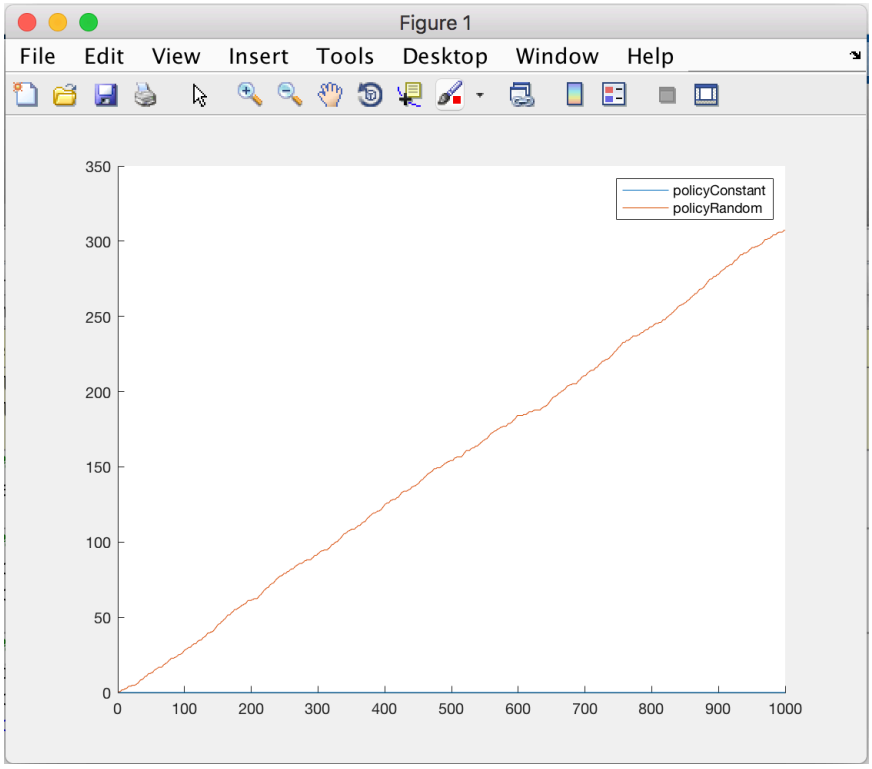
RoboStats Assignment 3
lte

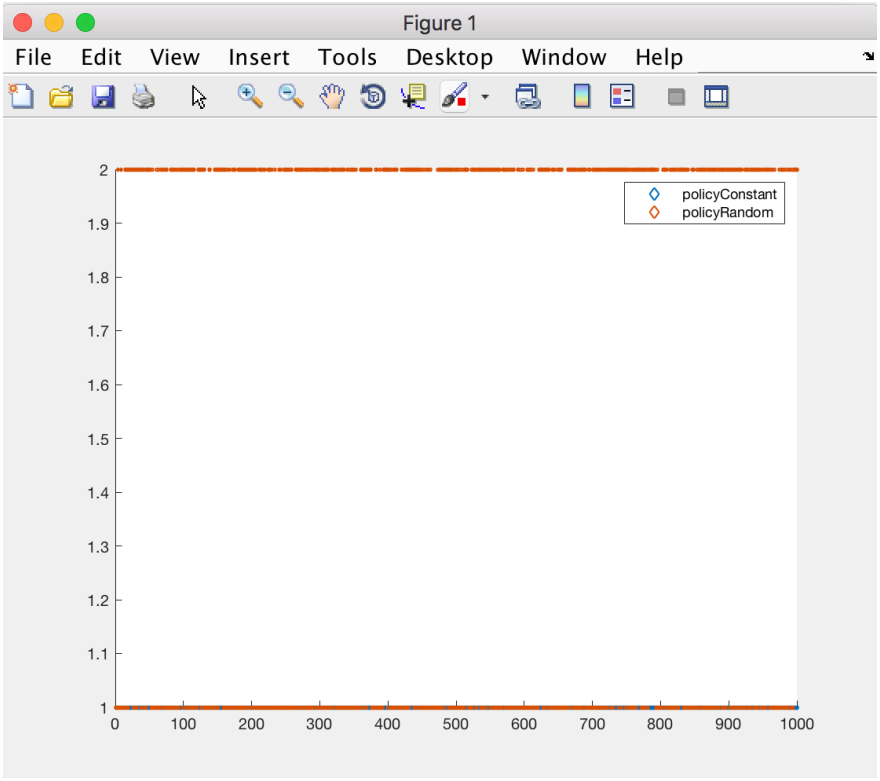
Question 2.5

Answer:

Regret:



Actions:



Does regret rise over time for policyRandom? What if you try multiple random restarts? Do you always get the same results? Why?
Answer:

Yes it does. I have done 10 different random restarts, and for all of them regret rises over time. When zooming into the jaggedness of the regret, you can observe slight differences here and there, but overall, the curves look almost the same, and all yield pretty much the same results.

The reason for this similarly is because in the random policy, you are equally likely within each round of carrying out the wrong or the right action. As a result, you will never really deviate too much from what would be the perfect case of one wrong, one right, one wrong, one right.

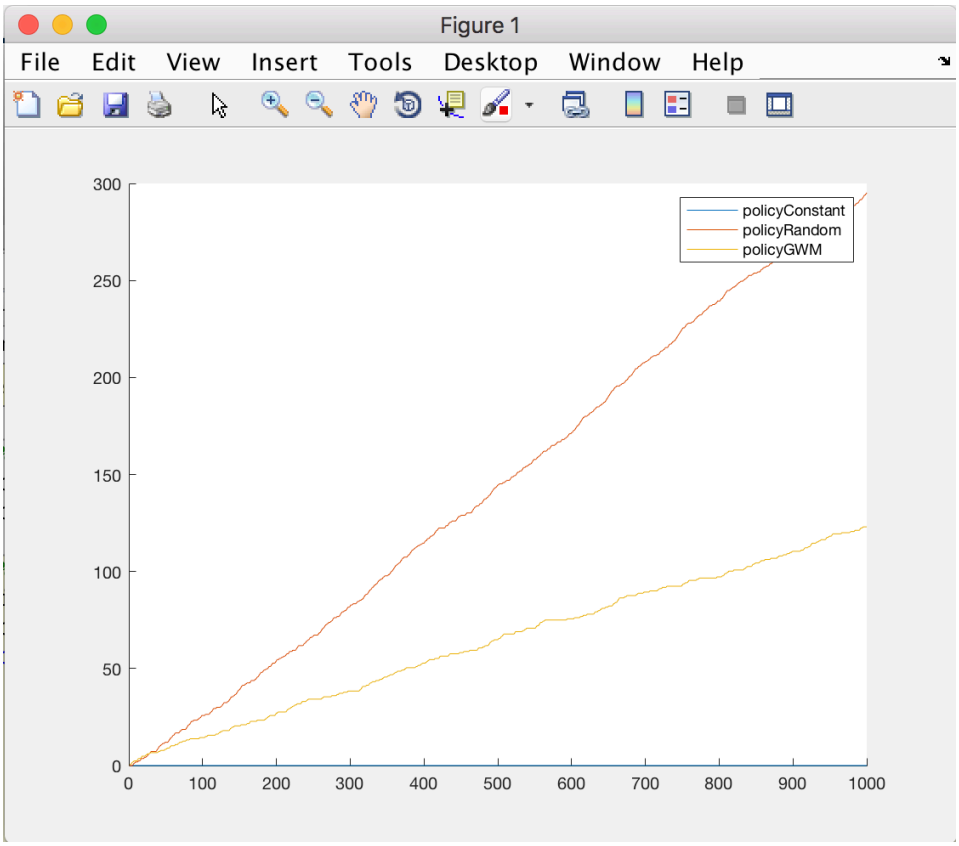
Does regret rise over time for policyConstant? What if you try multiple random restarts? Do you always get the same results? Why?
Answer:

Here the algorithm sticks with whichever action it picked first. Therefore if this action is the right one from the get go, it will have a much different regret profile than if it started off and stuck with the wrong option.

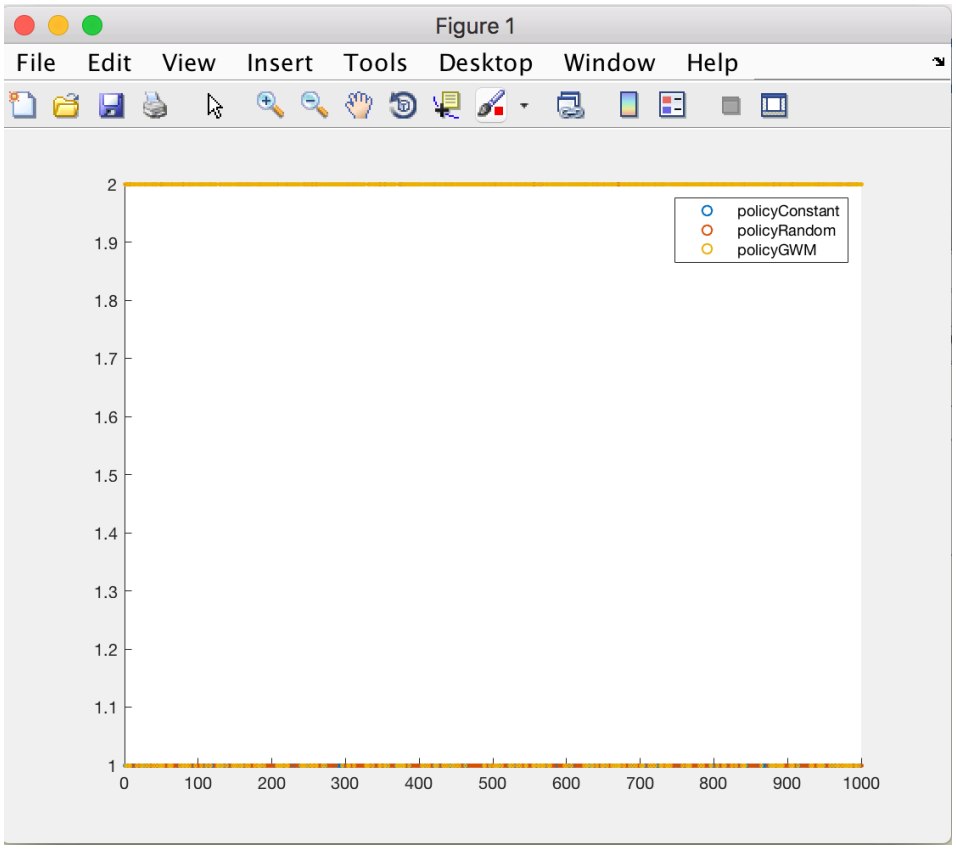
So in this case, it all depends, some trials show no regret because it picks the best hypothesis and sticks with it, whereas other trials show that if picks the wrong option, and because it systematically stays with it, it has no way of escaping out of this systemic regret from focusing on the wrong one, which a policy like policyRandom has.

3.1

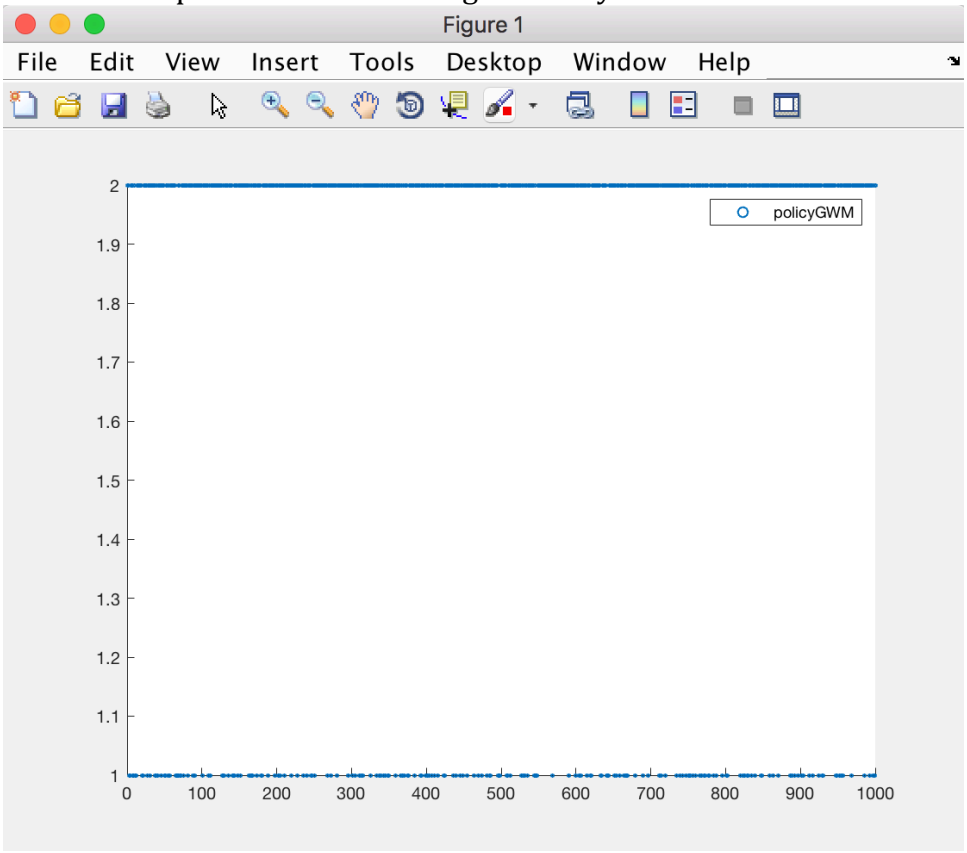
Regret:



Actions:



Here is the action profile of the EXP3 algorithm by itself:



Did the GWM algorithm work well here?
Answer:
Given the nature of this game (constant), it worked well.

Does it score better than the random policy?
Answer:
The regret profile is ramps up less aggressively than the random policy, which shows that this algorithm is less regretful of its actions, and therefore worked better than random. It didn't work better than the constant policy, which was lucky enough to stick with the right hypothesis from the start.

Is the GWM algorithm no regret? If so, under what assumptions?
Answer:
The thing about the GWM algorithm is that the feedback signal is almost like the front-end display of something, instead of information rich content of the back-end.

The GWM algorithm benchmarks how well it is doing by looking at hypothesis in absolute terms, it doesn't compare them relative to each other, and so regrets taking advice from very good hypothesis (compared to the rest of the group) in a way that is a bit too punishing.

Because of this it has regret, but if you had a way to show how good each hypothesis out of all possible hypothesis it could follow the best advice out of all of them, and in that case would become a no-regret algorithm.

3.1.2
Not attempted

3.1.3
Starting with equation 6 of the notes:

$$\tilde{l}^t{}_n = \frac{l^t{}_n}{p^t{}_n} l^t{}_a = n$$

We know that for all t we obtain:

$$\sum_{t=1}^{T} \tilde{l}^t{}_n = \sum_{t=1}^{T} \frac{l^t{}_n}{p^t{}_n} l^t{}_a = n$$

We know that $l^t{}_a = 1$ because there is only one action at that time-step. And when we take the expectation of this expression we get:

$$E\left[\sum_{t=1}^{T} \tilde{l}^t{}_n\right] = E\left[\sum_{t=1}^{T} \frac{l^t{}_n}{p^t{}_n}(1)\right] = E[n]$$

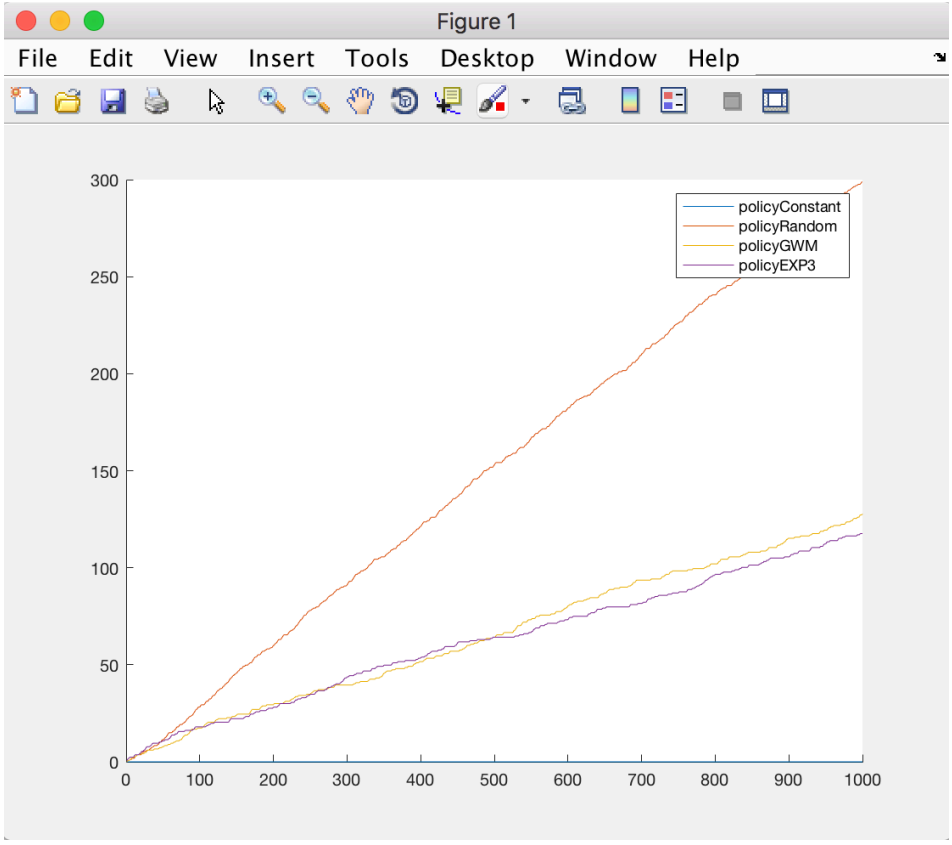
Where

$$E\left[\sum_{t=1}^{T} \frac{l^t{}_n}{p^t{}_n} l^t{}_a\right] = p^t{}_n \sum_{t=1}^{T} \frac{l^t{}_n}{p^t{}_n}$$

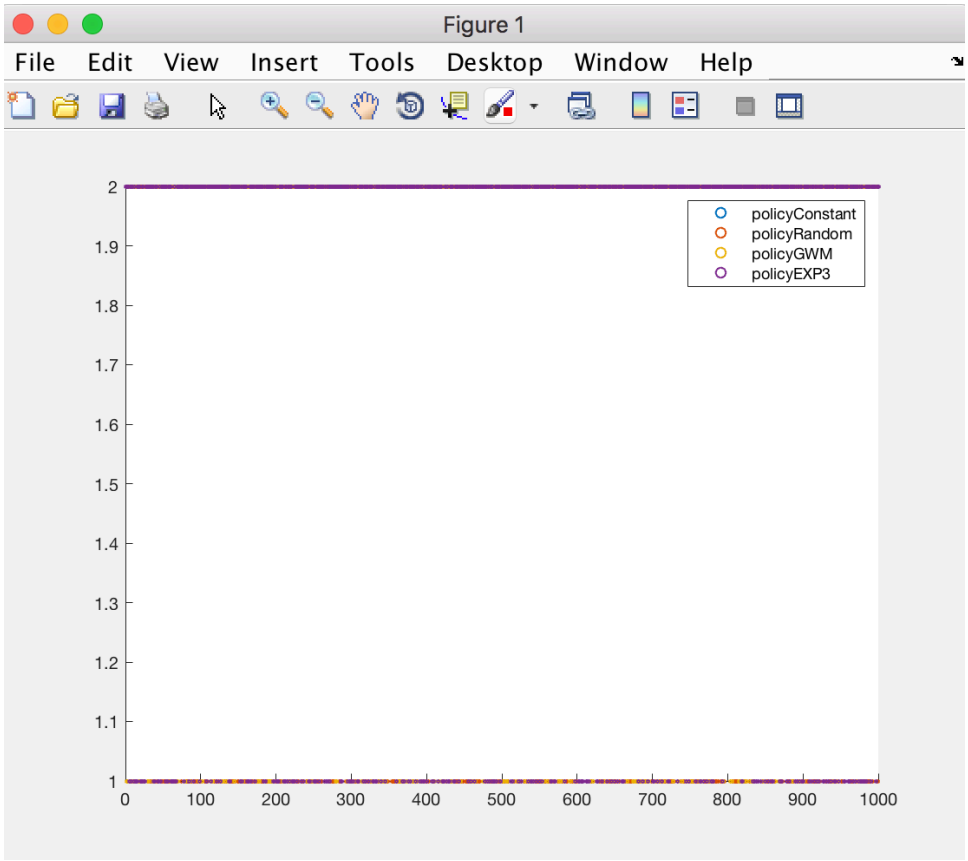
$$\therefore E\left[\sum_{t=1}^{T} \tilde{l}^t{}_n\right] = \sum_{t=1}^{T} l^t{}_n$$

3.2.1

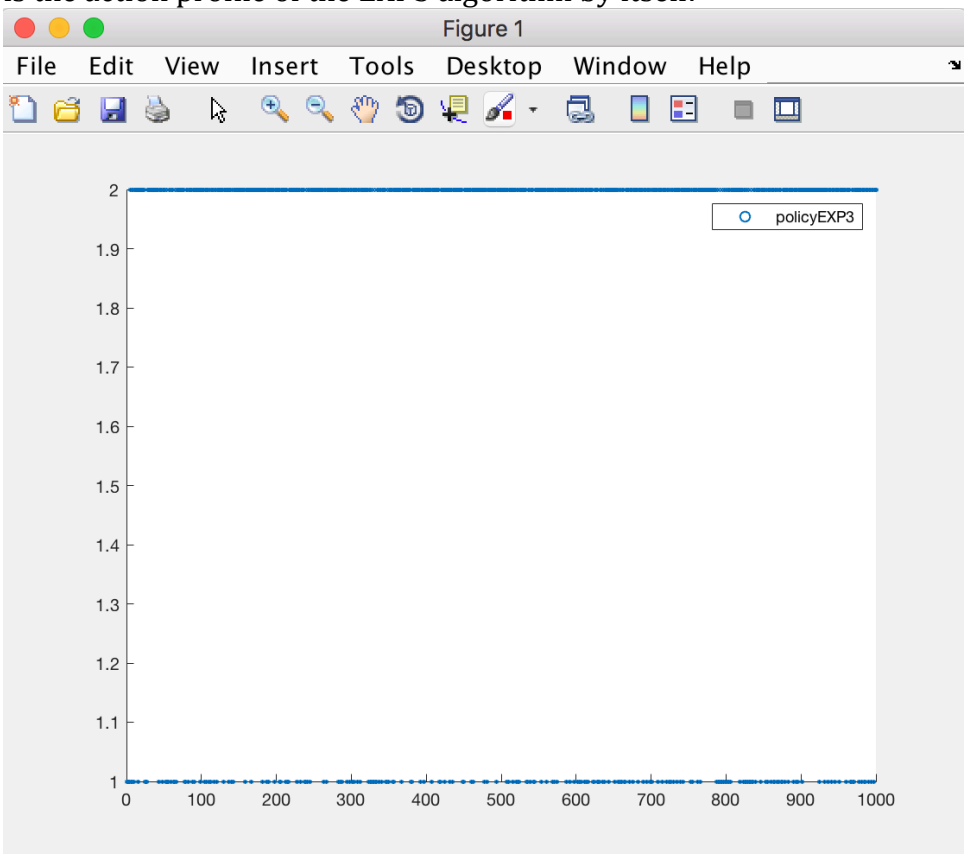
Regret



Actions:



Here is the action profile of the EXP3 algorithm by itself:



How does EXP3 behave near the beginning of training?

Answer:

It ascends at the same rate as GWM, both of which ascend slower in regret than the randomPolicy methods.

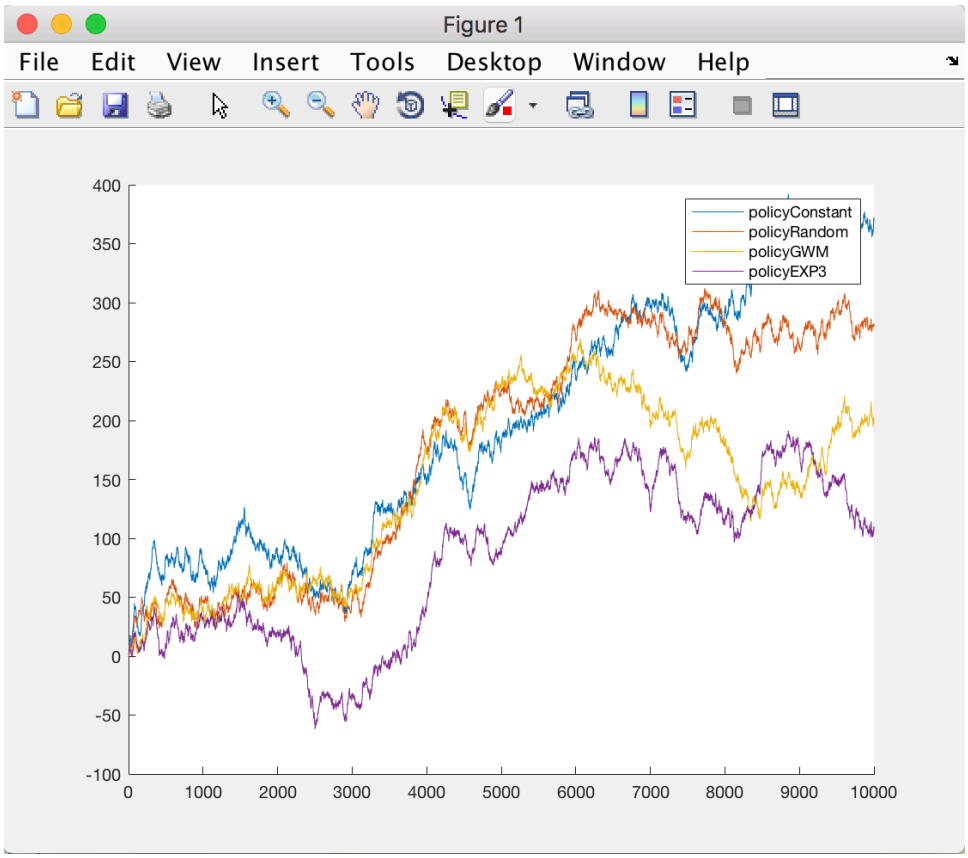
How does EXP3 behave near the end of training?

Answer:

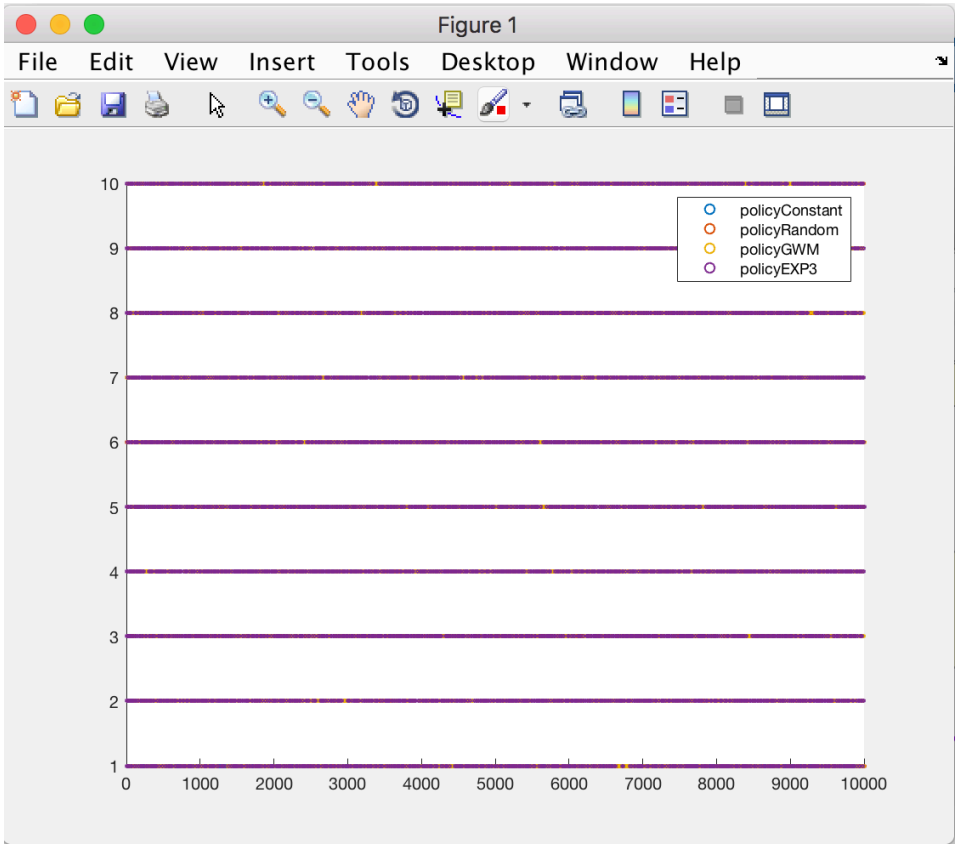
It slowly diverges away from the regret performance of GWM, where as the program continues, the regret of the EXP3 algorithm drops more and more compared to it. Some trials show this more so than others.

3.3.1

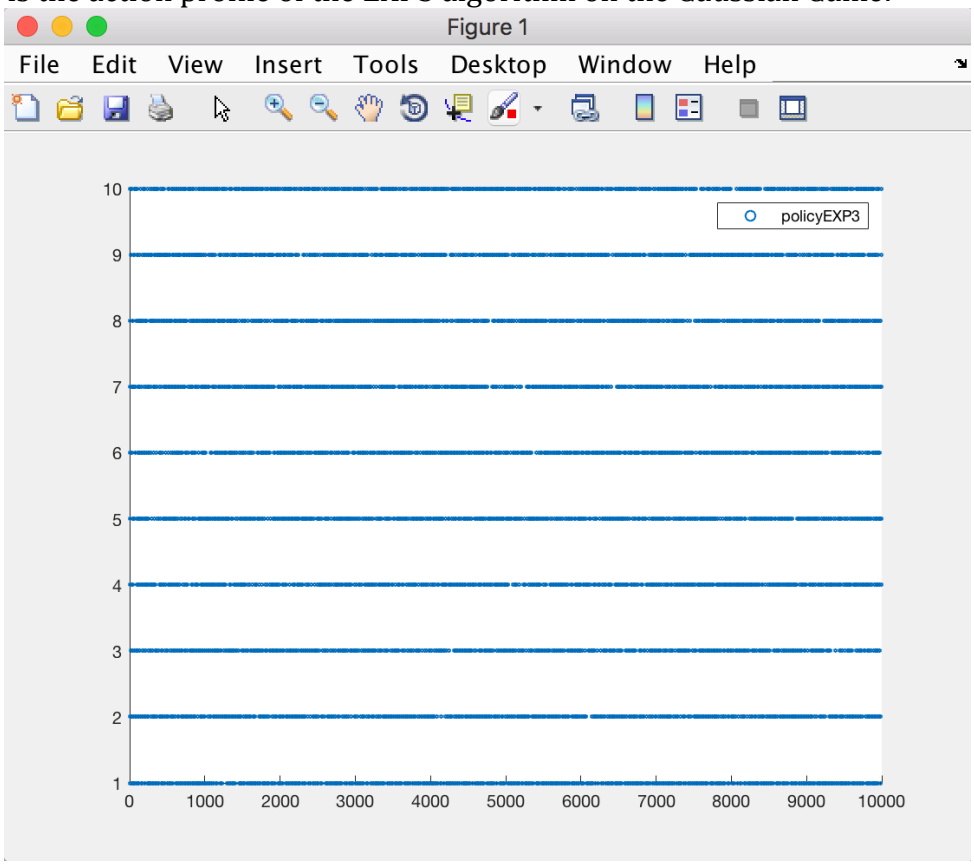
Regret:



Actions:



Here is the action profile of the EXP3 algorithm on the Gaussian Game:



4.2.1

We start off by remembering that:

$$1 - e^{-2m\epsilon^2} \leq P(\mu - \hat{\mu} \geq \epsilon) \leq e^{-2m\epsilon^2}$$

And that

$$P(\mu \leq \hat{\mu} + \epsilon) \geq 1 - e^{-2m\epsilon^2}$$

Now,

$$1 - \delta = 1 - e^{-2m\epsilon^2}$$

So

$$\delta = e^{-2m\epsilon^2}$$

Finding the logs of both sides, we get:

$$\log(\delta) = \log\left(e^{-2m\epsilon^2}\right)$$

$$\log(\delta) = -2m\epsilon^2$$

Referring back to

$$P(\mu \leq \hat{\mu} + \epsilon) \geq 1 - e^{-2m\epsilon^2}$$

This $\mu \leq \hat{\mu} + \epsilon$ calls for isolating the ϵ to one side, and getting rid of a negative by inversing the δ we get:

$$\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}} = \epsilon$$

Plugging this back into $\mu \leq \hat{\mu} + \epsilon$ we get:

$$\mu \leq \hat{\mu} + \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}$$

Which is:

$$\mu \leq \frac{1}{m}\sum_{m=1}^{m} X_i + \sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{2m}}$$

4.2.2
Show that the upper bound of the mean reward for an action n is the following:

$$\mu_n^t \leq \hat{\mu}_n^t + \sqrt{\frac{\log t}{2C_n^t}}$$

Answer:
Start off with the expression from the previous example:

$$\mu \leq \frac{1}{m}\sum_{i=1}^{m} X_i + \sqrt{\frac{\log(\delta^{-1})}{2m}} \qquad (1)$$

In this expression, when dealing with upper bounds the confidence δ needs to be inversely proportional to time. As time goes by, it should get worse. And so we can plug in the following (also from the previous question) $\delta = \frac{1}{t}$ in equation (1).

Plugging it in and we get:

$$\mu^t = \frac{1}{m}\sum_{i=1}^{m} X_i^{\ t} + \sqrt{\frac{\log\left(\frac{1}{t}^{-1}\right)}{2C_n^t}}$$

Where C_n^t is the number of times that the action in question was taken.
Simplifying we get

$$\therefore \mu^t = \frac{1}{m}\sum_{i=1}^{m} X_i^{\ t} + \sqrt{\frac{\log(t)}{2C_n^t}}$$

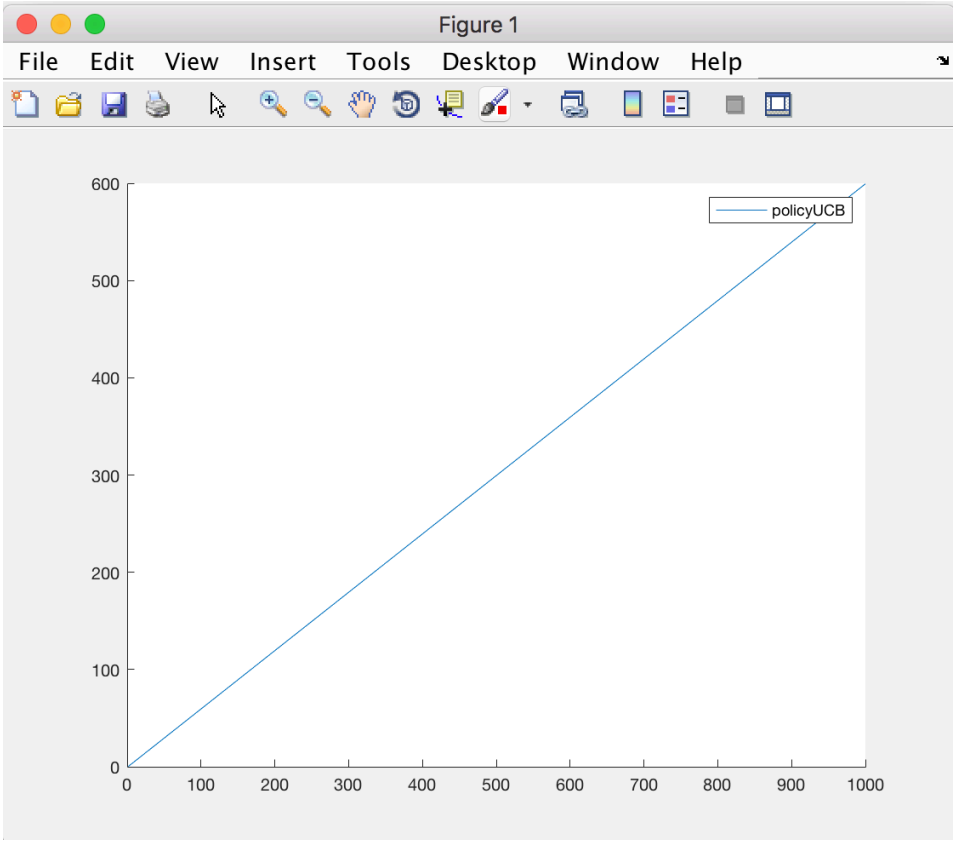
And obviously the following is the sample mean:

$$\frac{1}{m}\sum_{i=1}^{m} X_i = \mu_n$$

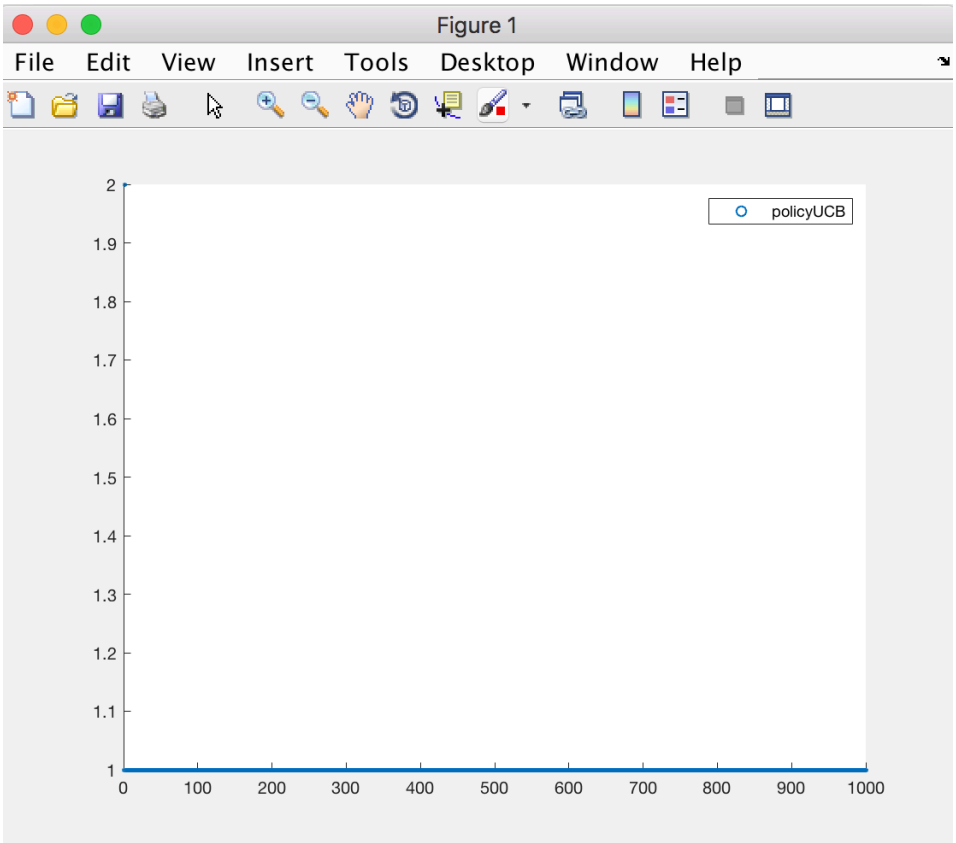
$$\therefore \mu^t = \mu_n^t + \sqrt{\frac{\log(t)}{2m}}$$

4.3

Regret:



Actions:



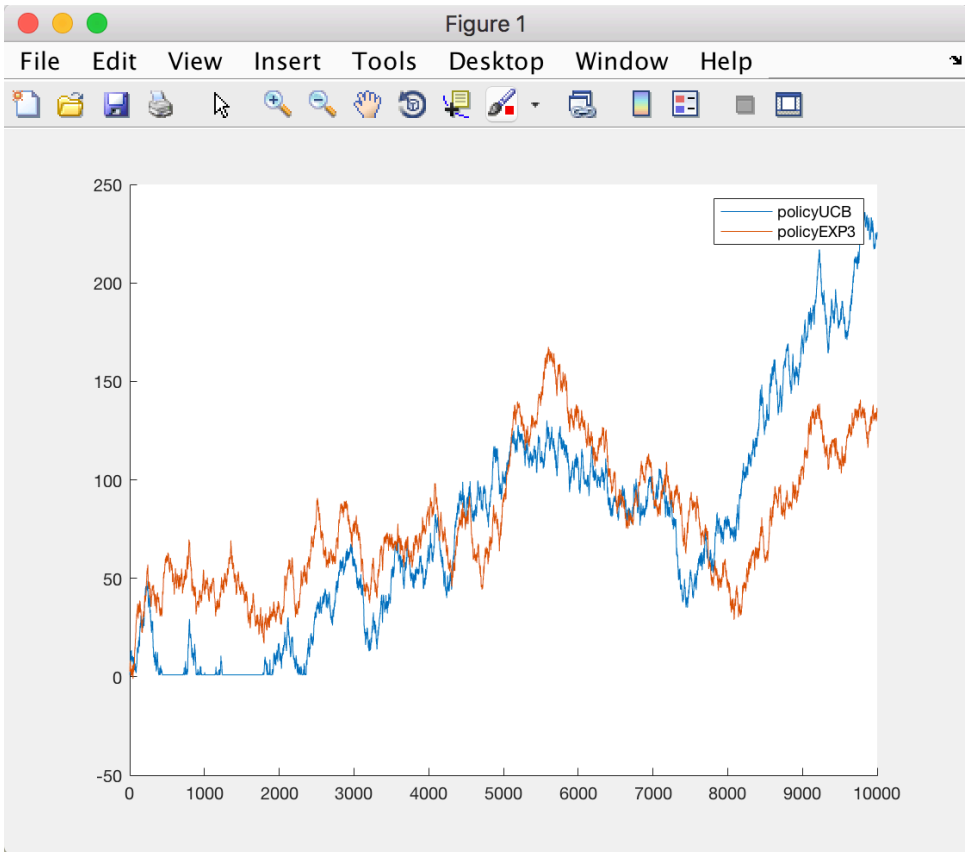
How does UCB behave near the beginning of training?
Answer:
Flatlines, systematically acts out the same thing, regret is almost unchanging and diagonally linear.

How does UCB behave near the end of training?
Answer:
Flatlines, systematically acts out the same thing, regret is almost unchanging and diagonally linear.

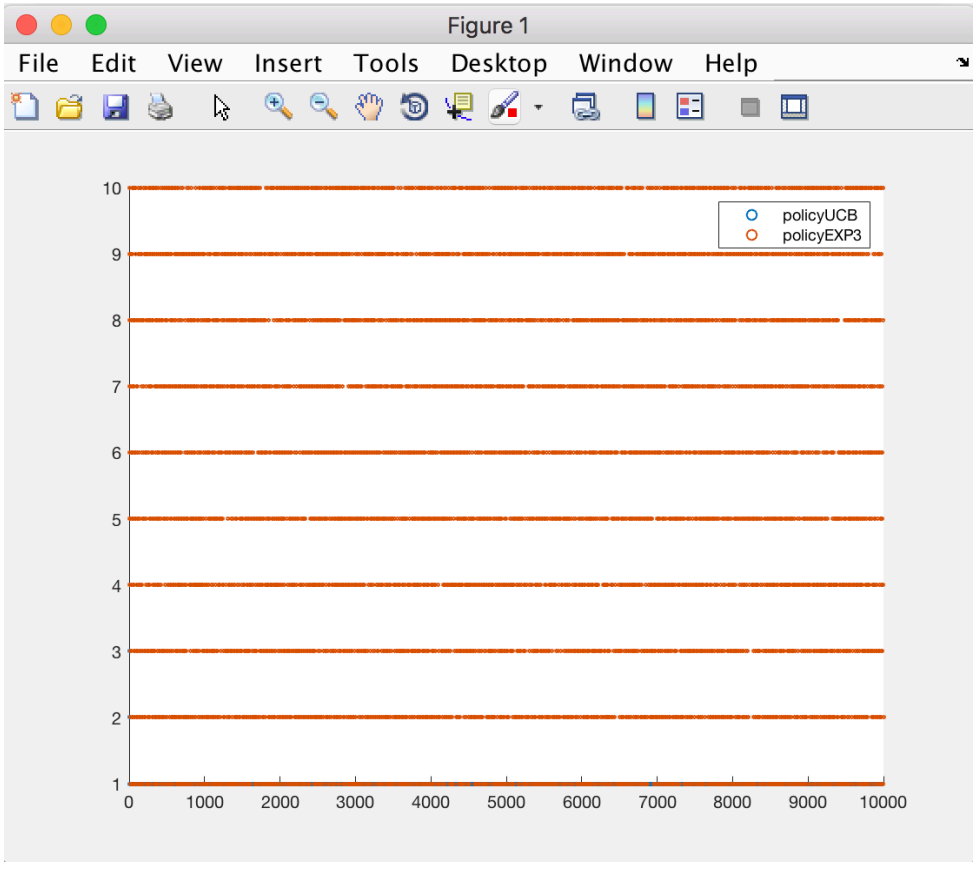
Which action is the policy most confident in?
Answer:
Action 1, it only seems to be choosing that action

4.4

Regret:



Actions:



Which policy performs better on average?

Answer:

This is one of those situations where they both get the best of each other every now and then, but don't quite show a noticeable performance difference. Generally speaking the early portions of the signal is flatlined (sort off for Upper Confidence Bound) whereas it is jagged and varied for EXP3 (searching a lot for best bandit arm to pull, and thereby more prone to acting out something that will trigger regret).

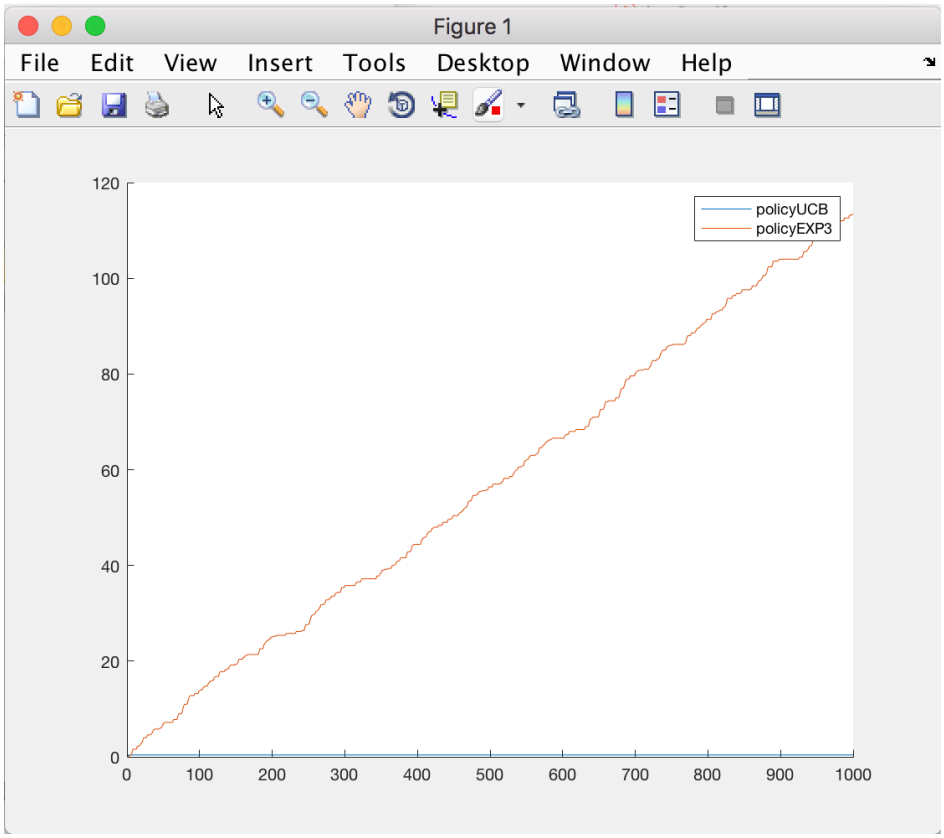
Which policy performs better after 10,000 rounds?

Answer:

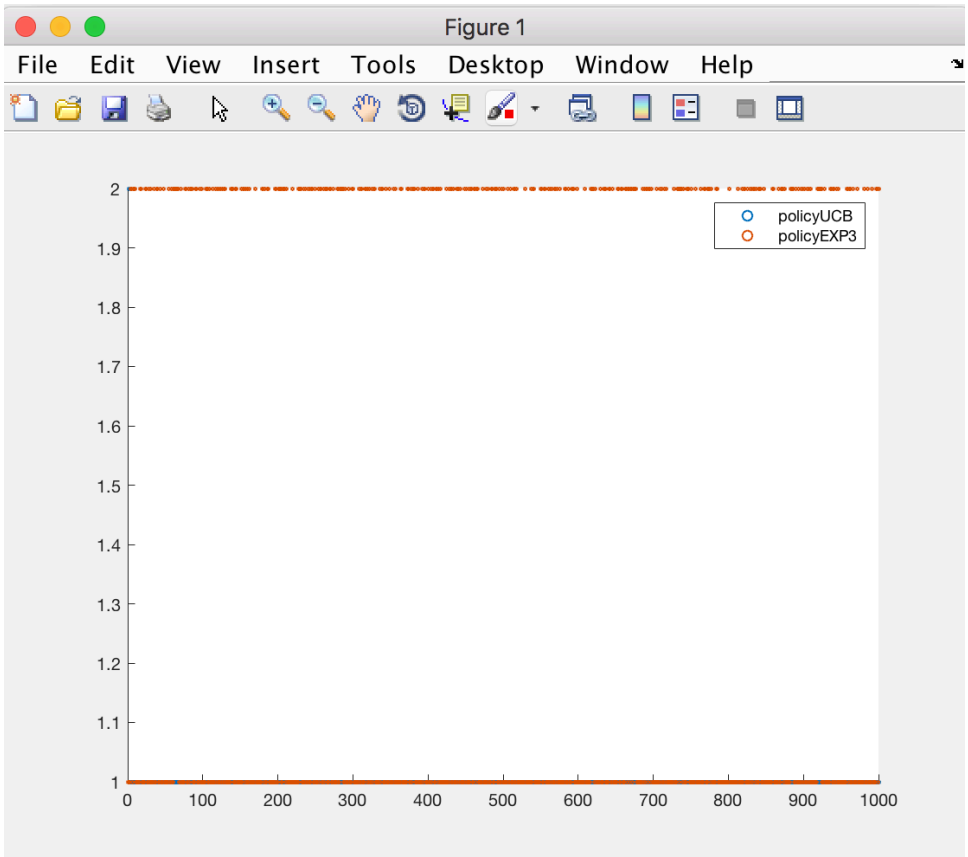
It depends on which test, in some cases it is UCB, whereas in others it is EXP3.

Q4.5

Regret:



Actions:



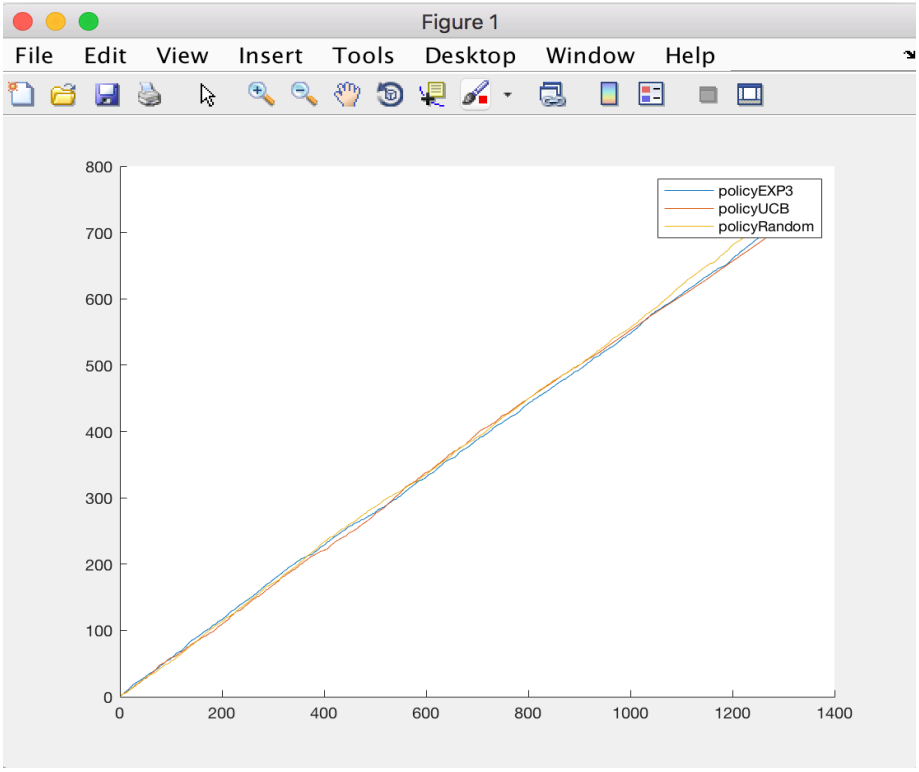
Which policy performs better on average?
Answer:

EXP3

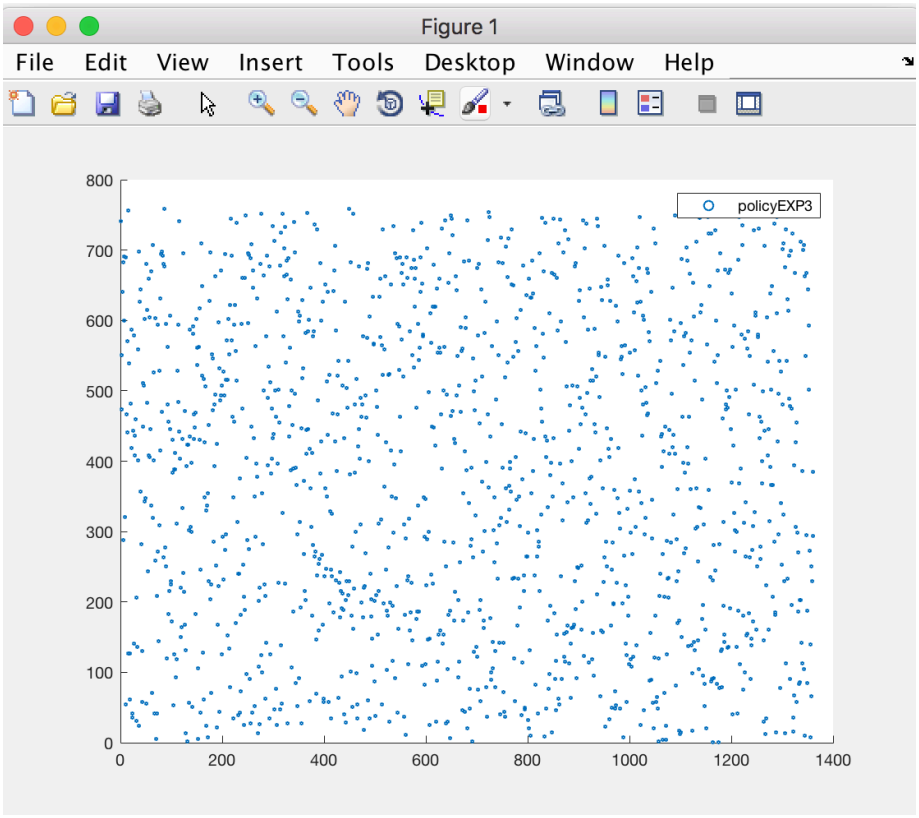
Considering the algorithm that did better: why did it do better? What is the critical difference from the other algorithm?
Answer:

EXP3 is rapidly adjusting and sensitive to odd, quirky game styles, and makes no assumption about the behaviour of nature in that game. UCB is lead-footed so to speak and assumes an underlying continuity and uniformity in the nature of the game, and very slowly and lethargically adapts to adversarial nature.

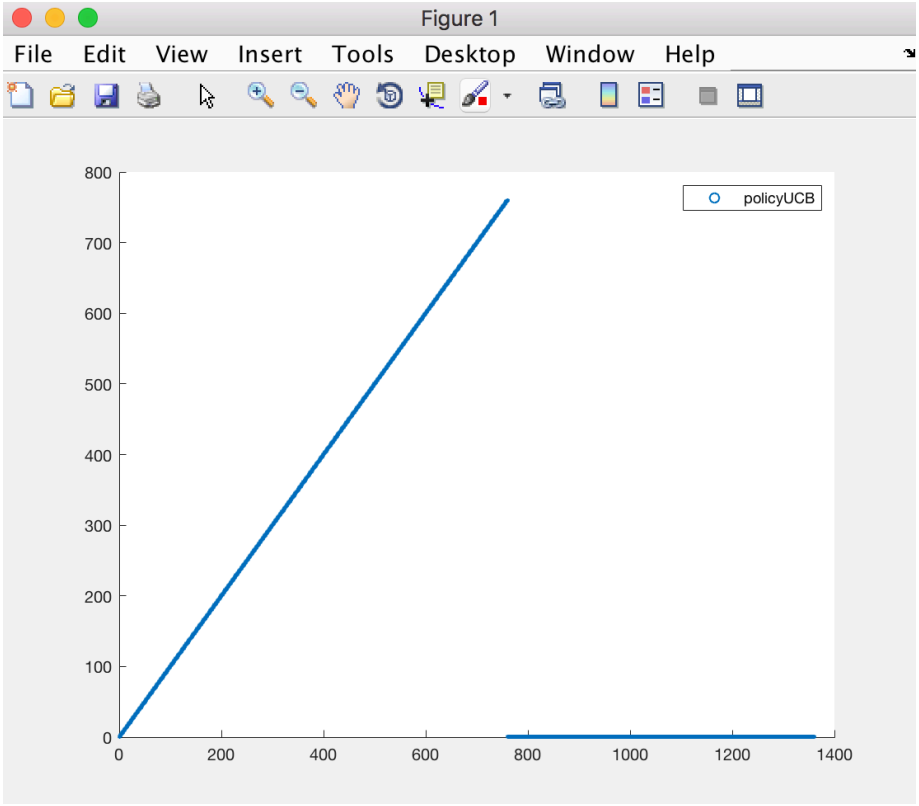
Q5.2 University Website Latency Dataset
Regret:



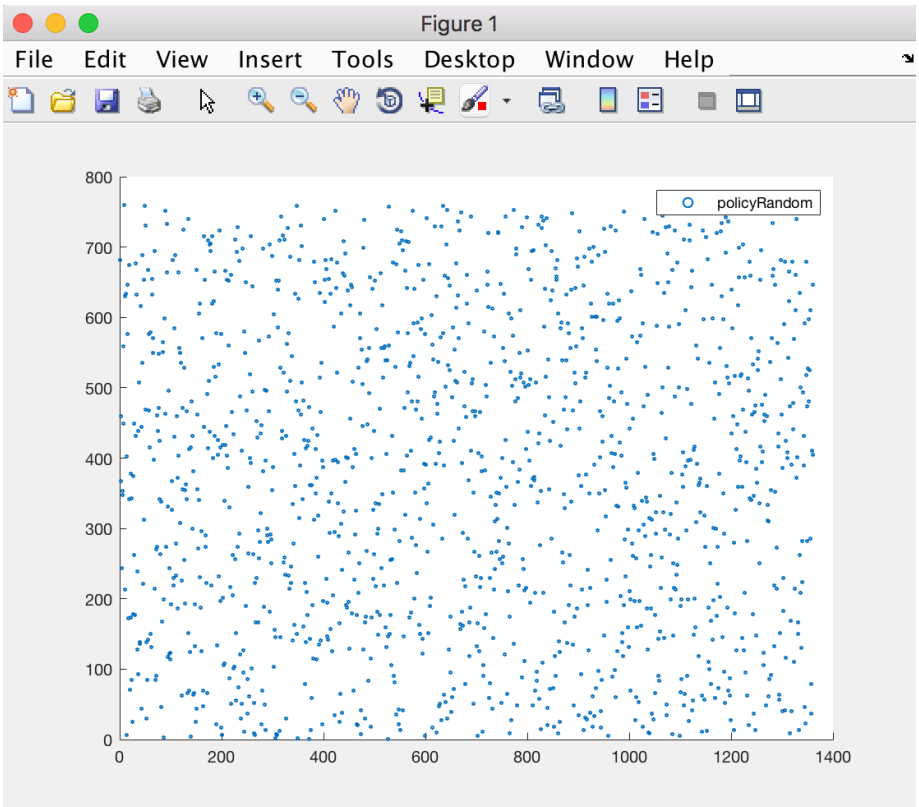
Altogether each algorithms actions over time will be presented separately, because the charts looks like it is filled with information. They will then be presented together. Here is the actions taken by Policy EXP3



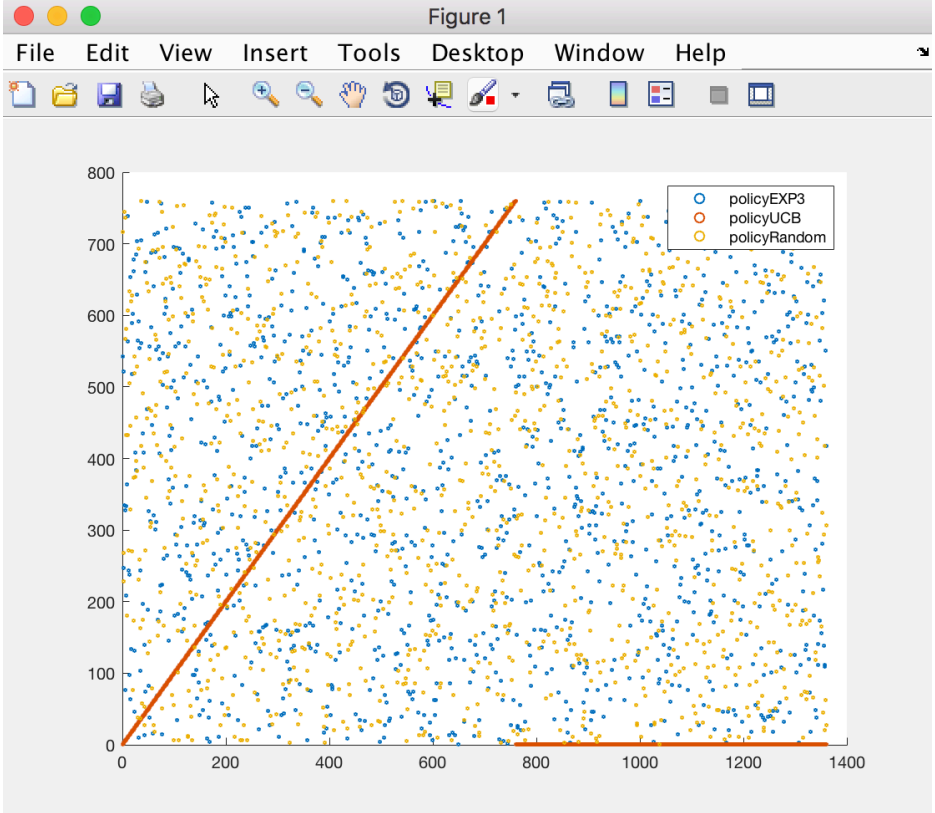
Here are the actions taken by Policy UCB.



Here are the actions taken by the Random Policy



Altogether they look like:



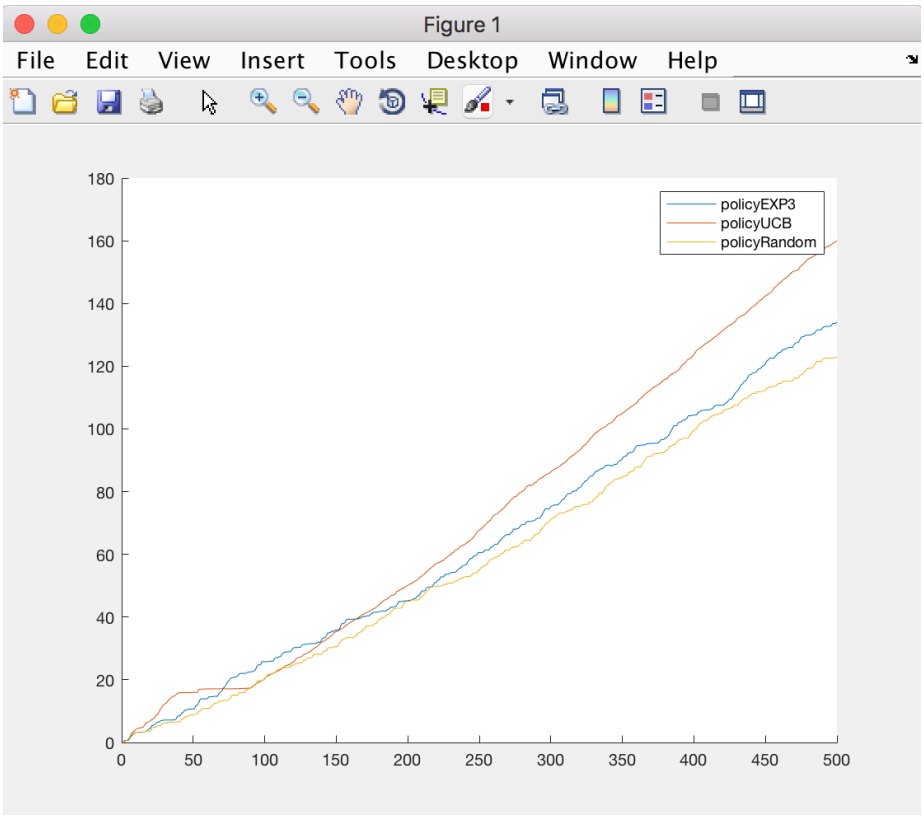
How do UCB and EXP3 compare with each other?
Answer:
As far as regret goes there is not much of a difference between the two other than a slight increase in regret for UCB towards the end of the game.

How do UCB and EXP3 each compare with the random policy?
Answer:
There is no significant difference in performance between random and both UCB and EXP3.

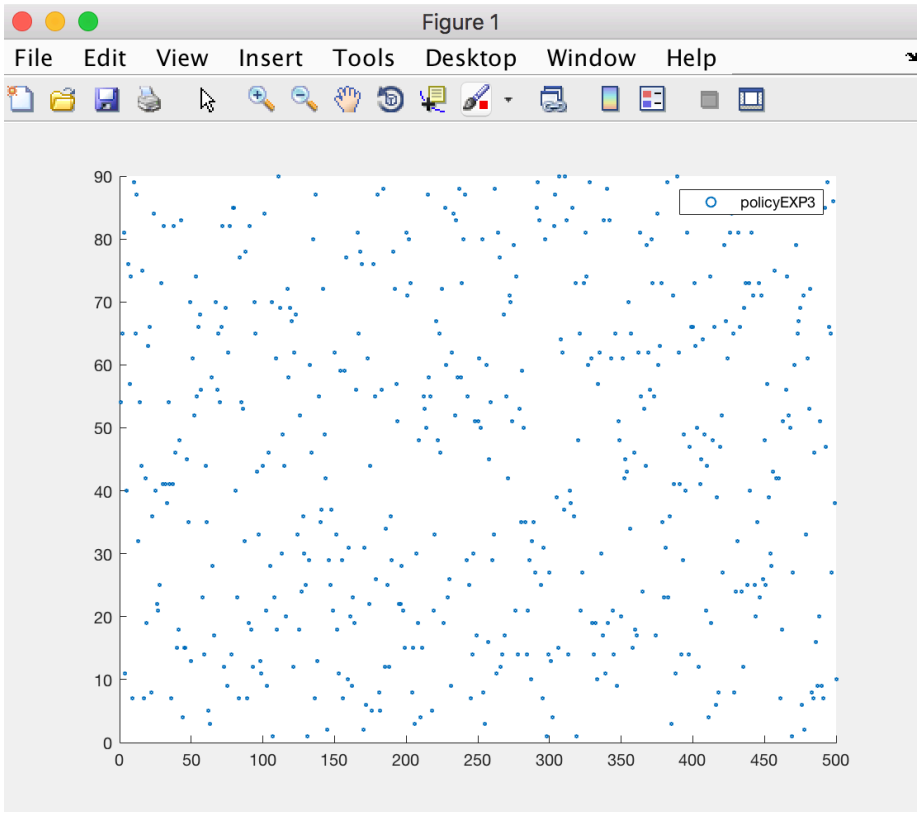
In the actions-over-time plot, there should be a visible difference between the action exploration strategy of UCB vs that of EXP3. What is this difference? Where does this come from (e.g., what line(s) in the algorithm)?
Answer:
Their action behavior is starkly different. One of them (EXP3 seems to choose actions based on all the hypothesis throughout the tests) whereas the other is (Upper Confidence Bound) seems to systematically search throughout the list of all hypothesis (resulting in a sloped line), followed by a sudden flat-lining where it sticks with the same action. These happen because UCB explores the output of all hypothesis before deciding on which action to take.

Q 5.3 Planar Dataset

Regret:



Action (partitioned just like that previous question and then shown together:



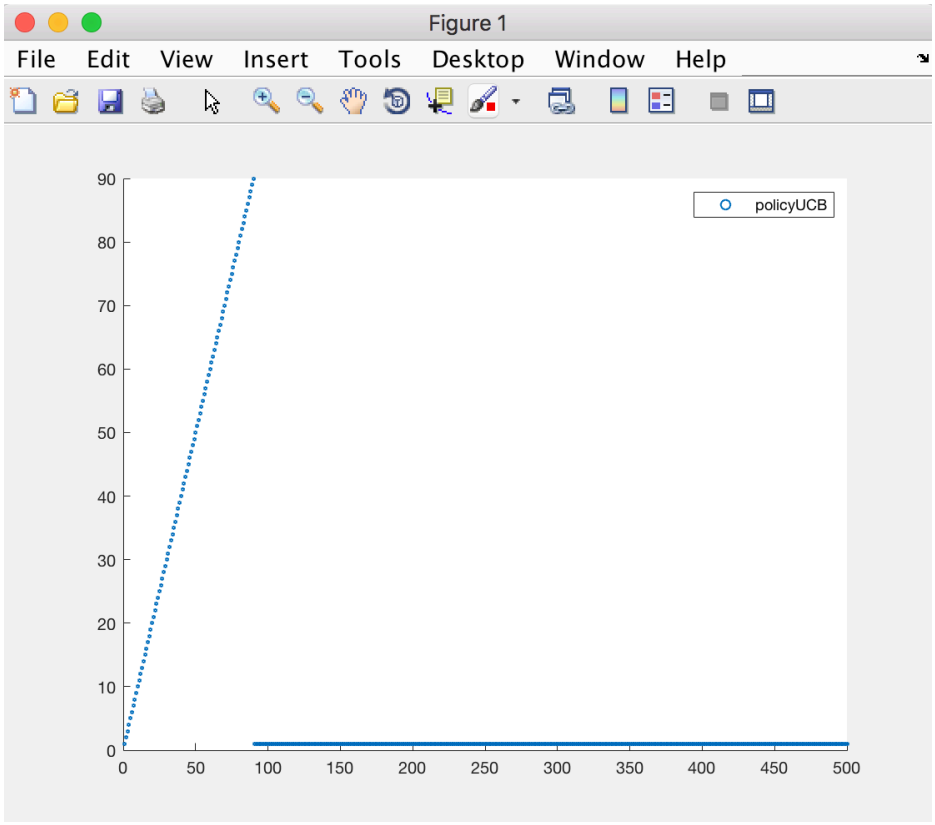


Figure 1
File Edit View Insert Tools Desktop Window Help
policyUCB
0 10 20 30 40 50 60 70 80 90
0 50 100 150 200 250 300 350 400 450 500

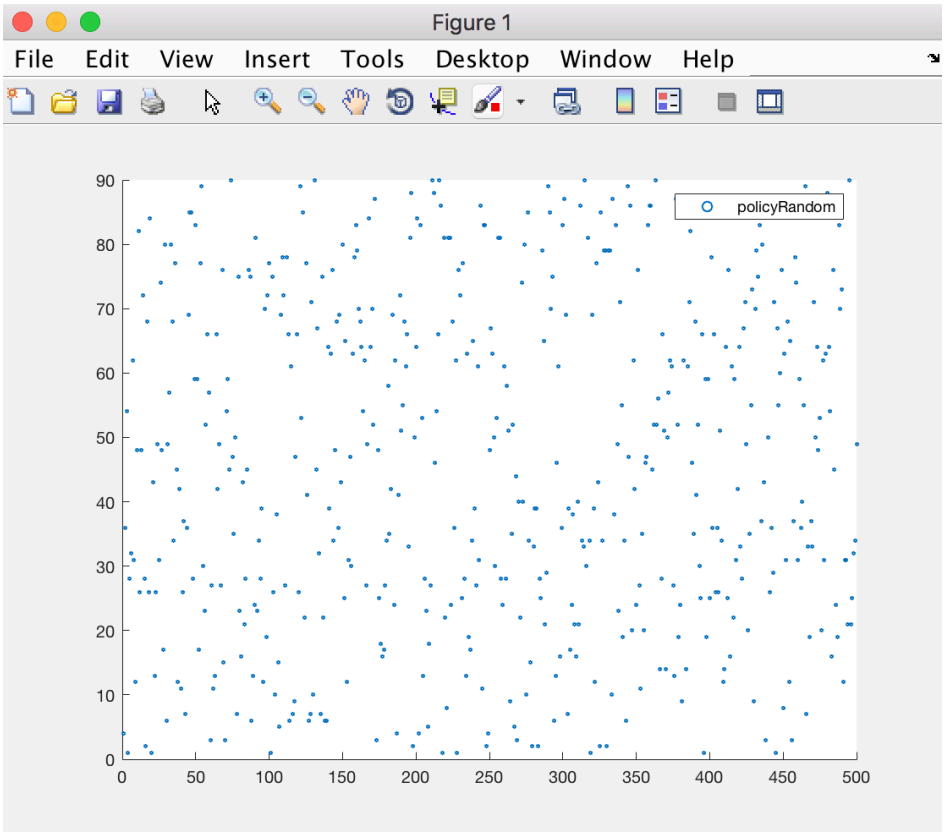
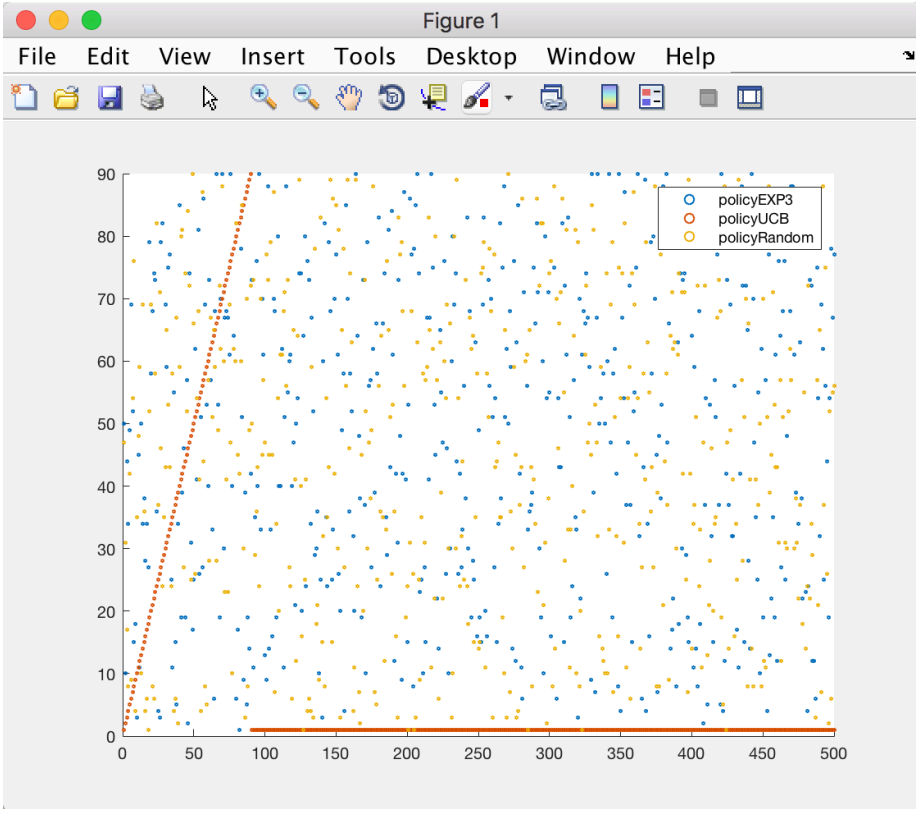


Figure 1
File Edit View Insert Tools Desktop Window Help
policyRandom
0 10 20 30 40 50 60 70 80 90
0 50 100 150 200 250 300 350 400 450 500

Altogether they look like:



Q6
Only the coding for 6.1 was attempted.

Q7
Not attempted