Working with Big Data

Vancouver Summer Program (VSP)

Daniel Jaramillo Calderon

UBC

July 16, 2024

Notes modified from Josh Catalano (VSP 2023), and Mike Gelbart (CPSC 340)

An Overview of Big Data

- Information is collected and stored at an unprecedented rate.
- Examples:
  - Social Networks: YouTube, Facebook, MOOCs.
  - Credit cards transactions and Amazon purchases.
  - Transportation data (Google Maps, Waze, Uber)
  - Phone call records and speech recognition results.
  - Scientific experiments (biology, astronomy).
  - Video game worlds and user actions.

An Overview of Big Data

- What do you do with all this data?
  - → Too much data to search through it manually.
- But there is valuable information in the data.
  - → How can we use it?
- We will introduce the concepts and coding techniques necessary to make sense of data.

About this Course

General Outcomes

- In this class, you will learn the fundamentals of coding, data preparation, and data analysis.
- No coding experience is assumed.
- When relevant, we will discuss how big data relates to the material and will eventually have the opportunity to work with relatively large datasets.

Introductions

Material

- All of the course materials are stored on the course's GitHub Page.
- We will use Canvas only to keep track of grades, and general announcements

Introductions

Material

- All of the course materials are stored on the course’s GitHub Page.
- We will use Canvas only to keep track of grades, and general announcements

About Me

- Daniel Jaramillo Calderon
- jaramillocalderondc@gmail.com
- PhD Candidate in Economics at UBC

Introductions

Material

- All of the course materials are stored on the course's GitHub Page.
- We will use Canvas only to keep track of grades, and general announcements

About Me

- Daniel Jaramillo Calderon
- jaramillocalderondc@gmail.com
- PhD Candidate in Economics at UBC

What About You?

Course Goals

By the end of this course, students will be able to:

- Code in Python with an understanding of programming fundamentals
- Use GitHub to store and collaborate on code
- Load, clean, manipulate, and visualize data effectively
- Fit regression and classification models on data
- Perform select numerical computations
- Understand some of the math behind these techniques
- Learn about coding on their own more easily

Let’s Start!

What is a Programming Language?

Generally speaking, a programming language is a language that can be interpreted by a computer as a series of commands that it then executes.

- Programming languages tend to be text-based.
- Hundreds of programming languages (e.g. Python, R, Julia, C++)
- The more people use a programming language, the better that language becomes (especially true for open-source languages).

What is Open Source Software?

Open source software stands in contrast to proprietary software.

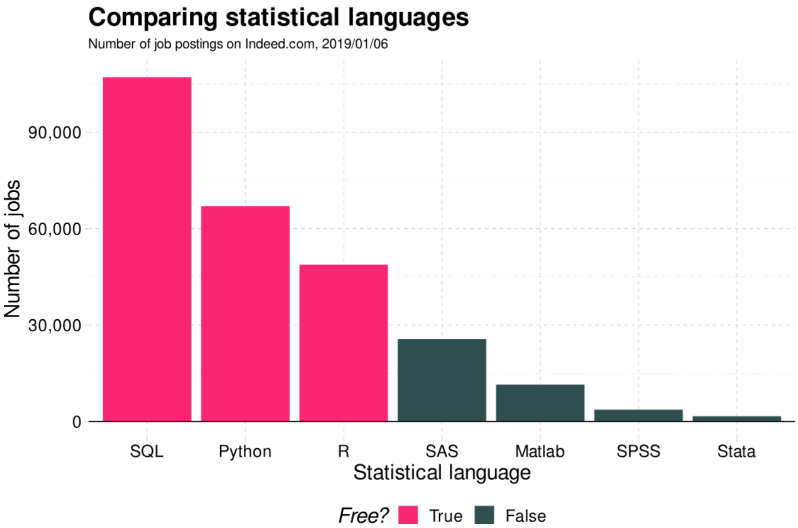
- Free (not paid)
- Licenses still exist and ensure people and organizations have an incentive to contribute
- Easier to collaborate
- Package management systems makes it easier to share your own packages and download packages others have made into your coding environment.

We will be using Python!

Python is a general purpose coding language with many benefits:

- Relatively simple to use and learn
- Readable
- General purpose (can be used for data analysis, websites, web scraping, etc.)
- Very popular – lots of pre-built tools
- Open-source!

Another Reason to Learn Python



Jupyter Notebooks

Jupyter Notebook is an in-browser interactive development environment (IDE). It allows you to display code, markdown-style text, figures, and data in a notebook-like structure.

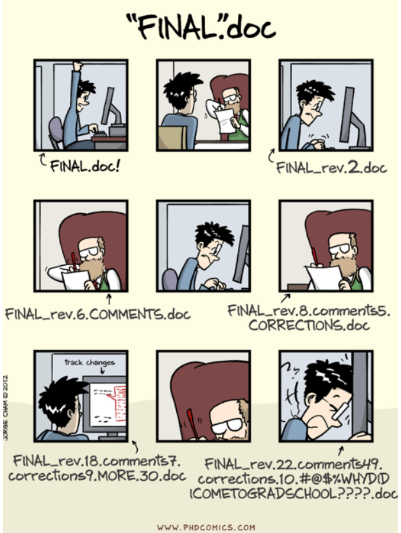
- Supports many coding languages
- Allows for easy code segmentation
- Great for integrating figures and code into lectures, homework assignments, projects, and reports.

Using Jupyter Notebooks

- Cloud Base:
  - → No local installation Required
  - → Several of these. You either can use: UBC Jupyter Open (needs a CWL) or Google Colab (needs a Gmail account).
- Local Based:
  - → Requires installation on your computer
  - → Instructions on how to install here
  - → Worth having in the long-run

Version Control

The Problem


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Git and GitHub

Git

- Git is a distributed version control system. (???)
- Imagine if Dropbox and the ”Track changes” feature in MS Word had a baby. Git would be that baby.
- It is optimized for working on code.

GitHub

- Git and GitHub are distinct things.
- GitHub is an online hosting platform that provides an array of services built on top of the Git system. (Other platforms include Bitbucket and GitLab)
- Just like Python doesn’t require Jupyter Notebooks to run, we don’t need GitHub to use Git... But it will make our lives so much easier.

How are we going to use it?

Code Developers use GitHub (among other Git-centric services) to share and work on projects together. In fact, we are hosting this course on GitHub!

- Code scripts and other files are stored in a repository in the cloud
- Changes to the repository are tracked and can be reversed (version control done through Git)
- Multiple collaborators can simultaneously work on code (Git resolves conflicts between changes)
- Repositories can be easily copied and merged back together.

Today's Lab

In today's lab, we will get all the necessary pieces together so we can start working.

Steps

1. Create a free GitHub account
2. Download GitHub Desktop
3. Log into Jupyter Hub using via Canvas using your Campus-wide Login (CWL) or Google Colab
4. Create and save a basic Jupyter notebook
5. Create your own repository for the course via GitHub desktop
6. Commit and push that repository to the cloud
7. Clone the course repository

Step 1 – Create GitHub Account

To create a GitHub account:

- Go to www.github.com
- Click the “Sign up” button.
- Go through all the sign up steps including confirming your email address
- Go back to GitHub and login – make sure you remember your login details!
- Optional: There is a way for students to get free GitHub pro accounts. Not required for this class, but you can view details here or by searching the ”GitHub Student Developer Pack” on Google.

Step 2 – Download GitHub Desktop

To download GitHub desktop:

- Go to https://desktop.github.com/ or Google GitHub Desktop
- Download the GitHub desktop installer that matches your operating system.
- Run the installer on your computer once it has finished downloading.
- Open GitHub desktop on your computer once it has finished installing.

Step 3 – Access Jupyter Hub

Or Google Colab

To use Jupyter Open, you only need your Campus-wide login.

- Navigate to canvas.ubc.ca.
- Log into canvas using your CWL and click on this course “Working with Big Data”
- On the left side of the screen, there is a menu. Select “Jupyter” from that menu. This will open a new tab in your browser.
- In that new tab, click the “Start My Server” button.
- Once this loads, you will see a Jupyter Notebook and a navigation bar on the side.

Step 4 – Create Jupyter Notebook

Jupyter Open should automatically generate a blank notebook for you

- Click on one of the cells and type “x = 3“ into the first cell without the quotes
- Go to the file drop down menu and click “Save notebook as“. Save the file as “test notebook.ipynb”
- Find the file on your file navigator on the left-hand side pane, right click it, and click ”download” from the drop down menu.

Step 5 – Create Your Own Course Repository

You are now going to create your own repository where you will store your lab assignments, final project, and any other course materials you might want.

- Create a folder for this class on your computer
- Now open GitHub desktop and click on the file drop down menu and select “New repository“
- Title the repository “VSP_your_name“
- For local path, select the folder you created for this class
- After this, press create repository. Then push “publish this repository.“
- Log on to GitHub via a browser to see your repository

Step 6 – Commit and Push a Change to Repository

Now that the repo is created, we will learn how to push changes we make in the local repository to the cloud.

- First, we have to make a change. Move “test notebook.ipynb” into the directory “VSP_your_name”
- Open GitHub Desktop and navigate to the repository using the drop down menu on the left-hand side. You should see the changes you’ve made.
- Click commit – this prepares your files to be pushed to the cloud.
- Click push to finalize these changes and push them to the cloud.
- Log on to GitHub via a browser to see the changes.

Step 7 – Clone Course Repository

Now we are going to clone the course repository

- First, navigate to the course GitHub page. You can do this by clicking the link on canvas or by searching my username “Daniel Jaramillo Calderon” on Google.
- Navigate to my repositories and click on “Working-With-Big-Data-VSP-2024”.
- Copy the URL of this repository
- Open GitHub Desktop, click on the file drop down, and click “clone repository“
- Paste the URL in the first box and set your local path to your class folder (the same folder your repository is in)