**Reinforcement Learning Assignment – 1**

**Analysis of Single-Armed and Multi-Armed Bandit Problems**

**Introduction**

This report investigates the convergence behavior of sample statistics in the SingleArmedBandit notebook and the decision-making process in the MultiArmedBandit notebook. The SingleArmedBandit experiment demonstrates how sample means and variances converge to their population counterparts under varying conditions, while the MultiArmedBandit notebook illustrates reinforcement learning strategies for balancing exploration and exploitation.

**Single-Armed Bandit problem**

* **Objective**

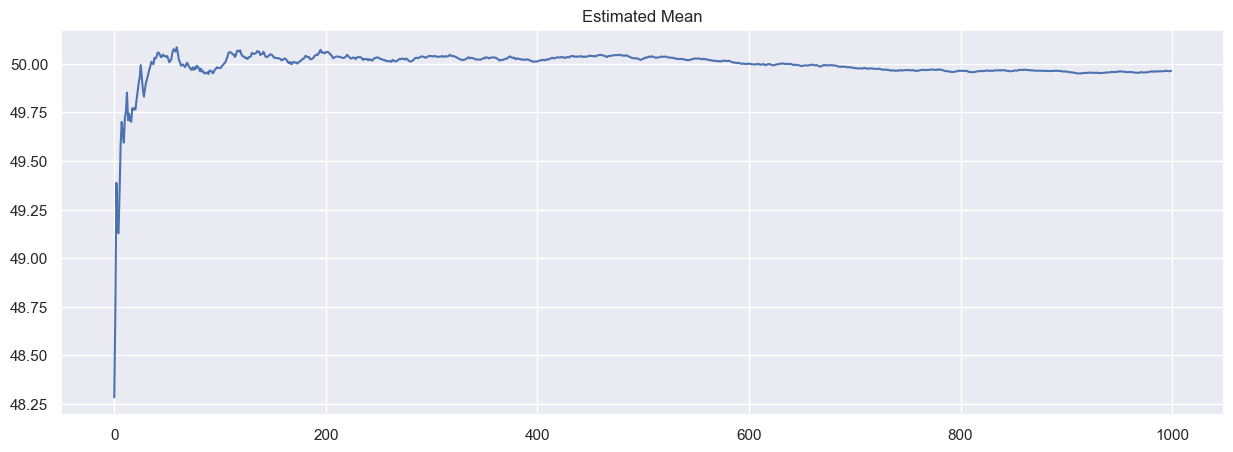
The SingleArmedBandit notebook demonstrates how quickly the sample mean converges to the population mean and how quickly the sample variance converges to the population variance. This is achieved by simulating data from normal distributions with specified population mean (*μ*) and standard deviation (*σ*) values.

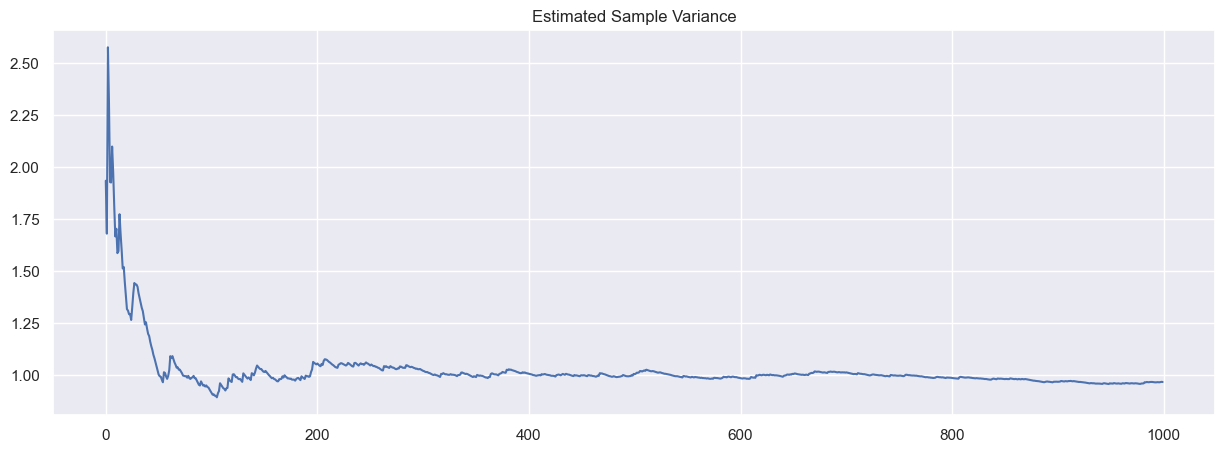
* **Approach**

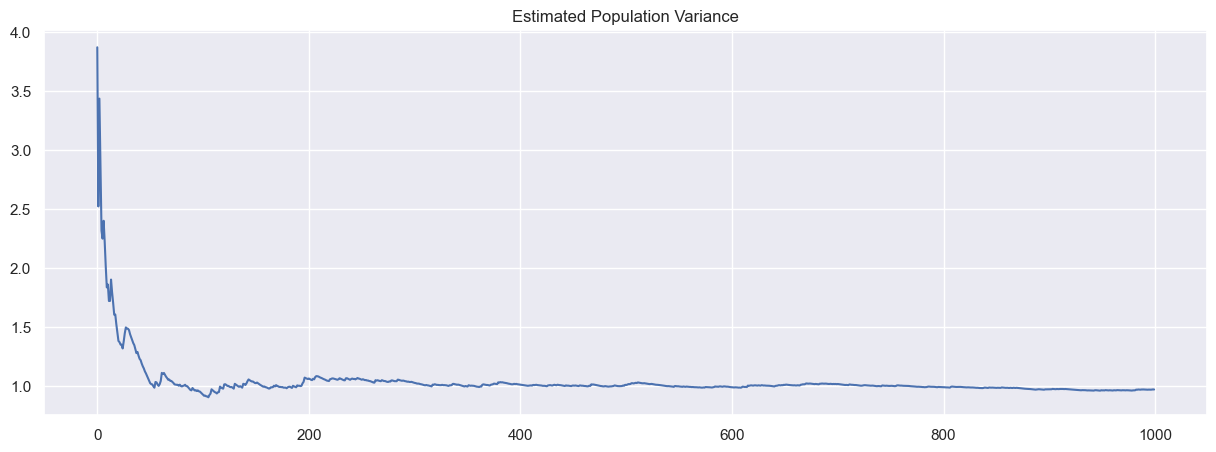
1. **Initial Experiment:** Population mean (*μ*) and standard deviation (*σ*) were varied to observe their impact on convergence.

Let us consider, few experiments with varying mean and standard deviations

* + mean (*μ*) = 50, standard deviation (*σ*) = 1

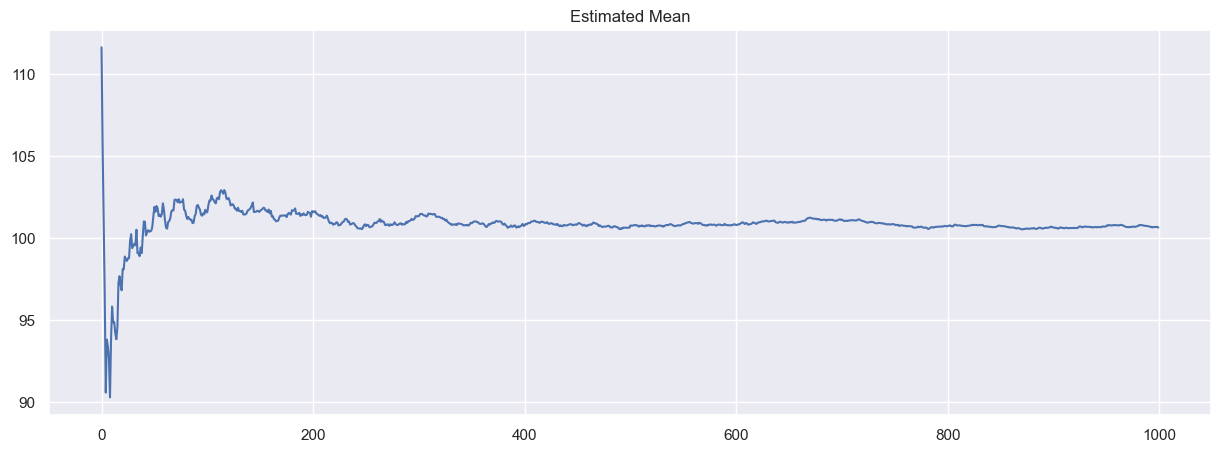


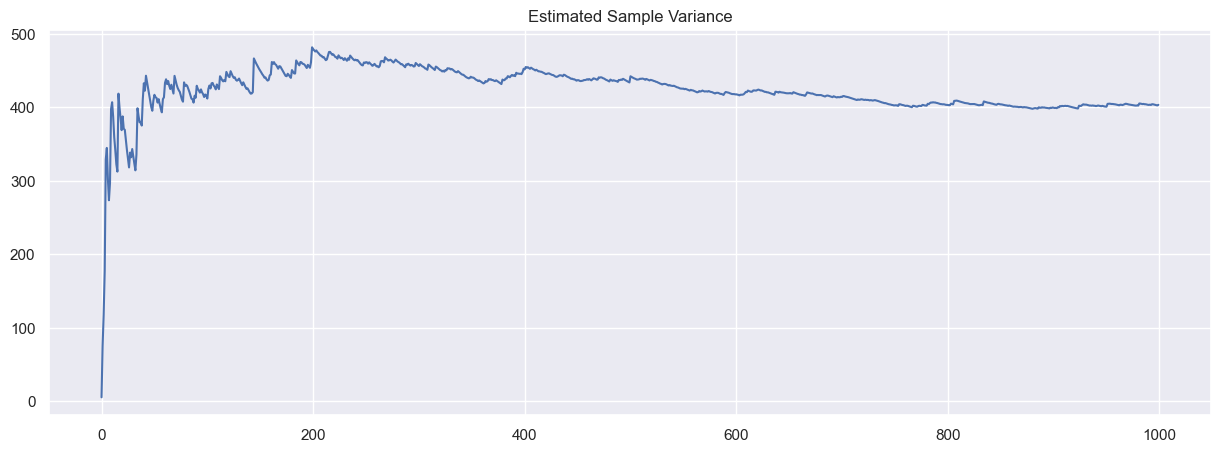


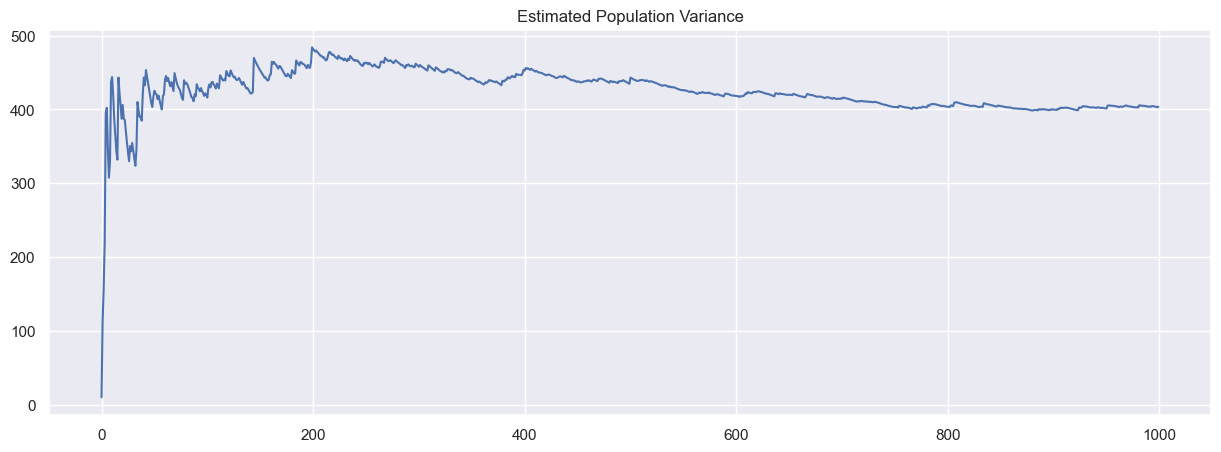


Observations:

* Sample mean converges rapidly to μ=50within ~300 iterations.
* Sample variance stabilizes at σ2=1 within ~100 iterations.
  + mean (*μ*) = 100, standard deviation (*σ*) = 20







Observations:

* Sample mean converges slowly, requiring >1000 iterations.
* Sample variance stabilizes at σ2=400 after ~800 iterations.

We just changed the values of means and standard deviations and observed the plots and see their convergence.

1. **Fixed Mean, Varying Standard Deviations:** The mean was fixed (*μ*=100), and standard deviations (σ=1,10,20) were tested.

Modify this file to analyze convergence for a fixed mean and varying standard deviations.

**Modifications in code:**

Run\_experiment 1: we have modified the code so that it iterated over the set of standard deviations provided using for loop. Generated separate plots for each standard deviation to visualize convergence

* Single\_armed bandit: we have set the value of mean to 100 and provided a list of SD as input. For each standard deviation value, generate random data points. Plots were added to visualize the convergence of the sample mean and variance for each standard deviation.

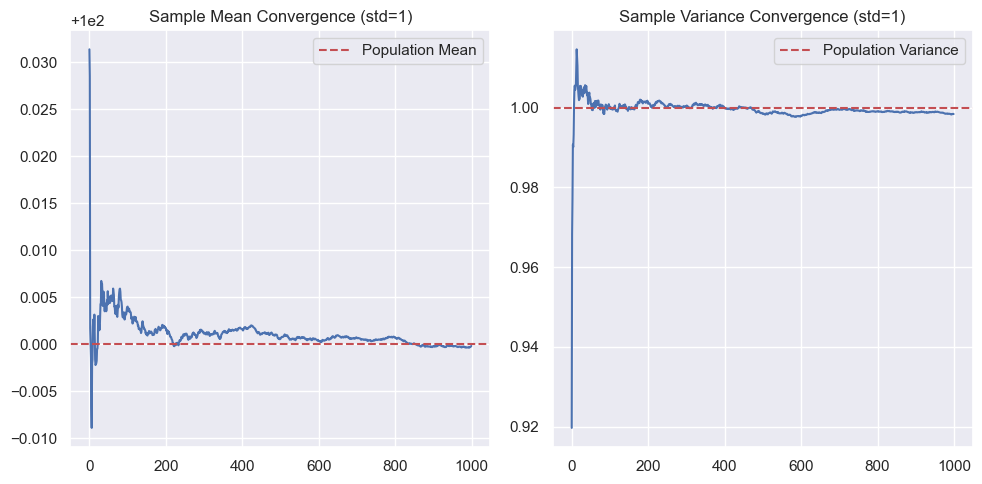
**Convergence analysis:**

* Plots were generated to visualize how quickly the sample mean and variance converge to their population values.
* The number of iterations required for convergence was estimated by eyeballing the graphs.

**Results and Interpretation**

**Plot 1:** *μ* = 100, *σ=1*

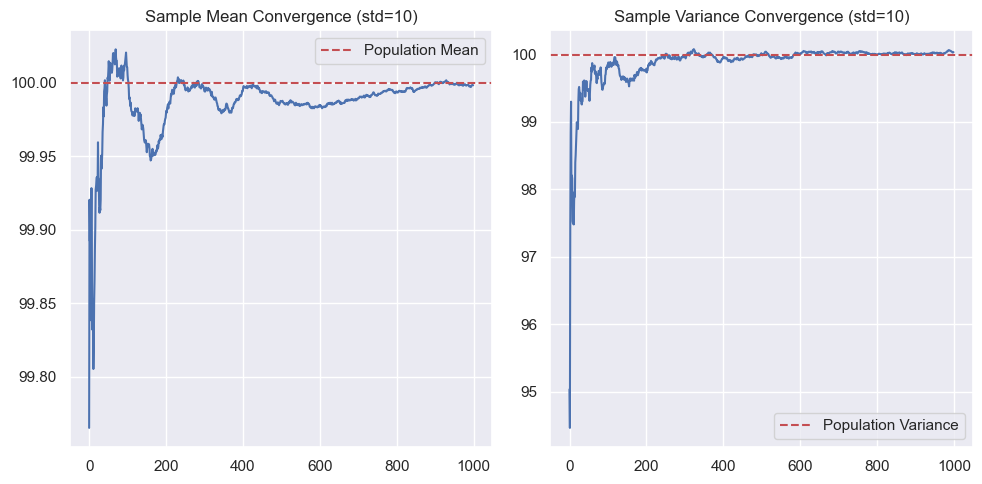
* **Sample Mean Convergence**:



* + The sample mean converges rapidly to the population mean (μ=100) within approximately 800 iterations.
  + Early fluctuations are minimal due to low variability in the data.
* **Sample Variance Convergence**:
  + The sample variance converges to the population variance (σ=1) within 80 iterations.

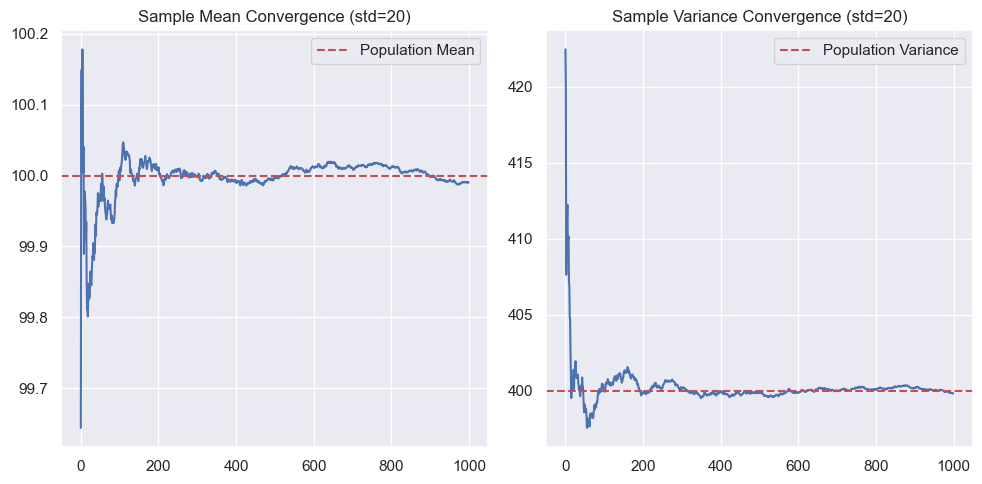
**Plots 2:***μ* = 100, *σ=10:*

**Sample Mean Convergence**:



* + The sample mean converges to the population mean (*μ*=100) more slowly, requiring approximately 900 iterations for stabilization.
  + Larger initial fluctuations are observed due to increased variability in the data.
* **Sample Variance Convergence**:
  + The sample variance converges to the population variance (σ2=100, *σ=10*) after approximately 300 iterations.

**Plots 3:** *μ* = 100, *σ=20*



**Sample Mean Convergence**:

* + Sample mean convergence is slower, i.e., > 1000 iterations.
  + Sample variance convergence was obtaimed to higher variability (σ2=400) i.e., around 700.

**Observation:**

* "Learning" refers to achieving stability close to population values. This is consistent with eyeballing the graphs.
* For *σ*=1,10, and 20, the number of iterations required increases as variability increases.

Thus, asking "how many iterations were required" is effectively the same as analyzing convergence behavior visually.

* Smaller standard deviations (σ=1) lead to faster convergence of both sample means and variances due to reduced variability in data.
* Larger standard deviations (*σ*=10,20) delay convergence due to increased fluctuations in early iterations.

Future Improvements we can implement automated methods (e.g., statistical thresholds) to determine convergence instead of relying on visual inspection.

This analysis highlights how variability impacts statistical estimation and provides insights into designing experiments with controlled parameters.

**Multi-Armed Bandit Problem**

The MultiArmedBandit notebook portrays a learning activity focused on decision-making under uncertainty, specifically using an epsilon-greedy policy to maximize rewards. Here's a detailed explanation of the technique and activity.

The objective is to devise a strategy (policy) that maximizes cumulative rewards over a series of pulls from multiple bandits (arms), each with an unknown reward distribution.

**Learning Activity Description**

The multi-armed bandit code uses epsilon-greedy policy

**Epsilon-Greedy Policy**

The epsilon-greedy policy is a simple yet effective strategy for balancing exploration and exploitation:

* **Exploitation**: With probability 1−*ϵ*, choose the arm with the highest estimated mean reward.
* **Exploration**: With probability *ϵ*, choose an arm randomly.

This policy ensures that the agent learns about all arms while favoring those with higher rewards.

**Approach**

1. **Initialization**:
   * Create multiple bandit objects with different mean rewards.
   * Initialize variables to track the number of pulls and estimated mean rewards for each bandit.
2. **Iteration Loop**:
   * At each iteration, generate a random number between 0 and 1.
   * If the number is less than ϵ*ϵ*, explore by choosing a random bandit.
   * Otherwise, exploit by choosing the bandit with the highest estimated mean reward.
3. **Update Statistics**:
   * After each pull, update the estimated mean reward for the chosen bandit.
   * Keep track of cumulative rewards.
4. **Visualization**:
   * Plot the cumulative rewards or average rewards over iterations to evaluate the policy's effectiveness.

**Details of activity in given code:**

* **Number of Bandits**: Three bandits with mean rewards of 10, 5, and 1, respectively.
* **Standard Deviation**: All bandits have a standard deviation of 1.
* **Epsilon Value**: *ϵ*=0.25, meaning 25% of the time is spent exploring.
* **Number of Iterations**: The experiment runs for 10,000 iterations.

The notebook tracks and prints the number of times each bandit is pulled, their estimated mean rewards, and the total cumulative reward.

**Key Observations**

* Over time, the agent learns to favor arms with higher rewards.
* The epsilon-greedy policy effectively balances exploration and exploitation.
* Reward accumulation increases as the agent refines its strategy based on observed outcomes.

**Conclusion**

1. In the SingleArmedBandit notebook:
   * Sample means converge faster than variances.
   * Higher variability (σ2) slows convergence.
2. In the MultiArmedBandit notebook:
   * The epsilon-greedy policy successfully balances exploration and exploitation.
   * Over time, agents learn optimal strategies for maximizing rewards.

**References:**1. Samishawl. (2023).Epsilon-Greedy Algorithm in Reinforcement Learning.

<https://www.geeksforgeeks.org/epsilon-greedy-algorithm-in-reinforcement-learning/>

2. [Volodymyr Kuleshov](https://arxiv.org/search/cs?searchtype=author&query=Kuleshov,+V), [Doina Precup](https://arxiv.org/search/cs?searchtype=author&query=Precup,+D). (2014). Cornell University. Algorithms for multi-armed bandit problems.  
<https://arxiv.org/abs/>  
3.

[James LeDoux. (2020). Multi-Armed Bandits in Python: Epsilon Greedy, UCB1, Bayesian UCB, and More.  
https://jamesrledoux.com/algorithms/bandit-algorithms-epsilon-ucb-exp-python/   
4. McGill University of Computer Science. (2013). Lecture 15: Bandit problems. Markov Processes  
https://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-lecture15.pdf.   
5. Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press. ISBN: 978-0262039246. Available online at http://incompleteideas.net/book/the-book.html](James LeDoux. (2020). Multi-Armed Bandits in Python: Epsilon Greedy, UCB1, Bayesian UCB, and More.https://jamesrledoux.com/algorithms/bandit-algorithms-epsilon-ucb-exp-python/ 4. McGill University of Computer Science. (2013). Lecture 15: Bandit problems. Markov Processeshttps://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-lecture15.pdf. 5. Sutton, R. S., Barto, A. G. (2018). Reinforcement Learning: An Introduction. The MIT Press. ISBN: 978-0262039246.  Available online at http://incompleteideas.net/book/the-book.html   )