Capstone Project 1: SP500 Price Swing Analysis - Exploratory Data Analysis - Inferential Statistics – Tom Preston

Study Background:

This study will analyze the SP500 “e-mini” contract to identify hidden price patterns in the data. The e-mini is a synthetic financial futures contract electronically traded on the CME (Chicago Mercantile Exchange). The e-mini is used by institutional and day traders for short term and long term trading and hedging. The focus of this study will be day trading short-term (ie – minutes) swings that the day trader can capitalize on for a higher volume of short term trades with less risk than longer term (ie – hours, days).

The margin requirements to day trade the e-mini is typically on 10% ($500) per contract compared to $5,000 per contract margin required to hold an e-mini contract overnight. The e-mini moves in .25 point “tick” increments (i.e., 2015.25 to 2015.50). Each .25 point move is worth $12,50 per contract and each full point move it worth $50 per contract. Day traders can adjust the number of contracts traded per trade signal (typically 1 contract up to 20 or more contracts per trade). The SP e-mini is the most liquid financial futures trading averaging over 1.5 million contracts traded daily.

Over the past several years, I have considered developing and automated trading “bot” as a side project. Many day trading web-sites focus on a day trading approach of identifying high probability trades where the day trader can get 3 ticks (.75) points risking six ticks (1.50) points per contract. This trading approach has to be at least 67% correct to break even (before commissions – typically $3 - $4 per contract). As shown in the left column of the table below, if a trading system can increase the number of profitable trades to 75%, the profit increases quickly due to less 6 tick losses. As shown on the right column of the table below, if the trader can average 3.5 ticks profit per trade, this definitely helps the system profitability.

Scenario A	**Trades Per Week**	**Ticks**	**P/L**		**Scenario D**	**Trades Per Week**	**Ticks**	**P/L**
Wins	67	**3**	201		Wins	67	**3.5**	234.5
Losses	33	-6	-198		Losses	33	-6	-198
		Net Ticks	3				**Net Ticks**	36.5
Scenario B					**Scenario E**			
Wins	75	**3**	225		Wins	75	**3.5**	262.5
Losses	25	-6	-150		Losses	25	-6	-150
		Net Ticks	75				**Net Ticks**	112.5
Scenario C					**Scenario F**			
Wins	60	**3**	180		Wins	60	**3.5**	210
Losses	40	-6	-240		Losses	40	-6	-240
		Net Ticks	-60				**Net Ticks**	-30

This study will look at the price swings of the e-mini to estimate the viability of this approach and seeing what reasonable profit targets ranges might be based on various attributes (trades with and against the trend, time of day, etc).

Data Source:

There are several trading platform packages that plot e-mini. I use Ninjatrader (https://ninjatrader.com). Ninjatrader plots the e-mini data and related studies however that data must be exported for use in my Capstone study. Ninjatrader plots data on many time intervals (tick data, minutes, hours, daily, weekly, etc). Most day traders trade small time intervals (1 minute, 3 minute bars). I personally monitor the 687 tick chart (ie price trades / ticks 687 times and then a new bar is drawn). Given the high trading volume, the 687 tick chart is approximately a 20 second bar chart. While this may seem like too short of a time interval, the 687 tick chart shows the price movements more thoroughly than arbitrary time charts and it helps keep the risk lower. This study will review price data from 1/1/2019 – 1/18/2019

Data Wrangling:

Data Setup and Cleaning

Ninjatrader has a data export feature. It will export each individual price tick of the e-mini data. Each day has approximate 1 million "ticks" of data. I exported a CSV file of each data tick for the data range of 1/1/2019 – 1/18/2019. Python Pandas has the ability to resample data down to specific time intervals (i.e. – seconds, minutes, hours, etc). The 20 second bar mimics the price action of the 687 tick mentioned above. I used the Pandas resample methods to read in the large CSV file of tick data to create a file of 20 second price bars for the study. The data has a date-time stamp, open, high, low, and close for each 20 second bar. Below is a sample of the data.

df.head()

Date_Time	Open	High	Low	Close
2019-01-01 23:00:00	2493.00	2493.00	2492.75	2493.00
2019-01-01 23:00:20	2493.25	2493.50	2493.25	2493.50
2019-01-01 23:00:40	2493.25	2493.25	2493.00	2493.25
2019-01-01 23:01:00	2493.50	2493.50	2493.25	2493.25
2019-01-01 23:01:20	2493.25	2493.50	2493.25	2493.25

The e-mini is electronically traded on a 24 hour basis. The trading volume is highest during the U.S. stock market trading hours (9:30 am EST – 4:00 pm EST). There is lighter trading volume during the other times and volume typically picks up during European and Asian financial markets trading. The data file from 1/1/2019 – 1/18/2019 had 72,361 20 second bars of data. 20,408 bars of this data were null. This due to weekend hours where no trading took place. I could have dropped the nulls values but I left them in as it shows in the price chart where volume increase and decreases around the null data points. This showed what trading times where the most affected by the null values.

Adding Features:

Moving Averages: The basic e-mini price data needs additional features for the analysis. First, 2 exponential moving averages (21 bar and 55 bar) moving averages were added to visually show the shorter and longer term trends.

```
df['21ema'] = pd.Series.ewm(df['close'], span=21).mean()
df['55ema'] = pd.Series.ewm(df['close'], span=55).mean()
```

Market Trend: Next, the overall market trend was calculated. If the price bar high goes 1.5 points above the 55 EMA, then the trend was set to long. The trend stays long until the market low goes 1.5 points below the EMA and the trend turns to short. The main concept around this approach for developing the market trend is that a simple moving average crossover (i.e. the 21 EMA crosses over the 55 EMA and makes the trend long) can have too many trend changes in sideways price action. This 1.5 point price move above or below the 55 EMA helped to minimize but not totally eliminate frequent trend changes.

Python code was written to calculate the trend, write the trend changes (long to short, etc) to a dataframe and then merge the trend dataframe with the price dataframe. The trends changes were marked with a green diamond for up trend and a red diamond for down trends. The chart below shows a sample of the price chart, up and down trend markers, and the 21 and 55 EMA. This is a 20 second bar price chart of the SP500 from Jan 2019.

Price Swings: The market trend is identified, and this feature was added to the dataframe to allow a categorical grouping for price swings. A price swing is a shorter price move up or down. A price swing can be identified by several variables (size in points of the swing, number of bars in the swing) etc. For the study, we identify a price up swing as begins when price goes above the highest high for the past 5 bars and a price down swing begins when price goes below the lowest low for the past 5 bars.

Once the beginning and ending of a price swing is known, a blue line is plotted to show the start of an upswing and a red line is plotted to show the start of a downswing (and the end of the upswing). The chart below shows the blue and red lines for the beginning of upswings and downswings. The size (in points) and length (number of bars) is calculated and maintained in

the dataframe.

Calculating the location of the 5 bar highest high and lowest low (excluding the current bar) required rolling 5 period min and max functions. Also, the hour of each 20 second price bar was extracted for subsequent analysis.

```
df_mini['trail_hi'] = df_mini['high'].rolling(5,min_periods=1).max()
df_mini['trail_hix'] = df_mini['high'] - df_mini['trail_hi'].shift(1)
df_mini['trail_lo'] = df_mini['low'].rolling(5,min_periods=1).min()
df_mini['trail_lox'] = df_mini['trail_lo'].shift(1) - df_mini['low']
df_mini['hour'] = df_mini.index.hour
```

Price Swing Calculations: Based upon the new features (moving averages, trend indicator, and 5 period trailing high and low calculations), complex python code was developed to read each price bar of the dataframe and to determine the swing information (trend, up / down swing, points, and length). The swings were written to a new dataframe for additional analysis.

PLOTLY EDA Challenges:

Date_time Index Plotting Defect: Plotly is used for the data visualizations. The ability to plot price bar data and plot additional items on the screen (diamonds for the up and down trends, lines showing the beginning and ending of price swings) was a key reason for using Plotly. The challenge was that Plotly has a defect for plotting time series data. My dataframe has a data_time index. Unfortunately, Plotly would not correctly plot the time series data index, the

price bars and indicators were not plotting on the correct time. In reviewing Stack Overflow, this appears to be a known issue. The solution was to convert the dataframe index to string. This caused several extra steps to be created however this solved the issue.

Plotting Lines on the Chart: Plotting the swing beginning and ending lines on the chart provide a great visual reference when reviewing the price data. The challenge is that each line plotted needs to have X and Y coordinates and related data attributes (color, size, etc) for each individual item. This led to the creation of a large list of data to plot all the related lines.

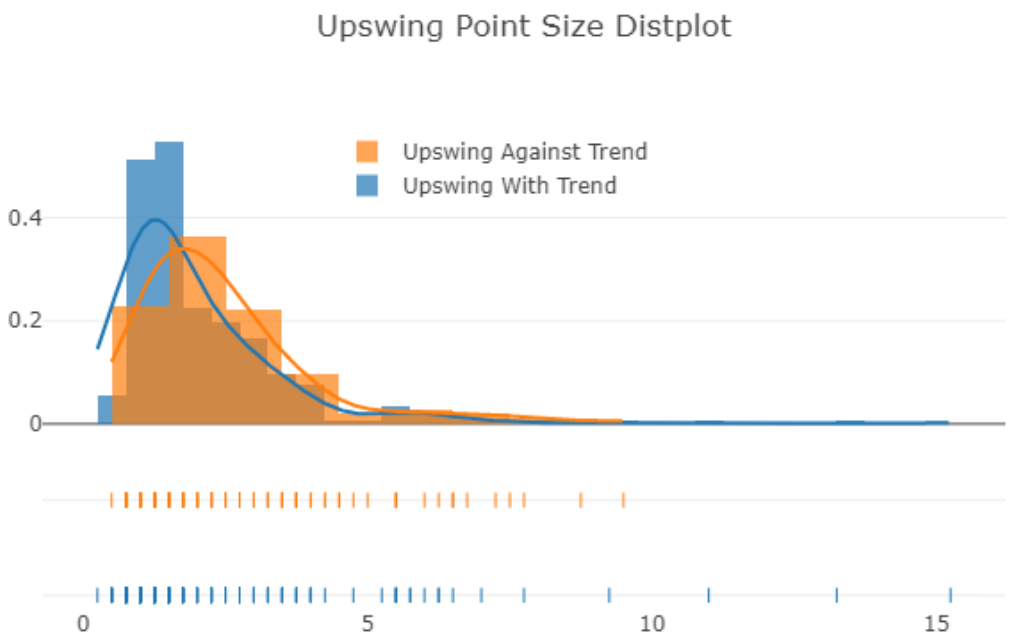
EDA:

All swings histogram: After adding features and developing python code to identify both price swings and trend swings, I developed various analyses to better understand the price swings.

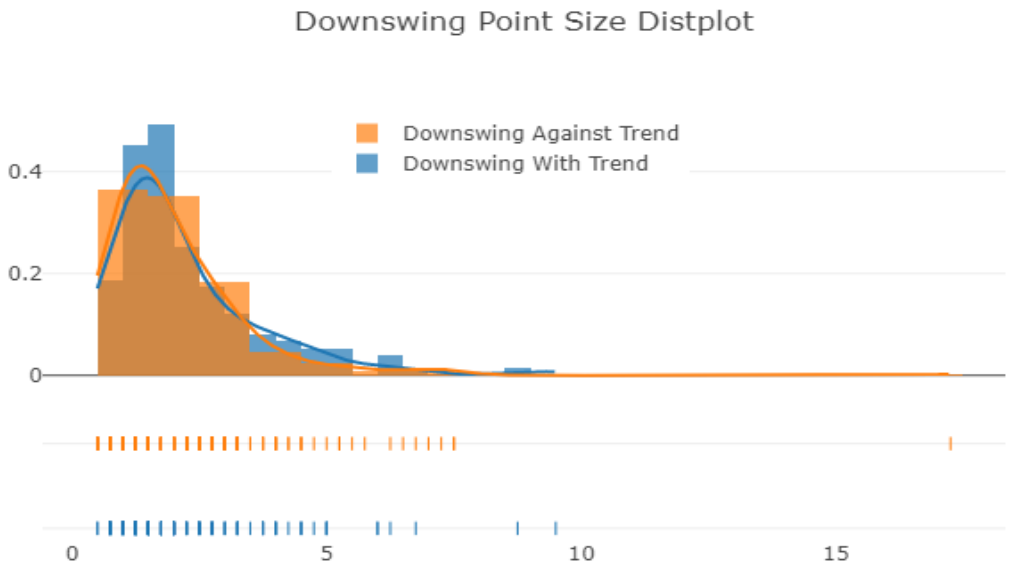
First, all the swings were grouped by percentiles. As shown in the table below, there were 956 swings identified. The mean swing is 2.15 points (8 ticks) and 11.4 bars (@ 20 seconds per bar = 3.8 minutes per swing). As shown by the chosen histogram percentages, 30 % of swings are 1.25 points or less. This is too small to reasonably day trade however additional analysis is needed to see if a majority of these smaller swings (smaller point size and shorter time length) are during off-hours.

	Points	**Swing_len**
count	956.000000	956.000000
mean	2.155073	11.488494
std	1.603521	12.188522
min	0.250000	0.000000
10%	0.750000	3.000000
20%	1.000000	4.000000
30%	1.250000	6.000000
40%	1.500000	7.000000
50%	1.750000	9.000000
75%	2.750000	15.000000
80%	2.750000	17.000000
90%	3.750000	23.000000
max	17.250000	205.000000

The distplot of all the upswings below clarifies the grouping of data. When viewed in aggregate, over 50% of the swing sizes are 1.75 points or less.

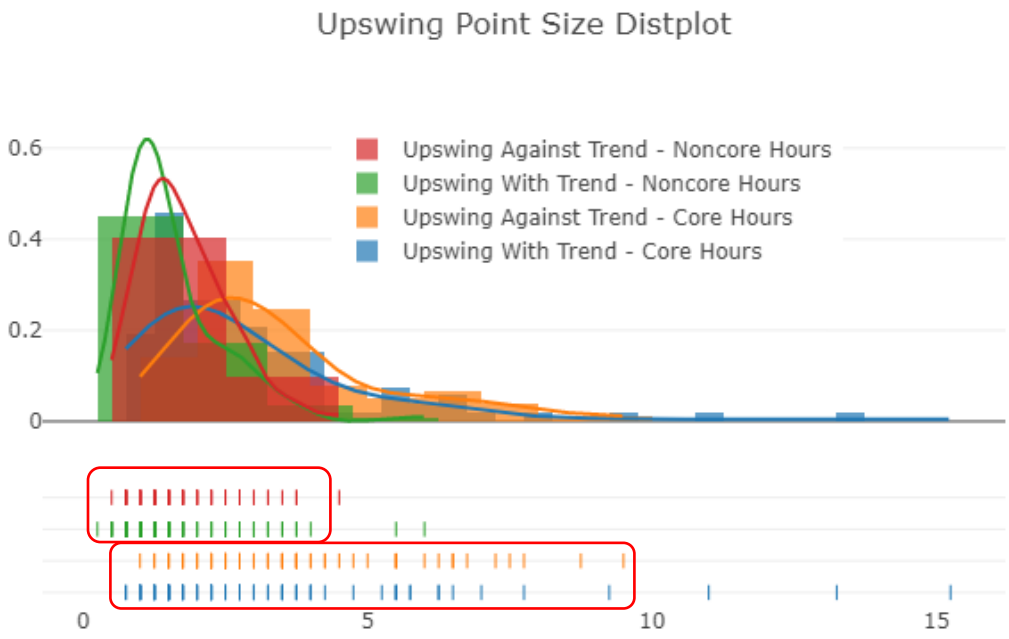


The distplot of all the downswings shows a slightly different but similar view.

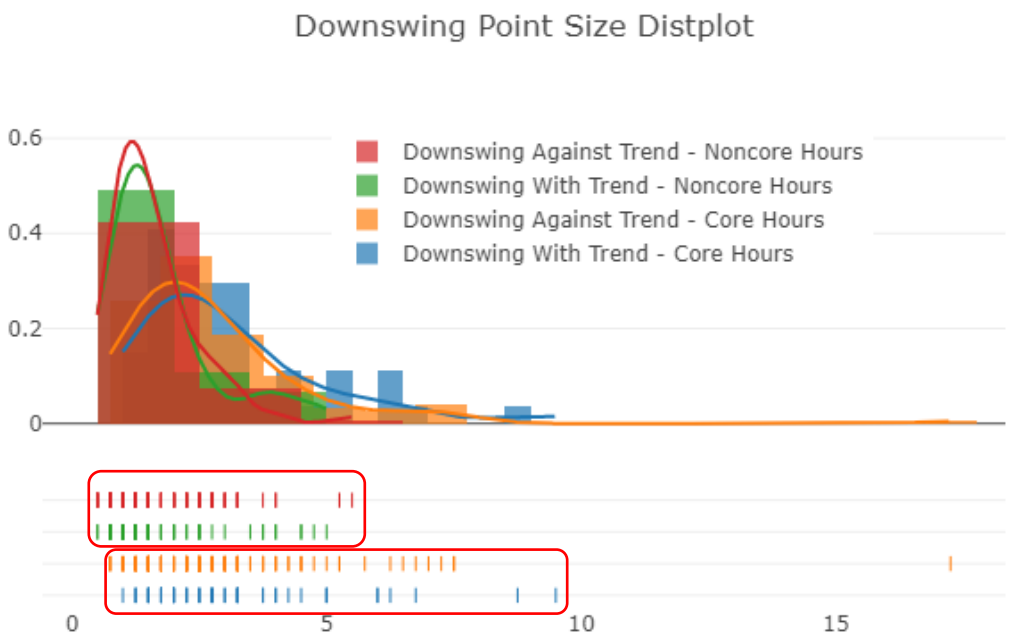


Based on the two distplots above, we need to further segment the data to determine the viability for day trading. First, we will break the data into 2 groups – “Core” (price swings originating from 9 am to 3:59 pm EST) and “Noncore” (price swings originating from 4:00 pm EST to 8:59 am EST). Given the higher trading volume during Core hours, I expect the swing sizes to be larger and the swing time length (i.e., the number of 20 second price bars) to be longer. Secondly, the upswings will be separated from the downswings. The upswings and downswings are often impacted by 2 things, the overall trend (up or down) and that prices often fail at a faster rate than they rise. Separating these price swings will help with the EDA.

Core and Noncore Histogram - Upswings: The histogram below shows 4 overlaid histograms. The first two (red and green) are Noncore Upswings with and against the trend. These show a tighter distribution (typically in the .5 to 3.5 point range) than the Core Upswings. The Core Upswings have many more observations in the 4.0 – 10.0 range.

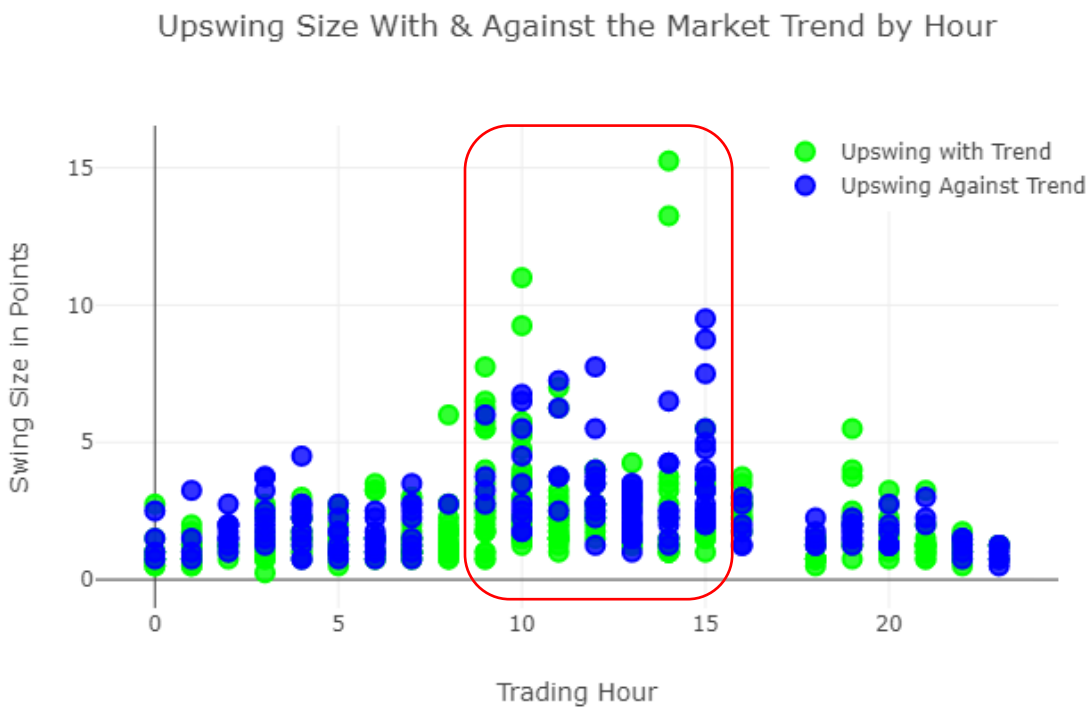


Core and Noncore Histogram - Downswings: The Downswings have a different distribution. The chart below shows 4 overlaid Downswing histograms. The first two (red and green) are Noncore Downswings with and against the trend. These also show a tighter distribution (typically in the .5 to 4.0. point range) than the Core Downswings. The Core Downswings have many more observations in the 1.5 – 10.0 range.



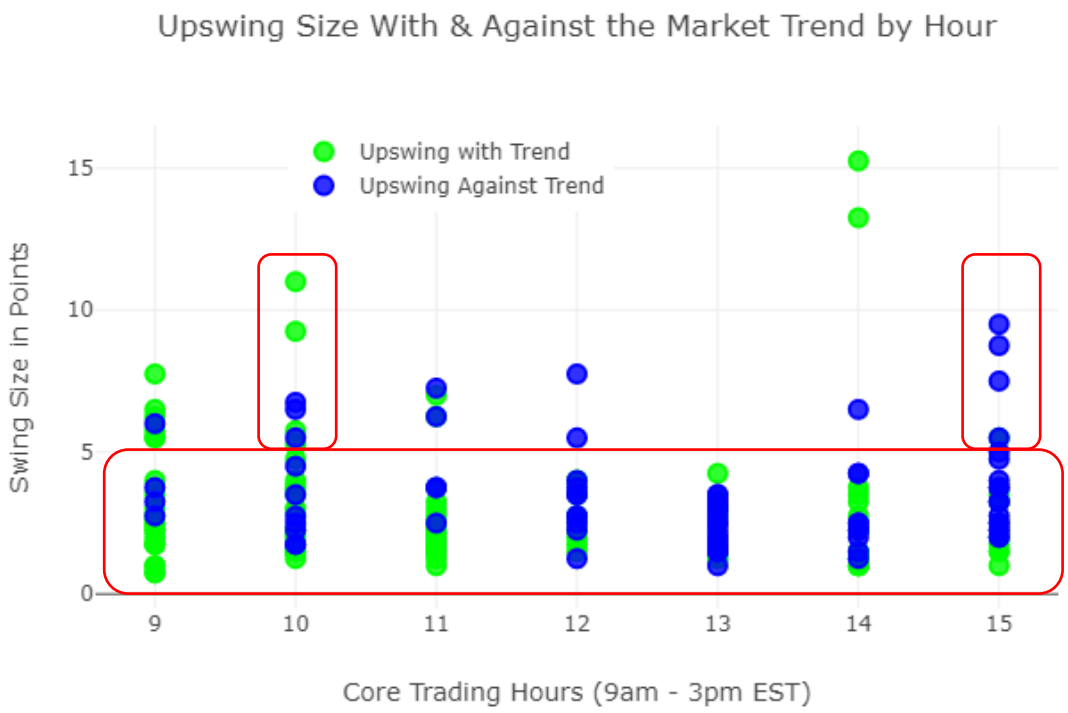
Price Swings and Market Trend: The price swings were calculated for all up and down price swings. The market trend was calculated and merged with the price swing dataframe allowing an analysis of price swings with and against the market trend. The overall hypothesis is that up swings when the trend is long will be larger than those against the market trend. The second hypothesis is that grouping the trading signals by the trading hour they start in will show the best times to trade. In order to analyze the data, the swings were divided into up swings and down swings.

Upswing Swing Size: The Upswing size with and against the market trend are shown below.

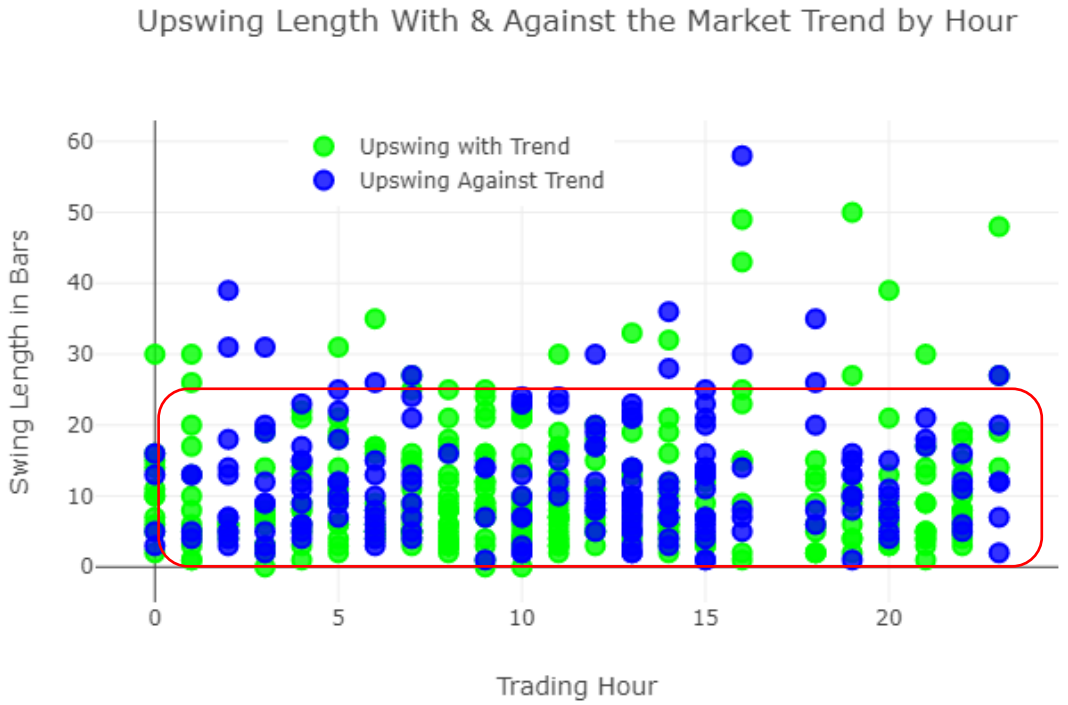


The chart above shows that the busiest trading time is for upswings is between 9:00 EST and 3pm EST (note – typically data traders trade until 3 pm as the final hour of trading is typically institutional traders hedging their positions. This can lead to erratic price swings in the final hour of trading. The chart also shows a majority of trades outside 9 am EST to 3 pm EST exceed 4 points while many trades exceed 5 points inside the core US market trading hour.

Upswing Swing Size – Core Hours: When the Core hours Upswings are plotted separately, it's easier to see the patterns, both in size and timing. For size, a majority are 4.0 points or less. Many Upswings to extend to 6 – 10 pts or greater in specific time frames (typically 10 am and 3 pm). This may be explained by larger institutional players establishing trading positions in the morning and then hedging / closing their position before market close each day. The outlier moves in the 2pm hour correspond to interest rate related announcements by the U.S. Federal Reserve.

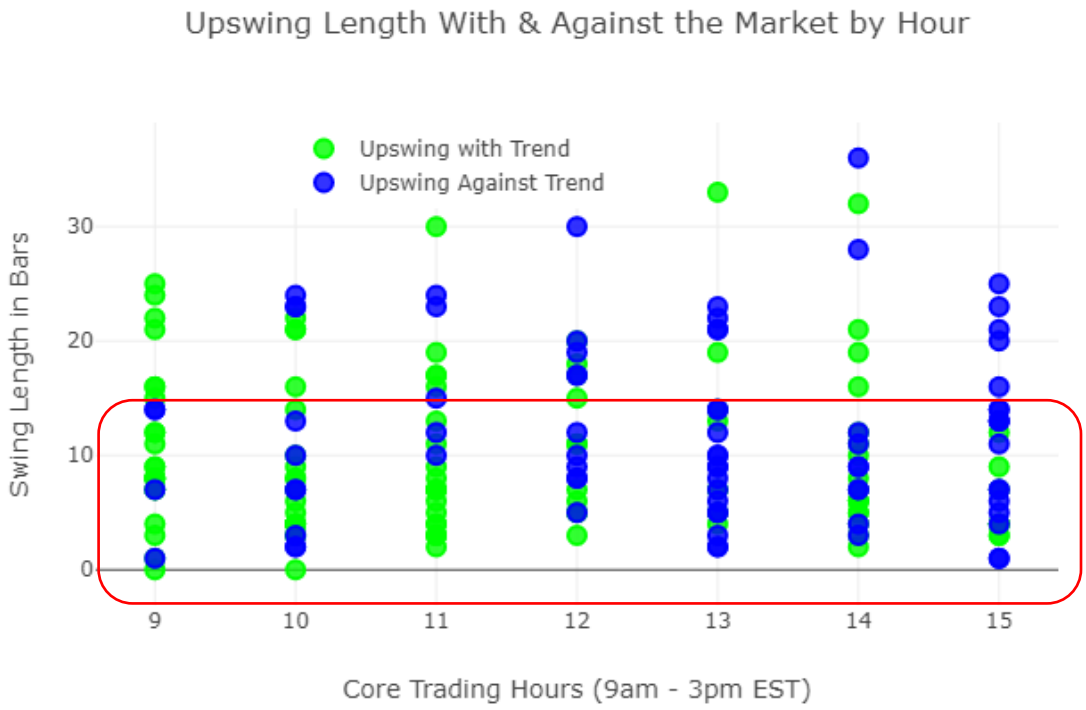


Upswing Swing Length: The Upswing swing length with and against the market trend are shown below.

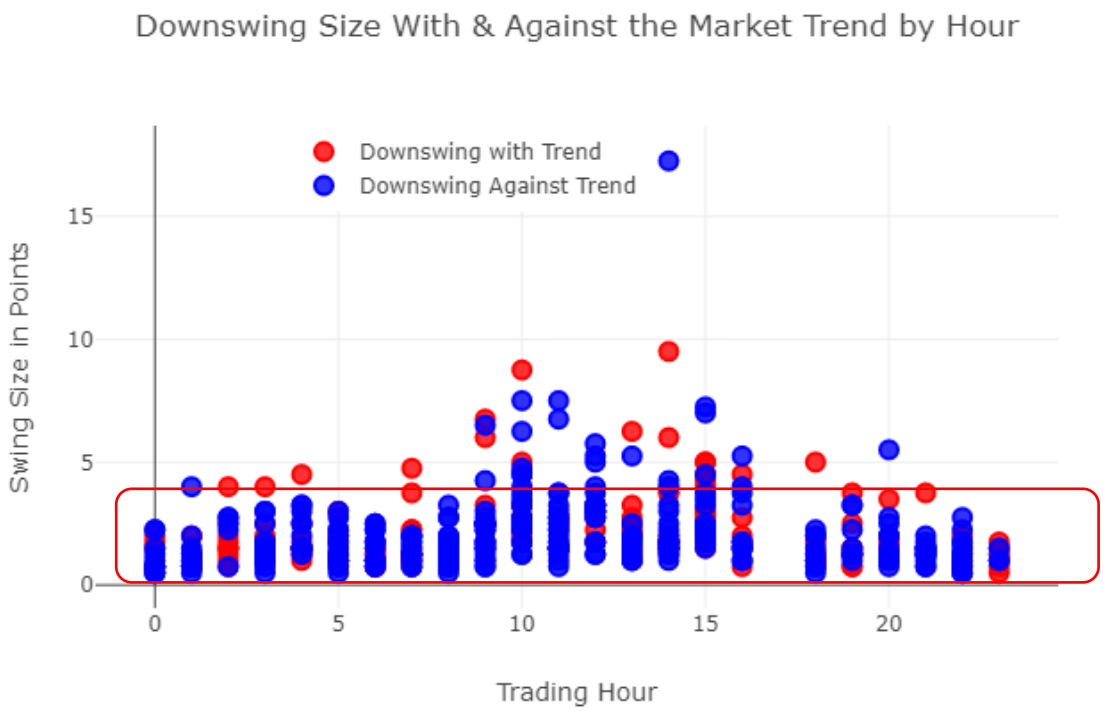


The most interesting item in the chart above is the approximately similar trend length across all time frames. While the price swings points are larger in during U.S. market hours, it looks like all time frames have a similar length. While this means the price movement is typically more volatile during U.S. Market hours, it does mean the non-market hours move in a similar price structure however it's typically smaller sized movements.

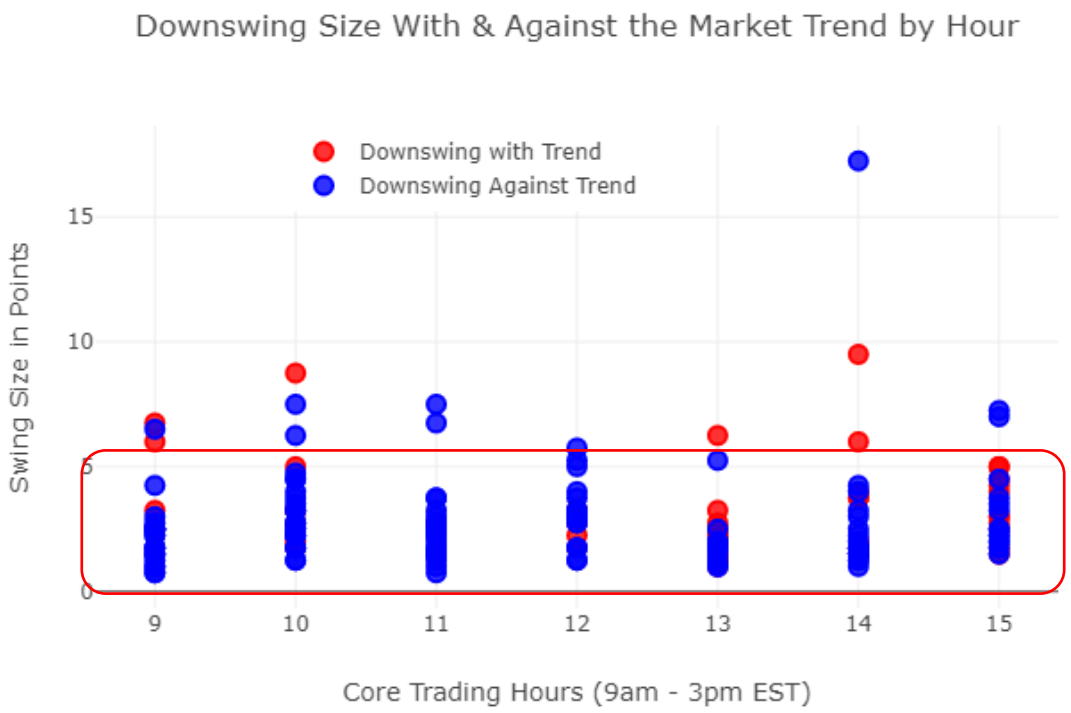
Upswing Swing Length – Core Hours : When focusing only on the Upswing Core hours, it's clearer that a majority of swings only go for 15 – 20 bars (5 – 7 minutes) The Upswing swing length with and against the market trend are shown below.



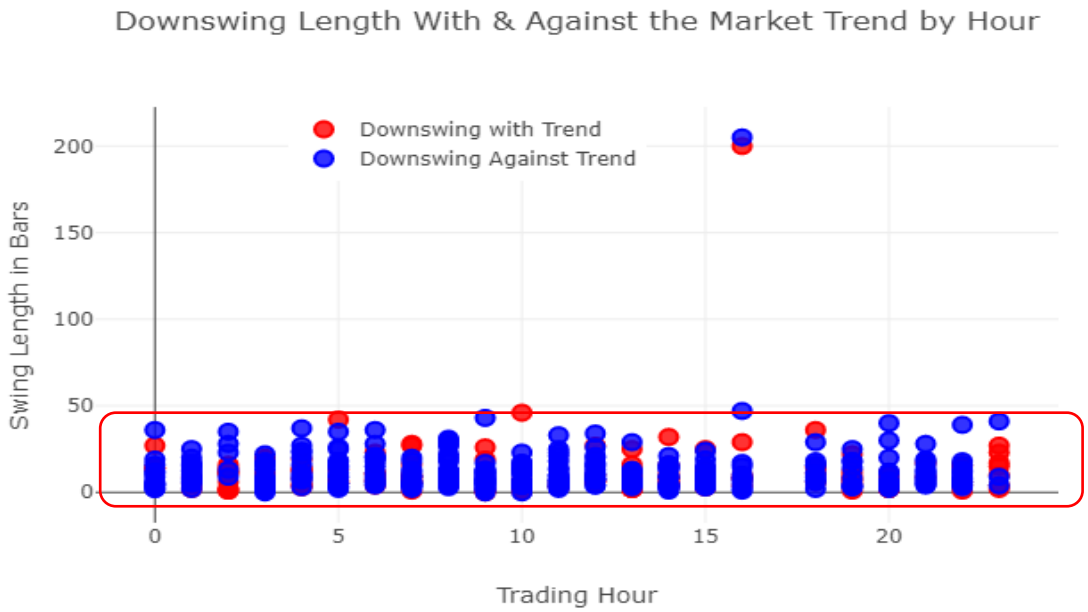
Downswing Size: The chart below shows downswing size with and against the trend. The most interesting observation from this chart is that there are many more downswings against the trend than with the trend. This could mean that most of the trends were up during the study period. The swings had the biggest size during the core U.S. market hours their size is less than the upswings during the same period.



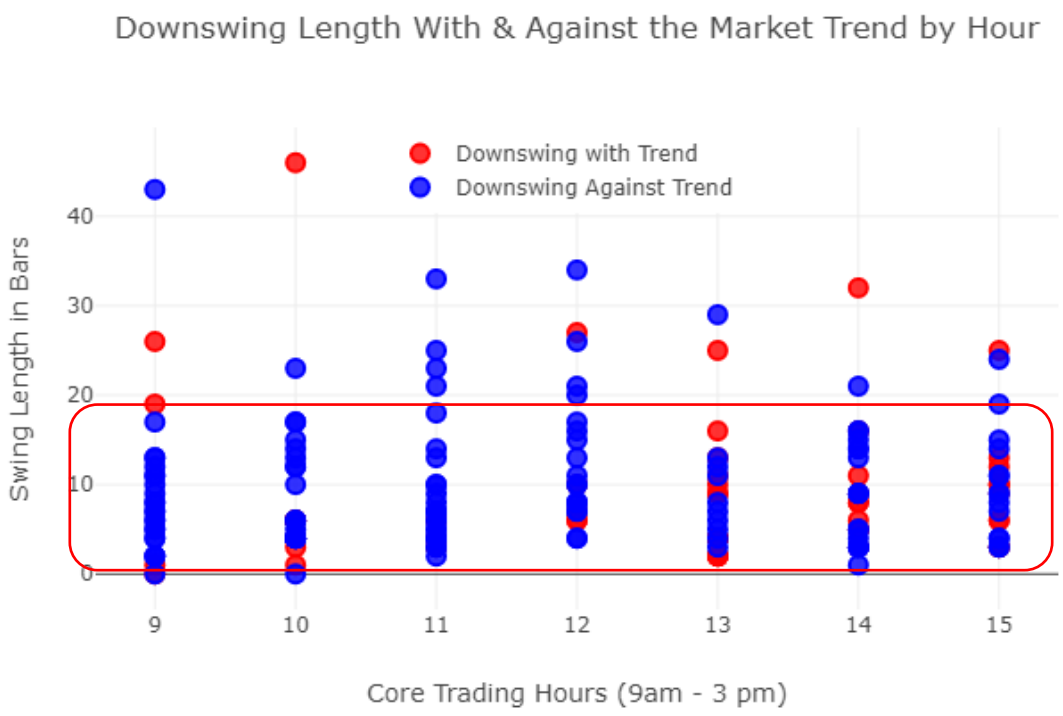
Downswing Size – Core Hours: The chart below shows Downswing Core hours size with and against the trend. This clarifies that a majority of swings are 4 points or less. The swings above 4 points are approximately spread against the Core hours.



Downswing Length: The chart below shows downswing length with and against the trend. With the exception of 2 outliers (200 bars – approx 60 min long in non-core trading hoursswing during a U.S Federal Reserve interest rate announcement), the downswings are typically less than 50 bars (16 mins). This is consistent across all hours of trading.



Downswing Length – Core Hours: The chart below shows Downswing Core hours length with and against the trend. Most Downswings are 20 bars (7 mins or less).



Analysis Summary

Up and Down Swings: The charts above shows the swing size and number of 20 second bars for up swing and down swings, both with and against the trend. The trend is either long or short. The chart below condenses that information so the swing information can be more easier compares. For day trading purposes, we need the ability to make at least 3 ticks (.75 points) of profit on a trade. The All Swings entry below shows that 50% of all 956 swings on average move 1.75 points so we can estimate these swings offer at least .75 points profit. The key is trading the core hours (9 am – 3 pm EST) as the 50% move size increases for both trades with and against the trend. Interestingly, the number of bars in a swing length (as measures in 20 second bars) stays roughly the same for core hours versus non core hours. This can mean that the core and non core hours moves follow the same basic structure however the increased size of core hours show there's more price volatility and hence more profit opportunity during core hours.

Swing Type	Sample Size	Move Size in Points (% Levels)			# of 20 second bars (% Levels)		
		20%	**50%**	**80%**	**20%**	**50%**	**80%**
All Swings	956	1.00	1.75	2.75	4	9	7

Up swings – Core Hours – with trend	106	1.50	2.25	3.75	4	8	16
Up swings – Non Core Hours – with trend	187	.75	1.25	2.25	4	8	16
Up swings – Core Hours – against trend	77	2.00	3.00	4.50	5	10	20
Up swings – Non Core Hours – against trend	108	1.25	1.5	2.5	5	11	18
Down swings – Core Hours – with trend	55	1.75	2.75	4.00	3	6	12
Down swings – Non Core Hours – with trend	96	1.00	1.50	2.00	4	8	15
Down swings – Core Hours – against trend	128	1.5	2.5	3.75	4	8	15
Down swings – Non Core Hours – against trend	199	1.00	1.25	1.75	5	9	18

Up and Down Trend Moves: Most Favorable Excursion (MFE) One day trading approach is to just trade the trend moves. For example, go long when a long trend is established (i.e. price closes 6 ticks (1.50 points) about the 55 EMA) and go short when a short trend is established (i.e. price closes 6 ticks (1.50 points) below the 55 EMA). This has a couple ramifications. First, as shown above, we have 956 price swings in our sample size but only 93 trend moves (shown below). A trend move is made up of several price swings both with and against the trend. The table below shows the "Most Favorable Excursion (MFE) points and length in the trend move. The MFE means what was the maximum available profit points for a trend move. The 20 second bars means how many price bars from entry the MFE occurs.

Trend Moves	Sample Size	Most Favorable Excursion Points (% Levels)			20 second bars (% Levels)		
		20%	**50%**	**80%**	**20%**	**50%**	**80%**
All Trend Moves	93	1.0	2.0	5.0	3	27	97
All trends – Core Hours	61	.5	1.75	5.0	2	16	46
All trends – Non Core Hours –	32	.75	1.25	2.25	17	68	225

As shown above, the trend moves actually provide less profit overall then the swings themselves. This can be attributed to the fact the in choppy markets, the trend changes are frequent. The positive attribute trading trend moves is the length (number of 20 second bars) to hit the MFE helps establish a time target for the trade.

Up and Down Trend Moves: Most Adverse Excursion (MAE) The Most Adverse Excursion (MAE) is the price move against a position before the MFE (maximum profit) is achieved. The

table below shows the trend move MAE pullback points (hence the negative numbers) and the number of bars after trade enter the MAE occurs.

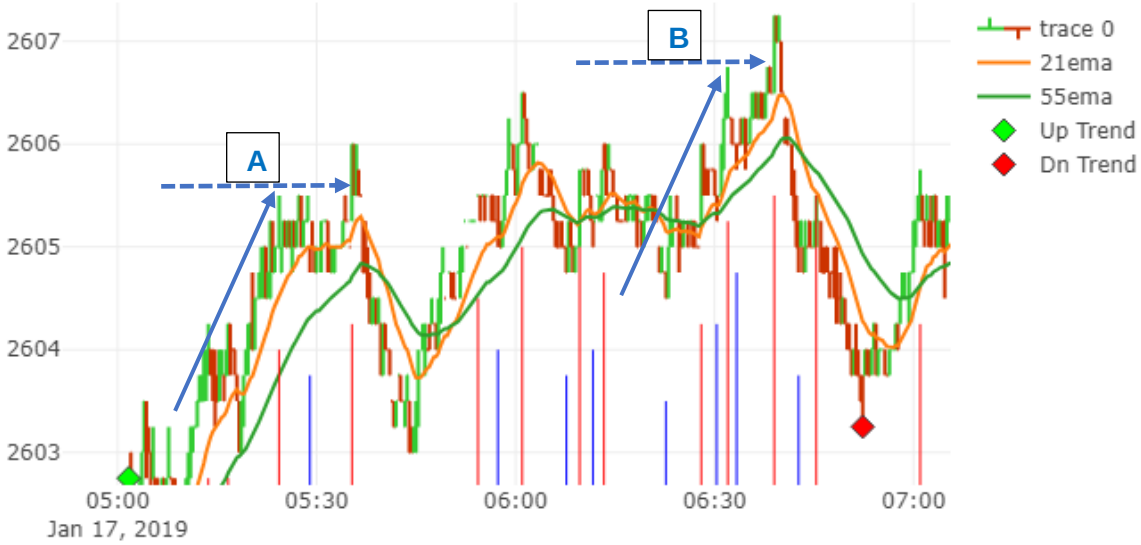
Trend Moves	Sample Size	Most Adverse Excursion Points (% Levels)			20 second bars (% Levels)		
		20%	**50%**	**80%**	**20%**	**50%**	**80%**
All Trend Moves	93	(.50)	(1.0)	(2.0)	3	27	97
All trends – Core Hours	61	(.25)	(1.0)	(1.75)	1	2	15
All trends – Non Core Hours –	32	(.50)	(1.25)	(2.00)	2	15	37

The MAE information about highlights two key items. First, over 80% of trades pull back at least .50 points for all trades and at least .25 points for trend trades. This means that entering a trade on a limit order should be beneficial. Second, The low number for the Core hours MAE (20% is 1 bar, 50% is 2 bars) means that the MAE maximum pull bar is hit 1 – 2 bars after trade entry for over 50% of the trades. Since the 80% level is 15 bars, additional analysis is needed to refine this number. The main take away from the MAE bars is that successful trades during the core hours typically only have a few bars of pullback (1 – 2 bars) and only pullback around .25 to 1.0 points. This is key to refining a day trading strategy. We will now apply machine learning to determine what other attributes are key to predicting price moves.

Machine Learning Application

As shown by the analysis above, intraday swings of 3 – 4 minutes are common throughout the trading day. The challenge is finding a consistent way to trade these swings. The typical swing is 2 – 3 points with typically .75 – 1.25 points of potential profit. Some machine learning algorithms try to predict if the next price bar will close higher or lower. Unfortunately, these higher volume price swings do not readily lend themselves to machine learning techniques. For example, the typical range of each 20 second price bar is 2 – 3 ticks (.50 to .75 points) and often price only exceeded the previous price bar by .25 points. This is too small of an amount to be useful for predicting where the next price bar will close up or down.

A good alternative is to use machine learning to predict whether the price swing will exceed the previous price swing high (for an uptrend) or go below the previous price swing low (for a down trend). By looking at the entire price swing (instead of just a single price bar), predicting whether the previous swing high or low provides an objective price target to predict being exceeded. This provides a better opportunity to apply machine learning.



As shown in the figure above, there are 2 up swings (A & B). The solid area shows the move up. The dotted line horizontal arrow points to the top of the swing for A and B moves. We will use the random forest algorithm to predict if the next up moves after swing A and B will exceed the high of A and B (note – we'll use random forest to predict this for all upswings and down swings).

Random Forest – sklearn: First, we'll need to import the sklearn RandomForestClassifier as shown below.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score,
confusion_matrix, roc_curve, auc
```

Second, our data has text categorizations for describing the swing data (LWTR – long swing with trend, SWTr – short swing with trend, etc). This data will need to all be replaced with numerical categorizations.

```
test_df2 = test_df2.replace('noncore',0)
test_df2 = test_df2.replace('core',1)
test_df2 = test_df2.replace('LWTr',1)
```

test_df2 = test_df2.replace('LATr',2)
test_df2 = test_df2.replace('SWTr',3)
test_df2 = test_df2.replace('SATr',4)

Our data is reduced to these columns:
21 EMA
55 EMA
Breakout (our target variable)
Retrace = the percentage retracement of the previous swing
Swing (-1 = downswing, 1 = upswing)
Swing_len – number of 20 second bars in swing
Swing_type (4 types of swings – Long and Short, with and against the trend)
Trend (1 = up trend, -1 = down trend)
Hour = trading hour (24 hour scale)
Hour_type (9 am – 3 pm = 1 (“core” hours) and 4 pm – 8am = 0 (“noncore” hours)

	21EMA	55E	Brkout	retrace	swing	swing_len	swing_type	trend	hour	hour_type
0	2614.33	2614.14	1.0	1.00	-1	2	4	1.0	3	0
1	2614.33	2614.15	1.0	2.00	1	2	1	1.0	3	0
2	2614.28	2614.14	1.0	1.50	-1	4	4	1.0	3	0
3	2613.97	2614.03	0.0	2.33	1	7	1	1.0	3	0
4	2614.03	2614.04	0.0	0.86	-1	4	4	1.0	3	0

The target and feature variables are identified:
target = "Brkout"
feature_cols = test_df2.columns[test_df2.columns != target]

 # Create X and Y variables
X = test_df2[feature_cols]
Y = test_df[target]

The training and test data is created:
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
print (X_train.shape, y_train.shape)
print (X_test.shape, y_test.shape)

(764, 9) (764,)
(192, 9) (192,)

The RandomForestClassifier is run:
clf = RandomForestClassifier(n_jobs=2, random_state=0)
clf.fit(X_train, y_train)

The predictions are created and a cross tab report is run:

preds= clf.predict(X_test)
pd.crosstab(y_test, preds, rownames=['Actual Result'], colnames=['Predicted Result'])

Predicted Result	0.0	1.0
Actual Result		
0.0	53	36
1.0	32	71

The accuracy is calculated
Accuracy: 0.65
Precision: 0.66
Recall: 0.69

While 65% accuracy is a solid starting point, the real question is what variables are important to the analysis.

list(zip(X_train, clf.feature_importances_))

[('21EMA', 0.16784893965323588), <<<<<<<<<<<<
('55E', 0.1474478287573294),
('retrace', 0.29807399850023714), <<<<<<<<<<<<
('swing', 0.015242575697336433),
('swing_len', 0.19725555099071168), <<<<<<<<<<<<
('swing_type', 0.03437945102718543),
('trend', 0.015427164478980612),
('hour', 0.11245784433977743),
('hour_type', 0.011866646555205956)]

As shown by the feature importance list about, the 3 “<<<<” items are the most important features for this random forest. First the “retrace” amount is the most important. This makes sense as if prices retrace too much of the previous swing then price often reverses direction. The swing length is the second most important feature. Finally. the 21 Exponential Moving Average is the third most important feature.

While random forest is a good predictive algorithm, we would need to do a decision tree on these features to understand what values of the features drive the decisions. For example, what retracement amount predicts if a swing high or low will be exceeded. This is the subject for a future study.