Mock Exam

REMARKS

- The exam will be a combination of questions on
  - Using Python more generally
  - Financial theory
  - Applying Python to a financial application
- The balance regarding the number of questions will be similar during the real exam.

Handing in the exam

- Do not forget to save and hand in this file in TestVision Online.

Instructions

- Answers can be worked out in this notebook or on paper. Both will be graded together.
- Efficient code is required for full marks.
- Make sure that any function you write has a docstring, and comments where appropriate.
- Before submitting your work, **click on `Kernel -> Restart & Run All`** and verify that your notebook produces the desired results and does not error.
- Some questions require you to write code to obtain a numerical result (e.g., an option price). In that case, don't just give the function, but also the result of calling it with the given parameter values (i.e., the numerial value that it returns). If your function uses random numbers, then set the seed to 0 before calling it. This makes it much easier to grade the exam (at least as long as the answer is correct).

Question 1

1. Write a docstring for the function `f(z)` , that takes the list `z` input, defined in the cell below. In other words, what does the function do?

In [1]:

```
def f(z):
    '''

    '''

    j = 0

    while z[j] <= 0:
        j = j + 1

    return j
```

1. We define three arrays `a`, `b` and `c`. Please do the following:

   - Select the last three elements in `a` without using that you know that `a` is of length 6.
   - Select the elements in `a` corresponding to the elements in `b` that are equal to 0.
   - Compute a matrix `d` where each element `d[i, j] = a[i] + b[j]` without using a `for`-loop.

In [ ]:

```
import numpy as np
a = np.array([1, 2, 3, 4, 5, 6])
b = np.array([0, -2, 0, -3, 4, 0])
c = np.array([5, 10, -2])
```

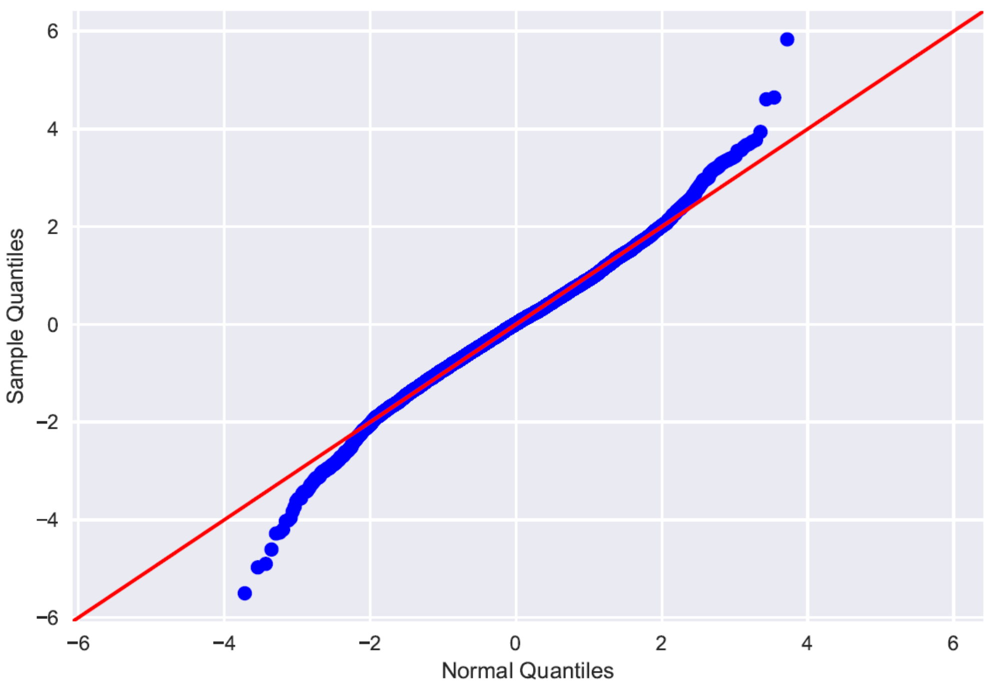
In [ ]:

```
```

Question 2

You are given a set of N returns $\{R_t\}_{t=T-N+1}^{T}$.

1. Suppose that $N = 10$ and $\{R_t\} = \{-0.07, 0.87, 1.13, -0.48, 0.22, 0.94, -0.62, -0.01, 2.06, -0.08\}$. What is the historical 20\% VaR?
2. You gather some more returns and produce the graph in the figure below. Interpret it.



In [ ]:

```
```

Question 3

Let

$$dX_t = \nu dt + \sigma dW_t, \quad \nu = -\frac{1}{2}\sigma^2,$$

a Brownian motion with drift.

Derive the SDE satisfied by $Y_t \equiv f(t, X_t) = tX_t$, and state whether Y_t is a martingale.

You can work out the answer on paper or in this notebook. You only get full points for a complete derivation.

In [ ]:

Question 4

Suppose that we are in a world where there is stock, a bond and a shout option. The shout option is an option where the holder can "shout" at one time during the lifetime of the option. If the holder shouts, then the payoff will be the either the payoff of the call option using the final value, or the value at time of shouting. At maturity, the payoff is then the largest of (i) the payoff of the usual option $\max(0, S_T - K)$ or (ii) the payoff with the price at the time of shouting $\max(0, S_\tau - K)$, where S_τ is the price at the time of shouting.

1. Show that *if the holder shouts* at time τ, the payoff at time T can be written as:
$$\max(0, S_T - S_\tau) - (S_\tau - K) \tag{†}$$
2. Use the result in equation (†) to write a function `shouttree(S0, K, T, r, sigma, N)` , which will return the estimate of the price of the shout option, where τ can be any period in the lifetime of the option. Use the following parameter values: $S_0 = 12$, $K = 11$, $T = 10$, $r = 0.03$, $\sigma = 0.4$, $N = 500$.
3. What if we restrict the period during which the holder can shout. Explain how this will impact its value.

In [ ]: