Data Bootcamp Final Project: Drivers Behind the Performance of Retailers

Crystal Falcon and Shawn An

crystal.falcon@stern.nyu.edu / shawn.an@stern.nyu.edu

Abstract

For our project, we were interested in analyzing the ongoing transformation of the clothing retail sector as our economy becomes increasingly digital. As the threat of e-commerce and online shopping looms larger for brick-and-mortar retailers, we sought to isolate the impact of e-commerce activity on the performance of three different types of retailers: department stores, discount retailers, and brand holding companies. We limited our sample size to forty publicly-traded retailers in the U.S. and collected data capturing performance and performance drivers for each of these firms in the past five years.

Data

For the forty firms in our sample, we observed two performance metrics and three performance drivers during the period between fiscal year 2013 and fiscal year 2017. Our sample includes twelve department stores, five discount retailers, and twenty-three brand holding companies of varying size and market positioning. We chose enterprise value-to-sales multiples and net income margins as our performance metrics, and SG&A margin, e-commerce sales margin, and total square footage as performance drivers. The data was sourced primarily through three sources: Bloomberg, another market research platform called eMarketer, and SEC-filed 10-K's.

In regards to performance metrics, we used EV-to-sales multiples in order to capture public market sentiment, which would be valuable in determining whether our performance drivers have substantial impact on the value investors attribute to specific retailers. Additionally, we wanted to use a revenue multiple given its importance in the retail space. Our other performance metric, net income margin would be a direct representation of each firms' profitability, allowing us to analyze which of our performance drivers have the largest impact on retailers' bottom lines.

With regards to performance drivers, e-commerce as a percentage of revenue is intended to capture how readily consumers adopt a retailers' e-commerce platform if they operate one. We included two additional drivers to assess the impact of e-commerce sales relative to other factors. We thought SG&A margin would be a significant indicator of the degree of investment firms place on sales, advertising, and other marketing tactics. Finally, total square footage tells us whether management is deciding to expand or contract the physical retail space operated by their firm, demonstrating how efficiently or effectively retailers are utilizing their brick-and-mortar stores.

In [1]:
```
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pylab as pl
import pandas as pd
import numpy as np
import pandas_datareader.data as web
import statsmodels.api as sm
import statsmodels.formula.api as smf

%matplotlib inline
```

```
/Users/Crystal/anaconda/lib/python3.6/site-packages/statsmodels/compat/
pandas.py:56: FutureWarning: The pandas.core.datetools module is deprec
ated and will be removed in a future version. Please use the pandas.tse
ries module instead.
  from pandas.core import datetools
```

In [2]:

```
#Importing Data
url_nim = "https://raw.githubusercontent.com/shawnawn/Data_Bootcamp_Fina
l_Project/master/netincomemargin.V1.csv"
net_income_margin = pd.read_csv(url_nim)

evsales_url = "https://raw.githubusercontent.com/shawnawn/Data_Bootcamp_
Final_Project/master/EVsales1.csv"
evsales = pd.read_csv(evsales_url)

url_sga = "https://raw.githubusercontent.com/shawnawn/Data_Bootcamp_Fina
l_Project/master/SGAmargin.V1.csv"
sga_margin = pd.read_csv(url_sga)

url_ec = "https://raw.githubusercontent.com/shawnawn/Data_Bootcamp_Final
_Project/master/ecommercemargin.V1.csv"
ec_margin = pd.read_csv(url_ec)

url_sf = "https://raw.githubusercontent.com/shawnawn/Data_Bootcamp_Final
_Project/master/squarefootage.V1.csv"
square_footage = pd.read_csv(url_sf)

net_income_margin.head()
```

Out[2]:

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015	2016
0	Department Stores	Kohl's Corp	KSS	Net Income Margin	5.11%	4.67%	4.56%	3.50%	2.98%
1	Department Stores	Macy's Inc	M	Net Income Margin	4.82%	5.32%	5.43%	3.96%	2.40%
2	Department Stores	Nordstrom Inc	JWN	Net Income Margin	6.06%	5.85%	5.33%	4.16%	2.40%
3	Department Stores	Dillard's	DDS	Net Income Margin	4.98%	4.84%	4.89%	3.99%	2.64%
4	Department Stores	JC Penney Co Inc	JCP	Net Income Margin	-7.59%	-10.78%	-5.85%	-4.06%	0.01%

In [3]:
```
net_income_margin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 9 columns):
Category        40 non-null object
Company Name    40 non-null object
Ticker          40 non-null object
Measure         40 non-null object
2012            40 non-null object
2013            40 non-null object
2014            40 non-null object
2015            40 non-null object
2016            40 non-null object
dtypes: object(9)
memory usage: 2.9+ KB
```

Pandas recognizes our percentages as strings when we want them to be floating numbers. We will first clean up our data so we can convert it.

In [4]:
```
net_income_margin = net_income_margin.replace('%','', regex=True) #getti
ng rid of %
net_income_margin.head()
```

Out[4]:

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015	2016
0	Department Stores	Kohl's Corp	KSS	Net Income Margin	5.11	4.67	4.56	3.50	2.98
1	Department Stores	Macy's Inc	M	Net Income Margin	4.82	5.32	5.43	3.96	2.40
2	Department Stores	Nordstrom Inc	JWN	Net Income Margin	6.06	5.85	5.33	4.16	2.40
3	Department Stores	Dillard's	DDS	Net Income Margin	4.98	4.84	4.89	3.99	2.64
4	Department Stores	JC Penney Co Inc	JCP	Net Income Margin	-7.59	-10.78	-5.85	-4.06	0.01

Now that it's a string of numbers, lets convert it.

In [5]:
```
net_income_margin = net_income_margin.apply(pd.to_numeric, errors='ignor
e')
net_income_margin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 9 columns):
Category        40 non-null object
Company Name    40 non-null object
Ticker          40 non-null object
Measure         40 non-null object
2012            40 non-null float64
2013            40 non-null float64
2014            40 non-null float64
2015            40 non-null float64
2016            40 non-null float64
dtypes: float64(5), object(4)
memory usage: 2.9+ KB
```

Great, now we can manipulate them as numbers. Let's do the same for our other dataframes.

In [6]:
```
sga_margin = sga_margin.replace('%','', regex=True)

ec_margin = ec_margin.replace('%','', regex=True)

square_footage = square_footage.replace('Holidng','Holding', regex=True)
square_footage = square_footage.replace('-','0', regex=True)
```

In [7]:
```
evsales = evsales.apply(pd.to_numeric, errors='ignore')

sga_margin = sga_margin.apply(pd.to_numeric, errors='ignore')

ec_margin = ec_margin.apply(pd.to_numeric, errors='ignore')

square_footage = square_footage.apply(pd.to_numeric, errors='ignore')
```

We want to be able to do analysis on the different sectors within retail, rather than each individual company. We will use a groupby function to get the mean data for each of the three sectors.

In [8]:
```
evsales_groups = evsales.groupby('Category').mean()
nim_groups = net_income_margin.groupby('Category').mean()
sga_groups = sga_margin.groupby('Category').mean()
ec_groups = ec_margin.groupby('Category').mean()
sf_groups = square_footage.groupby('Category').mean()

sga_groups.head()
```

Out[8]:

	2012	2013	2014	2015	2016
Category					
Brand Holding Companies	30.716087	30.954783	31.576957	31.742174	32.473913
Department Stores	25.630833	25.477500	25.227500	25.313333	25.995833
Discount Retail	24.774000	24.974000	24.628000	24.534000	25.010000

In [9]:
```
evsales_groups.head()
```

Out[9]:

	4/30/12	7/31/12	10/31/12	1/31/13	4/30/13	7/31/13	10/31/13	1/31/
Category								
Brand Holding Companies	2.028261	1.905217	1.906957	1.881739	1.769565	1.866522	1.903043	1.899
Department Stores	0.683333	0.659167	0.707500	0.687500	0.663333	0.729167	0.738333	0.785
Discount Retail	0.722500	0.795000	0.835000	0.757500	0.792500	0.890000	0.962000	1.024

Our EV/Sales dataframe is tricky because we have quarterly data. Because we have data for the years 2012-2016, we gathered data starting form the fiscal year ended Jan 2017 and worked back 5 years. We will have to consolidated this data to years to be consistent with the rest of our data.

In [10]:
```
# set the dates as the index so we can resample and graph
evsales_groups = evsales_groups.T
evsales_groups.index
```

Out[10]:
```
Index(['4/30/12', '7/31/12', '10/31/12', '1/31/13', '4/30/13', '7/31/1
3',
       '10/31/13', '1/31/14', '4/30/14', '7/31/14', '10/31/14', '1/31/1
5',
       '4/30/15', '7/31/15', '10/31/15', '1/31/16', '4/30/16', '7/31/1
6',
       '10/31/16', '1/31/17'],
      dtype='object')
```

In [11]:
```
# Changing the dtype from String to datetime
evsales_groups = evsales_groups.set_index(pd.DatetimeIndex(evsales_group
s.index))
evsales_groups.index
```

Out[11]:
```
DatetimeIndex(['2012-04-30', '2012-07-31', '2012-10-31', '2013-01-31',
               '2013-04-30', '2013-07-31', '2013-10-31', '2014-01-31',
               '2014-04-30', '2014-07-31', '2014-10-31', '2015-01-31',
               '2015-04-30', '2015-07-31', '2015-10-31', '2016-01-31',
               '2016-04-30', '2016-07-31', '2016-10-31', '2017-01-31'],
              dtype='datetime64[ns]', freq=None)
```

Now we can start acting like the index is a time function rather than a string. Unfortunately it does not recognize that we have quarterly data.

In [12]:
```
evsales_groups_quarter = evsales_groups.resample("Q-OCT").mean()
```

In [13]:
```
#taking the rolling average for the past 4 quarters
evsales_groups_quarter = evsales_groups_quarter.rolling(window=4).mean()
evsales_groups_quarter
```

Out[13]:

Category	Brand Holding Companies	Department Stores	Discount Retail
2012-04-30	NaN	NaN	NaN
2012-07-31	NaN	NaN	NaN
2012-10-31	NaN	NaN	NaN
2013-01-31	1.930543	0.684375	0.777500
2013-04-30	1.865870	0.679375	0.795000
2013-07-31	1.856196	0.696875	0.818750
2013-10-31	1.855217	0.704583	0.850500
2014-01-31	1.859565	0.728958	0.917125
2014-04-30	1.867391	0.751250	0.958000
2014-07-31	1.832065	0.753542	0.965000
2014-10-31	1.793696	0.754375	0.975500
2015-01-31	1.759674	0.754583	1.007500
2015-04-30	1.752500	0.763125	1.060500
2015-07-31	1.764022	0.769167	1.108000
2015-10-31	1.744239	0.755625	1.118500
2016-01-31	1.687283	0.711250	1.078500
2016-04-30	1.611848	0.662500	1.044500
2016-07-31	1.525109	0.606250	1.027500
2016-10-31	1.461304	0.576875	1.043000
2017-01-31	1.407174	0.572917	1.064000

Now we have the rolling 4 quarter averages, meaning that the yearly data we are looking for is found on each Jan 31st row. We use resampling().last to isolate this.

In [14]:
```
#aggregating so we can get the average EV/Sales ratio in each year ended
 Jan 31
evsales_groups_yearly = evsales_groups_quarter.resample("12M", closed="l
eft",loffset="-3M").last()
evsales_groups_yearly.head()
```

Out[14]:

Category	Brand Holding Companies	Department Stores	Discount Retail
2013-01-31	1.930543	0.684375	0.777500
2014-01-31	1.859565	0.728958	0.917125
2015-01-31	1.759674	0.754583	1.007500
2016-01-31	1.687283	0.711250	1.078500
2017-01-31	1.407174	0.572917	1.064000

Although it says the years are 2013-2017, retailers' fiscal years ending in January means that it reflects data of the past calendar year. In other words, the quarterly rolling average on January 31st 2017 is mostly informed by things that happened in the 2016 calendar year. We will rename the index labels so that it is consistent with the rest of our data.

In [15]:
```
# Convert it back to a string so it's easier to rename
evsales_groups_yearly.index = evsales_groups_yearly.index.astype(str, co
py=True)

evsales_groups_yearly = evsales_groups_yearly.rename(index={'2013-01-31'
: '2012',
                                                              '2014-01-3
1':'2013',
                                                              '2015-01-3
1':'2014',
                                                              '2016-01-3
1':'2015',
                                                              '2017-01-3
1':'2016'})
evsales_groups_yearly
```

Out[15]:

Category	Brand Holding Companies	Department Stores	Discount Retail
2012	1.930543	0.684375	0.777500
2013	1.859565	0.728958	0.917125
2014	1.759674	0.754583	1.007500
2015	1.687283	0.711250	1.078500
2016	1.407174	0.572917	1.064000

Now that we have our data all cleaned up, we can now move on to graph them to try to gain some more insight on the retail sector.

Data Visualization

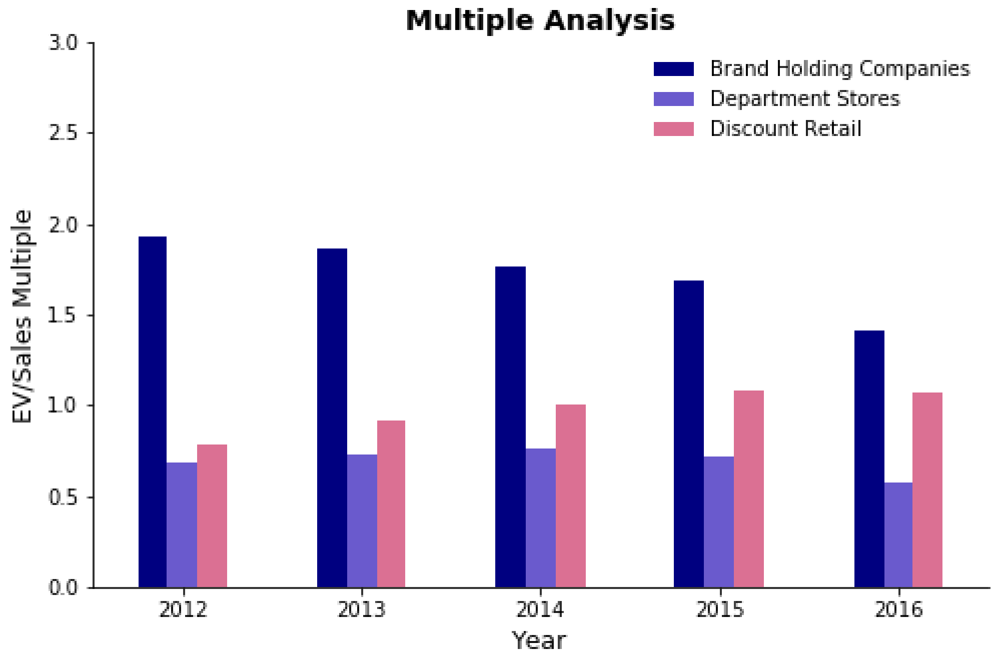
We created a series grouped bar graphs to visualize the change in performance metrics and drivers over the five year period of observation. Because forty observations for each year was too chaotic visually, we grouped the firms by their categories (department store, discount retail, and brand holding company) and plotted three bars for each year. Our analysis produced several noteworthy findings.

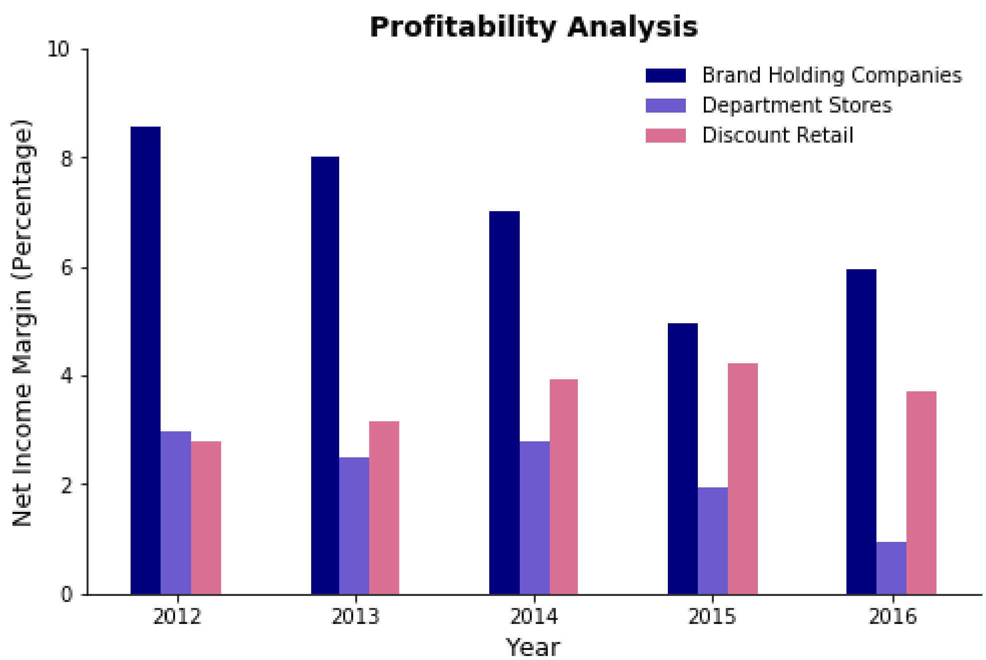
Performance metrics

In [16]:

```
fig, ax = plt.subplots()        # create axis object axe
evsales_groups_yearly.plot(ax=ax, #graphing EV/Sales by sector over time
                kind='bar',      #we decided that bar graphs was the bes
t way of displaying
                                 #data since we have three distinct cate
gories
                 color=['navy','slateblue','palevioletred'],
                 rot = 0,        #make year labels horizontal
                 figsize = (8,5))  #make graph bigger
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.legend(frameon = False)          #get rid of the box around the legen
d
ax.set_title('Multiple Analysis', #setting title
             fontsize = 14,
             color = 'black',
            fontweight = 'bold')
ax.set_ylabel('EV/Sales Multiple', #setting y axis
              fontsize = 12.5,
              color = 'black')
ax.set_ylim(0,3)
ax.set_xlabel('Year',
              fontsize = 12.5,
              color = 'black')

fig, ax = plt.subplots()        # create axis object axe
nim_groups.T.plot(ax=ax,        #graphing net income margin by sector, t
ransforming it so that
                                # year is on the x axis
                kind='bar',
                 color=['navy','slateblue','palevioletred'],
                 rot = 0,            #want year labels to be horizontal
                 figsize = (8,5))      #change size
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.legend(frameon = False)            #get rid of box around legend
ax.set_title('Profitability Analysis',
             fontsize = 14,
             color = 'black',
            fontweight = 'bold')
ax.set_ylabel('Net Income Margin (Percentage)',
              fontsize = 12.5,
              color = 'black')
ax.set_ylim(0,10)
ax.set_xlabel('Year',
              fontsize = 12.5,
              color = 'black')
```

```
Out[16]: <matplotlib.text.Text at 0x10ebdce48>
```





Performance Metrics Analysis

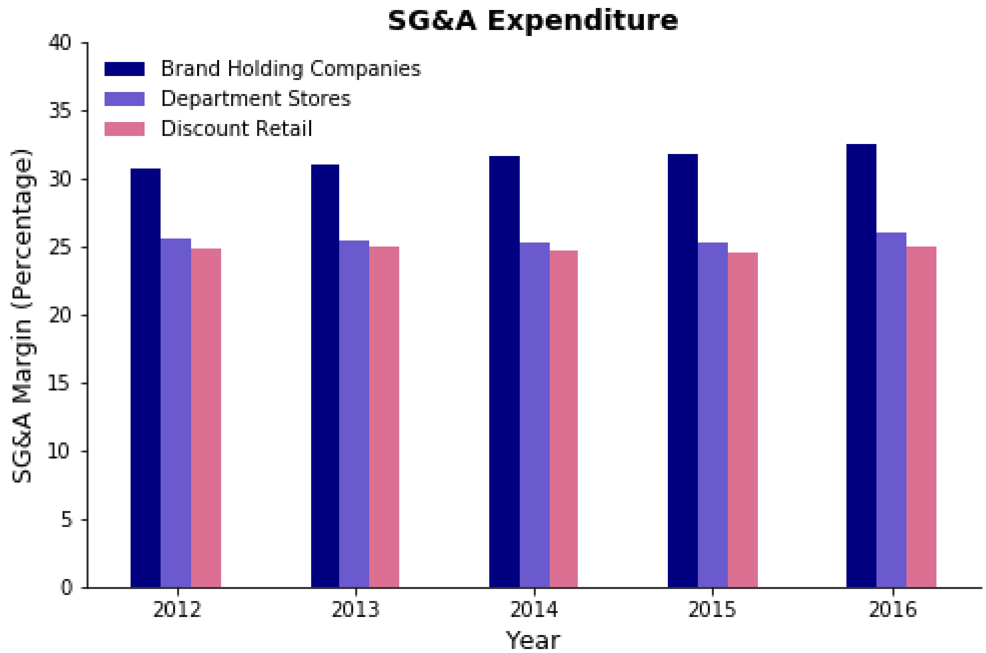
In the charts titled 'Multiple Analysis' and "Profitability Analysis," we observe parallels between the performances of the three segments. Brand holding companies widely outpace the other two segments in terms of EV/sales and profitability. Unsurprisingly, department stores lag behind significantly, likely due to overexposure to brick-and-mortar real estate as well as other factors. Finally, we observe steadily rising EV/sales and profitability from discount retailers, which is potentially due to the fact that consumers have exhibited increased frugality when purchasing goods and apparel, instead opting for experiences when allocating their discretionary spending (think MoviePass). Interestingly enough, between 2015 and 2016, the profitability trajectories of brand holding companies and discount retailers reversed, with the former outperforming the previous year and the latter underperforming the previous year.

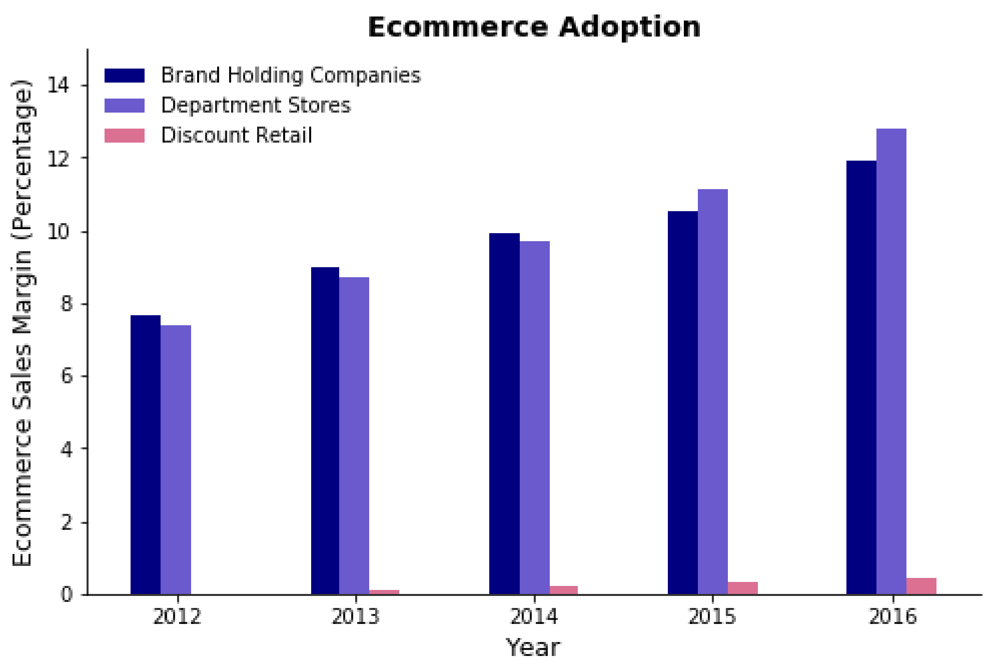
Performance Drivers

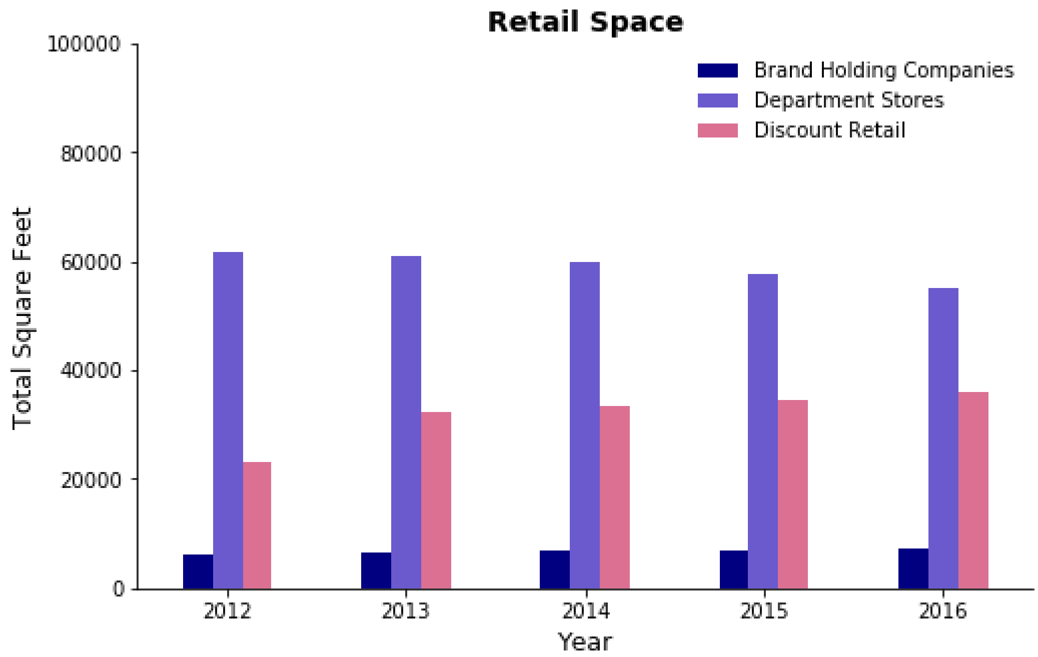
In [17]:
```
fig, ax = plt.subplots()        # create axis object axe
sga_groups.T.plot(ax=ax,        #SG&A Margin, transposed so year can be
 on x axis
                kind='bar',
                 color=['navy','slateblue','palevioletred'],
                 rot = 0,
                 figsize = (8,5))
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.legend(frameon = False)
ax.set_title('SG&A Expenditure',
             fontsize = 14,
             color = 'black',
            fontweight = 'bold')
ax.set_ylabel('SG&A Margin (Percentage)',
              fontsize = 12.5,
              color = 'black')
ax.set_ylim(0,40)
ax.set_xlabel('Year',
              fontsize = 12.5,
              color = 'black')

fig, ax = plt.subplots()
ec_groups.T.plot(ax=ax,         #E-commerce as a percentage of revenue
                kind='bar',
                 color=['navy','slateblue','palevioletred'],
                 rot = 0,
                 figsize = (8,5))
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.legend(frameon = False)
ax.set_title('Ecommerce Adoption',
             fontsize = 14,
             color = 'black',
            fontweight = 'bold')
ax.set_ylabel('Ecommerce Sales Margin (Percentage)',
              fontsize = 12.5,
```

```
              color = 'black')
ax.set_ylim(0,15)
ax.set_xlabel('Year',
              fontsize = 12.5,
              color = 'black')

fig, ax = plt.subplots()        # create axis object axe
sf_groups.T.plot(ax=ax,         #square footage
                kind='bar',
                 color=['navy','slateblue','palevioletred'],
                 rot = 0,
                 figsize = (8,5))
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.legend(frameon = False)
ax.set_title('Retail Space',
             fontsize = 14,
             color = 'black',
            fontweight = 'bold')
ax.set_ylabel('Total Square Feet',
              fontsize = 12.5,
              color = 'black')
ax.set_ylim(0,100000)
ax.set_xlabel('Year',
              fontsize = 12.5,
              color = 'black')
```

```
Out[17]: <matplotlib.text.Text at 0x10ec31dd8>
```







Performance Drivers Analysis

Brand holding companies exhibit the highest SG&A margins, comparable e-commerce adoption rates as department stores, and minimal retail space. SG&A margins and retail space have remained largely constant over the period while e-commerce sales margins have increased, falling behind department stores in 2015.

Department stores spend slightly more than discount retailers on SG&A as a percentage of revenue but still lag behind brand holding companies significantly. They have also been ramping up on encouraging consumers to utilize digital mediums when purchasing goods, indicated by the strong growth in ecommerce sales margins, outpacing brand holding companies in 2015. Department stores have the largest presence in terms of total retail space, which has gradually declined during the five year period. We observe that though department stores exhibit the strongest e-commerce platforms of the three segments, they still post the weakest performance with regards to multiples and profitability.

Finally, discount retailers spend the slightly less on SG&A than department stores. Their e-commerce presence is essentially non-existent, as four out of the five firms in our sample did not generate any revenue online at all. The quantity of retail space for discount retailers falls very much at the center between brand holding companies on the low end and department stores at the high end. We find that despite lack of e-commerce presence, discounters are faring very well. Additionally, we observe that the average quantity of total retail space per firm has slightly increased for discount retailers over the period, despite this trending the opposite way for department stores.

Regression Analysis

Next, we want to run regressions to see how these drivers will specificaly affect the performance metrics. In order for us to do this, we need to consolidate these drivers and performance metrics into one dataframe.

We will once again have to consolidate our EV/Sales data by year, only this time without the groupby. This makes things trickier because we have several columns that are not datetime. We will drop these columns, except for Ticker, which we will set as the index so that we can keep track of which company is which.

In [18]:
```
evsales.head()
```

Out[18]:

	Category	Company Name	Ticker	Measure	4/30/12	7/31/12	10/31/12	1/31/13	4/30/13	7
0	Department Stores	Kohl's Corp	KSS	EV Sales	0.82	0.77	0.84	0.77	0.76	0
1	Department Stores	Macy's Inc	M	EV Sales	0.80	0.75	0.79	0.78	0.78	0
2	Department Stores	Nordstrom Inc	JWN	EV Sales	1.19	1.08	1.17	1.10	1.05	1
3	Department Stores	Dillard's	DDS	EV Sales	0.58	0.60	0.66	0.71	0.65	0
4	Department Stores	JC Penney Co Inc	JCP	EV Sales	0.57	0.48	0.50	0.46	0.44	0

5 rows × 24 columns

In [19]:
```
evsales = evsales.drop(['Category','Company Name', 'Measure'], axis=1)
evsales = evsales.set_index(['Ticker'])
```

In [20]:
```
evsales.head()
```

Out[20]:

	4/30/12	7/31/12	10/31/12	1/31/13	4/30/13	7/31/13	10/31/13	1/31/14	4/30/14	7
Ticker										
KSS	0.82	0.77	0.84	0.77	0.76	0.81	0.80	0.83	0.81	0
M	0.80	0.75	0.79	0.78	0.78	0.86	0.81	0.91	0.93	0
JWN	1.19	1.08	1.17	1.10	1.05	1.12	1.06	1.11	1.08	1
DDS	0.58	0.60	0.66	0.71	0.65	0.70	0.65	0.70	0.69	0
JCP	0.57	0.48	0.50	0.46	0.44	0.55	0.57	0.57	0.54	0

Transforming so that the dates become the index

In [21]:
```
evsales = evsales.T
evsales.head()
```

Out[21]:

Ticker	KSS	M	JWN	DDS	JCP	SHLD	DSW	BONT	FINL	DKS	...	GCO	FOSL	CA
4/30/12	0.82	0.80	1.19	0.58	0.57	0.23	0.98	0.34	0.63	0.98	...	0.74	2.56	0.2
7/31/12	0.77	0.75	1.08	0.60	0.48	0.20	1.03	0.36	0.58	0.99	...	0.66	2.19	0.3
10/31/12	0.84	0.79	1.17	0.66	0.50	0.21	1.17	0.37	0.59	1.05	...	0.69	1.81	0.3
1/31/13	0.77	0.78	1.10	0.71	0.46	0.21	1.20	0.42	0.52	1.01	...	0.58	1.94	0.4
4/30/13	0.76	0.78	1.05	0.65	0.44	0.20	1.19	0.39	0.52	0.96	...	0.56	2.10	0.3

5 rows × 41 columns

In [22]:
```
evsales = evsales.set_index(pd.DatetimeIndex(evsales.index))
evsales.index
```

Out[22]:
```
DatetimeIndex(['2012-04-30', '2012-07-31', '2012-10-31', '2013-01-31',
               '2013-04-30', '2013-07-31', '2013-10-31', '2014-01-31',
               '2014-04-30', '2014-07-31', '2014-10-31', '2015-01-31',
               '2015-04-30', '2015-07-31', '2015-10-31', '2016-01-31',
               '2016-04-30', '2016-07-31', '2016-10-31', '2017-01-31'],
              dtype='datetime64[ns]', freq=None)
```

We will once again go through what we did before to consolidate into annual data

In [23]:
```
evsales_quarter = evsales.resample("Q-OCT").mean()
evsales_quarter.head()
```

Out[23]:

Ticker	KSS	M	JWN	DDS	JCP	SHLD	DSW	BONT	FINL	DKS	...	GCO	FOSL	CAL
2012-04-30	0.82	0.80	1.19	0.58	0.57	0.23	0.98	0.34	0.63	0.98	...	0.74	2.56	0.29
2012-07-31	0.77	0.75	1.08	0.60	0.48	0.20	1.03	0.36	0.58	0.99	...	0.66	2.19	0.30
2012-10-31	0.84	0.79	1.17	0.66	0.50	0.21	1.17	0.37	0.59	1.05	...	0.69	1.81	0.37
2013-01-31	0.77	0.78	1.10	0.71	0.46	0.21	1.20	0.42	0.52	1.01	...	0.58	1.94	0.41
2013-04-30	0.76	0.78	1.05	0.65	0.44	0.20	1.19	0.39	0.52	0.96	...	0.56	2.10	0.39

5 rows × 41 columns

In [24]:
```
evsales_quarter = evsales_quarter.drop(evsales_quarter.columns[-1],
                                       axis=1) #dropping the last NaN co
lumn
evsales_quarter = evsales_quarter.rolling(window=4).mean()
#taking the rolling average for the past 4 quarters
evsales_yearly = evsales_quarter.resample("12M", closed="left",loffset=
"-3M").last()
evsales_yearly
```

Out[24]:

Ticker	KSS	M	JWN	DDS	JCP	SHLD	DSW	BONT	FINL	DKS	...	
2013-01-31	0.8000	0.7800	1.1350	0.6375	0.5025	0.2125	1.0950	0.3725	0.5800	1.0075	...	1.
2014-01-31	0.8000	0.8400	1.0850	0.6750	0.5325	0.2300	1.3625	0.4150	0.6250	1.0600	...	0.
2015-01-31	0.8275	0.9600	1.1900	0.7800	0.5700	0.2275	1.1050	0.3975	0.6475	0.9200	...	0.
2016-01-31	0.8175	0.9075	1.1275	0.6825	0.5525	0.2025	0.9700	0.3775	0.5025	0.8275	...	0.
2017-01-31	0.6325	0.6925	0.7375	0.4525	0.5575	0.1975	0.6550	0.3800	0.4175	0.7750	...	0.

5 rows × 40 columns

In [25]:
```
# convert index to Years (string) so it's consistent with the rest of ou
r dataframes
evsales_yearly.index = evsales_yearly.index.astype(str, copy=True)
evsales_yearly = evsales_yearly.rename(index={'2013-01-31': '2012',
                                                              '2014-01-3
1':'2013',
                                                              '2015-01-3
1':'2014',
                                                              '2016-01-3
1':'2015',
                                                              '2017-01-3
1':'2016'})
```

In [26]:
```
evsales_yearly.head()
```

Out[26]:

Ticker	KSS	M	JWN	DDS	JCP	SHLD	DSW	BONT	FINL	DKS	...	
2012	0.8000	0.7800	1.1350	0.6375	0.5025	0.2125	1.0950	0.3725	0.5800	1.0075	...	1.
2013	0.8000	0.8400	1.0850	0.6750	0.5325	0.2300	1.3625	0.4150	0.6250	1.0600	...	0.
2014	0.8275	0.9600	1.1900	0.7800	0.5700	0.2275	1.1050	0.3975	0.6475	0.9200	...	0.
2015	0.8175	0.9075	1.1275	0.6825	0.5525	0.2025	0.9700	0.3775	0.5025	0.8275	...	0.
2016	0.6325	0.6925	0.7375	0.4525	0.5575	0.1975	0.6550	0.3800	0.4175	0.7750	...	0.

5 rows × 40 columns

Now we need to transpose it back so that we can append it to the rest of our data.

In [27]:
```
evsales_yearly = evsales_yearly.T.reset_index()
evsales_yearly.head()
```

Out[27]:

	Ticker	2012	2013	2014	2015	2016
0	KSS	0.8000	0.8000	0.8275	0.8175	0.6325
1	M	0.7800	0.8400	0.9600	0.9075	0.6925
2	JWN	1.1350	1.0850	1.1900	1.1275	0.7375
3	DDS	0.6375	0.6750	0.7800	0.6825	0.4525
4	JCP	0.5025	0.5325	0.5700	0.5525	0.5575

Great, except we don't know what this is! Let's add back the Measure column so we know it's EV/Sales

In [28]:
```
evsales_yearly['Measure'] = 'EV Sales'
evsales_yearly.head()
```

Out[28]:

	Ticker	2012	2013	2014	2015	2016	Measure
0	KSS	0.8000	0.8000	0.8275	0.8175	0.6325	EV Sales
1	M	0.7800	0.8400	0.9600	0.9075	0.6925	EV Sales
2	JWN	1.1350	1.0850	1.1900	1.1275	0.7375	EV Sales
3	DDS	0.6375	0.6750	0.7800	0.6825	0.4525	EV Sales
4	JCP	0.5025	0.5325	0.5700	0.5525	0.5575	EV Sales

Now we're ready to append.

In [29]:

```
combo1 = sga_margin.append(ec_margin) # Append E-Commerce data to SG&A M
argin Data
combo2 = combo1.append(square_footage) # add on square footage data
combo_evs = evsales_yearly.append(combo2) # add on EV/Sales Data
```

In [30]:
```
combo_evs
```

Out[30]:

	2012	2013	2014	2015	2016	Category	Company Name
0	0.8000	0.8000	0.8275	0.8175	0.6325	NaN	NaN
1	0.7800	0.8400	0.9600	0.9075	0.6925	NaN	NaN
2	1.1350	1.0850	1.1900	1.1275	0.7375	NaN	NaN
3	0.6375	0.6750	0.7800	0.6825	0.4525	NaN	NaN
4	0.5025	0.5325	0.5700	0.5525	0.5575	NaN	NaN
5	0.2125	0.2300	0.2275	0.2025	0.1975	NaN	NaN
6	1.0950	1.3625	1.1050	0.9700	0.6550	NaN	NaN
7	0.3725	0.4150	0.3975	0.3775	0.3800	NaN	NaN
8	0.5800	0.6250	0.6475	0.5025	0.4175	NaN	NaN
9	1.0075	1.0600	0.9200	0.8275	0.7750	NaN	NaN
10	0.7175	0.7175	0.9575	1.1425	1.0400	NaN	NaN
11	0.3725	0.4050	0.4725	0.4250	0.3375	NaN	NaN
12	1.2550	1.4100	1.4650	1.5575	1.5575	NaN	NaN
13	1.4625	1.4375	1.4875	1.8175	1.9300	NaN	NaN
14	NaN	NaN	0.8625	1.0450	1.1925	NaN	NaN
15	0.1875	0.3825	0.4125	0.4275	0.3475	NaN	NaN
16	0.2050	0.5150	0.8100	0.5450	0.2925	NaN	NaN
17	1.8150	2.2850	2.5500	3.1200	2.6400	NaN	NaN
18	1.8075	2.0175	2.4800	2.5850	2.2825	NaN	NaN
19	1.6625	1.8650	2.0825	2.6025	2.0650	NaN	NaN
20	3.7125	2.8525	2.1900	2.0650	2.2700	NaN	NaN
21	1.0950	1.4850	2.2325	2.6100	2.2900	NaN	NaN
22	1.9925	2.1150	1.8325	1.4325	1.0875	NaN	NaN
23	6.0800	5.5500	4.7925	2.2050	1.7600	NaN	NaN
24	3.1850	3.3300	4.8525	5.5475	3.8950	NaN	NaN
25	0.7225	0.6950	0.6125	0.4025	0.3500	NaN	NaN
26	0.9500	0.8325	0.6575	0.8350	0.7650	NaN	NaN
27	0.9175	0.7100	0.5775	0.5775	0.4900	NaN	NaN
28	0.9500	1.1300	1.1100	0.8800	0.6400	NaN	NaN
29	1.3250	1.5825	1.6175	1.7650	1.7025	NaN	NaN
...	...	...	...	...	...	...	...

	2012	2013	2014	2015	2016	Category	Company Name
10	12316.0000	12705.0000	12734.0000	12918.0000	13118.0000	Department Stores	Foot Locker Inc
11	15633.0000	15799.0000	15409.0000	15130.0000	14588.0000	Department Stores	Stage Stores Inc
12	69984.0000	73209.0000	76537.0000	80480.0000	83798.0000	Discount Retail	TJX Cos Inc
13	27800.0000	28900.0000	30400.0000	31900.0000	33300.0000	Discount Retail	Ross Stores Inc
14	0.0000	41680.0000	42276.0000	43659.0000	44992.0000	Discount Retail	Burlington Store Inc
15	9205.0000	9240.0000	8640.0000	8896.0000	9280.0000	Discount Retail	Stein Mart
16	8641.0000	8593.0000	8341.0000	8326.0000	8507.0000	Discount Retail	Tuesday Morning
17	8495.0000	9896.0000	10907.0000	12326.0000	13436.0000	Brand Holding Companies	Nike Inc
18	5559.0000	5942.0000	6960.0000	7660.0000	8245.0000	Brand Holding Companies	VF Corp
19	13561.0000	13961.0000	14418.0000	14878.0000	15495.0000	Brand Holding Companies	L Brands
20	2414.0000	2703.0000	2962.0000	3040.0000	3096.0000	Brand Holding Companies	Tapestry Inc
21	1125.0000	1286.0000	1196.0000	1206.0000	1206.0000	Brand Holding Companies	Hanesbrands Inc.
22	2800.0000	3000.0000	3000.0000	3600.0000	3800.0000	Brand Holding Companies	Ralph Lauren Corp

	2012	2013	2014	2015	2016	Category	Company Name
23	702.0000	944.0000	1376.0000	1777.0000	2258.0000	Brand Holding Companies	Michael Kors Holding Ltd
24	545.0000	683.0000	835.0000	1246.0000	1587.0000	Brand Holding Companies	Under Armour Inc
25	7958.0000	7736.0000	7517.0000	7292.0000	7007.0000	Brand Holding Companies	Abercrombie & Fitch Co.
26	4963.0000	5206.0000	5295.0000	5285.0000	5312.0000	Brand Holding Companies	American Eagle Outfitters
27	20800.0000	21000.0000	21200.0000	21200.0000	26900.0000	Brand Holding Companies	Ascena Retail Group
28	36900.0000	37200.0000	38100.0000	37900.0000	36700.0000	Brand Holding Companies	Gap Inc
29	3082.0000	3452.0000	3834.0000	4327.0000	4902.0000	Brand Holding Companies	Carter's, Inc
30	3271.0000	3547.0000	3706.0000	3652.0000	3612.0000	Brand Holding Companies	Chico's FAS, Inc
31	4038.0000	4341.0000	4687.0000	4844.0000	4522.0000	Brand Holding Companies	Genesco Inc
32	791.0000	919.0000	981.0000	1087.0000	1029.0000	Brand Holding Companies	Fossil Group Inc
33	7551.0000	7378.0000	7260.0000	7243.0000	7288.0000	Brand Holding Companies	Caleres Inc
34	596.0000	740.0000	894.0000	1071.0000	1190.0000	Brand Holding Companies	Lululemon Athletica Inc
35	2371.0000	2329.0000	2301.0000	2211.0000	2198.0000	Brand Holding Companies	Guess?, Inc

	2012	2013	2014	2015	2016	Category	Company Name
36	5423.0000	5498.0000	5529.0000	5640.0000	5662.0000	Brand Holding Companies	Express, Inc
37	7108.0000	7116.0000	10113.0000	10018.0000	9483.0000	Brand Holding Companies	Tailored Brands
38	188.0000	211.0000	294.0000	315.0000	355.0000	Brand Holding Companies	Steve Madden, Ltd
39	3409.0000	3595.0000	3833.0000	3953.0000	4132.0000	Brand Holding Companies	Urban Outfitters, Inc

160 rows × 9 columns

The index is repeating, so we have to tell it to be a continuous stream.

In [31]:

```
combo_evs.index = range(len(combo_evs.index))
combo_evs
```

Out[31]:

	2012	2013	2014	2015	2016	Category	Compar Nam
0	0.8000	0.8000	0.8275	0.8175	0.6325	NaN	NaN
1	0.7800	0.8400	0.9600	0.9075	0.6925	NaN	NaN
2	1.1350	1.0850	1.1900	1.1275	0.7375	NaN	NaN
3	0.6375	0.6750	0.7800	0.6825	0.4525	NaN	NaN
4	0.5025	0.5325	0.5700	0.5525	0.5575	NaN	NaN
5	0.2125	0.2300	0.2275	0.2025	0.1975	NaN	NaN
6	1.0950	1.3625	1.1050	0.9700	0.6550	NaN	NaN
7	0.3725	0.4150	0.3975	0.3775	0.3800	NaN	NaN
8	0.5800	0.6250	0.6475	0.5025	0.4175	NaN	NaN
9	1.0075	1.0600	0.9200	0.8275	0.7750	NaN	NaN
10	0.7175	0.7175	0.9575	1.1425	1.0400	NaN	NaN
11	0.3725	0.4050	0.4725	0.4250	0.3375	NaN	NaN
12	1.2550	1.4100	1.4650	1.5575	1.5575	NaN	NaN
13	1.4625	1.4375	1.4875	1.8175	1.9300	NaN	NaN
14	NaN	NaN	0.8625	1.0450	1.1925	NaN	NaN
15	0.1875	0.3825	0.4125	0.4275	0.3475	NaN	NaN
16	0.2050	0.5150	0.8100	0.5450	0.2925	NaN	NaN
17	1.8150	2.2850	2.5500	3.1200	2.6400	NaN	NaN
18	1.8075	2.0175	2.4800	2.5850	2.2825	NaN	NaN
19	1.6625	1.8650	2.0825	2.6025	2.0650	NaN	NaN
20	3.7125	2.8525	2.1900	2.0650	2.2700	NaN	NaN
21	1.0950	1.4850	2.2325	2.6100	2.2900	NaN	NaN
22	1.9925	2.1150	1.8325	1.4325	1.0875	NaN	NaN
23	6.0800	5.5500	4.7925	2.2050	1.7600	NaN	NaN
24	3.1850	3.3300	4.8525	5.5475	3.8950	NaN	NaN
25	0.7225	0.6950	0.6125	0.4025	0.3500	NaN	NaN
26	0.9500	0.8325	0.6575	0.8350	0.7650	NaN	NaN
27	0.9175	0.7100	0.5775	0.5775	0.4900	NaN	NaN
28	0.9500	1.1300	1.1100	0.8800	0.6400	NaN	NaN
29	1.3250	1.5825	1.6175	1.7650	1.7025	NaN	NaN
...	...	...	...	...	...	...	...

	2012	2013	2014	2015	2016	Category	Compar Nam
130	12316.0000	12705.0000	12734.0000	12918.0000	13118.0000	Department Stores	Foot Locke Inc
131	15633.0000	15799.0000	15409.0000	15130.0000	14588.0000	Department Stores	Stage Store Inc
132	69984.0000	73209.0000	76537.0000	80480.0000	83798.0000	Discount Retail	TJX Cos Inc
133	27800.0000	28900.0000	30400.0000	31900.0000	33300.0000	Discount Retail	Ross Stores Inc
134	0.0000	41680.0000	42276.0000	43659.0000	44992.0000	Discount Retail	Burlington Store Inc
135	9205.0000	9240.0000	8640.0000	8896.0000	9280.0000	Discount Retail	Stein Mart
136	8641.0000	8593.0000	8341.0000	8326.0000	8507.0000	Discount Retail	Tuesday Morning
137	8495.0000	9896.0000	10907.0000	12326.0000	13436.0000	Brand Holding Companies	Nike Inc
138	5559.0000	5942.0000	6960.0000	7660.0000	8245.0000	Brand Holding Companies	VF Corp
139	13561.0000	13961.0000	14418.0000	14878.0000	15495.0000	Brand Holding Companies	L Brands
140	2414.0000	2703.0000	2962.0000	3040.0000	3096.0000	Brand Holding Companies	Tapestry Inc
141	1125.0000	1286.0000	1196.0000	1206.0000	1206.0000	Brand Holding Companies	Hanesbranc Inc.
142	2800.0000	3000.0000	3000.0000	3600.0000	3800.0000	Brand Holding Companies	Ralph Laure Corp

	2012	2013	2014	2015	2016	Category	Compar Nam
143	702.0000	944.0000	1376.0000	1777.0000	2258.0000	Brand Holding Companies	Michael Koı Holding Ltd
144	545.0000	683.0000	835.0000	1246.0000	1587.0000	Brand Holding Companies	Under Armour Inc
145	7958.0000	7736.0000	7517.0000	7292.0000	7007.0000	Brand Holding Companies	Abercrombi & Fitch Co.
146	4963.0000	5206.0000	5295.0000	5285.0000	5312.0000	Brand Holding Companies	American Eagle Outfitters
147	20800.0000	21000.0000	21200.0000	21200.0000	26900.0000	Brand Holding Companies	Ascena Retail Grou
148	36900.0000	37200.0000	38100.0000	37900.0000	36700.0000	Brand Holding Companies	Gap Inc
149	3082.0000	3452.0000	3834.0000	4327.0000	4902.0000	Brand Holding Companies	Carter's, Inc
150	3271.0000	3547.0000	3706.0000	3652.0000	3612.0000	Brand Holding Companies	Chico's FAS Inc
151	4038.0000	4341.0000	4687.0000	4844.0000	4522.0000	Brand Holding Companies	Genesco In
152	791.0000	919.0000	981.0000	1087.0000	1029.0000	Brand Holding Companies	Fossil Grou Inc
153	7551.0000	7378.0000	7260.0000	7243.0000	7288.0000	Brand Holding Companies	Caleres Inc
154	596.0000	740.0000	894.0000	1071.0000	1190.0000	Brand Holding Companies	Lululemon Athletica Inc
155	2371.0000	2329.0000	2301.0000	2211.0000	2198.0000	Brand Holding Companies	Guess?, Inc

	2012	2013	2014	2015	2016	Category	Compa Nam
156	5423.0000	5498.0000	5529.0000	5640.0000	5662.0000	Brand Holding Companies	Express, Inc
157	7108.0000	7116.0000	10113.0000	10018.0000	9483.0000	Brand Holding Companies	Tailored Brands
158	188.0000	211.0000	294.0000	315.0000	355.0000	Brand Holding Companies	Steve Madden, Lt
159	3409.0000	3595.0000	3833.0000	3953.0000	4132.0000	Brand Holding Companies	Urban Outfitters, Inc

160 rows × 9 columns

For the regression we will be looking at how these companies changed over the five periods, so we will create a new column for this.

In [32]:
```
combo_evs['Delta'] = combo_evs['2016']/combo_evs['2012'] - 1
```

Next, we will make new dataframes based on "Measure" and merge them together so that each company will have multiple columns that are change in measure over the 5 years. We will also correct for the companies that do not have a number in their cell for the year 2012.

In [33]:

```
combo_evs.at[134, 'Delta'] = combo_evs.at[134, '2016']/combo_evs.at[134,
 '2013'] - 1 # Correct for Burlington

evs_delta = combo_evs[combo_evs['Measure'] == 'EV Sales'] # selecting EV
 Sales
evs_delta.at[14, 'Delta'] = evs_delta.at[14, '2016']/evs_delta.at[14, '2
014'] - 1 #correct for Burlington
evs_delta.at[15, 'Delta'] = evs_delta.at[15, '2016']/evs_delta.at[15, '2
013'] - 1 #correct for Stein Mart
evs_delta.at[23, 'Delta'] = evs_delta.at[23, '2016']/evs_delta.at[23, '2
014'] - 1 #correct for MK

sga_delta = combo_evs[combo_evs['Measure'] == 'SG&A Margin'] # selecting
 SG&A
sga_delta.at[54, 'Delta'] = sga_delta.at[54, '2016']/sga_delta.at[54, '2
014'] - 1 # correct for Burlington
sga_delta.at[55, 'Delta'] = sga_delta.at[55, '2016']/sga_delta.at[55, '2
013'] - 1 #correct for Stein Mart
sga_delta.at[63, 'Delta'] = sga_delta.at[63, '2016']/sga_delta.at[63, '2
014'] - 1 #correct for MK

ec_delta = combo_evs[combo_evs['Measure'] == 'Ecommerce Sales Margin'] #
 selecting E-commerce
ec_delta.at[94, 'Delta'] = ec_delta.at[94, '2016']/ec_delta.at[94, '201
4'] - 1 #correct for Burlington
ec_delta.at[95, 'Delta'] = ec_delta.at[95, '2016']/ec_delta.at[95, '201
3'] - 1  # correct for Stein Mart
ec_delta.at[103, 'Delta'] = ec_delta.at[103, '2016']/ec_delta.at[103, '2
014'] - 1 #correct for MK
ec_delta = ec_delta.fillna(0)  # change discount retailers with 0 ecomme
rce sales from NaN to 0

sf_delta = combo_evs[combo_evs['Measure'] == 'Total Square Footage']
sf_delta.at[134, 'Delta'] = sf_delta.at[134, '2016']/sf_delta.at[134, '2
014'] - 1 #correct for Burlington
sf_delta.at[135, 'Delta'] = sf_delta.at[135, '2016']/sf_delta.at[135, '2
013'] - 1  # correct for Stein Mart
sf_delta.at[143, 'Delta'] = sf_delta.at[143, '2016']/sf_delta.at[143, '2
014'] - 1 #correct for MK
```

```
/Users/Crystal/anaconda/lib/python3.6/site-packages/ipykernel_launcher.
py:14: RuntimeWarning: invalid value encountered in double_scalars
```

Regression - EV/Sales

In [34]:
```
#Create a shortcut for the columns that we want to take out from the dat
aframes,
# that we will then use to merge to form a new dataframe
col_list = ['Ticker', 'Delta']

#Merge based on companies
a_delta1 = pd.merge(evs_delta[col_list], sga_delta[col_list], how='left'
, on='Ticker')
b_delta1 = pd.merge(a_delta1, ec_delta[col_list],how='left',on='Ticker')
finaldelta1 = pd.merge(b_delta1, sf_delta[['Category',
                                           'Company Name',
                                           'Ticker',
                                           'Delta']],how='left',on='Tick
er')
finaldelta1.columns = ['Ticker',
                       'EV_Sales',
                       'SGA_Margin',
                       'Ecommerce_Margin',
                       'Category',
                       'Company_Name',
                       'Square_Footage']

finaldelta1.head()
```

Out[34]:

	Ticker	EV_Sales	SGA_Margin	Ecommerce_Margin	Category	Company_Name	Squ
0	KSS	-0.209375	0.072300	1.050734	Department Stores	Kohl's Corp	-0.0
1	M	-0.112179	0.046345	0.847936	Department Stores	Macy's Inc	-0.1
2	JWN	-0.350220	0.056740	0.758871	Department Stores	Nordstrom Inc	0.17
3	DDS	-0.290196	0.042003	0.789655	Department Stores	Dillard's	-0.0
4	JCP	0.109453	-0.187320	0.571247	Department Stores	JC Penney Co Inc	-0.0

To make things consistent, we will set the order of the columns

In [35]:
```
columnorder = ['Category', 'Company_Name', 'Ticker', 'EV_Sales', 'SGA_Ma
rgin', 'Ecommerce_Margin', 'Square_Footage']
finaldelta1 = finaldelta1.reindex(columns=columnorder)

finaldelta1.head()
```

Out[35]:

	Category	Company_Name	Ticker	EV_Sales	SGA_Margin	Ecommerce_Margin	Squ
0	Department Stores	Kohl's Corp	KSS	-0.209375	0.072300	1.050734	-0.0
1	Department Stores	Macy's Inc	M	-0.112179	0.046345	0.847936	-0.1
2	Department Stores	Nordstrom Inc	JWN	-0.350220	0.056740	0.758871	0.17
3	Department Stores	Dillard's	DDS	-0.290196	0.042003	0.789655	-0.0
4	Department Stores	JC Penney Co Inc	JCP	0.109453	-0.187320	0.571247	-0.0

Now we have the dataframe that we can run the regression on.

Regressing EV/Sales for All Sectors

In [36]:
```
evs_results = smf.ols("EV_Sales ~ SGA_Margin + Ecommerce_Margin + Square
_Footage",
                    data=finaldelta1).fit()

print(evs_results.summary())
```

OLS Regression Results

```
==============================================================================
=======
Dep. Variable:                EV_Sales   R-squared:
  0.127
Model:                             OLS   Adj. R-squared:
  0.054
Method:                  Least Squares   F-statistic:
  1.743
Date:                 Thu, 21 Dec 2017   Prob (F-statistic):
  0.176
Time:                         23:47:02   Log-Likelihood:
-15.274
No. Observations:                   40   AIC:
  38.55
Df Residuals:                       36   BIC:
  45.30
Df Model:                            3

Covariance Type:             nonrobust

==============================================================================
=============
                      coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------------
-------------
Intercept           0.0121      0.089      0.136      0.893      -0.16
9       0.193
SGA_Margin         -1.1567      0.574     -2.017      0.051      -2.32
0       0.007
Ecommerce_Margin   -0.0688      0.062     -1.105      0.276      -0.19
5       0.057
Square_Footage      0.0695      0.157      0.444      0.660      -0.24
8       0.387
==============================================================================
=======
Omnibus:                        10.865   Durbin-Watson:
  1.975
Prob(Omnibus):                   0.004   Jarque-Bera (JB):
11.656
Skew:                            0.898   Prob(JB):
0.00294
Kurtosis:                        4.942   Cond. No.
   14.7
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

Negative coefficients for SG&A Margin and Ecommerce Margin indicate negative relationships between EV/Sales growth over time and growths of SG&A Margin and Ecommerce Margin (only slightly negative). It is logical for investors to react adversely to increased SG&A spending, which would definitely whittle a firm's bottom line and could potentially indicate inefficient cost allocation.

Next we'd like to see how these relationships might be different depending on the type of retail. We will run three regressions, one for each category.

In [37]:
```
# Make seperate dataframes based on category
ds_evs = finaldelta1[finaldelta1['Category'] == 'Department Stores']
dr_evs = finaldelta1[finaldelta1['Category'] == 'Discount Retail']
bhc_evs = finaldelta1[finaldelta1['Category'] == 'Brand Holding Companie
s']
```

Regressing EV/Sales for Department Store

In [38]:

```
#Run regression for EV/Sales department stores
ds_evs_results = smf.ols("EV_Sales ~ SGA_Margin + Ecommerce_Margin + Squ
are_Footage",
                  data=ds_evs).fit()

print(ds_evs_results.summary())
```

```
                            OLS Regression Results
========================================================================
=======
Dep. Variable:                EV_Sales   R-squared:
  0.491
Model:                             OLS   Adj. R-squared:
  0.301
Method:                  Least Squares   F-statistic:
  2.577
Date:                 Thu, 21 Dec 2017   Prob (F-statistic):
  0.127
Time:                         23:47:02   Log-Likelihood:
4.9339
No. Observations:                   12   AIC:
-1.868
Df Residuals:                        8   BIC:
0.07190
Df Model:                            3

Covariance Type:             nonrobust

========================================================================
=============
                      coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------
-------------
Intercept          -0.1051      0.123     -0.858      0.416      -0.38
8       0.178
SGA_Margin         -1.8050      0.752     -2.399      0.043      -3.54
0      -0.070
Ecommerce_Margin    0.0324      0.128      0.252      0.807      -0.26
4       0.329
Square_Footage     -0.5870      0.361     -1.628      0.142      -1.41
9       0.245
========================================================================
=======
Omnibus:                         6.954   Durbin-Watson:
  1.978
Prob(Omnibus):                   0.031   Jarque-Bera (JB):
  3.177
Skew:                            1.155   Prob(JB):
  0.204
Kurtosis:                        4.008   Cond. No.
   18.3
========================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/scipy/stats/stats.p
y:1334: UserWarning: kurtosistest only valid for n>=20 ... continuing a
nyway, n=12
  "anyway, n=%i" % int(n))
```

These coefficients tell us for department stores, there is a strongly negative relationship between changes in EV/Sales and SG&A Margin, and a negative relationship between changes in EV/Sales and Square Footage. The second observation indicates that department stores tend to garner worse sentiments from investors when they expand their brick-and-mortar operations. In the case of department stores, we observe a positive, but potentially insignificant relationship between EV/Sales growth and Ecommerce Margin growth.

Regressing EV/Sales for Discount Retail

In [39]:

```
#Run regression for EV/Sales discount retail stores
dr_evs_results = smf.ols("EV_Sales ~ SGA_Margin + Ecommerce_Margin + Squ
are_Footage",
                  data=dr_evs).fit()

print(dr_evs_results.summary())
```

```
                            OLS Regression Results
=======================================================================
=======
Dep. Variable:               EV_Sales   R-squared:
  1.000
Model:                            OLS   Adj. R-squared:
  0.999
Method:                 Least Squares   F-statistic:
  1072.
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
0.0224
Time:                        23:47:02   Log-Likelihood:
21.546
No. Observations:                   5   AIC:
-35.09
Df Residuals:                       1   BIC:
-36.65
Df Model:                           3

Covariance Type:            nonrobust

=======================================================================
=============
                      coef    std err          t      P>|t|      [0.02
5      0.975]
-----------------------------------------------------------------------
-------------
Intercept           0.3967      0.007     60.998      0.010       0.31
4       0.479
SGA_Margin         -1.1641      0.151     -7.703      0.082      -3.08
4       0.756
Ecommerce_Margin   -0.1111      0.002    -45.688      0.014      -0.14
2      -0.080
Square_Footage     -0.4422      0.052     -8.464      0.075      -1.10
6       0.222
=======================================================================
=======
Omnibus:                          nan   Durbin-Watson:
  2.336
Prob(Omnibus):                    nan   Jarque-Bera (JB):
  0.554
Skew:                          -0.750   Prob(JB):
  0.758
Kurtosis:                       2.362   Cond. No.
   93.8
=======================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/statsmodels/stats/s
tattools.py:72: ValueWarning: omni_normtest is not valid with less than
8 observations; 5 samples were given.
  "samples were given." % int(n), ValueWarning)
```

Wow, that's a high R-squared! This is probably because we have a low number of observations and many discount retailers did not have an E-commerce margin. Let's try it without that variable.

In [40]:
```
#Run regression for EV/Sales discount retail stores
dr_evs_results = smf.ols("EV_Sales ~ SGA_Margin + Square_Footage",
                  data=dr_evs).fit()

print(dr_evs_results.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:               EV_Sales   R-squared:
  0.351
Model:                            OLS   Adj. R-squared:
-0.298
Method:                 Least Squares   F-statistic:
0.5408
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
  0.649
Time:                        23:47:02   Log-Likelihood:
2.4359
No. Observations:                   5   AIC:
  1.128
Df Residuals:                       2   BIC:                             -
0.04344
Df Model:                           2

Covariance Type:            nonrobust

==============================================================================
===========
                    coef    std err          t      P>|t|      [0.025
     0.975]
------------------------------------------------------------------------------
-----------
Intercept         0.1903      0.151      1.259      0.335      -0.460
      0.841
SGA_Margin       -4.3382      4.337     -1.000      0.423     -22.998
     14.321
Square_Footage    1.0169      1.336      0.761      0.526      -4.731
      6.765
==============================================================================
=======
Omnibus:                          nan   Durbin-Watson:
  2.877
Prob(Omnibus):                    nan   Jarque-Bera (JB):
  0.474
Skew:                          -0.415   Prob(JB):
  0.789
Kurtosis:                       1.740   Cond. No.
   42.0
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/statsmodels/stats/s
tattools.py:72: ValueWarning: omni_normtest is not valid with less than
8 observations; 5 samples were given.
  "samples were given." % int(n), ValueWarning)
```

This looks much more reasonable now. These results indicate there there may also be a strong negative relationship between changes in EV/Sales and SG&A Margin for discount retailers, but a positive relationship between changes in EV/Sales and Square Footage. These are both logical as discount retailers rely on cost cutting and also depend heavily (almost completely) on brick-and-mortar operation to generate revenue.

Regressing EV/Sales for Brand Holding Companies

In [41]:
```
#Run regression for EV/Sales Brands
bhc_evs_results = smf.ols("EV_Sales ~ SGA_Margin + Ecommerce_Margin + Sq
uare_Footage",
                  data=bhc_evs).fit()

print(bhc_evs_results.summary())
```

```
                            OLS Regression Results

==============================================================================
=======
Dep. Variable:               EV_Sales   R-squared:
  0.079
Model:                            OLS   Adj. R-squared:
-0.067
Method:                 Least Squares   F-statistic:
0.5399
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
  0.661
Time:                        23:47:02   Log-Likelihood:
-12.425
No. Observations:                  23   AIC:
  32.85
Df Residuals:                      19   BIC:
  37.39
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
=============
                       coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------------
-------------
Intercept           -0.0916      0.160     -0.573      0.573      -0.42
6       0.243
SGA_Margin          -0.8721      0.805     -1.083      0.292      -2.55
7       0.813
Ecommerce_Margin    -0.0478      0.106     -0.452      0.656      -0.26
9       0.173
Square_Footage       0.1448      0.218      0.666      0.514      -0.31
0       0.600
==============================================================================
=======
Omnibus:                       11.524   Durbin-Watson:
  2.177
Prob(Omnibus):                  0.003   Jarque-Bera (JB):
  9.711
Skew:                           1.300   Prob(JB):
0.00779
Kurtosis:                       4.837   Cond. No.
   12.9
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

As observed in both department stores and discount retailers, we also observe a negative relationship between EV/Sales growth and increased SG&A spending for brand holding companies. Like in discount retailers, we observe a positive relationship between EV/Sales growth and increased retail square footage. This could mean that due to the strong performance and minimal brick-and-mortar operations of brand holding companies, this segment benefits from conservatively growing their number of physical locations.

Regression - Net Income Margin

Now we will run regressions using Net Income Margin as the performance metric, repeating what we did for EV/Sales

In [42]:
```
#Add the three drivers (combo2) to the net income margin dataframe
combo_ni = net_income_margin.append(combo2)
combo_ni.index = range(len(combo_ni.index)) #make index a continuous non
-repeating stream of numbers
#Make column that shows the percent change from 2012 to 2016
combo_ni['Delta'] = combo_ni['2016']/combo_ni['2012'] - 1
```

In [43]:
```
combo_ni
```

Out[43]:

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015	
0	Department Stores	Kohl's Corp	KSS	Net Income Margin	5.11	4.67	4.56	3.50	2
1	Department Stores	Macy's Inc	M	Net Income Margin	4.82	5.32	5.43	3.96	2
2	Department Stores	Nordstrom Inc	JWN	Net Income Margin	6.06	5.85	5.33	4.16	2
3	Department Stores	Dillard's	DDS	Net Income Margin	4.98	4.84	4.89	3.99	2
4	Department Stores	JC Penney Co Inc	JCP	Net Income Margin	-7.59	-10.78	-5.85	-4.06	(
5	Department Stores	Sears Holdings	SHLD	Net Income Margin	-2.33	-3.77	-5.39	-4.49	-
6	Department Stores	DSW Inc.	DSW	Net Income Margin	6.49	6.39	6.14	5.19	4
7	Department Stores	Bon-Ton Stores	BONT	Net Income Margin	-0.72	-0.13	-0.25	-2.05	-
8	Department Stores	Finish Line Inc	FINL	Net Income Margin	4.95	4.60	4.50	1.22	-
9	Department Stores	Dick's Sporting Goods	DKS	Net Income Margin	4.98	5.43	5.05	4.54	3
10	Department Stores	Foot Locker Inc	FL	Net Income Margin	6.42	6.59	7.27	7.30	8
11	Department Stores	Stage Stores Inc	SSI	Net Income Margin	2.32	1.03	1.88	0.24	-
12	Discount Retail	TJX Cos Inc	TJX	Net Income Margin	7.37	7.79	7.62	7.36	6

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015
13	Discount Retail	Ross Stores Inc	ROST	Net Income Margin	8.09	8.18	8.37	8.55
14	Discount Retail	Burlington Store Inc	BURL	Net Income Margin	0.61	0.36	1.36	2.93
15	Discount Retail	Stein Mart	SMRT	Net Income Margin	2.03	2.02	2.04	1.74
16	Discount Retail	Tuesday Morning	TUES	Net Income Margin	-4.19	-2.62	0.18	0.62
17	Brand Holding Companies	Nike Inc	NKE	Net Income Margin	9.52	9.81	10.27	11.84
18	Brand Holding Companies	VF Corp	VFC	Net Income Margin	9.98	10.60	8.82	10.24
19	Brand Holding Companies	L Brands	LB	Net Income Margin	7.20	8.38	9.10	10.31
20	Brand Holding Companies	Tapestry Inc	TPR	Net Income Margin	21.31	19.59	12.65	8.64
21	Brand Holding Companies	Hanesbrands Inc	HBI	Net Income Margin	3.64	7.14	7.60	7.48
22	Brand Holding Companies	Ralph Lauren Corp	RL	Net Income Margin	10.36	10.38	9.62	6.46
23	Brand Holding Companies	Michael Kors Holding Ltd	KORS	Net Income Margin	17.31	20.11	20.42	18.39
24	Brand Holding Companies	Under Armour Inc	UA	Net Income Margin	7.02	6.96	6.75	5.87
25	Brand Holding Companies	Abercrombie & Fitch Co.	ANF	Net Income Margin	5.25	1.33	1.38	1.01

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015
26	Brand Holding Companies	American Eagle Outfitters	AEO	Net Income Margin	6.68	2.51	2.45	6.19
27	Brand Holding Companies	Ascena Retail Group	ASNA	Net Income Margin	3.45	3.03	2.31	-5.82
28	Brand Holding Companies	Gap Inc	GPS	Net Income Margin	7.25	7.93	7.68	5.82
29	Brand Holding Companies	Carter's, Inc	CRI	Net Income Margin	6.77	6.08	6.73	7.89
...	...	...	...	...	...	...	...	...
130	Department Stores	Foot Locker Inc	FL	Total Square Footage	12316.00	12705.00	12734.00	12918.00
131	Department Stores	Stage Stores Inc	SSI	Total Square Footage	15633.00	15799.00	15409.00	15130.00
132	Discount Retail	TJX Cos Inc	TJX	Total Square Footage	69984.00	73209.00	76537.00	80480.00
133	Discount Retail	Ross Stores Inc	ROST	Total Square Footage	27800.00	28900.00	30400.00	31900.00
134	Discount Retail	Burlington Store Inc	BURL	Total Square Footage	0.00	41680.00	42276.00	43659.00
135	Discount Retail	Stein Mart	SMRT	Total Square Footage	9205.00	9240.00	8640.00	8896.00
136	Discount Retail	Tuesday Morning	TUES	Total Square Footage	8641.00	8593.00	8341.00	8326.00
137	Brand Holding Companies	Nike Inc	NKE	Total Square Footage	8495.00	9896.00	10907.00	12326.00
138	Brand Holding Companies	VF Corp	VFC	Total Square Footage	5559.00	5942.00	6960.00	7660.00

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015
139	Brand Holding Companies	L Brands	LB	Total Square Footage	13561.00	13961.00	14418.00	14878.00
140	Brand Holding Companies	Tapestry Inc	TPR	Total Square Footage	2414.00	2703.00	2962.00	3040.00
141	Brand Holding Companies	Hanesbrands Inc.	HBI	Total Square Footage	1125.00	1286.00	1196.00	1206.00
142	Brand Holding Companies	Ralph Lauren Corp	RL	Total Square Footage	2800.00	3000.00	3000.00	3600.00
143	Brand Holding Companies	Michael Kors Holding Ltd	KORS	Total Square Footage	702.00	944.00	1376.00	1777.00
144	Brand Holding Companies	Under Armour Inc	UA	Total Square Footage	545.00	683.00	835.00	1246.00
145	Brand Holding Companies	Abercrombie & Fitch Co.	ANF	Total Square Footage	7958.00	7736.00	7517.00	7292.00
146	Brand Holding Companies	American Eagle Outfitters	AEO	Total Square Footage	4963.00	5206.00	5295.00	5285.00
147	Brand Holding Companies	Ascena Retail Group	ASNA	Total Square Footage	20800.00	21000.00	21200.00	21200.00
148	Brand Holding Companies	Gap Inc	GPS	Total Square Footage	36900.00	37200.00	38100.00	37900.00
149	Brand Holding Companies	Carter's, Inc	CRI	Total Square Footage	3082.00	3452.00	3834.00	4327.00
150	Brand Holding Companies	Chico's FAS, Inc	CHS	Total Square Footage	3271.00	3547.00	3706.00	3652.00
151	Brand Holding Companies	Genesco Inc	GCO	Total Square Footage	4038.00	4341.00	4687.00	4844.00

	Category	Company Name	Ticker	Measure	2012	2013	2014	2015
152	Brand Holding Companies	Fossil Group Inc	FOSL	Total Square Footage	791.00	919.00	981.00	1087.00
153	Brand Holding Companies	Caleres Inc	CAL	Total Square Footage	7551.00	7378.00	7260.00	7243.00
154	Brand Holding Companies	Lululemon Athletica Inc	LULU	Total Square Footage	596.00	740.00	894.00	1071.00
155	Brand Holding Companies	Guess?, Inc	GES	Total Square Footage	2371.00	2329.00	2301.00	2211.00
156	Brand Holding Companies	Express, Inc	EXPR	Total Square Footage	5423.00	5498.00	5529.00	5640.00
157	Brand Holding Companies	Tailored Brands	TLRD	Total Square Footage	7108.00	7116.00	10113.00	10018.00
158	Brand Holding Companies	Steve Madden, Ltd	SHOO	Total Square Footage	188.00	211.00	294.00	315.00
159	Brand Holding Companies	Urban Outfitters, Inc	URBN	Total Square Footage	3409.00	3595.00	3833.00	3953.00

160 rows × 10 columns

Again we will select out the data that we want to merge and adjust for the cells that don't have data

```
In [44]: ni_delta = combo_ni[combo_ni['Measure'] == 'Net Income Margin'] # select
         ing Net Income Margin
         ni_delta.at[14, 'Delta'] = ni_delta.at[14, '2016']/ni_delta.at[14, '201
         4'] - 1 #correct for Burlington
         ni_delta.at[15, 'Delta'] = ni_delta.at[15, '2016']/ni_delta.at[15, '201
         3'] - 1 #correct for Stein Mart
         ni_delta.at[23, 'Delta'] = ni_delta.at[23, '2016']/ni_delta.at[23, '201
         4'] - 1 #correct for MK

         sga_delta = combo_ni[combo_ni['Measure'] == 'SG&A Margin'] # selecting S
         G&A
         sga_delta.at[54, 'Delta'] = sga_delta.at[54, '2016']/sga_delta.at[54, '2
         014'] - 1 # correct for Burlington
         sga_delta.at[55, 'Delta'] = sga_delta.at[55, '2016']/sga_delta.at[55, '2
         013'] - 1 #correct for Stein Mart
         sga_delta.at[63, 'Delta'] = sga_delta.at[63, '2016']/sga_delta.at[63, '2
         014'] - 1 #correct for MK

         ec_delta = combo_ni[combo_ni['Measure'] == 'Ecommerce Sales Margin'] # s
         electing E-commerce
         ec_delta.at[94, 'Delta'] = ec_delta.at[94, '2016']/ec_delta.at[94, '201
         4'] - 1 #correct for Burlington
         ec_delta.at[95, 'Delta'] = ec_delta.at[95, '2016']/ec_delta.at[95, '201
         3'] - 1  # correct for Stein Mart
         ec_delta.at[103, 'Delta'] = ec_delta.at[103, '2016']/ec_delta.at[103, '2
         014'] - 1 #correct for MK
         ec_delta = ec_delta.fillna(0)  # change discount retailers with 0 ecomme
         rce sales from NaN to 0

         sf_delta = combo_ni[combo_ni['Measure'] == 'Total Square Footage']
         sf_delta.at[134, 'Delta'] = sf_delta.at[134, '2016']/sf_delta.at[134, '2
         014'] - 1 #correct for Burlington
         sf_delta.at[135, 'Delta'] = sf_delta.at[135, '2016']/sf_delta.at[135, '2
         013'] - 1  # correct for Stein Mart
         sf_delta.at[143, 'Delta'] = sf_delta.at[143, '2016']/sf_delta.at[143, '2
         014'] - 1 #correct for MK
```

```
/Users/Crystal/anaconda/lib/python3.6/site-packages/ipykernel_launcher.
py:12: RuntimeWarning: invalid value encountered in double_scalars
  if sys.path[0] == '':
```

In [45]:
```
#Create a shortcut for the columns that we want to take out from the dat
aframes,
# that we will then use to merge to form a new dataframe

col_list = ['Ticker', 'Delta']

a_delta2 = pd.merge(ni_delta[['Category', 'Company Name', 'Ticker', 'Del
ta']], sga_delta[col_list],
                  how='left',
                  on='Ticker')
b_delta2 = pd.merge(a_delta2, ec_delta[col_list],
                   how='left',
                   on='Ticker')
finaldelta2 = pd.merge(b_delta2, sf_delta[col_list],
                   how='left',
                   on='Ticker')

finaldelta2.columns = ['Category',
                       'Company_Name',
                       'Ticker',
                       'Net_Income_Margin',
                       'SGA_Margin',
                       'Ecommerce_Margin',
                       'Square_Footage']

finaldelta2.head()
```

Out[45]:

	Category	Company_Name	Ticker	Net_Income_Margin	SGA_Margin	Ecommerce_M
0	Department Stores	Kohl's Corp	KSS	-0.416830	0.072300	1.050734
1	Department Stores	Macy's Inc	M	-0.502075	0.046345	0.847936
2	Department Stores	Nordstrom Inc	JWN	-0.603960	0.056740	0.758871
3	Department Stores	Dillard's	DDS	-0.469880	0.042003	0.789655
4	Department Stores	JC Penney Co Inc	JCP	-1.001318	-0.187320	0.571247

Regressing Net Income Margin for All Sectors

In [46]:
```
nim_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Ecommerce_Margin
 + Square_Footage",
                  data=finaldelta2).fit()

print(nim_results.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:
  0.062
Model:                            OLS   Adj. R-squared:
-0.016
Method:                 Least Squares   F-statistic:
0.7991
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
  0.503
Time:                        23:47:03   Log-Likelihood:
-54.637
No. Observations:                  40   AIC:
  117.3
Df Residuals:                      36   BIC:
  124.0
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
=============
                       coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------------
-------------
Intercept            0.0866      0.239      0.363      0.719      -0.39
7       0.570
SGA_Margin           0.3393      1.534      0.221      0.826      -2.77
3       3.451
Ecommerce_Margin    -0.2347      0.167     -1.408      0.168      -0.57
3       0.103
Square_Footage      -0.2378      0.419     -0.567      0.574      -1.08
7       0.612
==============================================================================
=======
Omnibus:                       18.339   Durbin-Watson:
  2.538
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
24.670
Skew:                           1.386   Prob(JB):
4.40e-06
Kurtosis:                       5.668   Cond. No.
   14.7
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

Okay, that's a pretty low R-squared. Maybe looking at each individual sector will be more helpful. Let's split it into three different dataframes and regress them seperately.

In [47]:
```
ds_ni = finaldelta2[finaldelta2['Category'] == 'Department Stores']
dr_ni = finaldelta2[finaldelta2['Category'] == 'Discount Retail']
bhc_ni = finaldelta2[finaldelta2['Category'] == 'Brand Holding Companie
s']
```

Regressing Net Income Margin for Department Stores

In [48]:
```
#Run regression for department stores
ds_ni_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Ecommerce_Marg
in + Square_Footage",
                    data=ds_ni).fit()

print(ds_ni_results.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:
  0.223
Model:                            OLS   Adj. R-squared:
-0.068
Method:                 Least Squares   F-statistic:
0.7672
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
  0.544
Time:                        23:47:03   Log-Likelihood:
-19.761
No. Observations:                  12   AIC:
  47.52
Df Residuals:                       8   BIC:
  49.46
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
=============
                       coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------------
-------------
Intercept          -0.0680      0.960     -0.071      0.945      -2.28
1       2.145
SGA_Margin          5.9486      5.890      1.010      0.342      -7.63
5      19.532
Ecommerce_Margin   -0.0902      1.006     -0.090      0.931      -2.41
0       2.230
Square_Footage     -2.5943      2.824     -0.919      0.385      -9.10
6       3.917
==============================================================================
=======
Omnibus:                        0.050   Durbin-Watson:
  1.884
Prob(Omnibus):                  0.975   Jarque-Bera (JB):
  0.261
Skew:                           0.093   Prob(JB):
  0.878
Kurtosis:                       2.302   Cond. No.
   18.3
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/scipy/stats/stats.p
y:1334: UserWarning: kurtosistest only valid for n>=20 ... continuing a
nyway, n=12
  "anyway, n=%i" % int(n))
```

Great, much better! Looking at P-value for Ecommerce it seems that it's not that important of a variable. This goes against what we initally expected due to how much online retail and omni-channel integration is stressed these days. Let's run the regression without that variable.

In [49]:

```
ds_ni_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Square_Footag
e", #took out Ecommerce
                  data=ds_ni).fit()

print(ds_ni_results.summary())
```

```
                            OLS Regression Results
=======================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:
  0.223
Model:                            OLS   Adj. R-squared:
  0.050
Method:                 Least Squares   F-statistic:
  1.289
Date:                Thu, 21 Dec 2017   Prob (F-statistic):
  0.322
Time:                        23:47:03   Log-Likelihood:
-19.767
No. Observations:                  12   AIC:
  45.53
Df Residuals:                       9   BIC:
  46.99
Df Model:                           2

Covariance Type:            nonrobust

=======================================================================
===========
                    coef    std err          t      P>|t|      [0.025
     0.975]
-----------------------------------------------------------------------
-----------
Intercept        -0.1434      0.435     -0.330      0.749      -1.127
      0.840
SGA_Margin        5.8051      5.347      1.086      0.306      -6.291
     17.901
Square_Footage   -2.6663      2.554     -1.044      0.324      -8.443
      3.111
=======================================================================
=======
Omnibus:                        0.019   Durbin-Watson:
  1.866
Prob(Omnibus):                  0.991   Jarque-Bera (JB):
  0.234
Skew:                           0.045   Prob(JB):
  0.889
Kurtosis:                       2.321   Cond. No.
   12.8
=======================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/scipy/stats/stats.p
y:1334: UserWarning: kurtosistest only valid for n>=20 ... continuing a
nyway, n=12
  "anyway, n=%i" % int(n))
```

That gives us a similar R-squared, but lowers the p-scores and standard errors slightly. We see a strong and substantive positive relationship between SG&A spending growth and profit margin growth. This is contrary to what we found in the previous section, which showed a negative relationship between SG&A spending growth and EV/Sales growth.

Regressing Net Income Margin for Discount Retail

In [50]:
```
dr_ni_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Ecommerce_Marg
in + Square_Footage",
                   data=dr_ni).fit()

print(dr_ni_results.summary())
```

```
                            OLS Regression Results                            
=======================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:                     
  0.336
Model:                            OLS   Adj. R-squared:                
-1.654
Method:                 Least Squares   F-statistic:                   
0.1690
Date:                Thu, 21 Dec 2017   Prob (F-statistic):            
  0.907
Time:                        23:47:03   Log-Likelihood:                
-6.0408
No. Observations:                   5   AIC:                           
  20.08
Df Residuals:                       1   BIC:                           
  18.52
Df Model:                           3                                  

Covariance Type:            nonrobust                                  

=======================================================================
=============
                      coef    std err          t      P>|t|      [0.02
5      0.975]
-----------------------------------------------------------------------
-------------
Intercept          -0.0350      1.619     -0.022      0.986     -20.61
3      20.543
SGA_Margin        -13.0625     37.627     -0.347      0.787    -491.16
2     465.037
Ecommerce_Margin   -0.1898      0.606     -0.313      0.807      -7.88
5       7.505
Square_Footage      3.2815     13.007      0.252      0.843    -161.98
6     168.549
=======================================================================
=======
Omnibus:                          nan   Durbin-Watson:                 
  2.336
Prob(Omnibus):                    nan   Jarque-Bera (JB):              
  0.554
Skew:                           0.750   Prob(JB):                      
  0.758
Kurtosis:                       2.362   Cond. No.                      
   93.8
=======================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/statsmodels/stats/s
tattools.py:72: ValueWarning: omni_normtest is not valid with less than
8 observations; 5 samples were given.
  "samples were given." % int(n), ValueWarning)
```

Once again, a very high P-value for E-commerce, but this is to be expected since discount retailers usually don't offer online sales. Given the large standard errors, we probably should not rely on this data. Let's try it again without E-Commerce like we did for EV/Sales.

In [51]:
```
dr_ni_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Square_Footag
e",
                  data=dr_ni).fit()

print(dr_ni_results.summary())
```

```
                            OLS Regression Results                            
=======================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:                       
  0.271
Model:                            OLS   Adj. R-squared:                  
-0.457
Method:                 Least Squares   F-statistic:                     
0.3723
Date:                Thu, 21 Dec 2017   Prob (F-statistic):              
  0.729
Time:                        23:47:03   Log-Likelihood:                  
-6.2750
No. Observations:                   5   AIC:                             
  18.55
Df Residuals:                       2   BIC:                             
  17.38
Df Model:                           2                                    

Covariance Type:            nonrobust                                    

=======================================================================
===========
                    coef    std err          t      P>|t|      [0.025
     0.975]
-----------------------------------------------------------------------
-----------
Intercept        -0.3875      0.863     -0.449      0.697      -4.102
      3.327
SGA_Margin      -18.4838     24.761     -0.746      0.533    -125.024
     88.056
Square_Footage    5.7735      7.628      0.757      0.528     -27.046
     38.592
=======================================================================
=======
Omnibus:                          nan   Durbin-Watson:                   
  2.895
Prob(Omnibus):                    nan   Jarque-Bera (JB):                
  0.803
Skew:                           0.930   Prob(JB):                        
  0.669
Kurtosis:                       2.372   Cond. No.                        
   42.0
=======================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.

/Users/Crystal/anaconda/lib/python3.6/site-packages/statsmodels/stats/s
tattools.py:72: ValueWarning: omni_normtest is not valid with less than
8 observations; 5 samples were given.
  "samples were given." % int(n), ValueWarning)
```

Removing the E-commerce variable amplifies the coefficients of the remaining two variables. Increasing SG&A has a particularly acute impact on net income for discount retailers. Additionally, those that increased their square footage benefited, indicating that those that expanded were able to take advantage of the increase in patronage at discount retailers, perhaps also gaining market share. It is important to note, however, that although the standard errors went down, they are still very large.

Regressing Net Income Margin for Brand Holding Companies

In [52]:
```
bhc_ni_results = smf.ols("Net_Income_Margin ~ SGA_Margin + Ecommerce_Mar
gin + Square_Footage",
                    data=bhc_ni).fit()

print(bhc_ni_results.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=======
Dep. Variable:      Net_Income_Margin   R-squared:                      
  0.132
Model:                            OLS   Adj. R-squared:                 
-0.005
Method:                 Least Squares   F-statistic:                    
0.9664
Date:                Thu, 21 Dec 2017   Prob (F-statistic):             
  0.429
Time:                        23:47:03   Log-Likelihood:                 
-19.832
No. Observations:                  23   AIC:                            
  47.66
Df Residuals:                      19   BIC:                            
  52.21
Df Model:                           3                                   

Covariance Type:            nonrobust                                   

==============================================================================
=============
                       coef    std err          t      P>|t|      [0.02
5      0.975]
------------------------------------------------------------------------------
-------------
Intercept            0.0133      0.221      0.060      0.952      -0.44
8       0.475
SGA_Margin          -0.8908      1.111     -0.802      0.433      -3.21
6       1.435
Ecommerce_Margin    -0.2269      0.146     -1.556      0.136      -0.53
2       0.078
Square_Footage       0.0445      0.300      0.148      0.884      -0.58
4       0.673
==============================================================================
=======
Omnibus:                       10.263   Durbin-Watson:                  
  2.628
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               
  8.109
Skew:                           1.300   Prob(JB):                       
0.0173
Kurtosis:                       4.306   Cond. No.                       
   12.9
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is 
correctly specified.
```

The negative relationship between SG&A margin growth and profitability growth aligns with that between SG&A margin growth and EV/sales growth. It is interesting that e-commerce sales growth also has a significant negative relationship with profitability growth. Square footage doesn't seem to be a significant factor, which makes sense because a lot of these companies sell wholesale or have a larger online presence.

Conclusion

After analyzing performance metrics, performance drivers, and the relationships between them for a sample size of forty clothing retailers in the U.S., we have arrived at a few insights and found avenues for further exploration. To begin, the impact of change in e-commerce sales on both multiple expansion and profitability growth was a lot less notable than we had predicted. Regression analyses of the entire set of companies and category-specific analyses demonstrated that change in e-commerce sales margin is a weaker predictor of both multiple expansion and profitability growth than change in SG&A margin. We believe that these findings are a result of two factors. First, e-commerce sales currently do not occupy a large percentage of a firm's revenue generation. Second, e-commerce sales margin may not be the most optimal metric with which to capture a firm's digital capabilities. This is due to the fact that a firm's digital ecosystem designed for customer extends far past merely the execution of transactions, capturing marketing, loyalty programs, and other mediums of interaction.

Finally, we found that department stores exhibited negative correlations between retail square footage change and multiple expansion/profitability growth, while discount retailers and brand holding companies exhibited the opposite relationship. We believe that this outcome is contingent on how much retail space they already occupy (in the case of brand holding companies), whether firms have utilized the retail space effectively and efficiently (in the case of department stores), and whether their business model depends on brick-and-mortar retail (in the case of discount retailers).

Through our analyses, we have observed different characteristics between different segments of the clothing retail universe, not all of which can be captured and demonstrated by pure data analysis. Given more time, we find it pertinent to analyze other performance metrics and performance drivers that could offer alternative perspectives of the current and future state of the retail industry. If given the objective to delve deeper into the impact of ecommerce and digitization on financial performance, we would aim to collect more data on customer loyalty, webpage visits, and percentage of revenue spent on specifically advertising, rather than SG&A as a whole.