Appendix C

Assignments

1 HW #0A: PostgreSQL Installation

Complete the tasks below in order to complete the assignment. Importantly, nothing needs to be turned in to complete this assignment.

Nothing needs to be physically turned in for this assignment, just make sure to complete the final step.

1. Install PostgreSQL on your computer. Note that installing PostgreSQL can be difficult and I recommend doing some research before beginning. If you are using a mac, I recommend using homebrew to install it. There is also something called PostgresAPP, which you can try.[1]

2. Once PostgresSQL is installed, please create a schema for the stocks database, which can be done using the command below.

   ```
   create schema stocks;
   commit;
   ```

   We will also create a schema for some other datasets that are using in the class, which is "cls" and can be done using the commands below:

   ```
   create schema cls;
   commit;
   ```

3. All of the data required for the homework can be found on the canvas page and the queries required to load the data onto your Postgres instance can be found in the data dictionary. Broadly speaking to load the data you must:

   (a) Have a schema to place the table in (which is what was done in the previous step)

   (b) Create a table to load the data into (the CREATE TABLE commands can be found in the data dictionary)

   (c) Use a COPY command to move the data from its raw format into the database.

4. Please load the following datasets onto your local SQL instance: (1) stocks.s2010 (2) stocks.s2011, (3) stocks.fnd

5. For an SQL Client, I would recommend using PopSQL. One important trick when installing is that if you are referring to your local machine the host is "localhost."

6. Make sure that on Slack and on the Canvas page you have a photo of yourself that will help me recognize you. After bootcamp you are free to make your Slack icon whatever you want, but during bootcamp you must have a recognizable photo as your avatar.

[1] I am not IT and am not going to diagnose issues relating to installing your software.

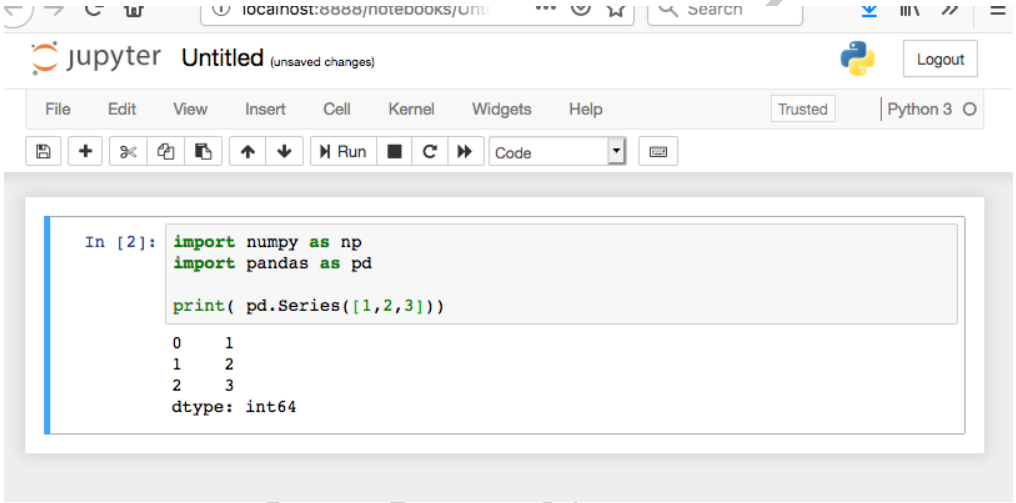
2 HW #0B: Pandas Installation

In order to do the assignments associated with pandas you will need to install Python (and specifically the package pandas) on your computer. The easiest way to do this is by using Anaconda (`https://www.anaconda.com/`) and then use Jupyter Notebooks (`https://jupyter.org/install`).

If you have this installed, you should be able to see a screen like the below and then run the commands below:

```
import numpy as np
import pandas as pd

print( pd.Series([1,2,3]) )
```



3 HW #0C: MS CAPP Installation instructions

The first assignment is to set up and access the data used in this course.

- Make sure that on Slack channel and on the Canvas page.
- Install PostgreSQL on your computer.[2] Note that installing PostgreSQL can be difficult and I recommend doing some research before jumping in.
  - The data itself can be found in the repo here: `https://github.com/NickRoss/sql-data`.
  - You are welcome to install the PostgreSQL server however you like. The instructions in the repo use docker and set up all the data (including table creation, loading data, etc.). However, if you do not wish to install docker you are welcome to use an alternative method. Two alternatives are: Postgres.app (`https://postgresapp.com/`) and brew (for macs).[3]
  - If you use a non-docker based method you will be required to load the data into the database yourself. Information and specific commands can be found in the data dictionary and the additional instructions at the end of this document.
- You will also be required to install a PostgreSQL client. I personally use one called Postico, but there are many, many others. PopSQL is a fun one to try too, but it requires an internet connection. One important trick when installing is that if you are referring to your local machine the host is "localhost."
- Note that there is nothing to turn in on this assignment.
- **If you installed WITHOUT using docker you need to do the following:**
  - Once PostgresSQL is installed, please create a schema for the stocks database, which can be done using the command below.

    ```
    create schema stocks;
    commit;
    ```

    We will also create a schema for some other datasets that are using in the class, which is "cls" and can be done using the commands below:

    ```
    create schema cls;
    commit;
    ```
  - All of the data required for the homework can be found in the repo and the queries required to load the data onto your Postgres instance can be found in the data dictionary. Broadly speaking to load the data you must:
    1. Have a schema to place the table in (which is what was done in the previous step)
    2. Create a table to load the data into (the CREATE TABLE commands can be found in the data dictionary)
    3. Use a COPY command to move the data from its raw format into the database.
  - Please load the following datasets onto your local SQL instance: (1) stocks.s2010 (2) stocks.s2011, (3) stocks.fnd in order to get access to the stocks data.

[2] I am not IT and am not going to diagnose issues relating to installing your software.

[3] Note that if you are installing with a mac you need to be careful regarding installation instructions for ARM based processors and older models.

4 HW #1A: Basic SQL Querying

The following questions utilize the financial data in the s2010, s2011 and fnd tables. Before beginning the assignment, *please read the data dictionary to better understand the data.* When doing so, keep an eye on data types for different columns as well as table organization.

- If no table information is given, use the 2010 data.
- If the query returns a significant number of rows, please only copy a few rows in your response.
- For those queries which require specifying a date, please use the format 'YYYY-MM-DD' (as in '2010-01-11'), making sure to use single quotes around the date itself.

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!*

First Five

Using the 2010 stocks data, write a query that returns the following.

1. All rows and columns relating to AAPL.
2. The date, open and closing price for AAPL on the 7th of January in 2010.
3. Write a query which returns the stock symbol, the date, the open and close price for the top five open prices in 2010 for stocks on the New York Stock Exchange (NYSE).
4. The days when AAPL has a volume more than 20 million and where the high is great than 45 dollars (2010 data)
5. Write a query which returns 3 columns: the return date, stock symbol and volume, but only for stocks that have a volume larger than 200 million in 2010.

Main Problems

1. Write a query which returns all information about about Google (GOOG), NetFlix (NFLX), Amazon (AMZN) and Microsoft (MSFT) in 2010.
2. Consider stocks on the NYSE which had a volume of more than 1 million. Which stocks (symbol and date) had their open price the same as their low and their closing price the same as their high (2010 data). Order them by symbol alphabetically.
3. Consider stocks on the NYSE in 2010 which had a volume of more than 1 million. Which stocks (symbol and date) had their closing price the same as their low and their opening price the same as their high? Sort them by reverse chronological order.
4. Consider stocks on the NYSE in 2010 which had a volume of more than 1 million. Of those days which a stock had either (a) open = low and close = high or (b) open = high and close = low, which symbol and date has the largest volume traded?
5. Which company (ticker symbol) had the highest net income over all the years that are in the FND table?
6. Which company (ticker symbol) had the highest net income in fiscal year 2011 (use the FND table)?

7. Which company (ticker symbol) had the lowest positive net income over all years (use the FND table)?
8. Which company (ticker symbol), which had a net-income per employee over $1,000, had the largest number of employees (over all years)? Keep units in mind (use the FND table)!
9. Which company (ticker symbol) had the lowest, positive, non-zero, net income in fiscal year 2011 (use the FND table)?
10. Of the companies which had more than 1,000 employees in 2011 which had the highest net income per employee in 2011 (use the FND table)?

5 HW #1B: Basic Pandas

Repeat HW #1A, this time using Pandas. 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 following provides a template that you may wish to use.

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!*

```
import pandas as pd
import numpy as np

df2010 = pd.read_csv( '/Users/ncross/git/sqlnotes/newserver/data/2010.tdf',
        sep='\t', engine='python', names=['symb', 'retdate', 'opn', 'high', 'low',
        'cls', 'vol', 'exch'])

df2011 = pd.read_csv( '/Users/ncross/git/sqlnotes/newserver/data/2011.tdf',
        sep='\t', engine='python', names=['symb', 'retdate',  'opn', 'high', 'low',
        'cls', 'vol', 'exch'])

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'])

## Question #1
ans = df2010.loc[(df2010.loc[:, 'symb']=='AAPL'), :]
print(ans.head())

## Question #2
df2010C = df2010.copy()
df2010C = df2010C.loc[(df2010.loc[:, 'retdate'] =='07-Jan-2010') &
        (df2010.loc[:, 'symb']=='AAPL'), :]
df2010C.loc[:, 'diff'] = df2010C.loc[:, 'opn'] - df2010C.loc[:, 'cls']
ans = df2010C
print(ans.head())
```

First Five

Using the 2010 stocks data, write a query that returns the following.

1. All columns relating to AAPL.
2. All columns from the table and a column with the difference between open and close (open - close) for AAPL on the 7th of January.
3. Write a query which returns the stock symbol, the date, the open and close price for the top five differences (open - close) in 2010 for only those stocks on the New York Stock Exchange (NYSE).
4. The days when AAPL has a volume more than 20 million and where the high is $3 or more dollars greater than the low. Write it twice, once to return a series and once as a dataFrame.
5. Write a query which returns 3 columns: the return date, SYMB and volume, but only for stocks that

have a volume larger than 200 million

Main Problems

1. Write a query which returns all information about about Google (GOOG), NetFlix (NFLX), Amazon (AMZN) and Microsoft (MSFT) in 2010.
2. Write a query which returns the date and symbol of the largest "one-day gainer", that is the stock which has the highest close - open on the NYSE.
3. Write a query which returns the date and symbol of the largest "one-day percentage gainer", that is the stock which has the highest (close - open) / open on the NYSE.
4. Consider stocks on the NYSE which had a volume of more than 1 million. Which stocks (symbol and date) had their open price the same as their low and their closing price the same as their high?
5. Consider stocks on the NYSE which had a volume of more than 1 million. Which stocks (symbol and date) had their closing price the same as their low and their opening price the same as their high?
6. Consider stocks on the NYSE which had a volume of more than 1 million. Of those days which a stock had either (a) open = low and close = high or (b) open = high and close = low, which symbol and date has the largest volume traded?
7. Which company (ticker symbol) had the highest net income over all the years that are in the FND table?
8. Which company (ticker symbol) had the highest net income in fiscal year 2011 (use the FND table)?
9. Which company (ticker symbol) had the lowest, non-zero, net income over all years (use the FND table)?
10. Which company (ticker symbol), which had a net-income per employee over $1,000, had the largest number of employees (over all years)? Keep units in mind (use the FND table)!

Even more problems

1. Which company (ticker symbol) had the lowest, non-zero, net income in fiscal year 2011 (use the FND table)?
2. Of the companies which had more than 1,000 employees in 2011 which had the highest net income per employee in 2011 (use the FND table)?

6 HW #2A: Basic Functions

The following questions utilize the financial data in the s2010, s2011 and fnd tables. Before beginning the assignment, *please read the data dictionary to better understand the data.* When doing so, keep an eye on data types for different columns as well as table organization.

- If no table information is given, use the 2010 data.
- If the query returns a significant number of rows, please only copy a few rows in your response.
- For those queries which require specifying a date, please use the format 'YYYY-MM-DD' (as in '2010-01-11'), making sure to use single quotes around the date itself.

In the problems below you may need to use the following definitions:

- **Profit Margin:** Net Income divided by Revenue.
- **Turnover:** Revenue divided by Inventory.
- **Dollar-volume:** This is the dollar value of stocks traded based on the closing price, so equal to the closing price of the shares traded multiplied by the volume.

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!*

First Five

1. Write a query which returns the date and symbol of the largest "one-day gainer" on the NYSE in 2010, that is the stock which has the highest close - open.
2. Return the symbol, return date and the dollar volume traded for the highest dollar volume traded stocks in 2010 on the NYSE.
3. Using the fnd data, which companies (company name), in fiscal year 2010 had a profit margin greater than 20%, turnover more than 2 and more than 10,000 employees?
4. What are the symbols and dollar volume traded for the companies with the top 5 dollar (based on closing price) volume traded on February 3rd 2010 (NYSE only)?
5. Write a query which returns the stock (symbol only) which has the largest (absolute) difference between high and low price for those stocks which have an absolute difference between their high and low of less than $1 dollar and a volume greater than 5,000 (NYSE only in 2010).

Main Problems

1. The "one-day percentage gain" is equal to $\frac{close-open}{open}$. Write a query which returns the date and symbol of the largest one-day percentage gainer of NYSE stocks in 2010.
2. Write a query which returns the date and symbol of the largest one-day percentage gainer for those stocks on the NYSE whose symbol begins with the letter "R" in 2010.
3. Write a query which returns all stocks (symbol and date) with a one-day percentage gain of more than 70 percent whose symbol either begins with R or ends with C.
4. Write a query which returns the stock (symbol) whose second letter (in their symbol) is "T" and is the largest one-day percentage gainer.

5. For those stocks in fiscal year 2010 with a negative net income, which stock (company name) had the largest amount of inventories (fnd table)?

6. Using the fnd table, write a query which returns the company name and the net income for the stock (in 2010) with the largest net income among those stocks with the phrase "data" (case-insensitive) in the company name.

7. Using the fnd table, write a query which returns the top-5 most profitable (highest net income) companies (and their net income) for those companies with either "bank" or "financial" (case-insensitive) in their company name for fiscal year 2010.

8. Using the fnd table, write a query which returns the minimum of ebitda or net income (call it min_profit) and the company name for companies with "apple" (case-insensitive) in their name. Order the results by number of employees from highest to lowest and only include those companies which have all three numeric columns (ebidta, netinc and emp) present.

9. Using the fnd table, write a query which returns squared difference between ebitda and net income (call it sqr_diff) as well as the company name for companies in fiscal year 2010 whose name includes both a Z and a K, but does not contain a C.

10. Write a query which returns the 2 lowest, positive, net incomes (as well as company names) for those companies in fiscal year 2010 with "ING" in their name where the total number of characters in their name is between 5 and 12 (inclusive).

7 HW #3A: Subqueries

Answer the following questions using only the syntax discussed in class. If a year is unspecified, please use the 2010 data and refer to the data dictionary for questions regarding the contents of each table. Be careful when using the fnd data as many of the items in that dataset are scaled by a factor (e.g. in thousands or millions).

Three terms that are defined in this assignment:

- **Profit Margin:** Net Income divided by Revenue.
- **Turnover:** Revenue divided by Inventory.
- **Dollar-volume:** This is the dollar value of stocks traded based on the closing price, so equal to the closing price of the shares traded multiplied by the volume.

For each question, please provide the query which will generate the result.

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!*

First Five

1. Using the daily stock data from 2010, return a list of the unique trading days in 2010.
2. Using the 2010 data return the stock (symbol), the date and the dollar-volume for the stock with the largest dollar-volume traded on the NYSE (on a single day).
3. Using the 2010 data, return the stock (symbol only) with the largest volume on Jan 11th that also appears on Dec 1st.
4. Using the 2010 data, return the stock symbol and a column called "HFlag" which is equal to 1 if the high - low is greater than 1 and zero otherwise. Only return those companies whose stock symbol begins or ends with "A".
5. Write a query which returns (a) the date, (b) closing price and (c) a flag ("gt30") which is equal to "1" when the closing price is greater than $30.00 and "0" otherwise for "AAPL" in 2010.

Main Problems

1. Return the list of symbols that exist in 2011, but not 2010.[4]
2. Using the fnd data, return company name, year and the a column called "HFlag" which is equal to 1 if the company has a net income larger than $1 Billion dollars and 0 otherwise. Only include those companies whose name begins with "B".
3. Using the fnd data, which ticker symbols have a net income to employee ratio greater than $1,000 in fiscal year 2010 and also have a net income between 20 and 30 million dollars in 2011?
4. The lowest five symbols by volume from Janaury 11th, 2010 that have a volume between 1 million and 10 million on December 1st, 2011. In other words, of those stocks which had between 1 and 10 million shares traded on December 1st, 2011, which five have the lowest volume traded on January 11th, 2010.

[4] If this is slow, try using distinct and see what happens. Any ideas why this may happen?

5. Of the stocks (symbols) that existed in 2011, but not in 2010, which had the highest closing price in 2011?
6. Which symbols were in the top 500 of dollar volume on the 2nd, 3rd and 4th days of February 2011 (The stock needs to be in the top 500 for all days)?
7. Of the symbols that had volume between 100,000 and 1,000,000 on the 2nd and 3rd of February 2011, which had volume greater than 5,000,000 on the 4th on February?
8. Write a query to generate the following dataset:
    - company name, ticker symbol, revenue for all companies whose name begin with "A" or "a"
    - A column, revflag which is 1 if revenue is greater than $25,000,000 and 0 otherwise.
9. Write a query to generate the following dataset:
    - company name, ticker symbol, revenue, inventory and employee information from fiscal year 2010
    - A column called turnflag which is 1 for companies with turnover greater than 2, 0 otherwise
    - For a company to be included it must have revenue, inventory and employee all greater than zero for both 2010 and 2011

Additional Problems

1. Of the stocks (ticker symbols) that have a net income to revenue ratio (called a profit margin) greater than 20%, which have more than 25,000 employees in fiscal year 2011?
2. We define revenue divided by inventory as the turnover. It expresses how many times the inventory has turned-over during the year in the form of sales. For companies (ticker symbols) with revenue between 1 and 2 million dollars in 2010, what company has the highest turnover in 2011?
3. Of the stocks (ticker symbol) that have profit margin greater than 20% in 2010, which had a profit margin greater than 30% in fiscal year 2011?
4. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and more than 1,000 employees in 2011, what is the highest profit margin in fiscal year 2011 and what is the ticker symbol?
5. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and more than 1,000 employees in 2011, what is the lowest profit margin in fiscal year 2011?
6. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and between 1,000 and 2,000 employees in 2011, what is the highest profit margin in fiscal year 2011 and what is the ticker symbol?
7. Of the companies (ticker symbols) with turnover between 1 and 2 in 2010, which companies also had a net income to employee ratio greater than $1,000 in 2010?
8. Of the companies (ticker symbols) with turnover between 1 and 2 in 2010, which companies also had a net income to employee ratio greater than $1,000 in 2011?
9. Write a select statement to generate the following dataset:
    - company name, ticker symbol, revenue, inventory and employee information from both 2010 and 2011 fiscal years.

- A column called invtflag which is equal to 1 for companies with turnover between 2 and 3, 2 for turnover between 3 and 4 and 5 for turnover greater than 4 and zero otherwise.
- A column called invtProfit which is equal to 1 for companies with less than 20% profit margin and turnover greater than 2, 2 for companies with profit margin greater than 40% and turnover greater than 2 and 0 otherwise.
- A column called EmployeeProfit which is equal to 0 for companies that have profit margins between 20% and 40% and have more than 10,000 employees, is equal to a company's profit margin if the margin is less than 20%, is equal to twice the number of employees (if it exists) if the profit margin is greater than 40% and is -1 otherwise.

8 HW #3B: Subqueries in Pandas

Repeat HW #3A, this time using Pandas. 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 same requirements as in HW #1B apply.

Three terms that are defined in this assignment:

- **Profit Margin:** Net Income divided by Revenue.
- **Turnover:** Revenue divided by Inventory.
- **Dollar-volume:** This is the dollar value of stocks traded based on the closing price, so equal to the closing price of the shares traded multiplied by the volume.

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
df2010 = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2010.tdf',
        sep='\t', engine='python', names=['symb', 'retdate', 'opn', 'high', 'low',
        'cls' , 'vol', 'exch'])

df2011 = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2011.tdf',
        sep='\t', engine='python', names=['symb', 'retdate',  'opn', 'high', 'low',
        'cls', 'vol', 'exch'])

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'])
```

First Five

1. Using the daily stock data from 2010, return an array of the unique trading days in 2010.
2. Return the symbol, return date and the dollar volume traded for the highest dollar volume traded stocks in 2010 on the NYSE.
3. Using the 2010 data, return the stock (symbol only) with the largest volume on Jan 11th that also appears on Dec 1st.
4. Using the 2010 data, return the stock symbol and a column called "HFlag" which is equal to 1 if the high - low is greater than 1 and zero otherwise. Only return those companies whose stock symbol begins or ends with "A".
5. Write a query which returns (a) the date, (b) closing price and (c) a flag ("gt30") which is equal to "1" when the closing price is greater than $30.00 and "0" otherwise for "AAPL" in 2010.

Main Problems

1. Return the list of symbols that exist in 2011, but not 2010.

2. Using the fnd data, return company name, year and the a column called "HFlag" which is equal to 1 if the company has a net income larger than $1 Billion dollars and 0 otherwise. Only include those companies whose name begins with "B".

3. Using the fnd data, which ticker symbols have a net income to employee ratio greater than $1,000 in fiscal year 2010 and also have a net income between 20 and 30 million dollars in 2011?

4. Using the fnd data, which companies (company name), in fiscal year 2010 had a profit margin greater than 20%, turnover more than 2 and more than 10,000 employees?

5. The lowest five symbols by volume from Janaury 11th, 2010 that have a volume between 1 million and 10 million on December 1st, 2011. In other words, of those stocks which had between 1 and 10 million shares traded on December 1st, 2011, which five have the lowest volume traded on January 11th, 2010.

6. Of the stocks (symbols) that existed in 2011, but not in 2010, which had the highest closing price in 2011?

7. Which symbols were in the top 500 of dollar volume on the 2nd, 3rd and 4th days of February 2011 (The stock needs to be in the top 500 for all days)?

8. Of the symbols that had volume between 100,000 and 1,000,000 on the 2nd and 3rd of February 2011, which had volume greater than 5,000,000 on the 4th on February?

9. Generate the following dataset:

   - company name, ticker symbol, revenue for all companies whose name begin with "A" or "a"
   - A column, revflag which is 1 if revenue is greater than $25,000,000 and 0 otherwise.

10. Generate the following dataset:

    - company name, ticker symbol, revenue, inventory and employee information from fiscal year 2010
    - A column called turnflag which is 1 for companies with turnover greater than 2, 0 otherwise
    - For a company to be included it must have revenue, inventory and employee all greater than zero for both 2010 and 2011

Additional Problems

1. Of the stocks (ticker symbols) that have a net income to revenue ratio (called a profit margin) greater than 20%, which have more than 25,000 employees in fiscal year 2011?

2. We define revenue divided by inventory as the turnover. It expresses how many times the inventory has turned-over during the year in the form of sales. For companies (ticker symbols) with revenue between 1 and 2 million dollars in 2010, what company has the highest turnover in 2011?

3. Of the stocks (ticker symbol) that have profit margin greater than 20% in 2010, which had a profit margin greater than 30% in fiscal year 2011?

4. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and more than 1,000 employees in 2011, what is the highest profit margin in fiscal year 2011 and what is the ticker symbol?

5. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and more than 1,000 employees in 2011, what is the lowest profit margin in fiscal year 2011?

6. Of the stocks (ticker symbols) that have a net-income to employee ratio greater than $1,000 in fiscal year 2010 and between 1,000 and 2,000 employees in 2011, what is the highest profit margin in fiscal year 2011 and what is the ticker symbol?

7. Of the companies (ticker symbols) with turnover between 1 and 2 in 2010, which companies also had a net income to employee ratio greater than $1,000 in 2010?

8. Of the companies (ticker symbols) with turnover between 1 and 2 in 2010, which companies also had a net income to employee ratio greater than $1,000 in 2011?

9. Write a select statement to generate the following datasets:

   - company name, ticker symbol, revenue, inventory and employee information from both 2010 and 2011 fiscal years.
   - A column called invtflag which is equal to 1 for companies with turnover between 2 and 3, 2 for turnover between 3 and 4 and 5 for turnover greater than 4 and zero otherwise.
   - A column called invtProfit which is equal to 1 for companies with less than 20% profit margin and turnover greater than 2, 2 for companies with profit margin greater than 40% and turnover greater than 2 and 0 otherwise.
   - A column called EmployeeProfit which is equal to 0 for companies that have profit margins between 20% and 40% and have more than 10,000 employees, is equal to a company's profit margin if the margin is less than 20%, is equal to twice the number of employees (if it exists) if the profit margin is greater than 40% and is -1 otherwise.

10. What are the symbols and dollar volume traded for the companies with the top 5 dollar volume traded (based on closing price) on February 3rd 2010 (NYSE only)?

9 HW #4A: Aggregation

Answer the following questions using only the syntax discussed in class. If a year is unspecified, please use the 2010 data and refer to the data dictionary for questions regarding the contents of the data.

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!*

First Five

1. What is the total number of rows in the 2010 database?
2. How many unique symbols are there in 2010?
3. What is the minimum closing price for a stock when it had a volume greater than 1,000,000 shares in 2010?
4. Return the average closing price for all stocks from the NYSE in 2010?
5. Return the average closing price for all stocks and the total number of rows by the exchange of each stock for 2010. Order the results from lowest to highest average closing price. This should return two rows and three columns (exchange, average closing price and total number of rows)

Main Problems

1. Which symbols have less than 50 rows in 2010?
2. How many symbols have less than 50 rows in 2010?
3. Write a query which returns one row and two columns. The first column should contain the number of symbols which have less than 50 rows in 2010 and the second column should have the number of symbols with more than 100 rows in 2010.
4. Write a query which returns two column and two rows. The first column should be named "numtype" which should be equal to "less than 50" or "more than 100" and the second column should have the number of unique symbols which correspond to this condition. In other words, the same numbers as the previous problem, transposed with an column providing a description.
5. Write a query which returns three rows and two columns. The first column should contain the average yearly total traded volume for symbols which had (1) more than 100 trading days (2) less than 50 trading days and (3) between 50 and 100 trading days. The other column should identify each row and be called "numType."
6. Write a query which returns three rows and two columns. The first column should contain the average *daily* traded volume for symbols which had (1) more than 100 trading days (2) less than 50 trading days and (3) between 50 and 100 trading days. The other column should identify each row and be called "numType."
7. How many of the symbols had a day where the dollar volume (closing price multiplied by number of shares traded) was greater than 100 million dollars in 2010?
8. What percentage of the symbols had a day where the dollar volume of shares traded was greater than 100 million dollars in 2010?

9. Using only the SUM, AVG and COUNT aggregate functions, compute the covariance between the the closing price and volume in 2010.

10 HW #4B: Aggregation in Pandas

Repeat HW #4A, this time using Pandas. 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.

Answer the following questions using only the syntax discussed in class. If a year is unspecified, please use the 2010 data and refer to the data dictionary for questions regarding the contents of the data.

Here are the statements that will load the data, note that you will need to change the directory.

```
## Initial Information
import pandas as pd
import numpy as np
df2010 = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2010.tdf',
        sep='\t', engine='python', names=['symb', 'retdate', 'opn', 'high', 'low',
        'cls' , 'vol', 'exch'])

df2011 = pd.read_csv('/Users/ncross/git/sqlnotes/newserver/data/2011.tdf',
        sep='\t', engine='python', names=['symb', 'retdate',  'opn', 'high', 'low',
        'cls', 'vol', 'exch'])

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'])
```

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!*

First Five

1. What is the total number of rows in the 2010 database? Return this as an integer.
2. How many unique symbols are there in 2010? Return this as an integer.
3. What is the minimum closing price for a stock when it had a volume greater than 1,000,000 shares in 2010?
4. Return the average closing price for all stocks from the NYSE in 2010?
5. Return the average closing price for all stocks and the total number of rows by the exchange of each stock for 2010. Order the results from lowest to highest average closing price. This should return two rows and three index/value columns (exchange, average closing price and total number of rows).

Main Problems

1. Which symbols have less than 50 rows in 2010?
2. How many symbols have less than 50 rows in 2010?
3. Write a query which returns one row and two columns (DataFrame or Series). The first column should contain the number of symbols which have less than 50 rows in 2010 and the second column should have the number of symbols with more than 100 rows in 2010.

4. Write a query which returns two column and two rows (either series or a DataFrame). The first column should be equal to "lessThan50" or "moreThan100" and the second column should have the number of unique symbols which correspond to this condition. In other words, the same numbers as the previous problem, transposed with an column providing a description.

5. Write a query which returns three rows and two columns (note that one column maybe an index). One column should contain the average yearly total traded volume for symbols which had (1) more than 100 trading days (2) less than 50 trading days and (3) between 50 and 100 trading days. The other column should identify each row and be called "numType."

6. Write a query which returns three rows and two columns. The first column should contain the average *daily* traded volume for symbols which had (1) more than 100 trading days (2) less than 50 trading days and (3) between 50 and 100 trading days. The other column should identify each row and be called "numType."

7. How many of the symbols had a day where the dollar volume (closing price multiplied by number of shares traded) was greater than 100 million dollars in 2010?

8. What percentage of the symbols had a day where the dollar volume of shares traded was greater than 100 million dollars in 2010?

11 HW #5A: Aggregate Functions and Dates

Answer the following questions using only the syntax discussed in class. If a year is unspecified, please use the 2010 data and refer to the data dictionary for questions regarding the contents of the data.

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!*

First Five

The queries below rely on information from the 2010 stock return data.

1. Which day of the week (0,1,2,...) had the largest number of shares traded?
2. Which day of the week (0,1,2,..) has the highest average shares traded?
3. Which day of the week-month (January-Monday, January-Tuesday, etc.) combination had the highest average return (close - open)? Note that both day of the week and month can be kept as integers.
4. Write a query which returns 3 columns and 5 rows with each row should represent a day of the week. One column should be the English day of the week ("Monday,", "Tuesday," etc.) while the next column should be equal to the average number of shares traded on that day from stocks that have a volume traded between 1 million and 2 million shares on that day ("C2"). The final column ("C3") should be the average number of shares traded on that day from stocks that had a volume traded outside of 1 million to 2 million.
5. Write a query which returns the maximum closing price for each symbol in 2010, sorting the results from from high-to-low closing price.

Main Questions

1. Which quarter in 2010 has the most trading days?[5]
2. Write a query which returns symbol and a column "DFlag", which is equal to 1 if the max closing price in 2010 is larger than 100, 2 if the max closing price in 2010 is between 50 and 100 and 3 if the max closing price is less than 50. There should be one row per symbol.
3. Write a query which returns the number of distinct symbols of each type of Dflag (from the previous problem). This should be 3 rows and 2 columns (one of the columns should indicate what each row means).
4. Write a query which returns the number of distinct symbols of each type of Dflag (from the previous problems), this should be 3 columns and a single row.
5. Calculate the number of distinct trading days per month in 2010. This should return 12 rows with 2 columns.
6. For each symbol, calculate the difference between the maximum and minimum closing price for December, 2010. Only include those stocks with 22 observations (there are 22 trading days in December, 2010).

[5] Define Q1 as Jan-Mar, Q2 as Apr-Jun, etc.

7. Calculate the average difference between the maximum and minimum closing price for Tuesdays in January, 2010 for stocks on NYSE. The max and min should be calculated per-stock and then averaged. Only include those stocks with 4 observations which fulfill the criteria.[6]

8. Calculate the average closing price for Tuesday in January 2010 from the NYSE. Only include those stocks with 4 observations which fulfill the criteria. In other words, calculate the average price for each stock and then take the average of that number.

9. Calculate the average closing price for all stocks on the NYSE, by month, in 2010. Only include those stocks which have a closing price greater than \$100 in *2011*.

10. Calculate the average closing price in 2010 for all stocks (NYSE only) which are "not extreme". We define a stock as not extreme if the closing price is less than .1% of the max closing price (for all stocks) for the entire year. In other words, identify those stocks which are not extreme and then calculate their average price.

[6]There are 4 Tuesday trading days in January, 2010.

12 HW #5B: Aggregate Functions and Dates

Repeat HW #5A, this time using Pandas. 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 are different than the previous commands because they 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'])
```

First Five

1. Which day of the week (0,1,2,...) had the largest number of shares traded?
2. Which day of the week (0,1,2,..) has the highest average shares traded?
3. Which day of the week-month (January-Monday, January-Tuesday, etc.) combination had the highest average return (close - open)? Note that both day of the week and month can be kept as integers.
4. Write a query which returns 3 columns and 5 rows with each row should represent a day of the week. One column should be the english day of the week ("Monday,", "Tuesday," etc.) while the next column should be equal to the average number of shares traded on that day from stocks that have a volume traded between 1 million and 2 million shares on that day ("C2"). The final column ("C3") should be the average number of shares traded on that day from stocks that had a volume traded outside of 1 million to 2 million.
5. Write a query which returns the maximum closing price for each symbol in 2010, sorting the results the final table from from high-to-low closing price.

Main Questions

1. Which quarter in 2010 has the most trading days?[7]

[7]Define Q1 as Jan-Mar, Q2 as Apr-Jun, etc.

2. Write a query which returns symbol and a column "DFlag", which is equal to 1 if the max closing price in 2010 is larger than 100, 2 if the max closing price in 2010 is between 50 and 100 and 3 if the max closing price is less than 50. There should be one row per symbol.

3. Write a query which returns the number of distinct symbols of each type of Dflag (from the previous problem). This should be 3 rows and 2 columns (one of the columns should indicate what each row means).

4. Write a query which returns the number of distinct symbols of each type of Dflag (from the previous problems), this should be 3 columns and a single row.

5. Calculate the number of distinct trading days per month in 2010. This should return 12 rows with 2 columns.

6. For each symbol, calculate the difference between the maximum and minimum closing price for December, 2010. Only include those stocks with 22 observations (there are 22 trading days in December, 2010).

7. Calculate the average difference between the maximum and minimum closing price for Tuesdays in January, 2010 for stocks on NYSE. The max and min should be calculated per-stock and then averaged. Only include those stocks with 4 observations which fulfill the criteria.[8]

8. Calculate the average closing price for Tuesday in January 2010 from the NYSE. Only include those stocks with 4 observations which fulfill the criteria. In other words, calculate the average price for each stock and then take the average of that number.

9. Calculate the average closing price for all stocks on the NYSE, by month, in 2010. Only include those stocks which have a closing price greater than \$100 in *2011*.

10. Calculate the average closing price in 2010 for all stocks (NYSE only) which are "not extreme". We define a stock as not extreme if the closing price is less than .1% of the max closing price (for all stocks) for the entire year. In other words, identify those stocks which are not extreme and then calculate their average price.

[8]There are 4 Tuesday trading days in January, 2010.

13 HW #6A: SQL Joins (I)

The following questions utilize the financial data in the s2010, s2011 and fnd tables. Before beginning the assignment, *please read the data dictionary to better understand the data.* When doing so, keep an eye on data types for different columns as well as table organization.

- If no table information is given, use the 2010 data.
- If the query returns a significant number of rows, please only copy a few rows in your response.

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!*

First Five

1. Using a JOIN, create a dataset which contains symbol, the max closing price for that symbol from 2010 and the max closing price for that symbol from 2011. This should only include those symbols which are in both 2010 and 2011. Are you sure that both sides are unique? Why?
2. Using a LEFT JOIN, create a dataset which contains the following information: symbol, the last day it is traded in 2011 and the last day it is traded in 2010. Make sure to include all rows from 2011 and only those matching from 2010. There should be one row per symbol.
3. Using a cross join, create a dataset which contains every possible combination of symbol (in 2010) and return date (in 2010).
4. Write a query which returns the number of rows in the above query. How does this compare to the number of rows in the 2010 dataset? Does this make sense?
5. Write a query which has 12 rows and 3 columns. The first column should be Month (1,2,3...,12) the second column should be the number of rows from that month in 2010 and the third column should be the number of rows from that month in 2011.

Main Problems

1. Using a LEFT JOIN, count the number of symbols which are in 2010, but not in 2011.
2. For each symbol, return the closing price on the first day that it is traded in 2010.
3. For each symbol, return the closing price on both the first day and last day that it is traded in 2010.
4. Create a dataset which contains 4 columns: the symbol, the retdate, the closing price and the closing price on the day after. Note that this dataset should *only* include Monday to Tuesday transitions, so retdate there should only be one row per-symbol per-Monday in the dataset. Specifically, if there are 50 trading weeks in a year and assuming that a symbol is traded every day, there would be 50 observations for that symbol
5. By matching the fnd data and the stocks 2010 data create a table which contains three columns and one row. The columns should represent the number of *unique* symbols which (a) are in both datasets, (b) are only in the 2010 dataset and (c) are only in the fnd data. Make sure to ignore all observations which are missing ticker symbols.

6. By combining the fnd and the stocks 2010 data, generate a dataset which contains the number of unique symbols of each of the three types in the previous problem. This time return two columns and three rows (one of the columns should describe what data is in the row).

7. Create a dataset which is 5 rows by 3 columns. The first column should be DOW, the second column should be the average closing price of all stocks from 2010 on that day of the week and the third should be the average price of all stocks from 2011 for that day of the week.

8. We want to divide all stocks by the following criteria: if their max closing price in 2010 was less than 50, between 50 and 100 (inclusive) and more than 100. Return a table which contains the average net income (from fyear 2010) for each type of stock. Note that net income can be found in the fnd table and, if there are two net-income values for a particular ticker symbol, take the max. Only include those symbols in both datasets (fnd and s2010) that do not have a missing net income.

Extra Problems

1. Create a dataset which contains the first day that each symbol is traded in 2010, the last day that the symbol is traded in 2011 and only includes those symbols which are in both 2010 and 2011.

2. For those symbols which had a closing price larger than \$100 *anytime* in 2010, return the symbol, first day that it was traded in 2010 and all the dates that it had a closing price larger than \$200 in 2010. If the symbol was never above \$200, return no rows for it.

3. What are the first and last date listed for each symbol in 2010? Be careful to return this for *each* symb.

4. For each symbol that appears anywhere in 2010, calculate the number of missing trading days that it has in each month in 2010. This should return three columns: symbol, month, number of missing values.

5. Create a dataset which is 10 rows by 3 columns. The first column should be the year, the second column should be the day-of-the-week and the third column should be the average closing price of all stocks for that day-of-the-week. Include both 2010 and 2011.

6. How many cars (total), on an average *day*, go through each toll plaza in both directions combined (return a row for each toll plaza)? Make sure to sum up to the *day* level before computing the average.

7. Which day-of-the-week (Monday, Tuesday, etc.) has the highest number of cars going through Plaza #1, both directions combined, with EZ pass? This should be the total number of cars over the entire time period in the dataset.

8. Which day-of-the-week-plaza combination has the lowest percentage of users cars using the EZ pass in the outbound direction? In other words, if you look at outbound cars through each plaza, which day of the week has the lowest percentage of cars using EZ pass. You can compute the percentage over the entire time period.

9. Calculate the average number of cars going through Plaza #1, outbound, with EZ pass for each day-of-the-week. This should be a *daily* average and should return 7 rows.

10. In an average week on Plaza #1 with EZ pass (outbound), what percentage of cars go through each day? (E.g. basically the above, but this time percent of total).

11. For each plaza, what was the change (percent) in average number of cars on a Monday using EZ-pass in both directions, between 2015 and 2016? (Calculate the average number of cars for a Monday in 2015 and 2016 and then calculate the percentage change based off of that.)

12. Calculate, for each hour, plaza and day-of-the-week (so $7 \cdot 24$ rows per plaza), the ratio of inbound to outbound traffic.

13. Using a join, create a dataset with three columns and 7 rows. The first column should be the DOW, the second column should be the average number of cars, per-day-of-the-week, through toll Plaza #1 in either direction with an EZ pass in 2016 and the final column should be the average number of cars, per-day-of-the-week, through tool Plaza #2 with an EZ pass in 2015.

14. Create a dataset which contains twenty-four rows and two columns. The first column represents the hour and the second column represents the max number of EZ pass cars, during that hour, outbound, through Plaza #1.

15. Create a dataset which contains 24x7 rows and two columns. The first column represents the DOW-hour combination (you may need to combine two columns using "||" or the concatenate operator) and the second represents the max number of EZ pass cars, during that hour-day, through Plaza #1 in the outbound direction.

16. Using at least one join, create a dataset which contains twenty-four rows and 4 columns. Each row should represent an hour, and the first column should be an hour identifier. Column #2 should contain the maximum number of EZ pass cars, in the inbound direction, through Plaza #1 during that hour, Column #3 should contain the minimum number of outbound EZ-pass cars, during that hour, through Plaza #2 and Column #3 should be the maximum number of EZ-pass cars in either directions combined, during that hour, on Plaza 3.

17. Create a dataset which contains the following columns: hour, day-of-the-week, plaza, the ratio of inbound to outbound traffic in 2014 and the ratio of inbound to outbound traffic in 2013.

18. For the day with the most traffic (inbound, outbound and both payment types combined), calculate the ratio of inbound to outbound traffic over the entire dataset (not by plaza), for each hour. Return three columns, the day-of-the-week of that date, hour and the percent for that hour.

14 HW #6B: Pandas Joins (I)

Repeat HW #6A, this time using Pandas. 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. Using a join, create a dataset which contains symbol, the max closing price for that symbol from 2010 and the max closing price for that symbol from 2011. This should only include those symbol which are in both 2010 and 2011. Are you sure that both sides are unique? Why?

2. Using a LEFT JOIN, create a dataset which contains the following information: symbol, the last day it is traded in 2011 and the last day it is traded in 2010. Make sure to include all rows from 2011 and only those matching from 2010. There should be one row per symbol.

3. Using a cross join, create a dataset which contains every possible combination of symbol (in 2010) and return date (in 2010).

4. Write a query which returns the number of rows in the above query. How does this compare to the number of rows in the 2010 dataset? Does this make sense?

5. Write a query which has 12 rows and 3 columns. The first column should be Month (1,2,3...,12) the

second column should be the number of rows from that month in 2010 and the third column should be the number of rows from that month in 2011.

Main Problems

1. Using a LEFT JOIN, count the number of symbols which are in 2010, but not in 2011.
2. For each symbol, return the closing price on the first day that it is traded in 2010.
3. For each symbol, return the closing price on both the first day and last day that it is traded in 2010.
4. Create a dataset which contains 4 columns: the symbol, the retdate, the closing price and the closing price on the day after. Note that this dataset should *only* include Monday to Tuesday transitions, so retdate there should only be one row per-symbol per-Monday in the dataset. Specifically, if there are 50 trading weeks in a year and assuming that a symbol is traded every day, there would be 50 observations for that symbol
5. By matching the fnd data and the stocks 2010 data create a table which contains three columns and one row. The columns should represent the number of *unique* symbols which (a) are in both datasets, (b) are only in the 2010 dataset and (c) are only in the fnd data. Make sure to ignore all observations which are missing ticker symbols.
6. By combining the fnd and the stocks 2010 data, generate a dataset which contains the number of unique symbols of each of the three types in the previous problem. This time return two column and three rows (one of the columns should describe what data is in the row).
7. Create a dataset which is 5 rows by 3 columns. The first column should be DOW, the second column should be the average closing price of all stocks from 2010 on that day of the week and the third should be the average price of all stocks from 2011 for that day of the week.
8. We want to divide all stocks by the following criteria: if their max closing price in 2010 was less than 50, between 50 and 100 (inclusive) and more than 100. Return a table which contains the average net income (from fyear 2010) for each type of stock. Note that net income can be found in the fnd table and, if there are two net-income values for a particular ticker symbol, take the max. Only include those symbols in both datasets (fnd and s2010) and that do not have a missing net income.

Extra Problems

1. Create a dataset which contains the first day that each symbol is traded in 2010, the last day that the symbol is traded in 2011 and only includes those symbols which are in both 2010 and 2011.
2. For those symbols which had a closing price larger than \$100 *anytime* in 2010, return the symbol, the first day that it was traded in 2010 and all the dates that it had a closing price larger than \$200 in 2010. If the symbol was never above \$200, return no rows for it.
3. What are the first and last date listed for each symbol in 2010? Be careful to return this for *each* symb.
4. For each symbol that appears anywhere in 2010, calculate the number of missing trading days that it has in each month in 2010. This should return three columns: symbol, month, number of missing values.
5. Create a dataset which is 10 rows by 3 columns. The first column should be the year, the second column should be the day-of-the-week and the third column should be the average closing price of all stocks for that day-of-the-week. Include both 2010 and 2011.

15 HW #7A: SQL Joins (II)

Please answer the following questions, making sure to only use the syntax from class.[9]

Before beginning the assignment make sure that you have indexes applied to the symbol and return date variables in both stock tables (otherwise the queries will take an eternity). The following commands will create the indexes necessary to complete the assignment.

```
create index s2010_symb_retdate on stocks.s2010 (symb, retdate);
create index s2011_symb_retdate on stocks.s2011 (symb, retdate);
```

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!*

First Five

1. Write a query which returns the symbol, date, volume and the total volume traded for that stock up to (but not including) that *day's* volume. Only consider 2010 data. Make sure that if there is no previous volume that the cumulative volume is set to *zero* and isn't null.
2. Write a query which returns the number of days that the stock has been traded, cumulatively, in 2010. Specifically, the first time that the stock appears it should be "1", the second date that it exists it should be "2". This should return a table with 3 columns (symb, retdate and cumulative trading days) and should have the same number of rows as the original s2010 dataset.
3. For each exchange in 2010 return the stock with the highest total traded volume in the year 2010. This should return two rows (one for each exchange) and 3 columns (exchange name, symbol and total volume traded for that year)
4. For each exchange in 2010 return the stock with the *second* highest total traded volume in the year 2010. This should return two rows (one for each exchange) and 3 columns (exchange name, symbol and total volume traded for that year)
5. Create a dataset which contains the following columns: symbol, date that the symbol first appears in 2010 and the total volume traded in the first 35 days that the stock is traded in 2010. If a stock has a first date late in the year, ignore the spill over into 2011.

Main Problems

1. Using the data from 2010, write a query which returns the seven day moving average for each stock's closing price. Only look at stocks whose symbols begin with the letter "A". Note that this should *not* be the last seven points, but instead the last seven *days*, not including the current day.
2. For each stock from 2010, write a query which returns the symbol, the closing price, the return date and the closing price on the previous day it was traded. Note that you just want to take the price from the previous row, if the rows are ordered by return date. Also, only do this for stocks that begin with the letter 'A'.
3. For each symbol in 2010, return the day(s) where it has its highest volume traded[10]

[9] Specifically if you decided to look ahead, analytic functions are not to be used to answer these questions.
[10] There could be multiple days for a symb.

4. Using only a single join, for each symbol, return the closing price on the first and last day that the stock is traded in 2010.
5. How many missing days are there in total? Make sure to only count missing days **after** a symbol has been in the data. So if a stock doesn't appear in the data until February, January does not count as missing. If a stock leaves the market before the end of the year, you can either count the days past their exit as missing or as not missing, just be consistent across all stocks.[11]
6. For each symbol that appears in 2011, calculate the number of missing trading days that it has in January 2010.
7. Write a query which returns the userid, trans_dt, amt, and the total amount the user has spent up to (but not including) that *day's* transaction. Make sure that if there is no previous transaction the amount is set to *zero* and isn't null.
8. Write a query which returns a purchase number for each order. In other words, for each row return the amount, userid, date and the number of the sale, incrementing from one for each order.
9. For each local in the table, return the most common month of an order.
10. For each local in the table, return the *second* most common month of an order.
11. Create a dataset which has the following information: (1) userid, (2) date of first transaction and (3) number of transactions within the first 35 days of their first transaction.

[11] In other words, if a stock leaves the data it maybe because the stock delisted, in which case the data is not missing.

16 HW #7B: Pandas Joins (II)

NEEDS TO BE REWRITTEN AK

Repeat HW #7A, this time using Pandas. 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. Write a query which returns the userid, trans_dt, amt and the total amount that the user has spent up to (but not including) that *day's* transaction. Make sure that if there is no previous transaction that the amount is set to *zero* and isn't null.

2. Write a query which returns a purchase number for each order. In other words, for each row return the amount, userid, date and the number of the sale, incrementing from one for each order.

3. For each local in the table, return the most common month of an order.

4. For each local in the table, return the *second* most common month of an order.

5. Create a dataset which has the following information: (1) userid, (2) date of first transaction and (3) number of transactions within the first 35 days of their first transaction.

Main Problems

1. Using the data from 2010, write a query which returns the seven day moving average for each stock's closing price. Only look at stocks whose symbols begin with the letter "A". Note that this should *not* be the last seven points, but instead the last seven *days*, not including the current day.
2. For each stock from 2010, write a query which returns the symbol, the closing price, the return date and the closing price on the previous day it was traded. Note that you just want to take the price from the previous row, if the rows are ordered by return date. Also, only do this for stocks that begin with the letter 'A'.
3. For each symbol in 2010, return the day(s) where it has its highest volume traded[12]
4. Return the closing price on the first and last day that the stock is traded in 2010.
5. How many missing days are there total? Make sure to only count missing days **after** a symbol has been in the data. So if a stock doesn't appear in the data until February, January does not count as missing. If a stock leaves the market before the end of the year, you can either count the days past their exit as missing or as not missing, just be consistent across all stocks.[13]
6. For each symbol that appears in 2011, calculate the number of missing trading days that it has in January 2010.

[12] There could be multiple days for a symb.

[13] In other words, if a stock leaves the data it maybe because the stock delisted, in which case the data is not missing.

17 HW #8A: SQL Window Functions: OLD

THIS ONE IS REPLACED AND NEEDS TO BE FIXED

Using only the functions and syntax that we have learned in class, please provide a query to answer the following questions. If a dataset is not specified, please use the 2010 dataset. **Do not create any tables or views.**

Before beginning the assignment, *please read the data dictionary to better understand the data.* When doing so, keep an eye on data types for different columns as well as table organization.

- If no year information is provided for a financial question, assume 2010.
- If the query returns a significant number of rows, please only copy a few rows in your response.

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!*

First Five

1. What is the median daily closing price on January 4th, 2010?
2. 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.
3. Write a query which returns five columns: symbol, return date, closing price, the moving average of the price (covering the last two days it was traded, but not including the current day) and the difference between that moving average and the current price.
4. Using the FND data and an analytic function, return the number of stocks alphabetically before each stock (e.g. if "A" was the first company would be 0, AA would be 1, etc.) in 2010. Make sure to only include each name once. Feel free to include or exclude company names that begin with a number.[14]
5. Without using an analytic function answer the same question as above.

Main Problems

1. For each stock in 2010, return the largest average daily return ((close - open)/open) by the first letter of the symbol. In other words, there should be one row for each first letter of the ticker symbol. The dataset should return (a) the symbol which has the largest, (b) the total number of symbols which begin with that character and (c) the average daily return for the symbol.
2. Repeat the above, this time without using an analytic function. Make sure that you *aren't* joining on a float as joining on a floating variable will lead to uneven results.
3. For stocks in 2010, write a query which creates a dataset containing closing price, symbol, retdate and the nominal change between yesterday's stock (symbol) price and today. If there is a missing day then the nominal change should be missing (which is different from the above question).

[14] Comparisons of the form *string* $\leq$ *string* do alphabetical comparison. Also keep in mind that you can join using any conditional expression.

4. Return a dataset which contains symbol and the number of trading days that the price is within 10% of the max price for that year for each stock. For example, if the max price of a symbol is $100, then return the number of times that price of that stock is ≥ 90.
5. For each symbol return the number of days it took to reach its maximum closing price for that year. If a stock is not traded on a day, then that should *not* count toward the total days. Note that there should be one row per symbol in the final dataset.
6. Repeat the previous question without using any analytic functions.
7. For each stock symbol return the number of days it took to reach its maximum closing price for that year. If a stock is not traded on a day, then it should count toward the total days.

Extra Problems

1. In the Transaction data, what percentage of users, who start by purchasing a Unit end up Subscribing?
2. What percentage of users, in the transaction data, purchase both a Unit and a subscription?
3. What is the average amount of time between Unit Purchases?
4. Calculate the 25, 50 and 75 percentile of the amount of revenue generated in the first six months (per user). Only include those users who made their first purchase more than six months ago. Make sure that this query moves with time: if I run this query next month it should return updated data.

18 HW #8A: SQL Window Functions

Using only the functions and syntax that we have learned in class, please provide a query to answer the following questions. If a dataset is not specified, please use the 2010 dataset. **Do not create any tables or views.**

Before beginning the assignment, *please read the data dictionary to better understand the data.* When doing so, keep an eye on data types for different columns as well as table organization.

- If no year information is provided for a financial question, assume 2010.
- If the query returns a significant number of rows, please only copy a few rows in your response.

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!*

First Five

1. Write a query which returns, for stocks in 2010, the symbol, the date and the cumulative sum of traded volume for that stock from the start of the year to that date, including that date.
2. Repeat the above without an analytic function.
3. Write a query which returns, for stocks in 2010, the symbol, the date and the cumulative sum of traded volume for that stock from the start of the year to that date, *not* including that date.
4. Repeat the above without an analytic function.
5. Write a query which returns, for the stocks in 2010, the symbol, the date, and the moving average of the last five days (including the current date) of closing prices for that stock.

Main Problems

1. Write a query which returns, for stocks in 2010, the symbol, the date and the cumulative sum of traded volume for that stock from the start of the *current month*, including that date.
2. Repeat the above without an analytic function.
3. Write a query which returns, for stocks in 2010, the symbol, the date, the ratio of that days stock closing price to the stock's closing price on the first day that the stock is traded that year ($\frac{current\_price}{first\_price}$)
4. Write a query which returns, for stocks in 2010, the symbol, the date and the difference between the max closing price that the stock achieves in 2010 and the current day's closing price.
5. What is the median closing price for all stocks on January 4th, 2010?
6. For stocks in 2010, write a query which returns the 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.
7. Write a query which returns, for stocks in 2010, a set of unique symbols and a column which is the alphabetical rank (e.g. so "A" should be 1, "AA" should be 2, etc.)

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.

20 BART Project

The objective of this assignment is to create a Python script which loads the BART data into your local database. In order to receive full credit on this assignment you will need to write a Python Script which takes the raw Excel files and loads the "core" ridership data.[15]

The table that holds the data should have the following form:

```
CREATE TABLE cls.bart (
        mon int
        , yr int
        , daytype varchar(15)
        , start varchar(2)
        , term varchar(2)
        , riders float
);
```

Requirements:

- Your code should be callable from a *single* function. While you can have multiple functions (or use objects), the entire script should be run via a single command.
- The code should be in a single text file. No notebooks.
- The code should be robust to being run more than once. If the code is run twice in a row, it should not break, crash or duplicate the data in the database.
- For older time periods the clipper/fastpass data may be broken out, just use the main data and ignore the clipper data.
- You should assume that the code is going to be run on a clean computer. Any implied file structure or libraries that need to be present should be removed.
- The overall structure of the program should be as follows:
  1. Assume that all of the zip files are in a single directory (*dataDir*), which is taken as a parameter in the function.
  2. The code should unzip the files into a directory (*tmpDir*).
  3. The code should process the Excel files, extracting necessary data and reshaping it so that it can be loaded.
  4. A table should be created in your database.
  5. The clean, reshaped and standardized data should then be copied in.
- Things that you will need to standardize:
  - The format for year and month changes over time. Your code should standardize these changes.
  - The number of stations changes over time. If a particular file does not have a station, there is no need to add it.
  - The daytypes ("Weekday", "Saturday" and "Sunday") change their names throughout the data. Make sure that they are standardized. You can ignore the phrase "adjustments." The data was calculated the same way over the entire time period.

[15] For more information about the BART data, please look at `https://www.bart.gov/about/reports/ridership`

- You can assume that the schema has already been created, but you will need to handle the table creation yourself.
- You need to verify that you only load the appropriate files. In other words, make sure to either track files through the process or delete everything within the temp directory before placing files in it.
- Don't use Pandas. It's janky.
- Note that the data in the Excel spreadsheets is presented in a wide format – each column represents the average exits for a particular station. The target table ("cls.bart") is long, not wide; the data will require reshaping before it is copied in.
- The function *ProcessBart* should be called in the following manner:

  ```
  ProcessBart( tmpDir, dataDir, SQLConn=None, schema='cls', table='bart')
  ```

  the parameters of the function:

  - *tmpDir*: Directory where the unzipped files should be stored.
  - *dataDir*: Directory where the zipped files are stored.
  - *SQLConn*: Psycopg2 connection.
  - *schemaName*: The schema where the data should be loaded.
  - *tableName*: The table where the data should be loaded.
- By "core" data, I mean the Weekday, Saturday and Sunday data. Note that in many of the files there are secondary tables or sheets. For example, in the January 2011 data on the "Weekday OD" sheet, the only information that should be copied is B3:AR45.
- Think hard about what code can be repeated and what code should be put into loops or turned into functions. Needlessly repetitive code will be penalized.

Hints:

- Libraries that I used when writing this code:
  - Psycopg2
  - glob
  - xlrd
  - zipfile
  - os
  - shutil

  You can use any other library that can be installed via pip.
- Think hard about what needs to be standardized between years. The difficult part of this code is creating a data structure that allows you to iterate over the years smoothly.
- Please use psycopg2 in order to interface with the database.
- In my code, the create table (using psycopg2) looks like:

```
## Load into DB
SQLCursor = SQLConn.cursor()
SQLCursor.execute("""
  CREATE TABLE %s.%s
  (
  mon int
  , yr int
  , daytype varchar(15)
  , start varchar(2)
  , term varchar(2)
  , riders float
  );""" %(SchemaName, TableName))
  SQLCursor.execute("""COPY %s.%s FROM '%s' CSV;"""
          % (SchemaName, TableName, tmpDir + 'toLoad.csv'))
  SQLConn.commit()
```

- Note that I created a CSV file, "toLoad.csv" inside *tmpDir* to put the formatted and reshaped data.
- Finally, when I grade this code, I am going to download your python script to my personal computer. I will then append the following to your script and run it.

```
LCLconnR = psycopg2.connect("dbname='ncross' user='ncross'
         host='localhost' password='XXX'")

ProcessBart( '\home\ncross\tmp\', '\home\ncross\BART\',
         SQLConn=LCLconnR, schema='cls', table='bart')
```

  Assuming that your code runs (and I hope it does), I will then run 3-5 SQL queries on the resulting data to verify that it loaded completely and correctly.

- I will also be reading over the code itself. While I do not expect you to be Python wizards, I do expect you to be able to code efficiently. This means using loops, functions and variables to create well-written code that also contains comments to include readability.
- Please make sure that the code removes files from the temp directory before trying to load or only works on specific files that you choose. If a file is in that directory that you do not expect it should not cause your code to fail.

21 HW #5A: Info Schema and Price-Volume Relationship*

First Five

Using the information schema answer the following questions:

1. Write a query which returns the count of data types (int, float, etc.) of each columns in the stocks schema.
2. Write a query which returns the number of distinct column types in the entire database.
3. Write a query which returns 3 columns: schema name, column data type, and the number of columns in that schema of that column type.
4. Rewrite the above query in a wide-format. Each row should represent a single schema.
5. Create a pie chart of the above information for the schema "information_schema". Which data format (wide or long) did you use?

In the following exercise, we will investigate the relationship between the dollar volume of shares traded and the returns of that company. Exploring the relationship between dollar volume and return:

1. Write a query which returns the return rounded to the nearly thousandth of a percent while dealing with any data issues. Return the data in hundredths, so if the return is .037123, 3.7 should be returned. Include the dollar volume of stocks traded that day, rounded to the nearest 1,000. Also, only take a 1/16 sample using the following where statement:

   ```
   where md5( permno::varchar(100) ) like '0%'
   ```

2. Create a scatter plot of your rounded returns vs. the rounded dollar volume.
3. Run simple linear regression on the rounded returns vs. the rounded dollar volume and report the results. Do you believe that there is a relationship between trading size and dollar volume traded?
4. Recreate the scatter plot making sure to remove days with less than 250 million shares traded and only include returns between -10 and 10. Did the pattern change?
5. Run simple linear regression on the rounded returns vs. the rounded dollar volume and report the results for the sample of more than 250 million shares and returns between -10 and 10. Do you believe that there is a relationship between trading size and volume traded?
6. Using only the SUM, AVG and COUNT aggregate functions, compute the variance of both the rounded volume and the rounded returns of the sample.
7. The problem below is from the analytic function lecture and should be incorporated in to the LTV estimate. Write a query which returns the following information. Cohort should be defined monthly.
   (a) For each *complete* month, calculate the percentage revenue generated, per cohort, when compared to the previous month. For example, if Month #2 after first purchase the amount of revenue generated is equal to $12,755.54 and the amount of money generated in Month #1 after purchase is equal to $24,885.32 then return $\frac{12{,}755.54}{24{,}885.32} = .513376$[16]
   (b) Average this over all the cohorts with complete month data. Be careful to only consider dates that are complete from both the start and end of the table.
   (c) This should return a set of month-over-month multipliers that could be used to estimate the expected revenue generated from a new cohort. Explain how these numbers could be used to

[16] A complete month is one that is 100% in the data. For example, if a company launches on January 12, 2017 then January is not a complete month. Similarly, if today's date is September 19th, then September is not a complete month.

estimate the lifetime value of a customer (in their first year). E.g. If a customer generated $1 of revenue in their first month, what would you do with those multipliers to estimate the lifetime value?