**Outline Slide**

* Providing structure, I won’t read through
* Presentation order – PJ, Mason, Joe, PJ
* At each hand-off we’ll share our introduction story to Python

**PJ’s Story**

* Introduced during an astrophysics internship at Lehigh university.
* Studying detached EB’s, known from their distinct light-curve
* Project:
  + Run data through an algorithm on a linux computer
  + Examine resulting flat file
  + Create decision-tree-esque rules for what is and is not a real detached EB
    - Strong signal, period between 1 day and 120 days, high amplitude…
* No coding experience
* Frustrating not knowing coding or language to troubleshoot
* Inefficient code
* 3 months later, outputting pretty plots
* Grown to love community (open source, sharing), stackoverflow, python.

**Purposes**

* General purpose
  + Not specific to stats like R or SaS, used for prototyping
* Popular amongst data scientists
  + Number of users
  + Sharing
  + Ease-of-use
* Walk through others
* Parallel Compute
  + Used for big data
  + Or small-medium data you want distributed for speed reasons
* Python libraries (famous)
  + What is a library?
  + Interchangibly use python library/package

**Comparisons**

* Subjective scoring, agreed on by presenters
* Open source = Free
* Visualizations scored on most common/widely used visualization tools within language
  + Plot9
* Community (stack overflow personal experience)
  + Library volume ( Python >300k as opposed to R >19k )
* Versatility -> way for anything
* Easy to read, start learning, library documentation on websites, not PDF

**Selling Python**

* Machine learning and LLM’s are a hot topic.
  + GPT trained using Python at a high level, low level C/C++
  + **Most-used** language in data analytics, public health to catch up
    - Helps us create tech industry quality processes
* Community/Survey results
  + 4th most popular technology amongst professional devs
  + 3rd most popular over all surveyed
    - Note Javascript/CSS for frontend software devs & web development
  + 1st “most wanted” language – wish they knew (19%)
  + Loved vs. Dreaded
    - ~68% loved for python
    - ~44% for R

**Web-Scraping**

* Notable libraries
* Questions to ask:
  + Does this website already have a pre-built API I can use?
    - Census API Key -> request directly without knowing URLs, types of HTTP requests to perform to get your data
  + Am I allowed here? Robots.txt
  + Number or requests / Min. DDOS (distributed denial of service) attack
    - Threading/multiprocessing
  + Will this website change and how frequently will it?

**Web-Scraping Walkthrough**

* Show web-page we’ll be scanning
* Show robots.txt
* Requests
  + Most popular api scraping
  + GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
  + Used in conjunction with beautifulsoup
  + Note response output
* Selenium
  + Automated web-navigation. Seemingly takes over your browser
  + Not used in tech industry as much, but useful for clicking through pages, doing more human-esque parsing
  + Features
    - Find page elements using underlying HTML / CSS structures
    - Send keys to elements
    - Click elements (left or right click)
    - Navigate multiple windows
    - Work in “headless” mode
  + Applications
    - Website you need to do a sign in (note code-privacy, store passwords safely)
    - Used for WA-GISAID

**Machine Learning**

* Popular libraries
  + Deep learning:
    - Tensorflow developed by google, pytorch serves similar functionality – competing.
    - Pytorch generally considered easier to pick up.
  + Not a solution for everything
    - Using a simpler alternative or deferring to previous research often sufficient.

**Machine Learning (cont)**

* Python makes ML easy, so briefly go over ideas to consider:
  + Preprocessing steps, training data key to good ML process
  + CICD, also known as MLOps, guides life-cycles of great models
    - Often overlooked
    - Accounts for data drift over time

**Machine Learning Application**

* Record Linkage between negative covid electronic labs and our disease reporting system
* >10,000 new labs per day, needed to be linked to positive labs
  + Older process probabilistic matching
  + Manual review team reviews thousands of pairs each day

**ML Walkthrough**

* Skip preprocessing step, included as a python jupyter notebook in our Github folder.
* View data
* ML models only fed numeric data, similarity scores
* Could be done in 2 commands
* Visualizations

**Scaling with python**

* Briefly touch on why scalability is important
* Point to libraries

**Pyspark**

* SQL nature
* Can operate on any machine, very scalable to cloud computing platforms like databricks
* Explain negative labs process