

Statistics 21

Python & Other Technologies for Data Science

Vivian Lew, PhD - Monday, Week 1

# Highlights from the Syllabus

(adapted from material 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 would be nice to see you during office hours. My office is 8923 Math Sciences.

- Mondays 5:00 - we're done (in-person only)
- Tuesdays 10:30 - 12:30AM (Zoom only)
- Wednesdays 5:00 - we're done (in-person only)
- Fridays 5:00 - we're done (in-person only)

**please schedule appointments if you cannot attend any of these**

## Grade Breakdown

- 10% Attending Section (see TA for details)
- 15% Attending Lecture (with in class activity) OR viewing recorded version (with quiz)
- 20% Homework
- 20% Labs
- 25% Final Exam (take home)
- 10% Campuswire Participation

## Grading

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

- 49.9 and below: F
- 50.0 - 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

## Attending Lecture (with activity) or Viewing Recorded Lecture (with quizzes)

- 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 lecture you can watch the recorded lecture video and submit answers to the viewing quiz.
- Quiz answers are given audibly during the recording. Answers to the quiz are not found in the lecture notes. Sharing quiz answers with others is considered facilitating academic dishonesty.
- The opportunity to complete quizzes for each lecture will close before the next lecture begins. Three lowest lecture quiz scores will be dropped. If for some reason you miss more than three lectures, please make an appointment with me.

## Homework and Lab Assignments

*I will assign homework and labs that need to be submitted on Canvas AKA BruinLearn. Labs are homework assignments with multiple concepts to help prepare you for the (take home) final exam*

- Homework will be posted as an Jupyter Notebook ( .ipynb file).
  - Students will complete their exercises in the notebook,
  - Then save the notebook as a PDF
  - Then submit the PDF to Gradescope.
  - Also submit the ipynb file to Canvas.
- Please read the late policy in the syllabus.
  - A 72-hour extension granted if documentation is submitted with the homework.
  - **No need to contact the professor if you will 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

## Personal note about office hours:

- Office hours are my preferred method of contact.
- 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 when students come to office hours to tell me more about themselves and to seek counsel about classes to take or next steps. 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.

Campuswire for Python questions. Office hours for individual matters.

- Please post your Python questions on Campuswire. You will likely get a quick response from classmates or me.
- If it is a question you don't want public, you can DM me on Campuswire. You are likely to get a faster response via Campuswire than email.
- 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 DM/post private unless it's an emergency

# Course Outline

(subject to change)

1. Week 1: Git, GitHub, conda, Jupyter
2. Week 2: Python Basics: variables, expressions, statements
3. Week 3: Python Data types: strings, lists, dictionaries, tuples
4. Week 4: Pythonic code: List comprehensions, kwargs; Numpy
5. Week 5: Pandas: Importing, reshaping, and cleaning data
6. Week 6: Pandas: Wrangling, and Aggregation
7. Week 7: Data Visualization
8. Week 8: Python Functions: encapsulation, recursion, return values, iteration
9. Week 9: Python OOP: Classes and objects, methods, inheritance
10. Week 10: Python with SQL

**GOT QUESTIONS?**

**just a little break before...**

# Academic Integrity and Plagiarism

THANK YOU DR. MILES CHEN FOR WRITING THIS PART

Let's talk about Plagiarism

**Some truths:**

- **There is a lot of high quality code that does exactly what you need available out on the Internet. Some of it is available in ready-to-install packages and some are available as solutions on places like stackexchange and github.**
- **If the goal is to accomplish a task, you should use the readily available packages or code solutions out there.**
- **However, the goal of this class is not to accomplish some task. The goal is to help you learn how to write code.**

## No Pain, No Gain

- **Think of the gym. The goal of lifting weights at the gym is not to lift weights.**  
**Lifting weights is a means to the real goal of gaining strength.**
- **"No Pain, No Gain": if your weight training does not result in some muscle soreness, you probably did not exert enough effort to expect muscle gain.**  
**Experiencing muscle soreness is a sign that your muscles will go through repairs and get stronger.**
- **Your brain is similar: if your brain does not struggle when writing code, then it has no reason to create additional neuron connections that will improve your abilities as a coder.** On the other hand, if your brain struggles with writing code, then your brain will try to create new connections between neurons so the next time will not be as hard. And thus you become a better coder.

## No Pain, No Gain

- **Plagiarizing code for a difficult assignment is like having a stronger person lift the weights that are too heavy for you.**
- **This would be a good solution if the goal of lifting weights were to lift the weights. But it does not help achieve the goal of gaining strength.**
- **Copying, pasting, and modifying a stronger programmer's code works if the goal is to accomplish a coding task. It does not help towards the goal of creating neuron connections in the brain that will make you a better coder.**

## Course Goals

- **I think students resort to plagiarism because they have confused the goals of the course.**
- **Students who plagiarize believe the goal for them is to get good grades and avoid bad grades in the class. For these students, the goal of learning is secondary to the goal of getting the desired grade.**
- **But this is short-sighted. The goal of the course is your learning.**
- **I will admit, a major conflicting issue here is that I am not able to create individualized grading schemas that evaluate exactly how much each student learned over the course. All students are graded on the same criteria and evaluated on what they turn in for the assignments.**
- **That said, I hope you can judge your performance in class based on what you learned and not your letter grade.**

## Expectations

- When you face a challenging homework assignment, I expect:
  - you will try your best
  - you will not seek out solutions online or view another (current or former) student's code
  - if you are not able to complete everything required by the assignment by the deadline
    - you submit what you have and accept a grade that is less than 100%
    - you view this not as a failure of your coding abilities, but as a indication of areas for growth and improvement

Faculty members are required to take issues of plagiarism seriously and will escalate cases to the Dean of Students. Full details regarding academic integrity are in the syllabus.

## Collaboration Policy

- **Read the course collaboration policy and be sure you understand it.**
- **For all homework assignments, verbal collaboration is encouraged but no code sharing.**
- **You are allowed to collaborate verbally with other students but you are not allowed to look at or show someone else the code you are writing. This applies to discussions on Campuswire.**

## Allowed vs Not allowed

- **Question:** "How did you approach problem 2?"
- **Allowed 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:** "Let me show you what I did... [proceeds to show screen with code, or actually shares code via email on Campuswire]"

## Allowed vs Not allowed

- **Question:** "Can I see how you did problem 2 to double check my work?"
- **Allowed response:** "I can't show you my code but I can tell you what I did."
- **Policy Violating response:** "Let me show you what I did... [proceeds to show screen with code, or actually shares code via email or Campuswire]"

## Allowed vs Not allowed

- **Question:** "Can you help me find what I'm doing wrong? I've got a bug but I can't figure out where. It keeps saying 'missing value where TRUE/FALSE needed'"
- **Allowed response:** "Did you check to make sure ....?"
- **Allowed response:** "That error message often pops up if you have an NA inside an if statement."
- **Policy Violating response:** "Let me see your code... [proceeds to examine and modify code]"

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 bBpost and discuss code that appears in lecture. You are encouraged to modify the examples the appear in lecture and discuss 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.**

**Before we start, if you have time – listen to the wisdom of Dr. Chen on Grades and Life (about 20 minutes)**

Stats 21 - 2022-01-03 - Week 1 Monday - Welcome



## Installing Python - Anaconda and Conda

- **Python is a little messy. There are different versions of Python (we will use Python 3).**
- **Also different packages may have others as dependencies and if you install the wrong version, code can break.**
- **Anaconda is collection of commonly used Python libraries along with Conda to facilitate installing and maintaining Python libraries and installations.**
- **Included with Anaconda are:**
  - **Jupyter Notebook and Jupyter Lab**
  - **NumPy, SciPy, Matplotlib**
  - **Pandas, Scikit-Learn, Seaborn**

# Download and Install Anaconda

<https://www.anaconda.com/products/individual>

The screenshot shows the Anaconda website at <https://www.anaconda.com/products/individual>. The page features a navigation bar with links for Products (selected), Pricing, Solutions, Resources, Partners, Blog, and Company. A prominent green 'Get Started' button is visible on the right. The main content area highlights the 'Individual Edition' with a large green 'Q' icon and a brief description: 'Your data science toolkit'. Below this, a detailed paragraph explains the distribution's purpose: 'With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.' To the right, a callout box for the 'Anaconda Individual Edition' includes a 'Download' button with an Apple logo, a note for Mac OS users ('For Mac OS'), and download links for Python 3.9 (64-Bit Graphical Installer, 515 MB). At the bottom, there's a 'Get Additional Installers' section with icons for Windows, Apple, and Linux.

Individual Edition

# Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Anaconda Individual Edition

Download

For Mac OS

Python 3.9 • 64-Bit Graphical Installer • 515 MB

Get Additional Installers

| |

## 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".

# Q terminal — Terminal



## Terminal

Docker Quickstart Terminal

terminal application



terminal

## Definition

Aa 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 Feb 19 14:03:17 on ttys001

(base) vivian@Vivians-MacBook-Pro ~ % █

## Opening Anaconda PowerShell on Windows

**Use the Start Menu to find "Anaconda Powershell Prompt". It is found under the Anaconda3 folder.**

**You can find it quickly by hitting the Windows Key and then immediately typing " anaconda pow " then enter**

**I suggest pinning the shortcut to the Start Menu.**

**If you start regular PowerShell instead of Anaconda Powershell, it will not have environmental variables set up correctly and you will not be able use conda or start Jupyter.**

# 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 type).**

**`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.

`conda`

**Conda is a package and environment manager. When you install Anaconda, `conda` is a utility that gets installed too. You can use `conda` from the terminal or Anaconda PowerShell.**

**Most of the packages you need will in Stats 21 come pre-installed with Anaconda. However, you may need to install a new package in the future or update your packages.**

**Installation is done with**

`conda install packagename`

**Updating can be done with**

`conda update packagename`

**or you can update all of them with**

`conda update --all`

**If you are using Windows, I recommend starting Anaconda Powershell Prompt with Administrator rights before running `conda update --all`**

# Jupyter Lab and Jupyter Notebook

**I will discuss this further in the coming days. But if you just want to start playing with Python, you can launch Jupyter Notebook (preferred -simpler) or Jupyter Lab (if you already possess Python skills and know Jupyter Notebook).**

```
jupyter lab
```

```
jupyter notebook
```

See you next time

Loving Caliber feat. Frank Moody - Drive Drive Drive

