Guía 3: Introducción a las variables aleatorias

Bioinformática - Pasto 2017

Guillermo Torres

Introduction

Con esta guía vamos a introducir conceptos estadísticos necesarios para entender p-values e intervalos de confianza. Usaremos como ejemplo usaremos las sentencias del paper de Winzell and Ahrén, 2004:

> "Body weight was higher in mice fed the high-fat diet already after the first week, due to higher dietary intake in combination with lower metabolic efficiency."

Para apoyar esto, ellos proporcionan los siguientes resultados:

> "Already during the first week after introduction of high-fat diet, body weight increased significantly more in the high-fat diet-fed mice ($+ 1.6 \pm 0.1$ g) than in the normal diet-fed mice ($+ 0.2 \pm 0.1$ g; $P < 0.001$)."

Qué sigunifica P<0.001?, Qué significa el ±?. A continuación aprenderemos a calcular estos valores en R e interpretar su significado. El primer paso es entender el significado de variables aleatorias, para lo cual usaremos los datos la base de datos de ratones (proporcionada por Karen Svenson via Gary Churchill and Dan Gatti and partially funded by P50 GM070683). Importaremos los datos a R como ya se ha descrito en las guías anteriores y explicaremos a cerca de variables aleatorias y distribución nula (null) usando programación R.

Utilizaremos el set de datos contenido en el archivo `femaleMiceWeights.csv` en la carpeta `extdata`:

```
dir <- ("~/Documents/Proyectos/2017/12PastoWorkshop/CourseLab/ws/extdata/") #ruta a extdata
file <- paste0(dir,"femaleMiceWeights.csv")
dat <- read.csv(file)
```

Primer vistaso a los datos

Ahora estamos interesados en determinar si siguiendo una determinada dieta hace que los ratones engorden despues de varias semanas. Estos datos fueron producidos en "The Jackson Lab" usando 24 ratones a quienes se les asigno aleatoriamente una dieta chow (normal) o alta en grasas (hf). Despuesd de varias semanas, los cientificos pesaron cada raton z obtuvieron los datos (`head()` solo muestra las 6 primeras filas):

```
head(dat)
```

```
##   Diet Bodyweight
## 1 chow      21.51
## 2 chow      28.14
## 3 chow      24.04
## 4 chow      23.45
## 5 chow      23.68
## 6 chow      19.79
```

En RStudio, es posible ver todos el dataset con:

```
View(dat)
```

Ahora entonces; los ratones bajo "hf" son mas pesados?. El raton 24 (hf) con 20.73gr es uno de los más livianos, mientras que el raton 21 (hf) con 34.02gr es uno de los más pesados. Así que solo mirando los datos podemos ver que existe *variabilidad*. Pero como es el peso promedio de cada grupo?:

```
library(dplyr)
control <- filter(dat,Diet=="chow") %>% select(Bodyweight) %>% unlist
treatment <- filter(dat,Diet=="hf") %>% select(Bodyweight) %>% unlist
print( mean(treatment) )
```

```
## [1] 26.83417
```

```
print( mean(control) )
```

```
## [1] 23.81333
```

```
obsdiff <- mean(treatment) - mean(control)
print(obsdiff)
```

```
## [1] 3.020833
```

```
print(obsdiff*100/mean(control))
```

```
## [1] 12.68547
```

De esta manera los ratones bajo dieta hf son alrededor de 12% más pesados que aquellos bajo dieta chow. Y eso es todo? para que necesitaría el p-value e intervalos de confianza?. La razón es que los promedios son **variables aleatorias**, lo cual sugnifica que pueden tomar muchos valores ya que si repetimos el experimento con 24 nuevos ratones entonces tendremos nuevos promedios, y cada vez que repitamos el experimento obtendremos valores diferentes. a este tipo de comportamiento es lo que llamamos **variable aleatoria**

Variables aleatorias

Para explorar más a fondo el comportamiento de las variables aleatorias, imaginemos que tenemos el peso de todos los ratones control. En estadistica nos referimes a este conjunto como **la población**. Estos son todos los ratones controls a partir de los cales muestramos 24 para el ensayo anterior. Aunque en la realidad esto no sucede, para este ejercicio tenemos un dataset especial:

```
dir <- ("~/Documents/Proyectos/2017/12PastoWorkshop/CourseLab/ws/extdata/") #ruta a extdata
file <- paste0(dir,"femaleControlsPopulation.csv")
population <- read.csv(file)
## Usar unlist para convertirlo en un vector numérico
population <- unlist(population)
```

Ahora realicemos 3 muestras de 12 ratos aleatorios y observemos que valor tiene el promedio:

```
control <- sample(population,12)
mean(control)
```

```
## [1] 23.81333
```

```
control <- sample(population,12)
mean(control)
```

```
## [1] 23.77083
```

```
control <- sample(population,12)
mean(control)
```

```
## [1] 24.18667
```

Notece que el promedio varia. Nosotros podemos seguir repitiendo este ejercicio y comenzaremos a entender algo a cerca de la distribución de esta variable.

Hipótesis nula (Null Hypothesis)

Devolvamonos un momento al primer ejemplo sobre los cambios de dieta y al pormedio de la diferencia de pesos `obsdiff`. Como cientificos nosotros necesitamos ser escépticos. Así que debemos preguntarnos; Como sabemos que este valor `obsdiff` es debido a la dieta y no por casualidad?. Que pasa si nosotros les damos a los 24 ratones la misma dieta?, miraremos alguna diferencia así de grande como `obsdiff`. Los estadisticos se refieren a este escenario como la **hipótesis nula**. El nombre de nula se usa para recordarnos que nosotros actuamos como escépticos: dando la posibilidad de que no hay diferencia.

Dado que en este caso tenemos acceso a la población, es posible observar tantos valores como queramos para identificar la diferecia de los promedios cuano la dieta no tiene effecto. Podemos hacer esto tomando muestras aleatorias de 24 ratones control, dado que ellos están bajo la misma dieta, y entonces almacenar las diferencies en la media entre dos grupos de 12 ratos escogidos aleatoriamente. Así lo hariamos en R:

```
##12 ratones control
control <- sample(population,12)
##Los otros 12 ratones control que actuan como no lo fueran (falsos tratamiento)
treatment <- sample(population,12)
print(mean(treatment) - mean(control))
```

```
## [1] 0.6375
```

Ahora haremos este mismo ejercicio 10.000 veces. Para esto usaremos un "for-loop", una operación que nos permite automaizar un proceso.

```
n <- 10000
null <- vector("numeric",n)
for (i in 1:n) {
  control <- sample(population,12)
  treatment <- sample(population,12)
  null[i] <- mean(treatment) - mean(control)
}
```

Los valores almacenados en el vector `null` (arriba), son lo que llamamos la **distribución nula**. Ahora nosotros podemos calcular el porcentaje de entre los 10.000 valores aleatorios generados, cuales son mayores a `obsdiff`:

```
mean(null >= obsdiff)
```

```
## [1] 0.0151
```

Eso significa que solamente una pequeña proporción de las 10.000 simulaciones tuvieron valor mayor a `obsdiff`. Así que como escéptcios concluimos que cuando no hay efecto de la dieta, nosotros podemos ver una diferencia tan grande como la observada en `obsdiff` unicamente en el 1.5% de las veces. Y ésto es lo que conocemos como p-value.

Distributions

La manera más simple entender una **distribución** es pensarla como una descripción compacta de muchos números. Por ejemplo, suponer que uno mide la altura de todos los hombres de una población. Ahora imaginar que uno necesita describir estos numeros a unos aliengenas que nunca han visitado la tierra. Suponer que todas estas alturas estan contenidas en el siguiente dataset:

```
data(father.son,package="UsingR")
x <- father.son$fheight
```

Una aproximación para resumir estos numeros de manera simple es hacer una lista de ellos para que el alien los vea. A continuación seleccionaremos 10 alturas aleatorias de las 1.078:

```
round(sample(x,10),1) ##round permite redondear los decimales; en este caso a 1
```

```
##  [1] 66.4 67.4 64.9 62.9 69.2 71.4 69.2 65.9 65.3 70.2
```

Histograms

Los histogramas describen la información de una manera fácil de interpretar. Los histogramas nos musetras las proporciones de los valores en intervalos:

$$\Pr(a \leq x \leq b) = F(b) - F(a)$$

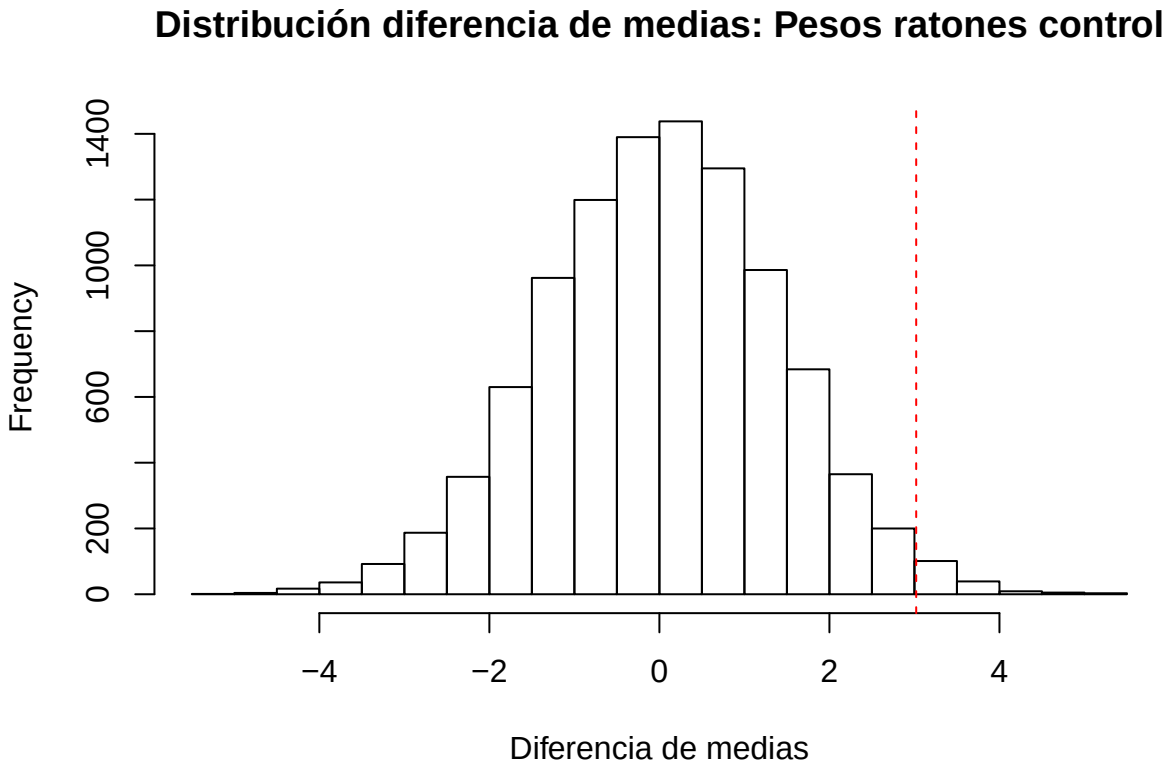
Este es el plot mas útil que podemos generar para explorar los datos, puesto que nos muestra los intervalos que toma nuestra valriable problema. Adicionalmente a través *histogramas* es posible distinguir diferentes tipos o familias de distribuciónes. Revisemos primero el histograma de alturas de nuestra población:

```
hist(x,xlab="Altura (en pulgadas)",main="Altura de hombres adultos")
```

Mostrando este plot a los aliens seguro será más informativo que mostrar solo números. Con este plot, es posible aproximar el número de individuos en cualquier intervalo de altura. Por ejemplo haz cerca de 70 individuos sobre 72 pulgadas de altura.

Previamente corrimos una simulación llamada **Monte Carlo** y obtuvimos 10.000 resultados de la variable aleatoria diferencia de medias bajo la hipótesis nula. Ahora podemos utilizar el histograma para visualizar el comportamiento de nuestra distribución nula y observar donde se ubican los valores menores a `obsdiff`:

```
hist(null,xlab="Diferencia de medias",main="Distribución diferencia de medias: Pesos ratones control")
abline(v=obsdiff,col="red",lty=2)
```



Ahora repitamos la misma simulación de monte carlo y la acoplamos al histograma para observar lo que esta sucediendo. Veremos entonces como la distribución nula se va formando con cada uno de los valores observados quevamos generando y quienes se van ubicando uno sobre el otro:

```
n <- 100
library(rafalib)
nullplot(-5,5,1,30, xlab="Frequencias observadas (grams)", ylab="Frequency")
totals <- vector("numeric",11)
for (i in 1:n) {
  control <- sample(population,12)
  treatment <- sample(population,12)
  nulldiff <- mean(treatment) - mean(control)
  j <- pmax(pmin(round(nulldiff)+6,11),1)
  totals[j] <- totals[j]+1
  text(j-6,totals[j],pch=15,round(nulldiff,1))
  if(i < 50) Sys.sleep(0.1) ## puede variar sleep para hacerlo más despacio
}
abline(v=obsdiff,col="red",lty=2)
```

Conociendo la distribución de frecuencias de una variable aleatoria podemos describir que tan probable es que un valor específico sea tomado por la variable aleatoria. Por ejemplo, sí nosotros elejimos un valor de diferencia de pesos aleatorio de nuestra lista, entonces la probabilidad de que mi variable aleatoria tome un valor entre a y b sera:

$$\Pr(a \leq X \leq b) = F(b) - F(a)$$

Notar que X esta ahora en mayusculas para distinguirlo como una variable aleatoria y que la ecuación de

arriba define la **distribución de probabilidades** de la variable aleatoria. Conociendo esta distribución es increiblemente util en ciencias. Por ejemplo, es posible calcular la probabilidad de observar un valor tan grande como alugno definido (por nosotros); concepto conocido como ***p-value***.

Un punto a tener en cuenta es que aquí se ha calculado $\Pr(a)$ a partir de una población simulando, pero en muchas casos no tenemos la información de la población para lo cual los matemáticos nos dan formulas de como es $\Pr(a)$, ahorrandonos el problema de computar la distribución como lo hicimos aquí. Un ejemplo de esta poderosa aproximación es la distribución normal.

Normal Distribution

Como vimos en el ejemplo de arriba, la distribución de probabilidades se aproxima a una que es muy común en la naturaleza: la curva de la campana, tambien conocida como distribución normal o Gaussiana. Cuando el histograma de una lista de nurmeros se aproxima a la distribución normal, es posible y conveniente usar la formula matemática de la distribución normal para aproximar los valores de proporcion (o outcomes) a cualquier intervalo dado:

$$\Pr(a < x < b) = \int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) dx$$

Aunque la formula se ve intimidante, nunca tendremos que escribirla, dado que en R esta almacenada como la función `pnorm` donde "**a**" esta definido como $-\infty$, y "**b**" sera el argumento definido por el usuario.

Aquí "μ" y "σ" hacen referencia a la media y la desviación estandar de la población. Si nuestra lista de valores siguen una **distribución normal**, entonces los valores de la media y varianza de la población pueden ser usados en la formula de arriba.

Para el ejemplo del peso de los ratones, calculamos que solamente el 1.5% (0.0151) de los valores de la hipótesis nula serían mayores que `obsdiff` (3.02gr). Ahora es posible calcular la proporción de valores menor al valor 'x' con `pnorm(x,mu,sigma)`. Que tal funciona la aproximación normal?:

```
1 - pnorm(obsdiff,mean(null),sd(null))
```

```
## [1] 0.01468484
```