学习python的数据分析功能时需要注意的几大问题

- python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、组2测量值、组3测量值...

	测量值	组别
1	12	1
2	24	1
3	16	1
4	23	1
5	45	2
6	34	2
7	26	2
8	37	2
9	25	2

	组1测量值	组2测量值
1	12	45
2	24	34
3	16	26
4	23	37
5	.	25

- 统计分析的功能实现不完整

  如statsmodels中的很多功能还在开发中，很多统计方法的实现不完整/有bug
  已有功能和官方文档不完全对应，经常出现有文档无函数，或者有函数无文档的情形

- 重复建设严重，各程序包交叉重叠，体系较杂乱

  几乎每种方法都可以找到2~3个功能包来实现

本课程对此的应对思路

- 以统计学为主线串起相关功能，帮助学员构建完整的统计知识体系
- 对python尚未完整实现的方法细节，仍然进行方法学的介绍，保留未来的操作升级空间
- 尽量以pandas、statsmodels、sklearn包为课程轴心进行讲授，使软件操作知识体系化，替代并减少对使用过于灵活的numpy、scipy等底层包功能的介绍
- 使用真实商业项目数据，将python的学习代入具体分析场景

In [ ]:

```
import pandas as pd
import scipy.stats as ss
import matplotlib

# 解决绘图的兼容问题
%matplotlib inline
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
```

In [ ]:

```
ccss = pd.read_excel("CCSS_sample.xlsx", sheet_name = 'CCSS')
ccss.head()
```

1 变量的统计描述

1.1 中国消费者信心指数项目概况

1.2 连续变量的统计描述

numpy中内置了诸如mean，median等汇总函数，但是基于pandas框架的变量统计描述更为方便和灵活。

1.2.1 直接使用汇总函数

可以直接使用的汇总函数（绝大部分在不分组状态下也可以使用）

```
count()    Number of non-null observations
size()    group sizes
sum()    Sum of values
mean()    Mean of values
median()    Arithmetic median of values
min()    Minimum
max()    Maximum
std()    Unbiased standard deviation
var()    Unbiased variance
skew()    Unbiased skewness (3rd moment)
kurt()    Unbiased kurtosis (4th moment)
quantile()    Sample quantile (value at %)
apply()    Generic apply
cov()    Unbiased covariance (binary)
corr()    Correlation (binary)
```

与数值定位有关的特殊函数

```
argmin / argmax 最小值和最大值对应的绝对位置（整数）
idxmin / idxmax 最小值和最大值对应的索引值
```

In [ ]:

```
print(ccss.median())
ccss.s3.mean()
```

In [ ]:

```
# 注意可能有输出混乱的函数
ccss.sum()
```

In [ ]:

```
# 案例分组时的统计描述
ccss.groupby('s0').mean()
```

In [ ]:

```
ccss.groupby('s0').s3.mean()
```

In [ ]:

```
ccss.groupby('s0')['s3', 'index1'].mean()
```

In [ ]:

```
ccss.s3.plot.hist()
```

In [ ]:

```
ccss.s3.plot.box()
```

1.2.2 describe命令

一次性输出常用的集中趋势和离散趋势汇总指标。

百分位数的输出为其特色功能。

df.describe(

```
    percentiles : 需要输出的百分位数，列表格式提供，如[.25, .5, .75]
    include = 'None' : 要求纳入分析的变量类型白名单
        None (default) : 只纳入数值变量列
        A list-like of dtypes : 列表格式提供希望纳入的类型
        'all' : 全部纳入
    exclude : 要求剔除出分析的变量类型黑名单，选项同上
```

)

In [ ]:

```
ccss.describe()
```

In [ ]:

```
# 案例分组时的统计描述
ccss.groupby('s0').s3.describe(percentiles=[.05,.1])
```

1.2.3 statsmodels的实现方式

statsmodels中的统计描述对数据格式的要求更严格，但功能更强。

DescrStatsW类不仅可以用于进行变量的统计描述，更是进一步进行各种比较的基础对象。

class statsmodels.stats.weightstats.DescrStatsW(

```
    data : 希望分析的一维数组或者二维数据框（案例 * 变量 的二维表格式）
    weights = None : 案例权重，总和应当等于样本量
    ddof = 0 : 用于计算第二统计量的校正自由度，罕用
```

)

基于DescrStatsW类可计算的统计量：

```
nobs()    equal to sum of weights
sum()    weighted sum of data
sum_weights()

mean()    weighted mean of data
quantile(probs)    Compute quantiles for a weighted sample.

std()    standard deviation with default degrees of freedom correction
std_mean()    standard deviation of weighted mean

sumsquares()    weighted sum of squares of demeaned data
var()    variance with default degrees of freedom correction
```

In [ ]:

```
from statsmodels.stats import weightstats as ws

des = ws.DescrStatsW(ccss.loc[:, ['index1', 'index1a', 'index1b']])
des.nobs # 无参函数不能写括号，否则报错
```

In [ ]:

```
des.mean() # 无参函数不能写括号，否则报错
```

In [ ]:

```
des.quantile([.05, .1, .5, .9, .95])
```

In [ ]:

```
print(des.mean)
des.var
```

In [ ]:

```
# 混入字符串变量时会出错
des = ws.DescrStatsW(ccss)
des.var
```

1.3 分类变量的统计描述

python目前尚未对多选题的统计描述提供任何支持，因此只能按照多个离散变量的方式来对多选题数据进行分析。

1.3.1 单变量的频数统计

Series.value_counts(

```
normalize = False : 是否返回构成比而不是原始频数
sort = True : 是否按照频数排序(否则按照原始顺序排列)
ascending = False : 是否升序排列
bins : 对数值变量直接进行分段，可看作是pd.cut的简便用法
dropna = True : 结果中是否包括NaN
```

)

In [ ]:

```
ccss.time.value_counts()
```

In [ ]:

```
ccss.s3.value_counts() # 数值变量也可直接列出频数表
```

In [ ]:

```
ccss.s3.value_counts(bins = 20) # 数值变量也可直接列出频数表
```

In [ ]:

```
ccss.s5.value_counts(True)
```

In [ ]:

```
ccss.s5.value_counts().plot.bar()
```

In [ ]:

```
ccss.s5.value_counts().plot.pie()
```

1.3.2 交叉表

pandas的crosstab命令可以完成基本的制表任务，但是和统计软件中的同类命令不同，缺少进行行列变量关联性检验的功能，这方面的任务需要使用statsmodels完成

pd.crosstab(

行列设定
index / columns : 行变量/列变量，多个时以list形式提供
rownames / colnames = None : 交叉表的行列名称
单元格设定
values : 在单元格中需要汇总的变量列，需要进一步指定aggfunc
aggfunc : 相应的汇总函数
行列百分比计算
normalize = False : {'all', 'index', 'columns'}, or {0,1}
'all' / True : 总计百分比
'index' / 0 : 分行计算百分比
'columns' / 1 : 分列计算百分比
当margins = True时，也同时计算边际汇总的百分比
汇总设定
margins = False : 是否加入行列汇总
margins_name = 'All' : 汇总行/列的名称
dropna = True :

)

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