A General Approach to Sample Size Analysis

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

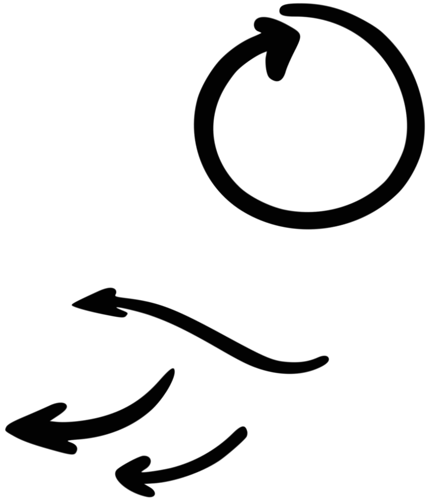
mihai@mihaiconstantin.com

Tilburg University

So Far

We've seen:

- what sample size planning is and why it matters
- two criteria for searching for an optimal sample size
  - statistical power
  - predictive accuracy
- two approaches for conducting sample size analysis
  - analytical
  - simulation
- applications to time series models, i.e., $AR(1)$ and $VAR(1)$

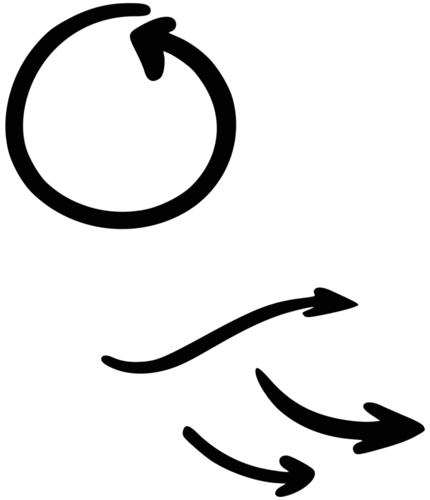


So Next

We'll talk about:

- the requirements of simulation-based sample size analysis
  - ...and the questions we can formulate
- a general method to answer sample size questions
  - ...and obtain recommendation
- a software implementation
  - ...and an example

- end with a `/sample(?=size)/`



Simulation Approaches

Typical Monte Carlo Setup

- the process goes as follows:
  - select true parameter values for your model
  - generate one dataset with the true parameters
  - estimate the model parameters
  - test your hypothesis

Typical Monte Carlo Setup

- the process goes as follows:
  - select true parameter values for your model
  - generate one dataset with the true parameters
  - estimate the model parameters
  - test your hypothesis

→ repeat many times

↓

calculate
empirical power

We’ve seen this before…

Steps of the simulation-based approach

- Example: select the number of measurement occasions T to test if the autoregressive effect of PA is positive

Given T, Hypothesis of interest (e.g., $H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 > 0$), and α

Step 1	Step 2	Step 3	Step 4	Step 5
Select values for the parameters of the AR(1) model	Generate large amounts of data sets (e.g., 1000) using the AR(1) model and parameter values	For each of these data sets estimate AR(1) model	For each of estimated model test the hypothesis of interest	Calculate statistical power: % of data sets in which the H_0 is rejected
$\beta_0^*, \beta_1^*, \sigma_\epsilon^*$ (usually based on pilot data of data from prior studies)	$y_t = \beta_0^* + \beta_1^* y_{t-1} + \epsilon_t$	OLS	$H_0\colon \beta_1 = 0$ $H_1\colon \beta_1 > 0$	$\dfrac{\#\ data\ sets\ H_0\ is\ rejected}{Total\ \#\ of\ data\ sets}$

…and even how to code it

The Monte Carlo simulation function

This function conducts the Monte Carlo simulation for a set of sample sizes (i.e., several different number of observations) and computes the statistical power for a given hypothesis. It takes several arguments as follows:

- `vars` is the number of variables of the $\mathrm{VAR}(1)$ model
- `Tobs_list` is a list of numbers of repeated measurements (i.e., `Tobs`)
- `delta` the intercept matrix
- `psi` the transition matrix (which contains the auto-regressive and cross-regressive effects)
- `sigma` the variance-covariance matrix of the innovation
- `R` is the number of Monte Carlo replicates (e.g., 1000)
- `alpha` is the Type I error rate (or significance level of a test statistic)

```
# Function to conduct the Monte Carlo power simulation.
mc_power <- function(vars, Tobs_list, delta, psi, sigma, R, alpha) {
    # Prepare simulation storage.
    df_pow <- data.frame()

    # For each sample size in the list.
    for (i in 1:length(Tobs_list)) {
        # Extract the sample size.
        Tobs <- Tobs_list[i]

        # Print the progress.
        print(paste0("Power analysis for N = ", Tobs))

        # For each Monte Carlo replication.
        for (r in 1:R) {
            # Generate data.
            data <- sim_VAR_data(vars, Tobs, delta, psi, sigma)
```

Typical Monte Carlo Setup

We still leverage this this setup…

- but thinking about it more generally
- along two acts

Typical Monte Carlo Setup

Simulation Approaches

We still leverage this this setup…

- but thinking about it more generally
- along two acts

the first act

What is the required input for running a simulation-based power analysis?

Typical Monte Carlo Setup

We still leverage this this setup…

- but thinking about it more generally
- along two acts

the second act

How can we process the input to get a sample size recommendations?

The Requirements

what

At a Glance

The Requirements

For a simulation approach we need to:

- generate or specify true model parameters
- generate data based on the true model parameters
- estimate model parameters from data
- specify a performance measure of interest
- specify a working definition for power

At a Glance

The Requirements

For a simulation approach we need to:

- generate or specify true model parameters
- generate data based on the true model parameters
- estimate model parameters from data
- specify a performance measure of interest
- specify a working definition for power

what to be able to perform

At a Glance

The Requirements

For a simulation approach we need to:

- generate or specify true model parameters
- generate data based on the true model parameters
- estimate model parameters from data
- specify a performance measure of interest
- specify a working definition for power

what to be able to provide

True Model Parameters

The Requirements

- the set of hypothesized model values used to generate data
- akin to an effect size in typical power analysis
- let’s call it **Θ**

Generated Data

The Requirements

- the observed data
  - is a sample from a data generating process with unknown parameters
- the generated data
  - is what we get when we pretend to know
    - the data generating process and
    - the values of its parameters → our hypothesized Θ
  - we typically generate datasets of varying sizes for a given Θ

Generated Data

The Requirements

- the observed data
  - is a sample from a data generating process with unknown parameters
- the generated data
  - is what we get when we pre
    - the data generating proce
    - the values of its parameters → our hypothesized θ
  - we typically generate datasets of varying sizes for a given θ

What do you think we need the generated data for?

Estimated Model Parameters

The Requirements

- if Θ represents the hypothesized true model parameters
- then $\hat{\Theta}$ holds the estimated model parameters
  - estimated from the generated data

Estimated Model Parameters

The Requirements

- if Θ represents the hypothesized true model parameters
- then $\hat{\Theta}$ holds the estimated model parameters
  - estimated from the generated data

What does your intuition say will happen to $\hat{\Theta}$ if the generated dataset is very large?

Performance Measure

The Requirements

- is a statement about the data generating process
  - quantifies the quality of the estimation
- expressed as $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$ that
  - compares the true model parameters in $\boldsymbol{\Theta}$ to the estimated model parameters in $\widehat{\boldsymbol{\Theta}}$
  - and the result of this comparison is dependent on the sample size

Performance Measure

The Requirements

- is a statement about the data generating process
- expressed as $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$ that
  - compares the true model parameters in $\boldsymbol{\Theta}$ to the estimated model parameters in $\widehat{\boldsymbol{\Theta}}$
  - and the result of t

How is the performance measure $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$ connected to the sample size?

Performance Measure

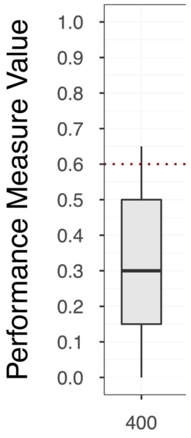
The Requirements

- is a statement about the data generating process
- expressed as $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$ that
  - compares the true model parameters in $\boldsymbol{\Theta}$ to the estimated model parameters in $\widehat{\boldsymbol{\Theta}}$
  - and the result of this comparison is dependent on the sample size
- should be driven by the research question
- has a target value δ

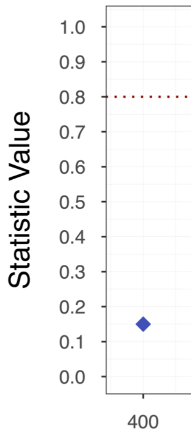
- is a definition for the empirical power
    - that tells us how we want to observe the performance measure
        - e.g., we want a sample size such that 80% of the performance measures reached the target δ

- is a definition for the empirical power
  - that tells us how we want to observe the performance measure
    - e.g., we want a sample size such that 80% of the performance measures reached the target δ
  - is expressed as a function $g(\boldsymbol{\xi})$ with a target τ
    - where

$$\boldsymbol{\xi} = \begin{bmatrix} f\left(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}\right) \\ \vdots \\ f\left(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}\right) \end{bmatrix} =$$



$$g(\boldsymbol{\xi}) =$$



what

in a nutshell

The Required Input

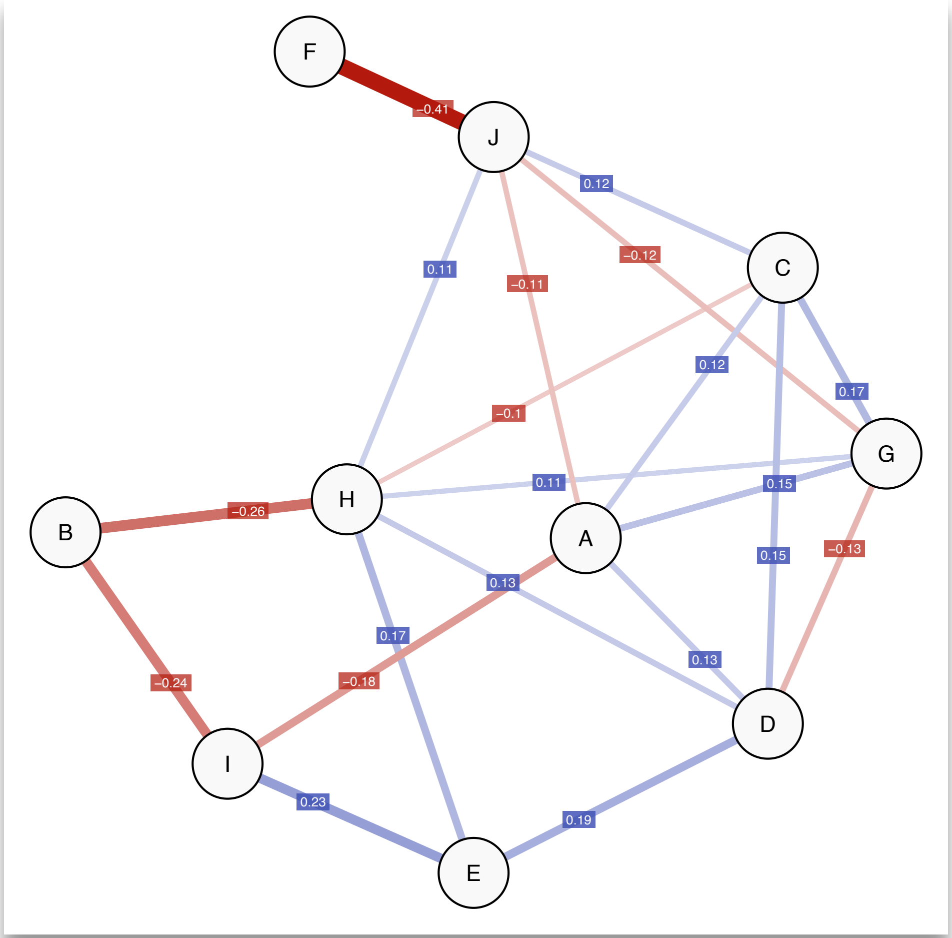
- true model Θ
  - it can be many things

The Required Input

The Requirements

- true model Θ
  - it can be many things

Gaussian Graphical Model

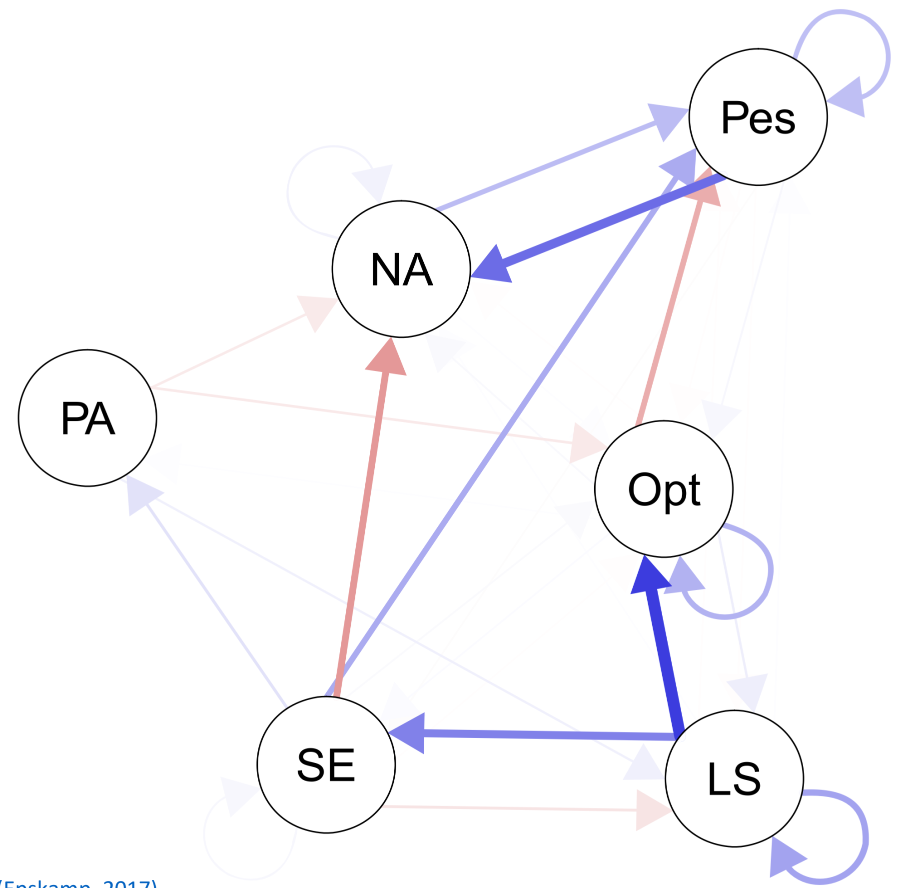


The Required Input

The Requirements

- true model Θ
  - it can be many things

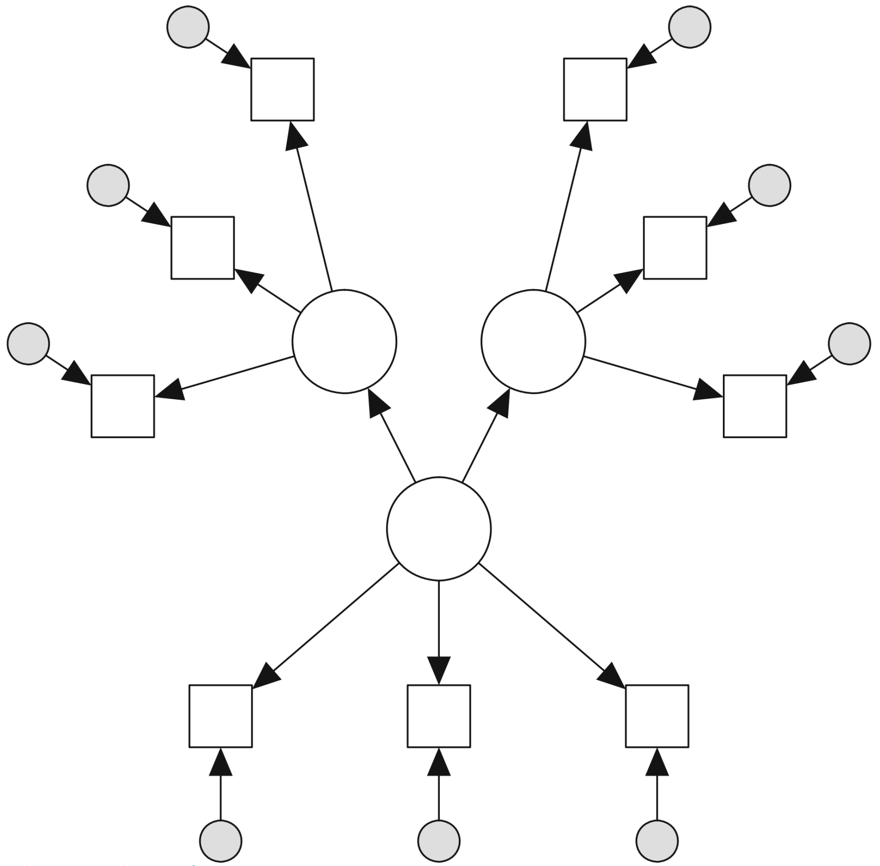
Vector Autoregressive Model



(Epskamp, 2017)

- true model $\boldsymbol{\Theta}$
  - it can be many things

Structural Equation Model



(Epskamp et al., 2017)

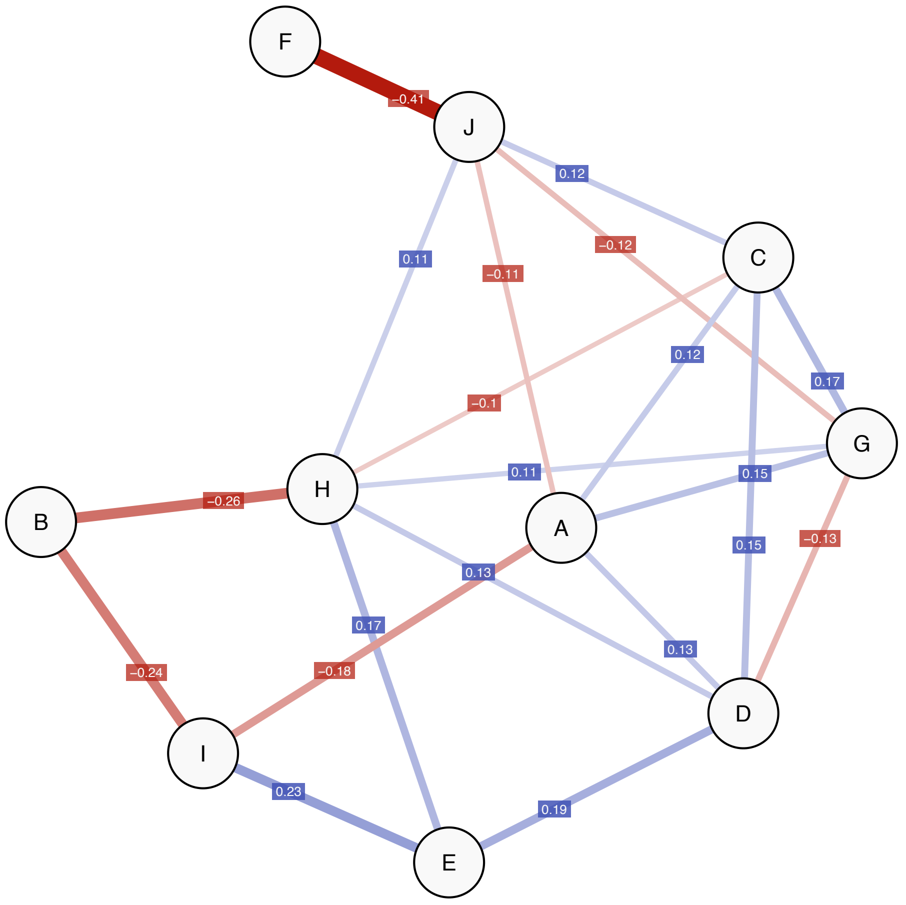
The Required Input

- true model $\boldsymbol{\Theta}$
- performance measure $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$
  - should reflect the research question

The Required Input

- true model $\boldsymbol{\Theta}$
- performance measure $f(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}})$
  - should reflect the research question
    - e.g., suppose we want to recover the network structure
    - we look at sensitivity → the proportion of edges correctly estimated to be present

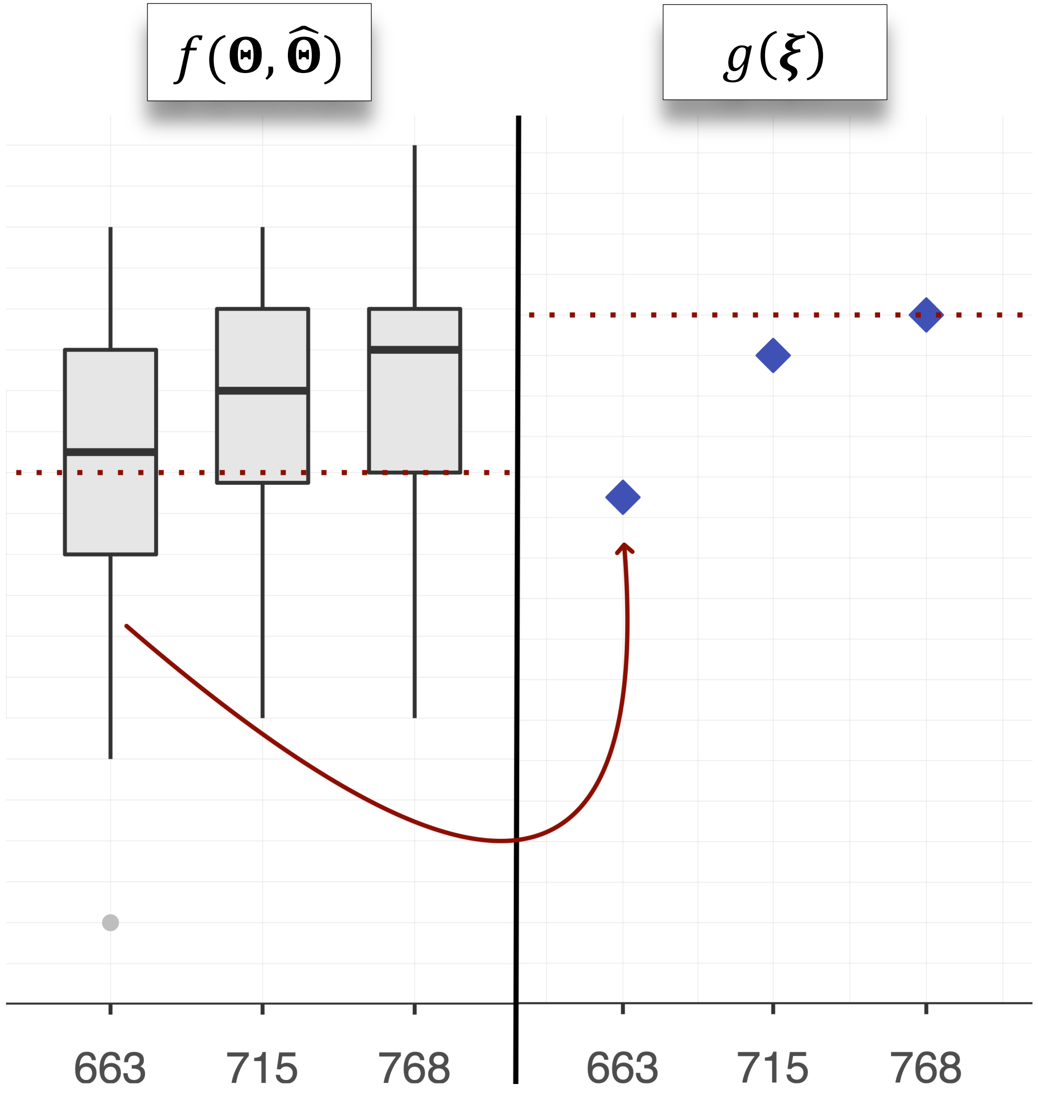
Gaussian Graphical Model



The Required Input

The Requirements

- true model $\boldsymbol{\Theta}$
- performance measure $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$
- a statistic $g(\xi)$
    - most intuitively defined as a probability, but it may take other forms



The Required Input

- true model $\boldsymbol{\Theta}$
- performance measure $f(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}})$
- a statistic $g(\boldsymbol{\xi})$

The Required Input

The Requirements

Based on this input, we can ask…

Given the hypothesized $\mathbf{\Theta}$, what sample size do we need to observe a $f\left(\mathbf{\Theta}, \widehat{\mathbf{\Theta}}\right) \geq \delta$ with probability τ as defined by $g(\boldsymbol{\xi})$?

The Required Input

The Requirements

One could ask…

I have some idea about a $VAR(1)$ model I plan to fit, and I want to test that all my autoregressive coefficients are significant with a power of 0.8. How much data do I need?

But in reality…

I have ~~some idea about~~ a $VAR\cancel{(1)}$ model ~~I plan to fit, and I want to test that all my autoregressive coefficients are significant with a power of~~ $\cancel{0.8}$~~.~~

How much data do I need?

Here’s The Deal

The Requirements

you provide us with the required input

↓

we provide you with the sample size

The Method

how

We use a three-step Monte Carlo (MC) method that

- iteratively searches for an optimal sample size
- efficiently concentrates the MC simulations on relevant sample sizes
- can extend to other models and performance measures

(Constantin et al., 2021)

Step 1

The Method

The goal of this step is to get a rough understanding of the behavior of $f\left(\mathbf{\Theta}, \widehat{\mathbf{\Theta}}\right)$ as a function of sample size.

Step 1

The Method

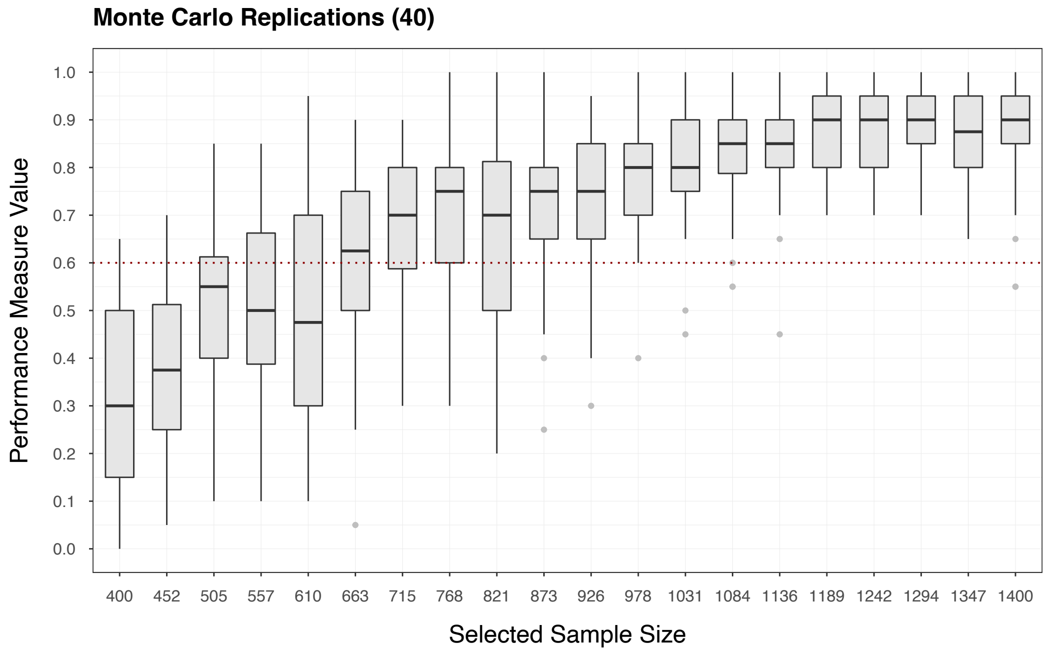
- start with a candidate sample size range $\mathbb{N}_s$
- select T equidistant samples $S = \{s_1, \dots, s_T\} \subseteq \mathbb{N}_s$

Step 1

The Method

- start with a candidate sample size range $\mathbb{N}_s$
- select T equidistant samples $S = \{s_1, \dots, s_T\} \subseteq \mathbb{N}_s$
- for each $s_t \in S$ perform R MC replications as follows:
  - generate data with s_t number of cases using $\mathbf{\Theta}$
  - estimate $\widehat{\mathbf{\Theta}}$ using the generated data
  - compute $f\left(\mathbf{\Theta}, \widehat{\mathbf{\Theta}}\right)$

- obtain $R \times T$ matrix $\boldsymbol{\Xi}$, where each entry is a performance measure computed for a sample size during a MC replication

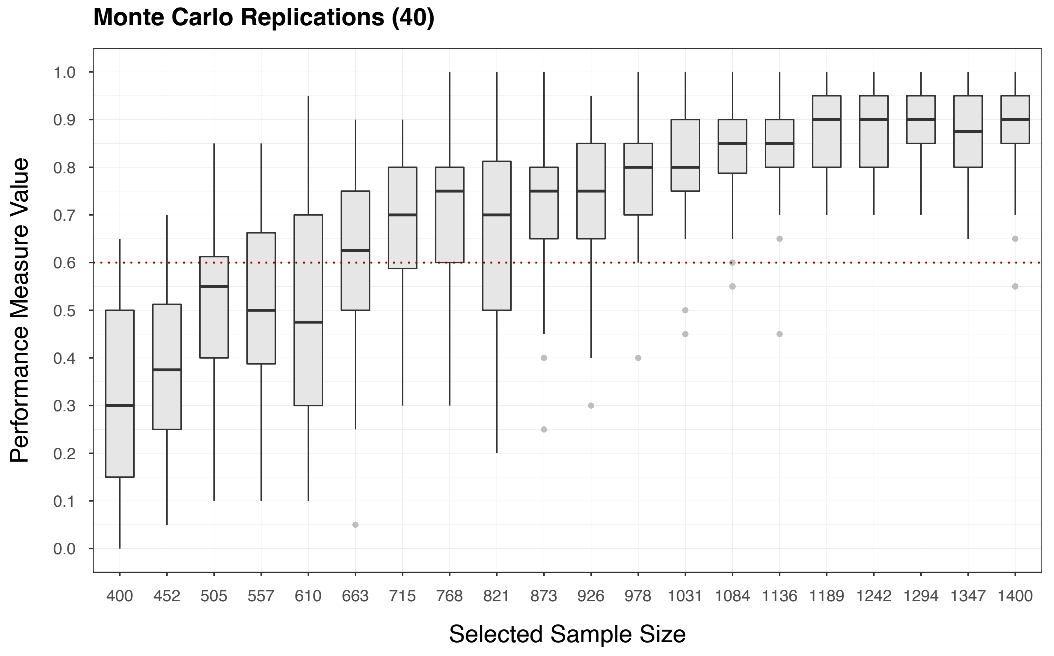


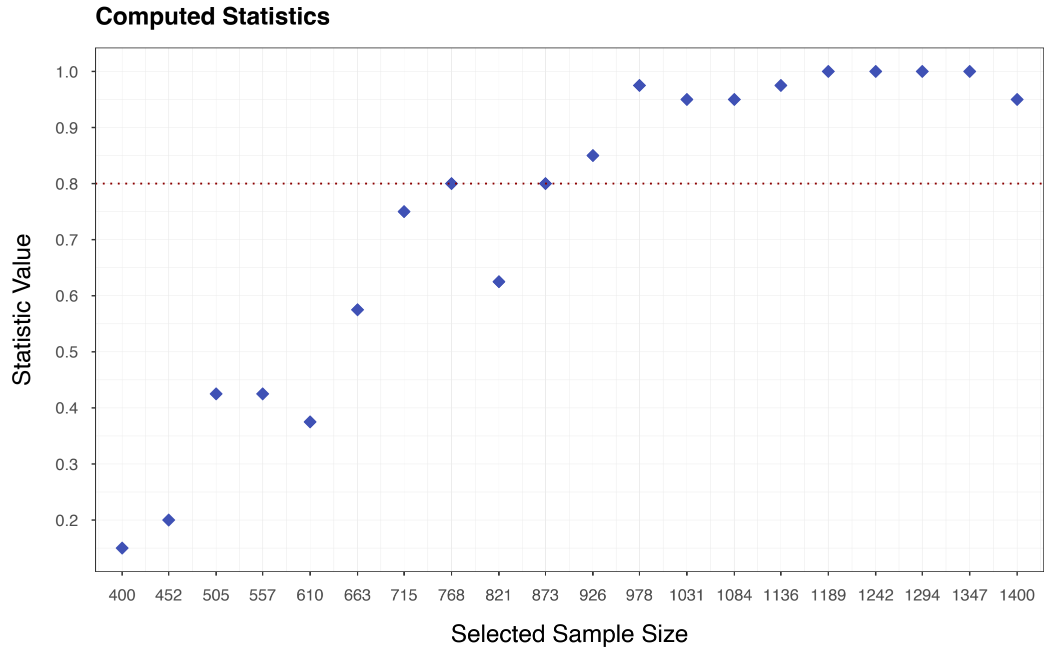
Step 1

The Method

- obtain $R \times T$ matrix Ξ, where each entry is a performance measure computed for a sample size during a MC replication

- apply $g(\xi)$ over each column of Ξ to compute the statistic (e.g., power)

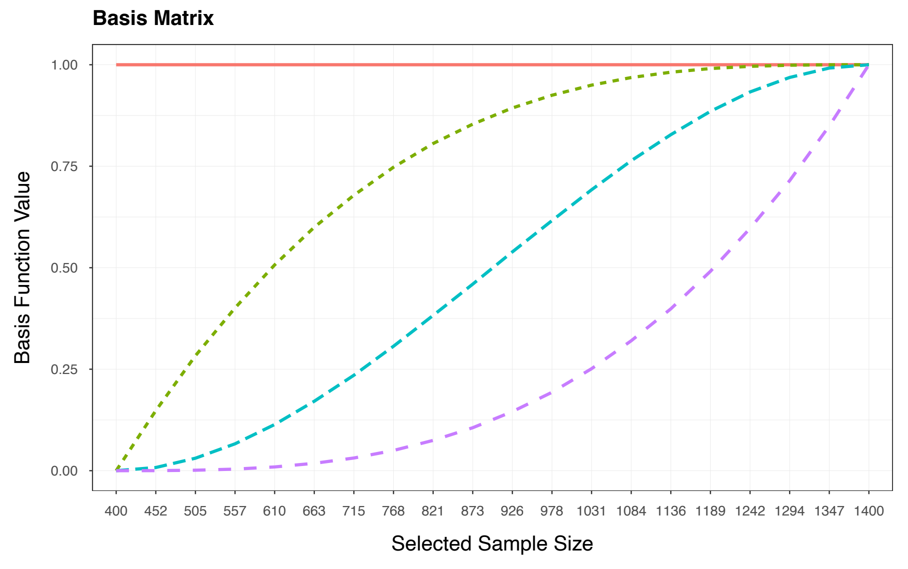


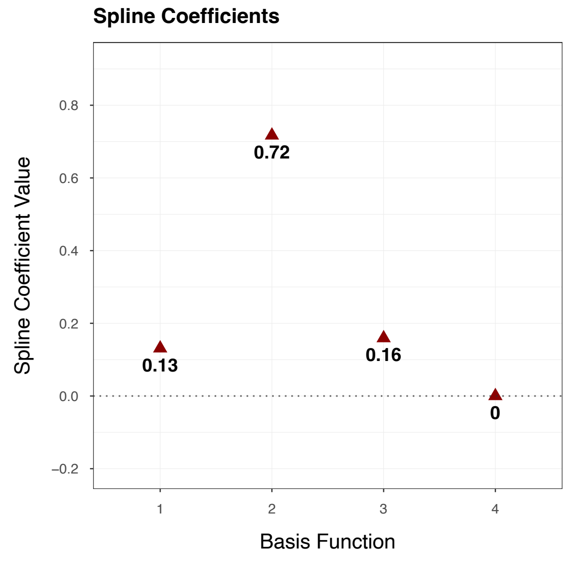


The goal of this step is to obtain a smooth (power) function and interpolate the statistic for all sample sizes in the range $\mathbb{N}_s$.

Step 2

- assume monotonicity and use cubic *I-Spline* bases with inner knots selected based on cross-validation

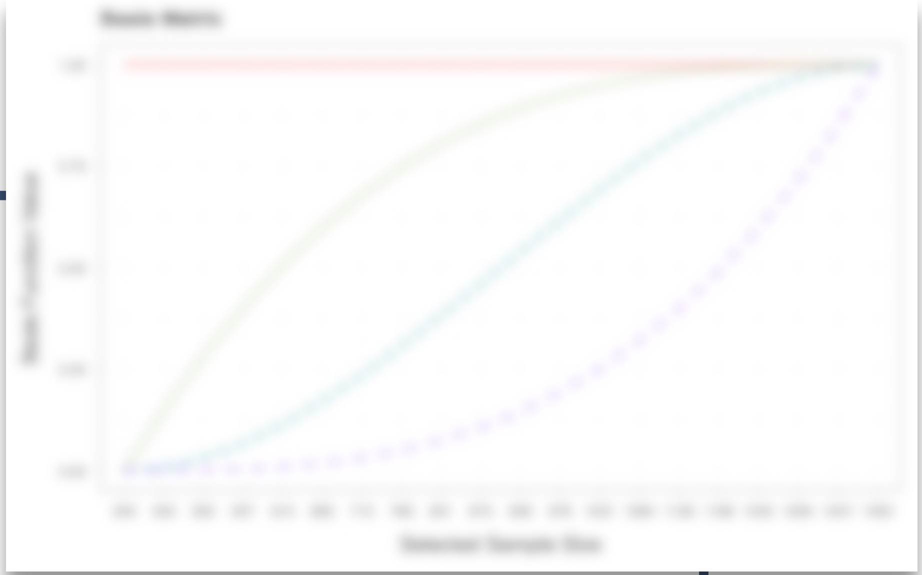


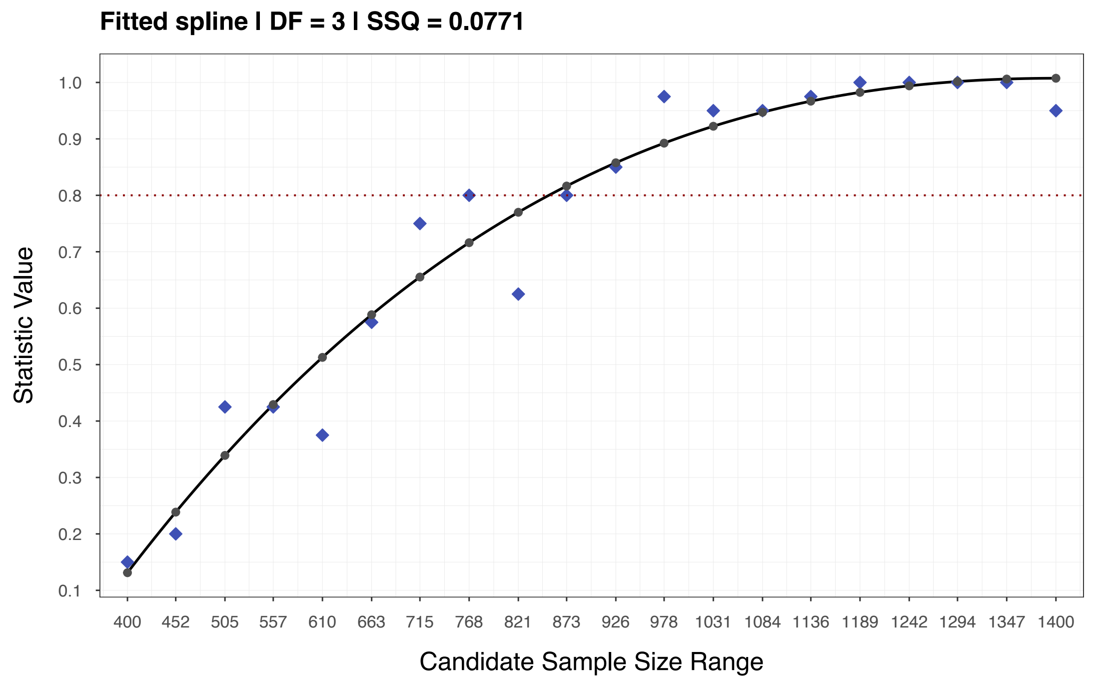


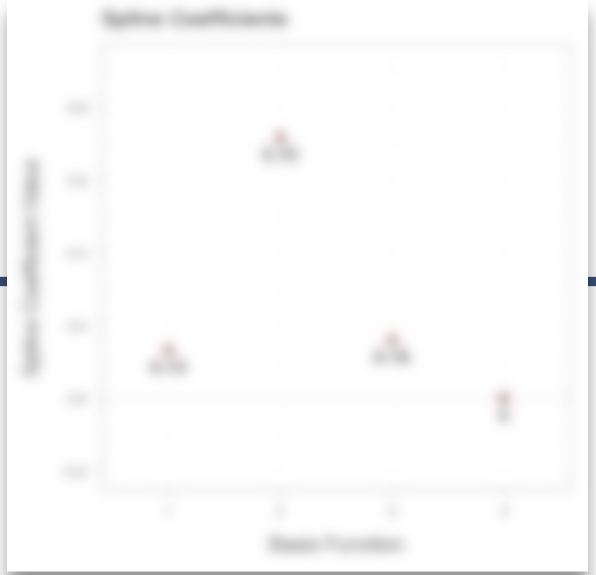
Step 2

The Method

- assume monotonicity and use cubic *I-Spline* bases with inner knots selected based on cross-validation







The goal of this step is to account for the MC error and provide a measure of uncertainty around the interpolated spline.

Step 3

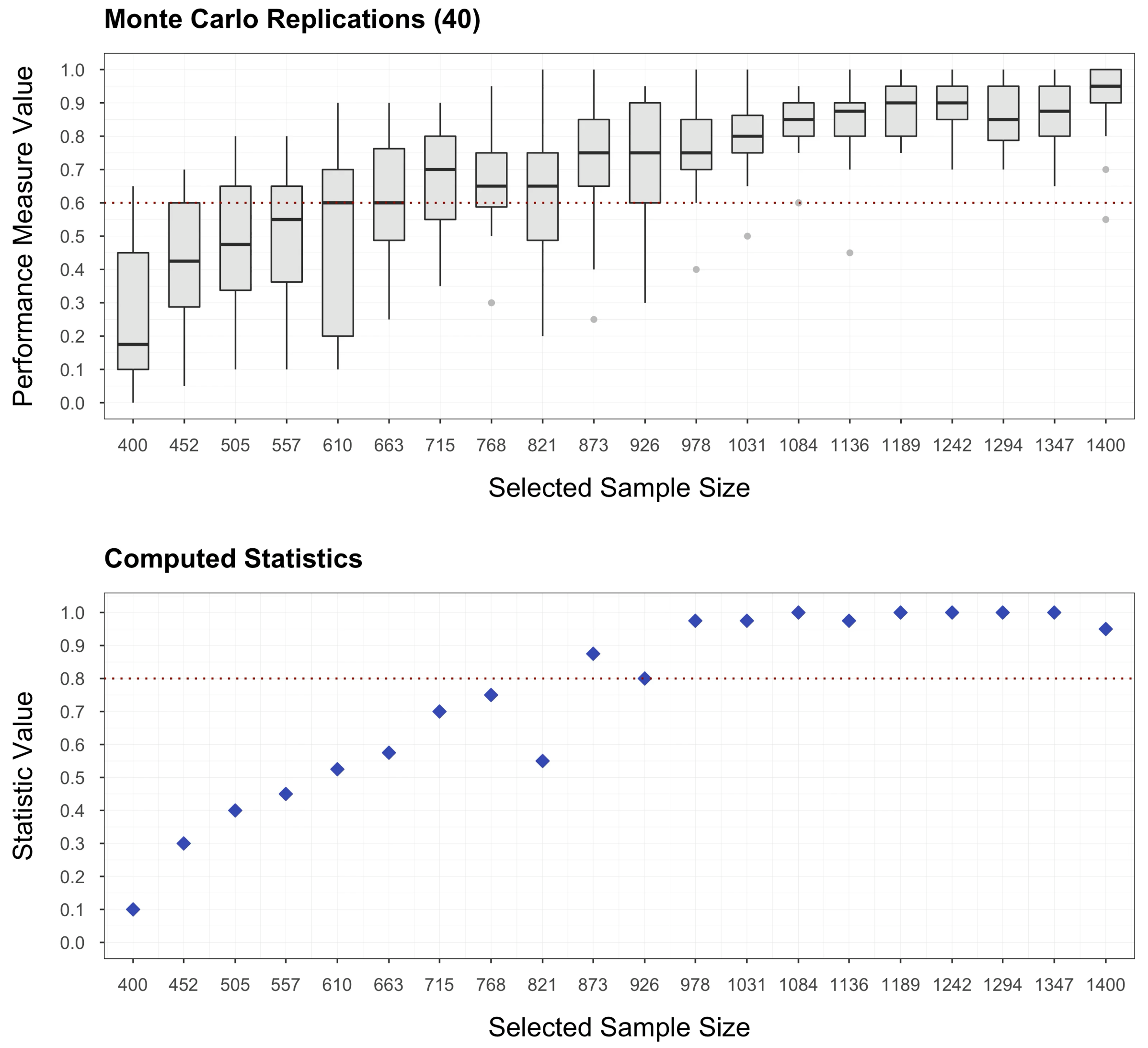
The Method

- use stratified bootstrapping to represent the variability in the replicated performance measures for each sample size $s_t \in S$

Step 3

The Method

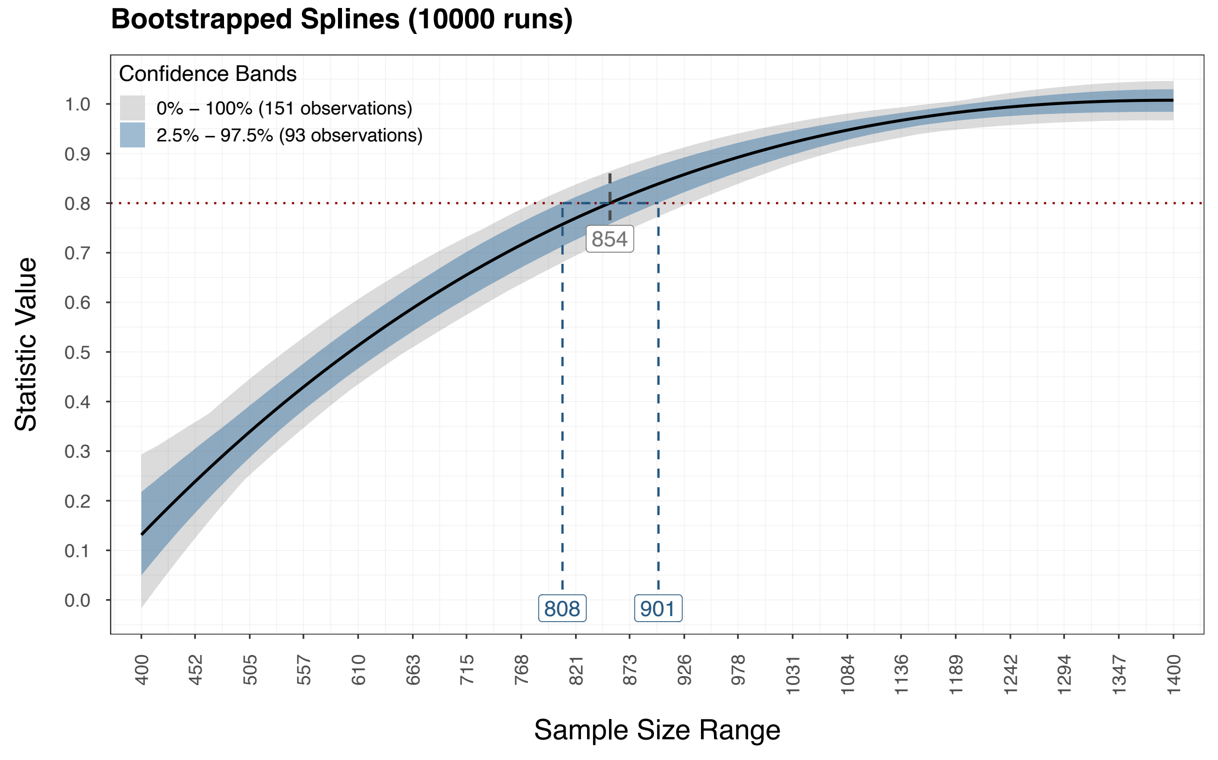
- use stratified bootstrapping to represent the variability in the replicated performance measures for each sample size $s_t \in S$
- we bootstrap the performance measures and, thus, re-estimating the model is not necessary

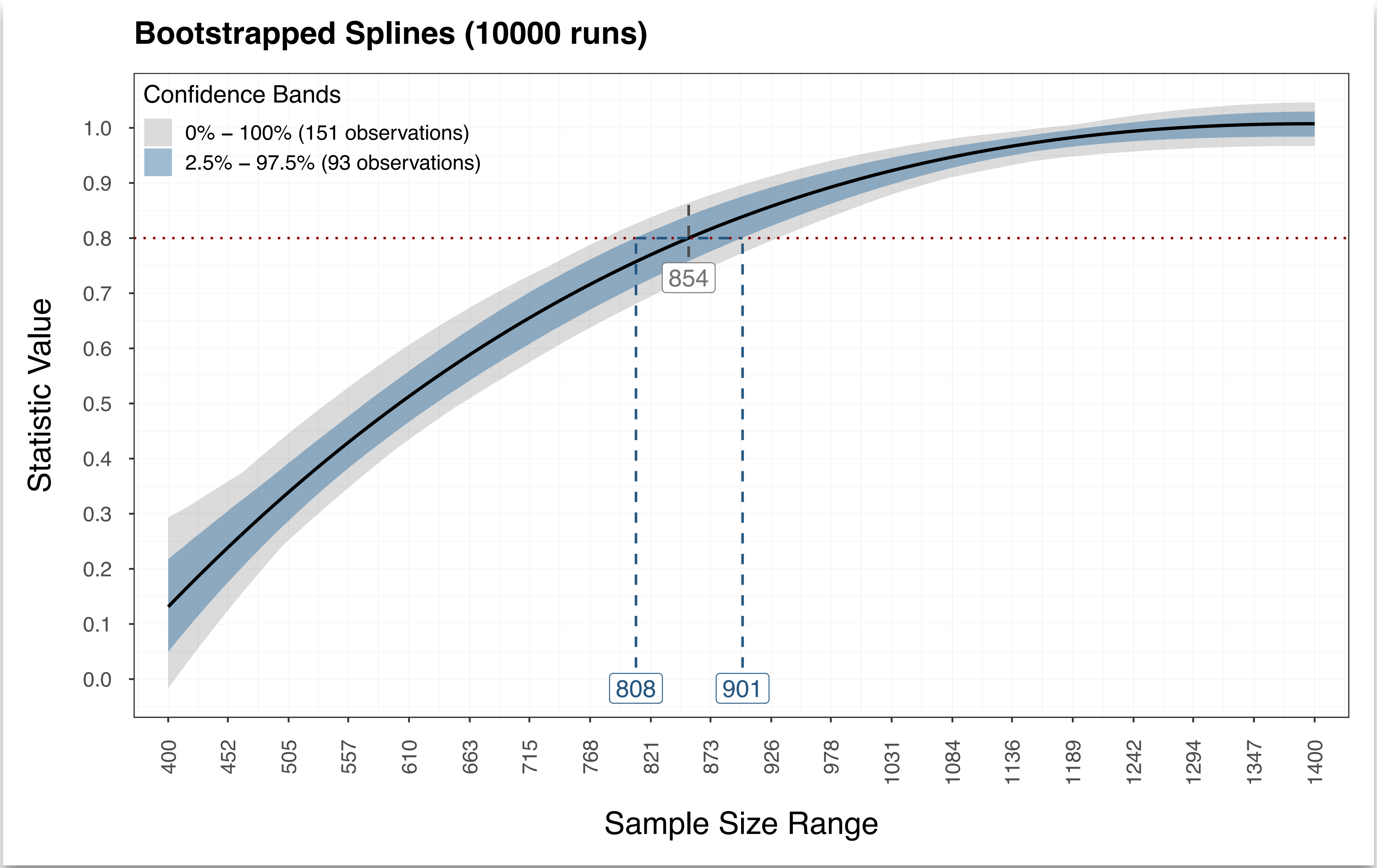


Step 3

The Method

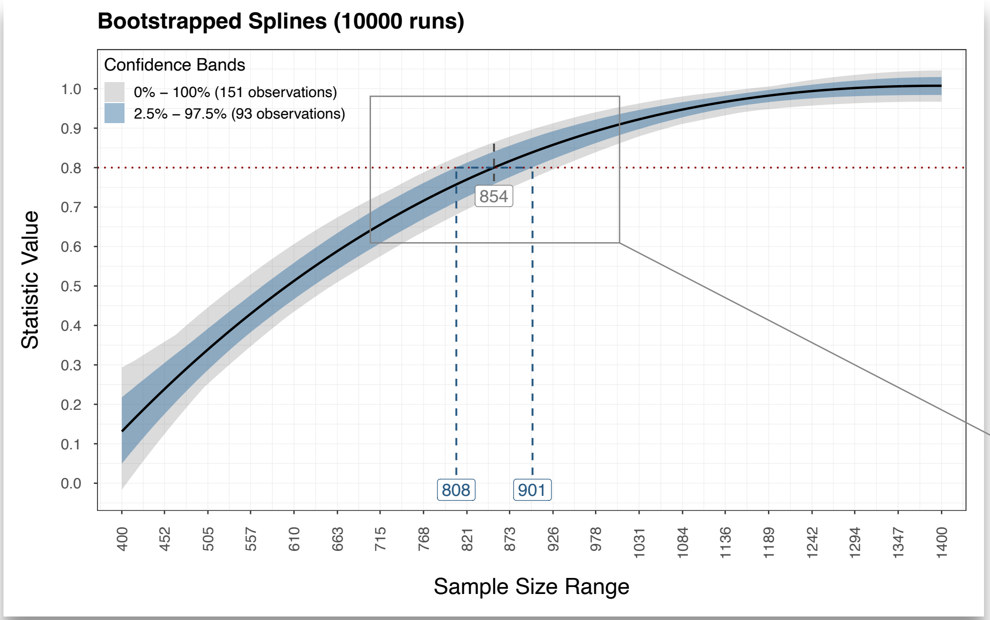
- use stratified bootstrapping to represent the variability in the replicated performance measures for each sample size $s_t \in S$
- we bootstrap the performance measures and, thus, re-estimating the model is not necessary
- fit a new spline to each bootstrapped matrix of performance measures

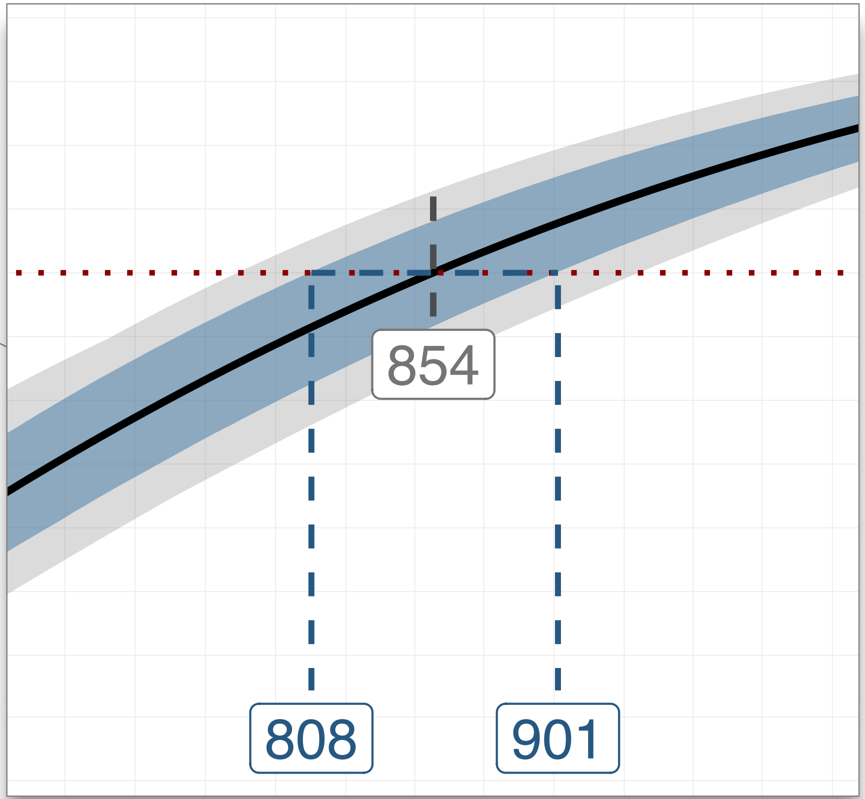



Bootstrapped Splines (10000 runs)
Confidence Bands
0% – 100% (151 observations)
2.5% – 97.5% (93 observations)
Statistic Value
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
854
808
901
400
452
505
557
610
663
715
768
821
873
926
978
1031
1084
1136
1189
1242
1294
1347
1400
Sample Size Range

Step 3

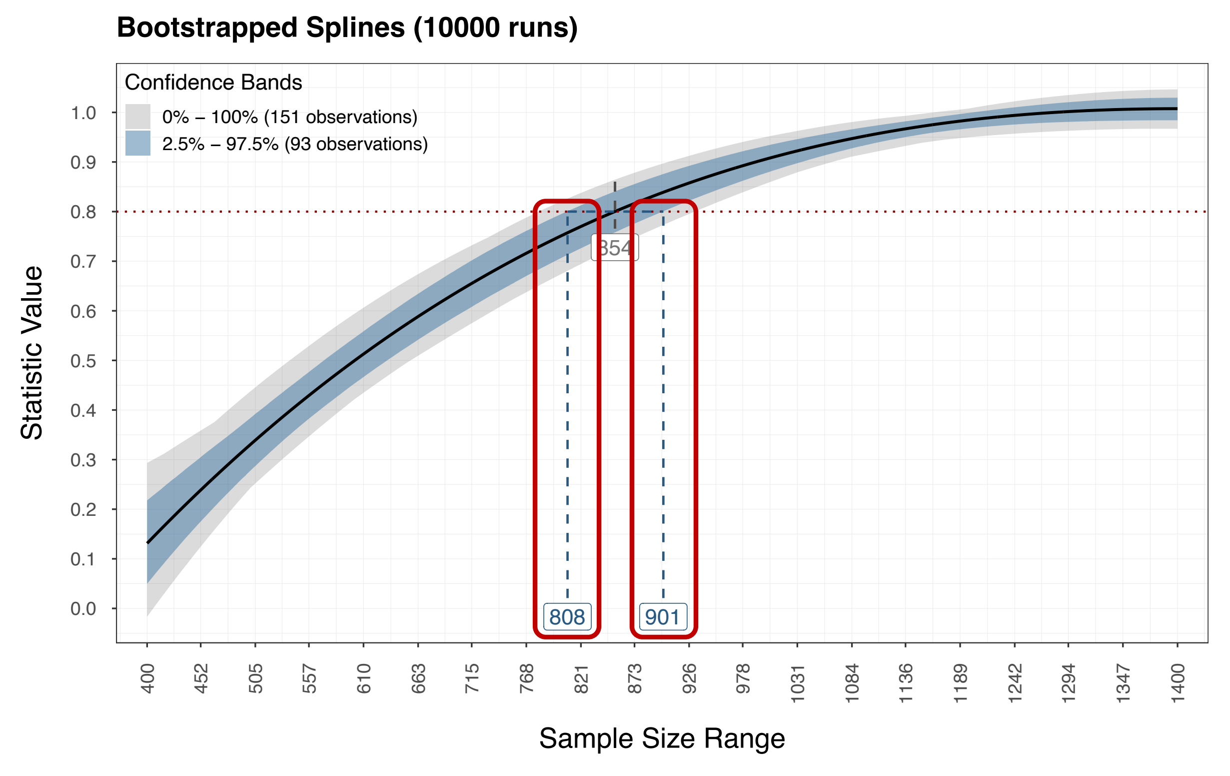
The Method





Convergence

The Method



- update candidate range $\mathbb{N}_s$ based on the confidence bands
- repeat Steps 1 to 3 until range $\mathbb{N}_s$ becomes small enough

The Implementation

- an **R** package

powerly

powerly.dev

In Code

The Implementation

```r
# Load the library.
library(powerly)
```

In Code

The Implementation

```
# Load the library.
library(powerly)
```

```
# Generate a true model.
true_model <- generate_model(
    type = "...",
    ...
)
```

In Code

The Implementation

```r
# Load the library.
library(powerly)
```

```r
# Generate a true model.
true_model <- generate_model(
    type = "...",
    ...
)
```

```r
# Load the library.
library(powerly)
```

```r
# Generate a true model.
true_model <- generate_model(
    type = "...",
    ...
)
```

```r
# Run the method.
results <- powerly(
    range_lower = 300,
    range_upper = 1000,
    samples = 30,
    replications = 20,
    measure = ”...",
    statistic = "power",
    measure_value = .6,
    statistic_value = .8,
    model = "...",
    model_matrix = true_model
)
```

In Code

The Implementation

```r
# Load the library.
library(powerly)
```

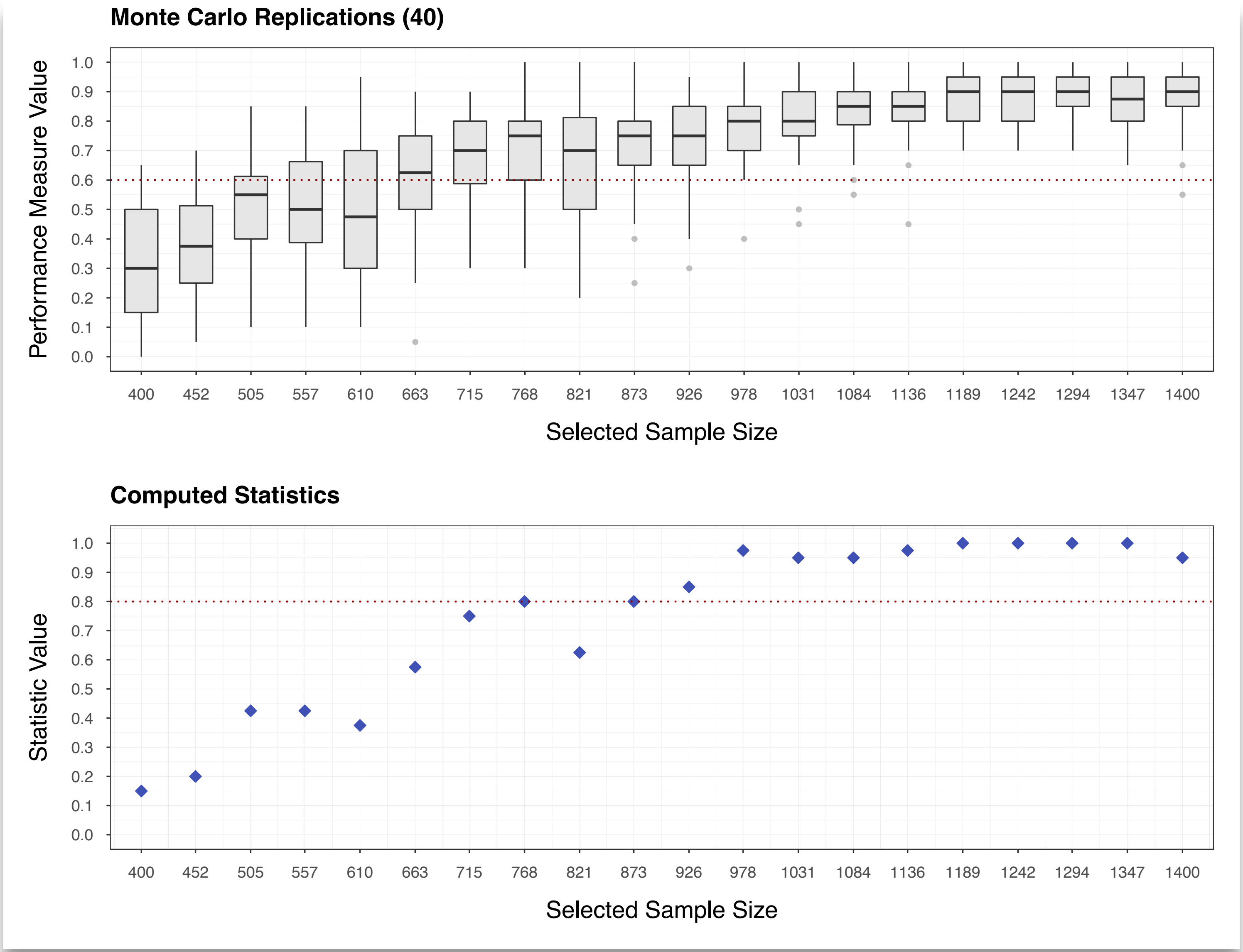
```r
# Generate a true model.
true_model <- generate_model(
    type = "...",
    ...
)
```

```r
# Run the method.
results <- powerly(
    range_lower = 300,
    range_upper = 1000,
    samples = 30,
    replications = 20,
    measure = "...",
    statistic = "power",
    measure_value = .6,
    statistic_value = .8,
    model = "...",
    model_matrix = true_model
)
```

Step 1

The Implementation

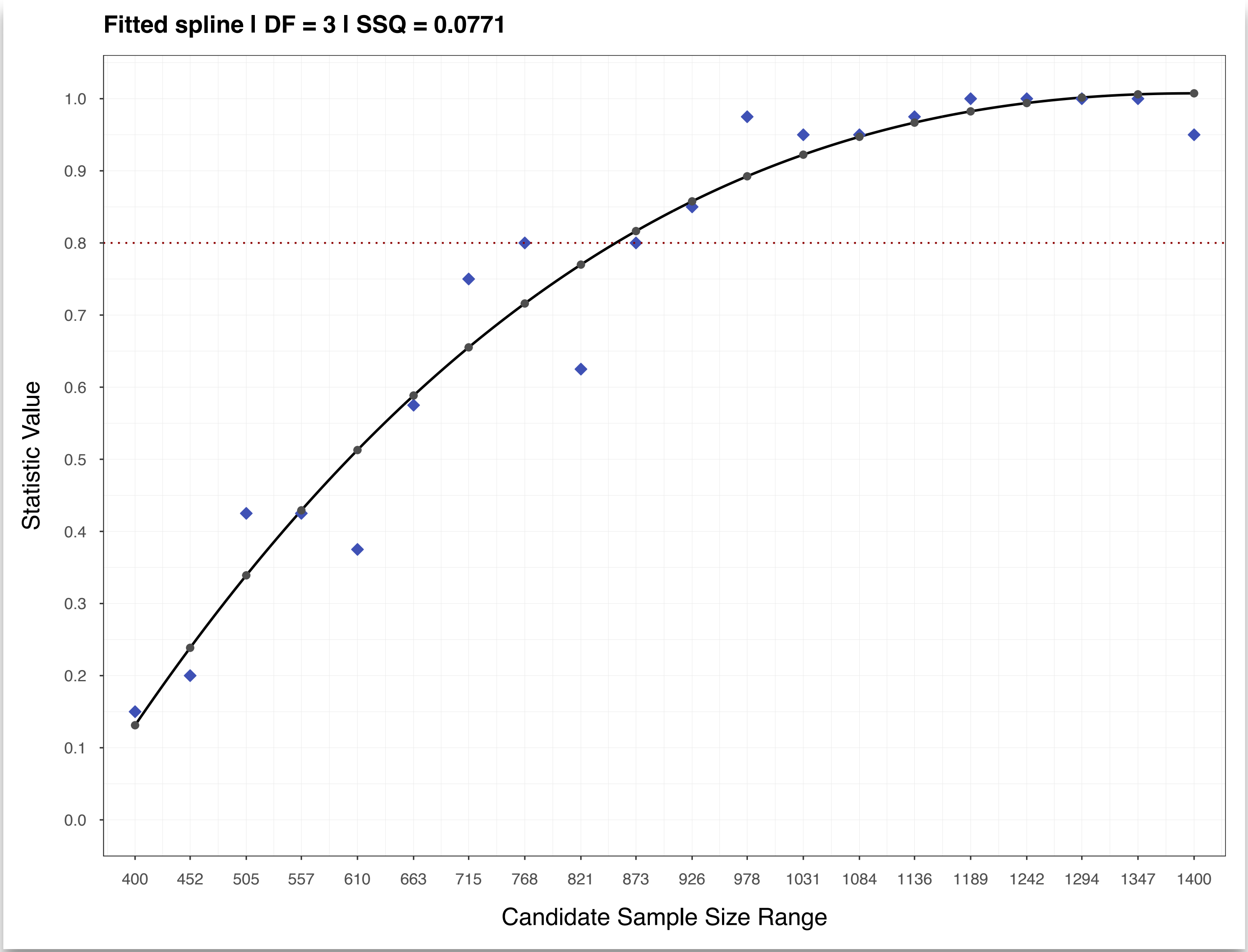
```
plot(results, step = 1)
```



Step 2

The Implementation

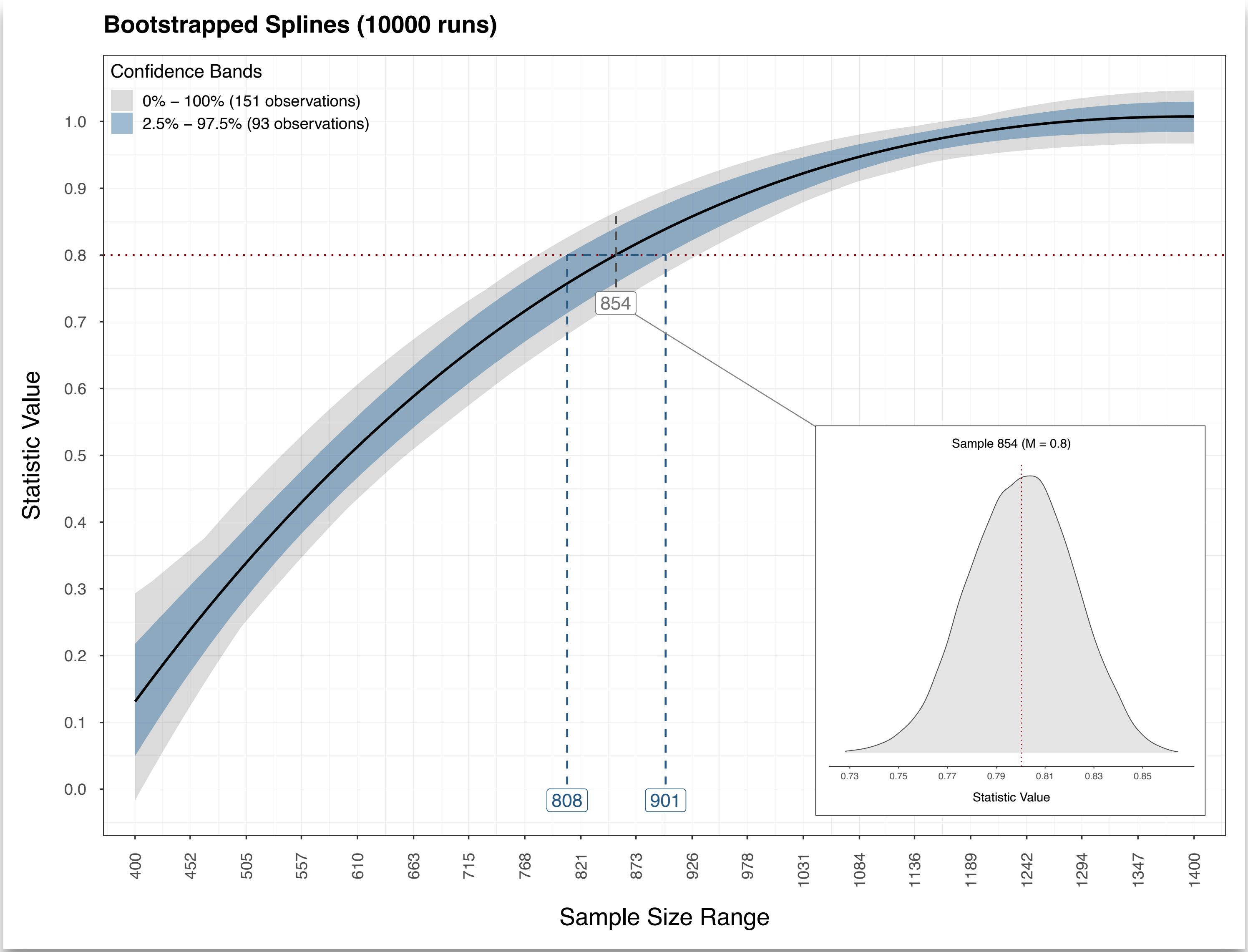
```
plot(results, step = 2)
```



Step 3

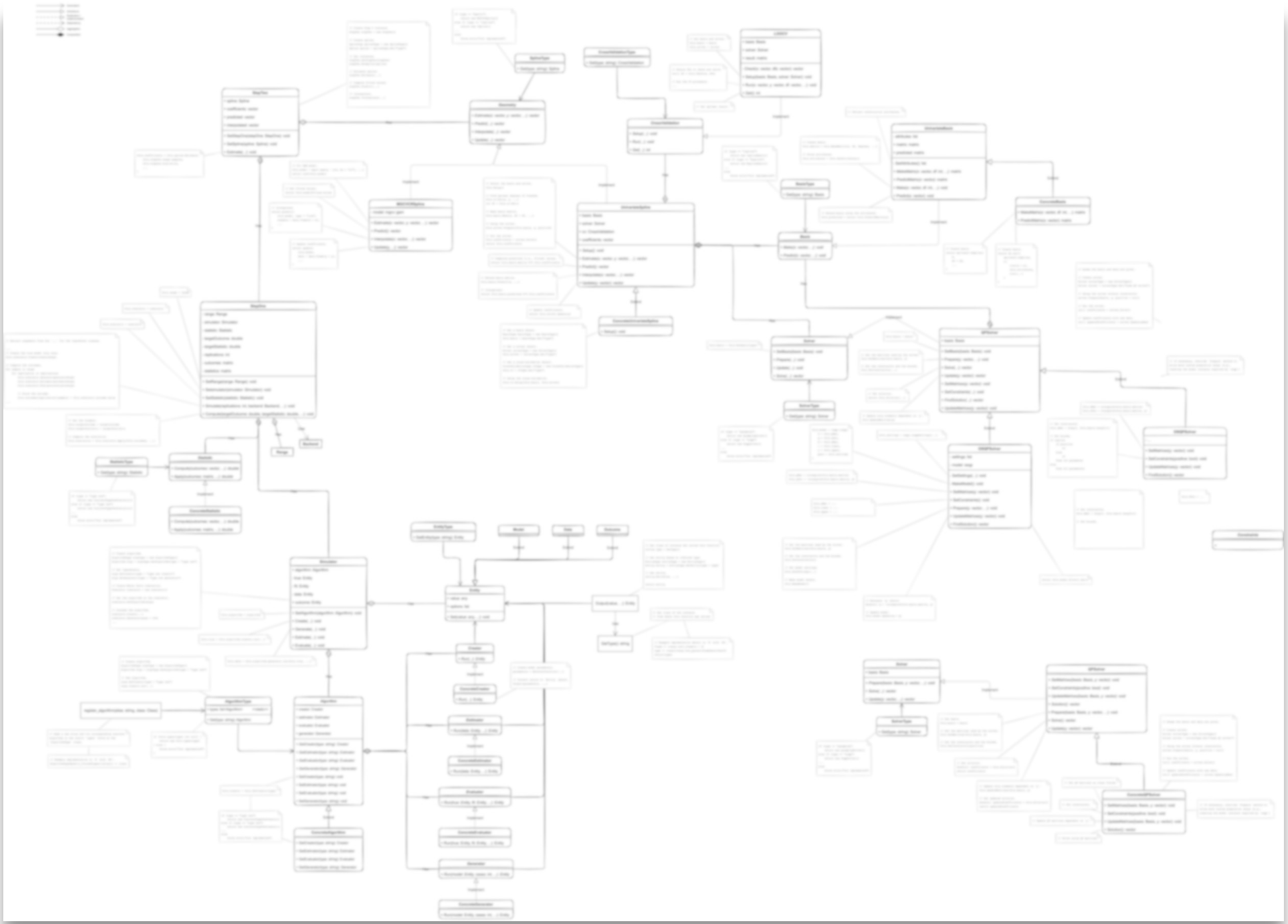
The Implementation

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



Positive Lookahead

A General Framework



by Marie Mainguy

Why?

- sample sizes tailored to specific research questions
- sample size analysis as an ecosystem
  - growing collection of models and performance measures
  - developer API for enabling sample size computations
- upcoming tutorial paper where we
  - discuss these ideas
  - and show how to apply them

Our Final Frontier

we aim to make sample size analysis so accessible that there is no way around not doing it

by Marie Mainguy

Workshop Resources

samplesize.help