Ranking System Optimizer Tutorial

1. Preparing Piotroski For Optimization
2. Setting Up An Optimizer Study
3. Using EXCEL With The Ranking System Optimizer
4. Pruning Nodes From The Ranking System
5. The Final Ranking System

Ranking System Optimizer (Advanced)

1. Advanced Ranking System Optimization
2. Restructuring The Advanced Ranking System
3. EXCEL Spreadsheet Rev 2

Ranking System Conditional Nodes

1. Using the The Portfolio123 Conditional Node
2. Conditional Node Versus Eval (...) Function
3. The Problem With Conditional Nodes

Ranking System Design

October 26, 2013

I am about to embark on a new project that will set the course of my future stock portfolio development efforts. This effort is to create a brand new ranking system using all of the latest information and strategies I have collected over the last several months. I would like to invite readers to follow along so as to understand the thought/development process that has gone into this new project. While I will provide a detailed account of my activities, the actual (fundamental) stock factors will be left to the imagination (sorry - this is my livelihood).

There are (at least) two methods used to design a stock port/simulation. The first method depends on pre-existing ranking system(s). A stock simulation is designed and optimized using the ranking system, some buy/sell rules are put in place, and perhaps some sort of stop-loss and/or hedging. The developer

may have a target stock universe in mind but in the end, the universe is customized to give the "best", or should I say the "most optimized" results. There are several iterative optimization steps that usually occur, including selection of the "best" ranking system, tweaking the ranking system for optimal backtest results, tweaking the buy/sell rules and re-visiting the custom universe, all in the name of designing the "best" or "most optimized" simulation. The universe is quite often customized by setting limits on Market Cap and other parameters. Thus whatever the initial target stock universe eventual gets adjusted until there is a happy compromise between stock universe, ranking system, and backtest performance.

The second method for designing a stock portfolio is to start with a set of criteria, then providing an optimal design to meet that criteria. The ranking system is the heart of the system and is designed from scratch to meet the desired specifications.

There may be some merits to the first design method, it is certainly faster, and the ranking system may be "proven" by other stock models in use. But there will always be some doubt in my mind as to whether the customized universe is a form of cherry picking, or data mining. It is not my place or intent to provide a determination of whether this method of development is satisfactory and I will leave it up to the reader to decide for him/her self. This series of posts will be about the design of a new ranking system from scratch to match the developers specifications and will cover the following topic outline:

- Specifying the target stock universe
- Stock factor discovery
  - Test environment
  - Discovery process
- Stock factor selection process
- Ranking system optimization
  - Using the Portfolio123 RS optimizer
  - Out of sample verification
- Stock port/simulation using the new ranking system*

* This last step is proof that the effort was worthwhile. You can end up with the best looking Ranking System performance chart but if you can't practically use it then it isn't much good.

This will be a very aggressive design for a dynamic ranking system that is intended to adjust for interest rates. I don't know at this point in time whether I will be successful or not. It is a quest, not a "here's how I did it" exercise. The reader is asked for understanding in this matter.

Selecting The Target Stock Universe

October 27, 2013

As I outlined in the last post (overview), I will be performing several steps in this design-from-scratch ranking system project. This post describes the first stage of the design process and is the most straightforward. There are four items I am going to consider for the target universe:

1. Choice of standard Portfolio123 universe
2. Minimum liquidity
3. Minimum stock price
4. Ranking system design process

Choice of Standard Portfolio123 Universe

Since this is a brand new process I am inventing, I have decided to go with a fairly broad universe. I don't like Over-the-counter (OTC) stocks and I also don't want Master Limited Partnerships (MLPs). Therefore, my decision is to use the S&P 1500. All stocks in this universe are listed on a major US exchange with (I believe) no MLPs and all companies are either U.S. companies or international companies with headquarters in the U.S. For me this is preferable to trying to filter out Chinese companies, a rather difficult task.

Liquidity

I normally design very high liquidity models with at least $5M $Volume per day, sometimes a lot higher. For this project I am going to drop my requirement down to $2M $Volume per day. The lower $Volume should give this ranking system better performance than I am accustomed to. My portfolio design will use a buy rule of AvgDailyTot(60)>2,000,000 (minimum $2M $Volume per day on average). But this is an ***average***, and the filter only applies to the days preceding the trade, not the actual day of trade so one has to expect that the $Volume could be lower, either on trade day or while the stock is held. Thus my target universe will be set to $1M, allowing stock liquidity to bounce around a little bit without dropping out of the universe.

Minimum Stock Price

I like to work with a minimum stock price of $2. I find that this price gives very good performance in stock simulations without too much compromise on stock volatility. I know most R2G models buy stocks down to $1 but I don't wish to go down that low. That is my personal preference and I don't care to try to justify it. If a developer is comfortable with buying stocks at $1 then that is OK. You should keep in mind however that the stock price may drop down to penny stock status, and possibly fall off the exchange it is listed on.

So, if I set the minimum stock price at $2 as a portfolio buy rule then I will need to test below this price as the stock price may drop below $2. For example, I don't want to buy at $2 and have to sell the next week because the stock price dropped to $1.95. So I am going to set my minimum stock price for this exercise at $1.50 in the target universe. As a general rule, I require the closing price to be above $1.50 and also a 60 day moving average to be above $1.50. The moving average requirement is to prevent stocks with an unusual price around the lower price from being bought.

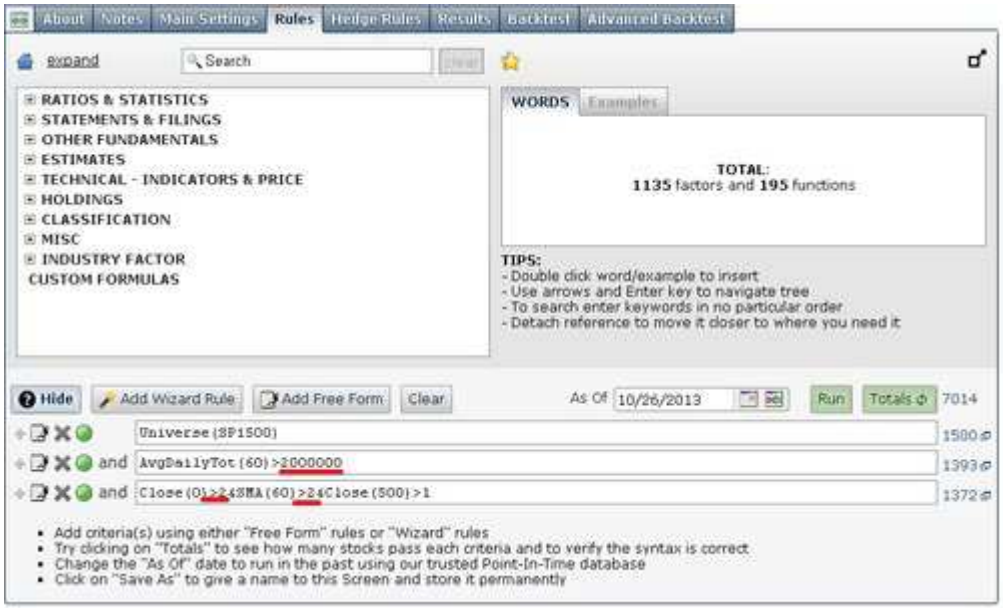
One additional check that I like to include is to make sure the stock has been around for awhile i.e. no IPOs. For this filter I use**Close(500)>1**.

Ranking system design process

As a developer, I need to have a basic understanding of the ranking system design process up front. This is important because I want to know how many stocks will be in each ranking bin which determines the reliability for the entire process.

The first step is to enter all of the rules for the target universe into a screen as shown in the figure below. The most recent date shows a total of 1440 stocks in the target universe. By sampling various dates, I found a minimum of 1233 stocks at one point. For the purpose of this analysis I am going to assume a minimum of 1200 stocks in the target universe, a little more conservative than the minimum number that I had found by sampling.

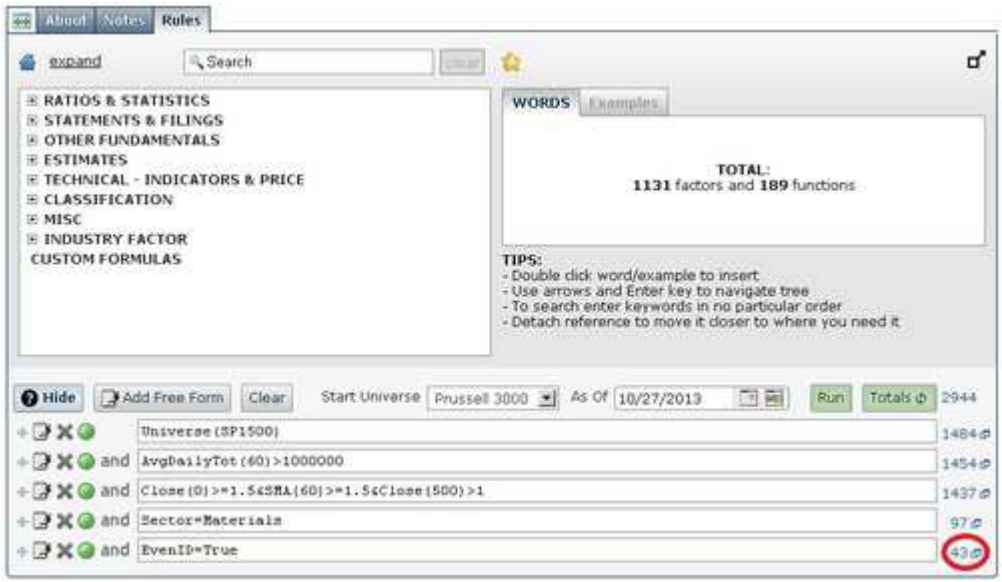
Out of curiosity I decided to repeat this experiment using the Buy rules for a potential port as shown in the figure below. By sampling different dates, I found a minimum of 1100 stocks that fulfilled the rules. As liquidity has increased over the years, my conclusion is that I could expect at least 1100 stocks that will meet portfolio liquidity buy rules at any given point in time in the future.



Now getting back to the target universe analysis...

I will be analyzing the 10 sectors independently, and testing with odd/even stock IDs. Therefore, individual test universes have a minimum of (1200 / 10) /2 = 60 stocks each. I generally test at a rank of 80 and higher, therefore I will be testing 20% of each test universe or 60/5*0.2 = 12 stocks. I don't know if this will be a sufficient number of stocks for test but I am going to proceed with this. If I run into trouble there are some workarounds that I could use - I can't explain just now as I would be getting ahead of myself.

Below is a snapshot of the Test Universe with one sector specified. Already I am below my estimate of 60 stocks in each test universe as I forgot to consider the variation in number of stocks between each sector. Anyways, I will proceed on while keeping a close eye on this issue. Note that I chose the Even stock-IDs as there are more stocks than the odd stock-IDs by ~50 stocks.



Stock Factor Discovery

October 27, 2013

At this stage of my development process, I want to find the best stock factors to use for the target universe (see previous post). For this purpose I have assembled a list of 107 stock factors. When I talk about "stock factors", I really mean "stock formulae" as I don't make a distinction between the two.

The 107 factors are located in a spreadsheet located here:

Stitts-Factor-Discovery

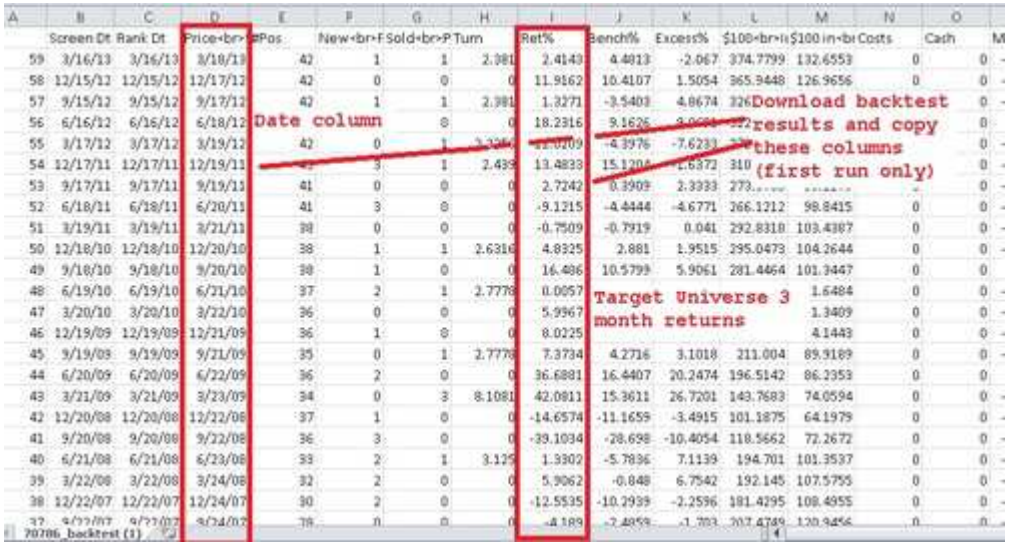
The spreadsheet has five worksheets and the second last sheet lists the factors. They are labeled FACT01, FACT02, FACT03, ..., FACT107. Labels in **red** signify that "lower is better". And labels in black signify that "higher is better".

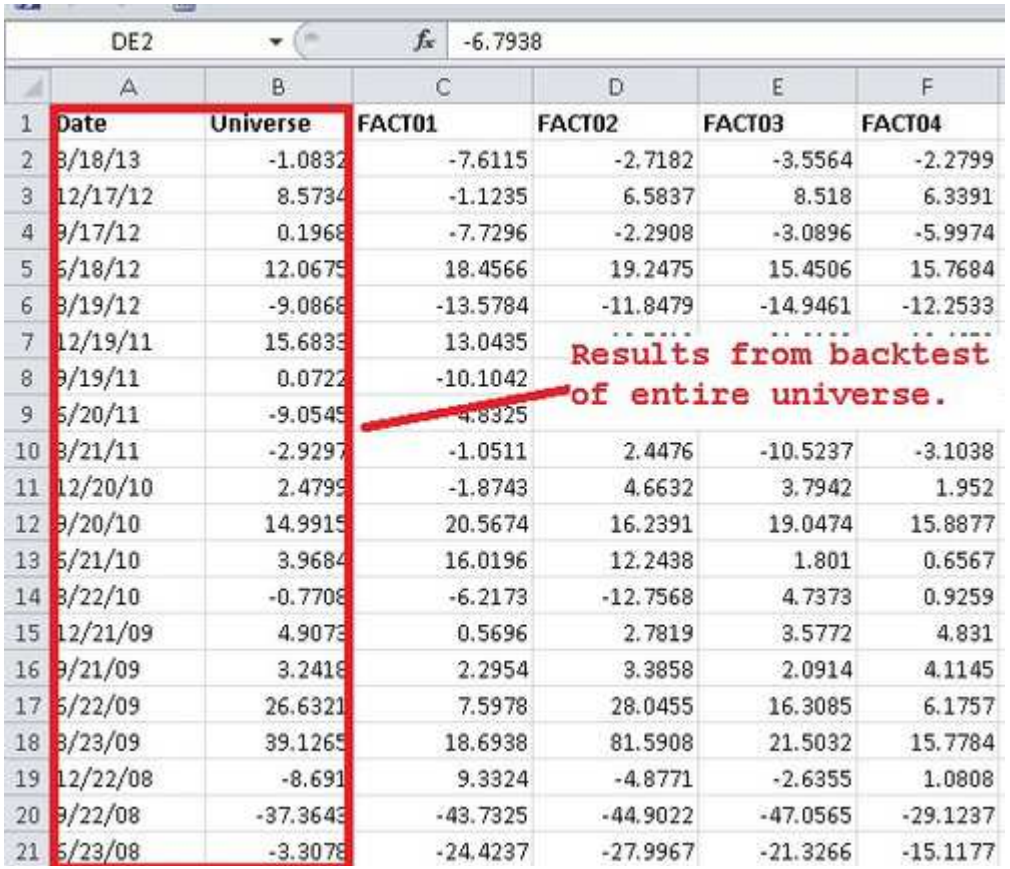
NOTE: I have just given everyone the stock factors that I use for free. I would appreciate it if you could provide me with any additional factors that you find useful. You can use the contact form to send me any additional factors.

While I am at it, I might as well explain how the stock factor discovery process will be performed and how the spreadsheet calculates the best stock factors. I am going to test each factor individually using the screen backtester and target universe. The most recent 10 years are tested with a rebalance frequency of 3 months. This is certainly a brute force approach and is extremely time consuming but the results are worth it.

Sheet 5 (RawData)

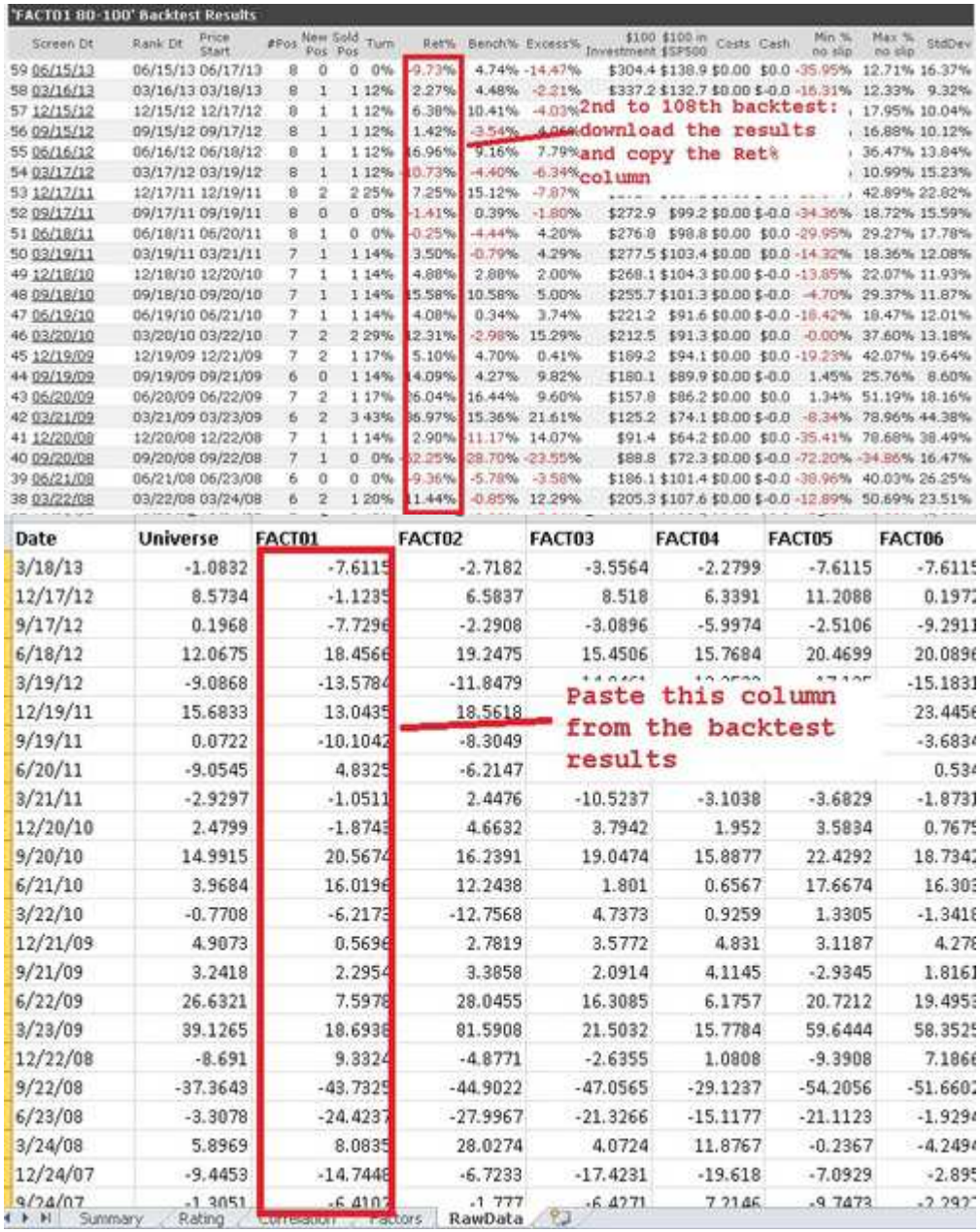
This is where the backtester raw results are deposited, starting with the stock universe. The backtester is executed 108 times, the first time the target universe is tested as is. The backtest results are downloaded and the date plus Ret% fields are copied/pasted into factor discovery spreadsheet as shown below.





	A	B	C	D	E	F
1	Date	Universe	FACT01	FACT02	FACT03	FACT04
2	3/18/13	-1.0832	-7.6115	-2.7182	-3.5564	-2.2799
3	12/17/12	8.5734	-1.1235	6.5837	8.518	6.3391
4	9/17/12	0.1968	-7.7296	-2.2908	-3.0896	-5.9974
5	6/18/12	12.0675	18.4566	19.2475	15.4506	15.7684
6	3/19/12	-9.0868	-13.5784	-11.8479	-14.9461	-12.2533
7	12/19/11	15.6833	13.0435	.. ....	.. ....	.. ....
8	9/19/11	0.0722	-10.1042			
9	6/20/11	-9.0545	4.8325			
10	3/21/11	-2.9297	-1.0511	2.4476	-10.5237	-3.1038
11	12/20/10	2.4799	-1.8743	4.6632	3.7942	1.952
12	9/20/10	14.9915	20.5674	16.2391	19.0474	15.8877
13	6/21/10	3.9684	16.0196	12.2438	1.801	0.6567
14	3/22/10	-0.7708	-6.2173	-12.7568	4.7373	0.9259
15	12/21/09	4.9073	0.5696	2.7819	3.5772	4.831
16	9/21/09	3.2418	2.2954	3.3858	2.0914	4.1145
17	6/22/09	26.6321	7.5978	28.0455	16.3085	6.1757
18	3/23/09	39.1265	18.6938	81.5908	21.5032	15.7784
19	12/22/08	-8.691	9.3324	-4.8771	-2.6355	1.0808
20	9/22/08	-37.3643	-43.7325	-44.9022	-47.0565	-29.1237
21	6/23/08	-3.3078	-24.4237	-27.9967	-21.3266	-15.1177

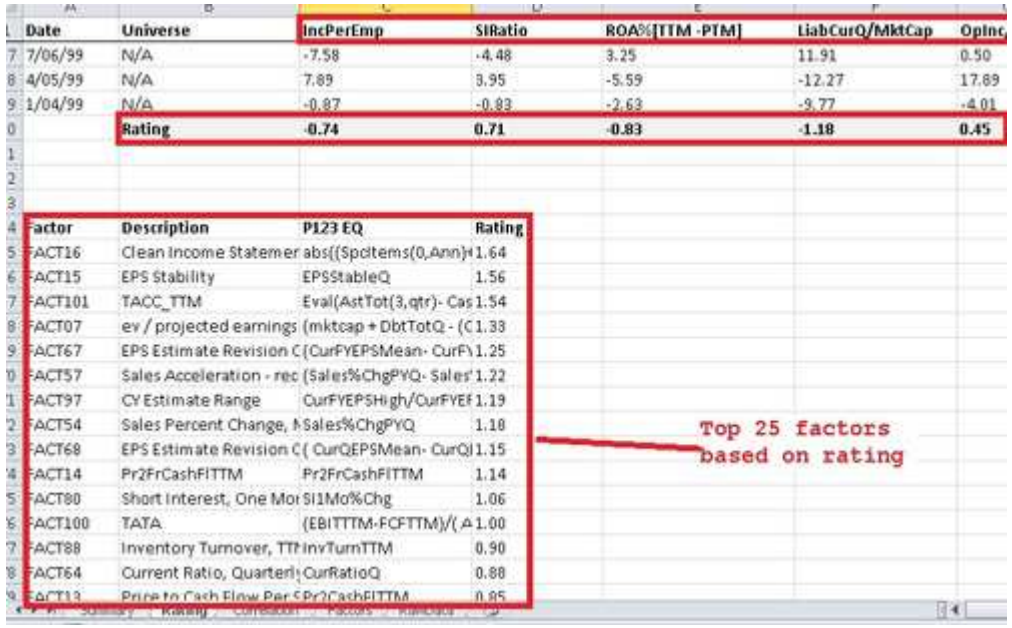
For the 2nd to 108th run, only the Ret% field is cut/pasted into the factor discovery spreadsheet. For these runs, each stock factor is tested individually by using a ranking system with one factor only. The screen has one Rule: **Rank>80.**



This is a lot of work, isn't it? If someone out there has the programming skills to automate this effort then I would love to hear from you.

Sheet 2 (Rating)

On this worksheet, the delta between each factor and the target universe as a whole is calculated for each three month period. Then a rating is calculated for each stock factor. The rating is essentially a crude version of the **Sharpe Ratio, except it is not annualized and uses the target universe in place of the no risk return. This sheet also determines and lists the top 25 stock factors in order of rating.**



Sheet 3 (Correlation)

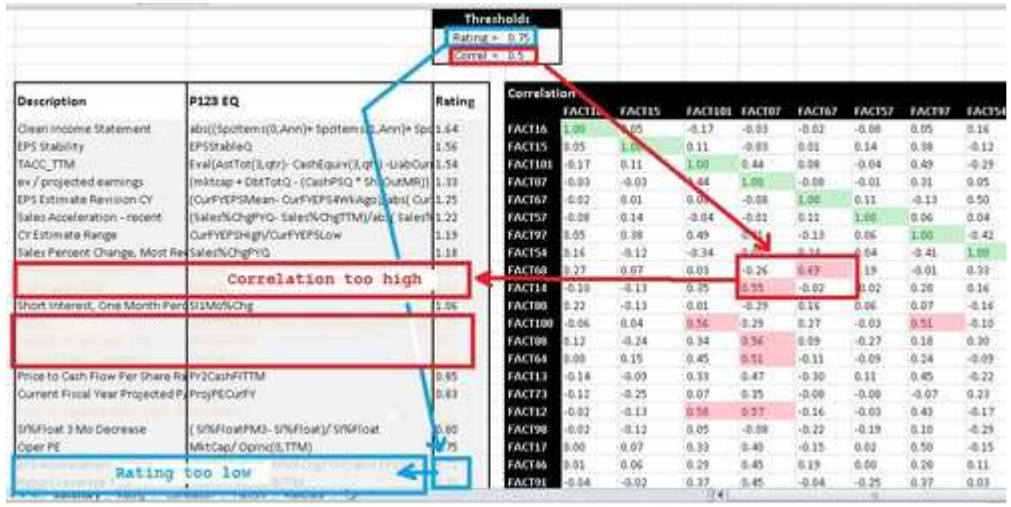
This worksheet calculates the correlation between each of the 107 stock factors based on the 3 month dela returns calculated on Sheet 2.

	FACT01	FACT02	FACT03	FACT04	FACT05	FACT06	FACT07	FACT08	FACT09	FACT10	FACT11
FACT01	1.00	0.07	0.46	0.35	0.34	0.36	-0.01	0.42	0.42	-0.31	0.41
FACT02	0.07	1.00	-0.05	-0.05	0.43	0.23	0.52	0.05	0.43	0.26	0.29
FACT03	0.46	-0.05	1.00	0.34	0.28	0.22	0.12	0.37	0.29	-0.18	0.28
FACT04	0.35	-0.05	0.34	1.00	-0.15	-0.17	-0.19	0.29	0.18	-0.51	-0.10
FACT05	0.34	0.43	0.28	-0.15	1.00	0.58	0.60	0.40	0.59	0.27	0.56
FACT06	0.36	0.23	0.22	-0.17	0.58	1.00	0.33	0.13	0.37	0.20	0.94
FACT07	-0.01	0.52	0.12	-0.19	0.60	0.33	1.00	0.50	0.68	0.39	0.34
FACT08	0.42	0.05	0.37	0.29	0.40	0.13	0.50	1.00	0.74	-0.27	0.18
FACT09	0.42	0.43	0.29	0.18	0.59	0.37	0.68	0.74	1.00	-0.06	0.41
FACT10	-0.31	0.26	-0.18	-0.51	0.27	0.20	0.39	-0.27	-0.06	1.00	0.18
FACT11	0.41	0.29	0.28	-0.10	0.56	0.94	0.34	0.18	0.41	0.18	1.00
FACT12	-0.29	0.51	-0.25	-0.42	0.45	0.25	0.57	-0.09	0.16	0.78	0.26
FACT13	-0.32	0.32	-0.03	-0.42	0.40	0.55	0.47	-0.17	0.05	0.70	0.53
FACT14	-0.06	0.59	0.00	-0.25	0.49	0.40	0.55	-0.06	0.20	0.58	0.41
FACT15	-0.25	-0.07	-0.06	0.41	-0.30	-0.25	-0.03	-0.08	-0.01	-0.28	-0.28
FACT16	-0.06	-0.07	-0.06	-0.07	-0.19	-0.12	-0.03	-0.06	0.00	0.08	-0.04
FACT17	-0.35	0.40	0.06	-0.23	0.39	0.29	0.40	-0.25	0.15	0.65	0.29
FACT18	0.62	0.11	0.31	0.11	0.45	0.62	0.16	0.34	0.44	-0.18	0.65
FACT19	0.60	-0.04	0.39	0.41	0.26	0.33	-0.08	0.33	0.28	-0.47	0.33
FACT20	0.38	0.51	0.21	-0.08	0.72	0.79	0.47	0.29	0.48	0.17	0.79
FACT21	0.31	0.26	0.09	0.01	0.36	0.63	0.29	0.18	0.32	0.10	0.65
FACT22	0.20	0.37	0.11	-0.09	0.57	0.61	0.31	0.10	0.23	0.36	0.65
FACT23	-0.38	-0.51	-0.32	-0.15	-0.39	-0.26	-0.36	-0.13	-0.51	-0.13	-0.37

mmary | Rating | Correlation | Factors | RawData

Sheet 1 (Summary)

The first worksheet lists the top stock factors with lower rated factors that are highly correlated to higher rated factors being "grey'ed out". Factors below a certain rating threshold are also grey'ed out. These thresholds are programmable.



That's enough info for one post. Next time I will attempt to describe the test environment, or how the tests are set up to make it easier (on me) :)

Test Environment

October 28, 2013

I've changed my plan a little bit from what I described in yesterday's post. I realized that no I'm not Superman and I can't afford to spend several months on this project. In other words, a few simplifications were in order.

First of all, I decided not to test every sector for every factor. Instead I will test the entire target universe for all 107 factors, then afterwards determine if individual sectors are covered by the stock factors. If not, then I will further test those sectors.

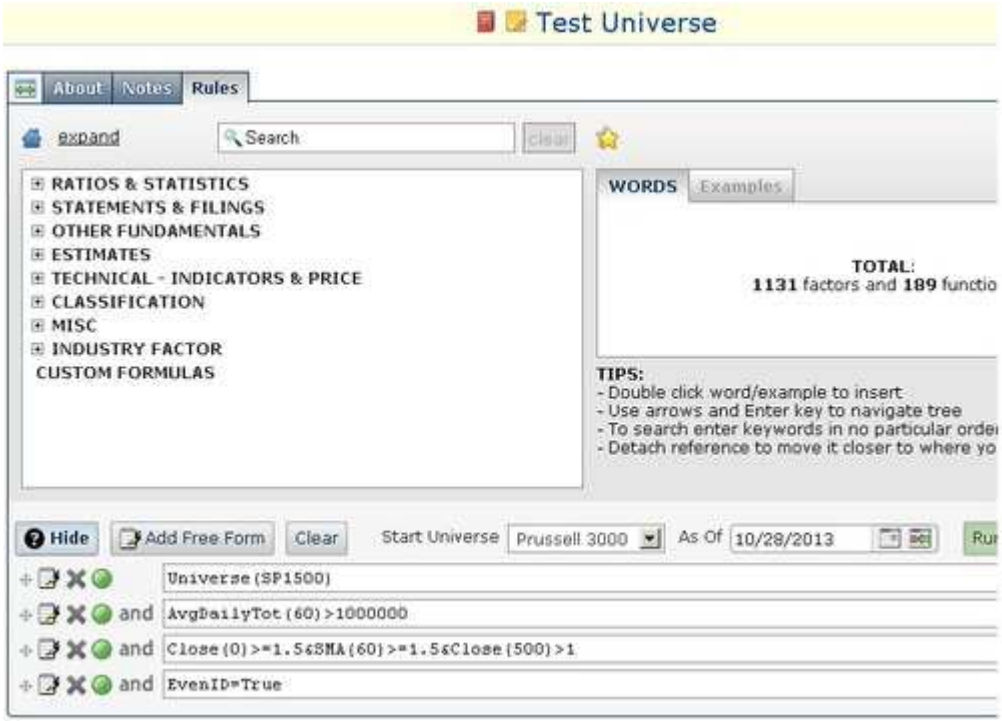
In addition, I won't be testing for rising/falling interest rates at this stage but instead I will do so during RS optimization. A separate project will be initiated to discover what factors work best in various economic conditions.

Stock Ranking Systems

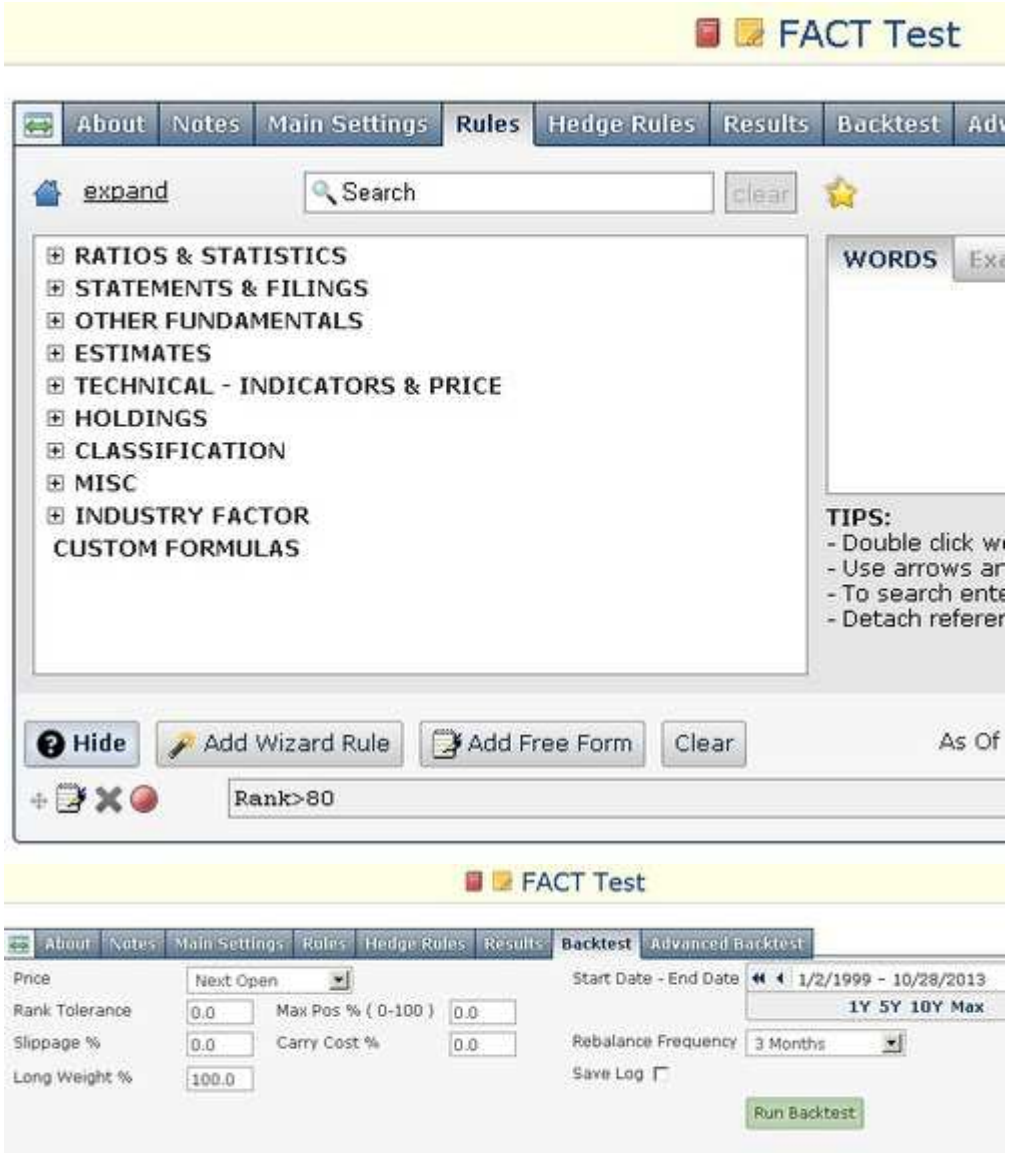
So now I would like to show my "test environment". Believe it or not, I have 107 individual ranking systems, each with one stock factor corresponding to what is in the spreadsheet (see last post). Skeptics can view the screenshot below.

Test Universe

The test universe is the same as I described in the last post except I have removed the sector rule.



The first step will be to backtest the complete target universe, even stock-IDs only. Note that the same screen is used to test the target universe as the individual stock factors. The Rule **Rank>80** is disabled for the universe backtest. The backtest setup and results are shown below. The stock universe backtest Alpha is 7.5% so individual factors have to perform much better to ultimately be chosen for the ranking system.


FACT Test
About
Notes
Main Settings
Rules
Hedge Rules
Results
Backtest
expand
Search
clear
RATIOS & STATISTICS
STATEMENTS & FILINGS
OTHER FUNDAMENTALS
ESTIMATES
TECHNICAL - INDICATORS & PRICE
HOLDINGS
CLASSIFICATION
MISC
INDUSTRY FACTOR
CUSTOM FORMULAS
WORDS
TIPS:
Hide
Add Wizard Rule
Add Free Form
Clear
As Of
Rank>80
FACT Test
About
Notes
Main Settings
Rules
Hedge Rules
Results
Backtest
Advanced Backtest
Price
Next Open
Rank Tolerance
0.0
Max Pos % (0-100)
0.0
Slippage %
0.0
Carry Cost %
0.0
Long Weight %
100.0
Start Date - End Date
1/2/1999 - 10/28/2013
1Y 5Y 10Y Max
Rebalance Frequency
3 Months
Save Log
Run Backtest

Stock Factor Tests

Once the universe results have been downloaded and cut/pasted into the factor-discovery spreadsheet (see previous post) then I can proceed with backtesting each individual factor. The ranking system is selected from the **Main Settings** Tab. The rule Rank>80 is enabled on the **Rules** Tab. The screen setup is shown below. The **Backtest** Tab setup is the same as for the universe backtest.

And now I'm away to the races! I should be finished the first round of testing (all 107 factors) in 4-8 hours.

Ranking System Optimizer Tutorial

October 30, 2013

Alchemist

A number of Portfolio123 members have been asking for a tutorial on the stock ranking system optimizer since it became available almost one year ago. P123 staff have been silent about its use ever since. Recently I became curious about it and decided to explore the feature to see how (or if) I could make use of it.

What I discovered was that the optimizer has some limitations that make it difficult to use without proper documentation. In addition, usage also requires use of EXCEL or other spreadsheet program in a fashion that is not very intuitive. But once the limitations and interface are understood then one can make effective use of it.

Just remember that it is only a tool and is only as good as the person using it. Optimization is not a science as some would like to believe. There is no correct end-point, or time to stop. Two people attempting to optimize the same ranking system will invariable come up with two different results.

In this tutorial I will attempt to optimize a popular stock ranking system, *All Stars: Piotroski*. It is the ideal ranking system to start with as there are only eight ranking factors and most investors are familiar with Piotroski.

The tutorial will be broken up into several parts as there are a lot of pictures, and google is already complaining about the load time of some of my articles :)

Preparing Piotroski For Optimization

October 30, 2013

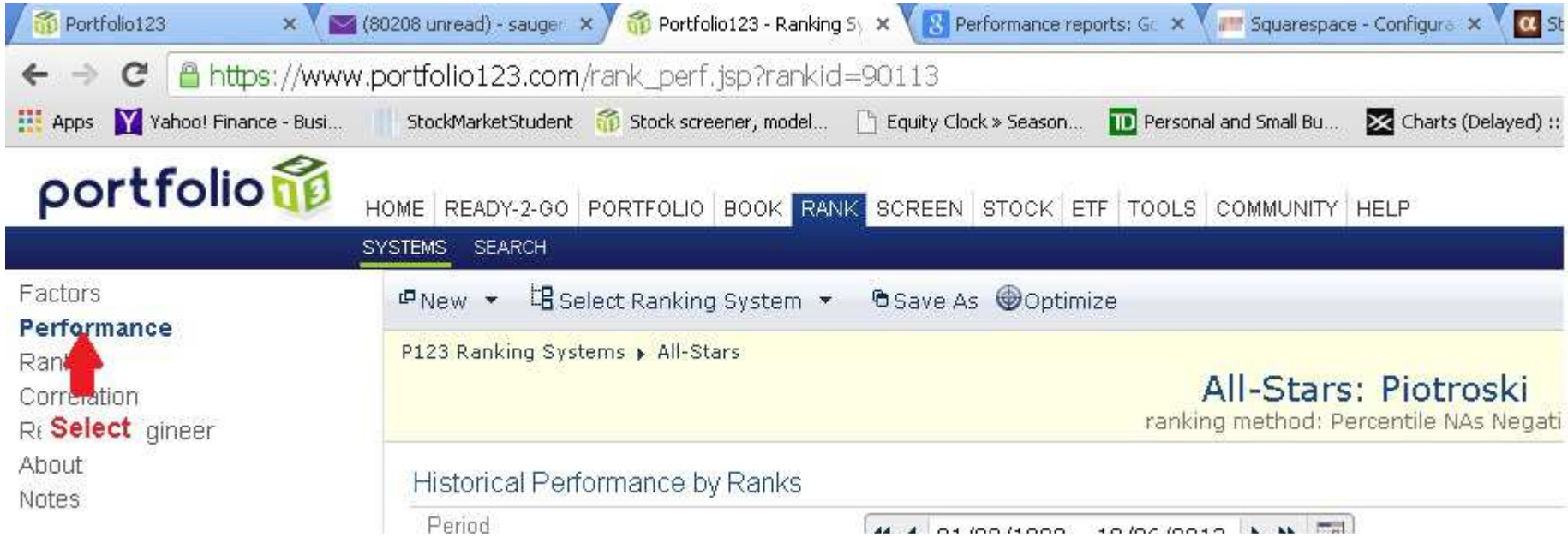
I am going to attempt to optimize the Portfolio123 ranking system *All Stars: Piotroski*. The Portfolio123 implementation of Piotroski's screening factors is different than the original but provides a reasonable implementation.

Limitation of Stock Factors/Functions

The ranking system consists of eight factors. This is important because the P123 Ranking System Optimizer can handle fifteen factors and not much more. If you try to optimize a ranking system with 20 factors, for example, the optimizer will give errors when you try to generate permutations. The ranking system is shown below.

Performance Baseline

Before getting started with this exercise it is a good idea to measure the performance of the ranking system as is. To do this, you need to select the All Stars: Piotroski Ranking System (as shown above) and then select the performance link on the left side of the screen as shown below.



Once you arrive at the performance screen then the appropriate fields should be set up for the performance chart calculation. In this case, the defaults are going to be used. Click on the Run button to calculate the performance graph.

The ranking system performance is shown below. With 20 ranking buckets, the highest ranking bucket **Rank > 95**, is the right-most bar on the chart and is 20% per annum calculated over the full data history starting in 1999.

All-Stars: Piotroski

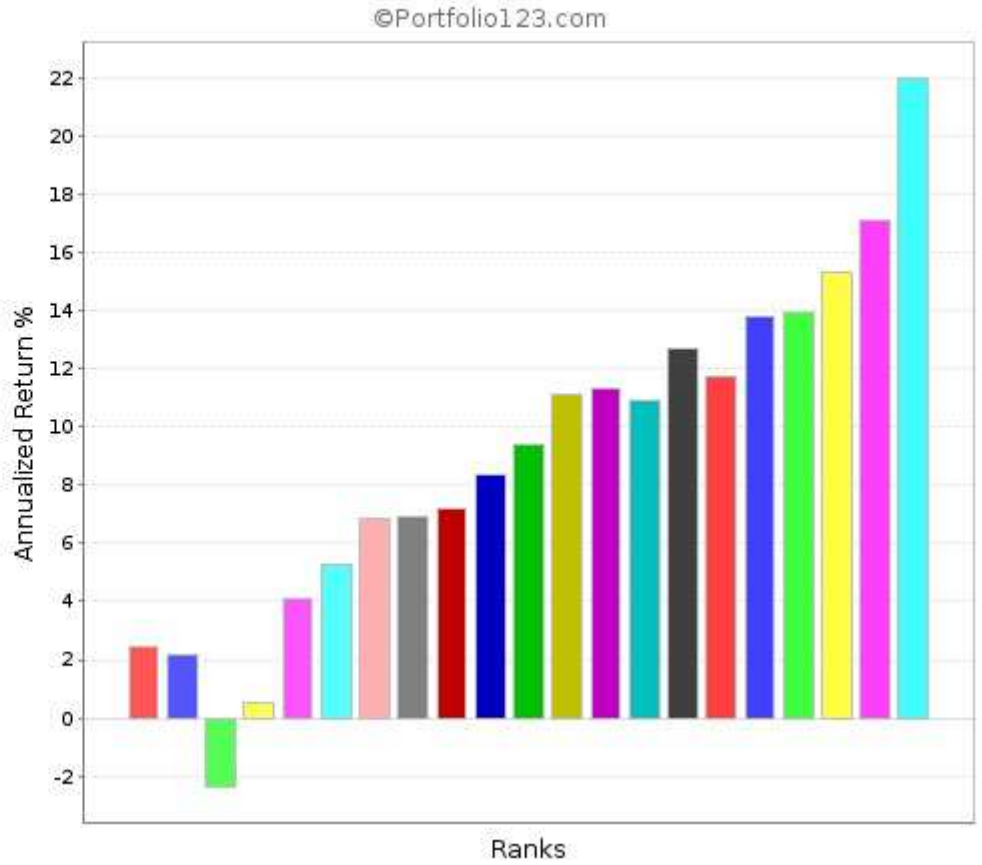
ranking method: Percentile NAs Negative

Historical Performance by Ranks

Period: 01/02/99 - 10/26/13
Rebalance Frequency: 4 Weeks
Ranking Method: Percentile NAs Negative
Slippage (% of trade amount): 0.0
Transaction Type: Long
Universe: All Fundamentals
Benchmark: S&P 500

Filters:
Number of Buckets: 20
Minimum Price: 3.0
Sector: -- ALL --

Change Settings | Download



Now that we know the baseline performance is, let's massage the ranking system into a form that is compatible with the optimizer.

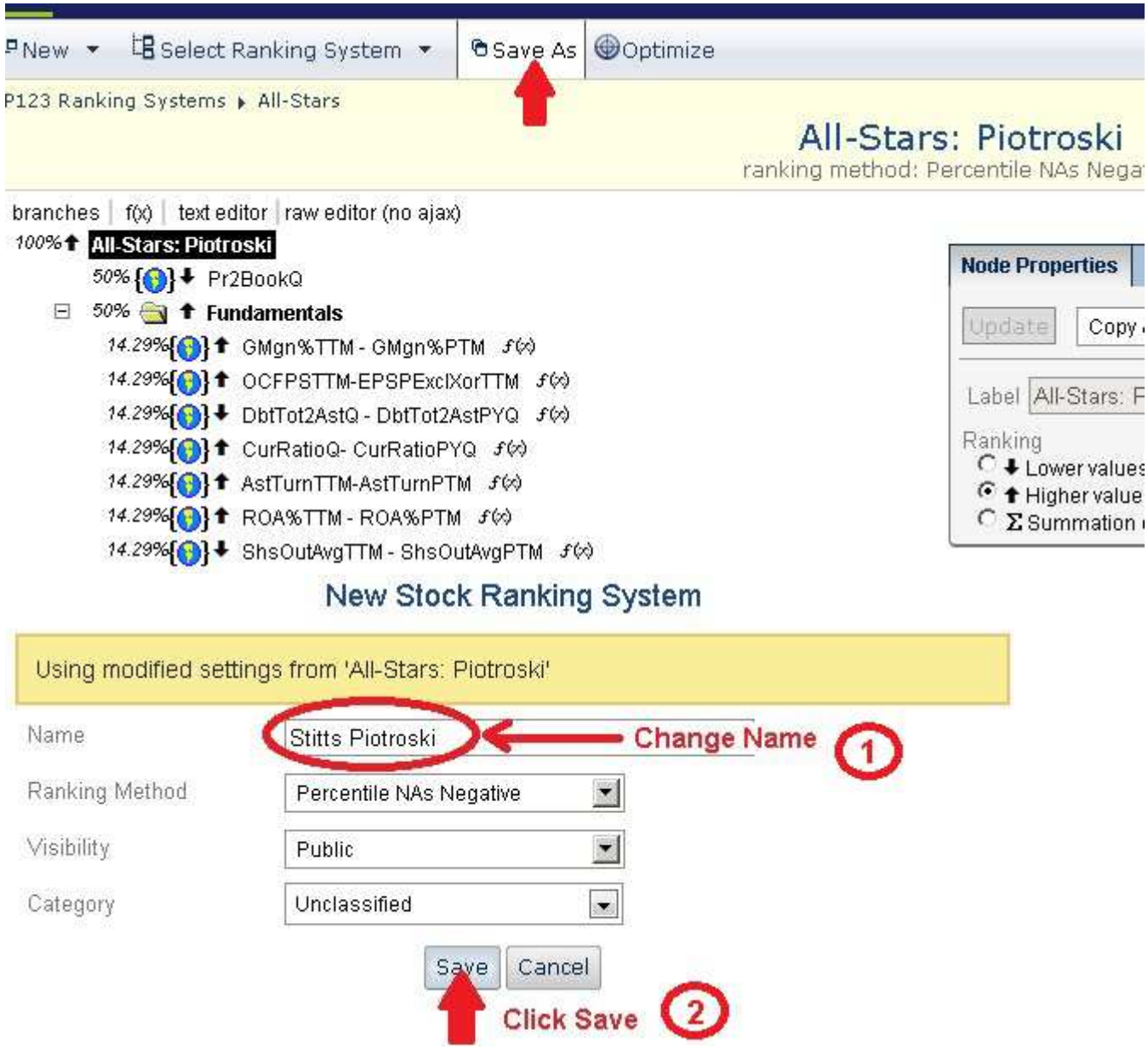
Node Limitations

The next limitation of note is that the ranking system optimizer cannot able to handle composite nodes nodes such as the fundamentals node along with the 7 factors within the node. The optimizer also can't handle conditional nodes. **All complex nodes, apart from stock factor and formula, have to be eliminated.** In this case, the process is not too painful and can be done using either the text editor or the procedure described below.

Copying the Ranking System

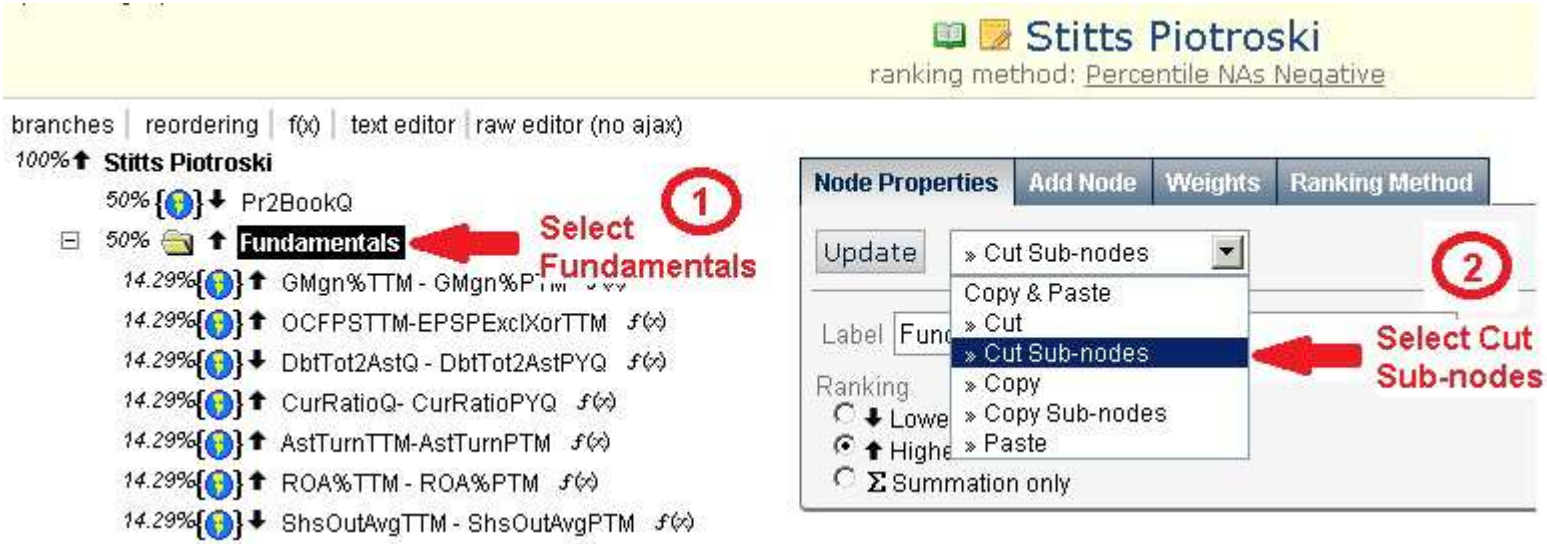
All Stars: Piotroski is a P123 Ranking System and can't be edited directly. Therefore an editable copy must be created. This is done by clicking on **Save As**.

Change the name of the ranking system then click on **Save**.



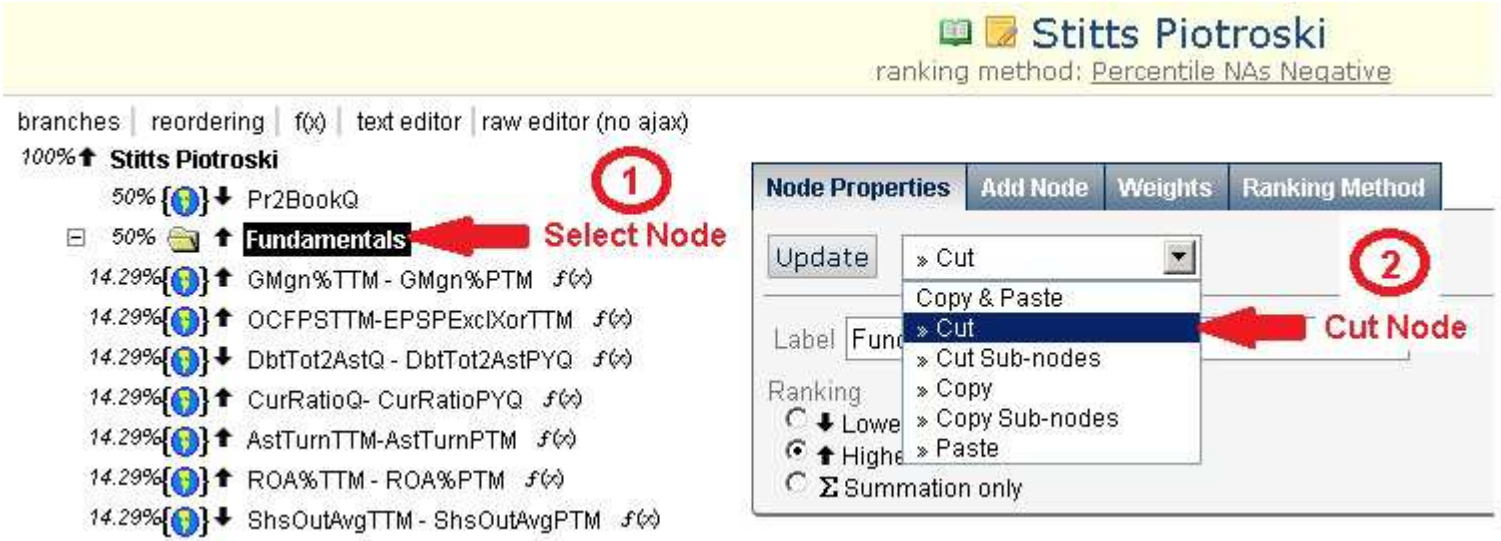
Eliminating the Composite Node

After the ranking system has been copied and renamed, the composite node can be deleted. As mentioned before, this can be done by using the text editor (if you are a programmer). Otherwise, select Fundamentals and then cut the sub-nodes as shown below.



Now paste the sub-nodes at the top level as shown below.

Now cut the **Fundamentals** node.



We are going to start optimizing with equal-weighted stock factors/formula. Select the top level node **Stitts Piotroski, then select theWeights** Tab, click on **Distribute Evenly** then **Update**. These actions are shown below.

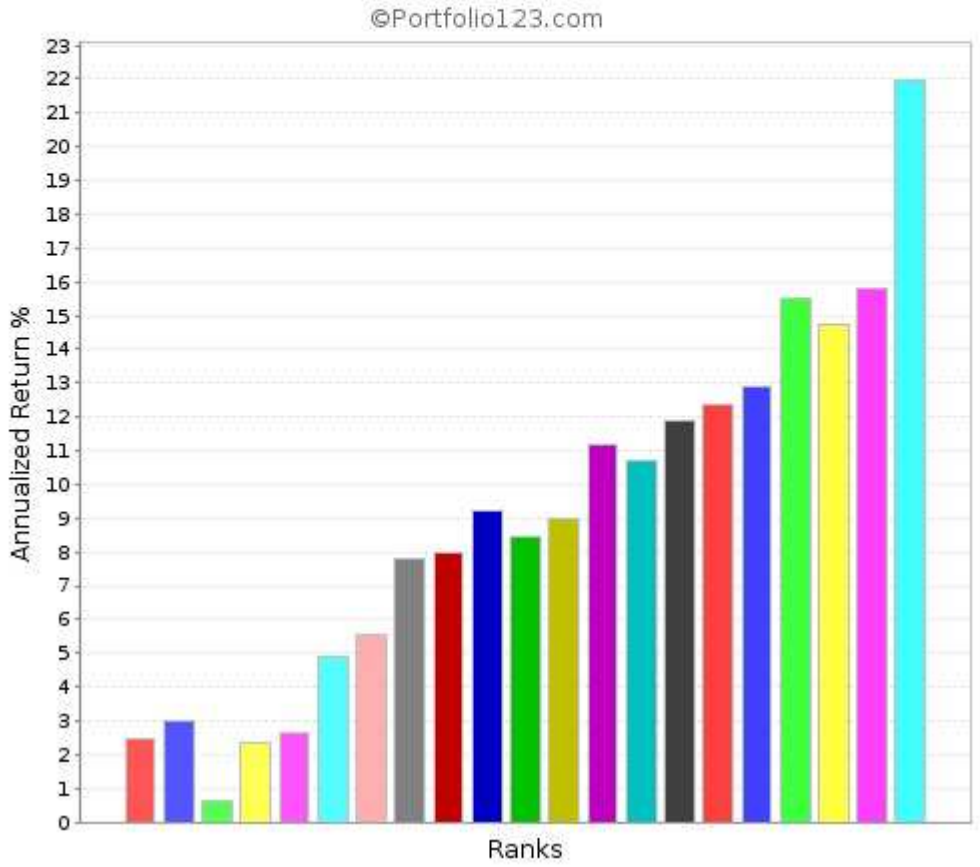
Got all that? Good. Now save the ranking system.

Now lets check the performance of this new ranking system. Use the same procedure as before. Despite changing the node weights, the performance is pretty similar. The top bucket is still 22% per annum. (Click to enlarge).

Period: 01/02/99 - 10/26/13
Rebalance Frequency: 4 Weeks
Ranking Method: Percentile NAs Negative
Slippage (% of trade amount): 0.0
Transaction Type: Long
Universe: All Fundamentals
Benchmark: S&P 500

Filters:
Number of Buckets: 20
Minimum Price: 3.0
Sector: -- ALL --

Change Settings | Download

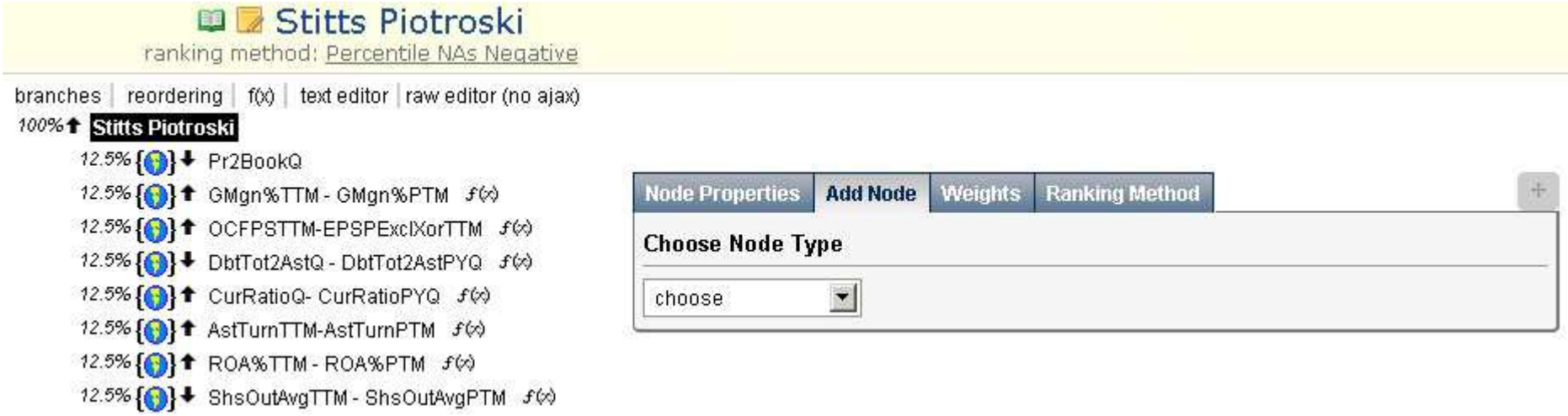


The modification to the All Stars: Piotroski ranking system is now complete and we are ready to begin the optimization process.

Setting Up An Optimizer Study

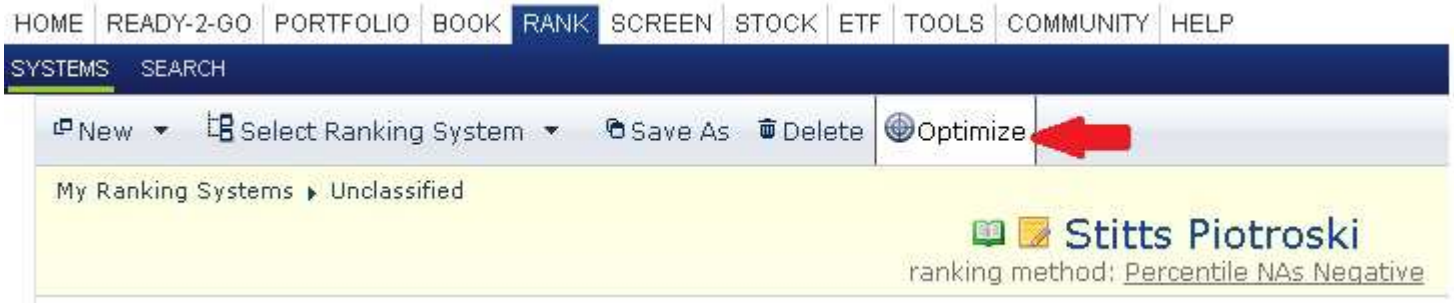
October 31, 2013

In the last post, the ranking system was prepared by eliminating the composite node. The number of nodes (8) was not a concern as it was less than 15, one of the optimizer's limitations. The ranking system now looks like this:



Start the Optimizer

Now click on **Optimize** to start up the ranking system optimizer.

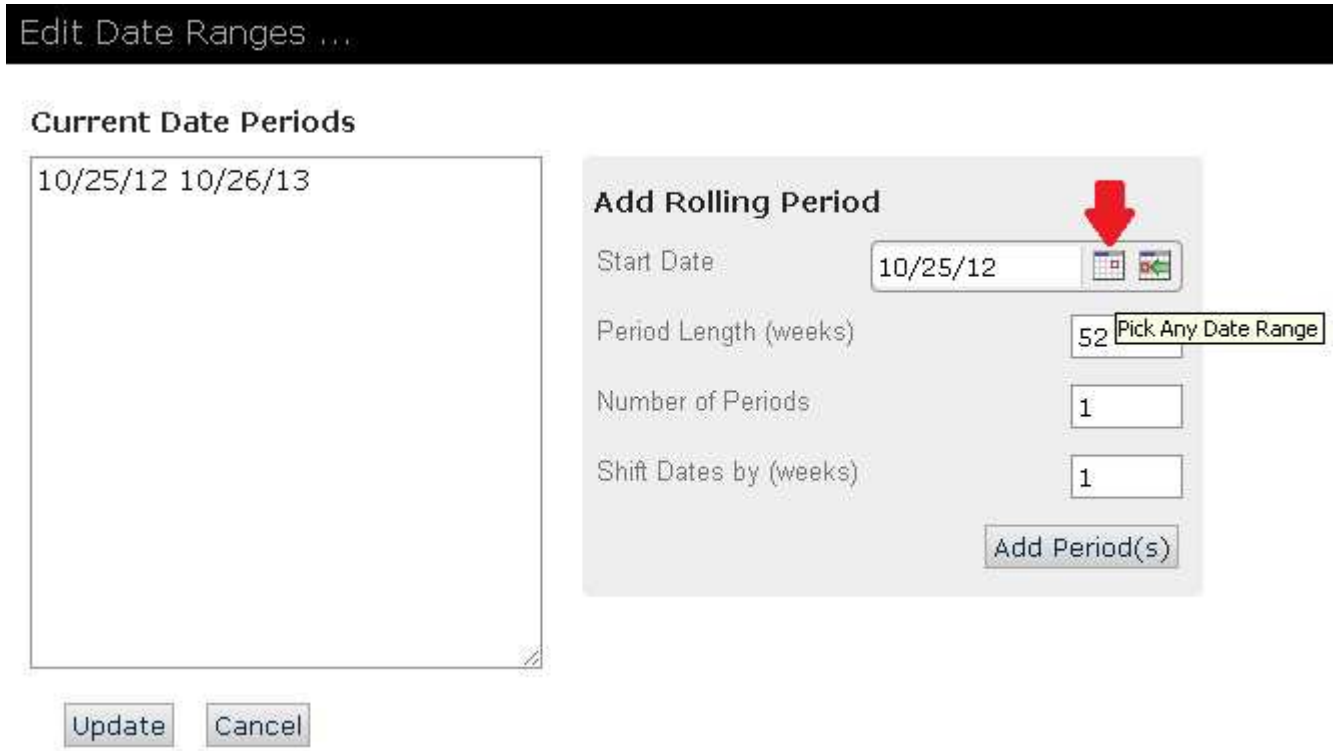


A page called **New Ranking System Study** will be displayed where you can edit several parameters including the rebalance frequency, number of buckets, etc. For this tutorial, the default settings will be used. Click on **Save**.

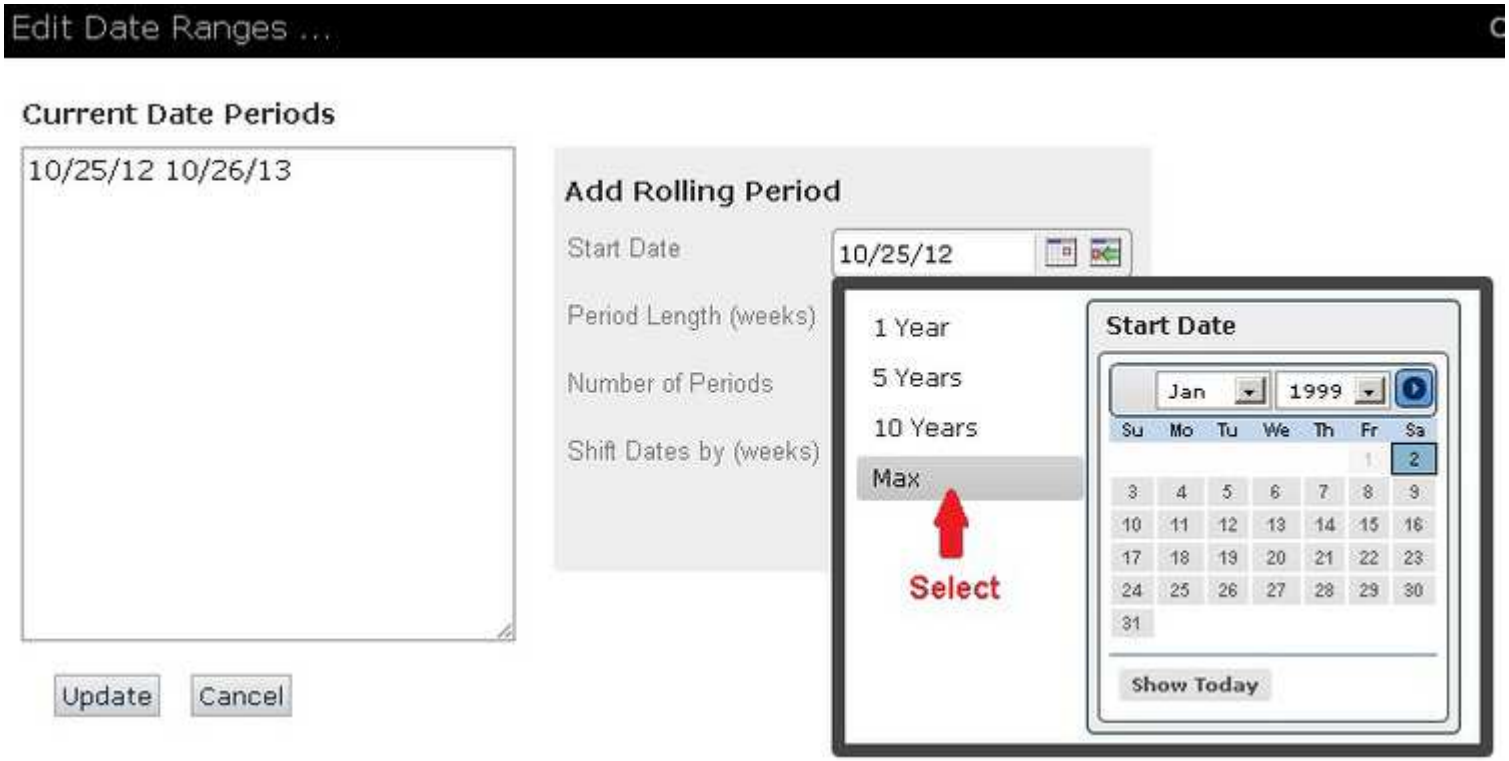
Set the Date Range

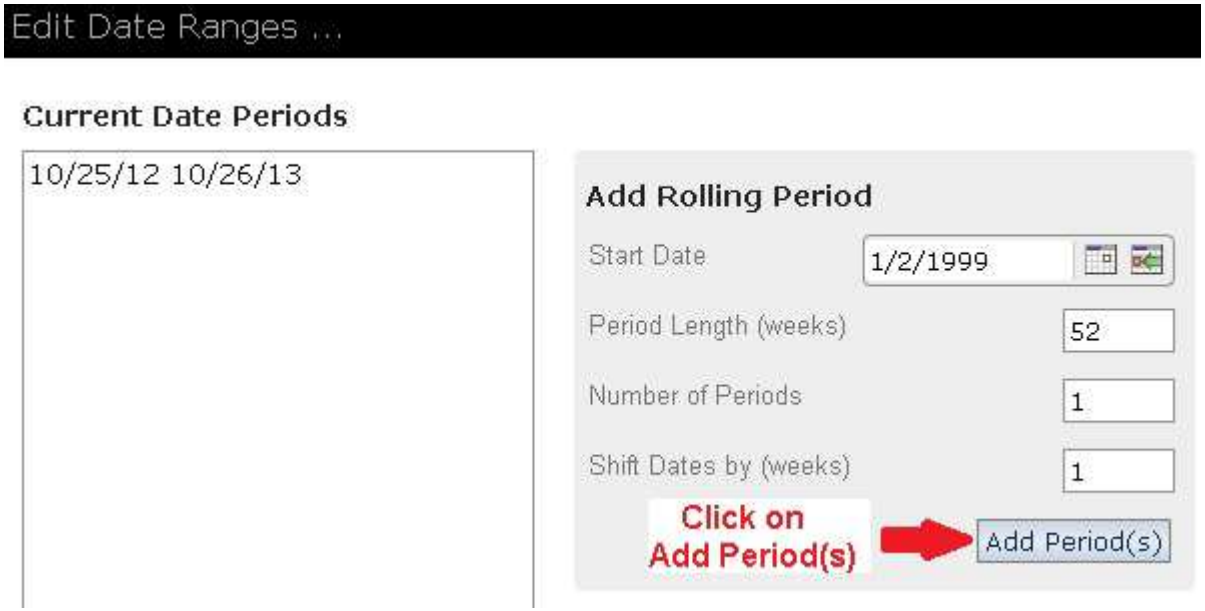
Click on the little plus sign button to edit the date periods.

The default date range is one year. We want to use the full data history. This can be done by editing the start date in the Current Date Periods, or clicking on the little calendar button shown below.

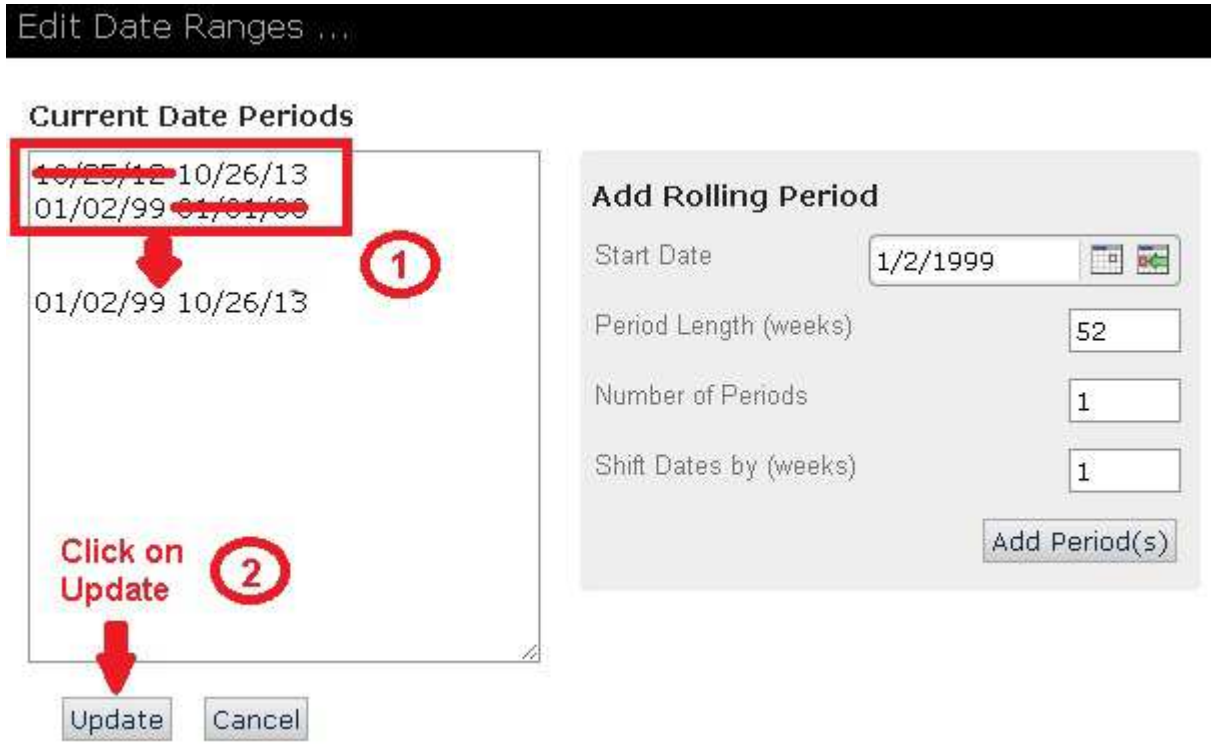


Select Max followed by Add Period(s) as shown below.

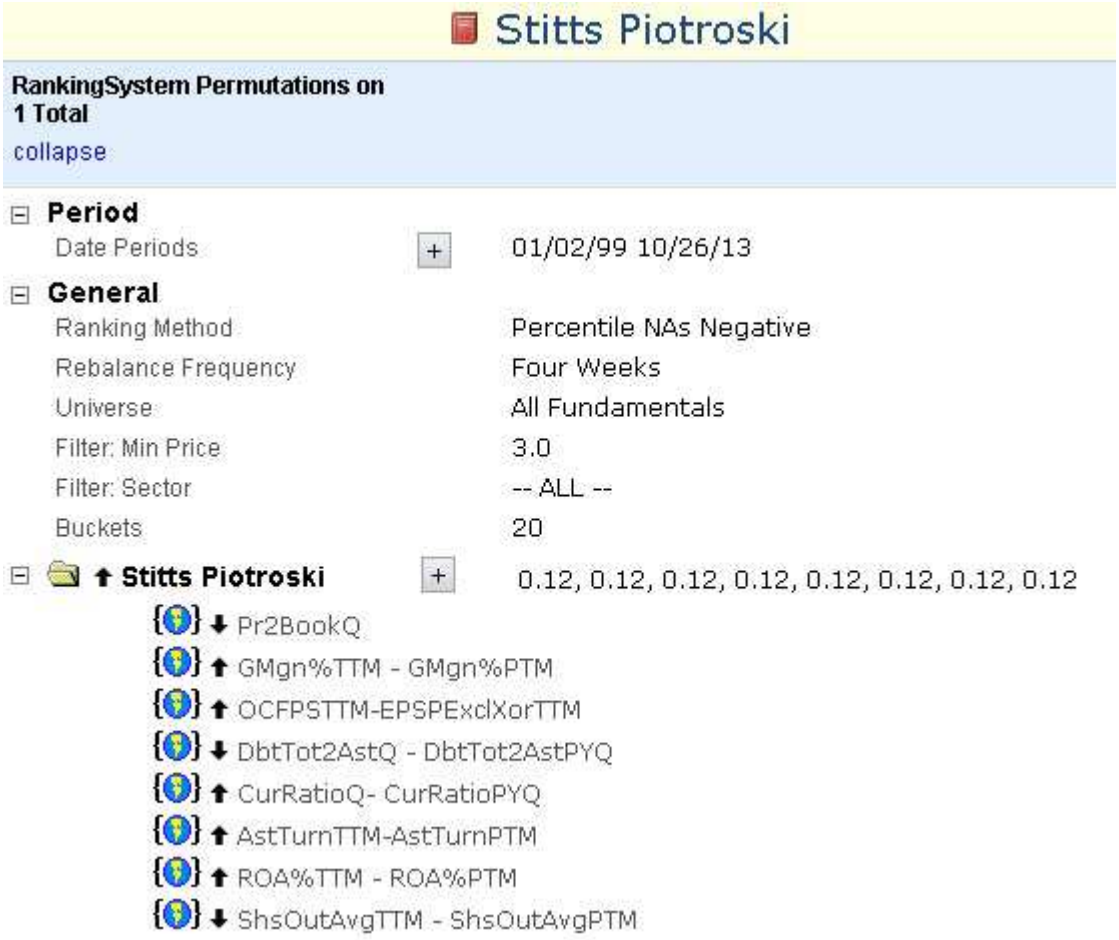




The end date for the maximum date range isn't correct. Delete the extra dates and leave the correct date range. Click on update as shown below.



The optimizer screen is shown below. We are now ready to start generating permutations.



In the next post I will introduce a special spreadsheet I created in order to assist in the generation of permutations.

Using EXCEL With The Ranking System Optimizer

October 31, 2013

In the last post a new stock ranking system optimization study was set up. Now I am going to introduce you to a special spreadsheet that I have prepared that will make it a little easier to perform the optimization steps. The spreadsheet can be found here: RS-Optimizer

Setting Up EXCEL

When EXCEL is opened you will need to set Calculation to **Manual**. The spreadsheet should only perform calculations when you force it to. Follow the steps shown below to set manual calculation.

You will also need to add the **Calculate** button to your **Quick Access Toolbar** as shown below.

Spreadsheet Details

Now lets have a look at the spreadsheet.

The spreadsheet is designed for a maximum of 15 stock factor nodes and 10 optimization iterations. This is the maximum size that you should consider using. There are several fields that you need to be aware of before using the spreadsheet.

Reference Node Weights - The 15 entries in this array represent the current weight settings for each factor. The starting value should be set to 10.

Randomized Array of Node Weights - this is an array of random weight values generated from the reference node weights when the EXCEL manual calculate is forced. This array will be eventually be copied and pasted into the Portfolio123 ranking system optimizer.

Randomization Formula - this is the formula used to generate the randomized array (shown above). The formula takes the reference node weight and adds +/- random offset. The maximum offset is determined by the sensitivity setting. In the example above the sensitivity is 50% and as you can see the values in the randomized array range between 7.5 and 12.5. The range is 50% of the reference weight.

On the next post I will perform the first steps of the optimization.

Ranking System Optimizer: The First Iteration

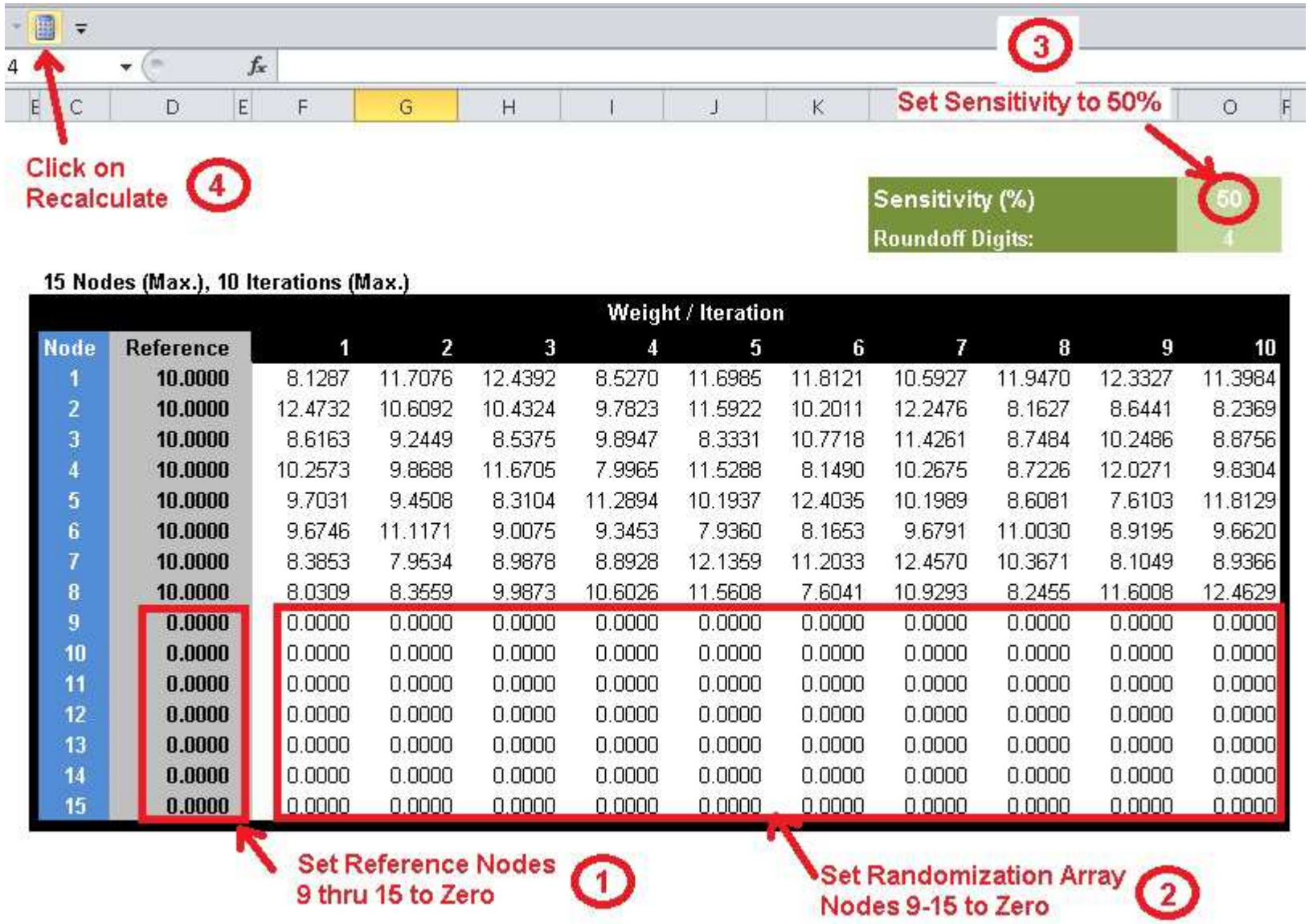
October 31, 2013

Up to this point, the reader has been introduced to the Portfolio123 rnking system optimizer and also the custom spreadsheet that I have prepared. Now it is time to use them both together. If possible, you should try to use two monitors with your computer. You don't have to, but it makes this procedure a lot easier as you will be going back and forth between the optimizer and spreadsheet frequently.

Lets start with the EXCEL spreadsheet.

Zero Out Unused Nodes

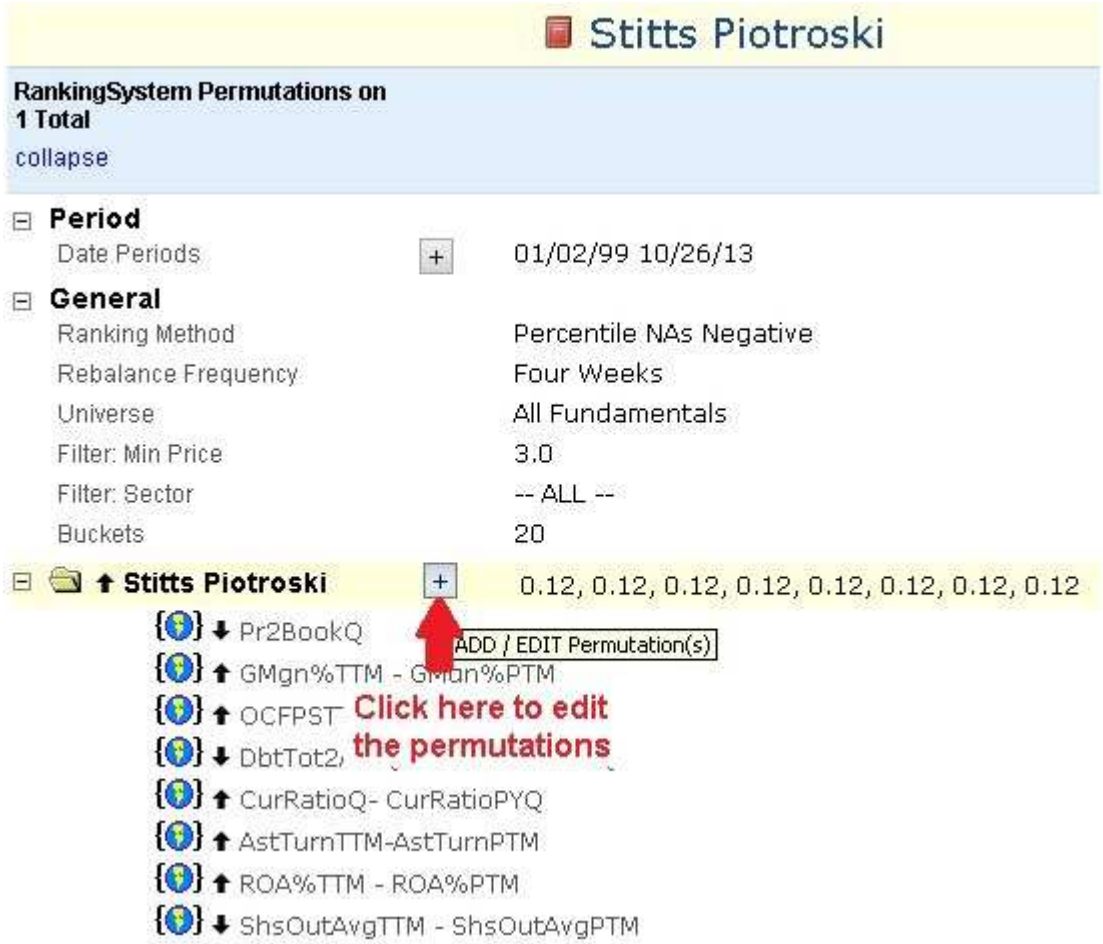
The Stitts Piotroski ranking system has eight nodes (stock factors/formula) so it is a good idea to zero out the remaining nodes in the spreadsheet as shown below. You don't have to perform this action but it helps visually to prevent copying the wrong size array in the future. It is also a good idea to manually recalculate just to ensure that you are starting with good randomized data. I recommend starting with 50% sensitivity setting although you can always experiment with this setting.



Node	Reference	1	2	3	4	5	6	7	8	9	10
1	10.0000	8.1287	11.7076	12.4392	8.5270	11.6985	11.8121	10.5927	11.9470	12.3327	11.3984
2	10.0000	12.4732	10.6092	10.4324	9.7823	11.5922	10.2011	12.2476	8.1627	8.6441	8.2369
3	10.0000	8.6163	9.2449	8.5375	9.8947	8.3331	10.7718	11.4261	8.7484	10.2486	8.8756
4	10.0000	10.2573	9.8688	11.6705	7.9965	11.5288	8.1490	10.2675	8.7226	12.0271	9.8304
5	10.0000	9.7031	9.4508	8.3104	11.2894	10.1937	12.4035	10.1989	8.6081	7.6103	11.8129
6	10.0000	9.6746	11.1171	9.0075	9.3453	7.9360	8.1653	9.6791	11.0030	8.9195	9.6620
7	10.0000	8.3853	7.9534	8.9878	8.8928	12.1359	11.2033	12.4570	10.3671	8.1049	8.9366
8	10.0000	8.0309	8.3559	9.9873	10.6026	11.5608	7.6041	10.9293	8.2455	11.6008	12.4629
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

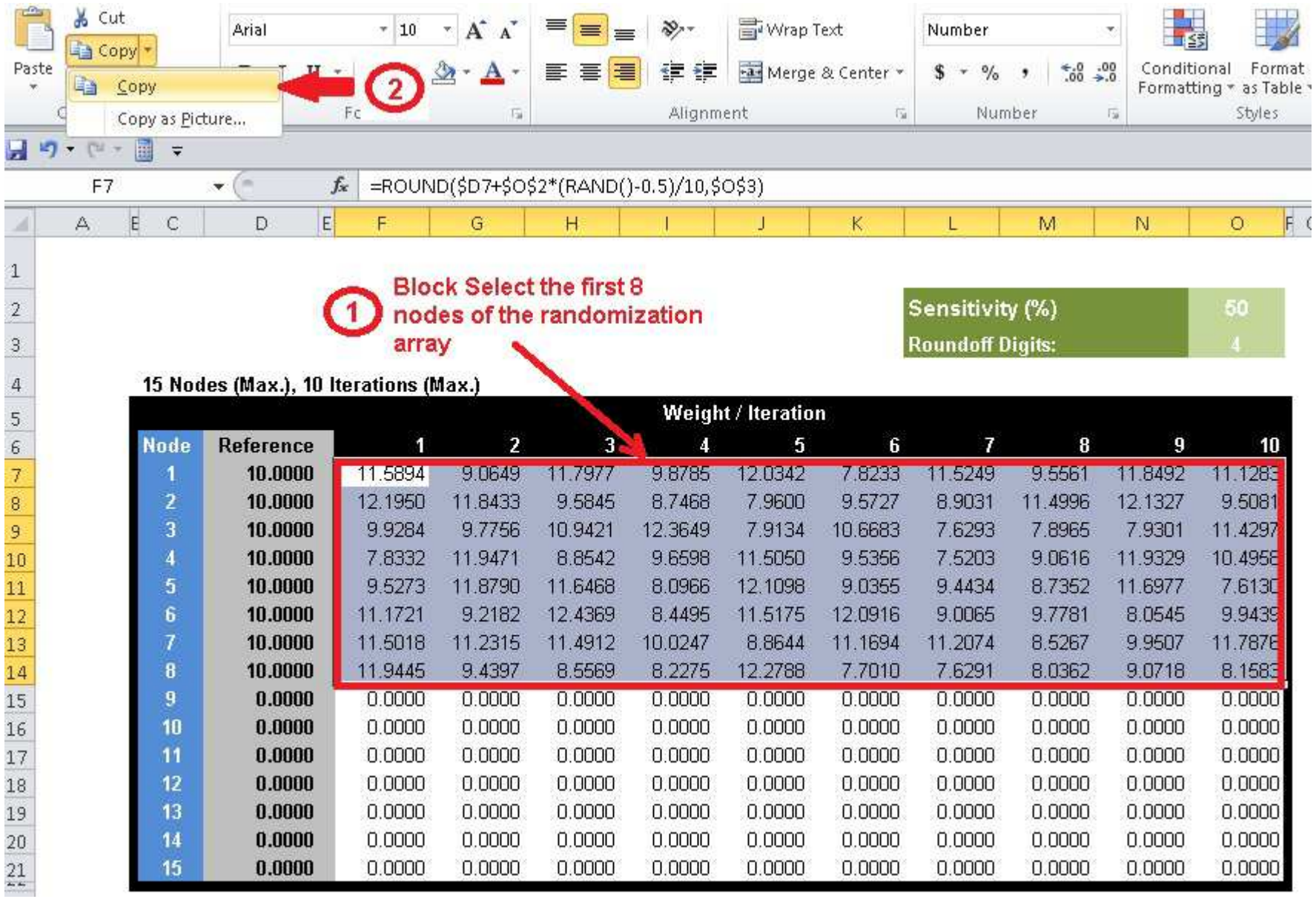
Generate Permutations

Now open up the ranking system study if not already open. We are about to perform the first iteration of the optimization process. Click on the button with a "plus sign" as shown below to add permutations.

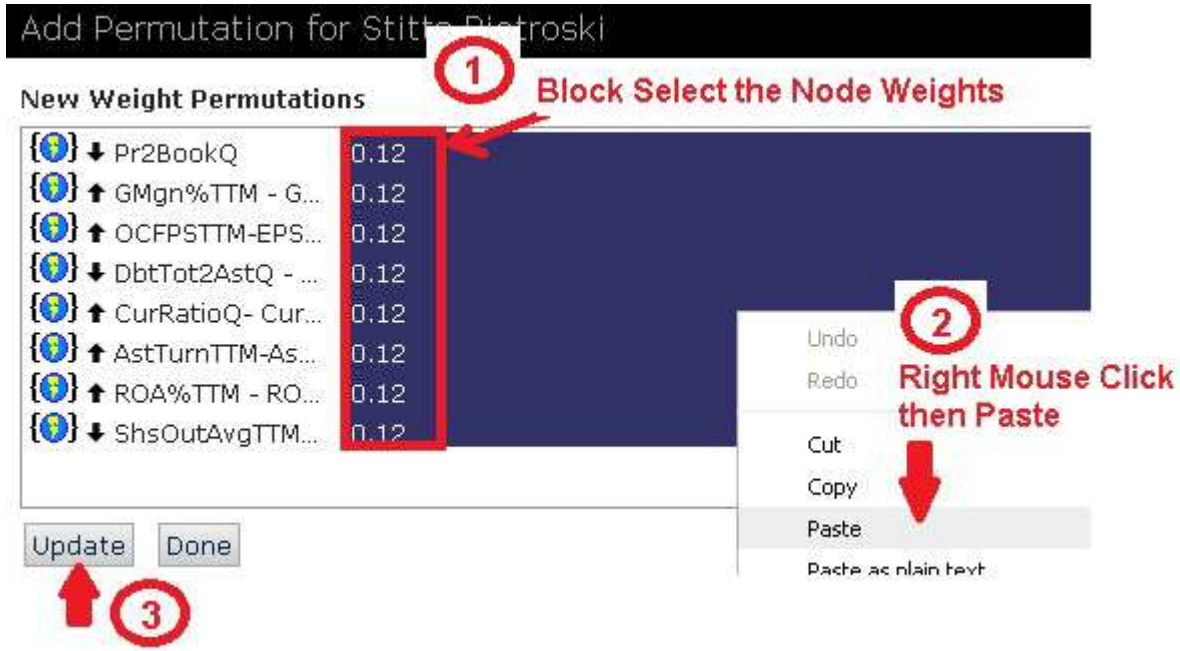


Switch back to the spreadsheet and copy the array of randomized node weights.

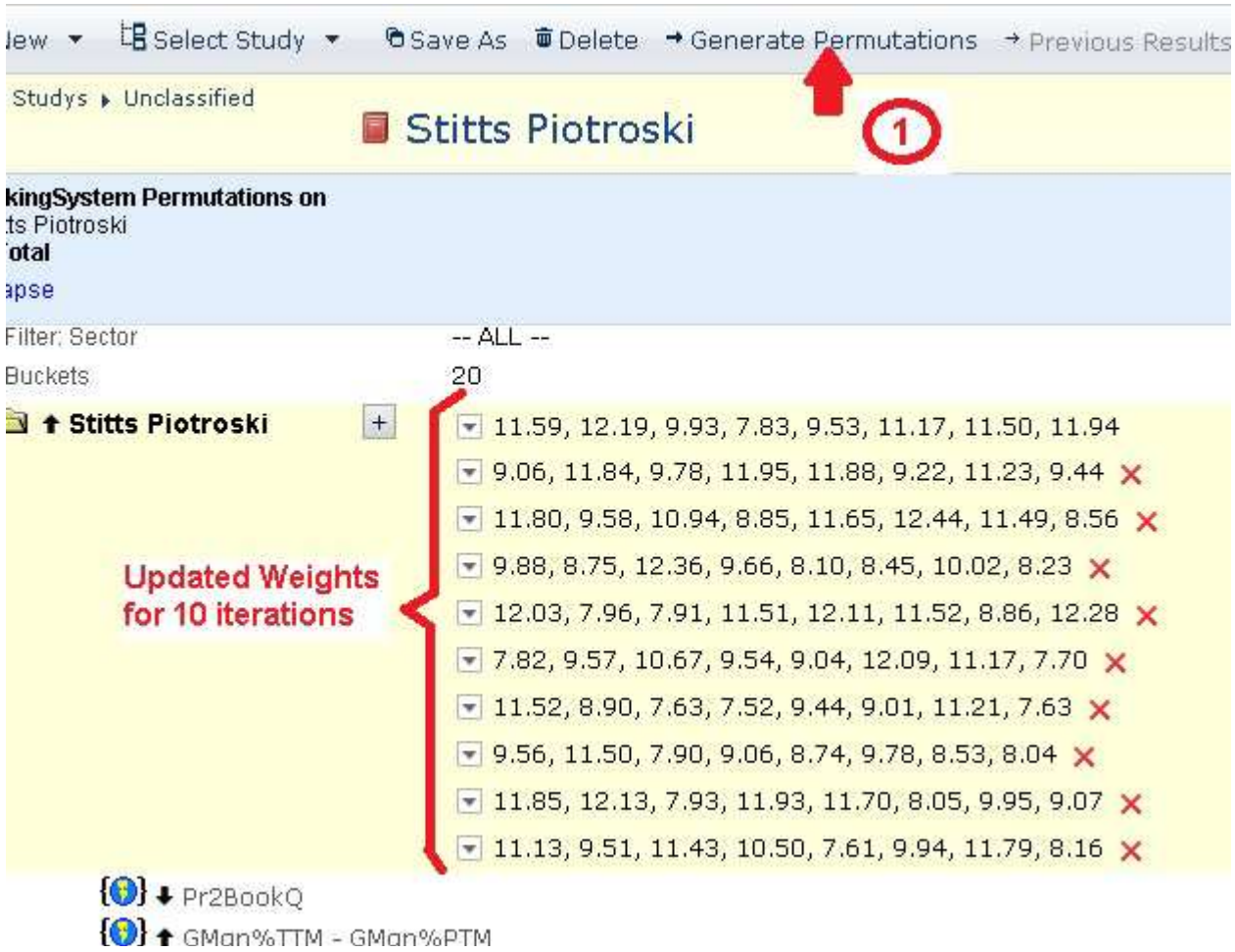
WARNING: Be sure not to copy the reference weights and also do not copy nodes 9 through 15. Doing so will cause an error because the wrong amount of data would be pasted into the ranking system optimizer.



Now return to the ranking system optimizer. You should be on the screen "**Add Permutation for Stitts Piotroski**". Select all of the permutations listed as shown in the figure below. Paste the 8 rows by 10 columns of randomized data, overwriting the original permutations listed in the ranking system optimizer. Then click on **Update**.

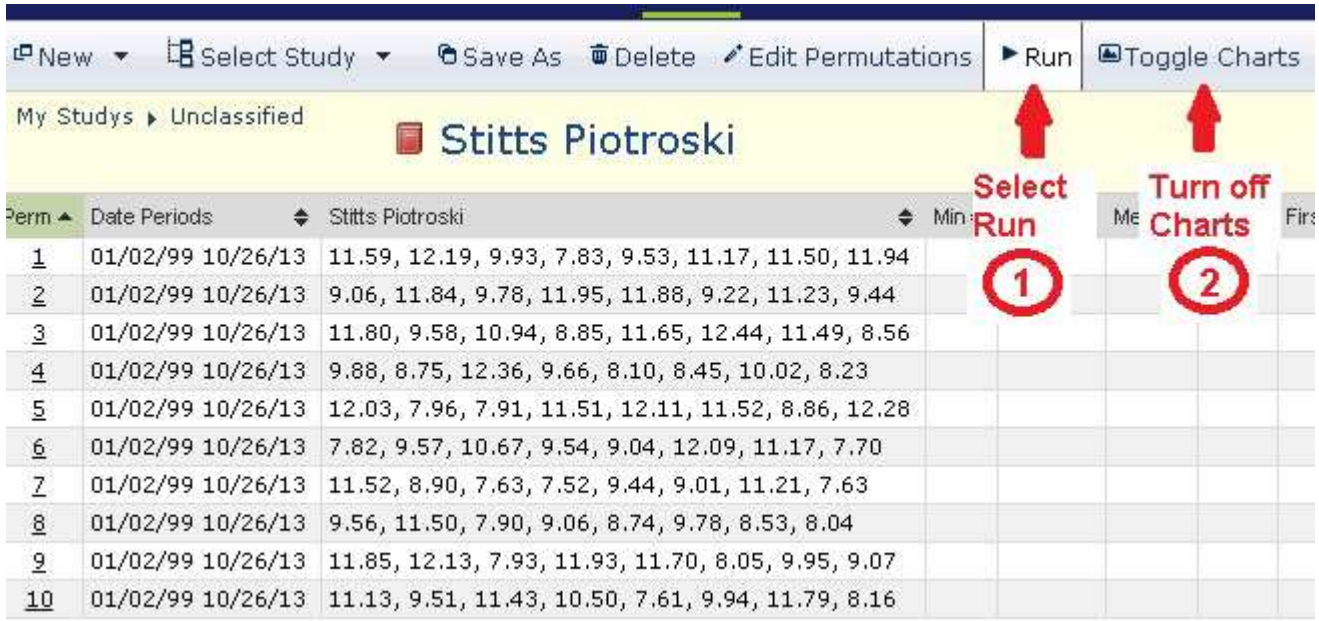


Now you will see the permutations listed as shown below. Click on **Generate Permutations**.



Process the Permutations

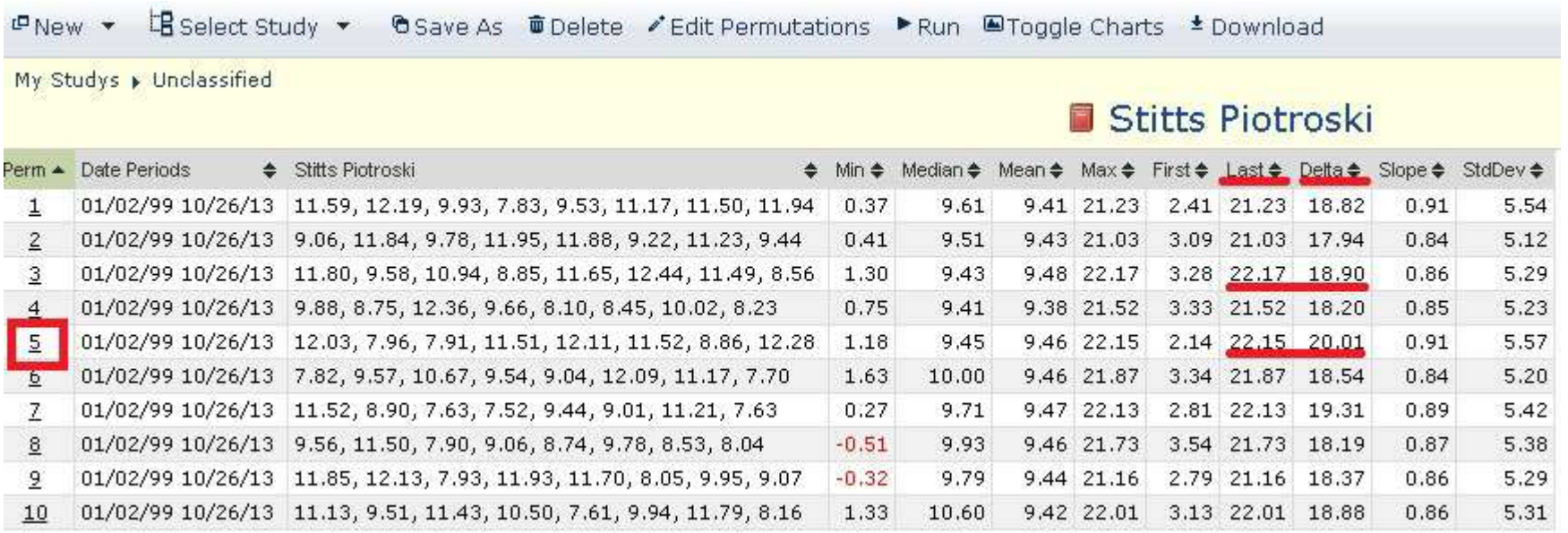
Select **Run** to start processing the 10 permutations. Click on **Toggle Charts** to disable the display of charts as they are not necessary for this exercise.



Perm	Date Periods	Stitts Piotroski	Min	Me	Firs
1	01/02/99 10/26/13	11.59, 12.19, 9.93, 7.83, 9.53, 11.17, 11.50, 11.94			
2	01/02/99 10/26/13	9.06, 11.84, 9.78, 11.95, 11.88, 9.22, 11.23, 9.44			
3	01/02/99 10/26/13	11.80, 9.58, 10.94, 8.85, 11.65, 12.44, 11.49, 8.56			
4	01/02/99 10/26/13	9.88, 8.75, 12.36, 9.66, 8.10, 8.45, 10.02, 8.23			
5	01/02/99 10/26/13	12.03, 7.96, 7.91, 11.51, 12.11, 11.52, 8.86, 12.28			
6	01/02/99 10/26/13	7.82, 9.57, 10.67, 9.54, 9.04, 12.09, 11.17, 7.70			
7	01/02/99 10/26/13	11.52, 8.90, 7.63, 7.52, 9.44, 9.01, 11.21, 7.63			
8	01/02/99 10/26/13	9.56, 11.50, 7.90, 9.06, 8.74, 9.78, 8.53, 8.04			
9	01/02/99 10/26/13	11.85, 12.13, 7.93, 11.93, 11.70, 8.05, 9.95, 9.07			
10	01/02/99 10/26/13	11.13, 9.51, 11.43, 10.50, 7.61, 9.94, 11.79, 8.16			

Choosing the Best Permutation

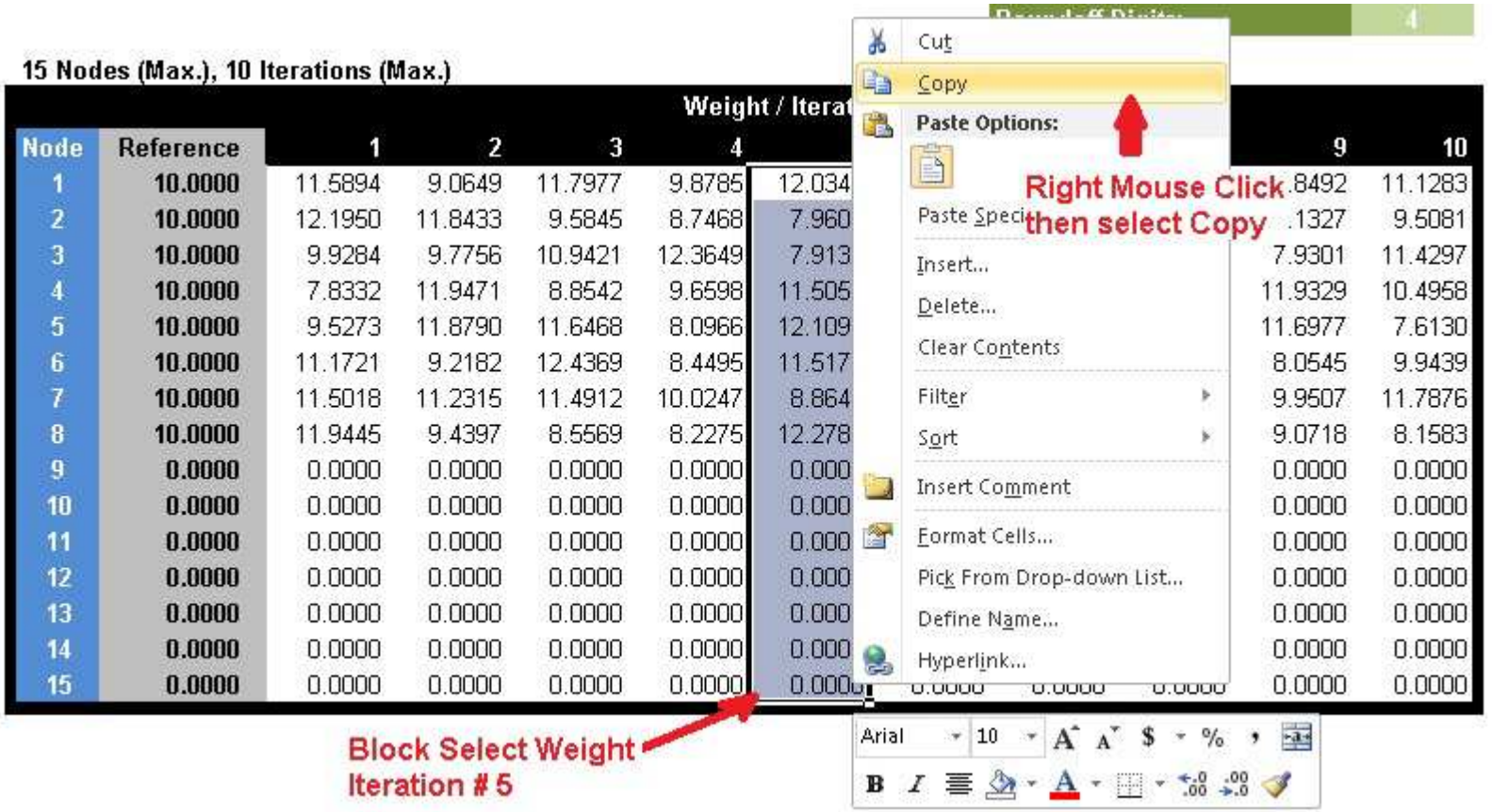
When the permutations are finished processing you will see the results. At this point in time you must choose the "best" result from the 10 permutations. Now this is very much a personal decision. I look at the last bucket first, and the Delta (highest bucket to lowest bucket) second. As you can see from the figure below, permutation #3 has the highest last bucket with 22.17%. However, permutation #5 is a fraction lower but has a much larger delta of 20.01 versus 18.90. Thus I decided that permutation #5 is the best permutation. As I said before, there is no right or wrong choice so don't get hung up on your decision.



Perm	Date Periods	Stitts Piotroski	Min	Median	Mean	Max	First	Last	Delta	Slope	StdDev
1	01/02/99 10/26/13	11.59, 12.19, 9.93, 7.83, 9.53, 11.17, 11.50, 11.94	0.37	9.61	9.41	21.23	2.41	21.23	18.82	0.91	5.54
2	01/02/99 10/26/13	9.06, 11.84, 9.78, 11.95, 11.88, 9.22, 11.23, 9.44	0.41	9.51	9.43	21.03	3.09	21.03	17.94	0.84	5.12
3	01/02/99 10/26/13	11.80, 9.58, 10.94, 8.85, 11.65, 12.44, 11.49, 8.56	1.30	9.43	9.48	22.17	3.28	22.17	18.90	0.86	5.29
4	01/02/99 10/26/13	9.88, 8.75, 12.36, 9.66, 8.10, 8.45, 10.02, 8.23	0.75	9.41	9.38	21.52	3.33	21.52	18.20	0.85	5.23
5	01/02/99 10/26/13	12.03, 7.96, 7.91, 11.51, 12.11, 11.52, 8.86, 12.28	1.18	9.45	9.46	22.15	2.14	22.15	20.01	0.91	5.57
6	01/02/99 10/26/13	7.82, 9.57, 10.67, 9.54, 9.04, 12.09, 11.17, 7.70	1.63	10.00	9.46	21.87	3.34	21.87	18.54	0.84	5.20
7	01/02/99 10/26/13	11.52, 8.90, 7.63, 7.52, 9.44, 9.01, 11.21, 7.63	0.27	9.71	9.47	22.13	2.81	22.13	19.31	0.89	5.42
8	01/02/99 10/26/13	9.56, 11.50, 7.90, 9.06, 8.74, 9.78, 8.53, 8.04	-0.51	9.93	9.46	21.73	3.54	21.73	18.19	0.87	5.38
9	01/02/99 10/26/13	11.85, 12.13, 7.93, 11.93, 11.70, 8.05, 9.95, 9.07	-0.32	9.79	9.44	21.16	2.79	21.16	18.37	0.86	5.29
10	01/02/99 10/26/13	11.13, 9.51, 11.43, 10.50, 7.61, 9.94, 11.79, 8.16	1.33	10.60	9.42	22.01	3.13	22.01	18.88	0.86	5.31

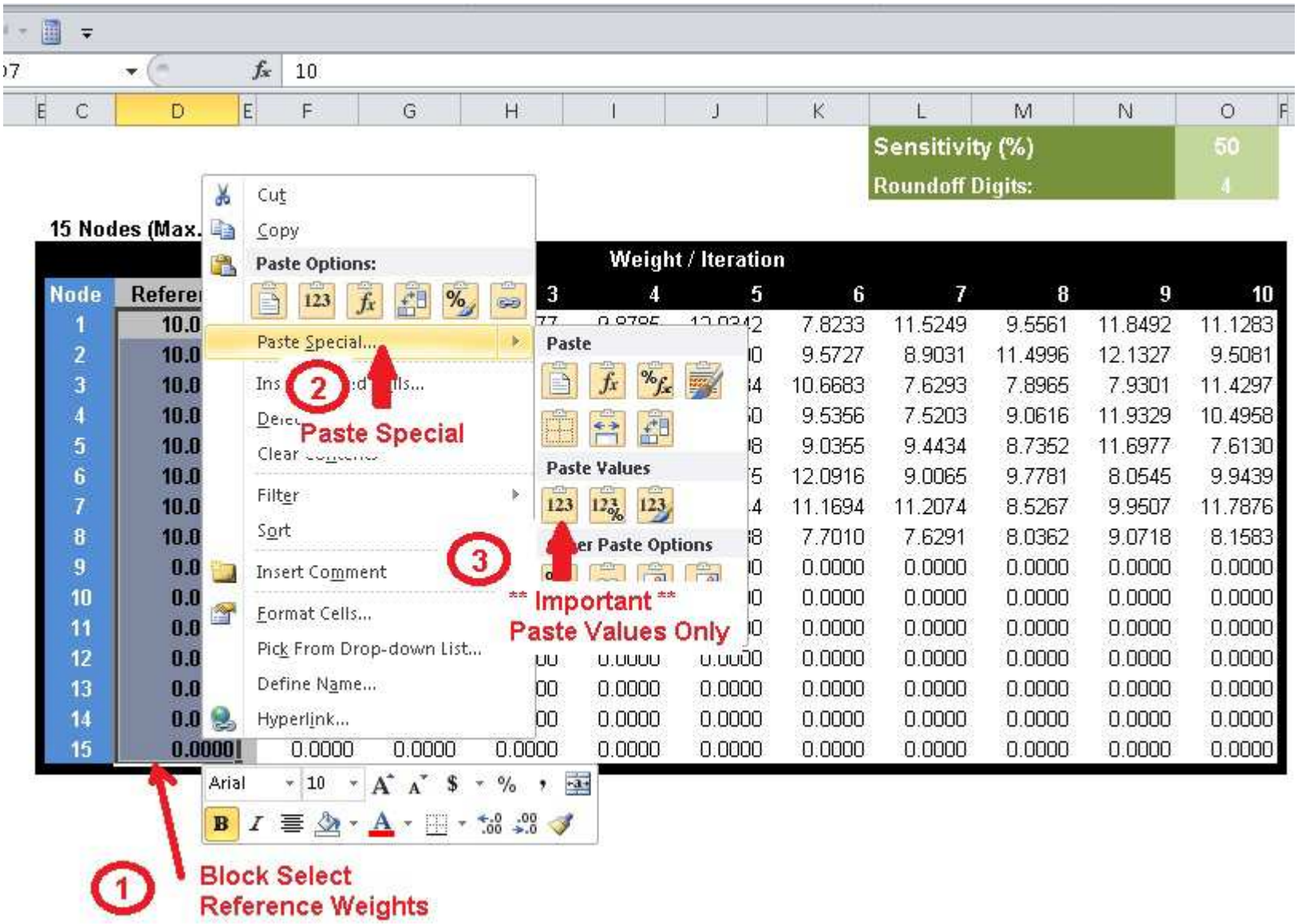
Updating the Spreadsheet

Now that I have chosen permutation #5 I can go back and update the spreadsheet. This is done by copying iteration #5 from the randomized array as shown below.

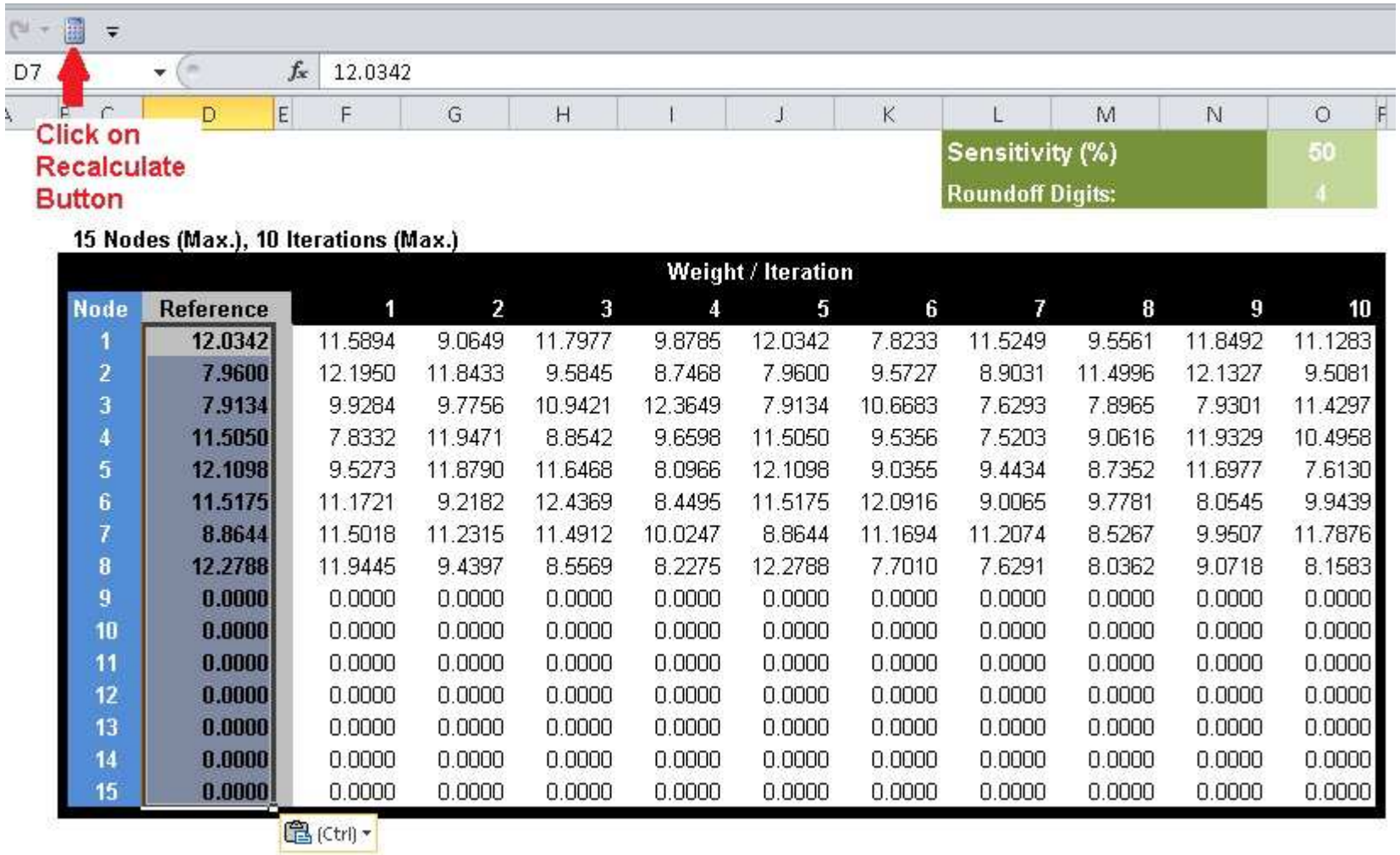


Now paste the weights from iteration #5 into the reference array.

WARNING: Make sure that you paste **Values Only into the reference array. In the past I have succeeded in completely locking up my spreadsheet by pasting values and formulas. It isn't easy to recover from such an event.**



One iteration of the optimization process has now been completed. This will be performed over and over again, starting with **Manual Calculation**.



That completes this portion of the ranking system optimizer tutorial. The next post will focus on pruning the nodes and completing the optimization.

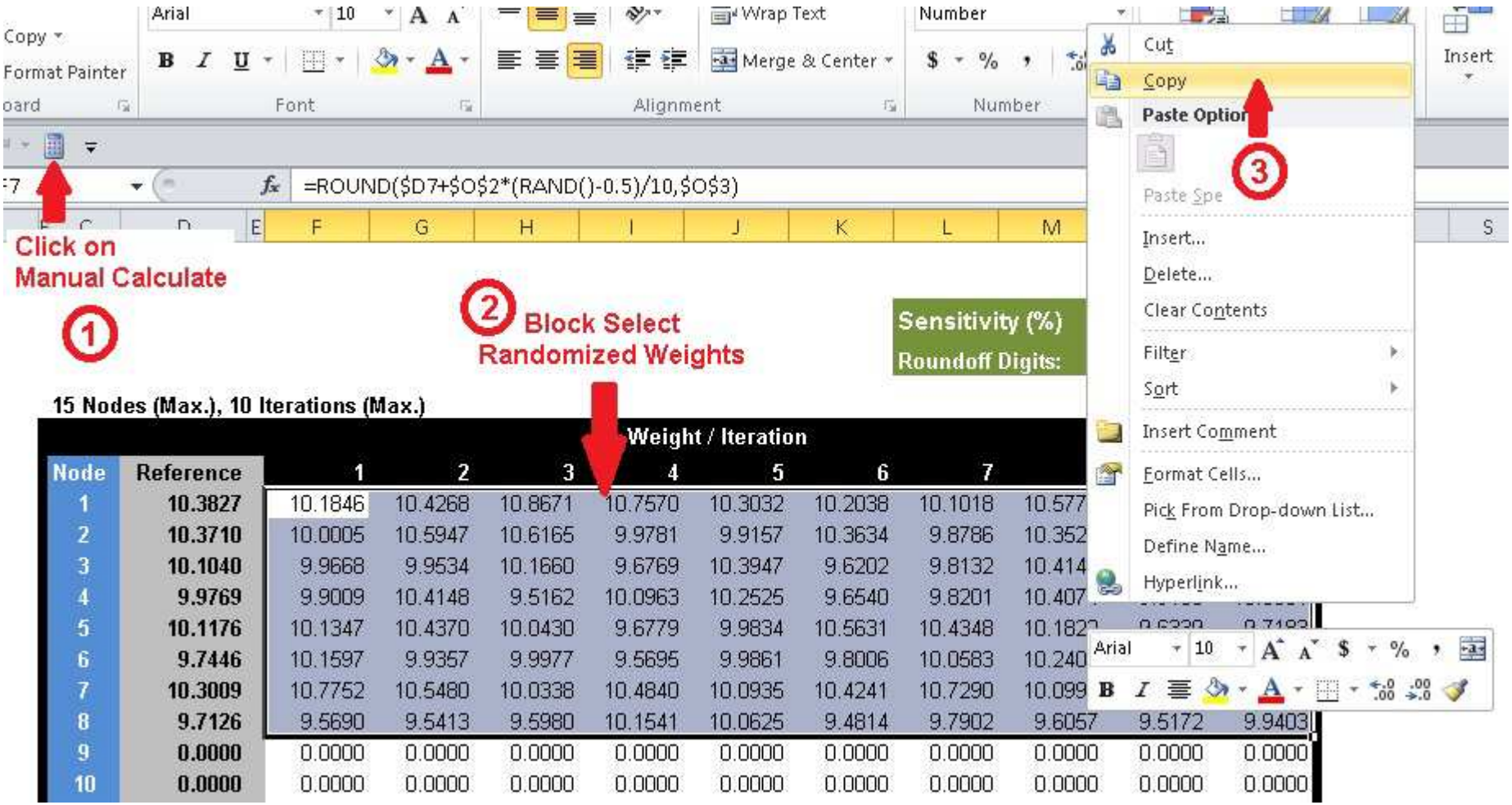
Pruning Nodes From The Ranking System

November 1, 2013

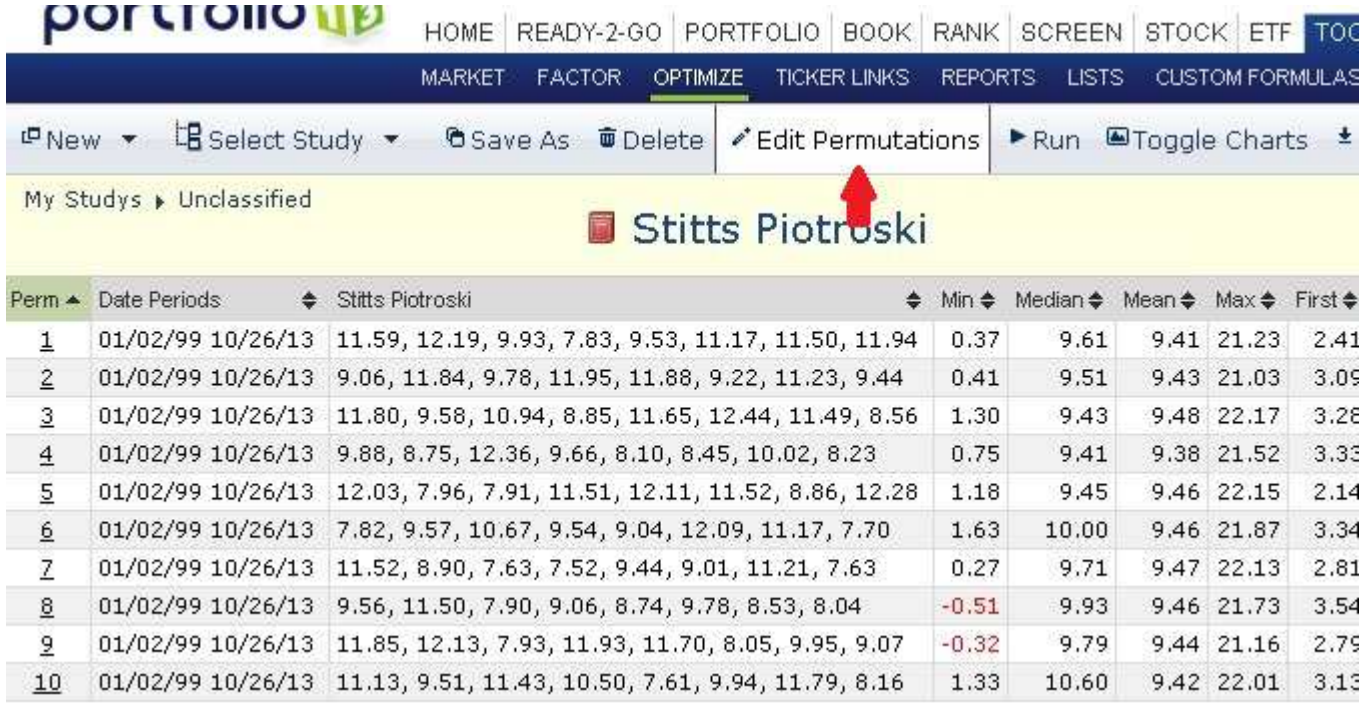
Recap of the Optimization Sequence

In the last post I described the sequence of actions required to generate new permutations, run the optimizer and select the best permution. This is repeated over and over again until you decide you can't improve the performance any further. Before going any further I would like to recap the iterative optimization process.

From the spreadsheet you start by clicking on the manual calculate button. This causes new randomized data to be generated from the reference weights. Then block select the randomized data (8 nodes x 10 iterations). Copy the data.



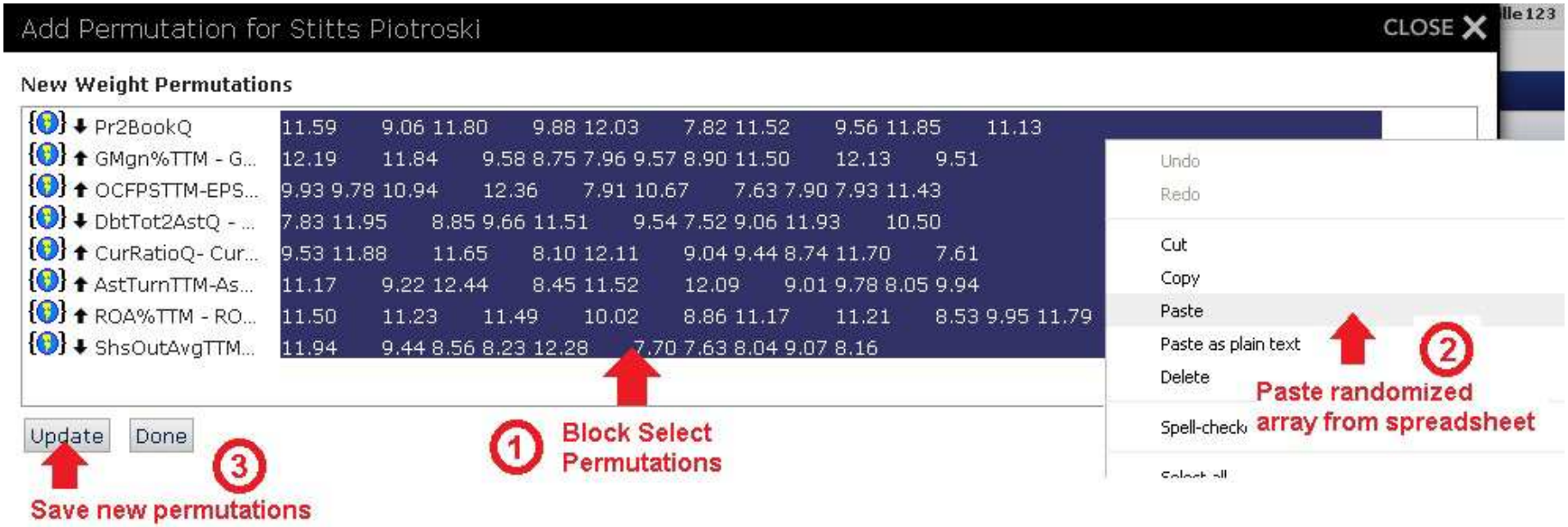
At Portfolio123, select **Edit Permutations** if you are still at the last optimizer results page. Otherwise skip this step.



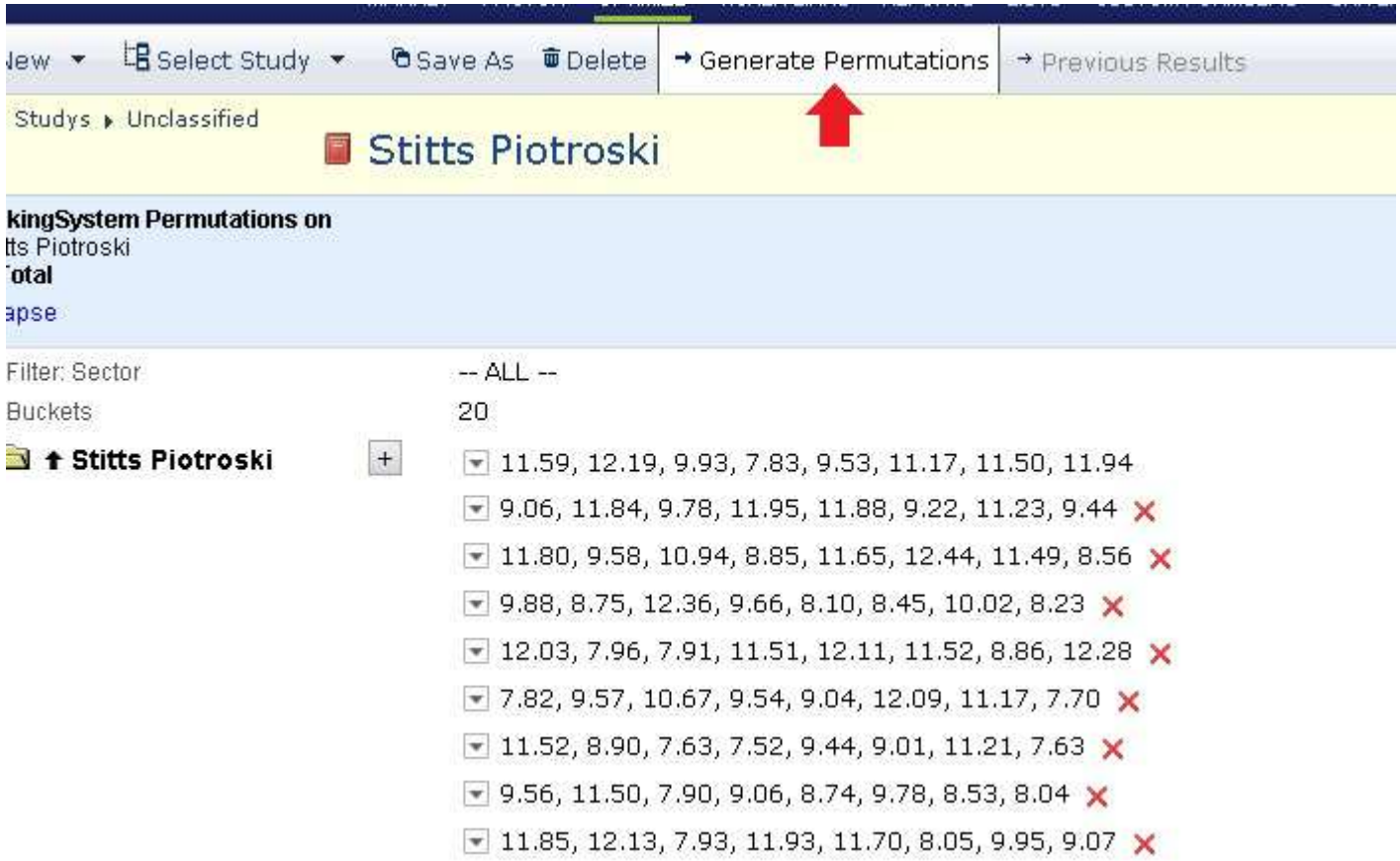
Perm	Date Periods	Stitts Piotroski	Min	Median	Mean	Max	First
1	01/02/99 10/26/13	11.59, 12.19, 9.93, 7.83, 9.53, 11.17, 11.50, 11.94	0.37	9.61	9.41	21.23	2.41
2	01/02/99 10/26/13	9.06, 11.84, 9.78, 11.95, 11.88, 9.22, 11.23, 9.44	0.41	9.51	9.43	21.03	3.09
3	01/02/99 10/26/13	11.80, 9.58, 10.94, 8.85, 11.65, 12.44, 11.49, 8.56	1.30	9.43	9.48	22.17	3.28
4	01/02/99 10/26/13	9.88, 8.75, 12.36, 9.66, 8.10, 8.45, 10.02, 8.23	0.75	9.41	9.38	21.52	3.33
5	01/02/99 10/26/13	12.03, 7.96, 7.91, 11.51, 12.11, 11.52, 8.86, 12.28	1.18	9.45	9.46	22.15	2.14
6	01/02/99 10/26/13	7.82, 9.57, 10.67, 9.54, 9.04, 12.09, 11.17, 7.70	1.63	10.00	9.46	21.87	3.34
7	01/02/99 10/26/13	11.52, 8.90, 7.63, 7.52, 9.44, 9.01, 11.21, 7.63	0.27	9.71	9.47	22.13	2.81
8	01/02/99 10/26/13	9.56, 11.50, 7.90, 9.06, 8.74, 9.78, 8.53, 8.04	-0.51	9.93	9.46	21.73	3.54
9	01/02/99 10/26/13	11.85, 12.13, 7.93, 11.93, 11.70, 8.05, 9.95, 9.07	-0.32	9.79	9.44	21.16	2.79
10	01/02/99 10/26/13	11.13, 9.51, 11.43, 10.50, 7.61, 9.94, 11.79, 8.16	1.33	10.60	9.42	22.01	3.13

Click on the small plus sign button to modify the ranking system weights.

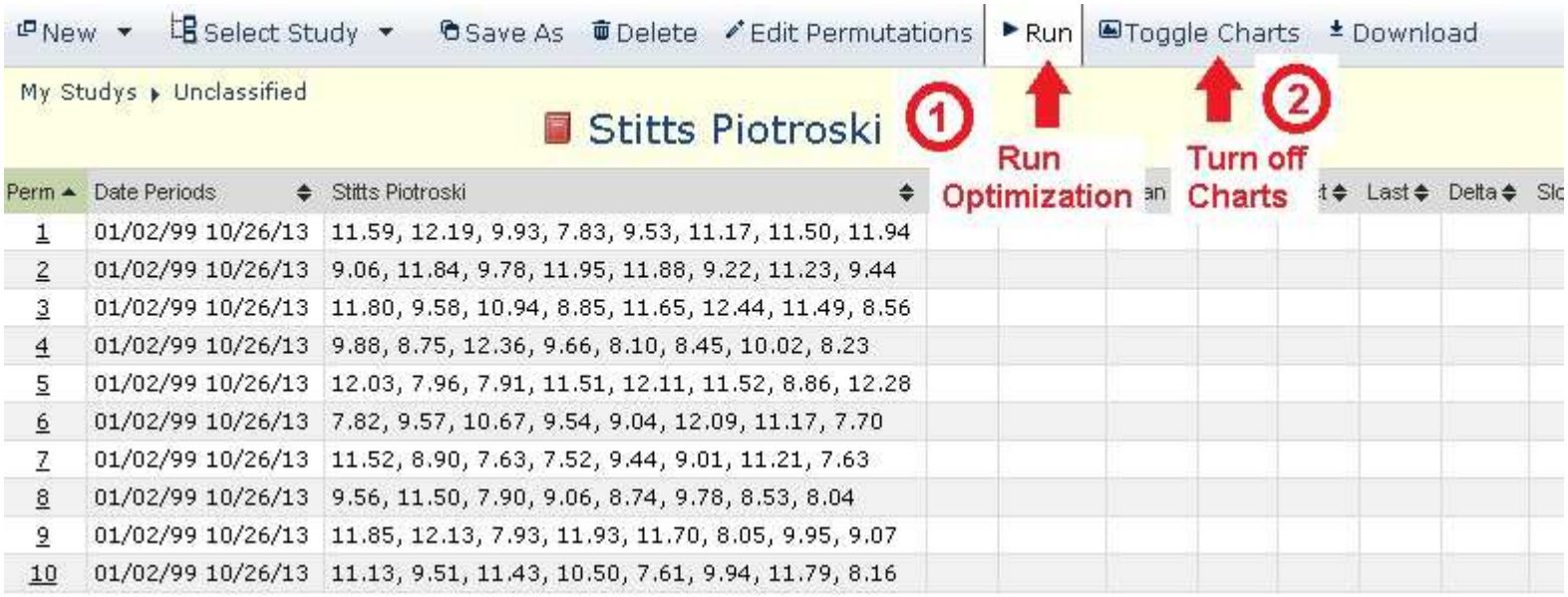
Block select the permutations that are already there. Right click with your mouse and overwrite the existing permutations with the new iterations. Save the new permutations.



Generate the permutations.



Click on Run and then click on Toggle Charts. Wait for the optimizer to process all of the permutations.



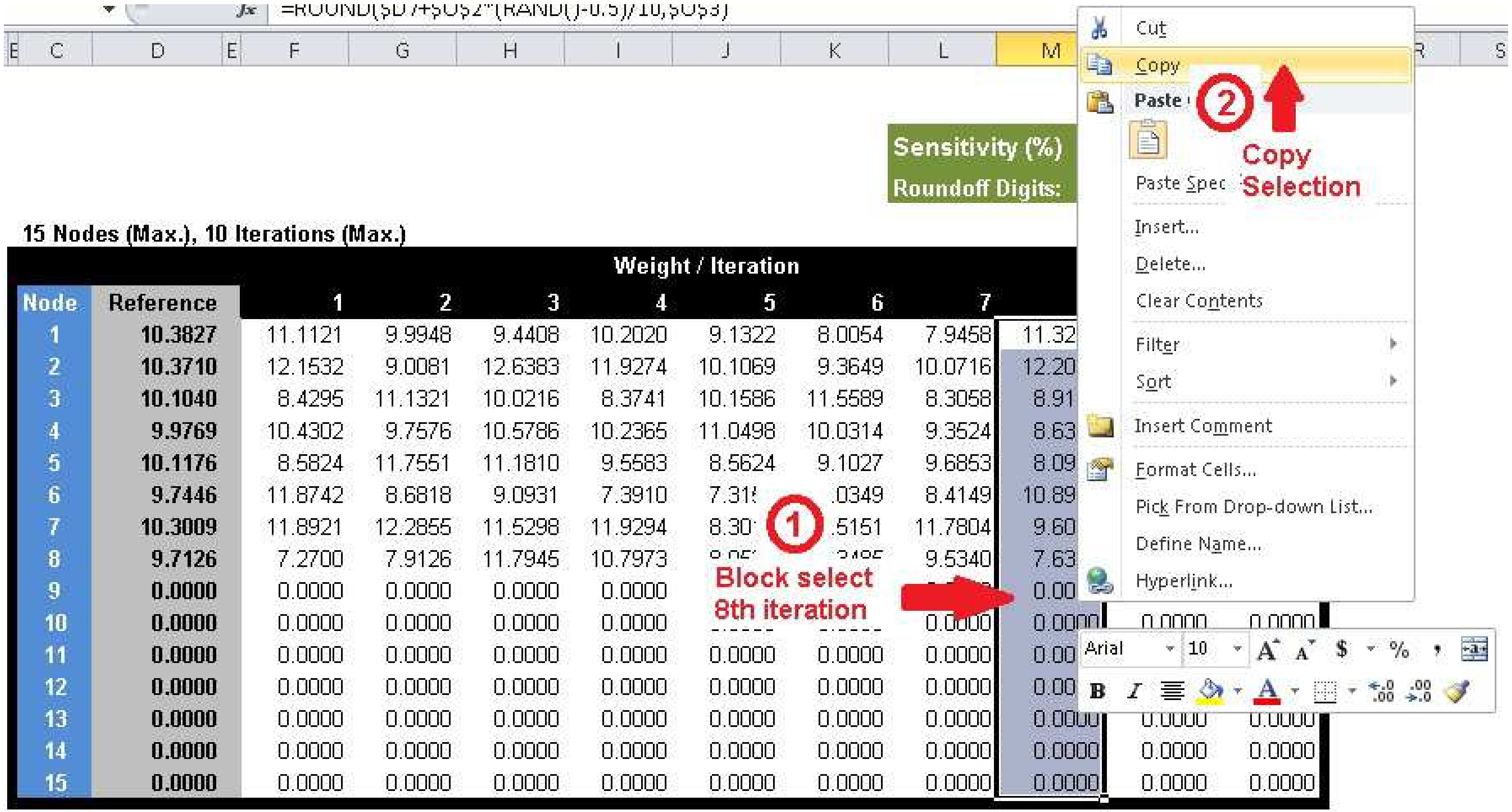
Perm	Date Periods	Stitts Piotroski	Last	Delta
1	01/02/99 10/26/13	11.59, 12.19, 9.93, 7.83, 9.53, 11.17, 11.50, 11.94		
2	01/02/99 10/26/13	9.06, 11.84, 9.78, 11.95, 11.88, 9.22, 11.23, 9.44		
3	01/02/99 10/26/13	11.80, 9.58, 10.94, 8.85, 11.65, 12.44, 11.49, 8.56		
4	01/02/99 10/26/13	9.88, 8.75, 12.36, 9.66, 8.10, 8.45, 10.02, 8.23		
5	01/02/99 10/26/13	12.03, 7.96, 7.91, 11.51, 12.11, 11.52, 8.86, 12.28		
6	01/02/99 10/26/13	7.82, 9.57, 10.67, 9.54, 9.04, 12.09, 11.17, 7.70		
7	01/02/99 10/26/13	11.52, 8.90, 7.63, 7.52, 9.44, 9.01, 11.21, 7.63		
8	01/02/99 10/26/13	9.56, 11.50, 7.90, 9.06, 8.74, 9.78, 8.53, 8.04		
9	01/02/99 10/26/13	11.85, 12.13, 7.93, 11.93, 11.70, 8.05, 9.95, 9.07		
10	01/02/99 10/26/13	11.13, 9.51, 11.43, 10.50, 7.61, 9.94, 11.79, 8.16		

Choose what you consider to be the best permutation. In this example, I chose permutation #8.

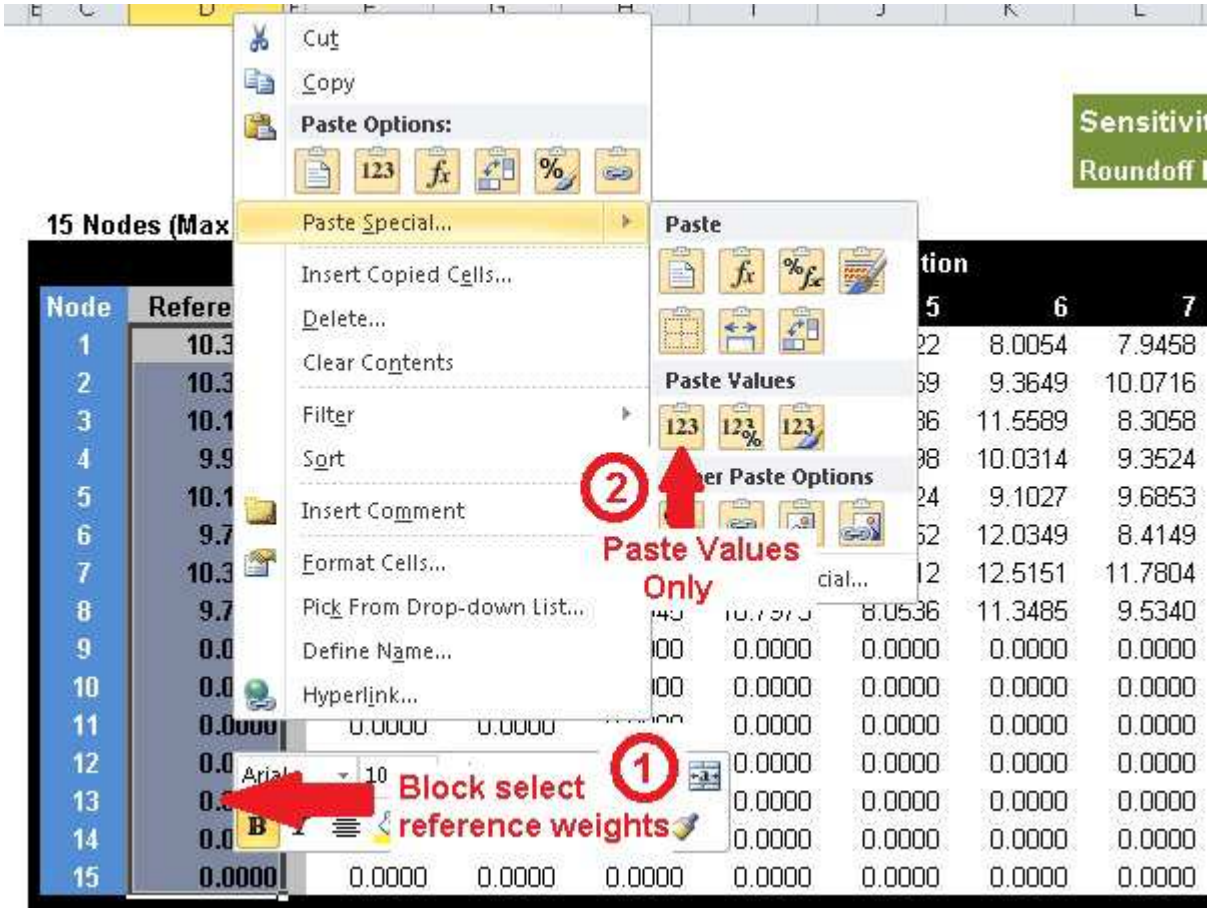
Stitts Piotroski

Perm	Date Periods	Min	Median	Mean	Max	First	Last	Delta	Slope	StdDev
1	01/02/99 10/26/13	0.78	10.13	9.50	22.20	3.19	22.20	19.02	0.86	5.29
2	01/02/99 10/26/13	1.39	9.40	9.43	20.89	2.95	20.89	17.94	0.83	5.06
3	01/02/99 10/26/13	0.09	9.04	9.39	21.38	2.88	21.38	18.51	0.88	5.39
4	01/02/99 10/26/13	0.56	9.04	9.38	21.04	2.35	21.04	18.69	0.88	5.36
5	01/02/99 10/26/13	0.26	9.46	9.39	21.22	3.11	21.22	18.11	0.85	5.21
6	01/02/99 10/26/13	1.77	9.96	9.41	21.69	2.71	21.69	18.98	0.88	5.35
7	01/02/99 10/26/13	0.50	9.15	9.42	21.21	2.84	21.21	18.37	0.87	5.32
8	01/02/99 10/26/13	0.93	9.68	9.47	22.30	3.04	22.30	19.26	0.88	5.37
9	01/02/99 10/26/13	1.71	9.59	9.43	21.18	3.03	21.18	18.15	0.84	5.18
10	01/02/99 10/26/13	0.80	9.99	9.40	22.21	2.61	22.21	19.60	0.88	5.39

Now return to the spreadsheet, block select iteration #8 from the randomized data array, and copy it.



Paste iteration #8 into the reference node array.



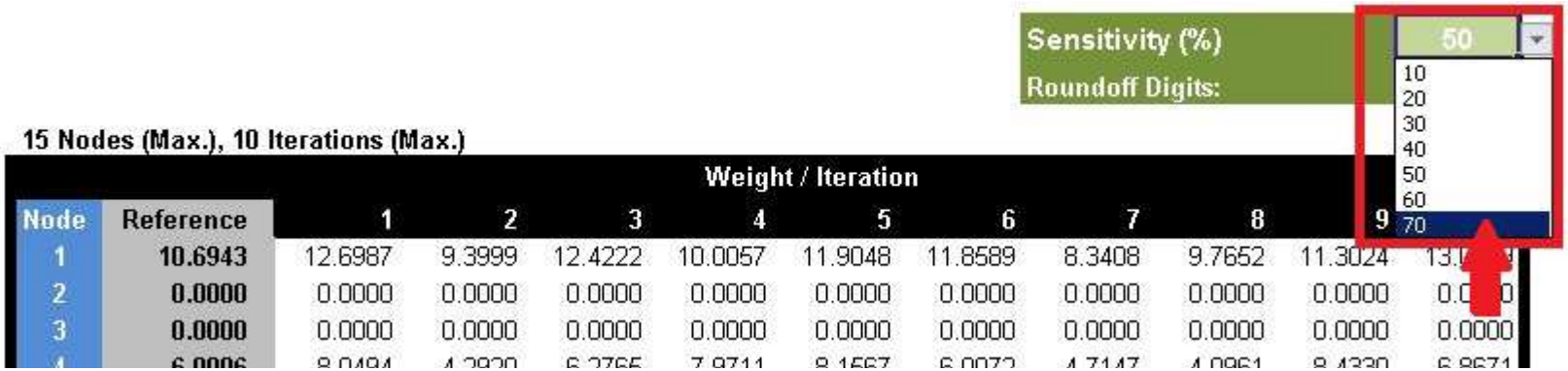
This completes one optimization cycle.

Hitting The Brick Wall

This seems to be pretty easy although a bit monotonous. But you will find that it doesn't take too long before you run into a brick wall. The performance of the top bucket will stop increasing no matter how many times you generate new permutations.

At this point in time you may need to shake things up a little bit. This is kind of like being in a maze, running into a dead end and having to backtrack before you can go forward again.

Try using a different tactic for deciding on the "best" permutation. I find that selecting the lowest first bucket while ignoring the top bucket for one or two iterations often works. You might also consider upping the sensitivity to 70% until you find a new "best" permutation.



Pruning Nodes

One of the interesting aspects of this optimization process is that you can eliminate nodes (stock factors) without sacrificing performance. Think of this as pruning a plant, cutting out the dead leaves, branches allowing the plant's energy to be focused on the healthy parts.

The time to prune is when a reference weight decreases to about 1 (from the original 10).

15 Nodes (Max.), 10 Iterations (Max.)

		Weight / Iteration									
Node	Reference	1	2	3	4	5	6	7	8	9	10
1	14.9543	16.3100	12.8632	17.0015	13.3853	14.6973	14.1632	12.6196	15.3276	13.3705	14.5377
2	2.9938	3.2861	4.0244	1.9659	3.5905	1.3295	4.1167	3.2741	0.6670	3.0871	3.1597
3	2.3156	4.0130	3.3490	1.9846	4.7300	3.9908	1.7646	3.6552	3.1950	0.1366	4.3330
4	5.6671	7.5038	7.9233	7.2758	7.4925	7.8271	4.5384	4.7778	5.0167	7.9613	5.1651
5	0.1763	1.8448	0.5701	1.2218	0.3853	1.0507	2.3586	0.1084	0.0997	0.1781	1.2030
6	10.0611	10.7830	8.8583	12.4182	8.5601	7.7350	10.0188	8.3795	12.2871	11.9686	12.1345
7	12.3027	12.1263	11.7541	14.4810	12.0404	11.2399	13.0911	13.7579	13.6234	10.7123	12.5339
8	11.3757	8.9722	9.1500	10.3179	12.7062	13.0294	11.6828	9.6474	13.8647	12.7307	12.3536
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

As you can see from the example above, node 5 has a reference weight of 0.1763. It is time to eliminate this node. This is done by zeroing out the reference weight and the corresponding randomized data. Note that Nodes 2 and 3 are almost ready to be pruned as well, but not this time around.

15 Nodes (Max.), 10 Iterations (Max.)

		Weight / Iteration									
Node	Reference	1	2	3	4	5	6	7	8	9	10
1	**14.9543**	16.3100	12.8632	17.0015	13.3853	14.6973	14.1632	12.6196	15.3276	13.3705	14.5377
2	**2.9938**	3.2861	4.0244	1.9659	3.5905	1.3295	4.1167	3.2741	0.6670	3.0871	3.1597
3	**2.3156**	4.0130	3.3490	1.9846	4.7300	3.9908	1.7646	3.6552	3.1950	0.1366	4.3330
4	**5.6671**	7.5038	7.9233	7.2758	7.4925	7.8271	4.5384	4.7778	5.0167	7.9613	5.1651
5	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	**10.0611**	10.7830	8.8583	12.4182	8.5601	7.7350	10.0188	8.3795	12.2871	11.9686	12.1345
7	**12.3027**	12.1263	11.7541	14.4810	12.0404	11.2399	13.0911	13.7579	13.6234	10.7123	12.5339
8	**11.3757**	8.9722	9.1500	10.3179	12.7062	13.0294	11.6828	9.6474	13.8647	12.7307	12.3536
9	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Once the node is zeroed out then continue on with the same steps as before. The optimizer can handle the zero weights.

Pruning nodes often results in a performance setback with the last bucket dropping in value. But it is usually made up fairly quickly with subsequent optimization iterations.

When you do achieve a new high for the last bucket it is a good idea to set aside the reference weights for future use. Simply copy and paste into an unused section of the spreadsheet. You can always come back to these numbers at a later time if need be.

15 Nodes (Max.), 10 Iterations (Max.)

		Weight / Iteration									
Node	Reference	1	2	3	4	5	6	7	8	9	10
1	10.2529	9.7203	7.7482	8.6547	13.3543	13.6212	7.1741	8.0610	11.3500	13.6614	9.6826
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	7.4351	8.6489	7.8951	5.2059	4.5953	4.7132	6.5669	4.7064	10.5503	4.8384	7.8087
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	7.4381	5.6647	4.6553	5.8003	4.5245	9.0254	5.0463	4.3438	10.7415	7.0335	6.7277
7	18.3135	[illegible]	18.6923	17.8851	21.7608	[illegible]	14.9232	17.7709	20.4440	[illegible]	16.0120
8	6.5182	9.6878	6.5357	9.7544	9.6964	[illegible]	3.4435	3.7635	4.6309	[illegible]	5.8188
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

10.25
0
0
7.435
0
7.438
18.31
6.518
0
0
0
0
0
0
0

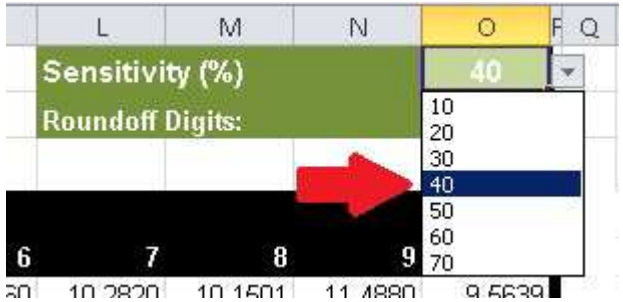
So that is the optimization process. Next post I'll outline how to finish it off, create the final ranking system, and as a bonus, create ten (or more) noisy ranking systems for robustness testing of future models.

The Final Ranking System

November 2, 2013

Reduce The Sensitivity Setting

In the last post it was determined that enough optimization had been performed and it was time to wrap up this project. At this point in time I suggest performing a few more optimization cycles while reducing the sensitivity from 50 down to 40, 30, 20 and finally 10. Doing so will find a local maximum. The optimization steps are the same as described in the last post. You only need to change the sensitivity setting in the spreadsheet.



Generate Noisy Test Ranking Systems

Once this has been done and the reference weights have been updated and considered to be final then the next step is to generate a set on noisy ranking systems for future test purposes. This is accomplished by doing a manual recalculate on the spreadsheet, then copying the randomized data into the ranking system optimizer permutations, and finally running the ranking system optimizer.

The ten permutations should be saved as individual ranking systems. This is done by selecting the permutation as shown below.

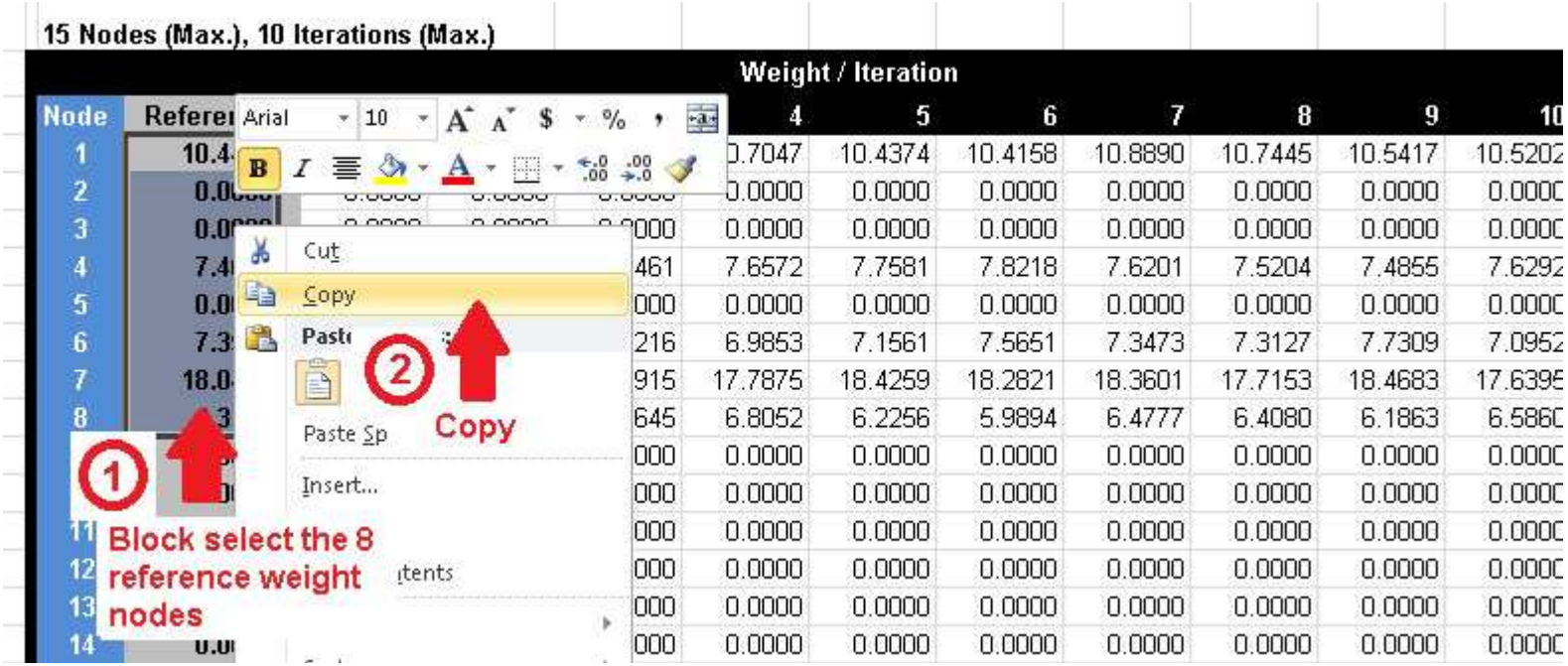
Stitts Piotroski

Perm	Date Periods	Stitts Piotroski	Min	Median	Mean	Max	First	Last	Delta	Slope	StdDev
1	/99 10/26/13	10.35, 0.00, 0.00, 7.71, 0.00, 7.07, 17.70, 6.58	0.46	9.21	9.58	23.30	1.54	23.30	21.76	0.85	5.43
2	01/02/99 10/26/13	10.62, 0.00, 0.00, 7.89, 0.00, 7.35, 18.14, 5.99	0.44	9.52	9.59	22.98	1.66	22.98	21.32	0.84	5.36
3	01/02/99 10/26/13	10.21, 0.00, 0.00, 7.85, 0.00, 7.42, 17.89, 6.16	0.76	9.29	9.60	23.08	1.58	23.08	21.50	0.84	5.35
4	01/02/99 10/26/13	10.70, 0.00, 0.00, 7.66, 0.00, 6.99, 17.79, 6.81	0.72	9.19	9.57	23.39	1.34	23.39	22.05	0.86	5.48
5	01/02/99 10/26/13	10.44, 0.00, 0.00, 7.76, 0.00, 7.16, 18.43, 6.23	0.33	9.47	9.59	23.40	1.67	23.40	21.73	0.84	5.41
6	01/02/99 10/26/13	10.42, 0.00, 0.00, 7.82, 0.00, 7.57, 18.28, 5.99	0.53	9.42	9.60	23.05	1.80	23.05	21.25	0.84	5.34
7	01/02/99 10/26/13	10.89, 0.00, 0.00, 7.62, 0.00, 7.35, 18.36, 6.48	0.33	9.28	9.59	23.48	1.59	23.48	21.89	0.85	5.44
8	01/02/99 10/26/13	10.74, 0.00, 0.00, 7.52, 0.00, 7.31, 17.72, 6.41	0.38	9.27	9.59	23.40	1.47	23.40	21.93	0.86	5.46
9	01/02/99 10/26/13	10.54, 0.00, 0.00, 7.49, 0.00, 7.73, 18.47, 6.19	0.31	9.50	9.61	23.68	1.65	23.68	22.03	0.85	5.44
10	01/02/99 10/26/13	10.52, 0.00, 0.00, 7.63, 0.00, 7.10, 17.64, 6.59	0.39	9.44	9.58	23.43	1.56	23.43	21.87	0.86	5.46

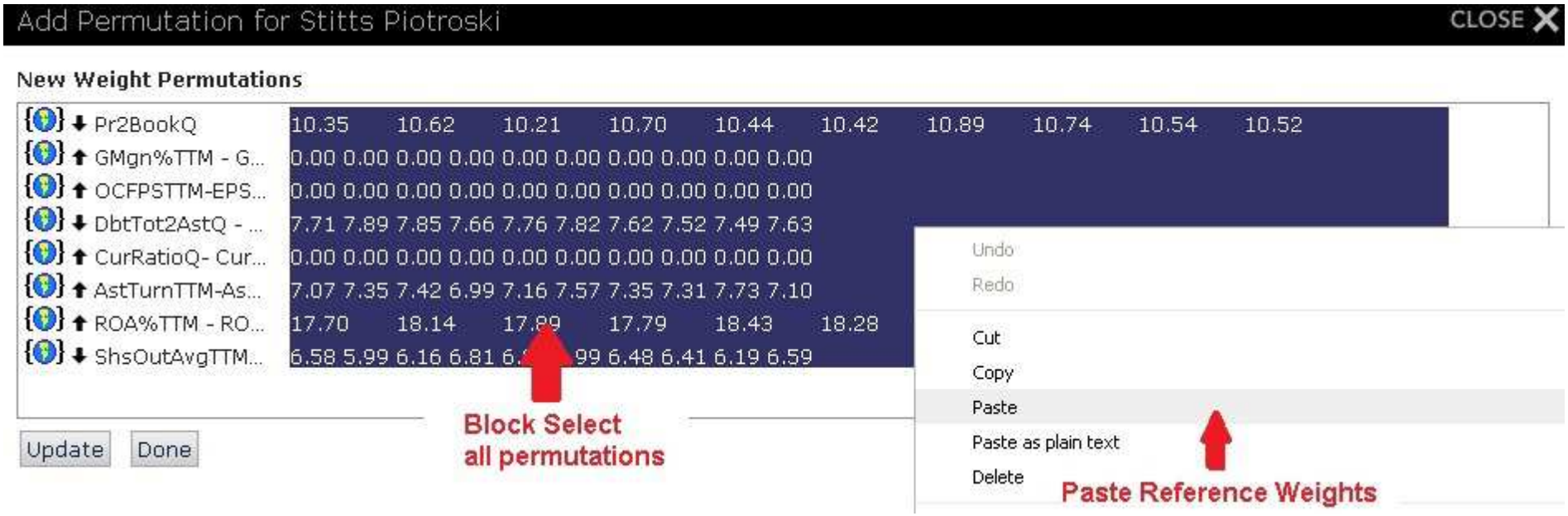
A window will pop-up where you enter the name of the ranking system. Once this has been done then click on Save. You can then close the window.

Generate the Final Ranking System

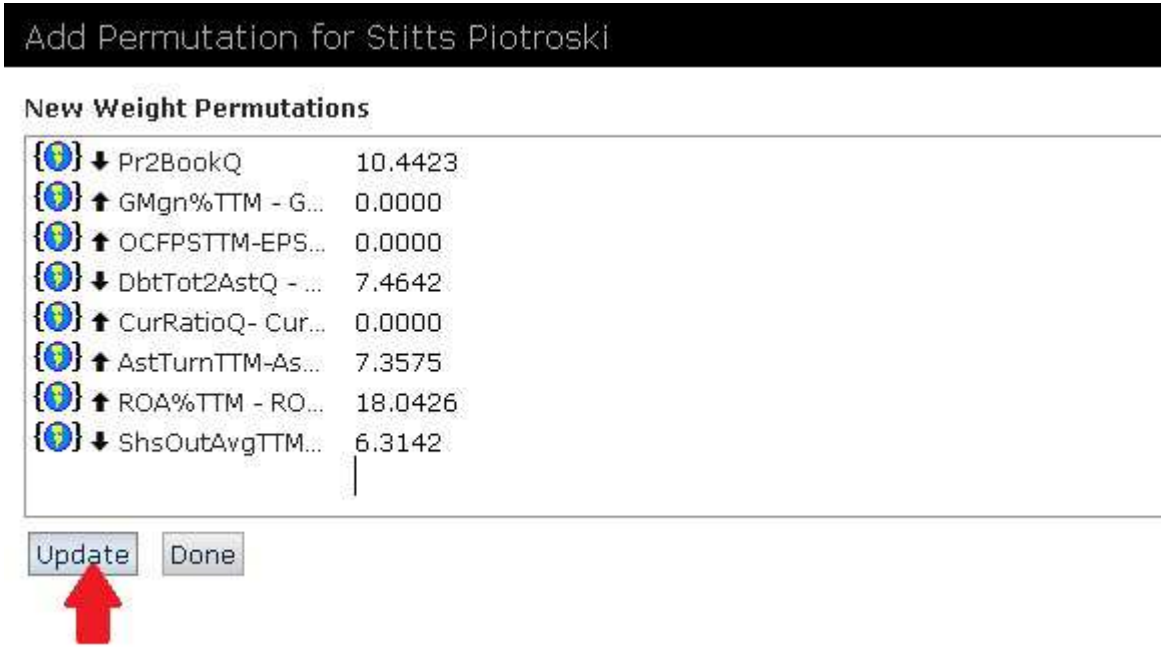
After all ten ranking systems are saved then the final ranking system will be created. Start by block selecting the reference weights, then copying them.



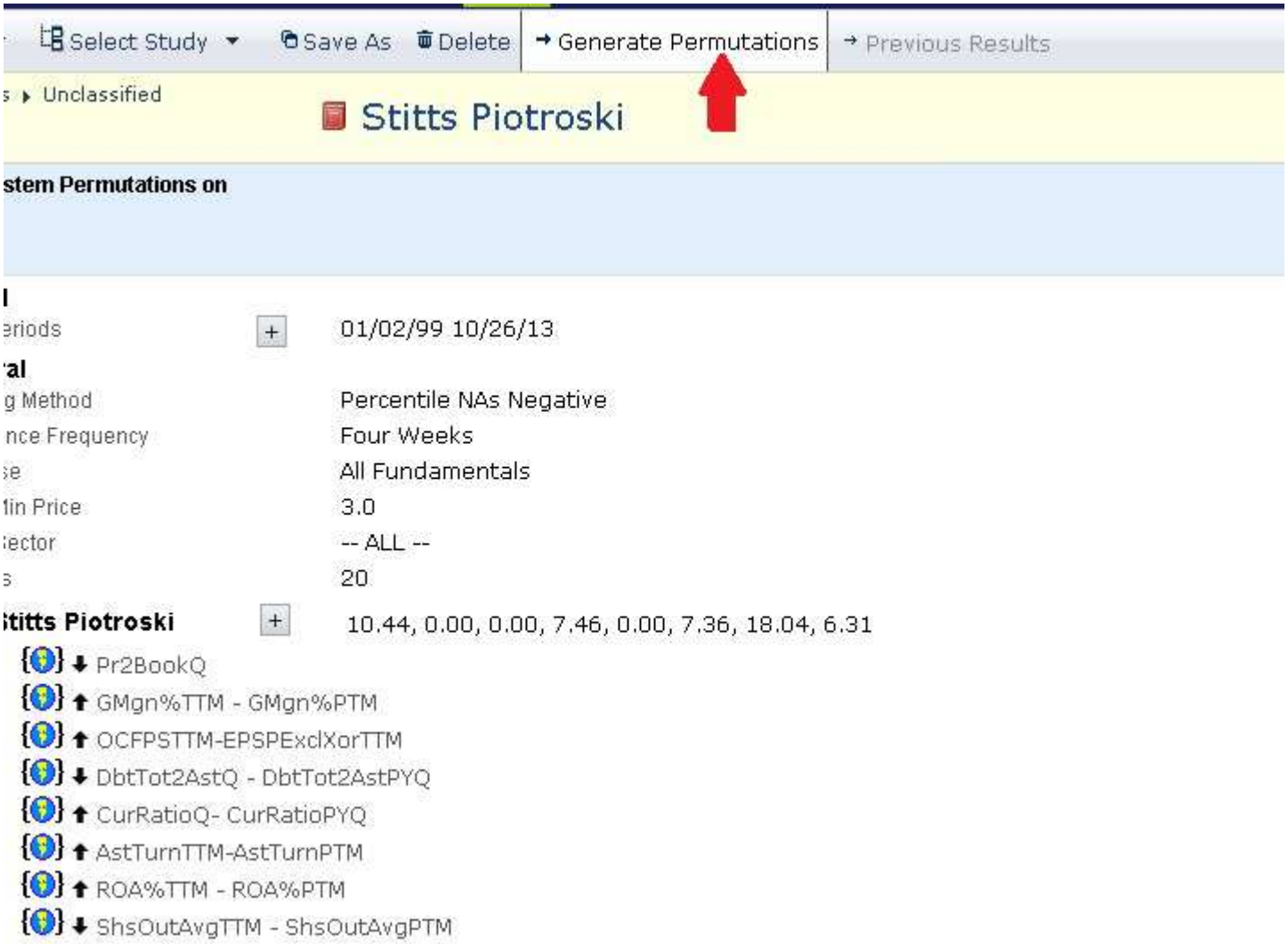
Now block select the previous permutations in the Edit Permutations screen in the ranking system optimizer.



Click on the Update button to save the permutations.



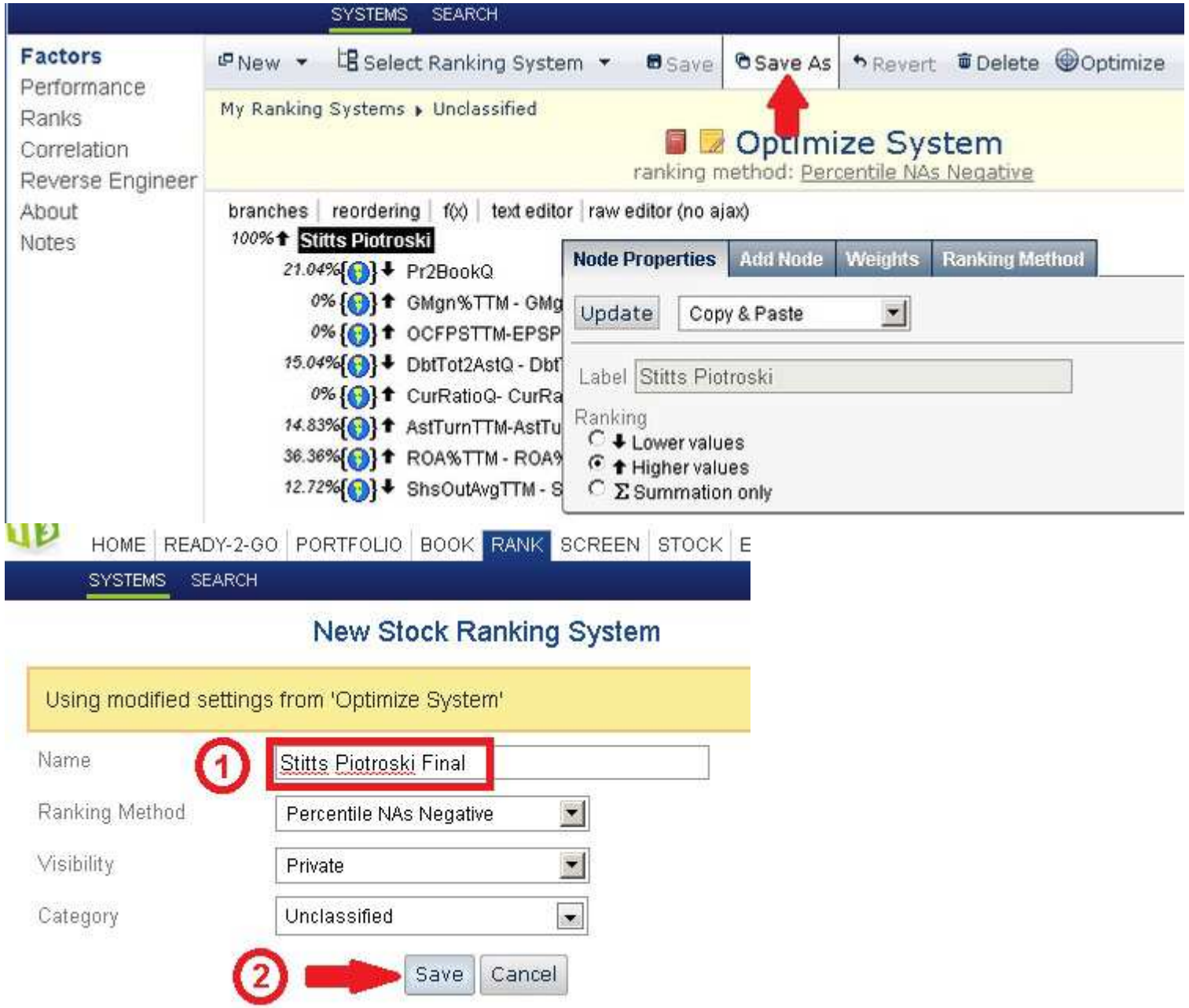
Click on the Generate Permutations menu item.



Run the ranking system optimizer.

Once the optimizer has finished then click on permutation #1 as shown below. Note: there should only be one permutation.

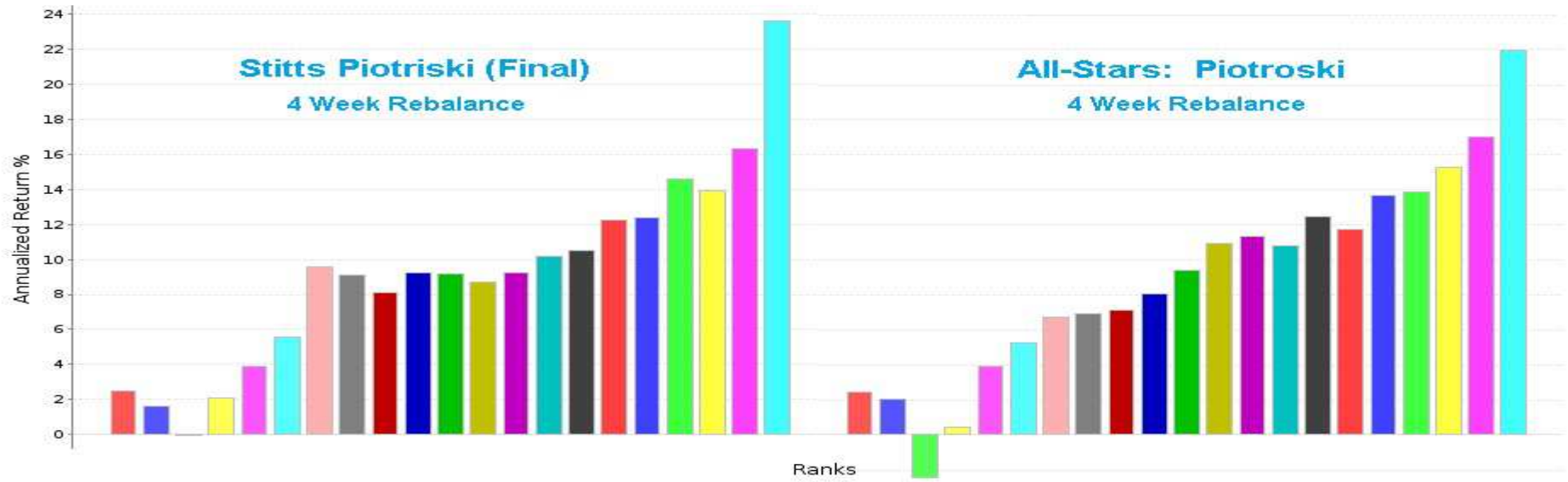
Select **Save As** to rename the final ranking system.

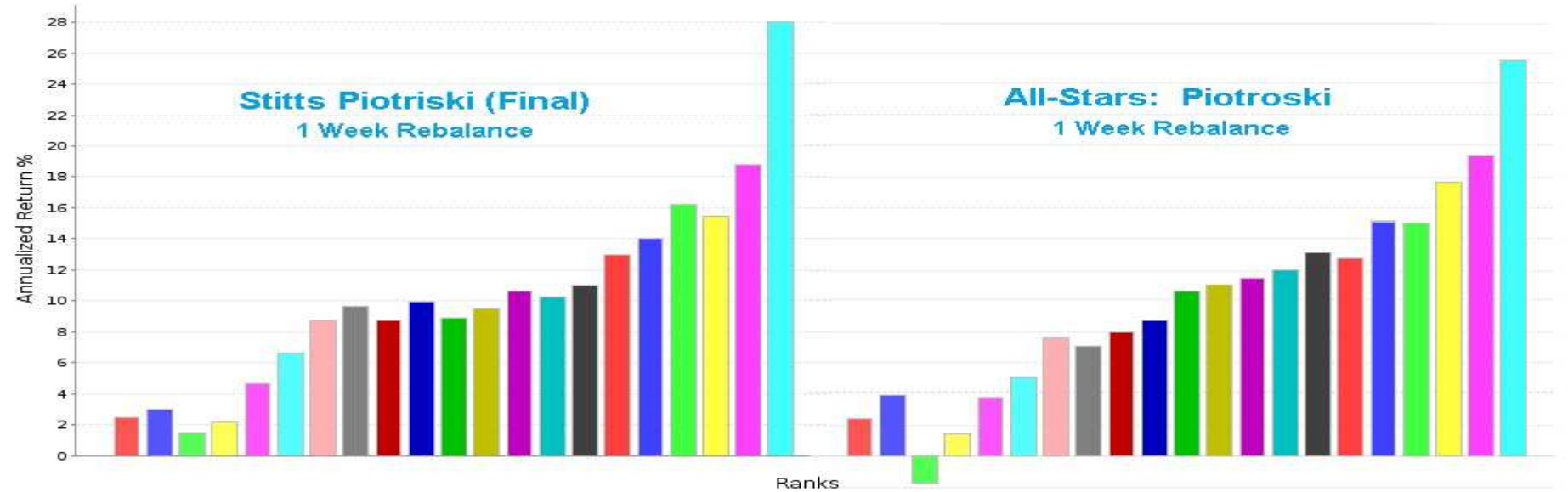


Check to make sure you have all eleven (11) ranking systems as shown below.

Performance Check

3 of the 8 stock factors have been deleted from the original All Stars: Piotroski. The performance in the last bucket on the right has increased from 22% to 23.7% for 4 week rebalance. This doesn't sound like much but consider that the last bucket holds approximately 150 stocks and that the original ranking system was already quite an optimal configuration. The performance for weekly rebalance also improved.


Stitts Piotriski (Final)
4 Week Rebalance
All-Stars: Piotroski
4 Week Rebalance
Annualized Return %
Ranks


Stitts Piotriski (Final)
1 Week Rebalance
All-Stars: Piotroski
1 Week Rebalance
Annualized Return %
Ranks