北京建筑大学

2019 / 2020 学年　第 2 学期

课　程　设　计

课程名称 数据分析实践

设计题目 数据分析python实现

（工程名称）

系　　别 信息与计算科学

班　　级 信171

学生姓名 宋卡妮 李金哲

学　　号 201707010124
201707010119

完成日期 2020/6/28

成绩	
指导教师 （签名）	

目录

《数据分析实践》课程设计任务书

指导教师姓名	王恒友	**教研室**	信计		
课程设计题目	数据分析方法的 Python 实现	**人数**	2	**学时**	5 天

设计目的、任务和要求

（一）目的

要求学生利用 Python 实现所学习的所有数据分析方法，并结合数据案例进行分析，并对对某医院检查诊断的数据（数据取自北京市某医院）进行分析。

（二）任务

本数据共有两大项：分别为 2007、2010 年。每项检查项目包括：性别、年龄、腰围(cm)、TC(mmol/L)、TG(mmol/L)、HbA1C(%)、HDL-C(mmol/L)、LDL-C(mmol/L)。

首先对所给数据作基本描述性统计分析，并检查各组数据的分布服从哪种分布。

其次对所给数据做相关性分析，检验各组数据之间是否有相关性。

最后对所给数据做两独立样本的参数检验、非参数检验分析，检验两年的各项数据是否有显著差异。同时根据性别整理数据，检验由于性别不同，各项指标是否有显著差异。认真分析结果，撰写分析报告。

具体数据见附件。

（三）要求

学生 2 个人组成一组，要求根据上述任务，通过查找资料，完成 SPSS 的统计分析，并撰写分析报告。

设计的方法和步骤

选择描述性统计分析过程、方差分析过程、参数检验、非参数检验（使用 SPSS 软件），按照下列要求对数据进行分析说明，具体要求如下：

1、 熟悉本次医院检查诊断数据，对数据分析目标进行深入的认识；

2、 首先运用 Python 程序语言，选择描述性统计分析过程对各组数据做基本描述性分析，并做正态性检验；

3、 运用 Python 程序语言，选择相关性分析过程对各组数据做相关性分析；

4、 运用 Python 程序语言，对各组数据做参数检验、非参数检验；

设计工作计划

本案例时间为 5 天，具体安排如下

- 运用 Python 程序语言，实现所学数据分析方法及案例分析：1 天
- 分析案例，设计相应的数据分析方案：1 天
- 运用 Python 程序语言，采用描述性统计分析等对各组数据做基本描述性分析，并做正态性检验；并完成相应分析报告：1 天
- 运用 Python 程序语言，利用相关性分析过程对各组数据做相关性分析，并完成相应分析报告：1 天
- 整理报告及准备答辩：1 天

主要参考资料

《数据分析》　　范金城、梅长林　　科学出版社

《数据分析方法》梅长林、范金城　　高等教育出版社

《SPSS 统计分析方法及应用》 薛薇　　电子工业出版社

教研室签字：

年　　月　　日

院签字：

年　　月　　日

摘要

报告基于 python 复现了本学习数据分析课程上学习的参数检验、方差分析、相关分析、回归分析、聚类分析、因子分析等方法，并且对某医院检查诊断的数据进行分析，并使用上述方法对数据的分布，相关性进行检验，得出腰围(cm)、TC(mmol/L)、TG(mmol/L)、HbA1C(%)、HDL-C(mmol/L)、LDL-C(mmol/L)六个变量的分布随年份的变化分布趋势改变，偏度及峰度变化大，且大部分变量之间的相关性很弱，个别变量之间相关性很强。同时，age，waist，HDL_C，LDL_C 在 2007 年符合正态分布的特征，age，waist，TC 在 2010 年符合正态分布的特征。

通过查找期刊数据，了解到 TC、TG、Hba1C、HDL-C、LDL-C 变量是包含在糖尿病的检测方式，研究该地区糖尿病人群的变化趋势。通过参数检验确定 2007 年样本和 2010 年的样本具有一致性，进而检验发现两样本的各项指标是超过正常人的数值的范围，大致推断出该地区人群的有高概率患有糖尿病。再经由聚类分析，分别将两年的样本数据聚类，得到与 2007 年样本人群相比，2010 年样本人群可以因为具有了调理自身饮食、注重保养身体等原因，高概率患糖尿病的人数和所占比例均有显著的减少。

关键词： SPSS 功能实现　Python　数据分析

1 参数检验

1.1 单样本 T 检验

1.1.1 原理分析

单总体 t 检验是检验一个样本平均数与一个已知的总体平均数的差异是否显著。当总体分布是正态分布，如总体标准差未知且样本容量小于 30，那么样本平均数与总体平均数的离差统计量呈 t 分布。

单总体 t 检验统计量为：

$$t = \frac{\bar{X} - \mu}{\frac{\sigma X}{\sqrt{n}}} \tag{1}$$

其中 $i = 1 \cdots n, \bar{x} = \frac{\sum_{i=1}^{n} xi}{n}$ 为样本平均数，$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$为样本标准偏差，n 为样本数。该统计量 t 在零假说：μ=μ0 为真的条件下服从自由度为 n 的 t 分布。

1.1.2 模块核心代码解读

模块引用

```
from scipy.stats import ttest_1samp
```

函数引用

```
ttest_1samp(a, popmean, axis=0, nan_policy='propagate'):
```

传递参数

a：样本矩阵

popmean：检验统计量

axis：数据变化的方向（1 表示横轴，方向从左到右；0 表示纵轴，方向从上到下），默认值为 0

nan_policy：处理 Nan 值的方法，默认值为**'propagate'**

核心代码

```
d = np.mean(a, axis) - popmean
v = np.var(a, axis, ddof=1)
denom = np.sqrt(v / n)
with np.errstate(divide='ignore', invalid='ignore'):
    t = np.divide(d, denom)
```

1）d = np.mean(a, axis) - popmean

计算即均值与检验值的差

2）v = np.var(a, axis, ddof=1)

计算 样本标准差方

3）denom = np.sqrt(v / n)

计算 样本标准差与根号 n 的比值

4）t = np.divide(d, denom)

计算 t 统计量

1.1.3 结果分析

图 1.1 是 SPSS 的单样本 t 检验结果，而图 1.2 是 Python 实现的单样本 t 检验，通过观测结果，可以得知：两种方法实现的单样本检验结果相同，其中 t 统计量为 5.37074162，概率 P 值为 1.2043995e-07。故可以大致判定为，Python 和 SPSS 实现的单样本 t 检验算法具有一致性。

而且由于相关性概率 p 值十分小，可以约等于 0；而且远小于接受 H_0 的概率 α=0.05。所以拒接原假设 H_0，接受假设 H_1。即信用卡消费不低于 3000。

T-检验

单样本统计

	个案数	平均值	标准差	标准误差平均值
月平均刷卡金额	500	4781.8786	7418.71786	331.77515

单样本检验

	检验值 = 3000					
	t	自由度	显著性（双尾）	平均值差值	差值 95% 置信区间 下限	上限
月平均刷卡金额	5.371	499	1.2044E-7	1781.87860	1130.0302	2433.7270

图 1.1 SPSS 单样本 t 检验

```
from scipy import stats

# 第二个变量如果是变量，则为配对样本t检验；如果为数字则为单样本t检验
t, p=stats.ttest_1samp(data1,3000)
print("t=",t,"p=",p)
# 由于判断是否低于3000，故p值应该除2
p /= 2
print(data1.mean())

if p < 0.05:
    print('信用卡消费不低于3000')
else:
    print('信用卡消费低于3000')
```

```
t= [5.37074162] p= [1.2043995e-07]
月平均刷卡金额    4781.8786
dtype: float64
信用卡消费不低于3000
```

图 1.2 python 实现单样本 t 检验

1.2 两独立样本 t 检验

1.2.1 原理分析

双总体 t 检验是检验两个样本平均数与其各自所代表的总体的差异是否显著。其中独立样本 t 检验（各实验处理组之间毫无相关存在，即为独立样本），该检验用于检验两组非相关样本被试所获得的数据的差异性。

独立样本 t 检验统计量为：

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}} \quad (2)$$

S12 和 S22 为两样本方差；n1 和 n2 为两样本容量。

1.2.2 模块核心代码解读

模块引用

from scipy.stats **import** ttest_ind

函数引用

def ttest_ind(a, b, axis=0, equal_var=**True**, nan_policy=**'propagate'**)

传递参数

a：样本矩阵 1

b：样本矩阵 2

axis：数据变化的方向（1 表示横轴，方向从左到右；0 表示纵轴，方向从上到下），默认值为 0

equal_var：a，b 样本是否为方差齐性（默认为 True）

nan_policy：处理 Nan 值的方法，默认值为**'propagate'**

核心代码

```
v1 = np.var(a, axis, ddof=1)
v2 = np.var(b, axis, ddof=1)
    n1 = a.shape[axis]
    n2 = b.shape[axis]

        if equal_var:
df, denom = _equal_var_ttest_denom(v1, n1, v2, n2)
        else:
```

df, denom = _unequal_var_ttest_denom(v1, n1, v2, n2)

res = _ttest_ind_from_stats(np.mean(a, axis), np.mean(b, axis), denom, df

代码解读

1) v1 = np.var(a, axis, ddof=1)

按照 列或行 求平均样本矩阵 a 的平均值

2）v2 = np.var(b, axis, ddof=1)

按照 列或行 求平均样本矩阵 b 的平均值

3）

if equal_var:

df, denom = _equal_var_ttest_denom(v1, n1, v2, n2)

else:

df, denom = _unequal_var_ttest_denom(v1, n1, v2, n2)

通过判断两样本是否为方差齐性，决定抽样方差估值计算方式

$$\sigma_{12}^2 = \frac{S_p^2}{n_1} + \frac{S_p^2}{n_2} \tag{3}$$

4）res = _ttest_ind_from_stats(np.mean(a, axis), np.mean(b, axis), denom, df)

运用公式计算 t 统计量及其相关数据

$$t = \frac{\overline{x_1} - \overline{x_2} - (\mu_1 - \mu_2)}{\sqrt{\sigma_{12}^2}} \tag{4}$$

1.2.3 结果分析

图 1.3 是 SPSS 的两独立样本 t 检验结果，而图 1.4 是 Python 实现的两独立样本 t 检验，通过观测结果，可以得知：两种方法实现的单样本检验结果相同，其中 t 统计量为-34.758580660626215，概率 P 值为 2.757952775215341e-168。故可以大致判定为，Python 和 SPSS 实现的两独立样本 t 检验结果具有一致性。

而且由于相关性概率 p 值十分小，可以约等于 0；而且远小于接受 H_0 的概率 α=0.05。所以拒接原假设 H_0，接受假设 H_1。即男女对有明确的职业目标有显著差异。

独立样本检验

		莱文方差等同性检验		平均值等同性 t 检验						
		F	显著性	t	自由度	显著性（双尾）	平均值差值	标准误差差值	差值 95% 置信区间 下限	上限
有明确的职业目标	假定等方差	6.747	.010	-34.759	896	2.758E-168	-1.810	.052	-1.912	-1.708
	不假定等方差			-34.169	742.160	.000	-1.810	.053	-1.914	-1.706

图 1.3 SPSS 两独立样本 t 检验

```python
# 检验方差是否齐性，决定两独立样本t检验的分析方法
from  scipy.stats import ttest_ind, levene
t, p = levene(man['Q63'].astype('int'),woman['Q63'].astype('int'))
if p < 0.05:
    print('方差非齐性')
else:
    print('方差齐性')
```

```
方差齐性
```

```python
# 由于方差齐性,equal_var = True
t, p = ttest_ind(man['Q63'].astype('int'),woman['Q63'].astype('int'), equal_var = True)
print("t=",t,"p=",p)
if p < 0.05:
    print('男女对有明确的职业目标有显著差异')
else:
    print('男女对有明确的职业目标无显著差异')
```

```
t= -34.758580660626215 p= 2.757952775215341e-168
男女对有明确的职业目标有显著差异
```

图 1.4 python 实现两独立样本 t 检验

1.3 两独立样本的 K-S 检验

1.3.1 原理分析

K-S 检验不仅能够检验单个总体是否服从某一理论分布，还能够检验两总体分布是否存在显著差异。其原假设 H 是:两独立样本来自的两总体的分布无显著

差异。

两独立样本的 K-S 检验的基本思想与单样本 KS 检验的基本思想大体致，主要差别在于:这里以变量值的秩作为分析对象，而非变量值本身。

首先，将两样本混合并按升序排序;然后，分别计算两样本秩的累计频数和累计频率;最后，计算两组累计频率差的绝对值，得到累计频率绝对差序列并得到 D 统计量(同单样本 KS 检验，但无须修正)

然后计算在大样本下的$\frac{1}{2}D\sqrt{n}$的观测值和概率 P-值。

1.3.2 模块核心代码解读

模块引用

```
from scipy.stats import ks_2samp
```

函数引用

```
def ks_2samp(data1, data2)
```

传递参数

a：样本矩阵 1

b：样本矩阵 2

核心代码

```
d = np.max(np.absolute(cdf1 - cdf2))
# Note: d absolute not signed distance
en = np.sqrt(n1 * n2 / (n1 + n2))
try:
    prob = distributions.kstwobign.sf((en + 0.12 + 0.11 / en) * d)
```

代码解读

1) d = np.max(np.absolute(cdf1 - cdf2))

计算极端差值的绝对值

2）distributions.kstwobign.sf((en + 0.12 + 0.11 / en) * d)

“标准化”数据，通过 **kstwobign** 描述渐近分布（即$\frac{1}{2}D\sqrt{n}$）在 KS 检验的零假设下的随机变量

1.3.3 结果分析

图 1.5 是 SPSS 的两独立样本的 K-S 检验结果，而图 1.6 是 Python 实现的两独立样本的 K-S 检验结果。由于 Python 的内核运算时使用的单侧检验，故观测结果时，应当将 Python 的概率 P 值*2，即为 SPSS 的概率 P 值。因此得知：两种方法实现的单样本检验结果相同，其中 Z 统计量为 0.7321428571428572，概率 P 值为 0.033957514799215306。故可以大致判定为，Python 和 SPSS 实现的两独立样本的 K-S 检验结果具有一致性。

由于相关性概率 p 值约为 0.0169；大于接受 H_0 的概率 α=0.05。所以接受原假设 H_0。即甲、乙两种工艺下的产品使用寿命的分布存在显著差异。

检验统计[a]

		使用寿命
最极端差值	绝对	.732
	正	.732
	负	.000
柯尔莫戈洛夫-斯米诺夫 Z		1.415
渐近显著性（双尾）		.037

a. 分组变量：使用工艺

图 1.5 SPSS 实现两独立样本的 K-S 检验

```
#引用模块
from scipy.stats import ks_2samp

#进行KS test
z, p = ks_2samp(gy_1['sysm'], gy_2['sysm'])
print("z=", z, "p=", p, '\n')

if p < 0.05:
    print('甲、乙两种工艺下的产品使用寿命的分布存在显著差异')
else:
    print('甲、乙两种工艺下的产品使用寿命的分布不存在显著差异')
```

```
z= 0.7321428571428572 p= 0.016978757399607653

甲、乙两种工艺下的产品使用寿命的分布存在显著差异
```

图 1.6 python 实现两独立样本的 K-S 检验

1.4 两独立样本的曼-惠特尼 U 检验

1.4.1 原理分析

曼-惠特尼 U 检验（Mann-Whitney U test），又称曼-惠特尼秩和检验，可以看作是对两均值之差的参数检验方式的 T 检验或相应的大样本正态检验的代用品。由于曼-惠特尼秩和检验明确地考虑了每一个样本中各测定值所排的秩，它比符号检验法使用了更多的信息。单样本 KS 检验，但无须修正)。

检测方法：

1） 两组样本数据混合，并按照数据大小的升序编排等级。最小的数据等级为 1，第二小的数据等级为 2，以此类推（注意，如果混合后的数据中存在相等的情况，那么相同数据的等级值应该是相同的，并取未经排名的数组中的平均值。）

2） 分别求出两个样本的等级和 **R1**,**R2**。

3） 假设 **n1** = “一号样本观察值的项数”；**n2** = “二号样本观察值的项数”；**R1**

= “一号样本各项等级和”；**R2** = “二号样本中各项等级和”。那么 **U1**, **U2** 的计算公式分别如下所示：

$$U1 = R1 - n1 * \frac{n1+1}{2} \tag{5}$$

$$U2 = R2 - n2 * \frac{n2+1}{2} \tag{6}$$

那么 U1 与 U2 之和的计算公式如下所示，

$$U1 + U2 = R1 + R2 - \frac{n1 * (n1+1) + n2 * (n2+1)}{2} \tag{7}$$

设 2 组样本总共数据有 **N** 个，即 N = n1 + n2，又因为 R1 + R2 = N(N + 1)/ 2 ，代入上式，可得

$$U1 + U2 = n1 * n2 \tag{8}$$

选择 **U1** 和 **U2** 中最小者与临界值 **Uα** 比较，当 **U < Uα** 时，拒绝 **H0**，接受 **H1**。

在原假设为真的情况下，随机变量 **U** 的均值和方差分别为：

$$E(U) = n1 * \frac{n2}{2} D(u) = n1 * n2 * \frac{n1+n2+1}{12} \tag{9}$$

当 **n1** 和 **n2** 都不小于 **10** 时，随机变量近似服从正态分布。

4）作出判断。

1.4.2 模块核心代码解读

模块引用

```
from scipy.stats import mannwhitneyu
```

函数引用

```
def mannwhitneyu(x, y, use_continuity=True, alternative=None):
```

传递参数

x：样本矩阵 1

y：样本矩阵 2

use_continuity：对样本进行连续性填补（样本(a+b)/2）,默认 True

alternative：概率 P 值的检验方法，默认为 None

核心代码

```
u1 = n1*n2 + (n1*(n1+1))/2.0 - np.sum(rankx, axis=0)
u2 = n1*n2 - u1
        sd = np.sqrt(T * n1 * n2 * (n1+n2+1) / 12.0)
        meanrank = n1*n2/2.0 + 0.5 * use_continuity
        bigu = max(u1, u2)
        z = (bigu - meanrank) / sd
p = distributions.norm.sf(abs(z))
```

代码解读

1) u1 = n1*n2 + (n1*(n1+1))/2.0 - np.sum(rankx, axis=0)
u2 = n1*n2 - u1

分别计算样本 1 和样本 2 的 u 检验值

2）sd = np.sqrt(T * n1 * n2 * (n1+n2+1) / 12.0)

meanrank = n1*n2/2.0 + 0.5 * use_continuity

在原假设为真的情况下，分别计算随机变量 **U** 的均值和方差

3）p = distributions.norm.sf(abs(z))

通过检验结果的正态分布，获取概率 p 值

1.4.3 结果分析

图 1.7 是 SPSS 的曼-惠特尼 U 检验结果，而图 1.8 是 Python 实现的曼-惠特尼 U 检验结果。观测可得知：两种方法实现的曼-惠特尼 U 检验的 u 统计量结果相同均为4, 概率P值可以因为检验函数的区别存在出入, Python为0.00326,

SPSS 为 0.004。故可以大致判定为，Python 和 SPSS 实现的两独立样本 t 检验结果具有相对一致性。

由于两种方法的相关性概率 p 值均小于接受 H_0 的概率 α=0.05。所以拒绝原假设 H_0，接受假设 H_1。即甲、乙两种工艺下的产品使用寿命的分布不存在显著差异。

检验统计[a]

	使用寿命
曼-惠特尼 U	4.000
威尔科克森 W	40.000
Z	-2.777
渐近显著性（双尾）	.005
精确显著性 [2*(单尾显著性)]	.004[b]

a. 分组变量：使用工艺

b. 未针对绑定值进行修正。

图 1.7 SPSS 实现两独立样本的曼-惠特尼 U 检验

```
[76]: #引用模块
      from scipy.stats import mannwhitneyu

      mannwhitneyu(gy_1['sysm'], gy_2['sysm'])

[76]: MannwhitneyuResult(statistic=4.0, pvalue=0.0032680805871213377)
```

图 1.8 python 实现两独立样本的曼-惠特尼 U 检验

2 方差分析

2.1 单因素方差分析

2.1.1 原理分析

单因素方差分析用来研究一个控制变量的不同水平是否对观测变量产生了显著影响。方差分析认为： 观测变量值的变动会受控制变量和随机变量两方面的影响。据此，单因素方差分析将观测变量总离差平方和分解为组间(Between Gro ups) 离差平方和与组内离差平方和两部分，用数学形式表述为：

$$SST = SSA + SSE \tag{10}$$

SST 为观测变量总离差平方和； SSA 为组间离差平方和，是由控制变量的不同水平造成的变差； SSE 为组内离差平方和，是由抽样误差引起的变差。

2.1.2 核心代码

```
w,p = stats.levene(*args)
print ('w='+str(w),'p='+str(p))
if p<0.5:
print('方差齐性假设不成立')
else:
print('方差齐性假设成立')
#进行方差分析
f,p = stats.f_oneway(*args)
```

先进行方差齐性检验，再方差分析。代码以第六章方差分析（广告城市与销售额）的数据为例，分析不同的广告形式是否对销售额产生显著影响。数据按照广告形式分组，每组设为一个变量，args 指所有变量。

2.1.3 结果分析

从图 2.1、图 2.2、图 2.3 可以看出两个软件得出的 F 统计量及概率 p 值结果相同，F 检验概率 p 值小于 0.05，不同的广告形式对销售额产生了显著影响。

方差齐性检验

销售额

莱文统计	自由度 1	自由度 2	显著性
.765	3	140	.515

图 2.1 方差齐性检验(spss)

ANOVA

销售额

	平方和	自由度	均方	F	显著性
组间	5866.083	3	1955.361	13.483	.000
组内	20303.222	140	145.023		
总计	26169.306	143			

图 2.2 方差分析 F 检验(spss)

```
w=0.8271229016619859 p=0.4810325160349648
方差齐性假设不成立
f=13.483108866135096 p=8.849476396209582e-08
```

图 2.3 方差分析 python 结果

2.2 多因素方差分析

2.2.1 原理分析

多因素方差分析用来研究两个及两个以上控制变量是否对观测变量产生显著影响。多因素方差分析将观测变量的总变差分解为 SST=SSA+SSB+SSAB+SSE。SST 为观测变量的总变差；SSA,SSB 分别为控制变量 A，B 独立作用引起的变差；SSAB 为控制变量 A,B 两两交互作用引起的变差；SSE 为随机因素引起的变差。设控制变量 A 有 K 个水平，控制变量 B 有 r 个水平，SSA 的定义为

$$SSA = \sum_{i=1}^{k}\sum_{j=1}^{r} n_{ij}\left(\bar{x}_i{}^{A} - \bar{x}\right)^2 \tag{11}$$

式中，n_{ij}为因素 A 第 i 个水平和因素 B 第 j 个水平下的样本观测值个数；$\bar{x}_i{}^{A}$为因素 A 第 i 个水平下观测变量的均值。SSB 的定义为：

$$SSB = \sum_{i=1}^{k}\sum_{j=1}^{r} n_{ij}\left(\bar{x}_i{}^{B} - \bar{x}\right)^2 \tag{12}$$

式中，n_{ij}为因素 A 第 i 个水平和因素 B 第 j 个水平下的样本观测值个数；$\bar{x}_i{}^{B}$为因素 B 第 i 个水平下观测变量的均值。SSE 的定义为：

$$SSE = \sum_{i=1}^{k}\sum_{j=1}^{r}\sum_{m=1}^{n_{ij}}\left(x_{ijm} - \bar{x}_{ij}^{AB}\right)^2 \tag{13}$$

式中，x_{ijm}为因素 A 第 i 个水平和因素 B 第 j 个水平下的第 m 个观测值，$\bar{x}_{ij}^{AB}$是因素 A,B 在水平 i,j 下的观测变量的均值。于是，交互作用可解释的变差为：

$$SSAB = SST - SSA - SSB - SSE \tag{14}$$

在观测变量总离差平方和中，如果 SSA 所占比例较大，则说明控制变量 A 是引起观测变量变动的主要因素之一，观测变量的部分变动可以由控制变量 A 来解释，反之，如果 SSA 所占比例较小，则说明控制变量 A 不是引起观测变量变动的主要因素之一，观测变量的部分变动可以由控制变量 A 来解释；反之，如果 SSA 所占比例较小，则说明控制变量 A 不是引起观测变量变动的主要因素之一，观测变量的变动无法通过控制变量 A 来解释。对 SSB 和 SSAB 同理。

基本步骤：

1. 提出原假设：各控制变量不同水平下观测变量各总体的均值无显著差异，控制变量各效应和交互作用效应同时为 0
2. 选择统计量 F
3. 计算检验统计量的观测值和概率 P－值

4. 给定显著性水平 a，并作出决策

2.2.2 核心代码

选用的数据是第六章的数据方差分析(广告城市与销售额), x1 代表广告形式，x2 代表地区，x3 代表销售额，分析 x1,x2 单独作为一个变量对 x3 的影响及 x1,x2 的混合变量对 x3 的影响。

```
formula = 'x3~ C(x1)+C(x2)+C(x1):C(x2)'

moore_lm = ols(formula,data).fit()

anova_results = anova_lm(moore_lm,typ=2)

print(anova_results)
```

ols 是普通最小二乘估计，这里考虑了 x1,x2 单独对 x3 的影响及 x1、x2 交互对 x3 的影响，anova_lm 即是对最小二乘模型进行方差分析 。

```
print('tukey 多重比较')

#结果说明 reject=True，说明两种广告形式或地区有显著性差异。

print(pairwise_tukeyhsd(data['x3'], data['x1']))

print(pairwise_tukeyhsd(data['x3'], data['x2']))
```

这里是用 tukey 方法进行多重比较检验，比较哪种广告形式或地区对销售额的影响显著。

2.2.3 结果比较

从图 2.4 与图 2.5 看，两个软件出来的 F 统计量及概率 p 值是相同的，只是保留位数不同。x1、x2 的 F 统计量均远大于 1，从概率 p 值也能看出，均远小于

显著性水平 0.05，拒绝原假设，认为这两个影响因素下不同水平的方差是有显著差异的，即在不同广告形式下销售有显著差异，不同地区下销售有显著差异。x1、x2 交互变量的 F 统计量接近 1，概率 p 值大于 0.05，接受原假设，认为在交互作用下，不同水平的方差是无显著差异的。

	sum_sq	df	F	PR(>F)
C(x1)	5866.083333	3.0	23.174650	1.320657e-10
C(x2)	9265.305556	17.0	6.459472	6.602032e-09
C(x1):C(x2)	4962.916667	51.0	1.153328	2.858172e-01
Residual	6075.000000	72.0	NaN	NaN

图 2.4 Python 多因素方差分析结果

主体间效应检验

因变量：销售额

源	III 类平方和	自由度	均方	F	显著性
修正模型	20094.306[a]	71	283.018	3.354	.000
截距	642936.694	1	642936.694	7619.990	.000
x2	9265.306	17	545.018	6.459	.000
x1	5866.083	3	1955.361	23.175	.000
x2 * x1	4962.917	51	97.312	1.153	.286
误差	6075.000	72	84.375		
总计	669106.000	144			
修正后总计	26169.306	143			

a. R 方 = .768（调整后 R 方 = .539）

图 2.5 SPSS 多因素方差分析结果

3 相关分析

3.1 散点图

3.1.1 散点图作用

绘制散点图是相关分析过程中极为常用且非常直观的分析方式。它将数据以点的形式画在直角平面上。通过观察散点图能够直观发现数据点的大致走向，进而探索变量间的统计关系以及它们的强弱程度。在具体求相关系数前一般会画散点图判断整体的相关关系。

3.1.2 代码举例

以“腰围和体重”的数据举例。

```
fpath='D:\data2\waist_weight.csv'
data = pd.read_csv(fpath,header=None,names=['Waist','Weight','BodyFat'])
plt.scatter(data['Waist'],data['BodyFat'])
```

这里以 Waist 为因变量，BodyFat 做自变量画散点图。

3.1.3 结果

由图 3.1 可以看出 BodyFat 与 Waist 存在一定的正线性关系，可以用一元线性方程表示。

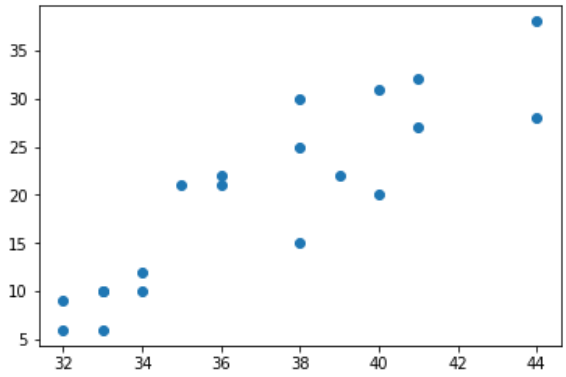


图 3.1 BodyFat 与 Waist 散点图

3.2 相关系数

3.2.1 概念

相关系数以数值的方式精确地反映了两个变量间线性相关的强弱程度，取值在-1~1 之间，r>0 表示两变量存在正的线性相关关系； r<0 表示两变量存在负的线性相关关系。r=1 表示两变量存在完全正相关关系；r=-1 表示两变量存在完全负相关关系； r=0 表示两变量不存在线性相关关系。 | r | > 0.8 表示两变量之间具有较强的线性相关关系；| r | <0.3 表示两变量之间的线性相关关系较弱。相关系数分为 Pearson 相关系数、Spearman 等级相关系数，Kendall τ相关系数。

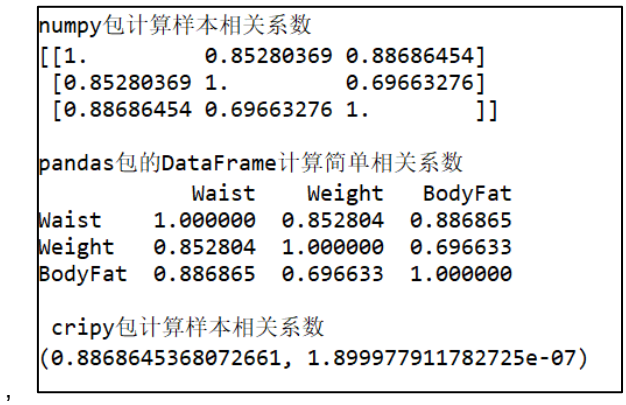
3.2.2 核心代码

代码采用第八章腰围体重的数据，包括腰围体重体脂，这里使用 python 的不同包内三个计算样本相关系数的方法，分别是 np.corrcoef,data.corr 以及 stats.pearsonr,前两个方法传入数组数据，输出数组数据，最后一个方法只能传入两个变量计算皮尔逊相关系数及概率 P 值。

```
# numpy 包计算样本相关系数
print('numpy 包计算样本相关系数')
print(np.corrcoef((data['Waist'],data['Weight'],data['BodyFat'])))
#pandas 包的 DataFrame 计算简单相关系数
print('pandas 包的 DataFrame 计算简单相关系数')
print(data.corr())
# cripy 包计算样本相关系数  两个变量
print(' cripy 包计算样本相关系数')
print(stats.pearsonr(data['Waist'],data['BodyFat']))     #(样本相关系数, P 值)
```

3.2.3 结果分析

采用案例 8-1 的数据，得到以下结果



，

图 3.2 python 计算相关系数

waist 与 BodyFat 的皮尔逊相关系数等于 0.886865，概率 p 值小于 0.05，拒绝原假设，二者有显著线性相关关系。

3.3 偏相关系数

3.3.1 概念

单纯利用相关系数来评价变量间的相关性显然是不准确的，而需要在剔除其他相关因素影响的条件下计算变量间的相关性，这就是偏相关分析的意义。偏相关分析也称净相关分析，它在控制其他变量的线性影响的条件下分析两变量间的线性相关性， 所采用的工具是偏相关系数(净相关系数)。核心公式:

$$r_{y1,2} = \frac{r_{y1} - r_{y2}r_{12}}{\sqrt{(1 - r_{y2}^2)(1 - r_{12}^2)}} \tag{15}$$

3.3.2 核心代码

```
# 偏相关系数

# python 中无模块可计算偏相关系数，自定义一个偏相关系数函数
def partial_corr(x, y, partial = []):
    xy, xyp = stats.pearsonr(x, y)
    xp, xpp = stats.pearsonr(x, partial)
    yp, ypp = stats.pearsonr(y, partial)
    n = len(x)
    df = n - 3
    r = (xy - xp*yp)/(np.sqrt(1 - xp*xp)*np.sqrt(1 - yp*yp))
    if abs(r) == 1.0:
        prob = 0.0
    else:
        t = (r*np.sqrt(df))/np.sqrt(1 - r*r)
        prob = (1 - stats.t.cdf(abs(t),df))**2
    return r,prob

pcorrelation = []
print('偏相关系数')
```

```
    for i in data[['Waist']].columns:
        pcorrelation.append(partial_corr(data[i],data['BodyFat'],partial = data['Weight']))
    print(pcorrelation)
```

这里排除了 Weight 的影响，研究 waist 与 BodyFat 的相关关系。python 没有内置的求偏相关系数的函数，使用上文的核心方程构造偏相关函数。

3.3.3 结果分析

由图 3.3 与图 3.4 得 Waist 与 BodyFat 的偏相关系数为 0.781，概率 p 值小于 0.05，即在排除 weight 因素影响的条件下 Waist 与 BodyFat 也存在显著正相关关系。

```
偏相关系数
[(0.7814305878080346, 1.5194435128641588e-09)]
```

图 3.3 Waist 与 BodyFat 的偏相关系数(python)

相关性

控制变量			腰围（英寸）	%脂肪比重
体重（磅）	腰围（英寸）	相关性	1.000	.781
		显著性（双尾）	.	.000
		自由度	0	17
	%脂肪比重	相关性	.781	1.000
		显著性（双尾）	.000	.
		自由度	17	0

图 3.4 Waist 与 BodyFat 的偏相关系数(SPSS)

4 回归分析

4.1 一元线性回归

4.1.1 步骤

回归分析是一种应用极为广泛的数量分析方法。它用于分析事物之间的统计关系，侧重考察变量之间的数量变化规律，并通过回归方程的形式描述和反映这种关系，帮助人们准确把握变量受其他一个或多个变量影响的程度，进而为预测提供科学依据。一元线性回归模型的参数通常使用普通最小二乘函数拟合。

1. 最小二乘得到参数
2. 拟合优度检验
3. 回归方程显著性检验
4. 回归系数显著性检验

代码核心采用了 OLS 模型，即最小二乘估计，达到被解释变量的所有观测值与估计值之差的平方和最小的效果，通过对残差式子求偏导求出参数方程的系数。得到参数方程为

$$\begin{cases} \widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X} \\ \widehat{\beta_1} = \dfrac{\sum x_i y_i}{\sum x_i^2} \end{cases} \tag{16}$$

4.1.2 核心代码

代码思路：载入 pandas，为了读取数据；载入 statsmodels 为了分析数据；提取 BodyFat 一列，将其变为矩阵的形式，作为 x 变量；提取 Waist 一列，作为 y 变量；增加常数（截距）；使用 OLS 进行模型拟合；查看结果

```
x = data['BodyFat']

X= x.values.reshape(-1,1)#转成矩阵形式

y = data['Waist']

#增加常数项

X1 = sm.add_constant(X)

re = sm.OLS(y,X1).fit()

print(re.summary())
```

4.1.3 结果分析

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Waist   R-squared:                       0.787
Model:                            OLS   Adj. R-squared:                  0.775
Method:                 Least Squares   F-statistic:                     66.32
Date:                Sat, 27 Jun 2020   Prob (F-statistic):           1.90e-07
Time:                        19:50:26   Log-Likelihood:                -39.217
No. Observations:                  20   AIC:                             82.43
Df Residuals:                      18   BIC:                             84.43
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         30.0575      0.949     31.657      0.000      28.063      32.052
x1             0.3540      0.043      8.144      0.000       0.263       0.445
==============================================================================
Omnibus:                        1.986   Durbin-Watson:                   1.706
Prob(Omnibus):                  0.370   Jarque-Bera (JB):                1.317
Skew:                           0.623   Prob(JB):                        0.518
Kurtosis:                       2.836   Cond. No.                         51.3
==============================================================================
```

图 4.1 一元回归结果（python）

系数[a]

模型		未标准化系数		标准化系数	t	显著性
		B	标准误差	Beta		
1	(常量)	30.058	.949		31.657	.000
	%脂肪比重	.354	.043	.887	8.144	.000

a. 因变量：腰围（英寸）

图 4.2 一元回归 t 检验(spss)

模型摘要

模型	R	R 方	调整后 R 方	标准估算的误差
1	.887[a]	.787	.775	1.812

a. 预测变量：(常量), %脂肪比重

图 4.3 一元回归拟合优度(spss)

ANOVA[a]

模型		平方和	自由度	均方	F	显著性
1	回归	217.829	1	217.829	66.320	.000[b]
	残差	59.121	18	3.284		
	总计	276.950	19			

a. 因变量：腰围（英寸）

b. 预测变量：(常量), %脂肪比重

图 4.4 一元回归 F 检验(spss)

不论是 SPSS 还是 python 都是使用普通最小二乘方法拟合参数，得到方程为 y=0.354x+30.0575,F 统计量值为 66.32，概率 p 值远小于 0.05，即整体变量可以用一元线性回归方程表示，t 检验概率 p 值都小于 0.05，即系数与 0 有显著差异。SPSS 与 python 得出的结果是相同的。

4.2 多元线性回归

4.2.1 步骤

1. 拟合参数
2. 整体显著性检验
3. 拟合优度检验
4. 系数检验
5. 自相关检验-DW

4.2.2 核心代码

多元回归方程和一元回归方程原理与步骤类似，这里的变量变成了多元 BodyFat 和 Weight。

```
#多元
x = data[['BodyFat','Weight']]
#X = x.values.reshape(-1,1)#转成矩阵形式
y = data['Waist']
#增加常数项
X1 = sm.add_constant(x)
re = sm.OLS(y,X1).fit()
print(re.summary())
```

4.2.3 结果分析

SPSS 得出的结果与 python 相同，得到方程为 y=0.065x1+0.227x2+20.236。F 统计量是 71.55，概率 P 值小于 0.05，所有变量可以用多元线性方程表示，t 检验概率 p 值均小于 0.05，即系数与 0 均有显著差异。R 方为 0.894，拟合优度好，DW=0.571，残差序列存在一定的自相关，综上多元回归方程合理。

OLS Regression Results

Dep. Variable:	Waist	R-squared:	0.894
Model:	OLS	Adj. R-squared:	0.881
Method:	Least Squares	F-statistic:	71.55
Date:	Sat, 27 Jun 2020	Prob (F-statistic):	5.27e-09
Time:	20:25:44	Log-Likelihood:	-32.235
No. Observations:	20	AIC:	70.47
Df Residuals:	17	BIC:	73.46
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	20.2355	2.468	8.199	0.000	15.028	25.443
BodyFat	0.2271	0.044	5.163	0.000	0.134	0.320
Weight	0.0654	0.016	4.144	0.001	0.032	0.099

Omnibus:	2.881	Durbin-Watson:	0.571
Prob(Omnibus):	0.237	Jarque-Bera (JB):	1.764
Skew:	0.727	Prob(JB):	0.414
Kurtosis:	3.057	Cond. No.	1.61e+03

图 4.5 多元线性回归 python 结果

模型摘要

模型	R	R 方	调整后 R 方	标准估算的误差
1	.945[a]	.894	.881	1.315

a. 预测变量：(常量), 体重（磅）, %脂肪比重

图 4.6 多元线性回归拟合优度(spss)

ANOVA[a]

模型		平方和	自由度	均方	F	显著性
1	回归	247.541	2	123.770	71.545	.000[b]
	残差	29.409	17	1.730		
	总计	276.950	19			

a. 因变量：腰围（英寸）

b. 预测变量：(常量), 体重（磅）, %脂肪比重

图 4.7 多元回归 F 检验

系数[a]

模型		未标准化系数		标准化系数	t	显著性
		B	标准误差	Beta		
1	(常量)	20.236	2.468		8.199	.000
	%脂肪比重	.227	.044	.569	5.163	.000
	体重（磅）	.065	.016	.457	4.144	.001

a. 因变量：腰围（英寸）

图 4.8 多元回归 t 检验

5 聚类分析

5.1 K-means 聚类

5.1.1 原理分析

聚类的基本思想

1.给定一个有 N 个对象的数据集，构造数据的 K 个簇，k<=n，并且满足下列条件：

每一个簇至少包含一个对象。

每一个对象属于且仅属于一个簇。

将满足上述条件的 K 个簇称作一个合理划分。

2.基本思想：对于给定的类别 K，首先给定初始的划分，通过迭代改变样本和簇的隶属关系，使得每一次改进之后的划分方案都较前一次好。

K-Means 算法

K-means 算法，也被称为 K-平均或 K-均值，是一种广泛使用的聚类算法，或者成为其他聚类算法的基础。

假定输入样本为 S=x1, x2, ……,

xm，则算法步骤为：

选择初始的 k 个类别中心，u1, u2, ……, uk。

对于每个样本的 xi，将其中标记为距离类别中心最近的类别，即：

$$label_i = arg \min_{1 \le j \le k} \|x_i - \mu_i\| \tag{17}$$

将每个类别中心更新为隶属该类别的所有样本的均值。

$$\mu_j = \frac{i}{|c_j|}\sum_{i \in c_j} x_j \tag{18}$$

重复后面的两步，直到类别中心变化小于某阈值。

终止条件:

迭代次数，簇中心变化率，最小平方误差 MSE。

5.1.2 模块核心代码解读

模块引用

from sklearn.cluster import KMeans

函数引用

KMeans(self, n_clusters=8, init=**'k-means++'**, n_init=10,

max_iter=300, tol=1e-4, precompute_distances=**'auto'**,

verbose=0, random_state=**None**, copy_x=**True**,

n_jobs=**None**, algorithm=**'auto'**) .fit(data)

传递参数

n_clusters：整型，缺省值=8,生成的聚类数。

max_iter：整型，缺省值=300,执行一次 k-means 算法所进行的最大迭代数。

n_init：整型，缺省值=10.用不同的聚类中心初始化值运行算法的次数，最终解是在 inertia 意义下选出的最优结果。

init：有三个可选值：'k-means++''random'，或者传递一个 ndarray 向量。

此参数指定初始化方法，默认值为 'k-means++'。

（1）‘k-means++’用一种特殊的方法选定初始聚类中发，可加速迭代过程的收敛。

（2）‘random’随机从训练数据中选取初始质心。

（3）如果传递的是一个 ndarray，则应该形如（n_clusters, n_features）并给出初始质心。

precompute_distances：三个可选值，‘auto’，True 或者 False。

预计算距离，计算速度更快但占用更多内存。

（1）‘auto’：如果 样本数乘以聚类数大于 12million 的话则不预计算距离。

（2）True：总是预先计算距离。

（3）False：永远不预先计算距离。

tol：float 类型，默认值= 1e-4 与 inertia 结合来确定收敛条件。

n_jobs：整形数。 指定计算所用的进程数。内部原理是同时进行 n_init 指定次数的计算。

（1）若值为-1，则用所有的 CPU 进行运算。若值为 1，则不进行并行运算。

（2）若值小于-1，则用到的 CPU 数为(n_cpus + 1 + n_jobs)。因此如果 n_jobs 值为-2，则用到的 CPU 数为总 CPU 数减 1。

random_state：确定一个 seed。此参数默认值为 numpy 的随机数生成器。

copy_x：布尔型，默认值=True 当我们 precomputing distances 时，将数据中心化会得到更准确的结果。如果把此参数值设为 True，则原始数据不会被

改变。如果是 False，则会直接在原始数据上做修改并在函数返回值时将其还原。但是在计算过程中由于有对数据均值的加减运算，所以数据返回后，原始数据和计算前可能会有细小差别。

核心代码

```
kmeans = KMeans(n_clusters=3, random_state=923).fit(df)
print(kmeans.labels_)
```

代码解读

1）KMeans(n_clusters=3, random_state=923).fit(df)

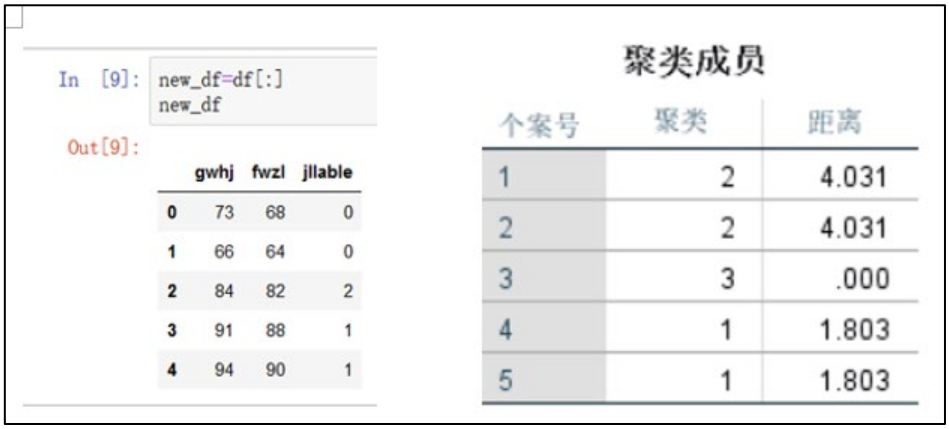
根据数据 df 进行 K-means 聚类，聚类中心为 3，随机数种子为 923

2）print(kmeans.labels_)

输出聚类后的分类信息

5.1.3 结果分析

图 5.1 是 Python 和 SPSS K-means 聚类信息对比，通过观测结果，可以得知：**两种方法实现的 K-means 聚类结果相同**，第一类：样本 1、样本 2；第二类：样本 3；第三类：样本 4、样本 5。故可以大致判定为，**Python 和 SPSS 实现的 K-means 聚类结果具有一致性。**



```
In [9]: new_df=df[:]
        new_df
Out[9]:
```

	gwhj	fwzl	jllable
0	73	68	0
1	66	64	0
2	84	82	2
3	91	88	1
4	94	90	1

聚类成员

个案号	聚类	距离
1	2	4.031
2	2	4.031
3	3	.000
4	1	1.803
5	1	1.803

图 5.1　Python 和 SPSS 的 K-means 聚类信息对比

5.2 层次聚类

5.2.1 原理分析

层次聚类的原理

层次法（Hierarchicalmethods）先计算样本之间的距离。每次将距离最近的点合并到同一个类。然后，再计算类与类之间的距离，将距离最近的类合并为一个大类。不停的合并，直到合成了一个类。其中类与类的距离的计算方法有：最短距离法，最长距离法，中间距离法，类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。

层次聚类算法根据层次分解的顺序分为：自下底向上和自上向下，即凝聚的层次聚类算法和分裂的层次聚类算法（agglomerative 和 divisive），也可以理解为自下而上法（bottom-up）和自上而下法（top-down）。自下而上法就是一开始每个个体（object）都是一个类，然后根据 linkage 寻找同类，最后形成一个“类”。自上而下法就是反过来，一开始所有个体都属于一个“类”，然后根据 linkage 排除异己，最后每个个体都成为一个“类”。这两种路方法没有孰优孰劣之分，只是在实际应用的时候要根据数据特点以及你想要的“类”的个数，来考虑是自上而下更快还是自下而上更快。至于根据 Linkage 判断“类”的方法就是最短距离法、最长距离法、中间距离法、类平均法等等（其中类平均法往往被认为是最常用也最好用的方法，一方面因为其良好的单调性，另一方面因为其空间扩张/浓缩的程度适中）。为弥补分解与合并的不足，层次合并经常要与其它聚类方法相结合，如循环定位。

层次聚类的流程

凝聚型层次聚类的策略是先将每个对象作为一个簇，然后合并这些原子簇为越来越大的簇，直到所有对象都在一个簇中，或者某个终结条件被满足。绝大多数层次聚类属于凝聚型层次聚类，它们只是在簇间相似度的定义上有所不同。这里给出采用最小距离的凝聚层次聚类算法流程：

(1) 将每个对象看作一类，计算两两之间的最小距离；

(2) 将距离最小的两个类合并成一个新类；

(3) 重新计算新类与所有类之间的距离；

(4) 重复(2)、(3)，直到所有类最后合并成一类。

5.2.2 模块核心代码解读

模块引用

```
from scipy.cluster import hierarchy
```

函数引用

```
hierarchy.linkage(y, method='single', metric='euclidean',
                                    optimal_ordering=False):
```

传递参数

y：可以是 1 维压缩向量（距离向量），也可以是 2 维观测向量（坐标矩阵）。若 y 是 1 维压缩向量，则 y 必须是 n 个初始观测值的组合，n 是坐标矩阵中成对的观测值。

method：str,'single'为最近邻点算法；'complete'为最远邻点算法；'average'为非加权组平均法；'weighted'为加权分组平均法；'centroid'为质心的无加权 paire-group 方法；'median'为 WPGMC 算法；'ward'为沃德方差最小化算法。

metric：str 或 function，默认为'euclidean'。可选。在 y 维观测向量的情况下使用该参数，否则忽略。参照有效距离度量列表的 pdist 函数，还可以使用自定义距离函数。

optimal_ordering:bool。若为 true，linkage 矩阵则被重新排序，以便连续叶子间距最小。当数据可视化时，这将使得树结构更为直观。默认为 false，因为数据集非常大时，执行此操作计算量将非常大.

核心代码

```
Z = hierarchy.linkage(df1, method ='average')
hierarchy.dendrogram(Z,orientation='right',labels = df.index)
```

代码解读

1）Z = hierarchy.linkage(df1, method ='average')

根据数据 df1 进行层次聚类，方法为非加权组平均法

2）hierarchy.dendrogram(Z,orientation='right',labels = df.index)

输出聚类后的分层图像

5.2.3 结果分析

图 5.2 是 Python 和 SPSS 的层次聚类信息对比，通过观测结果，可以得知：**两种方法实现的层次结果相同**，第一类：北京、上海、天津；第二类：江苏、山东、辽宁、浙江、广东、福建；第三类：其他省份。故可以大致判定为，**Python 和 SPSS 实现的层次聚类结果具有一致性。**

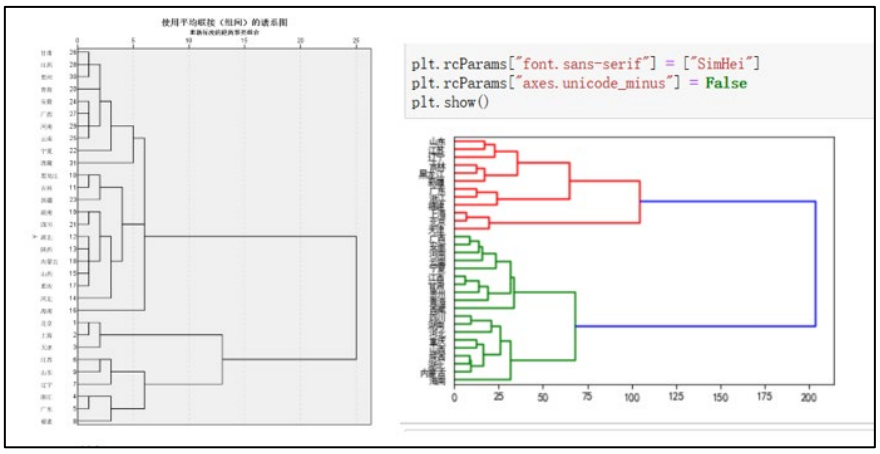


图 5.2 SPSS 和 Python 层次聚类图像对比

5.3 因子分析

5.3.1 原理分析

因子分析是指研究从变量群中提取共性因子的统计技术。最早由英国心理学家 C.E.斯皮尔曼提出。他发现学生的各科成绩之间存在着一定的相关性，一科成绩好的学生，往往其他各科成绩也比较好，从而推想是否存在某些潜在的共性因子，或称某些一般智力条件影响着学生的学习成绩。因子分析可在许多变量中找出隐藏的具有代表性的因子。将相同本质的变量归入一个因子，可减少变量的数目，还可检验变量间关系的假设。

5.3.2 模块核心代码解读

模块引用

```
from scipy.cluster import hierarchy
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
```

函数引用

```
FactorAnalyzer(n_factors=3,rotation='promax',method='minres',use_smc=True,is_corr_matrix=False,bounds=(0.005, 1),impute='median',rotation_kwargs=None)
```

传递参数

factors: 检验的因子个数，默认为 3

rotation：旋转模型：None（不旋转）；'varimax'（正交旋转）、'promax'（斜旋转）、'oblimin'（斜旋转）、' oblimax'（正交旋转）、quartimin'（斜旋转）、'quartimax（'正交旋转）、'equamax'（正交旋转），默认为"promax"

method：使用的拟合方法'minres'（最小），'ml'（最大），'principal'（等可能性），默认为"minres"

use_smc：使用平方多重相关作为因子分析的猜测，默认为 True

is_corr_matrix：数据是否有相关性，默认为 False

bounds: 变量的上下界，默认为（0.005，1）

impute:如果数据中存在缺失值的估值方法'drop'（"删除"），'mean'（"中值"），'median'（"平均值"），默认为'median'

rotation_kwargs:旋转角，可附加关键字参数传递给旋转方法。默认为None

calculate_bartlett_sphericity(x):传递参数**,**X：检验的矩阵向量

calculate_kmo(x):传递参数**,**X：检验的矩阵向量

核心代码

```
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
chi_square_value,p_value=calculate_bartlett_sphericity(df_model)
print('卡方值：',chi_square_value,'P 值', p_value)
```

```
fa = FactorAnalyzer(n_factors=5)
fa.fit(df)
fa.get_factor_variance()
```

代码解读

1）calculate_bartlett_sphericity(df_model)

根据数据 df_model 进行矩阵的相关性和巴特利球形度检验

2）FactorAnalyzer(n_factors=5)

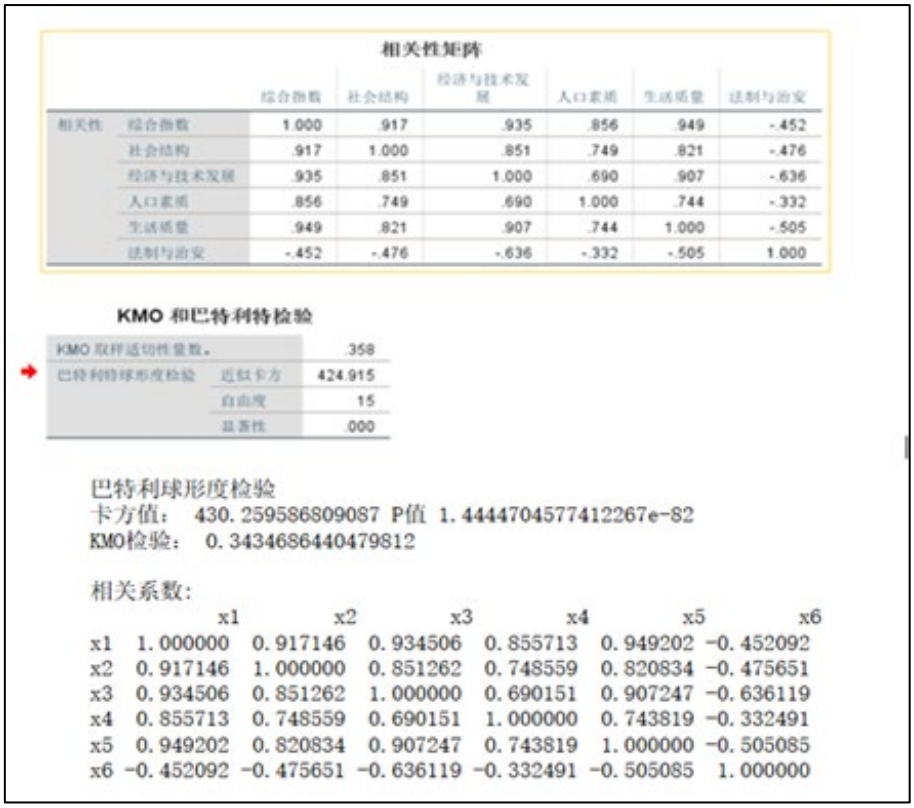
fa.fit(df)

fa.get_factor_variance()

初始化因子检验函数的因子个数为 5，将 df 数据传入到因子检验函数中，输出结果：因子的贡献率

5.3.3 结果分析

图 5.3 是 SPSS 和 Python 的相关性和巴特利球形度检验对比。Python 和 SPSS 生成的相关性矩阵，巴特利特球检验结果均相同。而且由于巴特利特球度检验统计量的观测值为 430.259，相应的概率 P-值接近 0。如果显著性水平 a 为 0.05，由于概率 P 值小于显著性水平 a，则应拒绝原假设，认为相关系数矩阵与单位阵有显著差异。同时，KMO 值为 0.3434，根据 Kaiser 给出的 KMO 度量标准可知原有变量适合进行因子分析。故可以大致判定为，**Python 和 SPSS 实现的相关性和巴特利球形度检验结果具有一致性。**



相关性矩阵

		综合指数	社会结构	经济与技术发展	人口素质	生活质量	法制与治安
相关性	综合指数	1.000	.917	.935	.856	.949	-.452
	社会结构	.917	1.000	.851	.749	.821	-.476
	经济与技术发展	.935	.851	1.000	.690	.907	-.636
	人口素质	.856	.749	.690	1.000	.744	-.332
	生活质量	.949	.821	.907	.744	1.000	-.505
	法制与治安	-.452	-.476	-.636	-.332	-.505	1.000

KMO 和巴特利特检验

KMO 取样适切性量数。		.358
巴特利特球形度检验	近似卡方	424.915
	自由度	15
	显著性	.000

巴特利球形度检验
卡方值： 430.259586809087 P值 1.4444704577412267e-82
KMO检验： 0.3434686440479812

相关系数：

	x1	x2	x3	x4	x5	x6
x1	1.000000	0.917146	0.934506	0.855713	0.949202	-0.452092
x2	0.917146	1.000000	0.851262	0.748559	0.820834	-0.475651
x3	0.934506	0.851262	1.000000	0.690151	0.907247	-0.636119
x4	0.855713	0.748559	0.690151	1.000000	0.743819	-0.332491
x5	0.949202	0.820834	0.907247	0.743819	1.000000	-0.505085
x6	-0.452092	-0.475651	-0.636119	-0.332491	-0.505085	1.000000

图 5.3 SPSS 和 Python 的相关性和巴特利球形度检验对比

图 5.4 是 SPSS 和 Python 的主成分分析对比。Python 的主成分分析显示的为总计、方差百分比和累计方差比。这 3 项的结果 **Python 和 SPSS 实现的主成分分析结果相同，因此认为结果具有一致性，函数实现成功。**

解释的总方差（即贡献率）：
(factor1 4.692125
factor2 0.761755
factor3 0.000012
factor4 0.069999
factor5 0.290811
dtype: float64, factor1 0.782021
factor2 0.126959
factor3 0.000002
factor4 0.011667
factor5 0.048469
dtype: float64, factor1 0.782021
factor2 0.908980
factor3 0.908982
factor4 0.920648
factor5 0.969117
dtype: float64)

总方差解释

成分	初始特征值 总计	初始特征值 方差百分比	初始特征值 累积 %	提取载荷平方和 总计	提取载荷平方和 方差百分比	提取载荷平方和 累积 %
1	4.692	78.202	78.202	4.692	78.202	78.202
2	.762	12.696	90.898			
3	.291	4.847	95.745			
4	.185	3.088	98.833			
5	.070	1.167	100.000			
6	1.195E-5	.000	100.000			

提取方法：主成分分析法。

图 5.4 SPSS 和 Python 的因子的贡献率对比

6 样本数据基本统计分析

6.1 直观描述

6.1.1 图类展示

从第六章开始使用医院统计数据。首先对 2007 年和 2010 年性别比例做饼图。从图 6.1、6.2 中可以看出相比于 2007 年，2020 年男性比例略微下降，女性比例略微上升。

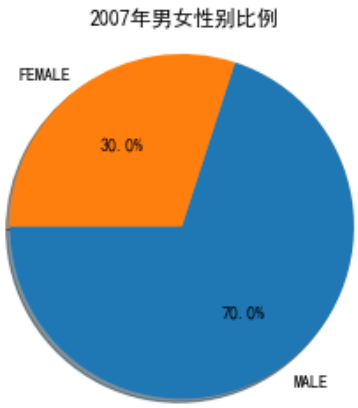


图 6.1 2007 年男女性别比例

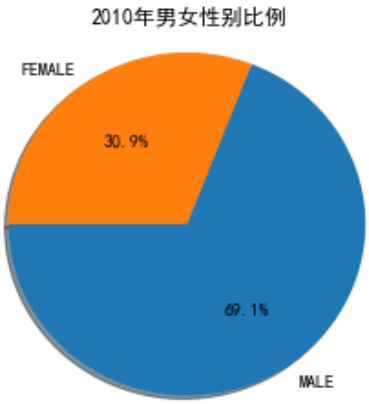


图 6.2 2010 年男女性别比例

下面根据年龄画分布直方图及标准正态分布曲线。从图 6.3，6.4 两个直方图的对比可以看出 2007 年来看门诊的居民年龄更加分散，更加符合正态分布曲线，2010 年来看门诊的居民年龄向中间集中。

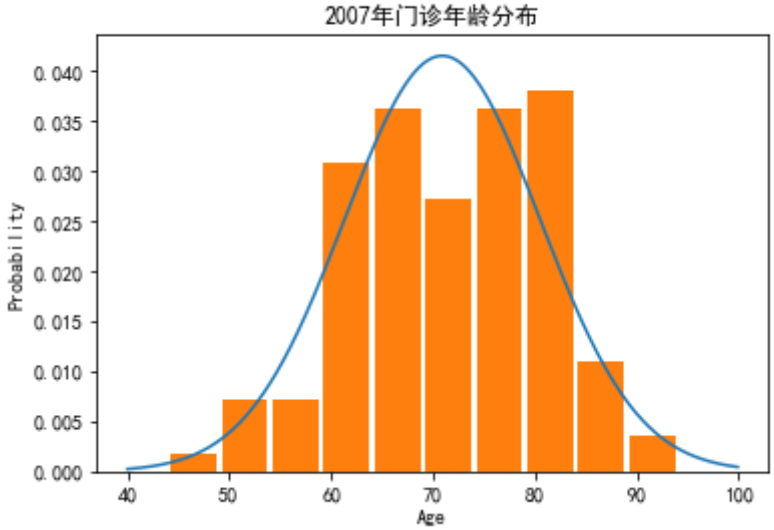


图 6.3 2007 年门诊年龄分布

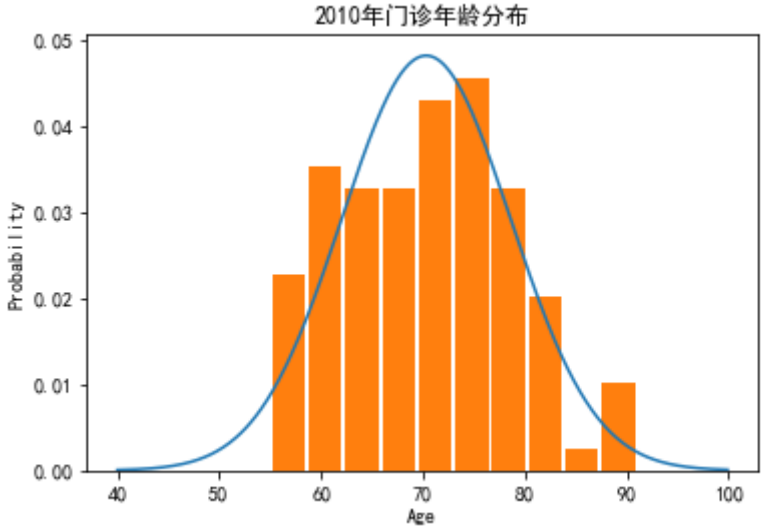


图 6.4 2010 年门诊年龄分布

6.1.2 图类代码

饼图代码：

```
#绘制饼状图
labels=['MALE','FEMALE']
#绘图显示的标签
values=[man,woman]
plt.pie(values,labels=labels,startangle=180,
shadow=True,autopct='%1.1f%%')
plt.axis('equal')
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
plt.title('2007 年男女性别比例')
plt.show()
```

使用 pie 函数画饼图，使用 plt.rcParams['font.sans-serif']=['SimHei']显示中文标签，缺少这行代码中文显示是乱码。

直方图代码：

```
def AGE():
age = sheet2.col_values(2) #获取年龄
age = np.array(age,dtype = float)
mean = age.mean() #获得年龄平均值
std = age.std() #获得年龄的标准差
# 设定 X 轴前两个数字是 X 轴的起止范围，第三个数字表示步长
# 步长设定得越小，画出来的正态分布曲线越平滑
```

```
x = np.arange(40, 100, 0.1)
# 设定 Y 轴，载入刚才定义的正态分布函数
y = np.exp(-((x - mean) ** 2) / (2 * std ** 2)) / (std * np.sqrt(2 * np.pi))
# 绘制数据集的正态分布曲线
plt.plot(x, y)
# 绘制数据集的直方图
plt.hist(age,bins=10,rwidth=0.9,density=True)
plt.title('2007 年门诊年龄分布')
plt.xlabel('Age')
plt.ylabel('Probability')
plt.show()
```

使用 numpy 导入数据，用 hist 画直方图，bins 代表分几类，可以修改。

6.1.3 基本描述

基本描述从均值、均差、偏度、峰度四个方面衡量，偏度系数(Skewness) 是描述变量取值分布形态对称性的统计量。当分布是对称分布时，正负总偏差相等，偏度值等于 0 ；当分布是不对称分布时，正负总偏差不相等，偏度值大于 0 或小于 0 。偏度值大于 0 表示正偏差值较大，为正偏或称右偏，直方图中有一条长尾拖在右边；偏度值小于 0 表示负偏差值较大，为负偏或称左偏，直方图中有一条长尾拖在左边。偏度绝对值越大，表示数据分布形态的偏斜程度越大。从图 6.5 skew 这一列的结果可以看出腰围的偏度值小于 0，负偏差值大，其他五个都是右偏，TG 的绝对值最大，即右偏幅度最大，其次是 HbA1C。峰度系数(Kurtosis) 是描述变量取值分布形态陡缓程度的统计量。当数据分布与标准正态分布的陡缓程度相同时，峰度值等于 0；峰度值大于 0 表示数据的分布比标准正态分布更陡峭，称为尖峰分布；峰度值小于 0 表示数据的分布比标准正态分布更平缓，称为平峰分布。六个变量的峰度均大于 0，是尖峰分布，分布陡峭。

```
*****************************2010年数据均值方差偏度峰度情况************************
program         average             std             skew             kurt
腰围      87.19090909090909 10.022447699118612 -0.0684844738760365 0.2745411366190402
TC      5.0949090909090895 1.0555895308853542 0.22681381918727436 0.4429529508549095
TG      1.283545454545455 0.7067078167686427 1.8618351326272102 4.984582307512522
HbA1C      6.343454545454548 0.8632384474645369 1.3423146233314336 3.5909078963440115
HDL_C      1.433181818181818 0.32430852854147596 0.6317926508329779 0.6231811913162115
LDL_C      3.141363636363636 0.8845384192782788 0.5798694047033313 1.690051111740257
```

图 6.5　2010 年数据基本描述

与 2010 年的数据比较，腰围，TC,HDL_C 这三个因素的偏度小于 0，呈左偏分布，TG,HbA1C 呈右偏分布，HbA1C 的偏度绝对值最大，右偏趋势最明显。同时只有 HbA1C 这个因素的峰度大于 0 ，比标准正态分布更陡峭，其他五个因素峰度均小于 0，比标准正态分布更平缓。

```
*****************************2007年数据均值方差偏度峰度情况************************
program         average             std             skew             kurt
腰围      92.2 10.084778247119665 -0.23106330481923373 -0.4876823831326429
TC      6.388272727272725 1.1913248034531847 -0.14493847822550365 -1.4898380630899482
TG      2.4550000000000005 0.81908625527351 0.0033098810557858995 -1.1915003888838775
HbA1C      6.602727272727272 1.2196750274740684 1.0307499240852835 0.3951158569734403
HDL_C      1.174727272727273 0.2673406550730343 -0.003567489363043766 -0.4779310998036399
LDL_C      4.361818181818183 1.3822579465711347 0.032664750960546045 -0.6475156851108603
```

图 6.6 2007 年数据基本描述

6.2 正态性检验

从图 6.7 看，统计量越接近 0 表明数据和标准正态分布拟合的越好，这里七个因素的统计量都大于 1，远大于 0，同时第三个、第四个、第五个因素的概率 p 值均小于 0.05，拒绝原假设，这三个因素样本数据不服从正态分布，其他四个（年龄、腰围、HDL_C、LDL_C）服从正态分布。（从上到下依次是年龄、腰围、TC、TG、HbA1C、HDL_C、LDL_C）。

```
**********2007年数据正态性估计**********
NormaltestResult(statistic=1.5693102376886756, pvalue=0.45627703600475555)
NormaltestResult(statistic=2.580358862733736, pvalue=0.27522139530787476)
NormaltestResult(statistic=224.67672584756818, pvalue=1.629554448417582e-49)
NormaltestResult(statistic=37.28657895769512, pvalue=8.004279675670517e-09)
NormaltestResult(statistic=16.603232078265098, pvalue=0.00024811553852660745)
NormaltestResult(statistic=1.4463293568493183, pvalue=0.4852142764862062)
NormaltestResult(statistic=3.6189066475465355, pvalue=0.16374362708757742)
```

图 6.7 2007 年数据正态性估计

同理可得 2010 年数据的正态性估计，相比于 2007 的数据，TC 的分布变成了正态分布，HDL_C 和 LDL_C 变成了非正态分布。

```
**********2010年数据正态性估计**********
NormaltestResult(statistic=1.9074909688453603, pvalue=0.3852952003198011)
NormaltestResult(statistic=0.6564943931353566, pvalue=0.7201849704468373)
NormaltestResult(statistic=2.1051135912315084, pvalue=0.3490441726414321)
NormaltestResult(statistic=54.50775424175485, pvalue=1.4581142554920222e-12)
NormaltestResult(statistic=37.41978760195505, pvalue=7.488526400892377e-09)
NormaltestResult(statistic=8.730429949283296, pvalue=0.012711922140586852)
NormaltestResult(statistic=12.470372750904797, pvalue=0.0019592640215773853)
```

图 6.8 2010 年数据正态性估计

7 数据相关性分析

7.1 相关系数

7.1.1 分析步骤

1.计算样本相关系数 r

2.对样本来自的两总体是否存在显著的线性关系进行推断

7.1.2 核心代码

先用 pandas 读取数据，然后使用 DataFrame 的 corr 方法求解多个变量之间的相关系数，corr 默认参数求取的是皮尔逊相关系数，返回值是相关系数矩阵。

```
fpath=r'D:\data2\newdata3.csv'
data =pd.read_csv(fpath,header=None,
names=['age','waist','TC','TG','HbA1C','HDL_C','LDL_C'])
print('pandas 包的 DataFrame 计算简单相关系数')
print(data.corr())
```

7.1.3 结果分析

相关系数以数值的方式精确地反映了两个变量间线性相关的强弱程度，取值在-1~1 之间，r>0 表示两变量存在正的线性相关关系； r<0 表示两变量存在负的线性相关关系。r=1 表示两变量存在完全正相关关系；r=-1 表示两变量存在完全负相关关系； r=0 表示两变量不存在线性相关关系。 I r I > 0.8 表示两变量之间具有较强的线性相关关系；I r I <0.3 表示两变量之间的线性相关关系较弱。从上图可以看出 TC 与 LDL_C 之间有强相关性，其次是 TG 与 HbA1C 。整体看

表内数据多小于 0.3 或稍大于 0.3，也就是 waist（腰围）与 TC、TG、HbA1C、LDL_C 有稍强的相关性，其他变量之间相关关系很弱。

```
**********DataFrame计算2007年数据简单相关系数**********
            age     waist        TC        TG     HbA1C     HDL_C     LDL_C
age    1.000000  0.066127  0.131219 -0.126784  0.095023  0.031661  0.131818
waist  0.066127  1.000000  0.381564  0.363194  0.468436 -0.268396  0.415569
TC     0.131219  0.381564  1.000000 -0.021802 -0.039617 -0.212102  0.821166
TG    -0.126784  0.363194 -0.021802  1.000000  0.605240 -0.273916 -0.037410
HbA1C  0.095023  0.468436 -0.039617  0.605240  1.000000 -0.179126 -0.067241
HDL_C  0.031661 -0.268396 -0.212102 -0.273916 -0.179126  1.000000 -0.131886
LDL_C  0.131818  0.415569  0.821166 -0.037410 -0.067241 -0.131886  1.000000
```

图 7.1 2007 年数据简单相关系数结果

从图 7.2 看，概率 P 值小于 0.05 则拒绝原假设，认为两总体存在显著线性关系，大于 0.05 不能拒绝原假设，认为两总体存在零相关。从概率 P 值看，waist 与 TC、TG、HbA1C、HDL_C、LDL_C 存在显著线性关系，TC 与 HDL_C、LDL_C 存在显著线性关系，TG 与 HbA1C、HDL_C 存在显著线性关系，与上文的相关系数得出的结论类似。

```
age与其他五个变量的皮尔逊相关系数及概率p值
(0.0661274453643448, 0.4924737951056747)
(0.13121924385036987, 0.1718028424712467)
(-0.1267838597538257, 0.18688067104491662)
(0.09502337919010226, 0.3234145421061123)
(0.03166080665846452, 0.7426467570917579)
(0.1318176821834745, 0.16983927838107832)

waist与其他五个变量的皮尔逊相关系数及概率p值
(0.0661274453643448, 0.4924737951056747)
(0.3815638249918479, 3.907445525298635e-05)
(0.36319389887160786, 9.629911935071652e-05)
(0.4684360628577283, 2.4584880588279473e-07)
(-0.2683959803308966, 0.004582333423619525)
(0.41556911707060845, 6.341972266400293e-06)

TC与其他五个变量的皮尔逊相关系数及概率p值
(0.13121924385036987, 0.1718028424712467)
(0.3815638249918479, 3.907445525298635e-05)
(-0.021802438221357606, 0.8211387294680974)
(-0.039616586403720525, 0.681133278078926)
(-0.21210212249005003, 0.026114270658742063)
(0.8211659266665965, 4.561351201988662e-28)

TG与其他五个变量的皮尔逊相关系数及概率p值
(-0.1267838597538257, 0.18688067104491662)
(0.36319389887160786, 9.629911935071652e-05)
(-0.021802438221357606, 0.8211387294680974)
(0.6052404991110792, 2.4936521581812945e-12)
(-0.27391582871574194, 0.003783962921196896)
(-0.037409889273977356, 0.6980071521212738)

HbA1C与其他五个变量的皮尔逊相关系数及概率p值
(0.09502337919010226, 0.3234145421061123)
(0.4684360628577283, 2.4584880588279473e-07)
(-0.039616586403720525, 0.681133278078926)
(0.6052404991110792, 2.4936521581812945e-12)
(-0.179126369766855, 0.061150432473982455)
(-0.06724149272019321, 0.485196572853576)

HDL_C与其他五个变量的皮尔逊相关系数及概率p值
(0.03166080665846452, 0.7426467570917579)
(-0.2683959803308966, 0.004582333423619525)
(-0.21210212249005003, 0.026114270658742063)
(-0.27391582871574194, 0.003783962921196896)
(-0.179126369766855, 0.061150432473982455)
(-0.13188564644276196, 0.16961733094636375)

LDL_C与其他五个变量的皮尔逊相关系数及概率p值
(0.1318176821834745, 0.16983927838107832)
(0.41556911707060845, 6.341972266400293e-06)
(0.8211659266665965, 4.561351201988662e-28)
(-0.037409889273977356, 0.6980071521212738)
(-0.06724149272019321, 0.485196572853576)
(-0.13188564644276196, 0.16961733094636375)
```

图 7.2 相关系数及概率 p 值

7.2 偏相关系数

7.2.1 分析步骤

1. 计算样本的偏相关系数

2. 对样本来自的两总体是否存在显著的净相关进行推断

7.2.2 核心代码

由上文分析，waist、TG、HDL_C 三者之间均存在显著线性关系，代码以这三个变量举例，排除 HDL_C 的影响，分析 waist 与 TG 之间的线性关系。

```
def partial_corr(x, y, partial = []):
# x，y 分别为考察相关关系的变量，partial 为控制变量
xy, xyp = stats.pearsonr(x, y)
xp, xpp = stats.pearsonr(x, partial)
yp, ypp = stats.pearsonr(y, partial)
n = len(x)
df = n - 3
r = (xy - xp*yp)/(np.sqrt(1 - xp*xp)*np.sqrt(1 - yp*yp))
if abs(r) == 1.0:
prob = 0.0
else:
t = (r*np.sqrt(df))/np.sqrt(1 - r*r)
prob = (1 - stats.t.cdf(abs(t),df))**2
```

```
return r,prob

pcorrelation = []

print('偏相关系数')

for i in data[['waist']].columns:

pcorrelation.append(partial_corr(data[i],data['TG'],partial = data['HDL_C']))

print(pcorrelation)
```

7.2.3 结果分析

偏相关系数相较原来的皮尔逊相关系数 0.36319389887160786 数值下降，即相关性下降，概率 P 值小于 0.05，拒绝原假设，在排除 HDL_C 的影响下，二者依然有显著线性关系。

```
偏相关系数
[(0.3126678163752401, 2.174258426427478e-07)]
```

图 7.3 waist 与 TG 的偏相关性分析

8　通过参数/非参数检验和聚类分析对该地区样本人群进行分析并提出建议

8.1 使用曼-惠特尼 U 检验判断影响体重因素的变化

8.1.1 分析步骤

1、通过查找资料，确定给定数据中腰围能够在一定程度上与体重呈现正相关性（ DOI：10.19813/j.cnki.weishengyanjiu.2013.04.039 ）

2、根据题意可知，由于门诊登记信息的人员变动率不大，因此就体重而言两个样本分别来自除了总体均值以外完全相同的两个总体，符合曼-惠特尼秩和检验的检验条件。

3、通过分析曼-惠特尼秩和检验的概率 P 值和 2007 年与 2010 年病人腰围的均值，判断从 2007 年到 2010 年病人的腰围是否发生显著性变化，其结果呈现什么样的趋势

8.1.2 重要代码

```
s,p = mannwhitneyu(data_2007['腰围(cm)'], data_2010['腰围(cm)'])
print(s, p)
data_2007['腰围(cm)'].mean()
data_2010['腰围(cm)'].mean()
```

8.1.3 结果分析

根据图 8.1 可以判断出两个样本的曼-惠特尼 U 检验的概率 P 值约为 0。

假定显著性水平 a 为 0.05，由于概率 P-值小于显着性水平 a，应拒绝原假设，认为 **2007 年和 2010 年的门诊患者中腰围有显著变化**。

又因为 2007 年患者的腰围平均值为约 92.2cm，而 2010 年患者的平均腰围为约 87.2cm。能够认为 2007 年到 2010 年门诊患者的腰围有显著下降的趋势。而且根据腰围与体重具有一定的正相关性的结论，可以得出**患者体重有明显的下降**。

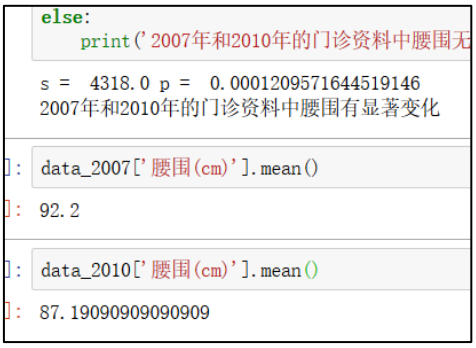


图 8.1 曼-惠特尼检验结果

8.2 使用主成分分析和独立样本 T 检验判断影响两个样本中影响糖尿病的 5 个因素是否产生显著性变化

8.2.1 分析步骤

1、 通过 PCA 算法对'TC','TG', 'HbA1C','HDL-C', 'LDL-C' 5 个变量进行降维处理，得到这五个变量的低维指标

2、将两组数据中评价指标的降维数据作为评价指标进行两独立样本 T 检验，判断数据是否有显著性变化

8.2.2 重要代码

```
pca = PCA(n_components=1)
data_2007['score'] =
pd.DataFrame(pca.fit_transform(data_2007.loc[:,"TC(mmol/L)":]))
data_2010['score'] =
pd.DataFrame(pca.fit_transform(data_2010.loc[:,"TC(mmol/L)":]))
from    scipy.stats import ttest_ind, levene
t, p = levene(data_2007['score'],data_2010['score'])
t, p = ttest_ind(data_2007['score'],data_2010['score'], equal_var = False)
```

8.2.3 结果分析

图 8.2 是主要因素的显著性分析，根据图可以得知将 5 个重要影响因素经过 PCA 降维后，进行方差齐性检验所得到的概率 P 值为 5.374209731837323e-15 约等于 0。

假定显著性水平 a 为 0.05，由于概率 P-值小于显着性水平 a，应拒绝原假设，认为 **2007 年和 2010 年的门诊患者中腰围 5 个重要影响因素的方差为非齐性**。再将降维后的数据进行配对样本 T 检验得到概率 P 值为 0.999。假定显著性水平 a 为 0.05，由于概率 P-值大于显着性水平 a，应拒绝原假设，认为 **2007 年和 2010 年的门诊患者中腰围 5 个重要影响因素的指标没有发生显著性变化**。

```
# 由于方差齐性,equal_var = True
t, p = ttest_ind(data_2007['score'], data_2010['score'], equal_var = False)
print("t=", t, "p=", p)

if p < 0.05:
    print('2007年和2010年的门诊资料中登记患者的糖尿病指标有显著变化')
else:
    print('2007年和2010年的门诊资料中登记患者的糖尿病指标无显著变化')
```

```
t= 70.72757031453884 p= 5.374209731837323e-15
方差非齐性
t= 3.1580212238914553e-15 p= 0.9999999999999976
2007年和2010年的门诊资料中登记患者的糖尿病指标无显著变化
```

图 8.2 主要因素显著性分析

8.3 根据医学期刊数据，判断该样本群体是否有可能患糖尿病

8.3.1 分析步骤

1、根据期刊《*多项指标联合检测对早期糖尿病肾病的临床价值*》(1005-8982)，中得知当 TG（甘油三酯）>6.0mmol/L、TG（总胆固醇）> 1.69mmol/L、HbA1c（糖化血红蛋白）> 6.4、HDL-C（高密度脂蛋白胆固醇测定）> 1.04mmol/L、LDL-C（低密度脂蛋白胆固醇）>130mg/dl ,即高度怀疑其存在糖尿病的可能

2、对 2007 年和 2010 年中的任意一年样本进行单样本检验，判断数据是否显著高于可能患有糖尿病的诊断值

3、根据 7.2 可知由于 2007 年和 2010 年中重要影响因素的指标没有发生显著性变化，因此可以将 2 的结论推广至该地区

8.3.2 重要代码

```
from scipy import stats
t, p=stats.ttest_1samp(data_2007['TC(mmol/L)'],6.0)
print("t=",t,"p=",p)
# 由于判断是否低于 3000，故 p 值应该除 2
p /= 2
if p < 0.05:
print('甘油三酯参数超标')
else:
print('甘油三酯参数合格')
```

8.3.3 结果分析

图 8.3 显示的是 2007 年样本的各项因子的 T 检验结果，根据图可以得知：TC、TG、HbA1C、HDL-C、LDL-C 的概率 P 值分别为 0.0008、1.251 e-16、0.084、5.4795 e-75、1.2945 e-220，由于判断的条件为单侧，所以概率 P 值应当除以 2。根据数据，假定显著性水平 a 为 0.05，由于概率 P-值小于显着性水平 a，应拒绝原假设，认为**各项指标均超出标准范围，即该样本人群有大概率患有高血糖**。由于两个样本的检查样例大致相同，而且时间跨度较长、影响因素的显著性没有明显的改变，因此可以认定**该地区的人群有高概率患有高血糖**

```
p /= 2

if p < 0.05:
    print('低密度脂蛋白胆固醇参数超标')
else:
    print('低密度脂蛋白胆固醇参数合格')
```

```
t= 3.418243880144285 p= 0.0008872528486033231
甘油三酯参数超标
t= 9.795534520137306 p= 1.2512235078386322e-16
总胆固醇参数超标
t= 1.7432689250193496 p= 0.084106428869023
糖化血红蛋白参数超标
t= 47.83436122059833 p= 5.479501225192957e-75
高密度脂蛋白胆固醇参数超标
t= -1061.1035608760183 p= 1.294515140038078e-220
低密度脂蛋白胆固醇参数超标
```

图 8.3 各项因子的结果分析

8.4 聚类分析两样本患糖尿病人群的大致趋势

8.4.1 分析步骤

1、使用 K-Mean 分别将 2007 年样本和 2010 年样本分成 3 类

2、计算每类样本的各项指标的平均值，与评判糖尿病指标做对比，给出每个

类名

3、分析分类人数，判断发展趋势

8.4.2 重要代码

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=3,
random_state=923).fit(data_2010.loc[:,"TC(mmol/L)":'LDL-C    (mmol/L)'])
data_2010['llable']=kmeans.labels_
data_2010[data_2010['llable'] == 0].loc[:,"TC(mmol/L)":'LDL-C    (mmol/L)'].mean()
data_2010.groupby('llable').apply(np.size)
```

8.4.3 结果分析

根据图 8.4，我们可以将数据大致分成三类：第一类各项指标均超过正常值，即有高概率患有糖尿病；第二类：有 2-4 项目指标超过正常值，即具有患糖尿病的可能行；第三类：指标基本处于正常值中，即有低概率患有糖尿病。

图 7-4-3 中右侧为 2007 年的样本分类，其中第一组结果为第一类，第二组结果为第三类；第 3 组结果为第二类。**第一类、第二类、第三类人群在 2007 年的占比为 572：220：418，可以约为 2.86：1：1.9**。

图 7-4-3 中左侧为 2010 年的样本分类，其中第一组结果为第三类，第 2 组结果为第一类，第三组结果为第二类。**第一类、第二类、第三类人群在 2010 年的占比为 165：649：396，可以约为 1：3.93：2.4**.

根据两年分类后的人群比可以得出：与 2007 年样本人群相比，2010 年样本人群可以因为具有了调理自身饮食、注重保养身体等原因，高概率患糖尿病的人

数和所占比例均有显著的减少，但是由于身体机能无法快速的恢复到正常值，所以大量原高概率患病的样本人员停留在具有一定概率患有糖尿病类别中。而低概率患有糖尿病的样本人群可能因为不重视相关条例事宜，所以部分样本人员到了有可能性患病的类别中。

建议相关地区继续加大对糖尿病的相关知识的宣传，对原先低概率患糖尿病人群进行科普教育，降低他们因为相关知识不充足导致的患病可能性。而对中高概率的患病人群继续保持原有的教育力度，争取做到中概率患病人群的康复结果不反弹，高概率患病人群能够降低自身的患病几率。

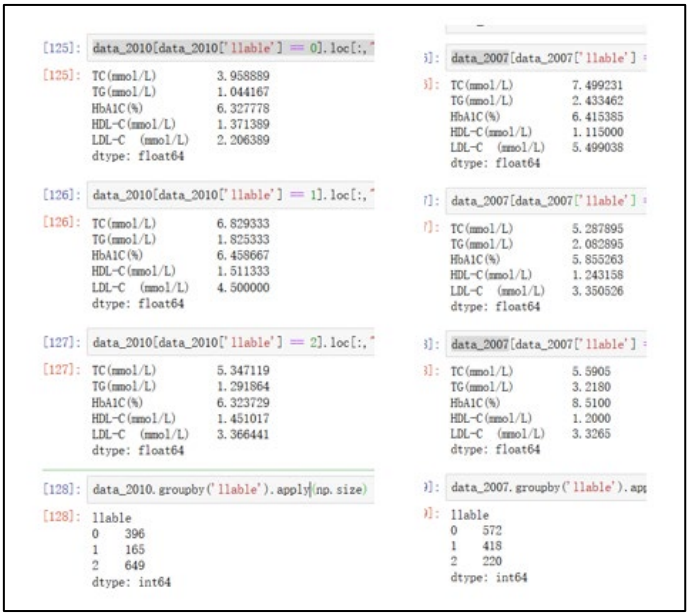


图 8.4 聚类数据对比

课设总结

宋卡妮个人总结

这次课程设计将所有学到的方法用 python 复现了一遍，这个过程对我提升起到了很大帮助 ，不仅复习了本学期学习的内容，还锻炼了代码能力，一举两得。

这次课设的难度在于对 python 代码的调用，虽然很多方法有 python 内置函数，但是要理解函数的调用方法及参数的含义，以及如何改写参数，看不懂函数时也需要找到源码跟书上的公式对比，这个过程需要大量的时间，同时这次课设工程量很大，这也是一种挑战。

课程设计的过程也是存在困难的，有时 python 代码出来的结果与 SPSS 不同，这让我很焦虑，python 的底码可以找到，但是 SPSS 的内核代码是看不见的，这俩结果不一样只能不断的从 python 代码入手，看是哪出了问题 ，一遍一遍的修改。好在是因为我代码的错误导致结果不同，经过修改代码就解决了这个问题。

这次课设感谢李金哲同学的帮助，在他的指导下我才能修改成功很多代码，同时我们俩分工明确，快速完成了很多任务，感谢队友的鼎力相助。

这次课设也存在一定不足，比如由于数据是给定的，缺少清洗数据的过程，而且给出的数据仅有一个 excel，与实际生活联系并不大，得出的结论就有些枯燥。

这学期的课程快要结束了，真是不平凡的一学期，感谢老师不厌其烦的解答，未来我也会继续努力，不断提高自己的专业能力。

李金哲个人总结

本次课程实际，使用时下流行的 python 语言，完成部分主要的 SPSS 功能实现，做到了新老结合，在增加理解基本算法原理的同时，帮助自身提高了 python 编程的能力。

在本次课程设计的过程中，我的队友起到了重要的作用，当我在完成项目思路堵塞，经常和队友讨论，一同研究问题的解决方案，从多角度、多维度的思考实现办法、函数算法，有效的提升了我的做题效率。我的队友还经常帮我发现我的思维漏洞、指出我的不足，因为疫情的缘故，目前的课程设计无法如同在校期间可以随时讨论，我们往往是完成自己的一个模块后，进行代码汇总、思维的碰撞，在这个时候队友常常可以提出一些我没有想到的好想法，帮助我提高考虑问题的角度和维度，这让我十分感动，

最后，在这次课程设计的过程中，我学会了如何分析研究代码的源码、更加深入的了解掌握的已学方法的算法和实现，有效的提升了我的代码水平和思维高度，让我感受到了数据分析的魅力，虽然本次课程设计已经结束了。但是我学习的脚步不会停止，我会继续关注有关数据分析的最新知识，努力了解和学习有关数据分析的最新算法和知识。

参考文献

[1] Rui Sarmento;Vera Costa.Comparative Approaches to Using R and Python for Statistical Data Analysis: .IGI Global, 2017. iresearchplatform. Web. 28 Jun. 2020.

[2] Rui Sarmento;Vera Costa.Comparative Approaches to Using R and Python for Statistical Data Analysis: .IGI Global, 2017. iresearchplatform. Web. 28 Jun. 2020.

[3]薛薇.基于 SPSS 的数据分析[M].中国人民大学出版社:北京,2017:69.

[4]Ivan Idris.python 数据分析[M].东南大学出版社:南京,2016:20.

[5]张轲.多项指标联合检测对早期糖尿病肾病的诊断价值[J].河南医学研究,2017,26(24):4467-4469.