HKUST
VISLAB

COMP 4462
Data Visualization Tutorial

PAN Ziqi
CHEN Chang

Friday 1 November, 2024

D3.js Introduction

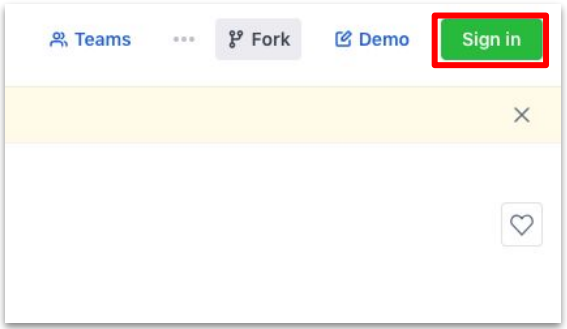
Visualization with D3.js

- SVG (Scalable Vector Graphics)
  - An extension of HTML for representing scalar graphics in XML syntax
  - Available in all the web browsers
- D3.js
  - The most widely used visualization library
  - The library behind Vega, Vega-Lite and Altair
  - Binding data with SVG DOM, marking data points visually onto screen
  - Imperative syntax, compared to the declarative syntax of Vega-Lite and Altair
- Why D3.js
  - ❌Exploring a dataset, use Altair with Python or Tableau instead
  - ❌Embed visualization in web applications, use Vega-Lite instead
  - ✅Make customized plots, customized interactions or transitions
- Cost
  - Much more coding, much easier to make mistakes
  - Check Vega-Lite Gallery and Vega Gallery before committing to D3.js

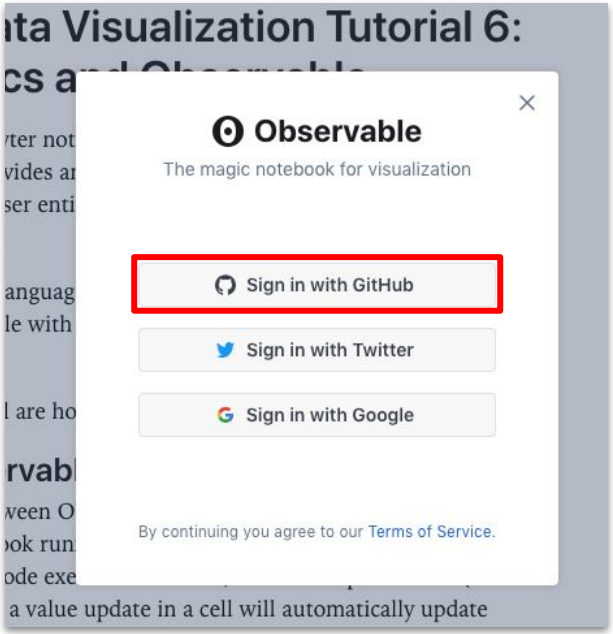
Sign in Observable

1. Go to the [notebook of this tutorial]

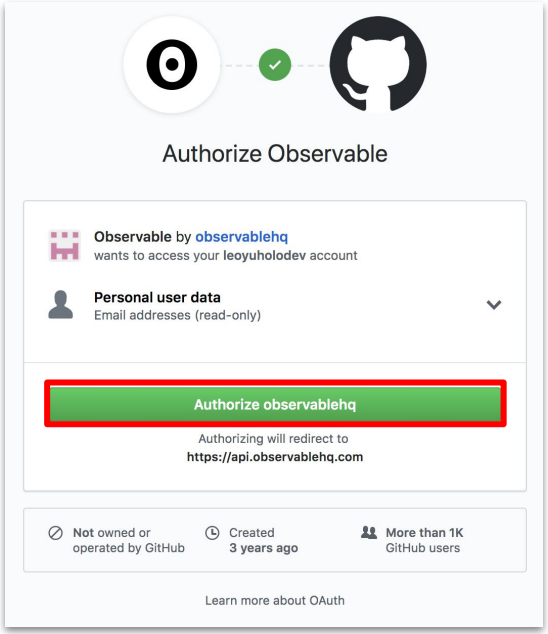
2. Click Sign in



3. Sign in with GitHub (recommended)



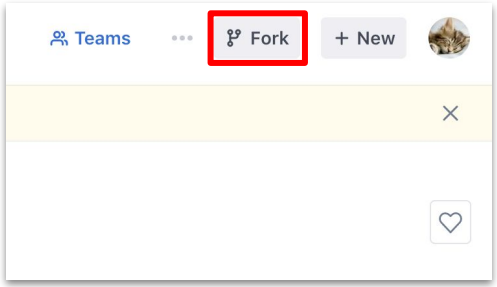
4. Authorize observablehq



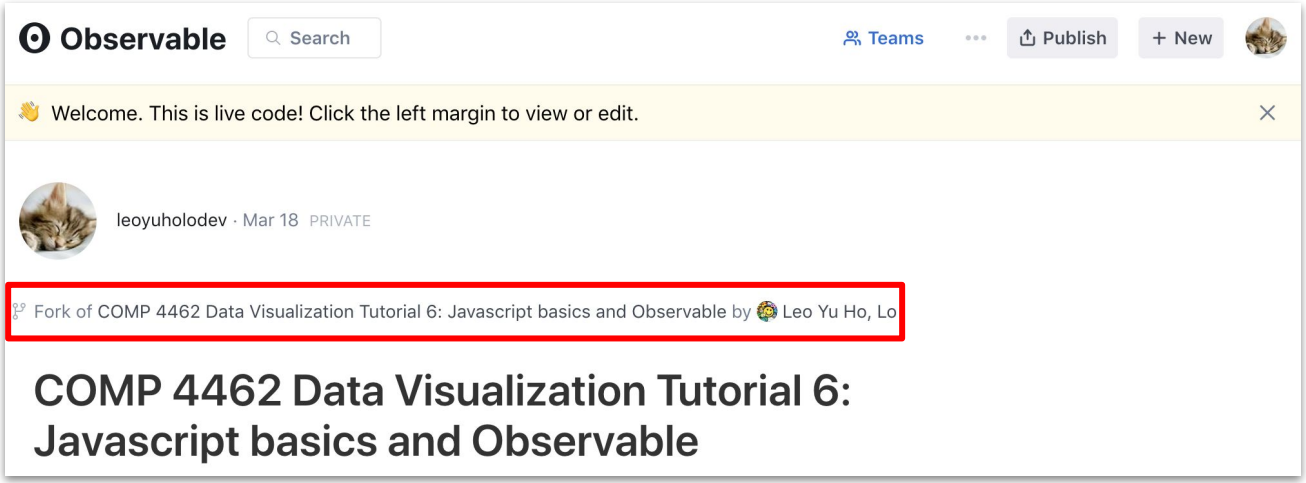
Fork Observable notebook

1. Go to the notebook of this tutorial

2. Click Fork



3. Check if you're working on your copy of the notebook (otherwise, your work will not be saved)



Visualization with D3.js

- See the Observable notebook of this tutorial
- SVG
- Scales
  - Linear scale (numeric, color)
  - Time scale
  - Point scale (categorical)
- Coordinate System
  - Axes
  - Cartesian coordinate (X and Y)
  - Polar coordinate (angular and distance)
- Marks and channel
  - Point: scatter plot
  - Line: (multi-)line chart, parallel coordinate, radar chart
- Selection

Interaction with D3.js

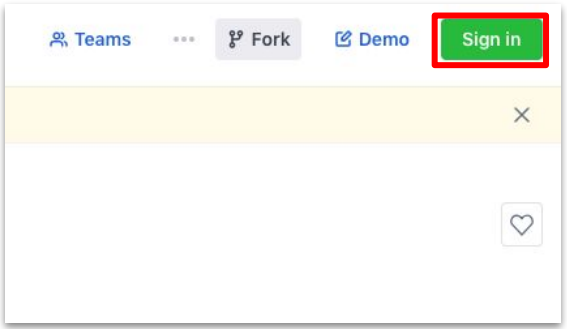
Visualization and Interaction with D3.js

- Interaction with visualization
  - Visualization has well established before the invention of computer
    - But interaction with visualization only available through the use of computers
  - Huge space of possibilities
    - But all successful interaction designs follow **“Overview first, details on demand”**
  - Visualization interactions mostly through mouse
    - Seldomly with keyboard
    - Interaction through touch devices is a grand challenge in data visualization
- Animation
  - Makes interaction smoother, more responsive
  - Keep conceptual consistency, objects enter the scene instead of appear suddenly
  - Motion is a very attention attractive channel
    - It is built-in in our mind to track moving objects (because of primal instincts?)
    - But too much moving objects will overwhelm viewers

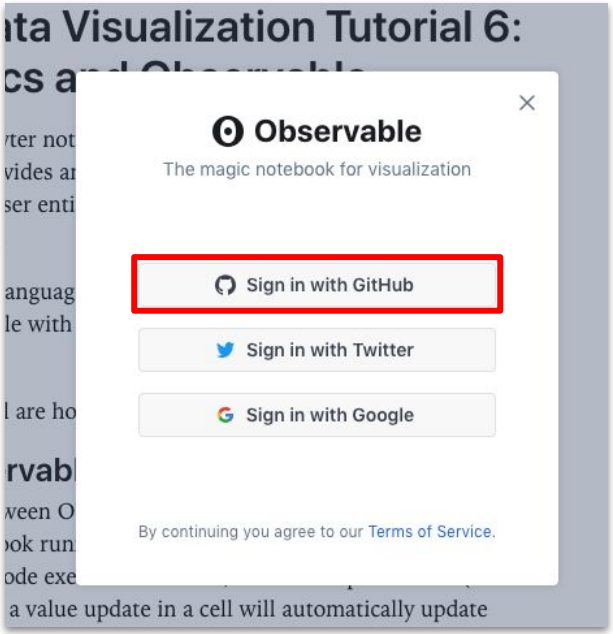
Sign in Observable

1. Go to the [notebook of this tutorial]

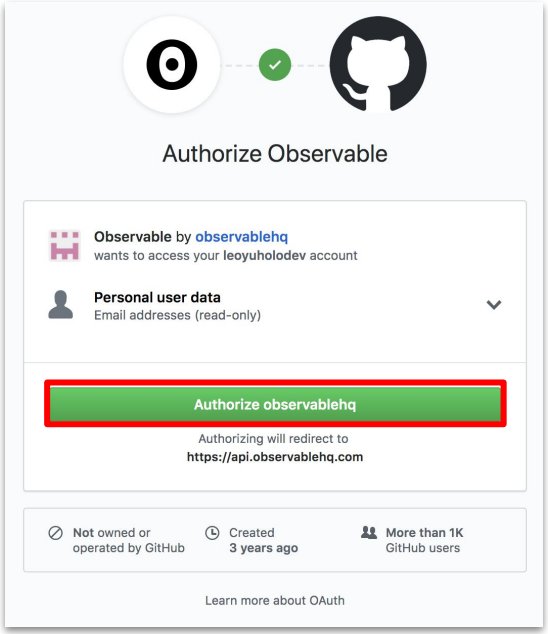
2. Click Sign in



3. Sign in with GitHub (recommended)



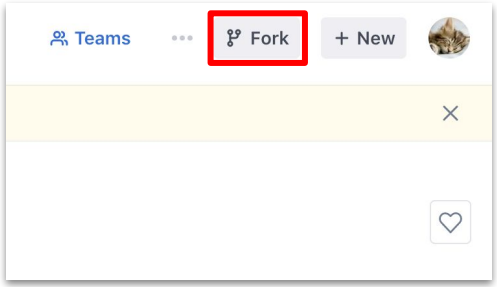
4. Authorize observablehq



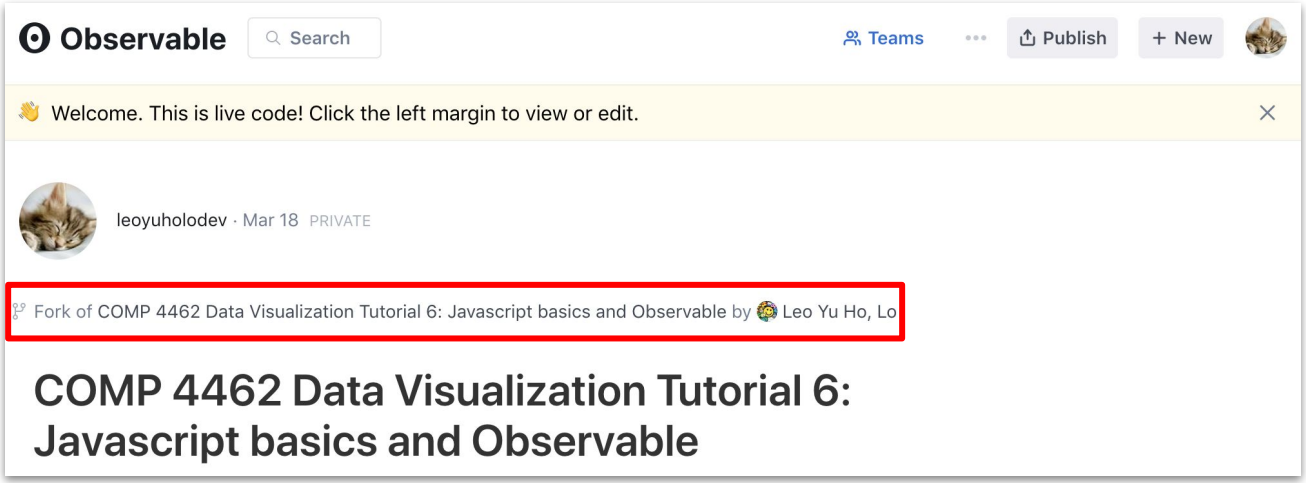
Fork Observable notebook

1. Go to the notebook of this tutorial

2. Click Fork



3. Check if you're working on your copy of the notebook (otherwise, your work will not be saved)



Visualization and Interaction with D3.js

- See the Observable notebook of this tutorial
- Choropleth (maps with color encoding)
- Interaction
  - Overview first, details on demand!
  - Tooltip with <title> element, d3-tip
  - Mouse events: mouseover, mouseout, click
  - Observable inputs: dropdown menu, slider
  - Linked views
- Animation
  - Eyes beat memory!
  - Animation with redraw, D3.js transition
  - Motion encoding, pop-out effect
- Data analysis techniques
  - Daily average over month total
  - How to handle missing data?

More on interactions and D3.js

- More on interactions
  - D3.js: d3-drag, d3-zoom, d3-brush
    - Demos: d3-drag, d3-zoom, d3-brush
  - Vega-Lite:
    - Interactive Plots with Selection in Vega-Lite
  - Altair:
    - Making Charts Interactive in Altair
- Visualizations not covered in tutorials
  - Wordle (a.k.a. Word Cloud)
    - Javascript implementation of wordle by Jason Davies
    - Vega Word Cloud Example
  - Graph visualization
    - D3 in Depth: Layouts and D3 in Depth: Force layout
    - Vega Force Directed Layout Example
    - Besides D3, Gephi is a professional graph visualization tool

Lab Exercise

- Tasks
  - Sign in Observable
  - Open the two observable notebooks and fork them:
    - Part 1: Notebook 1
    - Part 2: Notebook 2
  - Read through the notebook and fill in the “TODO” cells, For Notebook 1, there are five todos, For Notebook 2, there are four todos.
  - Copy the URL of your Observable notebook and submit to Canvas