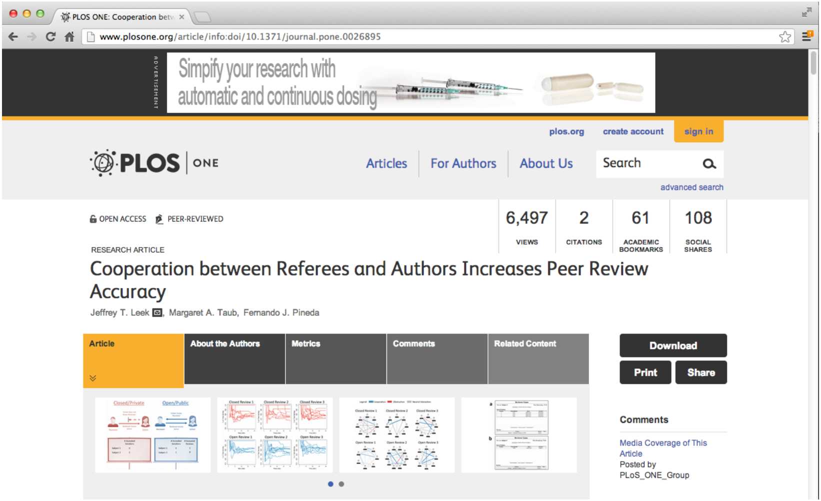
Merging data

Jeffrey Leek

May 18, 2016

Peer review experiment data



PLOS ONE: Cooperation betw ×

www.plosone.org/article/info:doi/10.1371/journal.pone.0026895

plos.org create account sign in

PLOS | ONE

Articles For Authors About Us Search

advanced search

OPEN ACCESS PEER-REVIEWED

6,497	2	61	108
VIEWS	CITATIONS	ACADEMIC BOOKMARKS	SOCIAL SHARES

RESEARCH ARTICLE

Cooperation between Referees and Authors Increases Peer Review Accuracy

Jeffrey T. Leek, Margaret A. Taub, Fernando J. Pineda

Article | About the Authors | Metrics | Comments | Related Content

Download

Print Share

Comments

Media Coverage of This Article
Posted by
PLoS_ONE_Group

http://www.plosone.org/article/info:doi/10.1371/journal.pone.0026895

Peer review data

```
if(!file.exists("./data")){dir.create("./data")}
fileUrl1 = "https://dl.dropboxusercontent.com/u/7710864/dat
fileUrl2 = "https://dl.dropboxusercontent.com/u/7710864/dat
download.file(fileUrl1,destfile="./data/reviews.csv",methoc
download.file(fileUrl2,destfile="./data/solutions.csv",meth
reviews = read.csv("./data/reviews.csv"); solutions <- reac
head(reviews,2)
```

```
##   id solution_id reviewer_id      start       stop time_
## 1  1           3          27 1304095698 1304095758
## 2  2           4          22 1304095188 1304095206
```

```
head(solutions,2)
```

```
##   id problem_id subject_id      start       stop time_le
## 1  1        156         29 1304095119 1304095169      23
## 2  2        269         25 1304095119 1304095183      23
```

Merging data - merge()

- Merges data frames
- Important parameters: *x*,*y*,*by*,*by.x*,*by.y*,*all*

```
names(reviews)
```

```
## [1] "id"          "solution_id" "reviewer_id" "start"
## [6] "time_left"   "accept"
```

```
names(solutions)
```

```
## [1] "id"         "problem_id" "subject_id" "start"
## [6] "time_left"  "answer"
```

Merging data - merge()

```
mergedData = merge(reviews,solutions,by.x="solution_id",by
head(mergedData)
```

```
##   solution_id id reviewer_id    start.x     stop.x time_
## 1           1  4          26 1304095267 1304095423
## 2           2  6          29 1304095471 1304095513
## 3           3  1          27 1304095698 1304095758
## 4           4  2          22 1304095188 1304095206
## 5           5  3          28 1304095276 1304095320
## 6           6 16          22 1304095303 1304095471
##   problem_id subject_id    start.y     stop.y time_left.
## 1        156         29 1304095119 1304095169        234
## 2        269         25 1304095119 1304095183        232
## 3         34         22 1304095127 1304095146        236
## 4         19         23 1304095127 1304095150        236
## 5        605         26 1304095127 1304095167        234
## 6        384         27 1304095131 1304095270        224
```

Default - merge all common column names

```
intersect(names(solutions),names(reviews))
```

```
## [1] "id"        "start"     "stop"      "time_left"
```

```
mergedData2 = merge(reviews,solutions,all=TRUE)
head(mergedData2)
```

```
##   id      start       stop time_left solution_id reviewe
## 1  1 1304095119 1304095169      2343          NA
## 2  1 1304095698 1304095758      1754           3
## 3  2 1304095119 1304095183      2329          NA
## 4  2 1304095188 1304095206      2306           4
## 5  3 1304095127 1304095146      2366          NA
## 6  3 1304095276 1304095320      2192           5
##   problem_id subject_id answer
## 1        156         29      B
## 2         NA         NA   <NA>
## 3        269         25      C
```

Using join in the plyr package

Faster, but less full featured - defaults to left join, see help file for more

```r
library(plyr)
df1 = data.frame(id=sample(1:10),x=rnorm(10))
df2 = data.frame(id=sample(1:10),y=rnorm(10))
arrange(join(df1,df2),id)
```

```
## Joining by: id
```

```
##    id           x           y
## 1   1 -0.45498563 -0.59625161
## 2   2 -0.12201497 -1.12408267
## 3   3 -0.07178439  0.70093741
## 4   4 -1.18864797 -0.26891077
## 5   5  0.24046655  0.32878848
## 6   6 -0.38000897  0.16617171
## 7   7 -0.09085086 -0.89902213
## 8   8  0.68841395  0.69725431
```

If you have multiple data frames

```
df1 = data.frame(id=sample(1:10),x=rnorm(10))
df2 = data.frame(id=sample(1:10),y=rnorm(10))
df3 = data.frame(id=sample(1:10),z=rnorm(10))
dfList = list(df1,df2,df3)
join_all(dfList)
```

```
## Joining by: id
## Joining by: id
```

```
##    id           x           y          z
## 1   5  1.25831999  0.63503556  1.7584569
## 2   4  0.68991831 -0.79643182  0.4744776
## 3   9  0.73262640 -0.42689441  0.2940598
## 4   2  2.03965909  1.84355695 -0.8964207
## 5   7 -0.08288068  0.09888743 -0.6039897
## 6   6  0.26775971  1.17182242 -0.2378004
## 7  10 -1.12067552 -1.07575191 -0.8923557
## 8   3 -0.07280973 -1.03961438  1.3730000
```

More on merging data

- The quick R data merging page - http://www.statmethods.net/management/merging.html
- plyr information - http://plyr.had.co.nz/
- Types of joins - http://en.wikipedia.org/wiki/Join_(SQL)