Module 1, Assignment 3

2023-02-13

Assignment Details

Purpose

The goal of this assignment is to use and become comfortable with working with 2-dimensional data using the `tidyverse`.

Task

Write R code to successfully answer each question below or write text to successfully answer the question.

Criteria for Success

- Code is within the provided code chunks
- Code chunks run without errors
- Code produces the correct result
  - Code attempts will get half points
  - Code that produces the correct answer will receive full points
- Text answers correctly address the question asked

Due Date

February 20 at midnight MST

Assignment Questions

All questions are worth 1 point unless otherwise specified.

If you create an object in a question, please type the name of the object in the next line so it will print out when you save your file as a PDF. For example:

```
# this doesn't actually run--it's just an example
new_object <- dataframe[row, column]
new_object
```

Definitions

In your own words, define/describe the following terms. These don't need to be technical descriptions but rather how you are thinking about them.

1. *data frame/tibble*: 2D data (rows and columns)
2. *R package*: group of pre-written function/code chunks/data that we can use in R
3. *tidyverse*: package of packages with more-intuitive syntax for humans
4. *the pipe (%>%)*: an operator that connects one line of (left-side) to the next (right-side). It sends the results of the first line through to the next.

Using the `tidyverse`

For the rest of the assignment, you will be writing or interpreting code from the `tidyverse` using the merged cactus pad data (the same as Assignment 2).

5. Load the `tidyverse` package into RStudio.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Important!

Make sure you run the following code chunk in order to read in the cactus pads data set that we will be using for this assignment.

```
pads <- read_csv("../data_raw/CactusPads_joined.csv")
```

```
## Rows: 108 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): spines, insects, damage, location, species, size
## dbl (5): group_id, paddle_id, length_in, width_in, depth_in
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pads
```

```
## # A tibble: 108 x 11
##    group_id paddle_id length_in width_in depth_in spines insects damage location
##       <dbl>     <dbl>     <dbl>    <dbl>    <dbl> <chr>  <chr>   <chr>  <chr>
##  1        1         5       1        1      0.5   N      N       Most   Sixth
##  2        1         4       2        1      0.3   N      N       None   Eigth
##  3        3         7       2        1      0.125 Y      N       None   Sixth
##  4        2         9       2        3      0.5   N      N       All    Seventh
##  5        7         9       2.5      2.5    0.5   Y      Y       None   Fifth
##  6        3         8       3.5      2.5    0.25  N      Y       All    Third
##  7        7         4       4        3.75   0.5   Y      Y       Some   Sixth
##  8        7         5       4        3.8    0.25  Y      Y       None   Fifth
##  9        7         8       4        3.8    0.5   Y      Y       Some   Fifth
## 10        7        10       4        3.5    0.25  Y      Y       Most   First
## # ... with 98 more rows, and 2 more variables: species <chr>, size <chr>
```

6. There are many different functions we can use to look at our data set. Choose your favorite to get an idea of all the data in `pads`.

```
str(pads)
```

```
## spec_tbl_df [108 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ group_id : num [1:108] 1 1 3 2 7 3 7 7 7 7 ...
##  $ paddle_id: num [1:108] 5 4 7 9 9 8 4 5 8 10 ...
##  $ length_in: num [1:108] 1 2 2 2 2.5 3.5 4 4 4 4 ...
##  $ width_in : num [1:108] 1 1 1 3 2.5 2.5 3.75 3.8 3.8 3.5 ...
##  $ depth_in : num [1:108] 0.5 0.3 0.125 0.5 0.5 0.25 0.5 0.25 0.5 0.25 ...
##  $ spines   : chr [1:108] "N" "N" "Y" "N" ...
##  $ insects  : chr [1:108] "N" "N" "N" "N" ...
##  $ damage   : chr [1:108] "Most" "None" "None" "All" ...
##  $ location : chr [1:108] "Sixth" "Eigth" "Sixth" "Seventh" ...
##  $ species  : chr [1:108] "Opuntia ficus-indica" "Opuntia ficus-indica" "Opuntia ficus-indica" "Opun
##  $ size     : chr [1:108] "Large" "Large" "Large" "Large" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   group_id = col_double(),
##   ..   paddle_id = col_double(),
##   ..   length_in = col_double(),
##   ..   width_in = col_double(),
##   ..   depth_in = col_double(),
##   ..   spines = col_character(),
##   ..   insects = col_character(),
##   ..   damage = col_character(),
##   ..   location = col_character(),
##   ..   species = col_character(),
##   ..   size = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# head(pads)
# names(pads)
# whatever they choose
```

7. Which of the columns in **pads** are data class "character"? List the columns below.

   Answer: spines, insects, damage, location, species, size

8. Make a data frame from **pads** that has the columns with dimensions (length, width, depth) of the cactus pads and the species. (Hint: make sure you are typing the correct column names!)

```
pads %>%
  select(length_in, width_in, depth_in, species)
```

```
## # A tibble: 108 x 4
##    length_in width_in depth_in species
##        <dbl>    <dbl>    <dbl> <chr>
##  1       1        1       0.5   Opuntia ficus-indica
##  2       2        1       0.3   Opuntia ficus-indica
##  3       2        1       0.125 Opuntia ficus-indica
##  4       2        3       0.5   Opuntia ficus-indica
##  5       2.5      2.5     0.5   Opuntia santa-rita
##  6       3.5      2.5     0.25  Opuntia ficus-indica
##  7       4        3.75    0.5   Opuntia santa-rita
##  8       4        3.8     0.25  Opuntia santa-rita
##  9       4        3.8     0.5   Opuntia santa-rita
## 10       4        3.5     0.25  Opuntia santa-rita
## # ... with 98 more rows
```

```
# OR select(pads, length_in, width_in, depth_in, species)
```

9. Make a data frame from **pads** that only includes paddles with a depth that is greater that 1 in.

```
pads %>%
  filter(depth_in > 1)
```

```
## # A tibble: 6 x 11
##   group_id paddle_id length_in width_in depth_in spines insects damage location
##      <dbl>     <dbl>     <dbl>    <dbl>    <dbl> <chr>  <chr>   <chr>  <chr>
## 1        4         6       9        8       1.5  N      N       Most   Fifth
## 2        9         5       9.5      8       1.5  Y      Y       Most   First
## 3        9         6       9.5     11       1.5  Y      Y       Most   First
## 4        9         9      10        7.5     1.25 Y      N       None   Third
## 5        9         2      11       10.8     1.25 Y      N       Most   First
## 6        9        10      12        8.5     1.25 Y      Y       Some   Third
## # ... with 2 more variables: species <chr>, size <chr>
```

```
# OR filter(pads, depth_in > 1)
```

10. Run the lines of code below. Describe what each of the three lines are doing (3 points).

```
large_no_spines <- pads %>%
  filter(spines == "N", size == "Large") %>%
  select(group_id, paddle_id, spines, size, species)
```

Line 1: creates a new dataframe called large_no_spines and starts with the pads dataframe, which is piped to line 2
Line 2: chooses rows based on conditions (only cactus pads with no spies and cactuses that were large) and pipes that through to the next line
Line 3: selects 5 columns from the resulting dataframe (group_id, paddle_id, spines, size, species)

11. Let's convert the length column to centimeters. Create a new column named `length_cm` which has the length of each cactus pad in centimeters. (Hint: there are 2.54 cm per inch).

    Use the pipe (`%>%`) and the `select()` function to show the `length_in` and `length_cm` columns side by side. (Hint: order matters here! You can't select a column that doesn't yet exist...).

```
mutate(pads, length_cm = length_in * 2.54) %>%
  select(length_in, length_cm)
```

```
## # A tibble: 108 x 2
##    length_in length_cm
##        <dbl>     <dbl>
##  1       1        2.54
##  2       2        5.08
##  3       2        5.08
##  4       2        5.08
##  5       2.5      6.35
##  6       3.5      8.89
##  7       4       10.2
##  8       4       10.2
##  9       4       10.2
## 10       4       10.2
## # ... with 98 more rows
```

12. Use the `summarize()` function to calculate the mean (`mean()`) and standard deviation (`sd()`) of the cactus pad widths (in inches).

```
pads %>%
  summarise(mean_width = mean(width_in),
            sd_width = sd(width_in))
```

```
## # A tibble: 1 x 2
##   mean_width sd_width
##        <dbl>    <dbl>
## 1       5.66     2.31
```

13. Create a new data frame with a new column called "volume_in3" that has the volume of each cactus pad (Hint: volume = length * width * depth). Save this data frame by naming it `pads_volume`.

```
pads_volume <- pads %>%
  mutate(volume_in3 = length_in * width_in * depth_in)
pads_volume
```

```
## # A tibble: 108 x 12
##    group_id paddle_id length_in width_in depth_in spines insects damage location
##       <dbl>     <dbl>     <dbl>    <dbl>    <dbl> <chr>  <chr>   <chr>  <chr>
##  1        1         5       1        1      0.5   N      N       Most   Sixth
##  2        1         4       2        1      0.3   N      N       None   Eigth
##  3        3         7       2        1      0.125 Y      N       None   Sixth
##  4        2         9       2        3      0.5   N      N       All    Seventh
##  5        7         9       2.5      2.5    0.5   Y      Y       None   Fifth
##  6        3         8       3.5      2.5    0.25  N      Y       All    Third
##  7        7         4       4        3.75   0.5   Y      Y       Some   Sixth
##  8        7         5       4        3.8    0.25  Y      Y       None   Fifth
##  9        7         8       4        3.8    0.5   Y      Y       Some   Fifth
## 10        7        10       4        3.5    0.25  Y      Y       Most   First
## # ... with 98 more rows, and 3 more variables: species <chr>, size <chr>,
## #   volume_in3 <dbl>
```

14. Using the `pads_volume` dataframe that you just created, calculate the mean and standard deviation of the cactus pad volumes for *each* species of cactus. (2 points)

```
pads_volume %>%
  group_by(species) %>%
  summarize(mean_volume = mean(volume_in3),
            sd_volume = sd(volume_in3))
```

```
## # A tibble: 3 x 3
##   species              mean_volume sd_volume
##   <chr>                      <dbl>     <dbl>
## 1 Opuntia engelemannii       52.2      41.8
## 2 Opuntia ficus-indica       21.4      19.2
## 3 Opuntia santa-rita          7.87      7.55
```

16. Calculate the coefficient of variation for *each* species for the volume of the cactus pads. (3 points)

    This question has a lot of steps, so think through each one individually before trying to write the code.

    - which function will you need to get values for *each* species?
    - what is the equation for CV?
    - how will you calculate the necessary values for the CV? (Hint: see question above)
    - what function will you use to create a new column for the CV?

```
pads_volume %>%
  group_by(species) %>%
  summarize(mean_volume = mean(volume_in3),
            sd_volume = sd(volume_in3)) %>%
  mutate(CV = sd_volume / mean_volume * 100)
```

```
## # A tibble: 3 x 4
##   species              mean_volume sd_volume    CV
```

```
##   <chr>                     <dbl>     <dbl> <dbl>
## 1 Opuntia engelemannii      52.2      41.8   80.2
## 2 Opuntia ficus-indica      21.4      19.2   89.7
## 3 Opuntia santa-rita         7.87      7.55  96.0
```

Which cactus species has the highest relative variation in cactus pad volume?

Answer: Opuntia santa-rita

Turning in Your Assignment

Follow these steps to successfully turn in your assignment on D2L.

1. Click the `Knit` button up near the top of this document. This should produce a PDF file that shows up in the `Files` panel on the bottom-right of your screen.
    - if this doesn't work, that's ok! just follow the same steps with the .Rmd file
2. Click the empty box to the left of the PDF file.
3. Click on the blue gear near the top of the `Files` panel and choose Export.
4. Put your last name at the front of the file name when prompted, then click the Download button. The PDF file of your assignment is now in your "Downloads" folder on your device.
5. Head over to D2L and navigate to Module 1 Assignment 2. Submit the PDF file that you just downloaded.