

Day 2: Online experiments

Amy Perfors

DAY 2: EXPERIMENTS

Tentative plan

1. Experiment logic, motivation, and design
2. R basics for coding: branching, functions, lists
3. Creating a template experiment with jaysire and putting it online
4. Making a more complex experiment
5. Pre-registration and ethics

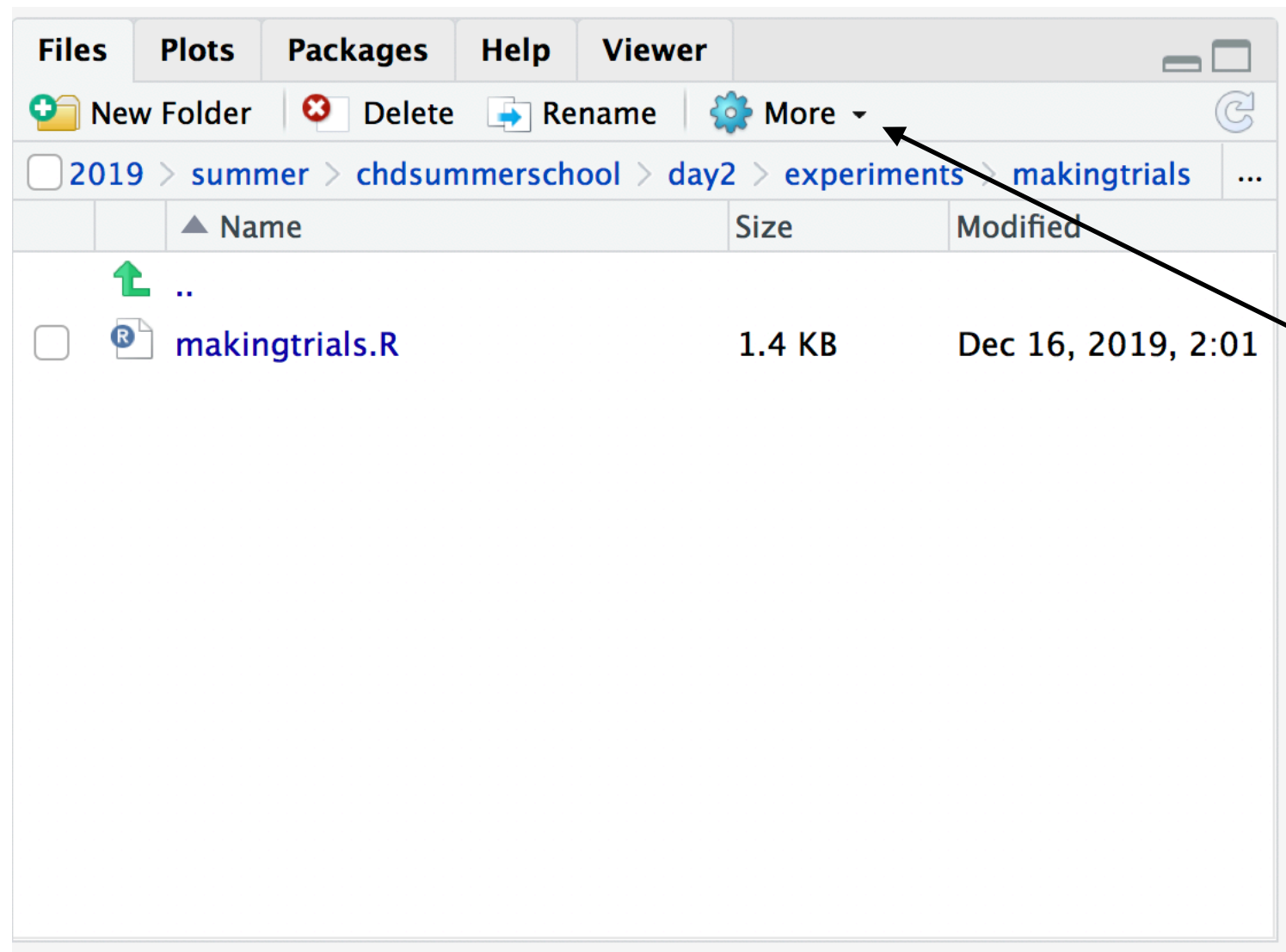
DAY 2: EXPERIMENTS

Tentative plan

1. Experiment logic, motivation, and design
2. R basics for coding: branching, functions, lists
3. Creating a template experiment with jaysire and putting it online
4. Making a more complex experiment
5. Pre-registration and ethics

LET'S GO TO A NEW DIRECTORY WITH A NEW EXPERIMENT

Navigate yourself to a new folder, the `makingtrials` one -
we'll put our new experiment there.



Remember to
set it as your
working
directory

LET'S GO TO A NEW DIRECTORY WITH A NEW EXPERIMENT

Click on makingtrials.R to open it up in RStudio

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Displays the `makingtrials.R` script. The code includes a library call, directory setup, and instructions for a trial.
- Environment Pane:** Shows the Global Environment with a table of objects.
- Files Pane:** Shows the file structure of the project, including the `makingtrials.R` file.
- Console:** Shows the output of the `run_appengine` command and the `pwd` command.

Source Editor Code:

```
1 library(jaysire)
2
3 # set directory (deletes any existing old experiment builds in it)
4 my_directory <- file.path("adv_exp")
5 # create the empty folder if necessary
6 if(dir.exists(my_directory)) {
7   unlink(my_directory, recursive = TRUE)
8 }
9
10 # ----- insert your stimuli here -----
11
12
13 # ----- instructions -----
14 instructions <- trial_instructions(
15   pages = c(
16     "Welcome! Use the arrow buttons to browse these instructions",
17     "Your task is to decide if an equation like '2 + 2 = 4' is true",
18     "You will respond by clicking a button",
19     "Press the 'Next' button to begin!"
20   )
21 )
22
23 # ----- insert your timeline here -----
```

Environment Pane Table:

Name	Type	Length	Size	Value
test3_trials	list	3	7.6 KB	List of 3
tests	list	16	4 KB	List of 16
train1	list	9	2.4 KB	List of 9
train1_files	character	2	200 B	chr [1:2] "data1_cate...
train1_tria...	timeline	2	3.9 KB	List of 2
train2	list	9	2.4 KB	List of 9
train2_files	character	4	360 B	chr [1:4] "data3_cate...
train2_tria...	timeline	2	4.7 KB	List of 2
train3	list	9	2.4 KB	List of 9
train3_files	character	6	520 B	chr [1:6] "data1b_cat...
train3_tria...	timeline	2	5.5 KB	List of 2

Files Pane:

- 2019 > summer > chdsummerschool > day2 > experiments > makingtrials
- makingtrials.R (1.1 KB, Dec 16, 2019, 2:07)

Console:

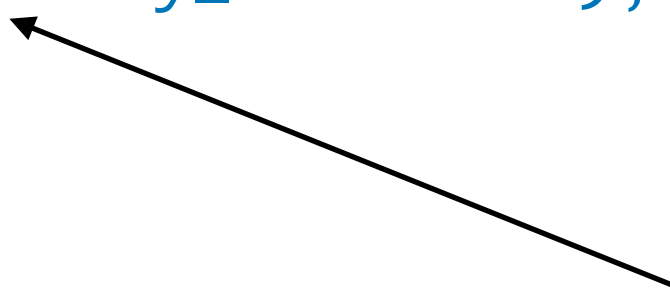
```
> run_appengine(my_directory, chdss-expt)
To deploy, enter the following command at the terminal:
gcloud app deploy exp/experiment/app.yaml --project=chdss-expt
> source('~/.Documents/teaching/2019/summer/chdsummerschool/samplingframes/
samplingframes.R')
> pwd
Error: object 'pwd' not found
>
>
>
```

SETS OF TRIALS

Now instead of defining each trial individually, let's define a set of trials and let R randomise them

We do this using a timeline variable instead of defining the stimulus directly

```
# ----- insert your trial template here -----  
trial_template <- trial_html_button_response(  
  stimulus = insert_variable(name = "my_stimulus"),  
  choices = c("true", "false"),  
  post_trial_gap = 1000  
)
```

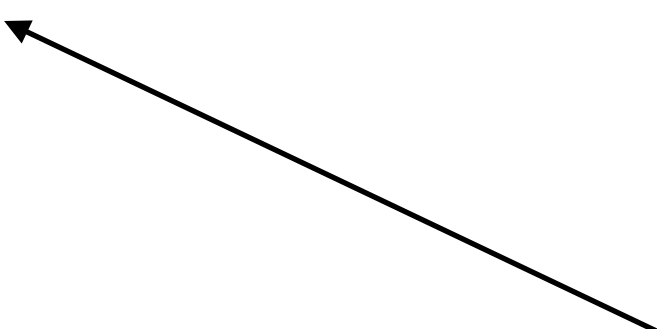


Same as before except this is a function which tells R to insert a variable stimulus that we tell it goes by the name my_stimulus (can be anything)

SETS OF TRIALS

Now we have to define what possible values the variables can have

```
# ----- insert your stimuli here -----  
equations <- c(  
  "13 + 23 = 36",  
  "17 - 9 = 6",  
  "125 / 5 = 25",  
  "2 - 4 = 6",  
  "12 + 39 = 43",  
  "4 * 23 = 92"  
)
```



There are six trials, each corresponding to an equation

SETS OF TRIALS

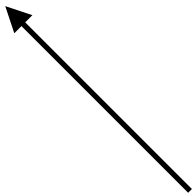
That's nice, but how does R know that equations has anything to do with `my_stimulus`?

It doesn't; we have to tell it:

```
# ----- insert your timeline here -----  
trials <- build_timeline(trial_template) %>%  
  tl_add_variables(my_stimulus = equations)
```



Telling R that the equations are the stimuli for those trials



This is a pipe, it basically means to do the next thing in line with the output of the first

So we've built a timeline of trials of the form `trial_template` and made it so the stimuli for those trials are the equations. This entire thing is saved as its own variable called `trials`

SETS OF TRIALS

Make sure you have your code to build the experiment now:

```
# ----- build the experiment -----  
build_experiment(  
    timeline = build_timeline(instructions, trials, finish),  
    path = my_directory,  
    on_finish = fn_save_locally()  
)
```

Add this bit

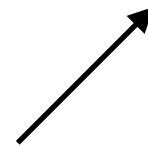


Source, and run!

RANDOMISATION

That didn't put the trials in random order though. To do this we use another timeline function:

```
# ----- insert your timeline here -----  
trials <- build_timeline(trial_template) %>%  
  tl_add_variables(stimulus = equations) %>%  
  tl_add_parameters(randomize_order = TRUE)
```



Adds parameters of different
sorts to the timeline

EXERCISE

Change your code so instead of presenting people with six equations and getting true/false judgments, it presents people with the following four sentences and gets ratings of grammatical acceptability from 1 to 7 with a slider

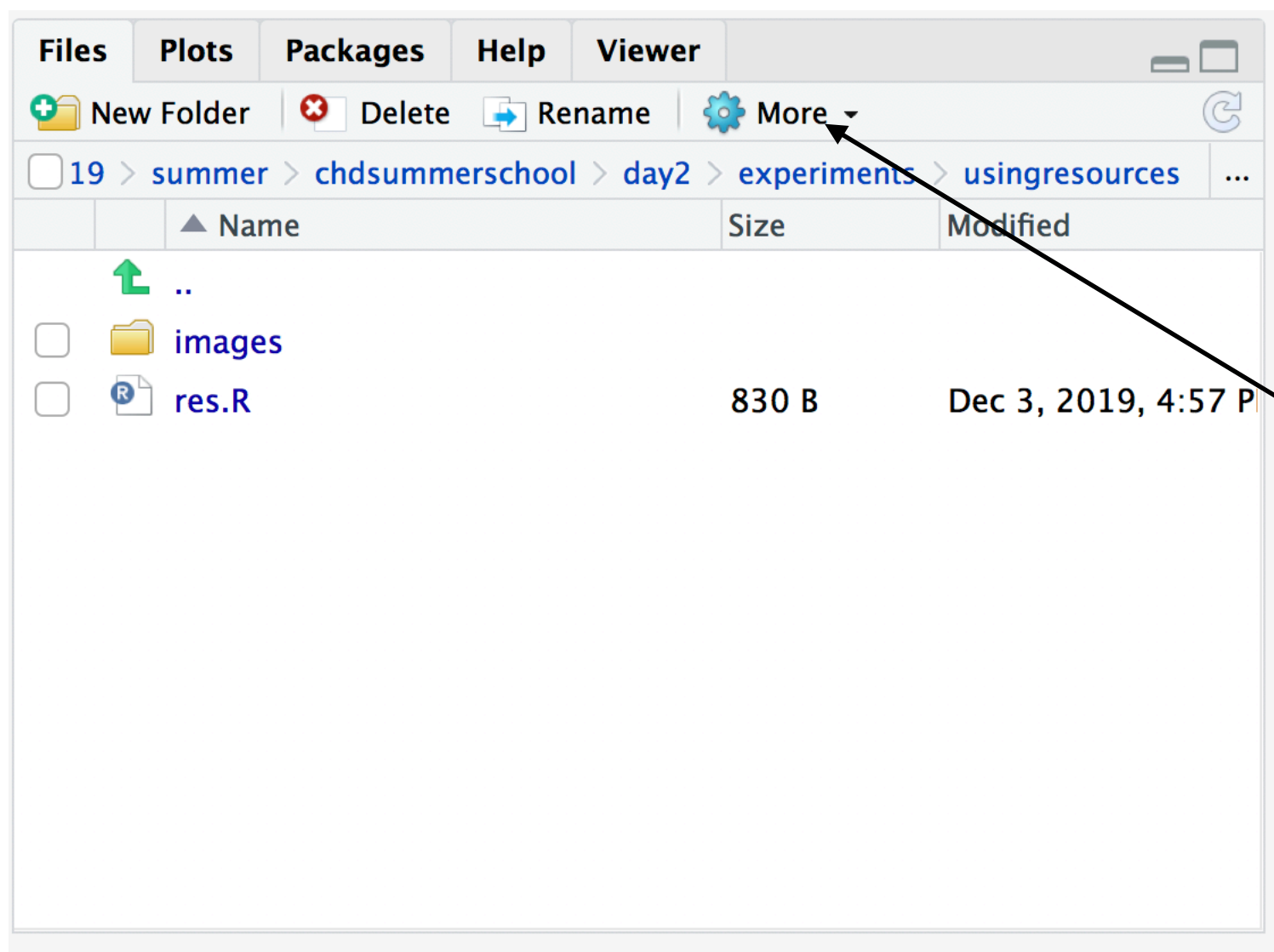
I is working hard
Working hard I am
I'm working hard
Hard working is I

Hint: Use the function `trial_html_slider_response()`

USING RESOURCES

Suppose you want to add images, audio, etc. For that you need to have those files on your computer

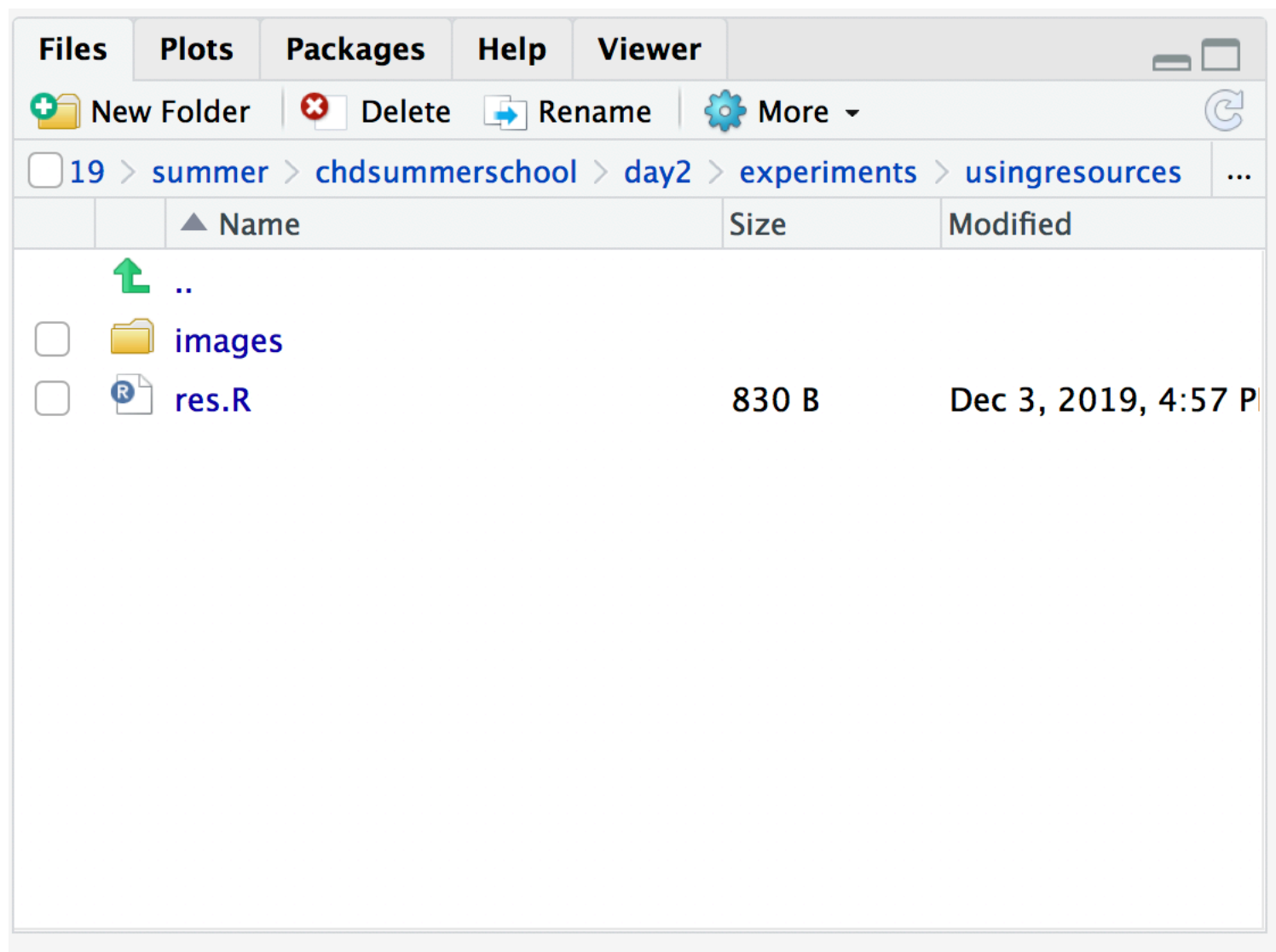
Navigate yourself to a new folder, the `usingresources` one - we'll put our new experiment there.



Remember to
set it as your
working
directory

USING RESOURCES

You'll note it has a folder with images in it. Jaysire assumes this exists. It isn't yet part of your experiment (it hasn't been built yet!) and is in an external place on your computer



USING RESOURCES

Open res.R
and have a
look

```
1 library(jaysire)
2
3 # set directory (deletes any existing old experiment builds in it)
4 my_directory <- file.path("res_exp")
5 # create the empty folder if necessary
6 if(dir.exists(my_directory)) {
7   unlink(my_directory, recursive = TRUE)
8 }
9
10 # ----- resource information -----
11 # need to do this
12
13 # ----- trial template -----
14 trial_template <- trial_image_button_response(
15   stimulus = insert_variable(name = "my_stimulus"),
16   stimulus_height = 200,
17   stimulus_width = 200,
18   choices = c("ugly", "okay", "pretty"),
19   post_trial_gap = 200
20 )
21
22 # ----- build the timeline -----
23 # need to do this
24
25 # ----- build the experiment -----
26 build_experiment(
27   timeline = trials,
28   path = my_directory,
29   on_finish = fn_save_locally()
30 )
```

USING RESOURCES

First you need to tell jaysire where the resources are:

```
# ----- resource information -----  
resource_folder <- file.path("images")  
image_files <- list.files(resource_folder)
```

USING RESOURCES

Then you need to build the timeline where your stimuli are the resources

```
# ----- build the timeline -----  
trials <- build_timeline(trial_template) %>%  
  tl_add_variables(my_stimulus = insert_resource(image_files)) %>%  
  tl_add_parameters(randomize_order = TRUE)
```


USING RESOURCES

Then when you build the experiment you need to tell it to put the resources into the experiment

```
# ----- build the experiment -----  
build_experiment(  
    timeline = trials,  
    path = my_directory,  
    resources = build_resources(resource_folder),  
    on_finish = fn_save_locally()  
)
```

USING RESOURCES

Source and run!

EXERCISE

Find two more images of your own from anywhere and add them to the experiment

Advanced: for those two images only, have people rate them on a slider

FINAL EXERCISE

Put this up onto Google App Engine

IF TIME

Let's take a look at the code for the `samplingframes` experiment we saw earlier