HTML Parsing of 10-K SEC Filings for Nike, Apple, United Health, Home Depot & Storing in Mongo DB

By

Jigar Mehta, Jaya Manish, Taj Pirzada, Preethi Bojja

SEC filings are the financial statements that give detailed information about the company's fiscal and financial aspects that gives investors and financial professionals' overview of company's operations in the past and glimpse of it in future. 10-K filings are extensively used for the analysis of company's performance. These annual filings have information about the risks, revenue trend over years, mergers and acquisitions, profit and loss etc., the study of which gives clear picture of the company's health.

Objective: In order to study the SEC filings and understand the financial aspects of company based on these reports, 10-K filings of

- **Nike Inc.**
- **Apple Inc.**
- **Home Depot Inc.**
- **United Health Group Inc.**

are considered from year 2009-2015. Complete filings of four companies for all the years are stored in the NoSQL database Mongo dB and company's health is measured based on financial data that is parsed from html files. Below sections gives detailed description of the process involved.

A. Data Modeling:

Data storage in Mongo dB is flexible. The two ways of data storage are **normalized** and **embedded** structures. There are many advantages and disadvantages associated with these two forms of data storage that can be associated **with CAP** (Consistency, Availability, and Partition).

Normalized structure: This type of storing is advantageous when the need is reducing data duplication. But it's also associated with disadvantages of latency, increasing number of documents which is undesirable in large volume of data etc.

Embedded structure: This type of storing is flexible in terms of easy retrieval of data thus reducing latency. But it has its own disadvantages of integrity, duplication etc.

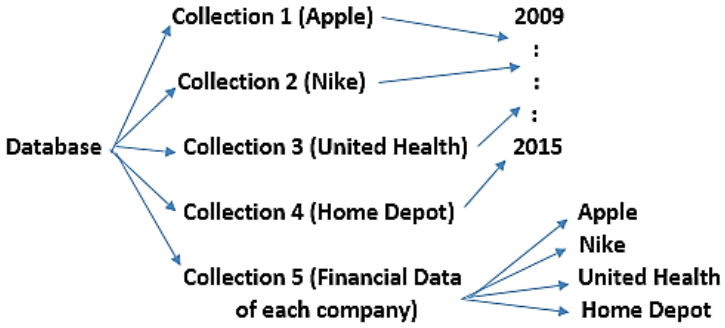
The data model that's used in our project is embedded modelling because of:

- ✓ Small volume of data
- ✓ Need for lesser latency rather availability
- ✓ No updating of data hence no integrity issues

Databases, collections, documents:

In view of the data model explained above we have one database with 5 collections four of which are the collection of 10-k SEC filing dump in Mongo DB. Each of the first four collections is assimilation of 10-k filings from 2009-2015 of each company. Extract of financial data is stored in fifth collection to ease the retrieval process for computations.

As MongoDB is atomic at document level, all the dependencies of the document i.e. all the required, dependent fields of the 10K filing are embedded in the same document.



Data structure in Mongo DB

The documents in each of the collections for complete 10-k filings are 7 per collection for 7 years. Each collection represents 1 year of SEC filing per company and fifth collection having 4 documents. Thus, we have

1 Database -> 5 Collections -> 32 Documents

```
> show dbs
SEC_filings_10k   0.453GB
local             0.078GB
>
```

```
> show collections
SEC_10k_financial_collec
apple_col
homedepot_col
nike_col
system.indexes
unitedhealth_col
```

The main advantage of this schema over a collection containing all the 10k data of all companies is prevention of the collection level lock which occurs during write operation disabling the read access to all other companies 10k files. This schema makes the database more scalable and can be developed widely in multi-tasking environment.

Constraints, Indexing and Document Validation in MongoDB:

A constraint on the year can be applied on the application side to eliminate the duplication of data for the same year in a particular collection.

Indexing:

Indexes support the efficient execution of queries in MongoDB. Without indexes MongoDB must perform a collection scan, i.e. scan every document in a collection, to select those

documents that match the query statement. If an appropriate index exists for a query, MongoDB can use the index to limit the number of documents it must inspect.

- The index stores the value of a specific field or set of fields, ordered by the value of the field. The ordering of the index entries supports efficient equality matches and range-based query operations. In addition, MongoDB can return sorted results by using the ordering in the index.
- Creating a unique index so that the collection will not accept insertion of documents where the index key or keys match an existing value in the index.
- **The reason we created a unique index** so that the collection will not accept insertion of documents where the index key or keys match an existing value in the index. We created single index on field – year and enforced uniqueness constraint. A single index consists of {year: -1}, the index sorts the documents by year field in descending order.

```
> db.apple_col.createIndex({"year":-1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

Document validation:

MongoDB provides the capability to **validate documents** during updates and insertions. Validation rules are specified on a per-collection basis using the validator option, which takes a document that specifies the validation rules or expressions. MongoDB also provides the validation Level option, which determines how strictly MongoDB applies validation rules to existing documents during an update, and the validation Action option, which determines whether MongoDB should error and reject documents that violate the validation rules or warn about the violations in the log but allow invalid documents.

By default, validation Action is error and MongoDB rejects any insertion or update that violates the validation criteria. When validation Action is set to warn, MongoDB logs any violations but allows the insertion or update to proceed.

```
>
> db.runCommand( {
...     collMod: "nike_col",
...     validator: { $or: [ { year: { $exists: true } },
...     { name: { $exists: true } } ] },
...     validationLevel: "strict", validationAction: "warn"
... } )
{ "ok" : 1 }
>
```

Format:

Mongo DB accepts **JSON file** by default and the structure of data storage in BSON/JSON is **key value pair**. Thus, for storing our data we chose to follow the same by storing all the data fields in dictionary and then uploading into Mongo DB.

B. Data Migration process:

(First the entire 10k report is into MongoDB is dumped and then the financial details section as separate documents)

Importing HTML file:

- Platform: Python
- Package used: OS

The 10-k html files of each company for e.g. Apple were read as input in the process using the OS package in python. A list of company names, identifiers, address are made (for simplicity we chose manual creation, they can be parsed directly). Using nested for loop, 10-k html files of each year are read automatically into python as:

Storing data:

- All the files of each and every year are stored into a dictionary with name of company, Emp_IRS id, year, address of the company along with files as:
- After reading each year's 10-k file of each company the resulting dictionary was appended to a list. Thus, there are 28 elements in the list each of which points to the 10-k html file of each year of a company. For simplicity, the appended list is split into company's files.

Saving data in Mongo dB:

- Using PyMongo connection is established to MongodB using MongoClient.
- Each company's all years data is dumped into a collection under the database which contains 7 documents (for example). Thus there are 4 collections with 7 documents in one database.

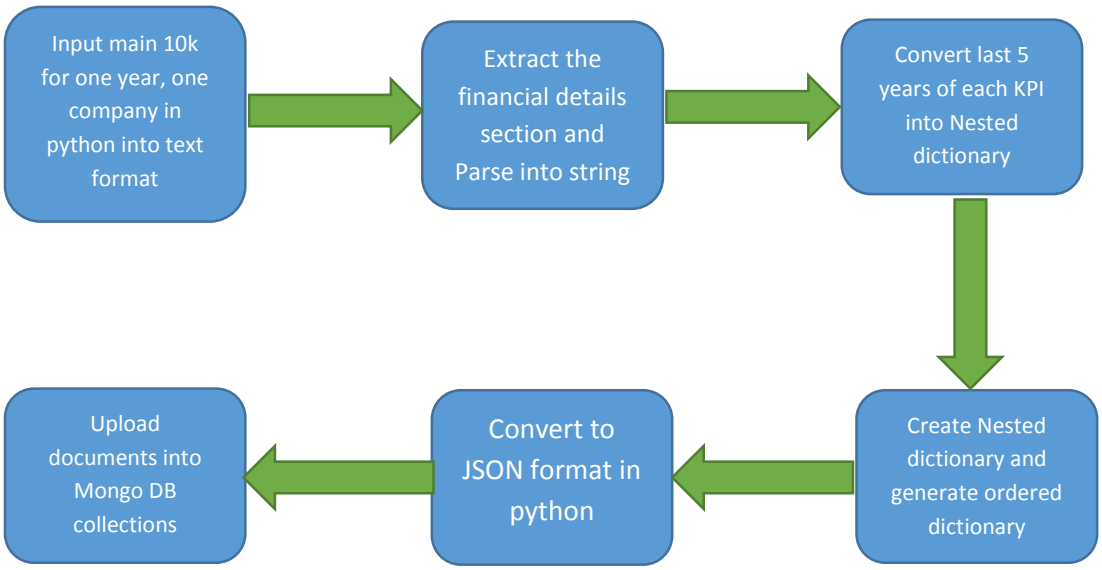
The data in the 10-k filings contain information about the financial details in the form of text and numerical data stored in tables. In short, the data from the html files is read and parsed to extract the text which implied description of a section and numerical data which detailed about the company's operation.

Snapshot for Counts of documents in each collection:

Snapshot for Actual Data in MongoDB:

```
In [13]: cur=db.apple_col.find({"year":2014})
    ...:
    ...: for document in cur:
    ...:     print(document)
    ...:
    ...:
{u'name': u'Apple Inc', u'address': u'1 Infinite Loop in Cupertino,
California', u'year': 2014, u'Emp_IRS_id': u'94-2404110', u'report':
u'<DOCUMENT>\r\n<TYPE>10-
K\r\n<SEQUENCE>1\r\n<FILENAME>d783162d10k.htm\r\n<DESCRIPTION>10-
K\r\n<TEXT>\r\n<HTML><HEAD>\r\n<TITLE>10-K</TITLE>\r\n</HEAD>\r\n <BODY
BGCOLOR="WHITE">\r\n<h5 align="left"><a href="#toc">Table of
Contents</a></h5>\r\n\r\n <P STYLE="line-height:1.0pt;margin-top:0pt;margin-
bottom:0pt;border-bottom:1px solid #000000"> </P>\r\n<P STYLE="line-
height:3.0pt;margin-top:0pt;margin-bottom:2pt;border-bottom:1px solid
#000000"> </P> <P STYLE="margin-top:0pt; margin-bottom:0pt; font-
size:13pt; font-family:Arial" ALIGN="center"><B>UNITED STATES </B></P>\r\n<P
STYLE="margin-top:0pt; margin-bottom:0pt; font-size:13pt; font-family:Arial"
ALIGN="center"><B>SECURITIES AND EXCHANGE COMMISSION </B></P>\r\n<P
STYLE="margin-top:0pt; margin-bottom:0pt; font-size:11pt; font-family:Arial"
ALIGN="center"><B>Washington, D.C. 20549 </B></P> <P STYLE="font-
size:3pt;margin-top:0pt;margin-bottom:0pt"> </P><center>\r\n<P
STYLE="line-height:6.0pt;margin-top:0pt;margin-bottom:2pt;border-bottom:1px
solid #000000;width:21%"> </P></center> <P STYLE="margin-top:3pt;
margin-bottom:0pt; font-size:17pt; font-family:Arial" ALIGN="center"><B>Form
<FONT\r\nSTYLE="white-space:nowrap">10-K</FONT> </B></P><center> <P
STYLE="line-height:6.0pt;margin-top:0pt;margin-bottom:2pt;border-bottom:1px
solid #000000;width:21%"> </P></center>\r\n<P STYLE="margin-top:3pt;
margin-bottom:0pt; font-size:8pt; font-family:Arial" ALIGN="justify">(Mark
One) </P>\r\n<TABLE STYLE="BORDER-COLLAPSE:COLLAPSE; font-family:Arial;
font-size:9pt" BORDER="0" CELLPADDING="0" CELLSPACING="0"
WIDTH="100%">\r\n<TR>\r\n<TD WIDTH="4%" VALIGN="top"
ALIGN="left"><B></B><FONT STYLE="FONT-
```

Flow chart depicting the process of migration from HTML to uploading in Mongo dB



The next section gives detailed explanation of each block and how the data is migrated in Mongo DB:

Data Parsing:

- Package used: Regular expression (re)
- Platform Python

Once the data file is opened in read mode (rb), information either text or numerical data is parsed by using **Index** methodology. In this, the section name or table name or text can be given as input and that needs to be found from the file. As per the requirement extraction can be limited using the indexing operator.

Garbage treatment:

When data is parsed the result contains lot of garbage along with the necessary information. Using regular expression (re) all the unwanted texts are replaced with white space. For demo purpose, the following section illustrates step by step process in extracting numerical data of certain parameters from a table.

As the 10-k filings contains cumulative data from previous years, parsed numbers are in the same order in descending order of years. Thus, values are stored in one list for each parameter rather nested or multiple lists.

Dictionary creation:

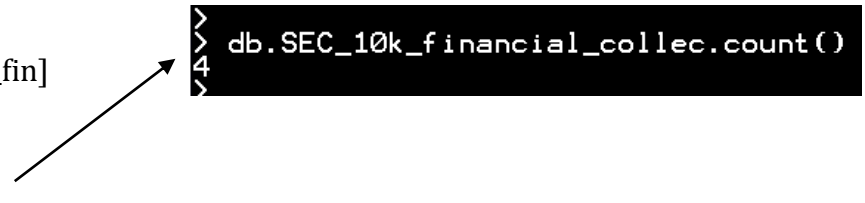
Each list is converted into a dictionary and then a single ordered dictionary is created by merging the dictionaries.

Data Storing in Mongo DB:

- Package: PyMongo, MongoClient
- Platform: Python

Each nested dictionary is uploaded into Mongo dB by establishing connection via MongoClient and storing the data in a collection in which each document is sample financial data of each company.

```
# Mongo DB Data upload, using Python
from pymongo import MongoClient
cl = MongoClient()
db = cl.SEC_filings_10k
all=[apple_fin,nike_fin,hd_fin,united_fin]
c5=db.SEC_10k_financial_collec
res=c5.insert_many(all)
db.SEC_10k_financial_collec.count()
```



Snapshot of Financial details collections consist of 4 documents as shown below:

Nike


```
{'Current Ratio': {'2011': 2.9,
  '2012': 3.0,
  '2013': 3.5,
  '2014': 2.7,
  '2015': 2.5},
 'Gross Profit': {'2011': 9202,
  '2012': 10148,
  '2013': 11034,
  '2014': 12446,
  '2015': 14067},
 'Gross margin %': {'2011': 45.7,
  '2012': 43.5,
  '2013': 43.6,
  '2014': 44.8,
  '2015': 46.0},
 'Price/Earnings ratio': {'2011': 19.3,
  '2012': 23.0,
  '2013': 22.8,
  '2014': 25.9,
  '2015': 27.5},
 'Return on Assets %': {'2011': 15.0,
  '2012': 15.1,
  '2013': 15.3,
  '2014': 14.9,
  '2015': 16.3},
 'Revenues': {'2011': 20117,
  '2012': 23331,
  '2013': 25313,
  '2014': 27799,
  '2015': 30601},
 '_id': ObjectId('56af0c18b877da19406ace0c'),
 'name': 'Nike'}
```


Apple


```
{'EPS': {'2010': 2.2, '2011': 4.01, '2012':
6.38, '2013': 5.72, '2014': 6.49},
 'Net earnings': {'2010': 14013,
  '2011': 25922,
  '2012': 41733,
  '2013': 37037,
  '2014': 39510},
 'Revenues': {'2010': 65225,
  '2011': 108249,
  '2012': 156508,
  '2013': 170910,
  '2014': 182795},
 'Total Assets': {'2010': 75183,
  '2011': 116371,
  '2012': 176064,
  '2013': 207000,
  '2014': 231839},
 'Total liabilities': {'2010': 27392,
  '2011': 39756,
  '2012': 57854,
  '2013': 83451,
  '2014': 120292},
 '_id': ObjectId('56af1156b877da19406ace0f'),
 'name': 'Apple'}
```


UnitedHealth Group


```
{'Basic EPS': {'2010': 4.14,
  '2011': 4.81,
  '2012': 5.38,
  '2013': 5.59,
  '2014': 5.78},
 'Debt to debt-plus-equity ratio': {'2010': 30
  '2011': 29.1,
  '2012': 35.0,
  '2013': 34.4,
  '2014': 34.9},
 'Earnings from operations': {'2010': 7864,
  '2011': 8464,
  '2012': 9254,
  '2013': 9623,
  '2014': 10274},
 'Return on equity %': {'2010': 18.7,
  '2011': 18.9,
  '2012': 18.7,
  '2013': 17.7,
  '2014': 17.3},
 'Revenues': {'2010': 94155,
  '2011': 101862,
  '2012': 110618,
  '2013': 122489,
  '2014': 130474},
 'Total assets': {'2010': 63063,
  '2011': 67889,
  '2012': 80885,
  '2013': 81882,
  '2014': 86382},
 '_id': ObjectId('56af0c18b877da19406ace0e'),
 'name': 'United Health Group'}
```


HomeDepot


```
{'Current Ratio': {'2010': '1.33:1',
  '2011': '1.55:1',
  '2012': '1.34:1',
  '2013': '1.42:1',
  '2014': '1.36:1'},
 'EPS increase (%)': {'2010': 29.7,
  '2011': 22.9,
  '2012': 21.5,
  '2013': 25.3,
  '2014': 25.3},
 'Net earnings increase (%)': {'2010': 27.4,
  '2011': 16.3,
  '2012': 16.8,
  '2013': 18.7,
  '2014': 17.8},
 'Return on invested capital %': {'2010': 12.8,
  '2011': 14.9,
  '2012': 17.0,
  '2013': 20.9,
  '2014': 24.9},
 'Revenues': {'2010': 67997,
  '2011': 70395,
  '2012': 74754,
  '2013': 78812,
  '2014': 83176},
 'Total debt-to-equity (%)': {'2010': 51.6,
  '2011': 60.3,
  '2012': 60.7,
  '2013': 117.6,
  '2014': 184.5},
 '_id': ObjectId('56af0c18b877da19406ace0d'),
 'name': 'HomeDepot'}
```

C. Company health:

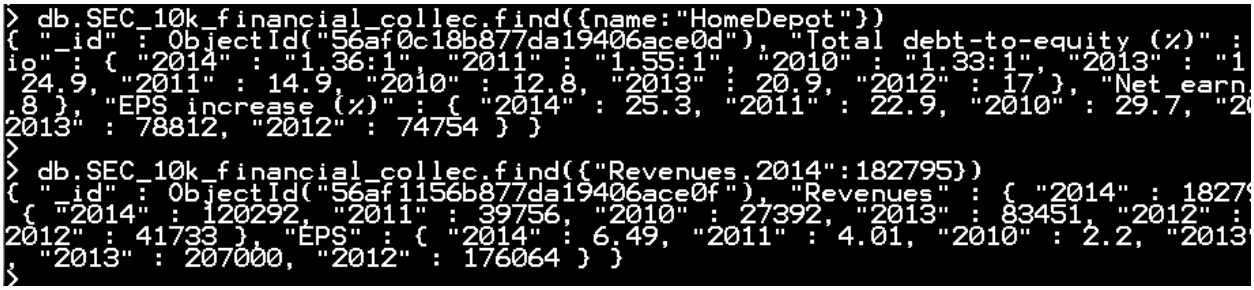
Rationale:

We looked over the 10K reports in great detail to fully understand all the financial declarations. It was quite comprehensive and full of information. We also noticed that while 10K reports followed a certain reporting format, it was clear that each company was allowed to report financial information in the manner that it determined was the best for them. There was consistency across each individual filing.

We broke apart the reports and used our best judgment to capture and assess what we felt was a good indicator of company health. First, we looked at revenues and year over year growth. Concurrently, we looked at gross margins and net margins to verify growth. Then we looked at basic ratios like earnings per share to make an assessment of what investors were expecting the stock to do in terms of future growth and the current ratios to measure liquidity.

Queries & Summary:

Company's health can be judged based on various parameters in various situations. Below are the inferences and the description of the parameters:



 # Connect to Mongo DB

```
from pymongo import MongoClient
cl = MongoClient()
db = cl.SEC_filings_10k
```

 # MongoDB queries – DB is SEC_filings_10k, collection is SEC_10k_financial_collec.

```
hd_fin=db.SEC_10k_financial_collec.find({name:u'HomeDepot'})
app_fin=db.SEC_10k_financial_collec.find({name:u'Apple'})
nike_fin=db.SEC_10k_financial_collec.find({name:u'Nike'})
utd_fin=db.SEC_10k_financial_collec.find({name:u'United Health Group'})
```

 # Import necessary packages

```
import pandas as pd
from pandas.io.json import json_normalize
```

```
from bson import json_util, ObjectId
import json
import glob
from pymongo import MongoClient

# Funtion to convert json data into pandas dataframe.

def mongo_to_dataframe(mongo_data):
    sanitized = json.loads(json_util.dumps(mongo_data))
    normalized = json_normalize(sanitized)
    df = pd.DataFrame(normalized)
    return df

# call the function and convert to dataframe
a1=mongo_to_dataframe(apple_fin)
h1=mongo_to_dataframe(hd_fin)
n1=mongo_to_dataframe(nike_fin)
u1=mongo_to_dataframe(united_fin)

# Export dataframe to csv for all companies

a1.to_csv("apple.csv")
h1.to_csv("homedepot.csv")
n1.to_csv("nike.csv")
u1.to_csv("united.csv")

# Once, we got data into csv, we did calculations in excel
```

Summary of Findings:

❖ Home Depot (2010-2014)

- Revenues increased from $67,997M in 2010 to $83,176M in 2014 with a year over year increase of 0.035 in 2011, 0.062 in 2012, 0.054 in 2013, and 0.055 in 2014. This is a clear indication that the company was growing.
- The company chose to declare the earnings based on net year over year percentages with the highest increase of 27.4% in 2010 and then lesser growth in 16.3% in 2011, 16.8% in 2012, 18.7% in 2013, and 17.8% in 2014. This shows overall positive growth.
- They chose to focus on earnings per share increase percentage of 29.7% in 2010, 22.9% in 2011, 21.5% in 2012, 25.3% in 2013, and 25.3% in 2014. These numbers area also positive.
- The current ratio was presented by the company at 1.33 in 2010, 1.55 in 2011, 1.34 in 2012, 1.42 in 2013, and 1.36 in 2014. Since the ratio was greater than 1, it is considered a positive sign of the company's liquidity and its ability to pay off debt.

❖ **United Health Group (2010-2014)**

- Revenues increased from $94,155M in 2010 to $130,474M in 2014 with a year over year increase of 0.082 in 2011, 0.806 in 2012, 0.107 in 2013, and 0.065 in 2014. This is a clear indication that the company was growing.
- The company chose to declare the earnings from operations and they increased from $7,864M to $10,274M. We calculated the year over year percentages as 0.076 in 2011, 0.093 in 2012, 0.040 in 2013, and 0.068 2014. This shows overall positive growth as it relates to earnings since they also grew with the revenues.
- The basic earnings per share was presented by the company at $4.14 in 2010, $4.81 in 2011, $5.38 in 2012, $5.59 in 2013, and $5.78 in 2014 with an average of $5.14. Companies with a complicated capital structure must report both basic EPS and diluted EPS. Basic EPS is always higher of the two with net profit divided by outstanding shares.
- Total assets also showed a year over year increase from $$63,063M to $86,382 with increase percentages calculated as 0.077 in 2011, 0.191 in 2012, 0.012 in 2013, and 0.055 in 2014. This is a positive sign of company strength.

❖ **Nike (2011-2015)**

- Revenues increased from $20,117M in 2011 to $30,601M in 2015 with a year over year increase of 0.160 in 2012, 0.098 in 2013, 0.085 in 2014, and 0.160 in 2015. This is a clear indication that the company was growing.
- The gross profit increased from $9,202M in 2011 to $14,067M in 2015. This indicates a positive growth, however, since the revenues also increased, we looked at the gross margin percentage at 45.7% in 2011, 43.5% in 2012, 43.6% in 2013, 44.8% in 2014, and 46% in 2015. This is consistent with a benchmark of 46% for retail business.
- The current ratio presented by the company was 2.9 in 2011, 3.0 in 2012, 3.5 in 2013, 2.7 in 2014, and 2.5 in 2015. Since the ratio was in the 2.5-3.5 range, it is considered a very positive sign of the company's liquidity and its ability to payoff short-term debt.
- The price per earnings ratio was 19.3 in 2011, 23.0 in 2012, 22.8 in 2013, 25.9 in 2014, and 27.5 in 2015. These numbers area also a positive sign of the growth in stock price and the increase in premium that investors are willing to pay for expectation of future growth of the company.

❖ **Apple (2010-2014)**

- Revenues increased from $62,225M in 2010 to $182,795M in 2014 with a year over year increase of 0.660, in 2011, 0.446 in 2012, 0.092 in 2013, and 0.070 in 2014. Revenues nearly tripled over this period. This is a clear indication that the company was experiencing tremendous growth.
- Net earnings increased from $14,013M in 2010 to $39,510M in 2014 with a year over year increase of 0.850 in 2011, 0.610 in 2012, -0.113 in 2013, and 0.067 in 2014. While this may not look significant, looking at other information tells us the revenues and assets nearly tripled over this time.

- Total assets also showed a year over year increase from $75,183M to $231,839 with increase percentages calculated as 0.548 in 2011, 0.513 in 2012, 0.176 in 2013, and 0.120 in 2014. This is a great sign of company strength.
- The price per earnings ratio was 2.2 in 2010, 4.01 in 2011, 6.38 in 2012, 5.72 in 2013, and 6.49 in 2014. These numbers area also a positive sign of the growth in stock price.

D. Acquisitions:

Rationale:

We scanned through the 10K documents. We then went over the definitions and criteria of acquisition reporting.

By SEC rule, we assumed that all acquisitions indeed get disclosed in some manner on financial statements since the purchase price has to flow through them. We also assumed that it would be exciting for a company to report shareholders and potential investors this activity as a positive sign of research, proposal development, and the eventual assimilation of another company, giving it a competitive edge. We additionally assumed that supporting activities that cost significant man-hours and operating expenses running in millions would be clearly reported. Sometimes, these transactions could run in billions, as in case of the lofty and well-documented $3B purchase of Beats by Apple in 2014. Our second assumption was highly inaccurate. For example, we found that Apple actually likes to hides these transactions for as long as possible. This was evidenced by a cross check of Apple's Acquisitions on Wikipedia where we found missing information.

In terms of identifying a single acquisition, it can easily be picked up from the statement of cash flows if the transition is in cash, or shareholder equity if it was purchased with company stock. However, sometimes it is a combination of the two. If there are multiple acquisitions in the reporting period, it was much more difficult to draw inferences this way. We also concluded that the information was reported if it was 'material.' Accounting rules define 'materiality' in an intentionally broad way. From the Financial Accounting Standards Board's Statement of Concepts #2: "The omission or misstatement of an item in a financial report is ***material*** if, in the light of surrounding circumstances, the magnitude of the item is such that it is probable that the judgment of a reasonable person relying upon the report would have been changed or influenced by the inclusion or correction of the item."

So, who decides what to report and how to report it? A company's auditors are charged with the responsibility of setting a materiality threshold based on revenues, assets, and net income. This grey area gives large companies the ability to make small acquisitions without disclosing as true information.

United Healthcare (2014)

Found within the statement of cash flows called by United Healthcare as:

Summary of our Major Sources and Uses of Cash and Cash Equivalents

Uses of cash:
Cash paid for acquisitions and non-controlling interest shares, net of cash assumed ($1,923M)

Home Depot (2014)

Found within the statement of cash flows called by Home Depot as:

Consolidated Statement of Cash Flows

Cash Flows from Investing Activities:

Payments for Businesses Acquired, net ($206M)

In January 2015, we acquired HD Supply Hardware Solutions, known as Crown Bolt, a leading supplier of fasteners and builders hardware to retailers in the U.S. We expect this acquisition to further enhance our supply chain capabilities and product offerings in hardware. (Page 20)

Nike (2010 and 2014)

Found within the statement of cash flows from 2010 called by Nike as:

Consolidated Statement of Cash Flows

Cash used by investing activities:

Acquisition of subsidiary, net of cash acquired (Note 4) ($571.1M - 2008)

Found within the statement of cash flows from 2014 called by Nike as:

NIKE, Inc. Consolidated Statements of Cash Flows

Cash (used) provided by investing activities:

- Purchases of short-term investments
- Maturities of short-term investments
- Sales of short-term investments
- Additions to property, plant and equipment
- Disposals of property, plant and equipment
- Proceeds from divestitures
- Increase in other assets, net of other liabilities ($2M)

Apple (2015)

Found within the statement of cash flows called by Apple as:
Consolidated Statement of Cash Flows

Investing Activities:
Payments made in connection with business acquisitions, net ($3,765M)
Apple (Beats Purchase reported 2015)

Recent Sales of Unregistered Securities

On July 31, 2014, in connection with its acquisitions of Beats Music, LLC and Beats Electronics, LLC (collectively "Beats"), the Company issued approximately 5.1 million shares of its common stock to certain former equity holders of Beats in reliance on the exemption from registration pursuant to Section 4(a)(2) of the Securities Act of 1933, as amended. The majority of these shares will vest over time based on continued employment with Apple.

Note 4 – Goodwill and Other Intangible Assets

On July 31, 2014, the Company completed the acquisitions of Beats Music, LLC, which offers a subscription streaming music service, and Beats Electronics, LLC, which makes Beats® headphones, speakers and audio software (collectively, "Beats"). The total purchase price consideration for these acquisitions was $2.6 billion, which consisted primarily of cash, of which $2.2 billion was allocated to goodwill, $636 million to acquired intangible assets and $258 million to net liabilities assumed. Concurrent with the close of the acquisition, the Company repaid $295 million of existing Beats outstanding debt to third-party creditors. In conjunction with the Beats acquisitions, the Company issued approximately 5.1 million shares of its common stock to certain former equity holders of Beats. The restricted stock was valued at approximately $485 million based on the Company's common stock on the acquisition date. The majority of these shares, valued at approximately $417 million, will vest over time based on continued employment with Apple.

The Company also completed various other business acquisitions during 2014 for an aggregate cash consideration, net of cash acquired, of $957 million, of which $828 million was allocated to goodwill, $257 million to acquired intangible assets and $128 million to net liabilities assumed.

The Company completed various business acquisitions during 2013 for an aggregate cash consideration, net of cash acquired, of $496 million, of which $419 million was allocated to goodwill, $179 million to acquired intangible assets and $102 million to net liabilities assumed. The Company's gross carrying amount of goodwill was $4.6 billion and $1.6 billion as of September 27, 2014 and September 28, 2013, respectively. The Company did not have any goodwill impairments during 2014, 2013 or 2012.

How would you automate this process?

There is no standard or one-fixed way one can automate the above process. We can automate it to a by using Tree-walking approach and Tags-based querying, Natural language processing and Text Mining techniques. Here are some more details:

Read the 10k html file as text in a tool like Python. Using the buzzwords like "acquired, acquisitions, cash paid in acquiring," etc. identify and navigate to the section where these information is present. Using contextual filtering, there is a mechanism to make sure that we hit the right section. Filter out the remaining text. Also, refer to consolidated Cash flow statements, Balance Sheets which contains the necessary data. Next, we can use either of the two approaches:

1. **Tree-walking approach and Tags-based querying**:

   Use package like BeautifulSoup to parse the html and convert it into bs4 object. Next, use the tags-based querying and identify the table which has the relevant data. Use a loop and pick the "tr", "td" tags to extract the data present in the table. Use package like re (regular expression) to remove the unwanted tags, stop words and other non-info words. Convert the remaining data into an ordered dictionary using key-value pair. At some instances, this method throws garbage data inside the relevant info. Often, manual text mining techniques might be needed to arrive at accurate information.

2. **Natural language processing and Text Mining techniques:**

   Once the data is in text format, we can use indexing in a forward direction to pick out the required sections. Text mining in the form of string based find and index are extremely accurate, but require lengthy coding as some words need to be hard-coded. It involves complex logical coding and use of multiple nested loops which become time consuming. Once, the relevant text is extracted, we can use regular expressions to remove the unwanted tags, stop words and other non-info words. Convert the remaining data into a list and then into an ordered dictionary using key-value pair.

What were the difficulties you encountered (which might result in inaccurate derivations of the values)?

- There is no dedicated section for mergers and acquisitions. There are multiple mentions of words acquisitions which made it difficult to determine the accurate values of acquisitions.
- Moreover, 10k reports are not standardized in the way that different companies have mentioned it in separate sections. Certain companies have this information in Cash flow statements and balance sheets; whereas some have it in a table which contains information other than acquisition also and some have it as free flowing text.
- Any of 10k reports does not contain the number of companies acquired or merged with in a very definitive manner, except free flowing text.
- At some instances, lack of in-depth financial knowledge made us difficult for us to derive the right values of acquisitions.