

RMHI/ARMP Assignment 2024

Hello everyone! This is the description for the assignment, which is due on Canvas on Monday April 15, 2024 before 08:00am 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 we 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, or use print().
6. We'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) or type eval=FALSE in the R chunk 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!), or unless it is specifically instructed by the question. 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, we 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 our 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 Word¹ 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.) We know that's annoying; sorry. Anything else we thought of, like specifying a number of sentences or having no limit, was worse in terms of equity across students. The word counts we'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 1070, 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. We will automatically mark the latest submitted assignment. Submitting often 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 submit a corrupted file or the wrong assignment that is not grounds for waiving any late penalties; it is your responsibility to make sure that the submission is correct. If you run into last-minute computer issues and can't even succeed in uploading an Rmd, *email us (rmhi-armp@unimelb.edu.au) 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 We know different software calculates word count in slightly different ways, so we are using Word as the standard, as per the guidelines in the student manual.

Talent Show!

Our friends in Bunnyland are starting to get upset and angry at each other, so in an effort to have some fun and promote bonding, they all decide to have a talent show. They decide to have two different levels: a *fun* one where people just do their talent, and a *competitive* one where there are judges giving 1st, 2nd, and 3rd place trophies. There are also lots of different kinds of talents and some rules for participation, explained in the description of the dataset below.

The nerds of the group (ahem, Shadow) decided to keep track of how it went. 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 Table 1 below describes the columns.

Variable	Description	Table 1: variable description for tibble <code>d</code>
<i>name</i>	The name of the person in that row	
<i>kind</i>	The species of the person in that row	
<i>level</i>	The level at which they were performing (i.e., fun or competitive). Thus each person contributes two rows, one for each level. There were two rules for participation: 1) everybody needed to participate in both the <i>fun</i> and the <i>competitive</i> levels; and 2) they could not present the same kind of talent at both levels (e.g., you couldn't play a musical instrument for both your fun and competitive talents, but you could play an instrument for one and sing for the other). That said, the rules were not necessarily always followed by all of the participants!	
<i>talent</i>	The talent type they performed at that level. Options included singing a song (singing), dancing to a song (dancing), playing a musical instrument (instrument), a comedy routine (comedy), a magic trick (magic), or something else (other). Only two people chose other; Shadow solved a Rubik's cube very rapidly and Fluffy the cat delivered a speech about the importance of napping.	
<i>audience</i>	The rating from the audience for that performance. This was obtained by randomly sampling 10 people from the audience and asking each person if they liked the performance or not. Thus a score of 8 means that 8 of the 10 people liked the performance. Occasionally an audience rating may be NA because the person recording it kept getting hungry and leaving to find food.	
<i>judge</i>	This can be a 1, 2, 3, or NA. All fun performances were NA, because no judges rated them. A competitive performance was 1 if it won 1st place for that talent type, 2 if it won second, 3 if it won third, and NA if it did not win an award.	

The Markdown also loads a few other tibbles. `dd` contains additional data and will be explained in Q4; you don't need it before then. There are a few other tibbles (e.g., `d3b`, `d6`) which will be explained on the questions where they are relevant and you can ignore until then.

Q1 [8% of total mark]

(a) Use the `table()` function to determine how many performances there were for each type of *talent* at each *level*. Make sure the table shows up in the knitted Markdown. You don't need to report anything else or assign the table to a variable.

(b) Change the order that the talents show up in the table. We have not taught you how to do this but the very first chunk in the Markdown contains code that changes the order of the *level* variable in `d`, so you just need to adapt that code and apply it to the *talent* variable. The new order should be the same order as the *talent* variable description in Table 1. Now use the `table()` function to display how many performances there were for each talent (don't split by level this time). You don't need to assign the table to a variable but make sure the output of the `table()` function shows up in the knitted Markdown. Which talent was most common, and how many performances of it were there?

(c) Rename the *kind* variable to *species* and use the `head()` function to make sure that only the top rows of *d* are visible in the knitted document. (Note: we have not taught you how to rename variables, you will need to google around yourself to figure out how to do this. It can be done with one function but if you code it in another way, as long as it works and your code comments make it clear that you understand what it does and how, it is possible to earn full marks).

Q2 [11% of total mark]

(a) Use baseR only (i.e., only things you were taught before Week 3) to keep only the people who won 1st or 2nd *and* achieved an audience rating of 8 or more. You don't need to assign the result to any tibble (and don't write over the existing *d*!) but your output should look like the screenshot below when it is knitted. (Don't worry if the order of the rows/columns is different, but there should be the same number of rows and columns and they should have the same values).

```
## # A tibble: 26 × 6
##   name      species level      talent audience judge
##   <fct>    <fct>   <fct>    <fct>    <dbl> <dbl>
## 1 gladly    bear    competitive singing      8      2
## 2 <NA>      <NA>    <NA>      <NA>     NA     NA
## 3 <NA>      <NA>    <NA>      <NA>     NA     NA
## 4 bunny     bunny   competitive magic       8      2
## 5 <NA>      <NA>    <NA>      <NA>     NA     NA
## 6 cuddly paws bunny   competitive dancing      8      1
## 7 <NA>      <NA>    <NA>      <NA>     NA     NA
## 8 flopsy    bunny   competitive singing     10      1
## 9 <NA>      <NA>    <NA>      <NA>     NA     NA
## 10 <NA>      <NA>    <NA>      <NA>     NA     NA
## # 16 more rows
```

(b) Use function(s) from tidyverse that you were taught in Week 3 to accomplish the same task as in part (a): keep only the people who won 1st or 2nd *and* achieved an audience rating of 8 or more. As before, you don't need to assign the result to any tibble (and don't write over the existing *d*!). Your output should look like the screenshot below when it is knitted. (Don't worry if the order of the rows/columns is different, but there should be the same number of rows and columns and they should have the same values).

```
## # A tibble: 6 × 6
##   name      species level      talent audience judge
##   <fct>    <fct>   <fct>    <fct>    <dbl> <dbl>
## 1 gladly    bear    competitive singing      8      2
## 2 bunny     bunny   competitive magic       8      2
## 3 cuddly paws bunny   competitive dancing      8      1
## 4 flopsy    bunny   competitive singing     10      1
## 5 lfb       bunny   competitive comedy       8      1
## 6 shadow    bunny   competitive magic       9      1
```

(c) You will notice that (b) and (a) do not match. Why? Answer in terms of what exactly the relevant part of baseR code is doing and how that is different from what exactly the relevant tidyverse code is doing. Note that you don't need to discuss all of the components of your code, just the parts that are relevant to explaining the difference between (a) and (b).

[Suggested word count: 100]

(d) Use baseR only (i.e., only things you were taught before Week 3) to create output that matches the screenshot in (b). As before you don't need to assign the result to any tibble, just make sure that the output when knitted looks like (b). (Don't worry if the order of the rows/columns is different).

Q3 [12% of total mark]

(a) Use a single tidyverse function you were taught to remove the *judge* and *audience* columns from *d* and assign the result to a new tibble called *dshort*. Make sure that the top rows of *dshort* are visible in the knitted Markdown.

(b) Use tidyverse function(s) you were taught in Week 3 to transform *dshort* so that it looks like the tibble in the screenshot below. (Don't worry if the order of the rows/columns is different, but there should be the same number of rows and columns with the same values). Assign the result to a new tibble called *d2*. Make sure that the top rows of *d2* are visible in your knitted Markdown.

```
## # A tibble: 23 × 4
##   name      species competitive fun
##   <fct>    <fct>    <fct>    <fct>
## 1 gladly   bear      singing   dancing
## 2 panda    bear      <NA>      dancing
## 3 snowy    bear      <NA>      singing
## 4 bunny    bunny     magic     comedy
## 5 cuddly   paws     bunny     dancing   instrument
## 6 flopsy   bunny     singing   instrument
## 7 giganticky bunny     magic     dancing
## 8 lfb      bunny     comedy    singing
## 9 pink     bunny     instrument singing
## 10 shadow  bunny     magic     other
## # [i] 13 more rows
```

(c) Why did we have you perform the transformation in (b) using *dshort* instead of *d*? In other words, what happens if you were to do it on *d*, and why does this happen? You do not need to show any code or output to get full marks on this question but you can if you want to. If you do, be sure to refer to the code or output in your answer so it is clear why/how it is relevant.

[Suggested word count: 100]

(d) Use your *d2* tibble to determine if anybody broke either of the two rules of the talent show that are explained in the description for *level* in Table 1. For each rule, you should include code that identifies individuals that broke this rule – don't just look at the tibble manually to find them. In your answer, be sure to list everyone who broke a rule along with what rule(s) they broke. If you did not succeed in creating *d2* in part (b), you can use the tibble called *d3b* that has already been loaded for you.

Q4 [7% of total mark]

(a) Change *d* so that the order of the *name* variable in it is alphabetical. Make sure that the top rows of *d* are visible in the knitted Markdown.

(b) One of the tibbles that has already been loaded for you is called *dd*. It contains the same data as *d* in the columns *name*, *level*, and *talent* (i.e., the same people and performances) but contains a new variable. A full explanation of the variables in *dd* is shown in Table 2.

Variable	Description	Table 2: variable description for tibble <code>dd</code>
<i>name</i>	The name of the person in that row	
<i>level</i>	The level at which they were performing (i.e., fun or competitive). Thus each person contributes two rows, one for each level. There were two rules for participation: 1) everybody needed to participate in both the <i>fun</i> and the <i>competitive</i> levels; and 2) they could not present the same kind of talent at both levels (e.g., you couldn't play a musical instrument for both your fun and competitive talents, but you could play an instrument for one and sing for the other). That said, the rules were not necessarily always followed by all of the participants!	
<i>talent</i>	The talent type they performed at that level. Options included singing a song (singing), dancing to a song (dancing), playing a musical instrument (instrument), a comedy routine (comedy), a magic trick (magic), or something else (other). Only two people chose other; Shadow solved a Rubik's cube very rapidly and Fluffy the cat delivered a speech about the importance of napping.	
<i>duration</i>	The duration in minutes of that performance. For instance, a value of 8.3 means that the performance of that person at that level with that talent took 8.3 minutes.	

Combine `d` and `dd` together using the function `full_join()`. We have not taught you this function so you will need to use your investigative skills to look it up and play around with it until you have figured it out. Assign the combined dataset to a new tibble called `d_full`, and make it so the top rows of `d_full` show up in the knitted Markdown. It should look like the screenshot below (rows may be in a different order, but the column order, column names², size of the tibble, and data in each cell should be the same).

```
## # A tibble: 42 × 7
##   name      species level talent audience judge duration
##   <fct>    <fct>   <fct> <chr>      <dbl> <dbl>    <dbl>
## 1 barky     dog      competitive comedy      7      3     11.6
## 2 black     dog      competitive dancing     7      3      5.3
## 3 black     dog      fun      comedy     9     NA      9.9
## 4 bunny     bunny    competitive magic      8      2      8.9
## 5 bunny     bunny    fun      comedy    10     NA      8.3
## 6 cuddly paws bunny    competitive dancing     8      1      4.7
## 7 cuddly paws bunny    fun      instrument 9     NA      5.5
## 8 doggie    dog      competitive comedy     NA      2      9.2
## 9 doggie    dog      fun      singing     9     NA      4.1
## 10 douglas  hedgehog competitive singing     9      3      4.4
## #   32 more rows
```

(c) The code given in the chunk here combines two tibbles by using the function `cbind()` rather than the function `full_join()`. The output has been assigned to a tibble called `dc` whose output in the console is shown below. Based on a comparison of `dc` and `d_full`, describe two major differences between what `cbind()` and `full_join()` do, making clear reference to the parts of the tibbles that illustrate each difference. Finally, explain *why* these differences have occurred: how exactly `cbind()` combines tibbles that is different from how `full_join()` combines tibbles.

[Suggested word count: 90]

```
> head(dc)
  name species level talent audience judge name level talent duration
1  barky     dog competitive comedy      7      3  barky competitive comedy     11.6
2  black     dog competitive dancing     7      3  black      fun comedy      9.9
3  black     dog      fun      comedy     9     NA  bunny      fun comedy      8.3
4  bunny     bunny competitive magic      8      2  doggie competitive comedy      9.2
5  bunny     bunny      fun      comedy    10     NA  lfb competitive comedy      8.9
6 cuddly paws bunny    competitive dancing     8      1 paw paw competitive comedy      9.0
```

² Note that if you did not succeed in Q1(c) in renaming *kind* to *species*, your tibble here will have a column called *kind* instead. That is fine; you will only be penalised for this in Q1(c) and can still obtain full marks in Q4(b).

Q5 [15% of total mark]

(a) A tibble has been loaded for you called `df`, which is the same as `d_full`. We are providing you with `df` here in case you weren't able to create `d_full` in Q4(b). Use the `mutate()` function along with `case_when()` to make a new character variable in `df` called `durType`. [Note: We have not taught you `case_when()`]. The value of `durType` is "long" if `duration` is more than 10, "short" if it is less than 5, and "medium" otherwise. Be sure to show the top of `df` in the knitted Markdown.

(b) Using only functions we have taught you, use `df` as the basis to create the tibble shown in the screenshot below. Assign it to the name `ds`, and make sure `ds` is visible in your knitted Markdown. Helpful hint: all of the variables are calculated from the `audience` variable. `medAud` indicates the median, and the others are self-explanatory.

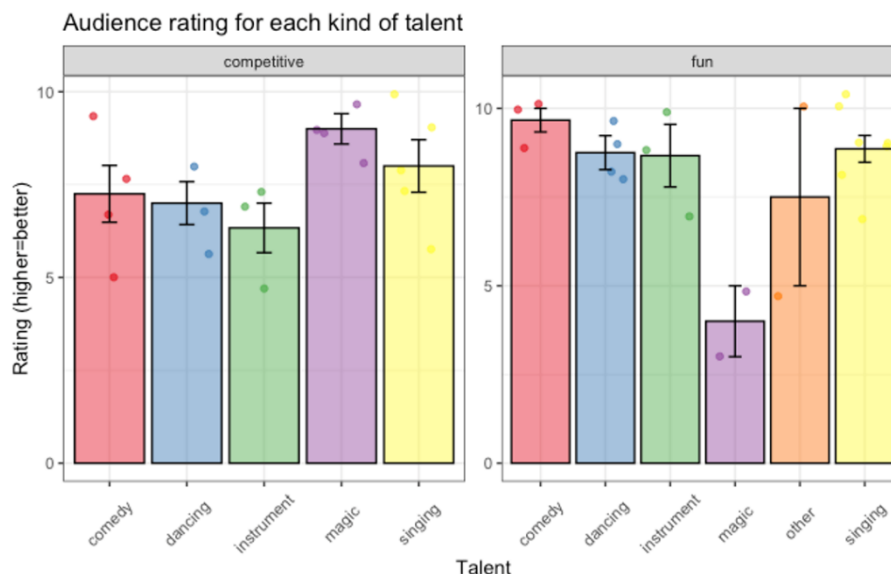
```
## # A tibble: 6 × 6
##   talent      medAud mnAud sdAud      n sderrAud
##   <chr>      <dbl> <dbl> <dbl> <int>    <dbl>
## 1 comedy         9    8.29  1.80     8    0.636
## 2 dancing         8     8    1.29     7    0.488
## 3 instrument      7    7.5   1.76     6    0.719
## 4 magic          8.5   7.33  2.73     6    1.12
## 5 other          7.5   7.5   3.54     2    2.5
## 6 singing        9    8.5   1.31    13    0.365
```

(c) Based on the data in `ds`, what talent is the least popular based on the mean audience ratings, and what is the least popular based on median audience ratings? Why do the mean and median ratings for these give different results? Your answer should refer to the idea of central tendency that both mean and median each capture, and it should explain the discrepancy by relating this idea to the actual talent show data.

[Suggested word count: 100]

Q6 [12% of total mark]

(a) Make a bar plot like the one below using the `d6` tibble, which has been loaded for you. For full credit, your figure should have all the components in the figure below (i.e., two panels, semi-transparent bars, dots, error bars, title, angled x-axis tick labels, three y-axis tick labels, etc.). Note that your individual data points will not be in exactly the same place as here because the `geom` introduces randomness; that is fine. The error bars should indicate one standard error. It's fine if your colours aren't exactly the same (you aren't expected to guess what palette was used) as long as you use a sensible palette and theme, and the colours of the dots match the bars and vary as they do here. Note that if your knitted figure has a slightly different aspect ratio that is fine, as long as all of the elements are present and correct; different systems knit figures in slightly different ways.



(b) Based on the graph in 6(a), describe any trends or regularities in performance that you observe. This is not a R question but rather a thought question asking you to critically think about what the data might be demonstrating and why this might be happening (you should speculate; just make sure to ground the speculation in the pattern of data and clearly indicate the part that is speculative). 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 the talent show.

[Suggested word count: 120]

Q7 [11% of total mark]

(a) Make a figure of your own using any of the tibbles provided (or any that you make from them if you want). Your goal is to show something new about the data that hasn't been shown by the previous figure. You should use at least one `geom` that you didn't use in Q6, and you also need to incorporate two elements that you haven't been taught in this subject. These can be anything from new `geoms`, a different palette package than `RColorBrewer`, a different theme, changing the size 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, and a theme and colour palette other than the default. The aesthetic choices should add to its clarity rather than detract from it; part of what you are being marked on is if the figure illustrates the data in a clear and useful way.

(b) Explain what each of the two new elements are and how you made them. Your explanations don'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 (and panel, if you have multiple panels) as well as what the pattern is and what it suggests about what is going on. (It is fine for you to say there is no pattern and it suggests that nothing much is happening if that is what you observe!) You won't be evaluated on how interesting your result is, but on how clear and appropriate your explanation is given the figure. That said, it's worth thinking about what kinds of research questions would be interesting to look at, since those are more likely to yield interesting patterns which are easier to discuss.

[Suggested word count: 130]

Q8 [3% of total mark]

Gladly ran a statistical test and obtained a p -value of 0.07. "That means the null hypothesis is true according to the traditional alpha threshold of 0.05," he explains. "However, I'm going to set my alpha threshold to be 0.1 instead; that will make the test statistic significant, so I can conclude the null hypothesis is false instead." There are several distinct problems with Gladly's idea. Explain two of them to him. For each, be sure to be clear about what the problem is and why it is a problem.

[Suggested word count: 80]

Q9 [11% of total mark]

You are provided with a code chunk that calculates the highest and lowest audience scores in our dataset (called *highest* and *lowest* respectively). Note also that part (b) and (c) use the tibble that you used in Q5 called `df`. Regardless of whether or not you succeeded in completing Q5, you can use `df` for Q9.

(a) Bunny observes that on average, in past talent shows about 70% of the audience sample has liked any given act. If we presume that average describes this talent show as well, what is the probability of observing the highest score we saw? The lowest? You should answer these questions using the function(s) taught in Week 5; you do not need to use any of the datasets themselves. Report probabilities as percentages, rounded to one decimal place.

(b) Gladly points out that they have other data from previous talent shows as well, not just about audience ratings. For instance, in previous years the average duration was 6.5 minutes, with a standard deviation of 3. Shadow, inspired, writes the code given to you in the code chunk. What does the calculated variable *prob* reflect? How is this related to the idea of a p -value? Is it possible to identify which individual data points are significantly different from previous averages? If so, which ones, and why? If not, why not?

[Suggested word count: 100]

(c) Can we draw conclusions about how significant the entire variable *duration* (i.e., the full dataset of data about *duration*) is, based on a single calculation combining only the individual *prob* values? If so, explain why. If not, explain why not and what other information is necessary. Note that you do not need to do any calculations here; this is a thought question about Week 5 concepts.

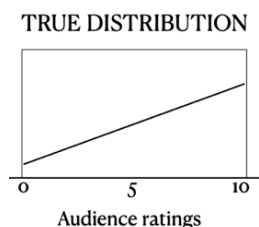
[Suggested word count: 130]

Q10 [8% of total mark]

It's evident from the data in Q6 that some kinds of talents have a much larger range of audience ratings than others. For instance, the range of magic tricks is 7 (i.e., with a low rating of 3 to a high rating of 10) while the range for singing and dancing is 4 (i.e., a low rating of 6 to a high of 10). Foxy starts wondering what kind of range one might *expect* to see in a random talent show, and how to determine if magic tricks are unusual.

Let's help her out! 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 range, which applies when thinking about this question. 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.

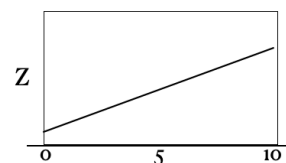
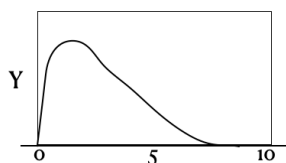
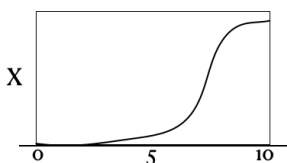
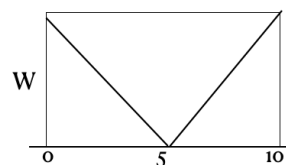
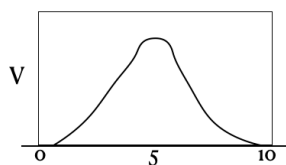
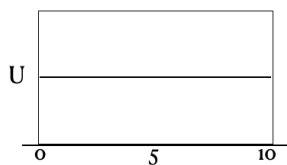
Foxy thinks that the *true* underlying distribution the audience ratings looks something like the figure directly below this paragraph: it's very unlikely for 0 people to like a performance, slightly more likely for exactly 1 people to like it, and so forth, with it being most likely that 10 audience members like it. For the purposes of this question, let's assume that she is correct and this is the true distribution.



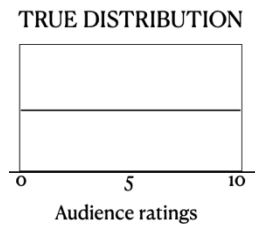
(a) Suppose talent shows become the next huge thing and as a result over the next few years there are 1000 talent shows. Each of the 1000 shows is divided into timeslots with 30 performances each. It is possible to calculate the range of audience rating for each of these timeslots.

Consider now the six panels U through Z below. Give the letter of the panel that most accurately captures what you expect the **sampling distribution of the range** to look like, on the assumption that the true distribution of audience ratings is as shown in the figure above. Explain your answer, making reference to the definition of sampling distribution and the figure. **Hint: begin by thinking about what you would expect the range for a single timeslot of 30 performances to be.**

[Suggested word count: 100]



(b) Suppose now that the underlying true distribution was uniform, as in the figure directly below this sentence.



How would this change your answer to part (a), if at all? Considering the same panels U through Z, give the letter of the panel that you would pick as being the closest answer in this case. Explain why. How is the behaviour of the sampling distribution of the range similar to and different from the behaviour of the sampling distribution of the mean, as the shape of the underlying true distribution varies?

[Suggested word count: 100]

* Note: You do not need to code or do any calculations in order to answer this question. This is a conceptual question designed to probe your knowledge about what a sampling distribution is. Moreover, if your intuition about the nature of a range are incorrect but your explanation of sampling distributions in general is solid, 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)