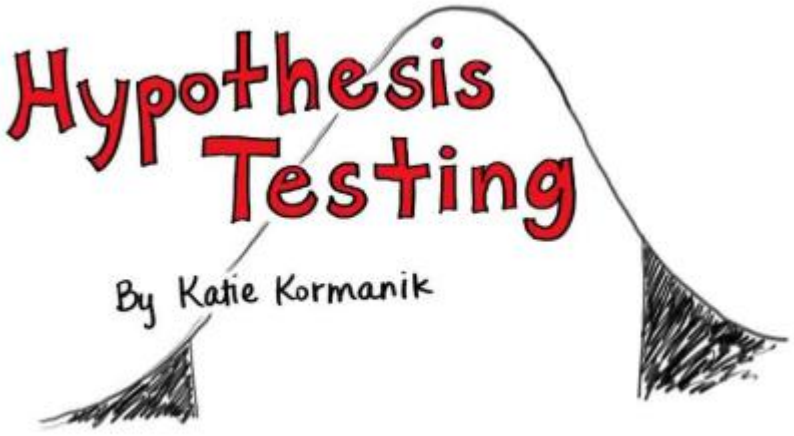
Python 统计分析

1 假设检验



1.1 为什么要学习假设检验?

从统计描述结果中发现可能的数据规律

但如果是抽样研究的样本，此时获取的只是样本的信息

研究者关心的并不仅仅是样本，更希望了了解相应的总体特征，对相应的总体的推估分以下:

- **参数估计**: 直接推估样本所在的总体特征，比如数据是否正态，均数多少，标准差多少(精确值获取不到，但可以知道一个范围)，一般能满足很多研究需求，比如说调查市场的总容量，可以通过抽样样本，然后后用样本的数据来反推一个总体的大致的总容量以及每个市场的占有率。

- **假设检验**: 假设检验就是结合之前的统计描述以及参数估计所得到的信息，再结合专业知识提出一些总体假设，然后用检验方法来对总体假设进行分析判断，最后决定之前的假设是正确还是错误，做出统计决策

假设检验

假设检验的应用价值：药物筛选研究

制药业的研发周期长达 10-15 年，最终上市药品的背后，掩埋着被淘汰的数十万候选药物，以及高达几十亿美元的开支

本质上就是一个极端化的新品研发流程

计算机分子模拟---实验室探针筛选---动物毒性实验---人体毒性实验---临床实验---上市后副作用检测

从实验室筛选开始，假设检验方法就在深入而广泛的加以应用，以尽量准确的甄别出符合需求的化合物

一个候选药物的失败，可能会使用公司几年内都元气大伤

假设检验步骤之前工作：

运用统计知识根据研究设计和资料的性质正确选择分析过程

初步的统计描述(集中趋势、离散趋势)和统计分析

- 集中趋势：均数、中位数
- 离散趋势：标准差/方差，四分位数间距
- 分布特征
- 异常值及其他

1.2 假设检验的基本原理

一家大型超市连锁店上个月接到几例消费者投诉说，某品牌的 100g 装膨化食品包装内的量太少

店方猜想引起这些投诉的原因是运输过程中沉积在食品袋底部的食品碎屑过多，而不是整包的重量不足，但为了保险起见，店方仍然决定对这批存货的平均重量进行检验

问题：

检验多少包才合适?

究竟差异要和 100g 大到什么程度才能认为确实有差异呢?

现有的样本均数和已知总体均数不同，其差别可能有两个方便的原因造成

- 样本来自己已知总体(假定的那个总体)，现有差别为抽样误差
- 样本所来自的总体与已知总体不同，存在本质差异

为了识别这两种可能，应当对其做假设检验

生活中的假设检验

- 掷骰子，猜到点数为胜

其实大家都明白如果筛子没问题，则 6 个点的出现概率应当相等(均为 1/6,这就是一个事先假设)，我们只是看每次具体的试验中谁的运气好。

假如一共下了 600 次注，竟然一共只猜中了 1 次。

- 虽然平均应当赢约 100 次，但运气太差
- 筛子有鬼，掷筛子的人可以控制结局

假设检验原理：

小概率原理，即认为小概率事件在一次随机抽样中不会发生。经典的小概率事件(瞎猫碰上死耗子)

基本思想：假设检验的基本思想是“小概率事件”原理，其统计推断方法是带有某种概率性质的反证法。小概率思想是指小概率事件在一次试验中基本上不会发生。反证法思想是先提

出检验假设，再用适当的统计方法，利用小概率原理，确定假设是否成立。即为了检验一个假设 H0 是否正确，首先假定该假设 H0 正确，然后根据样本对假设 H0 做出接受或拒绝的决策。如果样本观察值导致了“小概率事件”发生，就应拒绝假设 H0，否则应接受假设

H0 事实上，小概率事件在随机抽样中还是可能发生的，只是发生的概率很小。若正好碰上了，则假设检验的结论就是错误的。当然，犯这种错误的概率很小，是我们为了做出统计决策而愿意付出的代价

1.3 假设检验的标准步骤

1. 根据统计推断的目的而提出的对总体特征的假设

统计学中的假设有两方面的内容

检验假设/原假设/无效假设，记为 H0

备择假设，记为 H1，与 H0 相对立，意义在于当 H0 被拒绝时供采用。两者互斥

2. 确定检验水准

实际上就是确定拒绝 H0 时的最大允许误差的概率

检验水准α，是指检验假设 H0 本来成立，却根据样本信息拒绝 H0 的可能性大小，换言之，检验是拒绝了实际上成立的 H0 的概率

常用的检验水准为 0.05

其意义是：在所设 H0 的总体中随机抽得的一个样本，其均数比现有样本均数更偏离总体均数的概率不超过 5%

3. 统计量和 P 值

实际上在此之前还有一步叫做进行试验，样本数据即从此得来

统计量只是工具，概率值才是目的，它可客观衡量样本对假设总体偏离程度

从 H0 假设的总体中抽出现有样本(及更极端情况)的概率，P 值

检验统计量的特点：

该统计量应当服从某种已知分布(T 分布，卡方分布，F 分布)，可以利用统计量的数值大小、自由度的大小，从而可以计算出 P 值

各种检验方法利用的分布及计算原理不同，从而检测统计量也不同

4. 得出推断结论

按照事先确定的检验水准界定上面得到的 P 值，并按小概率原理认定对 H0 的取舍，作出推断结论

若 P<=α:

基于 H0 假设的总体情况出现了小概率事件

则拒绝 H0,接收 H1,可以认为样本与总体的差别不仅仅是抽样误差造成的，可以存在本质上的差别，属于“非偶然的”，因此，可以认为两者的差别有统计学意义

进一步根据样本信息引申，得出使用性的结论

若 P> α

基于 H0 出现了很常见的事件

则样本与总体间的差别尚不能排除纯粹由抽样误差造成，可能的确属“偶然”，不能拒绝 H0

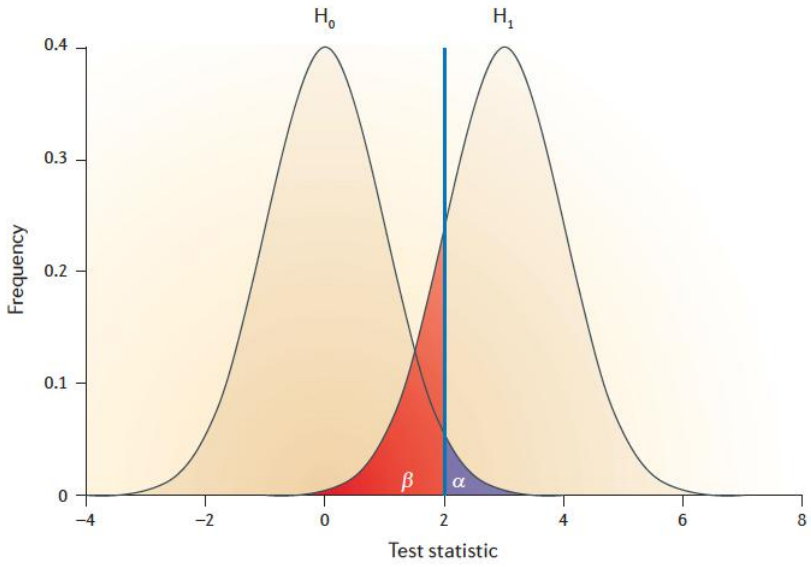
因此，认为两者的差别无统计学意义，便这并不意味着可以接受 H0

1.4 一类错误、二类错误与检验效能

	拒绝 H0,接受 H1	不拒绝 H0
H0 真实	I 型错误(α0.05)	正确推断 1-α0.05
H0 不真实	正确推断(1-β)	II 型错误（β）

检验效能：H1 是真的，实际拒绝 H0 的概率=1-β，称为 Power,又称为检验效能

由于两种可能的结论中，往往希望得到的是拒绝 H0 的结论，所以实际问题在分析时检验效能不应该太低



如何控制两种错误?

α可以事件人为设定(希望拒绝结论更可信，则减少α)

β只能间接控制(增大样本量以减小标准误，放大α减少β)

实质：牺牲一个来保障另一个

1.5 假设检验的注意事项

1.5.1 假设检验的单侧和双侧问题

双侧

不知道样本所在总体和假定总体的相应指标谁高谁低

得到拒绝结论更困难，因此相应的结果也更稳妥

单侧

在专业上可知所在总体的相应指标不可能更高/更低于假定总体值

单侧检验更为敏感，但设定单侧检验需要更有充分的专业知识支持

1.5.2 统计方法应当注意其使用条件

独立性：各观察值间相互独立，不能互相影响

正态性：理论上要求样本取自正态总体，因为正态后我们才可以只使用均数和标准差两个值来把分部固定下来

方差齐性：两样本所对应的总体方差相等/数据的离散程度相等

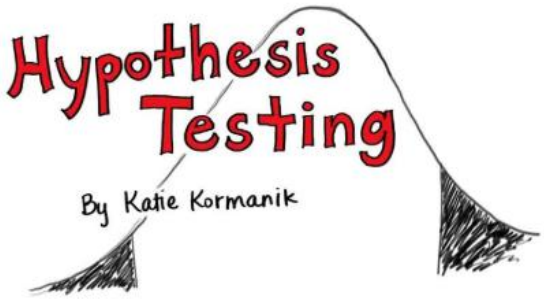
1.5.3 假设检验的结论不能绝对化

本身就保留了犯错的可能性

样本量导致的检验效能问题

- 样本量太小，导致检验效能不足，从而无法检出可能存在的差异
- 样本量太大，得出的有统计学意义的结论可能根本就没有实际意义

1.6 单样本 t 检验



单样本 T 检验用于比较样本数据与一个特定数值之间是否的差异情况（常用的假设检验最多的一种）。

例如：某类产品的质量指数为 100,但是这是全部产品的平均水平，请考察某个产品的质量是否和基准值有差异

推断样本是否来自某已知总体，即要检验样本所在总体的均数是否等于已知总体均数

为了回答该问题，统计学上采用了小概率反证法的原理：我们有如下两种假设：

H0:样本均数与(假定的)总均数的差异完全是抽样误差造成

H1:样本均数与总体均数的差异除由样本误差造成外，也反映了两个总均数确实存在的差异

先假设 H0 成立，即一切都是抽样误差造成。在这个前提下，现有的样本是从已知均数的大总体中抽出来的

$$t = \frac{\text{样本均值} - \text{总体均值}}{\text{标准误差}}$$

显然，样本均数的假设总体均数之差就代表了偏离假设的程度

但差异所对应的概率究竟是大还是小？仅看这一个数字很难做出判断。因为这还和数据的离散程度有关，为此需要找到某种方式对这个差值进行标准化(去除样本量大小与数据离散程度的影响)

标准化的基本方式就是将差值除以表示样本均数离散程度的指标(单样本的情况下，样本的均数服从 t 分布)

$$t = \frac{\bar{x} - \mu_0}{s_{\bar{x}}}$$

这个被标准化的差值，就是本次检验中所谓的统计量(由于该统计量服从 t 分布，可利用该分布得到相应的概率值，故而此处的方法被称为单样本 t 检验)

最终求得的 P 值表示从假设总体中抽出当前样本均数(及更极端情况)的概率总和

如果 P 值太小，成为了我们所定义的小概率事件(小于等于α0.05 水准)，则我们怀疑所做的假设不成立，从而拒绝 H0(基本信念：小概率事件在一次实验中不可能发生)

反之，我们就不能拒绝 H0，但一般也不太好说去接受它(尚不能认为有罪 != 可确认无罪)

1.6.1 适用的条件

因为有中心极限定理,一般均数的抽样分布都不会有问题，真正限制该方法使用的是均数是否能够代表相应数据的集中趋势(强烈偏态分布时，均数已经无法正确代表数据的集中趋势)

也就是说，只有数据分布不是强烈的偏态，一般而言单样本 t 检验都是适用

1.6.2 Python 实现单样本 t 检验

案例：往年班级学生身高平均指数是 170，请考察此次学生值数是否和基准值有差异

scipy 实现方式

scipy.stats 包中可以实现各种常用的假设检验的方法，但并没有配置详细选项，例如不能指定检验的单双侧。

scipy.stats.ttest_1samp(

a : list 格式的样本数值

popmean: H0 所对应的总体均数

)

```
from scipy import stats as ss
ss.ttest_1samp(df.身高,170)
```

statsmodels 的实现方式

DescrStatsW 类中的 tconfint_mean 可以计算可信区间，

ttest_mean 则可直接实现单样本 t 检验。

```
DescrStatsW.tconfint_mean( #  计算均数的可信区间
    alpha = 0.05
    alternative = 'two-sided'
) #  结果输出：下限、上限

DescrStatsW.ttest_mean( #  进行单样本 t 检验
value = 0 : H0 所对应的总体均数
alternative = 'two-sided' : 'larger' | 'smaller'
) #  结果输出：t 值、P 值、自由度
```

```
# %%
```

```
from statsmodels.stats import weightstats as ws

des = ws.DescrStatsW(df.身高)

des.mean

# %%

des.tconfint_mean()

# %%

des.ttest_mean(170)

# %%

des.ttest_mean(170,'smaller')

# %%

des.ttest_mean(170,'larger')
```

1.7 两样本 t 检验

1.7.1 分析目的与假设

目的：推断两个样本是否来自相同的总体，更具体地说，是要检验两个样本所代表的总体均数是否相等

- 当数据服从正态分布时，总体需要两个参数就可以确定

检验假设

无效假设：H0:μ1 = μ2

备择假设：H1:μ1 != μ2

检验水准：α = 0.05

1.7.2 基本原理

基本原理和单样本 t 检验相同

首先假设 H0:两个样本来自同一个总体

当该总体服从正态分布时，就可以采用两样本 t 检验来计算从该总体中抽得这样两个样本(及更加极端情况)的概率为多少，从而做出统计推断

1.7.3 适用条件

由于 H0 假设的是两样本来自同一总体，分析目的只涉及到均值，因此两样本 t 检验在推导过程中除了要求总体服从正态分布外，还要求两样本各自所在总体方差相同

独立性：对结果的影响较大，但一般没问题

正态性：有一定的耐受能力，可以通过直方图等进行观察(注意应当要分组考察)

方差齐性：相对而言对结论的影响较大，需要进行方差齐性检验

1.7.4 适用条件不被满足时

情况较轻时，可以采用校正 t 检验的结果

否则应使用变量变换使之满足

或采用：

- 非参数检验方法
- 贝叶斯推断方法
- 计算统计学方法

1.7.6 statsmodels 实现两样本 t 检验

statsmodels 中可以实现 t 检验的所有功能，但是未发现完成方差齐性检验
CompareMeans.ttest_ind(

alternative = 'two-sided' : 'larger' | 'smaller'

usevar='pooled' : 'pooled' or 'unequal'，方差是否齐同

value = 0 : H0 假设所对应的均数差值

)

```
from statsmodels.stats import weightstats as ws
```

```
d1 = ws.DescrStatsW(df.身高[df.性别=='男'])

d2 = ws.DescrStatsW(df.身高[df.性别=='女'])


comp = ws.CompareMeans(d1, d2)

comp.ttest_ind()

# %%

comp.ttest_ind(usevar = 'unequal')
```

CompareMeans 类下面的均数比较功能：

ttest_ind([alternative, usevar, value]) 成组设计两样本 t 检验

ttost_ind(low, upp[, usevar]) 基于成组 t 检验的等效性检验

tconfint_diff([alpha, alternative, usevar]) 基于 t 检验的均数差值可信区间

ztest_ind([alternative, usevar, value]) 成组设计两样本 z 检验

ztost_ind(low, upp[, usevar]) 基于成组 z 检验的等效性检验

zconfint_diff([alpha, alternative, usevar]) 基于 z 检验的均数差值可信区间

```
# %%

comp.ttost_ind(0, 3)

# %%

comp.tconfint_diff()

# %%

comp.ztest_ind()
```

2 检验方法条件考察

2.1 独立性的考察

不同案例间的取值互相独立，不受到除研究中考虑到的分组因素、配对因素等研究因素之外任何其他因素的影响

其他非研究因素的影响基本可以等同于随机误差

非独立的情形

研究儿童生长发育，样本中存在兄弟、表兄弟等近亲关系

流感等传染病的发病与否

数据非独立的影响较大，因为传统统计模型均按照数据独立假设进行推导，非独立会导致数据方差估计不准确（每个参数的标准误不准确），进行导致假设检验结果错误

很严重的数据非独立也会使得参数估计值出偏差

研究设计阶段

尽量保证数据不违反独立性

保证随机化真正得到实施

随机抽样、随机分组

数据分析阶段

游程检验、DW 统计量、自相关分析等考察独立性，但实际上较少使用

应对策略

采用专门的方差分析模型(如果本来做的就是重复测量数据研究)，或专门对方差等进行校正（临时）

GEEs、混合效应模型等专门的非独立数据分析模型

2.2 正态性的考察

普通方法：如 t 检验,单因素方法分析等，事实上对正态性略微偏离都有比较的耐受性，结果仍然稳健的

偏离程度过大时，仍然需要处理，且此时均数的代表性也会有问题

简单统计模型:需要进行模型残差考察，光靠观察原始变量的分布并不能得到最终结论

典型错误：多重线性回归，直接看 Y 的分布，而不做残差分析

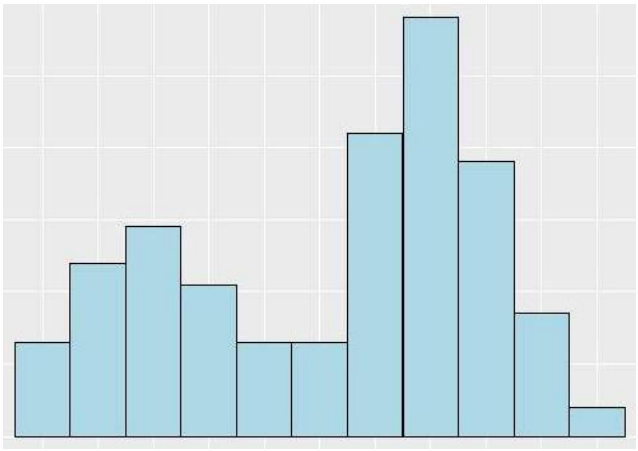
复杂统计模型：为避免结果受到干扰，原始变量都需要先做标准正态变换再建模（偏态转正态，均数为 0，标准差为 1 的标准分布，再建模）

用图形方法考察正态性

直方图、茎叶图：基于初步汇总的原始数据进行全面考察

注意需要分组考察

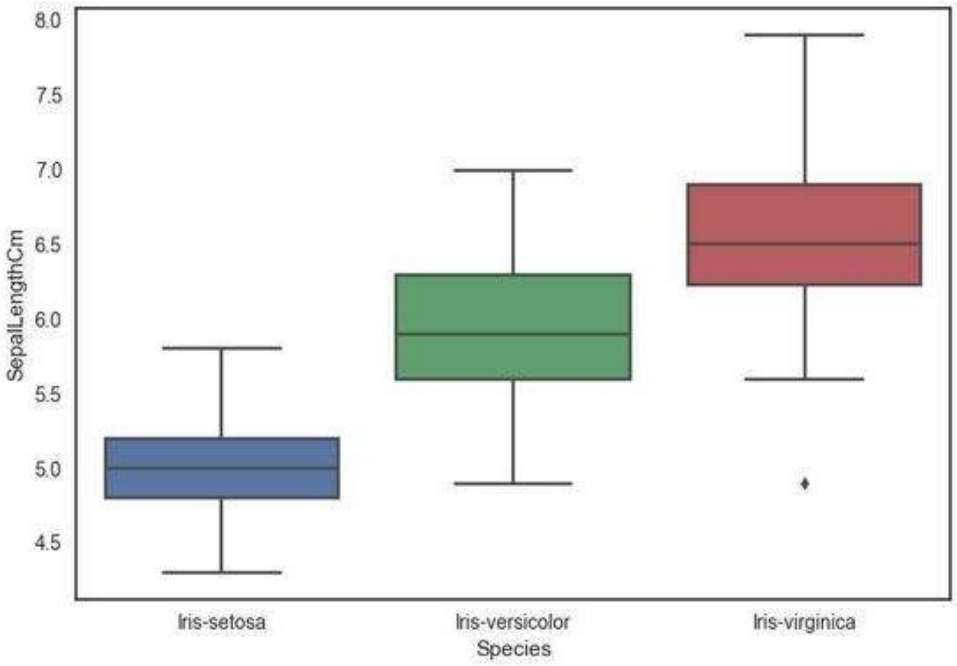
优点是简单直观，但缺点是结果不够精细



箱图：基于百分位数体系进大致观察

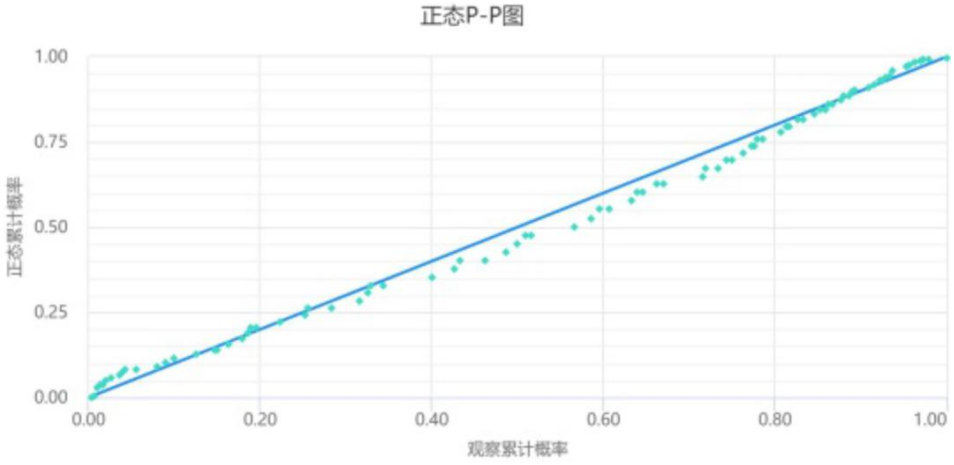
可同时考察和比较多组的数据分布

精细度比直方图还要差



P-P 图：绘制出变量实际累积概率与其假定理分布累积概率的符合程度，以判断数据是否服从所考察的分布类型。如果变量服从理论分布，则实际累积概率与理论累积概率应该基本一致

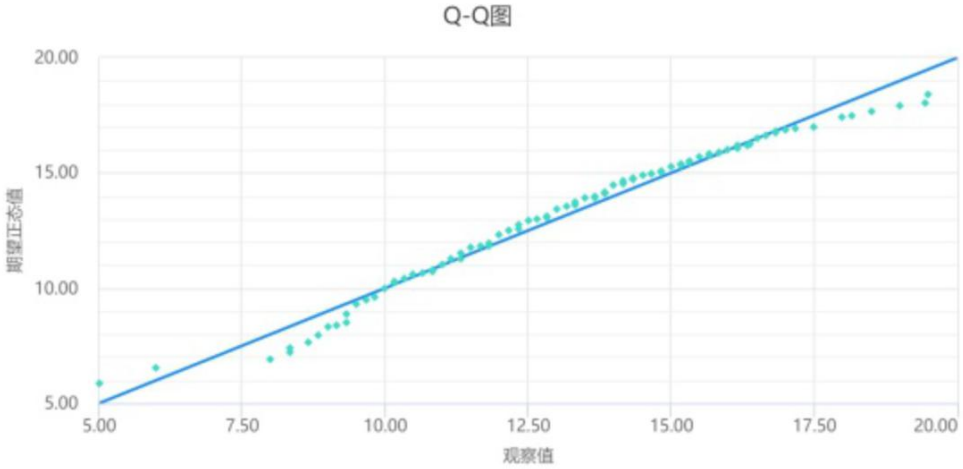
在图形上表现为实际数据的散点和理论上的直线完全重合



Q-Q 图则是感觉变量的实际百分位数与理论百分位数进行绘图，或者说得更通俗一点，相比之下 Q-Q 图的适用条件更宽松，结果也更稳健一些

1.7.5 scipy 实现两样本 t 检验

案例：根据分析，不同性别的身高可能存在差异，现希望进一步用假设检验对此差异进行确认



scipy.stats.ttest_ind(# 进行两样本 t 检验

a, b： 类数组格式的两组数值

equal_var = True： 两组方差是否齐同，方差不齐时给出 Welch’s t 检验的结果.

nan_policy = propagate： 针对缺失值的处理方式

propagate： 返回 nan

raise： 是否抛出错误

omit： 忽略 nan

)

方差齐性检验方法：

scipy.stats.bartlett() : Bartlett’s 方差齐性检验

scipy.stats.levene() : Levene 方差齐性检验，该结果针对非正态总体更稳健，相对更常用

```
df.groupby('性别').身高.describe()
# %%
ss.levene(df.身高[df.性别=='女'],df.身高[df.性别=='男'])
```

```
# %%
ss.ttest_ind(df.身高[df.性别=='女'],df.身高[df.性别=='男'])
# %%
ss.ttest_ind(df.身高[df.性别=='女'],df.身高[df.性别=='男'],equal_var=False)
```

统计计算器方式实现 t 检验

```
scipy.stats.ttest_ind_from_stats(
    mean1, std1, nobs1,
     mean2, std2, nobs2,
    equal_var = True
)
```

```
ss.stats.ttest_ind_from_stats(
    165.360000, 5.179125,25.0,
    171.444444,9.103395,27.0,
    equal_var=False
)
```

2.3 正态性的假设检验考察

2.3.1 Kolmogorov-Smirnov 检验

- 考察 H0 理论分布(可以是任何分布)下**累计相对频数**和**实际频数间**的差异
- 取频数差值绝对值的最大值为统计量
- 结论比较严格，即使只有个别案例有偏离也会拒绝 H0

2.3.2 Shapiro-Wilk 检验

- 基于数据的**偏度系数**和**峰度系数**进行正态性偏离程度的检验
- 相对 KS 检验而言更稳妥一些(由于偏度系数和峰度系数要基于全部数据进行的，所以说他也用全部数据进行检验)

2.3.3 残差分析

建模完毕后将残差储存为新变量，就可以进行图形观察甚至正态性检验

2.3.4 假设检验方式考察有缺陷

样本量偏低时往往过于迟钝

样本量较大时又会过敏，其结论会失去使用价值

2.4 Python 实现正态性的考察

2.4.1 用图形方法考察正态性

```
df.体重.hist(bins=10)
# %%
df.体重.plot.box()
```

2.4.2 用假设检验考察正态性

scipy.stats.kstest(a_vector_like_data, 'norm')

K-S 检验，特点是比较严格，理论上可以检验任何分布。

scipy.stats.shapiro(a_vector_like_data)

Shapiro 检验，专门用来检验正态分布。

scipy.stats.normaltest(a_vector_like_data)

D'Agostino and Pearson's 检验，基于峰度和偏度系数进行正态性检验。

scipy.stats.anderson(a_vector_like_data, dist='norm')

Anderson-Darling 检验，为 KS 检验的改进。

```
from scipy import stats as ss
ss.kstest(df.体重, "norm")
# %%
ks = lambda x: ss.kstest(x, "norm")
```

```
df.groupby('性别').体重.apply(ks)

# %%

df.groupby('性别').体重.apply(ss.shapiro)
```

2.5 非正态时的应对策略

2.5.1 变量变换

评估严重程度

变量变换

正偏态分布：对数变换或平方根变换

有负值时，加常数再变换

超过 30%-60%区间的率：平方根反正弦变换

传说中的 Box-cox 变换：在一个变换空间中搜索，找一个最佳的变换函数，把原始的变量变的跟接近正态分布（慎用：有可能会扭曲原始的自变量与因变量的关系）

2.5.2 更换分析方法

非参数方法(首选)：接受度高，一看就知道是干什么的，非常稳妥，可信的，前提是样本量不能特别小

秩和检验、秩变换分析

专用的分析方法

校正的 t 检验、校正的方差分析等(校正的是方差齐性的问题)

一些特殊的计算统计学方法

Bootstrap 技术

贝叶斯推断方法

2.6 方差齐性的考察

在各组间样本含量相差不太大时，方差轻微不齐仅会对方差分析的结论有少许影响

- 一般而言，只要最大/最小方差之比小于 3，分析结果都是稳定的

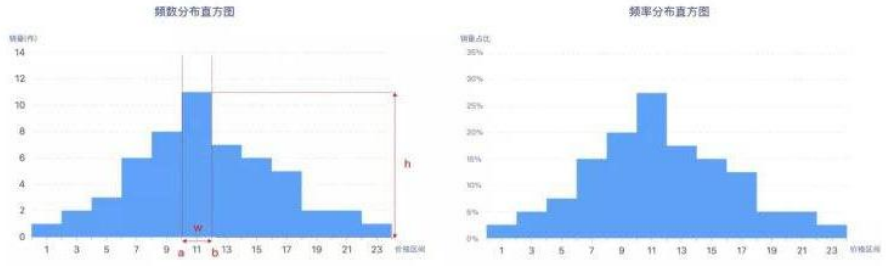
如果各组间样本含量相差过大时，方差不齐会对方差分析造成相当大的影响

主要是基于 H0 假设下的标准误估计会严重失真

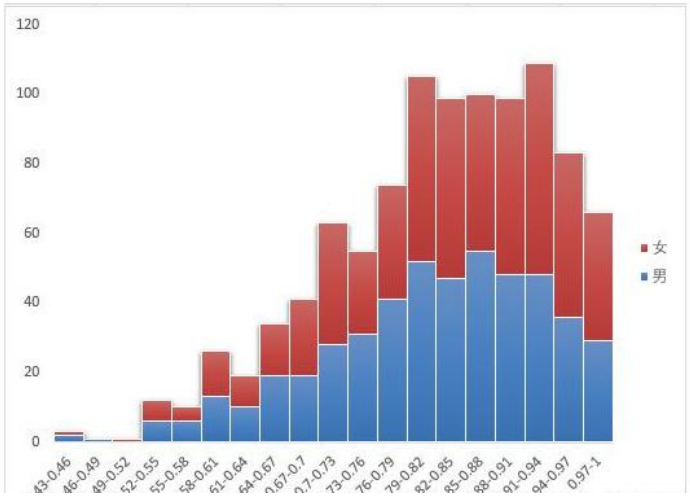
因此，统计软件往往会直接提供选项进行方差齐性检验的计算，甚至于进一步提供校正后的分析结果

2.6.1 用图形考察方差齐性

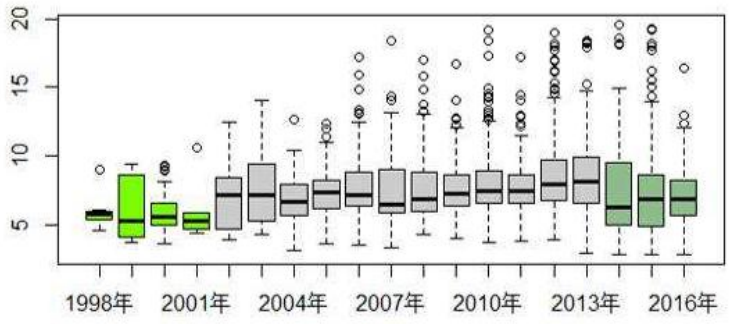
- **分组直方图**



- **堆积直方图**



- **分组箱图**



2.6.2 用假设检验方法考察方差齐性

Bartlett's 方法齐性检验

- 基于数据服从正态分布的假设对其方差齐性埋检验
- 数据略微偏离正态分布时，检验结果就会出现较大偏差

Levene 方差齐性检验

- 可用于正态或者非正态分布的资料(平均水平可以使用均数或者中位数等)
- 在数据可能非正态的情况下，精度比 Bartlett 检验好
- 可作为标准的方差齐性检验方法

Fligner-Killeen 检验

- 一种非参数的方差齐性检验方法，不依赖于具体分布

2.6.3 非方差齐性的应对策略

对原检验方法做校正

- 校正的 t 检验
- 校正的方差分析
- 校正的事后两两比较方法

方差严重不齐时，最好考虑变量变换，或者改用非参数方法

2.7 Python 实现方差齐性考察

2.7.1 用图形方法考察方差齐性

```
df.groupby('性别').体重.plot.hist()

# %%

df.体重.hist(by = df.性别)

# %%

df.boxplot(column = '体重', by = '性别')


# %%

df.groupby('性别').boxplot(column = '体重')
```

2.7.2 用假设检验考察方差齐性

scipy.stats.bartlett(a, b)

Bartlett’s 方差齐性检验，对数据有正态性要求。

scipy.stats.levene(a, b, center = {'mean', 'median', 'trimmed'})

Levene 检验，在数据非正态时精度比 Bartlett 检验好，可调中间值的度量，默认 median。

scipy.stats.fligner(a, b, center='mean')

Fligner-Killeen 检验，非参检验，不依赖于分布。

```
from scipy import stats as ss

ss.bartlett(df.体重[df.性别  == '男'],

df.体重[df.性别  == '女']

)


# %%
```

```
ss.levene(df.体重[df.性别 == '男'],
df.体重[df.性别 == '女']
)
```

3 单因素方差分析

检验某一个影响因素的差异是否会给观察变量带来显著影响

实例:

- 不同的推销策略是否对推销额产生显著影响

- 不同的运动量是否能身体健康产生显著影响

- 学习不同的课程概率统计、统计方法、编程技术、都学习过、都未学过的学习的消费水平，现希望考察 5 个类学生的消费水平是否存在差异

多组比较可以做两两检验吗?

共有 5 组均数需要比较。如果要用 t 检验进行两两比较，共要进行 10 次 t 检验

如果每次 t 检验犯 1 类错误的概率是 0.05，则不犯 1 类错误的概率为 0.95

10 次都不犯 1 类错误的概率为 0.95 的 10 次方 0.5987，

10 次 t 检验至少有一次犯 1 类错误的概率 0.4

由此可见两两检验的方式进行多组间的比较会增大犯 1 类错误的概率

3.1 基本知识

方差分析又称变异数分析(F 检验)。 对成组设计的多个样本均值比较，应采用**完全随机设计**的方差分析，即**单因素方差分析**。是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。这里，由于仅研究单个因素对观测变量的影响，因此称为**单因素方差分析**。

单因素方差分析中的认为整体样本的差异基本来源有两个:

实验条件，即不同的处理造成的差异，称为**组间差异。**用变量在各组的均值与总均值之偏差平方和的总和表示，记作 SSb，组间自由度 dfb

随机误差，如测量误差造成的差异或个体间的差异，称为**组内差异**，用变量在各组的均值与该组内变量值之偏差平方和的总和表示， 记作 SSw，组内自由度 dfw

总差异 = 组间的差异 + 组内的差异

组内 SSw、组间 SSb 除以各自的自由度(组内 dfw =n-m，组间 dfb=m-1，其中 n 为样本总数，m 为组数)，得到其均方 MSw 和 MSb，一种情况是处理没有作用，即各组样本均来自同一总体，MSb/MSw≈1。另一种情况是处理确实有作用，组间均方是由于误差与不同处理共同导致的结果，即各样本来自不同总体。那么，MSb>>MSw(远远大于)。

3.1.1 单因素方差分析概念理解步骤

例如，分析不同施肥量是否给农作物产量带来显著影响，考察地区差异是否影响妇女的生育率，研究学历对工资收入的影响等。这些问题都可以通过单因素方差分析得到答案

单因素方差分析的第一步是明确观测变量和控制变量。例如，上述问题中的观测变量分别是农作物产量、妇女生育率、工资收入；控制变量分别为施肥量、地区、学历

单因素方差分析的第二步是剖析观测变量的方差。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的离差平方和分解为组间离差平方和和组内离差平方和两部分，用数学形式表述为：SST=SSA+SSE

单因素方差分析的第三步是通过比较观测变量总离差平方和各部分所占的比例，推断控制变量是否给观测变量带来了显著影响。

3.1.2 单因素方差分析原理总结

简单理解：

在观测变量总离差平方和中，如果组间离差平方和所占比例较大，则说明观测变量的变动主要是由控制变量引起的，可以主要由**控制变量**来解释，控制变量给观测变量带来了**显著影响**；

反之，如果组间离差平方和所占比例小，则说明观测变量的变动不是主要由**控制变量**引起的，不可以主要由控制变量来解释，控制变量的不同水平没有给观测变量带来显著影响，观测变量值的变动是由**随机变量**因素引起的

3.1.3 单因素方差分析基本步骤

1、提出原假设：H0——无差异；H1——有显著差异

2、选择**检验统计量**：方差分析采用的检验统计量是 F **统计量**，即 F 值检验。

3、计算检验统计量的**观测值**和概率 P 值：该步骤的目的就是计算检验统计量的观测值和相应的概率 P 值。

4、给定**显著性水平**，并作出决策

方差分析本身是完美的，但在解决实际问题的时候，我们往往仍需要回答多个均数间究竟是哪些和哪些存在差异，这样问题又回到了两两比较上

3.1.4 方差分析的应用条件如下

- 独立，各组数据相互独立，互不相关
- 正态：即各组数据符合正态分布(因为后面推导用到了方差，均方，所以必须正态，不然没有离散程度的代表性)
- 方差齐性：即各组方差相等(因为 F 推导过程中加了这样的先用条件，类似样 t 检验考虑在了均数)

3.2 单因素方差分析 Python 的实现

案例：现希望分析根据学生对开设情况分析消费水平是否存在在差异。

3.2.1 scipy 的实现方式

scipy 由于使用的数据格式并非标准格式，代码实现上非常笨拙。

scipy 中没有提供两两比方法

```
df.groupby('开设').支出.describe()

df.groupby('开设').支出.mean().plot()


 # %%
def tranform(arg):
```

```
        if arg == '不清楚':
            return 1
        elif arg == '不必要':
            return 2
        else:
            return 3
df['开设 2'] = df.开设.apply(tranform)
df.boxplot('支出','开设 2')

# %%
from scipy import stats as ss
a = df[df.开设 2 == 1].支出
b = df[df.开设 2 == 2].支出
c = df[df.开设 2 == 3].支出
ss.levene(a,b,c)
ss.f_oneway(a,b,c)
```

3.2.2 statsmodels 的实现方式

statsmodels 中实际上是直接进行了一般线性模型的拟合，因此需要先进行模型设定

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('支出 ~ 开设', data = df).fit()
restable = sm.stats.anova_lm(model, typ = 3)
restable
```

3.3 两两比较必须解决的问题

多组比较时，无论是方差分析，还是卡方检验，在拒绝 H0 之后，所接受的 H1 都是各种情况的组合

- 各组间两两均有差异
- 各组间只有两组有差异
- 各组间有若干组存在差异

例如，如果确定了不同施肥量对农作物的产量有显著影响，那么还需要了解 10 公斤、20 公斤、30 公斤肥料对农作物产量的影响幅度是否有差异，其中哪种施肥量水平对提高农作物产量的作用不明显，哪种施肥量水平最有利于提高产量等。掌握了这些重要的信息就能够帮助人们制定合理的施肥方案，实现低投入高产出。

为此，研究者还需要进一步回答各组间究竟是哪些和哪些存在差异，这样问题又回到两两比较上

- 最直接的原始的思路就是对总的一类错误大小进行校正
- 关键：如何控制好总的一类错误大小

3.3.1 两两比较中会遇到的一类错误

CER:比较误差，即每做一次比较犯一类错误的概念

EERC:在完全无效假设下的试验误差率，即在 H0 成立时做完全部比较所犯一类错误的概率

- 事实上，很少会遇到 H0 完全成立的情形
- 方差分析检验/卡方检验本身控制的就是 EERC

MEER：最大试验误差率，即在任何完全或部分无效假设下，做完全部比较所犯一类错误的最大概率值

- MEER 的适用范围显然更广

3.3.2 直接校正 P 值：Sidak 校正

当无效假设实际上成立，各组均数无差别时，完全两两比较犯第一类错误的概率为 $1-0.95^{k(k-1)/2}$,即 EERC（约等于 0.05）

因此，进行一类错误控制时最直接的想法就是将总的α水准控制到 0.05，从而由上述公式反推得出每一个检验所使用的αij,这种校正方式被称为 Sidak 校正

注意:如果统计软件直接进行校正操作，则往往会倒过来将每个检验的 P 值放大(最大放大为 1)，而固定每个比较的α水准仍为 0.05 以便于阅读

3.3.3 直接校正 P 值：Bonferroni 校正

Sidak 校正只是在无效假设成立的情况下才适用的校正方式

- 多数实际问题中，都是有些组的均数相同，而有些组的均数不同，因此控制 MEER 更为合适

Bonferroni 不等式被广泛的用于此目的，它通过控制 CER,使得 MEER 被控制在所设定的水准以内，其计算公式为：CER=a/c

- c 为需要进行的比较次数

3.3.4 直接校正法的缺陷

将两两比较分别进行，不仅使用麻烦，也增加了误差的影响

- 联合检验可解决此类问题

对一类错误的控制太严，结果往往便宜保守

但是，当研究者希望得到的阳性结论真实可信时，直接校正法过于严格的缺陷反而变得可资利用

3.3.5 联合检验：LSD 检验

LSD 方法称为**最小显著性差异**（Least Significant Difference）**法。**最小显著性差异法的字面就体现了其检验敏感性高的特点，即水平间的均值只要存在一定程度的微小差异就可能被检验出来。

它利用全部观测变量值，而非仅使用某两组的数据。LSD 方法适用于各总体方差相等的情况，但它并没有对犯一类错误的概率问题加以有效控制

3.3.5 S-N-K 方法

S-N-K 方法是一种有效划分相似性子集的方法。该方法适合于各水平观测值个数相等的情况

3.3.6 两两比较的结果阅读和解释

常见的嵌套检验结果

- ab bc 但 a 组和 c 组有差异

统计结论本身就是概率性的，不存在递推关系

- 也存在检验效能的问题

由统计结论进行的应用性外推，统计方法本身不负任何责任

如果结果可以直接外推

- 头上一根头发都没有的人毫无疑问就是秃子

- 头上有一根头发的人和一根头发都没有的人之间看不出什么差别(差别无统计学意义)，所以也是秃子

- 依次类推，最后会得到一个满头黑发的人也是一个秃子的荒谬结论

3.4 均数间的多重比较

3.4.1 statsmodels 的实现方式

statsmodels.sandbox.stats.multicomp 包中提供了比较完整的两两比较方法，但使用上比较复杂

直接进行 P 值校正

直接给出两两比较的结果

statsmodels.sandbox.stats.multicomp.multipletests(

pvals： 类数组格式的原始 P 值

alpha = 0.05： 希望控制的总 Alpha 水准， FWER

method = 'hs'： 具体的校正方法，写全称或者可区别的前几个字母均可

'bonferroni' : one-step correction

'sidak' : one-step correction

'holm-sidak' : step down method using Sidak adjustments

'holm' : step-down method using Bonferroni adjustments

'simes-hochberg' : step-up method (independent)

'hommel' : closed method based on Simes tests (non-negative)

'fdr_bh' : Benjamini/Hochberg (non-negative)

'fdr_by' : Benjamini/Yekutieli (negative)

'fdr_tsbh' : two stage fdr correction (non-negative)

'fdr_tsbky' : two stage fdr correction (non-negative)

is_sorted = 'False' : 是否将结果按照 P 值升序排列

) # 返回值：检验结果、检验 P 值、alphacSidak、alphacBonf

```
from statsmodels.sandbox.stats import multicomp as mc
mc.multipletests([0.1, 0.2, 0.3], method = 'b')
```

GroupsStats 和 MultiComparison 命令的输出更接近 oneway ANOVA 的需求，但是目前尚未完善

```
poshoc = mc.MultiComparison(df.支出, df.开设)
res = poshoc.tukeyhsd()
res
res.summary()
```

3.4.2 scikit_posthocs 的实现

pip install scikit_posthocs

scikit_posthocs.posthoc_conover(

data

val_col =

group_col =

p_adjust =

'bonferroni' : one-step correction

'sidak' : one-step correction

'holm-sidak' : step-down method using Sidak adjustments

'holm' : step-down method using Bonferroni adjustments

'simes-hochberg' : step-up method (independent)

'hommel' : closed method based on Simes tests (non-negative)

'fdr_bh' : Benjamini/Hochberg (non-negative)

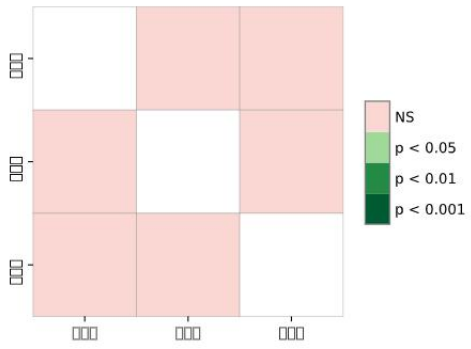
'fdr_by' : Benjamini/Yekutieli (negative)

'fdr_tsbh' : two stage fdr correction (non-negative)

'fdr_tsbky' : two stage fdr correction (non-negative)

)

```
import scikit_posthocs as sp
pc = sp.posthoc_conover(df, val_col='支出', group_col='开设',
p_adjust = 'bonferroni')
pc
```




```
# 使用热力图显示比较结果
heatmap_args = {'linewidths': 0.25, 'linecolor': '0.5', 'clip_on': False, 'square': True, 'cbar_ax_bbox': [0.80, 0.35, 0.04, 0.3]}
sp.sign_plot(pc, **heatmap_args)
```

4 非参数方法(有序分类变量)

4.1 参数统计方法的局限

需要事先明确假定总体分布（t 检验,方差分析）

总体分布未知，或分布不符合要求时无法使用

结果为有序分类变量时无法使用

样本数据两端有不确定值时无法使用

4.2 非参数分析方法的特点

不依赖总体分布的具体形式

不是对分布参数进行估计或假设检验(而是对总体的分布位置/形状进行估计或假设检验)

适用范围广，几乎可用于任何情况

注意:当资料符合参数检验方法的适用条件时，使用非参数方法的检验效能较低(效能最高的秩和检验效能可达到参数方法的 90%~95%)

4.3 分布类型/形状的检验方法

亦称拟合优度检验方法

检验样本所在总体是否服从已知的理论分布

- 多项分类变量分布的卡方检验
- 二项分类变量分布的二项分布检验
- 考察连续变量是否服从各种常用分布的单样本 K-S 检验

检验样本序列随机性的 Runs 过程

4.4 分布位置检验方法

检验样本所在总体的分布位置(集中趋势)是否相同

平时说的非参数检验方法实际上指的就是这一类方法

两独立样本/多个独立样本中心位置的检验

- 秩和检验
- 中位数检验
- 两样本 K-S 检验(本质上不是检验分布位置的方法，而是检验两样本总体曲线是否一样)

两相关样本/多个相关样本中心位置的检验

- 配对秩和检验
- 配伍 Friedman 双向秩方差分析

4.5 使用非参数检验基本思想

首先应该明确该数据确实无法使用效率更高的参数方法！(数据没正态，变换也变换不过来;数据有序分类;两端数据有不确定值)

可以考虑使用的信息量

- 中位数：大于/小于基于 H0 的中位数的样本个数分布是否均匀
- 次序关系：数值和中位数的距离可以提供进一步的信息

顺序统计量：非参数检验的理论基础

- 通过对数据从小到大排序(即排队),并用排序号(排队号)代替原始数据进行统计分析

秩(Rank)：排序号(排队号)在统计学上称为秩(秩次)

结(ties):绝对值相等称为结，又称同秩，一般取平均秩次

4.6 独立样本比较的非参数方法

Wilcoxon 两样本秩和检验

- H0：两总体所在的中心位置相同
- 中位数可以代表中心位置，但只能使用中位数的话信息量太少
- 基于 H0 假定，混合编秩，分组求秩和
- 考察各组秩和的大小是否明显偏离 H0

- 在非参数检验方法中效能最高

Mann-Whitney U

- 基于等价于两样本秩和检验
- 可近似理解为基于秩次进行了两样本 t 检验

Kolmogorov-Smirnov Z

- 两样本 K-S 检验，检验效能不高
- 考虑的是整个分布是否相同，而不是只针对中间位置

Kruskal-Wallis H 检验（针对多样本比较）

- 本质上就是基于秩次的单因素方差分析，可用于两组或多组

多组比较的事后两两比较

- 本质上仍然需要考虑如何控制总的一类错误
- 样本量较小时，也有统计学家倾向于不做检验水准的校正

4.7 Python 实现独立样本比较的非参数方法

scipy.stats.median_test()

中位数检验，两组或者多组时均可使用

scipy.stats.ranksums(a, b)

wilcox 秩和检验，相对使用较少

scipy.stats.mannwhitneyu(a, b, use_continuity, alternative)

Mann-Whitney U 检验，实际使用中一般直接代替 wilcox 秩和检验

scipy.stats.ks_2samp(data1, data2)

两样本 KS 检验

在分析中，已经确认了不同性别总消费均值存在差异，现需要进一步分析究竟是哪些构成指标导致了消费均值存在差异

```
# %%
ss.mannwhitneyu(df.开设[df.性别=='男'],df.开设[df.性别=='女'])
# %%
ss.mannwhitneyu(df.课程[df.性别=='男'],df.课程[df.性别=='女'])
# %%
ss.mannwhitneyu(df.软件[df.性别=='男'],df.软件[df.性别=='女'])
#%%
ss.ranksums(df.开设[df.性别=='男'],df.开设[df.性别=='女'])
# %%
ss.median_test(df.身高[df.性别=='男'],df.身高[df.性别=='女'])
# %%
ss.ks_2samp(df.开设[df.性别=='男'],df.开设[df.性别=='女'])
# %%
ss.kruskal(df.开设[df.性别=='男'],df.开设[df.性别=='女'])
```

scipy.stats.kruskal(sample1, sample2, ... , nan_policy = 'propagate')

nan_policy : {'propagate', 'raise', 'omit'}

多样本的 Kruskal-Wallis H 检验

现已经发现学过不同课程的况存在差异，现需要进一步分析究竟是哪些构成指标导致了出现差异

```
ss.kruskal(
    df.开设[df.课程=='都未学过'],
    df.开设[df.课程=='概率统计'],
    df.开设[df.课程=='统计方法'],
```

```
df.开设[df.课程=='编程技术'])
```

4.8 两配对样本比较：Wilcoxon 符号秩检验

注意：一般研究设计做的合理的时候，关于配对配伍的数据不大会出现违反正态性条件的情况。

求出各组配对差值

基于 H0 假定成立，差值应当围绕 0 上下对称分布

按照差值绝对值计算秩次，然后分正、负组分别计算秩和

考察正负秩物的大小是否明显偏离 H0

4.9 配伍样本比较的非参数方法：Friedman 双向秩方差分析

配伍因素影响很大，不能忽视

在配伍因素必须考虑的前提下，基于 H0 假设进行区组内编秩

按照比较组计算秩和，并比较其是否明显偏离 H0 假设

这实际上是一个没什么用的方法

- 配伍设计的资料很难会遇到不符合应用条件的情形(已经把强有力的因素匹配掉了，如果结果是连续变量)

- 该检验的检验效能非常低，与其采用秩和检验，不如进行变量变换

4.10 配伍样本比较的其它非参数方法

Kendall's W

主要用于计算 Kendall 和协系数，用于表示 k 个指标间的关联程序

Cochran's Q

是 McNemar 检验针对多组的推广，只适用于两分类资料

Friedman 双向秩方差分析只适用于连续变量

4.11 Python 实现配伍样本比较的其它非参数方法

4.11.1 配对样本

scipy.stats.wilcoxon(a, b, zero_method='wilcox', correction=False)

两配对样本的 Wilcox 符号秩检验，实际工作中很少用到

zero_method : {'pratt', 'wilcox', 'zsplit'}

检验中包括 0 差值、丢弃 0 差值、将 0 差值对半分入两组

```
df.体重.describe()
import numpy as np
df['体重 2'] = pd.Series(np.random.randint(50,85,52))
# %%
from scipy import stats as ss
ss.wilcoxon(df['体重 2'] ,df['体重'] )
ss.wilcoxon(df['支出'] ,df['体重'] )
```

4.11.2 配伍样本

scipy.stats.friedmanchisquare(measurements1, measurements2, measurements3...)

friedman 卡方检验，至少需要提供三组数据

```
ss.friedmanchisquare(
    df.支出[df.课程=='都未学过'][:5],
    df.支出[df.课程=='概率统计'][:5],
    df.支出[df.课程=='统计方法'][:5],
    df.支出[df.课程=='编程技术'][:5])
```

4.12 秩变换分析的原理

秩和检验方法都可以看作是基于 **H0 假设**求出秩次，然后对**秩次**完成相应的**参数检验**

秩变换分析方法，就是利用这一原理，基于 H0 假设成立的情况，先求出**原变量的秩次**，然后使用**秩次代替原变量**进行参数分析

当样本含量较大时，其分析结果和相应的非参数方法基本一致(原因在于相应分析方法可能对数据的分布进行一些校正，而这使用秩次方式不是考虑秩次的间断性和秩次序连续性关系的进行校正，但一般不行影响检验的结论)

但该方法可以进一步充分利用已知的参数方法，如多组样本的**两两**比较，**多变量回归**等，从而大大扩展了非参数分析方法的范围

4.13 秩变换分析的优缺点

优点:

适用范围广，**样本量充足**的情况下均可使用(秩变换后，样本可能有些小的损失)

分析结果更为**稳健**，不易受极端值影响

缺点:

检验效能相对**稍低**，存在信息损失，不适用于中小样本

而且其分析结果相对没有那么”定量”，毕竟其描述的是影响因素对因变量秩次的作用，而不是对因变量本身的作用

4.14 Python 实现秩变换分析

scipy.stats.rankdata(

a：需要编秩的数值类数组结构。

method = 'average'：对结的处理方式。

'average'：取对应秩次的平均值。

'min'/'max'：取对应秩次的最小/最大值。

'dense'：所有相同的数值只赋予一个秩次，随后继续流水编号。

'ordinal': 按照数值出现的顺序依次赋予不同的秩次。

)

案例：现希望分析根据学生对开设情况分析消费水平是否存在在差异

```
import pandas as pd


df = pd.read_excel('stu_data.xlsx')
df


# %%
tmp = df.loc[:,['开设','支出']]


# %%
from scipy import stats as ss
tmp['支出 2'] = ss.rankdata(tmp.支出)
tmp
# %%
ss.f_oneway(
      tmp[tmp.开设  =='不清楚'].支出 2,
      tmp[tmp.开设  =='有必要'].支出 2,
      tmp[tmp.开设  =='不必要'].支出 2,
      )


# %%
   tmp[tmp.开设  =='没必要'].支出 2
# %%
```

```
import scikit_posthocs as sp

sp.posthoc_conover(tmp,val_col='支出 2',group_col='开设',p_adjust="bonferroni")
```

5 卡方检验(无序分类变量)

5.1 基本原理

卡方检验是用途很广的一种方法，主要用于分类数据的统计推断

- 分类资料的分布是否符合假设（筛子是均匀）
- 两个率或两个构成比比较的卡方检验(城市与农村 2 孩的概率)
- 多个率或多个构成比比较的卡方检验(不同城市 2 孩的概率)
- 分类资料的相关分析（行变量与列变理是不有关联）
- 模型是否和样本数据完美拟合

H0: 观察频数与期望频数没有差别

其原理为考察基于 H0 的理论频数分布和实际频数分布间的差异大小，据此求出相应的 P 值

5.2 Python 实现卡方检验

5.2.1 scipy 实现

scipy 中列联表相关的功能被拆分到多个命令中

chisquare(f_obs[, f_exp, ddof, axis]) 单样本卡方检验

- 检验一个分类变量，哪个类别的概率或者规律和我们假设的总体规律是一样

chi2_contingency(observed[, correction, lambda_])

列联表卡方检验

contingency.expected_freq(observed)

列联表的期望频数

contingency.margins(a)

列联表的边际汇总

fisher_exact(table[, alternative])

为 2*2 表格计算 Fisher 确切概率检验结果

scipy.stats.chi2_contingency(

observed：观察到的列联表，类二维数组结构

correction = True：是否计算 Yates 校正的卡方结果

lambda_ = None：换用 Cressie-Read power divergence family 统计量

) # 输出结果：卡方值，P 值，自由度，期望频数

```
# %%
t = pd.crosstab(df.性别,df.开设)
t
# %%
# 列联表的期望频数
ss.contingency.expected_freq(t)
# %%
# 列联表的边际汇总
ss.contingency.margins(t)
# %%
# 列联表卡方检验
ss.contingency.chi2_contingency(t,False)
```

5.2.2 statsmodels 的实现方式

statsmodels 中首先需要建立对应的列联表对象。

class statsmodels.stats.contingency_tables.Table(

table

shift_zeros = True： 如果有单元格频数为 0，则所有单元格频数一律+0.5 防止计算溢出

)

Table 类的属性注意：有些功能尚未完成

table_orig 表格原始数据

marginal_probabilities 估计的行列边际概率分布

independence_probabilities 基于行列独立的 H0 假设的单元格概率分布(单元格的百分比)

fittedvalues 期望频数

resid_pearson Pearson 残差(相应每个单元格的贡献量的卡方平方根)

standardized_resids 标化残差

chi2_contribs 每个单元格的卡方贡献值

local_logodds_ratios 拆分成多个相邻 2*2 表格后计算的 lnOR 值

local_oddsratios 拆分成多个相邻 2*2 表格后计算的 OR 值

cumulative_log_oddsratios 依次切分成 2*2 表格后计算的 lnOR 值

cumulative_oddsratios 依次切分成 2*2 表格后计算的 OR 值

Table 类的方法

test_nominal_association() 无序分类行、列变量的独立性检验(卡方检验)

test_ordinal_association([row_scores, ...]) 有序分类行/列变量独立性检验

Cochran-Armitage 趋势检验

```
# %%
```

```
from statsmodels.stats import contingency_tables as tbl
table = tbl.Table(t)
table

# 表格原始数据
table.table_orig

# 期望频数
table.fittedvalues
# Pearson 残差
table.resid_pearson
# 每个单元格的卡方贡献值
table.chi2_contribs

# 估计的行列边际概率分布
table.marginal_probabilities
```

test_nominal_association()返回一个 Object，通过以下属性获取值：

statistic： 卡方统计量

df： 自由度

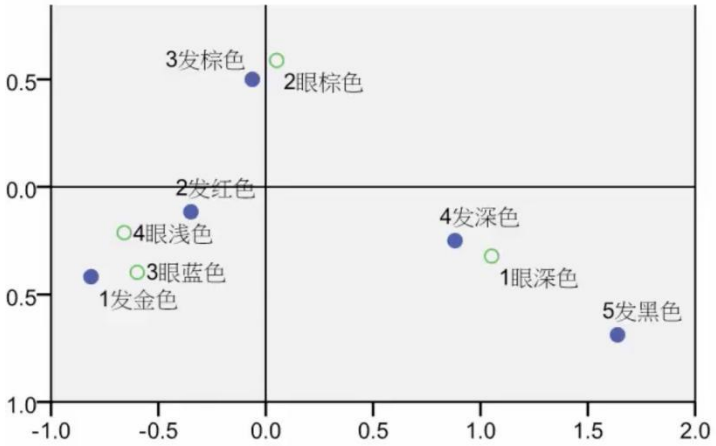
pvalue : P 值

```
res = table.test_nominal_association()
print(res.statistic,res.df,res.pvalue)
```

5.3 卡方检验的两两比较

当多分类变量检验出总体存在差异之后，

- 仍需要进行检验水准校正之下的两两比较
- 复杂建模方法，进行哑变量的精细比较(比较适合做研究，一般不用于商用)
- 也可以换用图示分析的方法



5.4 Python 实现卡方检验的两两比较

python 中目前没有提供对列联表自动拆分进行两两比较的功能

statsmodels.sandbox.stats.multicomp 等程序包提供的是直接对 P 值进行校正的方法，因此也可以对卡方检验的两两比较结果进行校正。

Table 类中的两两拆分或者依次切分功能可以作为初步观察工具，未来可能会发展成两两比较命令。

注意：有些功能尚未完成

local_logodds_ratios 拆分成多个相邻 2*2 表格后计算的 lnOR 值

local_oddsratios 拆分成多个相邻 2*2 表格后计算的 OR 值

cumulative_log_oddsratios 依次切分成 2*2 表格后计算的 lnOR 值

cumulative_oddsratios 依次切分成 2*2 表格后计算的 OR 值

（OR 值等于 1，表示该因素对疾病的发生不起作用；
OR 值大于 1，表示该因素是危险因素；
OR 值小于 1，表示该因素是保护因素。）

```
from statsmodels.stats import contingency_tables as tbl
table = tbl.Table(t)
table.local_oddsratios
# %%
table.cumulative_oddsratios
```

tab.local_oddsratios

性别	开设不必要	不清楚	有必要
女	0.4	5.7	NaN
男	NaN	NaN	NaN

tab.cumulative_oddsratios

性别	开设不必要	不清楚	有必要
女	1.389474	3.5625	NaN
男	NaN	NaN	NaN

5.5 卡方校正

英国统计学家 Yates 认为，X^2 分布是一种连续分布，而四格表资料是分类资料，属离散分布，由此计算的 X^2 值的抽样分布也应当是不连续的

当样本量较小时，两者间的差异不可忽略，应进行连续性校正

Yates 提出的办法是每个单元格的残差中都减去 0.5

- 若 n>40,此时有 1 < T < 5 时，需计算 Yates 连续性校正 X^2 值

- T < 1，或 n < 40 时，应改用 Fisher 确切概率法直接计算概率

5.5.1 如何进行列联表的检验

一般的看法

- 当 n>=40 且所有 T>=5 时，用普通的卡方检验，若所得 P 约等于 Alpha,改用确切概率法

- 当 n>=40 但有 1 <= T < 5 时，用校正的卡方检验（如果有）

- 当 n<40 时或有 T<1 时，不能用卡方检验，改用确切概率法

建议的做法

- 卡方检验可能有问题时，直接使用确切概率法
- 计算量太大，无法得出确切概率时，使用蒙特卡洛抽样计算概率区间

5.6 Python 实现卡方校正与确切概率法

```
scipy.stats.fisher_exact(
    table
    alternative = 'two-sided' : {'two-sided', 'less', 'greater'}
) # 输出结果：先验 OR 值，确切概率值
```

```
ss.chi2_contingency(t2)
# %%
ss.fisher_exact(t)
# %%
def trans(arg):
    if arg == '有必要':
        return arg
    else:
        return '不清楚'
df['开设 2'] = df.开设.apply(trans)
# %%
t2 = pd.crosstab(df.性别,df.开设 2)
t2
# %%
ss.fisher_exact(t2)
```

5.7 配对卡方检验

用 A、B 两种方法检查已确诊的某种疾病患者 140 名，A 法检出 91 名(65%)，B 法检出 77 名(55%)，A、B 两法一致的检出 56 名(40%),问哪种方法阳性检出率更高?

A法		B法	合计
	+	−	
+	56a	35b	91
−	21c	28d	49
合计	77	63	140

对同一个体，分别有两次不同的测量，并最终构成了两组数据，因此研究框架是自身配对设计

基本思路：求出各对的差值，然后考察样本中差值的分布是否按照 H0 假设的情况对称分布

显然，主对角线上的样本，两种检验方法的结论相同

H0：两法总体阳性检出率无差别，即 B=C

H1：两法总体阳性检出率有差别，即 B!=C

- 非主对角线上的单元格才携带检验方法的差异信息

根据 H0 得到 b、c 两格的理论数均为 Tb=Tc=（b+c）/2,对应的配对检验统计量为

$X^2=(b-c)^2/b+c$,v=1

- 一般在 b+c <40 时，需要确切概率法进行检验，或者进行校正

5.8 Python 实现配对卡方检验

案例：用 A、B 两种方法检查已确诊的某种疾病患者 140 名，A 法检出 91 名(65%)，B 法检出 77 名(55%)，A、B

两法一致的检出 56 名(40%)，问哪种方法阳性检出率更高?

5.8.1 配对卡方检验的实现

statsmodels.stats.contingency_tables 中有三个模块都涉及到配对卡方的分析问题。

tbl.SquareTable 用于分析行列变量类别相同的对称结构方表（近似结果）

tbl.mcnemar 用于分析配对四格表（确切概率结果）

bl.cochrans_q 将配对四格表检验从两组向多组扩展的方法，两组时等价于 mcnemar

用 mcnemar 类分析

```
import statsmodels.stats.contingency_tables as tbl
table = tbl.mcnemar(pd.DataFrame([[56, 35], [21, 28]]))
table.pvalue
```

```
import pandas as pd...

0.08142681460950622
```

用 SquareTable 类分析

class statsmodels.stats.contingency_tables.SquareTable(

table：表格的行列变量必须按照相同分类、相同顺序排列，且指定为方阵结构

shift_zeros = True

)

SquareTable 类的方法

summary([alpha, float_format])：输出下面两个检验的汇总结果

symmetry([method])：检验表格是否沿主对角线对称,2x2 表时即为 mcnemar 卡方

homogeneity([method])：检验行列变量的边际分布是否相同

```
import numpy as np
import statsmodels.stats.contingency_tables as tbl
table = tbl.SquareTable(np.asarray([[56, 35], [21, 28]])) # 必须为方阵结构数据
table


table.summary()
```

```
tbl.SquareTable(pd.DataFrame([[5, 35], [21, 2]])).summary()
```

	Statistic	P-value	DF
Symmetry	3.500	0.061	1
Homogeneity	3.500	0.061	1

5.9 二项分布检验

之前一直用卡方检验，因为卡方检验用的最多。但是有些检验，还是需要回归到最简单的二项分布，例如：计算率的可信区间

5.9.1 Bernoulli 试验序列

在重复实验中，如果对每一次实验，出现的结果只两种情况，即 Bernoulli 试验

每次试验的条件不变，即每次试验中，结果 A 发生的概率不变(假设均为 pi)

各次试验独立，即一次试验出现什么样的结果与前面已出现的结果无关

满足以上三个条件的 n 次 Bernoulli 试验构成的序列被称为 Bernoulli 试验序列

5.9.2 二项分布

对于 Bernoulli 试验序列的 n 次试验，结局 A 出现的次数 X 的概率分布服从二项分布

二项分布指的是概率的分布

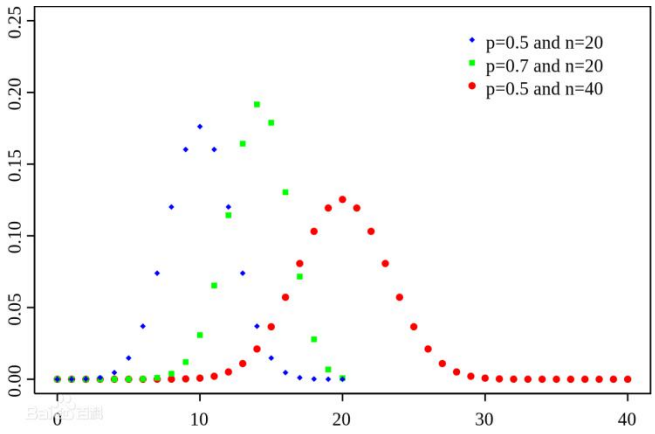
二项分布是一个离散型分布

一般地，如果随机变量 X 服从参数为 n 和 p 的二项分布，我们记为 X~B(n，p)或 X~b(n,p)。n 次试验中正好得到 k 次成功的概率由概率质量函数给出

$$P\{X=k\}=\binom{n}{k}p^k(1-p)^{n-k}.$$

当 pi=0.5 时，图形对称;当 pi!=0.5，图形呈偏态，但随 n 的增大，图形逐渐对称

因此，当 n 较大，pi 不太极端时，可以采用正态近似方法计算概率分布规律(例如计算参考值范围，计算可信值区间，假设性检验)



5.9.2 样本率的抽样分布

对于大量重复随机抽样而言，样本率 P 围绕着总体率波动，样本量 n 的值越大，这种波动就越小

当 n 的值充分大时，p 的分布就近似于均数为 pi,标准差为 sqrt(pi*(1-pi)/n)的正态分布

- 一般的标准是 n*pi 和 n*(1-pi)均大于 5，且 n>40

- 当样本情况接近此标准时，往往进行校正

注意：上文所说的样本率的标准差，实际上就是阳性数 x 的标准误

5.9.3 总体率的区间估计

直接计算概率

- 在样本例数较小，且样本率接近 1 或 0，即阳性事件发生率很高或很低时，应当按照率的抽样分布规律确定总体率的可信区间（整个率的抽样分布，明显偏离正态分布）

正态近似

- 当 n 较大，pi 和 1-pi 均不太小时，样本率的抽样分布近似正态分布;因此可按正态近似法求总体率的 1-a 的可信区间

5.9.3 率的比较

如前所述，当 n 较大，pi 和 1-pi 均不太小时，样本率的抽样分布近似正态分布，可利用正态分布的原理作假设检验

反之，则可使用二项分布自身的概率分布进行假设检验，这种方法被称为二项分布检验(确切概率法)

5.10 Python 实现二项分布检验

5.10.1 率的可信区间

statsmodels.stats.proportion.proportion_confint(# 二项分布指标的率的可信区间

count： 成功次数

nobs： 总样本数

alpha = 0.05

method = 'normal'： 具体可信区间的计算方法，默认为正态近似法

normal : asymptotic normal approximation

agresti_coull : Agresti-Coull interval

beta : Clopper-Pearson interval based on Beta distribution

wilson : Wilson Score interval

jeffreys : Jeffreys Bayesian Interval

binom_test : experimental(试验性的), inversion of binom_test 基于二项分布精确的可信区间

) # 结果输出：可信区间上、下限

statsmodels.stats.proportion.multinomial_proportions_confint(

多项分布指标的率的可信区间

counts： 每个类别的观测频数，类一维数组结构

alpha = 0.05

method = 'goodman' : 可信区间的计算方法

goodman: 基于卡方近似，适用条件同卡方检验

sison-glaz: 更准确一些，但只针对有 7 个及以上类别时才有效

)

```
from statsmodels.stats import proportion as sp
sp.proportion_confint(5, 10)
sp.proportion_confint(5, 10, method = 'binom_test')
sp.multinomial_proportions_confint([5, 5])
```

5.10.2 二项分布检验

statsmodels.stats.proportion.binom_test(# 基于二项分布的单样本确切检验 (两样本 fisher)

count

nobs

prop = 0.5 : H0 所对应的总体率

alternative = 'two-sided'

) # 输出：P 值

statsmodels.stats.proportion.proportions_chisquare(# 基于卡方的多项分布检验

count : 按列表形式给出各类别频数

nobs

) # 输出：卡方值、P 值、 (表格，期望值)

```
sp.binom_test(1, 600)
sp.proportions_chisquare([5, 3, 2], 10)
```

5.10.3 近似 Z 检验

statsmodels.stats.proportion.proportions_ztest(

count : 成功次数，单一数值/类数组结构列表

nobs : 总样本量，单一数值/类数组结构列表

value = None : H0 所对应的总体率/率差

alternative = 'two-sided'

prop_var = False : 指定方差分配比例，默认按照样本比例进行计算

) # 输出：Z 统计量、P 值

```
sp.proportions_ztest(30, 100, 0.2)
sp.proportions_ztest([30, 65], [100, 200], 0)
```

6 关联性分析(相关分析)

6.1 什么是变量的关联分析

用于考察变量间数量(数值)关联密切程度的统计分析方法

- 焦不离孟，孟不离焦
- 身高越高，则体重一般也越大

对于统计学来说，研究就是数值变异的学科，对于多个变量之间的变异数值关联，其实就是关联分析了。

几乎所有涉及到多个变量的假设检验方法，都可以被看作是这些变量间的关联性分析

- t 检验：分组变量与连续因变量间的关联性分析
- 卡方检验：行、列分类变量间的关联性分析
- 聚类分析：案例间的关联性分析
- 多变量回归：因变量和一组自变量间的关联性分析

6.2 相关分析的分类

按照变量数量

- 一个变量 VS 另一个变量
- 一个变量 VS 一组变量
- 一组变量 VS 另一组变量
- 多组变量之间的相关分析

按照变量种类

- 连续变量
- 有序分类变量
- 无序分类变量

变量数量 x 变量种类：比价复杂了

6.3 各种相关系数

连续 VS 连续：Pearson 相关系数

- Spearman 秩相关系数

有序 VS 有序：Gamma 系数

- 值为-1 到 1，绝对负相关与绝对正相关

无序 VS 无序: 列联系数等

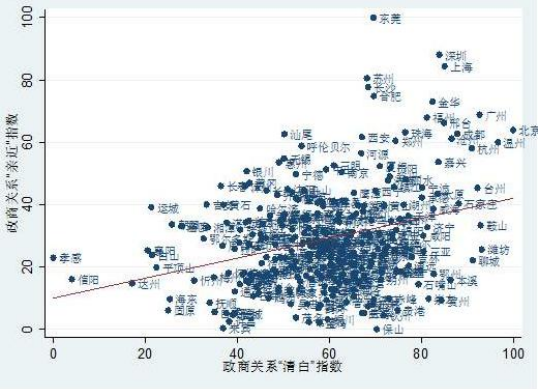
- 基于卡方统计量进一步推导而来
- 无方向(因为是无序变量)，0~1(因卡方值可以无穷大)
- OR/RR： 一类特殊的关联强度指标(在医学方面用的比较广泛)

连续 VS 分类： Eta

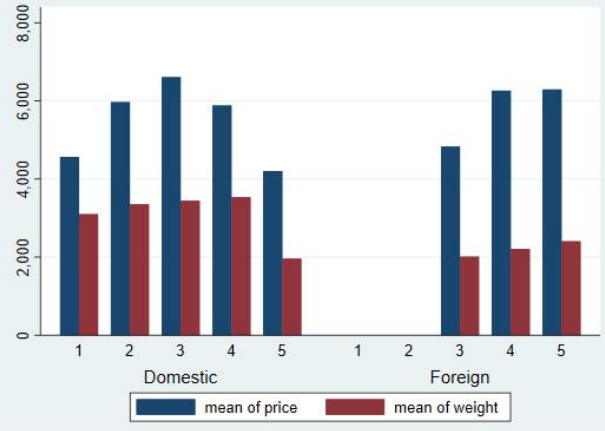
- 实质为方差解释度

6.4 统计图/统计表在相关分析中的重要性

连续变量：用散点图确认关联趋势是否为直线（可以直接计算相应的相关系数）



分类变量：分组条图、马赛克图(分组百分条图)等工具



6.5 两连续变量的相关：常用术语

直线相关：两变量呈线性共同增大，或者呈线性一增一减的情况

曲线相关：两变量存在相关趋势，但是为各种可能的曲线趋势

- 此时如果直接进行直线相关分析，有可能得出无相关性的结论

正相关与负相关：如果 A 变量增加时 B 变量与增加，则为正相关，如果 A 变量增加时 B 变量减少，则为负相关

完全相关：不属于统计学的研究范畴

- 可分为完成正相关和完全负相关两种

- 零相关

6.6 两连续变量的相关：Pearson 相关系数

在双变量正态分布的前提假设下，基于方差、协方差等指标而来

- 方差：每个变量的离散程度(信息量大小)

- 协方差：各变量共同携带的信息量大小(基于 2 个或多个变量进行计算的)

- 协方差 ^{2}/(方差 X*方差 Y):决定系数，两个变量**总信息量**(方差 X*方差 Y)中的**共同部分**（协方差 ^{2}）占比。决定系数=100%，意味这两个变量携带的信息量为共同信息量，没有单独的携带信息，在散点图上就会表现为**完全正相关**或**负相关**。决定系数=0%，意味这两个变量携带的信息量为没有同信息量，在散点图上就会表现的为**完全的没相关**

- sqrt(决定系数)：相关系数 r

相关系数 r 是一个无单位的量值，且-1<r<1

- 其正负反映了相关的方向

- |r|越接近于 1，说明相关性越好

- |r|越接近于 0，说明相关性越差

6.7 Pearson 相关系数的检验

H0：两变量间无直线相关关系，P=0

H1：两变量间有直线相关关系

检验的方法仍然是 t 检验

$T=(r-0)/s_r, v=n-2$ $s_r=\sqrt{(1-r2)/(n-2)}$

6.8 Pearson 相关系数的适用条件

必须是线性相关的情形

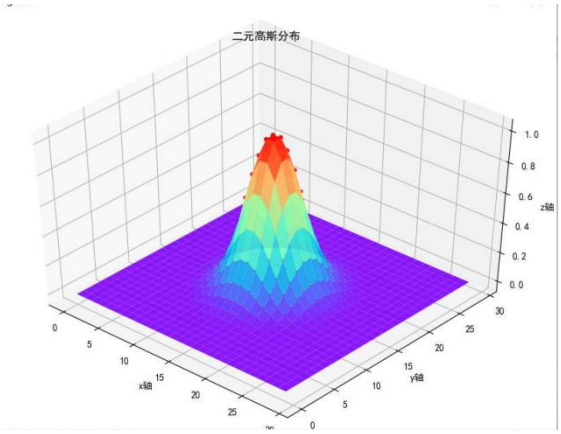
极端值对相关系数的计算影响极大，因此要慎重考虑和处理

要求相应的变量呈双变量正态分布

- 其实基本做不到

- 对此有一定的耐受性

- 只要分布不是特别偏就成



6.9 Spearman 秩相关系数

当相关分析的两变量 X、Y 并不符合适用条件时，r 就不能正确地反映 X、Y 间的相关性

- 方差/协方差能否正确代表数据的离散程度/信息量?

Spearman 提出首先对数据做出秩变换，然后再计算两组秩间的直线相关系数

- 显然，这实际上就是秩变换分析思想的一个具体应用

为了有别于 r 该相关系统用 rs 表示

6.10 Python 相关分析的实现

相关分析作为比较简单的方法，在 statsmodels 中并未作进一步的完善，因此主要使用 scipy 实现

两个连续变量，且符合双变量正态分布：Pearson 相关系数

scipy.stats.pearsonr(a, b)

两个连续变量，不符合双变量正态分布：Spearman 等级相关系数

scipy.stats.spearmanr(a, b)

两分类变量 vs. 连续变量：Point-biserial 相关系数

scipy.stats.pointbiserialr(a, b)

两个有序变量：Kendall's Tau

scipy.stats.kendalltau(a, b, initial_lexsort=None, nan_policy='omit')

```
df.plot.scatter('身高','体重')
# %%
from scipy import stats as ss
ss.pearsonr(df.体重,df.身高)
# %%
ss.spearmanr(df.体重,df.身高)
#%%
df.体重.describe()
# %%
import numpy as np
df['体重 2'] = pd.Series(np.random.randint(50,87,52))
# %%
ss.kendalltau(df.体重,df.体重 2)
```

6.11 相对危险度(RR Relative Risk)

表示两种情况下发病密度或发病概率之比

- Pt：实验组人群反应阳性概率(实验组发病人数/实验组的总人数)

- Pc：对照组人群反应阳性概率(对照组发病人数/对照组的总人数)

- RR= Pt/Pc

如果 RR>1，说明相应的自变量取值增加，会导致个体的发病/死亡风险增加若干倍

例：吸烟者的发病概率是非吸烟者的 5 倍

RR 在医学中得到了极为广泛的应用

6.12 优势比 OR(RR Odds Ratio)

是否暴露于某因素	病例组	对照组	合计
是	a	b	a+b
否	c	d	c+d
合计	a+c	b+d	N

$$OR = \frac{a/c}{b/d} = \frac{ad}{bc}$$

RR 的计算要求得到各组的**反应概率**,这在回顾性研究中很难满足(无法得到良性概率或者反映概率的准确估计值)

- 此时研究者往往使用 OR 值代替 RR 值

OR：为下列两种比例之比

- 反应阳性人群中实验因素有无的比例 a/b

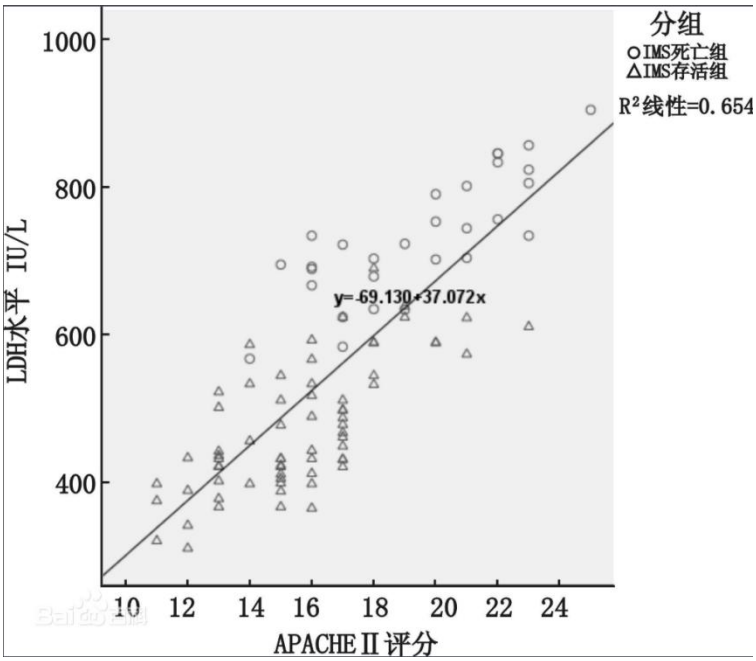
- 反应阴性人群中实验因素有无的比例之比 c/d

- OR = (a/b)/(c/d)

例：该疾病病例中吸烟/非吸烟者的比例是非病例中吸烟/非吸烟者比例的 3 倍

OR 可以间接反映**关联强度**，但是**理解**上比较困难

发病概率较低时，OR 往往近似的在按照 RR 的含义进行解释和使用



6.13 Python 实现 OR 和 RR

6.13.1 scipy 的实现方式

scipy.stats.fisher_exact()中可以计算 OR 值，相应的检验 P 值则是确切概率法的 P 值

```
OR, P = ss.fisher_exact(pd.crosstab(df.性别, df.开设))
OR
```

6.13.2 statsmodels 的实现方式

statsmodels.stats.contingency_tables.Table 类可以直接提供分块 2X2 表 OR 的估计值

statsmodels.stats.contingency_tables.Table2x2 类可以直接提供 OR、RR 的估计和检验结果

class statsmodels.stats.contingency_tables.Table2x2(

table

shift_zeros = True

)

Table2x2 类的属性

log_oddsratio / log_oddsratio_se lnOR / lnOR 的标准误

oddsratio OR 值

riskratio RR 值

log_riskratio / log_riskratio_se lnRR / lnRR 的标准误

Table2x2 类的方法

* 注意：有些方法尚未开发完成

summary([alpha, float_format, method]) 汇总输出 OR、RR 的估计和检验结果

symmetry([method]) 对称性检验

test_nominal_association() 关联性的检验(有序)

test_ordinal_association([row_scores, ...]) 关联性的检验(无序)

log_oddsratio_confint([alpha, method]) 可信区间

log_oddsratio_pvalue([null]) P-value

log_oddsratio_se() 标准误

log_riskratio()

log_riskratio_confint([alpha, method]) CI

log_riskratio_pvalue([null]) p-value

log_riskratio_se()

oddsratio()

oddsratio_confint([alpha, method]) CI

oddsratio_pvalue([null]) P-value

riskratio()

riskratio_confint([alpha, method]) CI

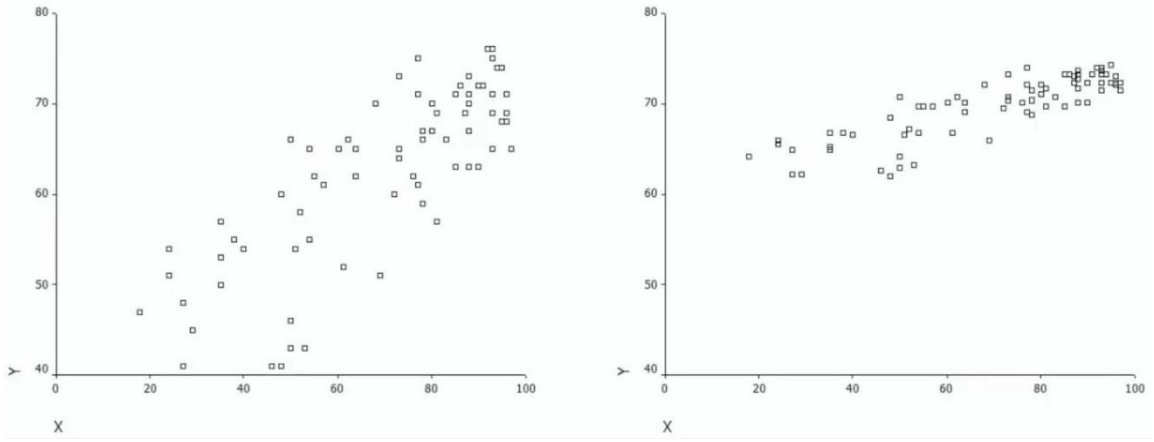
riskratio_pvalue([null]) p-value

```
import numpy as np
import statsmodels.stats.contingency_tables as tbl
# 这里必须使用 np.asarray 函数进行转换，否则后续计算可能报错
table = tbl.Table2x2(np.asarray(pd.crosstab(df.性别, df.开设)))
table.oddsratio
table.summary()
```

7 线性回归模型入门

7.1 相关与回归的区别和联系

相关与回归都是统计中很常用的方法，相关往往来用筛选候选变量，回归往往用来做后期的精细建模。



7.1.1 回归分析与相关分析的区别

相关分析所研究的两个变量是对等关系，回归分析所研究的两个变量不是对等关系，必须根据研究目的确定其中的自变量、因变量。

对于变量 x 与 y 来说，相关分析只能计算出一个反映两个变量间相关密切程度的相关系数，计算中改变 x 和 y 的地位不影响相关系数的数值。回归分析有时可以根据研究目的不同分别建立两个不同的回归方程。

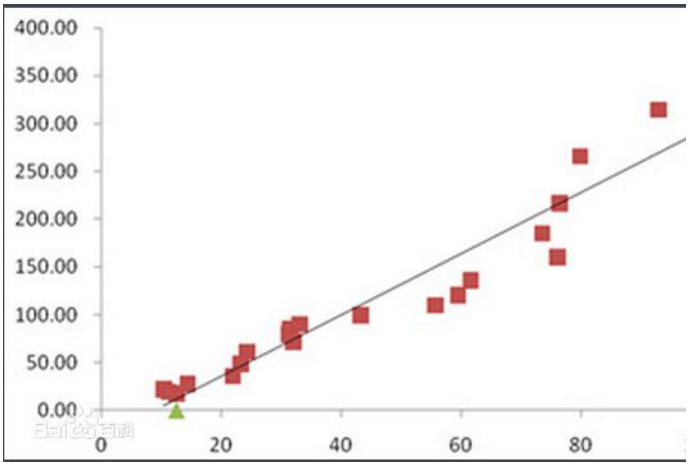
相关分析对资料的要求是，两个变量都是随机的，也可以是一个变量是随机的，另一个变量是非随机的。而回归分析对资料的要求是，自变量是可以控制的变量（给定的变量），因变量是随机变量。

7.1.2 回归分析与相关分析的联系

相关分析是回归分析的基础和前提。假若对所研究的客观现象不进行相关分析，直接作回归分析，则这样建立的回归方程往往没有实际意义。只有通过相关分析，确定客观现象之间确实存在数量上的依存关系，而且其关系值又不确定的条件下，再进行回归分析，在此基础上建立回归方程才有实际意义。

回归分析是相关分析的深入和继续。对所研究现象只作相关分析，仅说明现象之间具有密切的相关关系是不够的，统计上研究现象之间具有相关关系的目的，就是要通过回归分析，将具有依存关系的变量间的不确定的数量关系加以确定，然后由已知自变量值推算未知因变量的值，只有这样，相关分析才具有实际意义。

7.2 线性回归分析概述



研究一个连续性变量(因变量)的取值随着其它变量(自变量)的数值变化而变化的趋势

通过回归方程解释两变量之间的关系显的更为**精确**，可以计算出自变量改变一个单位时因变量**平均**改变的单位数量，这是相关分析无法做到的(只能说明关系的强弱，数量是无法标识的)

除了描述两变量的关系以外，通过**回归方程**还可以进行**预测**和**控制**，这是实际工作中尤为重要

- 预测：当自变量在某一个排列组合下，因变量大概应该是什么情况(天气预报：气压，气象指标等到下雨性的可能性有多大)

- 控制：出于各种目的，需要某个因变量不超过某个数值(不小于某个数值)，这样的情况如何对自变量进行控制(奥运开会，控制天气。建立一个方程：根据每天车辆的上路情况，工业排放的情况等，建立方程，根据方程推算下某个指标能应该在某个范围)

7.3 线性回归模型的基本框架

线性回归假定自变量对因变量的影响强度始终保持不变

$Y= a+bx$

Y：给定自变量的取值时，y 的估计值(所估计的平均水平)

- 因变量的预测值可以被分解成两部分：常量+回归部分

a: 常量，自变量 x 取值均为零时 y 的平均估计量

- 可以被看成是一个基线水平

- 多数情况下不没有实际意义，研究者也不关心

b: (偏)回归系数，自变量 x 改变一个单位，y 估计值的改变量

- 因变量 y 的变异中，可以由 x 直接估计的部分

$y_i = a+bx+e_i \quad e_i \sim N(0,\sigma^2)$

估计值和每一个实测值之间的差被称为残差。它刻画了因变量 y 除了自变量 x 以外的其它所有未进入该模型，或未知但可能与 y 有关的随机和非随机因素共同引起的变异，即不能由 x 直接估计的部分

为了方程可以得到估计，往往假定 e_i 服从正态分布 $N(0,\sigma^2)$

作为整体，模型可以对残差的离散程度进行估计

7.4 线性回归模型中的常用指标

决定系数

- 模型整体价值的衡量指标(这个模型是对因变量做预测，那么究竟模型能解释因变量多大程度上的变异)

- 相应的相关系数的平方，用 R^2 表示

- 反映因变量 y 的全部变异中能够通过回归关系被自变量解释的比例(知道了自变量的范围，并控制后，因变量的变异可以减少多少)

偏回归系数

- 反映某一个自变量在数量上对因变量的影响程度

- 相应的自己变量上升一个单位时，因变量取值的变动情况

标化偏回归系数：量纲问题

- 用于自变量间重要性的比较

7.5 线性回归模型中的适用条件

线性趋势：如果不是，则不能采用线性回归来分析

- 可能通过散点图来加以判断

独立性：实际上就是要求残差间相互独立，不存在自相关，否则应当采用自回归模型来分析

- 数据背景考察，DW 检验，自相关/偏相关分析

正态性：就自变量的任何一个线性组合，因变量 y 均服从正态分布

- 实际上是要求残差 e_i 服从正态分布

方差齐性：就自变量的任何一个线性组合，因变量 y 的方差均相同

- 实质就是要求残差的方差齐

7.5.1 适用条件：变量分布

因变量

- 无法直接获取正态性、方差齐性等信息，但可以做初步观察

- 一般而言，如果因变量分布极端偏态，则多半残差分布会有问题

自变量

- 均为可被精确测量的确定值(只能是连续的变量)

自变量分布特征极端偏态时

- 直接建模往往实用性不佳，因此可以考虑变量变换

- 但是变量变换又会导致模型假定的线性关联发生改变，因此不能一刀切

7.5.2 适用条件：样本量

可以对模型所需样本量做精确计算，但这种方法在筛选多个候选自变量的研究背景下基本没有实用价值

根据经验，案例数应当在希望分析的自变量数的 20 倍以上为宜

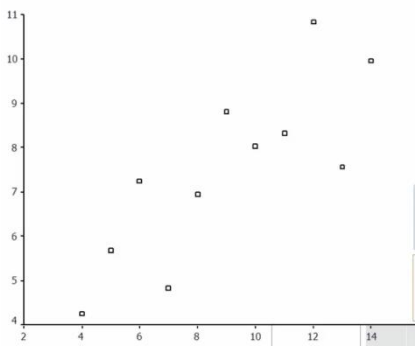
- 实质上样本量和模型的决定系数有关(关联系数越小，样本就越大)，可通过迭代的方法进行计算

样本量不足不意味不能拟合模型，只是参数估计值可能不稳定，或者不能得到应当本来有统计学意义的检验结果

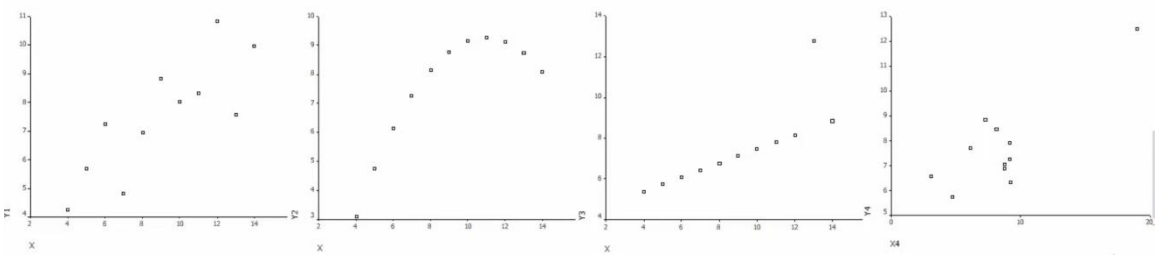
7.6 线性回归模型的建模步骤

做出散点图，观察变量间的趋势

散点图考察的三个信息层次



只有确认为线性趋势的候选自变量才能直接纳入后续分析步骤



考察数据的分布，进行必要的预处理

- 观察自变量取值是否过于极端

- 初步观察因变量的正态性

- 初步观察可能的方差不齐迹象等问题

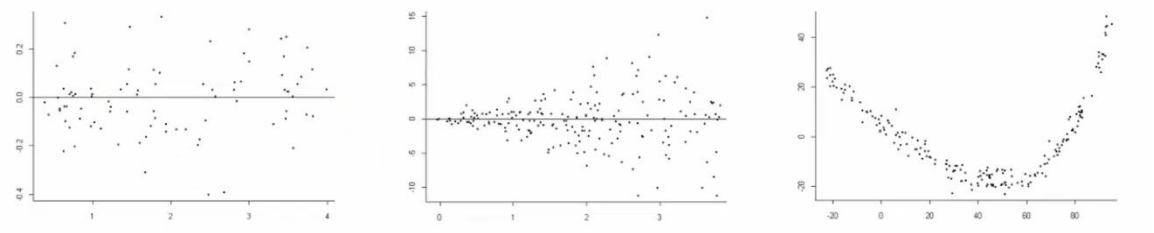
进行候选自变量筛选，初步建立直线回归模型

- 科研领域的分析，尽量采用手工筛选

残差分析

- 残差间是否独立(Durbin-Watson 检验)

- 残差分布是否为正态(图形或统计量)



残差分析

强影响点的诊断

多重共线性问题的判断

这三个步骤往往混在一起，难以完全分布先后

7.7 线性回归模型的 scipy 实现

scipy.stats.linregress(

x, y：类数组格式的自变量、因变量，均为一维，也可以直接以 k*2 的二维数组格式提供

注意：该命令的参数格式是自变量 x 在前！

)

返回结果：

slope：回归系数 b

intercept： 常数项 a

r-value： 两个变量的相关系数

p-value： 回归系数的双侧检验 （相关系数的 P 值）

stderr： 回归系数的标准误

```
from scipy import stats as ss
ss.linregress(df.身高,df.体重)
ss.linregress(df.loc[:,['身高','体重']])
```

7.8 statsmodels 的实现方式

class statsmodels.regression.linear_model.OLS(

endog： 因变量，1 维数组格式

exog = None : n*k 格式数组，k 代表自变量数量

missing = 'none'： 对缺失值的处理方式

'none'： 不做任何检查

'drop'： 发现缺失值后该案例直接删除

'raise'： 检查并抛出错误

hasconst = None : T/F，是否允许用户自定义的常数被纳入方程，模型不默认常数项

)

无缺失值时拟合单自变量模型

```
df2 = df.loc[:,['身高']]
df2['height'] =100
df2

# %%
```

```
from statsmodels.regression.linear_model import OLS
reg = OLS(df.体重,df2[['height','身高']]).fit()
reg.summary()
```

多个自变量

```
df3 = df.loc[:,['身高','性别']]
df3['height'] = 1
df3
df3.replace(['男','女'],[1,2],inplace = True)
# %%
reg2 = OLS(df.体重,df3).fit()
reg2.summary()
```

检验残差

```
td = pd.DataFrame({
    'fit':reg2.fittedvalues, #  每个案例的预测值
    'resid':reg2.resid, #  预测值与实测值之差(残差)
    'zresid':reg2.resid_pearson #标准的残差
})
# %%
td.resid.plot.hist()
td.zresid.plot.hist()
# %%
td.plot.scatter('fit','resid')
td.plot.scatter('fit','zresid')
```

8 样本量的计算

8.1 为什么要做样本量估计

太小：无法检验出预期存在的差异

- 某 2 年，15 种临床医学杂志，450 个两样本率比较，设差异为 0.5 倍，只有 19%(85 个)的 Power 在 0.8 以上

太大：造成无谓的浪费

- 混杂因素可能更多
- 管理实施上更加困难
- 结果可能无专业意义
- 只要样本足够大，必然会有统计学差异

8.2 样本量估计的应用场景

设计阶段

- 按给定条件计算需要的样本量，以严格控制成本，提高产出效率(做什么样的研究，期望什么样的差异被发现，需要控制的干扰因素等)

分析阶段

- 对已进行的实验计算检验效能

用的最多就是临床实验，因为样本非常非常贵，一个样本就有可能上百万的投入，不要只看吃的药，实际管理成本和人力成本，还要前期的研发成本，加起来是很吓人的。一个药几百万美元是很正常的。因此在临床领域应用还是比较广泛的

8.3 计算样本量时所需要的考虑的问题

主要目的与主要指标

- 当存在多个指标时，以最主要指标为准进行计算

设计/分析方法

- 好的分析方法权节约样本量

- 但复杂的设计会给样本量计算带来困难(对样本要求高，不能样本的丢失，样本的计算比较难，不如普通的成组，配对样本好算)

第一类错误的大小(控制 0.05)

单、双侧问题(专业没把握双侧)

基线水平(如果做两样本比较会有对照组的问题，或单本样会有个和基线水平的比较)，基线水平就是用来对照的平均水平(均数)什么样，百分比什么样

- 一般不直接参与计算，但是有助于制定期望检出的差值

期望检出的差值

- 一般为专业上认为有意义的差值

- 差值如果太大，有可能达不到标准

- 差值如果太小，有可能计算出过多的样本量，从而导致信息的浪费

资料的离散程度(正常情况时资料的方差，标准差是多少？这个是最重要的，也是最难算的)

- 预试验(I 期临床试验，前期的小样本研究)

- 查历史资料

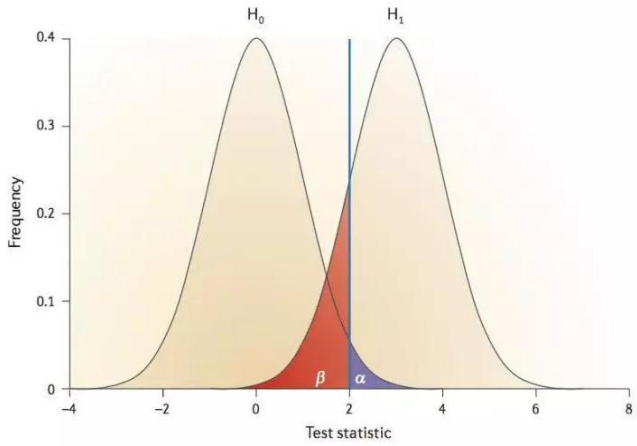
- 均数的 1/2 ~ 1/4（相应的变量不能小于 0,顶多等于 0）

- 分类资料不用另行查找(用二项分布的均数是用 n 和 PI 记性计算的)

检验效能的大小

- 不应当低于 0.75，一般都在 0.8 以上

8.4 样本量计算的基本原理



直接从假设检验的 I、II 类错误问题发展而来

原假设成立时，如果超过某个 C 点，则原假设被拒绝，即 I 类错误

在备择假设成立的情况下，如果低于某个 C 点，则原假设不能被拒绝，备择假设不能被接受，即 II 类错误

两个公式中的 C 点应当相同，联立求解，得到相应的 N

8.5 单组/两组均值比较的样本量计算

单样本/配对样本

$$N=\left[\frac{(u_\alpha+u_\beta)\sigma_d}{\delta}\right]^2 \qquad u_\beta=\delta\frac{\sqrt{N}}{\sigma_d}-u_\alpha$$

成组两样本

$$N=\left[\frac{2(u_\alpha+u_\beta)\sigma}{\delta}\right]^2 \qquad u_\beta=\frac{\delta\sqrt{N}}{2\sigma}-u_\alpha$$

公式中的δ/σ也被称为 effect size,即希望能够检验出的标准化的相对差异

实际使用中需要加上 10%~20%的例数，以补充可能的损失

8.6 样本量估计的 Python 实现

8.6.1 单样本/配对样本 t 检验

class statsmodels.stats.power.TTestPower()

TTestPower 类的方法：

power(# 计算相应检验的效能

effect_size：标准化（差值/标准差）之后的希望检验出的差距，必须为正数

nobs：设定的样本量

alpha : alpha 水准

df = None：自由度，其实可以不设

alternative = 'two-sided' : {'two-sided, 'larger', 'smaller'}

) # 返回：检验效能值

solve_power(# 基于检验效能反推任何一个参数

effect_size = None

nobs = None

alpha = None

power = None

alternative = 'two-sided'

) # 返回：缺失的相应参数

单样本 t 检验实例

举例：

研究新药降低高血脂患者的胆固醇的效果，研究者规定试验组的血清胆固醇平均降低 0.5mmol/L 以上， 才有进一步研究的价值。引用文献中胆固醇的标准差为 0.8mmol/L，规定单侧α = 0.05，power = 0.90，要求估计样本含量

```
from statsmodels.stats import power as sp
sp1 = sp.TTestPower()
```

```
# 计算样本量
sp1.solve_power(effect_size = 0.5/0.8, nobs = None, alpha = 0.05,
power = 0.9, alternative='larger')
```

```
# 计算检验效能
sp1.power(effect_size = 0.5/0.8, nobs = 24, alpha=0.05, alternative='larger')
```

8.6.2 两样本 t 检验

class statsmodels.stats.power.TTestIndPower()

TTestIndPower 类的方法：

power(# 计算相应检验的效能

effect_size：标准化（差值/标准差）之后的希望检验出的差距，必须为正数

nobs1：样本量较小的组 1 的样本量

alpha : alpha 水准

ratio = 1 ：两组样本量之比，组 2 = 组 1 * ratio

df = None：自由度，其实可以不设

alternative = 'two-sided' : {'two-sided, 'larger', 'smaller'}

) # 返回：检验效能值

solve_power(# 基于检验效能反推任何一个参数

effect_size = None

nobs1 = None

alpha = None

power = None

```
    ratio = 1
    alternative = 'two-sided'
) # 返回：缺失的相应参数
```

两样本 t 检验实例

案例：

研究用新药降低高血脂患者胆固醇的效果，研究者规定试验组与对照组（安慰剂）相比，血清胆固醇平 均降低 0.5mmol/L 以上，才有推广价值。引用文献中胆固醇的标准差为 0.8mmol/L，规定单侧α = 0. 05， power = 0.90，要求估计样本含量。

```
# 计算样本量
sp2 = sp.TTestIndPower()
sp2.solve_power(effect_size = 0.5/0.8, nobs1 = None, alpha = 0.05,
power = 0.9, ratio = 1.0, alternative = 'larger')
```

```
# 计算可检验的效应大小
sp2 = sp.TTestIndPower()
sp2.solve_power(effect_size = None, nobs1 = 44, alpha = 0.05,
power = 0.9, ratio = 1.0, alternative='larger')
```

8.7 多组均值比较的样本量计算

实用：根据最重要的两样本组数据的特征进行样本总量外推

严格：基于组间变异进行计算

- 难点在于 effect size 如何进行估计

$$f = \sqrt{\frac{\sum_{i=1}^{k} p_i * (\mu_i - \mu)^2}{\sigma^2}}$$

where $p_i = n_i / N$,
n_i = number of observations in group i
N = total number of observations
μ_i = mean of group i
μ = grand mean
σ^2 = error variance within groups

- Cohen 建议将 f = 0.1, 0.25, 0.4 分别作为小、中、大的效应值用于估计

8.7.1 单因素方差分析

class statsmodels.stats.power.FTestAnovaPower()

FTestAnovaPower 类的方法：

power(# 计算相应检验的效能

effect_size：标准化（差值/标准差）之后的希望检验出的差距，必须为正数

nobs：每组样本量

alpha : alpha 水准

k_groups = 2 ：用于比较的样本组数

) # 返回：检验效能值

solve_power(# 基于检验效能反推任何一个参数

effect_size = None

nobs = None

alpha = None

power = None

k_groups = 2 ：用于比较的样本组数

) # 返回：缺失的相应参数

```
# 计算样本量
sp3 = sp.FTestAnovaPower()
sp3.solve_power(effect_size = 0.1, nobs = None, alpha = 0.05,
power = 0.9, k_groups = 3)
```

```
# 计算样本量
sp3 = sp.FTestAnovaPower()
```

```
sp3.solve_power(effect_size = 0.25, nobs = None, alpha = 0.05,
power = 0.9, k_groups = 3)
```

```
# 计算样本量
sp3 = sp.FTestAnovaPower()
sp3.solve_power(effect_size = 0.4, nobs = None, alpha = 0.05,
power = 0.9, k_groups = 3)
```

8.8 率的比较的样本量估计

分类变量不需要另行估计变异，因为可以从“均数”计算得出

计算原理上和均数比较完全相同，但是难点在于如何给出 effect size 的合理估计值

方法 1：effect size = 2*(arcsin(sqrt(p1))-arcsin(sqrt(p2)))

- Cohen 建议将 h=0.2, 0.5, 0.8 分别作为小、中、大的效应值用于估计

方法 2：正态近似估计

$$N=\frac{\left(u_{\alpha}+u_{\beta}\right)^{2}4\pi_{c}\left(1-\pi_{c}\right)}{\left(\pi_{1}-\pi_{2}\right)^{2}} \qquad u_{\beta}=\frac{\sqrt{N}\left|\pi_{1}-\pi_{2}\right|}{2\sqrt{\pi_{c}\left(1-\overline{\pi}\right)}}-u_{\alpha}$$

8.8.1 单样本/配对率的比较

常规条件下某动物模型出现阳性结局的概率为 40%，某研究人员考虑采用另一种方法进行试验，使用 60 个动物进行该研究，预期的成功概率为 50%，请估计该检验的效能是否充足

```
p1 = 0.5; p2 = 0.4
h = 2*(np.arcsin(np.sqrt(p1))-np.arcsin(np.sqrt(p2)))
```

```
# 使用单样本 t 检验框架进行计算
sp1.solve_power(effect_size = h, nobs = 60, alpha = 0.05, power = None)
```

8.8.2 两样本率的比较

原方法下某动物模型出现阳性结局的概率为 15%，现考虑采用改进的新方法进行比较，预期新方法阳性概率为 30%，请估计该研究所需的动物样本量。

```
p1 = 0.3; p2 = 0.15
h = 2*(np.arcsin(np.sqrt(p1))-np.arcsin(np.sqrt(p2)))
```

```
# 使用两样本 t 检验框架进行计算
sp2.solve_power(effect_size = h, nobs1 = None, alpha = 0.05, power = 0.8, alternative = 'larger')
```