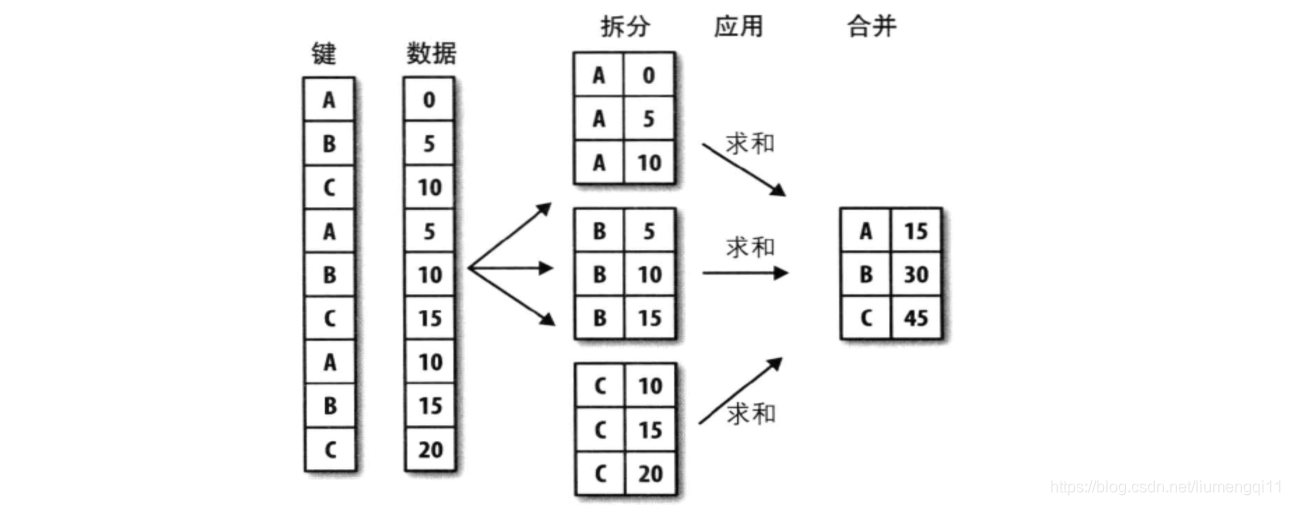
**数据分组运算**

分组计算根据某个或者某几个字段对数据集进行分组，然后运用特定的函数，得到结果



1.

使用groupby方法进行分组计算，得到分组对象GroupBy

2.

语法为df.groupby(by=)

3.

分组对象GroupBy可以运用描述性统计方法, 如count、mean 、median 、max和min等

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

df = pd.read\_csv('online\_order.csv',encoding = 'gbk',dtype={'customer':str,'order':str})

# df.dtypes

# customer object

# order object

# total\_items int64

# discount% float64

# weekday int64

# hour int64

# Food% float64

# Fresh% float64

# Drinks% float64

# Home% float64

# Beauty% float64

# Health% float64

# Baby% float64

# Pets% float64

# dtype: object

grouped = df.groupby('weekday') #创建分组对象,按照星期进行分组

grouped.mean() #调用平均数方法(mean)算每个分组中各个不同列的平均数

# total\_itemsdiscount%hourFood%Fresh%Drinks%Home%Beauty%Health%Baby%Pets%

# weekday

# 130.6621778.58070514.69312222.69086620.00090422.52299313.9325536.9723941.15228511.5925621.007306

# 231.8686128.63801414.96619723.99491519.40773824.34645913.5591914.9033661.07942311.2772841.272638

# 331.8697967.79450715.05989824.30927419.95765323.82247013.2820886.7026401.1568299.5913890.937205

# 432.2518998.06815514.324185

grouped.sum()['total\_items']# 计算不同的星期，商品数量的总和

# weekday

# 1 191240

# 2 158387

# 3 150043

# 4 131620

# 5 79710

# 6 74149

# 7 167049

# Name: total\_items, dtype: int64

grouped = df.groupby(by =['customer','weekday'])#创建分组对象,按照用户和星期

grouped.sum()['total\_items'].head(50) #调用方法,计算不同的用户周一到周天的订购商品数量的总和

# customer weekday

# 0 4 96

# 5 38

# 1 1 423

# 2 127

# 4 37

# 5 36

**数据集**

链接：<https://pan.baidu.com/s/1VVn_B0VPaRWdjaZwVUREpw>提取码：1111

在线杂货店订单数据
customer ：消费者ID，一个消费者可能有多个订单
order ：订单ID，订单的唯一标识，不重复
total\_items ：订单中购买的商品数量
discount% ：收到的总折扣百分比
weekday ：下单时间：星期几下单，1-7 为 周一至周日
hour ：下单时间：几点下单，0-23 为二十四小时制
Food% ：食物在订单总价中占比，食物为非生鲜类食物
Fresh% ：生鲜类食物在订单总中占比
Drinks% ：饮品在订单总价占比，由于高糖税可能导致总折扣为负
Home% ：家居用品在订单总价中占比
Beauty% : 美妆类产品在订单总价中占比
Health% ：保健类产品在订单总价中占比
Baby% ：母婴类产品在订单总价占比
Pets% ：宠物用品在订单总价占比

**聚合函数使用**

1.

对分组对象使用agg聚合函数

2.

Groupby.agg(func)

3.

针对不同的变量使用不同的统计方法

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

df = pd.read\_csv('online\_order.csv',encoding = 'gbk',dtype={'customer':str,'order':str})

# df.dtypes

# customer object

# order object

# total\_items int64

# discount% float64

# weekday int64

# hour int64

# Food% float64

# Fresh% float64

# Drinks% float64

# Home% float64

# Beauty% float64

# Health% float64

# Baby% float64

# Pets% float64

# dtype: object

grouped = df.groupby('weekday')#创建分组对象,按照周一到周天进行分组

#agg是一个作用于series或者DataFrame的函数,主要目的是针对分组后的对象，使用相关函数进行计算

grouped.agg([np.mean,np.max]).head(20) #对分组后对象，计算均值和总和

# total\_itemsdiscount%hourFood%Fresh%...Home%Beauty%Health%Baby%Pets%

# meanamaxmeanamaxmeanamaxmeanamaxmeanamax...meanamaxmeanamaxmeanamaxmeanamaxmeanamax

# weekday

# 130.6621771828.580705100.014.6931222322.690866100.020.000904100.0...13.932553100.06.972394100.01.152285100.0011.592562100.01.007306100.0

# 231.8686122828.638014100.014.9661972323.994915100.019.407738100.0...13.559191100.04.903366100.01.079423100.0011.277284100.01.272638100.0

# 331.8697962987.794507100.015.0598982324.309274100.019.957653100.0...13.282088100.06.702640100.01.156829100.009.591389100.00.937205100.0

# 432.2518992218.068155100.0

grouped.agg({'total\_items':np.sum,'Food%': [np.mean,np.median]}) #对2个变量分别计算不同的统计量

# total\_itemsFood%

# summeanmedian

# weekday

# 119124022.69086618.740

# 215838723.99491519.815

# 315004324.30927420.510

# 413162024.37436419.940

# 57971024.60279020.130

# 67414923.74319620.655

# 716704922.27151218.925

df[['total\_items','Food%','Drinks%']].agg([np.sum,np.mean]) #也可以直接对数据进行汇总

# total\_itemsFood%Drinks%

# sum952198.000000706812.190000700477.060000

# mean31.73993323.56040623.3

**数据集**

链接：<https://pan.baidu.com/s/1VVn_B0VPaRWdjaZwVUREpw>提取码：1111

在线杂货店订单数据
customer ：消费者ID，一个消费者可能有多个订单
order ：订单ID，订单的唯一标识，不重复
total\_items ：订单中购买的商品数量
discount% ：收到的总折扣百分比
weekday ：下单时间：星期几下单，1-7 为 周一至周日
hour ：下单时间：几点下单，0-23 为二十四小时制
Food% ：食物在订单总价中占比，食物为非生鲜类食物
Fresh% ：生鲜类食物在订单总中占比
Drinks% ：饮品在订单总价占比，由于高糖税可能导致总折扣为负
Home% ：家居用品在订单总价中占比
Beauty% : 美妆类产品在订单总价中占比
Health% ：保健类产品在订单总价中占比
Baby% ：母婴类产品在订单总价占比
Pets% ：宠物用品在订单总价占比

**分组对象与apply函数**

1.

函数apply即可用于分组对象,也可以作用于dataframe数据

2.

Groupby.apply(func)

3.

需要注意axis=0和axis=1的区别

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

df = pd.read\_csv('online\_order.csv',encoding = 'gbk',dtype={'customer':str,'order':str})

# df.dtypes

# customer object

# order object

# total\_items int64

# discount% float64

# weekday int64

# hour int64

# Food% float64

# Fresh% float64

# Drinks% float64

# Home% float64

# Beauty% float64

# Health% float64

# Baby% float64

# Pets% float64

# dtype: object

grouped = df.groupby('weekday')

grouped.apply(np.mean)[['total\_items','Food%']] #可以做聚合

# total\_itemsFood%

# weekday

# 130.66217722.690866

# 231.86861223.994915

# 331.86979624.309274

# 432.25189924.374364

# 531.40661924.602790

# 632.15481423.743196

# 732.37383722.271512

# apply 中,axis=0 或者 axis=1的区别

df.columns

# Index(['customer', 'order', 'total\_items', 'discount%', 'weekday', 'hour',

# 'Food%', 'Fresh%', 'Drinks%', 'Home%', 'Beauty%', 'Health%', 'Baby%',

# 'Pets%'],

# dtype='object')

var\_name = ['Food%', 'Fresh%', 'Drinks%', 'Home%', 'Beauty%', 'Health%', 'Baby%',

'Pets%'] # 不同类型的商品占比

df[var\_name].apply(np.sum,axis = 0) #相当于计算每列的总和

# Food% 706812.19

# Fresh% 606818.38

# Drinks% 700477.06

# Home% 406187.25

# Beauty% 176788.48

# Health% 33988.76

# Baby% 332884.34

# Pets% 31292.61

# dtype: float64

df['sum'] = df[var\_name].apply(np.sum,axis=1) # 相当于计算每行的总和

var\_name.append('sum')

df[var\_name] #查看数据，会发现总和为1

# Food%Fresh%Drinks%Home%Beauty%Health%Baby%Pets%sum

# 09.4687.063.480.000.000.000.000.00100.00

# 115.8775.806.222.120.000.000.000.00100.01

df[var\_name].apply(lambda x: x[0] - x[1],axis = 1) #计算食物在订单总价中占比 - 生鲜类食物在订单总中占比

0 -77.60

1 -59.93

2 -39.87

3 -7.18

4 -36.25

5 -100.00

**数据集**

链接：<https://pan.baidu.com/s/1VVn_B0VPaRWdjaZwVUREpw>提取码：1111

在线杂货店订单数据
customer ：消费者ID，一个消费者可能有多个订单
order ：订单ID，订单的唯一标识，不重复
total\_items ：订单中购买的商品数量
discount% ：收到的总折扣百分比
weekday ：下单时间：星期几下单，1-7 为 周一至周日
hour ：下单时间：几点下单，0-23 为二十四小时制
Food% ：食物在订单总价中占比，食物为非生鲜类食物
Fresh% ：生鲜类食物在订单总中占比
Drinks% ：饮品在订单总价占比，由于高糖税可能导致总折扣为负
Home% ：家居用品在订单总价中占比
Beauty% : 美妆类产品在订单总价中占比
Health% ：保健类产品在订单总价中占比
Baby% ：母婴类产品在订单总价占比
Pets% ：宠物用品在订单总价占比

**透视图与交叉表**

在数据分析中，数据透视表是常见的工具之一，需要根据行或列对数据进行各个维度数据的汇总，在pandas中，提供了相关函数解决此类问题，交叉表更多用于频数的分析。pivot\_table( data, index, columns,values, aggfunc, fill\_value,margins, margins\_name=)Index : 行分组键columns: 列分组键values: 分组的字段，只能为数值型变量aggfunc: 聚合函数margins: 是否需要总计

交叉表用于计算分组频率pd.crosstab(index,columns,normalize)Index: 行索引Columns: 列索引Normalize: 数据对数据进行标准化，index表示行，column表示列

import pandas as pd

import numpy as np

import os

os.chdir(r'C:\Users\chell\Desktop')

#读取数据

#读取数据

df = pd.read\_csv('online\_order.csv',encoding = 'gbk',dtype={'customer':int,'order':str})

df.columns

# Index(['customer', 'order', 'total\_items', 'discount%', 'weekday', 'hour',

# 'Food%', 'Fresh%', 'Drinks%', 'Home%', 'Beauty%', 'Health%', 'Baby%',

# 'Pets%'],

# dtype='object')

# 单个变量

#margin =True 表示是否需要总计

pd.pivot\_table(data= df,index='weekday',values='total\_items',aggfunc=[np.sum,np.size],margins=True,margins\_name='总计')#按照周一到周天计算购买的商品数量总数和次数

# sum size

# total\_items total\_items

# weekday

# 1 191240 6237

# 2 158387 4970

# 3 150043 4708

# 4 131620 4081

# 5 79710 2538

# 6 74149 2306

# 7 167049 5160

# 总计 952198 30000

# 使用交叉表

#是一种计算分组频数的特殊透视表

# 不同的星期，不同的折扣交叉表

pd.crosstab(index =df['weekday'], columns=df['discount%'],margins=True)

# discount% -65.15 -63.64 -47.26 -39.84 ... 98.54 98.6 100.0 All

# weekday ...

# 1 1 0 0 0 ... 1 0 71 6237

# 2 0 0 0 0 ... 0 0 78 4970

# 3 0 0 0 0 ... 0 0 59 4708

# 4 0 0 0 0 ... 0 0 59 4081

# 5 0 0 1 0 ... 0 1 52 2538

# 6 0 0 0 1 ... 0 0 25 2306

# 7 0 1 0 0 ... 0 0 52 5160

# All 1 1 1 1 ... 1 1 396 30000

#按照行进行汇总，计算频数占比

pd.crosstab(index =df['weekday'], columns=df['discount%'],margins=True, normalize='all')

# discount% -65.15 -63.64 -47.26 ... 98.6 100.0 All

# weekday ...

# 1 0.000033 0.000000 0.000000 ... 0.000000 0.002367 0.207900

# 2 0.000000 0.000000 0.000000 ... 0.000000 0.002600 0.165667

# 3 0.000000 0.000000 0.000000 ... 0.000000 0.001967 0.156933

# 4 0.000000 0.000000 0.000000 ... 0.000000 0.001967 0.136033

# 5 0.000000 0.000000 0.000033 ... 0.000033 0.001733 0.084600

# 6 0.000000 0.000000 0.000000 ... 0.000000 0.000833 0.076867

# 7 0.000000 0.000033 0.000000 ... 0.000000 0.001733 0.172000

# All 0.000033 0.000033 0.000033 ... 0.000033 0.013200 1.000000

#

# [8 rows x 3580 columns]

#按照行进行汇总，计算频数占比

pd.crosstab(index =df['weekday'], columns=df['discount%'],margins=True, normalize='all')#index表示计算行百分比，columns表示计算列百分比

# discount% -65.15 -63.64 -47.26 ... 98.6 100.0 All

# weekday ...

# 1 0.000033 0.000000 0.000000 ... 0.000000 0.002367 0.207900

# 2 0.000000 0.000000 0.000000 ... 0.000000 0.002600 0.165667

# 3 0.000000 0.000000 0.000000 ... 0.000000 0.001967 0.156933

# 4 0.000000 0.000000 0.000000 ... 0.000000 0.001967 0.136033

# 5 0.000000 0.000000 0.000033 ... 0.000033 0.001733 0.084600

# 6 0.000000 0.000000 0.000000 ... 0.000000 0.000833 0.076867

# 7 0.000000 0.000033 0.000000 ... 0.000000 0.001733 0.172000

# All 0.000033 0.000033 0.000033 ... 0.000033 0.013200 1.000000

#

# [8 rows x 3580 columns]

#

# Process finished with exit code 0

pd.crosstab(index =df['weekday'], columns=df['discount%'],margins=True, normalize='columns')#index表示计算行百分比，columns表示计算列百分比

# discount% -65.15 -63.64 -47.26 -39.84 ... 98.54 98.6 100.0 All

# weekday ...

# 1 1.0 0.0 0.0 0.0 ... 1.0 0.0 0.179293 0.207900

# 2 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.196970 0.165667

# 3 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.148990 0.156933

# 4 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.148990 0.136033

# 5 0.0 0.0 1.0 0.0 ... 0.0 1.0 0.131313 0.084600

# 6 0.0 0.0 0.0 1.0 ... 0.0 0.0 0.063131 0.076867

# 7 0.0 1.0 0.0 0.0 ... 0.0 0.0 0.131313 0.172000

**数据集**

链接：<https://pan.baidu.com/s/1VVn_B0VPaRWdjaZwVUREpw>提取码：1111

在线杂货店订单数据
customer ：消费者ID，一个消费者可能有多个订单
order ：订单ID，订单的唯一标识，不重复
total\_items ：订单中购买的商品数量
discount% ：收到的总折扣百分比
weekday ：下单时间：星期几下单，1-7 为 周一至周日
hour ：下单时间：几点下单，0-23 为二十四小时制
Food% ：食物在订单总价中占比，食物为非生鲜类食物
Fresh% ：生鲜类食物在订单总中占比
Drinks% ：饮品在订单总价占比，由于高糖税可能导致总折扣为负
Home% ：家居用品在订单总价中占比
Beauty% : 美妆类产品在订单总价中占比
Health% ：保健类产品在订单总价中占比
Baby% ：母婴类产品在订单总价占比
Pets% ：宠物用品在订单总价占比