

Quantitative Social Science in the Age of Big Data and AI

Lecture 1: Toolkits: Introduction to R, RStudio, Git, Github and LLMs

Zhaopeng Qu

Hopkins-Nanjing Center, Nanjing University

March 05 2025



Roadmap

Today we will cover the following topics:

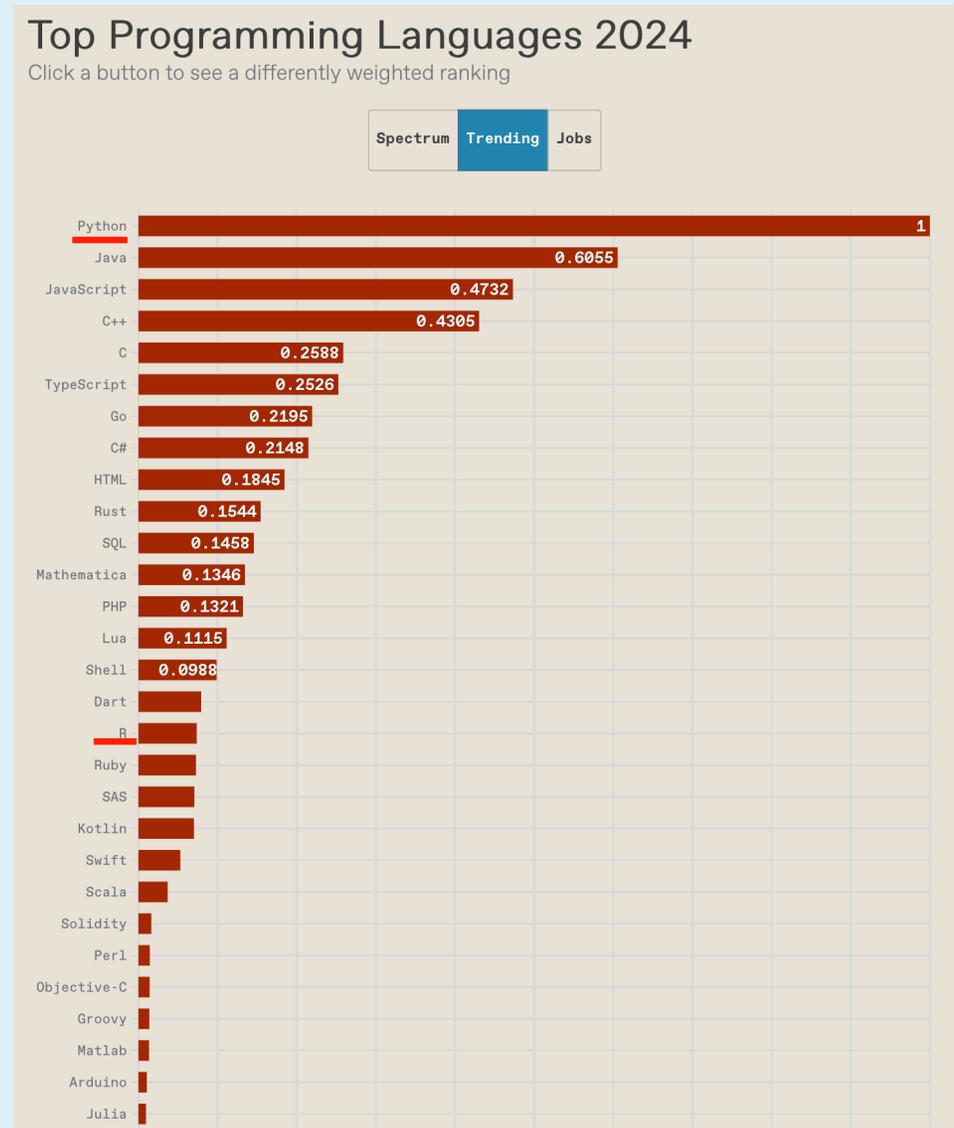
- **Programming Language Selection:**
 - Install and Setup for R
 - Install and Setup for RStudio
- **Version Control, Git and GitHub**
 - Why do we need it?
 - How to use it?
- **Introduction to LLMs Models and Applications**
 - Which one should you use?
 - How to use it in R/RStudio?

Programming Languages

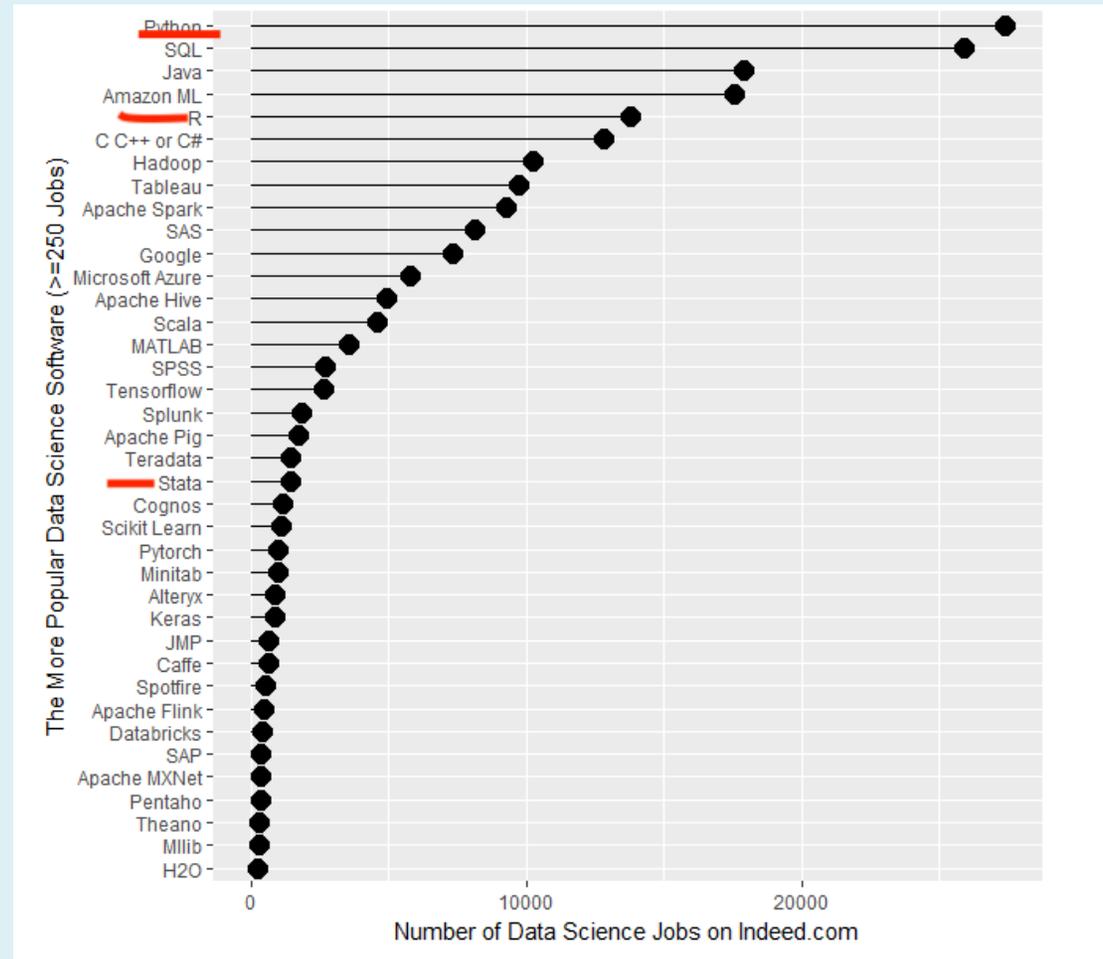
Top Programming Languages by IEEE in 2019-2024

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

2019



Popular Softwares by Jobs in 2019



2019

Popular Softwares by Jobs in 2024

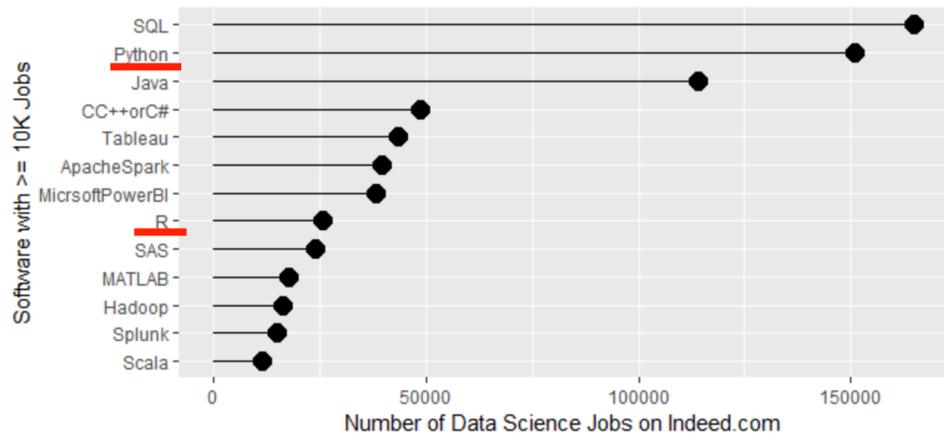
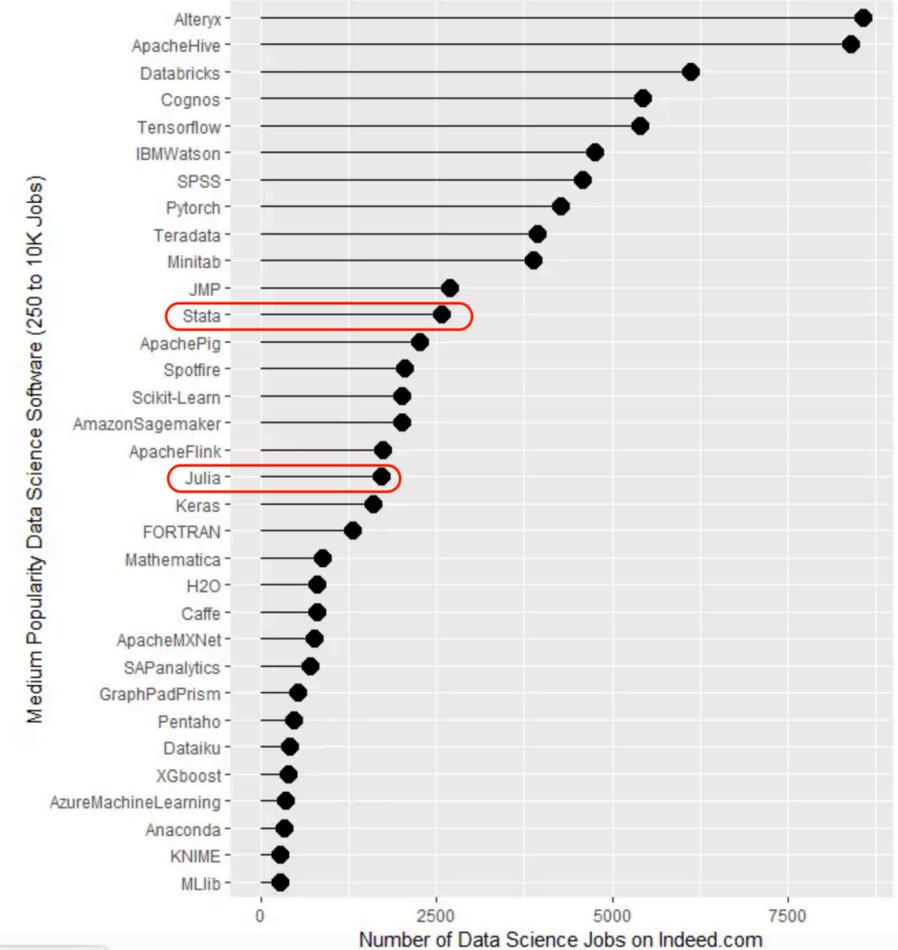


Figure 1a. Number of data science jobs for the more popular software ($\geq 10,000$ jobs).

2024



2024

Languages: Python, R, Julia and Stata

Python:

- **Pros:**
 - **General-purpose:** web development, data analysis, machine learning, etc.
 - **Rich libraries:** numpy, pandas, scikit-learn, tensorflow, etc.
 - **Community:** Stack Overflow, GitHub, etc.
- **Cons:**
 - **Speed:** slower than C/C++/Fortran
 - **Memory:** not good at memory management
 - **Syntax:** not as clean as R
 - **IDE:** not as good as RStudio

Julia

- **Pros:**
 - **Speed:** as fast as C/C++/Fortran
 - **Syntax:** as clean as Python
 - **Memory:** good at memory management
 - **Community:** Stack Overflow, GitHub, etc.
- **Cons:**
 - **IDE:** not as good as RStudio
 - **Libraries:** not as rich as Python and R

Languages: Python, R, Julia and Stata

R:

- **Pros:**
 - **Statistical language:** designed for data analysis
 - **Rich libraries:** dplyr, ggplot2, tidyr, etc.
 - **Community:** Stack Overflow, GitHub, etc.
 - **IDE:** RStudio
- **Cons:**
 - **Speed:** slower than C/C++/Fortran
 - **Memory:** not good at memory management
 - **Syntax:** not as clean as Python
 - **General-purpose:** not as good as Python

Stata

- **Pros:**
 - **Statistical language:** designed for statistics and econometrics
 - **Community:** Stata Forum, etc. mailist
 - **IDE:** Stata
- **Cons:**
 - **Speed:** slower than C/C++/Fortran
 - **Memory:** not good at memory management
 - **Syntax:** the most clean even better than python
 - **General-purpose:** not as good as Python

Which programming language should you learn?

- Of course, **more is better than less**. But we'd better focus on **one or two languages** at the beginning.
 - It depends on what you already know and what you want to do.
- For Data Science & AI, **Python** and **R** are the most popular languages.
- For Econometrics, **R** and **Stata** are the most popular languages.
- For High Performance Computing, **Julia** is the best choice.
- For this course, we will focus on **R** and **Python**.

Installing and Setting Up R

Installing R and RStudio

R:

- **Windows:** Download and install R from [CRAN](#).
- **Mac:** Download and install R from [CRAN](#).

IDE for R

- **Integrated Development Environment(IDE)** is a software application that provides comprehensive facilities to computer programmers for software development.
 - R is the "engine" of the car, and IDE is the navigator of the car. And **RStudio** is the most popular IDE for R.
 - Download and install RStudio from [RStudio](#).
- **NOTE:** You should *install R first and then install RStudio*.

Using IDE(RStudio) v.s Terminal(R)



R



RStudio

Source: [ModernDive\(2025\)](#)

Learning R within RStudio

Install `swirl` to learn R within RStudio

```
install.packages("swirl")  
library(swirl)  
swirl()
```

- There are many courses to learn R within RStudio, you can search for them on [swirl](#)
 - **Course repository** on github: [swirl_courses](#)
 - **Course Network** of swirl: [course network](#)

Learning R within RStudio

Two Companion Packages to our QSS textbook

Install `qss-swirl` package to as a companion package to QSS

```
swirl::install_course_github("kosukeimai", "qss-swirl")  
swirl()
```

Install `qsslearn` package to as a companion package to QSS

```
remotes::install_github("kosukeimai/qss-package", build_vignettes = TRUE)  
remotes::install_github("rstudio/learnr")  
remotes::install_github("rstudio-education/gradethis")  
remotes::install_github("mattblackwell/qsslearnr")
```

- Start the Tutorial

```
learnr::run_tutorial("00-intro", package = "qsslearnr")
```

Learning R within RStudio: Homework

Finish QSS tutorial 0 in the `qsslearnr` package.

QSS Tutorial 0: Introduction to R

Basics of R

Working with Data

Submit

Start Over

Working with Data

✓ Working with real data

Next, we are going to start working with real data: estimates of world population (in thousands). A vector of data called `world.pop` has been loaded with this lesson. The first element is for the year 1950 up to the last element for 2010. You can see that we create the vector by using the `c()` function which concatenates multiple values together into one vector. We enter the data one value at a time, each separated by a comma.

- Print the `world.pop` data by simply typing it into a line of code.

```
R Code   
```

```
1 ## create the world.pop data
2 world.pop <- c(2525779, 3026003, 3691173, 4449049, 5320817, 6127700, 6916183)
3
4 ## print the world.pop data
5
6 print(world.pop)
7
8
```

```
[1] 2525779 3026003 3691173 4449049 5320817 6127700 6916183
```

✓ Indexing and subsetting

Vectors are just a series of objects in R that are all stored together in a specific order. What if we want to access a specific value in the vector? Well, for that, we can use the indexing and subsetting tools in R. Specifically, we will use the the square brackets, `[]` to access specific values within the vector.

- Use the brackets to access the fourth entry in the `world.pop` data.

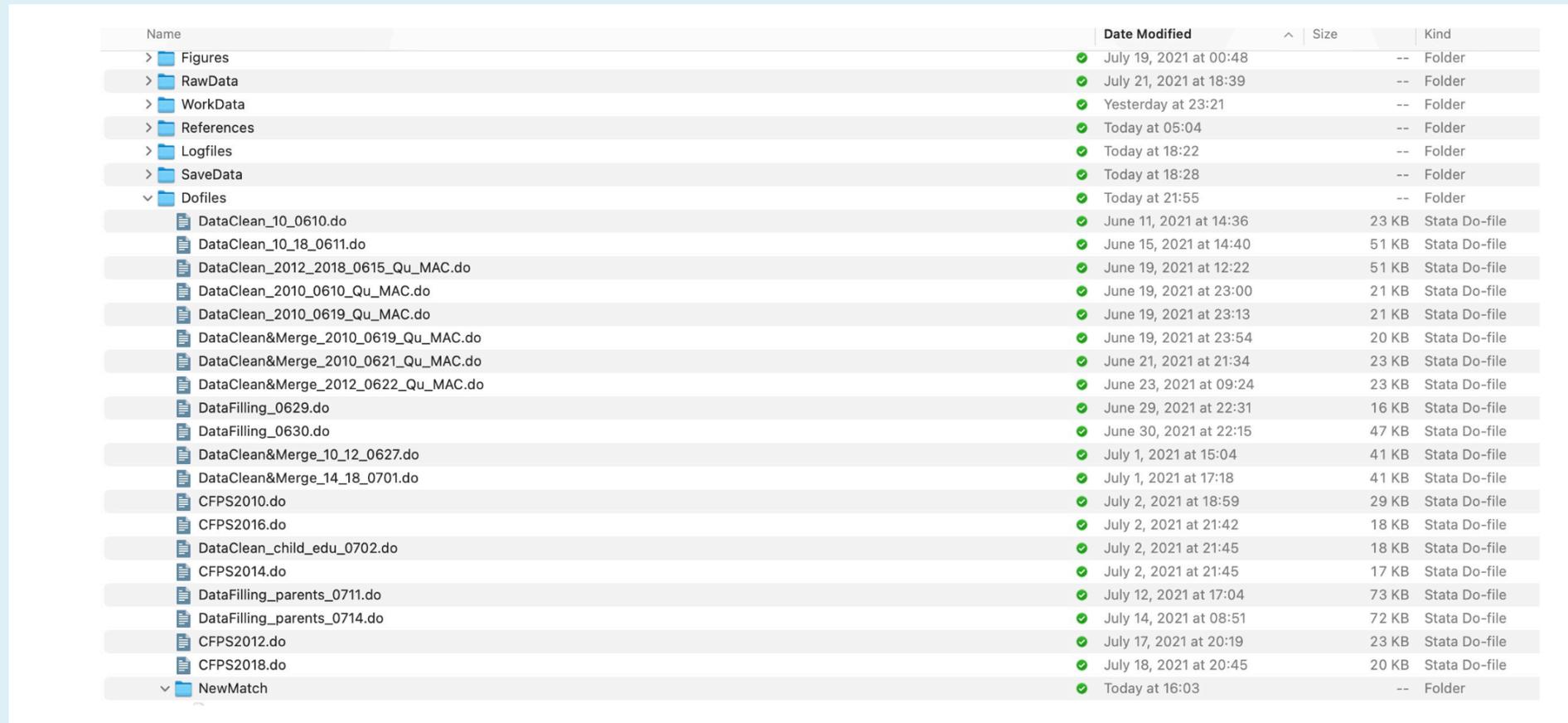
```
R Code   
```

```
1 ## access and print the 4th value of world.pop
2 world.pop[4]
3
```

Introduction to Version Control, Git and GitHub

Why do we need Version Control?

- Once upon a time, there was a researcher who worked on a project for a while.



Name	Date Modified	Size	Kind
> Figures	✓ July 19, 2021 at 00:48	--	Folder
> RawData	✓ July 21, 2021 at 18:39	--	Folder
> WorkData	✓ Yesterday at 23:21	--	Folder
> References	✓ Today at 05:04	--	Folder
> Logfiles	✓ Today at 18:22	--	Folder
> SaveData	✓ Today at 18:28	--	Folder
▼ Dofiles	✓ Today at 21:55	--	Folder
DataClean_10_0610.do	✓ June 11, 2021 at 14:36	23 KB	Stata Do-file
DataClean_10_18_0611.do	✓ June 15, 2021 at 14:40	51 KB	Stata Do-file
DataClean_2012_2018_0615_Qu_MAC.do	✓ June 19, 2021 at 12:22	51 KB	Stata Do-file
DataClean_2010_0610_Qu_MAC.do	✓ June 19, 2021 at 23:00	21 KB	Stata Do-file
DataClean_2010_0619_Qu_MAC.do	✓ June 19, 2021 at 23:13	21 KB	Stata Do-file
DataClean&Merge_2010_0619_Qu_MAC.do	✓ June 19, 2021 at 23:54	20 KB	Stata Do-file
DataClean&Merge_2010_0621_Qu_MAC.do	✓ June 21, 2021 at 21:34	23 KB	Stata Do-file
DataClean&Merge_2012_0622_Qu_MAC.do	✓ June 23, 2021 at 09:24	23 KB	Stata Do-file
DataFilling_0629.do	✓ June 29, 2021 at 22:31	16 KB	Stata Do-file
DataFilling_0630.do	✓ June 30, 2021 at 22:15	47 KB	Stata Do-file
DataClean&Merge_10_12_0627.do	✓ July 1, 2021 at 15:04	41 KB	Stata Do-file
DataClean&Merge_14_18_0701.do	✓ July 1, 2021 at 17:18	41 KB	Stata Do-file
CFPS2010.do	✓ July 2, 2021 at 18:59	29 KB	Stata Do-file
CFPS2016.do	✓ July 2, 2021 at 21:42	18 KB	Stata Do-file
DataClean_child_edu_0702.do	✓ July 2, 2021 at 21:45	18 KB	Stata Do-file
CFPS2014.do	✓ July 2, 2021 at 21:45	17 KB	Stata Do-file
DataFilling_parents_0711.do	✓ July 12, 2021 at 17:04	73 KB	Stata Do-file
DataFilling_parents_0714.do	✓ July 14, 2021 at 08:51	72 KB	Stata Do-file
CFPS2012.do	✓ July 17, 2021 at 20:19	23 KB	Stata Do-file
CFPS2018.do	✓ July 18, 2021 at 20:45	20 KB	Stata Do-file
▼ NewMatch	✓ Today at 16:03	--	Folder

Why do we need Version Control?

File Name	Status	Date	Size	Type
Writing	Folder	July 17, 2021 at 21:59	--	Folder
农民工工资趋同_0511_Qu—无分组.docx	✓	May 15, 2021 at 13:00	2.4 MB	Microso...(.docx)
农民工工资趋同_0515_Qu—无分组.docx	✓	May 16, 2021 at 12:03	3 MB	Microso...(.docx)
城市职工的结果.docx	✓	May 16, 2021 at 13:41	279 KB	Microso...(.docx)
农民工工资趋同_0515_by_QU.docx	✓	May 16, 2021 at 18:59	1.9 MB	Microso...(.docx)
农民工工资趋同_0516_by_QU.docx	✓	May 18, 2021 at 12:19	1.9 MB	Microso...(.docx)
农民工工资趋同_0518_qu.docx	✓	May 19, 2021 at 11:34	1.3 MB	Microso...(.docx)
农民工工资趋同_20210606_qu02.docx	✓	June 7, 2021 at 02:09	827 KB	Microso...(.docx)
农民工工资趋同_20210606_中国工业经济.docx	✓	June 7, 2021 at 08:19	829 KB	Microso...(.docx)
农民工工资趋同_20210606.docx	✓	June 7, 2021 at 08:27	826 KB	Microso...(.docx)
农民工工资趋同_20210606.pdf	✓	June 7, 2021 at 08:28	2.2 MB	PDF Document
农民工工资趋同_20210607.docx	✓	June 7, 2021 at 08:50	826 KB	Microso...(.docx)
最终版本	Folder	July 17, 2021 at 09:51	--	Folder
农民工工资趋同_20210607_final.docx	✓	June 7, 2021 at 08:50	826 KB	Microso...(.docx)

My Messy Drafts

- Different versions of the same file are a mess for the data scientist.
- **Replicability** and **Reproducibility** are the most important things for social science research.
 - Many journals require you to provide your code and data for replication.

Why do we need Version Control?

- **Version Control** is a system that records changes to a file or set of files over time so that you can recall specific versions later.
 - Like a tracking system for your files.
- **Git** is a distributed version control system that allows you to track changes in your files. And it can be deployed on
 - a local server(PC)
 - or cloud server or web-based platform.
- **Guidebook: Pro Git 2nd Edition, [free download](#) and [Chinese Edition](#).**

Common Misconceptions

|"Github is a data science tool for sharing data"

- It's built more for version controlling plain-text **code** (that analyzes data) and **text** (that documents it).

|"Git is only relevant for software developers"

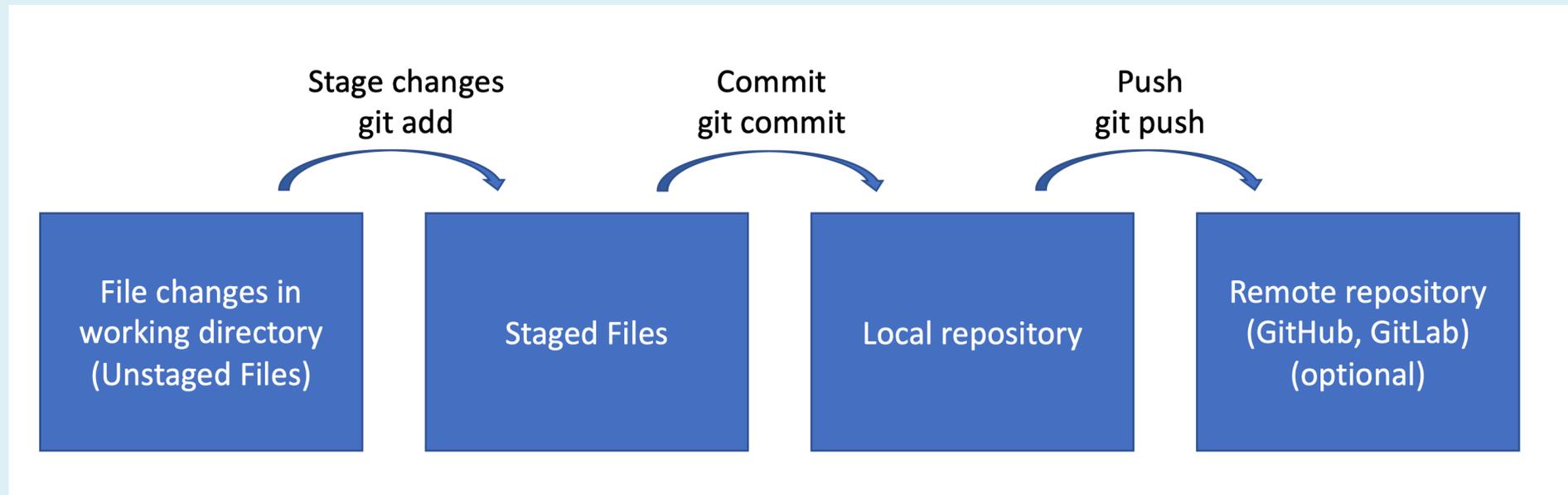
- It also has distinct benefits for the applied researchers' workflow

|"Version control is *only* useful for collaborative projects"

- No, in fact putting your **solo work** under version control at first.
 - then move on to more complicated collaborations.

Git: The Basic Workflow

1. *do work in your own working directory (your own PC).*
2. *stage files(暂存文件), adding snapshots of them to your staging area.*
3. *commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.*
4. *if you have a remote repository, push your commit to it.*



Git Installing and Setup

- Open your **terminal** for MacOS or **Command Prompt** for Windows and type the following command:
 - `git --version`.
- **Git installing**
 - **Windows:** Download and install Git from [here](#).
 - **Mac:** Download and install Git from [here](#).

Git Installing and Setup

- **Git Setup:** Open your terminal for MacOS or Command Prompt for Windows and type the following command:
 - `git config --global user.name "Your Name"`
 - `git config --global user.email "Your email"`
 - `git config --global --list`
- **Git Help:** Type `git help` or `git help -a` for more information.

Using GitHub

- You can use **Git** on your local server, but you can also use **GitHub** to manage your files.
- **GitHub** is a web-based platform that provides hosting for software development and version control using **Git**.
 - Other providers are **GitLab**, **Bitbucket**, **Gitee(domestic version)** etc.
- Register an account
 - **Sign up**: Create an account on **GitHub**.
 - **Educational Account**: Apply for an educational account on **GitHub**.
 - **The Big Benefits**: **Free Copilot Access** and Unlimited Private Repositories.

My GitHub Website

byelenin

Overview Repositories 65 Projects Packages Stars 120

Search: Type / to search



Frank Qu
byelenin

Economist and Number Cruncher. Using data to explore the economic and social logic in China.

[Edit profile](#)

77 followers · 470 following

Nanjing University
Nanjing, P. R. China
qu@nju.edu.cn
https://byelenin.github.io
@zhaopengqu

Achievements

Popular repositories

[MBAcourse](#) Public
Forked from TaddyLab/MBAcourse
Material from the Big Data course at Chicago Booth
TeX 2 1

[stat545.stat.ubc.ca](#) Public
Forked from Econometrics/stat545.stat.ubc.ca
Repository that produces the new STAT 545/547 @ UBC website
HTML 2

[course-website](#) Public
Forked from svmiller/course-website
This is a Jekyll template, patterned off my main Jekyll template, for organizing a course website and hosting it on Github.
CSS 2

[lecture-notes-2021](#) Public
Forked from ml4econ/lecture-notes-2021
Lecture notes for "Machine Learning for Economists", spring 2021 at HUIJ
HTML 2 3

[SublimeStataEnhanced](#) Public
Forked from andrewheiss/SublimeStataEnhanced
Plugin that adds support for Stata 11-14 for Sublime Text 2 and 3
Python 1

[taddylab.github.io](#) Public
Forked from TaddyLab/taddylab.github.io
website
CSS 1

183 contributions in the last year

Contribution settings

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												
Wed												
Fri												

Learn how we count contributions

Less More

2024
2023
2022
2021
2020

Git and GitHub Management

- **Git** is originally a command-line tool, and you can use it in the following ways:

1. Command Line Interface (CLI)

- **Git Bash**: A command-line tool for Windows.
- **Terminal**: A command-line tool for Mac.

2. **GUI**: A graphical user interface for Git, such as **GitHub Desktop**, **SourceTree**, **GitKraken**,

3. **IDE**: Integrated Development Environment, such as **RStudio**, **VS Code**...

4. **Web-based Platform**: A web-based platform for Git management, such as **GitHub**, **GitLab**, **Bitbucket**...

- As a beginner, I recommend you use **RStudio** / **GitHub Web** / **GitHub Desktop** / for Git management.

GitHub Practice and Homework

- Lab: Using RStudio to manage your GitHub repository.
- **Homework:** Practice using RStudio to manage your GitHub repository. You and your partners will create a repository on GitHub and practice the following operations:
 - **Create a repository:** Create a repository on GitHub.
 - **Clone a repository:** Clone a repository to **your local PC**
 - **Commit changes:** Commit changes to the repository
 - **Push changes:** Push changes to the repository in the cloud
 - **Pull changes:** Pull changes from the repository in the cloud
 - ...

AI tools: LLMs Models and Applications

Introduction to Large Language Models (LLMs)

- **Large Language Models (LLMs)** are a type of artificial intelligence that can generate human-like text.
- LLMs are trained on large amounts of text data and can generate text that is indistinguishable from human-written text.
- Generative AI's poster child is ChatGPT, but the new wave of AI extends way beyond large language models:
 - Images
 - Music
 - Coding
 - Video
 - Math
 - 3D ...

Introduction to Large Language Models (LLMs)

- The basic idea behind LLMs is to **predict the next word** in a sentence based on the words that came before it.
- The math behind LLMs is complex, but the basic idea can be summarized as a probability of the next word given the previous words:

$$P(W) = P(w_1, w_2, w_3, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)P(w_4|w_1, w_2, w_3) \dots P(w_n|w_1, w_2, \dots, w_{n-1})$$
$$\Rightarrow P(w_n|w_1, w_2, \dots, w_{n-1}) = \frac{P(w_1, w_2, \dots, w_n)}{P(w_1, w_2, \dots, w_{n-1})}$$

- The W is the entire sentence, and $w_1, w_2, w_3, \dots, w_n$ are the words in the sentence.
- The **right answer** are the words that have the highest probability of being the next word in the sentence.

Introduction to Large Language Models (LLMs)

"There is only one true heroism in the world: to see the world as it is, and to love it"

- We can break this sentence into individual words (tokens). Here are the first few and last few tokens:

$w_1 = \text{"There"}$

$w_2 = \text{"is"}$

$w_3 = \text{"only"}$

$w_4 = \text{"one"}$

\vdots

$w_{18} = \text{"to"}$

$w_{19} = \text{"love"}$

$w_{20} = \text{"it"}$

- The probability of the last four can be expressed as:

$$P(\text{it} | \text{There is only one true heroism in the world : to see the world as it is, and to love}) = \frac{P(\text{There is only one true heroism in the world : to see the world as it is, and to love it})}{P(\text{There is only one true heroism in the world : to see the world as it is, and to love})}$$

- "him", "her", "it"... are all likely to be the next word in the sentence but "it" has the highest probability.

Introduction to Large Language Models (LLMs)

- The more data you have, the better the model can predict the next word. And the algorithm of LLMs is also important.
 - GPT-4 use **45TB** of data to train the model, over 300 trillion tokens and has **175B** parameters.
- As of now, we cannot confidently say hi-red[LLMs truly understand the meaning of the text].
 - LLMs excel at **memorizing** text and **predicting** the next word.
 - However, this is not the same as **understanding** the meaning of the text.
- That is why sometimes LLMs can generate **nonsense** and **misinformation**.
 - **Hallucination**: generating false or misleading information.
 - **Bias**: reflecting the biases in the training data.
- **NOTE**: **treating LLMs as a good assistant for your work** instead of a **replacement** for your work.

Prompt Engineering(提示词工程学)

- Since LLMs are predicting the next word based on the previous words, so the "previous words" are very important.
- **Prompt Engineering** is a technique to improve the performance of LLMs by using **prompt** to guide the LLMs to generate the text we want.
 - **Basic Prompt Structure:** Create a raw template for the prompt.
 - **Language:** Be concise and direct.
 - **Role Assignment:** Assign a specific role to the LLMs for better context.
 - **Examples:** Provide examples to help the LLMs understand the task.
 - **Chain-of-Thought:** For complex tasks, use a chain-of-thought approach to guide the LLMs in generating the desired text.
 - **Avoid Hallucination:** Ensure the LLMs do not generate false or misleading information.
 - **Feedback Loop:** Use feedback from the LLMs to improve the prompt.
- The specific design needs to be tailored to the specific task and your actual situation, so we won't go into detail here.

LLMs in the real world

- As of March 2025, The most popular LLMs in the world market:

Company	Name	Models	APP	API	Open Sources	Service in China
OpenAI	ChatGPT	GPT o3	Yes	Yes	No	No
Google	Gemini	Gemini 2.0	Yes	Yes	No	No
Microsoft	Copilot	ChatGPT o3 and others	Yes	Yes	No	Yes
Anthropic's	Claude	Claude 3.7 sonnet	Yes	Yes	No	No
Meta	Llama	Llama 3.3	Yes	Yes	Yes	Yes
DeepSeek	DeepSeek	DeepSeek R1	Yes	Yes	Yes	Yes
Alibaba	Qwen	Qwen 2.5	Yes	Yes	Yes	Yes

- Copilot is the only foreign company whose services are accessible to Chinese users.**
- DeepSeek** and **Qwen** are both Chinese companies who provide open sources for LLMs.

Popular Chinese LLMs

- As of August 2024, there are over **200 players** in China LLMs market. Some general models providers:

Company	Name	APP	API	Open Sources
百度	文心一言 4.0	Yes	Yes	No
阿里云	通义千问 2.1	Yes	Yes	No
腾讯	混元	Yes	Yes	No
科大讯飞	星火3.5	Yes	Yes	No
月之暗面	Kimi	No	Yes	No
字节跳动	豆包	No	Yes	No
DeepSeek	DeepSeek	Yes	Yes	Yes

Three Types of LLMs

- **Closed source LLMs:**
 - Anything is in the black box.
 - Most LLMs are known as closed source LLMs, like **ChatGPT, Gemini, Claude**, etc.
- **Semi-open source LLMs:**
 - The model is open source, but the training data which is used to train the model remains proprietary.
 - **Deepseek** is a good example.
- **Open source LLMs:**
 - Both model architecture and training data are public.

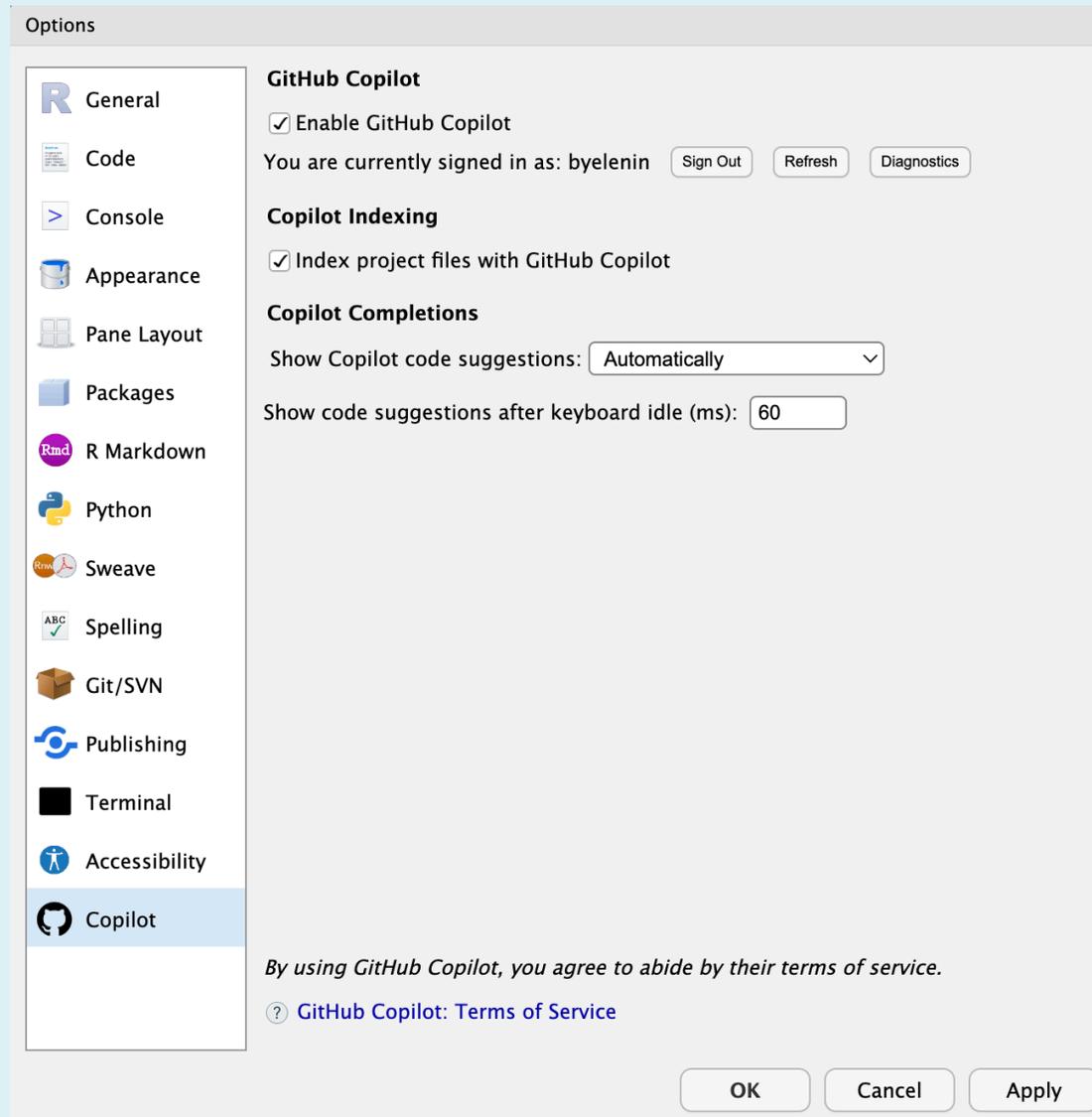
Ways to use LLMs:

- **Web interface** (e.g., OpenAI's GPT-3 Playground)
- **Apps** on devices (e.g., mobile phones, tablets, even PCs)
- **Command-line interface (CLI)**: can be used in terminal, etc.
- **Application Programming Interface (API)**: can be used in R/RStudio, Python, etc.
- **Integrated into applications** (e.g., Microsoft Offices, Google Docs, etc.)
- **Local installation** (e.g., on a server, a PC, etc.) for Open source LLMs

Which model should I use?

- Each product on the previous slide has strengths and weaknesses.
- As a student studying in China, I strongly recommend you to use at least two LLMs:
 - One from outside China: **ChatGPT**, **Claude** or **Copilot**.
 - The other from China: 文心一言, 通义千问, 混元, 星火, **Kimi** or **Deepseek**.
- As our course is about QSS with Big Data & AI, and we mainly use LLMs for coding, therefore I recommend you to use **Copilot** and **Deepseek** as examples.

Using Official Copilot in RStudio



Chatting with Copilot in RStudio

```
install.packages("chattr")
```

```
library(chattr)  
chattr_use("copilot")  
chattr_test()  
chattr("你如何评价南京大学约翰斯霍普金斯大学中美文化研究中心?")
```

```
## 南京大学约翰斯·霍普金斯大学中美研究中心 (The Hopkins-Nanjing Center for Chinese and American Studies) 是由中国的南  
##
```

```
## 1. 学术声誉:
```

```
## - 作为两所知名大学的合作项目，该中心提供了高质量的教育和研究资源。约翰斯·霍普金斯大学以其在国际关系和公共事务领域的强大实  
##
```

```
## 2. 跨文化交流:
```

```
## - 中心为中美学生提供了一个独特的文化交流平台，使他们能够在多元化的环境中学习和生活。这种跨文化体验对于促进双方的理解  
##
```

```
## 3. 课程设置:
```

```
## - 中心提供中英文授课，涵盖国际关系、政治科学、法律、环境政策等多个领域的课程。学生可以通过双语学习，提升自己的语言能力和  
##
```

```
## 4. 网络资源:
```

```
## - 中心的校友网络遍布全球，毕业生在国际组织、政府机构、学术界和商业界都有显著的影响力。
```

Chatting with Copilot in RStudio

```
chattr("How do you evaluate the Hopkins Nanjing Center for Chinese and American Studies?")
```

```
## The Hopkins-Nanjing Center for Chinese and American Studies (HNC) is a collaborative educational institut
##
## 1. Academic Reputation:
## - The HNC is backed by two highly reputable universities. Johns Hopkins University is renowned for its
##
## 2. Cultural Exchange:
## - The HNC offers a unique platform for cross-cultural exchange between Chinese and international stude
##
## 3. Bilingual Education:
## - Courses are taught in both English and Mandarin, allowing students to improve language skills while
##
## 4. Curriculum and Focus:
## - The HNC offers a wide range of courses in international relations, economics, law, environmental pol
##
## 5. Alumni Network:
## - Graduates of the HNC are part of a robust alumni network that spans the globe. Alumni work in divers
##
## 6. Facilities and Environment - Located on the historicou campus of Nanjing University, the HNC fea
##
```

Using all models including Deepseek in RStudio

- Use `ellmer` package to use all LLM models including Deepseek

```
install.packages("ellmer")  
library(ellmer)  
ellmer::use_model("deepseek")
```

- Use `gptstudio` package to use all LLM models including ChatGPT, Gemini, Claude, etc.

```
install.packages("gptstudio")  
library(gptstudio)  
gptstudio::use_model("gpt-4o")
```

- Both require an **API key** to use most of the models, which is usually not free. (We will discuss this later.)

One of the Most Powerful AI Code Editor

Cursor Settings

- General
- Models**
- Features
- Beta

Model Names

Add new model names to Cursor. Often used to configure the latest OpenAI models or OpenRouter models.

claude-3-opus	<input type="checkbox"/>
claude-3.5-sonnet	<input checked="" type="checkbox"/>
cursor-small	<input type="checkbox"/>
gpt-3.5-turbo	<input type="checkbox"/>
gpt-4	<input type="checkbox"/>
gpt-4-turbo-2024-04-09	<input type="checkbox"/>
gpt-4o	<input type="checkbox"/>
gpt-4o-mini	<input checked="" type="checkbox"/>
o1-mini	<input checked="" type="checkbox"/>
o1-preview	<input type="checkbox"/>

+ Add model

One of the Most Powerful AI Code Editor

OpenAI API Key

You can put in [your OpenAI key](#) to use Cursor at public API costs. Note: this can cost more than pro and won't work for custom model features.

Override OpenAI Base URL (when using key) ▾

Anthropic API Key

You can put in [your Anthropic key](#) to use Claude at cost. When enabled, this key will be used for all models beginning with "claude-".

Google API Key

You can put in [your Google AI Studio key](#) to use Google models at-cost.

Azure API Key

Instead of OpenAI's API or pro, you can use Cursor at-cost through the Azure API.

Base URL

Deployment Name

API Key

Saved ✓



Let's Start Our Journey