TLDR:

* Begin with N of 10, increase by 10 until p < 0.05 or max N reached.
* This design has inflated type-I error.
* Lower p-value threshold needed to ensure specified type-I error rate.
* The number of interim analyses and max N affect the type-I error rate.
* Threshold can be identified using simulation.

For simplicity (and completeness), consider a block-randomized two sample comparison using a t-test. Then, begin with a sample size of 10 in each group and increase by 10 until p < 0.05 or until a maximum sample size is reached.

The code chunk below simulates this study design. This process was repeated many times under the null hypothesis (no difference in mean outcome) and the test results used to estimate the actual type-I error rate, as well as the median (IQR) total sample size. When the maximum sample size is 100 per group, the type-I error rate is about 19% and the median (IQR) sample size is 100 (100, 100) per group. If the sample size increment is 5 instead of 10, the type-I error rate is about 24%. When the maximum sample size is 1000 per group, type-I error rate is about 37% and the median (IQR) sample size is 1000 (200, 1000) per group. Thus, smaller sample size increment (more frequent interim analysis) and larger maximum sample size both result in a larger type-I error rate.

19%, 24%, and 37% are obviously unacceptably large type-I error rates. However, the specified type-I error rate (usually 5%) can be achieved by modifying the p-value threshold. The final code chunk below demonstrates that, by using a p-value threshold of 0.01 (found by guess-and-check), maximum sample size of 100, and sample size increment at 10, the type-I error rate is about 5%. This also has the effect that the maximum sample size is reached in about 95% of simulated studies, under the null.

## consider a two sample problem tested using t-test; start with 'n'

## in each group, increase by 'n' until p< 'alp' or maximum sample size

## 'nmx' reached

## simulate data

## n - sample size increment per group

## eff - effect size (difference in means; eff=0 is null hypothesis)

sim <- function(n=10, eff=0)

data.frame(y1=rnorm(n), y2=rnorm(n, mean=eff))

## compute test

## dat - data from sim() function

## alp - significance threshold

tst <- function(dat, alp=0.05)

t.test(dat$y1, dat$y2)$p.value < alp

## apply the 10+10 algorithm

## n - sample size in each of two groups

## eff - effect size (difference in means; eff=0 is null hypothesis)

## alp - significance threshold

## nmx - maximum sample size in each of two groups

alg <- function(n=10, eff=0, alp=0.05, nmx=1000) {

dat <- sim(n,eff)

rej <- tst(dat, alp)

while(nrow(dat) < nmx && !rej) {

dat <- rbind(dat, sim(n,eff))

rej <- tst(dat, alp)

}

list(n = 2\*nrow(dat), rej = rej)

}

## calculate overall type-I error by simulating study under null

## repeat procedure 5k times under null with nmx=100

out <- replicate(5000, alg(nmx=100), simplify=FALSE)

## estimate type-I error; fraction of times null rejected

mean(sapply(out, `[[`, 'rej'))

## distribution of total sample size (pairs)

quantile(sapply(out, `[[`, 'n')/2, probs=c(0.25, 0.50, 0.75))

## calculate overall type-I error by simulating study under null

## repeat procedure 5k times under null with nmx=100, and interim

## analysis at every 5 samples

out <- replicate(5000, alg(n=5, nmx=100), simplify=FALSE)

## estimate type-I error; fraction of times null rejected

mean(sapply(out, `[[`, 'rej'))

## distribution of total sample size (pairs)

quantile(sapply(out, `[[`, 'n')/2, probs=c(0.25, 0.50, 0.75))

## calculate overall type-I error by simulating study under null

## repeat procedure 5k times under null with nmx=1000

out <- replicate(5000, alg(nmx=1000), simplify=FALSE)

## estimate type-I error; fraction of times null rejected

mean(sapply(out, `[[`, 'rej'))

## distribution of total sample size (pairs)

quantile(sapply(out, `[[`, 'n')/2, probs=c(0.25, 0.50, 0.75))

## can the type-I error be fixed by adjusting threshold?

## repeat procedure

out <- replicate(5000, alg(alp=0.01, nmx=100), simplify=FALSE)

## estimate type-I error

mean(sapply(out, `[[`, 'rej'))

## distribution of total sample size (pairs)

table(sapply(out, `[[`, 'n'))