Elementos de Estadísticas Descriptiva

Daniela Pinto Veizaga

9/7/2019

Paquetería:

```
library("MASS")
```

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

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library ("ggplot2")
```

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

```
library ("prob")
```

```
## Loading required package: combinat
```

```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

```
## Loading required package: fAsianOptions
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

```
## Loading required package: fBasics
```

```
## Loading required package: fOptions
```

```
##
## Attaching package: 'prob'
```

```
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```
library ("tidyr")
```

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

Ejercicio 1

Considérese el experimento de lanzar 20 veces una moneda y obtener la secuencia:

$$H, T, H, H, T, H, H, T, H, H, T, T, H, T, T, T, H, H, H, T$$

a) Tabular los resultados del experimento anterior encontrando las proporciones de H y T en los 20 lanzamientos.

```
coin_toss_20 = c("H","T","H","H","T","H","H","T","H","H","T","T","H","T","T","T","H","H","H","T")
coin_toss_20 = factor(coin_toss_20, labels=c("Head","Tails"))
coin_toss_20
```

```
##  [1] Head  Tails Head  Head  Tails Head  Head  Tails Head  Head  Tails
## [12] Tails Head  Tails Tails Tails Head  Head  Head  Tails
## Levels: Head Tails
```

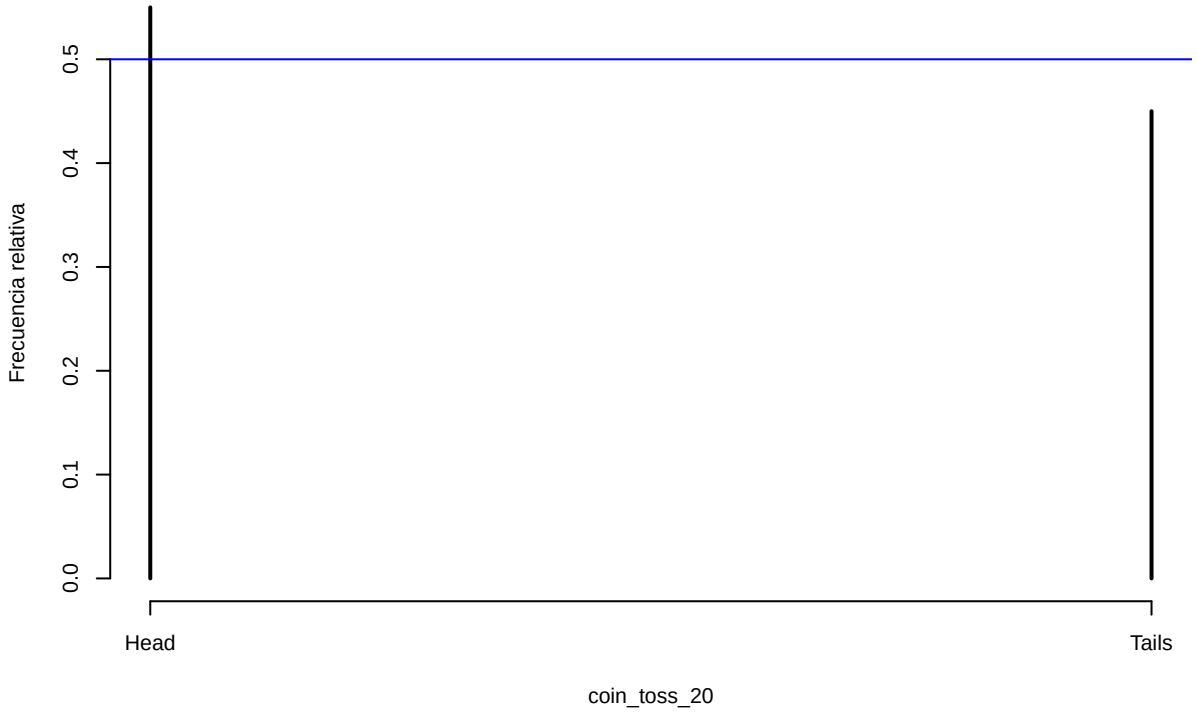
```
prop <- prop.table(table(coin_toss_20))
prop
```

```
## coin_toss_20
##  Head Tails
##  0.55  0.45
```

b) Graficar las proporciones con barplot y plot.

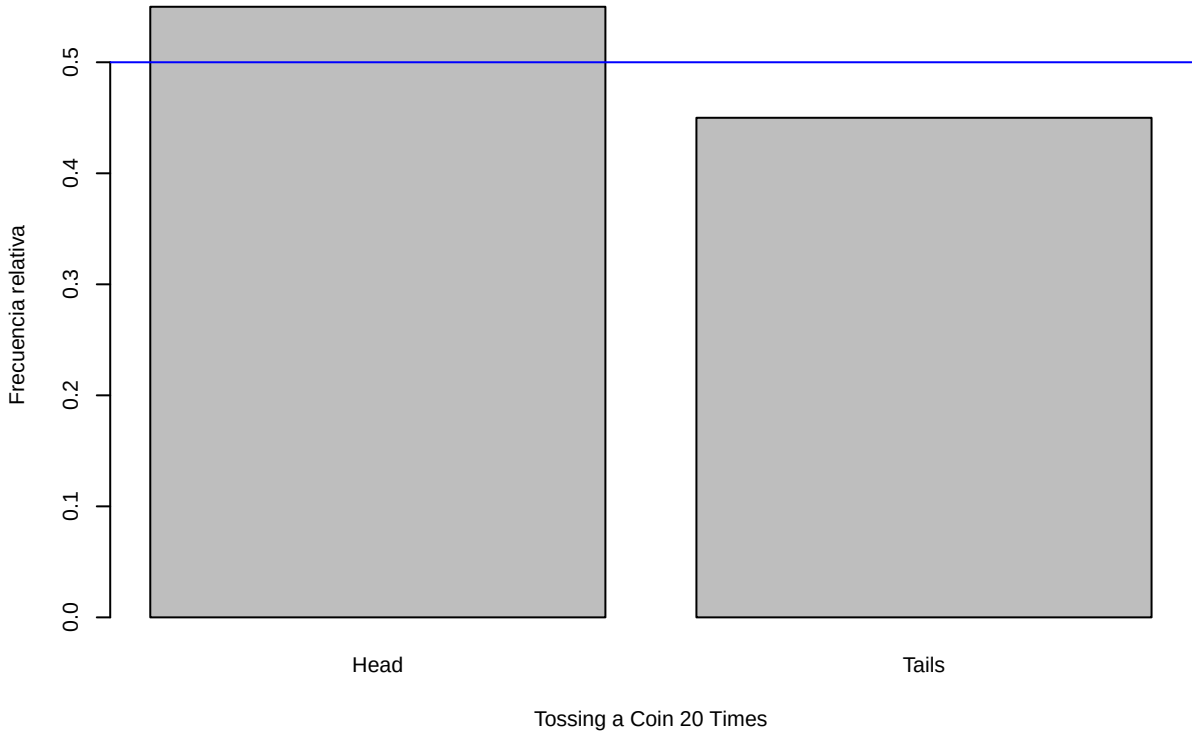
Plot function

```
par(cex=0.7) #control size of labels
plot(prop, ylab='Frecuencia relativa')
abline(h = .5, col='blue')  #se traza una línea con abline en .
```



Barplot function

```
par(cex=0.7) #control size of labels
barplot(prop, xlab='Tossing a Coin 20 Times', ylab='Frecuencia relativa')
abline(h = .5, col='blue') #se traza una línea con abline en .5
```



Ejercicio 2

Para el dataset analizado anteriormente y que se creó: log_mammals realizar:

```
head(mammals) #returns the first 5(default) rows of the matrix mammals
```

```
##                    body brain
## Arctic fox        3.385  44.5
## Owl monkey        0.480  15.5
## Mountain beaver   1.350   8.1
## Cow             465.000 423.0
## Grey wolf        36.330 119.5
## Goat             27.660 115.0
```

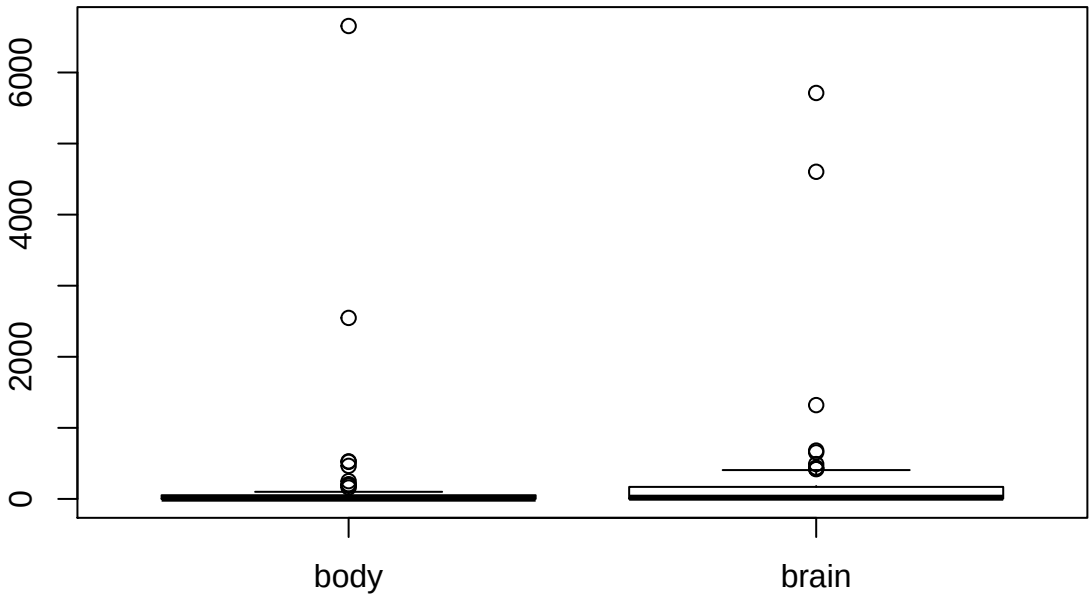
```
head(as.matrix(mammals))
```

```
##                    body brain
## Arctic fox        3.385  44.5
## Owl monkey        0.480  15.5
## Mountain beaver   1.350   8.1
## Cow             465.000 423.0
## Grey wolf        36.330 119.5
## Goat             27.660 115.0
```

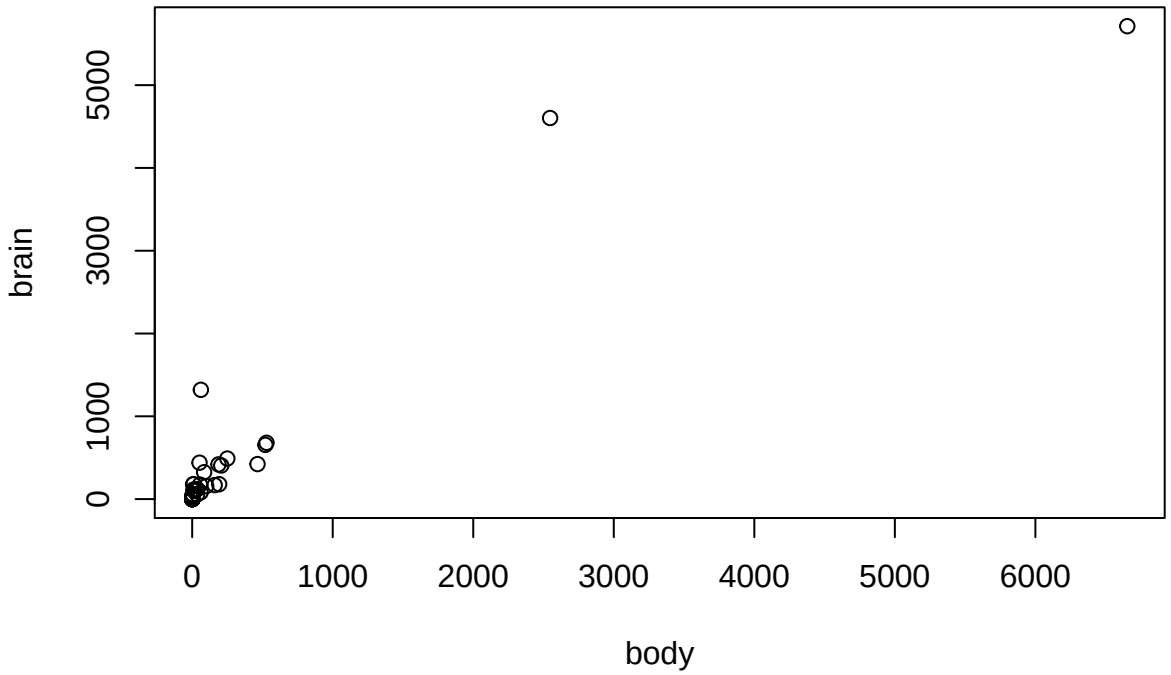
```
options(digits=7)
summary(mammals)
```

```
##       body              brain
##  Min.   :   0.005   Min.   :   0.14
##  1st Qu.:   0.600   1st Qu.:   4.25
##  Median :   3.342   Median :  17.25
##  Mean   : 198.790   Mean   : 283.13
##  3rd Qu.:  48.203   3rd Qu.: 166.00
##  Max.   :6654.000   Max.   :5712.00
```

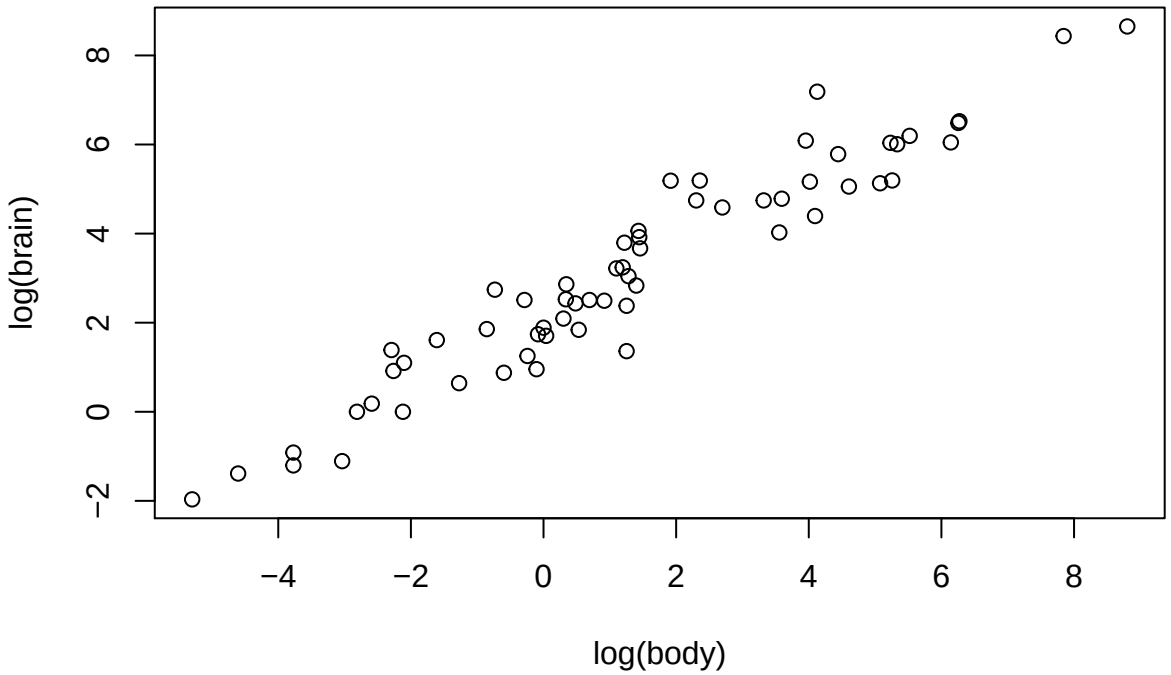
```
boxplot(mammals)
```



```r
plot(mammals) #scatterplot del data.frame mammals
```




```r
plot(log(mammals$body),  #se aplica logaritmo a todo un vector
     log(mammals$brain), #se aplica logaritmo a todo un vector
     xlab="log(body)", ylab="log(brain)")
```




```r
log_mammals <- log(mammals)
colnames(log_mammals) <- c('log_body', 'log_brain')
head(log_mammals, 10) #returns the first 10 rows of the matrix mammals
```

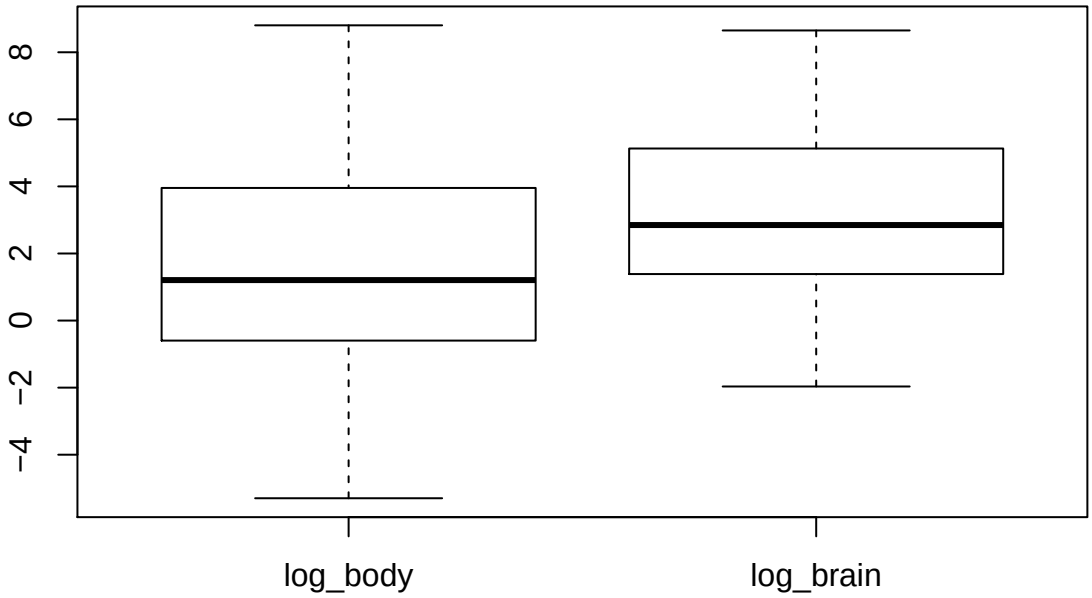
```
##                    log_body log_brain
```

```
## Arctic fox       1.21935391  3.795489
## Owl monkey      -0.73396918  2.740840
## Mountain beaver  0.30010459  2.091864
## Cow              6.14203741  6.047372
## Grey wolf        3.59264385  4.783316
## Goat             3.31998733  4.744932
## Roe deer         2.69665216  4.587006
## Guinea pig       0.03922071  1.704748
## Verbet           1.43270073  4.060443
## Chinchilla      -0.85566611  1.856298
```

```
summary(log_mammals)
```

```
##     log_body         log_brain
##  Min.   :-5.2983   Min.   :-1.966
##  1st Qu.:-0.5203   1st Qu.: 1.442
##  Median : 1.2066   Median : 2.848
##  Mean   : 1.3375   Mean   : 3.140
##  3rd Qu.: 3.8639   3rd Qu.: 5.111
##  Max.   : 8.8030   Max.   : 8.650
```

```
boxplot(log_mammals)
```




```
print((boxplot.stats(mammals$body)$out)) #obteniendo los outliers de la variable body
```

```
##  [1]  465.0 2547.0  187.1  521.0  529.0  207.0 6654.0  250.0  192.0  160.0
```

```
print((boxplot.stats(mammals$brain)$out)) #obteniendo los outliers de la variable brain
```

```
## [1]  423 4603  419  655  680 1320 5712  490  440
```

```
df_mammals_body <- data.frame(medida = rep('body', times=nrow(mammals)), mammal=rownames(log_mammals), v
df_mammals_body
```

```
##    medida                  mammal       valor
## 1    body              Arctic fox  1.21935391
## 2    body              Owl monkey -0.73396918
## 3    body          Mountain beaver  0.30010459
```

```
## 4    body                       Cow  6.14203741
## 5    body                 Grey wolf  3.59264385
## 6    body                      Goat  3.31998733
## 7    body                  Roe deer  2.69665216
## 8    body                Guinea pig  0.03922071
## 9    body                    Verbet  1.43270073
## 10   body               Chinchilla -0.85566611
## 11   body           Ground squirrel -2.29263476
## 12   body    Arctic ground squirrel -0.08338161
## 13   body African giant pouched rat  0.00000000
## 14   body Lesser short-tailed shrew -5.29831737
## 15   body           Star-nosed mole -2.81341072
## 16   body     Nine-banded armadillo  1.25276297
## 17   body                Tree hyrax  0.69314718
## 18   body              N.A. opossum  0.53062825
## 19   body            Asian elephant  7.84267147
## 20   body             Big brown bat -3.77226106
## 21   body                    Donkey  5.23164323
## 22   body                     Horse  6.25575004
## 23   body         European hedgehog -0.24207156
## 24   body              Patas monkey  2.30258509
## 25   body                       Cat  1.19392247
## 26   body                    Galago -1.60943791
## 27   body                     Genet  0.34358970
## 28   body                   Giraffe  6.27098843
## 29   body                   Gorilla  5.33271879
## 30   body                 Grey seal  4.44265126
## 31   body              Rock hyrax-a -0.28768207
## 32   body                     Human  4.12713439
## 33   body          African elephant  8.80297346
## 34   body             Water opossum  1.25276297
## 35   body             Rhesus monkey  1.91692261
## 36   body                  Kangaroo  3.55534806
## 37   body     Yellow-bellied marmot  1.39871688
## 38   body             Golden hamster -2.12026354
## 39   body                    Mouse -3.77226106
## 40   body          Little brown bat -4.60517019
## 41   body                Slow loris  0.33647224
## 42   body                     Okapi  5.52146092
## 43   body                    Rabbit  0.91629073
## 44   body                     Sheep  4.01638302
## 45   body                    Jaguar  4.60517019
## 46   body                Chimpanzee  3.95431592
## 47   body                    Baboon  2.35612586
## 48   body           Desert hedgehog -0.59783700
## 49   body           Giant armadillo  4.09434456
## 50   body              Rock hyrax-b  1.28093385
## 51   body                   Raccoon  1.45582042
## 52   body                       Rat -1.27296568
## 53   body          E. American mole -2.59026717
## 54   body                 Mole rat -2.10373423
## 55   body               Musk shrew -3.03655427
## 56   body                       Pig  5.25749537
## 57   body                   Echidna  1.09861229
```

```
## 58   body           Brazilian tapir  5.07517382
## 59   body                    Tenrec -0.10536052
## 60   body                 Phalanger  0.48242615
## 61   body                Tree shrew -2.26336438
## 62   body                   Red fox  1.44338333
```

```
df_mammals_brain <- data.frame(medida = rep('brain', times=nrow(mammals)), mammal=rownames(log_mammals)
df_mammals_brain
```

```
##    medida                    mammal      valor
## 1   brain                Arctic fox  3.7954892
## 2   brain                Owl monkey  2.7408400
## 3   brain           Mountain beaver  2.0918641
## 4   brain                       Cow  6.0473722
## 5   brain                 Grey wolf  4.7833164
## 6   brain                      Goat  4.7449321
## 7   brain                  Roe deer  4.5870062
## 8   brain                Guinea pig  1.7047481
## 9   brain                    Verbet  4.0604430
## 10  brain                Chinchilla  1.8562980
## 11  brain           Ground squirrel  1.3862944
## 12  brain    Arctic ground squirrel  1.7404662
## 13  brain African giant pouched rat  1.8870696
## 14  brain Lesser short-tailed shrew -1.9661129
## 15  brain           Star-nosed mole  0.0000000
## 16  brain     Nine-banded armadillo  2.3795461
## 17  brain                Tree hyrax  2.5095993
## 18  brain              N.A. opossum  1.8405496
## 19  brain            Asian elephant  8.4344635
## 20  brain             Big brown bat -1.2039728
## 21  brain                    Donkey  6.0378709
## 22  brain                     Horse  6.4846352
## 23  brain         European hedgehog  1.2527630
## 24  brain              Patas monkey  4.7449321
## 25  brain                       Cat  3.2425924
## 26  brain                    Galago  1.6094379
## 27  brain                     Genet  2.8622009
## 28  brain                   Giraffe  6.5220928
## 29  brain                   Gorilla  6.0063532
## 30  brain                 Grey seal  5.7838252
## 31  brain              Rock hyrax-a  2.5095993
## 32  brain                     Human  7.1853870
## 33  brain          African elephant  8.6503245
## 34  brain             Water opossum  1.3609766
## 35  brain             Rhesus monkey  5.1873858
## 36  brain                  Kangaroo  4.0253517
## 37  brain     Yellow-bellied marmot  2.8332133
## 38  brain            Golden hamster  0.0000000
## 39  brain                     Mouse -0.9162907
## 40  brain          Little brown bat -1.3862944
## 41  brain                Slow loris  2.5257286
## 42  brain                     Okapi  6.1944054
## 43  brain                    Rabbit  2.4932055
## 44  brain                     Sheep  5.1647860
## 45  brain                    Jaguar  5.0562458
```

```
## 46  brain               Chimpanzee  6.0867747
## 47  brain                   Baboon  5.1901752
## 48  brain           Desert hedgehog  0.8754687
## 49  brain           Giant armadillo  4.3944492
## 50  brain              Rock hyrax-b  3.0445224
## 51  brain                   Raccoon  3.6686767
## 52  brain                       Rat  0.6418539
## 53  brain          E. American mole  0.1823216
## 54  brain                 Mole rat  1.0986123
## 55  brain               Musk shrew -1.1086626
## 56  brain                       Pig  5.1929569
## 57  brain                   Echidna  3.2188758
## 58  brain           Brazilian tapir  5.1298987
## 59  brain                    Tenrec  0.9555114
## 60  brain                 Phalanger  2.4336134
## 61  brain                Tree shrew  0.9162907
## 62  brain                   Red fox  3.9199912
```

```
df_mammals_gg <- rbind(df_mammals_body, df_mammals_brain) # appends or combines vector, matrix or data
df_mammals_gg
```

```
##     medida                    mammal       valor
## 1     body                Arctic fox  1.21935391
## 2     body                Owl monkey -0.73396918
## 3     body           Mountain beaver  0.30010459
## 4     body                       Cow  6.14203741
## 5     body                 Grey wolf  3.59264385
## 6     body                      Goat  3.31998733
## 7     body                  Roe deer  2.69665216
## 8     body                Guinea pig  0.03922071
## 9     body                    Verbet  1.43270073
## 10    body                Chinchilla -0.85566611
## 11    body           Ground squirrel -2.29263476
## 12    body    Arctic ground squirrel -0.08338161
## 13    body African giant pouched rat  0.00000000
## 14    body Lesser short-tailed shrew -5.29831737
## 15    body           Star-nosed mole -2.81341072
## 16    body     Nine-banded armadillo  1.25276297
## 17    body                Tree hyrax  0.69314718
## 18    body              N.A. opossum  0.53062825
## 19    body            Asian elephant  7.84267147
## 20    body              Big brown bat -3.77226106
## 21    body                    Donkey  5.23164323
## 22    body                     Horse  6.25575004
## 23    body         European hedgehog -0.24207156
## 24    body              Patas monkey  2.30258509
## 25    body                       Cat  1.19392247
## 26    body                    Galago -1.60943791
## 27    body                     Genet  0.34358970
## 28    body                   Giraffe  6.27098843
## 29    body                   Gorilla  5.33271879
## 30    body                 Grey seal  4.44265126
## 31    body              Rock hyrax-a -0.28768207
## 32    body                     Human  4.12713439
## 33    body          African elephant  8.80297346
```

```
## 34    body             Water opossum  1.25276297
## 35    body             Rhesus monkey  1.91692261
## 36    body                  Kangaroo  3.55534806
## 37    body     Yellow-bellied marmot  1.39871688
## 38    body            Golden hamster -2.12026354
## 39    body                     Mouse -3.77226106
## 40    body          Little brown bat -4.60517019
## 41    body                Slow loris  0.33647224
## 42    body                     Okapi  5.52146092
## 43    body                    Rabbit  0.91629073
## 44    body                     Sheep  4.01638302
## 45    body                    Jaguar  4.60517019
## 46    body                Chimpanzee  3.95431592
## 47    body                    Baboon  2.35612586
## 48    body           Desert hedgehog -0.59783700
## 49    body           Giant armadillo  4.09434456
## 50    body              Rock hyrax-b  1.28093385
## 51    body                   Raccoon  1.45582042
## 52    body                       Rat -1.27296568
## 53    body          E. American mole -2.59026717
## 54    body                  Mole rat -2.10373423
## 55    body                Musk shrew -3.03655427
## 56    body                       Pig  5.25749537
## 57    body                   Echidna  1.09861229
## 58    body           Brazilian tapir  5.07517382
## 59    body                    Tenrec -0.10536052
## 60    body                 Phalanger  0.48242615
## 61    body                Tree shrew -2.26336438
## 62    body                   Red fox  1.44338333
## 63   brain                Arctic fox  3.79548919
## 64   brain                Owl monkey  2.74084002
## 65   brain           Mountain beaver  2.09186406
## 66   brain                       Cow  6.04737218
## 67   brain                 Grey wolf  4.78331637
## 68   brain                      Goat  4.74493213
## 69   brain                  Roe deer  4.58700622
## 70   brain                Guinea pig  1.70474809
## 71   brain                    Verbet  4.06044301
## 72   brain                Chinchilla  1.85629799
## 73   brain           Ground squirrel  1.38629436
## 74   brain    Arctic ground squirrel  1.74046617
## 75   brain African giant pouched rat  1.88706965
## 76   brain Lesser short-tailed shrew -1.96611286
## 77   brain           Star-nosed mole  0.00000000
## 78   brain     Nine-banded armadillo  2.37954613
## 79   brain                Tree hyrax  2.50959926
## 80   brain              N.A. opossum  1.84054963
## 81   brain            Asian elephant  8.43446354
## 82   brain              Big brown bat -1.20397280
## 83   brain                    Donkey  6.03787092
## 84   brain                     Horse  6.48463524
## 85   brain         European hedgehog  1.25276297
## 86   brain              Patas monkey  4.74493213
## 87   brain                       Cat  3.24259235
```

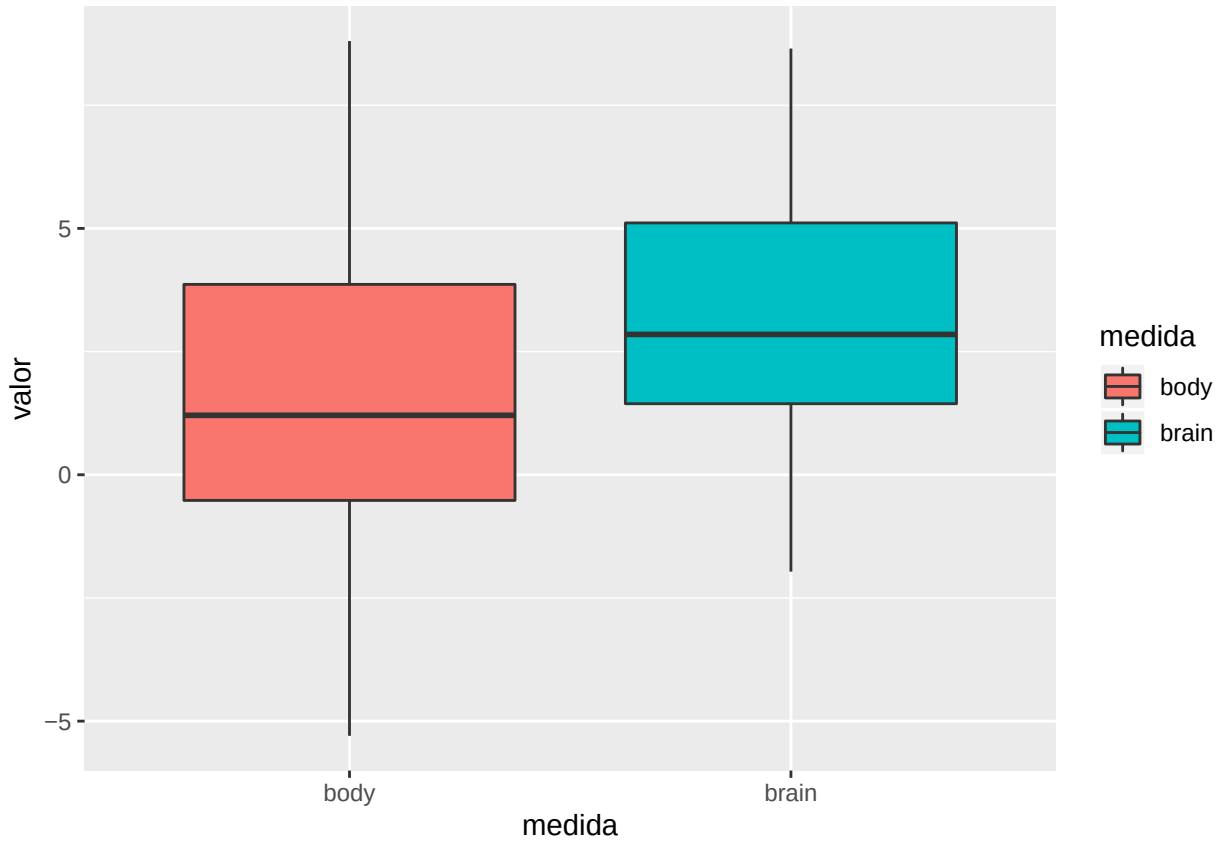
```
## 88   brain                   Galago  1.60943791
## 89   brain                    Genet  2.86220088
## 90   brain                  Giraffe  6.52209280
## 91   brain                  Gorilla  6.00635316
## 92   brain                Grey seal  5.78382518
## 93   brain             Rock hyrax-a  2.50959926
## 94   brain                    Human  7.18538702
## 95   brain          African elephant  8.65032450
## 96   brain             Water opossum  1.36097655
## 97   brain             Rhesus monkey  5.18738581
## 98   brain                  Kangaroo  4.02535169
## 99   brain     Yellow-bellied marmot  2.83321334
## 100  brain            Golden hamster  0.00000000
## 101  brain                    Mouse -0.91629073
## 102  brain          Little brown bat -1.38629436
## 103  brain                Slow loris  2.52572864
## 104  brain                    Okapi  6.19440539
## 105  brain                   Rabbit  2.49320545
## 106  brain                    Sheep  5.16478597
## 107  brain                   Jaguar  5.05624581
## 108  brain                Chimpanzee  6.08677473
## 109  brain                   Baboon  5.19017521
## 110  brain           Desert hedgehog  0.87546874
## 111  brain           Giant armadillo  4.39444915
## 112  brain             Rock hyrax-b  3.04452244
## 113  brain                  Raccoon  3.66867675
## 114  brain                      Rat  0.64185389
## 115  brain          E. American mole  0.18232156
## 116  brain                 Mole rat  1.09861229
## 117  brain                Musk shrew -1.10866262
## 118  brain                      Pig  5.19295685
## 119  brain                  Echidna  3.21887582
## 120  brain           Brazilian tapir  5.12989871
## 121  brain                   Tenrec  0.95551145
## 122  brain                Phalanger  2.43361336
## 123  brain                Tree shrew  0.91629073
## 124  brain                   Red fox  3.91999118
```

```
dim(df_mammals_gg)
```

```
## [1] 124   3
```

```
#install.packages("ggplot2")
library(ggplot2)
gf <- ggplot(data=df_mammals_gg)  #objeto de ggplot
gf
```

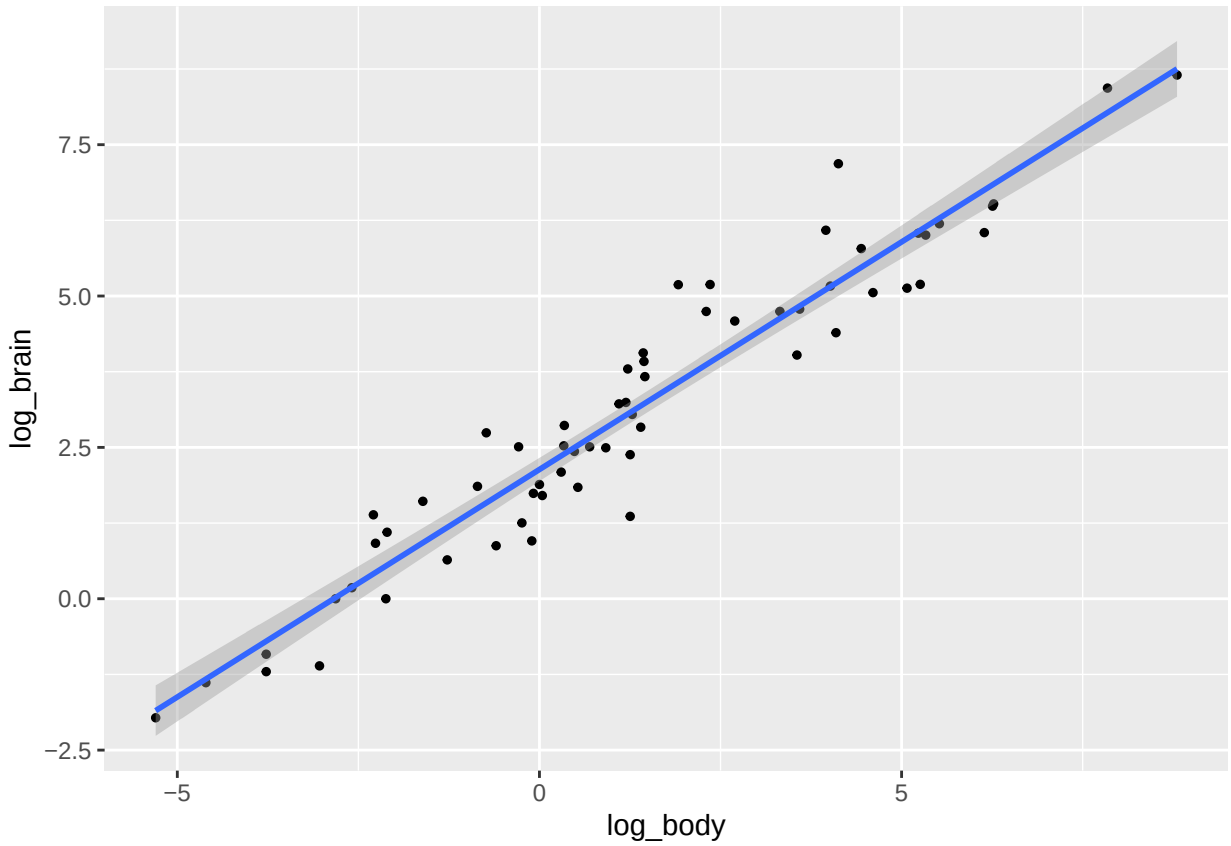
```
gf + geom_boxplot(aes(x=medida, y=valor, fill = medida)) #bloxplot con ggplot2
```




```
options(digits=3)#para sólo imprimir 3 número de dígitos
cor(log_mammals) #correlación de las logaritmos del tamaño del cuerpo y del cerebro
```

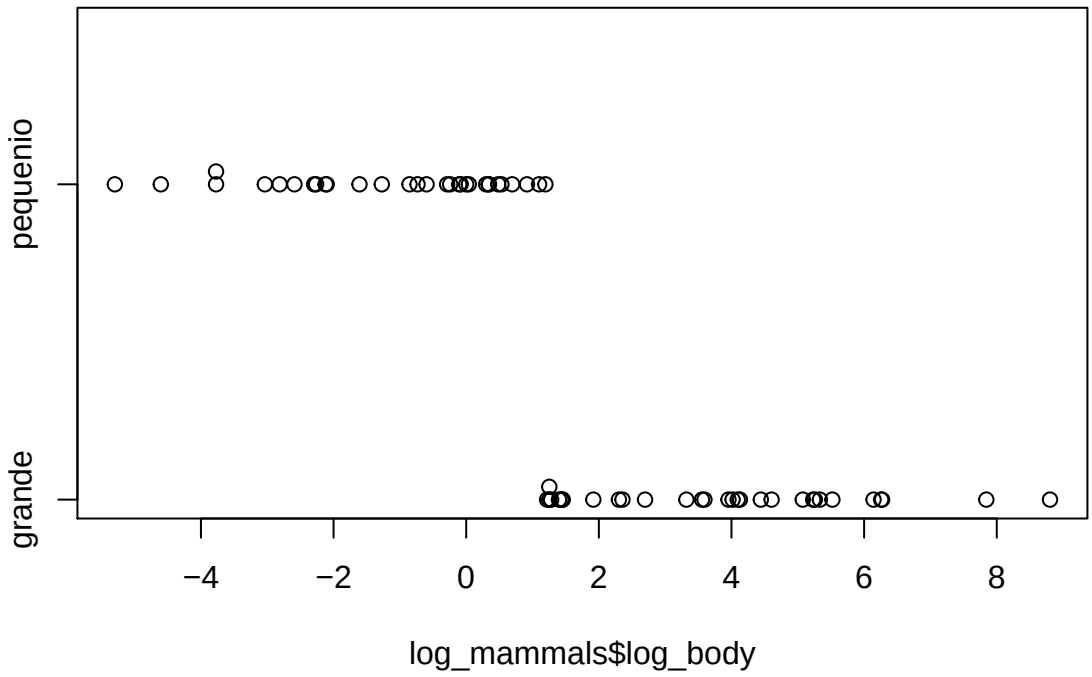
```
##           log_body log_brain
## log_body      1.00      0.96
## log_brain     0.96      1.00
```

```
ggplot(log_mammals)+geom_point(aes(x=log_body, y=log_brain),size=1) + geom_smooth(aes(x=log_body, y=log
```

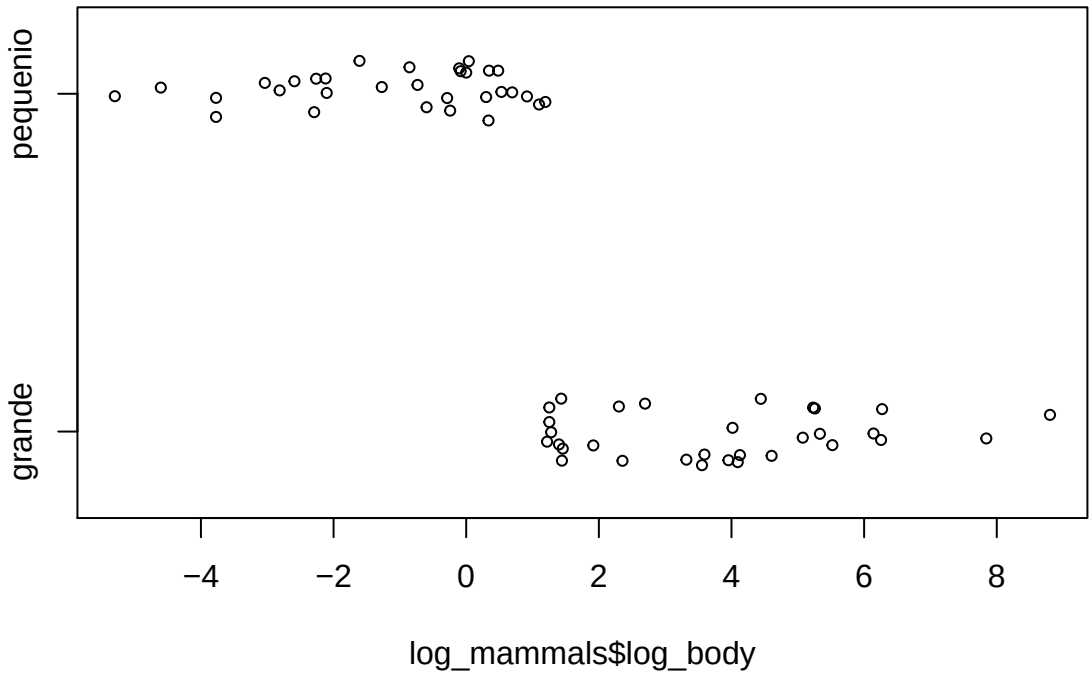



```
mediana_body = median(log_mammals$log_body)
log_mammals$size_body = ifelse(log_mammals$log_body >= mediana_body, "grande", "pequenio")
#añadimos también la de brain:
mediana_brain = median(log_mammals$log_brain)
log_mammals$size_brain = ifelse(log_mammals$log_brain >= mediana_brain, "grande", "pequenio")

stripchart(log_mammals$log_body~log_mammals$size_body, method="stack", pch=1, cex=1) #stack grafica los
```




```
stripchart(log_mammals$log_body~log_mammals$size_body, method="jitter", jitter=.1, pch=1, cex=.7)
```

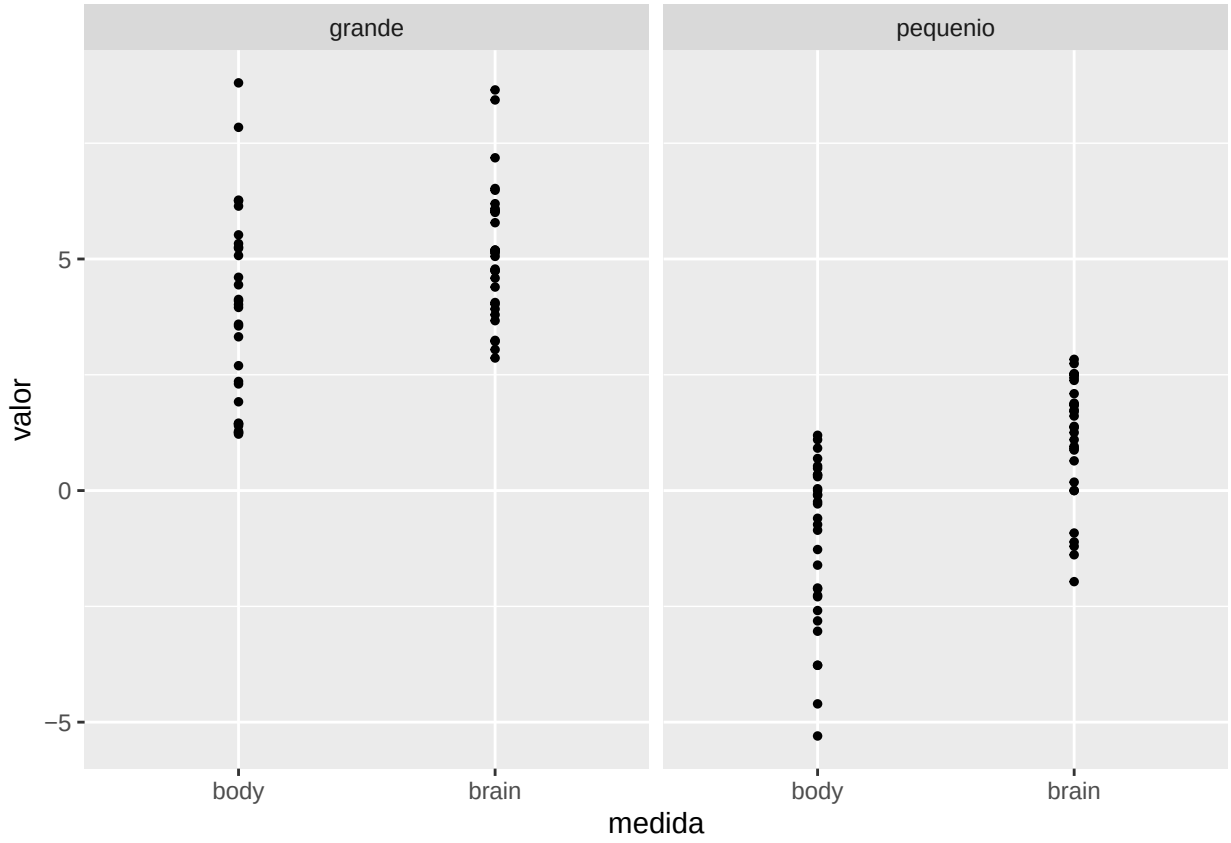



```
df_mammals_body$size_body <- ifelse(df_mammals_body$valor >= mediana_body, "grande", "pequenio") #crean
df_mammals_brain$size_brain <- ifelse(df_mammals_brain$valor >= mediana_brain, "grande", "pequenio") #c
df_mammals_gg$size <-c(df_mammals_body$size_body, df_mammals_brain$size_brain)
head(df_mammals_gg)
```

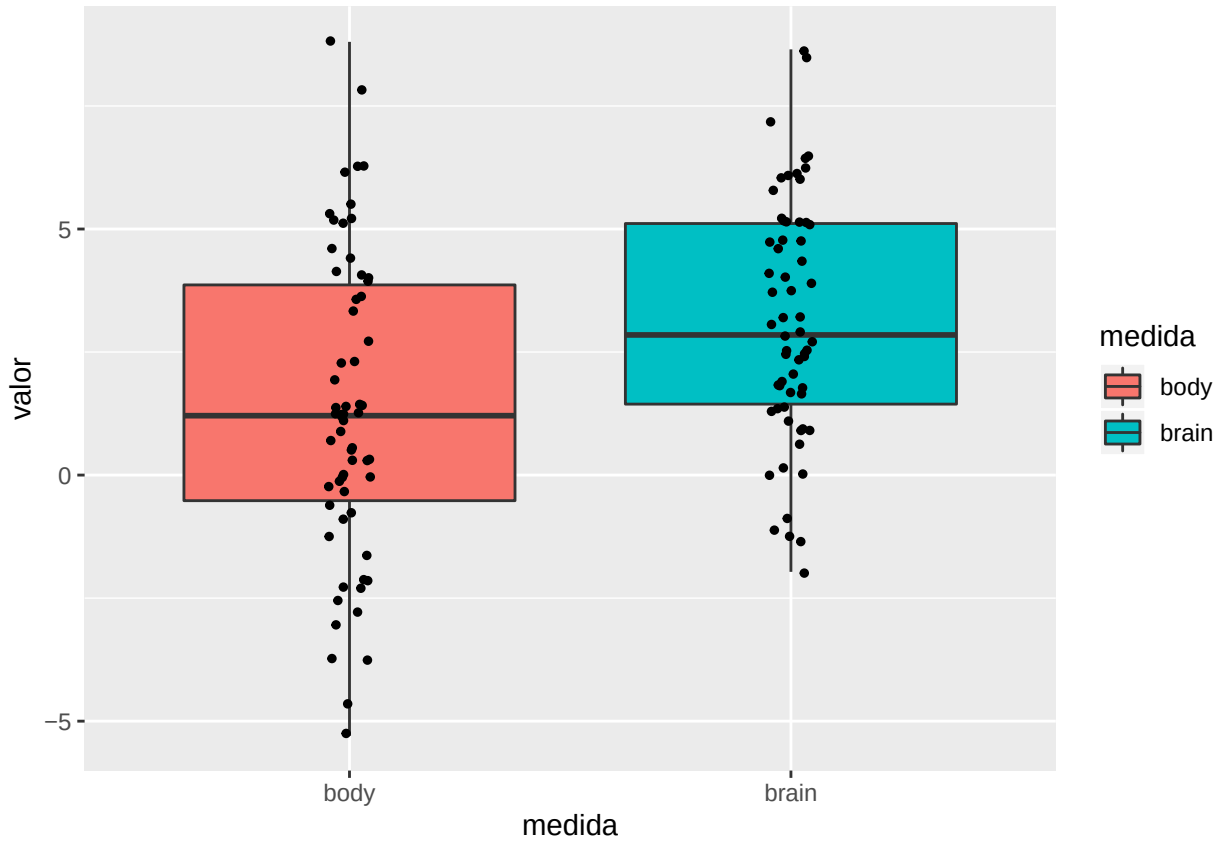
```
##   medida          mammal  valor     size
## 1   body      Arctic fox  1.219   grande
## 2   body      Owl monkey -0.734 pequenio
## 3   body Mountain beaver  0.300 pequenio
```

```
## 4   body             Cow  6.142   grande
## 5   body       Grey wolf  3.593   grande
## 6   body            Goat  3.320   grande
```

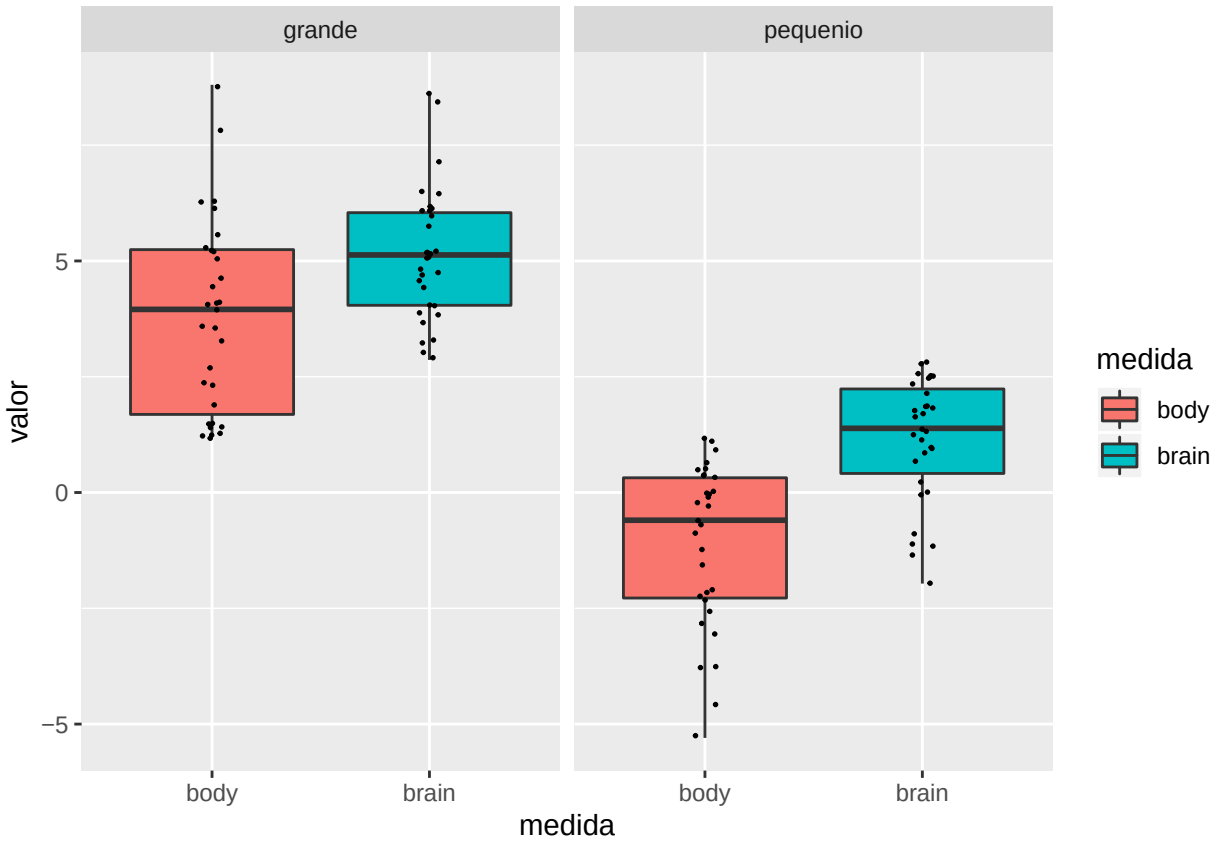
```
gf <- ggplot(data=df_mammals_gg)
gf + geom_point(aes(x=medida,y=valor),size=1)+ facet_wrap(~size) #stripchart or dotplot with ggplot2
```




```
gf + geom_boxplot(aes(x=medida, y=valor, fill = medida)) + geom_jitter(aes(x=medida,y=valor),position=po
```




```
gf + geom_boxplot(aes(x=medida, y=valor, fill = medida)) + geom_jitter(aes(x=medida,y=valor),position=p
```




```
summary(log_mammals)
```

```
##     log_body          log_brain        size_body          size_brain
##  Min.   :-5.30   Min.   :-1.97   Length:62          Length:62
##  1st Qu.:-0.52   1st Qu.: 1.44   Class :character   Class :character
##  Median : 1.21   Median : 2.85   Mode  :character   Mode  :character
##  Mean   : 1.34   Mean   : 3.14
##  3rd Qu.: 3.86   3rd Qu.: 5.11
##  Max.   : 8.80   Max.   : 8.65
```

a) Para datos cuantitativos es común realizar diagramas de tallos y hojas. Investigar la interpretación de estos diagramas y usar la función stem del paquete base de R para realizar tales diagramas a las variables log_body y log_brain e interpretarlas.

```
stem(log_mammals$log_body, scale = 1, width = 80, atom = 1e-08)
```

```
##
##   The decimal point is at the |
##
##   -4 | 36
##   -2 | 880863311
##   -0 | 639763211
##    0 | 00333557912233344459
##    2 | 347366
##    4 | 00114612335
##    6 | 1338
##    8 | 8
```

```
stem(log_mammals$log_brain, scale = 1, width = 80, atom = 1e-08)
```

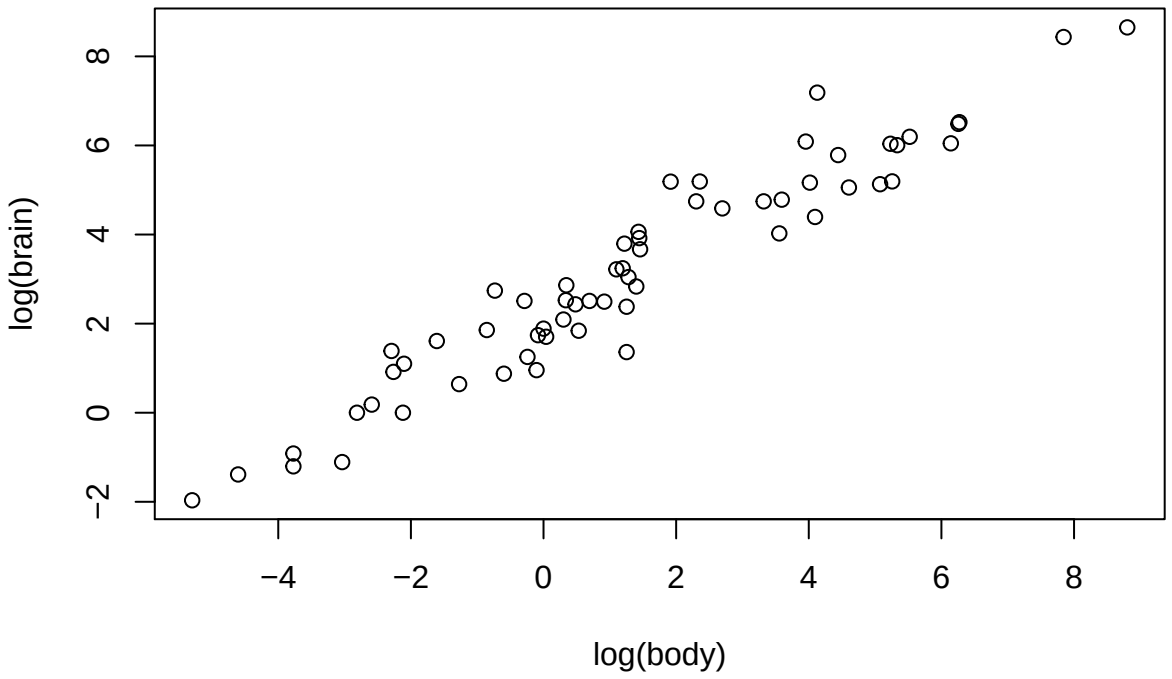
```
##
##   The decimal point is at the |
##
##   -2 | 0
##   -0 | 4219
##    0 | 00269901344677899
##    2 | 1445555789022789
##    4 | 01467781122228
##    6 | 00012552
##    8 | 47
```

Interpretación El gráfico de "tallos y hojas" permite obtener simultáneamente una distribución de frecuencias de la variable y su representación gráfica. En este caso, observamos que los datos se concentran alrededor del 0. Además, los datos correspondientes a la variable log_body son más dispersos que los datos correspondientes a la variable log_brain.

b) Scatterplot, coloreando diferente a cada grupo que se creó: mamífero pequeño y mamífero grande con la variable size_body.

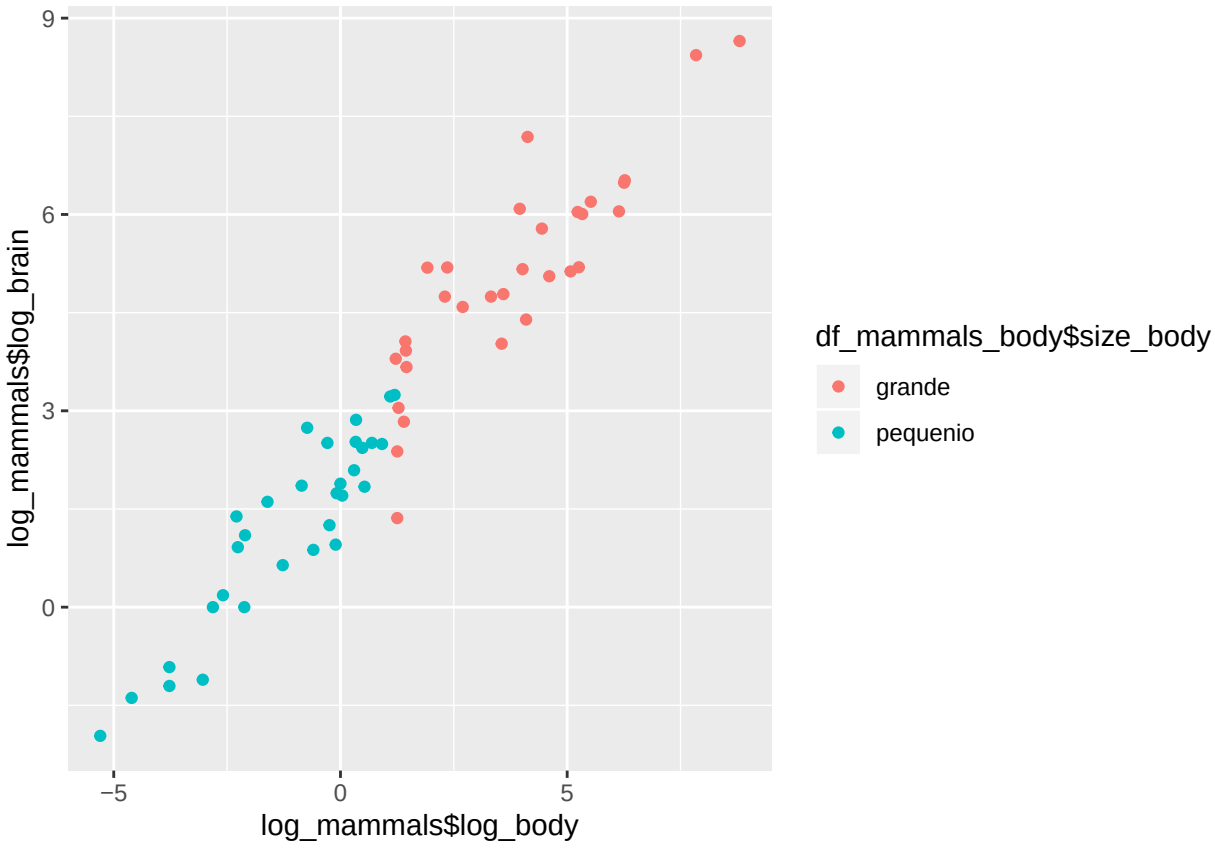
Con plot:

```
plot(log(mammals$body),log(mammals$brain), xlab="log(body)", ylab="log(brain)")
```



Con ggplot:

```
ggplot(log_mammals) + geom_point(aes(x=log_mammals$log_body, y=log_mammals$log_brain, color=df_mammals_
```



c) Instalar el paquete dplyr para colocar los datos en un formato long y se pueda graficar con el paquete de ggplot2 con la función gather el dataframe log_mammals. Después de instalar tal paquete, realizar mismo enunciado que a) pero con geom_points y paquete ggplot2.

```
tbl_df(log_mammals)
```

```
## # A tibble: 62 x 4
##    log_body log_brain size_body size_brain
##       <dbl>     <dbl> <chr>     <chr>
##  1   1.22        3.80 grande    grande
##  2  -0.734       2.74 pequenio  pequenio
##  3   0.300       2.09 pequenio  pequenio
##  4   6.14        6.05 grande    grande
##  5   3.59        4.78 grande    grande
##  6   3.32        4.74 grande    grande
##  7   2.70        4.59 grande    grande
##  8   0.0392      1.70 pequenio  pequenio
##  9   1.43        4.06 grande    grande
## 10  -0.856       1.86 pequenio  pequenio
## # ... with 52 more rows
```

```
log_mammals_long <- gather(log_mammals, body_part, measurement, log_body:log_brain, factor_key=TRUE)
log_mammals_long
```

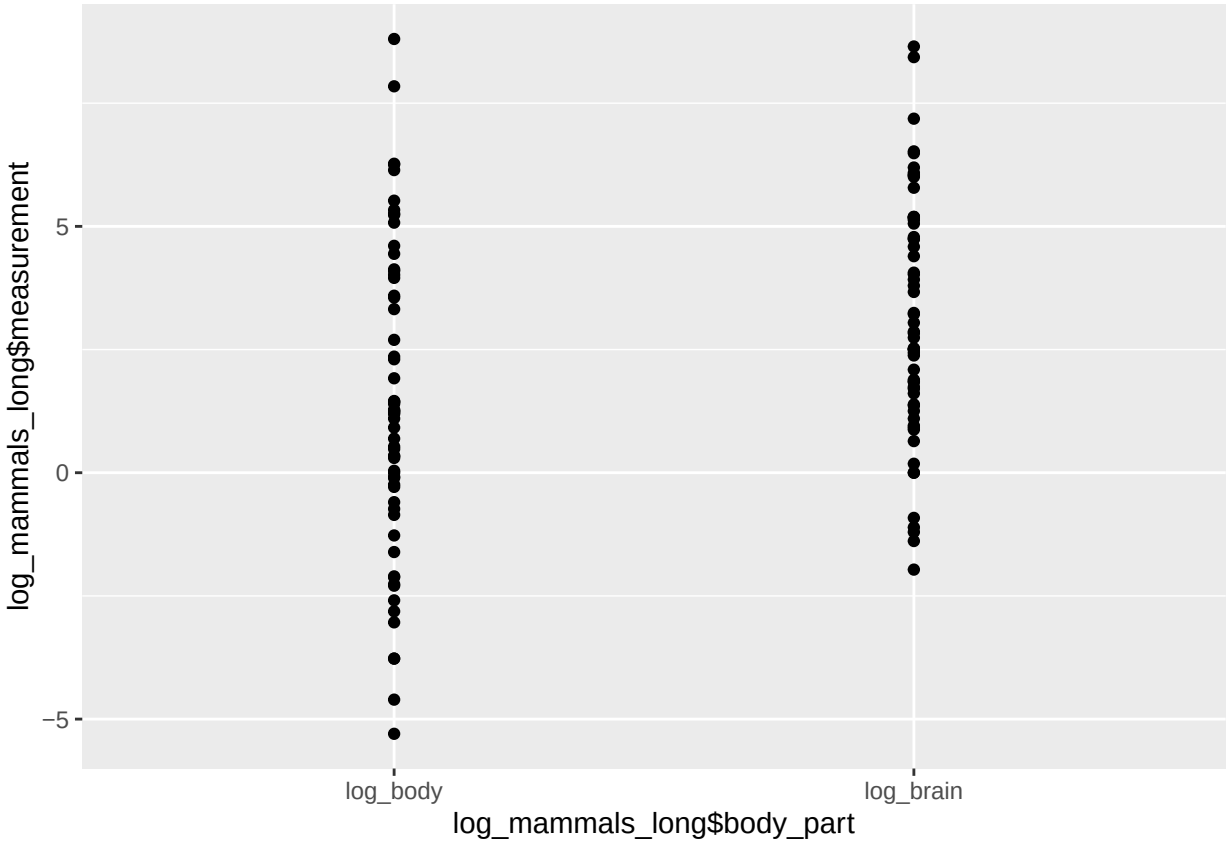
```
##     size_body size_brain body_part measurement
## 1      grande     grande  log_body      1.2194
## 2    pequenio   pequenio  log_body     -0.7340
## 3    pequenio   pequenio  log_body      0.3001
```

```
## 4      grande     grande  log_body      6.1420
## 5      grande     grande  log_body      3.5926
## 6      grande     grande  log_body      3.3200
## 7      grande     grande  log_body      2.6967
## 8    pequenio   pequenio  log_body      0.0392
## 9      grande     grande  log_body      1.4327
## 10   pequenio   pequenio  log_body     -0.8557
## 11   pequenio   pequenio  log_body     -2.2926
## 12   pequenio   pequenio  log_body     -0.0834
## 13   pequenio   pequenio  log_body      0.0000
## 14   pequenio   pequenio  log_body     -5.2983
## 15   pequenio   pequenio  log_body     -2.8134
## 16     grande   pequenio  log_body      1.2528
## 17   pequenio   pequenio  log_body      0.6931
## 18   pequenio   pequenio  log_body      0.5306
## 19     grande     grande  log_body      7.8427
## 20   pequenio   pequenio  log_body     -3.7723
## 21     grande     grande  log_body      5.2316
## 22     grande     grande  log_body      6.2558
## 23   pequenio   pequenio  log_body     -0.2421
## 24     grande     grande  log_body      2.3026
## 25   pequenio     grande  log_body      1.1939
## 26   pequenio   pequenio  log_body     -1.6094
## 27   pequenio     grande  log_body      0.3436
## 28     grande     grande  log_body      6.2710
## 29     grande     grande  log_body      5.3327
## 30     grande     grande  log_body      4.4427
## 31   pequenio   pequenio  log_body     -0.2877
## 32     grande     grande  log_body      4.1271
## 33     grande     grande  log_body      8.8030
## 34     grande   pequenio  log_body      1.2528
## 35     grande     grande  log_body      1.9169
## 36     grande     grande  log_body      3.5553
## 37     grande   pequenio  log_body      1.3987
## 38   pequenio   pequenio  log_body     -2.1203
## 39   pequenio   pequenio  log_body     -3.7723
## 40   pequenio   pequenio  log_body     -4.6052
## 41   pequenio   pequenio  log_body      0.3365
## 42     grande     grande  log_body      5.5215
## 43   pequenio   pequenio  log_body      0.9163
## 44     grande     grande  log_body      4.0164
## 45     grande     grande  log_body      4.6052
## 46     grande     grande  log_body      3.9543
## 47     grande     grande  log_body      2.3561
## 48   pequenio   pequenio  log_body     -0.5978
## 49     grande     grande  log_body      4.0943
## 50     grande     grande  log_body      1.2809
## 51     grande     grande  log_body      1.4558
## 52   pequenio   pequenio  log_body     -1.2730
## 53   pequenio   pequenio  log_body     -2.5903
## 54   pequenio   pequenio  log_body     -2.1037
## 55   pequenio   pequenio  log_body     -3.0366
## 56     grande     grande  log_body      5.2575
## 57   pequenio     grande  log_body      1.0986
```

```
## 58     grande     grande  log_body      5.0752
## 59   pequenio   pequenio  log_body     -0.1054
## 60   pequenio   pequenio  log_body      0.4824
## 61   pequenio   pequenio  log_body     -2.2634
## 62     grande     grande  log_body      1.4434
## 63     grande     grande log_brain      3.7955
## 64   pequenio   pequenio log_brain      2.7408
## 65   pequenio   pequenio log_brain      2.0919
## 66     grande     grande log_brain      6.0474
## 67     grande     grande log_brain      4.7833
## 68     grande     grande log_brain      4.7449
## 69     grande     grande log_brain      4.5870
## 70   pequenio   pequenio log_brain      1.7047
## 71     grande     grande log_brain      4.0604
## 72   pequenio   pequenio log_brain      1.8563
## 73   pequenio   pequenio log_brain      1.3863
## 74   pequenio   pequenio log_brain      1.7405
## 75   pequenio   pequenio log_brain      1.8871
## 76   pequenio   pequenio log_brain     -1.9661
## 77   pequenio   pequenio log_brain      0.0000
## 78     grande   pequenio log_brain      2.3795
## 79   pequenio   pequenio log_brain      2.5096
## 80   pequenio   pequenio log_brain      1.8405
## 81     grande     grande log_brain      8.4345
## 82   pequenio   pequenio log_brain     -1.2040
## 83     grande     grande log_brain      6.0379
## 84     grande     grande log_brain      6.4846
## 85   pequenio   pequenio log_brain      1.2528
## 86     grande     grande log_brain      4.7449
## 87   pequenio     grande log_brain      3.2426
## 88   pequenio   pequenio log_brain      1.6094
## 89   pequenio     grande log_brain      2.8622
## 90     grande     grande log_brain      6.5221
## 91     grande     grande log_brain      6.0064
## 92     grande     grande log_brain      5.7838
## 93   pequenio   pequenio log_brain      2.5096
## 94     grande     grande log_brain      7.1854
## 95     grande     grande log_brain      8.6503
## 96     grande   pequenio log_brain      1.3610
## 97     grande     grande log_brain      5.1874
## 98     grande     grande log_brain      4.0254
## 99     grande   pequenio log_brain      2.8332
## 100  pequenio   pequenio log_brain      0.0000
## 101  pequenio   pequenio log_brain     -0.9163
## 102  pequenio   pequenio log_brain     -1.3863
## 103  pequenio   pequenio log_brain      2.5257
## 104    grande     grande log_brain      6.1944
## 105  pequenio   pequenio log_brain      2.4932
## 106    grande     grande log_brain      5.1648
## 107    grande     grande log_brain      5.0562
## 108    grande     grande log_brain      6.0868
## 109    grande     grande log_brain      5.1902
## 110  pequenio   pequenio log_brain      0.8755
## 111    grande     grande log_brain      4.3944
```

```
## 112    grande     grande log_brain      3.0445
## 113    grande     grande log_brain      3.6687
## 114  pequenio   pequenio log_brain      0.6419
## 115  pequenio   pequenio log_brain      0.1823
## 116  pequenio   pequenio log_brain      1.0986
## 117  pequenio   pequenio log_brain     -1.1087
## 118    grande     grande log_brain      5.1930
## 119  pequenio     grande log_brain      3.2189
## 120    grande     grande log_brain      5.1299
## 121  pequenio   pequenio log_brain      0.9555
## 122  pequenio   pequenio log_brain      2.4336
## 123  pequenio   pequenio log_brain      0.9163
## 124    grande     grande log_brain      3.9200
```

```
ggplot(log_mammals_long, aes(x=log_mammals_long$body_part, y=log_mammals_long$measurement)) + geom_poin
```



d) Calcular estadísticas como el promedio, mediana, máximo y mínimo por grupos de mamífero pequeño y mamífero grande con dplyr y funciones como group_by, sumarise y el operador %>% (pipe).Ver como ayuda: Data Wrangling with dplyr and tidyr o Data Transformation with dplyr o bien como ejemplo de uso: https://genomicsclass.github.io/book/pages/dplyr_tutorial.html u otro tutorial en la red de dplyr.

agrupando los datos según el valor de la función: df_mammals_gg

```
log_mammals %>% group_by(df_mammals_body$size_body) %>%

summarize(average_log_size_body = mean(df_mammals_gg$valor), mediana_log_mammals_size_body = median(df_
```

```
## # A tibble: 2 x 5
##   `df_mammals_bod~ average_log_siz~ mediana_log_mam~ min_log_mammals~
##   <chr>                       <dbl>            <dbl>            <dbl>
## 1 grande                       2.24             2.00            -5.30
## 2 pequenio                     2.24             2.00            -5.30
## # ... with 1 more variable: max_log_mammlas_size_body <dbl>
```

```
log_mammals %>% group_by(df_mammals_body$size_body) %>%

summarize(average_log_size_brain = mean(log_brain), mediana_log_mammals_size_brain = median(log_brain),
```

```
## # A tibble: 2 x 5
##   `df_mammals_bod~ average_log_siz~ mediana_log_mam~ min_log_mammals~
##   <chr>                       <dbl>            <dbl>            <dbl>
## 1 grande                       5.05             5.13             1.36
## 2 pequenio                     1.23             1.61            -1.97
## # ... with 1 more variable: max_log_mammlas_size_brain <dbl>
```

¿Qué mamíferos en cada grupo están en los tres primeros lugares (pensando que se ordenan de forma decreciente en log_brain)?.

```
df_mammals_brain %>%
  select(mammal,valor,size_brain) %>%
  arrange(valor) %>%
  head
```

```
##                      mammal  valor size_brain
## 1 Lesser short-tailed shrew -1.966   pequenio
## 2          Little brown bat -1.386   pequenio
## 3             Big brown bat -1.204   pequenio
## 4                Musk shrew -1.109   pequenio
## 5                     Mouse -0.916   pequenio
## 6           Star-nosed mole  0.000   pequenio
```

Entonces, de los animales con cerebro pequeño, los animales con el cerebro más pequeño son: -> Lesser short-tailed shrew -> Little brown bat -> Big Brown bat

```
df_mammals_brain %>%
  select(mammal,valor,size_brain) %>%
  arrange(desc(valor)) %>%
  head
```

```
##             mammal valor size_brain
## 1 African elephant  8.65     grande
## 2   Asian elephant  8.43     grande
## 3            Human  7.19     grande
## 4          Giraffe  6.52     grande
## 5            Horse  6.48     grande
## 6            Okapi  6.19     grande
```

Entonces, de los animales con cerebro grande, los primeros animales con los cerebros más grandes son:

-> African Elephant -> Asian Elephant -> Human

e) Crea una nueva variable r que sea el cociente entre brain y body.

Ordena en orden creciente el dataset de mammals de acuerdo a esta nueva variable.

```
r <- mammals$brain/mammals$body
r
```

```
##  [1] 13.146 32.292  6.000  0.910  3.289  4.158  6.622  5.288 13.842 15.059
## [11] 39.604  6.196  6.600 28.000 16.667  3.086  6.150  3.706  1.807 13.043
## [21]  2.239  1.257  4.459 11.500  7.758 25.000 12.411  1.285  1.961  3.824
## [31] 16.400 21.290  0.858  1.114 26.324  1.600  4.198  8.333 17.391 25.000
## [41]  8.929  1.960  4.840  3.153  1.570  8.436 17.014  4.364  1.350  5.833
## [51]  9.142  6.786 16.000 24.590  6.875  0.938  8.333  1.056  2.889  7.037
## [61] 24.038 11.901
```

```
r_mammals_brain_body <- data.frame(medida = rep('r', times=nrow(mammals)), mammal=rownames(log_mammals)
r_mammals_brain_body
```

```
##    medida                    mammal  valor
## 1       r                Arctic fox 13.146
## 2       r                Owl monkey 32.292
## 3       r           Mountain beaver  6.000
## 4       r                       Cow  0.910
## 5       r                 Grey wolf  3.289
## 6       r                      Goat  4.158
## 7       r                  Roe deer  6.622
## 8       r                Guinea pig  5.288
## 9       r                    Verbet 13.842
## 10      r                Chinchilla 15.059
## 11      r           Ground squirrel 39.604
## 12      r    Arctic ground squirrel  6.196
## 13      r African giant pouched rat  6.600
## 14      r Lesser short-tailed shrew 28.000
## 15      r           Star-nosed mole 16.667
## 16      r     Nine-banded armadillo  3.086
## 17      r                Tree hyrax  6.150
## 18      r              N.A. opossum  3.706
## 19      r            Asian elephant  1.807
## 20      r             Big brown bat 13.043
## 21      r                    Donkey  2.239
## 22      r                     Horse  1.257
## 23      r         European hedgehog  4.459
## 24      r              Patas monkey 11.500
## 25      r                       Cat  7.758
## 26      r                    Galago 25.000
## 27      r                     Genet 12.411
## 28      r                   Giraffe  1.285
## 29      r                   Gorilla  1.961
## 30      r                 Grey seal  3.824
## 31      r              Rock hyrax-a 16.400
## 32      r                     Human 21.290
## 33      r          African elephant  0.858
## 34      r             Water opossum  1.114
## 35      r             Rhesus monkey 26.324
## 36      r                  Kangaroo  1.600
## 37      r     Yellow-bellied marmot  4.198
## 38      r            Golden hamster  8.333
## 39      r                     Mouse 17.391
## 40      r          Little brown bat 25.000
```

```
## 41      r                Slow loris  8.929
## 42      r                     Okapi  1.960
## 43      r                    Rabbit  4.840
## 44      r                     Sheep  3.153
## 45      r                    Jaguar  1.570
## 46      r                Chimpanzee  8.436
## 47      r                    Baboon 17.014
## 48      r            Desert hedgehog  4.364
## 49      r            Giant armadillo  1.350
## 50      r               Rock hyrax-b  5.833
## 51      r                    Raccoon  9.142
## 52      r                        Rat  6.786
## 53      r          E. American mole 16.000
## 54      r                  Mole rat 24.590
## 55      r                 Musk shrew  6.875
## 56      r                        Pig  0.938
## 57      r                   Echidna  8.333
## 58      r            Brazilian tapir  1.056
## 59      r                    Tenrec  2.889
## 60      r                 Phalanger  7.037
## 61      r                Tree shrew 24.038
## 62      r                   Red fox 11.901
```

```
r_mammals_brain_body<- arrange(r_mammals_brain_body, valor) # Orden directo
head(r_mammals_brain_body)
```

```
##   medida           mammal valor
## 1      r African elephant 0.858
## 2      r              Cow 0.910
## 3      r              Pig 0.938
## 4      r  Brazilian tapir 1.056
## 5      r    Water opossum 1.114
## 6      r            Horse 1.257
```

```
tail(r_mammals_brain_body)
```

```
##    medida                    mammal valor
## 57      r                    Galago  25.0
## 58      r          Little brown bat  25.0
## 59      r              Rhesus monkey  26.3
## 60      r Lesser short-tailed shrew  28.0
## 61      r                 Owl monkey  32.3
## 62      r           Ground squirrel  39.6
```

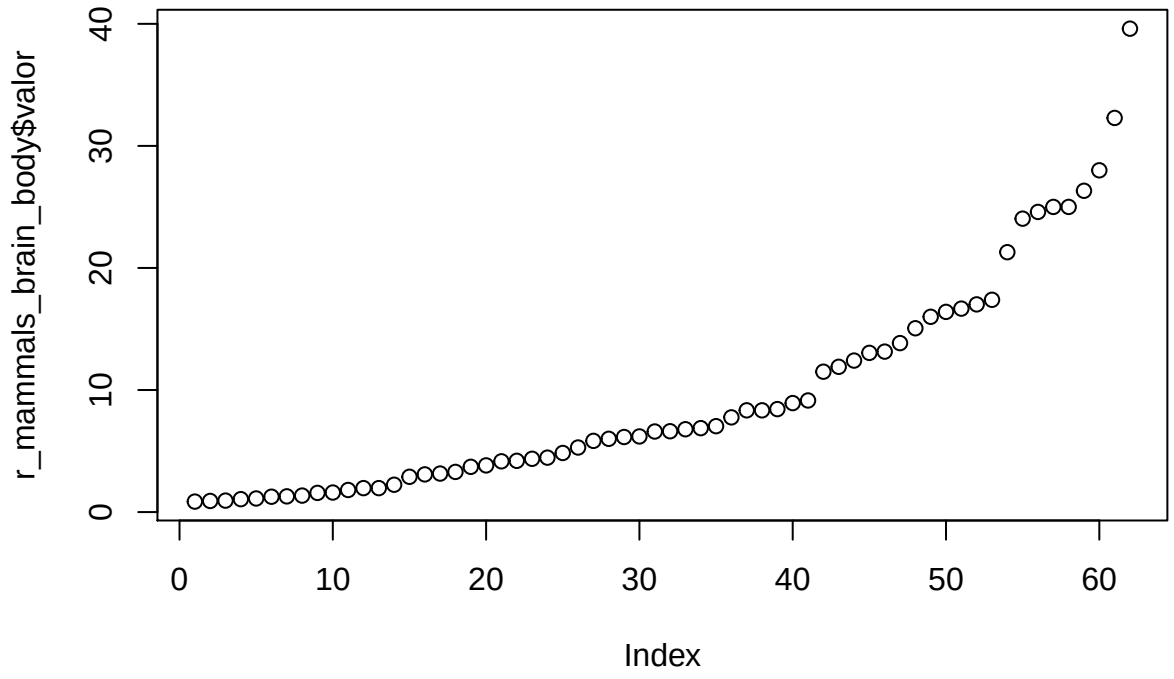
¿Qué mamíferos tienen los cocientes más grandes? ¿y cuáles los más pequeños?

-> Los cinco animales con los cocientes (cerebro/cuerpo) más grandes son: Ground Squirrel, Owl Monkey, Lesser Short-Tailed Shrew, Rhesus Monkey, Little Brown Bat

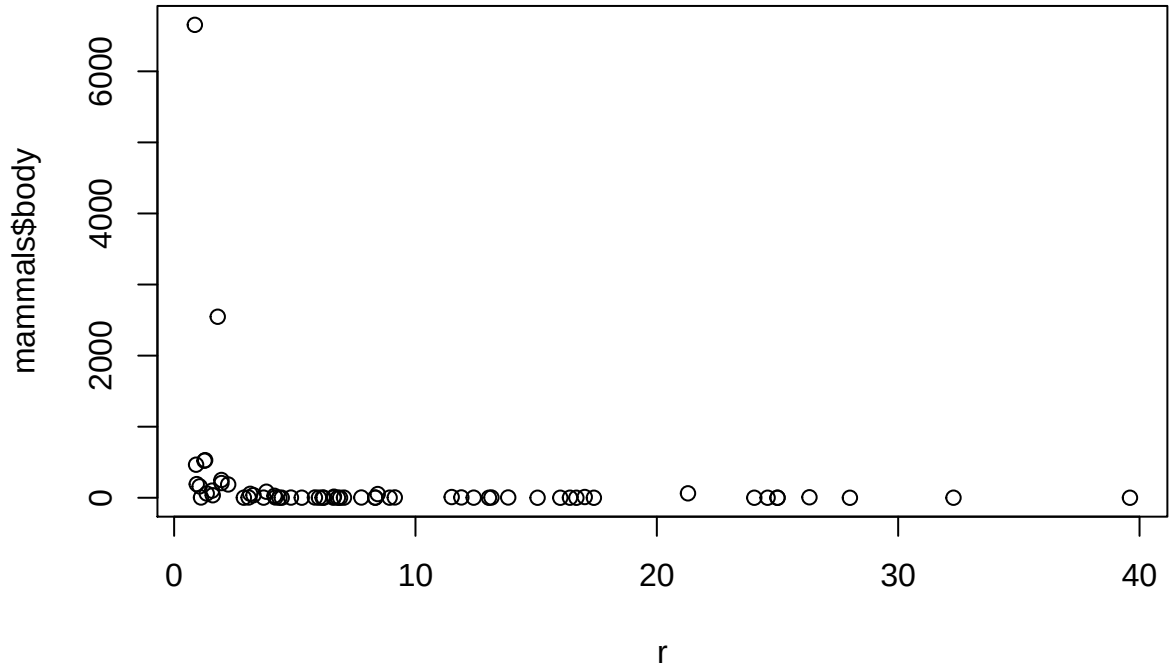
-> Los cinco animales con los cocientes más pequeños son: African Elephant, Cow, Pig, Brazilian Tapir, Water Opossum, Horse

f) Usando e) realiza un scatterplot de r vs body.

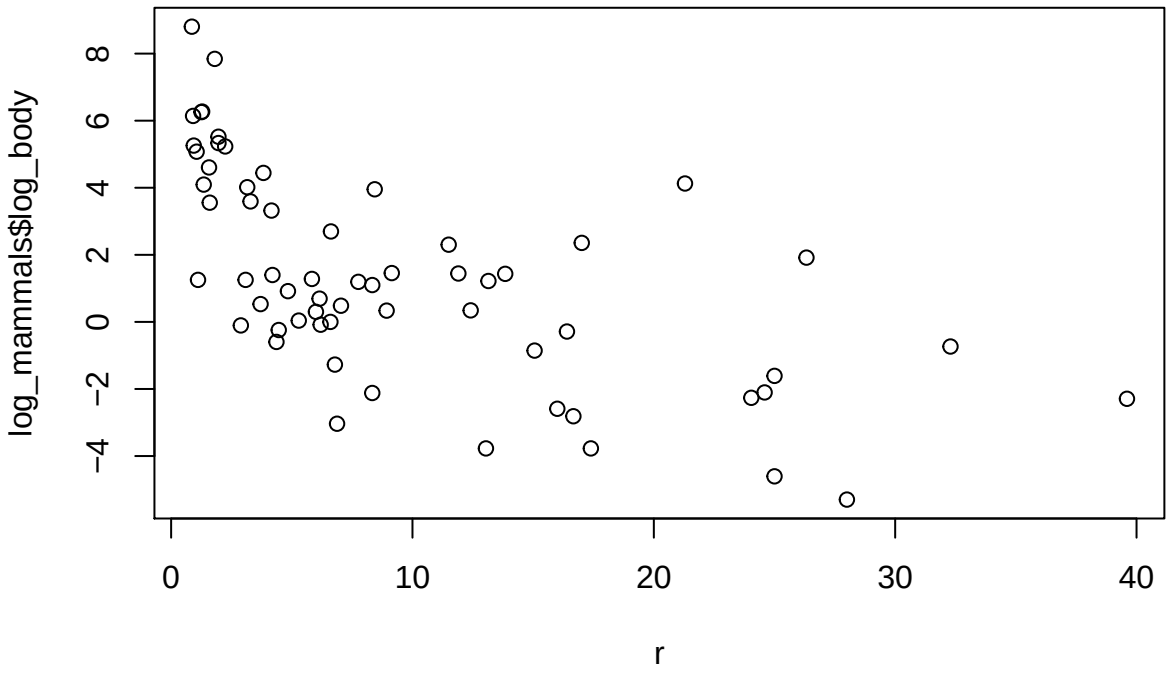
```
plot(r_mammals_brain_body$valor)
```




```
plot(r, mammals$body) #no se puede apreciar muy bien la relación entre r y body porque existen valores
```




```
plot(r, log_mammals$log_body)
```



Ejercicio 3) Considérese el dataset randu (ejecutar ?randu para una descripción)

```
?randu
```

a) Usar la función mean para calcular la media muestral en cada uno de los números que forman la tripleta: x, y, z. Asimismo usar la función var para calcular una matriz de varianzas y covarianzas muestral. Calculando la Media Muestral de X, Y, Z

```
mean(randu$x) #media muestral de x
```

```
## [1] 0.526
```

```
mean(randu$y) #media muestral de y
```

```
## [1] 0.486
```

```
mean(randu$z) #media muestral de z
```

```
## [1] 0.481
```

```
apply(randu, 2 , mean)
```

```
##     x     y     z
## 0.526 0.486 0.481
```

Calculando la varianza

```
var(randu) #matriz de varianzas y covarianzas muestral de la tripleta x, y, z
```

```
##          x        y        z
## x  0.08123 -0.00406  0.00464
## y -0.00406  0.08627 -0.00515
## z  0.00464 -0.00515  0.07786
```

```
var(randu$x) #varianza de x
```

```
## [1] 0.0812
```

```
var(randu$y) #varianza de y
```

```
## [1] 0.0863
```

```
var(randu$z) #varianza de z
```

```
## [1] 0.0779
```

b) Queremos ver la distribución del promedio por renglón de cada observación del dataset randu, para esto utilizar la función de apply para calcular tal promedio, alternativamente usar rowMeans.

```
average_distribution <-apply(randu, 1 , mean)
rowMeans(randu)
```

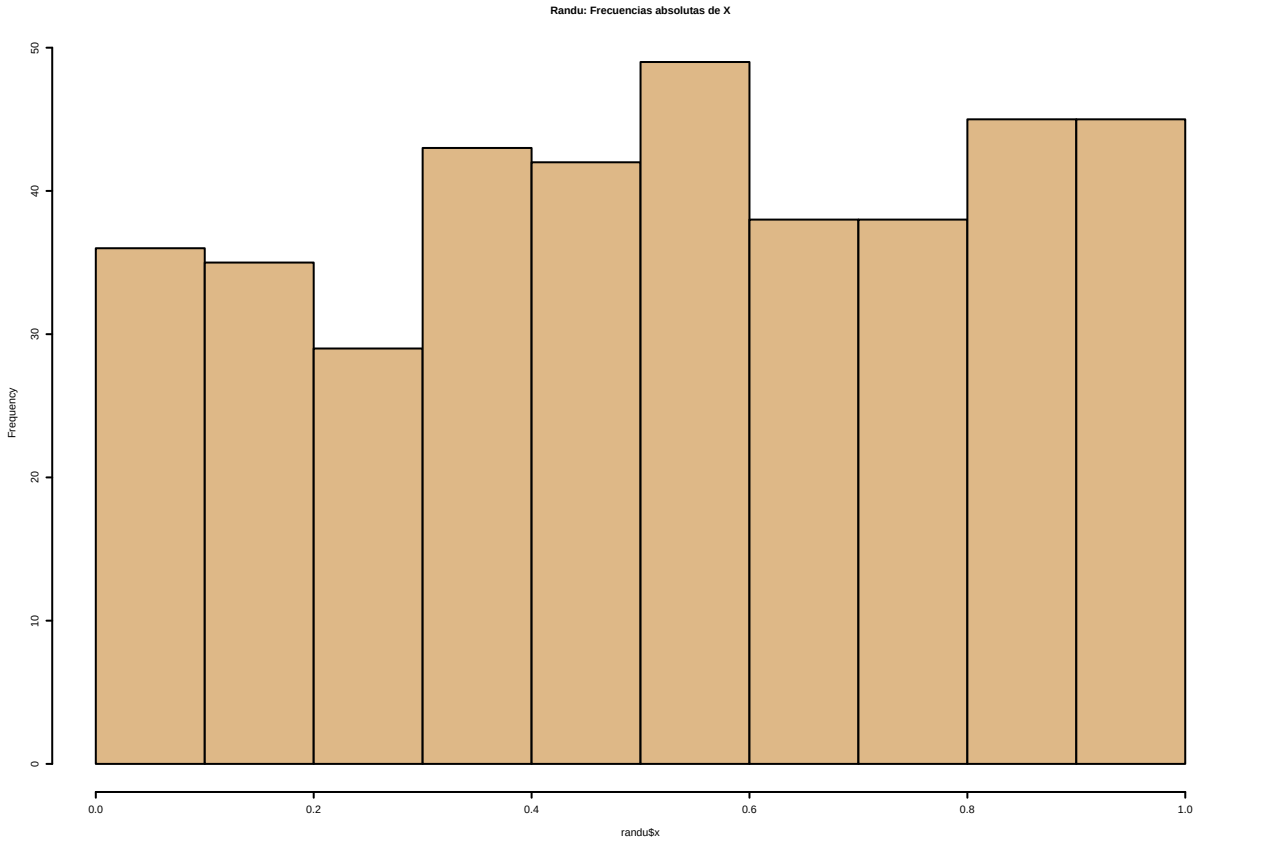
```
##        1        2        3        4        5        6        7        8
## 0.000346 0.244722 0.844799 0.653828 0.546450 0.671117 0.516280 0.785096
##        9       10       11       12       13       14       15       16
## 0.404362 0.707980 0.596615 0.755051 0.399032 0.609959 0.673345 0.127344
##       17       18       19       20       21       22       23       24
## 0.873060 0.753820 0.342786 0.598900 0.553721 0.658799 0.487217 0.688225
##       25       26       27       28       29       30       31       32
## 0.813216 0.561629 0.674451 0.165911 0.613825 0.421261 0.304814 0.435659
##       33       34       35       36       37       38       39       40
## 0.429617 0.228274 0.463840 0.407485 0.436394 0.493968 0.105706 0.404788
##       41       42       43       44       45       46       47       48
## 0.261525 0.460813 0.859058 0.313597 0.585347 0.575920 0.746901 0.515680
##       49       50       51       52       53       54       55       56
## 0.552463 0.769198 0.778883 0.480503 0.284597 0.411201 0.374309 0.557709
##       57       58       59       60       61       62       63       64
## 0.453347 0.492324 0.240888 0.511787 0.537840 0.565024 0.667533 0.340593
##       65       66       67       68       69       70       71       72
## 0.392273 0.657395 0.522900 0.586061 0.415020 0.366432 0.584655 0.678768
##       73       74       75       76       77       78       79       80
## 0.610974 0.672642 0.491126 0.524682 0.505445 0.351016 0.607034 0.542436
##       81       82       83       84       85       86       87       88
## 0.534618 0.520022 0.428274 0.359454 0.537584 0.536948 0.484096 0.375265
##       89       90       91       92       93       94       95       96
## 0.641592 0.447934 0.695017 0.400338 0.498861 0.534361 0.478125 0.578770
##       97       98       99      100      101      102      103      104
## 0.833299 0.830589 0.224781 0.469830 0.588772 0.488055 0.442385 0.496687
##      105      106      107      108      109      110      111      112
## 0.303948 0.860721 0.851870 0.303997 0.672020 0.186875 0.399253 0.566010
##      113      114      115      116      117      118      119      120
## 0.577012 0.408961 0.472927 0.453526 0.564963 0.673083 0.618335 0.301375
##      121      122      123      124      125      126      127      128
## 0.708353 0.716545 0.708388 0.684756 0.488748 0.351733 0.527931 0.446091
##      129      130      131      132      133      134      135      136
## 0.322682 0.588024 0.730614 0.547398 0.355764 0.600047 0.459822 0.623191
##      137      138      139      140      141      142      143      144
## 0.483348 0.417302 0.592006 0.442233 0.556102 0.543459 0.560733 0.153135
##      145      146      147      148      149      150      151      152
## 0.395465 0.380350 0.386473 0.522158 0.577345 0.664984 0.537118 0.371520
##      153      154      155      156      157      158      159      160
```

```
## 0.609035 0.461243 0.910885 0.426564 0.457698 0.581267 0.566627 0.446786
##      161      162      163      164      165      166      167      168
## 0.388743 0.644869 0.493199 0.515706 0.072439 0.318619 0.376223 0.697930
##      169      170      171      172      173      174      175      176
## 0.250413 0.609638 0.653917 0.605078 0.587051 0.422394 0.820313 0.412209
##      177      178      179      180      181      182      183      184
## 0.330798 0.586860 0.600880 0.533126 0.410342 0.566363 0.625535 0.496182
##      185      186      187      188      189      190      191      192
## 0.590709 0.386780 0.224056 0.562285 0.647572 0.662093 0.635553 0.626754
##      193      194      195      196      197      198      199      200
## 0.518136 0.591293 0.257002 0.280026 0.386911 0.468316 0.555846 0.568886
##      201      202      203      204      205      206      207      208
## 0.664712 0.246649 0.604986 0.333222 0.485898 0.700308 0.198503 0.326630
##      209      210      211      212      213      214      215      216
## 0.682164 0.722826 0.506446 0.329201 0.524567 0.639263 0.560345 0.460843
##      217      218      219      220      221      222      223      224
## 0.525443 0.739577 0.527784 0.570110 0.925232 0.575003 0.401050 0.240227
##      225      226      227      228      229      230      231      232
## 0.489182 0.559134 0.391489 0.287299 0.572291 0.415347 0.654615 0.625706
##      233      234      235      236      237      238      239      240
## 0.410817 0.678919 0.304261 0.375691 0.265338 0.128829 0.340809 0.421462
##      241      242      243      244      245      246      247      248
## 0.373718 0.690917 0.451805 0.294824 0.672296 0.687396 0.309971 0.259318
##      249      250      251      252      253      254      255      256
## 0.243643 0.641051 0.660286 0.469895 0.449208 0.509125 0.487793 0.749770
##      257      258      259      260      261      262      263      264
## 0.705199 0.221893 0.557794 0.526132 0.526691 0.400925 0.286787 0.466373
##      265      266      267      268      269      270      271      272
## 0.464963 0.798702 0.569127 0.689839 0.229732 0.334584 0.517741 0.430106
##      273      274      275      276      277      278      279      280
## 0.287946 0.602136 0.410588 0.689435 0.564144 0.389472 0.467838 0.427088
##      281      282      283      284      285      286      287      288
## 0.534045 0.420956 0.418111 0.489830 0.336360 0.528587 0.478790 0.826281
##      289      290      291      292      293      294      295      296
## 0.527842 0.461406 0.375382 0.526798 0.439889 0.541261 0.358289 0.230534
##      297      298      299      300      301      302      303      304
## 0.428388 0.373251 0.508632 0.441358 0.641775 0.485868 0.624805 0.294292
##      305      306      307      308      309      310      311      312
## 0.599001 0.442486 0.314764 0.314141 0.535722 0.632536 0.585705 0.374636
##      313      314      315      316      317      318      319      320
## 0.143720 0.284045 0.556142 0.399772 0.661885 0.572516 0.582048 0.348992
##      321      322      323      324      325      326      327      328
## 0.610431 0.367843 0.422052 0.361238 0.459999 0.827564 0.400043 0.599506
##      329      330      331      332      333      334      335      336
## 0.527329 0.378151 0.523492 0.337604 0.389163 0.625975 0.515842 0.497419
##      337      338      339      340      341      342      343      344
## 0.439371 0.072968 0.387963 0.178382 0.547641 0.511965 0.506994 0.387444
##      345      346      347      348      349      350      351      352
## 0.444736 0.310069 0.454261 0.511244 0.792644 0.455041 0.630575 0.738802
##      353      354      355      356      357      358      359      360
## 0.897954 0.573039 0.567272 0.643057 0.360131 0.459379 0.567976 0.462936
##      361      362      363      364      365      366      367      368
## 0.613023 0.330654 0.306091 0.627600 0.317925 0.339866 0.336362 0.731219
##      369      370      371      372      373      374      375      376
```

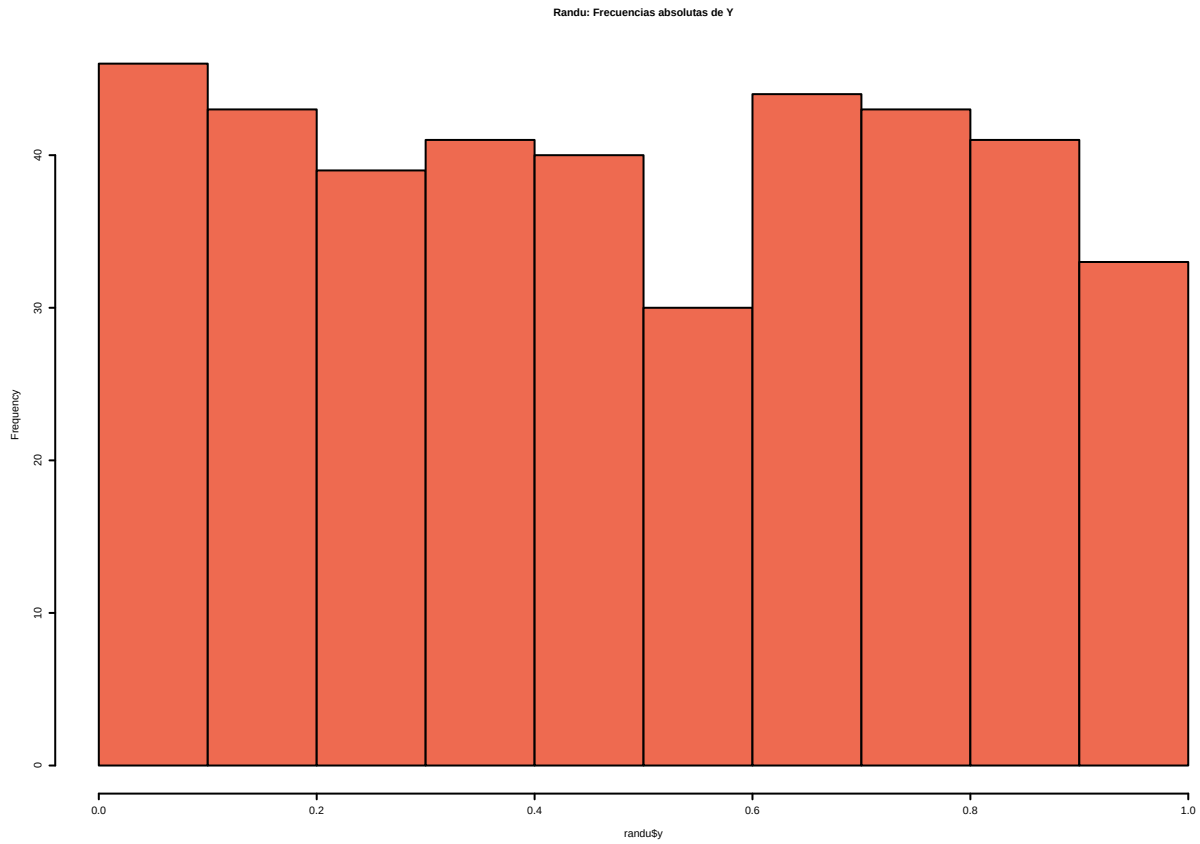
```
## 0.233209 0.229587 0.478819 0.499932 0.185514 0.664803 0.366799 0.292659
##      377      378      379      380      381      382      383      384
## 0.434166 0.787115 0.450685 0.344103 0.387166 0.198622 0.749046 0.291608
##      385      386      387      388      389      390      391      392
## 0.627063 0.583831 0.305506 0.194197 0.538397 0.511252 0.643009 0.585474
##      393      394      395      396      397      398      399      400
## 0.828373 0.623014 0.507144 0.435371 0.565756 0.754168 0.523537 0.562424
```

c) Utilizar la función hist del paquete base para calcular un histograma con los breaks definido por tal función.

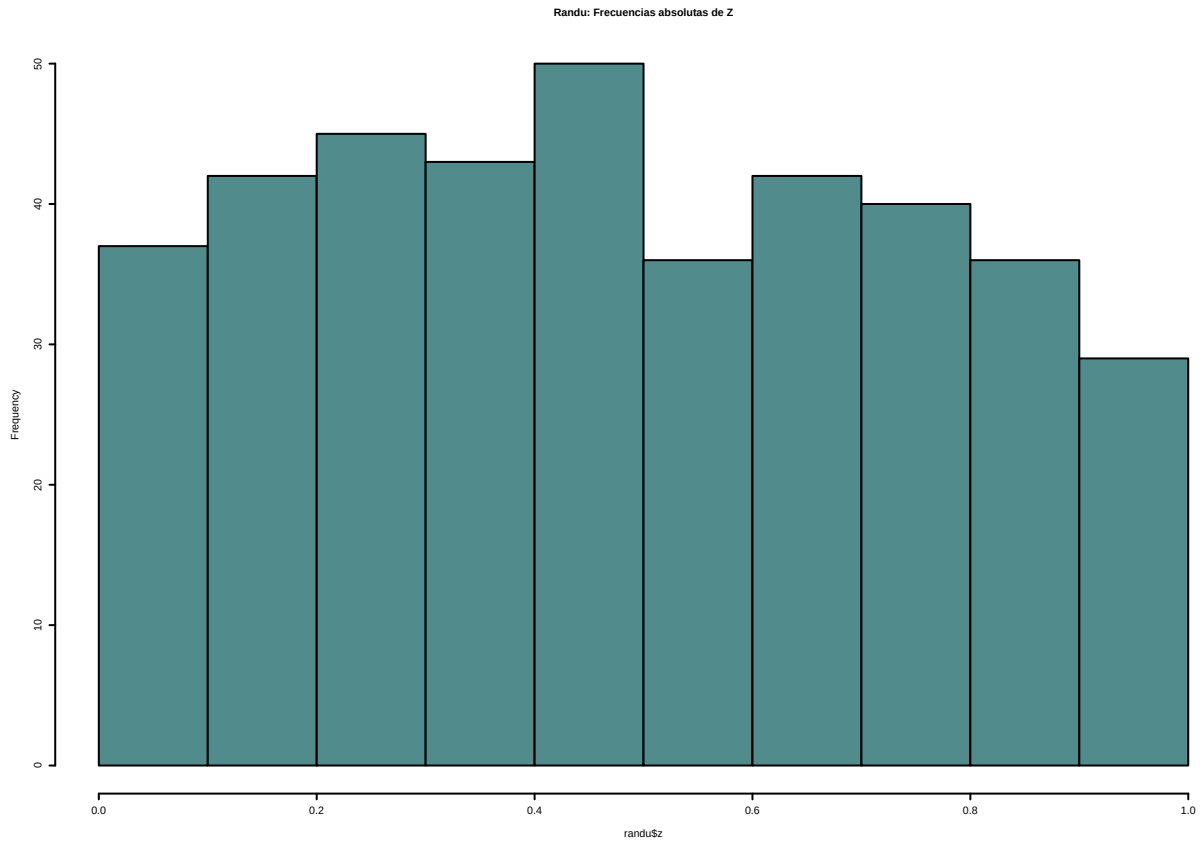
```
par(mfrow=c(1,1)) #subplots
par(cex=0.3) #control size of labels
hist(randu$x, main="Randu: Frecuencias absolutas de X", col='burlywood')
```



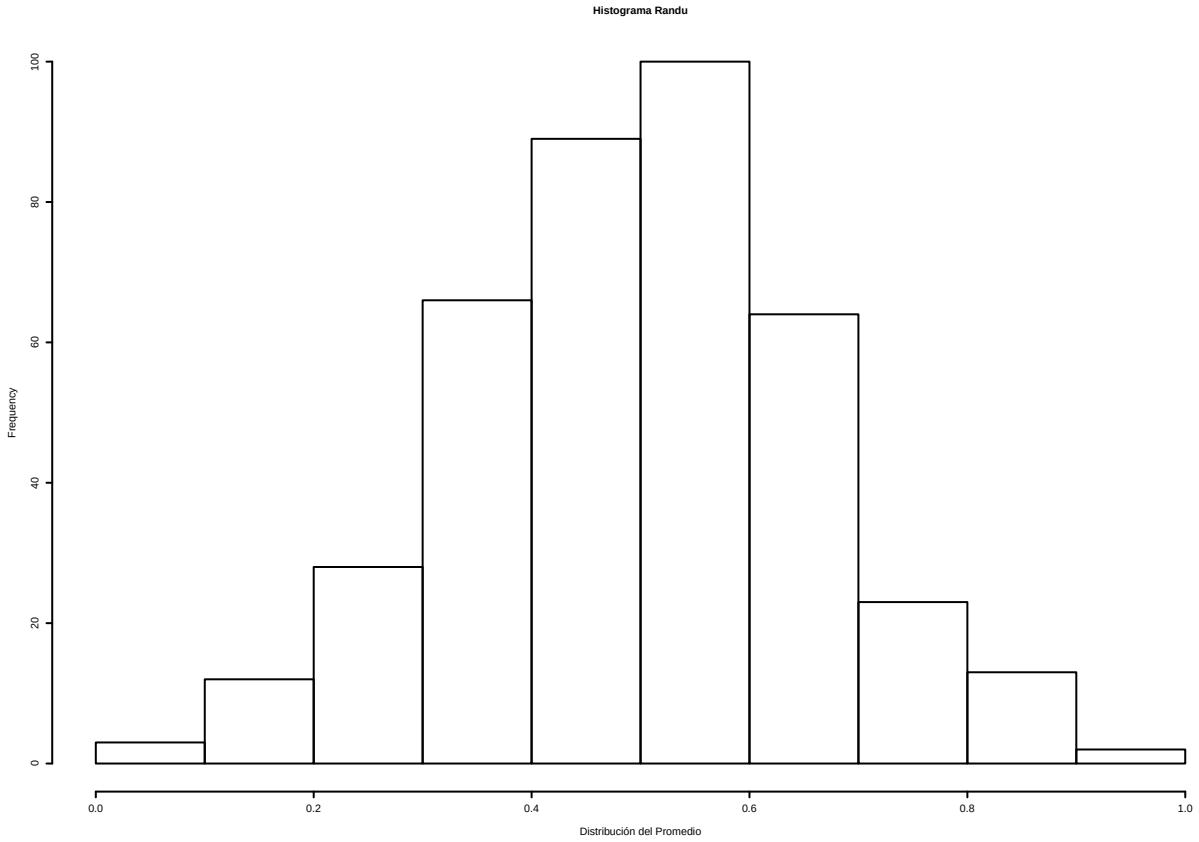

```
hist(randu$y, main="Randu: Frecuencias absolutas de Y", col ='coral2')
```




```
hist(randu$z, main="Randu: Frecuencias absolutas de Z", col='darkslategray4')
```




```r
hist(average_distribution, main = "Histograma Randu", xlab= "Distribución del Promedio")
```



d) Calcular sus propios breaks y volver a graficar el histograma con hist y con geom_histogram del paquete ggplot2.

```
summary(randu)
```

```
##        x                y                z          
##  Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:0.300   1st Qu.:0.228   1st Qu.:0.252  
##  Median :0.541   Median :0.483   Median :0.463  
##  Mean   :0.526   Mean   :0.486   Mean   :0.481  
##  3rd Qu.:0.779   3rd Qu.:0.740   3rd Qu.:0.711  
##  Max.   :1.000   Max.   :1.000   Max.   :0.998
```

para randu$x

```
dif <- 0.1
width <-0.38
minimo <- min(randu$x) - dif
maximo <- 1 + dif
maximo
```

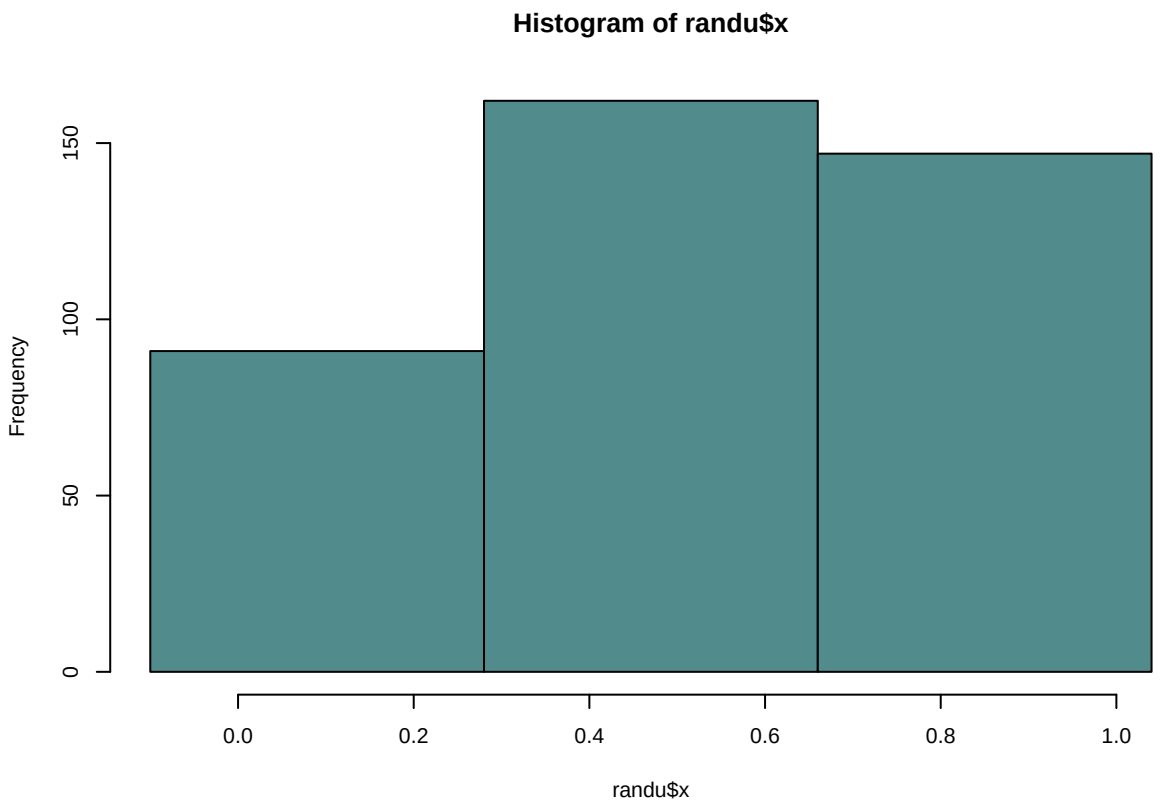
```
## [1] 1.1
```

```
cortes_x <- seq(minimo, maximo, by=width)
cortes_x
```

```
## [1] -0.10  0.28  0.66  1.04
```

```
randu$intervalo_x <- cut(randu$x, breaks = cortes_x)
```

```r
par(cex=0.7) #control size of labels
hist(randu$x, breaks = cortes_x, col='darkslategray4')
```

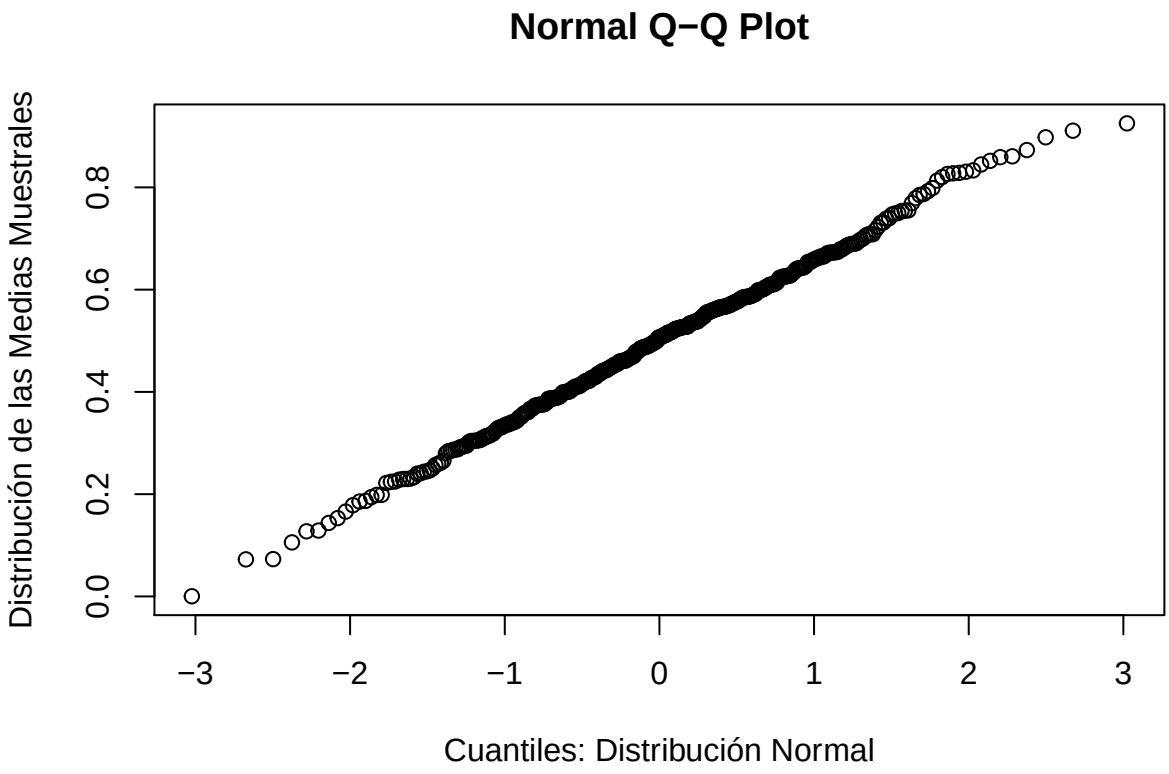



```r
table(randu$intervalo_x)
```

```
##
## (-0.1,0.28] (0.28,0.66] (0.66,1.04]
##          91         162         147
```

e) Utilizar la función qqnorm para comparar los cuantiles de una distribución normal con la distribución de las medias muestrales obtenidos en el inciso b). Sólo de forma visual ¿qué se puede concluir sobre la distribución de las medias muestrales calculadas en b)? (puedes añadir una línea al gráfico con qqline).

```r
qqnorm(average_distribution, xlab="Cuantiles: Distribución Normal", ylab = "Distribución de las Medias
```

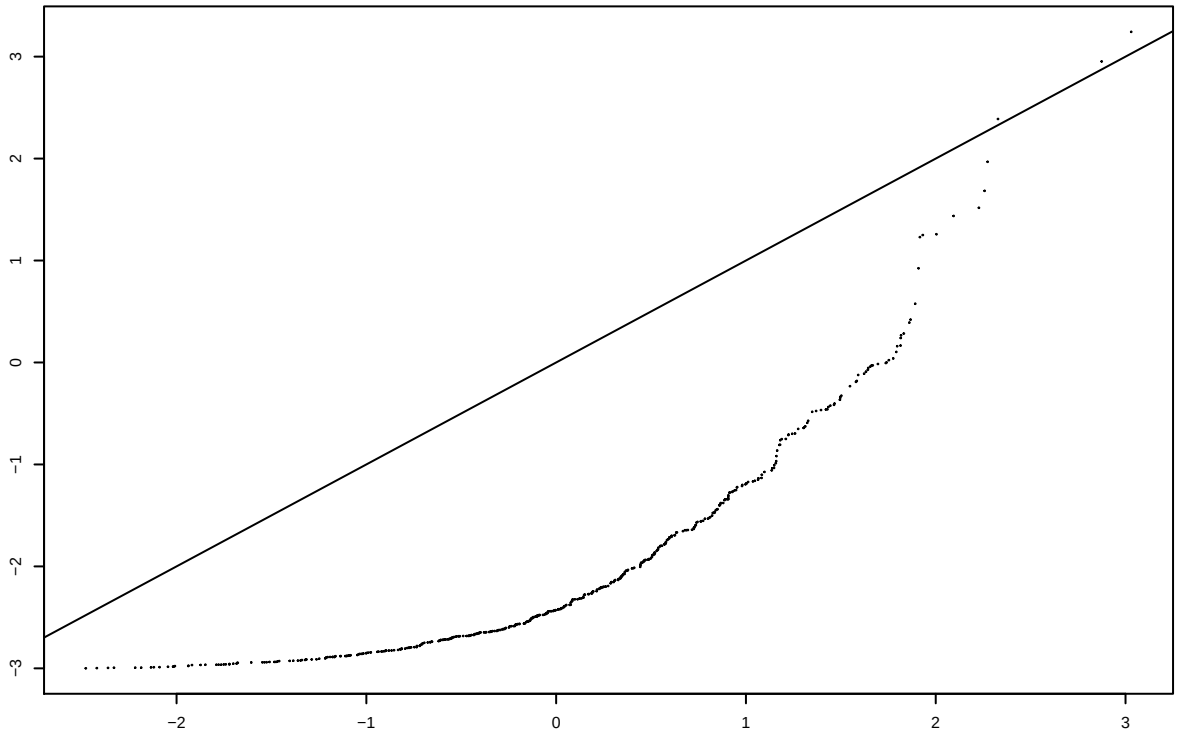


Visualmente: La presencia de la recta diagonal en la gráfica nos permite concluir que las medias muestrales tienen una distribución normal.

4) Realizar gráficas con el comando qqplot para comparar 2 distribuciones de datos

Ejemplo

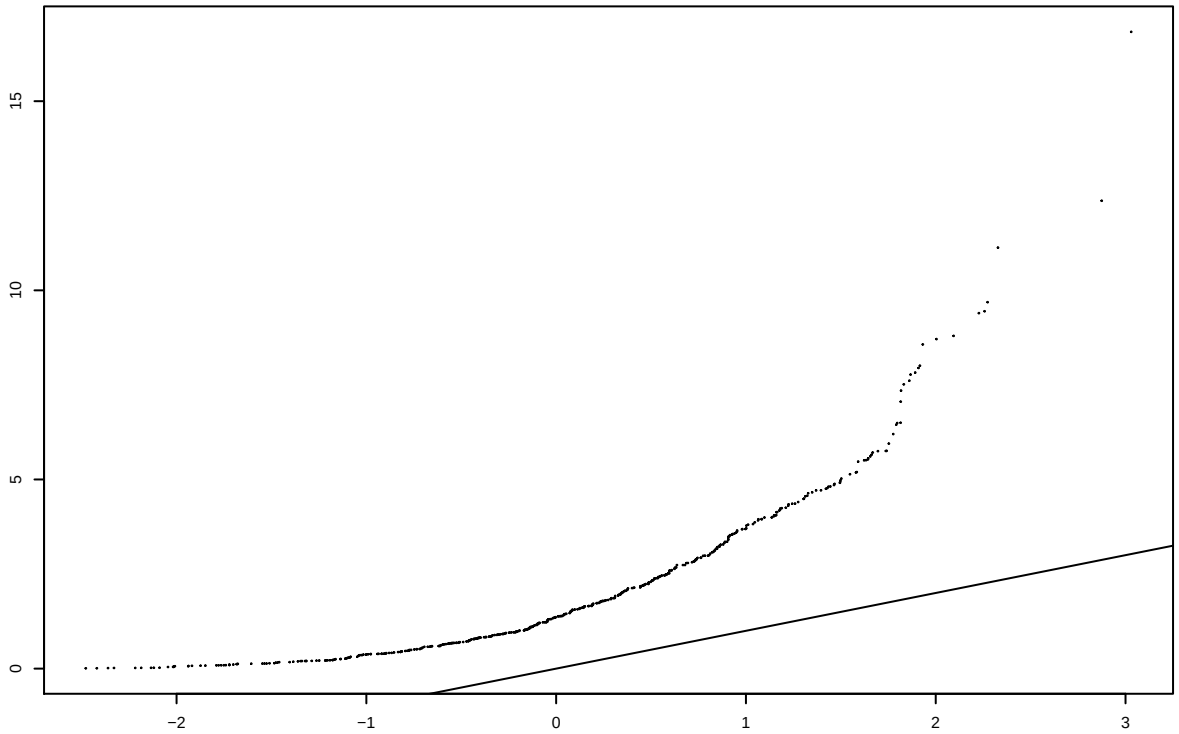
```
n<-500
set.seed(2000)
par(cex=0.5) #control size of labels
datos<- qqplot(rnorm(n), rexp(n)-3, cex=.1, xlab='', ylab='')
abline(0,1) #recta a 45 grados, sirve de apoyo para realizar comentarios
```



En el gráfico anterior se observa que una de las distribuciones está sesgada respecto a la otra. Este tipo de gráfica es similar al de una distribución sesgada a la derecha por lo que o bien, la cola izquierda de la otra distribución está más cercana a la mediana o tiene colas ligeras.

Distribución Normal vs Distribución Chi-Cuadrado

```
n<-500
set.seed(2000)
par(cex=0.5) #control size of labels
datos<- qqplot(rnorm(n), rchisq(n, df=2), cex=.1, xlab='', ylab='')
abline(0,1) #recta a 45 grados, sirve de apoyo para realizar comentarios
```



En el gráfico anterior se observa que una de las distribuciones está sesgada respecto a la otra. Este tipo de gráfica es similar al de una distribución sesgada a la izquierda por lo que o bien, la cola derecha de la otra distribución está más cercana a la mediana o tiene colas ligeras .