Five Number Summary

JuJu

For this example we will use the ***HTWT*** data. In particular, we will look at the weights (***WEIGHT***) of the 20 individuals in the data set.

htwt = read.csv("http://facweb1.redlands.edu/fac/jim\_bentley/downloads/math111/htwt.csv")  
 head(htwt)

## Height Weight Group  
## 1 64 159 1  
## 2 63 155 2  
## 3 67 157 2  
## 4 60 125 1  
## 5 52 103 2  
## 6 58 122 2

## The Min and Max

The raw weight data for the sample data set can be seen by asking for the proper column. Three equivalent ways of accessing the second, ***Weight*** column are given below.

htwt$Weight

## [1] 159 155 157 125 103 122 101 82 228 199 195 110 191 151 119 119 112 87 190  
## [20] 87

htwt[,"Weight"]

## [1] 159 155 157 125 103 122 101 82 228 199 195 110 191 151 119 119 112 87 190  
## [20] 87

htwt[,2]

## [1] 159 155 157 125 103 122 101 82 228 199 195 110 191 151 119 119 112 87 190  
## [20] 87

To find the minimum and maximum, we first sort the data. The min and the max are the first and last observations respectively. The internal functions ***min*** and ***max*** can be used as well. Finally, the function ***range*** finds both the minimum and maximum.

n = length(htwt$Weight)  
 n

## [1] 20

wt.sorted = sort(htwt$Weight)  
 wt.sorted

## [1] 82 87 87 101 103 110 112 119 119 122 125 151 155 157 159 190 191 195 199  
## [20] 228

wt.min = wt.sorted[1]  
 wt.min

## [1] 82

wt.max = wt.sorted[n]  
 wt.max

## [1] 228

min(htwt$Weight)

## [1] 82

max(htwt$Weight)

## [1] 228

range(htwt$Weight)

## [1] 82 228

## The Median

The median is a value that divides the sorted data in half. If there are an even number of observations, the median is not unique. Convention dictates that in the case of an even number of observations we take the average of the two middle values.

If there are an odd number of observations we use the (n+1)/2 observation in the sorted data. If the number of observations is even we average the (n/2) and (n/2)+1 observations. Since the ***Weight*** data has 20 observations, we will use the second method.

R has the built-in function ***median*** that also provides the median.

# Create a function to figure out if an integer n is odd  
 odd = function(i){  
 # If i is not an integer then stop  
 if (!is.integer(i)){stop("Not an integer.")}  
 # Otherwise, if the division by 2 has no remainder, then i is even  
 else if (i %% 2 == 0) {return(FALSE)} else {return(TRUE)}  
 }  
 n = length(htwt$Weight)  
 odd(n)

## [1] FALSE

n.low = n/2  
 n.low

## [1] 10

n.high = (n/2)+1  
 n.high

## [1] 11

wt.sorted = sort(htwt$Weight)  
 wt.sorted

## [1] 82 87 87 101 103 110 112 119 119 122 125 151 155 157 159 190 191 195 199  
## [20] 228

wt.sorted[n.low]

## [1] 122

wt.sorted[n.high]

## [1] 125

wt.median = (wt.sorted[n.low]+wt.sorted[n.high])/2  
 wt.median

## [1] 123.5

median(htwt$Weight)

## [1] 123.5

## The Upper and Lower Quartiles

The lower and upper quartiles, Q1 and Q3, are the 25th and 75th percentiles respectively. These values are not necessarily unique. One way to find the values is to find the medians of the sorted observations that are less and greater than the median.

R has a function ***quantile*** that returns the requested quantiles (percentiles/100).

To compute the lower quartile we first note that n=20 is even. So the median splits the data into two groups of 10 observations. Since 10 is even, we must average the 5th and 6th observations. Similarly, the upper quartile is the average of the 15th and 16th observations.

n

## [1] 20

wt.q1 = (wt.sorted[5]+wt.sorted[6])/2  
 wt.q1

## [1] 106.5

wt.q3 = (wt.sorted[15]+wt.sorted[16])/2  
 wt.q3

## [1] 174.5

quantile(htwt$Weight, c(.25, .75))

## 25% 75%   
## 108.25 166.75

# Note that q1 and the 25 percentile are both between the 5th and 6th obs  
 wt.sorted[c(5,6)]

## [1] 103 110

# Note that q3 and the 75 percentile are both between the 15th and 16th obs  
 wt.sorted[c(15,16)]

## [1] 159 190

## The Five Number Summary

The five number summary is composed of the minimum, Q1, median, Q3, and the maximum. We can find these numbers as above or use the ***quantile*** funtion. The function ***summary*** can also be used. It should be noted that the ***summary*** function includes the mean.

c(wt.min, wt.q1, wt.median, wt.q3, wt.max)

## [1] 82.0 106.5 123.5 174.5 228.0

quantile(htwt$Weight)

## 0% 25% 50% 75% 100%   
## 82.00 108.25 123.50 166.75 228.00

summary(htwt$Weight)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82.0 108.2 123.5 139.6 166.8 228.0