RMHI/ARMP Assignment 2023

Hello everyone! This is the description for the assignment, which is due on Canvas on Thursday April 6, 2023 before 11:59pm Melbourne time. You'll need to submit a Word-knitted version of the completed R Markdown file found in this zip file, according to the following instructions:

1. Rename the document called pset1.Rmd as studentID-pset1.Rmd. (Replace studentID with your student ID number). This is your R Markdown file, where you'll be putting all your code and answers.

2. Replace "Your name and ID goes here" in the header of the R Markdown file with your name and student ID. (Keep the quotes or it won't knit properly.)

3. While we encourage collaboration in tutorials and learning in general, *you should not be collaborating with anybody AT ALL for this assignment*. **That means sharing code privately or publicly; even talking in the abstract about problems will effectively be collusion.** You should be completing it independently, with no help from any other person in any capacity. Of course, as always, you are free to use any of the resources from the class to help you, and you're also free to google or look anything up that you like (as long as you aren't asking anybody, including discussion boards or AIs, questions related to this assignment). Note that I do look at places like chegg and will follow up if anything from this problem set is posted there.

4. Plagiarism check is enabled and you can check the similarity report on your submission. In previous years we have found people who tried to cheat, so please don't risk it! That said, understand that we will not be naively looking at the overall % figure: with this sort of assignment a certain amount of overlap is inevitable, so don't worry if you get what looks like a high % score as long as you know you didn't plagiarise or collude. With this sort of assessment, that % overlap is higher than essays and the like. We will be using the plagiarism check for the parts of the assignment where we'd expect some variability, and to give a general sense of the overall gestalt.

5. Complete all of the problems below in the R Markdown document. *Do not remove any of the arguments to the code chunks, like the names of the code chunks or where it says message=FALSE or whatever*. If a problem asks you to display a tibble or variable so it shows up in the knitted version, **make sure that you do** as the marker cannot evaluate it without seeing it, and if they can't see it then they won't be able to award you points for it! Remember that to display a tibble (or any variable) you just type its name on a line of its own within the R chunk.

6. I've structured this so that, as much as possible, **questions do not build on each other**. That means that if, say, you can't get Q5 then you can still get Q6. Try to do all of them.

7. **Go for partial credit!** Many of these questions have some form of partial credit possible. What that means is that if it is asking for some R code, break down the problem into pieces. Even if you can only do some of the pieces, or do them part of the way, that will be worth something. [Note that there is no question-by-question rubric available because designing one would mean giving away the answers. In general we will give full credit for responses that correctly address all of the parts of the question.] Short answer questions (SAQs) can also be given partial credit and are generally asking for some thoughtful interpretation. If it is based on a previous graph or test you've done, if you did the first part wrong but discuss it well, you can still get most or all points for the SAQ part. If your code does not run but you want to include it for possible partial credit, just comment it out (using the # sign) so that it shows up in the knitted document but R does not try to run it. If you include a lot of commented-out code and some is correct and some isn't, we will not give you credit for the commented-out code; put the thing in there that you think is the closest to the correct answer, don't just include everything.

8. We are not overly worried about to what decimal place you round answers to and you will not lose credit for this unless you round so much that your answer is impossible to discern (e.g., don't round p-values to the nearest integer!). Similarly, you will not lose points for trivial presentation things like using parentheses instead of commas around statistical references, as long it's clear.

That said, for those who want a guideline, I'll suggest that you follow APA format or round p-values to three decimal places, degrees of freedom to one, and test statistics and probabilities to two. (Note: this problem set doesn't incorporate all of these things, this is just my standard guideline).

9. Some questions specify a word count. In that case you need to either calculate it from the knitted document or type up your answer in Word[1] and then cut and paste it into the R Markdown file. (Please put your answer in between the word ANSWER and [Word count: XX]; needless to say, those two bits do not count towards your word count.) I know that's annoying; sorry. Anything else I thought of, like specifying a number of sentences or having no limit, was worse in terms of equity across students. The word counts I've specified in each question are designed to give you a guideline about the maximum amount of words you should need answer completely and correctly. So don't feel like you must use all of the words; if you can answer it fully with less, that's fine. In fact, the total word count for the solution set I wrote up is around 900, so it's possible to fully answer the questions while going substantially under the word limit. That said, it is okay to go over the word limit for individual questions as long as the total word count for all of the questions combined is fewer than 1320 words (i.e., fewer than 1200+10%, with the standard penalty if it is 1200+10% or over. See the student manual for details on word count penalties).

10. There is no word count for code chunks. Word count only applies to the short answer questions as indicated. **Remember to report** your total word count for the assignment as a whole at the top of the document. Your total word count is the sum of the word counts for all of the SAQs.

10. You'll be turning in the knitted output of your R Markdown file. We prefer that you knit to Word but if you can't get Word to knit then html is okay. In the worst case, you can turn in the completed Rmd file. **I highly, *highly* recommend that you knit as you go**: (a) knitting can identify problems in your code that you would have otherwise missed; and (b) you do not want to get close to the deadline and think you're done only to find that you're having troubles knitting. Save yourself the panic and knit often.

11. Similarly, you can turn in the assignment multiple times before the deadline, so I strongly encourage you to turn it in even before it's perfectly polished. That will save you last-minute panic or computer issues. Also, take a screenshot for proof of having turned it in just in case you need it. If you run into last-minute computer issues and can't even succeed in uploading an Rmd, email Andy your assignment as soon as possible to demonstrate that it was done at that time. We cannot make promises about whether you will receive any late penalties if you do this, but if you don't, you very probably will get penalised because we have no way of knowing if the problems were genuine.

[1] I know different software calculates word count in slightly different ways, so we are using Word as the standard.

Athletics Day!

Our friends in Bunnyland are starting to get upset and angry at each other, so in an effort to have some fun and promote team bonding, they all decide to have a fun day. About a year ago they had their first annual Athletics Day, where they competed in various events like sprints, long jump, weight lifting, hide and seek, and hurdles. That was super fun so they decided to have another one!

The nerds of the group (ahem, Shadow) decided to keep track of everyone's performance. Great for motivation and very interesting to see if anybody got new personal bests. This data can be found in the tibble `d`, which has been loaded for you in the R Markdown document. Each row is a person, and the columns are as follows:

name: the name of that person
age: the age of that person
gender: the gender of that person: male, female, or nb (non-binary)
year: either current (this year) or previous (the last Athletics day)
event: five events: sprint, weightlifting, hide and seek, hurdles, and long jump
rank: where that person placed in that event (1=first, 9=last. Note that there are some ties)
detail: this has a different meaning for each event, as follows:
    sprint: time in seconds to run 100m (smaller = better)
    weightLift: maximum weight[2] in kg they lifted (larger = better)
    hideAndSeek: time in minutes to get found (larger = better)
    hurdles: time in seconds to hurdle 100m (smaller = better)
    longJump: distance in meters of longest jump (larger = better)

There is also `dw1` and `dw2`, which are wide-form versions of `d` (in other words, they have exactly the same data, just in a different format). They have been loaded for you. `dw1` has no column for *year* and instead has columns *currentRank*, *currentDetail*, *previousRank*, and *previousDetail*. `dw2` has no column for *event* and instead has rank and detail columns for each of the five events. Be sure you have a look at both of these and familiarise yourself with how they are organised.

Q1 [3% of total mark]

Use the `table()` function to determine how many people there are of each gender in the dataset, and report your answer in the blank space in the Markdown file.

Q2 [8% of total mark]

(a) Use baseR (i.e., only things you were taught before Week 3) to add a new variable to `d`. The variable should be called *medal* and its value on each row should be TRUE only if the rank of the person on that row for the event on that row is 1, 2, or 3. (b) Create the same variable as in (a) but instead use function(s) from tidyverse (i.e., something you were taught in Week 3). (c) Use any function(s) in R you like to determine which animal(s) got the most medals and how many medals that was.

Q3 [7% of total mark]

Use tidyverse function(s) to create two smaller datasets out of `d`. `dt` should contain just the events involving times (i.e., *sprint*, *hideAndSeek*, and *hurdles*) and `do` should contain the other two. Use the function `nrow()` to report how many rows are in each dataset (note: I did not teach you this function, you will need to use the help files and google to figure it out). Do the results suggest that you created these datasets correctly? Explain your reasoning. [Suggested word count: 60]

[2] You'll notice that Quackers has an NA value for all of the weightlifting events. He can't lift weights with his little wings. ☹

Q4 [12% of total mark]

(a) Use tidyverse function(s) to remove the variables *detail* and *medal* from `d` and then create a tibble called `d_new` that looks like the one below (don't worry if your rows and columns are in a different order as long as they are all there and the values are correct). Make sure that the top rows of the tibble are visible when you knit your Markdown document.

```
> d_new
# A tibble: 45 × 6
   name          age gender event       current previous
   <chr>       <dbl> <chr>  <chr>         <dbl>    <dbl>
 1 bunny          10 female hideAndSeek       2        5
 2 bunny          10 female hurdles           3        5
 3 bunny          10 female longJump          6        5
 4 bunny          10 female sprint            7        6
 5 bunny          10 female weightLift        8        7
 6 cuddly paws     9 female hideAndSeek       3        4
 7 cuddly paws     9 female hurdles           4        3
 8 cuddly paws     9 female longJump          2        4
 9 cuddly paws     9 female sprint            4        2
10 cuddly paws     9 female weightLift        4        4
# … with 35 more rows
```

(b) Do the exact same thing as in part (a) except don't remove *detail* and *medal* first and save the resulting tibble in a variable called `d_weird`. Make sure the top rows of `d_weird` are visible when you knit your document. Why is `d_weird` so different from `d_new`? [Suggested word count: 80]

Q5 [14% of total mark]

(a) Using tidyverse functions from Week 3, create a tibble called `d_sum` from `d`. It should look like the tibble shown below, with one column for *year*, one for *event*, and two new variables. *mnDetail* is the mean value for *detail* for that event for that year, and *sdDetail* is the standard deviation of *detail* for that event for that year. The tibble `ds`, which has been loaded for you, is identical to `d_sum` in case you want to compare them directly (again, don't worry if the order of your rows or columns is different as long as all of the rows and columns are there, and the values are equal). Make sure `d_sum` is visible when you knit your document.

```
> d_sum
# A tibble: 10 × 4
   year     event       mnDetail sdDetail
   <fct>    <chr>          <dbl>    <dbl>
 1 previous hideAndSeek     6.82    0.905
 2 previous hurdles        20.6     2.93
 3 previous longJump        2.41    0.414
 4 previous sprint         18.1     2.85
 5 previous weightLift     12.3     2.45
 6 current  hideAndSeek     6.77    0.905
 7 current  hurdles        22.4     3.17
 8 current  longJump        2.1     0.410
 9 current  sprint         19.3     2.73
10 current  weightLift     10.6     2.35
```

(b) The code that is given to you in the q5b chunk in the Markdown uses a new function called rowwise() along with functions you've seen to create the tibble dbest, shown below. This tibble shows the best rank for each person over all of their events in both years (e.g., Bunny's best was 2nd place, Doggie's was 1st place, etc). Poor LFB! ☹

```
> dbest
# A tibble: 9 × 2
  name        overallBest
  <chr>             <dbl>
1 bunny                 2
2 cuddly paws           2
3 doggie                1
4 flopsy                1
5 foxy                  2
6 gladly                2
7 lfb                   3
8 quackers              1
9 shadow                1
```

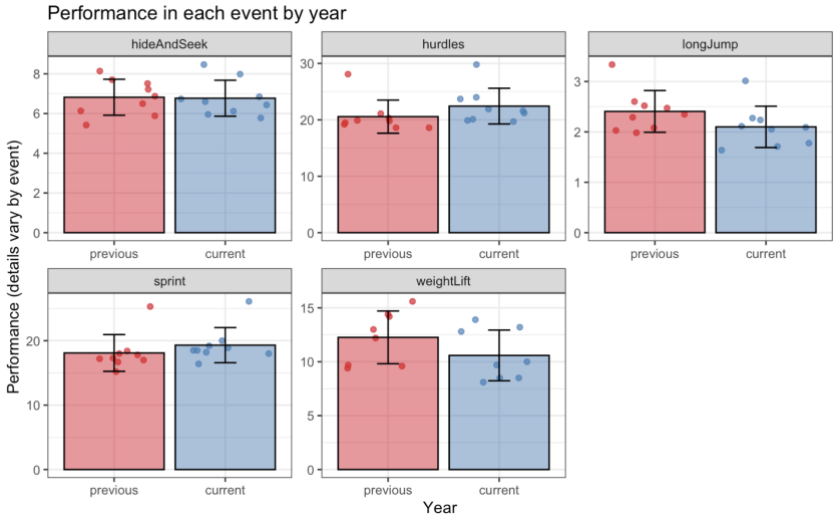
Based on this example code, help files, and google, figure out what rowwise() does and how to use it. Describe what it did using the code in the q5b chunk as an example: if you were trying to teach somebody how to use rowwise() using the code given as an example, what would you say? Note that if you only give a generic explanation of the function, you will not receive full credit. The key is to explain what it is doing in the q5b code *specifically*. Note also you don't need to explain the parts of the code that are completely unrelated to anything rowwise() does. [Suggested word count: 80]

(c) Using rowwise() along with whatever other tidyverse functions you need, add a column called *meanRank* to the dw1 tibble and put the result into a new tibble called dw1mean. The new column should show the mean of the ranks of the *currentRank* and *previousRank* columns, as shown below. (Note that if you create dw1mean without using rowwise() you will not get marks for that part, because the point is to exercise and demonstrate your skill in figuring out this new function. In other words, it is better for you to use rowwise() somewhat wrongly and get partial credit than to create dw1mean without using rowwise(). As before, don't worry if the order of your rows or columns is different as long as all of the rows and columns are there, and the values are equal).

```
> dw1mean
# A tibble: 45 × 9
   name          age gender event       currentRank previousRank currentDetail previousDetail meanRank
   <chr>       <dbl> <chr>  <chr>             <dbl>        <dbl>         <dbl>          <dbl>    <dbl>
 1 bunny          10 female hideAndSeek           2            5          7.98           6.87      3.5
 2 cuddly paws     9 female hideAndSeek           3            4          6.85           7.22      3.5
 3 doggie          7 male   hideAndSeek           8            9          5.96           5.42      8.5
 4 flopsy          5 nb     hideAndSeek           6            7          6.44           6.13      6.5
 5 foxy           13 female hideAndSeek           7            2          6.12           7.7       4.5
 6 gladly          8 male   hideAndSeek           4            3          6.74           7.51      3.5
 7 lfb             9 female hideAndSeek           5            6          6.6            6.5       5.5
 8 quackers        6 male   hideAndSeek           9            8          5.78           5.89      8.5
 9 shadow          5 female hideAndSeek           1            1          8.47           8.14      1
10 bunny          10 female hurdles               3            5          20.1           19.8      4
# … with 35 more rows
```

Q6 [13% of total mark]

(a) Let's visualise some data. In the code chunk for this question make a bar plot like the one below using whatever tibble(s) you think are best (this can be any of the ones that were loaded for you and/or any that you made). For full credit, your figure should have all the components in the figure below (i.e., five panels, semi-transparent bars, dots, error bars, title, labels, etc). The bars should indicate the mean of *detail* for the corresponding event, with each dot indicating an individual data point corresponding to the *detail* for a single person in that event. (Note that your individual data points will not be exactly in the same place as this figure because the geom introduces randomness; that is fine). The error bars should indicate one standard deviation. It is fine if your colours aren't exactly the same (you aren't expected to guess what palette was used here) as long as you use a sensible palette, the colours of the dots match the bars, and the colours vary by year as shown here.



(b) Based on this graph, describe any trends or regularities in performance that you see from the previous year to the current one. This is not a R question but rather a thought question asking you to critically think about what the data might be demonstrating and what this might suggest is going on in Bunnyland (speculation is fine as long as it is grounded in the data and you are clear that it is speculation). You're not expected to make claims about significance but think about the meaning of the variables and discuss what (if anything) this figure might suggest about overall performance from year to year. [Suggested word count: 130]

Q7 [13% of total mark]

(a) Make a figure of your own using any of the datasets, with the goal of learning something new about the data that hasn't been shown by the previous figure. It needs to incorporate at least two things that you haven't been taught; these can be anything from new geoms, a different palette package than RColor Brewer, a different theme, changing the size, orientation, or style of your fonts, putting text inside the figure, changing aesthetic properties, or many other possibilities; you can do basically whatever you want as long as it's new. The figure should have an informative title and axis labels, a theme and colour palette other than the default, and the aesthetic choices should add to its clarity rather than detract from it.

(b) Explain what each new thing is and how you made it. This doesn't need to be extensive – for instance, if you hadn't already been taught `show.legend` you might say "I got rid of the legend by adding `show.legend=FALSE` as an argument to the geom". [Suggested word count: 50]

(c) Explain what your figure suggests about the data. In your explanation be sure to describe the variables on each axis as well as what the pattern is (if there is one) and what it suggests about what (if anything) is going on. You won't be evaluated on how interesting your result is, but on how clear and appropriate your explanation is given the figure. [Suggested word count: 130]

Q8 [4% of total mark]

"I don't understand," Gladly says. "Why don't we just set alpha very low, like 0.001? Then if something is significant, that means that 99.9% of the time the alternative hypothesis is true."

There are two main problems with Gladly's idea. Explain them to him. For each be sure to be clear about what the problem is and why it is a problem. [Suggested word count: 120]

Q9 [10% of total mark]

The tibble we created in Q5 shows us that the sample mean for *sprint* in the previous year was 18.1 seconds (sd = 2.85), while the sample mean for *sprint* in the current year is 19.3 seconds (sd=2.73). This shows that the mean speed has slowed: 19.3 seconds takes longer than 18.1 seconds.

The dataset shows that, although Doggie won the sprint both times, he also slowed down. In the previous year his time was 15.2 seconds and the current year his time is 16.4 seconds.

Suppose we wanted to figure out how well Doggie did *relative* to everyone else each year. One way to achieve that might be to calculate the probability of observing his time or faster given the sample mean that year. This is what you will do here using the function(s) taught in Week 5. (You do not need to use any of the datasets you have loaded).

(a) What is the probability of Doggie achieving the time he did or faster in the **previous** year, assuming that the sample mean and sample standard deviation that year reflect the "true" underlying (normal) speed distribution that year?

(b) What is the probability of Doggie achieving the time he did or faster in the **current** year, assuming that the sample mean and sample standard deviation that year reflect the "true" underlying (normal) sprint speed distribution that year?

(c) Relative to everyone else that year, did Doggie do better in the previous year or the current year? Explain your answer with reference to the probabilities you calculated in parts (a) and (b). Note that you can receive full credit for this even if your calculations earlier were wrong as long as you correctly reason about the numbers you calculated. [Suggested word count: 70]

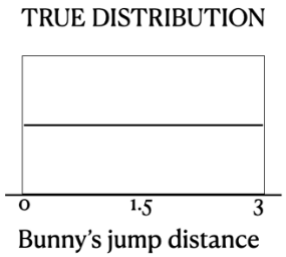
(d) How do the numbers you calculated in (a) and (b) relate to a p-value? Explain your answer with reference to the null hypothesis and the definition of p-value. This is a conceptual question; you do not need to do any calculations here. [Suggested word count: 120]

Q10 [14% of total mark]

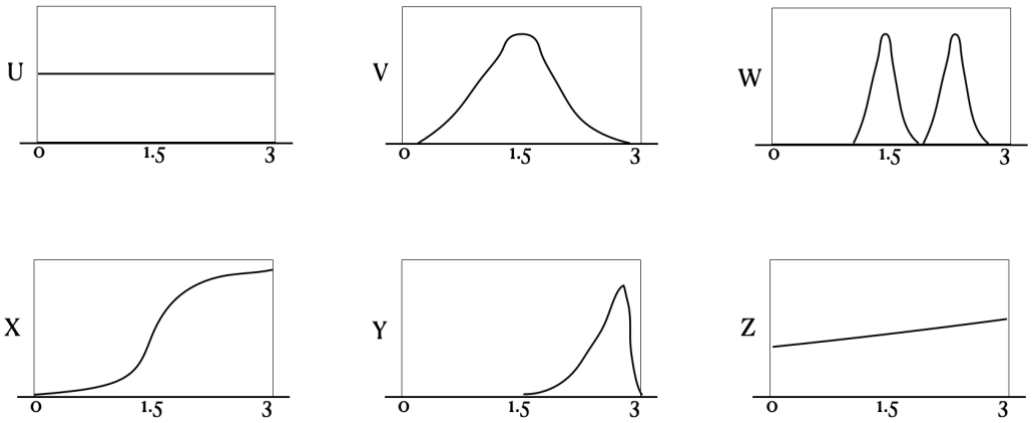
The long jump is an unusual event because a person is evaluated based on their *best* attempt out of several. In sprinting, like most events, there is only one attempt and the winner is the person who runs fastest in it. By contrast, in long jump people are given a number of jumps and their score is the maximum distance out of all of the attempts. To illustrate, suppose Bunny had the following three attempts: 1.9 meters, 0.32 meters (she tripped), and 2.05 meters. Her score would be the maximum of all of them: 2.05 meters. This is the same score she'd get if her three jumps were 2.01, 2.03, and 2.05. However, if her score were the mean of the three attempts, these two situations would result in very different scores: in the first it would be 1.42 and in the second it would be 2.03.

Now, remember that one can have sampling distributions of any kind of statistic. We've spent a lot of time talking about the sampling distribution of the mean, but we could also think about the sampling distribution of the maximum, which applies when thinking about long jump scores. In this problem you will reason about this situation, by direct analogy and extrapolation from what you've learned about the sampling distribution of the mean.

Bunny thinks that the *true* underlying distribution of her long jump scores is uniform between 0 and 3 meters. In other words, she thinks she is equally likely to jump 0 meters or 1 meter or 2 meters or 3 meters or any length in between, but she has 0% probability of jumping further than 3 meters.[3] She has helpfully drawn you the picture on the right to illustrate what she thinks it looks like. For the purposes of this question let's assume that she is correct and this is the true distribution.



(a) Suppose Bunny participates in 1000 long jump competitions. In each competition she jumps 50 times and her score is the *maximum* distance out of that 50.[4] Consider now the six panels U through Z. Give the letter of the panel that most accurately captures what you would expect the sampling distribution of her scores after 1000 competitions to look like. Explain your answer, making reference to the definition of sampling distribution. Hint: begin by thinking about what you would expect the maximum value of a single set of 50 jumps to be. [Suggested word count: 120]



(b) Suppose that Bunny participates in 1000 long jump competitions of 50 jumps each, but now her score is the *mean* distance out of that 50. Consider the same panels U through Z and give the letter of the panel that most accurately captures what you would expect the sampling distribution of her scores to look like. Is this the same as in (a)? Why or why not? [Suggested word count: 120]

(c) Suppose instead that in each of the 1000 competitions Bunny only jumps once instead of 50 times. Now it does not matter whether her score is the maximum or the mean of that one jump; the sampling distribution of her scores is the same in each case. Give the letter corresponding to the panel that most accurately captures that distribution. Explain why it has that shape and why it is the same whether or not her score is the mean or the maximum. [Suggested word count: 120]

* Note: You do not need to do any calculations or code in this question! And if your intuitions about maximums are incorrect but your explanation of sampling distributions in general (and the sampling distribution of the mean as it applies here) is correct, you can still get most of the partial credit.

Q11 [2% of total mark]

These marks are free as long as you say anything! What is your current theory about why everyone in Bunnyland is going hungry? (No word limit here, say as much or as little as you want)

[3] Let's not worry about whether Bunny is correct about this. She probably isn't (because it's a very silly assumption), but for the purposes of this question we're going to assume she is and see what we can figure out given that.

[4] I know it is extremely unrealistic to think she could do that much jumping without getting tired, even though she is a bunny. Just go with it because the point of this is not to think about bunnies; it is to exercise your understanding of sampling distributions, so we must live in a world of idealisations. Assume that she did that much jumping, she did not ever get tired, and each jump was an independent sample from her underlying true distribution.