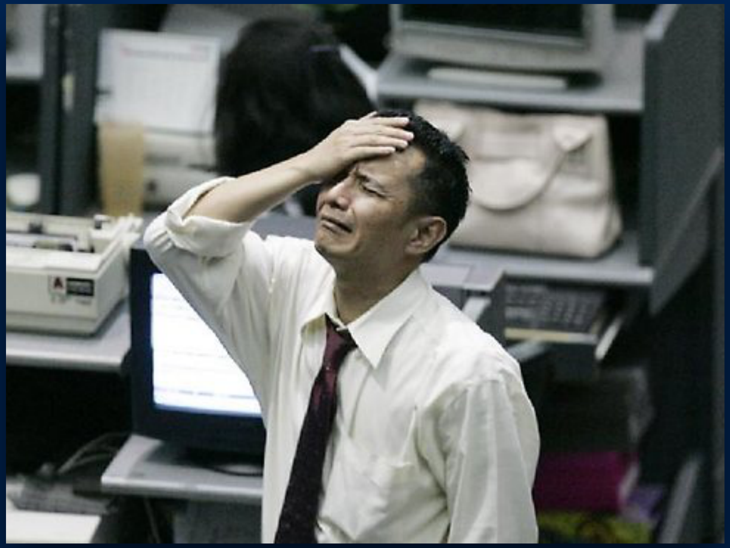
BRAVE model risk assessment for large portfolios

Kristof Verbeken

June 2015

How much value can a portfolio lose?

At a given confidence level (e.g. 95%)
For a given period (e.g. one day)

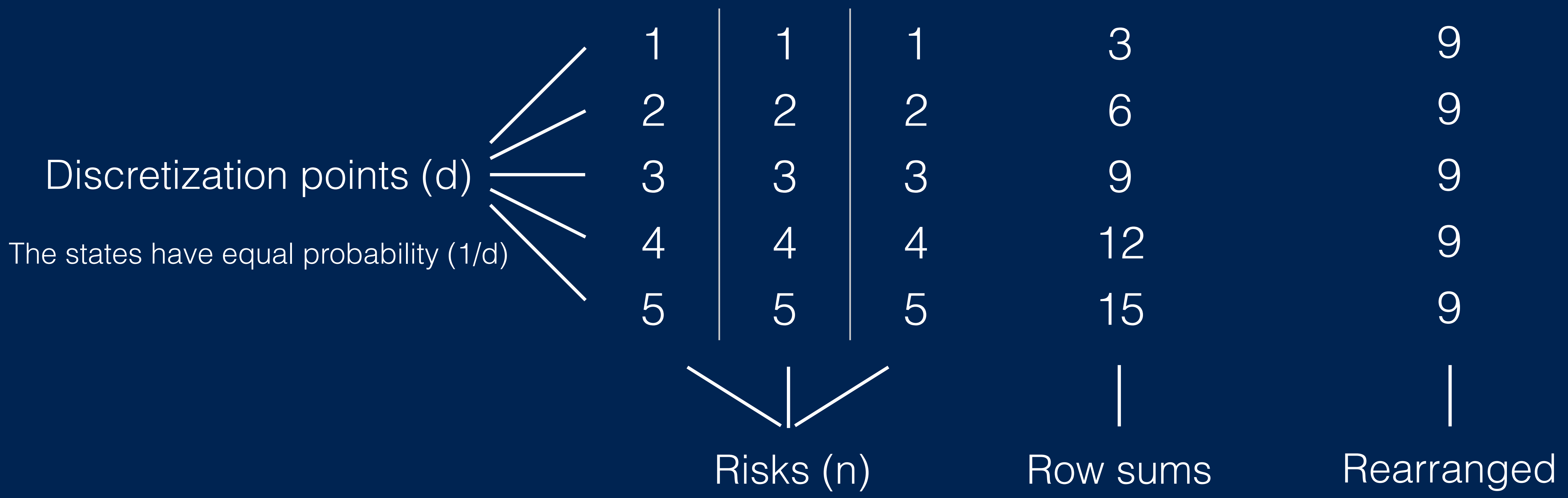


Contents

- Portfolio Value-at-Risk (VaR)
- Rearrangement Algorithms
- The blockra package
- Conclusion

Portfolio Value-at-Risk (VaR)

Notation



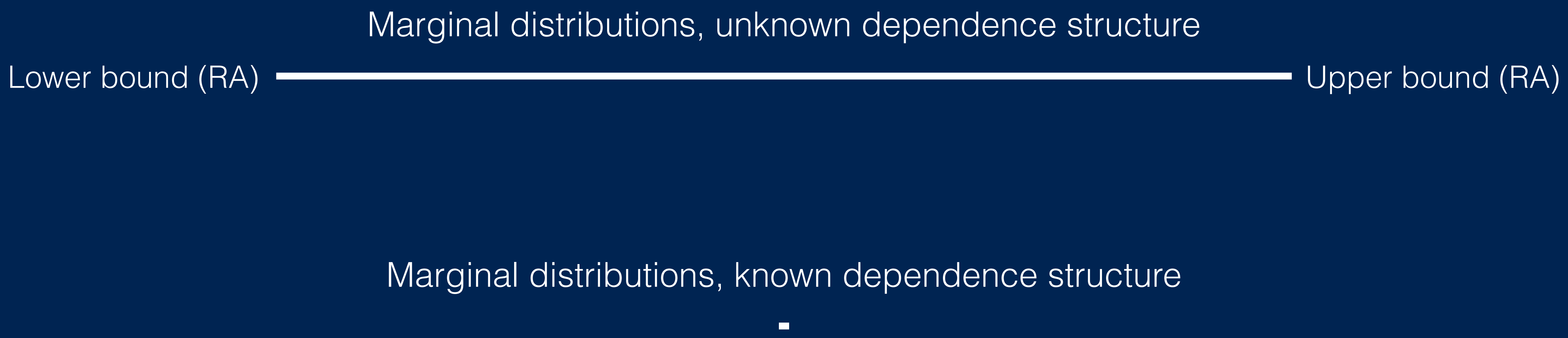
Discretization points (d)

The states have equal probability (1/d)

Risks (n)			Row sums	Rearranged
1	1	1	3	9
2	2	2	6	9
3	3	3	9	9
4	4	4	12	9
5	5	5	15	9

Portfolio Value-at-Risk (VaR)

Model risk



Marginal distributions, unknown dependence structure

Lower bound (RA) Upper bound (RA)

Marginal distributions, known dependence structure

Rearrangement Algorithms

Historic overview

1982 Two risks - Rüschendorf

2013 Rearrangement Algorithm (RA) - Embrechts, Puccetti and Rüschendorf

2014 Block Rearrangement Algorithm (BRA) - Bernard, Rüschendorf and Vanduffel

Rearrangement Algorithms

Overview

- Rearrangement Algorithm (RA)
    - ❖ Rearrange one column per iteration
- Block Rearrangement Algorithm (BRA)
    - ❖ Rearrange a random block per iteration
- Block Rearrangement Algorithm Variance Equalizer (BRAVE)
    - ❖ Rearrange the *best* block per iteration based on covariances

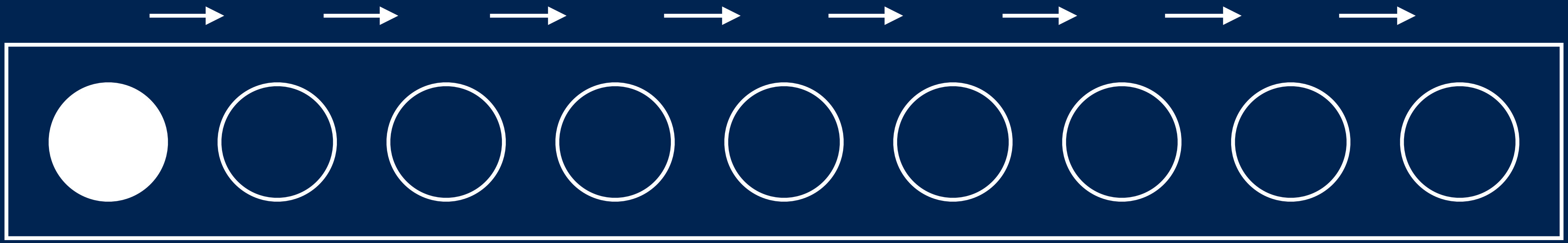
Rearrangement Algorithms

Algorithm structure

1. Partition
   - Split the matrix in two blocks
2. Rearrange
   - Rearrange blocks so that their row sums are inversely ordered
3. Repeat
   - Verify if convergence criterium is satisfied

Rearrangement Algorithms

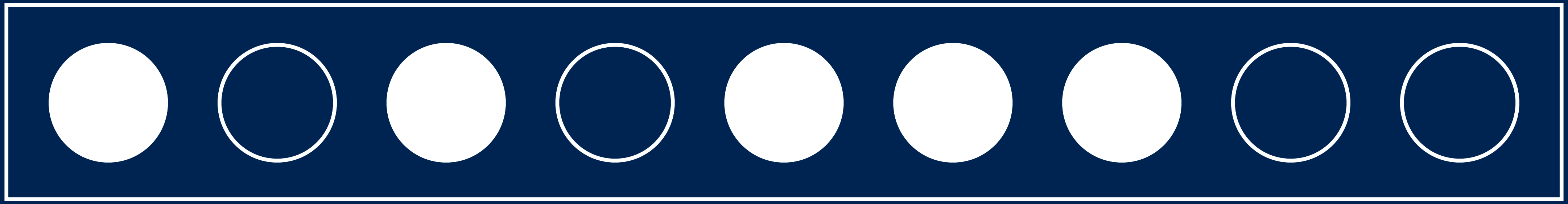
Partition - RA



- Select one column per iteration
  - Start with the first column
  - End with the last column

Rearrangement Algorithms

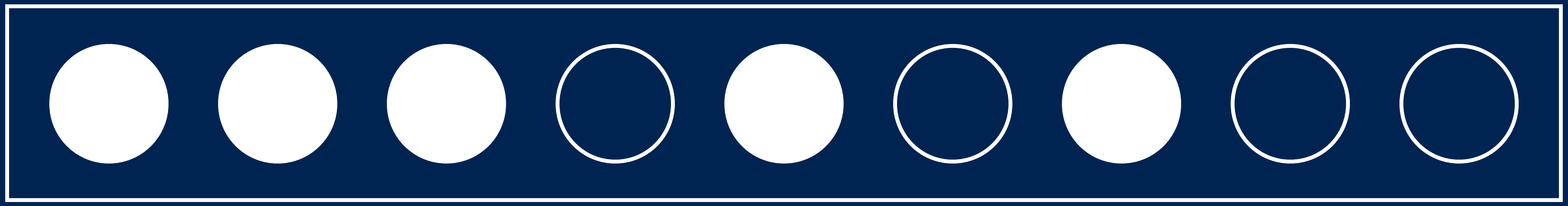
Partition - BRA



- Select two random subsets of the matrix
  - These subsets are referred to as “blocks”

Rearrangement Algorithms

Partition - BRAVE



Intelligently choose partitions to minimize rearrangements

1. Calculate covariances
    - Between each column and the total row sums
2. Divide the covariances in two blocks
    - Blocks have a cumulative covariance as equal as possible
    - Use the greedy partition problem

Rearrangement Algorithms

Rearrange

1. Calculate row sums per block
2. Sort blocks in opposite order
   - Based on row sums
   - Only rearrange the smallest partition

The blockra package

Introduction

- *blockra* package on R-forge
  - *http://r-forge.r-project.org/projects/blockra*
- Three main functions
  - *ra*: Rearrangement Algorithm
  - *bra*: Block Rearrangement Algorithm
  - *brave*: Block Rearrangement Algorithm Variance Equalizer

The blockra package

Function parameters

- Each rearrangement function takes the same parameters
  - X: the portfolio as a numeric matrix, with n risks and d discretizations
  - Epsilon: allowed error margin
  - Shuffle: shuffle matrix first if input matrix has specific order
  - Fix first: fix the order of the first column
  - Objective function: function to minimize (default is variance)

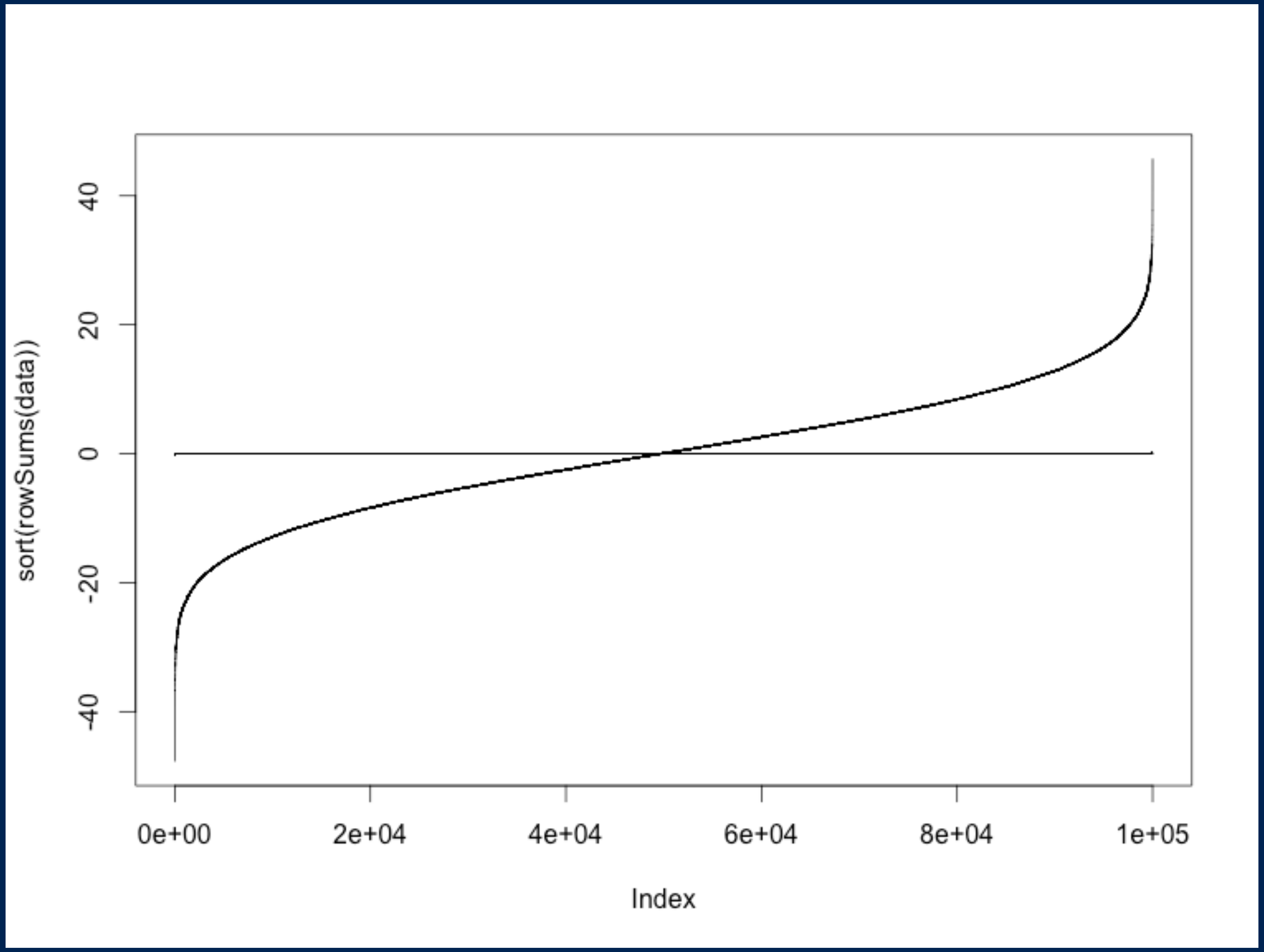
The blockra package

An example

```
> data <- rnorm(100000 * 100)
> dim(data) <- c(100000, 100)
>
> brave.1 <- brave(data, epsilon = 1)
> # That took only 1.28s
> brave.2 <- brave(data, epsilon = 0.001)
> # That took 1.7s
>
> var(rowSums(data))
[1] 100.5164
> var(rowSums(brave.1))
[1] 0.001234526
> var(rowSums(brave.1))
[1] 8.22296e-06
```

The blockra package

An example - continued



```
> plot(sort(rowSums(data)), type = “l”)
> lines(sort(rowSums(brave.1)))
```

Conclusion

Why R?

- Focus on algorithm, not implementation
  - Java version has >10x lines of code
  - R allows rapid prototyping
- Plotting is simple
- Open source
- Popularity in finance

Thank you!