

R Notebook

```
library(pander)
library(MASS)
library(ggplot2)
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(tidyr)
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(readr)
library(stringr)
```

Ejercicios

- 1) Considérese el experimento de lanzar 20 veces una moneda y obtener la secuencia: $H, T, H, H, T, H, H, T, H, H, T, T, H, T, H, H, H, T$
 - a) Tabular los resultados del experimento anterior encontrando las proporciones de H y T en los 20 lanzamientos.

```
NumLanz<-(1:20)
ResLanz<-c("H","T","H","H","T","H","H","T","H","H","T","T","H","T","T","T","H","H","H","T")
tab<-cbind(NumLanz,ResLanz)
tab<-data.frame(NumLanz,ResLanz)

Heads<-sum(tab=="H")
Tails<-sum(tab=="T")
Total<-nrow(tab)

#Proporción de Heads:
PropH<-c(Heads/Total)
PropH
```

```
## [1] 0.55
```

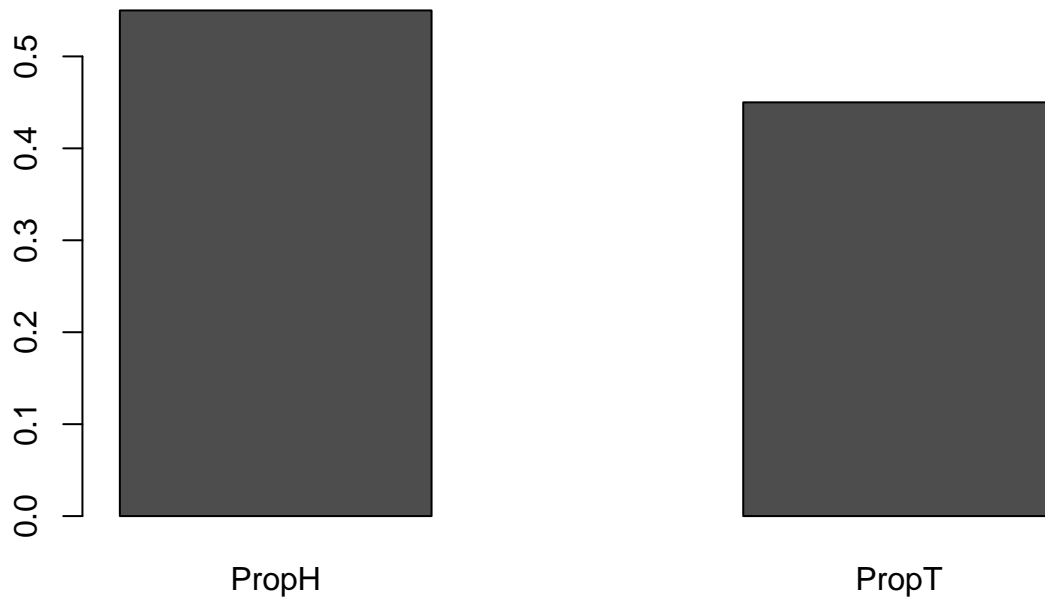
```
#Proporción de Tails:
PropT<-c(Tails/Total)
PropT
```

```
## [1] 0.45
```

- b) Graficar las proporciones con barplot y plot.

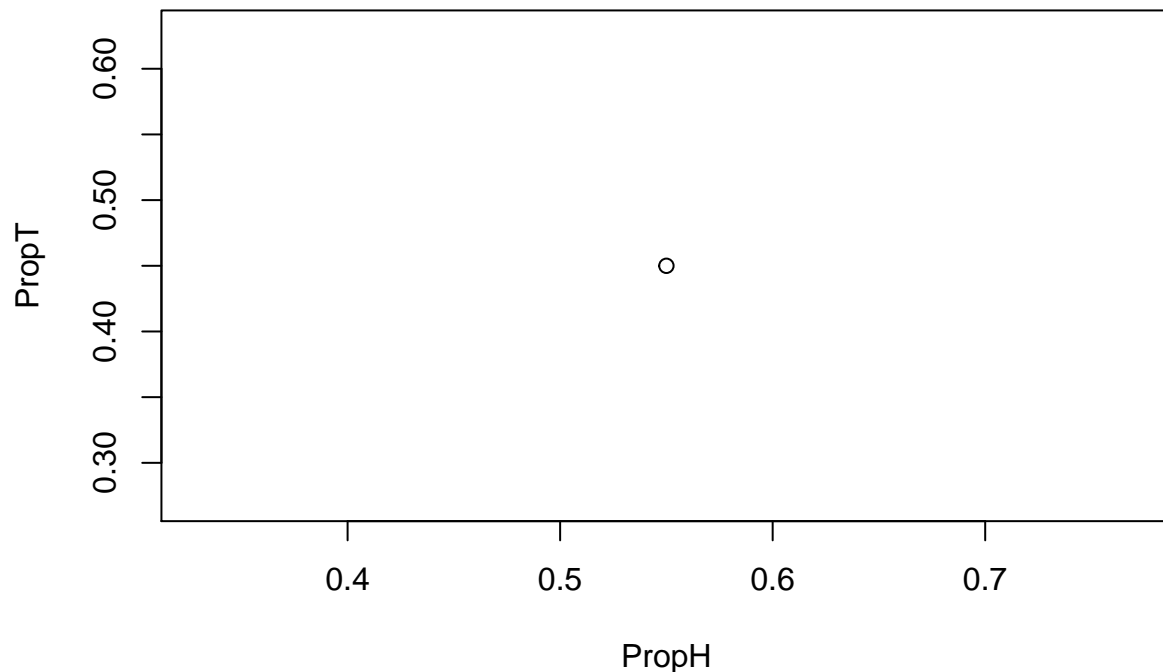
```
proporciones<-cbind(PropH,PropT)
barplot(proporciones, beside=TRUE,main = "Proporción Heads and Tails con barplot")
```

Proporción Heads and Tails con barplot



```
plot(proporciones,main = "Proporción Heads and Tails con plot")
```

Proporción Heads and Tails con plot



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

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.

Con el diagrama de tallo y hojas, podemos darnos una idea de la distribución de frecuencia y la ordenación de los datos. Por ejemplo, el siguiente es el diagrama de tallo y hoja para `log_body`, con este podemos ver que los datos están distribuidos entre -4 y 8 y que la concentración de datos está alrededor del 0 es bastante fuerte.

```
log_mammals <- log(mammals)
stem(log_mammals$body)
```

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

Ahora del siguiente diagrama de tallo y hoja para `log_brain`, vemos que los datos varían entre -2 y 8 y que la mayor concentración de datos también está alrededor del 0.

```
stem(log_mammals$brain)
```

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

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

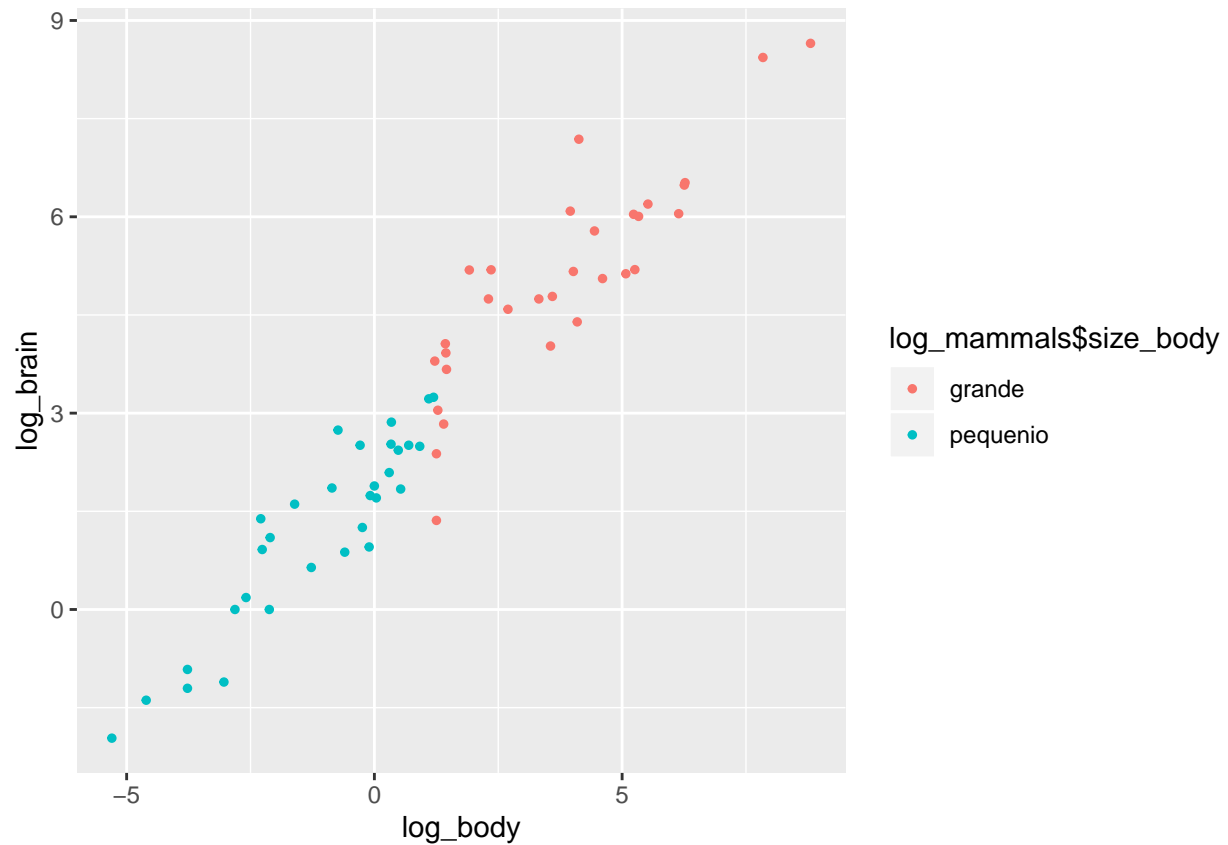
```
colnames(log_mammals) <- c('log_body', 'log_brain')
mediana_body <- median(log_mammals$log_body)
log_mammals$size_body <- ifelse(log_mammals$log_body >= mediana_body, "grande", "pequeno")
#añadimos también la de brain:
mediana_brain <- median(log_mammals$log_brain)
log_mammals$size_brain <- ifelse(log_mammals$log_brain >= mediana_brain, "grande", "pequeno")

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

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

df_mammals_gg<-rbind(df_mammals_body, df_mammals_brain)

ggplot(log_mammals)+geom_point(aes(x=log_body, y=log_brain,color=log_mammals$size_body),size=1)
```

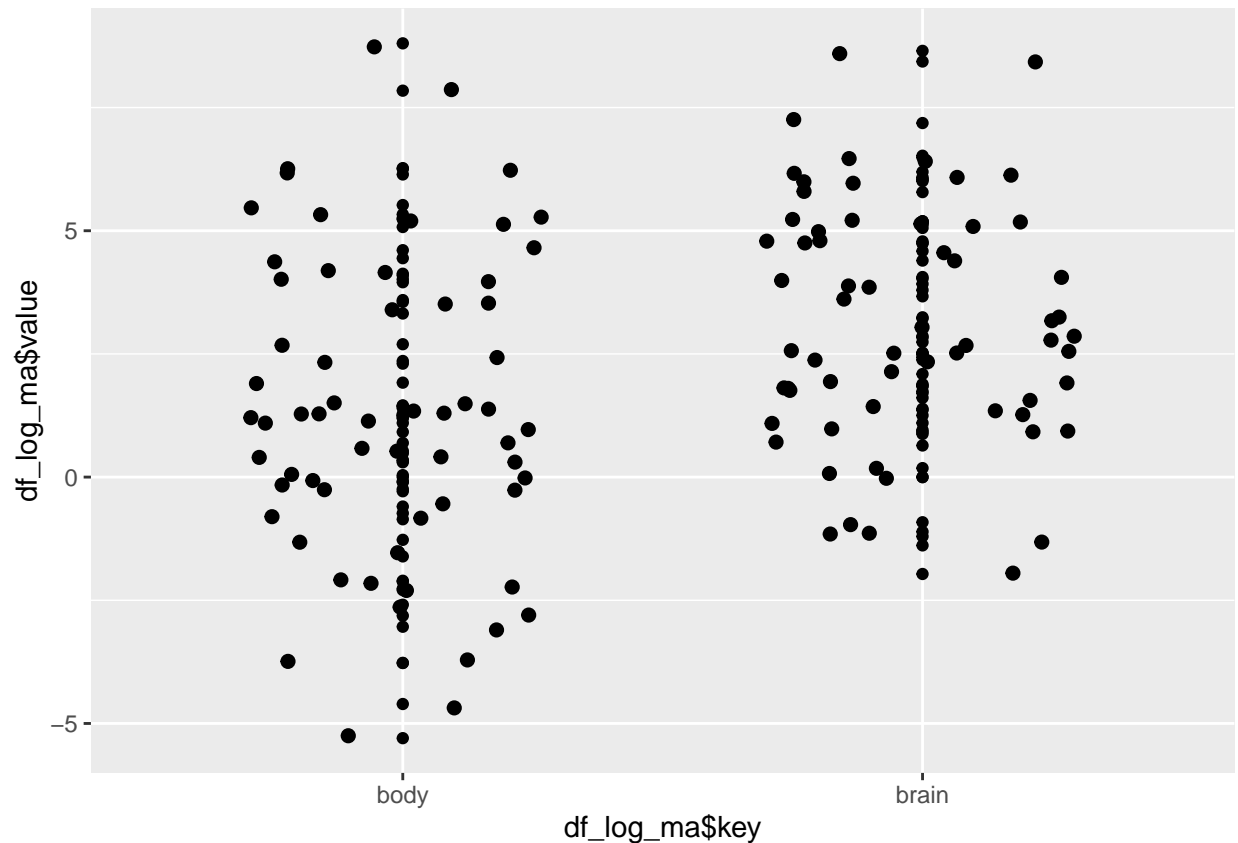


- c) Instalar el paquete dplyr para colocar los datos en un formato long y se pueda graficar con el paquete de ggplot2 con la función gather el dataframe log_mammals. Después de instalar tal paquete, realizar mismo enunciado que a) pero con geom_points y paquete ggplot2.

```
logmammals <- log(mammals)

df_log_ma<-gather(logmammals)

ggplot(df_log_ma)+geom_point(aes(x=df_log_ma$key,y=df_log_ma$value))+
geom_jitter(aes(x=df_log_ma$key,y=df_log_ma$value),position=position_jitter(width=0.3,height=0.08),size=
```



- 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`, `summarise` y el operador `%>%` (pipe). Ver como ayuda: [Data Wrangling with dplyr and tidyr](https://genomicsclass.github.io/book/pages/dplyr_tutorial.html) o [Data Transformation with dplyr](https://genomicsclass.github.io/book/pages/dplyr_tutorial.html) o bien como ejemplo de uso: https://genomicsclass.github.io/book/pages/dplyr_tutorial.html u otro tutorial en la red de dplyr ¿qué mamíferos en cada grupo están en los tres primeros lugares (pensando que se ordenan de forma decreciente en `log_brain`)?

Estadísticas:

```
log_mammals <- log(mammals)
colnames(log_mammals) <- c('log_body', 'log_brain')

mediana_body = median(log_mammals$log_body)
log_mammals$size_body = ifelse(log_mammals$log_body >= mediana_body, "grande", "pequeno")

mediana_brain = median(log_mammals$log_brain)
log_mammals$size_brain = ifelse(log_mammals$log_brain >= mediana_brain, "grande", "pequeno")

log_mammals %>%
  group_by(size_brain) %>%
  summarise(avg_log_mammals_brain = mean(log_brain),
            med_log_mammals_brain = median(log_brain),
            min_log_mammals_brain = min(log_brain),
            max_log_mammals_brain = max(log_brain))
```

```
## # A tibble: 2 x 5
```

```
## size_brain avg_log_mammals~ med_log_mammals~ min_log_mammals~
## <chr> <dbl> <dbl> <dbl>
## 1 grande 5.14 5.13 2.86
## 2 pequeno 1.14 1.39 -1.97
## # ... with 1 more variable: max_log_mammals_brain <dbl>
```

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

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

df_mammals_brain$size_brain= ifelse(log_mammals$log_brain >= mediana_brain, "grande", "pequeno")
```

Los mamíferos en los primeros 3 lugares son: Lesser short-tailed shrew, Little brown bat y Big brown bat

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

```
##          mammal log_brain size_brain
## 1 Lesser short-tailed shrew -1.9661129 pequeno
## 2 Little brown bat -1.3862944 pequeno
## 3 Big brown bat -1.2039728 pequeno
## 4 Musk shrew -1.1086626 pequeno
## 5 Mouse -0.9162907 pequeno
## 6 Star-nosed mole 0.0000000 pequeno
```

Los mamíferos en los primeros lugares son: African elephant, Asian elephant y Human

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

```
##          mammal log_brain size_brain
## 1 African elephant 8.650325 grande
## 2 Asian elephant 8.434464 grande
## 3 Human 7.185387 grande
## 4 Giraffe 6.522093 grande
## 5 Horse 6.484635 grande
## 6 Okapi 6.194405 grande
```

- 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. ¿Qué mamíferos tienen los cocientes más grandes?

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



```

        valor_body = mammals$body)

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

df_mammals_gg<-cbind(df_mammals_body, df_mammals_brain)

df_mammals_gg<-df_mammals_gg[,-1:-2]

df_mammals_gg$r=df_mammals_gg$valor_brain/df_mammals_gg$valor_body

df_mammals_gg %>%
  select(mammal,r,valor_brain,valor_body) %>%
  arrange(desc(r)) %>%
  head

```

```

##           mammal      r valor_brain valor_body
## 1   Ground squirrel 39.60396      4.00      0.101
## 2         Owl monkey 32.29167     15.50      0.480
## 3 Lesser short-tailed shrew 28.00000      0.14      0.005
## 4         Rhesus monkey 26.32353    179.00      6.800
## 5           Galago 25.00000      5.00      0.200
## 6   Little brown bat 25.00000      0.25      0.010

```

¿y cuáles los más pequeños?

```

df_mammals_gg %>%
  select(mammal,r,valor_brain,valor_body) %>%
  arrange(r) %>%
  head

```

```

##           mammal      r valor_brain valor_body
## 1 African elephant 0.8584310    5712.0    6654.0
## 2           Cow 0.9096774     423.0     465.0
## 3           Pig 0.9375000     180.0     192.0
## 4 Brazilian tapir 1.0562500     169.0     160.0
## 5   Water opossum 1.1142857       3.9       3.5
## 6           Horse 1.2571977     655.0     521.0

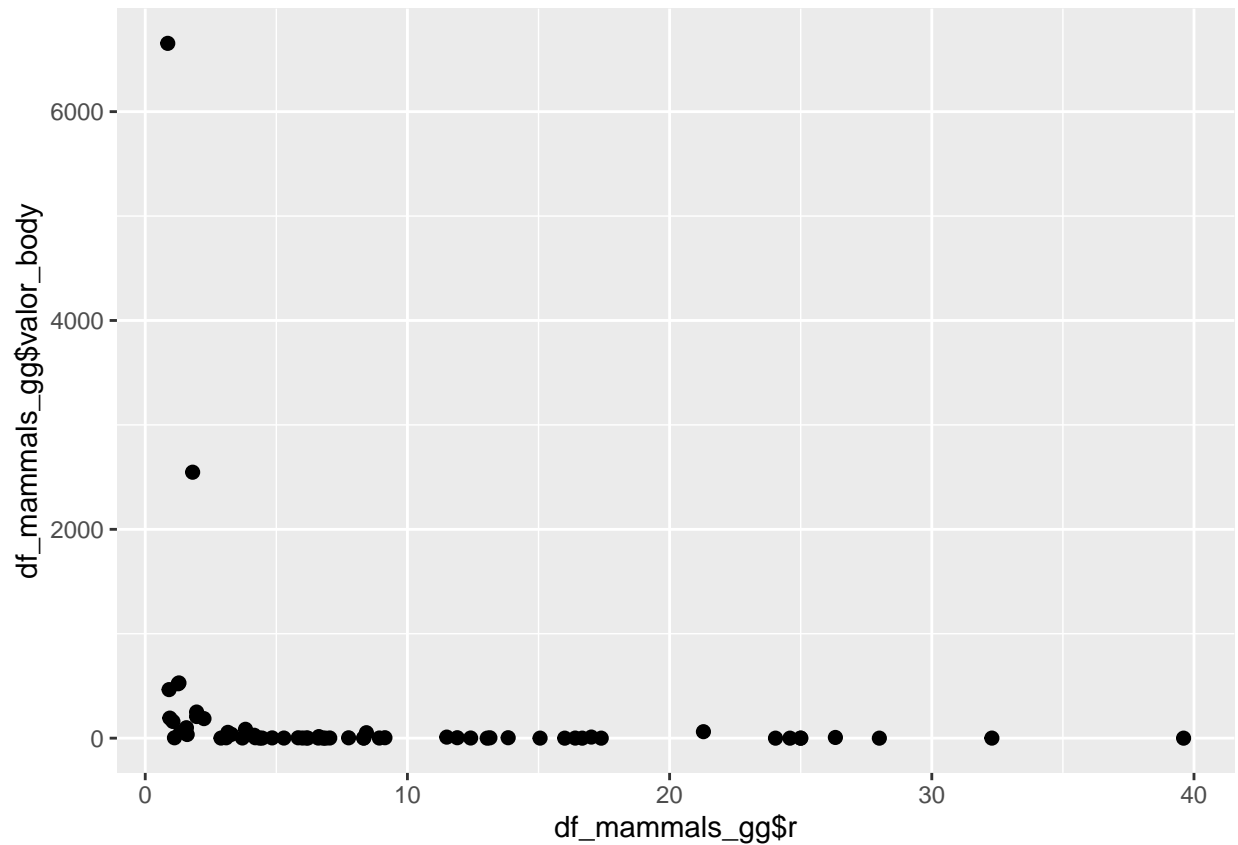
```

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

```

ggplot(df_mammals_gg)+geom_point(aes(x=df_mammals_gg$r, y=df_mammals_gg$valor_body),size=2)

```



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

- 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.

```
#media muestral de x, y, z
sapply(randu, mean)
```

```
##           x           y           z
## 0.5264293 0.4860531 0.4809547
```

```
#matriz de varianzas y covarianzas muestral
var(randu)
```

```
##           x           y           z
## x 0.081231885 -0.004057683 0.004637656
## y -0.004057683 0.086270206 -0.005148432
## z 0.004637656 -0.005148432 0.077860433
```

- 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, alternatively usar rowMeans.

```
distprom1<-apply(randu,MARGIN=1,FUN=mean)
distprom1
```

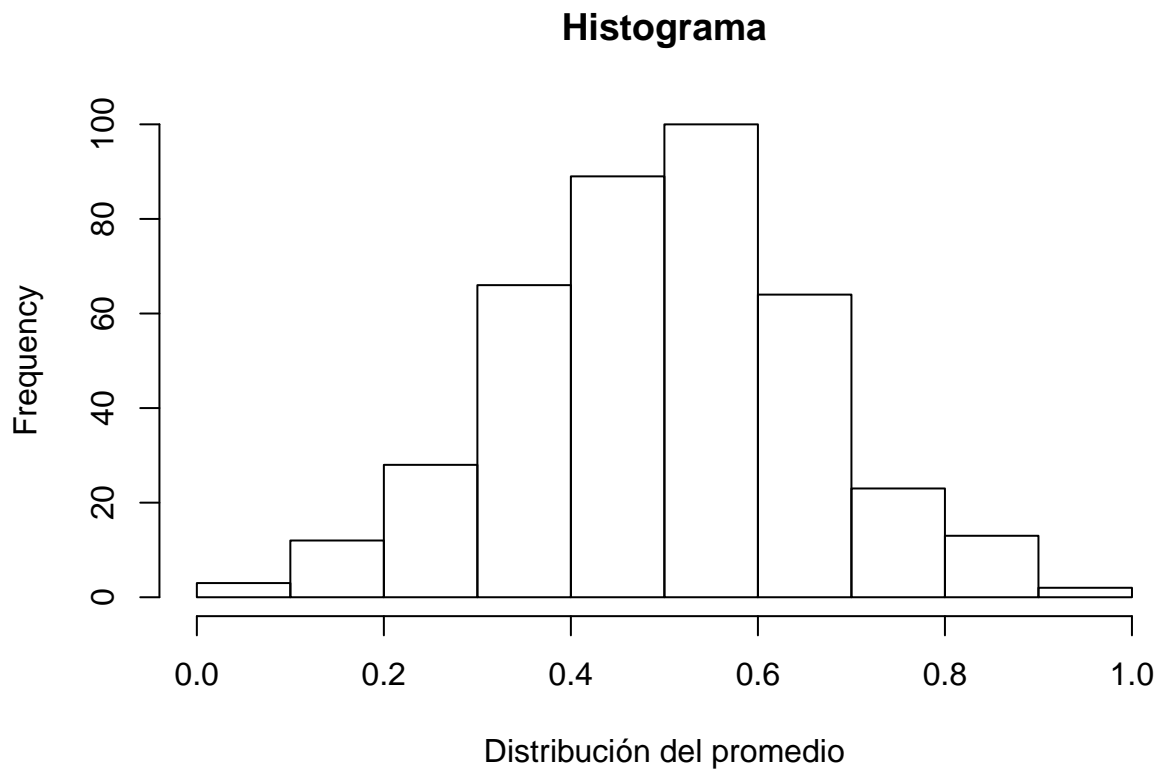
```
##          1          2          3          4          5          6
## 0.00034600 0.24472200 0.84479933 0.65382800 0.54644967 0.67111700
##          7          8          9         10         11         12
## 0.51628000 0.78509600 0.40436167 0.70798033 0.59661533 0.75505067
##          13         14         15         16         17         18
## 0.39903167 0.60995867 0.67334500 0.12734433 0.87306033 0.75382000
##          19         20         21         22         23         24
## 0.34278600 0.59890000 0.55372133 0.65879933 0.48721667 0.68822500
##          25         26         27         28         29         30
## 0.81321633 0.56162867 0.67445133 0.16591133 0.61382500 0.42126133
##          31         32         33         34         35         36
## 0.30481433 0.43565900 0.42961733 0.22827400 0.46383967 0.40748533
##          37         38         39         40         41         42
## 0.43639400 0.49396767 0.10570567 0.40478767 0.26152467 0.46081267
##          43         44         45         46         47         48
## 0.85905767 0.31359700 0.58534667 0.57591967 0.74690067 0.51568000
##          49         50         51         52         53         54
## 0.55246300 0.76919800 0.77888267 0.48050300 0.28459667 0.41120133
##          55         56         57         58         59         60
## 0.37430933 0.55770900 0.45334733 0.49232367 0.24088800 0.51178667
##          61         62         63         64         65         66
## 0.53784033 0.56502367 0.66753267 0.34059300 0.39227267 0.65739533
##          67         68         69         70         71         72
## 0.52290033 0.58606133 0.41502000 0.36643200 0.58465467 0.67876833
##          73         74         75         76         77         78
## 0.61097433 0.67264200 0.49112600 0.52468233 0.50544467 0.35101600
##          79         80         81         82         83         84
## 0.60703433 0.54243600 0.53461767 0.52002167 0.42827433 0.35945400
##          85         86         87         88         89         90
## 0.53758433 0.53694767 0.48409633 0.37526500 0.64159167 0.44793367
##          91         92         93         94         95         96
## 0.69501733 0.40033833 0.49886067 0.53436067 0.47812467 0.57877033
##          97         98         99        100        101        102
## 0.83329867 0.83058867 0.22478067 0.46983000 0.58877233 0.48805533
##          103        104        105        106        107        108
## 0.44238533 0.49668667 0.30394767 0.86072100 0.85187000 0.30399733
##          109        110        111        112        113        114
## 0.67201967 0.18687533 0.39925300 0.56601033 0.57701167 0.40896067
##          115        116        117        118        119        120
## 0.47292667 0.45352567 0.56496300 0.67308300 0.61833533 0.30137467
##          121        122        123        124        125        126
## 0.70835300 0.71654500 0.70838833 0.68475633 0.48874833 0.35173267
##          127        128        129        130        131        132
## 0.52793133 0.44609100 0.32268167 0.58802367 0.73061367 0.54739800
##          133        134        135        136        137        138
## 0.35576367 0.60004733 0.45982233 0.62319133 0.48334800 0.41730233
##          139        140        141        142        143        144
## 0.59200600 0.44223267 0.55610167 0.54345867 0.56073267 0.15313533
##          145        146        147        148        149        150
## 0.39546467 0.38035033 0.38647300 0.52215833 0.57734467 0.66498433
```

##	151	152	153	154	155	156
##	0.53711833	0.37151967	0.60903533	0.46124267	0.91088500	0.42656400
##	157	158	159	160	161	162
##	0.45769800	0.58126733	0.56662700	0.44678600	0.38874333	0.64486867
##	163	164	165	166	167	168
##	0.49319867	0.51570567	0.07243933	0.31861933	0.37622333	0.69793033
##	169	170	171	172	173	174
##	0.25041300	0.60963833	0.65391700	0.60507767	0.58705133	0.42239400
##	175	176	177	178	179	180
##	0.82031300	0.41220900	0.33079800	0.58686000	0.60088000	0.53312567
##	181	182	183	184	185	186
##	0.41034233	0.56636333	0.62553533	0.49618167	0.59070867	0.38678033
##	187	188	189	190	191	192
##	0.22405633	0.56228533	0.64757200	0.66209267	0.63555267	0.62675433
##	193	194	195	196	197	198
##	0.51813633	0.59129333	0.25700200	0.28002600	0.38691100	0.46831600
##	199	200	201	202	203	204
##	0.55584600	0.56888567	0.66471233	0.24664867	0.60498567	0.33322233
##	205	206	207	208	209	210
##	0.48589833	0.70030800	0.19850267	0.32662967	0.68216433	0.72282567
##	211	212	213	214	215	216
##	0.50644633	0.32920067	0.52456667	0.63926333	0.56034467	0.46084267
##	217	218	219	220	221	222
##	0.52544333	0.73957667	0.52778433	0.57011000	0.92523233	0.57500267
##	223	224	225	226	227	228
##	0.40105000	0.24022733	0.48918167	0.55913433	0.39148933	0.28729900
##	229	230	231	232	233	234
##	0.57229133	0.41534733	0.65461467	0.62570600	0.41081700	0.67891900
##	235	236	237	238	239	240
##	0.30426133	0.37569067	0.26533767	0.12882867	0.34080933	0.42146233
##	241	242	243	244	245	246
##	0.37371767	0.69091700	0.45180500	0.29482367	0.67229633	0.68739633
##	247	248	249	250	251	252
##	0.30997067	0.25931767	0.24364300	0.64105100	0.66028633	0.46989467
##	253	254	255	256	257	258
##	0.44920800	0.50912500	0.48779267	0.74977000	0.70519867	0.22189300
##	259	260	261	262	263	264
##	0.55779367	0.52613233	0.52669067	0.40092533	0.28678733	0.46637300
##	265	266	267	268	269	270
##	0.46496333	0.79870233	0.56912667	0.68983933	0.22973167	0.33458433
##	271	272	273	274	275	276
##	0.51774067	0.43010633	0.28794567	0.60213567	0.41058833	0.68943500
##	277	278	279	280	281	282
##	0.56414400	0.38947200	0.46783800	0.42708800	0.53404500	0.42095600
##	283	284	285	286	287	288
##	0.41811100	0.48982967	0.33636033	0.52858700	0.47878967	0.82628100
##	289	290	291	292	293	294
##	0.52784200	0.46140600	0.37538167	0.52679833	0.43988933	0.54126067
##	295	296	297	298	299	300
##	0.35828900	0.23053433	0.42838833	0.37325100	0.50863167	0.44135800
##	301	302	303	304	305	306
##	0.64177500	0.48586833	0.62480467	0.29429233	0.59900067	0.44248567
##	307	308	309	310	311	312
##	0.31476367	0.31414133	0.53572167	0.63253600	0.58570467	0.37463633

```
##      313      314      315      316      317      318
## 0.14371967 0.28404467 0.55614200 0.39977167 0.66188533 0.57251633
##      319      320      321      322      323      324
## 0.58204767 0.34899233 0.61043133 0.36784300 0.42205233 0.36123800
##      325      326      327      328      329      330
## 0.45999867 0.82756367 0.40004333 0.59950633 0.52732900 0.37815067
##      331      332      333      334      335      336
## 0.52349200 0.33760400 0.38916333 0.62597467 0.51584233 0.49741933
##      337      338      339      340      341      342
## 0.43937067 0.07296833 0.38796267 0.17838233 0.54764067 0.51196500
##      343      344      345      346      347      348
## 0.50699433 0.38744367 0.44473600 0.31006900 0.45426133 0.51124367
##      349      350      351      352      353      354
## 0.79264433 0.45504100 0.63057467 0.73880167 0.89795367 0.57303900
##      355      356      357      358      359      360
## 0.56727200 0.64305733 0.36013067 0.45937933 0.56797567 0.46293633
##      361      362      363      364      365      366
## 0.61302300 0.33065400 0.30609100 0.62760033 0.31792533 0.33986567
##      367      368      369      370      371      372
## 0.33636167 0.73121933 0.23320933 0.22958700 0.47881900 0.49993233
##      373      374      375      376      377      378
## 0.18551367 0.66480333 0.36679933 0.29265900 0.43416633 0.78711500
##      379      380      381      382      383      384
## 0.45068500 0.34410333 0.38716633 0.19862167 0.74904600 0.29160800
##      385      386      387      388      389      390
## 0.62706300 0.58383067 0.30550600 0.19419733 0.53839733 0.51125200
##      391      392      393      394      395      396
## 0.64300900 0.58547367 0.82837333 0.62301433 0.50714433 0.43537067
##      397      398      399      400
## 0.56575600 0.75416800 0.52353667 0.56242400
```

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

```
hist(distprom1,main="Histograma",xlab="Distribución del promedio")
```



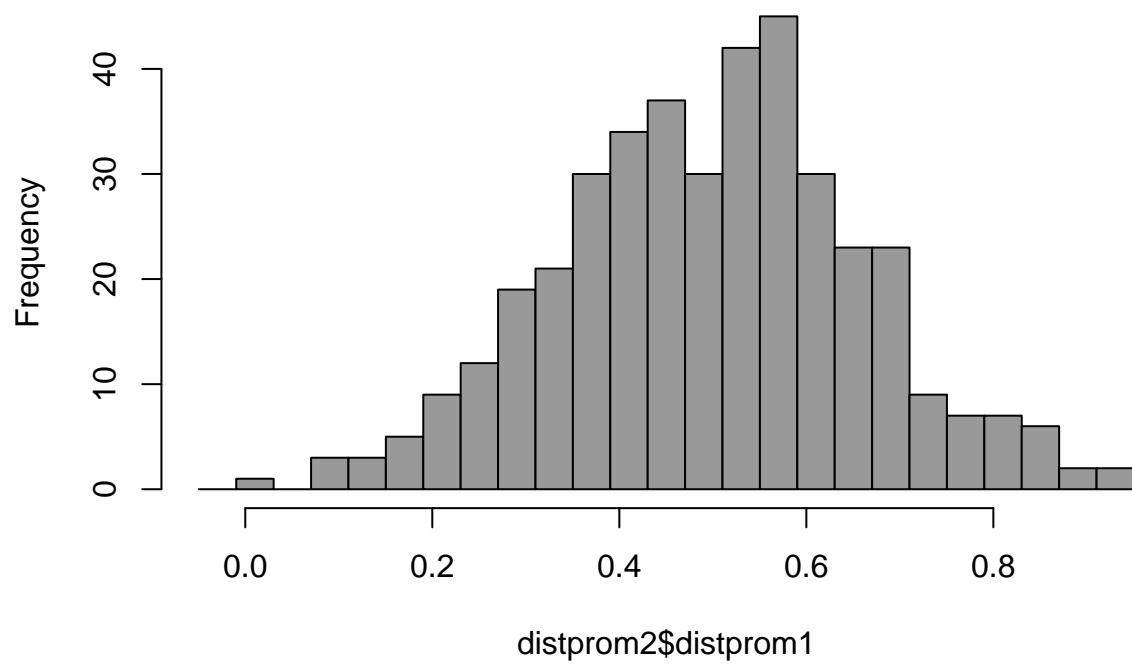
d) Calcular sus propios breaks y volver a graficar el histograma con `hist` y con `geom_histogram` del paquete `ggplot2`.

```
dif <- 0.05
width <- .04
minimo <- min(distprom1) - dif
maximo <- max(distprom1) + dif
cortes <- seq(minimo, maximo, by = width)

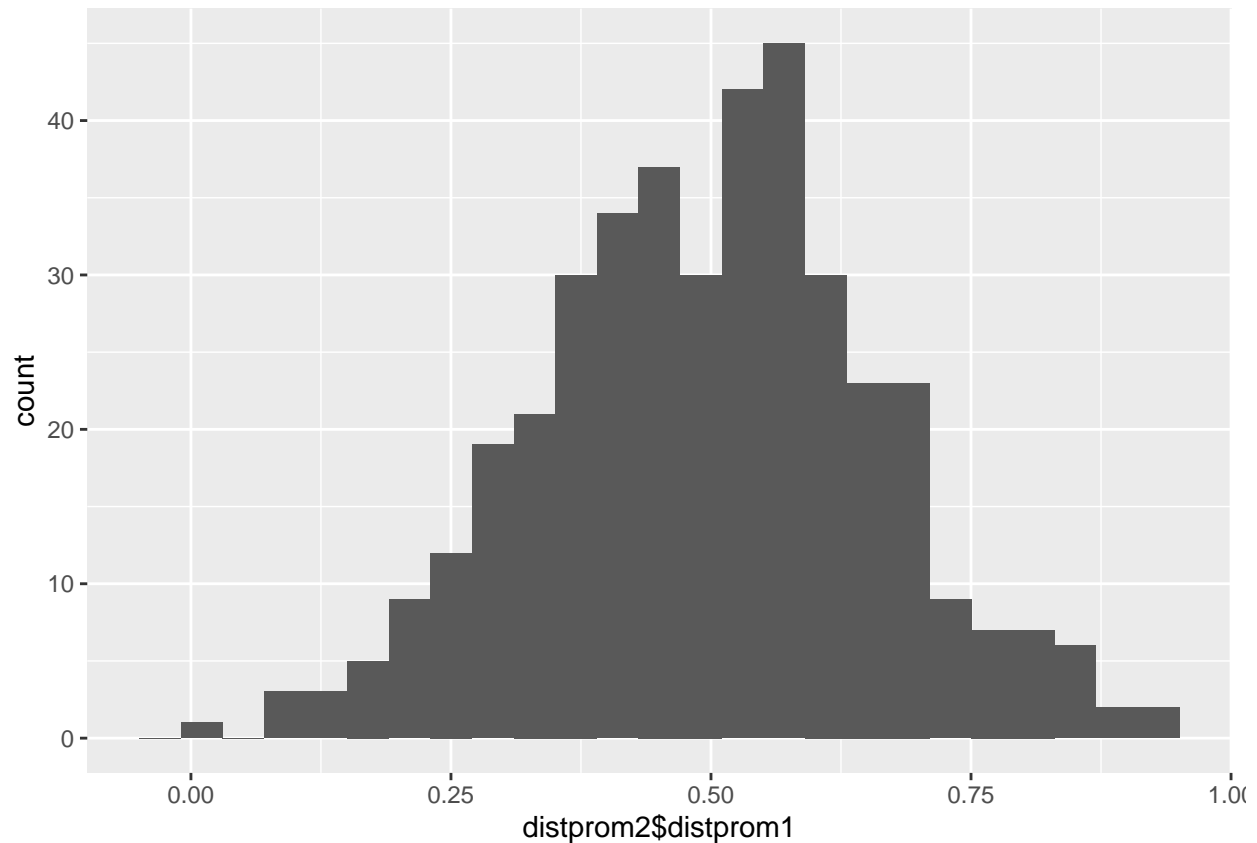
distprom2<-data.frame(distprom1)
distprom2$intervalo <- cut(distprom2$distprom1, breaks = cortes)

hist(distprom2$distprom1, breaks=cortes,
      main="Histograma con breaks propios", col='gray60') #conteos
```

Histograma con breaks propios



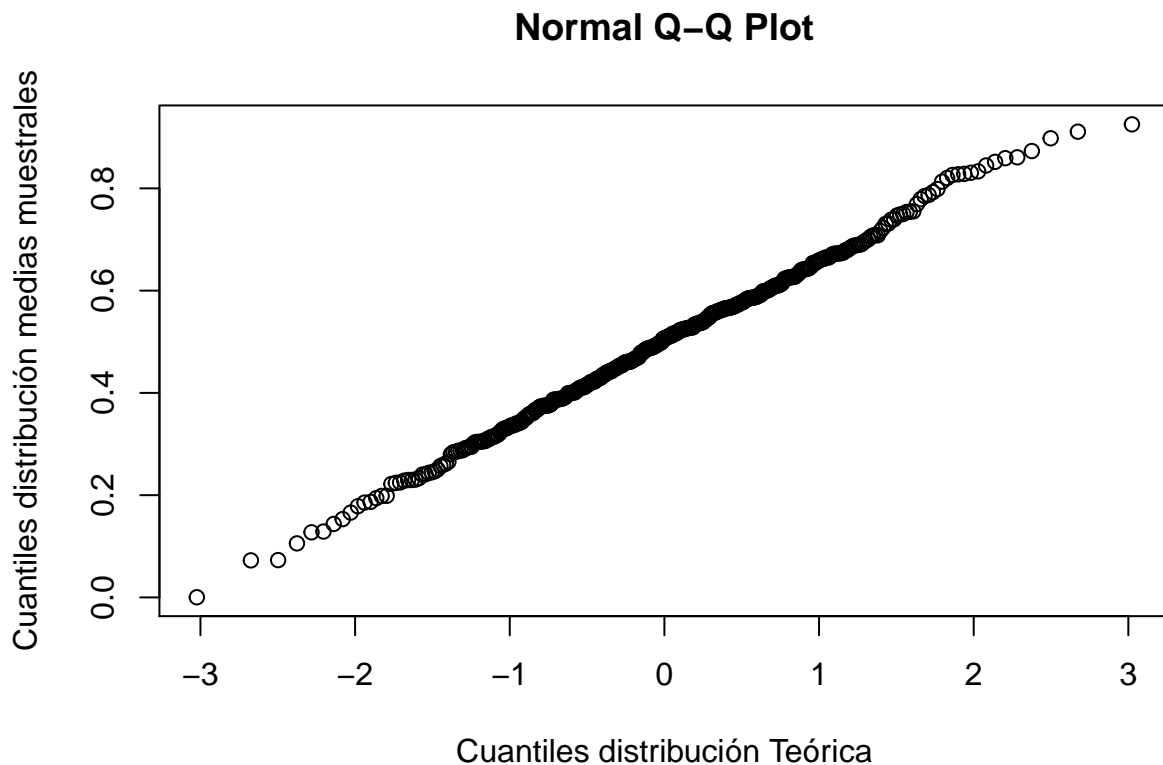
```
ggplot(data=distprom2) + geom_histogram(aes(x=distprom2$distprom1),  
                                         breaks=c(cortes))
```



- 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`).

Sí, podríamos concluir que las medias se distribuyen de forma normal ya que se aprecia de forma visual una recta diagonal

```
qqnorm(distprom1,xlab = "Cuantiles distribución Teórica",ylab="Cuantiles distribución medias muestrales")
```

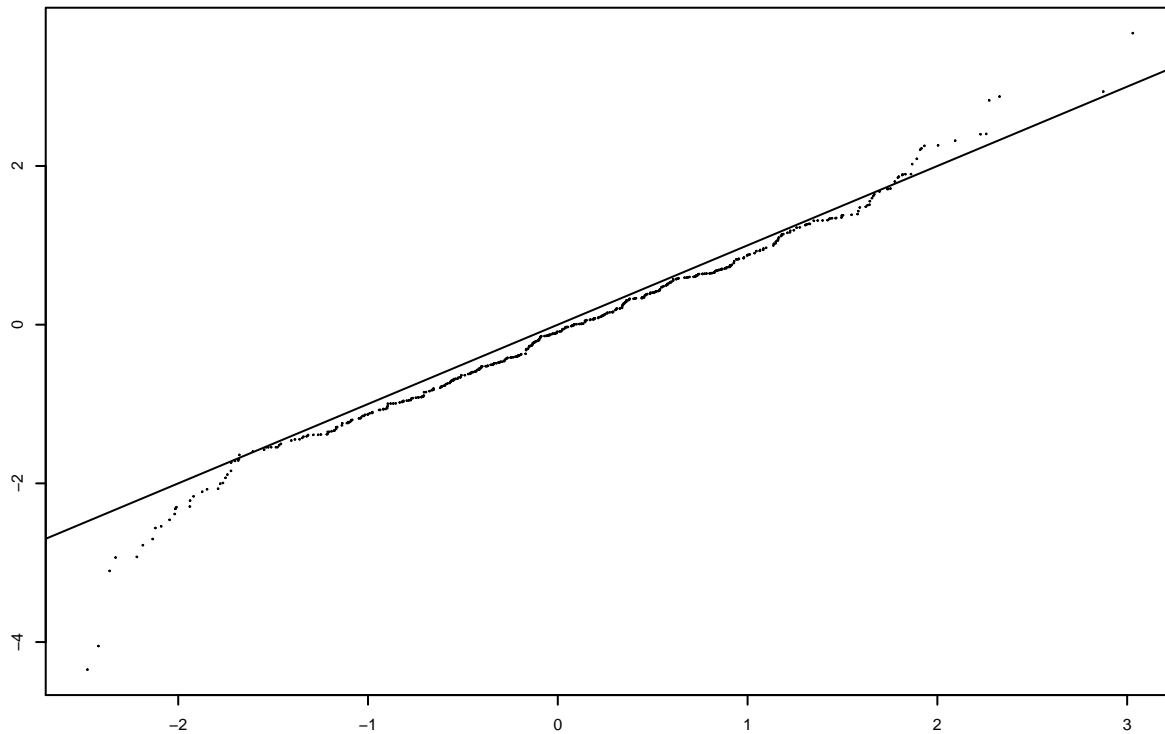



- 4) Realizar gráficas con el comando qqplot para comparar 2 distribuciones de datos Y Realizar conclusiones de sus gráficas qqplot.

Gráfica Normal vs T Student

En el siguiente gráfico se observa que ambas distribuciones tienen un comportamiento muy parecido ya que los valores observados están muy cercanos a la línea recta. Sólo en las colas de la distribución, se observa que las distribuciones tienen un sesgo.

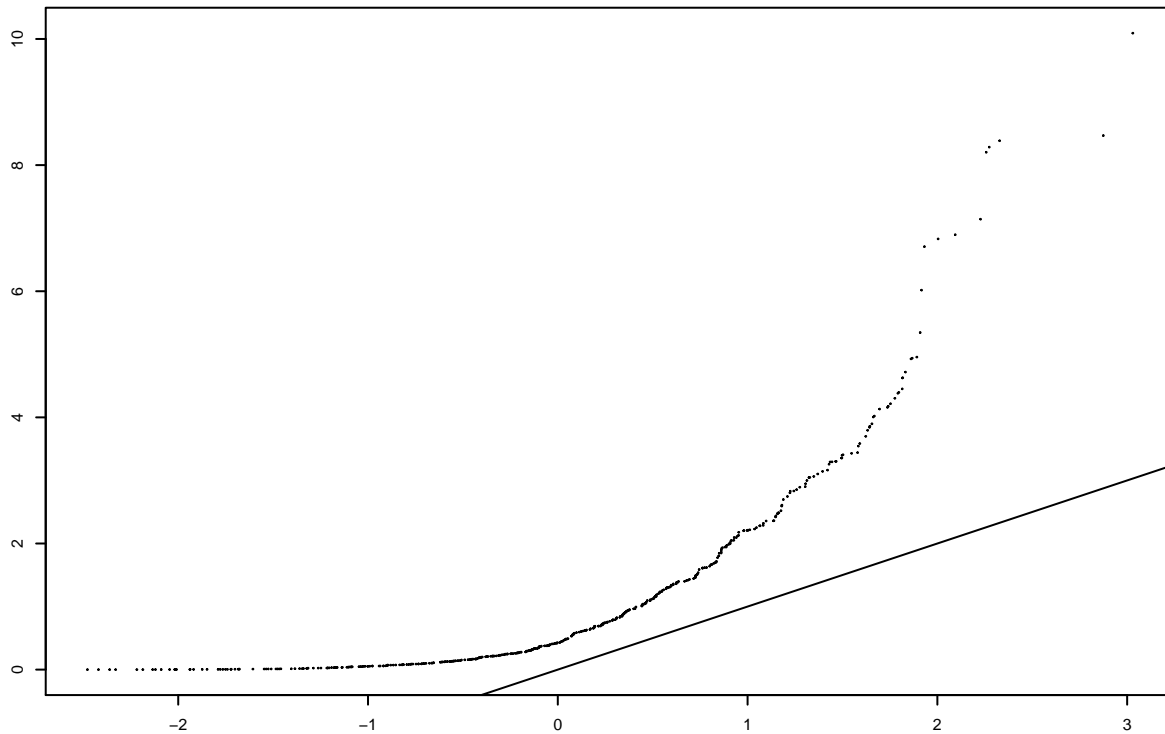
```
n<-500
set.seed(2000)
par(cex=0.5) #control size of labels
datos<- qqplot(rnorm(n), rt(n,df=10), cex=.1, xlab='', ylab='')
abline(0,1) #recta a 45 grados, sirve de apoyo para realizar comentarios
```



Gráfica Normal vs Chi Cuadrada

En el siguiente gráfico se observa que una de las distribuciones está totalmente 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

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



Ejercicios 1) Supóngase que el tiempo (en minutos) que una persona espera el Metrobús los días de semana sigue una distribución uniforme dada por, $f(x)=1/12$ si $0 \leq x \leq 12$, 0 en otro caso

a) ¿Cuál es la probabilidad que una persona espere menos de 6 minutos?.

```
a=0
b=12
punif(6,min = a,max=b)
```

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

b) ¿Su tiempo medio de espera? ¿su desviación estándar?.

```
a=0
b=12
#su tiempo medio de espera en minutos es:
mu=(a+b)/2
mu
```

```
## [1] 6
```

```
#su desviación estándar en minutos es:
sd=sqrt(((b-a)^2)/12)
sd
```

```
## [1] 3.464102
```

- 2) Considérese lanzamientos de una moneda no cargada. Sea X la variable aleatoria que cuenta el número de águilas en 10 lanzamientos. Calcular en R la probabilidad $p(X \geq 3)$ con la función `pbinom` y `dbinom`, el número medio de águilas y su desviación estándar.

```
n=10
p=.5
#Calcular la probabilidad de que haya 3 o más águilas es lo mismo que 1-probabilidad(hayan caído hasta 2)
#Esto se puede hacer con dbinom:
1-(dbinom(0,size =n,prob = p)+dbinom(1,size =n,prob = p)+dbinom(2,size =n,prob = p))
```

```
## [1] 0.9453125
```

```
#O con pbinom:
1-pbinom(2,size =n,prob = p)
```

```
## [1] 0.9453125
```

```
#El número medio de águilas es:
mu=n*p
mu
```

```
## [1] 5
```

```
#Su desviación estándar es:
sd=sqrt(n*p*(1-p))
sd
```

```
## [1] 1.581139
```

- 3) Considérese una población de votantes en la ciudad de México. Se reporta que la proporción de votantes que favorecen al candidato del partido Naranja es igual a 0.40. Dada una muestra aleatoria de 200 votantes, ¿cuál es la probabilidad de que más de la mitad de ellos tengan intención de voto por el candidato naranja? Usar:

a) `pbinom`

```
n=200
pfav=0.4
pbinom(n/2,size =n,prob =pfav,lower.tail = F)
```

```
## [1] 0.001684787
```

- b) El teorema del límite central y `pnorm` considerando lo que se conoce como corrección por continuidad.

Si X es variable binomial $B(n,p)$ ent. se puede aproximar por $N(np, \sqrt{npq})$. Además por la corrección por continuidad Binomial $P(X > k) \rightarrow P(X > k + 0.5)$ Normal

Por ello, la aproximación queda:

```
n=200
pfav=0.4
mu=n*pfav
sig=sqrt(n*pfav*(1-pfav))
pnorm((n/2)+0.5,mu,sig,lower.tail = F)
```

```
## [1] 0.001543596
```

- 4) Supongamos que el ingreso anual de un inmigrante mexicano en los EUA se distribuye de forma normal con una media de 30,000 y una desviación estándar de 10,000 dólares. ¿Cuál es la probabilidad de que un inmigrante mexicano gane menos de 20,000 dólares anuales? ¿Cuál es la probabilidad de que su ingreso sea mayor a los 50,000 dólares anuales? (usar R).

```
mu=30000
sd=10000
ia1=20000
ia2=50000
#Probabilidad de que un inmigrante gane menos de 20,000 dólares anuales:
pnorm(ia1,mu,sd)
```

```
## [1] 0.1586553
```

```
#Probabilidad de que un inmigrante gane más de 50,000 dólares anuales:
pnorm(ia2,mu,sd,lower.tail = F)
```

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

Ejercicios Estadística: 2_distribuciones_comunes

- 2) Si X se distribuye $N(0,1)$ calcular: $P(Z>0)$, $P(Z<-1)$, $P(-1<Z<0)$

Para el cálculo de $P(Z>0)$

```
ext_izq<-0
pnorm(ext_izq,lower.tail=F)
```

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

Para el cálculo de $P(Z<-1)$:

Por simetría de la normal se tiene: $P(Z<-1)=P(Z>1)$

```
ext_izq<-1
pnorm(ext_izq,lower.tail=F)
```

```
## [1] 0.1586553
```

De igual forma, por simetría: $P(-1<Z<0)=P(0<Z<1)$

Ejercicio: calcular este caso

```
ext_izq <- 0
ext_der <- 1
1-pnorm(ext_izq)-pnorm(ext_der, lower.tail = F)
```

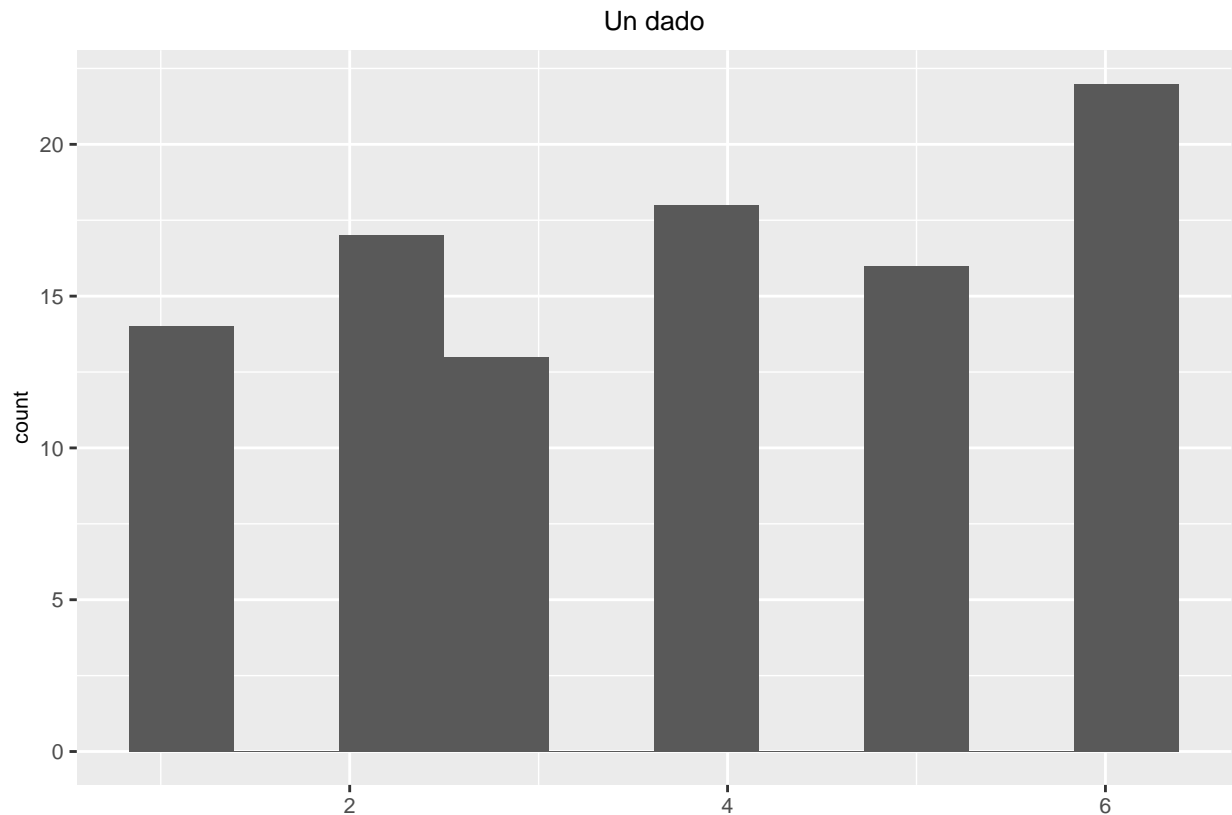
```
## [1] 0.3413447
```

Lanzamos **un dado** 100 veces y hacemos un histograma con los resultados. Después tomamos **dos dados** y los lanzamos 50 veces. **Sumamos** los números de las dos caras que aparecen en cada lanzamiento y realizamos un histograma con ellos. Tomamos ahora **tres dados** y lanzamos los tres juntos 50 ocasiones. **Sumamos** los números de las tres caras que aparecen en cada lanzamiento y realizamos un histograma con los resultados. Las siguientes figuras muestran estos histogramas. Visualmente observamos como la **distribución de la suma de las caras tiende a ser normal**.

Caso de un dado

```
set.seed(2000)
n_sample<-100
prob<-rep(1/6,6)
dado <-sample(1:6,size=n_sample, replace=T,prob=prob)
gf<-ggplot(data.frame(valores=dado),aes(x=dado))
```

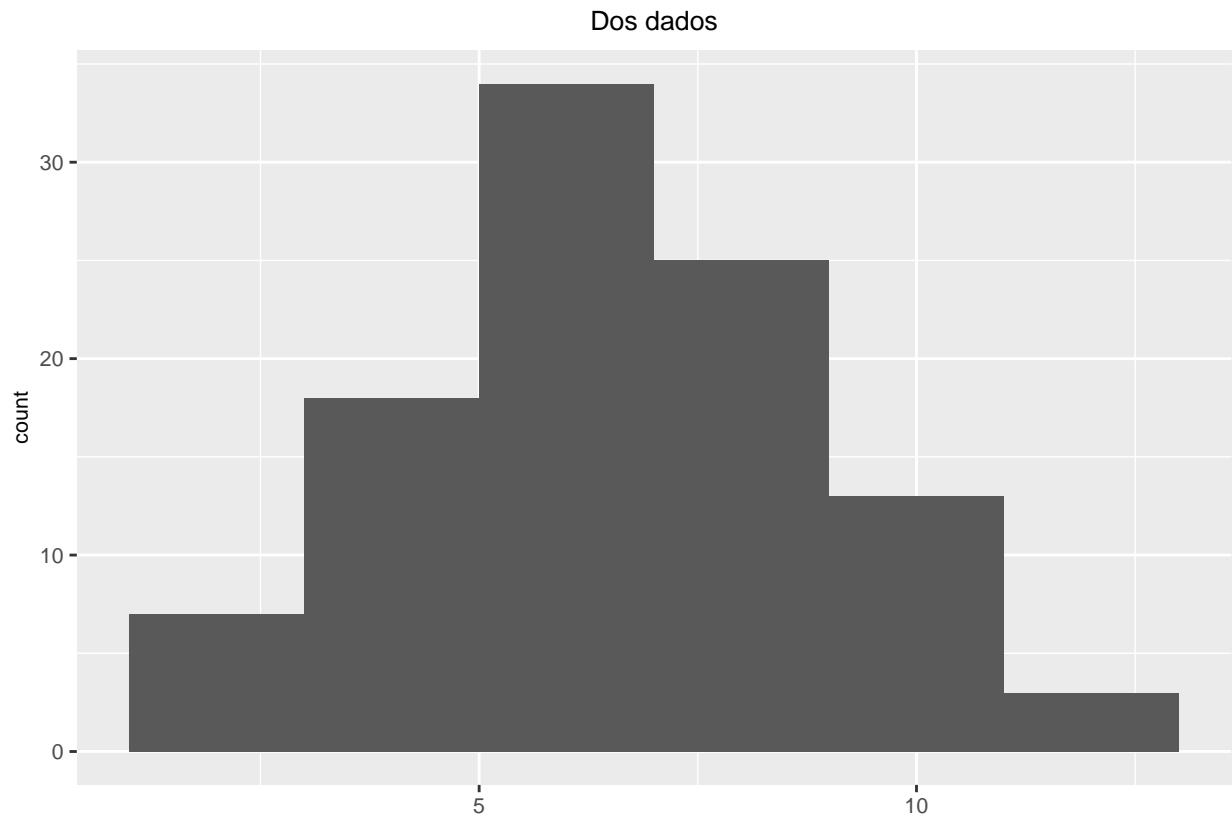
```
gf + geom_histogram(bins = 10)+
ggtitle('Un dado') +
theme(plot.title = element_text(size=10, hjust = 0.5),
      axis.text=element_text(size=8),
      axis.title=element_text(size=8)) +
xlab('')
```



Caso de dos dados

```
set.seed(2000)
n_sample<-100
prob<-c(1,2,3,4,5,6,5,4,3,2,1)/36
suma<-2:12
dos_dados<-sample(suma,n_sample,prob=prob,replace=T)
gf<-ggplot(data.frame(valores=dos_dados),aes(x=dos_dados))
```

```
gf + geom_histogram(bins = 6)+
ggtitle('Dos dados') +
theme(plot.title = element_text(size=10, hjust = 0.5),
      axis.text=element_text(size=8),
      axis.title=element_text(size=8)) +
xlab('')
```



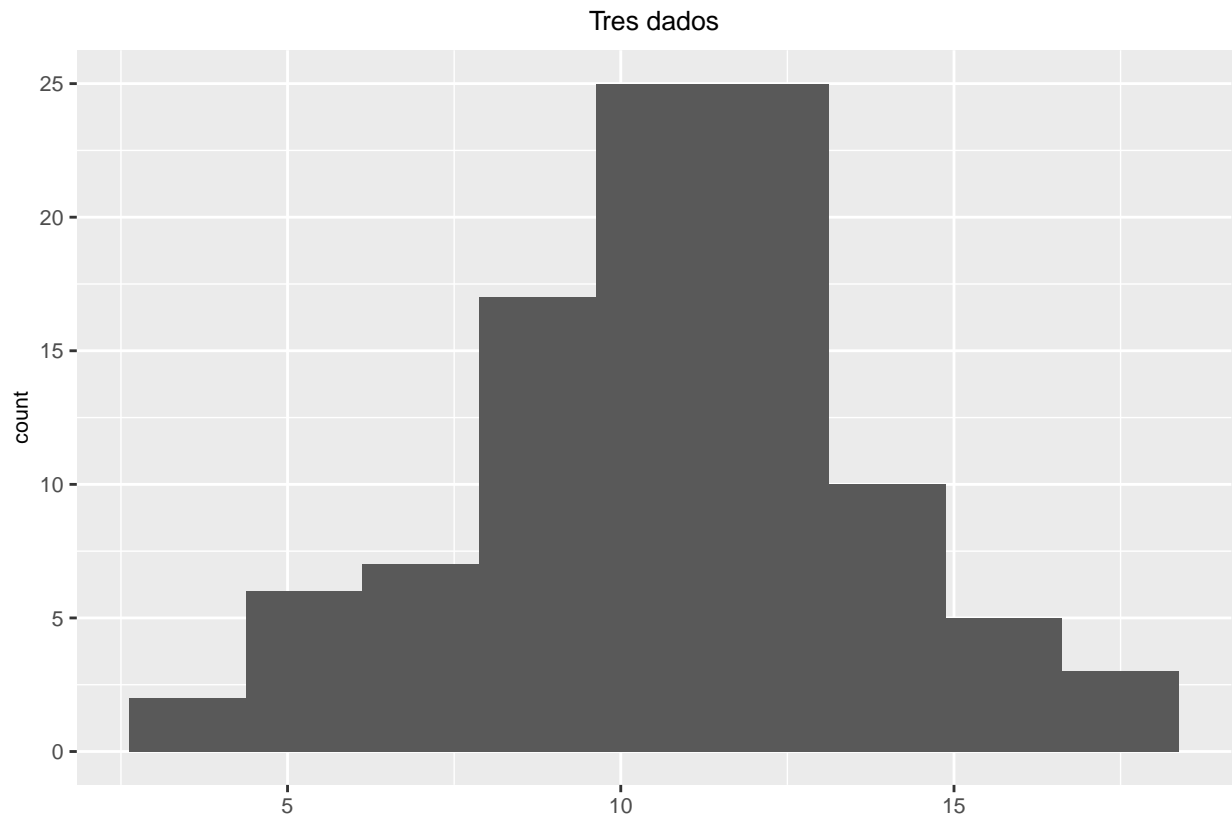
Caso de tres dados

Ejercicio: calcular este caso utilizando `set.seed(2000)`, `sample`, `ggplot2`

```
S <- rolldie(3, nsides = 6, makespace = TRUE) #lanzamiento de un dado tres veces.
S <- addrv(S, Suma3d = X1+X2+X3)
sumas<-marginal(S,vars = "Suma3d")

set.seed(2000)
n_sample<-100
prob<-sumas$probs
suma<-3:18
tres_dados<-sample(suma,n_sample,prob=prob,replace=T)
gf<-ggplot(data.frame(valores=tres_dados),aes(x=tres_dados))

gf + geom_histogram(bins = 9)+
ggtitle('Tres dados') +
theme(plot.title = element_text(size=10, hjust = 0.5),
      axis.text=element_text(size=8),
      axis.title=element_text(size=8)) +
xlab('')
```

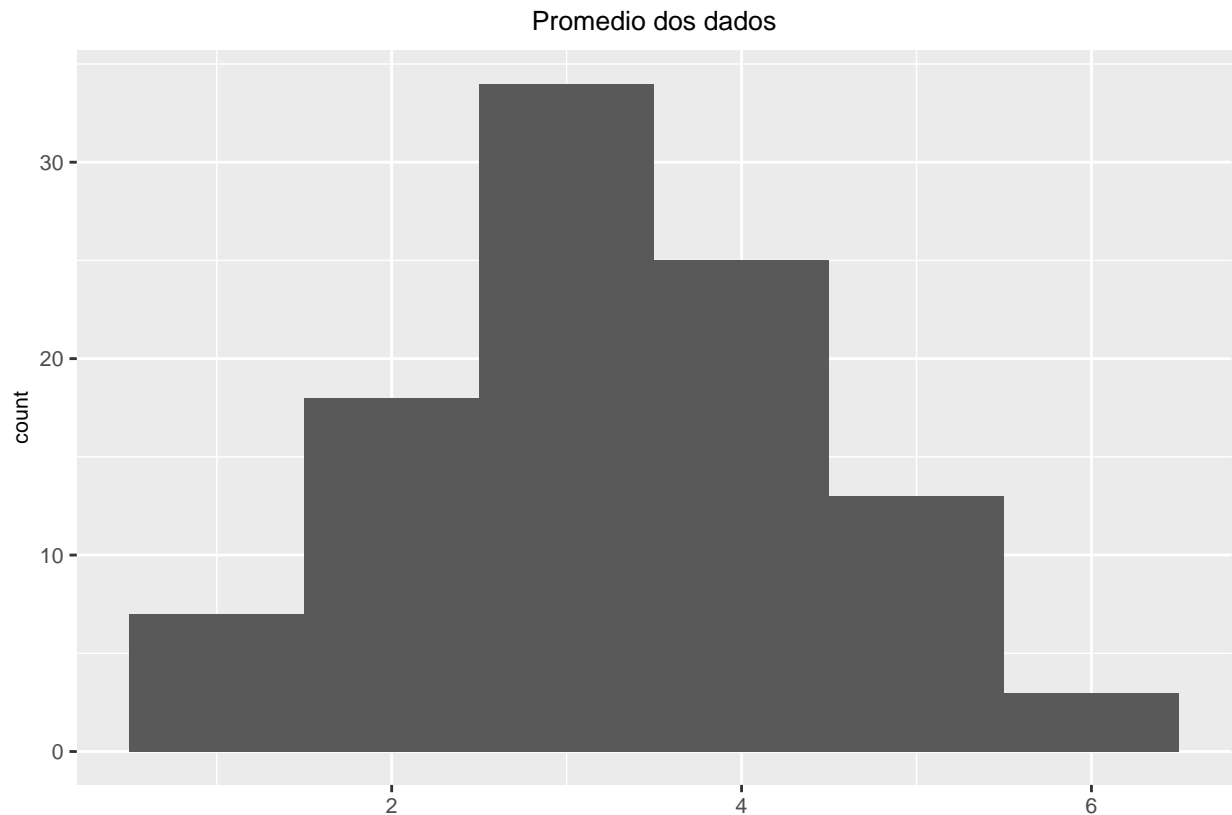



Promedios

Caso de dos dados

```
set.seed(2000)
n_sample<-100
prob<-c(1,2,3,4,5,6,5,4,3,2,1)/36
promedio<-(2:12)/2
dos_datos<-sample(promedio,n_sample,prob=prob,replace=T)
gf<-ggplot(data.frame(valores=dos_datos),aes(x=dos_datos))
```

```
gf + geom_histogram(bins = 6)+
ggtitle('Promedio dos dados') +
theme(plot.title = element_text(size=10, hjust = 0.5),
      axis.text=element_text(size=8),
      axis.title=element_text(size=8)) +
xlab('')
```



Caso de tres dados

Ejercicio: calcular este caso utilizando `set.seed(2000)`, `sample`, `ggplot2`

```
S <- rolldie(3, nsides = 6, makespace = TRUE) #lanzamiento de un dado tres veces.
S <- addrv(S, Suma3d = X1+X2+X3)
sumas<-marginal(S,vars = "Suma3d")

set.seed(2000)
n_sample<-100
prob<-sumas$probs
promedio<-(3:18)/3
tres_dados<-sample(promedio,n_sample,prob=prob,replace=T)
gf<-ggplot(data.frame(valores=tres_dados),aes(x=tres_dados))
```

La gráfica se ve así:

```
gf + geom_histogram(bins = 9)+
ggtitle('Promedio tres dados') +
theme(plot.title = element_text(size=10, hjust = 0.5),
      axis.text=element_text(size=8),
      axis.title=element_text(size=8)) +
xlab('')
```

