STATS 507

Data Science and Analytics using Python

Xian Zhang

Stats 507 Basics

1. “Data science and analytics using Python”

2. Reserved for Stats/Data science masters and Ph.D. student, but open for wide variety of students: economics, political science, engineering, agricultural…have senior undergrads enrolled too.

3. Enrollment: 92 for 2024 Fall, 94 in 2025 Winter

Course Goals

1. Establish a broad background in **Python** programming
2. Prepare you for the inevitable coding interview
3. Survey popular **tools** in academia/industry for data analysis
4. Learn how to read documentation and quickly get familiar with new tools.

Tool/programming languages might be obsolete one day…

…But not your ability to learn new frameworks and solve problems!

Scope of STATS 507

Part 1: Introduction to Python

Data types, functions, classes, objects, algorithm thinking, git and program profiling.

Part 2: Numerical Computing and Data Visualization

Numpy, Scipy, Scikit-learn, Matplotlib, Seaborn

Part 3: Dealing with Structured Data

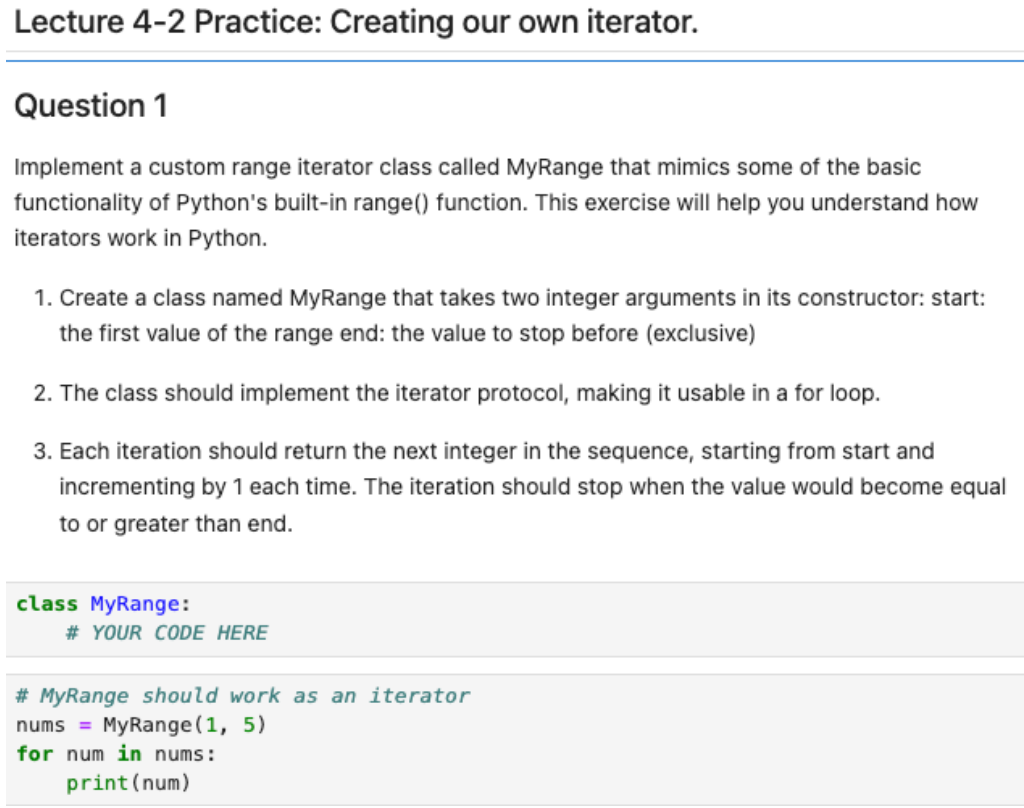
Pandas, SQL, real datasets

Part4: Intro to Deep Learning

PyTorch, Perceptron, Multi-layer perceptron, SGD, regularization, CNN…

Course Format - Lectures

- This is a **lecture** format class
- Slides and in-class practice will be uploaded the night before class.
- Each lecture will have an in-class practice for interactive coding.



Lecture 4-2 Practice: Creating our own iterator.

Question 1

Implement a custom range iterator class called MyRange that mimics some of the basic functionality of Python's built-in range() function. This exercise will help you understand how iterators work in Python.

1. Create a class named MyRange that takes two integer arguments in its constructor: start: the first value of the range end: the value to stop before (exclusive)
2. The class should implement the iterator protocol, making it usable in a for loop.
3. Each iteration should return the next integer in the sequence, starting from start and incrementing by 1 each time. The iteration should stop when the value would become equal to or greater than end.

```
class MyRange:
    # YOUR CODE HERE
```

```
# MyRange should work as an iterator
nums = MyRange(1, 5)
for num in nums:
    print(num)
```

Course Format – HWs (40%)

- 8 HWs in total
- Jupyter notebook or Google Colab
  - Upload of .ipynb
- Auto-grade using nbgrader
  - Adding hidden tests and grading are very convenient.
  - But download students notebook and give feedback need manual work from GSI.

3.2: Group Anagram (3 points)

Given an array of strings, group the anagrams together. You can return the answer in any order.

An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once.

```
def group_anagrams(my_string):
    ### BEGIN SOLUTION
    d = {}
    for word in my_string:
        s = "".join(sorted(word))
        if s not in d:
            d[s] = [word]
        else:
            d[s].append(word)

    res = []

    for item in d.values():
        res.append(item)

    return res
    ### END SOLUTION
```

```
assert group_anagrams(["", "b"]) == [[''], ['b']]
assert group_anagrams(["listen", "silent", "enlist"]) == [['listen', 'silent', 'en
assert group_anagrams(["eat","tea","tan","ate","nat","bat"]) == [['eat', 'tea', 'a
assert group_anagrams(["Tea", "Eat", "tea", "eat"]) == [['Tea'], ['Eat'], ['tea',
### BEGIN HIDDEN TESTS
assert group_anagrams(["hello", "world", "openai"]) == [['hello'], ['world'], ['op
assert group_anagrams(["a", "b", "c", "a"]) == [['a', 'a'], ['b'], ['c']]
### END HIDDEN TESTS
```

Course Format – Midterms (30%)

An **in-person** midterm is scheduled after SciPy introduction

- Evaluation of internalization of core ideas covered in lecture. (Focus on what and why, instead of how…)
- Closed book.
- First tried it in 2024 FA, average 70/100.

Individual final project (30%)

Assuming knowing Python and related tools, the final project for this course is to integrate with at least one open model or dataset on Hugging Face.

Proposal Requirements

- Format: at least 2 pages, at least 1 figure.
- Sections:
  Overview
  - Describe the background and motivation of the project.
  - Why would you want to do this project?
  - What kind of data/model do you want to work on?
  - What kind of insights are you expecting from this projects?

  Prior Work
  - Literature review: conduct a comprehensive review of existing literature, focusing on recent advancements and state-of-the-art.
  - Describe potential methods that can be used to achieve the project goal.

  Preliminary Results
  - Data understanding (dataset statistics, data quality assessment, initial insights, potential challenge)
  - Basic model (resource requirements, performance bottlenecks…)
  - What tools from class did you use, what tools will you explore?

Their favorite!

Suggestions?