

Python 应用之一：基本财务分析

一、结果

每一种产品在某一周的工作日店均日均销售数量、销售成本、销售收入、销售毛利和销售毛利率

```
Out[643]:
```

		avgSkuCnt	avgPayPriceTotal	avgCost	avgProfit	avgProfitRate
weekNum	sku_name					
17	...	21.60	181.37	89.44	91.92	50.68%
	...	25.02	269.68	148.12	121.57	45.08%
18	...	15.87	136.11	61.57	74.54	54.76%
	...	19.34	205.11	111.02	94.08	45.87%
19	...	16.57	141.01	65.23	75.78	53.74%
	...	20.70	222.75	118.98	103.77	46.59%
20	...	11.58	101.87	45.57	56.30	55.27%
	...	11.94	122.84	68.59	54.25	44.16%
21	...	7.62	78.24	27.34	50.91	65.06%
	...	6.75	58.42	26.57	31.85	54.53%

二、数据源

- 商品销售明细，包含字段：订单日期，商品名称，订单价格，商品销售数量，商品成本等

```
In [672]: dataSource.head()
```

```
Out[672]:
```

	b.fm	cost_price	cost_tax	cost_total	order_date	order_hour	payable_price	payable_price_total	sell_price	sku_cnt	sku_code	sku_name	store_code	st
0	0.06	3.39	0.49	3.88	2018-04-26	1.0	5.94	6.0	12.0	1.0
1	0.13	5.05	0.70	5.75	2018-04-26	1.0	11.87	12.0	12.0	1.0
2	0.01	3.39	0.49	3.88	2018-04-26	8.0	5.99	6.0	12.0	1.0
3	0.01	5.05	0.70	5.75	2018-04-26	8.0	11.99	12.0	12.0	1.0
4	10.87	15.15	2.10	17.25	2018-04-26	11.0	19.13	30.0	36.0	3.0

- 店铺明细，包含字段：店铺名称，开店日期，闭店日期等

```
In [673]: storeMapping.head()
```

```
Out[673]:
```

	store_code	store_name	week_index	上架日期	下架日期	批次
0			17.0	2018-04-25	2099-12-31	第一批次
1			17.0	2018-04-25	2099-12-31	第一批次
2			17.0	2018-04-25	2099-12-31	第一批次
3			17.0	2018-04-25	2099-12-31	第一批次
4			17.0	2018-04-25	2099-12-31	第一批次

三、其他数据

- 日期Mapping，包含字段：日期，日期处于第几周，日期是星期几，工作日/节假日

```
In [671]: dateDataFrame.head()
```

```
Out[671]:
```

	date	dayType	weekDay	weekNum
0	2018-01-01	工作日	1	1
1	2018-01-02	工作日	2	1
2	2018-01-03	工作日	3	1
3	2018-01-04	工作日	4	1
4	2018-01-05	工作日	5	1

四、使用的模块

- pandas
- xlrd
- datetime
- time

五、涉及知识点

- 读取Excel文件里的数据并将数据转化为DataFrame
- 合并不同Excel文件的数据（相当于Vlookup）
- 清理DataFrame里的无效和重复字段
- 转换日期字符串为datetime对象
- 对数据进行聚合计算（相当于PivotTable）
- DataFrame转换index为column
- 调整输出数据的格式

知识点一：读取Excel文件里的数据并将数据转化为DataFrame

```
def read_excel_workbook(path):
    workbook = xlrd.open_workbook(path)
    sheet = workbook.sheet_by_index(0)
    data = {}
    for i in range(sheet.ncols):
        # 按列读取数据，第一个数据是列的名字，剩下的数据是列的值
        h, *rest = sheet.col_values(i)
        data[h] = rest
    frame = pd.DataFrame(data)
    return frame
```

知识点二：合并不同Excel文件的数据（相当于Vlookup）

```
dataSource = read_excel_workbook('/Volumes/Transcend/Bliallifeng/
专题数据/果汁/datasource.xlsx')
storeMapping = read_excel_workbook('/Volumes/Transcend/
Bliallifeng/专题数据/果汁/store_mapping.xlsx')
# pd.merge: 前两个参数是待合并数据，<how>规定了合并的类型，<on>规定了关联
的字段，这个字段同时存在于两个数据源，如果字段名不一致，可以用
<left_on>,<right_on>
dataSource = pd.merge(dataSource,storeMapping[['store_code','批
次']],how='inner',on='store_code')
dataSource =
pd.merge(dataSource,dateDataFrame,how='inner',on=None,left_on='or
der_date',right_on='date')
```

知识点三：清理DataFrame里的无效和重复字段

```
dataSource.drop(['type','date'],axis='columns',inplace=True)
```

知识点四：转换日期字符串为datetime对象

```
def ChangeDateFormat(x):
    if x.count('/')>0:
        dt = time.strptime(x,'%Y/%m/%d')
    elif x.count('-')>0:
        dt = time.strptime(x,'%Y-%m-%d')
    dtStamp = time.mktime(dt)
    dtDateTime = datetime.datetime.fromtimestamp(dtStamp)
```

```
return dtDateTime
```

知识点五：对数据进行聚合计算（相当于PivotTable）

```
sum1 = dataSource[dataSource['dayType']=='工作日']  
[[ 'cost_total', 'sku_cnt', 'payable_price_total' ]].groupby([dataSource['weekNum'], dataSource['order_date'], dataSource['sku_name']]).  
sum()  
sum2 =  
sum1[[ 'avgSkuCnt', 'avgPayPriceTotal', 'avgCost' ]].groupby([sum1['weekNum'], sum1['sku_name']]).mean()
```

知识点六：DataFrame转换index为column

```
# 将合并计算后的DataFrame的index转化为column,然后与其他数据关联  
sum1 =  
pd.merge(sum1.reset_index(), dateDataFrame[['date', 'storeCount']],  
how='inner', left_on='order_date', right_on='date')
```

知识点七：调整输出数据的格式

```
# 定义函数：  
float_format = lambda x : '%.2f' % x  
percent_format = lambda x : '%.2f%%' % (x * 100)  
# DataFrame使用applymap, Series使用map  
sum2[[ 'avgSkuCnt', 'avgCost', 'avgPayPriceTotal', 'avgProfit' ]] =  
sum2[[ 'avgSkuCnt', 'avgCost', 'avgPayPriceTotal', 'avgProfit' ]].applymap(float_format)  
sum2['avgProfitRate'] = sum2['avgProfitRate'].map(percent_format)
```