MODULE / WEEK 02

QMB4400

DATA ANALYSIS AND OPTIMIZATION

RASMUSSEN UNIVERSITY

PYTHON FEATURES

WHY PYTHON?

- Interpreted
- Intuitive and minimalistic code
- Expressive language
- Dynamically typed
- Automatic memory management

PYTHON FEATURES

Advantages

- Ease of programming
- Minimizes the time to develop and maintain code
- Modular and object-oriented
- Large community of users
- A large standard and user-contributed library

Disadvantages

- Interpreted and therefore slower than compiled languages such as C, C++, Java
- Decentralized with packages

PYTHON FEATURES

- Increasingly being used as a scientific language
- Matrix & vector manipulations – extremely important
- NumPy and Pandas emerged to be essential libraries
  - Due to their intuitive syntax & high-performance matrix computation capabilities

PYTHON LIBRARIES FOR DATA SCIENCE

By far, the most commonly used packages are those in the SciPy stack. We will focus on these in this class. These packages include:

- NumPy
- SciPy
- Matplotlib – plotting library.
- IPython – interactive computing.
- Pandas – data analysis library.
- SymPy – symbolic computation library.

NUMPY

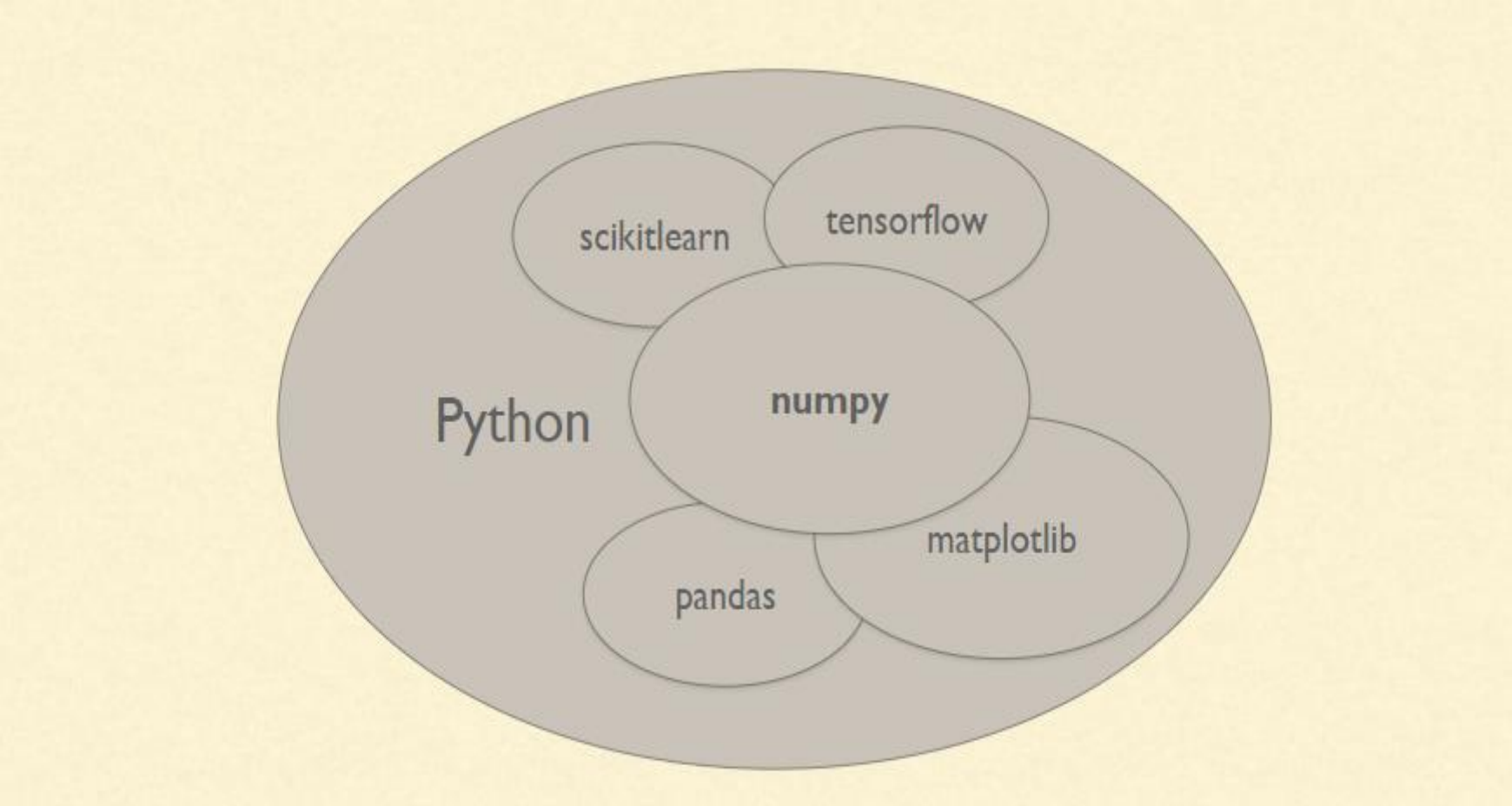
NumPy:

- Numerical Python or Numeric Python
- Open Source Module
- Fast mathematical computation on arrays and matrices
- Part of the Python Machine Learning Ecosystem

Link: http://www.numpy.org/

RASMUSSEN UNIVERSITY

NUMPY



RASMUSSEN UNIVERSITY

NUMPY

NumPy:

- Provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation
- provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance
- many other python libraries are built on NumPy

Link: http://www.numpy.org/

RASMUSSEN UNIVERSITY

NUMPY

Here are the top four benefits that NumPy can bring to your code:

1. **More speed:** NumPy uses algorithms written in C that complete in nanoseconds rather than seconds.
2. **Fewer loops:** NumPy helps you to reduce loops and keep from getting tangled up in iteration indices.
3. **Clearer code:** Without loops, your code will look more like the equations you're trying to calculate.
4. **Better quality:** There are thousands of contributors working to keep NumPy fast, friendly, and bug free.

Because of these benefits, NumPy is the de facto standard for multidimensional arrays in Python data science, and many of the most popular libraries are built on top of it. Learning NumPy is a great way to set down a solid foundation as you expand your knowledge into more specific areas of data science.

NUMPY

NumPy:

- Main object – homogeneous multidimensional array
- Is a table with same type elements
  - Integers
  - Strings
  - Characters
- Dimensions - axes
  - Number of axes = Rank
- Several ways to create an array like np.array, np.zeros, no.ones, etc. – each provides some flexibility

RASMUSSEN UNIVERSITY

Commands to Create an Array

Command to create an array	Example
np.array	

```
>>> a = np.array([1, 2, 3])
>>> type(a)
<type 'numpy.ndarray'>

>>> b = np.array((3, 4, 5))
>>> type(b)
<type 'numpy.ndarray'>
```

np.ones	

```
>>> np.ones( (3,4), dtype=np.int16 )
array([[ 1,  1,  1,  1],
       [ 1,  1,  1,  1],
       [ 1,  1,  1,  1]])
```

np.full	

```
>>> np.full( (3,4), 0.11 )
array([[ 0.11,  0.11,  0.11,  0.11],
 [ 0.11,  0.11,  0.11,  0.11],
 [ 0.11,  0.11,  0.11,  0.11]])
```

Commands to Create an Array

Command to create an array	Example
np.arange	
	1 >>> np.arange(10, 30, 5) 2 array([10, 15, 20, 25]) 3 4 >>> np.arange(0, 2, 0.3) 5 # it accepts float arguments 6 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8])
np.linspace	
	1 >>> np.linspace(0, 5/3, 6) 2 array([0. , 0.33333333 , 0.66666667 , 1. , 1.33333333 1.66666667])
np.random.rand(2,3)	
	1 >>> np.random.rand(2,3) 2 array([[0.55365951, 0.60150511, 0.36113117], 3 [0.5388662 , 0.06929014, 0.07908068]])
np.empty((2,3))	
	1 >>> np.empty((2,3)) 2 array([[0.21288689, 0.20662218, 0.78018623], 3 [0.35294004, 0.07347101, 0.54552084]])

RASMUSSEN UNIVERSITY

Important Attributes of a NumPy object

Ndim: displays the dimension of the array

Shape: returns a tuple of integers indicating the size of the array

Size: returns the total number of elements in the NumPy array

Dtype: returns the type of elements in the array, i.e., int64, character

Itemsize: returns the size in bytes of each item

Reshape: Reshapes the NumPy array

NumPy Array Elements Accessed Using Indexing

```
import numpy as np
a = np.array([2,4,6])
Print (a)


[2  4  6]
```

The array above contains three values: 2, 4 and 6. Each of these values has a different index.

Index (or location)	Value
0	2
1	4
2	6

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

```
import numpy as np
a = np.array([2,4,6])
Print (a)
```

```
[2  4  6]
```

The array above contains three values: 2, 4 and 6. Each of these values has a different index.

Index (or location)	Value
0	2
1	4
2	6

Individual values stored in an array can be accessed with indexing.
The general form to index a NumPy array is below:
<value> = <array>[index] Where <value> is the value stored in the array, <array> is the array object name
and [index] specifies the index or location of that value.
In the array above, the value 6 is stored at index 2.

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

```
import numpy as np
a = np.array([2,4,6)
(a) value = a[2]
(value)
```

```
[2 4 6]
6
```

Multi-dimensional Array Indexing:

Multi-dimensional arrays can be indexed as well. A simple 2-D array is defined by a list of lists.

```
import numpy as np
a = np.array([[2,3,4],[6,7,8]])
print(a)
```

```
[[2 3 4]
 [6 7 8]]
```

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

Values in a 2-D array can be accessed using the general notation below:
`<value> = <array>[row,col]`

Where `<value>` is the value pulled out of the 2-D array and `[row,col]` specifies the row and column index of the value.

Remember Python counting starts at `0`, so the first row is row zero and the first column is column zero. We can access the value 8 in the array above by calling the row and column index `[1,2]`.

This corresponds to the 2nd row (remember row 0 is the first row) and the 3rd column (column 0 is the first column).

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

```
import numpy as np
a = np.array([[2,3,4],[6,7,8]])
(a) value = a[1,2]
(value)
```

```
[[2 3 4]
[6 7 8]]
8
```

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

Array indexing is used to *access* values in an array. And array indexing can also be used for *assigning* values of an array.

The general form used to assign a value to a particular index or location in an array is : `<array>[index] = <value>`

Where `<value>` is the new value going into the array and `[index]` is the location the new value will occupy.

The code below puts the value `10` into the second index or location of the array `a`.

```
import numpy as np
a = np.array([2,4,6]) a[2] = 10
(a)


[ 2 4 10]
```

RASMUSSEN UNIVERSITY

NumPy Array Elements Accessed Using Indexing

Values can also be assigned to a particular location in a 2-D arrays using the form: `<array>[row,col] = <value>`

The code example below shows the value `20` assigned to the 2nd row (index 1) and 3rd column (index 2) of the array.

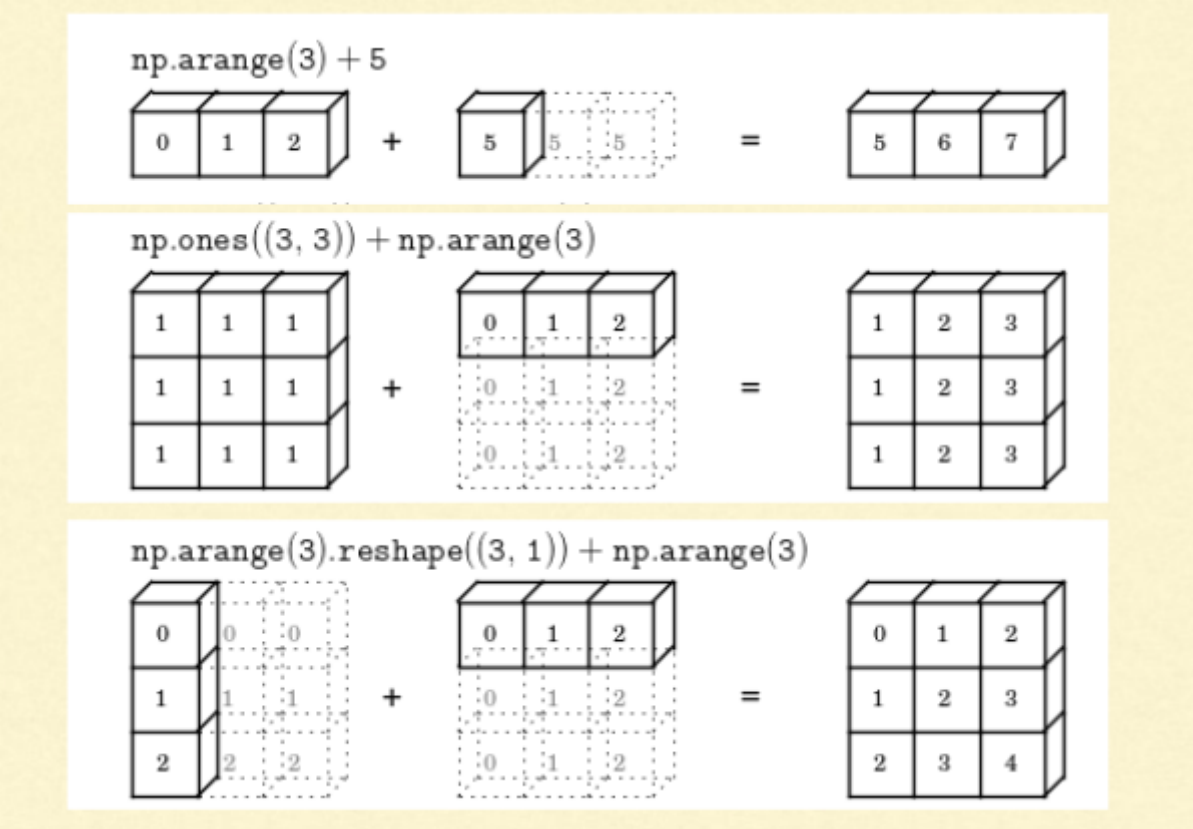
```
import numpy as np
a = np.array([[2,3,4],[6,7,8]])
print(a)

a[1,2]=20
print(a)
```

```
Output:
[[2 3 4]
 [6 7 8]]
[[ 2 3 4]
 [ 6 7 20]]
```

RASMUSSEN UNIVERSITY

NumPy Arrays Broadcasting



When NumPy expects arrays of the same shape but finds that this is not the case, it applies the so-called broadcasting rules.

RASMUSSEN UNIVERSITY

NumPy Arrays Broadcasting

Basically, there are 2 rules of Broadcasting to remember:

- For the arrays that do not have the same rank, then a 1 will be prepended to the smaller ranking arrays until their ranks match. For example, when adding arrays A and B of sizes (3,3) and (,3) [rank 2 and rank 1], 1 will be prepended to the dimension of array B to make it (1,3) [rank=2]. The two sets are compatible when their dimensions are equal or either one of the dimension is 1.
- When either of the dimensions compared is one, the other is used. In other words, dimensions with size 1 are stretched or "copied" to match the other. For example, upon adding a 2D array A of shape (3,3) to a 2D ndarray B of shape (1, 3). NumPy will apply the above rule of broadcasting. It shall stretch the array B and replicate the first row 3 times to make array B of dimensions (3,3) and perform the operation.

NumPy – Linear Equations

NumPy provides basic mathematical and statistical functions like mean, min, max, sum, prod, std, var, summation across different axes, transposing of a matrix, etc.

A particular NumPy feature of interest is solving a system of linear equations. NumPy has a function to solve linear equations. For example,

```
2x + 6y = 6
5x + 3y = -9
```

Can be solved in NumPy using

```
>>> coeffs  = np.array([[2, 6], [5, 3]])
>>> depvars = np.array([6, -9])
>>> solution = linalg.solve(coeffs, depvars)
>>> solution
array([-3.,  2.])
```

RASMUSSEN UNIVERSITY

NumPy – Example

- NumPy array of Values
- Random Selection of 100 numbers between 1 and 200
- Count the numbers between 25 and 100 not including these numbers

NumPy – Example

```
>>> import numpy as np
>>> a = np.random.randint(200,size=100)
>>> a
array([194, 131,  10, 100, 199, 123,  36,  14,  52, 195, 114, 181, 138, 144,  70, 185, 127,
52,  41, 126, 159,  39,  68, 118, 124, 119, 45, 161,  66,  29, 179, 194, 145, 163, 190,
150, 186,  25,  61, 187,   0,  69,  87,  20, 192,  18, 147,  53,  40, 113, 193, 178, 104, 170,
133,  69,  61,  48,  84, 121,  13,  49,  11,  29, 136, 141,  64,  22, 111, 162, 107,  33, 130,
11,  22, 167, 157,  99, 59,  12,  70, 154,  44,  45, 110, 180, 116,  56, 136,  54, 139, 26,
77, 128,  55, 143, 133, 137,   3,  83])
>>> np.compress((25 < a) & (a < 100),a).size
34
>>> a[(25 < a) & (a < 100)].size
34
```

RASMUSSEN UNIVERSITY

Pandas

- One of the most widely used Python Libraries in data science
- High performance
- Easy to use structures
- Data analysis tools
- Imported into Python using:

```
>>> import pandas as pd
```

RASMUSSEN UNIVERSITY

Pandas

Pandas also provide SQL-like functionality to filter, sort rows based on conditions. For example,

```
>>> people_dict = { "weight": pd.Series([68, 83, 112],index=["alice", "bob",
"charles"]),   "birthyear": pd.Series([1984, 1985, 1992], index=["bob", "alice", "charles"],
name="year"),
"children": pd.Series([0, 3], index=["charles", "bob"]),
"hobby": pd.Series(["Biking", "Dancing"], index=["alice", "bob"]),}
```

```
>>> people = pd.DataFrame(people_dict)
>>> people
```

	birthyear	children	hobby	weight
alice	1985	NaN	Biking	68
bob	1984	3.0	Dancing	83
charles	1992	0.0	NaN	112

RASMUSSEN UNIVERSITY

Pandas

```
1 >>> people[people["birthyear"] < 1990]
```

	birthyear	children	hobby	weight
alice	1985	NaN	Biking	68
bob	1984	3.0	Dancing	83

RASMUSSEN UNIVERSITY

Pandas

- New columns and rows can be easily added to the dataframe. In addition to the basic functionalities, pandas dataframe can be sorted by a particular column.
- Dataframes can also be easily exported and imported from CSV, Excel, JSON, HTML and SQL database. Some other essential methods that are present in dataframes are:
  - **head():** returns the top 5 rows in the dataframe object
  - **tail():** returns the bottom 5 rows in the dataframe
  - **info():** prints the summary of the dataframe
  - **describe():** gives a nice overview of the main aggregated values over each column

NUMPY & PANDAS HELPS

https://www.tutorialspoint.com/python_pandas/index.htm

https://www.datacamp.com/community/tutorials/pandas

https://www.tutorialspoint.com/numpy/index.htm

https://numpy.org/doc/stable/user/whatisnumpy.html

https://problemsolvingwithpython.com/

RASMUSSEN UNIVERSITY

DISCUSSION FORUM

- Requirements
  - Thread Post must be posted by Tuesday at 11:59 PM.
  - Reply Post must be posted by Saturday at 11:59 PM.

RASMUSSEN UNIVERSITY

DISCUSSION FORUM GRADING RUBRIC

LEVELS OF ACHIEVEMENT				
Criteria	Advanced	Proficient	Developing	Limited
Quality of Comments **Total Points:** **4**	90 to 100 % Timely and appropriate comments as defined by the instructor. Thoughtful and reflective, responds respectfully to other students' remarks, provokes questions and comments from the group.	75 to 89 % Volunteers comments, most are appropriate and reflect some thoughtfulness as defined by the instructor. Leads to other questions or remarks from students.	60 to 74 % Volunteers comments but lacks depth, may or may not lead to other questions from students. Off topic or irrelevant contributions.	0 to 59 % Did not participate.
Interaction with Course Resources **Total Points:** **3**	90 to 100 % Clear reference to text being discussed and connects it to other text of reference points from previous readings and discussions.	75 to 89 % Has done the reading with some thoroughness, may lack some detail or critical insight.	60 to 74 % Has done the reading; lacks thoroughness or understanding or insight.	0 to 59 % Did not participate.
Active Listening and Participation **Total Points:** **3**	90 to 100 % Comments clearly demonstrate respect and attentiveness to others; creatively builds on others' comments to offer insights. Directly answers the question(s) asked AND provides additional insights. Exceptional level of interaction as defined by the instructor.	75 to 89 % Shows consistency in responding to the comments of others. Stays focused on the stream of discussion rather than own ideas. Directly answers the questions(s) asked. Appropriate level of interaction as defined by the instructor.	60 to 74 % Does not stay focused on others' comments (too busy formulating own) or loses continuity of discussion. Inconsistent in tracking discussion stream. Indirectly answers the assigned question(s). Poor level of interaction as defined by the instructor.	0 to 59 % Did not participate

RASMUSSEN UNIVERSITY

COURSE PROJECT

Please download the Noshowappointments.csv dataset from the **Course Files** page.

Then write a Python script that can perform following tasks.

- Step 1: Read the CSV data into NumPy and Pandas.
- Step 2: Use Statistics/Linear Algebra concepts and determine PatientID, count (No-Shows=Yes).
- Step 3: Calculate Neighbourhood, count (No-Shows=Yes).
- Step 4: Create output dataset.

For your submission, include the following:

- Attach a Python script.
- An output screenshot.

RASMUSSEN UNIVERSITY

COURSE PROJECT

Submit these files as a single zipped ".zip" file to the drop box below. Please check the **Course Calendar** for specific due dates

Note: For help with zipping or compressing your files, visit the How do you zip files? Answers page.

The name of the file should be your first initial and last name, followed by an underscore and the name of the assignment, and an underscore and the date. (Mac users, please remember to append the ".zip" extension to the filename.) An example is shown below:

- Jstudent_exampleproblem_101504

RASMUSSEN UNIVERSITY

COURSE PROJECT GRADING RUBRIC

Criteria	Points
A correct Python script is attached.	50
Output screenshots are attached.	50
Total	100

RASMUSSEN UNIVERSITY

MODULE 02 ASSIGNMENT: United States Government

You are working as an analytics developer for the United States Government.

You need to refer to the trump_forbes_richest_americans.csv file in the **Course Files** folder. You need to calculate average worth from Column C. Please follow below steps.

- Step 1: Read trump_forbes_richest_americans.csv file using NumPy.
- Step 2: Use masked array concept to mask the array value where value=NA.
- Step 3: Calculate average using masked.Mean function.

For your submission, include the following:

Attach a Python script.

An output screenshot.

RASMUSSEN UNIVERSITY

MODULE 02 ASSIGNMENT: United States Government

Submit these files as a single zipped ".zip" file to the drop box below. Please check the **Course Calendar** for specific due dates.

Note: For help with zipping or compressing your files, visit the How do you zip files? Answers page.

The name of the file should be your first initial and last name, followed by an underscore and the name of the assignment, and an underscore and the date. (Mac users, please remember to append the ".zip" extension to the filename.) An example is shown below:

- Jstudent_exampleproblem_101504

RASMUSSEN UNIVERSITY

MODULE 02 ASSIGNMENT: Grading Rubric

Criteria	Points
A correct Python script is attached.	50
Output screenshots are attached.	50
Total	100

RASMUSSEN UNIVERSITY