RMHI/ARMP Problem Set 1

Henrik Formoe 1575157 [Word Count: 1066]

Please put your answers here, following the instructions in the assignment description. Put your answers and word count tallies in the locations indicated; if none is indicated that means there is no word count for that question. Remember to knit as you go, and submit the knitted version of this on Canvas.

## Q1

**Q1a**

# Put your code here  
  
table(d$talent,d$level)

##   
## fun competitive  
## comedy 3 5  
## dancing 4 3  
## instrument 3 3  
## magic 2 4  
## other 2 0  
## singing 8 5

**Q1b**

# Put your code here  
   
d$talent <- factor(d$talent, levels = c("singing","dancing", "instrument", "comedy", "magic", "other"))  
  
table(d$talent)

##   
## singing dancing instrument comedy magic other   
## 13 7 6 8 6 2

*ANSWER: The most common talent was singing, with 13 performances.*

**Q1c**

# Put your code here  
d <- d %>% rename(species = kind)

## Q2

**Q2a**

# Put your code here  
  
d[(d$judge == 1 | d$judge == 2) & d$audience >= 8, ]

## # 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  
## # ℹ 16 more rows

**Q2b**

# Put your code here  
  
d %>%  
 filter((judge == 1 | judge == 2) & audience >= 8)

## # 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

**Q2c**

*ANSWER: The filter-function returns rows that match our condition. This is a part of dplyr, which again is a part of the tidyverse-package. On the other hand, we have the base R-approach where the use of the logic functions display the values that match the criteria put forward through the logic function. We do not extract anything, but apply our criteria to the table. In both cases the data has to be 1 or 2 in the judge-column, and 8 or above in the audience-column to either be extracted or displayed. Filter does not return NA, which the logic function does. [Word count: 100]*

**Q2d**

# Put your code here  
  
na.omit(d[(d$judge == 1 | d$judge == 2) & d$audience >= 8, ])

## # 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

## Q3

**Q3a**

# Put your code here  
  
dshort <- d %>% select (-c(audience, judge))  
head(dshort)

## # A tibble: 6 × 4  
## name species level talent   
## <fct> <fct> <fct> <fct>   
## 1 gladly bear competitive singing  
## 2 gladly bear fun dancing  
## 3 panda bear fun dancing  
## 4 snowy bear fun singing  
## 5 bunny bunny competitive magic   
## 6 bunny bunny fun comedy

**Q3b**

# Put your code here  
  
#(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.  
  
d2 <- dshort %>%  
 group\_by(name, species) %>%  
 mutate(  
 competitive = if\_else(level == "competitive", talent, NA\_character\_),  
 fun = if\_else(level == "fun", talent, NA\_character\_)  
 ) %>%  
 summarise(  
 competitive = first(na.omit(competitive)),  
 fun = first(na.omit(fun))  
 ) %>%  
 ungroup()  
  
 print(d2)

## # A tibble: 23 × 4  
## name species competitive fun   
## <fct> <fct> <chr> <chr>   
## 1 barky dog comedy <NA>   
## 2 black dog dancing comedy   
## 3 bunny bunny magic comedy   
## 4 cuddly paws bunny dancing instrument  
## 5 doggie dog comedy singing   
## 6 douglas hedgehog singing instrument  
## 7 flopsy bunny singing instrument  
## 8 fluffy cat singing other   
## 9 foxy fox dancing magic   
## 10 froggie frog instrument singing   
## # ℹ 13 more rows

**Q3c**

# optional code here

*ANSWER: In Q3a we only included data (dshort) that was relevant for the task in Q3b, then we applied our actions to only relevant data, before assigning it to d2. This helps ensure that our answer is correct without unecessary interference, e.g. if table d was filled with a big number of columns, it could be more difficult doing the actions in that table. Also, assigning it to a new table further allows us to continue working with the data. When I did try it in table d, I found it difficult to continue manipulating the new data. [Word count: 97]*

**Q3d**

# Put your code here  
  
# (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.  
  
# everybody needed to participate in both the fun and the competitive levels = fun & competitive   
# they could not present the same kind of talent at both levels. fun != competitive  
  
d2 %>%  
 filter(  
 is.na(competitive)  
 | is.na(fun)  
 | (fun == competitive)  
 )

## # A tibble: 6 × 4  
## name species competitive fun   
## <fct> <fct> <chr> <chr>   
## 1 barky dog comedy <NA>   
## 2 monkey monkey <NA> singing  
## 3 panda bear <NA> dancing  
## 4 quackers duck comedy comedy   
## 5 snowy bear <NA> singing  
## 6 tweak cat singing singing

*ANSWER: The names of the individuals who broke Rule 1 (i.e., that everybody needs to participate in both fun and competitive levels) are Barky, Monkey, Panda and Snowy. The names of the individuals who broke Rule 2 (i.e., that everybody must to do different kinds of talent at the fun and competitive levels) are Quackers and Tweak.*

## Q4

**Q4a**

# Put your code here  
  
d <- d %>%  
 arrange(name)  
   
head(d)

## # A tibble: 6 × 6  
## name species level talent audience judge  
## <fct> <fct> <fct> <fct> <dbl> <dbl>  
## 1 barky dog competitive comedy 7 3  
## 2 black dog competitive dancing 7 3  
## 3 black dog fun comedy 9 NA  
## 4 bunny bunny competitive magic 8 2  
## 5 bunny bunny fun comedy 10 NA  
## 6 cuddly paws bunny competitive dancing 8 1

**Q4b**

# Put your code here  
  
d\_full <- full\_join(d,dd)  
head(d\_full)

## # A tibble: 6 × 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

**Q4c**

# This code has been given to you, you just need to run it  
  
dc <- cbind(d,dd)  
head(dc)

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

*ANSWER: full\_join adds columns based on matching criteria (and does not duplicate data). cbind takes the two separate tibbles (e.g. y and x) and position them in that order: y, x – without removing any data. Therefore, full\_join only show e.g. name, level, talent once, and cbind displays these columns twice. [Word count: 49]*

## Q5

**Q5a**

# Put your code here  
  
df <- df %>%  
 mutate(durType = case\_when(  
 duration >= 10 ~ "long",  
 duration >= 5 ~ "medium",  
 duration < 5 ~ "short"  
 ))  
  
head(df)

## # A tibble: 6 × 8  
## name species level talent audience judge duration durType  
## <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <chr>   
## 1 barky dog competitive comedy 7 3 11.6 long   
## 2 black dog competitive dancing 7 3 5.3 medium   
## 3 black dog fun comedy 9 NA 9.9 medium   
## 4 bunny bunny competitive magic 8 2 8.9 medium   
## 5 bunny bunny fun comedy 10 NA 8.3 medium   
## 6 cuddly paws bunny competitive dancing 8 1 4.7 short

**Q5b**

# Put your code here  
  
ds <- df %>%  
 group\_by(talent) %>%  
 summarise(  
 medAud = median(audience, na.rm =TRUE),  
 mnAud = mean(audience, na.rm =TRUE),  
 sdAud= sd(audience, na.rm =TRUE),  
 n = n(),  
 sderrAud = sd(audience, na.rm = TRUE) / sqrt(n)  
 ) %>%  
 ungroup()  
  
head(ds)

## # 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

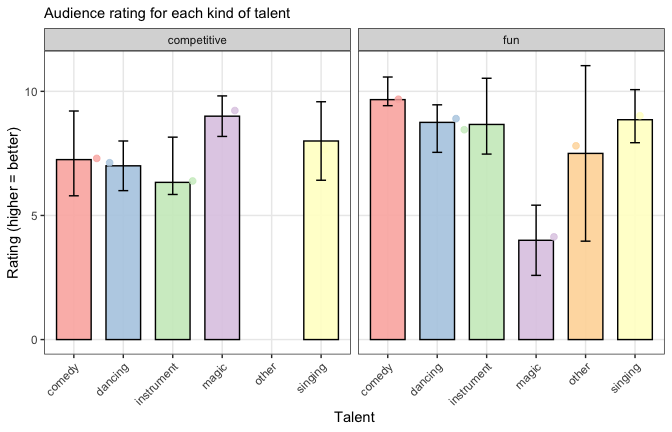
**Q5c**

*ANSWER: The mean audience rating shows that magic (mnAud = 7.33) was the least popular talent, and for median audience rating it was instrument (medAud = 7.0). Mean and median use different approaches for measuring typical tendencies in a set of data. Mean is found by adding all numbers together and then diving by number of observations, and median is the 50th percentile: the middle number when in ascending order. The discrepancy in our dataset can be explained by the distribution of values: less variation for instrument, and more “extreme” scores for magic. Therefore, magic has both a higher medAud and lower mnAud than instrument. [Word count: 104]*

## Q6

**Q6a**

# Put your code here  
  
 d6\_barplot <- ggplot(data = d6, aes(x = talent, y = mnAud, fill = talent,)) +  
 geom\_col(alpha = 0.9, color = "black", width = 0.7, show.legend = FALSE) +  
 scale\_y\_discrete(limits = c(0,5,10)) +  
 scale\_x\_discrete(guide = guide\_axis (angle=45)) +  
 geom\_jitter(aes(x = talent, y = mnAud, color = talent), position = position\_jitter(width = 0.5, height = 0.5), size = 2, alpha = 0.8, show.legend = FALSE) +  
 geom\_errorbar(aes(ymin = medAud - sdAud, ymax = medAud + sdAud), width = 0.2, position = position\_dodge(0.7)) +  
 facet\_wrap(~level) +  
 theme\_bw() +  
 labs(  
 subtitle = "Audience rating for each kind of talent",  
 x = "Talent",  
 y = "Rating (higher = better)"  
 ) +  
 scale\_fill\_brewer(palette = "Pastel1") +  
 scale\_color\_brewer(palette = "Pastel1")  
  
print(d6\_barplot)



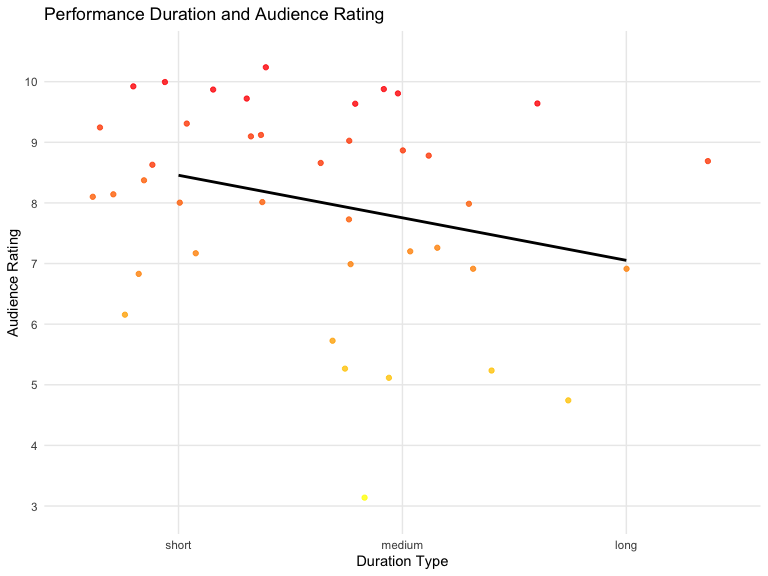
**Q6b**

*ANSWER: By looking at our descriptive data we are not able to draw inferences about casual relationships, but we may achieve some insights. When looking at the bar for magic and comparing the competitive and fun-levels, we notice a big difference in scores. Magic at the competitive-level received audience ratings between 8-10, and 3-5 at the fun-level. Also, we notice that the error bar for the other-category at fun-level is quite a lot bigger than for the other talents. The other-category contains both Rubik’s cube and a speech about napping, that also received very different ratings by the audience. On the other hand, comedy, dancing, and instrument received higher ratings at the fun-level. [Word count: 112]*

## Q7

**Q7a**

# Put your code here  
  
 df\_durAud <- df %>%  
 ggplot(aes(x = durType, y = factor(audience, exclude = NULL), colour = audience)) +  
 geom\_jitter(aes(fill = audience), show.legend = FALSE, alpha = 0.8) +  
 geom\_smooth(method = "lm", aes(group = 1), se = FALSE, colour = "black", show.legend = FALSE) +  
 scale\_y\_discrete(limits = 1:10) +  
 scale\_x\_discrete(limits = c("short", "medium","long")) +  
 theme\_minimal() +  
 scale\_colour\_gradient(  
 low = "yellow",  
 high = "red") +  
 labs(  
 title ="Performance Duration and Audience Rating",  
 x = "Duration Type",  
 y = "Audience Rating"  
 )  
   
   
 print(df\_durAud)



**Q7b**

*ANSWER: (1) First of all I decided to try out the geom\_smooth, which makes sense to show the linear model best fiting to our data points (jitter). (2) I decided to use the theme\_minimal and scale\_colour\_gradient which I haven’t used before now. [Word count: 40]*

**Q7c**

*ANSWER: My figure includes the duration type (short, medium, long) on the x-axis, and audience rating on the y-axis. The figure does not allow for causal inference, but could be interesting for discovering any trends or patterns between the audiences rating and the duration of the performance. The linear model in the figure may suggest that there was a negative correlation between rating and duration, in which the short performances received better ratings than medium, and medium revived better ratings than long. The dots in the figure are also graded from red = high value to yellow = low value, which makes it easier to see the distribution of audience rating based on duration (and not only duration type). This makes each of the audience ratings (1-10) distinguishable, beyond the three duration types. [Word count: 132]*

## Q8

*ANSWER: Gladly incorrectly believes that a p-value of 0.07 == the H0 “being true”. With a alpha threshold of 0.05: there is a 5% chance of falsely rejecting the H0, and a 95% chance of correctly rejecting it. Therefore, we should not make definitive statements, but rather use phrasing like e.g. the test was significant. Also, Gladly wants to adjust the alpha threshold after seeing the results. This is unethical and increases the possibility of false positives. Also, I’m sure Gladly already registered his research plan with the center of open science in Bunnyland - and therefore stick to his original plan (alpha < 0.05). [Word count: 104]*

## Q9

# get the lowest score  
lowest <- min(d$audience,na.rm=TRUE)  
lowest

## [1] 3

# get the highest score  
highest <- max(d$audience,na.rm=TRUE)  
highest

## [1] 10

**Q9a**

# Put your code here  
  
dbinom(x=10,size=10, prob=0.7)

## [1] 0.02824752

dbinom(x=3,size=10, prob=0.7)

## [1] 0.009001692

*ANSWER: The probability of the lowest score is 0.9% and the probability of the highest score is 2.8%*

**Q9b**

# this code is given  
df <- df %>%  
 mutate(prob = pnorm(duration,mean=6.5,sd=3))  
  
# you may add additional code here if it's useful to answer the question

*ANSWER: pnorm provides the probability of an outcome not exceeding a threshold. In this case we are interested in the duration of the performances, with the mean duration being 6.5 minutes and the sd = 3. The given value is related to p-value since it tells us how many who received the same score or lower in the dataset, e.g. 6.5 min == mean, and the score 12.2 min == 0.97 (duration is close to 6.5 mean + 2 SD). We can identify outliers in the dataset, like Fluffy (18.7 min, prob = 100%) and Shadow (1.4 min, 4.5%). [Word count: 98]*

**Q9c**

*ANSWER: No, it’s not possible to draw conclusions about how significant an entire variable is, based on a single calculation combining the individual prob values. We would need data about the dataset, like central tendencies etc. [Word count: 35]*

## Q10

**Q10a**

*ANSWER: Based on Foxy’s assumption we would expect the sampling distribution of the range (SDR) to most closely resemble panel Y. If the panel was to resemble a normal distribution our data would need to be based on means of lots of samples. Our measure SDR is not based on means, but based on the distance of extreme values (lowest and highest). Therefore, as explained, it is very unlikely for 0 people to like a performance, and most likely that 10 people like it. Making the SDR more likely to peak closer to the smaller values, with a tail towards the increasing range. [Word count: 102]*

**Q10b**

*ANSWER: With a true distribution being uniform, we would expect the SDR to resemble panel U. The difference between sampling distribution of the mean and SDR is first of all that mean is a measure of central tendencies where as SDR is a value reflecting the difference between the lowest and the highest values. A similarity between these two measures are that they may be susceptible to influence by extreme values. With a uniform true distribution all values would be as likely, which is why the range between the values also is equally likely. [Word count: 93]*

## Q11

*ANSWER: I’m not sure, but my guess would be that they either 1) that they are misunderstanding some of their data and need to attend RMHI in Bunnyland to learn this properly, or 2) they haven’t correctly operationalized what they are trying to measure.*