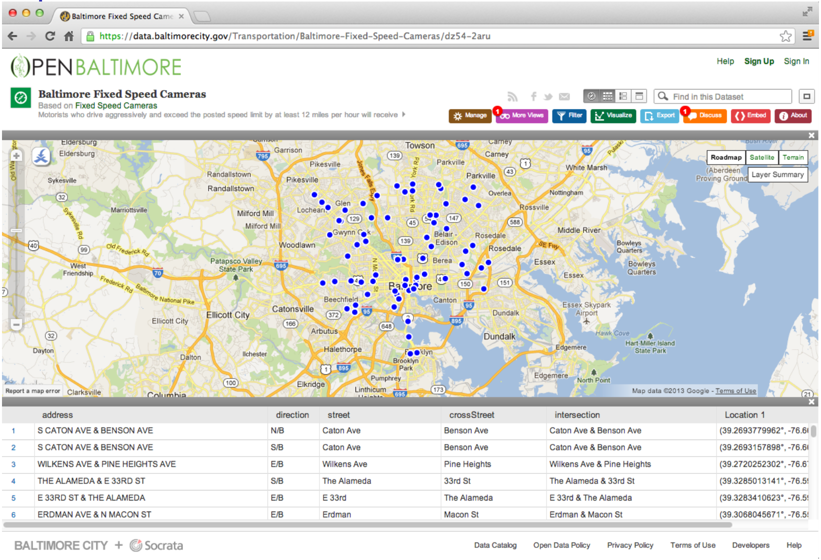
Editing text variables

Jeffrey Leek

May 18, 2016

Example - Baltimore camera data



Baltimore Fixed Speed Cameras

Based on Fixed Speed Cameras

Motorists who drive aggressively and exceed the posted speed limit by at least 12 miles per hour will receive

	address	direction	street	crossStreet	intersection	Location 1
1	S CATON AVE & BENSON AVE	N/B	Caton Ave	Benson Ave	Caton Ave & Benson Ave	(39.2693779962°, -76.6(
2	S CATON AVE & BENSON AVE	S/B	Caton Ave	Benson Ave	Caton Ave & Benson Ave	(39.2693157898°, -76.6(
3	WILKENS AVE & PINE HEIGHTS AVE	E/B	Wilkens Ave	Pine Heights	Wilkens Ave & Pine Heights	(39.2720252302°, -76.6
4	THE ALAMEDA & E 33RD ST	S/B	The Alameda	33rd St	The Alameda & 33rd St	(39.3285013141°, -76.5
5	E 33RD ST & THE ALAMEDA	E/B	E 33rd	The Alameda	E 33rd & The Alameda	(39.3283410623°, -76.5
6	ERDMAN AVE & N MACON ST	E/B	Erdman	Macon St	Erdman & Macon St	(39.3068045671°, -76.5

https://data.baltimorecity.gov/Transportation/Baltimore-Fixed-Speed-Cameras/dz54-2aru

Fixing character vectors - tolower(), toupper()

```r
if(!file.exists("./data")){dir.create("./data")}
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2
download.file(fileUrl,destfile="./data/cameras.csv",method=
cameraData <- read.csv("./data/cameras.csv")
names(cameraData)
```

```
## [1] "address"      "direction"    "street"       "crossS
## [5] "intersection" "Location.1"
```

```r
tolower(names(cameraData))
```

```
## [1] "address"      "direction"    "street"       "crosss
## [5] "intersection" "location.1"
```

Fixing character vectors - strsplit()

- Good for automatically splitting variable names
- Important parameters: *x*, *split*

```
splitNames = strsplit(names(cameraData),"\\.")
splitNames[[5]]
```

```
## [1] "intersection"
```

```
splitNames[[6]]
```

```
## [1] "Location" "1"
```

Quick aside - lists

```
mylist <- list(letters = c("A", "b", "c"), numbers = 1:3, r
head(mylist)
```

```
## $letters
## [1] "A" "b" "c"
##
## $numbers
## [1] 1 2 3
##
## [[3]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/

Quick aside - lists

```
mylist[1]
```

```
## $letters
## [1] "A" "b" "c"
```

```
mylist$letters
```

```
## [1] "A" "b" "c"
```

```
mylist[[1]]
```

```
## [1] "A" "b" "c"
```

http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/Lecture%203.pdf

Fixing character vectors - sapply()

- Applies a function to each element in a vector or list
- Important parameters: X, FUN

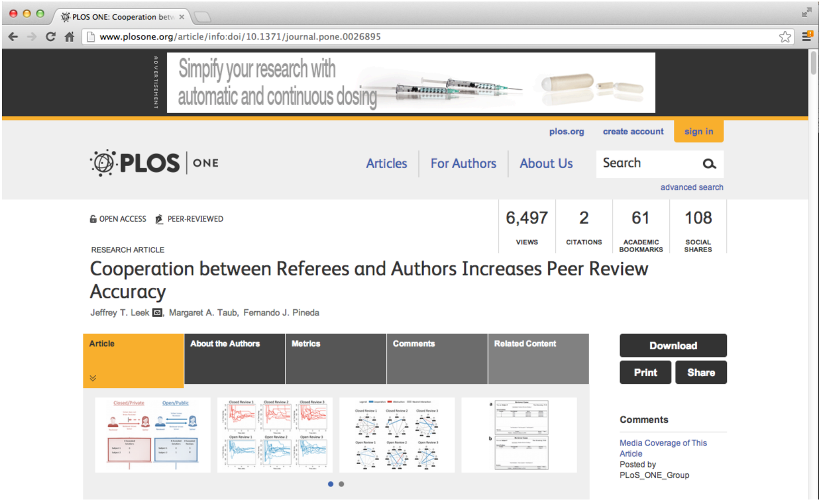
```
splitNames[[6]][1]
```

```
## [1] "Location"
```

```
firstElement <- function(x){x[1]}
sapply(splitNames,firstElement)
```

```
## [1] "address"      "direction"    "street"       "crossS
## [5] "intersection" "Location"
```

Peer review experiment data



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

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

```
fileUrl1 <- "https://dl.dropboxusercontent.com/u/7710864/da
fileUrl2 <- "https://dl.dropboxusercontent.com/u/7710864/da
download.file(fileUrl1,destfile="./data/reviews.csv",metho
download.file(fileUrl2,destfile="./data/solutions.csv",meth
reviews <- read.csv("./data/reviews.csv"); solutions <- rea
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
```

Fixing character vectors - sub()

- Important parameters: *pattern*, *replacement*, *x*

```
names(reviews)
```

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

```
sub("_","",names(reviews),)
```

```
## [1] "id"         "solutionid" "reviewerid" "start"
## [6] "timeleft"   "accept"
```

Fixing character vectors - gsub()

```r
testName <- "this_is_a_test"
sub("_","",testName)
```

```
## [1] "thisis_a_test"
```

```r
gsub("_","",testName)
```

```
## [1] "thisisatest"
```

Finding values - grep(),grepl()

```
grep("Alameda",cameraData$intersection)
```

```
## [1]  4  5 36
```

```
table(grepl("Alameda",cameraData$intersection))
```

```
##
## FALSE  TRUE
##    77     3
```

```
cameraData2 <- cameraData[!grepl("Alameda",cameraData$inter
```

More on grep()

```
grep("Alameda",cameraData$intersection,value=TRUE)
```

```
## [1] "The Alameda  & 33rd St"   "E 33rd  & The Alameda"
## [3] "Harford \n & The Alameda"
```

```
grep("JeffStreet",cameraData$intersection)
```

```
## integer(0)
```

```
length(grep("JeffStreet",cameraData$intersection))
```

```
## [1] 0
```

http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/
Lecture%203.pdf

More useful string functions

```r
library(stringr)
nchar("Jeffrey Leek")
```

```
## [1] 12
```

```r
substr("Jeffrey Leek",1,7)
```

```
## [1] "Jeffrey"
```

```r
paste("Jeffrey","Leek")
```

```
## [1] "Jeffrey Leek"
```

More useful string functions

```
paste0("Jeffrey","Leek")
```

```
## [1] "JeffreyLeek"
```

```
str_trim("Jeff      ")
```

```
## [1] "Jeff"
```

Important points about text in data sets

- Names of variables should be
- All lower case when possible
- Descriptive (Diagnosis versus Dx)
- Not duplicated
- Not have underscores or dots or white spaces
- Variables with character values
- Should usually be made into factor variables (depends on application)
- Should be descriptive (use TRUE/FALSE instead of 0/1 and Male/Female versus 0/1 or M/F)