gropuby_sorting

January 22, 2021

0.1 Groupby Sorting

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
### Reading in sales dataframe

sales = pd.read_csv('supermarket_sales.csv')
sales.head()
```

[2]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83	4.761905	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40	4.761905	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	4.761905	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17	4.761905	30.2085	5.3

Groupby is similar to an excel Pivot. You can group on anynumber of columns - but we'll start with just one. Typically, after grouping, you'll want to performs some function, here are some of the most common ones: - sum() - mean() - max() - min()

[3]:
```
sales.groupby('Branch').sum()
```

[3]:
```
        Unit price  Quantity     Tax 5%        Total       cogs  \
Branch
A         18625.49      1859  5057.1605  106200.3705  101143.21
B         18478.88      1820  5057.0320  106197.6720  101140.64
C         18567.76      1831  5265.1765  110568.7065  105303.53

        gross margin percentage  gross income  Rating
Branch
A                   1619.047619     5057.1605  2389.2
B                   1580.952381     5057.0320  2263.6
C                   1561.904762     5265.1765  2319.9
```

Notice how only the numerical columns remained.

[4]:
```
sales.groupby('Gender').mean()
```

[4]:
```
        Unit price  Quantity     Tax 5%       Total        cogs  \
Gender
Female   55.263952  5.726547  15.956936  335.095659  319.138723
Male     56.081944  5.292585  14.799487  310.789226  295.989739

        gross margin percentage  gross income    Rating
Gender
Female                 4.761905     15.956936  6.964471
Male                   4.761905     14.799487  6.980962
```

What if we want to apply certain fucntions to certain columns? We can use the agg() function with a dictionary. You can read more about it here: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.agg.html

[5]:
```
sales.groupby('Gender').agg({'Unit price':'mean', 'Quantity': 'sum'})
```

[5]:
```
        Unit price  Quantity
Gender
Female   55.263952      2869
Male     56.081944      2641
```

We can also groupby multiple columns - must be in the form of a list.

[6]:
```
sales.groupby(['Gender', 'Branch']).agg({'Unit price':'mean', 'Quantity':␣
↪'sum'})
```

[6]:
```
                Unit price  Quantity
Gender Branch
Female A         56.086149       909
       B         54.168148       911
       C         55.517584      1049
Male   A         53.606816       950
```

```
        B        57.080235       909
        C        57.904200       782
```

Finally, we can sort values by a column with the sort_values() function.

[7]:
```
sales.groupby(['Gender', 'Branch']).agg({'Unit price':'mean', 'Quantity':␣
↪'sum'}).sort_values(by = 'Quantity', ascending = False)
```

[7]:
```
                Unit price  Quantity
Gender Branch
Female C         55.517584      1049
Male   A         53.606816       950
Female B         54.168148       911
       A         56.086149       909
Male   B         57.080235       909
       C         57.904200       782
```

We can also sort by multiple columns (using a list). To sort in descending order, pass in "ascending = False".

[8]:
```
sales.groupby(['Gender', 'Branch']).agg({'Unit price':'mean', 'Quantity':␣
↪'sum'}).sort_values(by = ['Quantity', 'Unit price'], ascending = False)
```

[8]:
```
                Unit price  Quantity
Gender Branch
Female C         55.517584      1049
Male   A         53.606816       950
Female B         54.168148       911
Male   B         57.080235       909
Female A         56.086149       909
Male   C         57.904200       782
```

0.1.1 Now you try. Note that each of these can be done with just one line of code.

a) Find which city generated the most gross income last year.

b) Find out which customer type and gender gave the highest rating to any of the product lines.

[9]:
```
'''
a) Find which city generated the most gross income last year.
'''

### your code here
```

[9]:
```
'\na) Find which city generated the most gross income last year.\n'
```

[10]:
```
'''
b) Find out which customer type and gender gave the highest rating to any of␣
 ↪the product lines.
'''

### your code here
```

[10]:
```
'\nb) Find out which customer type and gender gave the highest rating to any of
the product lines.\n'
```