19 HW #8B: Pandas Window Functions

In order to receive full credit, please turn in a document which is python code containing what would be run to return the data asked.

The queries below rely on information from the stock return data. To load the data use the following commands. **Note: these load retdate as a date, rather than a string**

The best approach to learning from these problems is to complete them using pen and paper, working by yourself and then using your group to double check your results. The First Five problems provide a short overview of the core concepts in the assignment, so make sure that you understand them. The Main Problems section contains questions which range from easy to very difficult. Remember to don't get stuck! If a problem is taking a long time or is too difficult, *use your group!*

```
## Initial Information
import pandas as pd
import numpy as np

df2010D = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2010.tdf',
        sep='\t', engine='python', names=['symb', 'retdate', 'opn', 'high', 'low',
        'cls', 'vol', 'exch'], parse_dates=['retdate'])

df2011D = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2011.tdf',
        sep='\t', engine='python', names=['symb', 'retdate', 'opn', 'high', 'low',
        'cls', 'vol', 'exch'], parse_dates=['retdate'])

dffnd = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/fnd.tdf',
        sep='\t', engine='python', names=['gvkey', 'datadate', 'fyear', 'indfmr',
        'consol', 'popsrc', 'datafmt' , 'tic', 'cusip', 'conm', 'fyr', 'cash', 'dp',
        'ebitda', 'emp', 'invt', 'netinc', 'ppent', 'rev' , 'ui', 'cik'])

dfMTA = pd.read_csv('../sql-data/raw_data/mta/MTA_Hourly.tdf', sep='\t',
        engine='python', names=['plaza', 'mtadt', 'hr', 'direction', 'vehiclesez',
        'vehiclescash'])

dfTrans = pd.read_csv('../sql-data/raw_data/soapData.tdf', sep='\t',
        engine='python', names = ['orderid', 'userid', 'trans', 'type', 'local',
        'trans_dt', 'units', 'coupon', 'months', 'amt' ])
```

First Five

1. For each stock in 2010, return the original dataset as well as a column ("newcol") which is the 3 day moving average of the closing price (making sure to include the current closing price in the average).
2. For each stock in 2010, return the original dataset as well as a column ("newcol") which is the 3 day moving average of the closing price (making sure to exclude the current closing price in the average).
3. Calculate the average correlation between closing price and volume. To do this calculate the correlation for each stock and then take the average over all stocks. This should return a single number.
4. For each stock in 2010, calculate the percentage of the historical max closing price, up to (but not including) that point, that the current closing price is. Note that whenever a new max closing price is achieved the percent would be greater than 100.
5. Using `stack` or `unstack` create a DataFrame which is one row per symbol with columns for each month in 2010. The values in those columns should be the average closing price.

Main Problems

1. Using `stack` or `unstack` create a DataFrame which is one row per symbol with columns for each month in 2010. There should be multiple columns for each month, one for the average closing price, one for the average volume and one for the maximum volume (37 Columns: Symbol, Jan-Dec for average closing price, Jan-Dec for average volume and Jan-Dec for maximum volume). The DataFrame returned should not have a row index.
2. Using `stack` or `unstack` create a DataFrame which is one row per symbol with 12 columns which should be the cumulative volume for that month (including that month) over the entire year of 2010. E.g. This should be the running sum, but then accumulated per month.
3. Using `stack` or `unstack` create a DataFrame which is one row per symbol with columns for each month in 2010 *and* 2011. The values in those columns should be the average closing price for that month.
4. For stocks in 2010, write a query which creates a dataset containing closing price, symbol, retdate and the nominal change between yesterday's closing price and today's opening price. Ignore holes in the data, so that if the stock misses a day the change in price is from the last time listed.