Week 4 Assignment

Ellen Bledsoe

2025-02-14

Assignment Details

Purpose

The goal of this assignment is to work with data aggregation and joining data frames together using `dplyr` functions.

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 12 at midnight MST

Assignment Exercises

1. Set-Up (10 points)

Load the `readr` and `dplyr` packages.

```
library(readr)
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
```

Read in the following data sets using `read_csv()`:

- `surveys.csv`
- `species.csv`
- `plots.csv`

```
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.
```

```
species <- read_csv("species.csv")
```

```
## Rows: 54 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): species_id, genus, species, taxa
##
## 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.
```

```
plots <- read_csv("plots.csv")
```

```
## Rows: 24 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): plot_type
## dbl (1): plot_id
##
## 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.
```

2. Portal Data Aggregation (10 pts)

Using the `surveys` data frame, complete the following:

a. Use the `group_by()` and `summarize()` functions to get a count of the number of individuals in each species ID.
b. Use the `group_by()` and `summarize()` functions to get a count of the number of individuals in each species ID in each year.
c. Use the `filter()`, `group_by()`, and `summarize()` functions to get the mean mass of species `DO` in each year.

```
# 1. Use the group_by() and summarize() functions to get a count of the number of individuals in each s

surveys %>%
  group_by(species_id) %>%
  summarize(count = n())
```

```
## # A tibble: 49 x 2
##    species_id count
##    <chr>      <int>
##  1 AB           303
##  2 AH           437
##  3 AS             2
##  4 BA            46
##  5 CB            50
##  6 CM            13
##  7 CQ            16
##  8 CS             1
##  9 CT             1
## 10 CU             1
## # i 39 more rows
```

```
# 2. Use the group_by() and summarize() functions to get a count of the number of individuals in each s

surveys %>%
  group_by(species_id, year) %>%
  summarize(count = n())
```

```
## `summarise()` has grouped output by 'species_id'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 535 x 3
## # Groups:   species_id [49]
##    species_id  year count
##    <chr>      <dbl> <int>
##  1 AB          1980     5
##  2 AB          1981     7
##  3 AB          1982    34
##  4 AB          1983    41
##  5 AB          1984    12
##  6 AB          1985    14
##  7 AB          1986     5
##  8 AB          1987    35
##  9 AB          1988    39
## 10 AB          1989    31
## # i 525 more rows
```

```
# 3. Use the filter(), group_by(), and summarize() functions to get the mean mass of species DO in each

 surveys %>%
  filter(species_id == "DO") %>%
  group_by(year) %>%
  summarize(avg_mass = mean(weight, na.rm = TRUE))
```

```
## # A tibble: 26 x 2
##     year avg_mass
##    <dbl>    <dbl>
##  1  1977     42.7
##  2  1978     45
##  3  1979     45.9
##  4  1980     48.1
##  5  1981     49.1
##  6  1982     47.9
```

```
##  7  1983     47.2
##  8  1984     48.4
##  9  1985     48.0
## 10  1986     49.4
## # i 16 more rows
```

3. Shrub Volume Aggregation (15 pts)

This is a follow-up to Shrub Volume Data Basics (from a previous assignment).

Dr. Morales wants some summary data of the plants at her sites and for her experiments. The following code calculates the average height of all of the plants:

```
shrub_dims <- read_csv('shrub-volume-data.csv')
```

```
## Rows: 15 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (5): site, experiment, length, width, height
##
## 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.
```

```
shrub_dims %>%
  summarize(avg_height = mean(height))
```

```
## # A tibble: 1 x 1
##   avg_height
##        <dbl>
## 1         NA
```

Note: to answer the following questions, I recommend copying the code above and pasting it into a new code chunk, leaving the original code in the code chunk above.

a. Modify the code to calculate and the average height of a plant in *each experiment*.

b. Modify the code you wrote for (a) to calculate the maximum plant height in each experiment in addition to the average height.

c. Modify the code you wrote for (b) to calculate the minimum, maximum, and average plant height at *each site*.

```
print("3a")
```

```
## [1] "3a"
```

```
shrub_dims %>%
  group_by(experiment) %>%
  summarize(avg_height = mean(height, na.rm = TRUE))
```

```
## # A tibble: 3 x 2
##   experiment avg_height
##        <dbl>      <dbl>
## 1          1       4.7
## 2          2       5.12
## 3          3       3.85
```

```
print("3b")
```

```
## [1] "3b"
```

```r
shrub_dims %>%
  group_by(experiment) %>%
  summarize(avg_height = mean(height, na.rm = TRUE),
            max_height = max(height, na.rm = TRUE))
```

```
## # A tibble: 3 x 3
##   experiment avg_height max_height
##        <dbl>      <dbl>      <dbl>
## 1          1       4.7         9.6
## 2          2       5.12        7.6
## 3          3       3.85        7.5
```

```r
print("3c")
```

```
## [1] "3c"
```

```r
shrub_dims %>%
  group_by(site) %>%
  summarize(avg_height = mean(height, na.rm = TRUE),
            max_height = max(height, na.rm = TRUE),
            min_height = min(height, na.rm = TRUE))
```

```
## # A tibble: 5 x 4
##    site avg_height max_height min_height
##   <dbl>      <dbl>      <dbl>      <dbl>
## 1     1       6.47        9.6        2.2
## 2     2       2.83        4          1.5
## 3     3       4.77        7.5        2.3
## 4     4       4.13        6.5        2.7
## 5     5       5.2         5.2        5.2
```

4. Portal Data Joins (15 pts)

Using the Portal data sets, do the following:

a. Use `inner_join()` to create a table that contains the information from both the surveys table and the species table.
b. Use `inner_join()` twice to create a table that contains the information from all three tables.
c. Use `inner_join()` and `filter()` to get a data frame with the information from the surveys and plots tables where the plot_type is Control.

```r
print("4a")
```

```
## [1] "4a"
```

```r
inner_join(surveys, species, by = "species_id")
```

```
## # A tibble: 34,786 x 12
##    record_id month   day  year plot_id species_id sex   hindfoot_length weight
##        <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>           <dbl>  <dbl>
##  1         1     7    16  1977       2 NL         M                  32     NA
##  2         2     7    16  1977       3 NL         M                  33     NA
##  3         3     7    16  1977       2 DM         F                  37     NA
##  4         4     7    16  1977       7 DM         M                  36     NA
##  5         5     7    16  1977       3 DM         M                  35     NA
##  6         6     7    16  1977       1 PF         M                  14     NA
##  7         7     7    16  1977       2 PE         F                  NA     NA
```

```
##  8         8     7    16  1977       1 DM         M                 37     NA
##  9         9     7    16  1977       1 DM         F                 34     NA
## 10        10     7    16  1977       6 PF         F                 20     NA
## # i 34,776 more rows
## # i 3 more variables: genus <chr>, species <chr>, taxa <chr>
```

```
print("4b")
```

```
## [1] "4b"
```

```
inner_join(surveys, species, by = "species_id") %>%
inner_join(., plots, by = "plot_id")
```

```
## # A tibble: 34,786 x 13
##    record_id month   day  year plot_id species_id sex   hindfoot_length weight
##        <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>           <dbl>  <dbl>
##  1         1     7    16  1977       2 NL         M                  32     NA
##  2         2     7    16  1977       3 NL         M                  33     NA
##  3         3     7    16  1977       2 DM         F                  37     NA
##  4         4     7    16  1977       7 DM         M                  36     NA
##  5         5     7    16  1977       3 DM         M                  35     NA
##  6         6     7    16  1977       1 PF         M                  14     NA
##  7         7     7    16  1977       2 PE         F                  NA     NA
##  8         8     7    16  1977       1 DM         M                  37     NA
##  9         9     7    16  1977       1 DM         F                  34     NA
## 10        10     7    16  1977       6 PF         F                  20     NA
## # i 34,776 more rows
## # i 4 more variables: genus <chr>, species <chr>, taxa <chr>, plot_type <chr>
```

```
print("4c")
```

```
## [1] "4c"
```

```
surveys %>%
  inner_join(plots, by = "plot_id") %>%
  filter(plot_type == "Control")
```

```
## # A tibble: 15,660 x 10
##    record_id month   day  year plot_id species_id sex   hindfoot_length weight
##        <dbl> <dbl> <dbl> <dbl>   <dbl> <chr>      <chr>           <dbl>  <dbl>
##  1         1     7    16  1977       2 NL         M                  32     NA
##  2         3     7    16  1977       2 DM         F                  37     NA
##  3         7     7    16  1977       2 PE         F                  NA     NA
##  4        14     7    16  1977       8 DM         <NA>               NA     NA
##  5        16     7    16  1977       4 DM         F                  36     NA
##  6        18     7    16  1977       2 PP         M                  22     NA
##  7        19     7    16  1977       4 PF         <NA>               NA     NA
##  8        20     7    17  1977      11 DS         F                  48     NA
##  9        21     7    17  1977      14 DM         F                  34     NA
## 10        28     7    17  1977      11 DM         M                  38     NA
## # i 15,650 more rows
## # i 1 more variable: plot_type <chr>
```

5. Portal Data dplyr Review (25 pts)

We want to do an analysis comparing the size of individuals on the Control plots to the Long-term Krat Exclosures.

a. Create a data frame with the `year`, `genus`, `species`, `weight` and `plot_type` for all cases where the plot type is either "Control" or "Long-term Krat Exclosure." Only include cases where Taxa is "Rodent." Remove any records where the weight is missing.
b. Now, use the "split, apply, combine" approach to calculate the minimum, maximum, and average weight per plot type per year.

```
print("5a")
```

```
## [1] "5a"
```

```
surveys %>%
  inner_join(species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  filter(plot_type == "Control" | plot_type == "Long-term Krat Exclosure") %>%
  filter(taxa == "Rodent") %>%
  select('year', 'genus', 'species', 'weight', 'plot_type') %>%
  filter(!is.na(weight))
```

```
## # A tibble: 19,344 x 5
##     year genus       species  weight plot_type
##    <dbl> <chr>       <chr>     <dbl> <chr>
##  1  1977 Dipodomys   merriami     40 Long-term Krat Exclosure
##  2  1977 Dipodomys   merriami     29 Control
##  3  1977 Dipodomys   merriami     46 Control
##  4  1977 Dipodomys   ordii        52 Control
##  5  1977 Perognathus flavus        8 Control
##  6  1977 Onychomys   sp.          22 Long-term Krat Exclosure
##  7  1977 Perognathus flavus        7 Control
##  8  1977 Dipodomys   merriami     22 Control
##  9  1977 Perognathus flavus        8 Control
## 10  1977 Dipodomys   merriami     41 Control
## # i 19,334 more rows
```

```
print("5b")
```

```
## [1] "5b"
```

```
surveys %>%
  inner_join(species, by = "species_id") %>%
  inner_join(plots, by = "plot_id") %>%
  filter(plot_type == "Control" | plot_type == "Long-term Krat Exclosure",
         taxa == "Rodent",
         !is.na(weight)) %>%
  select('year', 'genus', 'species', 'weight', 'plot_type') %>%
  group_by(year, plot_type) |>
  summarize(min_weight = min(weight),
            max_weight = max(weight),
            mean_weight = mean(weight))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 52 x 5
## # Groups:   year [26]
##     year plot_type               min_weight max_weight mean_weight
##    <dbl> <chr>                        <dbl>      <dbl>       <dbl>
##  1  1977 Control                          6        149        50.4
```

```
##  2  1977 Long-term Krat Exclosure          7         50        34.8
##  3  1978 Control                           6        223        70.8
##  4  1978 Long-term Krat Exclosure          6        232        35.9
##  5  1979 Control                           7        274        68.1
##  6  1979 Long-term Krat Exclosure          6        122        24.4
##  7  1980 Control                           5        214        66.2
##  8  1980 Long-term Krat Exclosure          5        155        26.5
##  9  1981 Control                           4        264        68.0
## 10  1981 Long-term Krat Exclosure          4        195        34.7
## # i 42 more rows
```

6. Shrub Volume Bind (10 pts)

First, run the following code chunk to produce a data frame with additional data related to the shrub volume data (`shrub_dims`) from Question 3.

```
new_data <- data.frame(respiratory_rate = c(2.2, 4.0, 6.1, 2.3, 4.1, 6.2, 1.8, 3.5, 5.7, 1.9, 3.5, 5.8,
                       average_temp_C = c(15.1, 20.2, 24.7, 15.2, 22.0, 25.1, 14.2, 19.0, 23.6, 14.9, 2
```

Take a look at the new dataframe that has just been produced. Should this data be bound to the shrub volume data by `bind_rows()` or `bind_cols()`? In your own words, explain how you know.

Answer:

Based on your answer above, bind the `shrub_dims` and `new_data` data frames together.

```
bind_cols(shrub_dims, new_data)
```

```
## # A tibble: 15 x 7
##     site experiment length width height respiratory_rate average_temp_C
##    <dbl>      <dbl>  <dbl> <dbl>  <dbl>            <dbl>          <dbl>
##  1     1          1    2.2   1.3    9.6              2.2           15.1
##  2     1          2    2.1   2.2    7.6              4             20.2
##  3     1          3    2.7   1.5    2.2              6.1           24.7
##  4     2          1    3     4.5    1.5              2.3           15.2
##  5     2          2    3.1   3.1    4                4.1           22
##  6     2          3    2.5   2.8    3                6.2           25.1
##  7     3          1    1.9   1.8    4.5              1.8           14.2
##  8     3          2    1.1   0.5    2.3              3.5           19
##  9     3          3    3.5   2      7.5              5.7           23.6
## 10     4          1    2.9   2.7    3.2              1.9           14.9
## 11     4          2    4.5   4.8    6.5              3.5           20.3
## 12     4          3    1.2   1.8    2.7              5.8           24.1
## 13     5          1    2.6   0.8   NA                2             19.2
## 14     5          2    1.8  NA      5.2              4.7           22.7
## 15     5          3    3.1   2.2   NA                6.2           25
```

7. Shrub Volume Join (15 pts)

This is a follow-up to Question 3, Shrub Volume Aggregation.

In addition to the main data table on shrub dimensions, Dr. Morales has two additional data tables. The first describes the manipulation for each experiment. The second provides information about the different sites. Run the following code chunk to bring them into your environment.

```
experiments <- read_csv("https://datacarpentry.org/semester-biology/data/shrub-volume-experiments.csv")
```

```
## Rows: 3 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): manipulation
## dbl (1): experiment
##
## 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.
```

```
sites <- read_csv("https://datacarpentry.org/semester-biology/data/shrub-volume-sites.csv")
```

```
## Rows: 4 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): site, latitude, longitude, elevation
##
## 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.
```

a. Combine the `experiments` data frame with the shrub dimensions data to add a manipulation column to the shrub data.

b. Next, combine the `sites` data frame with both the data on shrub dimensions and the data on experiments to produce a single data frame that contains all of the data. Save this data frame as `shrub_data`.

```
# grading note; if someone saved over the `shrubs_dim` dataframe in question 3, then they might have a

# 1. Import the experiments data and then use `inner_join` to combine
print("7a")
```

```
## [1] "7a"
```

```
inner_join(shrub_dims, experiments, by = "experiment")
```

```
## # A tibble: 15 x 6
##     site experiment length width height manipulation
##    <dbl>      <dbl>  <dbl> <dbl>  <dbl> <chr>
##  1     1          1    2.2   1.3    9.6 control
##  2     1          2    2.1   2.2    7.6 burn
##  3     1          3    2.7   1.5    2.2 rainout
##  4     2          1    3     4.5    1.5 control
##  5     2          2    3.1   3.1    4   burn
##  6     2          3    2.5   2.8    3   rainout
##  7     3          1    1.9   1.8    4.5 control
##  8     3          2    1.1   0.5    2.3 burn
##  9     3          3    3.5   2      7.5 rainout
## 10     4          1    2.9   2.7    3.2 control
## 11     4          2    4.5   4.8    6.5 burn
## 12     4          3    1.2   1.8    2.7 rainout
## 13     5          1    2.6   0.8   NA   control
## 14     5          2    1.8  NA      5.2 burn
## 15     5          3    3.1   2.2   NA   rainout
```

```
# 2. Import the sites data and the combine it with both the data on shrub
# dimensions and the data on experiments to produce a single data frame
# that contains all of the data.
print("7b")
```

```
## [1] "7b"
```

```r
shrub_dims_experiments <- inner_join(shrub_dims, experiments, by = "experiment")
shrub_data <- inner_join(shrub_dims_experiments, sites, by = "site")
shrub_data
```

```
## # A tibble: 12 x 9
##     site experiment length width height manipulation latitude longitude
##    <dbl>      <dbl>  <dbl> <dbl>  <dbl> <chr>           <dbl>     <dbl>
##  1     1          1    2.2   1.3    9.6 control          29.6     -82.3
##  2     1          2    2.1   2.2    7.6 burn             29.6     -82.3
##  3     1          3    2.7   1.5    2.2 rainout          29.6     -82.3
##  4     2          1    3     4.5    1.5 control          29.3     -82.4
##  5     2          2    3.1   3.1    4   burn             29.3     -82.4
##  6     2          3    2.5   2.8    3   rainout          29.3     -82.4
##  7     3          1    1.9   1.8    4.5 control          29.8     -82.2
##  8     3          2    1.1   0.5    2.3 burn             29.8     -82.2
##  9     3          3    3.5   2      7.5 rainout          29.8     -82.2
## 10     4          1    2.9   2.7    3.2 control          30.0     -82.6
## 11     4          2    4.5   4.8    6.5 burn             30.0     -82.6
## 12     4          3    1.2   1.8    2.7 rainout          30.0     -82.6
## # i 1 more variable: elevation <dbl>
```