Apply Functions

Marcos Padilla-Ruiz

2022-11-15

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

```
library(stringr)
c(1, 2, 3) * 2
```

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

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

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

```
str_to_sentence(c("dipodomys", "chaetodipus"))
```

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

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

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

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

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

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

Apply functions

```r
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)
sapply(volumes, est_mass)
```

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

```r
c(est_mass(volumes[1]), est_mass(volumes[2]), est_mass(volumes[3]))
```

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

The mapply function

```r
est_mass <- function(volume, veg_type) {
  if (veg_type == "tree") {
  mass <- 2.65 * volume^0.9
  } else {
    mass <- NA
  }
  return(mass)
}
```

```r
volumes <- c(1.6, 5.6, 3.1)
veg_type <- c("shrub", "tree", "tree")

mapply(FUN = est_mass, volume = volumes, veg_type = veg_type)
```

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

Combining functions with dplyr

```r
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
```

```r
est_mass <- function(volume, veg_type) {
  if (veg_type == "tree") {
  mass <- 2.65 * volume^0.9
  } else {
    mass <- NA
  }
  return(mass)
}

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

volumes <- c(1.6, 5.6, 3.1)
veg_type <- c("shrub", "tree", "tree")
plant_data <- data.frame(volumes, veg_type)

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
```

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

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

```r
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
```