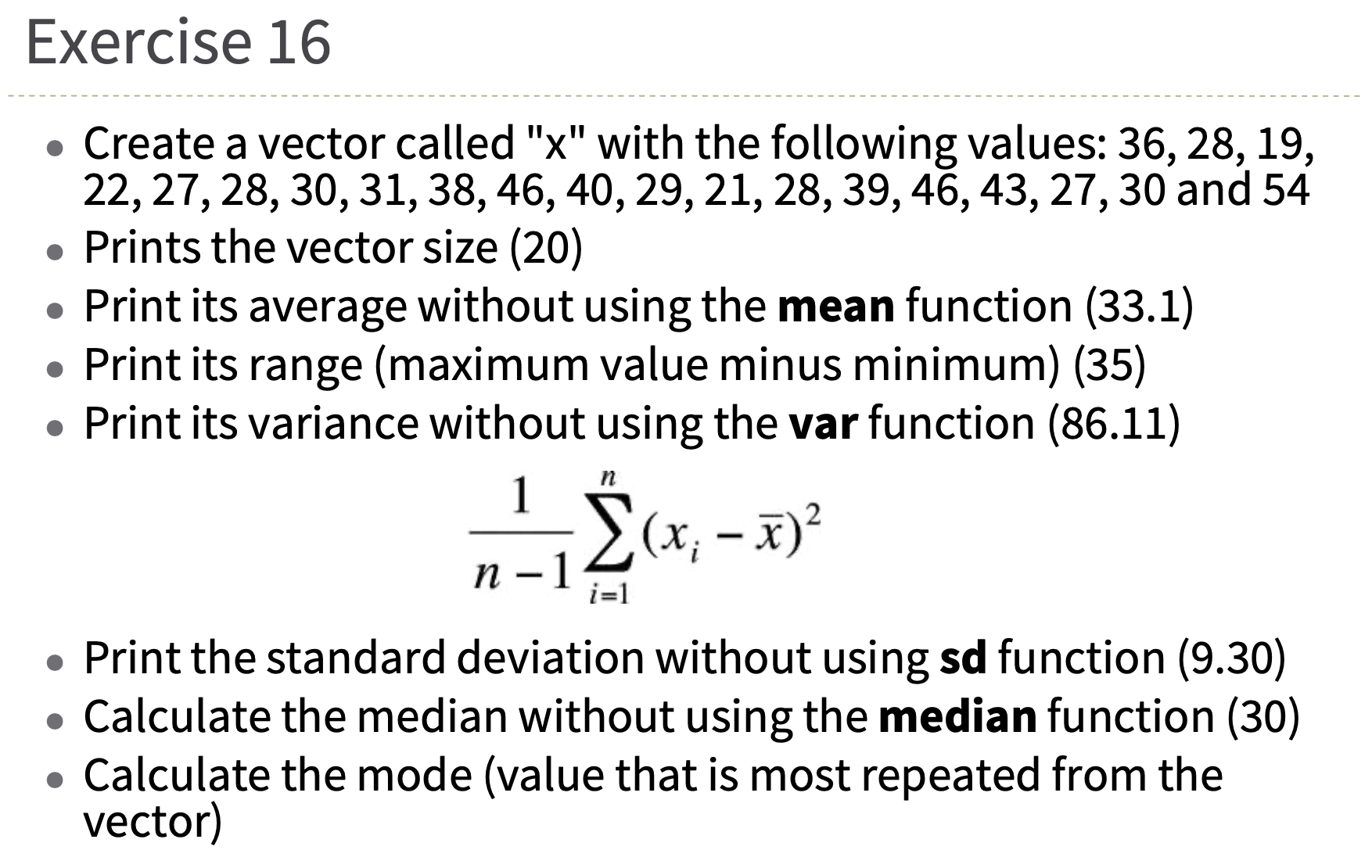
R Notebook

Juan de Toro

# Introduction to R\_Exercises

## Exercise 16



Exercise 16

* Create a vector called “x”:

x <- c(36, 28, 19, 22, 27, 28, 30, 31, 38, 46, 40, 29, 21, 28, 39, 46, 43, 27, 30, 54)

* Prints the vector size:

print(length(x))

## [1] 20

* AVG w/out mean function:

m <- sum(x)/length(x)  
print(m)

## [1] 33.1

* Variance w/out variance function:

v <- (1/(length(x)-1))\*sum((x-m)\*\*2)  
print(v)

## [1] 86.51579

var(x)

## [1] 86.51579

I have used the variance function to check that in the statement of the exercise there is an error in the result. CQD

* Standard Deviation using the function:

sd <- sd(x)  
print(sd)

## [1] 9.301386

* Median w/out median function:

We need sort the sample, ask for the lenght to know if it is odd or even, and at the end calculate the median

x\_sort <- sort(x)  
  
if(length(x)%%2 == 0){  
 x1 <- x\_sort[((length(x)+1)/2)-.5]  
 x2 <- x\_sort[((length(x)+1)/2)+.5]  
 med <- (x1+x2)/2  
}else{  
 med <- x\_sort[(length(x)+1)/2]  
}  
  
print(med)

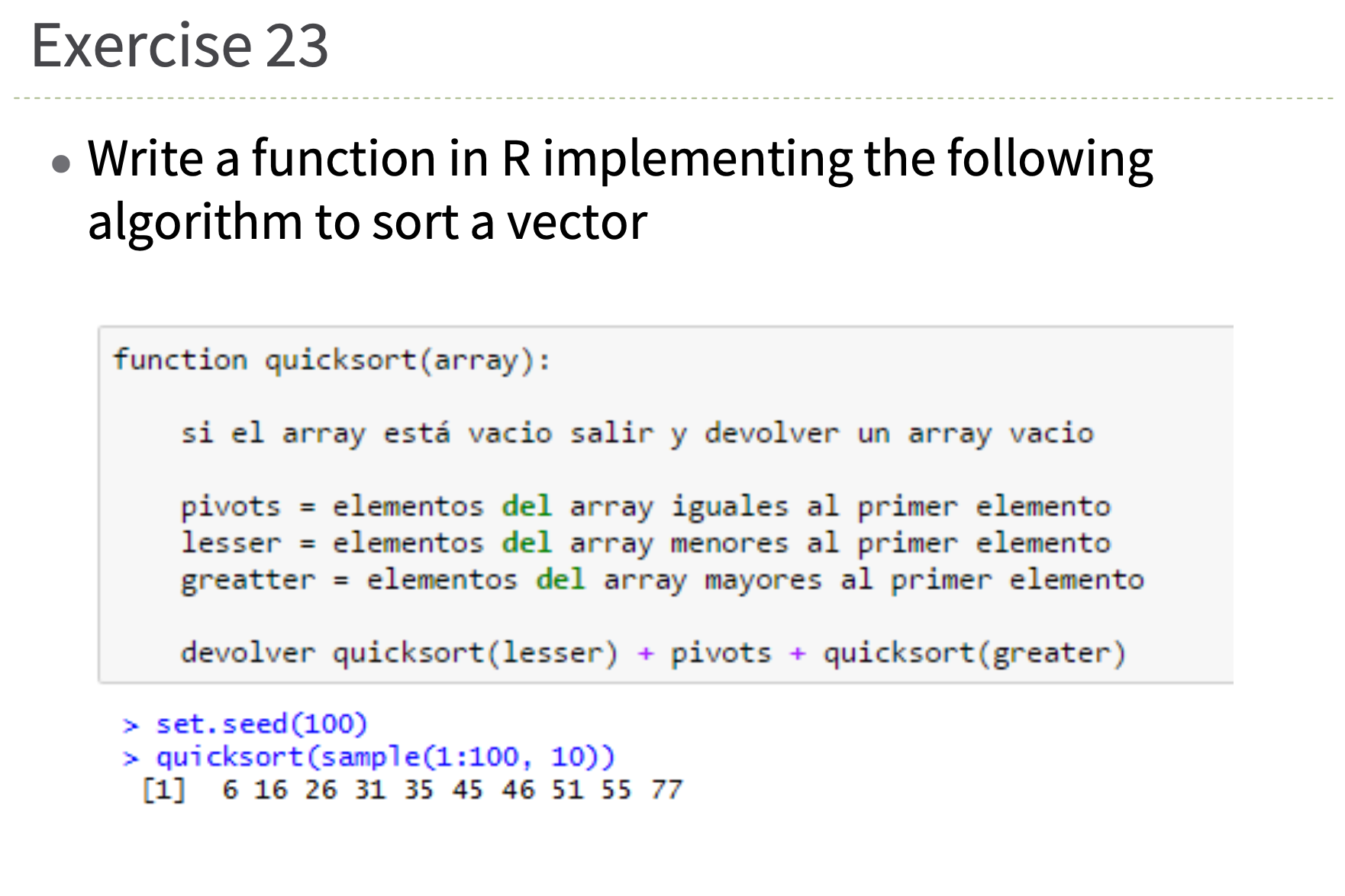
## [1] 30

* Calculate the mode:

uniqv <- unique(x)  
  
# I need to know the position of the first appearance of each element in order to make a count.   
match <- match(x, uniqv)  
  
# Now I count the number of occurrences of each value, even though they are positions, it doesn't matter, because they refer to a unique element due to the use of uniqv  
tab <- tabulate(match)  
  
# Now I want to know what is the maximum of the entire vector  
pos <- which.max(tab)  
  
# The previous number tells me the position of the element in the initial vector, so I have to paint the value of that element in its first occurrence  
print(uniqv[pos])

## [1] 28

## Exercise 23



Exercise 23

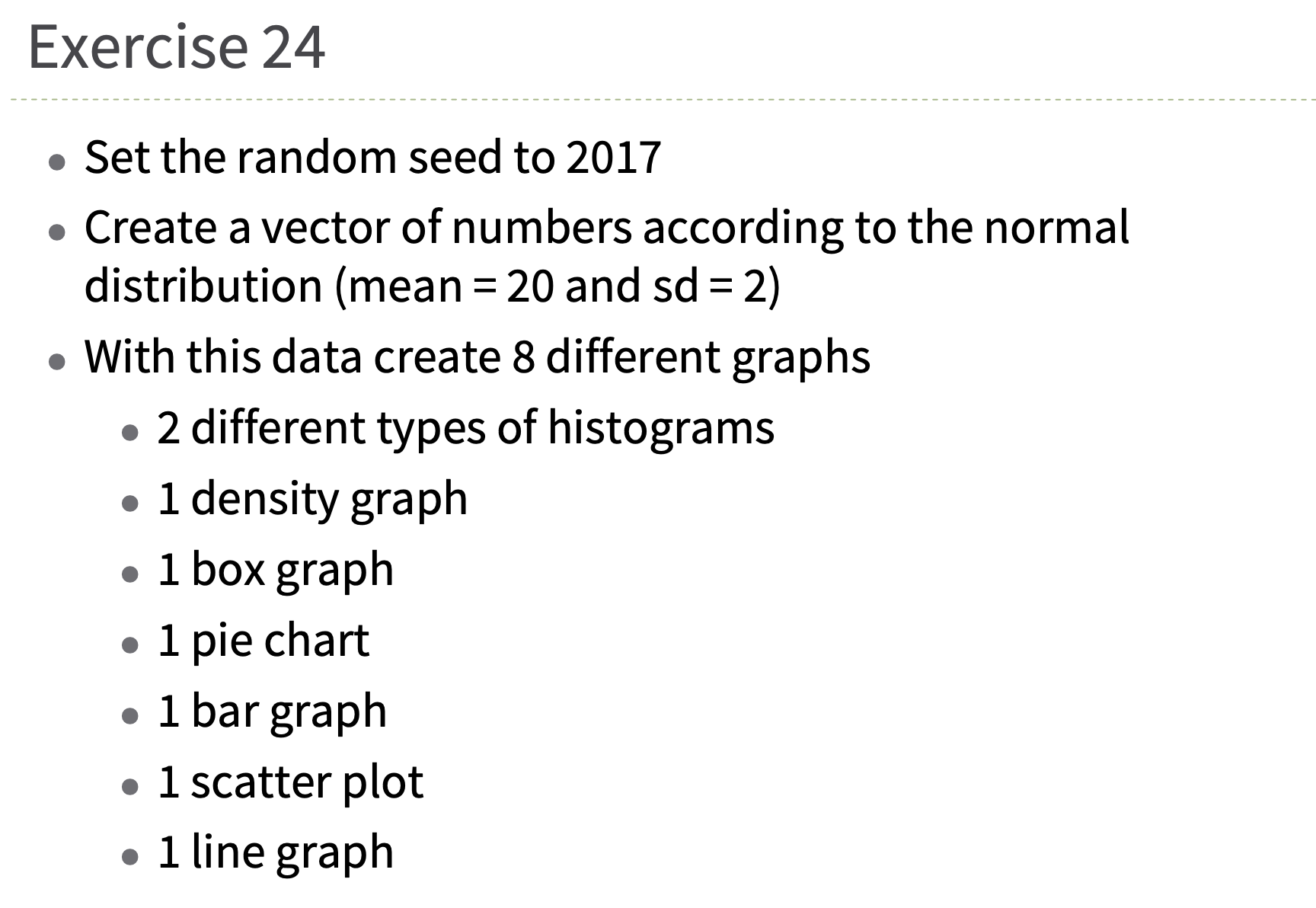
quicksort <- function (v){  
 if(length(v)==0){  
 print("Empty array")  
 close()  
 }else{  
 pivots <- v[match(v[1], v)]  
 lesser <- v[which(as.vector(v) < v[1])]  
 greatter <- v[which(as.vector(v) > v[1])]  
 result <- c(sort(lesser), sort(pivots), sort(greatter))  
 return(result)  
 }  
}

Try the function:

set.seed(100)  
quicksort(sample(1:100, 10))

## [1] 4 7 23 55 70 74 78 86 89 95

## Exercise 24



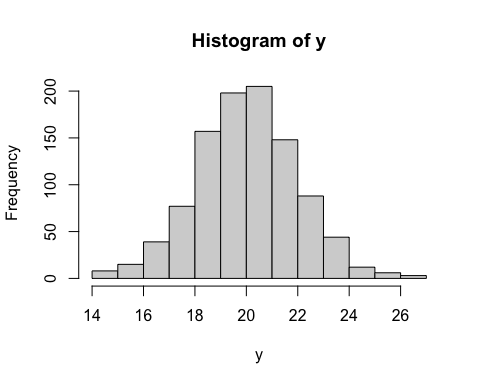
Exercise 24

* Seed in 2017:

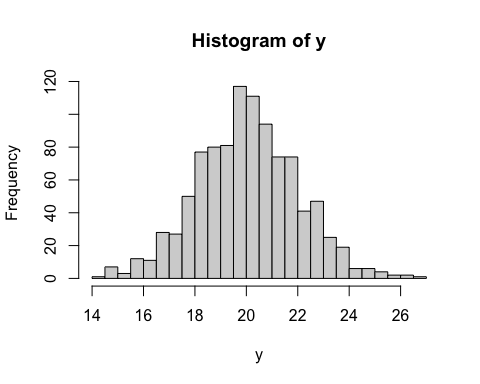
set.seed(2017)  
y <- rnorm(1000, mean=20, sd=2)

* 2 Histograms:

hist(y)

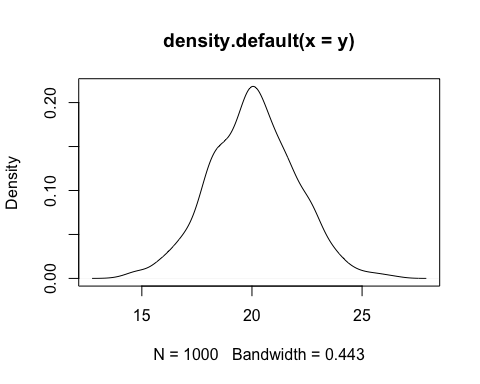


hist(y, breaks=20)



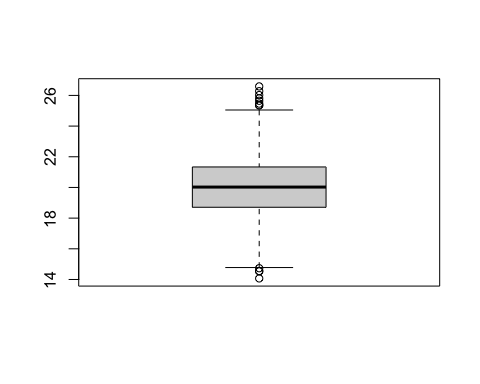
* Density:

plot(density(y))



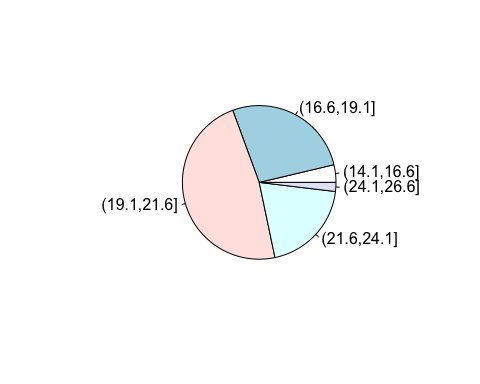
* Box:

boxplot(y)



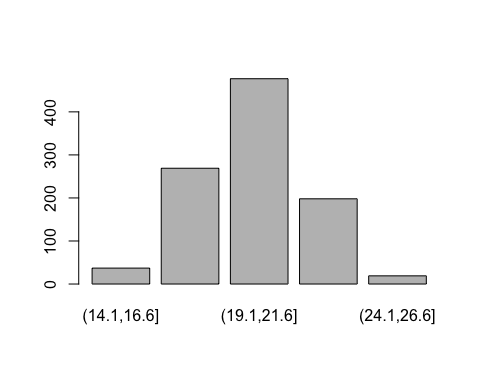
* Pie:

y\_c <- cut(y, breaks=5)  
pie(table(y\_c))



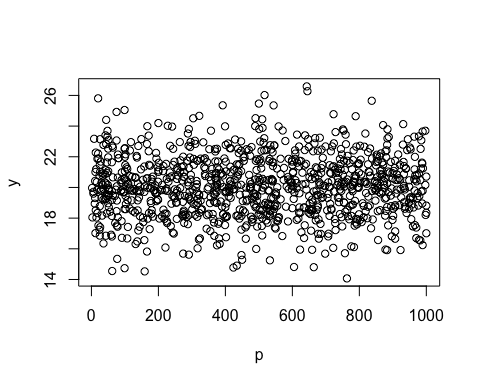
* Bar:

barplot(table(y\_c))



* Scatter:

p <- sample(1:1000, 1000, replace=T)  
plot(p,y)



* Line:

plot(sort(y), type = "l")

