apply-prep

Azul Carrillo

2022-11-15

Reusing code - Apply function - The apply family of functions

Introduction to repeating things in R.

```
est_mass <- function(volume) {
  mass <- 2.65 * volume^0.9
  return(mass)
}

est_mass(1.6)
```

```
## [1] 4.045329
```

```
est_mass(5.6)
```

```
## [1] 12.49151
```

Using vectorized functions.

```
c(1, 2, 3) * 2 #element wise calc
```

```
## [1] 2 4 6
```

```
volumes <- c( 1.6, 5.6, 3.1)
est_mass(volumes)
```

```
## [1]  4.045329 12.491515  7.336204
```

```
library(stringr)

str_to_sentence(c("dipodomys", "chaetodipus")) # capitalazes the first letter of a sentence
```

```
## [1] "Dipodomys"   "Chaetodipus"
```

```r
genus <- c("dipodomys", "chaetodipus", "dipodomys")
species <- c("ordii","baileyi","spectabilis")

combined_genus_species <- function(genus, species){
  genus_cap <-  str_to_sentence(genus)
  genus_species <- paste(genus_cap, species)
  return(genus_species)
}
combined_genus_species(genus, species)
```

```
## [1] "Dipodomys ordii"       "Chaetodipus baileyi"   "Dipodomys spectabilis"
```

```r
data <- data.frame(genus, species)
combined_genus_species(data$genus, data$species)
```

```
## [1] "Dipodomys ordii"       "Chaetodipus baileyi"   "Dipodomys spectabilis"
```

Apply functions

```r
#apply the function to each item in the vector and return a vector or list of th e same size this doesn
est_mass <- function(volume) {
  if(volume > 5 ){
   mass <- 2.65 * volume^0.9
  } else { mass <- NA
    }
  return(mass)
}
volumes <- c( 1.6, 5.6, 3.1)
est_mass(volumes)
sapply(volumes, est_mass) # first argument is a single vector that has the volume and the second argume

#what the above sapply is doing under the surface
c(est_mass(volumes[1]), est_mass(volumes[2]), est_mass(volumes[3]))
```

The mapply function For Functions with Multiple Vector Arguments

```r
est_mass <- function(volume, veg_type) {
  if(veg_type == "tree" ){
   mass <- 2.65 * volume^0.9
  } else { mass <- NA
    }
  return(mass)
}
volumes <- c( 1.6, 5.6, 3.1)
veg_type <- c("shrub", "tree", "tree")
#function is the firat argument, the name of the argument = the name our vectoror object in the outter
mapply(FUN = est_mass, volume = volumes, veg_type = veg_type)
```

```
## [1]        NA 12.491515  7.336204
```

```
mapply(est_mass, volumes, veg_type)
```

```
## [1]        NA 12.491515  7.336204
```

Combining functions with dplyr - every row in a data frame or for every group by using group by

to repeat things once for esch row to create new columns we use mutate if func is not vectorized we need to add rowwide we can repeat things by using group by

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
est_mass <- function(volume, veg_type) {
  if(veg_type == "tree" ){
   mass <- 2.65 * volume^0.9
  } else { mass <- NA
    }
  return(mass)
}
volumes <- c( 1.6, 5.6, 3.1)
veg_type <- c("shrub", "tree", "tree")
plant_data <- data.frame (volumes, veg_type)

est_mass_vectorized <- function(volume){
  mass <- 2.65 * volume^0.9
  return(mass)
}

plant_data %>%
  mutate(masses = est_mass_vectorized(volumes))
```

```
##   volumes veg_type    masses
## 1     1.6    shrub  4.045329
## 2     5.6     tree 12.491515
## 3     3.1     tree  7.336204
```

```
#working with non vectorized functions this wont work unless uyou add rowwise function
plant_data %>%
  rowwise() %>%
  mutate(masses = est_mass(volumes, veg_type))
```

```
## # A tibble: 3 x 3
## # Rowwise:
##   volumes veg_type masses
##     <dbl> <chr>     <dbl>
## 1     1.6 shrub     NA
## 2     5.6 tree      12.5
## 3     3.1 tree       7.34
```

```
get_biomass <-  function(volumes) {
  masses <- est_mass_vectorized(volumes)
  biomass <- sum(masses)
  return(biomass)
}
get_biomass(volumes)
```

```
## [1] 23.87305
```

```
plant_data%>%
  group_by(veg_type)%>%
  summarize(biomass= get_biomass(volumes))
```

```
## # A tibble: 2 x 2
##   veg_type biomass
##   <chr>      <dbl>
## 1 shrub       4.05
## 2 tree       19.8
```