

Handed out: 10/17/2017

Due by 11:59 PM (CST) on Tuesday, 10/24/2017

Download notebook file `python_fundamentals.ipynb` from the class site.. If you are new to Python, our recommendation is to go through every cell of the notebook and execute it. Save your notebook.

Problem 1. Install Anaconda for Python 2.7. We do not care whether it is the very latest release. If you already have a working Python 2.7 on your machine please be free to use it. Show to use what versions of Python, conda, pip and jupyter apps you have on your system. If you insist you can work with Python 3.

(10%)

PYTHON INSTALL

```
Kirks-MacBook-Pro:mckscripts el5vgxz$ python --version
Python 2.7.14
Kirks-MacBook-Pro:mckscripts el5vgxz$
```

ANACONDA INSTALL

```
[kirks-mbp:~ el5vgxz$ conda info
Current conda install:

    platform : osx-64
    conda version : 4.3.27
    conda is private : False
    conda-env version : 4.3.27
    conda-build version : 3.0.22
    python version : 2.7.13.final.0
    requests version : 2.18.4
    root environment : /Users/el5vgxz/anaconda2 (writable)
    default environment : /Users/el5vgxz/anaconda2
    envs directories : /Users/el5vgxz/anaconda2/envs
                    /Users/el5vgxz/.conda/envs
    package cache : /Users/el5vgxz/anaconda2/pkgs
                  /Users/el5vgxz/.conda/pkgs
    channel URLs : https://repo.continuum.io/pkgs/main/osx-64
                  https://repo.continuum.io/pkgs/main/noarch
                  https://repo.continuum.io/pkgs/free/osx-64
                  https://repo.continuum.io/pkgs/free/noarch
                  https://repo.continuum.io/pkgs/r/osx-64
                  https://repo.continuum.io/pkgs/r/noarch
                  https://repo.continuum.io/pkgs/pro/osx-64
                  https://repo.continuum.io/pkgs/pro/noarch
    config file : None
    netrc file : None
    offline mode : False
    user-agent : conda/4.3.27 requests/2.18.4 CPython/2.7.13 Darwin/16.7.0 OSX/10.12.6
    UID:GID : 501:20
kirks-mbp:~ el5vgxz$
```

PIP INSTALL

```
Kirks-MacBook-Pro:mckscripts el5vgxz$ easy_install pip
Searching for pip
Best match: pip 9.0.1
```

```
Adding pip 9.0.1 to easy-install.pth file
Installing pip script to /Library/Frameworks/Python.framework/Versions/2.7/bin
Installing pip3.5 script to /Library/Frameworks/Python.framework/Versions/2.7/bin
Installing pip3 script to /Library/Frameworks/Python.framework/Versions/2.7/bin

Using /Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-packages
Processing dependencies for pip
Finished processing dependencies for pip
Kirks-MacBook-Pro:mckscripts el5vgxz$
```

JUPYTER INSTALL

Ran this:

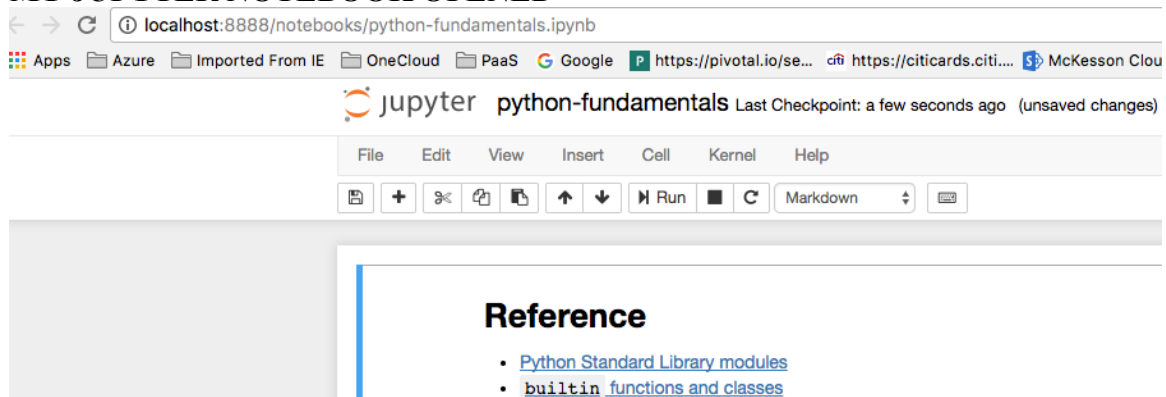
```
python -m pip install jupyter
```

After install I can execute: `jupyter notebook` as below and it launches it in a browser

```
kirks-mbp:~ el5vgxz$ jupyter notebook
[I 18:00:11.469 NotebookApp] Writing notebook server cookie secret to
/Users/el5vgxz/Library/Jupyter/runtime/notebook_cookie_secret
[I 18:00:11.514 NotebookApp] Serving notebooks from local directory: /Users/el5vgxz
[I 18:00:11.514 NotebookApp] 0 active kernels
[I 18:00:11.515 NotebookApp] The Jupyter Notebook is running at:
[I 18:00:11.515 NotebookApp]
http://localhost:8888/?token=a4efd8f6a97c3c02b413d786e8003d9f236940717d7e5780
[I 18:00:11.515 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 18:00:11.515 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=a4efd8f6a97c3c02b413d786e8003d9f236940717d7e578
0
[I 18:00:11.666 NotebookApp] Accepting one-time-token-authenticated connection from
::1
```

MY JUPYTER NOTEBOOK OPENED



Problem 2. Write a Python function that, when passed a single number as its argument, will return a tuple containing that number, square of that number and `sin()` of that number. Place this function in a loop going through a sequence of integers starting with 0 and ending with 9. Pass each integer to the function and print results to an output cell of your jupyter notebook.

(20%)

```
In [207]: def mytuple(num):  
          'Return my number, its square and its sine in a tuple'  
          sq=num**2  
          sn=math.sin(num)  
          result = num, sq, sn  
          return result  
          for x in range(0,10):  
              output=mytuple(x)  
              print output
```

```
(0, 0, 0.0)  
(1, 1, 0.8414709848078965)  
(2, 4, 0.9092974268256817)  
(3, 9, 0.1411200080598672)  
(4, 16, -0.7568024953079282)  
(5, 25, -0.9589242746631385)  
(6, 36, -0.27941549819892586)  
(7, 49, 0.6569865987187891)  
(8, 64, 0.9893582466233818)  
(9, 81, 0.4121184852417566)
```

Problem 3. Copy the code for problem #2 from your notebook into a standalone script. Name the script as you please but add suffix `.py` to it. You can modify that script in a text editor like Notepad. Do not do it in MS Word, Power Point or WordPad. If you have successfully install Anaconda, it came with a Python script editor called Spyder. You can choose to use Spyder. Demonstrate that you can run your script on the command prompt by typing:

```
C:..> python script.py
```

and obtain the same results as in the notebook. Copy back the content of your script into a cell of your notebook. Run it in that cell.

(20%)

To run the same code in a script, I had to first import math

```
[kirks-mbp:AzureTrain el5vgxz$ more week2.py
import math
def mytuple(num):
    'Return my number, its square and its sine in a tuple'
    sq=num**2
    sn=math.sin(num)
    result = num, sq, sn
    return result
for x in range(0,10):
    output=mytuple(x)
    print output
[kirks-mbp:AzureTrain el5vgxz$ python week2.py
(0, 0, 0.0)
(1, 1, 0.8414709848078965)
(2, 4, 0.9092974268256817)
(3, 9, 0.1411200080598672)
(4, 16, -0.7568024953079282)
(5, 25, -0.9589242746631385)
(6, 36, -0.27941549819892586)
(7, 49, 0.6569865987187891)
(8, 64, 0.9893582466233818)
(9, 81, 0.4121184852417566)
kirks-mbp:AzureTrain el5vgxz$ █
```

After copying the same code with the ‘import’ back to jupyter, it ran the same as the script or without the import. It did not matter.

```
In [208]: import math
def mytuple(num):
    'Return my number, its square and its sine in a tuple'
    sq=num**2
    sn=math.sin(num)
    result = num, sq, sn
    return result
for x in range(0,10):
    output=mytuple(x)
    print output

(0, 0, 0.0)
(1, 1, 0.8414709848078965)
(2, 4, 0.9092974268256817)
(3, 9, 0.1411200080598672)
(4, 16, -0.7568024953079282)
(5, 25, -0.9589242746631385)
(6, 36, -0.27941549819892586)
(7, 49, 0.6569865987187891)
(8, 64, 0.9893582466233818)
(9, 81, 0.4121184852417566)
```

Problem 4. Modify previous script so that it reads a sequence of integers from a simple text file and write its results into another text file. Prepare the input file with a single integer on every line. Write outputs into the output file similarly, one output per line. (25%)

The input file of numbers was “Strings”, so I had to convert the strings to Int to perform math. I read my “input.txt” file for list of numbers and then write to the output.txt file. I had to place the numbers back into str() and could not figure out how to put a newline in the same file.write function so I did it separately.

```

[kirks-mbp:AzureTrain el5vgxz$ more week2_4.py
import math
def mytuple(num):
    'Return my number, its square and its sine in a tuple'
    sq=num**2
    sn=math.sin(num)
    result = num, sq, sn
    return result
infile = open('input.txt')
outfile = open('output.txt','w')
for line in infile:
    conv_num = int(line)
    output=mytuple(conv_num)
    outfile.write(str(output))
    outfile.write("\n")
infile.close()
outfile.close()

```

```

[kirks-mbp:AzureTrain el5vgxz$ more input.txt
5
9
22
4
1
32
13
3
[kirks-mbp:AzureTrain el5vgxz$ python week2_4.py
[kirks-mbp:AzureTrain el5vgxz$
[kirks-mbp:AzureTrain el5vgxz$ more output.txt
(5, 25, -0.9589242746631385)
(9, 81, 0.4121184852417566)
(22, 484, -0.008851309290403876)
(4, 16, -0.7568024953079282)
(1, 1, 0.8414709848078965)
(32, 1024, 0.5514266812416906)
(13, 169, 0.4201670368266409)
(3, 9, 0.1411200080598672)
[kirks-mbp:AzureTrain el5vgxz$ █

```

Problem 5. Consider attached file `small_car_data.xlsx`. Import data into a panda DataFrame. Use panda machinery to calculate basic statistics for all numerical columns, min, max, median, average and standard deviation. Determine and present graphically statistical distributions of values in those columns. Try to establish correlations between values in different columns. Which two columns are mutually most correlated? (25%)

The best I could come up with as I ran out of time. And had difficulties with the requirements.

```

import math
import sys
import numpy as np

```

```

import pandas as pd
pd.options.display.max_rows = 20
pd.options.display.max_columns = 15
infile = 'small_car_data.csv'
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_colwidth', -1)
data = pd.read_csv(infile)
print data.values
print data.columns
#data.groupby(['Cylinders']).min()
old=data.groupby(['Acceleration']).agg([pd.np.min], [pd.np.max])

print data
print old

```

SUBMISSION INSTRUCTIONS:

Your main submission should be an MS Word or PDF document containing your code, results produced by that code and brief textual descriptions of what you did and why. Typically, you copy important snippets of your code as ASCII text and the results into this Word document. Please copy ASCII text rather than take a snapshot. If your MS Word document is larger than 1 MB save it as a MINIMIZED PDF. Describe the purpose of every code snippet and the significance of the results. Start with the text of this homework assignment as the template. Please add any other files that you might have used or generated. Please do not provide ZIP or RAR or any other archives. Canvas cannot open those archives and they turn into a nuisance for us.