DSSA 5001
Problem Set 1
Due 30 September 2024

(1) A common task in data science involves scraping data from a web site, cleaning it, and then analyzing it. For data that is displayed as a relatively small table, copying and pasting the table into a plain text file is an easy way to scrape the data (you'll learn more sophisticated methods in DSSA 5102: Data Gathering and Warehousing in the spring semester).

Writing pseudocode for cleaning a list of riders registered for the 2024 Midnight Sun Randonnée is the goal of this problem. Start by navigating to the list of riders to acquaint yourself with the information that is contained in the table.

The list of riders, as a tab-separated (TSV) file, can be found in the Problem Sets/PS01 folder on Blackboard. Learning to work with TSV files complements learning to work with CSV files because TSV files often result when copying and pasting from a spreadsheet or a web page. tidyverse provides read_tsv as a companion to read_csv.

What cleaning entails depends on the format of the data and the intended analysis. A small R script that reads the list of riders from the TSV file into a tibble, PS01_cleanTSV.R, can be found in the Problem Sets/PS01 folder on Blackboard. Run the script and enter View(MSR2024riders) in the RStudio Console pane to acquaint yourself with how the data was formatted upon copying and pasting it from the web page into the TSV file.

(a) Some rider names, club names, and city names are in all caps or have no capitalization. This may be a problem when alphabetizing the list. Add pseudocode to PS01_cleanTSV.R that describes what you would do to ensure that all last names, first names, club names, and city names follow the convention of capitalizing only the first letter of each word in the name.

(b) The names of the countries are duplicated in each element of MSR2024riders$Country. This results from the flag icons being replaced by the names of the countries when the data is pasted into the TSV file. Add pseudocode to PS01_cleanTSV.R that describes what you would do so that the names of the countries appear just once in each element of MSR2024riders$Country.

(2) A TCX (Training Center XML) file is a common format for encoding exercise data recorded by a device like a Fitbit. A TCX file is an XML (Extensible Markup Language) file whose rules were developed by Garmin. The exercise activity is recorded as a GPS track. Times at which data is recorded are known as trackpoints. Each trackpoint includes time, latitude, longitude, altitude, and distance.

Writing pseudocode for extracting altitude and distance, with the intention of graphing altitude vs distance to produce a profile of the exercise course, is the goal of this problem. The TCX file you will use for this problem is named GRR200K.tcx. It is in the Problem Sets/PS01 folder on Blackboard.

A TCX file is a plain text file formatted by XML tags and line breaks. Because a TCX file can be large (GRR200K.tcx has 72,652 lines) and its lines are not delimited like the lines of a CSV or TSV file, reading the entire file into R is not the best way to proceed. One option for exploring a TCX file is to open it in a web browser. After you've opened the file, scroll

down from the first line to identify how a point on the track is formatted. You'll use this information to write your pseudocode.

The specification of the task is to extract the altitude and distance along the track in a TCX file whose name is given. The file contains many lines unrelated to the track, so your pseudocode should read no more of the file than necessary. Identifying the tags that define the start and end of a track will help in this regard. Write your pseudocode in the RStudio Source pane as comment lines. Save the script as PS01_parseTCX.R.

(3) Designing an effective graph, as you're learning in DSSA 5103: Data Visualization, nearly always requires changing the default appearance of the graph. Two graphs are shown in the accompanying figure on the next page. The upper graph, labeled (a), is produced by Excel. The lower graph, labeled (b), is produced by Python. Write pseudocode that makes all the changes needed to convert the Excel graph into the Python graph. Some changes will be deletions, some will be additions, and some will be modifications to a graph element.

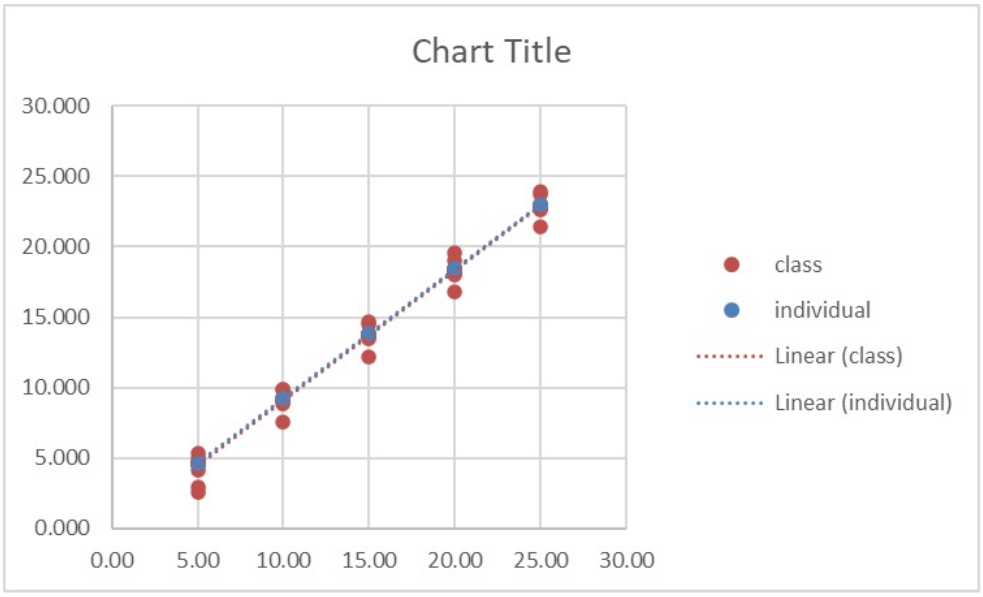
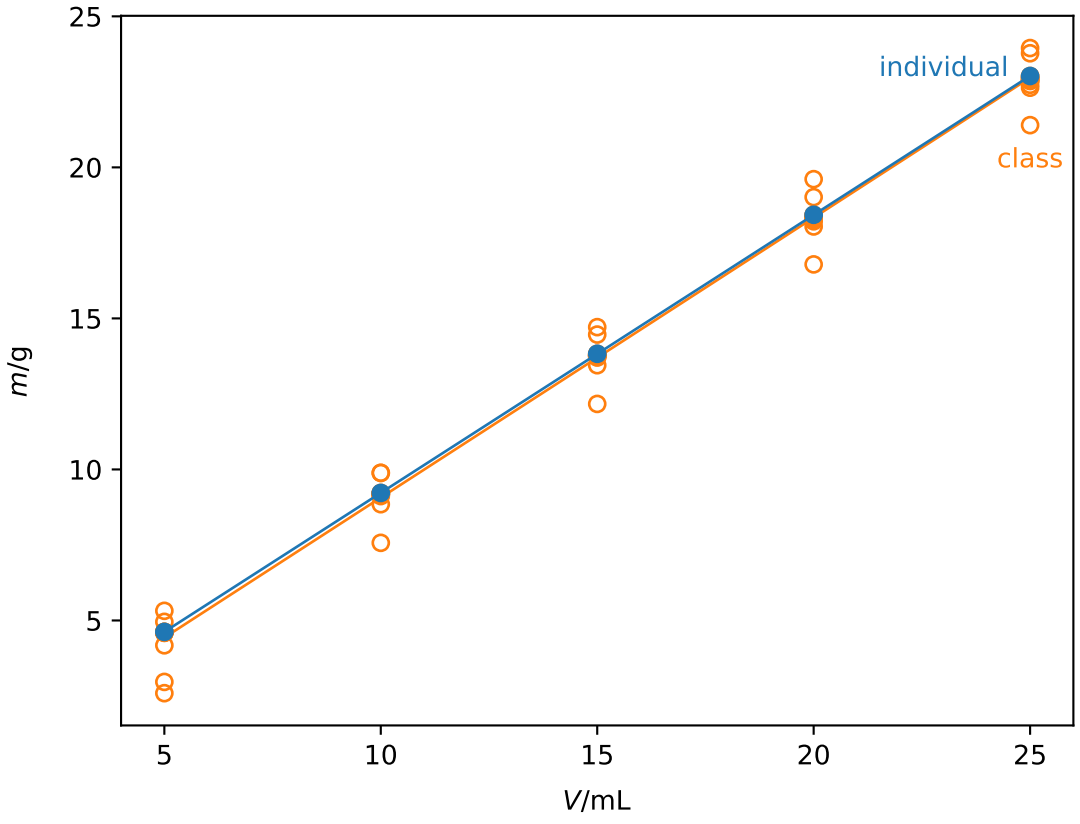


Chart Title
30.000
25.000
20.000
15.000
10.000
5.000
0.000
0.00
5.00
10.00
15.00
20.00
25.00
30.00
class
individual
Linear (class)
Linear (individual)

(a)


25
20
15
10
5
m/g
individual
class
5
10
15
20
25
V/mL

(b)