STAT 29000 Project 10

Topics: Introduction to tidyverse, part 2

Motivation: Although the base R functions and the `tidyverse` functions are both very popular, there are still further alternatives for wrangling data using R. (Remember that there are approximately 15,000 packages in R at present.) We are going to learn about a package called `data.table` and we will continue to hone our `ggplot2` skills.

Context: `Data.table` inherits properties from `data.frames`, but offers more functions. Many of these functions are optimzied to be extremely fast. It is worthwhile to have several ways to attack problems, so that our data analysis is versatible and well informed.

Scope: `Data.table` is especially helpful when working with huge data sets. (The documentation especially recommends it for data bigger than 100 GB.) Additionally, `data.table` is natural to use with `ggplot2`.

The full, official documentation about `data.table` is available online here:

https://cran.r-project.org/web/packages/data.table/data.table.pdf.

Question 1: Reading in data, and using data.table

1a. Read over the official documentation linked above. Name the three parts of the general form of data.table, and explain briefly what they do. Like the `readr` package in `tidyverse` (https://readr.tidyverse.org/), the `data.table` package comes with its own function to read in data. Read the documentation; what is the name of this function?

1b. Use the `microbenchmark` package to compare and contrast the performace of the function from **(1a)** and the `read_csv` function in `readr`. Which package is more faster, and how much faster, looking at its mean time performance? (You are welcome to use the function's default values; however, we recommend changing the number of times the functions are tested from its default of 100 times to (instead) 10 times.)

Test the function from **(1a)** versus the `read_csv` function on the following file:

`/class/datamine/data/8451/The_Complete_Journey_2_Master/5000_transactions.csv`

Warning: This benchmark will take a few minutes; please consider brewing some coffee while you wait.

Hint: It may be useful to run `options(readr.num_columns = 0)` *to suppress* `readr` *warnings.*

1c. Use the function from **(1a)** to read in the data found at

https://raw.githubusercontent.com/fastforwardlabs/couples-lime/master/couples.csv

into a variable named `couples`. What classes does our new object inherit from? What are the first five rows of `couples`?

Question 2: Working with a data.table

For the following problems, limit yourself to the `data.table` package and the `stringr` package from tidyverse.

2a. Characterize the number of times that partners visit relatives, according to education level. What education level visits home most often, on average?

2b. What combination of political party and ethnicity visits home most often, on average?

2c. There are many situations when we want to get many summaries of our data at once. Show the summary statistics (min, 1st quartile, median, mean, 3rd quartile and max) for number of children, according to political party.

Question 3: Lots of plots

3a. Use `ggplot` to create a density plot for age by housing. Use `fill` to color the inside of the densities. Filter out all housing (i.e., remove all housing values) except the value `a one-family house detached from any other house` and the value `a building with 2 or more apartments`. Use the `adjust` option in the density function to smooth the distribution. (`adjust = 1.5`)

Hint: The `alpha` value in `ggplot` controls the transparency. It is useful when comparing densities plotted over the same region. If you want to get fancy with your `ggplot`, consider using the function `dollar_format` from the package `scales` inside the function `scale_x_continuous` from `ggplot2`.

3b. Plots are a useful tool for not only getting a sense of the relationships in the data, but also to do some sanity checks and verify the data. Similarly to **3a**, plot age densities by work. Set the `adjust` option to 1.5.

Hint: Remember the transparency value `alpha` in `ggplot` to control the transparency. It is useful when comparing densities plotted over the same region.

Project Submission:

Submit your solutions for the project at this URL: https://classroom.github.com/a/CRYQzdCp using the instructions found in the GitHub Classroom instructions folder on Blackboard.