

Optional Assignment

37 pts

In this assignment, you are practicing using the popular data libraries, NumPy, Pandas and Matplotlib.

```
In [22]: import pandas as pd

# display your name and student id as a table by creating a Pandas DataFrame
# that uses "Name" and "Student ID" as column names and your information
# as the values in the first row.
columns = ['Name', 'StudentID']
df = pd.DataFrame([['Yong Seung Rho', 'W0447442']], columns=columns)
df
```

```
Out[22]:
```

	Name	StudentID
0	Yong Seung Rho	W0447442

	Name	Student ID
0	John Smith	W0123456

Part A - NumPy (12pts)

For this part of the assignment, you will answer each question using the NumPy library.

```
In [23]: # a student in the Logic and Programming I course obtained the following grades
# for the first five assignments:
# Assignment 1: 26/35, Assignment 2: 23/26, Assignment 3: 21/26,
# Assignment 4: 19/23, Assignment 5: 32/38
# the student wants to know what their final grade is in the course.

# create a NumPy array that contains the marks the student received and
# create a second NumPy array that contains the total grade for the assignments.
# then, use NumPy functions to calculate the average percentage
# for the assignments and print it out.

# 4 pts
import numpy as np

marks = np.array([26, 23, 21, 19, 32])
grades = np.array([35, 26, 26, 23, 38])

average = (marks / grades * 100).mean()
print('The average percentage is %1.1f' % average)
```

The average percentage is 82.1

```
In [24]: # generate an array of 100 random numbers between 1 and 100
# then, filter out all the numbers in the fifties without
# using a loop, then, sort the resulting numbers and print
# them out

# 4 pts
nums = np.random.randint(1, 100, 100)
nums = nums[(nums < 50) | (nums > 59)]
print(np.sort(nums))
```

[1 2 2 4 8 8 8 10 11 13 13 17 19 20 20 21 21 22 22 23 24 24 26 30
31 32 33 34 35 36 37 37 37 38 38 40 42 46 48 49 60 60 61 62 62 62 62 64
64 64 67 67 68 70 70 71 72 72 73 74 75 75 77 79 81 82 83 85 85 85 87 88
88 89 92 93 93 93 95 96 97 98 98 99 99 99 99]

```
In [25]: # create the following pattern as a NumPy array
# hint: slicing can make this easier!

# 4 pts
rows = 7
columns = 7
max = 4

pattern = np.zeros([rows,columns])
for i in range(max):
    pattern[i:rows - i, i:columns - i] = max - i

print(pattern)
```

```
[[4. 4. 4. 4. 4. 4. 4.]
 [4. 3. 3. 3. 3. 3. 4.]
 [4. 3. 2. 2. 2. 3. 4.]
 [4. 3. 2. 1. 2. 3. 4.]
 [4. 3. 2. 2. 2. 3. 4.]
 [4. 3. 3. 3. 3. 3. 4.]
 [4. 4. 4. 4. 4. 4. 4.]]
```

```
[[4. 4. 4. 4. 4. 4. 4.]
 [4. 3. 3. 3. 3. 3. 4.]
 [4. 3. 2. 2. 2. 3. 4.]
 [4. 3. 2. 1. 2. 3. 4.]
 [4. 3. 2. 2. 2. 3. 4.]
 [4. 3. 3. 3. 3. 3. 4.]
 [4. 4. 4. 4. 4. 4. 4.]]
```

Part B - Pandas (14pts)

For this part of the assignment, you will answer each question using the Pandas library.

```
In [26]: # create a new DataFrame that is contains the multiplication tables from 0 to 12
# then, locate the answer to "6 x 7" in the DataFrame and print it out
# hint: using a 2D NumPy array and 'np.outer' can be handy here!

# 3 pts
array1 = np.arange(13)
array2 = np.arange(13)
numpy_array = np.outer(array1, array2)

df = pd.DataFrame(data=numpy_array)
result = df.iloc[6, 7]
print(f'The result of 6 x 7 is {result}')
```

The result of 6 x 7 is 42

```
In [27]: # read the contents of 'medical.csv' into a dataframe
# use the 'Date' column as the index
# and display the first 4 records

# 3 pts
medical = pd.read_csv('medical.csv', index_col='Date')
medical.head()
medical.iloc[0:4]
```

Out[27]:

	Patient	Province	BillingCode	DiagnosisCode	Diagnosis	Doctor	ChartNum	Billed
Date								
15-Nov-2020	Gerlach, Antonia	Ontario	enh1	1365	Injury arm vessel NOS	Riviera, Nick	167480	\$122.19
16-Nov-2020	Casper, Annalisa	Ontario	03.03a	2742	Malig melanoma leg	Zoidberg, John	344216	\$107.93
16-Nov-2020	Kuphal, Gita	Nova Scotia	786.05	7967	3rd deg burn face NEC	Riviera, Nick	634145	\$61.72
18-Nov-2020	Harber, Gerry	Nova Scotia	pb002	204	Acute nephritis NEC	Jekyll, Henry	859568	\$127.83

```
In [28]: # display rows 8-10 and columns 5,7
# hint: remember that row and column numbers start at zero!

# 2 pts
medical = pd.read_csv('medical.csv')
medical.head()
medical.iloc[7:10, [4,6]]
```

Out[28]:

	DiagnosisCode	Doctor
7	3816	Jekyll, Henry
8	5341	Doofenshmirtz, Heinz
9	4899	Zoidberg, John

```

In [29]: # display all records for Nova Scotia patients
# but just the patient names and doctor names
# along with the date of the visit

# 3 pts
medical = pd.read_csv('medical.csv', index_col='Date')
medical.head()
medical.loc[medical['Province']=='Nova Scotia', ['Patient', 'Doctor']]

```

Out[29]:

	Patient	Doctor
Date		
16-Nov-2020	Kuphal,Gita	Riviera,Nick
18-Nov-2020	Harber,Gerry	Jekyll,Henry
18-Nov-2020	Murray,Tien	Jekyll,Henry
18-Nov-2020	Crist,Rheba	Jekyll,Henry
17-Nov-2020	White,Leonard	Riviera,Nick
...
17-Nov-2020	Schmitt,Vivan	Riviera,Nick
16-Nov-2020	Schaden,Mary	Doofenshmirtz,Heinz
16-Nov-2020	Ruecker,Linnie	Riviera,Nick
19-Nov-2020	Stehr,Stuart	Jekyll,Henry
16-Nov-2020	Satterfield,Shenika	Doofenshmirtz,Heinz

90 rows × 2 columns

```
In [30]: # find the last 5 patient records with a diagnosis code between 1000-2000
# hint: it helps to do this one in steps!

# 3 pts
medical = pd.read_csv('medical.csv')
medical.head()
diagnosis = medical[(medical['DiagnosisCode'] > 1000) & (medical['DiagnosisCode'] < 2000)]
diagnosis.iloc[-5:]
```

Out[30]:

	Date	Patient	Province	BillingCode	DiagnosisCode	Diagnosis	Doctor	ChartNum	Billed
938	17-Nov-2020	Kozey,Carlo	Ontario	786.05	1173	Adv eff ant pituitary	Riviera,Nick	317012	\$70.62
946	20-Nov-2020	Doyle,Chasity	Ontario	786.05	1672	Multiple contusion arm	Zoidberg,John	151676	\$29.43
947	16-Nov-2020	Durgan,Christiane	Ontario	pb003	1142	Open wound larynx-compl	Jekyll,Henry	213557	\$96.75
974	16-Nov-2020	Mraz,Theresa	British Columbia	595.9	1769	Oligospermia	Zoidberg,John	149350	\$66.61
977	15-Nov-2020	Lowe,Marty	Ontario	13.59I	1712	Fragile x syndrome	Jekyll,Henry	427587	\$126.47

Part C - Matplotlib (11pts)

For this part of the assignment, you will answer each question using the Matplotlib library.

The following are the results of a survey on favourite Christmas movies

3 pts

Create a pie chart to display the information and emphasize the most popular movie by exploding it out from the chart.

Movie Percentage

Movie	Percentage
It's a Wonderful Life	14%
A Christmas Story	12%
Home Alone	11%
National Lampoon's Christmas Vacation	11%
How the Grinch Stole Christmas	10%
Elf	9%
Miracle on 34th Street.	7%
A Charlie Brown Christmas	7%
Die Hard	5%
Love Actually	3%
Other	11%

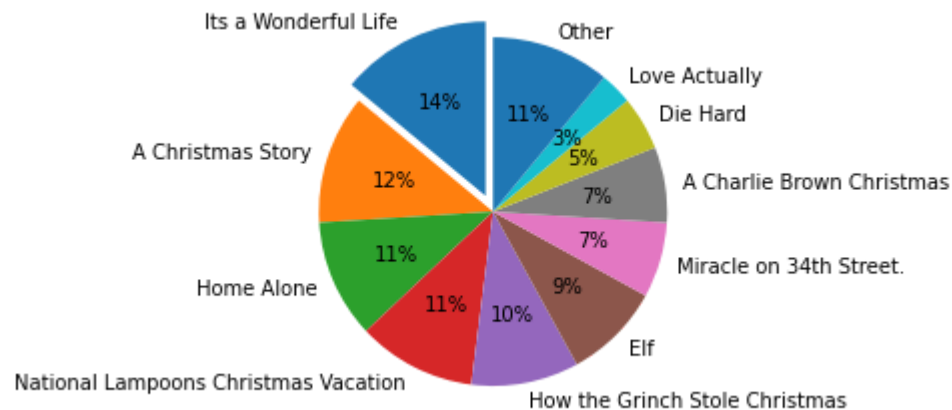

```

In [31]: import matplotlib.pyplot as plt

nums = [14, 12, 11, 11, 10, 9, 7, 7, 5, 3, 11]
labels = ['It's a Wonderful Life ',
          'A Christmas Story',
          'Home Alone',
          'National Lampoon's Christmas Vacation',
          'How the Grinch Stole Christmas',
          'Elf',
          'Miracle on 34th Street.',
          'A Charlie Brown Christmas',
          'Die Hard',
          'Love Actually',
          'Other']

# only "explode" the most popular movie
explode = (0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
plt.pie(nums, labels=labels, explode=explode, autopct='%1.0f%%', startangle=90)
plt.show()

```



The 'covid.csv' file contains information on Canada's reponse to COVID-19.

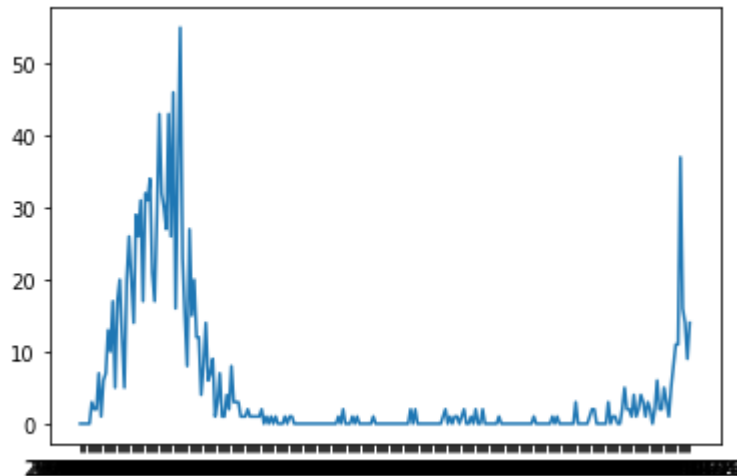
4 pts

Read the 'covid.csv' file into a Pandas DataFrame and view the data.
Extract the data related to Nova Scotia and the number of cases per day.
Create a line graph that displays number of cases each day in Nova Scotia.

```
In [32]: covid = pd.read_csv("covid.csv")
covid = covid.loc[covid['prname']=='Nova Scotia']

x = covid.date
y = covid.numtoday
plt.plot(x, y)

plt.show()
```



Rate Your NSCC Courses!

4 pts

Create a horizontal bar graph that lists all the courses you have taken in Fall 2020.

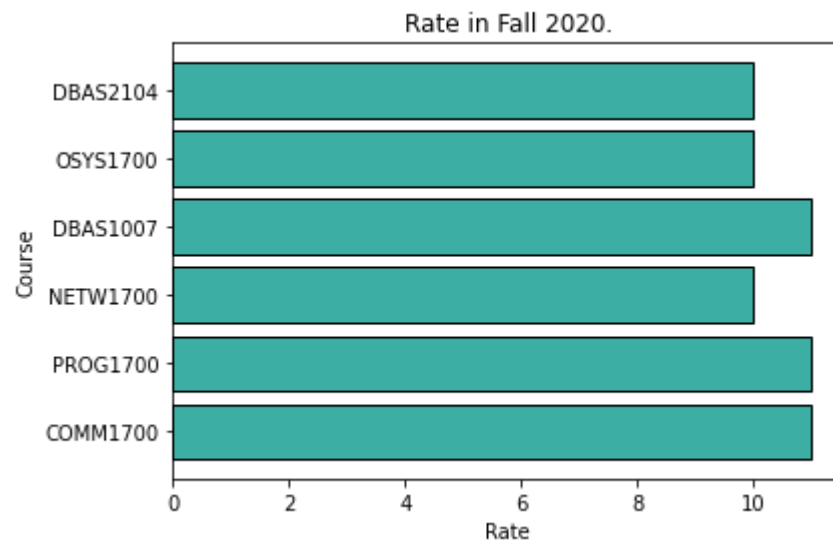
Rate each course based on a scale of 1-11, where 1=meh. and 11=super-amazeballs!

Give the graph an appropriate title, the X/Y axes appropriate labels and ensure all the rating levels (1-11) are displayed on the x-axis.

Warning: A rating below 11 for PROG1700 will result in this assignment self-destructing.

```
In [33]: y = [11, 11, 10, 11, 10, 10]
x = ['COMM1700', #/PPIT
      'PROG1700', #/Logic and Programming
      'NETW1700', #/Introduction to Networking and Security
      'DBAS1007', #/Data Fundamentals
      'OSYS1700', #/Introduction to Hardware and Operating Systems
      'DBAS2104', #/Business Analysis Essentials'
      ]

plt.barh(x,y, ec='black', fc='#3CAEA3')
plt.title('Rate in Fall 2020.')
plt.xlabel("Rate")
plt.ylabel("Course")
plt.show()
```



In []: