MAB Simulations

Sandeep Gangarapu

Simulation set up

We have 10 arms which have a normal outcome distribution with different means and variances. The values of these true means and variances are given below.

Every time we pull an arm (make an allocation), we sample from the normal distribution of given by the mean and variance of that arm.

We see from the below true values that ARM-8 should the winning arm as it has the highest mean of 4.9.

```
true_means = c(3.36139279, 2.440392, 4.12747587, 0.25, 4.04024982, 2.8280871, 1.48811249, 0.2334786, 4.9

true_vars = c(3.84896514, 3.7338355, 1.88719468, 2.47073726, 4.64474196, 1.97727022, 4.86978148, 2.6220
```

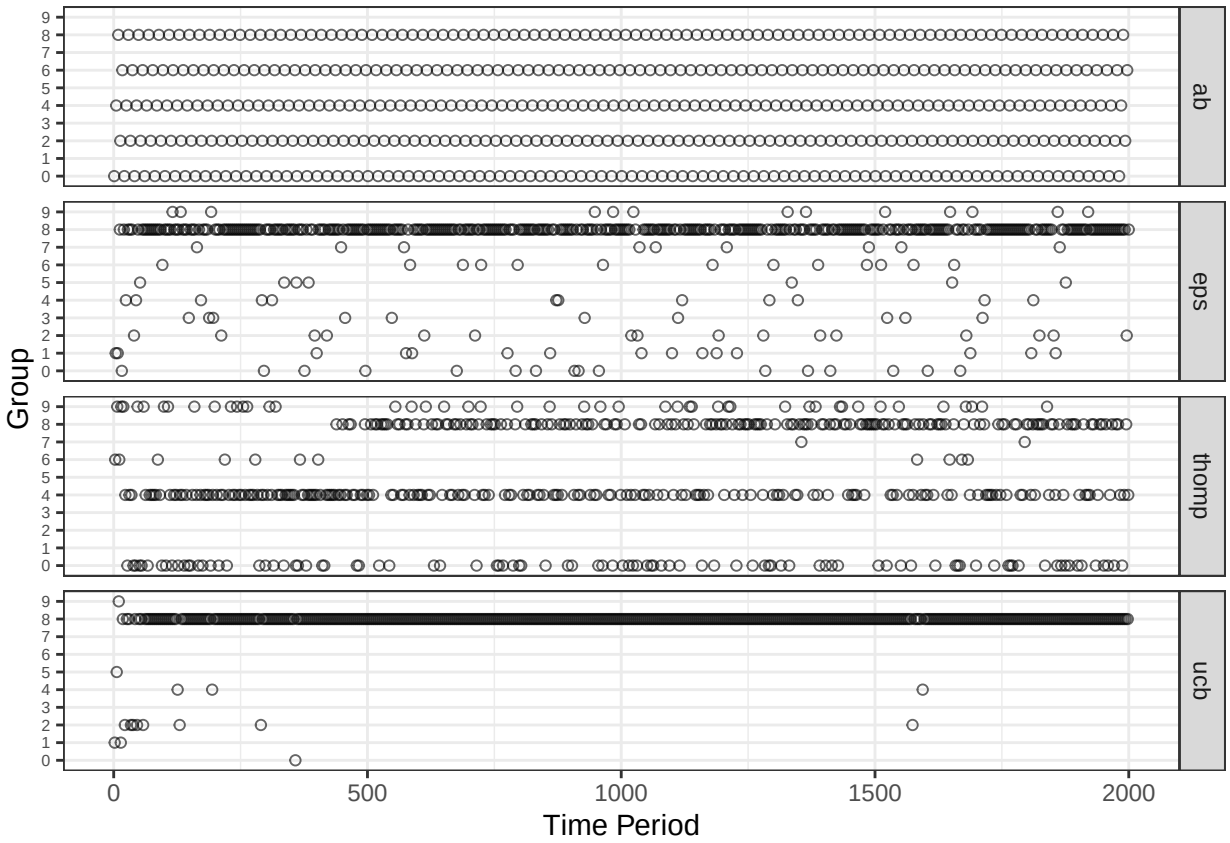
Time Horizon is the number of allocations we have available synonymous to number of experimental units in an experiment. In this simulation, we set time horizon to 2000 units.

We run this simulation for 20 different times using various seeds.

Group allocation graphs

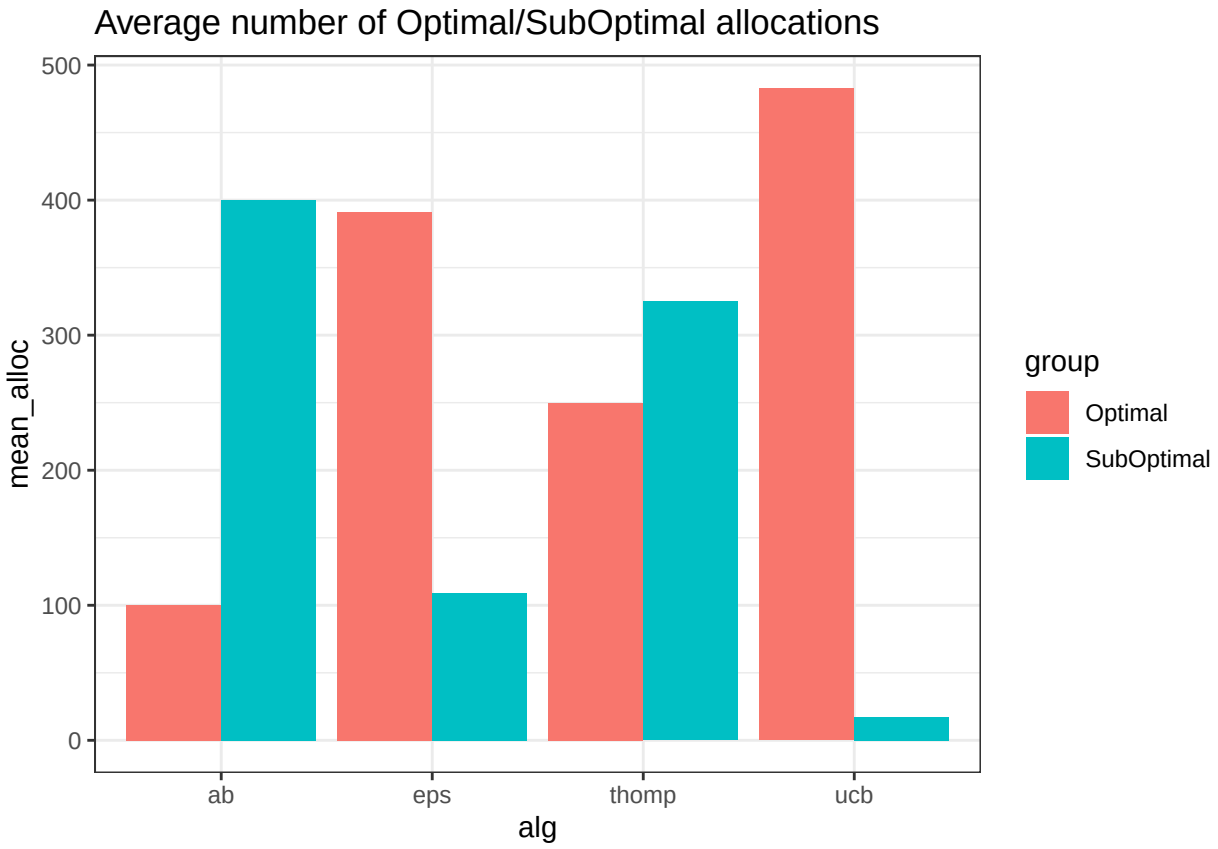
We first look at the allocations for widely used algorithms.

- ab - AB Testing
- eps - Epsilon Greedy
- thomp - Thompson Sampling
- ucb - Upper confidence bound



UCB is the most efficient. Looks like Thompson sampling is not as efficient as perceived. In this case, it is hovering between 8, 4, 0

Average number of optimal/sub-optimal allocations in each algorithm



UCB is the most optimal as it is efficient. AB does random allocation, so, it is the least optimal.

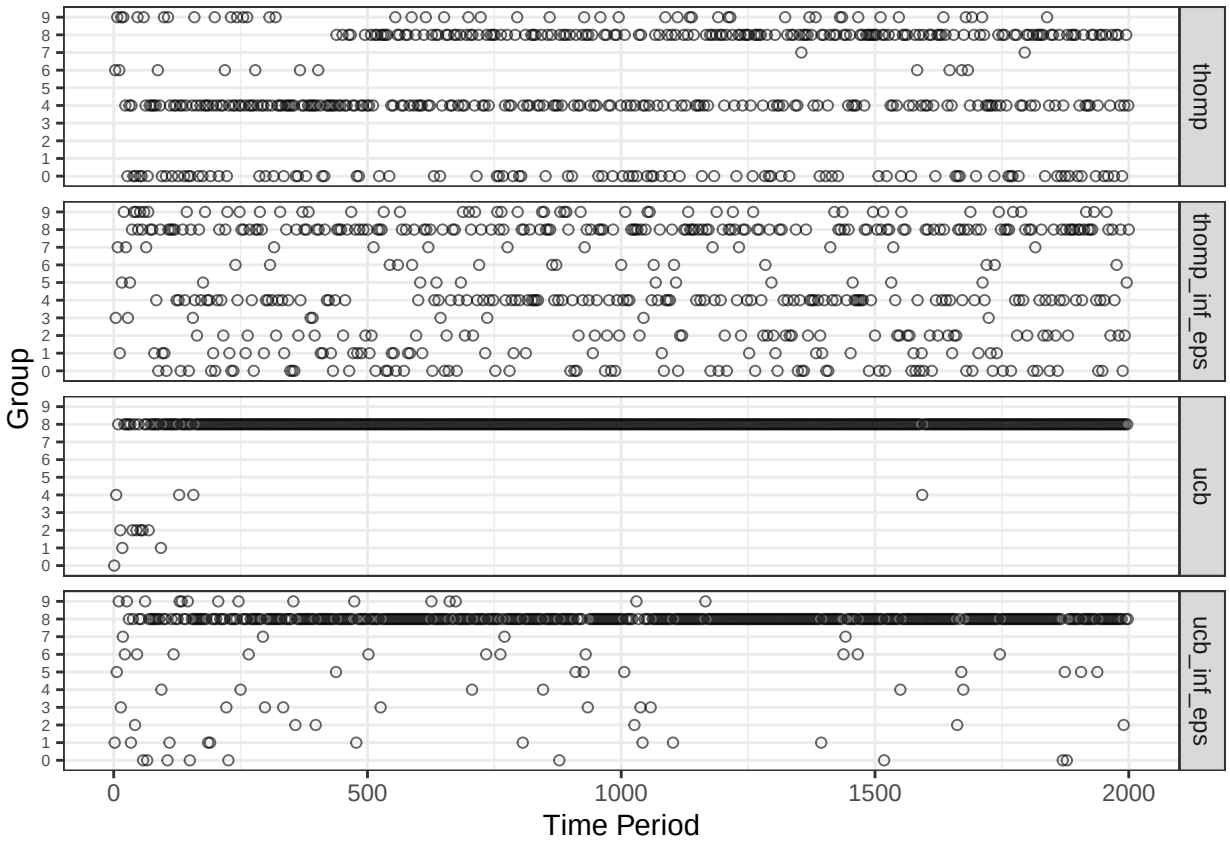
Group allocations for Inference based algorithms algorithms.

For inference based allocation, we deviate from the main allocation algorithm with the probability of ϵ_n defined by

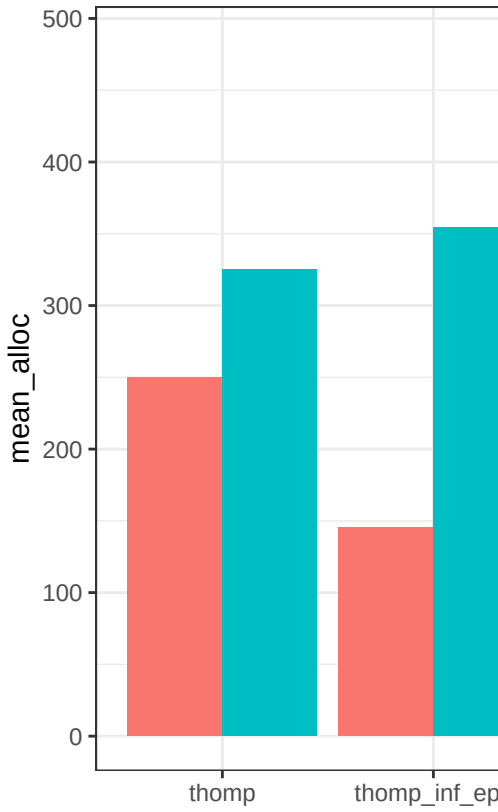
$\eta = \frac{\sum \frac{\sigma^2}{n}}{\xi K}$ $\epsilon_n = \frac{\eta}{1+\eta}$

ξ and K are user defined parameters that we will discuss later.

The main idea is that by deviating from main algorithm and making inference based allocations, we learn more about an arm. Learning here implies reducing standard error of an arm. This ofcourse comes at the cost of Utility.



"_inf_eps" suffix suggests that the algorithms makes inference based allocation with some probability. In these algorithms, the allocations are not as concentrated around few arms as those of the main algorithms (UCB, Thompson Sampling).

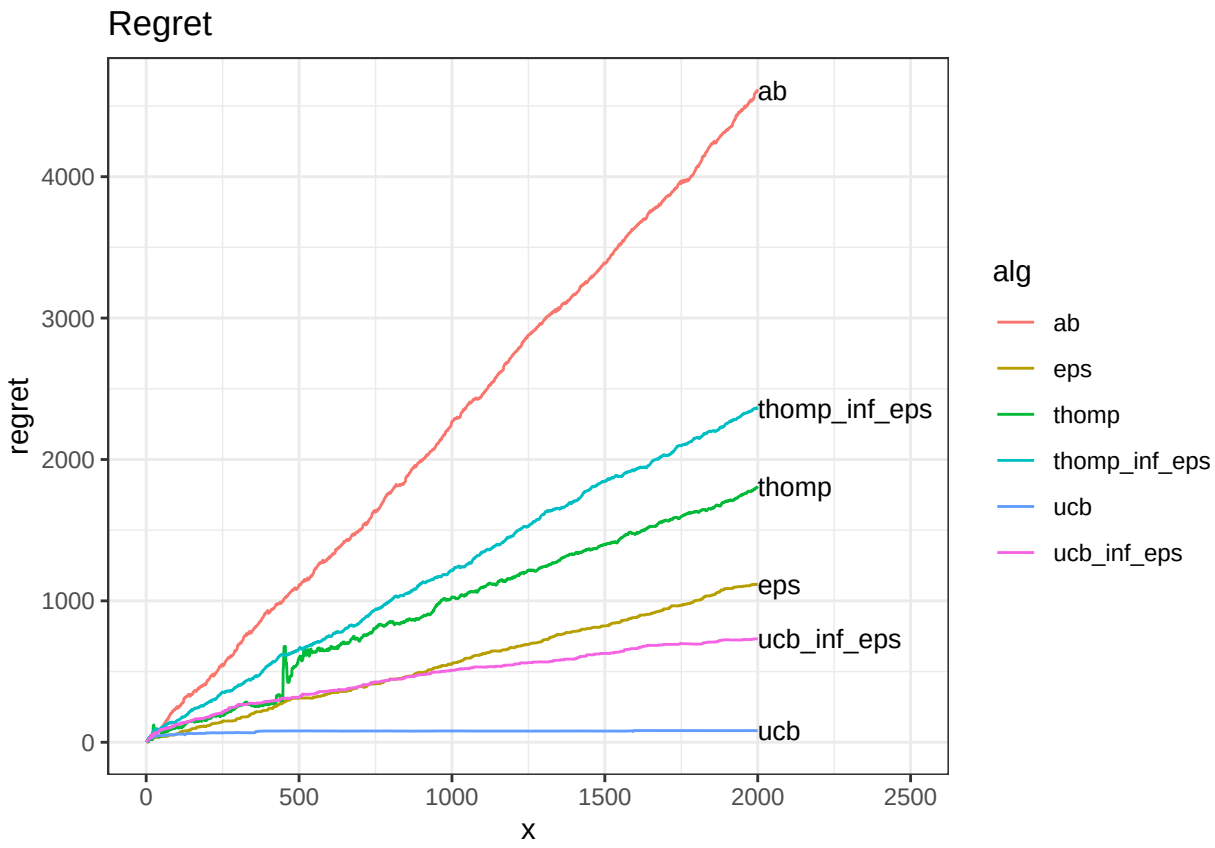


Average number of optimal/sub-optimal allocations in each algorithm

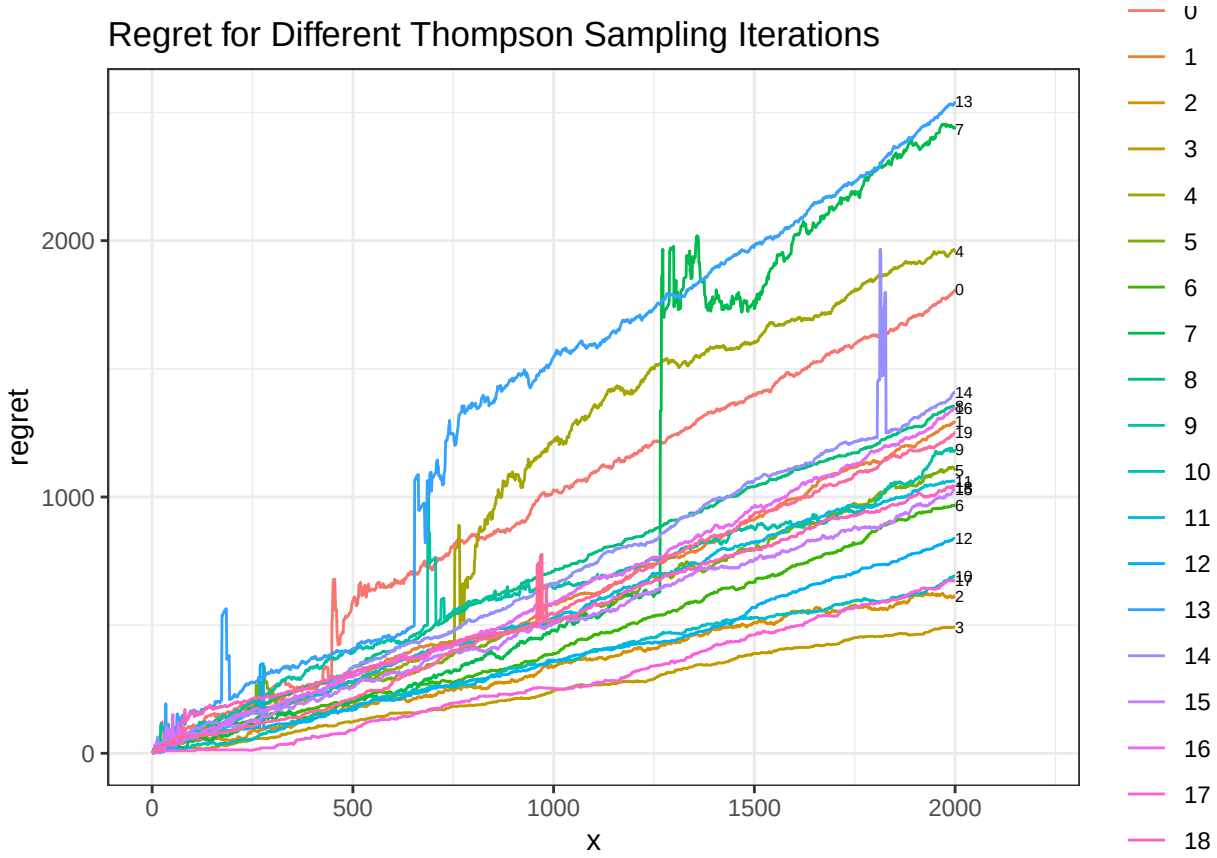
UCB_INF_EPS is slightly worse compared to UCB but is not as bad as Thompson Sampling based algorithms

The relative performance of these algorithms can be formalized in regret analysis.

Regret Analysis



UCB algorithm is the most efficient. One interesting observation is that ucb_inf_eps performs better than Thompson Sampling. This could be because of a bad seed (one random bad simulation). So, we check to see if this the same case for other iterations (other random seeds).



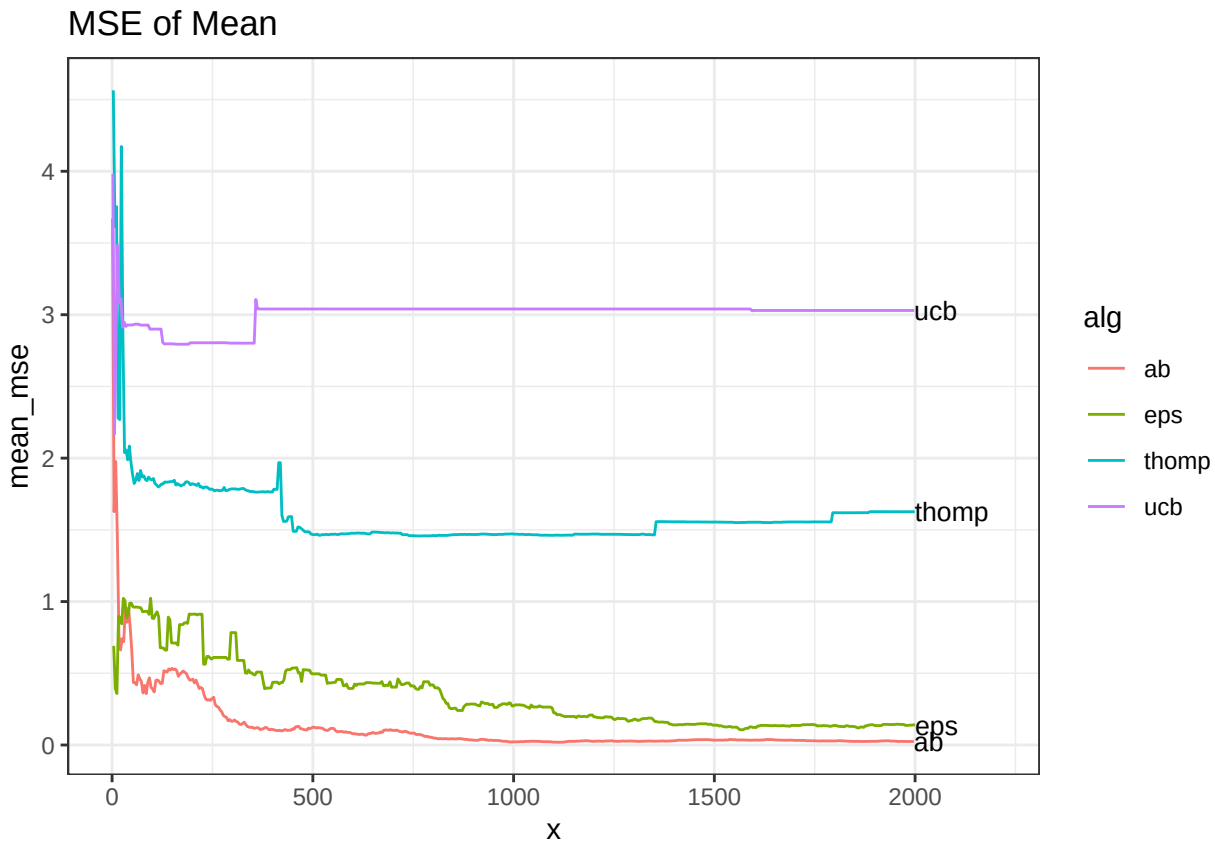
There is a huge variance among iterations, however, 0th iteration which was considered in comparison with other algorithms, sits right in the middle, this means we did not get an extreme performing iteration. What we have is reasonable.

The average performing iteration (ite=6) has regret of 1000, which seems to be very similar to regret of Epsilon_greedy

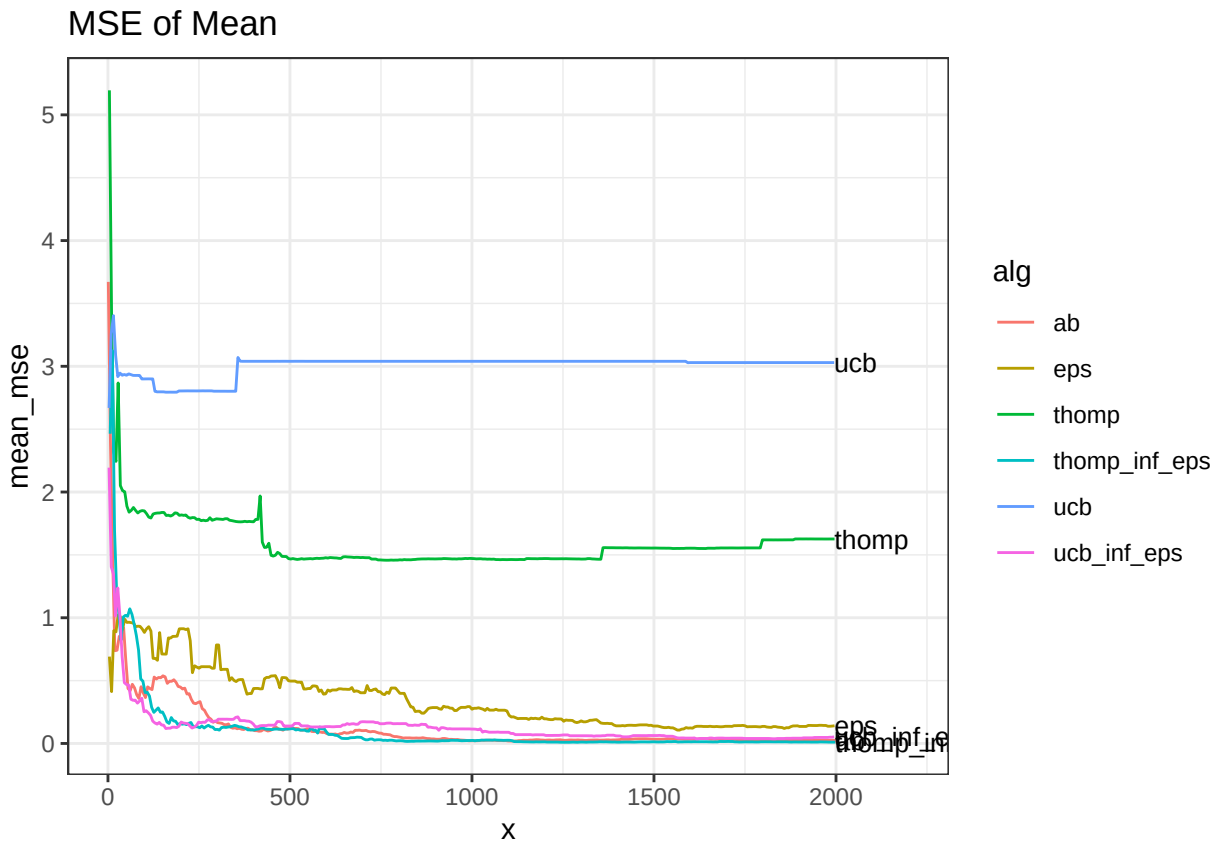
Mean Squared error of mean estimates

At each time horizong we use sample mean to estimate mean of each arm and calculate error = est.mean-true.mean.

$MSE = \sum_k (error_k)^2$

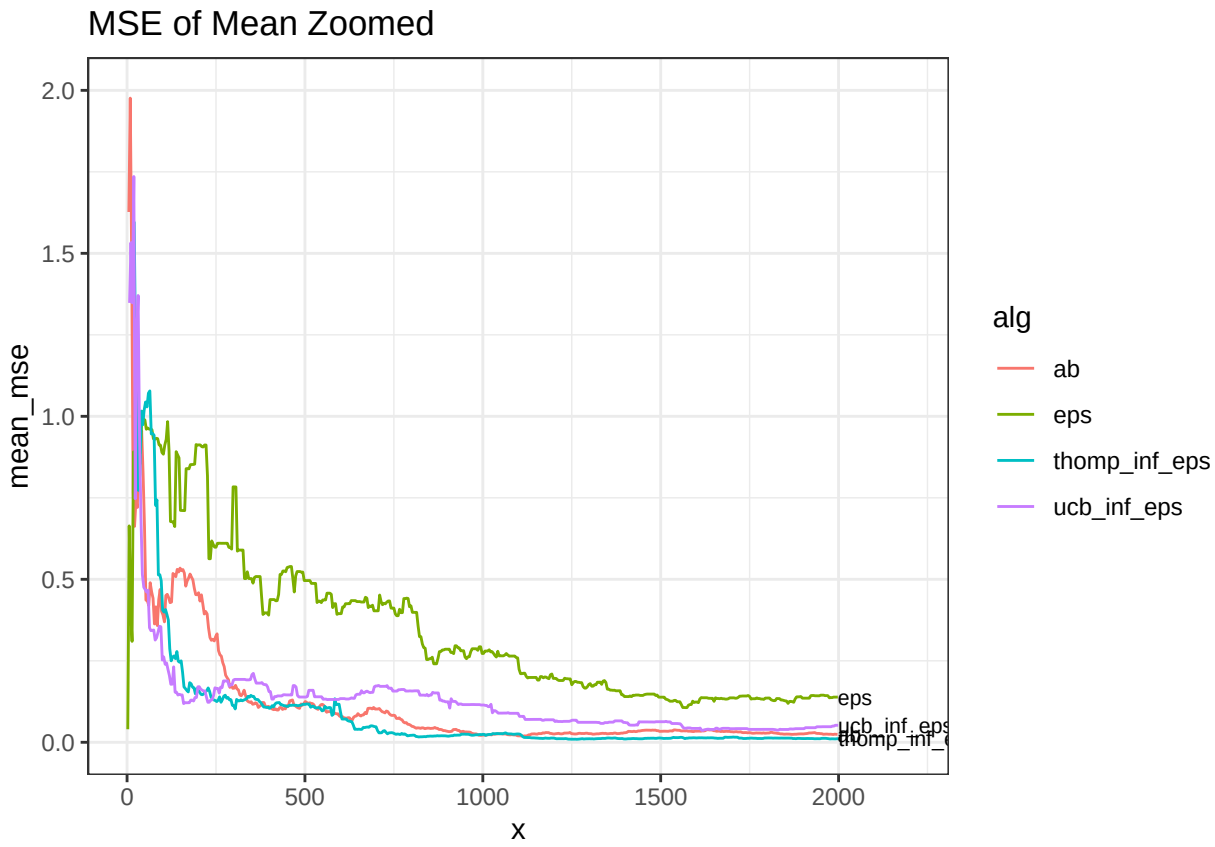


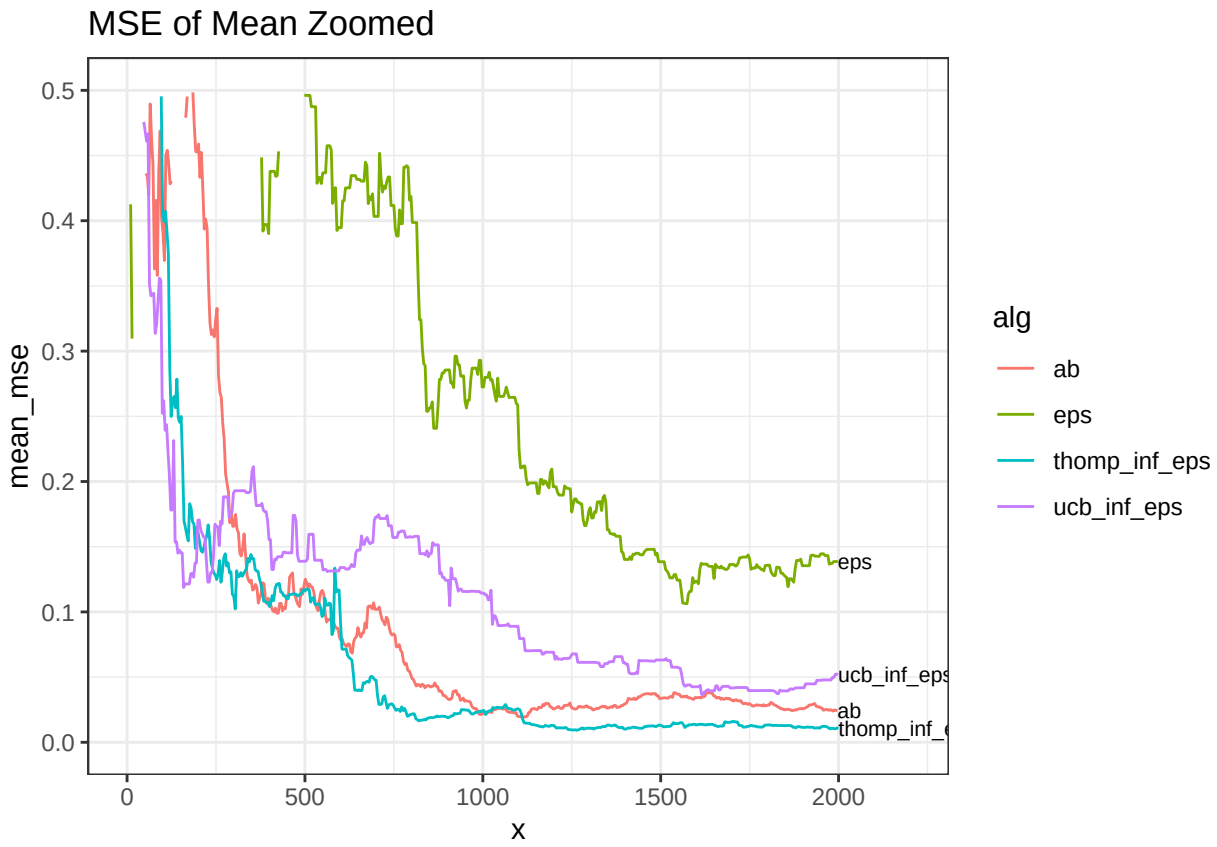
UCB and Thompson Sampling have higher MSE compared to AB or Epsilon Greedy. This means that the estimates of AB and Epsilon-Greedy are more accurate.



The Inference based algorithms are performing much better than Bandit algorithms in terms if MSE and are comparable to traditional algorithms.

Difference between the above algorithms is clear during later allocations. So we limit the Y axis to see these differences.


MSE of Mean Zoomed
mean_mse
2.0
1.5
1.0
0.5
0.0
0
500
1000
1500
2000
x
eps
ucb_inf_eps
alg
ab
eps
thomp_inf_eps
ucb_inf_eps



AB, UCB_INF_EPS, THOMP_INF_EPS perform the best and almost indistingushable. However, INF_EPS algorithms perform as good as AB.

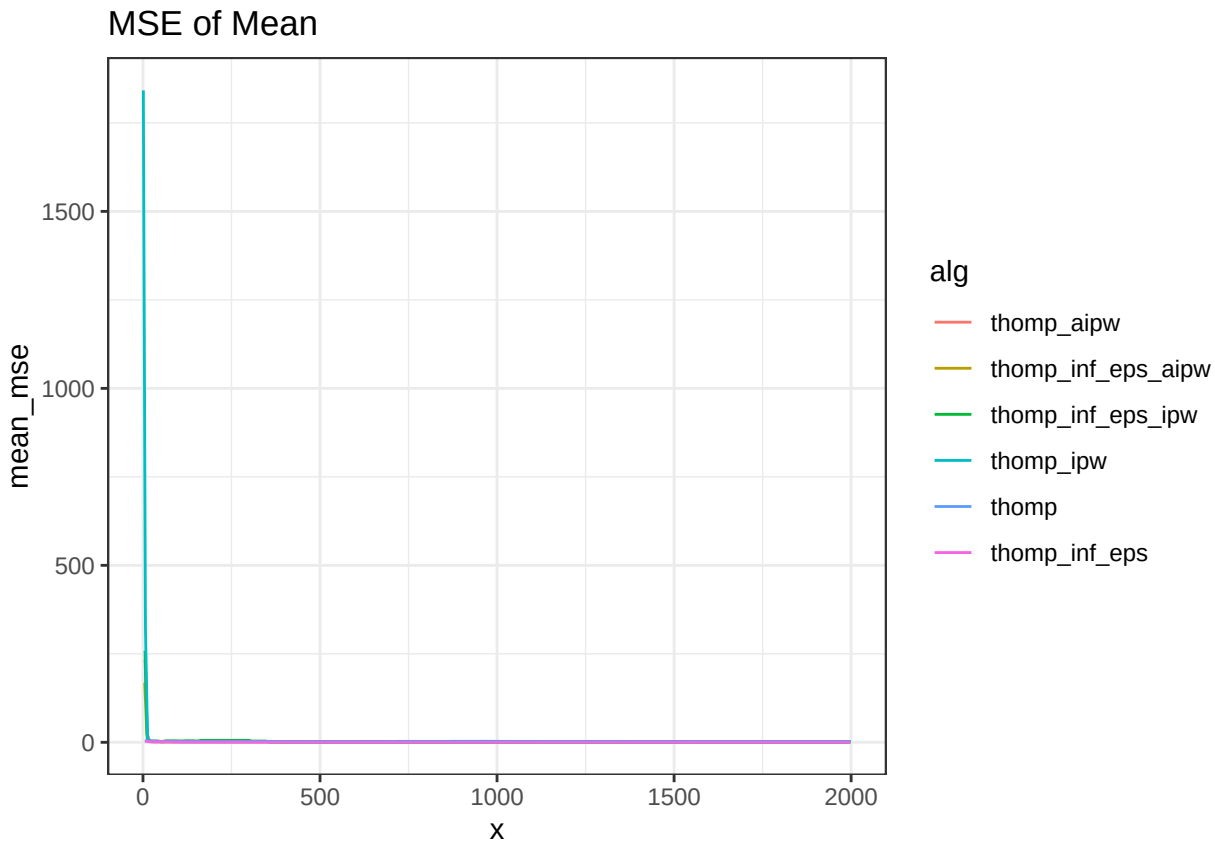
MSE of Weighed estimators.

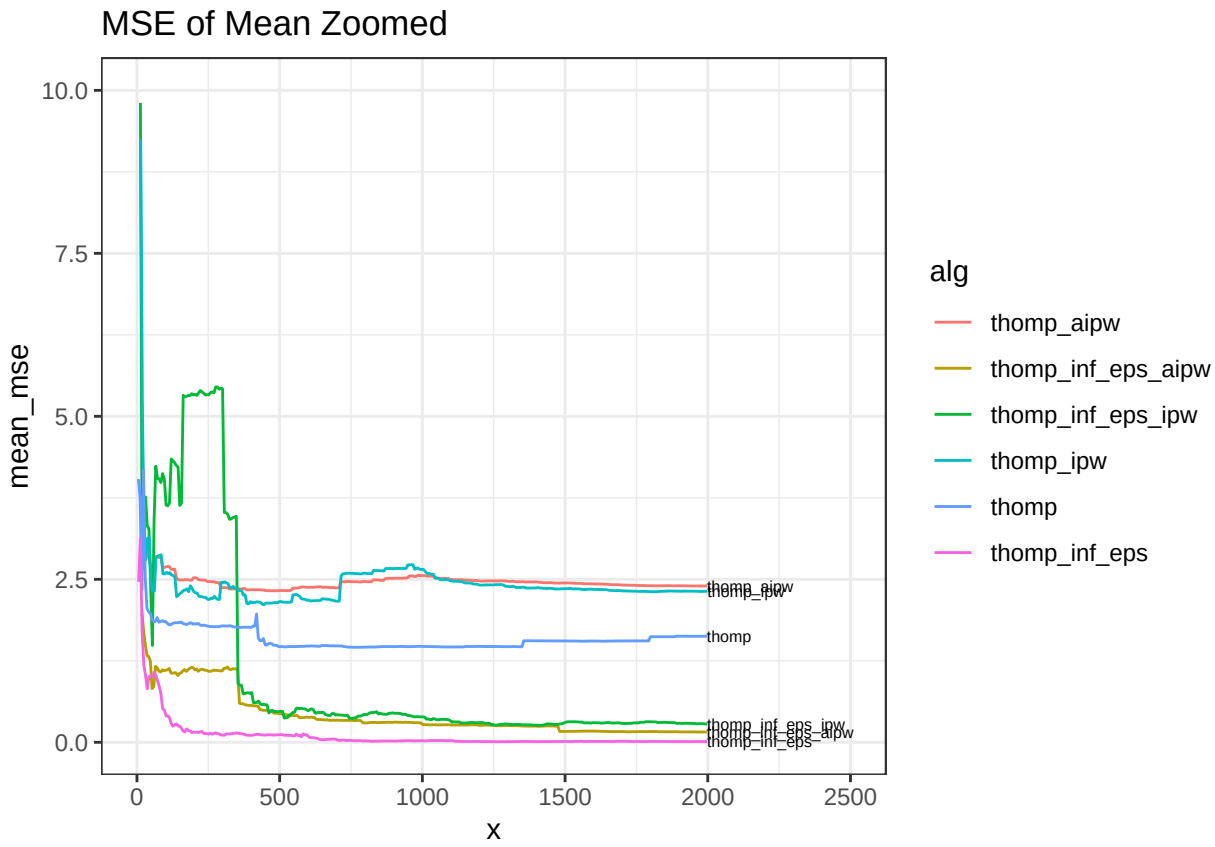
We also want to see how the weighed estimators perform wrt original MAB algorithms and their INF versions. We use two weighed estimators.

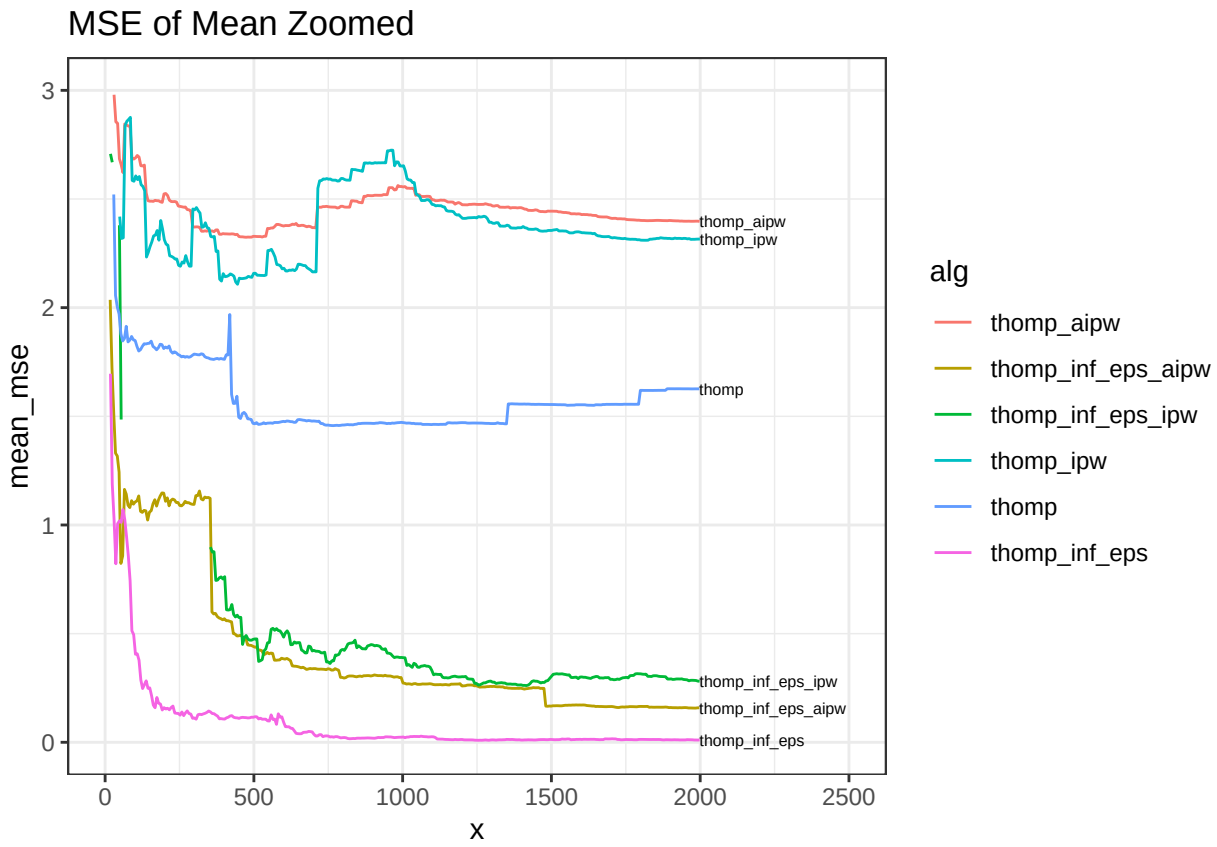
- ipw - Inverse propensity weighing
- aipw - Augmented inverse propensity weighing

These are not new MAB algorithms. The are only estimators to calculate mean of already existing algorithm. The main reason to use these estimators is for their unbiasedness properties. Theoretically, IPW is unbiased but had high variance and AIPW is unbiased and has lower variance.

In order to use IPW and AIPW, we need to know the propensity of choosing an arm at every allocation. Propensity of every arm at every time period can be calculated for Thompson Sampling using Monte Carlo simulations but it cannot be calculated for UCB. So we only use weighed estimators for TS.


MSE of Mean
mean_mse
1500
1000
500
0
0
500
1000
1500
2000
x
alg
thomp_aipw
thomp_inf_eps_aipw
thomp_inf_eps_ipw
thomp_ipw
thomp
thomp_inf_eps


MSE of Mean Zoomed
mean_mse
10.0
7.5
5.0
2.5
0.0
0
500
1000
1500
2000
2500
x
thomp_aipw
thomp_ipw
thomp
thomp_inf_eps_ipw
thomp_inf_eps_aipw
thomp_inf_eps
alg
thomp_aipw
thomp_inf_eps_aipw
thomp_inf_eps_ipw
thomp_ipw
thomp
thomp_inf_eps



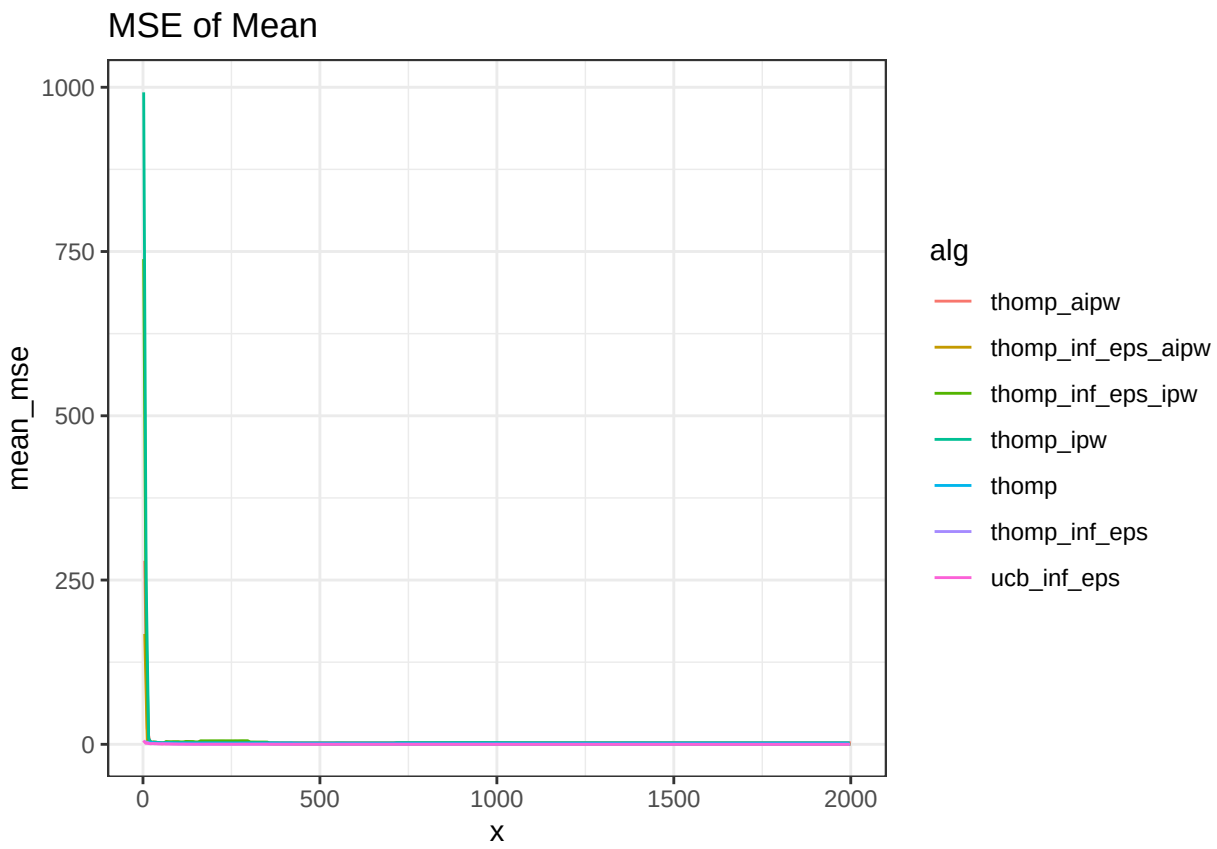
MSE of weighed estimators is very high at the start owing to lower propensity of arm selctions increasing the variance of estimate. This slowly goes down as we gather more data (horizon increase).

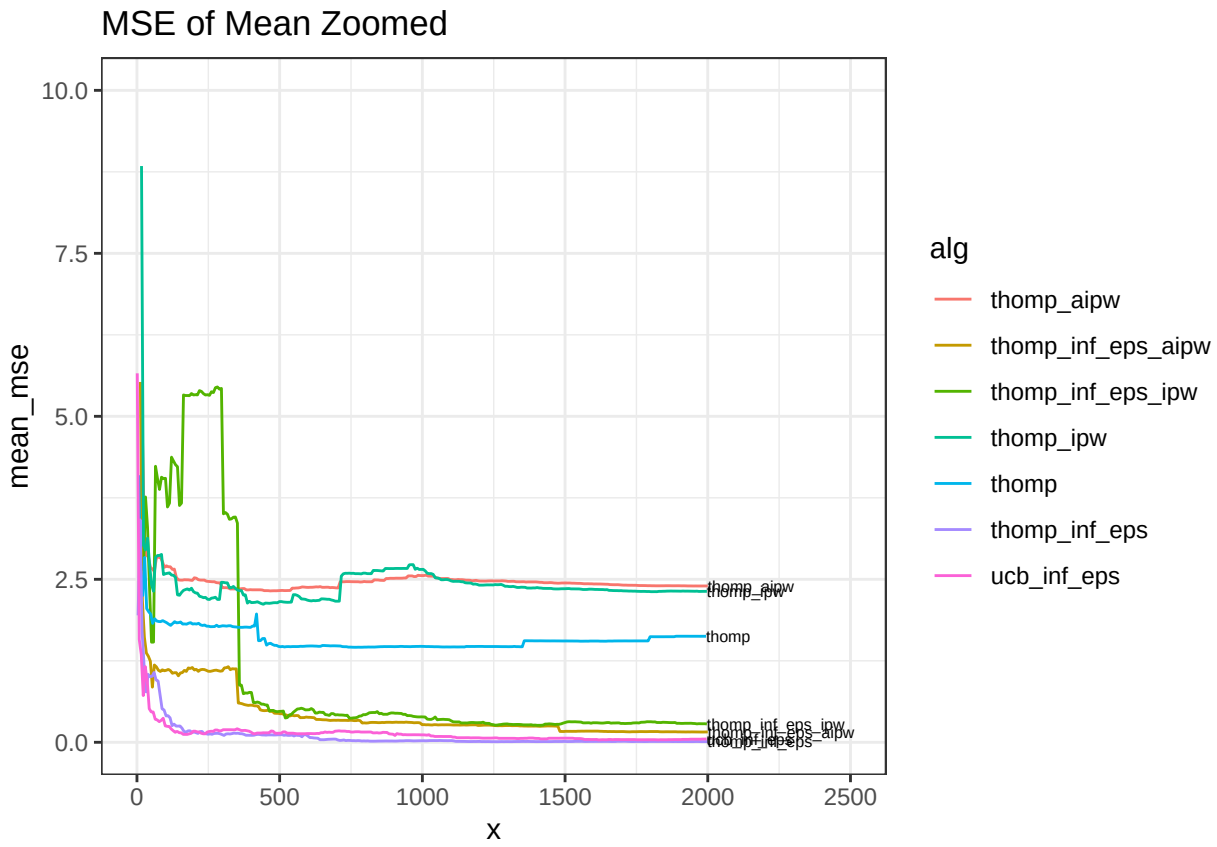
Also, the MSE of weighed estimators seems to be *higher* than the non-weighed counterparts. This is interesting.

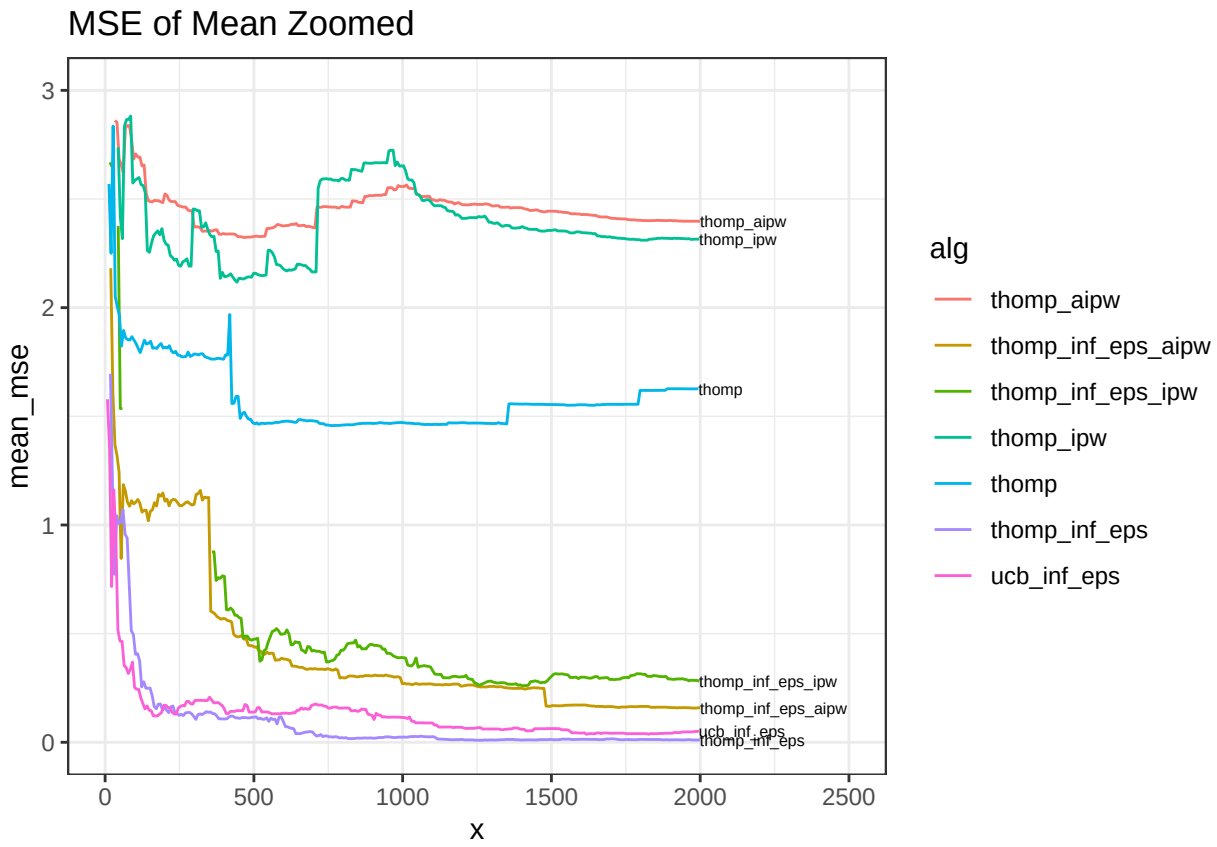
The main property of these algorithms is that their estimators are unbiased (in expectation). But, for one single iteration, the MSE seems to be worse than un-weighed counterpart.

Compare MSE of Weighed estimators with UCB_INF_EPS.

We also want to see how the weighed estimators perform wrt original MAB algorithms and their INF versions.


MSE of Mean
mean_mse
1000
750
500
250
0
0
500
1000
1500
2000
x
alg
thomp_aipw
thomp_inf_eps_aipw
thomp_inf_eps_ipw
thomp_ipw
thomp
thomp_inf_eps
ucb_inf_eps

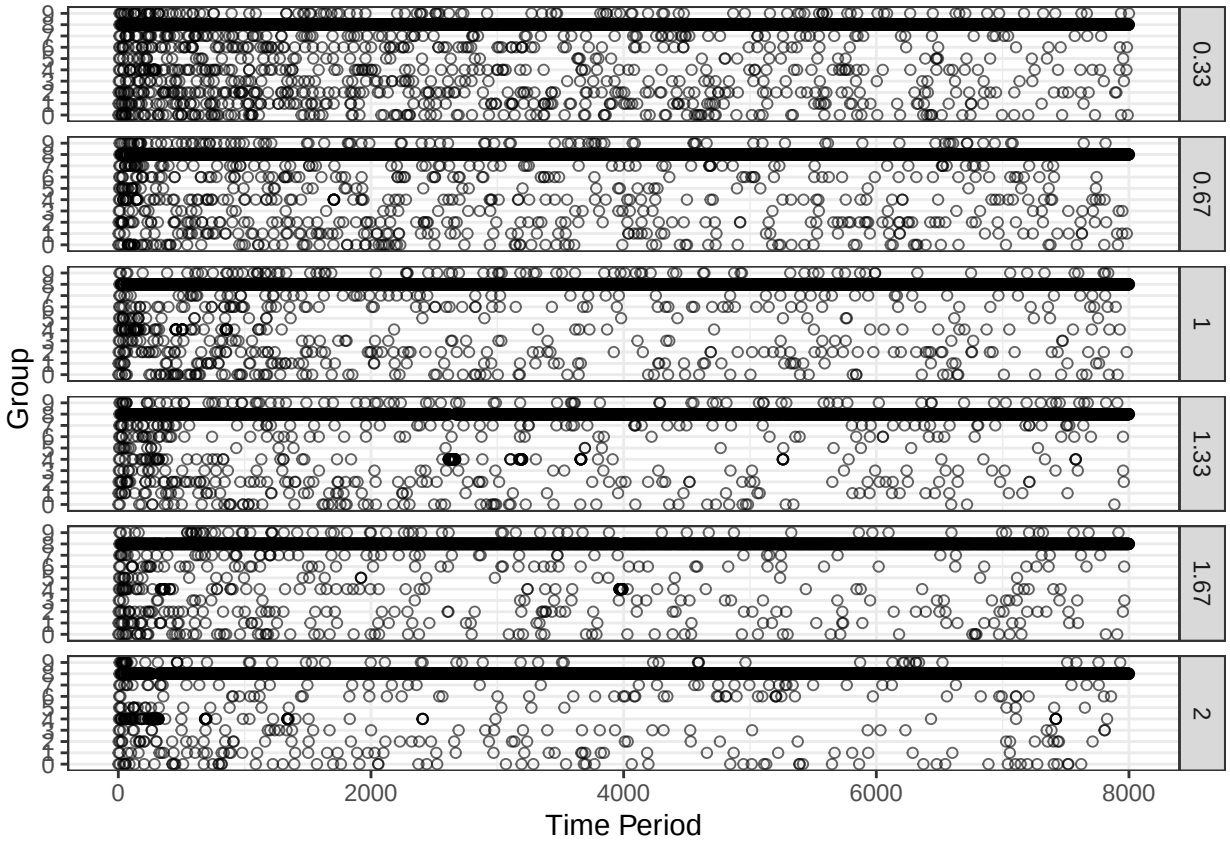

MSE of Mean Zoomed
mean_mse
10.0
7.5
5.0
2.5
0.0
0
500
1000
1500
2000
2500
x
thomp_aipw
thomp_ipw
thomp
thomp_inf_eps_ipw
thomp_inf_eps_aipw
thomp_inf_eps
alg
thomp_aipw
thomp_inf_eps_aipw
thomp_inf_eps_ipw
thomp_ipw
thomp
thomp_inf_eps
ucb_inf_eps

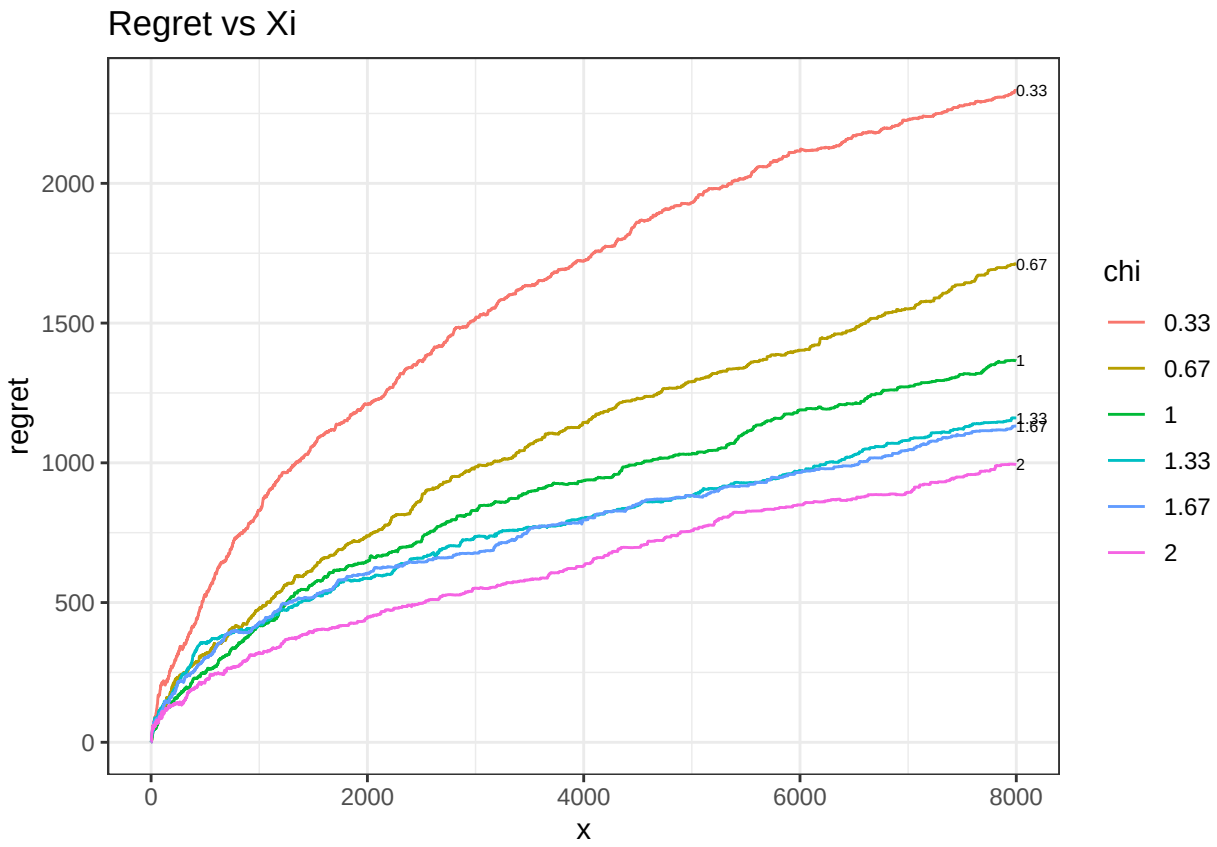


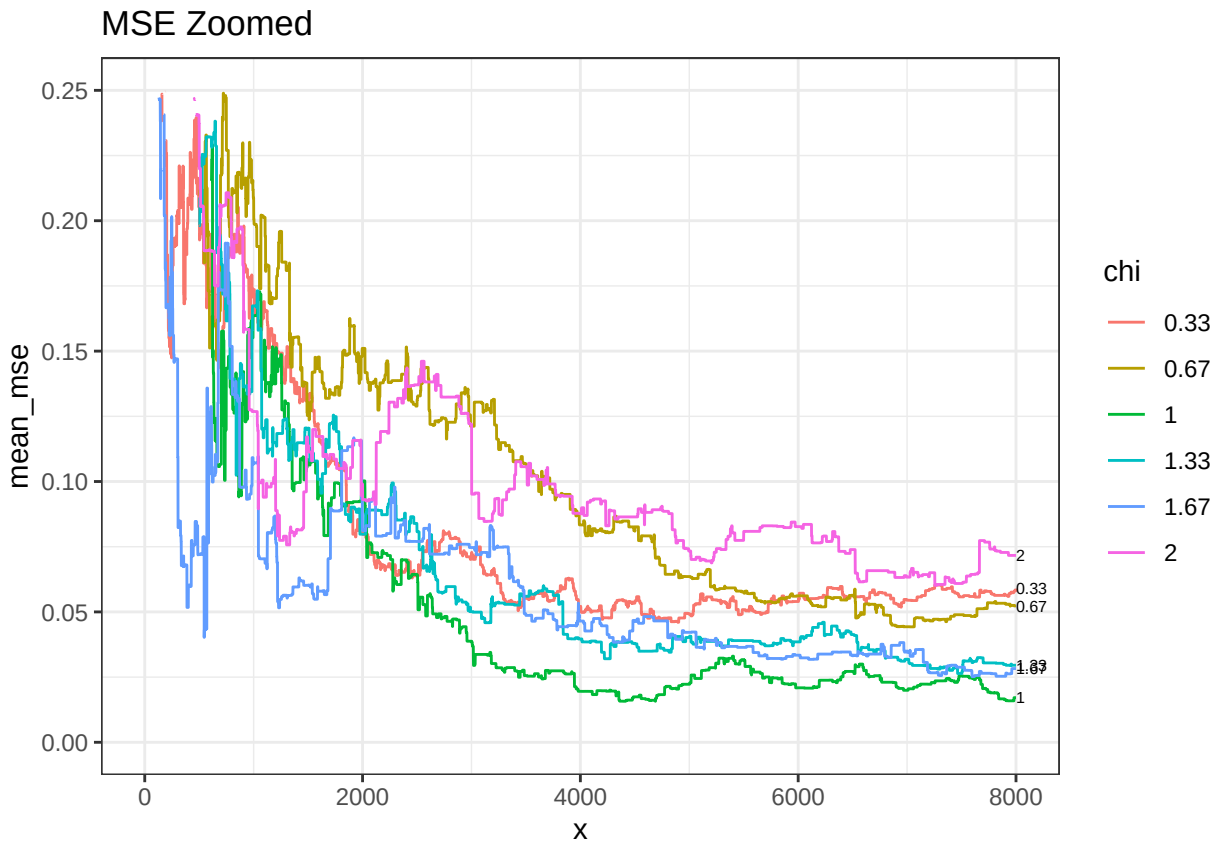
UCB_INF_EPS, THOMP_INF_EPS performs better than even Weighed algorithms.

ξ simulation for ucb_inf_eps

$\eta = \frac{\sum \frac{\sigma^2}{n}}{\xi K}$ $\epsilon_n = \frac{\eta}{1+\eta}$


Group
0.33
0.67
1
1.33
1.67
2
0
2000
4000
6000
8000
Time Period

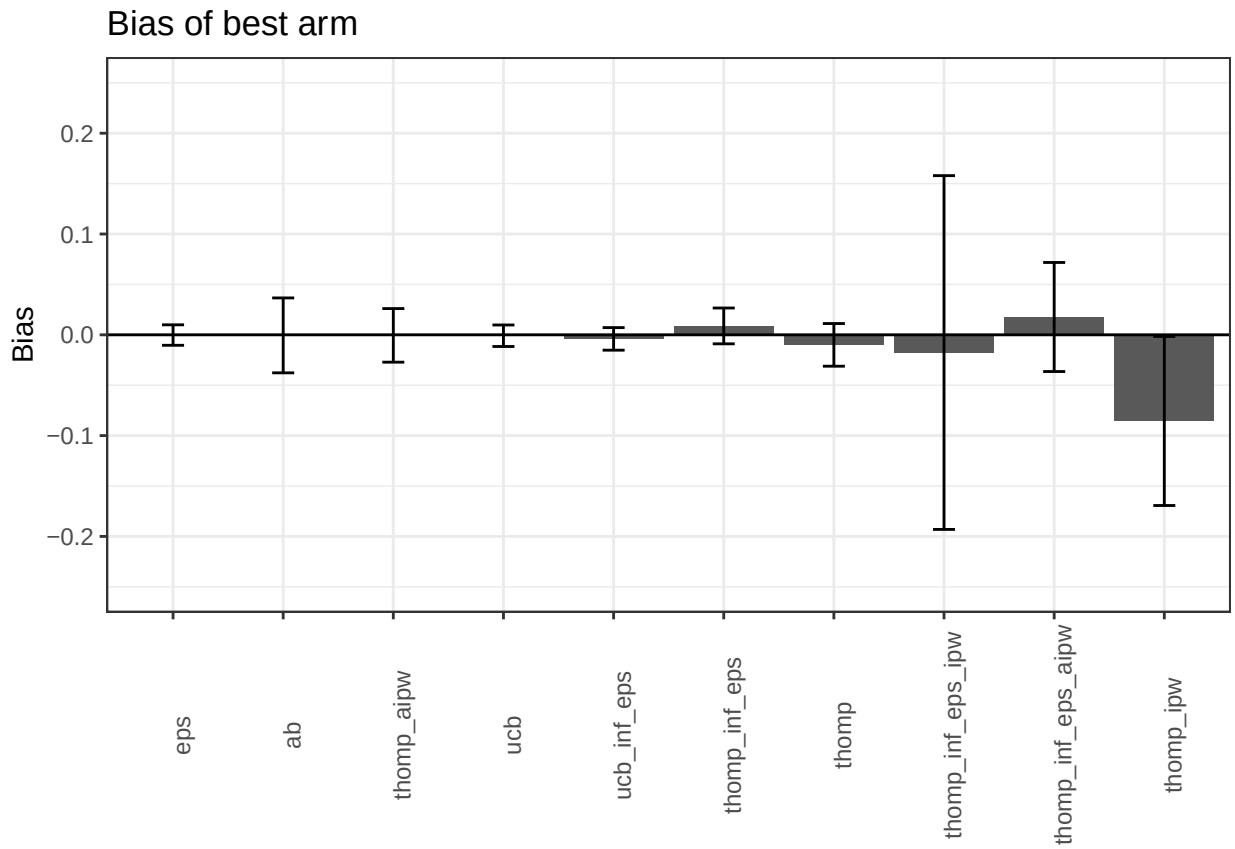

Regret vs Xi
regret
x
0
500
1000
1500
2000
0
2000
4000
6000
8000
0.33
0.67
1
1.33
1.67
2
chi
0.33
0.67
1
1.33
1.67
2



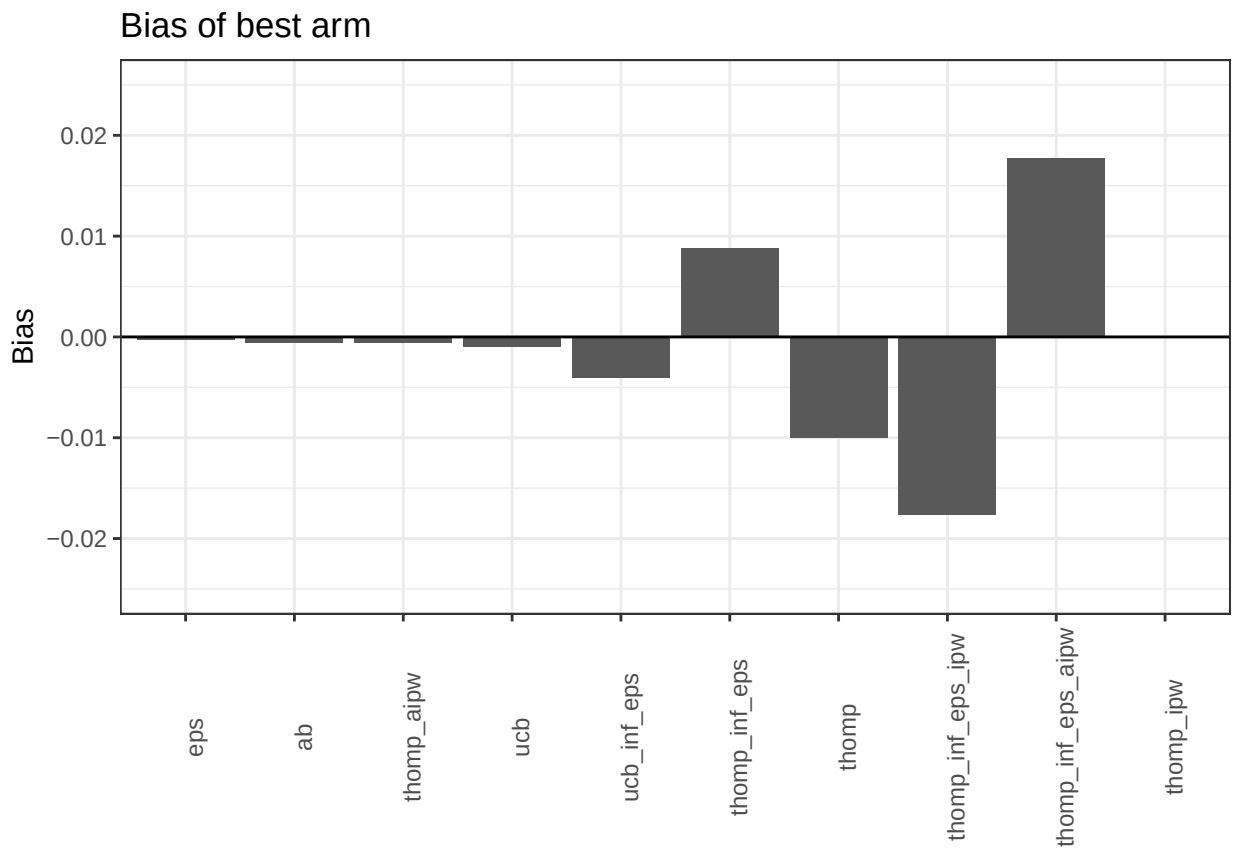
ξ serves as a knob for practitioner to control the amount of exploration in INF_EPS. ξ directly effects ϵ which in turn effects the amount of exploration. Higher the value of ξ, lower the amount of exploration.

Small sample properties of algortihms

Best arm bias

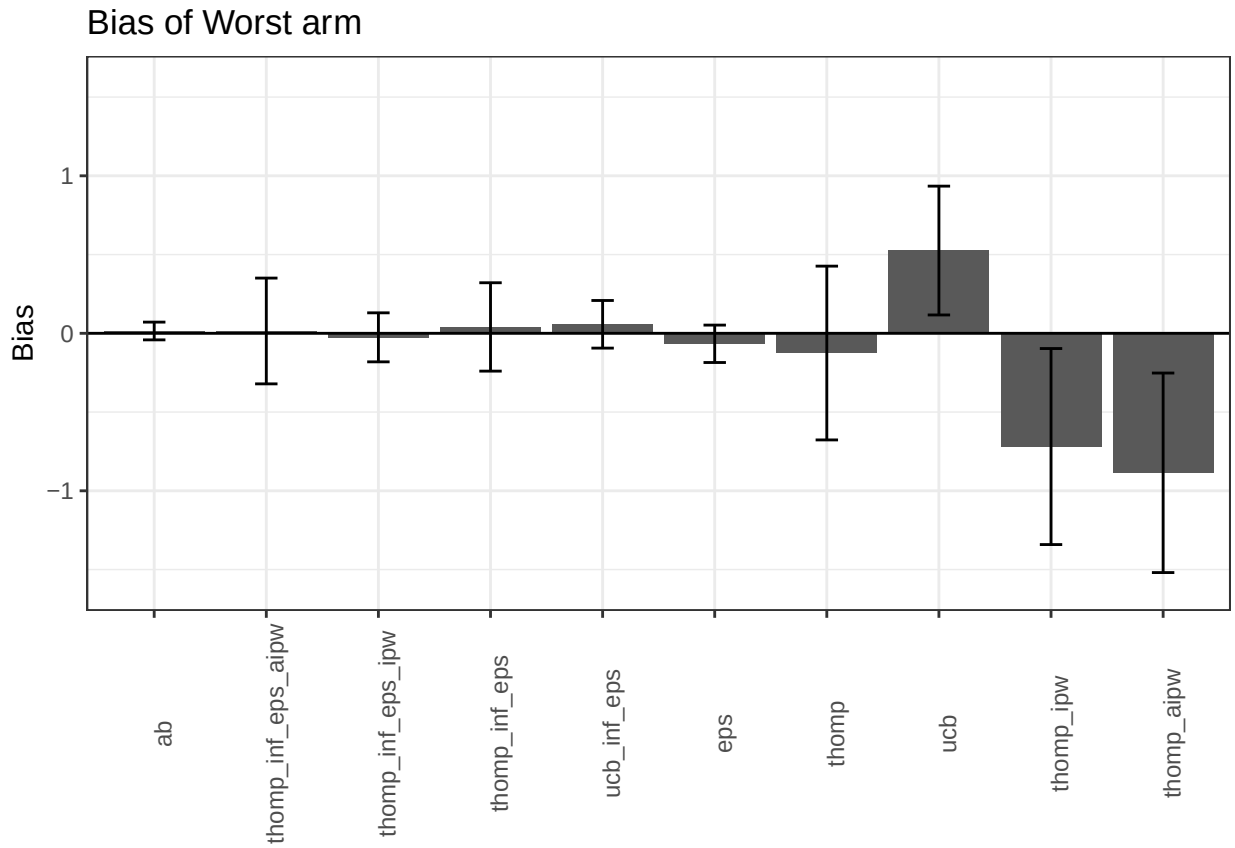


Weighed estimators have the highest bias, even for the best arm. This might be because of their high variance properties. As we only have 20 iterations, the bias might not be still converging to 0.



Although, looking at the relative values, might be deceiving. As the highest value of bias in 0.017, which is 0.3% of the actual value.

Worst arm bias



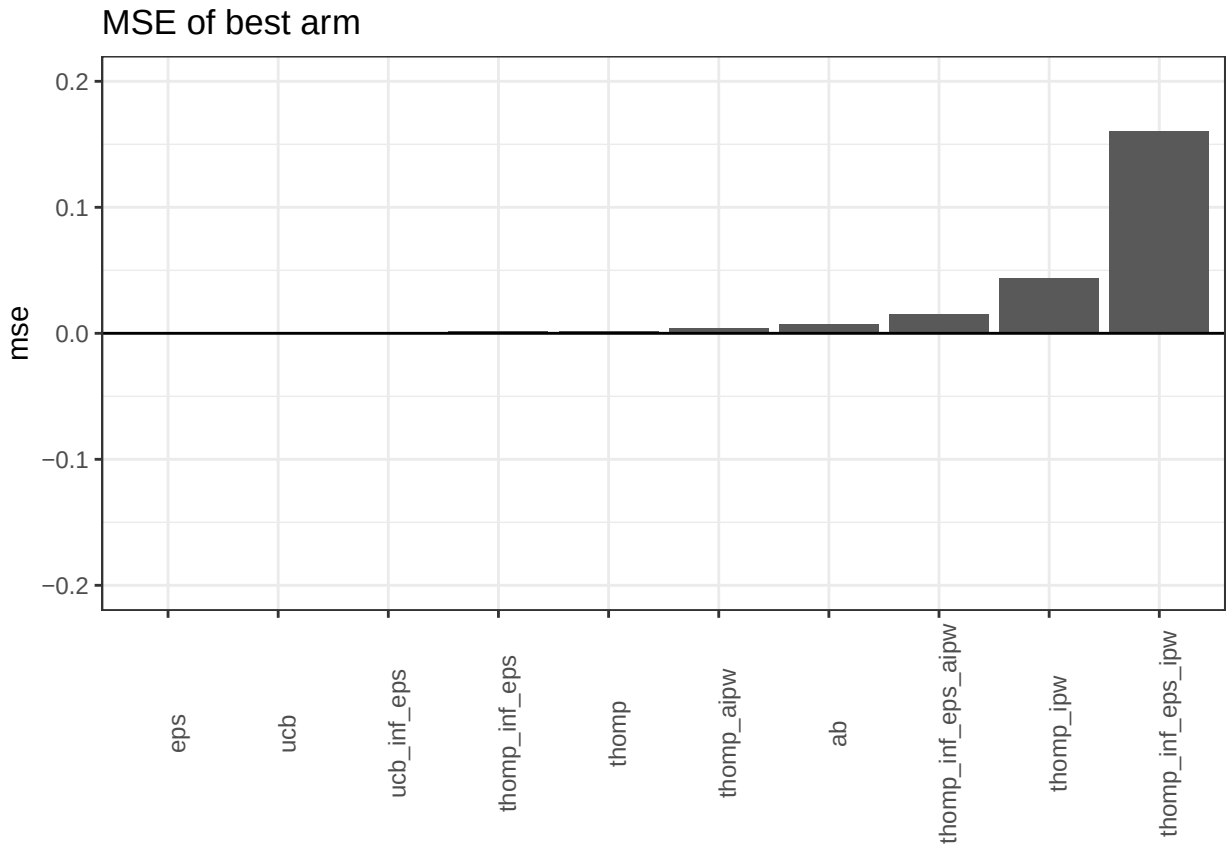
The relative value of bias is high for the worst arm.

One thing to note is that, for thompson sampling based algorithms, we have to choose priors and update posteriors based on rewards, This is problem for worst arm. If we choose bad priors and they barely get allocations, then Bias and MSE will be high, even after using weighed estimators

This makes thompson sampling even with weighed estimators, undesirable for practical use of inference.

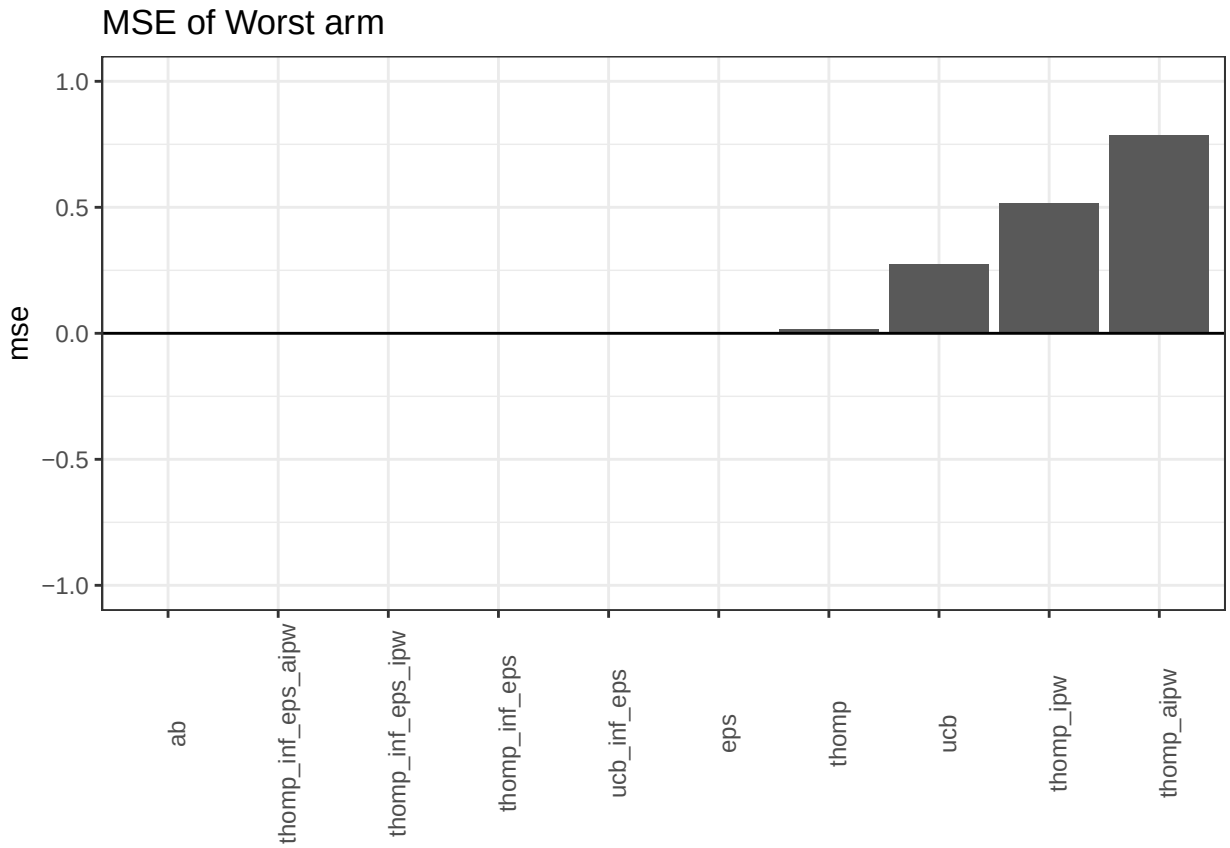
However, for INF_EPS based algorithms, this seem to correct with time as more allocations come from Inference allocations.

MSE of Best arm

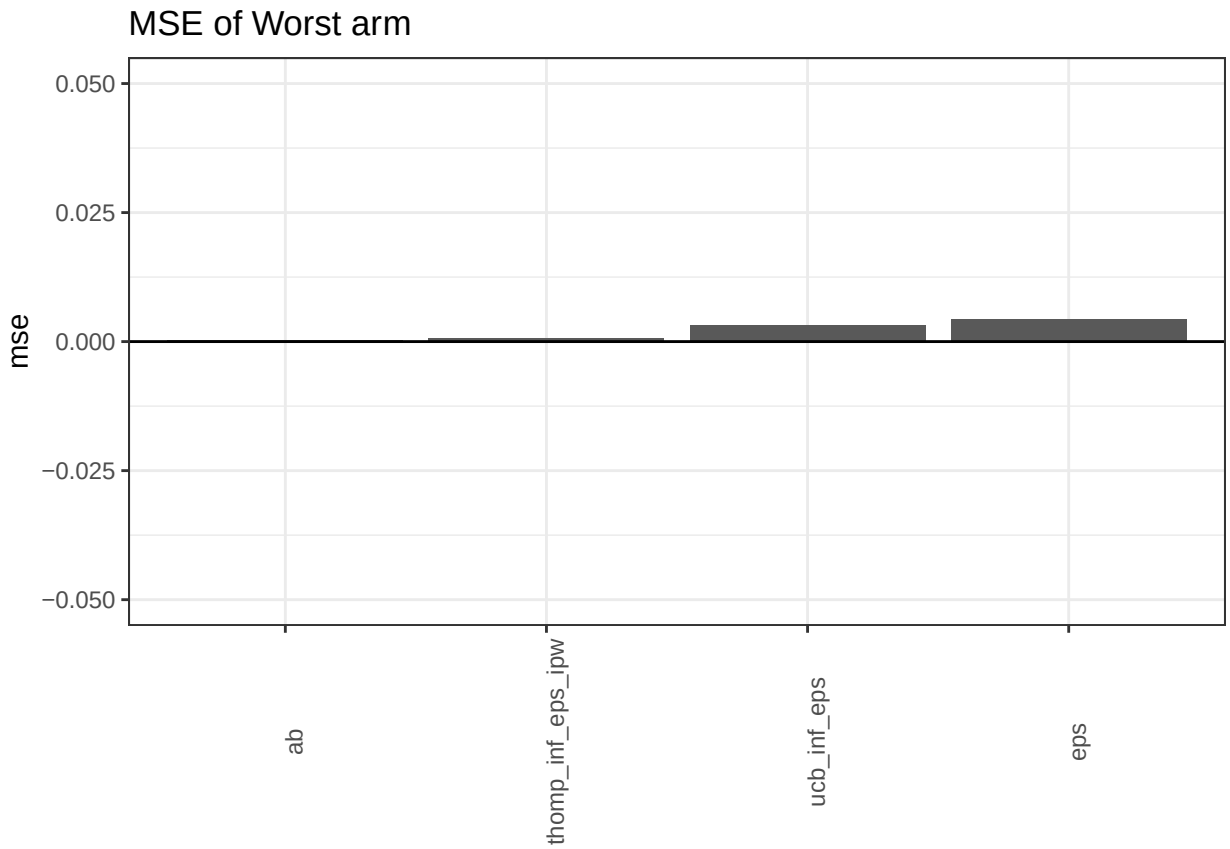


MSE is very very low for all algoritms in terms of scale, the only high values seem to be for weighed estimators.

MSE of Worst arm



Again, for vanilla MAB algorithms and their weighes estimators, MSE is high. This seem to reduce for INF_EPS algorithms.



Conclusions

- UCB_INF_EPS is clear winner for use cases where there is relative importance for both efficiency and Inference.
- Weighed Estimators of existing MAB algorithms may have nice properties *in expectation* but for practical purposes when the practitioner has one shot, UCB_INF_EPS will outperform in regret minimization and estimation.
- This problem is exacerbated if bad priors are chosen for Thompson Sampling based algorithms.