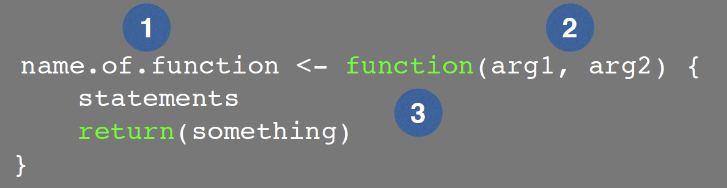
* Data import
* Read.table() and friends for flat files
  + The read.table() function is the base of all flat file import functions:
    - Read.delim(“filename.txt”, sep = “\t”) TAB
    - Read.csv(“filename.txt”, sep = “,”) comma
    - Read.csv2(“filename.txt”, sep = “;”) semi-colon
    - Read.table(“filename.txt”, sep = “”) white space
  + What other differences are there between these functions?
    - MS EXCEL file import options include:
      * Export your excel data to plain text CSV format
      * Or if you must, use readxl::read\_excel() to read specified parts of your sheets
    - For fast and convenient reading of very large flat files
      * Try data.table::fread() use is similar to read.table() but is automatically finds field separators and header rows. It is also much faster!
    - Saving and loading .RData files
      * Use the functions save() and load() for saving and loading multiple objects to space efficient binary format files
* Writing your own functions
  + What is a function
    - Name (can be almost anything you want)
    - Arguments (i.e. input to your function)
    - Body (where the work gets done)
  + Why would you write a function
    - When you find yourself doing the same thing 3 or more times it is time to write a function
    - consider the advantages
      * makes the purpose of the code more clear
      * reduce mistakes form copy/paste
      * makes updating your code easier
      * reduce duplication and facilitate re-use
  + how would you write a function
    - start with a working code snippet, simplify, reduce calculation duplication, finally turn it into a function
    - test, fail, change, test again
* side-note: seeing and using your function in RStudio
  + an easy way to visualize the code of a function is to type its name without the parentheses ()
  + if you have your new function saved to a separate file then you can load and execute it using the source() function. E.g. soure(“MyUtils.R”)
  + the return() statement is not required in a function but it is advisable to use it when the function performs several computations. It has the effect of ending the function execution and returning control to the code which called it