

O Metodă Monte Carlo Generală pentru Analiza Mărimii Eșantionului

În Contextul Analizei de Rețea



Mihai A. Constantin
Noémi N. K. Schuurman
Jeroen K. Vermunt

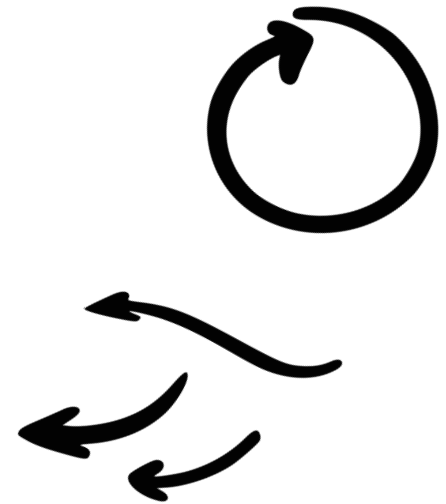
mihai@mihaiconstantin.com



Doctorand la . Lucrez la instrumente de **software statistic**. Pasionat de **R** și design-uri de **simulare**.

Overview

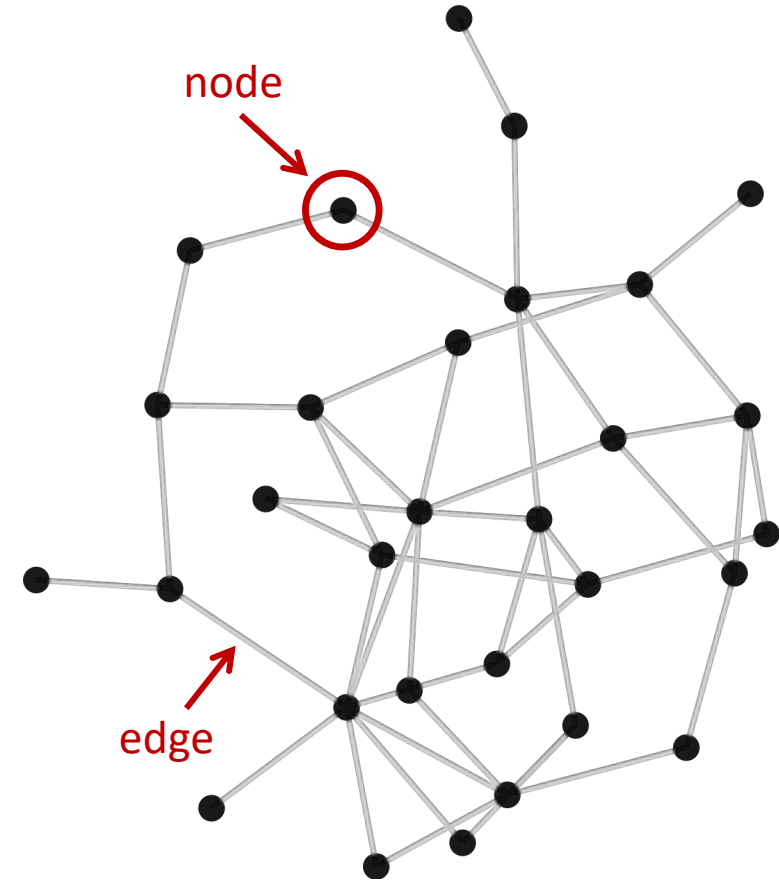
1. **Contextul** Problemei
2. **Input** pentru Metodă
3. **Pașii** Metodei
4. Câteva **Rezultate**
5. **Direcții** Viitoare



Contextul Problemei



Modele de rețea au crescut în popularitate în ultimii ani.



În ciuda popularității lor, **întrebarea** practică referitoare la...

Ce **dimensiune a eșantionului** este necesară pentru
a estima în mod fiabil **structura unei rețele**?

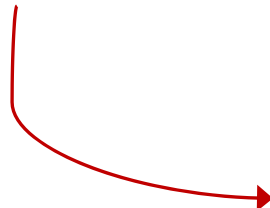
...rămâne în mare parte fără **răspuns**.

Ce se întâmplă dacă aveți nevoie de o recomandare în acest moment?

- utilizați **recomandări** din studii de simulare la scară largă
- realizați un **studiu de simulare personalizat**

Ce se întâmplă dacă aveți nevoie de o recomandare în acest moment?

- utilizați **recomandări** din studii de simulare la scară largă
- realizați un **studiu de simulare personalizat**



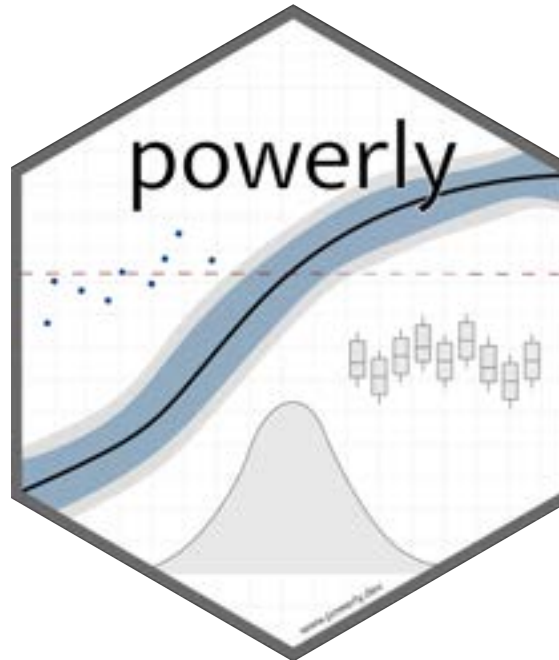
acest lucru nu este întotdeauna simplu...

Obiectivul este de a furniza o **metodă generală** pentru **analiza mărimii eșantionului** pentru **modelele de rețea** transversale.

Propunem o metodă **Monte Carlo** (MC) în trei pași care

- caută iterativ o mărime optimă a eșantionului
- concentrează eficient simulările MC pe mărimi de eșantion relevante
- se extinde ușor la diverse modele și măsurători de performanță

- un pachet **R** open-source



powerly.dev

Input pentru Metodă



- un **true model** Θ care reprezintă matricea de *edge weights*

- un true model Θ
- o măsură de performanță de interes exprimată precum o funcție $f(\Theta, \hat{\Theta})$ care compară parametrii din true model Θ cu parametrii din modelul estimat $\hat{\Theta}$

Target pentru $f(\Theta, \hat{\Theta})$

Input pentru Metodă

- un true model Θ
- o măsură de performanță $f(\Theta, \hat{\Theta})$
- o valoare **target** δ care indică valoarea dorită pentru $f(\Theta, \hat{\Theta})$

- un true model Θ
- o măsură de performanță $f(\Theta, \hat{\Theta})$
- o valoare target δ pentru $f(\Theta, \hat{\Theta})$
- o **definiție pentru puterea statistică** exprimată precum o **statistică** $g(\xi)$ cu ξ un vector de măsurători de performanță pentru o anumită mărime a eșantionului

Target pentru $g(\xi)$

Input pentru Metodă

- un true model Θ
- o măsură de performanță $f(\Theta, \hat{\Theta})$
- o valoare țintă δ pentru $f(\Theta, \hat{\Theta})$
- o statistică $g(\xi)$
- o **valoare target** τ pentru statistica $g(\xi)$

- un **true model** 🗯

```
10 x 10 sparse Matrix of class "dsCMatrix"
[1,] . -0.31 . . . . . -0.26 .
[2,] -0.31 . . 0.30 0.18 . . . .
[3,] . . . . . . . -0.22 -0.31
[4,] . 0.30 . . . . . . 0.23
[5,] . 0.18 . . . . . -0.23 -0.24
[6,] . . . . . . . 0.24 . 0.24
[7,] . . . . . -0.23 0.24 . . -0.23
[8,] . . . . . -0.23 0.24 . . -0.23
[9,] -0.26 . -0.22 . -0.24 . . . .
[10,] . . -0.31 0.23 . 0.24 . -0.23 . .
```

Exemplu de Input

Input pentru Metodă

- un **true model** Θ
- o **măsură de performanță** $f(\Theta, \hat{\Theta})$
- o **valoare target** δ pentru $f(\Theta, \hat{\Theta})$

```
10 x 10 sparse Matrix of class "dsCMatrix"
[1,] . -0.31 . . . . . -0.26 .
[2,] -0.31 . . 0.30 0.18 . . . .
[3,] . . . . . . . -0.22 -0.31
[4,] . 0.30 . . . . . . 0.23
[5,] . 0.18 . . . . . -0.23 -0.24
[6,] . . . . . . . 0.24 . 0.24
[7,] . . . . . . . . . .
[8,] . . . . -0.23 0.24 . . -0.23
[9,] -0.26 . -0.22 . -0.24 0.24 . .
[10,] . . -0.31 0.23 . 0.24 . -0.23 .
```

sensibilitate de 0.6

Exemplu de Input

Input pentru Metodă

- un **true model** Θ
- o **măsură de performanță** $f(\Theta, \hat{\Theta})$
- o **valoare target** δ pentru $f(\Theta, \hat{\Theta})$
- o **statistică** $g(\xi)$
- o **valoare target** τ pentru $g(\xi)$

```
10 x 10 sparse Matrix of class "dsCMatrix"
[1,] . -0.31 . . . . -0.26 .
[2,] -0.31 . . 0.30 0.18 . . .
[3,] . . . . . . -0.22 -0.31
[4,] . 0.30 . . . . . 0.23
[5,] . 0.18 . . . . -0.23 -0.24
[6,] . . . . . . 0.24 . 0.24
[7,] . . . . . . . . .
[8,] . . . -0.23 0.24 . . -0.23
[9,] -0.26 . -0.22 . -0.24 0.24 . .
[10,] . . -0.31 0.23 . 0.24 . -0.23 .
```

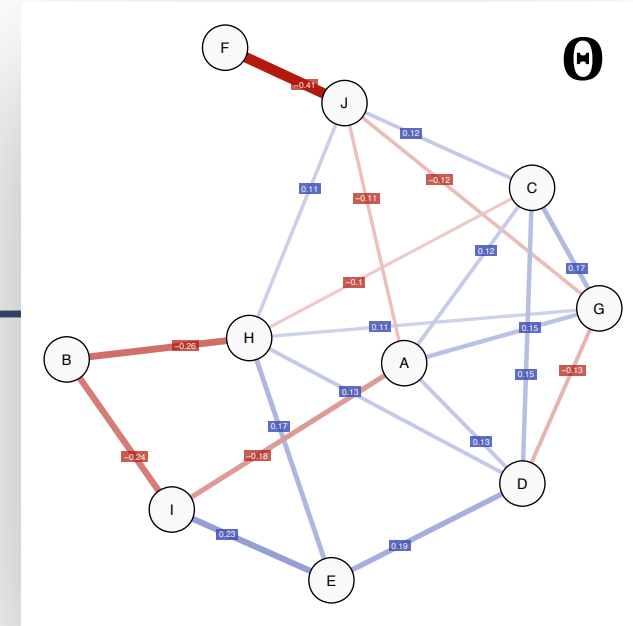
sensibilitate de 0.6

probabilitate de 0.8

Exemplu de Input

Input pentru Metodă

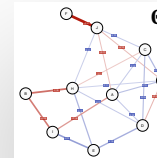
- un **true model** Θ
- o **măsură de performanță** $f(\Theta, \hat{\Theta})$
- o **valoare target** δ pentru $f(\Theta, \hat{\Theta})$
- o **statistică** $g(\xi)$
- o **valoare target** τ pentru $g(\xi)$



Pentru un anumit Θ ipotetic, ce mărime a eșantionului este necesară pentru a observa $f(\Theta, \hat{\Theta}) \geq \delta$ conform $g(\xi) \geq \tau$?

Pentru un anumit Θ ipotetic, ce mărime a eșantionului este necesară pentru a observa

$$f(\Theta, \hat{\Theta}) \geq \delta \text{ conform } g(\xi) \geq \tau?$$



Plecând de la o ipoteză privind un model de rețea, ce mărime a eșantionului avem nevoie pentru a observa o sensibilitate de δ cu o putere statistică de τ ?

Pașii Metodei

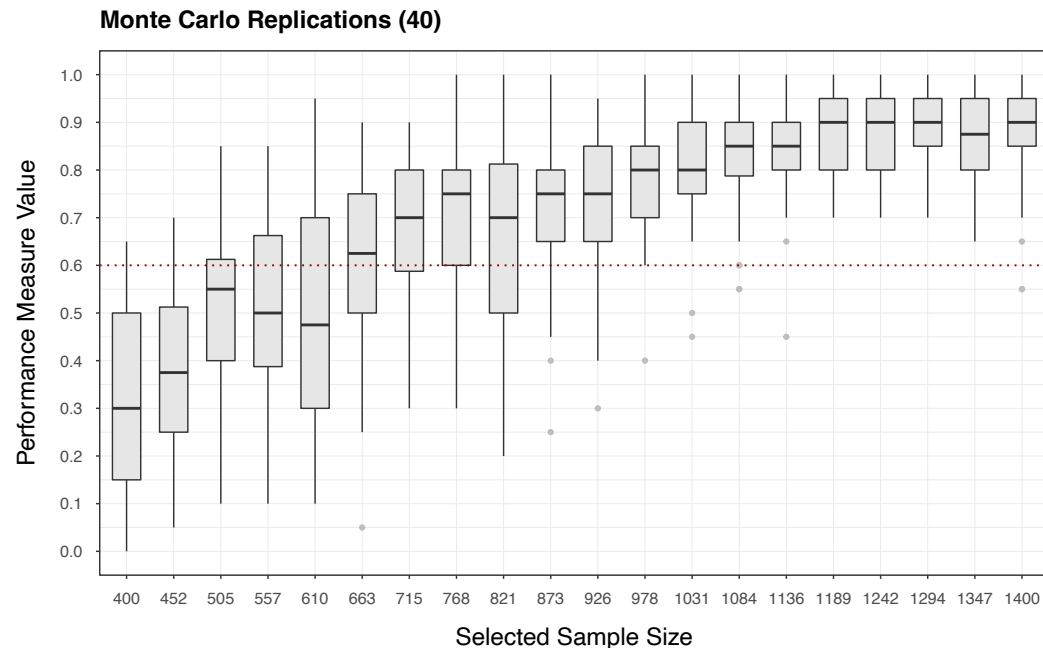


Rolul acestui pas este să înțelegem cum se **comportă** $f(\Theta, \hat{\Theta})$
în funcție de mărimea eșantionului.

- începem cu un **interval candidat** $\mathbb{N}_s$ pentru mărimea eșantionului
- selectăm T mărimi de eșantion echidistante $S = \{s_1, \dots, s_T\} \subseteq \mathbb{N}_s$

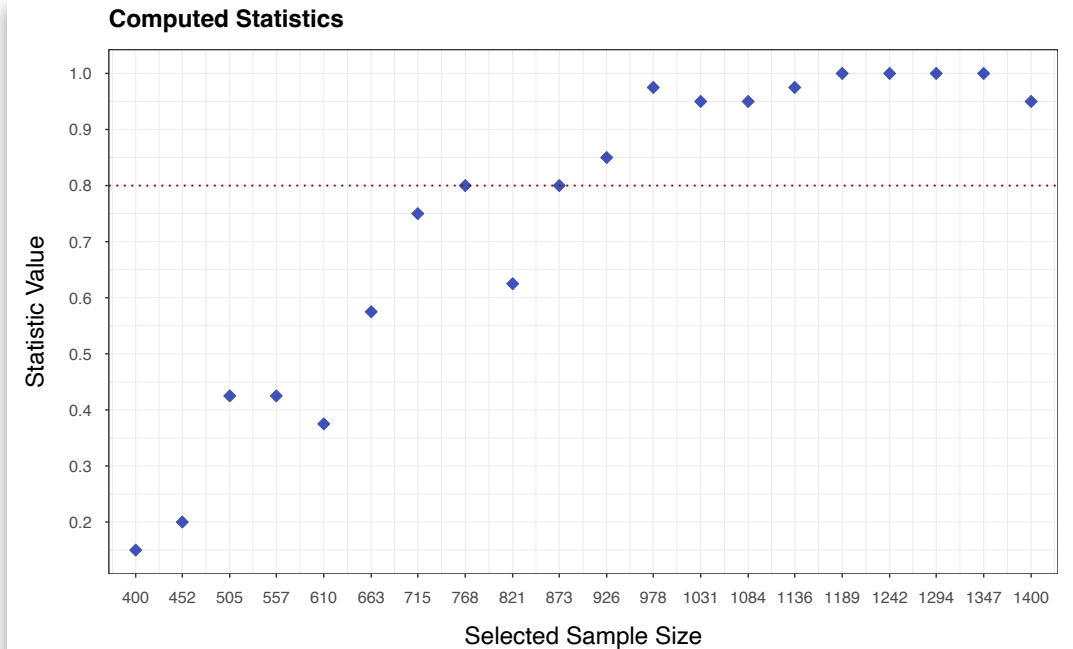
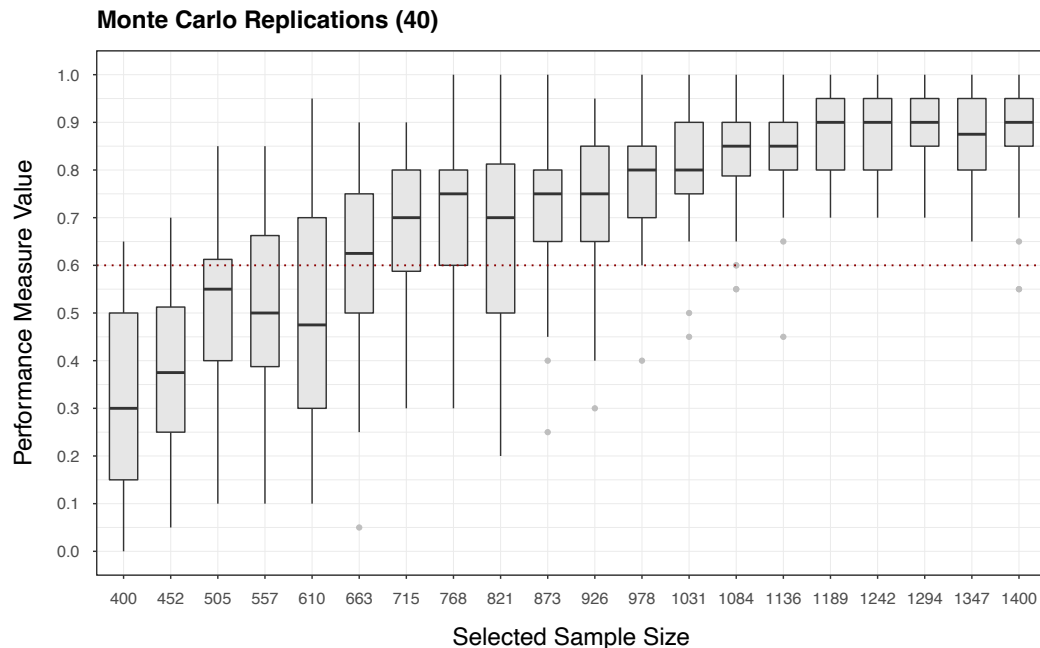
- începem cu un interval candidat $\mathbb{N}_s$ pentru mărimea eșantionului
- selectăm T mărimi de eșantion echidistante $S = \{s_1, \dots, s_T\} \subseteq \mathbb{N}_s$
- pentru fiecare $s_t \in S$ **efectuăm R replicări MC** după cum urmează:
 - generăm o bază de date cu s_t număr de cazuri folosind Θ
 - estimăm $\hat{\Theta}$ folosind baza de date generată
 - calculăm $f(\Theta, \hat{\Theta})$

- obținem o matrice $\mathbf{E}_{R \times T}$, unde fiecare element reprezintă o măsură de performanță calculată în timpul unei replicări MC pentru o mărime de eșantion



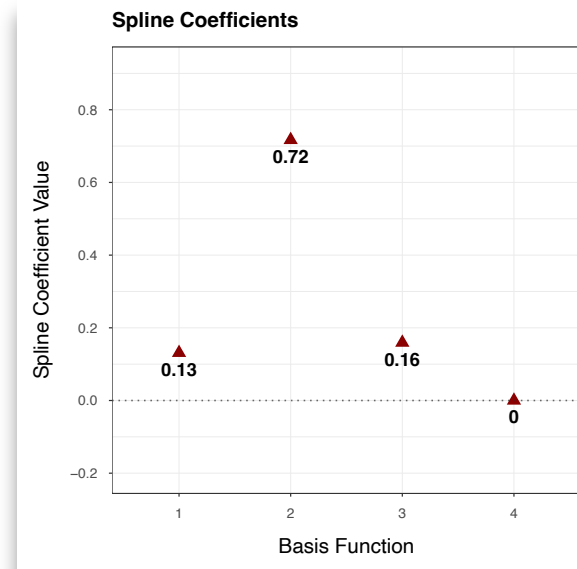
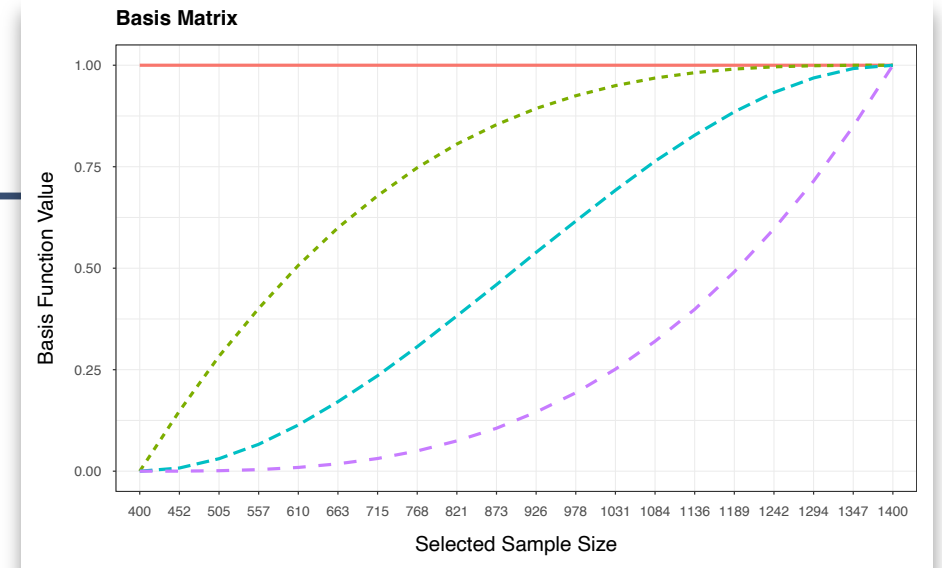
- obținem o matrice $\Xi_{R \times T}$, unde fiecare element reprezintă o măsură de performanță calculată în timpul unei replicări MC pentru o mărime de eșantion

- aplicăm $g(\xi)$ pe fiecare coloană din Ξ pentru a **calcula statistica** (e.g., puterea)

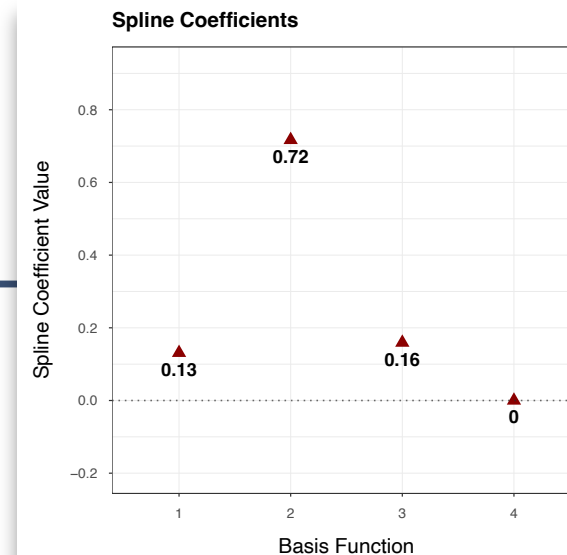
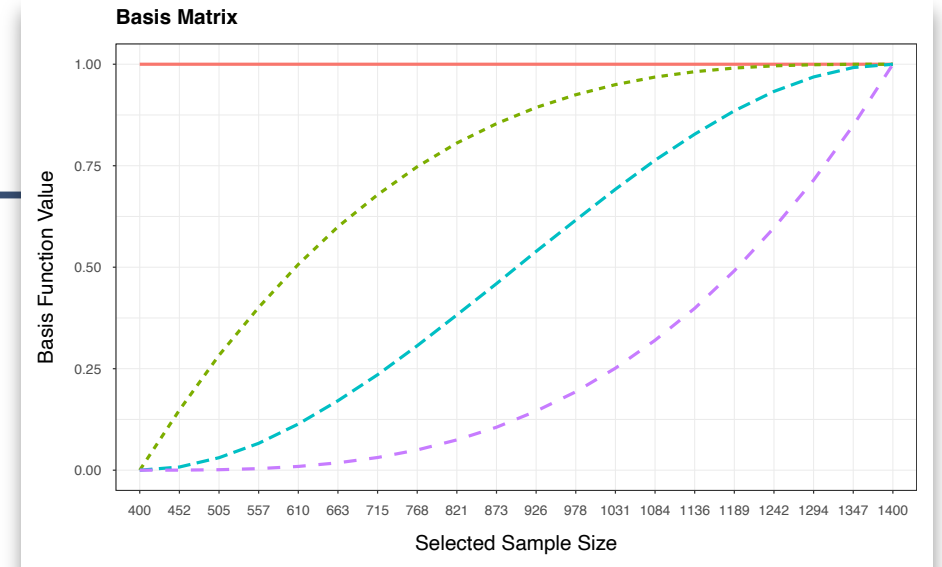
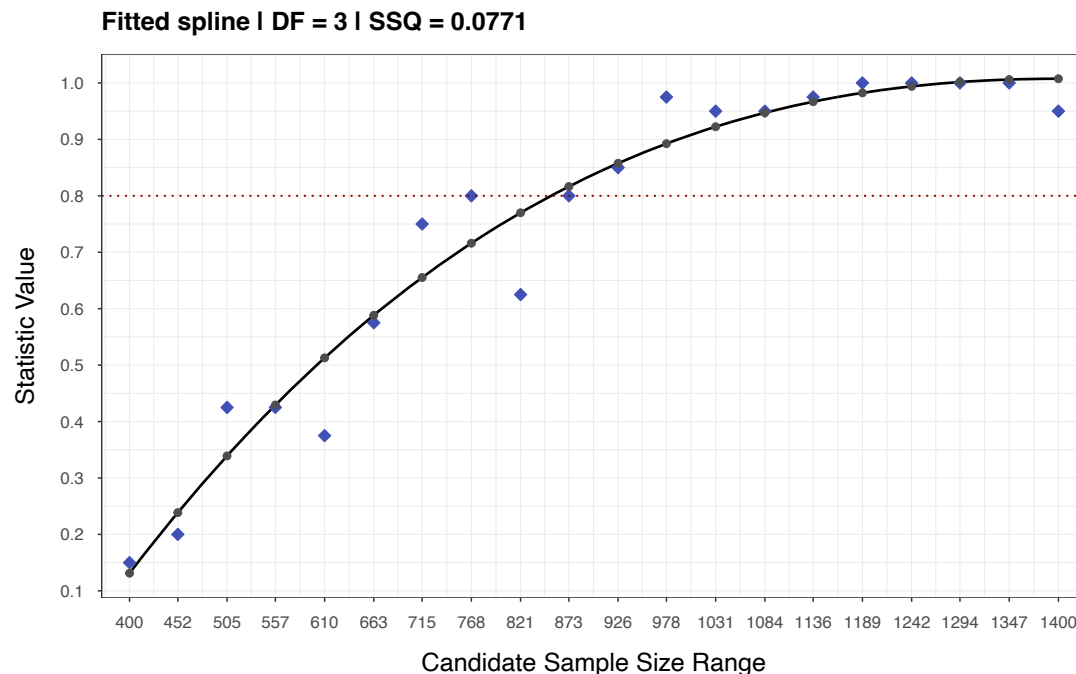


Rolul acestui pas este de a obține o funcție continuă (puterea) și de a **interpola statistica** pentru **toate mărimile de eșantion** din intervalul N_s .

- **assume monotonicity** și folosim cubic *I-Spline* bases cu inner knots selectate prin cross-validation



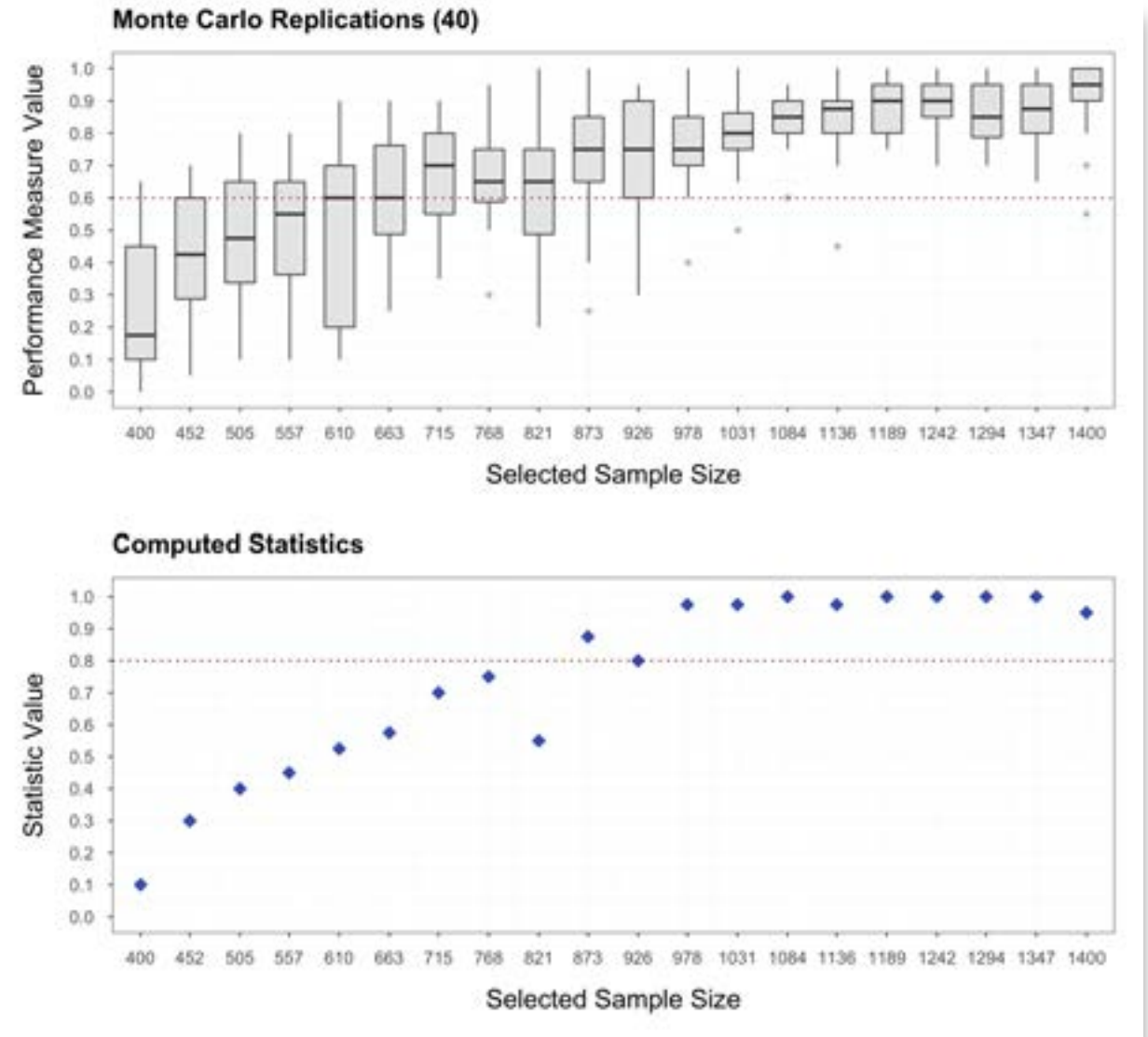
- **assume monotonicity** și folosim cubic *I-Spline* bases cu inner knots selectate prin cross-validation



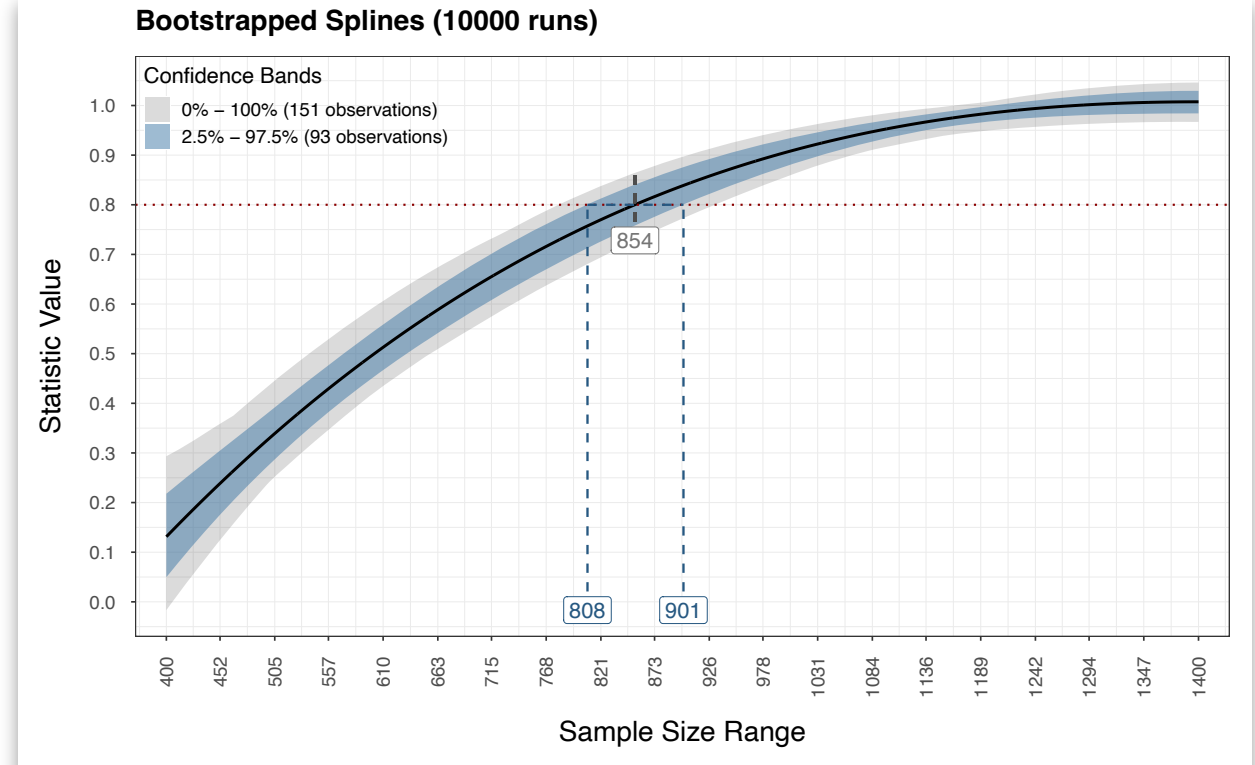
Scopul acestui pas este de a **lua în considerare eroarea MC** și de a cuantifica incertitudinea din jurul funcției interpolate.

- utilizăm **stratified bootstrapping** să reprezentăm variabilitatea măsurilor de performanță replicate pentru fiecare mărime de eșantion $s_t \in S$

- utilizăm stratified bootstrapping să reprezentăm variabilitatea măsurilor de performanță replicate pentru fiecare mărime de eșantion $s_t \in S$
- rulăm bootstrap pe **măsurile de performanță**, nefiind necesar să re-estimăm modelul

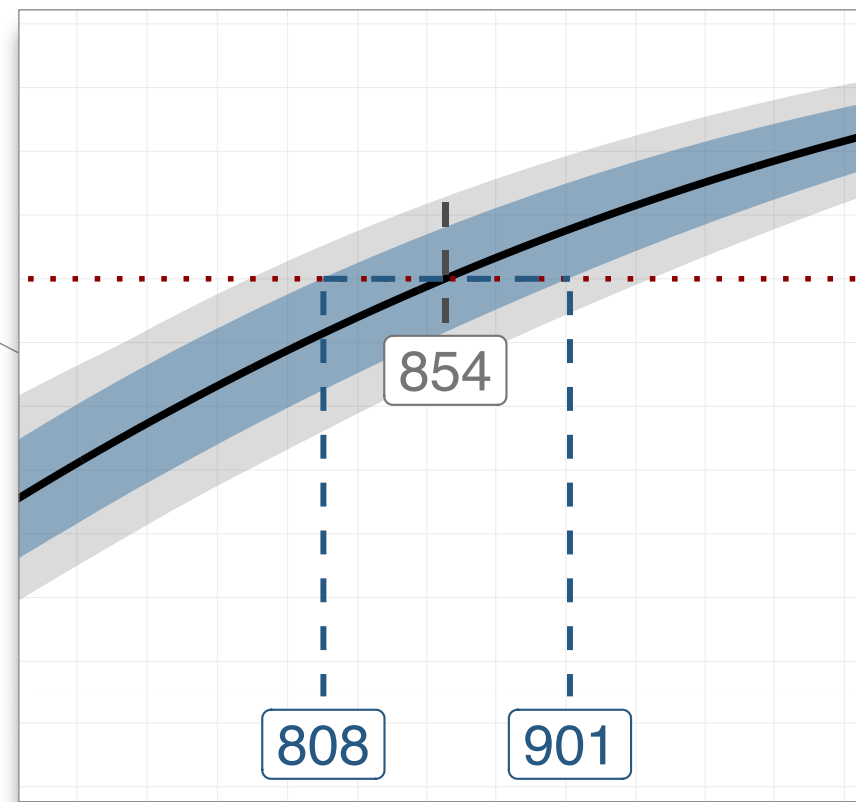
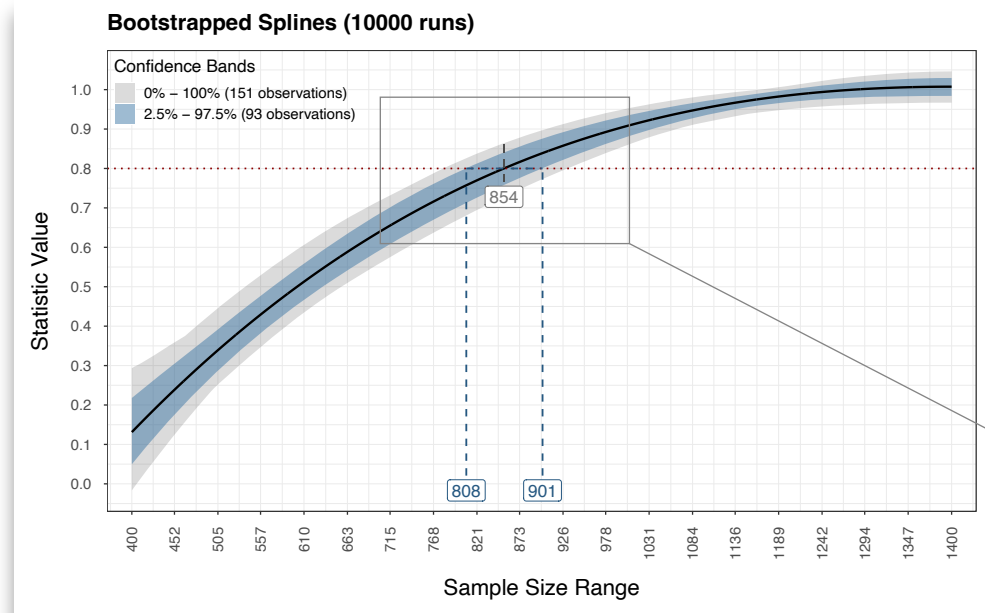


- utilizăm stratified bootstrapping să reprezentăm variabilitatea măsurilor de performanță replicate pentru fiecare mărime de eșantion $s_t \in S$
- rulăm bootstrap pe măsurile de performanță, nefiind necesar să re-estimăm modelul
- estimăm un **nou spline** pentru fiecare matrice de măsurători de performanță obținută prin bootstrap

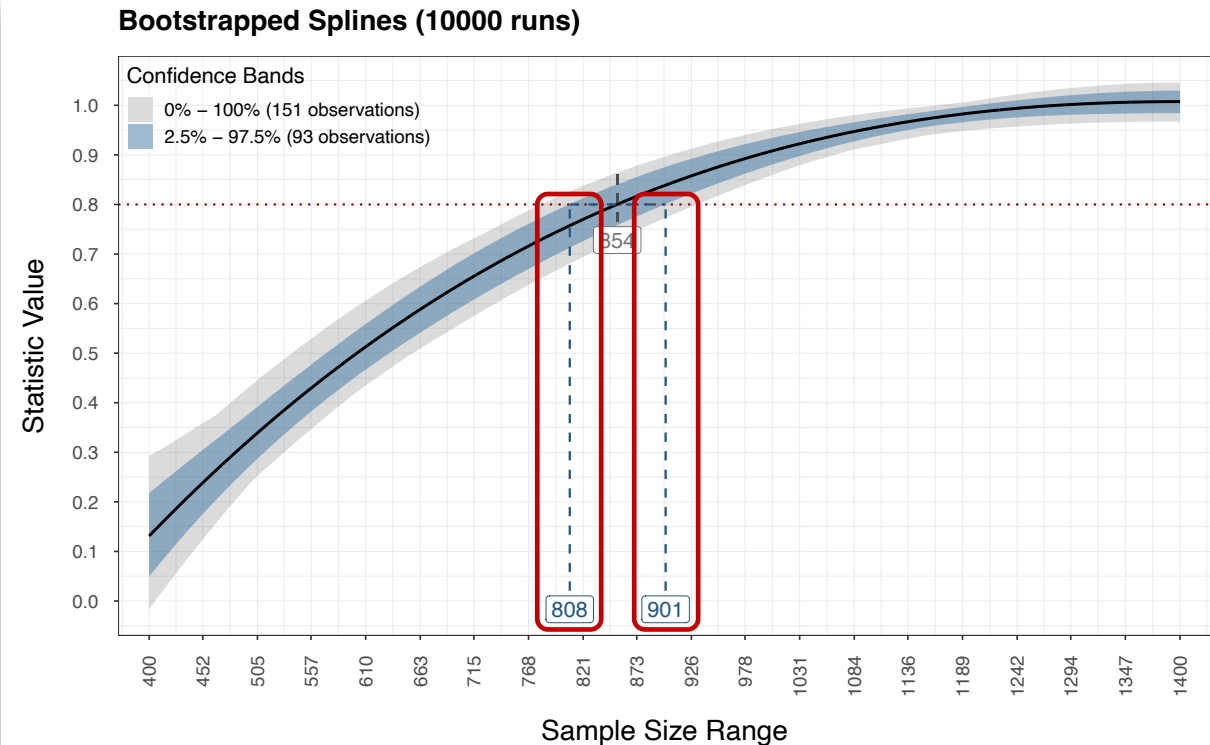


Pasul 3

Pașii Metodei



- **actualizăm intervalul candidat** N_s pe baza benzilor de încredere
- **repetăm** Pașii 1 până la 3 până când N_s devine suficient de mic



```
# Încărcăm librăria.  
library(powerly)
```

```
# Încărcăm librăria.  
library(powerly)
```

```
# Generăm un true model.  
true_model <- generate_model(  
  type = "ggm",  
  nodes = 10,  
  density = .4  
)
```

```
# Încărcăm librăria.  
library(powerly)
```

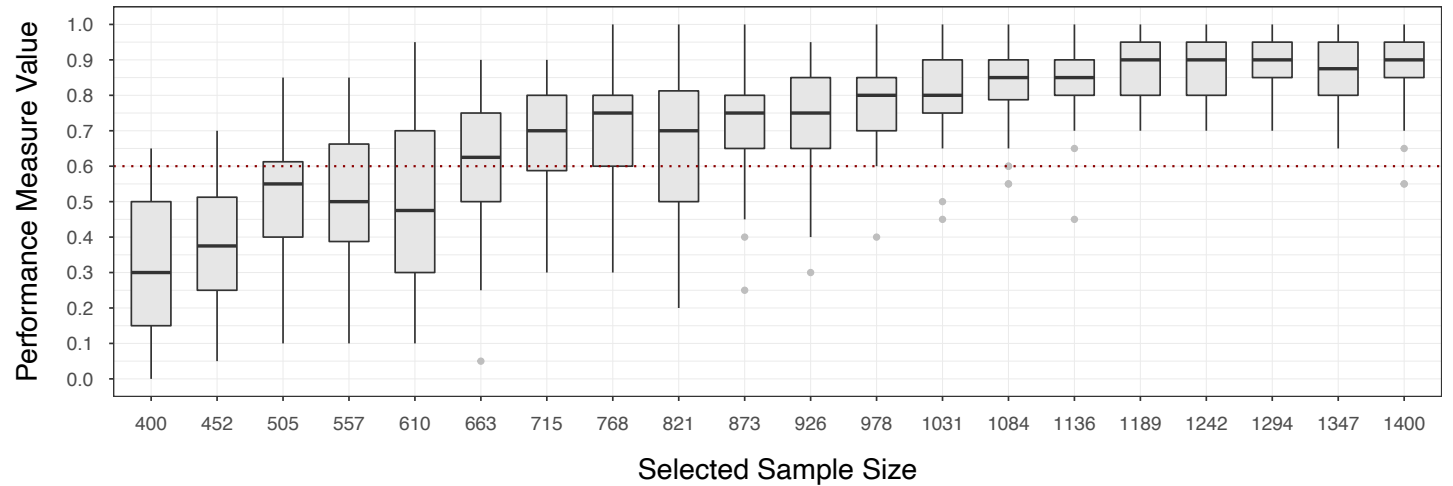
```
# Generăm un true model.  
true_model <- generate_model(  
  type = "ggm",  
  nodes = 10,  
  density = .4  
)
```

```
# Rulăm metoda.  
results <- powerly(  
  range_lower = 300,  
  range_upper = 1000,  
  samples = 30,  
  replications = 20,  
  measure = "sen",  
  statistic = "power",  
  measure_value = .6,  
  statistic_value = .8,  
  model = "ggm",  
  model_matrix = true_model  
)
```

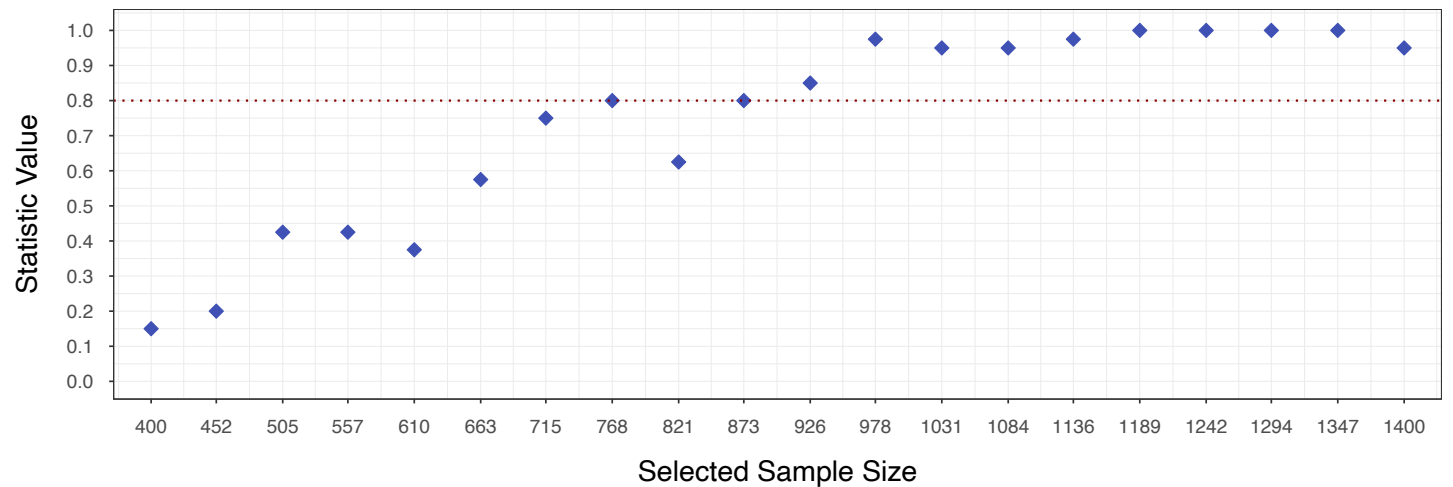


```
plot(results, step = 1)
```

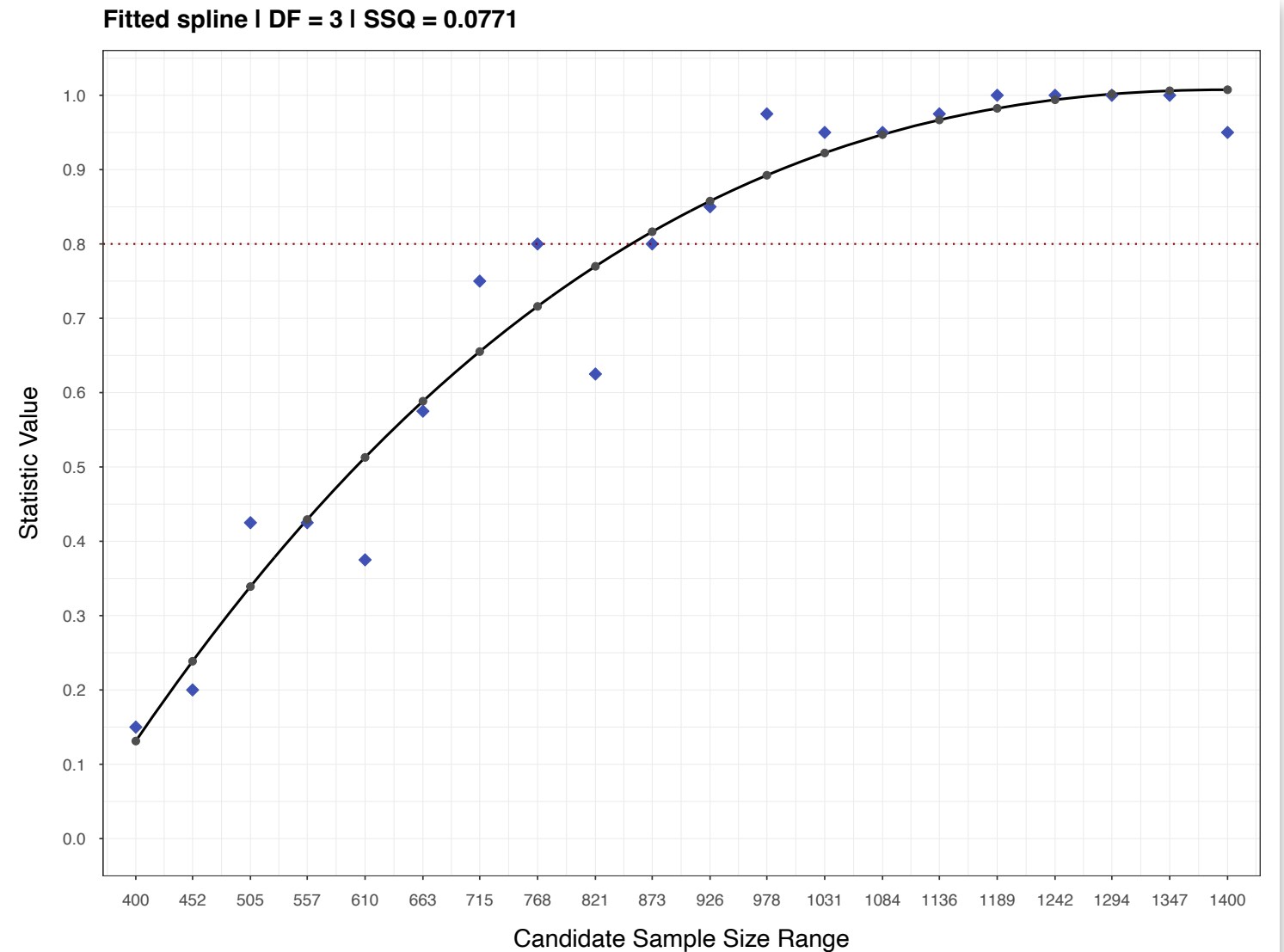
Monte Carlo Replications (40)



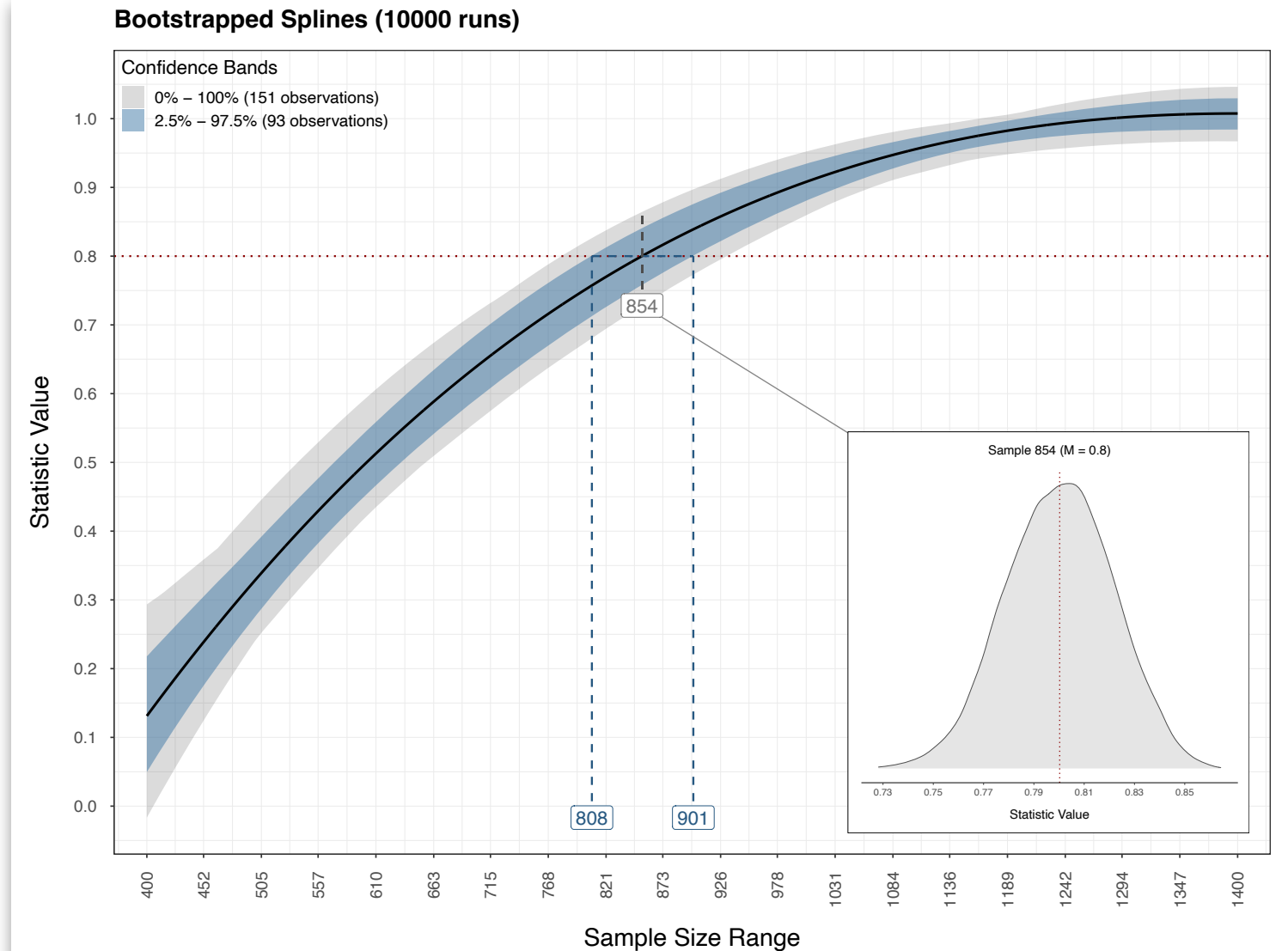
Computed Statistics



```
plot(results, step = 2)
```



```
plot(results, step = 3)
```



Studiu de Simulare

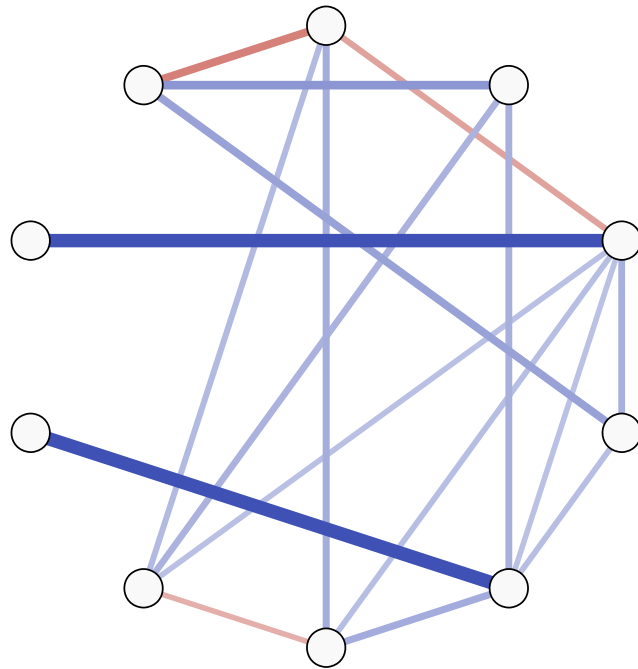


$$T = \{20, 40, 60\}$$

$$R = \{10, 30, 50\}$$

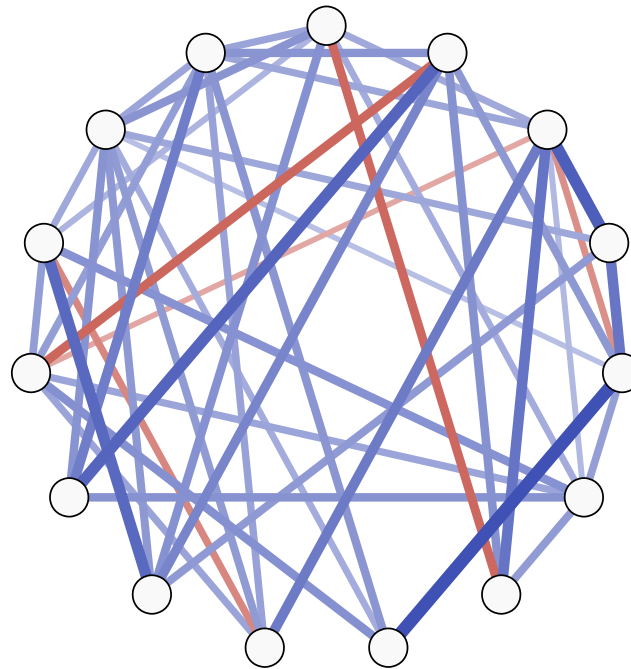
×

True Model 1



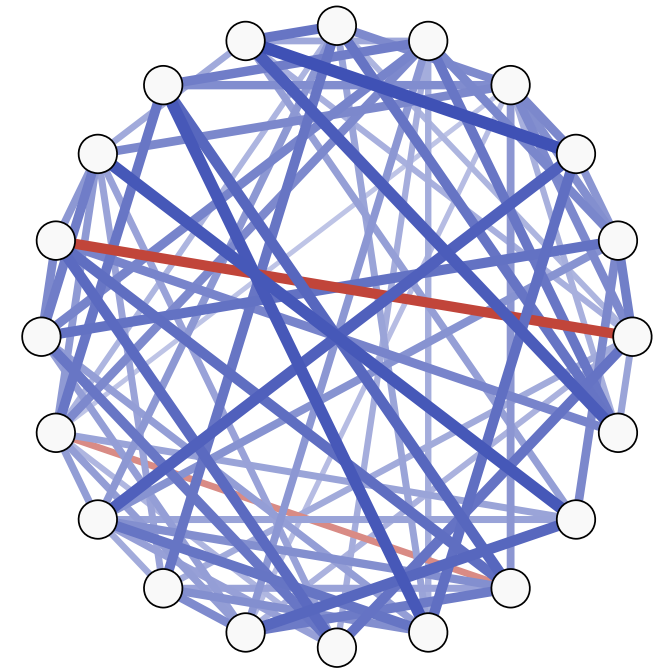
17 parameters

True Model 2

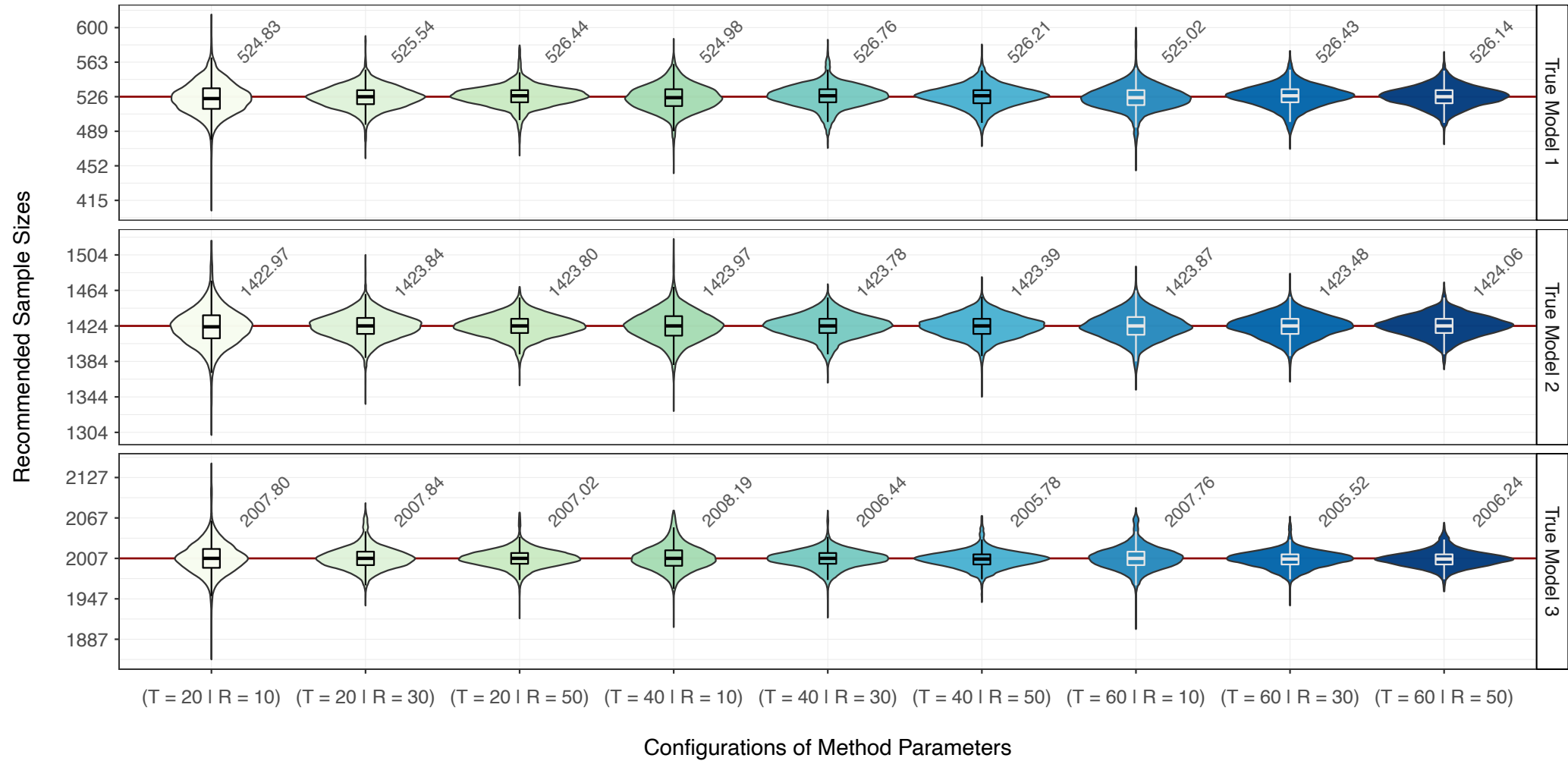


45 parameters

True Model 3



74 parameters





© 2023 American Psychological Association
ISSN: 1082-989X

Psychological Methods

<https://doi.org/10.1037/met0000555>

A General Monte Carlo Method for Sample Size Analysis in the Context of Network Models

Mihai A. Constantin¹, Noémi K. Schuurman², and Jeroen K. Vermunt¹

¹Department of Methodology and Statistics, Tilburg University

²Department of Methodology and Statistics, Utrecht University

Abstract

We introduce a general method for sample size computations in the context of cross-sectional network models. The method takes the form of an automated Monte Carlo algorithm, designed to find an optimal sample size while iteratively concentrating the computations on the sample sizes that seem most relevant. The method requires three inputs: (1) a hypothesized network structure or desired characteristics of that structure, (2) an estimation *performance measure* and its corresponding target value (e.g., a sensitivity of 0.6), and (3) a statistic and its corresponding target value that determines how the target value for the performance measure be reached (e.g., reaching a sensitivity of 0.6 with a probability of 0.8). The method consists of a Monte Carlo simulation step for computing the performance measure and the statistic for several sample sizes selected from an initial candidate sample size range, a curve-fitting step for interpolating the statistic across the entire candidate range, and a stratified bootstrapping step to quantify the uncertainty around the recommendation provided. We evaluated the performance of the method for the Gaussian Graphical Model, but it can easily extend to other models. The method displayed good performance, providing sample size recommendations that were, on average, within three observations of a benchmark sample size, with the highest standard deviation of 25.87 observations. The method discussed is implemented in the form of an R package called **powerly**, available on [GitHub](#) and [CRAN](#).

I



Directii Viitoare



Pentru un anumit Θ ipotetic, ce mărime a
eșantionului este necesară pentru a observa
$$f(\Theta, \hat{\Theta}) \geq \delta \text{ conform } g(\xi) \geq \tau?$$

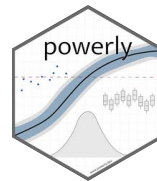
Către un Sistem Deschis

- un **API** pentru a extinde metoda la **runtime** cu
 - noi tipuri de modele
 - noi măsurători de performanță
- un tutorial în care arătăm cum se
 - implementează acest **API** pentru SEM via **lavaan**
 - și abordăm întrebări de cercetare dificile
- crearea unui **ecosystem**



by [Marie Mainguy](#)

Întrebări?



powerly.dev

