

Black Friday Excel Data

```
In [2]: import openpyxl as xl
        from tax_calculator import tax_calculator
        from pprint import pprint
```

read in maven_ski_shop_data.xlsx Then, assign the orders worksheet to an intuitive variable name.

```
In [6]: wb = xl.load_workbook (filename='maven_ski_shop_data.xlsx')
        orders = wb['Orders_Info']
```

```
In [8]: def column_printer (sheet,column):
        for i in range(1,sheet.max_row+1):
            print(f'{column}{i}', sheet[f'{column}{i}'].value)
```

```
In [9]: def column_printer(sheet , column):
        for cell in sheet[column]:
            print(f'{cell.coordinate}' , sheet[cell.coordinate].value)
```

```
In [29]: column_printer(orders, 'A')
```

```
A1 Order_ID
A2 100000
A3 100001
A4 100002
A5 100003
A6 100004
A7 100005
A8 100006
A9 100007
A10 100008
A11 100009
A12 100010
A13 100011
A14 100012
A15 100013
A16 100014
A17 100020
A18 100021
A19 100022
A20 100023
A21 100024
A22 100025
A23 100026
A24 100027
A25 100031
A26 100032
A27 100033
A28 100034
```

```
In [30]: column_printer(orders, 'D')
```

D1 Subtotal
D2 15.98
D3 899.97
D4 799.97
D5 117.96
D6 5.99
D7 599.99
D8 24.99
D9 1799.94
D10 99.99
D11 254.95
D12 29.98
D13 99.99
D14 25.98
D15 649.98
D16 89.99
D17 119.99
D18 599.99
D19 649.98
D20 24.99
D21 99.99
D22 99.99
D23 5.99
D24 24.99
D25 999.96
D26 99.99
D27 399.97
D28 89.99

```
In [37]: order_dict = {
    orders[f'A{order}'].value:[
        orders[f'B{order}'].value,
        orders[f'C{order}'].value,
        orders[f'D{order}'].value,
        orders[f'G{order}'].value,
        str(orders[f'H{order}'].value).split(', ')
    ]
    for order in range(2, orders.max_row + 1)
}
pprint (order_dict)
```

```

{100000: ['C00004', '11/26/2021', 15.98, 'Sun Valley', ['10001', '10002']],
100001: ['C00007', '11/26/2021', 899.97, 'Stowe', ['10008', '10009', '10010']],
100002: ['C00015',
        '11/26/2021',
        799.97,
        'Mammoth',
        ['10011', '10012', '10013']],
100003: ['C00016',
        '11/26/2021',
        117.96,
        'Stowe',
        ['10002', '10003', '10004', '10006']],
100004: ['C00020', '11/26/2021', 5.99, 'Sun Valley', ['10001']],
100005: ['C00010', '11/26/2021', 599.99, 'Mammoth', ['10010']],
100006: ['C00006', '11/26/2021', 24.99, 'Mammoth', ['10004']],
100007: ['C00001',
        '11/26/2021',
        1799.94,
        'Mammoth',
        ['10008', '10008', '10009', '10009', '10009', '10010', '10010']],
100008: ['C00003', '11/26/2021', 99.99, 'Sun Valley', ['10005']],
100009: ['C00014',
        '11/26/2021',
        254.95,
        'Sun Valley',
        ['10002', '10003', '10004', '10006', '10007']],
100010: ['C00001', '11/26/2021', 29.98, 'Mammoth', ['10002', '10003']],
100011: ['C00001', '11/26/2021', 99.99, 'Mammoth', ['10005']],
100012: ['C00005', '11/26/2021', 25.98, 'Sun Valley', ['10001', '10003']],
100013: ['C00008', '11/26/2021', 649.98, 'Stowe', ['10012', '10013']],
100014: ['C00013', '11/26/2021', 89.99, 'Sun Valley', ['10014']],
100020: ['C00004', '11/27/2021', 119.99, 'Sun Valley', ['10007']],
100021: ['C00017', '11/27/2021', 599.99, 'Stowe', ['10010']],
100022: ['C00019', '11/27/2021', 649.98, 'Sun Valley', ['10012', '10013']],
100023: ['C00002', '11/27/2021', 24.99, 'Stowe', ['10004']],
100024: ['C00008', '11/27/2021', 99.99, 'Stowe', ['10005']],
100025: ['C00021', '11/27/2021', 99.99, 'Mammoth', ['10008']],
100026: ['C00022', '11/27/2021', 5.99, 'Sun Valley', ['10001']],
100027: ['C00006', '11/28/2021', 24.99, 'Mammoth', ['10002']],
100031: ['C00018',
        '11/28/2021',
        999.96,
        'Stowe',
        ['10005', '10008', '10009', '10010']],
100032: ['C00018', '11/28/2021', 99.99, 'Stowe', ['10006']],
100033: ['C00010',
        '11/28/2021',
        399.97,
        'Mammoth',
        ['10005', '10008', '10009']],
100034: ['C00016', '11/28/2021', 89.99, 'Stowe', ['10014']]}

```

Sales Tax Calculation

- If location is Sun Valley, apply a sales tax of 8%
- If location is Mammoth, apply a sales tax of 7.75%.
- If location is Stowe, apply a sales tax of 6%

```
In [44]: from tax_calculator import tax_calculator

for order in order_dict.values():
    if order[3] == 'Sun Valley':
        transaction = tax_calculator(order[2] ,.08)
    elif order[3] == 'Mammoth' :
        transaction = tax_calculator(order[2] , 0.775)
    else :
        transaction = tax_calculator(order[2] , .06)
    order.insert(3 , transaction[1])
    order.insert(4 , transaction[2])

pprint (order_dict)
```

```

{100000: ['C00004',
          '11/26/2021',
          15.98,
          1.28,
          17.26,
          'Sun Valley',
          ['10001', '10002']],
 100001: ['C00007',
          '11/26/2021',
          899.97,
          54.0,
          953.97,
          'Stowe',
          ['10008', '10009', '10010']],
 100002: ['C00015',
          '11/26/2021',
          799.97,
          619.98,
          1419.95,
          'Mammoth',
          ['10011', '10012', '10013']],
 100003: ['C00016',
          '11/26/2021',
          117.96,
          7.08,
          125.04,
          'Stowe',
          ['10002', '10003', '10004', '10006']],
 100004: ['C00020', '11/26/2021', 5.99, 0.48, 6.47, 'Sun Valley', ['10001']],
 100005: ['C00010',
          '11/26/2021',
          599.99,
          464.99,
          1064.98,
          'Mammoth',
          ['10010']],
 100006: ['C00006', '11/26/2021', 24.99, 19.37, 44.36, 'Mammoth', ['10004']],
 100007: ['C00001',
          '11/26/2021',
          1799.94,
          1394.95,
          3194.89,
          'Mammoth',
          ['10008', '10008', '10009', '10009', '10009', '10010', '10010']],
 100008: ['C00003', '11/26/2021', 99.99, 8.0, 107.99, 'Sun Valley', ['10005']],
 100009: ['C00014',
          '11/26/2021',
          254.95,
          20.4,
          275.35,
          'Sun Valley',
          ['10002', '10003', '10004', '10006', '10007']],
 100010: ['C00001',
          '11/26/2021',
          29.98,
          23.23,
          53.21,
          'Mammoth',
          ['10002', '10003']],
 100011: ['C00001', '11/26/2021', 99.99, 77.49, 177.48, 'Mammoth', ['10005']],

```

```

100012: ['C00005',
        '11/26/2021',
        25.98,
        2.08,
        28.06,
        'Sun Valley',
        ['10001', '10003']],
100013: ['C00008',
        '11/26/2021',
        649.98,
        39.0,
        688.98,
        'Stowe',
        ['10012', '10013']],
100014: ['C00013', '11/26/2021', 89.99, 7.2, 97.19, 'Sun Valley', ['10014']],
100020: ['C00004', '11/27/2021', 119.99, 9.6, 129.59, 'Sun Valley', ['10007']],
100021: ['C00017', '11/27/2021', 599.99, 36.0, 635.99, 'Stowe', ['10010']],
100022: ['C00019',
        '11/27/2021',
        649.98,
        52.0,
        701.98,
        'Sun Valley',
        ['10012', '10013']],
100023: ['C00002', '11/27/2021', 24.99, 1.5, 26.49, 'Stowe', ['10004']],
100024: ['C00008', '11/27/2021', 99.99, 6.0, 105.99, 'Stowe', ['10005']],
100025: ['C00021', '11/27/2021', 99.99, 77.49, 177.48, 'Mammoth', ['10008']],
100026: ['C00022', '11/27/2021', 5.99, 0.48, 6.47, 'Sun Valley', ['10001']],
100027: ['C00006', '11/28/2021', 24.99, 19.37, 44.36, 'Mammoth', ['10002']],
100031: ['C00018',
        '11/28/2021',
        999.96,
        60.0,
        1059.96,
        'Stowe',
        ['10005', '10008', '10009', '10010']],
100032: ['C00018', '11/28/2021', 99.99, 6.0, 105.99, 'Stowe', ['10006']],
100033: ['C00010',
        '11/28/2021',
        399.97,
        309.98,
        709.95,
        'Mammoth',
        ['10005', '10008', '10009']],
100034: ['C00016', '11/28/2021', 89.99, 5.4, 95.39, 'Stowe', ['10014']]

```

```

In [46]: for index,order in enumerate(order_dict.values(), start=2):
        orders[f'E{index}'] = order[3]
        orders[f'F{index}'] = order[4]

```

```

In [47]: wb.save('maven_ski_shop_data_fixed.xlsx')

```

Analysis !

```

In [49]: def column_sum(column_index , dictionary):
        return round(sum([value[column_index] for value in dictionary.values()]) , 2)

```

```
In [50]: print(column_sum(2, order_dict))  
         print(column_sum(3, order_dict))  
         print(column_sum(4, order_dict))
```

```
8731.47  
3323.35  
12054.82
```

What is the average of our subtotals?

```
In [51]: round(column_sum(2, order_dict) / len(order_dict), 2)
```

```
Out[51]: 323.39
```

```
In [52]: subtotals = [value[2] for value in order_dict.values()]  
         round(sum(subtotals) / len(subtotals), 2)
```

```
Out[52]: 323.39
```

How many unique customers did we have?

```
In [54]: unique_cutomers = len(set([value[0] for value in order_dict.values()]))  
         orders_per_customers = len(order_dict) / unique_cutomers  
         print (unique_cutomers, orders_per_customers)
```

```
19 1.4210526315789473
```

How many items in total did we sell?

```
In [55]: sum([len(value[0]) for value in order_dict.values()])
```

```
Out[55]: 162
```

Sales By Location

```
In [67]: location_sums = {}  
  
         for data in order_dict.values():  
  
             location = data[5]  
  
             if location not in location_sums:  
  
                 location_sums[location]=0  
  
             location_sums[location] += data[2]  
  
         location_sums
```

```
Out[67]: {'Stowe': 89.99}
```

Aggregator Function

```
In [74]: def aggregator(category_index, field_to_sum_index , dictionary):  
        category_sums = {}  
        for data in dictionary.values():  
            category = data[category_index]  
            if category not in category_sums:  
                category_sums[category] = 0  
            category_sums[category] += data[field_to_sum_index]  
  
        return category_sums
```

```
In [75]: customer_id, date, subtotals, taxes, totals, location, items_sold = (0, 1, 2, 3, 4, 5,  
aggregator(1, 4, order_dict)
```

```
Out[75]: {'11/26/2021': 8255.18, '11/27/2021': 1783.99, '11/28/2021': 2015.65}
```

```
In [76]: aggregator(0, 4, order_dict)
```

```
Out[76]: {'C00004': 146.85,  
          'C00007': 953.97,  
          'C00015': 1419.95,  
          'C00016': 220.43,  
          'C00020': 6.47,  
          'C00010': 1774.93,  
          'C00006': 88.72,  
          'C00001': 3425.58,  
          'C00003': 107.99,  
          'C00014': 275.35,  
          'C00005': 28.06,  
          'C00008': 794.97,  
          'C00013': 97.19,  
          'C00017': 635.99,  
          'C00019': 701.98,  
          'C00002': 26.49,  
          'C00021': 177.48,  
          'C00022': 6.47,  
          'C00018': 1165.95}
```

```
In [77]: aggregator(customer_id, totals, order_dict)
```

```
Out[77]: {'C00004': 146.85,  
          'C00007': 953.97,  
          'C00015': 1419.95,  
          'C00016': 220.43,  
          'C00020': 6.47,  
          'C00010': 1774.93,  
          'C00006': 88.72,  
          'C00001': 3425.58,  
          'C00003': 107.99,  
          'C00014': 275.35,  
          'C00005': 28.06,  
          'C00008': 794.97,  
          'C00013': 97.19,  
          'C00017': 635.99,  
          'C00019': 701.98,  
          'C00002': 26.49,  
          'C00021': 177.48,  
          'C00022': 6.47,  
          'C00018': 1165.95}
```


In []: