exploreGLM:

Exploración automática de un Modelo Lineal Generalizado mediante la creación de un paquete para el software estadístico R

PATRICIA BALLESTA TORRE

Grau d'Estadística Aplicada de la Universitat Autònoma de Barcelona

Trabajo Final de Grado dirigido por Jose Barrera-Gómez

Dedicado a mis padres,
que tanto apoyo y amor me brindan cada día.

Agradecimientos

Agradezco a mi tutor, Jose Barrera, las ganas, el apoyo y las muchas horas que ha dedicado a ayudarme en este trabajo. Ha sido un gran guía en todo momento y sus críticas y saber hacer, sin duda me han ayudado a mejorar tanto en este proyecto, como en muchos otros aspectos de mi vida académica y profesional.

Quería agradecer, por supuesto, los consejos y las respuestas de mis dudas a algunos de los miembros de **RUG Barcelona**, Roger Borràs, Lluís Ramon Callao, Aleix Ruiz y Andreu Vall, a los que considero unos amigos muy especiales de los que siempre hay mucho que aprender en sus apasionantes conversaciones.

Quiero dar las gracias también al genial Albert Llorens por su *mágico* inglés.

exploreGLM: Exploración automática de un Modelo Lineal Generalizado.

Patricia Ballesta[a] patriciaballesta2@gmail.com

Jose Barrera-Gómez[a,b] jbarrera@creal.cat

4 de septiembre de 2013

[a]Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain.

[b]Centre for Research in Environmental Epidemiology (CREAL), Barcelona, Spain.

Índice

1. Introducción

1.1. Presentación

Este trabajo tiene como objetivo aprender a crear un paquete para R. R es un software libre, ampliamente extendido tanto en la comunidad académica, como en la profesional y, la creación de un paquete, aparte de ser algo muy beneficioso a nivel personal, es la mejor manera de compartir código con el resto de compañeros de estudios y de profesión.

1.2. Motivación

Uno de los conocimientos esenciales que me ha aportado el Grado de Estadística Aplicada es la programación en múltiples lenguajes y para múltiples aplicaciones. Desde que descubrí la programación en C, me di cuenta de que plantear un problema y crear un programa que lo solucionara de forma automática, era algo con lo que disfrutaba tanto como con la estadística en sí.

Desde el primer momento en que nos presentaron el software estadístico R, tuve curiosidad por saber cómo se programaba un paquete y este trabajo me da la oportunidad de aprender a hacerlo y de valorar toda la complejidad y esfuerzo que conlleva.

Este conocimiento, además puede ser útil en mi empleo actual relacionado con el control estadístico de los procesos de producción en la industria farmacéutica. Aunque existen unos pocos paquetes en `http://cran.r-project.org` dedicados al tema, muchas veces no se adaptan por completo a las necesidades y debo programar mis propias funciones y gráficos mucho más específicos. La creación de un paquete y la introducción a la documentación para la investigación reproducible (R + LaTeX con Sweave y Beamer) podría ser un proyecto que ayudaría en mi empresa a automatizar en gran parte la redacción de los informes de validación de procesos.

A lo largo del Grado de Estadística Aplicada hemos ido adquiriendo conocimientos básicos sobre el manejo de R, pero una de las utilidades más interesantes que comprenden este software, es la creación de paquetes propios y esto no me ha sido posible aprenderlo durante la carrera. Por ello, y ya que este conocimiento es uno de los requisitos básicos en innumerables ofertas de trabajo dirigidas a los profesionales de la Estadística, mi tutor y yo decidimos que aprender a programar un paquete para R era un tema adecuado como Trabajo Final de Grado.

Además, la utilidad del paquete en sí permite conocer algunos modelos estadísticos que no se estudian durante el Grado (o dependen de la elección de asignaturas) como son el modelo de regresión de Poisson y los politómicos con respuesta ordinal y nominal. Por último, el hecho de escribir la viñeta del paquete, la memoria y la presentación con R Sweave y Beamer me permiten profundizar un poco más en mis conocimientos de LaTeX adquiridos durante el último semestre de la carrera.

2. El paquete exploreGLM

2.1. Pasos básicos para la creación de un paquete en R

Las pautas básicas para la creación del paquete son las siguientes:

1. Pensamos qué queremos que haga nuestro paquete a nivel de computación.
2. Creamos nuestro código de R con las funciones necesarias para ello.
3. Creamos los métodos `print`, `plot`, `summary`, etc. necesarios para que el resultado que se imprime por pantalla tenga el formato deseado.
4. Creamos el esqueleto del paquete usando las herramientas que R proporciona.
5. Rellenamos la documentación del esqueleto.
6. Chequeamos, empaquetamos e instalamos el paquete. Éste es tal vez, el paso más delicado y en el que solemos encontrar más problemas ya que R es exigente y cualquier pequeño error, en el código o en la documentación, puede ocasionar incoherencias que el programa no permite.
7. Hacemos pruebas para evaluar su funcionamiento.
8. Podemos añadir nuevas funciones, datos, etc. al paquete repitiendo los pasos anteriores.
9. Creamos la viñeta que nos permite, por un lado, explicar detalladamente el funcionamiento del paquete y, por otro lado, realizar comprobaciones del correcto funcionamiento del paquete.

En la creación de un paquete también podemos hacer interaccionar R con `C` (para ganar tiempo en grandes computaciones).

2.2. Presentación del paquete exploreGLM

El paquete exploreGLM está pensado para la exploración de bases de datos por parte de usuarios con conocimientos básicos de estadística (estudiantes del Grado de Estadística Aplicada, por ejemplo, ya que como he comentado, explora los datos a través de modelos que no siempre utilizamos en la carrera). Por lo tanto, el paquete tiene como objetivo ayudar a que una persona sin muchos conocimientos de modelización, pueda ajustar de manera automática el modelo adecuado y le sirva de ayuda también, para iniciarse en él. Los casos considerados en la versión actual de exploreGLM son los modelos de regresión lineal, logístico, Poisson, politómico ordinal y politómico nominal.

Queda fuera del alcance de este trabajo explicar estos modelos de regresión en profundidad, aunque sí se incluye una breve exposición dentro del Apéndice A de esta memoria.

Por ser ésta una versión preliminar del paquete exploreGLM, es posible continuar la ampliación del mismo mejorando y agregando nuevas funcionalidades que se deseen aportar en el futuro. Algunos ejemplos de éstas se dejan expuestos en la sección 5 (*Conclusiones y valoración*) de esta memoria.

El paquete exploreGLM para R puede descargarse en `http://www.mat.uab.cat/~jbarrera/Software.html`. El manual y la viñeta que ilustran cómo debe utilizarse también están disponibles en la misma web y dentro del paquete exploreGLM una vez instalado y accediendo a la ayuda del mismo.

2.3. Objetivo del paquete exploreGLM

El objetivo principal de este paquete es realizar una exploración de la base de datos. Para ello, reconoce los distintos tipos de variables dentro de la misma, con el fin de (1) reportar los estadísticos descriptivos adecuados para cada una de estas variables en función de su naturaleza (R por defecto considera como variable numérica, cualquier variable codificada numéricamente) y también de (2) ajustar el modelo de regresión oportuno, según el tipo de variable respuesta, indicada por el usuario, de forma semi-automática. Los modelos considerados en esta versión son: Modelo de regresión lineal, logístico, Poisson, politómico ordinal y politómico nominal.

2.4. Introducción al paquete exploreGLM

A fin de que el paquete pueda desarrollar las tareas para las que ha sido generado, necesita en primer lugar una función que sea capaz de reconocer y guardar información sobre cada una de las variables y, en segundo lugar, una función que sea capaz de recopilar dicha información con el objetivo de ajustar un modelo de regresión adecuado a la respuesta. Las funciones encargadas de realizar esto, son las principales del paquete exploreGLM: (1) `prepareData` y (2) `exploreglm`. En la siguiente sección las describimos.

3. Funcionamiento del paquete exploreGLM

3.1. Función `prepareData`

Esta función identifica y almacena información sobre la naturaleza de cada variable de la base de datos. De esta manera puede crear el resumen descriptivo adecuado para cada una y además, preparar la base de datos con el objetivo de facilitar posteriormente la detección del modelo de regresión a ejecutar, en base al tipo de variable respuesta proporcionada.

Los argumentos de entrada que necesita son:

- `data`: Base de datos que contiene las variables que deben ser descritas por `summary` y/o ajustadas por `exploreglm`.
- `k`: Una variable codificada como numérica se considerará categórica si tiene un número de valores distintos $<= k$. Por defecto $k = 5$.
- `idpos`: Posición (número de columna) en la que se encuentra la variable identificadora, si la hay.
- `idname`: Nombre de la variable identificadora, si la hay.

Sólo debe proporcionarse uno de los dos argumentos `idpos` o `idname`, o ninguno en caso de no existir una variable de este tipo. En caso de proporcionar los dos, prevalecerá `idname`. La variable asociada a `idpos` o `idname` será omitida en los análisis.

3.2. Función `exploreglm`

Esta función es la encargada de ajustar el modelo adecuado. Para ello, identificará en qué posición dentro de la base de datos se encuentra la variable respuesta indicada por el usuario. El objeto de clase *prepareData*, requerido como argumento de entrada, se encarga de informar sobre el tipo de variable respuesta y el número de valores diferentes que componen la misma. Una vez identificada y teniendo en cuenta, en caso de ser numérica, si se indica como `count = TRUE/FALSE` o, en caso de ser categórica, como `ordinal = TRUE/FALSE` aplicará el modelo de regresión apropiado entre los cinco siguientes:

Tipo Respuesta Y	Modelo Regresión	Función y Librería R
Continua	Lineal	glm `stats` family = gaussian
Dicotómica	Logística	glm `stats` family = binomial
Recuento	Poisson	glm `stats` family = poisson
Categórica Ordinal	Politómica Ordinal	polr `MASS`
Categórica Nominal	Politómica Nominal	multinom `nnet`

Todas las variables de la base de datos, excepto la respuesta y la identificadora, serán utilizadas como variables explicativas.

Los argumentos de entrada que necesita son:

- `object`: Objeto creado por la función `prepareData`.
- `y`: Nombre de la variable respuesta.
- `ordinal`: Indicado como *TRUE* si la variable respuesta es de tipo ordinal. Por defecto será *FALSE*.
- `count`: Indicado como *TRUE* si la variable respuesta es un recuento. Por defecto será *FALSE*.

Los dos últimos argumentos (`count` y `ordinal`) requieren la aportación del usuario, ya que es imposible que la máquina pueda distinguir sola entre una variable numérica sin decimales y un recuento, o entre una variable nominal y una ordinal.

4. Ejemplos ilustrativos de la aplicación de exploreGLM

El primer paso, una vez instalada la librería [1], será cargarla de la forma siguiente:

```
> library("exploreGLM")
```

4.1. Datos `flowers`

Para ilustrar el funcionamiento utilizaremos la base de datos `flowers` incluida en el paquete exploreGLM.

```
> data(flowers)
```

Esta es una base de datos creada a partir de un experimento propio, realizado con el objetivo de explorar cuál era el efecto que producía una substancia medicamentosa, diluida en el agua que conserva un ramo de flores, dentro de un recipiente durante 10 días. El experimento fue aplicado con todo el rigor que caracteriza a un ensayo clínico.

Los datos hacen referencia a 120 flores de 3 clases diferentes: Dendranthema o Chrysanthemum grandiflorum blancos, Chrysanthemum cinerariaefolium rojos y Dianthus caryophyllus amarillos (nombre común: Crisantemo, Pelitre de Dalmacia y Clavel, etiquetados con sus nombres en inglés: Mum, Dalmatian y Carnation respectivamente).

Se consideraron 8 combinaciones distintas de tratamiento según el tipo de medicamento (Ácido acetilsalicílico -ASA-, Ibuprofeno, Paracetamol o Control -sin ningún medicamento-) y el tipo de agua contenida en el recipiente (Corriente o Mineral).

Podemos obtener la documentación correspondiente a los datos `data(flowers)` mediante la instrucción:

```
> ?flowers
```

De esta manera visualizaremos la siguiente información sobre las variables contenidas:

- `idflo`: Identificador de cada flor.
- `treat`: Tratamiento suministrado a la flor: `ASA`, `Control`, `Ibuprofen` o `Paracetamol`.
- `water`: Tipo de agua del recipiente: `Normal` o `Mineral`.
- `flower`: Especie de flor: `Carnation`, `Dalmatian` o `Mum`.
- `EVA`: Variable numérica sobre el estado de la flor al final del seguimiento. Escala de 0 a 100 de menos a más marchita.
- `color`: Color del agua al final del seguimiento: `Yellow` o `White`.
- `defects`: Número de defectos en la flor al final del seguimiento.
- `petal`: Factor ordinal sobre el tamaño de los pétalos: `Big` >`Medium` >`Small`.

4.2. Preparación y resumen descriptivo de la base de datos `flowers`

Prepararemos la base de datos aplicando en primer lugar la función `prepareData`:

```
> pDflo <- prepareData(data = flowers, idname = "idflo")
```

Para más información sobre la función `prepareData`:

```
> ?prepareData
```

En este punto, la función ha preparado los datos, que han sido formateados como un objeto que ya no es el estándar de R para bases de datos. Así, al utilizar el método `print` obtendremos la visualización que ha sido creada para los objetos con clase *prepareData*:

```
> pDflo
```

El mismo método se puede ejecutar, indicando cuántos registros desean visualizarse:

```
> print(pDflo, lines = 15)
```

```
Variables (those with are least 6 different numerical values are assumed
to be numerical):
     Name        Type  Different values
1   treat categorical                 4
2   water categorical                 2
3  flower categorical                 3
4     EVA   numerical                91
5   color categorical                 2
6 defects   numerical                24
7   petal categorical                 3

Printing the  15  first records:
   treat  water    flower  EVA  color defects  petal
1    ASA Normal Dalmatian 71.5 Yellow      13    Big
2    ASA Normal Dalmatian 86.5 Yellow      24    Big
3    ASA Normal Dalmatian 87.0 Yellow       1  Small
4    ASA Normal Dalmatian 94.5 Yellow      14  Small
5    ASA Normal Dalmatian 92.5 Yellow      12    Big
6    ASA Normal       Mum 89.5 Yellow       9 Medium
7    ASA Normal       Mum 83.5 Yellow       3    Big
8    ASA Normal       Mum 92.5 Yellow       8  Small
9    ASA Normal       Mum 93.0 Yellow      17    Big
10   ASA Normal       Mum 92.0 Yellow       8    Big
11   ASA Normal Carnation 50.5 Yellow       5  Small
12   ASA Normal Carnation 48.0 Yellow       1 Medium
13   ASA Normal Carnation 57.0 Yellow       5    Big
14   ASA Normal Carnation 70.0 Yellow      16    Big
15   ASA Normal Carnation 74.0 Yellow       2  Small
```

Hemos adaptado también, un método `summary` para un objeto de clase *prepareData*:

```
> summary(pDflo, digits = 1)

==========================================================================
Number of Variables:  7     Observations:  120
Maximum number of categories in a numerical variable to be treated as
categorical: 5
==========================================================================
Qualitative Variables:
======================
Variable: treat , Missing values: 0  ( 0 %)
            ASA Control Ibuprofen Paracetamol Total
Frequencies  30      30        30          30   120
Percentages  25      25        25          25   100
--------------------------------------------------------------------------
Variable: water , Missing values: 0  ( 0 %)
            Mineral Normal Total
Frequencies      60     60   120
Percentages      50     50   100
--------------------------------------------------------------------------
Variable: flower , Missing values: 0  ( 0 %)
            Carnation Dalmatian  Mum Total
Frequencies      40.0      40.0 40.0   120
Percentages      33.3      33.3 33.3   100
--------------------------------------------------------------------------
Variable: color , Missing values: 0  ( 0 %)
            White Yellow Total
Frequencies    60     60   120
Percentages    50     50   100
--------------------------------------------------------------------------
Variable: petal , Missing values: 0  ( 0 %)
             Big Medium Small Total
Frequencies 40.0   40.0  40.0   120
Percentages 33.3   33.3  33.3   100
--------------------------------------------------------------------------
Quantitative Variables:
======================
        nMiss %Miss Mean   SD Min 2.5%   5%   Q1 Median   Q3  95% 97.5%  Max
EVA         0     0 56.1 26.4 2.5    6 10.8 35.9   61.8 76.6 92.5  94.6 99.5
defects     0     0 12.5  7.0 1.0    1  2.0  6.8   12.5 18.2 23.0  24.0 24.0
```

En este resumen descriptivo aparecen en primer lugar las variables cualitativas y en segundo lugar las cuantitativas. Nótese también que la variable *idflo*, reconocida como identificadora, no aparece en el resumen y tampoco se usará en el ajuste de modelos.

4.3. Ajuste de modelos mediante `exploreglm`

A modo ilustrativo presentamos dos ejemplos: (1) un modelo de regresión lineal y (2) un modelo de regresión politómica para respuestas ordinales.

4.3.1. Ejemplo 1: Modelo de regresión lineal

Consideremos el caso en el que deseamos modelar la variable *EVA* en función del resto de variables de la base de datos. Dado que la función `exploreglm` determina el modelo más adecuado, según el tipo de variable respuesta y ésta es continua, el modelo de regresión lineal será el que ejecute en este caso.

No hará falta indicar los argumentos `ordinal` y `count` ya que por defecto son *FALSE*. Recordemos que el argumento `object` debe ser el objeto de clase `perpareData`, que hemos obtenido préviamente en la sección anterior.

```
> modEVA <- exploreglm(object = pDflo, y = "EVA")
```

En este momento el objeto *modEVA* contiene el modelo ajustado para las variables contenidas en la base de datos *pDflo*.

Para más información sobre esta función consultar:

```
> ?exploreglm
```

A través del método `print` creado para los objetos con clase *exploreglm*, obtendremos información básica sobre nuestra variable respuesta, el tipo de modelo ajustado y la fórmula asociada al mismo, como se muestra a continuación:

```
> modEVA
```

```
========================================================================
Dependent variable: EVA (Continuous variable)
Model type: Linear regression model
Model:  EVA ~ treat + water + flower + color + defects
========================================================================
Further information about this model with summary()
```

Para obtener más información sobre este modelo, tal y como indica la salida del método `print`, podemos utilizar la adaptación del método `summary`, que añade la cabecera anterior al `summary` genérico de `glm`.

```
> summary(modEVA)
```

```
==========================================================================
Dependent variable: EVA (Continuous variable)
Model type: Linear regression model
Model:  EVA ~ treat + water + flower + color + defects
==========================================================================

Call:
glm(formula = EVA ~ treat + water + flower + color + defects,
    family = gaussian, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-47.028  -14.598    2.402   14.196   42.029

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      101.5897     6.8616  14.806  < 2e-16 ***
treatControl     -43.9933     6.3027  -6.980 2.27e-10 ***
treatIbuprofen   -45.0012     6.2793  -7.167 9.00e-11 ***
treatParacetamol -21.6050     5.0542  -4.275 4.07e-05 ***
waterNormal       -7.3518     3.7509  -1.960   0.0525 .
flowerDalmatian  -12.1934     4.8648  -2.506   0.0136 *
flowerMum         -7.3619     4.3645  -1.687   0.0945 .
colorYellow       -7.1061     5.6105  -1.267   0.2080
defects           -0.3261     0.2713  -1.202   0.2320
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 380.3472)

    Null deviance: 83179  on 119  degrees of freedom
Residual deviance: 42219  on 111  degrees of freedom
AIC: 1064.1

Number of Fisher Scoring iterations: 2
```

Los gráficos genéricos del diagnóstico del modelo (Figura 1) se pueden obtener mediante la instrucción:

```
> plot(modEVA)
```

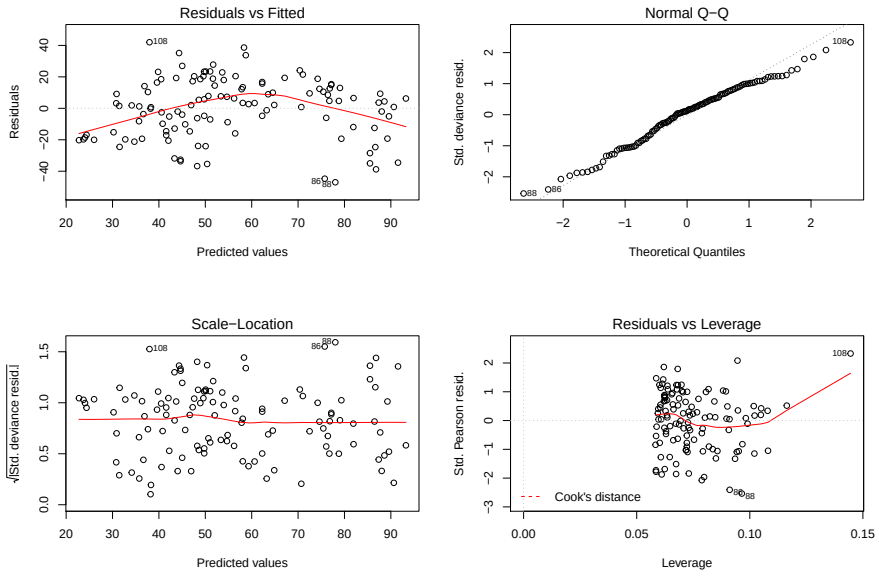

Figura 1: Gráficos de diagnóstico del modelo lineal para la respuesta *EVA*.

4.3.2. Ejemplo 2. Modelo de regresión politómica con respuesta ordinal

En el siguiente ejemplo, la variable respuesta que queremos evaluar es *petal*. Ésta es de tipo categórica y ordinal, por lo que la función `exploreglm` deberá aplicar un modelo de regresión ordinal para obtener la exploración del ajuste con el resto de variables. En este caso indicaremos en los argumentos de entrada `ordinal = TRUE`.

```
> modpetal <- exploreglm(object = pDflo, y = "petal", ordinal = TRUE)
```

Para acceder a los coeficientes y la información sobre el ajuste del modelo se utilizará el método `summary`, que añade una cabecera al método genérico para los objetos creados por `polr` con la información básica sobre nuestra variable respuesta, el tipo de modelo ajustado y la fórmula asociada al mismo.

```
> summary(modpetal)
```

```
========================================================================
Dependent variable: petal (Ordinal variable)
Model type: Logistic regression model for polytomous ordinal data
Model:  petal ~ treat + water + flower + EVA + color
========================================================================
Call:
polr(formula = petal ~ treat + water + flower + EVA + color,
    data = data)

Coefficients:
    treatControl   treatIbuprofen treatParacetamol      waterNormal
     -1.40957133      -1.71757356      -0.41467735      -0.73855034
 flowerDalmatian        flowerMum              EVA      colorYellow
     -0.30850022      -0.17929862      -0.01278989      -0.48010151

Intercepts:
  Big|Medium Medium|Small
   -3.133123    -1.640650

Residual Deviance: 252.5829
AIC: 272.5829
```

Dado que en este tipo de modelos no resultan útiles los diagnósticos gráficos, no se han incluido en el paquete, por lo que el método `plot` devolverá el siguiente mensaje:

```
> plot(modpetal)
> 'Error in plot.exploreglm(modpetal) :
+   ordinal and nominal models doesn´t have a defined plot in
+   exploreGLM package'
```

Se pueden obtener más ejemplos en la viñeta, que puede ser rescatada mediante la instrucción:

```
> vignette("exploreGLM")
```

5. Conclusiones y valoración

En este trabajo el objetivo principal era aprender cómo crear mi primer paquete para R y por ello pensamos en que un tema interesante a desarrollar, podía ser un paquete que automatizara la exploración de una base de datos. Hay detalles que no quedan reflejados, como por ejemplo la dificultad que supone la primera vez la implementación de todo lo requerido por el paquete, pero creo que los objetivos que nos planteamos al principio los hemos logrado alcanzar.

Una vez creadas las funciones que contiene el paquete, fue necesario probar y pensar en todas las posibilidades que podían dar lugar a errores en el funcionamiento del mismo y depurarlas y corregirlas mediante la sintaxis de validación oportuna dentro de cada una. Esta validación se encarga de informar al usuario sobre los errores cometidos en la introducción de los argumentos de entrada, o los corrige, informando de igual manera, sobre la decisión tomada en el momento de subsanar el error.

Aunque mi nivel de programación no es el suficiente como para crear un código más eficiente, durante este trabajo sí he mejorado los conocimientos que tenía respecto al programa y me he dado cuenta de que, en la creación de un paquete de R, a parte de las funciones en sí, la implementación de los archivos de documentación es algo muy importante (para poder compartirlo con otros usuarios) y requiere de muchas horas de dedicación.

El momento de empaquetar también es delicado y cada cambio en las funciones o en su documentación puede provocar fallos. Por ello, he aprendido que se debe prestar mucha atención y ser ordenado con tal de evitar descuidos que afecten a la creación del paquete.

El paquete exploreGLM deja abiertas las puertas a futuras mejoras y puede ampliarse si se desea aportándole más funcionalidad, eficiencia y/o mejoras. Por ejemplo, se podrían considerar los siguientes aspectos: permitir que el usuario seleccione un subconjunto de variables explicativas, crear gráficos que complementen el resumen descriptivo, implementar exploración bivariante, considerar el modelo binomial negativo, añadir modelos que consideren otro tipo de variable respuesta, añadir la posibilidad de elegir a través de qué funciones de enlace queremos transformar las respuestas, modificar el método `summary` para que muestre por pantalla una ayuda sobre cómo se interpretan los coeficientes del modelo o implementar métodos `anova` y `predict`.

Siempre he compatibilizado mis estudios con la vida laboral y, este año en especial, he comenzado una experiencia muy interesante como estadística, en un campo en el que creo que podré aplicar todo lo aprendido aquí.

Tiene una especial importancia para mí, a parte de la creación del paquete en sí, el haberme introducido en la documentación para la investigación reproducible (R + LaTeX con Sweave y Beamer) ya que he mejorado notablemente los conocimientos básicos que obtuve al final de la carrera y creo que las consecuencias que de ello se derivan, tanto en mi vida académica, como en mi vida profesional, son muy positivas.

Referencias

[1] Ballesta P, Barrera-Gomez J. *R Package exploreGLM*. 2013. `http://www.mat.uab.cat/~jbarrera/Software.html`.

[2] R Core Team. *Writing R Extensions*. Version 2.15.1. 2012.

[3] Ruiz-Ruano Campaña FJ. *LaTeX con LyX*. Versión 0.2. 2010.

[4] *Apuntes LaTeX*. `http://metodos.fam.cie.uva.es/~latex/apuntes/apuntes.html`.

[5] Apuntes de la asignatura *Dades transversals: temes avançats en Ciències de la Salut* del *Grau d'Estadística Aplicada* de la *Universitat Autònoma de Barcelona*. Barrera-Gómez J.

[6] Apuntes de la asignatura *Models Lineals* del *Grau d'Estadística Aplicada* de la *Universitat Autònoma de Barcelona*. Garcia C.

[7] Faraway JJ. *Extending the Linear Model with R*. Chapman & Hall, 2006.

[8] Agresti A. *Categorical Data Analysis*. Wiley, 2nd Ed. 2007.

Apéndices

A. Presentación de los modelos considerados en exploreGLM

De manera sencilla, podemos entender los modelos lineales generalizados (en inglés Generalized Linear Models, GLM) como una extensión de los modelos lineales que permiten modelizar diferentes tipos de variable respuesta, sin la necesidad de que ésta sea continua (por ejemplo, variables dicotómicas, de recuento, categóricas...). Aplicando diferentes transformaciones sobre la respuesta, se originan los diferentes tipos de GLM.

Si $X = (X_1, X_2, ..., X_k)$ es un vector de covariables, podemos ajustar los siguientes modelos en función del tipo de variable respuesta Y.

A.1. Modelo de regresión lineal

En este caso, Y es una variable respuesta continua y el modelo tiene la forma:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

Si una variable X_i es numérica, el coeficiente β_i representa el incremento o decremento (si fuese negativo) que experimenta la variable respuesta Y al aumentar X_i en una unidad, mientras el resto de variables del modelo se mantienen fijas.

Si X_i es una variable categórica, el coeficiente β_i representará el incremento o decremento en la variable respuesta Y al cambiar X_i de categoría, mientras el resto de variables del modelo se mantienen fijas.

A.2. Modelo de regresión logística

En este caso, tenemos una variable respuesta que puede tomar dos valores posibles (dicotómica), como por ejemplo $Y \in \{0,1\}$. Por ello, la variable necesitará una transformación a través de una función de enlace distinta de la *identidad* (como en el modelo de regresión lineal).

La transformación utilizada en el caso de exploreGLM es el *Logit* y el modelo se expresa como:

$$Logit(Y) = log\left(\frac{P(Y)}{1 - P(Y)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

La interpretación de los coeficientes del modelo se realiza en términos de Odds Ratio (OR) (véase [5], [7], [8]).

Si X_j es una variable dicotómica,

$$\widehat{OR}_{X_j}(Y) = \mathbf{e}^{\hat{\beta}_j}.$$

Manteniendo el resto de variables fijas:

- Si $\mathbf{e}^{\hat{\beta}_j}$ >1, diremos que la Odds es $\mathbf{e}^{\hat{\beta}_j}$ veces superior en la categoría $X_j = 1$ (respecto a $X_j = 0$).
- Si $\mathbf{e}^{\hat{\beta}_j} = 1$, no hay diferencias entre las Odds de las categorías de X_j.
- Si $\mathbf{e}^{\hat{\beta}_j}$ <1, diremos que la Odds es inferior en el grupo $X_j = 1$ en un $(1-\mathbf{e}^{\hat{\beta}_j})\cdot 100\,\%$ respecto al grupo $X_j = 0$.

Si X_j es una variable numérica, un incremento de r unidades en X_j, manteniendo el resto de variables fijas en el modelo, provoca una estimación del OR:

$$\widehat{OR}_{\Delta X_j}(Y) = \mathbf{e}^{r\hat{\beta}_j}.$$

A.3. Modelo de regresión de Poisson

En este caso, la variable respuesta expresa un recuento. De nuevo, dicha variable necesitará una transformación. En este tipo de modelos se utiliza como función de enlace el *logaritmo*.

El modelo tiene la forma:

$$\log\left(\mathbb{E}(Y)\right) = \log(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

donde $\mathbb{E}(Y) = \mu$ es la esperanza de la variable Y.

Si X_j es una variable dicotómica, la esperanza de la variable respuesta Y queda multiplicada por $\mathbf{e}^{\hat{\beta}_j}$ cuando pasamos de $X_j = 0$ a $X_j = 1$, mientras el resto de variables del modelo se mantienen fijas.

Si X_j es una variable numérica, un incremento de r unidades en X_j, manteniendo el resto de variables fijas en el modelo, provoca un efecto multiplicativo de $\mathbf{e}^{r\hat{\beta}_j}$ en la esperanza de la variable respuesta Y.

A.4. Modelo de regresión politómica con respuesta ordinal

En este caso, la variable respuesta es categórica y ordinal, $Y \in \{y_1 \leq y_2 \leq ... \leq y_p\}$. El modelo se expresa de la forma siguiente:

$$L(X) = \log\left(\frac{P(Y \leq y_j \mid X)}{1 - P(Y \leq y_j \mid X)}\right) = \alpha_j - (\beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k)$$

donde $j = \{1, 2, ..., p-1\}$.

Este modelo también se conoce por el nombre de modelo de *Odds proporcionales* [8]. La interpretación de los coeficientes se realiza, como en el modelo logístico, en términos de OR (véase [7], [8]).

A.5. Modelo de regresión politómica con respuesta nominal

En este modelo la variable respuesta es categórica, $Y \in \{y_1, y_2, ..., y_p\}$ pero sus categorías no siguen un orden natural. El modelo se expresa de la forma siguiente:

$$L(X) = \log\left(\frac{P(Y = y_j \mid X_i)}{P(Y = y_j \mid X_0)}\right) = \alpha_j + \beta_{j1}(X_{i1} - X_{01}) + ... + \beta_{jk}(X_{ik} - X_{0k})$$

donde $j = \{1, 2, ..., p\}$ y $i = \{1, 2, ..., n\}$.

La interpretación de los coeficientes se realiza, como en el modelo logístico, en términos de OR (véase [7], [8]).

Para profundizar más en el conocimiento de todos estos modelos véase [7], [8].