iterations: 5
```

Vemos que el coeficiente de regresión para `fibrinogen` es significativo al nivel de 5 %. Un aumento de una unidad en esta variable aumenta log-odds en favor de un valor ESR mayor que 20 por un $1,83$ estimado con 95 % intervalo de confianza.

```
confint(plasma_glm_1,parm="fibrinogen")
```

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

```
##     2.5 %    97.5 %
## 0.3387619 3.9984921
```

Estos valores son más útiles si se convierten a los valores correspondientes para las probabilidades ellos mismos exponenciando la estimación.

```
exp(coef(plasma_glm_1)["fibrinogen"])
```

```
## fibrinogen
##   6.215715
```

y el intervalo de confianza

```
exp(confint(plasma_glm_1, parm = "fibrinogen"))
```

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

```
##     2.5 %    97.5 %
##  1.403209 54.515884
```

El intervalo de confianza es muy amplio porque hay pocas observaciones en general y muy pocos donde el valor ESR es mayor de 20. Sin embargo, parece probable que el aumento de los valores de fibrinógeno conducir a una mayor probabilidad de un valor ESR mayor de 20. Ahora podemos caber un modelo de regresión logística que incluye ambas variables explicativas utilizando el código.

```
plasma_glm_2 <- glm(ESR ~ fibrinogen + globulin, data = plasma,family = binomial())
summary(plasma_glm_2)
```

```
##
## Call:
## glm(formula = ESR ~ fibrinogen + globulin, family = binomial(),
##     data = plasma)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9683  -0.6122  -0.3458  -0.2116   2.2636
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.7921     5.7963  -2.207   0.0273 *
## fibrinogen    1.9104     0.9710   1.967   0.0491 *
## globulin      0.1558     0.1195   1.303   0.1925
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 30.885  on 31  degrees of freedom
## Residual deviance: 22.971  on 29  degrees of freedom
## AIC: 28.971
##
## Number of Fisher Scoring iterations: 5
```

El coeficiente de gamma globulina no es significativamente diferente de cero.

Ambos modelos anidados se pueden comparar utilizando una prueba de razón de verosimilitud con la función `anova`

```
anova(plasma_glm_1, plasma_glm_2, test = "Chisq")
```

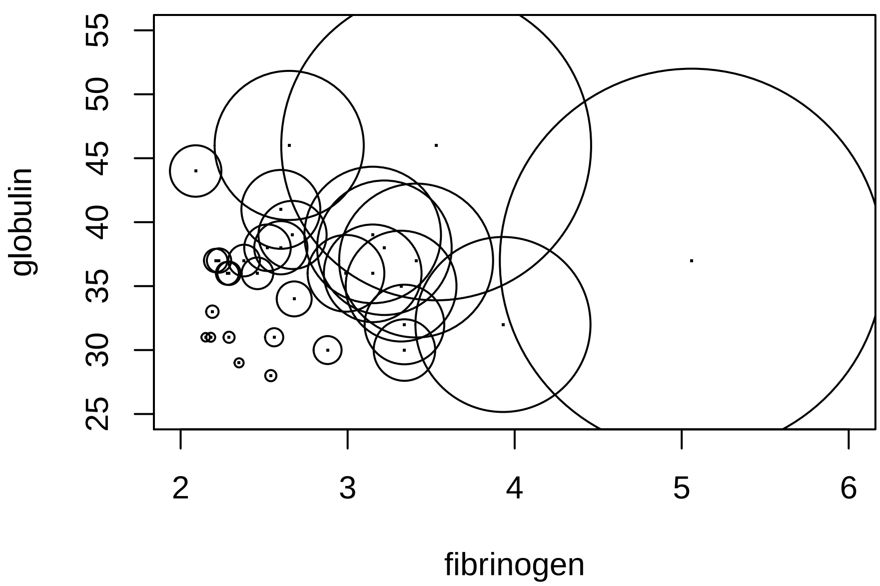
```
## Analysis of Deviance Table
```

```
##
## Model 1: ESR ~ fibrinogen
## Model 2: ESR ~ fibrinogen + globulin
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        30     24.840
## 2        29     22.971  1   1.8692   0.1716
```

Por lo tanto, concluimos que la gamma globulina no está asociada con el nivel de ESR.

La gráfica muestra claramente la probabilidad creciente de un valor de `ESR` por encima de 20 (círculos más grandes) como los valores de `fibrinogen` ya una en menor medida, un aumento de la gamma globulina.

```
prob <- predict(plasma_glm_2,type="response")
plot(globulin ~ fibrinogen, data = plasma, xlim = c(2, 6),ylim = c(25, 55), pch = ".")
symbols(plasma$fibrinogen, plasma$globulin, circles = prob,add = TRUE)
```



1.2. Casos de Estudio con la Regresión Logística

Ejemplo:

Supongamos el fichero de datos `adult.csv` disponible aquí

Vamos a tratar de predecir la variable respuesta `ABOVE50k` (sueldo >50k) a través de una regresión logistica en base a variables explicativas demográficas.

```
inputData <- read.csv("http://idaejin.github.io/bcam-courses/R/datahack/Modulo1/data/adult.c
```

1.3. Sesgo de clase

Idealmente, la proporción de eventos y no eventos en la variable Y debe ser aproximadamente la misma. Por lo tanto, primero vamos a comprobar la proporción de clases en la variable dependiente `ABOVE50K`.

Claramente, hay un *sesgo de clase*, una condición observada cuando la proporción de eventos es mucho menor que la proporción de no-eventos. Por lo tanto, debemos muestrear las observaciones en proporciones aproximadamente iguales para obtener mejores modelos.

```
table(inputData$ABOVE50K)
```

```
##
##     0     1
## 24720  7841
```

1.4. Crear muestra de entrenamiento y de validación (o test)

Una forma de abordar el problema del sesgo de clase es dibujar los 0 y 1 para el trainingData (muestra de desarrollo) en proporciones iguales. Al hacerlo, pondremos el resto del inputData no incluido en el entrenamiento en testData (muestra de validación). Como resultado, el tamaño de la muestra de desarrollo será menor que la validación, lo que está bien, porque, hay un gran número de observaciones (> 10K).

```
logitMod <- glm(ABOVE50K ~ RELATIONSHIP + AGE + CAPITALGAIN
                + OCCUPATION + EDUCATIONNUM, data=trainingData,
                family=binomial(link="logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
predicted <- plogis(predict(logitMod, testData))  # predicted scores or
# or
# predicted <- predict(logitMod, testData,type="response")  # predicted scores
```

Cuando usamos la función `predict()` en este modelo, se predice el `log(odds)` de la variable Y ($log(\frac{1}{1-p})$). Esto no es lo que queremos en última instancia porque, los valores predichos pueden no estar dentro del rango 0 y 1 como se esperaba. Por lo tanto, para convertirlo en puntajes de probabilidad de predicción que están enlazados entre 0 y 1, usamos el `plogis()` o *type="response"* como argumento de `predict()`.

1.5. Decidir el límite óptimo de probabilidad de predicción para el modelo

La puntuación de probabilidad de predicción de corte por defecto es `0,5` o la proporción de 1 y 0 en los datos de entrenamiento. Pero a veces, afinar el límite de probabilidad puede mejorar la precisión tanto en el desarrollo como en las muestras de validación. La función del paquete `InformationValue` llamada `optimalCutoff` proporciona maneras de encontrar el punto de corte óptimo para mejorar la predicción de 1's, 0's, tanto 1 como 0's y o reducir el error de clasificación errónea.

```
library(InformationValue)
optCutOff <- optimalCutoff(testData$ABOVE50K, predicted)[1]
optCutOff
```

```
## [1] 0.89
```

1.6. Error de clasificación errónea

El error de clasificación errónea es el porcentaje de desajuste de los valores reales predichos, independientemente de los 1 o los 0. Cuanto menor sea el error de clasificación errónea, mejor será su modelo.

```
misClassError(testData$ABOVE50K, predicted, threshold = optCutOff)
```

```
## [1] 0.0892
```

1.7. Curva ROC

La curva ROC (Receiver Operating Characteristics) es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación.

Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

Proporciona herramientas para seleccionar los modelos posiblemente óptimos y descartar modelos subóptimos independientemente de (y antes de especificar) el coste de la distribución de las dos clases sobre las que se decide.

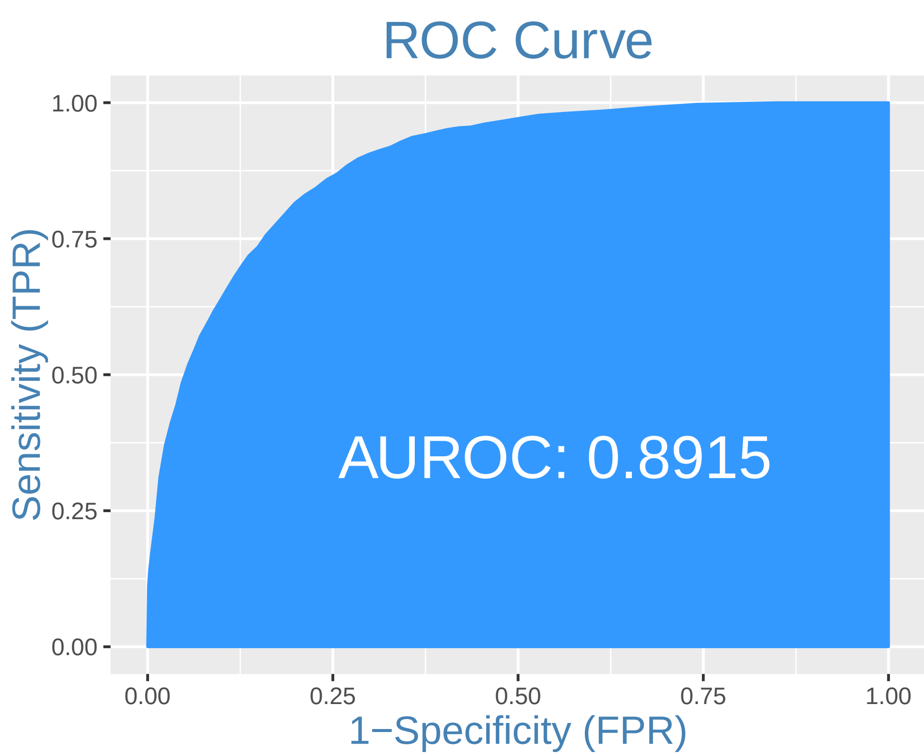
- Verdaderos Positivos (VP) o también éxitos
- Verdaderos Negativos (VN) o también rechazos correctos
- Falsos Positivos (FP) o también falsas alarmas o Error tipo I
- Falsos Negativos (FN) o también, Error de tipo II
- Sensibilidad o Razón de Verdaderos Positivos (VPR) o también razón de éxitos y, recuerdo en recuperación de información,

$$VPR = VP/P = VP/(VP + FN)$$

- Especificidad o Razón de Verdaderos Negativos

$$\text{ESPECIFICIDAD} = VN/N = VN/(FP + VN) = 1 - FPR$$

```
plotROC(testData$ABOVE50K, predicted)
```



1.7.1. German credit Data

```
data <- read.table("http://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german/ger

colnames(data)<-c("account.status","months",
                "credit.history","purpose","credit.amount",
                "savings","employment","installment.rate","personal.status",
                "guarantors","residence","property","age","other.installments",
                "housing","credit.cards","job","dependents","phone","foreign.worker","credit

head(data)
```

```
##   account.status months credit.history purpose credit.amount savings
## 1            A11      6            A34     A43          1169     A65
## 2            A12     48            A32     A43          5951     A61
## 3            A14     12            A34     A46          2096     A61
## 4            A11     42            A32     A42          7882     A61
## 5            A11     24            A33     A40          4870     A61
## 6            A14     36            A32     A46          9055     A65
##   employment installment.rate personal.status guarantors residence
## 1        A75                4             A93       A101         4
## 2        A73                2             A92       A101         2
## 3        A74                2             A93       A101         3
## 4        A74                2             A93       A103         4
## 5        A73                3             A93       A101         4
## 6        A73                2             A93       A101         4
##   property age other.installments housing credit.cards  job dependents
## 1     A121  67               A143    A152            2 A173          1
## 2     A121  22               A143    A152            1 A173          1
## 3     A121  49               A143    A152            1 A172          2
## 4     A122  45               A143    A153            1 A173          2
## 5     A124  53               A143    A153            2 A173          2
## 6     A124  35               A143    A153            1 A172          2
##   phone foreign.worker credit.rating
## 1  A192           A201             1
## 2  A191           A201             2
## 3  A191           A201             1
## 4  A191           A201             1
## 5  A191           A201             2
## 6  A192           A201             1
```

```
levels(data$account.status) <- c("<0DM","<200DM",">200DM","NoStatus")
levels(data$credit.history) <- c("No","Allpaid","Allpaidtillnow","Delayinpaying","Critical")
```

```
levels(data$purpose) <- c("car(new)","car(used)","furniture/equipment","radio/television","d
"repairs","education","vacation-doesnotexist?","retraining","business","others")
```

Vamos a dividir el conjunto de datos en 0.7: 0.3 para el entrenamiento y la prueba del modelo. Para la regresión logística, también necesitamos transformar el marco de datos con factores en la matriz con valor biométrico.

```
mat1 <- model.matrix(credit.rating ~ . , data = data)
n<- dim(data)[1]

set.seed(1234)
train<- sample(1:n , 0.7*n)
xtrain<- mat1[train,]
xtest<- mat1[-train,]

ytrain<- data$credit.rating[train]
ytrain <- as.factor(ytrain-1) # convert to 0/1 factor
 ytest<- data$credit.rating[-train]
ytest  <- as.factor(ytest-1) #  convert to 0/1 factor
```


Build the logistic Regression model

```
m1 <- glm(credit.rating ~ . , family = binomial, data= data.frame(credit.rating= ytrain, xtr
```

Key Variables for the regression model.

```
sig.var<- summary(m1)$coeff[-1,4] <0.01
names(sig.var)[sig.var == T]
```

```
## [1] "account.statusNoStatus"    "months"
## [3] "credit.historyCritical"    "purposecar.used."
## [5] "purposeradio.television"   "purposedomesticappliances"
## [7] "savingsA64"                "savingsA65"
## [9] "installment.rate"
```

Predict outcome with Logistic Regression model, then use the test dataset to evaluate the model.

```
pred1<- predict.glm(m1,newdata = data.frame(ytest,xtest), type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```r
result1<- table(ytest, floor(pred1+1.5))
result1
```

```
##
## ytest   1   2
##     0 176  25
##     1  51  48
```

```r
error1<- sum(result1[1,2], result1[2,1])/sum(result1)
error1
```

```
## [1] 0.2533333
```

Curva ROC con el paquete `ROCR`

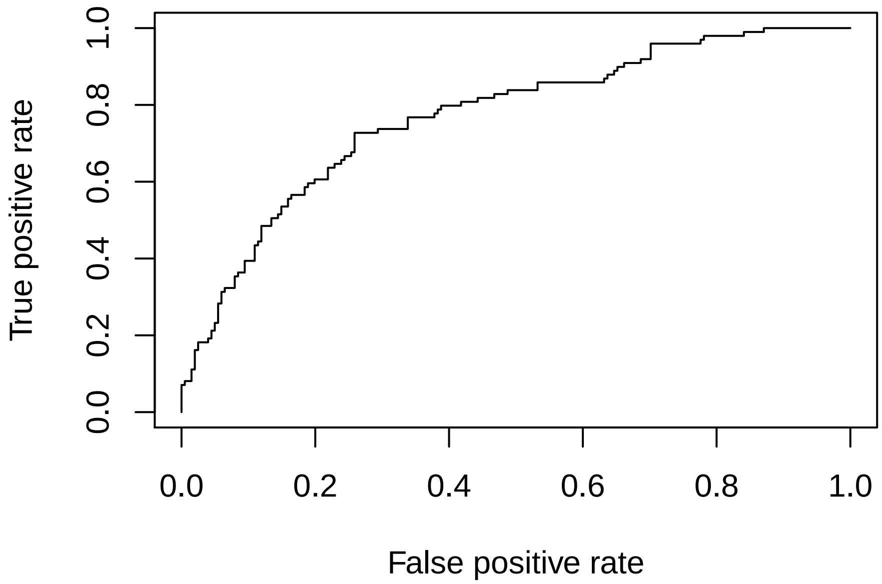
```r
library(ROCR)
```

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

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

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

```r
pred = prediction(pred1,ytest)
perf <- performance(pred, "tpr", "fpr")
plot(perf)
```




```
AUCLog1=performance(pred, measure = "auc")@y.values[[1]]
cat("AUC: ",AUCLog1,"n")
```

```
## AUC:  0.7699382 n
```

2. Regresión de Poisson

En la regresión de Poisson la variable respuesta/resultado Y es un conteo. Pero también podemos tener Y/t, la tasa (o incidencia) como la variable de respuesta, donde t es un intervalo que representa el tiempo, el espacio o algún otro agrupamiento.

Variables explicativas:

- Las variables explicativas, $X = (X_1, X_2, ...X_k)$, pueden ser continuas o una combinación de variables continuas y categóricas.
- Las variables explicativas, $X = (X_1, X_2, ...X_k)$, pueden ser TODAS categóricas. Entonces los conteos se representan en una tabla de contingencia, y la convención es llamar a tal modelo modelo log-lineal.
- Si Y/t es la variable de interés, entonces, incluso con todos los predictores categóricos, el modelo de regresión será conocido como regresión de Poisson, no un modelo log-lineal.

En la regresión de Poisson, los datos siguen una distribución de {Poisson} distribution, i.e. $y \sim \mathcal{P}\text{ois}(\mu)$. En la regresión se utiliza la transformación del logaritmo

$$\log(\mu) = \beta_0 + \beta_1 x_1$$

Por tanto, en teoría $\mathbb{E}[y] = \mathbb{V}\text{ar}[y] = \mu$

Por simplicidad, con una sola variable explicativa, escribimos: $\log(\mu) = \alpha + \beta_x$, que es equivalente a $\mu = exp(\alpha + \beta_x) = exp(\alpha)exp(\beta_x)$.

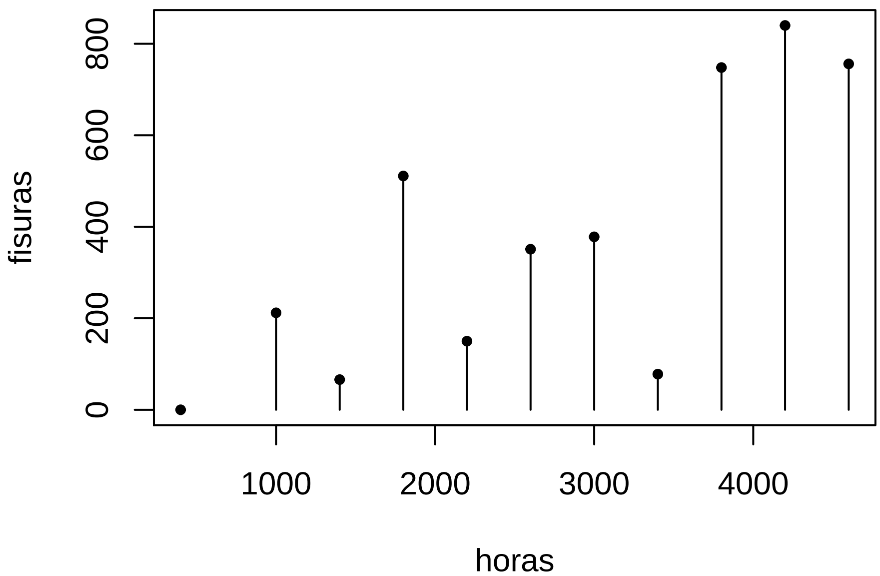
Interpretación de las estimaciones de parámetros:

- $exp(\alpha) =$ es el efecto sobre la media de Y, es decir μ, cuando $X = 0$
- $exp(\beta) =$ para cada unidad adicional de X, la variable predictora tiene un efecto multiplicativo de $exp(\beta)$ sobre la media de Y, esto es μ
  - Si $\beta = 0$, entonces $exp(\beta) = 1$, y el conteo esperado, $\mu = E(y) = exp(\alpha)$, por tanto Y y X no están relacionadas.
  - Si $\beta > 0$, entonces $exp(\beta) > 1$, y el conteo esperado $\mu = E(y)$ es $exp(\beta)$ veces más grande cuando $X = 0$
  - Si $\beta < 0$, entonces $exp(\beta) < 1$, y el conteo esperado $\mu = E(y)$ es $exp(\beta)$ veces más pequeño cuando $X = 0$

Ejemplo

horas	fisuras
400	0
1000	212
1400	66
1800	511
2200	150
2600	351
3000	378
3400	78
3800	748
4200	840
4600	756

```
turbinas <- read.table("https://idaejin.github.io/bcam-courses/R/datahack/Modulo1/data/fisur
plot(turbinas,type="h"); points(turbinas,cex=.6,pch=19)
```

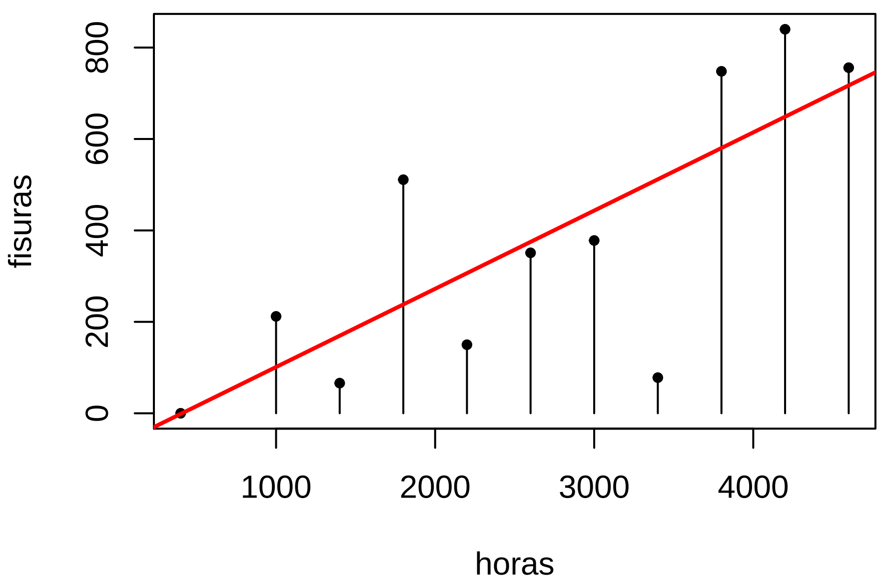



```
ex2<- lm(fisuras~horas,data=turbinas)
```

valores ajustados

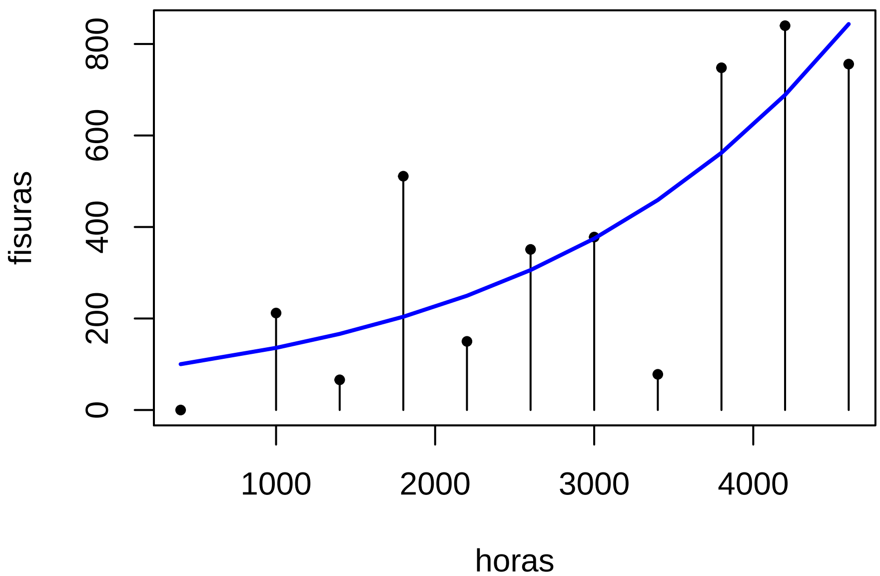
```
ex2$fitted
```

```
plot(turbinas,type="h"); points(turbinas,cex=.6,pch=19)
abline(ex2,col=2,lwd=2)
```



`family="poisson"`

```
ex3 <- glm(fisuras ~ horas,data=turbinas, family = "poisson")
plot(turbinas,type="h"); points(turbinas,cex=.6,pch=19)
lines(turbinas$horas,fitted(ex3,type="response"),col=4,lwd=2)
```



2.1. Regresión de Poisson para tasas

Logaritmo de la tasa: $\log(Y/t)$

El modelo de regresión para la tasa esperada de ocurrencia es:

$$\log(\mu/t) = \alpha + \beta x$$

que se puede reescribir como

$$\begin{aligned} \log(\mu) - \log(t) &= \alpha + \beta x \\ \log(\mu) &= \alpha + \beta x + \log(t) \end{aligned}$$

El termino $\log(t)$ se denomina *offset*. Es un término de ajuste y un grupo de observaciones puede tener el mismo `offset` o cada individuo puede tener un valor diferente de t. $\log(t)$ es una observación y cambiará el valor de los recuentos estimados:

$$\mu = exp(\alpha + \beta x + \log(t)) = (t)exp(\alpha)exp(\beta_x)$$

Esto significa que el conte medio es proporcional a t.

Ejemplo:

Los datos `credit.data` son una muestra de sujetos seleccionados aleatoriamente para un estudio italiano sobre la relación entre ingresos y si se posee una tarjeta de crédito de viaje (como American Express o Diner's Club). En cada nivel de ingresos anuales en millones de liras (la moneda en Italia antes del euro), el cuadro indica el número de sujetos que se tomaron muestras y el número de estos sujetos que poseen al menos una tarjeta de crédito de viaje.

Este ejemplo tiene información sobre los individuos agrupados por sus ingresos, el número de individuos (casos) dentro de ese grupo de ingresos y el número de tarjetas de crédito.

```
credit.card <- read.table("https://idaejin.github.io/bcam-courses/R/datahack/Modulo1/data/cr
names(credit.card) <- c("Income","NumberCases","CreditCards")

# creamos la variable lcases como offset
lcases <- log(credit.card[,2])

data <- cbind(credit.card,lcases)
```


`offset` sirve para normalizar los valores de la celda celda ajustada por algún espacio, agrupación o intervalo de tiempo con el fin de modelar las tasas.

variable serves to normalize the fitted cell means per some space, grouping or time interval in order to model the rates

```
poisson.mod <- glm(CreditCards~Income+offset(lcases),family=poisson, data)
summary(poisson.mod)
```

```
##
## Call:
## glm(formula = CreditCards ~ Income + offset(lcases), family = poisson,
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6907  -0.9329  -0.5675   0.2186   2.1681
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.386586   0.399655  -5.972 2.35e-09 ***
```

```
## Income       0.020758   0.005165   4.019 5.84e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 42.078  on 30  degrees of freedom
## Residual deviance: 28.465  on 29  degrees of freedom
## AIC: 67.604
##
## Number of Fisher Scoring iterations: 5
```

El modelo es:

$$log(\mu/t) = -2{,}3866 + 0{,}0208 Ö Income$$

donde $log(t) =$ casos.

¿Cuál es la tasa promedio estimada de incidencia, es decir, el uso de tarjetas de crédito dado el ingreso?

También podemos obtener el número predicho/ajustado/ esperado de tarjetas de crédito basado en el modelo ajustado.

```r
fitted(poisson.mod)
```

```
##         1         2         3         4         5         6         7
## 0.1513140 0.1610364 0.8220707 0.5035881 1.5424521 0.8748915 1.4291875
##         8         9        10        11        12        13        14
## 0.1823956 1.3035488 0.1901272 0.6070306 0.4131753 1.0546039 0.4306896
##        15        16        17        18        19        20        21
## 0.4397232 0.2339885 0.2490230 0.2542462 2.5957899 0.2705824 0.3128994
##        22        23        24        25        26        27        28
## 1.5973118 2.1264053 1.1315170 1.9658025 0.4739200 0.4838604 0.5257510
##        29        30        31
## 0.6470388 6.6599658 1.3660636
```

Así, en el grupo de seis personas que ganan unos 65 millones de liras, el número esperado en el grupo con al menos una tarjeta de crédito de viaje es 2.126, mientras que el número observado es de 6.

```r
predict(poisson.mod,data.frame(lcases=log(6),Income=65),type="response")
```

```
##        1
## 2.126405
```

$\hat{\mu} = 2{,}1264053$.

Notese, que `lcases = log(t) = log(6)` para este caso específico. La tasa esperada sería

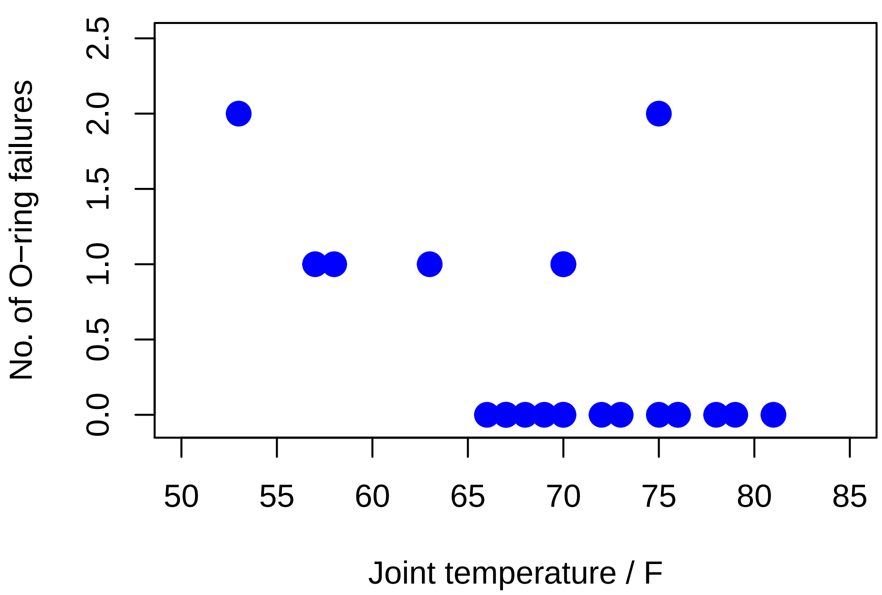
$$\frac{\hat{\mu}}{t} \approx 0{,}3544$$

Desastre del transbordador Challenger

El 7 de Enero de 1986, la noche antes del despegue del transbordador, hubo una reunión en la que se discutio sobre la la temperatura mínima predicha para del día siguiente, 31F, y el efecto de la misma sobre el sello en las juntas de los O-rings. En la discusión utilizaron el siguiente gráfico en el que se muestra la relación entre la temperatura y el número de O-rings que sufrian problemas

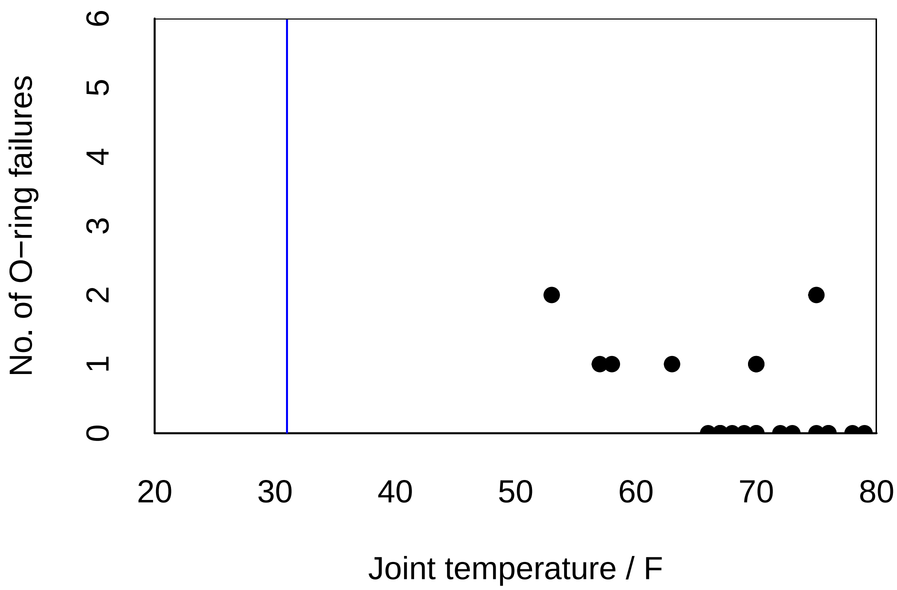
```
challenger<-read.table("http://idaejin.github.io/bcam-courses/R/datahack/Modulo1/data/challe
m<-challenger[,1]
r<-challenger[,2]
temperature<-challenger[,3]

### plot all data
plot(temperature, r,pch=19,cex=1.65,xlim=c(50,85),ylim=c(-0.05,2.5),
     xlab='Joint temperature / F', ylab='No. of O-ring failures',col="blue")
```




```
par(xaxs="i", yaxs="i")
x.at <- seq(20, 80, by=5)
```

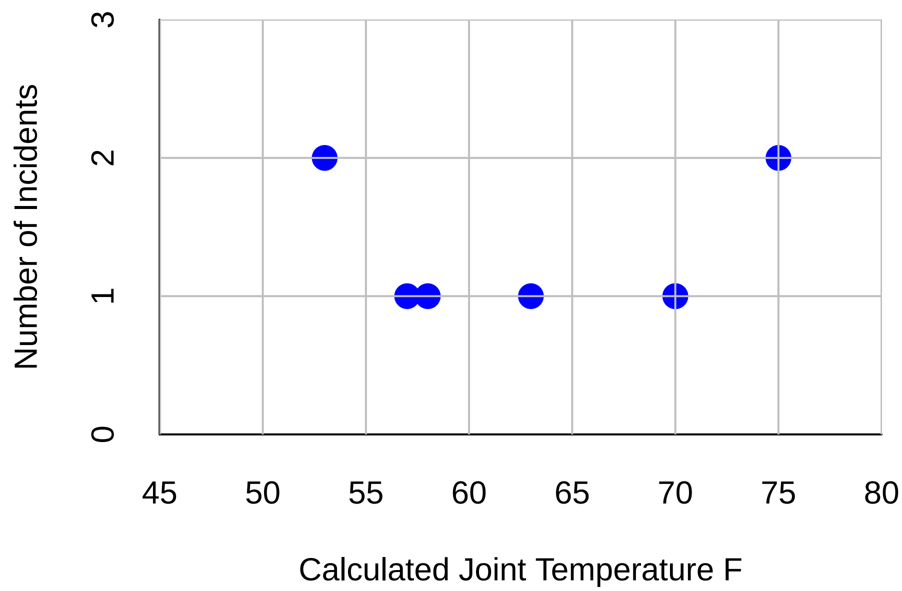
```
y.at <- seq(0, 6, by=1)
plot(temperature, r, xlim=range(x.at), ylim=range(y.at),
     xlab='Joint temperature / F', ylab='No. of O-ring failures',
     pch= 19,type="p",axes=FALSE)
axis(1, at= x.at, lwd.ticks = 0)
axis(2, at= y.at, lwd.ticks = 0)
abline(v=31, col='blue')
abline(h=6, col="black")
abline(v=80, col="black")
```



El primer error que cometieron, fue el no dibujar los casos en los que no había incidentes, para saber cuál eran las temperaturas más propicias. La decisión final fue permitir el despegue en el que murieron 7 astronautas debido a la combustión de gas a través de un O-ring. ng.

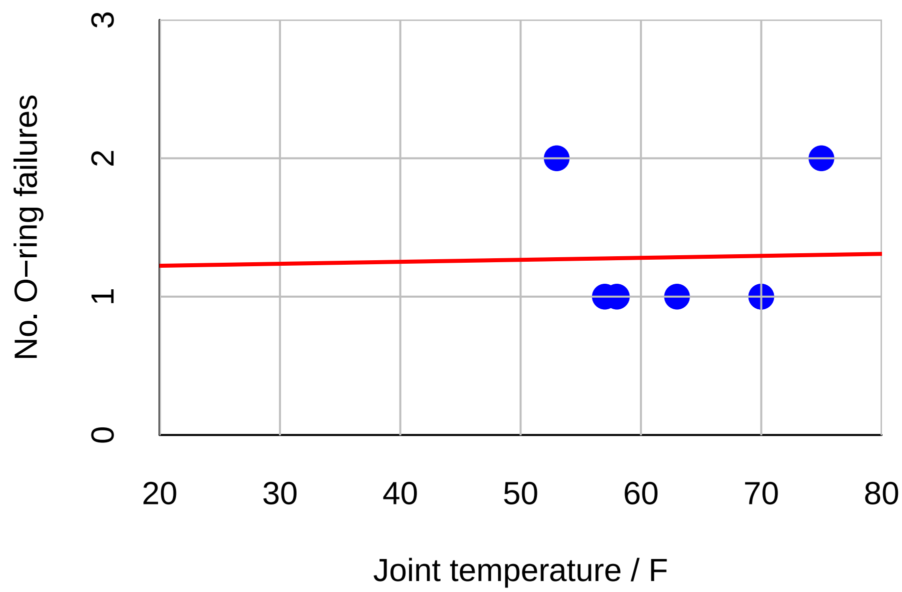
```
### original misleading plot!
par(xaxs="i", yaxs="i")
x.at <- seq(45, 80, by=5)
y.at <- seq(0, 3, by=1)
failures <- which(r!=0) ## selects failure cases
plot(temperature[failures], r[failures],xlim = range(x.at), ylim= range(y.at),
     pch= 19,type= "p", xlab='Calculated Joint Temperature F',
     ylab='Number of Incidents', axes=FALSE,col="blue",cex=1.65)
axis(1, at= x.at, lwd.ticks = -1)
axis(2, at= y.at, lwd.ticks = -1)
grid(ny=3, col="grey", lty="solid");
```

```
abline(h=3, col="grey")
```



Regresión lineal?

```
### plot only launches with at least one failure and their fitted curve
failures <- which(r!=0)
par(xaxs="i", yaxs="i")
x.at <- seq(20, 80, by=5)
y.at <- seq(0, 3, by=1)
plot(temperature[failures], r[failures], xlim=range(x.at), ylim=range(y.at),
     xlab='Joint temperature / F', ylab='No. O-ring failures',pch= 19,
     type= "p", axes = F, cex=1.65,col="blue")
axis(1, at= x.at, lwd.ticks = 0)
axis(2, at= y.at, lwd.ticks = 0)
grid(ny=3, col="grey", lty="solid");
abline(h=3, col="grey")
testtemp <- seq(10,100,1)
fit.lm <- lm(r ~ temperature, data=challenger, subset=which(r!=0))
lines(testtemp, predict(fit.lm, data.frame(temperature=testtemp)),
      col="red",lwd=2)
```




```
### plot all data
par(xaxs="i", yaxs="i")
x.at <- seq(20, 80, by=5)
y.at <- seq(0, 10, by=1)
plot(temperature, r, main = "Scatterplot of all Data", xlim=range(x.at),
     ylim=range(-0.1,y.at), xlab='Joint temperature / F',
     ylab='No. of O-ring failures',cex=1.65, col="blue",
     pch= 19,type= "p",axes=FALSE)
axis(1, at= x.at, lwd.ticks = 0)
axis(2, at= y.at, lwd.ticks = 0)
abline(v=31, col='red',lwd=3)
abline(h=10, col="black")
abline(v=80, col="black")
legend("topright", inset=.05,c("Fitted Curve","Temp at 31 F"),
       fill=c("orange", "red"))
### Fit and plot a curve using all data
fit.glm <- glm(cbind(r,m-r) ~ temperature, data=challenger, family=binomial)
# predict probability of failure of a single O-ring joint at the following temperature
testtemp <- seq(10,100,1)
pred.glm <- predict(fit.glm, data.frame(temperature=testtemp),
                    type="response",se.fit = TRUE )
lines(testtemp, 6*pred.glm$fit, col="orange",lwd=3)
lines(testtemp, 6*pred.glm$fit-1.96*pred.glm$fit, col="orange",lwd=3,lty=3)
lines(testtemp, 6*pred.glm$fit+1.96*pred.glm$fit, col="orange",lwd=3,lty=3)
points(temperature, r, main = "Scatterplot of all Data", xlim=range(x.at),
       ylim=range(-0.1,y.at), xlab='Joint temperature/F',
```

```
        ylab='No. of O-ring failures',cex=1.65, col="blue",pch= 19,type= "p")
```