1 Concepts and coding

(a)

What's the output when calling the following commands, and why? Write one sentence for each question.

What's the output when calling the following commands, and why? Write one sentence for each question.

1. `np.mean([1, 2, 3])`,
2. `np.array([1, 2, 3])+1`,
3. `np.zeros(10)`.

Solution:

1. The output is 2, since 2 is the mean of $[1, 2, 3]$ and `np.mean` returns the mean of an array.
2. The output is $[2, 3, 4]$, as `np.array([1, 2, 3])+1` = `np.array([1, 2, 3])+np.array([1, 1, 1])` since the operation is vectorized.
3. The function `np.zeros(n)` returns an array of n zeroes, hence `np.zeros(10)` $= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$.

(b)

Consider the code `np.array([1, 2, 3]) + [1, 2, 3]`.

1. What happens when you run this code? Write at most two sentences.
2. Name three other operations that behaves like `"+"` does above.

Solution:

1. It does pointwise addition of the array and list and returns an array, so that

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

2. Standard mathematical operations such as $*, -, /$, and $**$.

(c)

1. Why does `np.random.default_rng(seed = 313).random(10)` return the same vector every time you run it?
2. How can you modify to return a different vector?

Solution:

1. This happens because the `np.random.default_rng(seed = 313)` object initializes a new random state every time it is created.
2. You must supply another seed argument, e.g. `np.random.default_rng(seed = 13).random(10)`.

(d)

Consider the following function.

```
def fun(n):
x = np.zeros(n)
x[0] = 1
for i in range(n - 1):
x[i + 1] = 1 + x[i]
return(x)
```

1. What does this function do? Explain using two sentences.
2. `numpy` contains a function that does exactly the same. What is its name?

Solution:

1. The function takes a positive integer n and returns the vector $[1, 2, 3, \ldots, n]$.
2. That function's name was supposed to be `np.arange`, but then $x[0] = 1$ shouldn't have been included, hence you must write `np.arange(n)+1` instead. (This is a mistake in the mock exam; the final exam is far more thoroughly checked.)

2 Random walks

We'll take a close look at some code from one of the videos.

```
    import numpy as np
n_reps = 10 ** 5
n = 100
rng = np.random.default_rng(seed = 313)
```

We will study a function from video 2 on random walks.

```
def fun(n, rng, n_reps = 10 ** 5):

     bs = np.apply_along_axis(
         func1d = np.cumsum,
         axis = 1,
         arr = rng.integers(low = -1, high = 2, size = (n_reps, n)))

     results = np.apply_along_axis(
         func1d = np.max,
         axis = 1,
         arr = bs)

     uniques, counts = np.unique(results, return_counts = True)
     return {"result" : uniques, "probability" : counts / n_reps}
```

Use at most two sentences for each of the following questions.

(a)

What is the rôle of `n` and `n_reps`? Write one sentence for each.

Solution:

`n` is the number of steps of the random walk. `n_reps` is the number of times we simulate a random walk.

(b)

What is the rôle of `func1d`, and what does `np.max` do?

Solution:

The function in `func1d` is applied to every row (since `axis = 1`) of the array supplied to `arr` . `np.max` returns the maximal element of an array.

(c)

What is the data type of `bs`? What do you call each row of `bs`?

Solution:

`bs` is a numpy array. The rows of `bs` are random walks.

(d)

Explain what `results` is.

Solution:

`results` is an array of `n_reps` simulated maximums of random walks with n steps.

(e)

What does `np.unique` do in this function? Remember the `return_counts` argument!

Solution

`np.unique` usually returns all the unique elements in `results`, but since `return_counts` is true, it returns the counts of each element as well. These are returned as a tuple of the form `(uniques, counts)`.

(f)

What is returned by the function?

Solution:

The function returns a dictionary where the keys are the unique simulated maximums and the values are the probabilities of attaining that maximum.