Lightricks

Large Scale A/B Testing

Daniel Benzaquen

Overview

- A/B Testing - Introduction
- The problem of multiplicity
- The Multi-Armed Bandit Approach
- Generalized Weighted Thompson Sampling
- Experiments
- Conclusion


All Pro Tools
In All Enlight Apps
All Enlight Apps
12
Months
$49.99
1
Month
$9.99
Restore Purchases
Recurring billing. Cancel anytime.
*Billed in one payment of $49.99
Payment will be charged to your iTunes account at confirmation of purchase.
Subscriptions will automatically renew unless auto-renew is turned off at least

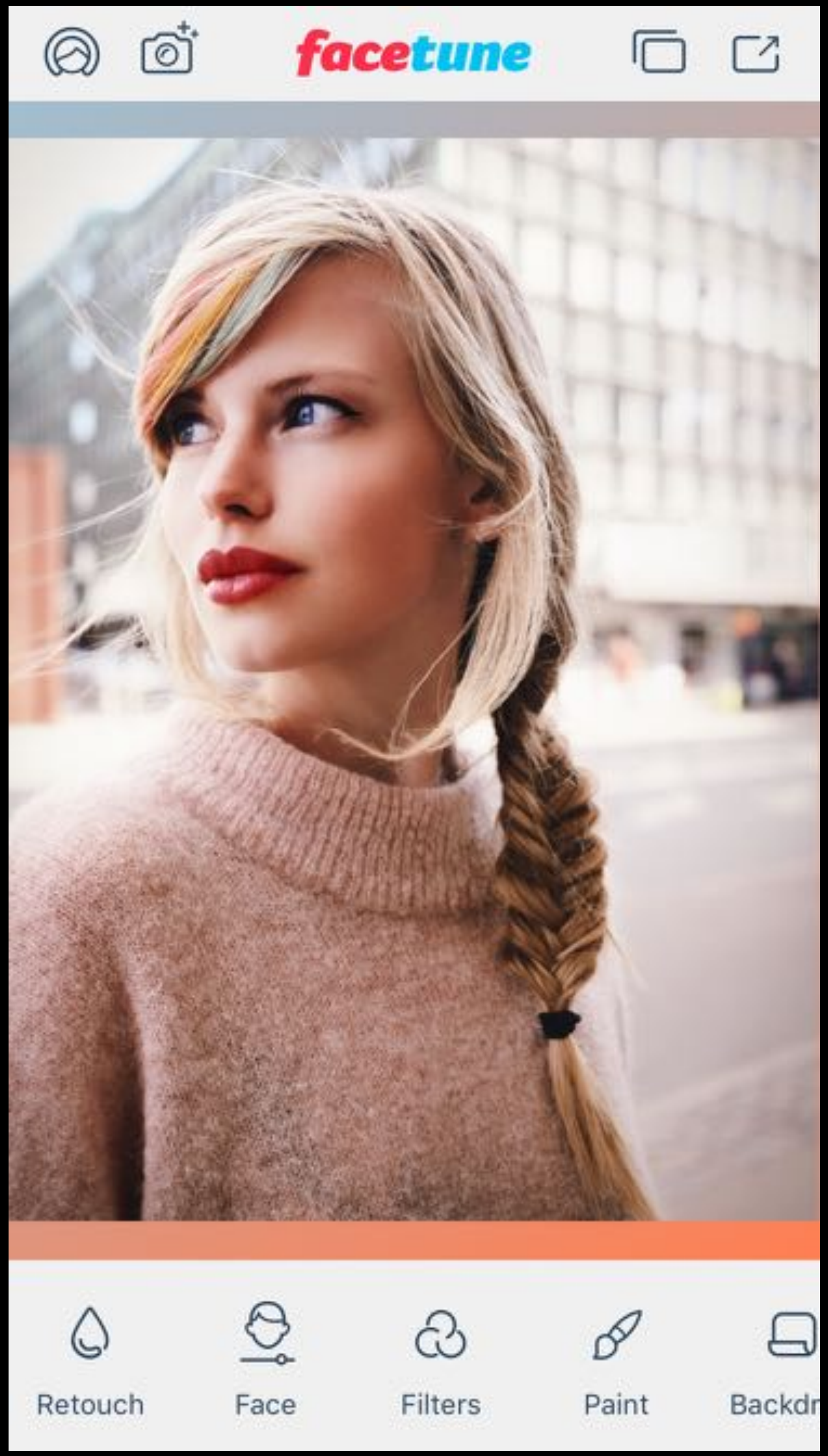
Multiple Testing

Multiple Testing

A/B Testing

- A/B Testing is a popular procedure for comparing two versions of a single variable.
- Tests with straightforward procedures are available for this kind of proportion comparison.

vs



Lightricks

Multiple Testing

A/B Testing

VS

VS

Multiple Testing

Motivation

* What if the company wants to release an application that has several versions? How can the optimal one be chosen?
* An intuitive way to do it would to perform several simple procedure tests between all the versions.
* More intuitively, the more tests we perform on the same sample, the more likely we are to incorrectly decide on correlated tests.

Multiple Testing

Possible Solutions

So, how do we proceed if we want to perform multiple testing?

* Many solutions are available to this problem but they are inadequate when the number of tests is large.

Multi Armed Bandit (MAB)

MAB

Introduction

- "One-Armed-Bandit" is a reference to a slot machine.

- In the spirit of gambling, the problem is defining the set of decisions that a gambler faces in order to maximize his benefits.

MAB

Introduction

- "One-Armed-Bandit" is a reference to a slot machine.

- In the spirit of gambling, the problem is defining the set of decisions that a gambler faces in order to maximize his benefits.

MAB

Introduction

- The application has several "versions" to be proposed and we need to choose the optimal one in order to maximize revenue.
- The Thompson sampling algorithm is a heuristic for choosing actions in the multi-armed bandit (MAB) problem.
- This algorithm is based on a Bayesian inference scheme.

MAB

Bayesian Inference: Reminder

- In the Bayesian setting there are no constant parameters.
- Rather, we think of these as random variables which have distributions, as all random variables do (we call them priors).
- In this setting, inference is carried out using the posterior distribution:

$$P(\theta \mid D)$$

- By Bayes rule:

$$P(\theta \mid D) \propto P(D \mid \theta) P(\theta)$$

MAB

Thompson sampling

The Thompson-sampling algorithm is a straightforward application of Bayesian inference. We initialize the algorithm by a user given distribution, gather some data and update the given distribution using Bayes formula in order to maximize the expected reward.

MAB

Thompson sampling: Example

A famous and useful example is the Bernoulli trial when the reward is binary (1/0, buy/didn't buy, etc.).

The parameters of interest in this example are:

1

The parameter: p_i - The proportion of buyers for version i.

2

The prior distribution on the p parameter: The $Beta(\alpha = 1, \beta = 1)$ distribution, which is the standard Uniform distribution.

3

The data: Observed data of buyers/non-buyers after proposal of version i.

4

The posterior distribution: $Beta(a_{old} + \#buyers, \beta_{old} + \#non\_buyers)$

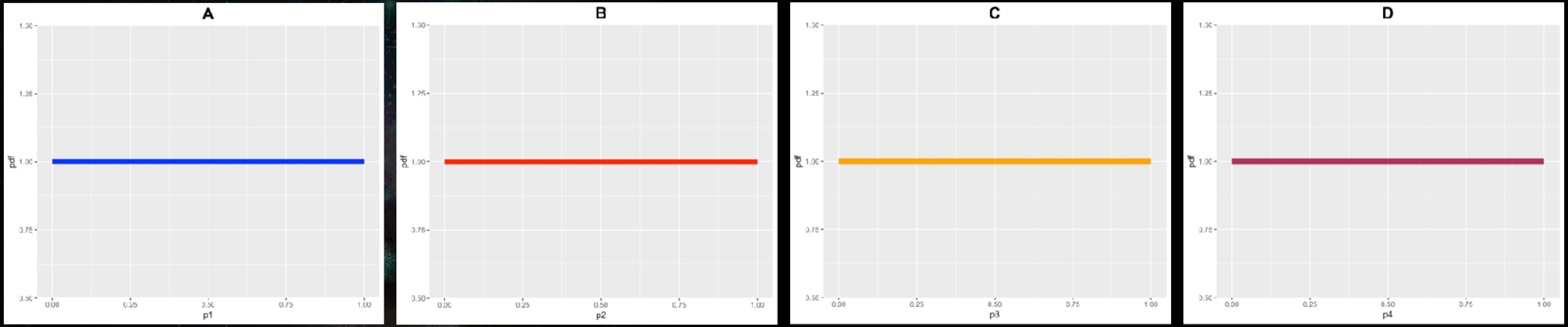
MAB

Thompson sampling: Example

Lightricks

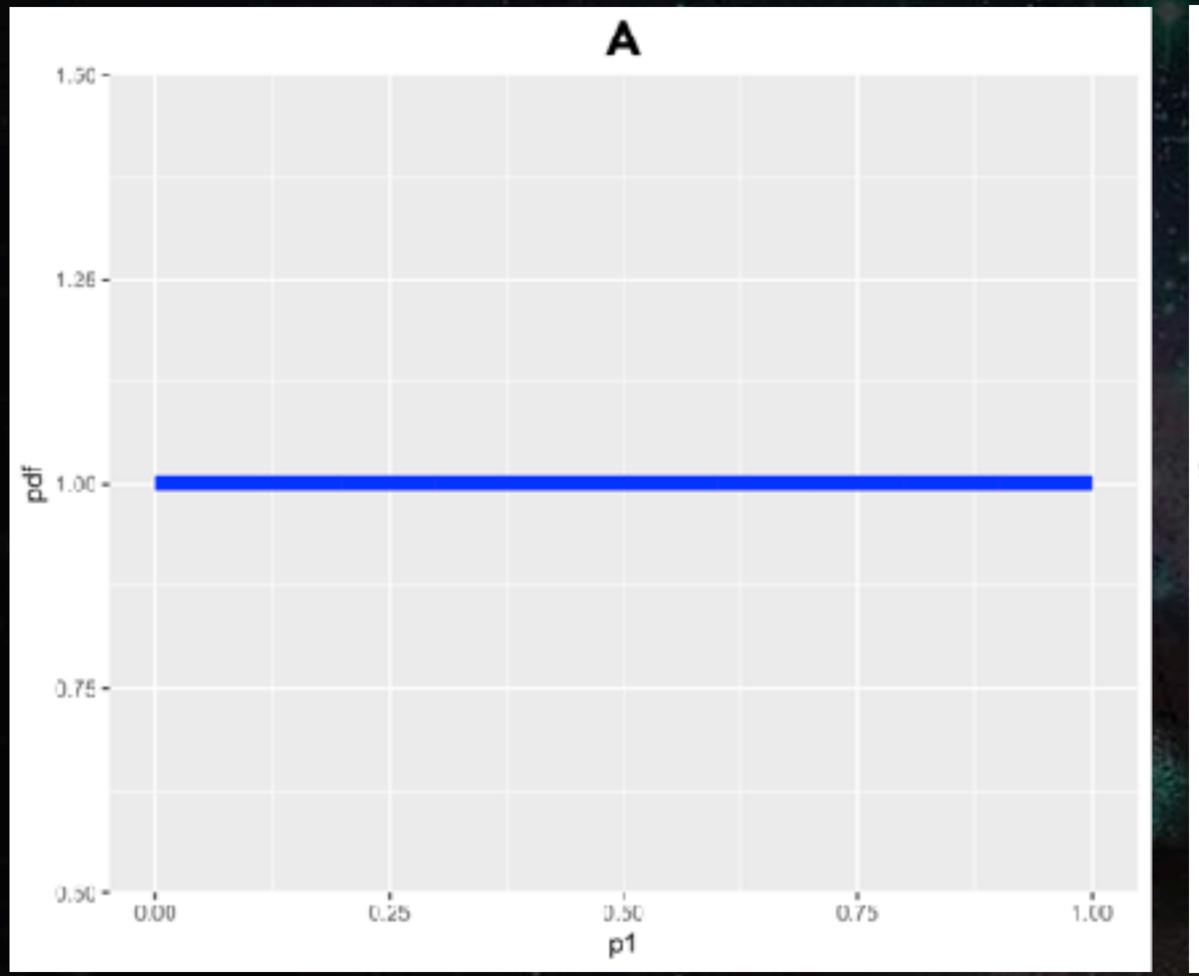
MAB

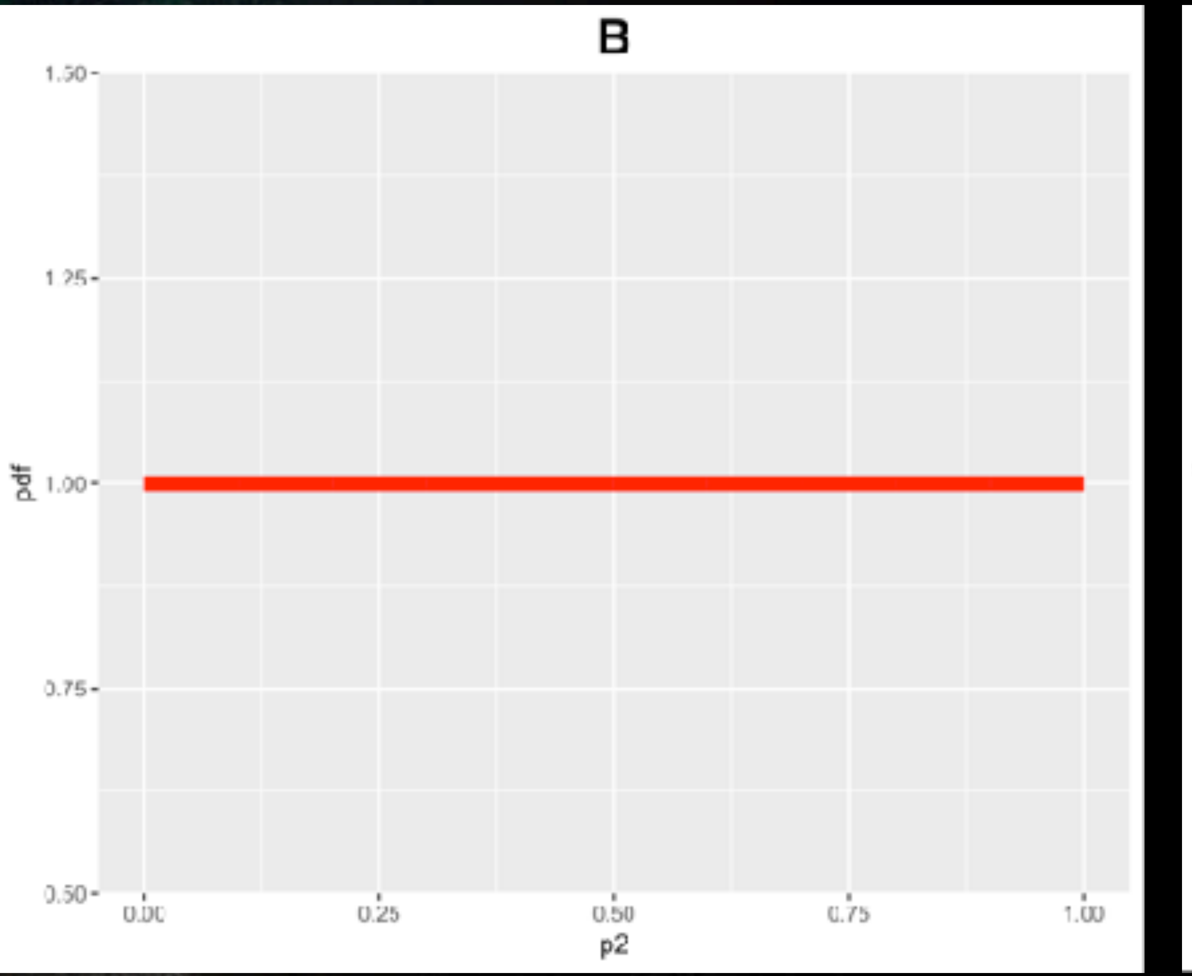
Thompson sampling: Example

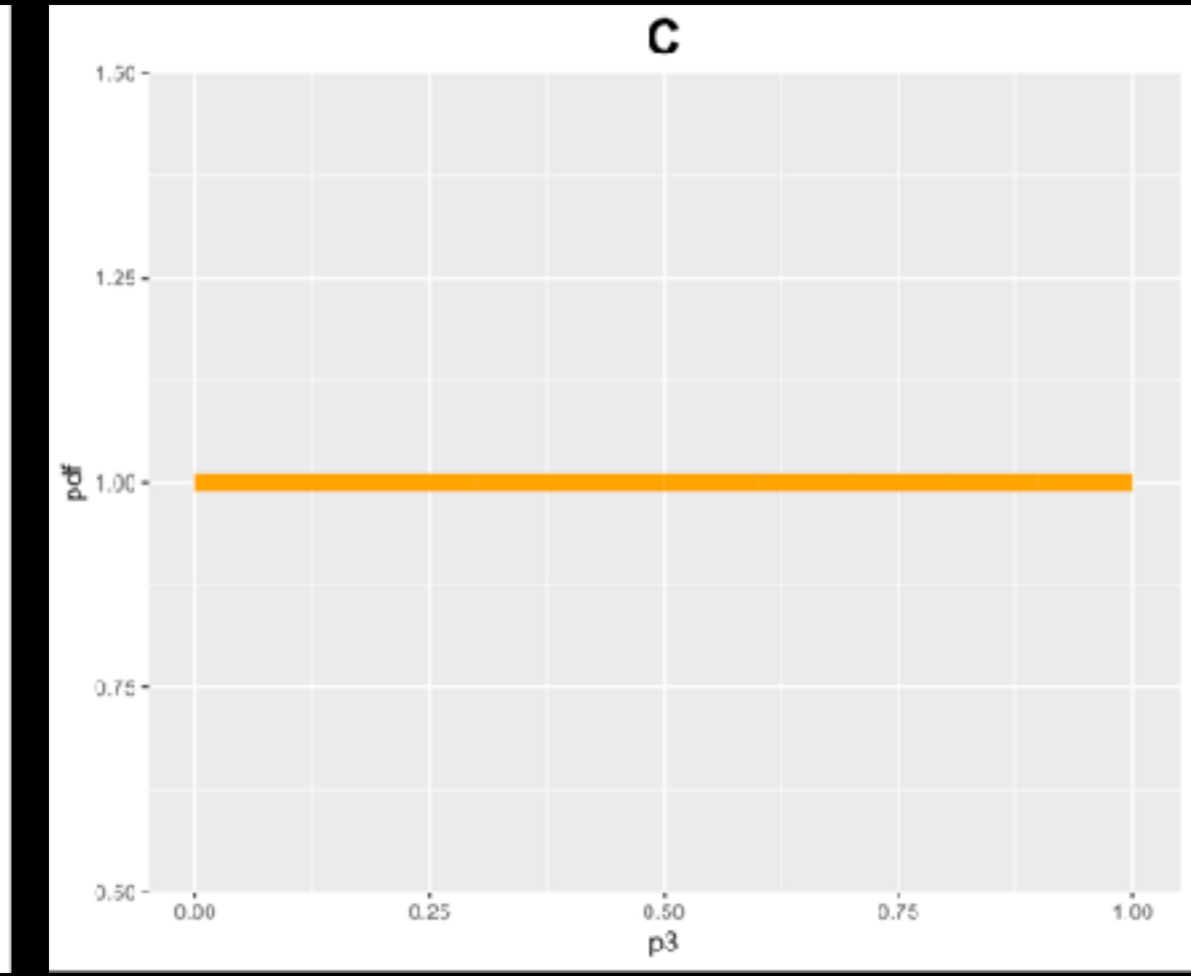


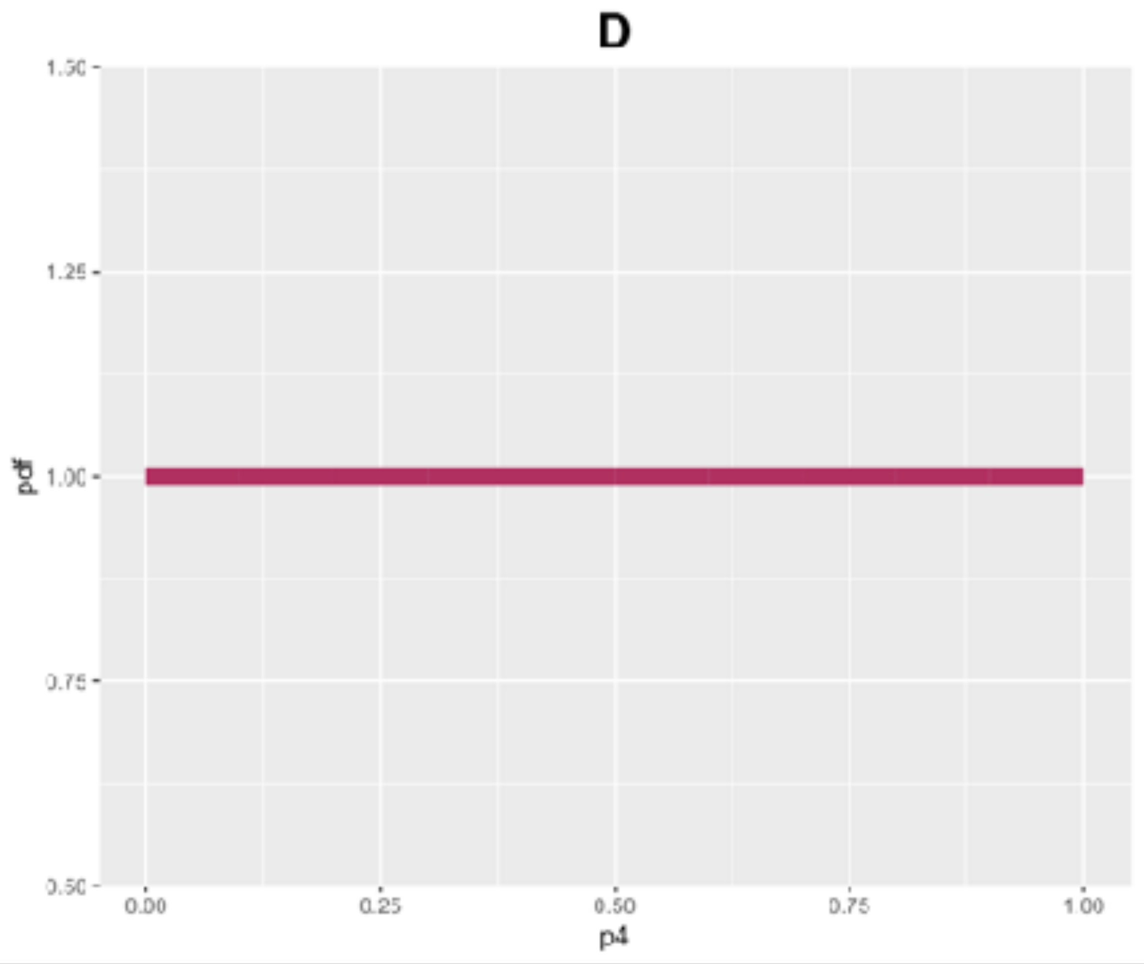
MAB

Thompson sampling: Example





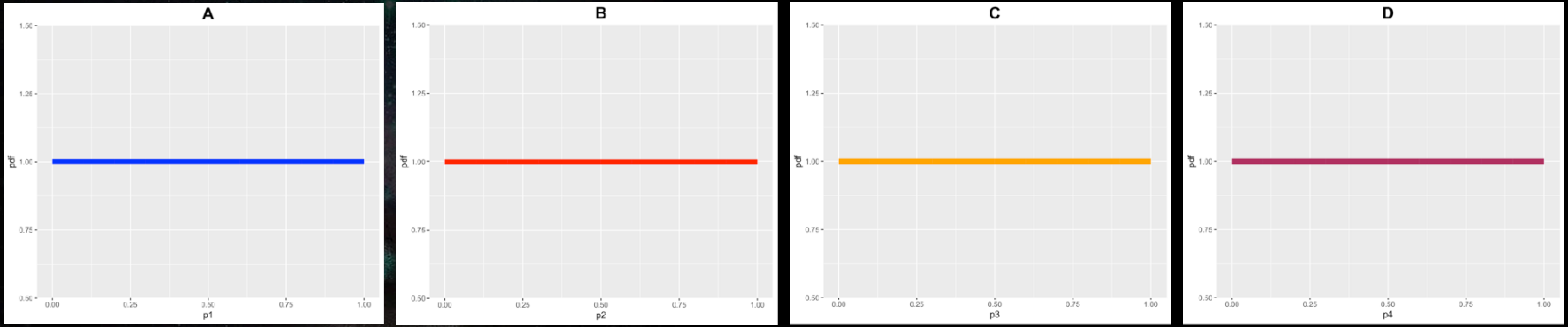




0.1 0.5 0.3 0.84

MAB

Thompson sampling: Example

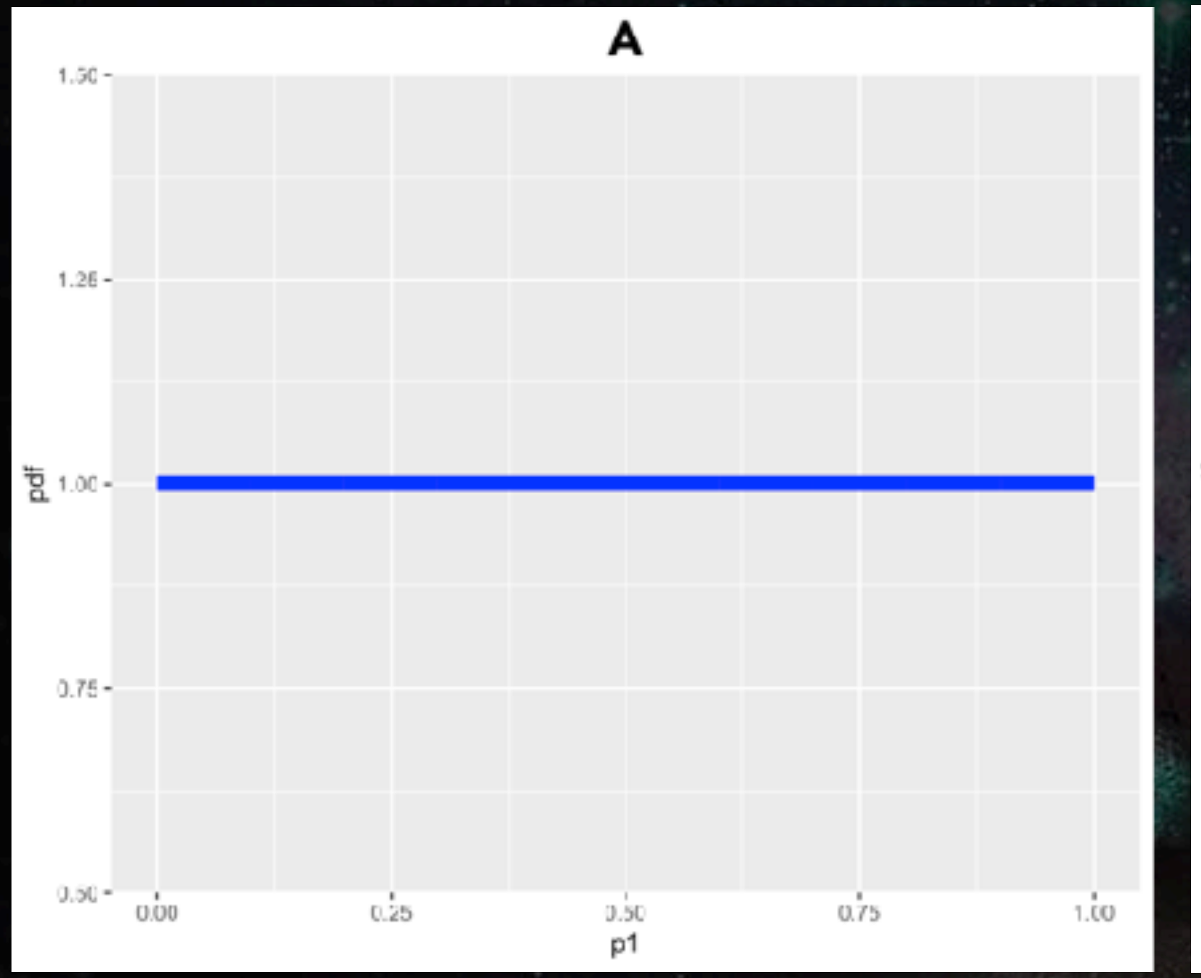


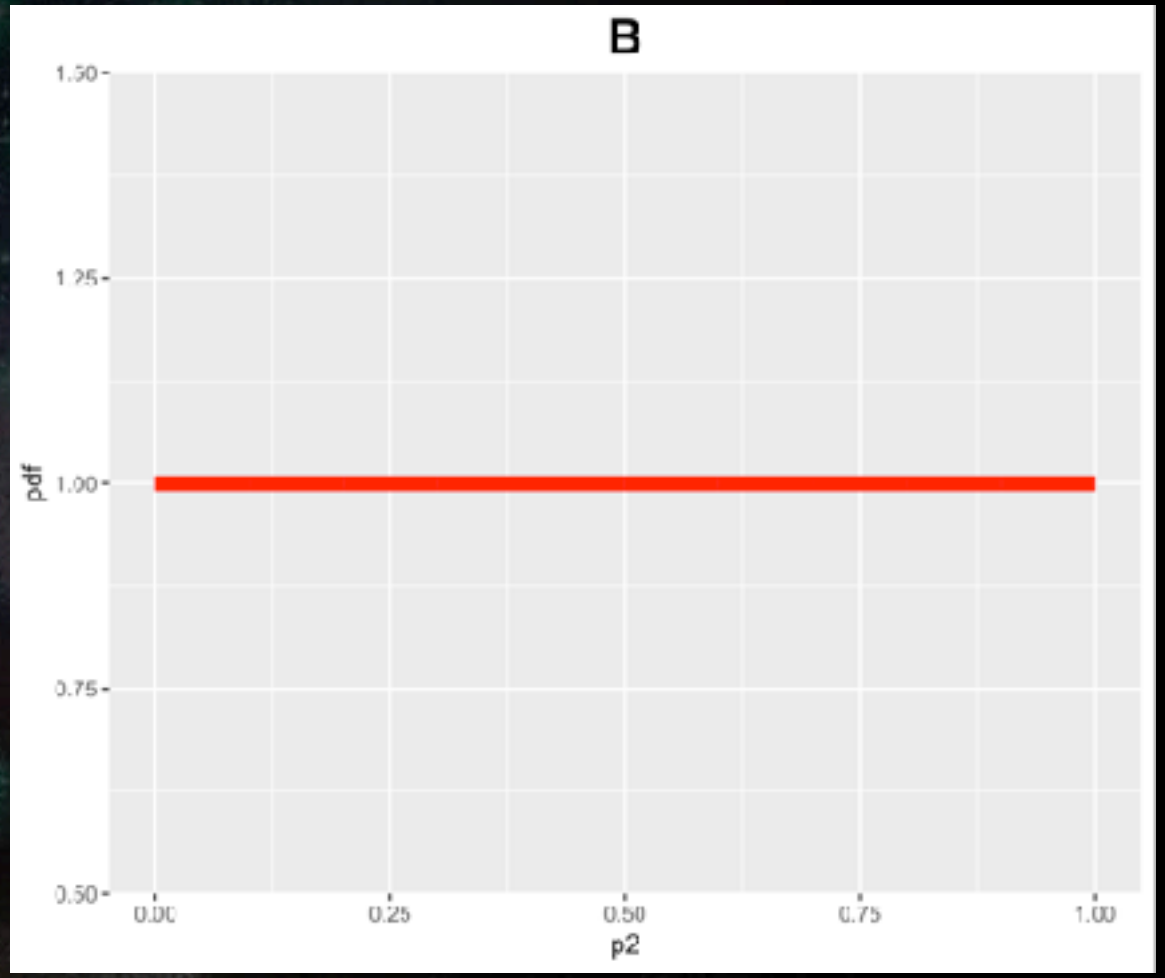
0.1 0.5 0.3 $\boxed{0.84}$

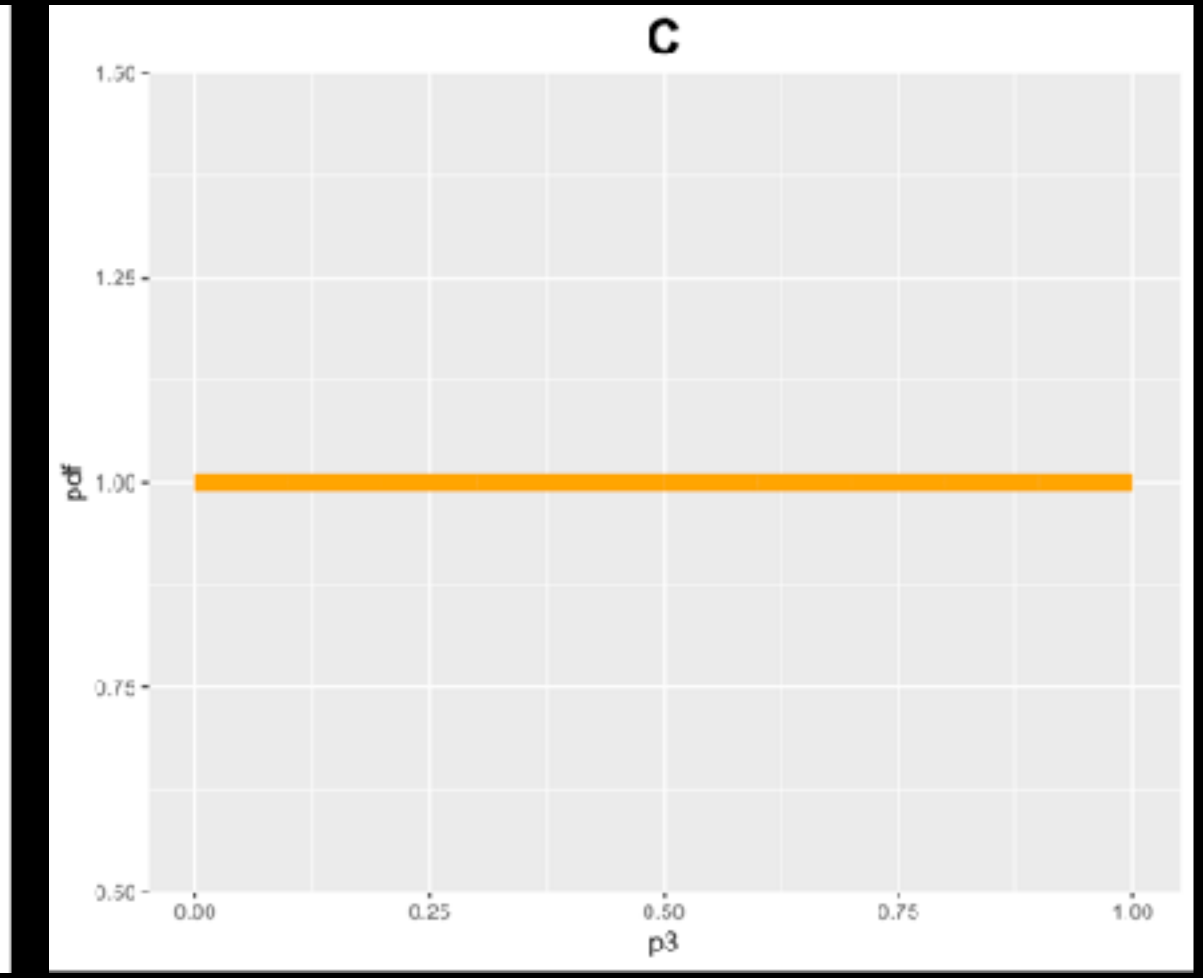
Lightricks

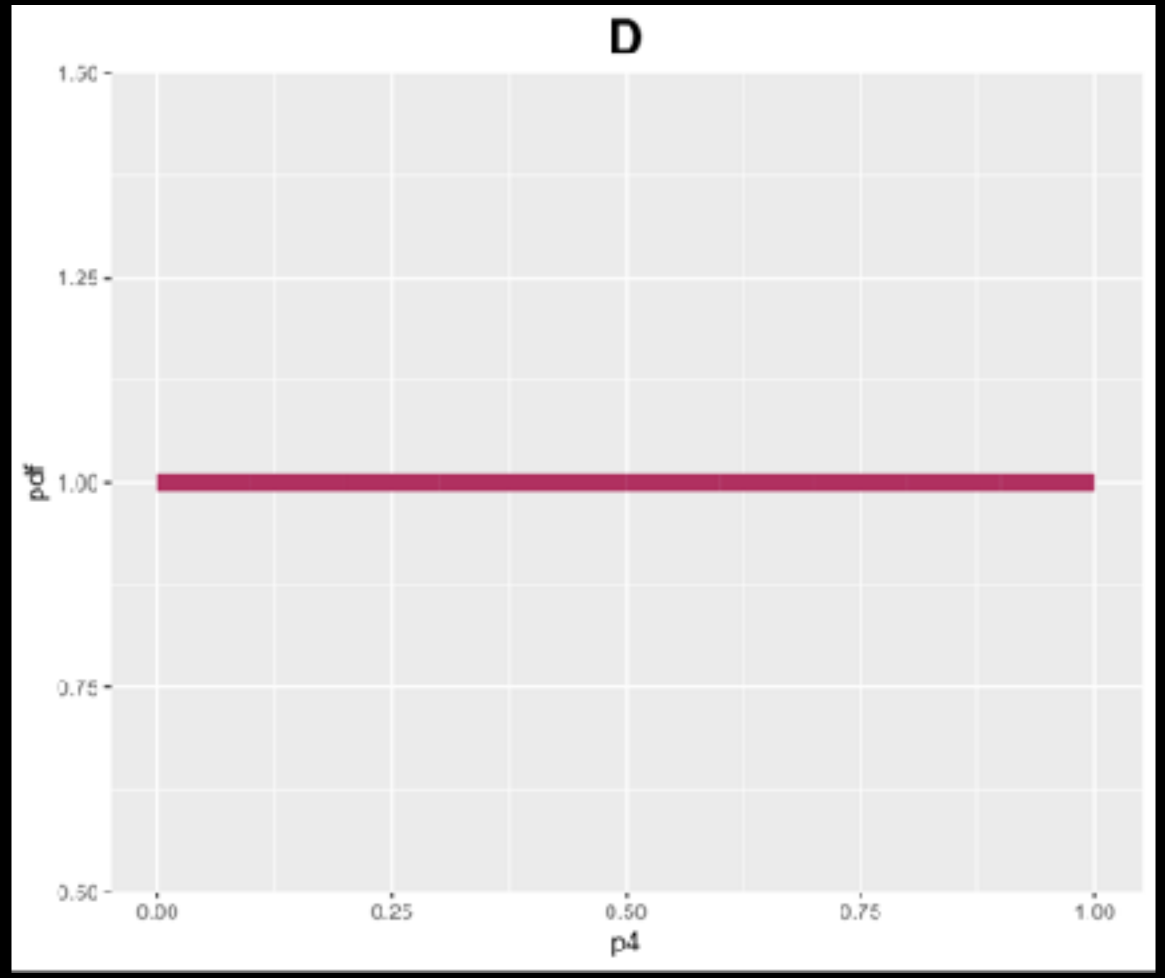
MAB

Thompson sampling: Example









0.1 0.5 0.3 $\boxed{0.84}$

Buy Did not buy

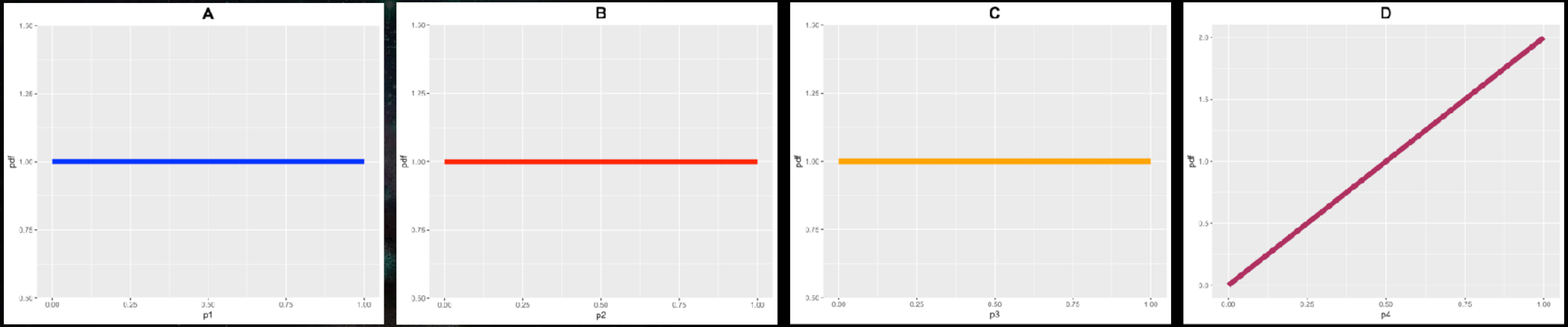
MAB

Thompson sampling: Example

Lightricks

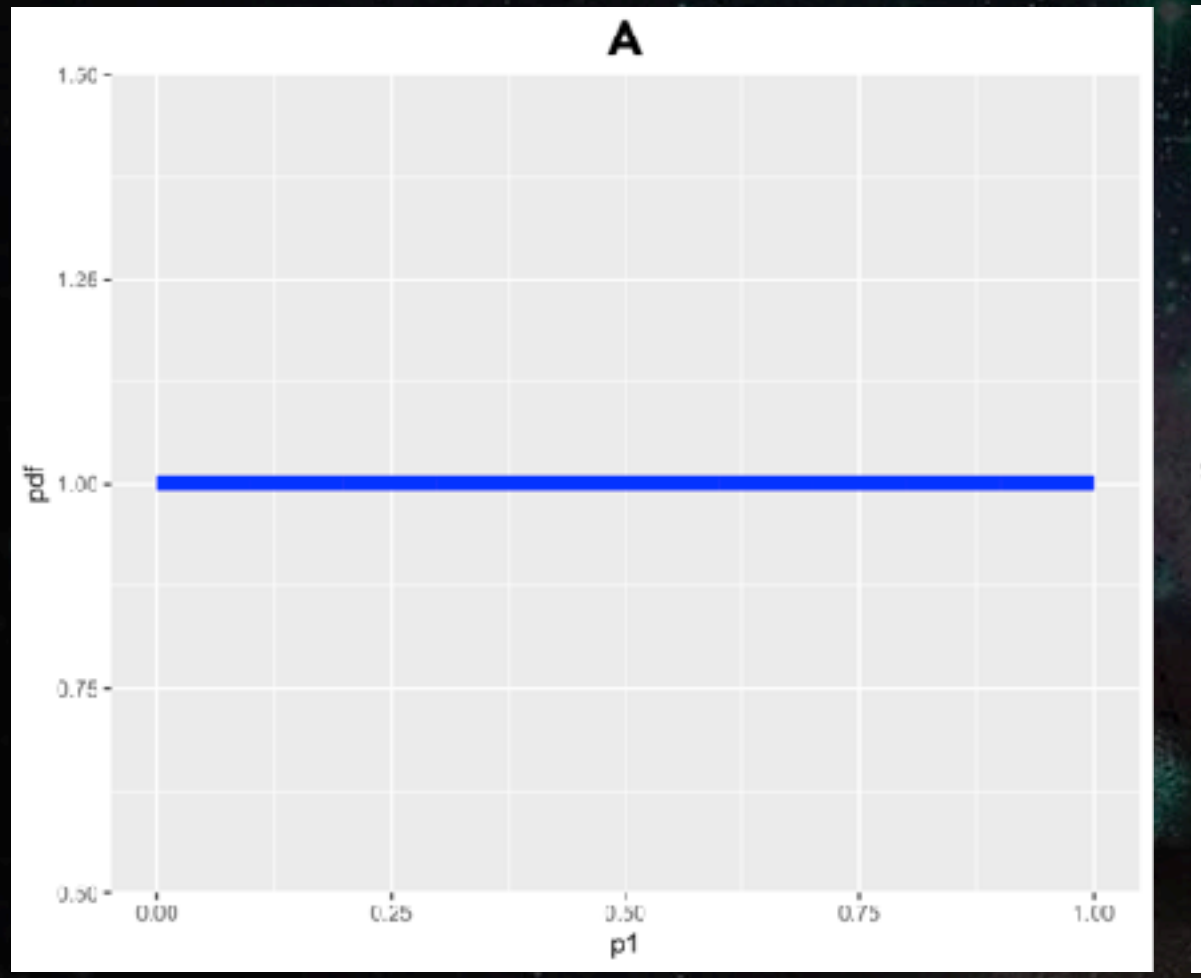
MAB

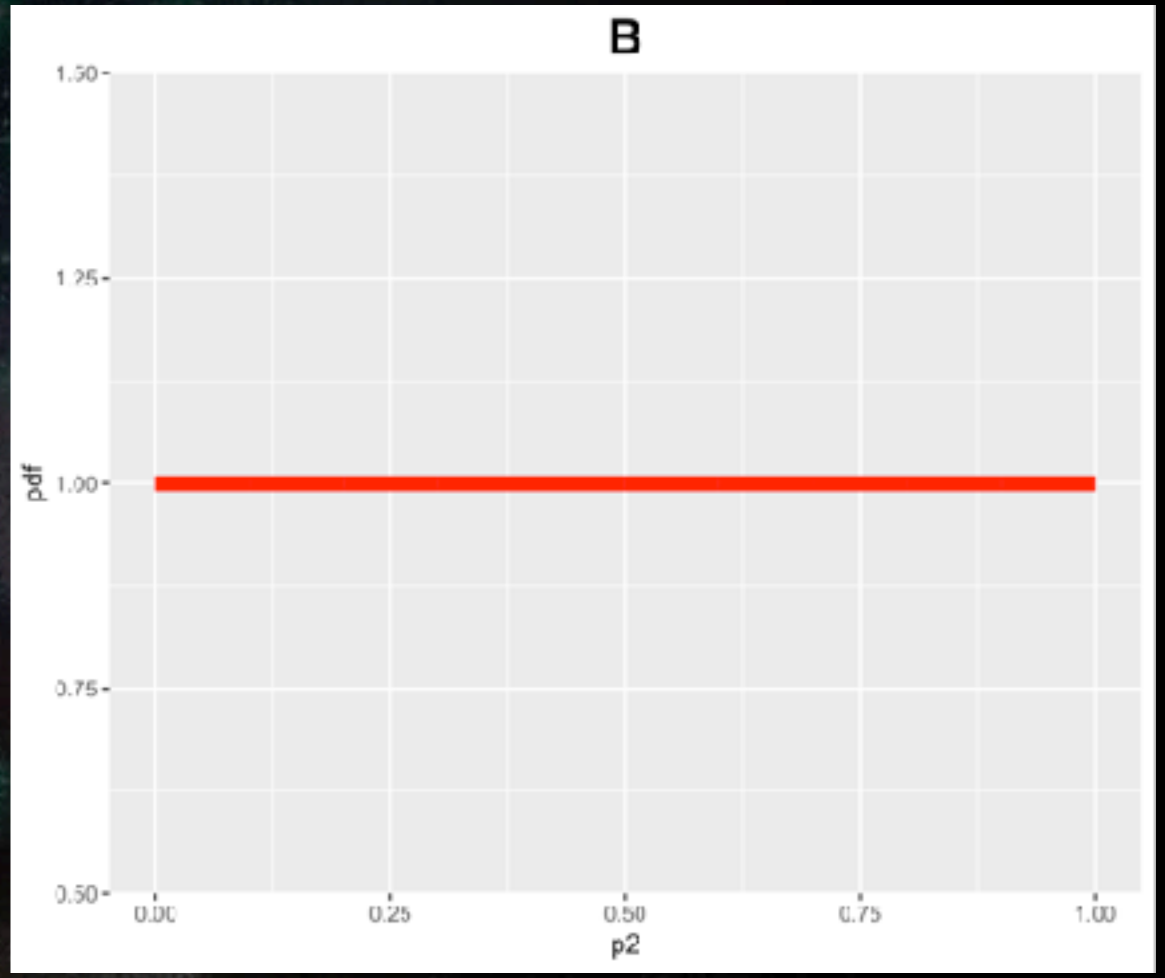
Thompson sampling: Example

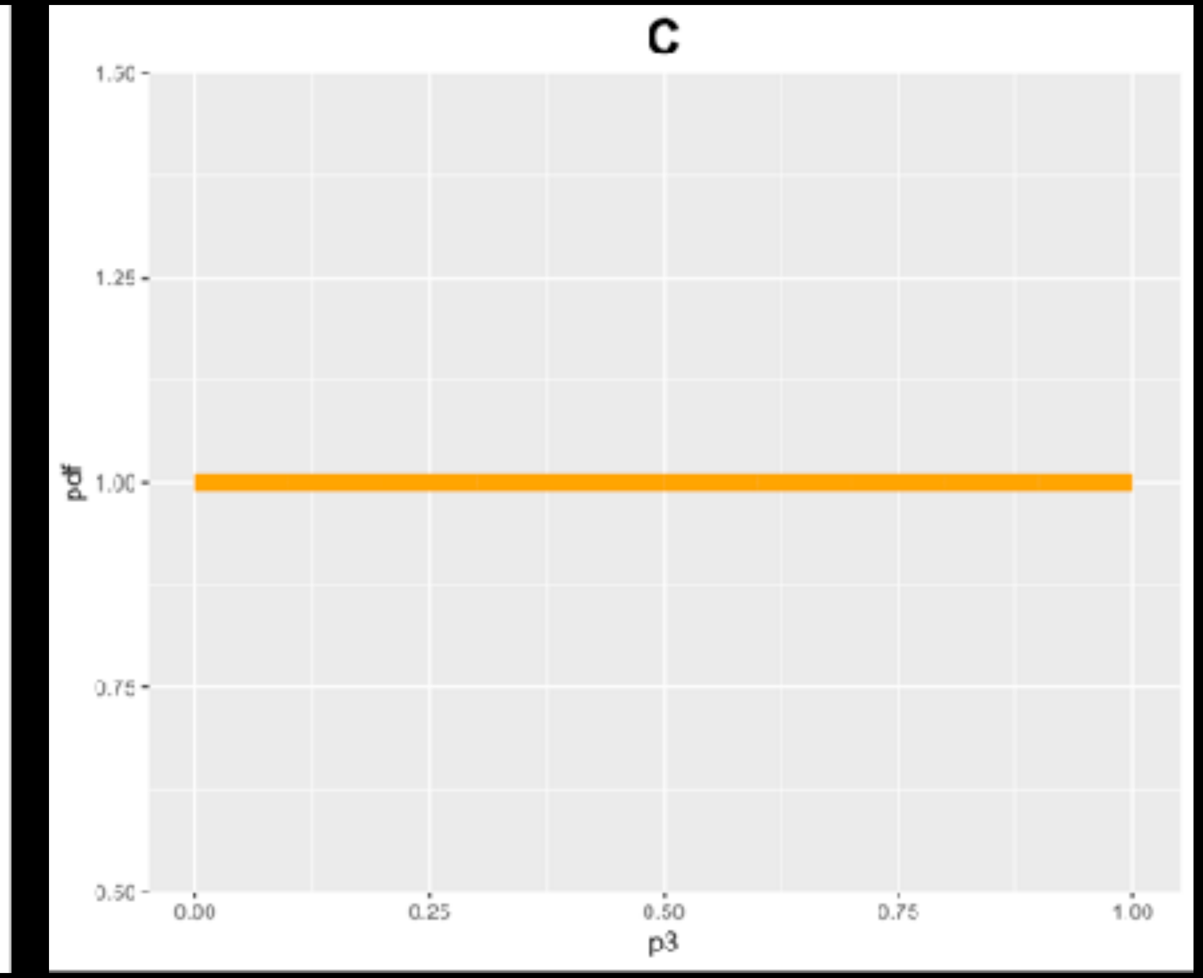


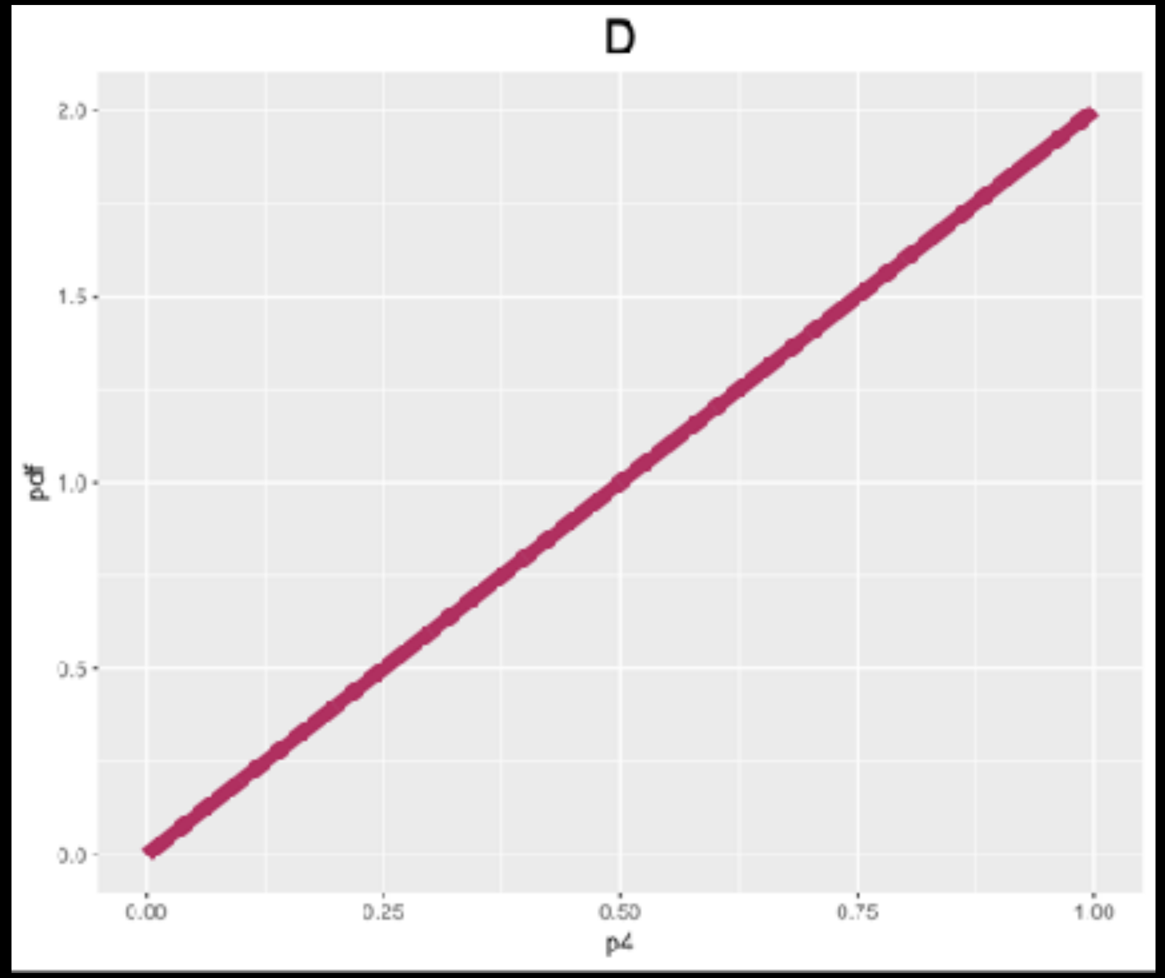
MAB

Thompson sampling: Example





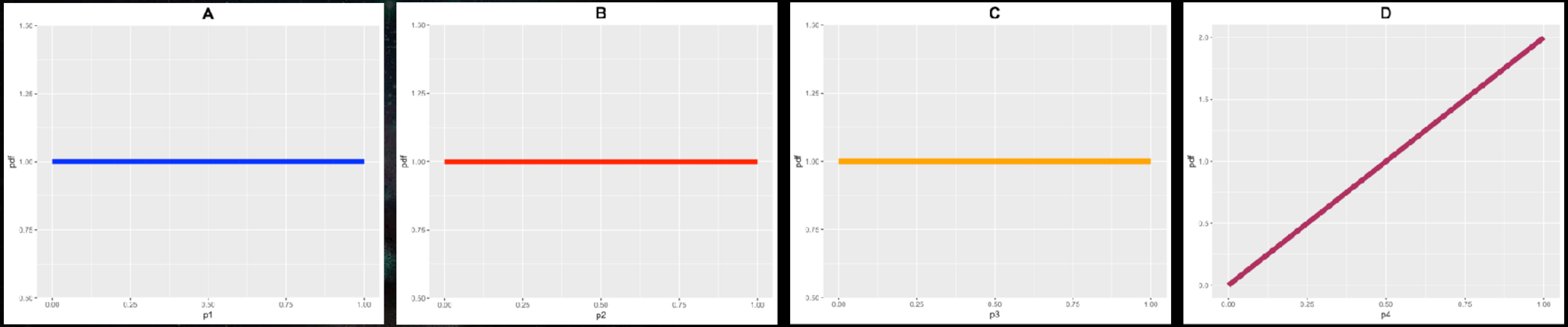




0.4 0.9 0.6 0.75

MAB

Thompson sampling: Example



0.4

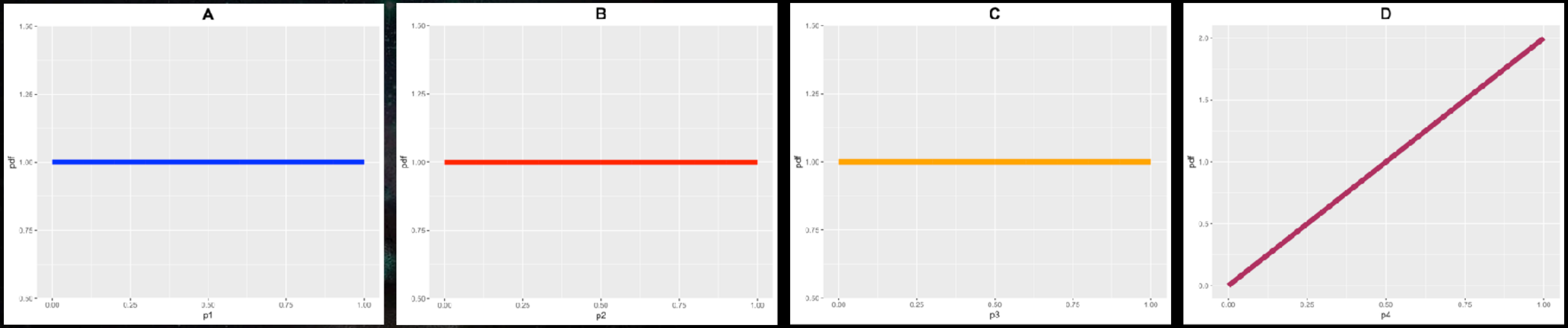
0.9

0.6

0.75

MAB

Thompson sampling: Example



0.4 | 0.9 | 0.6 | 0.75

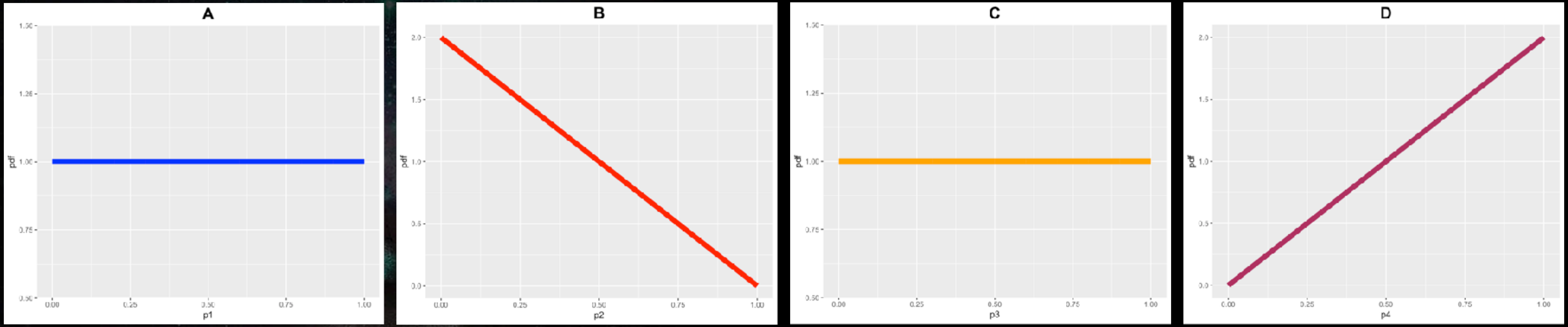
Buy | Did not buy

MAB

Thompson sampling: Example

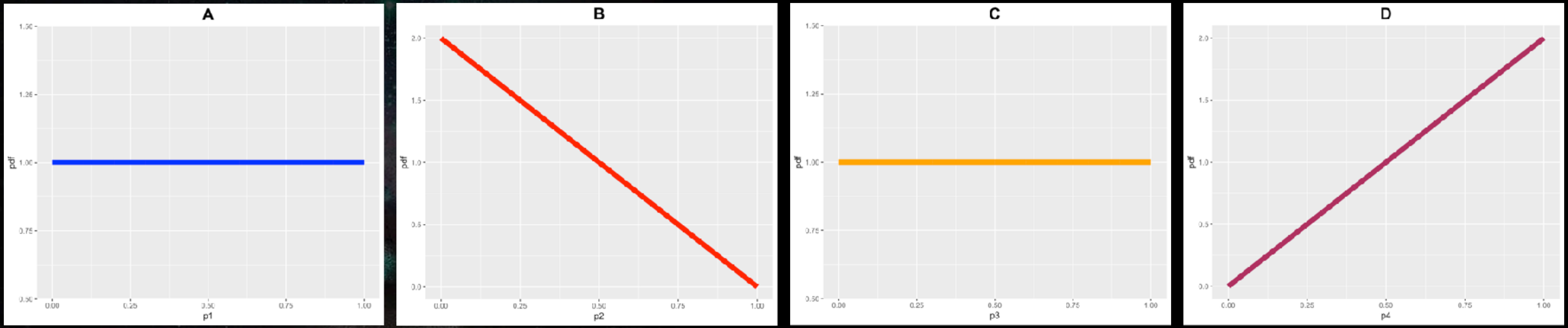
MAB

Thompson sampling: Example



MAB

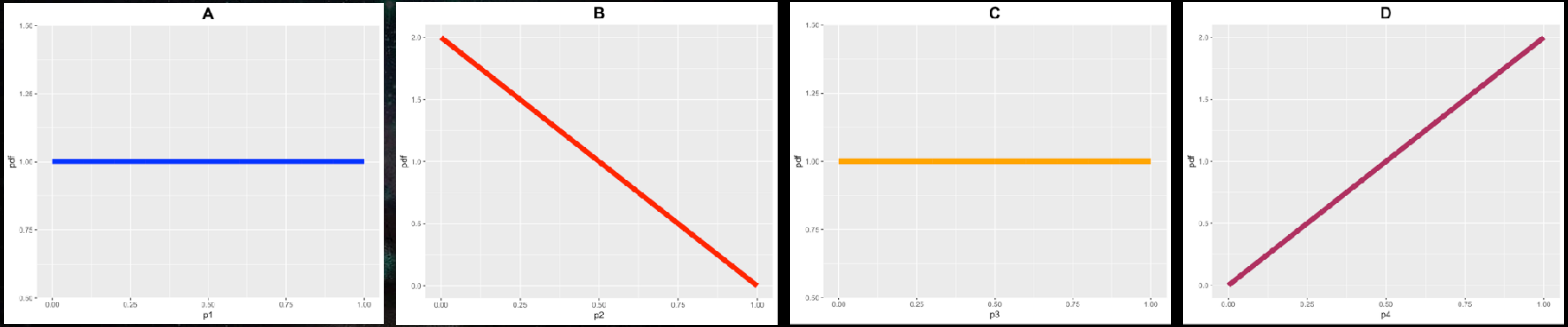
Thompson sampling: Example



0.3 0.5 0.2 0.8

MAB

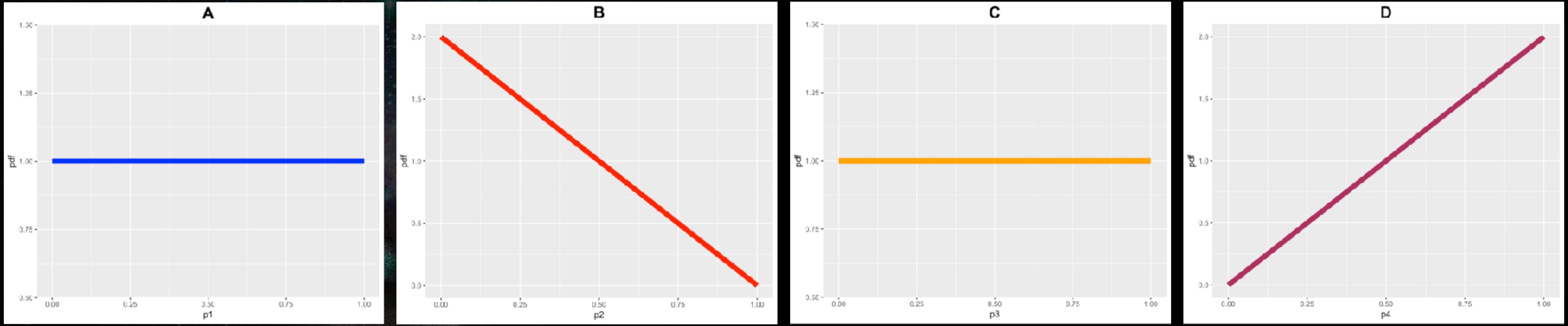
Thompson sampling: Example



0.3 0.5 0.2 $\boxed{0.8}$

MAB

Thompson sampling: Example

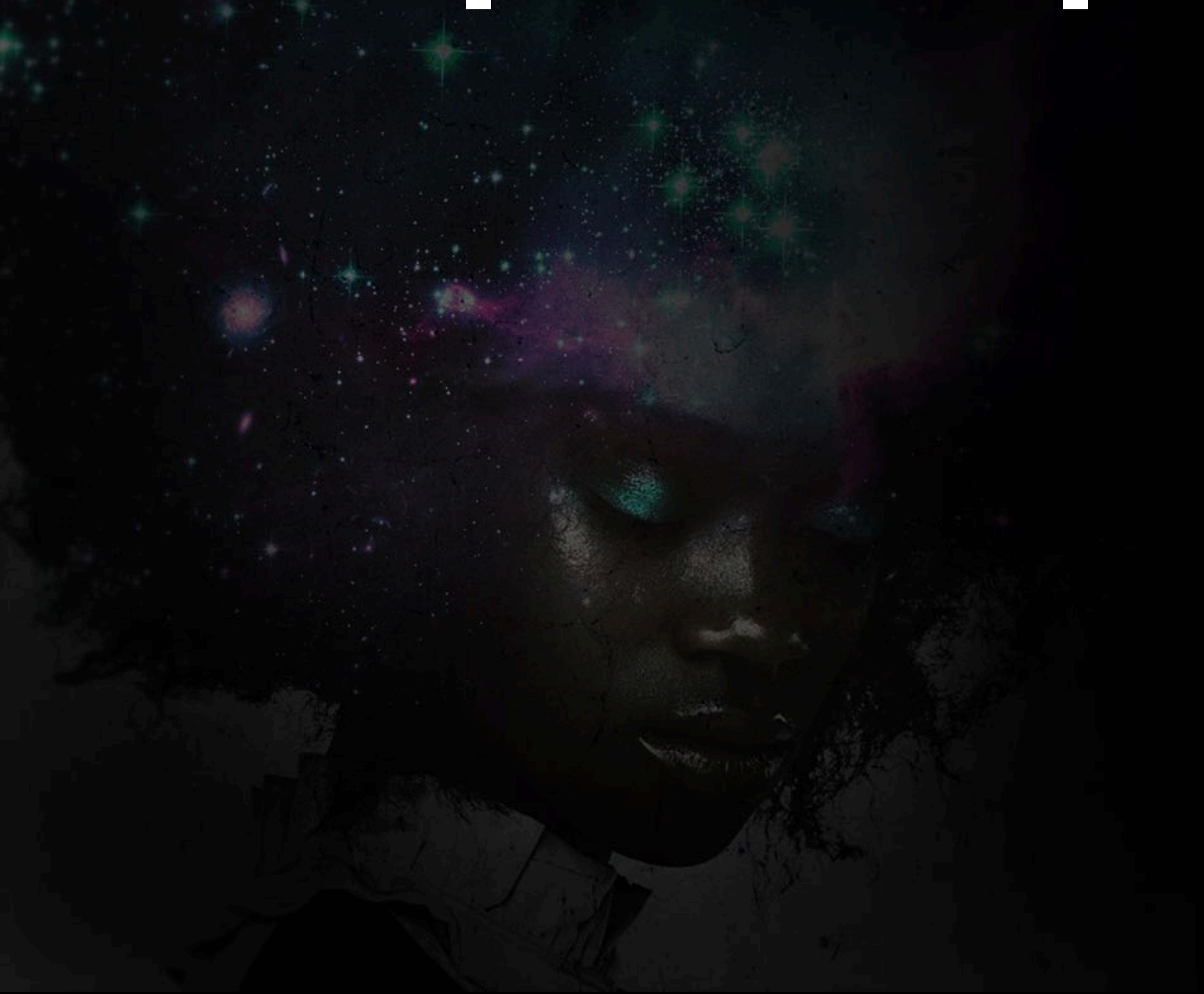


0.3 0.5 0.2 0.8

Buy Did not buy

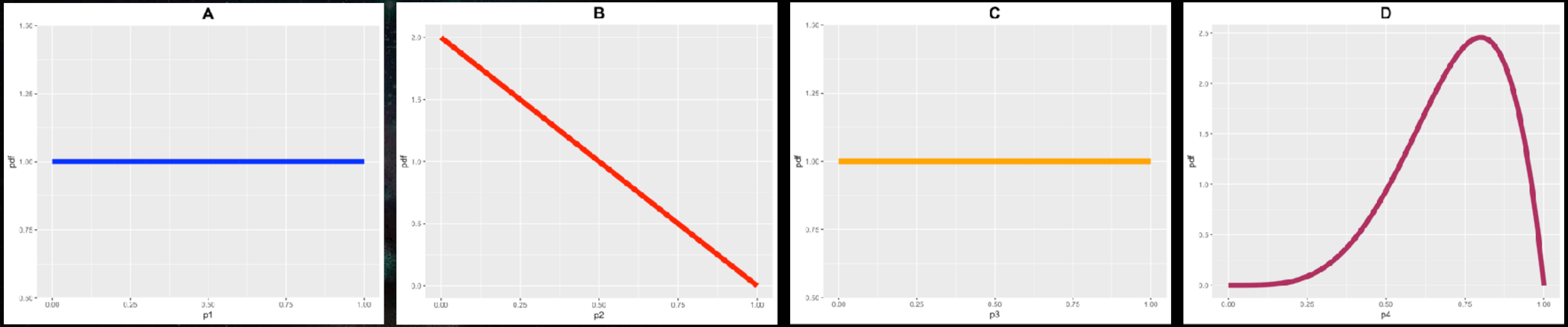
MAB

Thompson sampling: Example



MAB

Thompson sampling: Example



Lightricks

Generalized Weighted Thompson Sampling (GWTS)

GWTS

Motivation

* Naturally, our high-level goal is to select the version with the highest expected reward.
* When the reward is binary, this problem is equivalent to selecting the version with maximal p_i

GWTS

Motivation

However, this may not always be the case:

- For example, consider a case where a user can purchase one of the following:
  1. Monthly subscription
  2. Yearly subscription
  3. No subscription

- Assume for example that $reward_{yearly} = 2 \cdot reward_{monthly}$

- In this case, the expected reward is:

$$E(reward \mid version_i) = E(reward \mid monthly, version_i) \cdot P(monthly \mid version_i) + E(reward \mid yearly, version_i) \cdot P(yearly \mid version_i)$$

GWTS

Motivation

- For simplicity, assume that the reward is fixed for a given subscription among all versions.
- Intuitively, we may prefer a version with a smaller overall conversion rate but which has a high yearly subscription rate, over a version that has a higher overall conversion rate, but where all users bought a monthly subscription.

GWTS

GWTS Algorithm

The proposed algorithm extends the Thompson sampling to a more general framework.

GWTS alg:

A

Initialize with Dirichlet(**1**) distribution (which is the multidimensional standard Uniform distribution)

B

Generate m (numbers of versions) random vectors from the Dirichlet distribution with current parameters. Denote them as $(p_{11}, p_{12}, \ldots p_{1n}, p_{21}, \ldots, p_{2n}, p_{m1}, \ldots p_{mn})$ where n is the number of different choices within each version (i.e. choice of subscription)

C

Propose a version according to the previous distribution, i.e. the version i such that

$$\arg\max_i \sum_{j=1}^{d} p_{ij} w_j$$

Where w_j is the fixed reward of subscription j.

GWTS

GWTS Algorithm

The proposed algorithm extends the Thompson sampling to a more general framework.

GWTS alg:

D

Observe data with the proposed version.

E

Update the Dirichlet parameters of the chosen version according to the gathered data by the following formula:

$$a_{ij}^{new} = a_{ij}^{old} + \#\,buyers\ of\ version\ i\ with\ subscription\ j$$

F

Repeat step B-E a given number of times/until convergence to a specific version and return the optimal arm.

Results

Results

The Procedure

We now focus on the performance of the standard Thompson-sampling and the GWTS.

The standard Thompson sampling was performed in real time on actual data.

The procedure was as follows:

1. 38 versions of the app were tested.
2. Perform Thompson sampling to choose the 7 highest rates of convergence.
3. Perform A/B Testing on the 7 highest versions.

Results

A Variant

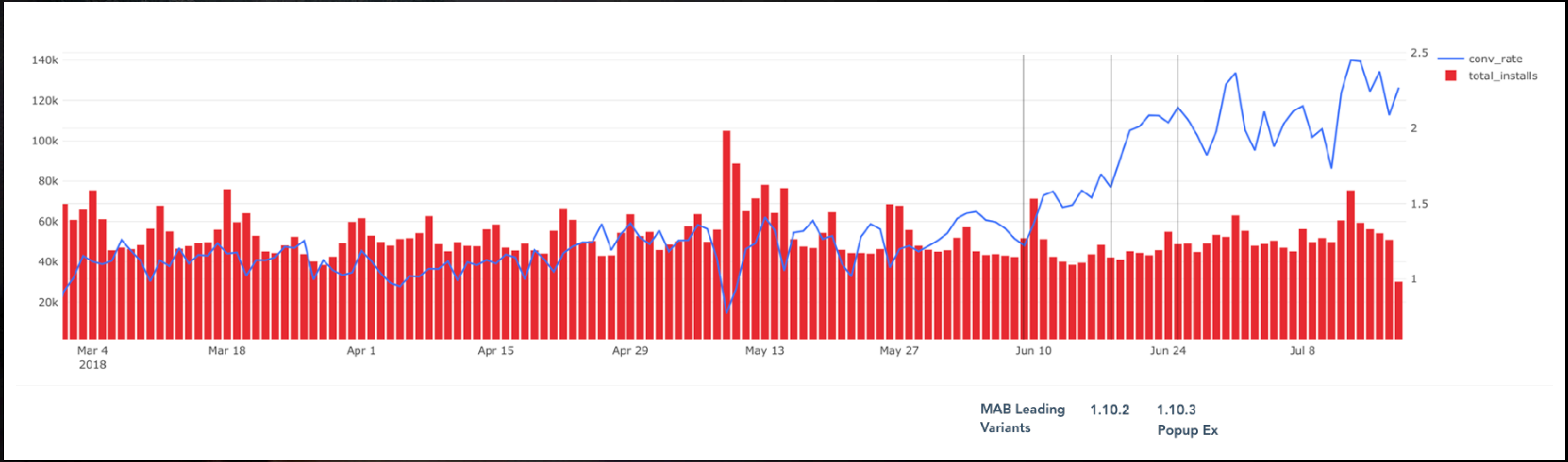
Here is a variant of this approach:

Use Thompson sampling to eliminate the lower performing versions and perform separate tests on the highest results.

Lightricks

Results

MAB Trial Results



- The graph shows conversion rate over time, before MAB and during MAB.
- We can see the ascending trend during the MAB trial and after it.

Results

GWTS Simulations Results

- Several simulations were performed on the GWTS, and some are presented in the following.

Results

General Simulation Framework

Number of Iterations

1000

Sample size

250

Reward value for monthly subscription

1

Reward value for yearly subscription

Between 1.5 and 5

Reward value for non-buying

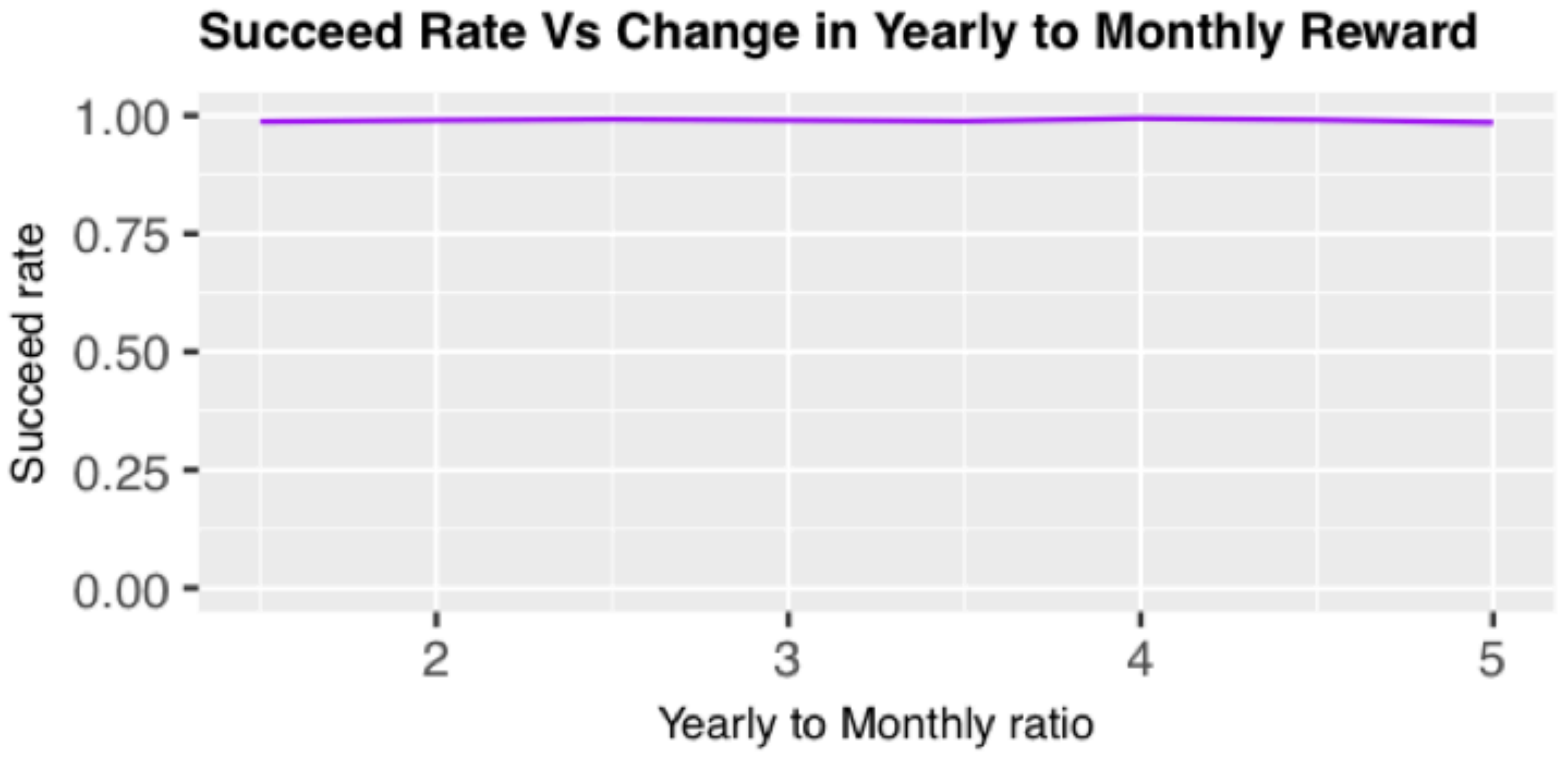
0

Results

Simulation: 2 versions

	real.p.1	real.p.2	real.p.3
A	0.01	0.0300	0.9600
B	0.01	0.0275	0.9625

	Succeed rate	Expected 1st arm reward	Expected 2nd arm reward
5	0.985	0.160	0.14750
4.5	0.991	0.145	0.13375
4	0.994	0.130	0.12000
3.5	0.988	0.115	0.10625
3	0.990	0.100	0.09250
2.5	0.992	0.085	0.07875
2	0.990	0.070	0.06500
1.5	0.987	0.055	0.05125



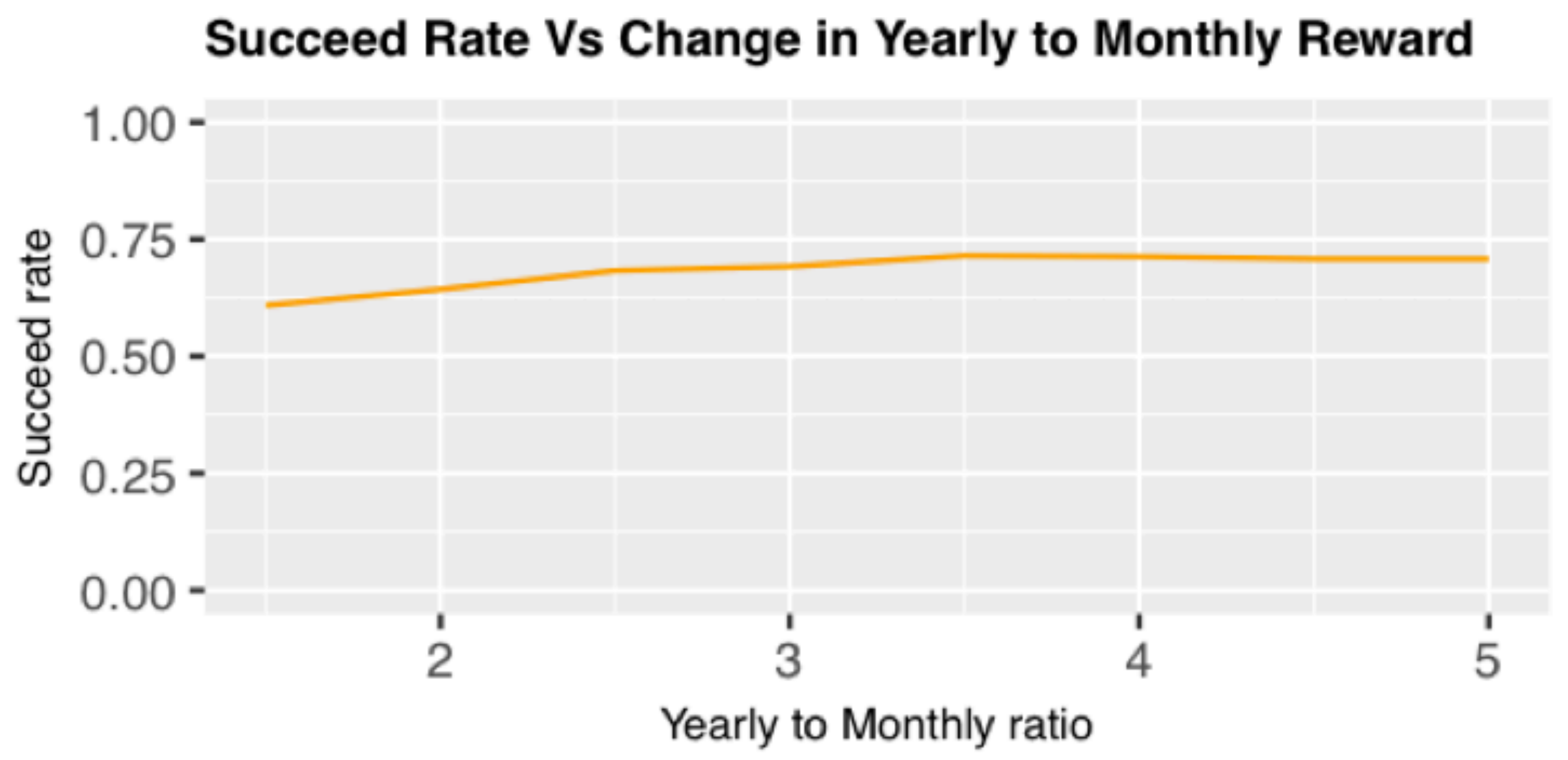
Results

Simulation: 2 versions with constraint

A more challenging and realistic scenario is when: $p_{monthly} + p_{yearly} = 2.5\%$

	real.p.1	real.p.2	real.p.3
A	0.016250	0.008750	0.975
B	0.016625	0.008375	0.975

	Succeed rate	Expected 1st arm reward	Expected 2nd arm reward
5	0.708	0.060000	0.0585000
4.5	0.708	0.055625	0.0543125
4	0.713	0.051250	0.0501250
3.5	0.715	0.046875	0.0459375
3	0.692	0.042500	0.0417500
2.5	0.683	0.038125	0.0375625
2	0.643	0.033750	0.0333750
1.5	0.609	0.029375	0.0291875



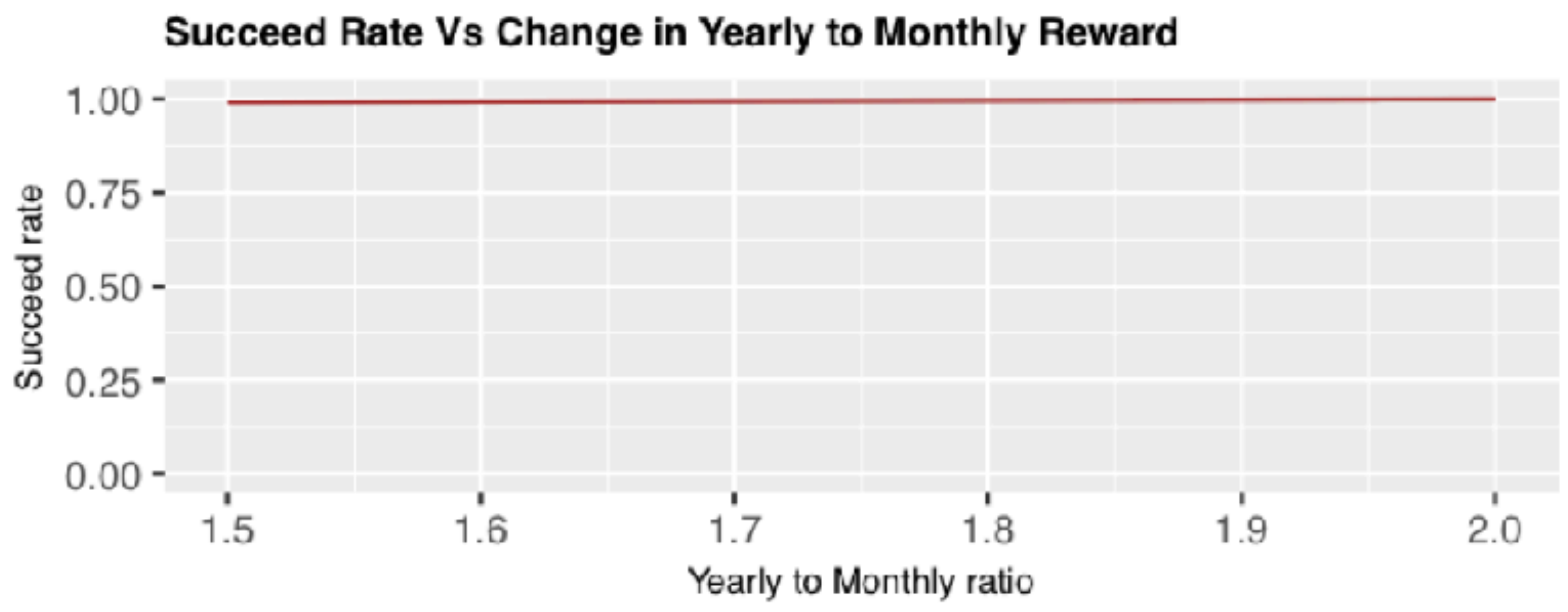
Results

Simulation: 10 Arms, Remarkable Winner

	real.p.1	real.p.2	real.p.3
A	0.0050000	0.0200000	0.975
B	0.0171599	0.0078401	0.975
C	0.0172039	0.0077961	0.975
D	0.0156136	0.0093864	0.975
E	0.0170520	0.0079480	0.975
F	0.0164820	0.0085180	0.975
G	0.0171080	0.0078920	0.975
H	0.0161546	0.0088454	0.975
I	0.0156137	0.0093863	0.975
J	0.0156941	0.0093059	0.975

	Succeed rate	1st	2nd	3rd	4th	5th
1.5	0.992	0.035	0.0289200	0.0288980	0.0296932	0.028974
2	1.000	0.045	0.0328401	0.0327961	0.0343864	0.032948

	Succeed rate	6th	7th	8th	9th	10th
1.5	0.992	0.029259	0.028946	0.0294227	0.0296931	0.0296529
2	1.000	0.033518	0.032892	0.0338454	0.0343863	0.0343059



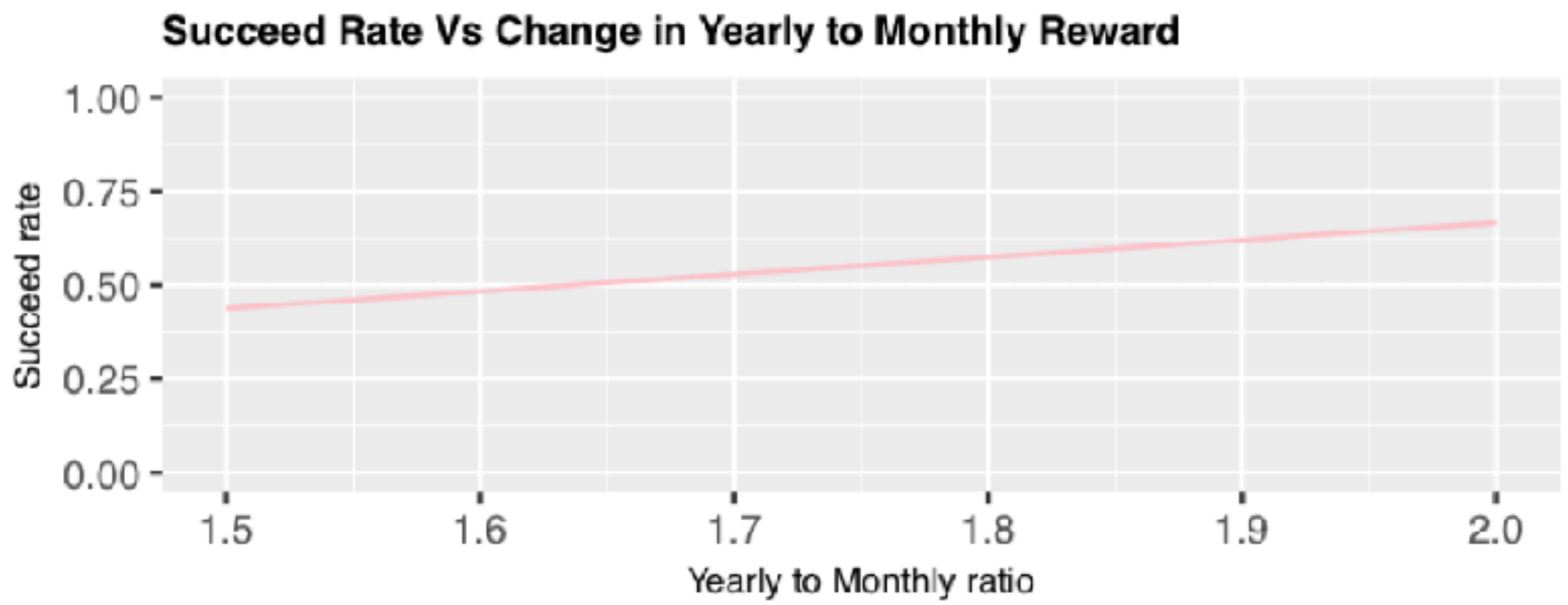
Results

Simulation: 10 Arms, Challenging Case

	real.p.1	real.p.2	real.p.3
A	0.0137500	0.0112500	0.975
B	0.0171599	0.0078401	0.975
C	0.0172039	0.0077961	0.975
D	0.0156136	0.0093864	0.975
E	0.0170520	0.0079480	0.975
F	0.0164820	0.0085180	0.975
G	0.0171080	0.0078920	0.975
H	0.0161546	0.0088454	0.975
I	0.0156137	0.0093863	0.975
J	0.0156941	0.0093059	0.975

	Succeed rate	1st	2nd	3rd	4th	5th
1.5	0.438	0.030625	0.0289200	0.0288980	0.0296932	0.028974
2	0.666	0.036250	0.0328401	0.0327961	0.0343864	0.032948

	Succeed rate	6th	7th	8th	9th	10th
1.5	0.438	0.029259	0.028946	0.0294227	0.0296931	0.0296529
2	0.666	0.033518	0.032892	0.0338454	0.0343863	0.0343059



Results

Conclusion

- The standard Thompson sampling performed incredibly well on real data in real time.
- Thompson sampling is easy to adjust for testing a wide range of statistical summaries.
- The GWTS has outstanding performance when the winner is quite differentiable from the others.
- Possible improvement: relax the assumptions about the rewards.

Results

Conclusion

* The standard Thompson sampling performed incredibly well on real data in real time.
* Thompson sampling is easy to adjust for testing a wide range of statistical summaries.
* Possible improvement: relax the assumptions about the rewards.

Lightricks

Thank You!