昵称：Treant
园龄：5年4个月
粉丝：197
关注：78
+加关注

【Python实战】Pandas：让你像写SQL一样做数据分析（二）

1. 引言

前一篇介绍了Pandas实现简单的SQL操作，本篇中将主要介绍一些相对复杂一点的操作。为了方便后面实操，先给出一份简化版的设备统计数据：

```
0   android NLL 387546520   2099457911
0   ios NLL 52877990    916421755
1   android 魅族  8995958 120369597
1   android 酷派  9915906 200818172
1   android 三星  16500493    718969514
1   android 小米  23933856    290787590
1   android 华为  26706736    641907761
1   ios 苹果  52877990    916421755
2   android 小米-小米4  2786675 55376581
2   android 魅族-m2-note  4642112 130984205
2   android OPPO-A31    4893428 62976997
2   ios 苹果-iPhone-6s    5728609 99948716
```

其中，第一列表示维度组合编号，第二列表示操作系统类型，第三列为维度值（NLL表示缺失，即第一行、第二行表示操作系统的统计，其余表示厂商或机型），第三列、第四列分别表示UV、PV；且字段之间为 `\t` 分隔。读取该文件为DataFrame：

```
import pandas as pd


df = pd.read_csv(path, names=['id', 'os', 'dim', 'uv', 'pv'],
sep='\t')
```

2. 实战

Add

在原dataframe上，增加一行数据；可通过dataframe的append函数来追加：

```
import numpy as np
row_df = pd.DataFrame(np.array([['2', 'ios', '苹果-iPad 4', 3287509,
32891811]]), columns=['id', 'os', 'dim', 'uv', 'pv'])
df = df.append(row_df, ignore_index=True)
```

增加一列数据，则比较简单：

```
df['time'] = '2016-07-19'
```

To Dict

关于android、ios的PV、UV的dict：

```
def where(df, column_name, id_value):
    df = df[df[column_name] == id_value]
    return df
```

Coder 无他，但手熟尔


```
    """
    {"pv" or "uv" -> {"os": os_value}}
    :return: dict
    """
    df = where(df, 'id', 0)
    df_dict = df.set_index('os')[['uv', 'pv']].to_dict()
    return df_dict
```

Top

group某列后的top值，比如，android、ios的UV top 2的厂商：

```
def group_top(df, group_col, sort_col, top_n):
    """
    get top(`sort_col`) after group by `group_col`
    :param df: dataframe
    :param group_col: string, column name
    :param sort_col: string, column name
    :param top_n: int
    :return: dataframe
    """
    return df.assign(rn=df.sort_values([sort_col], ascending=False)
                     .groupby(group_col)
                     .cumcount() + 1) \
        .query('rn < ' + str(top_n + 1)) \
        .sort_values([group_col, 'rn'])
```

全局top值加上group某列后的top值，并有去重：

```
def top(df, group_col, sort_col, top_n):
    """overall top and group top"""
    all_top_df = df.nlargest(top_n, columns=sort_col)
    grouped_top_df = group_top(df, group_col, sort_col, top_n)
    grouped_top_df = grouped_top_df.ix[:, 0:-1]
    result_df = pd.concat([all_top_df,
grouped_top_df]).drop_duplicates()
    return result_df
```

排序编号

对某列排序后并编号，相当于给出排序名次。比如，对UV的排序编号：

```
df['rank'] = df['uv'].rank(method='first',
ascending=False).apply(lambda x: int(x))
```

Left Join

Pandas的left join对NULL的列没有指定默认值，下面给出简单的实现：

```
def left_join(left, right, on, right_col, default_value):
    df = pd.merge(left, right, how='left', on=on)
    df[right_col] = df[right_col].map(lambda x: default_value if
pd.isnull(x) else x)
    return df
```

自定义

对某一列做较为复杂的自定义操作，比如，厂商的UV占比：

```
def percentage(part, whole):
    return round(100*float(part)/float(whole), 2)

os_dict = to_dict(df)
all_uv = sum(os_dict['uv'].values())
```

重复值

某列的重复值的行：

```
duplicate = df.duplicated(subset=columns, keep=False)
```

写MySQL

Pandas的to_sql函数支持Dataframe直接写MySQL数据库。在公司开发时，常常会有办公网与研发网是不通的，Python的sshtunnel模块提供ssh通道，便于入库debug。

```
import MySQLdb
from sshtunnel import SSHTunnelForwarder


with SSHTunnelForwarder(('porxy host', port),
                        ssh_password='os passwd',
                        ssh_username='os user name',
                        remote_bind_address=('mysql host', 3306)) as
server:
    conn = MySQLdb.connect(host="127.0.0.1", user="mysql user name",
passwd="mysql passwd",
                           db="db name",
port=server.local_bind_port, charset='utf8')
    df.to_sql(name='tb name', con=conn, flavor='mysql',
if_exists='append', index=False)
```


分类: Python

Treant
关注 - 78
粉丝 - 197
+加关注

0 1

« 上一篇：Scala比较器：Ordered与Ordering
» 下一篇：MySQL常用SQL总结

posted @ 2016-07-19 20:10 Treant 阅读(3407) 评论(0) 编辑 收藏

Coder 无他，但手熟尔

最新IT新闻:
· 阿里领投11亿美金入股印尼最大电商
· 爱立信或在瑞典以外裁员2.5万人 超1/5员工会被裁
· 福特自动驾驶汽车专利 方向盘和踏板都可拆卸
· 印度核查中国品牌手机 因担心用户数据有安全隐患
· 苹果CEO库克写备忘录谴责白人至上主义
» 更多新闻...

最新知识库文章:
· 做到这一点，你也可以成为优秀的程序员
· 写给立志做码农的大学生
· 架构腐化之谜
· 学会思考，而不只是编程
· 编写Shell脚本的最佳实践
» 更多知识库文章...