

# 情報処理概論

九州大学 工学部地球環境工学科 講義資料 担当:木村

05 リスト(配列・多次元配列)



## 多数のデータ処理

- (要求) 10,000人分の成績データをもらって、合計点の大きい順に並べ替えて出力したい。
- (方法) ひとまず、全部のデータを変数に入れておく。
- (問題) しかし、 $d_1, d_2, d_3, \dots, d_{10000}$ ;  
のように、10,000個も変数を書けるか？
- (解決策) 「リスト」を使おう!!

C言語では「配列」

Pythonの「リスト」は、配列よりも、ずっと高機能

# リスト型変数

リスト ” list”

例: list1 = []

何も入ってない袋

例: list2 = [1, 2, 3, 5, 8]

list3 = [ “dog” , “cat” , “fish” ]

好きな数字や文字列を好きなだけ入れられる

ただし可能な処理は限られる

例: list4 = [ “A” , 2] #複数の型が混在してもOK

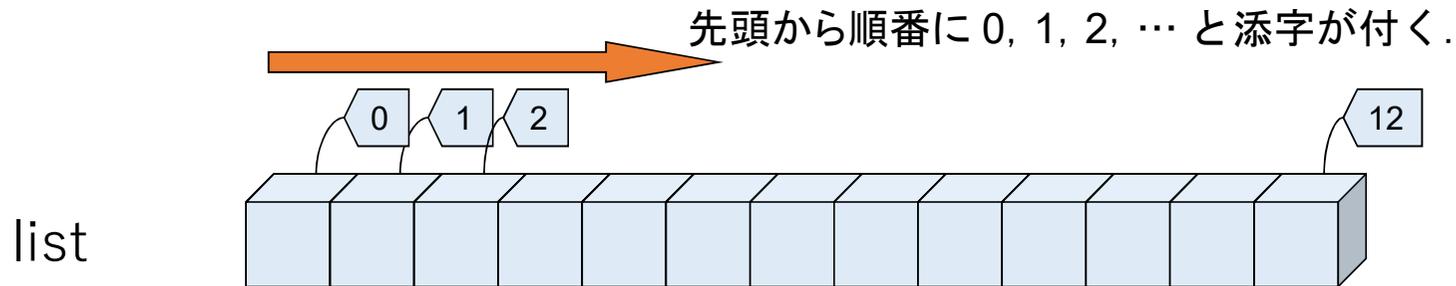
list5 = [ [1, 2, 3], [2, 5], , , . ] #入れ子にも出来る

後で「行列」として説明

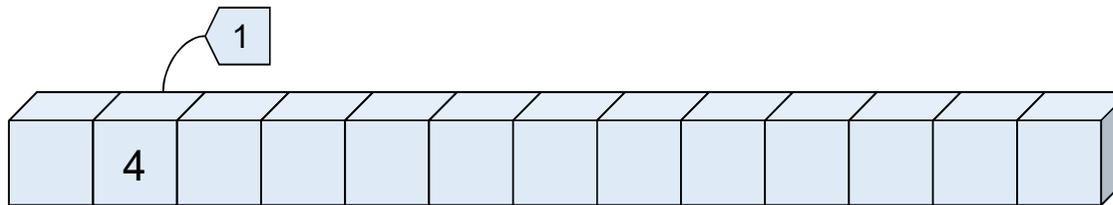
# リスト型のイメージ

list = [2, 4, 6, 8, 10, 12, 14]

7個の箱が並ぶと考える.



list[1] = 4 ← 0からカウント!



# リスト型の演算

- + : リストの連結
- \* : 繰り返し

例:

```
lis=[1, 2, 3]
print(lis + lis)
print(lis * 3)
```

実行結果: [1, 2, 3, 1, 2, 3]  
[1, 2, 3, 1, 2, 3, 1, 2, 3]

- リストの一部取り出し

例:

```
lis=[1, 2, 3, 4, 5]
print(lis[0]) #最初の要素を取り出す
print(lis[0:2]) #0番目から(2-1)番目を取り出し
print(lis[-1]) #リストとして返す(スライス)
```

実行結果: 1  
[1, 2]  
5

#リストの最後の要素を取り出す

# リストによる行列の表現

- 行列をリストの組(リスト)で表現する

■ 行列  $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$  各行をリストとみなす

入れ子=リストを要素としたリスト

例:

$X = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

$X[0] = [1, 2, 3]$ ,  $X[1] = [4, 5, 6]$ ,  $X[2] = [7, 8, 9]$  (i 行目)

$X[0][0] = 1$ ,  $X[1][2] = 6$ ,  $X[2][1] = 8$  (i 行 j 列目の要素)

# リストの要素の追加・削除

- appendメソッド: 末尾に要素を追加する

例:

```
list=[1,2,3]
list.append(5)
print(list)
```

実行結果: [1, 2, 3, 5]

- 特定の位置の要素を変更・削除もできる

例:

```
list=[1,2,3,4,5]
list[4]=6
list[0:2]=[]
print(list)
```

#4番目の要素を6に変更

#始め2つの要素を削除

実行結果: [3, 4, 6]

- リストの指定した位置へ要素を挿入

例:

```
mylist = ["Blue", "Red", "Green"]
mylist.insert(2, "White")
mylist.insert(0, "Black")
print(mylist)
```

#"White"を  
インデックス2の前に挿入

#"Black"を  
インデックス0の前に挿入

実行結果: ["Black", "Blue", "Red", "White", "Green"]

# リストのメソッド・関数(1)

- ・countメソッド 例) num = abc\_list.count(x) # リストabc\_list中の要素xの出現回数を出す
- ・len 関数 例) num = len( mylist ) # リストmylist中の要素の個数を出す
- ・min 関数 例) num = min( list\_1 ) # リストlist\_1中の最小の要素を出す
- ・max 関数 例) num = max( list\_2 ) # リストlist\_2中の最大の要素を出す
- ・sum 関数 例) num = sum( list\_3 ) # リストlist\_2中の要素の和を出す
- ・sort メソッド 例) list\_test.sort() # リスト list\_test を要素の小さい順にソートする
- ・sorted 関数 例) list\_4 = sorted( list\_test ) # リスト list\_test をソートした新しいリストlist\_4を作る  
sorted( リスト名, key=関数名 ) # リスト中の要素を関数名で表された関数の値の小さい順にソート

元のリストはそのまま

```
例) list_test=[5,-4,-7,6]
    print( sorted(list_test) )          実行結果 → [-7,-4,5,6]
    print( sorted(list_test, key=abs) ) 実行結果 → [-4,5,6,-7]
```

## リストのメソッド・関数(2) **Randomモジュールの利用**: import random とすること

・random.choiceメソッド リスト中の要素をランダムに1つ選んで返す

例)

```
import random
```

```
abc_list = [ 2,3,5,7,11,13,17]
```

```
x = random.choice( abc_list ) # リストabc_list中の要素をランダムに1つ選んでxへ代入
```

・random.shuffle メソッド リスト中の要素をランダムに並び替える

例)

```
import random()
```

```
List_test = [1,2,3,4,5,6,7,8]
```

```
random.shuffle(List_test)
```

```
print( List_test )                    実行結果:    [4,5,2,1,3,8,6,7]
```

・random.sampleメソッド リスト中の要素をランダムに非復元抽出した新しいリストを生成する

例)

```
import random()
```

```
List_test = [1,2,3,4,5,6,7,8]
```

```
List_2 = random.sample(List_test,3)
```

```
print( List_2,List_test )            実行結果:    [5,3,8][1,2,3,4,5,6,7,8]
```

元のリストはそのまま  
重複なし

# 文字列からのリストの生成

splitメソッド:

- `str.split()`: 文字列`str`を空白(複数まとめて)に基づいて分割し、部分文字列のリストを返す

```
sentence="This is a sentence"  
print(sentence.split())
```

```
['This', 'is', 'a', 'sentence']
```

- `str.split(sep=区入り文字列)`: 文字列`str`を区切り文字列に基づいて分割し、部分文字列のリストを返す

```
sentence="One-Two-Three"  
print(sentence.split(sep="-"))
```

```
['One', 'Two', 'Three']
```

詳細: <https://docs.python.org/ja/3/library/stdtypes.html>

# リストの条件文

- `x in list_1` リスト`list_1`に要素`x`が含まれれば`true`, さもないと`False`

例:

```
mylist = [2,3,4,5,6,8,10,11,12]
X=1
if( X in mylist ):
    print( "mylist中に", X, "が要素として含まれます")
else:
    print( "mylist中に", X, "が要素として含まれません")
```

- `for 変数 in list_2` `for`文において, `range`の代わりにリストを用いることができる

例:

```
for w in ["Monday", "Tuesday", "Wednesday"]:
    print(w)
```

実行結果: Monday  
Tuesday  
Wednesday

## for文を用いたリストの生成: リスト内包表記

例: `x=[1,2,3,...,10]` を生成する

例1:  
普通のforループ

```
x=[] #空リストを生成
for i in range(1,11): #i=1,...,10について繰り返し
    x.append(i) #リストxの末尾にiを追加
print(x)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

例2:  
リスト内包表記

```
x=[i for i in range(1,11)]
print(x)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# リスト内包表記と if 文

リスト内包表記に if 文を用いることもできる：

- for 変数 in (リスト, range等) if (条件文)

- 例：  $i = 1, \dots, 10$  のうち、  
奇数であるものを要素としたリストを生成

```
x=[i for i in range(1,11) if i % 2==1]  
print(x)
```

```
[1, 3, 5, 7, 9]
```

# 【例】メジアン(中央値)の計算プログラム

```
print("メジアンを計算します。カンマで区切って数字を入力してください")
moji=input() #----キーボードから文字列を入力
moji_list=moji.split(sep=",") #----文字列mojiをカンマで分割してリストmoji_list生成
num_list=[] #-----数字のリストを用意 最初は空
for index in range( len(moji_list) ):
    num_list.append( float( moji_list[index] ))
print("入力された数字のリスト",num_list)
num_list.sort() #-----リストをソート
print("ソート結果",num_list)
median=0.0 #-----計算すべき変数を用意
if len( num_list )%2==0 :
    #-----数字のリストnum_listの要素数が偶数
    med_index=int( (len(num_list)/2) -1 )
    median=(num_list[med_index]+num_list[med_index+1])/2
else:
    #-----数字のリストnum_listの要素数が奇数
    med_index=int( (len(num_list)-1)/2 )
    median=num_list[med_index]
print("メジアン =",median)
```

要素が文字列のリストmoji\_listから、要素を1つずつ取り出して数字に変換し、新しいリストnum\_listへ加えていく

リストから要素を取り出す場合に指示するインデックスの数字はint型 それ以外はエラー  
要素数を2で割ったりするとfloat型になるのでint型に変換する

# 課題: 統計量の計算プログラム

- 1) 標準入力から何個かの数値をカンマ区切りで入力 (例:1,6,4),
- 2) split メソッド を用いて入力の数字文字列からなるリストを作成 (例 words=["1", "6", "4"]),  
float関数を利用して上記リストより数値からなるリストを作成
- 3) それらを小さい順にソートして表示、
- 4) それらをシャッフルして表示、
- 5) それらの**最大値**, **最小値**, **平均**, **標準偏差**, **メジアン**を計算・表示

以上のプログラムを作成せよ. **【注意】statisticsパッケージを使用しないこと。**

ヒント1: 文字列から数値を生成するには以下を参考

```
moji = input()
```

キーボードから文字列入力し変数mojiへ格納

```
a = float( moji )
```

文字列をfloat(浮動小数)型へ変換し、変数xへ格納

プログラムの先頭行にコメント文で自分の氏名と学籍番号を入れておくこと

```
# ← この記号より左側の文字列はコメントになる
```

## まとめ

リストの生成、追加、削除

リストの演算、メソッド、関数

リストによる行列の表現

文字列を分割して要素を生成してリスト化: **splitメソッド**

リスト内包表記

## 第5回 レポート課題提出方法

課題のプログラムを

下記の課題提出用フォルダへ、課題の番号と提出者が分かるようにファイル名を以下のようにしてアップロードせよ

第5回1TE19xxxZ名前.py

[https://share.iii.kyushu-u.ac.jp/public/IRbwAAVITI5A2X4BE45t6TqQIE0UQSQUI5Bap\\_kZ\\_sjy](https://share.iii.kyushu-u.ac.jp/public/IRbwAAVITI5A2X4BE45t6TqQIE0UQSQUI5Bap_kZ_sjy)

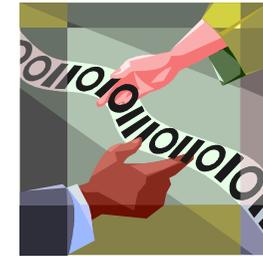
講義資料、および上記フォルダへのリンクは下記ホームページから

<http://sysplan.nams.kyushu-u.ac.jp/gen/edu/InfoProcess/2019/index.html>

## 【参考】データの統計量

統計データ: 観察の対象について得られた**測定値の集合**

統計分析: 統計データに含まれる**規則性**を見出す  
平均(mean) 分散(variance)  
標準偏差(standard deviation) } 特性値・統計量



nコのデータ:  $x_1, x_2, x_3, \dots, x_n$

標本(サンプル)

平均: 
$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i$$

分散: 
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

標準偏差: 
$$\sigma = \sqrt{\sigma^2}$$

データの散らばり  
具合を表現する

平均 (mean)  $\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i$

数学的には、各要素からの2乗距離の差

$$\sum_{i=1}^n (x_i - c)^2$$

を最小化する  $c$  を平均  $\bar{x}$  としている。

## 中位数 (中央値, メディアン: median)

データを大きさ順に並べたとき、ちょうど中央に位置するデータの値

データの数  $n$  が奇数のとき、 $(n+1)/2$  番目の値、

データの数  $n$  が偶数のとき、 $n/2$  番目と  $(n/2)+1$  番目の中間の値

数学的には、各要素からの絶対距離の和

$$\sum_{i=1}^n |x_i - c|$$

を最小化する  $c$  を中位数 としている。

この統計量は、解析が困難だが データの例外値(ノイズ)に影響されない という特徴がある。

## 最頻値 (モード: mode)

度数分布表で度数が最も多い階級値  
2つ以上存在することもある

工学的に非常に  
有用

# 分散の性質

平均値を計算後でない  
分散を計算できない  
→各データを2回読む

平均値と分散を同時に計算できる  
→各データを1回読めばよい

工学的  
に有用

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \left( \sum_{i=1}^n x_i^2 \right) - (\bar{x})^2$$

分散の定義式

2乗の平均値

(平均値)の2乗

【証明】

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

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

$$= \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - 2\bar{x} \left( \frac{1}{n} \sum_{i=1}^n x_i \right) + \left( \frac{1}{n} n (\bar{x})^2 \right)$$

$$= \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - 2(\bar{x})^2 + (\bar{x})^2$$

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

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$