Worksheet 03 - Producing Data and Basics of Probability

##Shawn

***Directions**: Please upload a PDF to Gradescope that includes both your written responses and corresponding R code inputs/outputs (if requested) for each problem. In code blocks, you can uncomment (delete #) the non-NOTE lines and fill in the blanks (FILL IN).*

Problem 1 A health study is being conducted on a group of volunteers (487 smokers and 1513 non-smokers) to determine the effect of a new drug.

Problem 1 Part a) Estimate how many smokers are in a simple random sample of size 200.

NOTE: Enter your response as text or an image. See R Worksheet 02 for the code for inserting a picture.

```
(487/2000) * 200
```

```
## [1] 48.7
```

49 smokers in a simple random sample size of 200

Problem 1 Part b) Using R, determine 10 simple random samples, each of size 200, and record the number of smokers in each of the samples. Let's agree to label the smokers with numbers 1 through 487 and the nonsmokers with numbers 488 through 2000.

NOTE: The code in the R chunk below walks through the calculation in R, where the variable `tmp_sample` denotes one SRS of size 200 taken from the population, while the variable `smoker_count` stores the number of smokers in the k-th SRS.

```
set.seed(2022) #NOTE: This makes the experiments able to be duplicated in this worksheet.
population = seq(from = 1, to = 2000, by = 1) # create a sequence of numbers 1 to 2000 by 1
smoker_count = rep(0, 10) # create vector to ultimately tally the number of smokers in 10 SRS

for (k in 1:10)  #k is a loop variable that starts at 1, increments by 1, and ends at 10
{
  #NOTE: { starts the loop and } ends the loop (THESE ARE VERY IMPORTANT)
# NOTE: The call sample() should include the vector to be sampled (population), along with the number t

tmp_sample = sample( population , 200, replace = FALSE )

# NOTE: The call to sum() should be an expression in terms of tmp_sample. For example, sum(tmp_sample <

smoker_count[k] = sum( tmp_sample < 488 )
}

# NOTE: Now, let's make a data frame to store the counts nicely, wrapping it in parenthesis so that the

(smoker.df = data.frame(trial = 1:10 , nsmokers = smoker_count ))
```

```
##    trial nsmokers
## 1      1       46
## 2      2       54
## 3      3       48
## 4      4       48
## 5      5       52
## 6      6       55
## 7      7       41
## 8      8       58
## 9      9       48
## 10    10       48
```

Problem 1 Part c) Using R, determine the mean and the standard deviation of the number of smokers in the samples. How does the sample mean conform (i.e., is it far/close) to your calculation for the predicted population mean?

Hint: When describing how far/close the sample mean is to the population mean, it is a good idea to reference the standard deviation in your description.

NOTE: For the first part, enter the necessary code in the R chunk below. For the last part (the question), enter your response as text outside/below the R chunk.

```
# ENTER CODE HERE
mean(smoker.df$nsmokers)
```

```
## [1] 49.8
```

```
sd(smoker.df$nsmokers)
```

```
## [1] 4.962078
```

My prediction of 49 (actual value 48.7) for the population mean is close to the sample mean of 49.8 and falls within one standard deviation of the sample mean.

Problem 1 Part d) The head researcher would like to choose 20 subjects for comprehensive medical imaging. Using R, perform a single stratified random sample having 10 smokers and 10 nonsmokers, and display the labels for the selected subjects in ascending order.

```
# NOTE: The command sample(x, size) chooses 'size' items from x.

# NOTE: Make sure you sample 10 from the population of 487 smokers!
tmp_smokers = sample(population[1:487], 10)

# NOTE: Make sure you sample 10 from the population of 1513 non-smokers!
tmp_nonsmokers = sample(population[488:2000], 10)

# NOTE: Let's make a data frame to display our results!

# NOTE: You will use the sort() to display the labels for the selected subjects
# in ascending order

# NOTE: FILL IN the sorted simulated data below
(smoker.df = data.frame(smokers = sort(tmp_smokers), nonsmokers = sort(tmp_nonsmokers)))
```

```
##    smokers nonsmokers
## 1       26        634
## 2       42        727
## 3      160        759
## 4      166        806
## 5      237        988
## 6      265       1034
## 7      293       1043
## 8      432       1580
## 9      452       1586
## 10     465       1975
```

Problem 2. Most desert tortoises live in creosote bush scrub habitat at elevations ranging from 1,000 to 3,000 feet above sea level. Their habitat covers a relatively large region including the Mojave and Sonoran Deserts. In some areas, the number of desert tortoises has decreased by 90% due primarily to human activity. A study area in Organ Pipe National Monument has 300 Sonoran desert tortoises. Fifty are captured, tagged, and released. A certain time later, 15 of the 300 are captured.

Problem 2 Part a) What is the probability that exactly one of the 15 in the second capture is tagged? State clearly what assumptions you are using.

Hint: If 15 tortoises are captured and exactly 1 is tagged, how many non-tagged tortoises are there in the captured sample? How many ways are there to capture these non-tagged tortoises?

```
# r uses the function choose(n,x) notation where n is the total number and x is
# the the number that you want to select.

# ADD r code here

(choose(250, 14) * choose(50, 1))/choose(300, 15)
```

```
## [1] 0.191826
```

The probability that exactly one is tagged out of 15 is 0.191826. My assumptions were that first, we are choosing 15 without replacement, so then we need to find the number of ways (combinations) to pick 15 tortoises with only one of them being tagged. To calculate that, we need to find the number of ways we can pick 14 non-tagged tortoises which is "250 choose 14" and number of ways we can pick 1 tagged tortoise which is "50 choose 1" (which is 50). Times these two numbers together and you get the total number of ways you can pick 15 with exactly 1 tagged. Divide by the total number of ways you can pick 15 tortoises which is "300 choose 15" and you get the probability of picking 15 with exactly 1 tagged. This calculation is shown in R above

Problem 2 Part b) Find the probability that $x = 0, 1, 2, \cdots, 15$ that are tagged. Check that the sum is 1.

```
tmp_probs <- rep(0, 16)  #zero vector for storage with 16 entries (for 0-15)

# NOTE: begin loop
for (k in 0:15) {
    # start at entry 1; k+1->0+1=1, needed because r starts counting at 1 not 0
    tmp_probs[k + 1] <- choose(250, 15 - k) * choose(50, k)/choose(300, 15)
}  #end loop

# NOTE: create data.frame to display data
(tagged_probs <- data.frame(no_tags = 0:15, probs = round(tmp_probs, 4)))
```

```
##    no_tags  probs
## 1        0 0.0604
## 2        1 0.1918
## 3        2 0.2776
## 4        3 0.2426
## 5        4 0.1431
## 6        5 0.0604
## 7        6 0.0188
## 8        7 0.0044
## 9        8 0.0008
```

```
## 10       9 0.0001
## 11      10 0.0000
## 12      11 0.0000
## 13      12 0.0000
## 14      13 0.0000
## 15      14 0.0000
## 16      15 0.0000
```

```r
# NOTE: check that sum=1 since all possible events are covered
sum(tmp_probs)
```

```
## [1] 1
```

Problem 2 Part c) Use the **sample** command to perform 10000 simulations of the capture, tag, and release situation described above. Give a table of outcomes and compare the proportion in the simulation to the probabilities computed in part (b).

```r
# NOTE: one simulation would be NOTE: tmp_sample <-
# sample(1:300,15,replace=FALSE) #15 samples from 300 NOTE: no_tags <-
# sum(tmp_sample < 51) #sum up entries that were in the first 50 NOTE: we need
# to create a loop to do this 10000 times

no_sims <- 10000  #number of simulations
no_tags_10k <- rep(0, no_sims)  #zero vector for storage with 10000 entries

# NOTE: begin first loop
for (k in 1:no_sims) {
    tmp_sample <- sample(1:300, 15, replace = FALSE)  #15 samples from 300
    no_tags_10k[k] <- sum(tmp_sample < 51)  #sum up entries in the first 50
}  #NOTE: end first loop

tmp_counts <- rep(0, 16)  #zero vector for storage with 16 entries (for 0-15)

# NOTE: begin second loop
for (k in 0:15) {
    # start at entry 1; k+1->0+1=1 sum up entries with k previously tagged
    tmp_counts[k + 1] = sum(no_tags_10k == k)
}  #NOTE:end second loop

# adds new column to table in part b
(tagged_probs <- data.frame(tagged_probs, sim_probs = round(tmp_counts/10000, 4)))
```

```
##    no_tags  probs sim_probs
## 1        0 0.0604    0.0598
## 2        1 0.1918    0.2028
## 3        2 0.2776    0.2749
## 4        3 0.2426    0.2374
## 5        4 0.1431    0.1440
## 6        5 0.0604    0.0588
## 7        6 0.0188    0.0175
## 8        7 0.0044    0.0042
## 9        8 0.0008    0.0005
## 10       9 0.0001    0.0001
```

```
## 11      10 0.0000    0.0000
## 12      11 0.0000    0.0000
## 13      12 0.0000    0.0000
## 14      13 0.0000    0.0000
## 15      14 0.0000    0.0000
## 16      15 0.0000    0.0000
```

the proportions in the simulation are fairly close to the probabilities computed in part b which lets me know that my probability equation was probably correct