Week 3: Assignment

Ellen Bledsoe

2024-01-29

Assignment Description

Purpose

The goal of this assignment is to get comfortable importing data and working with 2-dimensional data in R using `dplyr`, part of the `tidyverse`.

Task

Write R code to successfully answer each question below.

Criteria for Success

- Code is within the provided code chunks or new code chunks are created where necessary
- Code chunks run without errors
- Code chunks have brief comments indicating which code is answering which part of the question
- Code will be assessed as follows:
  - Produces the correct answer using the requested approach: 100%
  - Generally uses the right approach, but a minor mistake results in an incorrect answer: 90%
  - Attempts to solve the problem and makes some progress using the core concept, but returns the wrong answer and does not demonstrate comfort with the core concept: 50%
  - Answer demonstrates a lack of understanding of the core concept: 0%
- Any questions requiring written answers are answered with sufficient detail

Due Date

Feb 5 at midnight MST

Exercises

Set-up (5 pts)

To complete these exercises, we will need function from the `dplyr` package. Add a code chunk below and load `dplyr` into the workspace before beginning.

```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
library(readr)
```

1. Shrub Volume Data Basics (20 pts)

Dr. Morales is interested in studying the factors controlling the size and carbon storage of shrubs. She has conducted an experiment looking at the effect of three different treatments on shrub volume at four different locations.

Get familiar with the data by importing it using `read.csv()` and use `dplyr` functions to complete the following tasks. Remember to add a code chunk!

a. Select the data from the length column.
b. Select the data from the site and experiment columns.
c. Add a new column named `area` containing the area of the shrub, which is the length times the width.
d. Sort the data by length.
e. Filter the data to include only plants with heights greater than 5.
f. Filter the data to include only plants with heights greater than 4 and widths greater than 2 (using `,` or `&` to include two conditions).
g. Filter the data to include only plants from Experiment 1 or Experiment 3 (using '|' for "or").
h. Filter the data to remove rows with null values in the height column (using `!is.na`)
i. Create a new data frame called `shrub_volumes` that includes all of the original data and a new column containing the volumes (length * width * height). Remember to add a line of code afterwards with the name of the dataframe so the dataframe is printed.

```r
# read in shrubs data
shrubs <- read.csv("shrub-volume-data.csv")

# a. Select the data from the length column.
select(shrubs, length)
```

```
##    length
## 1     2.2
## 2     2.1
## 3     2.7
## 4     3.0
## 5     3.1
## 6     2.5
## 7     1.9
```

```
## 8     1.1
## 9     3.5
## 10    2.9
## 11    4.5
## 12    1.2
## 13    2.6
## 14    1.8
## 15    3.1
```

```
# b. Select the data from the site and experiment columns.
select(shrubs, site, experiment)
```

```
##    site experiment
## 1     1          1
## 2     1          2
## 3     1          3
## 4     2          1
## 5     2          2
## 6     2          3
## 7     3          1
## 8     3          2
## 9     3          3
## 10    4          1
## 11    4          2
## 12    4          3
## 13    5          1
## 14    5          2
## 15    5          3
```

```
# c. Add a new column named area containing the area of the shrub, which is the length times the width.
mutate(shrubs, area = length * width)
```

```
##    site experiment length width height  area
## 1     1          1    2.2   1.3    9.6  2.86
## 2     1          2    2.1   2.2    7.6  4.62
## 3     1          3    2.7   1.5    2.2  4.05
## 4     2          1    3.0   4.5    1.5 13.50
## 5     2          2    3.1   3.1    4.0  9.61
## 6     2          3    2.5   2.8    3.0  7.00
## 7     3          1    1.9   1.8    4.5  3.42
## 8     3          2    1.1   0.5    2.3  0.55
## 9     3          3    3.5   2.0    7.5  7.00
## 10    4          1    2.9   2.7    3.2  7.83
## 11    4          2    4.5   4.8    6.5 21.60
## 12    4          3    1.2   1.8    2.7  2.16
## 13    5          1    2.6   0.8     NA  2.08
## 14    5          2    1.8    NA    5.2    NA
## 15    5          3    3.1   2.2     NA  6.82
```

```
# d. Sort the data by length.
arrange(shrubs, length)
```

```
##    site experiment length width height
## 1     3          2    1.1   0.5    2.3
## 2     4          3    1.2   1.8    2.7
## 3     5          2    1.8    NA    5.2
## 4     3          1    1.9   1.8    4.5
## 5     1          2    2.1   2.2    7.6
## 6     1          1    2.2   1.3    9.6
## 7     2          3    2.5   2.8    3.0
## 8     5          1    2.6   0.8     NA
## 9     1          3    2.7   1.5    2.2
## 10    4          1    2.9   2.7    3.2
## 11    2          1    3.0   4.5    1.5
## 12    2          2    3.1   3.1    4.0
## 13    5          3    3.1   2.2     NA
## 14    3          3    3.5   2.0    7.5
## 15    4          2    4.5   4.8    6.5
```

```
# e. Filter the data to include only plants with heights greater than 5.
filter(shrubs, height > 5)
```

```
##   site experiment length width height
## 1    1          1    2.2   1.3    9.6
## 2    1          2    2.1   2.2    7.6
## 3    3          3    3.5   2.0    7.5
## 4    4          2    4.5   4.8    6.5
## 5    5          2    1.8    NA    5.2
```

```
# f. Filter the data to include only plants with heights greater than 4 and widths greater than 2 (usin
filter(shrubs, height > 4 & width > 2)
```

```
##   site experiment length width height
## 1    1          2    2.1   2.2    7.6
## 2    4          2    4.5   4.8    6.5
```

```
# g. Filter the data to include only plants from Experiment 1 or Experiment 3 (using '|' for "or").
filter(shrubs, experiment == 1 | experiment == 3)
```

```
##    site experiment length width height
## 1     1          1    2.2   1.3    9.6
## 2     1          3    2.7   1.5    2.2
## 3     2          1    3.0   4.5    1.5
## 4     2          3    2.5   2.8    3.0
## 5     3          1    1.9   1.8    4.5
## 6     3          3    3.5   2.0    7.5
## 7     4          1    2.9   2.7    3.2
## 8     4          3    1.2   1.8    2.7
## 9     5          1    2.6   0.8     NA
## 10    5          3    3.1   2.2     NA
```

```
# h. Filter the data to remove rows with null values in the height column (using !is.na)
filter(shrubs, !is.na(height))
```

```
##    site experiment length width height
## 1     1          1    2.2   1.3    9.6
## 2     1          2    2.1   2.2    7.6
## 3     1          3    2.7   1.5    2.2
## 4     2          1    3.0   4.5    1.5
## 5     2          2    3.1   3.1    4.0
## 6     2          3    2.5   2.8    3.0
## 7     3          1    1.9   1.8    4.5
## 8     3          2    1.1   0.5    2.3
## 9     3          3    3.5   2.0    7.5
## 10    4          1    2.9   2.7    3.2
## 11    4          2    4.5   4.8    6.5
## 12    4          3    1.2   1.8    2.7
## 13    5          2    1.8    NA    5.2
```

```
# i. Create a new data frame called shrub_volumes that includes all of the original data and a new colu
shrub_volumes <- mutate(shrubs, volume = length * width * height)
shrub_volumes
```

```
##    site experiment length width height  volume
## 1     1          1    2.2   1.3    9.6  27.456
## 2     1          2    2.1   2.2    7.6  35.112
## 3     1          3    2.7   1.5    2.2   8.910
## 4     2          1    3.0   4.5    1.5  20.250
## 5     2          2    3.1   3.1    4.0  38.440
## 6     2          3    2.5   2.8    3.0  21.000
## 7     3          1    1.9   1.8    4.5  15.390
## 8     3          2    1.1   0.5    2.3   1.265
## 9     3          3    3.5   2.0    7.5  52.500
## 10    4          1    2.9   2.7    3.2  25.056
## 11    4          2    4.5   4.8    6.5 140.400
## 12    4          3    1.2   1.8    2.7   5.832
## 13    5          1    2.6   0.8     NA      NA
## 14    5          2    1.8    NA    5.2      NA
## 15    5          3    3.1   2.2     NA      NA
```

2. Code Shuffle (15 pts)

We are interested in understanding the monthly variation in precipitation in Tucson, AZ. We'll use some data from the NOAA National Climatic Data Center. Each row of the data is a year (from 2000-2023) and each column is a month (January - December).

Rearrange the following program so that it:

a. Imports the data
b. Calculates the mean precipitation (ppt) in each month across years
c. Plots the monthly averages as a simple line plot

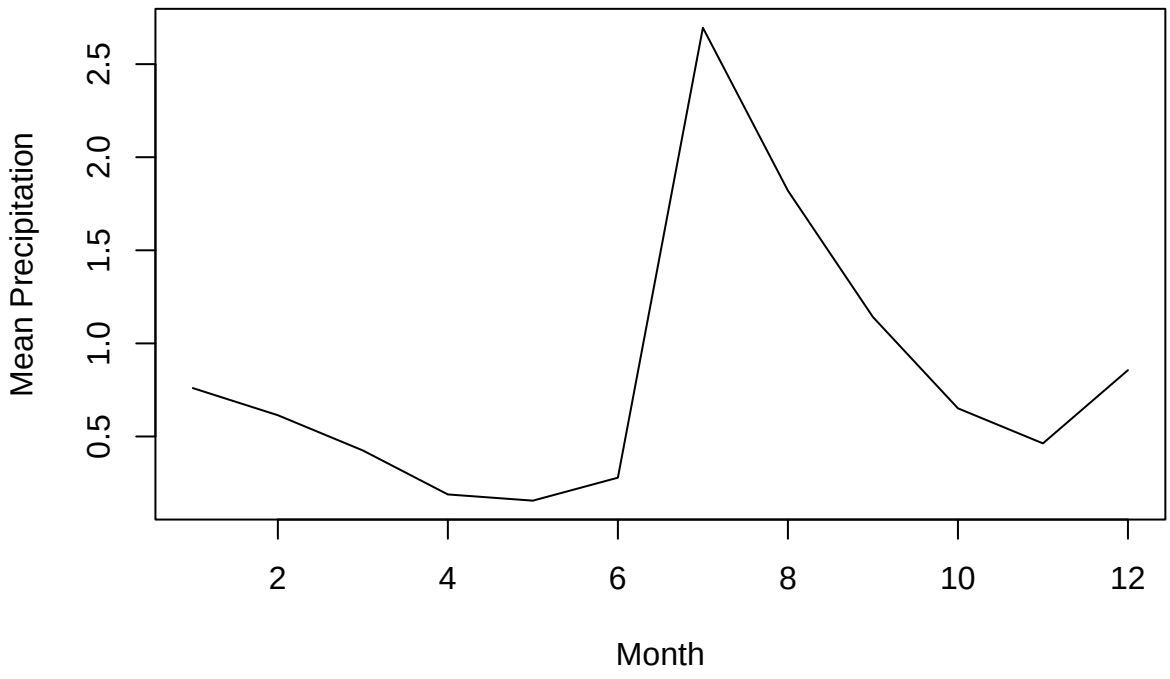
Finally, add a comment above the code that describes what it does. The comment character in R is `#`.

It's OK if you don't know exactly how the details of the program work at this point; you just need to figure out the right order of the lines based on when variables are defined and when they are used.

```r
# read in precip data
ppt_data <- read.csv("tucson_precip.csv", header = FALSE)

# calculate montly averages
monthly_mean_ppt <- colMeans(ppt_data)

# plot the monthly averages
plot(monthly_mean_ppt, type = "l", xlab = "Month", ylab = "Mean Precipitation")
```



3. Portal Data Manipulation (20 pts)

Load the `surveys.csv` file into R using `read.csv()`.

Note: Do not use pipes for this exercise.

a. Use `select()` to create a new data frame with just the year, month, day, and species_id columns in that order.
b. Use `mutate()`, `select()`, and `filter()` with `!is.na()` to create a new data frame with the year, species_id, and weight in kilograms of each individual, with no null weights. The weight in the table is given in grams so you will need to create a new column for weight in kilograms by dividing the weight column by 1000.
c. Use the `filter()` function to get all of the rows from the final data frame you created in b for the species ID `SH`.

```r
surveys <- read_csv("surveys.csv")
```

```
## Rows: 35549 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): species_id, sex
## dbl (7): record_id, month, day, year, plot_id, hindfoot_length, weight
##
## 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.
```

```r
# a
select(surveys, year, month, day, species_id)
```

```
## # A tibble: 35,549 x 4
##     year month   day species_id
##    <dbl> <dbl> <dbl> <chr>
##  1  1977     7    16 NL
##  2  1977     7    16 NL
##  3  1977     7    16 DM
##  4  1977     7    16 DM
##  5  1977     7    16 DM
##  6  1977     7    16 PF
##  7  1977     7    16 PE
##  8  1977     7    16 DM
##  9  1977     7    16 DM
## 10  1977     7    16 PF
## # i 35,539 more rows
```

```r
# b
select_cols <- select(surveys, year, species_id, weight)
filter_rows <- filter(select_cols, !is.na(weight))
weight_kg <- mutate(filter_rows, weight_kg = weight/1000)
weight_kg
```

```
## # A tibble: 32,283 x 4
##     year species_id weight weight_kg
##    <dbl> <chr>       <dbl>     <dbl>
##  1  1977 DM             40     0.04
##  2  1977 DM             48     0.048
##  3  1977 DM             29     0.029
##  4  1977 DM             46     0.046
##  5  1977 DM             36     0.036
##  6  1977 DO             52     0.052
##  7  1977 PF              8     0.008
##  8  1977 OX             22     0.022
##  9  1977 DM             35     0.035
## 10  1977 PF              7     0.007
## # i 32,273 more rows
```

```
# c
filter(weight_kg, species_id == "SH")
```

```
## # A tibble: 141 x 4
##     year species_id weight weight_kg
##    <dbl> <chr>       <dbl>     <dbl>
##  1  1978 SH             89     0.089
##  2  1982 SH            106     0.106
##  3  1982 SH             52     0.052
##  4  1986 SH             55     0.055
##  5  1987 SH             77     0.077
##  6  1987 SH             78     0.078
##  7  1987 SH            104     0.104
##  8  1987 SH             58     0.058
##  9  1987 SH             52     0.052
## 10  1988 SH             60     0.06
## # i 131 more rows
```

4. Portal Data Manipulation Pipes (20 pts)

Using the same data as you did in Exercise 3, use pipes (either `|>` or `%>%`) to combine the following operations to manipulate the data.

a. Use `mutate()`, `select()`, and `filter()` with `!is.na()` to create a new data frame with the year, species_id, and weight in kilograms of each individual, with no null weights.
b. Use the `filter()` and `select()` to get the year, month, day, and species_id columns for all of the rows in the data frame where species_id is `SH`.

```
# a.
surveys %>%
  select(year, species_id, weight) %>%
  filter(!is.na(weight)) %>%
  mutate(weight_kg = weight / 1000)
```

```
## # A tibble: 32,283 x 4
##     year species_id weight weight_kg
##    <dbl> <chr>       <dbl>     <dbl>
##  1  1977 DM             40     0.04
##  2  1977 DM             48     0.048
##  3  1977 DM             29     0.029
##  4  1977 DM             46     0.046
##  5  1977 DM             36     0.036
##  6  1977 DO             52     0.052
##  7  1977 PF              8     0.008
##  8  1977 OX             22     0.022
##  9  1977 DM             35     0.035
## 10  1977 PF              7     0.007
## # i 32,273 more rows
```

```
# b.
surveys %>%
  select(year, month, day, species_id) %>%
  filter(species_id == "SH")
```

```
## # A tibble: 147 x 4
##     year month   day species_id
##    <dbl> <dbl> <dbl> <chr>
##  1  1977     7    17 SH
##  2  1978    11     4 SH
##  3  1982     5    21 SH
##  4  1982     6    29 SH
##  5  1983     3    14 SH
##  6  1983     4    16 SH
##  7  1986    10     4 SH
##  8  1987     7    26 SH
##  9  1987     8    26 SH
## 10  1987    10    24 SH
## # i 137 more rows
```

5. Portal Data Challenge (20 pts)

Develop a data manipulation pipeline for the Portal surveys table that produces a table of data for only the three Dipodomys species (DM, DO, DS).

- The species IDs should be presented as lower case, not upper case.
- The table should contain information on the date, the species ID, the weight and hindfoot length.
- The data should not include null values for either weight or hindfoot length.
- The table should be sorted first by the species (so that each species is grouped together) and then by weight, with the largest weights at the top.

```
surveys %>%
  filter(species_id == "DO" | species_id == "DS" | species_id == "DM") %>%
  mutate(species_id = tolower(species_id)) %>%
  select(month, day, year, species_id, weight, hindfoot_length) %>%
  filter(!is.na(weight), !is.na(hindfoot_length)) %>%
  arrange(species_id, desc(weight))
```

```
## # A tibble: 14,558 x 6
##    month   day  year species_id weight hindfoot_length
##    <dbl> <dbl> <dbl> <chr>       <dbl>           <dbl>
##  1     4    29  1979 dm             65              37
##  2     8     7  1991 dm             65              37
##  3     5    16  2002 dm             64              35
##  4     5    13  1984 dm             63              35
##  5    12     3  1995 dm             63              38
##  6    10    12  1980 dm             62              35
##  7    10    28  1995 dm             62              37
##  8     1    28  1996 dm             62              38
##  9     1    28  1996 dm             62              38
## 10    11     7  1999 dm             62              36
## # i 14,548 more rows
```

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 you can't get your file to Knit to PDF, you can submit the .Rmd file instead.*)
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 the correct Assignment dropbox. Submit the PDF file that you just downloaded.