Performance Attribution of a Quantitative Investment Portfolio (is607 Final Project)

Prashant B. Bhuyan

December 13, 2014

The purpose of this project is to measure how much of the performance of a diversified quantitative investment portfolio is significantly impacted by random market behavior, if at all. If successful, the results of this analysis will lay the groundwork for a broader analysis pertaining to the separation of alpha and beta across the investment portfolio. If the "luck" portion of the portfolio can be measured dynamically (accounting for lags etc) then a hedging tool could potentially eliminate random market risk without eroding portfolio returns in times of erratic market behavior.

The methodology is to obtain historical performance data from 11 different trading models (mean reversion, pairs, market making, momentum, statistical arbitrage, etc) that together form a diversified investment portfolio over a particularly volatile trading period. I will explore the data by analyzing the distribution of performance across symbols and across time periods to reveal the structure of the performance data and how it relates to and is impacted by market behavior. I will then model the data to measure how much of the performance is explained by the market and market volatility, its clustering tendencies and its correlation to the predictor variables.

Finally, I will interpret the results and reconcile the results with my original hypothesis to determine if it makes sense to continue work to create a hedging instrument for the portfolio.

```r
options(width = 200)

# import requisite libraries
library(RPostgreSQL)
```

```
## Loading required package: DBI
```

```r
library(DBI)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:plyr':
##
##     arrange, desc, failwith, id, mutate, summarise, summarize
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(zoo)
library(timeSeries)
```

```
## Loading required package: timeDate
##
## Attaching package: 'timeSeries'
##
## The following object is masked from 'package:zoo':
##
##     time<-
```

```r
library(stats)
library(tm)
library(wordcloud)
```

```
## Loading required package: Rcpp
## Loading required package: RColorBrewer
```

```r
library(RColorBrewer)
library(bitops)
library(RCurl)
# library(yaml)


# set working directory and path that contains the raw data for 11 trading models.
setwd("~/Desktop/is607FinalProject/rawdata")
path <- "~/Desktop/is607FinalProject/rawdata"

# PARTS 1 & 2: OBTAIN & SCRUB.
# read performance data of 11 diff models into a single data frame
# approximately 75,000 observations of 22 variables
# across 11 unique csv files of differing length.

# connect to postgresql db where we will store our data frame so that we
# can easily extend the observational data when new data is available.
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, dbname = "postgres", host = "localhost", port = 5433,
                 user = "postgres", password = "z8yjcpfw")

# read data from all the files in the directory and store in a single data
# frame.
merged_data <- do.call(rbind,lapply(list.files(path),read.csv))

# create a new data frame containing only important variables for our analysis.
# make sure data types are appropriate for each variable.
pruned_data <-
  data.frame(merged_data$Account.Name, merged_data$Security,
             as.Date(merged_data$Trade.Date, '%m/%d/%Y'),
             as.numeric(merged_data$Realized.PnL.Gross),
             as.numeric(merged_data$Unrealized.Changed..Day.Wise.),
             as.numeric(merged_data$GMV),as.numeric(merged_data$NMV))

# rename columns to simplify references.
colnames(pruned_data) <- c('acct','sym','date','realized','unrealized','gmv','nmv')

# write aggregated data frame to database.
dbWriteTable(con, "pruned_datatb", pruned_data)
```

```
## Warning: table pruned_datatb exists in database: aborting assignTable
```

```
## [1] FALSE
```

```r
# check data was properly loaded to db by running a query and verifying the results.
dbGetQuery(con,
           "select * from pruned_datatb where pruned_datatb.acct =
           'model11' limit 5")
```

```
##   row.names    acct sym       date realized unrealized   gmv    nmv
## 1     24876 model11 AFL 0014-09-02     0.00      -5.46 16724 -16724
## 2     24877 model11 AFL 0014-09-03    46.87       0.00     0      0
## 3     24878 model11 AFL 0014-09-04     0.00       0.00     0      0
## 4     24879 model11 AFL 0014-09-05     0.00       0.00     0      0
## 5     24880 model11 AFL 0014-09-08     0.00       0.00     0      0
```

```r
# add all data to a table for future extensibility.
data_tb <- dbGetQuery(con, "select * from pruned_datatb")

# check data table.
head(data_tb)
```

```
##   row.names   acct sym       date realized unrealized gmv nmv
## 1         1 model1   A 0014-09-02        0          0   0   0
## 2         2 model1   A 0014-09-03        0          0   0   0
## 3         3 model1   A 0014-09-04        0          0   0   0
## 4         4 model1   A 0014-09-05        0          0   0   0
## 5         5 model1   A 0014-09-08        0          0   0   0
## 6         6 model1   A 0014-09-09        0          0   0   0
```

```
# PART 3: EXPLORE DATA

# slice out sym, realized and date, realized into two sep data frame.
sym_realized_datadf <- select(pruned_data,sym,realized)
day_realized_datadf <- select(pruned_data,date,realized)

# manipulate data - first group pnl by symbol and std of pnl by symbol
pnl_dist_symdf <- ddply(sym_realized_datadf,~sym,summarise,pnl=sum(realized),
                        stdev=sd(realized))
head(pnl_dist_symdf)
```

```
##   sym   pnl stdev
## 1   A 259.9 43.27
## 2  AA 655.3 98.71
## 3 AAN   0.0  0.00
## 4 AAP   0.0  0.00
## 5 AAT   0.0  0.00
## 6  AB   0.0  0.00
```

```
tail(pnl_dist_symdf)
```

```
##     sym    pnl  stdev
## 911  GG    0.0   0.00
## 912 LKQ -567.4 152.23
## 913 QLD    0.0   0.00
## 914 RAI  101.4  15.29
## 915 SBS 5019.7 361.56
## 916 QEP -205.5  30.98
```

```
# next group pnl by day and std of pnl by day.

pnl_dist_daydf <- ddply(day_realized_datadf,~date,summarise,pnl=sum(realized),
                        stdev=sd(realized))
head(pnl_dist_daydf)
```

```
##         date    pnl  stdev
## 1 0014-09-02 4923.6  64.10
## 2 0014-09-03 6433.9 100.30
## 3 0014-09-04 4060.0  94.69
## 4 0014-09-05 4359.7 108.76
## 5 0014-09-08 -551.8 157.82
## 6 0014-09-09 2393.8 100.44
```

```
tail(pnl_dist_daydf)
```

```
##          date   pnl stdev
## 39 0014-10-24 11218 379.1
## 40 0014-10-27  3668 202.0
## 41 0014-10-28 11088 114.2
## 42 0014-10-29  1058 230.7
## 43 0014-10-30  9021 351.1
## 44 0014-10-31  3515 323.6
```

```
# write pnl_dist data to db for later extensibility.
dbWriteTable(con, "pnl_dist_symdf_tb", pnl_dist_symdf)
```

```
## Warning: table pnl_dist_symdf_tb exists in database: aborting assignTable
```

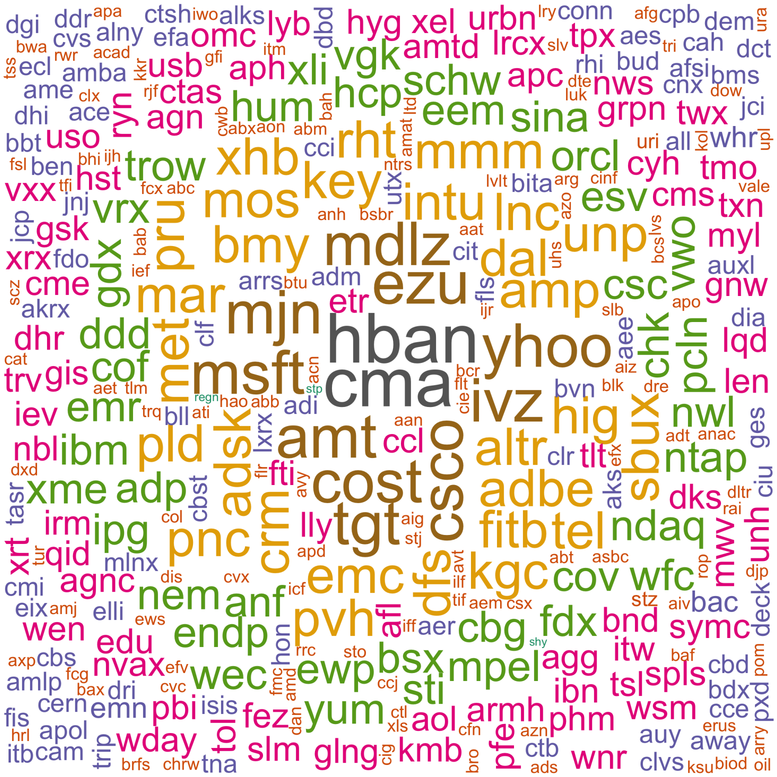
```
## [1] FALSE
```

```
dbWriteTable(con, "pnl_dist_daydf_tb", pnl_dist_daydf)
```

```
## Warning: table pnl_dist_daydf_tb exists in database: aborting assignTable
```

```
## [1] FALSE
```

```
# create a word cloud that represents the frequency of symbols traded in the portf.
# below is the code to create and save the word cloud image which I import below.
#
# sym_freq <- pruned_data$sym
# corp = Corpus(VectorSource(sym_freq))
# word_matrix <- TermDocumentMatrix(corp,control = list(removePunctuation=TRUE))
# matrix <- as.matrix(word_matrix)
# word_freqs <- sort(rowSums(matrix), decreasing = TRUE)
# word_freqdf <- data.frame(word = names(word_freqs),freq = word_freqs)

# wordcloud(word_freqdf$word, word_freqdf$freq, random.order = FALSE, colors = brewer
#.pal(.5,"Dark2"))

#png("SymbolFrequency.png",width = 5, height = 8, units = "in", res = 300)
#wordcloud(word_freqdf$word, word_freqdf$freq, random.order = FALSE, colors = brewer
#.pal(8,"Dark2"))
#dev.off()

library(grid)
library(png)
img <- readPNG("/Users/MicrostrRes/Desktop/is607FinalProject/SymbolFrequency.png")
grid.raster(img)
```



The word cloud above shows that the portfolio of strategies indeed does trade more frequenty in certain symbols as compared to other symbols. For example, hban and cma are very frequently traded as are yhoo, ivz, msft and ezu. On the other hand dgi, hrl, axp are much less frequently traded.

Below I rank the pnl by symbol and interestingly CMA which is the portfolio's most frequently traded symbol is also the 4th worst loss out of 916 symbols traded.

```
# let's see our pnl distribution over the traded symbols and days.
ranked_pnlby_sym <- dbGetQuery(con, "select * from pnl_dist_symdf_tb order by pnl desc")

ranked_pnlby_date <- dbGetQuery(con, "select * from pnl_dist_daydf_tb order by pnl desc")

# rank most profitable symbols.
head(ranked_pnlby_sym)
```

```
##   row.names  sym   pnl  stdev
## 1       171  EWZ 26439 1348.8
## 2       896  GRT 12939 1989.3
## 3       533 FITB 12833  507.4
## 4       488  VXX 11970  377.5
## 5       464    V 10609  456.4
## 6       153  ESV  8948  666.3
```

```
# sum of biggest gaining symbols
sum(head(ranked_pnlby_sym$pnl))
```

```
## [1] 83739
```

```
# rank least profitable symbols.
tail(ranked_pnlby_sym)
```

```
##     row.names  sym    pnl  stdev
## 911       631 URBN  -9687  856.3
## 912       324   NE -13203 1152.0
## 913        85  CMA -14453  581.9
## 914       417  STI -17829  647.0
## 915       892 GLNG -30008 1584.1
## 916       414  SPY -44252 4057.4
```

```
# sum of biggest losing symbols
sum(tail(ranked_pnlby_sym$pnl))
```

```
## [1] -129432
```

```
# rank most profitable days
head(ranked_pnlby_date)
```

```
##   row.names       date   pnl stdev
## 1        11 0014-09-16 23879 337.9
## 2        37 0014-10-22 14497 167.1
## 3        39 0014-10-24 11218 379.1
## 4        41 0014-10-28 11088 114.2
## 5        27 0014-10-08  9287 120.8
## 6        30 0014-10-13  9100 236.7
```

```
# rank least profitable days
tail(ranked_pnlby_date)
```

```
##    row.names       date    pnl stdev
## 39        22 0014-10-01  -4437 242.6
## 40        32 0014-10-15  -6836 479.9
## 41        35 0014-10-20 -14778 217.9
## 42        34 0014-10-17 -25294 404.7
## 43        33 0014-10-16 -45746 562.9
## 44        36 0014-10-21 -51469 918.0
```

```
# sum most profitable days
sum(head(ranked_pnlby_date$pnl))
```

```
## [1] 79070
```

```
# sum least profitable days
sum(tail(ranked_pnlby_date$pnl))
```

```
## [1] -148560
```

```r
# explore data types of pnl-by-sym table.
str(dbGetQuery(con, "select * from pnl_dist_symdf_tb"))
```

```
## 'data.frame':    916 obs. of  4 variables:
##  $ row.names: chr  "1" "2" "3" "4" ...
##  $ sym      : chr  "A" "AA" "AAN" "AAP" ...
##  $ pnl      : num  260 655 0 0 0 ...
##  $ stdev    : num  43.3 98.7 0 0 0 ...
```

```r
# explore the dimensionality of the data.  Here we see that the length of the
# pnl-dist-by-date table spans 44 days.
dim(dbGetQuery(con, "select * from pnl_dist_daydf_tb"))[1]
```

```
## [1] 44
```

```r
# explore data types of pnl-by-date table.
str(dbGetQuery(con, "select * from pnl_dist_daydf_tb"))
```

```
## 'data.frame':    44 obs. of  4 variables:
##  $ row.names: chr  "1" "2" "3" "4" ...
##  $ date     : Date, format: "0014-09-02" "0014-09-03" "0014-09-04" "0014-09-05" ...
##  $ pnl      : num  4924 6434 4060 4360 -552 ...
##  $ stdev    : num  64.1 100.3 94.7 108.8 157.8 ...
```

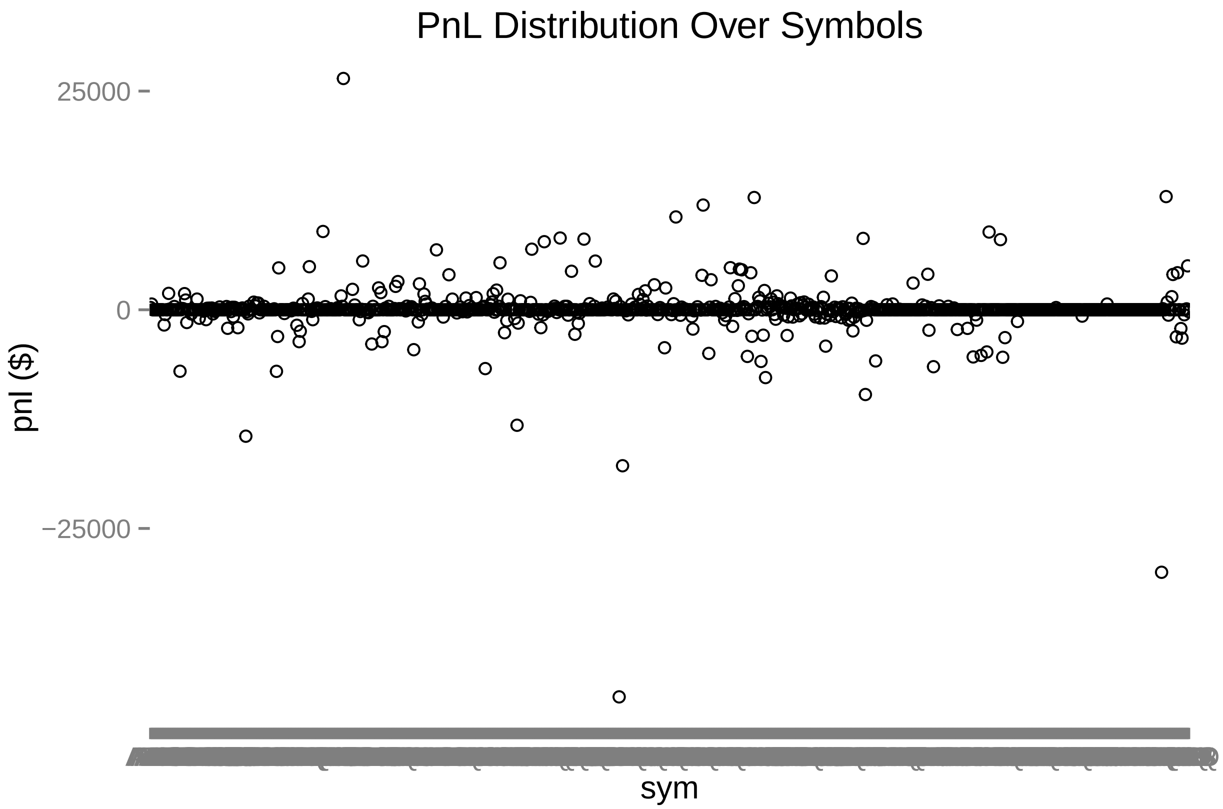
```r
# summarize pnl distribution by symbol.
summary(dbGetQuery(con, "select * from pnl_dist_symdf_tb"))
```

```
##   row.names             sym                 pnl             stdev
##  Length:916         Length:916         Min.   :-44252   Min.   :   0
##  Class :character   Class :character   1st Qu.:     0   1st Qu.:   0
##  Mode  :character   Mode  :character   Median :     0   Median :   0
##                                        Mean   :     7   Mean   :  77
##                                        3rd Qu.:    41   3rd Qu.:  45
##                                        Max.   : 26439   Max.   :4057
```

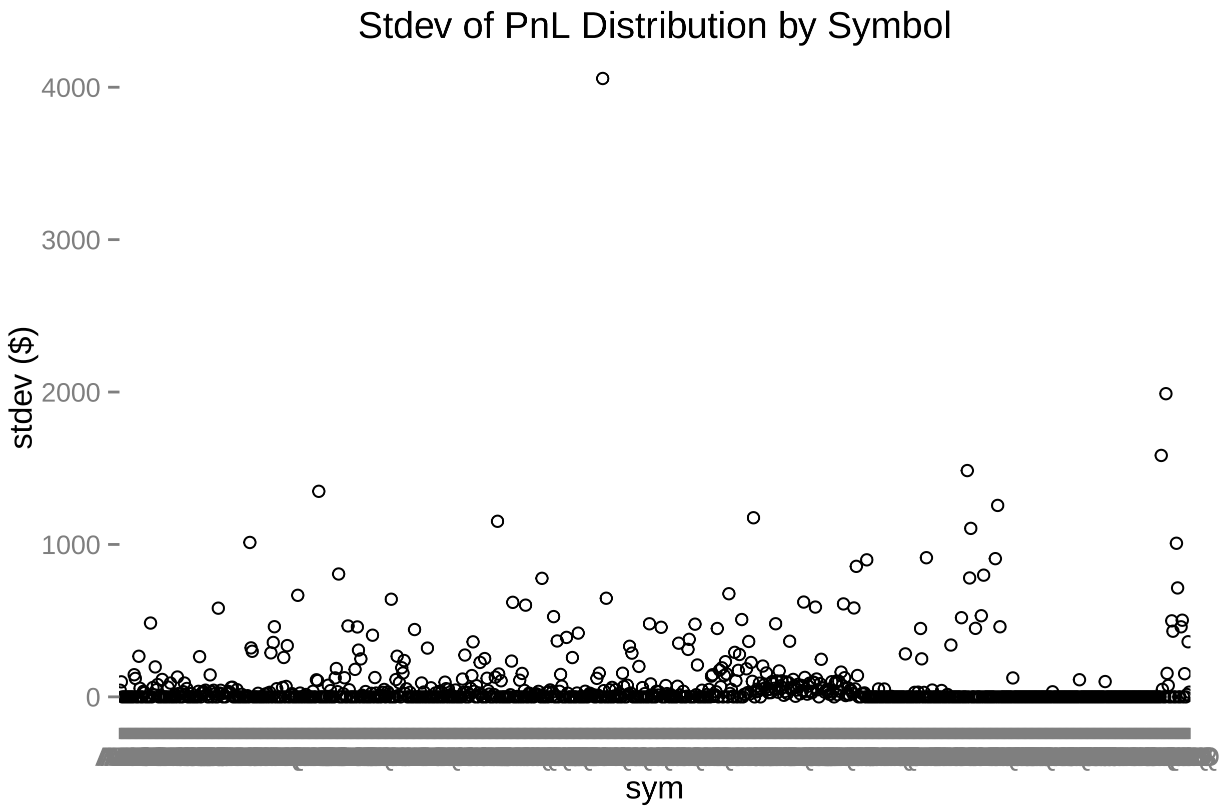
```r
# summarize pnl distribution by day.
summary(dbGetQuery(con, "select * from pnl_dist_daydf_tb"))
```

```
##   row.names              date                 pnl             stdev
##  Length:44          Min.   :0014-09-02   Min.   :-51469   Min.   : 15.5
##  Class :character   1st Qu.:0014-09-16   1st Qu.:  -864   1st Qu.:101.3
##  Mode  :character   Median :0014-10-01   Median :  2386   Median :144.2
##                     Mean   :0014-10-01   Mean   :   136   Mean   :200.2
##                     3rd Qu.:0014-10-16   3rd Qu.:  5944   3rd Qu.:238.2
##                     Max.   :0014-10-31   Max.   : 23879   Max.   :918.0
```

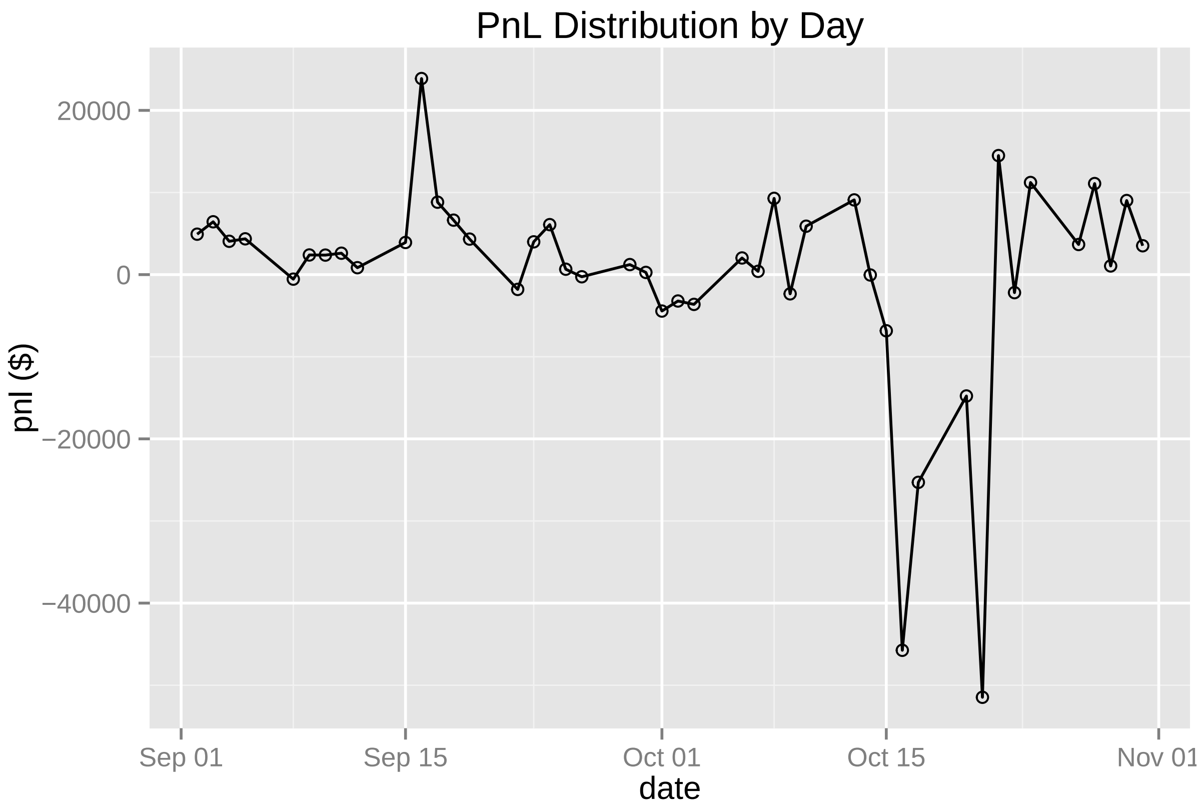
```r
# visualization of pnl-distribution-by-symbols.  it's interseting that 2 symbols
#yielded losses of over -$25,000 while only one symbol yielded a profit of greater
#than $25,000.
ggplot(pnl_dist_symdf, aes(x = pnl_dist_symdf$sym, y = pnl_dist_symdf$pnl))+
  geom_point(shape=1)+ggtitle("PnL Distribution Over Symbols")+xlab("sym")+ylab("pnl ($)")
```



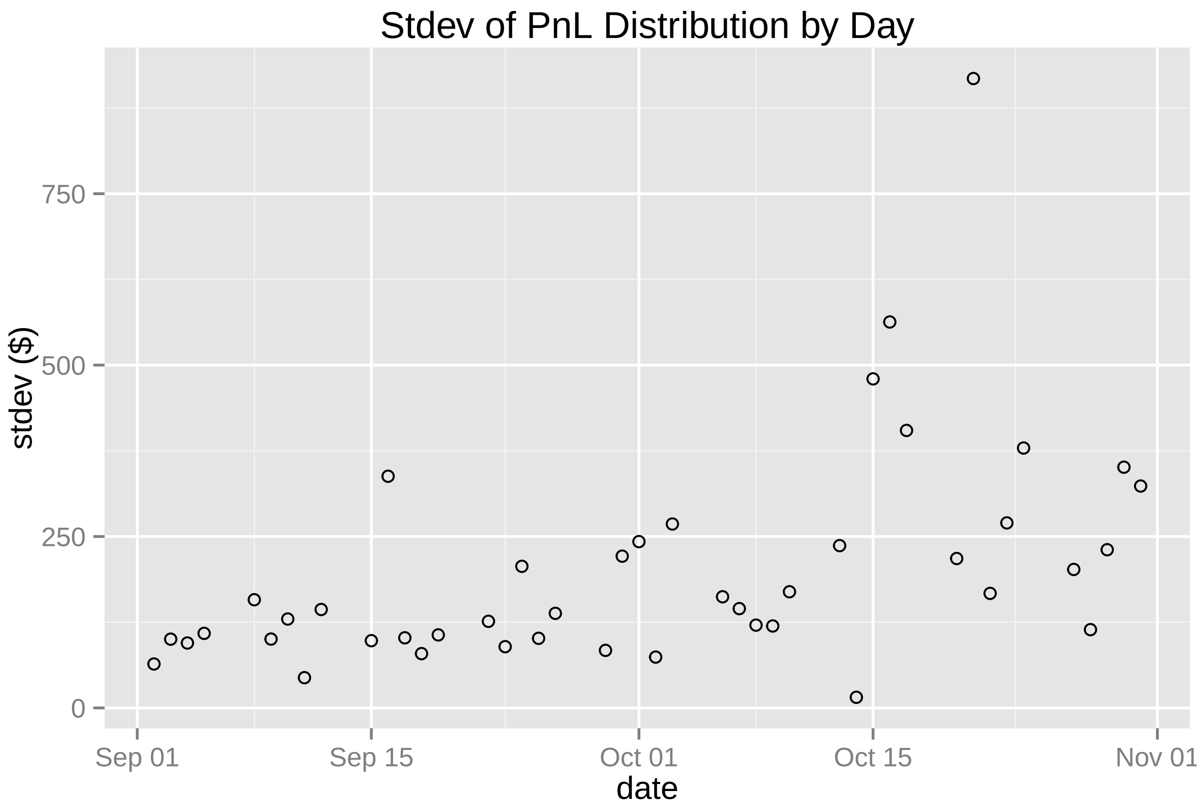
```
# visualization of std of pnl-distribution-by-sym. this is interesting because a hand
#full of stocks are very volatile.  A further area of exploration may be the beta
#composition of those stocks.
ggplot(pnl_dist_symdf, aes(x = pnl_dist_symdf$sym, y = pnl_dist_symdf$stdev))+
  geom_point(shape=1)+ggtitle("Stdev of PnL Distribution by Symbol")+xlab("sym")+
  ylab("stdev ($)")
```



```
# visualizations of pnl-distribution-by-date.  we can see that volatility in the
#distribution of pnl increased in Mid-October 2014.
ggplot(pnl_dist_daydf, aes(x = pnl_dist_daydf$date, y = pnl_dist_daydf$pnl))+
  geom_point(shape=1)+geom_line()+ggtitle("PnL Distribution by Day")+
  xlab("date")+ylab("pnl ($)")
```



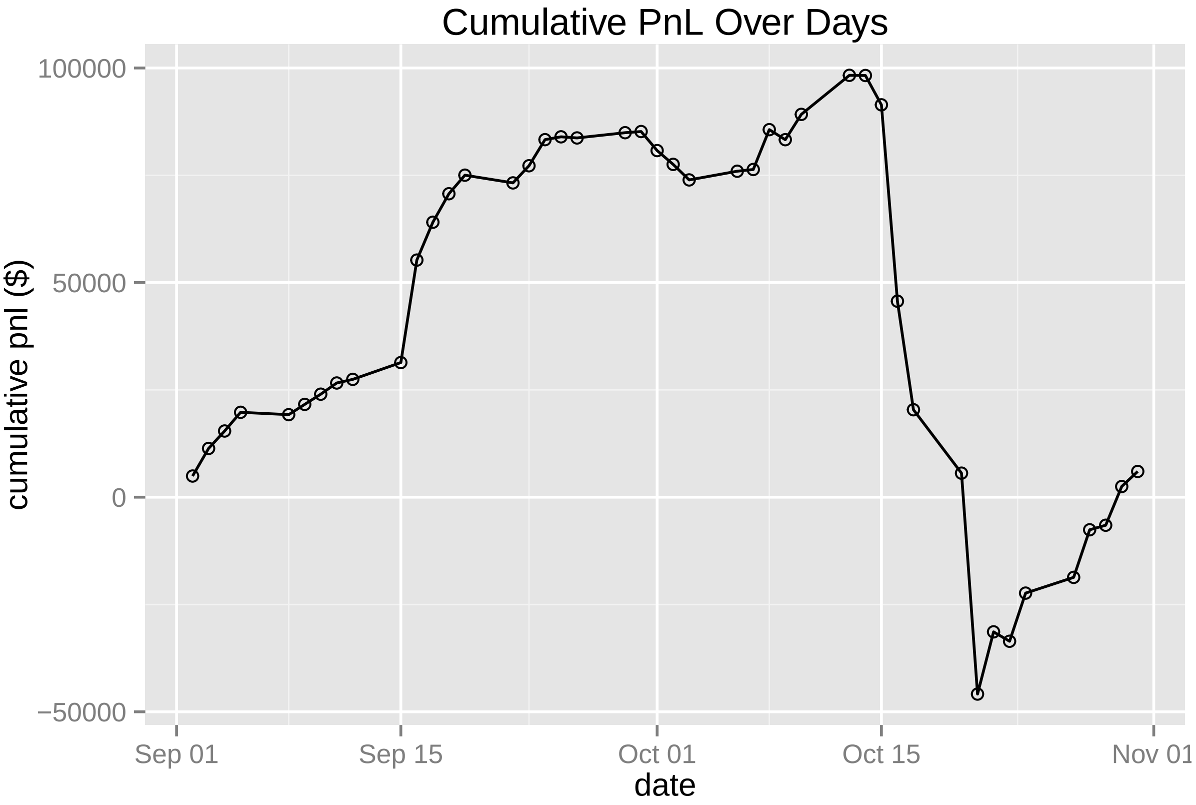
```
# visualization of std dev of pnl by date.  here we see that the volatility of our
#pnl increased substantively on days between october 15 and october 22.
ggplot(pnl_dist_daydf, aes(x = pnl_dist_daydf$date, y = pnl_dist_daydf$stdev))+
  geom_point(shape=1)+ggtitle("Stdev of PnL Distribution by Day")+
  xlab("date")+ylab("stdev ($)")
```



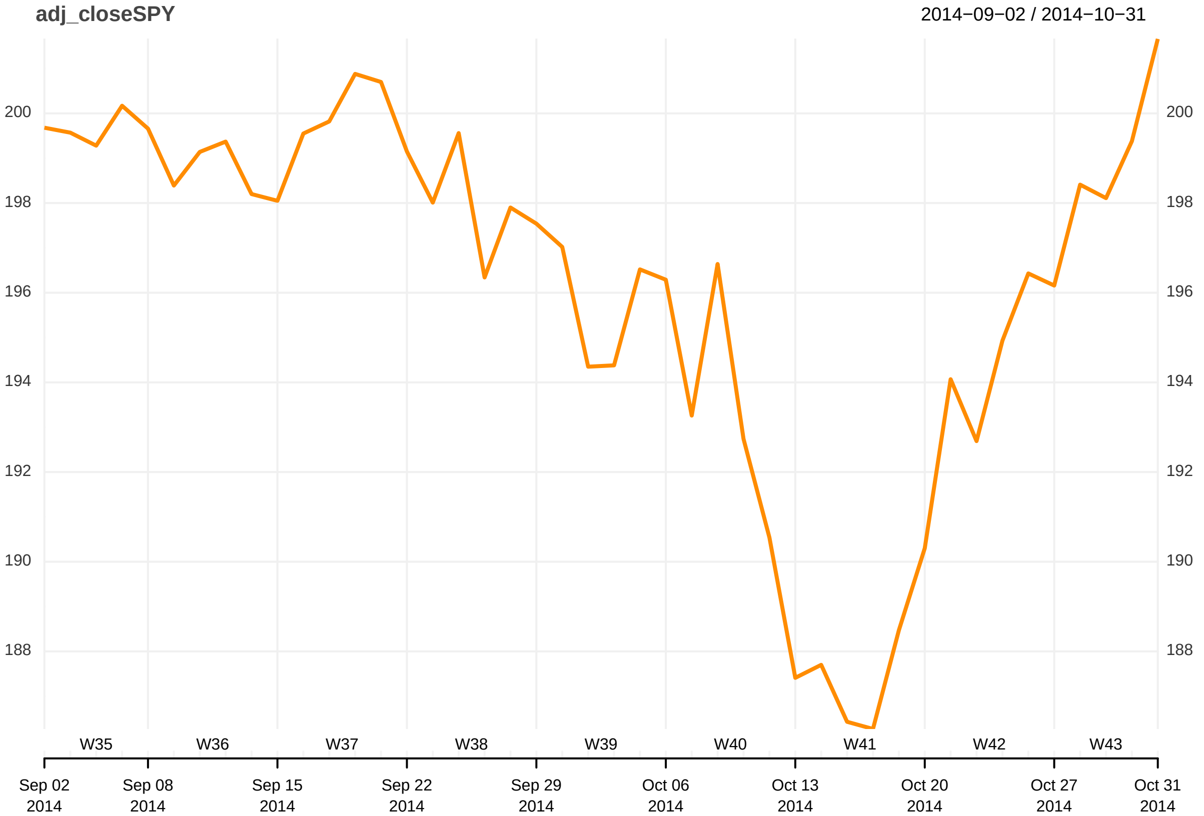
```
# visualization of the portfolio's cumulative pnl over dates.  here we see that a
# steady upward pnl movement was interrupted by a sharp drop in performance in mid
#October.
ggplot(pnl_dist_daydf, aes(x = pnl_dist_daydf$date, y = cumsum(pnl_dist_daydf$pnl)))+
  geom_point(shape=1) + geom_line() +
  ggtitle("Cumulative PnL Over Days") +
  xlab("date") + ylab("cumulative pnl ($)")

# let's read in market data for the SPY and VXX Adjusted Close Prices so that we can
#see how the market impacted our portfolio's performance over the given period.

tickers <- c('SPY', 'VXX')

start <- as.Date("2014-09-01")
end <- as.Date("2014-10-31")

getSymbols(tickers, src = "yahoo", from = start, to = end)
```

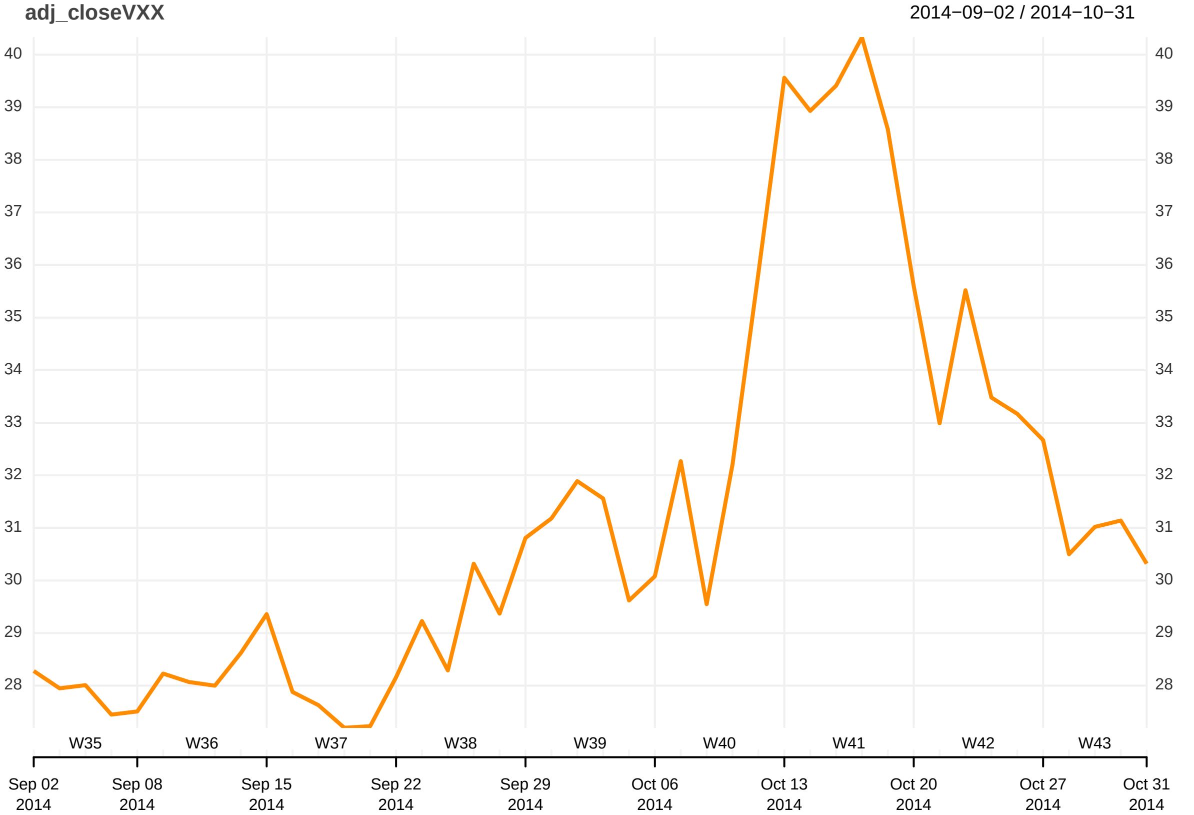
```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for alternate defaults
##
##  This message is shown once per session and may be disabled by setting
##  options("getSymbols.warning4.0"=FALSE). See ?getSymbol for more details
## [1] "SPY" "VXX"
```

```
adj_closeSPY <- SPY$SPY.Adjusted
adj_closeVXX <- VXX$VXX.Adjusted


# visualization of SPY performance which is a proxy for the overall market.
# we see that the sharp drop in performance in the SPY is similar to the sharp
# drop in performance of our portfolio's performance.  The primary question is
# what is the market's impact on the the portfolio's performance and can it be
# backed out via a hedge?  In other words how much of our portfolio's return is
# based on luck (whether the market goes up and down) and how much is based on
# skill in picking stock trades?

chart_Series(adj_closeSPY)
```



```
# visualization of volatility index over the same period.  What is role of market
#volatility in determining the returns of the portfolio?

chart_Series(adj_closeVXX)
```





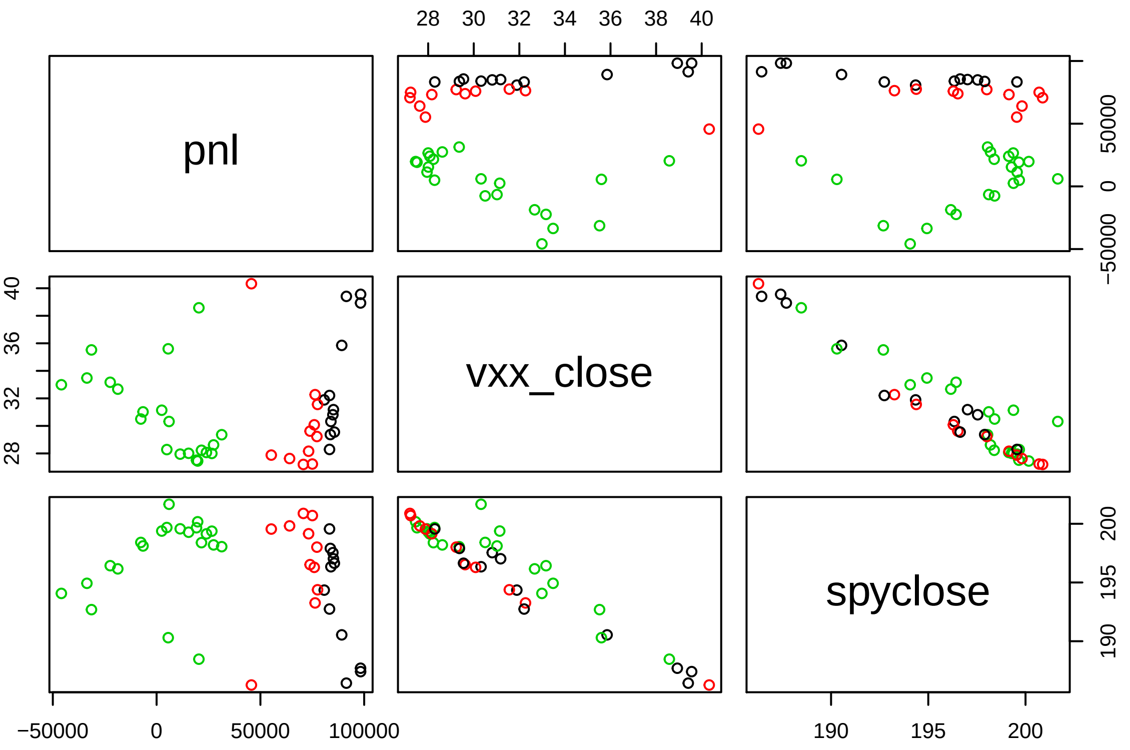
```
# PART4: MODEL
# K-Means Clustering Analysis
# I will use k-means clustering to try and identify any structure in the distribution
# of the pnl of the portfolio versus the market and the volatility of the market.

# First i need to create a new df that contains the portfolio returns, market returns
# and market volatility index returns.
model_data_df_pnldate <- data.frame(cumsum(pnl_dist_daydf$pnl),adj_closeSPY,adj_closeVXX)
colnames(model_data_df_pnldate) <- c('pnl','spyclose','vxx_close')

# check that the data frame was constructed properly.
head(model_data_df_pnldate)
```

```
##              pnl spyclose vxx_close
## 2014-09-02  4924    199.7     28.28
## 2014-09-03 11357    199.6     27.95
## 2014-09-04 15417    199.3     28.01
## 2014-09-05 19777    200.2     27.45
## 2014-09-08 19225    199.7     27.51
## 2014-09-09 21619    198.4     28.23
```

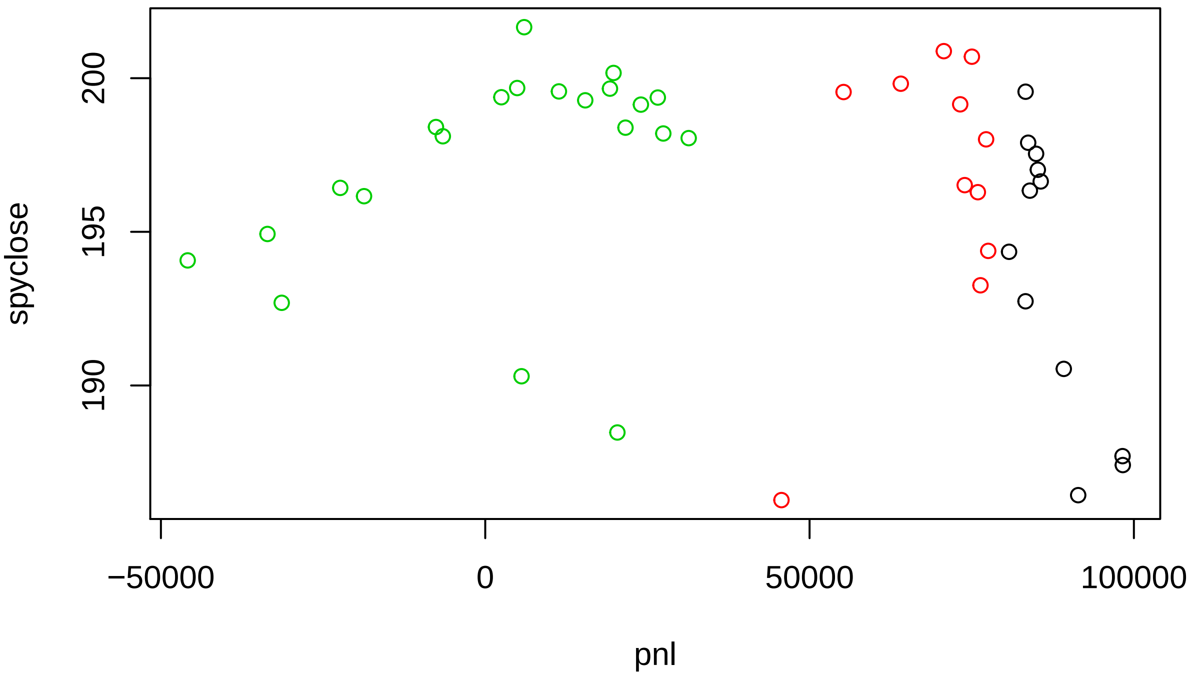
```
# compute results from k-means clustering analysis
results <- kmeans(model_data_df_pnldate,3)

# output results - there are certain dates, particularly from 10.15.2014 - 10.20.2014
# where the portfolio's pnl clusters with market returns and volatility.
results
```

```
## K-means clustering with 3 clusters of sizes 12, 11, 21
##
## Cluster means:
##     pnl spyclose vxx_close
## 1 87325    193.7     33.11
## 2 69535    196.8     30.11
## 3  3342    197.2     30.78
##
## Clustering vector:
## 2014-09-02 2014-09-03 2014-09-04 2014-09-05 2014-09-08 2014-09-09 2014-09-10 2014-09-11 2014-09-12 2
##          3          3          3          3          3          3          3          3          3
## 2014-09-26 2014-09-29 2014-09-30 2014-10-01 2014-10-02 2014-10-03 2014-10-06 2014-10-07 2014-10-08 2
##          1          1          1          1          2          2          2          2          1
## 2014-10-22 2014-10-23 2014-10-24 2014-10-27 2014-10-28 2014-10-29 2014-10-30 2014-10-31
##          3          3          3          3          3          3          3          3
##
## Within cluster sum of squares by cluster:
## [1] 3.730e+08 1.079e+09 1.006e+10
##  (between_SS / total_SS =  84.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
```

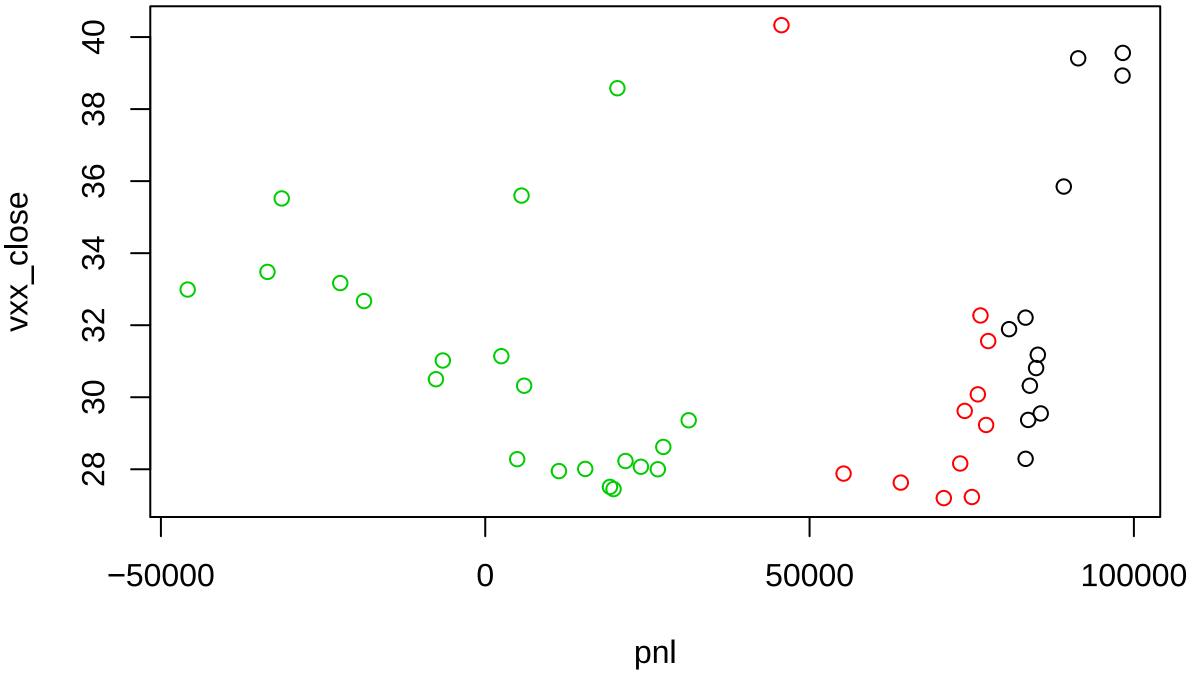
```
plot(model_data_df_pnldate[c('pnl','vxx_close','spyclose')],col=results$cluster)
```



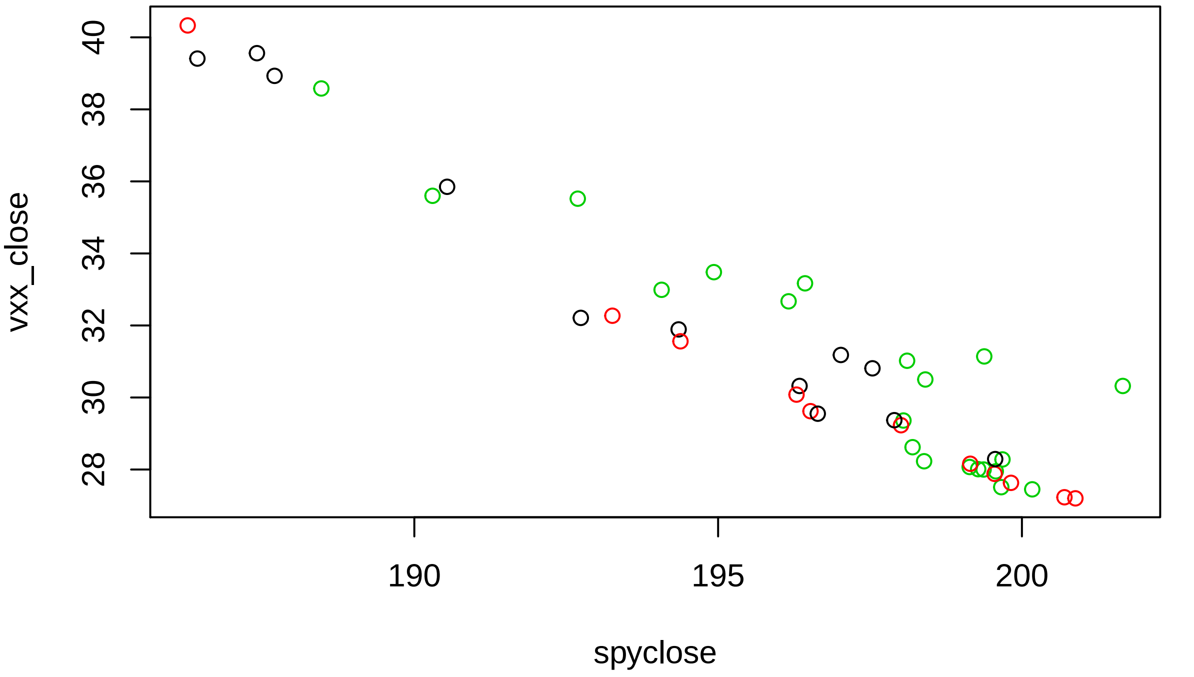
```r
plot(model_data_df_pnldate[c('pnl','spyclose')],col = results$cluster)
```



```r
plot(model_data_df_pnldate[c('pnl','vxx_close')],col = results$cluster)
```



```
plot(model_data_df_pnldate[c('spyclose','vxx_close')],col = results$cluster)
```



```
# Next I'll perform a multi regression analysis where the portfolio pnl is the
# dependent variable and the market and its volatility are the independent variables
# for the purpose of determining what part of portfolio return is explained by the
# independent variables and if both variables are significant or if one or both of
# them should be dropped.

# compute multi linear regression.
linear_model <- lm(model_data_df_pnldate$pnl ~ model_data_df_pnldate$spyclose +
                     model_data_df_pnldate$vxx_close)

# output results from multi-linear regression.  both predictor variables have an
# impact on portfolio pnl with low p values.  More in interpretation section below.
summary(linear_model)
```

```
##
## Call:
## lm(formula = model_data_df_pnldate$pnl ~ model_data_df_pnldate$spyclose +
##     model_data_df_pnldate$vxx_close)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -93519 -26998   5627  28522  57617
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      3885442    1007153    3.86  0.00040 ***
## model_data_df_pnldate$spyclose    -16812       4383   -3.84  0.00042 ***
## model_data_df_pnldate$vxx_close   -17432       4919   -3.54  0.00100 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36900 on 41 degrees of freedom
## Multiple R-squared:  0.266,  Adjusted R-squared:  0.23
## F-statistic: 7.42 on 2 and 41 DF,  p-value: 0.00178
```

```
# run a correlation analysis between pnl vs spy.
cor_model <- cor(model_data_df_pnldate$pnl,model_data_df_pnldate$spyclose,
                 method ="pearson")
cor_model
```

```
## [1] -0.2019
```

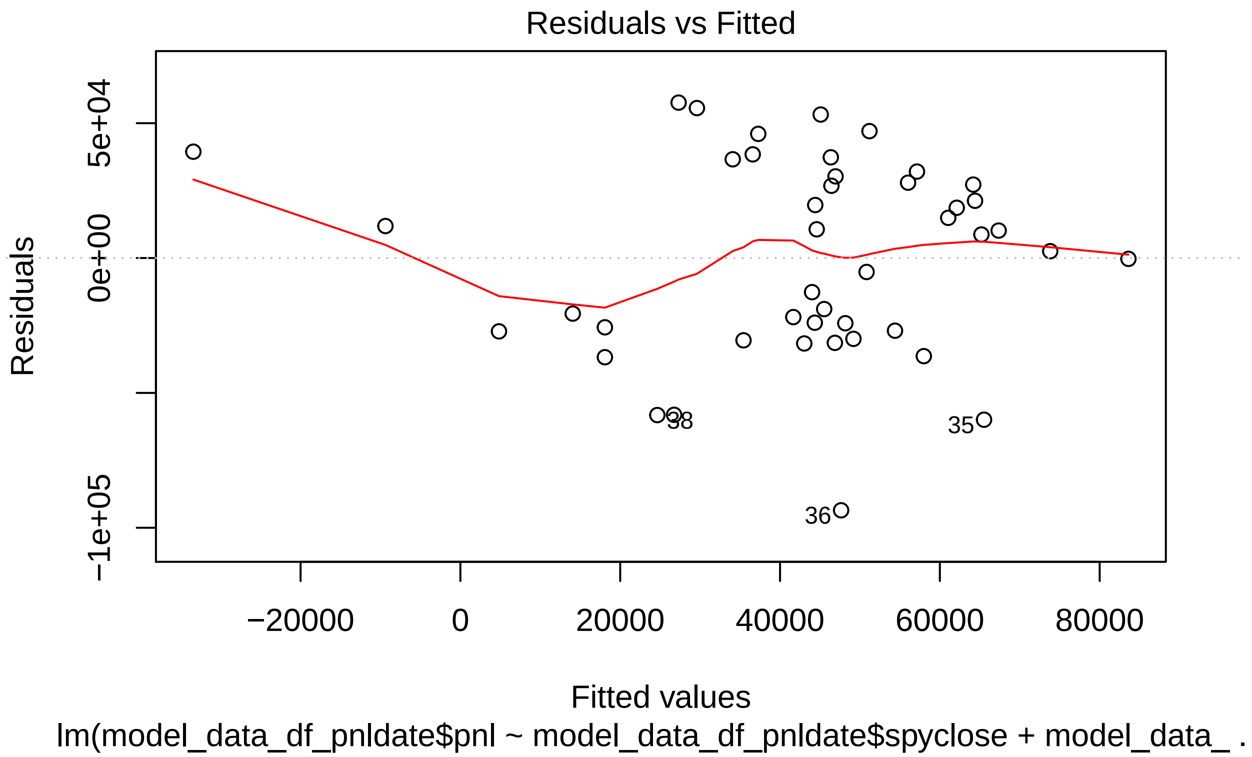
```
# run a correlation analysis between pnl vs vxx.
cor_model2 <- cor(model_data_df_pnldate$pnl,model_data_df_pnldate$vxx_close,
                  method = "pearson")
cor_model2
```

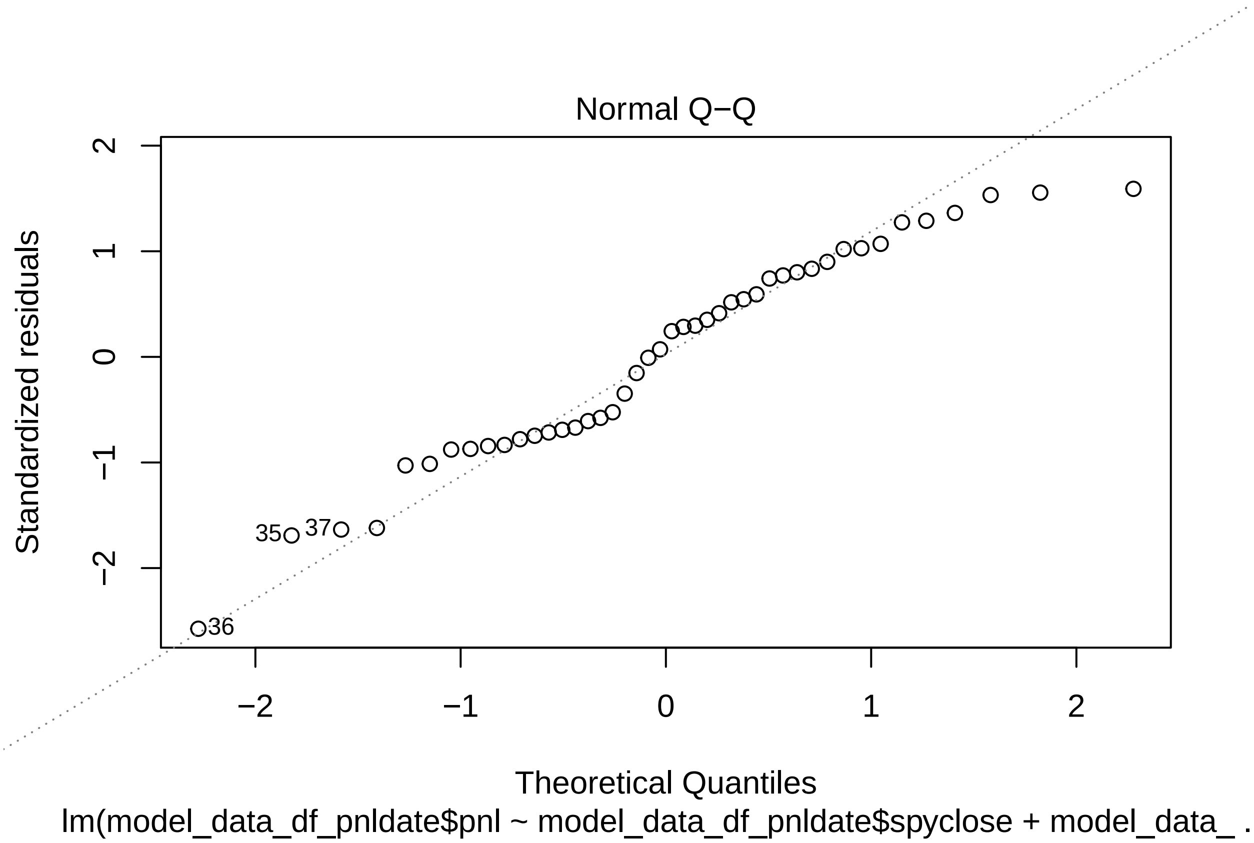
```
## [1] 0.0467
```

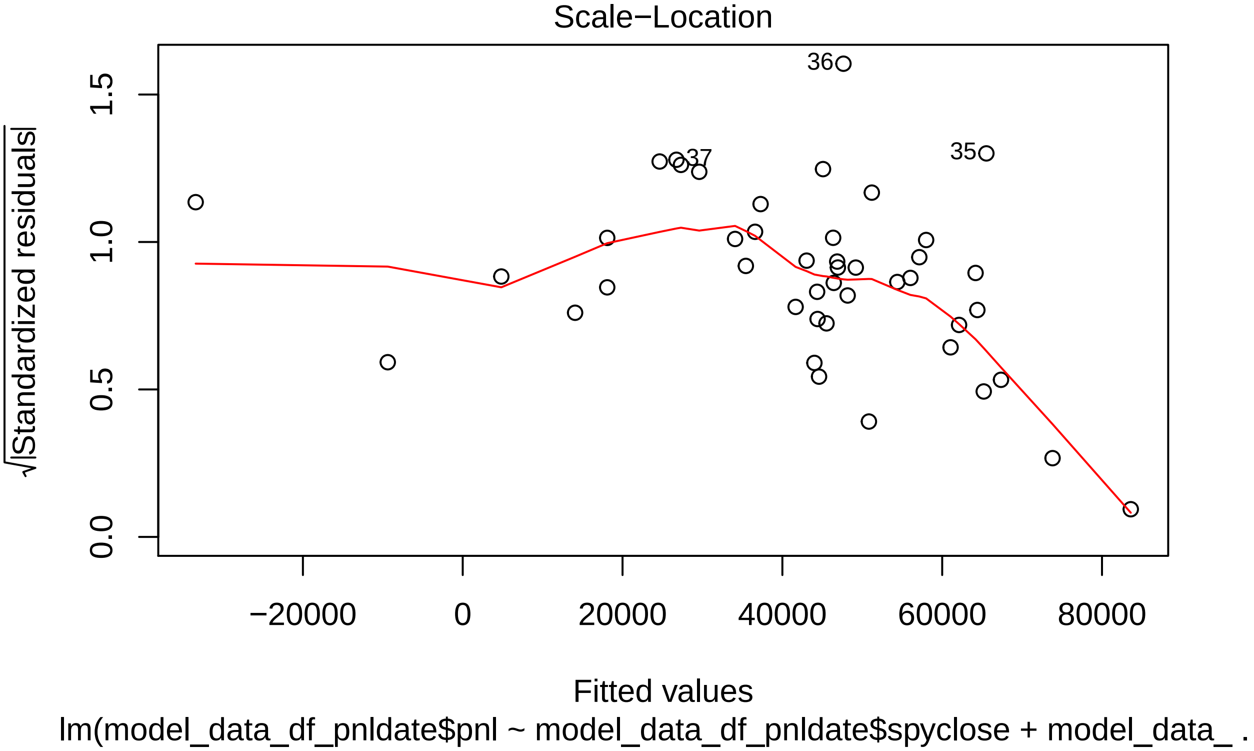
```
# run a correlation analysis between pnl and vxx+spy
cor_model3 <- cor(model_data_df_pnldate$pnl, (model_data_df_pnldate$vxx_close+
                                                model_data_df_pnldate$spyclose),
                  method = "pearson")

# compute confidence interval of slope of both spy and vxx predictors.
conf_interval <- confint(linear_model,conf.level = 0.95)
conf_interval
```

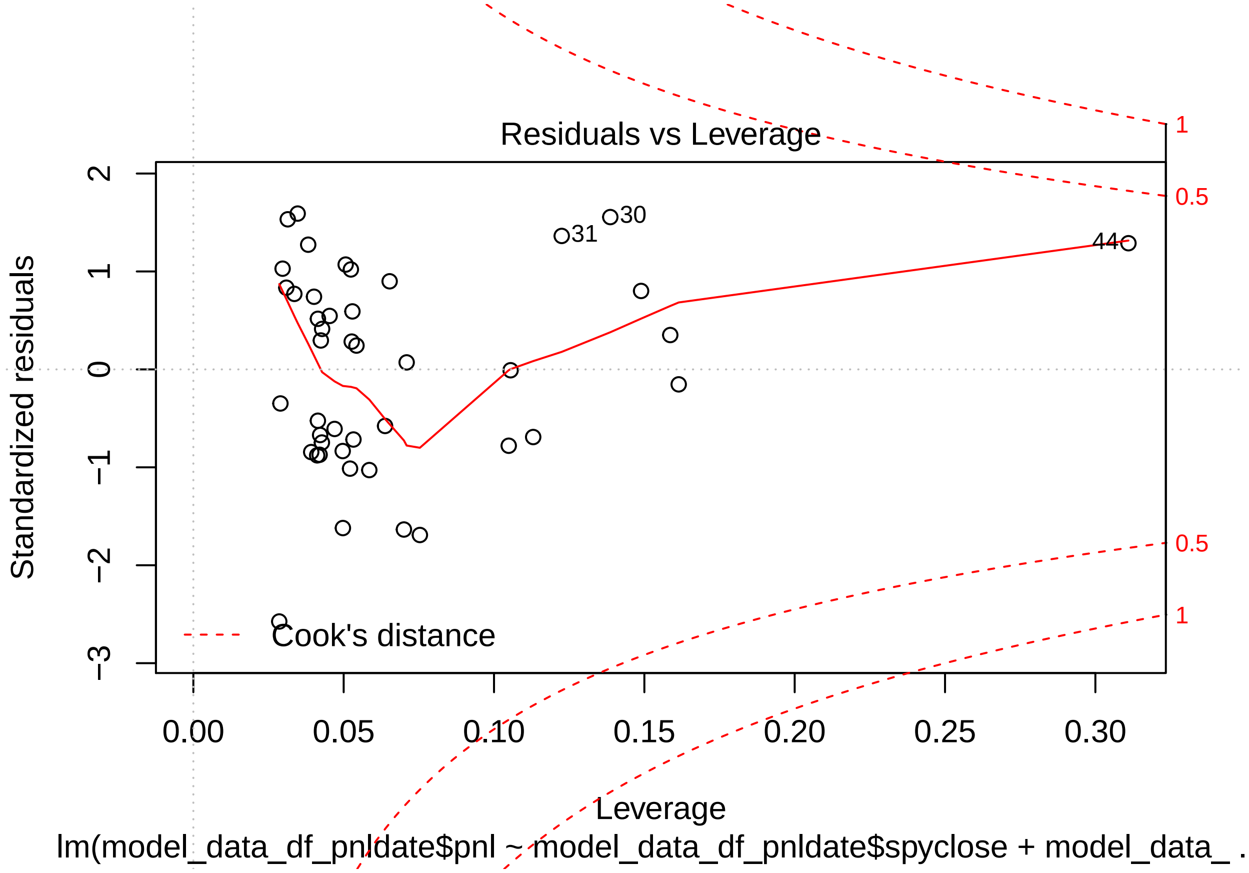
```
##                                    2.5 %  97.5 %
## (Intercept)                        1851456 5919428
## model_data_df_pnldate$spyclose    -25663   -7961
## model_data_df_pnldate$vxx_close   -27365   -7498
```

```
plot(linear_model)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
35
37
36
−2
−1
0
1
2
Theoretical Quantiles
lm(model_data_df_pnldate$pnl ~ model_data_df_pnldate$spyclose + model_data_ .


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
36
37
35
−20000
0
20000
40000
60000
80000
Fitted values
lm(model_data_df_pnldate$pnl ~ model_data_df_pnldate$spyclose + model_data_ .



INTERPRETATION

The data used in the analysis spans 44 trading sessions and 916 unique securities across 11 different models that make up one master portfolio. The data contain 75,048 observations of 22 variables. For this analysis the data was pruned to 75,048 observations of 7 variables.

PnL for the most part is evenly distributed across symbols; however, some symbols are traded more often than others. CMA in particular was an interesting case because it was amongst the most frequently traded symbols as well as the 4th worst loser of 916 symbols traded.

Based on the summaries of the pnl distributions by symbol and by day it seems that major wins and losses are concentrated by symbol and/or by day. This points to the conclusion that large losses and gains are concentrated in a handful of symbols and a handful of days.

In fact the sum of the top 5 winners is just +$83,738.91 while the sum of the top 5 losers is -$129,431.80.

The top 5 winning days yields a profit of just +$79,070.12 while the sum of the top 5 losing days yields a loss of -$148,560.50.

Since it is clear that the performance of the portfolio is highly dependent on just a few symbols and days its important to then investigate how the performance of the portfolio was impacted by market behavior.

In fact as is clear above the top 5 worst losing days were between October 15th and October 21st. Over that period of time the stock market benchmark index SPY dropped sharply and volatility of the market benchmarked by the VXX spiked sharply.

K-Means Clustering Results

The portfolio pnl indeed does cluster with the SPY and VXX indices from 10.15.2014 to 10.20.2014 which is an interesting pattern.

The next question that arises is does sharp increases in volatility of the market cause the portfolio pnl to move with market returns?

If this is the case, could a hedge be created to mitigate the market risk (beta) to protect against sharpe spikes in market volatility? How can the volatility and correlation to the market of individual symbols be measured in order to create such a hedge? What kind of lags are involved?

Multi Linear Regression and Correlation Results

We regress the SPY and VXX against the portfolio pnl to determine what portion of the portfolio pnl could be explained by SPY and VXX. In this case the portfolio pnl is the dependent variable while the SPY and VXX are independent variables.

From the multi linear regression above we see an R-squared of 0.2657 which means that 26.57% of the variation in the portfolio's pnl can be explained by the spy close and the vxx close prices.

The p-value tests the null hypothesis that the coefficients of the model with the predictors in place are 0 (have no effect). A p-value < .05 suggests that we can reject the null hypothesis and in this case since the p-value = .001779 we can conclude that the predictors - spy price and vxx price - are meaningful because changes in both indices affect how the portfolio's pnl changes. Individually each predictor affects the portfolio pnl and there is no need to discard either of the predictors.

Correlation Results

The portfolio pnl is negatively correlated to the SPY at -.2019.

The portfolio pnl is not very correlated to the vxx which is showing a correlation of .0467.

However, the portfolio pnl has a decent negative correlation to the SPY+VXX as a unit of -.5126.

Confidence Intervals:

At a 95% confidence interval the true slope for the SPY predictor is between -\$25,663.33 and -\$7,960.96.

At a 95% confidence true slope for the VXX predictor is -\$27,365.06 and -\$7,497.98.

CONCLUSIONS

My hypothesis was that the portfolio's performance was impacted by the overall market and volatility during its latest drawdown in pnl. The findings support my conclusion and show market returns and volatility account for a signficant portion of the portfolio pnl behavior and that the portfolio pnl is negatively correlated to the market and market volatility.

Further, it can be gleaned that performance is concentrated across a handful of symbols and days despite having traded across 916 symbols. On the days where market volatility is extreme the portfolio pnl clusters with market returns and volatility.

As such this study was successful and merits further investigation into the construction of a beta hedge that would seek to quantify on a symbol specific basis the market risk component. Once the market risk component can be measured on a symbol specific basis and aggregated across the portfolio a hedging mechanism could created to separate the alpha and beta components of the portfolio. If the future work proves successful then we would be left with a positive portfolio return that is not susceptiple to wild market moves and extensive risk concentration. In other words, a successful beta hedge would yield a cumulative portfolio pnl curve that would not have drawn down in mid-october.