Journal of Mathematical Finance, 2016, 6, 105-126
Published Online February 2016 in SciRes. http://www.scirp.org/journal/jmf
http://dx.doi.org/10.4236/jmf.2016.61012

LPM Density Functions for the Computation of the SD Efficient Set

Fred Viole[1], David Nawrocki[2]

[1]OVVO Financial Systems, Holmdel, USA
[2]Villanova University, Villanova School of Business, Villanova, USA
Email: Fred.viole@ovvofinancialsystems.com, david.nawrocki@villanova.edu

Received 22 October 2015; accepted 23 February 2016; published 26 February 2016

Abstract

The equivalence between partial moments and stochastic dominance dates back to Bawa [1] and Fishburn [2]. We present a test for first, second, and third degree stochastic dominance between two variables using Lower Partial Moments. The results uphold Hadar and Russell's [3] original conclusions about the odd moments of preferred prospects. We recall Nawrocki's [4] research comparing Mean/Variance portfolios against the continuum of risk-averse investors using Lower Partial Moments. The excess skewness of the LPM portfolios clearly demonstrates the preference of positive skewness for risk-averse investors. Finally, we provide an algorithm for efficiently determining stochastic dominance efficient sets among large numbers of variables.

Keywords

Stochastic Dominance Algorithm, Lower Partial Moments, Expected Utility Theory, Prospect Theory, Upper Partial Moments

1. Introduction

Stochastic dominance (SD) is a very powerful risk analysis tool. It converts the probability distribution of an investment into a cumulative probability curve. Next, math analysis of the cumulative probability curve is used to determine if one investment is superior to another investment. Stochastic dominance has two major advantages: It works for all probability distributions and it includes all possible risk averse utility assumptions. Its major disadvantage is that an optimization algorithm for selecting stochastically dominant efficient portfolios has never been developed limiting it to a two-step process where 1) the SD analysis is run and 2) the resulting securities are run through a portfolio optimization process. This actually is not too great of an issue because in practice,

How to cite this paper: Viole, F. and Nawrocki, D. (2016) LPM Density Functions for the Computation of the SD Efficient Set. *Journal of Mathematical Finance*, **6**, 105-126. http://dx.doi.org/10.4236/jmf.2016.61012

stochastic dominance can be run on thousands of securities and portfolio optimization algorithms are limited to about 150 securities due to singularity errors.[1] In this paper, we propose using a stochastic dominance algorithm using lower partial moments for step (1) and then using mean-semivariance or UPM/LPM algorithm for step (2).[2]

The use of semivariance or lower partial moment analysis to approximate stochastic dominance has already been suggested. Bey [5] proposes a mean-semivariance algorithm to approximate the second degree stochastic dominance efficient portfolio sets. More importantly, Bawa [1] and Fishburn [2] provide proofs for the equivalence of LPM degrees greater than one to second degree stochastic dominance and for LPM degrees greater than two to third degree stochastic dominance.[3]

Since stochastic dominance is an analysis of cumulative distribution functions; only below target deviations are considered over the interval [−∞, target] with the target encompassing all values of *X*. This "for all *X*" target condition is directly responsible for the generalization to all possible risk averse utility assumptions.

This paper proposes an integration of stochastic dominance analysis and lower partial moment analysis by defining a stochastic dominance (SD) test via the Lower Partial Moments (LPM) of the investment's probability distribution. From this SD-LPM test, we can quickly and efficiently determine the SD efficient set and generate optimal portfolios from that reduced universe of securities for any objective function (Mean/Variance, UPM/LPM, etc.). Equation (1) represents the LPM of an investment,

$$LPM\left(n,h,X\right)=\frac{1}{T}\sum_{t=1}^{T}\left\{\max\left(h-X_t\right),0\right\}^n \tag{1}$$

where X_t is the observation of variable *X* at time *t*; *h* is the target from which to compute the lower deviation; and *n* is the loss aversion weight assigned to the lower deviation. The next section explains the equivalence between LPM and SD for First Degree Stochastic Dominance (FSD), Second Degree Stochastic Dominance (SSD) and Third Degree Stochastic Dominance (TSD). In Section 3 we discuss the methodology of the SD routines and the SD efficient set algorithm. The appendices contain R code, commentary on the R code and working examples. In these examples we note the effects of SD on the odd moments of the distributions, consistent with Hadar and Russell [1969] original conclusions.

2. Historical Partial Moment Equivalence & Tests

Bawa [1] provides a proof that the LPM measure is mathematically related to stochastic dominance for risk tolerance values (***n***) of 0, 1, and 2. Fishburn [2] demonstrates the equivalence of the LPM measure to stochastic dominance for all values of ***n*** > 0. He provides the general case via:

$$F_n\left(h\right)=\int_{-\infty}^{h}\left(h-X\right)^n\mathrm{d}F\left(X\right). \tag{2}$$

Fishburn [2] also shows that the ***n*** value includes all types of investor behavior. Furthermore, Porter [10] illustrates that except for cases of identical means and semivariances, if *X* dominates *Y* by second degree stochastic dominance then *X* dominates *Y* by the mean-target semivariance model. Building from the general case, let F_X denote the cumulative distribution function of variable *X* such that:

$$LPM\left(0,h,X\right)=F_X\left(h\right) \tag{3}$$

$$LPM\left(1,h,X\right)=\int_{-\infty}^{h}\left(h-X\right)\mathrm{d}F\left(X\right) \tag{4}$$

$$LPM\left(2,h,X\right)=\int_{-\infty}^{h}\left(h-X\right)^2\mathrm{d}F\left(X\right). \tag{5}$$

Therefore, Equation (3) will be the statistic used for FSD testing, Equation (4) will be the statistic used for SSD testing, and Equation (5) will be the statistic used for TSD testing. We note that Equation (3) is a probability whereby the target deviation is taken to the 0th degree. *The LPM degree* 0 *is simply the empirical cumulative*

[1] Private correspondence with Harry Markowitz on the limits to the critical line algorithm. Markowitz [6] also recommends geometric mean-semivariance as the appropriate MPT approach.

[2] See Nawrocki [7] for a review of lower partial moment (LPM) risk measures.

[3] Although Tehranian [8] and Helms, Jean, and Tehranian [9] suggested stochastic dominance degrees greater than three and the fact is that LPM exists for degrees greater than two, there doesn't seem to be strong support in the literature for SD degrees greater than three. If we extend the Fishburn [2] and Tehranian [8] analysis, then SD degree four is equivalent to LPM degrees greater than three.

distribution function (*CDF*) *of the distribution*. When considering how far a deviation is from its target, increasing the degree per Equation (4) will give a linear weighting. This is the area of the distribution to the left of the target, *h*. Finally, if an investor is more sensitive to below target observations, increasing the degree further will compensate this behavior. The LPM Degree 2 used in Equation (5) is equivalent to the semivariance statistic. *These equations must be used from every target in the return distribution to generalize for all risk-averse investor types.*

In a bid to infer SD from aggregate statistics (and skip the every target "for all *h*" inconvenience), others have suggested various CDF tests to approximate SD results. For example, Klecan *et al*. [11] proposes a difference in CDFs, per the Kolmogorov-Smirnoff (KS) test, further bolstered by Monte Carlo simulation for robustness. However, the KS test used does not prove FSD or SSD. The test statistic, $D = \max\left[F_X(h) - F_Y(h)\right]$ is deficient because it does not satisfy the "for all *h*" previously alluded to. FSD is a very stringent qualification, whereby one observation in which $F_X > F_Y$ nullifies *X* FSD *Y*. What you have is a statistically significant difference in CDFs, which unfortunately does not tell us anything about stochastic dominance. Of course Klecan *et al*. [11] realize this and alter their hypothesis to test for a variable that is not *stochastically maximal*, *i.e*. weak stochastic dominance holds for some pair of prospects within their observed set.

> "The set **A** is first-degree [resp., second-degree] maximal if no prospect in **A** is weakly stochastically dominated by another prospect in **A**. First-degree dominance implies second-degree dominance, and second-degree maximality implies first degree maximality." Klecan *et al*. [11].

Developing an efficient routine to determine SD was originally hampered by computing power coupled with arcane techniques. Computing power has increased dramatically since the 1970s, thus enabling portfolio sizes unattainable back then. Furthermore, the use of partial moments are a computationally efficient method of determining CDFs flexible enough for the entire distribution and extending themselves to area analysis required for the higher SD degrees.

> "Finally, in order to conduct tests of SSD efficiency, all the calculations required for the FSD test must be made regardless of whether one is interested in FSD results; for TSD experiments, all the calculations required for FSD and SSD tests must be made." Porter, Wart and Ferguson [12].

The use of LPMs in the SD test eliminates the calculation redundancy Porter *et al*. [12] noted above. In fact, TSD takes less time to run than FSD per our routines, the explanation of which follows.

2.1. SD Routines

A complete presentation of the R code and commentary on the modules is presented in the Appendix A and Appendix C of this paper. Here, we are providing a general overview of how each SD routine performs its task on individual securities or aggregate portfolios.

2.1.1. FSD

To determine FSD between two variables we combine the observations into a single vector. Next we rank the observations in ascending order. This is the target vector. Using the combined sorted observations we compare CDFs from each target using Equation (3).

If the $LPM\left(0, \text{target}_i, X\right) < LPM\left(0, \text{target}_i, Y\right)$ then we store the instance with a binary indicator into an output vector. If at any target the relationship fails, the routine is stopped and the result of "*No FSD Exists*" is returned. This has the benefit of avoiding all of the observations as soon as a violation is detected. Otherwise we continue through the rest of the target vector until all observations are exhausted, thus affirming "*X*FSD *Y*" existence.

In an added bid of efficiency we incorporate an additional output vector for *Y* such that it can be checked simultaneously for "*Y* FSD *X*".

2.1.2. SSD

To determine SSD the same initial combining and ranking procedure is performed on the variables. However, we are no longer comparing CDFs rather areas of the functions up to that specific target point. These areas are compared using Equation (4) which is the degree 1 LPM.

If the $LPM(1,\text{target}_i,X) < LPM(1,\text{target}_i,Y)$ then we store the instance with a binary indicator into an output vector. All of the benefits from the FSD routine are preserved.

2.1.3. TSD

Again, using the identical combined sorted observation vector (yet another efficiency since it does not have to be generated for each degree tested) the two variables of interest are compared with their squared areas below a target, or simply their semivariance. Equation (5) represents this consideration.

If the $LPM(2,\text{target}_i,X) < LPM(2,\text{target}_i,Y)$ then we store the instance with a binary indicator into an output vector for that specific target. All of the benefits from the FSD and SSD routines are preserved.

2.2. SD Efficient Set

We incorporate these SD routines into a "SD Efficient Set" algorithm. The specific R code is also provided in the Appendix C. A full discussion of the algorithm is in Appendix A using a data frame (a list of vectors of equal lengths in R) of security or portfolio returns, we are able to generate the SD efficient set for the desired degree and return a reduced data frame. A visualization of the problem is presented by borrowing a reference to Braid Theory, which is an abstract geometric theory studying the everyday braid concept.

Our first step is to rank the securities or portfolios in ascending order by their LPM from the maximum observation across all variables. Using the terms "Base" and "Challenger" from Porter *et al*. [12], our first ranked variable is our "Base". The next ranked variable is the "Challenger". We test whether "Base" SD "Challenger". If "Base" SD "Challenger", then "Challenger" is placed in a "Dominated Set" which is an output vector. If "Base" does not SD "Challenger" then "Challenger" is added to the "Base" vector. The next "Challenger" must be tested against all members of the current "Base" vector. If it fails but one, it is immediately placed in the "Dominated Set" and the next "Challenger" is selected. When all "Challengers" are exhausted, the final SD Efficient Set is simply the original data frame less the "Dominated Set".

2.3. SD Algorithm Empirical Results

The resulting SD Efficient data frame can then be easily implemented (within the same command line) into an optimization routine. We verify the output from our method versus the DOMIN1 routine originally presented in Porter *et al*. [12] with the third degree SD correction suggested by Bey, Burgess, and Kearns [13]. We obtained a monthly total return dataset of 67 stocks from the CRSP dataset for the period 2010 to 2014. Summary statistics are provided in Appendix B. We then used the DOMIN1 routine and our R code routine to generate the FSD, SSD, and TSD efficient sets. Indeed the same securities are selected from a 67 security universe. The results for the SSD and TSD analysis are presented in **Table 1**.

One potential question is whether a simple reward to semivariability (R/SV) ratio ranking would be equivalent to the LPM SD algorithm. **Table 1** results indicate the answer is no. We ranked all 67 companies by R/SV ratio with the last column in **Table 1** indicating the company's rank among the 67 companies. For the SSD results, there are 12 undominated companies and only 7 companies are ranked in the top 12 of the R/SV rankings (5 companies are ranked out of the top 12). For the TSD results, there are 9 undominated companies and only 6 companies are ranked in the top 9 of the R/SV rankings (3 are ranked out of the top 12).

3. Discussion

While stochastic dominance implies mean/Semivariance or LPM dominance, LPM dominance does not imply stochastic dominance because of the nature of aggregate statistics versus individual observations. This is analogous to the common phrase: "*While dependence implies correlation, correlation does not imply dependence*." The stringent criterion of stochastic dominance defies implication from aggregate distributional statistics.

Given LPM's ability to consider multiple investor preferences, from a practical standpoint, it seems the best candidate to proxy stochastic dominance. We demonstrated how skewness is evident under SSD and TSD. This is not surprising given Nawrocki's [4] analysis of LPM portfolios versus their M/V counterparts, whereby excess positive skewness was an artifact of all risk-averse investors.

We have also provided an algorithm for efficiently determining SD efficient portfolios. The use of lower partial moments is consistent with the procedure originally proposed by Porter *et al*. [12] for rankings by means and

Table 1. Second and third degree undominated assets from 67 stocks, monthly data from January 2010 to December 2014 using DOMIN1 and our R code. The mean return is a monthly return relative and the Semi Deviation is a monthly percent change. (a) Second degree (SSD) undominated assets; (b) Third degree (TSD) undominated assets.

(a)

R/SV Asset	Mean	Variance	Skewness	Kurtosis	Semi Deviation	R/SV	Rank
N P S Pharmaceuticals	1.05186	0.02573	0.53738	2.89342	7.36910	0.5372	15
Under Armour	1.04309	0.00858	0.47474	2.76997	3.64820	1.0591	2
Hain Celestial	1.03491	0.00481	0.24761	2.92033	2.95020	1.0909	1
Gartner Inc	1.02787	0.00382	0.07640	2.39715	2.83310	0.9035	4
Ross Stores	1.02764	0.00343	0.05922	2.26037	2.61250	0.9780	3
P P G Industries	1.02707	0.00356	0.28773	3.56232	2.79310	0.8925	5
Visa Inc.	1.02064	0.00303	-0.75621	5.66774	3.28380	0.5692	14
Lorillard Inc	1.02045	0.00396	1.23399	5.63855	2.65120	0.6856	8
Amgen Inc	1.01988	0.00279	0.05548	3.07874	2.76420	0.6545	9
O G E Energy	1.01469	0.00268	0.61313	5.26623	2.70910	0.4791	18
Waste Connection	1.01305	0.00168	-0.11832	2.11724	2.26070	0.5220	16
Procter & Gamble	1.01010	0.00131	0.45848	2.82344	1.81910	0.4971	17

(b)

R/SV Asset	Mean	Variance	Skewness	Kurtosis	Semi Deviation	R/SV	Rank
N P S Pharmaceuticals	1.05186	0.02573	0.53738	2.89342	7.36910	0.5372	15
Under Armour	1.04309	0.00858	0.47474	2.76997	3.64820	1.0591	2
Hain Celestial	1.03491	0.00481	0.24761	2.92033	2.95020	1.0909	1
Gartner Inc	1.02787	0.00382	0.07640	2.39715	2.83310	0.9035	4
Ross Stores	1.02764	0.00343	0.05922	2.26037	2.61250	0.9780	3
P P G Industries	1.02707	0.00356	0.28773	3.56232	2.79310	0.8925	5
Lorillard Inc	1.02045	0.00396	1.23399	5.63855	2.65120	0.6856	14
Waste Connection	1.01305	0.00168	-0.11832	2.11724	2.26070	0.5220	16
Procter & Gamble	1.01010	0.00131	0.45848	2.82344	1.81910	0.4971	17

The Bey-Kearns-Burgess [13] TSD Correction in DOMIN1 Has Been Activated. The Number of Check Points = 6; ITEST = 1; JTEST = 3.

variances. This is verified with a 67 security universe example providing identical outputs for SSD and TSD.

One potential weakness of any empirical risk analysis approach is estimation error. Both SD and LPM are nonparametric and do not require knowledge of the underlying probability function. In simulation tests that we have conducted, the LPM measures are less sensitive to estimation error than either the mean or variance no matter which distribution is assumed.

Again, one consideration that needs to be reiterated is these risk analysis tools are only for below target deviations. When considering the reflection effect from Prospect Theory, above target deviations will have their own investor preferences (risk seeking for gains and risk aversion with gains). These positive observations are only considered when the target approaches the maximum observation under the stochastic dominance test. Stochastic dominance therefore cannot reward the right tail in a manner commensurate with the means it penalizes the left tail observations. But, this was never its intended purpose given the underlying utility assumptions of Hadar and Russel [3], and later Whitmore [14]. However, the above target returns have been found to be important in the discussions by Kahneman and Tversky [15] for non-concave functions (Reverse S-shaped) and for S-Shaped functions by Levy and Levy [16] [17], Post and Levy [18], Baltussen, Post and van Vliet [19], andPost, van Vliet and Levy [20].

In order to generalize further, one would have to expand the analysis into an Upper Partial Moment/Lower Partial Moment (UPM/LPM) framework, capable of incorporating the often observed four-fold pattern of risk behavior identified in prospect theory and expected utility theory such as the UPM/LPM optimization model described by Viole and Nawrocki [21] [22] and Cumova and Nawrocki [23].

4. Conclusions

The close relationship between lower partial moments and stochastic dominance has been known since Porter [10], Bawa [1], Fishburn [2], and Bey [5]. This paper uses the known cumulative density function properties and utility function properties of lower partial moments to generate stochastic dominant efficient sets. We hope that this paper demonstrates how powerful LPM analysis potentially is for statistical/financial analysis (and by extension UPM/LPM analysis). An efficient algorithm to generate SD efficient sets is proposed and tested alongside with the Porter *et al*. [12]. DOMIN1 algorithm which includes the third degree SD correction is suggested by Bey *et al*. [13]. Both algorithms provided the same efficient sets for SSD and TSD for a sample 67 security universe. A description of the LPM SD algorithm is provided in Appendix A and the R-code for the algorithm is included in Appendix C.

Future research should extend the analysis to the use of UPM/LPM models which are superior to SD and LPM models for incorporating the full range of utility functions available with expected utility theory and prospect theory.

References

[1] Bawa, V.S. (1975) Optimal Rules for Ordering Uncertain Prospects. *Journal of Financial Economics*, **2**, 95-121.

[2] Fishburn, P. (1977) Mean-Risk Analysis with Risk Associated with Below-Target Returns. *American Economic Review*, **67**, 116-126.

[3] Hadar, J. and Russel, W.R. (1969) Rules for Ordering Uncertain Prospects. *The American Economic Review*, **59**, 25-34.

[4] Nawrocki, D. (1992) The Characteristics of Portfolios Selected by N-Degree Lower Partial Moment. *International Review of Financial Analysis*, **1**, 195-209. http://dx.doi.org/10.1016/1057-5219(92)90004-N

[5] Bey, R.P. (1979) Estimating the Optimal Stochastic Dominance Efficient Set with a Mean-Semivariance Algorithm. *Journal of Financial and Quantitative Analysis*, **14**, 1059-1070.

[6] Markowitz, H.M. (2010) Portfolio Theory: As I Still See It. *Annual Review Financial Economics*, **2**, 1-23. http://dx.doi.org/10.1146/annurev-financial-011110-134602

[7] Nawrocki, D. (1999) A Brief History of Downside Risk Measures. *The Journal of Investing*, **8**, 9-25. http://dx.doi.org/10.3905/joi.1999.319365

[8] Tehranian, H. (1980) Empirical Studies in Portfolio Performance Using Higher Degrees of Stochastic Dominance. *The Journal of Finance*, **35**, 159-171. http://dx.doi.org/10.1111/j.1540-6261.1980.tb03478.x

[9] Helms, B.P., Jean, W.H. and Tehranian, H. (1986) An Algorithm for NTH Degree Stochastic Dominance. *Applied Stochastic Models and Data Analysis*, **2**, 71-81. http://dx.doi.org/10.1002/asm.3150020107

[10] Porter, R.B. (1974) Semivariance and Stochastic Dominance: A Comparison. *The American Economic Review*, **64**, 200-204.

[11] Klecan, L., McFadden, R. and McFadden, D. (1991) A Robust Test for Stochastic Dominance. Working Paper, Department of Economics, MIT, Cambridge.

[12] Porter, R.B., Wart, J.R. and Ferguson, D.L. (1973) Efficient Algorithms for Conducting Stochastic Dominance Tests on Large Numbers of Portfolios. *Journal of Financial and Quantitative Analysis*, **8**, 71-81. http://dx.doi.org/10.2307/2329749

[13] Bey, R.P., Burgess, R.C. and Kearns, R.B. (1984) Moving Stochastic Dominance: An Alternative Method for Testing Market Efficiency. *Journal of Financial Research*, **7**, 185-196. http://dx.doi.org/10.1111/j.1475-6803.1984.tb00369.x

[14] Whitmore, G.A. (1970) Third-Degree Stochastic Dominance. *The American Economic Review*, **60**, 457-459.

[15] Kahneman, D. and Tversky, A. (1979) Prospect Theory: An Analysis of Decision under Risk. *Econometrica*: *Journal of the Econometric Society*, **47**, 263-291. http://dx.doi.org/10.2307/1914185

[16] Levy, H. and Levy, M. (2004) Prospect Theory and Mean-Variance Analysis. *The Review of Financial Studies*, **17**, 1015-1041. http://dx.doi.org/10.1093/rfs/hhg062

[17] Levy, H. and Levy, M. (2009) The Safety First Expected Utility Model: Experimental Evidence and Economic Implications. *Journal of Banking & Finance*, **33**, 1494-1506. http://dx.doi.org/10.1016/j.jbankfin.2009.02.014

[18] Post, T. and Levy, H. (2005) Does Risk Seeking Drive Stock Prices? A Stochastic Dominance Analysis of Aggregate Investor Preferences and Beliefs. *Review of Financial Studies*, **18**, 925-953. http://dx.doi.org/10.1093/rfs/hhi021

[19] Baltussen, G., Post, T. and Van Vliet, P. (2006) Violations of Cumulative Prospect Theory in Mixed Gambles with Moderate Probabilities. *Management Science*, **52**, 1288-1290. http://dx.doi.org/10.1287/mnsc.1050.0544

[20] Post, T., Van Vliet, P. and Levy, H. (2008) Risk Aversion and Skewness Preference. *Journal of Banking & Finance*, **32**, 1178-1187. http://dx.doi.org/10.1016/j.jbankfin.2006.02.008

[21] Viole, F. and Nawrocki, D. (2011) The Utility of Wealth in an Upper- and Lower-Partial Moment Fabric. *The Journal of Investing*, **20**, 58-85. http://dx.doi.org/10.3905/joi.2011.20.2.058

[22] Viole, F. and Nawrocki, D. (2013) An Analysis of Heterogeneous Utility Benchmarks in a Zero Return Environment. *International Review of Financial Analysis*, **28**, 190-198. http://dx.doi.org/10.1016/j.irfa.2013.02.014

[23] Cumova, D. and Nawrocki, D. (2014) Portfolio Optimization in an Upside Potential and Downside Risk Framework. *Journal of Economics and Business*, **71**, 68-89. http://dx.doi.org/10.1016/j.jeconbus.2013.08.001

Appendix A

A.1. R Code Commentary

This section of the paper will provide the R code commentary for the stochastic dominance test using lower partial moments. The code can be found in Appendix C.

1) Module 1: Lower Partial Moment Function (LPM)

The LPM function is a fairly straightforward interpretation of Equation (1) whereby only below target observations are summed and raised to the loss aversion degree (*n*), and then divided by the number of observations.

2) Module 2: First Degree Stochastic Dominance (FSD)

Section A: Sort the variables in ascending order. Combine the vectors and sort the combined vector. Create output vectors for areas used in plots.

B: Create an output vector to store the instances of CDF inequality.

C: Uses the sorted *X* and *Y* variables as the LPM target under ***n*** = 0. Thus all observations are used in the stringent "for all *h*" qualification. If $F_X(h) < F_Y(h)$ for that observation target *h*, no instance is recorded into the output vector (output_x[i] < −0). Conversely, if $F_X(h) > F_Y(h)$ for that observation target *h*, the loop is stopped.

D: Uses the same logic as (C) above, only tests the inverse CDF relationship $F_Y(h) > F_X(h)$.

E: Plots the CDFs of each variable.

F: Reads the output vectors. If the output vector has 0 instances of $F_X(h) > F_Y(h)$, then *X* FSD *Y*.

G: Test case.

Figure A1 plots the cumulative distributions for all observations of *X* and *Y*. The combined sorted vector doubles the length of observations, and thus distorts the image compared to the familiar CDF as plotted in **Figure A2** for each variable.

We can see in **Figure A1** & **Figure A2** the multiple crossing of the CDFs (around the 300th observation and −1 value respectively), thus negating any first degree stochastic dominance between variables. Clearly though, on balance we can see $F(X)$ is more desirable than $F(Y)$.

For three ***h*** values (−0.99, −0.98, and −0.97) $F_X(h) > F_Y(h)$. Outside of those values, $F_X(h) < F_Y(h)$. This is where second degree stochastic dominance may offer some insight…

3) Module 3: Second Degree Stochastic Dominance (SSD)

Section A: Sort the variables in ascending order. Combine the vectors and sort the combined vector. Create output vectors for areas used in plots.

B: Create an output vector to store the instances of area inequality.

C: Uses the sorted ***X*** and ***Y*** variables as the LPM target under ***n*** = 1. Thus all observations are used in the

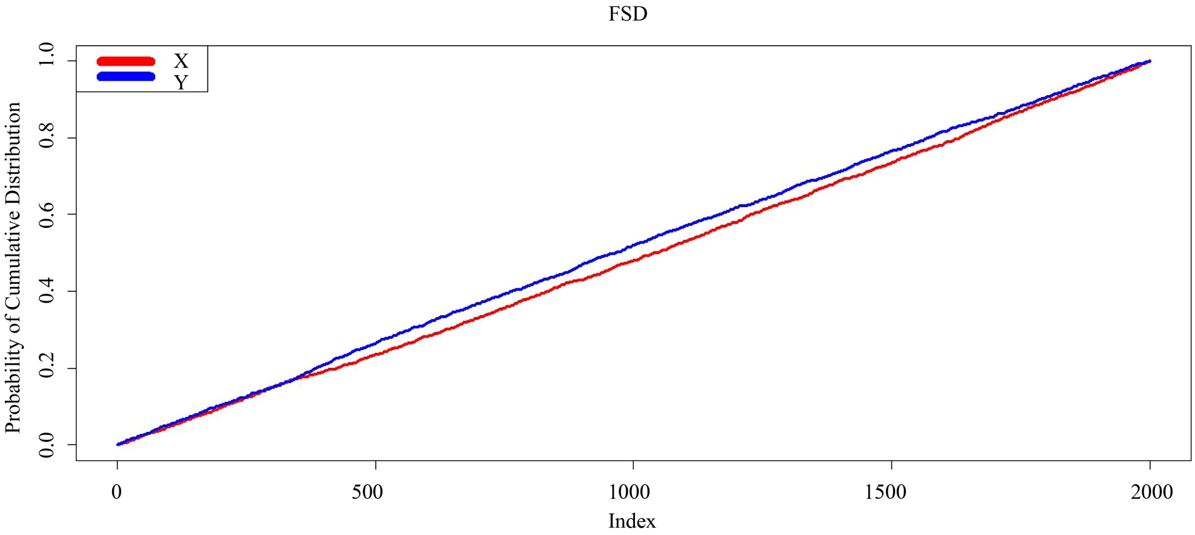


Figure A1. Plot of $F_X(h), F_Y(h)$ using all observations of *X* and *Y* as targets (*h*).

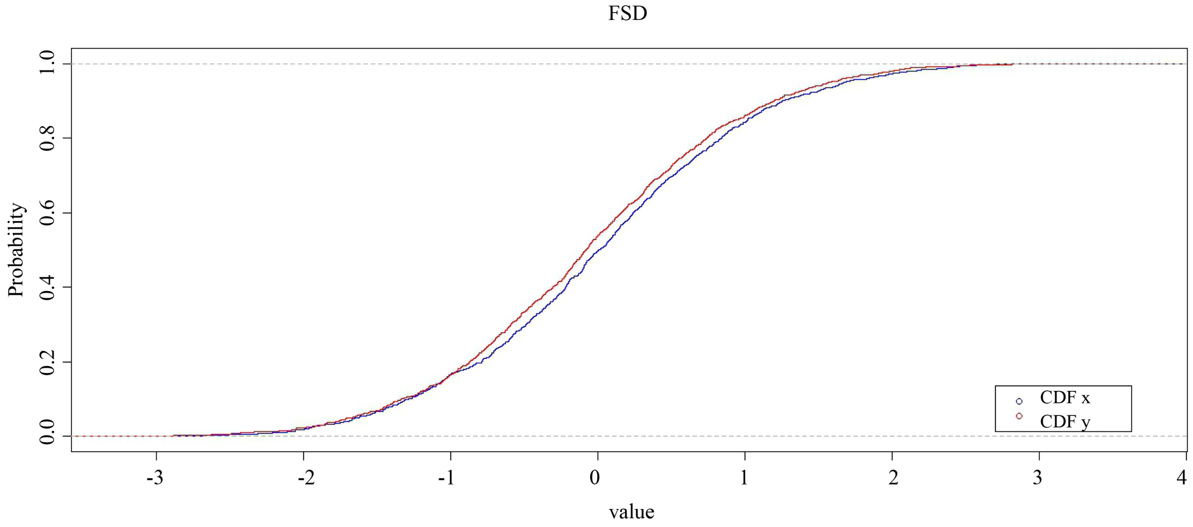


Figure A2. Typical CDF plot for variables *X* and *Y*.

stringent "for all *h*" qualification. If $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X) < \int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ for that observation target *h*, no instance is recorded into the output vector (output_x[i] < −0). Conversely, if $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X) > \int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ for that observation target *h*, the loop is stopped.

D: Uses the same logic as (C) above, only tests the inverse area relationships $\int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y) < \int_{-\infty}^{h}(h-X)\mathrm{d}F(X)$.

E: Plots the cumulative areas of each variable.

F: Reads the output vectors. If the output vector has 0 instances of $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X) > \int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$, then *X* SSD *Y*.

G: Test case.

Figure A3 shows the cumulative distribution areas for *X* and *Y*. The lower area and thus dominance is quite clear. Alternatively viewed as a histogram, **Figure A4** illustrates $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X) > \int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ for all positive values (a good thing), while simultaneously $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X) < \int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ for all negative values (also a good thing). This is evident when examining the skewness between both variables.

Skew(X)	Skew(Y)
0.06529391	0.04430945

Nawrocki [4] also finds excess positive skewness for the risk averse investor's portfolio compared to the maximum mean/variance portfolio. This in turn was expected, due to Hadar and Russell [3] original conclusions:

"For instance, we have indicated in the paper that FSD implies a certain relationship between the odd moments (and sometimes also between the even moments) of the prospects under consideration. Consequently, given that *P* is preferred to *P'* for all monotonic utility functions, we can immediately say that all the odd moments around zero of *P* are larger than the respective moments of *P'*."

Even when dominance interruptions occur in the histogram, the cumulative area to that point is not enough to negate the dominance of *X* over *Y*. Revisiting the LPMs from the FSD interval of question tells a different story when areas are compared:

The areas in **Table A1** never even really come close to intersecting like the CDFs did in **Table A2**. This result loosely supports the maximal hypothesis from Klecan *et al*. [11]—that the areas would have to be very volatile and crossing in order to negate SSD[4], and if that were true, it would of course reflect on the underlying CDFs

[4] Namely that this occurrence is less likely for larger differences in distribution CDFs.

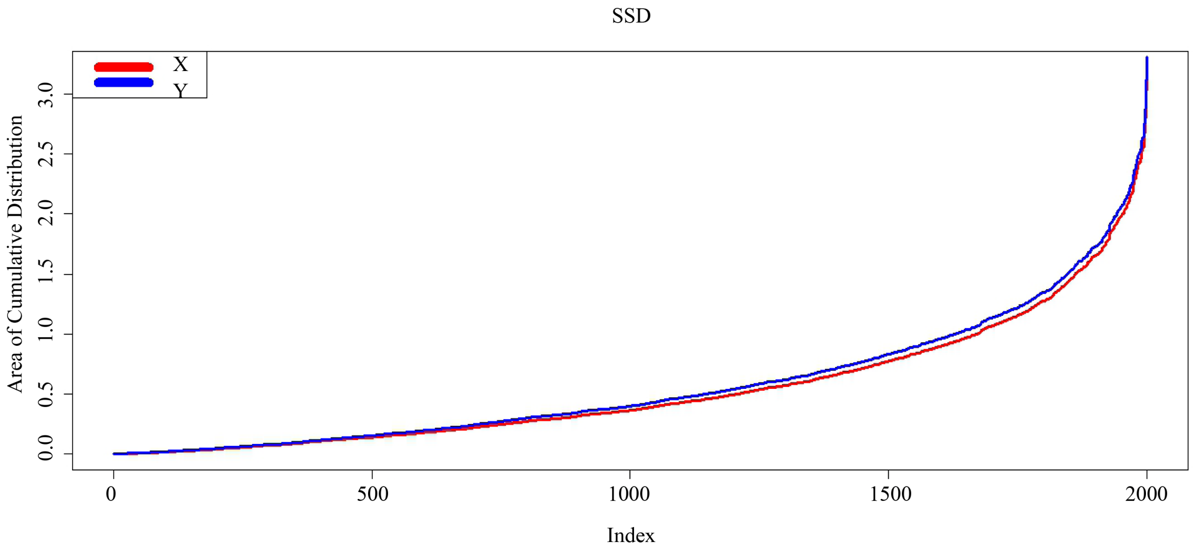


Figure A3. Plot of $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X),\int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ using all observations of *X* and *Y* as targets (*h*).

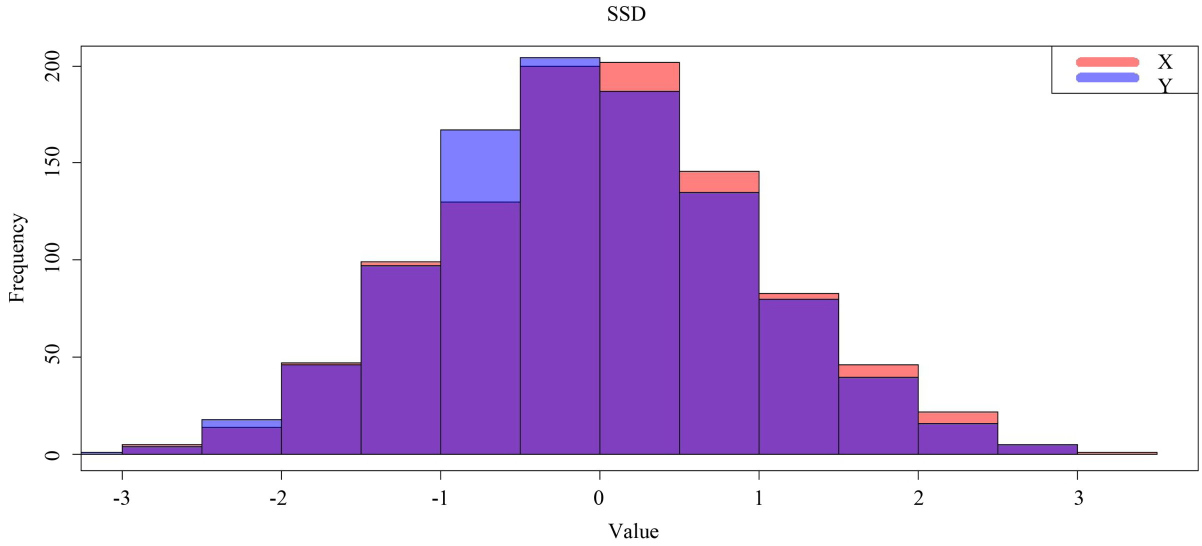


Figure A4. Histogram of $\int_{-\infty}^{h}(h-X)\mathrm{d}F(X),\int_{-\infty}^{h}(h-Y)\mathrm{d}F(Y)$ using all observations of *X* and *Y* as targets (*h*).

Table A1. Areas of variables *X*, *Y* over the interval [−1, −0.95].

Target	LPM (0,target,x)	LPM (0,target,y)	LPM (1,target,x)	LPM (1,target,y)
[−1.0]	0.165	0.166	0.0793658	0.0867652
[−0.99]	0.170	0.167	0.0810362	0.0884257
[−0.98]	0.170	0.169	0.0827362	0.0901005
[−0.97]	0.172	0.170	0.0844422	0.0917979
[−0.96]	0.173	0.174	0.0861641	0.0935241
[−0.95]	0.174	0.176	0.0878957	0.0952721

Table A2. CDF values for different targets for variables *X* and *Y*.

Target	LPM (0,target,x)	LPM (0,target,y)
[−1.0]	0.165	0.166
[−0.99]	0.17	0.167
[−0.98]	0.17	0.169
[−0.97]	0.172	0.17
[−0.96]	0.173	0.174
[−0.95]	0.174	0.176

used in FSD.

4) Module 4: Third Degree Stochastic Dominance (TSD)

Section A: Sort the variables in ascending order. Combine the vectors and sort the combined vector. Create output vectors for areas used in plots.

B: Create an output vector to store the instances of (area of the cumulative distribution)^2 inequality.

C: Uses the sorted *X* and *Y* variables as the LPM target under ***n*** = 2. Thus all observations are used in the stringent "for all *h*" qualification. If $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)<\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ for that observation target *h*, no instance is recorded into the output vector (output_x[i] < −0). Conversely, if $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)>\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ for that observation target *h*, the loop is stopped.

D: Uses the same logic as (C) above, only tests the inverse area relationships $\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)<\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)$.

E: Plots the cumulative areas squared for each variable.

F: Reads the output vectors. If the output vector has 0 instances of $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)>\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$, then *X* TSD *Y*.

G: Test case.

Figure A5 shows the TSD cumulative distribution areas for *X* and *Y*. The lower area and thus dominance is quite clear. Alternatively viewed as a histogram, **Figure A6** illustrates $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)>\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ for all positive values (a good thing), while simultaneously $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X)<\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ for all negative values (also a good thing).

A.2. Generalized Stochastic Dominance Efficient Sets

To extend the stochastic dominance tests and examine multiple portfolios, we use inspiration from Braid Theory. Braid Theory is an abstract geometric theory studying the everyday braid concept and we envision the CDFs as strings in these braids. Braids will nullify SD and avoid placing the CDFs in the "Dominated Set". **Figure A7** provides a visual representation of CDFs to braids, highlighting the crossing of CDFs the stochastic dominance routine is designed to decipher.

By testing for SD using the final ranks for that SD degree, we can derive the SD efficient sets. If a portfolio is dominated by a higher final ranked one, it is out of the efficient set. Example in Row 5 Column 4:

- The highest final ranked portfolio is the "Base". Test the "Base" portfolio against the next highest ranked, the "Challenger". 4 v 2. No SD exists. 2 joins the "Current Base" vector to run unidirectional[5] SD tests from.
- Test the new "Base" against the next "Challenger". 2 v 3.2 SD 3. 3 is placed in the "Dominated Set".
- Test the new "Base" against the next "Challenger". 2 is the last entry in the "Current Base". 2 v 1. No SD

[5] The earlier discussed tests were bi-directional, which involves more calculations between variables. Since the efficient sets are determined from unidirectional tests, a significant reduction in steps is realized. The attached code for determining SD efficient sets contains the unidirectional tests returning (1) for SD and (0) otherwise.

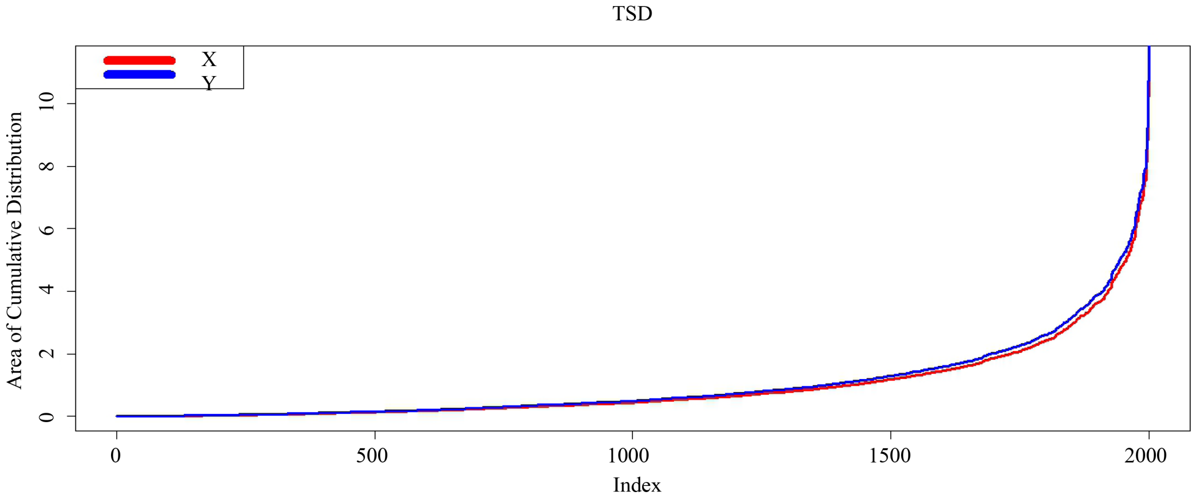


Figure A5. Plot of $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X),\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ using all observations of X and Y as targets (h). Note the effects of the higher loss aversion (n) on the distribution versus SSD cumulative plot in **Figure A3**.

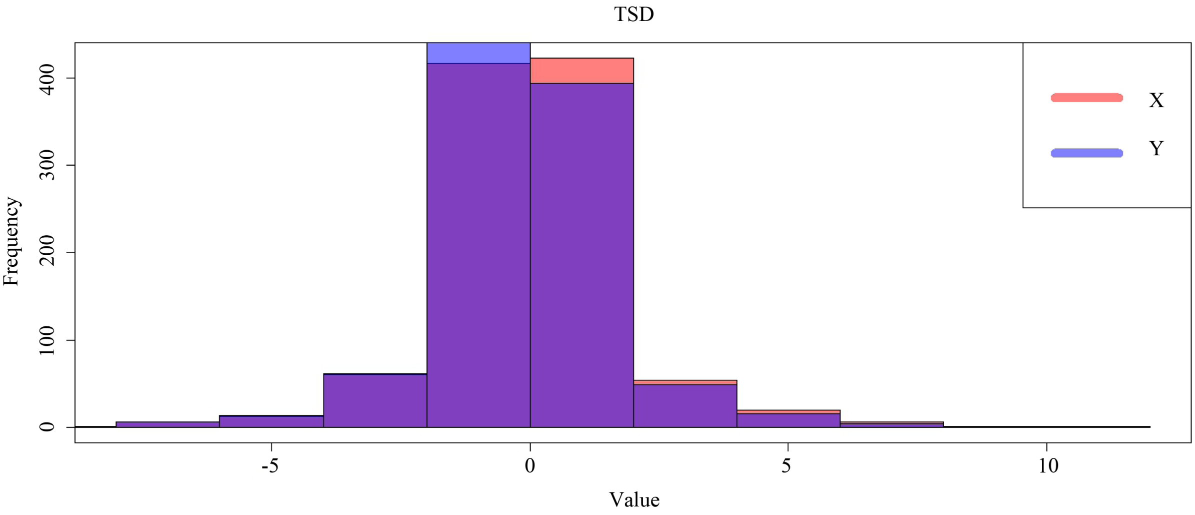


Figure A6. Plot of $\int_{-\infty}^{h}(h-X)^2\,\mathrm{d}F(X),\int_{-\infty}^{h}(h-Y)^2\,\mathrm{d}F(Y)$ using all observations of X and Y as targets (h). Note the effects of the higher loss aversion (n) on the distribution versus the SSD histogram in **Figure A4**.

exists. Test the remaining "Current Base" vector against 1. 4 v 1. No SD exists, 1 joins the "Current Base".

- There are no more "Challengers". Stop the procedure. The SD efficient set is the final ranked set less the "Dominated Set" {4,2,1}.
- #1 (the largest minimum value) can never be dominated, thus it is in every efficient set.

This procedure is different than that proposed by Porter, Wart and Ferguson [12] in several regards. One difference is that we do not check multiple SD degrees simultaneously[6]. Our second difference is we determine the final ranking by ordering the LPMs from the maximum of all observations. We substitute the degree 1 LPM final ranking for Porter *et al.*'s use of means and variances in ranking the securities. Finally, our third difference is the integration of the "Tricks" Porter *et al.* [12] identified to reduce the computational burden. We use the minimum value check as an "if" condition ("Trick 1"); and {break} commands at the end of each observation if a "Base" falls behind the "Challenger".

[6] A simple "for (i in 1:3)…" loop can be implemented to run all three SD tests with degree "i".

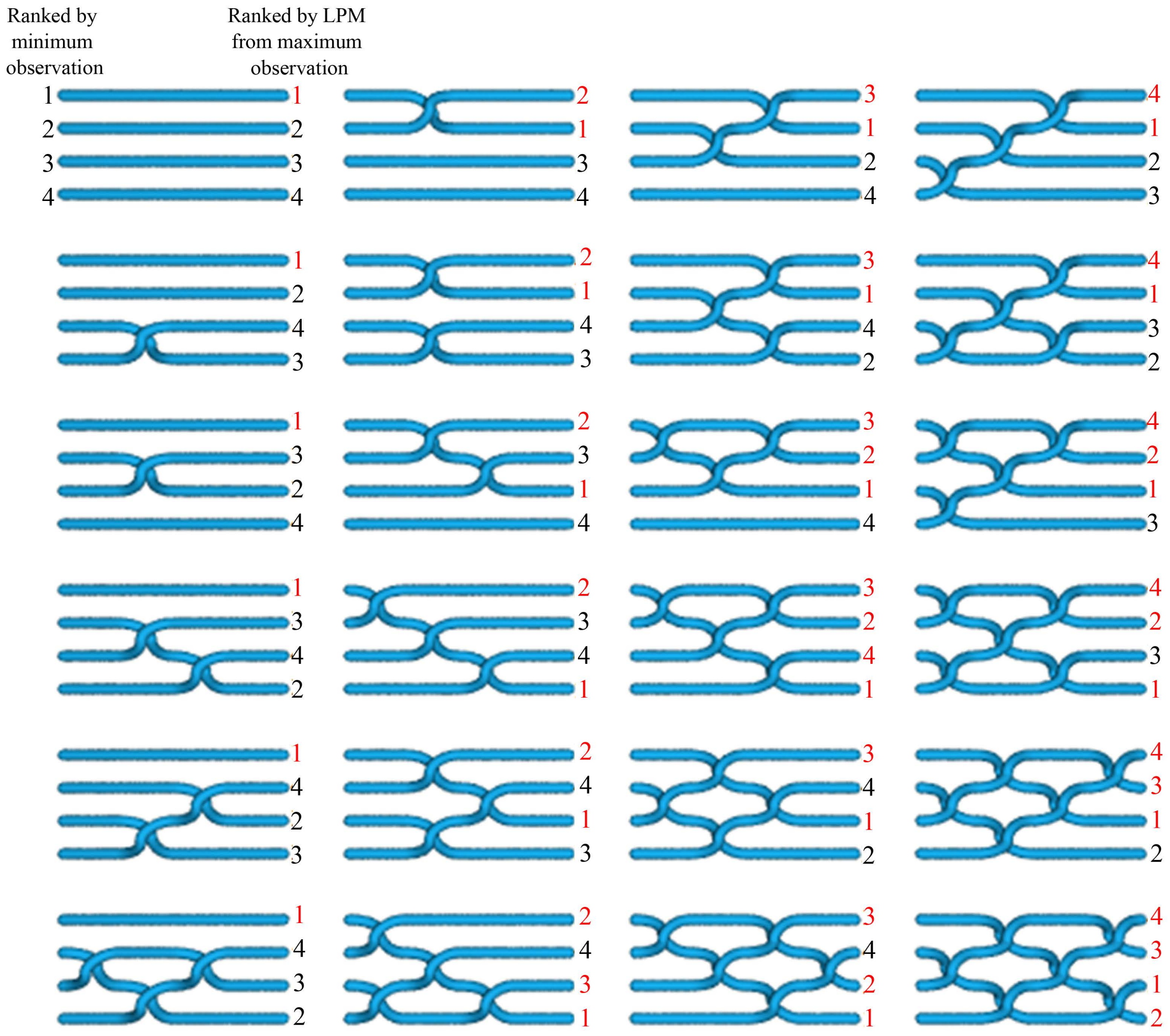


Figure A7. SD efficient sets (in red), versus ranked securities or portfolios in ascending order by their LPM from the maximum observation across all variables.

Appendix B. Security Listing (Return, StdDev and SemiDev Are in Monthly Percent)

```
Return   StdDevSemiDevSkewness Kurtosis    R/SV
    1    HGR  HANGER INC                   0.7690 11.2273  7.2207   0.1798   3.6307   0.0950
    2    ORCL ORACLE CORP                  1.0959  7.2366  4.4334   0.1860   3.6477   0.2285
    3    MSFT MICROSOFT CORP               0.9255  5.9767  3.7312  -0.0014   3.1440   0.2258
    4    ADM  ARCHER DANIELS MIDLAND C     1.0312  6.8616  4.2839   0.0109   2.6291   0.2214
    5    LH   LABORATORY CORP AMERICA      0.6116  4.5779  3.1067  -0.3210   2.6847   0.1702
    6    ITT  I T T CORP                   1.3730  7.3520  4.4869  -0.1042   2.1493   0.2875
    7    COP  CONOCOPHILLIPS               1.2691  6.1209  3.3869   0.3140   2.5133   0.3502
    8    AMGN AMGEN INC                    1.8507  5.3306  2.7830   0.1322   3.0371   0.6352
    9    CVX  CHEVRON CORP NEW             0.9117  5.4349  3.2878   0.1442   2.7255   0.2521
   10    AAPL APPLE INC                    2.2784  7.3861  3.9129   0.0715   2.6802   0.5611
   11    UTX  UNITED TECHNOLOGIES CORP     1.0408  5.1404  3.1077  -0.0384   2.7753   0.3082
   12    PG   PROCTER & GAMBLE CO          0.9458  3.6461  1.8420   0.5074   2.8102   0.4684
   13    CAT  CATERPILLAR INC              0.9894  8.5011  4.8733   0.6925   4.2032   0.1860
   14    PPG  PP G INDUSTRIES INC        2.5344  6.0189  2.8129     0.3712   3.5405   0.8715
   15    OGE  O G E ENERGY CORP            1.3395  5.2199  2.7282   0.6828   5.2428   0.4606
   16    DIS  DISNEY WALT CO               1.9123  5.7861  3.2662  -0.1700   2.9889   0.5601
   17    FITB FIFTH THIRD BANCORP          1.4059  7.4938  4.0196   0.8617   5.4833   0.3291
   18    PKI  PERKINELMER INC              1.3465  6.5214  3.8080   0.0830   3.2645   0.3318
```

19	SWKS	SKYWORKS SOLUTIONS INC	2.7716	10.7772	6.3028	-0.0077	3.3265	0.4266
20	TER	TERADYNE INC	1.0414	9.8089	5.5965	0.6466	3.2151	0.1713
21	MDT	MEDTRONIC INC	1.0267	5.3881	3.3171	-0.1938	3.1848	0.2845
22	NBL	NOBLE ENERGY INC	0.5592	8.2890	5.9252	-0.3302	4.2882	0.0804
23	VZ	VERIZON COMMUNICATIONS I	1.1050	4.8565	2.6757	0.2339	3.0469	0.3820
24	HCP	H C P INC	1.0377	5.4782	3.1289	0.2747	3.1027	0.3051
25	CGNX	COGNEX CORP	2.7093	9.1973	4.2666	1.0090	5.3626	0.6156
26	VTR	VENTAS INC	1.1874	5.1334	3.0539	-0.1536	2.9485	0.3616
27	AN	AUTONATION INC DEL	1.9332	8.1284	4.2809	0.5029	3.8272	0.4322
28	GILD	GILEAD SCIENCES INC	2.4832	7.9422	4.0822	0.1464	2.4284	0.5880
29	CAKE	CHEESECAKE FACTORY INC	1.4749	6.2218	3.4230	0.2825	3.0717	0.4066
30	SM	S M ENERGY CO	0.2130	11.3304	7.3964	0.3335	3.8515	0.0176
31	IT	GARTNER INC	2.6011	6.2311	2.8537	0.1652	2.3716	0.8824
32	NFX	NEWFIELD EXPLORATION CO	-0.9549	10.6741	8.0983	-0.1202	2.3751	-0.1282
33	HAIN	HAIN CELESTIAL GROUP INC	3.2600	6.9955	2.9708	0.3447	2.9107	1.0694
34	NPSP	N P S PHARMACEUTICALS IN	4.0002	16.1771	7.3916	0.7533	3.0338	0.5300
35	RWT	REDWOOD TRUST INC	1.0794	7.5331	4.8973	-0.2011	2.4926	0.2035
36	GWR	GENESEE& WYOMING INC	1.7033	7.3853	3.6917	0.7617	3.8703	0.4389
37	ETFC	E TRADE FINANCIAL CORP	0.5360	11.0846	7.5695	-0.1342	2.7340	0.0598
38	CBST	CUBIST PHARMACEUTICALS I	2.8203	9.4548	3.6849	1.4203	5.5048	0.7429
39	IM	INGRAM MICRO INC	0.7695	6.6373	4.2533	0.2126	3.6529	0.1614
40	NCR	N C R CORP NEW	1.6171	8.3190	4.9191	0.0094	2.7172	0.3119
41	DGX	QUEST DIAGNOSTICS INC	0.2891	5.4756	3.4928	0.3958	3.3968	0.0590
42	JLL	JONES LANG LASALLE INC	1.5605	9.1511	5.4426	0.2069	4.1737	0.2715
43	BBCN	BBCN BANCORP INC	0.4722	9.8331	5.9577	0.9461	5.9204	0.0653
44	WCN	WASTE CONNECTIONS INC	1.2216	4.1286	2.2822	-0.0569	2.0749	0.4990
45	AXL	AMERICAN AXLE & MFG HLGD	1.7409	11.1042	6.1676	0.5543	3.4137	0.2688
46	LPNT	LIFEPOINT HOSPITALS INC	1.3309	6.9184	3.9110	0.3368	3.6807	0.3191
47	UTHR	UNITED THERAPEUTICS CORP	1.5112	9.8108	5.6421	0.3078	3.6575	0.2531
48	MET	METLIFE INC	0.8996	8.4989	5.2792	0.2265	3.6371	0.1547
49	FLO	FLOWERS FOODS INC	1.2265	6.6653	3.6748	0.5856	4.1046	0.3112
50	SGEN	SEATTLE GENETICS INC	1.9374	12.6919	6.6612	0.7015	3.1473	0.2784
51	BTU	PEABODY ENERGY CORP	-2.7889	12.2612	9.9540	0.1402	2.8515	-0.2885
52	PRU	PRUDENTIAL FINANCIAL INC	1.2211	7.5936	4.8585	-0.3097	3.7000	0.2343
53	LO	LORILLARD INC	1.8594	6.3490	2.6717	1.3116	5.6931	0.6649
54	CNO	C N O FINANCIAL GROUP IN	2.1306	8.3733	4.5295	0.1901	2.6204	0.4521
55	AIZ	ASSURANT INC	1.5720	5.6719	3.3522	-0.3996	3.5490	0.4442
56	HOS	HORNBECK OFFSHORE SVCS I	0.1169	12.4657	7.9712	0.2774	4.1226	0.0043
57	GNW	GENWORTH FINANCIAL INC	-0.4808	13.0099	9.4124	-0.1679	2.7288	-0.0599
58	GOOG	GOOGLE INC	0.8974	7.5820	4.4331	0.4615	2.7851	0.1837
59	UA	UNDER ARMOUR INC	3.9052	9.3423	3.6693	0.6003	2.8163	1.0417
60	ROSE	ROSETTA RESOURCES INC	0.1890	13.8531	8.5618	0.7953	4.4816	0.0124
61	DK	DELEK U S HOLDINGS INC	2.5738	13.0881	6.9875	0.5362	3.2963	0.3565
62	WYN	WYNDHAM WORLDWIDE CORP	2.5986	7.3610	3.6669	0.4073	4.1462	0.6860
63	ROST	ROSS STORES INC	2.5966	5.9040	2.6335	0.1435	2.2341	0.9545
64	MASI	MASIMO CORP	0.0020	8.0120	5.1222	0.5320	2.9448	-0.0158
65	TDC	TERADATA CORP DE	0.5501	8.9215	5.7606	0.2042	2.6973	0.0811
66	V	VISA INC	1.9108	5.5480	3.3001	-0.6668	5.4948	0.5539
67	VSI	VITAMIN SHOPPE INC	1.3107	7.7425	4.8692	-0.1566	2.4296	0.2521

Appendix C

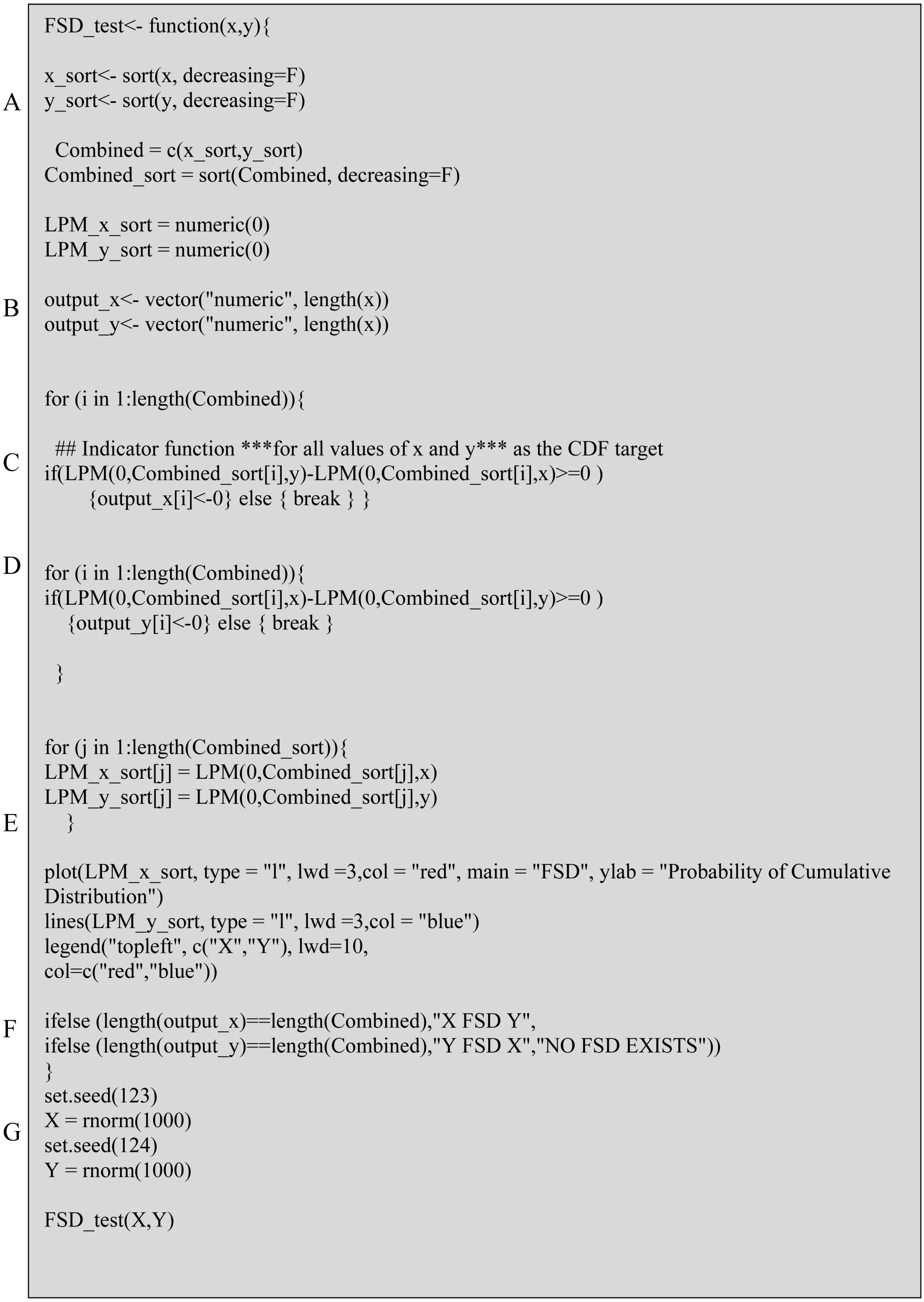
R CODE

This section of the paper will provide the R code for the stochastic dominance test using lower partial moments.

1) Module 1: Lower Partial Moment Function (LPM)

```
LPM<- function(degree,target,variable)
sum((target - (variable[variable < target]))^degree)/length(variable)
```

2) Module 2: First Degree Stochastic Dominance (FSD) [Bi-directional]



```
FSD_test<- function(x,y){

x_sort<- sort(x, decreasing=F)
y_sort<- sort(y, decreasing=F)

  Combined = c(x_sort,y_sort)
Combined_sort = sort(Combined, decreasing=F)

LPM_x_sort = numeric(0)
LPM_y_sort = numeric(0)

output_x<- vector("numeric", length(x))
output_y<- vector("numeric", length(x))


for (i in 1:length(Combined)){

  ## Indicator function ***for all values of x and y*** as the CDF target
if(LPM(0,Combined_sort[i],y)-LPM(0,Combined_sort[i],x)>=0 )
        {output_x[i]<-0} else { break } }


for (i in 1:length(Combined)){
if(LPM(0,Combined_sort[i],x)-LPM(0,Combined_sort[i],y)>=0 )
    {output_y[i]<-0} else { break }

  }


for (j in 1:length(Combined_sort)){
LPM_x_sort[j] = LPM(0,Combined_sort[j],x)
LPM_y_sort[j] = LPM(0,Combined_sort[j],y)
    }

plot(LPM_x_sort, type = "l", lwd =3,col = "red", main = "FSD", ylab = "Probability of Cumulative
Distribution")
lines(LPM_y_sort, type = "l", lwd =3,col = "blue")
legend("topleft", c("X","Y"), lwd=10,
col=c("red","blue"))

ifelse (length(output_x)==length(Combined),"X FSD Y",
ifelse (length(output_y)==length(Combined),"Y FSD X","NO FSD EXISTS"))
}
set.seed(123)
X = rnorm(1000)
set.seed(124)
Y = rnorm(1000)

FSD_test(X,Y)
```

3) Module 3: Second Degree Stochastic Dominance (SSD) [Bi-directional]

```
    SSD_test<- function(x,y){

    x_sort<- sort(x, decreasing=F)
A   y_sort<- sort(y, decreasing=F)

      Combined = c(x_sort,y_sort)
    Combined_sort = sort(Combined, decreasing=F)

    LPM_x_sort = numeric(0)
    LPM_y_sort = numeric(0)

B   output_x<- vector("numeric", length(x))
    output_y<- vector("numeric", length(x))

    for (i in 1:length(Combined)){

    if(LPM(1,Combined_sort[i],y)-LPM(1,Combined_sort[i],x)>=0 )
       {output_x[i]<-0} else { break }}
C
    for (i in 1:length(Combined)){

    if(LPM(1,Combined_sort[i],x)-LPM(1,Combined_sort[i],y)>=0 )
D      {output_y[i]<-0} else { break }

       }


    for (j in 1:length(Combined_sort)){
    LPM_x_sort[j] = LPM(1,Combined_sort[j],x)
    LPM_y_sort[j] = LPM(1,Combined_sort[j],y)
E      }

    plot(LPM_x_sort, type = "l", lwd =3,col = "red", main = "SSD", ylab = "Area of Cumulative
    Distribution")
    lines(LPM_y_sort, type = "l", lwd =3,col = "blue")
    legend("topleft", c("X","Y"), lwd=10,
    col=c("red","blue"))

F   ifelse (length(output_x)==length(Combined),"X SSD Y",
    ifelse (length(output_y)==length(Combined),"Y SSD X","NO SSD EXISTS"))
    }
    set.seed(123)
G   X = rnorm(1000)
    set.seed(124)
    Y = rnorm(1000)

    SSD_test(X,Y)
```

4) Module 4: Third Degree Stochastic Dominance (TSD) [Bi-directional]

```
    TSD_test<- function(x,y){

    x_sort<- sort(x, decreasing=F)
A   y_sort<- sort(y, decreasing=F)

      Combined = c(x_sort,y_sort)
    Combined_sort = sort(Combined, decreasing=F)

    LPM_x_sort = numeric(0)
    LPM_y_sort = numeric(0)

B   output_x<- vector("numeric", length(x))
    output_y<- vector("numeric", length(x))


    for (i in 1:length(Combined)){

    if(LPM(2,Combined_sort[i],y)-LPM(2,Combined_sort[i],x)>=0 )
C       {output_x[i]<-0} else { break }}

    for (i in 1:length(Combined)){

D   if(LPM(2,Combined_sort[i],x)-LPM(2,Combined_sort[i],y)>=0 )
        {output_y[i]<-0} else { break }

      }



    for (j in 1:length(Combined_sort)){
    LPM_x_sort[j] = LPM(2,Combined_sort[j],x)
    LPM_y_sort[j] = LPM(2,Combined_sort[j],y)
      }

E   plot(LPM_x_sort, type = "l", lwd =3,col = "red", main = "TSD", ylab = "Area of Cumulative
    Distribution")
    lines(LPM_y_sort, type = "l", lwd =3,col = "blue")
    legend("topleft", c("X","Y"), lwd=10,
    col=c("red","blue"))

F   ifelse (length(output_x)==length(Combined),"X TSD Y",
    ifelse (length(output_y)==length(Combined),"Y TSD X","NO TSD EXISTS"))
    }
G   set.seed(123)
    X = rnorm(1000)
    set.seed(124)
    Y = rnorm(1000)

    TSD_test(X,Y)
```

5) Module 5: Stochastic Dominance Efficient Set (SDES) & Uni-directional SD Tests

```
FSD <- function(x,y){

x_sort<- sort(x, decreasing=F)
y_sort<- sort(y, decreasing=F)

  Combined = c(x_sort,y_sort)
Combined_sort = sort(Combined, decreasing=F)

LPM_x_sort = numeric(0)
LPM_y_sort = numeric(0)

output_x<- vector("numeric", length(x))
output_y<- vector("numeric", length(x))

if (min(y)>=min(x)) {return(0)} else {

for (i in 1:length(Combined)){

    ## Indicator function ***for all values of x and y*** as the CDF target
if(LPM(0,Combined_sort[i],y)-LPM(0,Combined_sort[i],x)>=0 )
    {output_x[i]<-0} else { break } }


for (i in 1:length(Combined)){
if(LPM(0,Combined_sort[i],x)-LPM(0,Combined_sort[i],y)>=0 )
    {output_y[i]<-0} else { break }

  }

for (j in 1:length(Combined_sort)){
LPM_x_sort[j] = LPM(0,Combined_sort[j],x)
LPM_y_sort[j] = LPM(0,Combined_sort[j],y)
  }

ifelse (length(output_x)==length(Combined) & min(x)>=min(y),return(1),return(0))
}

}
```

```
SSD <- function(x,y){

x_sort<- sort(x, decreasing=F)
y_sort<- sort(y, decreasing=F)

  Combined = c(x_sort,y_sort)
Combined_sort = sort(Combined, decreasing=F)

LPM_x_sort = numeric(0)
LPM_y_sort = numeric(0)

output_x<- vector("numeric", length(x))
output_y<- vector("numeric", length(x))

if (min(y)>=min(x)) {return(0)} else {

for (i in 1:length(Combined)){

if(LPM(1,Combined_sort[i],y)-LPM(1,Combined_sort[i],x)>=0 )
    {output_x[i]<-0} else { break }}

for (i in 1:length(Combined)){

if(LPM(1,Combined_sort[i],x)-LPM(1,Combined_sort[i],y)>=0 )
    {output_y[i]<-0} else { break }

  }


for (j in 1:length(Combined_sort)){
LPM_x_sort[j] = LPM(1,Combined_sort[j],x)
LPM_y_sort[j] = LPM(1,Combined_sort[j],y)
  }

ifelse (length(output_x)==length(Combined),return(1),return(0))
}
}
```

```
TSD <- function(x,y){

x_sort<- sort(x, decreasing=F)
y_sort<- sort(y, decreasing=F)

  Combined = c(x_sort,y_sort)
Combined_sort = sort(Combined, decreasing=F)

LPM_x_sort = numeric(0)
LPM_y_sort = numeric(0)

output_x<- vector("numeric", length(x))
output_y<- vector("numeric", length(x))

if (min(y)>=min(x) | mean(y)>=mean(x)) {return(0)} else {

for (i in 1:length(Combined)){

if(LPM(2,Combined_sort[i],y)-LPM(2,Combined_sort[i],x)>=0 )
    {output_x[i]<-0} else { break }}

for (i in 1:length(Combined)){

if(LPM(2,Combined_sort[i],x)-LPM(2,Combined_sort[i],y)>=0 )
    {output_y[i]<-0} else { break }

  }


for (j in 1:length(Combined_sort)){
LPM_x_sort[j] = LPM(2,Combined_sort[j],x)
LPM_y_sort[j] = LPM(2,Combined_sort[j],y)
  }


ifelse (length(output_x)==length(Combined),return(1),return(0))
}
}
```

```
SD_Efficient_Set<- function(A,degree) {
 n <- ncol(A)
max_target<- max(A)
LPM_order<- numeric(0)
Dominated_set<- numeric(0)
current_base<- numeric(0)


for (i in 1:n){
LPM_order[i] <- LPM(max(0,1),max(A),A[,i])

   }

final_ranked<- A[,order(LPM_order)]

current_base<-1
if(degree==1){
for (i in 1:(n-1)) {

base<- final_ranked[,current_base[length(current_base)]]

challenger<- final_ranked[,i+1]

if (FSD(base,challenger)==1){ current_base[i]<- current_base[length(current_base)]}

if (FSD(base,challenger)==0){

for (j in current_base){
base<- final_ranked[,j]
if (FSD(base,challenger)==0){ next }
else
         {Dominated_set[i] <- i+1  }
       }

current_base[i]<- i+1}

else {Dominated_set[i]<- i+1 }
  }
}


if(degree==2){
for (i in 1:(n-1)) {

base<- final_ranked[,current_base[length(current_base)]]

challenger<- final_ranked[,i+1]

if (SSD(base,challenger)==1){ current_base[i]<- current_base[length(current_base)]}


if (SSD(base,challenger)==0){

for (j in current_base){
base<- final_ranked[,j]
if (SSD(base,challenger)==0){ next }
else
       {Dominated_set[i] <- i+1  }
     }

current_base[i]<- i+1}

else {Dominated_set[i]<- i+1 }
   }
 }
if(degree==3){
for (i in 1:(n-1)) {

base<- final_ranked[,current_base[length(current_base)]]

challenger<- final_ranked[,i+1]

if (TSD(base,challenger)==1){ current_base[i]<- current_base[length(current_base)]}


if (TSD(base,challenger)==0){

for (j in current_base){
base<- final_ranked[,j]
if (TSD(base,challenger)==0){ next }
else
        {Dominated_set[i] <- i+1  }
      }

current_base[i]<- i+1}

else {Dominated_set[i]<- i+1 }
   }
 }


if(length(Dominated_set)>0){
print(as.matrix(colnames(final_ranked[-na.omit(Dominated_set)])))}
else {print(as.matrix(colnames(final_ranked)))}

}
```