

Python 0-1基礎商務分析實戰 Day 3 課堂筆記

日期: 2022年10月18日

講師: Chandler Liu

Google Colab 操作方式

```
import os

from google.colab import drive
drive.mount('/content/drive')

##出現提示欄進行授權

os.chdir('/content/drive/My Drive/temp/未命名資料夾') #切換該目錄
os.listdir() #確認目錄內容
```

```
pio.renderers.default = 'notebook'
#在 Notebook 繪圖並呈現
%matplotlib inline
import cufflinks as cf
from plotly.offline import download_plotlyjs, init_notebook_mode, plot,
iplot

cf.go_offline()
init_notebook_mode(connected=False)
```

```
# 繪圖的 Function
def configure_plotly_browser_state():
    import IPython
    display(IPython.core.display.HTML('''
        <script src="/static/components/requirejs/require.js"></script>
        <script>
            requirejs.config({
                paths: {
                    base: '/static/base',
                    plotly: 'https://cdn.plot.ly/plotly-1.5.1.min.js?noext',
                },
            });
        </script>
        '''))
configure_plotly_browser_state()
```

小複習

常見 ETL 作法

```
###
# Pandas 常見 ETL 用法
# 第二, Transform 轉換
# 資料型態統一或轉換
df = transaction.copy()
df.info()

###
df['date']

###
pd.to_datetime(df['date'], format='%Y.%m.%d')

###
# Pandas 常見 ETL 用法
# 轉換成時間資料型態
df['date'] = pd.to_datetime(df['date'], format='%Y.%m.%d')

###
df['date']

###
# 轉換成任何型態
# df.astype()
print(df["uid"].dtypes) # 查看 df 的各欄位資料型態

###
df["uid"] = df["uid"].astype(str)

print(df["uid"].dtypes)

###
df.head()
###
df.info()

###
# Pandas 常見 ETL 用法
# 空缺值處理
```

```
# any() 只要有 true 都會回傳
# axis = 1 -> 欄位
# axis = 0 -> 列
null_mask = df.isnull().any(axis=1)

###
df.loc[null_mask]

###
# 很多種處理方式：
# 最基本直接的做法：dropna(), fillna()
df.dropna()

###
# 把 price 有空值的 drop 掉
# 把 product 有空值的 fill: "missing"
df['product'].fillna("missing")

###
# product 欄位 fillna : missing
# price 欄位 dropna

df['product'] = df['product'].fillna("missing")
df

###
df = df.dropna(axis=0)

# Pandas 常見 ETL 用法
# 新增欄位：跟 dictionary 很像
df['year'] = "2022"
df

###
revenue = []
for price, quantity in zip(df['price'], df['quantity']):
    revenue.append(price*quantity)
revenue

df['revenue2'] = revenue

###
df['revenue'] = df['price'] * df['quantity']
df

###
# Pandas 常見 ETL 用法
```

```
# 排序: sort_values
revenue_df = df.sort_values(by=["revenue"], ascending=False)
revenue_df
```

```
###
# Pandas 常見 ETL 用法
# 進階 Extract 方法: Groupby
# 每一筆交易中, 最低的商品價格
# grouped = df.groupby("tid")
# grouped.get_group('T0003')
```

```
###
df.groupby("tid")["price"].min()
```

Groupby 小試身手

```
###
# Q: 每種水果平均能賣出多少個?
avg_amount = df.groupby("product")["quantity"].mean()
avg_amount = pd.DataFrame(avg_amount)
avg_amount
```

```
###
# Q: 每種水果平均能賣到多少錢?
avg_revenue = df.groupby("product").revenue.mean()
avg_revenue = pd.DataFrame(avg_revenue)
avg_revenue
```

```
###
# Pandas 常見 ETL 用法
# 進階 Transform 方法: 合併
sales_report = pd.concat([avg_amount, avg_revenue], axis=1)
sales_report
```

```
###
# Q: 想將欄位名稱更改怎麼做?
# Avg. Quantity, Avg. Revenue
renamed = {"quantity": "Avg. Quantity", "revenue": "Avg. Revenue"}
final_sales_report = sales_report.rename(columns = renamed)
final_sales_report
```

```
###
# Pandas 常見 ETL 用法
# 最後, Load 載入
# df.to_csv()
```

```
# df.to_excel() .xlsx
final_sales_report.to_csv("sales_report.csv", encoding = "utf-8-sig")
```

Data Analysis : 80/20法則

```
# %%
## Plotly網頁動態繪圖操作

# %%
from show import showing
showing('https://imgur.com/PpbF1d6.png')

# %%
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.express as px
import plotly.io as pio
from datetime import datetime
pio.renderers.default = 'browser'

# %%
# 動態圖寫法
fig = px.line(x = [1, 2, 3, 4],
              y = [4, 8, 12, 16] )

fig.update_layout(
    xaxis_title="some numbers",
    yaxis_title="love you" )

# 在notebook中產出檔案
iplot(fig)

# 產出html檔案
plot(fig, filename='love.html')

# %%
## 如何找出利潤高的系列？並視覺化
showing('https://imgur.com/12bXFVC.png')

# %%
#### 問題：請讀取 sales_data2019.csv, 並存至all_data變數裡面

# %%
import pandas as pd
```

```
all_data = pd.read_csv("sales_data2019.csv", encoding = "utf-8-sig")
```

```
# %%  
all_data.info()
```

```
# %%  
##### 問題: 請創造「利潤」欄位在all_data  
# 使用all_data裡面的「單價」-「成本」
```

```
# %%
```

```
all_data['利潤'] = all_data['單價'] - all_data['成本']
```

```
# %%  
all_data
```

```
# %%  
##### 問題: 我只要抓取 `['會員','訂單時間','系列','利潤','產品','廣告代號all']` 這幾個欄位, 並  
覆蓋原先的all_data變數
```

```
# %%  
all_data = all_data.loc[:,['會員','訂單時間','系列','利潤','產品','廣告代號all']]
```

```
# 問題: 把訂單時間中的 "T" 去掉  
all_data['訂單時間'] = all_data['訂單時間'].str.replace('?', '')
```

```
# 將訂單時間改成datetime形式  
all_data['訂單時間'] = pd.to_datetime(all_data['訂單時間'])
```

```
# 設立選取資料起始時間 (2019年 11月 1日)  
begin = datetime(2019, 1, 1)
```

```
# 設立選取資料結束時間 (2019年 11月 1日)  
end = datetime(2019, 11, 1)
```

```
# 問題: 如何挑選訂單時間小於end( 結束時間之前 )的資料 ?
```

```
# 問題: 如何挑選訂單時間大於begin( 起始時間 )的資料 ?
```

```
print(all_data.info())
```

```

# %%
all_data

# %%
# ### 請根據每一個「系列」(groupby)來計算「利潤」的總和
# %%
showimg('https://imgur.com/Txmm0qE.png')

# %%
# as_index = False, 將所有欄位名稱轉換成正常的欄位名稱, 而不轉換成流水號

# as_index = True, 則將欄位(系列)轉換成流水號, 這樣即無法取值, 舉例:a['利潤'] ???

a = all_data.groupby('系列', as_index = True)['利潤'].sum()

# %%
a

# %%
# 請根據每一個「系列」(groupby)來計算「利潤」的總和
# as_index = False : 將「系列」欄位轉變成正常欄位, 不存在流水號(index)裡面
profit_data = all_data.groupby('系列', as_index = False)['利潤'].sum()

# %%

profit_data['利潤']

# %%
# ### 問題: 請根據「利潤」高至低排序profit_data

# %%

profit_data = profit_data.sort_values(by="利潤", ascending=False)

# %%
profit_data

# %%
# 問題: 請將profit_data輸出為 '系列利潤綜整.csv'

```

```
# 問題: 請將all_data輸出為 'process_data.csv'

# %%
# barchart:長條圖
fig = px.bar(profit_data,
              x = '系列', y = '利潤',
              title='2019年各系列利潤排序長條圖',
              color = '利潤')
iplot(fig)
plot(fig, filename= '利潤.html')
# %%
#### 抓出前 10 高利潤的商品, 並存到top10_profit變數中
# hint: iloc

# %%
profit_data

# %%
top10_profit = profit_data.iloc[ ???, :: ]

# %%
top10_profit

# %%
fig = px.bar(top10_profit,
              x = '系列',
              y = '利潤',
              title='2019年top10各系列利潤排序長條圖',
              color = '利潤')
fig.show()
plot(fig, filename= 'top10.html')

# %%
# 80/20法則
# 計算「累計利潤總額」
profit_data['累計利潤總額'] = profit_data['利潤'].cumsum()

# %%
# 計算「利潤總額」
all_profit = profit_data['利潤'].sum()

# %%
```


問題: 如何計算累計利潤佔比 = 累計利潤總額 / 利潤總額, 並存到profit_data裡面, 命名欄位名稱為「累計利潤佔比」

```
profit_data['累計利潤佔比'] = profit_data['累計利潤總額'] / total_profit
```

%%

問題: 請抓出['累計利潤佔比'] <= 0.8 的資料, 並宣告到一個變數, 名為profit_data80

概念: 抓小於貢獻利潤80%前的產品, 意涵為抓取貢獻 0~ 80% 利潤的產品

```
mask = profit_data['累計利潤佔比'] <= 0.8
```

```
profit_data80 = profit_data.loc[mask,:]
```

%%

```
profit_data80
```

%%

幾%的產品貢獻了80%的利潤

```
len(profit_data80) / len(profit_data)
```

```
fig = px.bar(profit_data80,
              x = '系列',
              y = '利潤',
              title='2019年80/20各系列利潤排序長條圖',
              color = '利潤')
plot(fig, filename= '8020.html')
```

%%

問題: 使用all_data計算每一會員的利潤總額

%%

```
all_data.head()
```

%%

每一個/每一種/每一... --> 樞紐分析(groupby)

計算每一個(groupby)會員(我主要要groupby的欄位)利潤(要計算的欄位)的總和(sum())

```
member_profit_data = all_data.???
```

```
# %%  
member_profit_data
```

```
# %%  
# ### 問題: 請根據「利潤」高至低排序member_profit_data
```

```
# %%  
member_profit_data = member_profit_data.??? 
```

```
# %%  
member_profit_data
```

```
# %%  
# ### 請從member_profit_data裡面挑選Top10的VIP客戶
```

```
# %%  
member10_profit_data = member_profit_data.iloc[??, :]
```

```
# %%  
member10_profit_data
```

```
# %%  
member10_profit_data.to_csv('最有價值會員.csv', encoding='utf-8-sig' )
```