Sprint 3

Straddle screening tool

Bassam Rizk - YU id 303525

Presentation here: https://1drv.ms/p/s!AhOBDDtkkBk4hr0x8oMUUt3AcSFulg?e=e95MHk

Table of Contents

Introduction

Straddle is an advanced options trading strategy where the trader expects high volatility levels on the stock, as a result of that a trader would buy an equivalent count of ATM – At-The-Money call and put options at the same strike price within the same expiry bucket.

The key challenge in selecting a good straddle opportunity would be predicting with a certain level of certainty which stocks will exhibit abnormal volatility in the foreseeable future.

The purpose of this project is to predict the volatility on S&P 100 stocks.

Other challenges that are beyond the scope of this paper but usually faced with straddle are Beta (correlation of the stock with the market) and Greeks (a bunch relationship metrics between the option price and time, stock price...)

Key terminologies:

> Stock: (also known as "shares" or "equity") is a type of security that signifies proportionate ownership in the issuing corporation. This entitles the stockholder to that proportion of the corporation's assets and earnings.
>
> Options: Are financial instruments that are derivatives that are based on the value of underlying securities such as stocks. An options contract offers the buyer the opportunity to buy or sell—depending on the type of contract they hold—the underlying asset
>
> Call Options: allow the holder to buy the asset at a stated price within a specific timeframe.
>
> Put Options: allow the holder to sell the asset at a stated price within a specific timeframe.

Volatility: volatility (symbol σ) is the degree of variation of a trading price series over time as measured by the standard deviation of logarithmic returns. ... Implied volatility looks forward in time, being derived from the market price of a market-traded derivative (in particular, an option)

High: Today's high is the highest price at which a stock traded during the course of the trading day. Today's high is typically higher than the closing or opening price.

Low: the trading day's intraday low price.

Open: The opening price is the price at which a security first trades upon the opening of an exchange on a trading day

Close: The closing price is the price of the final trade before the close of the trading session.

S&P 100 index: The S&P 100 Index is a stock market index of United States stocks maintained by Standard & Poor's. Index options on the S&P 100 are traded with the ticker symbol "OEX". Because of the popularity of these options, investors often refer to the index by its ticker symbol.

Symbol: A ticker symbol or stock symbol is an abbreviation used to uniquely identify publicly traded shares of a particular stock on a particular stock market. A stock symbol may consist of letters, numbers or a combination of both.

Dividends (stock event): A dividend is the distribution of reward from a portion of the company's earnings and is paid to a class of its shareholders. ... Dividends can be issued as cash payments, as shares of stock, or other property, though cash dividends are the most common.

Literature reviews

- Hedging volatility risk
    - M Brenner, EY Ou, JE Zhang - Journal of Banking & Finance, 2006 - Elsevier
    - Volatility risk plays an important role in the
    - management of portfolios of derivative assets as well as portfolios of basic assets. This risk is currently managed by volatility “swaps” or futures. However, this risk could be managed more efficiently using options on volatility that were proposed in the past but were never introduced mainly due to the lack of a cost efficient tradable underlying asset.
- The objective of this paper is to introduce a new volatility instrument, an option on a straddle, which can be used to hedge volatility risk. The design and valuation of such an instrument are the basic ingredients of a successful financial product. In order to value these options, we combine the approaches of compound options and stochastic volatility. Our numerical results show that the straddle option is a powerful instrument to hedge volatility risk. An additional benefit of such an innovation is that it will provide a direct estimate of the market price for volatility risk.
- Empirical properties of straddle returns

- F Goltz, WN Lai - The Journal of Derivatives, 2009 - jod.iijournals.com
- An at-the-money (ATM) straddle, ie, going long an ATM call and an ATM put with the same maturity, is generally thought of as a volatility trade. It is essentially delta-neutral, but a large
price move in either direction or an increase in implied volatility will produce a profit. A delta-neutral straddle position also has zero beta, so under the CAPMit should earn the riskless rate. Research has shown, however, that straddles with stock index options tend to lose money, which may be attributed to a volatility risk premium: it is the cost of hedging against a rise in volatility. If buying straddles produces losses, writing straddles should yield excess profits. An important aspect of the trade is that the delta (and beta) of the position change when the underlying index moves away from its initial level, and rebalancing is necessary if one wishes to maintain neutrality.
- In this article, Goltz and Lai examine the performance of buying and holding one-month straddles on the DAX index, with and without rebalancing, and find negative returns on average. If investors are entering the trade as a volatility hedge, one might expect the return to vary with other measures on volatility risk and potential hedging demand. They find that a widening credit spread on corporate bonds relative to government bonds, greater stock market turnover, and higher actual volatility all are related to straddle returns. But in considering what position an investor with constant relative risk aversion would take in straddles as part of an optimal portfolio including the underlying stock index and the riskless asset, they show that for risk aversion over a broad range, the optimal position would be to short straddles. That is, the "risk premium" in the market is too big to be consistent with utility maximization by investors with a reasonable level of risk aversion. The effect is most important for daily rebalancing, but that requires bearing heavy transaction costs, to the point that the potential improvement in utility would be largely wiped out in trying to capture it in the market.

Research Question

What is the highest reachable consistency in screening stocks that will exhibit abnormal price volatility at a foreseeable market or stock event (dividends announcement…)?

Methodology

- As part of data acquisition & exploration; consolidate, transform & clean key variables of S&P100 stocks:
    - Download individual stock information (high, low, open & close…)
    - Capture dividend events (amounts & date) and merge it with data set
    - Measure overnight and high/low stock price volatility during the last 12 years
    - Perform rounds of cleaning of NAs and transform all values into numeric.

    - Merge all 100 individual stock information into 1 data set
- Divide this data into training (70%), validation (15%) and training (15%) sets
- Using training data sets:
    - Targeting prediction of HLPPC – High Low Percentage Price Change
        - Run a principal component analysis to identify the m
        - Clustering for optimal straddle opportunities using multi variables
        - 3 different models
            - Decision Trees
            - Neural Network
            - Linear regression
- Using testing data set: Test all 3 methods to identify the best selection method.

Data Acquisition & Cleaning

Downloading Data

#There weren't any API to filter S&P100 stocks of all listed stocks. So, I got a list of S&P 100 from Wikipedia

#Converted that list from table to csv using Excel / notepad (I tried doing that using R - it was too complicated...)

#Copy/ pasted that list into a getSymbols function from Quantmod.

getSymbols(c('AAPL', 'ABBV', 'ABT', 'ACN', 'ADBE', 'AGN', 'AIG', 'ALL', 'AMGN', 'AMZN', 'AXP', 'BA', 'BAC', 'BIIB', 'BK', 'BKNG', 'BLK', 'BMY', 'C', 'CAT', 'CELG', 'CHTR', 'CL', 'CMCSA', 'COF', 'COP', 'COST', 'CSCO', 'CVS', 'CVX', 'DD', 'DHR', 'DIS', 'DOW', 'DUK', 'EMR', 'EXC', 'F', 'FB', 'FDX', 'GD', 'GE', 'GILD', 'GM', 'GOOG', 'GOOGL', 'GS', 'HD', 'HON', 'IBM', 'INTC', 'JNJ', 'JPM', 'KHC', 'KMI', 'KO', 'LLY', 'LMT', 'LOW', 'MA', 'MCD', 'MDLZ', 'MDT', 'MET', 'MMM', 'MO', 'MRK', 'MS', 'MSFT', 'NEE', 'NFLX', 'NKE', 'NVDA', 'ORCL', 'OXY', 'PEP', 'PFE', 'PG', 'PM', 'PYPL', 'QCOM', 'RTN', 'SBUX', 'SLB', 'SO', 'SPG', 'T', 'TGT', 'TXN', 'UNH', 'UNP', 'UPS', 'USB', 'UTX', 'V', 'VZ', 'WBA', 'WFC', 'WMT', 'XOM'))

#This worked!!!!!! It took me couple of days to figure out how to call this to multiple symbols… plus it

I received in my environment 100 xts objects with history from 2007-01-03 to 2019-08-02

	NFLX.Open	NFLX.High	NFLX.Low	NFLX.Close	NFLX.Volume	NFLX.Adjusted
2007-01-03	3.714286	3.824286	3.677143	3.801429	16440900	3.801429
2007-01-04	3.772857	3.828571	3.585714	3.621428	15959300	3.621428
2007-01-05	3.620000	3.620000	3.492857	3.544286	15190700	3.544286
2007-01-08	3.545714	3.555714	3.367143	3.404286	18344900	3.404286
2007-01-09	3.427143	3.440000	3.360000	3.427143	10611300	3.427143

Measure Volatility

I needed to add a trailing volatility measure of each stock for each time point.

#I added a column to measure volatility of each stock - I tried doing that in one function, it got too complicated.

AAPL$VOLA<- volatility(AAPL)

ABBV$VOLA<- volatility(ABBV)

… replicate to all 100 stocks

Clean up: #remove NAs FROM FIRST ROWS IN THE VOLATILITY COLUMN

AAPL[is.na(AAPL)] <-0

ABBV[is.na(ABBV)] <-0

… replicate to all 100 stocks

ADD SYMBOL

Add a column for ticker symbol - might be useful as we agreggate information

AAPL$Ticker <- NA

ABBV$Ticker <- NA

… replicate to all 100 stocks

Clean up Replace "NA"s with ticker symbol of each stock so I could aggregate the data at the end and have the stock symbol as an identifier of each row…

AAPL$Ticker[is.na(AAPL$Ticker)] <- "AAPL"

ABBV$Ticker[is.na(ABBV$Ticker)] <- "ABBV"

… replicate to all 100 stocks

Add Stock Events

 # Track historical dividends & stock splits events for each stock – this would serve as a key trigger to identify correlation with price volatility… had to spend hours in testing multiple sources (Finam…) most were very unreliable (by either lack of correctness per each dividends or coverage per stock)… Yahoo seemed the most reliable.

SiDiAAPL<- get_yahoo_splits_and_dividends('AAPL','2007-01-03', '2019-08-02')

SiDiABBV<- get_yahoo_splits_and_dividends('ABBV','2007-01-03', '2019-08-02')

SiDiABT<- get_yahoo_splits_and_dividends('ABT','2007-01-03', '2019-08-02')

… replicate to all 100 stocks

USING DATES as a row label MERGE dividends & symbol xts file (aka stock historical data)– this was a tricky one – having dividends API from a different library wasn't compatible with the work I had…

AAPL<- merge(AAPL, SiDiAAPL)

ABBV<- merge(ABBV, SiDiAAPL)

ABT<- merge(ABT, SiDiAAPL)

… replicate to all 100 stocks

Clean-up after dividends & ticker addition

#REMOVE NAs FROM TICKER SYMBOLS

(Ticker symbol seems to be lost while doing the merge as this was an external field that I manually added and the merge worked because its is being sourced from the same API… I had to add it back.

AAPL$Ticker[is.na(AAPL$Ticker)] <- "AAPL"

ABBV$Ticker[is.na(ABBV$Ticker)] <- "ABBV"

… replicate to all 100 stocks

#REMOVE NAs FROM DIVIDEND VALUE

AAPL$value[is.na(AAPL$value)] <- 0

ABBV$value[is.na(ABBV$value)] <- 0

… replicate to all 100 stocks

#Summary view of all stocks

summary (AAPL)

summary (ABBV)

… replicate to all 100 stocks

Key findings

####xts is being classified as factor – so I will transform all the symbols xts into data frames

than I will transform all columns (except date & ticker) into numeric

Clean-up: Transform Symbols into data frames as preparation to consolidate

```
AAPLfull<- data.frame(AAPL)

ABBVfull<- data.frame(ABBV)
```

… replicate to all 100 stocks

Cleanup: transform **Volatility** classification from factor into numeric

```
AAPLfull$VOLA<- as.numeric(as.character(AAPLfull$VOLA))

ABBVfull$VOLA<- as.numeric(as.character(ABBVfull$VOLA))
```

… replicate to all 100 stocks

Clean up: Transform **Open** column from factor to numeric

```
AAPLfull$AAPL.Open<- as.numeric(as.character(AAPLfull$AAPL.Open))

ABBVfull$ABBV.Open<- as.numeric(as.character(ABBVfull$ABBV.Open))
```

… replicate to all 100 stocks

Clean up: transform **High** column from factor to numeric

```
AAPLfull$AAPL.High<- as.numeric(as.character(AAPLfull$AAPL.High))

ABBVfull$ABBV.High<- as.numeric(as.character(ABBVfull$ABBV.High))
```

… replicate to all 100 stocks

Clean up: Transform **Low** Column classification from factor into numeric

```
AAPLfull$AAPL.Low<- as.numeric(as.character(AAPLfull$AAPL.Low))

ABBVfull$ABBV.Low<- as.numeric(as.character(ABBVfull$ABBV.Low))
```

… replicate to all 100 stocks

Clean up: Transform **Close** column classification from factor into numeric

```
AAPLfull$AAPL.Close<- as.numeric(as.character(AAPLfull$AAPL.Close))

ABBVfull$ABBV.Close<- as.numeric(as.character(ABBVfull$ABBV.Close))
```

… replicate to all 100 stocks

Clean up: Transform **Volume** column classification from factor into numeric

```
AAPLfull$AAPL.Volume<- as.numeric(as.character(AAPLfull$AAPL.Volume))

ABBVfull$ABBV.Volume<- as.numeric(as.character(ABBVfull$ABBV.Volume))

ABTfull$ABT.Volume<- as.numeric(as.character(ABTfull$ABT.Volume))
```

… replicate to all 100 stocks

Clean up: Transform **Adjusted** column classification from factor into numeric

AAPLfull$AAPL.Adjusted<- as.numeric(as.character(AAPLfull$AAPL.Adjusted))

ABBVfull$ABBV.Adjusted<- as.numeric(as.character(ABBVfull$ABBV.Adjusted))

… replicate to all 100 stocks

Clean up: Transform dividend **value** column's classification from factor to numeric

AAPLfull$value<- as.numeric(as.character(AAPLfull$value))

ABBVfull$value<- as.numeric(as.character(ABBVfull$value))

… replicate to all 100 stocks

Standardize nomenclature of header

This is done so we could merge the all data frames into 1 large one

setnames(AAPLfull, old=c("AAPL.Open", "AAPL.High", "AAPL.Low", "AAPL.Close", "AAPL.Volume", "AAPL.Adjusted", "VOLA", "Ticker", "value"), new=c("open", "High", "Low", "Close", "Volume", "Adjusted", "Volatility", "Symbol", "Dividends"))

setnames(ABBVfull, old=c("ABBV.Open", "ABBV.High", "ABBV.Low", "ABBV.Close", "ABBV.Volume", "ABBV.Adjusted", "VOLA", "Ticker", "value"), new=c("open", "High", "Low", "Close", "Volume", "Adjusted", "Volatility", "Symbol", "Dividends"))

… replicate to all 100 stocks

Below is sample stock data frame with no NAs, standardized headers and numeric value ready to be merged and analyzed

	open	High	Low	Close	Volume	Adjusted	Volatility	Symbol	Dividends
2007-01-03	97.18	98.40	96.26	97.27	9196800	69.29244	0.00000000	IBM	0
2007-01-04	97.25	98.79	96.88	98.31	10524500	70.03336	0.00000000	IBM	0
2007-01-05	97.60	97.95	96.91	97.42	7221300	69.39934	0.00000000	IBM	0
2007-01-08	98.50	99.50	98.35	98.90	10340000	70.45365	0.00000000	IBM	0
2007-01-09	99.08	100.33	99.07	100.07	11108200	71.28710	0.00000000	IBM	0
2007-01-10	98.50	99.05	97.93	98.89	8744800	70.44652	0.00000000	IBM	0
2007-01-11	99.00	99.90	98.50	98.65	8000700	70.27556	0.00000000	IBM	0
2007-01-12	98.99	99.69	98.50	99.34	6636500	70.76708	0.00000000	IBM	0
2007-01-16	99.40	100.84	99.30	100.82	9602200	71.82140	0.00000000	IBM	0
2007-01-17	100.69	100.90	99.90	100.02	8200700	71.25151	0.17593969	IBM	0
2007-01-18	99.80	99.95	98.91	99.45	14636100	70.84544	0.17512045	IBM	0
2007-01-19	95.00	96.85	94.55	96.17	26035800	68.50887	0.25379507	IBM	0

Merge all 100 data-frames into one data frame

SnP100full<- rbind(AAPLfull, ABBVfull, ABTfull, ACNfull, ADBEfull, AGNfull, AIGfull, ALLfull, AMGNfull, AMZNfull, AXPfull, BAfull, BACfull, BIIBfull, BKfull, BKNGfull, BLKfull, BMYfull, Cfull, CATfull, CELGfull, CHTRfull, CLfull, CMCSAfull, COFfull, COPfull, COSTfull, CSCOfull, CVSfull, CVXfull, DDfull, DHRfull, DISfull, DOWfull, DUKfull, EMRfull, EXCfull, Ffull, FBfull, FDXfull, GDfull, GEfull, GILDfull, GMfull, GOOGfull, GOOGLfull, GSfull, HDfull, HONfull, IBMfull, INTCfull, JNJfull, JPMfull, KHCfull, KMIfull, KOfull, LLYfull, LMTfull, LOWfull, MAfull, MCDfull, MDLZfull, MDTfull, METfull, MMMfull, MOfull, MRKfull, MSfull, MSFTfull, NEEfull, NFLXfull, NKEfull, NVDAfull, ORCLfull, OXYfull, PEPfull, PFEfull, PGfull, PMfull, PYPLfull, QCOMfull, RTNfull, SBUXfull, SLBfull, SOfull, SPGfull, Tfull, TGTfull, TXNfull, UNHfull, UNPfull, UPSfull, USBfull, UTXfull, Vfull, VZfull, WBAfull, WFCfull, WMTfull, XOMfull)

#successfully merged all data sets and added ticker – I've been struggling in trying to call this function since 2 weeks now – it took rounds of clean-up and data transformation to capture that…

```
Data
SnP100full              303361 obs. of 9 variables
   open : num 12.3 12 12.3 12.3 12.3 ...
   High : num 12.4 12.3 12.3 12.4 13.3 ...
   Low : num 11.7 12 12.1 12.2 12.2 ...
   Close : num 12 12.2 12.2 12.2 13.2 ...
   Volume : num 3.10e+08 2.12e+08 2.09e+08 1.99e+08 8.37e+08 ...
   Adjusted : num 10.5 10.7 10.6 10.7 11.6 ...
   Volatility: num 0 0 0 0 0 ...
   Symbol : Factor w/ 100 levels "AAPL","ABBV",..: 1 1 1 1 1 1 1 1 1 1 ...
   Dividends : num 0 0 0 0 0 0 0 0 0 0 ...
```

summary(SnP100full)

```
     open              High               Low              Close        
 Min.   :   1.31   Min.   :   1.55   Min.   :   1.01   Min.   :   1.26  
 1st Qu.:  35.49   1st Qu.:  35.87   1st Qu.:  35.11   1st Qu.:  35.49  
 Median :  57.93   Median :  58.48   Median :  57.36   Median :  57.93  
 Mean   :  96.66   Mean   :  97.59   Mean   :  95.67   Mean   :  96.65  
 3rd Qu.:  93.22   3rd Qu.:  94.00   3rd Qu.:  92.37   3rd Qu.:  93.23  
 Max.   :2210.93   Max.   :2228.99   Max.   :2174.07   Max.   :2206.09  
 NA's   :56        NA's   :56        NA's   :56        NA's   :56        
     Volume              Adjusted           Volatility          Symbol      
 Min.   :0.000e+00   Min.   :   0.8996   Min.   :0.0000   AAPL   :  3169  
 1st Qu.:3.533e+06   1st Qu.:  29.1209   1st Qu.:0.1409   ABT    :  3169  
 Median :6.668e+06   Median :  49.1364   Median :0.2010   ACN    :  3169  
 Mean   :1.398e+07   Mean   :  88.5902   Mean   :0.2511   ADBE   :  3169  
 3rd Qu.:1.379e+07   3rd Qu.:  83.5500   3rd Qu.:0.2939   AGN    :  3169  
 Max.   :1.227e+09   Max.   :2206.0901   Max.   :7.0796   AIG    :  3169  
 NA's   :56          NA's   :56          NA's   :56       (Other):284347  
   Dividends     
 Min.   :0.0000  
 1st Qu.:0.0000  
 Median :0.0000  
 Mean   :0.0051  
 3rd Qu.:0.0000  
 Max.   :0.7700  
```

CLEAN-UP work-space Environment

At this point my R-studio and laptop became too slow because of the 100s of data frames

In addition, I wasn't able to leverage some of the specialized charts my libraries had because I had changed the xts files they had initially produced.

I deleted all initial symbols and call upon them again - that will enable me to call on the generic stock charts from fnancial R libraries

AAPL<- NULL

ABBV<- NULL

… replicate to all 100 stocks

call the symbols back in their original format

getSymbols(c('AAPL', 'ABBV', 'ABT', 'ACN', 'ADBE', 'AGN', 'AIG', 'ALL', 'AMGN', 'AMZN', 'AXP', 'BA', 'BAC', 'BIIB', 'BK', 'BKNG', 'BLK', 'BMY', 'BRK.B', 'C', 'CAT', 'CELG', 'CHTR', 'CL', 'CMCSA', 'COF', 'COP', 'COST', 'CSCO', 'CVS', 'CVX', 'DD', 'DHR', 'DIS', 'DOW', 'DUK', 'EMR', 'EXC', 'F', 'FB', 'FDX', 'GD', 'GE', 'GILD', 'GM', 'GOOG', 'GOOGL', 'GS', 'HD', 'HON', 'IBM', 'INTC', 'JNJ', 'JPM', 'KHC', 'KMI', 'KO', 'LLY', 'LMT', 'LOW', 'MA', 'MCD', 'MDLZ', 'MDT', 'MET', 'MMM', 'MO', 'MRK', 'MS', 'MSFT', 'NEE', 'NFLX', 'NKE', 'NVDA', 'ORCL', 'OXY', 'PEP', 'PFE', 'PG', 'PM', 'PYPL', 'QCOM', 'RTN', 'SBUX', 'SLB', 'SO', 'SPG', 'T', 'TGT', 'TXN', 'UNH', 'UNP', 'UPS', 'USB', 'UTX', 'V', 'VZ', 'WBA', 'WFC', 'WMT', 'XOM'))

Challenges

Did a dry run on a test sample and have 2 challenges in running Clustering functions:

1. there are still NAs in relatively newer stocks (e.g. Netflix...) that were not existent throughout the 2007-2019 sample period.
2. Preset library functions for measuring percentage price change (Overnight & intra-day) do not seem to be compatible with various clustering functions.

Solutions

Go back to the individual stock data frames and do couple of clean-ups:

1. Transform all remaining NAs in individual data sets into nill value
2. add overnight and same day percentage price change
3. remerge the individual 100 stock data frames into a new large data frame

Transform all remaining NAs in individual data sets into nil value

AAPLfull[is.na(AAPLfull)] <-0

ABBVfull[is.na(ABBVfull)] <-0

… replicate for all stocks

Add overnight and same day percentage price change

 # start with overnight (difference between close of the day and the previous day)

AAPLfull$ON_PPC<- c(-diff(AAPLfull$Close)/AAPLfull$Close[-1]*100,0)

ABBVfull$ON_PPC<- c(-diff(ABBVfull$Close)/ABBVfull$Close[-1]*100,0)

… replicate for all 100 stocks

Measure percent change between a day's high & low

AAPLfull$HLppc<- c((AAPLfull$High - AAPLfull$Low)/AAPLfull$Low*100)

ABBVfull$HLppc<- c((ABBVfull$High - ABBVfull$Low)/ABBVfull$Low*100)

… replicate for all 100 stocks

Removing NAs

There seems to be few NAs in few stocks - this applies to period where a stock was not yet listed

Clean-up - another round of cleaning NAs in the percentage change fields

AAPLfull[is.na(AAPLfull)] <-0

ABBVfull[is.na(ABBVfull)] <-0

… replicate for all 100 stocks

Re-merging all 100 files after adding overnight and high/close percentage change

SnP100full<- rbind(AAPLfull, ABBVfull, ABTfull, ACNfull, ADBEfull, AGNfull, AIGfull, ALLfull, AMGNfull, AMZNfull, AXPfull, BAfull, BACfull, BIIBfull, BKfull, BKNGfull, BLKfull, BMYfull, Cfull, CATfull, CELGfull, CHTRfull, CLfull, CMCSAfull, COFfull, COPfull, COSTfull, CSCOfull, CVSfull, CVXfull, DDfull, DHRfull, DISfull, DOWfull, DUKfull, EMRfull, EXCfull, Ffull, FBfull, FDXfull, GDfull, GEfull, GILDfull, GMfull, GOOGfull, GOOGLfull, GSfull, HDfull, HONfull, IBMfull, INTCfull, JNJfull, JPMfull, KHCfull, KMIfull, KOfull, LLYfull, LMTfull, LOWfull, MAfull, MCDfull, MDLZfull, MDTfull, METfull, MMMfull, MOfull, MRKfull, MSfull, MSFTfull, NEEfull, NFLXfull, NKEfull, NVDAfull, ORCLfull, OXYfull, PEPfull, PFEfull, PGfull, PMfull, PYPLfull, QCOMfull, RTNfull, SBUXfull, SLBfull, SOfull, SPGfull, Tfull, TGTfull, TXNfull, UNHfull, UNPfull, UPSfull, USBfull, UTXfull, Vfull, VZfull, WBAfull, WFCfull, WMTfull, XOMfull)

Data Exploration

Explore % change by stock vs stock volatility (outside Rattle)

***Comment*: Obviously new stocks like Netflix and stocks that witnessed turmoil during the 2008 global economic crisis (e.g. AIG) top the list when sorted by average daily percentage price change.**

I wanted to explore a summarized table by stock of key metrics

```
SnPSummary <- group_by(SnP100full, SnP100full$Symbol)

SnPSummary = summarise(SnPSummary,

                avg_vlty = mean(Volatility),

                min_vlty = min(Volatility),

                max_vlty = max(Volatility),

                avg_ON_ppc = mean(ON_PPC),
```

min_On_ppc = min(ON_PPC),

max_on_ppc = max(ON_PPC),

avg_HL_ppc = mean(HLppc),

min_HL_ppc = min(HLppc),

max_HL_ppc = max(HLppc))

SnPSummary

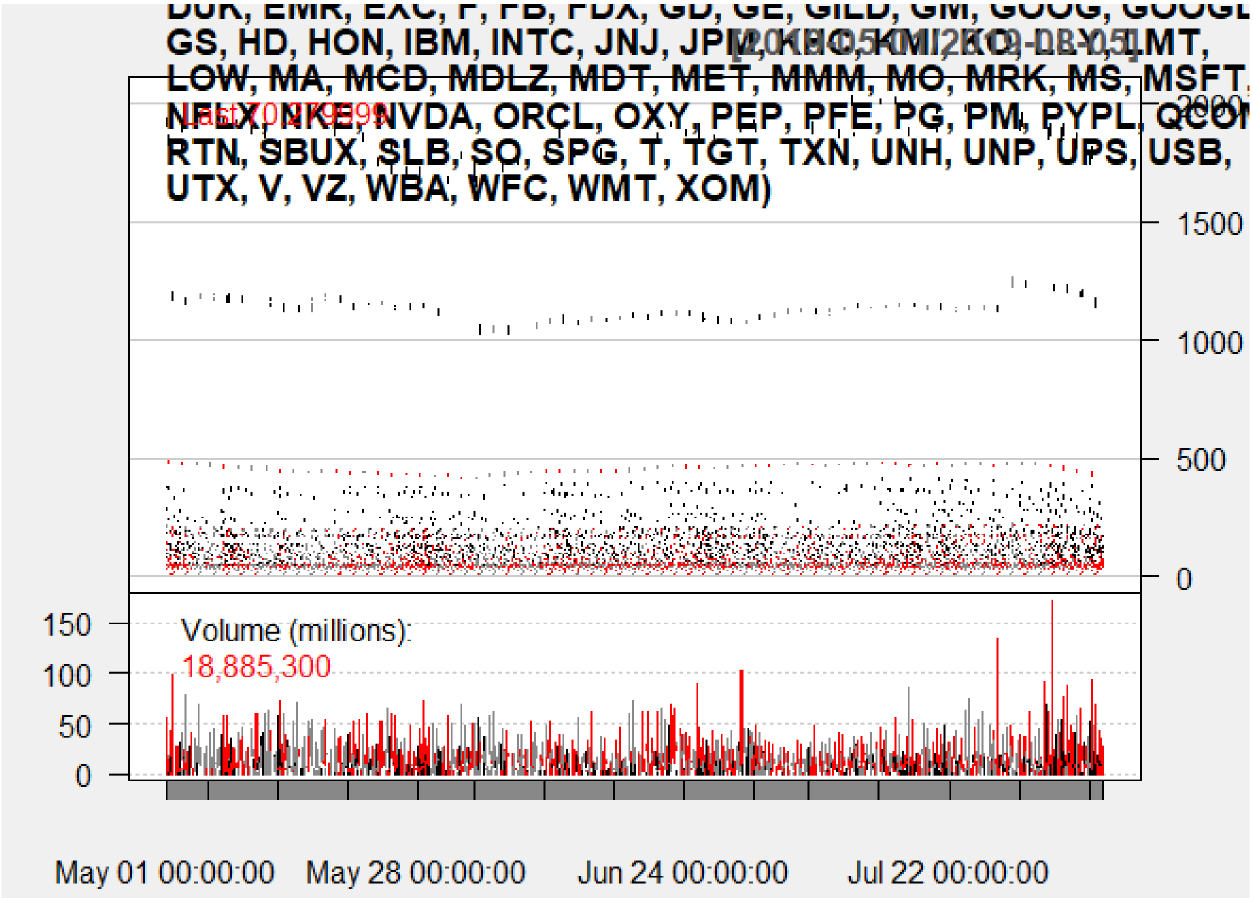
	SnP100full$Symbol	avg_vlty	min_vlty	max_vlty	avg_ON_ppc	min_On_ppc	max_on_ppc	avg_HL_ppc	min_HL_ppc	max_HL_ppc
100	NFLX	0.4582624	0	2.5101368	-0.08265853809	-29.688136	53.599581	3.808675	0.61086947	29.595746
99	AIG	0.4156295	0	7.0795822	0.21284567693	-39.759036	155.042028	3.639627	0.31728341	309.600008
98	NVDA	0.4155334	0	1.9600368	-0.01218033521	-22.962378	44.355492	3.550914	0.66979236	33.146067
97	MS	0.3868942	0	4.7694282	0.07635247392	-46.519337	34.939751	3.405129	0.48515410	111.282043
96	C	0.3832878	0	4.2090489	0.13016364372	-36.638654	64.000000	3.248808	0.42153862	81.311472
95	F	0.3324752	0	2.5995613	0.02972956565	-22.790698	33.333333	3.133792	0.50000000	160.476190
94	BAC	0.3793326	0	3.1143036	0.07719676457	-26.073298	40.784314	3.111502	0.39318480	61.660079
93	COF	0.3545884	0	2.4113365	0.04054039161	-20.904921	33.408072	3.105245	0.32330516	33.410138
92	MET	0.3391642	0	2.7983662	0.04699267873	-21.874999	36.555554	2.852607	0.43948903	31.444098

Charts

Sample 4 months view of a Google – a relatively volatile stockcandleChart(GOOG, subset= "last 4 months",multi.col=TRUE,theme="white", addMACD(fast = 12, slow = 26, signal = 9, type = "EMA"))

Spikes in volume definitely have effect on price – will test volume effect in clustering within sprint 2

Next will test a 4 months view across all 100 stocks barChart(c(AAPL, ABBV, ABT, ACN, ADBE, AGN, AIG, ALL, AMGN, AMZN, AXP, BA, BAC, BIIB, BK, BKNG, BLK, BMY, C, CAT, CELG, CHTR, CL, CMCSA, COF, COP, COST, CSCO, CVS, CVX, DD, DHR, DIS, DOW, DUK, EMR, EXC, F, FB, FDX, GD, GE, GILD, GM, GOOG, GOOGL, GS, HD, HON, IBM, INTC, JNJ, JPM, KHC, KMI, KO, LLY, LMT, LOW, MA, MCD, MDLZ, MDT, MET, MMM, MO, MRK, MS, MSFT, NEE, NFLX, NKE, NVDA, ORCL, OXY, PEP, PFE, PG, PM, PYPL, QCOM, RTN, SBUX, SLB, SO, SPG, T, TGT, TXN, UNH, UNP, UPS, USB, UTX, V, VZ, WBA, WFC, WMT, XOM), subset="last 4 months" ,multi.col=TRUE,theme="white")



This is a very messy view and couldn't conclude much in it – it's all over the place.

Modeling & Results

Rattle

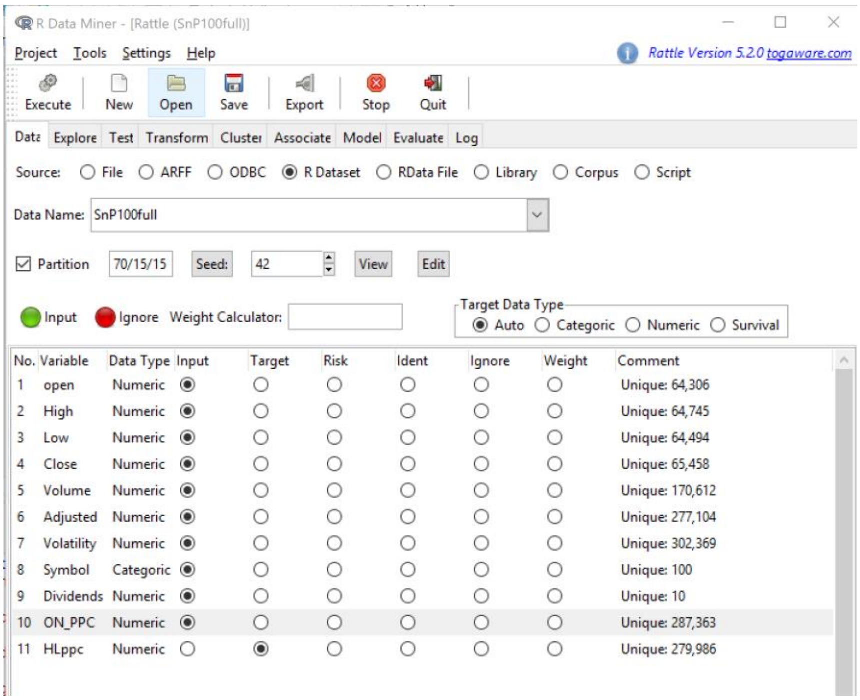
```
install.packages("rattle")

install.packages("rattle", dependencies=c("Depends", "Suggests"))

library(rattle)

rattle()
```

https://cran.r-project.org/src/contrib/Archive/RGtk2/RGtk2_2.20.35.tar.gz

Select in case of a difficulty to open .rattle file in RGtk2 – please use the above link to download the earlier 2.20.35 version of RGtk2 library – that should solve the issue

Re-Explore data within Rattle

Select file SnP100full with High Low percentage change as a target variable.



==

Below is a description of the dataset.

The data is limited to the training dataset.

crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)]

11 Variables 212352 Observations

The data was few pages and couldn't clearly move it from text to clean tables – I summarized in the table below

Rattle timestamp: 2019-08-09 09:33:06 bassa

==

Basic statistics for key numeric variable of the dataset.

metric	$Volatility	$ON_PPC	$HLppc	$Dividends
nobs	212352	212352	212352	212352
NAs	0	0	0	0
Minimum	0	-100	0	0
Maximum	7.079582	155.04203	309.6	0.77
1 Quartile	0.140928	-0.845982	1.203861	0
3 Quartile	0.29334	0.761575	2.55102	0
Mean	0.250731	**-0.012775**	**2.207995**	0.005104
Median	0.200953	-0.046004	1.712853	0
Sum	53243.1645	-2712.801	468872.2	1083.78137
SE Mean	0.00044	0.004421	0.004533	0.000116
LCL Mean	0.249868	-0.02144	2.199111	0.004876
UCL Mean	0.251593	-0.00411	2.216879	0.005332
Variance	0.041118	4.150226	4.362836	0.002871
Stdev	0.202775	2.03721	2.08874	0.053586
Skewness	5.600613	1.686744	23.2184	10.91106
Kurtosis	79.753947	287.57577	2484.471	121.978523

==

Kurtosis for each numeric variable of the dataset.

Larger values mean sharper peaks and flatter tails.

Positive values indicate an acute peak around the mean.

Negative values indicate a smaller peak around the mean.

Open	High	Low	Close	Volume	Adjusted	Volatility	ON_PPC	HLppc	Dividends
55.91057	55.84031	55.97522	55.86483	140.44138	57.52392	79.75395	287.57577	2484.47069	55.91057

==

Skewness for each numeric variable of the dataset.

Positive means the right tail is longer.

Open	High	Low	Close	Volume	Adjusted	Volatility	ON_PPC	HLppc	Dividends

6.709407	6.706561	6.712658	6.707722	8.901054	6.784499	5.600613	1.686744	23.218403	10.911060

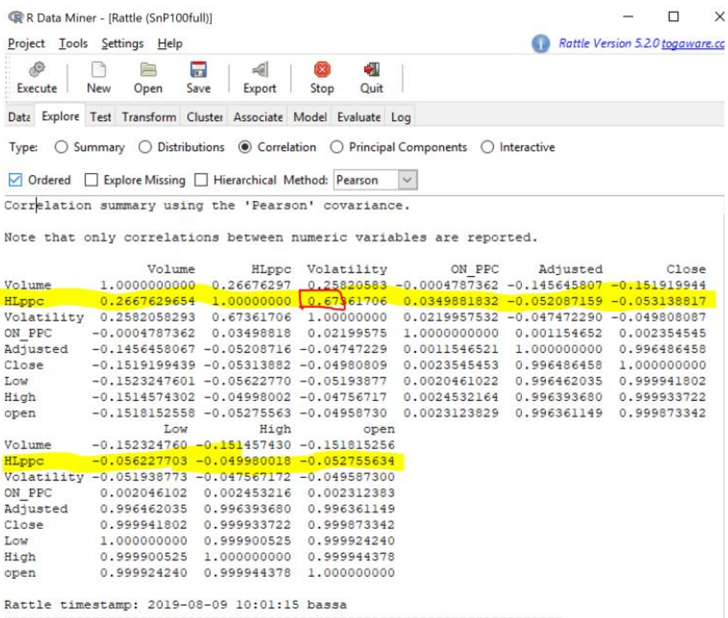
Rattle timestamp: 2019-08-09 09:33:09 bassa

==

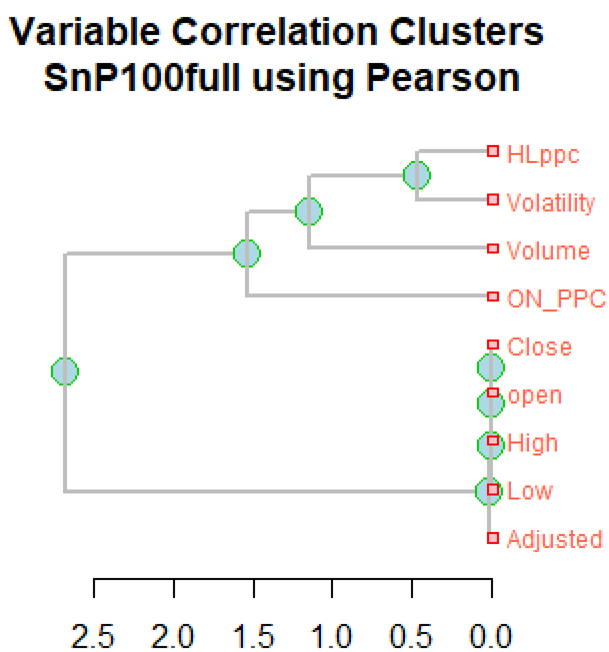
Explore Correlation Between Key Variables

<u>Take away</u>: Its very obvious that HLPPC High low percentage price change per day has a high correlation with stock volatility at 67%

Also a significant variable with high correlation to High/Low ppc is the traded volume at 26%



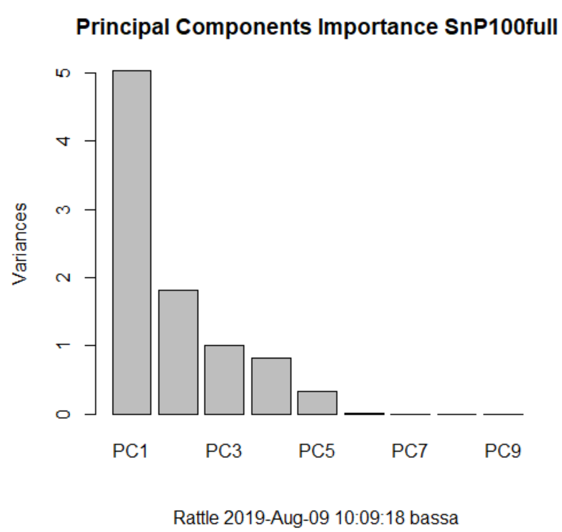

```
Correlation summary using the 'Pearson' covariance.

Note that only correlations between numeric variables are reported.

                  Volume       HLppc  Volatility        ON_PPC     Adjusted        Close
Volume      1.0000000000  0.26676297  0.25820583 -0.0004787362 -0.145645807 -0.151919944
HLppc       0.2667629654  1.00000000  0.67361706  0.0349881832 -0.052087159 -0.053138817
Volatility  0.2582058293  0.67361706  1.00000000  0.0219957532 -0.047472290 -0.049808087
ON_PPC     -0.0004787362  0.03498818  0.02199575  1.0000000000  0.001154652  0.002354545
Adjusted   -0.1456458067 -0.05208716 -0.04747229  0.0011546521  1.000000000  0.996486458
Close      -0.1519199439 -0.05313882 -0.04980809  0.0023545453  0.996486458  1.000000000
Low        -0.1523247601 -0.05622770 -0.05193877  0.0020461022  0.996462035  0.999941802
High       -0.1514574302 -0.04998002 -0.04756717  0.0024532164  0.996393680  0.999933722
open       -0.1518152558 -0.05275563 -0.04958730  0.0023123829  0.996361149  0.999873342
                    Low         High         open
Volume     -0.152324760 -0.151457430 -0.151815256
HLppc      -0.056227703 -0.049980018 -0.052755634
Volatility -0.051938773 -0.047567172 -0.049587300
ON_PPC      0.002046102  0.002453216  0.002312383
Adjusted    0.996462035  0.996393680  0.996361149
Close       0.999941802  0.999933722  0.999873342
Low         1.000000000  0.999900525  0.999924240
High        0.999900525  1.000000000  0.999944378
open        0.999924240  0.999944378  1.000000000

Rattle timestamp: 2019-08-09 10:01:15 bassa
======================================================================
```

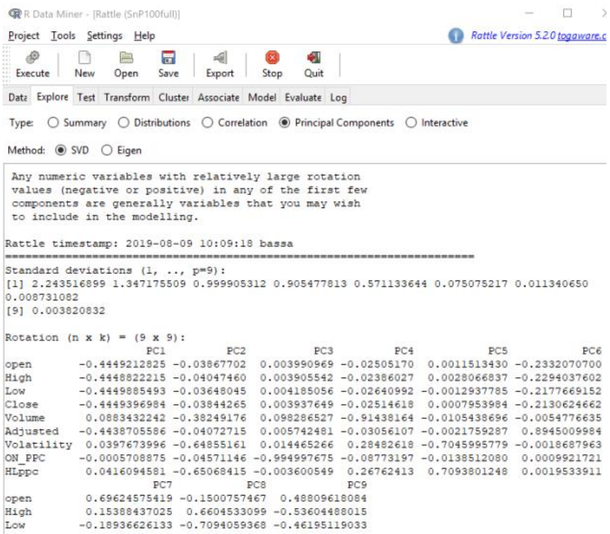


Principal component

<u>Take-away</u>: PC 1 and 2 provide the most significant results – interestingly:

- ***PC1 is driven relatively more by other variables than volatility and HLPPC.***
- ***PC2 is mostly driven by volatility and HLPPC***





Clustering

In clustering we used 2 methods Kmeans and EWKM

K means

<u>Take-away</u>: Very obvious from the data means that volatility & ON_PPC (overnight volatility) have the highest "Data means" vs our target variable HLPPC.

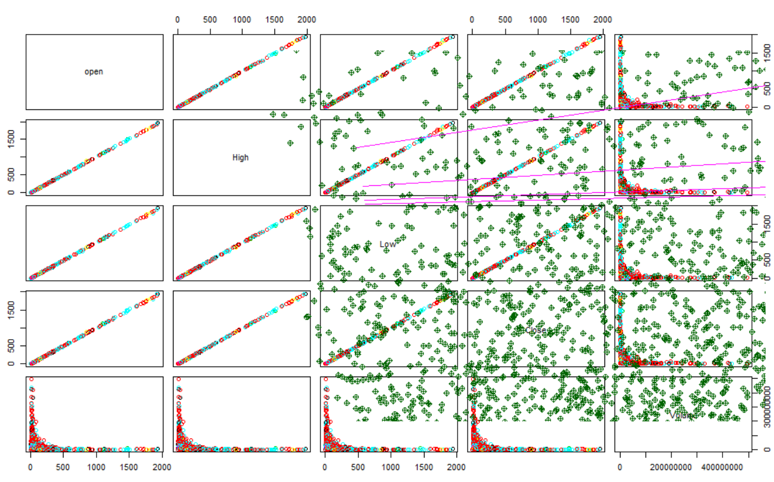
<u>Limitation</u>: It was challenging to map more than 5 variables in a clustering chart – the key variables (volatility & ON_PPC) could not be captured in the chart below – which made the chart plot less useful.


```
Cluster sizes:

[1] "8098 102386 2331 1927 65683 737 6802 2168 1754 20466"

Data means:

       open        High         Low       Close      Volume    Adjusted
0.043698124 0.043764123 0.043987077 0.043792787 0.011390931 0.040135184
 Volatility   Dividends      ON_PPC
0.035416030 0.006628185 0.392042150

Cluster centers:

         open       High        Low      Close      Volume    Adjusted
1  0.01609991 0.01648120 0.01581672 0.01611960 0.023074059 0.013373868
2  0.01710462 0.01713751 0.01721555 0.01714485 0.013391147 0.014307049
3  0.01050156 0.01059775 0.01045047 0.01050034 0.164794708 0.009164876
4  0.04395180 0.04395646 0.04424581 0.04403101 0.008290425 0.041285596
5  0.03860041 0.03861447 0.03891687 0.03869412 0.005413192 0.034162415
6  0.82010688 0.82098539 0.82587030 0.82155784 0.001577265 0.821557842
7  0.14571484 0.14600334 0.14651397 0.14597216 0.002997334 0.143084923
8  0.29287297 0.29315104 0.29483071 0.29342391 0.001742851 0.289259846
9  0.51976416 0.52043452 0.52314435 0.52090077 0.001217580 0.501276113
10 0.07871503 0.07881348 0.07926253 0.07888660 0.003803185 0.074256202
   Volatility     Dividends    ON_PPC
1  0.13244210 0.00000000000 0.3928479
2  0.03305692 0.00005254951 0.3919905
3  0.06702872 0.00000000000 0.3924286
4  0.02973600 0.70943998814 0.3916255
5  0.02753662 0.00009038747 0.3920493
6  0.03317629 0.01348041375 0.3919954
7  0.03594380 0.00005455124 0.3920990
8  0.03398906 0.00053912706 0.3919984
9  0.03373692 0.00962548471 0.3921376
10 0.03124923 0.00003626087 0.3919332

Within cluster sum of squares:
```

Within cluster sum of squares:

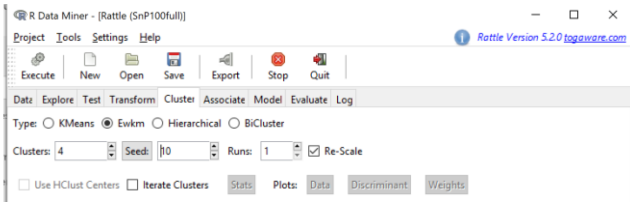
[1] 46.34335 70.02040 23.55418 62.58690 40.09805 29.81370 25.71628

[8] 30.84591 50.33915 29.06806

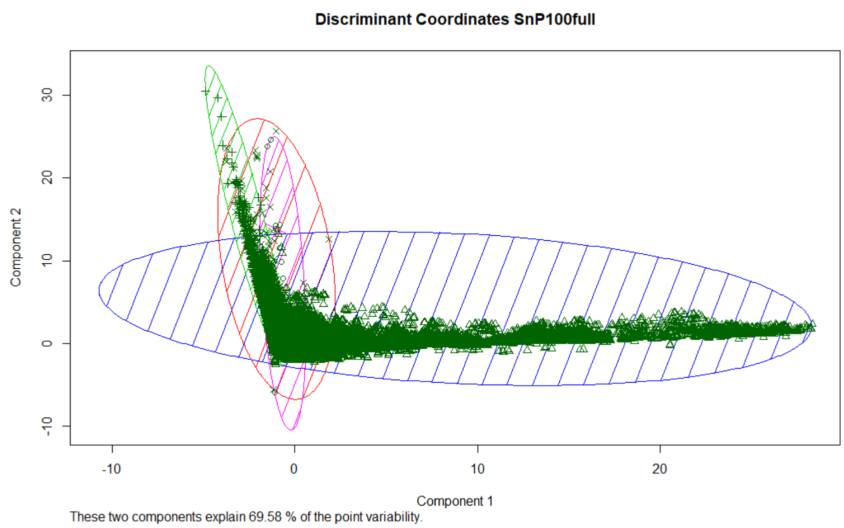


EWKM

<u>Take-away</u>: Because of the limitations in charting Kmeans cluster – I tried using the EWKM - Entropy Weighted K-Means which is more useful for high dimensional data.



Below is summarized version of the findings.



4 clusters, 1 iterations, 0 restarts, 2 total iterations.

Cluster sizes:

[1] "27464 122157 34997 27734"

Data means:

open	High	Low
0.043698124	0.043764123	0.043987077
Close	Volume	Adjusted
0.043792787	0.011390931	0.040135184
Volatility	Dividends	ON_PPC
0.035416030	0.006628185	0.392042150

Cluster centers:

open	High	Low

1	0.022540517	0.022620328	0.022644125
2	0.064403462	0.064470034	0.064862211
3	0.009875018	0.009922726	0.009907357
4	0.016131883	0.016204758	0.016180303

	Close	Volume	Adjusted
1	0.022589408	0.009935015	0.019211047
2	0.064544478	0.004567639	0.060118188
3	0.009897513	0.031075741	0.008027256
4	0.016158773	0.018046617	0.013354978

	Volatility	Dividends	ON_PPC
1	0.03710539	0.00798676223	0.3924519
2	0.03002261	0.00844149761	0.3919591
3	0.04609568	0.00444670015	0.3917770
4	0.04402251	0.00004870013	0.3923368

Cluster weights:

	open	High	Low	Close	Volume	Adjusted	Volatility	Dividends	ON_PPC
1	0.18	0.18	0.19	0.19	0.05	0.18	0	0.0	0.02
2	0.00	0.00	0.00	0.00	0.77	0.00	0	0.0	0.23
3	0.19	0.19	0.19	0.19	0.00	0.21	0	0.0	0.03
4	0.18	0.18	0.18	0.18	0.00	0.18	0	0.1	0.00

Within cluster sum of squares:

[1] 0 0 0 0

Comments: ONPPC – Overnight percentage price change seem to be the most determinant in the EWKM clustering – on the other hand Volatility seems to be at par with other variables.

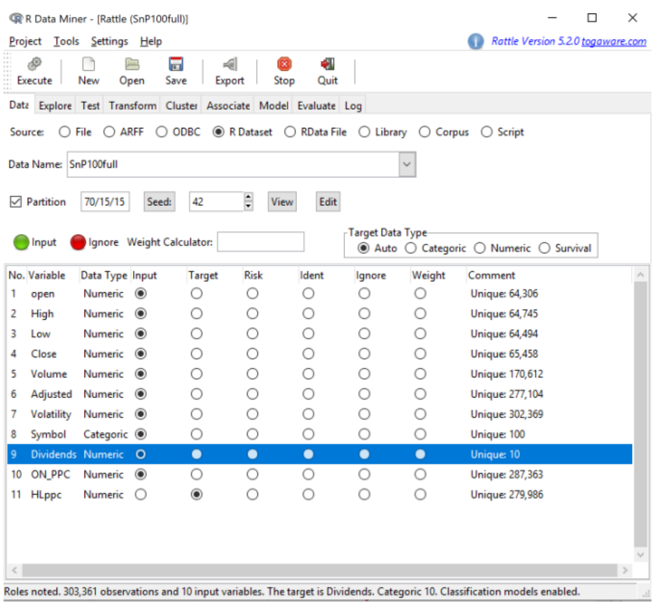
Modeling

Testing & Training

Take-aways:

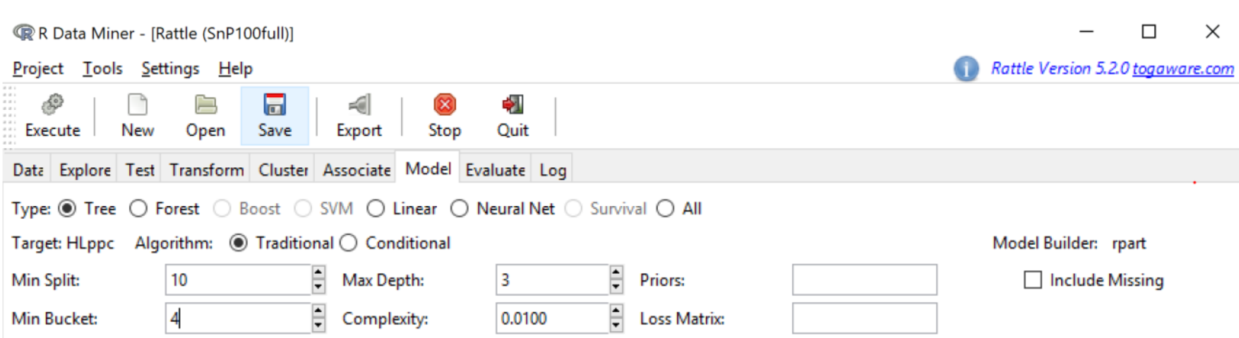
- ***Data was split in a 70% training, 15% validation & 15% testing format.***

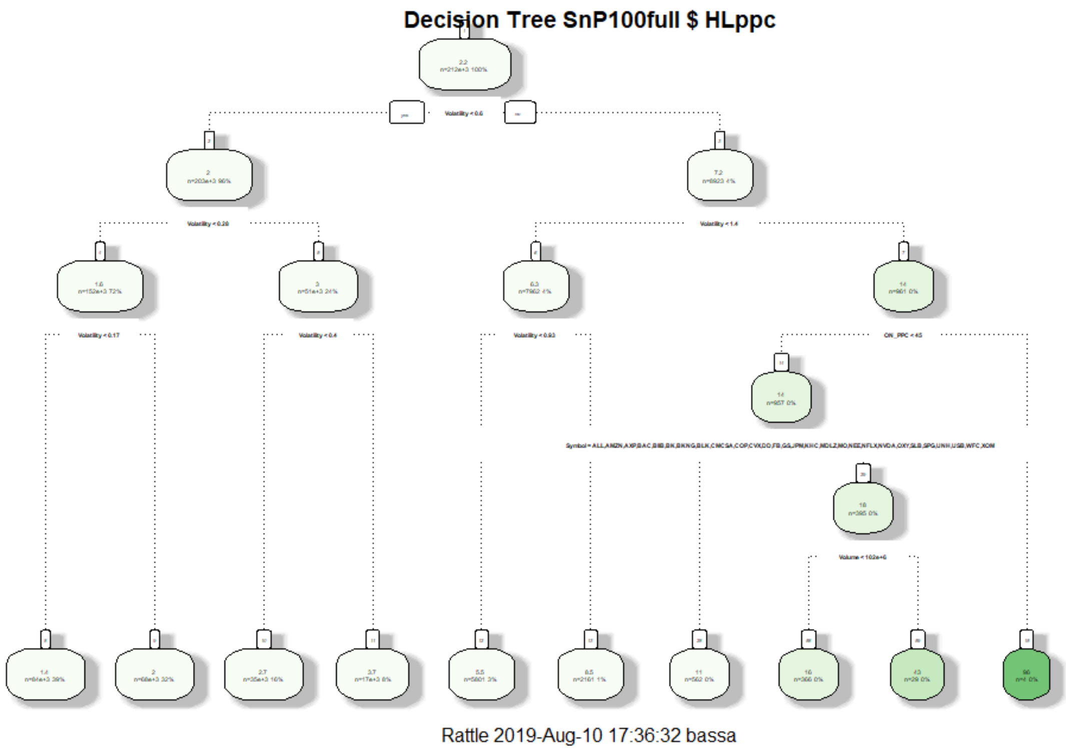
- ***We are trying to predict a stock's percentage change to screen the highest for a straddle strategy.***
- ***3 models were compatible with the nature our data frame.***
    1. ***Decision trees***
    2. ***Neural Networks***
    3. ***Linear regression***
- ***Testing and validation came back with very similar results – so only one was reported to save on real estate.***
- ***Also to save on space – the bulk of coefficient tables, nodes tables were removed.***
- ***Predict & Observe method was used to compare prediction capability of all 3 models with Pseudo R-Square score used as a benchmark.***



Decision trees

Take-away: Driven by volatility & volume variables – decision tree were able to predict at 0.476 R-square.





Summary of the Decision Tree model for Classification (built using 'rpart'):

n= 212352 node), split, n, deviance, yval

* denotes terminal node

1) root 212352 926452.60 2.207995

2) Volatility< 0.6044209 203429 322549.00 1.990545

4) Volatility< 0.2761405 152209 103667.00 1.637718

8) Volatility< 0.1741081 83749 31857.02 1.366199 *

9) Volatility>=0.1741081 68460 58082.70 1.969875 *

5) Volatility>=0.2761405 51220 143626.70 3.039031

10) Volatility< 0.4033684 34670 66522.50 2.708230 *

11) Volatility>=0.4033684 16550 65362.53 3.732015 *

3) Volatility>=0.6044209 8923 374988.00 7.165469

6) Volatility< 1.378729 7962 120587.00 6.334969

12) Volatility< 0.9283672 5801 54583.42 5.541613 *

13) Volatility>=0.9283672 2161 52551.00 8.464659 *

7) Volatility>=1.378729 961 203410.40 14.046260

14) ON_PPC< 44.62153 957 115484.80 13.704970

28)
Symbol=ALL,AMZN,AXP,BAC,BIIB,BK,BKNG,BLK,CMCSA,COP,CVX,DD,FB,GS,JPM,KHC,MDLZ,MO,NEE,NFLX,NVDA,OXY,SLB,SPG,UNH,USB,WFC,XOM 562 22652.93 10.935450 *

29) Symbol=AIG,C,COF,F,GE,MET,MS 395 82388.08 17.645390

58) Volume< 1.019247e+08 366 32205.36 15.597690 *

59) Volume>=1.019247e+08 29 29279.57 43.488750 *

15) ON_PPC>=44.62153 4 61144.00 95.701280 *

Regression tree:

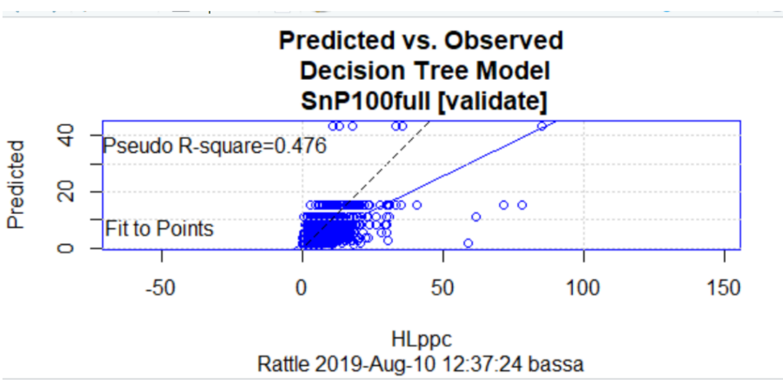
rpart(formula = HLppc ~ ., data = crs$dataset[crs$train, c(crs$input,

crs$target)], method = "anova", model = TRUE, parms = list(split = "information"),

control = rpart.control(minsplit = 10, minbucket = 4, usesurrogate = 0,

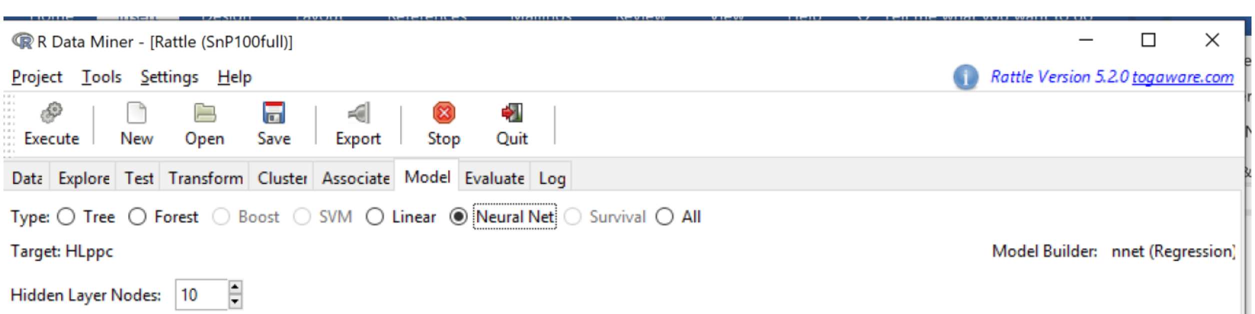
maxsurrogate = 0))

Test – validation test came back with very similar results



Neural Networks

Take-away: Neural Network was able to predict at 0.36 R-square (vs. 0.47 for decision trees)



Summary of the Neural Net model (built using nnet):

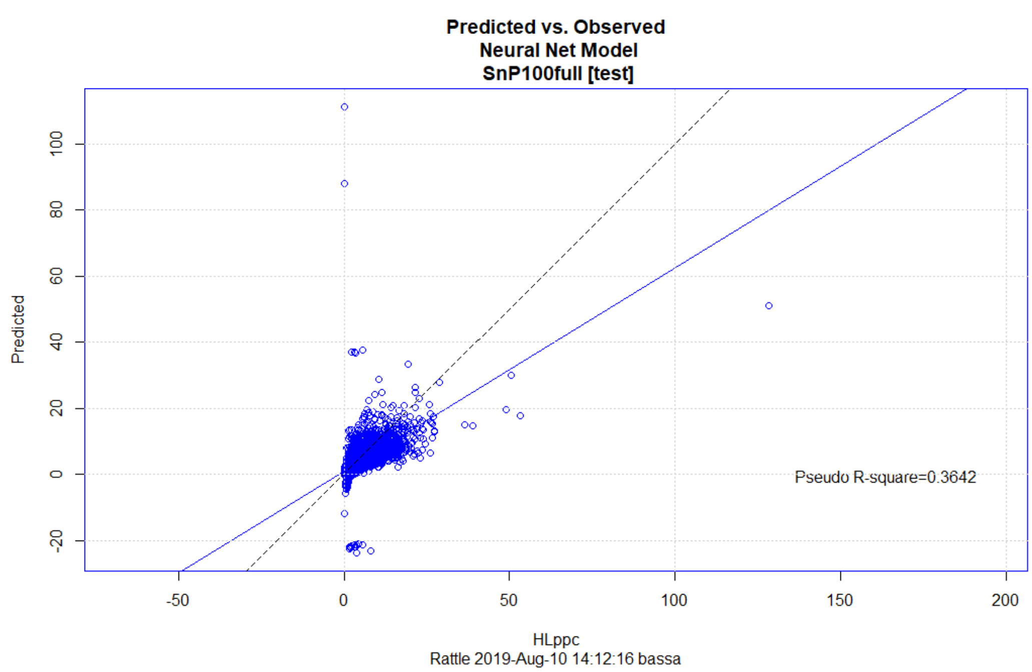
A 108-10-1 network with 1209 weights.

Inputs: open, High, Low, Close, Volume, Adjusted, Volatility, SymbolABBV, SymbolABT, SymbolACN, SymbolADBE, SymbolAGN, SymbolAIG, SymbolALL, SymbolAMGN, SymbolAMZN, SymbolAXP, SymbolBA, SymbolBAC, SymbolBIIB, SymbolBK, SymbolBKNG, SymbolBLK, SymbolBMY, SymbolC, SymbolCAT, SymbolCELG, SymbolCHTR, SymbolCL, SymbolCMCSA, SymbolCOF, SymbolCOP, SymbolCOST, SymbolCSCO, SymbolCVS, SymbolCVX, SymbolDD, SymbolDHR, SymbolDIS, SymbolDOW, SymbolDUK, SymbolEMR, SymbolEXC, SymbolF, SymbolFB, SymbolFDX, SymbolGD, SymbolGE, SymbolGILD, SymbolGM, SymbolGOOG, SymbolGOOGL, SymbolGS, SymbolHD, SymbolHON, SymbolIBM, SymbolINTC, SymbolJNJ, SymbolJPM, SymbolKHC, SymbolKMI, SymbolKO, SymbolLLY, SymbolLMT, SymbolLOW, SymbolMA, SymbolMCD, SymbolMDLZ, SymbolMDT, SymbolMET, SymbolMMM, SymbolMO, SymbolMRK, SymbolMS, SymbolMSFT, SymbolNEE, SymbolNFLX, SymbolNKE, SymbolNVDA, SymbolORCL, SymbolOXY, SymbolPEP, SymbolPFE, SymbolPG, SymbolPM, SymbolPYPL, SymbolQCOM, SymbolRTN, SymbolSBUX, SymbolSLB, SymbolSO, SymbolSPG, SymbolT, SymbolTGT, SymbolTXN, SymbolUNH, SymbolUNP, SymbolUPS, SymbolUSB, SymbolUTX, SymbolV, SymbolVZ, SymbolWBA, SymbolWFC, SymbolWMT, SymbolXOM, Dividends, ON_PPC.

Output: HLppc.

Sum of Squares Residuals: 22887008171185948.0000.

Neural Network build options: skip-layer connections; linear output units.

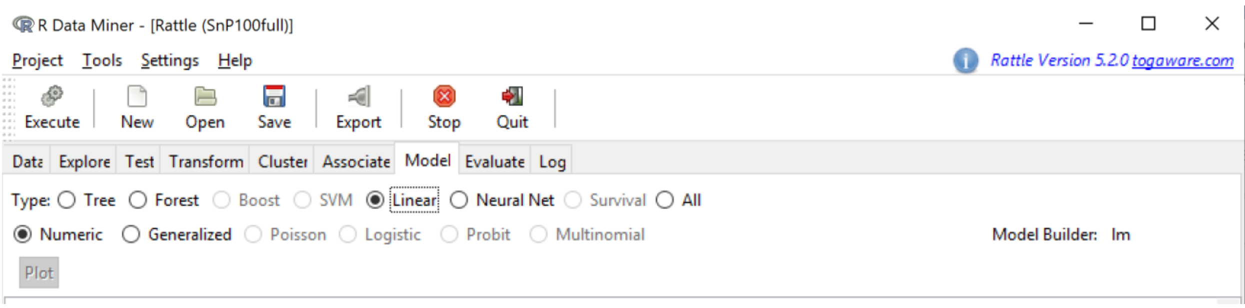


Linear

Take-away: Compared with decision trees and neural network, Linear regression was able to produce the highest R-square of 0.58.

Linear model also produced a clear list of predicted percentage change:

	Estimate	*Std. Error*	*t value*
SymbolABBV	*1.734e+00*	*5.441e-02*	*31.869*
SymbolABT	*1.573e+00*	*4.605e-02*	*34.162*



Summary of the Linear Regression model (built using lm):

Call:

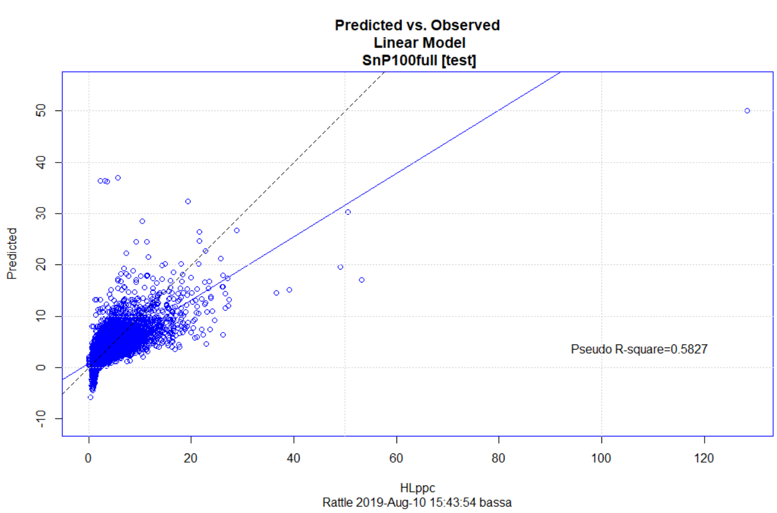
lm(formula = HLppc ~ ., data = crs$dataset[crs$train, c(crs$input,

crs$target)])

Residuals:

Min	1Q	Median	3Q	Max
-35.298	-0.478	-0.089	0.336	260.503

Test



Conclusion

Driven by volatility & volume variables – decision tree were able to predict at 0.476 R-square.

Neural Network was able to predict at 0.36 R-square (vs. 0.47 for decision trees)

Compared with decision trees and neural network, Linear regression was able to produce the highest R-square of 0.58.

Linear model also produced a clear list of predicted change – below is the top absolute estimate.

Rank	Symbol	Estimate	Std. Error	t-value
1	**COF**	2.0620	0.047	44.080
2	**SPG**	1.9810	0.047	42.116
3	**AIG**	1.9790	0.048	41.500
4	**MS**	1.9280	0.046	42.078
5	**UNH**	1.9270	0.046	41.560
6	**AGN**	1.9090	0.047	40.952
7	**NVDA**	1.9010	0.047	40.861
8	**BLK**	1.8820	0.047	39.765
9	**LMT**	1.8650	0.047	39.763
10	**BKNG**	1.8610	0.052	35.540