クラスター分析（非階層型）

はじめに

クラスター分析の 2 つの使用例をご紹介します。1 つ目は、連続値のデータを用いて cluster kmeans コマンドを実行します。2 つ目の例では、二値データを用いて cluster kmedians コマンドを実行します。2 つのコマンドは同じように動作しますが、データのタイプが異なります。

例題 1:

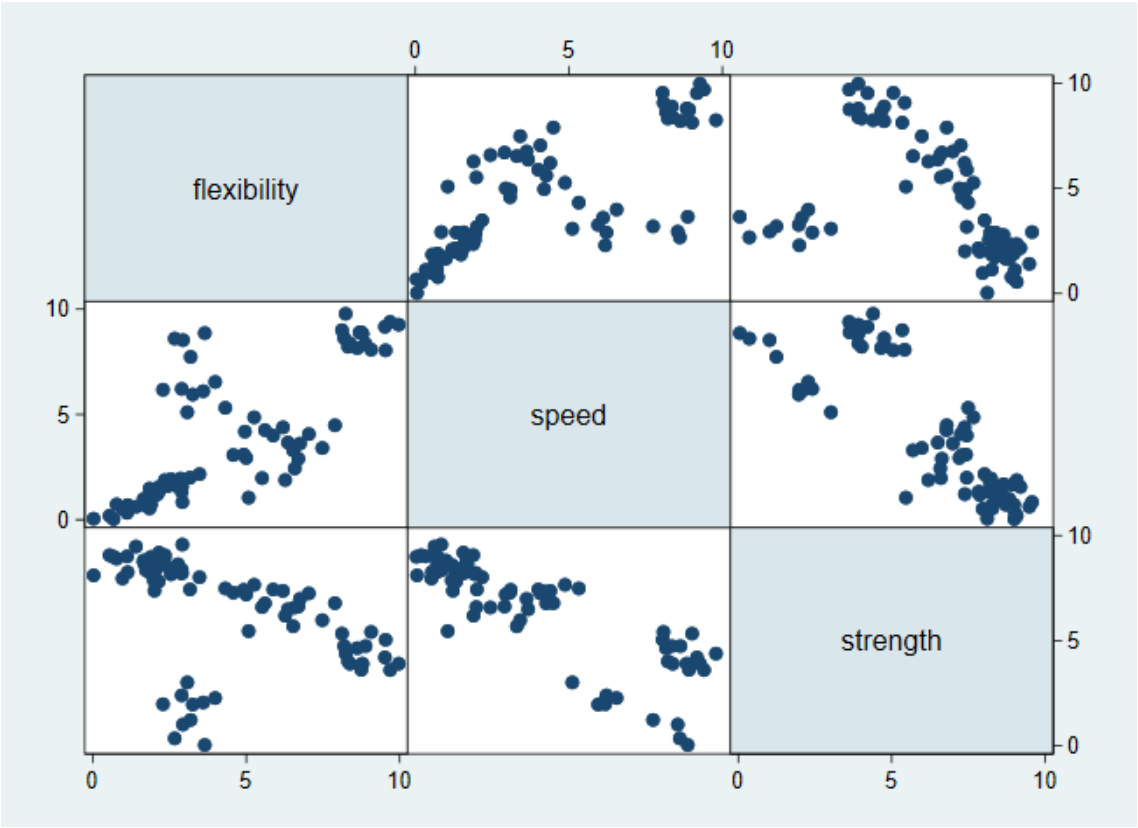
体育の授業で 80 人の生徒に対して、柔軟性（flexibility）、俊敏性（speed）、力強さ（strength）を計測したとしましょう。彼らの運動能力を改善する最良のトレーニングを受けさせるため、生徒の運動能力をもとに 4 つのクラスに分けることを考えます。

以下にデータの要約とグラフを表示します。

```
. use https://www.stata-press.com/data/r17/physed
. summarize flex speed strength
```

Variable	Obs	Mean	Std. dev.	Min	Max
flexibility	80	4.402625	2.788541	.03	9.97
speed	80	3.875875	3.121665	.03	9.79
strength	80	6.439875	2.449293	.05	9.57

```
. graph matrix flex speed strength
```



生徒のパフォーマンスは広い範囲でばらついていることがわかります。グラフからいくつかの特定のグループが存在しているように見えますが、グループ分けを上手く行うことができそうです。

クラスター分析を実行して、クラスのアシスタントごとに 1 つずつ、合計 4 つのグループを作成することにしました。

あなたは過去に kmeans クラスタリングを行った経験があり、一般的な距離の絶対値が好ましいと思っています。

これまではクラスター分析における初期値について気にしていませんでしたが、今回は分析を再実行した際に同じ結果を再現できるかについても確認したいと考えています。オプション krandom()を指定して、グループの中心の初期点としてランダムに k 個の観測値を抽出することにしました。再現性のために乱数の種を指定します。また、オプション keepcenters を指定して、4 つのグループの平均がデータセットの最後に付加されるようにしました。

```
. cluster k flex speed strength, k(4) name(g4abs) s(kr(385617)) mea(abs) keepcen
. cluster list g4abs
```

```
g4abs  (type: partition,  method: kmeans,  dissimilarity: L1)
      vars: g4abs (group variable)
     other: cmd: cluster kmeans flex speed strength, k(4) name(g4abs) s(kr(385617)) mea(abs) keepcen
            varlist: flexibility speed strength
            k: 4
            start: krandom(385617)
            range: 0 .
```

. table g4abs

	Frequency
Cluster ID	
1	15
2	20
3	10
4	35
Total	80

. list flex speed strength in 81/L, abbrev(12)

	flexibility	speed	strength
81.	8.852	8.743333	4.358
82.	5.9465	3.4485	6.8325
83.	3.157	6.988	1.641
84.	1.969429	1.144857	8.478857

. drop in 81/L

. tabstat flex speed strength, by(g4abs) stat(min mean max)

Summary statistics: Min, Mean, Max
Group variable: g4abs (Cluster ID)

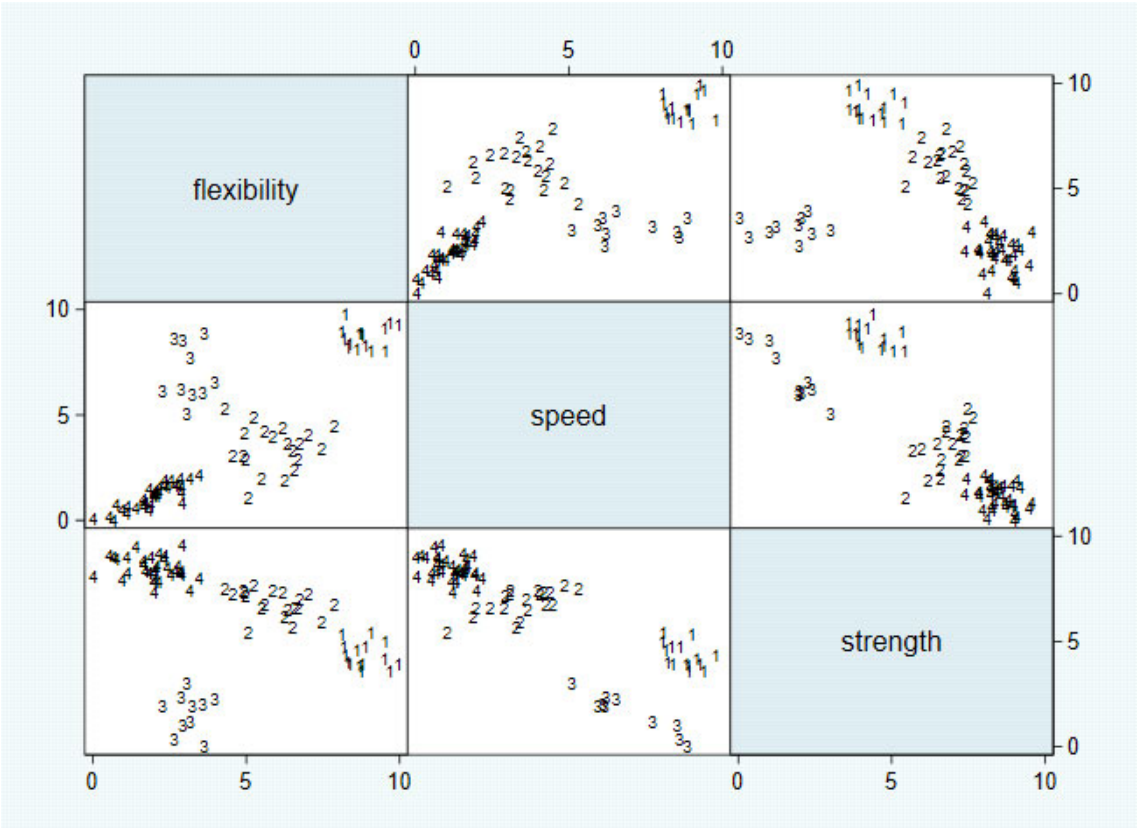
g4abs	flexib~y	speed	strength
1	8.12	8.05	3.61
	8.852	8.743333	4.358
	9.97	9.79	5.42
2	4.32	1.05	5.46
	5.9465	3.4485	6.8325
	7.89	5.32	7.66
3	2.29	5.11	.05
	3.157	6.988	1.641
	3.99	8.87	3.02
4	.03	.03	7.38
	1.969429	1.144857	8.478857
	3.48	2.17	9.57
Total	.03	.03	.05
	4.402625	3.875875	6.439875
	9.97	9.79	9.57

観測値の最後の 4 行を確認した後、4 つのグループに関する最小値、最大値、平均値を確認することにしました。観測行の最後の 4 行を除いて、tabstat コマンドで統計量を表示しました。

15 人の生徒が含まれるグループ 1 は、柔軟性と俊敏性が優れており、力強さに関するトレーニングが必要であるとわかりました。20 人の生徒が含まれるグループ 2 は、特に俊敏性のトレーニングが必要ですが、その他の項目もトレーニングが必要なようです。生徒が 10 人しかいない小数のグループ 3 は、柔軟性と力強さのトレーニングが必要です。35 人の生徒が属する最大のグループ 4 は、強さは十分ですが、柔軟性と早さの両方に問題を抱えるグループです。

グループ番号をシンボルとしてグラフを生成してみましょう。

```
. graph matrix flex speed strength, m(i) mlabel(g4abs) mlabpos(0)
```



グラフに示されているように、各グループはかなり明確に分かれて見えます。しかし、各グループの生徒数がなるべく同じになるようにしたいと考えています。3 グループや 5 グループでもクラスタリングを行うことを考えました。理由はありませんが、3 グループのクラスタリングにおける初期点として初めの k 行を使用し、5 グループのクラスタリングにおける初期点としてデータのレンジ内で乱数を生成して使用しました。

```
. cluster k flex speed strength, k(3) name(g3abs) start(firstk) measure(abs)
. cluster k flex speed strength, k(5) name(g5abs) start(random(33576)) measure(abs)
. table g3abs g4abs, totals(g3abs)
```

	Cluster ID				
	1	2	3	4	Total
Cluster ID					
1			10		10
2		18		35	53
3	15	2			17

```
. table g5abs g4abs, totals(g5abs)
```

	Cluster ID				
	1	2	3	4	Total
Cluster ID					
1	15				15
2		9			9
3			10		10
4		11			11
5				35	35

3 グループのクラスタリングでは、生徒数を均等にするという問題は悪化してしまいました。5 グループのクラスタリングでは、1 つのグループに 35 人の生徒が属するグループがあり、その他のグループの生徒数よりも多くなっています。このクラスにおいては 4 グループが最適であるようです。4 つ目のグループに対してはアシスタントを割当てることを検討しましょう。

この例において 4 グループを生成するために使用したコマンドにおける乱数の種を変えて、結果を確認したいと思うでしょう。これらのグループ分けは明確に定義されたものではないため、異なる初期点でクラスター分析を実行することでより良いグループ分けが見つかるかもしれません。

例題 2：

あなたは、女性クラブを創設しようと考えています。コミュニティを通して集まった 30 人の女性が加入の申込みを送ってきました。スポーツ、音楽、読書、趣味などに関する 35 問の二択の質問に回答してもらいました。35 問（35 変数）に関する要約は下記の通りです。

```
. use https://www.stata-press.com/data/r17/wclub, clear
. describe
```

```
Contains data from https://www.stata-press.com/data/r17/wclub.dta
 Observations:            30
    Variables:            35                  1 May 2020 16:56
```

Variable name	Storage type	Display format	Value label	Variable label
bike	byte	%8.0g		Enjoy bicycle riding Y/N
bowl	byte	%8.0g		Enjoy bowling Y/N
swim	byte	%8.0g		Enjoy swimming Y/N
jog	byte	%8.0g		Enjoy jogging Y/N
hock	byte	%8.0g		Enjoy watching hockey Y/N
foot	byte	%8.0g		Enjoy watching football Y/N
base	byte	%8.0g		Enjoy baseball Y/N
bask	byte	%8.0g		Enjoy basketball Y/N
arob	byte	%8.0g		Participate in aerobics Y/N
fshg	byte	%8.0g		Enjoy fishing Y/N
dart	byte	%8.0g		Enjoy playing darts Y/N
clas	byte	%8.0g		Enjoy classical music Y/N
cntr	byte	%8.0g		Enjoy country music Y/N
jazz	byte	%8.0g		Enjoy jazz music Y/N
rock	byte	%8.0g		Enjoy rock and roll music Y/N
west	byte	%8.0g		Enjoy reading western novels Y/N
romc	byte	%8.0g		Enjoy reading romance novels Y/N
scif	byte	%8.0g		Enjoy reading sci. fiction Y/N
biog	byte	%8.0g		Enjoy reading biographies Y/N
fict	byte	%8.0g		Enjoy reading fiction Y/N
hist	byte	%8.0g		Enjoy reading history Y/N
cook	byte	%8.0g		Enjoy cooking Y/N
shop	byte	%8.0g		Enjoy shopping Y/N
soap	byte	%8.0g		Enjoy watching soap operas Y/N
sew	byte	%8.0g		Enjoy sewing Y/N
crft	byte	%8.0g		Enjoy craft activities Y/N
auto	byte	%8.0g		Enjoy automobile mechanics Y/N
pokr	byte	%8.0g		Enjoy playing poker Y/N
brdg	byte	%8.0g		Enjoy playing bridge Y/N
kids	byte	%8.0g		Have children Y/N
hors	byte	%8.0g		Have a horse Y/N
cat	byte	%8.0g		Have a cat Y/N
dog	byte	%8.0g		Have a dog Y/N
bird	byte	%8.0g		Have a bird Y/N
fish	byte	%8.0g		Have a fish Y/N

クラブの初回打合せを計画する中で、あなたは共通の趣味を持つ女性ごとに 5 つのテーブル席を割り当てたいと考えています。同じ興味を持つ人たちが同じテーブルとなるように席を配置したいと思います。全ての変数は二値類似度となっており、共通の要素が占める割合を意味する Jaccard 係数を使用することにしました。

また、kmeans コマンドと kmedians コマンドによるクラスタリングで生成したグルーピングを調べることにしました。

```
. cluster kmeans bike-fish, k(5) measure(Jaccard) st(firstk) name(gr5)
. cluster kmed bike-fish, k(5) measure(Jaccard) st(firstk) name(kmedian5)
. cluster list kmedian5
```

```
kmedian5  (type: partition,  method: kmedians,  similarity: Jaccard)
      vars: kmedian5 (group variable)
     other: cmd: cluster kmedians bike-fish, k(5) measure(Jaccard) st(firstk)
                 name(kmedian5)
           varlist: bike bowl swim jog hock foot base bask arob fshg dart clas cntr jazz
                 rock west romc scif biog fict hist cook shop soap sew crft auto pokr
                 brdg kids hors cat dog bird fish
           k: 5
           start: firstk
           range: 1 0
```

kmeans コマンドと kmedians コマンドの初期点として初めの k 行を使用するため、オプション st(firstk)を指定します。

各手法で生成したグループはどのくらいのサイズで、結果はどのくらい似ているでしょうか。

```
. table gr5 kmedian5
```

	Cluster ID					
	1	2	3	4	5	Total
Cluster ID						
1	7					7
2	1	6				7
3			5			5
4				5		5
5	1		1		4	6
Total	9	6	6	5	4	30

cluster kmeans コマンドと cluster kmedians コマンドから得られる結果はよく合致しています。1 つのテーブルには 8 人が快適に座れるため、各グループの人数が 5 名から 7 名となっている cluster kmeans コマンドによって生成されたグルーピングを採用します。一方、cluster kmedians コマンドによって生成されたグルーピングは 4 名から 9 名のグループとなっています。