

The notion of sampling in probability and statistics is an important one. To **sample** from a distribution means that we generate values created by that particular distribution. Since computer systems can't roll dice or flip coins, they rely on algorithms to simulate the experiments in question.

Commands/Instructions needed for R:

- (1) Sampling random numbers:
 - (a) The command

```
sample(a:b,1,replace=TRUE)
```

generates a *single* random integer between **a** and **b**.

note: the “replace=TRUE” part of this command allows the same number to appear more than once

- (b) To generate a list of **n** random numbers, use the following command:

```
sample(a:b,n,replace=TRUE)
```

- (2) List and data frame creation:

- (a) To make a list named `listname` of some numbers, you can use

```
listname=c(1,62,412,32,9,-6)
```

- (b) To make a list of 15 random numbers between 1 and 6, first you need to come up with a name for the list, and then you use “=<appropriate_command>”. For example,

```
s1=sample(1:6,15,replace=TRUE)
```

will generate the 15 numbers mentioned and store that list of numbers as **s1**

- *note: To create multiple lists, you can copy/paste commands. However, if you don't change the name of the list, it will overwrite any previous version.*

- (c) Data Frames—these are basically tables or spreadsheets. The command

```
data.frame(<column1title>=<data1>,<column2title>=<data2>,...)
```

will create a table with the given column names, and the appropriate data. So, if we wanted to create two columns of data for two separate draws of 15 numbers each, we could store them together in a data frame called `ExpData1` in the following way:

```
ExpData1=data.frame(Trial1=sample(1:6,15,replace=TRUE),Trial2=sample(1:6,15,replace=TRUE))
```

Of course, you can create as many columns as you want.

- (3) Mean and Standard Deviation:

- (a) The mean of a data set can be found using the command `mean(<listname>)`. To find the mean of a particular column in a data frame, you need to do something like

```
mean(<dataframename>$<columnname>)
```

- (b) The `sd` function in RStudio gives the sample standard deviation. It works the same way that `mean` does. Also, the sample variance can be found using `var()`

Problems:

- (1) Generate a list of 2,347 random numbers between 38 and 913, and call it `RandNum1`.
 - (a) Calculate the mean, sample variance, and sample standard deviation for `RandNum1`
 - (b) Submit the correct command(s) you used to generate the list and the command(s) you used to find its mean.

- (2) In each of the problems below, a “trial” refers to you rolling a die 10 times and recording the outcomes.
 - (a) create a table named `Problem2` with 6 columns, each of which represents a different trial. The column names should be `Trial1`, `Trial2`, etc.
 - (b) Find the mean of each column, labeling them `m1`, `m2`, etc.
 - (c) Create a list called `Means10` consisting of `m1`, `m2`,... etc.
 - (d) Submit the mean and standard deviation of `Means10`

- (3) In each of the problems below, a “trial” refers to you rolling a die 100 times and recording the outcomes.
 - (a) create a table named `Problem3` with 6 columns, each of which represents a different trial. The column names should be `Trial1`, `Trial2`, etc.
 - (b) Find the mean of each column, labeling them `h1`, `h2`, etc.
 - (c) Create a list called `Means100` consisting of `h1`, `h2`,... etc.
 - (d) Submit the mean and standard deviation of `Means100`

- (4) In each of the problems below, a “trial” refers to you rolling a six-sided die 1000 times and recording the outcomes.
 - (a) create a table named `Problem4` with 6 columns, each of which represents a different trial. The column names should be `Trial1`, `Trial2`, ... etc.
 - (b) Find the mean of each column, labeling them `t1`, `t2`, etc
 - (c) Create a list called `Means1000` consisting of `t1`, `t2`,... etc.
 - (d) Submit the mean and standard deviation of `Means1000`