Sample code for:

Last name analysis of mobility, gender imbalance, and nepotism across academic systems

by Jacopo Grilli and Stefano Allesina, PNAS 2017

Data

The data are stored in the `data` directory, which contains a file for each dataset:

```
dir("../data")
```

```
## [1] "cnrs_maiden_2016.csv"  "cnrs_married_2016.csv" "ita_2000.csv"
## [4] "ita_2005.csv"          "ita_2010.csv"          "ita_2015.csv"
## [7] "us_2016.csv"
```

To read a data file, use either `read.csv` or the much faster `read_csv` from `tidyverse`:

```
library(tidyverse)
data <- read_csv("../data/ita_2000.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
glimpse(data)
```

```
## Observations: 52,004
## Variables: 10
## $ X1             (int) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...
## $ first_id       (int) 5001, 2749, 6516, 8635, 3064, 6703, 7117, 3938,...
## $ last_id        (int) 3, 23, 23, 94, 588, 685, 716, 1232, 2059, 2103,...
## $ gender         (chr) "F", "M", "M", "M", "M", "F", "M", "M", "F", "F...
## $ rank           (chr) "associate professor", "assistant professor", "...
## $ institution_id (int) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ city_id        (int) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ region         (chr) "Puglia", "Puglia", "Puglia", "Puglia", "Puglia...
## $ sector         (chr) "Math", "Math", "Math", "Math", "Math", "Math",...
## $ year           (int) 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000,...
```

For each researcher, `last_id` is a numeric code corresponding the researcher's last name. All data sets contain the columns `last_id`, `institution_id`,`city_id`, `region`, and `sector`. The Italian data also contains `gender` and `rank`.

Randomizations

The code to perform the randomizations illustrated in the article is contained in `run_randomizations.R`. The code requires the packages `data.table` and `tidyverse`.

To load the code:

```
source("run_randomizations.R")
```

To launch a randomization, invoke the function `randomize_compute_pval`. This function accepts three parameters:

- `data_file`: the data to use
- `randomization`: the type of randomization to perform. Possible choices are `"nation"`, `"city"`, and `"field"`.
- `nrand`: the number of randomizations to perform (in the article, 10^6—but for testing use a smaller number, as it might take a long time otherwise)

For example, to randomize the `us_2016.csv` data by shuffling last names within each field, use:

```
us_2016_byfield <- randomize_compute_pval(data_file = "../data/us_2016.csv",
                                          randomization = "field",
                                          nrand = 5000)
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
```

The object `us_2016_byfield` is a list with containing two tables, `field` and `region`:

```
us_2016_byfield$field
```

```
##           sector totpairs     mean        sd pvalue
##  1:          Agr       35  37.4568  6.503179 0.6594
##  2:          Bio      279 257.9708 18.199658 0.1246
##  3:         Chem       17  20.5280  4.663734 0.8048
##  4:         Econ       40  32.1042  5.778905 0.1046
##  5:      Eng-Ind       40  48.5210  7.315570 0.8972
##  6:          Geo        9   5.6550  2.406154 0.1168
##  7: Hist-Ped-Psi       29   9.2832  3.046539 0.0000
##  8:          Hum       44  41.0252  6.448082 0.3344
##  9:         Math      124 103.8120 10.577157 0.0374
## 10:          Med      634 508.2646 28.328314 0.0002
## 11:         Phys       64  53.7694  7.408875 0.1012
## 12:          Soc      164 129.9622 11.706202 0.0034
```

```
us_2016_byfield$region
```

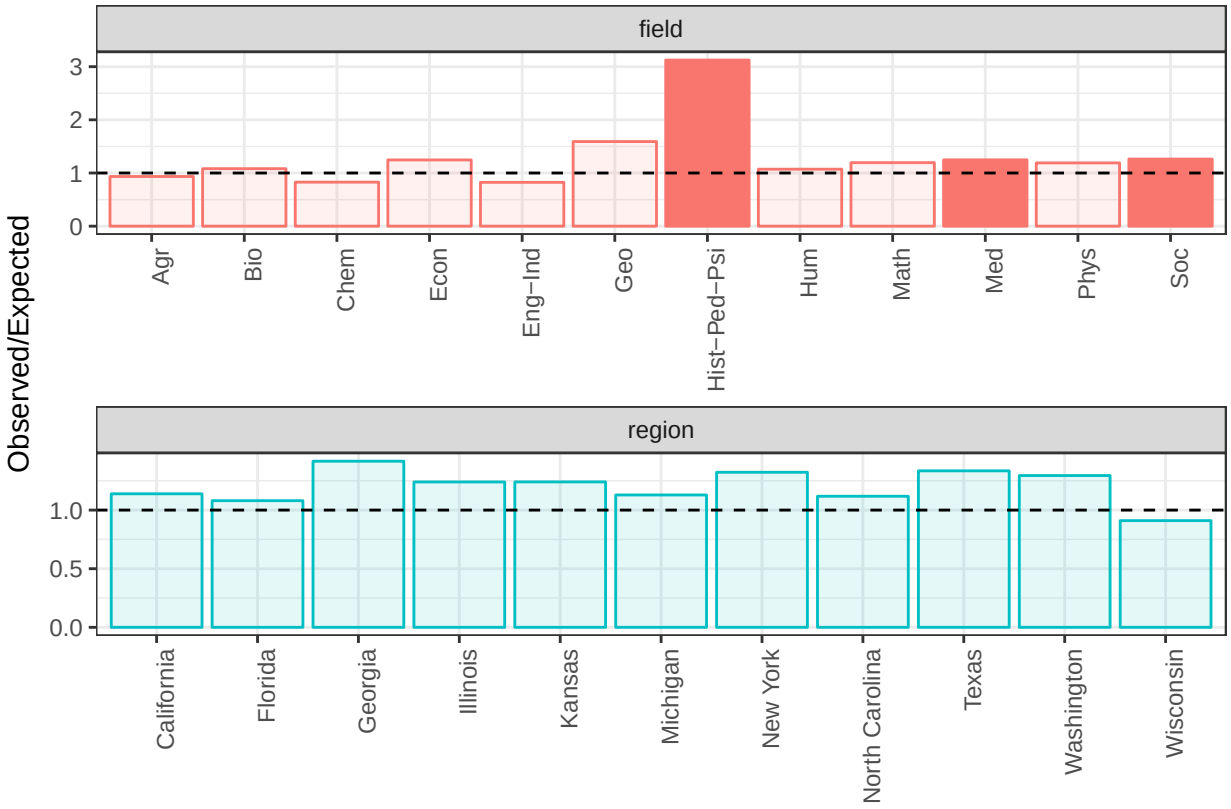
```
##            region totpairs     mean        sd pvalue
##  1:     California      203 178.2480 17.715160 0.0910
##  2:        Florida      199 184.2408 22.427617 0.2508
##  3:        Georgia       86  60.7056 10.245923 0.0140
```

```
##  4:       Illinois      104  83.8912 13.378347 0.0772
##  5:         Kansas       16  12.9016  4.214964 0.2426
##  6:       Michigan      137 121.3796 15.265317 0.1608
##  7:       New York       77  58.2318  9.726627 0.0392
##  8: North Carolina      173 154.7466 20.447875 0.1910
##  9:          Texas      161 120.6296 14.369398 0.0056
## 10:     Washington      250 193.1622 24.720289 0.0182
## 11:      Wisconsin       73  80.2154 13.293615 0.7056
```

For each table, `totpairs` is the total number of isonymous pairs observed in the discipline or region, `mean` is the expected number of pairs in the randomized data, `sd` the standard deviation, and `pvalue` the probability of observing a number of pairs in the randomization that is larger or equal than that in the original data.

To visualize the results, simply call

```
visualize_results(us_2016_byfield)
```



where saturated bars represent significant results once accounted for multiple hypothesis testing.