

# assignment\_2

February 2, 2020

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

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

```
[11]: %%file hw_data.csv
id,sex,weight,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

## 1 Python

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

```
[14]: def average(my_list):
    total = 0
    for item in my_list:
        total += item

    mean = total / len(my_list)

    return mean

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

```
[14]: 3.375
```

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

```
[15]: def counts(my_list):  
  
    counts = dict()  
  
    for item in my_list:  
        if item not in counts.keys():  
            counts[item] = 1  
        else:  
            counts[item] += 1  
  
    return counts  
  
counts([1,2,1,4,3,2,5,9])
```

```
[15]: {1: 2, 2: 2, 4: 1, 3: 1, 5: 1, 9: 1}
```

## 1.3 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.

```
[113]: import string  
  
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.'''

```
words = paragraph_text.split()
punct = [''', ''', '-']

# Clean up punctuation marks from text.
for i in paragraph_text:
    if i in string.punctuation:
        punct.append(i)

clean_string = ''.join(i for i in paragraph_text if not i in punct)

clean_text = clean_string.split()
answer = counts(clean_text)

print(type(answer))
print(answer)
```

```
<class 'dict'>
{'For': 3, 'a': 15, 'minute': 1, 'or': 2, 'two': 2, 'she': 6, 'stood': 1,
'looking': 2, 'at': 6, 'the': 32, 'house': 2, 'and': 17, 'wondering': 1, 'what':
2, 'to': 15, 'do': 1, 'next': 2, 'when': 2, 'suddenly': 1, 'footman': 3, 'in':
9, 'livery': 3, 'came': 2, 'running': 1, 'out': 5, 'of': 9, 'woodshe': 1,
'considered': 1, 'him': 3, 'be': 2, 'because': 3, 'he': 5, 'was': 8,
'otherwise': 1, 'judging': 1, 'by': 3, 'his': 6, 'face': 2, 'only': 2, 'would':
1, 'have': 1, 'called': 1, 'fishand': 1, 'rapped': 1, 'loudly': 1, 'door': 6,
```

'with': 2, 'knuckles': 1, 'It': 1, 'opened': 2, 'another': 1, 'round': 1, 'large': 3, 'eyes': 2, 'like': 1, 'frog': 1, 'both': 2, 'footmen': 1, 'Alice': 5, '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': 2, 'it': 3, 'about': 1, 'crept': 1, 'little': 2, 'way': 1, 'wood': 2, 'listen': 1, 'The': 2, 'FishFootman': 2, 'began': 1, 'producing': 1, 'from': 2, 'under': 1, 'arm': 1, 'great': 2, 'letter': 1, 'nearly': 2, 'as': 4, 'himself': 1, 'this': 4, 'handed': 1, 'other': 2, 'saying': 1, 'solemn': 2, 'tone': 2, 'Duchess': 2, 'An': 2, 'invitation': 2, 'Queen': 2, 'play': 2, 'croquet': 2, 'FrogFootman': 1, 'repeated': 2, 'same': 2, 'changing': 1, 'order': 1, 'words': 1, 'From': 1, 'for': 3, 'Then': 1, 'they': 1, 'bowed': 1, 'low': 1, 'curls': 1, 'got': 1, 'entangled': 1, 'together': 1, 'laughed': 1, 'so': 2, 'much': 1, 'run': 1, 'back': 1, 'into': 3, 'fear': 1, 'hearing': 1, 'her': 2, 'peeped': 1, 'gone': 1, 'sitting': 1, 'on': 4, 'ground': 1, 'near': 1, 'staring': 1, 'stupidly': 1, 'up': 3, 'sky': 2, 'went': 2, 'timidly': 1, 'knocked': 1, 'Theres': 1, 'no': 2, 'sort': 1, 'use': 1, 'knocking': 2, 'said': 3, 'Footman': 3, 'reasons': 1, 'First': 1, 'Im': 1, 'side': 1, 'you': 6, 'are': 2, 'secondly': 1, 'theyre': 1, 'making': 1, 'such': 1, 'noise': 2, 'inside': 2, 'one': 2, 'could': 2, 'possibly': 1, 'hear': 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': 2, 'crash': 1, 'if': 3, 'dish': 1, 'kettle': 1, 'been': 1, 'broken': 1, 'pieces': 2, 'Please': 1, 'how': 1, 'am': 2, 'I': 4, 'get': 2, 'There': 1, 'might': 3, 'some': 1, 'sense': 1, 'your': 1, 'without': 1, 'attending': 1, 'we': 1, 'between': 1, 'us': 1, 'instance': 1, 'were': 1, 'knock': 1, 'let': 1, 'He': 1, 'time': 1, 'speaking': 1, 'thought': 1, 'decidedly': 1, 'uncivil': 1, 'But': 2, 'perhaps': 1, 'cant': 1, 'help': 1, 'herself': 1, 'top': 1, 'head': 2, 'any': 1, 'rate': 1, 'answer': 1, 'questionsHow': 1, 'aloud': 1, 'shall': 1, 'sit': 1, 'here': 1, 'remarked': 1, 'till': 1, 'tomorrow': 1, 'At': 1, 'moment': 1, 'plate': 1, 'skimming': 1, 'straight': 1, 'Footmans': 1, 'just': 1, 'grazed': 1, 'nose': 1, 'broke': 1, 'against': 1, 'trees': 1, 'behind': 1}

#### 1.4 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

```
[2]: # method 1
string = 'This is the first line'
words = string.split()
new_words = [i.capitalize() for i in words]
' '.join(new_words)

# method 2
str2 = string.title()
print(str2)
```

This Is The First Line

## 2 Numpy

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

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

ans = np.mean(simple_list)
ans
```

```
[115]: 3.375
```

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

```
[178]: heights = [174, 173, 173, 175, 171]
weights = [88, 83, 92, 74, 77]

# assuming we are not working on the metric system here...
# BMI = (703 * weight(lbs))/(height(in)**2)
def bmi_calc(weights, heights, formula = 'metric'):
    if formula is 'metric':
        bmi = np.array(weights)/(np.array(heights)/100)**2
    elif formula == 'imperial':
        bmi = np.array(weights)* 703 / (np.array(heights)**2)
    return bmi
bmi_ans = bmi_calc(weights, heights, 'metric')
bmi_ans
```

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

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

```
[147]: ans_3 = np.random.uniform(0,1,20)
ans_3
```

```
[147]: array([0.66038482, 0.02865875, 0.39046869, 0.22645616, 0.91314194,
          0.98461939, 0.33228132, 0.47075539, 0.9009102 , 0.21744614,
          0.24411088, 0.07030759, 0.90170141, 0.11009564, 0.84920382,
          0.22137157, 0.36174934, 0.94491561, 0.68595384, 0.0990832 ])
```

2.4 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.

### 3 Pandas

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

```
[154]: raw_df = pd.read_csv("hw_data.csv")
       print(raw_df.dtypes)
```

```
id          int64
sex         object
weight      int64
height      int64
dtype: object
```

3.2 2. Find the average weight

```
[159]: avg_weight = raw_df.weight.mean(axis=0)
       avg_weight
```

```
[159]: 135.71428571428572
```

3.3 3. Find the Value Counts on column sex

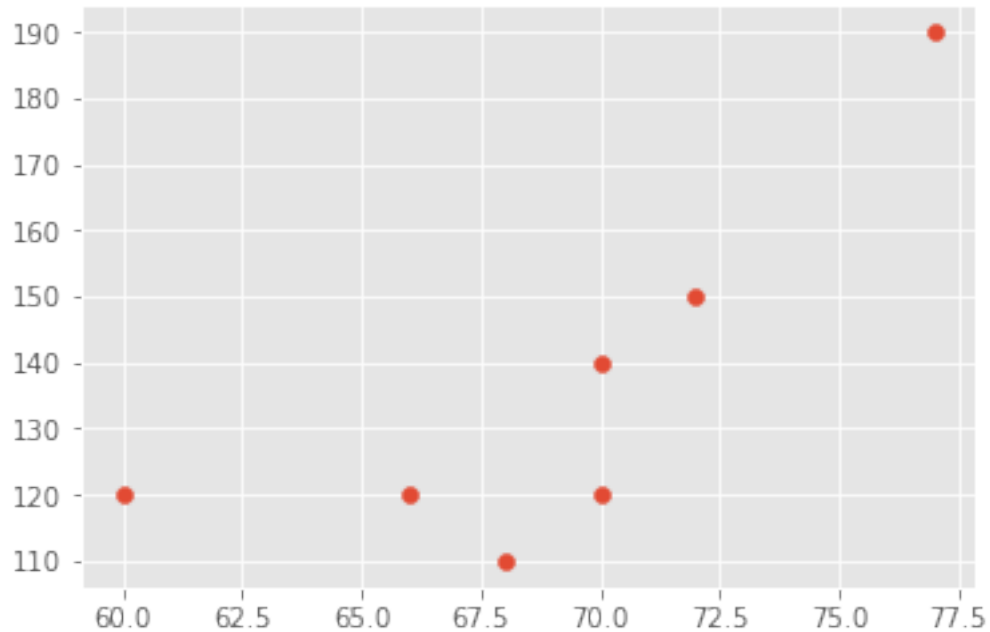
```
[162]: value_counts = counts(raw_df.sex)
       value_counts
```

```
[162]: {'M': 3, 'F': 3, 'O': 1}
```

3.4 4. Plot Height vs. Weight

```
[163]: h = raw_df.height
       w = raw_df.weight
       plt.scatter(h,w)
```

```
[163]: <matplotlib.collections.PathCollection at 0x123831550>
```



### 3.5 5. Calculate BMI and save as a new column

```
[180]: raw_df['bmi'] = bmi_calc(raw_df.weight, raw_df.height, 'imperial')
raw_df
```

```
[180]:
```

|   | id | sex | weight | height | bmi       |
|---|----|-----|--------|--------|-----------|
| 0 | 1  | M   | 190    | 77     | 22.528251 |
| 1 | 2  | F   | 120    | 70     | 17.216327 |
| 2 | 3  | F   | 110    | 68     | 16.723616 |
| 3 | 4  | M   | 150    | 72     | 20.341435 |
| 4 | 5  | O   | 120    | 66     | 19.366391 |
| 5 | 6  | M   | 120    | 60     | 23.433333 |
| 6 | 7  | F   | 140    | 70     | 20.085714 |

### 3.6 6. Save sheet as a new CSV file hw\_dataB.csv

```
[183]: raw_df.to_csv('hw_dataB.csv', index = None)
```

### 3.7 Run the following

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

```
id,sex,weight,height,bmi
1,M,190,77,22.528250969809413
2,F,120,70,17.216326530612246
```

3,F,110,68,16.723615916955016  
4,M,150,72,20.341435185185187  
5,O,120,66,19.366391184573004  
6,M,120,60,23.433333333333334  
7,F,140,70,20.085714285714285

[ ]: