クレジット:
UTokyo Online Educationデータマイニング入門 2018 森　純一郎

東京大学
THE UNIVERSITY OF TOKYO

CC BY NC ND

授業中で配布した.ipynbファイルでは、課題を完成させるためのコーディングのガイドおよび動作確認用の関数が含まれていますが、本ファイルでは省略されています。

課題2 データの集計

Q1

中央値はデータの中心傾向を観察するための記述統計の指標です。同じくデータの中心傾向の指標である平均値と比べると、データの外れ値の影響を受けにくいという特徴があります。1変数のn個のデータを考えた時、中央値を求めるにはまず、データの値の大きさの順に並び替えます。次に、データの数が奇数個の場合は、大きさの順に並び替えた$(n+1)/2$番目の値、データの数が偶数個の場合は、大きさの順に並び替えた$n/2$番目と$n/2+1$の値の平均の値、をそれぞれデータの中央値とします。

分散はデータのばらつきを観察するための記述統計の指標です。個々のデータがデータ全体の平均からどらぐらいばらついているかを表します。以下の通り、個々のデータの値x_iとデータ全体の平均の値$\bar{x}$の差の2乗和をデータ数nで割ったものが分散になります。

$$\frac{1}{n}\Sigma_{i=1}^{n}(x_i-\bar{x})^2$$

0から100までの整数を要素とする任意の長さのリストを引数で入力として受け取り、以下を要素とするリストを返す `avg_med_var()` 関数を完成させてください。

- 入力リストの全ての要素の平均値
- 入力リストの全ての要素の中央値
- 入力リストの全ての要素の分散

In [ ]:
```
def avg_med_var(input_list):

```

`avg_med_var()` 関数が完成したら、以下のセルを実行して動作を確認してください。

In [ ]:
```
print(avg_med_var([1,4,3,2,5,6,8,7,9,10]))
print(avg_med_var([1,4,3,2,5,6,8,7,9,10,11]))
```

Q2

以下のような形式の"user_score.csv"ファイルを読み込み、データフレーム score を作成します。

```
## user_score.csvファイル
user, kokugo, shakai, sugaku, rika
1, 30, 43, 51, 63
2, 39, 21, 49, 56
...
```

In [ ]:
```
import pandas as pd
score = pd.read_csv('user_score.csv')
score.head(5)
```

データフレームの desribe() メソッドを使って、各教科ごとの記述統計を確認してください

In [ ]:
```
score.describe()
```

Q2-1

データフレーム score から各教科の得点がすべて平均点以上であるようなユーザの数を返す above_mean() 関数を作成してください

In [ ]:
```
def above_mean(df):

```

above_mean() 関数が完成したら、以下のセルを実行して動作を確認してください。条件満たすユーザは46人となります。

In [ ]:
```
above_mean(score)
```

Q2-2

ユーザごとの全教科の得点の総和の値を表す'sum'列をデータフレーム score に新たに追加したデータフレームを返す score_sum() 関数を作成してください。以下のようにして、データフレームの指定した列の行方向の値の和を持つシリーズオブジェクト作ることができます。

データフレーム[[列1,列2,...,]].sum(axis=1)

In [ ]:
```
def score_sum(df):

```

score_sum() 関数が完成したら、以下のセルを実行して動作を確認してください。'sum'列が追加されていることを確認してください。

In [ ]:
```
score = score_sum(score)
score.head(5)
```

全教科の合計得点の上位3人の'user'を1位、2位、3位の順に要素として持つリストを返す score_top3() 関数を作成させてください。

In [ ]:
```
def score_top3(df):

```

score_top3() 関数が完成したら、以下のセルを実行して動作を確認してください。 [39, 4, 50] の'user'が合計得点の上位3人です。

In [ ]:
```
score_top3(score)
```

Q2-3

データフレーム score の'sum'列を対象に、以下を階級幅として各階級の度数を要素として持つリストを返すscore_hist()関数を完成させてください。

- 0以上、50以下
- 51以上、100以下
- 101以上、150以下
- 151以上、200以下
- 201以上、250以下
- 251以上、300以下
- 301以上、350以下
- 351以上、400以下

In [ ]:
```
def score_hist(df):

```

score_hist() 関数が完成したら、以下のセルを実行して動作を確認してください。 [4, 20, 39, 35, 29, 24, 13, 2] が'sum'の度数分布です。

In [ ]:
```
score_hist(score)
```

Q2-4

以下のような形式の"user_class.csv"ファイルを読み込み、データフレーム `user_class` を作成します。

```
## user_class.csvファイル
'user', 'class'
1, 'C'
2, 'C'
3, 'B'
...
```

In [ ]:
```
user_class = pd.read_csv('user_class.csv')
user_class.head(5)
```

'user'の列の値をキーに、2つのデータフレーム、user_classとscore、を結合して新しいデータフレームを作成した上で、データフレームの `groupby` メソッドを使って、'class'をインデックス、'class'ごとの'sum'の最高値を列として持つシリーズオブジェクトを返す `score_by_class()` 関数を完成させてください。

In [ ]:
```
def score_by_class(df1, df2):

```

`score_by_class()` 関数が完成したら、以下のセルを実行して動作を確認してください。 `{'A': 361, 'B': 349, 'C': 359}` が'class'ごとの'sum'の最高値です。

In [ ]:
```
score_by_class(score, user_class).to_dict()
```