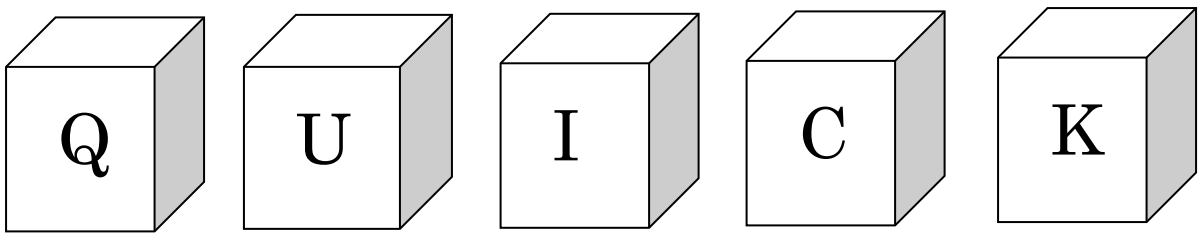
Quick Check

- Grammer＆Usage



>> Python で学び直す数学【統計編】

～ 統計指標の求め方／度数分布表を描画してみよう

仕事や趣味を通して Python のコードを書いている方であれば、データを読み込んで、統計指標を使った機会がある方も多いと思いますが、実際に少し興味はあるけれど、どう Python で書いてみればよいのかわからないという方も多いと思います。Python のライブラリの基本的な書き方を含め、読み込んだ CSV データの内容を、統計的に処理・表現するという形の演習をしていきたいと思います。

「数学的な問題を Python で簡単なスクリプトを作って動作を確認する」こと通して、Python に触れる機会をつくっていきたいと考えています。 Python に慣れるという点でも手を動かして考える機会にして頂ければ幸いです。

今回は、Python で学び直す数学【統計編】の確認をしていきます。

○ Python ライブラリ～Numpy, Pandas とは？

まず、読み込んだデータを統計情報として利用するための Python ライブラリについてみていきましょう。ここでは「Numpy」と「Pandas」を紹介します。

○ Numpy とは？
科学計算のための基本的なパッケージで、アレイを作成するのに便利なライブラリ。

Numpy を利用するには、以下のようにインポートを行います。As キーワードを使用して np で呼び出せるようにします。

In

```
import numpy as np
```

▹ 使用例

```
# 1 次元配列の作成
a = np.array( [1, 2, 3] )

# 2 次元配列の作成
b = np.array([ [1, 2, 3], [4, 5, 6] ])

# 等差数列の配列を返す（arange）
np.arange( 1, 11 )

Out >>> array( [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] )

np.arange( 1, 11, 2)

Out >>> array( [1, 3, 5, 7, 9] )
```

NumPy を使用することで、アレイや乱数生成など様々なことができるようになります。通常の Python で処理を行うよりもずっと早く処理できるので、大量のデータを扱う（機械学習など）場合に NumPy が利用されます。

○ Pandas とは？

Python でデータ処理をするために作られた高機能なライブラリ。代表的な使い方として Series や DataFrame を使ったデータの処理方法があります。

Pandas を利用するには、以下のようにインポートを行います。As キーワードを使用して pd で呼び出せるようにします。

In

```
import pandas as pd
```

▷ 使用例

```
# 1 次元データの利用
ser = pd.Series( [10, 20, 30, 40] )
ser

# 2 次元データの利用
df = pd.DataFrame( [10, “a”, True],
                   [20, “b”, False],
                   [30, “c”, False],
                   [40, “d”, True] )
df
```

csv や Excel のデータを読み込んだり、列や行を削除したり、フィルターをかけて抽出をしたり、
Excel やデータベース言語の SQL でできることが Pandas の機能にあります。

○ CSV ファイルの読み込み

次に、統計として扱う CSV データを読み込む方法について、見ていきましょう。
以下のような CSV データを読み込む例を考えてみましょう。

問題

testScore.csvを読み込むプログラムを実行して、
5番目までのデータの表示を確認(headを利用)し、また数学の全データを参照できることを確認してください。

▷ testScore.csv

	A	B	C
1	数学	英語	
2	69	50	
3	87	72	
4	56	82	
5	63	76	
6	53	49	
7	91	83	
8	74	77	
9	68	56	
	略		
20	58	32	
21	86	55	
22	37	63	

In

```
import pandas as pd

# testScore.csv  の読み込み
col_names = ['数学','英語']
data = pd.read_csv('testScore.csv', names = col_names, encoding='SHIFT-JIS')

# 5 件分の内容を確認
data.head()

#  数学の全データを参照
data['数学']
```

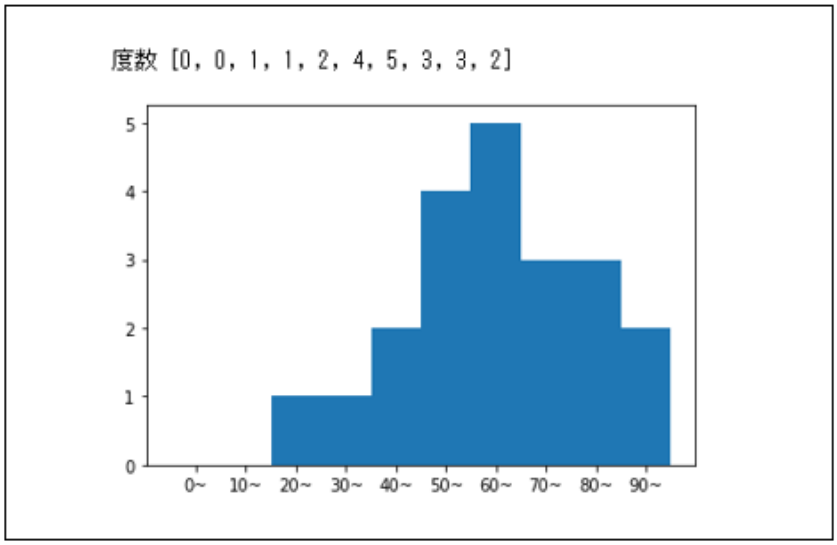
Out

	数学	英語
0	数学	英語
1	69	50
2	87	72
3	56	82
4	63	76

```
0     数学
1     69
2     87
3     56
4     63
5     53
6     91
7     74
8     68
9     85
10    62
11    41
12    73
13    46
14    64
15    94
16    71
17    24
18    55
19    58
20    86
21    37
Name: 数学, dtype: object
```

○　度数分布表を描画

問題
　読み込んだ testScore.csv のファイルから
　以下のような「数学の得点」に関する度数分布表を作成してください。
　今回は、プログラムを実行して、結果を確認してください。



In

```
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd

# testScore.csv の読み込み
nameColumns = ['数学', '英語']
testData = pd.read_csv('testScore.csv', header = None, skiprows=1, names = nameColumns, encoding='SHIFT-JIS')

# 各階層に含まれる度数を数える
hist = [0]*10
for data in testData['数学']:
    if int(data) < 10: hist[0] += 1
    elif int(data) < 20: hist[1] += 1
    elif int(data) < 30: hist[2] += 1
    elif int(data) < 40: hist[3] += 1
    elif int(data) < 50: hist[4] += 1
    elif int(data) < 60: hist[5] += 1
    elif int(data) < 70: hist[6] += 1
    elif int(data) < 80: hist[7] += 1
    elif int(data) < 90: hist[8] += 1
    elif int(data) <= 100: hist[9] += 1

print('度数', hist)

# 度数分布図
x = list(range(1, 11))          # x 軸の値
labels = ['0~', '10~', '20~', '30~', '40~', '50~', '60~', '70~', '80~', '90~']

# x 軸の目盛りラベル
plt.bar(x, hist, tick_label=labels, width=1)      # 棒グラフを描画
plt.show()
```

○ 統計指標（平均値，中央値，最頻値）

ここで、添付演習問題の「統計指標」シートの問題を解いてみましょう。
Numpy を使った平均値と中央値の求め方を調べて、下線部に入る式を埋めてみましょう。

問題
「CSVファイル読込」の方法に倣って、testScore.csv を読み込んだ後、
平均値、中央値（最頻値）をそれぞれ求めたいと思います。
下線部①②に入る式を記載してください。

(注) インポート方法

```
import pandas as pd
# testScore.csv の読み込み
dat = pd.read_csv('testScore.csv', encoding='SHIFT-JIS')
```

```
import pandas as pd
import numpy as np

# testScore.csv の読み込み
dat = pd.read_csv('testScore.csv', encoding='SHIFT-JIS')

# 平均値を求める
print('平均値', np.①_____(dat['数学']) )

# 中央値を求める
print('中央値', np.②_____(dat['数学']) )

# 最頻値を求める
sameCount = np.bincount(dat['数学'] )        # 同じ値の個数を数える
mode = np.argmax(sameCount)                  # sameCount の中で最も大きな値を取得
print('最頻値', mode)
```

○ 分散と標準偏差

ここで、添付演習問題の「分散と標準偏差」シートの問題を解いてみましょう。
Numpy を使った分散と標準偏差の求め方を調べて、下線部に入る式を埋めてみましょう。

問題
「CSVファイル読込」にて読み込んだ testScore.csv の
数学と英語のデータのそれぞれにおいて、
Numpyの分散を求める関数と、標準偏差を求める関数を使って、
ばらつきを数値で求めてください。
今回は、以下の下線部①～③に入る式を記載してください。

```
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# testScore.csv の読み込み
nameColumns = ['数学', '英語']
testData = pd.read_csv('testScore2.csv', header = None,  skiprows=1, names =
nameColumns, encoding='SHIFT-JIS')

print('数学----------')
print('平均：', np.①_____(testData['数学']))
print('分散：', np.②_____(testData['数学']))
print('標準偏差：', np.③_____(testData['数学']))

print('英語----------')
print('平均：', np.①____(testData['英語']))
print('分散：', np.②____(testData['英語']))
print('標準偏差：', np.③____(testData['英語']))
```

○ 偏差値を算出

ここで、添付演習問題の「偏差値算出」シートの問題を解いてみましょう。

問題

模試の結果が下表のような場合に
6月と1月にそれぞれ受験したそれぞれの偏差値を求めてください。
このとき、(偏差値を求める与式) = (点数 − 平均) / 標準偏差 × 10 + 50 とする。
＊平均点を取得した際の偏差値を50とする

表: 模試の結果(6月と1月)

	6月	1月
得点	425点	346点
平均	390点	290点
標準偏差	60	60

▹ 偏差値の計算式

```
>>> def deviation_value(score, heikin, hyoujyunhensa ):
        return (score - heikin) / hyoujyunhensa * 10 + 50
```

==========

以前学校で学んできた内容をもとに Python でスクリプトを実行しながら確認できるのは面白いなと感じる方もいらっしゃるかもしれません。自分にできる範囲のものから少しずつ Python にも挑戦してみようかなと思っていただければ幸いです。

以上となります。

参考文献：

・谷尻かおり『文系プログラマーのための Python で学び直す高校数学』日経 BP 社（2021 年）