In [72]:

```
# import requisite libraries
# make sure that mongodb alchemyapi is installed and running
from alchemyapi import AlchemyAPI
alchemyapi = AlchemyAPI()
from pymongo import MongoClient
from bs4 import BeautifulSoup
import os
import pymongo
import requests
import os
import urllib
import urllib2
import pandas as pd
import matplotlib as plt
import matplotlib as mpl
import scipy as sp
import numpy as np
from pprint import pprint
import pylab as p
from scipy import stats
import pandas.io.data
import datetime
from pandas import Series, DataFrame
import statsmodels.formula.api as smf
```

In [5]:

```
# function to obtain raw data from the specified site
def get_data(web_url):
    request = urllib2.Request(web_url)
    page = urllib2.urlopen(request)
    content = page.read()
    page.close()
    return content
```

In [90]:

```
# check that the get_data function works
# print get_data('http://www.bloomberg.com/news/stocks')
```

In [7]:

```
# function to parse the raw content to return the desired variables using beautif
ul soup
def parse_content(content):
    data = BeautifulSoup(content)
    headlines = data.findAll('a',{'data-type':'Story'})
    return headlines
```

In [8]:

```
# check that the parse_content function works
print parse_content(get_data('http://www.bloomberg.com/news/stocks'))
```

[Asian Stocks Advance Third Day as Materials Shares Lead Gains, China Stocks Rally to Four-Year High as Banks, PetroChina Surge, Canadian Stocks Cap Best Week Since 2009 on Energy Rally, Ibovespa Extends Weekly Advance as Petrobras Rises With Crude, U.K. Stocks Rise Fourth Day for Biggest Weekly Rally Since 2011, Saudi Arabia Says Hard for OPEC to Give Up Market Share, Russian Stocks Decline as Bank Shares Drop on Earnings Concern, Japan Stocks Swing as Oil Shares Gain, Airlines Drop, Next Prada Eludes Hong Kong as Foreign Companies Shun Listings, Haitong Securities to Raise $3.9 Billion Selling H-Shares, NQ Mobile Tumbles as Unit Sale, Earnings Fail to Impress, Dubai Shares Enter a Bull Market as Oil Rise Fuels Gulf Rally, Asian Stocks Rise as Crude Extends Gains While Kiwi Drops, January Is the Darkest Month for India Stocks: Chart of the Day, Citic Bigger Than Credit Suisse, Nearing UBS: Chart of the Day]

In [9]:

```
# store parsed content into a database so that we can extend and query the data w
ith ease
def store_data(content):
    headlines = parse_content(content)
    client = MongoClient(host = 'localhost', port = 27017)
    db = client.stocks_headlinesDB2
    collection = db.bloomberg

    for headline in headlines:
        if collection.find_one({'number':headline['data-id']})==None:
            temp = {'date':str(headline['href']).split('/')[2],
            'tags':[],
            'news':headline.string,
            'number':headline['data-id']}
            collection.insert(temp)

            print(headline.string)


        else:
            print("No New Headlines Added . . .")

    print "\nCurrent Contents in ", db, ": \n"

    return collection
```

In [10]:

```
def read_sentiment(text_arr):

    scores = []


    for i in text_arr:
        sentiment_reading = alchemyapi.sentiment("text", i)
        score = sentiment_reading["docSentiment"]
        scores.append(score)
        # sentiment_scores.append(sentiment_reading)


    return scores
```

In [11]:

```
if __name__ == '__main__':

    # OBTAIN & SCRUB DATA

    # target site is bloomberg
    url = "http://www.bloomberg.com/news/stocks/"
```

```
# get raw data and save it to a variable
extracted_data = get_data(url)

# parse the raw data
parse_content(extracted_data)

# save the parsed content to a mongo database for further extensibility
coll = store_data(extracted_data)

# create variable arrays
headlines = []
split_headlines = []
dates = []
sentiment_scores = []


# check to see if data was loaded properly into the data base
for doc in coll.find():
    print doc, "\n"
    headlines.append(doc['news'])
    dates.append(doc['date'])

# check to see if headlines variable was populated properly
for h in headlines:
    split_headlines.append(h.split(','))
    print h, "\n"


    # check to see if dates variable was populated properly
    # this is important so we can match up the market data we need for the dates.
    for d in dates:
        print d, "\n"

    scores = read_sentiment(headlines)

    # EXPLORE DATA

    # convert to numpy arrays



    # parse the sentiment scores


    # here we'll use pandas and matplotlib to read the data into dataframes and d
o some exploration.
```

```
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
```

```
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .
No New Headlines Added . . .

Current Contents in  Database(MongoClient('localhost', 27017), u'stocks_headlines
DB2') :

{u'date': u'2014-12-19', u'news': u'European Stocks Climb, Posting Biggest Four-D
ay Jump Since 2012', u'_id': ObjectId('5495d2228080bc6757539441'), u'number': u'N
GTIK06VDKHS01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Canadian Stocks Cap Best Week Since 2009 on E
nergy Rally', u'_id': ObjectId('5495d2228080bc6757539442'), u'number': u'NGU5O76T
TDS301', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Ibovespa Extends Weekly Advance as Petrobras
Rises With Crude', u'_id': ObjectId('5495d2228080bc6757539443'), u'number': u'NGT
YAV6TTDSB01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'U.K. Stocks Rise Fourth Day for Biggest Weekl
y Rally Since 2011', u'_id': ObjectId('5495d2228080bc6757539444'), u'number': u'N
GTLJH6K50XU01', u'tags': []}

{u'date': u'2014-12-18', u'news': u'Saudi Arabia Says Hard for OPEC to Give Up Ma
rket Share', u'_id': ObjectId('5495d2228080bc6757539445'), u'number': u'NGRWGR6JT
SE801', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Russian Stocks Decline as Bank Shares Drop on
 Earnings Concern', u'_id': ObjectId('5495d2228080bc6757539446'), u'number': u'NG
TQC16TTDSB01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Asian Stocks Rise Most in Two Months Amid Glo
bal Equities Rally', u'_id': ObjectId('5495d2228080bc6757539447'), u'number': u'N
GSXBW6K50YF01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Emerging Stocks Post Weekly Gain as Samsung S
urges; Ruble Gains', u'_id': ObjectId('5495d2228080bc6757539448'), u'number': u'N
GTBY56K50Y301', u'tags': []}

{u'date': u'2014-12-18', u'news': u'Alibaba Investors Aren\u2019t Fazed as IPO Lo
ckup Expiration Looms', u'_id': ObjectId('5495d2228080bc6757539449'), u'number':
u'NGP5KC6TTDST01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Japan Stocks Rise a Second Day as Fed Sparks
Global Rally', u'_id': ObjectId('5495d2228080bc675753944a'), u'number': u'NGSU086
```

K50Y001', u'tags': []}

{u'date': u'2014-12-19', u'news': u'China\u2019s Stocks Rise to Four-Year High as
 Chalco Rallies With BYD', u'_id': ObjectId('5495d2228080bc675753944b'), u'number
': u'NGT1BM6K50Y001', u'tags': []}

{u'date': u'2014-12-19', u'news': u'Short-Seller Block Bluffs His Way Onto NQ Mob
ile\u2019s Call', u'_id': ObjectId('5495d2228080bc675753944c'), u'number': u'NGUE
YH6VDKI201', u'tags': []}

{u'date': u'2014-12-19', u'news': u'U.S. Stocks Cap Best Week in Two Months on Fe
d-Led Rally', u'_id': ObjectId('5495d2228080bc675753944d'), u'number': u'NGU8VG6T
TDSM01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'S&P 500 Erases Monthly Loss on Biggest Rally
Since 2011', u'_id': ObjectId('5495d2228080bc675753944e'), u'number': u'NGTM9B6JI
JUO01', u'tags': []}

{u'date': u'2014-12-19', u'news': u'A Humbled Janus Capital Is Betting on Bill Gr
oss for Its Turnaround', u'_id': ObjectId('5495d2228080bc675753944f'), u'number':
 u'NGCAM66JIJWI01', u'tags': []}

{u'date': u'2014-12-21', u'news': u'Dubai Shares Enter a Bull Market as Oil Rise
Fuels Gulf Rally', u'_id': ObjectId('54972cbe8080bc034ab2e659'), u'number': u'NGX
7G76TTDS201', u'tags': []}

{u'date': u'2014-12-21', u'news': u'NQ Mobile Tumbles as Unit Sale, Earnings Fail
 to Impress', u'_id': ObjectId('54972cbe8080bc034ab2e65a'), u'number': u'NGUKW66V
DKIY01', u'tags': []}

{u'date': u'2014-12-21', u'news': u'Haitong Securities to Raise $3.9 Billion Sell
ing H-Shares', u'_id': ObjectId('5497385b8080bc034ab2e65b'), u'number': u'NGXNAC6
TTDS001', u'tags': []}

{u'date': u'2014-12-21', u'news': u'Next Prada Eludes Hong Kong as Foreign Compan
ies Shun Listings', u'_id': ObjectId('5497385b8080bc034ab2e65c'), u'number': u'NF
DW536K50Y201', u'tags': []}

{u'date': u'2014-12-21', u'news': u'Asian Futures Signal Gains on Oil Jump; Dolla
r Holds Gain', u'_id': ObjectId('549751f08080bc0bf8f2deaf'), u'number': u'NGYEDQ6
JTSE901', u'tags': []}

{u'date': u'2014-12-22', u'news': u'Asian Stocks Advance Third Day as Materials S
hares Lead Gains', u'_id': ObjectId('549797988080bc1aa8a41802'), u'number': u'NGY
HF16S972F01', u'tags': []}

{u'date': u'2014-12-22', u'news': u'China Stocks Rally to Four-Year High as Banks
, PetroChina Surge', u'_id': ObjectId('549797988080bc1aa8a41803'), u'number': u'N
GYGKW6JTSEC01', u'tags': []}

{u'date': u'2014-12-22', u'news': u'Japan Stocks Swing as Oil Shares Gain, Airlin
es Drop', u'_id': ObjectId('549797988080bc1aa8a41804'), u'number': u'NGYHBS6JIJUQ
01', u'tags': []}

{u'date': u'2014-12-21', u'news': u'January Is the Darkest Month for India Stocks : Chart of the Day', u'_id': ObjectId('549797988080bc1aa8a41805'), u'number': u'NGBDXF6TTDSR01', u'tags': []}

{u'date': u'2014-12-21', u'news': u'Citic Bigger Than Credit Suisse, Nearing UBS: Chart of the Day', u'_id': ObjectId('549797988080bc1aa8a41806'), u'number': u'NGRDUN6TTDT101', u'tags': []}

European Stocks Climb, Posting Biggest Four-Day Jump Since 2012

Canadian Stocks Cap Best Week Since 2009 on Energy Rally

Ibovespa Extends Weekly Advance as Petrobras Rises With Crude

U.K. Stocks Rise Fourth Day for Biggest Weekly Rally Since 2011

Saudi Arabia Says Hard for OPEC to Give Up Market Share

Russian Stocks Decline as Bank Shares Drop on Earnings Concern

Asian Stocks Rise Most in Two Months Amid Global Equities Rally

Emerging Stocks Post Weekly Gain as Samsung Surges; Ruble Gains

Alibaba Investors Aren’t Fazed as IPO Lockup Expiration Looms

Japan Stocks Rise a Second Day as Fed Sparks Global Rally

China’s Stocks Rise to Four-Year High as Chalco Rallies With BYD

Short-Seller Block Bluffs His Way Onto NQ Mobile’s Call

U.S. Stocks Cap Best Week in Two Months on Fed-Led Rally

S&P 500 Erases Monthly Loss on Biggest Rally Since 2011

A Humbled Janus Capital Is Betting on Bill Gross for Its Turnaround

Dubai Shares Enter a Bull Market as Oil Rise Fuels Gulf Rally

NQ Mobile Tumbles as Unit Sale, Earnings Fail to Impress

Haitong Securities to Raise $3.9 Billion Selling H-Shares

Next Prada Eludes Hong Kong as Foreign Companies Shun Listings

Asian Futures Signal Gains on Oil Jump; Dollar Holds Gain

Asian Stocks Advance Third Day as Materials Shares Lead Gains

China Stocks Rally to Four-Year High as Banks, PetroChina Surge

Japan Stocks Swing as Oil Shares Gain, Airlines Drop

January Is the Darkest Month for India Stocks: Chart of the Day

Citic Bigger Than Credit Suisse, Nearing UBS: Chart of the Day

2014-12-19

2014-12-19

2014-12-19

2014-12-19

2014-12-18

2014-12-19

2014-12-19

2014-12-19

2014-12-18

2014-12-19

2014-12-19

2014-12-19

2014-12-19

2014-12-19

2014-12-19

2014-12-19

2014-12-21

2014-12-21

2014-12-21

2014-12-21

2014-12-21

2014-12-22

2014-12-22

2014-12-22

2014-12-21

2014-12-21

In [12]:

```
positive_scores = []
negative_scores = []
neutral_scores = []
for k in scores:
    if k['type'] == 'positive':
        positive_scores.append(float(k['score']))
    elif k['type'] == 'negative':
        negative_scores.append(float(k['score']))
    else:
        neutral_scores.append(float('0.0'))


ranked_sentiment_scores = np.array(positive_scores+negative_scores+neutral_scores
)
print np.sort(ranked_sentiment_scores)
```

```
[-0.834683 -0.792255 -0.759776 -0.698389 -0.667045 -0.618499 -0.580298
 -0.530089 -0.425016 -0.362915 -0.344176 -0.323923 -0.245084 -0.229828  0.
  0.         0.         0.         0.         0.         0.200483  0.288021
  0.378668  0.394825  0.443122]
```

In [13]:

```
for i in split_headlines:
    print i
```

```
[u'European Stocks Climb', u' Posting Biggest Four-Day Jump Since 2012']
[u'Canadian Stocks Cap Best Week Since 2009 on Energy Rally']
[u'Ibovespa Extends Weekly Advance as Petrobras Rises With Crude']
[u'U.K. Stocks Rise Fourth Day for Biggest Weekly Rally Since 2011']
[u'Saudi Arabia Says Hard for OPEC to Give Up Market Share']
[u'Russian Stocks Decline as Bank Shares Drop on Earnings Concern']
[u'Asian Stocks Rise Most in Two Months Amid Global Equities Rally']
[u'Emerging Stocks Post Weekly Gain as Samsung Surges; Ruble Gains']
[u'Alibaba Investors Aren\u2019t Fazed as IPO Lockup Expiration Looms']
[u'Japan Stocks Rise a Second Day as Fed Sparks Global Rally']
[u'China\u2019s Stocks Rise to Four-Year High as Chalco Rallies With BYD']
[u'Short-Seller Block Bluffs His Way Onto NQ Mobile\u2019s Call']
[u'U.S. Stocks Cap Best Week in Two Months on Fed-Led Rally']
[u'S&P 500 Erases Monthly Loss on Biggest Rally Since 2011']
[u'A Humbled Janus Capital Is Betting on Bill Gross for Its Turnaround']
[u'Dubai Shares Enter a Bull Market as Oil Rise Fuels Gulf Rally']
[u'NQ Mobile Tumbles as Unit Sale', u' Earnings Fail to Impress']
[u'Haitong Securities to Raise $3.9 Billion Selling H-Shares']
[u'Next Prada Eludes Hong Kong as Foreign Companies Shun Listings']
[u'Asian Futures Signal Gains on Oil Jump; Dollar Holds Gain']
[u'Asian Stocks Advance Third Day as Materials Shares Lead Gains']
[u'China Stocks Rally to Four-Year High as Banks', u' PetroChina Surge']
[u'Japan Stocks Swing as Oil Shares Gain', u' Airlines Drop']
[u'January Is the Darkest Month for India Stocks: Chart of the Day']
[u'Citic Bigger Than Credit Suisse', u' Nearing UBS: Chart of the Day']
```

In [14]:

```
for d in dates:
    print d
```

```
2014-12-19
2014-12-19
2014-12-19
2014-12-19
2014-12-18
2014-12-19
2014-12-19
2014-12-19
2014-12-18
2014-12-19
2014-12-19
2014-12-19
2014-12-19
2014-12-19
2014-12-19
2014-12-21
2014-12-21
2014-12-21
2014-12-21
2014-12-21
2014-12-22
2014-12-22
2014-12-21
2014-12-21
```

In [15]:

```
# descriptive statistics of the sentiment scores
mu = np.mean(ranked_sentiment_scores)
print "mean of sentiment score distribution: ",  mu
```

```
mean of sentiment score distribution:  -0.22827428
```

In [16]:

```
sigma = np.std(ranked_sentiment_scores)
print "standard deviation of sentiment score distribution: ", sigma
```

```
standard deviation of sentiment score distribution:  0.389834376345
```

In [17]:

```
min = np.min(ranked_sentiment_scores)
print "min of sentiment score distribution: ", min
```

```
min of sentiment score distribution:  -0.834683
```

In [18]:

```
max = np.max(ranked_sentiment_scores)
print "max of sentiment score distribution: ", max
```

```
max of sentiment score distribution:  0.443122
```

In [19]:

```
median = np.median(ranked_sentiment_scores)
print "median of sentiment score distribution: ", median
```

```
median of sentiment score distribution:  -0.245084
```

In [20]:

```
pos_count = np.bincount(positive_scores)
print "number of positive scores: ", int(pos_count)
```

```
number of positive scores:  5
```

In [21]:

```
neg_count = np.bincount(negative_scores)
print "number of negative scores: ", int(neg_count)
```

```
number of negative scores:  14
```

In [22]:

```
neutral_count = np.bincount(neutral_scores)
print "number of neutral scores: ", int(neutral_count)
```

```
number of neutral scores:  6
```

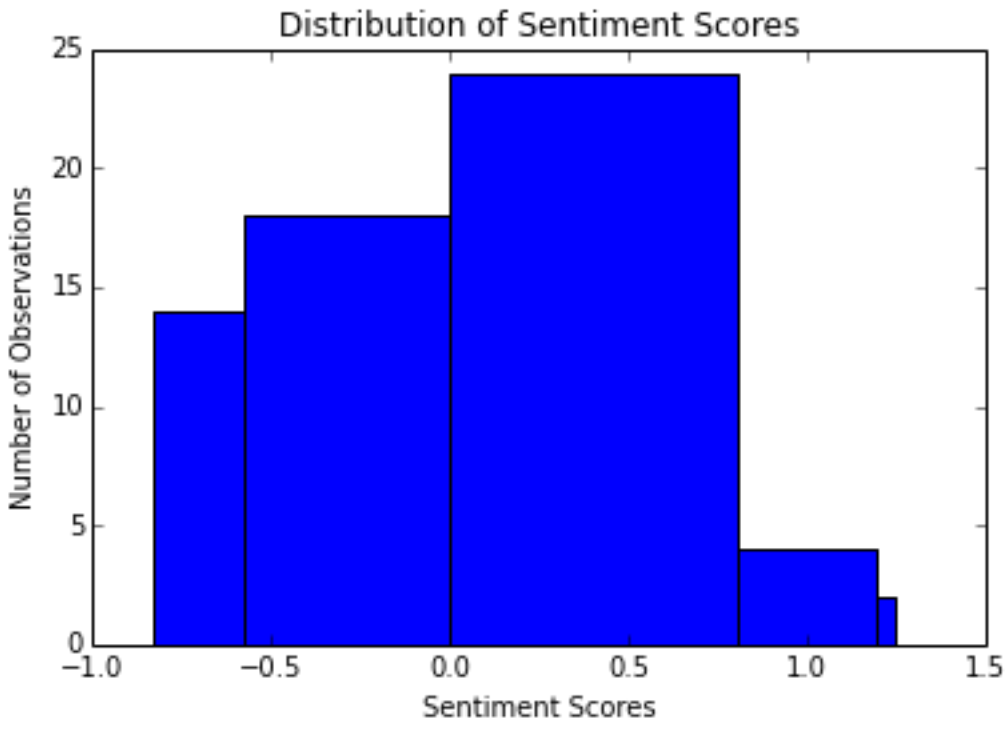
In [23]:

```
total_observations = pos_count+neg_count+neutral_count
print "total number of observations: ", int(total_observations)
```

```
total number of observations:  25
```

In [24]:

```
# visualizations of exploratory data analysis
ind = np.arange(len(positive_scores+negative_scores+neutral_scores))
p.bar((positive_scores+negative_scores+neutral_scores),ind)
p.xlabel('Sentiment Scores')
p.ylabel('Number of Observations')
p.title('Distribution of Sentiment Scores')
p.show()
```



In [30]:

```
# get spy data over date ranges of articles collected from bloomberg.  i will exa
mine stock market behavior
# in the 20 days leading up to the last article post using a rolling average of 3
 days.
spy = pd.io.data.get_data_yahoo('SPY', start=datetime.datetime(2014, 11, 11), end
=datetime.datetime(2014,12,21))
```

In [31]:

```
spy
```

Out[31]:

	Open	High	Low	Close	Volume	Adj Close
Date						
2014-11-11	204.06	204.31	203.65	204.18	54499000	203.06
2014-11-12	203.35	204.24	203.31	203.96	90120000	202.84
2014-11-13	204.16	204.83	203.21	204.19	85358000	203.07
2014-11-14	204.10	204.49	203.72	204.24	80418000	203.12
2014-11-17	203.85	204.58	203.65	204.37	80441000	203.25
2014-11-18	204.44	205.92	204.44	205.55	76068000	204.42
2014-11-19	205.31	205.55	204.30	205.22	82373000	204.09
2014-11-20	204.26	205.71	204.18	205.58	72840000	204.45
2014-11-21	207.64	207.84	205.98	206.68	142327000	205.55
2014-11-24	207.17	207.39	206.91	207.26	65881000	206.12
2014-11-25	207.54	207.79	206.80	207.11	79108000	205.97
2014-11-26	207.29	207.76	207.03	207.64	61928400	206.50
2014-11-28	207.49	207.87	206.91	207.20	57890100	206.06
2014-12-01	206.40	206.54	205.38	205.76	103968400	204.63
2014-12-02	205.81	207.34	205.78	207.09	72105500	205.95
2014-12-03	207.30	208.15	207.10	207.89	69450000	206.75
2014-12-04	207.54	208.27	206.70	207.66	89928200	206.52
2014-12-05	207.87	208.47	207.55	208.00	91025500	206.86
2014-12-08	207.52	208.12	205.93	206.61	108588200	205.48
2014-12-09	204.37	206.60	203.91	206.47	125180100	205.34
2014-12-10	205.91	205.98	202.93	203.16	154761600	202.04
2014-12-11	203.88	206.19	203.71	204.19	159012800	203.07
2014-12-12	202.64	203.82	200.85	200.89	202330200	199.79
2014-12-15	201.98	202.53	198.78	199.51	189965800	198.41
2014-12-16	199.51	202.40	197.86	197.91	259543800	196.82
2014-12-17	198.44	202.34	198.29	201.79	253754500	200.68
2014-12-18	204.74	212.97	203.92	206.78	247780600	205.64
2014-12-19	206.43	207.33	205.61	206.52	226418200	206.52

In [34]:

```
# slice out adjusted close prices
adj_close = spy['Adj Close']
```

In [35]:

```
adj_close
```

Out[35]:

```
Date
2014-11-11    203.06
2014-11-12    202.84
2014-11-13    203.07
2014-11-14    203.12
2014-11-17    203.25
2014-11-18    204.42
2014-11-19    204.09
2014-11-20    204.45
2014-11-21    205.55
2014-11-24    206.12
2014-11-25    205.97
2014-11-26    206.50
2014-11-28    206.06
2014-12-01    204.63
2014-12-02    205.95
2014-12-03    206.75
2014-12-04    206.52
2014-12-05    206.86
2014-12-08    205.48
2014-12-09    205.34
2014-12-10    202.04
2014-12-11    203.07
2014-12-12    199.79
2014-12-15    198.41
2014-12-16    196.82
2014-12-17    200.68
2014-12-18    205.64
2014-12-19    206.52
Name: Adj Close, dtype: float64
```

In [39]:

```
# compute rolling mean
threedayrolling_avg = pd.rolling_mean(adj_close, 3)
threedayrolling_avg[-25:]
```

Out[39]:

```
Date
2014-11-14    203.010000
2014-11-17    203.146667
2014-11-18    203.596667
2014-11-19    203.920000
2014-11-20    204.320000
2014-11-21    204.696667
2014-11-24    205.373333
2014-11-25    205.880000
2014-11-26    206.196667
2014-11-28    206.176667
2014-12-01    205.730000
2014-12-02    205.546667
2014-12-03    205.776667
2014-12-04    206.406667
2014-12-05    206.710000
2014-12-08    206.286667
2014-12-09    205.893333
2014-12-10    204.286667
2014-12-11    203.483333
2014-12-12    201.633333
2014-12-15    200.423333
2014-12-16    198.340000
2014-12-17    198.636667
2014-12-18    201.046667
2014-12-19    204.280000
dtype: float64
```

In [40]:

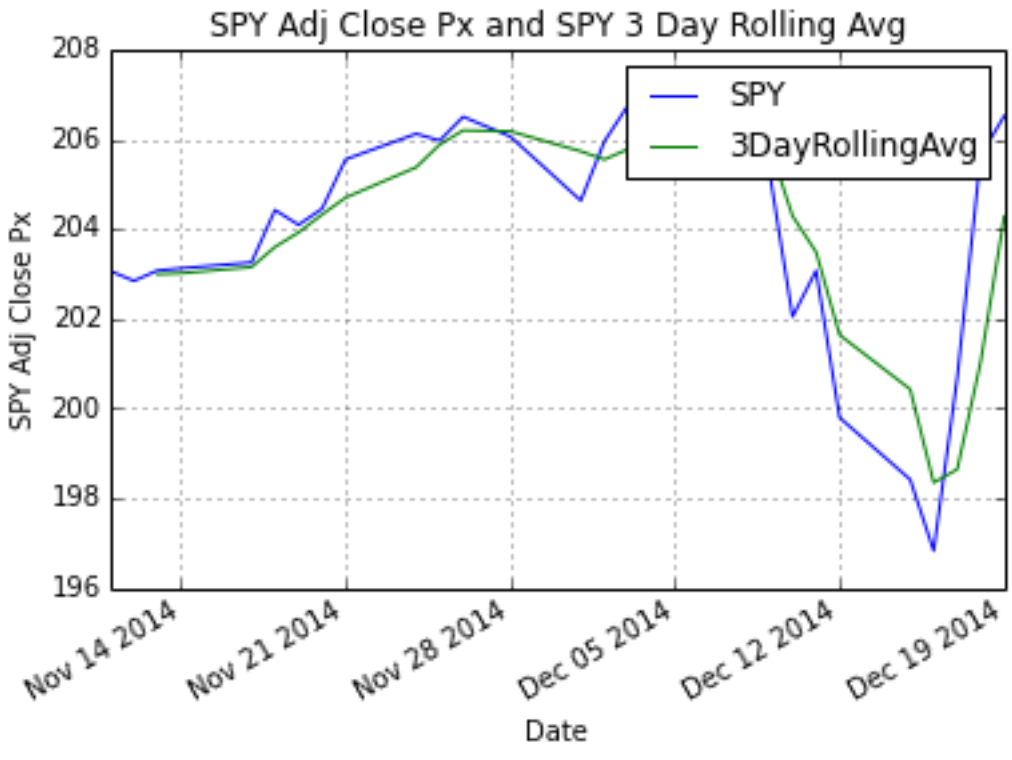
```
# compute returns
rets = adj_close / adj_close.shift(1) - 1
rets
```

Out[40]:

```
Date
2014-11-11         NaN
2014-11-12   -0.001083
2014-11-13    0.001134
2014-11-14    0.000246
2014-11-17    0.000640
2014-11-18    0.005756
2014-11-19   -0.001614
2014-11-20    0.001764
2014-11-21    0.005380
2014-11-24    0.002773
2014-11-25   -0.000728
2014-11-26    0.002573
2014-11-28   -0.002131
2014-12-01   -0.006940
2014-12-02    0.006451
2014-12-03    0.003884
2014-12-04   -0.001112
2014-12-05    0.001646
2014-12-08   -0.006671
2014-12-09   -0.000681
2014-12-10   -0.016071
2014-12-11    0.005098
2014-12-12   -0.016152
2014-12-15   -0.006907
2014-12-16   -0.008014
2014-12-17    0.019612
2014-12-18    0.024716
2014-12-19    0.004279
Name: Adj Close, dtype: float64
```

In [41]:

```
adj_close.plot(label='SPY')
threedayrolling_avg.plot(label='3DayRollingAvg')
p.ylabel('SPY Adj Close Px')
p.title('SPY Adj Close Px and SPY 3 Day Rolling Avg')
p.legend()
p.show()
```



In [43]:

```
# prune the returns
pruned_rows = rets[3:28]
pruned_rows
```

Out[43]:

```
Date
2014-11-14    0.000246
2014-11-17    0.000640
2014-11-18    0.005756
2014-11-19   -0.001614
2014-11-20    0.001764
2014-11-21    0.005380
2014-11-24    0.002773
2014-11-25   -0.000728
2014-11-26    0.002573
2014-11-28   -0.002131
2014-12-01   -0.006940
2014-12-02    0.006451
2014-12-03    0.003884
2014-12-04   -0.001112
2014-12-05    0.001646
2014-12-08   -0.006671
2014-12-09   -0.000681
2014-12-10   -0.016071
2014-12-11    0.005098
2014-12-12   -0.016152
2014-12-15   -0.006907
2014-12-16   -0.008014
2014-12-17    0.019612
2014-12-18    0.024716
2014-12-19    0.004279
Name: Adj Close, dtype: float64
```

In [44]:

```
# explore market data
# descriptive statistics of the market returns
mu_rets = np.mean(rets)
print "mean of daily return distribution: ",  mu_rets
```

```
mean of daily return distribution:  0.000661064613188
```

In [45]:

```
sigma_rets = np.std(rets)
print "standard deviation of daily return distribution: ", sigma
```

```
standard deviation of daily return distribution:  0.389834376345
```

In [46]:

```
min_rets = np.min(rets)
print "min of daily return distribution: ", min
```

```
min of daily return distribution:  -0.834683
```

In [47]:

```
max_rets = np.max(rets)
print "max of daily return distribution: ", max
```

```
max of daily return distribution:  0.443122
```

In [58]:

```
cum_rets = np.cumsum(rets)
print "cumulative return: ", cum_rets
```

```
cumulative return:  Date
2014-11-11         NaN
2014-11-12   -0.001083
2014-11-13    0.000050
2014-11-14    0.000297
2014-11-17    0.000937
2014-11-18    0.006693
2014-11-19    0.005079
2014-11-20    0.006843
2014-11-21    0.012223
2014-11-24    0.014996
2014-11-25    0.014268
2014-11-26    0.016842
2014-11-28    0.014711
2014-12-01    0.007771
2014-12-02    0.014222
2014-12-03    0.018106
2014-12-04    0.016994
2014-12-05    0.018640
2014-12-08    0.011969
2014-12-09    0.011288
2014-12-10   -0.004783
2014-12-11    0.000315
2014-12-12   -0.015837
2014-12-15   -0.022745
2014-12-16   -0.030758
2014-12-17   -0.011147
2014-12-18    0.013569
2014-12-19    0.017849
Name: Adj Close, dtype: float64
```

In [59]:

```
median_rets = np.median(rets)
print "median of daily return distribution: ", median
```

```
median of daily return distribution:  -0.245084
```

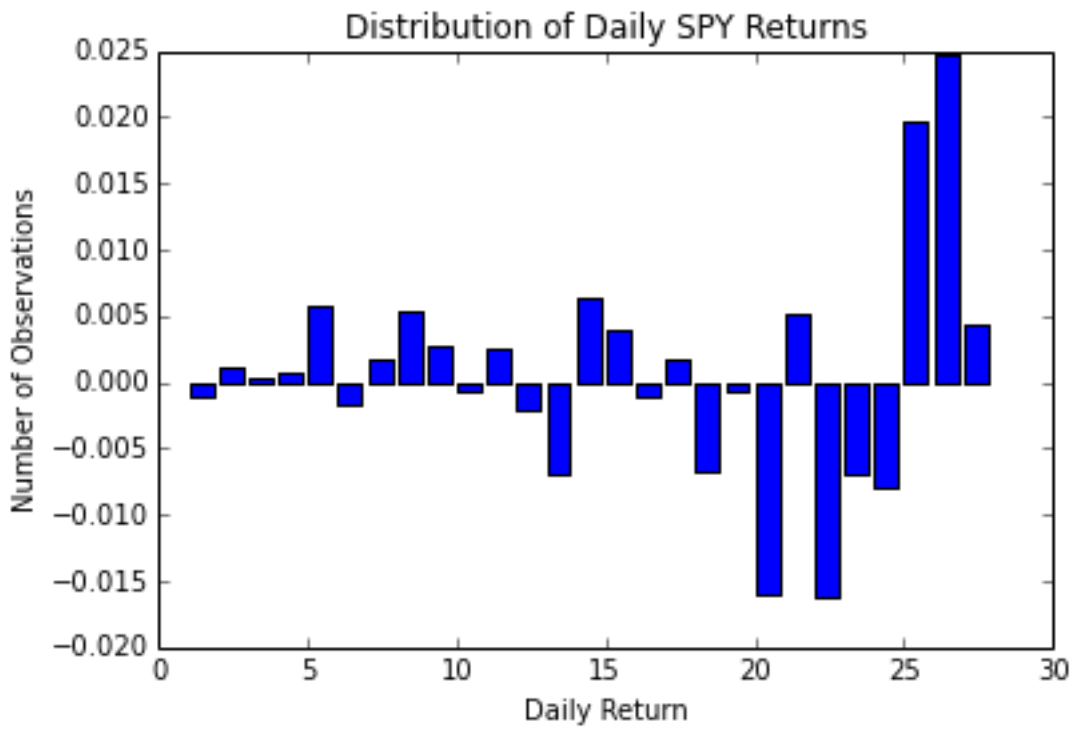
In [60]:

```
total_observations_rets = pos_count+neg_count+neutral_count
print "total number of observations: ", int(total_observations)
```

```
total number of observations:  25
```

In [61]:

```
# visualizations of exploratory data analysis of SPY market index
ind_rets = np.arange(len(rets))
p.bar(ind_rets,rets)
p.xlabel('Daily Return')
p.ylabel('Number of Observations')
p.title('Distribution of Daily SPY Returns')
p.show()
```



In [68]:

```
# model the data - correlation analysis of cumulative market returns and sentimen
t scores
# function returns (Pearson's correlation coefficient, 2-tailed p-value)
stats.pearsonr(cum_rets[3:28], ranked_sentiment_scores)
```

Out[68]:

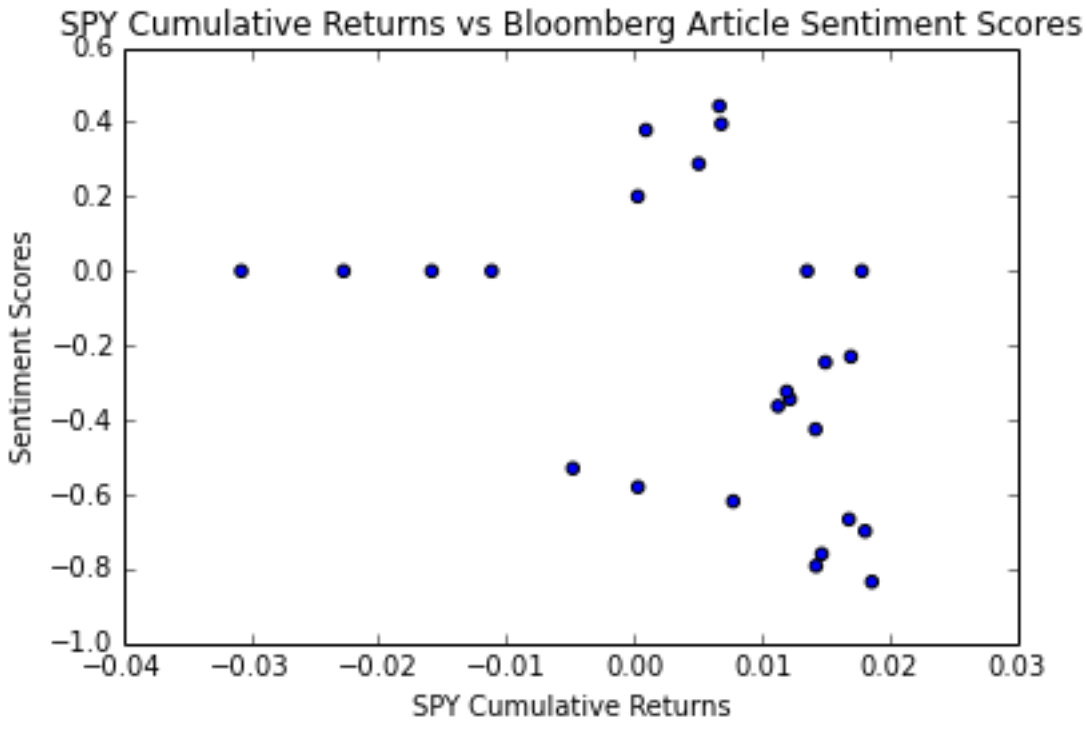
```
(-0.39979160333165109, 0.047694579773680371)
```

In [67]:

```
p.scatter(cum_rets[3:28],ranked_sentiment_scores)
p.xlabel('SPY Cumulative Returns')
p.ylabel('Sentiment Scores')
p.title('SPY Cumulative Returns vs Bloomberg Article Sentiment Scores')
```

Out[67]:

```
<matplotlib.text.Text at 0x109d39590>
```



In [70]:

```
# model the data - correlation analysis of daily spy returns and sentiment scores
# function returns (Pearson's correlation coefficient, 2-tailed p-value)
stats.pearsonr(rets[3:28], ranked_sentiment_scores)
```

Out[70]:

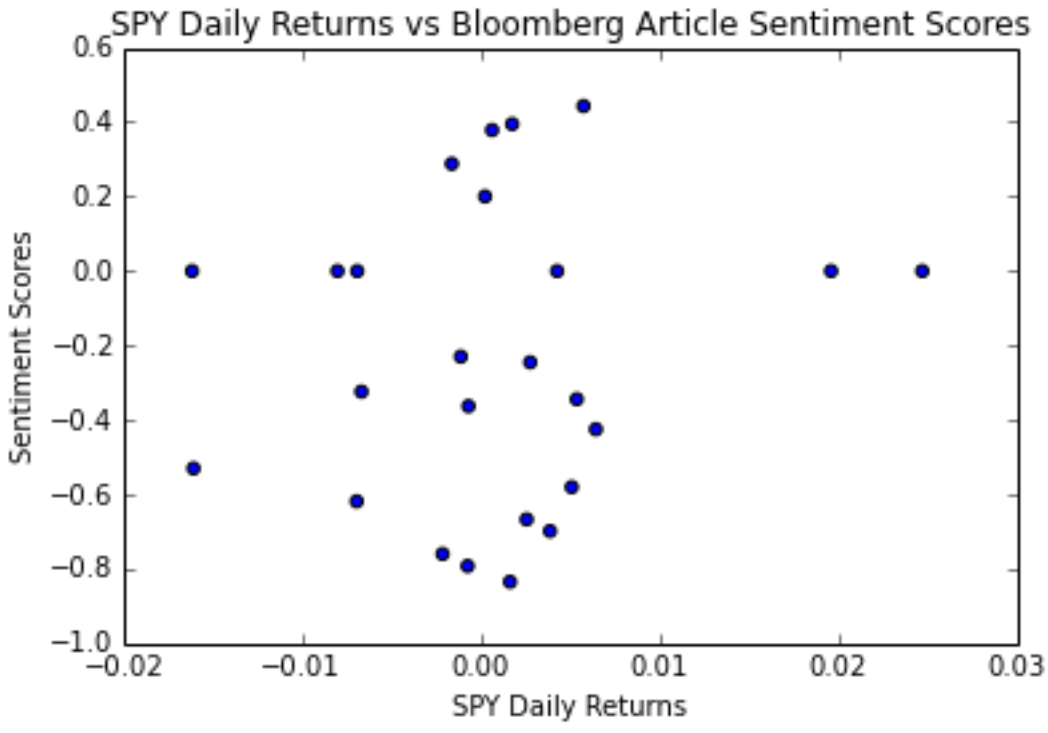
```
(0.12680417187625165, 0.5458385362196203)
```

In [71]:

```
p.scatter(rets[3:28],ranked_sentiment_scores)
p.xlabel('SPY Daily Returns')
p.ylabel('Sentiment Scores')
p.title('SPY Daily Returns vs Bloomberg Article Sentiment Scores')
```

Out[71]:

```
<matplotlib.text.Text at 0x1099c3a10>
```



In [80]:

```
# ordinary least squares regression
model = smf.OLS(cum_rets[3:28],ranked_sentiment_scores)
results = model.fit()
results.summary()
```

Out[80]:

OLS Regression Results

Dep. Variable:	Adj Close	**R-squared:**	0.265
Model:	OLS	**Adj. R-squared:**	0.234
Method:	Least Squares	**F-statistic:**	8.636
Date:	Sun, 21 Dec 2014	**Prob (F-statistic):**	0.00718
Time:	23:27:24	**Log-Likelihood:**	74.690
No. Observations:	25	**AIC:**	-147.4
Df Residuals:	24	**BIC:**	-146.2
Df Model:	1		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
x1	-0.0162	0.006	-2.939	0.007	-0.028 -0.005

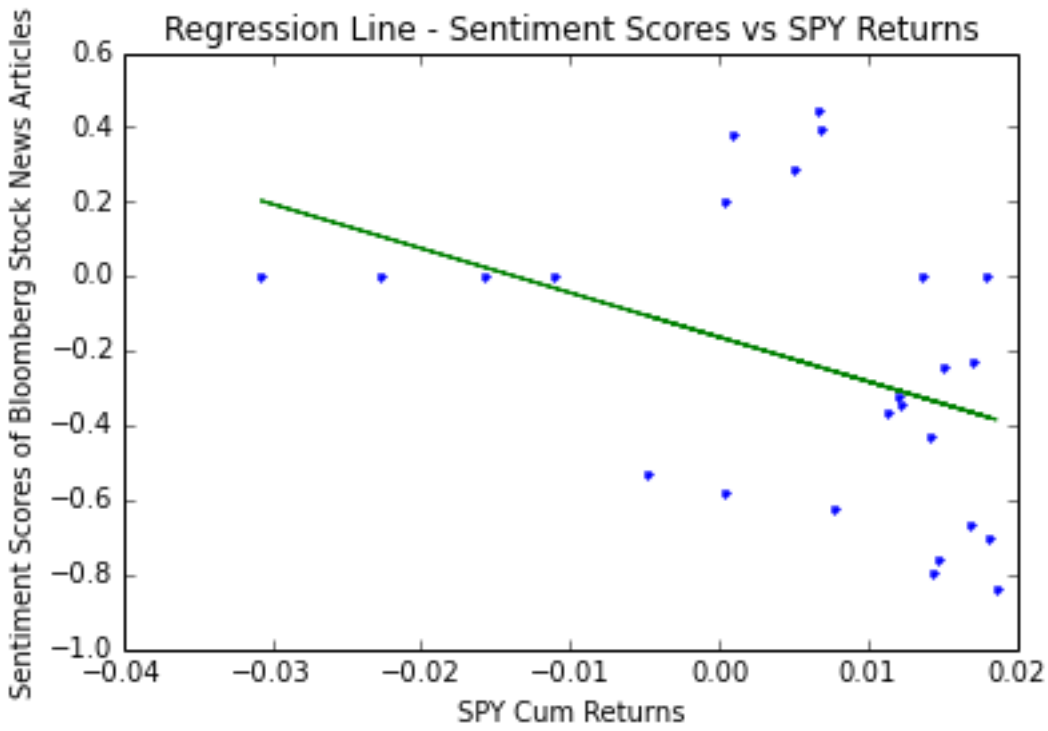
Omnibus:	7.224	**Durbin-Watson:**	0.550
Prob(Omnibus):	0.027	**Jarque-Bera (JB):**	5.569
Skew:	-1.130	**Prob(JB):**	0.0618
Kurtosis:	3.486	**Cond. No.**	1.00

In [85]:

```
# regression line
x = cum_rets[3:28]
y = ranked_sentiment_scores
m, b = np.polyfit(x, y, 1)
p.xlabel('SPY Cum Returns')
p.ylabel('Sentiment Scores of Bloomberg Stock News Articles')
p.title('Regression Line - Sentiment Scores vs SPY Returns')
p.plot(x, y, '.')
p.plot(x, m*x + b, '-')
```

Out[85]:

```
[<matplotlib.lines.Line2D at 0x109423b90>]
```



In [ ]: