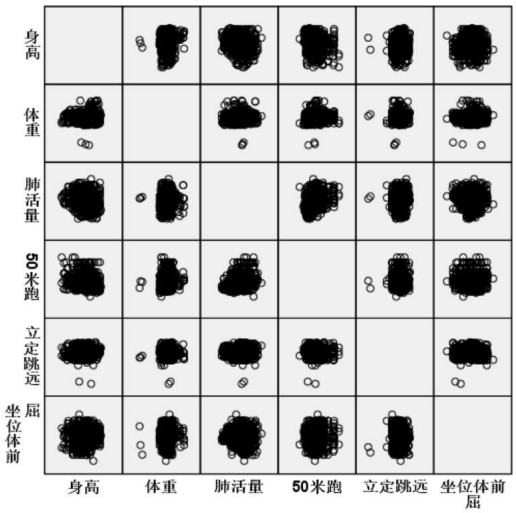
皮尔逊相关系数和斯皮尔曼相关系数(matlab版)

作用：用来衡量两个变量之间的线性相关程度。

前提是你自己判断出他们大概是线性相关的，方法是画图。

这里使用Spss比较方便: 图形 - 旧对话框 - 散点图/点图 - 矩阵散点图



注意：这个数据看起来特别奇怪，是因为数据是我随机生成的。。。
实际建模中遇到的数据应该不会这么奇怪~~~

下一步是对数据经行正态分布检验，用样本大于50，用JB检验，3到50之间用夏洛克-威尔逊检验。

雅克-贝拉检验(Jarque-Bera test)

对于一个随机变量$\{X_i\}$，假设其偏度为S，峰度为K，那么我们可以构造JB统计量：

$$JB = \frac{n}{6}[S^2 + \frac{(K-3)^2}{4}]$$

可以证明，如果$\{X_i\}$是正态分布，那么在大样本情况下$JB \sim \chi^2(2)$（自由度为2的卡方分布）

注：正态分布的偏度为0，峰度为3

那么进行假设检验的步骤如下：

H_0：该随机变量服从正态分布　H_1：该随机变量不服从正态分布

然后计算该变量的偏度和峰度，得到检验值JB^*，并计算出其对应的p值

将p值与0.05比较，如果小于0.05则可拒绝原假设，否则我们不能拒绝原假设。

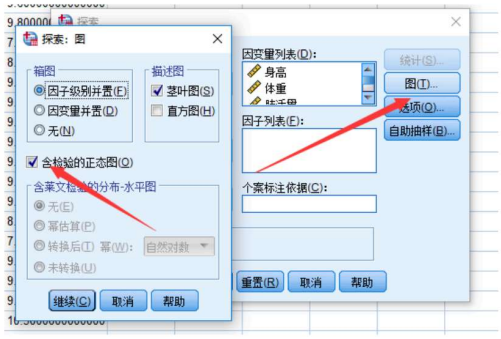
小样本3≤n≤50：Shapiro-wilk检验

Shapiro-wilk夏皮洛-威尔克检验

H_0：该随机变量服从正态分布　H_1：该随机变量不服从正态分布

计算出威尔克统计量后，得到相应的p值

将p值与0.05比较，如果小于0.05则可拒绝原假设，否则我们不能拒绝原假设。



In [ ]:
```
%% 正态分布检验
% 检验第一列数据是否为正态分布
[h,p] = jbtest(Test(:,1),0.05)
% 用循环检验所有列的数据
n_c = size(Test,2); % number of column 数据的列数
H = zeros(1,6);
P = zeros(1,6);
for i = 1:n_c
[h,p] = jbtest(Test(:,i),0.05);
H(i)=h;
P(i)=p;
end
disp(H)
disp(P)
```

1.检验成功，满足正太分布，则可以进行后续分析。计算皮尔逊相关系数：

```
%% 计算各列之间的相关系数
% 在计算皮尔逊相关系数之前,一定要做出散点图来看两组变量之间是否有线性关系
% 这里使用Spss比较方便: 图形 - 旧对话框 - 散点图/点图 - 矩阵散点图

R = corrcoef(Test)    % correlation coefficient


%% 假设检验部分
x = -4:0.1:4;
y = tpdf(x,28);  %求t分布的概率密度值 28是自由度
figure(1)
plot(x,y,'-')
grid on  % 在画出的图上加上网格线
hold on  % 保留原来的图，以便继续在上面操作
% matlab可以求出临界值，函数如下
tinv(0.975,28)     %     2.0484
% 这个函数是累积密度函数cdf的反函数
plot([-2.048,-2.048],[0,tpdf(-2.048,28)],'r-')
plot([2.048,2.048],[0,tpdf(2.048,28)],'r-')
```

```
%% 计算p值
x = -4:0.1:4;
y = tpdf(x,28);
figure(2)
plot(x,y,'-')
grid on
hold on
% 画线段的方法
plot([-3.055,-3.055],[0,tpdf(-3.055,28)],'r-')
plot([3.055,3.055],[0,tpdf(3.055,28)],'r-')
disp('该检验值对应的p值为：')
disp((1-tcdf(3.055,28))*2)  %双侧检验的p值要乘以2

%% 计算各列之间的相关系数以及p值
[R,P] = corrcoef(Test)
% 在EXCEL表格中给数据右上角标上显著性符号吧
P < 0.01  % 标记3颗星的位置
(P < 0.05) .* (P > 0.01)  % 标记2颗星的位置
(P < 0.1) .* (P > 0.05) % % 标记1颗星的位置
% 也可以使用Spss操作哦
```

2.检验失败，计算斯皮尔曼相关系数：

```
X = [3 8 4 7 2]' % 一定要是列向量哦，一撇'表示求转置
Y = [5 10 9 10 6]'
coeff = corr(X , Y , 'type' , 'Spearman')
disp(coeff)
```

斯皮尔曼相关系数可以用来计算定序数据之间的相关性。

In [ ]:

```python
#pearson相关系数，spearman相关系数
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from scipy.stats import shapiro,jarque_bera , pearsonr, spearmanr

# 设置中文字体
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 读取数据
try:
    mydata = pd.read_excel('data.xls')  # 读取数据
except FileNotFoundError:
    print("文件未找到，请检查文件路径和名称是否正确。")
    exit()

# 制作散点图矩阵
sns.pairplot(mydata)
plt.show()

# 定义样本大小阈值
sample_size_threshold = 50

# 判断每一列数据是否符合正态分布
normality_results = {}
```

```
for column in mydata.columns:
    sample_size = len(mydata[column].dropna())
    if sample_size < sample_size_threshold:
        # 小样本使用 Shapiro-Wilk 检验
        stat, p = shapiro(mydata[column].dropna())
        test_name = 'Shapiro-Wilk'
    else:
        # 大样本使用 Jarque-Bera 检验
        stat, p = jarque_bera(mydata[column].dropna())
        test_name = 'Jarque-Bera'

    print(f'{column} ({test_name}): Statistics={stat:.3f}, p={p:.3f}')
    normality_results[column] = p > 0.05

# 检查是否有不符合正态分布的数据
use_pearson = all(normality_results.values())

# 计算相关系数矩阵和 p 值矩阵
correlation_matrix = mydata.corr(method='pearson' if use_pearson else 'spearman'
if use_pearson:
    p_value_matrix = mydata.corr(method=lambda x, y: pearsonr(x, y)[1])
else:
    p_value_matrix = mydata.corr(method=lambda x, y: spearmanr(x, y)[1])


# 创建掩码，只显示下三角，包含对角线
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool), k=1)

# 绘制热力图
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, mask=mask, cmap='coolwarm', fmt=".2f
            cbar_kws={"shrink": .8, "label": "相关系数"})
plt.title('相关系数热力图')
plt.show()

# 相关系数矩阵的可视化
cluster_map = sns.clustermap(correlation_matrix, cmap='coolwarm', figsize=(10, 8
                             cbar_kws={"shrink": .8, "label": "相关系数"})
cluster_map.fig.suptitle('相关系数聚类图', fontsize=16)
plt.show()

# 打印 p 值矩阵
print("P 值矩阵:")
print(p_value_matrix)
# 相关显著结果
significant_results = p_value_matrix < 0.05
print("相关显著结果:")
print(significant_results)
```