Project Management with RStudio and Github

Yujun Zhou

September 21, 2018

Overview

- Why do we need project management ?
- Principles for organized project management
- Introductions to R, RStudio, Rmarkdown, Github
- How to manage project with RStudio and Github
- Readable Code Style

Why do we need project management ?

cluster_near_mkt.csv	2/20/2018 12:26	Microsoft Excel C...	38 KB
cluster_weather.R	2/19/2018 13:25	R File	2 KB
code.R	5/1/2017 14:37	R File	3 KB
code_final.R	5/1/2017 14:37	R File	6 KB
concordance_ipc.dta	1/23/2018 11:28	Stata Dataset	98 KB
confusion.R	2/14/2018 11:35	R File	24 KB
daily_rain_2013.dta	12/22/2017 07:52	Stata Dataset	185 KB
dailyrain.do	2/15/2018 17:39	Stata Do-file	3 KB
dailyrain.dta	5/5/2017 07:32	Stata Dataset	943 KB
dailyrain_cluster.dta	2/21/2018 08:56	Stata Dataset	54,256 KB
December.png	10/31/2017 12:03	PNG File	246 KB
density.csv	11/3/2017 09:50	Microsoft Excel C...	11,418 KB
density_plot_code.R	12/12/2017 17:41	R File	1 KB
density_plots_cluster.docx	2/22/2018 19:17	Microsoft Word D...	963 KB
Density0.png	11/4/2017 06:36	PNG File	501 KB
density2.png	11/4/2017 06:34	PNG File	292 KB

Figure 1: Messy project folder

Why do we need project management ?

Raw data, cleaned data, functions,scripts, graphs all in one place

- Original and processed data all in one place: possible contamination of raw data
- Hard to find the exact code to produced a particular table or figure: impossible to reproduce results
- Waste time trying to figure out where to start/proceed after a while

Why do we need project management ?

Name
Press release for approval.doc
Press release final.doc
Press release FINAL VERSION.doc
Press release FINAL FINAL VERSION.doc
IMPROVED FINAL PRESS RELEASE.doc
REVISED APPROVED FINAL PRESS RELEASE.doc
REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!! NEW REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!!! REVISED NEW REVISED APPROVED FINAL PRESS RELEASE v. 2.doc
!!!! Press release as sent.doc

Figure 2: Typical draft versions

Why do we need project management ?

Version control of drafts: Finalfinal_draft_V6.3.docx

- Version names are meaningless, even with dates
- Version notes: run model 0 with county fixed effects, generated figure 1 and table 2; Formatted draft in AJAE style.
- Code and draft are separated: restore results on earlier draft is impossible
- Replicating your work is hard, which makes it hard to collaborate with others

Principles for building an organzied project

0. Build good practices instead of taking the time to clean things up

- You will NEVER find the time

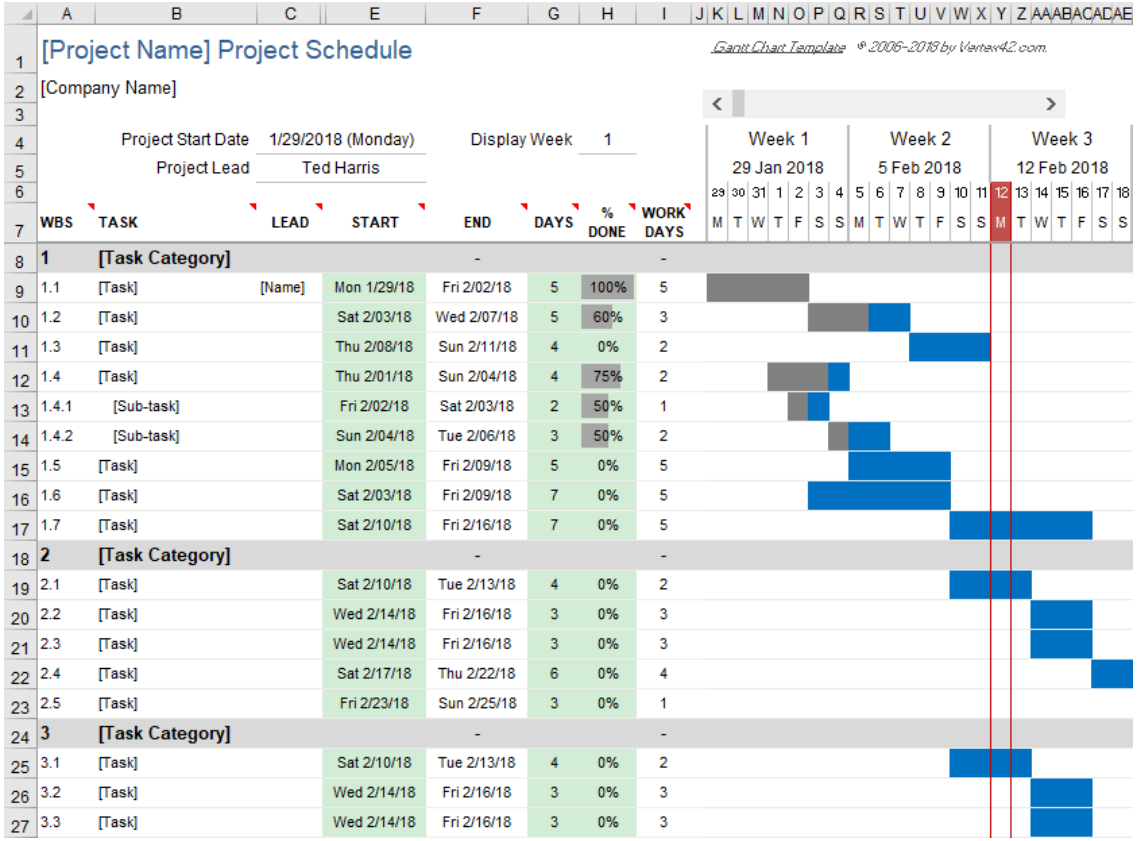
Principles for building an organzied project

1. Progress management

- Start planning as the project lead
- Define objectives with deadlines: IPAD presentation in two month, paper draft in a year, etc
- Break down into smaller tasks and then add daily to-do lists
- Prioritize between projects

Principles for building an organzied project

1. Progress management



[Project Name] Project Schedule

[Company Name]

Gantt Chart Template © 2006-2018 by Vertex42.com.

Project Start Date 1/29/2018 (Monday) Display Week 1

Project Lead Ted Harris

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	[Task Category]			-			-
1.1	[Task]	[Name]	Mon 1/29/18	Fri 2/02/18	5	100%	5
1.2	[Task]		Sat 2/03/18	Wed 2/07/18	5	60%	3
1.3	[Task]		Thu 2/08/18	Sun 2/11/18	4	0%	2
1.4	[Task]		Thu 2/01/18	Sun 2/04/18	4	75%	2
1.4.1	[Sub-task]		Fri 2/02/18	Sat 2/03/18	2	50%	1
1.4.2	[Sub-task]		Sun 2/04/18	Tue 2/06/18	3	50%	2
1.5	[Task]		Mon 2/05/18	Fri 2/09/18	5	0%	5
1.6	[Task]		Sat 2/03/18	Fri 2/09/18	7	0%	5
1.7	[Task]		Sat 2/10/18	Fri 2/16/18	7	0%	5
2	[Task Category]			-			-
2.1	[Task]		Sat 2/10/18	Tue 2/13/18	4	0%	2
2.2	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
2.3	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
2.4	[Task]		Sat 2/17/18	Thu 2/22/18	6	0%	4
2.5	[Task]		Fri 2/23/18	Sun 2/25/18	3	0%	1
3	[Task Category]			-			-
3.1	[Task]		Sat 2/10/18	Tue 2/13/18	4	0%	2
3.2	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3
3.3	[Task]		Wed 2/14/18	Fri 2/16/18	3	0%	3

Week 1 29 Jan 2018 | Week 2 5 Feb 2018 | Week 3 12 Feb 2018

Figure 3: Gantt Chart template in Excel

Principles for building an organzied project

2. Use "Project" to manage your project!

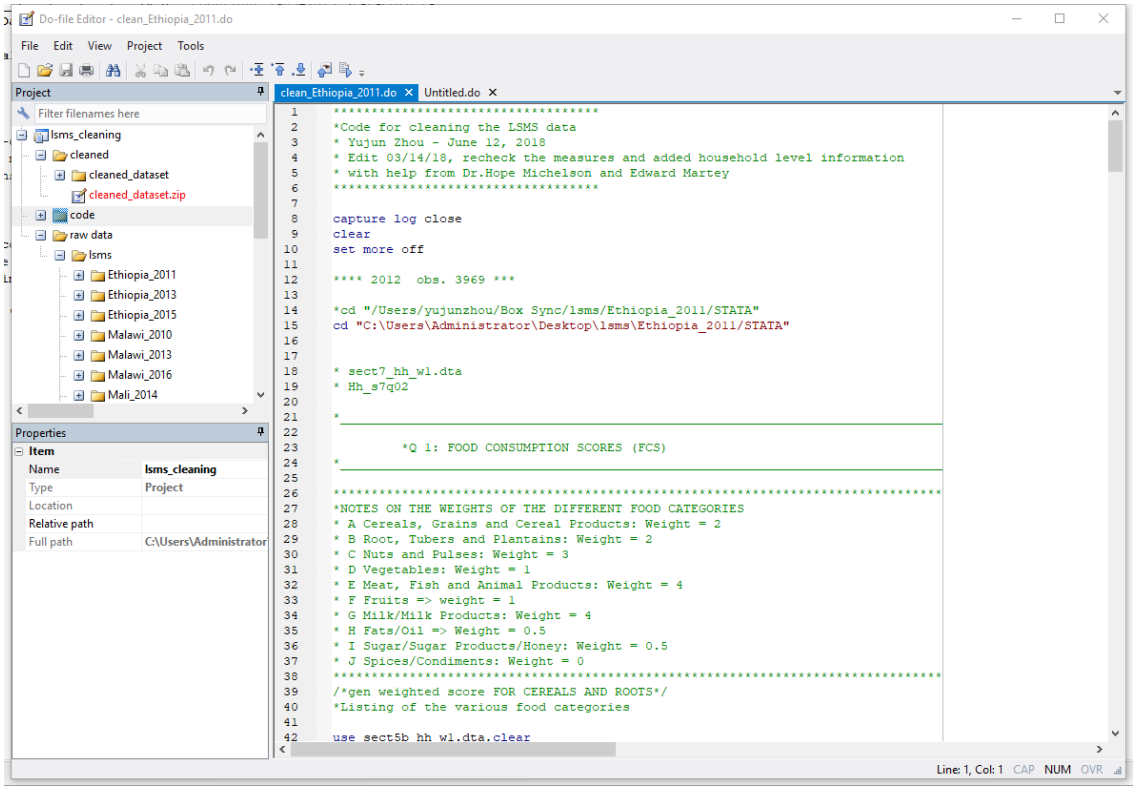
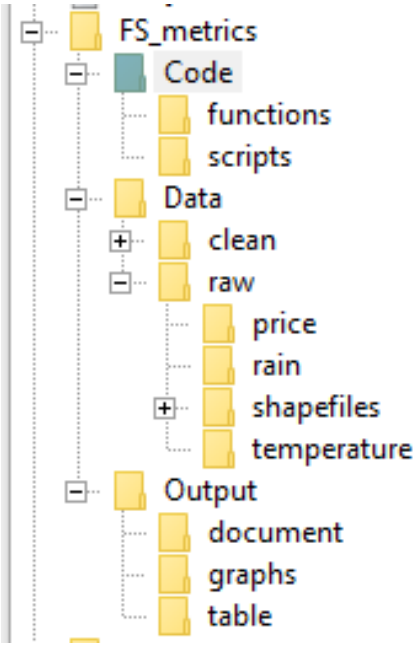


Figure 4: Stata project

Principles for building an organzied project

3. Organized layout with relative path



cluster_near_mkt.csv	2/20/2018 12:26	Microsoft Excel C...	38 KB
cluster_weather.R	2/19/2018 13:25	R File	2 KB
code.R	5/1/2017 14:37	R File	3 KB
code_final.R	5/1/2017 14:37	R File	6 KB
concordance_ipc.dta	1/23/2018 11:28	Stata Dataset	98 KB
confusion.R	2/14/2018 11:35	R File	24 KB
daily_rain_2013.dta	12/22/2017 07:52	Stata Dataset	185 KB
dailyrain.do	2/15/2018 17:39	Stata Do-file	3 KB
dailyrain.dta	5/5/2017 07:32	Stata Dataset	943 KB
dailyrain_cluster.dta	2/21/2018 08:56	Stata Dataset	54,256 KB
December.png	10/31/2017 12:03	PNG File	246 KB
density.csv	11/3/2017 09:50	Microsoft Excel C...	11,418 KB
density_plot_code.R	12/12/2017 17:41	R File	1 KB
density_plots_cluster.docx	2/22/2018 19:17	Microsoft Word D...	963 KB
Density0.png	11/4/2017 06:36	PNG File	501 KB
density2.png	11/4/2017 06:34	PNG File	292 KB

Manage files with relative path

- Separate raw and processed data: raw data should never be touched

```
write.csv(choma_maize_price,"data/clean/choma_maize.csv")
```

- Separate cleaning and analysis scripts
- Separate inputs and outputs (drafts, graphs, tables)

```
ggsave("output/graphs/myplot.png")
```

Use Version Control Tools

- Time capsule for entire projects

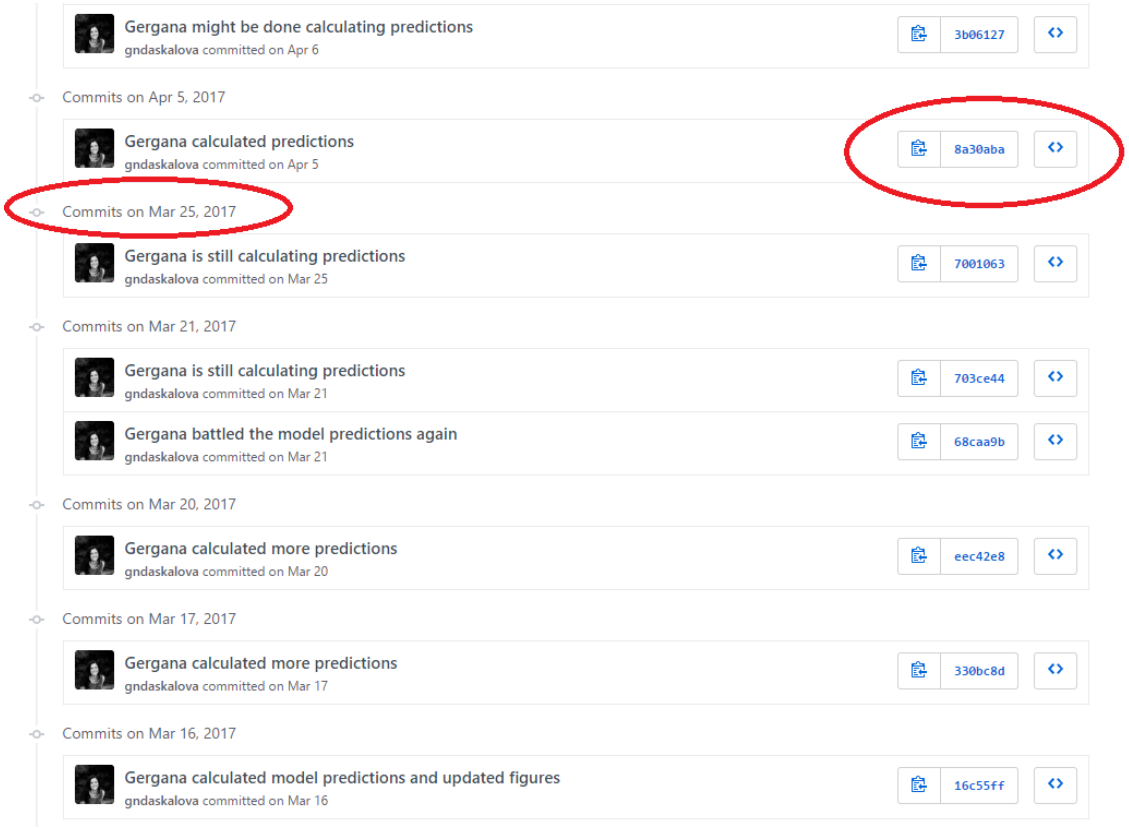


Figure 5: Version history

R, RStudio, Rmarkdown, Github

- R: Open source statistical computing software, flexible data structure, multiple purpose.
- RStudio: an open source development environment for R with nice GUI. Your hub to everything else: Github, latex, website management etc.
- Markdown/ RMarkdown: statistical analysis and results contained in documents
- Easiest way to formulate math formulas (Bye ! Latex compiling errors)
- Formats: PDF, Word, HTML, Beamer
- Github is a cloud-based repository for version control. You can store your code, collaborate on someone else's project, etc.

Benefits of using RStudio and Github integration

- A Github repository = R project
- Easy to work on multiple device
- Click to commit and push

Getting Started

- Sign up for Github and set up your profile
- Create a repository

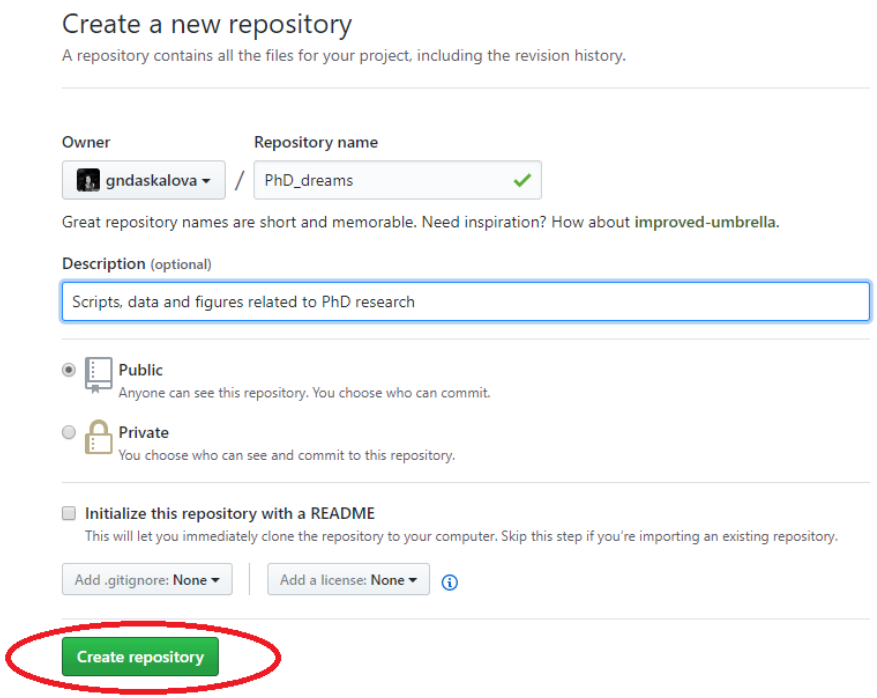


Figure 6: Create repository

Link Git in RStudio

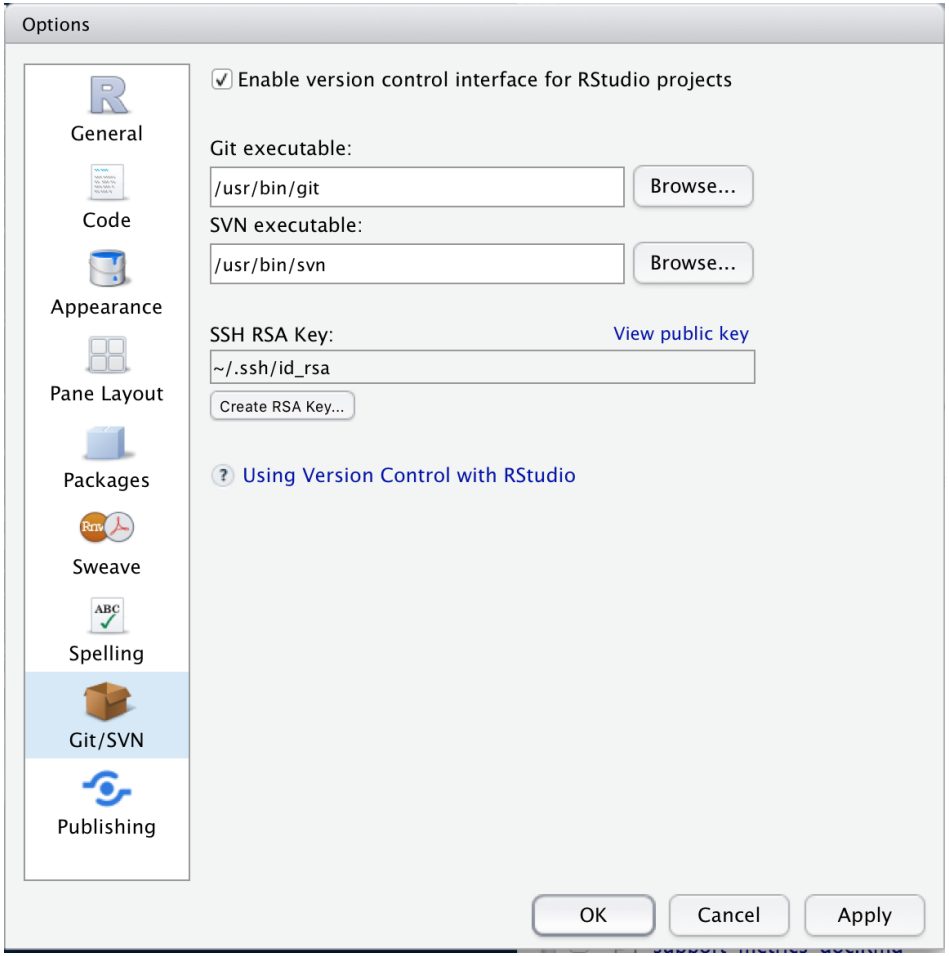


Figure 7: Git setup

How to manage Project with RStudio

- Create a local project from your Github repository

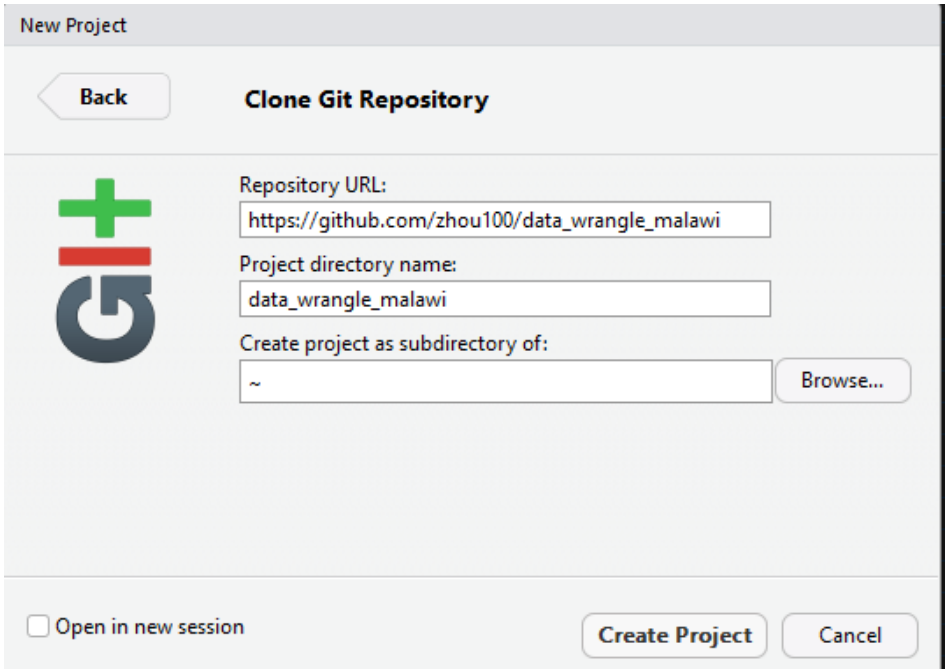


Figure 8: Create project from Git repository

How to manage Project with RStudio

- Commit local changes to online repository, then click push

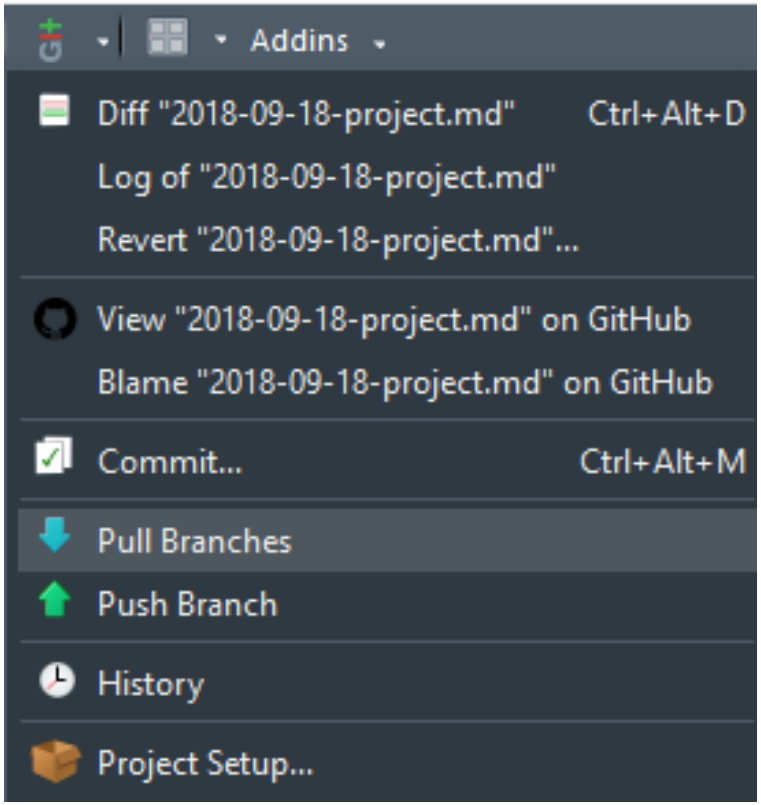


Figure 9: Commit, Pull and Push

Summary

- Make your life easier using projects with version control
- Make collaborations easier with lab-mates, coauthors and adviser
- Prof. Mindy Mallory's "Reproducible Research Practices for Economists"
- Demonstrate End-to-End project management ablity
- Graduate school is project-based learning experience

Github as a technical CV

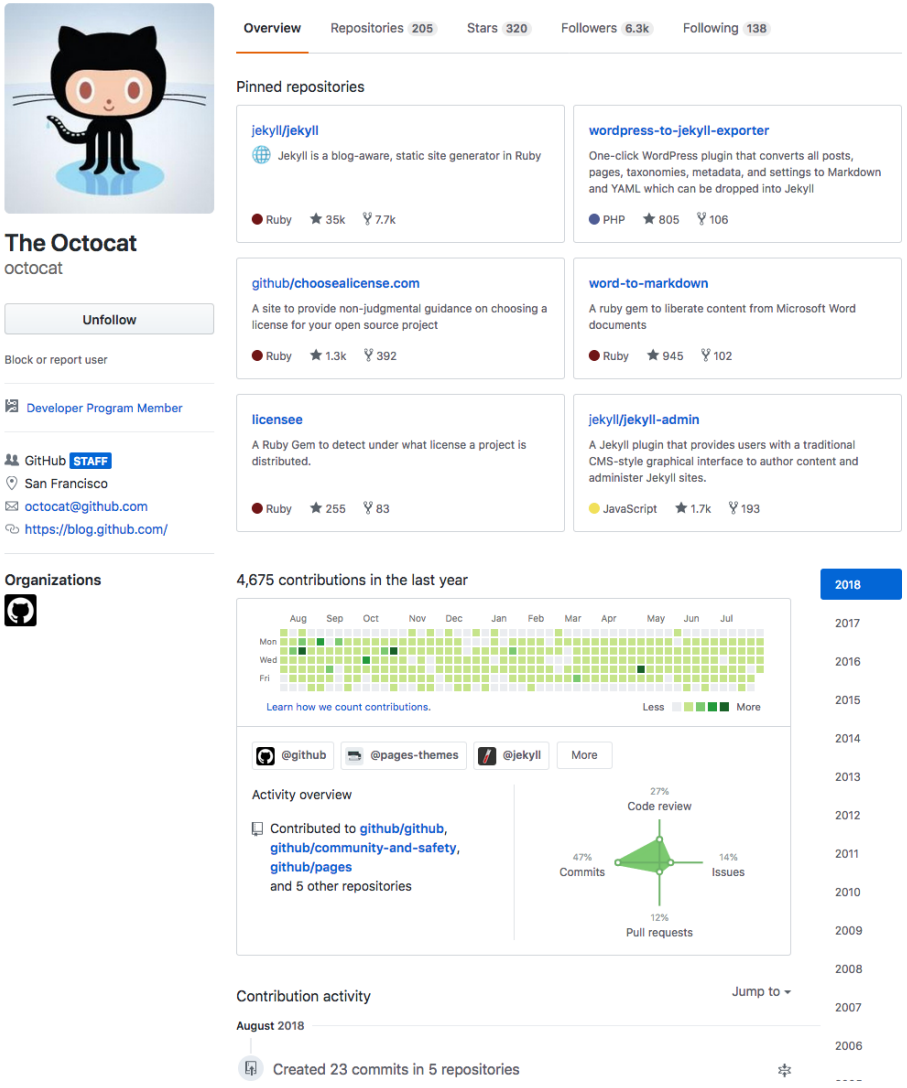


Figure 10: Github Profile

Readable Code Style

- Separate Functions from Scripts
  - Functions are for specific purposes and can be used elsewhere
  - Scripts are used to generate graphs and tables and save them as intermediate products
  - Make the body of the scripts more readable

Script vs Function

```r
##################################################################
# Goal : This script aims to clean up wfp price data, impute missing price
# purpose: use the clean market price and thinness measure to generate the

# Input :
# 1. raw csv files downloaded from wfp price
# 2. population density raster data
# 3. coordinates of clusters.
# 4. shapefile of livelihood zones.

# Output:
# 0. market coordinates generated from market_coordinates.R
# 1. a df of price by product by mkt by yearmon, with missing imputed by m
# 2. a df of market thinness measures by product by mkt by yearmon
# 3. matching the price and mkt_thinness to the cluster and ipczone level
#
# Yujun Zhou -  03/20/18
##################################################################


package = c("dplyr","maptools","rgeos", "rgdal", "raster")
lapply(package, require, character.only = TRUE)

source("R/functions/Yearmon.R")
source("R/functions/market_transpose.R")
source("R/functions/NearMkt.R")
source("R/functions/spatial_price_impute.R")
source("R/functions/PopuWeight.R")
source("R/functions/NameToPrice.R")
source("R/functions/WeightedPrice.R")
source("R/functions/MktReshape.R")

##################################################################
```

Figure 11: Scritps

Script vs Function

```r
#######################################################################
# Goal : generate Date and yearmon variable to help with join
# input: data frame with year and month variables
# output: data frame with Date and yearmon
#################################
library(zoo)

yearmon = function(df,year_var,month_var){

  # Find the year and month stored in the dataframe
  year = df[[year_var]]
  month = df[[month_var]]

  # Transform month to character
  month_character<-month.abb[month]

  # Year-month in character
  yearmon_character = paste(month_character,year,sep="/")

  # Year-month in numbers
  yearmon_format<-as.yearmon(yearmon_character,format = "%b/%Y")

  # Date format
  date_format<-as.Date(yearmon_format)

  # save yearmon and date in a dataframe
  df[["yearmon"]] = yearmon_format
  df[["date"]] = date_format

  return(df)
}
```

Figure 12: Function

Readable Code Style

- Comment at the top about the purpose of the code, input and output
- Use packages and source functions at the start
- Make comments on steps
- Use meaningful variable or function names
- More on code style: Google's R Style Guide

Readable Code Style

- Use dplyr package and piping
  - Example: list the median size of each type (at least 3), in decreasing order

```
data %>% group_by(type) %>%

  summarise(median_size = median(size, na.rm = TRUE)) %>%

  filter(median_size > 3) %>%

  arrange(desc(median_size)) %>%

  select(type, median_size)
```