**A Standard Problem: Determining Sample Size**

Recently, I was tasked with a straightforward question: “In an A/B test setting, how many samples do I have to collect in order to obtain significant results?” As ususal in statistics, the answer is not quite as straightforward as the question, and it depends quite a bit on the framework. In this case, the A/B test was supposed to test whether the effect of a treatment on the success rate p had the assumed size e. The value of the success rate had to be estimated in both test and control group, i.e. p\_test and pcontrol. In short, the test hypotheses were thus

H0 : ptest = pcontrol vs.

H1 : ptest = pcontrol + e

Now, for each statistical test, we aim at minimizing (or at least controlling for) the following types of error:

* Type I: Even though H0 is true, the test decides for H1
* Type II: Even though H1 is true, the test decides for H0.

Since I can never remember stuff like this, I immediately started looking for a simple mnemonic, and I found [this one:](https://churchofrationality.blogspot.com/2011/03/type-i-and-type-ii-errors-trick-to.html)

The null hypothesis is often represented as H0. Although mathematicians may disagree, where I live 0 is an even number, as evidenced by the fact that it is both preceded and followed by an odd number. Even numbers go together well. An even number and an odd number do not go together well. Hence the null hypothesis (even) is rejected by the type I error (odd), but accepted by the type II error (even).

**The test statistic**

In the given setup, the test now runs as follows: Calculate the contingency matrix of successes in both groups, and apply [Fisher’s exact test](https://en.wikipedia.org/wiki/Fisher's_exact_test). If the test is negative, i.e. does not reject the null hypothesis, we have to repeat the experiment. However, we are quite sure that the null hypothesis is wrong and would like to prove that with as little effort as possible.

The basic question in this situation is “how many observations do I need to collect, in order to avoid both errors of Type I and II to an appropriate degree of certainty?”. The “appropriate degree of certainty” is parametrized in the probability of errors of Type I (significance level) and Type II (power). The default choices for these values are 0.05 for the significance level, and 0.8 for power: In 5% of cases, we reject a “true” H0, and in 20% of cases we reject a “true” H1. Quite clearly, only the power of the test (and not the significance level) depends on the difference of the parameters ptest and pcontrol.

**Existing functions in R**

Are there already pre-defined functions to calculate minimal required sample sizes? A bit of digging around yields a [match in the package Hmisc](https://rdrr.io/cran/Hmisc/src/R/bpower.s). There, the authors implement a method developed Fleiss, Tytun and Ury[1](http://www.sastibe.de/2020/01/sample_size_r/#fn:1). However, according to the documentation, the function is written only for the two-sided test case and does not include the continuity correction. I disagree with both decisions:

* the continuity correction term can grow quite large, and is always positive (see (5) in the cited paper). Thus, neglecting this term will always end in an underestimation of the necessary number of observations and may therefore lead to unsuccessful experiments.
* the two-sided case is not the norm, but rather the exception. When testing pcontrol vs. ptest, the counterhypothesis will almost always read “ptest > pcontrol“, since the measures taken assume to have, if any, a positive effect.

**A new R function: calculate\_binomial\_samplesize**

After these considerations, I decided to write my own function. Below is the code, the function allows for “switching the continuity correction off”, and for differentiating between the one-sided and the two-sided case. In the two-sided case without continuity correction, it coincides with “Hmisc:bsamsize”, as can be seen from the example provided.

#' Calculate the Required Sample Size for Testing Binomial Differences

#'

#' @description

#' Based on the method of Fleiss, Tytun and Ury, this function tests the null

#' hypothesis p0 against p1 > p\_0 in a one-sided or two-sided test with significance level

#' alpha and power beta.

#'

#'

#' @usage

#' calculate\_binomial\_samplesize(ratio0, p0, p1)

#'

#'

#' @param ratio0 Numeric, proportion of sample of observations in group 0, the control

#' group

#' @param p1 Numeric, postulated binomial parameter in the treatment group.

#' @param p0 Numeric, postulated binomial parameter in the control group.

#' @param alpha Desired significance level for the test, defaults to 0.05

#' @param beta Desired pwoer for the test, defaults to 0.8

#' @param one\_sided Bool, whether the test is supposed to be one-sided or two-sided.

#' @param continuity\_correction Bool, whether the sample size should be

#' Defaults to TRUE.

#'

#'

#' @return

#' A named numeric vector, containing the required sample size for the treatment group,

#' the control group, and the required total (the sum of both numbers).

#'

#'

#' @seealso [Hmisc::bsamsize()]

#'

#' @author Sebastian Schweer \email{sastibe\_r@@[sastibe.de](http://sastibe.de)}

#'

#' @references

#' Fleiss JL, Tytun A, Ury HK (1980): A simple approximation for calculating sample sizes

#' for comparing independent proportions. Biometrics 36:343-346.

#'

#' @examples

#'# Same result

#' alpha = 0.02; power = 0.9; fraction = 0.4; p\_lower = 0.23; p\_higher = 0.34

#'

#' Hmisc::bsamsize(p1= p\_lower, p2 = p\_higher, fraction = fraction,

#' alpha = alpha, power = power)

#'

#' calculate\_binomial\_samplesize(ratio0 = fraction, p1= p\_higher, p0 = p\_lower,

#' alpha = alpha, beta = power, one\_sided = FALSE, continuity\_correction = FALSE)

#'

#'

#' @export

calculate\_binomial\_samplesize <- function(ratio0,

p1,

p0,

alpha = 0.05,

beta = 0.8,

one\_sided = TRUE,

continuity\_correction = TRUE){

if(!is.numeric(ratio0) | !is.numeric(p1) | !is.numeric(p0))

stop("Input parameters ratio0, p0 and p1 need to be numeric.")

if(!is.numeric(alpha) | !is.numeric(beta))

stop("Input parameters alpha and beta need to be numeric.")

if(max(c(alpha, beta, ratio0, p0, p1)) >= 1 | min(c(alpha, beta, ratio0, p0, p1)) <= 0)

stop("Input parameters ratio0, p0, p1, alpha, beta need to be in the interval (0,1)")

delta = p1 - p0 # Nomenclature as in the paper

r = 1 / ratio0 - 1 # Uniting the definitions

if(one\_sided == FALSE) { # Last statement of the paper

alpha = alpha / 2

delta = abs(p1 - p0)

}

p\_bar = (p0 + r\*p1)/(r+1)

m\_dash\_1 = qnorm(1 - alpha, mean = 0, sd = 1)\*sqrt((r+1)\*p\_bar\*(1 - p\_bar))

m\_dash\_2 = qnorm(beta, mean = 0, sd = 1)\*sqrt(p1\*(1-p1) + r\*p0\*(1-p0))

m\_dash = ( m\_dash\_1 + m\_dash\_2 )^2 / (r \* delta^2)

if(continuity\_correction == TRUE){

m\_dash = m\_dash + (r + 1) / (r\*delta)

}

return(c(size\_0 = m\_dash,

size\_1 = r\*m\_dash,

size\_overall = m\_dash + r\*m\_dash))

}