

writing functions

Davinder Singh

2023-03-21

in class exercise

```
get_mass_from_length_theropoda <- function(length){  
  mass <- 0.73 * length ^ 3.63  
  return(mass)  
}
```

```
get_mass_from_length_theropoda(16)
```

```
## [1] 17150.56
```

Use and modify function exercise

```
get_mass_from_length_theropoda <- function(length = 16){  
  mass <- 16 * length ^ 3.63  
  return(mass)  
}
```

```
get_mass_from_length <- function(length){  
  mass <- a * length ^ b  
  return(mass)  
}
```

```
get_mass_from_length <- function(a, b, length){  
  mass <- a * length ^ b  
  return(mass)  
}  
get_mass_from_length(214.44, 1.46, 26)
```

```
## [1] 24955.54
```

writing functions exercise

```
#convert_pounds_to_grams <- function(pounds = 1) {
  #grams = 453.6 * pounds
  #return(grams)
# }
#convert_pounds_to_grams() # object naming
# {
  #grams = 453.6 * pounds
  #return(grams)
#} #setting fuction up
# function(x) # function varaible
```

```
convert_pounds_to_grams <- function(grams, pounds) {
  pounds <- 3.75
  grams = 453.6 * pounds
  return(grams)
}
convert_pounds_to_grams(453.6, 3.75)
```

```
## [1] 1701
```

why do we use functions in programming?

- we already use functions a lot. Basically everything we run in R is a function. -We already use functions a lot. Basically everything we run in R is a function. -Reduce the amount of code we write -No need to repeat code, so less chance of repeating errors or introducing new errors by miss-typing code.
- Helps with reproducibility: no need to repeat code allows others (and future self) to run code in a straightfoward way, with accuracy -Makes it easier to write modular code that you can reuse for other projects
- Makes it easier to run and remember it.
- It allows to organize analysis by putting code intended to run something together in the same function.

Excerise 4

```
convert_kg_to_pounds <- function(kg = 0.45){
  pounds = 2.205 * kg
  return(pounds)
}
convert_kg_to_pounds()
```

```
## [1] 0.99225
```

```
convert_kg_to_pounds(50)
```

```
## [1] 110.25
```

```
get_mass_from_length(a = 10.95, b = 2.64, length = 12) %>%
  convert_kg_to_pounds()
```

```
## [1] 17055.37
```

```
length_in_pounds <- function(length, a, b){  
  get_mass_from_length(a = a, b = b, length = length) %>%  
  convert_kg_to_pounds()  
}
```

```
length_in_pounds(33, 5.6, 6.7)
```

```
## [1] 184348673945
```

excerise 5

```
surveys <- read.csv(file = "../data-raw/surveys (1).csv")  
species <- read.csv(file = "../data-raw/species.csv" )  
plots <- read.csv(file = "../data-raw/plots.csv")  
intersect(colnames(surveys), colnames(species))
```

```
## [1] "species_id"
```

```
a = inner_join(surveys, species, by = "species_id")  
head(a)
```

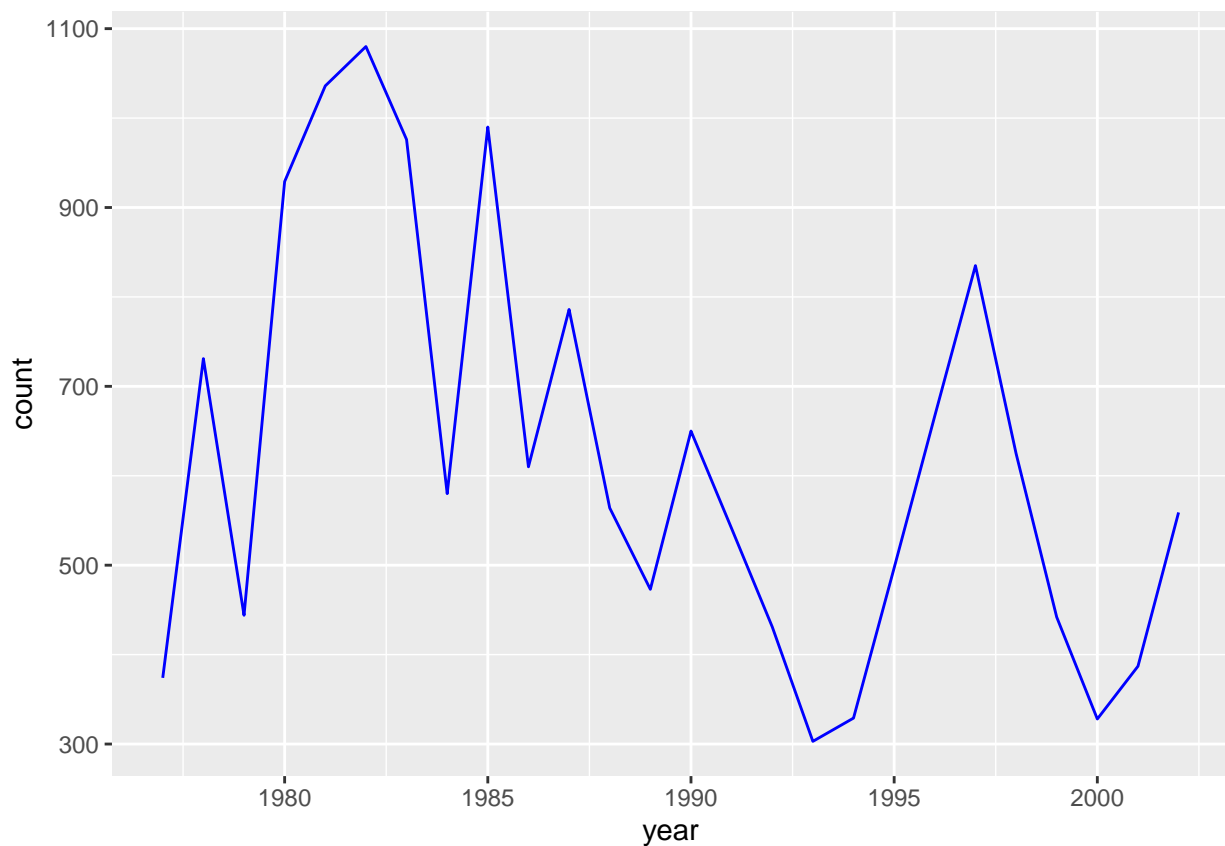
```
##   record_id month day year plot_id species_id sex hindfoot_length weight  
## 1         1     7  16 1977      2        NL  M             32      NA  
## 2         2     7  16 1977      3        NL  M             33      NA  
## 3         3     7  16 1977      2        DM  F             37      NA  
## 4         4     7  16 1977      7        DM  M             36      NA  
## 5         5     7  16 1977      3        DM  M             35      NA  
## 6         6     7  16 1977      1        PF  M             14      NA  
##           genus species  taxa  
## 1    Neotoma albigula Rodent  
## 2    Neotoma albigula Rodent  
## 3  Dipodomys merriami Rodent  
## 4  Dipodomys merriami Rodent  
## 5  Dipodomys merriami Rodent  
## 6 Perognathus  flavus Rodent
```

```
my_function <- function(a, genus_name = "Dipodomys", species_name = "merriami") {  
  a %>%  
  group_by(year) %>%  
  filter(genus == genus_name | species == species_name) %>%  
  summarize(count = n()) -> output  
  return(output)  
}  
my_function(a = a)
```

```
## # A tibble: 26 x 2  
##   year count
```

```
##      <int> <int>
## 1  1977   374
## 2  1978   731
## 3  1979   444
## 4  1980   929
## 5  1981  1036
## 6  1982  1080
## 7  1983   976
## 8  1984   580
## 9  1985   990
## 10 1986   610
## # ... with 16 more rows
```

```
my_function(a = a) %>%
  ggplot(mapping = aes(x = year, y = count)) +
  geom_line(color = "blue")
```



```
my_function(a = a, "Chaetodipus", "penicillatus") %>%
  ggplot(mapping = aes(x = year, y = count)) +
  geom_point(size = 1, color = "blue" ) +
  geom_line(size = 2, color = "blue") +
  labs(x = "year", y = "Number of Individuals")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```

