HW4 Instructions

Data Science for Biologists, Spring 2020

R script due Thursday 10/8/20 by 12:01 PM to Canvas

Instructions

For your assignment, you will be asking and answering scientific questions about four datasets we have previously seen: `olives`, `urine`, `wine`, and `pima`. Please use the provided `hw4.R` R script template in the class RStudio Cloud Workspace.

For each dataset:

- Ask a scientific questions about the data. "Scientific question" means asking about trends in the data, as opposed to asking more "logistical" questions like "which columns are numeric are categorical?"
- Create a plot *of your choosing and styling* that addresses the question. Each plot should be saved to an output folder called `figures/`.
    - Plot files should be named as `<dataset>_figure.png` (e.g., `olives_figure.png`).
    - You do NOT need to upload your saved figures as part of the submission. Your code just needs to save the plots properly.
- In 1-2 sentences written as comments, answer the question you asked. Interpretations that do not address your question will be considered unanswered.
    - *As is templated for you*, the question/answer comments should be ABOVE your plotting code.

Please read these additional instructions carefully:

- CODE MUST BE PRESENT FOR READING FILES AND SAVING PLOTS. NO CREDIT WILL BE GIVEN FOR ANY PLOT WHERE YOU DO NOT HAVE ASSOCIATED CODE TO PERFORM ALL TASKS.
    - NEVER CLICK THE "IMPORT DATASET" BUTTON. YOU MUST USE `read_csv()`.
    - NEVER CLICK THE "EXPORT" BUTTON IN THE "Plots" PANE. YOU MUST USE `ggsave()`.
- All plots must be professionally labeled with clean X and Y axes. You are not required to include a title, subtitle, or caption, but you may include one or more of these labels if you choose.
- DO NOT RE-MAKE THE SAME PLOTS YOU SAW ON HW #2!! Plots here should be independent of previous assignments and address questions you are asking here.
- All plots using aesthetic color/fill mappings MUST USE a non-default color scale. Similarly, all legend titles must be professionally re-named from their default.
- All plots must be clearly legible with an appropriate aspect ratio. This means you may need to change the output plot size using arguments `width` and `height` when calling `ggsave()`. *Make extra sure to LOOK AT YOUR SAVED PLOTS* and tweak the size until it looks clearly professional.
    - You may also need to modify certain theme components to achieve a professional look!
- Each plot must use a different geom.
- Set a single default theme using `theme_set()` at the top of your script. You should choose a theme you like the most, ensuring that all plots look nice on that theme. Of course, you can always modify components of the theme for each individual plot as described.
- Ask me for help AT LEAST 24 hours before the deadline. I cannot guarantee time to help you if only short notice is given. Given my schedule, I can probably (unfortunately) guarantee that I won't have time to help.
- Write all code yourself and include COMMENTS for each plot you create.

- Please *comment out* (add `#` in front of) code that causes errors or bugs. This will ensure that your code runs without errors, while also showing me your code attempt if you only got part-way through recreating the plot.

Dataset documentation

`olives.csv`

This dataset contains information about 572 olives collected from different regions across Italy. The dataset contains information about: a) what region the olive was from, b) what smaller area was the the olive from, c) what are the percentages of different fatty acids in the olive's oil profile.

Variables include:

- `region` : General region of Italy.
- `area` : Area of Italy.
- *All remaining variables are the percentage of the given fatty acid in the olive.*

`urine.csv`

This dataset contains urinalysis measurements (don't worry about units) from 78 men, indicating whether traces of kidney stones (aka "crystal") were detected their urine samples.

Variables include:

- `crystal` : Whether calcium oxalate crystals (kidney stones) were detected
- `gravity` : The specific gravity of the urine
- `ph` : The pH of the urine
- `osmo` : The osmolarity of the urine. Osmolarity is proportional to the concentration of molecules in solution.
- `conduct` : The conductivity of the urine. Conductivity is proportional to the concentration of charged ions in solution.
- `urea` : The urea concentration in millimoles per liter
- `calcium` : The calcium concentration in millimoles per liter

`wine.csv`

This dataset contains information from a chemical analysis of three different cultivars (A, B, and C) of wine, including alcohol percentage and amounts of different chemical components.

Variables include:

- `Cultivar` : The wine cultivar (A, B, or C)
- `Alcohol` : The alcohol percentage of the wine
- `MalicAcid` : The percentage of the wine that is malic acid
- `MalicAcid` : The percentage of the wine that is malic acid

- `Ash` : The percentage of the wine that is ash (it's a wine thing…)
- `Magnesium` : The percentage of the wine that is magnesium
- `TotalPhenol` : The percentage of the wine that is phenols
- `Flavanoids` : The percentage of the wine that is flavanoids
- `NonflavPhenols` : The percentage of the wine that is non-flavanoid phenols
- `Color` : The color intensity of the wine, measured numerically

`pima.csv`

This dataset contains physical measurements from Pima Indian women from the American southwest. This population has been heavily studied by epidemiologists since they tend to have high levels of Diabetes.

Variables include:

- `npreg` : number of times the woman was pregnant
- `glucose` : plasma glucose concentration at 2 hours in an oral glucose tolerance test (units: mg/dL)
- `dpb` : diastolic blood pressure (units: mm Hg)
- `skin` : triceps skin-fold thickness (units: mm)
- `insulin` : 2-hour serum insulin level (units: μU/mL)
- `bmi` : Body Mass Index
- `age` : age in years
- `diabetic` : whether or not the individual has diabetes

Example

This example shows you the general concept of what to do using the `iris` dataset. **You are not meant to replicate my exact work. You should use this as a guiding example to *follow your own path to coding!***. I highly recommend running this code yourself to make sure you understand what *each aspect of the code* does!!!!

Assume that earlier in code, `theme_set(theme_classic())` was run.

```r
# iris plot ------------------------------------------------

# Scientific question: Which species has the longest average petal width?
# Answer: Virginica has the largest average petal width, roughly at 2 cm.

# First, make sure you explore the data interactively (but comment out or remove explora
       tion code from script!)
# head(iris)

# Make plot and save plot to variable
ggplot(iris, aes(x = Petal.Width, fill = Species)) +
  geom_density(alpha = 0.5) +
  # Set fill scale and color scales
  scale_fill_brewer(palette = "Reds", name = "Iris species") +
  # label axes
  labs(x = "Petal Width (cm)", y = "Density") +
  # spielman's visual preference: move theme to bottom
  # plot has really large legend keys, so might want to make them a ltitle smaller
  theme(legend.position = "bottom",
        legend.key.size = unit(0.5, "cm")) -> iris_figure

# Save plot to file
ggsave("figures/iris_figure.png", width = 8, height = 4)
```