

(a) > x = np.random.permutation(1000)

x has rank 1 and shape (1000,) with numbers from 0 or 999 in random sequence.

(b) > a = np.array([[1,2,3],[4,5,6],[7,8,9]])

b = a[2,:]

Ans. b is rank 1 array and shape is (3,) containing row 3 of a i.e [7,8,9].

(c) > a = np.array([[1,2,3],[4,5,6],[7,8,9]])

> b = a.reshape(-1)

Ans. b is rank 1 array with shape (9,) with 9 elements. '-1' is telling to just flatten array.

(d) > f = np.random.randn(5,1)

> g = f[f>0]

f is rank 2 and (5,1) shape array with random values belonging to univariate gaussian distribution with mean 0 and variance 1. g is single dimensional array containing only positive values present in f array.

(e) > x = np.zeros(10)+0.5

> y = 0.5*np.ones(len(x))

> z = x + y

x is rank 1 array of shape (10,) with all elements equal to 0.5. This is due to broadcasting. y is rank 1 array with shape (10,) with all elements equal to 0.5. z will therefore be rank array of shape (10,) array with all elements 1.

(f) > a = np.arange(1,100)

> b = a[::-1]

a is rank 1 array with shape (99,) with elements from 1 to 99. b is reverse of a.

(a) Use numpy.random.rand to write a function that returns the roll of a six-sided die over N trials.

```
import numpy as np
```

```
def dieroll(N):
```

```
    output = []
```

```
    output.append((np.random.rand(N)* 5)+1)
```

```
    output = np.round(output, 0)
```

```
    return output;
```

(b) Let y be the vector: y = np.array([1, 2, 3, 4, 5, 6]). Use the reshape command to form a

new matrix z that looks like this: [[1,2],[3,4],[5,6]]

```
y = np.array([1, 2, 3, 4, 5, 6])
```

```
z = np.reshape(y, (3,2))
```

(c) Use the numpy.max and numpy.where functions to set x to the maximum value that occurs in z

(above), and set r to the row it occurs in and c to the column it occurs in.

```
x = np.arange(9.).reshape(3, 3)
```

```
max = np.amax(x)
```

-2
r and c
should
contain the rows and columns

```
r,c = np.where(x == max)
```

(d) Let v be the vector: `v = np.array([1, 8, 8, 2, 1, 3, 9, 8])`. Set a new variable x to be

the number of 1's in the vector v.

```
v = np.array([1, 8, 8, 2, 1, 3, 9, 8])
```

```
x = 0
```

```
for i in np.nditer(v):
```

```
    if i == 1:
```

```
        x = x + 1
```

-5 output not copied into the answer sheet

-2 need to plot histogram of vectorized A

-2 output intensity should be red and black not blue

-5 PS0Q2 gives error: Correct input file not named correctly

IOError: [Errno 2] No such file or directory: 'matrix.png'.