Exercise class 3

(week 10)

Introduction to Programming and Numerical Analysis

Class 4 and 8

Rosa Haslund Meyer

Spring 2024

KØBENHAVNS UNIVERSITET

Random numbers

User-defined modules

Exercises: Problem set 2

Poll

Plan for today's exercise class

- 15.15-15.30: recap – random numbers and user-defined modules + a few tips
- 15.30-16.00: exercises – problem set 2
- 16.00-16.15: Break
- 16.15-16.20: tips - I'll go through the rest of the tips
- 16.20-16.50: exercises – problem set 2
- 16.50-17.00: poll and follow-up questions

Random numbers

Simulating random numbers is an essential (and fundamental) tool in numerical economics:

- Models which include uncertainties regarding outcome
- Models with heterogeneous preferences, endowments, productivity, etc.
- Simulation of difficult integrals such as Monte Carlo Integrations/simulations

Important take-aways from this week's lectures and problem set:

- Drawing random numbers from different distribution
- Using seed and states to ensure reproducibility
- Solving exchange economies with heterogenous preferences
- Numerical integration (by Monte Carlo)

User-defined modules (.py-files)

When programming it's fundamental to keep your code clean and as readable as possible.

One way of doing this is to collect functions and classes in **modules**!

It's possible to write functions etc. that you'll need in your main body of your project in .py-files and then **import** them into your main notebook (or .py-file). This is done in the same way as you would import for example NumPy.

This helps to ensure a nice clean notebook/project and is also a really great way to eliminate many types of **bugs**!

Tips!

Task 1:

- First you want to save a state and secondly you want to reset that state.
- You should get a table looking like this:

```
(0,0): x = 0.569
(0,1): x = 0.077
(1,0): x = 0.569
(1,1): x = 0.077
(2,0): x = 0.569
(2,1): x = 0.077
```

Task 2:

- You need to find the expected values, i.e. mean and variance using the transformation function:
  a. Draw random numbers from the given distribution of x_i
  b. Define the transformation function g(x,omega)
  c. Define the mean and variables and print them

- You should get something like this:

```
Mean of transformed data:     -0.002641941946664676
Variance of transformed data: 2.698040741876887
```

Tips!

Task 3:

- Firstly, import needed packages: `import ipywidgets as widgets and from scipy.stats import norm`
- Define x_low and x_high using norm.ppf() from SciPy. The norm.ppf() function is the probability value for which you want to find the corresponding quantile

Second part of task 3:

- Add widgets to your histogram made in the above cell (task 3)
- You should use the ipywidgets.interact() function. The arguments of the function should be your histogram (from previous cell), random draws from the given distribution using ipywidgets.fixed() and two sliders: one for mu and one for sigma using ipywidgets.FloatSlider()

Time for exercises

Problem set 1:

- Task 1-2: drawing random numbers and finding expected values
- Problem 3: interactivity and widgets
- Problem 4: user defined-modules
- (Problem 4.5): Git
- Problem 5-6: solve exchange economy + pickles (save and load)

At **16:15**, I'll go through the rest of the tips.

Tips!

Task 4:

- You can import py-files like you import NumPy and then call the functions from within that py-file like you call a function from NumPy
- Be aware that the function you need to call takes an argument
- Be aware that you'll not get any output if your argument is not a strictly positive integer: i.e. n>0 (because of the function in the py-file you make a call to)

Git task:

- Take your time! It will make sense as soon as you start to use Git regularly!

Tips!

Task 5:

- Look in the lecture notebook: Random_numbers_basic and/or the below cell
- Be aware of default parameter values
- Make sure you understand the code – sometime in the future you might need to modify it!

Task 6:

- The my_data variables are merely some user-defined data, you should define `my_data['C']` and `my_np_data['F']`
- Next you want to write `my_data` to a file called data.p (saving data using pickle)
- Then you want to write `my_np_data` to a file called data.npz (using NumPy)

Poll

I want to know your opinion on the exercise classes so far using this poll!

Next time...

Video lectures:

- Structure
- Debugging
- Git
- Consumer problem
- Labour supply
- Production economy

Exercises – Git + inaugural project:

- Introduction to Git (by me)
- Work on your inaugural project which is due 24th march! Remember physical lecture Monday March 11th

Questions and/or comments?