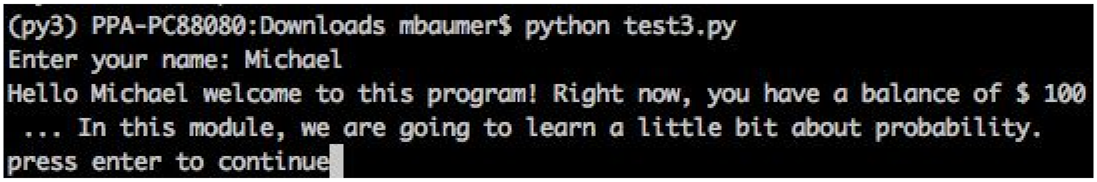
DIY P-Hacking (D.A.T.A) Final Report

Mike Baumer, Dakin Henderson, and Emma Lejeune

DESIGN SUMMARY

The overall goal of our design approach was to teach students in EDUC200A how to assess education research they might encounter in their future careers as education practitioners who are **consumers of education research**, rather than statistics experts.
We aimed to motivate students and encourage a sense of learner belonging and self-efficacy by transforming the traditional mathematical presentation of statistics to one centered around intuition. To do this, we developed a module consisting of four "phases", each of which focused on a different component learning goal:

Phase 1: Expected Value Casino
Learning Goal: **Compute** expected values and use them to **make decisions** in simple situations.

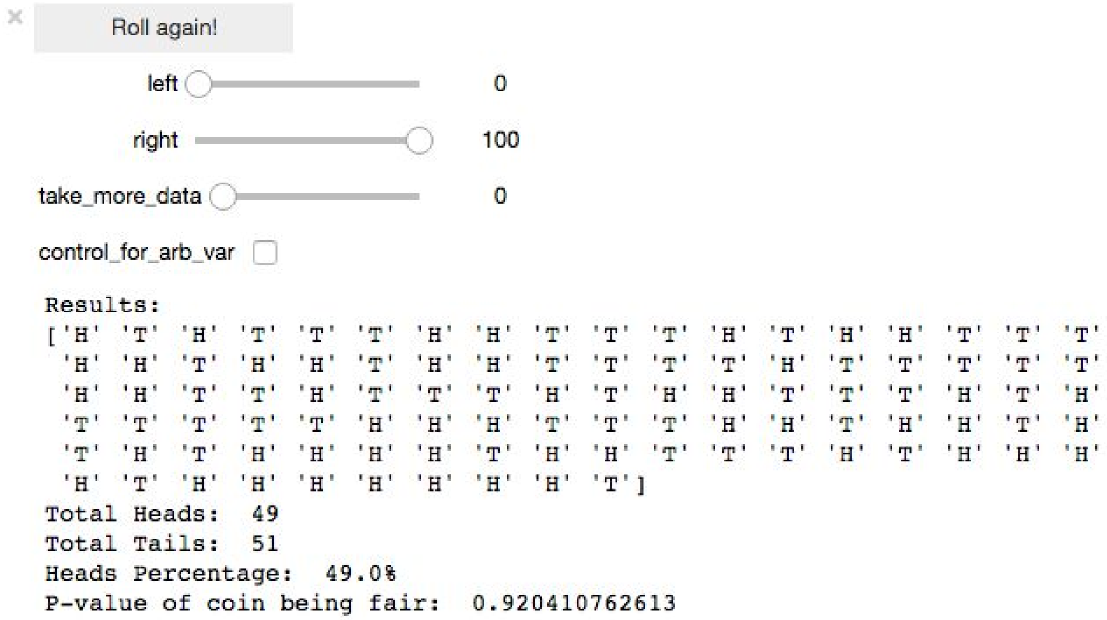


```
(py3) PPA-PC88080:Downloads mbaumer$ python test3.py
Enter your name: Michael
Hello Michael welcome to this program! Right now, you have a balance of $ 100
 ... In this module, we are going to learn a little bit about probability.
press enter to continue
```

This game puts the user in a casino, where the tool of expected value can be exploited to move on to the next stage.

Phase 2: P-values at the Casino
Learning Goal: **Interpret** p-values in terms of the intuitive feeling of surprise relative to expectation.
Preceded by some questions regarding intuition in the openEdX, this module continues the casino game started in Phase 1, although winning money in this stage requires using interpreting p-values to know when coins on which the learner is betting are likely fair or unfair.

Phase 3: P-hacking at the Casino
Learning Goal: **Manipulate** parameters of a synthetic study to produce "p-hacked" results.



left 0

right 100

take_more_data 0

control_for_arb_var □

```
Results:
['H' 'T' 'H' 'T' 'T' 'T' 'H' 'H' 'T' 'T' 'T' 'H' 'T' 'H' 'H' 'T' 'T' 'T'
 'H' 'H' 'T' 'H' 'H' 'T' 'H' 'H' 'T' 'T' 'T' 'T' 'H' 'T' 'T' 'T' 'T' 'T'
 'H' 'H' 'T' 'T' 'H' 'T' 'T' 'T' 'H' 'T' 'H' 'H' 'T' 'T' 'T' 'H' 'T' 'H'
 'T' 'T' 'T' 'T' 'T' 'H' 'H' 'H' 'T' 'T' 'T' 'H' 'H' 'T' 'H' 'H' 'T' 'H'
 'T' 'H' 'T' 'H' 'H' 'H' 'H' 'T' 'H' 'H' 'T' 'T' 'T' 'H' 'T' 'H' 'H' 'H'
 'H' 'T' 'H' 'H' 'H' 'H' 'H' 'H' 'H' 'T']
Total Heads:  49
Total Tails:  51
Heads Percentage:  49.0%
P-value of coin being fair:  0.920410762613
```

This game, which would eventually be integrated with the monetary scorekeeping system of Phases 1 and 2, puts the learner in the shoes of the casino owner, who must find ways to manipulate coin-flip data to convince unsuspecting casino-goers that his coins are unfair when they really aren't!

Phase 4: Connecting the Casino to real life: P-hacking in social science
Learning Goal: **Explain** the relationship of "p-hacking" to good study design.
This openEdX module explores the relationship of the previous three Phases to interpreting statistics and p-values in the context of social science research.

In the full module we envision for the future, these modules would not only be unified into a single interface, but also be followed by a module where learners apply these concepts to interpret text from real scientific studies to assess their credibility.

FOUR DESIGN PRINCIPLES

Game design
In our module, we take advantage of the core engagement loop to keep our players engaged and motivated to continue progressing with the module. At the start of the module, we give our users $100 (in fake virtual currency) and tell them that to progress from Phase 1 to Phase 2 they need to turn that money into $1000, and then to progress from Phase 2 to Phase 3 they need to turn their balance from Phase 1 into $5000. The user can make money by making bets. We chose to do this following the logic of a "core engagement loop" by allowing the user to

decide if they want to take a bet, or given a set of bets which one they choose to engage in, and then act. Specifically, the core engagement loop is as follows:

Assess	What is the user’s current balance? How close are they to advancing to the next level? What is the EV of the bet(s) on offer? If there are multiple options, which one has the highest EV? If there is only one option, is it likely that the EV is > 0?
Decide	The user chooses a bet based on what they learned from assessing the situation
Act	The user hits <enter> to flip a coin or roll a die, either one time or multiple times depending on the scenario.
Reward	If the user made a good choice, they will make $ and see their balance increase, getting them closer to the next level. If the user made a bad choice, they will see the $ decrease, if the user chooses not to bet their balance will stay the same. Hopefully both not making $ or losing $ will motivate the user to try again and try to make money rather than causing them to feel so discouraged that they quit. When the user loses money, we immediately review the concepts needed to make the correct choice, which will hopefully inspire them to continue playing, and there is no additional penalty for losing money or having a balance go below $0.

We structure each of the betting scenarios such that if the user understands the basic principles of probability (computing expected value) and/or hypothesis testing they will, on average, make money from betting. We hope that the desire to make money within the game will motivate the user to learn how to make the correct bets and that repeatedly working through the core engagement loop will facilitate deliberate practice.

Generation and decision making

Prior to game play, we have the users answer diagnostic questions on the open edX platform. The purpose of these questions is to evaluate the level of the player. However, during the actual game play, we require the user to answer questions for a different reason. During gameplay, we require the user to calculate values and make decisions not only to track their performance, but also to lead them towards "generation". Research in the educational literature, ex: "G is for generation" chapter in the “The ABCs of How We Learn: 26 Scientifically Proven Approaches, How They Work, and When to Use Them”, claims that people are better able to remember what they have learned if they are required to "generate" something as a part of the learning process. When we ask the user to input expected value, the user must perform the calculation for expected value, either entirely from memory or from the formulas provided. Furthermore, we repeatedly ask the users to decided if they want to take a bet, or given a set of bets which one they want to choose. The justification for this structure is that the user will learn more if they are actively engaged and making choices rather than passively being fed material.

Scaffolding/Zone of Proximal Development

We provide scaffolding in the form of worked examples and problems/topics that conceptually build on each other. For example, a learner must understand how to determine or compute what they expect to happen (Phase 1), before understanding a p-value, which of course must be expressed relative to an expectation (Phase 2). In addition to conceptual scaffolding, we provide worked examples as part of the games in Phases 1 and 2, and also include worked examples in the openEdX module, for example in the solution that appears to the EV diagnostic question (it is displayed to all students regardless of whether or not students earn the right to skip past the Phase 1 EV game).

Scaffolding is of course related to the concept of keeping learners in the "zone of proximal development", or state of flow. Too little scaffolding, and the student's ability to complete the task will be so low that they become overwhelmed and discouraged. The danger of too much mandatory scaffolding, however, is that students who have seen some of the material before may become bored if they are forced to complete tasks at which they are already highly competent. We employ diagnostic questions to assess students' prior knowledge, and allow them to skip the expected value module if they demonstrate the competency expressed in the learning goal of that phase. Even within the betting games, the propositions get increasingly difficult to evaluate as the learner successfully progresses through the module, with a level of challenge that scales with student development

Feedback

The final principle we employed was feedback, which ties into our assessment strategy.

Within the betting games, students receive timely feedback in the form of monetary gain or loss. If the learner makes a bad bet, in addition to losing money, they are presented with targeted feedback reviewing the mistakes they may have made in their expected value calculation. For the selected response EV diagnostic question, the distractors are chosen based on anticipated common student errors, with targeted feedback delivered after an incorrect answer hinting at the error that likely led students to the distractor.

In general, we have restricted our assessments to selected-response and quantitative constructed-response, to allow for automatically generated assessment, and corresponding timely, targeted feedback.

TARGET PRINCIPLE:

"Generation and decision making" is very important to the structure of our module. As a self guided online learning module we wanted to engage the user every step of the way and avoid feeling like a textbook or lecture video where the user is just fed information and must passively absorb it. We anticipate that with and without applying this principal student interaction is as follows:

	Mechanics of interaction	Outcomes of interaction

<table>
<tr><td colspan="3">Example 1: calculating expected value</td></tr>
<tr><td>With target principle</td><td>After a brief worked example detailing how to compute expected value, students are asked to compute the expected value and enter the number.</td><td>Through this process the students must learn how to correctly compute expected value, if they don’t get the correct value, they must go through the worked example again step by step</td></tr>
<tr><td>Without target principle</td><td>After a brief worked example detailing how to compute expected value, it is assumed that students know how to do it and they are not required to do it until either much later in the assignment or during a follow up formal evaluation period</td><td>Students may learn how to compute expected value, but without the timely requirement to “generate” they will most likely, on average, learn less and/or make mistakes when asked to do it in the future.</td></tr>
<tr><td colspan="3">Example 2: choose a game to play/ accept or reject a game</td></tr>
<tr><td>With target principle</td><td>After being told that the best bet to pick is the one associated with the highest expected value, students are asked to pick the bet that they think is the best option. Ideally (if the student takes the module seriously) this will cause them to go through the process for computing expected value and/or thinking about what a p-value and null hypothesis means and analyzing prior coin flip results before they make their decision.</td><td>When students make the right choice, they are rewarded by on average making more money from the betting scheme. Because they have agency and have to think through their decisions and recall the important steps in decision making students are more likely to acquire and remember the skills needed to make bets. Furthermore, if they don’t pick the correct bet then they automatically are exposed to additional review before moving on.</td></tr>
<tr><td>Without target principle</td><td>After being told that the best bet to pick is the one associated with the highest expected value, the module will automatically select the best bet to take and play it without any student input.</td><td>Students may learn how to make the right choice and evaluate p-values bet without the timely requirement to “generate” they will most likely, on average, learn less and/or make mistakes when asked to draw on these skills in the future.</td></tr>
</table>

PROTOTYPE FEEDBACK

Expected Value Diagnostic Question (openEdX)

The users--none of whom identified themselves as “math people”--were not able to answer the diagnostic question. The cost of the game seemed to throw them off, one person thought it was a “trick question.” Users also were wondering how many times the coin will be flipped, and were confused that it was irrelevant to the question.

In conclusion, the diagnostic question seemed to do its job, in that these users needed some brushing up on expected values.

Phase 1: Expected Value at the Casino

Users answered the first question--about whether or not to take the bet--intuitively. The values were so obviously in favor of the person betting that they didn’t need to calculate using any formulas. This was successful. One user: “I didn’t compute the expected value, I just did it intuitively. I knew that if we’re gonna get $10, we can lose like 9 times and it’ll be fine.”

The transition to the worked example which introduces them to the equation seemed to throw users off. They were expecting to play another betting game, but instead the game took a different tack. One user responded to the sentence about pi x vi: “annnnnnnnnd I’m done playing.” They suggested a metacognitive step where the game explicitly states that we’re going into a worked example, and going to tell you how to calculate expected value, etc.

Specifically referring to the worked example which breaks down the equation, there was some confusion around why you subtract the cost of the game in *both* parts of the equation, as opposed to at the end. As in, (0.5)(10-1) + (0.5)(1-1) as opposed to (0.5)(10) + (0.5)(1) - 1. Of course, the answer is the same, but this was not clear to users.

It was also suggested that the worked example somehow be interactive. Like, have the user plug in the values for the bet they just made. Just a thought.

Phase 3: P-Hacking at the Casino

The intro to this game in OpenEdX needs work. Users weren’t sure what the goal was, or the reason they were playing it. The takeaway message could hit home a little better.

There should be a way to track what users are doing while they play the game. Some cumulative representation of all the flips and tweaks they’ve made. This would help for continuity, as well as the takeaway lesson. When users win, and achieve a significant result, it would be good to then zoom out and say “sorry to burst your bubble, but if you take ALL the coin flips you made, the p value is actually X…”

Suggestion: it would be interesting to let a script find the lowest possible p-value with the given values, and see if the point is made any better that way, instead of letting users do it themselves.

If you’ve cut off some of the right side of the data, and then you use the add more data function, is it adding new flips, or is it representing the same flips that were just cut off? It should add new flips, but this could be made clearer to the user. We could try using a button to add 10 new flips, instead of a slider.

When you leave “control for arbitrary variable” checked, the p value jumps all over the place because it’s taking a new random subgroup every time. The control for arbitrary variable function seems weird. It shouldn’t generate new data within the data set, it should just eliminate random data from the current data set. Which means, when you uncheck the box, it should just put the same original data back.

Many data sets were challenging, or perhaps impossible, to achieve $p<.05$ by manipulating the tails or controlling for an arbitrary variable. These moments were frustrating to users. They felt helpless. We had to suggest to them to use the “roll again” function. Several times, the “roll again” function led to a significant result with no manipulating of the sample necessary. This was not as satisfying for the users. Is there a way to make it easier to achieve a significant result *without* rolling again? Perhaps we could start with a much larger sample size, so there’s more room to play in. If it’s still challenging, we could allow for 2 subgroups within the data, instead of one.