



Statistics 21

Python & Other Technologies for Data Science

Vivian Lew, PhD - Monday, Week 1

## Highlights from the Syllabus

(adapted from material originally created by Dr. Miles Chen)

Welcome!

- My name is Vivian Lew
- Please call me whatever name you are comfortable using

- But I don't like being called Mrs. Lew
- This is the real Mrs. Lew... 😊



## Office Hours Schedule

It is always great to see you during office hours.

- Mondays, Wednesdays, Fridays 4:50pm here until we're done (in-person only)
- Mondays 1:00PM - 3:00PM in my office - Math Sciences 8923

**please schedule appointments with me if you cannot meet at any of these times**

**please schedule your initial 10% meeting with me using the shared spreadsheet**

## Note about office hours:

- Office hours are a nice way to meet and answer questions.
- Questions and issues are generally resolved much more quickly via office hours.
- When you come to office hours, please **introduce yourself**. For example "Hi, I'm Joe Bruin." Please do this until I learn your name.
- I like office hours so much that everyone is going to need to meet with me at least once to earn 10% of their final grade.
- Office hours are a place to tell me more about yourself, maybe seek counsel about classes, maybe anything. I am happy to make accommodations for students who face difficult circumstances and may need extensions for assignment deadlines. Please do not hesitate to visit office hours.
- It's fine to ask me to correct grading mistakes or to ask about grading decisions during office hours but don't talk about grades in front of others (I can ask people to leave or schedule a private appointment)

## Grade Breakdown

- 10% A 5 (min) - 15 minute (max) meeting with your professor before Week 4 (in-person preferred, Zoom is OK)
- 20% Attending and participating in team activities on Mondays and Wednesdays during Lecture
- 10% Attending and participating during section (TA will determine the specifics)
- 10% Mid-quarter Exam (end of Week 4 or start of Week 5, TBD)
- 10% Programming Homework (10 assignments - drop your two lowest scores)
- 10% Reflective Thinking (10 exercises - graded complete/incomplete - no drops)
- 20% Final Team Mini Project & Presentation Video (less than 10 minutes due Finals Week)
- 10% Either attending and participating in team activities in lecture on Fridays **OR (not and)** Final 15 minute (max) exam modeled on a technical job interview with your professor (in-person only, no notes, no electronic devices allowed and no internet access, schedule it anytime during Finals week)

## Grading

Letter grades are assigned on a modified straight scale as follows:

- 50 and below: F
- 51.0 - 76.9: C, 77.0 - 79.9: C+
- 80.0 - 82.9: B-, 83.0 - 86.9: B, 87.0 - 89.9: B+
- 90.0 - 92.9: A-, 93.0 and up A

## Attendance

- I do not take traditional attendance during lecture, instead there is an in-class activity involving classmates for credit.
- If you are unable to attend during the in-person lecture there will a weekly recap recorded. Lectures are not broadcasted live to encourage discussion/interaction.
- Monday of 9th week is a holiday (no lecture) and I allow 3 missed lectures from the MW portion before deductions begin
- There will be no lecture on Friday of 8th Week, and if you miss up to 3 of the remaining Friday lectures, you can take part of the final exam to make up for the deduction.

## Homework

- Homework will be posted as an Jupyter Notebook ( .ipynb file).
  - Students will complete their exercises in the notebook ( .ipynb file),
  - Then save the notebook as a PDF and upload it.
  - And also upload the .ipynb file.
- Please incorporate this as homework late policy in the syllabus.
  - Late is declared once the grading begins
  - A 72-hour extension granted if documentation (e.g., doctor's note) is submitted with the homework .ipynb file.
  - **No need to contact the professor if you can include documentation with the homework submission.**
  - If you need an even longer extension, please visit professor in office hours or submit an e-mail request

BruinLearn Discussion for Python questions. Office hours for individual matters.

- Please post your Python questions on the discussion page on BruinLearn. You will likely get a quick response from classmates or me.
- If it is a question you don't want public, you should e-mail me via BruinLearn, it shows up as a notification on my phone.
- If you need talk about private/personal matters (e.g., anything related to grades or health) or just want to talk about things non-Python too - please come to office hours - don't email/post private things unless it's an emergency

## Course Outline

(subject to change)

- Week 1: Prep: Shell, Python, PyCharm, Jupyter, pip, Git, ChatGPT
- Week 2: Python Fundamentals: data types, data structures.
- Week 3: Flow Control, Functions, Problem Solving, Algorithms
- Week 4: More Data: strings, lists, dictionaries, tuples
- Week 5: Midquarter Exam, Modules and Libraries, Pandas intro
- Week 6: Pandas: Wrangling and Analysis
- Week 7: Numpy and SciPy
- Week 8: Data Visualization: Matplotlib, Seaborn, Plotly
- Week 9: Python OOP: Classes and objects, methods, inheritance
- Week 10: Work on Final Team Project and TBD
- Finals Week: Final Exit Interview (if necessary), turn in all remaining work

## Academic Integrity and Plagiarism

## ChatGPT is a game changer

- Ask ChatGPT any Stats 21 level programming question
- Example from last year
- Let's talk about it. First, you should be using it. Second...

Think about 10 years, not 10 weeks

- There are skills you need to develop
- Problem solving skills - analyze complex problems, break the problem down into smaller components, and identify the most effective solution
- Critical thinking skills: analyzing and evaluating information to make wise decisions and solve problems by possessing the ability to identify fallacies, biases, credibility, correctness
- Communication skills- you will need to communicate with others: classmates, colleagues, supervisors, stakeholders, and clients. both in writing and verbally with the initial goal of articulating your ideas and thoughts clearly, but also to motivate, encourage, inspire...
- Flexibility - able to adapt to change and develop alternative solutions

## Your Classmate is not the enemy

- The course is straight-scale 90% and higher is some form of "A" (includes A-)
- Part of your final grade is a team project to help you to develop your skills in communication
- Also work on leadership, teaching ability, professionalism - the things that will help carry you further in the future
- Your classmates are your "Data Theory Cohort", they will exit with the same degree as you -- you want employers and graduate programs to think positively about UCLA's Data Theory major

## Collaboration Policy

- For all assignments, verbal collaboration is strongly encouraged but no code sharing.
- You are allowed to collaborate verbally with other students and ChatGPT but you are not allowed to look at or show someone else the code you are writing. They need to learn how to code and develop their own abilities and problem solving processes.

## Allowed vs Not allowed

- Question to a classmate: "How did you do problem 2?"
- Policy Conforming response: "I created a for loop and within each iteration I subset the x vector to the desired values and then used the sort function on the result. Be sure to assign the results to update the output object." "Thank you! I'll be sure to note your help."
- Policy Violating response: "I can e-mail code to you"

## Allowed on ChatGPT

- "Can you help me find what I'm doing wrong? Here is my code..."
- Please help me to understand the difference between an attribute and a method in Python
- Please show me how to write a nested for loop.

## Beware of doing this on ChatGPT

- This was a homework problem from last year which I copied and pasted to ChatGPT to answer for me this year: please write a function called `mul_time` (multiply time) that takes a Time object and a number and returns a new Time object that contains the product of the original Time and the number (next slide)
- If a classmate submits the exact same answer as you (so you both copied and pasted the same question to ChatGPT and got the same answer from ChatGPT), your assignments will be flagged on BruinLearn for Academic Dishonesty and I am required to report it.
- It is risky to copy and paste for a number of reasons.





Sure, here's a Python function called `mul\_time` that takes a `Time` object and a number like and returns a new `Time` object that contains the product of the original `Time` and the number:

python

Copy code

```
def mul_time(time, num):
    """
    Multiply a Time object by a number and return the result as a new Time object.

    time: a Time object
    num: a number to multiply the Time by

    returns: a new Time object that is the product of the original Time and
    """
    seconds = time_to_int(time) * num
    return int_to_time(seconds)
```

This function first converts the `Time` object to an integer representing the number of seconds, multiplies it by the given number, and then converts the result back into a new `Time` object using the `int\_to\_time` function from an earlier example. Note that this function assumes the `Time` object and the number are both positive, and it does not modify the original `Time` object.

You are encouraged to discuss code that is not part of an assignment!

- Yes This is a coding class! As long as the code is not part of a homework assignment, you can post and discuss code.
- You can always post and discuss code that appears in lecture. You are encouraged to modify the examples that appear in lecture and consider the effect of each change you make.
- You can post and discuss code that is for the purpose of learning a particular concept or how a function works.

Time to Change Gears

## Time for the in-class team response

- If you haven't done so already, negotiate with your neighbors and make teams for today.
- Maximum team size is 4. Minimum is 3. Anything else does not receive credit.
- Visit the people page on the left hand menu of the BruinLearn course page
- Please join a group, any group, first come, first served
- Introduce yourselves, meet someone new, learn a name
- Take a team selfie/greelfie/usie and upload it (just one group member needs to upload) for credit

Team Selfie/Greelfie/Usie Examples from Fall 2021



## CHECK: Does everyone have a copy of Python?

- Python is a little messy. There are different versions of Python (we will use Python 3.10 but 3.9-3.11 are fine).
- Teams - check and see - if you have your own computer with you, start up Python and see which version you have

## Installing Python

- You can find Python at [python.org](http://python.org) (links available)
- You can also run Python online (more on this later)

## Shell Basics

### Opening Terminal on MacOS

- Probably the quickest: Open spotlight with Command + space. Start typing "Terminal". Terminal will appear as the top hit after you type the first few letters. Hit Enter to start.
- Another method: You can open Launchpad from the dock. Click "Other". Click "Terminal".

# terminal — Terminal



## Terminal

Docker Quickstart Terminal

terminal application



terminal

## Definition

Terminal — Apple Dictionary • An Apple app that provides a command line to enter commands for macO...

## PDF Documents

Chapter\_3 — 615 KB • PDF document • Last Opened 2/17/22, 9:32 PM

Combinatorics.pdf — 1.5 MB • PDF document • Last Opened 2/16/22, 11:26 PM

1-2\_Git\_basics.pdf — 165 KB • PDF document • Last Opened 2/17/22, 9:32 PM

## Documents

nuforc\_reports.csv — 121.6 MB • CSV Document • Last Opened 1/31/22, 3:31 PM



vivian — zsh — 80x24

Last login: Sat Mar 11 20:10:00 on ttys001  
vivian@Vivians-MacBook-Pro-3 ~ % █



vivian — Python — 80×24

```
Last login: Sat Mar 11 20:10:00 on ttys001
[vivian@Vivians-MacBook-Pro-3 ~ % python
Python 3.10.5 (v3.10.5:f377153967, Jun  6 2022, 12:36:10) [Clang 13.0.0 (clang-1
300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> ]
```



vivian — Python — 80×24

```
Last login: Sat Mar 11 20:10:00 on ttys001
[vivian@Vivians-MacBook-Pro-3 ~ % python
Python 3.10.5 (v3.10.5:f377153967, Jun  6 2022, 12:36:10) [Clang 13.0.0 (clang-1
300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> quit()
```

## Opening PowerShell on Windows

- Use the Start Menu to find "Powershell".
- You can find it quickly by hitting the Windows Key and then immediately typing powershell then enter
- If you have Windows 11, I'd recommend using Windows Terminal instead
- If I have any Windows experts in this class, please let me know if you are able to help your classmates

## Getting Help

- Windows PowerShell

```
help commandname
```

- Mac OS and git bash

```
commandname --help
```

Type `q` to exit help. Hit the space bar to scroll to the next page.

## Using the command line

shell/terminal/console

We typically interact with computers through the use of GUIs (Graphical User Interfaces)

But there are times when it is better to interact directly and bypass these windows

- quicker/more efficient task completion
- easier/more efficient task automation
- lower overhead lower resource usage for the same task

Your shell skills are a good foundation for more advanced tasks and programming.

## Shell Basics: Navigation

- `pwd` will tell you where you are currently located. (present working directory)
- `cd` is the command to change your directory

Wherever you are, you can switch to your home directory with `cd`

## Shell Basics: Navigation

- Directories are listed in a hierarchy. For example, you may decide to store content for this class in:

Desktop/classes/stats21

- Let's assume this is your present working directory (the response when `pwd` is typed at the prompt).  
`cd homework` will change to the directory `homework` **IF** it exists in your current directory. If there is a `homework` folder inside the `stats21` folder, it will take you to

Desktop/classes/stats21/homework

`cd ..` will take you to the parent directory. If you are currently in `homework`, `cd ..` will take you to

Desktop/classes/stats21

`cd ../../..` will take you two levels up.

- Shell support tab completion. If you have the folder `homework` inside `stats21`, you can begin by typing `cd ho` and then hit TAB. Shell will try to complete what you are typing. If there are multiple items that start with `ho` then you can hit TAB multiple times until it finds the item you are looking for.

## Shell Basics

`ls` will list the contents of your current directory.

`mkdir name` will create a new directory called `name` inside your current working directory.

`clear` will clear the screen.

## *pip (Pip Installs Packages or Pip Installs Python maybe preferred installer program)*

- I've had more success with pip than conda for installing packages
- Check that you have it

```
python -m pip --version
```

- Installation is done with

```
python -m pip install packagenames (note: Mac users might need to preface this with sudo)
```

- Updating can be done with

```
python -m pip install --upgrade packagenames
```

- Check what you have

```
python -m pip list
```

## PyCharm

(<https://www.jetbrains.com/pycharm/>)

- is an IDE (integrated development environment) for Python development
- we can use the free community edition
- It has features and tools like code highlighting, code completion, and error checking.
- makes it easier to find/fix mistakes - step through code line by line, inspect variables, and set breakpoints (designated pause/stop)

PyCharm File Edit View Navigate Code Refactor Run Tools VCS Window Help

36 MB Free Sun Mar 12 12:06 AM

Day1 - Day\_1.py

Day1 > Day\_1.py

Project Day1 ~/PycharmProjects/Day1 venv Day\_1.py External Libraries Scratches and Consoles

```
10 # if we wrote it as 2. it becomes float
11 print(2 + 2.)
12
13 # Note the quotes. This is string
14 print("Hello UCLA")
15
16 # Strings behave differently than numbers and so...
17
18 print("Stats " + "21")
19 print("Data " * 3)
20
21 print(type(2))
22 print(type(2.0))
23 print(type("Hello UCLA"))
24
```

Terminal: Local (venv) vivian@Vivians-MacBook-Pro-3 Day1 %

Version Control Python Packages TODO Python Console Problems Terminal Services

24:1 LF UTF-8 4 spaces Python 3.10 (Day1)

The screenshot shows the PyCharm IDE interface. The top menu bar includes PyCharm, File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The status bar at the bottom right shows system information like battery level (36 MB Free), date (Sun Mar 12), and time (12:06 AM). The main window has a dark theme. On the left is the Project tool window showing a project named 'Day1' with subfolders 'Day1' (~/PycharmProjects/Day1), 'venv', and files 'Day\_1.py', 'External Libraries', and 'Scratches and Consoles'. The central editor window displays the code for 'Day\_1.py'. The code uses print statements to demonstrate string concatenation ('+' operator), string repetition ('\*' operator), and type conversion (print(type(2)), print(type(2.0)), print(type("Hello UCLA"))). The code also includes a multi-line comment explaining the behavior of strings compared to numbers. Below the editor is a terminal window showing a local session with the prompt '(venv) vivian@Vivians-MacBook-Pro-3 Day1 %'. At the bottom of the interface are tabs for Version Control, Python Packages, TODO, Python Console, Problems, Terminal, and Services. The status bar at the very bottom provides file statistics (24:1 LF, UTF-8, 4 spaces) and the Python version (Python 3.10 (Day1)).

## JupyterLab

<https://jupyter.org/install>

- I will discuss this further in the coming days. But if you just want to start playing with Jupyter Lab from the terminal prompt.

```
pip install jupyterlab  
jupyter lab
```

localhost:8888/lab

File Edit View Run Kernel Tabs Settings Help

HW3\_4\_Notebook\_firstname.ipynb Week01\_01\_ChatGPT.ipynb Week\_1-1.ipynb

Python 3 (ipykernel)

**JupyterLab (<https://jupyter.org/install>)**

- I will discuss this further in the coming days. But if you just want to start playing with Jupyter Lab from the terminal prompt.

```
pip install jupyterlab
```

```
jupyter lab
```

**Google Colab (<https://colab.research.google.com/>)**

- Google Colab (short for "collaboratory") is a cloud-based development environment that provides a Jupyter Notebook interface.
- Colab allows you to run and execute Jupyter Notebooks on Google's cloud servers,
- Benefit allows you to perform high-performance computations for machine learning and other data-intensive tasks.

**See you next time**

```
[5]: from IPython.display import IFrame
IFrame(src="https://www.youtube.com/embed/hlqdW24ZGS0",
       width="560", height="315")
```

[5]:

```
<script>
    setBackgroundImage("Window1.jpg", "black");
```

Simple 0 3 Python 3 (ipykernel) | Idle Mode: Command Ln 2, Col 72 Week\_1-1.ipynb

## Google Colab

(<https://colab.research.google.com/>)

- Google Colab (short for "collaboratory") is a cloud-based development environment that provides a Jupyter Notebook interface.
- Colab allows you to run and execute Jupyter Notebooks on Google's cloud servers,
- Benefit allows you to perform high-performance computations for machine learning and other data-intensive tasks.

← → ⌂ colab.research.google.com

File Edit View Insert Runtime Tools Help

Share    Update 

Table of contents   + Code + Text  Connect 

Getting started  
Data science  
 Machine learning  
 More Resources  
Featured examples  
 Section

Welcome To Colaboratory

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.



What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

See you next time

Sleep Token - The Summoning Ending

