SoftwareX 33 (2026) 102461

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

RSD: An R package to calculate stochastic dominance

Shayan Tohidi, Sigurdur Olafsson*

Industrial Engineering Department, Iowa State University, Ames, IA, United States

ARTICLE INFO

Keywords:
R package
Stochastic dominance
Almost stochastic dominance
Stochastic order

ABSTRACT

Stochastic dominance is a classical method for comparing two random variables using their probability distribution functions. As for all stochastic orders, stochastic dominance does not always establish an order between the random variables, and almost stochastic dominance was developed to address such cases, thus extending the applicability of stochastic dominance to many real-world problems. We developed an R package that consists of a collection of methods for testing the first- and second-order (almost) stochastic dominance for discrete random variables. This article describes the package and illustrates these methods using both synthetic datasets covering a range of possible scenarios that can occur, and a practical example where the comparison of discrete random variables using stochastic dominance can be applied to aid decision-making.

Metadata

Table 1
Code metadata.

Nr.	Code metadata description	Metadata
C1	Current code version	Version: 0.2.0
C2	Permanent link to code/repository used for this code version	https://github.com/ShayanTohidi/RSD.git
C3	Permanent link to Reproducible Capsule	https://cran.r-project.org/web/packages/RSD/index.html
C4	Legal Code License	GPL-3.0
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	R
C7	Compilation requirements, operating environments & dependencies	dplyr, tidyr, ggplot2
C8	If available Link to developer documentation/manual	https://cran.r-project.org/web/packages/RSD/RSD.pdf
C9	Support email for questions	shayant@iastate.edu

1. Motivation and significance

1.1. Scientific background

Stochastic ordering compares random variables or probability distributions in terms of their likelihood of producing larger or smaller outcomes. This field has a long history and one of the earliest methods is the Mean-Variance (MV) method of Markowitz [1]. The MV method is still widely used in portfolio selection and uses variance as the risk index. In decision analysis, stochastic ordering is typically based on Stochastic Dominance (SD), which does not depend on the selection of a risk index. The SD theory was developed by Hadar and Russell [2], Hanoch and Levy [3], Rothschild and Stiglitz [4] and Whitmore [5], independently. However, for most real-world problems SD provides only partial ordering; that is, for many pairs of random variables it is not possible to say that one random variable dominates the other according to classical SD. Motivated by this, Leshno and Levy [6] developed Almost Stochastic Dominance (ASD) to obtain an order that is acceptable for most, but not necessarily all, decision-makers. In this paper, we refer to this as the ASD^{LL} rule. Later, other researchers proposed variants of ASD, including Tzeng et al. [7], who proposed an ASD variant that we will refer to as the ASD^{THS} rule. Both variants have been widely applied in fields such as economics and finance [8–11].

[*] Corresponding author.
Email addresses: shayant@iastate.edu (S. Tohidi), olafsson@iastate.edu (S. Olafsson).

https://doi.org/10.1016/j.softx.2025.102461
Received 15 August 2025; Received in revised form 9 November 2025; Accepted 24 November 2025
Available online 30 January 2026

Although SD has been used in various application areas for decades, there is still a shortage of general-purpose software that facilitates its use. Two recent software packages partially address this shortage. For users working in the R statistical programming environment there is a recent package available called *stodom* [12] that has two SD functions for testing first- and second-order stochastic dominance. A more comprehensive Python package called *psysdtest* is also available which includes further testing procedures for classical SD [13], but neither package includes tests for ASD, which we argue is often the most useful SD procedure in practice. There is therefore a significant need for a user-friendly software package to easily conduct SD tests, especially for ASD where no prior software packages exist.

The RSD package described here addresses the need for an easy-to-use R package for SD tests, including ASD tests. Before describing the details of this package, we provide a brief mathematical description of the SD rules implemented in the package. For this description we assume that we have two random variables with their cumulative distribution functions (CDFs) F and G, and a utility function u reflecting the risk tolerance of the decision maker. The classical SD rules are first- and second-order stochastic dominance, which can be mathematically stated as follows.

- **FSD:** For all rational decision makers with increasing utility functions ($u' \geq 0$), F dominates G by first-order stochastic dominance (FSD) if and only if

  $$F(x) \leq G(x), \forall x.$$

  The strict inequality must be true for at least one level of x. This also means at each level of x, the probability of achieving less than that level is smaller for F than that for G.
- **SSD:** For all rational and risk-averse decision makers with increasing concave utility functions ($u' \geq 0$ and $u'' \leq 0$), F dominates G by second-order stochastic dominance (SSD) if and only if

  $$\int_{-\infty}^{x} [F(t) - G(t)]\, dt \leq 0, \forall x.$$

  The strict inequality must be true for at least one level of x.

If these conditions are not met, these classical SD rules cannot establish an order for the random variable. In applications, this occurs frequently and, as previously stated, motivates the need for ASD rules. To mathematically state the ASD rules implemented here, we require some additional notation. Given $F^{(2)}$ and $G^{(2)}$ as the SSD values, let $||F - G||$ denote the amount of area between CDFs, and $||F^{(2)} - G^{(2)}||$ denote the amount of area between SSDs. Given a user-defined tolerance threshold $\epsilon \in (0, 0.5)$, the basic ASD and the two ASD variants considered here can be defined as follows:

- **AFSD:** F dominates G by AFSD if and only if

  $$\int_{S_1} [F(x) - G(x)]\, dx \leq \epsilon ||F - G||,$$

  where S_1 is a subset of support when $G(x) < F(x)$.
- **ASSD$^{\text{LL}}$:** F dominates G by ASSD$^{\text{LL}}$ if and only if

  $$\int_{S_2} [F(x) - G(x)]\, dx \leq \epsilon ||F - G||$$

  and $E_F(X) \geq E_G(X)$, where S_2 is a subset of S_1 when $G^{(2)}(x) < F^{(2)}(x)$.
- **ASSD$^{\text{THS}}$:** F dominates G by ASSD$^{\text{THS}}$ if and only if

  $$\int_{\hat{S}_2} \left[F(x)^{(2)} - G^{(2)}(x)\right] dx \leq \epsilon ||F^{(2)} - G^{(2)}||$$

  and $E_F(X) \geq E_G(X)$, where $\hat{S}_2$ is a subset of support when $G^{(2)}(x) < F^{(2)}(x)$.

The idea here is that most decision makers will agree with the comparison, that is, which distribution dominates the other, but there are some decision makers with valid utility functions who would prefer the alternative distribution. The ratio of all decision makers who prefer the alternative distribution is given by ϵ. The smaller the ϵ value, the smaller the ratio of eliminated utility functions and the larger the acceptance rate among all decision makers.

1.2. Package contributions and limitations

The contribution of the *RSD* package is to provide users with easy-to-use functions to calculate both SD and ASD tests for two discrete random variables given their distribution functions. This enables pairwise comparison of discrete random variables and establishes a stochastic order when possible. Thus, it is also the basis for ranking multiple random variables according to their stochastic dominance order. Specifically, the package has functions to perform comparisons using FSD, SSD, AFSD, and ASSD tests according to existing definitions for any two discrete random variables. This is the only package available that provides all of this functionality, although as noted above some SD rules are implemented in other packages. Specifically, the R package *stodom* [12] implements consistent tests for the first- and second-order SD rules. This package has a total of five functions, three of which facilitate the plotting and comparison of distribution functions, and two that implement first-order and second-order stochastic dominance tests, respectively. The *pysdtest* package in Python [13] similarly implements statistical tests for SD rules and also has capabilities for plotting and comparing distribution functions. The *pysdtest* package has an extensive variety of tests for first- and second-order SD, and both packages use approximate testing procedures to identify the dominant distribution. However, neither package implements ASD, which in practice may be more important than SD, since in many real-world problems the SD rules are often violated, and classical SD thus only provides partial order. To address this limitation, the *RSD* package provides users with an integrated way to conduct SD and ASD tests, along with useful visualization functions for both FSD and SSD.

The primary limitation of the package is that it assumes we are interested in discrete distributions. If users are interested in continuous distributions, they can use the existing packages mentioned above for approximate FSD and SSD tests. Another potential limitation is that the package is restricted to first- and second-order SD and ASD and does not calculate higher-order tests. While some users may be interested in such higher-order tests, the methods implemented here, namely FSD, SSD and the corresponding ASD methods appear to have received the most interest in the literature.

2. Software description

The *RSD* package can be used in the R environment to calculate first- and second-order stochastic and almost stochastic dominance results for discrete random variables. Such random variables are defined by their probability mass functions (PMFs) and the only input requirement is thus the PMFs of the two random variables to be ordered. Given this input, the FSD and SSD values are computed using exact methods, and the results are stored in an instance of a structured class within this library, which we refer to as an SD object. Given an SD object, there are functions to visualize these values to have a clearer picture of the problem. In addition, there are FSD, SSD, AFSD, and ASSD functions to compare those distributions to identify the stochastic order, if it exists. Currently, the two most frequently applied versions of ASSD rules are implemented in this package.

2.1. Software architecture

The *RSD* package is designed in a modular way that separates data input, dominance computation, and analytical functionality and visualization into distinct layers as illustrated in Fig. 1. The raw inputs are first entered by a user as the outcomes and their associated probabilities. These inputs are validated and processed to create an internal SD object,

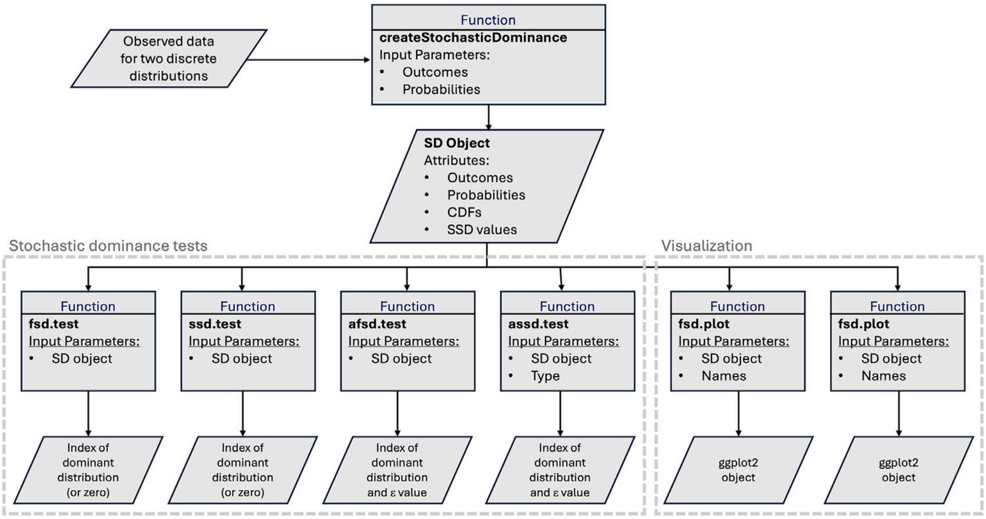

Fig. 1. Overview of the RSD package structure. The raw data is processed into a an internal SD object that becomes the input to all the stochastic dominance tests and visualization functions.

a structured data class that stores modified inputs as well as computed elements of FSD and SSD. The SD object serves as the core element of the package, and is the primary input to all the high-level analysis and visualization functions. These functions include FSD and SSD tests, their corresponding visualizations, and the extension of AFSD and ASSD rules. This design ensures clarity, modularity, and extensibility allowing users to seamlessly transition from raw inputs to advanced SD analysis in a unified computational framework.

2.2. *Software functionalities*

The RSD package is designed and implemented for SD and ASD calculations to compare two discrete distributions efficiently, and can be applied to large datasets to rank random values effectively. The package includes seven visible functions that implement the most important tasks required for comparing and ranking distributions according to stochastic dominance. These functions fall into three categories: (i) class constructor, (ii) stochastic dominance tests, and (iii) visualization (see Fig. 1).
Class constructor:

- `createStochasticDominance` is a class constructor function for creating an instance of the SD class, which is used by all the remaining functions. This function has four input parameters: two vectors for outcomes and two for probabilities. It combines the distributions and calculates the CDF and SSD values. All the information is encapsulated inside the object as attributes.

Tests:

- `fsd.test` compares two distributions based on the FSD rule. It has one input parameter, an SD object, and returns an integer (0, 1, or 2) indicating the results of the test. It returns 0 if there is no dominant distribution, and otherwise it returns the index of the dominant distribution.
- `ssd.test` works the same as `fsd.test`, but for SSD instead of FSD.
- `afsd.test` compares two distributions within the object, which is the input parameter, via the AFSD rule. The return values are the dominant index, epsilon, and other details about the calculations.
- `assd.test` compares two distributions within the object, which is the input parameter, via the ASSD rule. The type of the ASSD rule is determined via another input parameter. The return values are the dominant index, epsilon, and other details about the calculations.

Visualization:

- `fsd.plot` visualizes the FSD values of two distributions inside the object, which is the input parameter, as a step function of the CDFs.
- `ssd.plot` visualizes the SSD values of two distributions of the object, which is the input parameter, as a strictly increasing piecewise linear function.

3. Illustrative examples

In this section, examples are discussed for illustration and explanation of different scenarios that can occur when using the RSD package functions.

3.1. *Simple stochastic dominance examples*

We start with some simple examples that demonstrate when one discrete random variable dominates another according to FSD, SSD and ASSD. The first example illustrates a scenario where one distribution dominates another distribution by FSD.

```
outcome1 <- c(1,3,5,7)
outcome2 <- c(2,4,6,8)
pr <- rep(1/4,4)
# Create SD object
sd.obj <- createStochasticDominance(outcome1,outcome2,pr,pr)
# Compare distributions based on FSD rule
fsd.test(sd.obj) # 2
# Visualizee the FSD values
fsd.plot(sd.obj)
```

In this example, the second distribution dominates by FSD, and hence the `fsd.test` function returns 2. Fig. 2 presents the CDFs of this example, which is the output of the `fsd.plot` function.

The next example is designed to illustrate a case where stochastic dominance is achieved by SSD, but not by FSD.

```
outcome1 <- c(1,3,5,7,9)
outcome2 <- c(2,4,6,8)
pr1 <- rep(1/5,5)
pr2 <- rep(1/4,4)
# Create SD object
sd.obj <- createStochasticDominance(outcome1,outcome2,pr1,pr2)
# Compare distributions based on SSD rule
ssd.test(sd.obj) # 2
# Visualize the SSD values
ssd.plot(sd.obj)
```

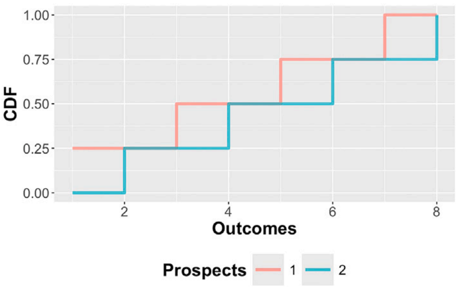

Fig. 2. Plot of the CDFs when there is a dominance by FSD.

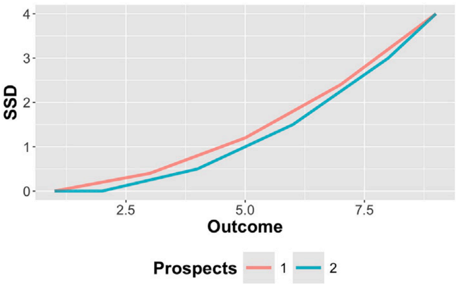

Fig. 3. Plot of the SSD values when there is a dominance by SSD.

In this example, the second distribution dominates by SSD (and hence, the `ssd.test` function returns 2), while the FSD does not determine an order (and hence, the `fsd.test` function returns 0).

An alternative when there is no FSD dominance is to check the AFSD rule, which means not all but most decision-makers agree with the result.

```
# Compare distributions based on AFSD rule
afsd.test(sd.obj) # 0
```

The `afsd.test` function returns 0, which means that neither distribution dominates the other according to the AFSD rule. The intuitive reason is that the violated areas of both distributions are exactly equal.

The ASSD rule can be checked when the SSD fails. There are various versions of ASSD rules that may yield similar conclusions.

```
# Compare distributions based on ASSD rule (LL version)
assd.test(sd.obj, type = 'll')
# Compare distributions based on ASSD rule (THS version)
assd.test(sd.obj, type = 'ths')
```

The index of the dominated distribution and additional details are included in the output of this `assd.test` function.

3.2. Application example

We now discuss a simple but practical example that motivates the need to compare two discrete distributions and how the *RSD* package can be used to support decision-making. In plant breeding there is a need to compare the yield of crop cultivars and select the ones that have better yields. However, the major source of variability of the yield of plant cultivars is the environment–whether the crop is planted in a favorable location (soil and climate) and whether the weather for the year is favorable or not. The yield may thus be considered a discrete random variable defined on the environments [14]. A stochastic order for these random variables allows the decision maker to conclude if one cultivar is truly better than another.

To demonstrate the package, we compare two rapeseed cultivars from a previously reported experiment by Shafii and Price [15]. The largest complete subset of this data includes 6 cultivars planted in 27 environments. We assume that *df* is a data frame with three variables: *env* describing the year and location, *gen* describing the cultivar, and *yield* with values shown in Table 2. We would like to compare two cultivars, Bienvenu and Dwarf, which are the first and fourth in the list of cultivars. Bienvenu has the higher mean yield, but that does not mean it is preferred across the environments, that is, it does not automatically imply that it has stochastic dominance over Dwarf when yield is viewed as a discrete random variable on the observed environments. We start by creating an SD object for these two cultivars.

Table 2
Data for the plant breeding example. The values of the *env* variable are shown as rows, values of the *gen* variable are shown as columns, and values of the *yield* variable are shown as the values.

env	gen					
	Bienvenu	Bridger	Cascade	Dwarf	Glacier	Jet
87_GGA	1442	1363	1505	1295	1681	1091
87_ID	1242	947	773	932	1111	1064
87_MT	2616	2828	2916	3452	3307	3660
87_NC	1001	1064	745	1014	1229	1674
87_OR	4556	2530	3336	3932	4185	3220
87_SC	2500	2705	2119	1894	2717	2833
87_TGA	1258	1868	1708	873	1453	954
87_TX	838	1069	735	988	952	1408
87_WA	4375	4604	4464	3974	4740	4344
88_ID	7638	5674	5987	6095	6230	5859
88_KS	1460	1373	1172	1489	1549	1363
88_MS	2737	2759	1879	2247	3142	2147
88_OR	3753	2671	2349	3328	2831	3251
88_SC	1216	1294	1062	1121	1674	732
88_TGA	975	1178	1291	479	1340	594
88_TX	1073	706	929	391	562	705
88_VA	2774	3351	3127	3116	3023	3132
88_WA	5727	3152	4208	4089	3895	3198
89_GGA	1845	1296	1731	1499	2099	1650
89_ID	5347	4154	5484	5952	5556	5863
89_NC	1377	1613	1097	1346	1393	696
89_NY	2861	3057	2514	3573	3229	3164
89_SC	1815	2847	2765	863	1685	901
89_TGA	447	1763	1478		534	317
89_TN	2892	2413	1934	2615	2192	2780
89_VA	1683	1653	1464	1593	1912	1806
89_WA	1726	4048	3936	3115	2787	3121

```
pr <- rep(1/27, 27)
outcome1 <- df$yield[df$gen == cultivars[1]]
outcome2 <- df$yield[df$gen == cultivars[4]]
pair <- createStochasticDominance(outcome1, outcome2, pr, pr)
```

First check if one cultivar has FSD over the other.

```
fsd.test(pair)
```

The test returns 0, which implies no FSD. We then move on to check for SSD.

```
ssd.test(pair)
```

The function returns 1, which means that Bienvenu has SSD over Dwarf, and this implies that all risk-averse decision makers would prefer Bienvenu for the set of observed environments. Both the lack of FSD and the existence of SSD are demonstrated in Fig. 4. The graphs in this figure are generated using the package visualization functions as follows:

```
fsd.plot(pair,names=c("Bienvenu","Dwarf"))+
  labs(
    x = "Yield",
    y = "CDF",
    color = "Cultivar")

ssd.plot(pair,names=c("Bienvenu","Dwarf"))+
  labs(
    x = "Yield",
    y = "Integral of CDF",
    color = "Cultivar")
```

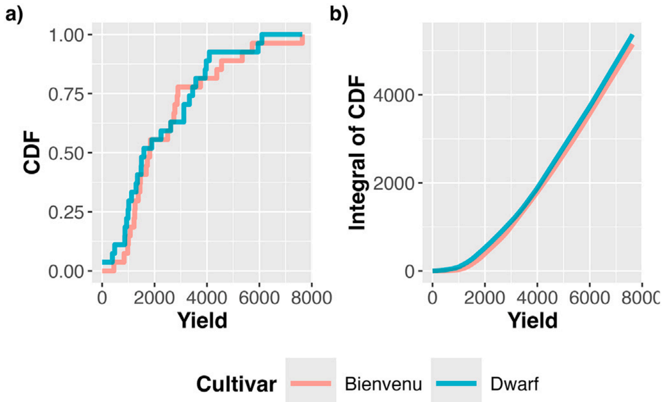

Fig. 4. Comparison of Bienvenu versus Dwarf rapeseed cultivars. The plots illustrate that (a) there is no FSD, but (b) there is a SSD of Bienvenu over Dwarf.

This demonstrates how stochastic dominance can be used to establish that even though the Bienvenu cultivar is not better than the Dwarf cultivar in all environments (e.g., Dwarf has a much higher yield in 87MT as shown the third row of Table 2), all risk-averse decision makers will prefer the Bienvenu cultivar to the Dwarf. And the new package makes all the relevant tests and plots straightforward for the decision maker to produce.

4. Impact

The *RSD* package will be useful for researchers and practitioners interested in establishing stochastic order between two or more discrete random variables using first- and second-order SD and ASD rules. It is the first available software package to include an implementation of ASD rules. Moreover, the SD and ASD rules are implemented using the exact methods rather than approximations, which makes the package reliable without sacrificing efficiency. As the calculations are fast, any of the functions of the *RSD* package can be used to rank probability distributions and solve large-scale real-world problems.

The user-friendly and generic implementation for this package can help researchers and even the decision makers themselves to solve real-world problems and have a better picture of the rankings by addressing the variability or risk. Moreover, it uses an open-source environments for developing, publishing, and accessing, which allow for further expansion and development by the community to include even more ranking algorithms based on SD and ASD. Thus, it makes comparing and ranking risky prospects efficient and reliable, so beyond academic usage, practitioners can also apply it to large datasets. In addition, since it is open-source, it can evolve to implement new approaches and meet versatile applications.

5. Conclusions

Comparing probability distributions is a classical problem. Stochastic dominance is perhaps the best known method for this problem and has been applied in various fields. SD is an exact method to compare distributions using probabilities to address outcomes and risks. However, like other methods for stochastic ordering, it only provides partial order as the rules can be violated easily by extreme utilities. This motivates almost stochastic dominance, which addresses the problem by eliminating the extreme utility functions. The ASD rules are not accepted by all, but most decision-makers. Although ASD rules may also order distributions partially, they rank more distributions than SD rules.

The *RSD* package addresses two shortcomings of previously available software for SD testing. First, there is no prior package that calculates SD rules for discrete distributions using exact methods (without any approximation). Second, and more importantly, there is no prior publicly available package that implements ASD. The SD and ASD methods play major roles in decision-making under uncertainty, because they are efficient, reliable and also use all the information of the distributions. The *RSD* package is developed in the R environment and implements SD and ASD algorithms using an exact approach for comparing discrete distributions. Thus, the results are valid and it runs fast because of using vectorized operations as much as possible. It can thus be used for ranking distributions in large datasets. In addition, it is an open-source framework, which allows others to expand and improve its implementation and application.

CRediT authorship contribution statement

Shayan Tohidi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis. **Sigurdur Olafsson:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Markowitz H. Portfolio selection. J Finance 1952;7(1):77–91. http://www.jstor.org/stable/2975974.

[2] Hadar J, Russell WR. Rules for ordering uncertain prospects. Am Econ Rev 1969;59(1):25–34. http://www.jstor.org/stable/1811090.

[3] Hanoch G, Levy H. The efficiency analysis of choices involving risk. Rev Econ Stud 1969;36(3):335–46. http://www.jstor.org/stable/2296431.

[4] Rothschild M, Stiglitz JE. Increasing risk: I. A definition. J Econ Theory 1970;2(3):225–43. https://doi.org/10.1016/0022-0531(70)90038-4. https://www.sciencedirect.com/science/article/pii/0022053170900384.

[5] Whitmore GA. Third-degree stochastic dominance. Am Econ Rev 1970;60(3):457–9. http://www.jstor.org/stable/1817999.

[6] Leshno M, Levy H. Preferred by "all" and preferred by "most" decision makers: almost stochastic dominance. Manag Sci 2002;48(8):1074–85. http://www.jstor.org/stable/822676.

[7] Tzeng LY, Huang RJ, Shih P-T. Revisiting almost second-degree stochastic dominance. Manage Sci 2013;59(5):1250–4. arXiv:https://doi.org/10.1287/mnsc.1120.1616

[8] Levy H. Stochastic dominance: investment decision making under uncertainty. Springer; 2015.

[9] Levy M. Almost stochastic dominance and stocks for the long run. Eur J Oper Res 2009;194(1):250–7. https://doi.org/10.1016/j.ejor.2007.12.017. https://www.sciencedirect.com/science/article/pii/S0377221707012118.

[10] Bali TG, Demirtas KO, Levy H, Wolf A. Bonds versus stocks: investors' age and risk taking. J Monet Econ 2009;56(6):817–30. https://doi.org/10.1016/j.jmoneco.2009.06.015. https://www.sciencedirect.com/science/article/pii/S0304393209001007.

[11] Levy H, Leshno M, Leibovitch B. Economically relevant preferences for all observed epsilon. Ann Oper Res 2010;176(1):153–78. https://doi.org/10.1007/s10479-008-0470-7

[12] Schaub S, Benni NE. How do price (risk) changes influence farmers' preferences to reduce fertilizer application? Agric Econ 2024;55(2):365–83. https://doi.org/10.1111/agec.12824. https://onlinelibrary.wiley.com/doi/pdf/10.1111/agec.12824.

[13] Lee K, Whang Y-J. Pysdtest: a python/stata package for stochastic dominance tests; 2024. arXiv:2307.10694. https://arxiv.org/abs/2307.10694.

[14] Tohidi S, Olafsson S. Probabilistic ranking of plant cultivars: stability explains differences from mean rank. Front Plant Sci 2025;16. https://doi.org/10.3389/fpls.2025.1553079

[15] Shafii B, Price WJ. Analysis of genotype-by-environment interaction using the additive main effects and multiplicative interaction model and stability estimates. J Agric Biol Environ Stat 1998:335–45. https://doi.org/10.2307/1400587