PYTHON数据科学系列课程（四）
PANDAS统计与数据分析

东南大学 学习科学研究中心 儿童发展与教育研究所
夏小俊
http://www.seuct.com

本章内容提要

- 4.1 统计与排序
- 4.2 导入导出
- 4.3 数据清洗
- 4.4 数据转换
- 4.5 数据抽取
- 4.6 数据合并
- 4.7 数据计算

本章内容提要

- 4.8 对比分析
- 4.9 基本统计分析
- 4.10 分组分析
- 4.11 结构分析
- 4.12 分布分析
- 4.13 交叉分析

4.1 统计和排序

In[1]:

```
import numpy as np
import pandas as pd
df2 = pd.read_csv('shopping.csv')
df2 = df2[["id","date","money"]]
df2.describe()
```

Out[1]:

	id	area_mean
count	5.690000e+02	569.000000
mean	3.037183e+07	654.889104
std	1.250206e+08	351.914129
min	8.670000e+03	143.500000
25%	8.692180e+05	420.300000
50%	9.060240e+05	551.100000
75%	8.813129e+06	782.700000
max	9.113205e+08	2501.000000

4.1 统计和排序

In[2]:

```
df2.head(8) #前8条
df2.tail()  #默认后5条
```

Out[2]:

4.1 统计和排序

In[3]:
```
df2[df2.money > 10].count() #统计非na的数量
```

Out[3]:
```
id      6
date    6
money   6
dtype: int64
```

4.1 统计和排序

In[4]:

```
df2.sort_values(by="money",axis=0,ascending=True).head()
```

Out[4]:

4.1 统计和排序

In[5]:
```
df2.sort_index(axis=1).head(3) #按列
```

Out[5]:

In[6]:
```
df2.sort_index(axis=0,ascending=False).head(3) #按行
```

Out[6]:

4.2 导入/导出

In[7]:

```
import os
print(os.getcwd())
```

In[8]:

```
df2.head(3).to_csv("df2.csv")
```

In[9]:

```
import pandas as pd
df3 = pd.read_csv('df2.csv')
```

4.2 导入/导出

In[10]:
```
df2.head(3).to_excel("df3.xls” ,index=False)
```

In[11]:
```
df3 = pd.read_excel("df3.xls")
df3
```

Out[11]:

4.3 数据清洗

在获取数据集之后，通常需要对原始的数据进行清洗和整理，以消除原始数据（脏数据）中的错误、缺失、重复等问题。

数据清洗的目的就是将原始数据转换可进行数据分析的形式，使得数据保持准确性、一致性和有效性。

常用的数据清洗方法包括数据排序、重复数据处理、缺失数据处理、空格数据处理等。

4.3.1 数据排序

In[1]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data1.csv')
sort_data = data.sort_values(
    by = ['年龄' , '性别'],
    ascending = [True , False]
)
```

提示：汉字排序使用ord函数

4.3.2 重复数据处理

In[2]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data2.csv')
print(data)
dIndex = data.duplicated() #完全重复
dIndex

dIndex = data.duplicated(['性别']) #性别列存在重复
dIndex

data[dIndex] #提取重复行
```

4.3.2 重复数据处理

In[3]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data2.csv')
print(data)
cdata = data.drop_duplicates(['性别']) #删除重复行
cdata
```

4.3.3 缺失数据处理

数据缺失的原因：

1）数据暂时无法获得，如未婚者的配偶等

2）数据遗漏或丢失

缺失值的处理方法

1）数据补齐：用平均值、中位数、众数等来填充

2）删除缺失行：视数据量大小决定

3）暂不处理

4.3.3 缺失数据处理

In[4]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data3.csv')
print(data)
print(data.isnull())  #或notnull 判断是否缺失

data.消费 = data.消费.fillna(data.消费.mean()) #平均值填充
print(data)

cdata = data.dropna() #删除缺失
print(cdata)
```

4.3.3 缺失数据处理

In[5]:

```
import pandas as pd
import numpy as np
A=pd.DataFrame(np.array([10,11,20,21]).reshape(2,2),columns=list(
"ab"),index=list("SW"))
B=pd.DataFrame(np.array([1,1,1,2,2,2,3,3,3]).reshape(3,3),
columns=list("abc"),index=list("SWT"))
C=A+B
D=A.add(B,fill_value=0)
E=A.add(B,fill_value=A.stack().mean())
C.dropna(axis=0)
C.fillna(0) #C.fillna(method="bfill") #下一个非缺失值填充该缺失值
C.fillna(method=”ffill“ ,axis=1) #前一个非缺失值填充该缺失值
```

4.3.4 空格数据处理

In[6]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data4.csv')
data.name = data.name.str.strip()
data.to_csv('data4_new.csv', index=0)
```

4.4.1 数值和字符的转换

In[7]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data5.csv') #数值形的数据会被默认设定为数字类型
data.电话号码 = data.电话号码.astype(str)
data.电话号码.dtype
data.电话号码 = data.电话号码.astype(float)
data.电话号码.dtype
```

4.4.2 字符转时间

In[8]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data6.csv')
data['时间'] = pd.to_datetime(
    data.注册时间,
    format='%Y/%m/%d'
)
data[‘年月’] = data.时间.dt.strftime(‘%Y-%m’) #时间的格式化

提示：%Y 年 %m月 %d日 %H时 %M分 %S秒
```

4.5 数据抽取

数据抽取也称为数据拆分，指保留提取数据表的某些字段或部分记录的信息，形成新的字段和记录，主要方法有字段拆分、记录抽取和随机抽样。

4.5.1 字段拆分

In[9]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data7.csv')
data.tel = data.tel.astype(str)
data[‘bands’] = data.tel.str.slice(0,3) #向量化运算，不能直接切片
data['areas'] = data.tel.str.slice(3,7)
data['nums'] = data.tel.str.slice(7,11)
data
```

4.5.1 字段拆分

In[10]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data8.csv')
datas = data.name.str.split(‘ ’ , 1 , True) #第二个参数展开n+1，第三个是否展开为数据框
datas.columns = ['band' , 'name']
datas
```

4.5.1 字段拆分

In[11]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data9.csv')
data['时间'] = pd.to_datetime(data.注册时间)
data['年'] = data.时间.dt.year
data['周'] = data.时间.dt.weekday
data
```

4.5.2 记录抽取

In[12]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data10.csv')
data1 = data[data.title.str.contains('华为', na=False)] #关键词抽取
data2 = data[data.title.isnull()]
data3 = data[data.comments>10000]
data4 = data[data.comments.between(1000 , 10000)]
```

4.5.2 记录抽取

In[12]:

```
data['ptime'] = pd.to_datetime(data.ptime)
dt1 = datetime(year = 2015,month=1,day=1)
dt2 = datetime(year = 2015,month=12,day=31)
data5 = data[(data.ptime>=dt1) & (data.ptime <= dt2)] #注意该式子的写法！
data6 = data[~data.title.str.contains('华为', na=False)]
data7 = data[(data.comments>1000) & (data.comments<10000) | data.title.str.contains('华为', na=False)]
```

4.5.3 随机抽样

随机抽样在数据量较大的情况下具有较高的实用价值。

随机抽样包括简单随机抽样、分层抽样和系统抽样等方法，其中简单随机抽样最常用。

简单随机抽样又分为重复抽样（放回）和不重复抽样（不放回）。

4.5.3 随机抽样

In[13]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data11.csv')
data1 = data.sample(n=3)
data2 = data.sample(frac=0.2)
data3 = data.sample(n=3,replace=True)
```

4.6 数据合并

数据合并是指根据某几个字段的信息或不同的记录的值，组合为一个新的字段、新记录数据。

常用的数据合并方法包括记录合并、字段合并、字段匹配。

4.6.1 记录合并

In[14]:

```
import numpy as np
import pandas as pd
data1 = pd.read_csv('data12_1.csv')
data2 = pd.read_csv('data12_2.csv')
data3 = pd.read_csv('data12_3.csv')
data = pd.concat([data1 , data2 , data3])
data = pd.concat([
    data1[['id','comments']],
    data2[['comments','title']],
    data3[['id','title']]
    ],sort=True)
```

4.6.2 字段合并

In[15]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data13.csv')
data = data.astype(str) #全部转为str
data['tel'] = data['band'] + data['area'] + data['num']
data
```

4.6.3 字段匹配

In[16]:

```
import numpy as np
import pandas as pd
items = pd.read_csv('data14_name.csv')
prices = pd.read_csv('data14_price.csv')
itemprices =
pd.merge(items,prices,left_on='id',right_on='id',how='outer')
#left,right,outer
itemprices
```

4.7.1 简单计算

In[17]:

```
import numpy as np
import pandas as pd
data = pd.read_csv('data15.csv')
data['total'] = data.price * data.num
data
```

4.7.2 时间计算

In[18]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data16.csv')
data['时间'] = pd.to_datetime(data.注册时间)
data['注册天数'] = datetime.now() - data.时间
data['注册天数'] = data.注册天数.dt.days
data
```

4.7.3 数据标准化

In[19]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data17.csv')
data['消费标准化'] = round(
    (data.消费 - data.消费.min()) /
    (data.消费.max() - data.消费.min())
    ,2
    )
data
```

4.7.4 数据分组

根据不同的分组方式，将数据进行等距或非等距的分组，这个过程也称为数据离散化。

通过分组可以对数据对象按不同的区间进行研究，以揭示其内在的联系和规律性。

4.7.4 数据分组

In[20]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data18.csv')
bins = [0,20,40,60,80,100]
data['cut'] = pd.cut(data.cost,bins,right=True)
mylabels = ['0到20' ,'20到40' , '40到60', '60到80','80到100']
data['cut'] = pd.cut(data.cost,bins,right=False,labels=mylabels)
data
```

4.8 对比分析

数据分析的三大作用：现状分析、原因分析、预测分析

现状分析：以对比为基本方法，包括对比分析、描述性分析、分组分析、结构分析、分布分析、交叉分析、RFM分析、矩阵分析……等

原因分析：以细分为基本方法，包括结构分解法、因素分解法、漏斗图分析等

预测分析：以预测为基本方法，包括相关分析、回归分析、时间序列等

4.8 对比分析

对比分析：将两个或更多的数据进行比较，分析它们的差异，揭示事物发展变化的情况和规律性。

对比分析需要对指标从不同的维度进行分析。

指标：衡量事物发展程度的单位和方法，如人口、GDP、利润率等；

案例：QQ（数量、质量（广度、深度））模型

维度：事物或现象的某种特质，如产品类型、用户类型、地区、时间等，维度可以是定性或定量的。

案例：时间维度（环比、同比、定基比）

4.9 基本统计分析

基本统计分析，也叫描述性统计分析，它是指通过制表、分类、图形以及计算概括性来描述数据特质的各项活动，以发现其内在规律的统计分析方法。

描述性统计分析主要包括数据的集中趋势分析、离散程度分析、数据的频数分析等，指标包括计数、均值、求和、方差等。

pandas中通过describe等函数进行描述性分析。

4.9 基本统计分析

In[21]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data19.csv')
data.sales.describe()
data.sales.count()
data.sales.max()
data.sales.min()
data.sales.sum()
data.sales.mean()
data.sales.var()
data.sales.std()
data.sales.quantile(0.3 , interpolation='nearest') #lower higher等
```

4.10 分组分析

分组分析是指根据分组字段，将对象划分为不同的部分，以对比分析各组差异的一种方法。

分组类型有两类：定性分组和定量分组

pandas中通过groupby和agg等聚合（统计）函数完成分组功能。

4.10 分组分析

In[22]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data20.csv')
ga = data.groupby(by=['gender'],as_index=True)['age'].agg('mean')
print(ga)



def myfunc(x):
  x['age'] /= x['age'].sum()
  return x
data.groupby('gender').apply(myfunc).head()
```

4.11 结构分析

结构分析是在分组的基础上，计算各部分所占的比重，从而分析总体的内部构成。

通常结构分析通过饼图、圆环图、树状图等进行数据展现。

4.11 结构分析

In[23]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data21.csv')
ga = data.groupby('gender')['id'].agg('count')
ga
ga.sum()
ga/ga.sum()
```

4.12 分布分析

分布分析通常建立在定量分组的基础上，重点在查看数据的分布情况，在生活当中非常常见。

分布分析的要点：横轴也就是分组的依据不能更改顺序。

4.12 分布分析

In[24]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data22.csv')
bins = [0,20,30,40,100]
agelabels = ['20岁及以下','21至30岁','31至40岁','41岁以上']
data['年龄分层'] = pd.cut(data.年龄,bins,labels=agelabels)
ageresult = data.groupby('年龄分层')['用户ID'].agg('count')
ageresult
ageresult / ageresult.sum()
pageresult = round(ageresult / ageresult.sum() , 4) * 100
```

4.13 交叉分析

交叉分析通常用来建立两个或更多分组变量的关系，以交叉表的形式来进行变量间的对比分析。

用来交叉的分组变量，定性和定量均可。

交叉分析的维度，通过不宜超过两个，维度越多分组越细，越容易丧失重点。

4.13 交叉分析

In[25]:

```
import numpy as np
import pandas as pd
from datetime import datetime
data = pd.read_csv('data23.csv')
bins = [0,20,30,40,100]
agelabels = ['20岁及以下','21至30岁','31至40岁','41岁以上']
data['年龄分层'] = pd.cut(data.年龄,bins,labels=agelabels)
ptresult = data.pivot_table(
    values = '用户ID',
    index = '年龄分层',
    columns = '性别',
    aggfunc = 'count'
)
```

谢谢大家！