Statistics with R

February 7, 2019

Contents

Para que sirve la estadística y la teoría de la probabilidad?

Básicamente, necesitamos la estadística y la teoría de la probabilidad para extraer conclusiones generales acerca de procesos para los que sólo podemos medir una pequeña muestra. Por ejemplo, utilizando los datos del ejercicios de alturas que realizamos en la clase la semana pasada podríamos intentar contestar a la pregunta: Son los hombres mas altos en media que las mujeres? Para contestar a esto, sólo tenemos una muestra, de ahí que tengamos que utilizar conocimiento previo construído durante siglos para definir como de extrapolables son nuestras conclusiones sobre una pequeña muestra a la población general.

Sin embargo, los nuevos tiempos en los que grandes cantidades de datos son medibles hacen que se produzca un cambio de paradigma: cada vez va a ser más común utilizar los propios datos para realizar inferencias acerca de toda la población en lugar de confiar en las propiedades formales o teóricas de las distribuciones de probabilidad. Por eso en la clase de hoy hablaremos de distribuciones de probabilidad pero también de **distribuciones empíricas** que van a ser a menudo utilizadas por vosotros durante vuestra carrera como data scientists.

Procesos aleatorios o estocásticos: Incertidumbre no significa caos

Cualquier suceso cuyo resultado no pueda ser predicho con total seguridad es un proceso estocástico. El resultado cuantitativo de un proceso estocástico es una *variable aleatoria*. El hecho de que el resultado de un proceso tenga una componente estocástica, de incertidumbre no significa que sea un proceso totalmente

azaroso: se puede modelar la incertidumbre y controlarla. **Este es uno de los principales trabajos de un data scientist: cuantificar el azar, la estocasicidad. La inferencia estadística propone un marco teórico para hacerlo**

Por ejemplo, si queremos medir un segmento usando una regla, no siempre obtendremos exactamente el mismo valor si queremos la precisión en micras, debido a la incertidumbre propia de la medición (la regla puede estar un poco más horizontal o un poco menos, el que mide puede redondear hacia un valor o hacia otro...). Esto no quiere decir que no sepamos mas o menos cual es el tamaño del segmento. Otro proceso aleatorio sería si voy a obtener cara o cruz al lanzar una moneda al aire.

Un proceso aleatorio puede ser de dos tipos dependiendo de lo que está intentando medir: una variable **continua** o una **categórica/discreta**. La medida del segmento es una variable aleatoria continua y tirar la moneda al aire es una variable aleatoria categórica.

Ejercicio 1: Ejemplos de sucesos de nuestro alrededor que pueden modelarse con variables aleatorias continuas y con discretas

Errores sistemáticos y errores aleatorios en los datos

- Errores sistemáticos son aquellos que se pueden estimar y posteriormente transformar los datos para eliminarlos.Por ejemplo, si la regla con la que medimos nuestro segmento está mal calibrada podemos calibrarla y corregir nuestras medidas con estos valores
- Errores aleatorios son aquellos introducidos de manera aleatoria en nuestro datos. No podemos estimarlos con total certeza pero puede que sepamos como modelarlos. Por ejemplo, los errores de medicion de los que hablábamos antes siguen una distribución normal, que en ocasiones se denomina "ruido blanco". Otra fuente importante de error aleatorio que podemos modelar y tener en cuenta en nuestros datos es el error de muestreo.

Población y muestras aleatorias: simulaciones de Monte Carlo

En general, con la inferencia estadística intentamos llegar a conclusiones generales a partir de una muestra de datos limitada. Esto introduce una incertidumbre en nuestras conclusiones (qué pasaría si en lugar de haber llamado a estas 1000 casas para preguntar su intención de voto hubiera llamado a otras 1000?). Pero esta incertidumbre se puede modelar y tener en cuenta en nuestra toma de decisiones.

Vamos a simular una situacion que es la que querremos descubrir sin ver a partid de muestras aleatorias. La urna contiene 2 bolas rojas y 3 azules.

```
beads <- rep( c("red", "blue"), times = c(2,3))
beads
```

```
## [1] "red"  "red"  "blue" "blue" "blue"
```

De esta urna (es nuestro todo, nuestro *universo*) escogemos 1 bola al azar. Sabemos la probabilidad de que sea roja (2/5=0.4) y la probabilidad de que sea azul (3/5=0.6). Usamos la función *sample()* que tiene en cuenta estas probabilidades

```
sample(beads,1)
```

```
## [1] "blue"
```

Intuímos que si repetimos este experimento (extraer una bola al azar de la urna) muchas veces llegaremos a descubrir cuantas bolas de cada color hay en la urna. Usamos la función *replicate()* que repite N veces la función que le demos, en este caso *sample()*. Generamos una muestra aleatoria de 1000 elementos con la función de probabilidad subyacente:

```
B<-1000
realizaciones<-replicate(B,sample(beads,1))
table(realizaciones)
```

```
## realizaciones
## blue  red
##  579  421
```

```
prop.table(table(realizaciones))
```

```
## realizaciones
##  blue   red
## 0.579 0.421
```

Habriamos acertado tanto con solo 10 realizaciones?

```
B<-10
realizaciones<-replicate(B,sample(beads,1))
table(realizaciones)
```

```
## realizaciones
## blue  red
##    8    2
```

```
prop.table(table(realizaciones))
```

```
## realizaciones
## blue  red
##  0.8  0.2
```

Generamos 100 muestras de tamaño 10 a ver cómo de consistentes son los resultados

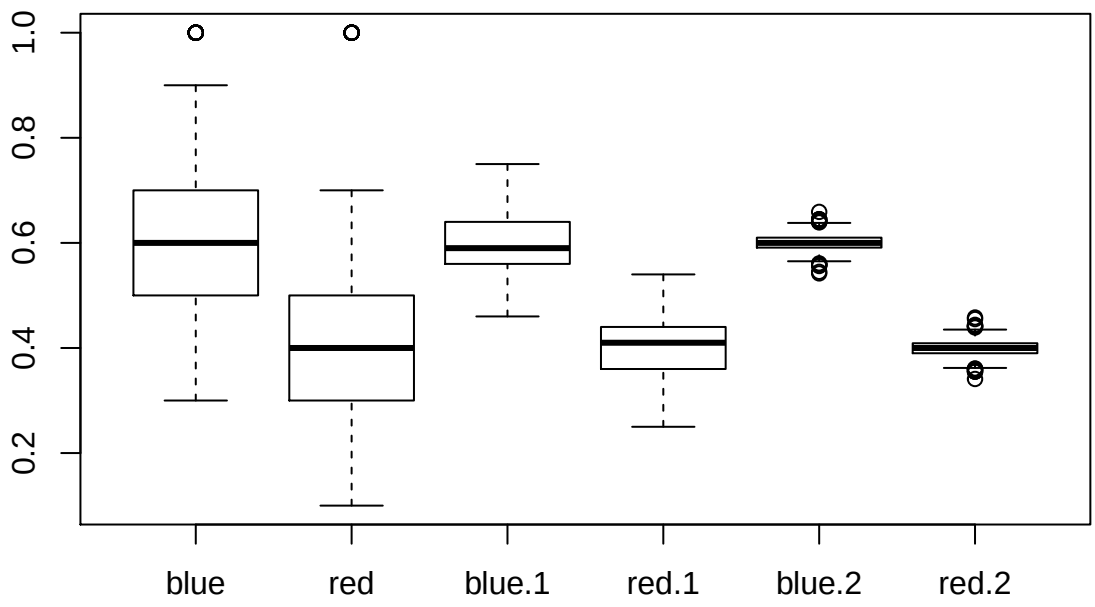
```
B<-10
x10<-numeric()
for (i in 1:100){
  realizaciones<-replicate(B,sample(beads,1))
  x10<-rbind(x10,prop.table(table(realizaciones)))
}
```

Hacemos lo mismo con muestras de 100 y de 1000 elementos

```
B<-100
x100<-numeric()
for (i in 1:100){
  realizaciones<-replicate(B,sample(beads,1))
  x100<-rbind(x100,prop.table(table(realizaciones)))
}

B<-1000
x1000<-numeric()
for (i in 1:1000){
  realizaciones<-replicate(B,sample(beads,1))
  x1000<-rbind(x1000,prop.table(table(realizaciones)))
}
```

Representamos las tres muestras de tamaños 10,100 y 1000 con un boxplot

```
boxplot(data.frame(x10,x100,x1000))
```



El tamaño muestral nos aproxima a la verdad y minimiza la incertidumbre que se deriva del sampleado que es intrínseco a cualquier disciplina en la que usamos data science: banca, seguros, medicina, aviónica, sociología...

Probabilidad discreta: Procesos estocásticos categóricos

Distribución Binomial

Uno de los experimentos categóricos más sencillos es la loteria de navidad. En un bombo hay (digamos) 100.000 bolas cada una con un número distinto.

Podemos calcular la probabilidad de extraer el número 000.000 $\Pr(000.000) = \frac{1}{100.000}$ exactamente igual que la probabilidad de escoger cualquiera de los otros 99.999 números del bombo

Hemos introducido dos conceptos:

- La variable aleatoria *Me tocará la loteria?* tiene dos posibles respuestas: Si (1) o No (0)
- Su función de probabilidad es

$$\Pr(Ganarlaloteria) = \begin{cases} \frac{1}{100.000} & si \quad X = 1 \\ \frac{99.999}{100.000} & si \quad X = 0 \end{cases}$$

Está variable aleatoria se llama Bernouilli y es una de las distribuciones que siguen más sucesos de la vida. La probabilidad de acierto en una Bernouilli es un número p entre 0 y 1 y la probabilidad del contrario (no sacar una bola roja) es *1-p. Esa es una de las características de una función de probabilidad: la suma de las probabilidades de los posibles outcomes de la variable es 1.* Así podemos modelar la probabilidad de que

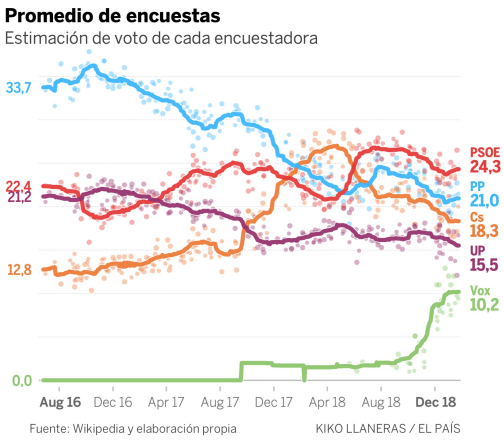


Figure 1: Fuente: CIS/ElPais

alguien devuelva un préstamo a un banco o no, la probabilidad de que alguien tenga una enfermedad o no o, como decíamos al principio, la probabilidad de que al lanzar una moneda al aire obtenga una cara o una cruz.

Si en realidad lo que queremos es saber cual es la probabilidad de que me toque la loteria si tengo 10 décimos y van a sacar 1000 bolas, tenemos una distribución *binomial*. La probabilidad de lo anterior se calcula como:

```
dbinom(10,1000,10^-5)
```

```
## [1] 2.608147e-27
```

donde p=1/100.000, x=10, n=1000

En el ejercicio anterior, la variable *Sacar una bola roja* sería un proceso de Bernouilli. Y la respuesta a la pregunta: *Si saco 2 bolas al azar, cual es la probabilidad de que las 2 sean rojas?* Seria una Binomial.

Ejercicio: Calcula esa probabilidad

Distribución Multinomial

Si tenemos varios posibles elementos entre los que elegir, no sólo dos, se trata de una distribución multinomial. Por ejemplo, en unas elecciones podemos elegir entre varios candidatos. La respuesta a la pregunta: *Qué partido ganará las elecciones* ya no es discotómica, sino que puede haber 4-5 respuestas. Supongamos que el CIS ha llamado a 100 personas y les ha preguntado por su intención de voto. Según el último CIS (Figure 1) el 24,3 de los encuestados votará al PSOE, el 21% al PP, 18.3 a ciudadanos y 15.5 a UP. Es decir, la función de distribución de la variable aleatoria: ganador de las elecciones será:

$$\Pr(Ganarelecciones) = \begin{cases} 0.243 & si \quad X = PSOE \\ 0.21 & si \quad X = PP \\ 0.183 & si \quad X = Ciudadanos \\ 0.155 & si \quad X = UP \end{cases}$$

Véis que en el momento en el que transformamos el lenguaje en números ya no hablamos de procesos aleatorios sino de variables aleatorias.

Distribución Hipergeométrica

Otra distribución de datos categóricos interesante es la hipergeométrica. Este tipo de distribución se utiliza cuando queremos saber la probabilidad de que haya un número determinado de positivos entre un conjunto predefinido. La diferencia con la binomial es que en la binomial cada suceso es independiente del anterior porque hay reemplazamiento, mientras que en la hipergeométrica no.

Un ejemplo serían los tests que se hacen para comprobar los resultados de las elecciones. Tomando una muestra de 10 individuos de una ciudad en la que sabemos que hay 80 hombres y 20 de mujeres, cual es la probabilidad de que en nuestra muestra haya 5 mujeres?

Distribución de Poisson

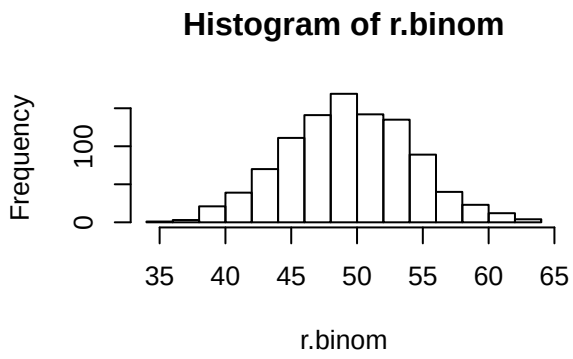
Cuando tenemos una variable que cuenta el número de eventos que suceden en un intervalo de tiempo, esos datos se distribuyen como una Poisson. Por ejemplo, cuántos coches llegan a un semáforo desde que se enciende hasta que se apaga.

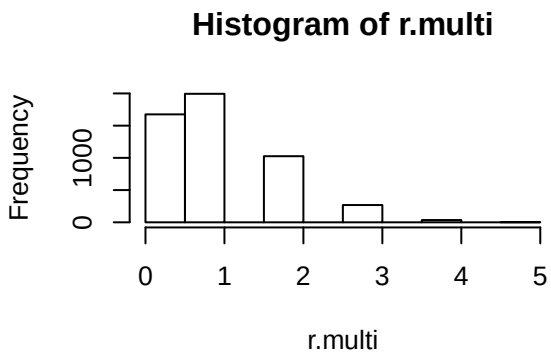
Funcion de probabilidad de estas variables

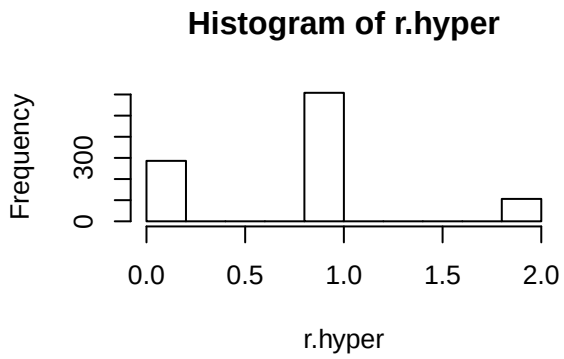
En R, podemos generar muestras aleatorias del tamaño que queramos para todas estas variables discretas.

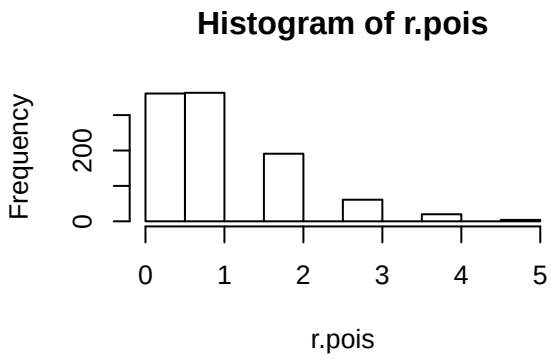
```
#rbinom(n, size, prob)

r.binom<-rbinom(1000,100,0.5)

#rmultinom(n, size, prob)
r.multi<-rmultinom(1000, 5, rep(0.2,5))

#rhyper(nn, m, n, k)
r.hyper<-rhyper(1000,2,3,2)

#rpois(n, lambda)
r.pois<-rpois(1000,1)
```

```
par(mfrow=c(2,2))
hist(r.binom)
hist(r.multi)
hist(r.hyper)
hist(r.pois)
```









Estas son las funciones de probabilidad. Otro tipo de funciones importantes son las de distribucion que nos ayudan a modelar eventos. Por ejemplo, para una binomial un evento seria: Cual es la probabilidad de que obtenga entre 2 y 3 aciertos? Para una Poisson seria: Cual es la probabilidad de que lleguen mas de 2 coches al semáforo desde que se abrio?

Las funciones de distribución siempre tienen una forma parecida:

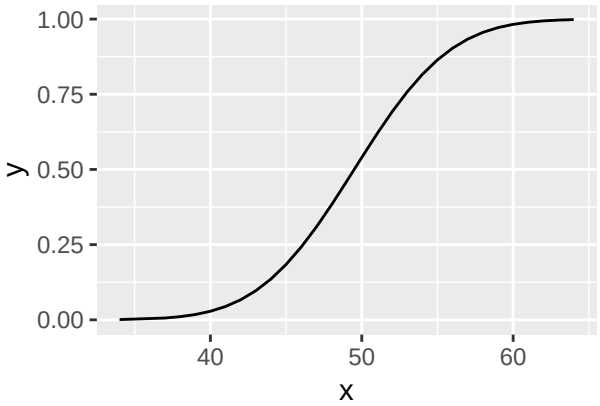
```
library(ggplot2)
```

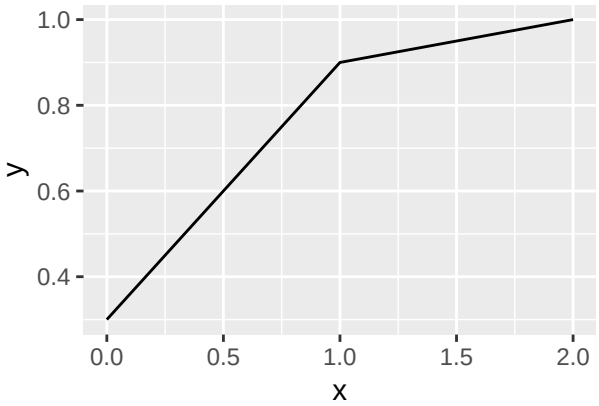
```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

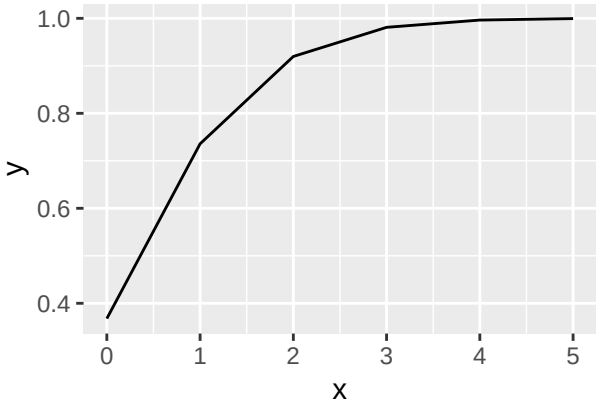
```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.2
```

```
bin<-data.frame(x=r.binom,y=pbinom(r.binom,100,0.5))
g1<-ggplot(bin, aes(x, y)) + geom_line()


hyper<-data.frame(x=r.hyper,y=phyper(r.hyper,2,3,2))
g2<-ggplot(hyper, aes(x, y)) + geom_line()


poisson<-data.frame(x=r.pois,y=ppois(r.pois,1))
g3<-ggplot(poisson, aes(x, y)) + geom_line()

grid.arrange(g1, g2, g3,nrow = 2)
```







Ejemplo: Problema del cumpleaños

Supongamos que los participantes de este curso habeis sido elegidos al azar de toda la poblacion. Sois 15 personas. Cual es la probabilidad de que no se repita ningun cumpleaños? (Asumimos que nadie nacio el 29/02)

Tomamos una muestra de tamaño 15 de los números 1 al 365

```
any(duplicated(bdays))
```

```
## [1] FALSE
```

Vamos a estimar esta probabilidad repitiendo lo mismo muchas veces:

```
same_birthday <- function(n){
  bdays <- sample(1:365, n, replace=TRUE)
  any(duplicated(bdays))
}
B <- 10000
results <- replicate(B, same_birthday(15))
mean(results)
```

```
## [1] 0.2536
```

Que pasaría si lo estuviéramos haciendo para un grupo de 50 personas?

```
B <- 10000
results <- replicate(B, same_birthday(50))
mean(results)
```

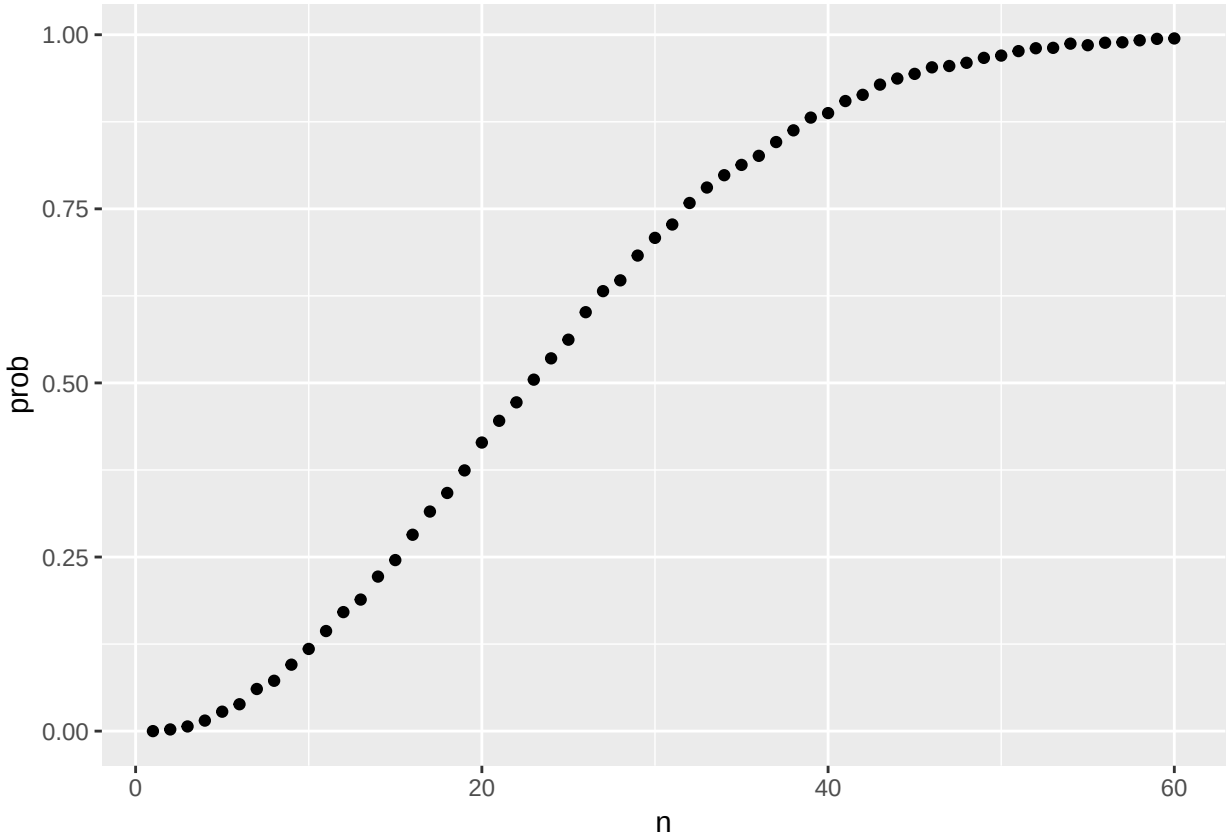
```
## [1] 0.97
```

Tendemos a subestimar la probabilidad. Pero claro, si tuvieramos un grupo de casi 365 individuos si que esperariamos esta probabilidad ser tan alta:

```
compute_prob <- function(n, B=10000){
  results <- replicate(B, same_birthday(n))
  mean(results)
}

n <- seq(1,60)
prob <- sapply(n, compute_prob)
```

Y podemos plotear la probabilidad de que dos personas tengan el mismo cumpleaños en un grupo de tamaño n:

```
prob <- sapply(n, compute_prob)
qplot(n, prob)
```



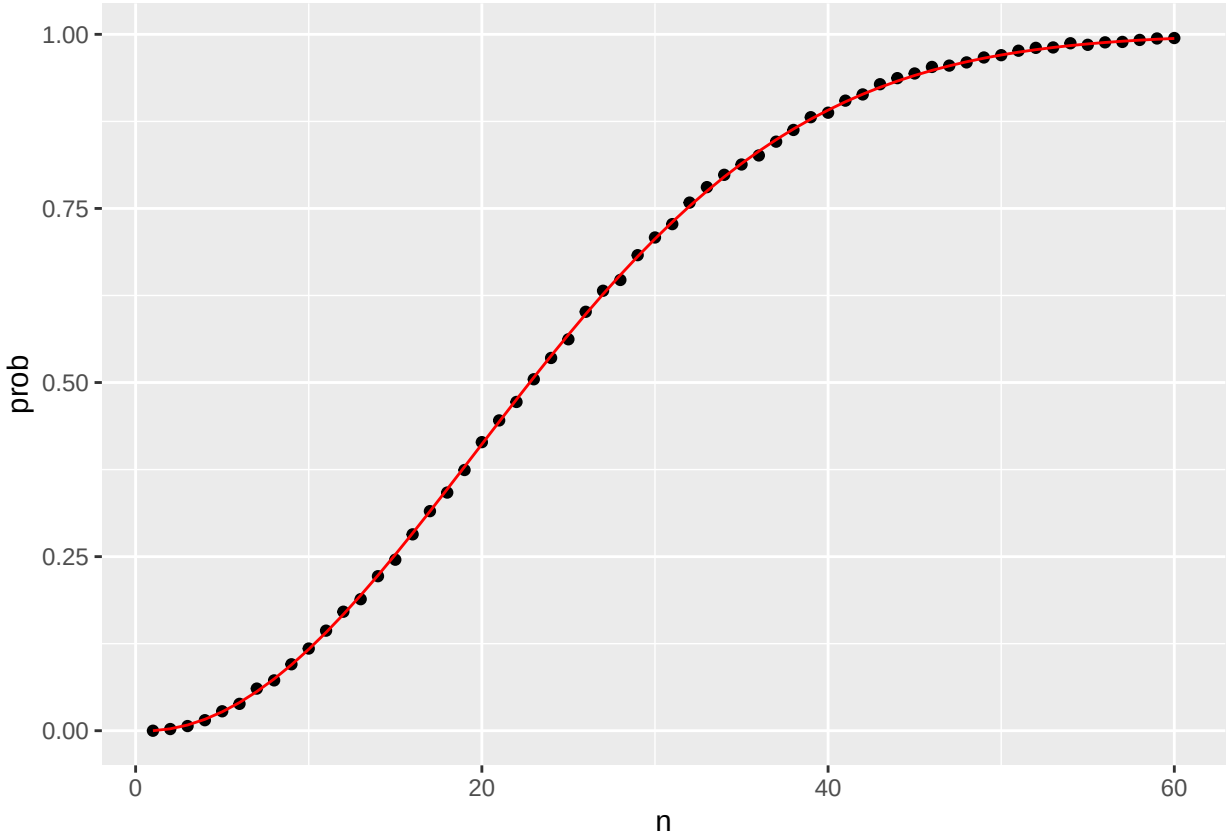
Cómo calcularíamos esta probabilidad analíticamente?

Calculamos la probabilidad de que NO suceda. Empezamos con la primera persona, cual es la probabilidad de que tenga un cumpleaños único? 1. EL siguiente... 364/365, el siguiente 363/365...

$$1 \times \frac{364}{365} \times \frac{363}{365} \dots \frac{365-n+1}{365}$$

Escribimos una función para simplificar:

```r
exact_prob <- function(n){
  prob_unique <- seq(365,365-n+1)/365
  1 - prod( prob_unique)
}
eprob <- sapply(n, exact_prob)
qplot(n, prob) +
  geom_line(aes(n, eprob), col = "red")
```

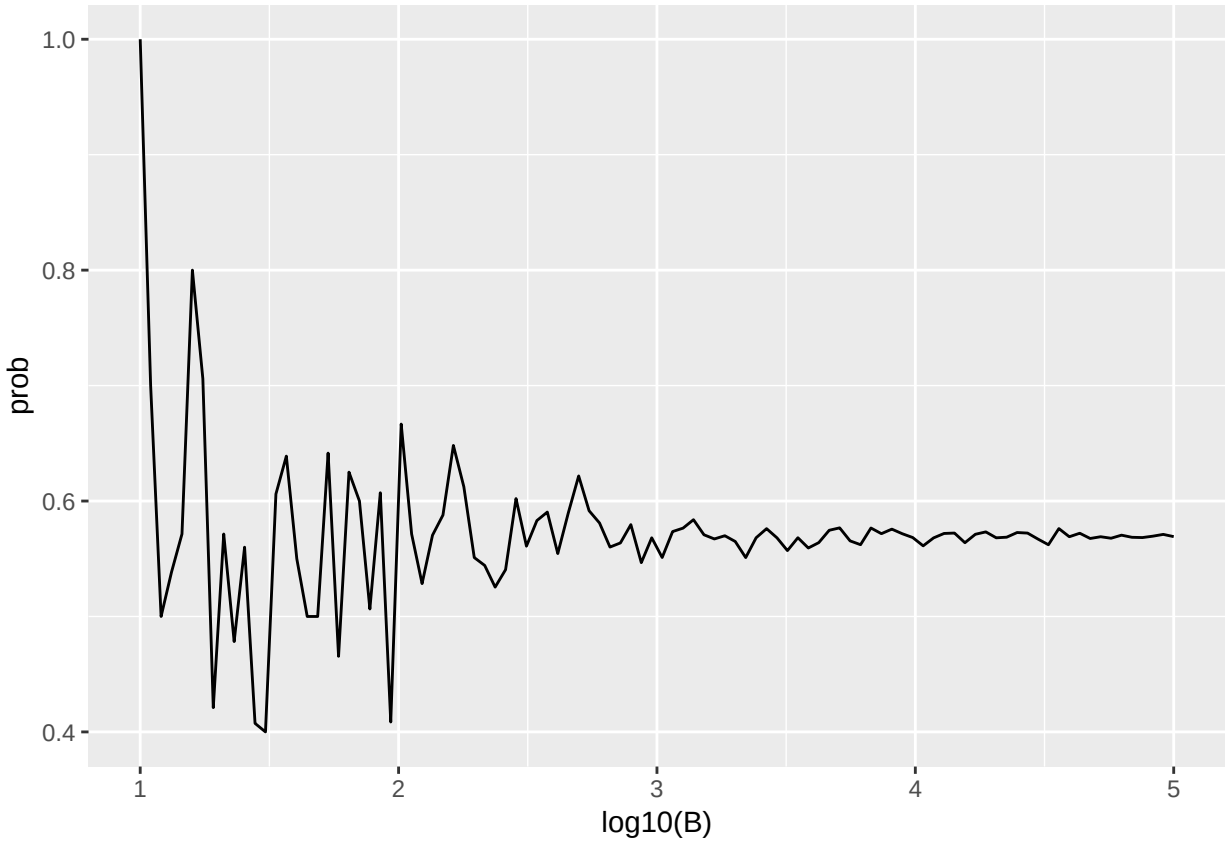


Lo que prueba que usando simulaciones hemos aproximado muy bien la probabilidad exacta.

Cuántos experimentos de Monte Carlo son suficientes?

Se trata de tener alguna forma de medir cuántas simulaciones de monte carlo necesitamos para aproximar bien el problema. 10.000 puede ser poco en algunos casos complejos o demasiado en terminos computacionales en algunas situaciones. Lo que debemos hacer es estimar la estabilidad del estimador, es decir, en que punto empieza a variar muy poco.

```r
B <- 10^seq(1, 5, len = 100)
compute_prob <- function(B, n=25){
  same_day <- replicate(B, same_birthday(n))
  mean(same_day)
}
prob <- sapply(B, compute_prob)
qplot(log10(B), prob, geom = "line")
```



Procesos aleatorios continuos

Si lo que estamos intentando modelar puede tomar cualquier valor (dentro de un rango) y no necesita ser seleccionado de un grupo de niveles, estamos ante una variable continua. Todas las variables que miden experimentalmente algo son variables continuas debido al ruido de las mediciones del que hablábamos antes. La distribución normal es la más conocida de las distribuciones de probabilidad continuas.

Función de densidad, función de distribución y distribución empírica

A diferencia de lo que ocurre con las variables aleatorias categóricas para las que podemos estimar la probabilidad de un outcome en concreto (Probabilida de obtener cara al lanzar una moneda=1/2) con las funciones continuas no podemos hacerlo. Trabajamos siempre con lo que hemos llamado "eventos": *Cuál es la probabilidad de que obtener un valor mayor de 5 al medir un segmento de longitud 4.9?* Para responder a estas preguntas necesitamos, al igual que para las variables categóricas, la función de distribución.

Vamos a trabajar primero sólo con nuestra muestra. Usando estos datos, no generales, podemos ver cómo es nuestra función de distribución **empírica**. Al igual que para las variables categóricas, la función de distribución de una variable aleatoria continua es la suma (o la integral) de las probabilidades de los valores menores o iguales a ella.

```
## -- Attaching packages ---------------------------------------------------------------------------
## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.2     v dplyr   0.7.8
## v readr   1.1.1     v stringr 1.3.1
## v tibble  1.4.2     v forcats 0.3.0
## -- Conflicts ------------------------------------------------------------------------------ tidy
```

```
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
## [1] 0.6231527
```

```
## [1] 0.3768473
```

```
## [1] 0.3743842
```

(Breve) Historia de la distribución normal

La variable aleatoria más sencilla en la que podemos pensar es la binomial. Si os fijáis en su histograma (que nos da su función de probabilidad) **??** cuando tenemos muchas realizaciones se aproxima a una curva *normal*. Esto es lo que formalizó de Moivre en 1733: intentó encontrar esa curva que hoy llamamos Gaussiana en honor a Gauss que planteó la fórmula en 1809. Mucho antes de esto, en el siglo XVII, Galileo ya planteó que los errores en las mediciones astronómicas eran simétricos respecto al verdadero valor a medir y que errores pequeños tendían a suceder mucho más frecuentemente que errores grandes.

De todo esto surge la formalización de una curva gaussiana que conocemos hoy en dia:

$$\Pr(X \leq b) = \int_{-\infty}^{b} \frac{1}{\sigma\sqrt{2*\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx$$

Esta es la forma analítica de una distribución normal. Lo que integramos es la función de densidad, el equivalente a una función de probabilidad (o masa) para una variable categórica. En R, al igual que hemos visto con las variables categóricas tenemos funciones para generar muestras aleatorias de números que sigan esta distribución de probabilidad. La bondad de tener fórmulas cerradas estudiadas durante muchos siglos es que podemos responder a la pregunta anterior sin necesidad de tener muestras, simplemente sabiendo que tipo de distribución siguen nuestros datos. En ejemplo anterior, si estamos trabajando con la variable aleatoria que modela la altura de un grupo de chicos y sabemos que en barcelona la media de altura de los hombres es de 1.70 con una desviación estandard de +/- 5 cm, podemos responder a la pregunta: *Cuál es la probabilidad de que un chico de barcelona mida más de 2m? Y de que mida más de 1.70?*

```
1-pnorm(2,1.70,0.05)
```

```
## [1] 9.865877e-10
```

```
1-pnorm(1.80,1.70,0.05)
```

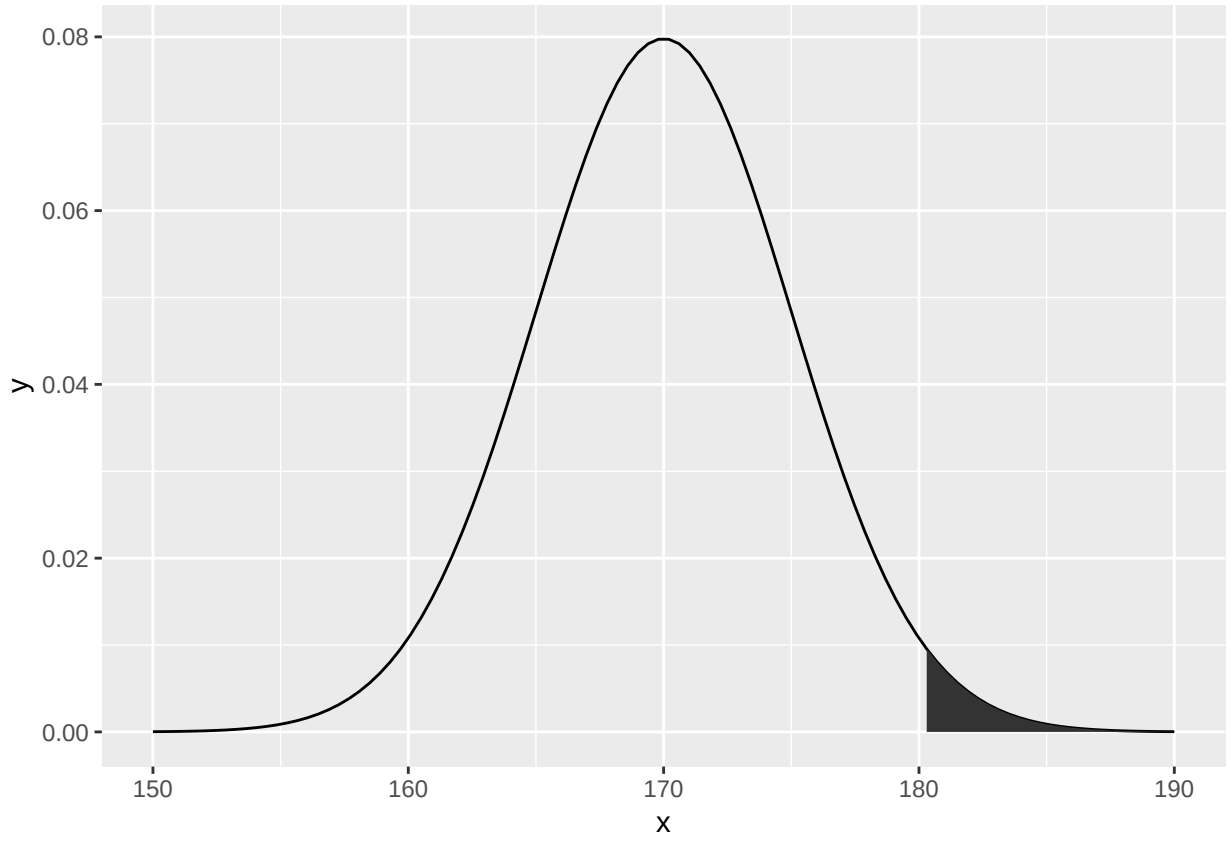
```
## [1] 0.02275013
```

```
1-pnorm(1.70,1.70,0.05)
```

```
## [1] 0.5
```

De visu, esta sería la forma de la distribución de densidad de esta normal:

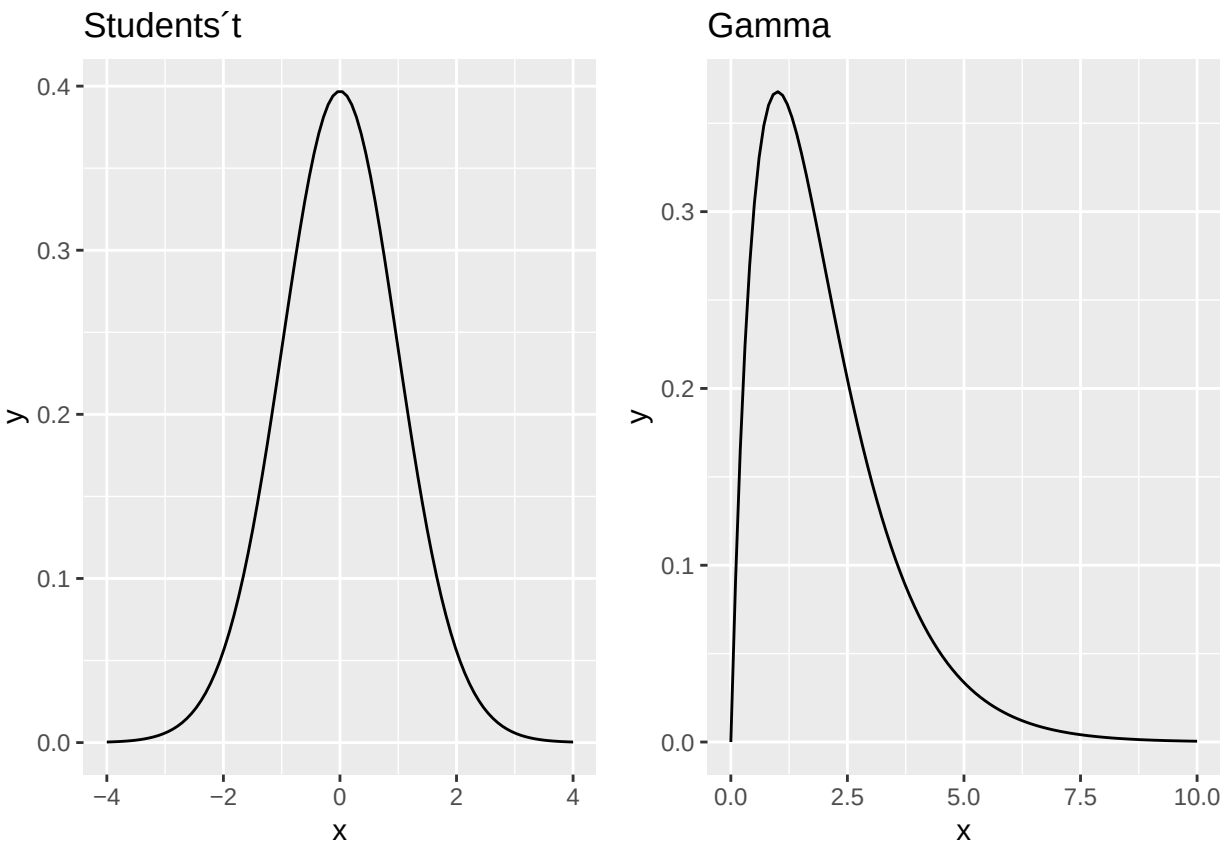
```
# generamos números aleatorios de una normal con media 170 cm y std=0.05
s<-5
m<-170
dat <- data_frame(x = seq(-4, 4, length=100) * s + m,
                  y = dnorm(x, m, s))
dat_ribbon <- filter(dat, x >= 2 * s + m)
ggplot(dat, aes(x, y)) +
  geom_line() +
  geom_ribbon(aes(ymin = 0, ymax = y), data = dat_ribbon)
```



Otras funciones de distribución continuas

La t de Student, la gamma, beta, Chi, exponencial, beta-binomial. . . son también funciones continuas con sus formas analíticas de la función de densidad y distribución.

```
dat.t <- data_frame(x = seq(-4, 4, length=100),
                    y = dt(x, 50))
gt<-ggplot(dat.t, aes(x, y)) +geom_line() +ggtitle("Students´t")

dat.g <- data_frame(x = seq(0, 10, length=100),
                    y = dgamma(x, 2))
gg<-ggplot(dat.g, aes(x, y)) +geom_line()+ggtitle("Gamma")

grid.arrange(gt, gg, nrow = 1)
```



Ejercicios:

1. Si sabemos que la distribución de las alturas de las chicas de barcelona es normal con media 160 cm y desviación standard 5 cm, cuál es la probabilidad de que una chica mida más de 180 cm? Y que mida menos de 150 cm?
2. Cuál es la probabilidad de que una chica mida entre 150 y 180 cm?
3. Cómo de alta sería la chica que ocupa el percentil 99? (qnorm)
4. La distribución del coeficiente intelectual es también normal con media 100 y desviación estandard 15. Quieres saber el IQ de la persona de tu barrio (10.000 habitantes) con mayor coeficiente intelectual. Genera una simulación de Monte Carlo con B=1000 cada una con 10^6 de IQ scores y guarda el mayor. Represéntalo en un histograma.

Inferencia estadística: Caso práctico

Como data scientists vamos a tener que trabajar frecuentemente con procesos estocásticos. En realidad, todos los procesos son estocásticos pero a veces optaremos por intentar minimizar la incertidumbre antes de trabajar los datos y utilizaremos modelos deterministas despues para su modelado. Pero, en general, utilizar modelos probabilisticos es una buena idea porque podremos hacerlo todo a la vez: cuantificar y ajustar por esa incertidumbre y extraer conclusiones robustas y generalizables a partir de muestras de datos.

Una de las principales tareas de un data scientist trata de estimar parámetros relevantes para cada dataset y compararlos entre distintos datasets. De estas dos tareas se encarga la inferencia estadística.

Vamos a trabajar con una caso de una publicación real. Se trata de investigar el efecto de una dieta rica en grasa en el peso y el desarrollo de diabetes en una cepa concreta de ratones. Este es uno de los primeros

estudios en los que se probó la relación entre una dieta grasa y la aparición de diabetes.

EDA (Exploratory Data Analysis)

Cargamos nuestras librerias de R:

```
library(dplyr)
library(lattice)
library(beeswarm)
library(ggplot2)
```

Leemos el dataset mice_pheno.csv. Se trata de un data frame con medidas de 846 ratones, 449 con dieta control y 397 en dieta High-Fat (HF). Al ser ratones, podemos imaginarnos que estamos observando toda la población de ratones.

```
setwd("C:/Users/fscabo/Desktop/MasterDataScience_KSchool/Session5_Statistics&R/Data")
pheno=read.csv("mice_pheno.csv")
View(pheno)

class(pheno)
```

```
## [1] "data.frame"
```

```
which(pheno[,2]=="hf")
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  76  77  78  79  80  81  82  83  84
##  [35]  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101
##  [52] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
##  [69] 144 145 146 147 148 149 150 176 177 178 179 180 181 182 183 184 185
##  [86] 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 226 227
## [103] 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244
## [120] 245 246 247 248 249 250 276 277 278 279 280 281 282 283 284 285 286
## [137] 287 288 289 290 291 292 293 294 295 296 297 298 299 300 351 352 353
## [154] 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370
## [171] 371 372 373 374 375 401 402 403 404 405 406 407 408 409 410 411 412
## [188] 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429
## [205] 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446
## [222] 447 448 449 500 501 502 503 504 505 506 507 508 509 510 511 512 513
## [239] 514 515 516 517 518 519 520 521 522 523 549 550 551 552 553 554 555
## [256] 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572
## [273] 573 574 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614
## [290] 615 616 617 618 619 620 621 622 623 624 649 650 651 652 653 654 655
## [307] 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 697
## [324] 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714
## [341] 715 716 717 718 719 720 721 772 773 774 775 776 777 778 779 780 781
## [358] 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 822 823
## [375] 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840
## [392] 841 842 843 844 845 846
```

```
length(which(pheno[,2]=="hf"))
```

```
## [1] 397
```

```
length(which(pheno[,2]=="chow"))
```

```
## [1] 449
```

```
hf.data=pheno[which(pheno[,2]=="hf"),]
chow.data=pheno[which(pheno[,2]=="chow"),]

summary(hf.data[,3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   15.97   25.14   29.92   30.48   35.40   54.08       4
```

```
summary(chow.data[,3])
```

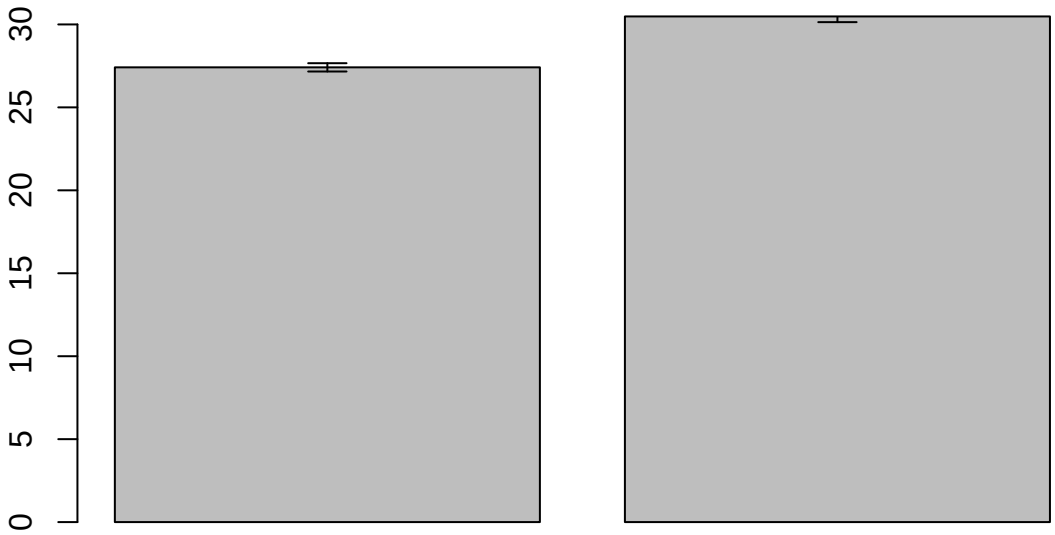
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   15.51   23.38   26.97   27.41   31.28   46.71       1
```

Lo primero que tenemos que hacer siempre es explorar los datos: Exploratory Data Analysis (EDA) que es lo que hicimos en las sesiones anteriores.

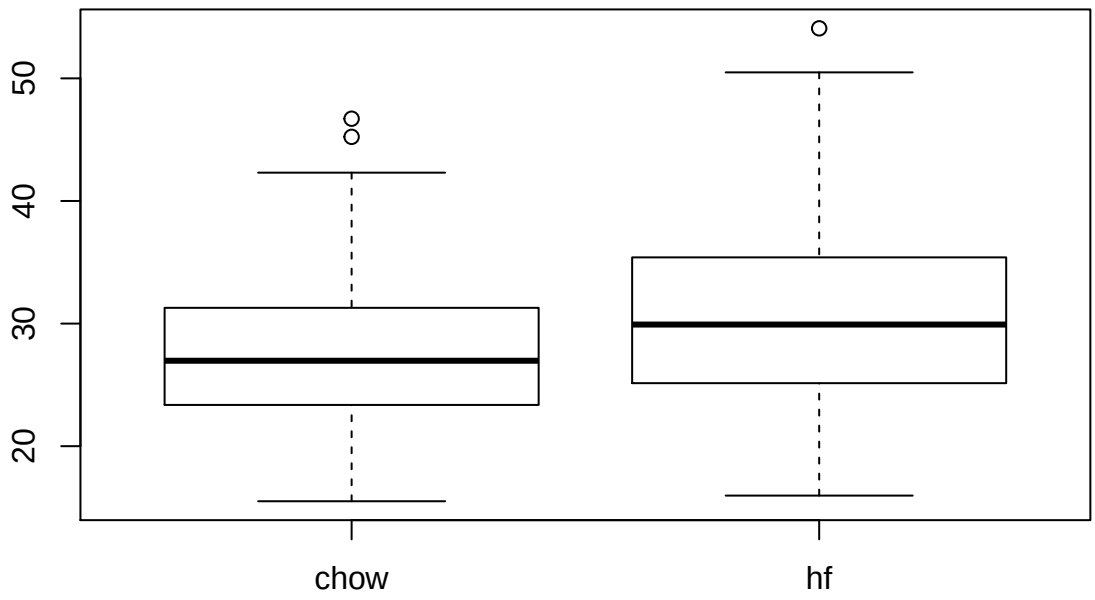
```
#several representations
? barplot
```

```
## starting httpd help server ... done
```

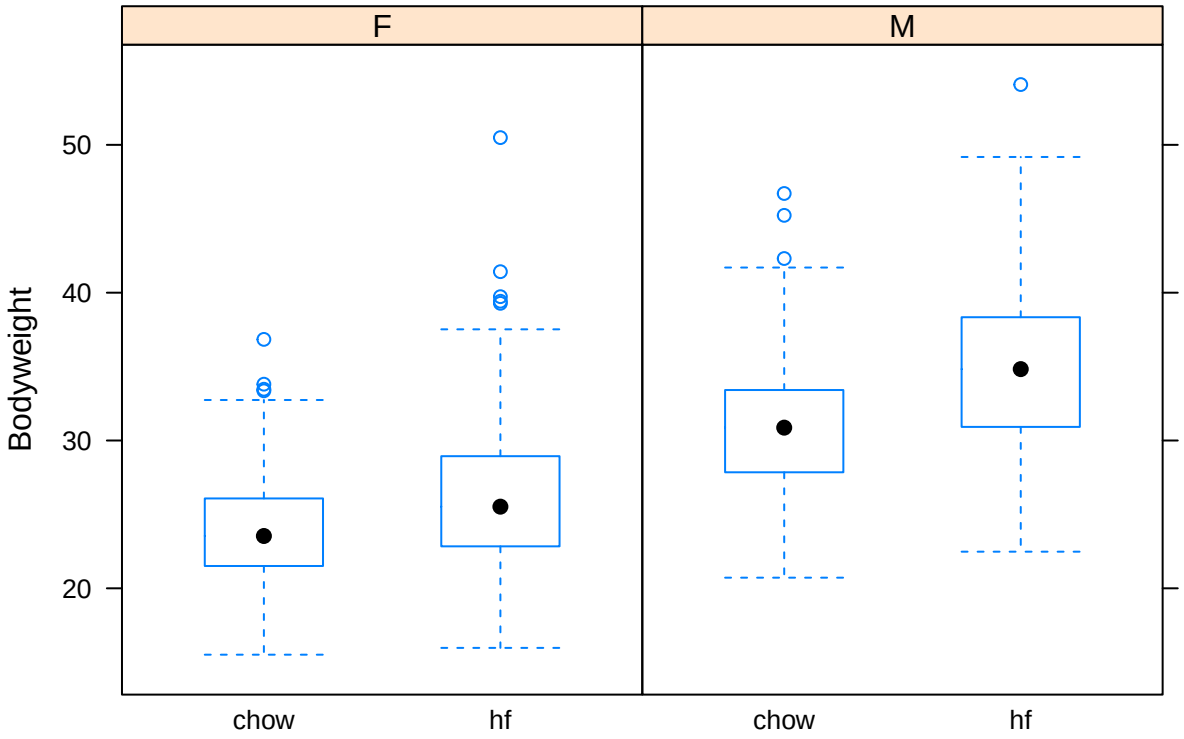
```
error.bar <- function(x, y, upper, lower=upper, length=0.1,...){
  if(length(x) != length(y) | length(y) !=length(lower) | length(lower) != length(upper))
    stop("vectors must be same length")
  arrows(x,y+upper, x, y-lower, angle=90, code=3, length=length, ...)
}

y.means=c(mean(chow.data[,3],na.rm=T),mean(hf.data[,3],na.rm=T))
y.sd=c(sqrt(var(chow.data[,3],na.rm=T)),sqrt(var(hf.data[,3],na.rm=T)))
y.se=y.sd/sqrt(c(length(chow.data[,3]),length(hf.data[,3])))
X11()
barx=barplot(y.means)
error.bar(barx,y.means, y.se)
```



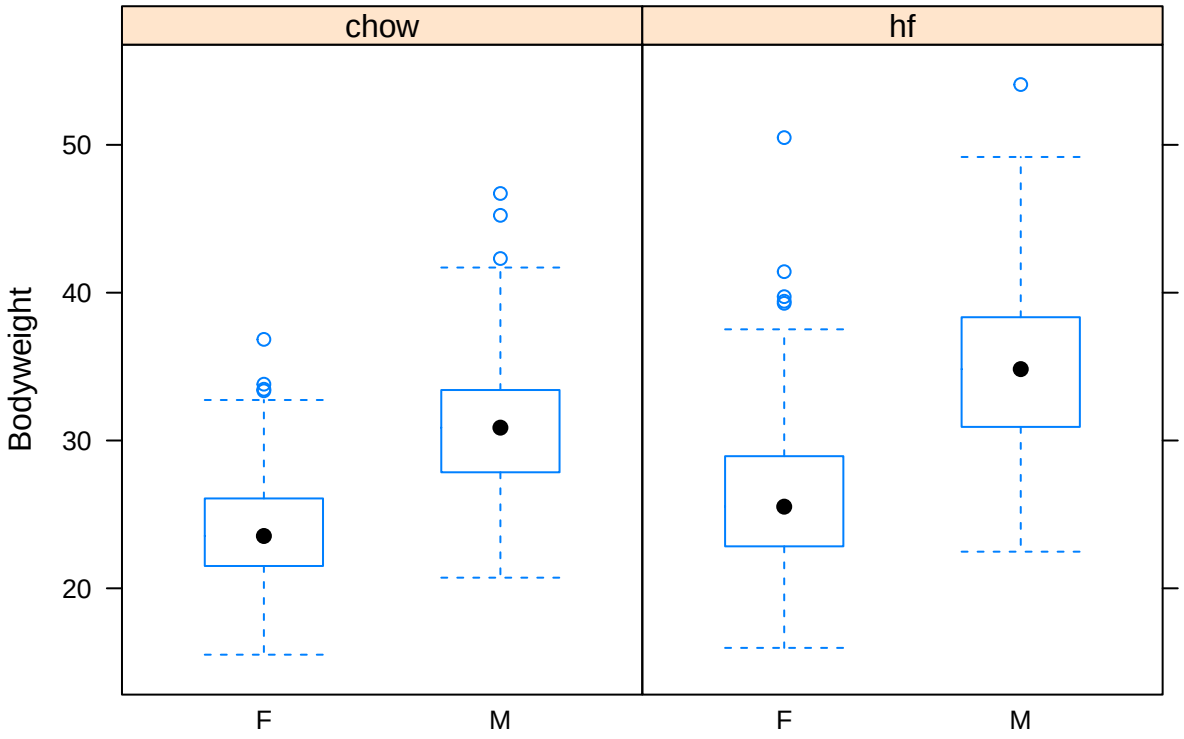

```
X11()
# BOXPLOT
boxplot(pheno[,"Bodyweight"]~pheno[,"Diet"])
```



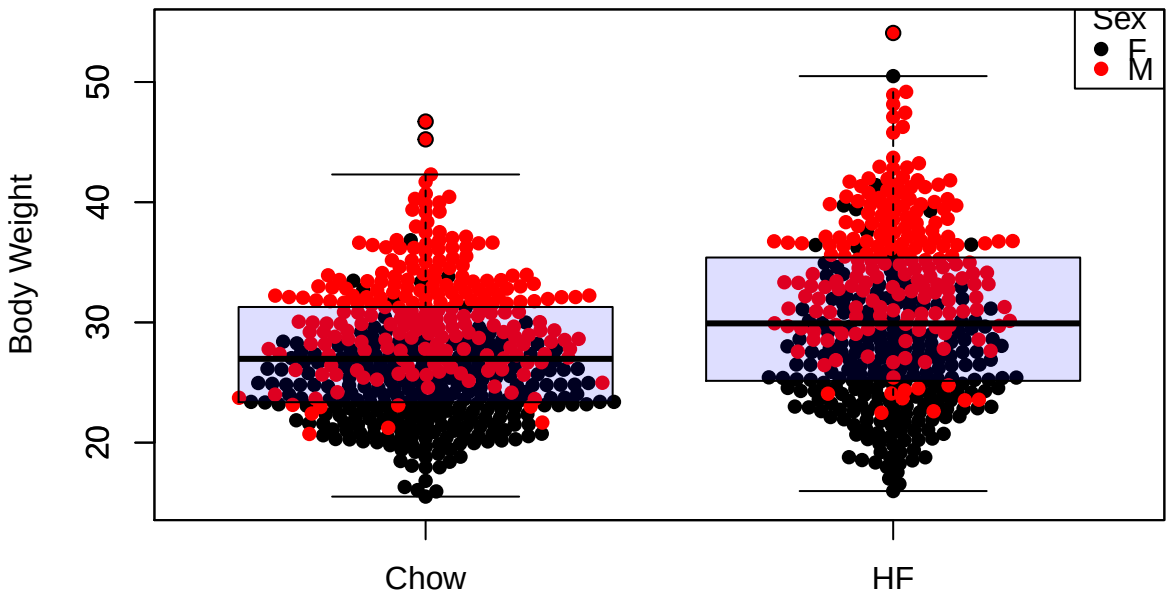

```
X11()
library(lattice)
bwplot(pheno$Bodyweight ~ pheno$Diet|pheno$Sex,
       ylab="Bodyweight", xlab="",
       main="",
       #layout=(c(1,3))
)
```




```
X11()
bwplot(pheno$Bodyweight ~ pheno$Sex|pheno$Diet,
       ylab="Bodyweight", xlab="",
       main="",
       #layout=(c(1,3))
)
```




```
library(beeswarm)
X11()
beeswarm(Bodyweight ~ Diet, data = pheno, pch = 16,
         pwcol = as.numeric(Sex), xlab = '',
         ylab = 'Body Weight',
         labels = c('Chow', 'HF'))
boxplot(Bodyweight ~ Diet, data = pheno, add = T,
        names = c("",""), col="#0000ff22")
legend('topright', legend = levels(pheno$Sex), title = 'Sex',
       pch = 16, col = 1:2)
```



Variables aleatorias: muestra vs población

Vamos ahora a repasar el concepto de variable aleatoria que hemos aprendido en la primera parte de la clase.

En este experimento tenemos una variable aleatoria (peso) medida en ratones control y en ratones con una dieta rica en grasa. El objetivo es comparar si la dieta está produciendo una diferencia en los pesos de los ratones.

En el fichero "femaleMiceWeights.csv" tenéis información de una muestra de 12 ratones hembras en dieta control y 12 en dieta HF elegidas al azar de la población completa que está en el fichero "mice_pheno.csv"

```
dat <- read.csv("Data/femaleMiceWeights.csv")
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
```

```
View(dat)

# creamos dos data frames separados
control <- filter(dat,Diet=="chow") %>% select(Bodyweight) %>% unlist
treatment <- filter(dat,Diet=="hf") %>% select(Bodyweight) %>% unlist
```

```
control
```

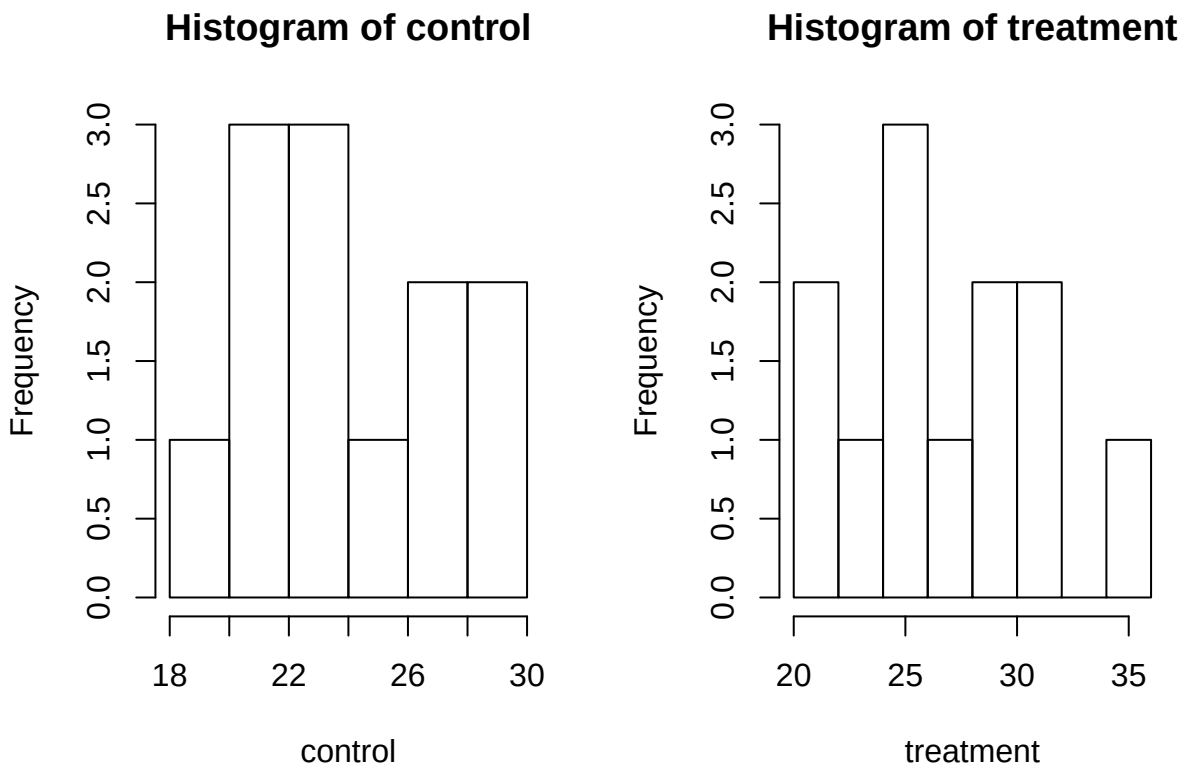
```
##  Bodyweight1  Bodyweight2  Bodyweight3  Bodyweight4  Bodyweight5
##        21.51        28.14        24.04        23.45        23.68
##  Bodyweight6  Bodyweight7  Bodyweight8  Bodyweight9 Bodyweight10
##        19.79        28.40        20.98        22.51        20.10
## Bodyweight11 Bodyweight12
##        26.91        26.25
```

```
treatment
```

```
##  Bodyweight1  Bodyweight2  Bodyweight3  Bodyweight4  Bodyweight5
##        25.71        26.37        22.80        25.34        24.97
##  Bodyweight6  Bodyweight7  Bodyweight8  Bodyweight9 Bodyweight10
##        28.14        29.58        30.92        34.02        21.90
## Bodyweight11 Bodyweight12
##        31.53        20.73
```

Vamos a ver cuan grande (o pequeña) es la diferencia de las medias del peso de estos dos grupos de 12 ratones, que es todo lo que tenemos para sacar una conclusión general de nuestro problema:

```
X11()
par(mfrow=c(1,2))
hist(control)
hist(treatment)
```




```
# cómo son sus medias y la diferencia de las medias?
print( mean(treatment) )
```

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

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

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

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

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

Para poder decir si esta diferencia es grande o pequeña necesitamos comparar con algo. Como tenemos toda la poblacion de ratones C57 podemos generar tantas muestras de 12 ratones como queramos a partir de 225 ratones que son toda la poblacion de controles:

```
#whole population
population <- read.csv("Data/femaleControlsPopulation.csv")
##use unlist to turn it into a numeric vector
population <- unlist(population)
mean(population)
```

```
## [1] 23.89338
```

```
#[1] 23.89338

summary(chow.data[,3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   15.51   23.38   26.97   27.41   31.28   46.71       1
```

```
summary(chow.data[which(chow.data$Sex=="F"),3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.51   21.51   23.54   23.89   26.08   36.84
```

```
#lets sample 12 mice from the control population repeatedly
control <- sample(population,12)
mean(control)
```

```
## [1] 22.9875
```

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

```
## [1] 24.845
```

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

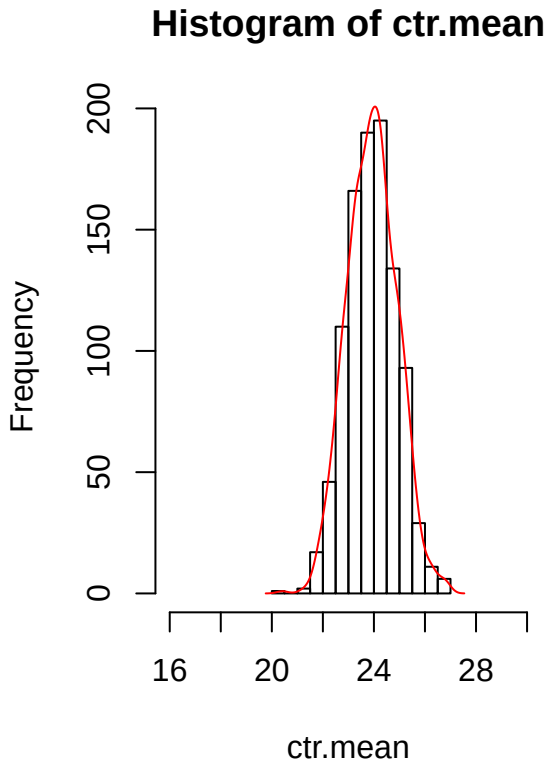
```
## [1] 24.43667
```

```
ctr.mean=numeric()
for (i in 1:1000){
  ctr.mean[i]=mean(sample(population,12))
}

par(mfrow=c(1,2))
myhist=hist(ctr.mean,xlim=c(16,30))
mydensity <- density(ctr.mean)
multiplier <- myhist$counts / myhist$density
```

```
mydensity$y <- mydensity$y * multiplier[1]
summary(ctr.mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.42   23.25   23.92   23.93   24.57   26.89
```

```
lines(mydensity,col="red")
```

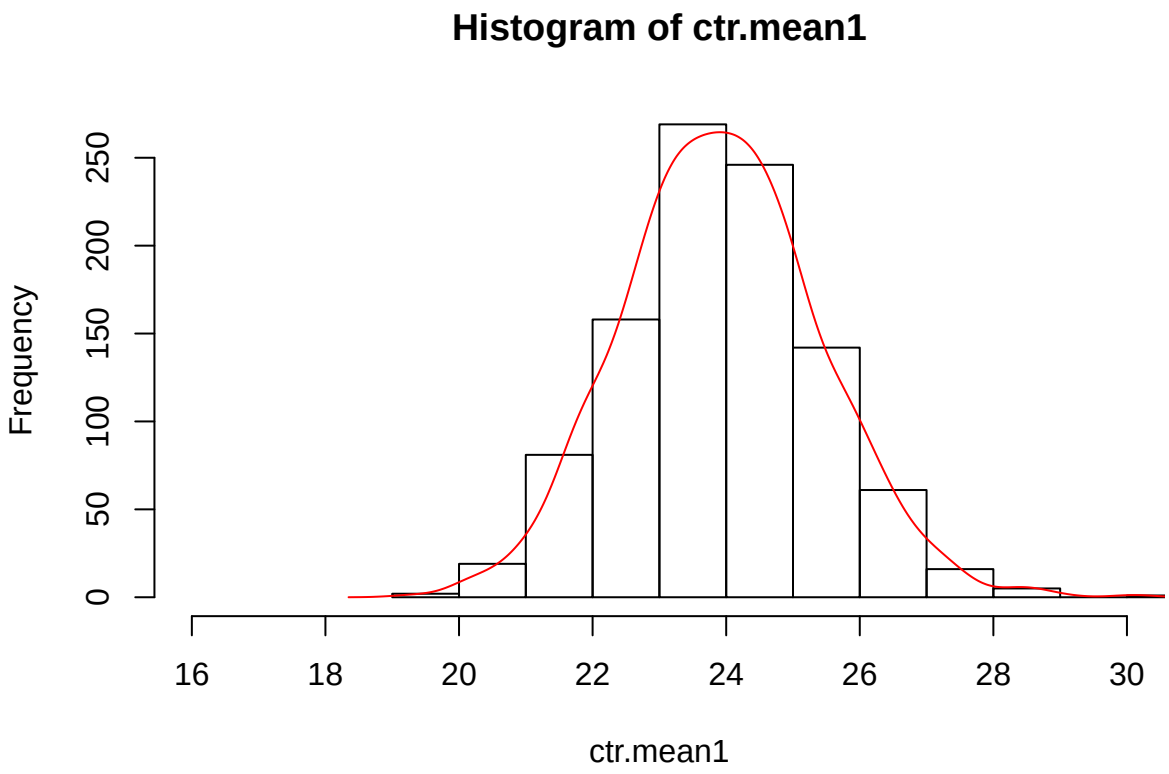


Que pasaría si samplearamos muestras de tan sólo 5 ratones?

```
ctr.mean1=numeric()
for (i in 1:1000){
  ctr.mean1[i]=mean(sample(population,5))
}
mean(ctr.mean1)
```

```
## [1] 23.93203
```

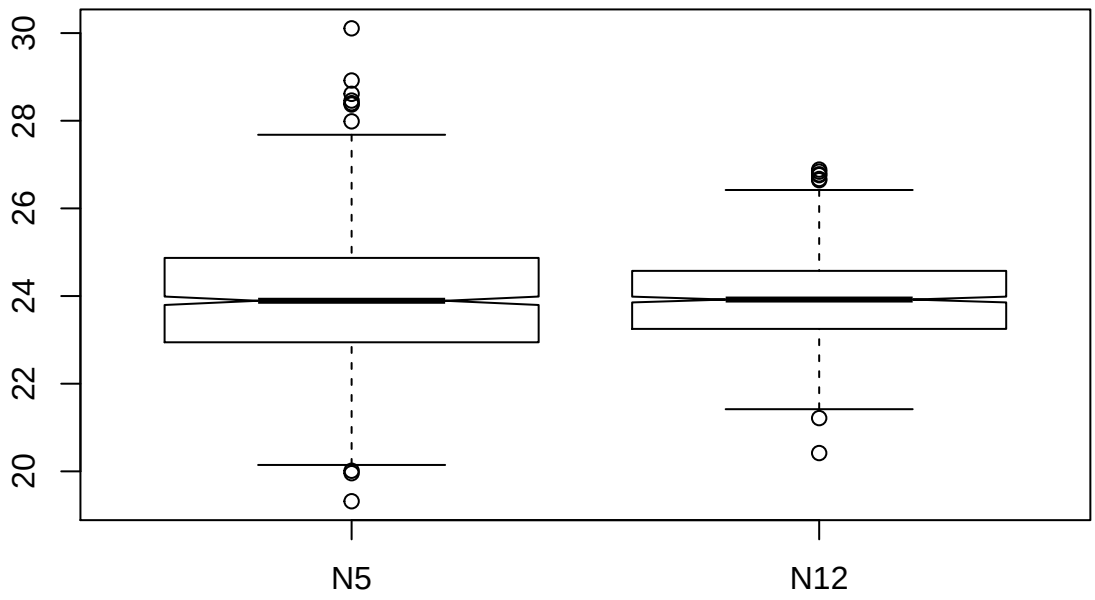
```
myhist=hist(ctr.mean1,xlim=c(16,30))
mydensity <- density(ctr.mean1)
multiplier <- myhist$counts / myhist$density
mydensity$y <- mydensity$y * multiplier[1]
lines(mydensity,col="red")
```




```
summary(ctr.mean1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.32   22.95   23.89   23.93   24.87   30.11
```

```
boxplot(data.frame(N5=ctr.mean1,N12=ctr.mean),notch=T)
```



Null distribution

El concepto de *null distribution* es muy importante en inferencia estadística. Cuando comparamos cosas siempre tenemos una hipótesis de partida que queremos saber si podemos rechazar o no con los datos que tenemos.

La hipótesis nula nunca se puede aceptar. Porque excepto en este caso en el que tenemos todos los ratones del planeta tierra de la cepa C57 normalmente no podemos tener la certeza de que algo sea verdad: buscamos contraejemplos en los que no se cumpla y si encontramos alguno es que no es verdad. Si no encontramos ningun contraejemplo no quiere decir que no exista, sólo que no lo hemos encontrado.

Volvemos a cargar el dataset que contiene tan solo una muestra de 12 ratones control y 12 HF.

```
dat <- read.csv("Data/femaleMiceWeights.csv")
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
```

```
View(dat)
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
```

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

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

Nuestra hipótesis nula sería: **No hay diferencias entre el peso del grupo control y del que recibe dieta grasa**. Podemos decir algo acerca de nuestra hipótesis nula mirando a la diferencia que nos ha salido? La diferencia es 3.

Si la diferencia de 3 que hemos obtenido para nuestra muestra fuera "grande" podríamos rechazar la hipótesis nula. Vamos a comparar la diferencia entre los grupos con diferencias en peso que pudieramos encontrar incluso dentro de ratones del grupo control. Seleccionamos muestras de 12 elementos todas del grupo sin tratamiento.

```
diff.null=numeric()
ns=10000
for (i in 1:ns){
  control=sample(population,12)
  treatment=sample(population,12)
  diff.null[i]=mean(treatment)-mean(control)
}

max(diff.null)
```

```
## [1] 4.74
```

```
min(diff.null)
```

```
## [1] -5.764167
```

```
mean(diff.null)
```

```
## [1] -0.02369508
```

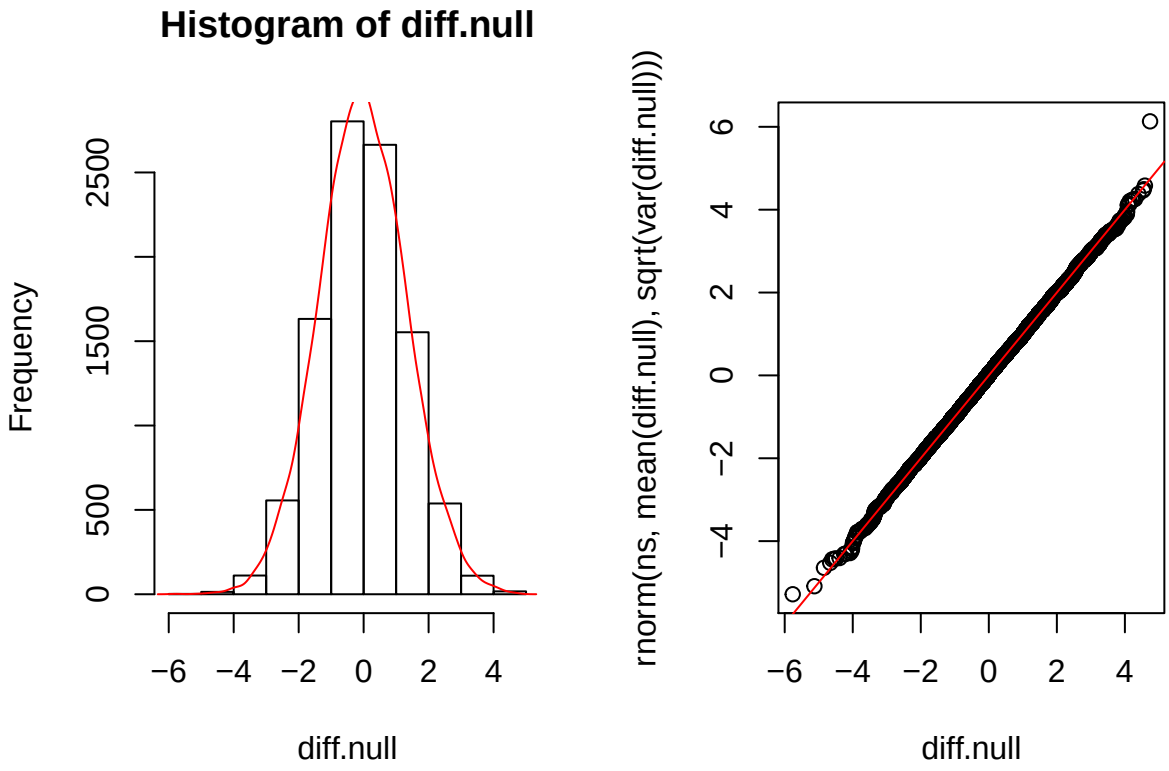
```
median(diff.null)
```

```
## [1] -0.03583333
```

```
sqrt(var(diff.null))
```

```
## [1] 1.342214
```

```
par(mfrow=c(1,2))
myhist=hist(diff.null)
mydensity <- density(diff.null)
multiplier <- myhist$counts / myhist$density
mydensity$y <- mydensity$y * multiplier[1]

lines(mydensity,col="red")
qqplot(diff.null,rnorm(ns,mean(diff.null),sqrt(var(diff.null))))
abline(a=0,b=1,col="red")
```



Acabamos de construir la distribución empírica de la differencia de pesos en animales control. Pese a ser todos iguales encontramos algunas diferencias de hasta 5 y -5. Esto es normal debido a la variabilidad intrínseca al peso de los propios animales. Sin embargo, como de probable es encontrar dos muestras de 12 ratones todos sin tratar para los que la diferencia de medias en su peso sea mayor de 3? (la diferencia que hemos obtenido entre controles y HFD)? Con lo aprendido en la primera sección de la clase:

```
#Probability of getting a value larger than three under the null
length(which(diff.null>=3))
```

```
## [1] 127
```

```
length(which(diff.null>=3))/ns
```

```
## [1] 0.0127
```

acabamos de calcular el p-valor de nuestra diferencia sin utilizar ninguna distribución formal, sólo distribuciones empíricas de los propios datos porque tenemos suficientes. Este es el cambio de paradigma que se está produciendo en data science.

p-value

Podríamos haber hecho lo mismo utilizando las distribuciones de los datos. Si la variable de la que viene mi muestra es normal, sabemos que la diferencia de medias sigue una distribución de la t. Por lo tanto podemos calcular formalmente el p-valor:

```
dat <- read.csv("Data/femaleMiceWeights.csv")
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
```

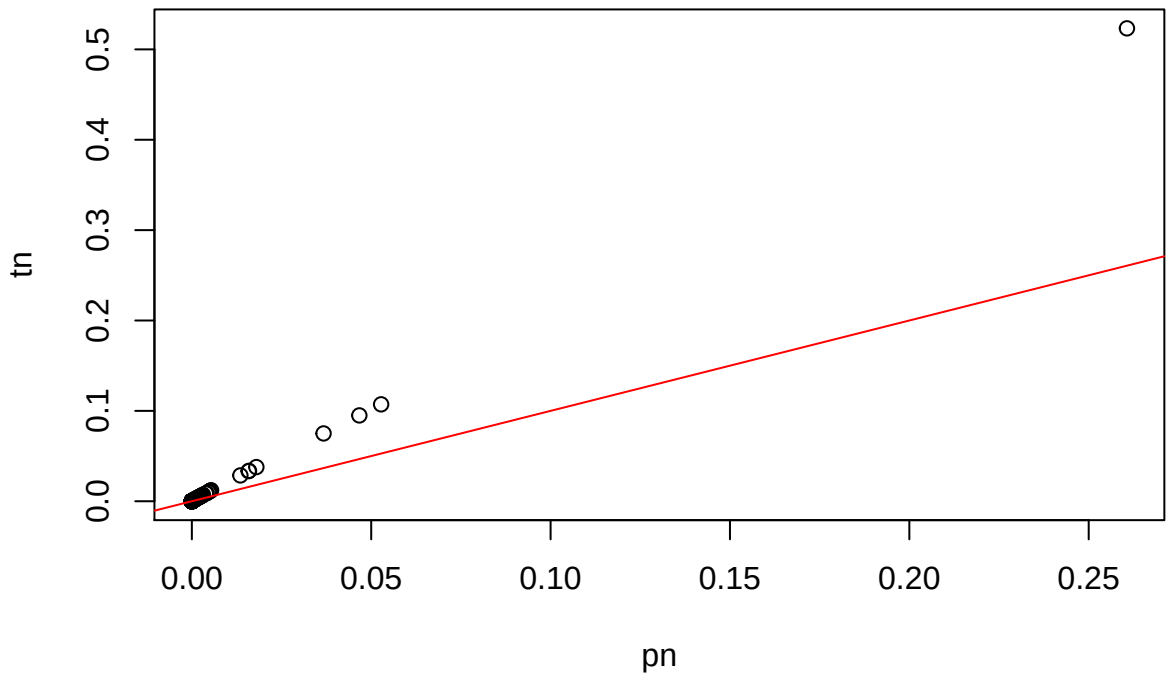
```
View(dat)

# sample of 12 mice under chow diet and 12 mice under HF diet
control <- filter(dat,Diet=="chow") %>% select(Bodyweight) %>% unlist
treatment <- filter(dat,Diet=="hf") %>% select(Bodyweight) %>% unlist

obs=mean(treatment)-mean(control)

sigma2.x=var(treatment,na.rm=T)
sigma2.y=var(control,na.rm=T)

se=sqrt(sigma2.x/length(treatment)+sigma2.y/length(control))
tstat=obs/se
1-pnorm(tstat)
```

```
## [1] 0.0199311
```

```
#0.0199311

t.test(treatment,control)
```

```
##
##  Welch Two Sample t-test
##
## data:  treatment and control
## t = 2.0552, df = 20.236, p-value = 0.053
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.04296563  6.08463229
## sample estimates:
## mean of x mean of y
##  26.83417  23.81333
```

```
pop.control <- filter(pheno,Diet=="chow") %>% select(Bodyweight) %>% unlist
pop.treatment <- filter(pheno,Diet=="hf") %>% select(Bodyweight) %>% unlist

pn=tn=numeric()
for (i in 1:100){
  treatment=sample(pop.treatment,100)
  control=sample(pop.control,100)
  obs=mean(treatment,na.rm=T)-mean(control,na.rm=T)
  se=sqrt(var(treatment,na.rm=T)/length(treatment)+var(control,na.rm=T)/length(control))
  pn[i]=1-pnorm(obs/se)
  tn[i]=t.test(treatment,control)[[3]]
}
plot(pn,tn)
abline(0,1,col="red")
```



Intervalos de confianza

Es cada vez más extendido el uso de los intervalos de confianza en lugar de reportar sólo el p valor. El intervalo de confianza nos da una idea de la incertidumbre de nuestra decision.

Volvemos a partir de toda la población de ratones C57:

```
dat <- read.csv("Data/mice_pheno.csv")
chowPopulation <- dat[dat$Sex=="F" & dat$Diet=="chow",3]
```

Nos quedamos solo con aquellos que tienen dieta control y son females. Exploramos su media.

```
mu_chow <- mean(chowPopulation)
print(mu_chow)
```

```
## [1] 23.89338
```

Que pasaría si tomásemos muestras de 30 animales de esta subpoblación? Cómo de bien estimaríamos la media?

```
N <- 30
chow <- sample(chowPopulation,N)
print(mean(chow))
```

```
## [1] 24.61033
```

```
se <- sd(chow)/sqrt(N)
print(se)
```

```
## [1] 0.7157741
```

Nuestra media tiene una probabilidad del 95% de pertenecer en valor absoluto a una distribución normal.

```
Q <- qnorm(1- 0.05/2)
interval <- c(mean(chow)-Q*se, mean(chow)+Q*se )
interval
```

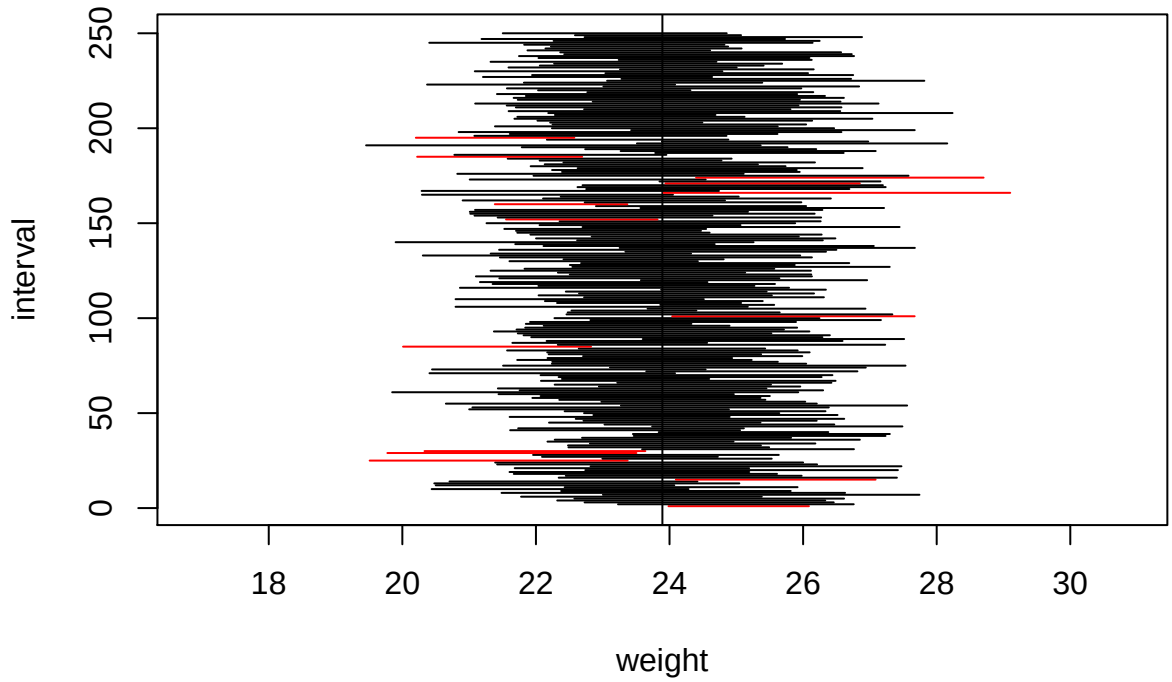
```
## [1] 23.20744 26.01322
```

```
interval[1] < mu_chow & interval[2] > mu_chow
```

```
## [1] TRUE
```

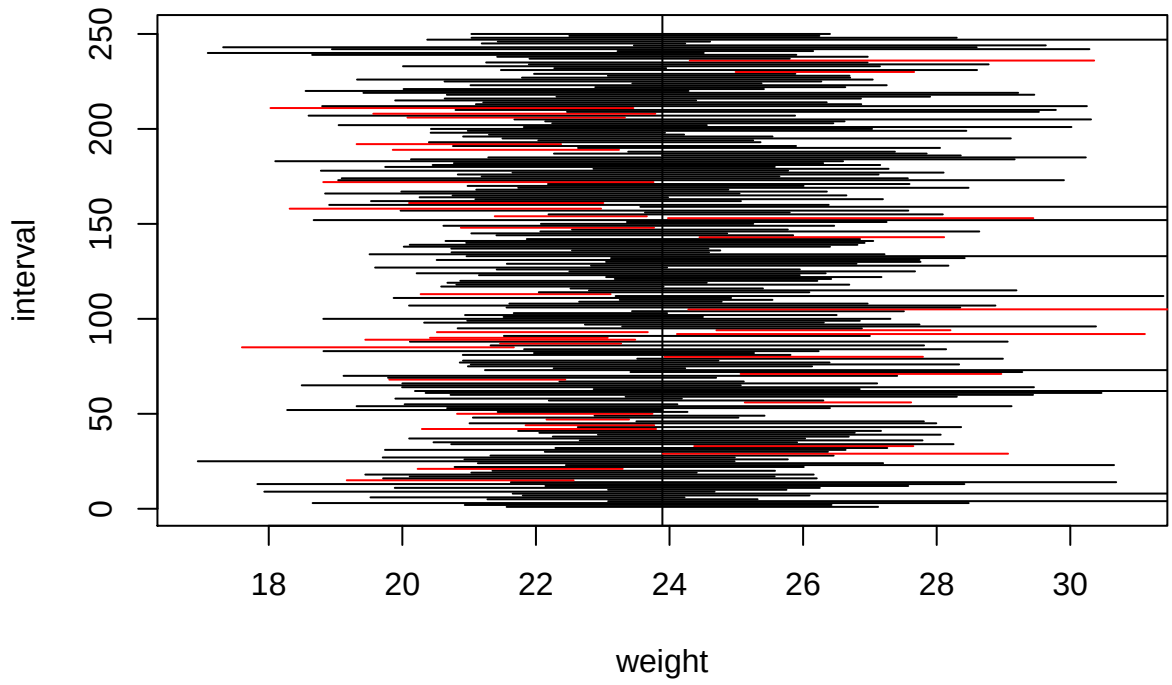
Vamos a repetir esto 250 veces. Cuántas veces sigue cayendo nuestra media en los límites de la distribución?

```
B <- 250
N<-12
plot(mean(chowPopulation)+c(-7,7),c(1,1),type="n",
     xlab="weight",ylab="interval",ylim=c(1,B))
abline(v=mean(chowPopulation))
for (i in 1:B) {
  chow <- sample(chowPopulation,N)
  se <- sd(chow)/sqrt(N)
  interval <- c(mean(chow)-Q*se, mean(chow)+Q*se)
  covered <-
    mean(chowPopulation) <= interval[2] & mean(chowPopulation) >= interval[1]
  color <- ifelse(covered,1,2)
  lines(interval, c(i,i),col=color)
}
```



Que pasa si nuestra muestra es de solo 5 elementos en lugar de 12?

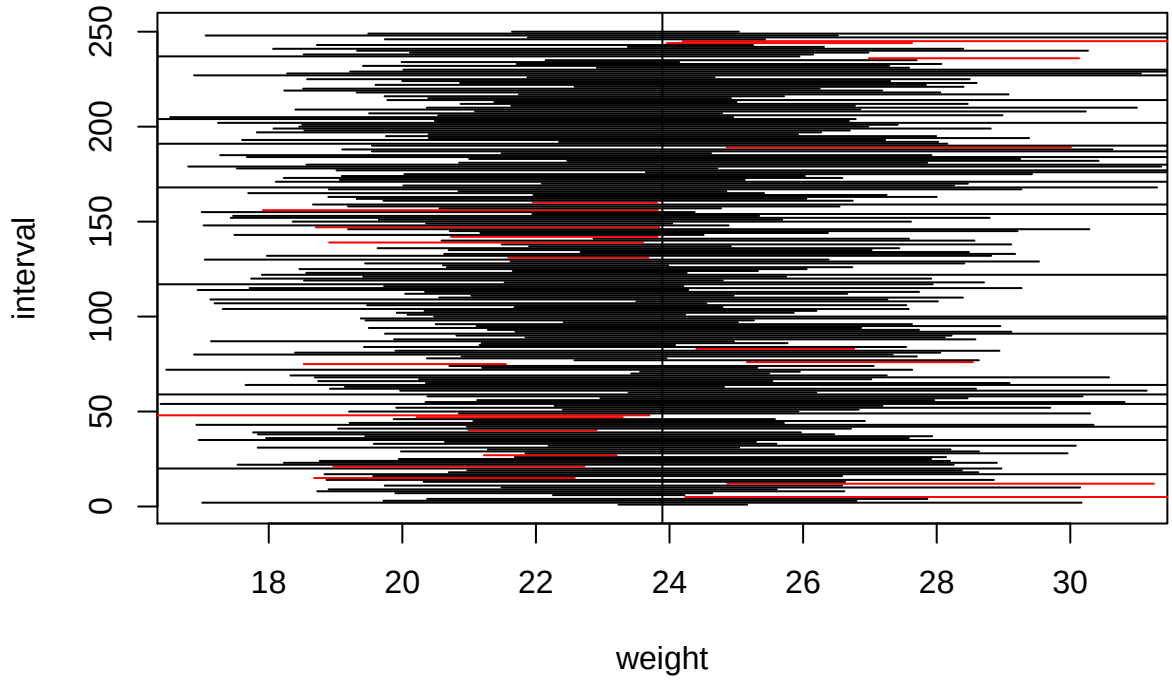
```
# Small sample size
plot(mean(chowPopulation)+c(-7,7),c(1,1),type="n",
     xlab="weight",ylab="interval",ylim=c(1,B))
abline(v=mean(chowPopulation))
Q <- qnorm(1- 0.05/2)

N <- 5
for (i in 1:B) {
  chow <- sample(chowPopulation,N)
  se <- sd(chow)/sqrt(N)
  interval <- c(mean(chow)-Q*se, mean(chow)+Q*se)
  covered <- mean(chowPopulation) <= interval[2] & mean(chowPopulation) >= interval[1]
  color <- ifelse(covered,1,2)
  lines(interval, c(i,i),col=color)
}
```



Para una N tan pequeña la diferencia de las medias no sigue una normal sino un students´ t:

```
plot(mean(chowPopulation) + c(-7,7), c(1,1), type="n",
     xlab="weight", ylab="interval", ylim=c(1,B))
abline(v=mean(chowPopulation))
##Q <- qnorm(1- 0.05/2) ##no longer normal so use:
Q <- qt(1- 0.05/2, df=4)
N <- 5
for (i in 1:B) {
  chow <- sample(chowPopulation, N)
  se <- sd(chow)/sqrt(N)
```

```
  interval <- c(mean(chow)-Q*se, mean(chow)+Q*se )
  covered <- mean(chowPopulation) <= interval[2] & mean(chowPopulation) >= interval[1]
  color <- ifelse(covered,1,2)
  lines(interval, c(i,i),col=color)
}
```




```
qt(1- 0.05/2, df=4)
```

```
## [1] 2.776445
```