Introduction to Social Data Analytics: Class 26

Arushi Kaushik

arkaushi@ucsd.edu

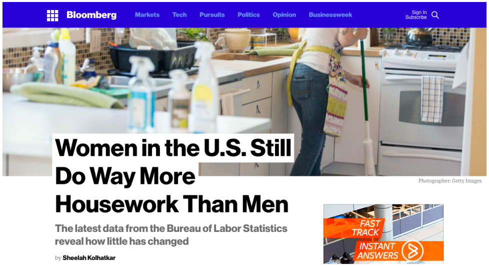
Today: (Homemade) Functions in R

By the end of today's lecture, you should be able to:

- Describe how functions can save time and space while writing code
- Construct functions that perform basic calculations, e.g. the mean of a subset of data
- Identify the inputs and output in a function

Please open class26.R and load AmericanTimeUse.RData if you haven't already.

Share of housework in American households



Women in the U.S. Still Do Way More Housework Than Men

The latest data from the Bureau of Labor Statistics reveal how little has changed

by Sheelah Kolhatkar

- American Time Use Survey (Bureau of Labor Statistics)
- Measures the amount of time people spend doing activities
- What did you do in the last 24 hours?
- Coded into categories by survey team

American Time Use Survey

```
# Read in Time use data
load("AmericanTimeUse.RData")
```

Documentation for data is:

`https://www.bls.gov/tus/lexiconnoex0315.pdf`
`https://www.bls.gov/tus/atusintcodebk0315.pdf`

Your first function from PC26

```
gender.mean <- function(gender) {
return(mean(time$household[time$TESEX == gender]))
}

gender.mean(1)
[1] 92.36724
gender.mean(2)
[1] 142.4052
```

How many inputs? How many outputs?

Let's create our own 'mean' function.

One input: a vector of numbers
One output: the mean of the vector of numbers

```
my.mean <- function(vector){
  mu <- sum(vector)/length(vector)
  return(mu)
}

my.mean(time$household)
[1] 120.5359
mean(time$household)
[1] 120.5359
```

Add ‘variable’ as an input:

```
my.mean <- function(variable, data){
  mu <- mean(data[,variable])
  return(mu)
}

my.mean("household", time)
[1] 120.5359
my.mean("children", time)
[1] 33.27538
```

Add gender as an input

Say we wanted to calculate the mean for men/women separately. This is one way we could do it outside of a function:

```
table(time$TESEX) # 1 is male; 2 is female
    1     2
74667 96175
mean(time$household[time$TESEX == 1]) # men
[1] 92.36724
mean(time$household[time$TESEX == 2]) # women
[1] 142.4052
```

Your turn! Write a function `gender.mean` that takes three inputs (variable, df, gender) and returns the mean of `df$variable` for the specified gender.

Add gender as an input

```
gender.mean <- function(variable, data, gender){
   mu <- mean(data[data$TESEX == gender,variable])
   return(mu)
}

gender.mean("household", time, 1)
[1] 92.36724
gender.mean("household", time, 2)
[1] 142.4052
```

Add age to the Function

Basically we're going to do the same thing that we just did, but we are going to subset the data by *both* gender and age.

Work with your partner to write the function `gender.age.mean`.

Return the mean in hours instead of minutes.

Add age to the Function

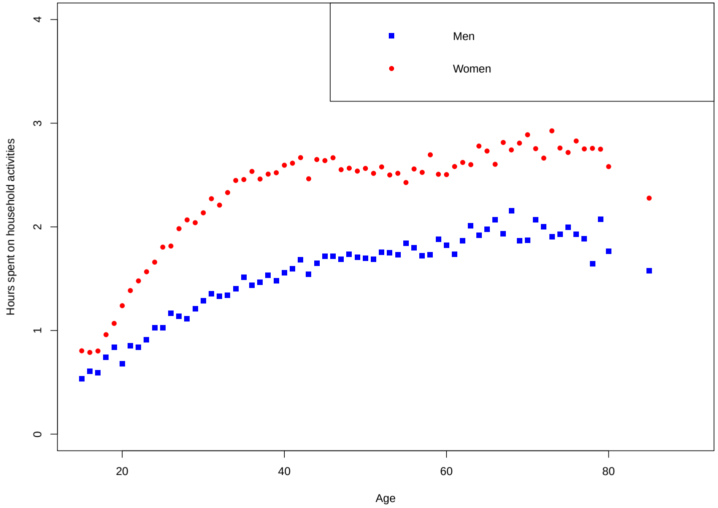
```
gender.age.mean <- function(variable, data, gender, age){
   mu <- mean(data[data$TESEX == gender &
              data$TEAGE == age, variable])/60
   return(mu)
}

gender.age.mean("work", time, 1, 20)
[1] 3.340932
gender.age.mean("work", time, 2, 20)
[1] 2.611888
gender.age.mean("work", time, 1, 30)
[1] 4.480926
gender.age.mean("work", time, 2, 30)
[1] 2.845387
```

Putting it all together

Your job is to create a function `plot.by.age` that does the following:

- Takes inputs variable, dataframe
- Outputs a plot
    - x-axis is years of age
    - y-axis is mean hours spent on the specified activity (variable)
    - Two data series, one for men, one for women
    - Each point is the mean hours for a given age
- Tip: you can use your created gender.age.mean() function inside your new function definition!

You final plot should look like this


Men
Women
Hours spent on household activities
Age