BoletinTema1DiegoCamachoMengual

Diego Camacho Mengual

2023-02-15

1. Considera los conjuntos de datos mammals del paquete MASS y Animals2 del paquete robustbase.

a. Mira la las características de ambos conjuntos usando la ayuda.

```
library(MASS)
library(robustbase)
```

```
## Warning: package 'robustbase' was built under R version 4.2.2
```

```
A <- mammals
B <- Animals
```

b. Usa las funciones dim, head, tail, str para una primera visión de los conjuntos de datos.

```
#Primera visión de los conjuntos de datos mediante funciones de R

dim(A)
```

```
## [1] 62  2
```

```
head(A)
```

```
##                    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
```

```
tail(A)
```

```
##                    body brain
## Echidna           3.000  25.0
## Brazilian tapir 160.000 169.0
## Tenrec            0.900   2.6
## Phalanger         1.620  11.4
## Tree shrew        0.104   2.5
## Red fox           4.235  50.4
```

```
str(A)
```

```
## 'data.frame':    62 obs. of  2 variables:
##  $ body : num  3.38 0.48 1.35 465 36.33 ...
##  $ brain: num  44.5 15.5 8.1 423 119.5 ...
```

```
dim(B)
```

```
## [1] 28  2
```

```
head(B)
```

```
##                      body brain
## Mountain beaver      1.35   8.1
## Cow                465.00 423.0
## Grey wolf           36.33 119.5
## Goat                27.66 115.0
## Guinea pig           1.04   5.5
## Dipliodocus      11700.00  50.0
```

```
tail(B)
```

```
##                     body brain
## Jaguar           100.000 157.0
## Chimpanzee        52.160 440.0
## Rat                0.280   1.9
## Brachiosaurus 87000.000 154.5
## Mole               0.122   3.0
## Pig              192.000 180.0
```

```
str(B)
```

```
## 'data.frame':    28 obs. of  2 variables:
##  $ body : num  1.35 465 36.33 27.66 1.04 ...
##  $ brain: num  8.1 423 119.5 115 5.5 ...
```

c. Muestra los nombres de las filas y las columnas (rownames, colnames)

```
colnames(A)
```

```
## [1] "body"  "brain"
```

```
rownames(A)
```

```
##  [1] "Arctic fox"               "Owl monkey"
##  [3] "Mountain beaver"          "Cow"
##  [5] "Grey wolf"                "Goat"
##  [7] "Roe deer"                 "Guinea pig"
##  [9] "Verbet"                   "Chinchilla"
## [11] "Ground squirrel"          "Arctic ground squirrel"
```

```
## [13] "African giant pouched rat" "Lesser short-tailed shrew"
## [15] "Star-nosed mole"           "Nine-banded armadillo"
## [17] "Tree hyrax"                "N.A. opossum"
## [19] "Asian elephant"            "Big brown bat"
## [21] "Donkey"                    "Horse"
## [23] "European hedgehog"         "Patas monkey"
## [25] "Cat"                       "Galago"
## [27] "Genet"                     "Giraffe"
## [29] "Gorilla"                   "Grey seal"
## [31] "Rock hyrax-a"              "Human"
## [33] "African elephant"          "Water opossum"
## [35] "Rhesus monkey"             "Kangaroo"
## [37] "Yellow-bellied marmot"     "Golden hamster"
## [39] "Mouse"                     "Little brown bat"
## [41] "Slow loris"                "Okapi"
## [43] "Rabbit"                    "Sheep"
## [45] "Jaguar"                    "Chimpanzee"
## [47] "Baboon"                    "Desert hedgehog"
## [49] "Giant armadillo"           "Rock hyrax-b"
## [51] "Raccoon"                   "Rat"
## [53] "E. American mole"          "Mole rat"
## [55] "Musk shrew"                "Pig"
## [57] "Echidna"                   "Brazilian tapir"
## [59] "Tenrec"                    "Phalanger"
## [61] "Tree shrew"                "Red fox"
```

```
colnames(B)
```

```
## [1] "body"  "brain"
```

```
rownames(B)
```

```
##  [1] "Mountain beaver"  "Cow"              "Grey wolf"        "Goat"
##  [5] "Guinea pig"       "Dipliodocus"      "Asian elephant"   "Donkey"
##  [9] "Horse"            "Potar monkey"     "Cat"              "Giraffe"
## [13] "Gorilla"          "Human"            "African elephant" "Triceratops"
## [17] "Rhesus monkey"    "Kangaroo"         "Golden hamster"   "Mouse"
## [21] "Rabbit"           "Sheep"            "Jaguar"           "Chimpanzee"
## [25] "Rat"              "Brachiosaurus"    "Mole"             "Pig"
```

d. Usa la función intersect y almacena en la variable commonAnimals los aminales que aparezcan en ambos conjuntos

```
#Comprobar los animales que aparecen en ambos df y asignarlos a la variable commonAnimals

commonAnimals <- intersect(rownames(A), rownames(B))
```

e. Usa setdiff para averiguar qué animales no están en ambos conjuntos. ¿Cuántos son ?. ¿Qué tipo de animales son?

```
#Comprobar que animales están solo en un df

setdiff(rownames(A), rownames(B))
```

```
##  [1] "Arctic fox"                "Owl monkey"
##  [3] "Roe deer"                  "Verbet"
##  [5] "Chinchilla"                "Ground squirrel"
##  [7] "Arctic ground squirrel"    "African giant pouched rat"
##  [9] "Lesser short-tailed shrew" "Star-nosed mole"
## [11] "Nine-banded armadillo"     "Tree hyrax"
## [13] "N.A. opossum"              "Big brown bat"
## [15] "European hedgehog"         "Patas monkey"
## [17] "Galago"                    "Genet"
## [19] "Grey seal"                 "Rock hyrax-a"
## [21] "Water opossum"             "Yellow-bellied marmot"
## [23] "Little brown bat"          "Slow loris"
## [25] "Okapi"                     "Baboon"
## [27] "Desert hedgehog"           "Giant armadillo"
## [29] "Rock hyrax-b"              "Raccoon"
## [31] "E. American mole"          "Mole rat"
## [33] "Musk shrew"                "Echidna"
## [35] "Brazilian tapir"           "Tenrec"
## [37] "Phalanger"                 "Tree shrew"
## [39] "Red fox"
```

```
#Son 4 los animales que solo aparecen en un df, son animales exóticos
```

f. Determina las diferencia entre los animales que no aparecen en ambos conjuntos.

-Los animales que aparecen en un solo conjunto son de especies distintas

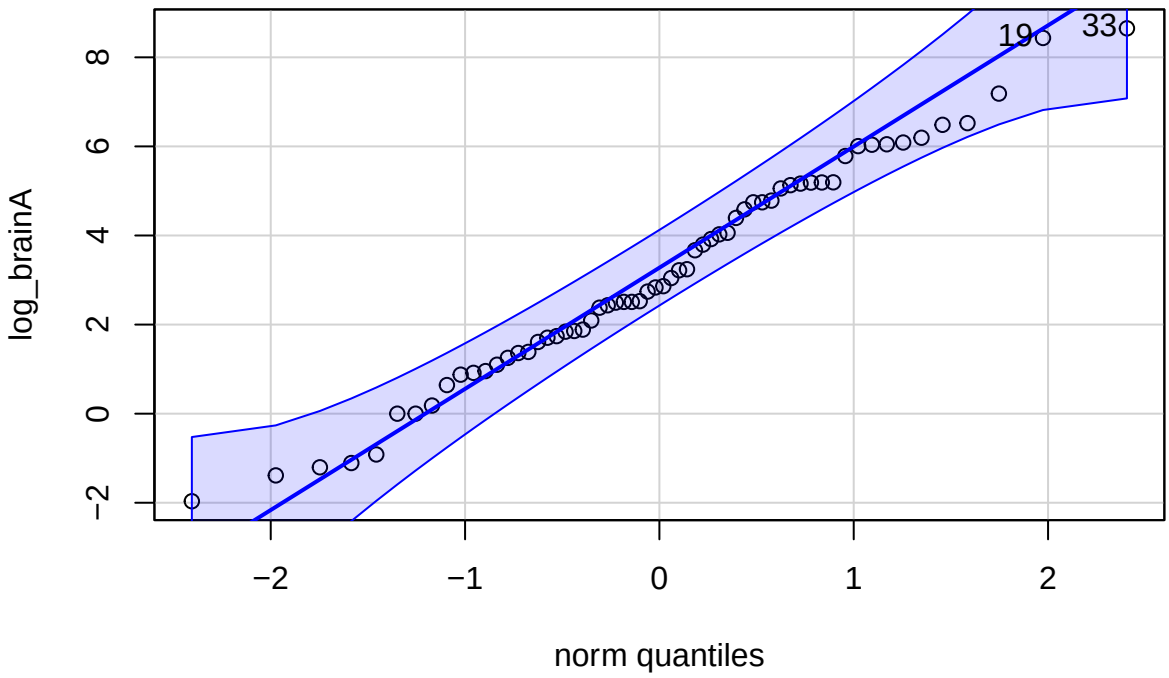
2. La funcion qqPlot del paquete car puede ser utilizada para determinar gráficamente si una serie de puntos siguen una distribución de datos Gaussiana. Si las muestras están dentro de las líneas discontinuas podemos indicar que siguen una distribución Gaussiana con un 95 % de confianza. Utilizando esta función representa el logaritmo neperiano (log) del peso del cerebro (brain weigths) del registro de datos mammals del paquete MASS y conjunto de datos Animals2 de la librería robustbase. ¿Presentan el mismo comportamiento ?.¿Podríamos decir que siguen una distribución Gaussiana ?

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
```

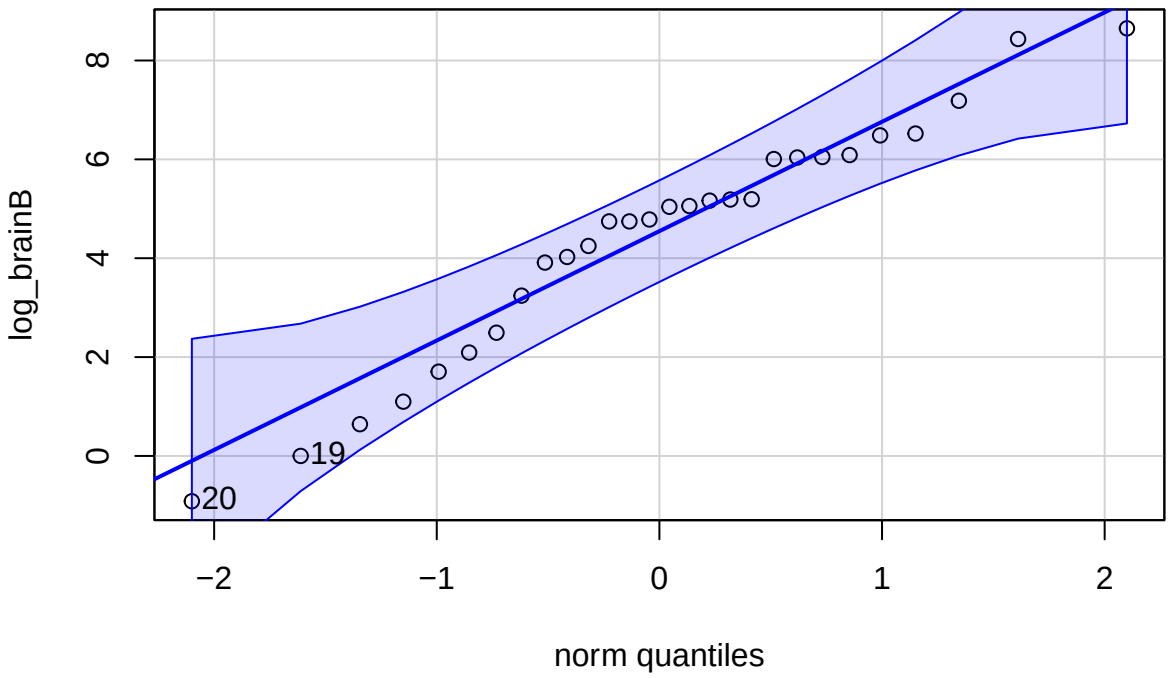
```
## Loading required package: carData
```

```
log_brainA <- log(A$brain)

qqPlot(log_brainA)
```




```
## [1] 33 19
```

```
log_brainB <- log(B$brain)

qqPlot(log_brainB)
```




```
## [1] 20 19
```

```
#Se podría decir que presentan comportamientos parecidos y que ambos siguen una distribución gaussiana
```

3. La función library sin argumentos abre una ventana y muestra las librerías que han sido instaladas.

a. Asigna el valor devuelto por esta función a la variable libReturn y observa su estructura.

```
libReturn <- library()
class(libReturn)
```

```
## [1] "libraryIQR"
```

```
#La función library devuelve una lista
```

b. Uno de los elementos de la lista es una matriz de caracteres. Muestra por pantalla los 5 primeros elementos de esta matriz usando la función head.

```
m_lib <- libReturn$results
class(m_lib)
```

```
## [1] "matrix" "array"
```

```
head(m_lib, 5)
```

```
##      Package      LibPath
## [1,] "abind"      "C:/Users/diego/AppData/Local/R/win-library/4.2"
## [2,] "askpass"    "C:/Users/diego/AppData/Local/R/win-library/4.2"
## [3,] "assertthat" "C:/Users/diego/AppData/Local/R/win-library/4.2"
## [4,] "backports"  "C:/Users/diego/AppData/Local/R/win-library/4.2"
## [5,] "base64enc"  "C:/Users/diego/AppData/Local/R/win-library/4.2"
##      Title
## [1,] "Combine Multidimensional Arrays"
## [2,] "Safe Password Entry for R, Git, and SSH"
## [3,] "Easy Pre and Post Assertions"
## [4,] "Reimplementations of Functions Introduced Since R-3.0.0"
## [5,] "Tools for base64 encoding"
```

c. Determina el número de librerías que tienes instaladas.

```
dim(m_lib)
```

```
## [1] 187   3
```

```
#Tengo instaladas 187 librerías
```

4. En las transparencias del tema 1 se citan los primeros pasos a seguir cuando se analiza un nuevo conjunto de datos.

a. Determina las tres primeras etapas para el conjunto de datos cabbages del paquete MASS

```
dim(cabbages)
```

```
## [1] 60  4
```

```
str(cabbages)
```

```
## 'data.frame':    60 obs. of  4 variables:
##  $ Cult  : Factor w/ 2 levels "c39","c52": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date  : Factor w/ 3 levels "d16","d20","d21": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HeadWt: num  2.5 2.2 3.1 4.3 2.5 4.3 3.8 4.3 1.7 3.1 ...
##  $ VitC  : int  51 55 45 42 53 50 50 52 56 49 ...
```

```
head(cabbages)
```

```
##   Cult Date HeadWt VitC
## 1  c39  d16    2.5   51
## 2  c39  d16    2.2   55
## 3  c39  d16    3.1   45
## 4  c39  d16    4.3   42
## 5  c39  d16    2.5   53
## 6  c39  d16    4.3   50
```

```
tail(cabbages)
```

```
##    Cult Date HeadWt VitC
## 55  c52  d21    1.7   71
## 56  c52  d21    1.6   72
## 57  c52  d21    1.4   62
## 58  c52  d21    1.0   68
## 59  c52  d21    1.5   66
## 60  c52  d21    1.6   72
```

```
summary(cabbages)
```

```
##   Cult     Date        HeadWt           VitC
##  c39:30   d16:20   Min.   :1.000   Min.   :41.00
##  c52:30   d20:20   1st Qu.:1.875   1st Qu.:50.75
##           d21:20   Median :2.550   Median :56.00
##                    Mean   :2.593   Mean   :57.95
##                    3rd Qu.:3.125   3rd Qu.:66.25
##                    Max.   :4.300   Max.   :84.00
```

b. Puedes determinar el número de valores perdidos (almacenados como NA en R) usando la función is.na. Determina el número de valores perdidos para cada una de las variables del conjunto cabbages.

```
sum(is.na(cabbages))
```

```
## [1] 0
```

```
#No hay ningún na
```

c. Repite los apartados anteriores con el conjunto de datos Chile del paquete carData.

```
library(carData)

dim(Chile)
```

```
## [1] 2700    8
```

```
str(Chile)
```

```
## 'data.frame':    2700 obs. of  8 variables:
##  $ region    : Factor w/ 5 levels "C","M","N","S",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ population: int  175000 175000 175000 175000 175000 175000 175000 175000 175000 175000 ...
##  $ sex       : Factor w/ 2 levels "F","M": 2 2 1 1 1 1 2 1 1 2 ...
##  $ age       : int  65 29 38 49 23 28 26 24 41 41 ...
##  $ education : Factor w/ 3 levels "P","PS","S": 1 2 1 1 3 1 2 3 1 1 ...
##  $ income    : int  35000 7500 15000 35000 35000 7500 35000 15000 15000 15000 ...
##  $ statusquo : num  1.01 -1.3 1.23 -1.03 -1.1 ...
##  $ vote      : Factor w/ 4 levels "A","N","U","Y": 4 2 4 2 2 2 2 2 3 2 ...
```

```
head(Chile)
```

```
##   region population sex age education income statusquo vote
## 1      N     175000   M  65         P  35000   1.00820    Y
## 2      N     175000   M  29        PS   7500  -1.29617    N
## 3      N     175000   F  38         P  15000   1.23072    Y
## 4      N     175000   F  49         P  35000  -1.03163    N
## 5      N     175000   F  23         S  35000  -1.10496    N
## 6      N     175000   F  28         P   7500  -1.04685    N
```

```
tail(Chile)
```

```
##      region population sex age education income statusquo vote
## 2695      M      15000   M  42         S  35000  -0.00233    U
## 2696      M      15000   M  42         P  15000  -1.26247    N
## 2697      M      15000   F  28         P  15000   1.32950    Y
## 2698      M      15000   F  44         P  75000   1.42045    Y
## 2699      M      15000   M  21         S  75000   0.18315 <NA>
## 2700      M      15000   M  20        PS  35000   1.38179    Y
```

```
summary(Chile)
```

```
##  region     population     sex           age        education
##  C :600   Min.   :  3750   F:1379   Min.   :18.00   P   :1107
##  M :100   1st Qu.: 25000   M:1321   1st Qu.:26.00   PS  : 462
##  N :322   Median :175000            Median :36.00   S   :1120
##  S :718   Mean   :152222            Mean   :38.55   NA's:  11
##  SA:960   3rd Qu.:250000            3rd Qu.:49.00
##           Max.   :250000            Max.   :70.00
##                                     NA's   :1
##      income         statusquo          vote
##  Min.   :  2500   Min.   :-1.80301   A   :187
##  1st Qu.:  7500   1st Qu.:-1.00223   N   :889
##  Median : 15000   Median :-0.04558   U   :588
##  Mean   : 33876   Mean   : 0.00000   Y   :868
##  3rd Qu.: 35000   3rd Qu.: 0.96857   NA's:168
##  Max.   :200000   Max.   : 2.04859
##  NA's   :98       NA's   :17
```

```
sum(is.na(Chile))
```

```
## [1] 295
```

d. Utiliza la función summary, sobre cabbages y Chile y observa como, además de otros estadísticos, también devuelve el número de valores perdidos de cada variable.

```
summary(Chile)
```

```
##  region     population     sex           age        education
##  C :600   Min.   :  3750   F:1379   Min.   :18.00   P   :1107
##  M :100   1st Qu.: 25000   M:1321   1st Qu.:26.00   PS  : 462
```

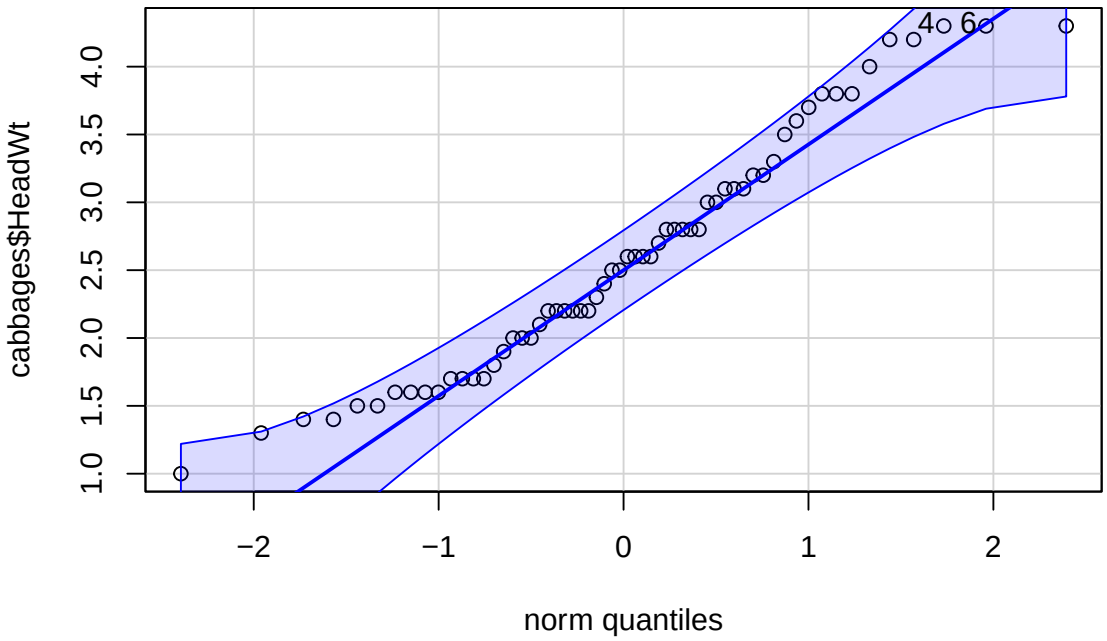
```
##  N :322   Median :175000           Median :36.00   S   :1120
##  S :718   Mean   :152222           Mean   :38.55   NA's:  11
##  SA:960   3rd Qu.:250000           3rd Qu.:49.00
##           Max.   :250000           Max.   :70.00
##                                    NA's   :1
##      income         statusquo          vote
##  Min.   :  2500   Min.   :-1.80301   A   :187
##  1st Qu.:  7500   1st Qu.:-1.00223   N   :889
##  Median : 15000   Median :-0.04558   U   :588
##  Mean   : 33876   Mean   : 0.00000   Y   :868
##  3rd Qu.: 35000   3rd Qu.: 0.96857   NA's:168
##  Max.   :200000   Max.   : 2.04859
##  NA's   :98       NA's   :17
```

```
summary(cabbages)
```

```
##   Cult     Date        HeadWt           VitC
##  c39:30   d16:20   Min.   :1.000   Min.   :41.00
##  c52:30   d20:20   1st Qu.:1.875   1st Qu.:50.75
##           d21:20   Median :2.550   Median :56.00
##                    Mean   :2.593   Mean   :57.95
##                    3rd Qu.:3.125   3rd Qu.:66.25
##                    Max.   :4.300   Max.   :84.00
```

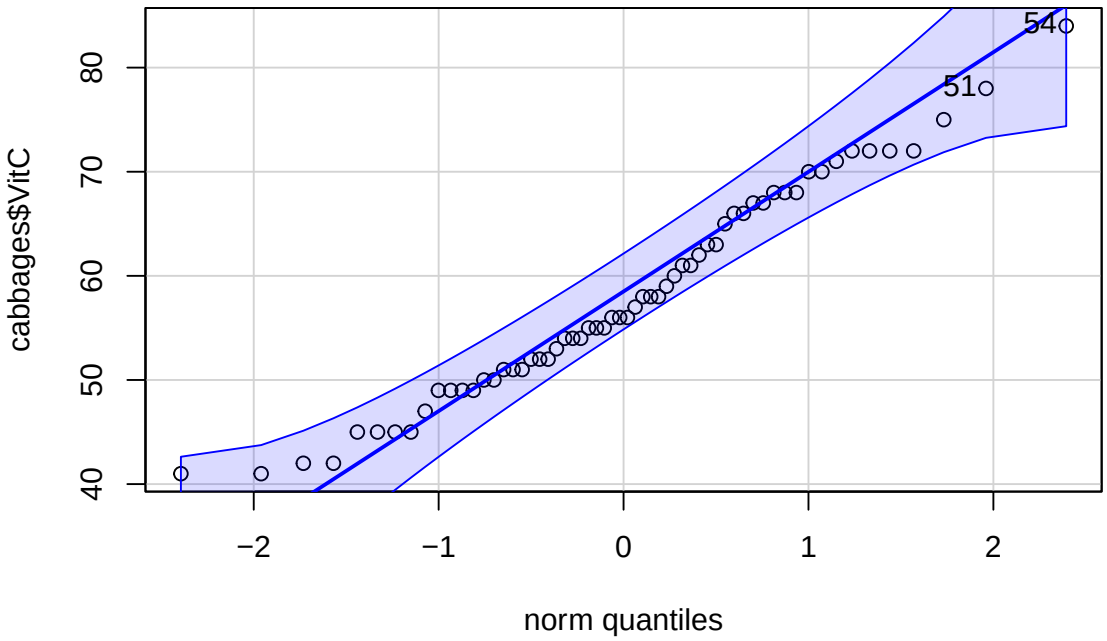
5. Muchas pruebas estadísticas suponen que los datos siguen una distribución Gaussiana. Utiliza la aproximación visual proporcionada por qqPlot para determinar si podemos asumir que las variables HeadWt y VitC del conjunto cabbages verifican esta condición.

```
qqPlot(cabbages$HeadWt)
```




```
## [1] 4 6
```

```
qqPlot(cabbages$VitC)
```




```
## [1] 54 51
```

```
#Podemos asumir que ambos cumplen la condición
```

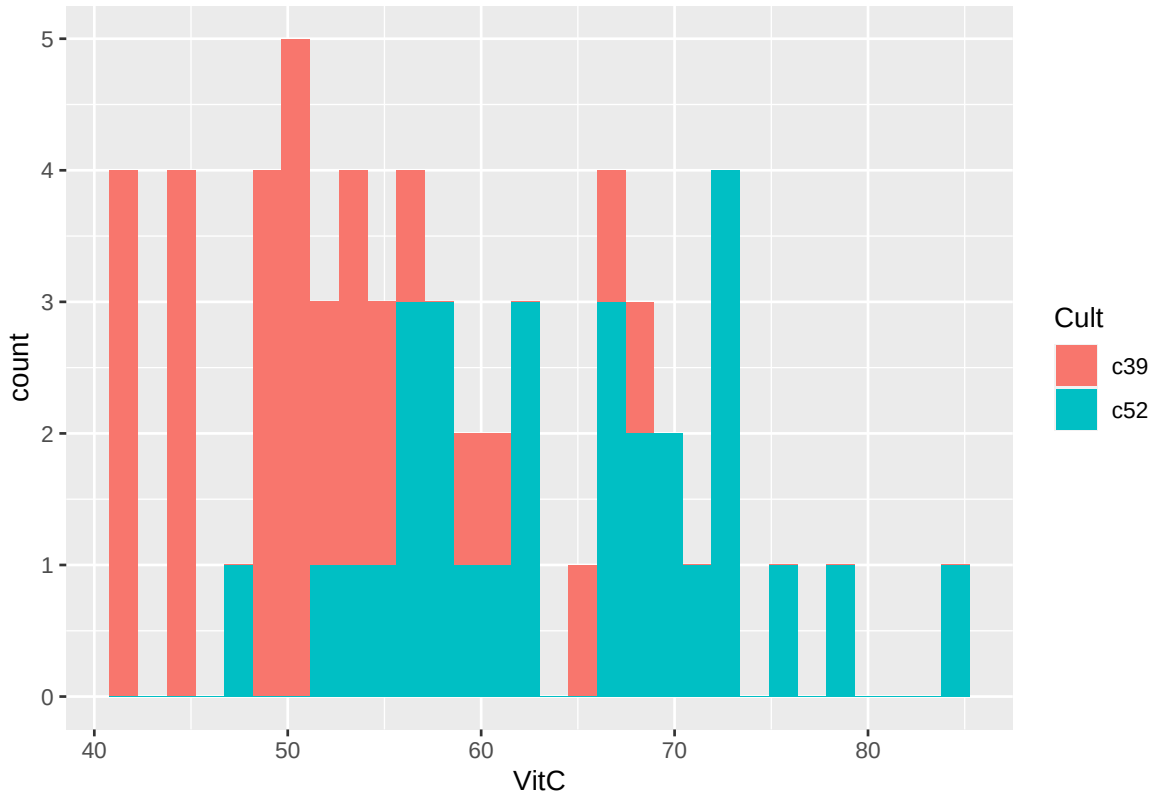
6. Una representación habitual, para determinar la distribución de los datos de una variable cuantitativa es el histograma (hist). Determina, de forma aproximada, utilizando el histograma, si hay diferencias entre los contenidos de vitamina C (VitC), para las diferentes variedades de calabaza (variable Cult), en el conjunto de datos cabbages.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

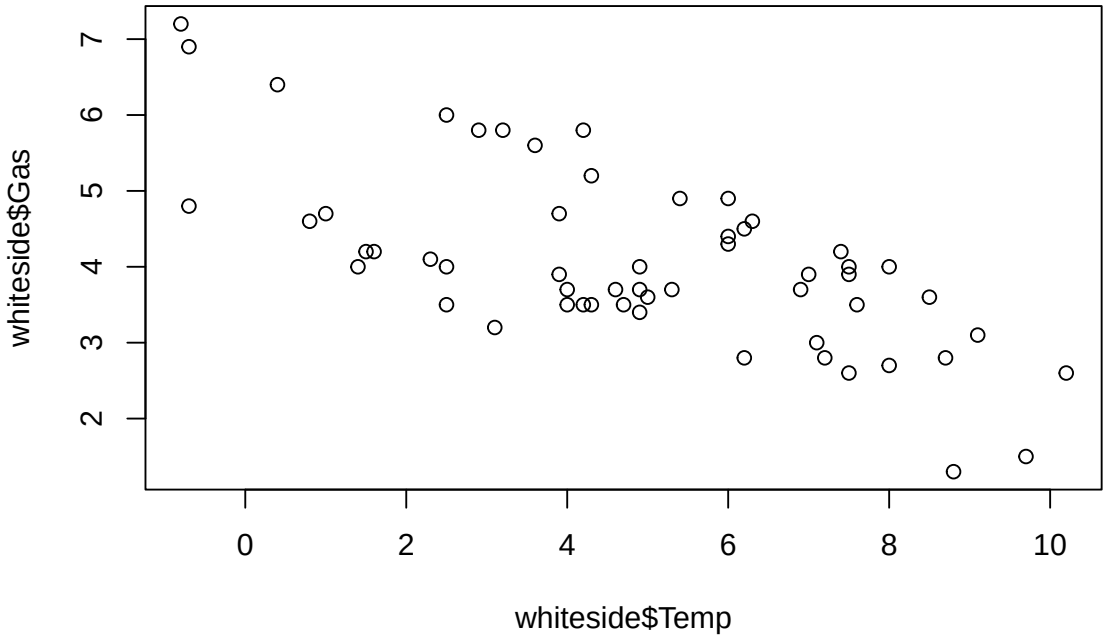
```
ggplot(data=cabbages, aes(x=VitC, fill=Cult))+geom_histogram(alpha=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

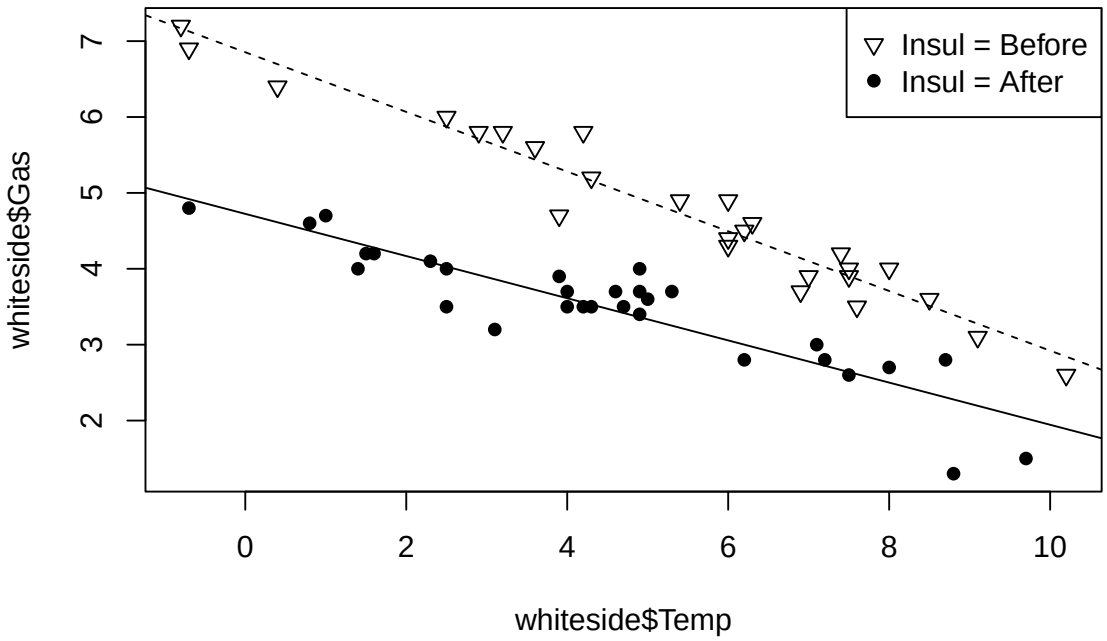


7. Un modelo sencillo para relacionar variables es la predicción lineal. En el siguiente ejemplo se utiliza el conjunto de datos whiteside, de la librería MASS. Esta aproximación propone un modelo que predice una variable a partir de otra. Una primera etapa para plantear esta aproximación sería representar ambas variables mediante un diagrama de dispersión (Gráfico XY) y determinar si la relación entre variables "parece" lineal. Si es así, podemos plantear un modelo lineal (en este caso según un factor), donde se aprecia claramente que existe una relación lineal entre las dos variables consideradas. Observa y ejecuta el siguiente código.

```
#Diagrama de dispersión global.
plot(whiteside$Temp, whiteside$Gas)
```

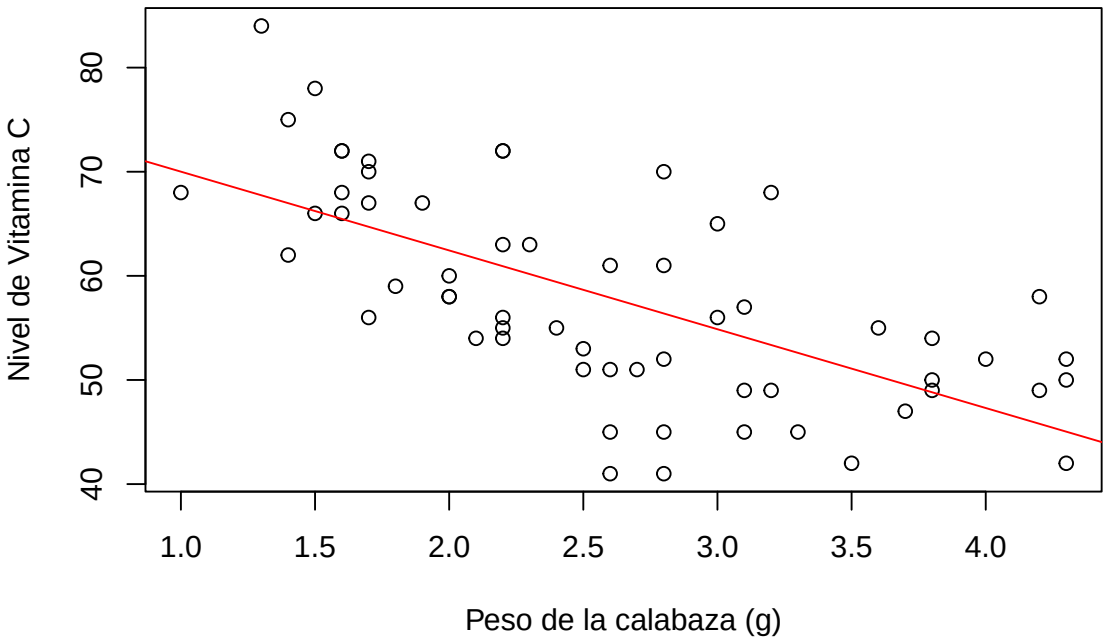



```
#Diagrama de dispersión etiquetando según un factor.
plot(whiteside$Temp, whiteside$Gas, pch=c(6,16)[whiteside$Insul])
legend(x="topright",legend=c("Insul = Before","Insul = After"), pch=c(6,16))
# Planteamos 2 modelos lineales, uno para los datos de cada factor
Model1 <- lm(Gas ~ Temp, data = whiteside, subset = which(Insul == "Before"))
Model2 <- lm(Gas ~ Temp, data = whiteside, subset = which(Insul == "After"))
# Representamos las rectas correspondientes a cada modelo lineal
abline(Model1, lty=2)
abline(Model2)
```



a. Utiliza un procedimiento análogo para determinar si se aprecia una relación lineal entre los niveles de vitamina C, VitC en función del peso de la calabaza, HeadWt, en el conjunto de datos cabbages.

```
plot(cabbages$HeadWt, cabbages$VitC, xlab="Peso de la calabaza (g)", ylab="Nivel de Vitamina C")

model <- lm(VitC ~ HeadWt, data=cabbages)

abline(model, col="red")
```




```
#Podemos comprobar que no se ajusta muy bien por lo que no habrá una relación lineal fuerte
```

b. Repite el apartado anterior, pero obteniendo un modelo para cada una de las dos variedades de calabaza, Cult. VerParámetros básicos plot.

c. Usa summary con cada uno de los modelos obtenidos y observa Coefficients. Dado que hemos planteado un modelo y = mx+n, donde y = V itC y x = HeadW t. La función lm nos permite obtener (Intercept); n y la pendiente HeadWt; m (además de otros parámetros adicionales que evalúan la caracterísiticas del modelo). Observa que en todos los casos, la pendiene es negativa indicando que las calabazas de más peso contienen menos vitamina C. No te preocupes por el resto de parámetros del modelo, por el momento