

# Professional development workshop

Research workflow: <https://github.com/irudik/dyson-workflow>

---

Ivan Rudik

# What is workflow?

# What is workflow?

The design of how you go about doing research

# What is workflow?

The design of how you go about doing research

The goal is to optimize the combination of things you care about:

- Efficiency
- Total research production
- Enjoyability
- Mental expenditures
- Research quality
- Having the process be easy and foolproof

# What is workflow?

The design of how you go about doing research

The goal is to optimize the combination of things you care about:

- Efficiency
- Total research production
- Enjoyability
- Mental expenditures
- Research quality
- Having the process be easy and foolproof

"You need to protect yourself from previous you." -Alex Hollingsworth

# Why does it matter?

Research is a marathon

# Why does it matter?

## Research is a marathon

Paper	Initial Work	Submitting	R&R	Accepted	Published	Total Time (Years)	Time to Submit (Years)
Climate Inertia	Feb-2012	Jun-2014	Dec-2015	Apr-2017	Oct-2017	5.67	2.33
RPS Theory	Feb-2013	Jan-2016	Dec-2017	Feb-2018	Jun-2018	5.33	2.92
RPS Empirics	Feb-2013	Nov-2015	Jan-2017	Feb-2018	Jan-2019	5.92	2.75
Climate Damage Risk	Feb-2014	Jan-2016	Apr-2017	Aug-2019	May-2020	6.25	1.92
Gas Flaring	Dec-2015	Dec-2017	Dec-2019	Apr-2020	Jul-2020	4.59	2
Parks Air Pollution	Jan-2016	Jan-2018	Apr-2018	Jun-2018	Jul-2018	2.5	2
Lead Mortality	Jan-2018	Jul-2019	Dec-2019	Jun-2020	Aug-2021	3.58	1.5
Lead Test Scores	Jan-2018	Dec-2020					2.92
Climate Adaptation	Sep-2018						
Birds and Pollution	Jan-2019	Jun-2020	Jul-2020	Oct-2020	Dec-2020	1.92	1.42
Valuing Forecasts	Jan-2019						
Growth and Biodiversity	Apr-2020						
Geography of Regulation	Apr-2020						
Social Cost of Carbon	Sep-2020						
Climate Networks	Jun-2021						

# Why does it matter?

Projects take **years**



# Why does it matter?

Projects take **years**

This means you're juggling 2-10 projects at a time depending on your style

# Why does it matter?

Projects take **years**

This means you're juggling 2-10 projects at a time depending on your style

Previous slide shows I have **seven** projects going right now (probably 2 are getting attention in any given week)

- 3 working papers
- 2 late stage projects
- 2 early stage projects

# Why does it matter?

Projects take **years**

This means you're juggling 2-10 projects at a time depending on your style

Previous slide shows I have **seven** projects going right now (probably 2 are getting attention in any given week)

- 3 working papers
- 2 late stage projects
- 2 early stage projects

Good workflow is a way to manage this work and get stuff done in a sustainable and efficient way

# Caveats and qualifications

What I'm going to tell you about is what **I** think is important and how **I** do stuff

# Caveats and qualifications

What I'm going to tell you about is what **I** think is important and how **I** do stuff

Others may disagree or would do things in a different way

# Caveats and qualifications

What I'm going to tell you about is what **I** think is important and how **I** do stuff

Others may disagree or would do things in a different way

The key thing is lots of people have thought about this before you so you don't need to reinvent the wheel here: just steal other people's idea and methods that sound good and give them a shot

# Caveats and qualifications

What I'm going to tell you about is what **I** think is important and how **I** do stuff

Others may disagree or would do things in a different way

The key thing is lots of people have thought about this before you so you don't need to reinvent the wheel here: just steal other people's idea and methods that sound good and give them a shot

Good sources: EconTwitter, Hidden Curriculum Podcast, Shapiro and Gentzkow notes, Grant McDermott, Alex Hollingsworth, Patrick Baylis, Stack Exchange

# (Some) key components

- Programming
- Version control
- Code/data management and project organization
- Time management



# Programming

---

# Programming is srs business



**Andrew Barnas**  
@AndrewBarnas



Spending too long on R code and then trying to explain to someone just how much work went into it



1,119 11:31 AM - Aug 25, 2021



**Jodi Beggs** ✓  
@jodiecongirl



R vs. Stata [twitter.com/obrienjohnjack...](https://twitter.com/obrienjohnjack)

**Ian Fleming's Jack O'Brien** @obrienjohnjack

a movie studio gives you \$100 million to write and direct your first movie. BUT you are contractually obligated for it to be a "versus" movie. what two characters do you match up?

77 2:01 PM - Nov 30, 2020



[See Jodi Beggs's other Tweets](#)



# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

- Your previous programming experience

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

- Your previous programming experience
- How much you like programming / learning new languages

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

- Your previous programming experience
- How much you like programming / learning new languages
- The field you work in



# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

- Your previous programming experience
- How much you like programming / learning new languages
- The field you work in
- How much you care about co-authoring

# Programming is srs business

Choice of programming language doesn't matter, but it kind of does

IMO it just a big fixed cost component of research and workflow

What you settle on should depend on at least the following things:

- Your previous programming experience
- How much you like programming / learning new languages
- The field you work in
- How much you care about co-authoring
- How much you're into open science and reproducibility

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

Why?

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

Why?

Both are super easy to learn, and have built-in integrated development environments (IDEs)

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

Why?

Both are super easy to learn, and have built-in integrated development environments (IDEs)

Both are proprietary software so typically they work well, don't have to worry a whole lot about backward compatability etc

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

Why?

Both are super easy to learn, and have built-in integrated development environments (IDEs)

Both are proprietary software so typically they work well, don't have to worry a whole lot about backward compatability etc

Stata has a **very** rich set of packages for economists

# Programming is srs business

Conventional choices for economists: Stata and MATLAB

Why?

Both are super easy to learn, and have built-in integrated development environments (IDEs)

Both are proprietary software so typically they work well, don't have to worry a whole lot about backward compatability etc

Stata has a **very** rich set of packages for economists

MATLAB's package ecosystem is bad but it has first-mover advantage



# Programming is srs business

What are the most common alternatives?

# Programming is srs business

What are the most common alternatives?

R and Julia are seeing large relative gains (small absolute gains) in usage by economists

# Programming is srs business

What are the most common alternatives?

R and Julia are seeing large relative gains (small absolute gains) in usage by economists

Why?

# Programming is srs business

What are the most common alternatives?

R and Julia are seeing large relative gains (small absolute gains) in usage by economists

Why?

Both are **way** better at stuff that is increasingly important in economics

# Programming is srs business

What are the most common alternatives?

R and Julia are seeing large relative gains (small absolute gains) in usage by economists

Why?

Both are **way** better at stuff that is increasingly important in economics

Handling large datasets (feols vs reghdfe), geospatial analysis (sf vs ArcGIS), large-scale quantitative models (CDP vs RLTO)

# Programming is srs business

What are the most common alternatives?

R and Julia are seeing large relative gains (small absolute gains) in usage by economists

Why?

Both are **way** better at stuff that is increasingly important in economics

Handling large datasets (feols vs reghdfe), geospatial analysis (sf vs ArcGIS), large-scale quantitative models (CDP vs RLTO)

Both are open source: free, but generally things aren't quite as polished

# Programming is srs business

Major tradeoffs:

R/Julia have a smaller userbase in econ, by a lot

- Every AEJ: App paper last year had Stata code
- Rudik (2020) is the second(?) AEA journal paper using Julia

But the R/Julia share of economists working on cutting-edge stuff with big data, ML, quant models is larger (e.g. Grant McDermott, Jonathan Dingel, Milena Almagro, Alex Hollingsworth, Ariel Ortiz-Bobea, me)

These network externalities matter

# Programming is srs business

Trade off of your actual time vs computer time

Some large-scale problems are unsolvable with Stata/MATLAB, you might need to learn another language anyway



# Version control

---

# Version control

Version control is how you manage and track changes to your files

# Version control

Version control is how you manage and track changes to your files

Research is a highly non-linear process: you inevitably will need to go back to previous versions of your project to undo changes, fix new errors you can't figure out, etc

# Version control

Version control is how you manage and track changes to your files

Research is a highly non-linear process: you inevitably will need to go back to previous versions of your project to undo changes, fix new errors you can't figure out, etc

There's **many** different methods to versioning, all have different levels of fixed and variable costs you need to incur to use them

# Choose your level of control

Local hard  
disk



Dropbox



Dropbox +  
date naming



Git +  
Github



# Versioning level 1

One way to do version control is to just keep different versions of files on your local hard drive: `file-03-01-2021`, `file-03-02-2021`, `file-final_FINAL2`, `file-myname`, `file-myname-yourname`

# Versioning level 1

One way to do version control is to just keep different versions of files on your local hard drive: `file-03-01-2021`, `file-03-02-2021`, `file-final_FINAL2`, `file-myname`, `file-myname-yourname`

You are technically doing version control, but it's messy, decentralized, and not really systematic

# Versioning level 1

One way to do version control is to just keep different versions of files on your local hard drive: `file-03-01-2021`, `file-03-02-2021`, `file-final_FINAL2`, `file-myname`, `file-myname-yourname`

You are technically doing version control, but it's messy, decentralized, and not really systematic

Don't do this



# Versioning level 1

One way to do version control is to just keep different versions of files on your local hard drive: `file-03-01-2021`, `file-03-02-2021`, `file-final_FINAL2`, `file-myname`, `file-myname-yourname`

You are technically doing version control, but it's messy, decentralized, and not really systematic

Don't do this

Seriously, don't

# Versioning level 2

A better way to do is is through a cloud storage provider like Dropbox, Box, etc

## Versioning level 2

A better way to do it is through a cloud storage provider like Dropbox, Box, etc

This centralizes your file locations across all co-authors and machines, provides a nice backup in case of a local hard drive failure

# Versioning level 2

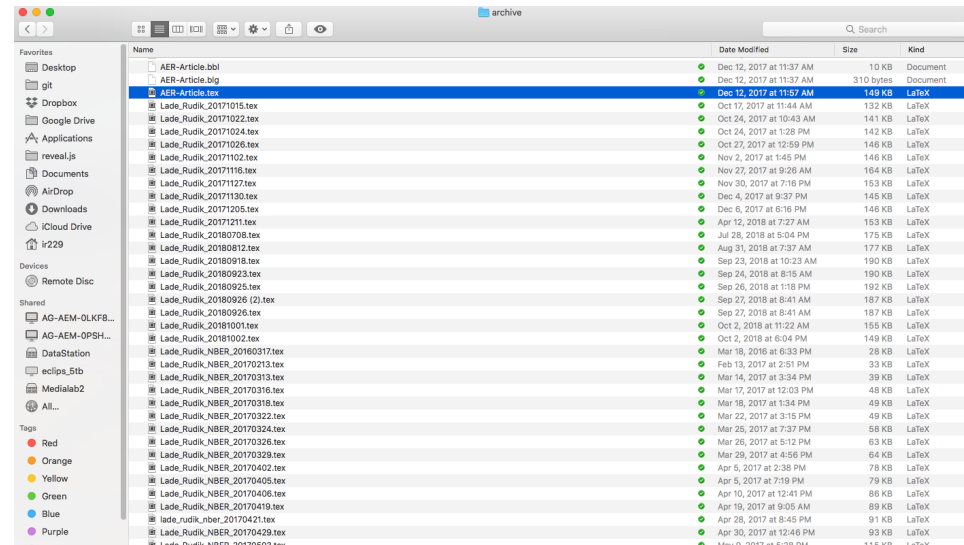
A better way to do it is through a cloud storage provider like Dropbox, Box, etc

This centralizes your file locations across all co-authors and machines, provides a nice backup in case of a local hard drive failure

If you pay for the fancier versions you can also rewind to previous versions up to X days ago

# Versioning level 3

A third way to do this is to combine 1 and 2: use a cloud provider but then use some sort of file naming system to keep track of different versions



The problem is then you have lots of files, little understanding of changes made between versions

# Versioning level 4

The ideal way to handle version control is with proper version control software along with cloud-based services

# Versioning level 4

The ideal way to handle version control is with proper version control software along with cloud-based services

This is typically **Git** using along with **GitHub**

# Versioning level 4

The ideal way to handle version control is with proper version control software along with cloud-based services

This is typically **Git** using along with **GitHub**

Git keeps track of changes of your files and stores these changes in a history along with messages, tags, author identifiers, etc



# Versioning level 4

The ideal way to handle version control is with proper version control software along with cloud-based services

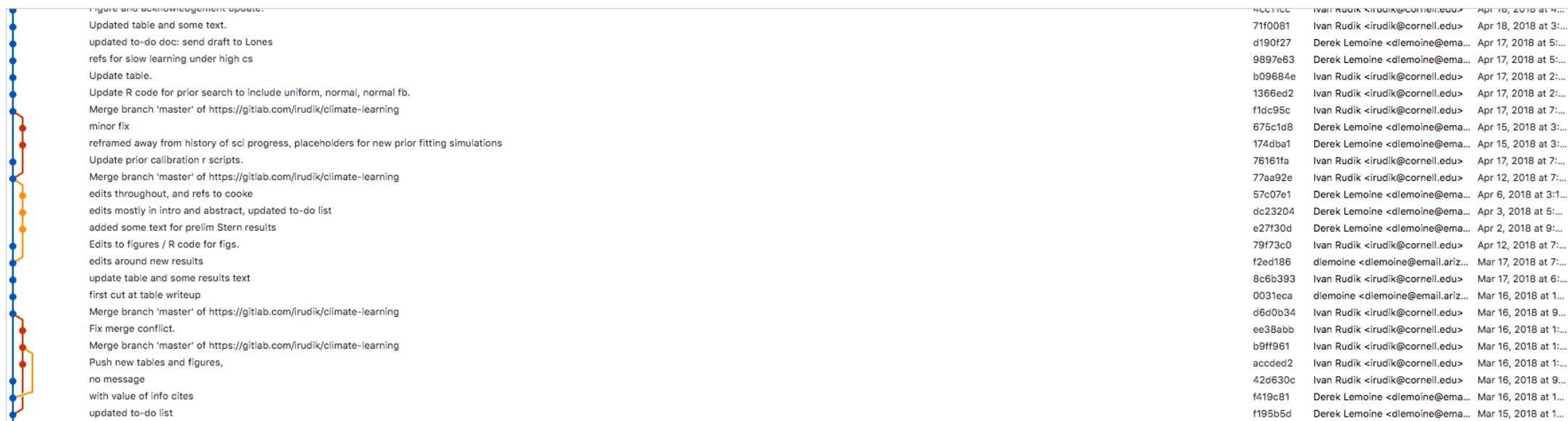
This is typically **Git** using along with **GitHub**

Git keeps track of changes of your files and stores these changes in a history along with messages, tags, author identifiers, etc

This makes it very easy to find who made changes and when and to see exactly what changes were made

# Versioning level 4: Git

Quick example of one of the programs (SourceTree) you can use to implement Git/GitHub workflow:



# Versioning level 4: Git

Quick example of a GitHub repo:

The screenshot shows the GitHub interface for the repository 'irudik / climate-trade'. The repository is private and has 15 issues, 4 branches, and 2 tags. The main content area displays a list of files and folders, including 'CDP\_codes\_four\_sectors', 'archive', 'counterfactual', 'counterfactual\_cdp', 'latex\_extras', 'references', '.gitignore', and several R scripts. The right sidebar contains sections for 'About' (Microfoundations for aggregate climate damages), 'Releases' (2 tags), 'Packages' (No packages published), and 'Contributors' (3 contributors: irudik, weiliangt).

irudik / climate-trade Private

Watch 2 Star 0

Code Issues 15 Pull requests Actions Projects Wiki Security Insights Settings

master 4 branches 2 tags

Go to file Add file Code

About

Microfoundations for aggregate climate damages.

Releases

2 tags

Create a new release

Packages

No packages published

Publish your first package

Contributors 3

irudik Ivan Rudik

weiliangt Weiliang Tan

File/Folder	Commit Message	Time Ago
CDP_codes_four_sectors	Merge branch 'master' into overleaf-2021-01-15-1813	8 months ago
archive	update file names	last month
counterfactual	fix single shock	26 days ago
counterfactual_cdp	Something with the data? not xbilat.	8 months ago
latex_extras	Updates from Overleaf	8 months ago
references	Updates from Overleaf	8 months ago
.gitignore	Update gitignore with renv.	6 months ago
0.00_initialize_cf_data.R	update file names	last month
1.00_download_weather_data.R	update file names	last month
1.01_global_map.R	update file names	last month
1.02_weather_aggregation_weights.R	update file names	last month
1.03_weather_aggregation_data.R	update file names	last month

# Code/data management and project organization

---

# Data/code management

How does data/code organization improve workflow?

# Data/code management

How does data/code organization improve workflow?

Less stuff you need to keep in your brain

# Data/code management

How does data/code organization improve workflow?

Less stuff you need to keep in your brain

If everything is the same, then working with your projects becomes largely automatic, no trying to remember where certain types of data are located, how you organize your coding scripts in run-order, etc

# Data/code management

How does data/code organization improve workflow?

Less stuff you need to keep in your brain

If everything is the same, then working with your projects becomes largely automatic, no trying to remember where certain types of data are located, how you organize your coding scripts in run-order, etc

Makes life easier to use best practices like relative file paths, `here :: here`, etc



# One way to organize data and code

- Parent folder
  - Code folder
    - Data cleaning code
    - Analysis code
  - Data folder
    - Raw data
    - Clean data
  - Output folder
  - LaTeX folder

# Code script organization

A project will likely have multiple files for data cleaning/analysis

# Code script organization

A project will likely have multiple files for data cleaning/analysis

This largely makes sense: having more modular / function-oriented code is generally better

# Code script organization

A project will likely have multiple files for data cleaning/analysis

This largely makes sense: having more modular / function-oriented code is generally better

When working with other people (or your future self), it's smart to order your files:

- 01.first-file-to-run
- 02.second-file-to-run
- ...
- 000.project-functions
- 000.RUN-ALL-SCRIPTS.sh

# Centralize and optimize your environments

One recommendation I have is to centralize and optimize your coding environment

# Centralize and optimize your environments

One recommendation I have is to centralize and optimize your coding environment

You will often have to use multiple languages

# Centralize and optimize your environments

One recommendation I have is to centralize and optimize your coding environment

You will often have to use multiple languages

Built-in IDEs for many languages are mediocre or just straight up bad (hi Stata)

# Centralize and optimize your environments

One recommendation I have is to centralize and optimize your coding environment

You will often have to use multiple languages

Built-in IDEs for many languages are mediocre or just straight up bad (hi Stata)

Having a single coding environment that is easy to modify and adapt will make you much more efficient at your job



# Centralize and optimize your environments

One recommendation I have is to centralize and optimize your coding environment

You will often have to use multiple languages

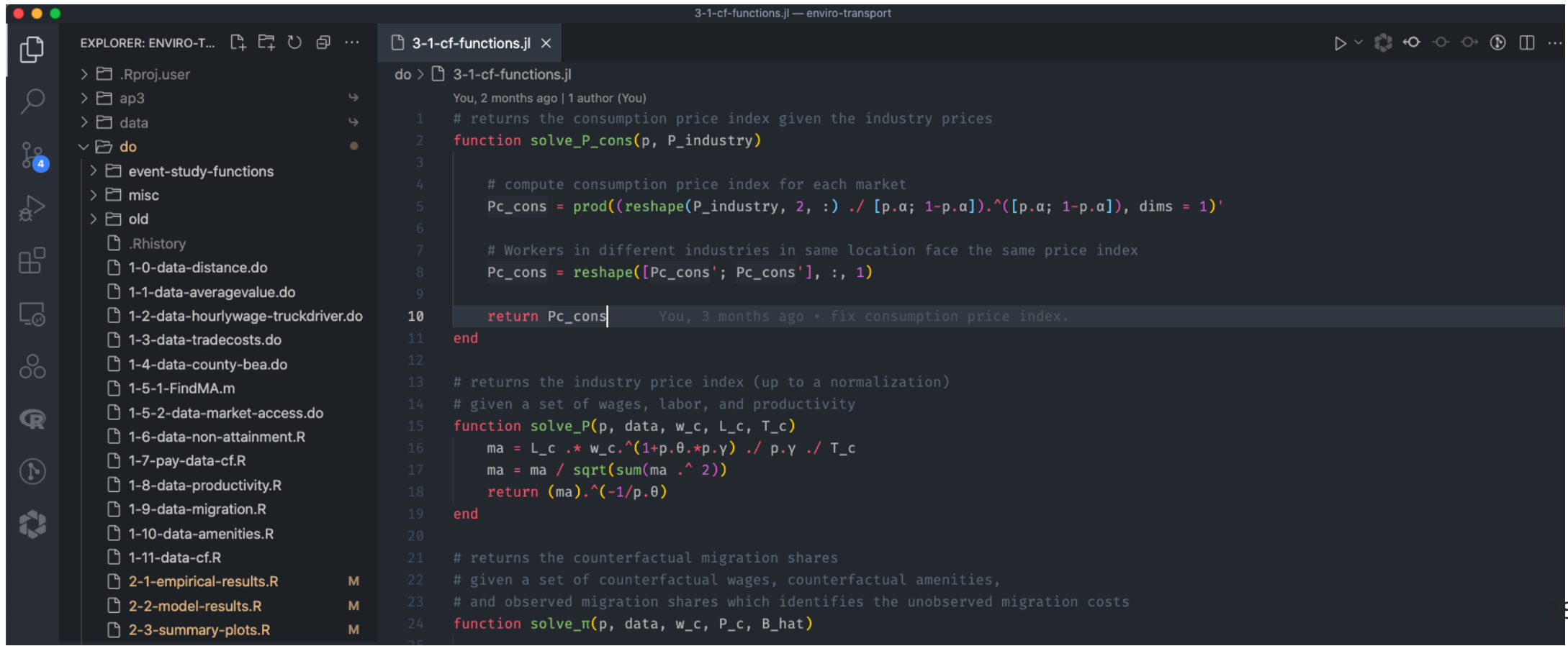
Built-in IDEs for many languages are mediocre or just straight up bad (hi Stata)

Having a single coding environment that is easy to modify and adapt will make you much more efficient at your job

There's a lot of options (Atom, emacs, Vim, Sublime Text) but I'll tell you about **Visual Studio Code**

# Centralize and optimize your environments: VSCode

**Visual Studio Code (VSCode)** is probably the most widely used development environment



```
3-1-cf-functions.jl — enviro-transport

EXPLORER: ENVIRO-T...
> .Rproj.user
> ap3
> data
> do
  > event-study-functions
  > misc
  > old
    .Rhistory
    1-0-data-distance.do
    1-1-data-averagevalue.do
    1-2-data-hourlywage-truckdriver.do
    1-3-data-tradecosts.do
    1-4-data-county-bea.do
    1-5-1-FindMA.m
    1-5-2-data-market-access.do
    1-6-data-non-attainment.R
    1-7-pay-data-cf.R
    1-8-data-productivity.R
    1-9-data-migration.R
    1-10-data-amenities.R
    1-11-data-cf.R
    2-1-empirical-results.R M
    2-2-model-results.R M
    2-3-summary-plots.R M

do > 3-1-cf-functions.jl
You, 2 months ago | 1 author (You)
1 # returns the consumption price index given the industry prices
2 function solve_P_cons(p, P_industry)
3
4     # compute consumption price index for each market
5     Pc_cons = prod((reshape(P_industry, 2, :) ./ [p.a; 1-p.a]).^([p.a; 1-p.a]), dims = 1)'
6
7     # Workers in different industries in same location face the same price index
8     Pc_cons = reshape([Pc_cons'; Pc_cons'], :, 1)
9
10    return Pc_cons
11 end
12
13 # returns the industry price index (up to a normalization)
14 # given a set of wages, labor, and productivity
15 function solve_P(p, data, w_c, L_c, T_c)
16     ma = L_c .* w_c.^(1+p.θ.*p.γ) ./ p.γ ./ T_c
17     ma = ma / sqrt(sum(ma.^2))
18     return (ma).^(-1/p.θ)
19 end
20
21 # returns the counterfactual migration shares
22 # given a set of counterfactual wages, counterfactual amenities,
23 # and observed migration shares which identifies the unobserved migration costs
24 function solve_π(p, data, w_c, P_c, B_hat)
```

# Centralize and optimize your environments: VSCode

How does using VSCode (or Atom/Sublime/etc) improve workflow?

1. The environment is the same no matter whether I'm writing in R/Stata/Julia/MATLAB/Python/LaTeX,whatever

# Centralize and optimize your environments: VSCode

How does using VSCode (or Atom/Sublime/etc) improve workflow?

1. The environment is the same no matter whether I'm writing in R/Stata/Julia/MATLAB/Python/LaTeX, whatever
2. VSCode is **highly** customizable, has useful linting, syntax highlighting, and my fav: bracket matching

# Centralize and optimize your environments: VSCode

How does using VSCode (or Atom/Sublime/etc) improve workflow?

1. The environment is the same no matter whether I'm writing in R/Stata/Julia/MATLAB/Python/LaTeX,whatever
2. VSCode is **highly** customizable, has useful linting, syntax highlighting, and my fav: bracket matching
3. VSCode has remote development capabilities so you can connect to BioHPC, RedCloud, AWS, etc

# Centralize and optimize your environments: VSCode

How does using VSCode (or Atom/Sublime/etc) improve workflow?

1. The environment is the same no matter whether I'm writing in R/Stata/Julia/MATLAB/Python/LaTeX,whatever
2. VSCode is **highly** customizable, has useful linting, syntax highlighting, and my fav: bracket matching
3. VSCode has remote development capabilities so you can connect to BioHPC, RedCloud, AWS, etc
4. Fully integrated with Git

# Centralize and optimize your environments: VSCode

How does using VSCode (or Atom/Sublime/etc) improve workflow?

1. The environment is the same no matter whether I'm writing in R/Stata/Julia/MATLAB/Python/LaTeX,whatever
2. VSCode is **highly** customizable, has useful linting, syntax highlighting, and my fav: bracket matching
3. VSCode has remote development capabilities so you can connect to BioHPC, RedCloud, AWS, etc
4. Fully integrated with Git
5. Lots of useful and customizable hot keys for mass editing

# Tasks and communication

Organizing tasks and project communication is important for actually getting things done



# Tasks and communication

Organizing tasks and project communication is important for actually getting things done

There are **many** ways to do this

# Tasks and communication

Organizing tasks and project communication is important for actually getting things done

There are **many** ways to do this

Here's a few suggestions:

1. **GitHub Issues**
2. **Slack**
3. Other task management providers: **Asana**, **Wrike**, etc (possibly with Slack)

# Tasks and communication

If you're a Git user I recommend using **GitHub Issues**:

1. Everything's on GitHub
2. Commits/pull requests etc can be referenced in task-specific issues so changes are easy to find
3. GitHub has a Projects feature that is similar to Asana/Wrike

# Writing

If you've got co-authors I recommend [Overleaf](#)

# Writing

If you've got co-authors I recommend **Overleaf**

It's cloud-based, integrates with Dropbox and GitHub, and has a RTF editor for your non-LaTeX co-authors

# Writing

If you've got co-authors I recommend **Overleaf**

It's cloud-based, integrates with Dropbox and GitHub, and has a RTF editor for your non-LaTeX co-authors

Solo projects: probably use Dropbox or Git+GitHub

# Meta-reproducibility

If you're interested in reproducibility of projects you should take Lars Vilhuber's workshop at the beginning of fall

# Meta-reproducibility

If you're interested in reproducibility of projects you should take Lars Vilhuber's workshop at the beginning of fall

Reproducibility is just ensuring that your project always works when you run it on as many types of machines and operating systems as possible



# Meta-reproducibility

If you're interested in reproducibility of projects you should take Lars Vilhuber's workshop at the beginning of fall

Reproducibility is just ensuring that your project always works when you run it on as many types of machines and operating systems as possible

When I say meta-reproducibility I mean the ability to quickly reproduce the foundation of a new (reproducible) project without much effort

# Meta-reproducibility

If you're interested in reproducibility of projects you should take Lars Vilhuber's workshop at the beginning of fall

Reproducibility is just ensuring that your project always works when you run it on as many types of machines and operating systems as possible

When I say meta-reproducibility I mean the ability to quickly reproduce the foundation of a new (reproducible) project without much effort

The key idea is that the framework for your code, writing, etc should all be the same and easy to implement

# Meta-reproducibility

Spinning up a new project with people should be **easy** (besides the research part)

# Meta-reproducibility

Spinning up a new project with people should be **easy** (besides the research part)

Easy way to set up the initial framework:

- Create a GitHub template repo
  - <https://github.com/cornell-seere/repo-template>
- Create a GitHub repo for your Dropbox folder for non-versioned files
  - <https://github.com/cornell-seere/dropbox-template>

If you do this you can spin up a full, clean project in ~1 minute

# Cross-project reproducibility

You're going to use the same functions a lot of the time across projects, e.g.  
here's my base plotting template:

```
main_theme ←  
  theme_minimal() +  
  theme(  
    legend.position = "none",  
    title = element_text(size = 24),  
    axis.text.x = element_text(size = 24), axis.text.y = element_text(size = 24),  
    axis.title.x = element_text(size = 24), axis.title.y = element_text(size = 24),  
    panel.grid.minor.x = element_blank(), panel.grid.major.y = element_blank(),  
    panel.grid.minor.y = element_blank(), panel.grid.major.x = element_blank(),  
    panel.background = element_rect(fill = "#ffffff", colour = NA),  
    plot.background = element_rect(fill = "#ffffff", colour = NA),  
    axis.line = element_line(colour = "black")  
  )
```

# Time management and organization

---

---

**“Work expands so as to fill the  
time available for its completion”**

**Parkinson's Law**

# Time management

DO WHAT YOU LOVE  
AND YOU'LL ~~NEVER~~  
~~WORK A DAY IN YOUR~~  
~~LIFE~~ WORK SUPER  
FUCKING HARD ALL  
THE TIME WITH NO  
SEPARATION OR ANY  
BOUNDARIES AND ALSO  
TAKE EVERYTHING  
EXTREMELY PERSONALLY

@ADAMJK



# Time management

There's a lot of fancy books and stuff on this

# Time management

There's a lot of fancy books and stuff on this

First-order thing that matters (imo): work when you're most productive

# Time management

There's a lot of fancy books and stuff on this

First-order thing that matters (imo): work when you're most productive

I schedule no meetings or obligations in the morning because that's when I like to do research

# Time management

There's a lot of fancy books and stuff on this

First-order thing that matters (imo): work when you're most productive

I schedule no meetings or obligations in the morning because that's when I like to do research

Second-order thing for me: I typically have some task in mind that I want to complete in the next 2-3 hours (e.g. get half my presentation slides done, finishing writing the code to aggregate the CPS to state-level, etc)