CMPT 473
Software Testing, Reliability and Security

A/B Testing & Bandit Based Solutions

Nick Sumner
wsumner@sfu.ca

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way

How do you know that a change adds value?

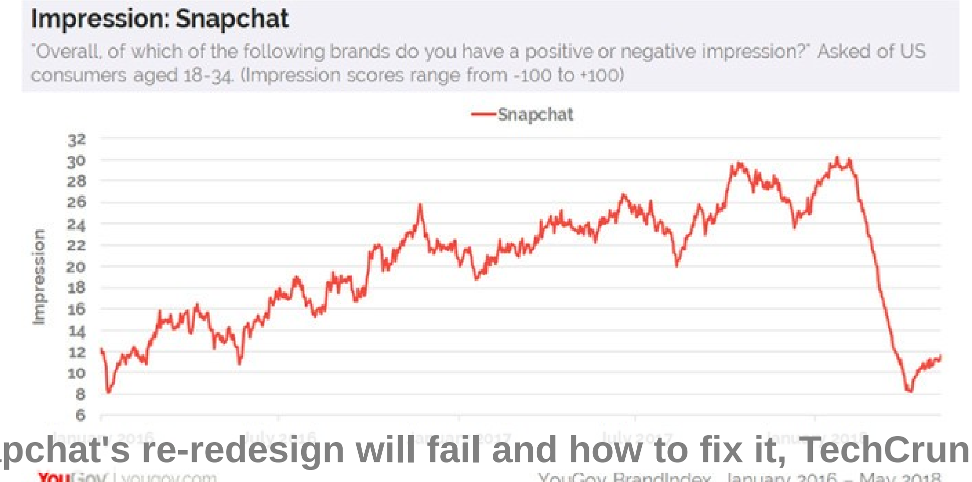
- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?



Why Snapchat's re-redesign will fail and how to fix it, TechCrunch, 2018.

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?

You should already have an *intuition* for attacking this.
What should you do?

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?
- Solutions
  - ***A/B Testing*** uses different forms of hypothesis testing
  - Alternatively, you can use ***multi-armed bandits*** to attack the problem

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?
- Solutions
  - ***A/B Testing*** uses different forms of hypothesis testing
  - Alternatively, you can use ***multi-armed bandits*** to attack the problem
  - Key idea: run controlled experiments live on the deployed software

How do you know that a change adds value?

- The scenario
  - You maintain a web site and are considering a change
  - You hypothesize that the change improves outcomes in some way
- The problem
  - How can you find out whether one change (or many!) improves results?
  - How can you do this without costing your company money?
- Solutions
  - ***A/B Testing*** uses different forms of hypothesis testing
  - Alternatively, you can use ***multi-armed bandits*** to attack the problem
  - Key idea: run controlled experiments live on the deployed software
- Caveat: We ***will not*** dive into a full stats background for these
  - We ***will*** discuss some common pitfalls that arise from misunderstandings

When might you want to know?

- Exploring ideas to improve usability

When might you want to know?

- Exploring ideas to improve usability
  - Or performance (throughput, latency, ...)

When might you want to know?

- Exploring ideas to improve usability
  - Or performance (throughput, latency, ...)
- Establishing the effectiveness of promotion before campaigns

When might you want to know?

- Exploring ideas to improve usability
  - Or performance (throughput, latency, ...)
- Establishing the effectiveness of promotion before campaigns
- Staged rollouts of major changes

When might you want to know?

- Exploring ideas to improve usability
  - Or performance (throughput, latency, ...)
- Establishing the effectiveness of promotion before campaigns
- Staged rollouts of major changes
  - Minimizing risk of: CD, fragmented configurations, ...

    e.g. rolling out apps to the Android store

Simple A/B Testing

- You have:
  - two solutions, A and B (e.g., **A** is old, **B** is new)
  - A hypothesis (e.g. **A** will improve conversion over **B** by at least 5%)

Simple A/B Testing

- You have:
  - two solutions, A and B (e.g., **A** is old, **B** is new)
  - A hypothesis (e.g. **A** will improve conversion over **B** by at least 5%)
- Basic solution:
  - Determine what data to collect (choose population, metric, & size up front!!!)

Simple A/B Testing

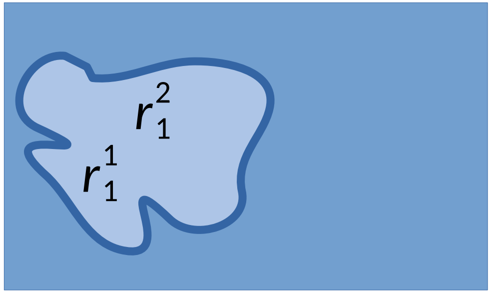
- You have:
  - two solutions, A and B (e.g., **A** is old, **B** is new)
  - A hypothesis (e.g. **A** will improve conversion over **B** by at least 5%)
- Basic solution:
  - Determine what data to collect (choose population, metric, & size up front!!!)
  - Randomly provide(/serve) A to one population and B to another to collect predetermined stats

Simple A/B Testing

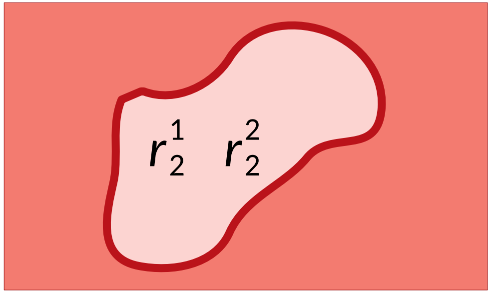
- You have:
  - two solutions, A and B (e.g., **A** is old, **B** is new)
  - A hypothesis (e.g. **A** will improve conversion over **B** by at least 5%)
- Basic solution:
  - Determine what data to collect (choose population, metric, & size up front!!!)
  - Randomly provide(/serve) A to one population and B to another to collect predetermined stats
  - Use a basic t-test to measure differences in the populations

Simple A/B Testing

- You have:
  - two solutions, A and B (e.g., **A** is old, **B** is new)
  - A hypothesis (e.g. **A** will improve conversion over **B** by at least 5%)
- Basic solution:
  - Determine what data to collect (choose population, metric, & size up front!!!)
  - Randomly provide(/serve) A to one population and B to another to collect predetermined stats
  - Use a basic t-test to measure differences in the populations

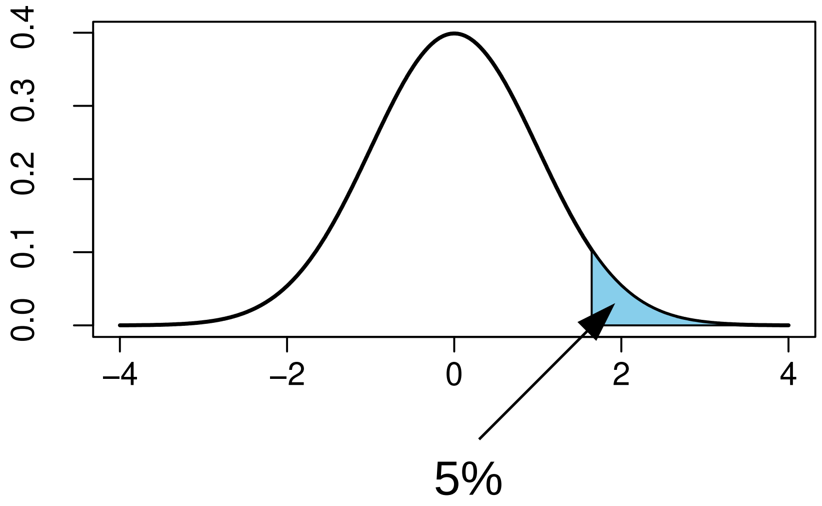


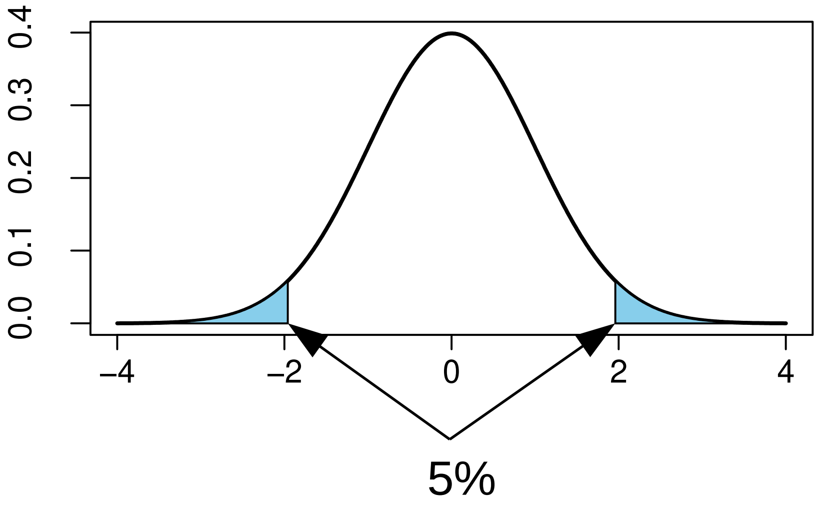
$$\mu_1 \overset{?}{<} \mu_2$$



Recalling T-tests

- Can be one-sided (tailed) or two sided (tailed)
  - distinguishing directed and undirected differences





Recalling T-tests

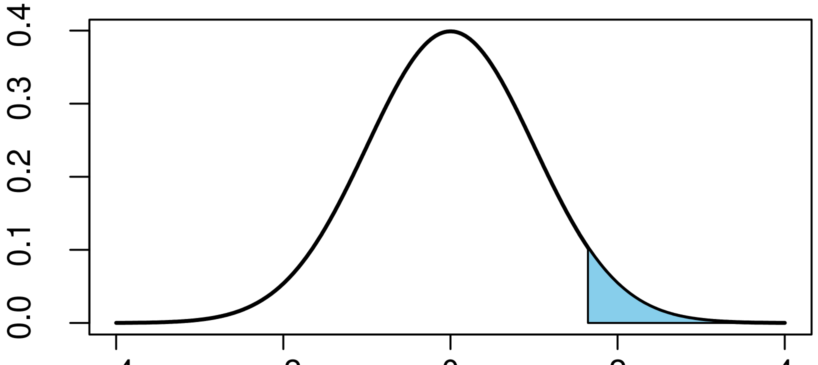
- Can be one-sided (tailed) or two sided (tailed)
  - distinguishing directed and undirected differences
- Assume (1) observation independence and (2) normal distribution

Recalling T-tests

- Can be one-sided (tailed) or two sided (tailed)
  - distinguishing directed and undirected differences
- Assume (1) observation independence and (2) normal distribution
- Distinguish 2 hypotheses (e.g.):
  - H_0: $\mu_1 - \mu_2 = 0$ (the null hypothesis – assumed true until disproven)
  - H_1: $\mu_1 < \mu_2$ (the alternative)

Recalling T-tests

- Can be one-sided (tailed) or two sided (tailed)
  - distinguishing directed and undirected differences
- Assume (1) observation independence and (2) normal distribution
- Distinguish 2 hypotheses (e.g.):
  - H_0: $\mu_1 - \mu_2 = 0$ (the null hypothesis – assumed true until disproven)
  - H_1: $\mu_1 < \mu_2$ (the alternative)



Recalling T-tests

- Can be one-sided (tailed) or two sided (tailed)
  - distinguishing directed and undirected differences
- Assume (1) observation independence and (2) normal distribution
- Distinguish 2 hypotheses (e.g.):
  - H_0: $\mu_1 - \mu_2 = 0$ (the null hypothesis – assumed true until disproven)
  - H_1: $\mu_1 < \mu_2$ (the alternative)
  - **RECALL:**
    We never prove a hypothesis!
    We gather sufficient evidence to reject the null hypothesis and thus accept the alternative

Recalling T-tests

$$t = \frac{\left(\overline{x}_1 - \overline{x}_2\right) - \Delta}{\frac{\sqrt{S_1^2}}{m} + \frac{\sqrt{S_2^2}}{n}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

Recalling T-tests

$$t = \frac{(\overline{x}_1 - \overline{x}_2) - \Delta}{\frac{\sqrt{S_1^2}}{m} + \frac{\sqrt{S_2^2}}{n}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$

H_a: $\mu_1 - \mu_2 < \Delta$

H_a: $\mu_1 - \mu_2 \neq \Delta$

Recalling T-tests

$$t = \frac{\left(\overline{x}_1 - \overline{x}_2\right) - \Delta}{\frac{\sqrt{S_1^2}}{m} + \frac{\sqrt{S_2^2}}{n}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$	$t > t_{\alpha, v}$
H_a: $\mu_1 - \mu_2 < \Delta$	$t < -t_{\alpha, v}$
H_a: $\mu_1 - \mu_2 \neq \Delta$	$\lvert t \rvert > t_{\alpha/2, v}$

Recalling T-tests

$$t = \frac{\left(\overline{x}_1 - \overline{x}_2\right) - \Delta}{\sqrt{\frac{S_1^2}{m} + \frac{S_2^2}{n}}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$	$t > t_{\alpha,v}$	$p = P\left[T \geq t \mid H_0\right]$
H_a: $\mu_1 - \mu_2 < \Delta$	$t < -t_{\alpha,v}$	$p = P\left[T \leq t \mid H_0\right]$
H_a: $\mu_1 - \mu_2 \neq \Delta$	$\lvert t \rvert > t_{\alpha/2,v}$	$p = P\left[\lvert T \rvert \geq t \mid H_0\right]$

Recalling T-tests

$$t = \frac{(\overline{x}_1 - \overline{x}_2) - \Delta}{\frac{\sqrt{S_1^2}}{m} + \frac{\sqrt{S_2^2}}{n}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$	$t > t_{\alpha, v}$	$p = P[T \geq t \mid H_0]$
H_a: $\mu_1 - \mu_2 < \Delta$	$t < -t_{\alpha, v}$	$p = P[T \leq t \mid H_0]$
H_a: $\mu_1 - \mu_2 \neq \Delta$	$\lvert t \rvert > t_{\alpha/2, v}$	$p = P[\lvert T \rvert \geq t \mid H_0]$

$$v = \frac{\left(\frac{S_1^2}{m} + \frac{S_2^2}{n}\right)}{\frac{(S_1^2/m)^2}{m-1} + \frac{(S_2^2/n)^2}{n-1}}$$

Recalling T-tests

$$t = \frac{(\overline{x}_1 - \overline{x}_2) - \Delta}{\frac{\sqrt{S_1^2}}{m} + \frac{\sqrt{S_2^2}}{n}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$	$t > t_{\alpha, v}$	$p = P[T \geq t \mid H_0]$
H_a: $\mu_1 - \mu_2 < \Delta$	$t < -t_{\alpha, v}$	$p = P[T \leq t \mid H_0]$
H_a: $\mu_1 - \mu_2 \neq \Delta$	$\lvert t \rvert > t_{\alpha/2, v}$	$p = P[\lvert T \rvert \geq t \mid H_0]$

$$v = \frac{\left(\frac{S_1^2}{m} + \frac{S_2^2}{n}\right)}{\frac{(S_1^2/m)^2}{m-1} + \frac{(S_2^2/n)^2}{n-1}}$$

Where α captures the level of confidence for a p-value

Recalling T-tests

$$t = \frac{(\overline{x}_1 - \overline{x}_2) - \Delta}{\sqrt{\frac{S_1^2}{m} + \frac{S_2^2}{n}}}$$

Where H_0: $\mu_1 - \mu_2 = \Delta$

H_a: $\mu_1 - \mu_2 > \Delta$	$t > t_{\alpha,v}$	$p = P[T \geq t \mid H_0]$
H_a: $\mu_1 - \mu_2 < \Delta$	$t < -t_{\alpha,v}$	$p = P[T \leq t \mid H_0]$
H_a: $\mu_1 - \mu_2 \neq \Delta$	$\lvert t \rvert > t_{\alpha/2,v}$	$p = P[\lvert T \rvert \geq t \mid H_0]$

$$v = \frac{\left(\frac{S_1^2}{m} + \frac{S_2^2}{n}\right)^2}{\frac{(S_1^2/m)^2}{m-1} + \frac{(S_2^2/n)^2}{n-1}}$$

a p-value

But subtle challenges arise in practice!

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone
  - Is there a specific user segment that it should apply to?
    (Users of features X,Y,Z? Users in a specific country? Early adopters?)

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone
  - Is there a specific user segment that it should apply to?
    (Users of features X,Y,Z? Users in a specific country? Early adopters?)
- The hypothesis might affect different subpopulations differently

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone
  - Is there a specific user segment that it should apply to?
    (Users of features X,Y,Z? Users in a specific country? Early adopters?)
- The hypothesis might affect different subpopulations differently
  - People familiar with workflow X
  - Different age groups
  - People speaking different languages
  - People using the software on different workdays

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone
  - Is there a specific user segment that it should apply to?
    (Users of features X,Y,Z? Users in a specific country? Early adopters?)
- The hypothesis might affect different subpopulations differently
  - People familiar with workflow X
  - Different age groups
  - People speaking different languages
  - People using the software on different workdays
- Possible factors in the results ought to be identified up front.
  Collecting them after the fact requires rerunning an experiment.

Problem: Choosing and tagging populations

- The hypothesis in question may not apply to everyone
  - Is there a specific user segment that it should apply to?
    (Users of features X,Y,Z? Users in a specific country? Early adopters?)
- The hypothesis might affect different subpopulations differently
  - People familiar with workflow X
  - Different age groups
  - People speaking different languages
  - People using the software on different workdays
- Possible factors in the results ought to be identified up front.
  Collecting them after the fact requires rerunning an experiment.
- Your sample ought to be representative.

Problem: False positives and negatives

		Type I error
	$P[\text{fail to reject } H_0 \mid H_0]$	$P[\text{reject } H_0 \mid H_0]$ α
β	$P[\text{fail to reject } H_0 \mid \neg H_0]$	$P[\text{reject } H_0 \mid \neg H_0]$
	Type II error	

There is *always* a risk of error

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives*** (P[reject $\mathbf{H_0}$ | $\mathbf{H_0}$])

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives*** (P[reject $\mathbf{H_0}$ | $\mathbf{H_0}$])

p=P[A sample is at least as extreme as observed | H_0]

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives*** (P[reject $\mathbf{H_0}$ | $\mathbf{H_0}$])
  - The temptation (and management pressure) favors ***p-hacking***

    p=P[A sample is at least as extreme as observed | H_0]

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives***
  - The temptation (and management pressure) favors ***p-hacking***

$$p=P[\text{A sample is at least as extreme as observed} \mid H_0]$$

Suppose you run 5 tests with p=0.1,
What is the likelihood of a false positive?

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives***
  - The temptation (and management pressure) favors ***p-hacking***

$$p=P[\text{A sample is at least as extreme as observed} \mid H_0]$$

Could you correct for this?

Problem: Choosing hypotheses

- Can you simply test any and all hypotheses?
  Can you run your tests and try many hypotheses later?
  - Define clear goals. Hypotheses not targetting goals are useless.
  - Testing many things increases the likelihood of ***false positives***
  - The temptation (and management pressure) favors ***p-hacking***
- The more hypotheses you test, the greater your risk of false positives
  - This can be mitigated, but you should choose hypotheses well up front

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05),
  the size of a test campaign with T-tests must be set up front.

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. $\alpha=0.05$), the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05),
  the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.
  - ***Do not*** just observe the process and stop it "after significance reached"

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05), the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.
  - ***Do not*** just observe the process and stop it “after significance reached”
- But then how many samples are required?

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05), the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.
  - ***Do not*** just observe the process and stop it "after significance reached"
- But then how many samples are required?
  - First determine the acceptable error probabilities, α and β (often 5% & 20%)

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05), the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.
  - ***Do not*** just observe the process and stop it "after significance reached"
- But then how many samples are required?
  - First determine the acceptable error probabilities, α and β (often 5% & 20%)
  - The ***power*** of a test is (1-β). P[reject H_0 | $\neg H_0$]
  - This can also be expressed as "minimum detectable effect size"

Problem: Stopping criteria & confidence

- In order to test with a certain significance (e.g. α=0.05),
  the size of a test campaign with T-tests must be set up front.
  - Calculate the number of samples required first, then run the test.
  - ***Do not*** just observe the process and stop it "after significance reached"
- But then how many samples are required?
  - First determine the acceptable error probabilities, α and β (often 5% & 20%)
  - The ***power*** of a test is (1-β). P[reject H_0 | $\neg H_0$]
  - This can also be expressed as "minimum detectable effect size"
  - If variance and sample sizes can differ, this is challenging,
    so most just use available sample size calculators based on α and β.

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.
- Suppose poorly performing students are put in a special program.

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.
- Suppose poorly performing students are put in a special program.
  - After completion of the program, they perform better.
  - *Is the program effective?*

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.
- Suppose poorly performing students are put in a special program.
  - After completion of the program, they perform better.
  - Is the program effective?
  - If they were already poor performers, improving was more likely anyway!

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.
- Suppose poorly performing students are put in a special program.
  - After completion of the program, they perform better.
  - Is the program effective?
  - If they were already poor performers, improving was more likely anyway!
  - This can be used to falsely justify punishment & rewards

Problem: Regression to the mean

- Following an extreme event, the next event is likely less extreme.
- Suppose poorly performing students are put in a special program.
  - After completion of the program, they perform better.
  - Is the program effective?
  - If they were already poor performers, improving was more likely anyway!
  - This can be used to falsely justify punishment & rewards
- The illusion of significance

Problem: Novelty effects

- Users are used to seeing a blue “buy” button and ignore it, so you change it to red.

Problem: Novelty effects

- Users are used to seeing a blue “buy” button and ignore it, so you change it to red.
  - Sales skyrocket. Red is clearly better!
  - Until a week later when sales return to normal…

Problem: Novelty effects

- Users are used to seeing a blue “buy” button and ignore it, so you change it to red.
  - Sales skyrocket. Red is clearly better!
  - Until a week later when sales return to normal…
- The novelty of the change for the sample may bias the underlying results of the study

Other forms of hypothesis testing

- T-tests are not the only approach and do not always apply

Other forms of hypothesis testing

- T-tests are not the only approach and do not always apply
  - Known variance?
  - Independence?
  - Normality?
  - Qualitative vs Quantitative measures? (does a relationship exist at all?)
  - Small sample sizes expected?
  - …

Other forms of hypothesis testing

- T-tests are not the only approach and do not always apply
  - Known variance?
  - Independence?
  - Normality?
  - Qualitative vs Quantitative measures? (does a relationship exist at all?)
  - Small sample sizes expected?
  - …

If the testing is important,
you should be doing something obvious
or consulting a statistician.

Other forms of hypothesis testing

- T-tests are not the only approach and do not always apply
  - Known variance?
  - Independence?
  - Normality?
  - Qualitative vs Quantitative measures? (does a relationship exist at all?)
  - Small sample sizes expected?
  - …
- But what if even the notion of a predetermined campaign does not fit?

Other forms of hypothesis testing

- T-tests are not the only approach and do not always apply
  - Known variance?
  - Independence?
  - Normality?
  - Qualitative vs Quantitative measures? (does a relationship exist at all?)
  - Small sample sizes expected?
  - …
- But what if even the notion of a predetermined campaign does not fit?
  - Sequential hypothesis testing & Bayesian approaches
  - Bandits

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed

Why might running a t-test be undesirable?

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed
- There may be sufficient evidence to stop the test early
  - Especially when an effect is extreme!

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed
- There may be sufficient evidence to stop the test early
  - Especially when an effect is extreme!
  - ✓✗✗✓✗✗✗✗…

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed
- There may be sufficient evidence to stop the test early
  - Especially when an effect is extreme!
  - ✓✗✗✓✗✗✗✗…

What new problem arises?

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed
- There may be sufficient evidence to stop the test early
  - Especially when an effect is extreme!
  - ✓✗✗✓✗✗✗✗…
  - What are the ***stopping criteria***?
    When is there enough evidence to be convinced?

Sequential Hypothesis Testing

- Consider managing an assembly line
  - Making components for computers
  - Up to 5% of the components can be faulty, otherwise the line should be stopped and inspected/fixed
- There may be sufficient evidence to stop the test early
  - Especially when an effect is extreme!
  - ✓ ✗ ✗ ✓ ✗ ✗ ✗ ✗ …
  - What are the ***stopping criteria***?
    When is there enough evidence to be convinced?
- NOTE: This problem is challenging and is an active area of research
  - We will only look at one approach

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A,B,S_K such that
  - $A < B$

Sequential Hypothesis Testing

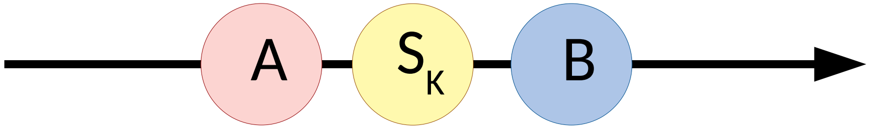
- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop

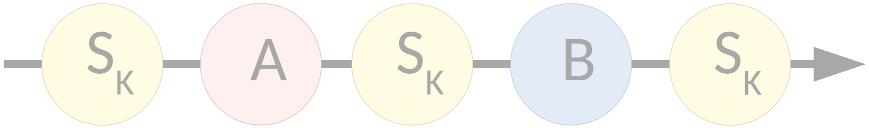
Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling



Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling



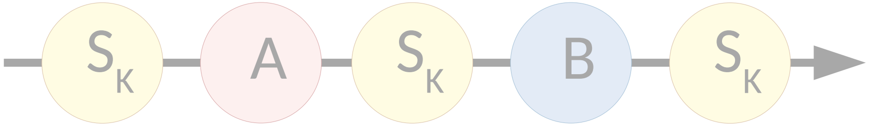
- Done using Wald's Sequential Probability Ratio Test

$$S_K = \log \prod_{i=1}^{K} \frac{p(X_i | H_A)}{p(X_i | H_0)}$$

a ***likelihood ratio test***

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3\ldots X_K$, we want A,B,S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling

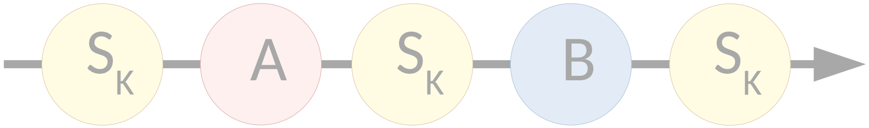


- Done using Wald's Sequential Probability Ratio Test

$$S_K = \log \prod_{i=1}^{K} \frac{p(X_i | H_A)}{p(X_i | H_0)}$$ a ***likelihood ratio test*** $$A = \log \frac{\beta}{1-\alpha} \qquad B = \log \frac{1-\beta}{\alpha}$$

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3\ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling



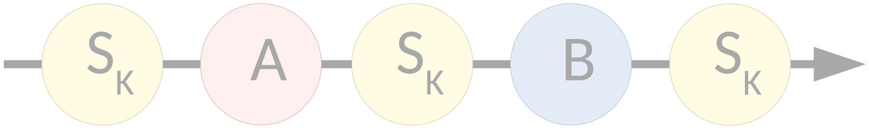
- Done using Wald's Sequential Probability Ratio Test

$$S_K = \log \prod_{i=1}^{K} \frac{p(X_i|H_A)}{p(X_i|H_0)}$$ a ***likelihood ratio test*** $$A = \log \frac{\beta}{1-\alpha} \qquad B = \log \frac{1-\beta}{\alpha}$$

$S_0 = 0$
$S_K = S_{K-1} + \log p(X_K|H_A) - \log p(X_K \mid H_0)$

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling



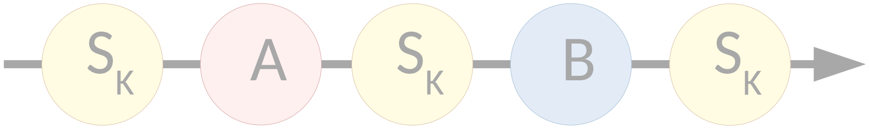
- Done using Wald's Sequential Probability Ratio Test

$$S_K = \log \prod_{i=1}^{K} \frac{p(X_i|H_A)}{p(X_i|H_0)}$$ a ***likelihood ratio test*** $$A = \log \frac{\beta}{1-\alpha} \qquad B = \log \frac{1-\beta}{\alpha}$$

- Caveat/risk:
  - May only be beneficial/useful for simple hypotheses. Otherwise it is complex.

Sequential Hypothesis Testing

- Given a sequence of observations $X_1X_2X_3 \ldots X_K$, we want A, B, S_K such that
  - $A < B$
  - $B < S_K$ $\Rightarrow$ reject H0 and stop
  - $S_K < A$ $\Rightarrow$ fail to reject H0 and stop
  - $A < S_K < B$ $\Rightarrow$ continue sampling



- Done using Wald's Sequential Probability Ratio Test

$$S_K = \log \prod_{i=1}^{K} \frac{p(X_i | H_A)}{p(X_i | H_0)}$$ a ***likelihood ratio test*** $$A = \log \frac{\beta}{1-\alpha} \qquad B = \log \frac{1-\beta}{\alpha}$$

- Caveat/risk:
  - May only be beneficial/useful for simple hypotheses. Otherwise it is complex.
- Simpler approaches exist based on the Gambler's Ruin (w/ no H0 estimate)

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?

Multi-Armed Bandits

- What if we don't really care whether H_0 is false;
  we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***

Multi-Armed Bandits

- What if we don't really care whether H_0 is false;
  we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen

Multi-Armed Bandits

- What if we don't really care whether H_0 is false;
  we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen
  - Want to maximize profit over time

Multi-Armed Bandits

- What if we don't really care whether H_0 is false;
  we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen
  - Want to maximize profit over time
  - Fundamentally choosing between ***exploration*** & ***exploitation***

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen
  - Want to maximize profit over time
  - Fundamentally choosing between ***exploration*** & ***exploitation***
  - We only want to spend enough effort on bad arms to believe they are bad

Multi-Armed Bandits

- What if we don't really care whether H_0 is false;
  we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen
  - Want to maximize profit over time
  - Fundamentally choosing between ***exploration*** & ***exploitation***
  - We only want to spend enough effort on bad arms to believe they are bad

So why might you prefer bandits over A/B tests
(or vice versa)?

Multi-Armed Bandits

- What if we don't really care whether H_0 is false; we just want to make a good choice *now*?
- Given options A, B, C, and D, which is the best to use based on evidence so far?
- This is attacked with ***multi armed bandits***
  - Each arm has an unknown likelihood of paying out when chosen
  - Want to maximize profit over time
  - Fundamentally choosing between ***exploration*** & ***exploitation***
  - We only want to spend enough effort on bad arms to believe they are bad
- Many solutions. Two common ones:
  - **ε**-greedy strategy
  - Thompson sampling

Multi-Armed Bandits

- Usual assumptions
  - Reward probabilities (like conversion rates) don’t change

Multi-Armed Bandits

- Usual assumptions
  - Reward probabilities (like conversion rates) don't change
  - Sampling is singular & instantaneous (choosing a version & its reward)

Multi-Armed Bandits

- Usual assumptions
  - Reward probabilities (like conversion rates) don’t change
  - Sampling is singular & instantaneous (choosing a version & its reward)
  - Samples are independent (i.i.d.)

Multi-Armed Bandits

- Usual assumptions
  - Reward probabilities (like conversion rates) don't change
  - Sampling is singular & instantaneous (choosing a version & its reward)
  - Samples are independent (i.i.d.)
- While solutions can be robust when assumptions are violated, there can be better variants or better solutions

Multi-Armed Bandits: **ε**-Greedy Strategy

- **ε**-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches

Multi-Armed Bandits: ε-Greedy Strategy

- ε-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches

```
on_choice():
  with probability 1-ε:
    pull the best arm so far
  else:
    pull a random arm
  update pulled arm stats
```

Multi-Armed Bandits: ε-Greedy Strategy

- **ε**-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches

```
on_choice():
  with probability 1-ε:
    pull the best arm so far
  else:
    pull a random arm
  update pulled arm stats
```

Multi-Armed Bandits: **ε**-Greedy Strategy

- **ε**-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches
  - Choosing **ε**
    - A higher **ε** favors exploration.
    - Lower **ε** favors exploitation.
    - 0.1 is common

```
on_choice():
  with probability 1-ε:
    pull the best arm so far
  else:
    pull a random arm
  update pulled arm stats
```

Multi-Armed Bandits: ε-Greedy Strategy

- ε-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches
  - Choosing ε
    - A higher ε favors exploration.
    - Lower ε favors exploitation.
    - 0.1 is common
- Can also vary/scale ε over time.
  - Can be used to logarithmically bound regret by limiting future exploration (decay)

```
on_choice():
  with probability 1-ε:
    pull the best arm so far
  else:
    pull a random arm
  update pulled arm stats
```

Multi-Armed Bandits: ε-Greedy Strategy

- ε-greedy strategy
  - Has the benefit of being dead simple
  - May be too sensitive to variance and perform worse than other approaches
  - Choosing ε
    - A higher ε favors exploration.
    - Lower ε favors exploitation.
    - 0.1 is common
- Can also vary/scale ε over time.
  - Can be used to logarithmically bound regret by limiting future exploration (decay)
- Feels a bit ad hoc. Why would you use it?

```
on_choice():
  with probability 1-ε:
    pull the best arm so far
  else:
    pull a random arm
  update pulled arm stats
```

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm

```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Multi-Armed Bandits: Thompson Sampling

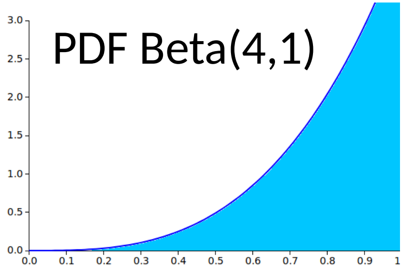
- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm

```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm

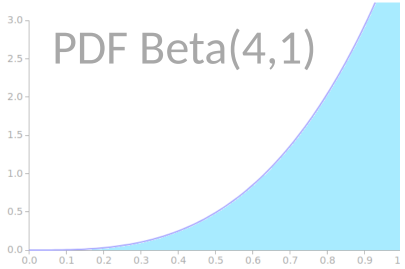


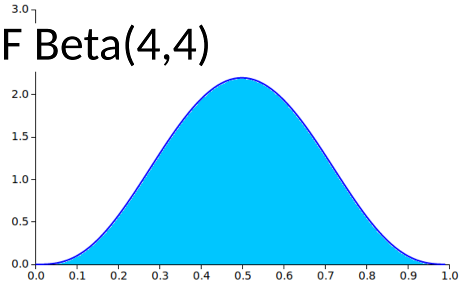

```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm



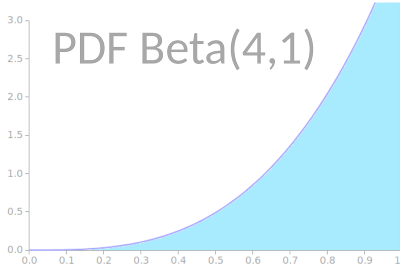


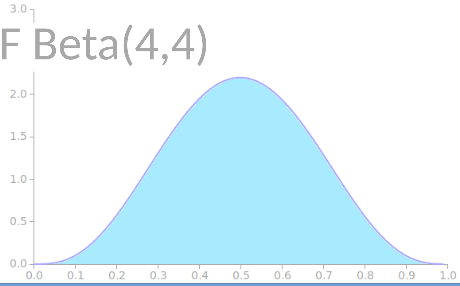

```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

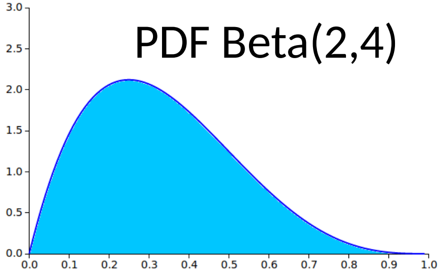
```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm





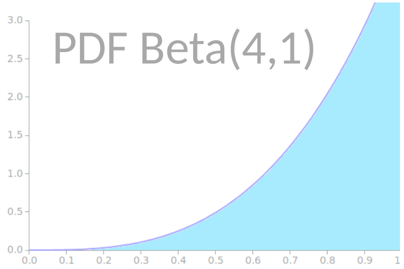


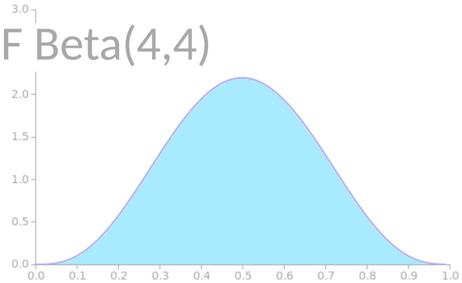

```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

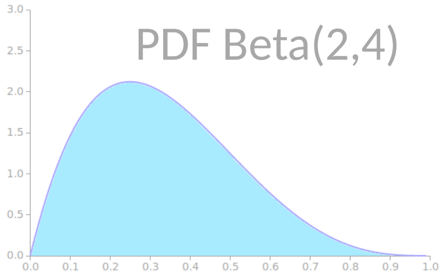
```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Multi-Armed Bandits: Thompson Sampling

- Thompson sampling
  - Tends to behave well with delayed feedback
  - Choose each arm based on the probability of being the best arm








```
initialize():
  for each arm i:
    failures[i] = 0
    successes[i] = 0
```

```
on_choice():
  for each arm i:
    sample from Beta(successes[i]+1,failures[i]+1)
  select argmax_i samples[i]
  update successes and failures for i
```

Contextual Bandits

- What if the reward likelihood depends on
  - History
  - Environmental state

Contextual Bandits

- What if the reward likelihood depends on
  - History
  - Environmental state
- *Contextual* Bandits are able to take features at time t into account

Other uses of bandits in software quality

- Fuzz testing

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization
  - Finding optimal configurations for cloud workloads

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization
  - Finding optimal configurations for cloud workloads
  - Command line options for compilers to improve performance

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization
  - Finding optimal configurations for cloud workloads
  - Command line options for compilers to improve performance
  - Fine tuning for databases

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization
  - Finding optimal configurations for cloud workloads
  - Command line options for compilers to improve performance
  - Fine tuning for databases
  - Hyperparameter tuning in machine learning
  - …

Other uses of bandits in software quality

- Fuzz testing
- Auto configuration / optimization
  - Finding optimal configurations for cloud workloads
  - Command line options for compilers to improve performance
  - Fine tuning for databases
  - Hyperparameter tuning in machine learning
  - …
- Verification & cryptanalysis
- …

Choosing a solution

- A/B Testing
  - Can be robust as long as the sample is representative
- Bandits
  - Allow you to take advantage of results as they find the solution
  - Can enable adaptation over time rather than one shot optimality

Summary: A/B Testing & Bandits

- Hypothesis testing can help you choose one version of something over another
- Sequential strategies can allow for early stopping & peeking
- Bandit based techniques allow for optimizing expected benefit while exploring options