2017

Selection Portfolio: Applying Modern Portfolio Theory to Personnel Selection

Eric Leingang
Minnesota State University, Mankato

Selection Portfolio: Applying Modern Portfolio Theory to Personnel Selection

By

Eric Leingang

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Arts

In

Industrial/Organizational Psychology

Minnesota State University, Mankato

Mankato, Minnesota

May 2017

Selection Portfolio: Applying Modern Portfolio Theory to Personnel Selection

Eric Leingang

This thesis has been examined and approved by the following members of the student's committee.

____
Daniel Sachau, Ph.D., Advisor

____
Kristie Campana, Ph.D., Committee Member

____
Paul Schumann, Ph.D., Committee Member

5/5/17
Date

Contents

Abstract

Modern Portfolio Theory (MPT) is a framework for building a portfolio of risky assets such that the ratio of risk to return is minimized. While this theory has been used in the field of financial economics for over sixty years, the method has not yet been applied to compensatory personnel selection. A common method for personnel selection is multiple regression to maximize the predicted performance of the selected group given a cut-off score on the predictor(s). Recognizing that maximizing the performance of the selected group is not the only consideration, and that, for many jobs and organizations, the outcomes of false positives and false negatives can be drastically different in terms of costs, is central to this study. MPT is offered as an additional method for generating weights that produce fewer false positives than multiple regression.

MPT generates a set of all possible combinations of predictors within the plane of risk and return and finds an optimal set of weights on the efficient frontier, the hyperbola that represents the best possible set of trade-offs between risk and return. This study uses Monte Carlo simulations to estimate boundary conditions where MPT can outperform multiple regression. Comparisons are drawn between MPT, multiple regression, and unit weighting, applying weights uniformly across all predictors. Comparisons between the methods are drawn consistent with Signal Detection Theory, categorizing prediction-criterion pairs in terms of "correct selections," "false positives," "correct rejections," and "false negatives." Boundaries for suitability for initial sample size, applicant pool size, and cutoff score of the performance measure are explored. Finally, an application of MPT for reducing adverse impact and promoting diversity by choosing combinations of variables that reduce the weight given to cognitive ability is explored.

Background

Personnel selection is a systematic process for identifying and deciding which applicants to hire and which to reject. For over 2000 years, instruments have been developed to determine which applicants will make the best employees (Bowman, 1989). However, most familiar selection instruments and methods are products of the more recent past. Over the last 100 years, employee selection has been systematically and scientifically studied producing measures of job-related knowledge, skills, and abilities, and methods for assessing applicant suitability, ranging from interviews and online tests to numerous physical ability tests (Vinchur, 2007). Rather than limit selection procedures to only one type of predictor, for many jobs, multiple predictors of job performance are often used to make employment decisions. There a variety of ways multiple predictors can be combined to make personnel selection decisions. This study explores an adaptation of Modern Portfolio Theory (MPT), a method of choosing proportions of assets for an investment portfolio, as a potential method for choosing how to differentially weight combinations of predictors for personnel selection.

Combining Predictors

There are many ways to combine multiple predictors for personnel selection. Hurdles are often employed as a means of reducing cost. A hurdle is a method used to eliminate some applicants early in the process. This reduces the number of applicants who will require more costly methods further in the process. A compensatory system works differently. All predictors are assessed for each applicant, and then the predictors are combined for the purpose selection. In this way one or two predictors may compensate for a low score on a third. The two methods can produce different results. The hurdle method may eliminate an applicant early in the selection process while compensatory method might find that the applicant is qualified given a full set of predictor scores. Often these two methods are combined and there is somewhat of a range between a complete multiple hurdles system and a full compensatory system.

Within a compensatory system there are several common options for combining predictor scores. The first option is to use multiple regression. Multiple regression returns weights for each predictor that can be used to estimate a criterion. On face, weights produced from a regression are relatively straightforward. Multiple regression seems to weight the best predictors the highest, and the worst predictors the lowest. This is a bit of an oversimplification. If predictors are correlated, the shared variance between predictors can obfuscate the relative importance of one or more predictors. The primary attractions to multiple regression are that it maximizes the performance of the selected group, and that it has a large body of research and legal precedent supporting the method.

A second option is to use unit weighting. This means each predictor is given the same weight in the decision-making process. Unit weighting has several advantages: it is simple to perform, inexpensive, and easy to interpret. Another major advantage is that unit weighting is not estimated from data from an initial sample. In multiple regression, idiosyncrasies in the initial sample can have major effects on how well a model can predict a criterion, especially when the initial sample is small (Schmidt, 1971). Additionally, unit weighting does not use up degrees of freedom, so adding more predictors does not make the model more dependent on the quality of the initial sample. Unit weights also do not have any standard error, and they do not change how other variables predict. Schmidt found, in a Monte Carlo simulation, that when initial samples were lower than 100, unit weighting outperforms multiple regression in 3 out of 5 trials (1971). Einhorn and Hogarth also found unit weighting to be comparable to multiple regression in many real-world situations (1975). It is also easy to justify unit weighting predictors if there is no evidence for a differential weighting method. This is important if the legality of a selection system comes into question. On the other hand, if several predictors are measuring essentially the same thing, that is they are highly correlated, unit weighting tends to exaggerate the influence of the related construct. For example, if we have three separate measures of job knowledge that are all highly correlated and a fourth measure that is not related such as physical ability, job knowledge receives three times as much weight as physical ability. Unit weighting then performs best if the predictors are correlated to performance, but weakly correlated with each other. So, for unit weighting, *a priori* knowledge of the relationship between predictors and between predictors and performance is needed, which generally requires some kind of regression analysis.

Other than unit weighting and multiple regression, many other methods for mechanically weighting predictors have been used, and comparisons have been made between these methods to multiple regression and unit weighting (Trattner, 1963; Lawshe & Schucker, 1959). The most common finding of studies comparing these methods yielded no significant advantage of one method over another. Most legal cases surrounding employee selection, after the Civil Rights Act of 1964, defer to expert opinions in the matter of selection strategies, which are generally considered to be contained within three documents, the Uniform Guidelines on Employee Selection Procedures, SIOP Principles, and APA Standards for Educational and Psychological Testing (Uniform Guidelines on Employee Selection Procedures, 1979; Principles for the validation and use of personnel selection procedures, 2003; Standards for educational and psychological testing, 1999). These documents generally suggest multiple regression, and mention unit weighting as an alternative, which may explain, to a large degree, why other methods of weighting predictors are seldom used.

Another popular method is to avoid mechanically combining predictors and instead defer to human judgement. This method requires a person or persons to review the predictors and decide, using their own judgment rather than an algorithm. In doing so, the person makes some combination of weights without a prescribed method. The problem with this type of weighting is that people are not very good at it. People tend to focus on one predictor too much and overestimate its relative importance. Also, there is a great deal of inconsistency in how predictors are combined from one situation to the next (Meehl, 1954). The result is consistently worse prediction than using mechanical combination, which holds true even for experts and across settings and situations (Grove, Zaid, Lebow, Snitz & Nelson, 2000). Similar problems can occur in the case of rational weighting, which allows subject matter experts to determine the weights applied to predictors.

An adaptation of MPT offers an additional method for generating weights for a set of predictors. MPT weighting, like multiple regression, is sample dependent but employs a different methodology, as described in the next section. As of the current research, it is unknown how MPT will compare to other methods, though based on comparisons between the effectiveness of other methods, some general patterns are expected to hold true. The focus of this study is to compare MPT to unit weighting and multiple regression weighting to better understand the conditions where MPT may be a superior alternative to these methods.

Modern Portfolio Theory

MPT is based largely on the work of Harry Markowitz (1952; 1959). The method laid out in his work, *Portfolio Selection*, chooses, from all combinations of available assets, those assets that will minimize the risk (volatility or variance), for a given rate of return.

The expected return of a portfolio, $E[r_p]$, is defined as the sum of the expected returns for each of n assets, $E[r_i]$, multiplied by the respective weights, w_i, or proportions of each asset in the portfolio, where the sum of all weights equals 1.

$$E[r_p] = \sum_{i=1}^{n} w_i E[r_i]\,; \tag{1}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{2}$$

The variance of a portfolio with *n* assets, σ_p^2, is a function of the individual variances of the assets and the covariance between each combination of assets, where σ_{ij} is the covariance between asset *i* and asset *j*. The variance of an asset is calculated where *i* and *j* are equal.

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i \sigma_{ij} w_j \tag{3}$$

The Sharpe Ratio for a portfolio, S_p, is the expected return of the portfolio, $E[r_p]$, minus the expected return of a risk-free asset, $E[r_{risk\ free}]$, divided by the standard deviation of the portfolio, σ_p, (Sharpe, 1966; 1975). A risk-free asset is usually considered to be a government issued T-bill, which carries virtually no risk. The difference between the expected return of the risk-free portfolio and the expected return of the risk-free asset is called the risk premium. The goal of MPT is to find the proportion of each asset that should be included in a portfolio that will maximize the Sharpe ratio.

$$S_p = \frac{E[r_p] - E[r_{risk\ free}]}{\sigma_p} \tag{4}$$

MPT approaches this problem as a minimization of the portfolio variance from Equation 3, subject to the constraints of Equations 1 and 2. One way to solve for local minima, subject to equality constraints, is to use the method of *Lagrange* multipliers (Lagrange, 1788). This method introduces two new constants called *Lagrange* multipliers, λ_1 and λ_2, and a function is created that is the sum of the first function, the first constraint multiplied by the first *Lagrange* multiplier, and the second constraint multiplied by the second *Lagrange* multiplier. The partial derivative of the *Lagrangian*, with respect to the common variable, is taken, and the result is set equal to zero. The extrema of the first function are then critical points of the *Lagrangian*, but may not be extrema of the *Lagrangian* if the constants, λ_1 and λ_2, are nonzero.

To simplify the math later, Equation 3 is multiplied by ½. This will not change the solution to the minimization problem. For this problem, three equations are used to set up the *Lagrangian*:

$$L = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i \sigma_{ij} w_j - \lambda_1\left(\sum_{i=1}^{n} w_i E[r_i] - E\left[r_p\right]\right) - \lambda_2\left(\sum_{i=1}^{n} w_i - 1\right) \qquad (5)$$

We set $\frac{\partial L}{\partial w_i} = 0$ for each w_i in n cases. These yields n equations of the form:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} w_j - \lambda_1 E[r_i] - \lambda_2 = 0, i \in \{1, 2, \dots, n\} \qquad (6)$$

We now have $n + 2$ unknowns, (w_1, $w_{2,\dots,}$ w_i, λ_1, λ_2), and $n + 2$ equations:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} w_j - \lambda_1 E[r_i] - \lambda_2 = 0, i \in \{1, 2, \dots, n\};$$

$$E\left[r_p\right] = \sum_{i=1}^{n} w_i E[r_i]\,;$$

$$\sum_{i=1}^{n} w_i = 1$$

Our solutions to the equations are then, (w_1, $w_{2,...,}$ w_i, λ_1, λ_2). The solutions will only exist for feasible pairs of expected return, $E(r)$, and risk, σ, for portfolios weights (w_1, $w_{2,...,}$ w_i). The feasible set is defined where:

$$E[r_p] = \sum_{i=1}^{n} w_i E[r_i]\,;$$

and

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i \sigma_{ij} w_j$$

The efficient frontier is a hyperbola consisting of all boundary points between the feasible set and points that are not feasible where allocations to the risk-free asset are equal to zero. Figure 1 displays portfolios within a feasible set, and the upper portion of the efficient frontier. For all portfolios on the lower half of the efficient frontier, there exists a corresponding point on the upper half with the same risk but higher return. For this reason, we ignore the lower half of the hyperbola. All points within the hyperbola are considered feasible but not efficient. For any point within the upper half of the hyperbola, there exists some other point within the hyperbola with higher return and lower risk. Points outside of the hyperbola are not feasible, meaning they are not possible portfolios.

Combining the risky assets with risk-free assets is the final step needed to obtain a complete portfolio. Because the rate of return for a risk-free asset is fixed, with zero variance, the returns are uncorrelated with risky assets. This results in a linear relationship between risk and return for changes in proportions of the risk-free asset. The Capital Allocation Line, CAL, represents all possible combinations of risky and risk-free assets that an investor could make. As seen in Figure 1, the line intercepts the zero-risk (zero-variance) proposition at the risk-free rate of return. As less of the risk-free asset is selected, returns for portfolios increase (it is assumed that risky assets have higher expected return) and risk also increases. The CAL runs tangent to the efficient frontier at the tangency portfolio where 100% of the assets in the portfolio are risky assets.

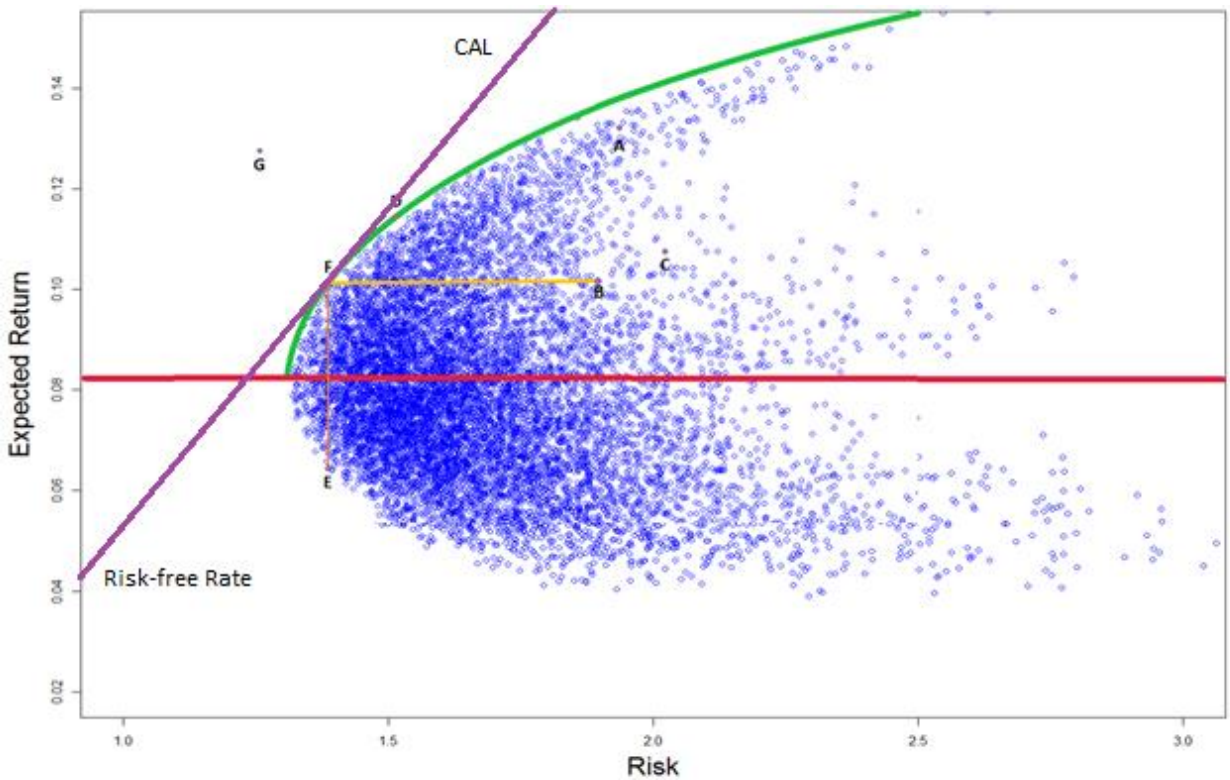

Figure 1 Risk and Return for 10,000 Randomly Generated Portfolios. Portfolios A, B, and C are all inefficient, meaning there are portfolios that have less risk and more reward. Observe that B has an identical return to F, but F is less risky. Portfolios D, E, and F are all on the boundary of the efficient frontier, but portfolio E is on the lower half of the hyperbola. For any point on the lower half there is a point on the upper half with more return and identical risk (observe E and F). The meaningful part of the efficient frontier is above the red line, and the relevant part of the efficient frontier is represented by the green curve. G represents a portfolio that is not feasible, as it lies outside the efficient frontier. The Capital Allocation Line (CAL) in purple represents possible combinations of risky and risk-free assets. The CAL meets the efficient frontier at the tangency portfolio, F. The tangency portfolio is the optimal portfolio given the data.

Adapting MPT to Personnel Selection

The purpose of this study is to adapt MPT to generate weights for predictors for employee selection. The method described above has been altered in a few important ways. Selection data must be organized in such a way that it conforms to the patterns of risky assets. First, the sample used to generate the weights is ordered in terms of performance rank. This has the effect of equating asset value over time with performance increase over ranks. Presumably, performance measurements were taken by the organization in a forced, normal distribution. The predictor scores, obtained from selection activities (tests, biodata, etc.), are arranged in the same order, as specified by the performance ranking. Differences between the predictor scores between each of these performance ranks are synonymous with the returns for each asset (the change in asset value between each time period). Because the predictors are arranged from smallest to largest in terms of performance, those predictors that are more highly correlated with performance will have smaller variance between ranks. Lower variance is preferred by the model and will result in higher weights for that predictor. If predictors in the model are similar they will tend to have higher positive covariance. This is not preferred by the model, and will result in lower weights for those predictors. In summary, instead of minimizing the total variance in price between days for each asset, the model now minimizes the total variance in predictor scores between ranks.

As mentioned in the last section, the CAL is used in MPT to determine the one optimal portfolio. In selection, there are no predictors analogous to risk-free assets. Such a predictor would have zero variance and some guaranteed increase in performance for each additional unit of the predictor that is used for prediction. It is possible to guarantee a zero increase in predicted performance. So, in this case, the risk-free rate is set to zero and the CAL intercepts at the zero variance, zero return proposition. The Sharpe ratio is then reduced to:

$$S_p = \frac{E[r_p]}{\sigma_p}$$

In this study, comparisons between methods are made through the lens of signal detection theory. Signal detection theory offers a way to assess the quality of decisions for each method in terms of the outcomes of decisions and the accuracy of decisions (Peterson, Birdsall, and Fox, 1954). The decision in both signal detection and selection is a binary classification decision. In terms of selection, the applicant scores at a certain level on a predictor, either above or below a chosen cutoff score, and then a decision is made whether to accept or reject the applicant based on the predictor score.

The selection decision results in four possible outcomes. Correct selections occur when the applicant scores above a cutoff score on the predictor and scores above a cutoff score on the criterion (performance). Correct rejections occur when the applicant scores below the cutoff score on the predictor and would have scored below the cutoff score on the criterion. False positives occur when the applicant scores above a cutoff score on the predictor but below a cutoff score on the criterion. False negatives occur when the applicant scores below a cutoff score on a predictor but would have scored above the cutoff score on the criterion.

In practice, it is impossible to know the number of correct rejections and false negatives because applicants in these categories are not actually hired. The solution to this problem is to create a predictive study, where predictor scores are taken and all applicants are hired without using the predictor. Later, the number of correct rejections and false negatives for applicants that would have been rejected can be determined. Given that the organization must hire all applicants and suffer from the poor performance of many more applicants than if they had used a selection method, predictive studies such as this are often very costly. For this reason, predictive designs are rarely used. Because this study involves simulated data, the true signal (whether the applicant will perform above the cutoff for the criterion) is known.

The application of MPT to personnel selection has thus far been unexplored. MPT has been applied to other decision making processes such as modeling a regional labor force (Conroy, 1974) and modeling a stable self-concept (Chandra & Shadel, 2007). For most organizations, maximizing performance is not the only goal. Just as maximizing return on investment is not the only goal for investors.

According to the Gauss-Markov theorem, ordinary least squares (OLS) regression provides the best linear unbiased estimation of the coefficients (1823). This means that any deviation from the regression line will only make prediction worse. It is reasonable to expect that the same pattern should exist for MPT weighting as with other weighting methods that are not the best linear fit. However, two favorable developments also can occur. The number of false positives may decrease, and the number of correct rejections may increase. Figure 2 below demonstrates how this might be represented graphically. The result is identical to what one would see from a change in slope between different regression lines.

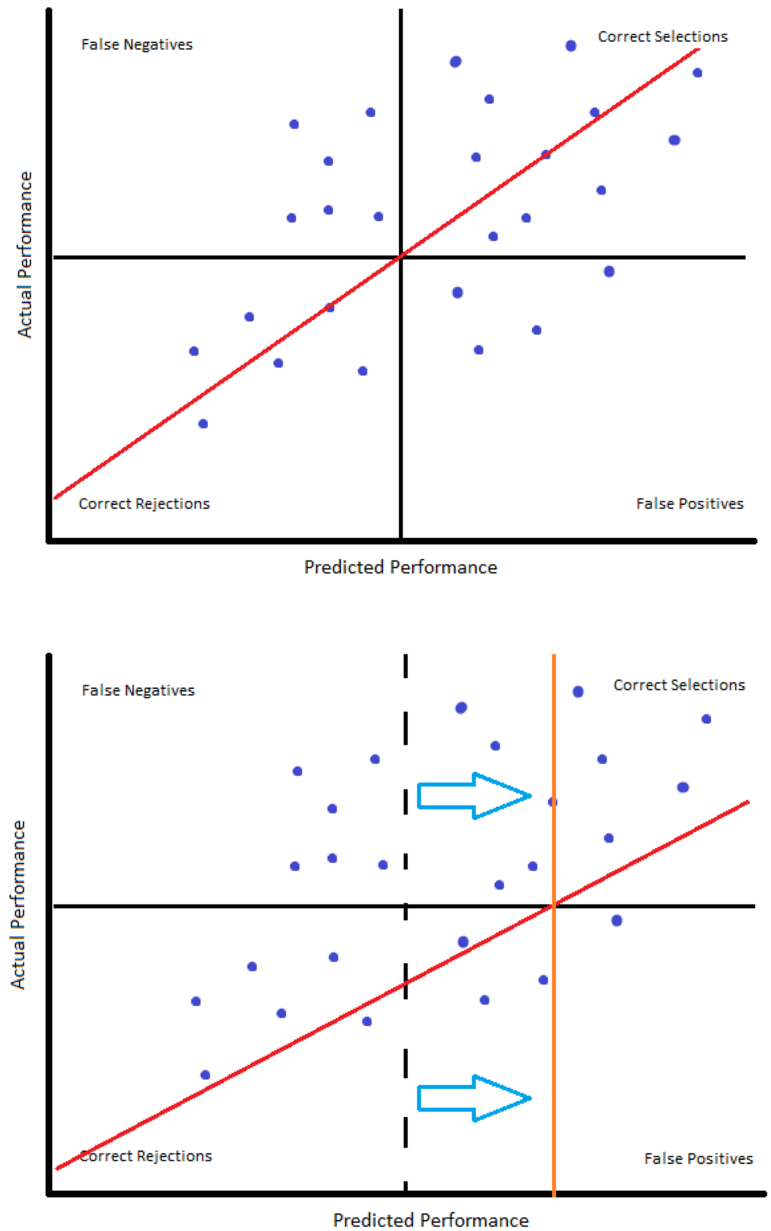

Figure 2: A Change in Validity. The top image represents how a cutoff score, represented by the vertical black line, based on a regression line, represented by the red line, separates selected and rejected applicants into four categories, correct selections, correct rejections, false positives, and false negatives. The bottom image shows how a change in slope can reduce false positives and correct selections, while increasing false negatives and correct rejections, by acting like a change in the cutoff score.

Consistent with the Gauss-Markov theorem, a change from the multiple regression weighting may result in two favorable outcomes:

Hypothesis 1a: MPT will produce fewer false positives than multiple regression.

Hypothesis 1b: MPT will produce more correct rejections than multiple regression.

While confirming these hypotheses may be evidence that MPT is a weaker model than multiple regression, it is important to consider the utility of outcomes when making this type of determination. For example, false positives may be extremely costly to an organization. If the cost of a poorly performing worker greatly exceeds the added value of a superior worker, then false positives may prove to be the most important consideration. For example, consider the job of automobile assembly. Poor performance could slow down the entire assembly line or could result in serious accidents, fines, or lawsuits. However, superior performance may be indistinguishable from just above average performance in terms of results. False negatives, on the other hand, are not directly measurable by the organization and represent a loss in terms of opportunity cost. The organization would have benefited by correctly hiring the applicants classified as false negatives.

Correct selections, correct rejections, false positives, and false negatives can all be important outcomes. But, because they do not represent the outcomes in terms of base rates, they cannot offer a clear-cur way to compare models in terms of accuracy. If a method always chose all applicants, then it would be 100% successful in terms of correct selections, but it would also be 100% unsuccessful in in terms of correct rejections. Likewise, it would produce no false negatives, but it would also produce the maximum number of false positives. How well the method can differentiate the data is often called accuracy or discriminability. This issue of accuracy should also be addressed. Signal detection theory offers a way to address the issue of accuracy within the binary decision making framework of selection.

In this study, researchers address four measures of accuracy. The *false positive rate* is the number of false positives divided by the sum of false positives and correct selections. The *false negative rate* is the number of false negatives divided by the sum of false negative and correct rejections. *Specificity* is defined as the number of correct rejections divided by the sum of correct rejections and false negatives, and *Sensitivity* is defined as the number of correct selections divided by the sum of correct selections and false positives. Consistent with the findings of Trattner and with those of Lawshe and Schucker, it is expected that MPT will not differ greatly in terms of how the method performs in terms of rates of accuracy (1963; 1959).

Hypothesis 2: MPT will perform comparably to unit weighting and multiple regression in a simulated predictive study in terms of binary classification performance measures.

The sample size of the initial group used to generate the weights should also play a critical role. In the case of a small sample, violations of assumptions are more likely, and weights generated from the samples would vary more for sample dependent weighting, especially for multiple regression. MPT weighting is also sample dependent, but is less reliant on a large sample than multiple regression, as the variance between ranks is expected to be more stable than the distance of data points to a regression line if the initial sample variance is normally distributed. The smallest of the sample size where MPT weighting is expected to be superior to multiple regression, in terms of stability of weights and prediction, is likely to be somewhere between 30, where violations of assumptions for multiple regression begin to disappear, and 100, where Schmidt observed multiple regression beginning to dominate unit weighting in terms of prediction (1971).

Hypothesis 3: There exists a lower-bound threshold for the size of the initial sample for observing noticeable differences between weighting methods in terms of the stability of weights and the quality of prediction.

It has long been established that the effectiveness of multiple regression in selection is affected by the selection ratio (Taylor & Russel, 1939). The selection ratio is the number of chosen applicants divided by the total number of applicants. If the cutoff score is raised, the selection ratio decreases. At higher chosen cutoff scores, the proportion of correctly selected applicants also increases. This is also true for MPT and unit weighting, or any combination of weighted predictors that are positively correlated with the criterion measure. Reexamining Figure 2, using a method other than multiple regression can have an effect like artificially raising the cutoff score. From this, it is unclear what the cumulative effect of both changing the cutoff score and choosing MPT or unit weighting over multiple regression will be. We suspect that there are cutoff scores where one method will outperform another.

Hypothesis 4: There exists a set of cutoff scores where each method will perform better than others in terms of binary classification measures and binary classification performance measures.

One of the defining features of MPT as it applies to risky financial assets, is that combinations of assets that are negatively correlated can be combined to reduce the overall variance of the model. In finance, it has long been noted that the prices of bonds rise when the prices of stocks fall and vice versa. Bonds represent a guaranteed investment return, albeit lower than the typical returns for stocks. So, investors fleeing stocks with falling prices will turn to bonds. The increase in demand for bonds then raises the selling price of the bond. Holding a combination of stocks and bonds can, therefore reduce the volatility of a portfolio, helping to make sure the value of the total portfolio does not take large swings in value. The presence of negatively correlated assets that are both positively correlated with returns is what makes MPT so useful for investors. Mathematically, this can be explained by noting that the total variance of a portfolio is the sum of all asset variances and covariances. When covariances are negative, the total variance of the portfolio is reduced.

For an investor, the important result of the effect is the reduction of portfolio volatility, but for selection the most useful aspect may be something different. A side effect of combining negatively correlated asset returns is that other assets in the model are weighed less heavily. The model seeks to minimize volatility, so stocks and bonds with a strong negative correlation are preferred and given higher relative weights. So, in a three-asset portfolio (ABC) where A and B are negatively correlated, but uncorrelated with C, and the assets have similar returns, MPT will select higher weights for A and B.

In employee selection, there is a long-standing problem related to subgroup differences in cognitive ability tests. Cognitive ability is a preferred measure because it is often the strongest predictor of performance. Particularly, Blacks and Hispanics have typically scored lower on these kinds of tests. If diversity is one of the goals of an organization, cognitive ability tests can often be a barrier to realizing that goal by helping to select fewer black and Hispanic applicants. This can also result in legal issues if the ratio of minority applicants hired to those that applied falls below four fifths of a similar ratio for the unprotected groups ("Disparate Impact and Reasonable Factors Other Than Age Under the Age Discrimination in Employment Act," 2012).

Measurement research as of now, has been able to reduce but not solve the problem. There has been a strong effort to create cognitive ability tests that produce smaller subgroup differences though differential item functioning, alternate delivery systems, reducing cultural references, and a reduction in reliance on reading comprehension (Ployhart & Holtz, 2008). So far, test makers have been unable to eliminate these problems completely. With multiple regression weighting, a common-sense approach might be to simply add more predictors that are not related to adverse impact. Unfortunately, this does not seem to solve the problem and, in some cases, makes the problem worse (Sackett & Ellingson, 1997). Applying methods that alter the standard top-down selection approach, such as score banding which capitalizes on the unreliability of measurements, can be used to reduce adverse impact (Schmidt, 1988; Truxillo & Bauer 1999). Similar problems with adverse impact can occur with other constructs and for other protected groups. Ployhart and Holtz summarize the key findings in subgroup differences for 19 commonly used predictors for racioethnic and gender subgroups and describe 16 different approaches to reducing adverse impact (2008). The variety of possible combinations of the presence of predictor-subgroup differences and adverse impact reduction strategies is astronomical, 36,893,488,147,419,103,232, making strategies for dealing with these problems, practically speaking, ad hoc.

MPT offers a different approach to the problem. If specific predictors are responsible for generating these subgroup differences in selection, adverse impact can be reduced by finding other predictors positively correlated to performance, but negatively correlated to each other. This is the same approach as using combinations of stocks and bonds in the financial application of the method. For example, if the use of cognitive ability as a predictor is causing adverse impact, adding two more predictors, positively correlated with performance, negatively correlated with each other, can reduce the volatility of the selection portfolio and the weight given to cognitive ability as a predictor.

Hypothesis 5: Introducing a new selection predictor that is uncorrelated with cognitive ability, correlated positively with performance, and negatively with a third predictor will preserve the predictive power of the model and reduce the weight given to a third predictor.

Methods

Data was simulated in R, version 3.3.2 (The R Core Team, 2013). The code used to generate the results is available on GitHub, https://github.com/EricLeingang/Thesis-Code, and is also available in the Appendix. A Cholesky decomposition of a specified correlation matrix (see Table 1) was used to generate random correlated data for hypotheses 1-4. Coefficients in the correlation matrix were reproduced from meta-analyses to represent common measures with typical values (Hunter & Hunter, 1984, Barrick & Mount, 1991; Tett, Jackson, & Rothstein, 2006). The decomposition was multiplied by a matrix of random normal numbers. A normal distribution was chosen to represent a forced distribution performance rating system and normally distributed predictor scores. The data was then rescaled to eliminate negative predictor scores and performance ratings. A diagnostic check was made to ensure that the data were still correlated in the same way as the original matrix, after the transformations. These data were considered to be an initial sample an organization would use to generate weights for predictors.

Table 1: Correlation Matrix Used for Hypotheses 1-4.

	Performance	Conscientiousness	Cognitive Ability	Work Sample
Performance	1.00	0.20	0.51	0.57
Conscientiousness	0.20	1.00	0.01	0.09
Cognitive Ability	0.51	0.01	1.00	0.34
Work Sample	0.57	0.09	0.34	1.00

Three sets of weights were generated from the simulated data. For MPT weighting, the data was organized by performance rank to create an, overall, positive, increasing trend in predictor scores, analogous to the positive, increasing trend in risky asset prices in the stock market. Each time period in the financial model was then analogous to one additional rank in the selection model. The difference in performance between subsequent ranks is analogous to the return in the financial model, or difference between the beginning and ending values during that time period. These differences, in predictor scores between performance ranks, were calculated and used to form a matrix of returns. MPT weights were calculated based on the point on the efficient frontier that maximized the Sharpe ratio, given a zero-return risk-free rate, Figure 3 below, using modified code by Matuszak, (n.d.). Quadratic programming, the quadprog package, was used to solve the system of equations that defined the optimal weights (version1.5-5, Berwin, Turlach, and Weingessel, 2013). Standardized multiple regression weights were calculated and rescaled so that the sum of weights would equal 1. Unit weights were also generated and rescaled. The weights for one trial, for each method, are summarized in Table 2, below.

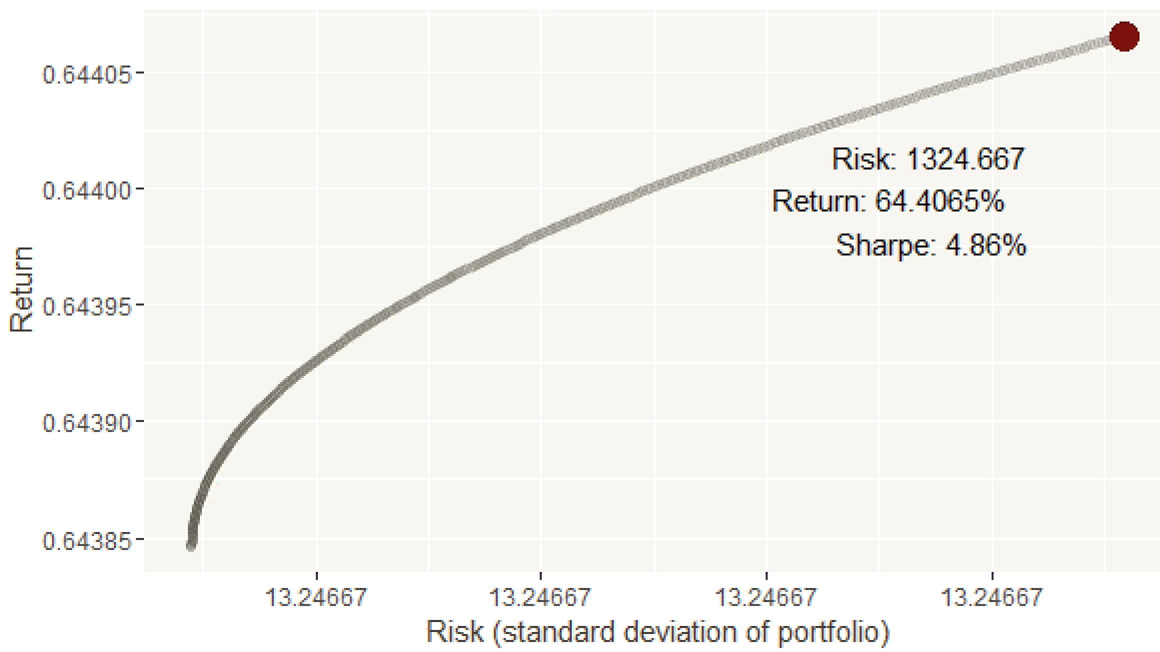

Figure 3: Efficient Frontier and Optimal Portfolio. This figure represents the efficient frontier and the optimal portfolio for one trial.

Table 2: Portfolio Weights Determined Through Different Weighting Methods. This table displays weights generated by unit weighting, multiple regression weighting, and MPT weighting.

Weighting Type	Conscientiousness	Cognitive Ability	Work Sample

Unit Weightings	0.33	0.33	0.33
Regression Weighting	0.14	0.38	0.48
MPT Weighting	0.32	0.26	0.42

An applicant pool was created using a process that was identical to that used to create the initial sample, but using a different seed. The initial matrix of applicant performance and predictor scores was 100 times larger than the matrix of final applicant scores. Final applicant scores were randomly chosen from the larger matrix of performance and predictor scores to generate the pool of applicant scores used in each trial using the plyr package (Wickham, 2011). This was done to create applicant pools that had more opportunity for outliers and other assumption violations that might occur naturally from sampling error. For each method, predicted performance was calculated using a weighted sum of predictor scores from each simulated applicant. Each simulated applicant's true performance score, was compared to their predicted performance scores. The effectiveness of each method was defined in terms of four basic categories, correct selections, correct rejections, false positives, and false negatives, and four composite categories, sensitivity, specificity, false positive rate, and false negative rate.

For the first simulation, the size of the final applicant pool size was held constant at 100. Performance was measured on a 100-point scale and cutoff score was held constant at 60. The initial sample size was held constant at 100. To test hypotheses 1 and 2, 10,000 trials were conducted and the cumulative results for all trials were recorded for correct selections, correct rejections, false positives, false negatives, sensitivity, specificity, false positive rate, and false negative rate. The average weights generated for each method were also recorded.

Sample size, and cutoff score final applicant pool size was held constant at 100. Performance was measured on a 100-point scale and cutoff score was held constant at 60. The initial sample size was allowed to vary from 10 to 200. For Hypothesis 4, the final applicant pool size was held constant at 100. The initial sample size used to generate weights was held constant at 100. Performance was measured on a 100-point scale. Cutoff scores were allowed to vary from 1 to 100. For hypotheses 3 and 4, the cumulative results of 1000 trials were recorded and summed at each level of applicant pool size and cutoff score, respectively, for each method to establish boundaries.

For the fourth simulation, a new correlation matrix was needed. Two predictors positively correlated with performance and negatively correlated with each other were needed to fulfill the conditions. The following scenario was used to generate the correlation matrix in Table 3:

> *Performance at a sales job was found to be related to several important predictor scores. Performance was highly correlated to cognitive ability, this was related to strategic thinking and decision making. Performance was moderately correlated to conscientiousness; organization and being prepared helped to generate repeat sales. Risk taking was moderately correlated to performance. Salespersons who were willing to try new methods and seek out new clients were more successful. Risk takers tended to be less conscientious than those who were not risk takers.*

Table 3: Correlation Matrix Used for Hypothesis 5.

	Performance	Conscientiousness	Cognitive Ability	Risk Taking
Performance	1.00	0.20	0.51	0.34
Conscientiousness	0.20	1.00	0.01	-0.25
Cognitive Ability	0.51	0.01	1.00	0.22
Risk Taking	0.34	-0.25	0.22	1.00

The size of the final applicant pool was held constant at 100. Performance was measured on a 100-point scale and cutoff score was held constant at 60. The initial sample size was held constant at 100. 10,000 trials were conducted and the cumulative results for all trials were recorded for correct selections, correct rejections, false positives, false negatives, sensitivity, specificity, false positive rate, and false negative rate. The average weights generated for each method were also recorded.

Results

The first simulation was conducted to test hypotheses 1a, 1b, and 2. The results of the first simulation are summarized in Table 5, below:

Table 4: Results of Simulation 1. For this simulation, the initial sample was held at 100, the number of applicants was held at 100, and the cutoff score was held at 60 on a 1-100 scale. The data in the tables represents the cumulative results of 10,000 trials. Under these conditions, false positives were higher for MPT weighting than multiple regression weighting. Correct rejections were lower for MPT than multiple regression. Hypothesis 1 was not supported. Unit Weighting, Multiple regression weighting, and MPT weighting performed about as well as each other in terms of binary decision performance measures, in support of Hypothesis 2.

	Unit Weighting	Multiple Regression Weighting	MPT Weighting
<u>Binary Decision Measures</u>			
Correct Selections	321094	327790	322037
Correct Rejections	391075	393663	391030
False Positives	135666	133078	135711
False Negatives	152165	145469	151222
<u>Binary Decision Performance Measures</u>			
Sensitivity	0.68	0.69	0.68
Specificity	0.74	0.75	0.74
False Positive Rate	0.30	0.29	0.30
False Negative Rate	0.28	0.27	0.28
<u>Average Predictor Weights</u>			
Conscientiousness	0.33	0.18	0.36
Cognitive Ability	0.33	0.36	0.34
Work Sample	0.33	0.46	0.30

For the given initial conditions of the simulated data, Hypothesis 1a and Hypothesis 1b were not supported. MPT weighting produced more false positives and fewer correct rejections than multiple regression weighting, although the results were very close to equal. False positives differed by less than 2% and correct rejections differed by less than 1%. Hypothesis 2 seemed well supported by the data, with only 1% differences between MPT and multiple regression weighting for specificity, sensitivity, false positive rate, and false negative rate.

Hypothesis 3 posited that, for some lower bound threshold of initial sample size, MPT weighting would predict better and have more stable weights than multiple regression weighting. The following figures graphically represent the output of the second simulation. In the first part of the simulation, 1,000 trials were conducted for the binary decision outcomes, for each initial sample size from 10 to 200. Figure 4, below, displays the results for correct selections. Figure 5, below, displays the results for correct rejections. Figure 6, below, displays the results for false positives. Figure 7, below, displays the results for false negatives.

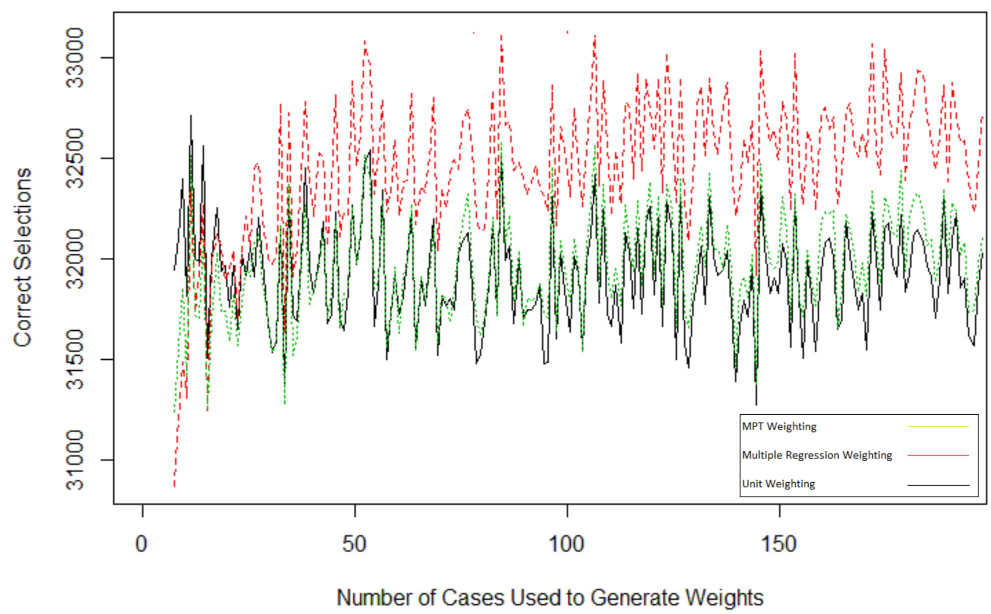

Figure 4: Correct Selections for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to produce more correct selections than unit or MPT weighting.

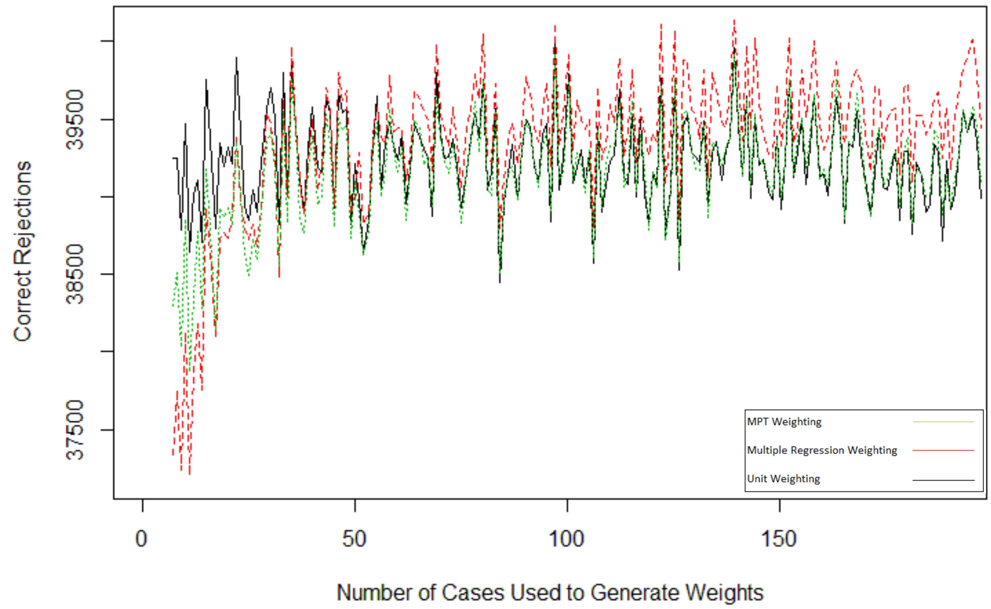

Figure 5: Correct Rejections for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30 multiple regression tends to produce more correct rejections than unit or MPT weighting.

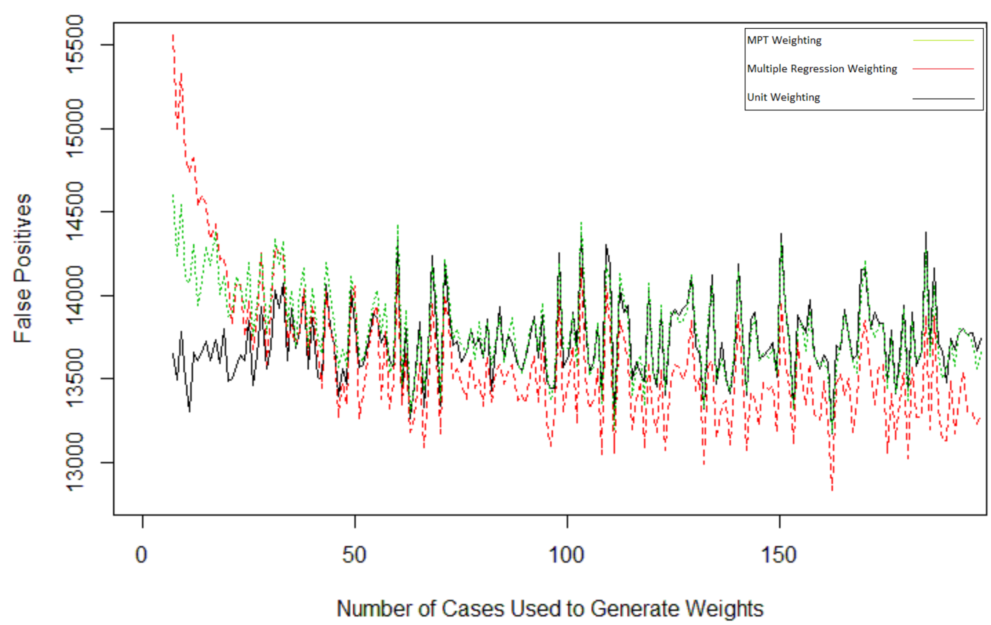

Figure 6: False Positives for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to produce fewer false positives than unit or MPT weighting.

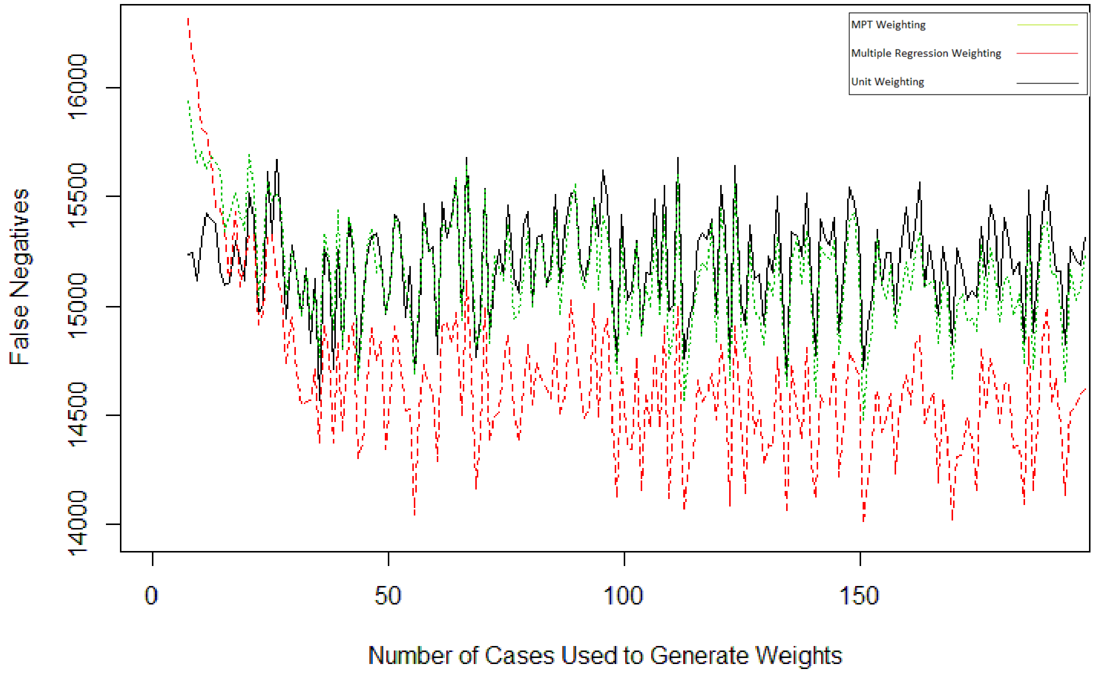

Figure 7: False Negatives for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to produce fewer false negatives than unit or MPT weighting.

Both MPT weighting and unit weighting appear to dominate multiple regression weighting when sample sizes are below 30, where violations of assumptions become problematic for multiple regression. Unit weighting appears to perform best. This offers partial support to Hypothesis 3. Further support can be found by examining the binary decision performance measures.

For the second part of the second simulation, the same data were used to produce binary decision performance measures. The next four figures display the results of the binary decision performance measures for 1,000 trials at each sample size. Figure 8, below, displays the results for sensitivity. Figure 9, below, displays the results for specificity. Figure 10, below, displays the results for the false positive rate, and Figure 11, below, displays the results for the false negative rate.

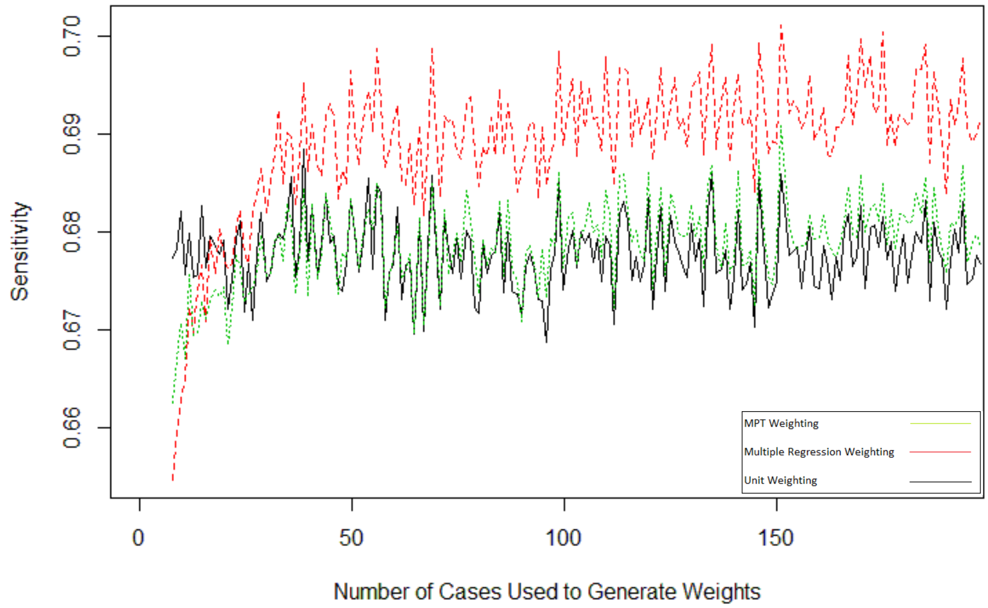

Figure 8: Sensitivity for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to have higher sensitivity than unit or MPT weighting.

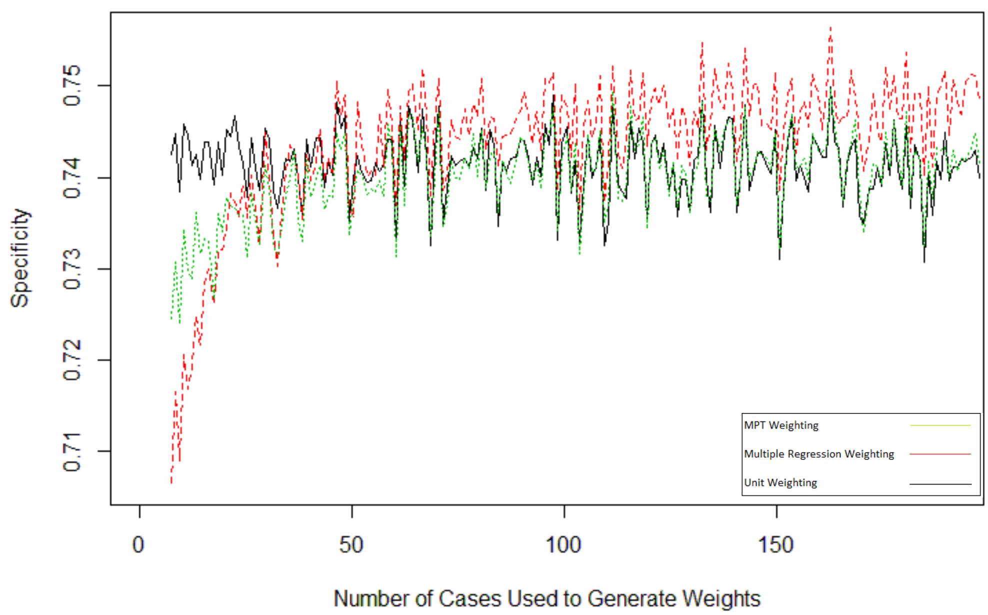

Figure 9: Specificity for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to have higher specificity than unit or MPT weighting.

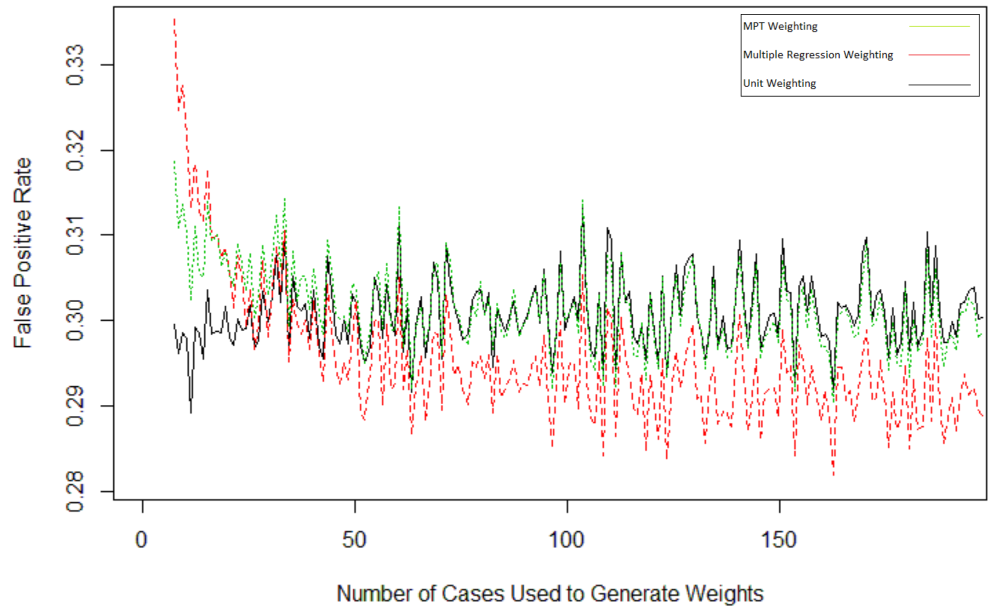

Figure 10: False Positive Rate for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to have a lower false positive rate than unit or MPT weighting.

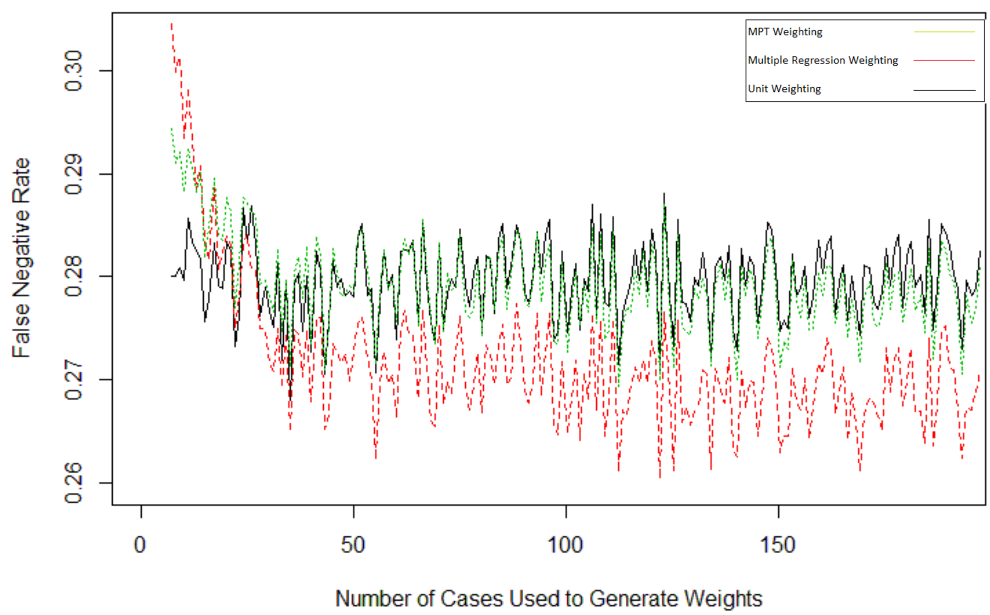

Figure 11: False Negative Rate for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. When the initial sample size is above ~30, multiple regression tends to have a lower false negative rate than unit or MPT weighting.

MPT weighting and unit weighting appear to dominate multiple regression weighting when sample sizes are below 30, and again, unit weighting appears to perform best. This offers additional support to Hypothesis 3.

The last part of the second simulation was designed to test the first part of Hypothesis 3 which suggests that, below a certain sample size, multiple regression weights are less stable than MPT weights. Figure 12, 13, and 14, below, display the weights generated by each method at each sample size.

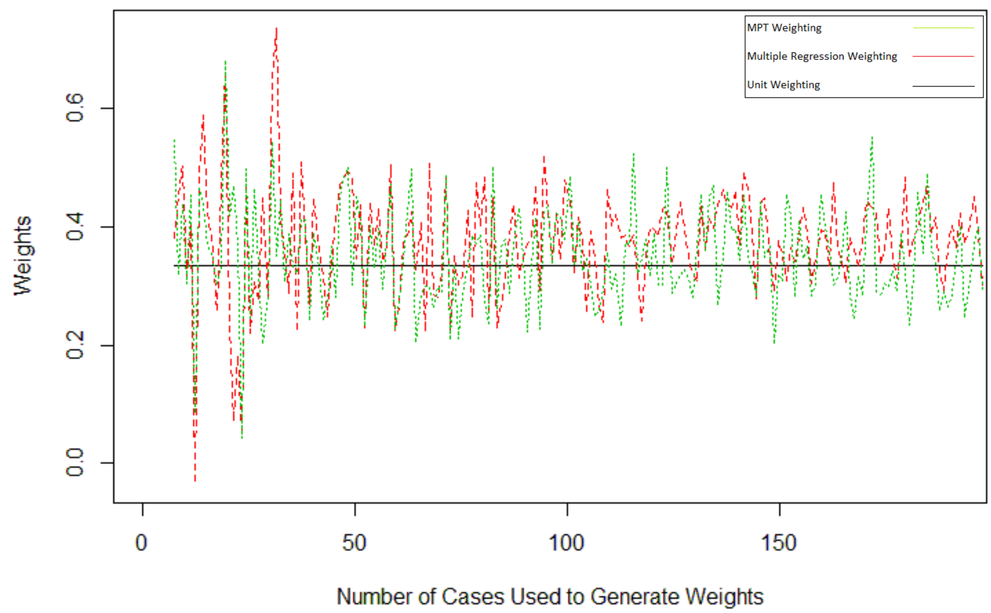

Figure 12: Cognitive Ability Weights for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. Both MPT and multiple regression weights were less stable with smaller sample sizes, less than ~30. Multiple regression weights are less stable than MPT weights.

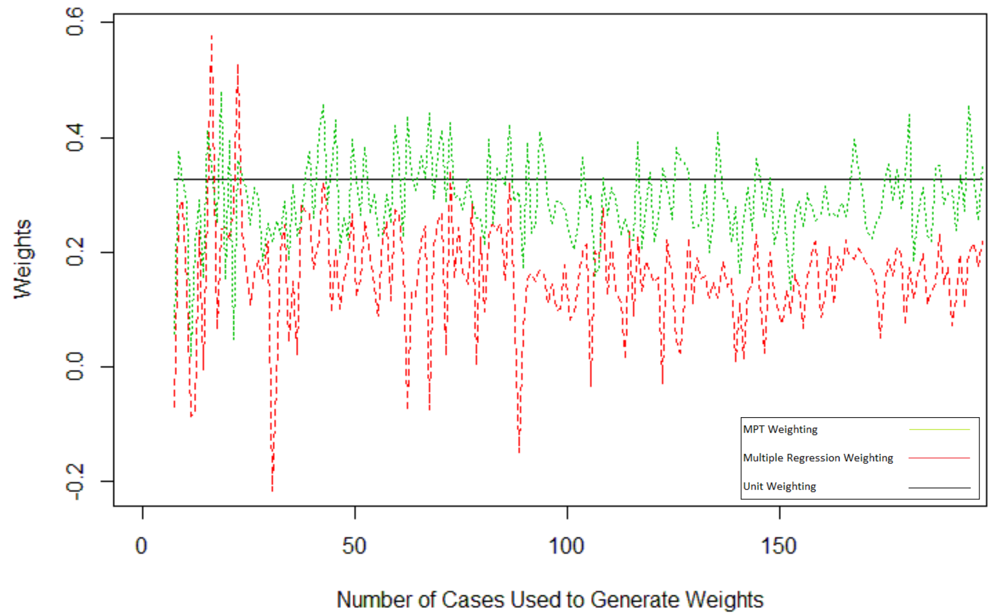

Figure 13: Conscientiousness Weights for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. Both MPT and multiple regression weights are less stable with smaller sample sizes, less than ~30. Multiple regression weights were less stable than MPT weights.

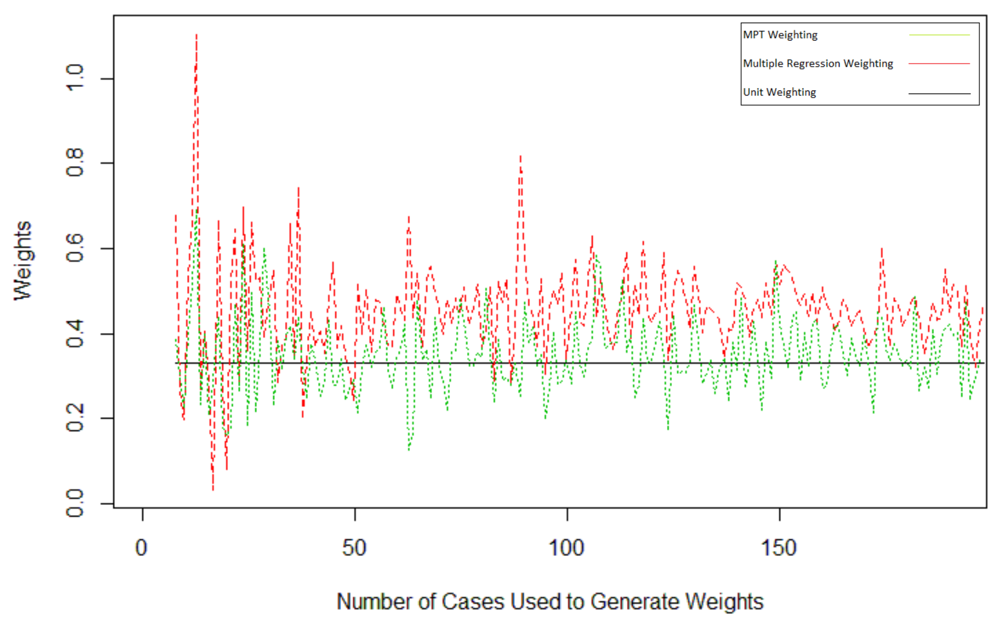

Figure 14: Work Sample Weights for Different Initial Sample Sizes. This graph represents data generated from an applicant pool of 100 and a cutoff score of 60 accumulated over 1,000 trials for each applicant pool size from 10 to 200. Both MPT and multiple regression weights are less stable with smaller sample sizes, less than ~30. Multiple regression weights were less stable than MPT weights.

The weights generated by the multiple regression have more variability when sample sizes are below ~30 than MPT weights. The relative instability of the multiple regression weights supports the first part of Hypothesis 3.

The third simulation allowed the cutoff score to vary in order to test Hypothesis 4. Hypothesis 4 states there is a set of cutoff scores such that MPT performs better than multiple regression in terms of binary classification measures and binary classification performance measures. In the following four figures the cumulative results of 1,000 trials are displyed for binary classifications at each cutoff score from 1 to 100. Figure 15, below, displays correct selections. Figure 16 displays correct rejections. Figure 17 displays false positives, and Figure 18 displays false negatives.

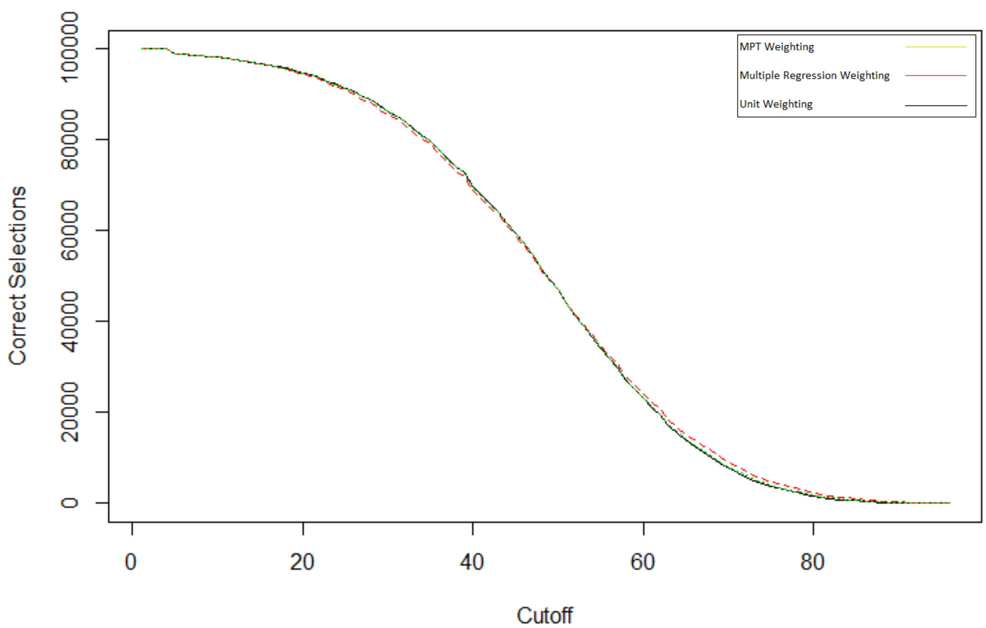

Figure 15: Correct Selections for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 50, multiple regression weighting produced more correct selections.

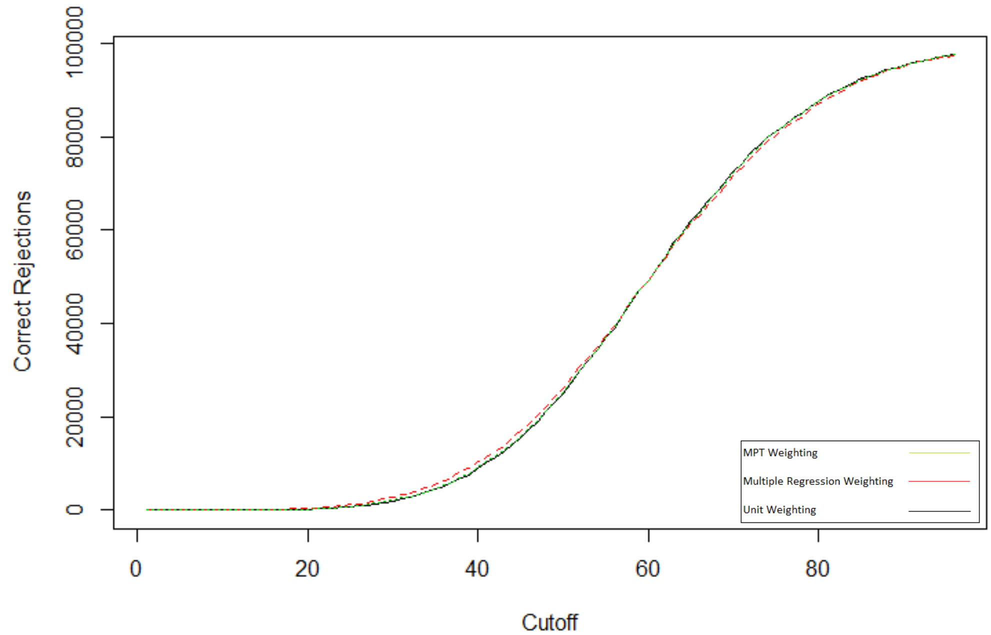

Figure 16: Correct Rejections for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 60, multiple regression weighting produced fewer correct rejections.

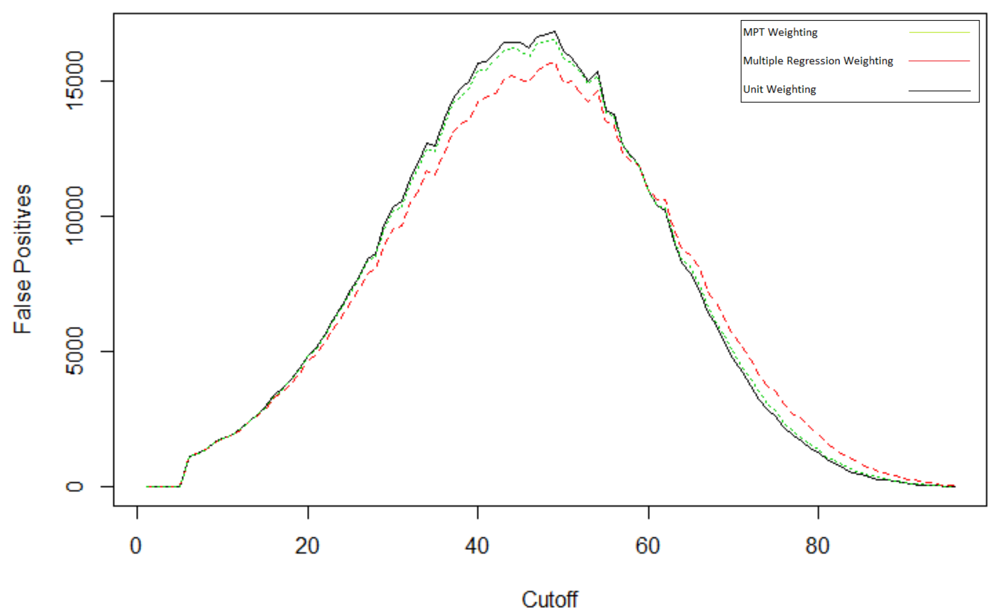

Figure 17: False Positives for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 60, multiple regression weighting produced more false positives.

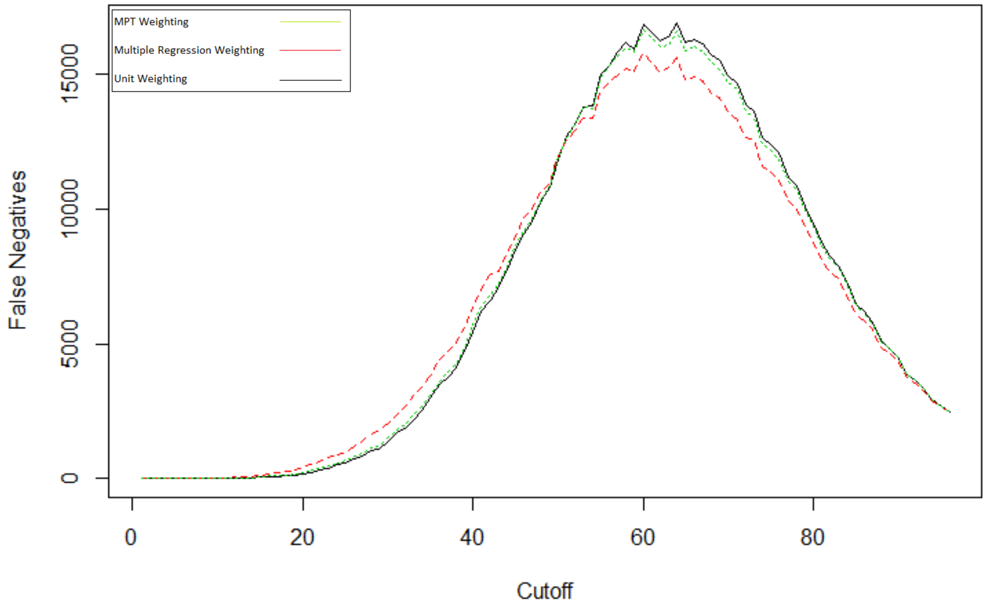

Figure 18: False Negatives for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 50, multiple regression weighting produced fewer false negatives.

To test how binary clasification performance measues may differ by cutoff score, the outcomes of each decision were recorded for each cutoff score, ranging from 1 to 100. The results were plotted in the figures below. Figure 19 displays sensitivity. Figure 20 displays specificity. Figure 21 displays false positive rate, and Figure 22 displays false negative rate.

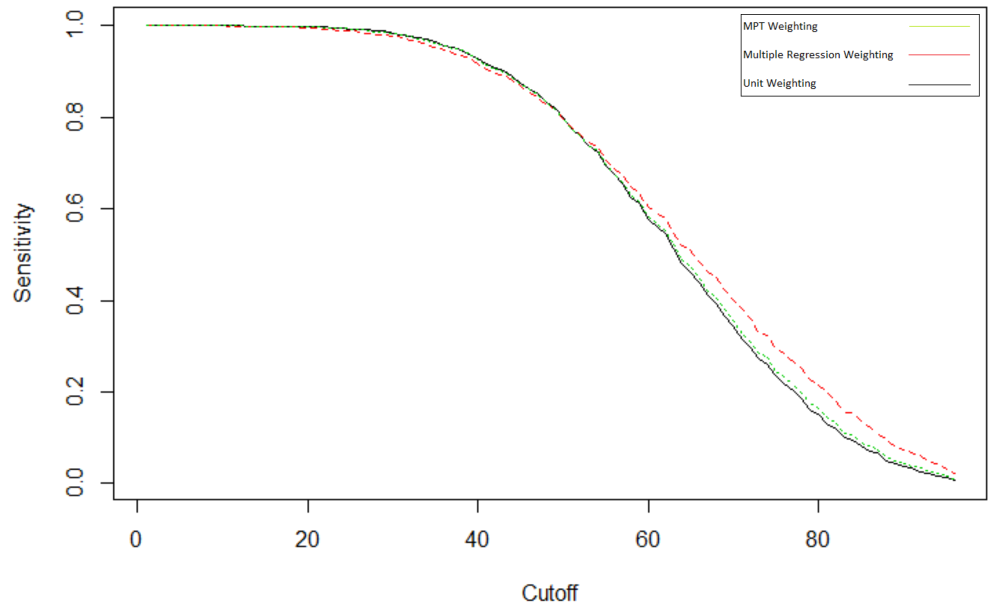

Figure 19: Sensitivity for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 50, multiple regression weighting produced a higher sensitivity.

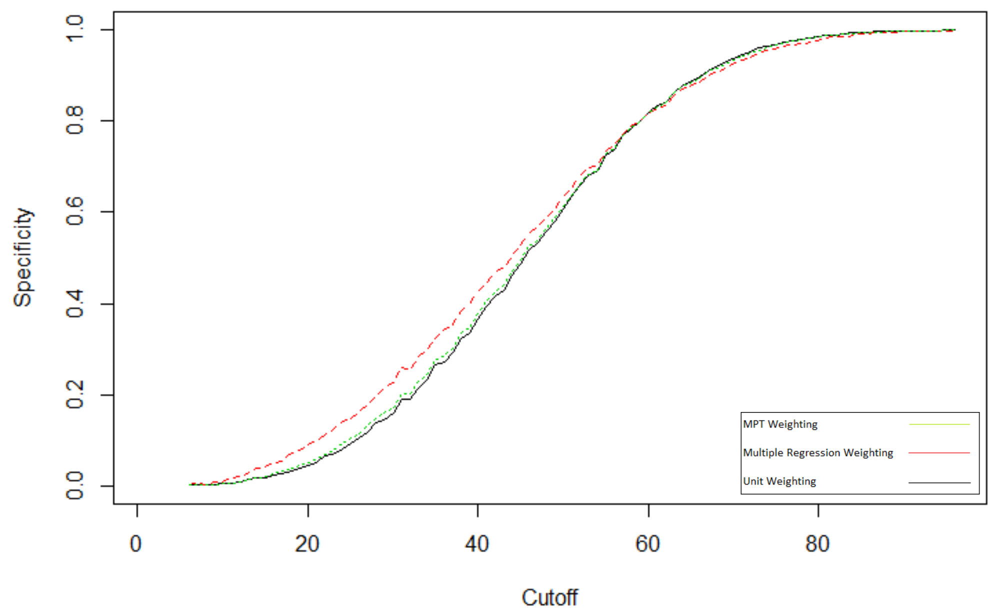

Figure 20: Specificity for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 60, multiple regression weighting produced a lower specificity.

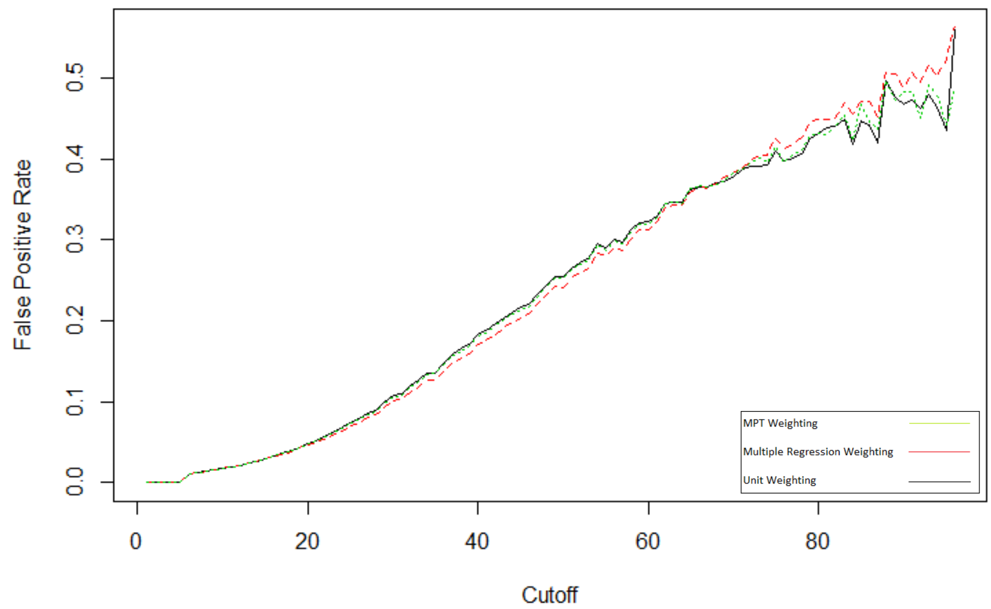

Figure 21: False Positive Rate for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 60, multiple regression weighting produced a higher false positive rate.

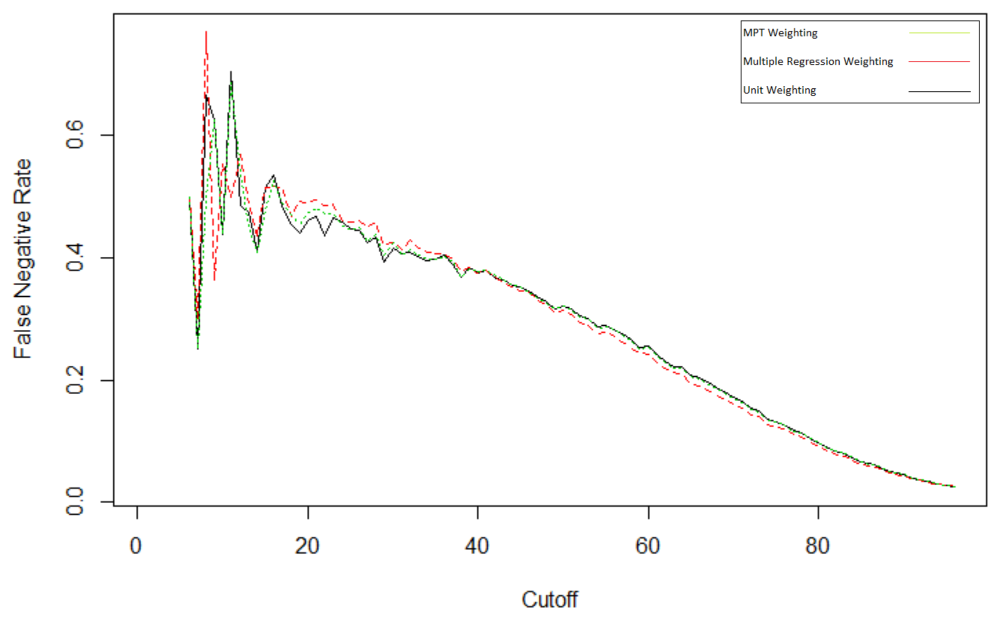

Figure 22: False Negative Rate for Different Cutoff Scores. This graph represents data generated from an applicant pool of 100 and an initial sample size of 100 accumulated over 1,000 trials for cutoff scores ranging from 1 to 100. For cutoff scores above 50, multiple regression weighting produced a lower false negative rate.

For each of the binary classification and binary classification performance measures, there appeared to be a cutoff score, between 50 and 60, where one method would begin to consistently have better results than another, albeit a very small advantage. A summary of which sets of cutoff scores are better for MPT weighting is presented in Table 5, below.

Table 5: Set of Approximate Cutoff Scores Where MPT was Observed to Produce More of Each Outcome or Performance Measure. The second column shows which set of cutoff scores produce more of the outcomes or performance measures for MPT weighting above that of multiple regression weighting. The third column shows which set of cutoff scores produces desirable outcomes.

Measure	Condition Where MPT Weighting Produced More Outcomes or Higher Performance Measures	Condition Where MPT is the Preferred Method
Binary Decision Measures		
Correct Selections	< 50	< 50
Correct Rejections	> 60	> 60
False Positives	< 60	> 60
False Negatives	> 50	< 50
Binary Decision Performance Measures		
Sensitivity	< 50	< 50
Specificity	> 60	> 60
False Positive Rate	< 60	> 60
False Negative Rate	> 50	< 50

These data have implications for hypotheses 1 and 4. Overall, these data support Hypothesis 4. There are observed cutoff scores where one method will consistently outperform the others. In addition, Hypothesis 1a and Hypothesis 1b become partially supported, but only when the cutoff score is above ~60. About this cutoff score MPT weighting will produce more correct rejections and fewer false positives than multiple regression weighting.

The fourth simulation involved using a second correlations matrix, specified in the methods section, to test Hypothesis 5. Binary decision classifications, binary classification performance measures, and the weights produced by each weighting method are summarized in Table 6, below.

Table 6: Results of Simulation 4. For this simulation, the initial sample was held at 100, the number of applicants was held at 100, and the cutoff score was held at 60 on a 1-100 scale. The data in the tables represents the cumulative results of 10,000 trials. Unit Weighting, Multiple regression weighting, and MPT weighting performed about as well as each other in terms of binary decision classification measures and binary decision performance measures. MPT generated smaller weights for cognitive ability in this condition, supporting hypothesis 5.

	Unit Weighting	Multiple Regression Weighting	MPT Weighting
Binary Decision Measures			
Correct Selections	309069	310681	305614
Correct Rejections	385314	383752	381598
False Positives	141427	142989	145143
False Negatives	164190	162578	167645
Binary Decision Performance Measures			
Sensitivity	0.68	0.69	0.68
Specificity	0.74	0.75	0.74
False Positive Rate	0.30	0.29	0.30
False Negative Rate	0.28	0.27	0.28
Predictor Weights			
Conscientiousness	0.33	0.28	0.41
Cognitive Ability	0.33	0.40	0.29
Risk Taking	0.33	0.32	0.29

The data in Table 6 supports Hypothesis 5. The weight given to cognitive ability is reduced below both multiple regression weighting and unit weighting. Binary decision outcomes are similar, with differences between methods, for all categories, at less than 2% of the totals, and binary classification performance measures are all within 1% of each other.

Discussion

Hypothesis 1a and 1b were supported for cases where the cutoff score was above ~60, or when the cutoff selects 45% of the applicants or less. MPT will produce fewer false positives and more correct rejections than multiple regression weighting. If these two outcomes are of concern, and the selection ratio is lower than 45%, the case can be made to use MPT weighting. The decision to use MPT weighting must then be coupled with a utility analysis that compares the relative utility of all four binary classification outcomes. When false positives and correct rejections are sufficiently costly, and false negatives and correct selections are less costly and beneficial, respectively, MPT weighting should be used.

Hypothesis 2 was also supported, making the case that MPT weighting did not differ greatly from multiple regression weighting in terms of accuracy. In fact, specificity and the false positive rate were slightly better for MPT weighting when the cutoff score was above ~60. Again, the discussion is incomplete without attaching utility to these performance measures. In practice, the utility of these outcomes would be specific to the job and organization. In general, and for almost all cutoff scores, MPT and multiple regression weighting were very close on these measures. Accuracy, in this case, is not a major differentiator between any of the three methods.

Hypothesis 3 specifically observed how a smaller initial sample to generate weights can reduce the quality of prediction for the two sample dependent models. Unit weighting was the best method in this circumstance, but choosing the predictors used for unit weighting will still require an *a priori* sample dependent approach. MPT weighting appeared to be a more robust method when sample size was below ~30, and it outperformed multiple regression on all outcome and performance measures. Unit weighting appear to be more successful when sample size was below ~30 as well. This is inconsistent with Schmidt's findings that place the threshold for differences at ~100 (1971). It is consistent with the evidence which points to another condition where MPT would be justified to generate predictor weights over multiple regression.

Hypothesis 4 was concerned with establishing a cutoff score where MPT would outperform multiple regression. This hypothesis was partially supported in that cutoff scores for individual outcome measures and performance measures each have an associated cutoff. Unfortunately, there was no distinct set of cutoff scores where all of these measures were better for MPT weighting than for multiple regression weighting. An organization must be able to prioritize outcome measures based on the utility of those outcomes to decide which method is most appropriate. The classification outcome measures and performance measures had two different cutoff scores where the dominance of one method over another changed. For false positives, correct rejections, specificity, and false positive rate, MPT performed better when the cutoff score was above ~60. For correct selections, false negatives, sensitivity, and false negative rate, MPT performed better when the cutoff was below ~60. It is unknown why the cutoff scores are different for these two categories.

Hypothesis 5 involved a new hypothetical set of predictors. Like combinations of stocks and bond in the financial application of MPT, negative covariance between the predictors helps to reduce the total variance of the model. Greater weights are given to the predictors that are negatively correlated with each other to balance the return to risk minimization problem. This has a side-effect of reducing the weight given to a third, unrelated predictor. The specific case of choosing cognitive ability, and two negatively correlated predictors, conscientiousness and risk-taking, produced the same patterns of output as the first three simulations, with little difference between methods in outcome measures or performance. However, the weight given to cognitive ability was much less in MPT weighting, below both multiple regression weighting and unit weighting. Using cognitive ability has been shown to lead to adverse impact for Blacks and Hispanics (Ployhart and Holtz 2008). Reducing the weight given to cognitive ability while maintaining the quality of prediction is extremely useful to promote diversity and reduce the risk of an adverse impact lawsuit.

Applications

These findings provide evidence that MPT weighting can be used to address issues of diversity with respect to protected groups, but other factors must also be considered. Much of the research regarding alternate weighting methods predates the civil rights act of 1964. Before that time, the rational approach, choosing weights based on expert opinions was much more widespread. After the act and a series of court cases, employers needed to be able to justify and validate their selection methods, and, for good reason, alternate methods for choosing predictor weights were riskier and much less popular. Additionally, prior to the Civil Rights Act of 1991, many other methods were available to address issues of diversity, making alternate weighting methods even less attractive to employers (1991). Making the argument for using MPT in the way described by Hypothesis 5 requires that the method meet criteria that fit the language and spirit of both the Civil Rights Act of 1964, Title VII, and the Civil Rights Act of 1991.

There are several strong utility-based justifications for choosing MTP related to the findings of Hypotheses 1-4. If MPT fits one or more of the criteria mentioned in these hypotheses, the procedure would be business justified. This is an important distinction, because many other methods for addressing diversity cannot satisfy this condition.

Given that is method is a mechanical combination, it avoids two major problems of the human judgement used in rational weighting. First, it is based on an algorithm that cannot be changed or influenced by prejudice or discrimination. Second, it does not suffer from overt or hidden biases that can hurt the quality of prediction. The only judgement involved is which predictors to include. Multiple regression can also use certain predictors for increasing the relative weight of a predictor in a model. These are called suppressor variables. Suppressor variables are correlated to the predictor, but not to the criterion. These variables are difficult to isolate but have the potential to produce similar effects as those described in simulation 4. The difficulty with recommending suppressor variables, is that they are unrelated to the criterion, meaning the predictor is not job-related. This is not the case for the MPT weighted predictors.

Limitations

The limitations of this research are mostly concerned with the fidelity of the simulated data. Of primary concern is the assumption of normality. It may be too simple to assume that performance is normally distributed, although we can impose a forced normal distribution. For the sake of argument, normality is also an assumption of multiple regression. In financial markets, returns are, most definitely, not normally distributed. Alternate methods that account for this have been developed to work around this problem. One of the most popular methods is Mean Absolute Deviation (MAD) portfolio optimization (Konno and Yamazaki, 1991).

Another limitation is that MPT weighting requires clearly and correctly established performance ranks. In some jobs performance can be much more objectively determined than others. Getting all employees correctly rated is very important to the structure of the model. This requires performance evaluations to be reliable and valid, which is difficult to put into practice. Again, these issues are also present with multiple regression weighting as well.

Real data may not act in ways that work as well for MPT. Until the method can be tested against a real data set, it is unknown what the actual differences between methods will look like. Real data has far more inconsistencies, outliers and anomalies than simulated data. Sampling from a large pool of simulated applicants helps to address this, but a real sample would be more telling.

Future Research

Future research for this topic falls within three basic categories. First is addressing limitations. The issues of normality can be addressed through using a forced distribution or by switching to a method like MAD portfolio optimization. Replacing the efficient frontier function in this simulation with a MAD function and changing the properties of the performance distribution would accomplish this effectively. Using real data rather than simulated data, is another avenue to explore. Because companies want to protect against the possibility of litigation, an archival study would be most appropriate. A researcher would need a data set where they could theoretically apply a higher cutoff than the original for the applicants to observe outcomes at and around that theoretical higher cutoff level.

The second recommendation for future research is an expansion of the questions that this model can address through simulation. How does MPT weighting perform with larger numbers of predictors? How does it perform with multiple pairings of negatively correlated predictors? Both questions can be immediately answered by changing the correlation matrix in the code found in the Appendix. How is MPT affected by range restriction at the predictor and/or criterion levels? This would require a simple subsetting of data in the performance and applicant functions. How much is adverse impact changed by the inclusion of negatively correlated predictors? This would require assigning a new variable indicating minority status and counting the outcomes for both groups, which would require more extensive reprogramming for this simulation.

The third area of future research regards what kinds of questions MPT can be used to address. Consider the issue of team performance. Can we use combinations of employees, to predict team performance? Can use MPT to address where to invest training time and money? Can we use MPT to budget HR resources toward the most effective combination of strategies? These questions can each be viewed through an MPT lens, and perhaps provide objective insight into these complex questions.

Summary

Overall, the results of the four simulations were favorable for recommending the use of MPT weighting. Together, the evidence supporting the hypotheses makes a case for using MPT weighting over multiple regression and unit weighting, in certain situations. The definitive answer as to which method is best, is deeply related to the utility of the decision-making outcomes that each method recommends. These decisions are job and organization-specific, and should be considered on a case-by case basis.

Findings from this study indicate that dependence on a measure that produces adverse impact can be reduced through a combination of careful predictor choice and MPT weighting. MPT weighting is suggested as a possible solution to adverse impact issues, although further research is needed, particularly studies involving real data. research is also needed to address additional questions regarding the robustness of the method given range restriction and the inclusion of additional predictors.

References

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (1999). *Standards for educational and psychological testing*. American Educational Research Association.

Barrick, M. R., & Mount, M. K. (1991). The Big Five Personality Dimensions and Job Performance: A Meta-Analysis. *Personnel Psychology*, *44*(1), 1-26. doi:10.1111/j.1744-6570.1991.tb00688.x

Bowman, M. L. (1989). Testing individual differences in ancient China. *American Psychologist, 44*(3), 576-578. doi:10.1037//0003-066x.44.3.576.b

Chandra, S., & Shadel, W. G. (2007). Crossing disciplinary boundaries: Applying financial portfolio theory to model the organization of the self-concept. *Journal of Research in Personality*, *41*(2), 346-373. doi:10.1016/j.jrp.2006.04.007

Civil Rights Act of 1991. (1991). Retrieved April 4, 2017, from https://www.eeoc.gov/eeoc/history/35th/thelaw/cra_1991.html

Conroy, M. E. (1974). Alternative Strategies for Regional Industrial Diversification*. *Journal of Regional Science J Regional Sci*, *14*(1), 31-46. doi:10.1111/j.1467-9787.1974.tb00427.x

Disparate Impact and Reasonable Factors Other Than Age Under the Age Discrimination in Employment Act. (2012, March 30). Retrieved from https://www.federalregister.gov/documents/2012/03/30/2012-5896/disparate-impact-and-reasonable-factors-other-than-age-under-the-age-discrimination-in-employment

Einhorn, H. J., & Hogarth, R. M. (1975). Unit weighting schemes for decision making. *Organizational Behavior and Human Performance, 13*(2), 171-192. doi:10.1016/0030-5073(75)90044-6

Gauss, C. F. (1823). *Theoria combinationis observationum erroribus minimis obnoxiae.-Gottingae, Henricus Dieterich 1823*. Henricus Dieterich.

Grove, W. M., Zald, D. H., Lebow, B. S., Snitz, B. E., & Nelson, C. (2000). Clinical versus mechanical prediction: A meta-analysis. *Psychological Assessment, 12*(1), 19-30. doi:10.1037//1040-3590.12.1.19

Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, 40(1), 1-29. URL http://www.jstatsoft.org/v40/i01/.

Hunter, J. E., & Hunter, R. F. (1984). Validity and utility of alternative predictors of job performance. *Psychological Bulletin, 96*(1), 72-98. doi:10.1037//0033-2909.96.1.72

Konno, H., & Yamazaki, H. (1991). Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management science, 37*(5), 519-531.

Lagrange, L. (1788). Mécanique analytique, Paris. *MJ Bertrand. Mallet-Bachelier.)[aPJHS]*.

Lawshe, C. H., & Schucker, R. E. (1959). The relative efficiency of four test weighting methods in multiple prediction. *Educational and psychological measurement*, *19*(1), 103-114.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance, 7*(1), 77. doi:10.2307/2975974

Markowitz, H. (1959). *Portfolio selection; efficient diversification of investments*. New York: Wiley.

Matuszak, A. (n.d.). Using R to build an optimized portfolio. Retrieved from http://economistatlarge.com/portfolio-theory/r-optimized-portfolio/r-code-graph-efficient-frontier

Meehl, P. E. (1954). *Clinical versus statistical prediction; a theoretical analysis and a review of the evidence*. Minneapolis: University of Minnesota Press.

Peterson, W. W., Birdsall, T. G. & Fox, W. C. (1954) The theory of signal detectability. Proceedings of the IRE Professional Group on Information Theory 4, 171-212.

Ployhart, R. E., & Holtz, B. C. (2008). The diversity–validity dilemma: Strategies for reducing racioethnic and sex subgroup differences and adverse impact in selection. *Personnel Psychology*, *61*(1), 153-172.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

S original by Berwin A. Turlach R port by Andreas Weingessel <Andreas.Weingessel@ci.tuwien.ac.at> (2013). quadprog: Functions to solve Quadratic Programming Problems. R package version 1.5-5. https://CRAN.R-project.org/package=quadprog

Sackett PR, Ellingson JE. 1997. The effects of forming multi-predictor composites on group differences and adverse impact. Pers. Psychol. 50:707–21

Schmidt, F. L. (1971). The relative efficiency of regression and simple unit predictor weights in applied differential psychology. *Educational and Psychological Measurement*, *31*(3), 699-714.

Schmidt, F. L. (1988). The problem of group differences in ability test scores in employment selection. *Journal of Vocational Behavior*, *33*(3), 272-292.

Sharpe, W. F. (1966). Mutual Fund Performance. *The Journal of Business J BUS*, *39*(S1), 119. doi:10.1086/294846

Sharpe, W. F. (1975). Adjusting for Risk in Portfolio Performance Measurement. *The Journal of Portfolio Management Portfolio Management*, *1*(2), 29-34. doi:10.3905/jpm.1975.408513

Society for Industrial, Organizational Psychology (US), & American Psychological Association. Division of Industrial-Organizational Psychology. (2003). *Principles for the validation and use of personnel selection procedures*. The Society.

Taylor, H. C., & Russell, J. T. (1939). The relationship of validity coefficients to the practical effectiveness of tests in selection: Discussion and tables. *Journal of applied psychology*, *23*(5), 565-578.

Tett, R. P., Jackson, D. N., & Rothstein, M. (2006). Personality Measures As Predictors Of Job Performance: A Meta-Analytic Review. *Personnel Psychology, 44*(4), 703-742. doi:10.1111/j.1744-6570.1991.tb00696.x

Trattner, M. H. (1963). Comparison of three methods for assembling aptitude test batteries. *Personnel Psychology*, *16*(3), 221-232.

Truxillo, D. M., & Bauer, T. N. (1999). Applicant reactions to test scores banding in entry-level and promotional contexts. *Journal of Applied Psychology*, *84*(3), 322.

Uniform guidelines on employee selection procedures. (1979). Washington, D.C.: Bureau of National Affairs.

Vinchur, A. J. (2007). A history of psychology applied to employee selection. *Historical perspectives in industrial and organizational psychology*, 193-218.

Appendix

```
library(ggplot2) # Used to graph efficient frontier, optional
library(quadprog) # Needed for solve.QP
library(plyr)
eff.frontier = function (returns, short = "no", max.allocation = NULL,
                         risk.premium.up=.5, risk.increment=.001, dimx = NULL,
                         dimlength = NULL)
{
  covariance = cov(returns)
  #print(covariance)
  n = ncol(covariance)
  Amat = matrix (1, nrow=n)
  bvec = 1
  meq = 1
  if(short == "no"){
    Amat = cbind(1, diag(n))
    bvec = c(bvec, rep(0, n))
  }
  if(!is.null(max.allocation)){
    if(max.allocation > 1 | max.allocation <0){
      stop("max.allocation must be greater than 0 and less than 1")
    }
    if(max.allocation * n < 1){
      stop("Need to set max.allocation higher; not enough assets to add to 1")
    }
    Amat = cbind(Amat, -diag(n))
    bvec = c(bvec, rep(-max.allocation, n))
  }
```

```
  loops = risk.premium.up / risk.increment + 1
  loop = 1
  eff = matrix(nrow=loops, ncol=n+3)
  colnames(eff) = c(dimx[2:dimlength], "Std.Dev", "Exp.Return", "sharpe")
  # Loop through the quadratic program solver
  for (i in seq(from=0, to=risk.premium.up, by=risk.increment)){
    dvec = colMeans(returns) * i # This moves the solution along the EF
    sol = solve.QP(covariance, dvec=dvec, Amat=Amat, bvec=bvec, meq=meq)
    eff[loop,"Std.Dev"] = sqrt(sum(sol$solution*colSums((covariance*sol$solution))))
    eff[loop,"Exp.Return"] = as.numeric(sol$solution %*% colMeans(returns))
    eff[loop,"sharpe"] = eff[loop,"Exp.Return"] / eff[loop,"Std.Dev"]
    eff[loop,1:n] = sol$solution
    loop = loop+1
  }
  return(as.data.frame(eff))
}
regression.model = function(dimx = NULL,dimlength = NULL, newX = NULL)
{
  regmodelpred = dimx[2]
  for(i in 3:dimlength)
  {
    regmodelpred = paste(regmodelpred,"+",dimx[i])
    next
  }
  regmodelpred = paste(regmodelpred,"-1")
  RegModel = lm(as.formula(paste(dimx[1],"~",regmodelpred)), data = newX)
  Regression.Weighting =
    as.matrix(coefficients(RegModel))/(sum(coefficients(RegModel)))
```

```
  RegWeights = as.data.frame(t(Regression.Weighting),
                             row.names = "Regression.Weighting")
  return(RegWeights)
}
unit.model = function(dimlength = NULL, dimx = NULL)
{
  Unit.Weighting = as.matrix(rep((1/(dimlength-1)),(dimlength-1)))
  UnitWeights = t(data.frame(Unit.Weighting, row.names = dimx[2:dimlength]))
  return(UnitWeights)
}
applicants = function(PerfMin = NULL, PerfMax = NULL, dimx = NULL, dimlength = NULL,
                      R = NULL, numapps = NULL, SEED = NULL)
{
  U = t(chol(R)) #Creates the Cholesky decomposition of the correlation matrix
  #This next section creates a random data set
  nvars = dim(U)[1]  #number of variables
  set.seed(SEED+1)  #Choose a seed to generate random numbers
  random.normal = matrix(rnorm(nvars*numapps*100,0,1), nrow=nvars, ncol=numapps*100);
  j = sample(1:100*numapps, numapps)
  random.normal = random.normal[,j]
  X = U %*% random.normal
  newX = as.data.frame(t(X))
  Xranked = newX[order(newX$Performance),]
  #rescale the predictors
  Xrescaled = (apply(Xranked, MARGIN = 2,
                     FUN = function(X) (X - min(X))/diff(range(X))))*(PerfMax-PerfMin)+
                    (matrix(rep(9,numapps*dimlength),ncol = dimlength))
  apps.mat = c(Xrescaled, newX)
```

```
  return(apps.mat)
  ### Plot the correlations
  # raw = as.data.frame(newX)
  # names(raw) = dimx
  # cor(raw)
  # plot(head(raw, 100)) #plot the corelations
}
performance = function(PerfMin = NULL, PerfMax = NULL, dimx = NULL,
                        dimlength = NULL, R = NULL, numobs = NULL, SEED = NULL)
{
  U = t(chol(R)) #Creates the Cholesky decomposition of the correlation matrix
  #This next section creates a a random data set
  nvars = dim(U)[1]  #number of variables
  set.seed(SEED)  #Choose a seed to generate random numbers
  random.normal = matrix(rnorm(nvars*numobs,0,1), nrow=nvars, ncol=numobs);
  X = U %*% random.normal
  newX = as.data.frame(t(X)) #transpose the result
  Xranked = newX[order(newX$Performance),] #rank orders the results by performance
  #rescale the predictors
  Xrescaled = (apply(Xranked, MARGIN = 2,
                    FUN = function(X) (X - min(X))/diff(range(X))))*(PerfMax-PerfMin)+
                   (matrix(rep(9,numobs*dimlength),ncol = dimlength))
  performance.mat = c(Xrescaled, newX)
  return(performance.mat)
  ### Plot the correlations
  # raw = as.data.frame(newX)
  # names(raw) = dimx
  # cor(raw)
```

```
  # plot(head(raw, 100)) #plot the corelations
}
get.returns = function(Xrescaled = NULL, dimlength = NULL)
{
  Xpred = Xrescaled[,2:dimlength] #Removes the performance variable
  returns = (tail(Xpred,-1) - head(Xpred,-1))
  return(returns)
  #hist(returns)
}
## Diagnostic/Graphic
# graph.eff = function(eff = NULL)
# {
#   # Find the optimal portfolio
#   eff.optimal.point = eff[eff$sharpe==max(eff$sharpe),]
#   # graph efficient frontier
#   # Start with color scheme
#   ealred = "#7D110C"
#   ealtan = "#CDC4B6"
#   eallighttan = "#F7F6F0"
#   ealdark = "#423C30"
#   eff.plot = ggplot(eff, aes(x=Std.Dev, y=Exp.Return)) + geom_point(alpha=.1,
#     color=ealdark) +
#     geom_point(data=eff.optimal.point, aes(x=Std.Dev, y=Exp.Return, label=sharpe),
#                color=ealred, size=5) +
#     annotate(geom="text", x=eff.optimal.point$Std.Dev,
#              y=eff.optimal.point$Exp.Return,
#              label=paste("Risk: ",
#                          round(eff.optimal.point$Std.Dev*100, digits=3),"\nReturn: ",
```

```
#                       round(eff.optimal.point$Exp.Return*100, digits=4),"%\nSharpe: ",
#                       round(eff.optimal.point$sharpe*100, digits=2), "%", sep=""),
#             hjust=1.5, vjust=2) +
#     ggtitle("Efficient Frontier\nand Optimal Portfolio") +
#     labs(x="Risk (standard deviation of portfolio)", y="Return") +
#     theme(panel.background=element_rect(fill=eallighttan),
#           text=element_text(color=ealdark),
#           plot.title=element_text(size=24, color=ealred))
#   ggsave("Efficient Frontier.png")
#   return(eff.plot)
# }
MPT.model = function(eff = NULL, dimx = NULL, dimlength = NULL)
{
  EFFSolutionsHead = head(eff)
  MPTWeights = EFFSolutionsHead[1,1:(dimlength -1)]
  rownames(MPTWeights) = "MPT.Weighting"
  #MPTRetAndSD = (EFFSolutionsHead[1, dimlength:(dimlength+1)])
  # print(MPTRetAndSD)
  return(MPTWeights)
}
SignalDetection = function(dimx = NULL, Perf = NULL, PerfMin = NULL, PerfMax = NULL,
                        Signal = NULL, Cutoff = NULL, numapps = NULL)
{
  Serf = Perf
  for(i in 1:numapps)
  {
    for(x in 2:4)
      {
```

```
        if((Perf[i,x] > Cutoff) & (Perf[i,1] > Cutoff)) Serf[i,x] = 1
        if((Perf[i,x] < Cutoff) & (Perf[i,1] < Cutoff)) Serf[i,x] = 2
        if((Perf[i,x] > Cutoff) & (Perf[i,1] < Cutoff)) Serf[i,x] = 3
        if((Perf[i,x] < Cutoff) & (Perf[i,1] > Cutoff)) Serf[i,x] = 4
      }
  }
  Signal = (Serf[,2:4])
  return(Signal)
}
#########Simulation 1##########################################################
Main = function(SEED = NULL, PerfMin = NULL, PerfMax = NULL, Cutoff = NULL,
                Weighting = NULL, dimx = NULL, dimlength = NULL,
                R = NULL, numobs = NULL, numapps = NULL, Trials = NULL,
                SigBox = NULL, Results = NULL, varylab = NULL)
{
for(j in (PerfMin+dimlength):PerfMax)
{
assign(varylab,j)
for(i in 1:Trials)
{
SEED = SEED + 1
performance.mat = performance(PerfMin = PerfMin,PerfMax = PerfMax, dimx = dimx,
                    dimlength = dimlength, R = R, numobs = numobs, SEED = SEED)
eff = eff.frontier(returns = get.returns(Xrescaled =
                    matrix(as.numeric(performance.mat[1:(numobs*dimlength)]),
                    byrow = FALSE, nrow = numobs, ncol = dimlength, dimnames =
                    list(c(1:numobs),dimx=dimx)),
                    dimlength = dimlength),dimx = dimx, dimlength = dimlength)
```

```
# print(graph.eff(eff = eff)) #diagnostic/graphic
apps.mat = applicants(PerfMin= PerfMin, PerfMax =PerfMax, dimx = dimx, dimlength =
        dimlength, R = R, numapps = numapps, SEED = SEED)
if(is.null(Weighting) == TRUE) Weighting = rbind(UnitWeights =
    as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
    RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
    newX = as.data.frame(do.call(cbind, performance.mat[
    (dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
    MPT.model(eff = eff, dimx = dimx, dimlength = dimlength))
if(is.null(Weighting) == FALSE) Weighting = rbind(Weighting,(rbind(UnitWeights =
    as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
    RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
    newX = as.data.frame(do.call(cbind, performance.mat[
    (dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
    MPT.model(eff = eff, dimlength = dimlength, dimx = dimx))))
#print(Weighting) #diagnostic
Signal = SignalDetection(Perf = cbind(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
    byrow = FALSE, nrow = numapps, ncol = dimlength, dimnames =
    list(c(1:numapps), dimx))[,1],t(as.matrix(Weighting[
    (3*(i-1)+1):(3*(i-1)+3),])%*%t(as.matrix(matrix(as.numeric(apps.mat[
    1:(numapps*dimlength)]), byrow = FALSE, nrow = numapps, ncol =
    dimlength, dimnames = list(c(1:numapps),dimx))[,2:dimlength])))),
    PerfMin = PerfMin, PerfMax = PerfMax, Signal = NULL, Cutoff =
    Cutoff, numapps = numapps)
# print(Signal)
SigTot = cbind(table(factor(Signal[,1],lev = 1:4)),table(factor(Signal[,2],lev = 1:4)),
    table(factor(Signal[,3],lev = 1:4)))
colnames(SigTot) = c("Unit Weighting","Multiple Regression Weighting", "MPT Weighting")
```

```
if(is.null(SigBox) == FALSE) SigBox = rbind(SigBox, SigTot)
if(is.null(SigBox) == TRUE) SigBox = SigTot
options("scipen"= 100, "digits"= 2)
}
CorrectSelections = colSums(SigBox[seq(1, nrow(SigBox), 4),])
CorrectRejections = colSums(SigBox[seq(2, nrow(SigBox), 4),])
FalsePositives = colSums(SigBox[seq(3, nrow(SigBox), 4),])
FalseNegatives = colSums(SigBox[seq(4, nrow(SigBox), 4),])
SigBox = rbind(CorrectSelections, CorrectRejections, FalsePositives, FalseNegatives)
rownames(SigBox) = c("Correct Selections","Correct Rejections","False Positives",
        "False Negatives")
Sensitivity = SigBox[1,]/(SigBox[1,]+SigBox[4,])
Specificity = SigBox[2,]/(SigBox[2,]+SigBox[3,])
FalsePositiveRate = SigBox[3,]/(SigBox[1,]+SigBox[3,])
FalseNegativeRate = SigBox[4,]/(SigBox[4,]+SigBox[2,])
WeightingAVG = rbind(colMeans(Weighting[seq(1,ncol(Weighting),3),]),colMeans(Weighting[
        seq(2,ncol(Weighting),3),]), colMeans(Weighting[
        seq(3,ncol(Weighting),3),]))
SigBox = rbind(SigBox,Sensitivity, Specificity, FalsePositiveRate,
        FalseNegativeRate,t(WeightingAVG))
if(is.null(Results) == FALSE) Results = rbind(Results, SigBox)
if(is.null(Results) == TRUE) Results = SigBox
SigBox = NULL
Weighting = NULL
}
return(Results)
}
Results = Main(SEED = 8675309, PerfMin = 1, PerfMax = 100, Cutoff = 60,
```

```
 Weighting = NULL, dimx = c("Performance", "Conscientiousness",
"Cognitive.Ability","Work.Sample"), dimlength = 4,
 R = matrix(cbind(  1,  .20,  .51, .57,
                   .20,  1,  .01, .09,
                   .51,  .01,  1, .34,
                   .57,  .09,  .34, 1), nrow = 4,
dimnames = list(c("Performance", "Conscientiousness",
"Cognitive.Ability","Work.Sample"), c("Performance", "Conscientiousness",
"Cognitive.Ability","Work.Sample"))),
 numobs = 100,   # number of observations
 numapps = 100,  #number of applicants
 Trials = 10000,  # number of times to test each level
 SigBox = NULL, Results = NULL, varylab = "Cutoff")
varylab = "Cutoff"
matplot(Results[seq(1,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Correct Selections", main = "Correct Selections")
matplot(Results[seq(2,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Correct Rejections", main = "Correct Rejections")
matplot(Results[seq(3,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "False Positives", main = "False Positives")
matplot(Results[seq(4,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "False Negatives", main = "False Negatives")
matplot(Results[seq(5,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Sensitivity", main = "Sensitivity")
matplot(Results[seq(6,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Specificity", main = "Specificity")
matplot(Results[seq(7,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "False Positive Rate", main = "False Positive Rate")
```

```
matplot(Results[seq(8,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "False Negative Rate", main = "False Negative Rate")
matplot(Results[seq(9,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Weights", main = "Conscientiousness")
matplot(Results[seq(10,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Weights", main = "Cognitive Ability")
matplot(Results[seq(11,nrow(Results),11),], type ="l", xlab = varylab, ylab =
    "Weights", main = "Work Sample")
##############Simulation
2###########################################################
Main = function(SEED = NULL, PerfMin = NULL, PerfMax = NULL, Cutoff = NULL,
        Weighting = NULL, dimx = NULL, dimlength = NULL,
        R = NULL,numobsmax = NULL,numobs = NULL, numapps = NULL,
        Trials = NULL, SigBox = NULL, Results = NULL, varylab = NULL)
{
 for(j in 10:numobsmax)
 {
  numobs = j
  for(i in 1:Trials)
  {
   SEED = SEED + 1
   performance.mat = performance(PerfMin = PerfMin,PerfMax = PerfMax, dimx = dimx,
                   dimlength = dimlength, R = R, numobs = numobs,
                   SEED = SEED)
   eff = eff.frontier(returns = get.returns(Xrescaled =
                   matrix(as.numeric(performance.mat[1:(numobs*dimlength)]),
                   byrow = FALSE, nrow = numobs, ncol = dimlength,
                   dimnames = list(c(1:numobs),dimx=dimx)), dimlength =
                   dimlength),dimx = dimx, dimlength = dimlength)
```

```
# print(graph.eff(eff = eff)) #diagnostic/graphic
apps.mat = applicants(PerfMin= PerfMin, PerfMax =PerfMax, dimx = dimx,
                      dimlength = dimlength, R, numapps, SEED)
if(is.null(Weighting) == TRUE) Weighting =
  rbind(UnitWeights = as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
  RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
  newX = as.data.frame(do.call(cbind,
  performance.mat[(dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
  MPT.model(eff = eff, dimx = dimx, dimlength = dimlength))
if(is.null(Weighting) == FALSE) Weighting = rbind(Weighting,(rbind(UnitWeights =
  as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
  RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
  newX = as.data.frame(do.call(cbind,
  performance.mat[(dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
  MPT.model(eff = eff, dimlength = dimlength, dimx = dimx))))
#print(Weighting) #diagnostic
Signal = SignalDetection(Perf =
  cbind(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
  byrow = FALSE, nrow = numapps, ncol = dimlength, dimnames = list(c(1:numapps),
  dimx))[,1],t(as.matrix(Weighting[(3*(i-1)+1):(3*(i-1)+3),])%*%
  t(as.matrix(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
  byrow = FALSE, nrow = numapps, ncol = dimlength,
  dimnames = list(c(1:numapps),dimx))[,2:dimlength])))),
  PerfMin = PerfMin, PerfMax = PerfMax, Signal = NULL, Cutoff = Cutoff,
  numapps = numapps)
# print(Signal)
SigTot = cbind(table(factor(Signal[,1],lev = 1:4)),table(factor(Signal[,2],
                     lev = 1:4)),table(factor(Signal[,3],lev = 1:4)))
```

```
    colnames(SigTot) = c("Unit Weighting","Multiple Regression Weighting",
                    "MPT Weighting")
    if(is.null(SigBox) == FALSE) SigBox = rbind(SigBox, SigTot)
    if(is.null(SigBox) == TRUE) SigBox = SigTot
    options("scipen"= 100, "digits"= 2)
  }
  CorrectSelections = colSums(SigBox[seq(1, nrow(SigBox), 4),])
  CorrectRejections = colSums(SigBox[seq(2, nrow(SigBox), 4),])
  FalsePositives = colSums(SigBox[seq(3, nrow(SigBox), 4),])
  FalseNegatives = colSums(SigBox[seq(4, nrow(SigBox), 4),])
  SigBox = rbind(CorrectSelections, CorrectRejections, FalsePositives,
            FalseNegatives)
  rownames(SigBox) = c("Correct Selections","Correct Rejections","False Positives",
                  "False Negatives")
  Sensitivity = SigBox[1,]/(SigBox[1,]+SigBox[4,])
  Specificity = SigBox[2,]/(SigBox[2,]+SigBox[3,])
  FalsePositiveRate = SigBox[3,]/(SigBox[1,]+SigBox[3,])
  FalseNegativeRate = SigBox[4,]/(SigBox[4,]+SigBox[2,])
  WeightingAVG = rbind(colMeans(Weighting[seq(1,ncol(Weighting),3),]),
                  colMeans(Weighting[seq(2,ncol(Weighting),3),]),
                  colMeans(Weighting[seq(3,ncol(Weighting),3),]))
  SigBox = rbind(SigBox,Sensitivity, Specificity, FalsePositiveRate,
            FalseNegativeRate,t(WeightingAVG))
  if(is.null(Results) == FALSE) Results = rbind(Results, SigBox)
  if(is.null(Results) == TRUE) Results = SigBox
  SigBox = NULL
  Weighting = NULL
}
```

```
  return(Results)
}
Results2 = Main(SEED = 8675309, PerfMin = 1, PerfMax = 100, Cutoff = 60,
        Weighting = NULL, dimx = c("Performance", "Conscientiousness",
        "Cognitive.Ability","Work.Sample"), dimlength = 4,
        R = matrix(cbind(  1,  .20,  .51, .57,
                    .20,  1,  .01, .09,
                    .51,  .01,  1, .34,
                    .57,  .09,  .34, 1), nrow = 4,
        dimnames = list(c("Performance", "Conscientiousness",
        "Cognitive.Ability","Work.Sample"), c("Performance", "Conscientiousness",
        "Cognitive.Ability","Work.Sample"))),
        numobs = 100,   # number of observations
        numapps = 100,  #number of applicants
        Trials = 1000,  # number of times to test each level
        SigBox = NULL, Results = NULL, varylab = "numobs", numobsmax = 200)
# print(SigBox)
# print(Weighting)

varylab = "Number of Cases Used to Generate Weights"
matplot(Results3[seq(1,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
      "Correct Selections", main = "Correct Selections")
matplot(Results3[seq(2,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
      "Correct Rejections", main = "Correct Rejections")
matplot(Results3[seq(3,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
      "False Positives", main = "False Positives")
matplot(Results3[seq(4,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
      "False Negatives", main = "False Negatives")
```

```
matplot(Results3[seq(5,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "Sensitivity", main = "Sensitivity")
matplot(Results3[seq(6,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "Specificity", main = "Specificity")
matplot(Results3[seq(7,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "False Positive Rate", main = "False Positive Rate")
matplot(Results3[seq(8,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "False Negative Rate", main = "False Negative Rate")
matplot(Results3[seq(9,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "Weights", main = "Conscientiousness")
matplot(Results3[seq(10,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "Weights", main = "Cognitive Ability")
matplot(Results3[seq(11,nrow(Results3),11),], type ="l", xlab = varylab, ylab =
        "Weights", main = "Work Sample")
############Simulation
3##########################################################
Main = function(SEED = NULL, PerfMin = NULL, PerfMax = NULL, Cutoff = NULL,
            Weighting = NULL, dimx = NULL, dimlength = NULL,
            R = NULL,numobsmax = NULL,numobs = NULL, numapps = NULL, Trials =
NULL,
            SigBox = NULL, Results = NULL, varylab = NULL)
{
   for(i in 1:Trials)
   {
     SEED = SEED + 1
     performance.mat = performance(PerfMin = PerfMin,PerfMax = PerfMax, dimx = dimx,
                  dimlength = dimlength,
                  R = R, numobs = numobs, SEED = SEED)
     eff = eff.frontier(returns = get.returns(Xrescaled =
```

```
            matrix(as.numeric(performance.mat[1:(numobs*dimlength)]),
            byrow = FALSE, nrow = numobs, ncol = dimlength, dimnames =
            list(c(1:numobs),dimx=dimx)), dimlength = dimlength),dimx =
            dimx, dimlength = dimlength)
# print(graph.eff(eff = eff)) #diagnostic/graphic
apps.mat = applicants(PerfMin= PerfMin, PerfMax =PerfMax, dimx = dimx, dimlength =
            dimlength, R, numapps, SEED)
if(is.null(Weighting) == TRUE) Weighting = rbind(UnitWeights =
        as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
        RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
        newX = as.data.frame(do.call(cbind,
        performance.mat[(dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
        MPT.model(eff = eff, dimx = dimx, dimlength = dimlength))
if(is.null(Weighting) == FALSE) Weighting = rbind(Weighting,(rbind(UnitWeights =
        as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
        RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
        newX = as.data.frame(do.call(cbind, performance.mat[
        (dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
        MPT.model(eff = eff, dimlength = dimlength, dimx = dimx))))
#print(Weighting) #diagnostic
Signal = SignalDetection(Perf = cbind(matrix(as.numeric(apps.mat[
        1:(numapps*dimlength)]), byrow = FALSE, nrow = numapps,
        ncol = dimlength, dimnames = list(c(1:numapps), dimx))[,1],
        t(as.matrix(Weighting[(3*(i-1)+1):(3*(i-1)+3),])%*%
        t(as.matrix(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
        byrow = FALSE, nrow = numapps, ncol = dimlength, dimnames =
        list(c(1:numapps),dimx))[,2:dimlength])))),
        PerfMin = PerfMin, PerfMax = PerfMax, Signal = NULL, Cutoff =
```

```
            Cutoff, numapps = numapps)
   # print(Signal)
   SigTot = cbind(table(factor(Signal[,1],lev = 1:4)),table(factor(Signal[,2],
            lev = 1:4)), table(factor(Signal[,3],lev = 1:4)))
   colnames(SigTot) = c("Unit Weighting","Multiple Regression Weighting",
            "MPT Weighting")
   if(is.null(SigBox) == FALSE) SigBox = rbind(SigBox, SigTot)
   if(is.null(SigBox) == TRUE) SigBox = SigTot
   options("scipen"= 100, "digits"= 2)
  }
  CorrectSelections = colSums(SigBox[seq(1, nrow(SigBox), 4),])
  CorrectRejections = colSums(SigBox[seq(2, nrow(SigBox), 4),])
  FalsePositives = colSums(SigBox[seq(3, nrow(SigBox), 4),])
  FalseNegatives = colSums(SigBox[seq(4, nrow(SigBox), 4),])
  SigBox = rbind(CorrectSelections, CorrectRejections, FalsePositives,
             FalseNegatives)
  rownames(SigBox) = c("Correct Selections","Correct Rejections","False Positives",
                 "False Negatives")
  Sensitivity = SigBox[1,]/(SigBox[1,]+SigBox[4,])
  Specificity = SigBox[2,]/(SigBox[2,]+SigBox[3,])
  FalsePositiveRate = SigBox[3,]/(SigBox[1,]+SigBox[3,])
  FalseNegativeRate = SigBox[4,]/(SigBox[4,]+SigBox[2,])
  WeightingAVG = rbind(colMeans(Weighting[seq(1,ncol(Weighting),3),]),
                 colMeans(Weighting[seq(2,ncol(Weighting),3),]),
                 colMeans(Weighting[seq(3,ncol(Weighting),3),]))
  SigBox = rbind(SigBox,Sensitivity, Specificity, FalsePositiveRate,
             FalseNegativeRate,t(WeightingAVG))
  if(is.null(Results) == FALSE) Results = rbind(Results, SigBox)
```

```
    if(is.null(Results) == TRUE) Results = SigBox
    SigBox = NULL
    Weighting = NULL
  return(Results)
}
Results3 = Main(SEED = 8675309, PerfMin = 1, PerfMax = 100, Cutoff = 60,
                Weighting = NULL, dimx = c("Performance", "Conscientiousness",
                "Cognitive.Ability","Work.Sample"), dimlength = 4,
                R = matrix(cbind(  1,  .20,  .51, .57,
                                  .20,  1,  .01, .09,
                                  .51,  .01,  1, .34,
                                  .57,  .09,  .34, 1), nrow = 4,
                dimnames = list(c("Performance", "Conscientiousness", "Cognitive.Ability",
                "Work.Sample"), c("Performance", "Conscientiousness", "Cognitive.Ability",
                "Work.Sample"))), numobs = 100,   # number of observations
                numapps = 100,  #number of applicants
                Trials = 1000,  # number of times to test each level
                SigBox = NULL, Results = NULL, varylab = "numobs", numobsmax = 300)
print(Results3)
################Simulation
4########################################################
Main = function(SEED = NULL, PerfMin = NULL, PerfMax = NULL, Cutoff = NULL,
                Weighting = NULL, dimx = NULL, dimlength = NULL, R = NULL,
                numobsmax = NULL, numobs = NULL, numapps = NULL, Trials = NULL,
                SigBox = NULL, Results = NULL, varylab = NULL)
{
  for(i in 1:Trials)
  {
    SEED = SEED + 1
```

```
performance.mat = performance(PerfMin = PerfMin,PerfMax = PerfMax, dimx = dimx,
            dimlength = dimlength,
            R = R, numobs = numobs, SEED = SEED)
eff = eff.frontier(returns = get.returns(Xrescaled =
            matrix(as.numeric(performance.mat[1:(numobs*dimlength)]),
            byrow = FALSE, nrow = numobs, ncol = dimlength,
            dimnames = list(c(1:numobs),dimx=dimx)),
            dimlength = dimlength),dimx = dimx, dimlength = dimlength)
# print(graph.eff(eff = eff)) #diagnostic/graphic
apps.mat = applicants(PerfMin= PerfMin, PerfMax =PerfMax, dimx = dimx,
        dimlength = dimlength, R, numapps, SEED)
if(is.null(Weighting) == TRUE) Weighting = rbind(UnitWeights =
        as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
        RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
        newX = as.data.frame(do.call(cbind, performance.mat[
        (dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
        MPT.model(eff = eff, dimx = dimx, dimlength = dimlength))
if(is.null(Weighting) == FALSE) Weighting = rbind(Weighting,(rbind(UnitWeights =
        as.data.frame(unit.model(dimlength = dimlength, dimx = dimx)),
        RegWeights = regression.model(dimx = dimx, dimlength = dimlength,
        newX = as.data.frame(do.call(cbind, performance.mat[
        (dimlength*numobs + 1):(numobs*dimlength +dimlength)]))),
        MPT.model(eff = eff, dimlength = dimlength, dimx = dimx))))
#print(Weighting) #diagnostic
Signal = SignalDetection(Perf = cbind(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
        byrow = FALSE, nrow = numapps, ncol = dimlength,
        dimnames = list(c(1:numapps),dimx))[,1],
        t(as.matrix(Weighting[(3*(i-1)+1):(3*(i-1)+3),])%*%
```

```
              t(as.matrix(matrix(as.numeric(apps.mat[1:(numapps*dimlength)]),
              byrow = FALSE, nrow = numapps, ncol = dimlength, dimnames =
              list(c(1:numapps),dimx))[,2:dimlength])))),
              PerfMin = PerfMin, PerfMax = PerfMax,
              Signal = NULL, Cutoff = Cutoff, numapps = numapps)
  # print(Signal)
  SigTot = cbind(table(factor(Signal[,1],lev = 1:4)),table(factor(Signal[,2],lev = 1:4)),
              table(factor(Signal[,3],lev = 1:4)))
  colnames(SigTot) = c("Unit Weighting","Multiple Regression Weighting", "MPT Weighting")
  if(is.null(SigBox) == FALSE) SigBox = rbind(SigBox, SigTot)
  if(is.null(SigBox) == TRUE) SigBox = SigTot
  options("scipen"= 100, "digits"= 2)
}
CorrectSelections = colSums(SigBox[seq(1, nrow(SigBox), 4),])
CorrectRejections = colSums(SigBox[seq(2, nrow(SigBox), 4),])
FalsePositives = colSums(SigBox[seq(3, nrow(SigBox), 4),])
FalseNegatives = colSums(SigBox[seq(4, nrow(SigBox), 4),])
SigBox = rbind(CorrectSelections, CorrectRejections, FalsePositives, FalseNegatives)
rownames(SigBox) = c("Correct Selections","Correct Rejections","False Positives",
            "False Negatives")
Sensitivity = SigBox[1,]/(SigBox[1,]+SigBox[4,])
Specificity = SigBox[2,]/(SigBox[2,]+SigBox[3,])
FalsePositiveRate = SigBox[3,]/(SigBox[1,]+SigBox[3,])
FalseNegativeRate = SigBox[4,]/(SigBox[4,]+SigBox[2,])
WeightingAVG = rbind(colMeans(Weighting[seq(1,ncol(Weighting),3),]),
            colMeans(Weighting[seq(2,ncol(Weighting),3),]),
            colMeans(Weighting[seq(3,ncol(Weighting),3),]))
SigBox = rbind(SigBox,Sensitivity, Specificity, FalsePositiveRate,
```

```
            FalseNegativeRate,t(WeightingAVG))
  if(is.null(Results) == FALSE) Results = rbind(Results, SigBox)
  if(is.null(Results) == TRUE) Results = SigBox
  SigBox = NULL
  Weighting = NULL
  return(Results)
}
Results4 = Main(SEED = 8675309, PerfMin = 1, PerfMax = 100, Cutoff = 60,
           Weighting = NULL, dimx = c("Performance", "Conscientiousness",
           "Cognitive.Ability","Work.Sample"), dimlength = 4,
           R = matrix(cbind(  1,  .20,  .51, .34,
                        .20,    1,  .01,-.25,
                        .51,  .01,    1, .22,
                        .34, -.25,  .22,   1), nrow=4,
           dimnames = list(c("Performance", "Conscientiousness",
           "Cognitive.Ability","Risk.Taking"), c("Performance", "Conscientiousness",
           "Cognitive.Ability","Risk.Taking"))),
           numobs = 100,   # number of observations
           numapps = 100,  #number of applicants
           Trials = 10000,  # number of times to test each level
           SigBox = NULL, Results = NULL, varylab = "numobs", numobsmax = 300)
print(Results4)
```