

Charity seeding experiment code and process

How do I make the code run?

DR: First you need to register and create an app id on JustGiving, and save this as in a file you call a file “my_app_id.R”, containing a single line of text

```
my_app_id <- "[ID]"
```

replacing “[ID]” with your ID, without the brackets.

Next...

Install the packages at the top of main.R. Open

```
fundraising_data_pull.Rproj
```

using R and run main.R . It will take 30 - 60 minutes to download all the data; this appears to be determined by Just Giving API limits.

What are the files created?

2 files are created (?and 4 are updated) each time data is drawn from the API, and these are stored to the folders mentioned below.

Note that we must run this regularly to retain data from expired pages (which can't be accessed through the api).

DR: The above needs clarification. The ‘all_’ files are no longer being created I think

The charities that this script uses (in effective_charities.csv) are all recommended by one or more organisations associated with effective altruism (although in some cases the lists only recommend targeting a particular part of the charity's work) [and see comment below].

(Note: we also give a broader list in the file effective_charities_plus, including some additional international mega-charities like MSF.)

[Note, 4 Aug 2018: ATM both lists seem to include the international megacharities]

###Created: A table of currently live (not yet expired) JG fundraising **pages** is created in

```
{data\just_giving_data_snapshots\fundraisers}
```

...with the current date appended. This contains only those “effective” charities that have a just giving id in the effective_charities.csv file (the effective_charities.csv is in the data folder of the project).

A table of **donations** to currently live pages is created in

```
{data\just_giving_data_snapshots\donations}
```

..with the current date appended.

These files are created as a record of the state of the full sample of pages. This is done:

- in case we find issues with the code or our data collection methodology during the experiment, and
- for transparency - this data can be published as a way of allowing our entire process to be visible.

###Updated: **data_pulls.csv** is updated after every pull with the date and the file paths of the 2 files created (fundraisers and donations). The most recent files referenced in this table are used to update the other files.

experimental_pages.csv contains the page data from all of the fundraising pages that have been selected for the experiment, both the treatment(s) and control pages.

donations_to_experimental_pages.csv (donations_to_experimental_pages.csv) contains all of the donations data for the pages in the experimental_pages.csv file. This will be the main source of data for testing the hypotheses.

treatments.csv contains a list of pages that have been or should be treated. There will be a link to the page and a column in which the additional donation can be filled in.

[@DR: Clarify---who will fill this in? In the last proposal, in Treatment_High we would donate twice the average previous donation, rounded to the nearest £10, and in Treatment_Low half this amount, rounded to the nearest £5.]

How do I make it run?

Expected experimental process

####Every 1-3 days: 1. Run the main.R script + This will download all the required data and split new pages (pages that have received at least one donation but are not yet part of the experiment) into treatment and control. + Treatment

and control are determined by the last digit (seconds) of the time of the first donations. Even = Control. Odd = Treatment. - ... [DR: 4 Aug 2018: Update this for our new plan to predict the donation quantile and then alternate Treatment(s) and Control within each quantile.] + All new pages in the treatment group are added to treatments.csv (with a 0 in the additional_donated_amount column) 2. After the script has finished running open treatments.csv in the data folder 3. Filter treatments.csv to the pages that have not yet recieved an additional donation 4. Open each of the links in turn and donate the stated additional donation to that page 5. Update the additional_donated_amount column with the additional amount donated.

The active part of the experiment will be over once we have depleted the pre-defined pot of money (and therefore also have a treatment group of approximately the pre-defined size).

[DR: We might try to plan this to run until we achieve a sufficient statistical power to detect a minimum-size effect]

####After the donation part of the experiment

There will need to be periodic data collection due to the fact that pages can expire (we would lose access to this data). As some pages expire after years, we will need pragmatically decide an end point to our observations [DR: and preregister this]. Most donations come before or immediatley after the event being fundraised for so this shouldn't be too long.

1. First we'll look at the sample of page expiry dates across both treatment and control groups.
2. Select the dates we will need to run the scripts to ensure we capture any donations pre-expiry.
3. By monitoring the rate of new donations, we can pick a stop point when we are confident that we have collected the vast majority of donation to experimental pages. [DR: perhaps we should simply pre-define this based on previous data, i.e., continue monitoring for T weeks after the event date or the start date, where T is the duration such that 99.5% of funds were raised in previous pages]
4. We will then implement our pre-registered statistical analysis, which will underly our main results (we may also add analyses that we have developed in the interim, or that occur as a result of inital data exploration; however, the focus of our inference will be the pre-registered analysis)

#You may need to add quotes around this, i.e., my_app_id <- "/id_number"