Class 6, Inuagural Project

Introduction to Programming and Numerical Analysis

Inaugral Project

Remember to download the new version !!!

Requirements

1. Apply simple numerical solution and simulation methods
    - Solve the problem using SciPy
2. Structure a code project
    - Notebook, .py-files, readme etc.
3. Document code
    - Document and explain your code using what you learned in lecture 05
4. Present results in text form and in figures
    - Create nice tables and figures

Deadline: 27/3 and not 20/3

General hints

Create Python file for your functions

General hints

Create Python file for your functions

In [1]:

```
import functions
functions.u(1,2,0.3)
```

Out[1]:

```
1.624504792712471
```

General hints

Docstrings: Example (See functions.py)

General hints

Document your code! At the exam you might have forgotten what you did in this project

UNIVERSITY OF COPENHAGEN

In [ ]:

```
# 0. Import packages
import numpy as np

# 1. Initiliaze vectors of goods and container for utilities
alpha = 0.3
x1s = np.linspace(0, 5, 10)
x2s = np.linspace(5, 10, 10)
utility = np.empty((x1s.shape)) # empty vector with same shape as the vector of goods

# 2. Calculate utility for all goods by looping
for idx, (x1, x2) in enumerate(zip(x1s, x2s)):
    # i. Calculate utility using functions module
    temp_utility = functions.u(x1,x2,alpha)

    # ii. Append utility to empty vector
    utility[idx] = temp_utility

    # iii. Print result
    print(f'u({np.round(x1,2)}, {np.round(x2,2)}) = {np.round(utility[idx],2)}, where \u03B1 = {alpha}')
```

General hints

UNIVERSITY OF COPENHAGEN

Incoporate markdown cells in your project that describes the model, intuition, etc. Remember you do can do equation in markdown using either `$$ math $$`:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

or `\begin{align} math \end{align}`

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{1}$$

Other formatting?

General hints

Scope of variables

Global or local??

In [4]:

```
global_var = 1
local_var = 99

def random_function(number, add_global):
    local_var = 1
    if add_global == True:
        return number + local_var + global_var
    if add_global == False:
        return number + local_var

print(f'With global: {random_function(global_var, True)}')
print(f'Without global: {random_function(global_var, False)}')
print(local_var)
```

```
With global: 3
Without global: 2
99
```

General hints

Presentation of results matter - you can't just hand in blocks of code

E.g., when creating figures remember:

```
ax.plot(x,y,label='Nice label')
```

```
ax.set_ylabel('$\pi$')
```

```
ax.legend()
```

and so on... Maybe an interactive figure?

Jupyter Lab tips and tricks

https://towardsdatascience.com/optimizing-jupyter-notebook-tips-tricks-and-nbextensions-26d75d502663

UNIVERSITY OF
COPENHAGEN

If you want to pass read this

ALLWAYS restart your kernel and run all the cells before handing in! This goes for the mandatory assignments along with the exam. It is crucial!

If you have old variables in memory that make your notebook run then i cannot run the notebook afterwards!

Hints for the project

Control your variables by using `SimpleNamespace()`

UNIVERSITY OF
COPENHAGEN

In [6]:

```
from types import SimpleNamespace
question1parameters = SimpleNamespace()
parameters.alpha = 0.3
parameters.beta = 1
print(parameters.alpha)
```

```
0.3
```

In [6]:

```
from types import SimpleNamespace
question1parameters = SimpleNamespace()
parameters.alpha = 0.3
parameters.beta = 1
print(parameters.alpha)
```

```
0.3
```

In [7]:

```
print(functions.u(1,2,parameters.alpha))
parameters.alpha = 8
print(functions.u(1, 2, parameters.alpha))
```

```
1.624504792712471
0.0078125
```

Hints for question 1

UNIVERSITY OF COPENHAGEN

Construct a function that takes (x, y, p) as arguments and returns the agents' optimal insurance coverage

1. Define functions for utility, premium and expected value
2. Create a function **that takes (x, y, p)** and optimize the expected value
    - this includes creating and objective function and using minimize_scalar with bounds - what is the upper bound?

Calculate and plot the optimal q for x in the range[0.01 - 0.9]

1. Create the range using NumPy
2. Loop through possible values of x and return the optimal q
3. Plot the result

Hints for the question 2

Find $\tilde{\pi}$ such that $V(q;\tilde{\pi})$ = V0 at each point in the grid

1. Create a new value function that takes π as argument - and not the premium function as before
2. Create new objective function: $V(q;\tilde{\pi}) = V0 \rightarrow V(q;\tilde{\pi}) - V0 = 0$
3. optimize.root -> solve objective = 0 -> coverage minus no coverage
4. Construct the q grid
5. Loop through the q's and return the optimal π

Hints for the question 3

1. Create a function for MC integration - see lecture 04. Let it take two arguments that both are SimpleNamespace()
    - SimpleNamespace() for parameters
    - SimpleNamespace() for the different policies

How to hand in?

Your new best friend GitHub!

Make sure your repo is up and running

UNIVERSITY OF
COPENHAGEN