Q2
Ex 2.6

Initially Q₀ dominates over any result given by any of the bandits. So the algorithm runs in batches of 10, where each arm is picked once in each batch. But within these batches the order of picking the bandits is entirely dependant on the result shown so far. And as best bandit, have the highest chance of having the best reward on the batches so far. Thus, the initial few picks in each batch have a much higher chance to be optimum. Thus when going from the last i.e 10n step to 10n+1 there is a sharp increase in % chance optimum, leading to the spike.

$Q_{n+1} = Q_{n-1} + \beta_n [R_n - Q_n]$

$= Q_n + \alpha$

Q3 $\bar{o}_n = \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1})$

$= \alpha + (1-\alpha)\bar{o}_{n-1}$

$= \alpha + (1-\alpha)(\alpha + (1-\alpha)\bar{o}_{n-2})$

$= \alpha + (1-\alpha)\alpha + (1-\alpha)^2\alpha \ldots (1-\alpha)^{n-1}\alpha + 0$

$= \alpha(1 + (1-\alpha) + (1-\alpha)^2 + \ldots (1-\alpha)^{n-1})$

$= \frac{\alpha(1 - (1-\alpha)^n)}{1-(1-\alpha)}$

$= 1 - (1-\alpha)^n \quad n \geq 1$

$Q_{n+1} = Q_n + \beta_n [R_n - Q_n]$

$= Q_n + \frac{\alpha}{\bar{o}_n} [R_n - Q_n]$

$= Q_n \left[1 - \frac{\alpha}{\bar{o}_n}\right] + \beta_n R_n$

$= \beta_n R_n + \left(1 - \frac{\alpha}{\bar{o}_n}\right)(\beta_{n-1} R_{n-1} + \left(1 - \frac{\alpha}{\bar{o}_{n-1}}\right)$

$= \beta_n R_n + (1-\beta_n)\beta_{n-1} R_{n-1} + (1-\beta_n)(1-\beta_{n-1}) R_{n-2} \cdots$

$+ (1-\beta_n)(1-\beta_{n-1})(1-\beta_{n-2}) \cdots (1-\beta_1) Q_1$

$= \beta_n R_n + \frac{(1-\alpha)\left(1-(1-\alpha)^{n-1}\right)}{1-(1-\alpha)^n} \beta_{n-1} R_{n-1} + \frac{(1-\alpha)^2\left(1-(1-\alpha)^{n-2}\right)}{1-(1-\alpha)^n} \beta_{n-2} R_{n-2}$

$+ \frac{(1-\alpha)^n\left(1-(1-\alpha)^0\right)}{1-(1-\alpha)^n} Q_1$

(using $1-(1-\alpha)^0 = 1-1 = 0$)

$= \sum_{i=1}^{n} \frac{(1-\alpha)^{n-i}\left(1-(1-\alpha)^i\right)}{1-(1-\alpha)^n} R_i \beta_i$

$= \sum_{i=1}^{n} \frac{(1-\alpha)^{n-i}\left(1-(1-\alpha)^i\right)}{1-(1-\alpha)^n} \frac{\alpha R_i}{1-(1-\alpha)^i} = \sum_{i=1}^{n} \frac{R_i (1-\alpha)^{n-i} \alpha}{1-(1-\alpha)^n}$

$$= \frac{\alpha}{1-(1-\alpha)^n}\left(R_n + (1-\alpha)R_{n-1} + (1-\alpha)^{n-1}R_1\right)$$

(This has no factor of Q_1, & factors of R_i are clearly ~~exponential~~ with geometric r $r = (1-\alpha)$)

To show weighted sum of factors to be 1

sum of factors, $= \frac{\alpha}{(1-(1-\alpha)^n)}\left(1 + (1-\alpha) \dots (1-\alpha)^{n-1}\right)$

$$= \frac{\alpha}{1-(1-\alpha)^n}\left(\frac{1(1-(1-\alpha)^n)}{1-(1-\alpha)}\right)$$

$$= \frac{\alpha}{1-(1-\alpha)^n} \frac{(1-(1-\alpha)^n)}{\alpha} = \underline{1}$$

hence Q_{n+1} is weighted sum of R_i's

Q4

Initially optimistic explores a lot, but allowing it make better greedy decisions but this effect quickly wears off, & then the q^* change significantly over time but optimistic due to its $\varepsilon = 0$, doesn't explore enough to keep up. While ε-greedy doesn't suffer from this as it continues to explore a ($\varepsilon = 0.1$) & isn't weighed down by past explorations too much due to constant step progression.

In stationary both UCB improve over time as their estimate of q^* improves and they increasingly waste less time exploring. UCB with higher c lags behind as it spends more time exploring. Optimistic however spends very little time exploring with most of the exploration in the early part. by which time it q^*

In non-stationary UCB actually optimistic starts logging, since with ($\epsilon = 0$) it severly reduces its exploration after initial few terms, so as q^* changes, optim fails to adjust.

UCB explores a little more but not as much ϵ-greedy since $\ln t$ grows for more slowly than N_t, thus eventually UCB also becomes outdated, if c is higher there is more exploration hence UCB with higher c perform better overtime.