Data Comm Science - Midterm problem set

2023-10-21

WARNING: Your answer to this take-home assignment may be scored zero if someone other than you assisted or took the assignment, and/or if there is reasonable evidence that the test was taken in an inappropriate manner (e.g., you solve the questions in a group). **You SHOULD NOT DISCUSS YOUR ANSWER WITH OTHER CLASSMATES – you should provide your own, independent answer to these questions.** You will be subject to further disciplinary action in case you violate this rule.

First, please enter the following command into your Rstudio script and/or console pane. You can copy and paste the code:

```
library(data.table)
library(tidyverse)
options(scipen = 999)

gapdata <- read_csv("http://bit.ly/46LXJy6")
```

- Please answer the following questions using `data.table` and/or `tidyverse` way of writing syntax.
- When appropriate, you can enter any comments on your code by putting `#` sign and then able to provide further information about your code.
- You have a total of 8 questions (Q1 - Q8), 12.5pt each unless otherwise noted, and two bonus question (Q9 & Q10) worth 12.5 each.

Q1 (5pt):

- Using the `continent` variable in `gapdata`, select any cases that belongs to `Europe` (based on the `continent` variable) AND year 2007 (based on the `year` variable).
- Create an object named `gapdata_EU07` that stores any cases that satisfy these criteria.

Q2.

- Using `gapdata_EU07` object you created above, let's create `Gdp_per_Exp` variable as following: population size (`pop`) multiplied by GDP per person (`gdpPercap`), and then divide it by life expectancy (`lifeExp`).
- This value therefore represent the yearly expected value of the GDP of a person in that country.

Q3.

- Find which country has the highest GDP per person (`gdpPercap`) value in the `gapdata_EU07` object.

Q4.

- Now, using `gapdata` (NOT `gapdata_EU07`), please find the average life expectancy (`lifeExp`) variable only for 1997, and save this value under the object name `mean_expt_97`.

** Tip: If you are using `tidyverse` way, you need to add `%>% unlist` at the end of your code in order to properly print actual value of the cell.

Q5.

- Now, using `mean_expt_97` value you created above, within `gapdata` object please make new variable called `less_than_expt97`, such that cases (countries) that have life expectancy (`lifeExp`) values *strictly less than* `mean_expt_97` are coded as `1` in this variable, and `0` for otherwise.
- Please use `ifelse` statement in creating this variable.

Q6.

- Using `gapdata`, please create a table summarizing `less_than_expt97` variable per continent.
- Resulting table should have two columns, where first column contains continent name, and the second column stores the number of cases (countries).

** Tip: You can use `count` or `length` function to get the number of cases in a given vector.

Q7.

- Using `gapdata_EU07` object (NOT `gapdata`), make use of `apply` function to calculate the mean (using `mean`) and sd (using `sd`) of following variables: `pop`, `lifeExp`, `gdpPercap`, and `Gdp_per_Exp`.

Q8 (20pt).

- Now, create your own function (aka. custom function) that returns `year` value of each country based on highest `gdpPercap` in `gapdata` object.
- Name your own function with `my_fun` and set the two input parameters of your `my_fun` function as `dat` (representing data object, like `gapdata`) and `country_name` (representing country name string, like `country` variable in `gapdata`).
- Your `dat` data object in this function assumes to have `country`, `year`, and `gdpPercap` variables.

Q9 (bonus: 12.5pt):

- Using `my_fun` function you created in Q8, please find `year` of each county of their highest `gdpPercap` value in `gapdata` object.

Q10 (bonus: 12.5pt).

- Execute the following code first:

```
suppressWarnings(if (!require(psych)) install.packages("psych"))
set.seed(42)
dat2 <- data.frame(id = 1:1000,
                   group_name = sample(letters[1:10],
                                       size = 1000, replace = T),
                   psych::sim.hierarchical(n = 1000, raw = T)$observed) %>% as_tibble
```

- The `dat2` data you created above should contain `id`, `group_name`, and variables called `V1` until `V9` per each observation.
- Create a new variable called `irv` for each observation `i` in a way that this new variable represents a standard deviation (`sd`) of variables `V1`, `V2`, `V3`, … `V9` per each observation, formally defined as below:

$$irv_i = sd(V1_i, V2_i, V3_i, V4_i, V5_i, V6_i, V7_i, V8_i, V9_i)$$