实验三 多元正态分布

姓名：林鹏　　班级：数科2194　　学号：202141084097

一、简单描述统计量

In [3]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
examp3_4_2 = pd.read_excel('./examp3.4.2.xlsx', header=[0]) # 从带分割符的文本文件中导入数据，第-
```

```
D:\Anaconda3\lib\site-packages\openpyxl\worksheet\header_footer.py:48: UserWarnin
g: Cannot parse header or footer so it will be ignored
  warn("""Cannot parse header or footer so it will be ignored""")
```

In [5]:

```
examp3_4_2.head(2)  # 列出数据框的前 2 行
```

Out[5]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
0	44	89.47	44.609	11.37	62	178	182
1	40	75.07	45.313	10.07	62	185	185

In [6]:

```
examp3_4_2.describe()   # 计算五数概括及均值
```

Out[6]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
count	31.000000	31.000000	31.000000	31.000000	31.000000	31.000000	31.000000
mean	47.677419	77.444516	47.375806	10.586129	53.741935	169.645161	173.774194
std	5.211443	8.328568	5.327231	1.387414	8.294447	10.251986	9.164095
min	38.000000	59.080000	37.388000	8.170000	40.000000	146.000000	155.000000
25%	44.000000	73.200000	44.964500	9.780000	48.000000	163.000000	168.000000
50%	48.000000	77.450000	46.774000	10.470000	52.000000	170.000000	172.000000
75%	51.000000	82.325000	50.131000	11.270000	58.500000	176.000000	180.000000
max	57.000000	91.630000	60.055000	14.030000	76.000000	186.000000	192.000000

In [7]:

```
examp3_4_2.std(axis=0)  # 计算标准差
```

Out[7]:

```
年龄              5.211443
体重              8.328568
肺活量             5.327231
1.5英里跑的时间       1.387414
休息时的脉搏          8.294447
跑步时的脉搏         10.251986
跑步时的最大脉搏        9.164095
dtype: float64
```

二、作相关分析

(一)协方差矩阵和相关性矩阵

In [8]:

```
from scipy import stats
```

In [9]:

```
Sigma = examp3_4_2.cov()     # 计算协方差矩阵
Sigma.round(4)  # 保留 4 位小数
```

Out[9]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
年龄	27.1591	-10.1365	-8.4563	1.3647	-6.1194	-18.0516	-20.6753
体重	-10.1365	69.3650	-7.2211	1.6583	1.5682	15.4987	19.0337
肺活量	-8.4563	-7.2211	28.3794	-6.3725	-15.3064	-21.7352	-11.5575
1.5英里跑的时间	1.3647	1.6583	-6.3725	1.9249	4.6093	4.4612	2.8748
休息时的脉搏	-6.1194	1.5682	-15.3064	4.6093	68.7978	27.0387	19.5731
跑步时的脉搏	-18.0516	15.4987	-21.7352	4.4612	27.0387	105.1032	87.3505
跑步时的最大脉搏	-20.6753	19.0337	-11.5575	2.8748	19.5731	87.3505	83.9806

In [10]:

```
rho = examp3_4_2.corr() # 计算相关矩阵
rho.round(4)
```

Out[10]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
年龄	1.0000	-0.2335	-0.3046	0.1887	-0.1416	-0.3379	-0.4329
体重	-0.2335	1.0000	-0.1628	0.1435	0.0227	0.1815	0.2494
肺活量	-0.3046	-0.1628	1.0000	-0.8622	-0.3464	-0.3980	-0.2367
1.5英里跑的时间	0.1887	0.1435	-0.8622	1.0000	0.4005	0.3136	0.2261
休息时的脉搏	-0.1416	0.0227	-0.3464	0.4005	1.0000	0.3180	0.2575
跑步时的脉搏	-0.3379	0.1815	-0.3980	0.3136	0.3180	1.0000	0.9298
跑步时的最大脉搏	-0.4329	0.2494	-0.2367	0.2261	0.2575	0.9298	1.0000

三、散点图矩阵和旋转图

In [11]:

```
from pylab import mpl
import seaborn as sns
```

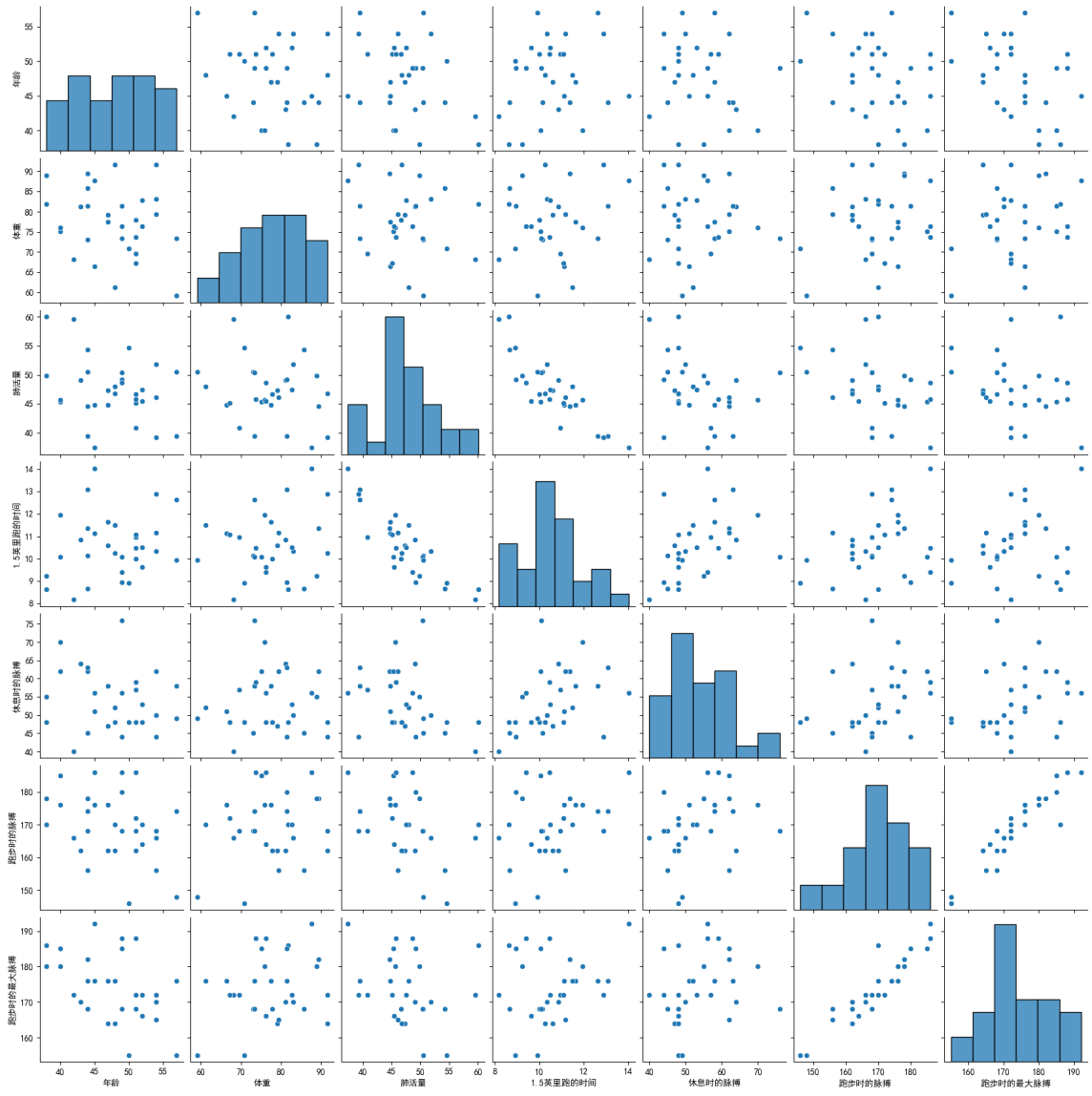
(一)散点图矩阵

In [12]:

```
mpl.rcParams["font.sans-serif"] = ["SimHei"]
sns.pairplot(examp3_4_2)
```

Out[12]:

<seaborn.axisgrid.PairGrid at 0x1badeef5a60>



（二）旋转图

In [13]:

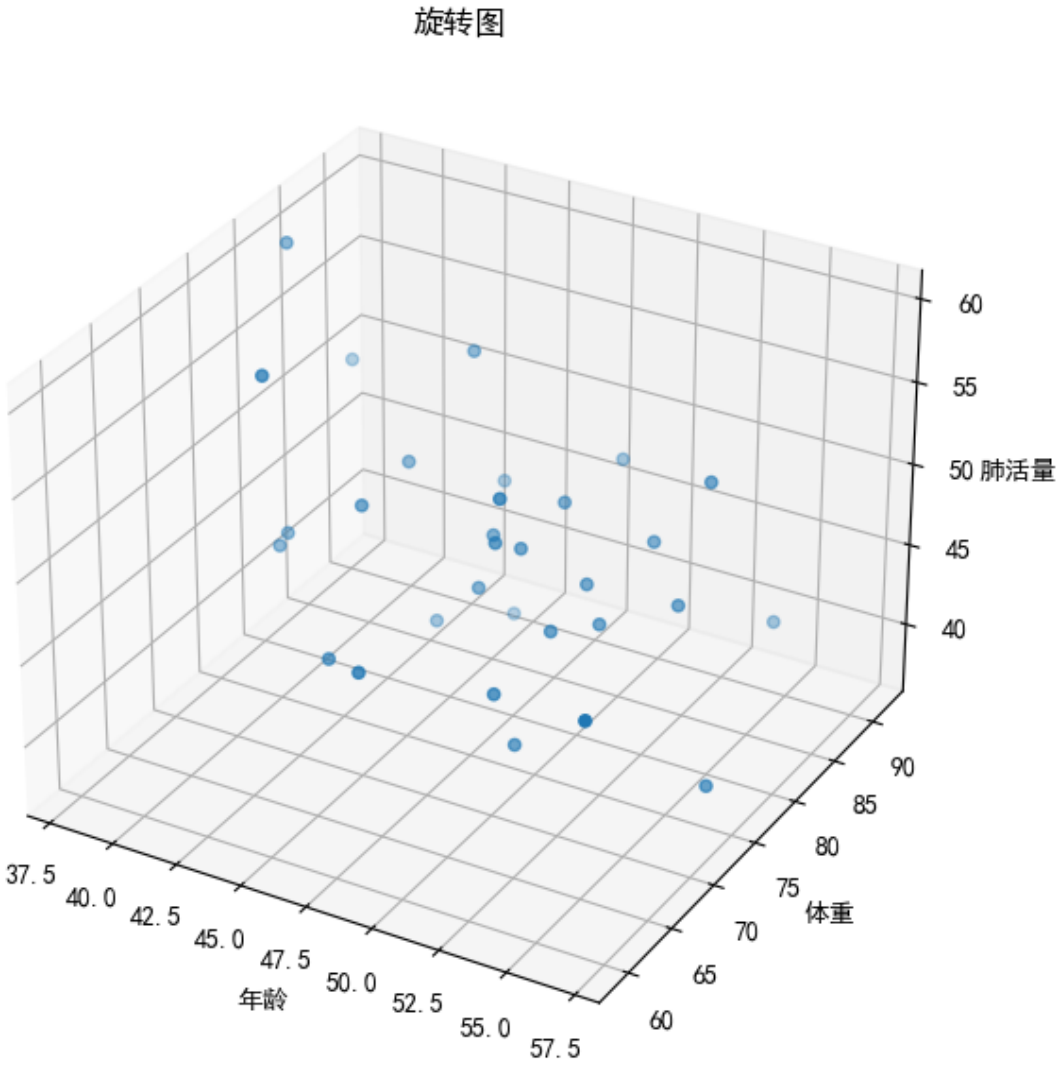
```
import matplotlib.pyplot as plt
examp3_4_2.head()
```

Out[13]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
0	44	89.47	44.609	11.37	62	178	182
1	40	75.07	45.313	10.07	62	185	185
2	44	85.84	54.297	8.65	45	156	168
3	42	68.15	59.571	8.17	40	166	172
4	38	89.02	49.874	9.22	55	178	180

In [14]:

```
#创建一个长为10，宽为7的图
fig = plt.figure(figsize = (10, 7))
#创建一个3D立体图
ax = plt.axes(projection ="3d")
#输入数据
ax.scatter3D(examp3_4_2['年龄'], examp3_4_2['体重'], examp3_4_2['肺活量'])
#填写x,y,z轴轴标及命名标题
ax.set_xlabel('年龄')
ax.set_ylabel('体重')
ax.set_zlabel('肺活量')
plt.title("旋转图")
plt.show()
```



四、正态随机数的产生

In [15]:

```
np.random.normal(size=30).round(3)  # 产生30个标准正态分布的随机数，保留3位小数
```

Out[15]:

```
array([-1.237,  0.152, -0.485, -0.354, -0.818,  0.334, -1.084,  0.819,
       -0.317,  0.926, -1.5  ,  0.105, -1.606, -1.37 ,  0.565, -2.28 ,
        0.272,  0.388, -1.927, -1.43 ,  1.746, -1.546,  0.413, -0.089,
        0.038, -1.073, -0.394, -0.145,  0.384, -3.111])
```

In [16]:

```
np.random.normal(loc=20, scale=6**2, size=10).round(3)  # 产生10个均值为20，标准差为6的正态分布随
```

Out[16]:

```
array([ 34.875, -28.654, -54.795,  59.973,  36.14 ,  34.288, -13.499,
         2.719,  19.991, -24.636])
```

In [17]:

```
mean = np.array([3, 1, 4])  # 指定均值矩阵
```

In [18]:

```
sigma = np.array([6, 1, -2, 1, 13, 4, -2, 4, 4]).reshape(3,3)   # 指定协方差矩阵
```

In [19]:

```
np.random.multivariate_normal(mean, sigma, 5)   # 生成5个三元正态分布的随机数
```

Out[19]:

```
array([[-0.54385449, -1.11070449,  5.21628571],
       [ 2.7455427 , -0.88618621,  1.72051651],
       [ 3.4276779 ,  4.23857431,  5.93724788],
       [ 1.75770395, -3.09427497,  0.43929006],
       [ 5.10544079,  6.69300432,  4.33424932]])
```

五、补充：

1、说明“五数概况”指的是哪五种指标，及这些指标的含义，当原数据中包含缺失值时，这五数概况是如何处理的

> 五数概括指的是最小值，第一分位数，中位数，均值，第三分位数，最大值

> 含义分别为：样本中最小的数，样本中第一个1/4的数，中间数，平均值，样本中第三个1/4的数，样本中最大的数

> 当原数据中包含缺失值时，这五数概括直接忽略缺失值

2、由协方差矩阵查看7个随机变量中方差最大的是哪一个，协方差最大的是哪一对；在相关系数矩阵中它们的相关系数还是最大的吗?

In [20]:

```
value = np.diag(Sigma)
col = Sigma.columns
sigma = pd.Series(data=value,index=col)
sigma.idxmax()
```

Out[20]:

'跑步时的脉搏'

> 方差最大的是"跑步时的脉搏"

In [21]:

```
max_sigma = Sigma.copy()
np.fill_diagonal(max_sigma.values, 0)
max_sigma.stack().idxmax()
```

Out[21]:

('跑步时的脉搏', '跑步时的最大脉搏')

> 协方差矩阵最大的一组是(跑步时的脉搏，跑步时的最大脉搏)

In [22]:

```
max_rho = rho.copy()
np.fill_diagonal(max_rho.values, 0)
max_rho.stack().idxmax()
```

Out[22]:

('跑步时的脉搏', '跑步时的最大脉搏')

> 在相关系数矩阵中它们的相关系数还是最大

3、实现单独求x1和x2的协方差和相关系数

In [23]:

```
df12 = examp3_4_2.iloc[:,[0, 1]]
df12.cov().iloc[0,1].round(4)
```

Out[23]:

-10.1365

x1和x2的协方差为-10.1365

In [24]:

```
df12.corr().iloc[0,1].round(4)
```

Out[24]:

-0.2335

x1和x2的相关系数为-0.2335

4、实现将生成的5个三元正态分布的随机数计算其均值和协方差矩阵，它们与文中的mean和sigma是否相同，为什么

In [25]:

```
mean = (3, 1, 4)
sigma = np.array([6, 1, -2, 1, 13, 4, -2, 4, 4]).reshape(3,3, order='F')
rd = np.random.multivariate_normal(mean, sigma, (5)).round(4)
rd
```

Out[25]:

```
array([[ 4.3537,  8.6627,  7.2991],
       [ 2.6485, -0.7632,  2.7686],
       [ 1.7742, -4.2385,  0.7667],
       [ 2.7024,  2.9407,  3.7528],
       [ 3.9685, -2.2552,  7.0099]])
```

In [26]:

```
rd_mean = rd.mean().round(4)
rd_mean
```

Out[26]:

2.7594

In [27]:

```
rd_Sigma = np.cov(rd).round(4)
rd_Sigma
```

Out[27]:

```
array([[ 4.8504, -3.1939, -5.9502,  0.5022, -5.0823],
       [-3.1939,  4.0213,  6.345 ,  0.3636,  9.0535],
       [-5.9502,  6.345 , 10.3699,  0.2623, 13.4551],
       [ 0.5022,  0.3636,  0.2623,  0.3033,  1.5393],
       [-5.0823,  9.0535, 13.4551,  1.5393, 22.3044]])
```

> 与文中的mean和Sigma不同，因为文中的mean是Sigma是总体的，这里的是样本的

5、实现examp3.4.2的数据标准化（Z标准化），在标准化前后都求一次协方差矩阵和相关系数矩阵，查看标准化后的数据是否存在协方差矩阵和相关系数矩阵一致的情况。

In [28]:

```
# 标准化前
before_Sigma = examp3_4_2.cov()
before_rho =  examp3_4_2.corr()
before_rho
```

Out[28]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
年龄	1.000000	-0.233539	-0.304592	0.188745	-0.141566	-0.337870	-0.432916
体重	-0.233539	1.000000	-0.162753	0.143508	0.022701	0.181516	0.249381
肺活量	-0.304592	-0.162753	1.000000	-0.862195	-0.346405	-0.397974	-0.236740
1.5英里跑的时间	0.188745	0.143508	-0.862195	1.000000	0.400536	0.313648	0.226103
休息时的脉搏	-0.141566	0.022701	-0.346405	0.400536	1.000000	0.317973	0.257503
跑步时的脉搏	-0.337870	0.181516	-0.397974	0.313648	0.317973	1.000000	0.929754
跑步时的最大脉搏	-0.432916	0.249381	-0.236740	0.226103	0.257503	0.929754	1.000000

In [29]:

```
from sklearn import preprocessing
# 标准化
z_score = preprocessing.scale(examp3_4_2.values)
z_score = pd.DataFrame(z_score, columns=examp3_4_2.columns)
```

In [30]:

```
# 标准化后
after_Sigma = z_score.cov()
after_Sigma
```

Out[30]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
年龄	1.033333	-0.241324	-0.314745	0.195037	-0.146285	-0.349133	-0.447346
体重	-0.241324	1.033333	-0.168178	0.148291	0.023458	0.187567	0.257694
肺活量	-0.314745	-0.168178	1.033333	-0.890935	-0.357952	-0.411240	-0.244632
1.5英里跑的时间	0.195037	0.148291	-0.890935	1.033333	0.413887	0.324103	0.233640
休息时的脉搏	-0.146285	0.023458	-0.357952	0.413887	1.033333	0.328572	0.266087
跑步时的脉搏	-0.349133	0.187567	-0.411240	0.324103	0.328572	1.033333	0.960746
跑步时的最大脉搏	-0.447346	0.257694	-0.244632	0.233640	0.266087	0.960746	1.033333

In [31]:

```
after_rho = z_score.corr()
after_rho
```

Out[31]:

	年龄	体重	肺活量	1.5英里跑的时间	休息时的脉搏	跑步时的脉搏	跑步时的最大脉搏
年龄	1.000000	-0.233539	-0.304592	0.188745	-0.141566	-0.337870	-0.432916
体重	-0.233539	1.000000	-0.162753	0.143508	0.022701	0.181516	0.249381
肺活量	-0.304592	-0.162753	1.000000	-0.862195	-0.346405	-0.397974	-0.236740
1.5英里跑的时间	0.188745	0.143508	-0.862195	1.000000	0.400536	0.313648	0.226103
休息时的脉搏	-0.141566	0.022701	-0.346405	0.400536	1.000000	0.317973	0.257503
跑步时的脉搏	-0.337870	0.181516	-0.397974	0.313648	0.317973	1.000000	0.929754
跑步时的最大脉搏	-0.432916	0.249381	-0.236740	0.226103	0.257503	0.929754	1.000000

In [32]:

```
np.allclose(before_Sigma, after_Sigma)
```

Out[32]:

False

协方差矩阵不同

In [33]:

```
np.allclose(before_rho, after_rho)
```

Out[33]:

True

相关矩阵相同