Research Skills: Programming with R

Assignment 1

This graded set of homework assignments must be handed in on Canvas before *Monday, May 11th, 21.00 PM.* It tests your mastery of Worksheets 1 to 3 and R programming style. You will be asked to manipulate, summarize and plot data with "base R", "dplyr and "ggplot2". The answers are included in the .Rmd version of this file.

The assignment will be graded as follows:

- 0.5 point each for Questions 1 through 5 (2.5 points)
- 1.0 points each for Questions 6 through 8 (3 points)
- 1.5 points each for Questions 9 through 10 (3 points)
- 1.0 point in total for overall code organisation & style
- 0.5 point in total for complying with the instructions below

The guidelines for overall code organisation & style can be found in the slides for Class 4. Note that to receive full marks for this aspect you will have to make use of the `%>%` operator where applicable, also as explained in Class 4.

The questions will be graded semi-automatically. Answer them exactly as asked, no deviations or elaborations. All questions are independent (except 6 & 7); copy the data set before modifying it, and start afresh with the original each time. Points available for Q7 are independent of the dataframe returned by Q6. If you are unable to solve Q6, you may use the `spells` dataframe for your solution.

Other instructions:

- solve all the questions in a single R script
- use `Assigment_1_DemoScript.R`, from Canvas, as the basis of this script
- load the data exactly as shown in this demo; do not adapt the relative path
- use any function from 'base R', `dplyr` and `ggplot2`, and no other packages
- name your script `assignment1.R`; Canvas will add your U-number automatically
- include your name and u-number at the top of your script
- store your solutions in the objects described

This is an individual assignment: You may discuss it with your fellow students in general terms but do not share code. Suspected plagiarism will be referred to the Exam Board. Good luck!

Data Set Information

This assignment concerns a data set called `spells`, which contains all magic spells in the fantasy roleplaying game Dungeons & Dragons 5th edition basic rulebook. Source: https://github.com/tadzik/5e-spells. Most variables are self-explanatory; a few are explained below.

The `Casting Time` columns refer to the time it takes a magic user to complete the magical spells. For the purposes of ordering time, a `reaction` is faster than a `bonus action`, which in turn is faster than a (standard) `action`. An `action` takes 6 seconds.

In the `Components` columns, `V` stands for Verbal, `S` stands for Somatic and `M` stands for Material. Any explanation of the Material component is given in parentheses "()".

The `Duration` columns specifies how long the magical effect lasts. For the purposes of ordering, `Instantaneous` is the shortest duration, an `Concentration` attribute does not alter the duration, "up to 1 day" is shorter than 1 day (etc.) and `1 round` equals 6 seconds. `Until dispelled` could be forever.

For the `level` variable, "0" represents the **lowest** level, and "9" represents the **highest** level.

For the `range` variable, ignore any specifications for the "Self" entries. For the purposes of ordering, `Self` is closer than `Touch`, which in turn is closer than any conventional distance marking used in the dataset.

Question 1 (0.5 points).

Create an object that's a copy of `spells`, but contains only the spells of 4th level and above, and omits the `description` columns as well as all columns that start with a "c" (without calling these "c" columns directly by name). Create this object with a meaningful name initially, then copy it into an object called `answer1`.

Question 2 (0.5 points).

Create an object that is a copy of `spells`, but with a new column called `number_of_components`. This *numeric* column should contain, for all spells, the total number of component **categories** (V,S and/or M) required. For example, a spell with V,S and M components would get be a "3". Create this object with a meaningful name initially, then copy it into an object called `answer2`.

Question 3 (0.5 points).

Create an object that contains, for each School of spells, for each spell level **up to and including level 7**, the total number of spells at this level in this school of magic. Its columns should be called `school`, `level`, `spells_per_level` respectively. Sort the data so that the the school with the most 7th level spells ends up on top. Create this object with a meaningful name initially, then copy it into an object called `answer3`.

Question 4 (0.5 points).

Create an object that holds the name of the alphabetically first 3 spells per spell level, excluding "Abjuration" and "Enchantment" spells, organised by ascending level. Create this object with a meaningful name initially, then copy it into an object called `answer4`.

Question 5 (0.5 points).

Create a variable that indexes whether a spell includes any material component, and a new **factor** variable that indexes whether a spell has a level lower than 4th. Create a side-by-side barplot that shows, per the categories of this second newly created variable, the number of spells in the data set. Apply faceting so that each school of spells has its own plot. Label the bar colors "No Material Components" and "With Material Components". Store this plot in an object called `answer5`.

Question 6 (1 point).

The `spells` data set contains one `duration` that is clearly a typo. Create a copy of `spells` that replace this entry with `NA`. Then, use **recode** (look up the function help for this function on https://dplyr.tidyverse.org/ to change all instances with an "up to" label (with or without Concentration) in their `duration` to their corresponding duration label already present in the dataset (e.g. "Concentration, up to 1 hour" becomes "1 hour". In addition, "24 hours" becomes "1 day"). Create this object with a meaningful name initially, then copy it into an object called `answer6`.

Question 7 (1 point).

Using the dataframe created in Q6, create a **stacked barplot** that, for each of the spell duration categories, shows the number of spells per school in different colors. Order the duration categories on from short to long (infinite; see the variable description at the top of this document). Make the orientation of the stacked bars horizontal rather than vertical. If you want, you can substitute the `spells` dataframe for the Q6 dataframe. In any case, omit the spell category that is a typo or `NA`. Store this plot in an object called `answer7`.

Question 8 (1 point).

Create a dotplot with centered stacks that shows, for the `Evocation` and `Illusion` spell schools, the number of spells per `level`. Specify the binwidth so that you get one layer of dots for each spell level and resize the dots to half their original (=default) size. Apply the minimal theme and adjust the y-axis to label it "Spell Level", and have the y-axis gridlines appear ONLY at the integers from 0 to 9. Store this plot in an object called `answer8`.

Question 9 (1.5 points).

Create a list with three entries, named "Fire", "Fleece" and "Stone". "Fire" should contain a dataframe that shows the number of spells referencing this element keyword (case-insenstive) at least once in its description, separately for each spell level. "Fleece" should contain a dataframe with the name, school, level and components of all spells, sorted by ascending by level, that mention fleece as a material component. Finally, "Stone" should be a vector of named variables with entries "lowest" and "highest" representing the lowest and highest spell level for spells with Acid (case-insensitive) in their name. Store the answer in an object called `answer9`

Question 10 (1.5 points).

Create a plot that looks as much as possible like the one below. Use ggsave(plot = answer10, width = 6, height = 6, filename = "test.pdf") to check the dimensions