First visit MC:

If each state is visited infinitely often, then \( V_k \rightarrow v^* \).

If each state-action pair is visited infinitely often, then \( Q_k \rightarrow Q^* \).

Doing this: MC Estimation of Action-Values:

\( Q_n (s, a) \) - average return from the first time action \( a \) taken in state \( s \) in each episode.

Problem: What if it never chooses \( a \) in state \( s \)?

Is \( Q_n (s, a) \) still defined? Yes

\( Q_n (s, a) \) will not be estimated.

What to do?

Solution 1: Exploring starts:

Randomize \( s_0 \) and \( a_0 \) s.t. every (\( s, a \)) pair has a non-zero probability.

Not possible for all systems!

Solution 2: Stochastic Policy \( \pi \)

Non-zero probability for every action in each state.
Monte Carlo Control with Exploring Starts

- Avoid generating episodes in the evaluation step by improving after a single episode.
- Accumulate returns over all episodes.

Pseudocode:

**Init:** For all s \( \in \mathcal{S} \), a \( \in \mathcal{A} \), q \_i(s,a) \leftarrow \text{arbitrary value}

\[ \pi_i(s) \leftarrow \text{"action"} \]

Returns \((s,a)\) \leftarrow \text{empty list}

Repeat forever

1. Generate an episode using exploring starts using \( \pi_i \)
2. For each \((s,a)\) in the episode,
   \( G \leftarrow \text{return following first occurrence of } (s,a) \text{ in the episode} \)
   Append \( G \) to \( \text{Returns } (s,a) \)
   \( q(s,a) \leftarrow \text{mean } (\text{Returns } (s,a)) \)
3. For each \( s \) in the episode, set
   \( \pi'_i(s) \leftarrow \text{argmax } \_i a q(s,a) \)

If this converges, it does so to an optimal \( \pi'_i \).

**Proof:** Suppose it converges to a suboptimal \( \pi'_i \). Then \( q \) has converged to the actual values for \( \pi'_i \). Then policy improvement must move \( \pi'_i \) to a better policy.

**Note:** It can fail to converge.
Monte Carlo Control with Stochastic Policies:

**E-greedy MC Control:**

Init: $q(s,a) \leftarrow$ arb. values 
$\pi(s,a) \leftarrow$ arb. s.t. $\pi(s,a)$
Returns $(s,a) \leftarrow$ empty

Repeat Forever:

1. Generate an episode using $\pi$
2. For each $(s,a)$ in the episode:
   - $G \leftarrow$ Return following first occurrence of $(s,a)$ in the episode
   - Append $G$ to Returns $(s,a)$
   - $q(s,a) \leftarrow$ mean (Returns $(s,a)$)
3. For each $s$ in the episode
   - $a^* \in \text{arg max}_a q(s,a)$
   - $\pi(s,a) \leftarrow \begin{cases} 1 - e^{-\frac{e}{|a'|}} & \text{if } a = a^* \\ \frac{e}{|a'|} & \text{if } a \neq a^* \end{cases}$

**Properties:**

- By variations on the Policy Improvement from Sutton & Barto, Vol. 1, Sect. 5.4, converges to an $\epsilon$-greedy policy.
- In fact to the optimal $\epsilon$-greedy policy (not necessarily optimal among all policies).
**TD (Temporal Difference) Learning:**

**Properties:**
- Like MC, learns directly from experiences:
  - does not need $P$ and $R$.
- Like DP (Dynamic Programming): updates estimates in terms of other estimates.

But first, another MC algorithm:
Consider updates: $f(x) \leftarrow f(x) + \alpha (Y - f(x))$  

- weights: $w$  
- gradient descent  
  - estimate from one sample: stochastic gradient descent  
  - $w \leftarrow w + \alpha (Y - f_w(x)) \frac{\partial f_w(x)}{\partial w}$  
  
**Bellman Equation:**
$$v(s) = E[R_t + \gamma v(s_{t+1}) | s_t = s]$$

**TD Update:**
$$v(s_t) \leftarrow v(s_t) + \alpha (R_t + \gamma v(s_{t+1}) - v(s_t))$$

**The TD Error:**
$$v(s) \leftarrow v(s) + \alpha S_t$$

"Renard Prediction Error"
MC is a stochastic gradient descent.

Is TD stochastic gradient descent?

If it is: \( f(v) = E \left[ \frac{1}{2} (R_t + \gamma v(S_{t+1}) - v(S_t))^2 \right] \)

\[ = E \left[ \frac{1}{2} \Delta t \Delta S_t^2 \right] \]

\( \nabla f(v) = \Delta t \cdot \frac{\partial \Delta S_t}{\partial v} \) (ignoring expectation)

\[ = (R_t + \gamma v(S_{t+1}) - v(S_t)) \left( \gamma \frac{\partial v(S_{t+1})}{\partial v} - \frac{\partial v(S_t)}{\partial v} \right) \]

\[ \begin{bmatrix} 0 \\ \frac{\partial}{\partial v} \end{bmatrix} \]

\[ S_{t+1} \quad \begin{bmatrix} 0 \\ \frac{\partial}{\partial v} \end{bmatrix} \quad \Delta S_t \]

\( v(S_t) \leftarrow v(S_t) + \alpha \Delta S_t \)

\( v(S_{t+1}) \leftarrow v(S_{t+1}) - \alpha \gamma \Delta S_t \)

Some properties:
- TD offers an update at each step.
- Converges a.s. to correct \( v \).

Seems close to s.g.d., but it is not properly sampled. (Will cover later in the course.)