Pandas基础命令速查表

数据的选取

（1）df.columns = ['a','b'] # 重命名数据框的列名称

（2）pd.isnull() # 检查数据中空值出现的情况，并返回一个由布尔值(True,Fale)组成的列

（3）pd.notnull() # 检查数据中非空值出现的情况，并返回一个由布尔值(True,False)组成的列

（4）df.dropna() # 移除数据框 DataFrame 中包含空值的行

（5）df.dropna(axis=1) # 移除数据框 DataFrame 中包含空值的列

（6）df.fillna(x) # 将数据框 DataFrame 中的所有空值替换为 x

（7）s.fillna(s.mean()) -> 将所有空值替换为平均值

（8）s.astype(float) # 将数组(Series)的格式转化为浮点数

（9）s.replace(1,'one') # 将数组(Series)中的所有1替换为'one'

（10）s.replace([1,3],['one','three']) # 将数组(Series)中所有的1替换为'one', 所有的3替换为'three'

（11）df.rename(columns=lambda x: x + 2) # 将全体列重命名

（12）df.rename(columns={'old_name': 'new_name'}) # 将选择的列重命名

（13）df.rename(index = lambda x: x+ 1) # 改变全体索引

（14）df.set_index('column_one') # 改变索引 ，指定某一列的值当整个DataFrame的索引

数据的过滤(filter),排序(sort)和分组(groupby)

filter（1）df[df['col'] > 0.5] # 以col列为标准，选取数值大于0.5的行为True，然后从df中选取这些行

（2）df[(df['col'] > 0.5) & (df['col'] < 0.7)] # 选取数据框df中对应行（col列>0.5,col<0.7)

sort（1）df.sort_values(col1) # 按照数据框的列col1升序(ascending)的方式对数据框df做排序

（2）df.sort_values(col2,ascending=False) # 按照数据框的列col2降序(descending)的方式对数据框df做排序

（3）df.sort_values([col1,col2],ascending=[True,False]) # 按照数据框的列col1升序，col2降序的方式对数据框df做排序

groupby
（1）df.groupby(col) # 按照某列对数据框df做分组

（2）df.groupby('A').count() #根据A列分组，并计数

（3）df.groupby([col1,col2]) # 按照列col1和col2对数据框df做分组

（4）df.pivot_table(index=col1,values=[col2,col3],aggfunc=mean) # 做透视表，索引为col1,针对的数值列为col2和col3，aggfunc为分组函数，默认为mean,可以设置np.sum等

apply（1）df.apply(np.mean) # 对数据框df的每一列求平均值

（2）df.apply(np.max,axis=1) # 对数据框df的每一行求最大值 #axis=0是指沿着行轴方向，即从上到下的列方向，axis=1是指沿着列轴方向，即从左到右的方向

数据的连接与组合

（1）df1.append(df2) # 在数据框df2的末尾添加数据框df1，其中df1和df2的列数应该相等

（2）pd.concat([df1,df2]) # 不指定axis时默认为0，沿着行轴方向增加，列数应该相同与append一样

（3）pd.concat([df1, df2],axis=1) # 在数据框df1的列最后添加数据框df2,其中df1和df2的行数应该相等

（4）df1.join(df2,on=col1,how='inner') # 对数据框df1和df2做内连接，其中连接的列为col1

（5）pd.merge(df1,df2,left_on='key',right_on='key') #索引上的合并

数据的统计

（1）df.describe() # 得到数据框df每一列的描述性统计

（2）df.mean() # 得到数据框df中每一列的平均值

（3）df.corr() # 得到数据框df中每一列与其他列的相关系数

（4）df.count() # 得到数据框df中每一列的非空值个数

（5）df.max() # 得到数据框df中每一列的最大值

（6）df.min() # 得到数据框df中每一列的最小值

（7）df.median() # 得到数据框df中每一列的中位数

（8）df.std() # 得到数据框df中每一列的标准差

数据的导入

(1)pd.read_csv(filename) # 导入csv格式文件中的数据

(2)pd.read_table(filename) # 导入有分隔符的文本 (如TSV) 中的数据

(3)pd.read_excel(filename) # 导入Excel格式文件中的数据

(4)pd.read_sql(query, connection_object) # 导入SQL数据表/数据库中的数据

(5)pd.read_json(json_string) # 导入JSON格式的字符，URL地址或者文件中的数据

(6)pd.read_html(url) # 导入经过解析的URL地址中包含的数据框 (DataFrame) 数据

(7)pd.read_clipboard() # 导入系统粘贴板里面的数据

(8)pd.DataFrame(dict) # 导入Python字典 (dict) 里面的数据，其中key是数据框的表头，value是数据框的内容。

(9)pd.read_csv('data.csv',index_col=0) #不读入csv文件中 “Unnamed:0” 列

数据的导出

(1)df.to_csv(filename) # 将数据框
(DataFrame)中的数据导入csv格式的文件中

(2)df.to_excel(filename) # 将数据框 (DataFrame)中的数据导入Excel格式的文件中

(3)df.to_sql(table_name,connection_object) # 将数据框 (DataFrame)中的数据导入SQL数据表/数据库中

(4)df.to_json(filename) # 将数据框 (DataFrame)中的数据导入JSON格式的文件中

(5)df.to_csv('result.csv',index=False) #不再写入默认的一序列

创建测试对象

（1）创建一个DataFrame对象pd.DataFrame(np.random.rand(10,5)) # 创建一个5列10行的由随机浮点数
成的数据框 DataFrame

（2）创建一个Series对象pd.Series(my_list) # 从一个可迭代的对象 my_list 中创建一个数据组

（3）从dict创建
pd.DataFrame({"label":labels,"res:",res})

数据的查看与检查

（1）df.head(n) # 查看数据框的前n行

（2）df.tail(n) # 查看数据框的最后n行

（3）df.shape # 查看数据框的行数与列数 shape是该对象的一个属性值，不是函数

（4）df.info() # 查看数据框 (DataFrame) 的索引、数据类型及内存信息

（5）df.describe() # 对于数据类型为数值型的列，查询其描述性统计的内容

（6）s.value_counts(dropna=False) # s是一个Series对象，查询每个独特数据值出现次数统计

（7）df.apply(pd.Series.value_counts) # 统计df中每一个数字在每一列中出现的次数

数据的选取

（1）df[col] # 以数组 Series 的形式返回选取的列，col是列名

（2）df[[col1, col2]] # 以新的数据框(DataFrame)的形式返回选取的列

（3）df.loc[0] # 选取第一行,没有设置index时，第一行index默认为0

（4）df.ix['a'] # 选取索引为a的那一行

（5）df.iloc[0,:] # 选取第一行

（6）df.iloc[0,0] # 选取第一行第一个元素

（7）s.iloc[0] # 按照位置选取

（8）s.loc['index_one'] # 按照索引选取 'index_one'是指第一个

（9）位置的index名，没有设置index时，inde默认为0

Made By
zhangqizky