

February 3, 2020

1 import numpy as np

import pandas as pd import scipy as sp

```
[253]: import numpy as np
import pandas as pd
import scipy as sp
```

```
[4]: %matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
[7]: %%file hw_data.csv
id,sex,height
1,M,190,77
2,F,120,70
3,F,110,68
4,M,150,72
5,O,120,66
6,M,120,60
7,F,140,70
```

Overwriting hw_data.csv

2 Python

2.1 1. Finish creating the following function that takes a list and returns the average value.

```
[39]: def average(my_list):

    avg = (sum(my_list)/len(my_list))
    print(avg)

average([1,2,1,4,3,2,5,9])
```

3.375

2.2 2. Using a Dictionary keep track of the count of numbers (or items) from a list

```
[71]: def counts(my_list):  
    freq = {}  
    for item in my_list:  
        if (item in freq):  
            freq[item] += 1  
        else:  
            freq[item] = 1  
  
    return freq  
  
counts([1,2,1,4,3,2,5,9])
```

```
[71]: {1: 2, 2: 2, 4: 1, 3: 1, 5: 1, 9: 1}
```

2.3 3. Using the counts() function and the .split() function, return a dictionary of most occurring words from the following paragraph. Bonus, remove punctuation from words.

```
[181]: paragraph_text = '''  
For a minute or two she stood looking at the house, and wondering what to do_  
→next, when suddenly a footman in livery came running out of the wood-(she_  
→considered him to be a footman because he was in livery: otherwise, judging_  
→by his face only, she would have called him a fish)-and rapped loudly at the_  
→door with his knuckles. It was opened by another footman in livery, with a_  
→round face, and large eyes like a frog; and both footmen, Alice noticed, had_  
→powdered hair that curled all over their heads. She felt very curious to_  
→know what it was all about, and crept a little way out of the wood to listen.  
The Fish-Footman began by producing from under his arm a great letter, nearly_  
→as large as himself, and this he handed over to the other, saying, in a_  
→solemn tone, 'For the Duchess. An invitation from the Queen to play croquet.  
→' The Frog-Footman repeated, in the same solemn tone, only changing the_  
→order of the words a little, 'From the Queen. An invitation for the Duchess_  
→to play croquet.'  
Then they both bowed low, and their curls got entangled together.  
Alice laughed so much at this, that she had to run back into the wood for fear_  
→of their hearing her; and when she next peeped out the Fish-Footman was_  
→gone, and the other was sitting on the ground near the door, staring_  
→stupidly up into the sky.  
Alice went timidly up to the door, and knocked.
```

‘There’s no sort of use in knocking,’ said the Footman, ‘and that for two reasons. First, because I’m on the same side of the door as you are; secondly, because they’re making such a noise inside, no one could possibly hear you.’ And certainly there was a most extraordinary noise going on within—a constant howling and sneezing, and every now and then a great crash, as if a dish or kettle had been broken to pieces.

‘Please, then,’ said Alice, ‘how am I to get in?’

‘There might be some sense in your knocking,’ the Footman went on without attending to her, ‘if we had the door between us. For instance, if you were inside, you might knock, and I could let you out, you know.’ He was looking up into the sky all the time he was speaking, and this Alice thought decidedly uncivil. ‘But perhaps he can’t help it,’ she said to herself; ‘his eyes are so very nearly at the top of his head. But at any rate he might answer questions.—How am I to get in?’ she repeated, aloud.

‘I shall sit here,’ the Footman remarked, ‘till tomorrow—’

At this moment the door of the house opened, and a large plate came skimming out, straight at the Footman’s head: it just grazed his nose, and broke to pieces against one of the trees behind him.’’

```
import string
words = paragraph_text.split()
words2 = [''.join(c for c in s if c not in punctuation) for s in words]

counts(words2)

#This solution to the bonus problem leaves Frog-footman and Fish-Footman as whole words, but does screw up contractions.
```

```
[181]: {'For': 3,
      'a': 15,
      'minute': 1,
      'or': 2,
      'two': 2,
      'she': 6,
      'stood': 1,
      'looking': 2,
      'at': 6,
      'the': 32,
      'house,': 1,
      'and': 17,
      'wondering': 1,
      'what': 2,
      'to': 15,
      'do': 1,
      'next,': 1,
      'when': 2,
```

'suddenly': 1,
'footman': 3,
'in': 7,
'livery': 1,
'came': 2,
'running': 1,
'out': 3,
'of': 9,
'woodshe': 1,
'considered': 1,
'him': 2,
'be': 2,
'because': 3,
'he': 5,
'was': 8,
'livery.': 1,
'otherwise.': 1,
'judging': 1,
'by': 3,
'his': 6,
'face': 1,
'only.': 1,
'would': 1,
'have': 1,
'called': 1,
'fishand': 1,
'rapped': 1,
'loudly': 1,
'door': 4,
'with': 2,
'knuckles.': 1,
'It': 1,
'opened': 1,
'another': 1,
'livery.': 1,
'round': 1,
'face.': 1,
'large': 3,
'eyes': 2,
'like': 1,
'frog.': 1,
'both': 2,
'footmen.': 1,
'Alice': 4,
'noticed.': 1,
'had': 4,
'powdered': 1,

'hair': 1,
'that': 3,
'curled': 1,
'all': 3,
'over': 2,
'their': 3,
'heads.': 1,
'She': 1,
'felt': 1,
'very': 2,
'curious': 1,
'know': 1,
'it': 2,
'about,': 1,
'crept': 1,
'little': 1,
'way': 1,
'wood': 2,
'listen.': 1,
'The': 2,
'Fish-Footman': 2,
'began': 1,
'producing': 1,
'from': 2,
'under': 1,
'arm': 1,
'great': 2,
'letter,': 1,
'nearly': 2,
'as': 4,
'himself,': 1,
'this': 3,
'handed': 1,
'other,': 1,
'saying,': 1,
'solemn': 2,
'tone,': 2,
'Duchess.': 1,
'An': 2,
'invitation': 2,
'Queen': 1,
'play': 2,
'croquet.': 2,
'Frog-Footman': 1,
'repeated,': 2,
'same': 2,
'only': 1,

'changing': 1,
'order': 1,
'words': 1,
'little,': 1,
'From': 1,
'Queen.': 1,
'for': 3,
'Duchess': 1,
'Then': 1,
'they': 1,
'bowed': 1,
'low,': 1,
'curls': 1,
'got': 1,
'entangled': 1,
'together.': 1,
'laughed': 1,
'so': 2,
'much': 1,
'this,': 1,
'run': 1,
'back': 1,
'into': 3,
'fear': 1,
'hearing': 1,
'her;': 1,
'next': 1,
'peeped': 1,
'gone,': 1,
'other': 1,
'sitting': 1,
'on': 4,
'ground': 1,
'near': 1,
'door,': 2,
'staring': 1,
'stupidly': 1,
'up': 3,
'sky.': 1,
'went': 2,
'timidly': 1,
'knocked.': 1,
'Theres': 1,
'no': 2,
'sort': 1,
'use': 1,
'knocking,': 2,

'said': 3,
'Footman,': 1,
'reasons.': 1,
'First,': 1,
'Im': 1,
'side': 1,
'you': 5,
'are;': 1,
'secondly,': 1,
'theyre': 1,
'making': 1,
'such': 1,
'noise': 2,
'inside,': 2,
'one': 2,
'could': 2,
'possibly': 1,
'hear': 1,
'you.': 1,
'And': 1,
'certainly': 1,
'there': 1,
'most': 1,
'extraordinary': 1,
'going': 1,
'withina': 1,
'constant': 1,
'howling': 1,
'sneezing,': 1,
'every': 1,
'now': 1,
'then': 1,
'crash,': 1,
'if': 3,
'dish': 1,
'kettle': 1,
'been': 1,
'broken': 1,
'pieces.': 1,
'Please,': 1,
'then,': 1,
'Alice,': 1,
'how': 1,
'am': 2,
'I': 4,
'get': 2,
'in?': 2,

'There': 1,
'might': 3,
'some': 1,
'sense': 1,
'your': 1,
'Footman': 2,
'without': 1,
'attending': 1,
'her,': 1,
'we': 1,
'between': 1,
'us.': 1,
'instance,': 1,
'were': 1,
'knock,': 1,
'let': 1,
'out,': 2,
'know.': 1,
'He': 1,
'sky': 1,
'time': 1,
'speaking,': 1,
'thought': 1,
'decidedly': 1,
'uncivil.': 1,
'But': 2,
'perhaps': 1,
'cant': 1,
'help': 1,
'it,': 1,
'herself;': 1,
'are': 1,
'top': 1,
'head.': 1,
'any': 1,
'rate': 1,
'answer': 1,
'questions.How': 1,
'aloud.': 1,
'shall': 1,
'sit': 1,
'here,': 1,
'remarked,': 1,
'till': 1,
'tomorrow': 1,
'At': 1,
'moment': 1,


```
'house': 1,
'opened.': 1,
'plate': 1,
'skimming': 1,
'straight': 1,
'Footmans': 1,
'head.': 1,
'just': 1,
'grazed': 1,
'nose.': 1,
'broke': 1,
'pieces': 1,
'against': 1,
'trees': 1,
'behind': 1,
'him.': 1}
```

2.4 4. Read in a file and write each line from the file to a new file Title-ized

This is the first line -> This Is The First Line

Hint: There's a function to do this

```
[180]: f = open("sample.txt")
f2 = open("output.txt", 'w')
for line in f:
    f2.write(line.title())
```

3 Numpy

3.1 1. Given a list, find the average using a numpy function.

```
[182]: simple_list = [1,2,1,4,3,2,5,9]

np.mean(simple_list)
```

```
[182]: 3.375
```

3.2 2. Given two lists of Heights and Weights of individual, calculate the BMI of those individuals, without writing a for-loop

```
[203]: heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]
h = np.array(heights)
w = np.array(weights)
```

```
m = (h/100)**2  
  
w/m
```

```
[203]: array([29.06592681, 27.73229978, 30.73941662, 24.16326531, 26.33288875])
```

3.3 3. Create an array of length 20 filled with random values (between 0 to 1)

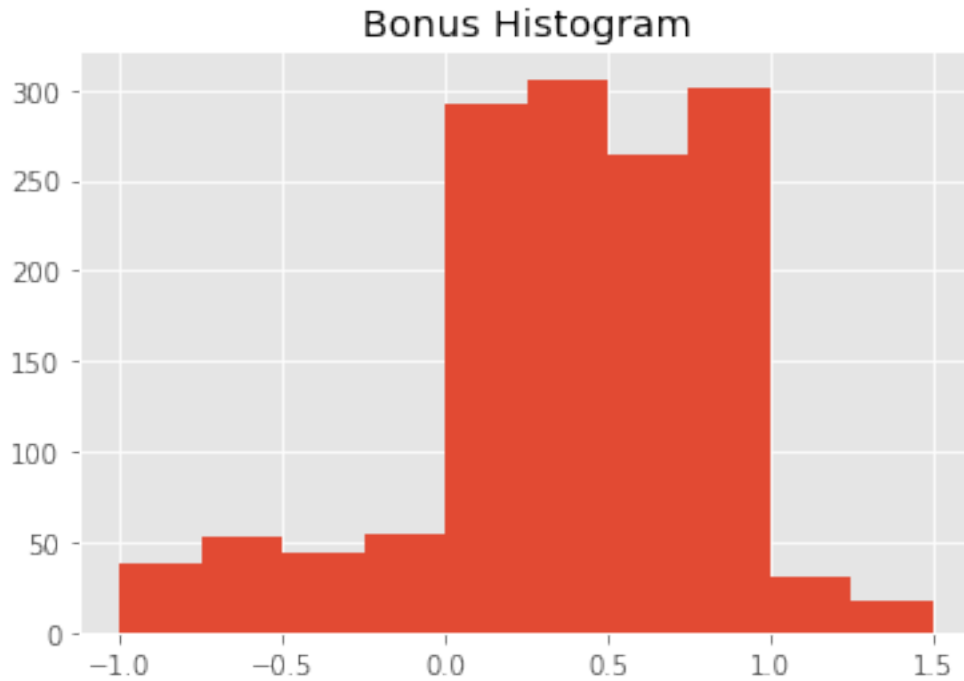
```
[206]: np.random.rand(20,)
```

```
[206]: array([0.9596771 , 0.6018137 , 0.5161972 , 0.61868466, 0.10613183,  
            0.99503322, 0.05764058, 0.51333414, 0.30606795, 0.63154324,  
            0.39098045, 0.29253388, 0.83523646, 0.82548883, 0.45626526,  
            0.33307204, 0.81724582, 0.42590418, 0.59521944, 0.11451932])
```

3.4 Bonus. 1. Create an array with a large (>1000) length filled with random numbers from different distributions (normal, uniform, etc.). 2. Then, plot a histogram of these values.

```
[251]: a = np.random.rand(500,)  
       b = np.random.randn(500,)  
       c = np.random.randint(500,)  
       d = np.random.random_sample(500,)  
  
       f = np.concatenate((a,b,c,d), axis=None)  
       g = np.array(f)  
  
       plt.hist(g, bins = [-1 , -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5])  
       plt.title("Bonus Histogram")
```

```
[251]: Text(0.5, 1.0, 'Bonus Histogram')
```



4 Pandas

4.1 1. Read in a CSV () and display all the columns and their respective data types

```
[264]: df = pd.read_csv('hw_data.csv', index_col='id')
df.dtypes
```

```
[264]: sex          object
weight       int64
height       int64
dtype: object
```

4.2 2. Find the average weight

```
[271]: df["weight"].mean()
```

```
[271]: 135.71428571428572
```

4.3 3. Find the Value Counts on column sex

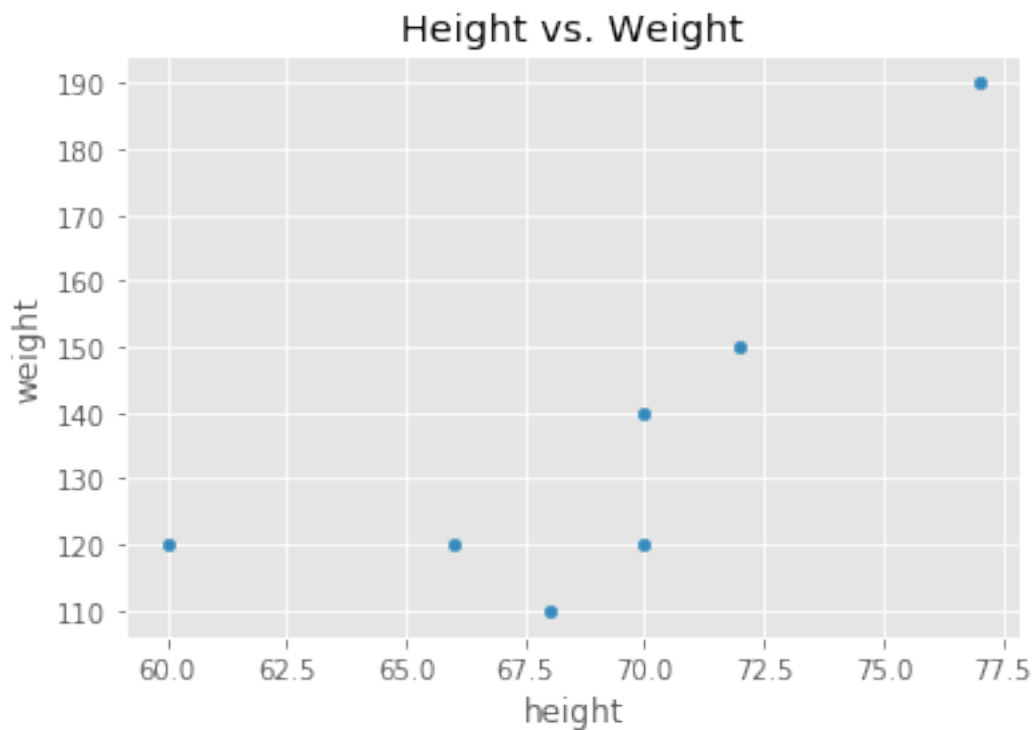
```
[272]: df["sex"].value_counts()
```

```
[272]: M    3
      F    3
      0    1
      Name: sex, dtype: int64
```

4.4 4. Plot Height vs. Weight

```
[280]: df.plot(x='height', y='weight', kind='scatter', title='Height vs. Weight')
```

```
[280]: <matplotlib.axes._subplots.AxesSubplot at 0x1f9ee648708>
```



4.5 5. Calculate BMI and save as a new column

```
[286]: H = (df["height"]*df["height"])
      BMI = (df["weight"]/H)*703
      BMI
      df['BMI'] = BMI
      df
```

```
[286]:   sex  weight  height      BMI
      id
      1    M    190      77  22.528251
      2    F    120      70  17.216327
```

3	F	110	68	16.723616
4	M	150	72	20.341435
5	O	120	66	19.366391
6	M	120	60	23.433333
7	F	140	70	20.085714

4.6 6. Save sheet as a new CSV file hw_dataB.csv

```
[287]: df.to_csv(r'hw_dataB.csv')
```

4.7 Run the following

```
[288]: !cat hw_dataB.csv
```

'cat' is not recognized as an internal or external command,
operable program or batch file.

```
[ ]:
```