Apuntes LAB5

ANOVA y Regresión

Objetivos

En este laboratorio vamos a generalizar la comparación de medias de dos grupos al caso en el que tenemos 3 o más grupos independientes utilizando el Análisis de la Varianza (ANOVA). Una vez que hemos decidido que los k grupos son distintos, tendremos que prestar atención a las comparaciones múltiples y la corrección por comparaciones múltiples.

ANOVA

Utilizamos el conjunto de datos famuss.dat, en el que se recogen varias características demográficas, fenotípicas y genéticas de 595 individuos. Se estudian un polimorfismo funcional que se piensa está asociado con el tamaño y la fuerza muscular humana. Las diferentes variables recogidas se resumen en la siguiente tabla,

Variable	Descripción
id	Identificador del individuo.
sex	Sexo del individuo. Es una variable cualitativa con 2 niveles: `Male` y `Female`.
age	Edad en años.
race	Raza del individuo. Es una variable cualitativa con 5 niveles: `African Am` (afroamericano), `Caucasian`, `Asian`, `Hispanic` y `Other`.
height	Altura en pulgadas.
weight	Peso en libras.
actn3.r577x	Genotipo en la localización r577x del gen ACTN3. Es una variable cualitativa con 3 niveles: **CC**, **CT**, y **TT**.
ndrm.ch	Porcentaje del cambio en la fuerza del brazo no dominante, comparando dicha fuerza antes y después de un entrenamiento.

¿El cambio en la fuerza del brazo no dominante después del entrenamiento de resistencia está asociado con el genotipo?

Cargamos los datos en R utilizando el código,

```
workingDir <- "D:/Ingenieria
Biomedica/Bioestadistica/Material/Laboratorios"
setwd(workingDir)
famuss <- read.table(file.path(workingDir, "datos/famuss.dat"), header
= TRUE, sep = "\t")
```

La función tapply()

La función tapply() permite aplicar una función específica a cada grupo de individuos de vector. Tiene la estructura,

```
tapply(y, x, FUN, …)
```

donde y es el vector de los datos, x la variable que agrupa a los individuos y FUN es la función de interés, que puede ser una función ya implementada en R, o definida por el usuario. Los argumentos de FUN se pueden añadir a continuación.

Para calcular la media del cambio de fuerza en cada genotipo,

```
tapply(famuss$ndrm.ch, famuss$actn3.r577x, mean)
```

```
      CC       CT       TT 
48.89422 53.24904 58.08385 
```

Packages en R

Una de las ventajas de R es que se mantiene actualizado gracias a que tiene una activa comunidad de áreas de conocimiento muy diferentes. R es un proyecto colaborativo con miles de personas trabajando y mejorando su funcionalidad. Los paquetes o librerías de R son extensiones para poder hacer ciertas funciones o que podamos acceder a diferentes funcionalidades como conjuntos de datos diferentes. Cuando instalamos R se incluyen ocho bibliotecas o paquetes estándar, pero otros muchos están disponibles a través de Internet. Por ejemplo, actualmente hay disponibles más de 4000 librerías desarrolladas en R que nos podemos instalar desde CRAN (http://www.r-project.org) y que cubren multitud de funciones desarrolladas para solucionar problemas de muy diferentes ámbitos.

El *package* car contiene la función leveneTest() que permite llevar a cabo el contraste de Levene para contrastar la hipótesis de homogeneidad de varianzas. Antes de poder usar esta función debemos instalar esta librería,

```
install.packages("car")
```

Esta sentencia sólo necesitaremos ejecutarla una vez, puesto que una vez instalado no tendremos que volver a hacerlo. Lo que si debemos hacer cada vez que vallamos a usar ese package es cargarlo en nuestra sesión,

```
library(car)
```

Después de lo cual, ya podemos utilizar las funcionalidades disponibles en esa librería. La función leveneTest() tiene la estructura básica:

```
leveneTest(y, group)
```

donde y es el vector que contiene la variable respuesta y `group` la variable cualitativa, que debe ser un `factor`.

Ajustar un modelo ANOVA

La función `aov()` ajusta un modelo ANOVA a los datos,

```
aov(y ~ x)
```

donde y es el vector que contiene los datos para la variable dependiente y x la variable cualitativa.

Para visualizar la tabla ANOVA debemos utilizar la función `summary()` pasando como argumento el objeto `aov`.

Comparaciones múltiples

La función `pairwise.t.test()` se puede utilizar para llevar a cabo la comparación de todos los pares junto con la corrección por comparaciones múltiples. Es estructura general de esta función es,

```
pairwise.t.test(y, x, p.adj)
```

donde y es el vector de datos, x es la variable cualitativa que agrupa a los datos en k grupos y `p.adj` puede ser uno de varios métodos disponibles para llevar a cabo el ajuste por comparaciones múltiples. Utilizaremos `p.adj = "none"` para no llevar a cabo ningún tipo de ajuste o `p.adj = "bonf"` para hacer el ajuste de Bonferroni.

Cuando `p.adj = "bonf"` los p-valores que devuelve son los p-valores ajustados, es decir multiplicados por el número de comparaciones m, por lo que deben compararse con α y no con α^*.

MODELOS DE REGRESIÓN LINEAL

Ajustar un modelo de regresión lineal

La relación entre dos variables cuantitativas puede visualizarse en los diagramas de dispersión. La variable explicativa, X, siempre será representada en el eje X, mientras que la variable respuesta Y en el eje Y.

Utilizamos la función `lm()` para ajustar modelos de regresión lineales. La estructura de esta función es,

```
lm(y ~ x, data)
```

donde el primer argumento especifica las variables utilizadas en el modelo y el segundo argumento es el `data.frame` que contiene los datos. Es necesario utilizarle cuando este `data.frame` no es especificado en el primer argumento como `data$y ~ data$x`.
Esta función devuelve un objeto de la clase "`lm`" que contiene varias componentes como los coeficientes estimados, que se presentan directamente al ejecutar la función sin asignarla a ningún objeto. Para acceder a otros componentes hay que utilizar el carácter especial `$` o funciones específicas como `summary()`.

En la práctica, lo habitual es que existan varias variables explicativas que se asocien con una variable respuesta. Necesitaremos un modelo de regresión múltiple que no es más que una extensión del modelo de regresión simple. Una aplicación muy común de los modelos de regresión múltiple es estimar la asociación entre una variable respuesta y una variable explicativa primaria ajustando dicha relación por otras variables que podrían ser confusores.
Para ajustar estos modelos en R podemos utilizar la función `lm()` de la siguiente forma,

```
lm(y ~ x1 + x2, data)
```

donde el primer argumento especifica las variables utilizadas en el modelo. En este ejemplo, el modelo tiene dos variables explicativas `x1` y `x2`. Se pueden añadir variables independientes añadiéndolas a la fórmula con el símbolo +.

Algunas funciones interesantes para los objetos de la clase `lm`

- Podemos utilizar la función `predict()` para obtener los valores predichos de Y especificando unos valores de X. La sintaxis de esta función es,
  ```
  predict(object, newdata = data.frame( ))
  ```
  donde `object` es el modelo ajustado, y `newdata` es un data frame con los valores de X en los que queremos hacer la predicción. Los nombres de las variables incluidas en este data frame deben de corresponder con los nombres de las variables X incluidas en el modelo.
- Podemos utilizar la función `confint()` para obtener intervalos de confianza para los parámetros de la regresión,
  ```
  confint(object, level)
  ```
  donde `object` es el modelo ajustado, y `level` es el nivel de confianza.
- Podemos utilizar la función `anova()` para obtener la tabla ANOVA,
  ```
  anova(object)
  ```
  donde `object` es el modelo ajustado.

Validación de los modelos de regresión

Los métodos para evaluar la validez de un modelo ajustado son similares para todos los modelos de regresión lineal, sean simples o múltiples. En ambos casos, la herramienta fundamental será el plot de residuales, que representa los residuos, calculados como la diferencia entre el valor observado y el predicho $e_i = y_i - \hat{y}_i$, frente a los valores predichos del modelo $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_k X_k$.

Las asunciones que deben verificarse para que un modelo de regresión lineal ajustado por mínimos cuadrados sea válido son:

Linealidad. Para cada una de las variables explicativas $X_j \; con \; \mathrm{j} = 1, \ldots, \mathrm{k}$, los cambios en esa variable están relacionados linealmente con los cambios en la variable respuestas, cuando se mantienen constantes el resto de variables explicativas.

Homogeneidad de varianza. Los residuos tienen varianza aproximadamente constante.

Independencia. Las observaciones son independientes.

Normalidad. Los residuos están distribuidos aproximadamente como una normal.

Obtenemos los residuos de un modelo ajustado con la sentencia `resid(model)` y los valores predichos con `fitted(model)`.

MODELOS DE REGRESIÓN LOGÍSTICA

Para ajustar un modelo de regresión logística en R utilizamos la función `glm()` que tiene la siguiente estructura,

```
glm(y ~ x1 + x2, data, family = binomial(link = "logit"))
```

donde el primer argumento especifica las variables utilizadas en el modelo, en este caso estamos modelizando la probabilidad de que $y = 1$ utilizando dos variables explicativas x1 y x2. Es importante tener en cuenta que y debe ser una variable que toma valores 0 o 1, siendo 1 el valor asignado al evento cuya probabilidad queremos modelizar. El segundo argumento es el `data.frame` que contiene los datos. Es necesario utilizarle cuando este `data.frame` no es especificado en el primer argumento como `data$y ~ data$x1 + data$x2`. El argumento `family = binomial(link = "logit")` es específico para modelos de regresión logística. Es necesario incluirle ya que esta función es capaz de ajustar diferentes tipos de modelos lineales.

Esta función devuelve un objeto de la clase "`lm`" y "`glm`" que contiene varias componentes como los coeficientes estimados, que se presentan directamente al ejecutar la función sin asignarla a

ningún objeto. Para acceder a otros componentes hay que utilizar el carácter especial $ o funciones específicas como `summary()`.