Getting Data (Part 2)

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Interacting more directly with files

- file - open a connection to a text file
- url - open a connection to a url
- gzfile - open a connection to a .gz file
- bzfile - open a connection to a .bz2 file
- *?connections* for more information
- Remember to close connections

readLines() - local file

- readLines - a function to read lines of text from a connection
- Important parameters: *con*, *n*, *encoding*

```
con <- file("./data/cameras.csv","r")
cameraData <- read.csv(con)
close(con)
head(cameraData)
```

```
                       address direction      street  crossStreet
1       S CATON AVE & BENSON AVE       N/B   Caton Ave   Benson Ave
2       S CATON AVE & BENSON AVE       S/B   Caton Ave   Benson Ave
3 WILKENS AVE & PINE HEIGHTS AVE       E/B Wilkens Ave Pine Heights
4        THE ALAMEDA & E 33RD ST       S/B The Alameda      33rd St
5        E 33RD ST & THE ALAMEDA       E/B      E 33rd  The Alameda
6
1     Caton Ave & Benson Ave (39.2693779962, -76.6688185297)
2     Caton Ave & Benson Ave (39.2693157898, -76.6689698176)
3 Wilkens Ave & Pine Heights  (39.2720252302, -76.676960806)
4     The Alameda  & 33rd St (39.3285013141, -76.5953545714)
```

readLines() - from the web

```
con <- url("http://simplystatistics.org","r")
simplyStats <- readLines(con)
close(con)
head(simplyStats)
```

```
[1] "<!DOCTYPE html>"
[2] "<html lang=\"en-US\">"
[3] "<head>"
[4] "<meta charset=\"UTF-8\" />"
[5] "<title>Simply Statistics</title>"
[6] "<link rel=\"profile\" href=\"http://gmpg.org/xfn/11\" />"
```

Reading JSON files {RJSONIO}

```
library(RJSONIO)
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.json?accessType=DOWNLOAD"
download.file(fileUrl,destfile="./data/camera.json",method="curl")
con = file("./data/camera.json")
jsonCamera = fromJSON(con)
close(con)
head(jsonCamera)
```

```
$meta
$meta$view
$meta$view$id
[1] "dz54-2aru"

$meta$view$name
[1] "Baltimore Fixed Speed Cameras"

$meta$view$attribution
[1] "Department of Transportation"

$meta$view$attributionLink
```

Writing data - write.table()

- The opposite of read.table
- Important parameters: *x*, *file*, *quote*, *sep*, *row.names*, *col.names*

```
cameraData <- read.csv("./data/cameras.csv")
tmpData <- cameraData[,-1]
write.table(tmpData,file="./data/camerasModified.csv",sep=",")
cameraData2 <- read.csv("./data/camerasModified.csv")
head(cameraData2)
```

```
  direction      street  crossStreet              intersection
1       N/B   Caton Ave   Benson Ave     Caton Ave & Benson Ave
2       S/B   Caton Ave   Benson Ave     Caton Ave & Benson Ave
3       E/B Wilkens Ave Pine Heights Wilkens Ave & Pine Heights
4       S/B The Alameda      33rd St     The Alameda  & 33rd St
5       E/B      E 33rd  The Alameda      E 33rd  & The Alameda
6
1 (39.2693779962, -76.6688185297)
2 (39.2693157898, -76.6689698176)
3  (39.2720252302, -76.676960806)
```

Writing data - save(), save.image()

- save is used to save R objects
- Important parameters: *list of objects*, *file*
- save.image saves everything in your working directory

```
cameraData <- read.csv("./data/cameras.csv")
tmpData <- cameraData[,-1]
save(tmpData,cameraData,file="./data/cameras.rda")
```

Reading saved data - load()

- Opposite of save()
- Important parameters: *file*

```
# Remove everything from the workspace
rm(list=ls())
ls()
```

```
character(0)
```

```
# Load data
load("./data/cameras.rda")
ls()
```

```
[1] "cameraData" "tmpData"
```

8/13

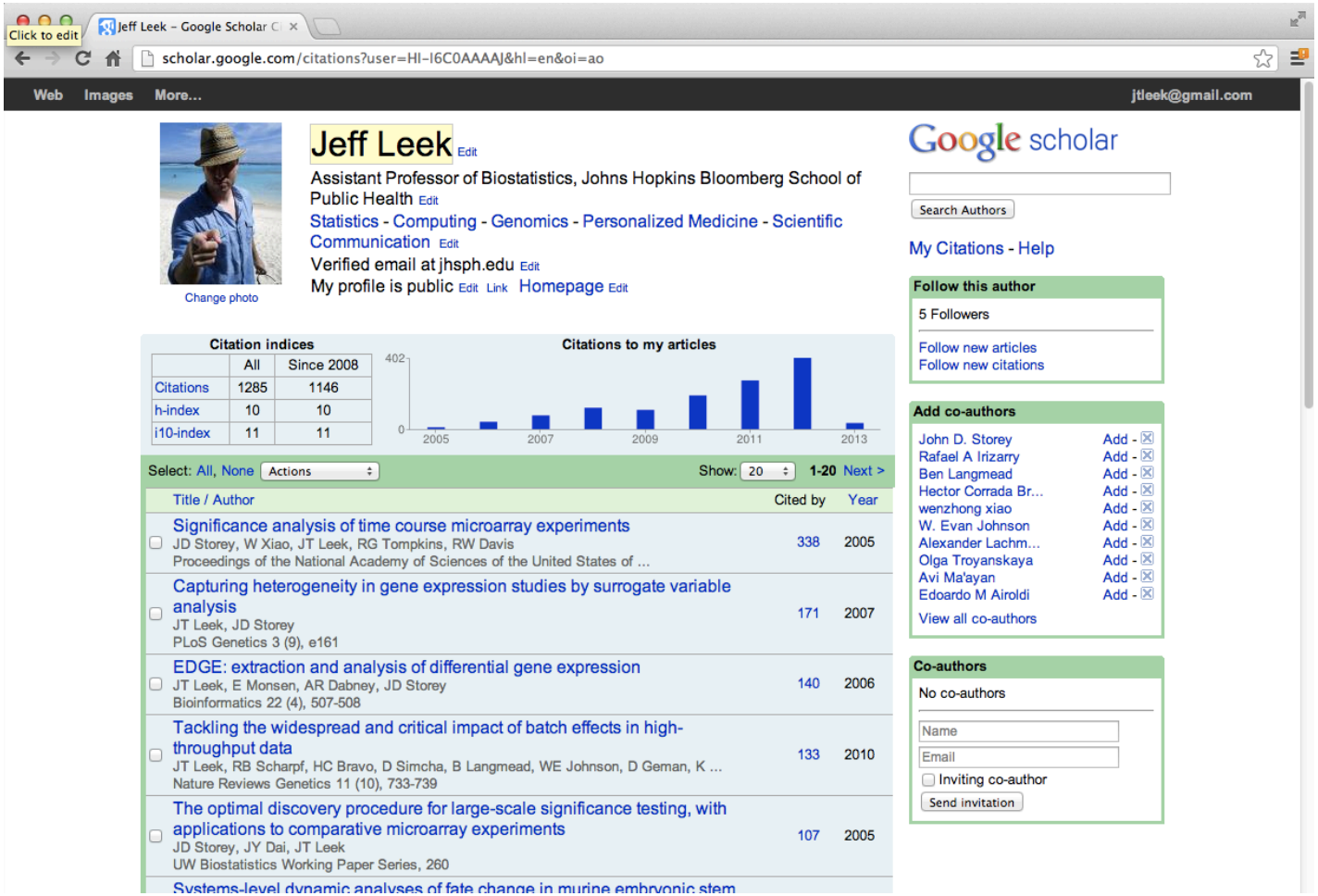
paste() and paste0()

- These functions are for pasting character strings together.
- Important parameters: *list of text strings*, *sep*
- paste0() is the same as paste but with *sep=""*
- Great for looping over files
- See also file.path

```
for(i in 1:5){
  fileName = paste0("./data",i,".csv")
  print(fileName)
}
```

```
[1] "./data1.csv"
[1] "./data2.csv"
[1] "./data3.csv"
[1] "./data4.csv"
[1] "./data5.csv"
```

Getting data off webpages



http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en

10/13

Getting data off webpages

```
library(XML)
con = url("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en")
htmlCode = readLines(con)
close(con)
htmlCode
```

```
[1] "<!DOCTYPE html><html><head><title>Jeff Leek - Google Scholar Citations</title><meta name=\"rob
```

Getting data off webpages

```
html3 <- htmlTreeParse("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en", useInternalNo
xpathSApply(html3, "//title", xmlValue)
```

```
[1] "Jeff Leek - Google Scholar Citations"
```

```
xpathSApply(html3, "//td[@id='col-citedby']", xmlValue)
```

```
 [1] "Cited by" "338"      "171"      "140"      "133"      "107"
 [7] "95"       "78"       "78"       "53"       "16"       "10"
[13] "9"        "9"        "8"        "8"        "6"        "6"
[19] "6"        "5"        "3"
```

Further resources

- Packages:
  - httr - for working with http connections
  - RMySQL - for interfacing with mySQL
  - bigmemory - for handling data larger than RAM
  - RHadoop - for interfacing R and Hadoop (by Revolution Analytics)
  - foreign - for getting data into R from SAS, SPSS, Octave, etc.
- Reading/writing R videos Part 1, Part 2