

wrangling-pipes.Rmd

Damian Sanchez

3/7/2023

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 intermediate 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 `magrittr`. It is a dependency of `dplyr`, so it is installed along.

```
library(magrittr)  
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
```

The original symbol of the pipe is `%>%`. But we also have a new symbol that is similar to bash `|>`. The purpose of this is to eliminate the need of intermediate variables. For the mean example

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

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

Pipes with the surveys data set

```
surveys <- read.csv(file = "../data-raw/surveys.csv")
str(surveys)
```

```
## 'data.frame':   35549 obs. of  9 variables:
##  $ record_id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int  2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr   "NL" "NL" "DM" "DM" ...
##  $ sex            : chr   "M" "M" "F" "M" ...
##  $ hindfoot_length: int  32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA NA ...
```

Calculate the mean of the year column using pipes

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

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

Calculate the mean of the weight column using pipes

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

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

Exercise 1

1)

```
surveys <- read.csv(file = "../data-raw/surveys.csv")
```

- 2) Use `select()` to create a new data frame object called `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()`, `filter()` with `!is.na()`.

```
surveys2 <- select(surveys, year, species_id, weight)
str(surveys2)
```

```
## 'data.frame': 35549 obs. of 3 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 ...
```

```
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))
str(surveys2)
```

```
## 'data.frame': 32283 obs. of 4 variables:
## $ year      : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
## $ species_id: chr  "DM" "DM" "DM" "DM" ...
## $ weight    : int  40 48 29 46 36 52 8 22 35 7 ...
## $ weight_kg : num  0.04 0.048 0.029 0.046 0.036 0.052 0.008 0.022 0.035 0.007 ...
```

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

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

```
## surveys2[c(1,3)]
str(surveys)
```

```
## 'data.frame': 35549 obs. of 9 variables:
## $ record_id : int 1 2 3 4 5 6 7 8 9 10 ...
## $ month : int 7 7 7 7 7 7 7 7 7 7 ...
## $ day : int 16 16 16 16 16 16 16 16 16 16 ...
## $ year : int 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
## $ plot_id : int 2 3 2 7 3 1 2 1 1 6 ...
## $ species_id : chr "NL" "NL" "DM" "DM" ...
## $ sex : chr "M" "M" "F" "M" ...
## $ hindfoot_length: int 32 33 37 36 35 14 NA 37 34 20 ...
## $ weight : int NA NA NA NA NA NA NA NA NA NA ...
```

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

4. Use the filter() function to get all of the rows

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

Exercise 2

```
read.csv(file = "../data-raw/surveys.csv") %>% select( year, month, day, species_id) -> surveys1
str(surveys1) # Read the data set then select columns and assign to object
```

```
## '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" ...
```

```
surveys %>% select( year, species_id, weight) %>%
  mutate(weight_kg = weight/1000) %>%
  filter(!is.na(weight)) %>%
  select(year, species_id, weight_kg) %>%
  filter(species_id == "SH") -> surveys_final
str(surveys_final)
```

```
## '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 ...
```

Exercise 3