

JupyterTutorial_CH

January 15, 2021

0.0.1 Intro to Jupyter Notebooks and Quick Python Warmup

0.0.2 University of California, Santa Barbara

0.0.3 PSTAT 135/235: Big Data Analytics

0.0.4 Last Updated: May 30, 2020

Welcome to this short assignment where you will demonstrate basic Jupyter notebook knowledge and do a quick Python warmup! Total points: 8

- 1) (1 PT) First, rename this notebook to JupyterTutorial_[your_initials], where you will enter your initials in place of [your_initials].
- 2) (1 PT) In the cell below, enter a list of data science topics you find interesting. Use the markdown style (you will need to change the style from the Code style).

Regression Analysis, Machine Learning, Big Data Analytics

- 3) (1 PT) In the cell below, enter the following Python list:

```
some_vals = [1, 6, 10, 55]
```

You will use the Code style, and run the cell.

```
[1]: some_vals = [1, 6, 10, 55]
     some_vals
```

```
[1]: [1, 6, 10, 55]
```

- 4) (1 PT) Next, use a list comprehension to return a filtered list containing only the values greater than 6.
Call this list *some_vals_filtered* and print it. You can chain multiple commands on a single line like this:

```
[2]: x=1; z= x+1; z
```

```
[2]: 2
```

```
[3]: some_vals_filtered = [x for x in some_vals if x > 6]; print(some_vals_filtered)
```

```
[10, 55]
```

```
[4]: some_vals_filtered = []
     for x in some_vals:
         if x > 6:
             some_vals_filtered.append(x)
     print(some_vals_filtered)
```

[10, 55]

Next, a small pandas dataframe is constructed.

```
[5]: import pandas as pd

     df = pd.DataFrame({'first_name': ['Andy', 'Crystal'],
                        'domain_facebook' : [1,1],
                        'domain_foursquare' : [0,0],
                        'age' : [20, 32]})

     df
```

```
[5]:  first_name  domain_facebook  domain_foursquare  age
     0      Andy                1                  0   20
     1   Crystal                1                  0   32
```

5) (1 PT) In the cell below, write a list comprehension that returns the fields names in the dataframe *df* containing the string *domain*. Run the cell to verify the correct result.

```
[6]: [x for x in df.columns if 'domain' in x]
```

```
[6]: ['domain_facebook', 'domain_foursquare']
```

6) (1 PT) Use the list comprehension from (5) to index into *df* and show the data for columns containing *domain*

```
[7]: df[[x for x in df.columns if 'domain' in x]]
```

```
[7]:  domain_facebook  domain_foursquare
     0              1                  0
     1              1                  0
```

7) (1 PT) In the cell below, print the *domain_facebook* column

```
[8]: print('domain_facebook')
```

domain_facebook

8) (1 PT) In the cell below, print the row with index 1.

```
[9]: print(df.iloc[1,0])
     #Loc
```

Crystal

```
[11]: # Save notebook as PDF document
!jupyter nbconvert --to pdf `pwd`/*.ipynb
```

```
[NbConvertApp] Converting notebook
/home/jovyan/assignments/M1_6/JupyterTutorial_CH.ipynb to pdf
[NbConvertApp] Writing 29169 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 40918 bytes to
/home/jovyan/assignments/M1_6/JupyterTutorial_CH.pdf
```

```
[ ]:
```