

# ASTRON 3705 / PHYSICS 3704

---

Statistics and Data Science

Spring 2025



# Instructor Information

---

- ❖ Lecturer: Prof. Jeffrey Newman
  - ❖ (I'll answer to Jeff or Jeffrey, Dr. Newman, Professor, Professor Newman... )
- ❖ Office: 310 Allen Hall
- ❖ Email: [jnewman@pitt.edu](mailto:jnewman@pitt.edu)
- ❖ Phone: (412) 592-3853



# Who am I?

- ❖ Joined the Department of Physics & Astronomy in Fall 2007, after graduate and postdoctoral work at UC Berkeley
- ❖ My specialties are galaxy evolution and cosmology: studying how both galaxies and the Universe as a whole formed and have grown over the last 13.7 billion years
- ❖ I primarily do research with the Keck Telescopes, the Hubble Space Telescope, and now DESI
- ❖ I am also heavily involved in planning for the upcoming Rubin Observatory LSST and Roman Space Telescope





# What's the point of this course?

---

- ❖ To learn basic statistical techniques, how to interface with data, what the limitations of real-world data are, and how to work with data in Python.
- ❖ The main goal is to prepare you for modern physics and astrophysics research - whether theoretical or observational.
- ❖ The goal: not everything we do will be useful for everyone, but most of the things we do will be useful to all, and everything we do will be helpful for someone.



# Goals: Statistics

- ❖ At the end of this course, you should be able to:
- 
- ❖ Apply a variety of statistical tests to measurements, and identify the correct test for the problem being faced
  - ❖ Apply Monte Carlo and resampling techniques to predict distributions of errors, estimate significances, etc. even for data with unknown probability distributions
  - ❖ Perform linear and nonlinear curve fitting
  - ❖ Apply maximum likelihood techniques and utilize robust statistics to make measurements
  - ❖ Use propagation of errors to determine uncertainties in derived quantities



# Goals: Datasets

- ❖ At the end of this course, you should be able to:
- 

- ❖ Read in data files from existing datasets, select objects / items of interest, and apply statistical methods to those data



# Goals: Programming

- ❖ At the end of this course, you should be able to:

---

- ❖ Perform data analysis, I/O, and plotting in the Python programming language.

- ❖ Apply a few machine learning techniques for both classification and regression (continuous variable prediction)

- ❖ You should be sufficiently proficient in using Python by the end of the course to use it daily in your research.



# So how will we achieve those goals?

---

- ❖ This course will NOT just consist of me lecturing.
  - ❖ In a wide variety of studies, lectures rarely result in lasting learning.
  - ❖ Lectures longer than about 15 minutes tend to become ineffective.
  - ❖ If I just lecture straight out all semester, we may cover a lot of material, but how much of it will you remember 10 months (or 10 years) later?



# Hybrid lecture-lab

---

- ❖ Instead, we'll mix lecture with in-class Python activities.
  - ❖ For instance, I might talk about some particular statistical technique... then you (in small groups) will use or produce a dataset and apply that technique to it, in Python.
  - ❖ A somewhat more complicated version of what you have done in class (e.g., comparing multiple techniques or an application to real data) might constitute the homework problem for the week.
  - ❖ To make room to cover more in class, some simpler material may be handled by lecture slides to be reviewed out of class or Jupyter notebooks to be completed as homework



# Textbook: *Practical Statistics for Astronomers* by Wall & Jenkins (second edition)

---

- ❖ This provides a good overview of statistical techniques useful to both astronomers and physicists, though with minimal detail. We'll generally make up for that through actual application of the techniques.
  - ❖ The book is available free online for people at Pitt
- ❖ A basic experimental statistics book (like Bevington or Taylor) may occasionally be a useful reference.
- ❖ I can also recommend *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data*, Updated Edition by Željko Ivezić, Andrew J. Connolly, Jacob T. VanderPlas, and Alexander Gray
  - ❖ Pitt has access to only one copy of it



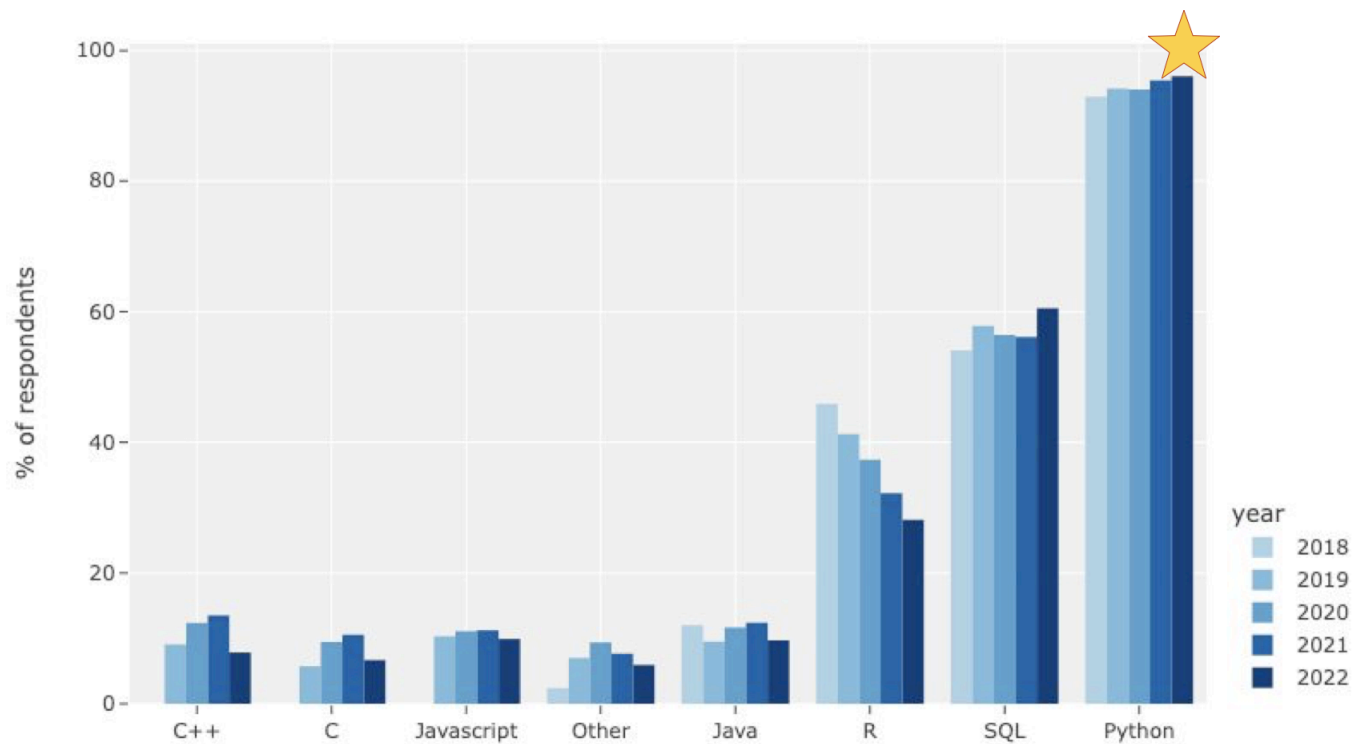
# Why will we use Python?

---

- ❖ Python has several advantages for physics and astrophysics research:
- ❖ There are extensive libraries of statistical and machine learning routines available, as well as field-specific packages (e.g., `astropy` for astronomy)
- ❖ The language is fairly well optimized for image processing and manipulating large datasets, if you use the right techniques
- ❖ Because Python is an interpreted, rather than compiled, language, code is comparatively easy to debug and exploration and manipulation of datasets in real time is straightforward.



# Python has become the clearly dominant language for data science



Kaggle,  
2023



# Notes on languages...

---

- ❖ I would be fine with students using other languages for homework or projects, but in-class activities will use Python by necessity.
- ❖ Does anyone not have their own laptop? Who uses Mac/Linux/Windows?



# Components of grades

---

- ❖ I expect grading to be based on:
  - ❖ 30% Homework
  - ❖ 20% In-class activities
  - ❖ 10% Astro coffee participation or alternative assignment
  - ❖ 40% Final project



# Homework:

---

- ❖ Every week or two there may be 1-2 small homework problems/tasks.
- ❖ Students are allowed (and encouraged) to collaborate on homework assignments in developing basic algorithms, but must present their own work (programs/plots, results, etc.).
- ❖ Please provide all relevant code and notes describing how you did a calculation, to help me understand the cause of any errors and give you feedback.
- ❖ I do NOT expect a detailed writeup of Python-based problems (e.g., it is not necessary to spend time on nicely formatted equations) unless I specify otherwise.



# In-class activities:

---

- ❖ In-class notebooks will often yield some work product (e.g. a completed notebook) which will contribute to your in-class activities grade (as completion/noncompletion, not graded in detail).
- ❖ If you do need more time, finishing after class will be fine. In-class activities are a critical part of the course, so keeping up with them is important.
- ❖ **I will collect your jupyter notebook files for grading at the end of the semester if not before. Keep them in a standard place to make this easy.**



# Astro coffee participation / alternative assignment

---

- ❖ For astrophysics graduate students, 10% of your grade is associated with attending and presenting papers at the twice-weekly "astro coffee" journal club
  - ❖ I expect students to present at least two papers over the course of the semester
  - ❖ Grading: zero papers = F, 1 paper = C, 2 papers = A
- ❖ As an alternative to this, non-astro students can produce an [astrobites.org](http://astrobites.org) - style, 2-3 page summary, written for nonexperts, of a paper in your field that applies some statistical or machine learning method.



# Final Project:

---

- ❖ The capstone of the course will be an original project – a new measurement, development of a new data reduction algorithm, a new analysis of an existing dataset, etc. You should generally work on this project in groups of two. Some previous projects have turned into published papers.
- ❖ I will provide a list of some project ideas mid-way through the semester, which you may sign up for, or I am happy to discuss your own project ideas (often this is a joint discussion with a research advisor).
- ❖ Projects should not simply consist of the research activities you are already doing, except in the case of undergrads (but applying a new statistical technique to solve problems in your research would be fine).



# Final Project:

---

- ❖ At the end of the semester, you will give a ~20 minute presentation of your project to the class.
- ❖ You will also need to provide a brief writeup describing what you have done (ideally 2-3 pages; **maximum 5 pages with 12-point or larger font in manuscript -- not two-column -- format** ) to allow both oral and written presentation skills to be evaluated.
- ❖ Communication of results is a key skill for all scientists. Your project will primarily be graded on content/work, but the presentation & writeup will make up a portion of the score.



# Calendar

---

- ❖ This class will meet every Monday and Wednesday of the semester, 12:15 PM - 1:30 PM, except MLK's Birthday on Monday, Jan. 20
- ❖ Spring Break is March 3-7
- ❖ Conceivably we might need to swap a class to a Friday occasionally
- ❖ The class will meet during exam week for final project presentations



# Office Hours

---

- ❖ How does Monday 2:30-3:30 PM work for everyone for scheduled office hours?
  - ❖ Different days?
  - ❖ Different times?
- ❖ I am also available by appointment (especially after 9 PM), typically on Zoom.



# A lesson from Sandra Faber

---

- ❖ You should come to class ready to ask questions about any parts of the reading that are unclear!
- ❖ If you don't understand something, other students probably don't either.
- ❖ Asking questions, not just answering them, can help you when final grades are assigned.





# Canvas and Slack

---

- ❖ A Canvas site for this class has been set up.
- ❖ - All handouts, lecture slides, etc. will be posted to this page. I will generally use Slack for quick communications, and Canvas for persistent announcements.
- ❖ Email me if you are not on the astro Slack and I can send you an invite (I tried to do this already - you may need to check your email)
- ❖ Let's go to Canvas now: [canvas.pitt.edu](https://canvas.pitt.edu)



# Where we're heading...

---

- ❖ Today: Introduction to statistics
  - ❖ By Friday: install Python (instructions upcoming). I should be on campus Friday ~2-5 PM to help with any installations that aren't working. **Everyone should have Python working before Monday's class!**
- ❖ Monday: Some basic Python review, some statistics. Read Chs. 1 & 2.1 of Practical Statistics for Astronomers before Monday.
- ❖ Short term: Basic statistics & Python
- ❖ Medium term: Working with data
- ❖ Long term: Applying statistics and machine learning methods to large datasets



# First, let's introduce ourselves

---

- ❖ What is your name?
- ❖ What area are you doing research in (or planning to)?
- ❖ Have you used Python before?
- ❖ How many undergraduate or graduate astro courses have you taken, if any? How many graduate physics classes?



Let's start talking about statistics...

---



# Why the focus on statistics?

---

- ❖ Pretty much every scientist needs to employ statistical analysis at one point or another.
- ❖ Observers / Experimentalists:
  - ❖ Is my data consistent with results in the literature? What scenarios/hypotheses can I prove or disprove?
  - ❖ What are the “error bars” on my measurement? What do they mean?
  - ❖ How much bigger samples / more exposure time would I need to rule something in or out?
  - ❖ Are there any systematic problems in my data?
  - ❖ What is the optimal technique to use to make a particular measurement?



# Why the focus on statistics?

---

- ❖ Theorists:

- ❖ Is the data from the literature consistent with (or does it rule out) my model?
- ❖ What sort of experiment would be needed to test my model / measure parameter  $X$  to accuracy  $Y$ ?
- ❖ What do error bars from the literature mean? What do I need to watch out for?



# Notice those are all questions...

---

- ❖ Statistical inference (a.k.a. "statistics"): the mathematical field devoted to determining what we can decide (or infer) given a specific set of observed data - it tells us how (and even whether) we can answer those questions.
- ❖ If science is all about trying to figure out what rules may apply in the natural world, statistics tells us how to judge whether those rules are consistent with experiments / observations or not.
- ❖ Formally, a *statistic* is a quantity calculated from data that summarizes its properties in some way (like an average).
- ❖ To usefully apply machine learning methods, we will still need to have a basic understanding of statistics



# Statistics in this course

---

- ❖ We will focus on practical applications of statistics in this course, not theory.
- ❖ Very rarely will we only be doing statistics on a given day - instead, we'll be learning Python while implementing statistical tests, or learning to make use of large datasets by applying statistics to them.
- ❖ Today, we'll take a first stab at defining 'probability' and prepare for doing simulations of random datasets on Monday.



# Probability

---

- ❖ Rarely do we obtain an absolute result with no uncertainty.
- ❖ Data tells us about what truths may be possible - e.g. “given our data, it is 95% likely that this star's true magnitude is between 19.9 and 20.1”.
- ❖ As a result of an experiment/ observation, we might use some statistic (like the average or sum) to assess the *probability* that a quantity we are interested in actually has a certain value compared to others.
- ❖ We might express this as a *probability density function* - which gives the **relative** probability of all possibilities, normalized to have integral 1.
  - ❖ For a discrete quantity, we instead use the *probability mass function* that has sum 1.



# An emphasis of this class: *Non-parametric statistics*

---

- ❖ Defining probabilities of something based on our observations often requires accurate knowledge / assumptions about how far off a measurement could be: i.e., knowing what the underlying probability distribution is.
- ❖ In practice, it is typically much harder to determine uncertainties well (e.g., the full range of values that could occur 68%, 95%, etc. of the time) than to determine what the most probable value is.
- ❖ As a result, *non-parametric* methods - ones that **don't** depend on knowing / assuming the form of the actual distribution of probabilities - can be particularly valuable.



# Non-parametric statistics

---

- ❖ For instance, we can assess whether two datasets could be drawn from the same probability density function (e.g., the same underlying process), even without knowing what those functions are!
- ❖ Non-parametric methods can be particularly valuable when we have little data (so it's hard to tell by eye what the distribution function could be) or when we compare observations that aren't on numerical scales.



# So what do we mean by “probability”?

---

- ❖ Probability theory arose out of the analysis of gambling games.
- ❖ In such a well-controlled situation, we can define the probability of an event as the fraction of times it will occur if we infinitely repeat an experiment - i.e., its frequency.

$$P(x) = \lim_{n \rightarrow \infty} \frac{n_x}{n_t}$$

- ❖ This is the oldest definition of probability - the “frequentist” view - but not the only one possible (more on that later). Note P is at most 1.



# An example

---

- ❖ Consider flipping two coins. The possible things that could happen are:
  - ❖ - 1st coin heads, second heads
  - ❖ - 1st coin heads, second tails
  - ❖ - 1st coin tails, second tails
  - ❖ - 1st coin tails, second heads
- ❖ If heads and tails are equally likely, then each of these possibilities is equally likely, so we'd expect each of these possibilities to occur  $1/4$  of the time.
- ❖ So the probability that both coins give the same result is  $1/4 + 1/4 = 1/2$ ; if we flip coins an infinite number of times, half the time we'd get this result.



# Let's test this.

---

- ❖ We can't flip coins an infinite number of times - or even a million - in the course of an hour.
- ❖ However, we can emulate that process in a computer.
- ❖ A simulation where we randomly generate data in some way is generally referred to as a *Monte Carlo* simulation.
- ❖ They are an excellent way to test statistical methods - or to interpret what is going on in your data.



# Getting ready to install Python

---

- ❖ Who hasn't used UNIX before?
- ❖ Who hasn't used Python before?
- ❖ Has anyone not installed miniforge python yet?



# Preparing to use Python

---

- ❖ Previously, I used Anaconda for this class...
  - ❖ No longer being treated as free for university use
- ❖ Instead we will use:
  - ❖ miniforge : to install python and manage packages
  - ❖ Visual Studio Code (vscode): as a programming environment



# Preparing to use Python

---

## 1) Install miniforge

- ❖ Go to <https://github.com/conda-forge/miniforge> to find instructions for your OS to download and install
- ❖ Follow those instructions!



# Preparing to use Python

---

- 2) Set up a new python environment to work in for the class
  - ❖ We will use the mamba package manager for this; see [https://mamba.readthedocs.io/en/latest/user\\_guide/mamba.html#mamba](https://mamba.readthedocs.io/en/latest/user_guide/mamba.html#mamba) for details
  - ❖ mamba is like conda (used by Anaconda), but much faster



# Preparing to use Python

---

- ❖ In a terminal window, type:

```
mamba init
```

- ❖ Then start a new shell / terminal. In that terminal, type:

```
mamba create --name NAME_OF_YOUR_ENVIRONMENT
```

```
mamba activate NAME_OF_YOUR_ENVIRONMENT
```

```
mamba install datascience jupyter matplotlib astropy scikit-learn
```

- ❖ This switches to the new environment and installs the listed packages
- ❖ `NAME_OF_YOUR_ENVIRONMENT` can be whatever string name you want to call the environment; e.g., it could be `datascience` for this class



# Preparing to use Python

---

- ❖ If you want Python to use this environment in new terminals, you need to add

`mamba activate NAME_OF_YOUR_ENVIRONMENT`

to the end of your `.bash_profile` / `.zshenv` file.

- ❖ Otherwise you will be returned to the `base` environment in new terminals



# Preparing to use Python

---

## 3) Download and install Visual Studio Code

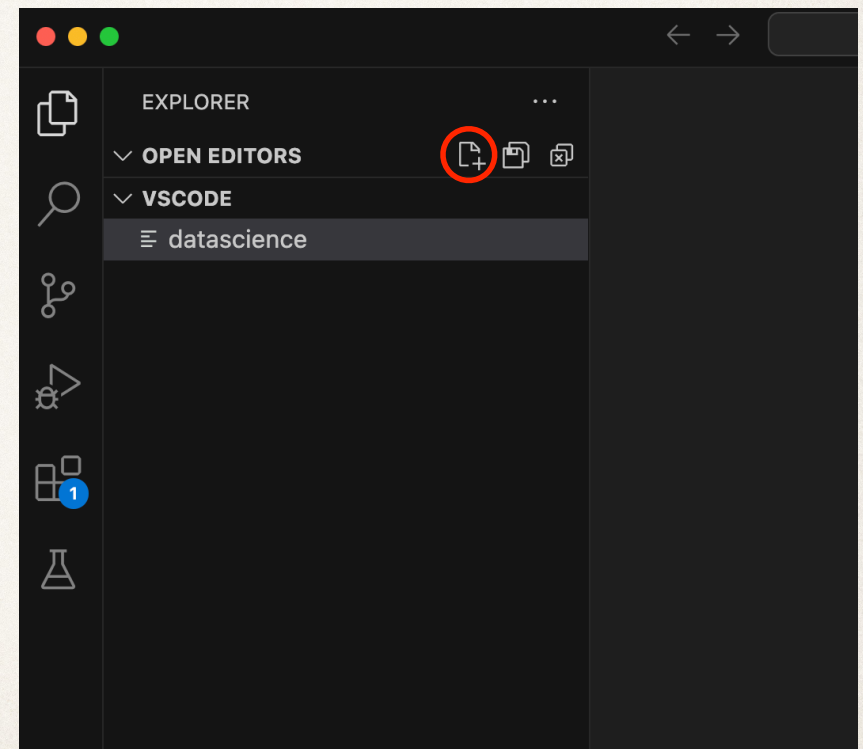
- ❖ You can get it at <https://code.visualstudio.com/>
- ❖ Then download and install the vscode Python extension from <https://marketplace.visualstudio.com/items?itemName=ms-python.python>



# Preparing to use Python

4) Start up vscode and test it

- ❖ Near the top left corner, use the indicated icon to open a new text file

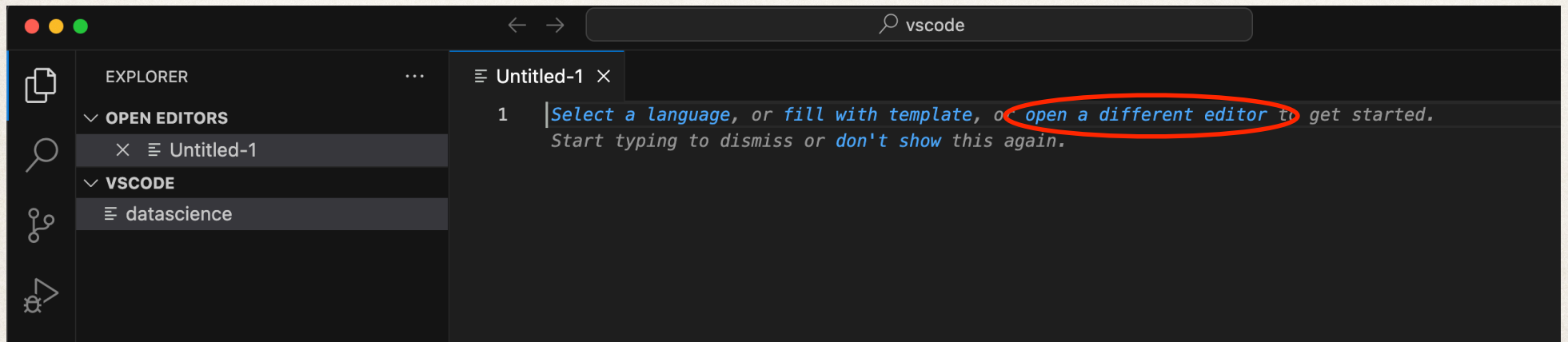




# Preparing to use Python

## 4) Start up vscode and test it

- ❖ In the new window, you can use 'select a language' or (easier) 'open a different editor' to select Python

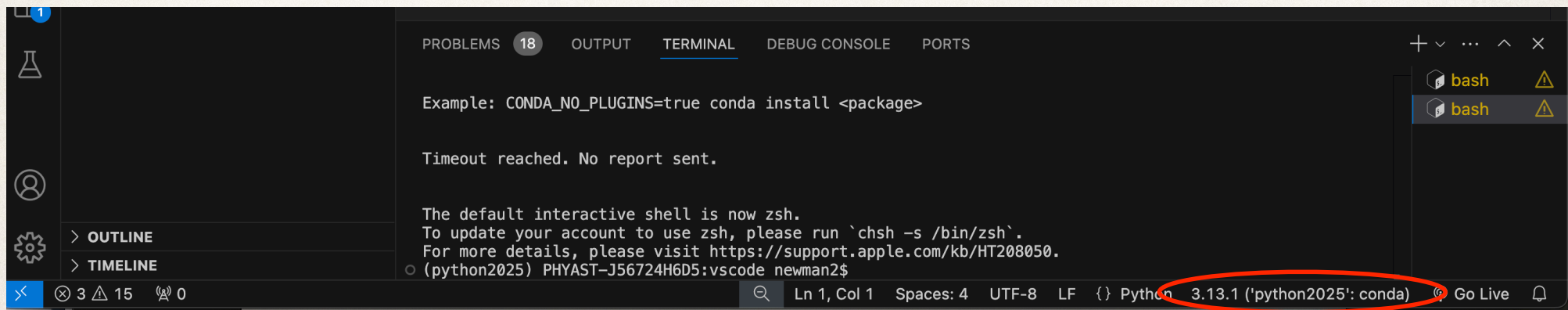




# Preparing to use Python

## 4) Start up vscode and test it

- ❖ Be sure to set vscode to use your new virtual environment!
- ❖ Your current python environment is listed near the bottom right; click on that to choose the right one (which will become default)





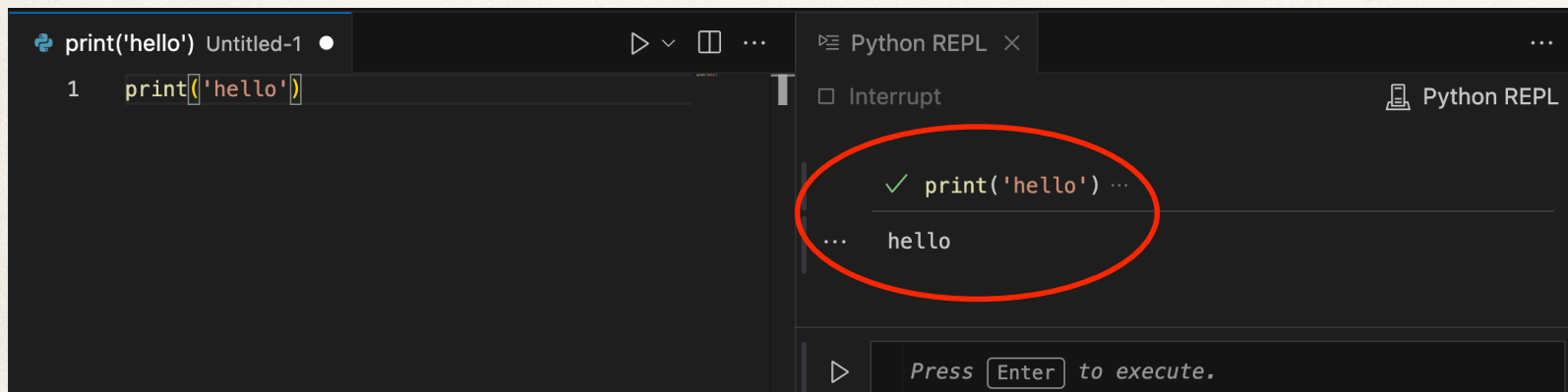
# Preparing to use Python

---

- ❖ Now, let's see if things are working
- ❖ In the editor, type

```
print('hello')
```

and press shift-enter (i.e., hold down shift and press enter). The results hopefully look something like this...



The screenshot shows a Python REPL window with a dark theme. On the left, a code editor shows the line `1 print('hello')`. On the right, the REPL output shows a green checkmark followed by `print('hello') ...` and then `... hello`. A red oval highlights the output section. At the bottom of the REPL window, there is a prompt that says "Press Enter to execute."



# Preparing to use Python

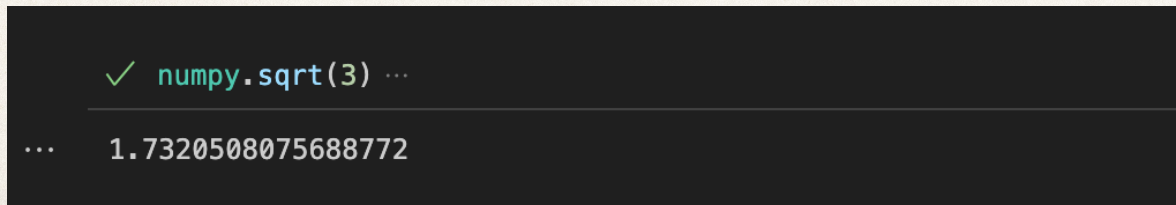
---

- ❖ Next, In the editor, type

```
import numpy  
numpy.sqrt(3)
```

and press shift-enter after each line. The results hopefully look something like this... if so you should be ready for Monday!

- ❖ The results show up in a 'Python REPL' window - that stands for Read, Evaluate, Print, Loop : basically testing code one line at a time



```
✓ numpy.sqrt(3) ...  
... 1.732050807568772
```