The basics: 01 dplyr

Ari Anisfeld

8/28/2021

Basics are practice to help you get a handle on skills that you saw in the video.

Questions

We will not have basics for reading files. Instead we will use a dataset built into the tidyverse called `msleep` which provides information about mammals sleeping habits.

```
glimpse(msleep)
```

```
## Rows: 83
## Columns: 11
## $ name         <chr> "Cheetah", "Owl monkey", "Mountain beaver", "Greater shor~
## $ genus        <chr> "Acinonyx", "Aotus", "Aplodontia", "Blarina", "Bos", "Bra~
## $ vore         <chr> "carni", "omni", "herbi", "omni", "herbi", "herbi", "carn~
## $ order        <chr> "Carnivora", "Primates", "Rodentia", "Soricomorpha", "Art~
## $ conservation <chr> "lc", NA, "nt", "lc", "domesticated", NA, "vu", NA, "dome~
## $ sleep_total  <dbl> 12.1, 17.0, 14.4, 14.9, 4.0, 14.4, 8.7, 7.0, 10.1, 3.0, 5~
## $ sleep_rem    <dbl> NA, 1.8, 2.4, 2.3, 0.7, 2.2, 1.4, NA, 2.9, NA, 0.6, 0.8, ~
## $ sleep_cycle  <dbl> NA, NA, NA, 0.1333333, 0.6666667, 0.7666667, 0.3833333, N~
## $ awake        <dbl> 11.9, 7.0, 9.6, 9.1, 20.0, 9.6, 15.3, 17.0, 13.9, 21.0, 1~
## $ brainwt      <dbl> NA, 0.01550, NA, 0.00029, 0.42300, NA, NA, NA, 0.07000, 0~
## $ bodywt       <dbl> 50.000, 0.480, 1.350, 0.019, 600.000, 3.850, 20.490, 0.04~
```

If you enter `?msleep` in the console, you'll find some definitions of the columns. And learn that sleep times and weights were taken from V. M. Savage and G. B. West. A quantitative, theoretical framework for understanding mammalian sleep. Proceedings of the National Academy of Sciences, 104 (3):1051-1056, 2007.

Using `arrange()` to sort data

Arrange sorts data. It takes data in the first position (so you can pipe data to it easily) and columns in the next position. The following code orders data by the animal's name in alphabetical order.

```
msleep %>%
  arrange(name)
```

To get data in descending order, we write:

```
msleep %>%
  arrange(desc(name))
```

1. Use arrange to sort the data by `sleep_total`. This shows use the mammals that sleep the fewest hours. Giraffe's must drink a lot of coffee.

2. We can sort by multiple columns. Compare the output of the following code. How does `arrange()` handle multiple columns?

   ```r
   msleep %>%
     arrange(name, vore)

   msleep %>%
     arrange(vore, name)
   ```

3. What is the largest animal by `bodywt` in our data?

Using `select()` to pick columns

We often want to subset our data to include certain columns. This is where `select()` comes in:

~~~
```r
msleep %>%
  select(name, genus)
```
~~~

The code restricts our data to `name` and `genus`. We will introduce helper functions to use with `select()` throughout the course.

1. Write code to select the the columns `name` along with the three columns about sleep.

Using `mutate()` to create new columns.

`mutate()` is used to add columns with new variables to the dataset. For example, here I create a new variable called `sleep_pecent` that shows what percent of the 24-hour day the mammal sleeps. I provide code with `select()` that helps you check that your work makes sense.

~~~
```r
msleep %>%
  mutate(sleep_percent = sleep_total / 24) %>%
  select(name, sleep_total, sleep_percent)
```
~~~

1. Create a new variable called `sleep_nonrem` that shows the number of hours the mammal sleeps that are not in REM.

2. Create a column called `class` which is `Mammalia` for each of our mammals. (We might want to do this if we were going to join this data with data from other classes of animals such as birds `Aves`).

Using `mutate()` to create multiple new columns.

`mutate()` allows you create multiple columns simultaneously and even use the new variables within the `mutate()` call. Run the code to save the output to the name `msleep_with_percents` which we'll use later.

~~~
```r
msleep_with_percents <-
  msleep %>%
    mutate(sleep_percent = sleep_total / 24,
           awake_percent = 1 - sleep_percent)
```
~~~

1. Create a new variable called `percent_brain` which is the percent of body weight taken up by the brain. In the same mutate call, create a variable called `big_brain` that is TRUE if the brain takes up more than 1 percent of mass[1]

Using `filter()` to filter data by rows

`filter()` subsets rows in the data (first position) that matches criteria.

~~~
```r
msleep %>%
  filter(conservation == "domesticated")
```
~~~

`filter()` relies on comparison operators such as `==` (equals), `!=` (not equal to), `>` (greater than), `>=` (greater than or equal to) and so forth.

1. Use `filter()` to restrict the data set to carnivores. (hint: `vore == "carni"`). You should find 19 carnivores.
2. Find the five mammals that are `awake` less than or equal to 6 hours per day!
3. You could combine the two filters above to find that the Thick-tailed opposum is the one carnivore that sleeps more than 3/4 of the day. In particular, `msleep %>% filter(vore == "carni", awake <= 6)`. Can you find all non-carnivores that sleep less than 6 hours per day?

Using `summarize()` to summarize your data

`summarize()` summarizes data. For example, if we want to know the average number of hours the mammals in our data sleep we can write:

~~~
```r
msleep %>%
  summarize(sleep_avg = mean(sleep_total))
```
~~~

As with our other functions, we can do many summaries at a time as seen here, where we calculate the median (`median()`) along with the mean.

~~~
```r
msleep %>%
  summarize(sleep_avg = mean(sleep_total),
            sleep_median = median(sleep_total))
```
~~~

For now, we'll summarize the entire data set, but we'll see in a few lessons that we can group our data by sub-groups (e.g. `vore`-types) and get summary statistics for each group.[2]

1. Above you created `msleep_with_percents`. Use that data and create a summary of `sleep_percent` that includes with mean, median, standard deviation (`sd()`).

[1] Hint: This is achieved using `>`.

[2] For the curious, the code looks almost identical except we add `group_by(vore) %>%` before `summarize()` like so: `msleep %>% group_by(vore) %>% summarize(sleep_avg = mean(sleep_total))`

Solutions

Using `arrange()` to sort data

1. Use arrange to sort the data by `sleep_total`. This shows use the mammals that sleep the fewest hours. Giraffe’s must drink a lot of coffee.

   ```
   msleep %>%
     arrange(sleep_total)
   ```

   ```
   ## # A tibble: 83 x 11
   ##    name   genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
   ##    <chr>  <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
   ##  1 Giraf~ Gira~ herbi Arti~ cd                   1.9       0.4      NA      22.1
   ##  2 Pilot~ Glob~ carni Ceta~ cd                   2.7       0.1      NA      21.4
   ##  3 Horse  Equus herbi Peri~ domesticated         2.9       0.6       1      21.1
   ##  4 Roe d~ Capr~ herbi Arti~ lc                   3        NA        NA      21
   ##  5 Donkey Equus herbi Peri~ domesticated         3.1       0.4      NA      20.9
   ##  6 Afric~ Loxo~ herbi Prob~ vu                   3.3      NA        NA      20.7
   ##  7 Caspi~ Phoca carni Carn~ vu                   3.5       0.4      NA      20.5
   ##  8 Sheep  Ovis  herbi Arti~ domesticated         3.8       0.6      NA      20.2
   ##  9 Asian~ Elep~ herbi Prob~ en                   3.9      NA        NA      20.1
   ## 10 Cow    Bos   herbi Arti~ domesticated         4         0.7       0.667  20
   ## # ... with 73 more rows, and 2 more variables: brainwt <dbl>, bodywt <dbl>
   ```

2. We can sort by multiple columns. Compare the output of the following code. How does `arrange()` handle multiple columns?

   First R sorts by the first column, then R sorts the second column within the first columns “groups”. Which we see in the second example. First, it sorts the data by `vore`-type and then within `vore`-type it alphabetizes by `name`.

   ```
   msleep %>%
     arrange(name, vore)

   msleep %>%
     arrange(vore, name)
   ```

3. What is the largest animal by `bodywt` in our data?

   The African elephant

   ```
   msleep %>%
     arrange(desc(bodywt))
   ```

Using `select()` to pick columns

1. Write code to select the the columns `name` along with the three columns about sleep.

   ```
   msleep %>%
     select(name, sleep_total, sleep_rem, sleep_cycle)

   # advanced users might use a tidyselect helper
   msleep %>%
     select(name, starts_with("sleep"))
   ```

Using mutate() to create new columns.

1. Create a new variable called `sleep_nonrem` that shows the number of hours the mammal sleeps that are not in REM.

```
msleep %>%
  mutate(sleep_nonrem = sleep_total - sleep_rem) %>%
  select(name, sleep_total, sleep_rem, sleep_nonrem)
```

2. Create a column called `class` which is `Mammalia` for each of our mammals. (We might want to do this if we were going to join this data with data from other classes of animals such as birds `Aves`).

```
msleep %>%
  mutate(class = "Mammalia")
```

Using mutate() to create multiple new columns.

1. Create a new variable called `percent_brain` which is the percent of body weight taken up by the brain. In the same mutate call, create a variable called `big_brain` that is TRUE if the brain takes up more than 1 percent of mass[3]

```
msleep %>%
  mutate(percent_brain = brainwt/bodywt,
         big_brain = percent_brain > .01)
```

```
## # A tibble: 83 x 13
##    name   genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##    <chr>  <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
##  1 Cheet~ Acin~ carni Carn~ lc                  12.1      NA        NA      11.9
##  2 Owl m~ Aotus omni  Prim~ <NA>                17         1.8      NA       7
##  3 Mount~ Aplo~ herbi Rode~ nt                  14.4       2.4      NA       9.6
##  4 Great~ Blar~ omni  Sori~ lc                  14.9       2.3       0.133   9.1
##  5 Cow    Bos   herbi Arti~ domesticated         4         0.7       0.667  20
##  6 Three~ Brad~ herbi Pilo~ <NA>                14.4       2.2       0.767   9.6
##  7 North~ Call~ carni Carn~ vu                   8.7       1.4       0.383  15.3
##  8 Vespe~ Calo~ <NA>  Rode~ <NA>                 7        NA        NA      17
##  9 Dog    Canis carni Carn~ domesticated        10.1       2.9       0.333  13.9
## 10 Roe d~ Capr~ herbi Arti~ lc                   3        NA        NA      21
## # ... with 73 more rows, and 4 more variables: brainwt <dbl>, bodywt <dbl>,
## #   percent_brain <dbl>, big_brain <lgl>
```

Using filter() to filter data by rows

1. Use `filter()` to restrict the data set to carnivores. (hint: `vore == "carni"`). You should find 19 carnivores.

```
msleep %>%
  filter(vore == "carni")
```

```
## # A tibble: 19 x 11
##    name   genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##    <chr>  <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
##  1 Cheet~ Acin~ carni Carn~ lc                  12.1      NA        NA      11.9
```

[3] Hint: This is achieved using >.

```
##  2 North~ Call~ carni Carn~ vu                   8.7       1.4       0.383  15.3
##  3 Dog    Canis carni Carn~ domesticated        10.1       2.9       0.333  13.9
##  4 Long-~ Dasy~ carni Cing~ lc                  17.4       3.1       0.383   6.6
##  5 Domes~ Felis carni Carn~ domesticated        12.5       3.2       0.417  11.5
##  6 Pilot~ Glob~ carni Ceta~ cd                   2.7       0.1      NA      21.4
##  7 Gray ~ Hali~ carni Carn~ lc                   6.2       1.5      NA      17.8
##  8 Thick~ Lutr~ carni Dide~ lc                  19.4       6.6      NA       4.6
##  9 Slow ~ Nyct~ carni Prim~ <NA>                11        NA        NA      13  
## 10 North~ Onyc~ carni Rode~ lc                  14.5      NA        NA       9.5
## 11 Tiger  Pant~ carni Carn~ en                  15.8      NA        NA       8.2
## 12 Jaguar Pant~ carni Carn~ nt                  10.4      NA        NA      13.6
## 13 Lion   Pant~ carni Carn~ vu                  13.5      NA        NA      10.5
## 14 Caspi~ Phoca carni Carn~ vu                   3.5       0.4      NA      20.5
## 15 Commo~ Phoc~ carni Ceta~ vu                   5.6      NA        NA      18.4
## 16 Bottl~ Turs~ carni Ceta~ <NA>                 5.2      NA        NA      18.8
## 17 Genet  Gene~ carni Carn~ <NA>                 6.3       1.3      NA      17.7
## 18 Arcti~ Vulp~ carni Carn~ <NA>                12.5      NA        NA      11.5
## 19 Red f~ Vulp~ carni Carn~ <NA>                 9.8       2.4       0.35   14.2
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

2. Find the five mammals that are `awake` less than or equal to 6 hours per day!

```
msleep %>%
  filter(awake <= 6)
```

```
## # A tibble: 5 x 11
##   name   genus  vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr>  <chr>  <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
## 1 North~ Didel~ omni  Dide~ lc                 18         4.9       0.333   6  
## 2 Big b~ Eptes~ inse~ Chir~ lc                 19.7       3.9       0.117   4.3
## 3 Thick~ Lutre~ carni Dide~ lc                 19.4       6.6      NA       4.6
## 4 Littl~ Myotis inse~ Chir~ <NA>               19.9       2         0.2     4.1
## 5 Giant~ Priod~ inse~ Cing~ en                 18.1       6.1      NA       5.9
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

3. You could combine the two filters above to find that the Thick-tailed opposum is the one carnivore that sleeps more than 3/4 of the day. In particular, `msleep %>% filter(vore == "carni", awake <= 6)`. Can you find all non-carnivores that sleep less than 6 hours per day?

```
msleep %>%
  filter(sleep_total < 6, vore != "carni")
```

```
## # A tibble: 11 x 11
##    name   genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##    <chr>  <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
##  1 Cow    Bos   herbi Arti~ domesticated         4         0.7       0.667  20  
##  2 Roe d~ Capr~ herbi Arti~ lc                   3        NA        NA      21  
##  3 Goat   Capri herbi Arti~ lc                   5.3       0.6      NA      18.7
##  4 Tree ~ Dend~ herbi Hyra~ lc                   5.3       0.5      NA      18.7
##  5 Asian~ Elep~ herbi Prob~ en                   3.9      NA        NA      20.1
##  6 Horse  Equus herbi Peri~ domesticated         2.9       0.6       1      21.1
##  7 Donkey Equus herbi Peri~ domesticated         3.1       0.4      NA      20.9
##  8 Giraf~ Gira~ herbi Arti~ cd                   1.9       0.4      NA      22.1
##  9 Afric~ Loxo~ herbi Prob~ vu                   3.3      NA        NA      20.7
## 10 Sheep  Ovis  herbi Arti~ domesticated         3.8       0.6      NA      20.2
## 11 Brazi~ Tapi~ herbi Peri~ vu                   4.4       1         0.9    19.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

Using summarize() to summarize your data

1. Above you created msleep_with_percents. Use that data and create a summary of sleep_percent that includes with mean, median, standard deviation (sd()).

```
msleep_with_percents <-
  msleep %>%
    mutate(sleep_percent = sleep_total / 24,
           awake_percent = 1 - sleep_percent)

msleep_with_percents %>%
  summarize(sleep_pct_avg = mean(sleep_percent),
            sleep_pct_median = median(sleep_percent),
            sleep_pct_sd= sd(sleep_percent))
```