In [ ]:

```
pd.crosstab(ccss.s2, ccss.s0)
```

In [ ]:

```
pd.crosstab(ccss.s2, ccss.s0, normalize = 0, margins = True)
```

In [ ]:

```
pd.crosstab([ccss.s2, ccss.O1], ccss.s0)
```

In [ ]:

```
pd.crosstab([ccss.s2, ccss.O1], [ccss.s0, ccss.s5])
```

In [ ]:

```
pd.crosstab(ccss.s2, ccss.s0).plot.bar()
```

In [ ]:

```
pd.crosstab(ccss.s2, ccss.s0).plot.bar(stacked = True)
```

In [ ]:

```
pd.crosstab(ccss.s2, ccss.s0, normalize = 0).plot.bar(stacked = True)
```

1.4 实战练习

请就CCSS_Sample数据，分析受访者的总指数、现状指数和预期指数分城市、月份的分布情况，包括集中趋势和离散趋势的分布变化情况。

请就CCSS_Sample数据，对性别、城市、职业等分类变量尝试进行交叉描述。

2 均数间的比较

2.1 假设检验的基本原理

2.2 单样本t检验

2.2.1 基本原理与适用条件

例2.1

CCSS项目基期的信心指数值被设定为100，但这是全部城市的平均水平，请考察基期时广州信心指数均值是否和基准值有差异。

2.2.2 scipy的实现方式

scipy.stat包中可以实现各种常用的假设检验方法，但并未配备详细选项，例如不能指定检验的单双侧。

scipy.stats.ttest_1samp(

```
    a : 类list格式的样本数值
    popmean : H0所对应的总体均数
```

)

In [ ]:

```
ccss.query("s0 == '广州' & time == 200704").index1.describe()
```

In [ ]:

```
ccss.query("s0 == '广州' & time == 200704").index1.hist()
```

In [ ]:

```
from scipy import stats as ss

ss.ttest_1samp(ccss.query("s0 == '广州' & time == 200704").index1, 100)
```

2.2.3 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

In [ ]:

```
from statsmodels.stats import weightstats as ws

des = ws.DescrStatsW(ccss.query("s0 == '广州' & time == 200704").index1)
des.mean
```

In [ ]:

```
# 计算均数的95% CI
des.tconfint_mean()
```

In [ ]:

```
# 进行单样本t检验
des.ttest_mean(100)
```

In [ ]:

```
des.ttest_mean(100, 'smaller')
```

In [ ]:

```
des.ttest_mean(100, 'larger')
```

2.3 两样本t检验

2.3.1 基本原理与适用条件

例2.2

图形化分析中研究者已经发现不同婚姻状况的信心指数均值可能存在差异，现希望进一步用假设检验对此差异进行确认。
变量S7婚姻为三分类，但是离异/分居/丧偶这一类别的样本只有14例，因此只考虑对已婚和未婚的人群进行比较。

2.3.2 scipy的实现方式

scipy最大的问题在于使用的数据格式和标准统计格式不同

统计软件使用的标准格式：连续变量、分组变量
python使用的格式：组1测量值列表、组2测量值列表

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

In [ ]:

```
# 分布的对称性考察
ccss.index1.plot.hist()
```

In [ ]:

```
# 分组描述
ccss.groupby('s7').index1.describe()
```

In [ ]:

```
# 方差齐性检验
ss.levene(ccss.index1[ccss.s7 == '未婚'], ccss.index1[ccss.s7 == '已婚'])
```

In [ ]:

```
ss.ttest_ind(ccss.index1[ccss.s7 == '未婚'], ccss.index1[ccss.s7 == '已婚'])
```

In [ ]:

```
ss.ttest_ind(ccss.index1[ccss.s7 == '未婚'], ccss.index1[ccss.s7 == '已婚'],
             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
```

)

In [ ]:

```
# 通过基本统计量来做独立两样本检验
ss.ttest_ind_from_stats(95.033106, 21.282487, 790.0,
                        98.282359, 19.959824, 343.0)
```

2.3.3 statsmodels的实现方式

statsmodels中可以实现t检验的所有功能，但是无法完成方差齐性检验，不知道以后是否会加进来

class statsmodels.stats.weightstats.CompareMeans(d1, d2)

```
    d1, d2均为DescrStatsW对象
    如果只有d1为DescrStatsW对象，也可以使用d1.get_compare(other)直接转换
```

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
```

)

In [ ]:

```
d1 = ws.DescrStatsW(ccss.index1[ccss.s7 == '未婚'])
d2 = ws.DescrStatsW(ccss.index1[ccss.s7 == '已婚'])

comp = ws.CompareMeans(d1, d2)

comp.ttest_ind()
```

In [ ]:

```
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
```

In [ ]:

```
comp.ttost_ind(0, 3)
```

In [ ]:

```
comp.tconfint_diff()
```

In [ ]:

```
comp.ztest_ind()
```

2.4 配对样本t检验

2.4.1 基本原理与适用条件

例2.3

为保证数据质量，接受过CCSS访问的受访家庭半年内不会再进行访问，但半年之后会进行抽样回访。在2007年12月，项目组对2007年4月的成功访问家庭进行了回访，共采集了88例有效样本，现希望比较这些样本的信心值是否发生变化，数据见表单CCSS_pair。

In [ ]:

```
ccss_p = pd.read_excel("CCSS_sample.xlsx", sheet_name = 'CCSS_pair')
ccss_p.head()
```

2.4.2 scipy的实现方式

scipy.stats.ttest_rel(

```
    a, b : array_like
    nan_policy : {'propagate', 'raise', 'omit'}
```

)

In [ ]:

```
ccss_p.loc[:, ['index1', 'index1n']].describe()
```

In [ ]:

```
# 用相关分析确认配对信息是否的确存在
ss.pearsonr(ccss_p.index1, ccss_p.index1n)
```

In [ ]:

```
ss.ttest_rel(ccss_p.index1, ccss_p.index1n)
```

In [ ]:

```
# 直接求出差值并进行单样本t检验
ss.ttest_1samp(ccss_p.index1 - ccss_p.index1n, 0)
```

2.4.3 statsmodels的实现方式

statsmodels没有提供直接实现配对t检验的方法，但是可以有两个变通的实现方式

```
    statsmodels.stats.ttost.paired : 提供两个界值点的单侧配对t检验结果
    求出差值，然后使用DescrStatsW.ttest_mean()得到所需检验结果
```

In [ ]:

```
des = ws.DescrStatsW(ccss_p.index1 - ccss_p.index1n)
des.ttest_mean()
```

2.5 实战练习

请考察北京、上海两地在2007年4月时的信心值是否有偏离基准值100。

请分北京、上海、广州三个城市来比较已婚人群和未婚人群的总指数、现状指数和预期指数是否有差异。

请自行完成CCSS_pair数据中针对现状指数和预期指数变化情况的检验。

3 检验方法适用条件的考察

3.1 独立性的考察与应对策略