Assignment 2

Name: Vijaykumar Maraviya
Student Number: 1006040320

1. **Explain clearly why V_π is not useful in the MC development above?**
   In GPI, iteratively, the value function is estimated with respect to current policy (Eval) and policy is being improved with respect to the value function (Impr). We achieve a fixed point when the value function is consistent with the current policy, and the policy is greedy with respect to its own value function. At which point, the bellman optimality equation holds, and the value function and policy are optimal.

   In dynamic programming policy improvement, we make policy greedy with respect to the current estimate of state-value function to improve the policy. For that, we used the following equation:

$$\pi(s) \leftarrow argmax_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$$

   As shown in the equation above, if we have a model of the environment, state values are sufficient: we can look ahead one step and choose an action that leads to the best combination of immediate reward and the next state. Without transition probability, the state-value function alone is not sufficient to select action because we do not know the probabilities of moving to the next states and hence can not evaluate the action. Instead, we use the action-value function and explicitly estimate the value of each action in order for values to be useful in improving a policy.

2. **The MC algorithm so far (ref: p 99), requires an infinite number of episodes for Eval to converge on Q_{π_k} (step k). We can modify this algorithm to the practical variant where Eval is truncated (c.f., DynProg GPI). In this case:**
   a. **Will we obtain Q_{π_k} (step k) from eval?**
      No, with a truncated policy evaluation, we do not converge to the true value function q_π (fixed-point k) for a given policy π.

   b. **If not, why are we able to truncate Eval? Explain clearly.**
      We are still able to use truncated evaluation because GPI (generalized policy iteration) still converges to optimum policy and value function, as explained below:

      GPI alternates between policy evaluation(E) and policy improvement(I), with the value function evaluated at an episode boundary rather than the true value function. Let assume that the GPI converges to sub-optimal policy π_c.

$$\pi_0 \xrightarrow{E\_trunc} Q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E\_trunc} Q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E\_trunc} \dots \xrightarrow{I} \pi_c \xrightarrow{E} q_{\pi_c}$$

      In this case, subsequent policy evaluation steps will improve the estimate of q_{π_c}. This is because the policy will remain fix to π_c (because of convergence) and agent with encounter more and more episodes using policy π_c. If we continue the policy

evaluation process, eventually, the estimate q_{π_c} converges to the true value. If we perform the policy improvement step with the true value function, we are guaranteed to get $\pi' \geq \pi_c$. If equality occurs, then π_c is a fixed point. And from bellman optimality equation, π_c is an optimal policy. If, however, $\pi' > \pi_c$, the process will continue with truncated evaluation until GPI eventually leads to the optimal policy, as just explained.

c. **Assuming ES (i.e., thorough sampling of the S x A space), and the above truncated Eval_trunc, is it possible to converge on a sub-optimal policy pi_c? Is this a stable fixed point of the GPI for MC? Explain clearly.**
In Monte Carlo ES, all the returns for each state-action pair are collected and averaged, regardless of the policy used to sample the episodes. It can be seen that the Monte Carlo ES would not converge to suboptimal policy. If it converges to suboptimal policy, then the value function would eventually converge to the true value function for that sub-optimal policy, as explained in the above question. The true value function will again lead to better policy through policy improvement. (policy improvement theorem). A fixed point in GPI is only achieved when both value function and policy are optimal. Hence, convergence to optimal policy seems inevitable; however, this has not been formally proven according to the textbook.

3. **Explain how you can synthesize a stochastic policy given what you know so far (you don't need to read ahead).**

To ensure that all the state action pairs are visited, we need to ensure $\pi(a|s) > 0$ for all actions in each state s. One way to ensure this is to assign probability greater than $\frac{\epsilon}{|A(s)|}$ to all actions in each state s. Here ϵ is a very small number greater than 0.

For example:

$$\boldsymbol{\pi(a \mid s)} = \begin{cases} \mathbf{1} - \boldsymbol{\epsilon} + \boldsymbol{\epsilon}/|\boldsymbol{A(s)}|, & \boldsymbol{if\ a} = \boldsymbol{A^*} = \boldsymbol{argmax_a q(s,a)} \\ \boldsymbol{\epsilon}/|\boldsymbol{A(s)}|, & \boldsymbol{if\ a} \in \mathbf{A(s)} - \boldsymbol{A^*} \end{cases}$$

This policy selects greedy action with probability $\mathbf{1} - \boldsymbol{\epsilon} + \boldsymbol{\epsilon}/|\boldsymbol{A(s)}|$ and chooses from all other actions uniformly with probability $\boldsymbol{\epsilon}/|\boldsymbol{A(s)}|$.

Output of code:

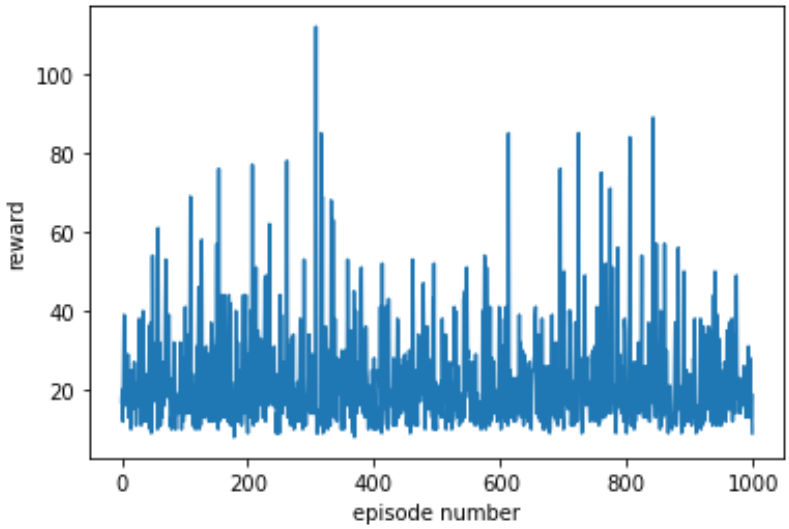
Cart-Pole problem:

A reward of +1 is provided for every timestep that the pole remains upright. Maximum timesteps, and hence the reward, allowed in a single episode of CartPole-v0 environment is 200.

To assess the policy learned using Monte-Carlo off-policy method, it is compared against the random policy.

Performance of random policy:

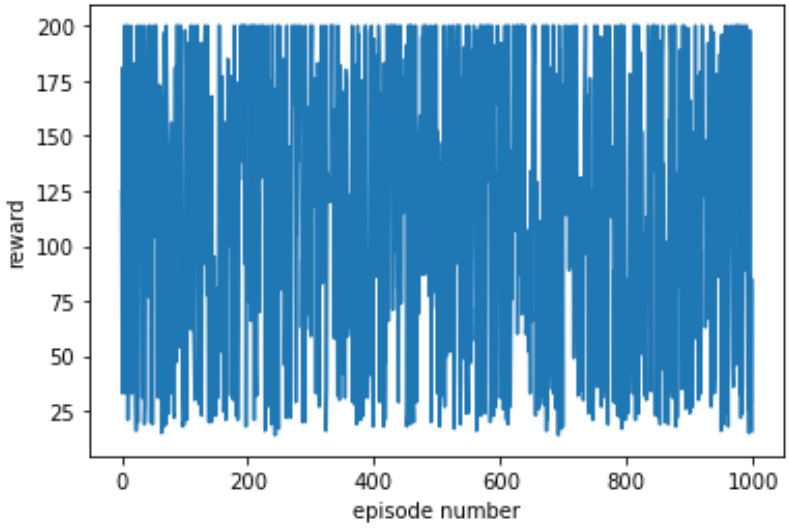
```
Average reward over 1000 episodes: 22.35
number of successes (reward >=200) in 1000 episodes: 0
```



The above figure shows the total reward received in each episode with random policy. The agent with random policy never achieved reward more than 120.

Performance of learned policy: The policy is learned using 100,000 episodes.

```
Average reward over 1000 episodes: 115.15
number of successes (reward >=200) in 1000 episodes: 180
```



The above figure shows the total reward received in each episode with learned policy. The agent frequently achieves the maximum possible reward.

Comparison:

The learned policy received 5 times more reward than random policy on average. The average reward for learned policy is 115, which means that on average, using the learned policy, agent can hold cartpole upright for 115 timesteps. Out of 1000 episodes during testing, the agent held cartpole upright for more than 200 steps (maximum possible) in 180 episodes.