IST687 – Lab – Distributions and Functions - Generate Distributions

A key focus of our class this week has been "distributions." A distribution is simply an arrangement of values of a variable such as the population size of a state. A "probability distribution," is an arrangement of all the values (potential outcomes) of a variable that reflect the frequency of those values in nature. A distribution can either be empirical, which means that it is an actual bunch of numbers, or it can be theoretical, in which case we are just imagining an ideal arrangement of numbers. The normal or "bell" curve is just such a theoretical distribution.

The R open source statistical system is great at creating empirical distributions that are made up of randomly generated numbers. The book includes several commands and explanations of randomly generated distributions. We can explore called a "Pareto" distribution. We can use R to generate a Pareto distribution of state populations that may be quite similar to the populations of the actual U.S. states. In other words, we can generate random numbers for the sizes of the Fictional States of America.

Task 1: Write, test, and submit the necessary code in R to accomplish the following:

1. Generate a normal distribution, or 1000 samples, with a mean of 80
2. Write a function that takes three variables – a vector, a min and a max, and returns the number of elements in the vector that are between the min and max (including the min and max)
3. Use the function to see how many of your normal distribution samples are within the range of 79 to 81
4. Repeat 3 times (creating a normal distribution and then calling your function), to see if the results vary

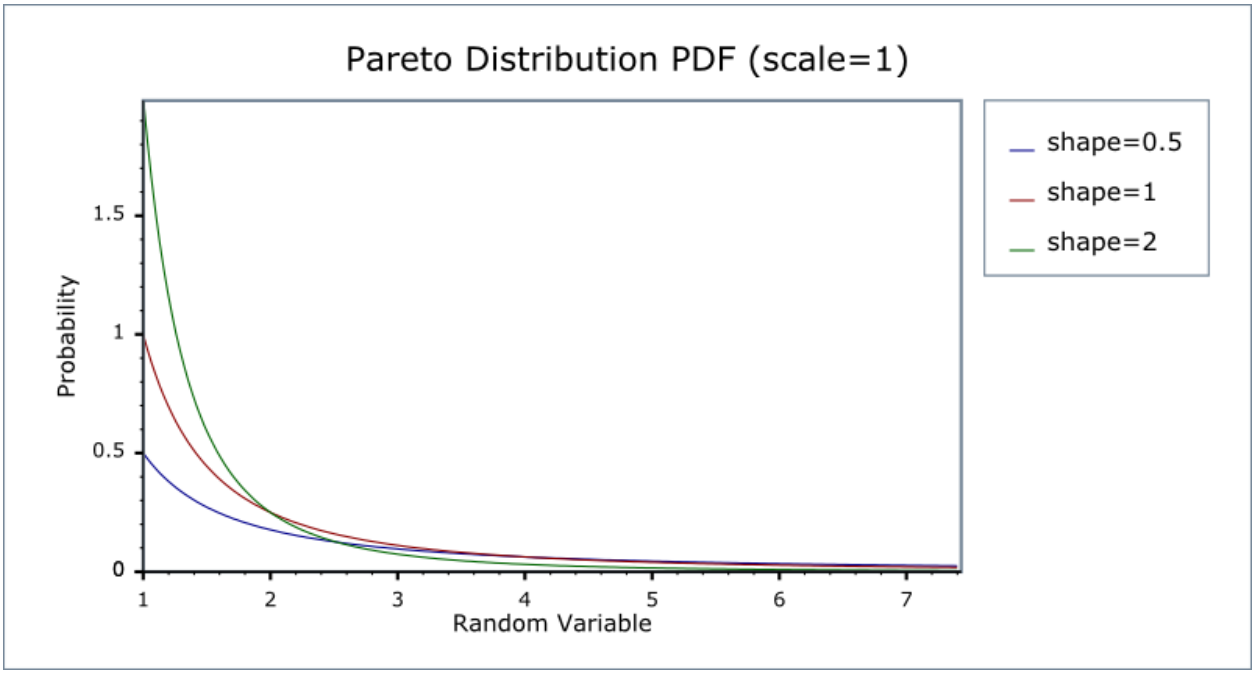
Task 2: Write, test, and submit the necessary code in R to accomplish the following:

1. Generate 51 random numbers in a Pareto distribution and assign them to a variable called "FSApops."
2. Specify a "location" and a "shape" for your Pareto distribution that makes it as similar as possible to the actual distribution of state populations.
3. Create a histogram that shows the distribution of values in FSApops.
4. Use a command to report the actual mean and the actual standard deviation of the 51 values stored in FSApops.
5. Use a command to report the population of your largest fictional state (i.e., your California) and your smallest fictional state (i.e., your Wyoming).

Hints:

- The necessary R command for generating random numbers in a Pareto distribution is located in a "package" called "VGAM". The code you submit should include the two necessary commands for making this happen (they should be the first two commands in your code). In the VGAM package, you will find a command for generating random numbers that fit a Pareto distribution.

- You will have to look up the meaning of the location and shape parameters so that you can figure out how to set them to make your Fictional States of America as similar to the real states as possible.
    - The scale/location parameter sets the position of the "left edge" of the probability density. The only outcomes that can be observed are greater than the value of the scale/location parameter. Your scale location has to be > 0.
    - The shape parameter determines the steepness of the "ski slope."
    - You will determine your shape and scale based on the US states population data we have been using. One way to find a good distribution is to vary the parameters. You are encouraged to play around with these numbers and get it as close to looking like our state pops as you can. In other words, change (or "play" with) the location and shape to see how they influence the distribution, and then pick location and shape that make the distribution closest to what you want.
- Note that random numbers will differ substantially every time you run the command, so we don't expect your data to be a perfect match. You do want your smallest state to be about the size of Wyoming and about 15 of your states to be under 2 million in population.
- A pareto distribution will look like a ski slope .. high on the left and a long tail down hill to the right (see pic below).



Learning Goals for this activity:

A. Generate random numbers in a Pareto distribution and assign a variable name.
B. Specify a "location" and a "shape" for a distribution to conform to a model.
C. Create a histogram depicting a distribution of values.
D. Use R commands to report mean and standard deviation.
E. Use appropriate R command to report the most extreme values of a variable.

Essential Guide for All IST687 Activities (appears at the end of all activity guides)

1. All IST687 activities work on what some people call a "constructivist learning" model. By developing a product on your own, testing it to find flaws, improving it, and comparing your solution to the solutions of other people, you can obtain a deeper understanding of a problem, the tools that might solve that problem, and a range of solutions that those tools may facilitate. The constructivist model only works to the extent that the student/learner has the drive to explore a problem, be frustrated, fail, try again, possibly fail again, and finally push through to a satisfactory level of understanding.
2. Each IST687 activity builds on skills and knowledge developed in the previous activities, so your success across the span of the course depends at each stage on your investment in earlier stages. Take the time to experiment, play, try new things, practice, improve, and learn as much as possible. These investments will pay off later.
3. Using the expertise of others, the Internet, and other sources of information is not only acceptable - it is expected. You must ***always, always, always*** give credit to your sources. For example, if you find a chunk of code from r-bloggers.com that helps you with developing a solution, by all means borrow that chunk of code, but make sure to use a comment in your code to document the source of the borrowed code chunk. The discussion boards in the learning management system have been setup to encourage appropriate sharing of knowledge and wisdom among peers. Feel free to ask a question or pose a solution on these boards.
4. Building on the previous point, when submitting code as your solution to the activity, the comments matter at least as much, if not more than the code itself. A good rule of thumb is that every line of code should have a comment, and every meaningful block of code should be preceded by a comment block that is just about as long as the code itself. As noted above, you can use comments to give proper credit to your sources and you can use comments to identify your submission as your own.