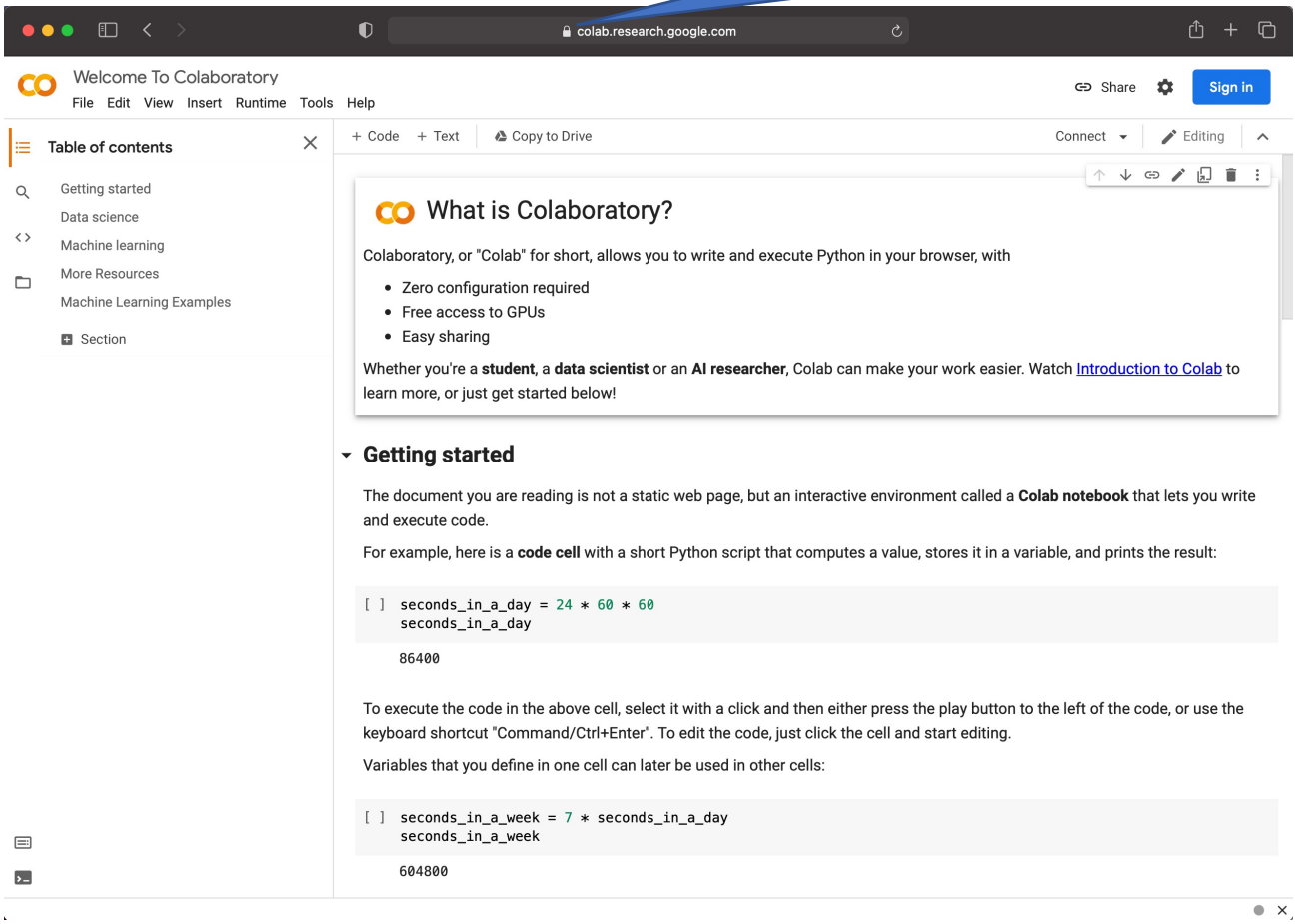
How to use Google Colab

Masaaki Nagahara

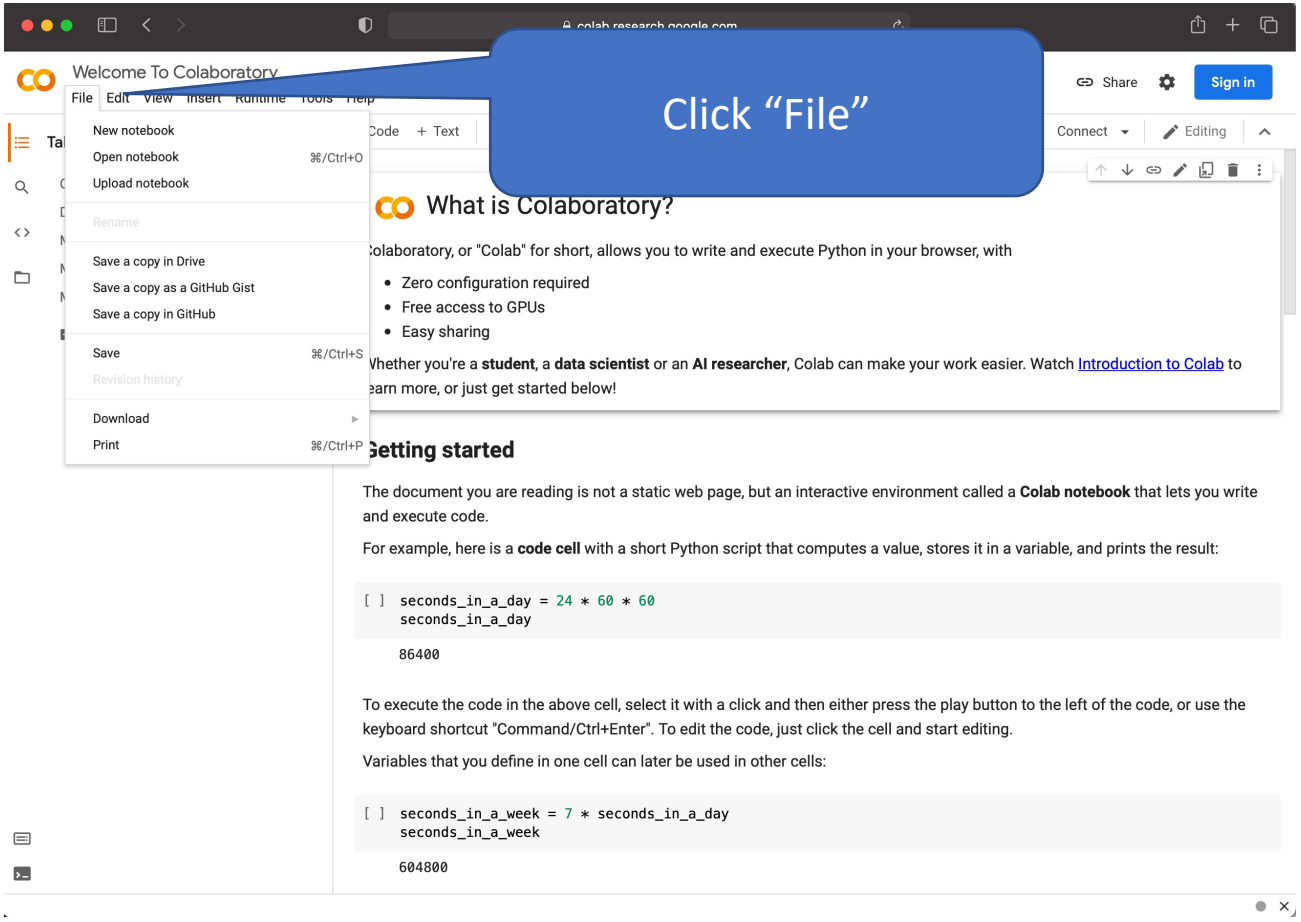
The University of Kitakyushu

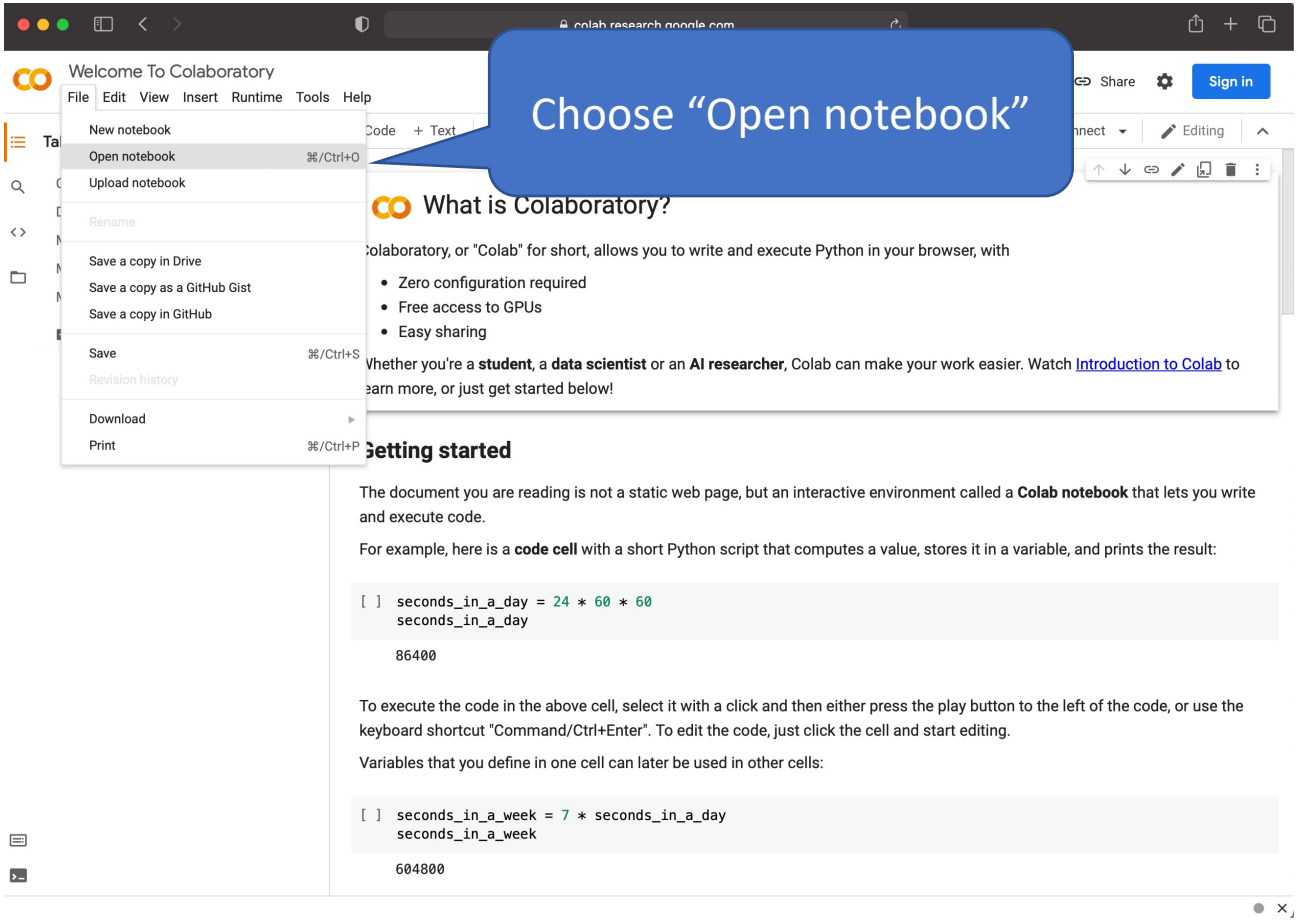
Open your web browser

https://colab.research.google.com

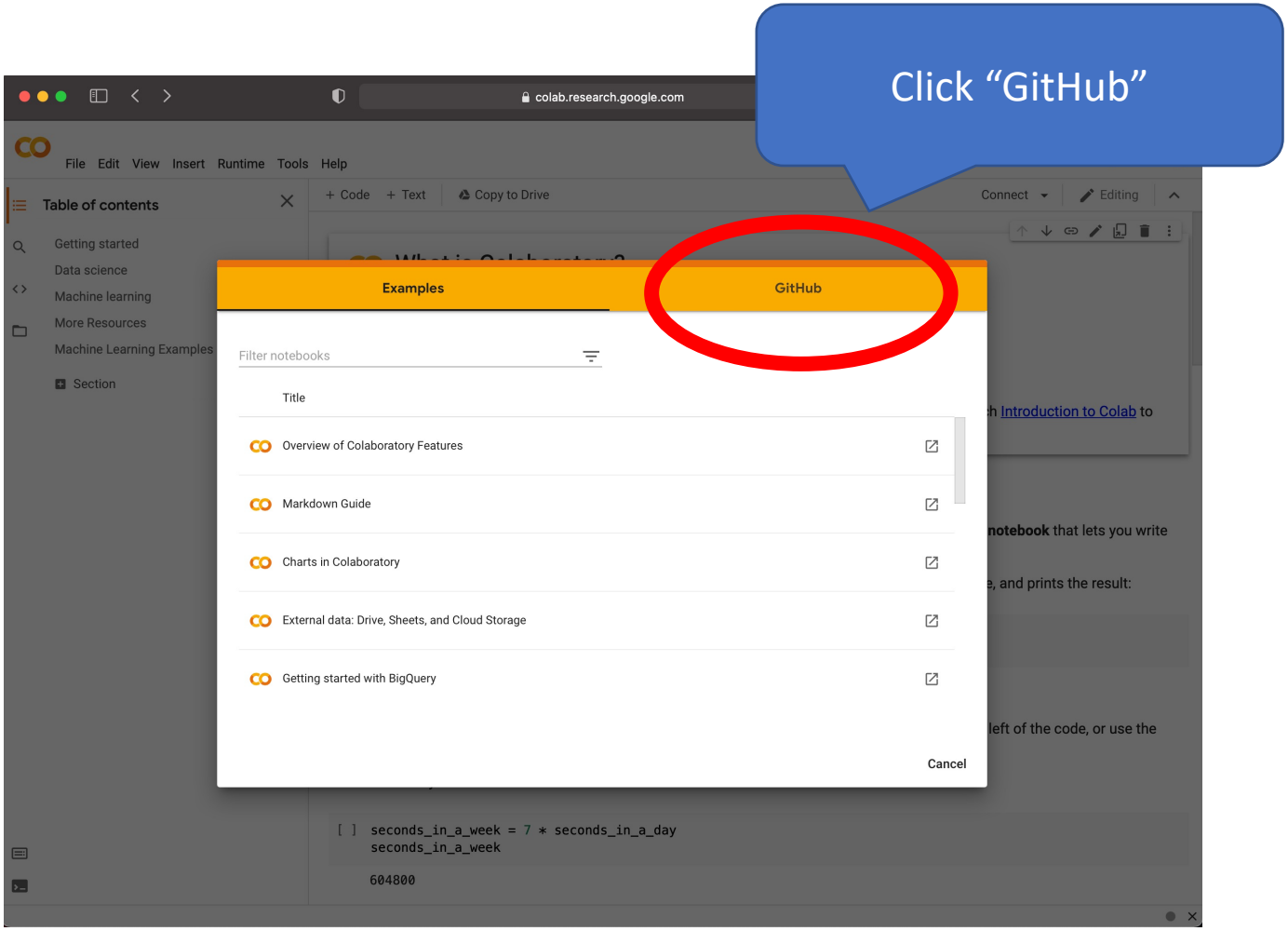


.ipynb file (Jupyter Notebook) for Lecture 2



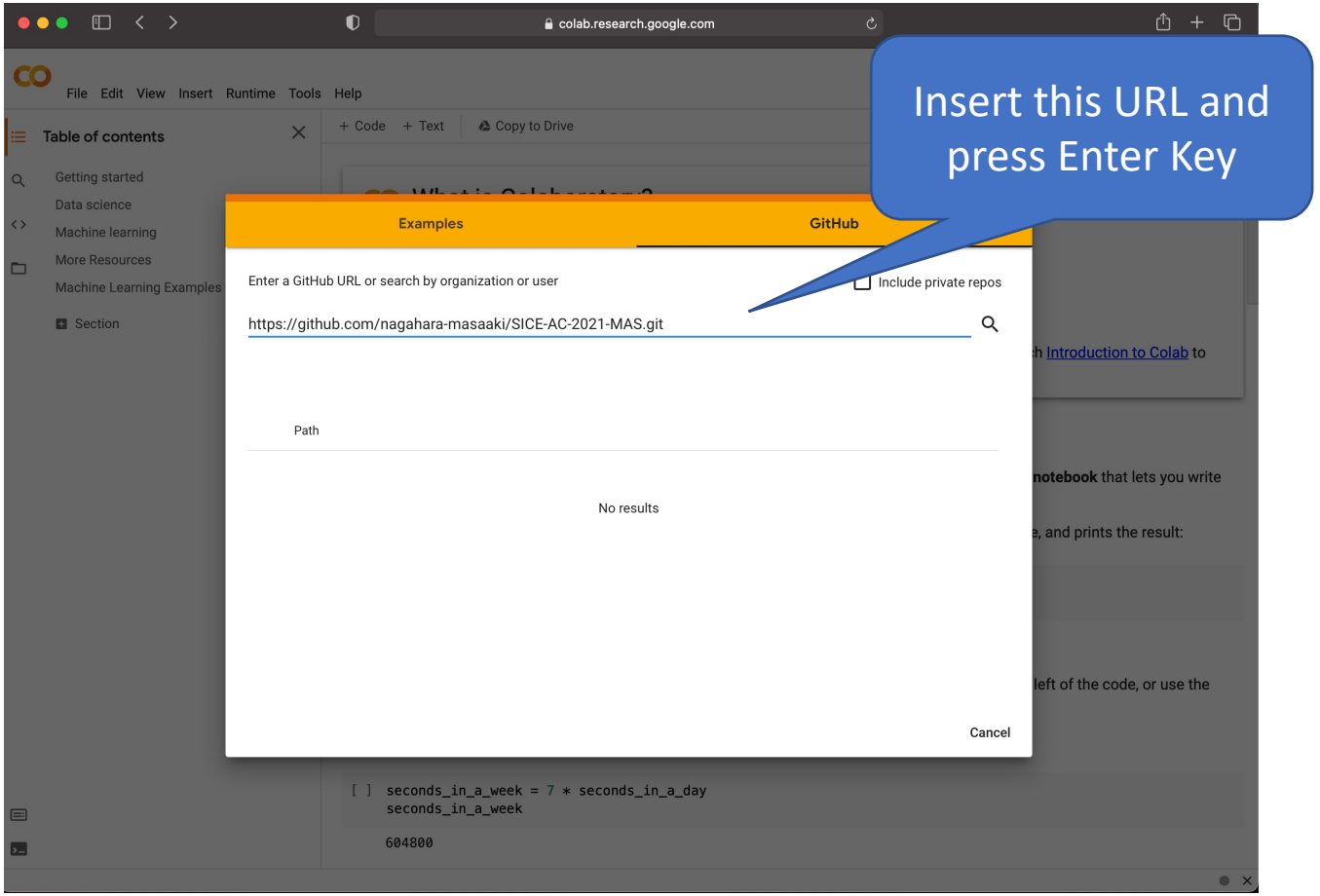

Choose “Open notebook”
Welcome To Colaboratory
File Edit View Insert Runtime Tools Help
New notebook
Open notebook ⌘/Ctrl+O
Upload notebook
Rename
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save ⌘/Ctrl+S
Revision history
Download
Print ⌘/Ctrl+P
Share
Sign in
Editing
What is Colaboratory?
Zero configuration required
Free access to GPUs
Easy sharing
Introduction to Colab
Getting started
The document you are reading is not a static web page, but an interactive environment called a Colab notebook that lets you write and execute code.
For example, here is a code cell with a short Python script that computes a value, stores it in a variable, and prints the result:
seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
86400
To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.
Variables that you define in one cell can later be used in other cells:
seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
604800

Open GitHub file

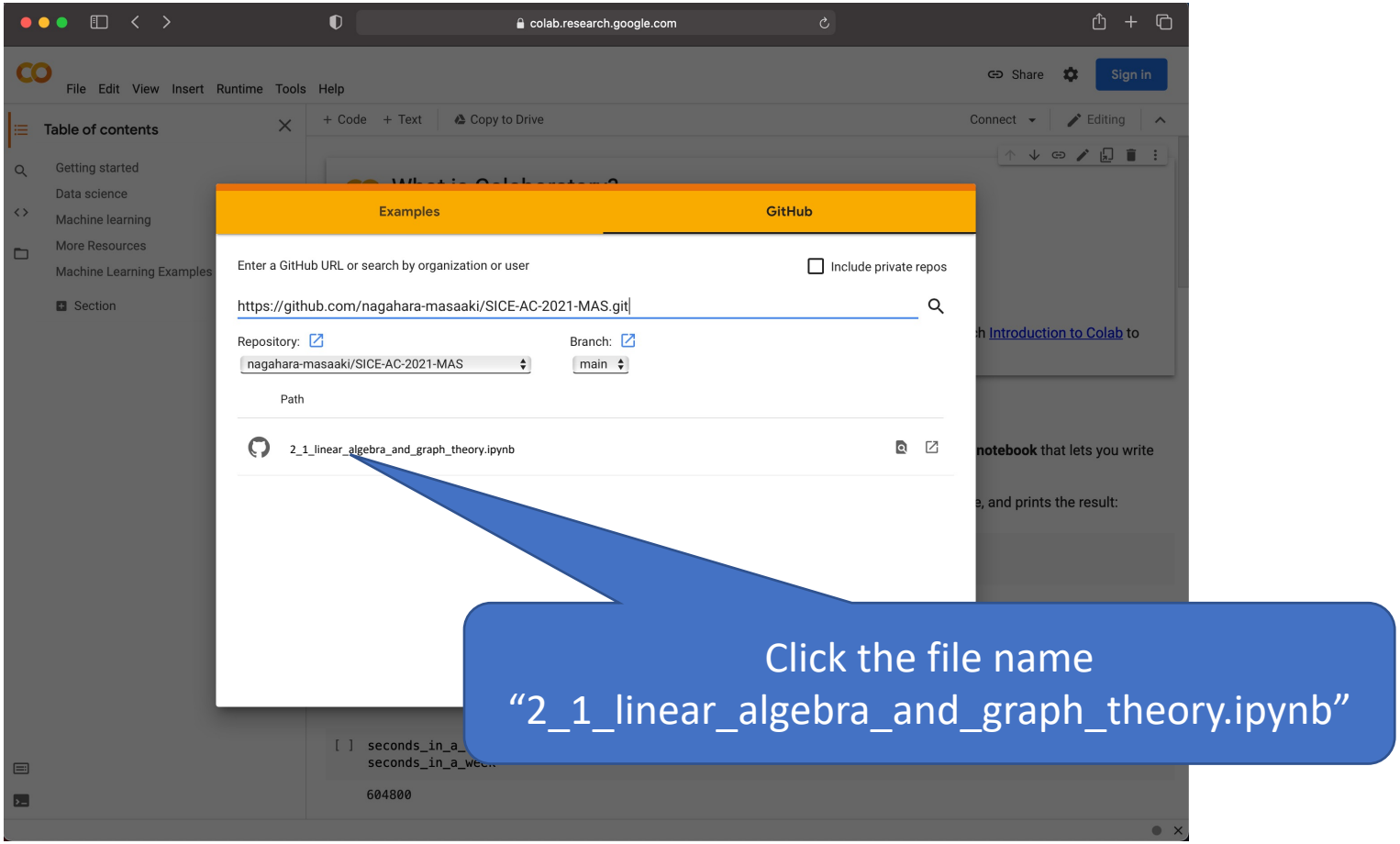


Enter GitHub URL

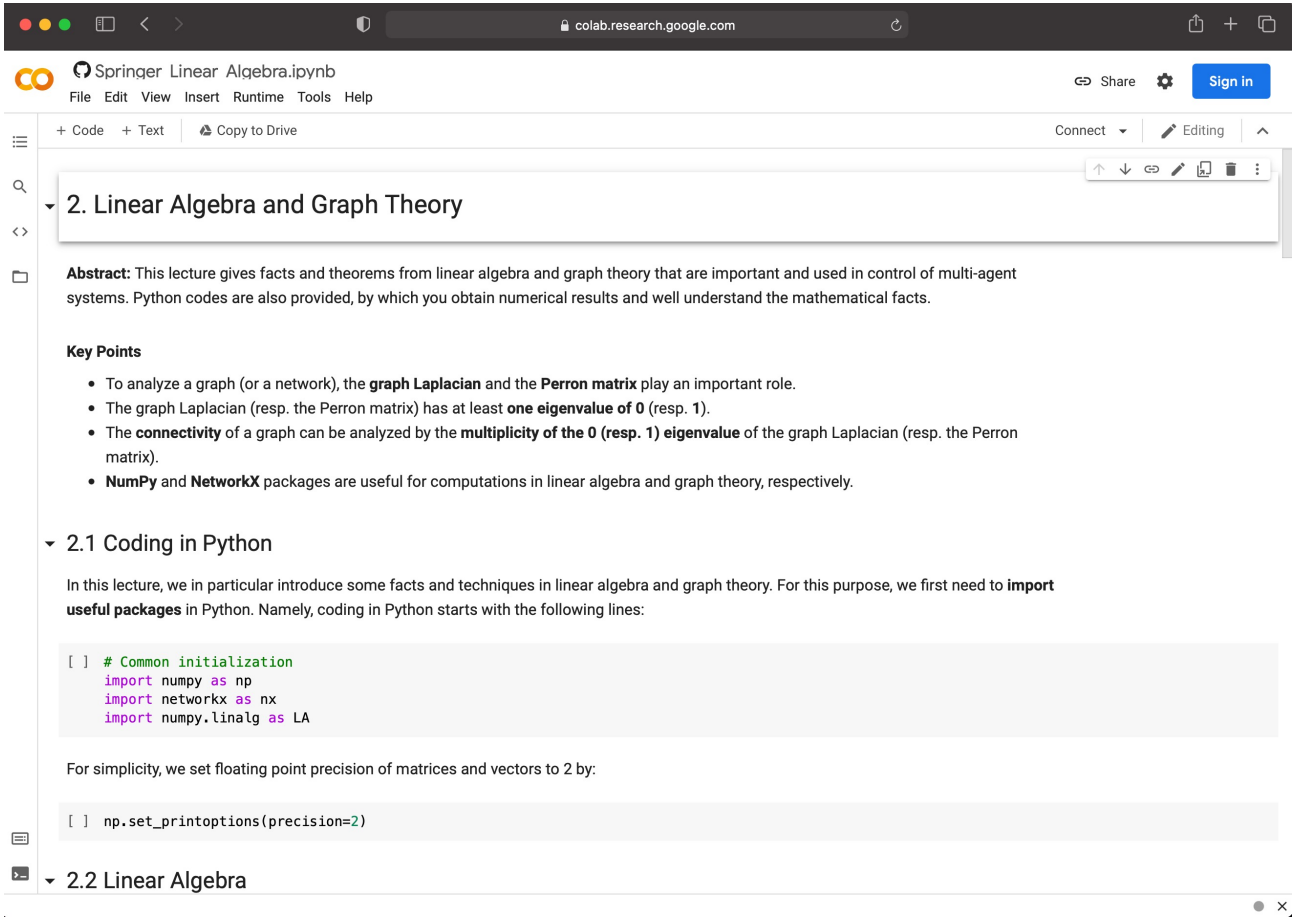
https://github.com/nagahara-masaaki/SICE-AC-2021-MAS.git



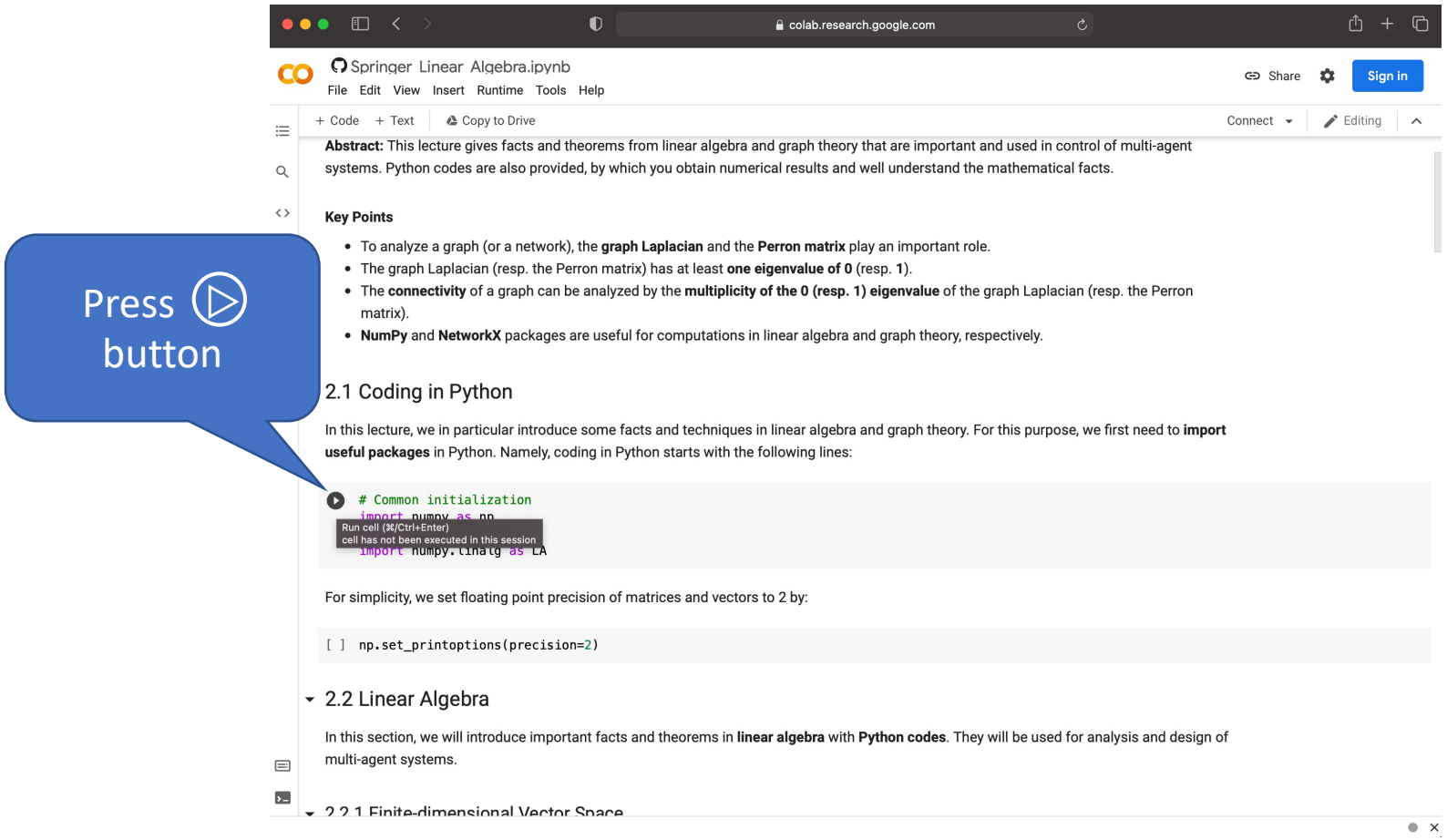
Choose the .ipynb file



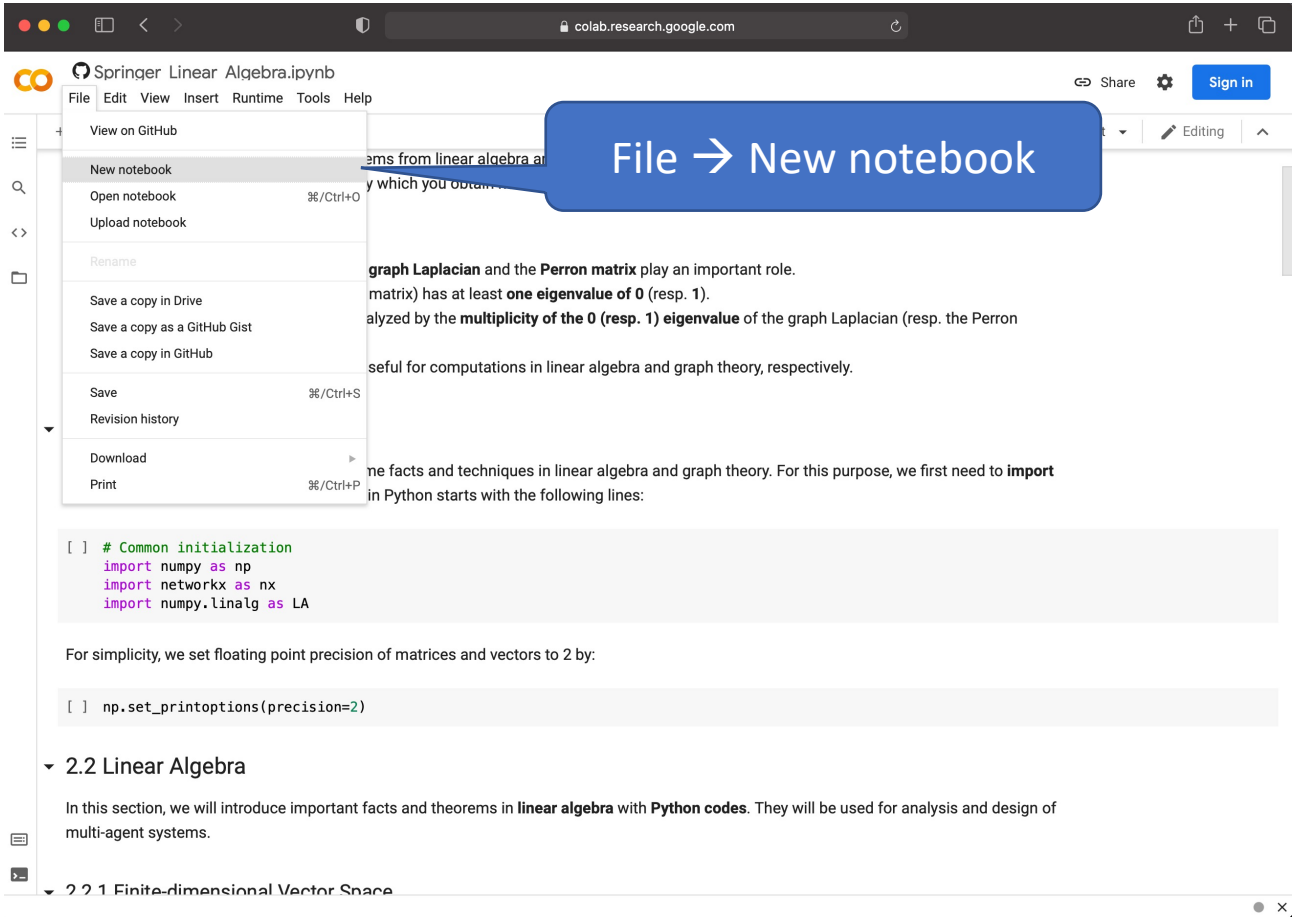
You will see


colab.research.google.com
Springer Linear Algebra.ipynb
File Edit View Insert Runtime Tools Help
Share
Sign in
+ Code + Text
Copy to Drive
Connect
Editing
2. Linear Algebra and Graph Theory
Abstract: This lecture gives facts and theorems from linear algebra and graph theory that are important and used in control of multi-agent systems. Python codes are also provided, by which you obtain numerical results and well understand the mathematical facts.
Key Points
To analyze a graph (or a network), the graph Laplacian and the Perron matrix play an important role.
The graph Laplacian (resp. the Perron matrix) has at least one eigenvalue of 0 (resp. 1).
The connectivity of a graph can be analyzed by the multiplicity of the 0 (resp. 1) eigenvalue of the graph Laplacian (resp. the Perron matrix).
NumPy and NetworkX packages are useful for computations in linear algebra and graph theory, respectively.
2.1 Coding in Python
In this lecture, we in particular introduce some facts and techniques in linear algebra and graph theory. For this purpose, we first need to import useful packages in Python. Namely, coding in Python starts with the following lines:
[ ] # Common initialization
import numpy as np
import networkx as nx
import numpy.linalg as LA
For simplicity, we set floating point precision of matrices and vectors to 2 by:
[ ] np.set_printoptions(precision=2)
2.2 Linear Algebra

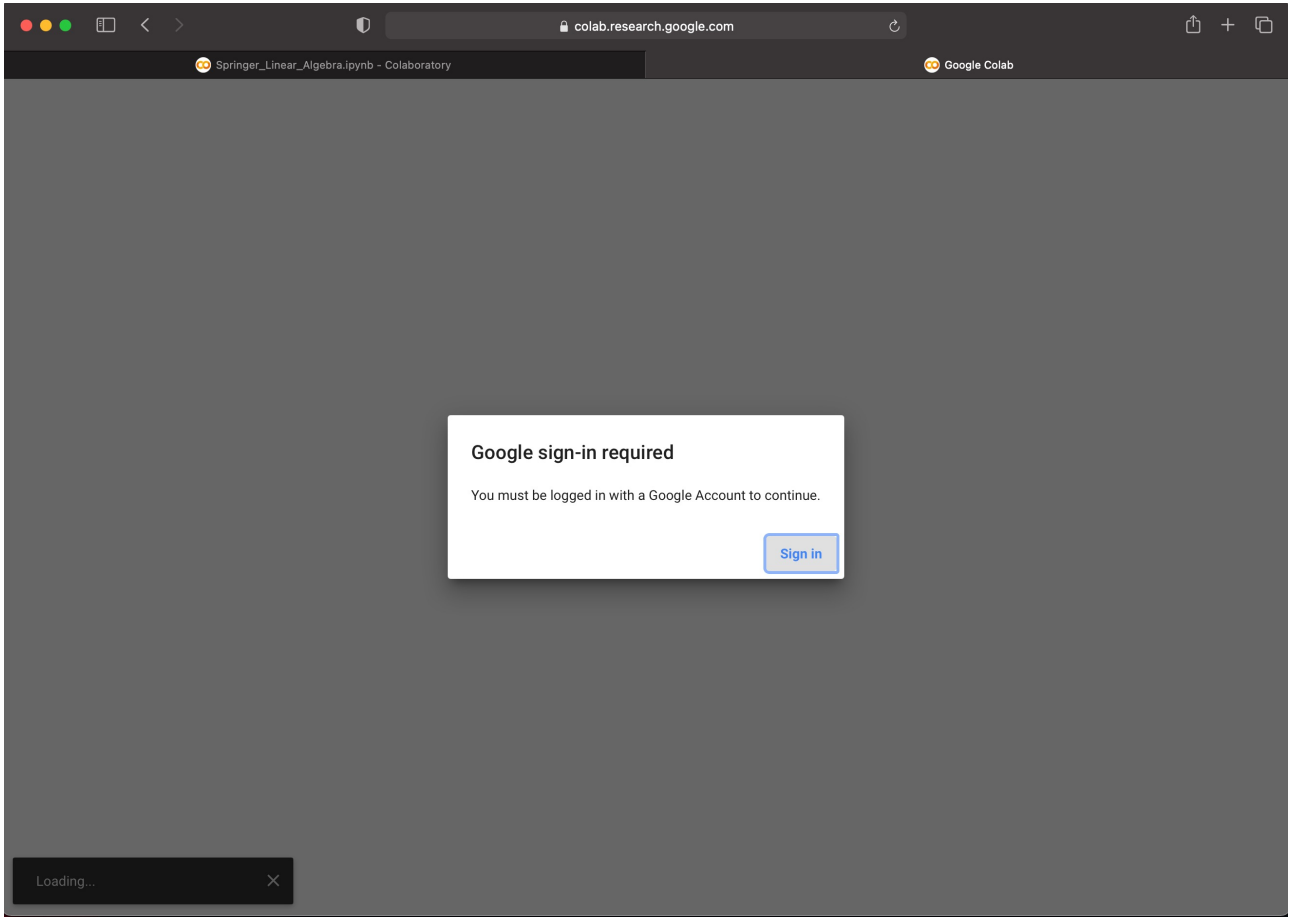
You can run Python code by



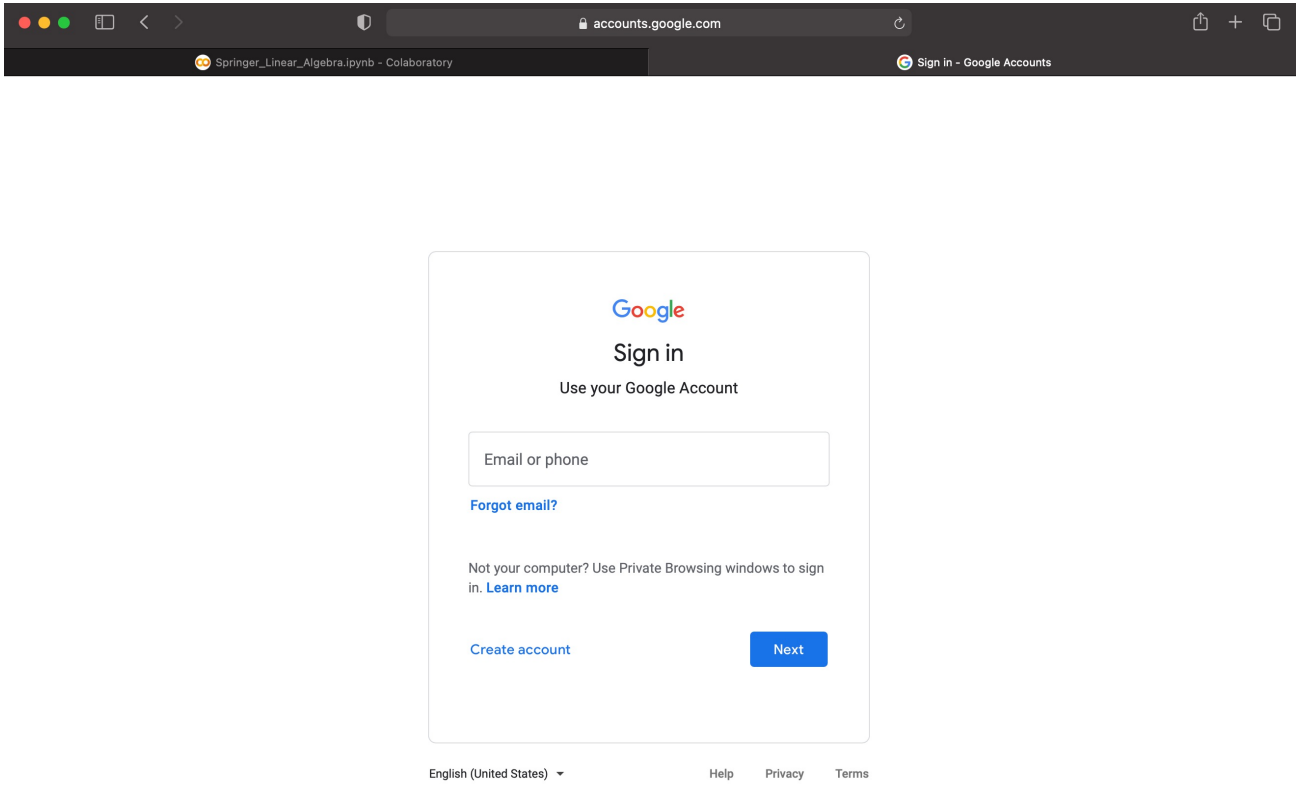
Open .py file (Lectures 3 and 4)



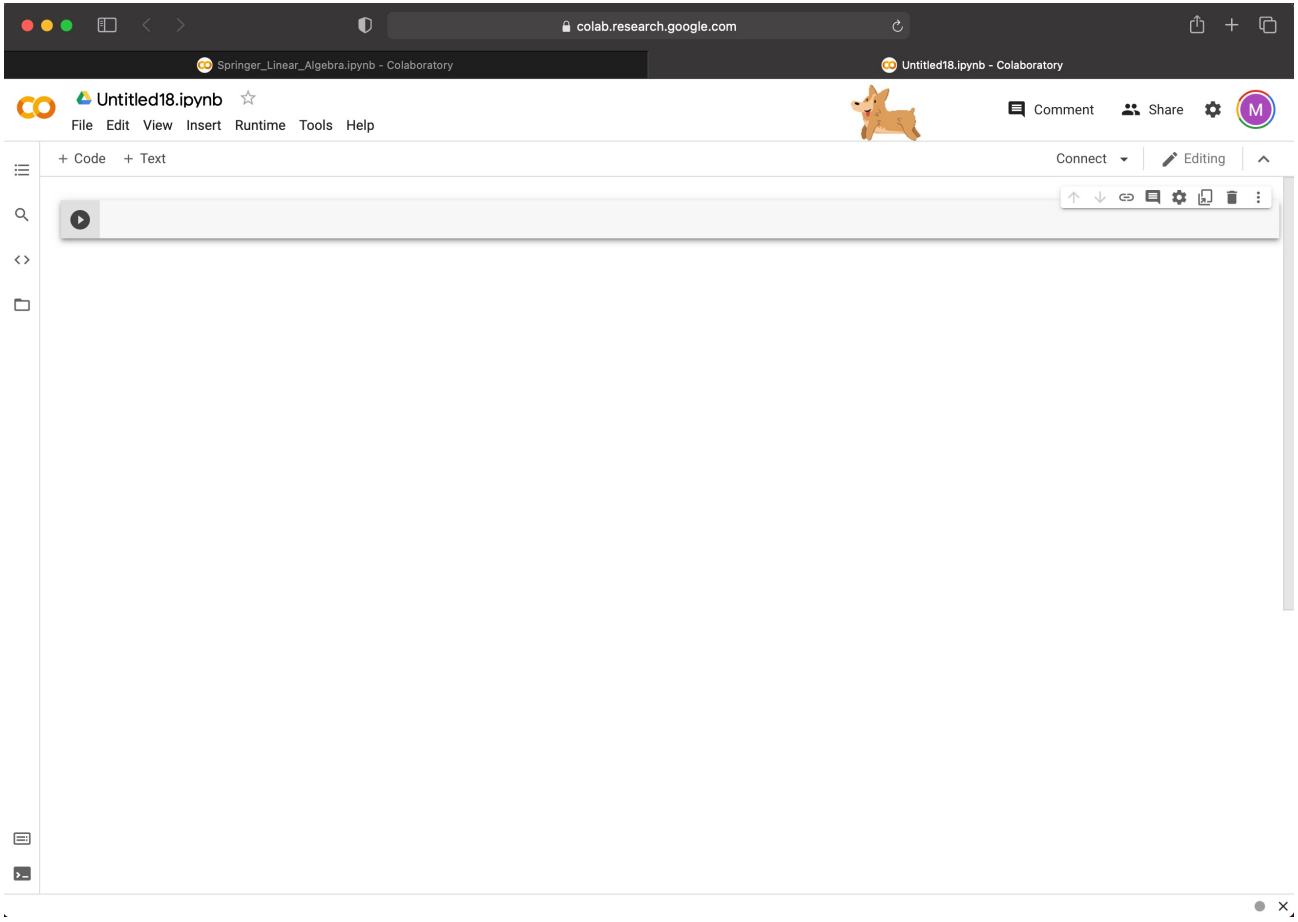
Google sign-in will be required if you haven’t logged in yet.



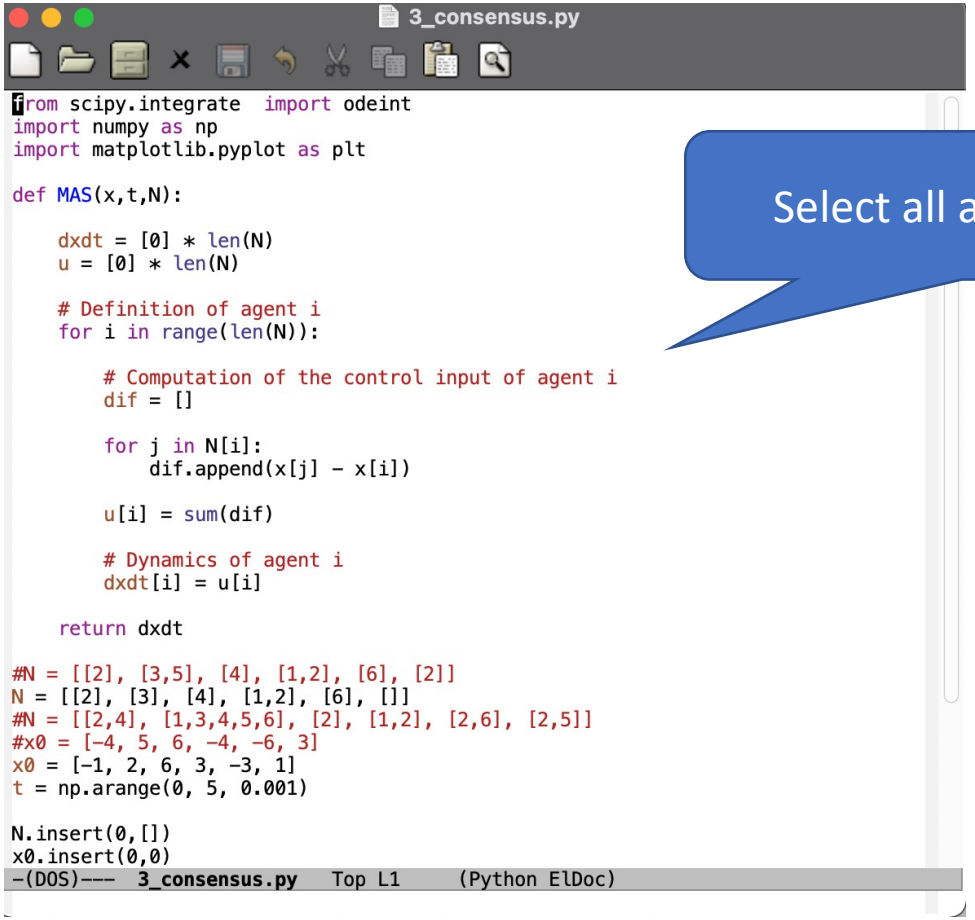
Sign-in with your Google account



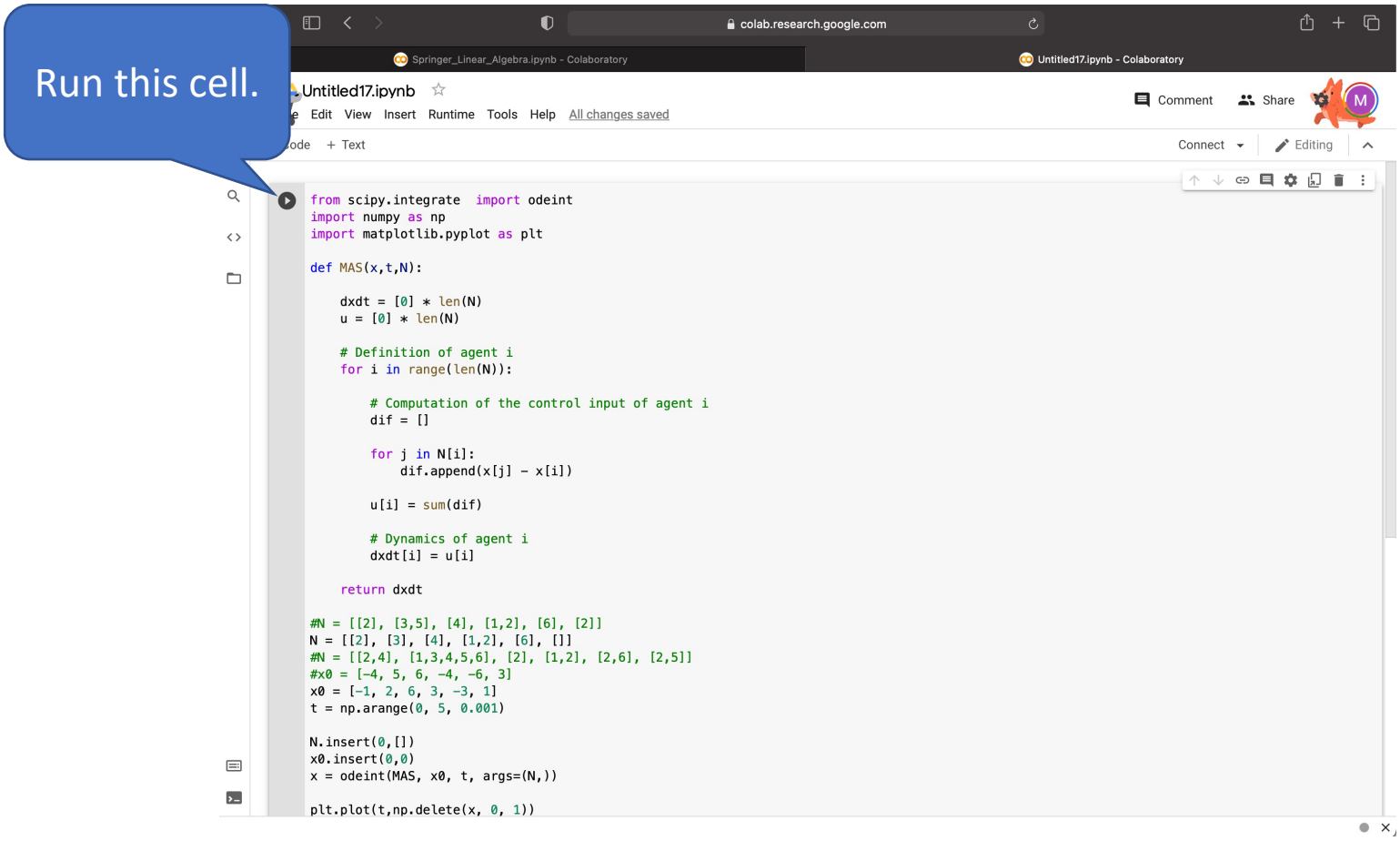
Then a new notebook is opened



Open “3_consensus.py” with your text editor



Past them into the blank cell



You will get the result!