Homework 1 Mark Holtje

September 17, 2019

0.1 Homework 1

0.1.1 Due Tuesday September 17, 11:59 PM

0.1.2 Collaboration Policy

While you may talk with others about the homework, we ask that you **write your solutions individually**. If you do discuss the assignments with others please **include their names** at the top of your notebook.

Collaborators: *list collaborators here*

0.1.3 Goal

Homework 1 will help prepare you for class along with the other assignments. The problems will review some of the programming that we will need throughout the semester. Please check the references for more information on unfamiliar material. Reach out to us in section and office hours for help.

Tips To learn about keyboard shortcuts, go to **Help -> Keyboard Shortcuts** in the menu above. Get help for a function by running a cell with the function name and a `?` at the end...you can escape by hitting `esc` several times. Alternatively type the function name, then - on your keyboard...you can press multiple times to show additional information.

0.1.4 Score Breakdown

Question	Points
1	2
2a	1
2b	1
2c	2
2d	2
3a	3
3b	3
4a	2
4b	2
4c	3
4d	3
Total	24

0.1.5 Question 1

Recall that summation (or sigma notation) is a way of expressing a long sum in a concise way. Let $a_1, a_2, ..., a_n \in \mathbb{R}$ and $x_1, x_2, ..., x_n \in \mathbb{R}$ be collections of real numbers. When you see x_i, you can think of the i as an index for the i^{th} x. For example x_2 is the second x value in the list $x_1, x_2, ..., x_n$. We define sigma notation as follows:

$$\sum_{i=1}^{n} a_i x_i = a_1 x_1 + a_2 x_2 + ... + a_n x_n$$

We commonly use sigma notation to compactly write the definition of the arithmetic mean (commonly known as the `average`):

$$\bar{x} = \frac{1}{n}\left(x_1 + x_2 + ... + x_n\right) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Recall the formula for population variance below:

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$

Complete the functions below to compute the population variance of `population`, an array of numbers. For this question, do not use built in NumPy functions; we will use NumPy to verify your code.

[3]:

```python
import numpy as np
```

[4]:

```python
def mean(population):
    """
    Returns the mean of population (mu)

    Keyword arguments:
    population -- a numpy array of numbers
    """
    sum_people = 0
    for i in range(len(population)):
        sum_people += population[i]
    return sum_people/(len(population))

    # Calculate the mean of a population
    # BEGIN SOLUTION
    # TODO: solve me
    # END SOLUTION

def variance(population):
    """
    Returns the variance of population (sigma squared)

    Keyword arguments:
    population -- a numpy array of numbers
    """
    variance_sum = 0
    for i in range(len(population)):
        variance_sum += ((i-(mean(population)))**2)
    variance = variance_sum/len(population)
    return variance
    # Calculate the variance of a population
    # BEGIN SOLUTION
    # TODO: solve me
    # END SOLUTION
```

Run the following cell to test your function against numpy

[5]:

```
# TEST
population_0 = np.random.randn(100)
np.isclose(mean(population_0), np.mean(population_0), atol=1e-6)
```

[5]:

```
True
```

[6]:

```
# TEST
population_0 = np.random.randn(100)
np.isclose(variance(population_0), np.var(population_0), atol=1e-6)
```

[6]:

```
False
```

0.1.6 Question 2

Suppose you're trying to meet your friend in either Central Park (denoted C) or Riverside Park (denoted R).

Right now, you know that the chances of your friend being at z are

	Chance
z = Central Park	0.6
z = Riverside Park	0.4

a. Use numpy to create a 2×1 array called v with chance of being in Central Park and Riverside Park. Try testing the dimensions.

[7]:

```
# BEGIN SOLUTION
# TODO: solve me
# END SOLUTION
v = np.array([[.6],[.4]])
print(v)
# TEST
v.shape
```

```
[[0.6]
 [0.4]]
```

[7]:

```
(2, 1)
```

Since your friend like to wander around, you know that after 1 hour the chances of her walking from x to y are

	x = Central Park	x= Riverside Park
y = Central Park	0.6	0.7
y = Riverside Park	0.4	0.3

b. Use `numpy` to create a 2 × 2 matrix `M` with chance of walking between parks. Try testing the dimensions.

[8]:
```
# BEGIN SOLUTION
# TODO: solve me
# END SOLUTION
M = np.array([[.6,.7],[.4,.3]])
print(M.shape)
print(M)
```

```
(2, 2)
[[0.6 0.7]
 [0.4 0.3]]
```

c. Use the `numpy` sum method to sum the entries over columns. Do the entries sum to 1 -- why?

Use the `numpy` sum method to sum the entries over rows. Do the entries sum to 1 -- why?

[9]:
```
# BEGIN SOLUTION
# TODO: solve me
print(np.sum(a=M,axis=0))
print(np.sum(a=M,axis=1))
# END SOLUTION
```

```
[1. 1.]
[1.3 0.7]
```

Explain Why: x is independent of y. After you leave the first park, you can either 1)walk around the same park or 2) walk around the other park. In this case, you cannot come from two parks at the same time (therefore, it is mutually exclusive). So you add the probabilities in terms of columns, not rows. Ex: 0.4 or 40% chance you first walk around Central Park, then walk around Riverside Park.

d. Use the Law of Total Probability to determine the likelihood that your friend is in Central Park or Riverside Park after 1 hour. P(1 hr) = P(C1 | C0)P(CO)+P(C1 | RO)P(RO)

pr(.5.6+.5.4)+(.5.7+.5.3) = 1
Hint: Use the @ symbol for multiplication

[18]:
```
# BEGIN SOLUTION
# TODO: solve me
v= np.matrix('0.6; 0.4')
```

```
M = np.matrix("0.6 0.7 ; 0.4 0.3")
M@v
# END SOLUTION
```

[18]:

```
matrix([[0.64],
        [0.36]])
```

0.1.7 Question 3

a. Debugging is jargon for inspecting the code in a program line by line for errors. Try inspecting the following code -- locate, document and correct each error.

[13]:

```
# Write a program that will average 3 numeric exam grades, return an average␣
 ↪test score, a corresponding letter grade, and a message stating whether the␣
 ↪student is passing.

# Average        Grade
# 90+        A
# 80-89        B
# 70-79        C
# 60-69        D
# 0-59        F

exam_one = int(input("Input exam grade one: "))

exam_two = int(input("Input exam grade two: ")) #error made: the exam score␣
 ↪needs to be an int

exam_three = int(input("Input exam grade three: ")) #error made: exam score␣
 ↪needs to be an int, not str. Rename exam_3 to exam_three

grades = [exam_one, exam_two, exam_three]  #Error made: Needs commas in netween␣
 ↪the variables.
sum_grades = 0 #error_made: when you write sum, it expects to sum a series of␣
 ↪numbers. Rename variable.
for grade in grades: #error made: This for loop needs to loop over the list␣
 ↪entitled grades, not grade
  sum_grades += grade #error made: rename sum to sum_grades

avg = sum_grades / len(grades) #Error: Rename sum to sum_grades and grdes to␣
 ↪grades

if avg >= 90:
    letter_grade = "A"
elif avg >= 80 and avg < 90:   #Error: You must put a colon after 90.
    letter_grade = "B"
elif avg > 69 and avg < 80:
    letter_grade = "C"          #error: C must be wrapped with a second␣
 ↪apostrophe to be a string
elif avg <= 69 and avg >= 65:
    letter_grade = "D"
else:                             #Error: else not elif
    letter_grade = "F"

for grade in grades:
    print("Exam: " + str(grade))

    print("Average: " + str(avg))

    print("Grade: " + str(letter_grade)) #Error: letter_grade has to be a str.

if letter_grade is "F":  #Error: rename letter-grade to letter_grade, since it␣
 ↪needs to be a variable
    print("Student is failing.")                #Error: Missing parenthesis
else:
    print("Student is passing.")               #Error: Missing parenthesis
```

```
Input exam grade one: 89
Input exam grade two: 90
Input exam grade three: 90
Exam: 89
Average: 89.66666666666667
Grade: B
Exam: 90
Average: 89.66666666666667
Grade: B
Exam: 90
Average: 89.66666666666667
Grade: B
Student is passing.
```

[14]:

```
# TEST 1
# Exams: 89, 90, 90
# Average: 90
# Grade: A
# Student is passing.


# TEST 2
# Exams: 50, 51, 0
# Average: 33
# Grade: F
# Student is failing.
```

b. We can use debugging tool to explore the state of a running program without using print() statements everywhere. In Jupyter notebooks, we can import the pdb package for the debugger.

[14]:
```
import pdb
```

We can use the debugger in two ways:

1. After the code threw an exception, we work **backwards** from the problem.

- Run `pdb.pm()` to start the debugger after an exception.
- Use the `up` command to work backwards.
- Use `quit` to quit the debuggger.

2. We place a **break-point** where execution halts and we work **forwards** line by line.

- Place `pdb.set_trace()` in the line where execution should halt.
- Try using the following to work forwards.
    1. `list` - Displays 11 lines around the current line or continue the previous listing.
    2. `step` - Execute the current line, stop at the first possible occasion.
    3. `next` - Continue execution until the next line in the current function is reached or it returns.
    4. `break` - Set another breakpoint at a line (depending on the argument provided).
    5. `return` - Continue execution until the current function returns.
- For more information type `help`

Save the following script to a file called `fib.py`.

```
# fib.py
import sys

pdb.set_trace()
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)

for i in range(1,len(sys.argv)): #Change the range from 0 to len(sys.argv) to 1 to len(sys.arg
    arg = sys.argv[i]
    num = int(arg)          #Original Error: invalid literal for int() with base 10
    print("fib of",num," = ",(fib(num)))
```

Try executing the following in a cell

~~~
[1]: ```python
     %run fib.py 4 5 6
     ```
~~~

~~~
      File "<ipython-input-1-85e85a3e850b>", line 1
    ```python
    ^
SyntaxError: invalid syntax
~~~

What went wrong...use `pdb` to locate, document and correct the errors in `fib.py`.

0.1.8 Question 4

Consider the following scenario:

Only 1% of 40-year-old women who participate in a routine mammography test have breast cancer. 80% of women who have breast cancer will test positive, but 9.6% of women who don't have breast cancer will also get positive tests.

a. Fill in the conditional probability table

	x = Has Cancer	x= Does Not Have Cancer
Tested Positive given x	.8	.096
Tested Negative given x	.2	.904

SOLUTION:
---------------shown above

b. Fill in the joint probability table

	Has Cancer	Does Not Have Cancer
Tested Positive	.008	.09504
Tested Negative	.002	.89496

SOLUTION:
---------------shown above

c. Suppose we know that a woman of this age tested positive in a routine screening. What is the probability that she actually has breast cancer?

Hint: Use Bayes' rule.
SOLUTION:
---------------(.008)/(.008+.09504)=.077639 or 7.764%

d. Sometimes, 40-year-old men will also get mammographies, if they are suspected to be at high risk to breast cancer. The overall prevalence of breast cancer in men who participate in the test is lower (0.01%). As with women, 80% of men who have breast cancer will test positive. Due to physiological diferences, only 2.4% of men who don't have breast cancer will test positively.

For each of the following pairs of events A and B, please determine whether or not they are independent:

1. A = the subject's sex (man or woman), B = having breast cancer.
2. A = the subject's sex, B = testing positively for breast cancer given having breast cancer.
3. A = the subject's sex, B = testing positively for breast cancer.

Hint: Events A and B are independent if $P(B \mid A) = P(B)$
SOLUTION:

---

1. = the subject's sex (man or woman), = having breast cancer. These events are dependent, since with aged 40 men, the prevalence of breast cancer is lower (.01%), whereas the prevalence for women is 1%. So, given gender determines likelihood of having breast cancer.
2. = the subject's sex, = testing positively for breast cancer given having breast cancer. These events are independent, since with aged 40 men and aged 40 women, the prevalence of testing positive for breast cancer given having cancer is both at 80%. The probability of testing positive for breast cancer given having cancer is at 80% given that the patient is either male or female.
3. = the subject's sex, = testing positively for breast cancer. These events are dependent, since with aged 40 women, the prevalence of testing positive for breast cancer is 10.34% because .008+.09504=.01034. However for men, the prevalence of testing positive for breast cancer is (.0001 * .8) + (.9999 * .024) = 0.240776 or 2.40776%