Estadistica descriptiva

Wilson Castro Z

Actualizado 23/02/2020

1. Practica de R: Estadistica Descriptiva

1.1 Datos

Creando el dataset (Estructura de datos). Se toman los datos de estatura de estudiantes:

```
datosEstud<-c(1.2,  1.21,  1.21,  1.21,  1.21,  1.22,  1.22,  1.22,  1.22,  1.23,  1.23,  1.24
```

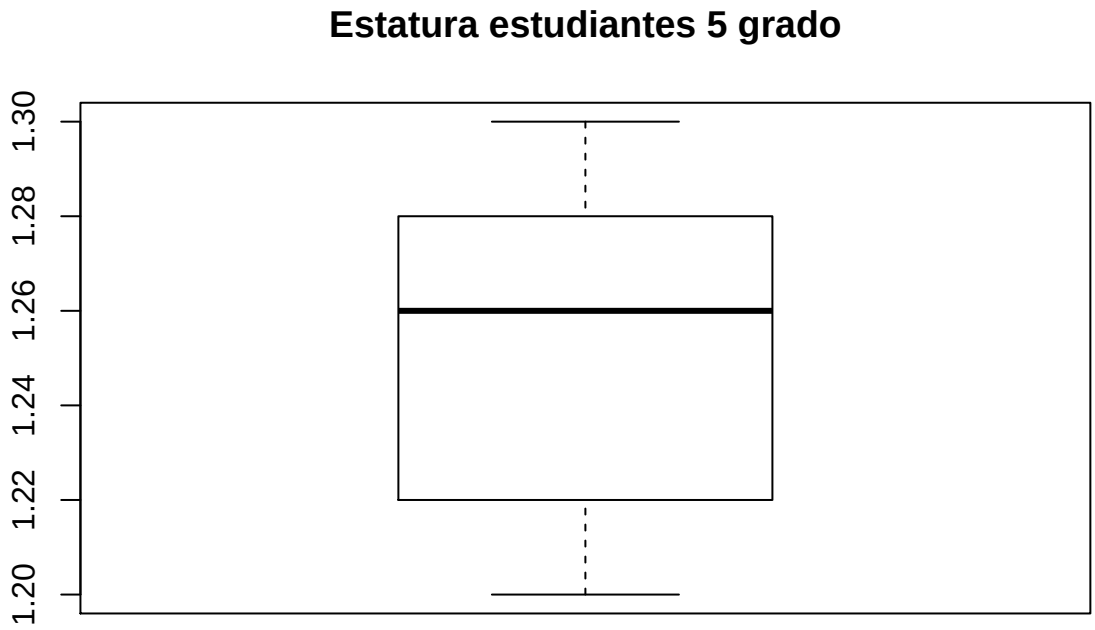
1.2 Exploracion del dataset y graficos.

Se realiza una primera exploracion de los datos con summary y se hace el diagrama de caja y bigotes (o caja y extension) con el comando boxplot():

```
summary(datosEstud,main="Estatura estudiantes 5 grado")
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   1.220   1.260   1.253   1.280   1.300
```

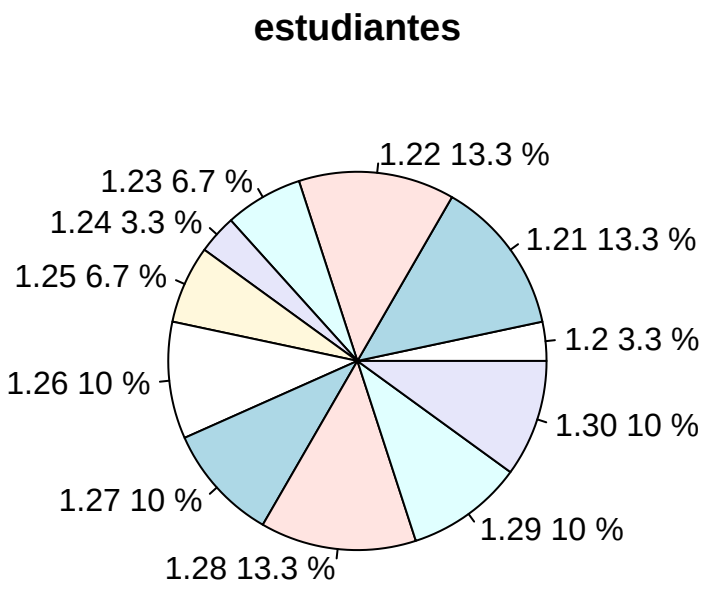
```
boxplot(datosEstud,main="Estatura estudiantes 5 grado")
```



Otros graficos Se plantea la construccion de un diagrama de torta (pie) o de sectores.

```r
#Calculo de los porcentajes (Sustituya datosEstud por el vector de datos)
xp=NULL
pct=table(datosEstud)
for (i in 1:length(pct))
{xp=round(cbind(xp,100*pct[[i]]/sum(pct)),1)}
```

```r
#Con table se establecen las frecuencias
#pct=table(datosEstud) # matriz
#Ejecute el xp de funcionesWC.R
etiquetas=c("1.2","1.21","1.22","1.23","1.24","1.25","1.26","1.27","1.28","1.29","1.30")
etiq=paste(etiquetas,xp,"%")
pie(table(datosEstud),labels=etiq,main="estudiantes")
```




```r
barplot(table(datosEstud),main="Estatura estudiantes 5")
```

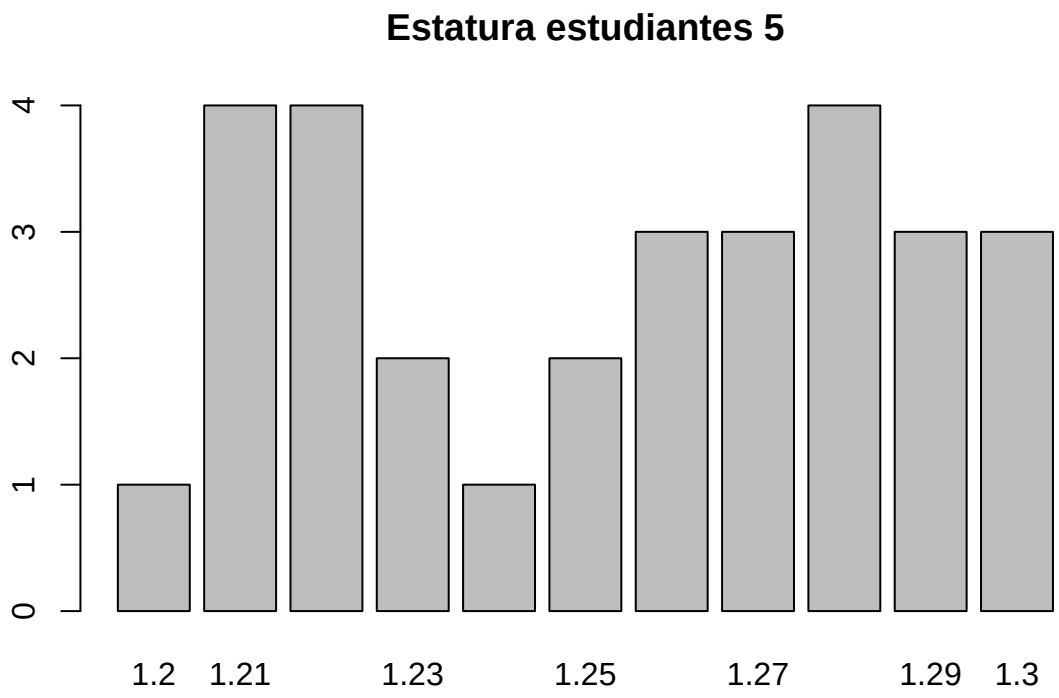


Diagrama de tallo y hojas

```
stem(datosEstud,scale = 2, width = 20, atom =0.01)
```

```
##
##   The decimal point is 2 digit(s) to the left of the |
##
##   120 | 0
##   121 | 0000
##   122 | 0000
##   123 | 00
##   124 | 0
##   125 | 00
##   126 | 000
##   127 | 000
##   128 | 0000
##   129 | 000
##   130 | 000
```

1.3 Medidas de tendencia central.

Primero se crean unas funciones para algunos calculos

```
#CODIGOS DE FUNCIONES, por Ing. Wilson Castro Zapata
#Copie el codigo correspondiente y ejecutelo, luego se llama la funcion
# dando el nombre de la funcion y entre parentesis el vector de datos al que le
#va a calcular la funcion
```

```r
#Funcion que suma dos numeros, vectores, etc
suma=function(x,y)
{
  return(x+y)
}
#Funcion moda tomada de: https://www.tutorialspoint.com/r/r_mean_median_mode.htm
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

#Funcion normalize
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

#Funcion normalizar que normaliza un vector segun la distribucion normal
znormalizar <- function(x) {
  return ((x - mean(x))/sd(x))
}

#Funcion para calcular la media geometrica
mgeometrica <-function(x)
{
  n=length(x)
  producto=prod(x)
  return(producto**(1/n))
}

#Funcion promedio ponderado
mponderada<-function(x,f)
{
  if(length(x)==length(f))
  {if(sum(f)==1) return(x%*%f)
    else return("el vector de probabilidad debe sumar 1.0")
    }
  return("Los vectores deben ser del mismo tamaÃ±o")
}
marmonica <-function(x) #Tambien se puede usar 1/mean(1/x)

{
  n=length(x)
  sumarecip=sum(1/x)
  return(n/sumarecip)
}
```

Las medidas de tendencia central son:

```r
#Media aritmetica
mean(datosEstud)
```

```
## [1] 1.253333
```

```r
median(datosEstud)
```

```
## [1] 1.26
```

```
#Para la moda como se trata del valor que mas se repite, se encuentra la
#frecuencia mÃ¡xima con table() por observacion. O se ejecuta la funcion getmode de funcionesWC.R:
moda=getmode(datosEstud)
#Media geometrica
mediaG=mgeometrica(datosEstud)
#Media armonica
mediaH=marmonica(datosEstud)
mediaH2=1/mean(1/datosEstud)
```

1.4 Medidas de variacion

```
#Rango:
Rango=max(datosEstud)-min(datosEstud)
#Rango intercuartilico:
Interc=IQR(datosEstud)
#De forma manual como:
Interc2=quantile(datosEstud,0.75)-quantile(datosEstud,0.25)
#Varianza:
S2=var(datosEstud)
#DesviaciÃ²n estÃ¡ndar:
DesvEstandar=sd(datosEstud)
```

1.5 Medidas de Asimetria y Forma (coeficiente de asimetria y curtosis)

```
#Instalar paquete moments si no esta instalado
library(moments)
skewness(datosEstud)
```

```
## [1] -0.1130675
```

```
kurtosis(datosEstud)
```

```
## [1] 1.605299
```

1.6 Aplicacion de las funciones de normalizacion:

```
#Ejecute el archivo de funciones funcionesWC.R
#1. Normalizacion entre el rango: maximo y minimo
xnormal1=normalize(datosEstud)
#2. Normalizacion con la D. normal
xnormz=znormalizar(datosEstud)
```

1.7 Tabla de frecuencias (Basado en estadistica descriptiva y probabilidad con aplicaciones en R, de Cristian Tellez y Diego Lemus)

Para la tabla de frecuencias:

```
fabsol2<-table(datosEstud)
frelativas2<-round(prop.table(fabsol2)*100,3)
fabsAcum2<-cumsum(fabsol2)
fRelatAcum2<-cumsum(frelativas2)
tabla2<-cbind(fabsol2,fabsAcum2,frelativas2,fRelatAcum2)
colnames(tabla2)<-c("n","N","f","F")
tabla2
```

```
##      n  N      f      F
## 1.2  1  1  3.333  3.333
## 1.21 4  5 13.333 16.666
## 1.22 4  9 13.333 29.999
## 1.23 2 11  6.667 36.666
## 1.24 1 12  3.333 39.999
## 1.25 2 14  6.667 46.666
## 1.26 3 17 10.000 56.666
## 1.27 3 20 10.000 66.666
## 1.28 4 24 13.333 79.999
## 1.29 3 27 10.000 89.999
## 1.3  3 30 10.000 99.999
```

1.8 Tabla con Datos Agrupados (Basado en estadistica descriptiva y probabilidad con aplicaciones en R, de Cristian Tellez y Diego Lemus)

Calculo mediante la formula de Sturges del No. de intervalos:

```
n<-length(datosEstud)
nclases<-1+3.3*log10(n)
nclases<-round(1+log10(n)/log10(2))
```

Luego se definen 6 intervalos. Como breaks toma desde el minimo hasta el maximo, se agrega 1

```
frecuencias<-hist(datosEstud,breaks=nclases,right=FALSE,plot=F)
marcas_clase<-frecuencias$mids
fabsolutas<-frecuencias$counts
frelativas<-(round(frecuencias$counts/sum(frecuencias$counts),3))*100
abs_acum<-cumsum(fabsolutas)
rel_acum<-cumsum(frelativas)
```

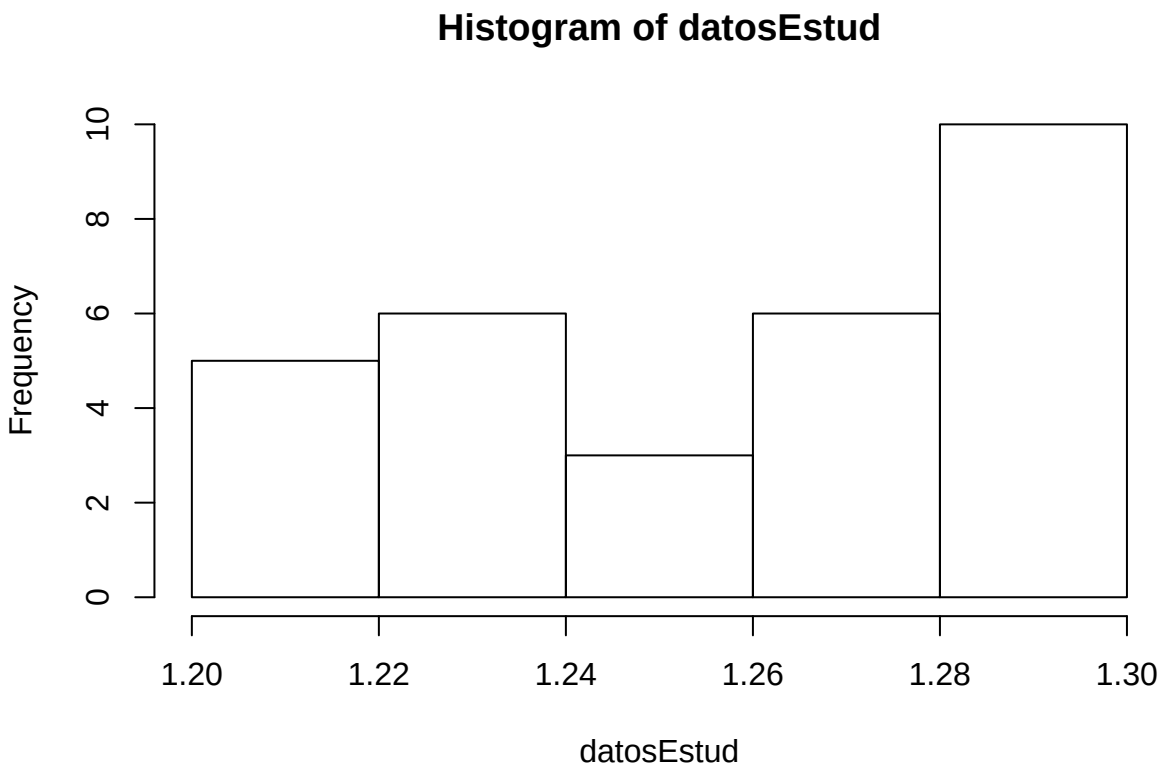
Generacion de la tabla de frecuencias con datos agrupados:

```
tabla3<-cbind(marcas_clase,fabsolutas,abs_acum,frelativas,rel_acum)
colnames(tabla3)<-c("Mc","n","N","f","F")
rownames(tabla3)<-c("[1.20, 1.22)","[1.22, 1.24)","[1.24, 1.26)","[1.26, 1.28)","[1.28, 1.30]")
tabla3
```

```
##                Mc  n  N    f     F
## [1.20, 1.22) 1.21  5  5 16.7  16.7
## [1.22, 1.24) 1.23  6 11 20.0  36.7
## [1.24, 1.26) 1.25  3 14 10.0  46.7
## [1.26, 1.28) 1.27  6 20 20.0  66.7
## [1.28, 1.30] 1.29 10 30 33.3 100.0
```

El histograma es:

```
hist(datosEstud,breaks=nclases,right=FALSE,plot=T)
```




```
summary(datosEstud)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.200   1.220   1.260   1.253   1.280   1.300
```

```
sum(frecuencias$counts,2)
```

```
## [1] 32
```

Prueba con "Sturges". Resuelva la tabla de frecuencias iniciando con los comandos dados a continuaciÃ³n y observe los resultados:

```
frecuencias<-hist(datosEstud,breaks="Sturges",right=FALSE,plot=F) #IDENTICO
frecuencias<-hist(datosEstud,breaks="Sturges",right=TRUE,plot=F) #RARO
```

2. Leer una base de datos

Ubique el dataset en la misma carpeta donde esta trabajando el archivo de R, puede en RStudio ajustar la Session a esta carpeta.

```
DiabetesData<-read.csv('diabetes.csv', header=TRUE,sep=';')
#Mirar los primeros 10 registros:
head(DiabetesData)
```

```
##     id chol stab.glu hdl       ratio       glyhb   location age gender
## 1 1000  203       82  56 3,599999905 4,309999943 Buckingham  46 female
## 2 1001  165       97  24 6,900000095 4,440000057 Buckingham  29 female
## 3 1002  228       92  37 6,199999809 4,639999866 Buckingham  58 female
```

```
## 4 1003   78       93  12         6,5 4,630000114 Buckingham  67   male
## 5 1005  249       90  28 8,899999619  7,71999979 Buckingham  64   male
## 6 1008  248       94  69 3,599999905 4,809999943 Buckingham  34   male
##   height weight  frame bp.1s bp.1d bp.2s bp.2d waist hip time.ppn
## 1     62    121 medium   118    59    NA    NA    29  38      720
## 2     64    218  large   112    68    NA    NA    46  48      360
## 3     61    256  large   190    92   185    92    49  57      180
## 4     67    119  large   110    50    NA    NA    33  38      480
## 5     68    183 medium   138    80    NA    NA    44  41      300
## 6     71    190  large   132    86    NA    NA    36  42      195
```

```
#Estructura de los datos
str(DiabetesData)
```

```
## 'data.frame':    403 obs. of  19 variables:
##  $ id      : int  1000 1001 1002 1003 1005 1008 1011 1015 1016 1022 ...
##  $ chol    : int  203 165 228 78 249 248 195 227 177 263 ...
##  $ stab.glu: int  82 97 92 93 90 94 92 75 87 89 ...
##  $ hdl     : int  56 24 37 12 28 69 41 44 49 40 ...
##  $ ratio   : Factor w/ 70 levels "","1,5","1,899999976",..: 23 56 49 52 69 23 35 39 23 53 ...
##  $ glyhb   : Factor w/ 240 levels "","10,06999969",..: 81 91 109 108 218 125 128 54 128 190 ...
##  $ location: Factor w/ 2 levels "Buckingham","Louisa": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age     : int  46 29 58 67 64 34 30 37 45 55 ...
##  $ gender  : Factor w/ 2 levels "female","male": 1 1 1 2 2 2 2 2 2 1 ...
##  $ height  : int  62 64 61 67 68 71 69 59 69 63 ...
##  $ weight  : int  121 218 256 119 183 190 191 170 166 202 ...
##  $ frame   : Factor w/ 4 levels "","large","medium",..: 3 2 2 2 3 2 3 3 2 4 ...
##  $ bp.1s   : int  118 112 190 110 138 132 161 NA 160 108 ...
##  $ bp.1d   : int  59 68 92 50 80 86 112 NA 80 72 ...
##  $ bp.2s   : int  NA NA 185 NA NA NA 161 NA 128 NA ...
##  $ bp.2d   : int  NA NA 92 NA NA NA 112 NA 86 NA ...
##  $ waist   : int  29 46 49 33 44 36 46 34 34 45 ...
##  $ hip     : int  38 48 57 38 41 42 49 39 40 50 ...
##  $ time.ppn: int  720 360 180 480 300 195 720 1020 300 240 ...
```

```
#Se hace la tabla de frecuencias para la variable weight
summary(DiabetesData$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    99.0   151.0   172.5   177.6   200.0   325.0       1
```

Como hay un NA, se quita con

```
names(DiabetesData)
```

```
##  [1] "id"       "chol"     "stab.glu" "hdl"      "ratio"    "glyhb"
##  [7] "location" "age"      "gender"   "height"   "weight"   "frame"
## [13] "bp.1s"    "bp.1d"    "bp.2s"    "bp.2d"    "waist"    "hip"
## [19] "time.ppn"
```

```
head(DiabetesData[,11])
```

```
## [1] 121 218 256 119 183 190
```

```
DiabetesWeight<-subset(DiabetesData, (!is.na(DiabetesData[,11])))
head(DiabetesWeight)
```

```
##     id chol stab.glu hdl       ratio       glyhb   location age gender
## 1 1000  203       82  56 3,599999905 4,309999943 Buckingham  46 female
```

```
## 2 1001  165       97  24 6,900000095 4,440000057 Buckingham  29 female
## 3 1002  228       92  37 6,199999809 4,639999866 Buckingham  58 female
## 4 1003   78       93  12         6,5 4,630000114 Buckingham  67   male
## 5 1005  249       90  28 8,899999619  7,71999979 Buckingham  64   male
## 6 1008  248       94  69 3,599999905 4,809999943 Buckingham  34   male
##   height weight  frame bp.1s bp.1d bp.2s bp.2d waist hip time.ppn
## 1     62    121 medium   118    59    NA    NA    29  38      720
## 2     64    218  large   112    68    NA    NA    46  48      360
## 3     61    256  large   190    92   185    92    49  57      180
## 4     67    119  large   110    50    NA    NA    33  38      480
## 5     68    183 medium   138    80    NA    NA    44  41      300
## 6     71    190  large   132    86    NA    NA    36  42      195
```

```
summary(DiabetesWeight$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    99.0   151.0   172.5   177.6   200.0   325.0
```

Y efectivamente ya no aparece e dato NA. Si solo se quiere tomar la variable Weight:

```
DiabetesWeight2<-subset(DiabetesData$weight, (!is.na(DiabetesData$weight)))
head(DiabetesWeight2)
```

```
## [1] 121 218 256 119 183 190
```

```
summary(DiabetesWeight2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    99.0   151.0   172.5   177.6   200.0   325.0
```

Queda un vector unicamente con los datos

El comando table calcula las frecuencias absolutas de la variable de interÃ©s.

```
fabsol<-table(DiabetesData[,11])
```

El comando prop.table calcula las frecuencias relativas de la variable de interÃ©s y necesita las frecuencias absolutas:

```
frelativas<-round(prop.table(fabsol)*100,3)
```

El comando cumsum permite acumular cantidades, que en este caso son las frecuencias absolutas:

```
fabsAcum<-cumsum(fabsol)
```

Y para las relativas:

```
fRelatAcum<-cumsum(frelativas)
```

Las siguientes 3 lineas permiten conformar la tabla de frecuencias y nombrar las filas y columnas:

```
tabla<-cbind(fabsol,fabsAcum,frelativas,fRelatAcum)
colnames(tabla)<-c("n","N","f","F")
tabla  #Muestra la tabla de frecuencias resultante.
```

```
##      n   N     f       F
## 99   1   1 0.249   0.249
## 100  1   2 0.249   0.498
## 102  1   3 0.249   0.747
## 105  2   5 0.498   1.245
## 109  1   6 0.249   1.494
## 110  2   8 0.498   1.992
```

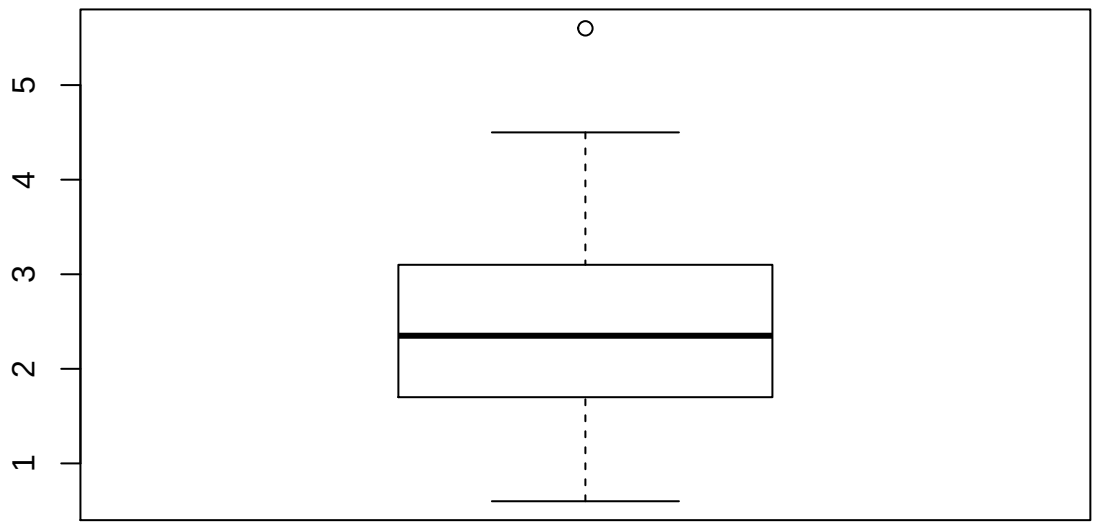
```
## 113  1   9 0.249   2.241
## 114  3  12 0.746   2.987
## 115  3  15 0.746   3.733
## 118  3  18 0.746   4.479
## 119  4  22 0.995   5.474
## 120  5  27 1.244   6.718
## 121  4  31 0.995   7.713
## 123  3  34 0.746   8.459
## 124  1  35 0.249   8.708
## 125  4  39 0.995   9.703
## 126  2  41 0.498  10.201
## 127  1  42 0.249  10.450
## 128  2  44 0.498  10.948
## 129  1  45 0.249  11.197
## 130  6  51 1.493  12.690
## 134  2  53 0.498  13.188
## 135  1  54 0.249  13.437
## 136  2  56 0.498  13.935
## 137  2  58 0.498  14.433
## 138  3  61 0.746  15.179
## 139  2  63 0.498  15.677
## 140  3  66 0.746  16.423
## 141  3  69 0.746  17.169
## 142  4  73 0.995  18.164
## 143  2  75 0.498  18.662
## 144  2  77 0.498  19.160
## 145 11  88 2.736  21.896
## 146  4  92 0.995  22.891
## 147  3  95 0.746  23.637
## 148  2  97 0.498  24.135
## 150  3 100 0.746  24.881
## 151  4 104 0.995  25.876
## 152  2 106 0.498  26.374
## 153  3 109 0.746  27.120
## 154  6 115 1.493  28.613
## 155  2 117 0.498  29.111
## 156  4 121 0.995  30.106
## 157  2 123 0.498  30.604
## 158  5 128 1.244  31.848
## 159  6 134 1.493  33.341
## 160  8 142 1.990  35.331
## 161  3 145 0.746  36.077
## 162  2 147 0.498  36.575
## 163  5 152 1.244  37.819
## 164  3 155 0.746  38.565
## 165 10 165 2.488  41.053
## 166  2 167 0.498  41.551
## 167  7 174 1.741  43.292
## 168  2 176 0.498  43.790
## 169  6 182 1.493  45.283
## 170 14 196 3.483  48.766
## 171  2 198 0.498  49.264
## 172  3 201 0.746  50.010
## 173  2 203 0.498  50.508
```

```
## 174  6 209 1.493  52.001
## 175  2 211 0.498  52.499
## 176  1 212 0.249  52.748
## 177  2 214 0.498  53.246
## 178  2 216 0.498  53.744
## 179 10 226 2.488  56.232
## 180  9 235 2.239  58.471
## 181  5 240 1.244  59.715
## 182  2 242 0.498  60.213
## 183 10 252 2.488  62.701
## 184  2 254 0.498  63.199
## 185  6 260 1.493  64.692
## 186  5 265 1.244  65.936
## 187  5 270 1.244  67.180
## 188  2 272 0.498  67.678
## 189  5 277 1.244  68.922
## 190  4 281 0.995  69.917
## 191  4 285 0.995  70.912
## 192  2 287 0.498  71.410
## 195  2 289 0.498  71.908
## 196  3 292 0.746  72.654
## 197  3 295 0.746  73.400
## 198  3 298 0.746  74.146
## 199  1 299 0.249  74.395
## 200  7 306 1.741  76.136
## 201  2 308 0.498  76.634
## 202  2 310 0.498  77.132
## 203  1 311 0.249  77.381
## 204  4 315 0.995  78.376
## 205  4 319 0.995  79.371
## 209  2 321 0.498  79.869
## 210  7 328 1.741  81.610
## 211  2 330 0.498  82.108
## 212  3 333 0.746  82.854
## 214  2 335 0.498  83.352
## 215  3 338 0.746  84.098
## 216  2 340 0.498  84.596
## 217  1 341 0.249  84.845
## 218  2 343 0.498  85.343
## 219  2 345 0.498  85.841
## 220  4 349 0.995  86.836
## 222  2 351 0.498  87.334
## 223  3 354 0.746  88.080
## 224  1 355 0.249  88.329
## 225  2 357 0.498  88.827
## 227  3 360 0.746  89.573
## 228  1 361 0.249  89.822
## 230  2 363 0.498  90.320
## 232  1 364 0.249  90.569
## 233  2 366 0.498  91.067
## 235  3 369 0.746  91.813
## 237  1 370 0.249  92.062
## 239  1 371 0.249  92.311
## 244  1 372 0.249  92.560
```

```
## 245  2 374 0.498  93.058
## 248  1 375 0.249  93.307
## 250  1 376 0.249  93.556
## 251  1 377 0.249  93.805
## 252  2 379 0.498  94.303
## 253  1 380 0.249  94.552
## 255  2 382 0.498  95.050
## 256  1 383 0.249  95.299
## 257  1 384 0.249  95.548
## 259  1 385 0.249  95.797
## 260  1 386 0.249  96.046
## 262  1 387 0.249  96.295
## 263  1 388 0.249  96.544
## 264  1 389 0.249  96.793
## 266  1 390 0.249  97.042
## 270  1 391 0.249  97.291
## 274  1 392 0.249  97.540
## 277  2 394 0.498  98.038
## 282  1 395 0.249  98.287
## 285  1 396 0.249  98.536
## 288  1 397 0.249  98.785
## 289  1 398 0.249  99.034
## 290  1 399 0.249  99.283
## 308  1 400 0.249  99.532
## 320  1 401 0.249  99.781
## 325  1 402 0.249 100.030
```

Ejercicio No. 3 del curso de Estadistica de www.seactuario.com

```
x<-c(0.6, 1.4, 1.6, 1.7, 1.9, 2.2, 2.0, 2.5, 2.5, 2.9, 3.1, 3.3, 4.5, 5.6)
boxplot(x,range=1.5)
```




```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.600   1.750   2.350   2.557   3.050   5.600
```

Ejemplo Diagrama de Tallo y Hojas

```
datospesos<-c(45,48,50,45,58,65,68,70,72,75,70,65,60,62,60,70,75,80,78,80,85,85,90,75)
stem(datospesos)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   4 | 558
##   5 | 08
##   6 | 002558
##   7 | 00025558
##   8 | 0055
##   9 | 0
```