# Project: predict the success of a JustGiving fundraiser based on its initial/early observable features; Dr. David Reinstein for BEE3066

*26 November, 2019*

## Just giving prediction - Data described briefly

Donation and fundraiser data pulled from JustGiving.com using (fundraising_data_pull R code)[https://github.com/TWJolly/fundraising_data_pull] created by Toby Jolly. Visit JustGiving.com to learn more.

Pulled: All (live) pages founded from first to last data pull; (hard-code: 2018-4-14 to 15 May 2019 or most recent update); only pages that are 'live' at the time of the pull are captured.

Highly-rated 'effective and international' charities selected only, and only pages with 1+ contributions.

### Data extracts to use here

`u_fdd`: Donations from UK-based pages in the above category, with information on the fundraisers these donations occurred on

`u_fdd_fd`: UK based fundraising pages in the above category, with aggregated statistics on donations by fundraiser"

- This is probably the key data frame to use

Other relevant datasets/frames/objects:

...

## Project goals (goal 1 is essential, 2-3 are optional)

1. Create the 'best predictive model' of how much a fundraising page (started on JustGiving 'highly effective international charities', i.e., from this extract) will raise

The key outcome variable is `uc_fdd_fd$sum_don` (in the other data set a similar variable is coded as `u_fdd$totalRaisedOnline`)

- ... within its "reasonable life" (e.g., until the fundraiser is ended or until it is 95% likely that 90% of the funds that will be raised have already been raised.)
- Alternately, the amount it will raise in a certain reasonable duration (e.g., 'after six months')

The model should be based solely on variables (features) one can observe within 12 hours after the first donation is made on the page. (We created several such variables already ... Should we create a list of these?)

Try to minimise (out-of-sample or cross-validated) prediction error (with error measured using either a squared-deviation or an absolute-deviation metric).

- Other outcome variables (e.g., number of donations made) or aspects of the outcome (e.g., the upper tail of amounts raised) are also of interest

2. Identify particular economically-interesting or practically-interesting predictors of total amounts raised. The impact of the timing, amounts, and messages left on the earliest contributions (those within 12 hours of the first contribution) are of particular interest.
3. Measure and test whether the model form (or key parameters of the model) are significantly different among the targeted charities.

### A proposed plan

0. Download the data using an interface to the JG API (this has been done for you, but you can try to do a download yourself if you like)
1. Define and organize the set of variables (features) available at intervention (done for you, but you can augment it if you like)

- 'cleaning and imputation' may be important here where variables are missing or problematic ( the 'recipes' package can help with this.)

2. Define and calculate the outcome variables (total amount raised) (again, done for you unless you want to look at an additional)

The key outcome variable is `df$var`

- We may need to filter only those pages that are plausibly 'completed' (ended or most of funds likely to be raised)... note in particular the 'event date'

3. Split and set-aside training and test data
4. Model the outcome... I suggest using a shrinkage model, e.g.:

- A Lasso Regression using all features. The regularization/penalty parameter could be optimized for best fit. (Cross-fold).
- A Ridge Regression using all features. The regularization/penalty parameter could be optimized for best fit. (Cross-fold).
- An 'Elastic net' optimising over both the regularisation parameter and the parameter determining how much the L1 and L2 norms are weighted.

5. After all analysis has been done measure the prediction success of the model on the set-aside data. *Do not* re-fit the model after this, as it risks over-fitting. (Instructor: you may choose to create a set-aside dataset yourself to make this a fair contest (i.e., to 'gamify' it, to use the trendy term)]

## Further data description and links to codebooks

See all html files in 'codebooks' folder, especially:

- codebook_u_fdd: donations, with page info
- codebook_u_fdd_fd: pages, with aggregated donation info

### Criterion for inclusion, continued

- Highly-rated (eaf,give_well_top_2017, give_well_standout_2017 life_you_can_save, ace, givewell_other) charities, plus charities with an international poverty and global health (poor countries) focus.
  - (Mostly the latter)
  - Airtable view HERE
- Only included those with a clearly identified ID on justgiving.

### How 'plausibly completed' was defined

In measuring 'predicted total donations' we need to consider 'completed' (or nearly-completed) fundraising pages. We need to remove pages only recently launched.

For now the 'uc_' data frames keep a fundraiser where - event date more than 25 weeks ago

- or expiry date passed

### More detail on this; Constructing a representative sample (advanced; undergraduates may skip this)

**Plausibly 'completed' fundraisers should be defined as:**

- page is no longer active *or*
- time elapsed from founding/event $\rightarrow$ 90% of donation (amounts) are raised 95% of the time. (Note: this should agree with the planned observation time for our experiment. )

**For expired pages, duration until 95% of contributions were recieved. Estimate 90% upper quantile on this**

Note: the sampling of 'live' pages on the site naturally oversamples 'surviving pages', i.e., those with longer expiry dates, with a stronger such bias the older the page.

- Adjustments are needed to recover a representative sample;
  - crudest method: measure duration covering '80% of pages last longer than' remove all older than this
  - medium-crude: construct 'likelihood of survival this long' probabilities, randomly cull pages with these probabilities
  - more sophisticated: constructs weights ('likelihood of surviving this long'), use in analysis

## Further code snippets and functions that may be helpful

### Some key lists of outcome and prediction variables

... you can find more and construct transformations of these, of course

#### The 'recipe' package

... some sample code to start with if you want to try this package; it will need adjusting to choose the right variables and imputations, and code below may have bugs!

```
rec_fd_don <- recipe(sum_don ~ ., data = uc_fdd_fd) %>%
  step_meanimpute(all_numeric(), -all_outcomes())   %>%
  step_modeimpute(all_nominal()) %>% #mode eimputation because nearest neighbor crashes
  step_dummy(all_nominal(), -all_outcomes(), -id) # need a 'design matrix'

fd_imputed <- prep(rec_fd_don, data = uc_fdd_fd ) %>%
  bake(new_data = uc_fdd_fd)

x <- model.matrix(sum_don ~., fd_imputed %>% dplyr::select(v_page, -id))[,-1]

y <- fd_imputed$sum_don

pp("todo: split training and testing data")
```

**Doing some modeling . . . code needs to be adapted; and it may contain bugs!**

```
don_cv_glmnet <- cv.glmnet(x, y, alpha = 0.5, family = "gaussian") #play around with this; can you find

# Fit the final model on  training data

model <- glmnet(x, y, alpha = 0.5, family = "gaussian",
                lambda = don_cv_glmnet$lambda.min)

# Make predictions on the test data (need to split that out)

x.test <- model.matrix(sum_don ~., fd_imputed %>% dplyr::select(everything(), -id))[,-1]
donation_predicted <- model %>% predict(newx = x.test)

sx_imputed_1_pred <- cbind(donation_predicted, fd_imputed) %>%
  dplyr::select(donation_predicted = s0, id)
```

## Data collection duration (an aside fun bit of code. . . )

```
#Now a table/histogram of dur_cd_95 and dur_ed_95 by pageShortName

print("'Time until 95% of funds raised' (days relative to created date)'")
```

```
## [1] "'Time until 95% of funds raised' (days relative to created date)'"
```

```
(qtls_dur_cd_95 <- quantile(u_fdd$dur_cd_95[u_fdd$donnum==1],  probs = seq(0, 1, 0.05),na.rm = TRUE))
```

```
##      0%      5%     10%     15%     20%     25%     30%     35%     40%
## 3.7e-04 5.8e-01 2.9e+00 6.1e+00 9.4e+00 1.3e+01 1.6e+01 2.0e+01 2.5e+01
##     45%     50%     55%     60%     65%     70%     75%     80%     85%
## 2.9e+01 3.5e+01 4.0e+01 4.8e+01 5.7e+01 7.1e+01 8.8e+01 1.1e+02 1.4e+02
##     90%     95%    100%
## 1.8e+02 2.5e+02 3.5e+03
```

```
stat.desc(u_fdd$cumsum_check[u_fdd$donnum==1]) %>% kable(format="html", caption="Sum of contributions 1
```

To get 95% of the contributions for a page we need to wait 34.55 days at median, 88.1 days for 75% of pages, 177.51 days for 90% of the pages and 250.45 days for 95% of the pages.

## Links to relevant Economics papers

Smith, Sarah, Frank Windmeijer, and Edmund Wright. "Peer effects in charitable giving: Evidence from the (running) field." The Economic Journal 125, no. 585 (2015): 1053-1071.

Payne, Abigail, Kimberley Scharf, and Sarah Smith. Online fundraising-the perfect ask?. No. 194. Competitive Advantage in the Global Economy (CAGE), 2014.