wrangling-portal

Austin Mercado

2023-03-07

The classic way of running code

For example I want the square root of the mean of a sequence of numbers

###Nested code

```
numbers <- 1:300
mean(numbers)
```

```
## [1] 150.5
```

```
sqrt(mean(numbers))
```

```
## [1] 12.26784
```

Sequential code

In this case we create intermedite variables

```
numbers <- -300:546
numbers <- 1:300
numbers_mean <- mean(numbers)
sqrt(x = numbers_mean)
```

```
## [1] 12.26784
```

Piping code

It can be implemented in R using the package 'magritte' It is a dependency of 'dplyr', so it is installed along.

```
install.packages("magrittr")
```

```
## Installing package into '/home/amercado44/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

The original The symbol of the pipe is '%>%' But we also have a new symbol that is simliar to bash '|>' The purpose of pipies is to eliminate or reduce to the max othe need of intermediate variables. For the mean example

```
1:300 %>% mean() %>% sqrt()
```

```
## [1] 12.26784
```

Pipes with the surveys data set

```
surveys <- read.csv("../data-raw/surveys.csv")
species <- read.csv("../data-raw/species.csv")
plots <- read.csv("../data-raw/plots.csv")
```

Calculate the mean of the year column using pipes

```
surveys$year %>% mean()
```

```
## [1] 1990.475
```

Calculate the mean of the weight column:

```
surveys$weight %>% mean(na.rm = TRUE)
```

```
## [1] 42.67243
```

```
# ?mean
```

Exercise 1

1. Load surveys.csv into R using read.csv()

```
surveys <- read.csv("../data-raw/surveys.csv")
species <- read.csv("../data-raw/species.csv")
plots <- read.csv("../data-raw/plots.csv")
```

Use select() to create a new data frame object surveys1 with just the year, month, day, and species_id columns in that order

```
surveys1 <- select(surveys, year, month, day, species_id)
str(surveys1)
```

```
## 'data.frame':    35549 obs. of  4 variables:
##  $ year      : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ month     : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day       : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ species_id: chr  "NL" "NL" "DM" "DM" ...
```

3. Create a new data frame called surveys2 with the year, species_id, and weight in kilograms of each individual, with no null weights. Use mutate(), select(), and filter() with !is.na(). The weight in the table is given in grams so you will need to create a new column called "weight_kg" for weight in kilograms by dividing the weight colum by 1000.

```
surveys2 <- select(surveys, year, species_id, weight)
surveys2 <- mutate(surveys2, weight_kg = weight/1000)
str(surveys2)
```

```
## ’data.frame’:    35549 obs. of  4 variables:
##  $ year      : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ species_id: chr  "NL" "NL" "DM" "DM" ...
##  $ weight    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ weight_kg : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
surveys2 <- filter(surveys2, !is.na(weight_kg))
```

```
surveys2 <- select(surveys2, year, species_id, weight_kg)
colnames(surveys2)
```

```
## [1] "year"       "species_id" "weight_kg"
```

```
# surveys2[, c(1,3)]
# surveys2[, c("year", "weight_kg")]
```

4. Use the filter() function to get all of the rows in the data frame surveys2 for the species ID “SH”.

```
surveys2_filtered <- filter(surveys2, species_id == "SH")
str(surveys2_filtered)
```

```
## ’data.frame’:    141 obs. of  3 variables:
##  $ year      : int  1978 1982 1982 1986 1987 1987 1987 1987 1987 1988 ...
##  $ species_id: chr  "SH" "SH" "SH" "SH" ...
##  $ weight_kg : num  0.089 0.106 0.052 0.055 0.077 0.078 0.104 0.058 0.052 0.06 ...
```