python-----因子分析

因子分析用Python做的一个典型例子

一、实验目的

采用合适的数据分析方法对下面的题进行解答

例 3.17 现有 48 位应聘者应聘某公司的某职位，公司为这些应聘者的 15 项指标打分，这 15 项指标分别是：求职信的形式 (FL)、外貌 (APP)、专业能力 (AA)、讨人喜欢 (LA)、自信心 (SC)、洞察力 (LC)、诚实 (HON)、推销能力 (SMS)、经验 (EXP)、驾驶水平 (DRV)、事业心 (AMB)、理解能力 (GSP)、潜在能力 (POT)、交际能力 (KJ) 和适应性 (SUIT). 每项分数是从 0 分到 10 分，0 分最低，10 分最高. 每位求职者的 15 项指标列在表 3.6 中. 公司计划录用 6 名最优秀的申请者，问公司将如何挑选这些应聘者？

二、实验要求

采用因子分析方法，根据48位应聘者的15项指标得分，选出6名最优秀的应聘者。

三、代码

```
import pandas as pd
import numpy as np
import math as math
import numpy as np
from numpy import *
from scipy.stats import bartlett
from factor_analyzer import *
import numpy.linalg as nlg
from sklearn.cluster import KMeans
from matplotlib import cm
import matplotlib.pyplot as plt
def main():
    df=pd.read_csv("./data/applicant.csv")
    # print(df)
    df2=df.copy()
    print("\n原始数据:\n",df2)
    del df2['ID']
    # print(df2)
    # 皮尔森相关系数
    df2_corr=df2.corr()
    print("\n相关系数:\n",df2_corr)
    #热力图
    cmap = cm.Blues
    # cmap = cm.hot_r
    fig=plt.figure()
    ax=fig.add_subplot(111)
    map = ax.imshow(df2_corr, interpolation='nearest', cmap=cmap, vmin=0, vmax=1)
    plt.title('correlation coefficient--headmap')

```

```
    ax.set_yticks(range(len(df2_corr.columns)))
    ax.set_yticklabels(df2_corr.columns)
    ax.set_xticks(range(len(df2_corr)))
    ax.set_xticklabels(df2_corr.columns)
    plt.colorbar(map)
    plt.show()
    # KMO测度
    def kmo(dataset_corr):
        corr_inv = np.linalg.inv(dataset_corr)
        nrow_inv_corr, ncol_inv_corr = dataset_corr.shape
        A = np.ones((nrow_inv_corr, ncol_inv_corr))
        for i in range(0, nrow_inv_corr, 1):
            for j in range(i, ncol_inv_corr, 1):
                A[i, j] = -(corr_inv[i, j]) / (math.sqrt(corr_inv[i, i] * corr_in
                A[j, i] = A[i, j]
        dataset_corr = np.asarray(dataset_corr)
        kmo_num = np.sum(np.square(dataset_corr)) - np.sum(np.square(np.diagonal(
        kmo_denom = kmo_num + np.sum(np.square(A)) - np.sum(np.square(np.diagonal
        kmo_value = kmo_num / kmo_denom
        return kmo_value
    print("\nKMO测度:", kmo(df2_corr))
    # 巴特利特球形检验
    df2_corr1 = df2_corr.values
    print("\n巴特利特球形检验:", bartlett(df2_corr1[0], df2_corr1[1], df2_corr1[2],
                                  df2_corr1[5], df2_corr1[6], df2_corr1[7], df2_c
                                  df2_corr1[10], df2_corr1[11], df2_corr1[12], df
    # 求特征值和特征向量
    eig_value, eigvector = nlg.eig(df2_corr)  # 求矩阵R的全部特征值，构成向量
    eig = pd.DataFrame()
    eig['names'] = df2_corr.columns
    eig['eig_value'] = eig_value
    eig.sort_values('eig_value', ascending=False, inplace=True)
    print("\n特征值\n: ",eig)
    eig1=pd.DataFrame(eigvector)
    eig1.columns = df2_corr.columns
    eig1.index = df2_corr.columns
    print("\n特征向量\n",eig1)
    # 求公因子个数m,使用前m个特征值的比重大于85%的标准，选出了公共因子是五个
    for m in range(1, 15):
        if eig['eig_value'][:m].sum() / eig['eig_value'].sum() >= 0.85:
            print("\n公因子个数:", m)
            break
    # 因子载荷阵
    A = np.mat(np.zeros((15, 5)))
    i = 0
    j = 0
    while i < 5:
```

```
        j = 0
        while j < 15:
            A[j:, i] = sqrt(eig_value[i]) * eigvector[j, i]
            j = j + 1
        i = i + 1
    a = pd.DataFrame(A)
    a.columns = ['factor1', 'factor2', 'factor3', 'factor4', 'factor5']
    a.index = df2_corr.columns
    print("\n因子载荷阵\n", a)
    fa = FactorAnalyzer(n_factors=5)
    fa.loadings_ = a
    # print(fa.loadings_)
    print("\n特殊因子方差:\n", fa.get_communalities())  # 特殊因子方差，因子的方差贡
    var = fa.get_factor_variance()  # 给出贡献率
    print("\n解释的总方差（即贡献率）:\n", var)
    # 因子旋转
    rotator = Rotator()
    b = pd.DataFrame(rotator.fit_transform(fa.loadings_))
    b.columns = ['factor1', 'factor2', 'factor3', 'factor4', 'factor5']
    b.index = df2_corr.columns
    print("\n因子旋转:\n", b)
    # 因子得分
    X1 = np.mat(df2_corr)
    X1 = nlg.inv(X1)
    b = np.mat(b)
    factor_score = np.dot(X1, b)
    factor_score = pd.DataFrame(factor_score)
    factor_score.columns = ['factor1', 'factor2', 'factor3', 'factor4', 'factor5'
    factor_score.index = df2_corr.columns
    print("\n因子得分：\n", factor_score)
    fa_t_score = np.dot(np.mat(df2), np.mat(factor_score))
    print("\n应试者的五个因子得分：\n",pd.DataFrame(fa_t_score))
    # 综合得分
    wei = [[0.50092], [0.137087], [0.097055], [0.079860], [0.049277]]
    fa_t_score = np.dot(fa_t_score, wei) / 0.864198
    fa_t_score = pd.DataFrame(fa_t_score)
    fa_t_score.columns = ['综合得分']
    fa_t_score.insert(0, 'ID', range(1, 49))
    print("\n综合得分：\n", fa_t_score)
    print("\n综合得分：\n", fa_t_score.sort_values(by='综合得分', ascending=False).
    plt.figure()
    ax1=plt.subplot(111)
    X=fa_t_score['ID']
    Y=fa_t_score['综合得分']
    plt.bar(X,Y,color="#87CEFA")
    # plt.bar(X, Y, color="red")
    plt.title('result00')
```

```
    axi.set_xticks(range(len(fa_t_score)))
    ax1.set_xticklabels(fa_t_score.index)
    plt.show()
    fa_t_score1=pd.DataFrame()
    fa_t_score1=fa_t_score.sort_values(by='综合得分',ascending=False).head()
    ax2 = plt.subplot(111)
    X1 = fa_t_score1['ID']
    Y1 = fa_t_score1['综合得分']
    plt.bar(X1, Y1, color="#87CEFA")
    # plt.bar(X1, Y1, color='red')
    plt.title('result01')
    plt.show()
if __name__ == '__main__':
    main()
```

四、实验步骤

（1）引入数据，数据标准化

原始数据：

	ID	FL	APP	AA	LA	SC	LC	HON	SMS	EXP	DRV	AMB	GSP	POT	KJ	SUIT
0	1	6	7	2	5	8	7	8	8	3	8	9	7	5	7	10
1	2	9	10	5	8	10	9	9	10	5	9	9	8	8	8	10
2	3	7	8	3	6	9	8	9	7	4	9	9	8	6	8	10
3	4	5	6	8	5	6	5	9	2	8	4	5	8	7	6	5
4	5	6	8	8	8	4	4	9	5	8	5	5	8	8	7	7
5	6	7	7	7	6	8	7	10	5	9	6	5	8	6	6	6
6	7	9	9	8	8	8	8	8	8	10	8	10	8	9	8	10
7	8	9	9	9	8	9	9	8	8	10	9	10	9	9	9	10
8	9	9	9	7	8	8	8	8	5	9	8	9	8	8	8	10
9	10	4	7	10	2	10	10	7	10	3	10	10	10	9	3	10
10	11	4	7	10	0	10	8	3	9	5	9	10	8	10	2	5
11	12	4	7	10	4	10	10	7	8	2	8	8	10	10	3	7
12	13	6	9	8	10	5	4	9	4	4	4	5	4	7	6	8
13	14	8	9	8	9	6	3	8	2	5	2	6	6	7	5	6
14	15	4	8	8	7	5	4	10	2	7	5	3	6	6	4	6
15	16	6	9	6	7	8	9	8	9	8	8	7	6	8	6	10
16	17	8	7	7	7	9	5	8	6	6	7	8	6	6	7	8
17	18	6	8	8	4	8	8	6	4	3	3	6	7	2	6	4
18	19	6	7	8	4	7	8	5	4	4	2	6	8	3	5	4
19	20	4	8	7	8	8	9	10	5	2	6	7	9	8	8	9
20	21	3	8	6	8	8	8	10	5	3	6	7	8	8	5	8
21	22	9	8	7	8	9	10	10	10	3	10	8	10	8	10	8
22	23	7	10	7	9	9	9	10	10	3	9	9	10	9	10	8
23	24	9	8	7	10	8	10	10	10	2	9	7	9	9	10	8
24	25	6	9	7	7	4	5	9	3	2	4	4	4	4	5	4
25	26	7	8	7	8	5	4	8	2	3	4	5	6	5	5	6
26	27	2	10	7	9	8	9	10	5	3	5	6	7	6	4	5

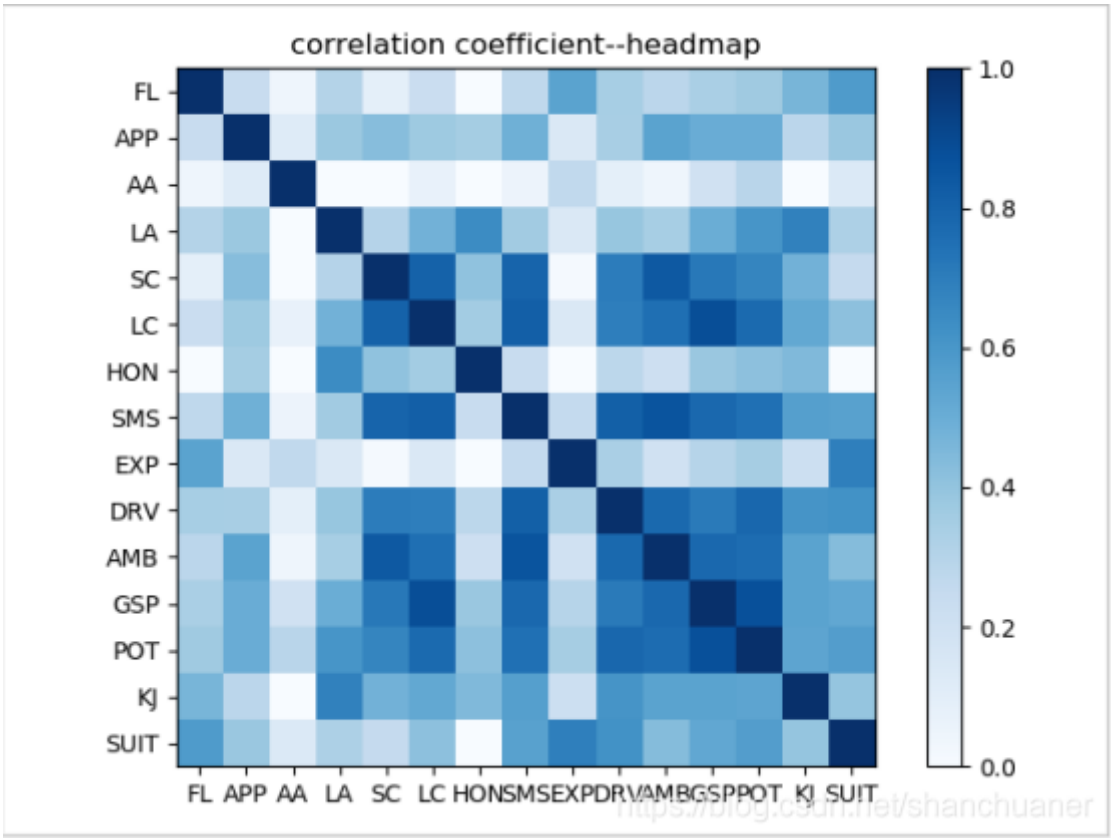
因为数据是面试中的得分，量纲相同，并且数据的分布无异常值，所以数据可以不进行标准化。

（2）建立相关系数矩阵

计算皮尔森相关系数，从热图中可以明显看出变量间存在的相关性。

相关系数：

	FL	APP	AA	...	POT	KJ	SUIT
FL	1.000000	0.238806	0.044041	...	0.367453	0.467206	0.585918
APP	0.238806	1.000000	0.123419	...	0.507377	0.284093	0.384208
AA	0.044041	0.123419	1.000000	...	0.290032	-0.323319	0.140017
LA	0.306313	0.379614	0.001590	...	0.605508	0.685156	0.326957
SC	0.092145	0.430769	0.001107	...	0.671821	0.482456	0.250283
LC	0.228432	0.371259	0.076824	...	0.777316	0.526836	0.416145
HON	-0.106749	0.353691	-0.030270	...	0.415657	0.448246	0.002756
SMS	0.270699	0.489549	0.054727	...	0.753610	0.563284	0.558036
EXP	0.548380	0.140925	0.265585	...	0.348339	0.214953	0.692636
DRV	0.345576	0.340549	0.093522	...	0.788400	0.612808	0.622554
AMB	0.284645	0.549636	0.044066	...	0.768870	0.547126	0.434768
GSP	0.338202	0.506299	0.197505	...	0.875831	0.549408	0.527816
POT	0.367453	0.507377	0.290032	...	1.000000	0.539397	0.573873
KJ	0.467206	0.284093	-0.323319	...	0.539397	1.000000	0.395799
SUIT	0.585918	0.384208	0.140017	...	0.573873	0.395799	1.000000



进行相关系数矩阵检验——KMO测度和巴特利特球体检验：

KMO值：0.9以上非常好；0.8以上好；0.7一般；0.6差；0.5很差；0.5以下不能接受；巴特利球形检验的值范围在0-1，越接近1，使用因子分析效果越好。

```
KMO测度: 0.783775605643526

巴特利特球形检验: BartlettResult(statistic=5.96957033201623, pvalue=0.9672526107058504)
```

通过观察上面的计算结果，可以知道，KMO值为0.783775605643526，在较好的范围内，并且巴特利球形检验的值接近1，所有可以使用因子分析。

（3）求解特征值及相应特征向量

特征值

:	names	eig_value
0	FL	7.513794
1	APP	2.056301
2	AA	1.455819
3	LA	1.197898
4	SC	0.739153
5	LC	0.494579
6	HON	0.351262
7	SMS	0.309902
8	EXP	0.256962
9	DRV	0.184910
10	AMB	0.152680
13	KJ	0.097563
14	SUIT	0.088819
12	POT	0.064633
11	GSP	0.035725

特征向量

	FL	APP	AA	...	POT	KJ	SUIT
FL	0.162440	0.428846	0.315375	...	0.091327	0.184823	0.031532
APP	0.213108	-0.035266	-0.022878	...	-0.087154	0.004062	-0.314357
AA	0.040184	0.236919	-0.430470	...	-0.010788	-0.174262	0.037602
LA	0.225078	-0.129796	0.465825	...	0.149087	-0.024951	-0.059241
SC	0.290481	-0.248896	-0.241026	...	-0.379547	-0.338609	0.001044
LC	0.314870	-0.130990	-0.150037	...	0.113247	-0.002121	-0.420695
HON	0.158117	-0.405450	0.283928	...	0.217349	0.145379	0.201511
SMS	0.324256	-0.029492	-0.185975	...	-0.031535	0.632561	0.299488
EXP	0.134068	0.553139	0.082591	...	0.046044	0.125766	-0.138344
DRV	0.315071	0.046243	-0.079635	...	0.224606	0.061844	-0.558286
AMB	0.318024	-0.068155	-0.208651	...	0.523673	-0.181224	0.377577
GSP	0.331497	-0.023150	-0.117142	...	0.161206	-0.162698	0.109070
POT	0.333289	0.022257	-0.072544	...	-0.557355	0.285995	0.131305
KJ	0.259208	-0.082272	0.467206	...	-0.300908	-0.285675	0.009124
SUIT	0.236037	0.420662	0.089152	...	-0.051240	-0.395659	0.292181

[15 rows x 15 columns]

公因子个数: 5

求公因子个数m,使用前m个特征值的比重大于85%的标准，选出了公共因子是五个。

（4）因子载荷阵

因子载荷阵

	factor1	factor2	factor3	factor4	factor5
FL	0.445270	0.614957	0.380523	0.103262	0.098166
APP	0.584156	-0.050571	-0.027604	-0.286947	0.748148
AA	0.110149	0.339737	-0.519393	-0.696393	-0.182963
LA	0.616969	-0.186125	0.562052	-0.378007	-0.110721
SC	0.796246	-0.356912	-0.290816	0.189131	0.004227
LC	0.863101	-0.187838	-0.181031	0.077745	-0.177803
HON	0.433418	-0.581408	0.342580	-0.455843	-0.054716
SMS	0.888828	-0.042291	-0.224392	0.216957	0.032148
EXP	0.367498	0.793190	0.099652	-0.074154	-0.088631
DRV	0.863652	0.066312	-0.096086	0.170726	-0.172758
AMB	0.871745	-0.097733	-0.251753	0.218121	0.140215
GSP	0.908676	-0.033197	-0.141341	-0.081786	-0.070855
POT	0.913588	0.031917	-0.087529	-0.205916	-0.109465
KJ	0.710524	-0.117976	0.563718	0.220403	-0.095880
SUIT	0.647007	0.603221	0.107568	0.021795	0.069571

特殊因子方差：

```
 FL      0.741535
APP      0.986622
AA       0.915762
LA       0.886344
SC       0.881757
LC       0.850656
HON      0.854033
SMS      0.890260
EXP      0.787490
DRV      0.818517
AMB      0.900108
GSP      0.858481
POT      0.897708
KJ       0.894311
SUIT     0.799380
dtype: float64
```

解释的总方差（即贡献率）：

```
 (factor1    7.513794
factor2    2.056301
factor3    1.455819
factor4    1.197898
factor5    0.739153
dtype: float64, factor1    0.500920
factor2    0.137087
factor3    0.097055
factor4    0.079860
factor5    0.049277
dtype: float64, factor1    0.500920
factor2    0.638006
factor3    0.735061
factor4    0.814921
factor5    0.864198
dtype: float64)
```

由上可以看出，选择5个公共因子，从方差贡献率可以看出，其中第一个公因子解释了总体方差的50.092%，四个公共因子的方差贡献率为86.42%，可以较好的解释总体方差。

（5）因子旋转

因子旋转：

	factor1	factor2	factor3	factor4	factor5
FL	0.106666	0.830036	0.096798	0.146775	0.101412
APP	0.325097	0.149292	0.215908	-0.057082	0.899317
AA	0.065186	0.120279	-0.013386	-0.946268	0.037998
LA	0.229594	0.240462	0.874979	0.041614	0.092134
SC	0.906854	-0.110057	0.141547	0.068346	0.150176
LC	0.876971	0.092289	0.267123	-0.040995	0.004957
HON	0.216746	-0.247016	0.848270	-0.021714	0.161259
SMS	0.897394	0.219691	0.077912	0.049814	0.167713
EXP	0.096702	0.849256	-0.046059	-0.230856	-0.038557
DRV	0.816915	0.345677	0.174706	0.011622	-0.031881
AMB	0.891125	0.156650	0.052890	0.074474	0.270408
GSP	0.808014	0.249143	0.312980	-0.156597	0.145061
POT	0.747264	0.319819	0.397570	-0.246644	0.134630
KJ	0.459516	0.361932	0.565427	0.481160	-0.030634
SUIT	0.369898	0.794922	0.050697	-0.071462	0.151583

（6）因子得分

因子得分：

	factor1	factor2	factor3	factor4	factor5
FL	-0.111880	0.374860	0.001757	0.161634	0.084685
APP	-0.128106	-0.006339	-0.052154	0.041396	1.033437
AA	-0.004663	-0.003754	0.073322	-0.737245	-0.071338
LA	-0.140034	0.070067	0.506222	-0.036746	-0.062229
SC	0.248754	-0.173037	-0.106656	0.054565	0.006978
LC	0.216132	-0.078094	0.012096	-0.051161	-0.208574
HON	-0.088501	-0.159693	0.493553	-0.102192	0.039505
SMS	0.210884	-0.019415	-0.151124	0.066056	0.028583
EXP	-0.066061	0.371123	-0.028068	-0.136820	-0.111698
DRV	0.188303	0.052491	-0.040102	0.009422	-0.231596
AMB	0.202443	-0.049356	-0.183693	0.096352	0.168282
GSP	0.137629	-0.007117	0.044198	-0.124110	-0.042579
POT	0.094606	0.028197	0.121148	-0.202259	-0.069206
KJ	0.001067	0.127011	0.231081	0.338604	-0.169310
SUIT	-0.025375	0.311448	-0.061484	-0.001206	0.081820

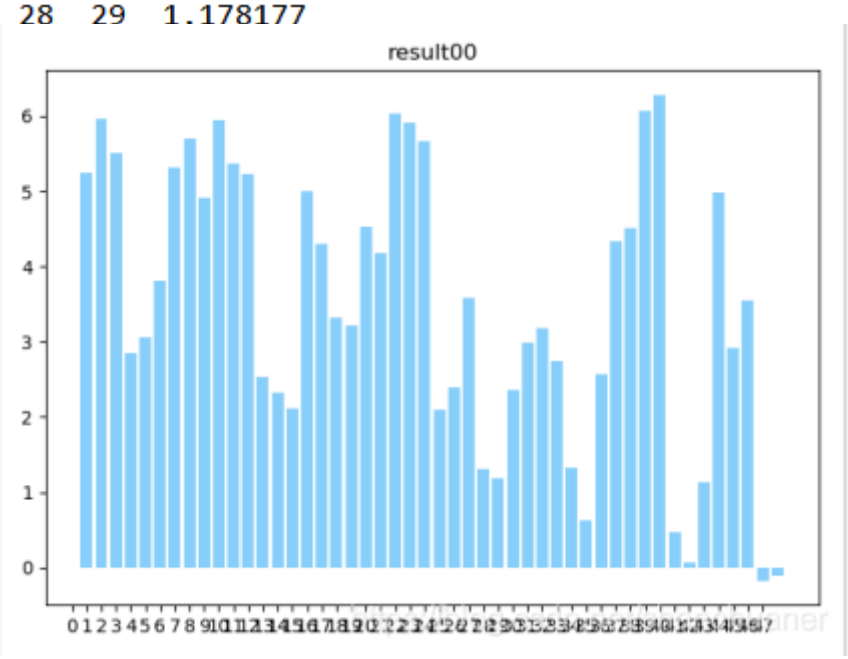
应试者的五个因子得分：

	0	1	2	3	4
0	6.524874	4.367623	4.153680	0.400840	4.744487
1	7.112559	5.970565	6.285794	-1.931819	6.517133
2	6.661235	4.982754	5.561203	-0.450129	4.913979
3	3.012665	4.643731	7.351100	-6.699392	2.743518
4	2.376710	6.412634	8.704217	-6.281653	4.682285
5	4.189914	5.500093	7.295517	-5.448369	3.253634
6	5.958209	8.370296	6.405041	-5.072268	5.139811
7	6.745431	8.287796	6.618982	-5.582193	4.423392
8	5.099233	8.082330	6.875704	-4.290462	5.138021
9	10.153699	2.557192	1.674221	-7.940784	3.574765
10	9.769354	2.279367	-1.092450	-7.991934	3.714129
11	8.907162	1.452614	3.770172	-8.419758	3.316798
12	1.410785	5.040534	9.270845	-5.428492	6.738697
13	1.111130	5.276408	8.306805	-5.636397	7.529463
14	1.486985	4.311309	8.467753	-7.231719	4.960690
15	6.023022	6.223485	5.543384	-4.273181	5.122205
16	4.811251	5.783340	6.047158	-3.602805	4.337620
17	4.376944	2.392961	4.190689	-4.116244	5.549346
18	4.321601	2.944730	3.802245	-4.921228	4.646849
19	5.669807	2.894188	8.668027	-4.772283	4.335706
20	5.388702	2.286936	7.876825	-5.172827	4.887924
21	7.684445	4.887475	8.022307	-3.078910	3.367635
22	7.404572	4.199587	8.386167	-3.420301	5.742154
23	6.787916	4.861702	9.470221	-3.254071	3.384584
24	1.338339	2.798158	7.839971	-4.302506	6.857348
25	1.580265	4.292215	7.813739	-4.570511	6.059691
26	4.673277	0.773849	8.253071	-5.988953	6.696125
27	1.128215	1.168106	4.194156	-1.083751	1.635245
28	1.043987	1.804096	1.602781	-0.008687	1.888138

（7）根据应聘者的五个因子得分，按照贡献率进行加权，得到最终各应试者的综合得分，然后选出前六个得分最高的应聘者。

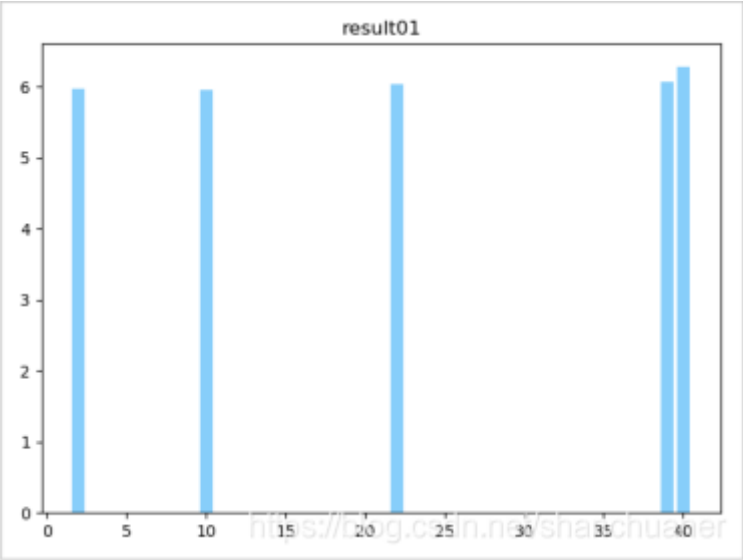
综合得分：

	ID	综合得
0	1	5.248942
1	2	5.968825
2	3	5.514660
3	4	2.845805
4	5	3.058899
5	6	3.802476
6	7	5.325041
7	8	5.704310
8	9	4.906470
9	10	5.949149
10	11	5.374803
11	12	5.227812
12	13	2.541093
13	14	2.322428
14	15	2.111374
15	16	4.998131
16	17	4.299714
17	18	3.323315
18	19	3.209289
19	20	4.525221
20	21	4.171571
21	22	6.037936
22	23	5.911305
23	24	5.661577
24	25	2.093515
25	26	2.397550
26	27	3.586807
27	28	1.303374
28	29	1.178177



综合得分前6名：

	ID	综合得分
39	40	6.283350
38	39	6.068583
21	22	6.037936
1	2	5.968825
9	10	5.949149
22	23	5.911305



所以我们用因子分析产生的前六名分别是：40，39，22，2，10，23