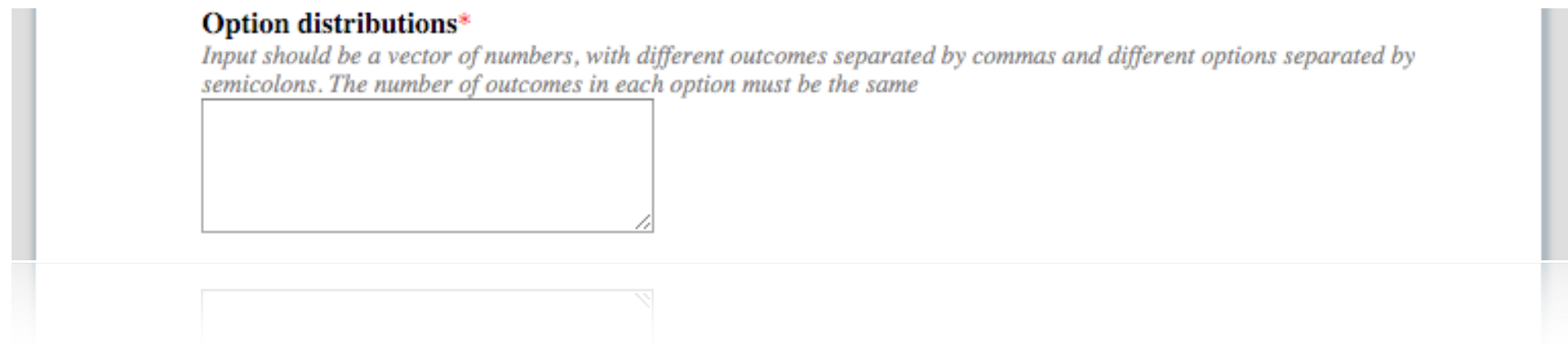


Dynamic peeks and
keeps

In the previous versions of the game, the administrator had to specify every outcome for every option explicitly in the Option Distribution field



The screenshot shows a web interface with a section titled "Option distributions*" in bold. Below the title is a line of instructional text: "Input should be a vector of numbers, with different outcomes separated by commas and different options separated by semicolons. The number of outcomes in each option must be the same". Below this text is a large, empty rectangular input box with a small cursor icon in the bottom right corner. The entire section is framed by a light gray border.

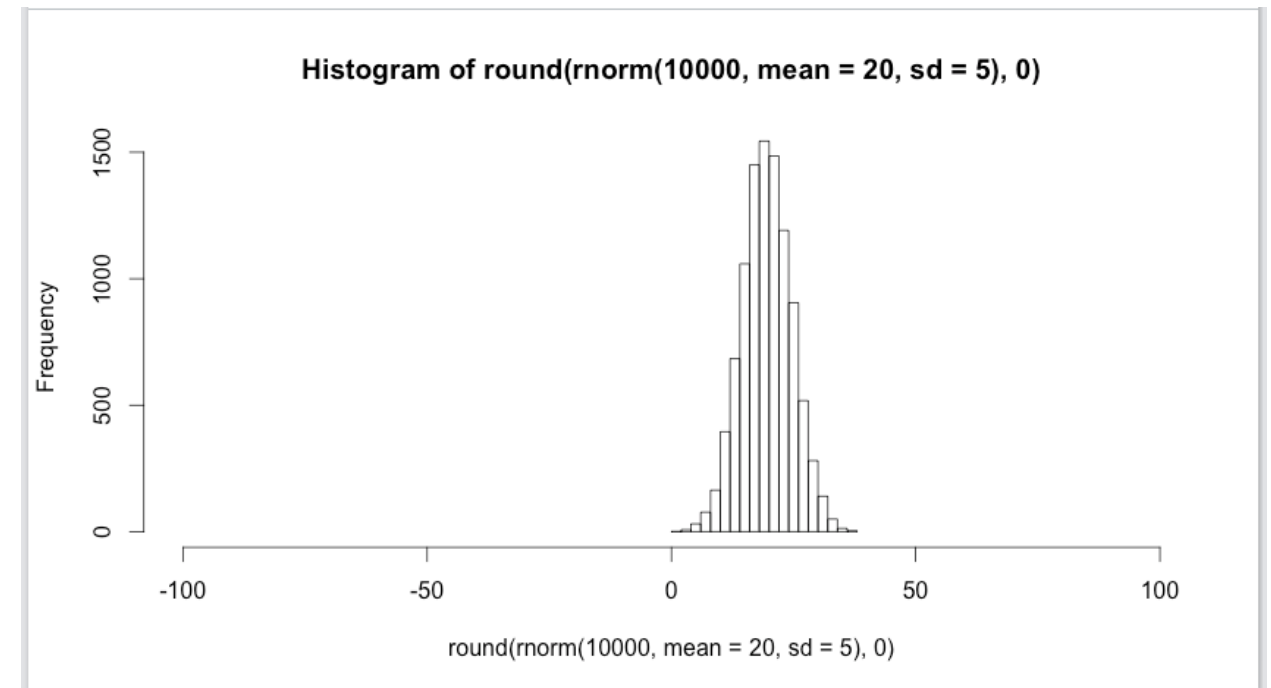
In the dynamic version of the game, we'd like the program to generate the outcomes for each option based on certain rules.

To start, we'll have the program generate option outcomes from Normal Distributions.

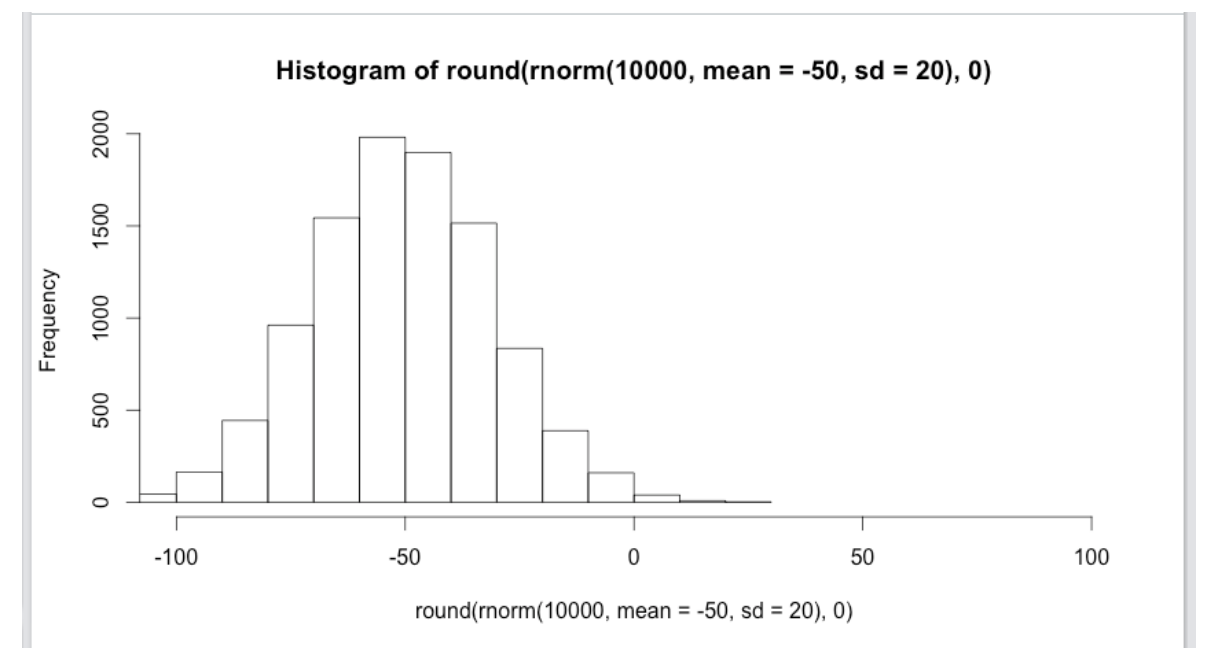
The Normal Distribution has two parameters, a mean and a standard deviation.

In R, you can generate samples from a normal distribution with the command `rnorm(n, mean, sd)`. I don't know how to do this in JavaScript, but I'm sure you'll find a way.

For example, here is 10000 samples from a normal distribution with mean = 0 and sd = 5.



For example, here are 10000 samples from a normal distribution with mean = -50 and sd = 20.



In this new version of the game, the administrator will specify the starting mean and standard deviation of each option in the game.

However, the means of the options will also need to change on EACH trial depending on which option the participant selected on the previous trial. (the option standard deviations do not change at all).

The administrator will specify how the means of the options changed with two new administrator parameters [SELECT.MEAN.CHANGE] and [NOSELECT.MEAN.CHANGE].

If an option is selected on trial i , then its mean on trial $i + 1$ will be its previous mean + SELECT.MEAN.CHANGE

If an option is NOT selected on trial i , then its mean on trial $i + 1$ will be its previous mean + NOSELECT.MEAN.CHANGE.

Given these rules, the server will need to calculate the mean of each option on each trial based on these parameters.

Old Version

These three
inputs are
deleted

And replaced by
four new inputs...

The Konstanz Game Administration

***Required**

Game Name*

Type*

- ☐ Peeks and Keeps
☐ Only Keeps

Trials*

Starting points*

Click delay (in seconds eg. 1000 for a second)*

Specify the number of milliseconds that outcomes will be displayed. The user must wait until the display clears before making another selection

Option distributions*

Input should be a vector of numbers, with different outcomes separated by commas and different options separated by semicolons. The number of outcomes in each option must be the same

Randomize option location*

- ☐ Yes, randomize the location of options for each participant
☐ No, don't change the location of options for any participant

Outcome order*

- ☐ Random: Outcomes is randomly selected with replacement from all outcomes in the current block
☐ Sequential: Outcomes are drawn sequentially from the option distributions. When this option is selected, blocks are ignored

Option distribution blocks*

Input a vector with the same dimensions as the option distributions. Note: Specifying the blocks only changes the game when outcome order is Random

- ☐ One block

Outcome display*

- ☐ All: Full feedback condition. Outcomes from all options are displayed
☐ Selected: Partial feedback. Only outcomes from selected options are displayed

Peek and Keep order*

- ☐ Free alternation: Players can freely alternate between peeking and keeping
☐ Peek then Keep: Players can only begin keeping after they are finished peeking

Number of Peeks*

Players must peek exactly this many times. This number must be less than the total number of trials in the game

- ☐ No limit ☐ Fixed number

Submit

CC BY-NC-SA



Option start means

Specify the mean of each option. E.g.; [5, 5, -5] means that there are three options with means of 5, 5, and -5 respectively



Option standard deviations

Specify the mean of each option. E.g.; [5, 5, -5] means that there are three options with means of 5, 5, and -5 respectively



In these two inputs, the administrator determines the starting mean and standard deviation of each option.

For example, option start means = [5, 2, -3] means that there are three options where the first option has a starting mean of 5, the second has a starting mean of 2, and the third has a starting mean of -3.

Similarly, option standard deviations = [2, 10, 5] means that the three options have a standard deviation of 2, 10, and 5 respectively.

The number of inputs to these two fields must be the same!

For example, let's say that
`OPTION.START.MEANS = [2, 4, -3]`

On Trial 1, the means of the options
are just `OPTION.START.MEANS`

Option Means

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2	4	3
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Here is now the server
would store the means
for each option for
each trial

Example inputs:

SELECT.MEAN.CHANGE = $-.1$

NO.SELECT.MEAN.CHANGE = $.2$

If on trial 1, the player selects option A, then the server looks up the mean of option A on trial 1 and sees that the mean of the option is 2. The server then shows the player sees a random sample from a Normal distribution with mean = 2 (and sd = 2). Then, for Trial 2, the server calculates the means of all options as follows: the mean of option A decreases by $-.1$ (SELECT.MEAN.CHANGE). and the mean of options B and C increase by $.20$ (NOSELECT.MEAN.CHANGE)

Option Means

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2	4	3
<input type="checkbox"/>	↓ 1.9	↑ 4.2	↑ 3.2
<input type="checkbox"/>			
<input type="checkbox"/>			

TRIAL 1: Player selects A and sees a random sample from $N(\text{mean} = 2, \text{sd} = 10)$

The server then determines the means of all options for trial 2 based on the means of the options at trial 1 and which option the player selected on trial 1

Now on Trial 2, If the player now selects option B, the server sees that the mean of option B on trial 2 is 4.2. It will then draw a random sample from a Normal distribution with mean = 4.2 (and sd = 10) and show it to the participant.

The server then updates the option means for trial 3 based on the same rule as before.

Option Means

	□	□	□
□	2	4	3
□	↓ 1.9	↑ 4.2	↑ 3.2
□	↑ 2.1	↓ 4.1	↑ 3.4
□			

TRIAL 2: Player selects B and sees a random sample from $N(\text{mean} = 4.2, \text{sd} = 10)$

The means of all options then change on Trial 3.

This updating process will continue throughout the game.

Option Means

	□	□	□
□	2	4	3
□	↓ 1.9	↑ 4.2	↑ 3.2
□	↑ 2.1	↓ 4.1	↑ 3.4
□	↓ 2.3	↑ 4.3	↑ 3.6
□			
▣▣	2.5	-2	10
▣▣▣	↑ 2.7	↑ -1.8	↓ 9.9

At the end of the game (let's say 100 trials) the table of option means could look like this (with circles indicating which option the participant selected).

In this example, on the last trial (trial 100), the means of options A B and C are changed to 2.7, -1.8, 9.0 (based on all the participant's selections on trials 1 through 99)

The server should continue to record all the basic information from the previous game version (e.g.; choices, outcomes etc.). However, instead of keeping a table of all option outcomes initially specified by the administrator, it will now need to update a table of option means that change over the course of a game.

CHOICES	OUTCOMES	MODE	OPTION.MEANS		
<div><div></div><div>A</div><div>A</div><div>B</div><div>C</div><div></div><div>B</div><div>B</div></div>	<div><div></div>4<div>9</div><div>-5</div><div>-2</div><div></div><div>5</div><div>2</div></div>	<div><div></div>Peek<div>Keep</div><div>Peek</div><div>Peek</div><div></div><div>Keep</div><div>Keep</div></div>	<div><div></div>2<div>1.9</div><div>2.1</div><div>2.3</div><div></div><div>2.5</div><div>2.7</div></div>	<div><div></div>4<div>4.2</div><div>4.1</div><div>4.3</div><div></div><div>-2</div><div>-1.8</div></div>	<div><div></div>3<div>3.2</div><div>3.4</div><div>3.6</div><div></div><div>10</div><div>9.9</div></div>