

# GitHub Basics Workshop

Stats Club

28th February, 2018

# What is Git/GitHub?

- Web-hosting service for code/projects
- Collaborative features
  - Organizations
  - Issue tracking /Bug reports
  - Access control
- Save and publish your code online
  - Documentation
  - Markdown support
  - GitHub Pages
- **Version Control**



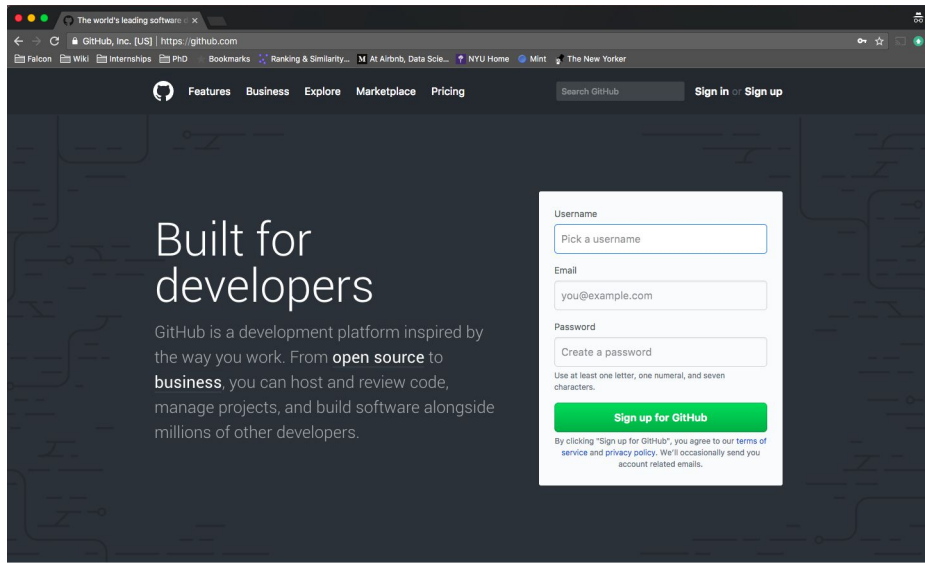
# Why do we need it?

- Helps avoid saving files like *Code\_v1.R*, *Code\_v2.R*, *Code\_final.R*, *Code\_final\_final.R*, etc.
- Ability to travel back in time!
  - Access code written the past
  - Revert back changes
- Easy way to publish your work online
- To find a job
  - More and more recruiters want to see your code
  - A lot of applications **require** a link to your GitHub



# Getting started

Create a GitHub account (It's Free!)



[www.github.com](https://www.github.com)

Install Git on your computer

## Terminal (Mac users)

Install Homebrew first  
`ruby -e "$(curl -fsSL`

`https://raw.githubusercontent.com/Homebrew/install/master/install)"`

Install Git using Homebrew  
`brew install git`

## Windows Users

<https://gitforwindows.org/>

# Platforms- Terminal/Shell

- Setting up the your Git username and emails
  - Credentials by which you code will be **saved** in git on the machine as well as on GitHub
- Global config
  - This config applies to **ALL** your repositories
- Setting up username, email, text editor
  - `git config --global user.name "Kaushik Htet"`
  - `git config --global user.email "inrazkicks@nyu.edu"`
  - `git config --global core.editor "nano -w" OR git config --global core.editor "vim" OR git config --global core.editor "emacs"`



# Platforms- RStudio (v1.1.3 or above)

Quick  
Commands

Same as  
Terminal/Shell

The screenshot displays the RStudio IDE interface. The top-left pane shows an R script with comments and library loading. The bottom-left pane is the console, showing the output of a git pull command. The top-right pane is the Environment pane, listing global objects. The bottom-right pane is the Help pane, showing the documentation for the glm function.

```
1 ---
2 title: "Chat text Analysis"
3 output: github_document
4 ---
5
6 In this project, we are trying to analyse the chat text data from the collaborative problem solving application
7 and trying to find indicators of positive collaborative performance.
8
9 ```{r, warning=FALSE, message=FALSE}
10 require(dplyr)
11 library(Matrix)
12 library(corpus)
13 library(tidytext)
14 library(SnowballC)
15 library(tm)
16 library(ggplot2)
17
18 ## Chat Data
19
20 To perform this analysis, we first define chunks of the chat based on the group and the question being answered.
21 We try to divide up the chat into pieces based on the data available about both members selecting an answer for
22 a particular question. A chunk of chat for a question module starts from the first instance of the selection of a
23 choice for the previous question to the last instance of selection of a choice for the question by members of
24 Dimensions: Affect, Valence, content distribution, mathematical notation, Distance between individual performance :
25
26 Console
27 ~/Documents/NYU/CPSX/git_repo/cpsx/collabassess_chat_text_analysis
28 rewrite Data/tf_idf.chunks.RSav (95%)
29 172-17-36-161:chat_text_analysis kaushikmohan$ git pull origin master
30 From https://github.com/collabassess/chat_text_analysis
31 * branch      master       -> FETCH_HEAD
32 Already up-to-date.
33 172-17-36-161:chat_text_analysis kaushikmohan$ git push origin master
34 Counting objects: 6, done.
35 Delta compression using up to 4 threads.
36 Compressing objects: 100% (6/6), done.
37 Writing objects: 100% (6/6), 412.67 KiB | 3.56 MiB/s, done.
38 Total 6 (delta 4), reused 0 (delta 0)
39 remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
40
41 To https://github.com/collabassess/chat_text_analysis.git
42   1a671f2..9f7cc83 master -> master
43 172-17-36-161:chat_text_analysis kaushikmohan$
44 172-17-37-110:collabassess_chat_text_analysis kaushikmohan$
```

Environment

Object	Class	Attributes
overlapping_modul...	matrix	2 obs. of 2 variables
plot_idf	matrix	36058 obs. of 20 variables
prob_data	matrix	97 obs. of 3 variables
TermByGroupQuesti...	matrix	66872 obs. of 16 variables
TermByGroupQuesti...	matrix	3296 obs. of 14 variables
TermByQuestions	matrix	694 obs. of 6 variables
tga_diff	matrix	13302 obs. of 15 variables
user_obs	matrix	int [1:2, 1:2] 2936 3210 21814 21815

Files Plots Packages Help Viewer

R: Fitting Generalized Linear Models

glm (stats) R Documentation

## Fitting Generalized Linear Models

### Description

glm is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.

### Usage

```
glm(formula, family = gaussian, data, weights, subset,
na.action, start = NULL, etastart, mustart, offset,
control = list(...), model = TRUE, method = "glm.fit",
x = FALSE, y = TRUE, contrasts = NULL, ...)
```

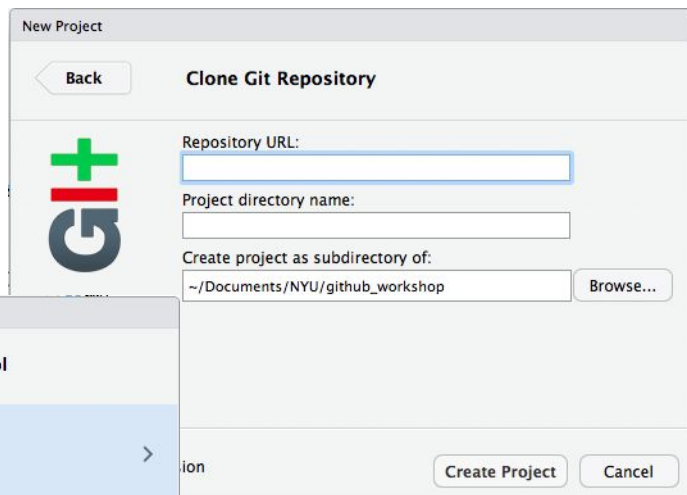
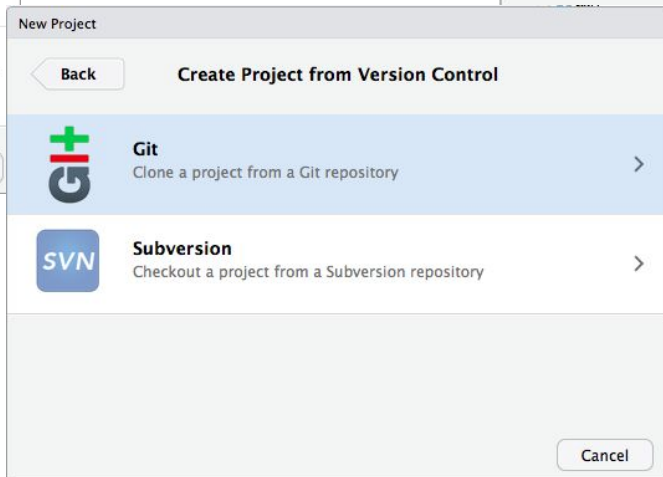
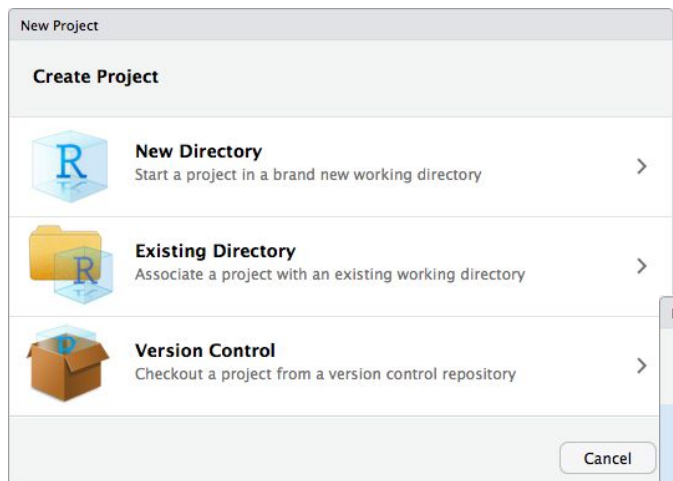
```
glm.fit(x, y, weights = rep(1, nobs),
start = NULL, etastart = NULL, mustart = NULL,
offset = rep(0, nobs), family = gaussian(),
control = list(), intercept = TRUE)
```

```
## S3 method for class 'glm'
weights(object, type = c("prior", "working"), ...)
```

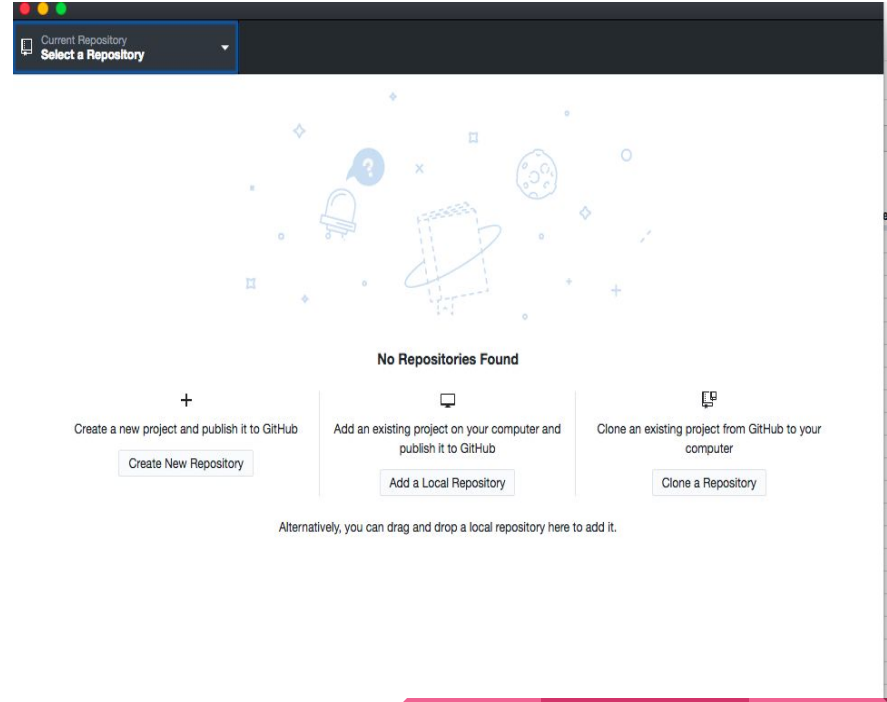
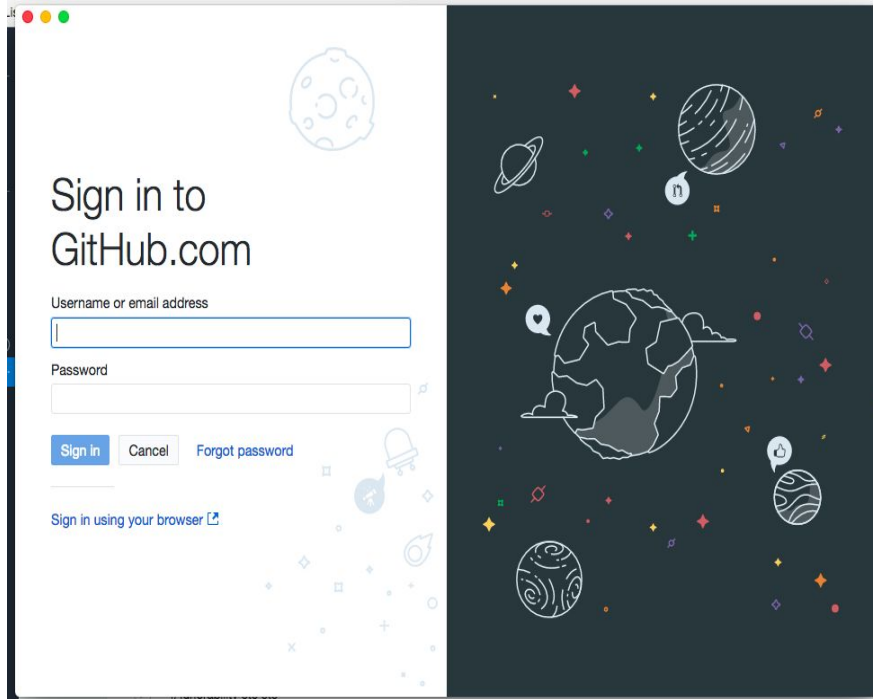
### Arguments

Detailed view

# Platforms- RStudio (v1.1.3 or above)



# Platforms- GitHub Desktop





# Repositories

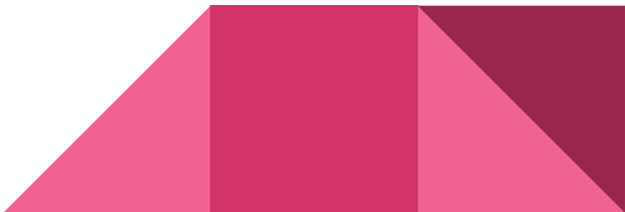
From the terminal:

*First basic terminal commands*

- *cd* - Change directory
- *mkdir* - Make a directory
- *ls* - List contents of the directory
- *pwd* - Print out current directory
- *rm* - Remove a file
- *rm -r* - Remove a folder

You can always do **man** “the command” to read what the command is about.

As to how to quit after the page show up, write  
**q** after the full colon : *for nano*, **wq** after the full colon: *for vim*



# Repositories cont'd

From the terminal:

1. Go to the directory you want to create a folder
  - a. `cd /Users/zarnihitet/Desktop`
2. Create a directory
  - a. `mkdir A3SRocks`
  - b. `cd A3SRocks`
3. Initiate a Git repository
  - a. `git init`
  - b. Add some codes like nano README.md and type in like "Stern sucks!"
  - c. Add in a gitignore file too (More on that later)
    - i. `touch .gitignore`

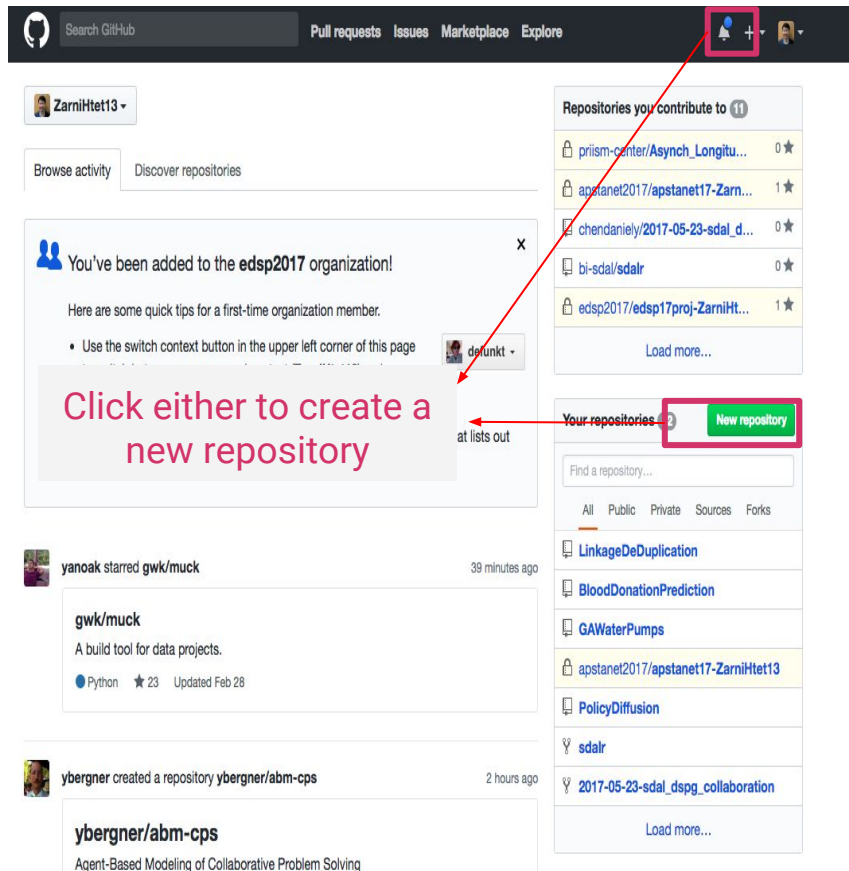
# Repositories cont'd

## 3. Initiate a Git repository (cont'd)

- `git add .` (Note: there's a period after add)
- `git commit -m "Preliminary Code"`

## 4. Go to github.com

- Create a repo
  - Give it a name
  - Write some short descriptions
- From the initial set up page
  - Follow the commands from existing repo setup chunk



# Repositories cont'd

## 5. What if you want to get a repo from Github?

- You need to **Clone**
- Go to repo that you want to clone
- Click on Clone or Download
  - Copy the https link
  - Go back to the terminal and the directory where you want to clone, then

`git clone "https://github.com/dfarnand/wordcloud.git"`

dfarnand / wordcloud

Watch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

A small R function that can pull important words from a selection, and optionally create a word cloud or provide a list of root words.

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request **Click on this button** ~~Create new file~~ ~~Upload files~~ ~~Find file~~ **Clone or download**

dfarnand	Clarification	Latest commit 793baf7 on Jun 2, 2017
.gitignore	Made a few big changes	11 months ago
ImportantWords.R	Made a few big changes	11 months ago
README.md	Clarification	9 months ago

README.md

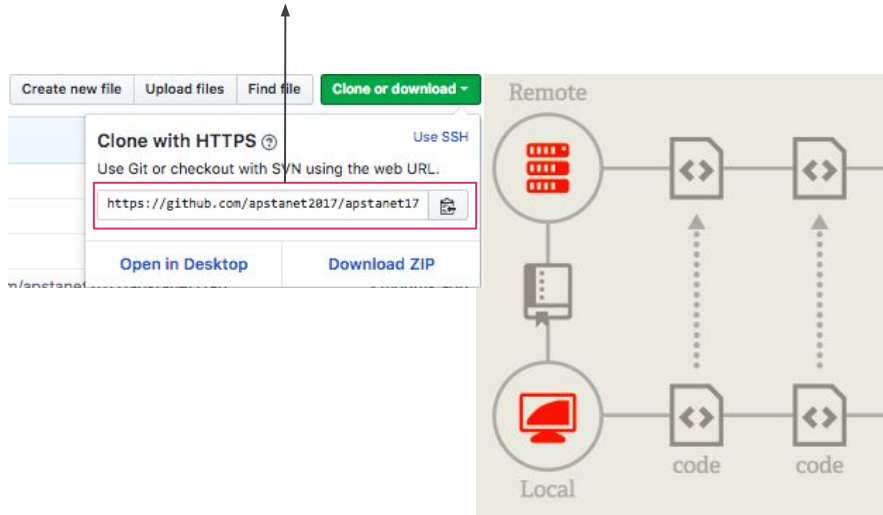
### An R-Based Script to List Important Words or Create a Wordcloud

This is a script in R that I wrote to import plain text, and return a list of "important" (i.e. ordered by frequency) words. It can alternately create a word cloud with argument `wordcloud=T` or reduce the words to their stems with `stem=T`.

# Basic Commands- I

## Remote add origin

```
git remote add origin  
"https://github.com/USERNAME/REPOSITORY.git"
```



## Status: check overall status of changes made

```
git status
```

```
172-17-37-110:networks_project kaushikmohan$ git status  
On branch master  
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.  
(use "git pull" to update your local branch)
```

```
Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified:   Readme.Rmd  
modified:   code/Data/edge_list.csv  
modified:   code/Data/page_details.csv
```

```
Untracked files:  
(use "git add <file>..." to include in what will be committed)
```

```
Readme_files/figure-html/
```

# Basic Commands- II

Diff: Check differences in your files

```
git diff filename
```

```
diff --git a/Readme.Rmd b/Readme.Rmd
index 3bd816a..8c5874b 100644
--- a/Readme.Rmd
+++ b/Readme.Rmd
@@ -1,7 +1,7 @@
---
title: "The Tree of Knowledge"
author: "Kaushik Mohan"
-output: github_document ← What was deleted
+output: html_document ← What was added
---
```

Add: Add files to commit

```
git add filename (to add files individually)
git add . (to add all files with changes)
```

Commit: Stage files before pushing

```
git commit -m "a message about the changes"
```

Make the message meaningful as it would come in handy in the future!

Stash: If you don't want to commit changes and go back to clean working directory

```
git stash
git stash pop (to get back the changes you stashed earlier)
```

# Basic Commands- III

Pull: get the latest updates from the online repo

```
git pull origin master
```

Origin: nickname for the remote repository

Master: branch name (more on this later if time permits)

\*This step is very important when collaborating with others\*

Push: Update the online repo with your changes

```
git push origin master
```

Log: see your recent commits

```
git log
```

```
commit 73fda13f1a4a67e6d1f7bde0c9033c943bed8043 (HEAD -> master)
```

```
Author: Kaushik Mohan <kaushik.s.mohan@gmail.com>
```

```
Date: Mon Dec 18 20:41:13 2017 -0500
```

```
knitted file
```

```
commit d9b6b184cd6bb0eef18292bf481b57c21c4a7be2
```

```
Author: Kaushik Mohan <kaushik.s.mohan@gmail.com>
```

```
Date: Mon Dec 18 20:23:29 2017 -0500
```

```
updated README.Rmd
```

Commit number

# Command to time travel

reset: The main command to go track back any changes made. It's a lifesaver but a bit tricky!

`git reset filename`

(this is the opposite of git add)

`git reset [<mode>] [<commit>]`

(the main command to reset changes)

mode : --hard, --mixed, --soft

commit: commit number

--hard: resets the working directory and removes any changes in the tracked files since <commit>

*Use under extreme circumstances*

--soft: resets the HEAD to <commit> and keeps changes in the tracked files since <commit> which can be seen if do `git status`



# Conflicts and resolutions

Markers of conflicts: <<<<<<, =====, >>>>>>

```
<<<<<< HEAD
This is an example file. We are using this file to show conflicts.
=====
### K and Z make aa good TEAM

### A3SR is better than STERN!|
This is an example file.

>>>>>> 0b1f8570f3c987d0deedc0ac78bfe9843237e22e
```

Remove the markers and retain what is needed. Save, add, commit and push changes normally after that

```
This is an example file. We are using this file to show conflicts.

### K and Z make aa good TEAM

### A3SR is better than STERN!!
```

# Github Issues for Collaboration

🔒 priism-center / **Asynch\_Longitudinal** Private

👁 Unwatch 3 ⭐ Star 0 🍴 Fork 0

<> Code ⓘ Issues 1 🔄 Pull requests 0 📁 Projects 0 📖 Wiki 📊 Insights ⚙ Settings

## Mid-Week Update: 01/02 #2

[Edit](#)[New issue](#)

ZarniHtet13 opened this issue 21 hours ago · 9 comments



ZarniHtet13 commented 21 hours ago

### Tasks Done

1. I modified Daphna's code (under Daphna's src file forZarni.R) to build our 1 data frame. It is from Step 1 to Step 6 of joining the two table sections starting from Line 230. The changes are half to do with code. I could not work to fix the `order` function in actually rearranging the time of each split. Therefore, I reorder the time by each subjectID with a `dplyr::arrange` function instead. The other addition is merging duplicated rows. Please refer to the folder `Visuals` merged\_datasample.png and unmerged\_data\_sample.png. Essentially, the BMI and MEDIA have the same timestamp (Surprise!) and there is no need to have NAs in separate columns for each row. We have to merge them.
2. Interpolation is *successfully* carried out and the data recomposed to the best of my knowledge from Step 7 to Step 10 starting from code line 326. The final output is under `data/final/final_interp_data.csv`.
3. There is a separate file `01_Addendum.Rmd` that deals with Singletons and Matches/Mismatches. More on Singletons in Tasks undone.

### Question with Handling Singleton

My understanding of a Singleton is that there is only 1-time value per subject ID. Then, to what time value do we extrapolate to? Each subject has differing time stamps and differing number of timestamps. What is the appropriate number of timestamps for those singletons to extrapolate to? What timestamps should we use?

### Tasks to follow

1. If the singletons are not removed, the `approx` function throws up. It requires that there must exist two data points for each subject. The current code run does not work. And I think I know why.

In the `01_Addendum.Rmd` file, we will see some initial exploration of singleton matches and general matches/mismatches. If we do *not* remove the Singletons, there is *no* subjectID in Media that has no corresponding subjectID in BMI. It is good because we can then use another time reference from either one data set to expand the singleton to non-singleton. However, it will *not* always be the case. Therefore, this brings back to the question above.

### Assignees



### Labels



### Projects

None yet

### Milestone

No milestone

### Notifications

Unsubscribe


You're receiving notifications because you authored the thread.


3 participants




Lock conversation

# Github Issues for Collaboration


 ZarniHtet13 added **good first issue** **Updates** labels 21 hours ago

 daphnaharel commented 7 hours ago



I'm still working through what you've done, but one request - can you save a version of the dataset that is merged and clean, but with NAs instead of the interpolated values as well.


 ZarniHtet13 commented 7 hours ago

@daphnaharel  
Step 6 does it. It is saved under here:  
/data/processing/merged\_arranged\_data.csv, unless you want something else. I can generate on the fly!


 daphnaharel commented 7 hours ago • edited ▾

Thanks, yes that is what I meant. Can you move it to the final data folder? Also call it something like "merged\_arranged\_data\_with\_na"



  daphnaharel closed this 7 hours ago


 ZarniHtet13 commented 7 hours ago

Done :)

 daphnaharel commented 7 hours ago

I don't see it. I think I need help understanding how to see changes to git hub repositories.

  daphnaharel reopened this 7 hours ago

 ZarniHtet13 commented 7 hours ago

@daphnaharel , my bad. I forgot to push it. If you are looking on Github, you will see it. If you are looking from your terminal you can do `git pull origin master` and that will do.

## avoiding duplicate code #2



ybergner opened this issue on Dec 21, 2017 · 0 comments

Edit

New issue



ybergner commented on Dec 21, 2017

Early on you create some ~30 different networks by duplicating the same 10 lines of code over and over again. Not only is this no fun, but what if you decide to change something? Also, some of the stuff created in each batch is temporary, but since you give each object a unique name, you never clean all of this stuff from memory.

Consider this alternative

```
# Function to Read in Data and Process Network Data
makeNet <- function(siteID) {
  filename <- paste0(siteID, "_Missing_BEHAVIOR.csv")
  tnpnet<-read.csv(filename, header=TRUE, row.names=1)
  tnpnet_matrix<-as.matrix(tpnnet)
  tnpnet_network<-network::network(tpnnet_matrix, directed = TRUE)
  tnpnet_attribute<-subset(full_attribute, Site_ID==siteID)
  tnpnet_network$V["pd_behavior"]<-tnpnet_attribute$PO_Days_1
  tnpnet_network$V["coaching_behavior"]<-tnpnet_attribute$Coaching_Amount_1
  tnpnet_network$V["education"]<-tnpnet_attribute$Degree
  tnpnet_network$V["certified"]<-tnpnet_attribute$Certified
  tnpnet_network$V["experience"]<-tnpnet_attribute$Child_Exp
  tnpnet_network$V["job_title"]<-tnpnet_attribute$Job_Title
  return(tpnnet_network)
}

# can make any individual network as follows
net502_network <- makeNet(502)
net507_network <- makeNet(507)

# even better, make as many as you want from a list
allsiteIDs <- c(502, 507, 508, 509) # or add more
for (siteID in allsiteIDs) {
  netname <- paste0("net",siteID)
  assign(netname, makeNet(siteID))
}
```

Now, if you ever change your network features, you don't have to duplicate the code 30 times.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this repository.

1 participant



Lock conversation

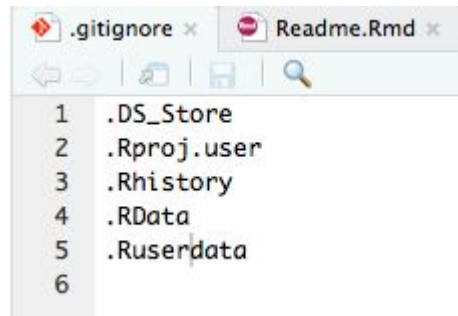
# Git++

## Git LFS

- Large File System (LFS) used for versioning files >50MB
- GitHub provides a 1GB limit for free
- Install Git LFS:
  - <https://git-lfs.github.com/>
- `git lfs install`
- `git lfs track "filename/type"`
- `git add .gitattributes`
- Add, Commit & push normally afterwards

## .gitignore

- Tells Git what files/types of files to ignore (i.e: don't monitor changes in)
- Some examples:
  - Packages: \*.7z, \*.dmg, \*.gz, \*.iso, \*.jar, \*.rar, \*.tar, \*.zip (good to unzip these and push the content)
  - OS generated files: \*.DS\_Store
  - R: \*.Rproj.user, \*.Rhistory, \*.Ruserdata, \*.RData
  - Python: \*.pyo, \*.pyc
  - Backup files: \*



# GitHub Flavoured Markdown

- Make your RMarkdown files look good on GitHub
  - *.html* and *.pdf* files don't display properly on GitHub
  - Easy to publish a *.md* file directly to GitHub pages
- At the top of RMarkdown file, set
  - output: github\_document
- Unfortunately Math mode doesn't work
  - Equations would have to be inserted as images
  - Use <https://www.codecogs.com/latex/eqneditor.php>



# Key points to remember

- add -> commit -> pull -> push
- Remember to commit / stash changes
- Add meaningful messages with commits
- Be sure before using hard reset
- Pull updates before you start!
- Make use projects in RStudio

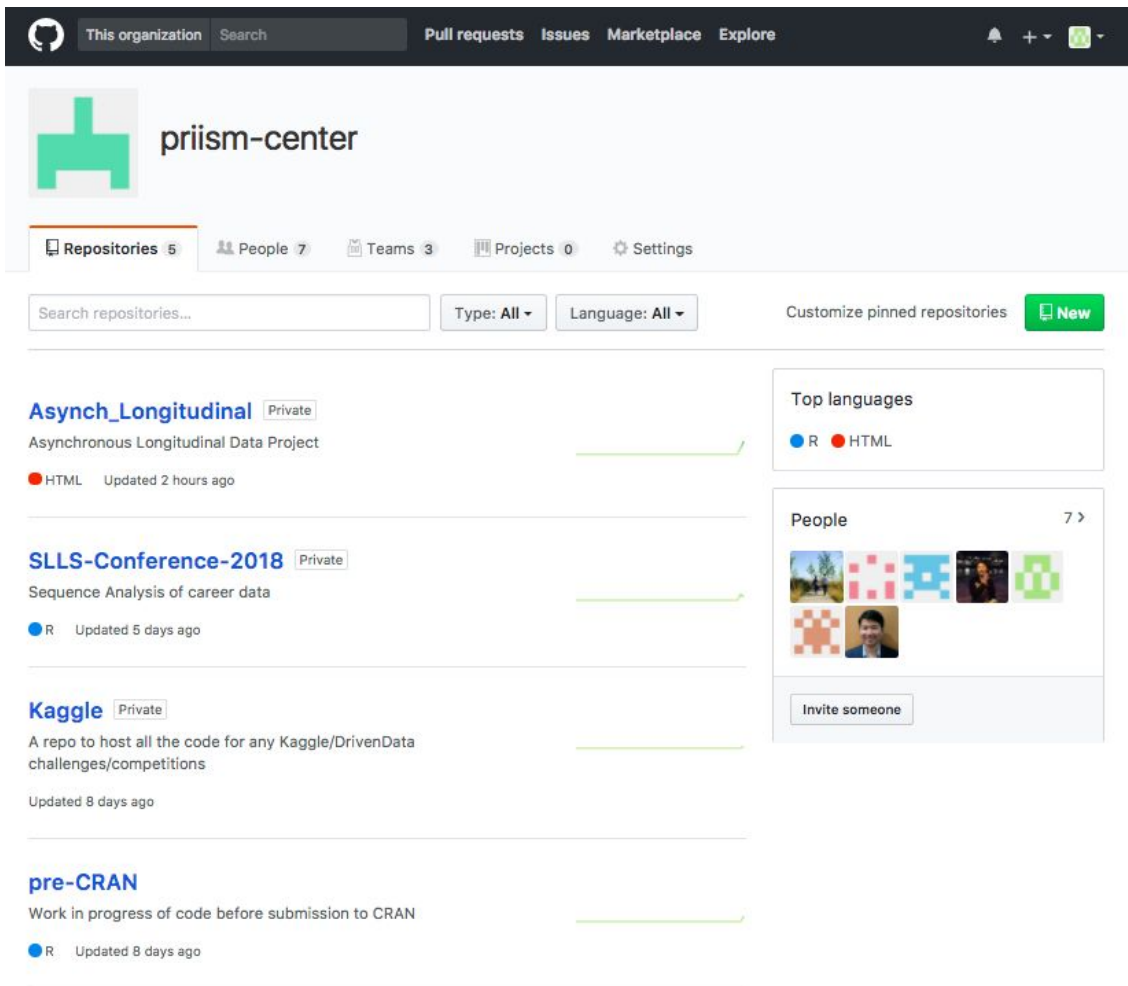
# Best practices

- Create a README file that documents your repository structure
  - What are the different folders? eg: *data*, *src*, *results*, etc.
  - What does each folder have?
- Check/skim through differences before commit
- Try to follow a good Project template structure as much as possible

ZarniHtet13 Data Switch		Latest commit 9f4abee 2 hours ago
Visuals	Data is properly shaped for interpolation function	2 days ago
data	Data Switch	2 hours ago
lit	Restructure Marc's Repo	8 days ago
mnotes	Restructure Marc's Repo	8 days ago
output	Restructure Marc's Repo	8 days ago
src	Clean data pushed	6 hours ago
.gitignore	Restructure Marc's Repo	8 days ago
Asynch_Longitudinal.Rproj	Restructure Marc's Repo	8 days ago
README.md	Initial commit	9 days ago

# PRIISM-center

- Private repos for student projects and student-A3SR faculty collaborations
  - Data and code can be kept private to the team
- Easy to publish student work in the future
- We'll add a repo with tutorials/guides on various topics over time
- Fill out the survey to be added to this group



The screenshot shows the GitHub organization page for 'priism-center'. At the top, there's a navigation bar with links for 'This organization', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the organization's name 'priism-center' is displayed next to its logo, a green stylized 'P'. A tab bar shows 'Repositories' (5), 'People' (7), 'Teams' (3), 'Projects' (0), and 'Settings'. A search bar for repositories is present, along with filters for 'Type: All' and 'Language: All'. A list of repositories is shown, each with a name, a 'Private' label, a description, a language icon, and an update time. The repositories listed are 'Asynch\_Longitudinal' (HTML, updated 2 hours ago), 'SLLS-Conference-2018' (R, updated 5 days ago), 'Kaggle' (updated 8 days ago), and 'pre-CRAN' (R, updated 8 days ago). On the right side, there are two panels: 'Top languages' showing 'R' and 'HTML' with their respective counts, and 'People' showing a grid of member avatars and a link to 'Invite someone'.

Organization: priism-center

Repositories: 5 | People: 7 | Teams: 3 | Projects: 0 | Settings

Search repositories... | Type: All | Language: All | Customize pinned repositories | New

**Asynch\_Longitudinal** Private  
Asynchronous Longitudinal Data Project  
HTML Updated 2 hours ago

**SLLS-Conference-2018** Private  
Sequence Analysis of career data  
R Updated 5 days ago

**Kaggle** Private  
A repo to host all the code for any Kaggle/DrivenData challenges/competitions  
Updated 8 days ago

**pre-CRAN**  
Work in progress of code before submission to CRAN  
R Updated 8 days ago

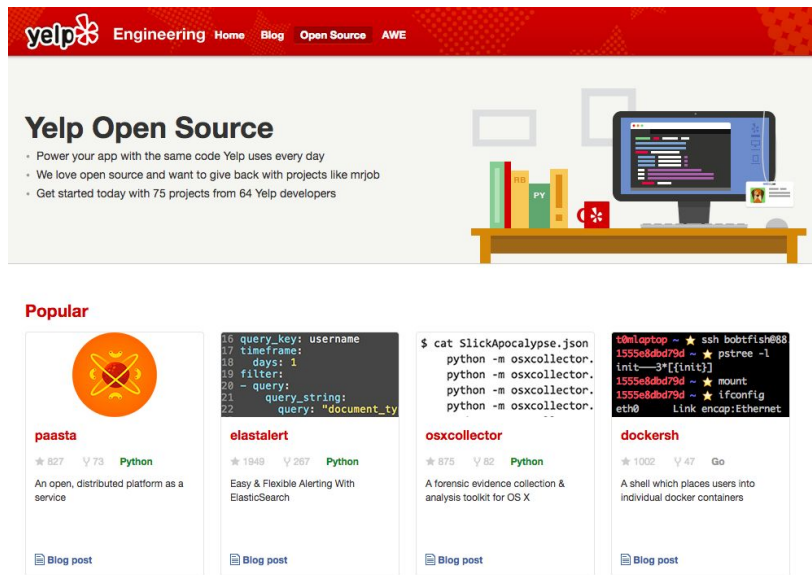
Top languages  
R HTML

People 7 >  
Invite someone



# Next time... (if there's enough interest)

## Create your own project portfolio online

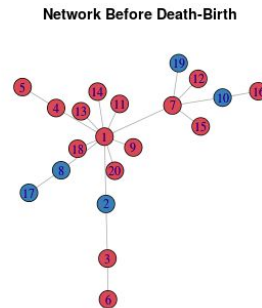
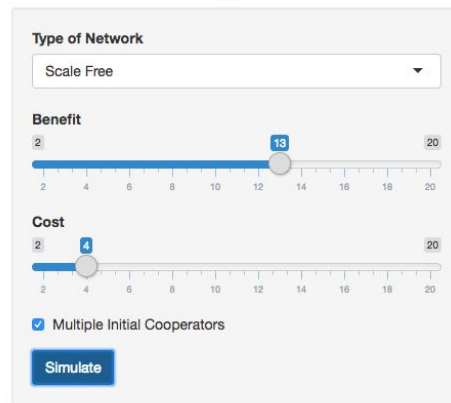


The screenshot shows the 'Yelp Open Source' website. At the top, there's a navigation bar with 'yelp Engineering Home Blog Open Source AWE'. Below the header, the 'Yelp Open Source' section lists three bullet points: 'Power your app with the same code Yelp uses every day', 'We love open source and want to give back with projects like mrjob', and 'Get started today with 75 projects from 64 Yelp developers'. To the right is an illustration of a desk with a computer monitor, books, and a coffee mug. Below this, a 'Popular' section displays four project cards: 'paasta' (a distributed platform as a service), 'elastalert' (flexible alerting with Elasticsearch), 'osxcollector' (forensic evidence collection), and 'dockersh' (a shell for Docker containers). Each card includes a star rating, a 'Blog post' link, and a 'Python' tag.

<https://yelp.github.io/>

## Build interactive Apps with R Shiny

### Evolution of cooperators in a Network



By our very own Bianca!

<https://biabbiassago.shinyapps.io/cooperationsim/>



Thank you!