Pandas-1

2020/6/24 10:08

2.1 查看DataFrame：

df.values # 查看所有元素 **df.index # 查看索引 df.columns # 查看所有列名 df.dtype # 查看字段类型** df.size # 元素总数 **df.ndim # 表的维度数 df.shape # 返回表的行数与列数** df.info # DataFrame的详细内容 df.T # 表转置

2.2 查看访问DataFrame中的数据

2.2.1 基本查看方式:

单列数据：df['col1'] 单列多行：df['col1'][2:7] 多列多行：df[['col1','col2']][2:7] **多行数据：df[:][2:7] 前几行： df.head() 后几行： df.tail()**

2.2.2 **loc，iloc**的查看方式（大多数时候建议用loc）

loc[行索引名称或条件，列索引名称] # iloc[行索引位置，列索引位置] 单列切片：df.loc[:,'col1'] df.iloc[:,3] 多列切片：df.loc[:,['col1','col2']] df.iloc[:,[1,3]] 花式切片：df.loc[2:5,['col1','col2']] df.iloc[2:5,[1,3]] **条件切片：df.loc[df['col1']=='245',['col1','col2']] df.iloc[(df['col1']=='245').values,[1,5]]**

2.3 更改某个字段的数据：

df.loc[df['col1']=='258','col1']=214 # 注意：数据更改的操作无法撤销，更改前最好对条件进行确认或者备份数据

2.4 增加一列数据：

df['col2'] = 计算公式/常量

2.5 删除数据：

删除某几行数据,inplace为True时在源数据上删除，False时需要新增数据集 df.drop(labels=range(1,11),axis=0,inplace=True) # 删除某几列数据 df.drop(labels=['col1','col2'],axis=1,inplace=True)

3. DataFrame的描述分析

#数值型： **df[['col1','col2']].describe() #类别型：** df['col1'].**value_counts**()[0:10] #category型： df['col1'] = df['col1'].**astype**('category') df['col1'].describe()

4. 处理时间序列数据

4.1 转换字符串时间为标准时间：

df['time'] = pd.to_datetime(df['time'])

4.2 提取时间序列信息

year = df['time'].year() # year-年，month-月，day-天，hour-小时，minute-分钟，second-秒，date-日期，time-时间 week-一年中第几周，quarter-季节，dayofweek-一周中第几天，weekday_name-星期名称

4.3 加减时间：

使用Timedelta,支持weeks，days，hours，minutes,seconds，但不支持月和年 df['time'] = df['time'] + pd.Timedelta(days=1) df['time'] = df['time'] - pd.to_datetime('2016-1-1') # 时间跨度计算： df['time'].max() - df['time'].min()

5. 使用分组聚合

5.1 使用**groupby拆分数据并计算**：

df.groupby(by='',axis=0,level=None,as_index=True,sort=True,group_keys=True,squeeze=False).count() # by--分组的字段 level--标签所在级别，默认None as_index--聚合标签是否以df形式输出，默认True， sort--是否对分组依据，分组标签进行排序，默认True group_keys--是否显示分组标签名称，默认True squeeze--是否对返回数据进行降维，默认False # 聚合函数有count， head,max,min,median,size,std,sum

5.2 使用**agg聚合数据**：

求出当前数据的统计量 df[['col1','col2']].agg([np.mean,np.sum]) **# 分别求字段的不同统计量 df.agg({'col1':np.sum,'col2':np.mean}) # 求不同字段不同数目的统计量 df.agg({'col1':np.sum,'col2':[np.mean,np.sum]})**

5.3 使用transform聚合数据：

实现组内数据离差标准化 dfgroup.transform(lambda x:(x.mean()-x.min())/(x.max()-x,min()))

5.4 重新索引

reindex(index=None, columns=None,...)方法 可改变或重排Series和DataFrame索引 reindex(index=None, columns=None,...)

index, colums 新的行列自定义索引

fill_value 在重新索引，用于填充缺失位置的值

method 填充方法，ffill当前值向前填充， bfill向后填充

limit 最大填充量

copy 默认为True，生成新的对象，False时，新旧相等，但不复制

d.reindex(index = [‘d’ , ‘c’ , ‘b’ , ‘a’])

d.reindex(colums = [‘two’ , ‘one’])

append(idx) 连接另外一个Index对象，产生新的Index对象

.diff(idx) 计算差集，产生新的Index对象

.intersection（idx） 计算交集，产生新对象

.union（idx） 计算并集

.delete（loc） 删除loc位置处的元素

.insert（loc, e） 在loc位置增加一各元素e

5.5 数据排序

.sort_index()方法在指定轴上根据索引进行排序，默认升序。

.sort_index（axis=0,ascending = True） ascending是指递增排序

.sort_values()方法在指定轴上根据数值进行排序，默认升序。

DataFrame.sort_values(by= ‘##’ ,axis=0,ascending=True, inplace=False, na_position=‘last’)

参数说明