

Sample Size Calculation for Pilot Trials in R

Xiangji Ying

2025-03-08

This document is to help researchers calculate sample sizes for external pilot randomized controlled trials using R. It serves as a companion application to the forthcoming article titled “Determining sample size for pilot trials: A tutorial” in XX. For details on the methods, examples, and sample size tables, please refer to the paper. Unless otherwise specified, all calculations refer to the total number of participants that will need to be enrolled in the pilot trial.

Estimate a feasibility parameter (Box 1)

Example 1

When estimating a binary feasibility parameter (e.g., recruitment uptake), the pilot sample size is calculated to achieve a specified confidence interval width for a single proportion at a stated confidence level. While multiple methods exist for estimating confidence intervals, the Wilson method is recommended for small sample sizes ($n < 40$). The `prec_prop` function from the `presize` package performs this calculation.

```
library(presize)
prec_prop(p=0.1, conf.width=0.2, conf.level = 0.9, method = "wilson") # One-sided 95% CI

##
##      sample size for a proportion with Wilson confidence interval.
##
##      p      padj      n conf.width conf.level      lwr      upr
## 1 0.1 0.1380199 25.75892      0.2      0.9 0.03801995 0.2380199
##
## NOTE: padj is the adjusted proportion, from which the ci is calculated.
```

Example 2

When estimating a Poisson feasibility parameter (e.g., recruitment rate), the pilot sample size is calculated to achieve a specified confidence interval width for a single Poisson rate at a stated confidence level. While several methods exist for estimating confidence intervals for Poisson rates, two approaches are commonly recommended: (1) The Score method: Generally preferred for most situations; (2) The Exact method: Most reliable for small Poisson rate (≤ 5). The `prec_rate` function from the `presize` package performs these calculations.

```
library(presize)
prec_rate(r = 10, conf.width = 6, conf.level = 0.95, met = "score") # Two-sided 95% CI

##
##      sample size for a rate with score confidence interval
##
##      r      radj      x      time conf.width conf.level      lwr      upr
## 1 10 10.44031 43.62255 4.362255      6      0.95 7.440307 13.44031
##
## NOTE: 'x / r' units of time are needed to accumulate 'x' events.
```

```
prec_rate(r = 10, conf.width = 6, conf.level = 0.95, met = "exact") # Two-sided 95% CI

##
##      sample size for a rate with exact confidence interval
##
##      r radj      x      time conf.width conf.level      lwr      upr
## 1 10      10 46.25533 4.625533          6          0.95 7.328066 13.32807
##
## NOTE: 'x / r' units of time are needed to accumulate 'x' events.
```

Test feasibility progression criteria (Box 2)

Example 1

To test progression criteria based on binary feasibility parameters (e.g., recruitment uptake), the pilot sample size is calculated to determine whether a population proportion (goal threshold, H1) significantly differs from a hypothesized value (minimum threshold, H0).

To determine if a proportion is less than or greater than a specified value, two commonly recommended tests are the normal approximation proportion test with continuity correction and the binomial exact test. Both methods are considered conservative compared to normal approximation without continuity correction. The binomial exact test often results in larger sample sizes compared to the normal approximation proportion test with continuity correction.

The `MKpower` and `EnvStats` packages can calculate sample sizes using the normal approximation proportion test with continuity correction.

```
# Proportion test with continuity correction
library(MKpower)
power.prop1.test(p1 = 0.5, p0 = 0.2, sig.level = 0.05, power = 0.95,
                 alternative = "greater", cont.corr = TRUE)

##
##      Power calculation for testing a given proportion (with continuity correction)
##
##              n = 27.58252
##          delta = 0.3
##              p1 = 0.5
##              p0 = 0.2
##          sig.level = 0.05
## exact.sig.level = 0.01486287
##              power = 0.95
##          exact.power = 0.9075333
##          alternative = greater
##
## NOTE: n = total sample size

library(EnvStats)

##
## Attaching package: 'EnvStats'

## The following objects are masked from 'package:stats':
##
##      predict, predict.lm
```

```
propTestN(p.or.p1 =0.5, p0.or.p2 =0.2, alpha = 0.05, power = 0.95,
  sample.type = "one.sample", alternative = "greater",
  approx = TRUE,round.up = FALSE, correct = TRUE)
```

```
## [1] 27.58252
```

```
# Binomial exact test
```

```
alpha = 0.05
```

```
beta = 0.05
```

```
p_ll = 0.2
```

```
p_ul = 0.5
```

```
N <- 1:10000
```

```
CritVal <- qbinom(p = 1 - alpha, size = N, prob = p_ll)
```

```
Beta <- pbinom(CritVal, N, p_ul)
```

```
Power <- 1 - Beta
```

```
min(which(Power > 1-beta)) # Sample size for the pilot trial
```

```
## [1] 28
```

Example 2

To test progression criteria based on Poisson feasibility parameters (e.g., recruitment rate), the pilot sample size is calculated to assess whether a population Poisson rate (goal threshold, $H1$) significantly differs from a hypothesized value (minimum threshold, $H0$).

To determine if a Poisson rate is less than or greater than a specified value, one can use the exact test for Poisson rate. Sample size calculation based on the exact Poisson test can be performed using the `samplesizeOneRateExact` from the `lrstat` package.

```
library(lrstat)
```

```
samplesizeOneRateExact(beta = 0.1, lambdaH0 = 6, lambda = 10, D = 1, alpha = 0.05)
```

```
##   alpha attainedAlpha      power n lambdaH0 lambda D   r
## 1  0.05    0.04625304 0.9354296 5         6     10 1 40
```

Detect a feasibility problem (Box 3)

To observe at least one occurrence of a feasibility problem during the pilot trial, the pilot trial sample size can be calculated using the following formula:

$$n = \frac{\ln(1 - \gamma)}{\ln(1 - \pi)}$$

Where n represents the pilot trial sample size, π is the probability that a problem may occur, and γ denotes the level of confidence interval that researchers want to have with observing at least one occurrence of those problems.

```
gamma=0.95 # Confidence level
```

```
pi=0.05 # Anticipated probability that the problem of interest will occur in one participant
```

```
log(1-gamma)/log(1-pi) # Sample size for the pilot trial
```

```
## [1] 58.40397
```

Minimize total sample size of pilot and definitive trials (Box 4)

This method is to find the pilot trial sample size that minimizes the combined sample size required for both pilot and definitive trials. As we did not find existing packages for these calculations, we developed functions

to perform them based on information from relevant literature. The sample size for the definitive trial is calculated using the `pwr` package.

Non-central t Distribution (NCT) Method

```
# NCT method
library(pwr)

# Create a function that calculates the sample size based on the NCT method
optPilotSizeNCT <- function(d, alpha = 0.05, beta = 0.2, min.np = 2, max.np = 10000) {
  # Check if standardized effect size d is within valid range and not NA, NaN, or Inf
  if (is.na(d) || is.nan(d) || is.infinite(d) || d < 0 || d > 1) {
    stop("d must be between 0 and 1 and cannot be NA, NaN, or Inf")
  }

  # Check if alpha and beta are numeric and within the range 0 to 1
  if (!is.numeric(alpha) || alpha <= 0 || alpha >= 1 ||
      !is.numeric(beta) || beta <= 0 || beta >= 1) {
    stop("Both alpha and beta must be numeric and between 0 and 1")
  }

  # Check if the minimum and maximum pilot sample sizes per group are smaller than 2
  if (min.np < 2 || max.np < 2) {
    stop("The minimum allowed value for min.np and max.np is 2")
  }

  # Define the range for possible values of pilot sample size per group (Step 1)
  n_p <- min.np:max.np

  # Definitive trial sample size per group (Step 2)
  n_m <- pwr.t.test(d = d, sig.level = alpha, power = 1 - beta, alternative = "two.sided")$n

  # Initialize vector for overall sample size
  N <- numeric(length(n_p))

  # Loop to find pilot sample size that minimizes the overall sample size per group (Step 3, 4, 5)
  for (i in seq_along(n_p)) {
    N[i] <- n_p[i]*2*(qt(1-beta, df = 2*n_p[i]-2, ncp = qt(1-alpha/2, 2*n_m-2),
                      lower.tail = T))^2/d^2 # assuming estimated std deviation s=1
  }

  # Return the pilot sample size per group that minimizes the overall sample size
  optimal_pilot_size <- n_p[which.min(N)]

  # Calculate the corresponding inflated definitive trial sample size per group
  full_scale_size <- 2*(qt(1-beta, df = 2*optimal_pilot_size-2, ncp = qt(1-alpha/2, 2*n_m-2),
                        lower.tail = T))^2/d^2

  # Return the overall sample size per group
  overall_size <- optimal_pilot_size + full_scale_size

  # Create a list to store the results
  results_list <- list(
    n_pilot = optimal_pilot_size*2,
    n_full_scale = full_scale_size*2,
```

```

    n = overall_size*2 # total, both arms
  )

  # Return the results list
  return(results_list)
}

# Calculate the pilot sample size for the examples
optPilotSizeNCT(d=0.4, alpha = 0.05, beta = 0.2, min.np = 2, max.np = 1000)

## $n_pilot
## [1] 22
##
## $n_full_scale
## [1] 215.5131
##
## $n
## [1] 237.5131

```

Upper Confidence Limit (UCL) Method

```

# UCL method
library(pwr)
# Create a function that calculates the sample size based on the UCL method
optPilotSizeUCL <- function(d, alpha = 0.05, beta = 0.2, std.conf.level = 0.8,
                             min.np = 2, max.np = 10000)
{
  # Check if standardized effect size d is within valid range and not NA, NaN, or Inf
  if (is.na(d) || is.nan(d) || is.infinite(d) || d < 0 || d > 1) {
    stop("d must be between 0 and 1 and cannot be NA, NaN, or Inf")
  }

  # Check if alpha and beta are numeric and within the range 0 to 1
  if (!is.numeric(alpha) || alpha <= 0 || alpha >= 1 ||
      !is.numeric(beta) || beta <= 0 || beta >= 1) {
    stop("Both alpha and beta must be numeric and between 0 and 1")
  }

  # Check if std.conf.level is within valid range
  if (std.conf.level <= 0 || std.conf.level >= 1) {
    stop("std.conf.level must be between 0 and 1")
  }

  # Check if the minimum and maximum pilot sample sizes per group are smaller than 2
  if (min.np < 2 || max.np < 2) {
    stop("The minimum allowed value for min.np and max.np is 2")
  }

  # Define the range for possible values of pilot sample size per group (Step 1)
  n_p <- min.np:max.np

  # Definitive trial sample size per group (Step 2)

```

```

n_m <- pwr.t.test(d = d, sig.level = alpha, power = 1 - beta, alternative = "two.sided")$n

# Initialize vector for overall sample size
N <- numeric(length(n_p))

# Loop to find pilot sample size that minimizes the overall sample size per group (Step 3, 4, 5)
for (i in seq_along(n_p)) {
  N[i] <- n_p[i] + n_m * (2 * n_p[i] - 2) / qchisq(1 - std.conf.level, 2 * n_p[i] - 2)
}

# Return the pilot sample size per group that minimizes the overall sample size
optimal_pilot_size <- n_p[which.min(N)]

# Calculate the corresponding inflated definitive trial sample size per group
full_scale_size <- n_m * (2 * optimal_pilot_size - 2) / qchisq(1 - std.conf.level,
  2 * optimal_pilot_size - 2)

# Return the corresponding overall sample size per group
overall_size <- optimal_pilot_size + full_scale_size

# Create a list to store the results
results_list <- list(
  n_pilot = optimal_pilot_size*2 ,
  n_full_scale = full_scale_size*2,
  n = overall_size*2 # total, both arms
)

# Return the results list
return(results_list)
}

# Calculate the pilot sample size for the examples
optPilotSizeUCL(d=0.4, alpha = 0.05, beta = 0.2, std.conf.level = 0.8, min.np = 2, max.np = 1000)

## $n_pilot
## [1] 36
##
## $n_full_scale
## [1] 250.1075
##
## $n
## [1] 286.1075

#optPilotSizeUCL(d=0.4, alpha = 0.05, beta = 0.2, std.conf.level = 0.95, min.np = 2, max.np = 1000)

```

Rule out interventions unlikely to produce clinically important effects (Box 5)

This method calculates sample sizes for pilot trials aiming to exclude, with high confidence, interventions unlikely to yield clinically meaningful effects. The sample size estimation is based on the confidence interval width for differences between two groups.

For continuous outcomes, the `ci.stdmean2` function in the `statpsych` package can be used, assuming the population variance is known.

For binary outcomes, sample size calculation can be performed using the `prec_riskdiff` function in the `presize` package, which offers several methods, including Wald, Newcombe, Miettinen-Nurminen, and

Agresti-Caffo. For smaller sample sizes (fewer than 30 per group), the Newcombe and Agresti-Caffo methods are recommended. The Newcombe method is particularly suitable for proportions near 0 or 1. For moderate to large sample sizes (greater than 30 per group), the Agresti-Caffo, Newcombe hybrid score, and Miettinen-Nurminen methods are robust options. The Wald method is acceptable for samples with 100 or more in each group.

Continuous Outcomes

```
# Set parameters
oneside_ci_level = 0.8 # confidence level
alpha = (1-oneside_ci_level)*2 # alpha corresponding to the confidence limit
w = 0.3*2 # width of the confidence interval or twice of the clinically meaningful difference
d = 0 # assume between group difference is 0
R = 1 # assume 1:1 allocation

# Calculate pilot trial sample size
library(statspsych)
size.ci.stdmean2(alpha, d, w, R) # z test
```

```
##                                n1 n2
## Unweighted standardizer:      16 16
## Single group standardizer:    16 16
```

Binary outcomes

```
# Set parameters
oneside_ci_level = 0.8 # confidence level
alpha = (1-oneside_ci_level)*2 # alpha corresponding to the confidence limit
p1 = p2 = 0.5 # proportion of the participants with the outcome in both treatment and control groups
w = 0.1*2 # width of the confidence interval or twice of the clinically meaningful difference

# Calculate pilot trial sample size
library(presize)
prec_riskdiff(p1 , p2 , conf.width = w, conf.level= 1-alpha, method="newcombe")
```

```
##
##      sample size for a risk difference with newcombe confidence interval
##
##      p1  p2      n1      n2      ntot r delta  lwr upr conf.width conf.level
## 1 0.5 0.5 34.70799 34.70799 69.41598 1    0 -0.1 0.1      0.2      0.6
prec_riskdiff(p1 , p2 , conf.width = w, conf.level= 1-alpha, method="wald")
```

```
##
##      sample size for a risk difference with wald confidence interval
##
##      p1  p2      n1      n2      ntot r delta  lwr upr conf.width conf.level
## 1 0.5 0.5 35.41632 35.41632 70.83263 1    0 -0.1 0.1      0.2      0.6
```