Exercise 1.1 Intro to Freelance and Python Tools

1. Project Timeline

		October					November					
	27	28	28	29	30	31	1	2	3	4	5	6
Workspace Setup	■											
Python Workspace		■										
Configure Git and SSH			■									
Test environment Setup			■	■								
Implementing Python Libraries into a Virtual Environment					■	■						
Execute a Data Scrape on a Website to Collect Web Data							■	■				
Install NLP libraries (NLTK or spaCy).									■			
Text Mining NLP (NetworkX, Plotly)										■	■	
Final Report												■

2. Reflect on what challenges you might encounter while working on this project as a freelancer.

✓ What are the pros and cons of working on your own timeline?

Working as a freelancer offers freedom and flexibility, but it also comes with challenges. One major difficulty is staying motivated and disciplined without the structure of a traditional workplace. It can be easy to lose focus when there's no supervisor setting deadlines. Another challenge is balancing multiple tasks, such as client communication, project planning, and actual coding or analysis work.

The main advantage of working on my own timeline is the ability to organize my day according to my personal rhythm. I can work when I feel most productive and take breaks when needed. However, the downside is that without external pressure, deadlines can easily slip if I don't manage my time carefully.

✓ What strategies will you use to hold yourself accountable?

I plan to set clear daily and weekly goals, use tools like Trello or Google Calendar to track progress, and regularly review my results. Breaking larger tasks into smaller steps will help me maintain focus and motivation throughout the project.

✓ Write 150 to 200 words about your thoughts on this Exercise's content in a text document.

This exercise presents an opportunity to combine data analysis, natural language processing (NLP), and visualization skills to explore historical relationships between countries in the twentieth century. As a freelance data analyst working with the Institute for Public Policy to provide an analytical overview and visualization of the interrelations between countries, I would begin the project by scraping relevant data from reliable online sources, such as Wikipedia. Through text mining and NLP techniques,
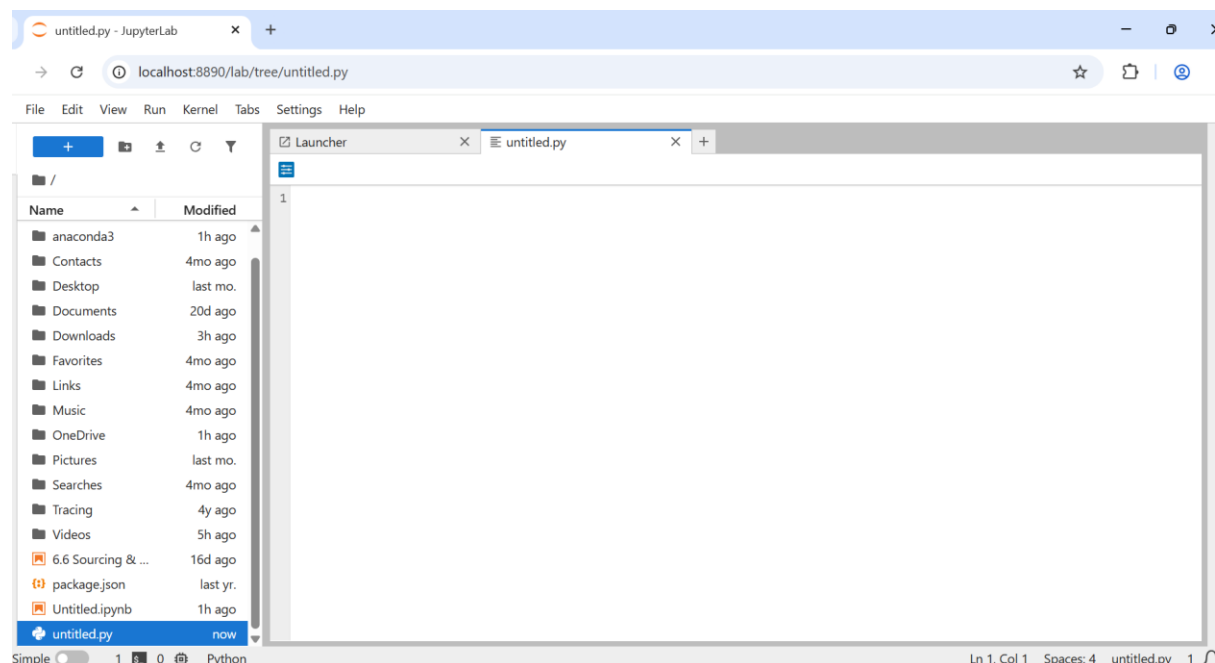
I could identify key connections between countries, events, and political movements that shaped global history.

The most challenging part of this task would likely be cleaning and organizing unstructured text data and ensuring that the extracted relationships are accurate and meaningful. Working as a freelancer on this project would require strong self-management skills, including scheduling, version control, and progress tracking. In the end, this project would be a valuable opportunity to connect data science with history, transforming complex information into visual stories that help us better understand the past and its impact on today's world.
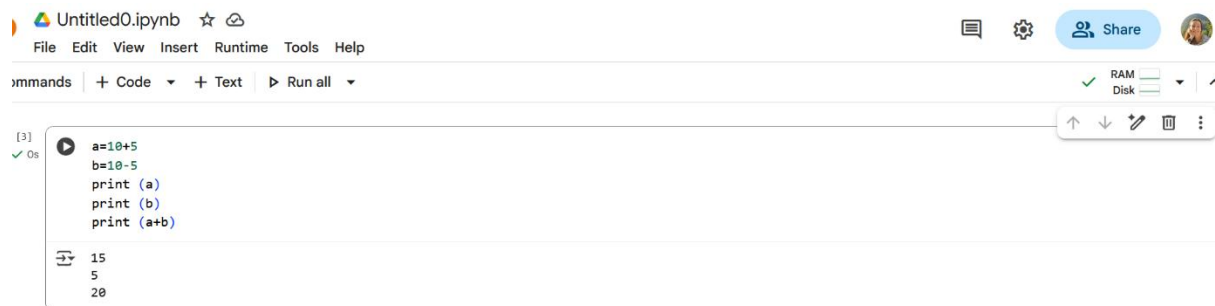
3. Install JupyterLab and explore its functionalities. Reflect on what you notice in terms of differences (good or bad) compared to Jupyter Notebook. Take a screenshot of a JupyterLab session and paste it into your text document.

Although JupyterLab and Jupyter Notebook have the same structure, JupyterLab offers a much more advanced and flexible interface. Unlike Jupyter Notebook, you can preview CSV and PDF files, start a terminal session, open text editors, and add Markdown—all within one application. This makes it much easier to manage different tasks in a single workspace. Another big advantage is that JupyterLab supports multiple programming languages, such as Python, R, Julia, C++, and even SQL. This flexibility can be very helpful in more complex data projects.

There's also a difference in how convenient it is to use JupyterLab for Python coding. Regardless of the file types you're working with, everything can be done smoothly in one program. Although Jupyter Notebook has a simpler and more beginner-friendly layout, JupyterLab provides better multitasking, file organization, and productivity tools. Overall, JupyterLab feels like a more modern and powerful environment for data science and analytics work.

4. Get started with Google Colab and try executing some basic code

5. Write 150 to 200 words about the pros and cons of using JupyterLab and Google Colab in your text document. Create a scenario where you think JupyterLab would be the best tool to use, then do the same for Google Colab.

JupyterLab and Google Colab are both tools for data analysis, machine learning, and Python programming.

JupyterLab offers a flexible, local environment where users can fully customize their workspace, install any library, and work offline. It's excellent for projects that require large datasets stored locally or when working in a secure environment. However, it requires manual setup and sufficient system resources, which can be a disadvantage for beginners or those with limited hardware.

Google Colab, on the other hand, runs entirely in the cloud, offering free access to GPUs and pre-installed libraries. It's ideal for collaboration, as notebooks can easily be shared via Google Drive. The main disadvantages are dependency on internet access, limited storage, and potential session timeouts.

Scenario 1: JupyterLab is best for a data analyst working on confidential data stored locally, where data privacy and offline access are essential.

Scenario 2: Google Colab is ideal for collaborating with colleagues on a shared project, since everyone can access the same notebook, run code effortlessly, make edits, leave comments, and share results instantly online.