Ejercicios de Gráficos en R

David Criado Ramón

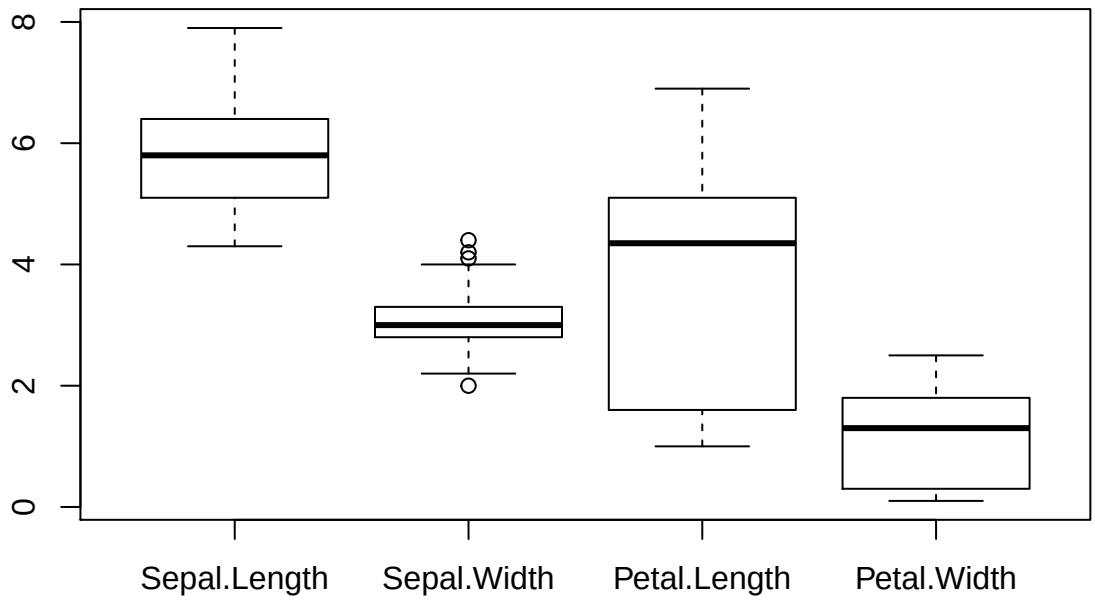
07/11/2019

1. Mostrar gráficamente la información correspondiente a summary(iris[1:4]). Pista: uso de boxplot.

```
summary(iris[1:4])
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

```
boxplot(iris[1:4], outlines=F, axis=F)
```

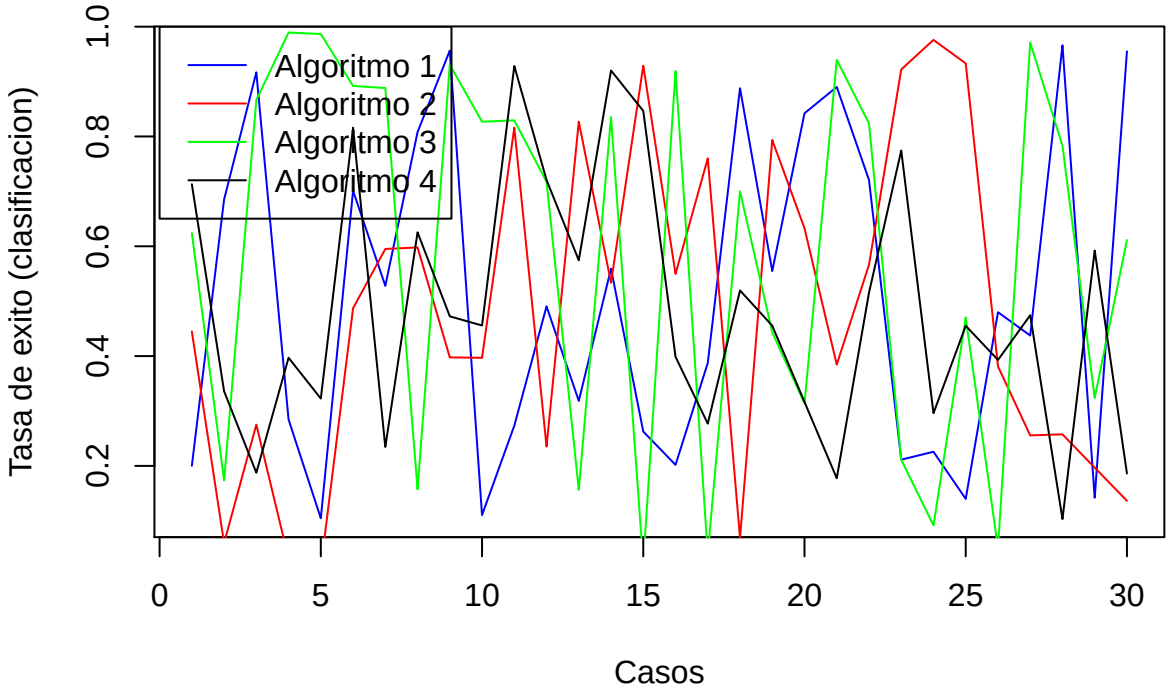


2. Rellenar una matriz nrow = 30, ncol = 4, con números aleatorios cada columna, supongamos que son las tasas de éxito en clasificación correspondiente a 4 algoritmos distintos. Pintar una gráfica de curvas de cada, identificando cada uno de los algoritmos con su leyenda.

```
matriz <- matrix(runif(120), nrow=30, ncol=4)
matriz
```

```
##            [,1]       [,2]       [,3]      [,4]
##  [1,] 0.2002145 0.44473056 0.62396650 0.7130181
##  [2,] 0.6852186 0.05936247 0.17374528 0.3359968
##  [3,] 0.9168758 0.27514918 0.86643458 0.1875372
##  [4,] 0.2843995 0.03114826 0.98933569 0.3969024
##  [5,] 0.1046501 0.01448234 0.98663576 0.3226703
##  [6,] 0.7010575 0.48716903 0.89201308 0.8159564
##  [7,] 0.5279600 0.59515162 0.88821761 0.2344320
##  [8,] 0.8079352 0.59785304 0.15734351 0.6254065
##  [9,] 0.9565001 0.39768051 0.93045947 0.4724108
## [10,] 0.1104530 0.39683040 0.82693025 0.4557002
## [11,] 0.2732849 0.81607607 0.82934342 0.9283003
## [12,] 0.4905132 0.23519875 0.71734030 0.7207331
## [13,] 0.3184040 0.82687025 0.15635962 0.5744543
## [14,] 0.5591728 0.53347593 0.83546141 0.9202377
## [15,] 0.2625931 0.92894271 0.02272462 0.8466150
## [16,] 0.2018752 0.54979473 0.91912709 0.3985657
## [17,] 0.3875257 0.76014253 0.03901895 0.2769806
## [18,] 0.8878698 0.06850976 0.70011959 0.5195266
## [19,] 0.5549226 0.79360934 0.44329984 0.4553322
## [20,] 0.8421794 0.63280343 0.31386845 0.3164831
## [21,] 0.8902071 0.38457694 0.93943648 0.1774952
## [22,] 0.7207010 0.56627268 0.82374548 0.5154622
## [23,] 0.2113403 0.92199057 0.21115411 0.7743666
## [24,] 0.2257173 0.97587761 0.09162944 0.2959247
## [25,] 0.1399837 0.93303384 0.47027488 0.4551702
## [26,] 0.4799139 0.38116266 0.05006142 0.3929327
## [27,] 0.4374119 0.25556407 0.97099802 0.4744489
## [28,] 0.9659641 0.25748370 0.78315554 0.1031385
## [29,] 0.1419139 0.19688638 0.32399045 0.5922262
## [30,] 0.9549789 0.13640090 0.61148645 0.1861557
```

```
plot(1:30,matriz[,1], type="l", col="blue", xlab="Casos",
     ylab="Tasa de exito (clasificacion)")
lines(1:30, matriz[,2], col="red")
lines(1:30, matriz[,3], col="green")
lines(1:30, matriz[,4], col="black")
legend(x=0, y = 1, legend=c("Algoritmo 1", "Algoritmo 2", "Algoritmo 3",
                            "Algoritmo 4"), col=c("blue", "red", "green", "black"), lty=1)
```

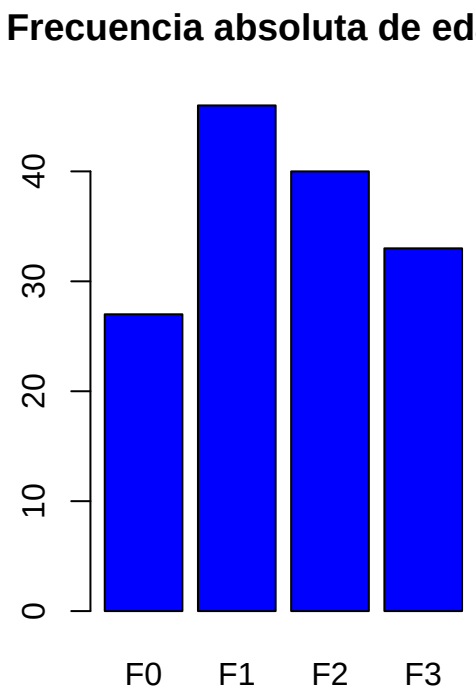


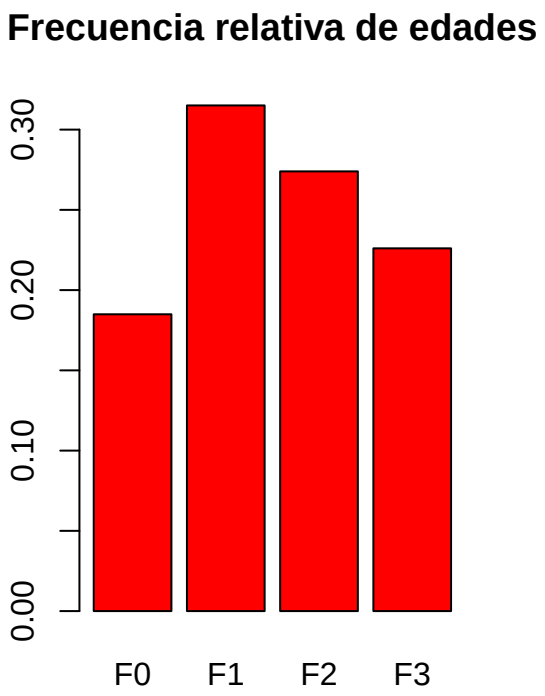
3. Ejecuta las siguientes instrucciones: library(MASS); str(quine); xtabs(~ Age, data=quine); prop.table(xtabs(~ Age, data=quine)). Haz un gráfico compuesto, con dos gráficas de barras correspondientes a xtabs y props.table, la frecuencia absoluta y la frecuencia relativa de las edades.

```
library(MASS)
str(quine)
```

```
## 'data.frame':    146 obs. of  5 variables:
##  $ Eth : Factor w/ 2 levels "A","N": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age : Factor w/ 4 levels "F0","F1","F2",..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ Lrn : Factor w/ 2 levels "AL","SL": 2 2 2 1 1 1 1 1 2 2 ...
##  $ Days: int  2 11 14 5 5 13 20 22 6 6 ...
```

```
par(mfrow=c(1,2))
x <- xtabs(~ Age, data=quine)
y <- prop.table(xtabs(~ Age, data=quine))
barplot(x, main="Frecuencia absoluta de edades", col=c("blue"))
barplot(y, main="Frecuencia relativa de edades", col=c("red"))
```

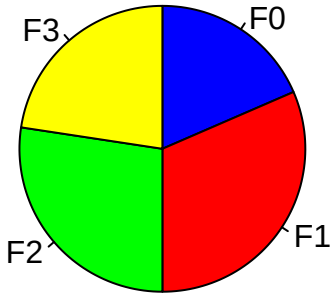




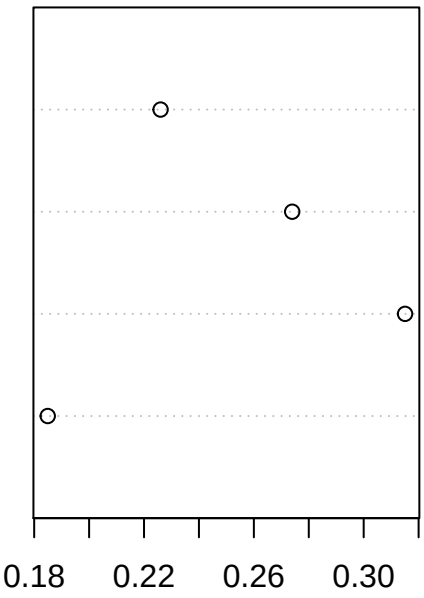
4. Representa la misma información anterior mediante gráficas de tipo pie y dotchart con título. En pie, fija colores y sentido horario.

```
par(mfrow=c(1,2))
pie(x, clockwise=T, col=c("blue", "red", "green", "yellow"),
    main="Frecuencia absoluta de edades")
dotchart(as.numeric(y), main="Frecuencia relativa de edades")
```

Frecuencia absoluta de edades

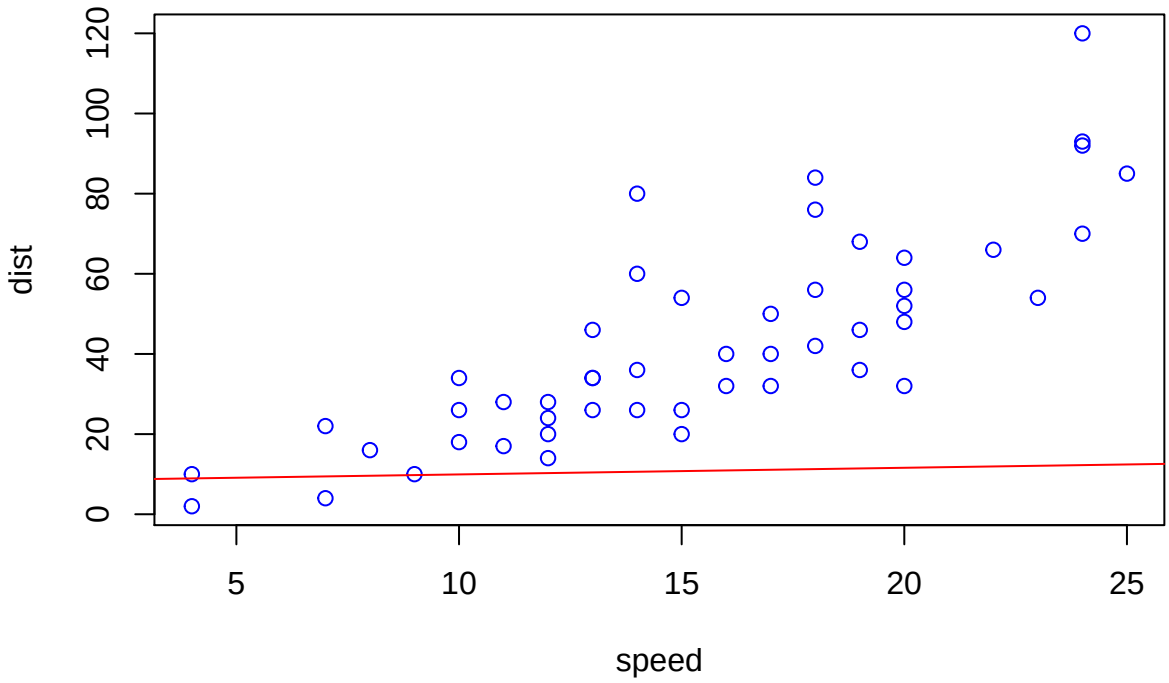


Frecuencia relativa de edades



5. Sea un dataset cars, representar los puntos dist vs speed, esto es, el atributo dist en ordenadas. Sea m, m = lm(speed~dist, data=cars), el resultado de aplicar un ajuste mediante regresión lineal. El valor resultado es una recta en forma. Pinta la línea de ajuste del modelo m, en rojo. Pista: abline.

```
m = lm(speed~dist, data=cars)
plot(cars, xlab="speed", ylab="dist", col="blue")
abline(m, col="red")
```


dist
0
20
40
60
80
100
120
5
10
15
20
25
speed