S&P 500 Index Analysis Report

Yanni Ge, Ying Shi, Hairong Xie

May 9, 2014

Part I

Introduction

The Standard & Poor's 500 (S&P 500) is a stock market index based on the largest 500 publicly-traded companies in terms of the market capitalization. Its diverse constituency and weighting methodology make it superior than other index as one of the most commonly followed equity indices, and many consider it one of the best representations of the U.S. stock market, and a bellwether for the U.S. economy. It is also well-known that the S&P 500 index has an increasing trend, best performance and lowest risk in the long term, so investors are recommended to follow the index closely as one of the most critical investment strategies.

In order to provide investors a deeper understanding of the S&P 500 index, this project introduces several statistical tools to wisely track, visualize and predict the index. Based on the historical data from Yahoo finance, we provide investors with an affordable portfolio by applying machine learning techniques, such as LASSO and Principal Components Analysis (PCA), on the stock prices of the S&P 500 index companies and the index itself.

We further analyze the daily return of the S&P 500 index in terms of its probability and distribution. By approximating the normal distribution of the index, we make predictions and validate on testing data. In order to improve the prediction power, we also apply the time series model ARMA based on log daily returns which is proved to be a better predictor. At last, we run 24 test to make sure our code and results are reasonable and efficient.

Part II

Packages and Github

The data analysis in this project is implemented entirely in ipython notebook, and it is completely reproducible with the notebook and internet access. All the Python packages needed are imported at the beginning. As per comments from the peer review, the BeautifulSoup package has been updated to the newest version called bs4 for the convenience of users working on their local machines.

```
/usr/lib/python2.7/dist-packages/nose/util.py:14: DeprecationWarning:
The compiler package is deprecated and removed in Python 3.x.
```

```
  from compiler.consts import CO_GENERATOR
WARNING:

Populating the interactive namespace from numpy and matplotlib

pylab import has clobbered these variables: ['mpl', 'poisson']
`%pylab --no-import-all` prevents importing * from pylab and numpy
```

We used Github for version control for the ease of the cooperation between the three of us. The link is https://github.com/hairong/YahooFinanceProject.

Part III

Data

Our data set contains two objects: a vector of the S&P 500 index starting from 01/04/2010 to now and a dataframe. The dataframe records the adjusted closing prices of the companies that S&P 500 consists of. Each row represents each date, the range of which is the same as the index. Each column represents each company, but those companies that did not start from 01/04/2010 have been deleted. Thus, there are no missing values in the data set.

For reproducibility purpose, the S&P 500 index and the adjust closing prices of the companies are automatically downloaded from the internet. The tickers of the stocks are parsed from Wikipedia, which contains a table listing the most up-to-date S&P 500 index companies. We then fixed the spelling mismatch with the Yahoo Finance, and then passed them into the DataReader function to download the index or prices. Through this automatic way, the tickers are always updated, since the companies that make the S&P 500 index change once in a while. Also, all the data that are latest available can be included every time the code runs. For the purpose of this report, we fix the ending date to be 05/08/2014, which gives a vector of length 1094 and a dataframe of dimension 1094 by 481.

Here is a snapshot of the first few rows and colunms of the dataframe.

```
/usr/lib/python2.7/dist-packages/pandas/core/config.py:570:
DeprecationWarning: height has been deprecated.

  warnings.warn(d.msg, DeprecationWarning)
/usr/lib/python2.7/dist-packages/pandas/core/config.py:570:
DeprecationWarning: height has been deprecated.

  warnings.warn(d.msg, DeprecationWarning)
```

```
Out [7]:             A     AA    AAPL    ABC    ABT    ACE    ACN    ACT
Date
2010-01-04  30.60  15.84  204.55  25.13  22.95  44.53  38.19  40.29
2010-01-05  30.27  15.35  204.91  24.95  22.76  43.52  38.43  39.89
2010-01-06  30.16  16.15  201.65  24.72  22.89  42.93  38.83  40.02
2010-01-07  30.12  15.80  201.28  24.32  23.08  43.17  38.80  39.70
2010-01-08  30.11  16.19  202.61  24.58  23.19  42.93  38.64  39.41
2010-01-11  30.13  16.60  200.83  24.86  23.31  43.39  38.61  39.77
2010-01-12  29.77  14.77  198.54  25.03  23.25  43.48  38.36  39.72
2010-01-13  30.00  15.20  201.34  25.52  23.47  43.62  38.80  40.89
2010-01-14  30.45  15.04  200.18  25.68  23.49  44.04  39.14  41.10
```

```
2010-01-15  29.75  14.87  196.83  25.40  23.55  43.57  38.86  40.90
2010-01-19  30.11  14.86  205.54  25.83  23.85  44.12  39.72  41.83
2010-01-20  29.94  14.49  202.37  25.61  23.87  44.05  39.50  41.54
2010-01-21  29.84  13.56  198.88  25.08  23.58  43.75  39.12  40.41
2010-01-22  28.52  12.75  189.01  25.16  23.14  43.28  38.29  39.51
2010-01-25  28.79  12.74  194.10  25.58  23.22  43.78  38.02  39.39
```

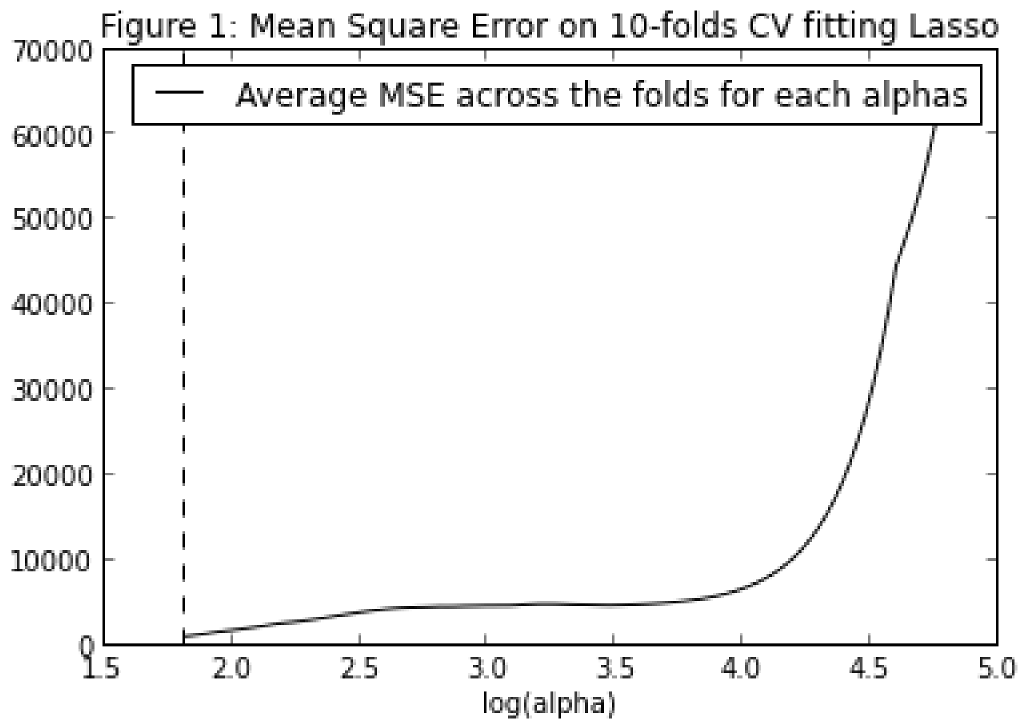
Part IV

Machine Learning for Index Tracking

For most investors, tracking the S&P 500 index by purchasing all 500 stocks is unrealistic due to the huge fund and commission fee involved. The solution that we will provide investors is a portfolio composed of a small subset of the stocks so that investors are capable of investing.

In order to choose such a subset that not only well represents the S&P 500 index but also has just a small number of stocks, Lasso naturally became a ideal choice of model. We regress the S&P 500 index on all the adjusted closing prices of the companies, i.e. the columns of the dataframe. The objective is to find the coefficients of all the columns that can minimize the least sum of squared error. In order to make most the coefficients to be zero so that only a small subset will be selected into the portfolio, we add a Lasso penalty term to the minimization. The penalty term is a constant penalty parameter, called alpha in Python, times the sum of the absolute values of all the columns' coefficients. The size of alpha will determine the number of the coefficients to be zero.

To obtain the optimal size of alpha, we did 10-fold cross-validation. The data set is randomly partitioned into 90% as training set and the rest 10% as testing set. For a particular alpha, the traning set is used to fit the model, and then the fitted model is applied on the testing set to get the fitted value. Then the mean squared error of the testing can be calulated. This process has to be repeated for 10 times so that every row has been included as testing data for once. The average of the mean squared errors calculated from the 10 partitions is reported. For a sequance of 100 alpha's, the corresponding averages of the mean squared errors are plotted in Figure 1.



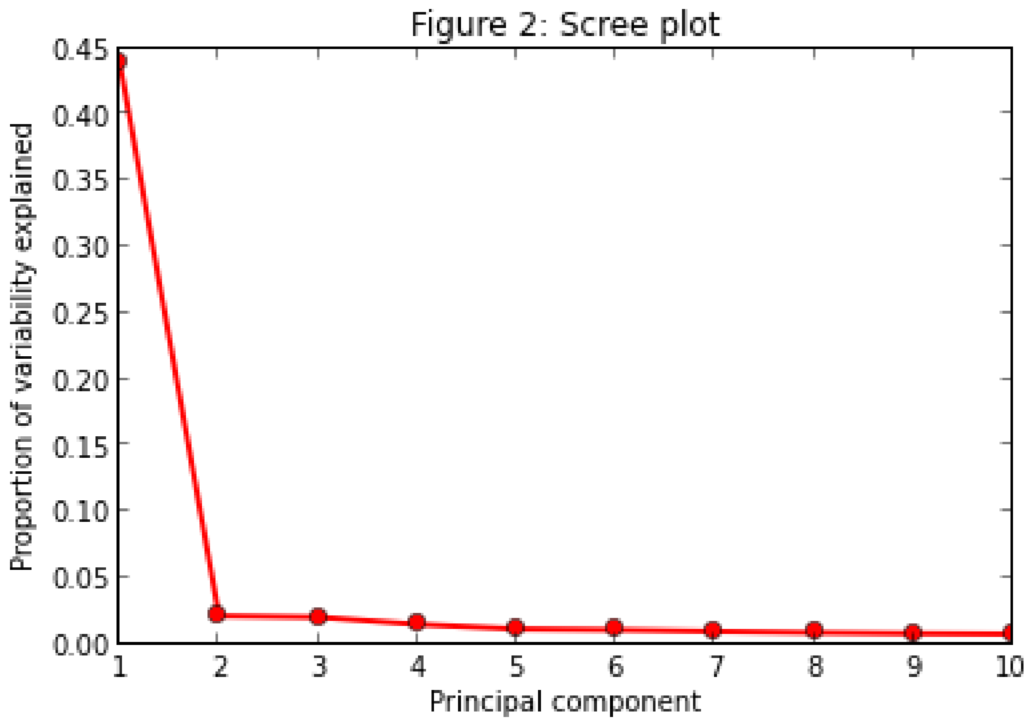
The alpha that gives us the least average mean squared error, denoted by the dotted line, is selected. Using this particular alpha, Lasso picks the 17 stocks shown below. The corresponding company names are Apple Inc., Amazon.com Inc., BlackRock Inc., Caterpillar Inc., Edwards Lifesciences Corp., First Solar Inc., Graham Holdings, Google Inc., The Goldman Sachs Group Inc., W.W. Grainger Inc., IBM Corp., Intuitive Surgical Inc., Netflix Inc., The Priceline Group Inc., Ralph Lauren Corp., The Sherwin-Williams Company, and Union Pacific Corp. Less than 20 companies are selected into the investment portfolio, which is much more manageable than purchasing all 500 stocks for most investors.

Ticker	Company name
AAPL	Apple Inc.
AMZN	Amazon.com Inc.
BLK	BlackRock Inc.
CAT	Caterpillar Inc.
EW	Chevron Corp.
FSLR	Edward Lifesciences Corp.
GHC	Graham Holdings.

GOOGL	Google Inc.
GS	The Goldman Sachs Group Inc.
GWW	W.W. Grainger Inc.
IBM	IBM Corp.
ISRG	Netflix Inc.
NFLX	The Priceline Group Inc.
PCLN	Raulph Lauren Corp.
RL	Sherwin-Williams Company
SHW	Simon Property Group Inc.
UNP	Union Pacific Corp.

Next step is to visualize the data as well as the portfolio to gain deeper insights. We apply PCA to reduce the data dimentionality and do further analysis. As a preprocessing step, we calculate the daily excess returns of all the companies, i.e. the percentage change of the prices between one day and the next day. Note that this new dataframe containing the daily excess returns has one less row, because we exclude the one for the first day. By applying PCA, we project the data onto new uncorrelated directions. The new directions are the ones that make the projection to have the minimum squared errors, or equivalently the maximum variances. Mathematically, they are the eigenvectors of the covariance matrix of the data, and the corresponding eigenvalues represent the proportions of the variances explained. The projections are called the principal components.

Figure 2 is the Scree plot displaying the eigenvalues. The proportion of variability explained by each pincipal component has been plotted in order. The first principal component can explain around 45% of the variability. Thus, the data can be reduced to lower dimensions.



Now, we plot the first eigenvector against the second eigenvector in Figure 3.

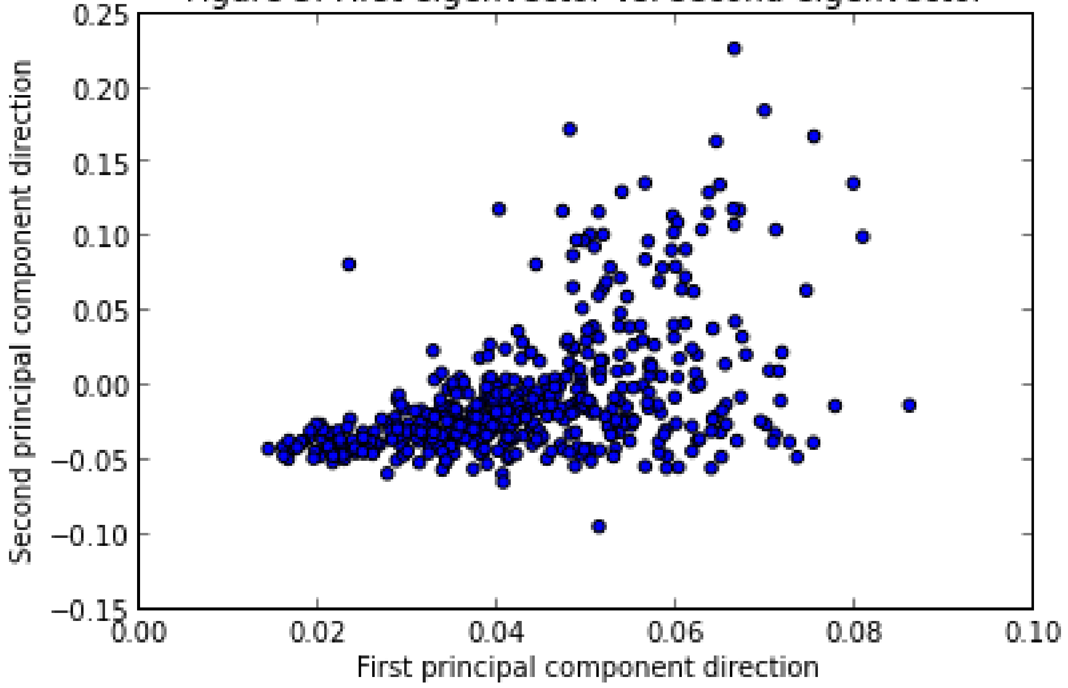


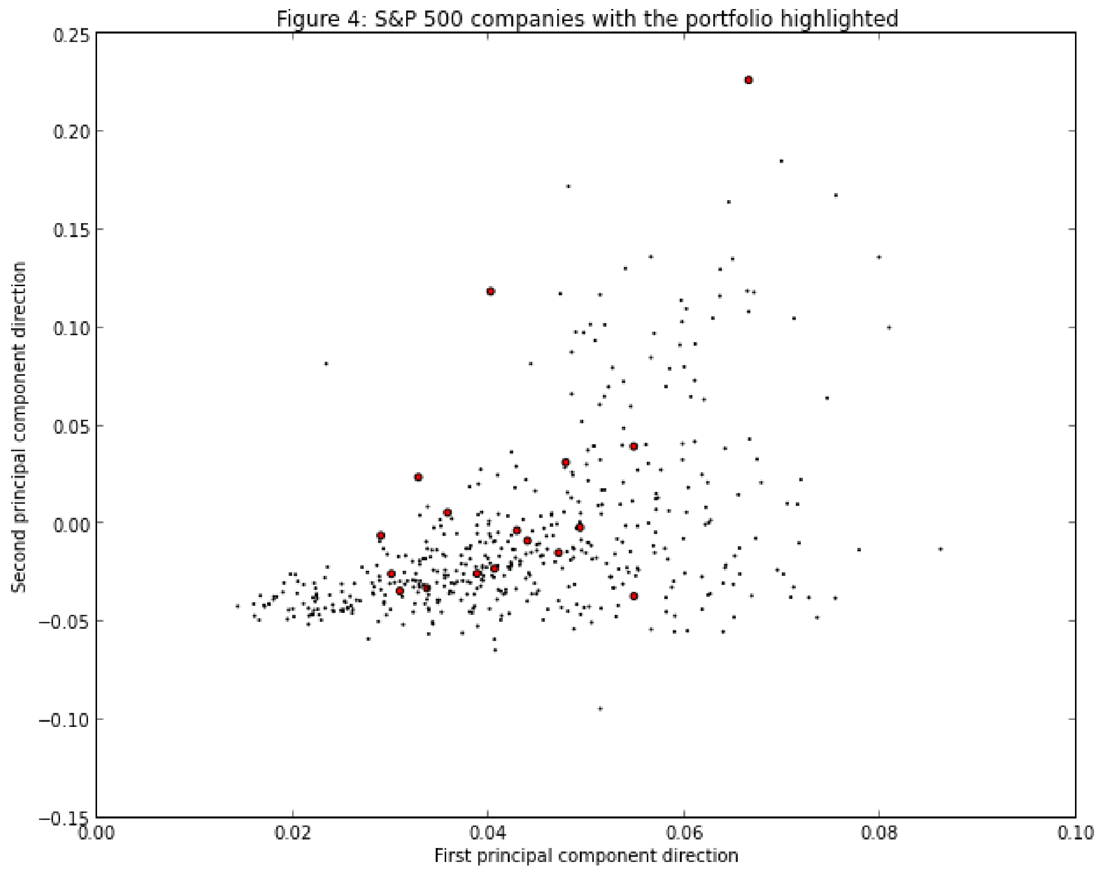
Figure 3: First eigenvector vs. second eigenvector

All the entries in the first eigenvector have the positive signs, which means that the dominant factor is market-wise and drives all the stocks move in the same direction. Similarly for the second eigenvector, about 1/4 of the entries have different signs with the remaining majority has positive signs. This reflects the factor that drives returns oppositely for different stocks. Tickers corresponding to those 1/4 stocks are listed below.

```
[u′A′ u′AA′ u′AAPL′ u′ADI′ u′ADSK′ u′AKAM′ u′ALTR′ u′AMAT′ u′AME′
u′APA′
 u′APC′ u′APD′ u′APH′ u′ARG′ u′ATI′ u′AVGO′ u′BHI′ u′BRCM′ u′BTU′
u′BWA′
 u′CAM′ u′CAT′ u′CBG′ u′CF′ u′CHK′ u′CMI′ u′CNX′ u′COG′ u′COP′ u′CRM′
 u′CSC′ u′CSX′ u′CTSH′ u′CTXS′ u′CVX′ u′DD′ u′DE′ u′DNR′ u′DO′ u′DOV′
 u′DOW′ u′DVN′ u′EA′ u′EBAY′ u′EMC′ u′EMN′ u′EMR′ u′EOG′ u′EQT′ u′ESV′
 u′ETFC′ u′ETN′ u′EXPE′ u′FCX′ u′FFIV′ u′FLR′ u′FLS′ u′FMC′ u′FOSL′
u′FSLR′
 u′FTI′ u′GLW′ u′GMCR′ u′GOOGL′ u′GT′ u′HAL′ u′HAR′ u′HES′ u′HP′
u′HPQ′
 u′IP′ u′IR′ u′JBL′ u′JCI′ u′JEC′ u′JNPR′ u′JOY′ u′KLAC′ u′LLTC′
u′LRCX′
 u′LUK′ u′MCHP′ u′MON′ u′MOS′ u′MRO′ u′MU′ u′MUR′ u′NBL′ u′NBR′ u′NE′
 u′NEM′ u′NFLX′ u′NFX′ u′NOV′ u′NRG′ u′NTAP′ u′NUE′ u′NVDA′ u′OI′
u′ORCL′
 u′OXY′ u′PCAR′ u′PCLN′ u′PH′ u′PX′ u′PXD′ u′QCOM′ u′RDC′ u′RHT′
u′RIG′
 u′ROK′ u′RRC′ u′SLB′ u′SNDK′ u′STX′ u′SWN′ u′SYMC′ u′TDC′ u′TEL′
u′TSO′
 u′TXN′ u′VLO′ u′VRSN′ u′WDC′ u′WMB′ u′WYNN′ u′X′ u′XLNX′ u′XOM′]
```

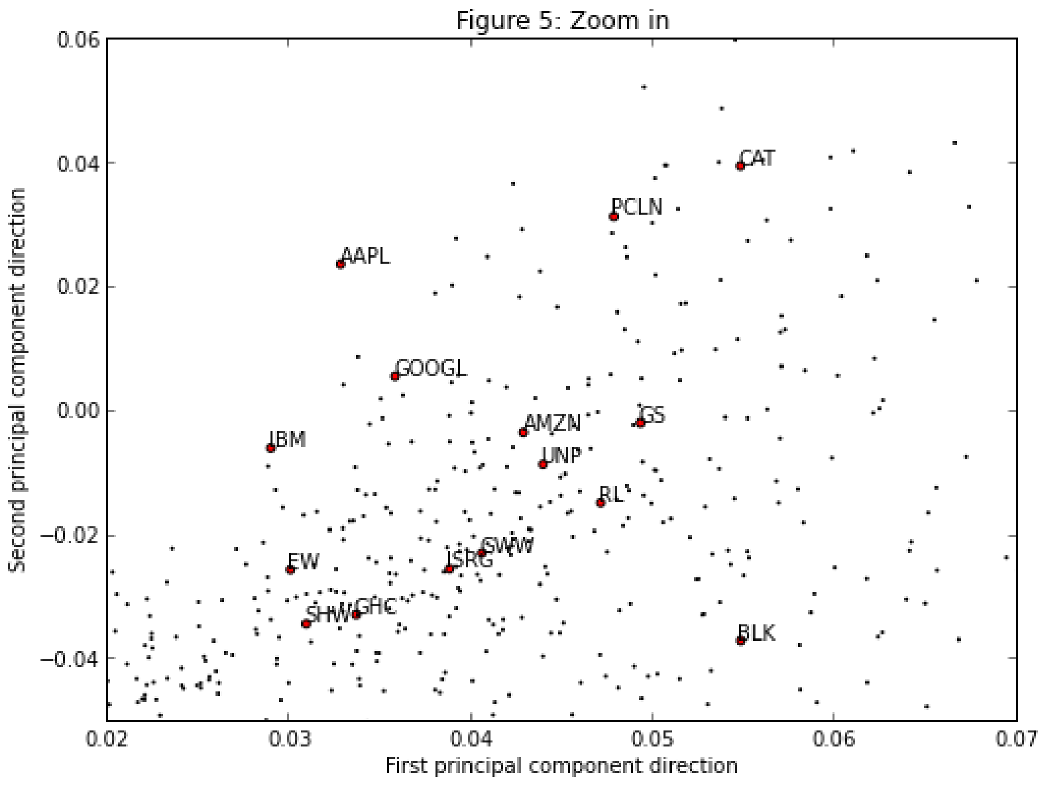
If we take a closer look at this list, we may find companies from manufacturing, materials, energy and natural resources. Some examples are Allegheny Technologies Inc. (ATI), Cummins Inc. (CMI), Consol Energy Inc. (CNX), Denbury Resources Inc. (DNR), Monsanto Company (MON), Newmont Mining Corp. (NEM), Southwestern Energy Co. (SWN), United States Steel Corp. (X) and etc. This makes sense since these industries in general move oppositely with the others. Thus, we may consider that the second eigenvector suggests the industry effect.

We are now ready to visualize the portfolio. In Figure 4, companies in the fortfolio are labeled by the red dots.



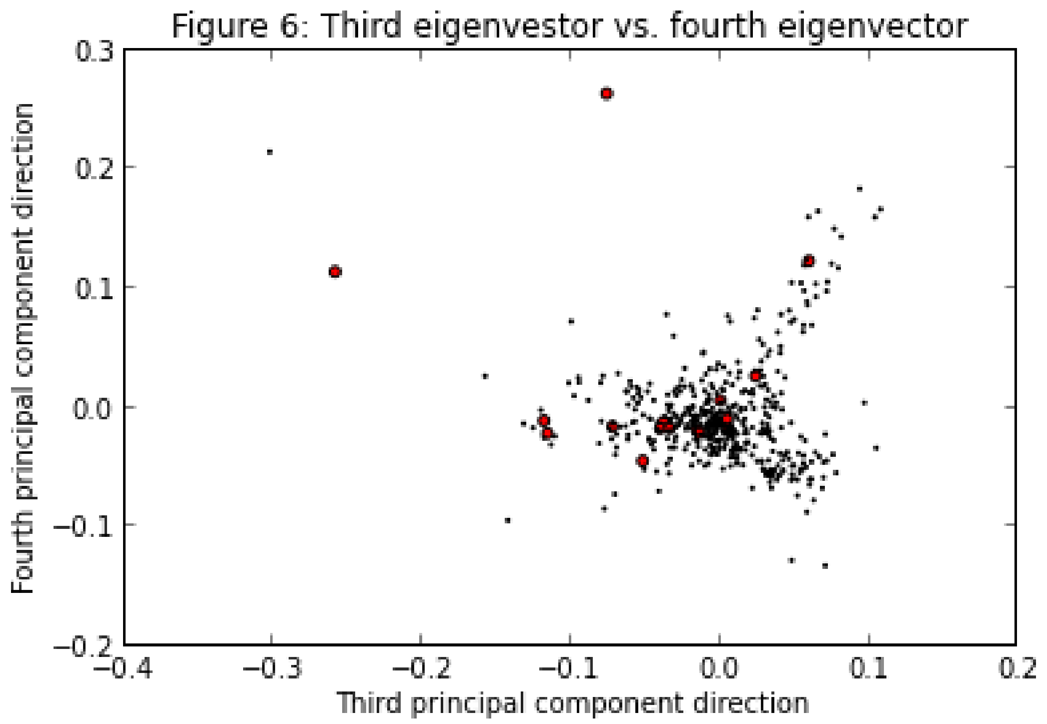
We can see that those red dots indeed well cover the whole trend so that the portfolio reflects the S&P 500 index companies. The portfolio can balance the return just as what the index achieves.

We zoom into the cloud in Figure 5 and label the companies in the portfolio.



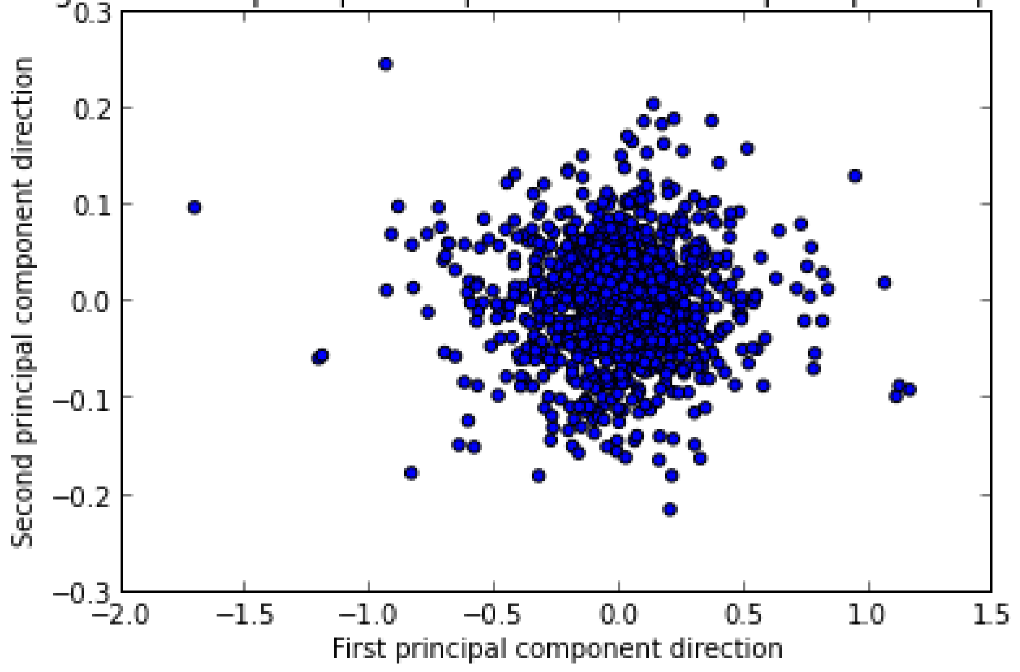
If taking a careful look at the Figure 5, we can find different industries are clustered, for example, Apple Inc., Google Inc. and IBM Corp. grouped toward the left hand side.

Here is a quick glance at the third eigenvector and fourth eigenvector in Figure 6.



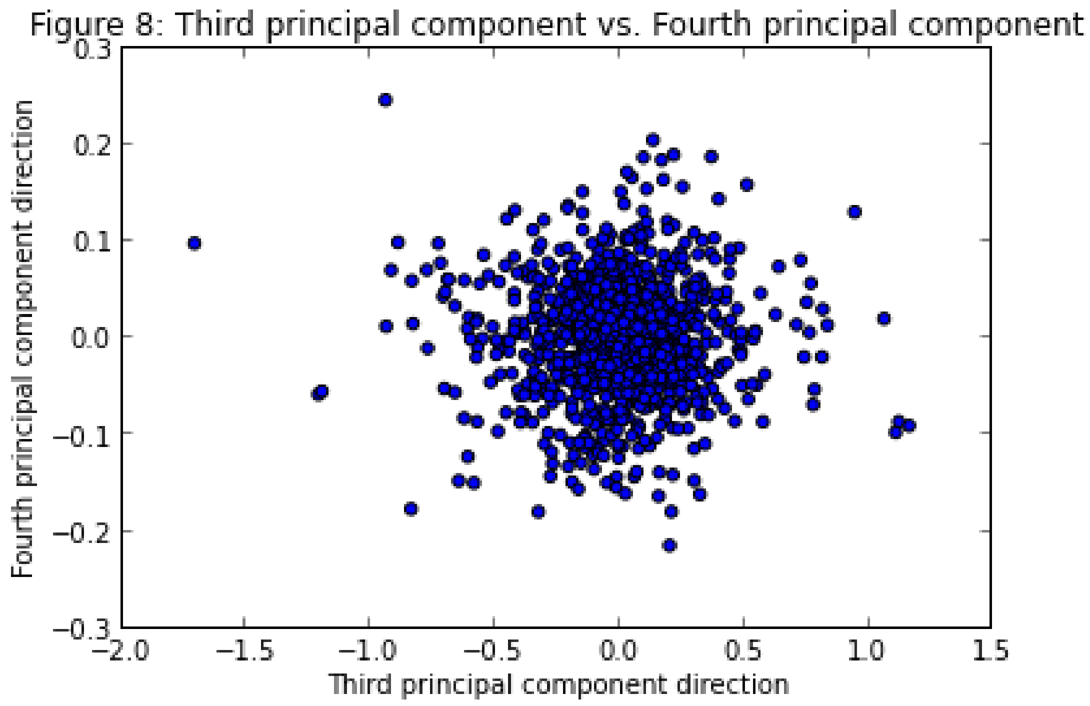
In addition, we will change a point of view and take a look at the dates instead of the companies, so we plot the first principal component against the second principal component in Figure 7.

Figure 7: First principal component vs. Second principal component



Here the dots represent the dates from 01/04/2010 to 05/09/2014. They are clustered around zero and have no obvious pattern, which makes sense since now the principal components are uncorrelated. This suggests that there are no extreme outliners within the time period. There have been several dates when big up-and-down's occurred, such as the points toward the edge. However, if an extremely unusual event occurs, it can make the rest of the dates form an obvious linear relationship so that the total correlation can be zero. Apparently that is not the case for the date set here.

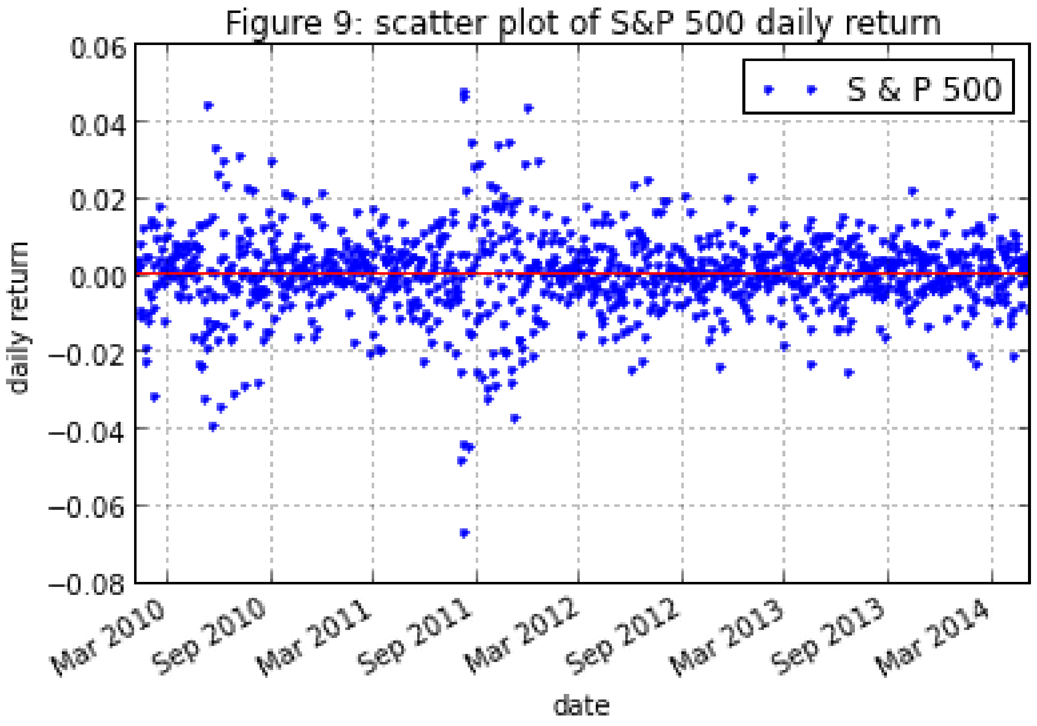
Here is a quick glance at the third principal component against the fourth principal component in Figure 8.



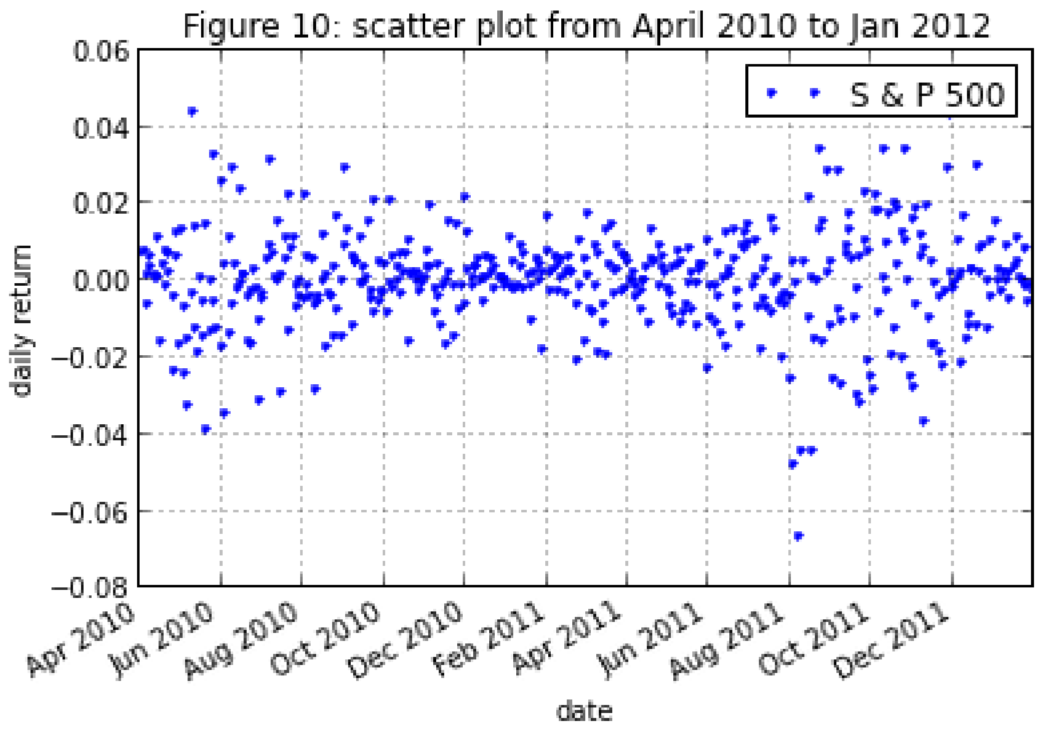
Part V

Probability Theory for Prediction

Naturally, investors have been most concerned about payback. To address this issue, we focus on the analyzing the daily return. Figure 9 is the scatter plot for the daily return of S&P 500 index. Almost all the data points are well surrounding a red center line with some random noise. From the plot, the larger variability around May 2010 and August 2011 suggests the "2010 Flash Crash" (http://en.wikipedia.org/wiki/2010_Flash_Crash) and "August 2011 stock markets fall" (http://en.wikipedia.org/wiki/August_2011_stock_markets_fall) when there were turmoil on stock market.



We zoom in the scatter plot for data from 04/01/2010 to 01/31/2012 in Figure 10 so that these large variablity could be shown more clearly.



The daily return of S&P 500 index has been standardized and its density is plotted in Figure 11.

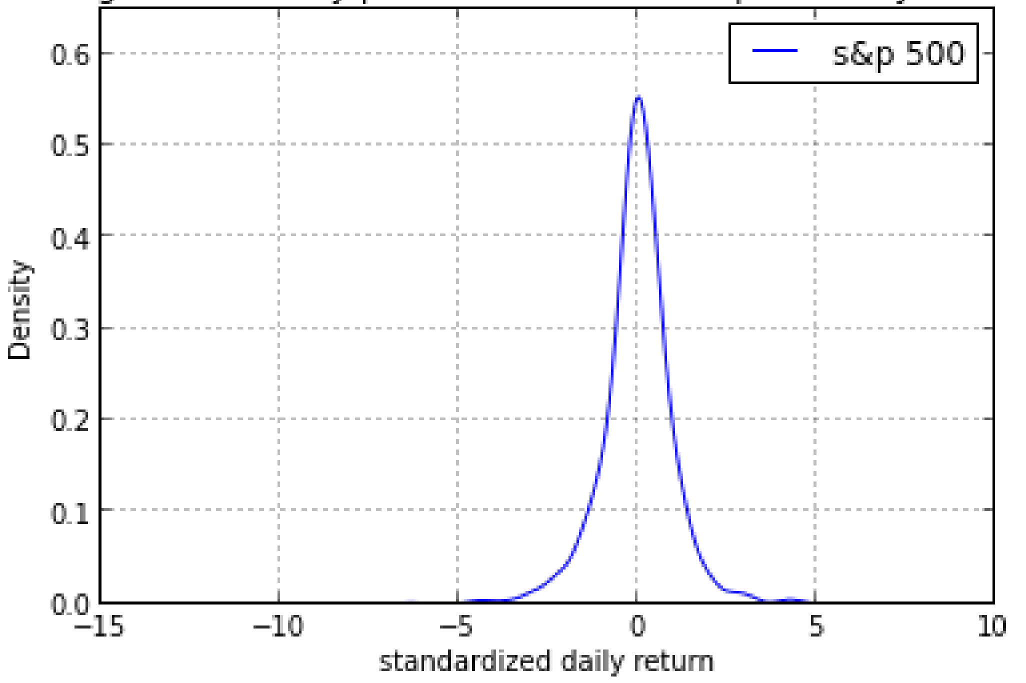
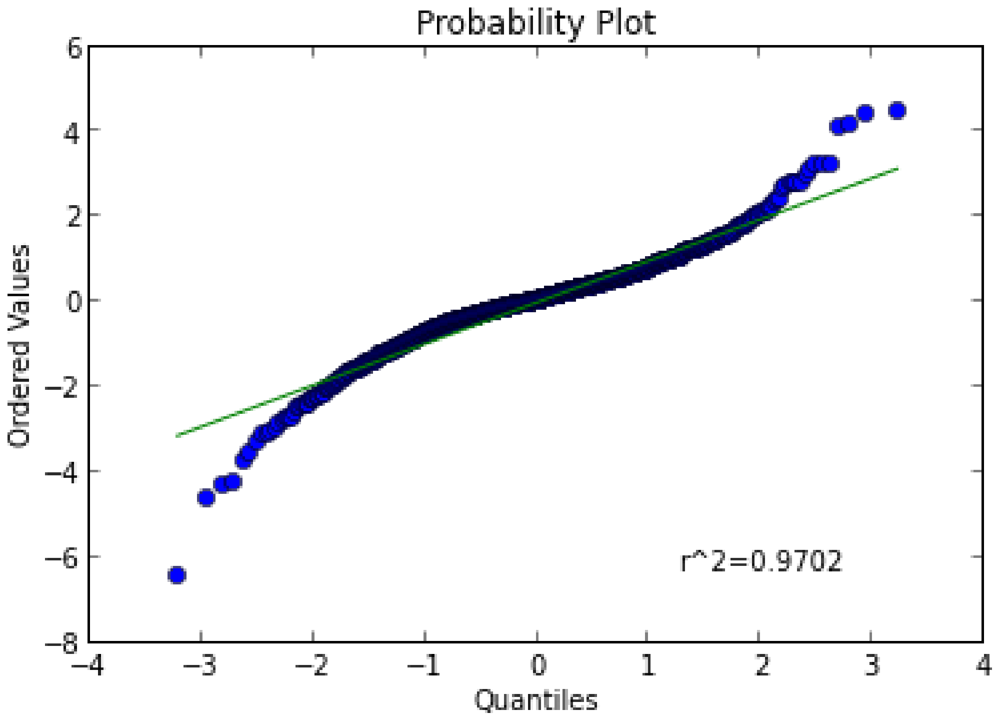
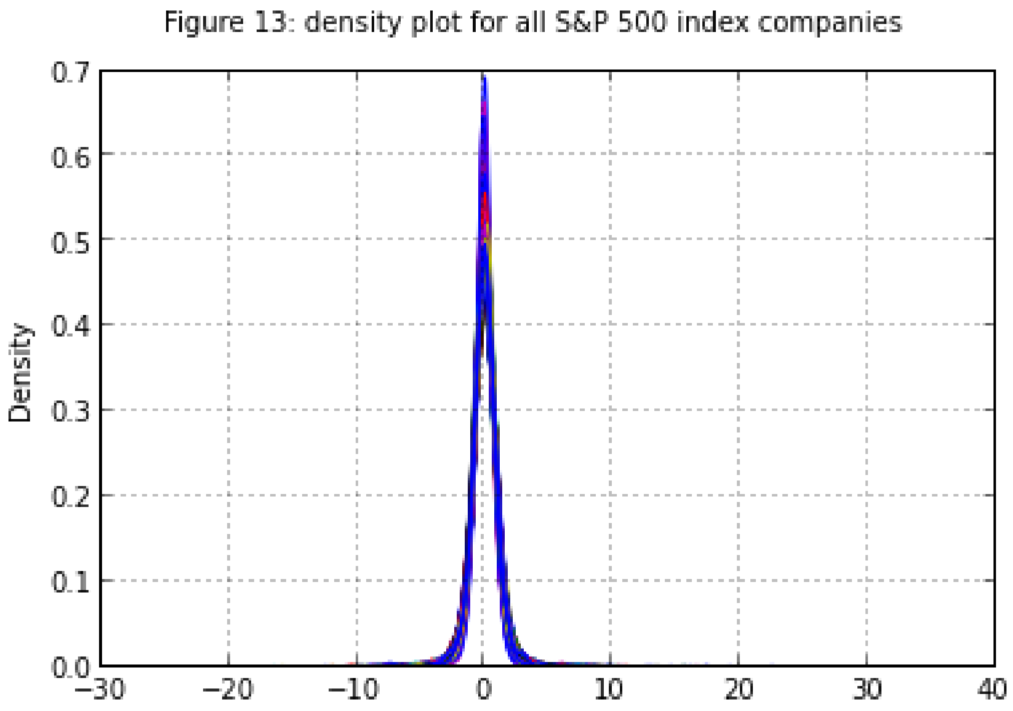


Figure 11:density plot of standardized s&p 500 daily return

The exact distribution of daily return has been discussed a lot by researchers. For example, Boothe and Glassman (1987) (Boothe, P., and Glassman, D., “The Statistical Distribution Of Exchange Rates,” Journal of International Economics, May 1987, pp. 297-319.) examined four distributions to approximate the exact distribution: the normal, the symmetric Stable Paretian, the Student t, and a mixture of normals. None of them has been confirmed to be the best model. From our plot, the daily return seems to follow a bell-shaped distribution with a small fluctuation on the tails. It is more centered to the mean than the standard normal distribution and there is large probability that the observations are close to the mean. The mean and mode almost coincide with each other.



The Q-Q plot in Figure 10 further examines the normality assumption. In the center, the observations nearly fit into the straight line which means they could be approximated as normal distribution well. However, the extreme values could be far away from the straight line and it suggests that the extreme values deviates from the normal distribution.



We try to explore the density for the daily return for all the companies in Figure 13. These plots seem to overlap with each other and it is difficult to distinguish them in this plot.

```
<matplotlib.figure.Figure at 0x57e7850>
```

Figure 14: density plots for companies selected by LASSO

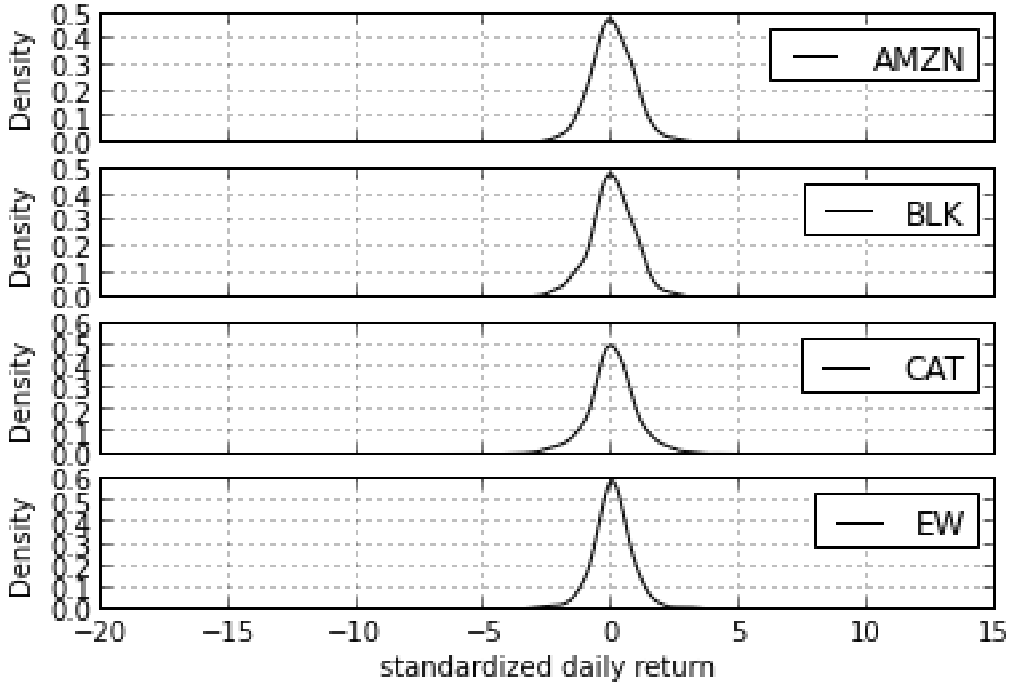


Figure 15: density plots for companies selected by LASSO

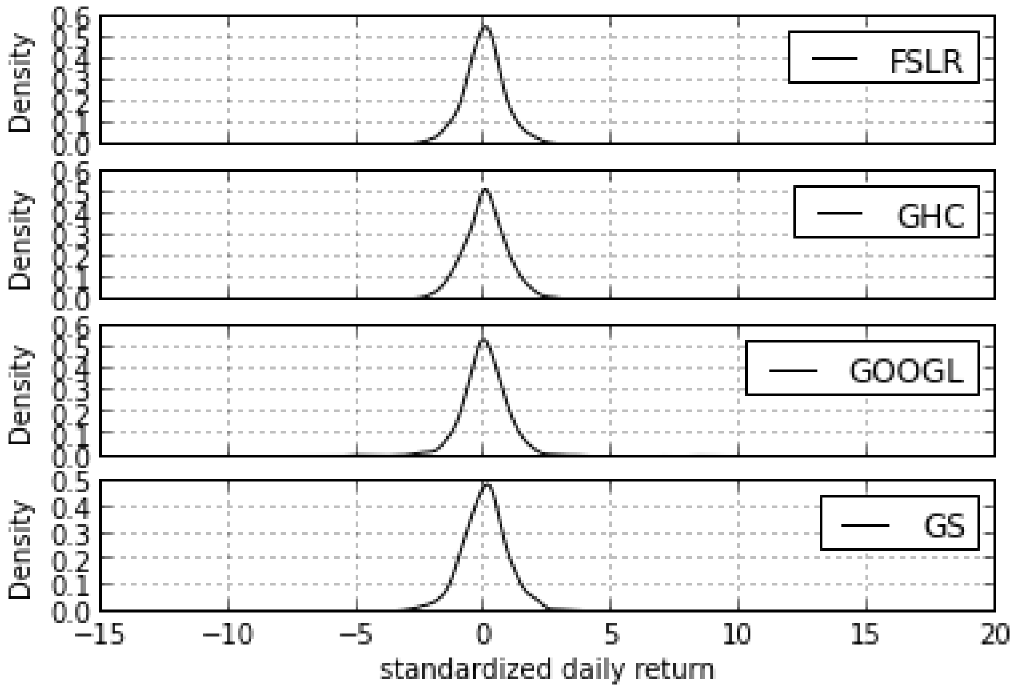


Figure 16: density plots for companies selected by LASSO

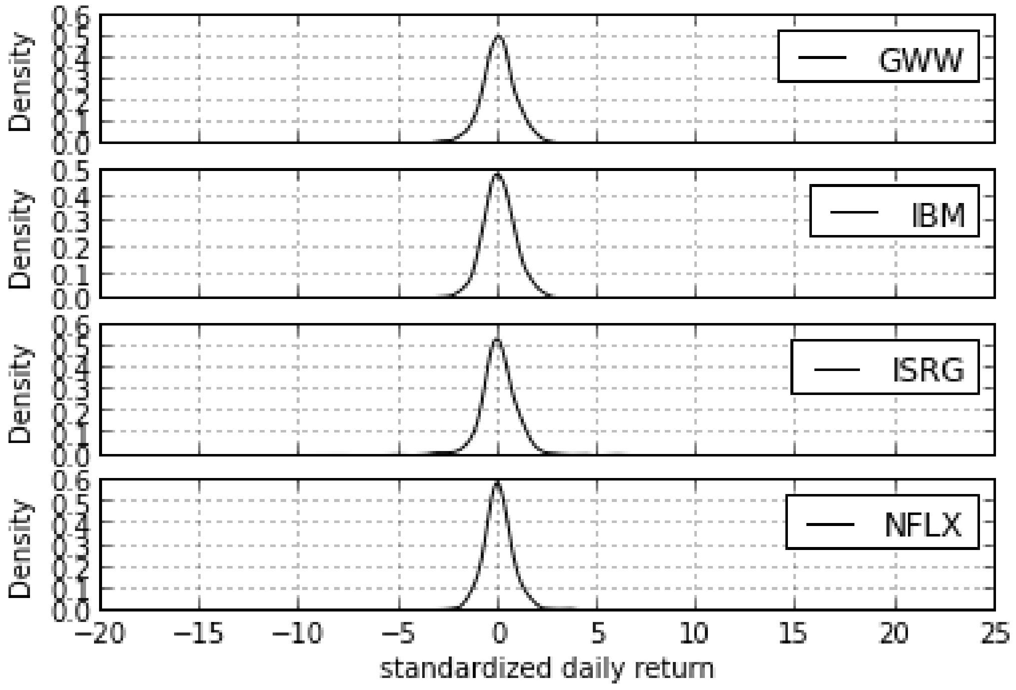
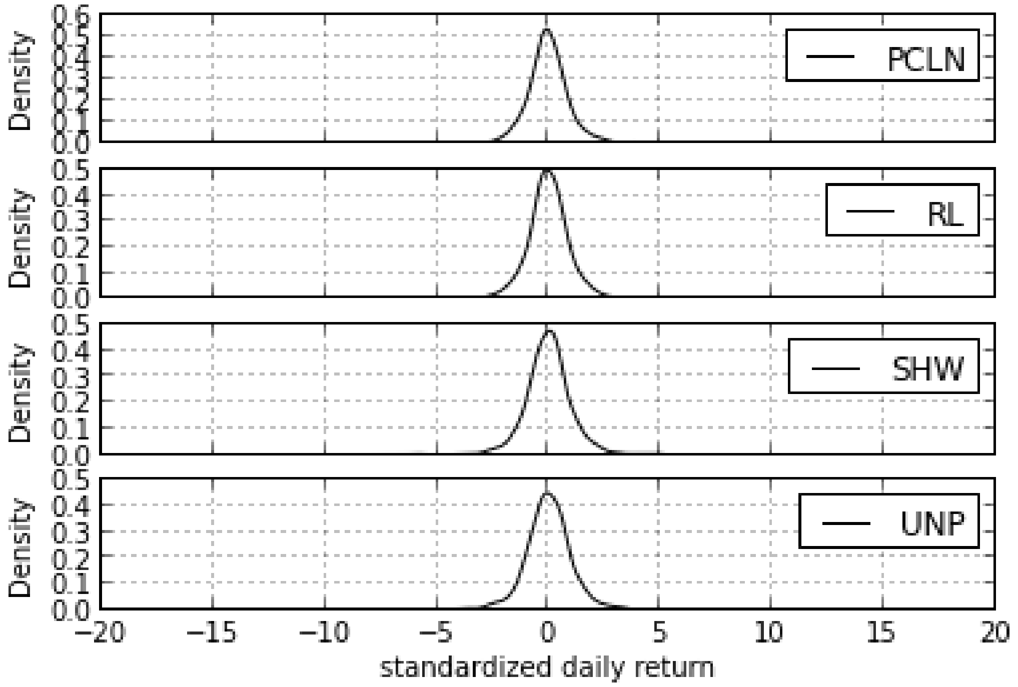


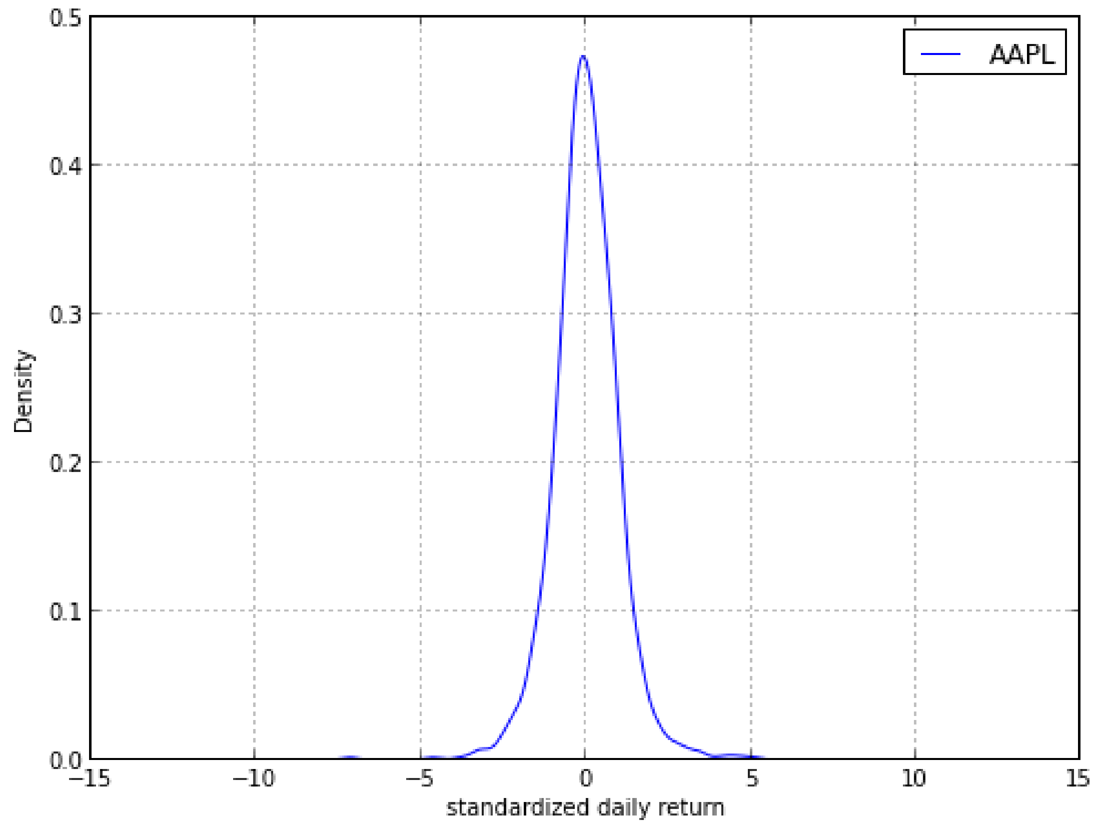
Figure 17: density plots for companies selected by LASSO



The bell-shaped distribution can be seen more clearly when we chose a small subset of the companies. So the daily return density plots of those stocks selected by LASSO have been plotted as the representatives. Figure 14-18 show their density plots and it is apparent that they are all bell-shaped with different variance. APPLE is plotted separately in figure 18 as we will make some inference for it in the next step.

```
<matplotlib.figure.Figure at 0x598b7d0>
```

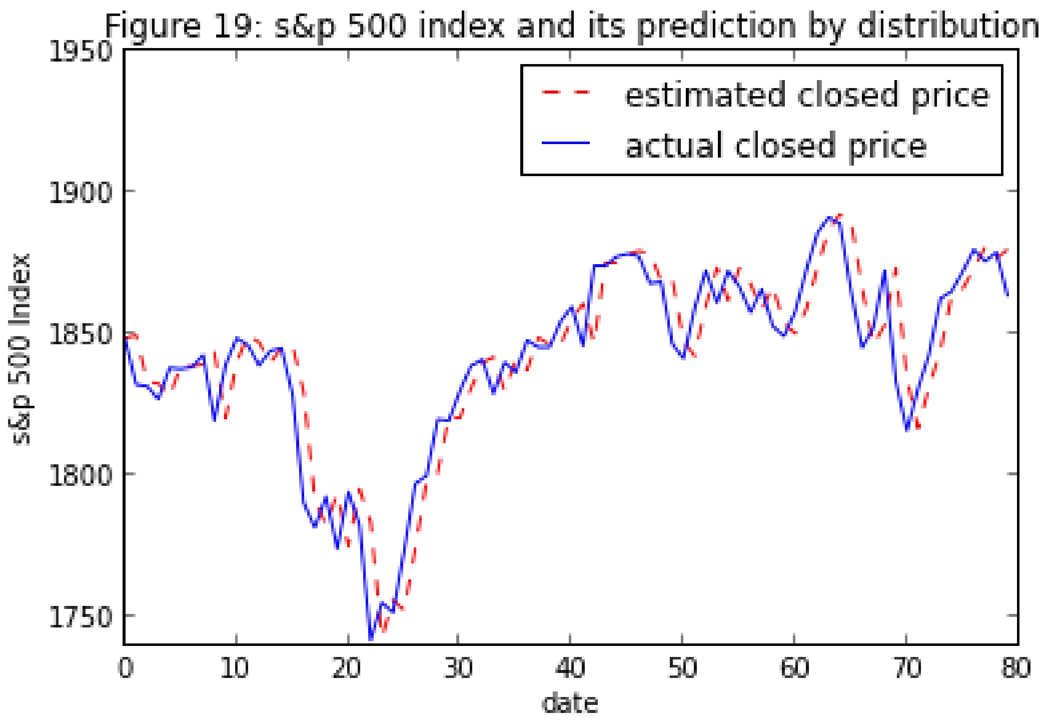
Figure 18: density plots for APPLE



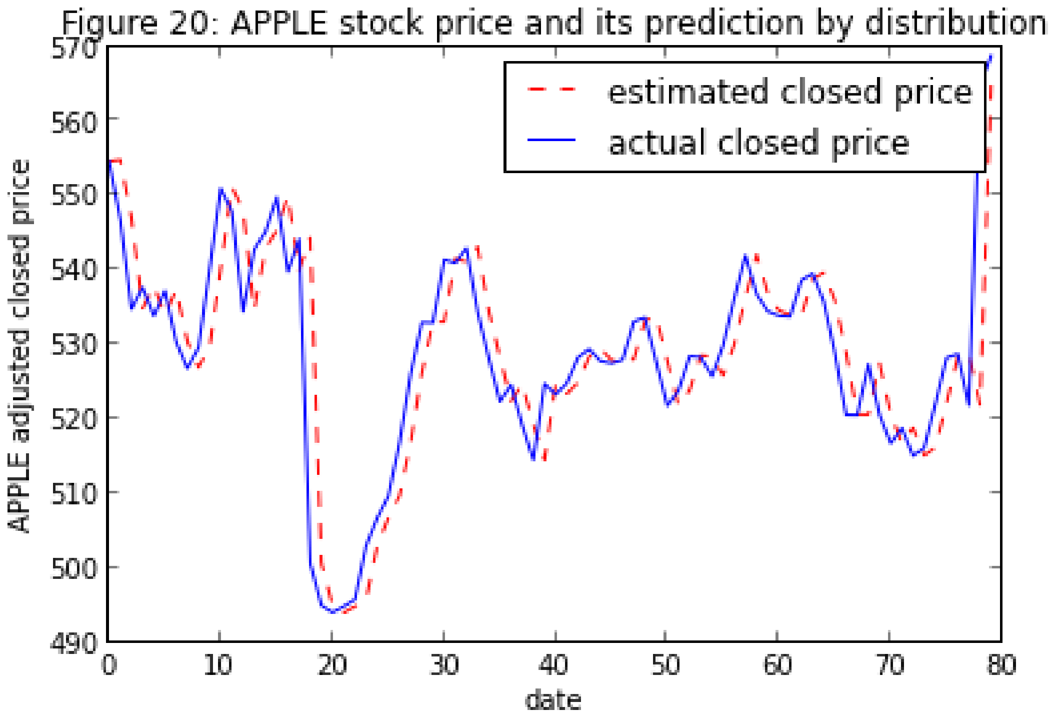
All of these density plots look symmetrical on each side of the vertical line where x equals mean daily return. The standard deviation seems to be very small and we use the mean daily return from the historical data to make a guess for the future daily return. For example, the adjusted closed value on March 14 for APPLE company is 524.69 and we use this mean daily return to predict the closed price for March 17. The prediction is 524.69 * (1 + 0:104%) = 525.24. The actual adjusted closed price is 526.74 and then the ralative error is calculated as (prediction - actual)/actual and the relative error for this prediction is:

```
Out [34]: -0.0027620174058040317
```

The relative error is small for the prediction for one day. In the following we use all the data for the S&P 500 index in 2014 as the testing data. The same method has been applied on S&P 500 index to make prediction and figure 17 shows the estimated and actual prices together.

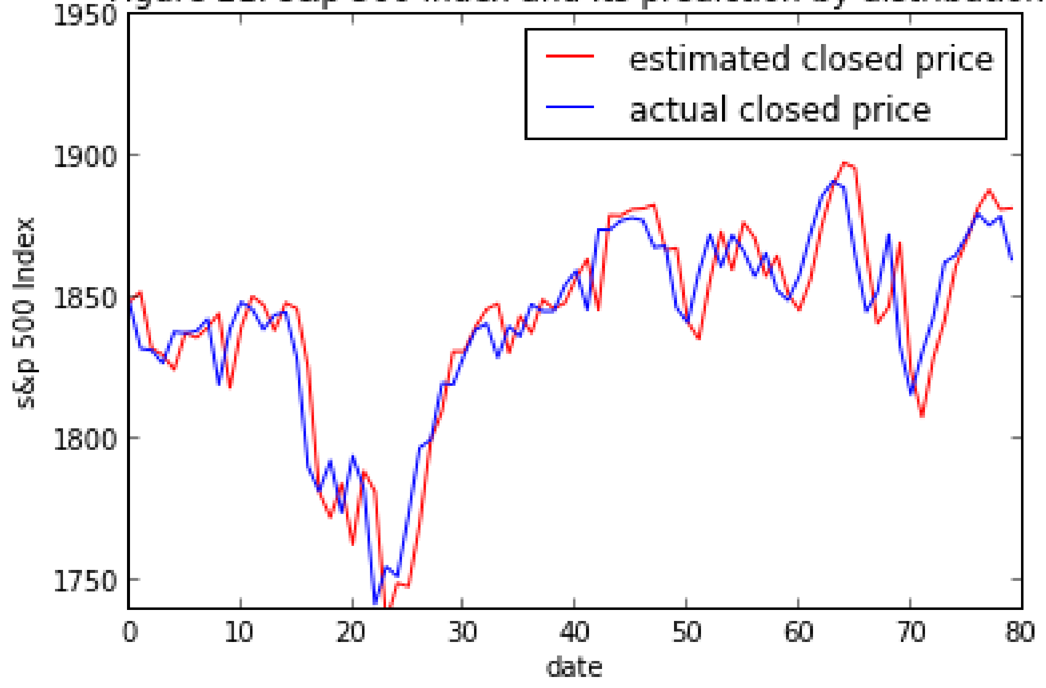


The two lines seem to be very close to each other. We also test our prediction on APPLE data in Figure 20 and it shows the same pattern. Each prediction is almost the same as the price in the previous date. It is understandable since the mean daily return is very close to 0 and thus prediction does not deviates too much from the value in the previous date. Therefore, this method turns out to be a not very accurate prediction.

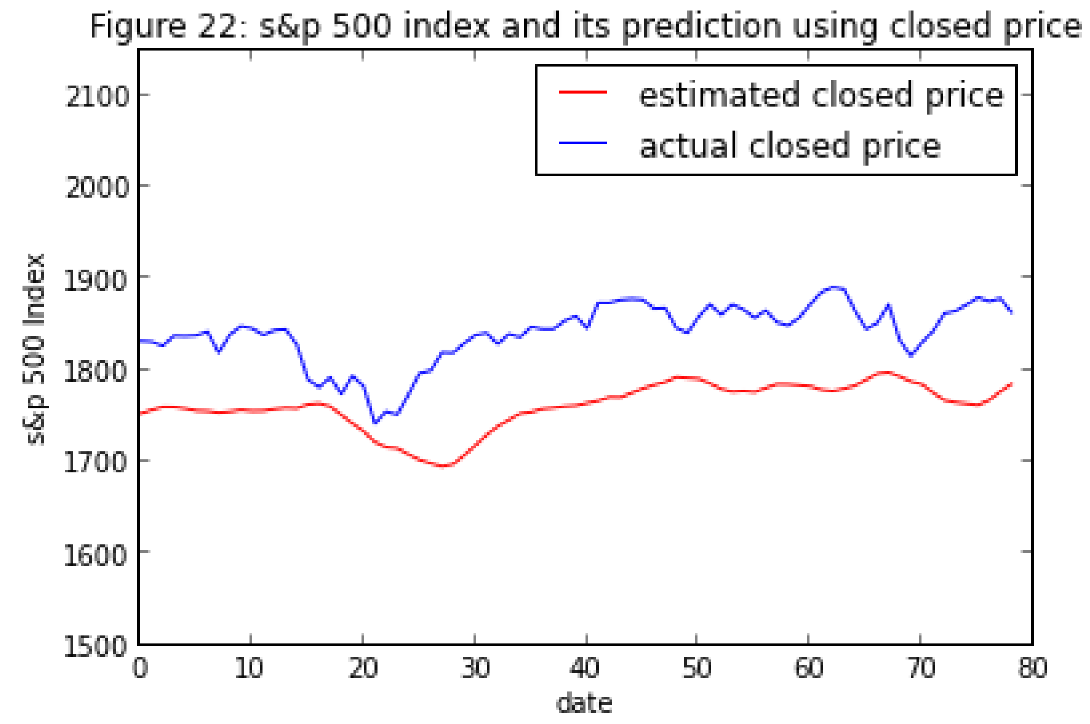


An idea to improve the result is to use posterior mean of daily return by bayesian method instead of the empirical mean daily return. More detailedly, we use mu∼normal(mu_1, sigma) as the prior distribution of daily return mean. This mu_1 could be the average of the daily return from the historical data until one week ago. Then we use the data from the recent week to calculate f(Y|mu). Then the kernel of posterior density is f(mu|Y)∼f(mu)*f(Y|mu). The we use the mean of f(mu|Y) as the daily return prediction for tomorrow. We guess this might improve the prediction to some extent. We applied this idea on daily return and the result from figure 21 indicates that it does not improve much since the daily return are all very small. So the prediction is apparently near the value in the previous date.

Figure 21: s&p 500 index and its prediction by distribution

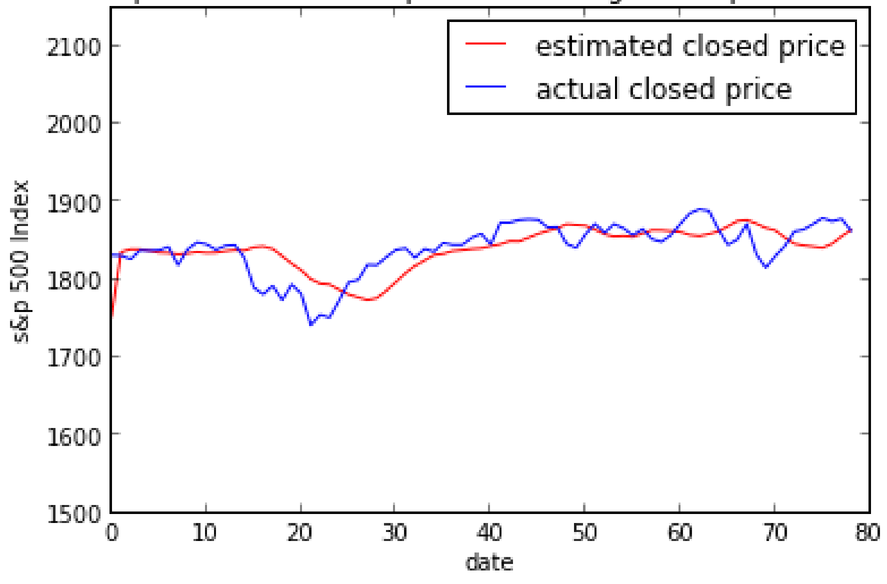


To make further improvement, we decide to work on the closed price directly instead of daily return. Bayesian method has been applied on the S&P 500 index and the estimation is shown in Figure 22.



The result seems to have improved using closed price. It draws our attention that there seems to be a constant difference between estimated price and the actual price. This inspired us to bias correction. We added the bias of estimation for Jan 1, 2014 to the following estimates as the prediction in Figure 23.

Figure 23: s&p 500 index and its prediction using closed price with adjustment



The prediction has been much more accurate as it takes the time effect into account. The result also suggests that bayesian method is promising in terms of forecasting the S&P 500 index.

Part VI

Time Series Model for Forecasting

1 Brief Introduction

Investors have an easier way to keep tracking the performance of S&P 500 Index since we get the most representative seventeen stocks out of S&P 500 by doing LASSO. Then how to predict future price in order to make a profitable investment becames the most important thing for stock investors. In finance, researchers always put a lot of interests in modeling and forecasting volatility of asset(stock) returns. The reason is that the volatility of returns can be seen as a measurement of the risk for investment and provides essential information for the investors to make the correct decisions. While the stock returns themselves are uncorrelated or nearly uncorrelated, however, there exists high order dependence within log return series. We try to accomplish this goal and offer a accurate prediction model by fitting a most suitable time series model based on historical data and then forecasting.

2 Model and Assumption

A time series is a sequence of observations in chronological order. Stock price, obviously, is a time series process since it has time effect. One of the most useful methods for modeling time series data is stationarity assumption. A process is said to be stationary if all aspects of its behaviors are unchanged by shifts in time.

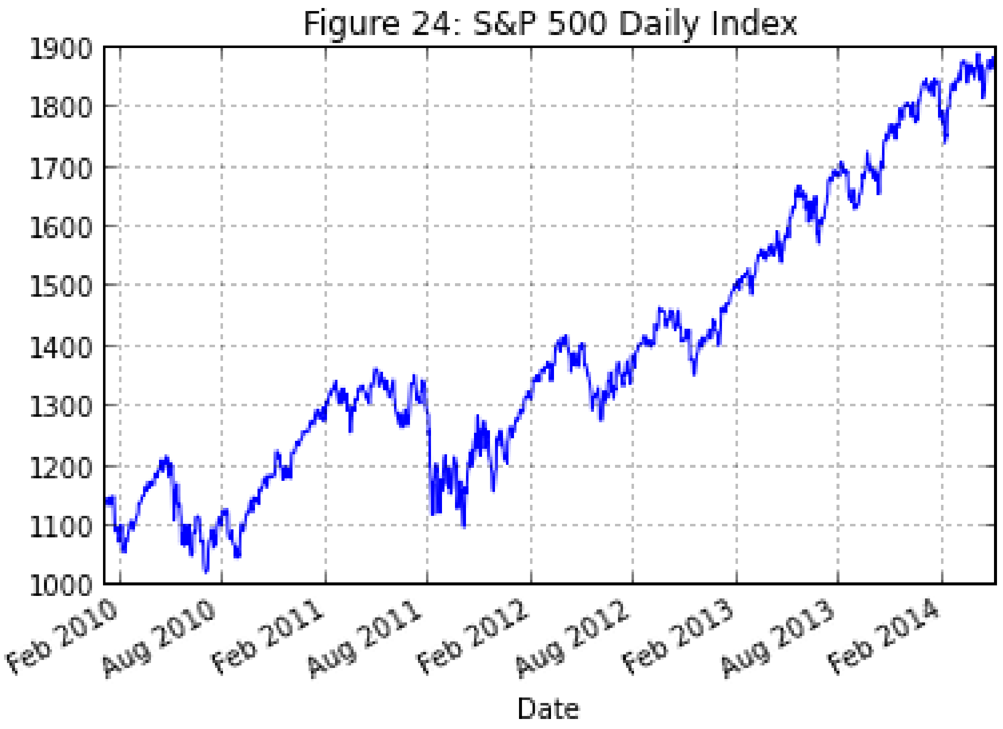


Figure 24: Plot the historical daily index from January 04, 2010 till now. The plot clearly shows that S&P 500 Index is a time series and it seems that the historical daily index is a stationary process.By literature review, we find the ARMA Model is fitted to stationary time series process to better understand the data and to predict future points in the series. ARMA is a combination of two processes — Autoregressive (AR) and Moving Average (MA). AR process is that Y_t (stock price) is modeled as a weighted average of past observations plus a white noise error, which is also called the "noise" or "disturbance." Here is a formal defination. The stationary process Y_t is an AR(p) process if

$$Y_t = \beta_0 + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} + \epsilon_t,$$

where $\epsilon_1, \ldots, \epsilon_n$ is MN(0,σ_ϵ^2), $\beta_0 = \{1 - (\phi_1 + \ldots + \phi_p)\}\mu$, the parameter β_0 is called the "constant" or "intercept" as in an AR(1) model.

MA process is where$Y_t$can be expressed as a weighted average (moving average) of the past values of the white noise process:

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q},$$

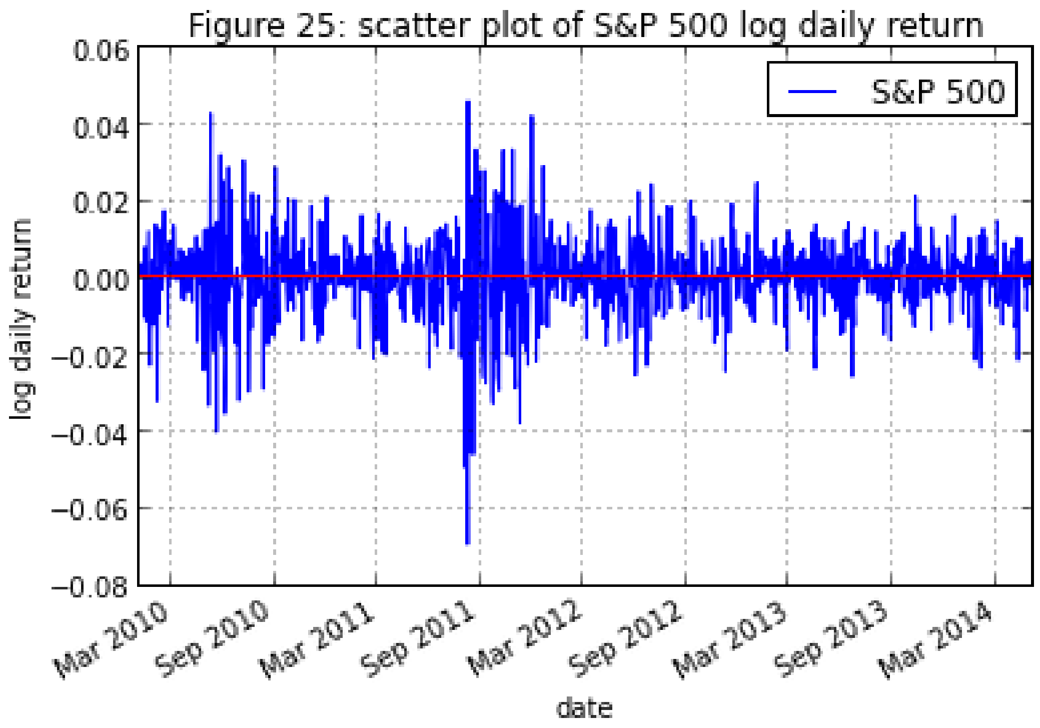
whereϵ_1,. . . ,$\epsilon_n$is MN(0,σ_ϵ^2)

The ARMA models are stationary and ergodic if the roots of the characteristic equation $\phi(z)$=0 lies outside the complex uni circle. The regression formulation for general ARMA(p,q) model in mean-adjusted form is given by :

$$Y_t = c + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} + \epsilon_t + \epsilon_{t-1}\theta_1 + \ldots + \epsilon_{t-q}\theta_q,$$

ARMA (p,q) models are often an aggregation of several simple time series models. If$Y_{1t}$is an ARMA (p_1,q_1) process and$Y_{2t}$is an ARMA (p_2,q_2) process, which may be contemporaneously correlated withY_{1t}, then$Y_{1t} + Y_{2t}$is an ARMA (p,q) process with$p = p_1 + p_2$and$q = max(p_1 + q_2, q_1 + p_2)$. For example, if$Y_{1t}$is an AR(1) process and$Y_{2t}$is a AR(1) process, then$Y_1 + Y_2$is an ARMA(2,1) process. High order ARMA (p,q) models are difficult to identify and estimate in practice and are rarely used in the analysis of financial data. Low order ARMA (p,q) models with$p$and$qThere are two time series datasets — S&P 500 Daily Index and Daily Return over the period Jan 04,2010 through April 2014. Let P_t denoted the successive adjusted closing price pbservation at time t, corresponding transform the price series $\{P_t\}$ into a daily return series $\{R_t\}$ using

$$R_t = ln\frac{P_{t+1}}{P_t} = lnP_{t+1} - lnP_t$$



In Figure 25, obvious volatility(variance) clustering in the returns is indicated that low volatility is followed by low volatility and high volatility is followed by high volatility.Before using this model, we need to test which dataset satisfies the stationarity assumption. There ar two methods to do the test. One is:

```
ADF Test Results:
For Daily Index (0.18345631454536648, 0.97131829426917626)
For Log Returns (-16.965082580439216, 9.2471683722413195e-30)
```

To test for stationary, we use ADF (augmented Dickey–Fuller) test to get the result from which we know that the S&P 500 daily index is non-stationary with great p-values while the log daily returns are stationary with extremely small p-value and it won't cause the nonlinearity in the returns. This indicates that S&P 500 daily return may be a better choice for ARMA model.Another way to test is ACF and PACF. In general, for an ARMA(p,q) process, the ACF behaves like the ACF for an AR(p) process for $p > q$, and the PACF behaves like the PACF for an MA(q) process for $q > p$. Hence, both the ACF and PACF eventually show exponential decay. Let's compute and plot ACF and PACF first.

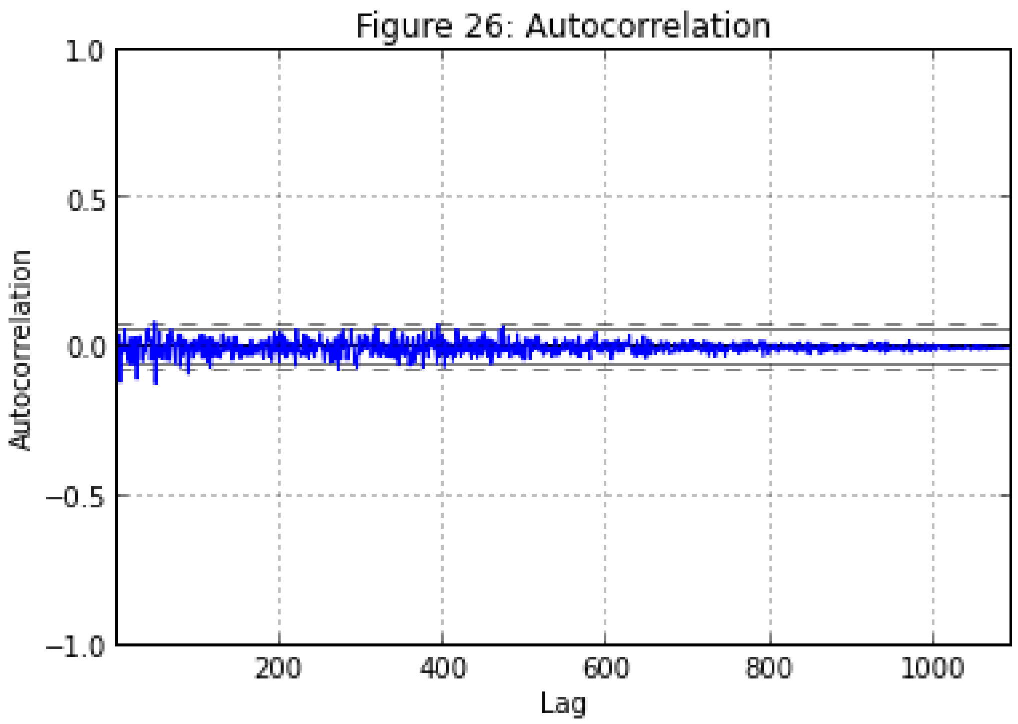


Figure 26 is the plot of ACF under the lags up to 1000 which perfectly demonstrates the convergence of its tail. ACF indicates that log daily returns is a stationary process. The autocorrelation of a random process describes the correlation between values of the process at different times, as a function of the two times or of the time lag. The tail decays quickly and converges to zero. The value of ACF with lags=40 is as follows:

```
         ACF of Log Daily Returns

Out [46]: array([ 1.        , -0.06750278,  0.04507265, -0.08382176,
         0.02258862,
               -0.11468699,  0.01064207, -0.00355592, -0.00670076,
         -0.01604792,
                0.05914849,  0.02272347, -0.00294931, -0.00270207, -0.0456185
         ,
               -0.01919028,  0.01593246,  0.0330026 , -0.05683321,
         -0.03902116,
                0.03074894, -0.02265991, -0.05149063,  0.03233947,
         0.03191587,
               -0.10509912, -0.03624554,  0.0313327 ,  0.03234537,
         0.03791299,
                0.0245974 , -0.02031744, -0.04579217, -0.00763138,
         -0.00191407,
               -0.01795482,  0.00770589,  0.05639455,  0.02015617,
         0.06325546,
                0.02705088])
```

Also here is the value of PACF with lags=40:

```
         PACF of Log Daily Returns

Out [47]: array([  1.00000000e+00,  -6.75645943e-02,   4.07764483e-02,
                  -7.88614547e-02,   1.06882930e-02,  -1.07968611e-01,
                  -1.07670565e-02,   6.58072727e-03,  -2.48734215e-02,
                  -1.56809060e-02,   4.78524793e-02,   2.84415115e-02,
                  -4.79890521e-03,   3.11312143e-04,  -4.76400603e-02,
                  -1.48910010e-02,   2.31091771e-02,   2.87392368e-02,
                  -5.74969120e-02,  -5.48559739e-02,   2.88443082e-02,
                  -2.49217158e-02,  -6.10313491e-02,   1.98887507e-02,
                   3.17007363e-02,  -1.05992614e-01,  -5.73854043e-02,
                   2.08045413e-02,   3.16185562e-02,   5.04942095e-02,
                   9.72148148e-03,  -2.61961689e-02,  -3.83750931e-02,
                  -1.22096142e-02,  -5.61971460e-04,  -9.18006607e-03,
                   5.14305757e-03,   5.13013737e-02,   2.78791391e-02,
                   5.24846843e-02,   1.53605501e-02])
```

Figure 27: Compare ACF & PACF

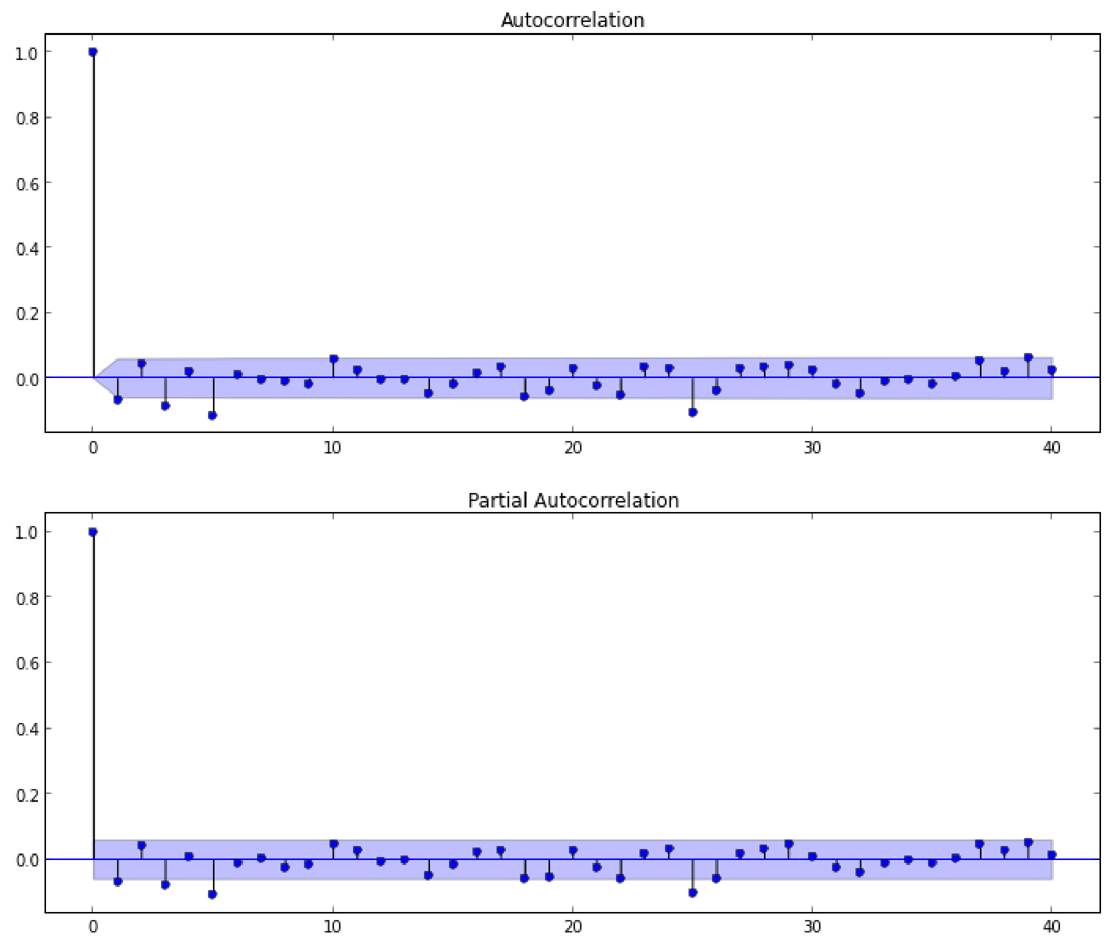


Figure 27 is a explicit comparision of ACF and PACF which shows both ACF and PACF has tails that converge to zero. The PACF is shown with 95% confidence limits about zero. Also there is only one value extreme greater than others, so the parameters of ARMA model should not be greater then 3.

3 Fit ARMA Model and Forecast

ARMA (p, q) models are generally estimated using the technique of maximum likelihood, which is often accomplished by putting ARMA (p, q) in state-space form from which the prediction error decomposition of the log-likelihood function may be constructed. Before an ARMA (p, q) may be estimated for a time series Y_t, the AR and MA orders p and q must be determined by statistical model selection criteria. AIC (Akaike Information Criteria) and BIC (Bayesian Information Criteria) :

$$AIC(p,q) = ln(\overline{\sigma}^2(p,q)) + \frac{2}{T}(p+q)$$

$$BIC(p,q) = ln(\overline{\sigma}^2(p,q)) + \frac{ln(T)}{T}(p+q)$$

The AIC criterion asymptotically overestimates the order with probability greater than zero, however BIC tends to estimate the order consistently under fairly general conditions if the true orders p and q are less than or equal to max(p) and max(q).Based on AIC and BIC criteria, the best p and q for this model are 3. Fit the model and give all parameters and value of AIC and BIC.

```
ARMA(3,3) parameters
const               0.000461
ar.L1.Adj Close    -0.830504
ma.L1.Adj Close     0.765392
dtype: float64
AIC BIC
-6870.39856944 -6850.41184349
```

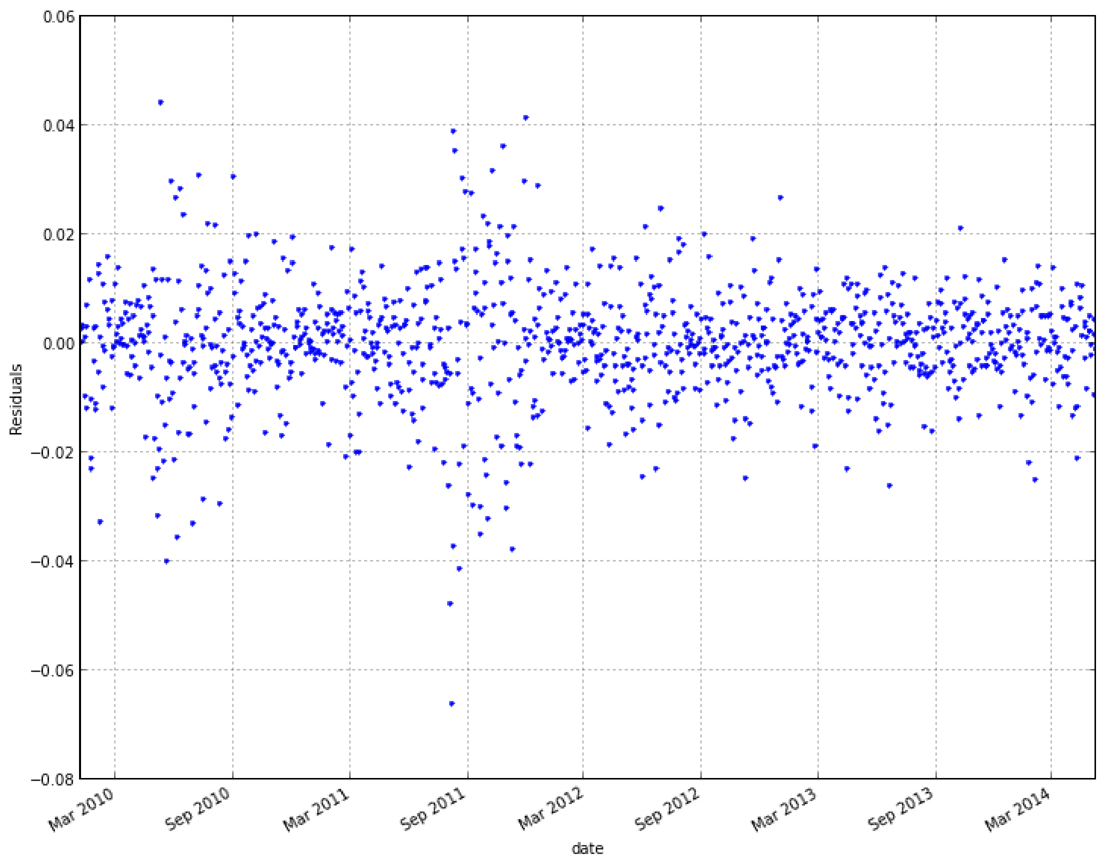
The values of AIC and BIC are quite small. We want to comfirn that ARMA(3,3) has the samllest AIC and BIC among all available parameters less then 4. The value of AIC, BIC with different ARMA model is as follows:

```
  MODEL           AIC             BIC
ARMA(1,1) -6870.39856944 -6850.41184349
ARMA(0,1) -6862.16566612 -6847.17562166
ARMA(0,2) -6861.50640288 -6841.51967692
ARMA(0,3) -6865.77782779 -6840.79442035
ARMA(1,0) -6862.51626652 -6847.52622205
ARMA(2,0) -6862.32610248 -6842.33937652
ARMA(2,1) -6868.57325576 -6843.58984832
ARMA(3,0) -6867.10158733 -6842.11817989
ARMA(3,1) -6868.44338482 -6838.46329589
ARMA(4,0) -6865.22607566 -6835.24598673
ARMA(4,1) -6866.48712527 -6831.51035485
```

Through comparing ARMA models with different p and q, ARMA (1,1) has the smallest AIC and BIC which verifies our model. Also, it seems that AIC and BIC are both increasing along with p and q so there is no need to worry about other parameters.

Does the residuals of our model obey the normality assumption?

Figure 28: Residuals of Model

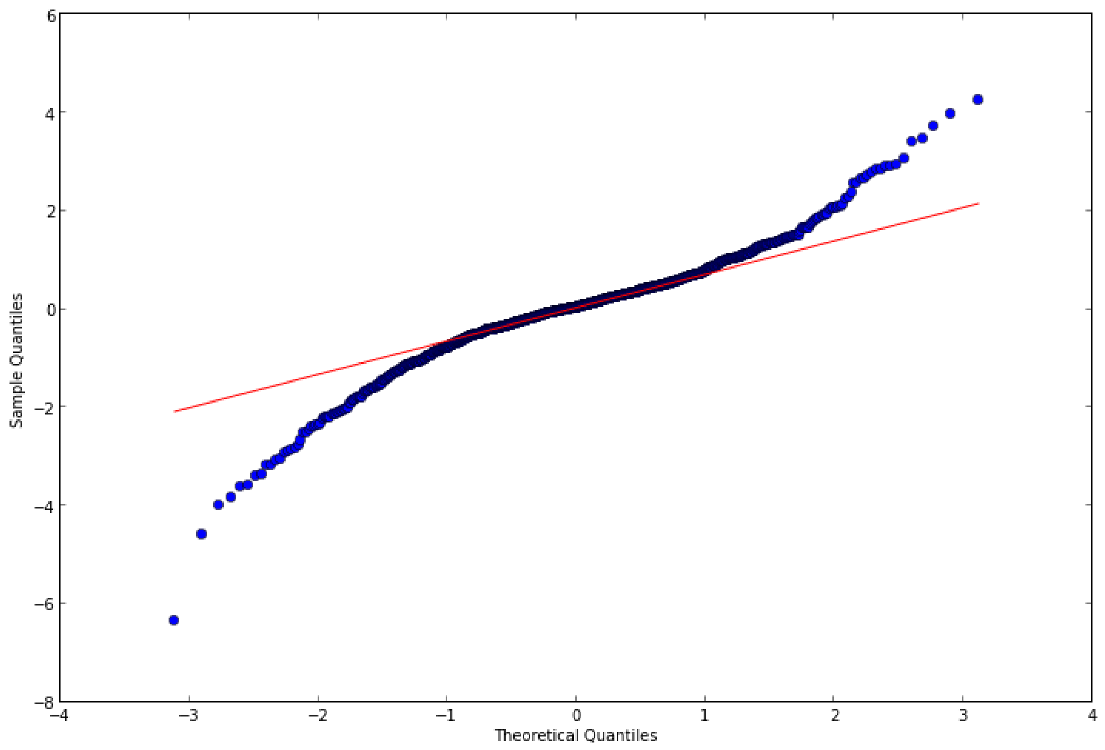


This is residuals of ARMA(1,1), but plot is not enough to confirm its normality.

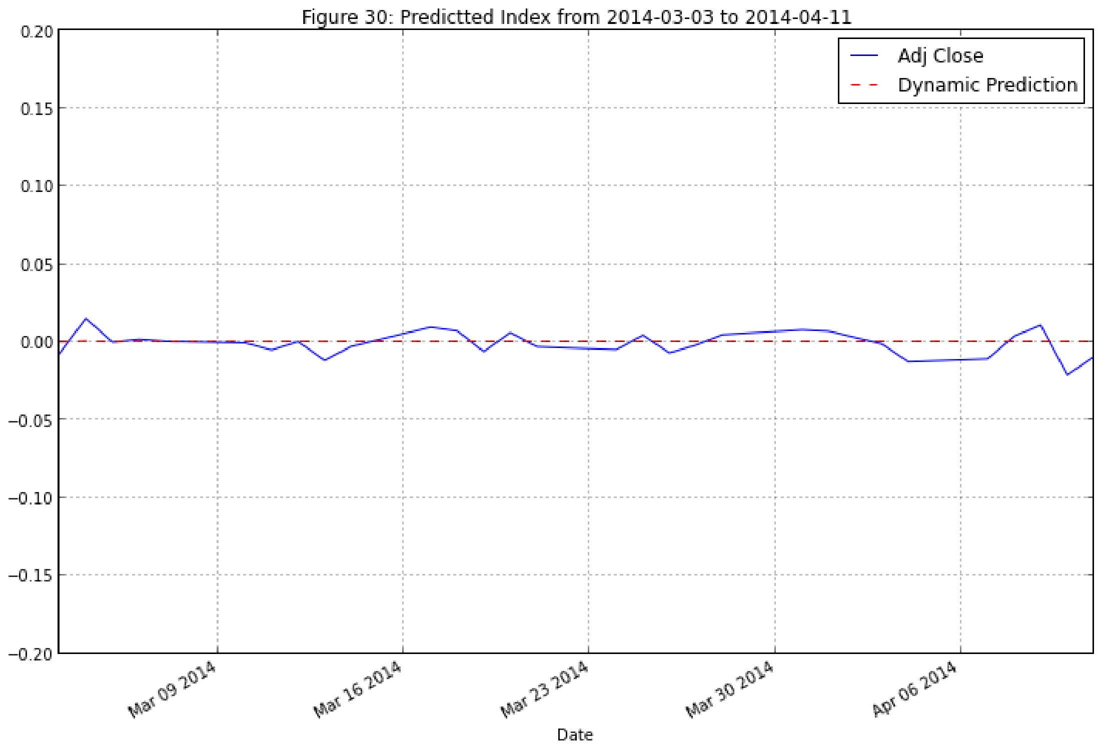
```
Out [53]: (8.4007970143844098e-147, -0.4683089785303386, 6.727453476474913)
```

This is Jarque-Bera test. In statistics, the Jarque–Bera test is a goodness-of-fit test of whether sample data have the skewness and kurtosis matching a normal distribution. If the data comes from a normal distribution, the JB statistic asymptotically has a chi-squared distribution with two degrees of freedom, so the statistic can be used to test the hypothesis that the data are from a normal distribution. The null hypothesis is a joint hypothesis of the skewness being zero and the excess kurtosis being zero. Samples from a normal distribution have an expected skewness of 0 and an expected excess kurtosis of 0 (which is the same as a kurtosis of 3). The first two numbers of JB test for residuals of ARMA(1,1) are skewness and kurtosis. They are close to zero which indicates it is asymptotically normal.

Figure 29: QQ Plot for Normality Test



QQ-plot indicates that the residual of ARMA model can be considered as roughly normal.Forecast according to fitted ARMA Model



In Figure 30, the blue line is the actual log returns and the red line is the predicted ones. We can see the two lines are pretty close to each other within a reasonable error. Log returns is not a good indicator for people to knwo real index.We make a transformation from daily log return to daily index, since it would be more straightforward for investors to know the future trends of S&P 500 Index.

```
Out [58]: [1815.6900000000001,
            1819.6570999854632,
            1817.8965503949494,
            1820.8958555712873,
            1819.9410503789927,
            1822.2724288449897,
            1821.8741090370693,
            1823.7445964051831,
            1823.7306906611009,
            1825.2832258851577]
```

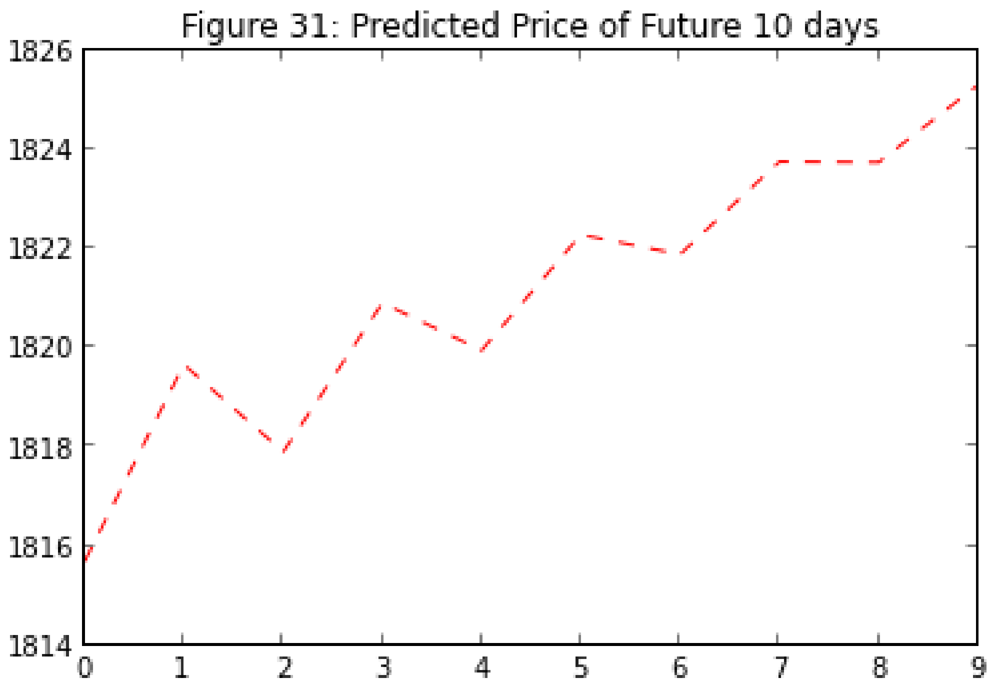


Figure 31 is the predicted index price of next 10 days. The ARMA model tells us in general, S&P 5000 Index will increase significantly in the next 10 days from April 11 but on the $6th$ day there will be a fall. The amazing thing is that it did happened in the real market.

4 Model Assessment

The result of fitting the model and prediction seems good but without cross-validation it is not convincing. Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurate a predictive model will perform in practice.Partition the data into training set (from 2010-01-04 to 2013-12-31) and testing set (from 2013-12-31 to 2014-01-07) to do cross-validation.

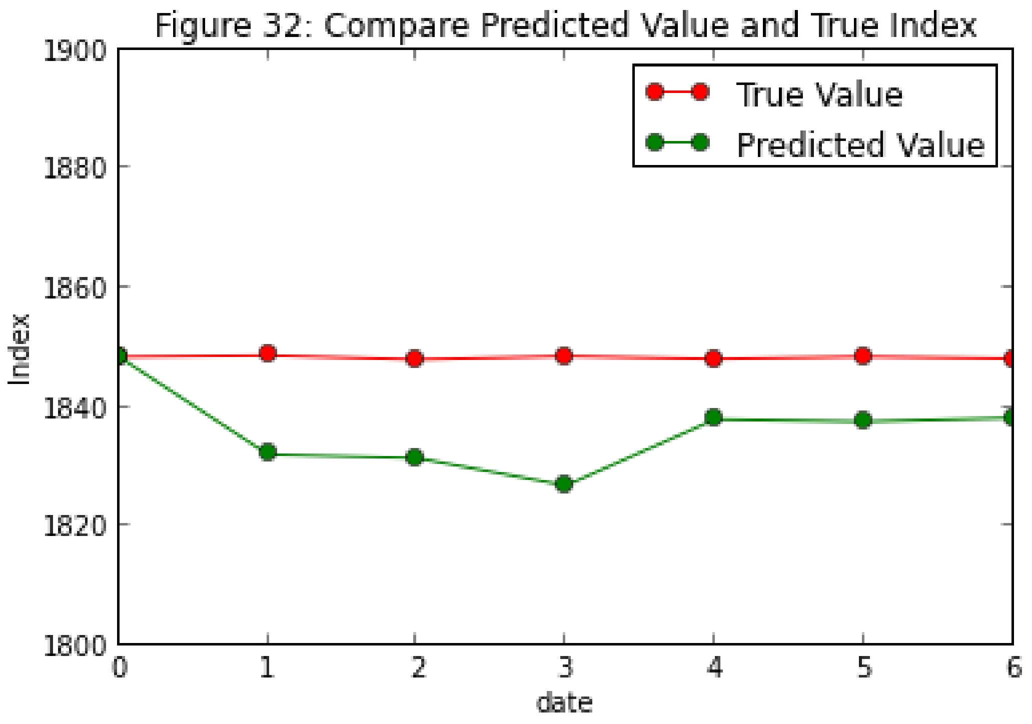
```
test model is ARMA(1,1)
ar.L1.Adj Close   -0.830632
ma.L1.Adj Close    0.762070
dtype: float64
       AIC          BIC
-6275.92225713 -6261.18402866

If test model is ARMA(2,1)
       AIC          BIC
-6274.30140921 -6254.65043793
If test model is ARMA(0,1)
       AIC          BIC
-6266.99534151 -6257.16985587
```

Through compare test model with ARMA(2,1) and ARMA(0,1), ARMA(1,1),obviously has smaller AIC and BIC.

```
Out [65]: [1848.3599999999999,
           1848.5832200204231,
           1847.9123595441097,
           1848.4695802621738,
           1848.0067233615894,
           1848.3911787317513,
           1848.0718323637454]
```

Plot and Compare Predicted Value and True Index



The red line is prediction and green one is true value. The general trend are the same but different on the $2nd$ and rd day within a reasonable error. The result of cross-validation and r-squared is pretty encouraging which indicates that our fitted model is a good fit.

Part VII

Test Suite

We created the following set of 23 testing funcitons to ensure the codes performing as expected. Since we have a automatic way to download massive data, we first check if the data has been read in correctly. The following testing functions are a few examples.

Test if Apple Inc. is included, since it has the largest capital

The rest of the majority testing functions check the technical details of the statistical models. The following examples serve as this purpose.

Randomly select one of the 100 penalty term; test if its corresponding MSE is larger than the one from the penalty term selected (In other words, the penalty term with the smallest MSE was actually selected)

Test if the relative error of predictive adjusted close price for APPLE falls into [-0.05, 0.05]

Test if autocorrelation and partial autocorrelation are close to 0

Test whether the residuals of the model is normally distributed

Test ARMA(1,1) is the best model for verifying model based on data from 2010 to 2013 comparing with ARMA(2,1) and ARMA(0,1)

All the tests are though in 8.658 seconds.

Part VIII

Conclusion

This project provides codes to extract the information about the S&P 500 index and stock prices of the companies for investors and researchers who are interested. For those who follow S&P 500 index to make investment, we suggest a portfolios of a small subset (less than 20 stocks) to invest in order to save tremendously on the fund and commission fee. These stocks count most of the variation of the S&P 500 index and characterize the patterns for S&P 500 index. As the S&P 500 index conveys rich information about all the 500 or so companies, we decomposed the information and fetch these portfolios that could most represent the index by utilizing some machine learning methods.

Further step to understand the S&P 500 index has been made to analyze its daily change in terms of probability and distribution. In this project we suggest that the daily return is well approximated as a bell-shaped symmetrical distribution. The standardized daily return is centered at the mean with small variation. Motivated by this observation, we use the mean of daily return from historical data to predict the return for next date. There is a drawback for this method as it does not take the time effect into account. Since the mean daily return simply takes the average of the overall data, we put equal weight on all of the historical data to make prediction for the feature price. This method may be appropriate for those stocks whose values are more stable in the stock markets and its daily price would have long-term effect on the future price. A Bayesian method using posterior mean instead of empirical mean has been applied to improve the prediction with much more accuracy.

Considering Standard & Poor's 500 Daily Index is a stochastic process, a time series ARMA model has been applied in order to take advantage of the time factor. Since in part 5, it indicates that S&P 500 daily return might not be the best predictor, to overcome the drawbacks, we use log daily return instead of in time series model. The result of forecasting demonstrates the general trend is increasing throughout all time, which further convinces investors that it is viable scheme to follow the index. It further supports our suggestion of the small subset of portfolios to invest on.

Part of the data has been used as the testing data to validate the proposed ARMA model and predicted result turns out to be encouraging. It not only describes the general patterns for the S&P 500 index, but also forecasts the future price closely. With same methodology, our model can be applied to other benchmarks (stock market) such as Dow Jones Index.