11/1 in Class – First Simulations

MATLAB has two different random number generators: `rand` and `randn`. `rand` generates a flat distribution and `randn` generates a Gaussian, or Normal, distribution.

We also learned about a plotting style called a histogram that counts the number of events in a given range. The command is `histogram(variable)`

1. Write code to simulate rolling a die 100 times.
   (a) Make two plots: one a regular plot with circles as the data points from each roll.
   (b) The second plot should be a histogram of the rolls.
   (c) Label the plots
   (d) Play with the number of times you roll the die. How many times do you need to roll until the distribution looks flat?
   (e) Show me the plots.
2. Write code to simulate rolling two dice 100 times. I'm interested in the sum of the two die each time.
   (a) Make two plots: one a regular plot with circles as the data points for the sum from each roll.
   (b) The second plot should be a histogram of the sum of each roll.
   (c) Label the plots
   (d) Play with the number of times you roll the die. This distribution might look a bit like a Gaussian, but it's not. It will probably not surprise you to learn this is called a triangular distribution.
   (e) Show me the plots
3. There have been several theoretical predictions about the decay probabilities of the Higgs Boson. For a Higgs with a mass of 125 GeV: (high energy physicists leave everything in units of $\hbar = c = 1$, so when you see units of energy (GeV) attached to a mass, you read GeV/c^2, which makes the units work.)

% chance	decay products
60%	b bbar
21%	WW
9%	gg
5%	$\tau\bar{\tau}$
2.5%	$c\bar{c}$
2%	ZZ
0.2%	$\gamma\gamma$
0.15%	γZ
0.15%	other, too small to list each

Make a histogram of Higgs decay particles. Given your previous known distributions and how many times you had to roll, run enough decays that you would expect to see a fairly accurate

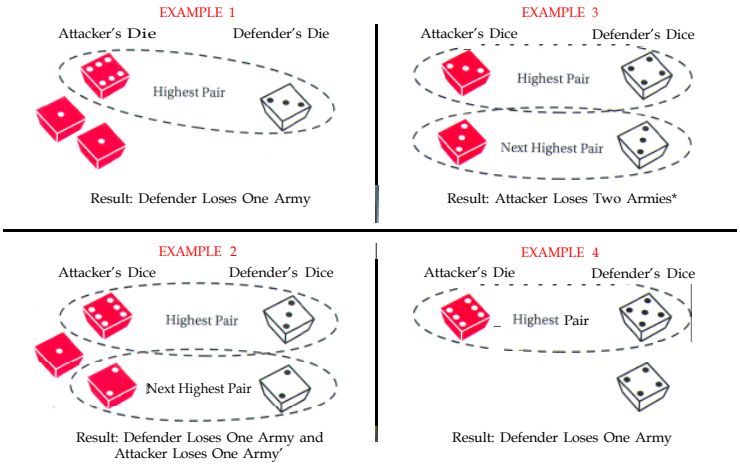
distribution. Label the bins (x-axis of histogram) with the decay products. You will have to look up how to do this. Hints: the slickest way is to use `categorical` with `discretize`

Show me the histogram

(If you would like a nice summary of the physics of Higgs decays, see: https://profmattstrassler.com/articles-and-posts/the-higgs-particle/the-standard-model-higgs/decays-of-the-standard-model-higgs/)

4. **A simplified Risk attack**: Risk is a game where you use dice to determine the outcome when one player attacks the other. The number of armies you control dictate the number of dice you can roll, up to a limit of three dice for the attacker (red) and two dice for the defender (white). (Advantage to the attacker here, they might get an extra die.) Let's say that you have to have one army for each die that you roll, and that you must roll the number of dice = to the number of armies you have, up to three dice for the attacker and two for defender. Once the attacker is down to two armies, they can only roll two dice. Once the defender is down to one, they can only roll one die. (These last three sentences are an oversimplification of the real rules to make it simpler to code.)

   Both players roll at the same time. You compare the dice in pairs to determine the winner. First you take the highest of each player's rolls and compare them. If the attacker's highest roll is higher than the defender, the defender loses one army. If the defender's roll is equal or higher, the attacker loses an army. (Advantage to defender here, since they win on a tie.) Then you take the next highest of the attacker, and the next highest (only other one) of the defender and do the same comparison.



Let's consider the case where the attacker has 5 armies and the defender has 5 armies. (You might not attack in the real game if you had a choice, but let's see what happens.) Run the simulation of this case lots of times and predict who will win. (The winner is the one who keeps an army at the end.)

Hint: at some point, you might want to use the `while` command. It is a bit like a `for` loop, but you don't need to know how many times it will run. For example, you can keep doing something `while` the number of armies > 0.