TEMA 3

MÓDULO:
TÉCNICAS AVANZADAS DE PREDICCIÓN

MODELO LINEAL GENERAL

Institut de Formació Contínua-IL3
UNIVERSITAT DE BARCELONA

ÍNDICE

Objetivos Específicos

1. Modelo Lineal General (MLG o GLM)

Actividad: Seguro de Auto

1.1 Modelo de Regresión Lineal vs GLM

1.2 Cambiar la función de distribución. La familia exponencial

1.3 Cambiar la función link

1.4 Calibración el modelo

1.5 Vuelta a nuestro problema

1.6 Modelos Logísticos

Actividad: Sensores de Movimiento

1.7 Estrategia ante la modelización GLM

Ideas clave

OBJETIVOS ESPECÍFICOS

- Aprender la diferencia entre los modelos lineales y los GLM.
- Distinguir los diferentes tipos de modelos dentro de los GLM.
- Conocer qué metodología de resolución tienen este tipo de problemas.

1. MODELO LINEAL GENERAL (MLG O GLM)

Este tema vamos a empezarlo, directamente, presentando una nueva base de datos. Aunque no nos vamos a olvidar del super online puesto que también la veremos.

ACTIVIDAD

SEGURO DE AUTO

Contamos con los datos de una empresa de seguros. Esta empresa quiere establecer una nueva tarifa para su producto de cobertura de lunas de coche. Para ello, quiere ofrecer un producto adecuado al nivel de riesgo de cada cliente. Contamos con una base de datos del número de siniestros sufridos por cada cliente durante el año pasado:

- **Garantía:** tres niveles de garantía:
  1. Menor Cobertura.
  2. Cobertura Premium.
  3. Cobertura Gold.
- **Edad Carnet:** número de años desde que se sacó el carnet de conducir el tomador del seguro.
- **Edad coche:** número de años desde que el coche fue fabricado.
- **Sex:** indica si es Varón o Mujer.
- **Edad:** edad del tomador de la póliza.
- **x:** longitud del domicilio del cliente.
- **y:** latitud del domicilio del cliente.
- **Response:** número de siniestros que tuvo el cliente el año pasado.
- **Expuesto:** porcentaje del año en el que este cliente ha estado expuesto.

Objetivo: el objetivo que tiene la empresa es el de modelizar la variable response (frecuencia de siniestros) con las diferentes variables explicativas que contamos. Con esta modelización, podrá definir la tarifa con la que salir al mercado. Para ello, sabemos que el coste medio de cada siniestro es de 300 euros y, además, que los gastos de marketing y ventas del seguro representan un 20% de la prima:

$$Prima = \frac{Siniestros * CosteSiniestro}{1 - Gastos} = \frac{ModeloSiniestros * 300}{1 - 0.20}$$

¿Generan todos los clientes el mismo número de siniestros?
¿Podemos discriminar perfiles y ofrecerles una prima adaptado a su riesgo?

```{r results='asis', size="small"}
df<-read.csv("./Data/table_5.03.csv",sep=",")[,-1]
pander(head(df), split.cell = 60, split.table = Inf,digits=2)
```

garantia	edad_carnet	edad_coche	sex	edad	x	y	siniestros_ly	response	Expuesto
1	26	10	1	45	-2.7	43	2	0	0.82
3	11	21	2	59	-2.7	43	1	8	0.26
1	50	17	2	73	-2.7	43	3	1	0.85
2	37	3	2	61	-2.7	43	1	1	0.68
2	10	12	2	29	-2.7	43	1	5	0.72
2	16	15	1	38	-2.7	43	3	10	0.77

¿Estamos ante un problema de regresión lineal?

1.1 MODELO DE REGRESIÓN LINEAL VS GLM

Recordemos que en un modelo de regresión lineal teníamos un conjunto de **variables explicativas** (x) y una **variable explicada** (y).

RECUERDA

Estábamos asumiendo que:

- El error (e) de la dependencia lineal entre **y** y **x** sigue una distribución normal.
- Por lo tanto, como consecuencia de lo anterior y / x sigue una distribución normal.
- El error (e) y las variables explicativas (x) son independientes.
- Asumimos que la varianza del error es homogénea.

Matemáticamente:

$$Para: Y = X\beta + e$$

$$e \sim N(0, \sigma^2)$$

$$Entonces: Y|X \sim N(X\beta, \sigma^2)$$

¿No hay ninguna alternativa si alguna de estas hipótesis no se cumple o queremos cambiarlas?

Justo para esto surge la generalización de la regresión lineal, en la que se relajan algunas hipótesis iniciales.

En nuestra base de datos, estamos intentando modelizar el número de siniestros que cada cliente va a tener el próximo año:

- A nuestro entendimiento, no sigue una distribución normal. Sino que parece más una distribución Poisson o quasi-Poisson o Binomial Negativa o Zero-Inflated Poisson.
- Además, debemos ponerle una restricción al modelo. No tendría sentido que nuestra predicción saliese un nº negativo ¿verdad?. Entonces modelizarlo como una normal no sería la mejor opción por el rango de nuestros datos.

Entonces, debemos explorar los GLM para ver si podemos modelizar este fenómeno con ellos.

Los próximos apartados contienen mucha fórmula y requieren especial atención. Os he intentado resumir el proceso y hacerlo amigable para que podáis entenderlo, matemáticamente, para luego en la práctica ser mucho más fuertes.

¿Qué nos permiten cambiar los GLM?

- Nos permite cambiar la distribución de Y|X. Es decir, la relación media-varianza. Función de distribución ().
- Nos permite cambiar la relación de linealidad entre la media de nuestra variable respuesta y las variables explicativas. Función Link g(u).

1.2 CAMBIAR LA FUNCIÓN DE DISTRIBUCIÓN. LA FAMILIA EXPONENCIAL

Recordamos las funciones de principales que vamos a tratar:

$$Y \sim Normal(\mu, \sigma^2) \rightarrow E(Y) = \mu \rightarrow Var(Y) = \sigma^2$$

$$Y \sim Poisson(\mu) \rightarrow E(Y) = \mu \rightarrow Var(Y) = \mu$$

$$Y \sim Binomial(\mu) \rightarrow E(Y) = \mu \rightarrow Var(Y) = \mu(1-\mu)$$

$$Y \sim Gamma(a\mu, a) \rightarrow E(Y) = \mu \rightarrow Var(Y) = \mu/a$$

Es importante entender la distribución que tenemos detrás de nuestros datos para realizar un buen ajuste en nuestro modelo.

¿Hay alguna función que pueda englobar la Normal, Poisson, Gamma….I can't believe it?

Toda aquella función de densidad que pueda escribirse así, pertenece a la familia exponencial:

$$f(y, \theta, \phi,) = exp(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi))$$

Donde a(), b(), c() son funciones.

ϕ Es un parámetro de dispersión que puede ser conocido o desconocido. θ es el parámetro natural o canónico.

¿La Normal se puede expresar con la función de la familia exponencial?

$$f() = \frac{1}{(2\pi\sigma^2)^{(0.5)}} exp(-\frac{(y-\mu)^2}{2\sigma^2}) =$$
$$exp(\frac{(y\mu - \mu^2/2)}{\sigma^2} - \frac{(y^2)}{2\sigma^2} - \frac{1}{2} ln(2\pi\sigma^2))$$

$$Entonces:$$
$$\theta = \mu$$
$$b(\theta) = \theta^2$$
$$a(\phi) = \sigma^2$$
$$c(y, \phi) = -\frac{1}{2} ln(2\pi\sigma^2)$$

¿La Poisson se puede expresar con la función de la familia exponencial?

$$f() = \frac{\mu^y}{(y!)} exp(-\mu) =$$
$$exp(yln(\mu) - \mu - ln(y!))$$

$$Entonces:$$
$$\theta = ln(\mu)$$
$$b(\theta) = exp(\theta)$$
$$a(\phi) = 1$$
$$c(y, \phi) = ln(y!)$$

1.3 CAMBIAR LA FUNCIÓN LINK

La función link es la que relaciona el valor esperado de nuestra variable respuesta (Y) con el predictor lineal:

$$E(Y|X) = \mu$$

$$g(\mu) = X\beta \rightarrow \mu = g^{-1}(X\beta)$$

RECUERDA

En la regresión lineal no hay transformación. Podemos decir que la función link es la identidad:

$$g(\mu) = \mu = X\beta$$

1.4 CALIBRACIÓN EL MODELO

Tenemos la función exponencial:

$$f(y,\theta,\phi) = exp(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi))$$

¿Estimamos por Log-Verosimilitud?

$$l(y,\theta) = \sum(\frac{y\theta - b(\theta)}{a(\phi)}) + \sum c(y,\phi))$$

Y también tenemos:

$$g(\mu) = X\beta$$

Maximizamos la log-verosimilitud:

$$U_j = \frac{\partial l(\theta,y)}{\partial \beta} = \sum \frac{(y-\mu)x}{Var(Y)g'(\mu)}$$

$$Matricialmente:$$

$$U = X^t M^{-1}(Y-\mu)$$

$$Siendo:$$
$$M = diag(m_i,,,m_n)$$
$$m_i = Var(Y_i)g'(\mu_i)$$

Las ecuaciones que obtenemos al intentar resolver cada una de las betas no tienen por qué ser lineales....

Vaya formulotes estamos viendo. ¿Entonces qué hacemos?

Tenemos que resolver esto por algún método numérico. Lo podemos resolver por Newton-Raphson o por Scoring Fisher, pero en ambos métodos hay que calcular matrices Hessianas que son de alto coste computacional.

Por eso optamos por el método de **Mínimos cuadrados ponderados iterativos:**

$$\beta^r = (X^t W^{-1} X)^{-1} X^t W^{-1} Z$$

$$Donde:$$
$$W = diag(Var(Y)g'(\mu)^2)$$

$$Z = X\beta + (y-\mu)diag(g'(\mu_1,,,,\mu_n))$$

SABÍAS QUE...

En ocasiones dada la complejidad de las operaciones matemáticas tenemos que recurrir a los denominados métodos numéricos que son iteraciones de operaciones de tal forma que aproximen la solución final.

Resolvemos un problema a mano con la matemática aprendida y con la función GLM. Simplemente vamos a verificar que podemos resolver el problema de los coeficientes del modelo para una submuestra de la base de datos que estamos trabajando.

En el siguiente código os voy a mostrar simplemente el proceso que hay detrás de las funciones de "R". (Por supuesto en "R" mucho más optimizado). Para calcular los coeficientes de una determinada regresión independientemente de la familia que pongamos (Poisson, Normal, Binomial), en todos los casos "R" lo resuelve con un proceso iterativo en el que va ajustando los estimadores. En este caso os he puesto 300 ejecuciones para asegurarme que llego a unas betas correctas. "R" lo que hace es que para el proceso iterativo cuando en una iteración nueva el valor de las betas nuevo conseguido no difiere del anterior.

```
```{r }
df_tratada<-dplyr::select(df,response,edad_carnet,edad_coche)[1:300,]

X<-as.matrix(dplyr::select(df_tratada,-response))
Y<-as.matrix(df_tratada$response)

X <- as.matrix(cbind(1,X))
beta<-as.matrix(c(1,0,0))

for(i in 1:300) {
Z <- as.matrix(X%*%beta+((Y-exp(X%*%beta))/exp(X%*%beta)))
W <- diag(as.numeric(exp(X%*%beta)))

beta <- solve(t(X)%*%W%*%X)%*%t(X)%*%W%*%Z
}

#Coeficientes GLM
glm(response~edad_carnet+edad_coche,data=df_tratada,family=poisson(link="log"))$coefficients
#Coeficientes Algebra
print(as.numeric(t(beta)))


```
```

```
(Intercept) edad_carnet  edad_coche
 1.14043333 -0.01573847  0.05309291
[1]  1.14043333 -0.01573847  0.05309291
```

Podemos ver que las betas que hemos conseguido en ambos procesos son idénticas. Evidentemente de cara a la producción de cualquier modelo utilizaremos las funciones desarrolladas en "R" las cuales están optimizadas y validadas. Conocer el funcionamiento de ellas es obvio que es fundamental para total conocimiento de estas herramientas estadísticas.

1.5 VUELTA A NUESTRO PROBLEMA

¿Qué distribución utilizo?

Existe una distribución que sea mejor que todas las demás. En la práctica tenemos que tirar del conocimiento que tenemos sobre los datos para poder elegirla:

- Si tenemos datos continuos y sin límites, podemos utilizar la función por defecto Gaussiana.
- Si tenemos datos continuos y no negativos, podemos pensar en distribuciones Gamma que son muy flexibles.

- Si tenemos datos binarios o categóricos, podemos estar delante de una distribución Binomial o Multinomial.

- Y si tenemos datos discretos basados en conteos, podemos estar hablando de una Poisson, Quasi-Poisson o Binomial Negativa.

¿Qué función Link elijo?

Una que tenga sentido para los datos que estamos trabajando. Las funciones para elegir son estas:

Table 1: Function $\Psi(\mathbf{x}_i'\boldsymbol{\beta})$ of Generalized Linear Model

Family	Link	Mean Function	$\Psi(\mathbf{x}_i'\boldsymbol{\beta})$
gaussian	identity	$\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$	$1/\sigma^2$
binomial	logit	$\mu_i = \frac{\exp(\mathbf{x}_i'\boldsymbol{\beta})}{1+\exp(\mathbf{x}_i'\boldsymbol{\beta})}$	$\mu_i(1-\mu_i)$
binomial	probit	$\mu_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta})$	$\frac{\phi(\mathbf{x}_i'\boldsymbol{\beta})^2}{\Phi(\mathbf{x}_i'\boldsymbol{\beta})(1-\Phi(\mathbf{x}_i'\boldsymbol{\beta}))}$
binomial	cloglog	$\mu_i = 1-\exp(-\exp(\mathbf{x}_i'\boldsymbol{\beta}))$	$\frac{1-\mu_i}{\mu_i}[\log(1-\mu_i)]^2$
poisson	log	$\mu_i = \exp(\mathbf{x}_i'\boldsymbol{\beta})$	μ_i
poisson	identity	$\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$	$1/\mu_i$
poisson	sqrt	$\mu_i = (\mathbf{x}_i'\boldsymbol{\beta})^2$	4
gamma	inverse	$\mu_i = (\mathbf{x}_i'\boldsymbol{\beta})^{-1}$	$a\mu_i^2$
gamma	identity	$\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$	a/μ_i^2
gamma	log	$\mu_i = \exp(\mathbf{x}_i'\boldsymbol{\beta})$	a
inverse gaussian	inverse squared	$\mu_i = (\mathbf{x}_i'\boldsymbol{\beta})^{-1/2}$	$\lambda\mu_i^3/4$

Fuente www.freakonometrics.hypotheses.org

¿Cómo lo programo?

Solamente tenemos que utilizar la función GLM.

Primero vamos a especificar el modelo con todo lo que tenemos:

```
# Establecemos la formula para  más adelate modelizar.
formula<-as.formula('response~garantia+edad_carnet+edad_coche+sex+edad+siniestros_ly')
formula
```

```
response ~ garantia + edad_carnet + edad_coche + sex + edad +
    siniestros_ly
```

¿Cómo voy a tratar variables categóricas?

Las variables categóricas son aquellas que no son numéricas. Tienen un valor, pero el valor no tiene sentido estadístico ordenarlo. (Peras/Manzanas), (Hombre/Mujer), (Tipo de Coche).

En nuestro caso, la variable **Garantía** aunque sea un nº, no tiene sentido de orden. Simplemente son distintos tipos de garantía que el cliente ha elegido. Entonces, lo primero que tenemos que hacer es asegurarnos que R está tratando esa variable como categórica. Lo siguiente que va a hacer nuestro modelo es generar tantas variables como categorías tiene la variable. En nuestro caso, dado que la variable **Garantía** tiene tres niveles, el modelo va a generar tres variables: Garantía1, Garantía2, Garantía3.

Cada una de estas variables tiene observaciones de 1 o 0.

- (1) en el caso de que la observación en **Garantía** sea igual a 1,
- (0) en el caso de que la observación en **Garantía** sea igual a 2 o 3.

Y así sucesivamente registro a registro. Luego cuando generemos el modelo vamos a ver que el modelo quita una de ellas. Se queda solamente con Garantía2, Garantía3.

PIENSA UN MINUTO

¿Por qué lo hace? ¿Por qué razón no podemos meter tres variables que agregadas siempre suman 1 dentro del modelo?

Porque no puede haber combinaciones lineales perfectas en el modelo. Os acordáis que cuando veíamos la correlación de Pearson os decía que medir las correlaciones es importante. Pues es por esto:

Garantía1+Garantía2+Garantía3=Intercept (Vector de 1)

Esto sería una correlación perfecta, entonces debemos quitar una variable en el modelo. En casi todos los softwares esto se hace automáticamente.

¿Cómo tratar la variable expuesto?

¿Es lo mismo haber estado expuesto 365 días, que haber estado expuesto 30 días?. Claro que no, las probabilidades deben estar corregidas. Esto hay que decírselo a nuestro programa, en concreto lo podemos meter directamente en la especificación.

Es el llamado Offset, que quiere decir el ajuste por exposición que debemos hacer a nuestra base de datos para decirle quien ha estado mucho tiempo expuesto y por lo tanto, sería normal que tuviese más siniestros, y quien ha estado menos expuesto.

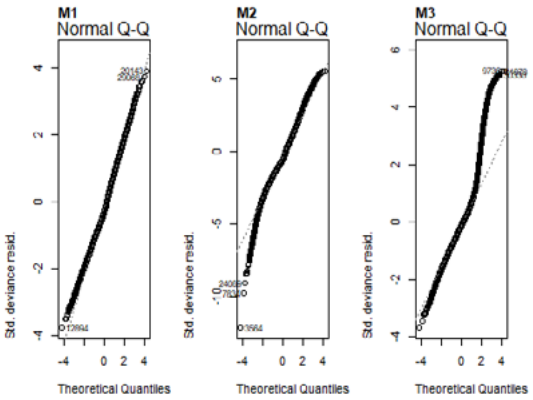
Hay que medir a todos los registros por igual

$$\mu = exp(X\beta + ln(t)) = texp(X\beta)$$

Vamos a crear tres q-q plots:

1. **M1:** El primero con el modelo perfecto.
2. **M2:** El segundo sin contemplar la exposición en la base de datos. Offset.
3. **M3:** Por último, utilizando la distribución Normal, en lugar de una Poisson.

```r
layout(matrix(c(1,2,3),1,3,byrow=T))
#Declaramos la variable categórica
df$garantia<-relevel(as.factor(df$garantia),ref=2)
#Formula Nueva
formula_new<-as.formula('response~garantia+edad_carnet+edad_coche+sex+edad+siniestros_ly+offset(log(Expuesto))')

m1<-glm(formula_new,data=df,family=poisson(link="log"))
m2<-glm(formula,data=df,family=poisson(link="log"))
m3<-glm(formula,data=df,family=gaussian)

#q-qPlit
plot(m1,which=c(2),main="M1", adj = 0)
plot(m2,which=c(2),main="M2", adj = 0)
plot(m3,which=c(2),main="M3", adj = 0)

#También lo podemos medir utilizando la función para comparar valores estimados y reales
#Hist(df,df$response,predict(m1,df,type="response"),df$edad_carnet,5,10)
#Hist(df,df$response,predict(m2,df,type="response"),df$edad_carnet,5,10)
#Hist(df,df$response,predict(m3,df,type="response"),df$edad_carnet,5,10)

```



También lo podemos medir utilizando la función para comparar valores estimados y reales:

Hist(df,df$response,predict(m1,df,type="response"),df$edad_carnet,5,10)
Hist(df,df$response,predict(m2,df,type="response"),df$edad_carnet,5,10)
Hist(df,df$response,predict(m3,df,type="response"),df$edad_carnet,5,10)

¿Alguna técnica adicional para comparar modelos?

Sí. La devianza de un modelo GLM es una medida de bondad del ajuste que compara dos modelos:

1. El modelo que tenemos programado.
2. Con el modelo saturado. Quiere decir con el modelo con todas las variables y combinaciones de variables posibles.

$$Devianza = -2[log.Mv(mimodelo) - log.Mv(modelo.saturado)]$$

Cuanta menor devianza, mejor para nuestros intereses.

Comparamos los modelos obtenidos y comprobamos que el que tiene menor devianza es el M1.

```r
m1$deviance
m2$deviance
m3$deviance
```

```
[1] 37333.47
[1] 92547.06
[1] 592645.4
```

Una vez hemos terminado el chequeo de residuos y hemos elegido nuestro modelo final en base al mejor ajuste, pasaríamos a valorar los resultados estadísticos que nuestro modelo proporcional.

Por un lado, tenemos la interpretación del modelo estadístico para ver si los resultados van en línea de nuestras hipótesis iniciales. (Ejemplo: Cuanto menor edad, mayor propensión a tener un accidente). Por otro lado, tenemos los efectos marginales de cada una de las variables. (Ejemplo: Un año adicional en la edad de una persona, repercute en x siniestros menos a lo largo de un año).

¿Interpretación del modelo y efectos marginales?

Cuando hacemos un summary() del modelo de regresión final que hemos planteado, vamos a encontrar la estimación de las betas y la varianza de las mismas. Esta información nos es útil para decidir si la variable es significativa y por lo tanto, en este entorno multivariante, susceptible de entrar en el modelo. También el signo (+ o -) de la beta nos indica la relación entre la variable explicativa y la variable respuesta.

Los efectos marginales nos dicen cuánto cambio se produce en nuestra variable respuesta estimada dado un cambio de una unidad en la variable explicativa.

```{r }
summary(m1)

poissonmfx(formula = formula_new,data=df)
```

```
Call:
glm(formula = formula_new, family = poisson(link = "log"), data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7611  -0.8508  -0.2785   0.5255   3.9038

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.5386080  0.0200287  -26.89   <2e-16 ***
garantia1     -1.4850501  0.0097883 -151.72   <2e-16 ***
garantia3      0.6886321  0.0057573  119.61   <2e-16 ***
edad_carnet   -0.0392408  0.0003967  -98.92   <2e-16 ***
edad_coche     0.0792939  0.0005597  141.66   <2e-16 ***
sex            0.6943937  0.0071272   97.43   <2e-16 ***
edad          -0.0008011  0.0003851   -2.08   0.0375 *
siniestros_ly  0.5908250  0.0020402  289.60   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 196734  on 37652  degrees of freedom
Residual deviance:  37333  on 37645  degrees of freedom
AIC: 118200

Number of Fisher Scoring iterations: 5


Number of Fisher Scoring iterations: 5

Call:
poissonmfx(formula = formula_new, data = df)

Marginal Effects:
                dF/dx  Std. Err.         z   P>|z|
garantia1     -4.9640035  0.0268498 -184.8804 < 2e-16 ***
garantia3      3.2392521  0.0322057  100.5799 < 2e-16 ***
edad_carnet   -0.1574099  0.0016122  -97.6344 < 2e-16 ***
edad_coche     0.3180785  0.0022991  138.3492 < 2e-16 ***
sex            2.3824669  0.0215540  110.5346 < 2e-16 ***
edad          -0.0032135  0.0015448   -2.0802 0.03751 *
siniestros_ly  2.3700286  0.0090474  261.9557 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

dF/dx is for discrete change for the following variables:

[1] "garantia1" "garantia3" "sex"
```

Recordamos que nuestro objetivo es el de generar una tarifa para poder comercializar. Ya tenemos nuestro modelo realizado, hemos comprobado que los residuos siguen todas las hipótesis marcadas, hemos visto la relación entre las variables.

A continuación, desarrollamos el tarificador de nuestro seguro de auto.

$$Prima = \frac{Siniestros * CosteSiniestro}{1 - Gastos} = \frac{ModeloSiniestros * 300}{1 - 0.20}$$

```r
garantia2<-1
garantia3<-0
edad_carnet<-17
edad_coche<-5
sex<-1
edad<-39
siniestros_ly<-2
Expuesto<-1

X<-cbind(1,garantia2,garantia3,edad_carnet,edad_coche,sex,edad,siniestros_ly)
beta<-m1$coefficients

Siniestralidad<-exp(as.matrix(X)%*%as.matrix(beta))
CosteSiniestro<-300
Gastos<-0.2

print("La tarifa que le correspondería sería:")
print(paste(floor(Siniestralidad*CosteSiniestro/(1-Gastos)),"Euros"))
```

```
[1] "La tarifa que le correspondería sería:"
[1] "239 Euros"
```

La tarifa que nos saldría para dicha modelización y las características de la persona que proponíamos serían 239 euros.

Podéis proponer las características que queráis para ver el precio que estaríamos ofreciendo.

1.6 MODELOS LOGÍSTICOS

¿La distribución Binomial se puede expresar con la función de la familia exponencial?

$$f(y) = exp(yln(\frac{p}{1-p}) + ln(1-p))$$

$$Donde:$$
$$Y = 0,1$$
$$p = \text{Probabilidad de que un suceso ocurra}$$

Contamos con una tercera base de datos bastante interesante y diferente a lo que hemos venido viendo a lo largo de los temas.

ACTIVIDAD

SENSORES DE MOVIMIENTO

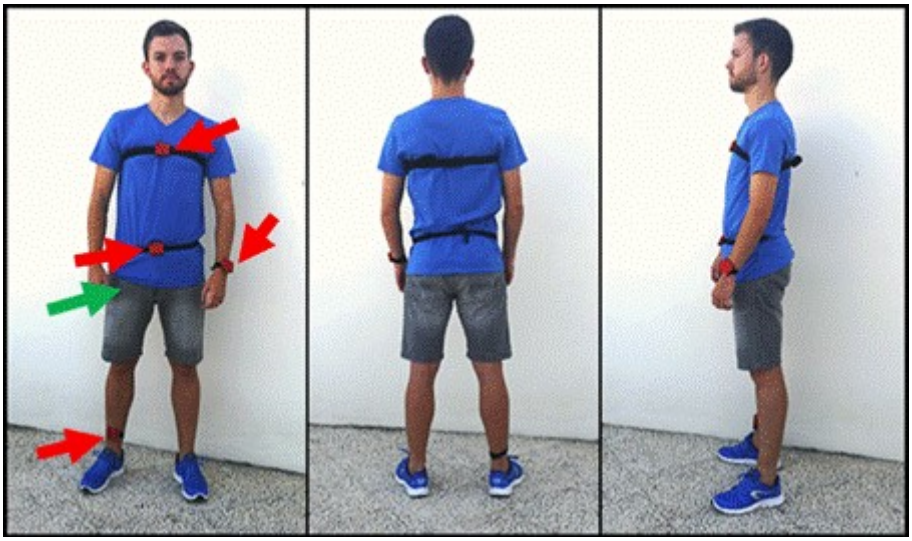
Contamos con los datos procedentes de diferentes sensores colocados en personas durante sus actividades cotidianas del día a día. Los sensores a través de un acelerómetro capturan los movimientos de las diferentes personas. Cada una de las personas van a hacer diferentes acciones que quedan reflejadas en la base de datos. Se capturan 300 milisegundos por cada una de las acciones que la persona desempeña. Para el trabajo en cuestión, he elegido los datos procedentes de 1 sensor colocado en la muñeca de la persona:

- **Nombre_Fichero:** Nos indica el tipo de movimiento que la persona ha realizado. (Caminar, Sentarse, AbrirPuerta, Caerse…)

- **X1…X303:** Nos indica el valor del acelerómetro desde el milisegundo 1 al milisegundo 303.

Objetivo: El objetivo que tiene la empresa es el de detectar cuando una persona se cae, de tal forma, que pueda generar dispositivos para personas mayores que viven solas que cuando sufran caídas automáticamente un algoritmo lo reconozca y llame automáticamente a un equipo de ayuda.

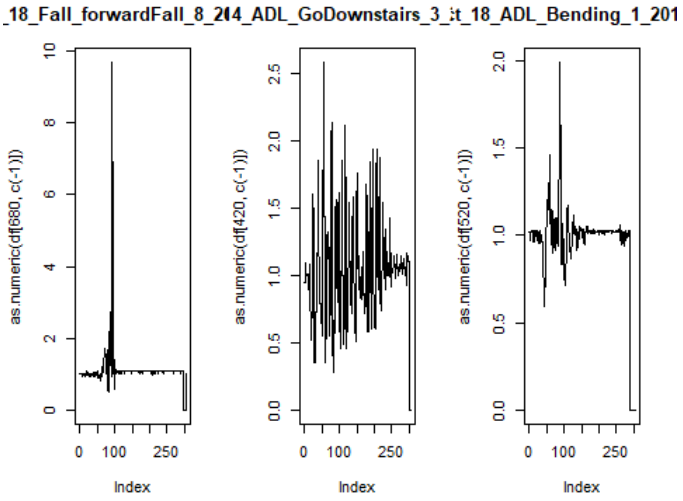
Para ello lo primero que vamos a hacer es establecer nuestra variable respuesta. Nuestra variable son aquellos registros que provengan de secuencias de caídas. Para ello las identificamos como "Fall_Forward".

```
```{r results='asis', size="small"}
df<-read.csv("./Data/table_5.04.csv",sep=",")[,-1]
df$response<-grepl("Fall_forwardFall",df$nombre_fichero)+0
table(df$response)
```
```

```
  0   1
675  71
```

Del total de 746 registros, tenemos 71 veces en las que el patrón del sensor representa una caída de una persona.



Fuente: www.webpersonal.uma.es

Antes de meternos en modelizar el fenómeno, vamos a visualizar el fenómeno en concreto. En este problema no tenemos variables interpretables, sino que tenemos variables que corresponden a una ventana en el tiempo vinculada con el acelerómetro instalado en el sensor de la muñeca de una serie de personas. Esta ventana temporal es la que queremos modelizar para poder predecir que es una caída, si vuelve a pasar en el real-time.

```
```{r}
layout(matrix(c(1,2,3),1,3,byrow=T))

plot(as.numeric(df[680,c(-1)]),type="l",main=df[680,c(1)])

plot(as.numeric(df[420,c(-1)]),type="l",main=df[420,c(1)])

plot(as.numeric(df[520,c(-1)]),type="l",main=df[520,c(1)])
```
```



Como parece que el movimiento de caída se produce principalmente en los primeros 120 milisegundos. Filtramos la base de datos y procedemos a modelizar.

```{r}
df<-df[,-c(1,120:304)]
colnames(df)
```

```
  [1] "x1"      "x2"      "x3"      "x4"      "x5"      "x6"      "x7"      "x8"      "x9"      "x10"
 [11] "x11"     "x12"     "x13"     "x14"     "x15"     "x16"     "x17"     "x18"     "x19"     "x20"
 [21] "x21"     "x22"     "x23"     "x24"     "x25"     "x26"     "x27"     "x28"     "x29"     "x30"
 [31] "x31"     "x32"     "x33"     "x34"     "x35"     "x36"     "x37"     "x38"     "x39"     "x40"
 [41] "x41"     "x42"     "x43"     "x44"     "x45"     "x46"     "x47"     "x48"     "x49"     "x50"
 [51] "x51"     "x52"     "x53"     "x54"     "x55"     "x56"     "x57"     "x58"     "x59"     "x60"
 [61] "x61"     "x62"     "x63"     "x64"     "x65"     "x66"     "x67"     "x68"     "x69"     "x70"
 [71] "x71"     "x72"     "x73"     "x74"     "x75"     "x76"     "x77"     "x78"     "x79"     "x80"
 [81] "x81"     "x82"     "x83"     "x84"     "x85"     "x86"     "x87"     "x88"     "x89"     "x90"
 [91] "x91"     "x92"     "x93"     "x94"     "x95"     "x96"     "x97"     "x98"     "x99"     "x100"
[101] "x101"    "x102"    "x103"    "x104"    "x105"    "x106"    "x107"    "x108"    "x109"    "x110"
[111] "x111"    "x112"    "x113"    "x114"    "x115"    "x116"    "x117"    "x118"    "response"
```

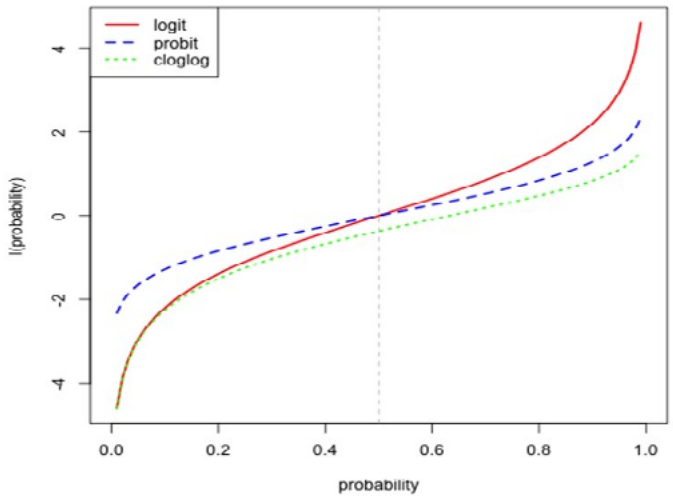
Cada uno de los milisegundos que componen la base de datos va a ser una variable explicativa. Dentro de un entorno multivariante donde todos los milisegundos consideraos como variables explicativas se combinan para intentar predecir correctamente el suceso. Independientemente de los milisegundos significativos y no significativos, vamos a introducir todos ellos en el modelo lineal para ver el ajuste al que nos lleva.

¿Cómo modelizamos un suceso binomial?

Las funciones link más habituales para respuesta binaria son las siguientes:

$$Logit(p) = ln(p/(1-p))$$
$$Probit(p) = \phi^{-1}(p)$$
$$Cloglog(p) = ln(-ln(1-p))$$

La resolución del modelo es con el método de máxima verosimilitud visto anteriormente y con un método numérico para su resolución. Así, propongo tres modelos diferentes:



Fuente www.towardsdatascience.com

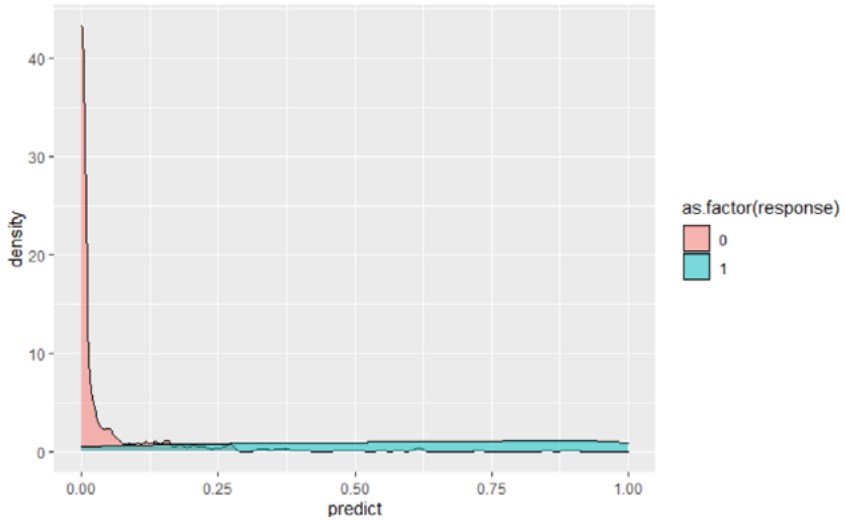
1. **Modelo Logit.**
2. **Modelo Probit.**
3. **Modelo Probit con el algoritmo MARS que vimos antes.**

```
```{r warning=FALSE,message=FALSE}
modelo_logit<-glm(response~.,data=df,family=binomial(link="logit"))
modelo_probit<-glm(response~.,data=df,family=binomial(link="probit"))
modelo_earth<-earth(response~.,data=df,glm=list(family=binomial(link=probit)))
```
```

En este problema, no nos interesa tanto la interpretabilidad de los datos como la capacidad de predicción.

¿Cómo evaluamos la capacidad de predicción en modelos binarios?

A través de la curva ROC. Pero antes de la ROC tenemos que saber qué es la matriz de confusión.

La matriz de confusión es la relación entre valor real y valor estimado:

1. **Verdadero Positivo (VP):** estimamos Positivo y Realidad Positivo.
2. **Falso Positivo (FP):** estimamos Positivo y Realidad Negativo.
3. **Verdadero Negativo (VN):** estimamos Negativo y Realidad Negativo.
4. **Falso Negativo (FN):** estimamos Negativo y Realidad Positivo.

De aquí surgen dos conceptos para relacionar los mismos, la especificidad y la sensibilidad.

1. **Especificidad:** VN/(VN+FP). Cuanto mayor mejor.
2. **Sensibilidad:** VP/(VP+FN). Cuanto mayor mejor.

IMPORTANTE

Los modelos Probit, Logit, Cloglog nos dan una probabilidad asociada al individuo. Es aquí donde la labor del analista es fundamental a la hora de determinar lo que se considera Positivo o Negativo.

PIENSA UN MINUTO

Si la probabilidad de que los valores del acelerómetro sean una caída es de un 80% ¿Lo podemos considerar caída? Si la probabilidad es un 70% ¿Lo podemos considerar caída? Si la probabilidad es un 60% ¿Lo podemos considerar caída? y así sucesivamente.

Hay que establecer uno corte para determinar lo que se considera Positivo y Negativo.

IMPORTANTE

Lo que nos mide la curva ROC es que dados todos los cortes desde el 0 hasta el 1, en cada uno de los intervalos vamos a medir la Especificidad y la Sensibilidad. Esto nos va a dar un área que es el área bajo la curva ROC (AUC). Lo cual quiere decir que cuanto mayor sea nuestra área, mayor será la capacidad predictiva del modelo.

```
```{r results='asis', size="small",warning=FALSE,message=FALSE}
auc(df$response,predict(modelo_logit,df,type="response"))
auc(df$response,predict(modelo_probit,df,type="response"))
auc(df$response,predict(modelo_earth,df,type="response"))
```
```

```
Area under the curve: 0.9541
Area under the curve: 0.9583
Area under the curve: 0.9573
```

Todos los modelos tienen un muy buen nivel de ajuste. Esto lo podemos confirmar viendo las distribuciones de los 1 y los 0 y los valores que hemos predicho. Este tipo de visualización nos ayuda a ver el punto de corte óptimo que tendríamos que meter en la probabilidad para maximizar las diferencias entre 0 y 1. En este problema lo que estamos visualizando es que los 0 los estamos captando muy bien, sin embargo, los 1 (cuando la persona se cae) recibe una probabilidad más elevada, pero no está concentrada alrededor de la probabilidad 1.

```
predict<-predict(modelo_probit,df,type="response")
ggplot(df, aes(x =predict , fill = as.factor(response))) +  geom_density(alpha = .5)
```



1.7 ESTRATEGIA ANTE LA MODELIZACIÓN GLM

Para finalizar el apartado voy a recapitular los pasos que hemos ido dando para confeccionar un buen modelo:

1. Creación de una buena base de datos con información interna y externa.
2. Decisión sobre el tipo de datos que seguirá nuestra variable respuesta condicionada a las variables explicativas.
3. Qué función / funciones Link vendría bien meter en este modelo.
4. Métodos de selección de variables.
5. Ajuste de variables frente a no linealidades.
6. Evaluación de los residuos del modelo. Normalidad, Independencia, Homocedasticidad.
7. Poder predictivo del modelo y comparativa contra otros modelos.

¿Sabemos todo sobre cómo hacer un muy buen modelo lineal?

Aún quedan algunos aspectos que tendremos que tener en cuenta:

- Cuando hemos generado nuestro modelo, hemos calculado las betas con una base de datos y hemos visto el grado de ajuste con esa misma base de datos. ¿No podríamos coger otra base de datos para comprobar el ajuste?
- Cuando hemos realizado el modelo, hemos tomado unas variables como dadas. ¿Le podemos meter al modelo algún tipo de parámetro que penalice cada variable que metemos de más?
- Cuando hemos testado la autodependencia de los residuos, lo hemos hecho con un entorno estático o bien visto a lo largo del tiempo. ¿Y si los residuos están relacionados espacialmente?

A estas cuestiones les vamos a dedicar un apartado especial.

IDEAS CLAVE

- Cuando las hipótesis del modelo lineal Gaussiano no se cumplen, los GLM nos ayudan a flexibilizar las asunciones iniciales. Al utilizar la función link junto con una distribución de probabilidad acorde a los datos que tenemos podemos conseguir un modelo robusto, consistente e insesgado, lo cual quiere decir resultados mucho mejores.

- Una vez tengamos nuestro GLM correctamente estimado tendremos que invertir las predicciones del modelo en función de la función link utilizada.

- Tanto si es un modelo lineal como un GLM debemos comprobar la heterocedasticidad y autocorrelación de los residuos para poder validar el modelo.