BDA_Ch3

2019 年 10 月 30 日

In [ ]:
```
# 首先需要引入 pandas 及相关的包
from pandas import Series

arr = [1,2,3,4,5]# 创建列表
series1 = Series(arr)# 通过列表创建 Series
# 遍历 Series
for index, val in series1.items():
    print(index, val)
series1[1] = 'two'# 根据位置下标修改值
series1.index = ['a','b','c','d','e']# 修改索引标签
series1['d'] = 'four'# 根据标签下表修改值
series1.name = 'column_name'
# 与字典不同，这里的索引标签可以重复
series1.index = ['a','b','c','d','c']
# 使用 rename 方法进行重命名，inplace = True：不创建新的对象，直接对原始对象进行修改
series1.rename('new_name',inplace = True)
print(series1['c'])# 列出所有索引标签为'c'的元素
print(series1[['a','d']])# 可以列多个标签，但是要用列表形式
print(series1[[1,2,4]])# 通过位置来进行检索
print(series1[1:4])
# 通过 append() 添加 Series，注意一定要有赋值操作
series1 = series1.append(Series([6],index=['f']))
series1.rename(index={'f':'e'},inplace = True)# 更改行标签
#drop() 删除索引标签为'a'的元素并创建一个新的对象，所以需要使用 inplace = True
series1.drop('a', inplace = True)
del series1['b']#del 命令直接修改原对象
print(Series({'aa':1,'bb':2}))# 也可以通过字典来创建 Series
```

In [ ]:
```
from pandas import DataFrame
import pandas as pd
import numpy as np

#np.nan 表示缺失值
df1 = DataFrame({'From_To': [' LoNDon_paris ', ' MAdrid_miLAN ', ' londON_StockhOlm '
                             ,'Budapest_PaRis', 'Brussels_londOn'],
                 'FlightNumber': [10045, np.nan, 10065, np.nan, 10085],
                 'RecentDelays': [[23, 47], [], [24, 43, 87], [13], [67, 32]],
                 'Airline': ['KLM(!)', '<Air France> (12)', '(British Airways. )'
                             ,'12. Air France', '"Swiss Air"']})
# 按行遍历 DataFrame，按列遍历则为 iteritems()
for index, row in df1.iterrows():
    print(row['From_To'])
#loc 进行数据查询，其中的参数均为索引
df1.loc[0,'Airline']
```

```
# 与 loc 不同，iloc 中的参数均为数字，表示二维表中的位置，-1 表示最后一位
df1.iloc[0,0]
# 切片略有不同
#iloc 中 1:3 不包括 3，loc 中的 1:3 包括 3
df1.iloc[1:3,0]
df1.loc[1:3,'Airline']
# 使用 loc 来进行切片选择一部分数据
#1:3 表示从行索引为 1 开始到行索引为 3，('From_To','Airline') 表示选择 From_To 和 Airline 这两列
df1.loc[1:3,('From_To','Airline')]
# 使用 iloc 的话
df1.iloc[1:3,[0,3]]
# 使用 loc 来切片选择 From_To 这一列，这里 () 可以省略
#df1.From_To 表示 df1 的 From_To 这一列，str.upper() 为所有的字母变为大写
df1.loc[:,'From_To'] = df1.From_To.str.upper()
df1.loc[:,'From_To'] = df1.From_To.str.lower()# 改为小写
# 剔除出 Airline 中的噪音数据
df1['Airline'] = df1.Airline.str.extract('([a-zA-Z\s]+)')
# 将出发点和目的地分开成两列
# 注意这里的 expand 参数，如果为 True 则分割后成为多列
# 如果为 False 则分割后仍然为一列，用一组列表来表示多个地点
df1.From_To.str.split('_', expand=True)
# 上述语句仅仅是生成一个新的 DataFrame 对象，并不修改 df1 本身
# 需要通过 join 来拼接两个 DataFrame，并将拼接后新生成的 DataFrame 赋值给 df1
df1 = df1.join(df1.From_To.str.split('_', expand=True))
# 对列进行重命名
# 如果没有标注 columns，默认为对行索引进行重命名
df1.rename(columns={0:'From',1:'To'},inplace=True)
# 删除 From_To 这一列，同样的，如果没有标注 columns，默认为删除'From_To'这一行
# 当然，因为没有这样一行，所以会报错
df1.drop(columns=['From_To'],inplace=True)
# 添加行
# 如果不指定行索引的话，必须使用 ignore_index=True
df1 = df1.append({'FlightNumber':'MF8169','RecentDelays':[],'Airline':'Xiamen Airline'
                  ,'From':'Xiamen','To':'Beijing'},ignore_index=True)
# 或者用如下方式指定行索引
# 此处需要先通过 DataFrame() 创建一个 DataFrame，然后再使用 append
df1 = df1.append(DataFrame([{'FlightNumber':'MF8159','RecentDelays':[],
                             'Airline':'Xiamen Airline','From':'Xiamen'
                             ,'To':'Beijing'}],index=[8]))
```

In [ ]:
```
import pandas as pd

#header 表示列名所在的行，如果没有列名则为 None
#sheet_name 表示要读取的 Sheet，可以是具体名字，也可以是第几张 Sheet
data = pd.read_excel('test.xlsx', sheet_name = 0, header = 0)
print(data)
# 查看前 5 行数据
print(data.head(5))
# 查看最后 5 行数据
print(data.tail(5))
# 查看简单统计数据
data.describe()
# 数据的筛选
```

```
data[data.日开盘价>10]
data[data.公司简称.str.startswith('万')]
data[data.公司简称.str.contains('A')]
# 列级别的操作
data['new_col1'] = data['日最高价'] - data['日最低价']
data['new_col2'] = data['日最高价'] * data['日最低价']

data.rename(columns={0:'stock_id',1:'company'}, inplace=True)
writer = pd.ExcelWriter('out.xlsx')
data.to_excel(writer, 'sheet1', header = True, index = False)
writer.save()
```