#### 实验报告2-相关性分析

* 现有一份美国餐厅评分数据，部分数据内容如下。请建立各餐厅各种评分（平均值）情况表，在该表上新添一列表明评分人数。  
  提取评分人数大于10的数据建立新表，分析总评分与食物评分  
  和服务评分的相关性。（自选扩展（加分题）：筛选无效客户）
  + 首先利用groupby分类，将每个地方的某一种评分平均值列举出
* from pandas import DataFrame
    
  from pandas import read\_table
    
  import numpy
    
  df = read\_table("D:/programming/condaProgram/resourse/rating\_final.csv",sep=','
    
   ,encoding='utf-8')
    
  print(df)
    
  print("最小值：",df.rating.describe());
    
  #df.userID.size
    
  #df.groupby(by=['placeID',]).mean()
    
  df.groupby(by=['placeID'])['rating'].agg({'平均值':numpy.mean})
    
  #某一个地方的某一种评分的平均值
    
  group\_df = df.groupby(['placeID'])['rating'].agg([
    
   ('rating平均值','mean'),('food\_rating平均值','mean'),('service\_rating平均值','mean')
    
   ]).reset\_index()
    
  group\_df
  + 新添一列表明评分人数
  + df0 = df.groupby(['placeID'])['userID'].agg([('人数','count')]).reset\_index()
      
    df0
      
    group\_df['人数'] = df0['人数']
      
    group\_df
  + 提取评分人数大于10的数据建立新表
  + group\_df[group\_df['人数']>10]
  + 分析总评分与食物评分
    - **交叉分析**用于分析两个变量之间的相互关系
    - pivot\_table(values,index,columns,aggfunc,fill\_value)
      * 首先，index是用来将某一列（如姓名）作行键的，可以设置多个，顺序无所谓，虽然有些许不同，但意义是一样的。不可缺。columns可省略
      * values是可选的，用来放values,如果不设置，默认用源数据，否则根据需求设置
      * aggfunc则是用来装方法
  + df1.pivot\_table(index=['rating平均值'])#这时rating平局值这一行会在最左边
      
    df1.pivot\_table(index=['rating平均值']，values=[],aggfunc=[numpy.mean])
    - **结构分析**基于分组分析与交叉分析之上，分组分析使用pandas.cut(df.column,bins,label)
      * 结构分析用来分析比重
      * 最后分析结果如下
      * #交叉分析
          
        df2=df1.pivot\_table(values=['food\_rating平均值','rating平均值'],index=['placeID'],aggfunc=[numpy.sum])
          
        df2
          
        #结构分析
          
        df2.div(df2.sum(axis=1),axis=0)#分析两列分别总和占比，发现一样
  + 分析总评分与食物评分和服务评分的相关性。
    - **相关分析**df.corr(method='',min\_periods=1),min\_periods表示样本最少的数据量

| * + - Pearson | * + - 线性数据分析 |
| --- | --- |
| * + - **kendall** | * + - **无序序列分析，非正态数据** |
| * + - **spearman** | * + - **非线性** |

* + - 相关分析df[].corr(df[''])两列之间的相关分析
    - df.loc[].corr()多列的相关分析
    - 代码如下
    - df1.loc[:,'rating平均值':'service\_rating平均值'].corr()
  + 自选扩展（加分题）：筛选无效客户
    - 将评分全评为0的用户视为无效
    - df4=DataFrame(df.loc[:,'rating':'service\_rating'].sum(axis=1)
        
       ,columns=['总分'])
        
        
      df4
        
      df4[df4['总分']==0].reset\_index()
    - 