

Power and Sample Size by Simulation

Alex Asher

Senior Biostatistician and Software Developer
StataCorp LLC

2025 UK Stata Conference

Outline

- Power and Sample Size Concepts
 - Null and alternative hypotheses
 - Critical values, alpha levels, power, effect size
- Power and Sample Size Using Stata
 - Built-in `power` methods
 - Power by simulation
 - Sample size by simulation
- Group Sequential Designs
 - Introduction to GSDs
 - Simulation-based GSDs

Research Question



image: Journal of Pharmacology & Clinical Research, licensed under CC BY 4.0

- Does our novel chemotherapy shrink tumors?
- Outcome: Tumor Shrinkage Rate (TSR)

$$\text{TSR} = (D_b - D_a) / (D_b \times t) \times 100\%$$

D_b is the longest diameter of the tumor before treatment

D_a is the longest diameter after treatment

t is the time elapsed in days

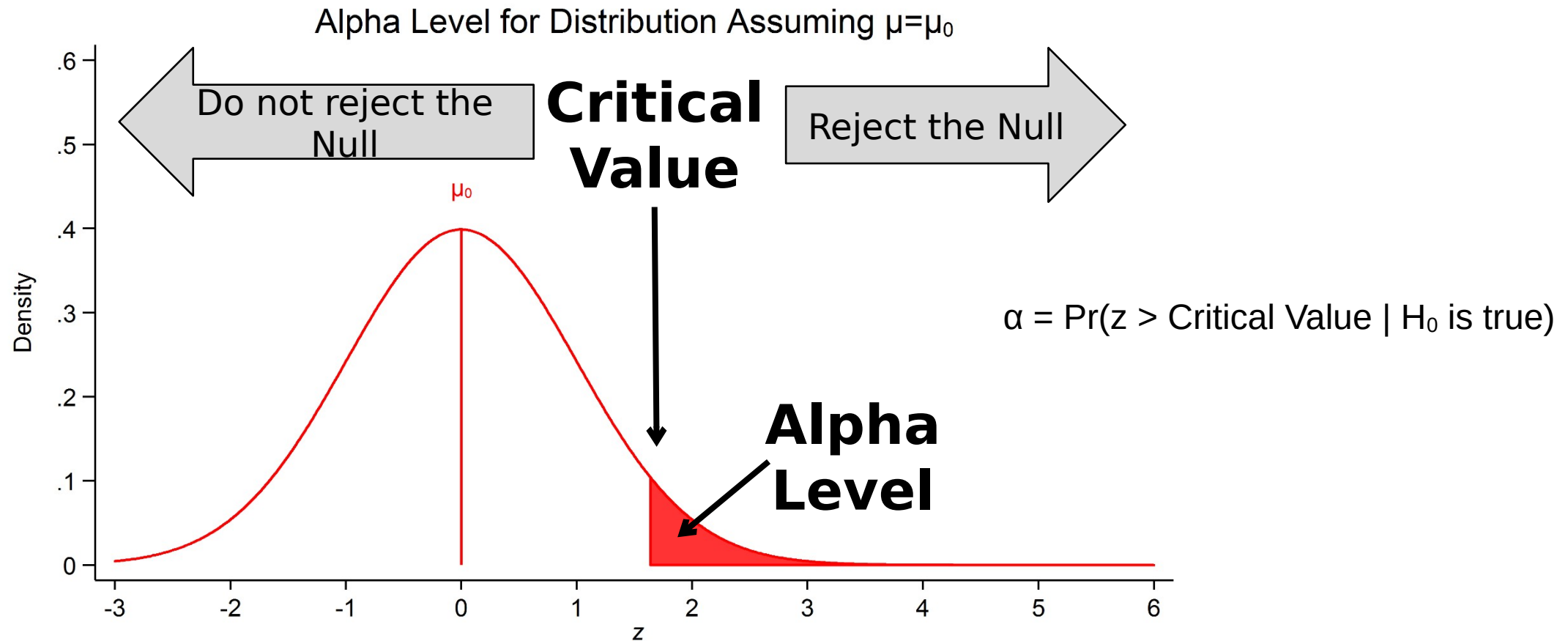
Null and alternative hypotheses

- $H_0: \text{TSR} = 0, \quad H_a: \text{TSR} > 0$
- Conduct a z-test with known SD

- Test statistic
$$Z = \frac{\bar{X} - \mu_0}{\sigma / \sqrt{n}}$$

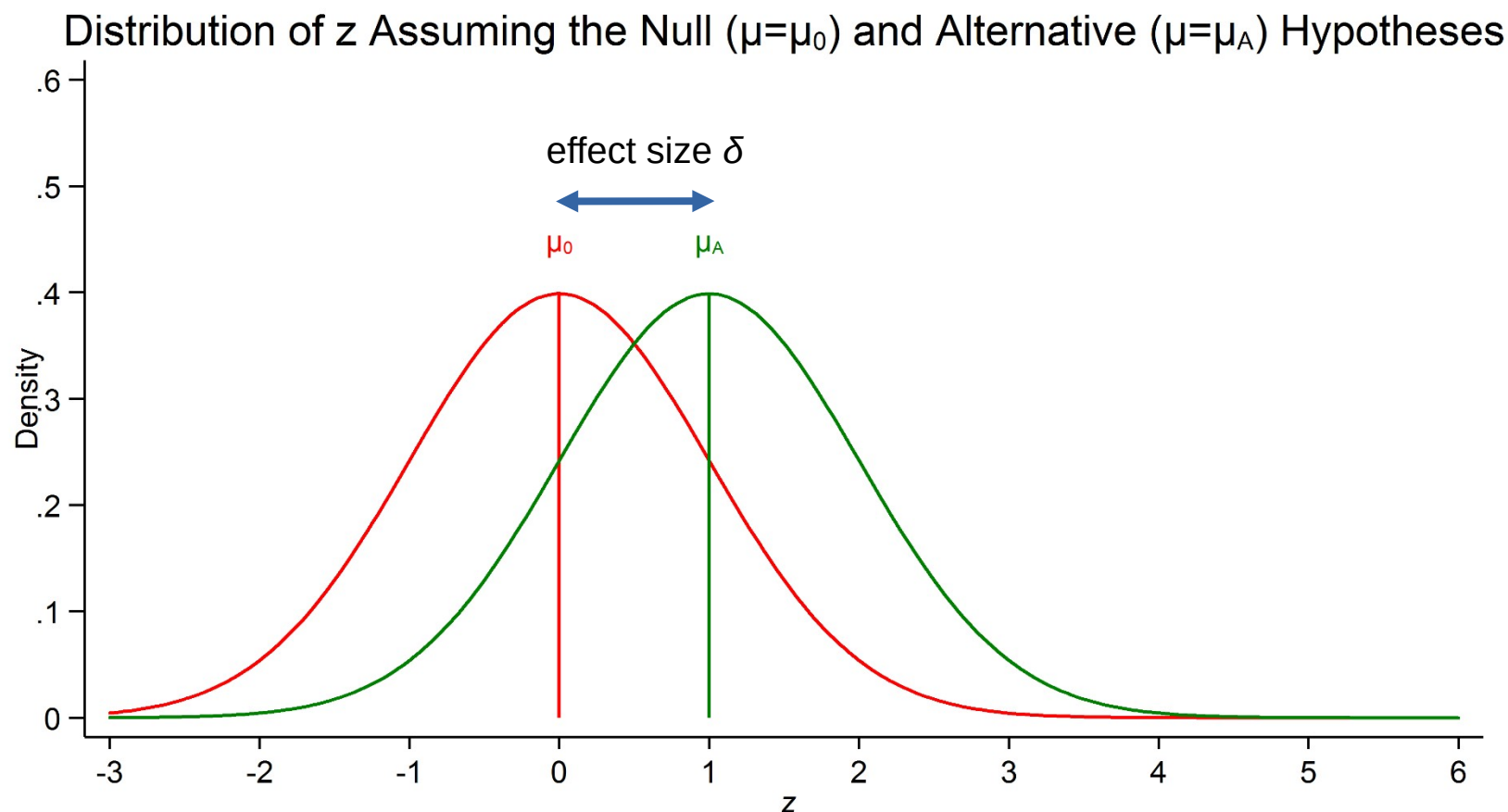
- x_i = observed TSR for participant i
 μ_0 = TSR under H_0
 σ = standard deviation (known)
 n = sample size

Critical Value and Alpha Level



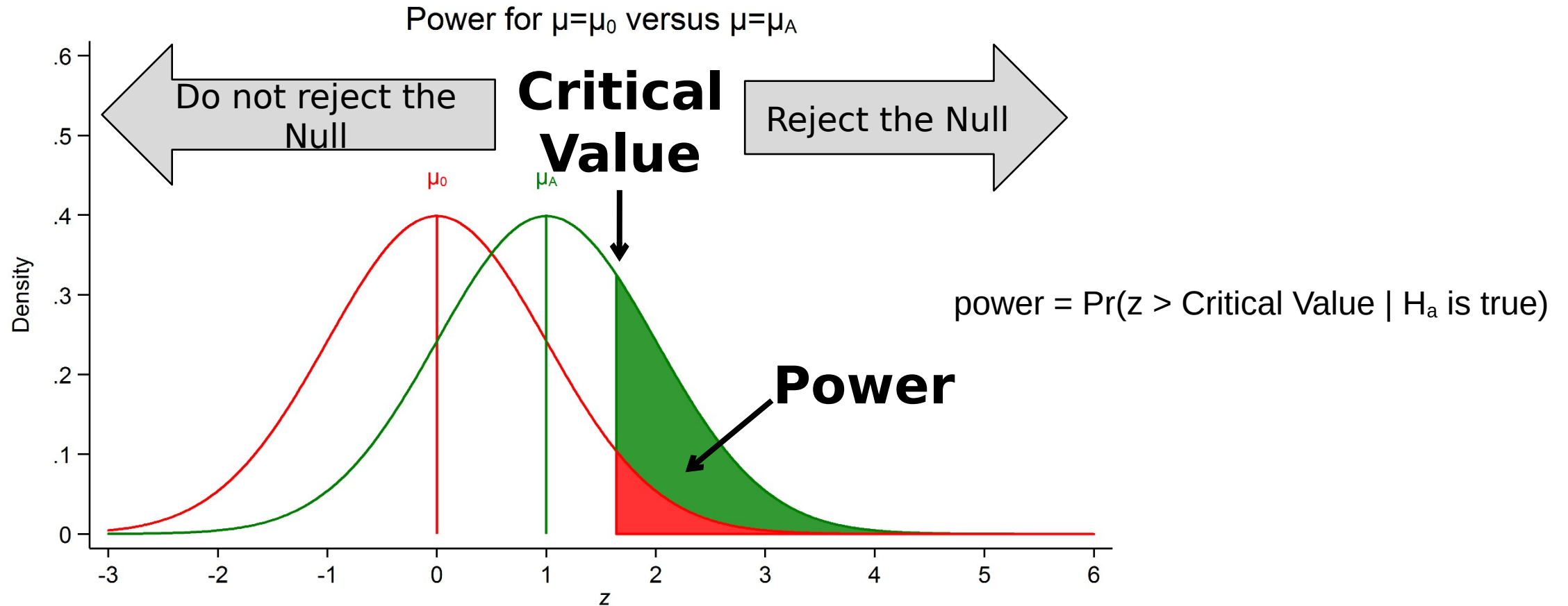
Graph courtesy of Chuck Huber, StataCorp

Distribution Under the Alternative



Graph courtesy of Chuck Huber, StataCorp

Critical Value and Power



Alpha Level and Power

- Alpha Level:
 - The probability of rejecting the null hypothesis assuming that the null hypothesis is true.
- Statistical Power:
 - The probability of rejecting the null hypothesis assuming that the alternative hypothesis is true.

Solve for Power

- User must specify:
 - Alpha
 - One-sided or two-sided
 - Often 0.05 or 0.01 for two-sided tests
 - We will conduct an upper one-sided test at the 0.025 level
 - Sample size
 - Often dictated by funding
 - We will use $n=50$
 - Effect size
 - Difference between hypothesized parameter values
 - Can be scaled or unscaled (specify SD if unscaled)
 - We will use $\delta=0.5$ and $\sigma=1$

Power calculation in Stata

The screenshot displays the StataNow/MP 18.5 interface. On the left, the 'Statistics' menu is open, showing a list of statistical methods. A blue arrow points from the 'Power, precision, and sample size' option in this menu to a larger window on the right. This window, titled 'Power, precision, and sample-size analysis', shows a list of methods organized by category. A second blue arrow points from the 'Population parameter' category to the 'Test comparing one mean to a reference value' method.

StataNow/MP 18.5

File Edit Data Graphics Statistics User Window Help

Summaries, tables, and tests

Linear models and related

Binary outcomes

Ordinal outcomes

Categorical outcomes

Count outcomes

Fractional outcomes

Generalized linear models

Choice models

Time series

Multivariate time series

Spatial autoregressive models

Longitudinal/panel data

Multilevel mixed-effects models

Survival analysis

Epidemiology and related

Endogenous covariates

Sample-selection models

Causal inference/treatment effects

SEM (structural equation modeling)

LCA (latent class analysis)

FMM (finite mixture models)

IRT (item response theory)

Multivariate analysis

Survey data analysis

Lasso

Meta-analysis

Multiple imputation

Nonparametric analysis

Exact statistics

Resampling

Power, precision, and sample size

Bayesian analysis

Bayesian model averaging

History

Filter commands here

Command

There are no items to

Power, precision, and sample-size analysis

Methods organized by:

Sort Filter methods here

All methods

Population parameter

- Correlations
- Hazard rates
- Means
- Odds ratio
- Proportions
- R-squared
- Regression slopes
- Standard deviations
- Survival rates
- Variances
- Outcome
- Hypothesis test
- Confidence interval
- Sample
- Group sequential design

- Fisher's z test comparing one correlation to a reference value
- Fisher's z test comparing two independent correlations
- Test comparing partial correlation to a reference value in a multiple linear regression
- Exponential test comparing two independent hazard rates
- One-way analysis of variance
- Two-way analysis of variance
- Repeated-measures analysis of variance
- Test comparing one mean to a reference value
- Test comparing one mean to a reference value in a cluster randomized design
- CI for one mean
- Test comparing one mean to a reference value in a group sequential design
- Test comparing two independent means
- Test comparing two independent means in a cluster randomized design
- CI for a two-means difference
- Test comparing two independent means in a group sequential design
- Paired test comparing two correlated means, specify correlation between paired observations
- Paired test comparing two correlated means, specify standard deviation of the differences
- CI for a paired-means difference, specify correlation between paired observations
- CI for a paired-means difference, specify the standard deviation of the difference
- Test of association in 1:M matched case-control studies

Power calculation in Stata

power onemean - Power analysis for a one-sample mean test

Main Table Graph Iteration

Compute: Power * Accepts numlist [\(Examples\)](#)

Error probabilities: 0.025 * Significance level

Sample size: 50 * Sample size

Effect size

Means: 0 * Null, 0.5 * Alternative

Standard deviation: 1 * Standard deviation, ☒ Assume a known standard deviation

* Finite population correction: None

Sides: One-sided test

☐ Treat number lists in starred(*) options as parallel

? ↺ 📄 Submit Cancel OK

```
. power onemean 0 0.5, n(50) alpha(0.025) knownsd onesided
```

Estimated power for a one-sample mean test
z test

$H_0: m = m_0$ versus $H_a: m > m_0$

Study parameters:

alpha = 0.0250
N = 50
delta = 0.5000
m0 = 0.0000
ma = 0.5000
sd = 1.0000

Estimated power:

power = 0.9424

Power calculation by simulation

```
**# program onmeansim: one sample z-test (upper one-sided test)
program onmeansim, rclass
    version 19
    syntax, n(integer)          /// Sample size
        [ alpha(real 0.025)    /// Alpha level
          m0(real 0)            /// Mean under the null
          ma(real 0.5)          /// Mean under the alternative
        ]
    // Generate a normal random variable with mean = `ma' and SD = 1
    quietly drawnorm y, n(`n') mean(`ma') sd(1) clear
    // Upper one-sided z-test
    ztest y == `m0'
    return scalar reject = (r(p_u) < `alpha')
end
```


Power calculation by simulation

```
. // Try it out  
. set seed 9  
  
. onmeansim, n(50)
```

One-sample z test

variable	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
y	50	.6792907	.1414214	1	.4021099	.9564715

```
mean = mean(y)                                z = 4.8033  
H0: mean = 0
```

Ha: mean < 0
Pr(Z < z) = 1.0000

Ha: mean != 0
Pr(|Z| > |z|) = 0.0000

Ha: mean > 0
Pr(Z > z) = 0.0000

Create a power *usermethod* command

```
*** program power_cmd_onmeansim: power by simulation
program power_cmd_onmeansim, rclass
    version 19
    syntax, n(integer)          /// Sample size
        [ alpha(real 0.025)    /// Alpha level
          m0(real 0)            /// Mean under the null
          ma(real 0.5)          /// Mean under the alternative
          reps(integer 100)     /// Number of replicates
          seed(numlist max=1)   /// Random seed (no default seed)
        ]

    // Set seed, if specified
    if ("`seed'" != "") set seed `seed'

    // Simulate data and test the null hypothesis
    quietly simulate reject=r(reject), reps(`reps'):    ///
        onmeansim, n(`n') m0(`m0') ma(`ma') alpha(`alpha')
    quietly summarize reject

    // Return results
    return scalar power = r(mean)
    return scalar N      = `n'
    return scalar alpha  = `alpha'
    return scalar m0     = `m0'
    return scalar ma     = `ma'
    return scalar delta  = `ma' - `m0'

end
```


Run `power onmeansim`

```
. // Start with N=50. Use 1000 reps for accuracy. Set seed for reproducibility.  
. power onmeansim, n(50) alpha(0.025) m0(0) ma(0.5) reps(1000) seed(9)
```

Estimated power

Two-sided test



alpha	power	N
.025	.938	50

MC Error

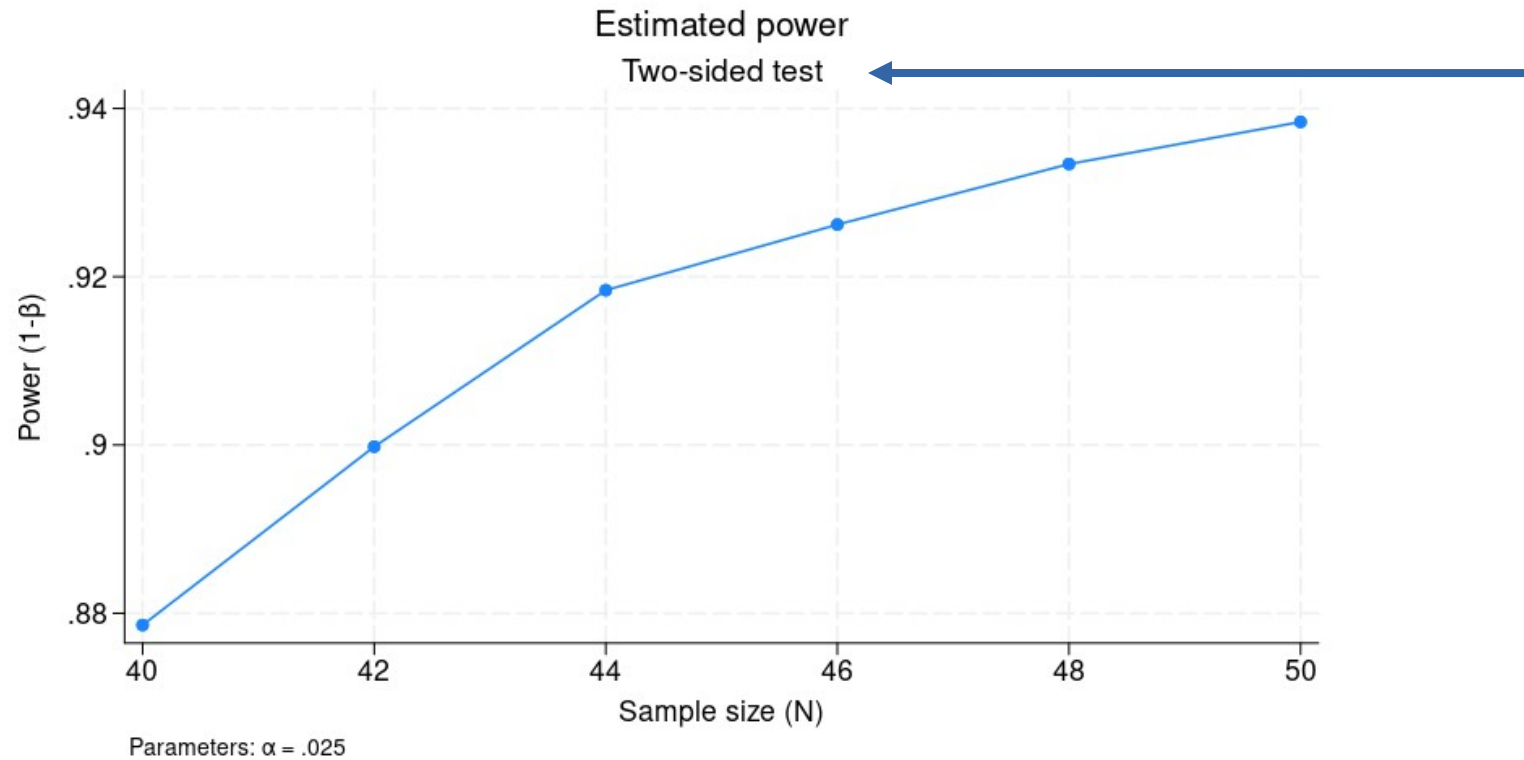
$$SE(\hat{power}) = \sqrt{\frac{power * (1 - power)}{reps}}$$

```
. // Calculate Monte Carlo error (standard error of power estimate)
. local SEpower = sqrt( r(power) * (1-r(power)) / 1000 )

. display "SE of power estimate: `SEpower'"
SE of power estimate: .007626008130077
```


Sample size calculation by simulation

```
. // Option 1: provide a numlist of sample sizes  
. set seed 9 // Bonus question: Why don't we use option seed()?   
  
. power onmeansim, n(40(2)50) alpha(0.025) m0(0) ma(0.5) reps(5000) graph
```



Customize output with an initializer

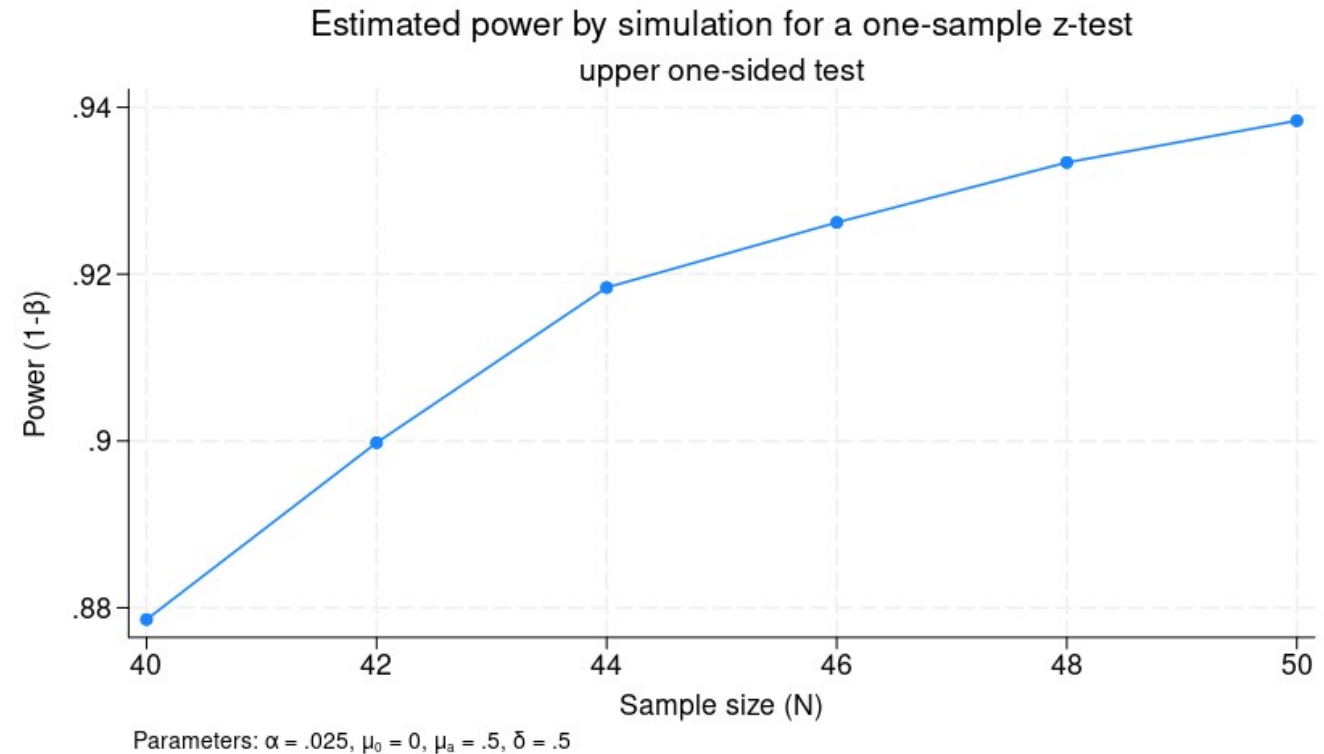
```
**# program power_cmd_onemeansim_init: use initializer to customize output
program power_cmd_onemeansim_init, sclass
    version 19
    sreturn clear
    sreturn local pss_numopts "m0 ma" // Options allowing numlists
    sreturn local pss_colnames "m0 ma delta" // Columns to add to default table
    sreturn local pss_colgrsymbols " $\mu_{\text{sub:0}}$   $\mu_{\text{sub:a}}$   $\delta$ " // Graph symbols
    sreturn local pss_title "by simulation for a one-sample z-test"
    sreturn local pss_subtitle "upper one-sided test"
end
```


Customize output with an initializer

```
. // Re-run with initializer  
. set seed 9  
  
. power onmeansim, n(40(2)50) alpha(0.025) m0(0) ma(0.5) reps(5000) graph table
```

Estimated power by simulation for a one-sample z-test
upper one-sided test

alpha	power	N	m0	ma	delta
.025	.8786	40	0	.5	.5
.025	.8998	42	0	.5	.5
.025	.9184	44	0	.5	.5
.025	.9262	46	0	.5	.5
.025	.9334	48	0	.5	.5
.025	.9384	50	0	.5	.5



Use `power` to calculate sample size by simulation

```
***# program power_cmd_onemeansampsize: simulation-based sample-size evaluator
program power_cmd_onemeansampsize, rclass
    version 19
    syntax, power(numlist >0 <1 max=1) /// Requested power
        [ alpha(real 0.05)          /// Alpha level
          m0(real 0)                  /// Mean under the null
          ma(real 0.5)                /// Mean under the alternative
          reps(integer 100)           /// Number of repetitions for power calculation
          seed(numlist max=1)         /// Random seed (no default seed)
          startval(integer 100)       /// Starting value for sample size
          maxval(integer 1000)        /// Maximum value for sample size
          minval(integer 1)           /// Minimum value for sample size
          powtol(real 0.01)           /// Tolerance for power search (default 1%)
          biter(integer 100)          /// Max iterations for bisection algorithm
          nfractional                /// Unused, but required by -gsdesign-
          onesided                    /// Unused, but required by -gsdesign-
        ]

    // Set seed, if specified
    if("`seed'" != "") set seed `seed'

    // Test powtol() to ensure it's not set too low
    assert (`powtol' > 0)
```


Use `power` to calculate sample size by simulation (continued)

```
// Test maxval() to ensure it's not set too low
power_cmd_onemeansim, n(`maxval') alpha(`alpha') m0(`m0') ma(`ma') reps(`reps')
if (r(power) < `power') {
    di as error "error: {bf:maxval()} too small"
    exit 480
}
// Test -minval()- to ensure it's not set too high
power_cmd_onemeansim, n(`minval') alpha(`alpha') m0(`m0') ma(`ma') reps(`reps')
if (r(power) > `power') {
    di as error "error: {bf:minval()} too large"
    exit 480
}

// Compare requested power to power attained via simulation at the starting value
local thisN = `startval'
local lastN = .
power_cmd_onemeansim, n(`thisN') alpha(`alpha') m0(`m0') ma(`ma') reps(`reps')
local pdiff = r(power) - `power'
```


Use `power` to calculate sample size by simulation (continued)

```
// Repeat guess-n-check process using bisection algorithm
local i = 1
while (abs(`pdiff`) > `powtol`) {
  local ++i
  if (`i` > `biter`) {
    di as error "convergence not achieved"
    exit 430
  }

  // Bisection algorithm
  local lastN = `thisN`
  if (`pdiff` > 0) { // Overpowered: decrease N
    local maxval = `thisN`
  }
  else { // Underpowered: increase N
    local minval = `thisN`
  }
  local thisN = ceil((`maxval` + `minval`) / 2)
  if (`thisN` == `lastN`) { // Already converged
    continue, break
  }
  power_cmd_onemeansim, n(`thisN`) alpha(`alpha`) m0(`m0`) ma(`ma`) reps(`reps`)
  local pdiff = r(power) - `power`
}
```


Use `power` to calculate sample size by simulation (continued)

```
// Return results
return scalar power      = r(power)
return scalar N          = `thisN'
return scalar alpha      = `alpha'
return scalar m0         = `m0'
return scalar ma         = `ma'
return scalar delta      = `ma' - `m0'
return local direction = "upper" ←
end
```


Customize output with an initializer

```
**# program power_cmd_onemeansampsize_init: use initializer to customize output
program power_cmd_onemeansampsize_init, sclass
    version 19
    sreturn clear
    sreturn local pss_numopts "m0 ma" // Options allowing numlists
    sreturn local pss_colnames "m0 ma delta" // Columns to add to default table
    sreturn local pss_colgrsymbols " $\mu_{\text{sub:0}}$   $\mu_{\text{sub:a}}$   $\delta$ " // Graph symbols
    sreturn local pss_title "by simulation for a one-sample z-test"
    sreturn local pss_subtitle "upper one-sided test"
end
```


Calculate sample size by simulation

```
. // Try it as a -power- command  
. power onemeansampsize, power(0.9) alpha(0.025) m0(0) ma(0.5) startval(100) reps(500) seed(9)
```

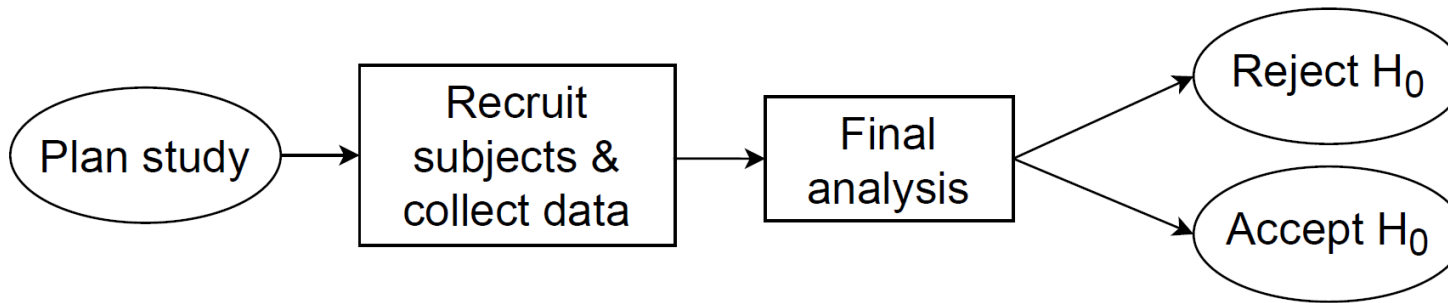
Estimated sample size by simulation for a one-sample z-test
upper one-sided test

alpha	power	N	m0	ma	delta
.025	.896	39	0	.5	.5

Fixed Sample Design vs Group Sequential Design

or: how I learned to stop worrying and accept the null

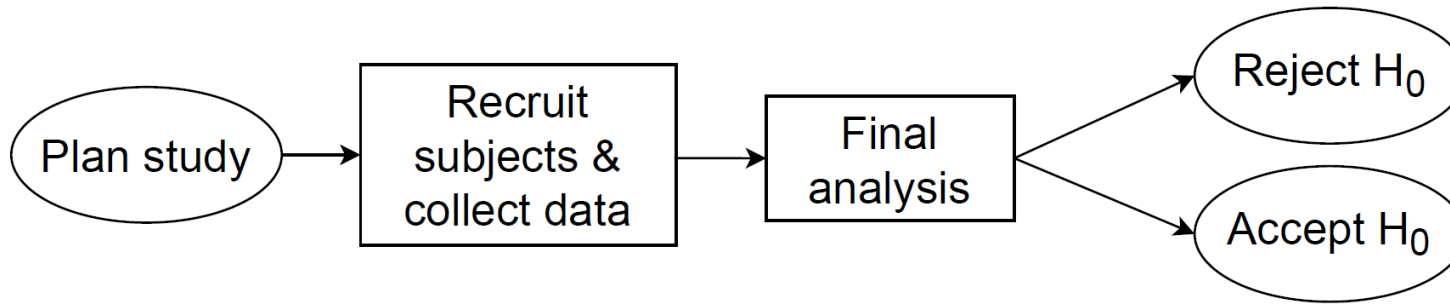
Fixed design



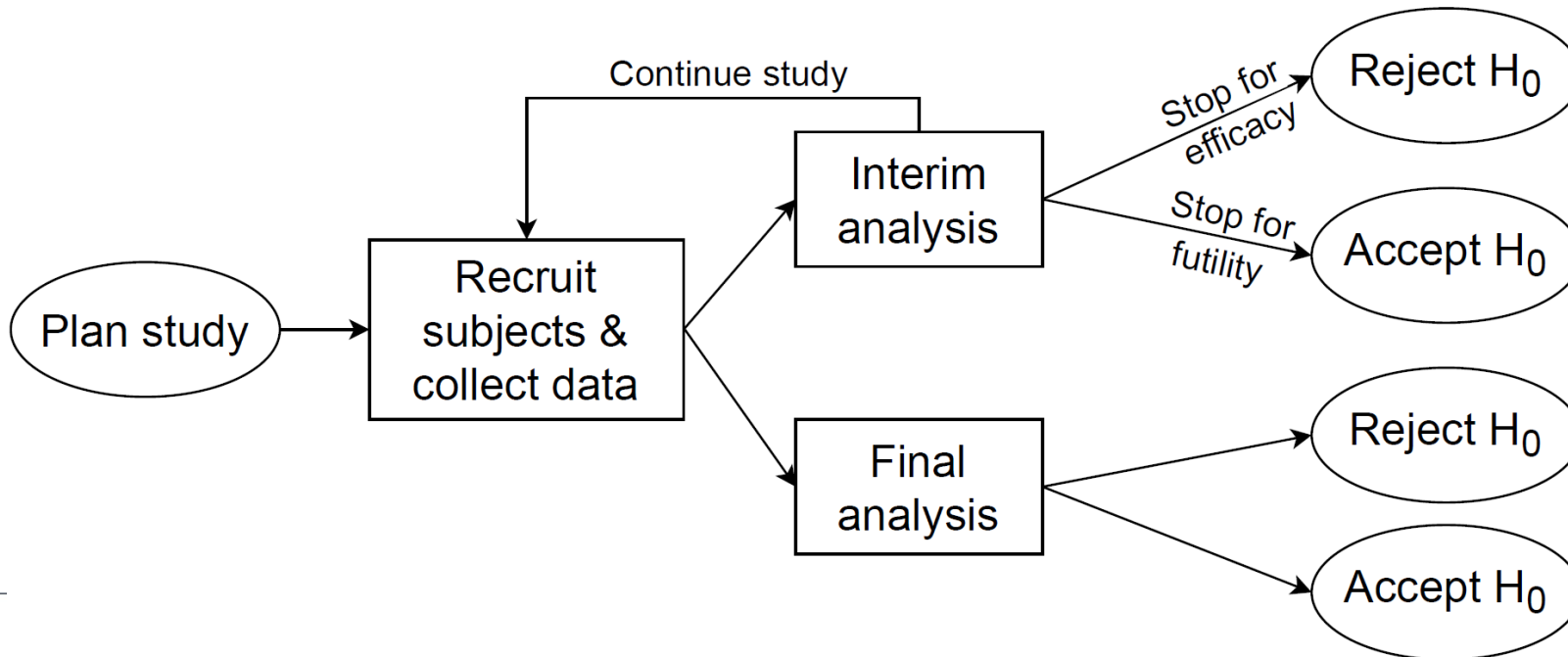
Fixed Sample Design vs Group Sequential Design

or: how I learned to stop worrying and accept the null

Fixed design

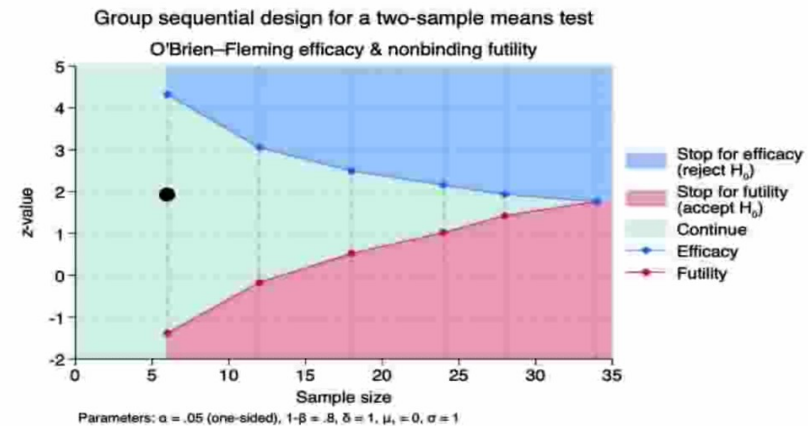


Group sequential design



GSD in action

Group sequential designs



STATA 18

From <https://youtu.be/hO2qW1NLrMk?si=gXsB8zDkAmq9K2IE> by Meghan Cain, StataCorp

gsdesign syntax

`gsdesign` *method* ... [, *designopts* *boundopts*]

where *method* ... refers to a power *method* that is used for sample-size calculation, *designopts* are options controlling the sample-size calculation, and *boundopts* are options controlling the calculation of the stopping boundaries.

<i>method</i>	Description
<code>onemean</code>	GSD for one-sample mean test
<code>twomeans</code>	GSD for two-sample means test
<code><u>oneproportion</u></code>	GSD for one-sample proportion test
<code><u>twoproportions</u></code>	GSD for two-sample proportions test
<code><u>logrank</u></code>	GSD for a log-rank test
<code><i>usermethod</i></code>	user-defined sample-size calculation

`gsdesign` supports the above methods when they are used to calculate sample size with simple random sampling.
To use an unsupported method, specify option `methodok`.

gsdesign syntax (continued)

<i>designopts</i>	Description
Main	
<i>methodopts</i>	method-specific options
<u>alpha</u> (#)	overall significance level for all tests; default is <code>alpha(0.05)</code>
<u>power</u> (#)	overall power for all tests; default is <code>power(0.8)</code>
<u>beta</u> (#)	overall probability of type II error for all tests; default is <code>beta(0.2)</code>
<u>onesided</u>	request a one-sided test; default is two-sided
<u>nfractional</u>	report fractional sample size
<u>force</u>	allow calculation with unsupported <i>methodopts</i>
<u>methodok</u>	allow calculation with unsupported <i>method</i>
<u>poweriteration</u> (<i>powiteropts</i>)	iteration options for the calculation of fixed-study sample size; not available with <i>method</i> <code>logrank</code> ; seldom used

GSD by simulation

```
. gsdesign onemeansampsize, power(0.9) alpha(0.025) m0(0) ma(0.5) startval(50) reps(500) seed(9) ///  
>                                onesided nlooks(4) efficacy(errobfleming) graphbounds
```

Group sequential design by simulation for a one-sample z-test
upper one-sided test

Efficacy: Error-spending O'Brien-Fleming style

Study parameters:

```
alpha = 0.0250 (upper one-sided)  
power = 0.9000  
m0 = 0.0000  
ma = 0.5000  
delta = 0.5000
```

Expected sample size:

```
H0 = 41.89  
Ha = 32.14
```

Info. ratio = 1.0183

```
N fixed = 41  
N max = 42
```

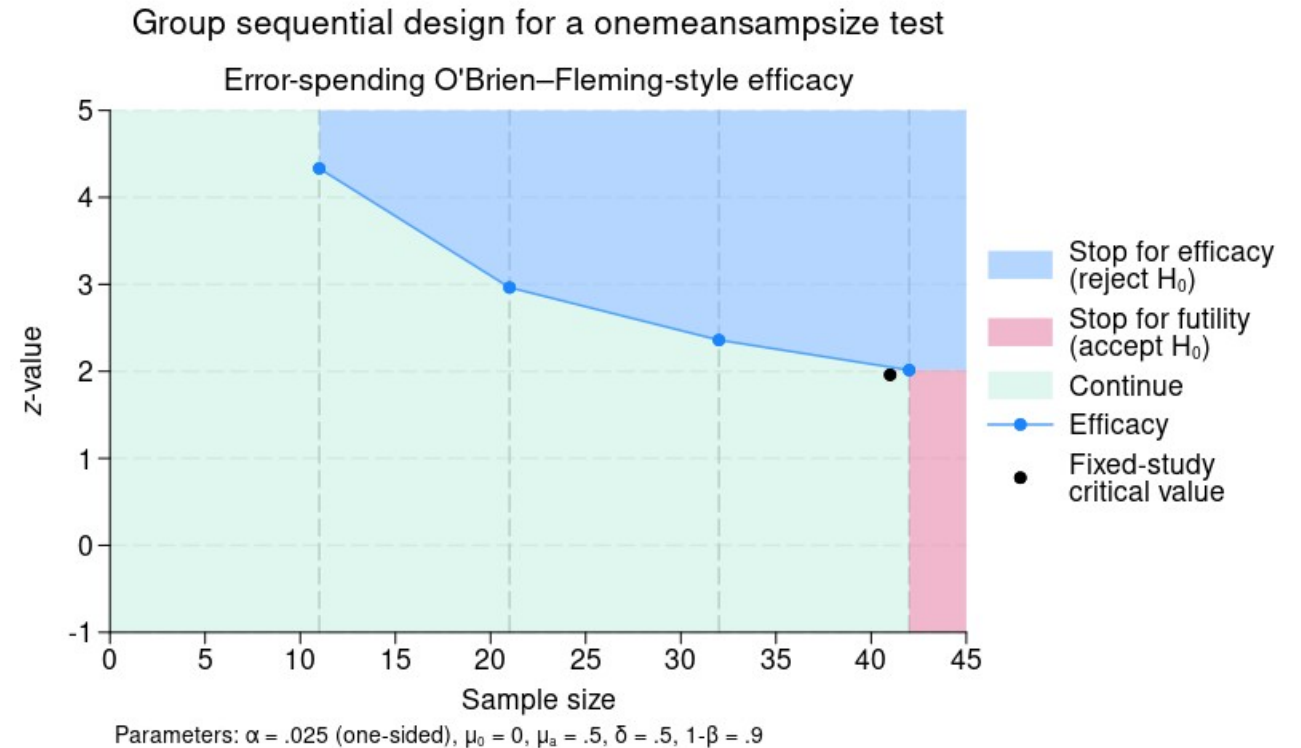
Fixed-study crit. value = 1.9600

GSD by simulation (continued)

Critical values, p-values, and sample sizes for a group sequential design

Look	Info. frac.	Efficacy Upper	p-value	Sample size N
1	0.25	4.3326	0.0000	11
2	0.50	2.9631	0.0015	21
3	0.75	2.3590	0.0092	32
4	1.00	2.0141	0.0220	42

Notes: Critical values are for z statistics; otherwise, use p-value boundaries. Requested information fraction not attained.



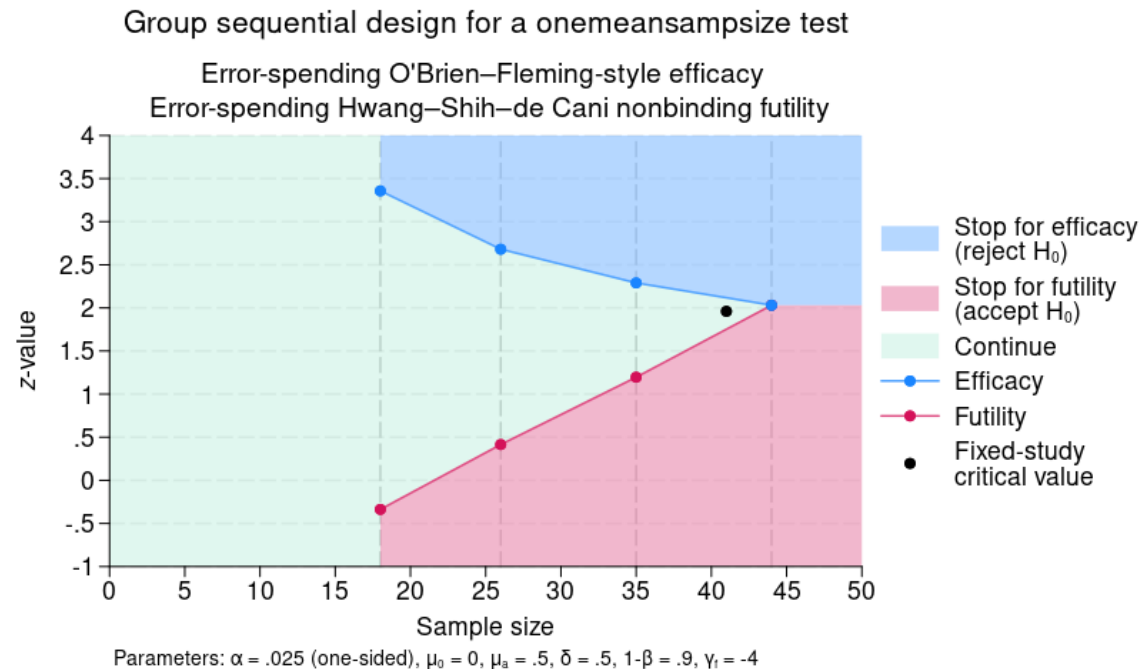
GSD with futility bounds

```
. gsdesign onmeansampsize, power(0.9) alpha(0.025) m0(0) ma(0.5) startval(50) reps(500) seed(9) ///  
>                                onesided information(40 60 80 100) efficacy(errobfleming)          ///  
>                                futility(hsdecani(-4)) graphbounds
```

Group sequential design by simulation for a one-sample z-test
upper one-sided test

Efficacy: Error-spending O'Brien-Fleming style

Futility: Error-spending Hwang-Shih-de Cani, nonbinding, $\gamma = -4.0000$





Code from all slides available upon request
aasher@stata.com