Assignment 2.2R

2024 Programming in Psychological Science

Contents

Assignment description

Each program is worth 1 point, although the second program may take more work. We will award partial points based on effort and correctness. You should also complete Assignment 2.1 for 8 points. The total of Assignments 2.1 and your choice of either 2.2R (this R challenge) or 2.2P (Python) are worth 10 points and 15% of your final course grade.

You must work individually, but feel free to ask questions in class and on Slack! Also, **using ChatGPT**, similar chatbots, LLMs, or other ANNs in any way for this assignment **is not allowed**. The reason for banning these tools is that we found they inhibit learning how to program. You are welcome to use these useful tools after the course is over (though check future UvA course restrictions).

You must submit **each** solution as an R program (.R file). So if you choose to write both programs, submit both files in a .zip folder. The best programs are those that work, follow style guidelines, have clear variable names, and have comments so that other programmers (and your future self) can read and adapt the code.

Q2.2R.1 (1 point)

Read about the infinite monkey theorem. See also this important cultural video.

Simulate one monkey typing random letters on a typewriter in sequences of 5 letters. The simulated monkey should type 5 letters, then a space, and so on. Create a **function** that returns the number of 5-letter sequences typed before making a coherent 5-letter word in a human language of your choice.

Use a file that contains all English (or other human language) 5 letter words (e.g., this one)

Run your function 100 times in a **for loop**. What is the mean and standard deviation of the number of 5-letter sequences? WARNING: if you loop many times, the loop could be quite slow. We recommend starting with a much smaller number (i.e., 5) for testing your code.

We are not awarding points based on plotting (one of the topics next week). However you could create nice plots and histograms of the result of running your function 100 times.

Q2.2R.2 (1 point)

Create a program that allows you to play video poker in the RStudio console (e.g. or other R console).

(a) Create a game of video poker as an R function. See this wikipedia page for more help on poker scenarios. Note that you do not need to know about other aspects of poker games. You are just simulating someone playing video poker **by themselves** at a casino or bar or at home (not with other players).

That game code should do the following: it draws 5 playing cards with the **kind** of card and **suit** of the card as separate strings for each card, e.g. `"king"` and `"hearts"`. You can choose to ignore other cards like `"jokers"` (but you can use them in your game if you wish).

The following scenarios should take away money (or points – if you are against gambling):

- High card (none of the same **kind** of cards nor are the cards of the same suit nor all ordered)
- Only one pair (e.g. two cards of the same **kind**)
- Two pair (two cards of the same **kind** with two other cards of the same **kind**)

The following scenarios should give the player money:

- Three of the same kind
- Straight
- Flush
- Full House
- Straight Flush

You have a choice of how to award money/points. You can also award or take away money/points for other scenarios other than the ones listed. You have the freedom to print certain messages given certain scenarios. You also have a choice of inputs to your function (maybe a player's bet?, although no inputs are necessary to draw cards). The function should at least return the amount of Euros/points gained or lost.

(b) Generate a while code that runs your game multiple times until a player has reached a certain amount of money/points. See your answer to Q2.1.15.

(c) We are not awarding points based on graphics. But graphics can be included in your program if you wish, either as text-based graphics in the R console or in a figure.

(d) The solutions should be somewhat unique across students. Although the internal dynamics of the function may be similar.