KEY_Lesson14_Basic_Stats_I

July 12, 2019

1 Introduction to Statistics Part I

This morning, we learned how to get started with Numpy and how to load data using Pandas. Now, we will learn how to use and intepret some simple statistical methods.

But, what is Statistics?

Statistics is the study of organizing and interpreting *data*. Usually, this includes calculating numbers that describe the data in a simpler fashion. In this lesson, we will be calculating two of these numbers: the *mean* and the *median*. These are metrics that you've likely learned about in math class before, and if you remember, we've already been using the mean in some of our lessons earlier this week!

First, let's load `numpy` and `pandas` as we learned yesterday and this morning:

[25]:
```
# import numpy and pandas
import numpy as np
import pandas as pd
# mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
path = '/content/gdrive/My Drive/SummerExperience-master/'
```

Now, let's load the iris data using `pandas` into a variable called `data_table`. We will be focusing on the `sepal_length` column here, so let's isolate that using our `pandas` skills and print that out as well!

[26]:
```
# import iris.csv and save to data_table variable
data_table = pd.read_csv(path + 'Lessons/SampleData/iris.csv')
print(data_table['sepal_length'])
```

```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
5      5.4
6      4.6
7      5.0
8      4.4
9      4.9
10     5.4
```

```
11     4.8
12     4.8
13     4.3
14     5.8
15     5.7
16     5.4
17     5.1
18     5.7
19     5.1
20     5.4
21     5.1
22     4.6
23     5.1
24     4.8
25     5.0
26     5.0
27     5.2
28     5.2
29     4.7
      ...
120    6.9
121    5.6
122    7.7
123    6.3
124    6.7
125    7.2
126    6.2
127    6.1
128    6.4
129    7.2
130    7.4
131    7.9
132    6.4
133    6.3
134    6.1
135    7.7
136    6.3
137    6.4
138    6.0
139    6.9
140    6.7
141    6.9
142    5.8
143    6.8
144    6.7
145    6.7
146    6.3
147    6.5
```

```
148    6.2
149    5.9
Name: sepal_length, Length: 150, dtype: float64
```

As mentioned in our earlier lesson, numpy has its own data type - numpy array. Though numpy functions can be used on numeric columns of pandas DataFrames, let's just see the difference between the pandas DataFrame colmun and the same data in a numpy array.

[16]:
```
# create sepal_length_np numpy array
sepal_length_np = np.array(data_table['sepal_length'])
print(sepal_length_np)
```

```
[5.1 4.9 4.7 4.6 5.  5.4 4.6 5.  4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.  5.  5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.
 5.5 4.9 4.4 5.1 5.  4.5 4.4 5.  5.1 4.8 5.1 4.6 5.3 5.  7.  6.4 6.9 5.5
 6.5 5.7 6.3 4.9 6.6 5.2 5.  5.9 6.  6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
 6.3 6.1 6.4 6.6 6.8 6.7 6.  5.7 5.5 5.5 5.8 6.  5.4 6.  6.7 6.3 5.6 5.5
 5.5 6.1 5.8 5.  5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.  6.9 5.6 7.7 6.3 6.7 7.2
 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.  6.9 6.7 6.9 5.8 6.8
 6.7 6.7 6.3 6.5 6.2 5.9]
```

1.1 Calculating the mean

Calculating the mean, or average, of a set of numbers is a useful metric which tells you what the "center" of the data is. If you remember from our previous lesson, the mean is calculated as follows:

mean = sum_of_all_values/total_number_of_values

First, let's calculate the mean as we previously did, using the sum and len built-in functions.

[20]:
```
# calculate mean and save to data_mean variable
data_mean = sum(data_table['sepal_length'])/ len(data_table['sepal_length'])
print(data_mean)
```

```
5.843333333333335
```

As we saw this morning, numpy includes many useful functions and this includes calculating the mean! Try using np.mean below to calculate the same value:

[24]:
```
# Use the numpy mean function to calculate the same mean value
# try using both the pandas DataFrame column and the numpy array versions of␣
 ↪the data
print(np.mean(data_table['sepal_length']))
print(np.mean(sepal_length_np))
```

```
5.843333333333335
5.843333333333334
```

1.2 Calculating the median

The *median* is a statistic that tells us the middle value of the sorted data. Another way to think of it is that 50% of our data points are *above* the median, and 50% are *below* the median. Another word for this is the *50th percentile* of the data.

Similarlly to above, we can use the `numpy` function `np.median` to calculate this statistic.

[23]:
```
# Use the numpy median function to find the median of your data
print(np.median(data_table['sepal_length']))
```

```
5.8
```

As we can see here, both the mean (5.84 cm) and the median (5.8 cm) are very close together, this tells us that this dataset is *symetrically distributed*. In other words, there are roughly the same amount of high-value variables as low-value variables.

Both the *mean* and *median* provide useful information about what the "average" datapoint in your set looks like. These statistics are often used summarize data from several subjects or observations in an experiment, which gives us a better idea of the **general trends** in our data. Next, we will learn about additional statistical tools.

In this lesson you learned how to: * Calculate the mean of set of values by "hand." * Use functions in `numpy` to calculate both the mean and median.

Now, lets continue to practice with your partner!