Git and GitHub Research Collaboration: Day 1

Richard W. Evans [2]

[2]University of Chicago, Open Source Economics Laboratory, M.A. Program in Computational Social Science, and Open Research Group, Inc.

September 20, 2019

New York University and Schmidt Futures

Before we begin: NYU and Schmidt Futures

- Thanks to New York University
- Thanks to Schmidt Futures

Before we begin: Evans

Richard W. Evans (https://sites.google.com/site/rickecon/)

- Associate Director and Senior Lecturer, M.A. Program in Computational Social Science (MACSS), University of Chicago
- Director, Open Source Economics Laboratory (OSE Lab)
- President, Open Research Group, Inc. (OpenRG)
- Steering Committee Member, QuantEcon

Economic Research

- Large scale macroeconomic models of tax policy (demographics, inequality, public debt)
- Synthetic data

Policy Work

- USA, European Commission, India, World Bank
- Policy Simulation Library

Before we begin: You

- Your name
- Where are you from originally?
- Undergraduate/masters institution
- Current research group and senior researchers (Who are you working with?)
- Current research projects
- Most comfortable computational tools

Before we begin: Software and Readings

- Training repository: https://github.com/rickecon/githubtutorial
- Python (Anaconda distribution)
  - Python tutorials
- Git (check if installed)
- GitHub account
  - My Git and GitHub tutorial chapter
  - Pro Git (Chacon and Straub)
  - QuantEcon open access Git and GitHub lecture
  - Atlassian Git tutorial
- Jupyter notebooks

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
- **Jupyter notebooks**
- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
- **Jupyter notebooks**
- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
- **Jupyter notebooks**
- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**

- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing

- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing
  - Negative: Not great for implementation
- **Python IDE (integrated development environment)**

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing
  - Negative: Not great for implementation
- **Python IDE (integrated development environment)**
  - PyCharm (not free)

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing
  - Negative: Not great for implementation
- **Python IDE (integrated development environment)**
  - PyCharm (not free)
  - Spyder (free with Anaconda)

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing
  - Negative: Not great for implementation
- **Python IDE (integrated development environment)**
  - PyCharm (not free)
  - Spyder (free with Anaconda)
  - Positive: Easy to use, similar to MATLAB

Note on Python and Python workflows

- **Python scripts in text editor + execute from terminal**
  - Visual Studio Code, Atom, Sublime Text, Vim
  - Positive: Most flexible, minimum dependency compatibility issues
  - Negative: A little bit cumbersome
- **Jupyter notebooks**
  - Positive: Great for teaching, tutorial, testing
  - Negative: Not great for implementation
- **Python IDE (integrated development environment)**
  - PyCharm (not free)
  - Spyder (free with Anaconda)
  - Positive: Easy to use, similar to MATLAB
  - Negative: Has some dependency/compatibility issues

Schedule

Day	**9:00a-12:00p (morning)**	**1:30-5:00p (afternoon)***
Fri	Introductions, collaborative economics, Git & GitHub basics	Project 1: Evaluating GitHub repositories
Sat	Git merge conflicts, data versioning best practices,	Project 2: General equilibrium model
Sun	Code cleanliness, model validation, closing comments	Project 3: Find a project to contribute to/produce

What I hope you take away

Short-run

- Start using Git and GitHub for all your coding projects
- Feel comfortable with workflow
- Any published and/or public work has associated GitHub repository

Long-run

- You contribute to larger collaborations
- You create a culture of transparency, openness, and inclusivity through
  - documentation
  - test driven development
  - open source help wanted

Computation increasingly important in Econ

- **More data**
- **More sophisticated methods**
- **More sophisticated models**

Computation increasingly important in Econ

- **More data**
  - World Economic Forum: By 2025, 463 exabytes of data created each day globally ($\approx$ 212,765,957 DVDs per day)
- **More sophisticated methods**
- **More sophisticated models**

Computation increasingly important in Econ

- **More data**
  - World Economic Forum: By 2025, 463 exabytes of data created each day globally ($\approx$ 212,765,957 DVDs per day)
- **More sophisticated methods**
  - Chip architecture faster
  - HPC scale more available
  - Better optimizers (math and CS)
  - Interesting uses of machine learning
- **More sophisticated models**

Computation increasingly important in Econ

- **More data**
  - World Economic Forum: By 2025, 463 exabytes of data created each day globally ($\approx$ 212,765,957 DVDs per day)
- **More sophisticated methods**
  - Chip architecture faster
  - HPC scale more available
  - Better optimizers (math and CS)
  - Interesting uses of machine learning
- **More sophisticated models**
  - Higher dimensions of heterogeneity in GE
  - Dynamic programming with neural nets
  - Mixing discrete choice with continuous choice
  - Network analysis and text analysis

Computation increasingly important in Econ

- **More data**
  - World Economic Forum: By 2025, 463 exabytes of data created each day globally ($\approx$ 212,765,957 DVDs per day)
- **More sophisticated methods**
  - Chip architecture faster
  - HPC scale more available
  - Better optimizers (math and CS)
  - Interesting uses of machine learning
- **More sophisticated models**
  - Higher dimensions of heterogeneity in GE
  - Dynamic programming with neural nets
  - Mixing discrete choice with continuous choice
  - Network analysis and text analysis

Git and GitHub are essential

Git-GitHub essential for version control of code

Collaboration is norm, but hard

- **Most papers are multiauthored**
- **Collaboration has benefits**
- **Collaboration has costs**

Collaboration is norm, but hard

- **Most papers are multiauthored**
  - AER current issue
  - Econometrica current issue
- **Collaboration has benefits**
- **Collaboration has costs**

Collaboration is norm, but hard

- **Most papers are multiauthored**
  - AER current issue
  - Econometrica current issue
- **Collaboration has benefits**
  - Comparative advantage
  - Division of labor
- **Collaboration has costs**

Collaboration is norm, but hard

- **Most papers are multiauthored**
  - AER current issue
  - Econometrica current issue
- **Collaboration has benefits**
  - Comparative advantage
  - Division of labor
- **Collaboration has costs**
  - Free-rider problem (not as relevant here, coauthors endogenous)
  - Attribution is difficult (who did what)
  - Very difficult to scale collaborators effectively

Collaboration is norm, but hard

- **Most papers are multiauthored**
  - AER current issue
  - Econometrica current issue
- **Collaboration has benefits**
  - Comparative advantage
  - Division of labor
- **Collaboration has costs**
  - Free-rider problem (not as relevant here, coauthors endogenous)
  - Attribution is difficult (who did what)
  - Very difficult to scale collaborators effectively

Git and GitHub are essential

Git-GitHub only way to scale collaborators (with attribution)

Economics replicability and transparency

- Recent increase in econ journal standards
- Recent increase in open source and replication publishing

Git and GitHub are essential

Git-GitHub most transparent way to make available data and code

Economics replicability and transparency

- Recent increase in econ journal standards
  - AEA Data and Code availability policy
  - Econometrica replication policy and experimental paper policy
- Recent increase in open source and replication publishing

Git and GitHub are essential

Git-GitHub most transparent way to make available data and code

Economics replicability and transparency

- Recent increase in econ journal standards
  - AEA Data and Code availability policy
  - Econometrica replication policy and experimental paper policy
- Recent increase in open source and replication publishing
  - Journal of Open Source Economics (JOSEcon, forthcoming)
  - Journal of Open Source Software (https://joss.theoj.org/)
  - Replication journals and articles

Git and GitHub are essential

Git-GitHub most transparent way to make available data and code

Economics replicability and transparency

- Recent increase in econ journal standards
  - AEA Data and Code availability policy
  - Econometrica replication policy and experimental paper policy
- Recent increase in open source and replication publishing
  - Journal of Open Source Economics (JOSEcon, forthcoming)
  - Journal of Open Source Software (https://joss.theoj.org/)
  - Replication journals and articles
    - Mueller-Langer, Fecher, Wagner (2019) paper
    - Replication Network list of econ journals

Git and GitHub are essential

Git-GitHub most transparent way to make available data and code

Economics replicability and transparency

- Open science vs. open source (need both)

Economics replicability and transparency

- Open science vs. open source (need both)
- Documentation at multiple levels

Economics replicability and transparency

- Open science vs. open source (need both)
- Documentation at multiple levels
- Data and privacy issues
  - Synthetic data is a promising solution

Economics replicability and transparency

- Open science vs. open source (need both)
- Documentation at multiple levels
- Data and privacy issues
  - Synthetic data is a promising solution
- Model sensitivity and robustness is a high dimensional problem that could be crowdsourced.

Economics replicability and transparency

- Open science vs. open source (need both)
- Documentation at multiple levels
- Data and privacy issues
  - Synthetic data is a promising solution
- Model sensitivity and robustness is a high dimensional problem that could be crowdsourced.

Git and GitHub are essential

Git-GitHub most conducive to multiple levels of documentation—open science and open source—plus crowdsourcing of sensitivity and robustness

Git and GitHub

- Git is local version control software
  - primarily checks differences between tracked directories
  - local functionality and commands (`git ...`)
  - some remote functionality (e.g., `git pull`, `git push`)
  - Large learning curve, not intuitive
  - Originated with/for development of Linux

Git and GitHub

- Git is local version control software
  - primarily checks differences between tracked directories
  - local functionality and commands (`git ...`)
  - some remote functionality (e.g., `git pull`, `git push`)
  - Large learning curve, not intuitive
  - Originated with/for development of Linux
- GitHub is a cloud platform for developing, sharing, collaborating on code.
  - Largest platform in world
  - Alternatives include Bitbucket, SourceForge, Backlog, SVN,

Git and GitHub

- Git is local version control software
  - primarily checks differences between tracked directories
  - local functionality and commands (`git ...`)
  - some remote functionality (e.g., `git pull`, `git push`)
  - Large learning curve, not intuitive
  - Originated with/for development of Linux
- GitHub is a cloud platform for developing, sharing, collaborating on code.
  - Largest platform in world
  - Alternatives include Bitbucket, SourceForge, Backlog, SVN,

Git and GitHub Together

Together Git and GitHub are the standard for code development and collaboration—both private and open source

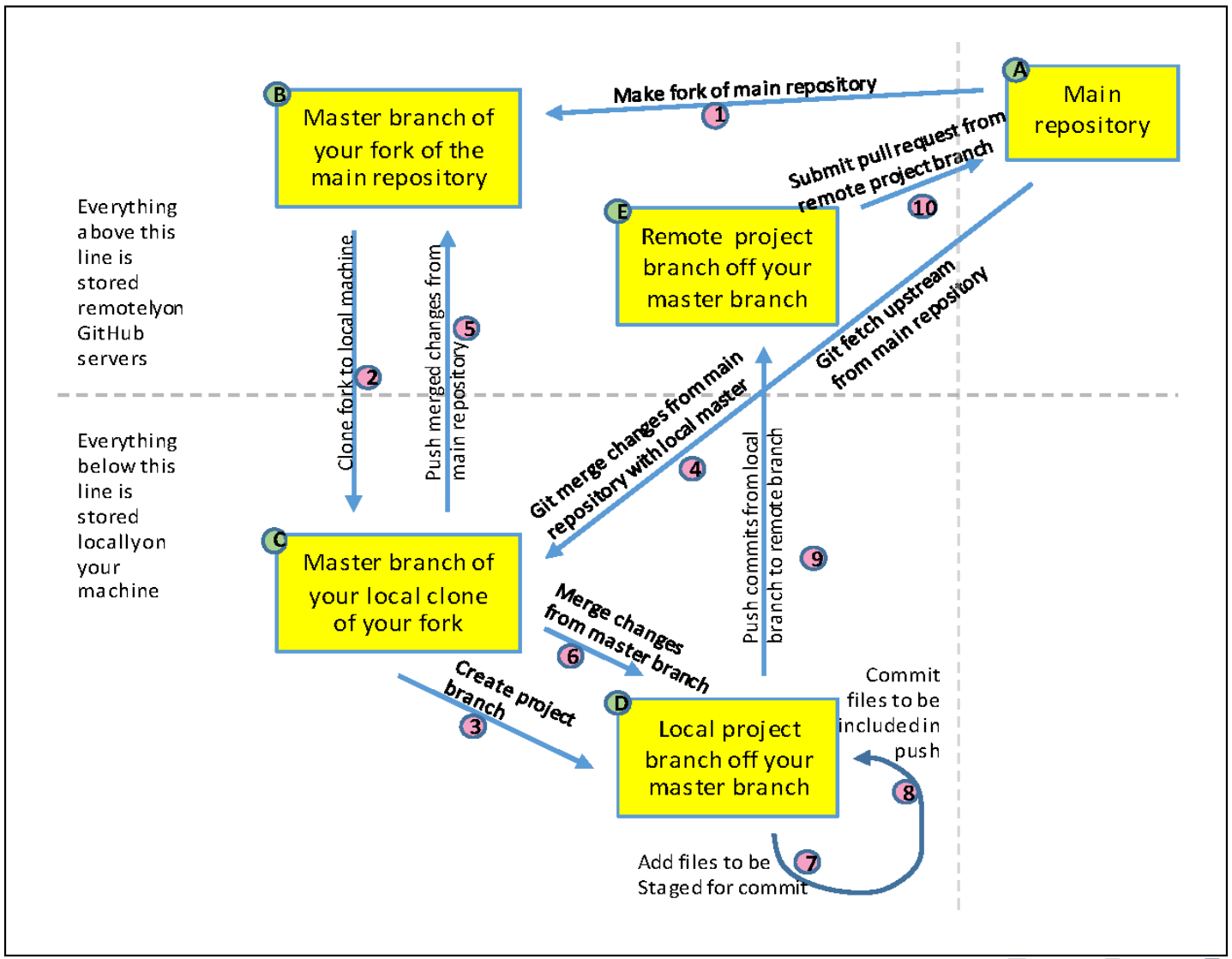
Why not Google Drive or Dropbox?

- Google Drive/Docs
- Dropbox
- Git and GitHub

First step: git config

- `git config -list -show-origin`
- `git config -global user.name "Your Name"`
- `git config -global user.email yourname@example.com`
- `git config -global core.editor vim`
- `git config -global core.editor vim`

Git workflow diagram



Fork, clone, fetch, merge, push

1. Fork the remote repository

Fork, clone, fetch, merge, push

1. Fork the remote repository
2. Clone your remote fork to local machine
   - `git clone <fork_url.git>`

Fork, clone, fetch, merge, push

1. Fork the remote repository
2. Clone your remote fork to local machine
   - `git clone <fork_url.git>`
3. Change directory to new local repo and add remote upstream
   - `git remote add upstream <master_url.git>`

Fork, clone, fetch, merge, push

1. Fork the remote repository
2. Clone your remote fork to local machine
   - `git clone <fork_url.git>`
3. Change directory to new local repo and add remote upstream
   - `git remote add upstream <master_url.git>`
4. Whenever a change happens to remote master
   - `git fetch upstream`
   - `git merge upstream/master`
   - `git push origin master`

Branch, add, commit, push

1. Make a local branch to make changes
2. `git add` stages files for commit
3. `git commit` says these changes are in
4. `git push <remote_name> <branch_name>` sends commits to remote branch

Pull requests

1. Pull request is a remote phenomenon
2. The name is indicative of hierarchy
3. The thread is important

Collaborative workflow

- GitHub workflow (fork, branch pull request, issues, comments)
- Favorite recent threads
  - Issue #435 “Haircut to the government interest rate”
  - Issue #434 “Large open economy option”

Collaborative workflow

- GitHub workflow (fork, branch pull request, issues, comments)
- Favorite recent threads
  - Issue #435 "Haircut to the government interest rate"
  - Issue #434 "Large open economy option"
- You can look at `Issues` to find places to contribute
- You can submit issues

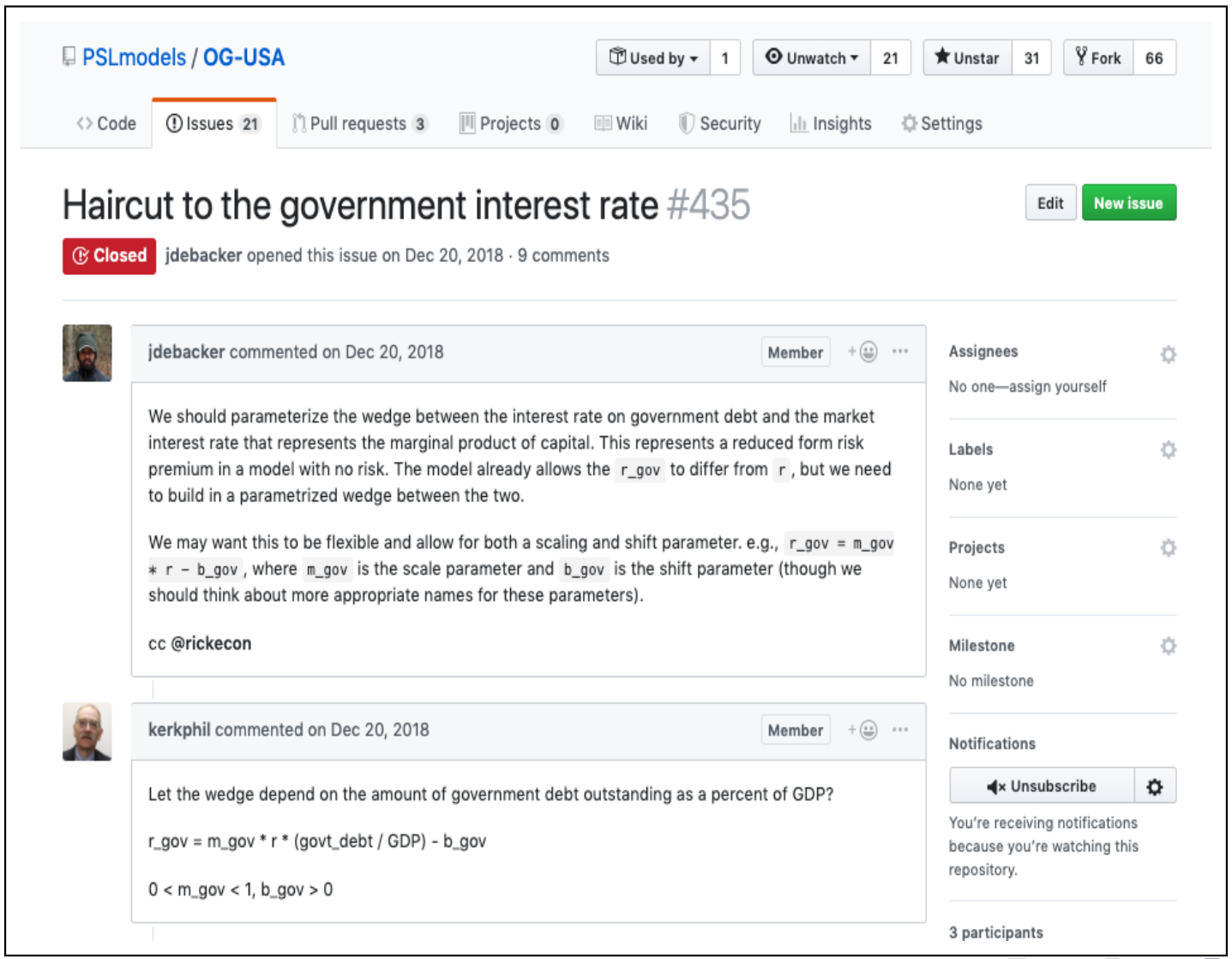
Collaborative workflow

- GitHub workflow (fork, branch pull request, issues, comments)
- Favorite recent threads
  - Issue #435 "Haircut to the government interest rate"
  - Issue #434 "Large open economy option"
- You can look at `Issues` to find places to contribute
- You can submit issues

Essential to large scale collaboration...

Unit testing and continuous integration testing allow project participation to scale.

GitHub issues and PR comments



GitHub issues and PR comments

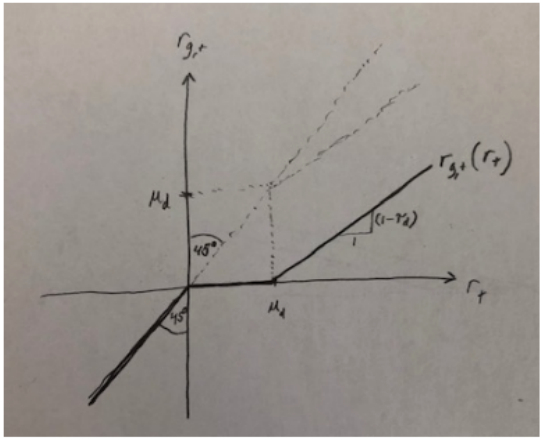
rickecon commented on Jan 8 • edited ▾ Member

@jdebacker, **@kerkphil**. OK, I agree that it is the best to set up a perfectly competitive mutual fund industry that takes savings from households and lends it to firms and government. And we can easily set the production function as CES, and just calibrate it to be Cobb-Douglas initially.

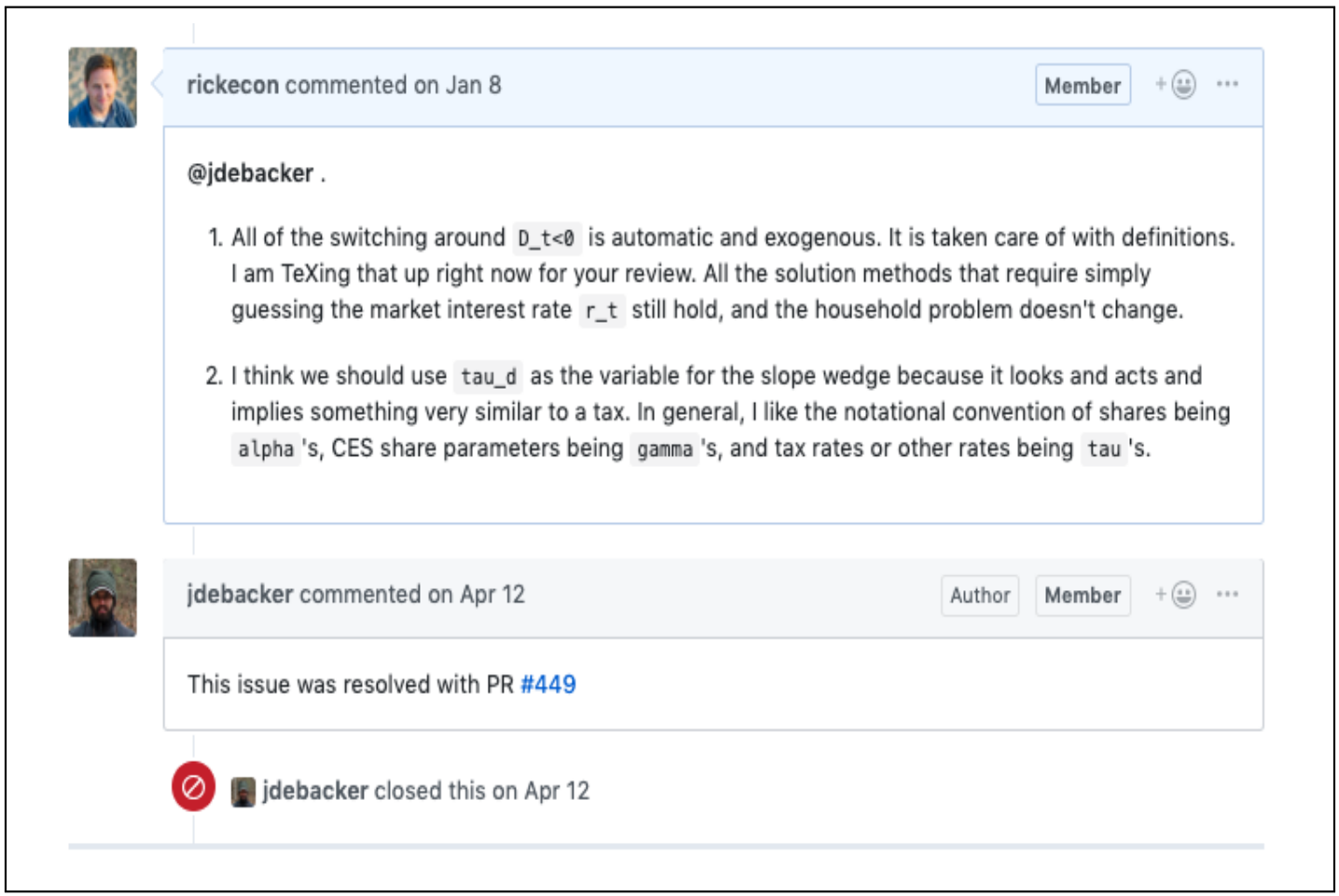
But I think that the interest rate wedge between the government borrowing rate `r_{g,t}` and the private rental rate `r_t` rule should have the following form that includes an indicator function.

$$r_{g,t} = \begin{cases} (1-\tau_d)r_t - \mu_d & \text{if} \quad r_t \geq \frac{\mu_d}{1-\tau_d} \\ 0 & \text{if} \quad r_t \in \left[0, \frac{\mu_d}{1-\tau_d}\right) \\ r_t & \text{if} \quad r_t < 0 \end{cases}$$

This is because I think it makes sense for the wedge to exist only when the private rate is positive (the bottom case). And I don't think it makes sense for the government rate to be negative if the private rate is not negative (middle case). The picture of this interest rate wedge is below. Let me know what you think.



GitHub issues and PR comments



rickecon commented on Jan 8 — Member

@jdebacker .

1. All of the switching around `D_t<0` is automatic and exogenous. It is taken care of with definitions. I am TeXing that up right now for your review. All the solution methods that require simply guessing the market interest rate `r_t` still hold, and the household problem doesn't change.

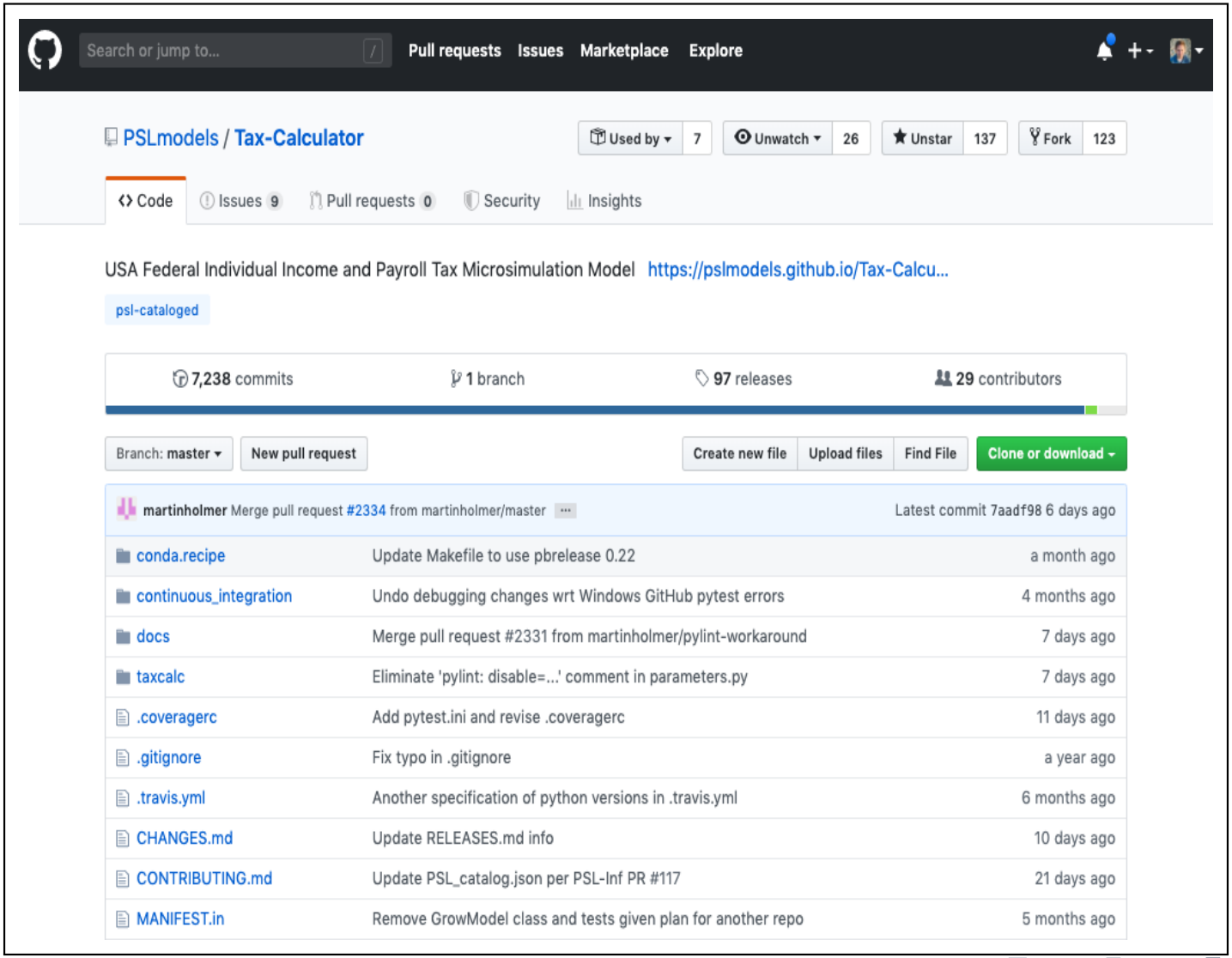
2. I think we should use `tau_d` as the variable for the slope wedge because it looks and acts and implies something very similar to a tax. In general, I like the notational convention of shares being `alpha`'s, CES share parameters being `gamma`'s, and tax rates or other rates being `tau`'s.

jdebacker commented on Apr 12 — Author Member

This issue was resolved with PR #449

jdebacker closed this on Apr 12

Using GitHub repositories



Using GitHub repositories