

# 主因子解を求める因子分析

青木繁伸

## 1 目的

因子分析を行う。

## 2 使用法

```
import sys
sys.path.append("statlib")
from pfa import pfa
pfa(dat, rotation="Varimax", eps1=1e-5, eps2=1e-5, max1=999, max2=999, factors=0,
    verbose=True)
```

因子負荷量や因子得点をプロットする。

```
import sys
sys.path.append("statlib")
from pfa import pfa_plot
pfa_plot(obj, type="fl", ax1=1, ax2=2, color="black", color2="blue")
```

### 2.1 引数

<code>dat</code>	データ行列または相関係数行列
<code>rotation</code>	因子軸の回転法 "Varimax" (デフォルト), "biQuartimax", "Quartimax", "Equimax", "None"
<code>eps1</code>	共通性の収束限界
<code>eps2</code>	因子軸の回転の収束限界
<code>max1</code>	共通性の推定のための収束計算の上限回数
<code>max2</code>	因子軸の回転を行う上限回数
<code>factors</code>	抽出する因子数
<code>verbose</code>	必要最小限のプリント出力をする (デフォルトは True)
<code>obj</code>	<code>pfa()</code> の戻り値
<code>type</code>	<code>type="fl"</code> (デフォルト) なら因子負荷量, <code>type="fs"</code> なら因子得点をプロットする。
<code>ax1</code>	横軸にとる因子の番号
<code>ax2</code>	縦軸にとる因子の番号
<code>color</code>	点の色
<code>color2</code>	点のそばに付けるテキストの色

## 2.2 戻り値の名前

"rotation"	因子軸の回転法
"r"	相関係数行列
"communality"	共通性
"f11"	回転前の因子負荷量行列
"eval1"	回転前の固有値 (因子負荷量の 2 乗の列和)
"cont"	各因子の寄与率
"cum"	各因子の累積寄与率
"f1"	回転後の因子負荷量行列
"eval2"	回転後の因子負荷量の 2 乗の列和
"cont2"	回転後の各因子の寄与率
"cum2"	回転後の各因子の累積寄与率
"fs"	因子得点
"name"	変数名

## 3 使用例

### 3.1 テストデータ

以下の 10 変数, サンプルサイズ 20 のデータを用いる。

結果の出力中に変数名を使うので, データはデータフレームで与えるのを基本とするが, 二次元配列または二重リストで与えてもよい (変数名を補完する)。

```
dat = [[935, 955, 926, 585, 1010, 925, 1028, 807, 769, 767],
       [817, 905, 901, 632, 1004, 950, 957, 844, 781, 738],
       [768, 825, 859, 662, 893, 900, 981, 759, 868, 732],
       [869, 915, 856, 448, 867, 874, 884, 802, 804, 857],
       [787, 878, 880, 592, 871, 874, 884, 781, 782, 807],
       [738, 848, 850, 569, 814, 950, 957, 700, 870, 764],
       [763, 862, 839, 658, 887, 900, 1005, 604, 709, 753],
       [795, 890, 841, 670, 853, 874, 859, 701, 680, 772],
       [903, 877, 919, 460, 818, 849, 884, 700, 718, 716],
       [761, 765, 881, 485, 846, 900, 981, 728, 781, 714],
       [747, 792, 800, 564, 796, 849, 932, 682, 746, 767],
       [771, 802, 840, 609, 824, 874, 859, 668, 704, 710],
       [785, 878, 805, 527, 911, 680, 884, 728, 709, 747],
       [657, 773, 820, 612, 810, 849, 909, 698, 746, 771],
       [696, 785, 791, 578, 774, 725, 932, 765, 706, 795],
       [724, 785, 870, 509, 746, 849, 807, 763, 724, 760],
       [712, 829, 838, 516, 875, 725, 807, 754, 762, 585],
       [756, 863, 815, 474, 873, 725, 957, 624, 655, 620],
       [622, 759, 786, 619, 820, 769, 807, 673, 698, 695],
       [668, 753, 751, 551, 834, 849, 807, 601, 655, 642]]

import sys
sys.path.append("statlib")
from pfa import pfa
```

```
a = pfa(dat)
```

Rotation: Varimax

	Fac.01	Fac.02	Communality
x1	-0.943	0.094	0.897
x2	-0.912	0.048	0.834
x3	-0.725	0.401	0.686
x4	0.124	0.323	0.120
x5	-0.643	0.124	0.429
x6	-0.252	0.723	0.586
x7	-0.398	0.405	0.323
x8	-0.493	0.359	0.372
x9	-0.257	0.694	0.548
x10	-0.226	0.469	0.271
Sq.Sum	3.256	1.810	
Cont.	32.6	18.1	
Cum.	32.6	50.7	

```
import pandas as pd

dat = pd.read_csv("data/pca.csv")

import sys
sys.path.append("statlib")
from pfa import pfa

a = pfa(dat)
```

Rotation: Varimax

	Fac.01	Fac.02	Fac.03	Communality
X1	-0.403	-0.308	-0.233	0.312
X2	-0.139	-0.359	0.135	0.166
X3	-0.272	-0.623	-0.070	0.467
X4	0.210	-0.263	-0.748	0.673
X5	-0.306	0.196	-0.657	0.563
X6	-0.902	-0.077	0.016	0.819
X7	-0.290	0.267	-0.440	0.349
X8	-0.156	-0.032	0.005	0.025
X9	-0.356	0.104	0.014	0.138
X10	0.185	-0.200	-0.328	0.182
X11	0.231	-0.756	0.027	0.625
X12	0.049	-0.312	-0.158	0.125
Sq.Sum	1.531	1.518	1.395	
Cont.	12.8	12.6	11.6	
Cum.	12.8	25.4	37.0	

```
from pfa import pfa_plot
pfa_plot(a)
```

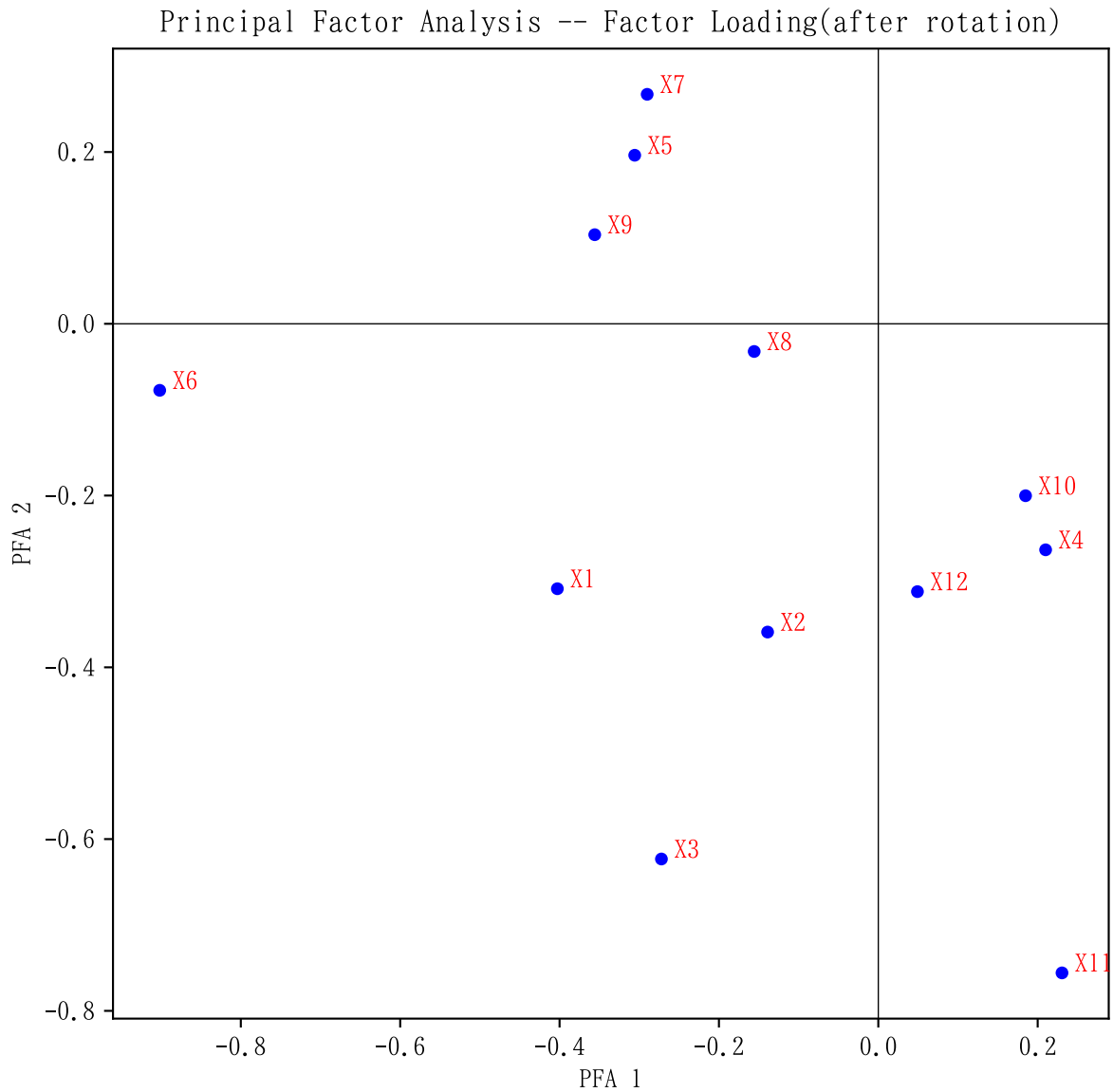


図1 因子負荷量のプロット

### 3.2 因子得点を表示する

```
pfa_plot(a, type="fs")
```

### 3.3 バイプロットを表示する

```
pfa_plot(a, type="biplot")
```

## Principal Factor Analysis -- Factor Score

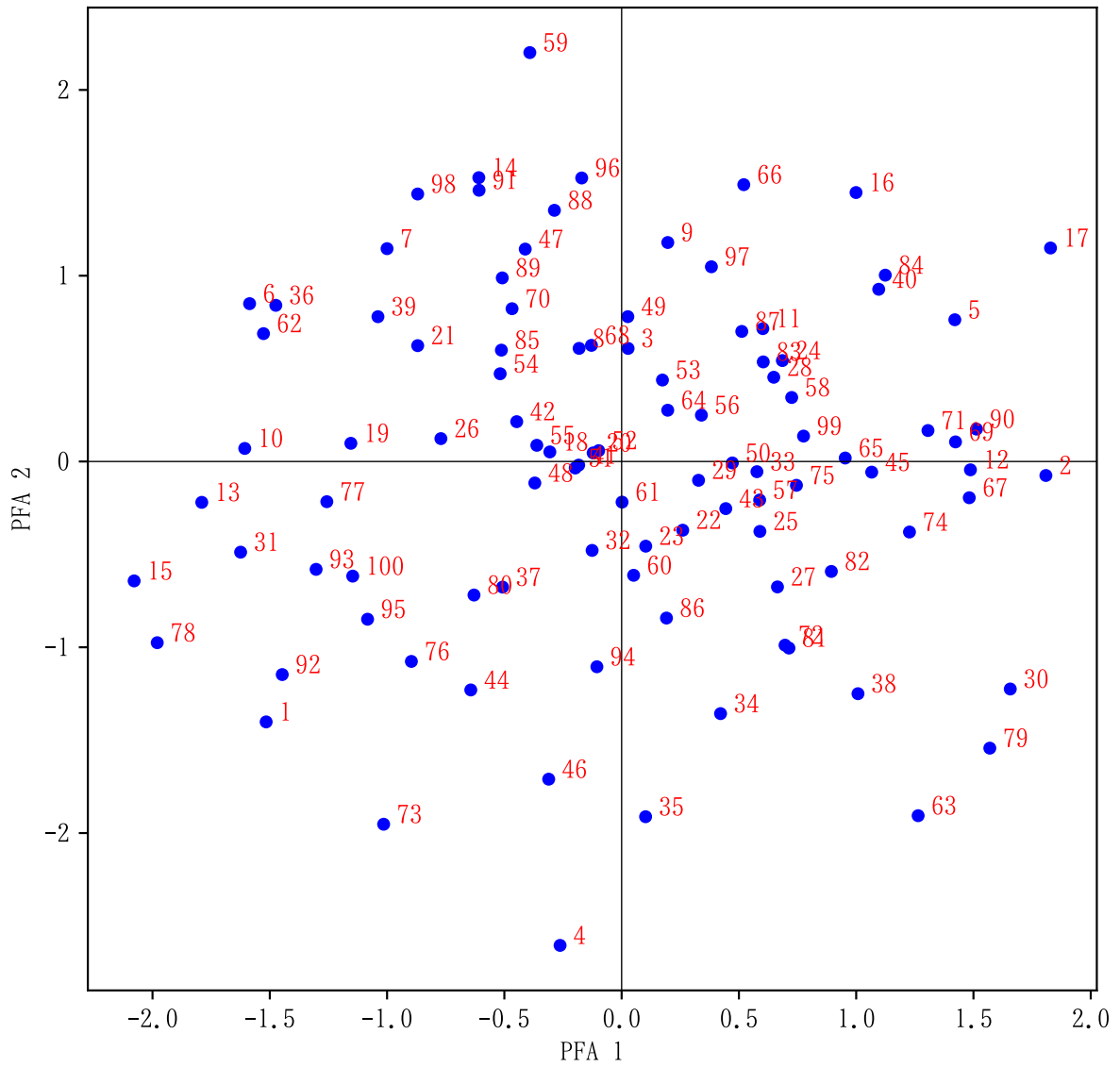


図2 因子得点のプロット

### 3.4 各種回転法の設定

```
x = pfa(dat, rotation="biQuartimax")
```

Rotation: biQuartimax

	Fac.01	Fac.02	Fac.03	Communality
X1	-0.398	-0.322	-0.223	0.312
X2	-0.148	-0.346	0.157	0.166
X3	-0.276	-0.624	-0.032	0.467
X4	0.233	-0.321	-0.718	0.673
X5	-0.280	0.148	-0.680	0.563
X6	-0.903	-0.068	-0.010	0.819
X7	-0.271	0.235	-0.469	0.349

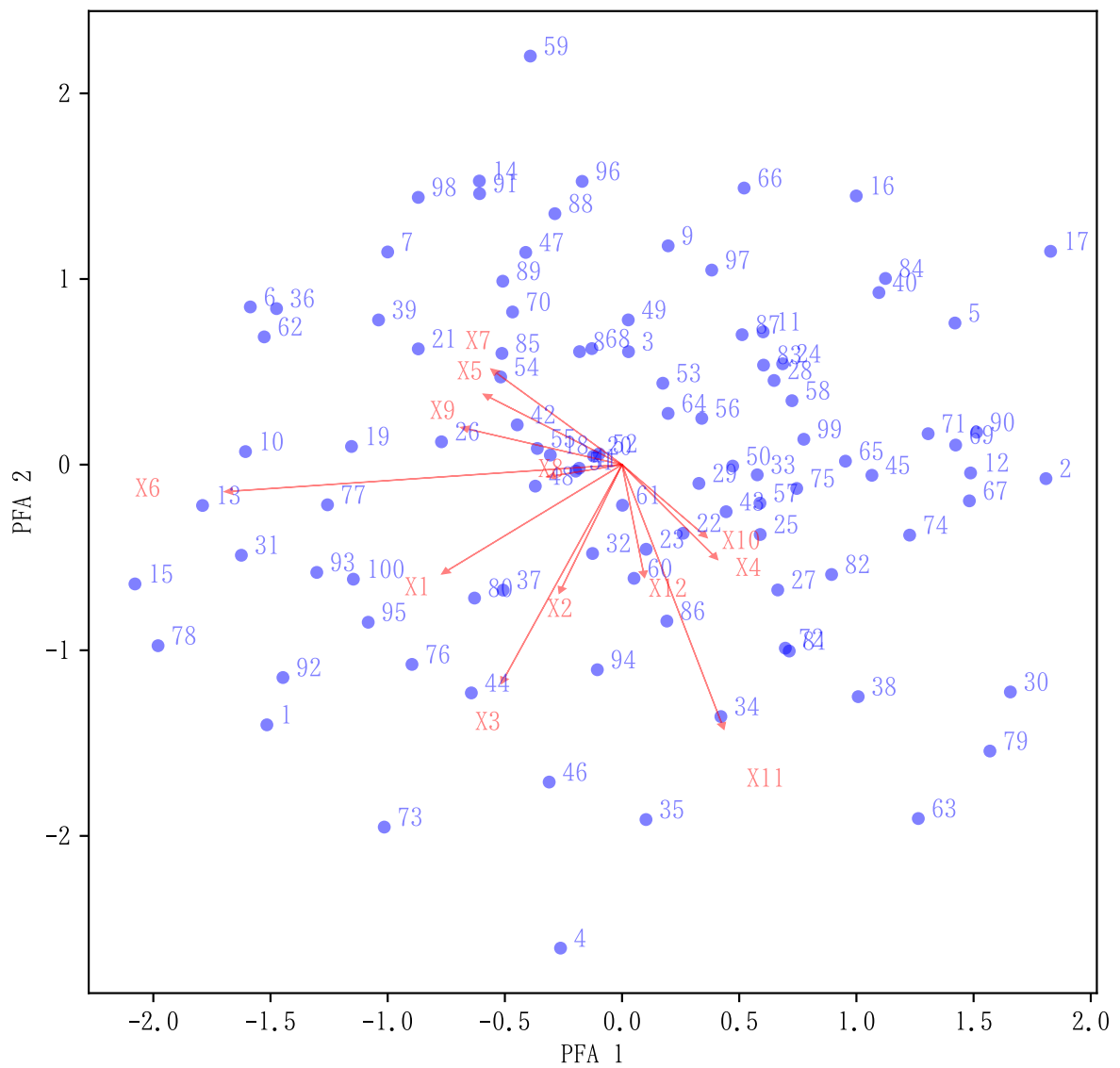


図3 バイプロット

X8	-0.156	-0.030	0.002	0.025
X9	-0.355	0.108	-0.006	0.138
X10	0.194	-0.226	-0.305	0.182
X11	0.221	-0.753	0.093	0.625
X12	0.051	-0.323	-0.132	0.125
Sq.Sum	1.517	1.535	1.393	
Cont.	12.6	12.8	11.6	
Cum.	12.6	25.4	37.0	

```
y = pfa(dat, rotation="Quartimax")
```

Rotation: Quartimax

	Fac.01	Fac.02	Fac.03	Communality
X1	-0.397	-0.328	-0.214	0.312
X2	-0.154	-0.337	0.171	0.166
X3	-0.282	-0.623	-0.007	0.467

X4	0.241	-0.357	-0.698	0.673
X5	-0.268	0.119	-0.691	0.563
X6	-0.903	-0.060	-0.022	0.819
X7	-0.261	0.216	-0.484	0.349
X8	-0.157	-0.029	0.000	0.025
X9	-0.354	0.111	-0.017	0.138
X10	0.196	-0.242	-0.291	0.182
X11	0.212	-0.750	0.131	0.625
X12	0.050	-0.330	-0.116	0.125
Sq.Sum	1.510	1.544	1.390	
Cont.	12.6	12.9	11.6	
Cum.	12.6	25.5	37.0	

```
z = pfa(dat, rotation="Equimax")
```

Rotation: Equimax

	Fac.01	Fac.02	Fac.03	Communality
X1	0.205	-0.263	0.448	0.312
X2	0.381	0.027	0.143	0.166
X3	0.558	-0.233	0.318	0.467
X4	-0.000	-0.815	-0.097	0.673
X5	-0.411	-0.505	0.373	0.563
X6	0.063	0.116	0.896	0.819
X7	-0.405	-0.281	0.326	0.349
X8	0.029	0.016	0.156	0.025
X9	-0.099	0.098	0.344	0.138
X10	0.081	-0.398	-0.129	0.182
X11	0.725	-0.258	-0.184	0.625
X12	0.241	-0.257	-0.009	0.125
Sq.Sum	1.436	1.437	1.571	
Cont.	12.0	12.0	13.1	
Cum.	12.0	23.9	37.0	

### 3.5 相関係数行列から分析を始める

```
import scipy as sp

r = a["r"]
b = pfa(r, factors=3)
```

Rotation: Varimax

	Fac.01	Fac.02	Fac.03	Communality
x1	-0.403	-0.308	-0.233	0.312
x2	-0.139	-0.359	0.135	0.166
x3	-0.272	-0.623	-0.070	0.467
x4	0.210	-0.263	-0.748	0.673
x5	-0.306	0.196	-0.657	0.563

x6	-0.902	-0.077	0.016	0.819
x7	-0.290	0.267	-0.440	0.349
x8	-0.156	-0.032	0.005	0.025
x9	-0.356	0.104	0.014	0.138
x10	0.185	-0.200	-0.328	0.182
x11	0.231	-0.756	0.027	0.625
x12	0.049	-0.312	-0.158	0.125
Sq. Sum	1.531	1.518	1.395	
Cont.	12.8	12.6	11.6	
Cum.	12.8	25.4	37.0	

相関係数からスタートした場合は、因子得点は計算できない。

```
print(b["fs"])
```

None