In [ ]:

```
# 取出所需数据，便于后续操作
dfrank = ccss.loc[ccss.s0 == '北京', ['time', 'Qa3']]
dfrank.head()
```

In [ ]:

```
dfrank['qa3r'] = ss.rankdata(dfrank.Qa3)
dfrank.head()
```

In [ ]:

```
ss.f_oneway(dfrank[dfrank.time == 200704].qa3r,
            dfrank[dfrank.time == 200712].qa3r,
            dfrank[dfrank.time == 200812].qa3r,
            dfrank[dfrank.time == 200912].qa3r
            )
```

In [ ]:

```
# 尝试使用其他编秩方法
dfrank['qa3r2'] = ss.rankdata(dfrank.Qa3, method = 'dense')
ss.f_oneway(dfrank[dfrank.time == 200704].qa3r2,
            dfrank[dfrank.time == 200712].qa3r2,
            dfrank[dfrank.time == 200812].qa3r2,
            dfrank[dfrank.time == 200912].qa3r2
            )
```

In [ ]:

```
# 进行两两比较
import scikit_posthocs as sp

sp.posthoc_conover(dfrank, val_col='qa3r', group_col='time',
                   p_adjust = 'bonferroni')
```

5.5 实战练习

对CCSS数据中不同城市受访者的年龄进行比较，尝试使用变量变换、秩和检验、秩变换分析等方法完成该任务，比较相应的分析结果，并思考各种方法的优缺点。

6 卡方检验

6.1 卡方检验的基本原理

6.2 行*列表的卡方检验

例6.1

在CCSS的分析报告中，所有受访家庭会按照家庭年收入被分为低收入家庭和中高收入家庭两类，现希望考察不同收入级别的家庭其轿车拥有率是否相同。

6.2.1 scipy的实现方式

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
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

In [ ]:

```
tbl = pd.crosstab(ccss.Ts9, ccss.O1)
tbl
```

In [ ]:

```
ss.contingency.expected_freq(tbl)
```

In [ ]:

```
ss.contingency.margins(tbl)
```

In [ ]:

```
ss.chi2_contingency(tbl, False)
```

6.2.2 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
fittedvalues 期望频数

resid_pearson Pearson残差
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
test_ordinal_association([row_scores, ...]) 有序分类行/列变量独立性检验
Cochran-Armitage趋势检验

In [ ]:

```
import statsmodels.stats.contingency_tables as tbl

table = tbl.Table(pd.crosstab(ccss.Ts9, ccss.O1))
table
```

In [ ]:

```
print(table.table_orig)
print(table.fittedvalues)
print(table.resid_pearson)
print(table.chi2_contribs)
```

In [ ]:

```
table.independence_probabilities
```

In [ ]:

```
table.marginal_probabilities
```

test_nominal_association()的输出：

statistic : 卡方统计量
df : 自由度
pvalue : P值

In [ ]:

```
table.test_nominal_association()
```

In [ ]:

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

6.2.3 事后两两比较

例6.2

> 不同城市的汽车牌照政策、生活习惯等并不相同，现希望考察北、上、广三地的家庭轿车拥有率是否相同。

和SPSS等专业统计软件不同，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

In [ ]:

```
import statsmodels.stats.contingency_tables as tbl2

table2 = tbl2.Table(pd.crosstab(ccss.s0, ccss.O1))
table2.table_orig
```

In [ ]:

```
res = table2.test_nominal_association()
print(res.statistic, res.df, res.pvalue)
```

In [ ]:

```
table2.local_oddsratios
```

In [ ]:

```
table2.cumulative_oddsratios
```

6.3 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

In [ ]:

```
tbl = pd.crosstab(ccss.Ts9, ccss.O1)
ss.chi2_contingency(tbl) # 默认直接进行Yates校正
```

In [ ]:

```
# 确切概率结果
ss.fisher_exact(tbl)
```

In [ ]:

```
# 超过2*2表格时会报错
ss.fisher_exact(pd.crosstab(ccss.s0, ccss.O1))
```

6.4 配对卡方检验

6.4.1 基本原理

例6.3

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？

6.4.2 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）
tbl.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

In [ ]:

```
import statsmodels.stats.contingency_tables as tbl

table = tbl.mcnemar(pd.DataFrame([[56, 35], [21, 28]]))
```

In [ ]:

```
table.pvalue
```

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

In [ ]:

```
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
```

In [ ]:

```
table.summary()
```

In [ ]:

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

6.5 分层卡方检验

例6.4

在分析中进一步控制城市的影响，在此前提下得到更准确的家庭收入分级和轿车拥有情况的关联程度测量指标。

class statsmodels.stats.contingency_tables.StratifiedTable(

tables : 多个2x2表构成的list，或者2x2xk结构的ndarray
shift_zeros = False

)

StratifiedTable类的属性

logodds_pooled / log_oddsratio_se 分层lnOR的M-H估计值 / SE
oddsratio_pooled 分层OR的Mantel-Haenszel估计值
risk_pooled 分层RR的估计值

StratifiedTable类的方法

注意：有些功能尚未完成

```
summary([alpha, float_format, method])     分层卡方结果的汇总输出

logodds_pooled()
logodds_pooled_confint([alpha, method])    CI
logodds_pooled_se()

oddsratio_pooled()
oddsratio_pooled_confint([alpha, method])    CI

risk_pooled()

test_equal_odds([adjust])    Test that all odds ratios are identical
test_null_odds([correction])    Test that all tables have OR = 1
```

In [ ]:

```
rawtbl = pd.crosstab([ccss.s0, ccss.Ts9], ccss.O1)
rawtbl
```

In [ ]:

```
pd.crosstab([ccss.s0, ccss.Ts9], ccss.O1, normalize = 0)
```

In [ ]:

```
import statsmodels.stats.contingency_tables as tbl

# pandas无法直接生成所需的2x2xk结构
table = tbl.StratifiedTable([rawtbl[:2],
                             rawtbl[2:4],
                             rawtbl[4:]])
table
```

In [ ]:

```
table.summary()
```

6.6 二项分布检验与近似Z检验

6.6.1 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, inversion of binom_test
```

) # 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
```

)

In [ ]:

```
from statsmodels.stats import proportion as sp

sp.proportion_confint(5, 10)
```

In [ ]:

```
sp.proportion_confint(5, 10, method = 'binom_test')
```

In [ ]:

```
sp.multinomial_proportions_confint([5, 5])
```

6.6.2 二项分布检验

statsmodels.stats.proportion.binom_test(# 基于二项分布的确切检验

```
    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
    value = None
```

) # 输出：卡方值、P值、（表格，期望值）

In [ ]:

```
sp.binom_test(1, 600)
```

In [ ]:

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

6.6.3 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

In [ ]:

```
sp.proportions_ztest(30, 100, 0.2)
```

In [ ]:

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

6.7 实战练习

计算北京、上海、广州三地的汽车拥有率可信区间。

考察不同收入级别的受访者其职业分布有无差异。 提示：需要考虑两两比较。

在上面分析的基础上，在控制城市的影响之后，考察不同收入级别的受访者其职业分布有无差异。

7 相关分析

7.1 相关分析的指标体系

7.2 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)
```