<Online ad recommendation>
In this demo, you are given an environment of online ad recommendation with deterministic rewards, which means that there is a function that takes in an action and outputs one reward deterministically (the reward is not sampled from a probability distribution)

1. Read the following article:
   https://lilianweng.github.io/lil-log/2018/01/23/the-multi-armed-bandit-problem-and-its-solutions.html

2. Implement IPS:
   https://github.com/sanghoonkim0918/2020_summer_internship/blob/191641c88c6bf95a2467b40e9f9c103610f169fc/CB_demo/cb_demo.py#L18-L19

3. Implement a stochastic policy and a deterministic policy. Both of the policies should be stationary:
   https://github.com/sanghoonkim0918/2020_summer_internship/blob/191641c88c6bf95a2467b40e9f9c103610f169fc/CB_demo/cb_demo.py#L105-L125

4. Evaluation:

   get the trace with the following (old policy, new policy) pairs:
   1) Uniformly Random, Epsilon Greedy
   2) Uniformly Random, Stochastic policy
   3) Uniformly Random, Deterministic policy
   4) Epsilon Greedy, Stochastic policy
   5) Epsilon Greedy, Deterministic policy

   For each trace, you will plot a graph using matplotlib where x_axis is number of samples [10^1, 10^2, 10^3, …, 10^6] in logarithmic scale and y-axis is the mean error of IPS estimates.

   Given a trace and a number of samples, do the following:
   1) get the IPS estimate of trace[:num_sample] and store it

2) Use bootstrapping 19 (or just 9) times to see the variance of IPS estimates.

Bootstrapping is done as follows:

- Get a new trace by: new_trace = np.random.choice(trace, size=len(trace))
- Store the IPS estimate of the new trace
- Repeat this 19 times and get 19 more IPS estimates

3) Plot the mean error of the 20 (or 10) IPS estimates. Each error is defined as

$(true\_performance - IPS\_estimate)/true\_performance$

where $true\_performance$ is in https://github.com/sanghoonkim0918/2020_summer_internship/blob/39e9320880a28f97e0a4dfc53e5d586e4dce5f22/CB_demo/cb_demo.py#L173

4) Plot the standard deviation of the errors as well using np.std

You have to see that whether we are using a stationary or non-stationary policy as the old policy, the mean error decreases as the number of samples increases