



August 16, 2023 Think-a-Thon

# SciAARe

The word "SciAARe" is written in a large, white, bold, sans-serif font. The letters "i" and "A" are stylized with a purple arrow pointing left and a purple arrow pointing right, respectively. Behind the letters "i" and "A" is a graphic of two orange and yellow clouds.

An Introduction to Python for Data Science

Part 2

Deborah Duran, PhD and Luca Calzoni, MD MS PhD Cand. | NIMHD

# We have registered you for ScHARe

To opt out,  
email us at  
[schare@mail.nih.gov](mailto:schare@mail.nih.gov)

## You have been:

- registered for **ScHARe**
  - added to a **free temporary billing project** that will allow you to run the event materials with your instructors
- You will be active on this billing project for the duration of the Think-a-Thon. If you want to access work-in-progress after this time, you will need to set up your own billing and copy your workspaces to it

# In preparation for the Think-a-Thon

Let's make sure that everyone:

1. has provided their Gmail address and has been registered for ScHARe
2. has created a Terra account
3. can access the tutorial we will be using today at: [bit.ly/schare-python](https://bit.ly/schare-python)
4. has configured their cloud environment
5. can run the tutorial in playground mode:





# SCIARe

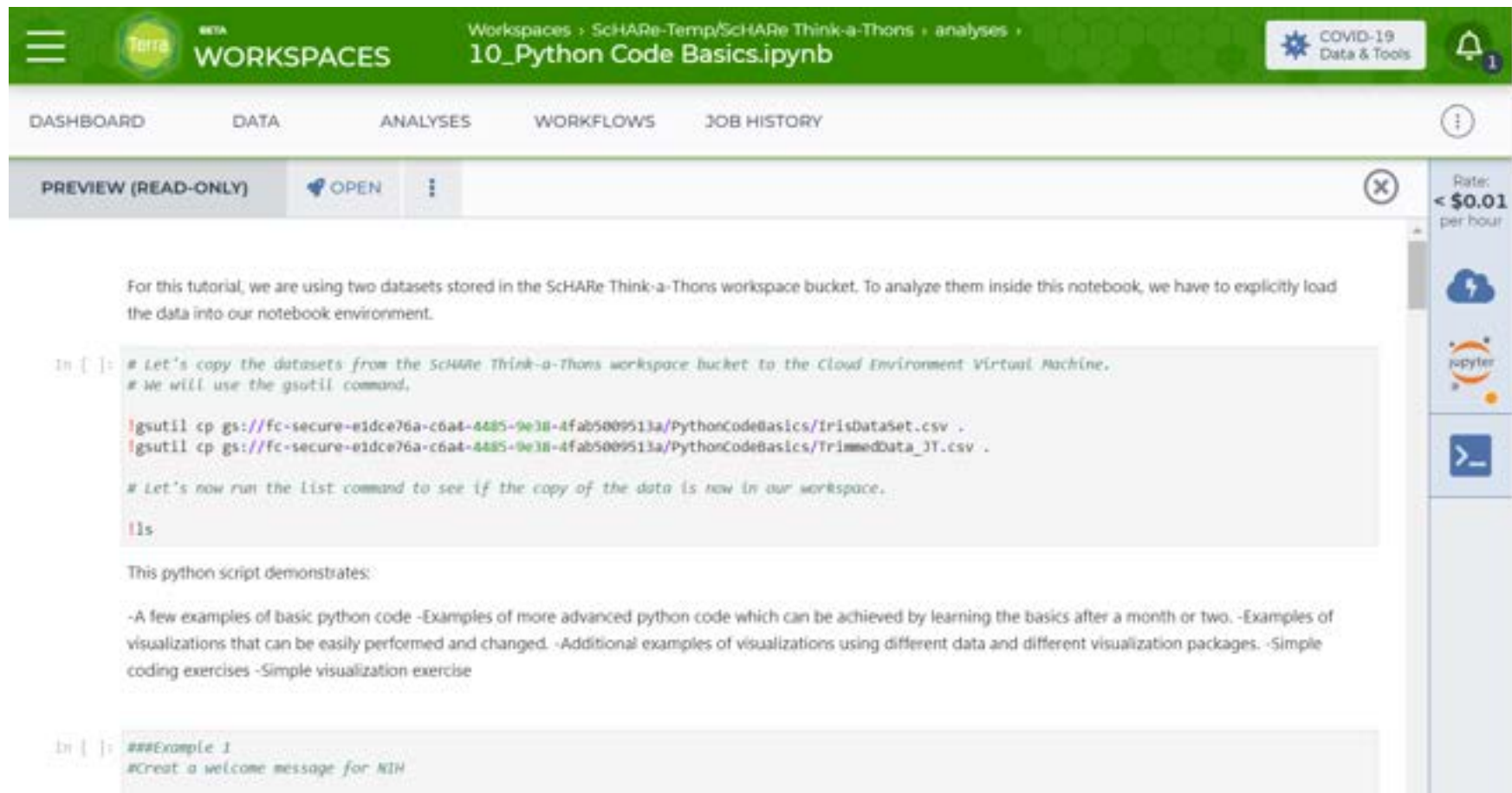
Workshop setup

**Please paste the address below in your browser:**

**[bit.ly/schare-python](https://bit.ly/schare-python)**

If you have already created a Terra account and are logged in, you will see this:

[bit.ly/schare-python](https://bit.ly/schare-python)



The screenshot displays the Terra Workspaces interface. At the top, there is a green header with the Terra logo, the text 'WORKSPACES', and the workspace path 'Workspaces > SchARE-Temp/SchARE Think-a-Thons > analyses > 10\_Python Code Basics.ipynb'. On the right side of the header, there are icons for 'COVID-19 Data & Tools' and a notification bell. Below the header is a navigation bar with tabs for 'DASHBOARD', 'DATA', 'ANALYSES', 'WORKFLOWS', and 'JOB HISTORY'. The main content area shows a 'PREVIEW (READ-ONLY)' view of a Jupyter notebook. The notebook content includes a text block explaining the tutorial's purpose, a code cell with shell commands for copying data and listing files, and another text block describing the Python script's content. The right sidebar contains a 'Rate: < \$0.01 per hour' indicator, a 'jupyter' logo, and a terminal icon.

For this tutorial, we are using two datasets stored in the SchARE Think-a-Thons workspace bucket. To analyze them inside this notebook, we have to explicitly load the data into our notebook environment.

```
In [ ]: # Let's copy the datasets from the SchARE Think-a-Thons workspace bucket to the Cloud Environment Virtual Machine.
# We will use the gsutil command,

!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/IrisDataSet.csv .
!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/TrimmedData_3T.csv .

# Let's now run the list command to see if the copy of the data is now in our workspace.

!ls
```

This python script demonstrates:

- A few examples of basic python code
- Examples of more advanced python code which can be achieved by learning the basics after a month or two.
- Examples of visualizations that can be easily performed and changed.
- Additional examples of visualizations using different data and different visualization packages.
- Simple coding exercises
- Simple visualization exercise

```
In [ ]: ##Example 1
#Creat a welcome message for NIH
```

If you have not logged in, or have not yet created a Terra account, you will see this:

[bit.ly/schare-python](https://bit.ly/schare-python)



Click on the login button:

[bit.ly/schare-python](https://bit.ly/schare-python)



The image shows the homepage of the Terra Community Workbench. At the top left, there is a green header with the Terra logo and the word "BETA". At the top right, there is a notification bell icon with a "1" next to it. The main heading is "Welcome to Terra Community Workbench". Below this, there is a paragraph: "Terra is a cloud-native platform for biomedical researchers to access data, run analysis tools, and collaborate. [Learn more about Terra.](#)" followed by "If you are a new user or returning user, click log in to continue." At the bottom left, there is a blue "LOG IN" button, which is circled in blue. The background features a grid of hexagons, some containing images of a cell, a test tube, and researchers in a lab.



# Use the Gmail address you provided us with to log in:

[terraprodb2c.b2clogin.com/terraprodb2c.onmicrosoft.com/oauth2/v2.0/authorize?response\\_mode=query&s...](https://terraprodb2c.b2clogin.com/terraprodb2c.onmicrosoft.com/oauth2/v2.0/authorize?response_mode=query&s...)

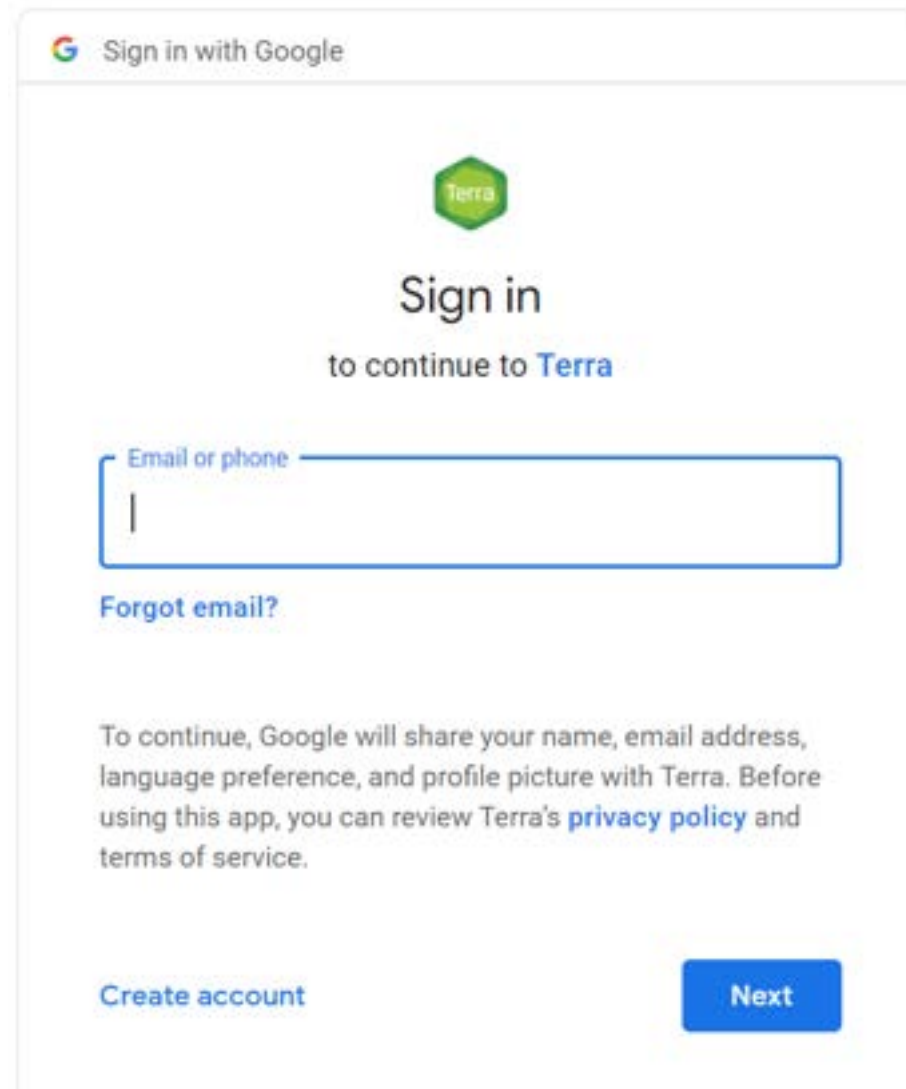


Sign in with Google




Sign in with Microsoft

# Use the Gmail address you provided us with to log in:



Sign in with Google



Sign in  
to continue to [Terra](#)

Email or phone


[Forgot email?](#)

To continue, Google will share your name, email address, language preference, and profile picture with Terra. Before using this app, you can review Terra's [privacy policy](#) and terms of service.


[Create account](#) [Next](#)

# Input the password associated with your Gmail account:

Sign in with Google

 Terra

Hi Luca

 healthcare@


Enter your password

Show password

To continue, Google will share your name, email address, language preference, and profile picture with Terra. Before using this app, you can review Terra's [privacy policy](#) and terms of service.

[Forgot password?](#)

If you are new to Terra, create an account now:



**TERRA**

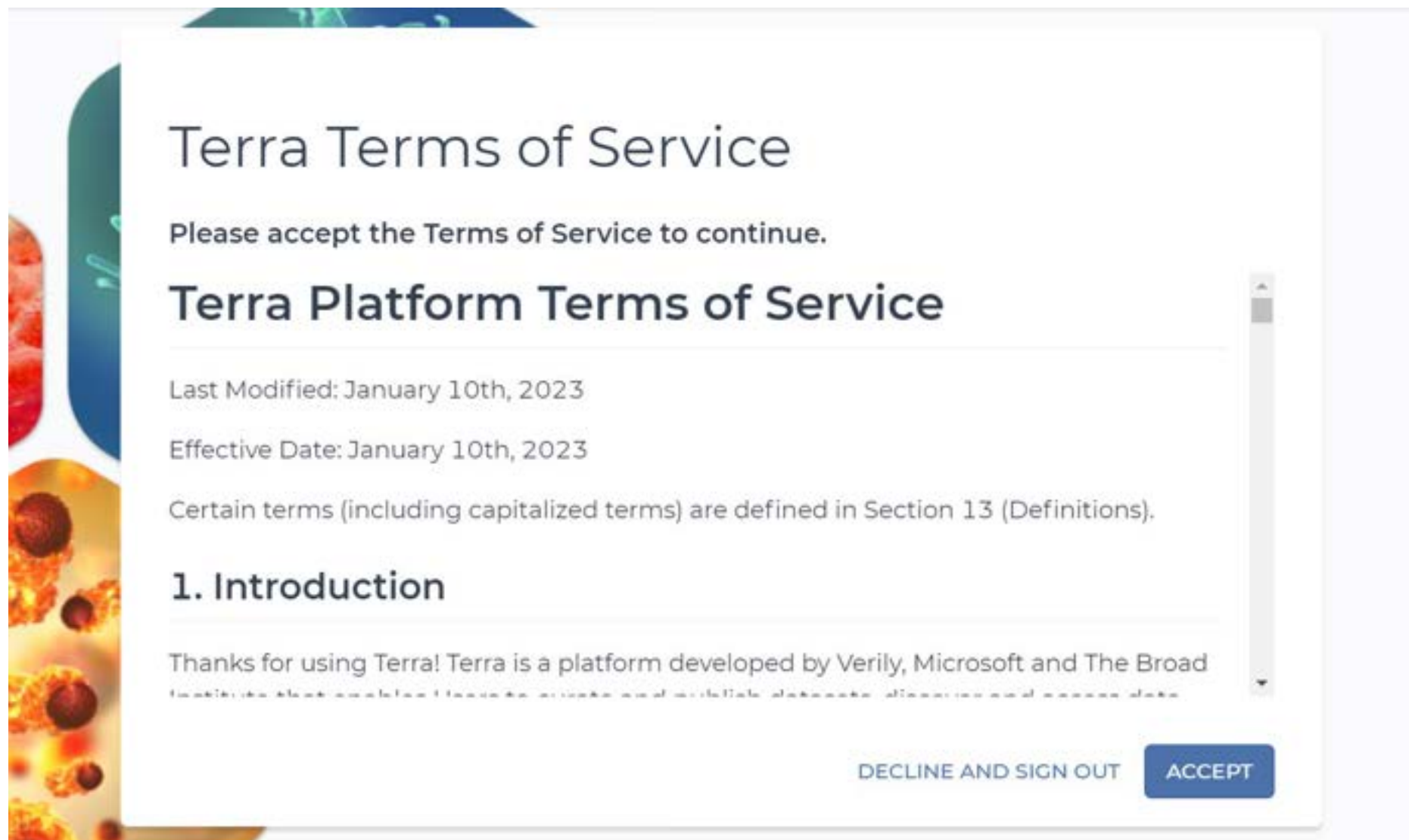
New User Registration

First Name \*

Last Name \*

Contact Email for Notifications \*

# Accept the Terra Terms of Service:



**Terra Terms of Service**

Please accept the Terms of Service to continue.

**Terra Platform Terms of Service**

---

Last Modified: January 10th, 2023

Effective Date: January 10th, 2023

Certain terms (including capitalized terms) are defined in Section 13 (Definitions).

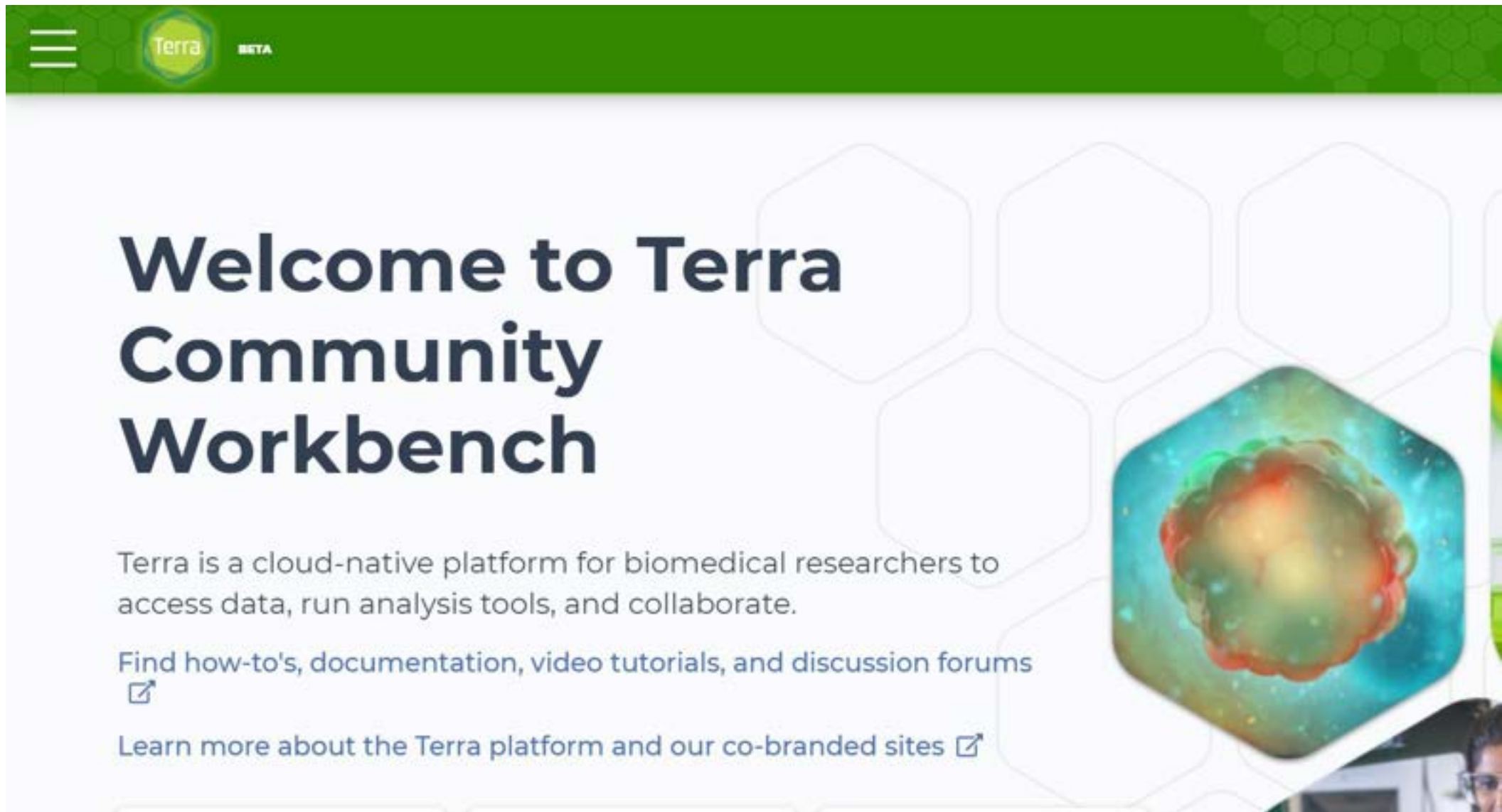
**1. Introduction**



---

Thanks for using Terra! Terra is a platform developed by Verily, Microsoft and The Broad Institute that enables users to create and publish datasets, discover and access data.

DECLINE AND SIGN OUT ACCEPT

You will see this welcome page:



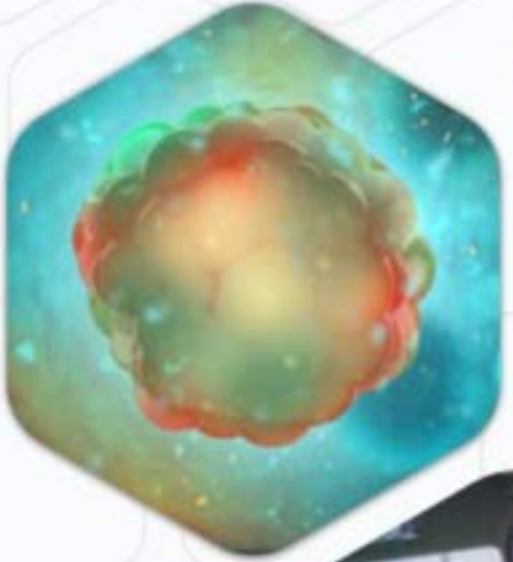
  BETA

# Welcome to Terra Community Workbench

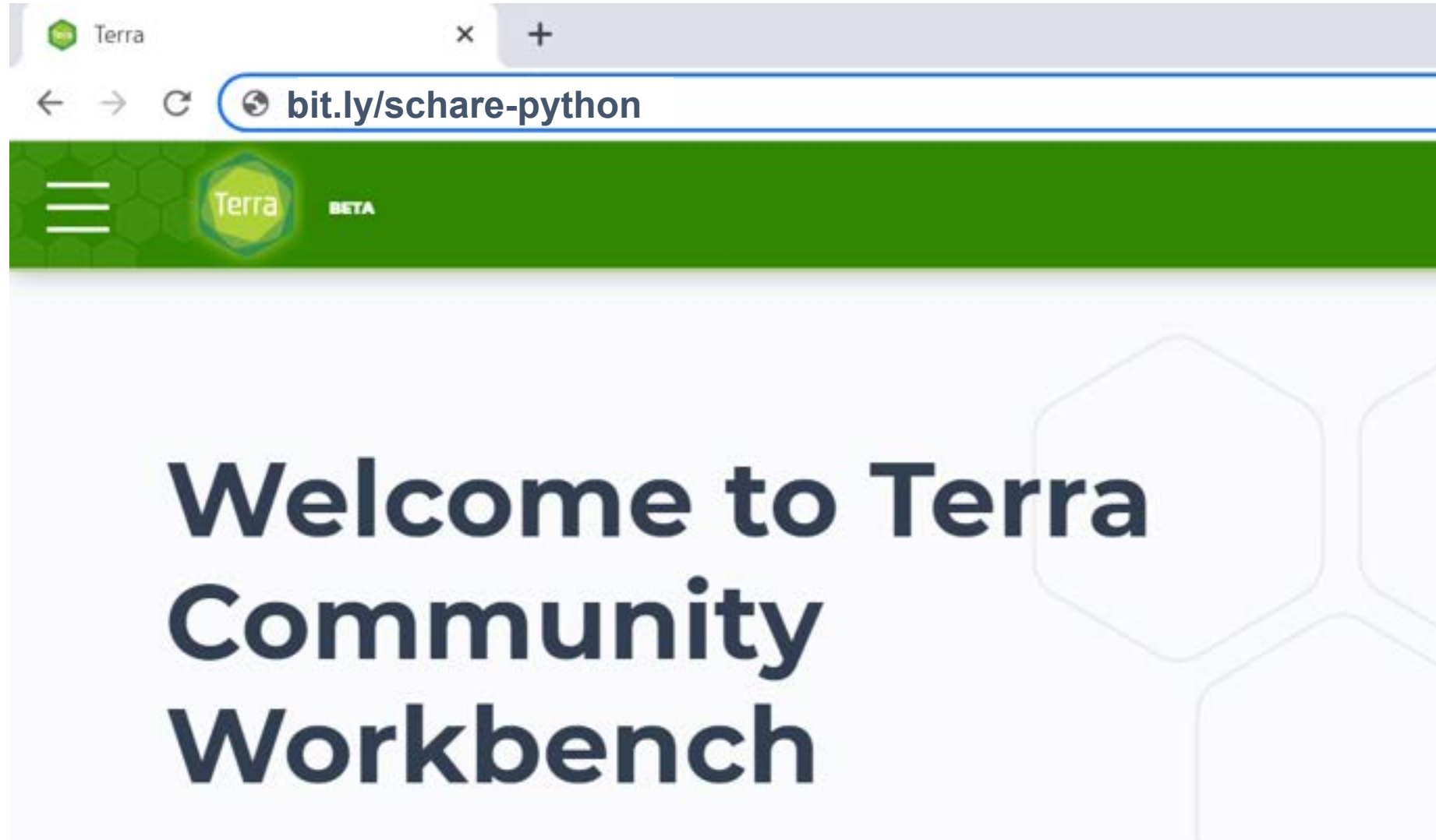
Terra is a cloud-native platform for biomedical researchers to access data, run analysis tools, and collaborate.

Find how-to's, documentation, video tutorials, and discussion forums [↗](#)

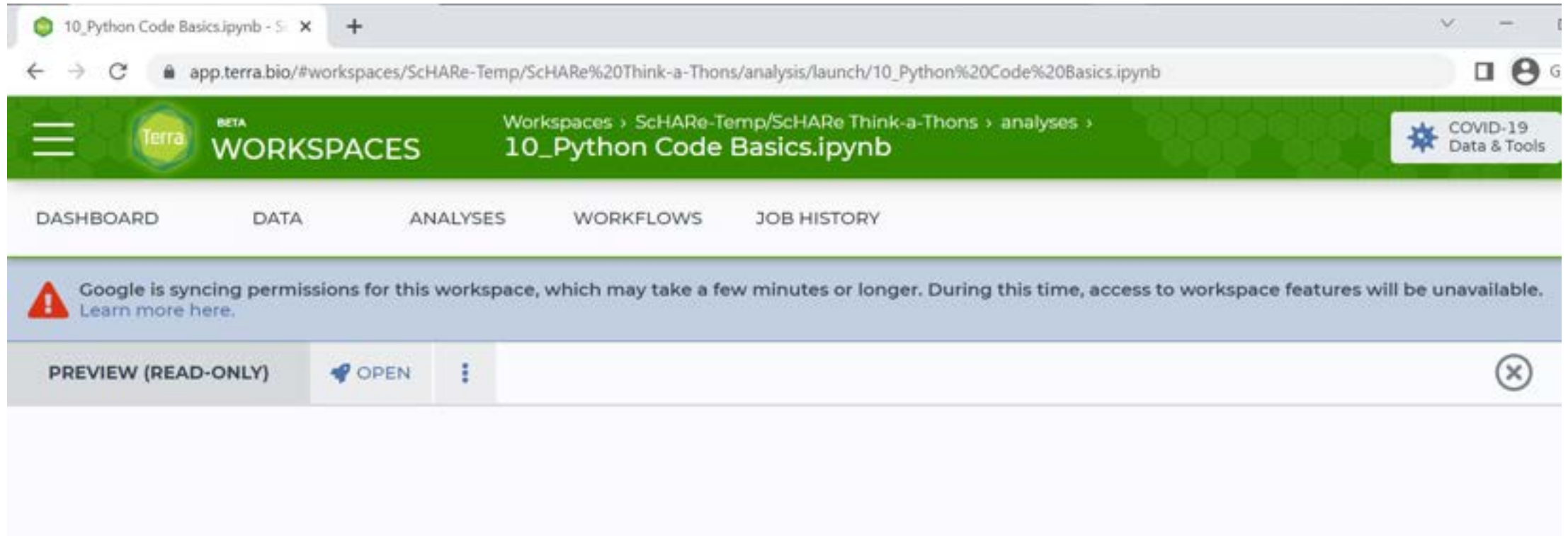
Learn more about the Terra platform and our co-branded sites [↗](#)



Paste this address in your browser: [bit.ly/schare-python](https://bit.ly/schare-python)



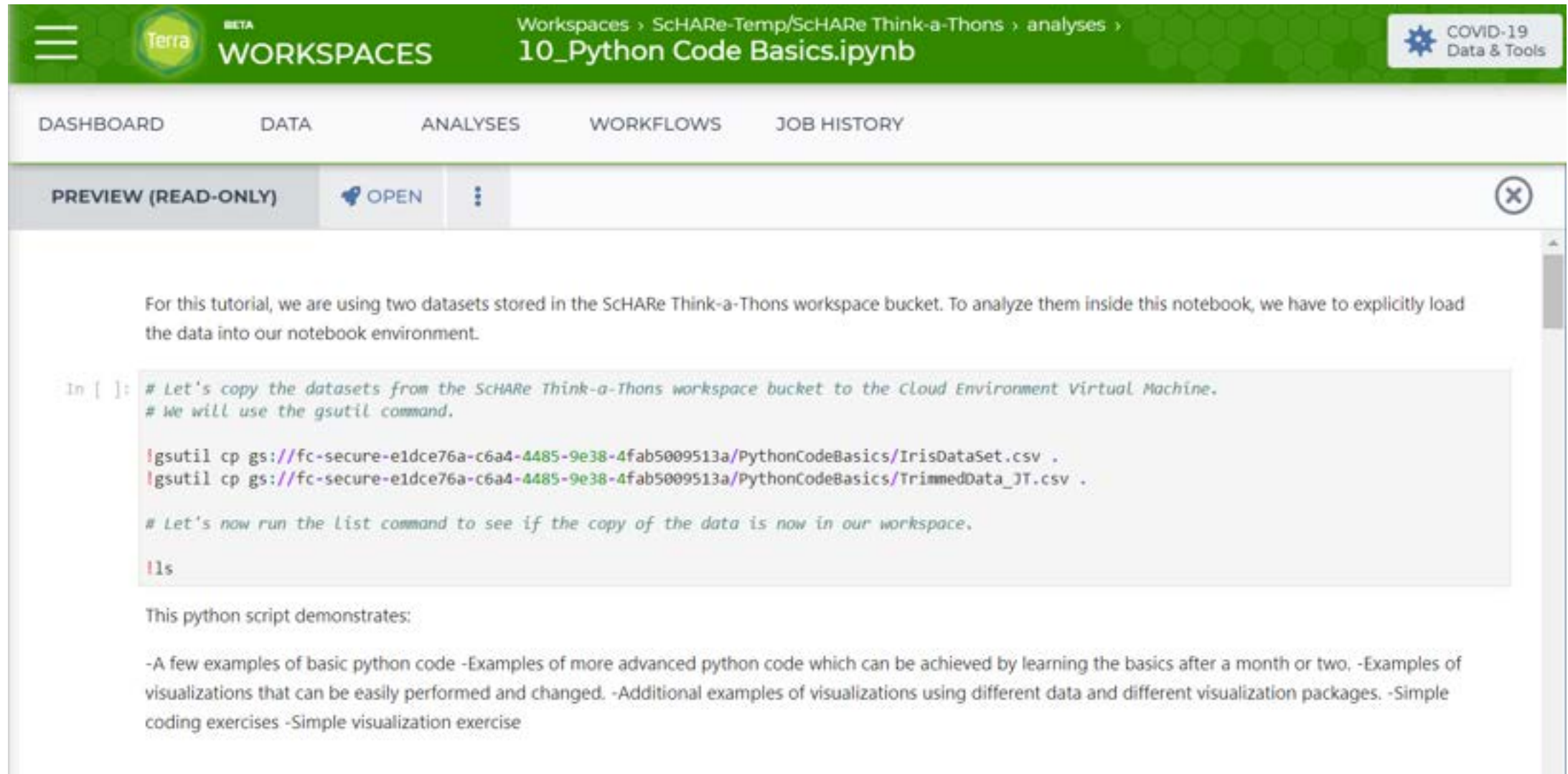
# Newly registered users might see this message:



This is normal: the message should go away in a few minutes



# Refreshing the page after a while, all users should see this:



The screenshot shows the Terra Workspaces interface. The top navigation bar is green and contains the Terra logo, the word "WORKSPACES", and the breadcrumb "Workspaces > ScHARe-Temp/ScHARe Think-a-Thons > analyses > 10\_Python Code Basics.ipynb". A "COVID-19 Data & Tools" button is in the top right. Below the navigation bar is a menu with "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOB HISTORY". The main content area is titled "PREVIEW (READ-ONLY)" and contains a notebook cell with the following text:

For this tutorial, we are using two datasets stored in the ScHARe Think-a-Thons workspace bucket. To analyze them inside this notebook, we have to explicitly load the data into our notebook environment.

```
In [ ]: # Let's copy the datasets from the ScHARe Think-a-Thons workspace bucket to the Cloud Environment Virtual Machine.
# We will use the gsutil command.

!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/IrisDataSet.csv .
!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/TrimmedData_JT.csv .

# Let's now run the list command to see if the copy of the data is now in our workspace.

!ls
```

This python script demonstrates:

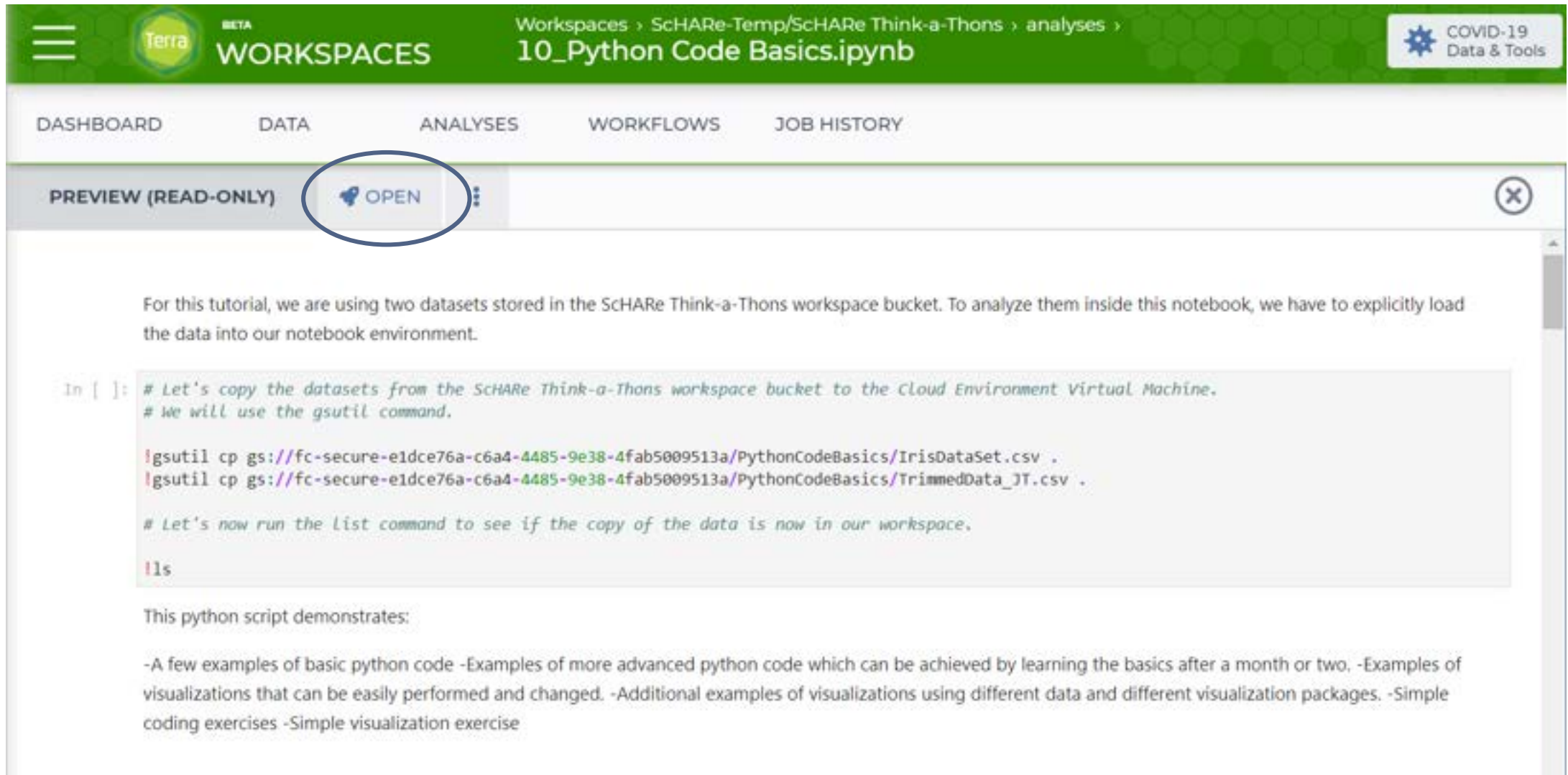
- A few examples of basic python code
- Examples of more advanced python code which can be achieved by learning the basics after a month or two.
- Examples of visualizations that can be easily performed and changed.
- Additional examples of visualizations using different data and different visualization packages.
- Simple coding exercises
- Simple visualization exercise

# Do you see a Playground mode button?



**If yes, click on it to start your virtual computer. You are done!**

# If you don't see Playground mode, click on the Open button:



The screenshot shows the Terra WORKSPACES interface. The top navigation bar is green and contains the Terra logo, the word "WORKSPACES", and the breadcrumb "Workspaces > ScHARe-Temp/ScHARe Think-a-Thons > analyses > 10\_Python Code Basics.ipynb". A "COVID-19 Data & Tools" button is in the top right. Below the navigation bar is a menu with "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOB HISTORY". The "ANALYSES" menu is expanded, showing "PREVIEW (READ-ONLY)" and "OPEN". The "OPEN" button is circled in blue. Below the menu is a notebook preview area with a text block and a code cell.

For this tutorial, we are using two datasets stored in the ScHARe Think-a-Thons workspace bucket. To analyze them inside this notebook, we have to explicitly load the data into our notebook environment.

```
In [ ]: # Let's copy the datasets from the ScHARe Think-a-Thons workspace bucket to the Cloud Environment Virtual Machine.
# We will use the gsutil command.

!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/IrisDataSet.csv .
!gsutil cp gs://fc-secure-e1dce76a-c6a4-4485-9e38-4fab5009513a/PythonCodeBasics/TrimmedData_JT.csv .

# Let's now run the list command to see if the copy of the data is now in our workspace.

!ls
```

This python script demonstrates:

- A few examples of basic python code
- Examples of more advanced python code which can be achieved by learning the basics after a month or two.
- Examples of visualizations that can be easily performed and changed.
- Additional examples of visualizations using different data and different visualization packages.
- Simple coding exercises
- Simple visualization exercise

# Configure your virtual computer – accept the default values:

The screenshot shows the Terra Jupyter Cloud Environment configuration page. The browser address bar displays the URL: `app.terra.bio/#workspaces/SchARE-Temp/SchARE%20Think-a-Thons/analysis/launch/10_Python%20Code%20Basics.ipynb`. The page title is "Jupyter Cloud Environment".

The interface includes a navigation menu with "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOBS". A warning message states: "Google is syncing permissions for this workspace, which may take a few minutes. Learn more here." Below this, there are buttons for "PREVIEW (READ-ONLY)", "OPEN", and a vertical ellipsis menu.

The main configuration area is titled "Jupyter Cloud Environment" and contains the following sections:

- Costs:** A table showing costs for different components:

Running cloud compute cost	Paused cloud compute cost	Persistent disk cost
\$0.05 per hr	\$0.00 per hr	\$2.00 per month
- Application configuration:** A dropdown menu set to "Default: (GATK 4.2.4.0, Python 3.7.12, R 4.3.0)". Below it, text indicates "What's installed on this environment?" and "Updated: Jun 8, 2023 Version: 2.2.14".
- Startup script:** A text input field labeled "Startup script" with the placeholder "Optional - Learn more about startup scripts." and a "URI" input field below it.
- Cloud compute profile:** Two dropdown menus for "CPUs" (set to 1) and "Memory (GB)" (set to 3.75). Below these is a checkbox for "Enable GPUs" (marked as BETA) and a link to "Learn more about GPU cost and restrictions."
- Compute type:** A dropdown menu at the bottom of the configuration area.

# Click on Create below:

10\_Python Code Basics.ipynb - 5 x +

app.terra.bio/#workspaces/SchARE-Temp/SchARE%20Think-a-Thons/analysis/launch/10\_Python%20Code%20Basics.ipynb

WORKSPACES

Workspaces > SchARE-Temp/SchARE%20Think-a-Thons/analysis/launch/10\_Python Code Basics

DASHBOARD DATA ANALYSES WORKFLOWS JOBS

Google is syncing permissions for this workspace, which may take a few minutes. [Learn more here.](#)

PREVIEW (READ-ONLY) OPEN

### Jupyter Cloud Environment

A cloud environment consists of application configuration, cloud compute and persistent disk(s).

Running cloud compute cost	Paused cloud compute cost	Persistent disk cost
\$0.05 per hr	\$0.00 per hr	\$2.00 per month

30 minutes of inactivity

Location BETA ⓘ

us-central1 (Iowa) (default)

**Persistent disk**

Persistent disks store analysis data. [Learn more about persistent disks and where your disk is mounted.](#)

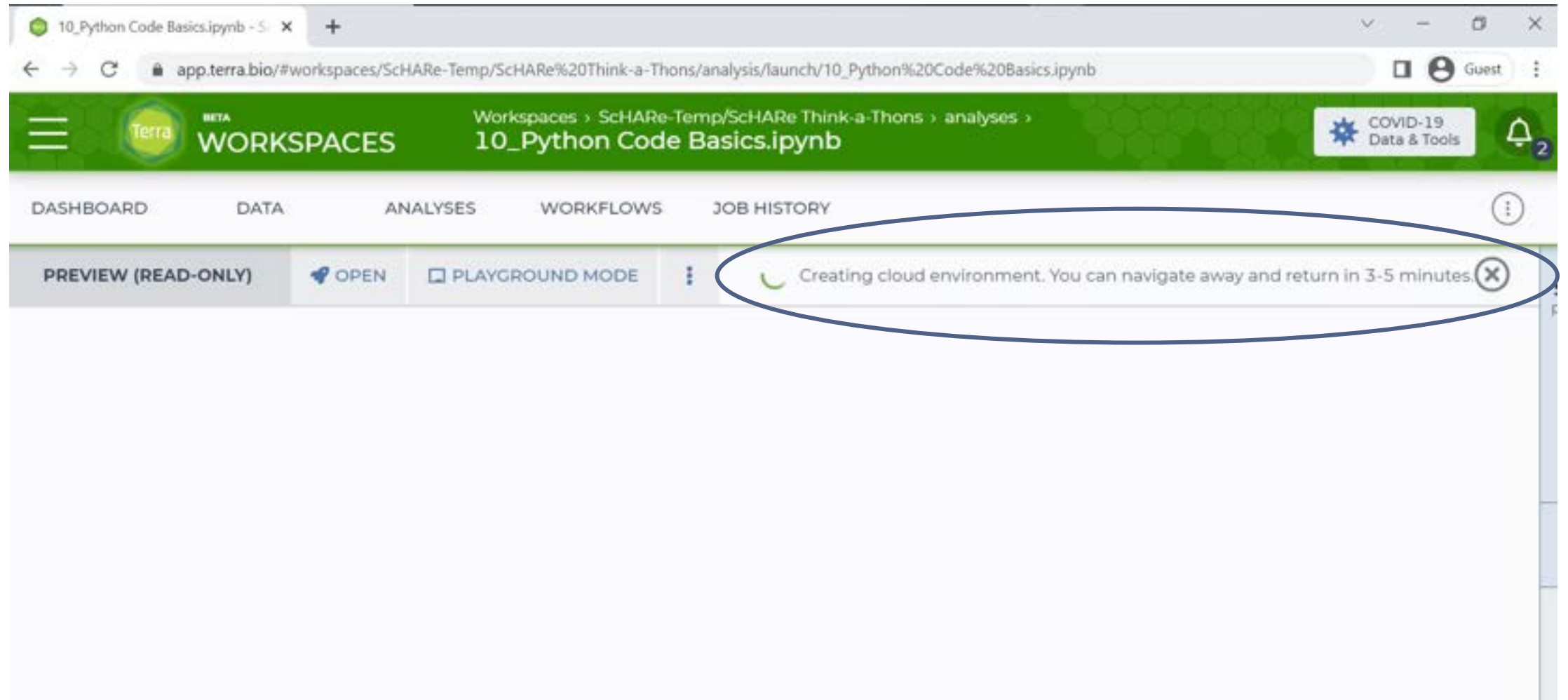
Disk Type: Standard

Disk Size (GB): 50

**CREATE**



# It will take some time...



The screenshot shows a web browser window with the URL `app.terra.bio/#workspaces/SchARE-Temp/SchARE%20Think-a-Thons/analysis/launch/10_Python%20Code%20Basics.ipynb`. The page header includes the Terra logo, the word "WORKSPACES", and the breadcrumb "Workspaces > SchARE-Temp/SchARE Think-a-Thons > analyses > 10\_Python Code Basics.ipynb". A notification bell icon shows 2 alerts. The main navigation bar contains "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOB HISTORY". Below this, there are buttons for "PREVIEW (READ-ONLY)", "OPEN", and "PLAYGROUND MODE". A blue oval highlights a message: "Creating cloud environment. You can navigate away and return in 3-5 minutes." with a close button (X).

When the system is ready, click on Playground mode:

The screenshot shows a web browser window with the URL `app.terra.bio/#workspaces/SchARE-Temp/SchARE%20Think-a-Thons/analysis/launch/10_Python%20Code%20Basics.ipynb`. The page header is green and contains the Terra logo, the word "WORKSPACES", and the breadcrumb "Workspaces > SchARE-Temp/SchARE Think-a-Thons > analyses > 10\_Python Code Basics.ipynb". A notification for "COVID-19 Data & Tools" is visible in the top right. Below the header is a navigation bar with "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOB HISTORY". The main content area has a control bar with "PREVIEW (READ-ONLY)", "OPEN", and "PLAYGROUND MODE" buttons. The "PLAYGROUND MODE" button is circled in blue. To the right of the buttons is a status message: "Creating cloud environment. You can navigate away and return in 3-5 minutes." with a close icon.

# Click on Continue:

10\_Python Code Basics.ipynb - S x +

app.terra.bio/#workspaces/SchARe-Temp/SchARe%20Think-a-Thons/analysis/launch/10\_Python%20Code%20Basics.ipynb

Guest

Workspaces > SchARe-Temp/SchARe Think-a-Thons > analyses >

### Playground Mode

Playground mode allows you to explore, change, and run the code, but your edits will not be saved.

To save your work, choose **Download** from the **File** menu.

Do not show again

CANCEL CONTINUE

Error Creating Cloud Environment

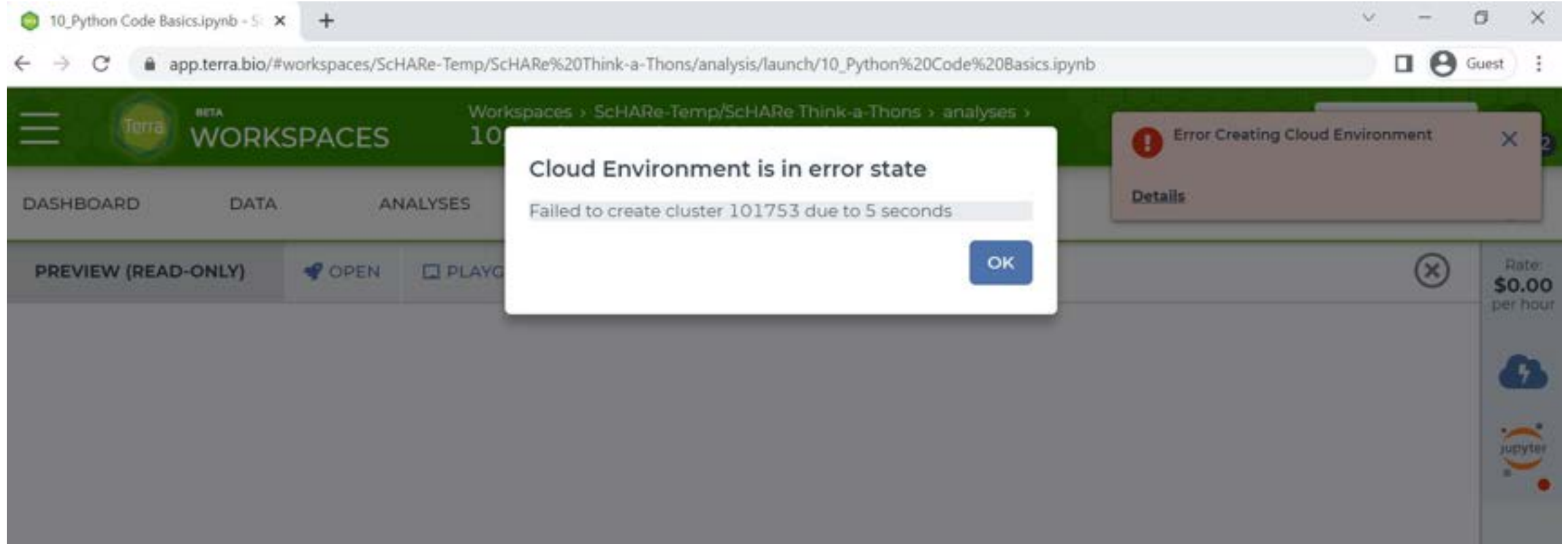
Details

Rate: \$0.00 per hour

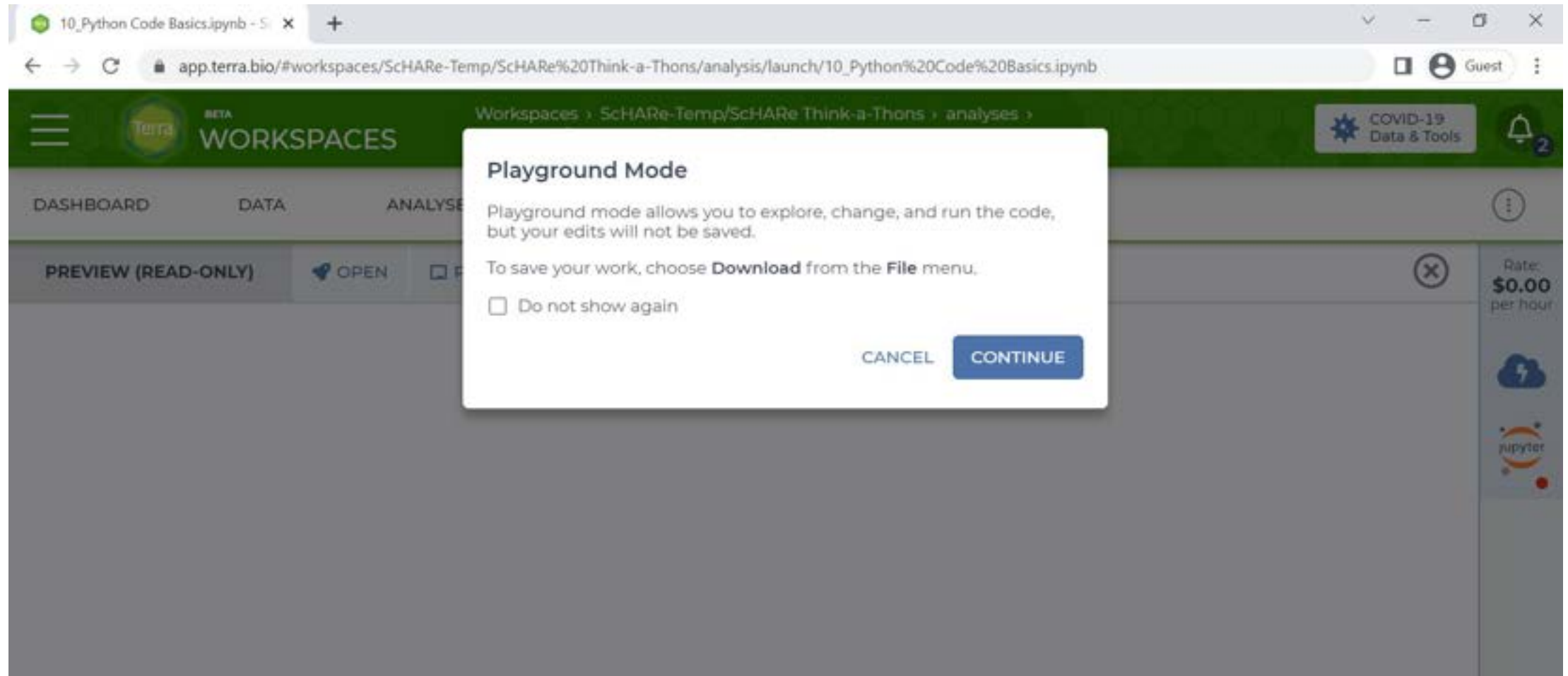
juptyer



**Note that you might encounter an error due to the large number of users – just try again in a few minutes:**



If all goes well you will see this:

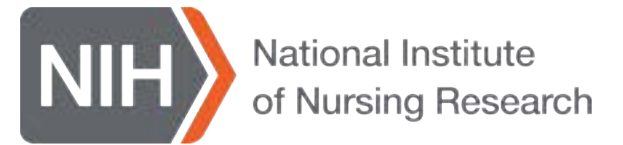


Click on Continue. You are done!



**Science**  
**collaborative for**  
**Health disparities and**  
**Artificial intelligence bias**  
**Reduction**

# Sci!ARe



# Thank you

## **NIMHD**

Dr. Eliseo  
Perez-Stable

## **ODSS**

Dr. Susan  
Gregurick

## **NIH/OD**

Dr. Larry  
Tabak

## **NINR**

Dr. Shannon  
Zenk

## **NINR**

Rebecca Hawes  
Micheal Steele  
John Grason

## **ORWH**

## **OMH**

## **NIMHD OCPL**

Kelli Carrington  
Thoko Kachipande  
Corinne Baker

## **BioTeam**

## **STRIDES**

## **Terra**

## **SIDEM**

## **RLA**

## **Broad Institute**

## **CCDE Working Group**

Deborah Duran  
Luca Calzoni  
Rebecca Hawes  
Micheal Steele  
Kelvin Choi  
Paula Strassle  
Michele Doose  
Deborah Linares  
Crystal Barksdale  
Gneisha Dinwiddie  
Jennifer Alvidrez  
Matthew McAuliffe  
Carolina Mendoza-Puccini  
Simrann Sidhu  
Tu Le

# Outline

**30'**      **Workshop setup**

- Experience poll

**5'**      **ScHARe and Terra overview**

- Interest poll

**5'**      **Why Python?**

**1h 10'**    **Guest Expert: Cindy Sheffield (NIH/OD/ORS)**

**An introduction to Python for Data Science – Part 2**

**5'**      **Python tutorials and resources**

**15'**      **How to set up a Terra billing project**

**20'**      **How to import ScHARe hosted data into your Terra workspace**

- Think-a-Thon poll

# Experience poll

Please check your level of experience with the following:

	None	Some	Proficient	Expert
Python	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cloud computing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Terra	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Health disparities research	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Health outcomes research	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Algorithmic bias mitigation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



# ScHARe

## Part I

### ScHARe and Terra overview





**SciARe**  
**Phase I**

**Population Science and SDoH datasets**  
**Tutorials and resources**  
**Think-a-Thons**

ScHARe is a **cloud-based population science data platform** designed to accelerate research in health disparities, health and healthcare delivery outcomes, and artificial intelligence (AI) bias mitigation strategies

ScHARe aims to fill **three critical gaps**:

- Increase participation of **women & underrepresented populations with health disparities** in data science through data science skills training, cross-discipline mentoring, and multi-career level collaborating on research
- Leverage population science, SDoH, and behavioral Big Data and cloud computing tools to foster a **paradigm shift** in healthy disparity, and health and healthcare delivery outcomes research
- **Advance AI bias mitigation and ethical inquiry** by developing innovative strategies and securing diverse perspectives



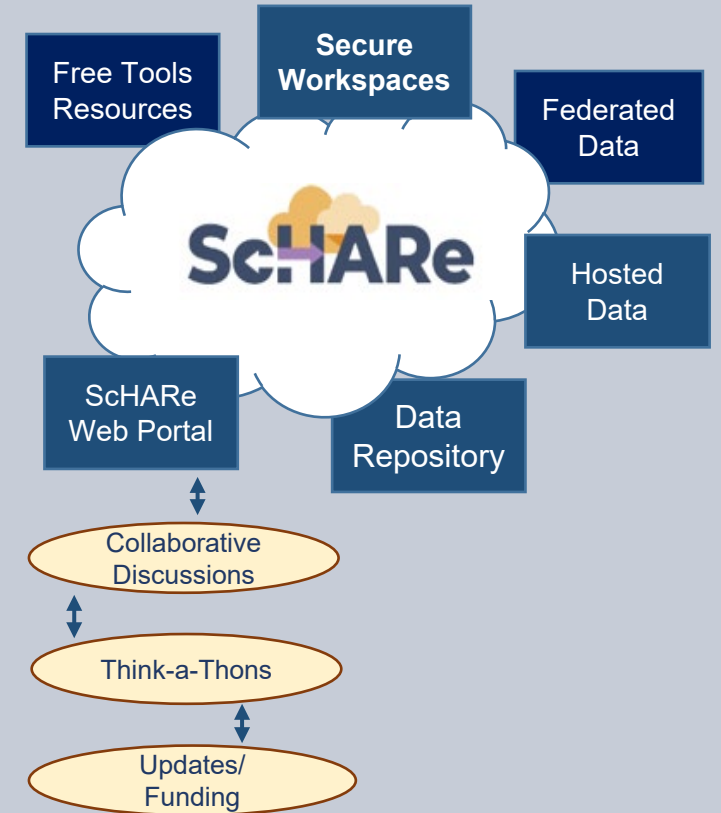
# ScHARe Components

ScHARe co-localizes within the cloud:

- **Datasets** (including social determinants of health and social science data) relevant to minority health, health disparities, and health care outcomes research
- **Data repository** to comply with the required hosting, managing, and sharing of data from NIMHD- and NINR-funded research programs
- **Computational capabilities and secure, collaborative workspaces** for students and all career level researchers
- **Tools for collaboratively evaluating and mitigating biases** associated with datasets and algorithms utilized to inform healthcare and policy decisions

**Frameworks:** Google Platform, Terra, GitHub, NIMHD Web ScHARe Portal

## Intramural & Extramural Resource



[nimhd.nih.gov/schare](http://nimhd.nih.gov/schare)

# SchARE Data Ecosystem

Researchers can access, link, analyze, and export a **wealth of datasets** within and across platforms relevant to research about health disparities, health care outcomes and bias mitigation, including:

- **Google Cloud Public Datasets:** publicly accessible, federated, de-identified datasets hosted by Google through the Google Cloud Public Dataset Program

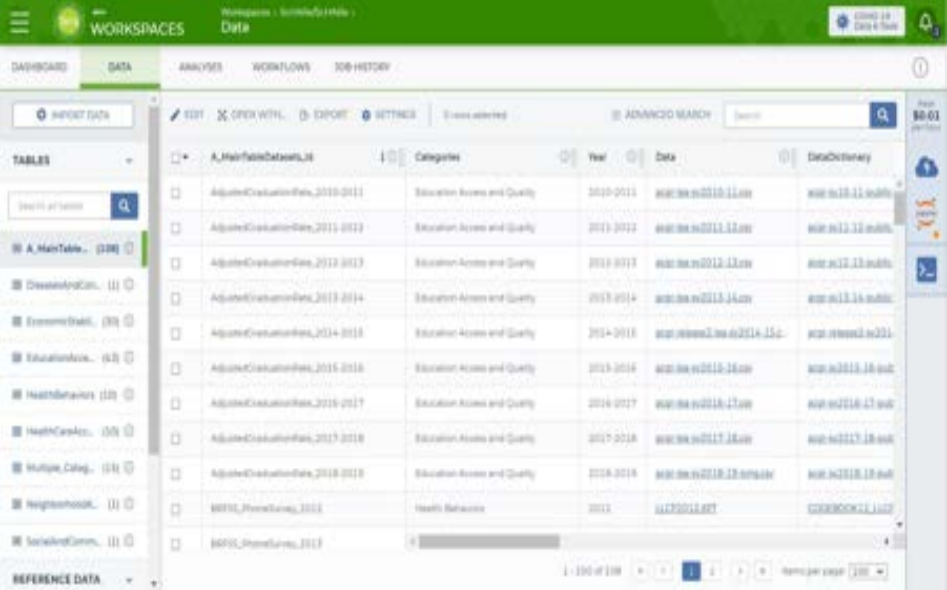
**Example:** *American Community Survey (ACS)*

- **SchARE Hosted Public Datasets:** publicly accessible, de-identified datasets hosted by SchARE

**Example:** *Behavioral Risk Factor Surveillance System (BRFSS)*

- **Funded Datasets on SchARE:** publicly accessible and controlled-access, funded program/project datasets using Core Common Data Elements shared by NIH grantees and intramural investigators to comply with the NIH Data Sharing Policy

**Examples:** *Jackson Heart Study (JHS); Extramural Grant Data; Intramural Project Data*



The screenshot shows the SchARE Data Ecosystem interface. The top navigation bar includes 'WORKSPACES', 'Data', and 'SEARCH'. Below the navigation bar, there is a 'TABLES' section on the left with a search bar and a list of categories: 'A\_MatData', 'DiseasesAndCon...', 'EconomicStab...', 'EducationAcc...', 'HealthBehav...', 'HealthCareAcc...', 'Multiple\_Categ...', 'Neighborhood...', and 'SocialAndComm...'. The main table displays a list of datasets with columns for 'Year', 'Data', and 'Dictionary'. A yellow box highlights the 'A\_MatData' category in the left sidebar.

Year	Data	Dictionary
2010-2011	AgeAdjustedRate_2010-2011	AgeAdjustedRate_2010-2011
2011-2012	AgeAdjustedRate_2011-2012	AgeAdjustedRate_2011-2012
2012-2013	AgeAdjustedRate_2012-2013	AgeAdjustedRate_2012-2013
2013-2014	AgeAdjustedRate_2013-2014	AgeAdjustedRate_2013-2014
2014-2015	AgeAdjustedRate_2014-2015	AgeAdjustedRate_2014-2015
2015-2016	AgeAdjustedRate_2015-2016	AgeAdjustedRate_2015-2016
2016-2017	AgeAdjustedRate_2016-2017	AgeAdjustedRate_2016-2017
2017-2018	AgeAdjustedRate_2017-2018	AgeAdjustedRate_2017-2018
2018-2019	AgeAdjustedRate_2018-2019	AgeAdjustedRate_2018-2019
2019-2020	AgeAdjustedRate_2019-2020	AgeAdjustedRate_2019-2020

On SchARE, datasets are categorized by content based on the CDC **Social Determinants of Health categories**:

1. Economic Stability
2. Education Access and Quality
3. Health Care Access and Quality
4. Neighborhood and Built Environment
5. Social and Community Context

with the addition of:

- **Health Behaviors**
- **Diseases and Conditions**

Users will be able to **map and link** across datasets

# Access to Population Science datasets



ScHARe Data Ecosystem will offer access to **300+ datasets**, including:

- Google Cloud Public Datasets
- ScHARe Hosted Public Datasets:
  - American Community Survey
  - U.S. Census
  - Social Vulnerability Index
  - Food Access Research Atlas
  - Medical Expenditure Panel Survey
  - National Environmental Public Health Tracking Network
  - Behavioral Risk Factor Surveillance System
- **Coming Soon:** Repository for Funded Datasets on ScHARe, in compliance with NIH Data Sharing Policy

# Cloud computing strategies



- Uses **workflows** in Workflow Description Language (**WDL**), a language easy for humans to read, for batch processing data
- **Python and R**, including most commonly used libraries
- Enables **customization** of computing environments to ensure everyone in your group is using the same software
- **Big Query** and **Tensorflow** access for advanced machine learning
- Enables researchers to create interactive **Jupyter notebooks** (documents that contain live code) and share data, analyses and results with their collaborators in real time
- For novice users, integration with **SAS** is planned



# AI bias mitigation strategies

- Widespread use of AI raises a number of ethical, moral, and legal issues – likely not to go away
- Algorithms often are “black boxes”
- **Biases can result from:**
  - social/cultural context not considered
  - design limitations
  - data missingness and quality problems
  - algorithm development and model training
  - Implementation
- If not rectified, biases may result in decisions that lead to discrimination, unequitable healthcare, and/or health disparities
- **Lack of diverse perspectives:** populations with health disparities are underrepresented in data science
- **Guidelines** and recommendations emerging from HHS, NIST, White House, etc.



Critical thinking can rectify AI biases

ScHARe was created to:

- foster participation of **populations with health disparities in data science**
- promote the collaborative identification of **bias mitigation strategies** across the continuum
- create a **culture of ethical inquiry** and critical thinking whenever AI is utilized
- build **community confidence** in implementation approaches
- focus on **implementation of AI bias** guidelines and recommendations



**SciARe**  
**Phase II**  
(in process)

**Data ecosystem and repository**

# SciHARe Data Repository

**CORE COMMON DATA ELEMENTS**

**NOVEL CDE FOCUSED  
REPOSITORY TO FOSTER  
INTEROPERABILITY**

**COMPLY WITH DATA SHARING  
POLICY - HOST PROJECT DATA**

## **DATA ECOSYSTEM**

- Map across datasets
- Map across platforms



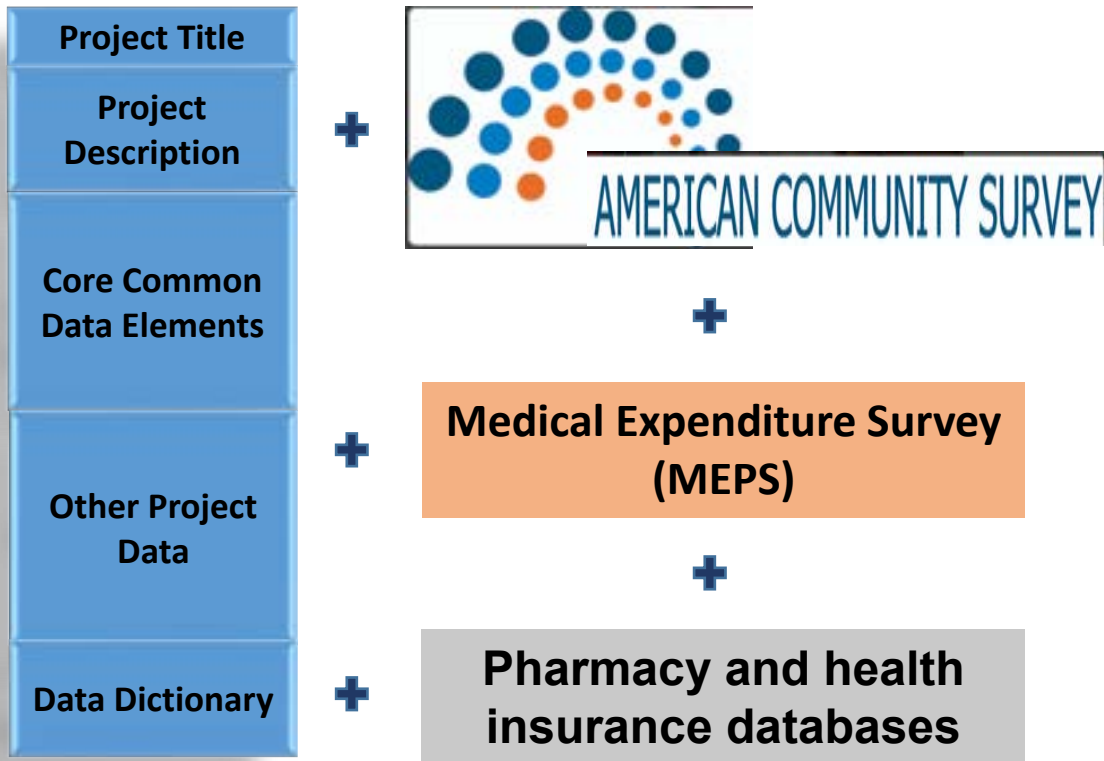
**UPCOMING**







## Project & federated dataset mapping



## Mapping across cloud platforms

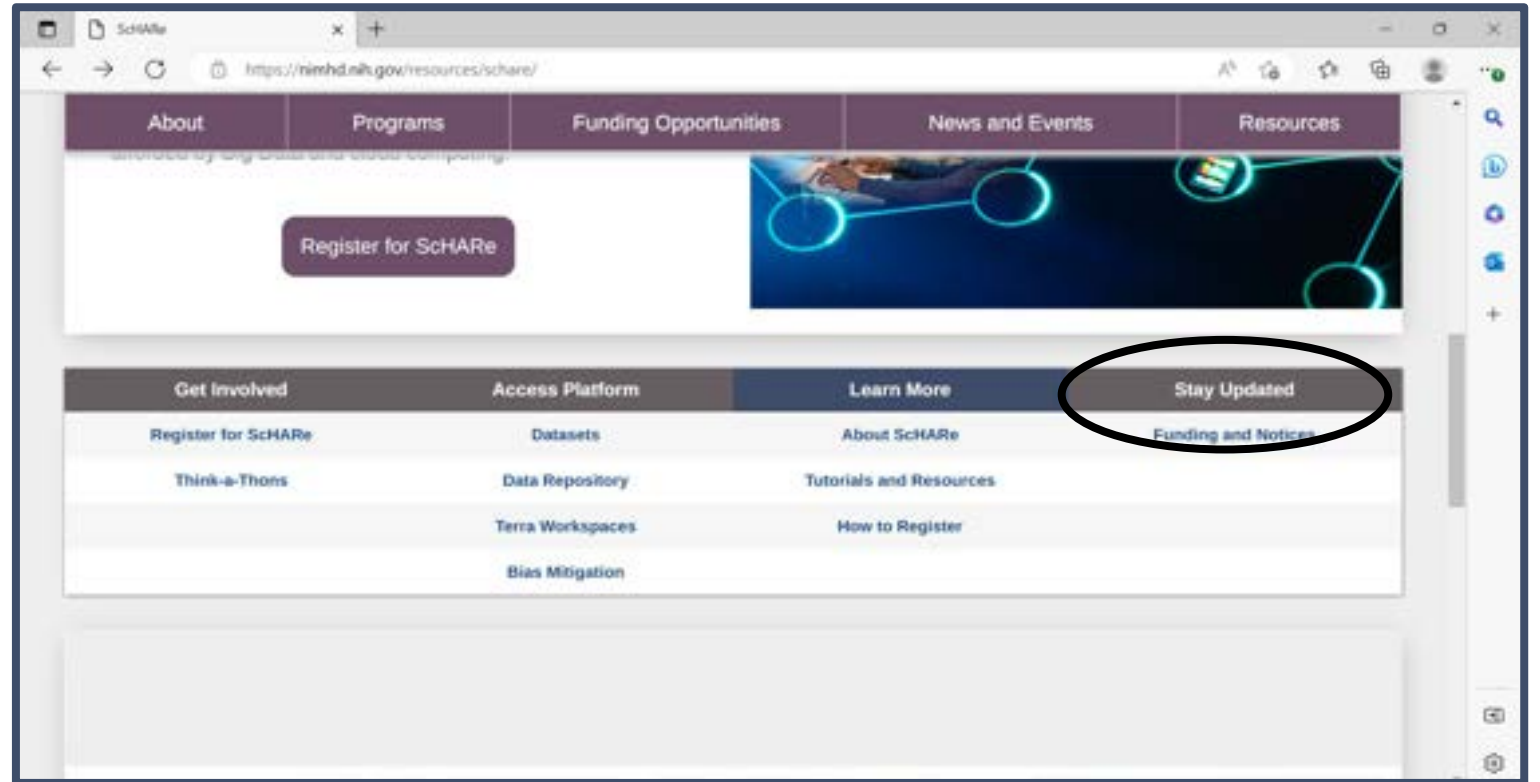


UPCOMING

# Two ways to sign up for ScHARe news



Scannable from your screen!



[nimhd.nih.gov/schare](https://nimhd.nih.gov/schare)

# Interest poll

**I am interested in (check all that apply):**

- Learning about Health Disparities and Health Outcomes research to apply my data science skills
- Conducting my own research using AI/cloud computing and publishing papers
- Connecting with new collaborators to conduct research using AI/cloud computing and publish papers
- Learning to use AI tools and cloud computing to gain new skills for research using Big Data
- Learning cloud computing resources to implement my own cloud
- Developing bias mitigation and ethical AI strategies
- Other

# ScHARE Think-a-Thons (TaT)

- Monthly sessions (2 1/2 hours)
- Instructional/interactive
- Designed for new and experienced users
- Research & analytic teams to:
  - Conduct health disparities, health outcomes, bias mitigation research
  - Analyze/create tools for bias mitigation
- Publications from research team collaboration
- Networking
- Mentoring and coaching
- Focus:
  - ✓ **Instructional**
  - ✓ **Collaboration research teams**
  - ✓ **Bias mitigation**

ScHARE

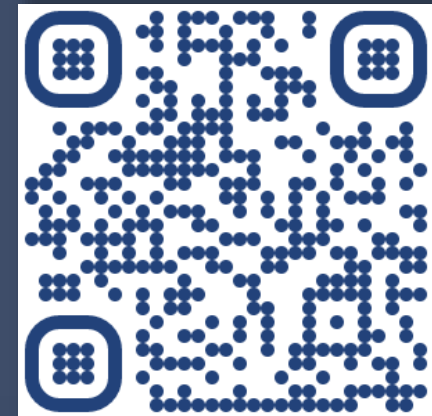
Think-a-Thon

Artificial Intelligence and  
Cloud Computing Basics

**Terra: Datasets and  
Analytics**



**Register:**



[bit.ly/think-a-thons](https://bit.ly/think-a-thons)



# SciARe

**Part II**  
**Why Python?**

# What is Python?

Python is a **computer programming language** used in data science to:

- manipulate and analyze data and conduct statistical calculations
- create data visualizations
- build machine learning algorithms

Python's **data science libraries** are powerful. Examples include:

- **Numpy** - for linear algebra and high-level mathematical functions
- **Pandas** - for handling data structures and manipulating tables
- **SciPy** - for data science tasks like interpolation and signal processing
- **Scikit-learn** - a machine learning library that is useful for classification, regression, and clustering algorithms
- **PyBrain** - for machine learning tasks and to test and compare algorithms



## Sources

[www.quanhub.com/python-for-data-science/](http://www.quanhub.com/python-for-data-science/)  
[coursera.org](https://www.coursera.org)



# What is R?

R is a **programming language** for statistical computing and graphics

It is used by data miners, bioinformaticians and statisticians for data analysis

Users have created **packages** to augment its functions

Third-party **graphical user interfaces** are also available, such as Rstudio



supports **both Python and R**



# Why Python?

According to SlashData:

- there are 8.2 million Python users
- **69%** of machine learning developers and data scientists **use Python (vs. 24%** of them **using R)**

Source  
[stackify.com/learn-python-tutorials/](https://stackify.com/learn-python-tutorials/)

# How to learn Python

## How long does it take to learn Python?

It can take **2 to 5 months**, but you can write your first short program in **minutes**

## Can you learn Python with no experience?

Python is the **perfect** programming language for **people without any coding experience**, as it has a simple syntax and is very accessible to beginners

**Unfamiliar terminology** may be a barrier, which today's workshop will hopefully help you overcome

Links to additional **free learning resources** will be provided at the end



# SciARe

## Part III

An introduction to Python for Data Science - 2

# Sci!ARe



**Guest expert**

**Cindy Sheffield**

**NIH/OD/ORS**

# About Cindy

Cindy is **Data Services Librarian** at the NIH Library.

She began her **library career** at the Johns Hopkins Medical Institutions with a focus on Evidenced Based Medicine. She progressed within the Welch Medical Library, leaving Hopkins as the Associate Director of Education Services.

Cindy has worked at several **federal agencies** including the Department of Homeland Security, the Department of Defense, and the Department of Health and Human Services. Within DHHS she was worked for both the National Institutes of Health and the Federal Drug Administration.

Her **focus** has always been on using key resources to identify the best evidence, and then to organize and manage that evidence in a way that makes sense for users. At the NIH she works with various user groups to support literature research and data science.

She is the Outreach Librarian for the NIH Clinical Centers, Pain and Palliative Care Team, the Eunice Kennedy Shriver, National Institute of Child and Human Development, the Administration for Children and Families, and the Office of the National Coordinator for Health Information Technology.

# ScHARe Think-a-Thon: An Introduction to Python for Data Science

Cindy Sheffield, NIH Library  
Data Services

Why learn Python?

What to know about reading, writing and running Python code?

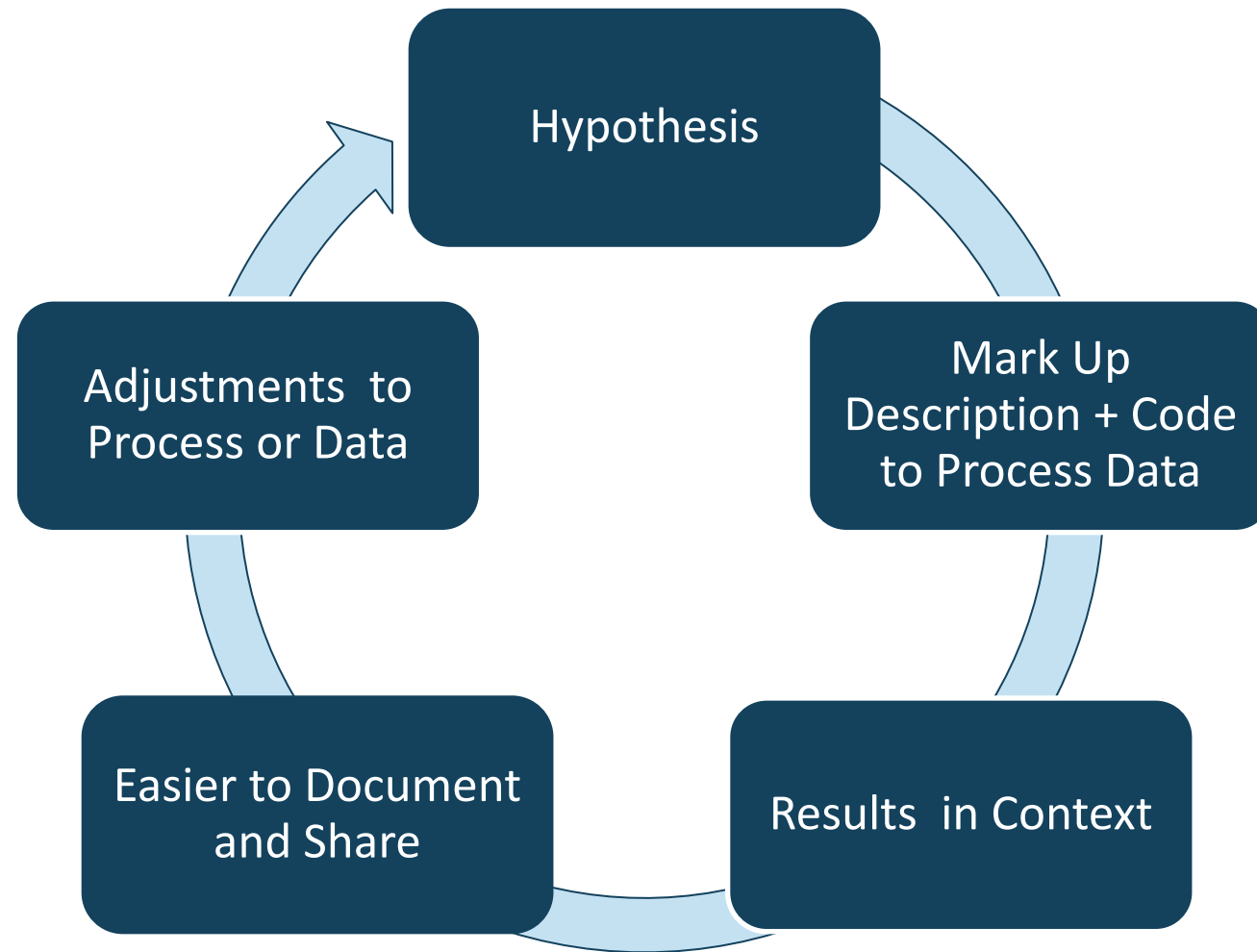
Popular IDEs for reading, writing and running Python code:

- Google Colab
- Spyder
- Jupyter Notebook
- Jupyter Lab

# Why Learn Python?



- **User friendly** - easy to read and easy to learn.
- **Efficient** - facilitates data management, data analysis, process documentation, and visualizations via an ecosystem of libraries.
- **Productivity** – clear syntax, object-oriented design, and code reusability enable developers to write and adapt code efficiently.
- **Dynamic** – Design cycles tend to be short as code can be written and tested
- **Open Source** – Python software and its IDEs are distributed freely for most part. There are inexpensive IDEs. Cost is not a barrier to learning. It simply takes time and effort.



# What you need to know about reading, writing, and running Python?

- Syntax
- Variables
- Functions
- Data Types
- Conditionals
- Libraries

#Comments

identifiers = a to z

(tuples use parenthesis)

[lists use brackets]

{dictionaries use squiggly brackets}

loops and if statements require a colon:  
and proper indentation  
**4 spaces, no less, no more!**

# Syntax: Reserve Words

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

<https://flexiple.com/python/python-reserved-words/#section1>

a = "A string of characters"

pi=3.1415

y = (1:5)      x = 2

myfile = "/myfile.csv"

c = a+b \* (3.14)

Fruits=('apples', 'bananas', 'cantaloupe', 'dates')



```
greet("New Python Coders")
```

```
def greet (named):  
    print("Hello " + named + " !")
```

**Hello New Python Coders!**

*In this example 'greet' is the Function name and 'named' is the Parameter variable, the function headers always begins with 'def' and ends with a colon.*

Python Built-in Functions

– <https://www.pythoncheatsheet.org/cheatsheet/built-in-functions>

**Tuples** – Sequence Type, parentheses

MyTuple = (parentheses, exponents, multiplication/division, addition/subtraction)

**Lists** – Sequence Type, square brackets

names = ["Annie", "Betty", "Cindy"]

cities = ["Albany", "Baltimore", "Cincinnati"]

items = ["apples", "bananas", "cantaloupe"]

**Dictionaries** – Mapping Type, squiggly brackets

MyDictionary = {"names": "Annie", "items": "apples"}

## Loops

```
fruit_list=['apples', 'bananas', 'cantaloupe']
```

```
for i in range(0,3):  
    print(fruit_list[i])
```

apples  
bananas  
cantaloupe

## If Statements

```
x = 4; y = 6
```

```
if x < y:  
    print(x, 'is less than', y)  
else:  
    print(x, 'is not less than', y)
```

**4 is less than 6**

## Pandas Library

```
import pandas as pd
```

function | library | assign | short for pandas

*Enables code:*

```
df_alphasong = pd.read_csv("/alphasong.csv")
```

*Example data frame*

alphasong.csv =

NAMES	CITIES	ITEMS
Annie	Albany	Apples
Betty	Baltimore	Bananas
Cindy	Cincinnati	cantaloupe

Python Libraries can be found at [PyPI.org](https://pypi.org).

PyPI is the Python Package Index: a repository of software for Python programming

- Statistical Analysis and Visualizations Libraries include:

- Pandas (data frames)
- NumPy (math functions for arrays)
- Statsmodels
- TensorFlow
- Scikit-learning
- Matplotlib (np + Visualizations)
- Plotly
- Seaborn (Visualizations)

```
import pandas as pd
import numpy as np
import seaborn as sb
```

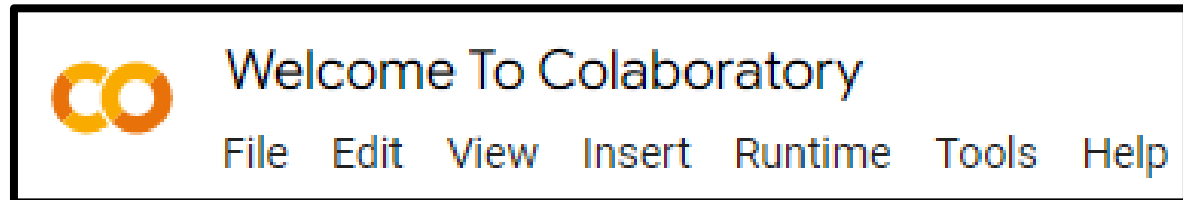
- Syntax
- Variables
- Functions
- Data types
- Conditionals
- Libraries

# IDE for Python Code



## Popular IDEs for reading, writing and running Python code:

- **Google Colab**
- Spyder
- Jupyter Notebook
- JupyterLab





Welcome To Colaboratory

File Edit View Insert

<https://colab.research.google.com/notebooks/welcome.ipynb>

How to Create Notebooks in Colab.ipynb ☆  
File Edit View Insert Runtime Tools Help Last edited on October 25

+ Code + Text

▶ **How to Create a Python File in Google Colab**

Step 1: Mount your Drive This will let Colab where to find your code and data files.

```
[ ] import os
from google.colab import drive
drive.mount('/content/drive/')

Mounted at /content/drive/

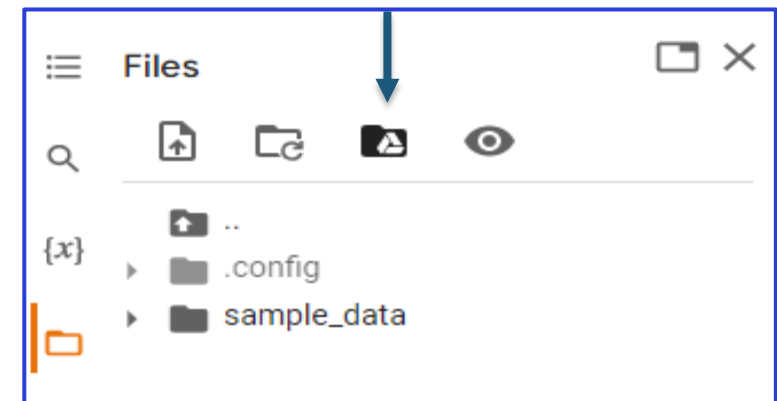
[ ] print ("Hello NIH")

Hello NIH
```

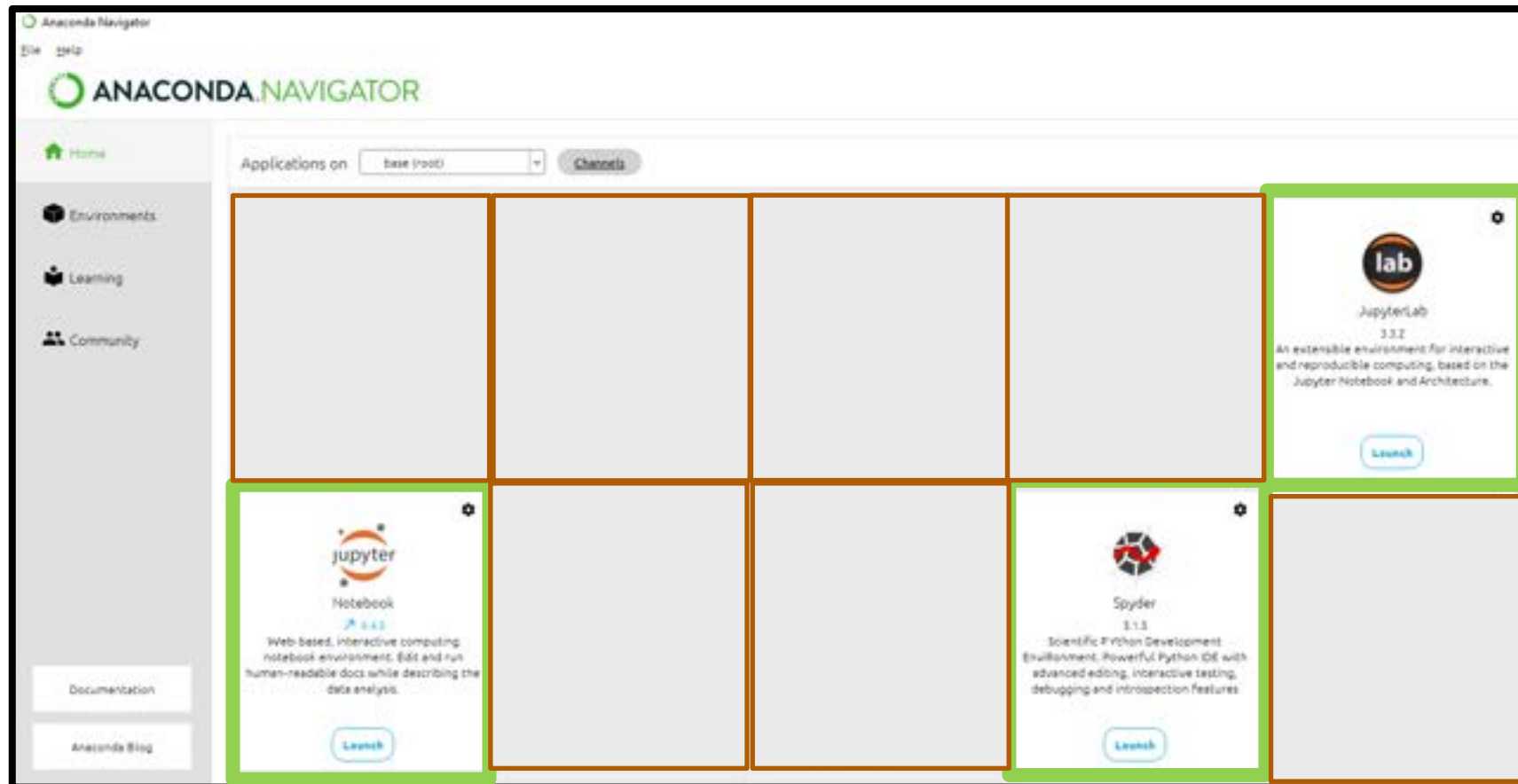
- Create a new Google Account for Colab.
- Go: <https://colab.research.google.com/notebooks/welcome.ipynb>
- Click on **Getting Started** for a quick overview of Colab
- Use **File / Notebook** to find online guides.
- Connect to files Google Drive by either:

**Type text below or select Drive icon under Files**

```
import os  
from google.colab import drive  
drive.mount('/content/drive/')
```



- Jupyter Notebook
- Spyder
- JupyterLab

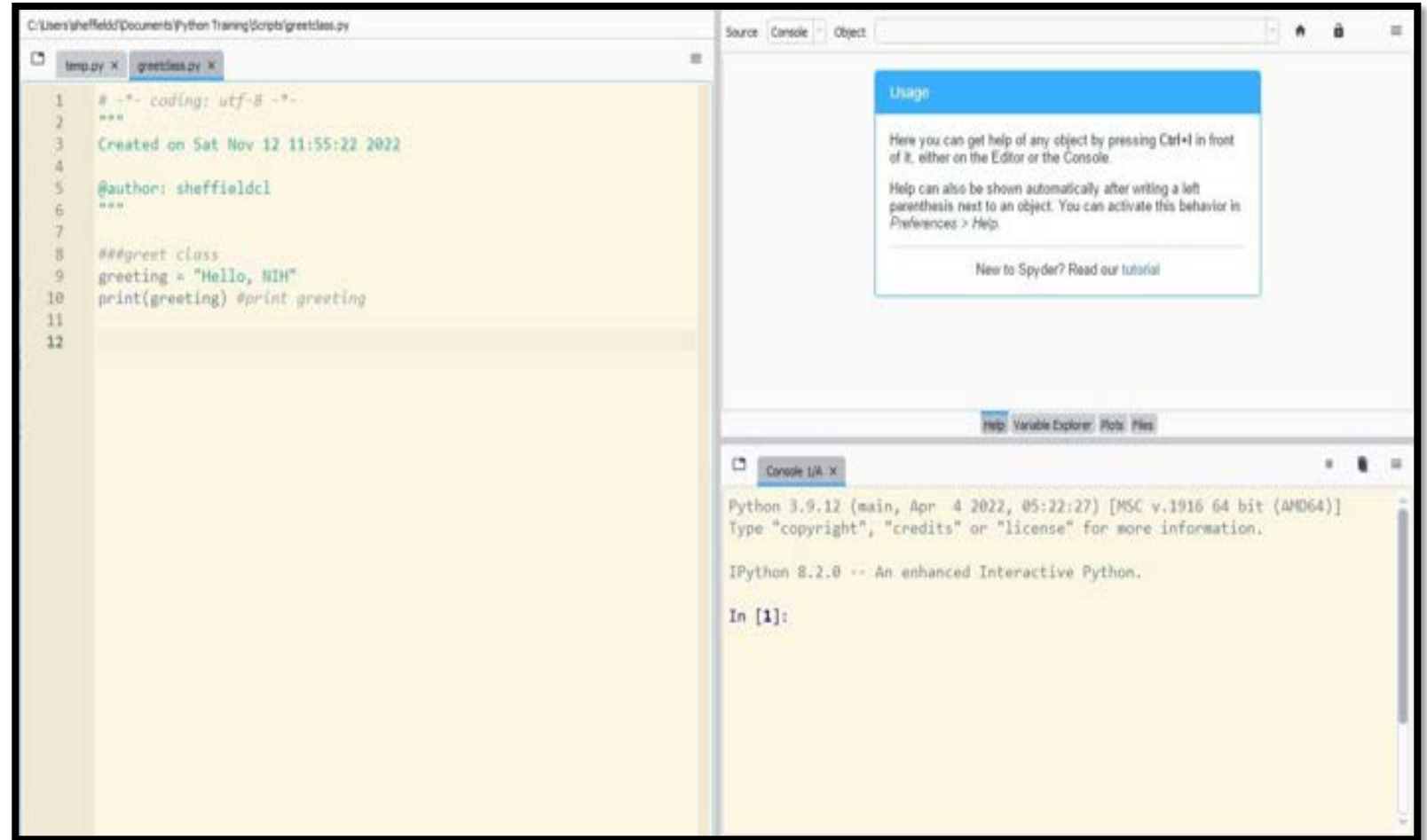


Install Anaconda  
Launch Spyder  
File/Open/*open script*

OR

File/Create/*create new*

- Ctrl-L will give you help.
- There is a tutorial.



<https://docs.spyder-ide.org/current/quickstart.html>

# Jupyter Notebook

- 1) Open Anaconda,
- 2) Launch Jupyter notebook
- 3) Select 'New'
- 4) Select 'Python 3 (ipykernel)'
- 5) Save "filename"

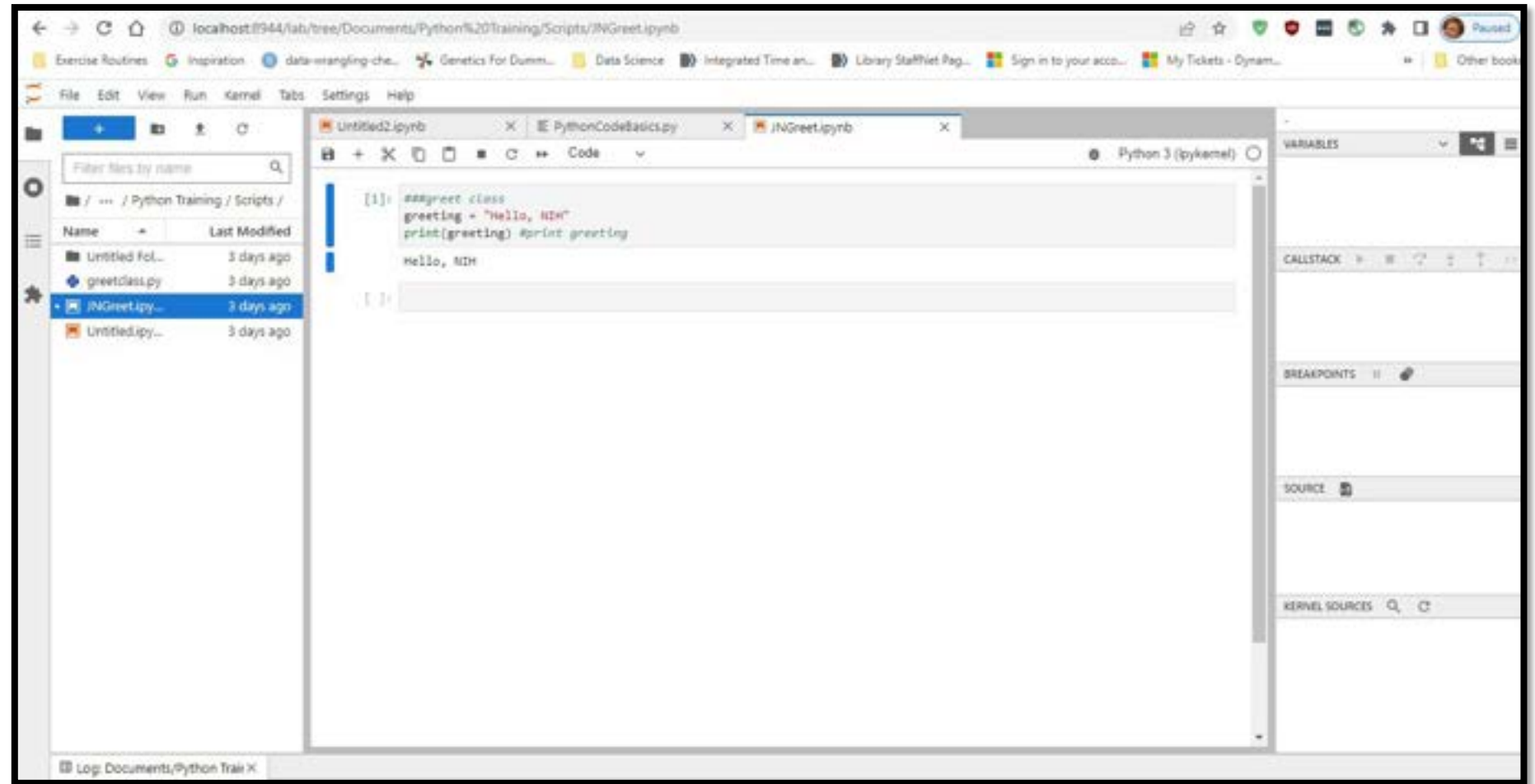


The screenshot displays the Jupyter Notebook web interface. At the top left is the 'jupyter' logo. On the right, there are 'Quit' and 'Logout' buttons. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' Below this is a file browser showing the path '/ Documents / Python Training / Scripts'. The file list includes 'Untitled Folder', 'Untitled.ipynb', and 'greetclass.py'. A red box highlights the 'New' dropdown menu, which is open and shows options: 'Python 3 (ipykernel)', 'Text File', 'Folder', and 'Terminal'. A red star with the number '1' points to the 'New' button, and another red star with the number '2' points to the 'Python 3 (ipykernel)' option.

- Install Anaconda
- Launch:  
JupyterLab
- File/Open/*open script*

OR

- File/Create/  
*create new*



- **Console:** - Where code runs
- **Project Files:** - Code and data files
- **Variables:** - One Identifier to assign something to a variable
- **Data View:** - View of the data frame being processed
- **Plots:** - View of the graph being generated
- **History:** - Code history processes in active memory
- **Autocomplete:** - Python feature to help with coding
- **Debugging:** - Python feature to help with finding errors
- **Markdown:** - Code to add text around the code



1. **Console:** Within Colab there isn't a separate console, but instead the active code and related error messages occur within each cell.
2. **Project Files:** To see where your Notebook is stored use the folder icon on the left to navigate to your MyDrive / Colab Notebooks. "PythonCodeBascis".
3. **Variables:** (Show current variables) We have a notebook here with code, but none of this code is NOT in our active memory, because the only code that has been run is the cell to initialize our PythonCodeBasics notebook and the little bit of code we just recreated. Let's run the code in a few more of these cells.
4. **Data View:** Simply double left click on the data file to view the table on the right side of the Colab screen.
5. **Libraries:** All of the Python libraries are available to Colab so it is a matter of importing each library you want to use and assigning that library to a shortcut or nickname such as `import pandas as pd`. You will need to have knowledge about the libraries you want to use and why you want to use them.
6. **Plots:** Once you run the code and import the associated data for plots, the plots will appear directly within Colab. There is also a 'View output full screen' feature using the menu to the right.
7. **History:** To view the History of code run during a session select the View menu, and then 'Execute code history'.
8. **Autocomplete:** Note the autocomplete feature will try and complete and function such as 'print' or argument that appears within your active Variables list.
9. **Debugging:** If you have an error in your code, an error message will appear to help with debugging.
10. **Comments and Markdown:** Comments can be added using 3 hash marks on any line of code. Markdown text can be added using the + Text tool.

1. **Console:** The console is immediately viewable in the lower right pane. If I highlight my first line of code in the script and run it using the F9 key, it will run the code and the results from the code in the Console. If I highlight and run the next line, “2\*3” and use the F9 key will give me that code and result as well.
2. **Project Files:** We can get to Project Files by using File / Open or Open Recent to find the .py file you choose to use.
3. **Variables:** Can be viewed in the upper right pane using the “Variable Explorer” tab. Note each data type will be a different color.
4. **Data View:** Using the Files tab in the top right pane, identify your data set, click on it to see the content. If you’d like to see the content in a spreadsheet you can right click on the file and use the dropdown menu to select “Open externally”. It will open in Excel outside of Spyder.
5. **Libraries:** Libraries are need for dataframe, calculation and visuals. Import needed libraries.
6. **Plots:** Can be viewed in the upper right pane, using the Plots tab.
7. **History:** The History can be viewed in the lower right pane, using the History tab.
8. **Autocomplete:** Similar to Colab Sypder will autocomplete functions and arguments that are in active memory.
9. **Debugging:** There is a Debug menu at on the top menu bar of Spyder
10. **Comments and Markdown:** Comments can be noted using triple single quotes for large areas of text and the # sign for line comments. Markdown is not an option in Spyder.

1. **Console:** The format is very similar to Google Colab, with the code and related error messages being generated in the main screen.
2. **Project Files:** Can be open via the File dropdown menu. New project files can be created using that option from the dropdown menu.
3. **Variables:** With pandas running, in one of the empty cells run “%whos”
4. **Data View:** To view data frames in one of the empty cells use the function `display(<data frame name>)`  
Example: `display(iris)` *may need to specify file path*
5. **Libraries:** Need to be familiar with libraries and import them and assign them short names.
6. **Plots:** Plots will appear directly in the cells with the code to generate them.
7. **History:** In one of the empty cells run “%history”
8. **Autocomplete:** Start typing the first few letters of a function or argument and then hit the Tab key, and Jupyter Notebooks will autocomplete the rest of the code.
9. **Debugging:** Error messages usually show within the cell. Alt + Shift + Enter is another way to see error codes.
10. **Comments and Markdown:** Comments can be created with quotes and # signs. Markdown is available from dropdown menu immediately under the Widgets option on the top menu bar.

1. Jupyter Labs is very similar to Jupyter Notebooks.
2. The main difference is that you can have multiple tabs open with notebooks or .csv files in this IDE. Multiple consoles can be run as well. This is helpful if you are trying to run different sets of code with different applications.

# Test Your Knowledge

Log on to one of the IDEs.

Type the follow code:

```
name = “#” #Add your name  
str = “ thinks Python is ” #note spaces before and after clause!  
descpt = “#” #Add descriptor to indicate how you feel about Python  
print (name + str +descpt)
```

- Google's Python Class
  - <https://developers.google.com/edu/python/>
- Learn Python – Free Python Courses for Beginners
  - <https://www.freecodecamp.org/news/learn-python-free-python-courses-for-beginners/>
- Books
  - Learning Python, O'Reilly
  - Head First Python, Paul Barry
  - Python Crash Course, Eric Matthes

- **BioPython: freely available Python tools for computational molecular biology and bioinformatics**  
<https://academic.oup.com/bioinformatics/article/25/11/1422/330687?login=true>
- **Design of Experiments (DOE) with python**  
<https://medium.com/mlearning-ai/design-of-experiments-doe-with-python-be88f5c013f5>
- **Introduction to Jupyter Notebook | Jupyter Notebook Tutorial**  
<https://youtu.be/1A7tea9LSEk>
- **JupyterLab Tutorial for Everyone**  
[https://youtu.be/mspsHlk\\_qUQ](https://youtu.be/mspsHlk_qUQ)



**National Institutes of Health**  
*Office of Management*





# SciIARe

Part IV

Python tutorials and resources

# Python resources

You can take advantage of the dozens of “**Python for data science**” **online tutorials** for beginners and advanced programmers listed here:

- [Stackify - 30+ Tutorials to Learn Python](#)
- [FreeCodeCamp - Code Class for Beginners](#)
- [Harvard – Free Python Course](#)
- [Coursera – Free and Paid Python Courses](#)
- [LearnPython – Free Interactive Python Tutorials](#)
- [BestColleges – 10 Places to Learn Python for Free](#)

# Python resources

## Stackify

### 30+ Tutorials to Learn Python

#### Top 30 Python Tutorials

In this article, we will introduce you to some of the best **Python tutorials**. These tutorials are suited for both beginners and advanced programmers. With the help of these tutorials, you can learn and polish your coding skills in Python.

1. [Udemy](#)
2. [Learn Python the Hard Way](#)
3. [Codecademy](#)
4. [Python.org](#)
5. [Invent with Python](#)
6. [Pythonspot](#)
7. [AfterHoursProgramming.com](#)
8. [Coursera](#)
9. [Tutorials Point](#)
10. [Codementor](#)
11. [Google's Python Class eBook](#)
12. [Dive Into Python 3](#)
13. [NewCircle Python Fundamentals Training](#)
14. [Studytonight](#)
15. [Python Tutor](#)
16. [Crash into Python](#)
17. [Real Python](#)
18. [Full Stack Python](#)
19. [Python for Beginners](#)
20. [Python Course](#)
21. [The Hitchhiker's Guide to Python!](#)
22. [Python Guru](#)
23. [Python for You and Me](#)
24. [PythonLearn](#)
25. [Learning to Python](#)
26. [Interactive Python](#)
27. [PythonChallenge.com](#)
28. [IntelliPaat](#)
29. [SoloLearn](#)
30. [W3Schools](#)

# Python resources

## FreeCodeCamp

Code Class for Beginners



The screenshot shows the freeCodeCamp website interface. At the top right, the logo 'freeCodeCamp (▲)' is visible. Below it, a blue navigation bar contains the text 'Learn to code — free 3,000-hour curriculum'. The main content area features two article cards. The first card has the title 'Python Tutorial for Beginners (Learn Python in 5 Hours)' and a description: 'In [this TechWorld with Nana YouTube course](#), you will learn about strings, variables, OOP, functional programming and more. You will also build a couple of projects including a countdown app and a project focused on API requests to Gitlab.' The second card has the title 'Scientific Computing with Python' and a description: 'In [this freeCodeCamp certification course](#), you will learn about loops, lists, dictionaries, networking, web services and more.'

freeCodeCamp (▲)

Learn to code — [free 3,000-hour curriculum](#)

### Python Tutorial for Beginners (Learn Python in 5 Hours)

In [this TechWorld with Nana YouTube course](#), you will learn about strings, variables, OOP, functional programming and more. You will also build a couple of projects including a countdown app and a project focused on API requests to Gitlab.

### Scientific Computing with Python


In [this freeCodeCamp certification course](#), you will learn about loops, lists, dictionaries, networking, web services and more.

# Python resources

## Harvard


Free Python Course


Catalog > Computer Science Courses > HarvardX's Computer Science for Web Programming




## Harvard University: CS50's Introduction to Computer Science

An introduction to the intellectual enterprises of computer science and the art of programming.

 **12 weeks**  
6-18 hours per week

 **Self-paced**  
Progress at your own speed

**There is one session available:**  
4,974,616 already enrolled! After a course session ends, it will be [archived](#) .

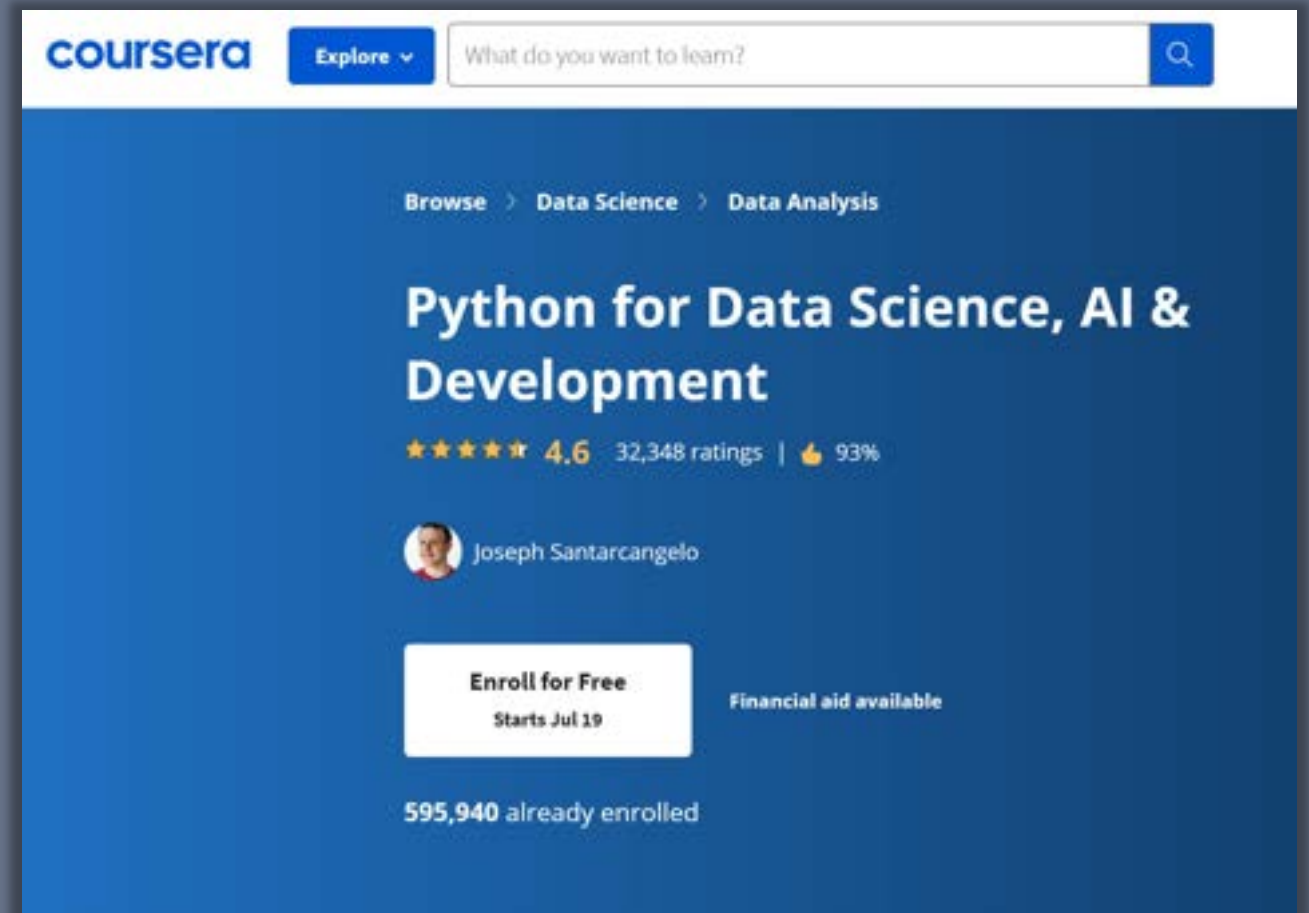
**Starts Jul 19**  
Ends Dec 31

**Enroll**

# Python resources

## Coursera

Free and Paid Python Courses



The screenshot shows the Coursera website interface. At the top, there is a search bar with the text "What do you want to learn?" and a search icon. Below the search bar, there is a navigation menu with "Browse", "Data Science", and "Data Analysis". The main heading is "Python for Data Science, AI & Development". Below the heading, there is a rating of 4.6 stars based on 32,348 ratings, and a thumbs-up icon with "93%". The instructor's name, "Joseph Santarcangelo", is displayed next to a profile picture. A prominent white button says "Enroll for Free" with "Starts Jul 19" below it. To the right of the button, it says "Financial aid available". At the bottom, it states "595,940 already enrolled".

**Enroll for Free**  
Starts Jul 19

Financial aid available

595,940 already enrolled

# Python resources

## LearnPython

Free Interactive Python Tutorials

### Learn the Basics

- [Hello, World!](#)
- [Variables and Types](#)
- [Lists](#)
- [Basic Operators](#)
- [String Formatting](#)
- [Basic String Operations](#)
- [Conditions](#)
- [Loops](#)
- [Functions](#)
- [Classes and Objects](#)
- [Dictionaries](#)
- [Modules and Packages](#)

### Data Science Tutorials

- [Numpy Arrays](#)
- [Pandas Basics](#)

### Advanced Tutorials

- [Generators](#)
- [List Comprehensions](#)
- [Lambda functions](#)
- [Multiple Function Arguments](#)
- [Regular Expressions](#)
- [Exception Handling](#)
- [Sets](#)
- [Serialization](#)
- [Partial functions](#)
- [Code Introspection](#)
- [Closures](#)
- [Decorators](#)
- [Map, Filter, Reduce](#)

# Python resources

## BestColleges

### 10 Places to Learn Python for Free

Bootcamp Types ▾ Reviews ▾ Resources ▾ About ▾ BestColleges.com

## Top 10 Free Python Courses

### Google's Python Class

Students with some programming language experience can learn Python with Google's intensive two-day course. While there are no official prerequisites, students need a basic understanding of programming language concepts, such as if statements.

Learners initially explore strings and lists using lecture videos and written materials. A coding exercise follows each section, and the exercises become increasingly complex.

This Python course gives students hands-on practice with complete programs, working with text files, processes, and HTTP connections.

### Microsoft's Introduction to Python Course

Students can learn Python online and build a simple input/output program with Microsoft's introductory Python course. There are no prerequisites for this short, eight-unit, 16-minute class.

This online Python course is part of Microsoft's Python learning paths. It prepares students with the concepts and basic skills to pursue more advanced learning.

Students explore Python code, where to run Python apps, learn how to declare variables, and use the Python interpreter. They also learn how to access free resources.



# Terra resources

If you are new to Terra, we also recommend exploring the following resources:

- [Overview Articles](#): Review high-level docs that outline what you can do in Terra, how to set up an account and account billing, and how to access, manage, and analyze data in the cloud
- [Video Guides](#): Watch live demos of the Terra platform's useful features
- [Terra Courses](#): Learn about Terra with free modules on the Leanpub online learning platform
- [Data Tables QuickStart Tutorial](#): Learn what data tables are and how to create, modify, and use them in analyses
- [Notebooks QuickStart Tutorial](#): Learn how to access and visualize data using a notebook
- [Machine Learning Advanced Tutorial](#): Learn how Terra can support machine learning-based analysis



# SCIARe

Part V

Billing and costs

# What are the cloud costs of working on Terra?

The Terra platform infrastructure is **free to use**

However, the following operations in Terra **may incur charges**:

## 1. Virtual Machine compute costs

In cloud computing, a **virtual machine** is an emulation of a computer system that provides the functionality of a physical computer

Terra allows you to **customize** the characteristics of your virtual machine based on your computation needs (more on this later)

- A **high-performance machine costs more**
- You will be charged for the **time you use the machine**

The screenshot displays the configuration options for a cloud compute profile. It includes a 'Cloud compute profile' section with 'CPUs' set to 1 and 'Memory (GB)' set to 3.75. There is an unchecked checkbox for 'Enable GPUs' with a 'BETA' label and a link to learn more. The 'Compute type' is set to 'Standard VM'. There is a checked checkbox for 'Enable autopause' with a link to learn more, and a '15' minute inactivity timer. The 'Location' is set to 'us-central1 (Iowa) (default)'. The 'Persistent disk' section shows 'Disk Type' as 'Standard' and 'Disk Size (GB)' as 10.

Setting	Value
CPUs	1
Memory (GB)	3.75
Enable GPUs	<input type="checkbox"/> BETA
Compute type	Standard VM
Enable autopause	<input checked="" type="checkbox"/>
Minutes of inactivity	15
Location	us-central1 (Iowa) (default)
Disk Type	Standard
Disk Size (GB)	10

# What are the cloud costs of working on Terra?

## 2. Data storage

- You will be charged for any data stored in the storage spaces (“**buckets**”) associated with your account

## 3. Data egress (i.e. moving data) costs

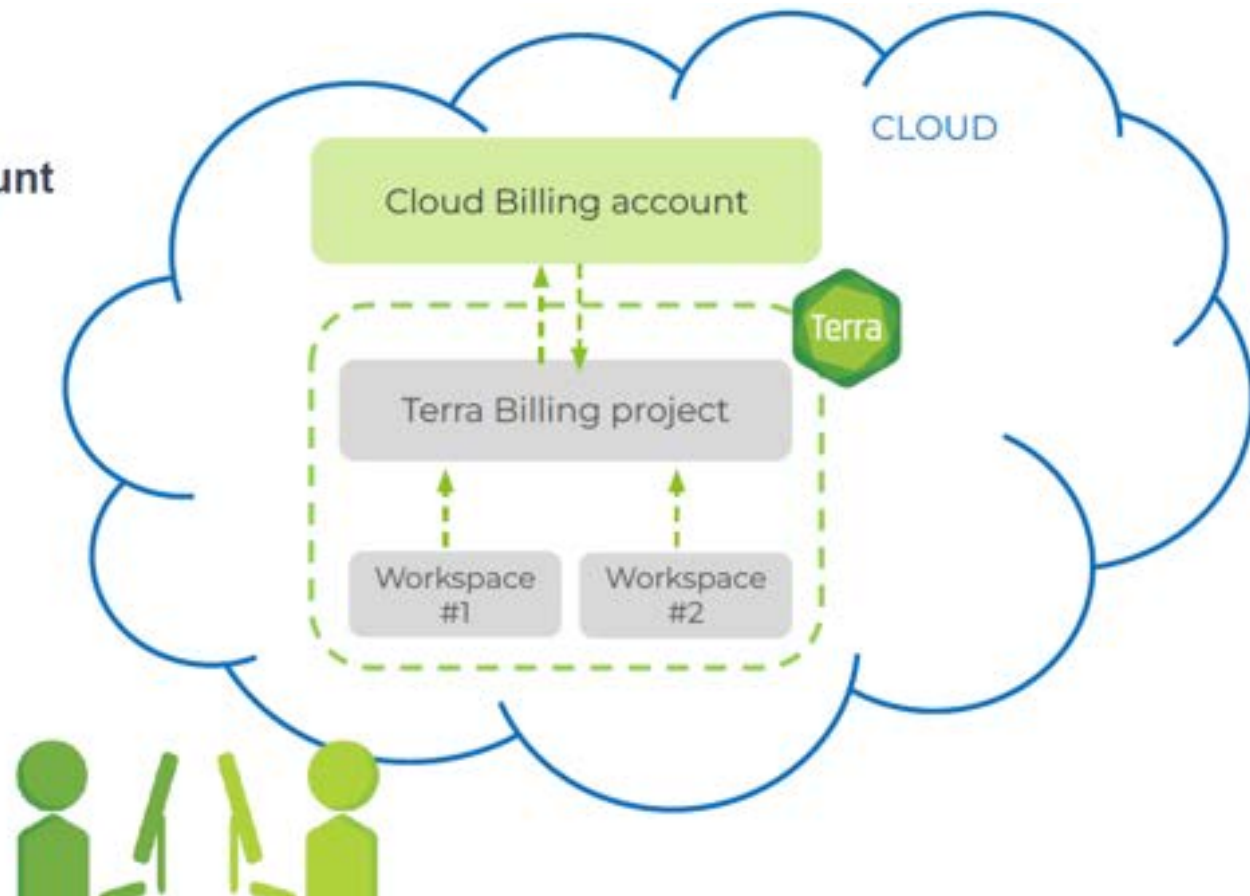
- When creating a bucket to store data, you are asked to set its location. This is because the data are going to be stored in data warehouses located in physical places (“**regions**” – more info [here](#)). Regions exist, among other reasons, to accommodate the need of certain users to keep their data in defined regions.

You will pay to **move stored data between regions**

# How will I be charged for these costs?

Terra runs on Google Cloud Platform (GCP). All Terra costs are GCP fees that are ultimately paid for by a **Google Cloud Billing account** linked to Terra – specifically, to a **Terra Billing project**

- ▶ Each Billing project is linked to an umbrella Google **Cloud Billing account**
- ▶ A **Terra Billing project** is a pass-through assigned to a workspace when you create it
- ▶ All GCP fees (storage, compute, egress) are charged **per workspace** - *regardless of who does the analysis or whether they have access to a billing project.*



# How will I be charged for these costs?

## Will I incur any costs today?

Today, **access to a free temporary billing project** will allow you to run all the materials with your instructors

## What happens after today?

You will no longer have access to the free temporary billing project. If you want to access work-in-progress from the Think-a-Thon, you will need to **set up your own billing** and copy any of your workspaces to your own billing

**Next, we will show you how to set up your own billing**

# Get \$300 in free Google Cloud credits

If you've never used Google Cloud before, **you are eligible for \$300 in free Google Cloud credits** you can use for working in Terra

## Conditions for Google Cloud credits eligibility

- You haven't previously signed up for the Free Trial
- You've never been a paying customer of Google Cloud, Google Maps Platform, or Firebase
- If you're part of an organization that uses Google Cloud, your email will likely not be eligible



Google Cloud

## What can I do with my credits in Terra?

The credits will cover anything that has a cost in Terra - such as storing data and running analyses. You can't use credits to add GPUs to your computing resources, and you are limited to 4 workspaces at a time

## How long will my \$300 credits be available?

Your credits will be available for 3 months, or until you have used up all \$300. Once your credits run out or expire, you can upgrade to a paid account

# 3 easy steps to set up billing

1. Sign in to the [Google Cloud Console](#) with your Terra user ID and **set up a Google Cloud Billing account**

You'll be invited to activate your free trial: **you won't be billed until the credits expire**

2. In the [Google Cloud Console Billing page](#), **link your Google Cloud Billing and Terra accounts**

Add terra-billing@terra.bio as a Principal, with Billing Account User role

Use the same Google ID for both the Cloud Billing account and your Terra user name

3. In the [Terra Billing page](#), **create a Terra Billing project**

Select the previously created Google Cloud Billing account to fund your Terra Billing project

For detailed instructions, see [this Terra page](#)



# Understanding and monitoring costs

You can **ESTIMATE COSTS**:

1. **analysis costs**
2. cloud storage costs
3. egress (i.e., data moving) costs

You can **CHECK ACTUAL COSTS** in the Google Cloud Platform Console

You can **REDUCE COSTS** in several ways (for advanced users)

1) *Adjust settings to optimize cost (VM and disk)*

2) *Estimate costs using real-time cost/hour in Cloud Environment widget*

- ▶ Updates based on the machine configuration you choose
- ▶ Total cost (estimate) = (cost/hour) x (hours the VM will be active) + cost of the Persistent Disk
- ▶ Autopause function saves money!

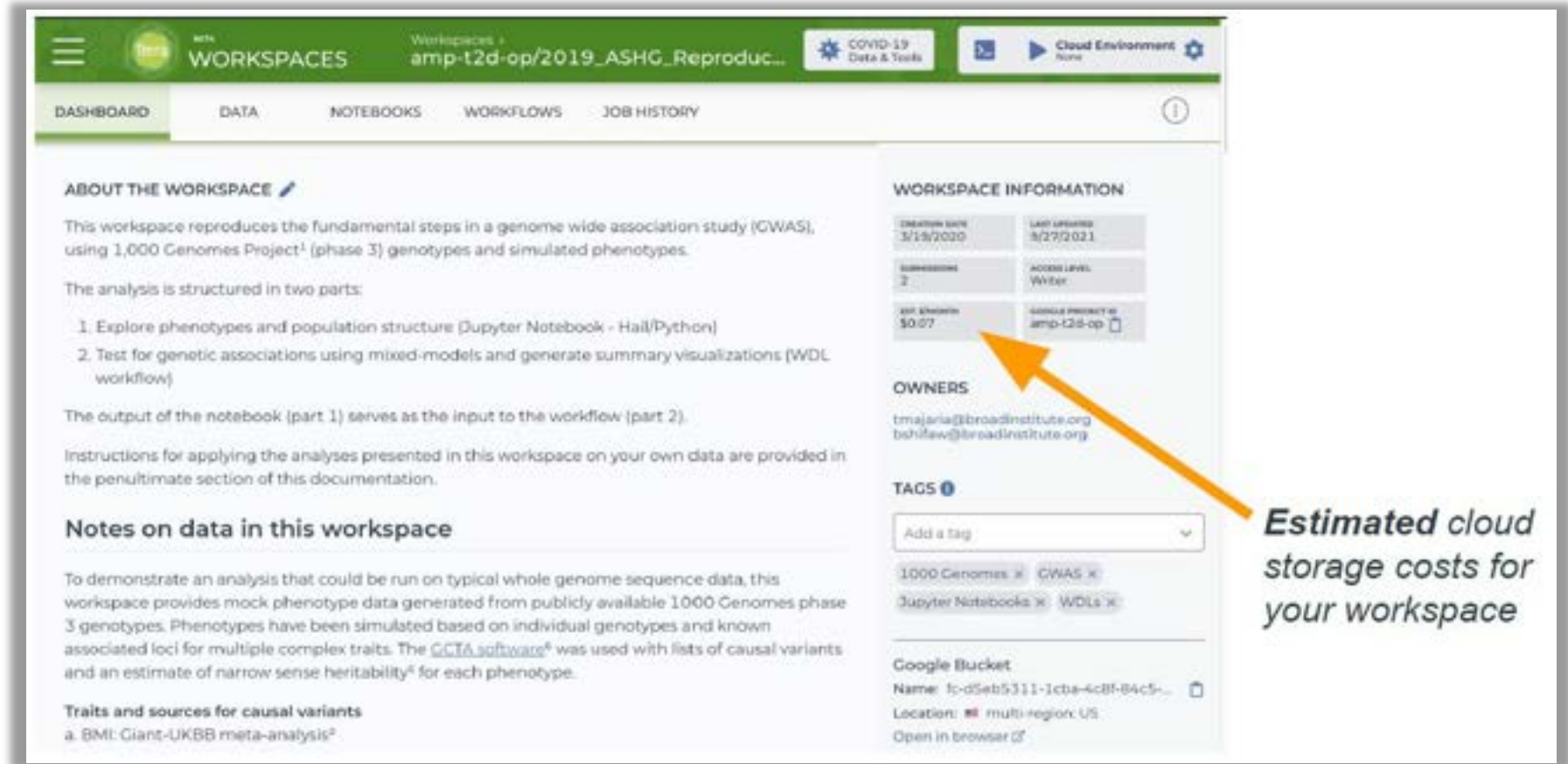
# Understanding and monitoring costs

You can **ESTIMATE COSTS**:

1. analysis costs
2. **cloud storage costs**
3. egress (i.e., data moving) costs

You can **CHECK ACTUAL COSTS** in the Google Cloud Platform Console

You can **REDUCE COSTS** in several ways (for advanced users)



The screenshot displays the Google Cloud Workspaces interface for a workspace named "amp-t2d-op/2019\_ASHG\_Reproduc...". The interface includes a navigation bar with "DASHBOARD", "DATA", "NOTEBOOKS", "WORKFLOWS", and "JOB HISTORY". The main content area is divided into two columns. The left column, titled "ABOUT THE WORKSPACE", contains text describing the workspace's purpose (reproducing GWAS steps) and a list of two parts: "1. Explore phenotypes and population structure (Jupyter Notebook - Hail/Python)" and "2. Test for genetic associations using mixed-models and generate summary visualizations (WDL workflow)". The right column, titled "WORKSPACE INFORMATION", contains a table with the following data:

CREATION DATE	LAST UPDATED
3/19/2020	3/27/2021
SUBMISSIONS	ACCESS LEVEL
2	Writer
EST. STORAGE	SAMPLE PROJECT ID
\$0.07	amp-t2d-op

An orange arrow points from the "EST. STORAGE" value of "\$0.07" to a text box on the right that reads "Estimated cloud storage costs for your workspace". Below the "WORKSPACE INFORMATION" section, there are sections for "OWNERS" (listing email addresses), "TAGS" (with a search box and tags like "1000 Genomes", "GWAS", "Jupyter Notebooks", "WDLs"), and "Google Bucket" (with details like Name, Location, and an "Open in browser" link).

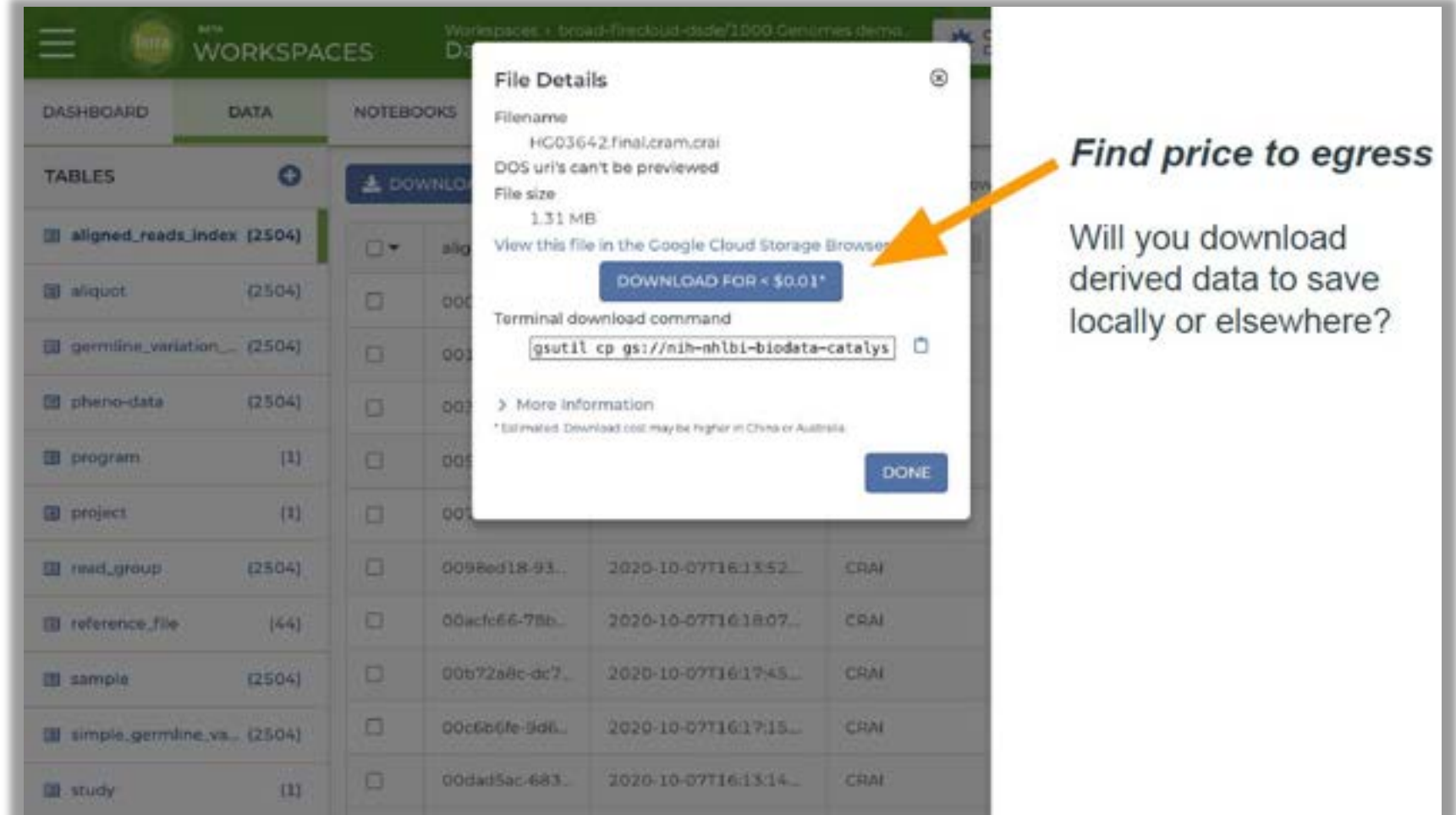
# Understanding and monitoring costs

You can **ESTIMATE COSTS**:

1. analysis costs
2. cloud storage costs
3. egress (i.e., data moving) costs

You can **CHECK ACTUAL COSTS** in the Google Cloud Platform Console

You can **REDUCE COSTS** in several ways (for advanced users)



**File Details**

Filename  
HG03642.final.cram.crai

DOS uri's can't be previewed

File size  
1.31 MB

View this file in the Google Cloud Storage Browser

**DOWNLOAD FOR < \$0.01\***

Terminal download command  
`gsutil cp gs://nih-nhlbi-biodata-catalys`

> More information

\* Estimated. Download cost may be higher in China or Australia.

**DONE**

*Find price to egress*

Will you download derived data to save locally or elsewhere?

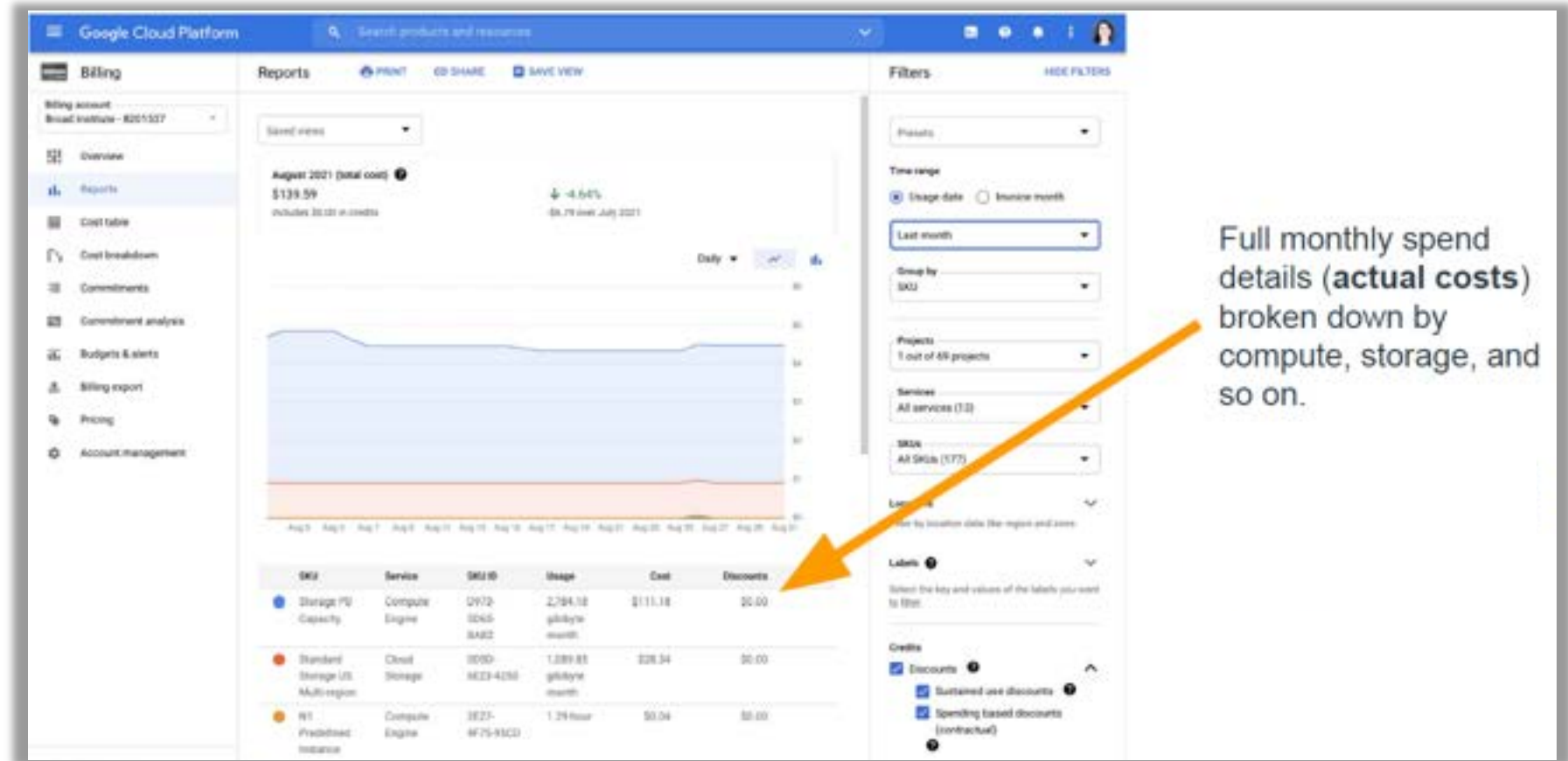
# Understanding and monitoring costs

You can **ESTIMATE COSTS**:

1. analysis costs
2. cloud storage costs
3. egress (i.e., data moving) costs

You can **CHECK ACTUAL COSTS** in the Google Cloud Platform Console

You can **REDUCE COSTS** in several ways (for advanced users)



# Understanding and monitoring costs

You can **ESTIMATE COSTS**:

1. analysis costs
2. cloud storage costs
3. egress (i.e., data moving) costs

You can **CHECK ACTUAL COSTS** in the Google Cloud Platform Console

You can **REDUCE COSTS** in several ways ([guides are for advanced users](#))

*Terra allows you to find the right balance between cost and time*

## **Saving on workflow costs**

- ▶ Delete intermediate files: [guide](#)
- ▶ Call-caching: [guide](#)
- ▶ Checkpointing: [guide](#)
- ▶ Preemptible VMs: [guide](#)

## **Saving Cloud Environment costs**

- ▶ Size application compute appropriately: [guide](#)
- ▶ Move generated data to regional or nearline storage: [guide](#)
- ▶ Autopause: [guide](#)

## **Saving on storage costs**

- ▶ Ask how much are you storing, where are you storing it, and how frequently will you access it?
- ▶ Move data to regional or nearline storage: [guide](#)



# ScHARe

## Part VI

How to import ScHARe hosted data into  
your Terra workspace



# ScHARe Data

As a reminder, on ScHARe you can work with:

## Data you upload to your workspace

This is your own personal project data, stored on your computer

## Data already in the ScHARe Data Ecosystem

1. Google Hosted Public Datasets
2. ScHARe Hosted Public Datasets
3. ScHARe Hosted Project Datasets

# ScHARe Ecosystem

The ScHARe Data Ecosystem is comprised of:

- 1. Google Hosted Public Datasets:** publicly accessible, federated, de-identified datasets hosted by Google through the Google Cloud Public Dataset Program  
*Example: American Community Survey (ACS)*
- 2. ScHARe Hosted Public Datasets:** publicly accessible, de-identified datasets hosted by ScHARe  
*Example: Behavioral Risk Factor Surveillance System (BRFSS)*
- 3. ScHARe Hosted Project Datasets:** publicly accessible and controlled-access, funded program/project datasets using Core Common Data Elements shared by NIH grantees and intramural investigators to comply with the NIH Data Sharing Policy  
*Examples: Jackson Heart Study (JHS); Extramural Grant Data; Intramural Project Data*



# SciARe Ecosystem: Google hosted datasets

Examples of interesting datasets include:

- **American Community Survey** (U.S. Census Bureau)
- **US Census Data** (U.S. Census Bureau)
- **Area Deprivation Index** (BroadStreet)
- **GDP and Income by County** (Bureau of Economic Analysis)
- **US Inflation and Unemployment** (U.S. Bureau of Labor Statistics)
- **Quarterly Census of Employment and Wages** (U.S. Bureau of Labor Statistics)
- **Point-in-Time Homelessness Count** (U.S. Dept. of Housing and Urban Development)
- **Low Income Housing Tax Credit Program** (U.S. Dept. of Housing and Urban Development)
- **US Residential Real Estate Data** (House Canary)
- **Center for Medicare and Medicaid Services - Dual Enrollment** (U.S. Dept. of Health & Human Services)
- **Medicare** (U.S. Dept. of Health & Human Services)
- **Health Professional Shortage Areas** (U.S. Dept. of Health & Human Services)
- **CDC Births Data Summary** (Centers for Disease Control)
- **COVID-19 Data Repository by CSSE at JHU** (Johns Hopkins University)
- **COVID-19 Mobility Impact** (Geotab)
- **COVID-19 Open Data** (Google BigQuery Public Datasets Program)
- **COVID-19 Vaccination Access** (Google BigQuery Public Datasets Program)

# ScHARe Ecosystem: ScHARe hosted datasets

Organized based on the **CDC SDoH categories**, with the addition of *Health Behaviors and Diseases and Conditions*:

200+ datasets

- What are the Social Determinants of Health?

Social determinants of health (SDoH) are the **nonmedical factors that influence health outcomes**.

They are the **conditions in which people are born, grow, work, live, and age, and the wider set of forces and systems shaping the conditions of daily life**.



# **ScHARe Ecosystem: ScHARe hosted datasets**

Examples of datasets for each category include:

## **Education access and quality**

Data on graduation rates, school proficiency, early childhood education programs, interventions to address developmental delays, etc.

Examples:

- **EDFacts Data Files** (U.S. Dept. of Education) - Graduation rates and participation/proficiency assessment
- **NHES - National Household Education Surveys Program** (U.S. Dept. of Education) – Educational activities

# ScHARe Ecosystem: ScHARe hosted datasets

## Health care access and quality

Data on health literacy, use of health IT, emergency room waiting times, preventive healthcare, health screenings, treatment of substance use disorders, family planning services, access to a primary care provider and high quality care, access to telehealth and electronic exchange of health information, access to health insurance, adequate oral care, adequate prenatal care, STD prevention measures, etc.

Example:

- **MEPS - Medical Expenditure Panel Survey (AHRQ)** - Cost and use of healthcare and health insurance coverage
- **Dartmouth Atlas Data** - Selected Primary Care Access and Quality Measures - Measures of primary care utilization, quality of care for diabetes, mammography, leg amputation and preventable hospitalizations

# ScHARe **Ecosystem**: ScHARe hosted datasets

## Neighborhood and built environment

Data on access to broadband internet, access to safe water supplies, toxic pollutants and environmental risks, air quality, blood lead levels, deaths from motor vehicle crashes, asthma and COPD cases and hospitalizations, noise exposure, smoking, mass transit use, etc.

Examples:

- **National Environmental Public Health Tracking Network** (CDC) - Environmental indicators and health, exposure, and hazard data
- **LATCH - Local Area Transportation Characteristics for Households** (U.S. Dept. of Transportation) – Local transportation characteristics for households

# ScHARe Ecosystem: ScHARe hosted datasets

## Social and community context

Data on crime rates, imprisonment, resilience to stress, experiences of racism and discrimination, etc.

Example:

- **Hate crime statistics** (FBI) - Data on crimes motivated by bias against race, gender identity, religion, disability, sexual orientation, or ethnicity
- **General Social Survey** (GSS) - Data on a wide range of characteristics, attitudes, and behaviors of Americans.

# ScHARe **Ecosystem**: ScHARe hosted datasets

## Economic stability

Data on unemployment, poverty, housing stability, food insecurity and hunger, work related injuries, etc.

Examples:

- **Current Population Survey (CPS) Annual Social and Economic Supplement** (U.S. Bureau of Labor Statistics ) - Labor force statistics: annual work activity, income, health insurance, and health
- **Food Access Research Atlas** (U.S. Dept. of Agriculture) – Food access indicators for low-income and other census tracts

# ScHARe Ecosystem: ScHARe hosted datasets

## Health behaviors

Data on health-related practices that can directly affect health outcomes.

Examples:

- **BRFSS - Behavioral Risk Factor Surveillance System (CDC)** - State-level data on health-related risk behaviors, chronic health conditions, and use of preventive services
- **YRBSS - Youth Risk Behavior Surveillance System (CDC)** – Health behaviors that contribute to the leading causes of death, disability, and social problems among youth and adults



# ScHARe **Ecosystem**: ScHARe hosted datasets

## Diseases and conditions

Data on incidence and prevalence of specific diseases and health conditions.

Examples:

- **U.S. CDI - Chronic Disease Indicators** (CDC) - 124 chronic disease indicators important to public health practice
- **UNOS - United Network of Organ Sharing** (Health Resources and Services Administration) – Organ transplantation: cadaveric and living donor characteristics, survival rates, waiting lists and organ disposition

**Let's see how to access the ScHARe hosted datasets:**

**Please paste the address below in your browser:**

**[bit.ly/schare-analyses](https://bit.ly/schare-analyses)**

# You should see the ScHARe workspace Analyses tab:

The screenshot shows the Terra WORKSPACES interface for the ScHARe workspace. The top navigation bar includes a menu icon, the Terra logo, the text 'WORKSPACES', and the current workspace name 'Workspaces > ScHARe/ScHARe > Analyses'. A 'COVID-19 Data & Tools' badge is in the top right. Below the navigation bar, the 'ANALYSES' tab is selected, with other tabs for 'DASHBOARD', 'DATA', 'WORKFLOWS', and 'JOB HISTORY'. The main content area is titled 'Your Analyses' and features a '+ START' button and a search bar labeled 'Search analyses'. A table lists the analyses:

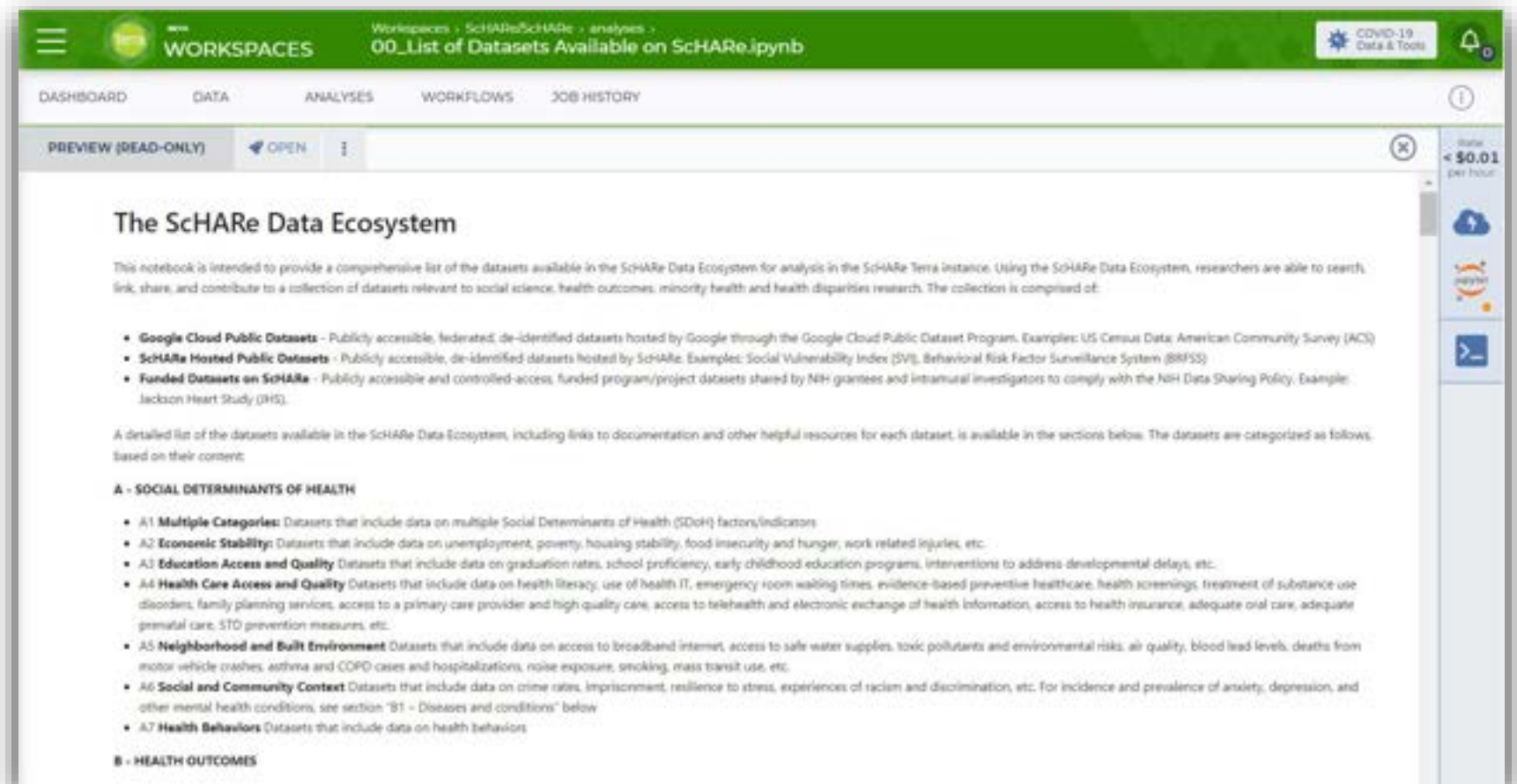
Application	Name ↓	Last Modified
Jupyter	00_List of Datasets Available on ScHARe.ipynb	Today
Jupyter	01_Introduction to Terra Cloud Environment.ipynb	May 10, 2023
Jupyter	02_Introduction to Terra Jupyter Notebooks.ipynb	Jun 23, 2023
Jupyter	03_R Environment setup.ipynb	Apr 7, 2023
Jupyter	04_Python 3 Environment setup.ipynb	Apr 7, 2023

[bit.ly/schare-analyses](https://bit.ly/schare-analyses)

# How to check what data is available on ScHARe

## Analyses tab

In the **Analyses** tab, the notebook **00\_List of Datasets Available on ScHARe** lists all of the datasets available in the ScHARe Datasets collection



The screenshot displays the ScHARe workspace interface. At the top, there is a green header bar with the text "WORKSPACES" and "00\_List of Datasets Available on ScHARe.ipynb". Below the header, there is a navigation menu with tabs for "DASHBOARD", "DATA", "ANALYSES", "WORKFLOWS", and "JOB HISTORY". The "ANALYSES" tab is currently selected. Below the navigation menu, there is a "PREVIEW (READ-ONLY)" button and an "OPEN" button. The main content area shows the title "The ScHARe Data Ecosystem" and a paragraph of text: "This notebook is intended to provide a comprehensive list of the datasets available in the ScHARe Data Ecosystem for analysis in the ScHARe Terra instance. Using the ScHARe Data Ecosystem, researchers are able to search, link, share, and contribute to a collection of datasets relevant to social science, health outcomes, minority health and health disparities research. The collection is comprised of:"

- **Google Cloud Public Datasets** - Publicly accessible, federated, de-identified datasets hosted by Google through the Google Cloud Public Dataset Program. Examples: US Census Data, American Community Survey (ACS)
- **ScHARe Hosted Public Datasets** - Publicly accessible, de-identified datasets hosted by ScHARe. Examples: Social Vulnerability Index (SVI), Behavioral Risk Factor Surveillance System (BRFSS)
- **Funded Datasets on ScHARe** - Publicly accessible and controlled-access, funded program/project datasets shared by NIH grantees and intramural investigators to comply with the NIH Data Sharing Policy. Example: Jackson Heart Study (JHS).

A detailed list of the datasets available in the ScHARe Data Ecosystem, including links to documentation and other helpful resources for each dataset, is available in the sections below. The datasets are categorized as follows, based on their content:

**A - SOCIAL DETERMINANTS OF HEALTH**

- **A1 Multiple Categories:** Datasets that include data on multiple Social Determinants of Health (SDOH) factors/indicators
- **A2 Economic Stability:** Datasets that include data on unemployment, poverty, housing stability, food insecurity and hunger, work related injuries, etc.
- **A3 Education Access and Quality:** Datasets that include data on graduation rates, school proficiency, early childhood education programs, interventions to address developmental delays, etc.
- **A4 Health Care Access and Quality:** Datasets that include data on health literacy, use of health IT, emergency room waiting times, evidence-based preventive healthcare, health screenings, treatment of substance use disorders, family planning services, access to a primary care provider and high quality care, access to telehealth and electronic exchange of health information, access to health insurance, adequate oral care, adequate prenatal care, STD prevention measures, etc.
- **A5 Neighborhood and Built Environment:** Datasets that include data on access to broadband internet, access to safe water supplies, toxic pollutants and environmental risks, air quality, blood lead levels, deaths from motor vehicle crashes, asthma and COPD cases and hospitalizations, noise exposure, smoking, mass transit use, etc.
- **A6 Social and Community Context:** Datasets that include data on crime rates, imprisonment, resilience to stress, experiences of racism and discrimination, etc. For incidence and prevalence of anxiety, depression, and other mental health conditions, see section "B1 - Diseases and conditions" below
- **A7 Health Behaviors:** Datasets that include data on health behaviors

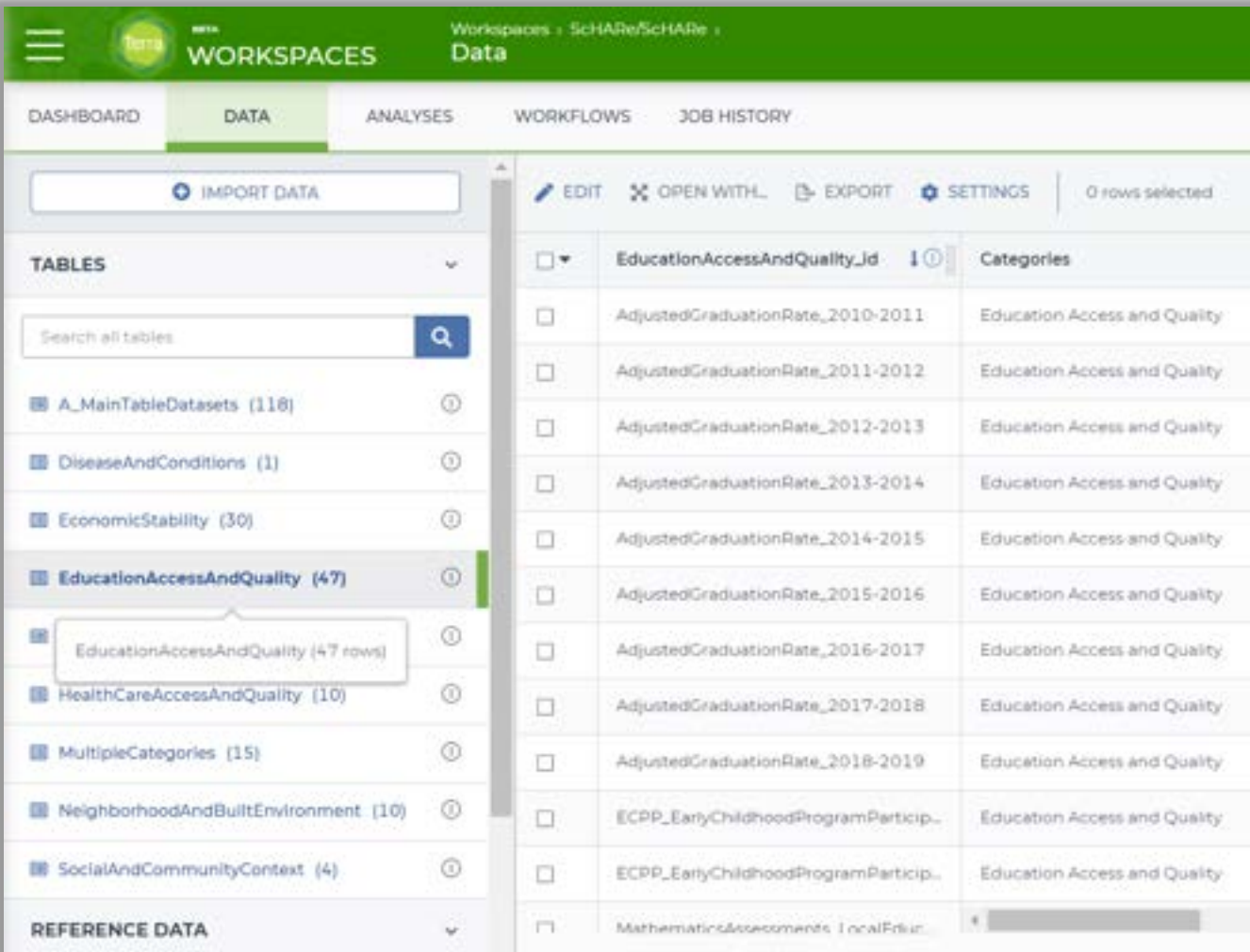
**B - HEALTH OUTCOMES**

# How to access available data on ScHARe

## Data tab

In the **Data** tab, data tables help access ScHARe data and keep track of your project data:

- In the ScHARe workspace, click on the Data tab
- Under Tables, you will see a list of dataset categories
- If you click on a category, you will see a list of relevant datasets
- Scroll to the right to learn more about each dataset



The screenshot displays the ScHARe workspace interface. The top navigation bar includes 'DASHBOARD', 'DATA', 'ANALYSES', 'WORKFLOWS', and 'JOB HISTORY'. The 'DATA' tab is active, showing an 'IMPORT DATA' button and a search bar for tables. A list of dataset categories is shown on the left, with 'EducationAccessAndQuality (47)' selected. On the right, a detailed view of this category is shown, listing various datasets such as 'AdjustedGraduationRate\_2010-2011' and 'AdjustedGraduationRate\_2011-2012', all categorized under 'Education Access and Quality'. The interface also includes options for 'EDIT', 'OPEN WITH...', 'EXPORT', and 'SETTINGS', and indicates '0 rows selected'.

# Today's hands-on tutorial

We will show you how you can access any ScHARe hosted dataset from your workspace:

## Data you upload to your workspace

This is your own personal project data, stored on your computer

## Data already in the ScHARe Data Ecosystem

1. Google Hosted Public Datasets
2. ScHARe Hosted Public Datasets
3. ScHARe Hosted Project Datasets

# We will demonstrate the process with BRFSS data



**BRFSS** is the nation's premier system of health-related telephone surveys that collect state-level data about U.S. residents regarding their:

- health-related **risk behaviors**
- chronic **health conditions**
- use of **preventive services**

State health departments use in-house interviewers or contract with telephone call centers or universities to administer the BRFSS surveys **continuously through the year**

BRFSS data is used to:

- help establish and track state and local health objectives
- plan **health programs**
- implement **disease prevention** and **health promotion activities**
- monitor **public health trends**

## In 3 steps, we will:

1. Create a Terra **workspace**
2. Create an analysis **notebook** inside the workspace
3. Show what code you need to use to **access ScHARe datasets** from the notebook



# 1. Let's create a workspace

Complete slides with **step-by-step instructions and screenshots** available at: [bit.ly/think-a-thons](https://bit.ly/think-a-thons)

Let's create a Terra workspace.

1. **Click on the menu** in the top left corner of the page, then on “**Workspaces**”
2. Click on the **+** button next to Workspaces
3. Input a **name** for the workspace
4. Select the **Billing Project** you want to associate with the workspace. For this example, select our free temporary Billing Project “SchARe-Temp”
5. Change the **Description** if desired

**Success!** The workspace is now listed among your workspaces. Click on it to access it.

## 2. Let's create a notebook

Complete slides with **step-by-step instructions and screenshots** available at: [bit.ly/think-a-thons](https://bit.ly/think-a-thons)

A Jupyter Notebook is an interactive analysis tool that includes:

- **code cells** for manipulating and visualizing data in real time (Terra notebooks support **Python or R**)
- **documentation** to make it easier to share and reproduce your analysis

Let's cover the basics of **creating your first notebook to work with your data.**

Access the workspace and:

1. Click on the **“Analyses”** tab
2. Click on the **“Start”** button
3. Select **“Jupyter”**
4. In the next window, **name** to the notebook and choose a language (**“Python”**)
5. Click **“Create analysis”**
6. If you are asked to configure your **Cloud Environment** (your virtual computer to be used to run the analyses), you can leave the default values unchanged

**Success!** Your notebook has been created. **Click on its name** to open it.

### 3. Let's access ScHARe data

1. Open this tutorial in Playground mode:

**[bit.ly/import-schare-data](https://bit.ly/import-schare-data)**

2. Copy the code and paste it into your notebook.

The code is also reported on this slide, to the right.

**Success!** You have imported data from the ScHARe hosted BRFSS dataset into your notebook.

**Code:**

```
import pandas

BRFSS_table =
pandas.read_csv("gs://fc-
secure-d6e25d73-4b50-4dbc-
ac10-
ec689987eaa9/uploads/Health
Behaviors/BRFSS2021.csv",
sep=',')

BRFSS_table.head()
```



# SCIENCE



# ARE

Thank you

# Think-a-Thon poll

1. Rate how useful this session was:

- Very useful
- Useful
- Somewhat useful
- Not at all useful

# Think-a-Thon poll

2. Rate the pace of the instruction for yourself:

- Too fast
- Adequate for me
- Too slow

# Think-a-Thon poll

3. How likely will you participate in the next Think-a-Thon?

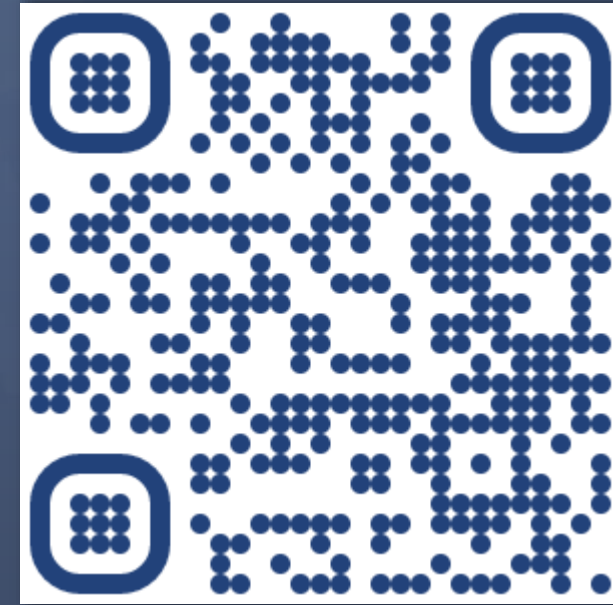
- Very interested, will definitely attend
- Interested, likely will attend
- Interested, but not available
- Not interested in attending any others

**Next Think-a-Thons:**



[bit.ly/think-a-thons](https://bit.ly/think-a-thons)

**Register for SchARe:**



[bit.ly/join-schare](https://bit.ly/join-schare)

 [schare@mail.nih.gov](mailto:schare@mail.nih.gov)