Experiment 1

February 16, 2018

Simulation Details

Considered $K = 3$, $T = 5005$, $N = 125$. Report statistics at $t = 1000, 3000, 5000$
The Bandit priors that were considered:

- Uniform: Draw the mean rewards for the arms from [0.25, 0.75]
- "HeavyTail": We took the mean rewards to be randomly drawn from Beta($\alpha = 0.6, \beta = 0.6$). With this distribution it was likely to have arms that were at the extremes (close to 1 and close to 0) but also some of the arms with intermediate value means.
- Needle-in-haystack
    1. Medium - 9 arms with mean 0.50, 1 arm with mean 0.55 (+ 0.05)
    2. High - 9 arms with mean 0.50, 1 arm with mean 0.70 (+ 0.20)

Algorithms considered:

1. ThompsonSampling with priors of $Beta(1, 1)$ for every arm.
2. DynamicGreedy with priors of $Beta(1, 1)$ for every arm
3. Bayesian Dynamic ϵ-greedy with priors of $Beta(1, 1)$ for every arm and $\epsilon = 0.05$

Agent Algorithms considered:

1. HardMax
2. HardMaxWithRandom
3. SoftMax

Memory Sizes

1. 10
2. 25
3. 100

Simulation Procedure

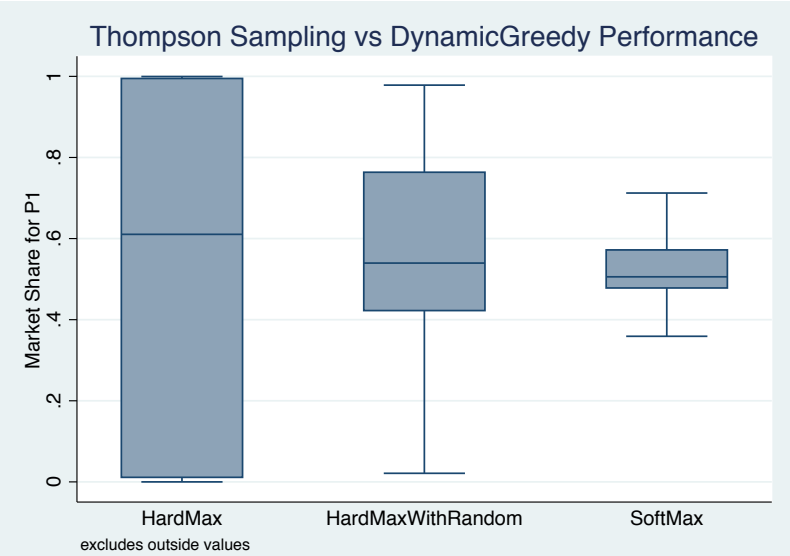
```
for Each prior p do
    for Each agent algorithm agentalg do
        for Each principal algorithm pair principalalg1, principalalg2 do
            for N simulations do
                Generate true distribution from p (except for needle-in-haystack, just use p itself)
                Give the agents 5 observations from each principal
                Run simulation for T periods
            end for
        end for
    end for
end for
```

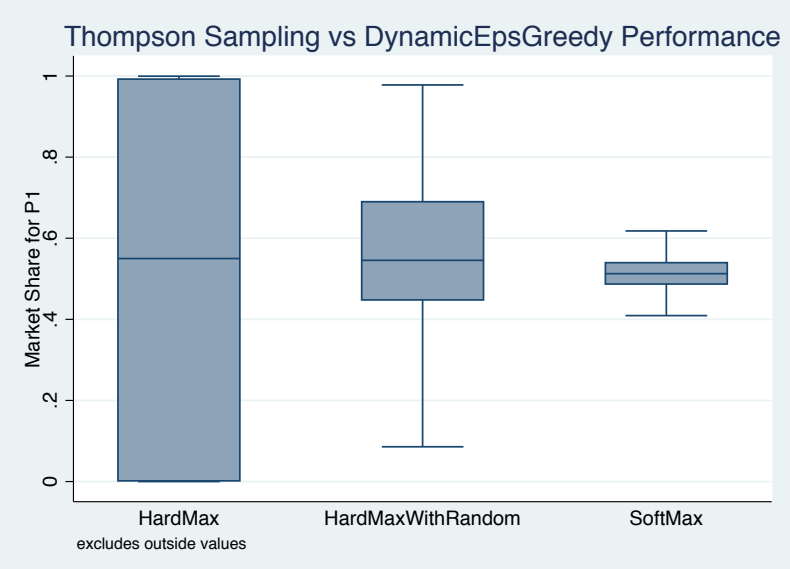
Results

One thing which is ambiguous to define is the regret value to use when a principal never gets chosen in a given simulation. When calculating any of the aggregate regret statistics we drop these simulations, but we do record how many rounds have an undefined regret.

First, we'll restrict focus to $t = 5000$ and look at the performance of ThompsonSampling. Note that the y axis here represents the market share that the ThompsonSampling principal gets.

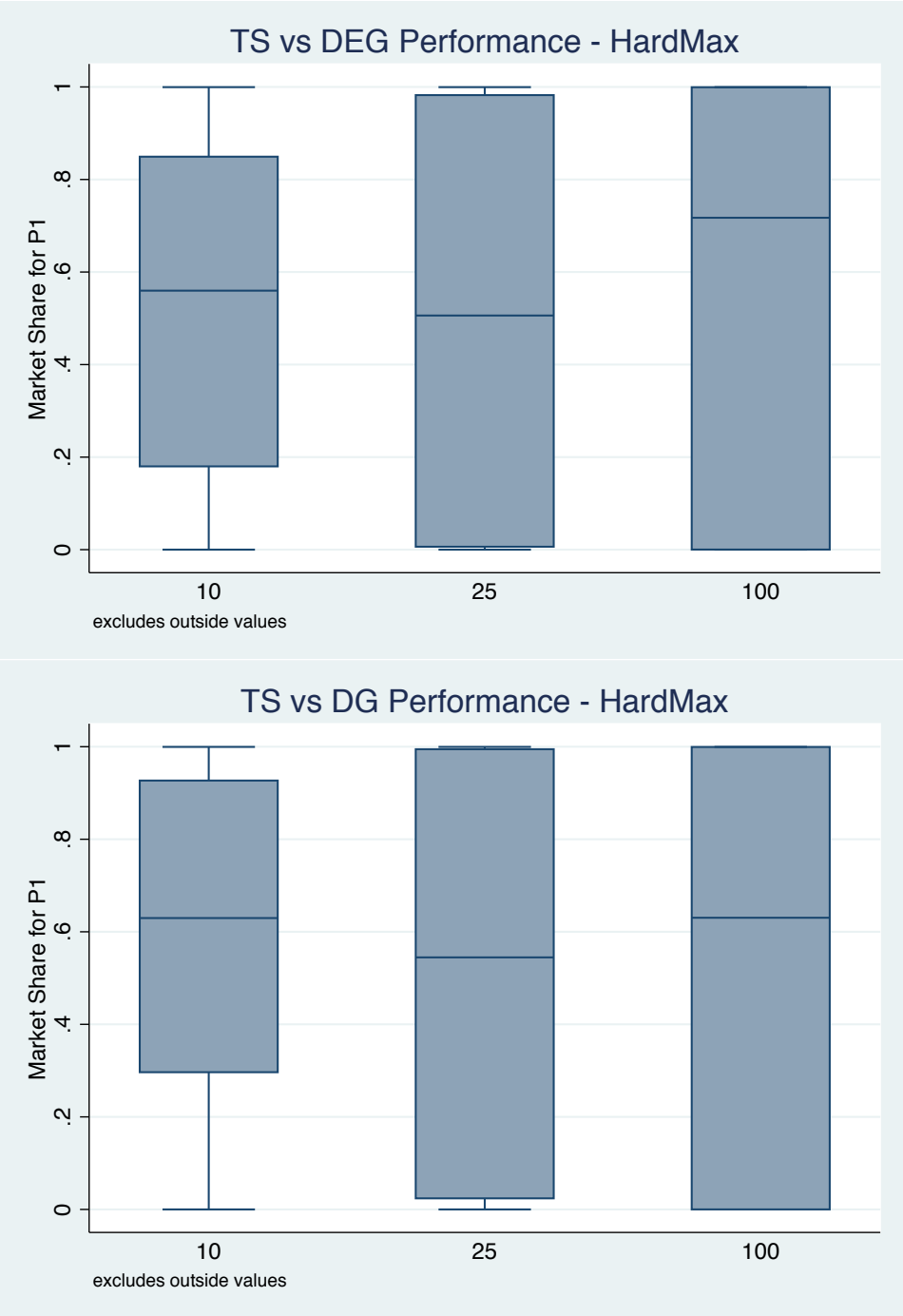
Performance of ThompsonSampling vs DynamicGreedy and DynamicEpsilonGreedy across all agent models, memory sizes, and priors:





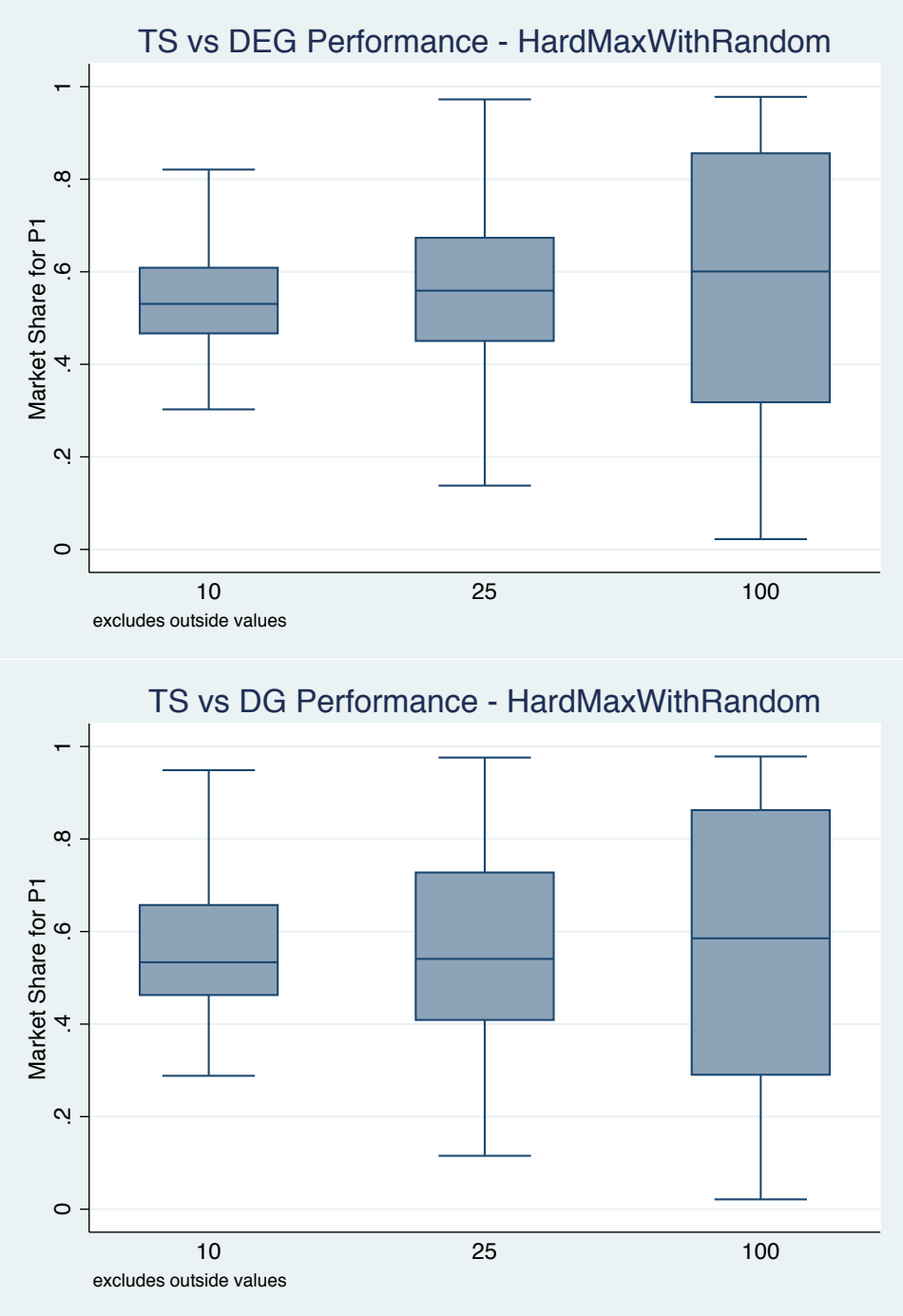
Now, looking across different memory sizes for each agent model (still using each prior):

HardMax



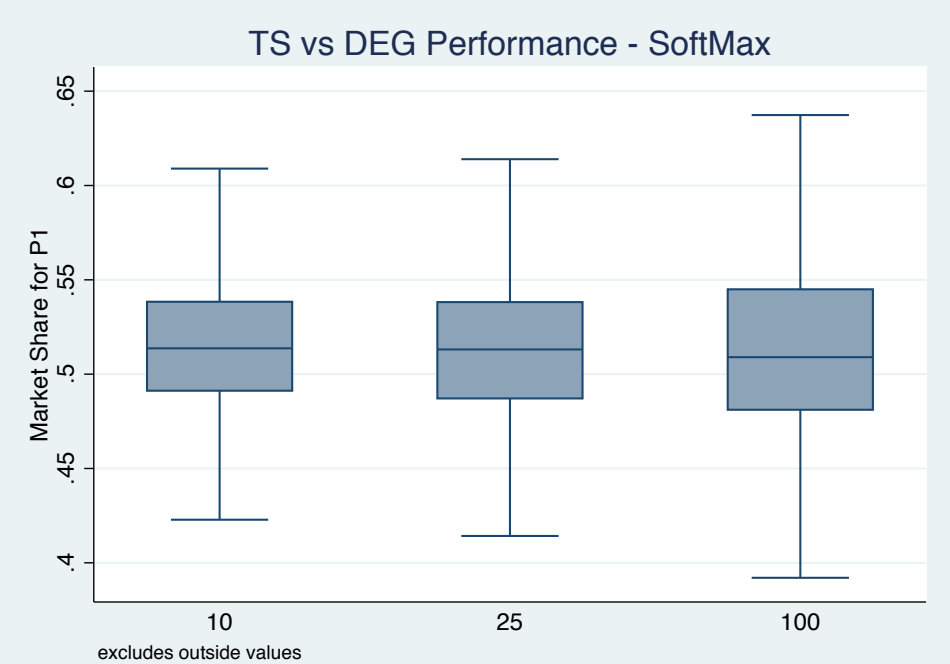
Under HardMax it's clear that there is high variability in market share that is increasing in memory size and indicative of less competition but the median across memory sizes still has ThompsonSampling winning.

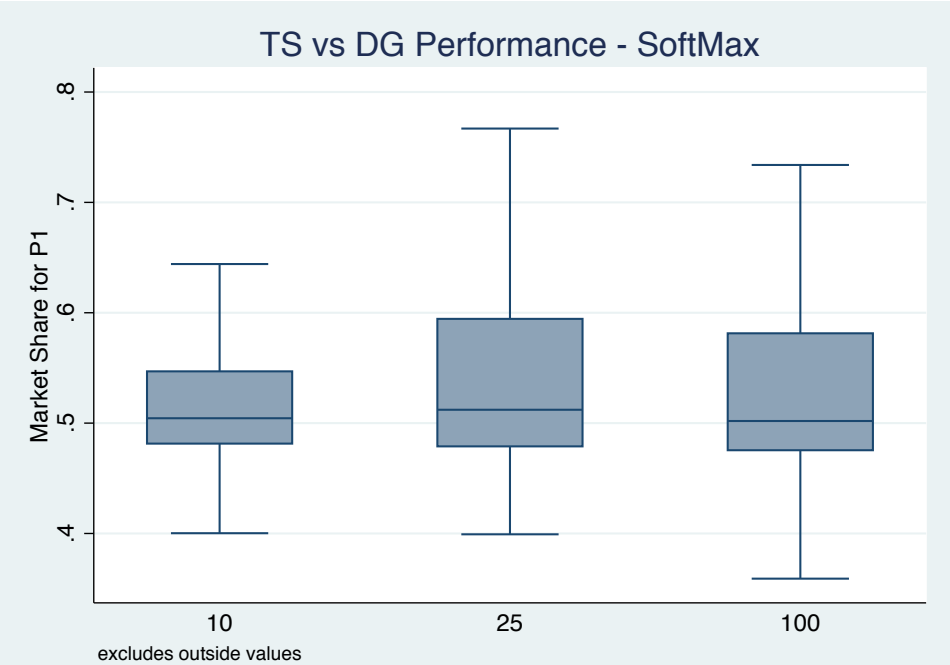
HardMaxWithRandom



Under HardMaxWithRandom, the amount of variability decreases (more competition) but is still increasing in memory size and ThompsonSampling still wins in most simulations.

SoftMax

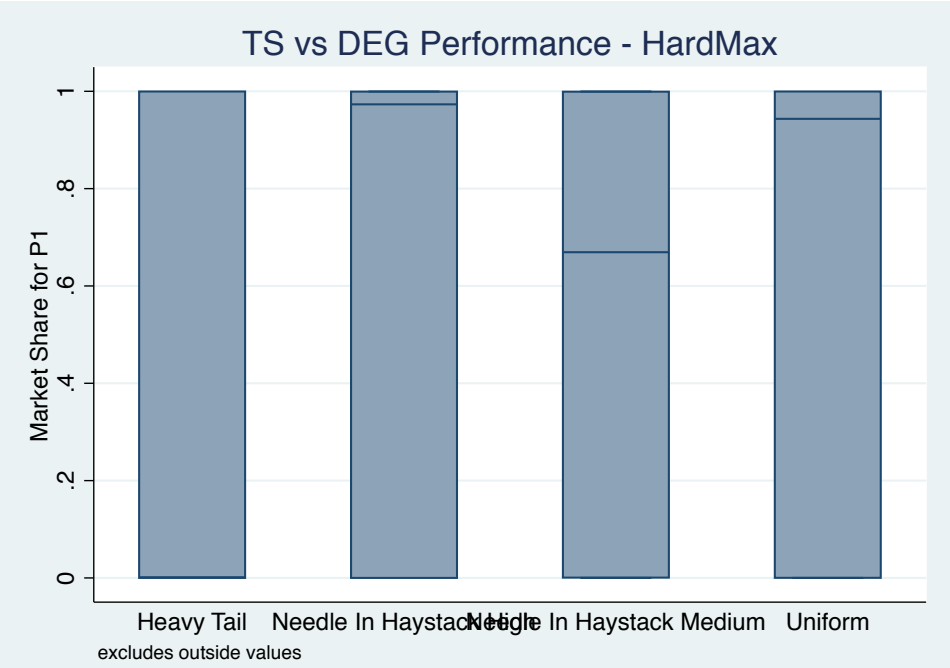


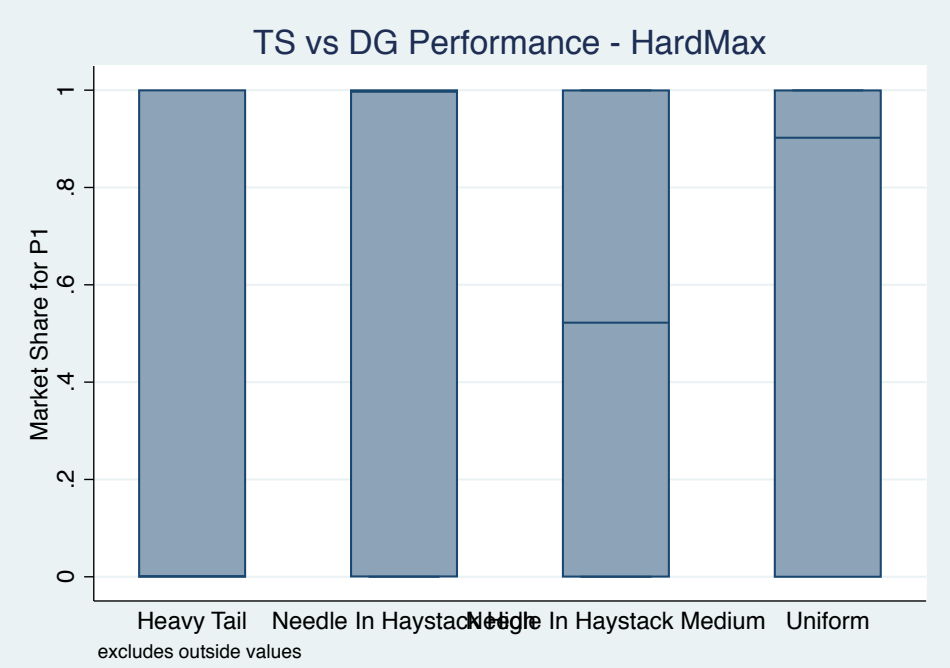


Under SoftMax there is less variability (more competition) that does not change much with memory size and here ThompsonSampling wins more in the median simulation but only a bit.

We'll now look fix memory size to be 100 and look at the performance across priors.

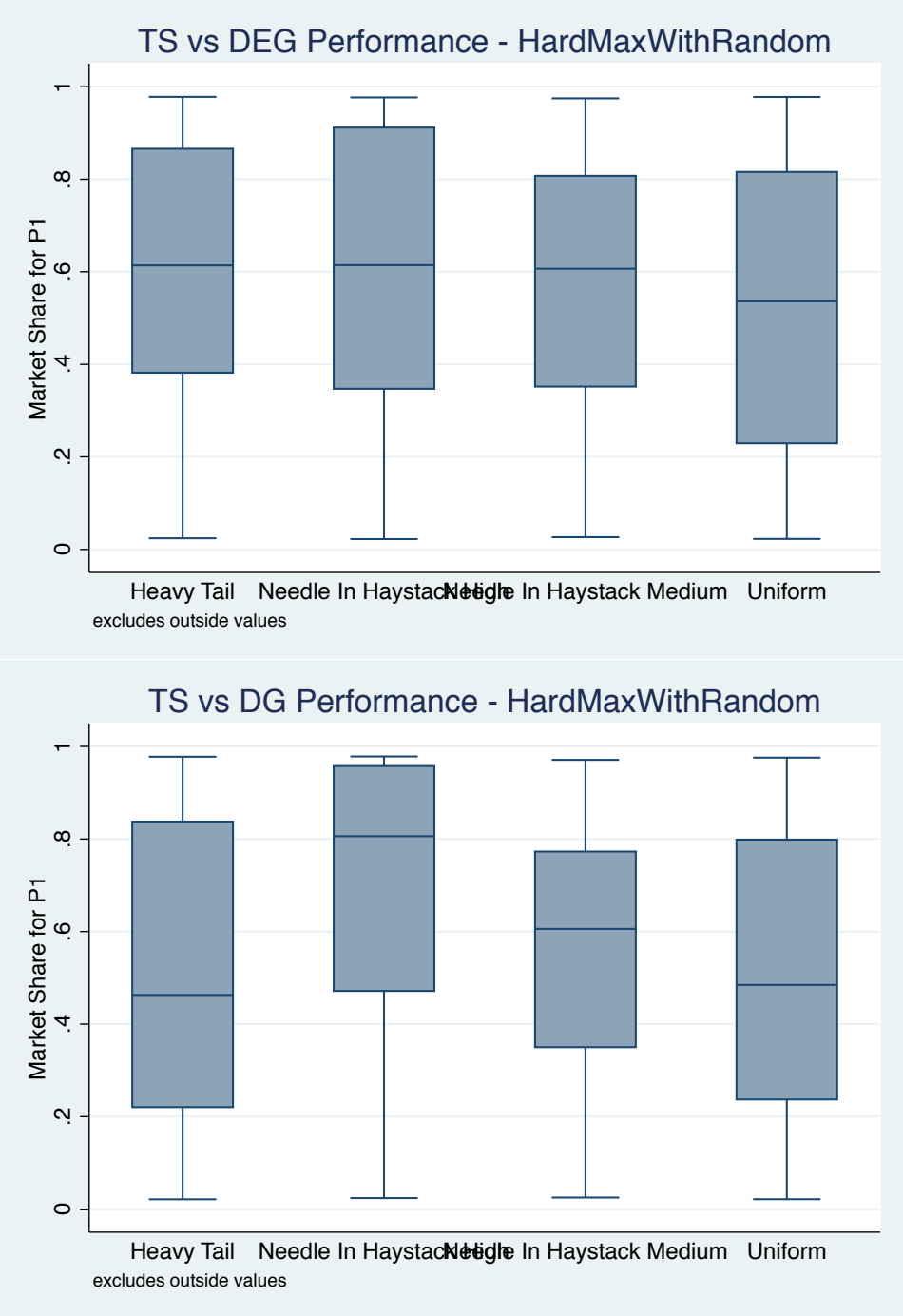
HardMax





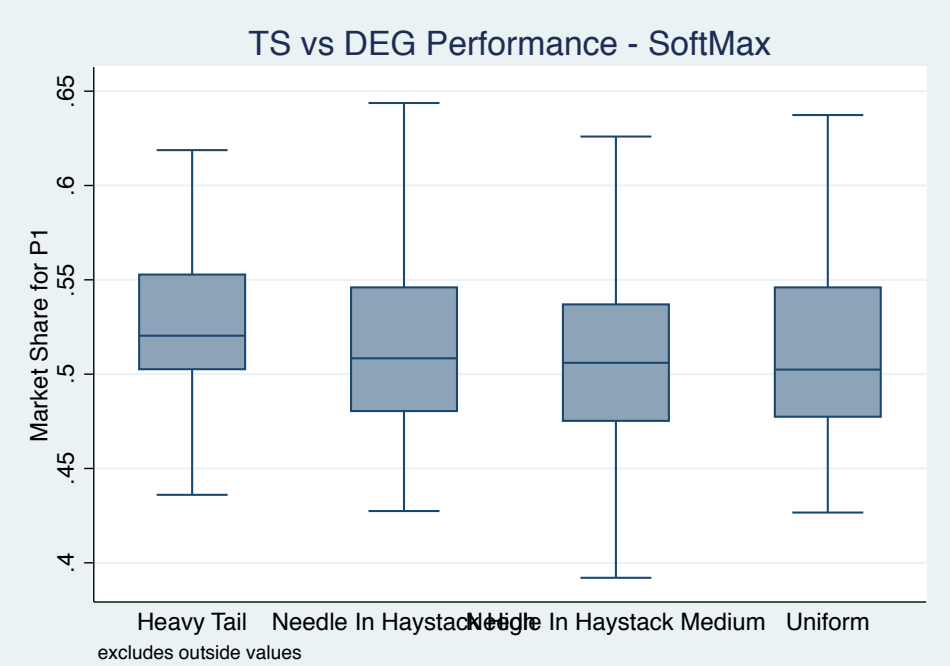
With HardMax there is significant variability (as noted before), but we have that in the needle in haystack medium case the median is approximately 0.5 while in Uniform + Needle in Haystack High the median is 1 and heavy tail it is almost 0. I don't know if this is something interesting or just noise due to the variability in HardMax. Will have to look into this more.

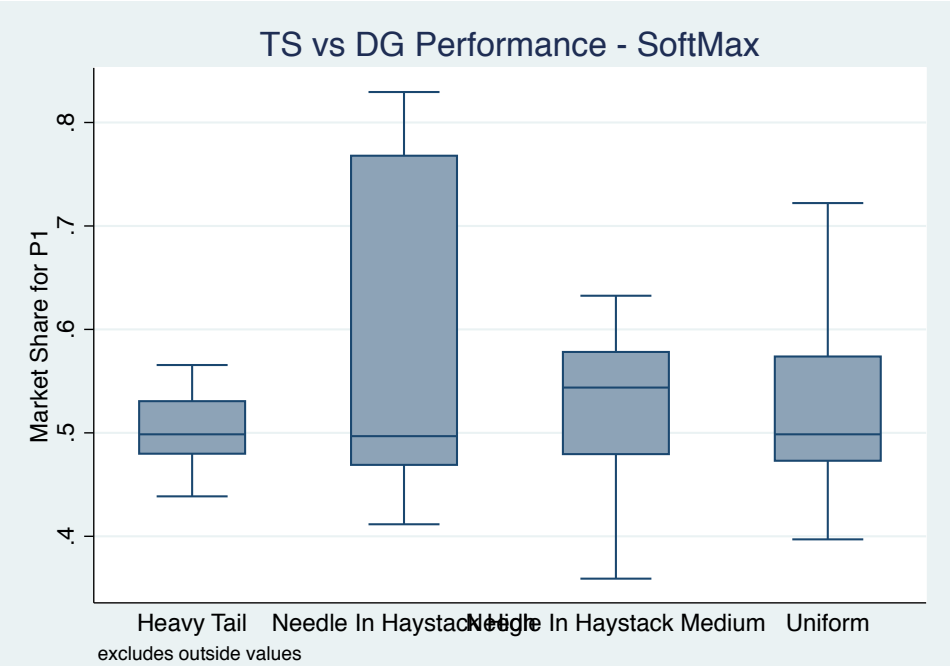
HardMaxWithRandom



With TS vs DEG we see roughly the same performance across priors, but for TS vs DG we see as with HardMax that in the needle in haystack high instance we have that TS wins by a lot but not so much in the heavy tail case. This makes sense as TS (and DEG) should generally find the "good" arm while DG may sometimes wrongly identify the best arm, leading to this disparity.

SoftMax





Nothing horribly interesting here besides noting that the same TS vs DG difference appears in the Needle in haystack high prior.

Now, a couple of plots giving a sense of regret values for ThompsonSampling across different agent algorithms and memory values.