1. Way-of-work
   (1) Have fully understanding of the paper's section.
   (2) Find if there any similar implementation on Github.
   (3) Implement unit function needed for implementation via brute force algorithm, and verify some test cases' answers with hand-calculated answers.
   (4) Implement unit function needed for implementation via more efficient algorithms, and verify test cases' answers with brute force algorithm answers.
   (5) Integrate functions into a class and simplify code as much as possible.
   (6) Add built-in exceptions for the class.
   (7) Write Jupyter Notebook for demo.
2. Design choices
   (1) Although the paper chose to ignore transaction costs and compounding effects, I decided to make them optional parameters of the functions.
   (2) I chose to generate returns series based on date intervals rather than number of instances, because I think it will be more efficient to compare with real world data.
   (3) I added a function which can let the user to dd custom stop-loss policies, in order to expand flexibility of the class.
3. Learnings.
   (1) Analytical framework of stop-loss policies.
   (2) Linking usage of Bokeh
4. UML Class Diagram