RNR 321 • Simple Random Sampling • Due 4 October 2021

There are two parts to this assignment. For each part, you'll collect your answers in the Excel template available on D2L (**SRS_Template.xlsx**).

Part I: Use simple random sampling to characterize the population of dark stones embedded in the sidewalk outside Marley. We'll consider each section of sidewalk as a unique population.

Steps:

1. Form teams of three and choose a section of sidewalk as your population.
2. Measure the length and width of the section (should be about 32" x 32").
3. To establish sample units, divide the section into a grid with 4" spacing and number the squares from 1 to *N* (see image for the layout below). **Be sure to record *N* in the Data Table below**, which is the total number of sample units in the population (i.e., the sampling frame or universe).
4. Use the table of random numbers on the back of this page to choose *n* = 8 units to survey.
   - Pick a random location in the table, then select eight consecutive values in any direction (left, right, up, down). If a value is greater than *N*, skip it and use the next value.
   - Record the values you selected in the first column of the **Data Table** below.
5. For each sample unit you selected, count the number of dark stones (only those bigger than a raisin) and record the counts in the Data Table below. If a stone is on the border of the sample unit, include it in your count only when half or more of the stone is in the unit you are surveying.
6. Once in the lab, enter your data into the yellow cells of the **Part I** sheet in the Excel template.

Problem 1. Estimate the <u>mean</u> number of black stones per sample unit in the population and the <u>total</u> number of black stones in the population.

Use the sample data you collected and the R code provided in **Part I** of **SRS_Lab_R_Code.R** and **Tips for using R** at the end of this document to estimate the quantities listed in **Part I** of the Excel template. Enter your estimates from R into the green cells in the template.

Example grid layout:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Data Table: (*N* = ______)

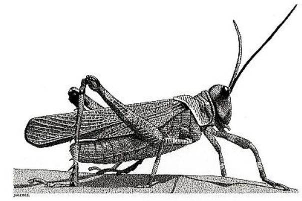
Cell Number	No. Black Stones

Random Numbers for Part I

38	19	59	16	16	60	61	41	14	35	10	41	12	58	36	48	26	25
25	37	47	17	19	34	42	59	29	21	25	21	31	58	29	15	18	52
16	28	49	17	10	43	28	19	29	6	51	50	62	25	20	11	30	1
26	36	46	34	35	30	30	43	49	20	21	2	4	8	42	13	23	6
35	49	2	31	29	49	52	7	19	41	12	21	63	59	30	61	6	2
23	45	40	33	36	53	16	53	42	22	44	12	3	42	51	30	62	55
10	34	54	12	20	11	53	14	49	21	9	28	34	53	59	12	33	1
24	56	18	48	56	45	13	2	5	53	24	15	54	17	53	35	18	56
6	48	9	47	28	47	36	36	44	8	27	16	14	42	60	19	58	32
41	28	62	22	29	43	61	15	54	35	1	50	23	54	36	8	17	50
24	8	39	40	60	59	11	49	53	23	36	10	21	27	61	33	31	37
64	38	43	13	27	2	30	15	48	15	61	41	62	26	31	12	1	44
54	33	20	57	52	29	44	38	21	25	40	10	32	64	27	40	21	63
36	4	57	2	41	48	28	11	1	14	17	17	17	27	11	47	17	16
57	5	64	55	16	34	48	61	25	19	1	64	45	64	29	61	5	63
51	44	33	64	51	23	33	16	42	49	21	59	26	43	34	29	25	30
42	34	49	59	58	13	4	50	32	3	36	16	28	31	10	13	18	37
48	5	7	43	50	49	25	34	45	56	42	20	1	37	10	43	3	24
60	37	16	59	54	15	12	48	31	16	14	49	38	49	25	57	46	25
17	59	54	50	7	2	39	61	54	48	20	42	8	46	39	46	7	8
45	20	52	22	8	14	35	21	11	10	62	20	60	29	40	25	11	46
64	13	34	46	49	59	43	21	25	47	62	34	31	30	62	8	47	32
30	47	15	55	43	2	10	18	40	17	48	23	56	47	8	22	17	24
15	61	61	56	24	5	3	20	1	49	48	16	35	21	4	23	21	59
57	23	12	9	54	10	38	2	61	48	36	21	28	52	60	54	11	14
19	8	14	48	37	1	59	7	28	18	13	25	49	22	37	44	15	48
49	37	59	24	37	5	29	7	54	24	64	45	9	43	63	36	1	42
40	2	59	22	58	47	45	20	30	1	52	24	58	51	28	64	1	22
64	20	46	54	8	54	25	17	3	2	37	51	49	59	17	35	20	13
49	28	46	52	11	55	49	9	3	40	41	3	8	25	55	39	49	7
63	49	2	43	29	32	17	60	54	28	15	10	28	10	20	64	27	45
3	46	57	13	32	36	47	1	51	39	53	13	28	39	59	53	4	55
55	64	16	12	46	2	37	53	62	36	9	19	19	52	58	7	37	25
12	3	10	41	27	35	34	29	3	59	53	56	32	44	13	21	50	10
9	53	42	8	43	24	1	58	19	7	25	20	10	45	50	44	13	13
9	63	37	23	32	63	21	59	3	43	36	51	28	39	28	19	64	45
24	33	21	13	43	5	15	17	48	15	40	54	18	11	48	16	15	46
43	27	6	3	56	38	59	22	64	27	55	63	49	12	17	26	24	45
30	19	40	22	50	2	39	63	15	27	14	54	5	1	59	5	54	53
55	51	45	37	42	14	53	19	49	54	28	48	20	64	43	55	32	7
61	50	18	50	22	50	29	37	4	18	24	22	4	50	48	55	5	48
53	54	1	46	38	36	30	25	5	17	9	52	11	49	61	61	49	18
31	44	50	42	32	39	21	1	24	32	13	26	33	32	21	64	27	63
51	4	22	62	27	41	48	31	61	30	6	13	10	16	41	28	43	31

Part II: Understanding bias and precision, which are two *unrelated* components of accuracy.

We wish to characterize a hypothetical population of horse lubbers (*Taeniopoda eques*), the largest grasshopper in southern Arizona. We've divided the study area into 1,000 plots each of which is 1 hectare (plot is the sample unit), and we'll use Simple Random Sampling to estimate two parameters for the study area, the mean number of insects per plot and the total number of insects in the population. Unlike real-world sampling, this is a simulation, so we know the *true* values of the population parameters (e.g., the true mean number of grasshoppers per plot, *mu* or μ, and the true total number of grasshoppers in the population, *tau* or τ).



Our goals are to:

1. Evaluate how well sampling works by comparing estimates from sample data to the true values of the parameters for the simulated population.
2. Assess the effects of samples size on bias and precision of estimates.

To explore how sampling works in general, we'll use R to draw multiple samples from the population and explore those as a set, which will allow us to evaluate bias and precision.

Follow the instructions that start at "**### --- Part II --- ###**" in this script on D2L: `SRS_Lab_R_code.R`

Problem 2. Copy the values you generated in the script to the green cells in the **Part II** sheet of the Excel template:

A. For each of the three parameters and for both sample sizes (n = 15 and n = 90), use Excel to compute:
   - **bias = true value – estimate**; that is, the difference between the true value of the parameter and the average of the estimates from sample data
   - **percent relative bias = (true value – estimate) / true value * 100**

   Enter your calculations in the ORANGE cells in the spreadsheet.

B. Note that variance among the multiple estimates of the same parameter is one way to measure **precision**.

Problem 3. In a MS Word file, answer these questions:

A. How did **bias** and **precision** of the sample estimates of the population mean and population total vary between samples of size of n = 15 and samples of size n = 90? Specifically, evaluate and contrast the values for **percent relative bias** and **precision** you have from Problem 2.

B. In general (that is, without assuming your specific results worked out ideally), how would you expect accuracy (that is, both bias and precision, considered separately) of your estimates (e.g., the mean and total) to change with sample size? Is that what you observed in part 3A?

Turn In (D2L): Answers to Problem 3 in a **MS Word file**, your **Excel template**, and your **R script**.

Tips for using R – these are all illustrated in the script **'SRS_Lab_R_code.R'**

Remember, always save your work in a script file.

Create an object (vector) to hold your sample data (I named mine "rocks"):

```
rocks <- c(15,7,13,14,8,12,6,9)
```

Use R's built-in functions whenever possible

```
mean_rocks <- mean(rocks)
var_rocks <- var(rocks)
```

A useful function is length() that counts the number of items in an object

```
n_rocks <- length(rocks)
```

When a function is not available within R, such as for computing the standard error, you can (1) create your own formula or (2) create a function for the formula that you can refer to repeatedly. For example, we can compute standard error of the mean with the objects we defined above:

```
SE_rocks <- sqrt(var_rocks/n_rocks)
```

Alternatively, we can compute the identical quantity by "nesting" the functions themselves in the formula rather than first creating the objects themselves (note the use of var() and length() functions):

```
SE_rocks <- sqrt(var(rocks)/length(rocks))
```

Last, we can define our own function for SE that we can reuse. First, define the function in terms of a generic object, x:

```
SE <- function(x) sqrt(var(x)/length(x))
```

Once it's defined, you can send any object to the function:

```
SE(rocks)
```

You can save the output to an object in the usual way so that you can reference it later say for computing CIs:

```
SE_rocks <- SE(rocks)
```

Finding *t*-values for confidence intervals in Excel and in R

To find the *t*-value needed to compute the half width of a confidence interval:

In Excel: use the function **T.INV.2T(alpha, df)**, where **alpha** is the error rate and **df** is degrees of freedom; note this function provides the two-tailed *t*-value, so no need to divide alpha by 2. For example, if alpha = 0.05 and df = 19, you would use T.INV.2T(0.05,19), which yields $t = 2.09$.

In R: use the function **qt(p, df)**, where **p** is the percentile of the *t*-distribution (**p = 1 – alpha**) and **df** is degrees of freedom. For example, if alpha = 0.05 and df = 19, you would use qt(1-(0.05/2), 19), which yields $t = 2.09$.