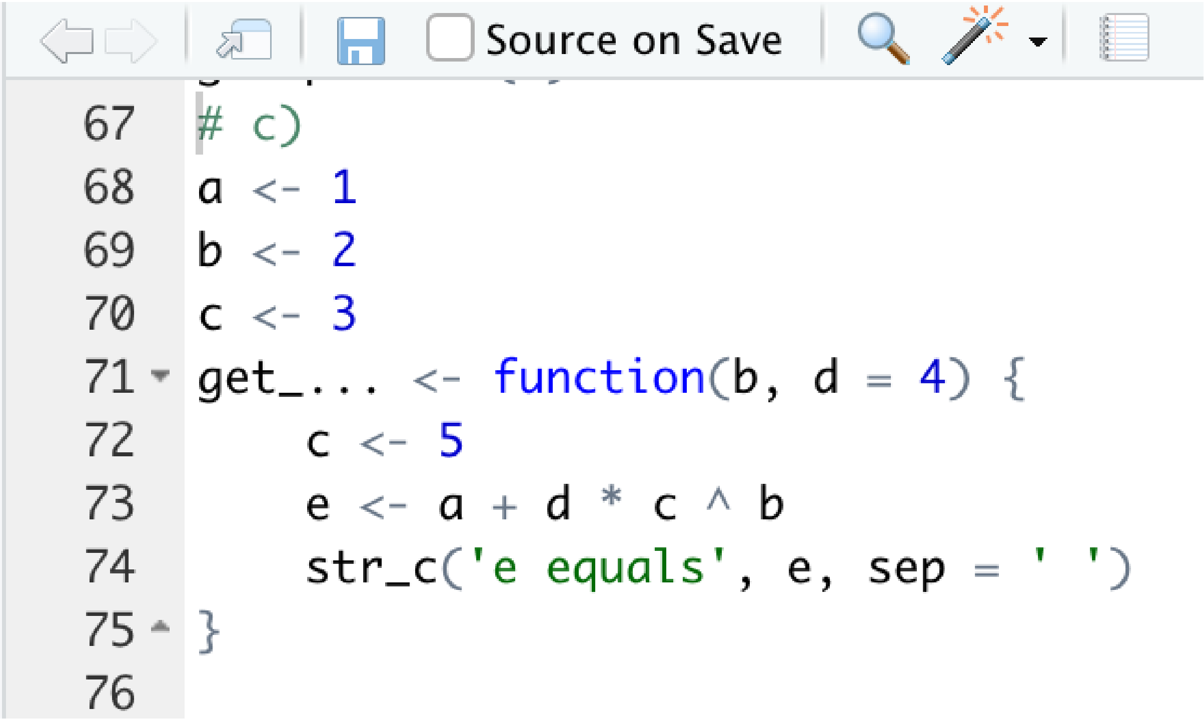
Programming - Functions

Data Science and Business Analytics

Exercise 1 – Functions

a) Write your own function with 2 arguments and return value the product of the two arguments.
b) Drop the second argument from your function call. Would your function still work? When would it (not)?
c) See the following function.



```
# c)
a <- 1
b <- 2
c <- 3
get_... <- function(b, d = 4) {
      c <- 5
      e <- a + d * c ^ b
      str_c('e equals', e, sep = ' ')
}
```

What would be the result of:

- get_...(b)
- get_...(1)
- get_...(d = 2)
- get_...(a = 2)
- get_...(b, d = 3)
- get_...(3, d = 1)
- f <- get_...(b)

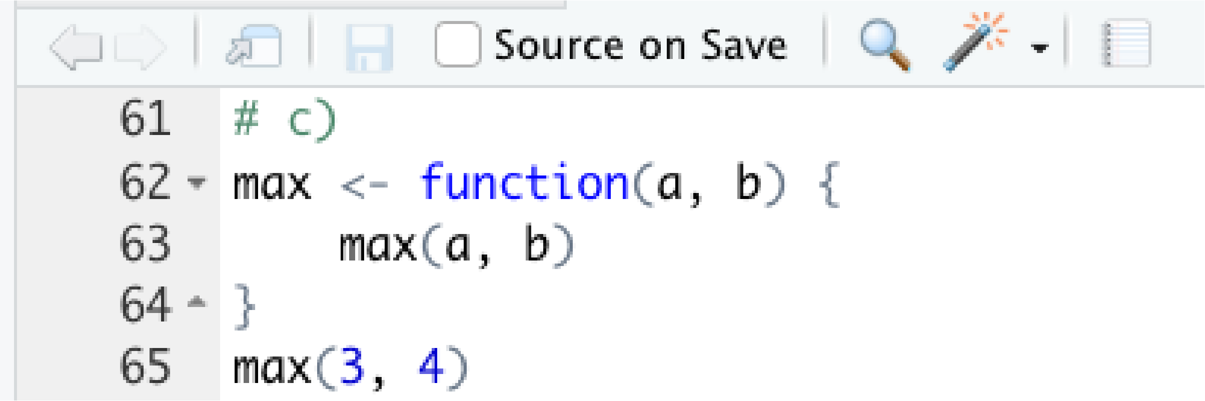
Why?

Exercise 2 – Functions

a) Create a function with 2 input arguments with the maximum of these inputs as output.

b) Break the function isolation and see what happens. Choose your way to do so.

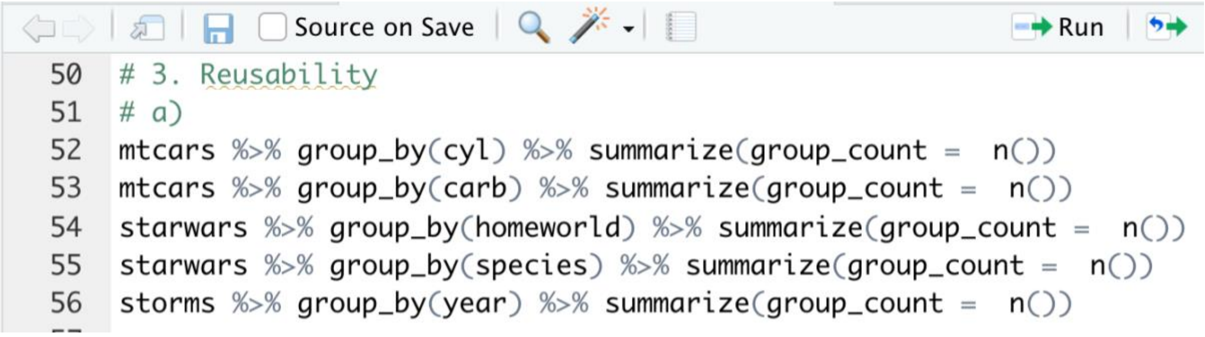
c) What is going wrong here, and why? If we want this function to work, how do we solve it?



```r
# c)
max <- function(a, b) {
      max(a, b)
}
max(3, 4)
```

Exercise 3 – Reusability

a) The below code has many copies. Replace this code with a convenient function using the afore mentioned steps:
- Copy own line
- Make a function of it using header and body
- Decide upon which parts of the body should change based on the input arguments of the function (do you need default values?)
- Change hard coded elements to parameter names (and use embracing where needed)



```r
# 3. Reusability
# a)
mtcars %>% group_by(cyl) %>% summarize(group_count =  n())
mtcars %>% group_by(carb) %>% summarize(group_count =  n())
starwars %>% group_by(homeworld) %>% summarize(group_count =  n())
starwars %>% group_by(species) %>% summarize(group_count =  n())
storms %>% group_by(year) %>% summarize(group_count =  n())
```

Exercise 4 – Reusability

We are using data on animal species diversity and weights found within plots at a study site. The dataset is stored as a comma separated value (CSV) file. Each row holds information for a single animal, and the columns represent:

record_id	unique observation-id
month	month of observation
day	day of observation
year	year of observation
plot_id	id of particular plot
species_id	id of particular species (2-letter code)
sex	sex of animal (‘F’ or ‘M’)
hindfoot_length	length of hindfoot (in mm)
weight	weight (in gram)
genus	genus (latin group name)
species	species (latin name)
taxa	class of animals
plot_type	type of plot

b) Download dataset animal_species.csv from Canvas and store it locally. Open a new R-script in R Studio and write the comment for loading the csv-data into R.

c) Write a function to find, by grouping argument, the average, standard deviation, 25 and 75 percent quantiles and the number of non-missing observations for a given variable.

d) Answer the following questions by using the function from b):

- What is the average weight in 1989?
- What is standard deviation of hindfoot_length for taxa ‘Rodent’?
- How many non-missing weights has genus ‘Onychomys’?