Advanced Data Wrangling with `dplyr`

FOR 128: Lab 8

Insert Your Name Here

2024-10-30

Welcome

Welcome to Lab 8! Today, we'll focus on writing extensive `dplyr` code on larger data. We will both use the verbs individually and "write a sentence" with the verbs by stringing them together with pipes.

Learning objectives

- Use `dplyr` verbs together with pipes on larger datasets.

Deliverables (i.e., what to put in the lab drop box)

Upload your rendered PDF (`lab_08.pdf`) **and** Quarto (`lab_08.qmd`) document to the lab drop box. Make sure the Quarto document properly renders to PDF.

Collaborator(s)

List any collaborators you worked with below.

Introduction

Data: `pdxTrees`

The `pdxTrees` R package contains measurements of every tree in the Portland, Oregon metro area. In particular, it contains two datasets, which we will call `parks` and `streets`. The `parks` dataset contains all of the trees in 174 parks in Portland. The `streets` dataset contains all

the trees on the streets of Portland. The hex sticker for `pdxTrees` (as seen below) contains a fun easter egg from the city of Portland, as its graphics are done in a similar fashion to the famous airport carpet.

Methods: `dplyr`

Use `dplyr` function that we've learned over the past few weeks to answer the questions in this lab.

Exercise 0

Load any packages you'll need for this lab below. Note, you'll need to install `pdxTrees` with `install.packages("pdxTrees")` before you can load it with `library(pdxTrees)`.

Exercise 1

Part (a)

Load the data. For this lab, we will load both the `parks` and `streets` datasets. To do so, run the `get_pdxTrees_parks()` and `get_pdxTrees_streets()` functions (note, you don't

need to specify any arguments for these functions) and assign them to `parks` and `streets`, respectively.

Part (b)

Take a look at the documentation by running `?get_pdxTrees_parks` and `?get_pdxTrees_streets` in your console, and tell me about two columns (what are their names? what do they measure? units? factor levels? etc.) from each of the datasets (four total).

Exercise 2

Find the top 8 most common species (use the common name column for species here and throughout the lab) in the `parks` dataset, along with how many of that species are in the dataset. Notice I've called the number of trees `num_trees`.

Exercise 3

A staple of Portland living in the springtime is to go out and see the Japanese Flowering Cherry trees beginning to bloom. Find the park with the most Japanese Flowering Cherry trees, and how many there are in that park.

Exercise 4

Find the top 5 parks with the most total DBH (i.e. the sum of all the trees DBH in those parks).

Exercise 5

There is a column in the `parks` dataset that represents the amount of carbon (in lbs) each tree has sequestered. Find the park with the highest carbon sequestration.

Exercise 6

The `parks` dataset has a wide variety of numeric columns measuring things from DBH to pollution removal value of a given tree. Take the mean of all of these numeric columns, grouped by park. Make sure to remove NAs in the `mean()` function.

HINT: In `across()` you can set `.cols = where(is.numeric)` to apply a function across all numeric columns.

Exercise 7

For this exercise, consider the `streets` dataset.

Part (a)

There is a column delineating the edibility of the trees in the dataset. Find the possible values for this column and how many trees correspond to each value.

Part (b)

Create a new tibble called `forager` that contains all rows of the `streets` with edible fruits or nuts but only the columns denoting the common name, neighborhood, condition, and edibility.

Part (c)

Create a new logical column and save it in the `forager` dataset. Name it `fruit_edible`. The values in this columns should be `TRUE` or `FALSE` depending on if the given tree has edible fruits or not

Part (d)

Arrange the forager dataset by the column you just created in (c), so that the `TRUE` values are at the top, and save this change to the `forager` dataset.

Part (e)

The `Edible` column in named in a bit of a funny way. Rename the column to `Edible_Component` and save this change to the `forager` dataset.

Part (f)

Which five neighborhoods would a fruit loving forager most like to live (in other words, what are the five neighborhoods with the most trees with edible fruit)?

HINT: using the `sum()` function on logical data will make all the `TRUE`s `1`s and all the `FALSE`s `0`s.

Part (g)

Of the five neighborhoods found in part (f), what are the conditions of the edible fruit bearing trees (in other words, how many are in Good condition, Fair condition, etc., grouped by these five neighborhoods).

HINT: you can use `group_by()` to group by multiple groups at once. Example: `group_by(group1, group2)`.

Wrap up

Congratulations! You've made it to the end of Lab 8. Make sure to render your final document and submit both the .pdf and .qmd file to D2L.