MATH 210 Project 1

Python Packages

Scientific computing in Python begins with the SciPy stack: NumPy, SciPy, Matplotlib and pandas. But there are *many* more Python packages that have been developed by the global Python community for scientific computing. **Your assignment is to choose a Python package which interests you and to write a Jupyter notebook to introduce the package to your peers.**

INSTRUCTIONS

Choose a package. The first step in this project is to choose a package that interests you. However, **there are 3 restrictions:**

1. You may **not** choose a package/subpackage/function that we will cover in class such as `scipy.integrate`, `scipy.linalg`, and the parts of `numpy` and `matplotlib` we have used so far. For example, you may not choose to write about NumPy arrays however you may choose to introduce the Fourier transform subpackage `numpy.fft`.
2. The package must be related to mathematics/statistics in some way.
3. The package must be supported by `ubc.syzygy.ca`. (For example, there are some animation and plotting packages which won't work in a Jupyter notebook.)

Get feedback. We will form small groups in class on **Wednesday March 1** and discuss the packages we've chosen. You will get feedback from your group about your package and you will provide feedback about theirs. When giving feedback, think about these questions regarding your peer's project proposal:

1. Is the project proposal well-defined? If not, does your peer need to focus on a more specific part of their chosen package?
2. Is their project too simple? If so, what would make their project more substantial?
3. How is their chosen package related to mathematics/statistics?
4. Are there interesting examples you would like to see demonstrated?
5. Is the project showing you something new or have we already seen it in class?

Write a first draft of your project. Follow the outline given in the next section below. Keep your audience in mind. You are introducing your chosen package to your peers. Think of what they would want to see and how they would want your notebook to be organized.

Get more feedback. We will form the same small groups in class on **Wednesday March 8** and discuss the first draft of your project. At this point, your notebook does not need to be complete but your ideas should be clear. Think about the questions from the previous week's feedback.

Submit your project. Submit your notebook to Connect by **11pm Friday March 17**.

PROJECT OUTLINE

Introduce the package. What package/subpackage/function have you chosen? What kinds of problems is it designed for? How is it used? Why did you choose this package?

Set the learning outcomes. What is your project focused on? If you're working with a package with many subpackages and functions, specify which subpackage/function you are focusing on. What do you want readers to learn from your notebook?

Write the tutorial. How do you use the package? Write Python code which demonstrates how to use the package. Write markdown language to organize your tutorial into sections and to explain your code. Use LaTeX to present theorems, definitions and examples from mathematics/statistics related to your package.

Give examples and provide further references. Consider several examples to show the variety of features of your package. Add links to online documentation. Add links to YouTube videos. Share anything that you think is helpful.

SCHEDULE and DELIVERABLES

Wednesday March 1 – Peer feedback I (1 mark)

Wednesday March 8 – Peer feedback II (1 mark)

Friday March 17 – Submit Jupyter notebook (10 marks)

March 20 – 24 – Share notebooks and learn from each other's work (3 marks)

INSPIRATION

SciPy and PyData conference videos on `YouTube`

`scikit-learn` – Machine learning

`pandas` – Data analysis

`scipy.stats` – Statistical functions

`scipy.special` – Special functions

`scipy.optimize` – Optimization

`sympy` – Symbolic computation: integration, differentiation, algebra, number theory, etc.

`sympy.physics` – Symbolic computation for physics

`sympy.ntheory` – Symbolic computation for number theory

`bokeh` – Interactive data visualization

`mpld3` – Interactive data visualization

`scikit-image` – Image processing

`matplotlib.animation` – Create animations with matplotlib

`ipywidgets` – Create interactive graphics and animations with `ipywidgets` and `matplotlib`