



Python Programming for Finance Exam 2019

Instructions

- Slides / calculators allowed
- Code can be tested using the environment of your choice (Spyder, Jupyter...)
- Personal documents allowed

In the followings section, you will be asked to either write python scripts or answer multiple choice questions. If a python script is requested, be careful to answer with the same syntax as the python script file (a code with syntax errors, not readable by Python, will be considered as incorrect). We advise to prepare the scripts on a dedicated python environment and then export the content of your script (copy/paste) as answer.

For multiple choices questions, several answers can be correct.

Part 1: Python syntax and objects

Question 1

Write a function “factorial” which takes one argument “n” and returns the factorial of n.

Example:

factorial(4) should return 24 ($4*3*2*1$)

factorial(5) should return 120 ($5*4*3*2*1$)

Question 2

Write a function “sort_list” which takes a list of numbers as argument, sorts the list in ascending order, and returns the sorted list. The function should not use the native “sort” method.

Example:

sort_list([9,5,3,1,2]) should return [1,2,3,5,9]

Question 3

Assuming you are using a python 2.XXX environment (2.7 or below) what is the value of y resulting from the below expression ?

```
x=150/100
```

```
y=x*10-20
```

```
print(y)
```

Answers:

A=-15

B=-5

C=-10

D: None of the above

Question 4

Assuming you are using a python 3.XXX environment what is the value of y resulting from the below expression ?

```
x=150/100  
y=x*10-20  
print(y)
```

Answers:

A=-15

B=-5

C=-10

D: None of the above

Question 5

Write a function “unique” which takes a list of numbers as argument and removes the duplicates.

Example:

unique([9,5,3,1,2, 9,2]) should return [9,5,3,1,2]

Question 6

Using a while loop, write a procedure creating the list of odd values between 0 and 100.

Question 7

What is result of the below python script ?

```
a = [[1, 2], [2, 4]]  
b = a.copy()  
b[1].append(3)  
print(a)
```

Answers:

A=[[1, 2, 3], [2, 4]]

B=[[1, 2, 3], [2, 4, 3]]

C=[[1, 2], [2, 4], 3]

D: [[1, 2], [2, 4]]

Question 8

What is result of the below python script ?

```
def square_values(arg):  
    arg=[x**2 for x in arg]  
    return arg
```

```
arg={1,2,3}  
square_values(arg)  
print(arg)
```

Answers:

A=[1,2,3]

B={1,2,3}

C=[1,4,9]

D: {1,4,9}

Question 9

Create a function "count_characters" which takes a string "string_arg" and a character "c" as argument and returns the number of times the character "c" is found in "string_arg".

Example:

count_characters("business", "s") should return 3

Question 10

Using the Numpy package, write a function "slicing" which takes a numpy matrix m as argument and returns only the second row of the matrix.

Example:

```
m=numpy.array([[1,2,3], [4, 6, 9], [5, 4, 0]])
```

slicing(m) should return [4,6,9]

Question 11

What is displayed by the below procedure ?

```
def square_values(arg):  
    for i in range(len(arg)):  
        arg[i]=arg[i]**2  
    return arg
```

```
arg=[1,2,3]  
square_values(arg)  
print(arg)
```

Answers:

A=[1,2,3]

B={1,2,3}

C=[1,4,9]

D={1,4,9}

Question 12

What is the result of below script ?

```
def rec_diff(n):  
    if n==1:  
        return 1  
    else:  
        return n-rec_diff(n-1)  
print(rec_diff(3))
```

Answers:

A=0

B=1

C=2

D=3

Question 13

Using the function linregress of scipy.stats, write a function “compute_linregress” which takes 2 lists of numbers as arguments and returns a tuple containing the slope and the intercept of the linear regression between the two lists.

Question 14

Using the function `numpy.random.normal(mu, sigma, N)` to draw random sample of the gaussian distribution of size N, write a function “draw_brownian” which takes 2 arguments (T, N) and returns a sample of N gaussian variables of mean 0 and variance T.

Question 15

Create a function “Spot_sample” which takes as argument an initial spot S_0 , a rate r , a volatility σ , a time T , and an integer N , and returns a numpy array sample of size N of the spot at time T in the Black-Scholes model. The function should re-use the function “draw_brownian” of the previous question. We remind that in the Black Scholes model we have:

$$S_t = S_0 e^{[r - 0.5 \sigma^2]t + \sigma * W_t}$$

Question 16

Create a function “Call_price” which takes as argument an initial spot S_0 , a rate r , a volatility σ , a time T , an integer N , and a strike K , which returns the price of a call option, re-using the previous functions “draw_brownian” and “Spot_sample”. We remind that the price of a call option is the expectation of the discounted payoff ($\text{Max}(\text{Spot}(T) - K, 0)$)

Question 17

What is the result of the below script ?

```
a=[1,2,3]
b=[1,2,5]
c=a+b
print(c)
```

Answers:

- A: [1,2,3,5]
- B: [1,2,3,1,2,5]
- C: [2, 4, 8]
- D: [1,4,15]

Question 18

What is creating the call of the function `numpy.linspace(10,25,5)` ?

Answers:

- A: an array starting at 10 and ending at 25 with a step of 5
- B: an array starting at 10 and ending at 24 with a step of 5
- C: an array of 5 numbers between 10 and 25 (equally spaced)
- D: an array of 10 number from 25 to 5 (equally spaced)

Question 19

What is the purpose of the below function ?

```
def grad(g, x0, epsilon, tolerance):  
    g_x0=g(x0)  
    error=abs(g_x0)  
    while error>tolerance:  
        g_x0=g(x0)  
        derivative=(g(x0+epsilon)-g(x0))/epsilon  
        x0=x0-g_x0/derivative  
        error=abs(g(x0))  
    return x0
```

Answers:

- A: Compute the derivative of a function
- B: Find the root a function using a dichotomy approach
- C: Find the root of a function using the Newton's method
- D: Find the minimum value of a function

Question 20

What is the output of the below python script ?

```
x=[i**2 for i in range(4)]  
y=set(x)  
y[0]=10  
print(y)
```

Answers:

- A=[10,1,4,9]
- B=[0,1,4,9,16]
- C=[10,1,4,9,16]
- D: An error message

Question 21

Which element(s) of the syntax below is/are generating syntax issue ?

```
def sum_odd_number(N):  
    tempsum=0  
    for i in range(0,N)  
        If i%2=0:  
            tempsum+=i  
    end for  
    Return tempsum
```

Answers:

- A: The indentation
- B: the increment of the tempsum variable
- C: the condition on the remainder %
- D: the structure of the for loop

Question 22

What is the output of the below python script ?

```
def my_function(arg_list):  
    list1=[abs(x) for x in arg_list]  
    value1=max(list1)  
    return value1  
  
print(my_function([-10, 200, -1.5, -350, 1e3]))
```

- A=200
- B=1000
- C=350
- D: An error message

Part 2: Python packages and general knowledge

Question 23

To manipulate matrices and arrays, which of the following packages should be used ?

- A: Numpy
- B: Pandas
- C: Bokeh
- D: Matplotlib

Question 24

If M is a Numpy array, which property allows to know its dimensions ?

- A: M.id
- B: M.dimensions
- C: M.shape
- D: M.len

Question 25

What is JSON ?

- A: a file format that uses text to transmit data objects consisting of attribute-value pairs
- B: a python package to encode/decode any text file
- C: a python package to store and transport data
- D: a function from the pandas package to read excel files

Question 26

Which method should be used to reshape a dataframe moving a column index to become a row index ?

- A: reshape()
- B: pivot_table()
- C: stack()
- D: unstack()

Question 27

Which python object is used to store a collection of keys and values ?

- A: a list
- B: a tuple
- C: a dataframe
- D: a dictionary

Question 28

A dictionary is a mutable object

- A: True
- B: False

Question 29

A set is an immutable object

- A: True
- B: False

Question 30

Which of the below statements are true regarding mutable objects ?

- A: A mutable object can be changed after it has been created
- B: A mutable object can contains immutable objects
- C: Mutable objects can be changed within functions (similar behavior as call by reference)
- D: An object can be both mutable and immutable

Question 31

How is a "For" loop ended in python ?

- A: with an "End For" statement
- B: with an "End" statement
- C: with a brace }
- D: by reducing the indentation

Part 3: Finance application

Detecting option prices with arbitrage opportunities.

The detection of option prices with arbitrage opportunities can be used to confirm the “coherence” of the data used to calibrate a volatility model. In the below example, we are going to prepare a script in order to detect those “abnormal” option prices which could trigger calibration issues.

We will focus on this exercise on the arbitrage opportunities resulting from the convexity of option prices. More precisely, if T is a maturity, K a strike, $C(K,T)$ the price of a call and $P(K,T)$ the price of a put, the non-arbitrage condition is translated into:

- If $K_1 < K_2$ then $C(K_1, T) > C(K_2, T)$
- If $K_1 < K_2$ then $P(K_1, T) < P(K_2, T)$

Question 32

Using the Pandas package and the attached file “goog_options.csv”, write a function which loads the CSV file as a dataframe and then adds a “Mid” column (average between “Bid” and “Ask”)

Question 33

Using the result of the previous question, write a python script which scroll all option “Mid” prices of the excel file and detects options with an arbitrage opportunity (i.e. not following the above non-arbitrage property). We advise to use a “for” loop on each maturity of the excel file and a “for” loop for each option type (“call” / “put”). The output of the script should be the corresponding list of contract “GOOG***” not following the non-arbitrage condition.

Question 34

Across all strikes and maturities of the excel file, how many contracts show “arbitrage” opportunities?

- A: 5
- B: 9
- C: 12
- D: 17