```r
fichero = read.csv("distancia_universitarios.csv")
fichero
```

```
##    Distancia
## 1       16.5
## 2       34.8
## 3       20.7
## 4        6.2
## 5        4.4
## 6        3.4
## 7       24.0
## 8       24.0
## 9       32.0
## 10      30.0
## 11      33.0
## 12      27.0
## 13      15.0
## 14       9.4
## 15       2.1
## 16      34.0
## 17      24.0
## 18      12.0
## 19       4.4
## 20      28.0
## 21      31.4
## 22      21.6
## 23       3.1
## 24       4.5
## 25       5.1
## 26       4.0
## 27       3.2
## 28      25.0
## 29       4.5
## 30      20.0
## 31      34.0
## 32      12.0
## 33      12.0
## 34      12.0
## 35      12.0
## 36       5.0
## 37      19.0
## 38      30.0
## 39       5.5
## 40      38.0
## 41      25.0
```

```
## 42       3.7
## 43       9.0
## 44      30.0
## 45      13.0
## 46      30.0
## 47      30.0
## 48      26.0
## 49      30.0
## 50      30.0
## 51       1.0
## 52      26.0
## 53      22.0
## 54      10.0
## 55       9.7
## 56      11.0
## 57      24.1
## 58      33.0
## 59      17.2
## 60      27.0
## 61      24.0
## 62      27.0
## 63      21.0
## 64      28.0
## 65      30.0
## 66       4.0
## 67      46.0
## 68      29.0
## 69       3.7
## 70       2.7
## 71       8.1
## 72      19.0
## 73      16.0
```

```
len = function(list){
        count = 0
        for (element in list){
                count = count + 1
        }
        count
}

distancias = fichero$Distancia

longitud = len(distancias)
longitud
```

```
## [1] 73
```

```
bubble = function(list, asc = TRUE){
        n = len(list)
        if(asc){
                for (i in 2:n){
                        for (j in 1:(n-1)){
                                if (list[j] > list[j+1]){
                                        temp = list[j]
                                        list[j] = list[j+1]
                                        list[j+1] = temp
                                }
                        }
                }
        }
        else {
                for (i in 2:n){
                        for (j in 1:(n-1)){
                                if (list[j] < list[j+1]){
                                        temp = list[j]
                                        list[j] = list[j+1]
                                        list[j+1] = temp
                                }
                        }
                }
        }
        list
}
distanciasordenadas = bubble(distancias, FALSE)
distanciasordenadas
```

```
##  [1] 46.0 38.0 34.8 34.0 34.0 33.0 33.0 32.0 31.4 30.0 30.0 30.0 30.0 30.0 30.0
## [16] 30.0 30.0 29.0 28.0 28.0 27.0 27.0 27.0 26.0 26.0 25.0 25.0 24.1 24.0 24.0
## [31] 24.0 24.0 22.0 21.6 21.0 20.7 20.0 19.0 19.0 17.2 16.5 16.0 15.0 13.0 12.0
## [46] 12.0 12.0 12.0 12.0 11.0 10.0  9.7  9.4  9.0  8.1  6.2  5.5  5.1  5.0  4.5
## [61]  4.5  4.4  4.4  4.0  4.0  3.7  3.7  3.4  3.2  3.1  2.7  2.1  1.0
```

```
rank = function(list){
        ordered_list = bubble(list)
        ordered_list[len(ordered_list)] - ordered_list[1]
}

rango = rank(distanciasordenadas)
rango
```

```
## [1] 45
```

```
absolute_freq = function(list){
        ordered_list = bubble(list)
        n = len(ordered_list)
        elements = vector()
        frequencies = vector()
        i = 1
        while (i <= n){
                actual_element = ordered_list[i]
                elements = append(elements, actual_element)
                actual_freq = 0
                j = i
                while(j <= n & actual_element == ordered_list[j]){
                        actual_freq = actual_freq + 1
                        j = j+1
                }
                frequencies = append(frequencies, actual_freq)
                i = j
        }
        rbind(elements, frequencies)
}
frecuencia_abs = absolute_freq(distancias)
frecuencia_abs
```

```
##             [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## elements       1  2.1  2.7  3.1  3.2  3.4  3.7    4  4.4   4.5     5   5.1
## frequencies    1  1.0  1.0  1.0  1.0  1.0  2.0    2  2.0   2.0     1   1.0
##             [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
## elements      5.5   6.2   8.1     9   9.4   9.7    10    11    12    13    15
## frequencies   1.0   1.0   1.0     1   1.0   1.0     1     1     5     1     1
##             [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34]
## elements       16  16.5  17.2    19    20  20.7    21  21.6    22    24  24.1
## frequencies     1   1.0   1.0     2     1   1.0     1   1.0     1     4   1.0
##             [,35] [,36] [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45]
## elements       25    26    27    28    29    30  31.4    32    33    34  34.8
## frequencies     2     2     3     2     1     8   1.0     1     2     2   1.0
##             [,46] [,47]
## elements       38    46
## frequencies     1     1
```

```
relative_freq = function(list){
        f_abs = absolute_freq(list)
        elements = f_abs[1,]
        abs_fvalues = f_abs[2,]
        rbind(elements,abs_fvalues/len(list))
}
```

```
frecuencia_rel = relative_freq(distancias)
frecuencia_rel
```

```
##                [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## elements 1.00000000 2.10000000 2.70000000 3.10000000 3.20000000 3.40000000
##          0.01369863 0.01369863 0.01369863 0.01369863 0.01369863 0.01369863
##                [,7]       [,8]       [,9]      [,10]      [,11]      [,12]
## elements 3.70000000 4.00000000 4.40000000 4.50000000 5.00000000 5.10000000
##          0.02739726 0.02739726 0.02739726 0.02739726 0.01369863 0.01369863
##               [,13]      [,14]      [,15]      [,16]      [,17]      [,18]
## elements 5.50000000 6.20000000 8.10000000 9.00000000 9.40000000 9.70000000
##          0.01369863 0.01369863 0.01369863 0.01369863 0.01369863 0.01369863
##                [,19]       [,20]       [,21]       [,22]       [,23]
## elements 10.00000000 11.00000000 12.00000000 13.00000000 15.00000000
##           0.01369863  0.01369863  0.06849315  0.01369863  0.01369863
##                [,24]       [,25]       [,26]       [,27]       [,28]
## elements 16.00000000 16.50000000 17.20000000 19.00000000 20.00000000
##           0.01369863  0.01369863  0.01369863  0.02739726  0.01369863
##                [,29]       [,30]       [,31]       [,32]       [,33]
## elements 20.70000000 21.00000000 21.60000000 22.00000000 24.00000000
##           0.01369863  0.01369863  0.01369863  0.01369863  0.05479452
##                [,34]       [,35]       [,36]       [,37]       [,38]
## elements 24.10000000 25.00000000 26.00000000 27.00000000 28.00000000
##           0.01369863  0.02739726  0.02739726  0.04109589  0.02739726
##                [,39]     [,40]       [,41]       [,42]       [,43]       [,44]
## elements 29.00000000 30.000000 31.40000000 32.00000000 33.00000000 34.00000000
##           0.01369863  0.109589  0.01369863  0.01369863  0.02739726  0.02739726
##                [,45]       [,46]       [,47]
## elements 34.80000000 38.00000000 46.00000000
##           0.01369863  0.01369863  0.01369863
```

```
acum_absolute_freq = function(list){
        f_abs = absolute_freq(list)
        elements = f_abs[1,]
        abs_fvalues = f_abs[2,]
        acum_abs_fvalues = vector()
        acum = 0
        for (i in 1:len(elements)){
                acum = acum + abs_fvalues[i]
                acum_abs_fvalues = append(acum_abs_fvalues, acum)
        }
        rbind(elements, acum_abs_fvalues)
}
frecuencia_abs_acum = acum_absolute_freq(distancias)
frecuencia_abs_acum
```

```
##                  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## elements            1  2.1  2.7  3.1  3.2  3.4  3.7    4  4.4   4.5     5   5.1
## acum_abs_fvalues    1  2.0  3.0  4.0  5.0  6.0  8.0   10 12.0  14.0    15  16.0
##                  [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## elements           5.5   6.2   8.1     9   9.4   9.7    10    11    12    13
## acum_abs_fvalues  17.0  18.0  19.0    20  21.0  22.0    23    24    29    30
##                  [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32]
## elements            15    16  16.5  17.2    19    20  20.7    21  21.6    22
## acum_abs_fvalues    31    32  33.0  34.0    36    37  38.0    39  40.0    41
##                  [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40] [,41] [,42]
## elements            24  24.1    25    26    27    28    29    30  31.4    32
## acum_abs_fvalues    45  46.0    48    50    53    55    56    64  65.0    66
##                  [,43] [,44] [,45] [,46] [,47]
## elements            33    34  34.8    38    46
## acum_abs_fvalues    68    70  71.0    72    73
```

```
acum_relative_freq = function(list){
        f_rel = relative_freq(list)
        elements = f_rel[1,]
        rel_fvalues = f_rel[2,]
        acum_rel_fvalues = vector()
        acum = 0
        for (i in 1:len(elements)){
                acum = acum + rel_fvalues[i]
                acum_rel_fvalues = append(acum_rel_fvalues, acum)
        }
        rbind(elements, acum_rel_fvalues)
}
frecuencia_rel_acum = acum_relative_freq(distancias)
frecuencia_rel_acum
```

```
##                        [,1]       [,2]       [,3]       [,4]       [,5]
## elements         1.00000000 2.10000000 2.70000000 3.10000000 3.20000000
## acum_rel_fvalues 0.01369863 0.02739726 0.04109589 0.05479452 0.06849315
##                        [,6]      [,7]       [,8]      [,9]      [,10]     [,11]
## elements         3.40000000 3.700000 4.0000000 4.4000000 4.5000000 5.0000000
## acum_rel_fvalues 0.08219178 0.109589 0.1369863 0.1643836 0.1917808 0.2054795
##                       [,12]     [,13]     [,14]    [,15]     [,16]     [,17]
## elements         5.1000000 5.5000000 6.2000000 8.100000 9.0000000 9.4000000
## acum_rel_fvalues 0.2191781 0.2328767 0.2465753 0.260274 0.2739726 0.2876712
##                       [,18]      [,19]      [,20]      [,21]      [,22]
## elements         9.7000000 10.0000000 11.0000000 12.0000000 13.0000000
## acum_rel_fvalues 0.3013699  0.3150685  0.3287671  0.3972603  0.4109589
##                       [,23]      [,24]      [,25]      [,26]      [,27]
## elements         15.0000000 16.0000000 16.5000000 17.2000000 19.0000000
```

```
## acum_rel_fvalues  0.4246575  0.4383562  0.4520548  0.4657534  0.4931507
##                       [,28]      [,29]      [,30]      [,31]      [,32]
## elements         20.0000000 20.7000000 21.0000000 21.6000000 22.0000000
## acum_rel_fvalues  0.5068493  0.5205479  0.5342466  0.5479452  0.5616438
##                       [,33]      [,34]      [,35]      [,36]      [,37]
## elements         24.0000000 24.100000 25.0000000 26.0000000 27.0000000
## acum_rel_fvalues  0.6164384  0.630137  0.6575342  0.6849315  0.7260274
##                       [,38]      [,39]      [,40]      [,41]      [,42]
## elements         28.0000000 29.0000000 30.0000000 31.400000 32.0000000
## acum_rel_fvalues  0.7534247  0.7671233  0.8767123  0.890411  0.9041096
##                       [,43]      [,44]      [,45]      [,46] [,47]
## elements         33.0000000 34.0000000 34.8000000 38.0000000    46
## acum_rel_fvalues  0.9315068  0.9589041  0.9726027  0.9863014     1
```

```
mean = function(list){
        total = 0
        n = len(list)
        for (i in 1:n){
                total = total + list[i]
        }
        mean = total / n
        mean
}

media = mean(distancias)
media
```

```
## [1] 18.53425
```

```
mode = function(list){
        frequencies = absolute_freq(list)
        elements = frequencies[1,]
        freq_values = frequencies[2,]
        actual_mode = 0
        actual_mode_val = 0
        for (i in 1:len(elements)){
                if (freq_values[i] > actual_mode_val){
                        actual_mode_val = freq_values[i]
                        actual_mode = elements[i]
                }
        }
        actual_mode
}

moda = mode(distancias)
moda
```

```
## [1] 30

median = function(list){
        n = len(list)
        ordered_list = bubble(list)
        if (n%%2 == 0){
                median = (ordered_list[n/2] + ordered_list[(n/2)+1]) / 2
        }
        else{
                median = ordered_list[(n+1)/2]
        }
        median
}

mediana = median(distancias)
mediana

## [1] 20

standard_desv = function(list){
        mean = mean(list)
        n = len(list)
        add = 0
        for (i in 1:n){
                add = add + ((list[i] - mean)^2)
        }
        sqrt(add/n)
}

desviacion = standard_desv(distancias)
desviacion

## [1] 11.23204

variance = function(list){
        desv = standard_desv(list)
        var = desv^2
        var
}

varianza = variance(distancias)
varianza

## [1] 126.1587
```

```
quant = function(list, c){
        ordered_list = bubble(list)
        n = len(list)
        if (c < 0){
                quant = NULL
        }
        else{
                if((n*c)%%1 == 0){
                        quant = (ordered_list[(n*c)] + ordered_list[(n*c) + 1]) / 2
                }
                else {
                        int_prod = floor(n*c)
                        quant = ordered_list[int_prod + 1]
                }
        }
        quant
}

cuartil1 = quant(distancias,0.25)
cuartil2 = quant(distancias,0.5)
cuartil3 = quant(distancias,0.75)
cuartiln = quant(distancias, -0.2)
cuartil1

## [1] 8.1

cuartil2

## [1] 20

cuartil3

## [1] 28

cuartiln

## NULL
```