PHP2510

MANIPULATING DATA w/ R WEEK 4

1. This week, we will be working with a dataset that I created, as well as the tidyverse package, and (eventually) rMarkdown language. Let's first get comfortable with the dataset. Create an R script that does the following:

    a. Sets the working directory
    b. Loads in the dataset "simulated_health_survey.csv" in Canvas (Files → Lecture Slides and Handouts)
    c. Determines the number of observations and column names
    d. Adds another column that codes whether the patient is 18+
        i. Call this column *adult_status*
        ii. This can be 1/0 or T/F. Can you figure out how to do it both ways?
    e. Tabulates how many people were in each treatment group, by adult_status
    f. Writes a .csv file that only contains only adults taking the placebo
    g. Confirm that the amount of data you saved in step (f) matches your numbers in step (e)

2. Review the code in SPEEGLE Example 6.6. Apply this to the original dataset, with the following changes:
    a. Instead of grouping by title, group by treatment
    b. Instead of summarizing the rating (with the mean), do so for each week's quality of life score

3. After viewing the result in (2), make the following changes/additions:
    a. Now group by both treatment and adult status
    b. Add the counts of each group
    c. Order the results by week8 scores.

4. After viewing the results in (3), make the following changes/additions:
    a. Also calculate the percent of healthy people (stored in "disease_status") in each group
    b. Remove any groups with less than 10 people
    c. Order the results by average healthiness

5. After viewing the results in (4), start fresh with the original dataset. Now pick your own:
    a. Variable(s) to group by
    b. Variable(s) to summarize (you can use something other than the mean or counts!)
    c. Variable(s) to order on
    d. Variable(s) to filter by
    e. View the results. Then explain, in words, the first row in your output

6. Are the number of daily steps different for current smokers aged 50+ compared to never-smokers aged 49 and under? Use the tidyverse language from above to report the relevant information, making sure to calculate differences using statistics other than just the average. Explain your findings in words.

ADDITIONAL QUESTIONS

1. You develop a patient-reported outcomes survey that asks each subject to score their daily level of pain on a scale of 1, 2, 3, 4, 5. After a pilot data collection, you detect that respondents are half as likely to answer (1) and (5) as they are (2), (3), (4), which they score in equal likelihoods. Write down the distribution f(x). What is the expectation of a patient's pain? What is the variance? Can you explain the variance calculation (via a step-by-step transformation or visualization, perhaps)?
2. X and Y are normal random variables. Consider their product Z=XY. Is Z a normally distributed variable? Use R simulations to determine.
3. [CODING CHALLENGE] Deathrolling in World of Warcraft works as follows: Player 1 tosses a 1000-sided die (numbered 1 to 1000). Say they get x_1. Then player two tosses a die with x_1 sides on it (numbered 1 to x_1). Say they get x_2. Player 1 tosses a die with x_2 sides on it. And so on, until a player rolls a 1 and therefore losses. Use simulations in R to estimate the pmf of the total number of rolls. Then calculate the expected total number of rolls. Would you rather be player 1 or player 2 (or does it not matter)?

HINTS
#1: For calculating the variance, use E[g(x)] = sum_x g(x)f(x), where g(x) = (x-E[x])^2.
#2: Check if Z has the correct mass within certain intervals, given its mean and variance
#3: Try writing it in pseudo-code first, then:

- Look up the *while* function. How can this help you?
- Research how to write *for* loops. How can this help you?
- Look up the *sample* function. How can this help you?
- If you just knew the number of rolls it took (e.g. 23, 14, 7), how could you tell who won? Use this to your advantage: simulate only on how many rolls it takes before the game ends, not who won
- Research the %% command via ?'%%'

#3: PLAYBOOK

- create an empty vector to hold your results
- create a *for* loop; within the loop:
    - initiate *rolls* (set to 1)
    - roll a 1000-sided die with *sample*
    - so long as your roll isn't a 1 ← *while* accomplishes this
        - try again (using your previous roll's result)
        - add 1 to the *rolls* object to keep track of the total rolls you've done
    - if you do get a 1, store *rolls* (the number of rolls) in your results object
- inspect your results vector using: table, mean, and table(results%%2)