conclusions_quiz

October 17, 2017

1 Drawing Conclusions Quiz

Use the space below to explore `store_data.csv` to answer the quiz questions below.

```
In [2]: # imports and load data
        import pandas as pd

        df = pd.read_csv('store_data.csv')
        df.head()
```

```
Out[2]:          week  storeA  storeB  storeC  storeD  storeE
        0  2014-05-04    2643    8257    3893    6231    1294
        1  2014-05-11    6444    5736    5634    7092    2907
        2  2014-05-18    9646    2552    4253    5447    4736
        3  2014-05-25    5960   10740    8264    6063     949
        4  2014-06-01    7412    7374    3208    3985    3023
```

```
In [3]: # explore data
        df.shape
```

```
Out[3]: (200, 6)
```

```
In [3]: df.dtypes
```

```
Out[3]: week      object
        storeA     int64
        storeB     int64
        storeC     int64
        storeD     int64
        storeE     int64
        dtype: object
```

```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
week      200 non-null object
```

```
storeA    200 non-null int64
storeB    200 non-null int64
storeC    200 non-null int64
storeD    200 non-null int64
storeE    200 non-null int64
dtypes: int64(5), object(1)
memory usage: 9.5+ KB
```

```
In [6]: df.describe()
```

```
Out[6]:             storeA        storeB       storeC       storeD       storeE
        count    200.000000    200.000000   200.000000   200.000000   200.000000
        mean    5865.480000   6756.710000  4942.105000  5431.405000  2580.025000
        std     2958.230318   3601.339489  1325.407768  1183.111323  1725.651381
        min      137.000000     14.000000   927.000000  2276.000000    39.000000
        25%     3812.250000   3884.500000  4053.500000  4717.000000  1235.000000
        50%     5713.500000   6771.000000  4962.500000  5382.000000  2522.000000
        75%     7662.250000   9350.500000  5801.750000  6243.750000  3574.250000
        max    14403.000000  15841.000000  8293.000000  8190.000000  7553.000000
```

```
In [10]: from datetime import datetime, timedelta
         # total sales for the last month
         max_week = max(df["week"])
         #print(max_week)
         d = datetime.strptime(max_week, '%Y-%m-%d')
         dstart = d - timedelta(weeks=4)
         start_month_of_max_week = dstart.strftime('%Y-%m-%d')
         #print(start_month_of_max_week)

         df_max = df[(df['week'] > start_month_of_max_week) & (df['week'] <= max_week)]
         #df_max.head()

         df_max_stores_summed = df_max.loc[:,'storeA':'storeE'].sum()
         print("total sales for last month: {}".format(sum(df_max_stores_summed)))
         print(df_max_stores_summed)
```

```
total sales for last month: 96739
storeA    25127
storeB    24595
storeC    16447
storeD    22783
storeE     7787
dtype: int64
```

```
In [15]: # average sales
         # (is this the average of all sales per month, across stores?)
         #df_all_stores_summed = df.loc[:,'storeA':'storeE'].sum()
         #row_count = df.shape[0]
         #sum(df_all_stores_summed/row_count)

         # no, it means Which store makes the most sales on average?
         df_each_store_summed = df.loc[:,'storeA':'storeE'].mean()
         print(df_each_store_summed)
```

```
storeA    5865.480
storeB    6756.710
storeC    4942.105
storeD    5431.405
storeE    2580.025
dtype: float64
```

```
In [25]: # sales on march 13, 2016
         # Which store sells the most during the week of March 13th, 2016?

         df_target_week = df[(df['week'] == '2016-03-13')]
         df_target_week.head()
```

```
Out[25]:           week  storeA  storeB  storeC  storeD  storeE
         97  2016-03-13    2054    1390    5112    5513    2536
```

```
In [41]: # worst week for store C
         #df_store_c = df.loc[:,'storeC':'storeC']
         df_store_c = df[['week','storeC']]
         #df_store_c.head()
         df_store_c.loc[df['storeC'].idxmin()]
```

```
Out[41]: week      2014-07-06
         storeC           927
         Name: 9, dtype: object
```

```
In [49]: # total sales during most recent 3 month period
         dstart_3mo = d - timedelta(weeks=3*4)
         dstart_3mo_week = dstart_3mo.strftime('%Y-%m-%d')
         #print(dstart_3mo_week)

         df_3mo = df[(df['week'] > dstart_3mo_week) & (df['week'] <= max_week)]
         #df_3mo.head()
         df_3mo_summed = df_3mo.loc[:,'storeA':'storeE'].sum()
         df_3mo_summed
```

```
Out[49]: storeA    82412
         storeB    75544
         storeC    60417
         storeD    68412
         storeE    27221
         dtype: int64
```

```
In [ ]:
```