# 第2~3章

1. 已知

all\_data = [['John', 'Emily', 'Michael', 'Mary', 'Steven'],

['John', 'Emily', 'Michael', 'Mary'],

['Maria', 'Juan', 'Javier', 'Natalia', 'Pilar']]

请用代码找出有重复字母的名字。

1. 写出下列代码的输出结果

key\_list = [1, 2, 3, 4]

value\_list = ['red', 'blue', 'green', 'white']

mapping = {}

for key, value in zip(key\_list, value\_list):

mapping[key] = value

print(mapping)

1. 给出以下代码的运行结果：

seq = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]

for a, b, c in seq:

print('a={0}, b={1}, c={2}'.format(a, b,c))

1. 已知字符串集合strings = ['foo', 'card', 'bar', 'aaaa', 'abab']，请根据字符串中不同字母的数量对该字符串集合进行排序，即将strings变为['aaaa', 'foo', 'abab', 'bar','card']
2. 求100以内的质数.
3. 生成列表[[0,0,0,0,0,],[0,1,2,3,4,],[0,2,4,6,8,],[0,3,6,9,12,]]
4. 已知all\_data=[['John','Emily','Michael','Mary','Steven'],

['Maria','Juan','Javier','Natalia','Pilar']];

请用列表推导式获得一个列表，其中包含所有含有2个以上字母e的名字。

1. 给出一个字典a = {1:1, 2:2, 3:3}, 请尝试输出字典a的key ,以',' 连接，如 ‘1,2,3'。
2. 已知some\_tuples = [(('Zohn','Mmily'), ('Michael','Mary'), ('Steven','Maria')), (('Juan','Javier'), ('Natalia','Pilar'), ('MmmmM','Linda')), ]，请用列表推导式获得一个列表，其中包含的字符串含有至少一个字母m(不区分大小写)
3. 请填补下面的程序

In[1]: import bisect

In[2]: a=[1,2,3,3,3,4,4,5]

In[3]: bisect.bisect(a,3)

out[3]: ?

In[4]: bisect.insort(a,2)

In[5]: a

out[5]: ?

1. 编写一条语句，实现将sex = {'Alan':'female', 'Bob':'male', 'Wes':'male', 'Steven':'female'}, age = {'Bob':'30', 'Alan':'45', 'Wes':'39', 'Steven':'41'}, job = {'Alan':'actuary', 'Steven':'teacher', 'Wes':'tourist guide','Bob':'engineer'}三个字典中key相同的value值合并,构建出一个新的字典all，且all = {'Alan': 'female, 45, actuary', 'Bob': 'male, 30, engineer', 'Wes': 'male, 39, tourist guide', 'Steven': 'female, 41, teacher'}。

# 第4章

1. 写出下列代码的输出

import numpy as np

arr = np.array([1.2, 3.4, 2.5, 6.9])

arr2 = arr + arr

arr3 = arr2.astype(np.int32)

print(arr3)

1. 给出以下数据：

import numpy as np

index = np.array(['red', 'green', 'red', 'blue', 'white'])

data = np.arange(15).reshape((5, 3))

对data使用index的布尔索引，使得输出结果为：

[[ 3 4 5]

[ 9 10 11]

[12 13 14]]

1. 给出array=np.ones\_like([[0,1,2,3],[1,2,2,3],[0,0,0,0]])\*6

array的结果

1. 给出以下数组：xarr = np.array([1.1, 1.2, 1.3,1.4, 1.5])

yarr= np.array([2.1, 2.2, 2.3, 2.4, 2.5])

cond= np.array([True, False, True, True, False])

用代码实现：如果cond中为true，我们去xarr中对应的值，否则就取yarr中的值

1. 创建一个10\*10的二维数组，要求第一行、最后一行、第一列、最后一列为1，其余中间的 元素为0。
2. 使用np.random模块一次性抽取1000次投掷硬币的结果，每次投掷的结果为1或-1，然后计算累积值。
3. 从1到10中随机选取10个数，构成一个长度为10的数组，并将其排序。获取其最大值最小值，求和，求方差。
4. 假设数组arr = np.array([[ 0.0929, 0.2817, 0.769 , 1.2464],

[ 1.0072, -1.2962, 0.275 , 0.2289],

[ 1.3529, 0.8864, -2.0016, -0.3718],

[ 1.669 , -0.4386, -0.5397, 0.477 ],

[ 3.2489, -1.0212, -0.5771, 0.1241],

[ 0.3026, 0.5238, 0.0009, 1.3438],

[-0.7135, -0.8312, -2.3702, -1.8608]])，

1)请将此数组的整数部分和小数部分输出

2)将arr中所有的负数都换为它对应的整数部分的绝对值，所有的正数都换为它对应的小数部分。

1. 给定下面数组arr = np.array ([[2,-1,1],[-3,3,2],[4,3,-1]])

1)对所有数字求和

2)请计算大于0的值按行累和。

1. 下面代码的输出结果是？

a = np.arange(9);a.shape = (3, -1);print(a)

1. 下面代码的输出结果是？

arr = np.arange(12).reshape(3,4)

arr[[2,0,1]][:,[1,3,2]]

1. 下面代码的输出结果是？

arr1 = np.array(['Mike', 'Bob'])

arr2 = np.array(['Bob', 'Alice'])

arr3 = np.array([True, False])

print(~(arr1 != arr2) ^ arr3)

1. 下面代码的输出结果是？

arr1=np.zeros((4,4))

arr2=np.array([[np.arange(4)],[np.arange(4)]])

arr=arr1+arr2

arr[:3,2:]

1. x = np.array([[ 0, 1, 2],[ 3, 4, 5],[ 6, 7, 8],[ 9, 10, 11]])

rows = np.array([[0,0],[3,3]])

cols = np.array([[0,2],[0,2]])

y = x[rows,cols]

print (y)

1. 写出以下代码的运行结果。 arr = np.array([[1., 2., 3.], [4., 5., 6.]]) arr[0]=1 arr
2. 下面代码的运行结果是什么?

arr=np.arange(16).reshape((2,2,4))

arr1 = arr.reshape(4,4)

print(arr)

arr2 = arr.resize(4,4)

print(arr)

1. arr3d =np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])

arr3d[0]=42

arr3d结果是？

1. 设计程序，使输出结果为array([['#', '$','$', '$', '$'],

['$', '#', '$', '$', '$'],

['$', '$', '#', '$', '$'],

['$', '$', '$', '#', '$'],

['$', '$', '$', '$', '#']],dtype='<U1')

1. aar= np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])

写出下面每条语句的输出结果。

arr3d[0]

arr3d[[1],:,:]

arr3d[1,:,:]

1. import numpy as np，arr = np.array([[1,2 ,3], [4, 5, 6]]),arr[~arr<4]

写出以上代码的输出结果。

1. 下列代码的运行结果是？

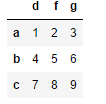
names=np.array([“A","B","C","A"])

data=np.arange(24).reshape(4,6)

print(data[names=="A"])

# 第5章

1. 请运用两种方法构建以下DataFrame。



1. 对于下列的代码，写出输出结果

import pandas as pd

data = {'year': [2000, 2001, 2002, 2003, 2004],

'team': ['John', 'Alice', 'Bob', 'Alex', 'Thomson'],

'score': [9, 8, 9, 7, 10]}

frame = pd.DataFrame(data)

print(frame.reindex([3, 4, 5, 1, 2]))

1. 已知ser1 = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))获取’ahcz’每个字符在ser1中的位置。
2. 若frame = pd.DataFrame({'b': [4, 7, -3, 2], 'a': [0, 1, 0, 1]}).给出frame.sort\_values(by=['a', 'b'])的运行结果。
3. 对于Series

s1=pd.Series([7,2,3],index=['a','b','d'])

s2=pd.Series([5,4,8,10],index=['a','c','d','e'])

写出代码s1+s2的输出。

1. 设有frame=pd.DataFrame(np.arange(12.0).reshape(4,3),

index=['Utah','Ohio','Texas','Oregon'],columns=['b','d','e'])

和series=frame.iloc[0]，请写出执行frame-series后得到的结果。

1. 已知s=pd.Series(['c','a','d','a','a','b','b','c','c'])，请用代码实现将s去重并按字母降序输出。(要求输出结果为Series)
2. 对于下面代码的输出结果是

obj=pd.Series([4,7])

obj.index=[2,4]

obj.reindex(range(5),method='ffill')

1. 写出下列程序的输出结果：

frame=pd.DataFrame(np.arange(8).reshape((2,4)),index=["three",

"one"],columns=["d","a","b","c"])

frame.sort\_index(axis=1,ascending=False)

1. 写出下列代码的运行结果

Frame=pd.DataFrame({‘b’:[4,7,-3,2],’a’:[0,1,0,1]})

Frame.sort\_values(by=[‘a’,’b’])

1. 已知数据如下

data ={'Chinese':[66,95,95,90,80,80],'English':[65,85,92,88,90,90],'Math':[None,98,96,77,90,90]}.请以index=['张飞','关羽','赵云','黄忠','典韦','典韦']，columns=['Chinese','Math','English']建立DataFrame，并删除重复行后，以数学成绩的均值为张飞的数学成绩，最后新增一列为总成绩并计算每个人的总成绩

1. pop = {'Nevada':{2001:2.4,2002:2.9},'Ohio':{2000:1.5,2001:1.7,2002:3.6}}

pd.DataFrame(pop)

1. a = pd.Series([2,0,6,-5,0,6,0])

a.rank()输出结果是：

1. 建立一个以 2018 年每一天为索引，值为随机数的 Series s,并且统计s中每一个周三对应值的和，以及每个月值的平均值
2. 对于下列的代码，写出输出结果

import pandas as pd

data = {'year': [2000, 2001, 2002, 2003, 2004],

'team': ['John', 'Alice', 'Bob', 'Alex', 'Thomson'],

'score': [9, 8, 9, 7, 10]}

frame = pd.DataFrame(data)

print(frame.reindex([3, 4, 5, 1, 2]))

# 第6章

1. 将一个处理好的DataFrame对象data分别输出为.csv文件和.xls文件，其中.csv文件以’|’作为分隔符，文件名称都为’data’
2. 从“http://api.github.com/repos/test” 中获取里面’number’,’title’,’labels’,’state’四列数据，请写出实现上述要求的代码
3. obj ='{"row 1":{"col 1":"a","col 2":"b"},

"row 2":{"col 1":"c","col 2":"d"}}'

data = pd.read\_json(obj)

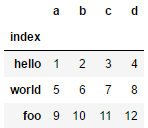
输出的结果是：

1. 设在SQLite数据库中已生成了表test，使用SQLAlchemy连接并读取整个表的数据。
2. 设有路径名为'examples/ex2.csv'的文件，请按每块1000行分块读取并遍历文件，按照其中的‘key’列聚合，获得‘key’列不同元素的计数值。
3. 对于csv文件

1,2,3,4,hello

5,6,7,8,world

9,10,11,12,foo，设计算法读取该文件，以最后一列为索引，按照下列形式输出：



1. 对于位于在C:\test文件夹中的ex1.csv文件，请使用适当的方法将其读取出来，并且要求指定列名为a、b、c，同时要求跳过第一二行数据。
2. 对于一个DataFrame对象Data，请使用适当的方法将其写入到一个位于data.csv文件中，并且以'、'为分隔符，将缺失值标记为'null'。
3. 如何跳过数据文件‘example.csv’的第一行和第三行
4. 如何为数据文件‘example.csv’创建一个分层索引key1,key2
5. 写出以下程序的运行结果

import numpy as np

import pandas as pd

dates=pd.date\_range('1/1/2000',periods=7)

ts=pd.Series(np.arange(7),index=dates)

ts.to\_csv('examples/tseries.csv')

!type examples\tseries.csv

1. 设有路径名为'examples/ex2.csv'的文件，请按每块1000行分块读取并遍历文件，按照其中的‘key’列聚合，获得‘key’列不同元素的计数值。
2. 编写代码，将[['a', 'b', 'c'], ['1', '2', '3'], ['1', '2', '3']]变为{'a': ('1', '1'), 'b': ('2', '2'), 'c': ('3', '3')}。
3. 已知frame = pd.read\_csv('../examples/ex1.csv')，将frame以二进制的格式来存储

# 第7章

1. 写出下列代码的输出

import pandas as pd

from numpy import nan as NA

data = pd.DataFrame({'one': [1, 2, 3, 4, 5],

'two': [6, 7, 8, 9, 10],

'three': [1, 3, 5, 7, 9]},

index = ['a', 'b', 'c', 'd', 'e'])

data.iloc[1: 3, 0] = NA

data.iloc[:3, 1] = NA

data.iloc[1: 2, 2] = NA

print(data.T.dropna(axis = 1, how = 'all').T)

1. 写出下列代码的输出

import pandas as pd

data = pd.DataFrame({'k1': [1, 2, 1, 3, 2, 1],

'k2': [2, 3, 1, 3, 2, 1],

'k3': [3, 1, 2, 2, 1, 3]})

print(data.drop\_duplicates(['k1']))

1. 写出下列代码的输出

import pandas as pd

ages = [11, 23, 17, 29, 30, 10]

bins = [10, 18, 25, 30]

cats = pd.cut(ages, bins)

print(cats)

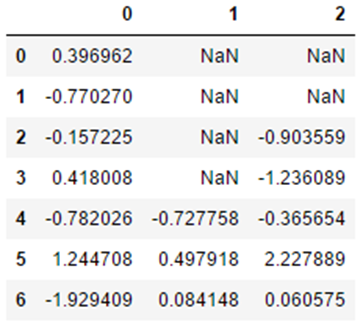
1. 已知ages = [17, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 132]，请将其分到四个箱里：18~25, 26~35, 36~60, >60，并统计每个箱里的元素的个数。
2. 如果text = "foo bar\t baz \tqux，caq"，请用代码得到如下列表

['foo', 'bar', 'baz', 'qux', 'ok']

1. 对于DataFrame对象df=pd.DataFrame(np.arange(20).reshape(5,4))，对其进行随机重排序并抽取3行。
2. 已知data = pd.Series([1., -999., 2., -999., -1000., 3.])，执行data.replace([-999, -1000], [np.nan, 0])后得到的结果是？
3. 设有data = pd.DataFrame(np.arange(12).reshape((3, 4)),index=['ohio', 'colorado', 'indiana'],columns=['one', 'two', 'three', 'four'])

请修改原数据集的索引，使得行索引标签变为只有首字母大写的形式，列索引标签变为全部字母大写的形式。

1. 已知一个用逗号分隔的字符串s='a,b , guido',请用代码实现清除字符串中的空格,并使用'::'作为分隔符将子字符串连接起来。
2. 已知text='Baidu 100085,Alibaba 311121,Tencent 518000',请使用正则表达式找出text中所有邮政编码，并以一个列表的形式返回(中国邮政编码有6位数字且首位不为0）。
3. 假设data=pandas.Series([1,NA,1.0,NA,1])，请过滤掉该数据中的空值并写出输出结果。
4. 已知DataFrame对象df如下：

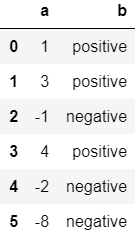


请修改df，用每一列的平均值来补全每一列出现的缺失值

1. 已知data = pd.DataFrame({'food': ['corned beef','Honey ham', 'nova lox'],'ounces': [3, 5, 6]})，meat\_to\_animal = {'corned beef': 'cow','honey ham': 'pig','nova lox': 'salmon'}，实现食物种类匹配输出如下结果



1. 对于一个名为data的DataFrame

请使用map方法将其转换为。

1. 写出下面代码的执行结果：

ages = [17, 18,45, 78, 132]

bins = [18, 25, 35, 60, 100]

pd.cut(ages,bins,right=False).codes

1. 写出下列语块执行后的结果：

data = pd.DataFrame([[1., 6.5, 3.], [1., NA, NA],

[NA, NA, NA], [NA, 6.5, 3.]])

data[4] = NA

data.dropna(axis=1, how='all')

1. 超市举办幸运大转盘活动，转盘被划分为四块区域对应一、二、三等奖，参与奖，区域面积对应8%、16%、24%、52%，求转1000次得各项奖的次数（要求使用本章学的分箱技术）
2. 写出下面程序的输出结果：

data02 = pd.Series([1.,-999.,2.,-999.,-1000,3.])

data02.replace([-999,-1000], [NA, 77],inplace =True)

print(data02)

1. 写出下面程序输出的结果

import pandas as pd

data=pd.Series([1,2,NA,3])

data.fillna(data.mean())

# 第8章

1. 对于下列代码

df = pd.DataFrame({'key':['a','b'],'A':[1,2],'B':[3,4]})

pd.melt(df,['key'])

请写出运行的结果。

1. 对于下列代码

df = pd.DataFrame({'A':[1,2],'B':[3,4]},

index=pd.Index(['a','b'],name='C'))

df.reset\_index()

请写出运行结果。

1. 运行下列代码，

s1 = pd.Series([0,1],index=['a','b'])

s2 = pd.Series([2,3],index=['c','d'])

s3 = pd.concat([s1,s2])

pd.concat([s1,s3],axis=1,join='inner').stack()

请写出输出结果

1. 请写出下列代码的输出

import numpy as np ；import pandas as pd

data = pd.DataFrame(np.arange(15).reshape(3, 5),

index = [['a', 'a', 'b'], [1, 2, 1]])

data.index.names = ['k1', 'k2']

print(data.mean(level = 'k1'))

1. 对下列两个DataFrame对象进行自然连接pd.merge(data1,data2)，请写出自然连接的结果

data1 = pd.DataFrame({'key': ['a', 'b', 'c', 'd'],

'data1': [1, 3, 5, 7]})

data2 = pd.DataFrame({'key': ['a', 'b', 'c', 'd', 'e'],

'data2': [2, 4, 6, 8, 0]})

1. 给出下面代码的运行结果：

df1 = pd.DataFrame({'lkey': ['b', 'b', 'a', 'c', 'a', 'a', 'b'],

'data1': range(7)})

df2 = pd.DataFrame({'rkey': ['a', 'b', 'd'], 'data2': range(3)})

pd.merge(df1, df2, left\_on='lkey', right\_on='rkey')

1. 给出运行结果：

df1 = pd.DataFrame(np.arange(6).reshape(3, 2), index=['a', 'b', 'c'],

columns=['one', 'two'])

df2 = pd.DataFrame(5 + np.arange(4).reshape(2, 2), index=['a', 'c'],

columns=['three', 'four'])

pd.concat([df1, df2], axis=1, keys=['level1', 'level'])

1. 已知df = pd.DataFrame({'key': ['foo', 'bar'],

'A': [1, 3],

'B': [5, 6]})

请将“key”列作为分组指标，用melt函数分组，并给出结果。

1. 对于left=pd.DataFrame({'key':['a','b','a','a','b','c'],

'value':range(6)})

right=pd.DataFrame({'group\_val':[3.5,7]},index=['a','b'])

写出代码

pd.merge(left,right,left\_on='key',right\_index=True,how='outer')的结果。

1. 对于s1=pd.Series([0,1],index=['a','b'])

s2=pd.Series([2,3,4],index=['c','d','e'])

s3=pd.concat([s1,s2])

写出代码pd.concat([s1,s3],axis=1,join='inner')的结果。

1. data=pd.Series(range(7),index=[['a','a','a','b','b','c','c'],[1,2,3,1,2,1,2]])，请写出执行data.unstack()后得到的结果。
2. 已知df1 = pd.DataFrame(np.arange(6).reshape(3, 2), index=['a', 'b', 'c'],columns=['one', 'two'])

df2 = pd.DataFrame(5 + np.arange(4).reshape(2, 2), index=['a', 'c'],

columns=['three', 'four'])

请写出

pd.concat([df1, df2], axis=1, keys=['level1', 'level2'],names=['upper','lower'])

结果。

1. 在豆瓣电影数据中，请为该数据创建一个数据透视表，第一个索引为年代，第二个索引为产地，我们只查看评分一列的数据。（假设数据的dataframe被命名为df）
2. 在数据透视表中，我们可以向aggfunc传入一个字典，对不同值执行不同的函数。请以产地为索引，查看投票人数和评分两列数据，投票人数用求和函数，评分用求平均值的函数。
3. 假设left2 = pd.DataFrame([[1., 2.], [3., 4.], [5., 6.]],

index=['a', 'c', 'e'],

columns=['Ohio', 'Nevada'])

right2 = pd.DataFrame([[7., 8.], [9., 10.], [11., 12.], [13, 14]],

index=['b', 'c', 'd', 'e'],

columns=['Missouri', 'Alabama'])

请运用两种方法，将left2和right2按照索引进行内连接。

1. 下列代码的输出结果是：

import pandas as pd

import numpy as np

df1=pd.DataFrame({'a':[1,np.nan,5,np.nan],'b':[np.nan,2,np.nan,6]})

df2=pd.DataFrame({'a':[np.nan,5,5,np.nan],'b':[np.nan,np.nan,2,6]})

df1.conbine\_first(df2)

1. 对于下列代码

data = pd.Series([1,2,3,4,5,6],

index=[['a','a','b','b','b','c'],[1,2,1,2,3,2]])

data.unstack().T

请使用swaplevel使其输出的结果相同。

1. 下面代码的运行结果是：

data = pd.Series(np.arange(4),index=[[1,2,3,4],[1,1,2,2]])

len(data.unstack().dropna(axis=1))

1. frame = pd.DataFrame({'a': range(7), 'b': range(7, 0, -1),

'c': ['one', 'one', 'one', 'two', 'two',

'two', 'two'],

'd': [0, 1, 2, 0, 1, 2, 3]})

frame2 = frame.set\_index(['c', 'd'])

frame2

输出frame2为：

1. data = pd.Series(np.range(9),

index=[['a', 'a', 'a', 'b', 'b', 'c', 'c', 'd', 'd'],

[1, 2, 3, 1, 3, 1, 2, 2, 3]])

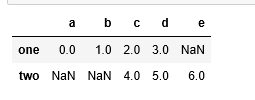
输入data.loc[:, 2]

输出结果为:

1. 写出下面代码的结果：arr =np.arange(12).reshape((3, 4))

np.concatenate([arr, arr], axis=0)

1. data2.unstack()的输出结果为



则data.2.ustack().stack()的输出结果为？

1. df1 = pd.DataFrame({'key1': ['b', 'b'],'key2': ['one', 'two'], 'data1': range(2)})

df2 = pd.DataFrame({'key1': ['a', 'b','d'], 'key2': ['one', 'two','one'], 'data2': range(3)})

pd.merge(df1,df2, on='key1',suffixes=('\_left', '\_right'))

写出运行结果：

1. data如下，写出data.unstack(）的输出：

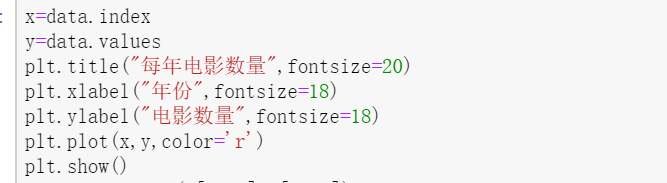


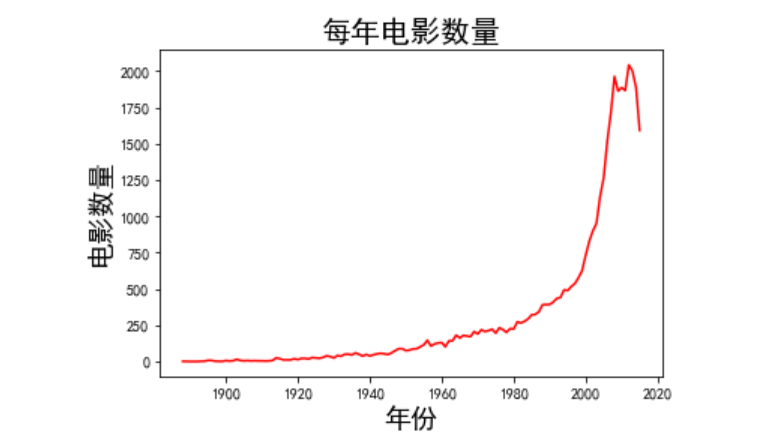
# 第9章

1. 请写出下面这条语句的参数解释

plt.plot(np.random.randn(50).cumsum(), linestyle = '-', color = 'r', marker = 'o')

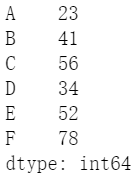
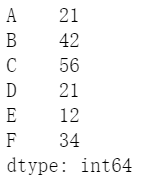
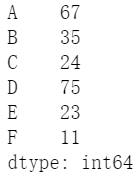
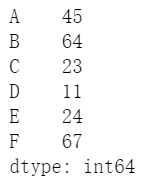
1. 使用pandas绘制一个累计柱状图，data= np.random.rand(10, 4)，要求设置列名为'a', 'b', 'c', 'd'。
2. 画出y=x\*x的图像，要求x的范围是0-6,设置间隔为0.001
3. 使用matplotlib画一个有四个子图的图像，要求每个子图都是直方图，取随机数50个，分5箱，用蓝色填充，所有子图都使用相同的X和Y刻度，间距为0。
4. 使用matplotlib绘制随机漫步的图，要求在位置0, 250, 500, 750, 1000上分别设置标签'one', 'two', 'three', 'four', 'five'
5. 请实现生成一个具有4个子图（2\*2）的Figure对象,并选择在第一行第一列的子图中绘制折线图，样本数据为100个符合正态分布的随机数，线的类型为‘—’，颜色为黑色。
6. 请实现生成一个折线图，样本数据为含有1000个正态分布的随机数的累加值的NumPy数组，并将x轴的刻度4等分，将刻度标签分别设置为'one','two','three','four','five'并逆时针旋转30度，然后为x轴设置名称为'stages'，为图标设置名称为'random'。
7. 设有DataFrame对象tips，其包含列'tip','smoker','day','time','size','tip\_pct'，请用seaborn的分面绘图函数绘制柱状图，按照day，time，smoker来对tip\_pct进行分组显示，并指定day为x轴，根据time在一个面内将不同的柱分组为不同的颜色，根据smoker来对图表分面。
8. 假设源代码执行结果如图：

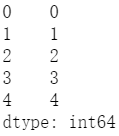


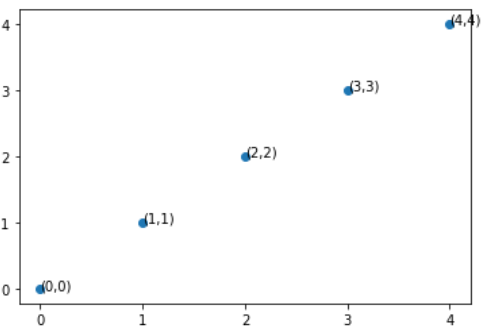


为使图片展示效果更好，请尝试多加几行代码，使得红色线条上每个十年显示一次电影数量，并在2012年该点旁添加箭头标注“2012年达到最大值”，在陡然上升部分添加文字标注“电影数量上升”。

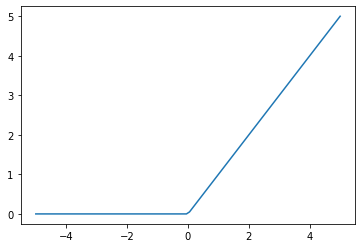
1. 绘制dataframe数据df中评分一列的直方图，设置柱子的个数为20个，黑色描边，透明度为0.5.
2. 请创建下面四个Series对象，并且分别在四个子图中画出柱状图图，要求：四个子图间的间距为0.5，为每个子图设置标题为对象名，data1和data2绘制垂直柱状图，data3和data4绘制水平柱状图。

data1: data2: data3: data4: 

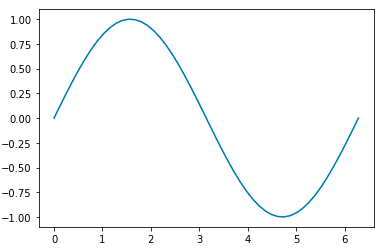
1. 对于一个Series对象Ser,请绘出下面图形。



1. 画出ReLU激活函数relu(x)=max(x,0)



1. 用matplotlib画个简单的正弦曲线图：



1. 对一万个随机生成的标准正态分布的值，画出其直方图（25个bins）和密度图（在同一张图中显示）
2. 画出y=x\*x的图像，要求x的范围是0-6,设置间隔为0.001
3. 画一个矩形，起始点位于(0.2, 0.75)，长为0.4，宽为0.15，不透明度为0.3
4. matplotlib 的 figure 就是一个 单独的 figure 小窗口, 小窗口里面还可以有更多的小图片 （ ）

# 第10章

1. 请写出以下代码的输出

import numpy as np

import pandas as pd

df = pd.DataFrame({'data1': np.arange(6),

'data2': np.arange(0, 12, 2),

'key': ['a', 'b', 'b', 'c', 'a', 'a']})

print(df.groupby('key').mean())

1. 现在存在一个DataFrame对象

data = pd.DataFrame(

{'part 1': [84, 82, 84, 92, 85, 75, 82, 85, 86],

'part 2': [91, 90, 93, 93, 89, 90, 88, 89, 94],

'part 3': [93, 91, 90, 89, 88, 92, 91, 87,90]},

index = [['A', 'A', 'A', 'B', 'B',3 'B', 'C', 'C', 'C'],

['John', 'Bob', 'Alice', 'Tony', 'Micheal','Jackson','James', 'Paul', 'Wiliiam']])

data.index.names=['group', 'name']

(1)进行groupby操作，按照’group’进行聚合，输出每一组’part 1’,’part 2’, ‘part 3’的平均值

(2)按照’name’进行聚合，输出每个人的平均分

(3)输出最高分的那个人的信息

1. 已知people = pd.DataFrame(np.arange(0,25).reshape(5, 5),columns=['a', 'b', 'c', 'd', 'e'],index=['Joe', 'Steve', 'Wes', 'Jim', 'Travis'])，people.groupby(len).sum()返回的结果是?
2. df = pd.DataFrame({'category': ['a', 'a', 'a', 'a',

'b', 'b', 'b', 'b'],

'data': np.random.randn(8),

'weights': np.random.rand(8)})

根据category进行分组然后按照weights的权值计算加权平均数：

1. 已知样本数据states=['Ohio','New York','Vermont','Florida',

'Oregon','Nevada','Califonia','Idaho']

group\_key=['East']\*4+['West']\*4

data=pd.Series(np.random.randn(8),index=states)

data[['Vermont','Nevada','Idaho']]=np.nan

对于data中的缺失值，将’East’分组的值替换为0.5，将’West’分组的值替换为-1。

1. 已知df = pd.DataFrame({'key1' : ['a', 'a', 'b', 'b', 'a'],

'key2' : ['one', 'two', 'one', 'two', 'one'],

'data1' : np.random.randn(5),

'data2' : np.random.randn(5)})

请根据key1和key2作为标签对数据进行分组并计算data1列的均值。

1. 设有数据集data:



请写出执行pd.crosstab(data.Nationality,data.Handedness,margins=True)后得到的输出。

1. 假设有df = pd.DataFrame({'key1' : ['a', 'a', 'b', 'b', 'a'],

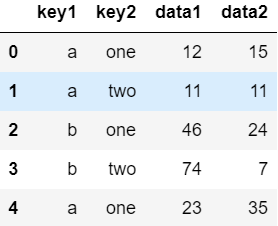
'key2' : ['one', 'two', 'one', 'two', 'one'],

'data1' : np.random.randn(5),

'data2' : np.random.randn(5)})

根据key1进行分组，并输出每个分组的每个列中平均数与中位数的差值。

1. 对于下图的DataFrame对象data



请将其创建出来，并且根据key1,key2进行分组，然后在data1列最大值，data2列求最小值。

1. 编写一个自定义函数，要求输出每组的最小值，最大值，均值以及每组的个数。
2. 下面代码输出的结果为？

df = pd.DataFrame({'key1' : ['a', 'a', 'b', 'b', 'a'],

'key2' : ['one', 'two', 'one', 'two', 'one'],

'data1' :[1,2,3,4,5],

'data2' : [4,1,5,2,3]})

df.pivot\_table('data1',index=['key1'],columns='key2',aggfunc='mean')

1. 下面的代码输出的结果是？

def set\_group(string):

if string[0].isdigit3():

return 'a'

else:

return 'b'

df = pd.DataFrame({'data1':range(5),'data2':range(6,11)},index = ['1a','bb','cc','3d','4e'])

df.groupby(set\_group).sum()

1. 下面代码的运行结果是：

mapping = {'a': 2 , 'b': 4}

data = pd.DataFrame([[2, 2, 4 ,4],[1, 2, 4 ,8]], columns=['a', 'a', 'b', 'b'])

data.groupby(mapping,axis=1).sum()

# 第11章

1. from datetime import datetime

import numpy as np

dates = [datetime(2011, 1, 2), datetime(2011, 1, 5),

datetime(2011, 1, 7), datetime(2011, 1, 8),

datetime(2011, 1, 10), datetime(2011, 1, 12)]

ts = pd.Series(np.arange(6), index=dates)

ts,给出运行结果。

1. datas=pd.date\_range('1/1/2019', periods=100, freq='W')

date= pd.Series(np.arange(100),index=datas)

date.truncate(before='1/27/2019',after='3/3/2019')

给出运行结果。

1. rng = pd.date\_range('1/1/2019', periods=10, freq='W')

t = pd.Series(np.arange(11,21), index=rng)

t.to\_period('M')

给出运行结果。

1. 已知Series

rng=pd.date\_range('2000-01-01',periods=12,freq='T')

ts=pd.Series(np.arange(12),index=rng)

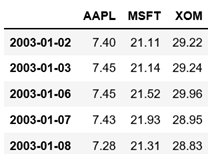
计算每一组的加和，将数据聚合到五分钟的块内，使用右箱体边界标记时间序列。

1. 已知带有每周数据的DataFrame

frame=pd.DataFrame(np.random.randn(2,4), index=pd.date\_range('1/1/2000',periods=2,freq='W-WED'), columns=['Colorado','Texas','New York','Ohio'])

将其转换为每天的数据，并在非星期三的日期上向前填充每周数值。

1. 设有数据集close\_px，前五行为



请计算出苹果公司（APPL）股票价格(AAPL)的100日移动窗口平均值，并绘制成折线图。

1. 已知一份记录用户浏览网站的信息的数据集records，采用json格式组织数据，其中‘tz’键值对表示的是用户所在的时区（某些记录中‘tz’键值对可能存在空值）。其中一条记录为



请按照每个时区生成计数。

1. ts=pd.Series(np.arange(5),index=pd.date\_range('7/10/2019',periods=5,freq='M'))

写出ts.shift(1,freq='M')的输出结果

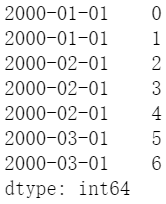
1. 请写出以下代码的运行结果：

p = pd.Period('2007', freq='A-JUN')

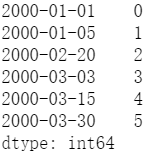
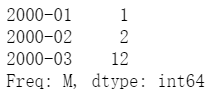
（1）p.asfreq('D', 'end')

（2）p.asfreq('M', 'start')

1. 获取当前日期，按照'2000.01.01-24:00:00'的格式输出。
2. 对于下面一个含有重复索引的时间序列，请计算相同索引对应的平均值。



1. 对于一个ser对象

，请将其转换为。

1. 写出下面代码的结果

ts=pd.Series(np.arange(5),index=pd.date\_range('15/11/2019',periods=5,freq='M'))

ts.shift(2,freq='D')

ts

1. 设计程序计算指定年份的第n个周m是几月几号，例如2019年第11个周周一是几月几号？
2. ts =pd.Series(np.arange(4),index =

pd.date\_range('1/1/2000',periods=4,freq='M'))

ts.shift(2,freq='M')

写出ts的结果。

1. datas=pd.date\_range('1/1/2019', periods=100, freq='W')

date= pd.Series(np.arange(100),index=datas)

date.truncate(before='1/27/2019',after='3/3/2019')

给出运行结果。

1. ts = pd.Series(range(1,20,2),index = pd.date\_range('2019-10-01',periods = 10,freq = '10d'))

ts.resample('M').mean()

给出运行结果：

# 第12章

1. 考虑如下DataFrame对象：data = pd.DataFrame({'key': ['a', 'b', 'c'] \* 4,'value': np.arange(12.)})

写出获得如下输出的代码：

0 0.0

1 1.0

2 4.0

3 9.0

4 16.0

5 25.0

6 36.0

7 49.0

8 64.0

9 81.0

10 100.0

11 121.0

Name: value, dtype: float64

其中，第二列为该对象’value’值的平方

1. import pandas as pd

categories = ['foo', 'bar', 'baz']

codes = [0, 1, 2, 0, 0, 1]

my\_cats\_2 = pd.Categorical.from\_codes(codes,categories,ordered=True)

my\_cats\_2，给出运行结果。

1. 给出下列代码运行结果：

import pandas as pd

import numpy as np

ts = pd.date\_range('1/1/2019', periods=5, freq='M')

df = pd.DataFrame({'num':np.arange(len(ts))},index=ts)

df.groupby(pd.TimeGrouper(freq='2M')).sum()

1. 已知时间数据集

N=15

times=pd.date\_range('2017-05-20 00:00',freq='1min',periods=N)

df=pd.DataFrame({'time':times.repeat(3),

'key':np.tile(['a','b','c'],N),

'value':np.arange(N\*3.)})

为每个’key’的值按照time进行索引，然后每5分钟进行采样，并将value值聚合。

1. 已知随机数组

draws=np.random.randn(1000)

请计算其带名称的四分位分箱，并使用groupby方法来提取每个分箱数据的数量与平均值。

1. 已知函数调用序列

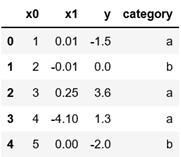
a=f(df,arg1=v1)

b=g(a,v2,arg3=v3)

c=h(b,arg4=v4)

请调用pipe方法重写代码进行方法链接。

1. 已知有数据集data=pd.DataFrame({'x0':range(1,6),'x1':[0.01,-0.01,0.25,-4.10,0.00],'y':[-1.5,0.0,3.6,1.3,-2.0], 'category': ['a','b','a','a','b']})如图所示



请用‘category’列生成虚拟变量去替代‘category’列，虚拟变量前缀设置为‘category’。

1. np.random.seed(12345)

draws=np.random.randn(1000)

bins=pd.qcut(draws,4,labels=['Q1','Q2','Q3','Q4'])

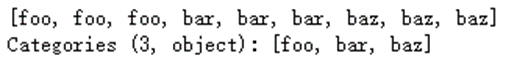
被标记的bins分类数据并不包含箱体边界信息，请尝试使用grounpby来提取一些汇总数值，每一部分的总数，最小边界，最大边界。

1. values=pd.Series([0,0,1]\*2)

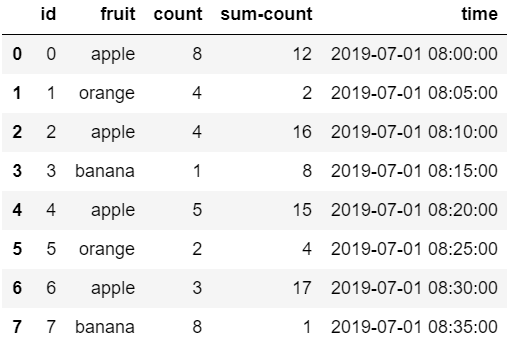
dim=pd.Series(['地铁','高铁'])

写出dim.take(values)的输出结果

1. 假设categories = ['foo', 'bar', 'baz']，请指定顺序生成以下categories数组，如下图所示：



1. 对于下图的DataFrame对象df



请计算每种水果每隔15分钟的count的总和。

1. 已知df = pd.DataFrame({'key': ['a', 'b', 'c']\*4, 'value': np.arange(12.)})，要求得到一个Series，它的尺寸和df['value']一样，但值都被按照'key'分组的均值替代。
2. df = pd.DataFrame({'key': ['a', 'b', 'c']\* 2,

'value': np.arange(6.)})

df['value'].groupby(df['key']).transform('mean')

代码的输出为：

1. 写出输出结果

s = pd.Series(['a','b','c','d']\*2)

cat\_s = s.astype('category')

cat\_s.value\_counts()

1. df = pd.DataFrame({'name': ['Alice', 'Bob', 'Charlie', 'Danielle'],

'gender': ['Female', 'Male', 'Male', 'Female'],

'balance': [100.0, 200.0, 300.0, 400.0],})

将性别属性转换成category对象，并指定Male编码为0，Female编码为1.

1. 给出下面代码的运行结果：

import pandas as pd

import numpy as np

ts = pd.date\_range('1/1/2019', periods=5, freq='M')

df = pd.DataFrame({'num':np.arange(len(ts))},index=ts)

df.groupby(pd.TimeGrouper(freq='2M')).sum()

1. 考虑如下DataFrame对象：data = pd.DataFrame({'key':

['a', 'b', 'c'] \* 4, 'value': np.arange(12.)})

G=data.groupby(‘key’).value

G.transform(lambda x:x.rank(ascending=False))结果为？