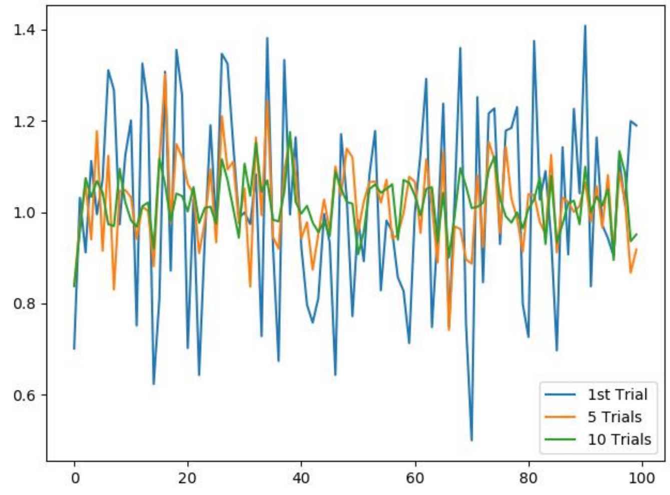
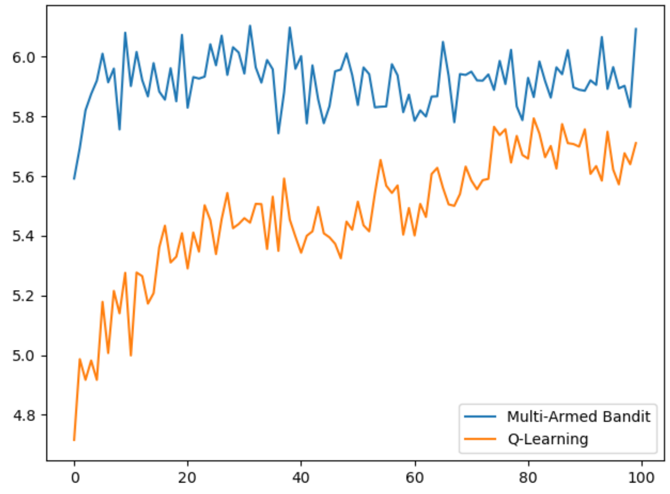
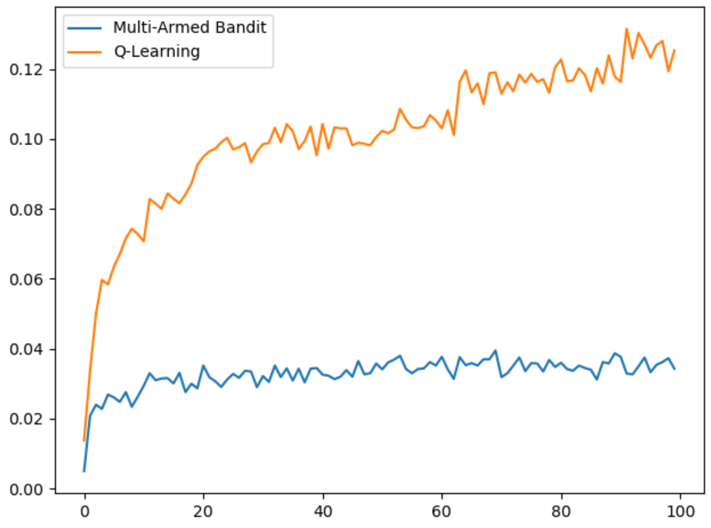
1. Tic-Tac-Toe
    a. States are the positions of the X's and O's on the board. Actions are either you or your opponent marking an X or an O on the board. The current state affects the actions you can take because you cannot make a mark where there is already a mark.
    b. The reward actions should be +1 for a win, 0 for a tie, and -1 for a loss. For actions that do not end the game, the reward should be given after your opponent plays to determine the quality of the action. The reward function can be modeled by the Q-Learning function Q(s, a)⟵Q(s, a) + a(r + gmaxa'Q(s', a') – Q(s, a)).
2. Bandits vs Q-Learning
    a. 

    b. 

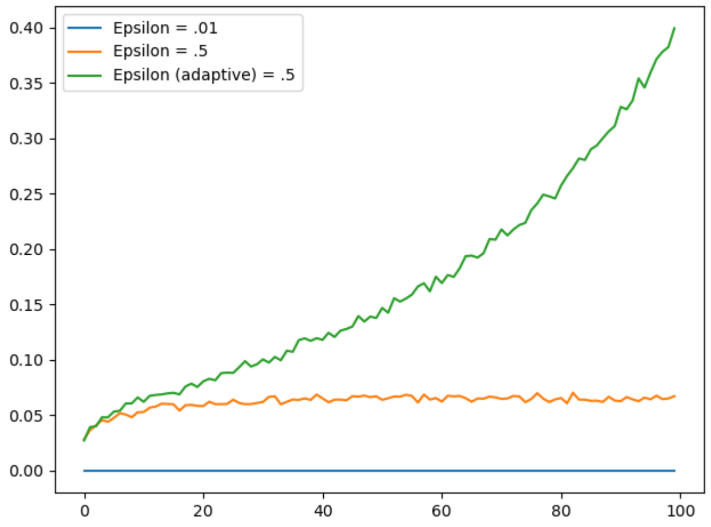
    c. It is important to average over many trials because it gives us a better understanding of the expected rewards. The jaggedness decreases significantly with more trials.
    d. The multi-armed bandit obtains higher reward than Q-Learning.



e.

   i. Q-learning is significantly better for Frozen Lake.
   ii. Multi-armed bandit struggled to learn because it assumes there is only one state and it assumes Q is identical for all states.

3. Exploration vs Exploitation



   a.
   b. The averaged rewards of the Q-Learning learner were maximized for epsilon of .01.
   c. See a.
   d. The adaptive epsilon returned the highest reward. This suggests the importance of exploration early, that then transitions to more exploitation.
   e. The epsilon value should reset when the holes and goal change in order to do more exploration. If n is known, epsilon's adaptiveness can be over n instead of over the progress.

4. On-Policy vs Off-Policy
   a. On-Policy algorithms are influenced by the exploration policy, meaning that it updates values based on the current policy. SARSA is an on-policy algorithm.

Off-policy algorithms are not influenced by the exploration policy, and values are updated based on greedy policy. Q-Learning is an off-policy algorithm.

5. MDP
    a. The robot will learn to deposit trash more often in order to pick up the larger reward. The reward function in the MDP can be edited to include the 10% chance of failure.
    b.
        i. The state-values are V(0) = 1, V(1) = 1. The state-action-values are Q(0, A) = 1, Q(0, B) = 1, Q(1, A) = 1, Q(1, B) = 1.
        ii. Any policy will be optimal as all paths lead to the goal.
        iii. The discount factor decreases the reward if you take extra steps. The new state-values are V(0) = 1, V(1) = .95. The state-action-values are Q(0, A) = 1, Q(1, A) = Q(1, B) = .95.