iqr.R

davidharvey

Fri Aug 12 08:09:18 2016

```
# format is name = function(arguments passed to the code) {code}

iqr = function(x) {

# x is a vector that contains our data; to work with it we need to create a new
# vector that is sorted from smallest-to-largest value, which we accomplish
# using sort()

  x_sorted = sort(x)

# we also need the vector's length, which we accomplish using length()

  x_length = length(x)

# how we find the median depends on whether the vector has an even or an odd
# number of elements; the modulus operator (%%) allows us to test if division
# by 2 has a remainder of zero; if it does, then we know the vector has an even
# number of elements and use the commands within the IF statement to divide the
# sorted vector into its lower and upper halves; if the vector has an odd number
# of elements, then we use the commands wihtin the ELSE statement to divide it
# into halves

  if(x_length %% 2 == 0) {
   mid = x_length/2
   lower = x_sorted[1:mid]
   upper = x_sorted[(mid + 1):x_length]
  } else {
   mid = (x_length/2) + 0.5
   lower = x_sorted[1:(mid - 1)]
   upper = x_sorted[(mid + 1):x_length]
  }

# now we find the median for each half and then calculate the iqr

  f.lower = median(lower)
  f.upper = median(upper)
  iqr = f.upper - f.lower

# the print() command returns iqr to the console, but we also create a list of
# values to return if the function is assigned to an object

  print(iqr)
  out = list("iqr" = iqr, "F_U" = f.upper, "F_L" = f.lower)
  invisible(out)
}
```