HEGEMAN'S AVERAGE

JOHN S. FARRELL

INTRODUCTION

This text will explain *Hegeman's Average*, a system for averaging and weighting stock analyses based on credibility and relevancy to calculate a normalized score for a stock's fundamentals. Hegeman's Average, along with any other calculation in this paper, are only a research experiment and shouldn't be applied beyond that fact. This project was only to play with data collection and manipulation, not to create a financial advisory resource, hence, don't use it as such. With that being said, the simplistic idea is a calculated weighted average of past analyses for the fundamental backing of a given stock. The weight falls on the credibility of a source's analysis among other select factors, and the values being averaged are normalized stock gradings. This process is broken down into a few pieces covered in this paper: static analyzer credibility, scraping and parsing, score conversion, weighting, and averaging.

1. FORMULAS

Throughout this paper, the following formulas will be referenced. Some are recognizable, but a few are ones that have been designed for this experiment. The first formula here is *ANGR Weight Formula (W)*:

$$W(a, n, g, r) = \frac{C(a,g) \cdot ln(n+e)}{r+1}$$

Date: November 19, 2022

This formula is the basis for weighting analyzers and takes in four variables: *a,* normalized accuracy of past analysis, ranging from 0 to 1; *n*, number of analysts for a given analysis, $n > 0$; *g*, significance of an analysis predicated on Google search ranking, where $g = 0$ is a first page ranking, and hence, $g > 0$; and *r*, relevancy of an analysis on the basis of how recent the analysis took place in months, with $r = 0$ being today. *W* will be discussed in more detail in section 5.

As you may have noticed, *W* has a sub formula, *C* that takes *a* and *g* as inputs. That is the *Credibility Formula (C)*[2]:

$$C(a, g) = \frac{a}{ln(g+e)}$$

The variables in *C* represent the same as in *W* above, and will be explained thoroughly in section 5. These two formulas make up the majority of weighting, but there is an addition here for weighting an analysis among other analyses of one given author:

$$W_2(n, r) = \frac{ln(n+e)}{r+1}$$

Variables are the same as in *W*. This formula will be discussed briefly in section 4. It should be noted that W_2 is an abridged version of *W*, without *C*. This is logical when taking into account W_2's use case. For calculating the weight of one analysis among many of one author, there's no need to take into account the *credibility* of the author, because all analyses will have this same value. Both *W* and W_2 are used as the weight for calculated weighted averages.

The rest of the formulas below are more concrete and recognized. They are not necessary for the understanding of this experiment but I've included them here for clarity sake. First, the *Normalization Formula (N)*:

$$N = \frac{(v-min)}{(max-min)}$$

This formula will be used directly in section 2, *Static Analyzer Credibility*, and indirectly when it comes closer to grading stock fundamentals. Simply for our use, *v* is a stock

[2] *C*'s separation from *W* will be discussed at the end of this section.

grading, *min* and *max* are the extremes among a plethora of gradings. More detail will be provided under section 2.

Next is the *Calculated Weighted Average Formula (A)*:

$$A = \frac{\sum_{0}^{k} w_i * v_i}{\sum_{0}^{k} w_i}$$

In this case, the calculated weighted average is determined from a list of *k*, values of *v* are weighted by values of *w*. The determinants of *v* and *w* will differ among the uses through this paper, more specifically in sections 4 and 7.

You may be wondering where the *Hegeman's Average Formula* is, but it's simply a calculated weighted average as in *A*. And while technically a combination of the formulas in this section could produce something of a single formula, *Hegeman's Average* is emphasized to be more of a process than a single formula. The system can be broken into smaller pieces to be more digestible, and use smaller formulas to come to a conclusion. Another note before moving on: It should be said, C and W may be combined and simplified to a cleaner equation. When displacing domain issues, it can be simplified all the way to a log function where the base is g^r and inner is n^a:

$$W(a, n, g, r) = \log_{g^r}\left(n^a\right)$$

This is included here because of its mathematically elegant solution. However, it doesn't work well in the context of this experiment due to the experiment's modularization and programming components. This will be discussed and emphasized further throughout the paper.

1. Static Analyzer Credibility

In this section, the process and reasoning behind calculating credibility of an analyzer's analysis will be described, as well as why it's been labeled *static*.

To start, it's important to understand the use of analyzer credibility. It falls under the criteria for the *ANGR Weight Formula*, as *a* and *g*. When it comes to weighting a bunch of analyses from many sources, it's important to weigh both relevancy *and* credibility for a well structured prompt. While one analysis made today may be very relevant, its source

may be not very credible and give inaccurate reports. More about relevancy will be discussed in section 5. Credibility is calculated with two variables: *a*, accuracy; *g*, significance. Highest credibility is given to values of higher *a* and lower *g* - lower g correspond to greater significance as discussed next.

To start with the simple piece, *g* is a static variable, Google's page ranking. Sources like *CNN*[3] or *WSJ* arrives first upon Google searches for stock rankings and gradings, and are given a value of 1. Sources like *SimplyWall* are much further down in the rankings and receive a higher *g* value. Front page analysts have earned a record of credibility and accuracy, which have indirectly made them popular among many. Along with this, there's a competitive drive and financial business goal to make the first page of Google. Analysts want to provide accurate ratings to make their way to the first page. While popularity is important and a positive the credibility of an analysis, it shouldn't be a deal breaker for pages that can't make the first page, hence *C*'s ln(*g*) parameter.

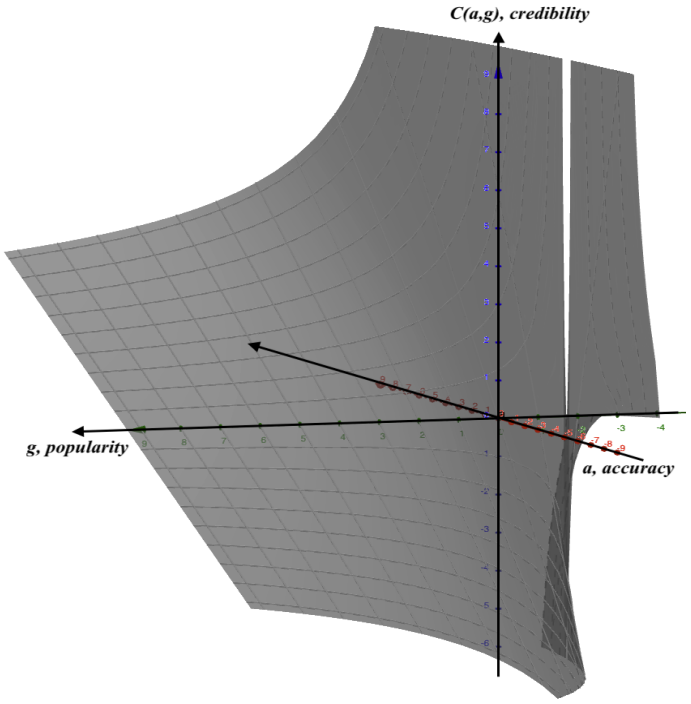
The more difficult calculation comes with accuracy analysis. The accuracy for an analyst is determined in a prolonged calculation that is both time consuming but clearly necessary to weigh a stock analyzer's significance. Here are the basic steps:

1. Stock Collection: Gather a plethora of stocks that follow the goal trend for a stock, a distinct increase in value of around a four week period. This four week period should be approximately through October, as it should line up with the quarterly earnings. This piece may be overlooked, but many stock analyses fall on a three month clock, lining up with quarterly earnings reports. These earnings reports generally take place June, September, December, and March. While it doesn't matter which is followed up, this analysis was done in November, meaning the three month analyses were from September and used to predict through December. In summary, picking stocks that would have been a top score grade during the month of October.
2. Data Collection: For each analyzer, record the sum of their 3 month converted raw scores. More on the specifics of score conversion is discussed in section 4, but are generally converted onto a five point scale, where one is labeled a 'buy' and five is labeled a 'sell'.

[3] No sources analyzed in this experiment are named in this article, including *CNN, WSJ,* and *Simply Wall*. Here they are named as examples of popularity, not because they were used during analysis.

3. Normalize: The final step is to take each analyzer's sum from step two and normalize them using the *Normalization Formula, N,* in section 1. These normalized values are the *a*, accuracy, for the given stock analyzers.

This accuracy piece should hold a higher significance than *g*, as it isn't *completely* necessary for an analyzer to have top priority to be the most accurate. Below is a graph displaying the relationship of *a* and *g* for calculating *C*. Again, a lower *g* value dominates a higher *g* value.



[4]

Credibility and popularity are *static*, because they fall under the fundamental variables of *W* that don't get repurposed on every Hegeman's Average. While credibility should be periodically recalculated, that process should happen at most every three months. Unlike values of relevancy, credibility values are held in cold storage in either the database or directly in a *.py* file, as there's no volatility in them over a period longer than a month. Relevancy values change not only per stock, but for some stocks, *per day*, depending on the frequency of analysis. So, while some of these relevancy values are stored temporarily for a handful of weeks, they aren't static like credibility.

[4] Graphs in this paper are meant to give a rough visualization of the relationship between values and their significance. E.g. Here, it should be clear that accuracy is more important than popularity of a site, while both still contribute credibility. Furthermore, *g* is a positive integer, which is not conveyed by this graph.

2. Scraping and Parsing

The most important part of this experiment was collecting data. To do this, data was collected from two sources: public APIs and public webpages. APIs were used to gather concrete data such as tickers on the U.S. exchange and their price history. Web pages were scraped for details regarding analysis of a given stock. The significant data to this project was from parsing, as that contained the analyses, but the APIs were necessary to even make use and gather that data. The process for gathering analysis data went as follows:

1. Gather list of common stock tickers through a given API (*Yahoo Finance's API* or *StockSymbol* are a few options).
2. Use another API for URL requesting (for Python 3, *Requests: HTTP for Humans*) and supply a user agent header.
3. Once you gather a response, parse the response to receive a readable HTML body (possible options include *BeautifulSoup* and *Selenium*).
4. Specifically parse the HTML document for wanted data (this step differs depending on your sources).
5. Package this data in an output that can be applied between different parsers.

It is *very* important to be cautious and mindful when parsing webpages. Besides possibly being against a website's terms of service, the traffic of scrapers can be costly to site owners and an overall waste of computing when not done carefully. In the data collection for this project, great precaution was taken in scraping. Websites disallowing scraping were neglected and all other data was stored directly into a cloud database so as to not have to query the page more than once, and followed basic guidelines of *robots.txt*. With that being said, the specific scrapers and sites used here haven't been published, which is why there is great detail above on how the scrapers functioned. It's unnecessary to see much beyond the direction above, and ethically isn't right to publish scrapers to specific sites, or the sites that were scraped at all in order to protect from possible harm of unjust users.

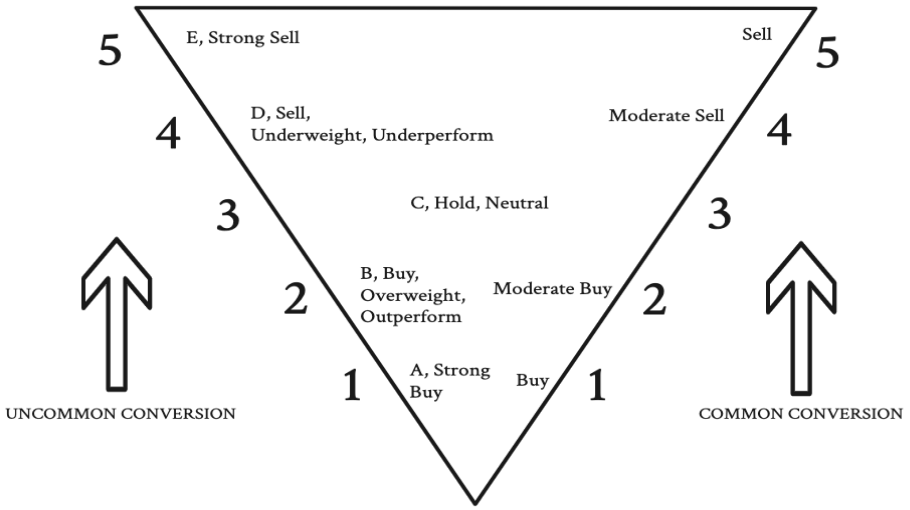
The important take away from this section is the use of APIs and scrapers to funnel specific data into a given database. And, again to save efficiency, data stored in the database is first processed, as to not do costly conversions, weighting, and averaging on after every query. Sections 4 and 5 will cover how the concrete and parse data is processed before being saved to the database.

3. Score Conversion

Stock analysis is often on an integer scale, 1 to 5. 1 represents a buy and 5 represents a sell. While many online analyses follow this scale, not all do, and must be converted. To do so a general conversion must be created, as shown in the table below:

Common Ratings Chart				
5	4	3	2	1
SELL	**UNDER PERFORM**	**HOLD**	**OUTPERFORM**	**BUY**
	Underweight		*Overweight*	
Strong Sell	Moderate Sell	Neutral	Moderate Buy	Strong Buy

While this chart does a great job converting most scores, some do pass through. For example, sites that grade on a letter scale or those that grade by percentile don't fit the table well. And another issue, some sites label sell as what would be a 4 instead of moderate sell, and label strong sell as a 5. These may also cause mislabeling of a 4 to be a 5, and skew the conversion. Below is an updated chart to deal with some of these problems.



Stock analyzers would either be labeled to use an uncommon conversion or common conversion. Common conversion followed a traditional[5] pattern, including if the analysis was already on a 1 to 5 scale. Uncommon includes all non-tradition analysis, including numbers not 1 to 5. All analyses that were on a greater scale were divided down with simple arithmetic.

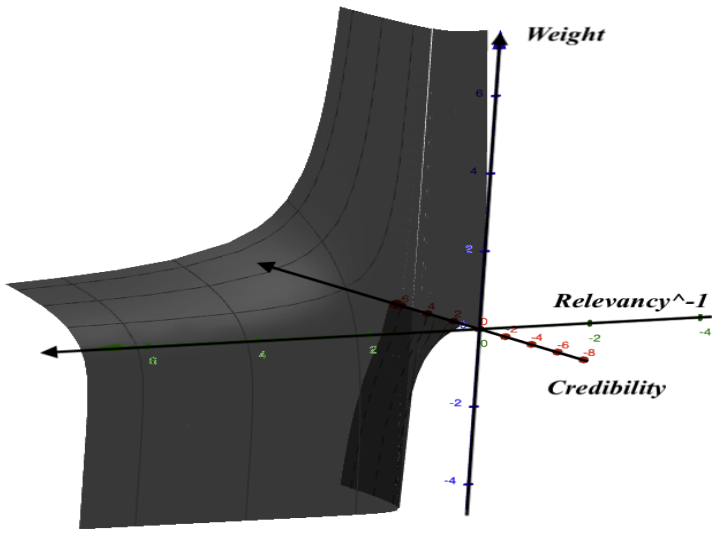
The last piece left out is for when encountering a page with multiple analyses of the same stock. It's common for a website to hold current, 1 month ago, 3 month ago, and

[5] This of course depends on the definition of tradition, but in this case are the common conversions according to the given picture above.

other analyses of a stock. Without accounting for this, it's possible for some analyzers to be weighed for triple the number of analyses or more, depending on the number of different analysis relevance. In order to offput this issue, a calculated weighted average is taken, where the weight follows W_2. This formula simply calculates a weight for one analyzer's analysis by compounding the number of analyzers by relevancy of the stop, with relevancy dominating the number of analyzers. The weighted average of analyses are returned for that analyzer's analysis on a given stock.

4. WEIGHTING AND AVERAGING

Once all the data for a given stock is collected, it's time to determine the weighted average, or the *Hegeman's Average*. Each analyzer must first be given a weight. To do this we use the *ANGR Weight Formula, W* from above. This formula takes in four variables which have each been discussed throughout this paper: accuracy, number of analyzers, Google's ranking, and relevancy. These variables are divided into two categories: credibility and relevancy. Accuracy, which discussed is a normalized value of how accurate past analysis has been, and Google search rank make up how credible a source is. The number of analyzers put into one analysis and when the analysis was published make up how relevant the analysis is. The relationship between these two categories is expressed in *W*, which's relationship is graphed:



[6]

It is quite difficult to give a full picture of what this function's graph will look like, as there's four input variables, meaning this function would have to be graphed to the fifth degree. The main take away is the logarithmic relationship, where relevancy of a month

[6] Again, this graph is only a mere visualization of a given relationship.

ago is much more significant than one two months ago. But an analysis two months ago will not be quite as significant compared to an analysis five months ago. Even though there's still a two month time window, it's less important that an analysis is three or five months ago vs one or three months ago. A similar relationship idea can be taken away inversely for analysis. As credibility increases, the difference in accuracy becomes less important, as the main idea is that the *source is credible*. While extremely low credibility is important to result in a lower weight. Once each analysis has a weight, a calculated weighted average is taken, where the weights are calculated with W, and the values are the raw converted scores from section 4. This output average *is* the Hegeman Average.

5. Conclusions

Creating and applying *Hegeman's Average* was a fun experiment in data manipulation. Again, there should be an emphasis that this was *only* an experiment. I am not registered as a security broker or an investment adviser, nor have I been educated in the stock market past simple technical signals and fundamental analyses. But, there is so much data around stocks that it often makes it a great option for educating one around data sourcing, gathering, and manipulating. Working with stock APIs, web parses to gather stock analysis, and creating grade conversions all are applicable skills far beyond economics. Along with learning how to responsibly parse sites and efficiently store necessary data to a database, working with stocks is an educational experience. Nevertheless, it's vital to not get carried away with complex calculations as complex don't always entail a correct application.

Formulated Summary

Variables	Description
w	Weight of a given analyzer. Derived from ANGR Weight formula.
r	Analyzer's converted raw score on a stock. Between 1 and 5.

Hegeman's Average

$$H(w_i,r_i) = SUM(w_i{*}r_i)/SUM(w_i)$$

For *k* analyzers each with a weight of *w*, giving a stock rating *r*, *Hegeman's Average* is the calculated weighted average of *k* analyzers.

Variables	Description
n	Number of analysts. Greater number receives a higher weight.
A	Accuracy of past analysis. Scaled between 0 to 1. Using analysis algorithm on another tab.
r	How recent the analysis took place. For example, some sites hold multiple stock ratings over year, 6mo, 3mo, current. Lower r is favored. r is the approximate # of months since analysis
g	Page ranking on google. g is the page of Google ranking would be found on via basic Google search. Lower numbers are favored. **g > 0**
C	Credibility score is calulated from A and g. Score is from 0 to 1.
W	Significance of a source

ANGR Weight (W) Formula

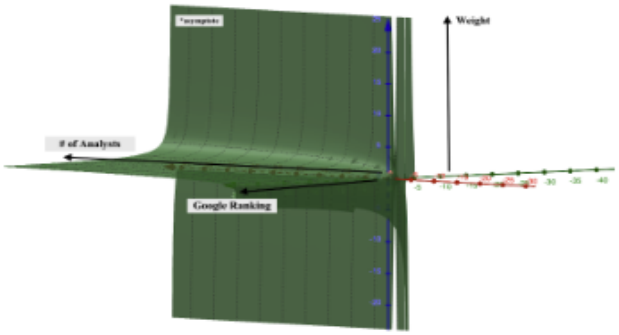
$$W(A,n,g,r) = [C(A,g)\cdot \ln(n+e)]/(r+1)$$

ANGR Weight (W) Formula: weight and signficance calculation of one analyzer's opinion on a stock. There are a few criteria: number of analysts, credibility (accuracy of past analysis and Google ranking), and relevence (age of analysis). Weights don't have a scale, as they are used in as the weight for a weighted average in further analysis (CAS).

Credibility (C) Formula

$$C(A,g) = A/\ln(g+e)$$

* Update to W formula, r -> r + 1 because analysis labeled as current has a value of 0



Variables	Description
n	Number of analyzers
r	Relevancy
w	ln(n+e)/(r+1)
v	1. Analyzer's grade, 2. n, 3. r Other variables such (e.g. Google Ranking) don't need to be changed.

Cumulative Average Score (CAS, A) Formula

$$A(w_i, v_i) = SUM(w_i * v_i) / SUM(w_i)$$

CAS is just a simple calculated weighted average.

The CAS formula is used for determining the average of one analyzer's scoring, in event they have multiple analysize on their page. The weight of a given analysis is the ln(n)/(r+1), similar to the ANGR formula but not exact. Then, each weight and its corresponding analysis added in a simple calculated weighted average for each v. The output values are the single variable inputs, allowed to now be applied in the Hegeman's Average and ANGR weight formula.

Weight

$$w = \ln(n+e)/(r+1)$$

Variables	Description
v	An analyzer's average raw score
min	Minimum of all analyzer's average raw scores *each analyzer has 1 average raw score as described below, so the minimum means the minimum of all analyzers
max	Maximum of all analyzer's average raw scores

Normalization (N) Formula (for accuracy)

$$N(v_i) = (v_i - min)/(max - min)$$

Calculating analyzer's accuracy. Accuracy is generally a '1 time' calculation.

1. Collect stocks that follow the pattern we are look to find: a general increase over around a 4 week period. In order to test the accuracy of analyzers, this for week increase should have around August (generally, analyzers release there analyses after quartley reports, hence the common 3 month relevancy). Among samples, there should be varying increase and decreases of the stock before and after this 4 week period.

2. Look at each stock's 3 month ago analysis and record the converted raw score.

3. Average these raw scores for each analyzer

4. Normalize average against other analyzers using the given normalization formula.

To summarize, pick stocks will have what we are looking for as a top level graded stock. Calculate how well each analyzer predicted these ticker's success. Normalize calculations.