Quantitative Equity Analysis:

A Primer on Valuation Models

And Backtesting

TEAM Systems

Introduction: The Quantitative Approach to Investing

Active equity investment managers often employ valuation models to accomplish their goals. That is, they attempt to come up with proprietary ways of seeing if stocks are over- or under-valued, using historical and expectational data sources. These models might be dividend discount models, factor models, or something else. They are tools used to forecast stock returns.

Equity investing today seems to be going more and more by the numbers—the active managers are using more mathematical techniques to get an edge on the competition, and are sometimes referred to as “quants”. If a manager employs quantitative techniques to select stocks, what does that mean? There seems to be no concise definition; quantitative techniques might involve everything from simply screening a database for companies with a low P/E ratio to constructing and statistically backtesting a multi-factor valuation model. This document was written to review some of the techniques used by those that build and backtest their own proprietary multi-factor models, and some of their concerns.

Stock Factors

It all starts with an idea. A quantitative manager wants to convert the idea into some sort of proprietary or unique computation, or a combination of such computations. These computations often involve company-level data such as market capitalization, P/E ratio, book-to-price ratio, revisions in earnings projections, etc. Quantitative managers assume these characteristics, or “factors”, of stocks are associated with stock returns, and that certain combinations of factors can outperform. When several factors are combined in a strategy or model, we have what is known as a “multi-factor model”. The hard part is coming up with the right combination of factors, and how to weight their relative importance (i.e., predicting factor returns). This is usually a trial-and-error process.

What factors might contribute to excess returns? Many factors have been studied, but managers often gravitate toward “value” factors like Book to Price, Price to Sales, and P/E, or toward “growth” factors like revisions in earnings estimates or earnings per share growth rates. “Momentum” and “size” factors are also popular. Managers look at factor returns over time to see what strategies worked best during certain market periods. This knowledge can help in their attempts to forecast future returns.

Statistical Techniques Used in Model-Building

Quantitative managers often use some statistical techniques to render their data more meaningful and make it easier to work with when creating multi-factor models. We will review a few techniques here:

- *Correlation Analysis* – before combining factors in a model, the manager might construct a “correlation matrix” to see the degree to which the selected factors are related to each other. Factors that are highly correlated with one another are redundant, and so are generally not combined in a model. A model with several uncorrelated factors is often a much more powerful and efficient predictor of returns. Correlation coefficients always lie between –1 and +1, so it is easy to quickly eyeball the strength of the relationship.
- *Variance/Covariance Analysis* – one can also construct a variance/covariance matrix to see another measure of association between variables, but this matrix also shows the variance, or the typical variability (from the mean) of the variables themselves. It helps one spot a distribution where “outliers” might seriously affect analysis results.
- *Demeaning* – demeaning a factor is a way of removing a “bias” from that factor, and is accomplished by subtracting the average value from the observed value. For example, to demean book/price by industry or sector, subtract the average book/price for the industry or sector from each company’s book/price figure. This reduces the industry’s or sector’s bias, and makes companies from different industries or sectors more comparable in analysis. This is quite an important step in model construction, since book/price for a high-tech firm will differ significantly differ from that of an electric utility, for example.
- *Standardization* – when combining factors into a model, it is useful to measure the factors in the same terms, or on the same scale. Standardization accomplishes this by rescaling the data distribution so that it has a specific mean and standard deviation (usually 0 and 1, respectively). Once a sample has been standardized, it is easy to determine a number’s relative position in that sample. To standardize a factor, the mean of the sample is subtracted from an observation, and the resulting difference is divided by the standard deviation.
- *Handling Outliers* – outliers, or extreme values in a distribution, can sometimes unduly influence the results of an analysis. One way to reduce the distortion caused by outliers (other than to delete them from the sample) is to “Winsorize” the outliers, or to pull them in toward the center of the distribution. A limit is set (often 3 standard deviations from the mean) and the outliers are pulled in to that limit with an iterative process.
- *Regression analysis* – quantitative managers employ regression analysis to uncover predictive relationships; they want to know the value of Y, given the value of X. Regression analysis can uncover linear relationships between X and Y, or between a P/E ratio and excess returns, so it can be helpful in devising a strategy that might outperform a benchmark (and the competition).

The Data Sources

Quantitative managers subscribe to various electronic databases and use some of the above techniques to add value to their analysis. The data are typically a combination of company fundamentals, stock prices, earnings estimates, macroeconomic data, etc. Some of the more well-known databases include *Compustat*, *Disclosure*, *First Call*, *Ford*, *Global Vantage*, *IBES*, *Value Line* and *Worldscope*. These databases are typically available from several vendors in a variety of formats (as well as direct from the database creator), so there are usually several ways to get one's hands on the historical data necessary for model testing.

When selecting a data source, keep in mind that your selection itself will have an impact on your analysis results. If you ran a backtest with historical data from *Compustat* and then used the same assumptions in a backtest utilizing data from *Ford* or *Value Line*, you might arrive at some different conclusions. Some databases (like *Ford* and *Value Line*) offer "snapshots" of the data that was known at a particular point in time; others (like *Compustat*) restate historical data to make companies more comparable to each other and over time. They do so because companies do not share the same reporting methods in 10-K reports, quarterly reports, etc.

You might ask a few other questions when evaluating a data source:

- Where do the ticker symbols come from? You will find that some companies have different tickers in different databases. It can be helpful if the database also contains a permanent ID for each company that will never change even if the ticker and CUSIP number does. Some data distributors offer this in their software.
- What happens to a company's data history when it is acquired by another firm, or when it stops reporting to the SEC for some other reason? Some databases carry this history in the same file as the active companies; others segregate it into a separate file.
- Will you be able to work with both split-adjusted data and data that is unadjusted for stock splits?

Testing a Model

Testing the valuation model usually involves applying a strategy to some historical data set to evaluate the model's performance over a particular time frame. Testing can include the use of *ordinary least squares* (OLS) regressions, if the model is linear, as well as full-fledged "backtesting" (portfolio simulation), where one simulates a trading strategy over time and takes transaction costs into account.

When statistically testing the performance of a multi-factor model via OLS, we want to see if the model factors were predictive of a dependent variable, usually a return figure. To discern the strength of the relationship, quantitative managers look at a group of statistics produced by the OLS regression:

- *Regression coefficient* – measures the relationship between each independent variable and the dependent variable.
- *R-squared* – measures the overall fit of the model; an R-squared of .25 tells us that the model explains 25 percent of the variation in the dependent variable. As more independent variables are added to the regression, the R-squared tends to overestimate fit. For this reason, analysts also look at the "adjusted" R-squared.
- *Adjusted R-squared* – also measures the fit of a model, but imposes a penalty for the use of additional independent variables in a regression. This prevents one from simply adding more independent variables to generate a higher R-squared. It is especially useful when comparing the results of several regression analyses that have different numbers of independent variables.
- *t-test* – measures the significance of the independent variable(s); a typical rule of thumb states that if the t-statistic is greater than +2 or less than –2, then the variable is significant at the 95% confidence level.
- *F-test* – measures the significance of the model as a whole; used when there are multiple independent variables in the regression.
- *Sum of the Squares of the Errors* – a measurement of the variability left unexplained by the regression; it measures the dispersion of the dependent variable about the regression line.

In sum, the above statistics (and others) are generally studied to see the level of predictive power a particular model might exhibit. We want to see if a strategy adds value, or has more than a random contribution to excess returns.

Assuming each factor in the model has exhibited value added, a composite score or alpha (expected return) is generated. Each factor's weight or percentage of the composite can be static or dynamic. A simple example of a static weight is to equally weight each factor in the model; when you use the model to forecast returns, each factor is given the same level of importance regardless of its historical variations. Some managers prefer to give more weight to those factors that have historically exhibited larger regression coefficients and will therefore not weight equally. Yet others prefer a more dynamic approach. They may look at factor regression coefficients every month, week, or even every day to incorporate the latest information content into their forecasts.

If the model or strategy appears to add value, then the next step might be to understand what sort of returns it has generated over time in a more "real world" environment. That is the purpose of backtesting, or portfolio simulation. Here the strategy is simulated over time using historical data, and the returns generated by the strategy are compared to some benchmark. The simulation not only uses the model to select stocks; it also makes assumptions regarding transaction costs, how many dollars will be invested, rebalancing frequency, etc.

Some of the more common concerns in portfolio simulation include:

- Historical data decision – what database or databases does one use? There are several to choose from, but keep a couple things in mind. First, you want a database that offers at least some data shown as "originally reported". Some databases restate the data historically to reflect an acquisition or some other event; you should not be using these restated numbers since they were not known at the time a trade could have been made. Second, your data should include history on companies that have gone out of business, been acquired, or have stopped reporting to the SEC for some other reason. If you exclude these companies from an historical backtest, then you are including only the surviving companies, which may not be representative of the investment universe. This "survivorship bias" can lead to misleading results. Another form of bias that concerns equity managers is "look-ahead" bias. A database may offer historical earnings figures, but when were these figures actually released to the media? If a company reports earnings late, and if your backtest model relies on an EPS figure, then you have "looked ahead" to the future and used a figure that was not really available at rebalancing time. This is where "lagging" data becomes important; e.g., when computing P/E at the end of a year, you won't use the earnings figure for that year as the denominator. You would use a previous one that was known at the time.
- The time range – some feel that a strategy should be tested over 30 or more years to really see its power through various market cycles, but one must also be practical. Not every organization can afford to purchase 30 years of data, and fewer data items were collected 30 years ago than are collected now by database vendors. Significant value can still be obtained from a shorter simulation, provided there are more frequent observations (e.g., daily).
- The ranking variable – this is the factor, alpha or composite score you use to rank the companies, usually sorted from highest to lowest. It also might be used to organize the companies into deciles or quintiles.
- Company population – what companies will you analyze? Will you run the simulation over an entire database population, or specific subsets of companies such as high caps, low caps, specific sectors, etc? What sorts of company screens will you set up that will be exercised at the beginning of each rebalancing period?
- The benchmark – for comparison of returns; this is usually an index or a specific group of stocks.
- Rebalancing frequency – how often will the strategy be applied (daily, monthly, etc.)?
- Reporting returns – how often should you compute the returns generated by the strategy?—usually for each discrete period (day, month, quarter, etc.) as well as on a cumulative basis (from the beginning of the test period to the end). Returns should be computed net of transaction costs, of course.
- Buy/Sell/Hold rules – let's call these the trading thresholds. Here you might take a couple of different approaches. One might be called the ranking approach. Simply rank a group of stocks from highest to lowest based on a selected variable or score (e.g., alpha), then simulate buys and sells that are dependent on where a stock falls in that ranking. For example, you might

simulate a buy if a stock's value for the ranking variable falls into the top decile of the distribution, a hold if it stays above the 70^{th} percentile, and a sell if it falls below the 70^{th} percentile. This ranking would of course be performed at the beginning of each rebalancing period. Another approach might use more specific buy/sell rules. For example: Buy the stocks in the Technology sector that have a Book/Price ratio greater than some absolute value OR that are new to a purchased buy list, and sell the stocks whose price falls below the group's average price AND whose market value is above the median for the group.

- Transaction cost assumptions – transaction costs are usually assumed to be the sum of brokerage commissions and the market impact of the trade itself. These assumptions vary widely, based on the orientation (large or small cap) of the manager. A small cap manager's backtest results could be severely skewed by an inappropriate market impact assumption.
- Handling missing data – sooner or later, if you are working with a database you will encounter missing data. What happens if 20 percent of your rank variable is missing during a given rebalancing period? You need some control here, perhaps just skipping over those periods with missing data and moving on to the next.
- Weighting purchases – when simulating a buy, you may want to weight purchases of stock by market capitalization or some other variable, or you may want to equal weight the purchases.
- Reviewing detail information – exactly what companies did your strategy buy, sell, or hold during a particular testing period?
- Turnover – how often does your strategy require you to turn over the average stock within the portfolio on an annual basis?
- Actual Costs – what was the dollar amount of the transaction costs, by period?

A comprehensive backtesting system should address each of the above concerns.

Summary

Some investment managers purchase rankings or composite scores from investment data or model vendors, and use them in selecting stocks. There are some good models out there, but one wonders how many managers might be investing based on the same model. Since a model's power tends to decrease as more and more managers trade with it, it seems only logical to want to devise a proprietary strategy. This is where a manager can get an edge.

Developing a proprietary, workable strategy is a trial-and-error process where one identifies and computes factors, combines them into a model, then statistically tests the model via regression analysis and further tests it via portfolio simulation. This is often time-consuming, and is precisely why so few managers really do it to a serious degree. However, more managers are moving into the quantitative realm, so it really helps to have the right software tools if you want to move in this direction. Ideally, your strategy-building and testing system should allow you to:

- Consolidate your favorite data sources into a single customized data repository, without requiring you to compose lines and lines of software code;
- Produce custom reports and export data to other tools (e.g., optimizers, spreadsheets, etc.);
- Screen your data to build lists of companies that exhibit certain characteristics;
- Quickly summarize your available data;
- Control for data outliers;
- Statistically adjust data distributions so they can be made comparable;
- Devise proprietary computations or factors, with the assistance of pre-defined mathematical and statistical functions;
- Check for correlation among your selected factors;
- Build a model, weighting your selected factors;
- Run repeated regressions with a minimum of setup chores;
- Run repeated backtests of your models with a minimum of setup chores;
- Put your models into production.