Final Examination
Quantitative Economics with Python

May 12, 2016

Instructions: This is a closed book examination. You may consult the Python reference sheet that accompanies this exam. Please answer all questions. Prospective points are indicated beside each problem.

(30 points) Programming and Computation

Q1. (10 points) One of your friends is undertaking a project that requires working with a large amount of data. You decide to introduce her to this great new Python package called **pandas**

a. How would you explain what **Pandas** does to your friend?

Solution: Pandas is a Python package that provides tools for working with data. It provides two primary objects (pd.Series and pd.DataFrame) that stores labeled data and provides a variety of functions to conduct quick and agile data analysis.

b. What is a `pd.Series()` object?

Solution: A pd.Series object is a one-dimensional ndarray with axis labels that can be single or hierarchical. The pd.Series supports both integer and label based indexing.

c. What is a `pd.DataFrame()` object?

Solution: A pd.DataFrame object is a two-dimensional tabular data structure. It supports labeled axes for both rows and columns, with operations that align on both row and column labels. It can be thought of as a collection of pd.Series objects that share a common set of labels.

Your friend finds some **Python** code and asks you to explain it to her:

```
In [1]: import pandas as pd

In [2]: s = pd.Series(np.random.randn(5), index=['a', 'b', 'c', 'd', 'e'])

In [3]: s1 = s[:4]

In [4]: s2 = s[1:]

In [5]: s1.align(s2)
Out[5]:
(a   -0.365106
 b    1.092702
 c   -1.481449
 d    1.781190
 e         NaN
 dtype: float64, a         NaN
 b    1.092702
 c   -1.481449
 d    1.781190
 e   -0.031543
 dtype: float64)

In [6]: s1.align(s2, join='inner')
Out[6]:
(b    1.092702
 c   -1.481449
 d    1.781190
 dtype: float64, b    1.092702
 c   -1.481449
 d    1.781190
 dtype: float64)

In [7]: s1.align(s2, join='left')
Out[7]:
(a   -0.365106
 b    1.092702
 c   -1.481449
 d    1.781190
 dtype: float64, a         NaN
 b    1.092702
 c   -1.481449
 d    1.781190
 dtype: float64)
```

d. What is contained in the variable **s**?

Solution: The variable **s** is a pandas Series type object which contains the following data:

```
a   -0.365106
b    1.092702
c   -1.481449
d    1.781190
e   -0.031543
```

e. What is contained in the variable **s1**?

Solution: The variable **s1** is a pandas Series type object which contains the following data:

```
a   -0.365106
b    1.092702
c   -1.481449
d    1.781190
```

f. What is contained in the variable **s2**?

Solution: The variable **s** is a pandas Series type object which contains the following data:

```
b    1.092702
c   -1.481449
d    1.781190
e   -0.031543
```

g. What is the **align()** method doing?

Solution: The **align()** method takes two pandas Series and aligns the objects based on their labels.

h. What does the keyword argument **join** do?

Solution: The keyword argument **join** is a way to specify how the align operation should perform. An 'inner' join takes the intersection of each series, or in other words will align the common elements that are contained in each series. An 'outer' join takes the union of each series.

i. What is the return type of the **align()** operation?

Solution: A **tuple** that contains two aligned series objects.

Q2. (10 Points) Given the following python code.

```
a = [1,2,3,4,5]
b = [0,1,0,1,0]
```

a. What would `type(a)` return?

Solution: `type(a)` would return: `list`

b. If a and b represent vectors. How would you use a **for** loop to compute the inner product?

Solution:

```
result = 0
for idx in range(0,len(a)+1,1):
    result += a[idx] * b[idx]
```

c. How would you use **numpy** to compute the inner product?

Solution:

```
result = np.dot(a,b)
```

For the following questions you are given another line of python code

```
c = [a,b]
```

d. What does the **python** variable c contain?

Solution: The python variable c contains two list objects. It is a list of lists.

```
[[1,2,3,4,5], [0,1,0,1,0]]
```

e. What would `type(c)` return?

Solution: list

You are in a discussion with a friend who is trying to convince you that this is a good data format to represent matrices.

f. How would you return the second row of the matrix?

Solution: To return the second row:

```
c[1]
```

g. How would you return the second column of the matrix?

Solution: To return the second column, one way of doing this is to use loops:

```
colidx = 1  #second column
col = []
for row in c:
    col.append(row[colidx])
```

h. Do you think this is a good data format to represent matrices?

Solution: This is not a particularly convenient way to interact with matrices as it is a very basic data structure. It would be better to use a specialized data structure such as numpy arrays. It provides more convenient access to row and column items and provides infrastructure for solving problems using linear algebra.

Q3. (10 points) Given a real valued function $f(x)$ of a real variable x, a root of f(x) is defined as a value of x that solves:

$$x : f(x) = 0$$

Given an initial guess x_0 for a root, one can use the following formula to get an improved guess x_1:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

We can repeat to produce an improved approximation, a procedure called Newton's Method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

a. Explain how you can use **Newton's Method** solve for the roots of the following function:

$$f(x) = x^3 - x + 1$$

Solution: First take the function

$$f(x) = x^3 - x + 1$$

and solve for its first derivative

$$f^{`}(x) = 3x^2 - 1$$

.

We can then use Newton's method to iteratively solve for improved approximations to solve the equation $f(x) = 0$. Given an initial guess, this is accomplished by looping over the equation:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

or

$$x_{n+1} = x_n - \frac{x^3 - x + 1}{3x^2 - 1}$$

b. A project that you are working on requires you to know the root of this $f(x)$ to <u>4 decimal</u> places. Use your work from part (a) as a basis for writing a Python program that numerically solves for the root.

Solution: One way to solve this problem using a **while** loop.

```
x = 0
diff_error = 1
while diff_error > 1e-4:
    xnext = x - (x**3 - x + 1)/(3*x**2 - 1)
    diff_error = xnext - x
    x = xnext
```

You can also use **for** loops to solve this problem. It can also be a good idea to add a **maxitr** variable (to define a maximum number of iterations) in case the loop condition is never met.

(70 points) Quantitative Economics

Q4. (20 Points) Finite Markov Chains

a. What are the properties of a well-defined stochastic matrix?

Solution: Refer to Course Notes

You are given two stochastic matrices:

$$A = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\alpha & \alpha \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix} \tag{2}$$

b. For what values of α is A a valid stochastic matrix?

Solution: Refer to Course Notes

c. If $\lambda > \alpha$ which Markov chain, that is implied by the stochastic matrix A or B, would you expect to exhibit **higher** persistence? Explain.

Solution: Refer to Course Notes

Assume the following **Markov** matrix:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

where, $S = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}$

d. Given that you are in state 1 today, what is the conditional probability of being in state 2 tomorrow?

Solution: Refer to Course Notes

e. Given that you are in state 2 today, what is the conditional probability of being in state 1 in two ($T = 2$) periods?

Solution: Refer to Course Notes

f. Starting from an initial probability distribution over states $\pi_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, what is the probability distribution of the state at time $T = 1000$?

Solution: Refer to Course Notes

g. Starting from an initial probability distribution over states $\pi_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, what would you expect the steady state probability distribution to be?

Solution: Refer to Course Notes

Q5. (25 Points) Consider an asset pricing model. The price of the asset satisfies the following equation

$$p_t = \beta \mathbb{E}_t[m_{t+1}(p_{t+1} + d_{t+1})]$$

a. First consider a deterministic version of the asset pricing model. Assume $d_t = \lambda^t$, and $m_t = 1$. The asset pricing function becomes

$$p_t = \beta(p_{t+1} + d_{t+1})$$

Solve for the price p_t as a function of λ, β, and t.

Solution: Assume $p_t = v d_t = v\lambda^t$. Then

$$v\lambda^t = \beta(v\lambda^{t+1} + \lambda^{t+1})$$

which leads to

$$v = \frac{\lambda}{1 - \beta\lambda} \tag{3}$$

Then $p_t = \frac{\lambda^{t+1}}{1-\beta\lambda}$

b. Still consider a deterministic version of the asset pricing model. Assume $d_t = \lambda^t$, and $m_t = \lambda^{-\gamma}$. Now solve for the price p_t as a function of λ, β, γ, and t.

Solution: Assume $p_t = vd_t = v\lambda^t$. Then

$$v\lambda^t = \beta(v\lambda^{t+1-\gamma} + \lambda^{t+1-\gamma})$$

which leads to

$$v = \frac{\lambda^{1-\gamma}}{1 - \beta\lambda^{1-\gamma}} \tag{4}$$

Then $p_t = \frac{\lambda^{t+1-\gamma}}{1-\beta\lambda^{1-\gamma}}$

c. Now we move to the stochastic case. Assume $m_t = \lambda_t^{-\gamma}$ and $d_{t+1} = \lambda_{t+1}d_t$. λ_t can take two possible values s_1 and s_2. A Markov transition probability matrix for λ_t is given by

$$P = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\theta & \theta \end{bmatrix} \tag{5}$$

Assume $p_t = v(\lambda_t)d_t$. Write down the two equations that $v(s_1)$ and $v(s_2)$ must satisfy.

Solution:

$$v(s_1) = \beta\alpha s_1^{1-\gamma}[1 + v(s_1)] + \beta(1-\alpha)s_2^{1-\gamma}[1 + v(s_2)]$$
$$v(s_2) = \beta(1-\lambda)s_1^{1-\gamma}[1 + v(s_1)] + \beta\lambda s_2^{1-\gamma}[1 + v(s_2)]$$

You are now given the following information.

1. There are n possible states in the economy
2. P is an $n \times n$ probability transition matrix (stochastic matrix)
3. x_t is a discrete random variable which takes the value λ_s if the state of the economy is s, where $s \in \{1, 2, ..., n\}$
4. R is an $n \times n$ matrix that contains the returns to some investment between state i today and state j tomorrow

d. Please express the following:

$$E_t[x_{t+1}|x_t = \lambda_s]$$

in mathematical notation using the $\sum$ notation.

Solution: Refer to Course Notes

e. Please write a **python** program to compute the conditional expectation

$$E_t[x_{t+1}|x_t = \lambda_s]$$

Solution:

```
P @ X
```

f. Please write a **python** program to compute the conditional expectation:

$$E[R|s] = \sum_{j=1}^{n} R_{sj} \times P_{sj}$$

Solution:

```
(R * X).sum(axis=1)
```

Q6. (25 Points) Consider the McCall search model. If a worker is unemployed, she will receive unemployment benefit c. At the end of the period, the worker may receive a job offer with probability λ. There are 2 possible job offers, $\{w_1, w_2\}$, with probability $\{1-p, p\}$ respectively. Assume $w_2 > w_1 > c$. The worker can decide whether to accept a job offer. If she accepts, then she will become employed next period. An employed worker receives wage w_s ($s = 1$ or 2). At the end of a period, with probability α, the employed worker can become unemployed; with probability $1 - \alpha$, the worker can be still employed with the same wage w_s next period.

Assume the discount factor $\beta \in (0, 1)$. A worker wants to maximize her lifetime utility

$$\mathbb{E} \sum_{t=0}^{\infty} \beta^t y_t$$

where y_t equals to c if the worker is unemployed, and y_t equals to w_s if the worker is employed with wage w_s.

A list of parameters are

- Offer arriving rate: λ
- Involuntary separation rate: α
- Discount factor: β
- Wage offers: $\{w_1, w_2\}$ and the associated probability $\{1 - p, p\}$
- Unemployment benefit: c

a. Let V_1 denote the lifetime value of being employed at wage w_1 today, V_2 denote the lifetime value of being employed at wage w_2 today, and U denote the lifetime value of being unemployed today. Write a dynamic program for V_1, V_2 and U.

Solution: The dynamic program is

$$\begin{aligned} V_1 &= w_1 + \beta \left[(1-\alpha) V_1 + \alpha U \right] \\ V_2 &= w_2 + \beta \left[(1-\alpha) V_2 + \alpha U \right] \\ U &= c + \beta(1-\lambda)U + \beta\lambda \left[(1-p) \max\{U, V_1\} + p \max\{U, V_2\} \right] \end{aligned}$$

b. Assume that the parameters are such that workers never want to accept the job offer w_1. Write down the transition probability matrix P between employment and unemployment.

Solution:

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \lambda p & 1-\lambda p \end{bmatrix}$$

c. Given the transition probability matrix P that you have computed in the last question, solve for a stationary distribution of employment rate e and unemployment rate u, such that

$$\begin{bmatrix} e & u \end{bmatrix} = \begin{bmatrix} e & u \end{bmatrix} P \tag{6}$$

Solution:

$$e = \frac{\lambda p}{\lambda p + \alpha}$$
$$u = \frac{\alpha}{\lambda p + \alpha}$$

d. Show that if $w_2 > w_1 > c$, then $V_2 > V_1$ and $V_2 > U$.

Solution: First,

$$V_2 - V_1 = \frac{w_2 - w_1}{1 - \beta(1-\alpha)} > 0$$

Second, if $U \geq> V_2$, then

$$U = \frac{c}{1-\beta}$$

However, if this is true, then we have

$$V_2 = w_2 + \beta\left[(1-\alpha)V_2 + \alpha U\right] \geq w_2 + \beta V_2$$

which leads to

$$V_2 \geq \frac{w_2}{1-\beta}$$

e. Assume $\alpha = 0$. Solve for V_1 and V_2.

Solution: It can be easily done.

$$V_1 = \frac{w_1}{1-\beta}$$
$$V_2 = \frac{w_2}{1-\beta}$$

f. Still assume $\alpha = 0$. Suppose that a worker always reject a wage offer w_1 and accepts a wage offer w_2. Compute the value of being unemployed, U.

Solution:

$$U = c + \beta(1-\lambda)U + \beta\lambda\left[(1-p)U + pV_2\right]$$

$$U = \frac{c + \beta\lambda p \frac{w_2}{1-\beta}}{1-\beta(1-\lambda p)}$$

g. Still assume $\alpha = 0$. If the worker always optimally rejects a job offer w_1 and accepts a wage offer w_2, then it must be that $U > V_1$. Intuitively, if the unemployment benefit c is large enough, the worker will not take the low paid job offer w_1. Show that if $c = w_1$, then $U > V_1$.

Solution:

$$\frac{U}{V_1} = \frac{\frac{c+\beta\lambda p\frac{w_2}{1-\beta}}{1-\beta(1-\lambda p)}}{\frac{w_1}{1-\beta}} = \frac{1-\beta+\beta\lambda p\frac{w_2}{w_1}}{1-\beta(1-\lambda p)} > \frac{1-\beta+\beta\lambda p}{1-\beta(1-\lambda p)} = 1$$