How Women are
Conquering the S&P500
VP of Product, Quantopian
As of 2013, women....
12.9% of top management
12.7% of boards
Global performance: companies market cap >USD 10 billion

Source: CSG 3000

- No women on board
- 1 or more women on board
Performance of companies tiered by female management participation

Source: CSG 3000
What if you invested in female CEOs?
# Import my csv and rename some of the columns
CEOs = local_csv('FemaleCEOs_v6.csv')
CEOs.rename(columns={'SID': 'Ticker', 'Start Date': 'start_date', 'End Date': 'end_date'}, inplace=True)

Number of CEOs = 80
Number of Companies = 74

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Company</th>
<th>Ticker</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Patricia Russo</td>
<td>Lucent</td>
<td>ALU</td>
<td>12/1/06</td>
<td>8/1/08</td>
</tr>
<tr>
<td>2</td>
<td>Katherine Krill</td>
<td>AnnTaylor Stores Corporation</td>
<td>ANN</td>
<td>10/1/05</td>
<td>12/31/14</td>
</tr>
<tr>
<td>3</td>
<td>Angela Braly</td>
<td>WellPoint</td>
<td>ANTM</td>
<td>6/1/07</td>
<td>8/1/12</td>
</tr>
<tr>
<td>4</td>
<td>Judy McReynolds</td>
<td>Arkansas Best Corp.</td>
<td>ARCB</td>
<td>1/2/10</td>
<td>12/31/14</td>
</tr>
<tr>
<td>6</td>
<td>Sheri McCoy</td>
<td>Avon Product</td>
<td>AVP</td>
<td>4/23/12</td>
<td>12/31/14</td>
</tr>
<tr>
<td>7</td>
<td>Susan N. Story</td>
<td>American Water Works Company</td>
<td>AWK</td>
<td>5/9/14</td>
<td>12/31/14</td>
</tr>
<tr>
<td>8</td>
<td>Gayla Delly</td>
<td>Benchmark Electronics</td>
<td>BHE</td>
<td>1/3/12</td>
<td>12/31/14</td>
</tr>
<tr>
<td>9</td>
<td>Elizabeth Smith</td>
<td>Bloomin' Brands</td>
<td>BLMN</td>
<td>8/9/12</td>
<td>12/31/14</td>
</tr>
<tr>
<td>10</td>
<td>Diane M. Sullivan</td>
<td>Brown Shoe Company</td>
<td>CAL</td>
<td>5/26/11</td>
<td>12/31/14</td>
</tr>
<tr>
<td>11</td>
<td>Sandra Cochran</td>
<td>Cracker Barrel Old Country Store</td>
<td>CBRL</td>
<td>9/12/11</td>
<td>12/31/14</td>
</tr>
<tr>
<td>12</td>
<td>Linda Massman</td>
<td>Clearwater Paper</td>
<td>CLW</td>
<td>1/2/13</td>
<td>12/31/14</td>
</tr>
<tr>
<td>13</td>
<td>Denise Morrison</td>
<td>Campbell Soup</td>
<td>CPB</td>
<td>8/1/11</td>
<td>12/31/14</td>
</tr>
<tr>
<td>14</td>
<td>Andrea Ayers</td>
<td>Convergys</td>
<td>CVG</td>
<td>10/2/12</td>
<td>12/31/14</td>
</tr>
<tr>
<td>15</td>
<td>Ellen Kullman</td>
<td>DuPont</td>
<td>DD</td>
<td>1/2/09</td>
<td>12/31/14</td>
</tr>
<tr>
<td>16</td>
<td>Sara Mathew</td>
<td>Dun &amp; Bradstreet Inc.</td>
<td>DNB</td>
<td>1/2/10</td>
<td>10/7/13</td>
</tr>
<tr>
<td>17</td>
<td>Lynn Good</td>
<td>Duke Energy</td>
<td>DUK</td>
<td>7/1/13</td>
<td>12/31/14</td>
</tr>
<tr>
<td>18</td>
<td>Margaret Whitman</td>
<td>eBay</td>
<td>EBAY</td>
<td>1/1/98</td>
<td>3/1/08</td>
</tr>
<tr>
<td>19</td>
<td>Mary Agnes Wilderotter</td>
<td>Citizens Communications</td>
<td>FTR</td>
<td>11/1/04</td>
<td>12/31/14</td>
</tr>
</tbody>
</table>
How many new CEOs per year?

In [2]:
CEOs['year_started'] = pd.DatetimeIndex(CEOs['start_date']).year
CEOs['year_ended'] = pd.DatetimeIndex(CEOs['end_date']).year
CEOs['year_started'].value_counts(sort=False).plot(kind='bar')
pyplot.grid(b=None, which='major', axis='both')
pyplot.box(on=None)
Any CEOs start? —-> BUY
Any CEOs end? —-> SELL
Rebalance
Move to next day
How many companies will I hold each year?

```python
In [55]:
    from pandas.tseries.offsets import YearBegin
    CEOs['year Ended'] = pd.DatetimeIndex(CEOs['end date']).year
    CEOs['year Started'] = pd.DatetimeIndex(CEOs['start date']).year

    counts = pd.Series(index=pd.date_range('2002-01-01', '2015-01-01', freq=YearBegin(1)))
    for year in counts.index:
        counts[year] = len(CEOs[(CEOs.start date <= year) & (CEOs.end date >= year)])

    counts.plot(kind='bar')
    pyplot.grid(b=None, which='major', axis='both')
    pyplot.box(on=None)
```
In [10]:

```python
    ... This is where I initialize my algorithm
    ...
    from zipline.api import order
    from zipline.finance.slippage import FixedSlippage

    def initialize(context):
        # load the CEO data and a variable to count the number of stocks
        # held at any time as global variables
        context.CEOs = CEOs
        context.current_stocks = []
        context.stocks_to_order_today = []
        context.stocks_to_sell_today = []
        context.set_slippage(FixedSlippage(spread=0))
```

In [11]:

```python
    ... Handle data is the function that is running every minute (or day) looking to make trades
    ...
    from zipline.api import order

    def handle_data(context, data):
        # Set my order and sell dictionaries to empty at the start of any day.
        context.stocks_to_order_today = []
        context.stocks_to_sell_today = []

        # Get todays date.
        today = get_datetime()

        # Get a dataframe with just the companies where start_date (or end date) is today.
        context.stocks_to_order_today = context.CEOs.SID[context.CEOs.start_date==today].tolist()
        context.stocks_to_sell_today = context.CEOs.SID[context.CEOs.end_date==today].tolist()
        context.stocks_to_sell_today = [s for s in context.stocks_to_sell_today if s!= None]
        context.stocks_to_sell_today = [s for s in context.stocks_to_sell_today if s!= None]

        # If there are stocks that need to be bought or sold today
        if len(context.stocks_to_order_today) > 0 or len(context.stocks_to_sell_today) > 0:
            # First sell any that need to be sold, and remove them from current_stocks.
            for stock in context.stocks_to_sell_today:
                if stock in context.current_stocks:
                    order_target(stock,0)
                    context.current_stocks.remove(stock)
                    #print "Selling %s" % stock

            # Then add any I am buying to current_stocks.
            for stock in context.stocks_to_order_today:
                context.current_stocks.append(stock)

            # Then rebalance the portfolio so I have an equal amount of each stock in current_stocks.
            for stock in context.current_stocks:
                if stock in data:
                    #print "Buying and/or rebalancing %s at target weight %s" % (stock, target_weight)
                    #calculate the value to buy
                    portfolio_value = context.portfolio.portfolio_value
                    num_stocks = len(context.current_stocks)
                    value_to_buy = portfolio_value/num_stocks
                    #print "Buying and/or rebalancing %s at value = %s" % (stock, value_to_buy)
                    order_target_value(stock,value_to_buy)
```
339% My Algorithm
122% SPY
217% Difference
Could it be...

YAHOO!

and...

Alibaba.com
In [47]:
fig = pyplot.figure()
ax2 = fig.add_subplot(212)
start_date = adm_df['start_date']
end_date = adm_df['end_date']
data[security].plot(ax=ax2, figsize=(15, 18), color='g')
ax2.plot(start_date, data.ix[start_date][security], '^', markersize=20, color='b', linestyle='')
ax2.plot(end_date, data.ix[end_date][security], 'v', markersize=20, color='b', linestyle='')
pyplot.ylabel('% Cumulative Return', fontsize=20)
pyplot.title('Cumulative Return', fontsize=20)
pyplot.grid(b='None', which='major', axis='both')
pyplot.box(on='None')
pyplot.legend([ 'Yahoo'], frameon=False, loc='best')

print adm_df['CEO']
My Algorithm: 320%
SPY: 122%
Difference: 197%
What about sector bias?
275% My Algorithm 122% SPY 153% Difference
SPY is the wrong benchmark
339%  
My Algorithm  

296%  
Quanto1000  

122%  
SPY
Quantopian Store
Institutional quality data feeds, available natively on Quantopian.

- Estimize Revisions: $10/mo from Estimize
- Accern Alpha One: $50/mo from Accern
- 13-D Filings: $50/mo from EventVestor
- Buyback Authorizations: $85/mo from EventVestor
- CEO Changes: $50/mo from EventVestor
- Clinical Trials: $50/mo from EventVestor
Incoming CEOs 2007-2015

Men

Women
28% Female CEOs
44% Male CEOs
Female CEOs: 68%
Male CEOs: 47%
68% Female CEOs
68% SPY
47% Male CEOs
187% F1000 Female CEOs
68% Female CEOs
68% SPY
47% Male CEOs
karen@quantopian.com

@karenrubin
Could it be...

...the outliers.
In [6]:
def CEO_returns(row):
    Cstart_date = row['start_date']
    Cend_date = row['end_date']
    Csid = row['SID']

    # get the price on the start date and end date
    start_price = get_pricing(Csid, Cstart_date, Cstart_date, fields='close_price')
    end_price = get_pricing(Csid, Cend_date, Cend_date, fields='close_price')

    # calculate %change and add as new column
    row['Pct_Change'] = ((end_price.iloc[0]) - (start_price.iloc[0])) / start_price.iloc[0] * 100
    return row

CEOs = CEOs.apply(CEO_returns, axis=1)
CEOs.sort(columns='Pct_Change', ascending=False).head(3)

Out[6]:

<table>
<thead>
<tr>
<th>CEO</th>
<th>Company Name</th>
<th>Ticker</th>
<th>start_date</th>
<th>end_date</th>
<th>SID</th>
<th>Pct_Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 Mindy F. Grossman</td>
<td>HSN</td>
<td>HSNI</td>
<td>2008-08-20</td>
<td>2014-12-31</td>
<td>Equity(36733 [HSNI])</td>
<td>506.060606</td>
</tr>
<tr>
<td>65 Debra Cafaro</td>
<td>Ventas</td>
<td>VTR</td>
<td>2002-01-02</td>
<td>2014-12-31</td>
<td>Equity(18821 [VTR])</td>
<td>505.642361</td>
</tr>
<tr>
<td>61 Carol Meyrowitz</td>
<td>TJX Corp</td>
<td>TJX</td>
<td>2007-01-29</td>
<td>2014-12-31</td>
<td>Equity(7457 [TJX])</td>
<td>366.530612</td>
</tr>
</tbody>
</table>

In [7]:
CEOs.sort(columns='Pct_Change', ascending=True).head(3)

Out[7]:

<table>
<thead>
<tr>
<th>CEO</th>
<th>Company Name</th>
<th>Ticker</th>
<th>start_date</th>
<th>end_date</th>
<th>SID</th>
<th>Pct_Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>41 Janet L. Robinson</td>
<td>The New York Times Company</td>
<td>NYT</td>
<td>2004-12-27</td>
<td>2012-01-03</td>
<td>Equity(5551 [NYT])</td>
<td>-80.866782</td>
</tr>
<tr>
<td>50 Mary Sammons</td>
<td>Rite Aid Corp</td>
<td>RAD</td>
<td>2003-06-25</td>
<td>2010-06-23</td>
<td>Equity(6330 [RAD])</td>
<td>-75.348837</td>
</tr>
<tr>
<td>49 Pamela Kirby</td>
<td>Quintiles Transnational</td>
<td>Q</td>
<td>2002-01-02</td>
<td>2003-09-25</td>
<td>Equity(17104 [Q])</td>
<td>-74.715100</td>
</tr>
</tbody>
</table>
267%  
My Algorithm  
122%  
SPY  
145%  
Difference
Female CEOs: 68%
Male CEOs: 47%

F1000: 187%
Q500: 130%
Female CEOs: 68%
SPY: 68%
Male CEOs: 47%
339%  My Algorithm
305%  Q500
211%  Q&P500
122%  SPY