1. Instructions

- You cannot look at anyone else's code.
- All code (except import statements) in homework1.py should be inside functions (importing homework1.py should not cause code to execute).
- To check your code, use the provided test files

2. Homework Submission Checklist

Read this carefully!!!

- Your homework must work with Python 3.5 or later (one could solution is to install the Anaconda distribution of python).
- Your homework will be graded with unit tests. To run the unit tests, you should have pytest and pickle python packages installed (they come automatically with Anaconda).
- To run unit tests: put test_homework1.py and data.pck in the directory containing homework1.py. From the shell, run "python -m pytest -v" in the directory containing your homework1.py
- Questions 1 and 3 involve randomness. Every time you run your code, you will get a different answer. You should ensure your answers are accurate enough to consistently pass the unit tests.
- Do not look at anyone else's code.
- The only modules your code can import are: numpy, random

3. Correlated Sports Performance and Permutation tests

Question 1. *In many sports, athletes believe that their recent performances are correlated. For example if athlete X scores a goal in one game he may believe his is more likely to score a goal in the next game, and if he does not score, he is more likely to remain scoreless in the next game (i.e. he is in a slump). The goal of this homework is to design a statistical test to detect if there is a such a clustering effect in time.*

- *The input will be a player history, an array of ones and zeroes. For example,* `history=[1,1,0,0]` *means that the player made the first shot and the second shot but missed the third and fourth shots.*
- *Generally, people consider clustering in time to be the following: if a player made a shot, the player is more likely to make the next shot, and if a player misses a shot, the next show it also likely to miss. So we will use the following test statistic: the number of i such that* `history[i]==history[i-1]`*. For example if* `history=[1,1,0,0,0]` *then the value of the test statistic is 3.*
- *Using this test statistic, design a permutation test to test for clustering in time. In homework1.py, fill in the body of the function* `q1(history)`*, so that it returns the p-value from the test. You will need to determine the number of permutations to use.*

4. Decision Theory

Question 2. *The Pennsylvania Chocolate Institute of Health has identified a new type of disease caused by a diet containing insufficient amount of chocolate. Treatment requires expensive imported chocolates. Fortunately, they are covered by insurance. There are two types of patients:*

A. *Healthy chocolate eaters*
B. *Patients suffering from low chocolate levels*

The insurance company has developed a test (machine learning model) which predicts p, the probability that the current patient is of type A. The company is experimenting with a cost model:

- *The cost of treating a patient of type A is α*
- *The cost of treating a patient of type B is* 1
- *The cost of not treating a patient of type A is* 2

- *The cost of not treating a patient of type B is* 5.

You must write a program to help the insurance company explore the cost of making decisions using decision theory.

- *In homework1.py, fill in the function* `q2(p)`*, where p is the probability the patient is of type A. The output of this function should be the value of α for which the insurance company is indifferent between treating and not treating the patient.*

5. Model Selection and Luck

Question 3. *The Faculty Rugby League® is a sports league of N teams (each from a different university). Each team is equally bad at playing rugby so, for any pair of teams A and B, the probability that A beats B is $\frac{1}{2}$ (there are no ties). The league operates as follows:*

- *Each team plays 16 games, one per day*
- *Every day, teams are randomly paired off (i.e., they are placed in a random order and the first team plays the second, the third team plays the fourth, etc.).*
- *At the end of the season, the team with the most wins (ties are broken arbitrarily) is declared the champion.*

Your job is to understand how lucky a team can get in a league that has parity (by the way, this homework problem might be a metaphor for model selection). Design a simulation experiment to determine the expected number of wins a champion would have (i.e. the average number of wins in a large number of seasons).

- *In homework1.py, fill in the function* `q3(N)`*, where N is the number of teams in the league. The function must output the expected number of wins a champion team would have.*