

Python for Math and Stat Fall 2023

Final Exam

Assume that all necessary packages have been imported.

1. (15 pts) For the following 4 problems, write down what each code block would display if executed in a Jupyter cell. If the code generates an error or infinite loop, write Error. Assume

```
arr = np.array([[10, 1, 9, 7, 12],  
               [ 3, 16, 5, 8, 9]])
```

(a) `arr[1, 4:0:-2]`

(b) `arr[arr > 9]`

(c) `kvals = {k: k*`

3. (10 pts) To check for errors when scanning or manually entering product bar codes, an extra check digit is included.



Here is a procedure for calculating the check digit for an integer code: (ex: 15694)

- Add every other digit in the code, starting with the first digit. (ex: 15694: $1 + 6 + 4 = 11$)
- Add every other digit in the code, starting with the second digit. (ex: 15694: $5 + 9 = 14$)
- Add the second sum to 3 times the first sum. (ex: $14 + 3 \cdot 11 = 47$)
- The units digit of the result is the check digit. (ex: 7)

Write a function `check_digit(code)` that takes an integer `code` greater than 9 and returns its check digit as an `int`. For example, `check_digit(15694)` returns 7.

4. (10 pts) Consider the polynomial

$$P(x) = 1 + 2x + 3x^2 + 4x^3 + \dots + nx^{n-1}$$

Write a function `poly_eval(x, n)` that calculates the value of $P(x)$ given values for x and positive integer n . Use `numpy` features (such as `arange` and vectorization). *Do not* include a loop.

Example: `poly_eval(2, 3)` returns 17 which equals $1 + 2(2) + 3(2)^2$.

5. (12 pts) The DataFrame `dfcocoa`, shown below, contains information about various cocoa powder products. Each row provides the name, weight (in ounces), and price (in dollars) for a distinct product.

| Product | Ounces | Price |
|-------------|--------|-------|
| Droste | 8.8 | 9.67 |
| Anthony's | 22.0 | 17.00 |
| Valrhona | 3.5 | 15.43 |
| Chocolatier | 7.0 | 20.7 |

Write code to do the following:

- Add a new Nestle product to the DataFrame with a weight of 8 ounces and a price of 2.75 dollars.
- Add a new column to the DataFrame called `UnitPrice` which equals the price per ounce for each product.
- Select the names of all products with a `UnitPrice` greater than the unit price for Hersheys (which is a product in `dfcocoa`). The result should be a pandas index or a list of strings.
- One of the products has the lowest unit price. Identify the name of that product as a string.

6. (20 pts) Create a class called `Coin`. Each instance of the class represents a coin with one attribute:

- `prob_H`: probability of flipping a head. Assume that `prob_H` is a value between 0 and 1. Set the **default** value to 0.5.

and these methods:

- `flip()`: returns 'H' or 'T' given probability `prob_H`. (For example, if `prob_H` equals 0.2, then out of 100 flips, 'H' will appear about 20 times.)
- `flip_until(outcome)`: simulates the flipping of the coin, printing the results in a row, until the desired outcome appears. Return the number of flips. Assume that `outcome` is either 'H' or 'T'. This method should call `flip()`.

Example: `flip_until('T')` might print HHHHT and return 5.