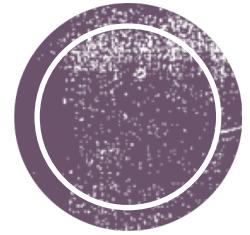


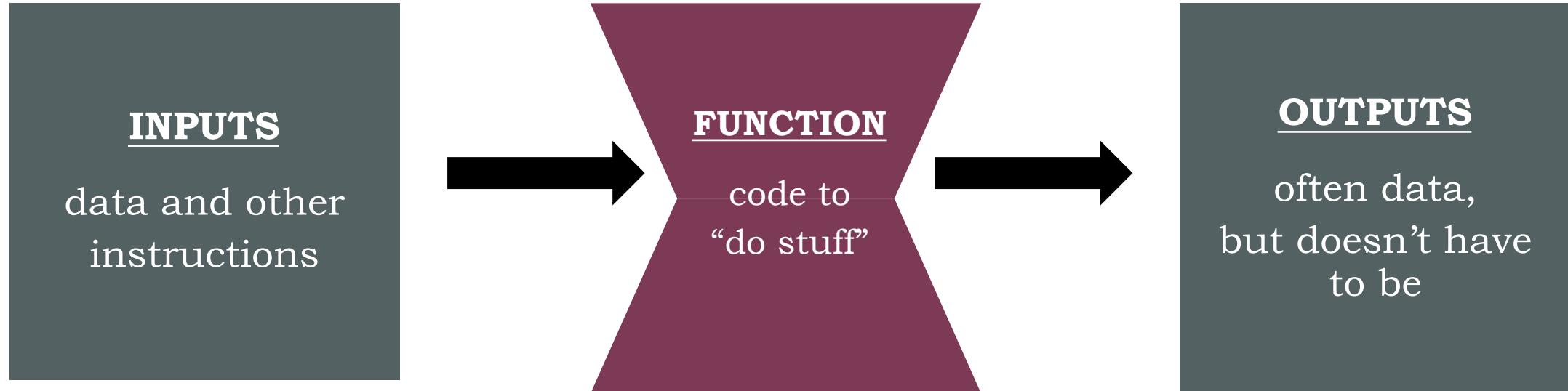
Writing Functions in R

BioFrass
January 29th, 2021



Download repository from GitHub

https://github.com/brittnibertolet/BioFrass2021_functions



What are functions?

- Tools that “do stuff”
- They take arguments in the form of data or instructions
- They return some output



mean(x)

What is the **mean** of this set of numbers?

5, 5, 7, 12, 18, 10

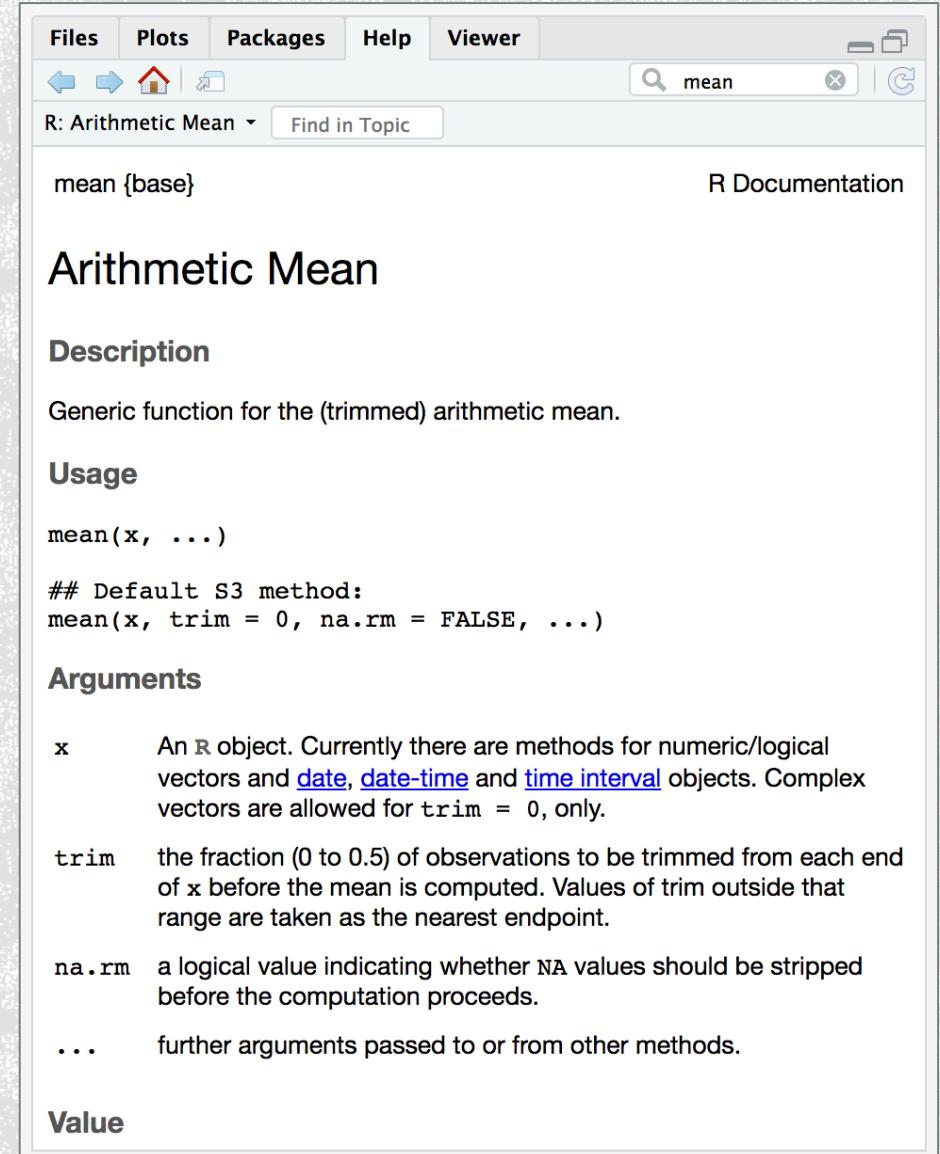
A human:

$$(5 + 5 + 7 + 12 + 18 + 10) / 6 = 9.5$$

R:

```
n = c(5, 5, 7, 12, 18, 10)
mean(x = n)
```

```
[1] 9.5
```



The screenshot shows the R documentation for the `mean` function. The title is "Arithmetic Mean". The description states it is a generic function for the (trimmed) arithmetic mean. The usage is `mean(x, ...)`. The default S3 method is `mean(x, trim = 0, na.rm = FALSE, ...)`. Arguments include `x` (an R object), `trim` (the fraction of observations to be trimmed from each end of `x` before the mean is computed), `na.rm` (a logical value indicating whether NA values should be stripped before the computation proceeds), and `...` (further arguments passed to or from other methods). The Value section is partially visible.



DEFINING A CUSTOM FUNCTION:

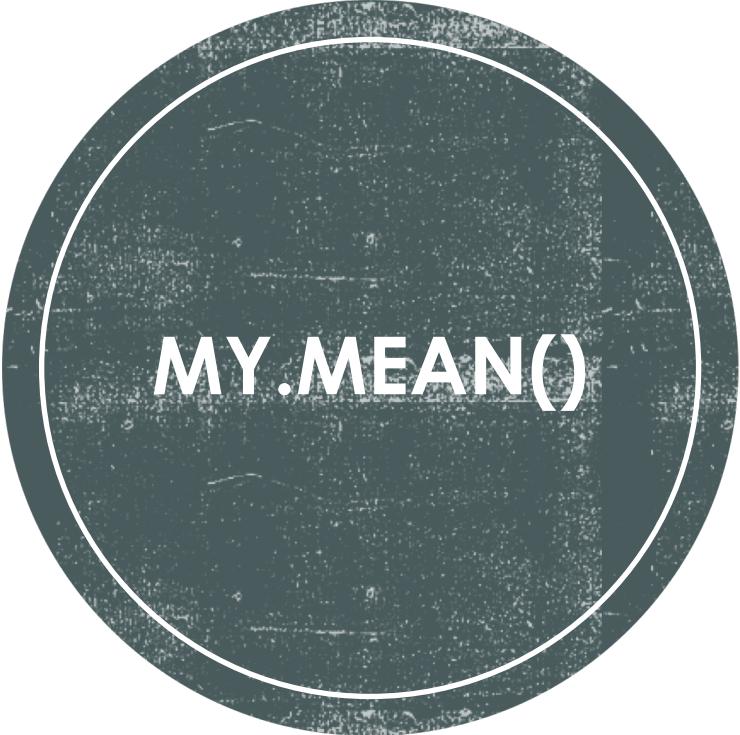
Create the function

NAME <- **function**(ARGUMENTS){ # Define inputs

ACTIONS # Code to “do stuff” on arguments

return(OUTPUT) # Return output to user

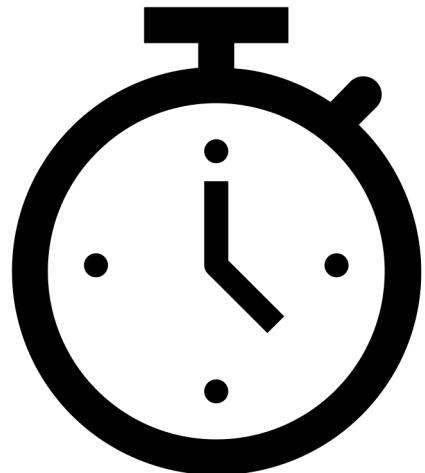
}



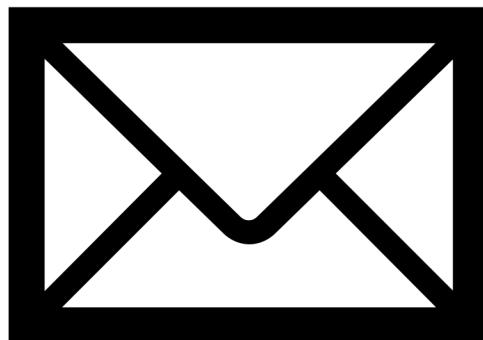
Create the function

```
my.mean <- function(x){ # Define inputs  
  output = sum(x)/length(x) # Code to “do stuff”  
  
return(output) # Return output to user  
}
```

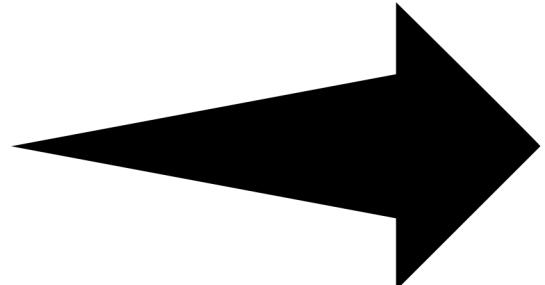
To do things faster



To share code
easier



To pass to other
functions



Why use custom functions?



Challenge #1

- In the **data** directory, you will find 5 csv (comma-separated values) files. Each file corresponds to water temperature data from a different lake at UNDERC, where the lake identifier is indicated by the two letter file name.
- Write a custom function to calculate the average water temperature of each lake at some depth. We want to use this function for multiple different lakes and different depths, so write the function so that the LakeID and depth are passed as arguments. Return a dataframe with one row and three columns as shown below:

lakeID	depth_m	meanTemp_C
CR	0	20.81429

Challenge #1

Pseudo-Code

First solve the problem, then write the code.
– John Johnson

```
# Create the function
lakeTemp<- function(lakeID, depth){  # Define inputs

    # Read in csv
    # Subset to only observations at depth
    # Calculate mean temperature
    # Create dataframe

    # Return dataframe to user
}
```

Challenge #2

- Modify your custom function from Challenge 1 to calculate the average water temperature of every lake in the **data** directory.
- Hint 1: Pass the directory name as an argument
- Hint 2: Use a **for loop** to loop through every lake in the directory

Challenge #2

Pseudo-Code

```
# Create the function  
lakeTemp<- function(dir, depth){ # Define inputs  
  
    # Get vector of every file in the directory  
    # Create your output dataframe  
    # Use a for loop to loop through every file  
        # Read in file for the ith lake  
        # Subset to only observations at depth  
        # Calculate mean temperature  
        # Create row  
        # Add the row to the bottom of the output  
  
    # Return dataframe to user  
}
```

COMMENT YOUR CODE!

I'm not a great programmer. I'm just
a good programmer with great habits.

- Kent Beck

