Power Analysis for Studies with Multiple Hypotheses

Daniel B. Wright
Alder Graduate School of Education

Marianna E. Carlucci
Loyola University

July 2, 2018

Abstract

Power analysis packages and tables report power for single test statistics, like the overall F value from a oneway ANOVA. Often researchers are interested in multiple hypotheses. Power analysis for multiple hypothesis studies is discussed and an example provided. Studies designed to test multiple hypotheses should usually have larger samples than those designed to test just one hypothesis. An R function called `pwAnova` estimates the power for studies with multiple hypotheses. The function allows statistical significance, effect size, and direction to determine if each hypothesis tested is 'successful'. The code and further examples are included in the supplementary materials.

keywords: power, ANOVA, multiple hypotheses, contrasts, R.

Power analysis is recommended by scientific societies and journals (Wilkinson and the Task Force on Statistical Inference, 1999; Wright, 2003), and required by some funding bodies (e.g., `ies.ed.gov/funding/resources.asp`, accessed February 17, 2016). Tables (Cohen, 1992) and computer packages (Faul et al., 2007) can assist researchers in conducting traditional power analyses. The algorithms used for traditional power analysis require the researcher to choose a single test statistic.

The reality of many research projects is more complex. Most processes in nature cause multiple effects, and as such a useful theory of this process should make multiple predictions. Fisher summed this up nicely: "Make your theories elaborate" (see Cochran, 1965, p. 252). Cochran explains this statement: "What Sir Ronald meant ... was that when constructing a causal hypothesis one should envisage as many *different* consequences of its truth as possible" (p. 252). While rejecting a single point hypothesis will be satisfactory for some research projects, for others a set of findings is required to provide strong support for a theory. Predicting multiple outcomes provides a more stringent evaluation of the theory and is good scientific practice for these projects.

Steps for Power Analysis and an Example

Suppose a researcher wants to explore whether two manipulations affect people's attitudes as measured on an interval scale. One manipulation is designed to shift attitudes in one direction and the other manipulation to shift attitudes by a similar amount in the other direction. The researcher plans a three-condition study with a control group and a group for each manipulation. Power tables provide estimates for rejecting a single null hypothesis $\mu_1 = \mu_{control} = \mu_2$ associated with a significant F value. However, finding a significant F may not be enough to support the researcher's more *elaborate* theory. The researcher's theory predicts $\mu_1 < \mu_{control}$ and $\mu_{control} < \mu_2$. Researchers will have many different notions of what constitutes 'success' and therefore it is important that power analysis procedures are flexible. Here, the researcher wants to design a study where each of the experimental conditions are likely to be significantly different from the control condition in the hypothesized directions. The sample size suggested for rejecting a single F value under-estimates the sample size needed to produce sufficient power for this more complex and elaborate definition of 'success', assuming the α levels are the same.

Assume the researcher's total sample size is fixed at 1,500. The researcher wants to estimate the power associated with allocating different numbers of people into the control group than into each experimental condition. The researcher decides to have equal numbers of people in the two experimental conditions because there is no *a priori* knowledge about whether, for example, the spread of the data in these conditions will differ. With the traditional power analysis it is assumed that the researcher would allocate equal numbers of people to each of the three conditions because this maximizes the power for the single F statistic if several assumptions are true (e.g., equal variances within each group).

However, this is not appropriate for this more complex two-hypothesis situation described above because the control group is involved with both hypotheses, but each of the experimental conditions is used in only one of the hypotheses. While the researcher may be aware that more people should be in the control condition than in each of the experimental conditions, it is unlikely the researcher knows how many more people to have. The following simulation is used to estimate the optimal allocation into the control condition, providing the assumptions are valid. This illustrates the basic simulation approach used in this paper for power analysis. Simulation relies on computation rather than the mathematics and the non-central distributions used within traditional power analysis. Simulation is used in power analysis in some packages where mathematical solutions do not exist for many of the designs (e.g., Browne et al., 2009).

The four steps for any power analysis are:

1. Define criteria for 'success,'
2. Define data and the minimum effect sizes to detect,
3. Pose power analysis questions, and
4. Produce output, perhaps in the form of a plot.

Define ‘Success’

The first step in doing a power analysis is to define ‘success’. In traditional power analysis ‘success’ is defined in only one way, by a single specified test statistic being statistically significant. If there are several hypotheses of interest for the study, and according to Fisher there often should be, a more elaborate definition of ‘success’ is necessary. Currently, researchers often just choose one hypothesis thereby under-estimating the sample size that they should use.

In addition to allowing multiple hypotheses, the current approach recognizes that sole reliance on statistical significance is problematic for science (Cohen, 1990, 1994). For example, researchers may want to design their study so that it has a high likelihood of observing effect sizes larger than some prescribed amount. The current approach broadens the scope of what is meant by a ‘success’. Power remains the probability of ‘success’ conditional on assumptions about the data, but because ‘success’ differs from traditional power analysis the notion of power here also differs.

If there is a single interval outcome variable and the independent variable is a single categorical variable (the situation considered in this paper), specific hypotheses can be defined by a series of contrasts. Usually if a variable has k categories, there are $k-1$ contrasts explored in the analyses. For this example the researcher defines ‘success’ as having both hypotheses being statistically significant at $\alpha = .05$ and in the specified direction. There is much debate about the $\alpha = .05$ threshold, but having the achieved significance level less than 5% is often seen (for better or worse) as a minimum requirement to provide publishable evidence in favor of the researcher’s theory. Although there are two statistical tests here, no adjustment is made for the number of tests because both results are desired by the researcher.

Define Data

The second step is to define the data to simulate. For this example, assume the data are normally distributed within each of three conditions and that the true difference between each experimental condition and the control condition being what Cohen (1992) describes as a small effect (i.e., $.2\sigma$). These distributional assumptions are typically made with traditional power analysis so that mathematical solutions can be easily reached. These assumptions are not required with the function discussed in this paper, but are used in this example to allow comparison with traditional power analysis.

Pose Power Analysis Questions

The next step is to define the questions for the power analysis to address. Statistical power packages allow you to vary different parameters, like the sample size, and see how these affect other values, like the power (e.g., Faul et al., 2007). For the current example the question is: "How many of the 1,500 people should be in the control condition to maximize the probability of observing both effects being significant?" The simulated number of people in the control group ranges from 400 to 900 in increments of 10. This means each experimental group has between 300 and 550 people for each simulation.

```
## Error in library(robust):  there is no package called 'robust'
```

This procedure was repeated 10,000 times for each control condition size. The number of replications that should be used will depend on the needed precision of the estimates, the complexity of the model, and the effect sizes. Because the outcome 'success' is binary and the differences in power due to characteristics like allocating among conditions can be small, several thousand replications are used here.

Create a Plot

A plot is often useful for visualizing the results of a power analysis. Figure 1 shows the probability of rejecting both null hypotheses, $\mu_1 = \mu_{control}$ and $\mu_2 = \mu_{control}$, at $\alpha = .05$ and that both are in the direction hypothesized by the researcher (the probability of having a significant effect in the wrong direction, what Gelman and Carlin (2014) call a Type S error, is very low in this situation). The smooth curve shows the predicted

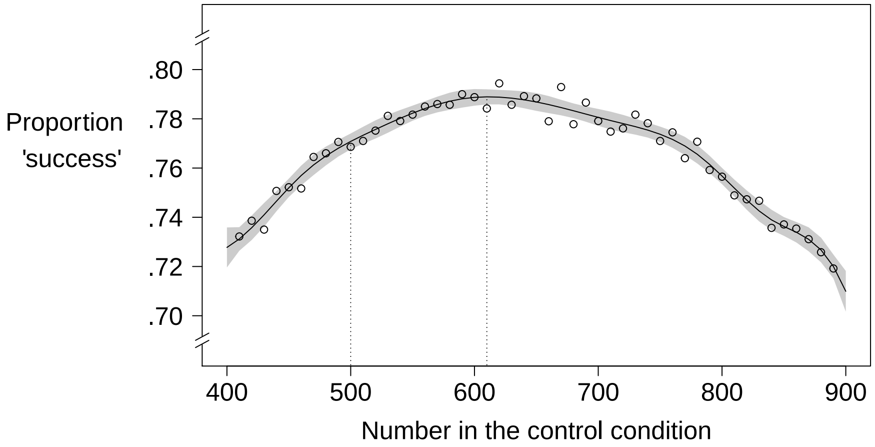

Figure 1: The proportion of 'successes' by the number of people in the control condition. The shaded region is ±2 standard errors using a B-spline within a logistic regression.

proportion of 'successes' using a logistic regression with a flexible spline to model the relationship. The shaded area is ±2 standard errors, which with 10,000 replications is narrow.

Dashed vertical lines are shown for equal sample sizes (500) in all conditions, which has a power of 0.771, and for the control sample size with the highest power (control size 610), which has power of 0.789. For comparison, **G*Power** (Faul et al., 2007) was used with these effect sizes and for having 500 people in each condition. It produced a power of approximately .91. Therefore, if a researcher wanted the study to find each hypothesis significant but conducted a traditional power analysis, the power would be overestimated. This could lead to researchers using too few participants. This example is used in the next section when describing the function.

The `pwAnova` function

The primary goal of `pwAnova` is too allow flexibility for defining 'success' based on multiple hypotheses that are often associated with the oneway ANOVA. The R environment (R Core Team, 2016) is used. Simulation allows flexibility with respect to the assumptions made about the data's distributions. This flexibility means that the user will need to think about the design, the data, and what constitutes a 'success' prior to estimating power. Traditional approaches to power analysis allow the user to calculate power without as much forethought about the study. There are advantages and disadvantages to require researchers to think more about their studies prior to conducting power analyses. The collection of examples in the supplementary materials will help guide researchers with some common designs.

Defining 'success' with `pwAnova`

There are a variety of ways that researchers might wish to define 'success'. The researcher may want to define 'success' according to a set of $k-1$ contrasts for a k category independent variable. The 'success' for the individual contrasts can be determined by effect size (here β divided by the residual standard error), p value (adjusted or un-adjusted for the number of contrasts), and the direction of the effect. The researcher can decide whether to require all the contrasts to meet their criteria (and all three of these criteria can be different for each contrast) or if fewer need to meet their criteria. The user can also use overall model fit to determine 'success' and this can be done with effect size (unadjusted or adjusted R^2) and/or the p value. With traditional power analysis only the significance of the overall fit is used to define 'success'. There is also an option to include user-defined functions for defining 'success'. Examples of this use include: power using non-point hypotheses, using other fit measures, using robust ANOVAs, non-standardized effect sizes, having all groups significantly different from the others, *etc.*. Use of this option is shown below in an example and in the supplementary materials.

Most of the options for `pwAnova` relate to defining 'success'. These options are listed in the supplementary materials and are illustrated in the examples.

Defining Data with pwAnova

The data can be in one of two formats, each placed into a slot called **dataset** of the **pwAnova** function. First, they can be a list of k functions that produce data for the k groups. It is likely this will involve R's built-in random number generators (e.g., **rnorm**) and **sample** (likely with **replace=TRUE**). The following is an example. Because there are three functions, the function stores $k = 3$ for the number of groups. Note that **n** and **ef** are used. These names must be used for varying different attributes that define the questions asked of the power analysis (e.g., "how do the sample size and power relate?"). These objects are created by the **pwAnova** function and control the sample sizes and the effect sizes for conditions, so this should be taken into account when defining these functions.

```
dataset <- list(
  function(n,ef=-.2,...) rnorm(n,ef),
  function(n,...)
    sample(1:5,size=n, replace=TRUE,prob=c(.1,.1,.1,.4,.3)),
  function(n,ef,...)
    sample(1:5,size=n,replace=TRUE,prob=c(1,2*ef,1,ef,4*ef)))
```

If the above code is executed, **dataset[[3]](10,1)** produces data for the third function with a sample size of 10 and an effect size of 1.

The second format to enter data is as a two-column data frame with the first column as the grouping variable and the second as the response variable. Replication data sets for each value of the variable in the first column are sampled from these with replacement as with a bootstrap. The number in each condition is fixed for all replications.

```
dataset <- matrix(c(rep(1:3,c(3,5,8)),
  43,43,46,64,53,43,43,45,23,2,34,23,23,21,43,54),ncol=2)
```

Some care is needed using this second approach to enter data. If sample data are used there will be variation due to sampling error and this error will be repeated each time that the data are created. This is discussed in detail in the first example of the supplementary materials.

Posing Power Analysis Questions with `pwAnova`

The researcher may be interested in how many different aspects of the data influence the power. This function allows the user to see the relationship between power and any aspect that is used to create the data. The variables `varyef` and `varyn` allow the user to vary aspects of the data independently for each condition. While these can be used to vary any aspect of the data, it is assumed that these are used to vary effects sizes and sample sizes, hence their names. The example used above in the text and as an example below assumes the researcher is interested in sample size allocation among conditions. Power analysis is often used to determine the overall sample size. Examples showing this are in the supplementary materials.

The format for `varyef` and `varyn` are described in the supplementary materials and illustrated in the examples. Briefly, they can be entered as matrices, vectors, or single values. If entered as a two-dimensional matrix the function assumes the columns refer to the different conditions and the rows refer to different sets of effect sizes or sample sizes to be tested. If a one-dimensional matrix or vector is entered, the function assumes this is either values for each condition or for different values for all conditions to be tested. If a single value is entered it is assumed that this value is used for all conditions and tests. Off-the-shelf "small", "medium", and "large" effect sizes in standardized units (Cohen, 1992) have not been incorporated. Baguley (2004), Baguley (2009), Lenth (2001), and others have argued that researchers should not be as reliant as they are on these conventions. Further, because of the flexibility of the data distributions no such "canned" effect sizes are available unless specific assumptions are made.

Creating Plots with pwAnova

The main output from pwAnova is a matrix with the proportion of ‘successes’ for each effect size and sample size combination. This matrix can be used for further analysis and for creating plots.

The option plotit (default TRUE) produces a plot of the proportion of ‘successes’ with one of the variables that is varied by either by varyn or varyef. The user can choose which variable to have on the *x-axis* with the varycolumn slot in pwAnova. The plot shows a flexible (spline) curve between the two variables and includes a region of ±2 standard errors. Details of how this is done are in the supplementary materials.

The complete function appears in the supplementary materials (Note: after the peer review process the CRAN location will be given here).

Repeating the Text Example using pwAnova

The following shows how to use pwAnova to conduct the power analysis example described earlier and also a more specific set of hypotheses that the researcher might have had. This allows us to show an application of pwAnova and to show the use of the extrasuccess option. First, the data are created according to the researcher's beliefs and the number of replications (reps) is defined.

```
EGdd <- list(function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef))
reps <- 10000
```

The following block of R code conducts the power analysis described in the text and produces the plot in the left panel of Figure 2. This is within sampling error of the plot in Figure 1, but with different axes and without some of the personal touches that we added to the earlier plot, like the slashes on the axis and writing the y axis label horizontally. The plot is just a “side-effect” of the function. The main output is a matrix that can be used for more personalized plot or in further statistical analyses.

The matrix of `ef`, `n`, and power value (the mean number of 'successes' divided by the number of replications) is stored as `egA`.

```
egA <- pwAnova(EGdd,conmat=contr.treatment(3,2),pcon = .05,
  signcon = c(-1,1),varyef = c(-.2,0,.2),dfv=4,replics=reps,
  varyn = cbind(seq(550,350,-5),seq(400,800,10),seq(550,350,-5)))
```

The `EGdd` tells the function to use the data defined above. The `conmat` option says to use R's built-in `contr.treatment` contrast and to use the second of three conditions as the baseline. `pcon` sets the α levels and because there is only one value it is the same for both contrasts. `signcon` says that for 'success' the value associated with the first contrast must be negative and the value associated with the second must be positive. `varyef` and `varyn` show the values for the effect sizes and sample sizes that will be varied. Because `varyef` has the same number of values as groups, the function determines that each of these corresponds to one of the data functions in `EGdd` (the `ef` value in those functions) and these are not varied. `varyn` has three sets of 41 values. The function simulates data for each of these 41 values. `dfv` is the number of degrees of freedom for the spline used in the plot. `replics` is for how many replications to have.

The right panel of Figure 2 represents a more elaborate hypothesis. If the researcher's theory predicts that the magnitude of the two effects should be similar but in opposite directions, the theory gains stronger support if in addition to having significant differences between each of the experimental groups and the control group, the magnitude of differences are not significantly different. This requires use of the `extrasuccess` option. There are several ways to measure this. One is if the size of the effects in terms of the differences in means are not significantly different. This can be written as:

```
g1 <- mean(dd[dd[,1]==2,2]) - dd[dd[,1]==1,2]
g3 <- dd[dd[,1]==3,2] - mean(dd[dd[,1]==2,2])
t.test(g1,g3,var.equal=TRUE)$p.value > .05
```

and is embedded in the following function call:

```
egB <- pwAnova(EGdd,conmat=contr.treatment(3,2),pcon = .05,
  signcon = c(-1,1),varyef = c(-.2,0,.2),replics=reps,
  varyn = cbind(seq(700,250,-10),seq(100,1000,20),seq(700,250,-10)),
  extrasuccess=list(function() {
     g1 <- mean(dd[dd[,1]==2,2]) - dd[dd[,1]==1,2]
     g3 <- dd[dd[,1]==3,2] - mean(dd[dd[,1]==2,2])
     t.test(g1,g3,var.equal=TRUE)$p.value > .05} ))
```

Other than the addition of the extra criterion for success, the range of the x-axis is changed by the `varyn` option to show that a smaller control group would be optimal.

Figure 2: The left panel recreates the plot from the illustrative example from Figure 1. The right panel adds the additional requirement that the magnitude of the two experimental effects should not be significantly different.

Discussion

Traditional power analysis reinforces the notion that the primary goal of a study is to achieve a single p value less than some value thereby warranting publication. This focus on the significance of a single test statistic simplifies what constitutes ‘success’ for many studies. Some designs can have only a single hypothesis to investigate. Where appropriate simple designs are preferred. Cohen’s students describe this in the extreme: “some of my students have spread the rumor that my idea of the perfect study is one with 10,000 cases and no variables. They go too far” (Cohen, 1990, p. 1305). This tongue-in-cheek ideal is in opposition to Fisher’s quotation at the start of this paper and what can occur when considering many potential causes that scientists investigate. For many scientific projects it is appropriate to investigate a multitude of effects that may be caused by any manipulation, or in association studies to speculate on the association patterns of a large set of variables (Wright, 2009). While power packages and tables exist for estimating power for simple one-hypothesis studies (so not quite as extreme as Cohen’s students jokingly suggest), the function presented here allows scientists to calculate power for studies with multiple hypotheses.

Acknowledgements

A reproducible report, using **knitr** (Xie, 2013), that was used to create this paper is available from the authors.

References

Thom S. Baguley. Understanding statistical power in the context of applied research. *Applied Ergonomics*, 35:73–80, 2004.

Thom S. Baguley. Standardized or simple effect size: what should be reported? *British Journal of Psychology*, 100:603–617, 2009.

William J. Browne, Mousa Golalizadeh Lahi, and Richard M. A. Parker. *A guide to sample size calculations for random effect models via simulation and the **MLPowSim** Software Package*. Bristol, United Kingdom, 2009. URL `http://www.bristol.ac.uk/cmm/software/mlpowsim/`.

William G. Cochran. The planning of observational studies of human populations. *Journal of the Royal Statistical Society. Series A.*, 128:234–266, 1965.

Jacon Cohen. Things I have learned (so far). *American Psychologist*, 45:1304–1312, 1990.

Jacon Cohen. A power primer. *Psychological Bulletin*, 112:155–159, 1992.

Jacon Cohen. The earth is round ($p < .05$). *American Psychologist*, 49:997–1003, 1994.

Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Alex Buchner. G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, 39:175–191, 2007.

Andrew Gelman and John Carlin. Beyond power calculations: Assessing type S (sign) and type M (magnitude) errors. *Perspectives on Psychological Science*, 9:641–651, 2014.

Frank E Harrell Jr, with contributions from Charles Dupont, and many others. *Hmisc: Harrell Miscellaneous*, 2015. URL `https://CRAN.R-project.org/package=Hmisc`. R package version 3.17-1.

Russ V. Lenth. Some practical guidelines for effective sample size determination. *The American Statistician*, 55:187–193, 2001.

Theodre Micceri. The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, 105:156–166, 1989.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

Jiahui Wang, Ruben Zamar, Alfio Marazzi, Victor Yohai, Matias Salibian-Barrera, Ricardo Maronna, Eric Zivot, David Rocke, Doug Martin, Martin Maechler, and Kjell Konis. *robust: Robust Library*, 2014. URL `https://CRAN.R-project.org/package=robust`. R package version 0.4-16.

Leland Wilkinson and the Task Force on Statistical Inference. Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, 54: 594–604, 1999.

Daniel B. Wright. Making friends with your data: Improving how statistics are conducted and reported. *British Journal of Educational Psychology*, 73:123–136, 2003.

Daniel B. Wright. Causal and associative hypotheses in psychology: Examples from eyewitness testimony. *Psychology, Public Policy, and Law*, 12:190–123, 2009.

Yihue Xie. *Dynamic documents with R and **knitr***. Chapman and Hall/CRC, Bocan Raton, FL, 2013.

Supplementary A: The Function

Note: This will appear on CRAN after the review process.

```
pwAnova <- function(dataset,replics,
   k = ifelse(is.matrix(dataset) || is.data.frame(dataset),
              length(unique(dataset[,1])),length(dataset)),
   dcon=0,pcon=1,signcon=0,numcon=k-1,conmat=contr.treatment(k,1),
   adjustcon="none",radjust=FALSE,r2size=0,r2p=1,extrasuccess=NULL,
   rseed=FALSE,varyef=0,varyn=0,plotit=TRUE,
   varycolumn=NULL,dfv = 10, ...)
  {
  if (rseed != FALSE) set.seed(rseed)
  if (plotit) require(splines)

  ##Checking input.
  if (is.data.frame(dataset)) dataset <- as.matrix(dataset)
  if (is.matrix(dataset)==FALSE && is.list(dataset)==FALSE)
     stop("dataset not of proper type")
  if (is.matrix(dataset) && ncol(dataset) != 2)
     stop("your data set needs two columns")
  if (numcon > k-1) warning("It appears you are requiring more contrasts
     to be successful than you have contrasts. Computation will proceed.
     Hope you know what you are doing.")
  if (numcon < 1) warning("It appears you are requiring less than one
     contrast to be successful. There are other ways to performance this.
     Hope you know what you are doing.")
  vinput <- list(dcon,pcon,signcon)
  vnames <- c("dcon","pcon","signcon")
  for (q in seq_along(vinput))
    if (length(vinput[[q]]) != 1 && length(vinput[[q]]) != k-1)
      stop(paste(vnames[q], "not proper length"))
  if (is.matrix(conmat) && (nrow(conmat) != k || ncol(conmat) != k-1))
    stop("conmat not the right dimensions")
  # If a non-contrasts keyword is used it gets dealt with by contrasts()

  #making varyef and varyn
  r <- max(NROW(varyn),NROW(varyef))
  if (r<=k) r <- 1
  if (NROW(varyef) > k && NROW(varyn) > k && plotit) {
        plotit <- FALSE
        warning("You are varying by both n and ef. The plotting is not
           designed for this and is off. Also, this function
           assumes these go equal numbers of trials. Be cautious.")}
  if (is.matrix(dataset) && (length(varyef) == k || NCOL(varyef) == k))
    warning("Looks like you are varying the effect size with the dataset
            as a data matrix. This may not produce what you want.
            Just the initial value in each row is used.")
```

```
if ((length(varyn) == 1 && varyn%%k != 0) ||
    (is.vector(varyn) && length(varyn) > k && any(varyn%%k!=0)))
    warning("varyn not divible by k, values have be truncated.
           Enter as a matrix if you want don't want truncation.")
if (length(varyn) == 1) n <- matrix(trunc(varyn/k),ncol=k,nrow=r)
if (is.vector(varyn) && length(varyn) > k)
  n <- replicate(k,trunc(varyn/3))
if (length(varyn) == k) n <- t(replicate(r,varyn))
if (is.matrix(varyn)) n <- varyn

if (length(varyef) == 1) ef <- matrix(varyef,ncol=k,nrow=r)
if (length(varyef) > k) ef <- replicate(k,varyef)
if (length(varyef) == k) ef <- t(replicate(r,varyef))
if (is.matrix(varyef)) ef <- varyef
if (identical(dim(ef),dim(n))==FALSE)
    stop("ef and n are not same dimensions. If you vary both
          they must have the same r values.")
if (any(n < 3))
  stop("Some population group sizes too small. Minimum is set to 3.")
pwOutput <- matrix(nrow=nrow(ef),ncol=2+2*k)
for (i in 1:r){
  numsucc <- 0
  nv <- n[i,]
  efv <- ef[i,]
  for (j in 1:replics){
    if (is.matrix(dataset)==FALSE) {
       dv <- {}
       for (x in 1:k) dv <- c(dv,dataset[[x]](n=nv[x],ef=efv[x]))
       dd <- as.data.frame(cbind(rep(1:k,nv),dv))}
    if (is.matrix(dataset)==TRUE && ncol(dataset)==2){
       dd1 <- rep(unique(dataset[,1]),nv)
       dd2 <- {}
       for (w in 1:k){
            subd <- dataset[dataset[,1]==unique(dataset[,1])[w],]
            dd2 <- c(dd2,
              subd[sample(1:nrow(subd),size=nv[w],replace=TRUE),][,2])}
       dd <- as.data.frame(cbind(dd1,dd2))
       }
    if (min(table(dd[,1])) < 3){
      warning("A group in a sample had a group n of 0, 1, or 2.
        The trial was assigned failure. It is possible varyn is
        not what you intended.")
      next()
    }
# the following needed so dd available within
# extrasuccess loop
assign("dd",dd,envir=.GlobalEnv)
```

```
  succ <- TRUE
  groupv <- as.factor(dd[,1])
  contrasts(groupv) <- conmat
  lmout <- lm(dd[,2]~groupv)
  if (summary(lmout)$r.square < r2size && radjust == FALSE) succ <- FALSE
  if (summary(lmout)$adj.r.square < r2size && radjust == TRUE) succ<-FALSE
  if (pf(summary(lmout)$fstat[1],summary(lmout)$fstat[2],
    summary(lmout)$fstat[3],lower.tail=FALSE) > r2p) succ <- FALSE
  if (any(signcon != 0))
   {if (length(signcon)==1) signcon <- rep(signcon,k-1)
    signson <- sum(sign(coef(lmout)[2:k])*signcon >= 0)
    if (signson < numcon) succ <- FALSE}
  if (any(pcon!= 1))
   {if (length(pcon)==1) pcon <- rep(pcon,k-1)
    pon <- sum(summary(lmout)$coef[2:k,4] < pcon)
    if (pon < numcon) succ <- FALSE
        }
  if (any(dcon!= 0))
   {if (length(dcon)==1) dcon <- rep(dcon,k-1)
    coefstan <- summary(lmout)$coef[2:k,1]/summary(lmout)$sigma
    don <- sum(coefstan > dcon)
    if (don < numcon) succ <- FALSE}
    if (is.null(extrasuccess)==FALSE)
    for (p in seq_along(extrasuccess))
       if (extrasuccess[[p]]()==FALSE) succ <- FALSE
    numsucc <- numsucc +succ
    }
     pwOutput[i,] <- c(i,numsucc/replics,ef[i,],n[i,])
      }
    colnames(pwOutput) <- c("rep","propsucc",
          paste("ef",1:k,sep=""),paste("n",1:k,sep=""))

if (plotit && r > k) {
    if (is.null(varycolumn))
      varycolumn <- 2+ which.max(c(apply(ef,2,sd),c(apply(n,2,sd))))
    x <- pwOutput[,varycolumn]
    x2 <- seq(min(x),max(x),length.out=200)
    if (r < 12 && dfv > r-3) {
      dfv <- r-3
      print("The df for bs you entered has been changed")
         }
    if (dfv < 1) {plotit <- FALSE
       warning("r to small for plotting")}
    ml <- predict(glm(cbind(replics*pwOutput[,2],replics*(1-pwOutput[,2]))
                      ~bs(x,df=dfv),family=binomial),
                  type="response",newdata=data.frame(x=x2),se.fit=TRUE)
    plot(x,pwOutput[,2],ylim=c(max(min(ml$fit-2*ml$se,pwOutput[,2]),0),
                               min(max(ml$fit+2*ml$se,pwOutput[,2]),1)),
       xlab=paste("Variable in",colnames(pwOutput)[varycolumn]),
       ylab="Proportion success")
    polygon(c(x2,rev(x2)),
       c(ml$fit-2*ml$se.fit,rev(ml$fit+2*ml$se.fit)),
          border=NA,col="grey90")
whichmx <- which.max(ml$fit)
powmax <- ml$fit[whichmx]
lines(x,pwOutput[,2],col="grey65")
lines(x2,ml$fit)
lines(rep(x2[whichmx],2),c(0,ml$fit[whichmx]),lty=3)
points(x,pwOutput[,2],col="grey65")
}
  return(pwOutput)
}
```

Supplementary B: Function Options

The following are options for `pwAnova`. Most of these are for defining 'success'.

`dataset` The data can be entered in one of two formats. First, it may be entered as a `list` of functions that define data for `k` conditions. The values `ef` and `n` can be used to vary any parameters used in constructing these functions, probably from a combination of R's built-in random number generators (e.g., `rnorm`) and `sample`. The second format is as a $n \times 2$ matrix of data where the first column is the group variable and the second column is the response variable. There is no default.

`replics` The number of replications to have. There is no default.

`k` This is the number of conditions. The default is determined from the number of conditions defined by `dataset`. The value `k` is used within the function. The user may enter others values, but if a value other than the number of conditions in `dataset` is entered then there may be unintended consequences. For most purposes the user should not enter any value for this as the default is appropriate.

`dcon` A vector of length $k-1$ for the effect size of each contrast, defined as β/sd where sd is the residual standard error reported by `summary.lm`. If only a single value is used this is assumed for all contrasts. The default is 0, and 0 should be used for any contrast for which the effect size is not used for defining 'success'.

`pcon` A vector of length $k-1$ for the p value needed for each contrast. If only a single value is used this is assumed for all contrasts. The default is 1, and 1 should be used for any contrast for which the significance of the contrast is not used for defining 'success'.

`signcon` A vector of length $k-1$ for the sign of the βs. The elements of the vector should be -1, 0, or 1, to match the output of the R function `sign`. If a single value is given it is assumed to be the value for all contrasts. The default is 0, and 0 should be used for any contrast for which the direction of the effect is not used for defining 'success'. In the rare case that $\hat{\beta} = 0$, this counts as a 'success' for this contrast for the direction (though it would be non-significant and have a zero effect size so would not pass these hurdles if they are included).

`numcon` The number of contrasts that need to meet criteria to have ‘success’. Defaults to $k - 1$.

`conmat` A contrast matrix of the appropriate dimension (k by $k - 1$) or reserved word (e.g., the `contr.treatment(3,2)`) appropriate for the R `contrasts` function’s `contrasts` slot. The default for this is `contr.treatment(k-1,1)`.

`adjustcon` The value entered into the R function `p.adjust` (see `p.adjust.methods`). The number of p values used is the number of elements of `pcon` not equal to 1. The default is `"none"`, which corresponds to no adjustment.

`radjust` Whether to use the adjusted or unadjusted R^2 produced by R’s `summary.lm`. Default is `FALSE` for unadjusted.

`r2size` The critical value for the R^2 (adjusted or unadjusted), defaults to 0. If the `radjust = TRUE` and adjusted $R^2 < 0$ this will be below the default size for `r2size`. This means that if `r2size` has not been changed this will produce a ‘failure’, which is appropriate.

`r2p` The critical p value for the overall fit of the model. Defaults to 1.

`extrasuccess` A list of functions appropriate for R that can be `TRUE` or `FALSE` with variables active when this part of the function is evaluated. For example,

```
extrasuccess <- list(function(...)
pi < 4,function(...)  mean(dd[,2])>1)
```

will work because `dd` is the name of the `data.frame` used within the function and it is in the global environment when this function begins evaluating ‘success’. Thus, this option may require some knowledge of the internal workings of `pwAnova` (though for many procedures the user only needs to know about `dd`). Examples are included later in this paper. Defaults to `NULL`.

`rseed` This controls the seed to allow exact replication of the simulated results by using any integer (non-integers are truncated). The default (`FALSE`) is not to change the seed. The value `NULL` can also be used which causes the package to re-set the seed based on the time and ID process.

`varyef` This can be entered as a matrix, a vector (or one-dimensional matrix), or a single value. If entered as a matrix it is a $r \times k$ matrix where r is the number of values which are evaluated and k is the number of conditions. If a vector is input with length greater than `k` it is assumed that its length is r and the values are replicated across the `k` columns. If the length is equal to `k` then it is assumed $r = 1$ and each value of `varyef` corresponds to one of the k conditions. If a single numeric value is given, this value is replicated to create a $1 \times k$ matrix.

`varyn` This can be entered as a matrix, a vector (or one-dimensional matrix), or a single value. If entered as a matrix it is a $r \times k$ matrix where r is the number of values which are evaluated and k is the number of conditions. If a vector is input with length greater than `k` it is assumed that its length is r and the values are divided by `k` (i.e., it is assumed the value is the total sample size) and replicated across the `k` columns. If the length is equal to `k` then it is assumed $r = 1$ and each value of `varyn` corresponds to one of the k conditions. If a single numeric value is given, this value is divided by `k` replicated to create a $1 \times k$ matrix.

`plotit` A binary (`TRUE/FALSE`) variable for whether to include a plot. The default is `TRUE`.

varycolumn This tells the function which column to plot on the *x-axis* of the plot. The default is to use the variable with the greatest variance for all the conditions. The numbers $1 \dots k$ are for the effect sizes (varyef) for each of the k conditions and the numbers $k+1 \dots 2k$ for the sample sizes (varyn) for each of the k conditions.

dfv Controls the degree of freedom of the spline in the plot. The default is 10 if the number of unique values in varycolumn is at least 12, otherwise it defaults to 3 less than the number of unique values. The default is higher than will be appropriate for many uses.

Supplementary C: Examples

Several examples illustrate the use of pwAnova. A set of examples will be maintained online. Others are welcome to submit examples, preferably using **knitr** (?Xie, 2013), to the authors.

Example 1. Entering Data with Data Matrix versus Functions

The first example creates a two-column (sample) data matrix with 500 cases and five groups. The interest is in how varying the overall sample size affects power. Sample size and power increase together, but the researcher is interested in how different ways of determining 'success' affect Type 1 error and power. For the plots in the top row of Figure 3 the null hypothesis is true so the y axis shows the Type 1 error. In the left panel all four contrasts need to be significant at $\alpha = .05$ (without adjusting for the number of contrasts), but in the right panel only one of them needs to be significant. Requiring all the contrasts to be significant produces a Type 1 error rate well below $\alpha = .05$ and requiring only one to be significant produces a Type 1 error rate well above $\alpha = .05$.

The null hypothesis was true when creating these sample data so the user might expect that neither of these should be affected by the sample size, but for both of these the probability of 'success' increases with sample size. When entering the data as a data matrix and then resampling from sample data, sampling variability in the original data is repeated and magnified as the re-sample size increases. Thus, some care is necessary using the data entry approach. This example is repeated below using functions.

The bottom row of Figure 3 shows the situation when the null hypothesis is false. The data in the five conditions are drawn from normal populations with μs of 0, .1, .1, .3, .3 in standard deviation units. pwAnova then treats these samples as their respective populations. Here, even when there are some differences among means, the probability of having all four contrasts significant remains low (so Type 2 error is high and power is low). The function abline is used in the lower right panel after the function call to add a dashed line for the often suggested power level of .80. Other functions can also be used to add to these plots.

```
reps <- 1000
par(mfrow=c(2,2))
dd1ab <- cbind(gr <- sample(1:5,500,replace=TRUE),rnorm(500))
options(warn=-1)
```

```
eg1a <- pwAnova(dd1ab,replics=reps,varyn=seq(100,500,20),
                pcon=.05,dfv=3)
eg1b <- pwAnova(dd1ab,replics=reps,varyn=seq(100,500,20),
               pcon=.05,numcon=1,dfv=5)
dd1cd <- cbind(gr <- sample(1:5,1000,replace=TRUE),
               rnorm(1000)+.1*(gr>1)+.2*(gr>3))
eg1c <- pwAnova(dd1cd,replics=reps,varyn=seq(100,500,20),
               pcon=.05,dfv=3)
eg1d <- pwAnova(dd1cd,replics=reps,varyn=seq(100,500,20),
               numcon=1,pcon=.05,dfv=5)
abline(h=.8,lty=3)
```

Figure 4 shows the same power analyses but using the functions to enter the data rather than relying on entering a sample data matrix. Here, the top row does show that the probability of a Type 1 error is not systematically related to the sample size. The sample size is related to the power values as shown in the lower row, but the power of having all four contrasts significant remains low in the lower left panel, and high in the lower right panel.

```
reps <- 1000
dd1B <- list(function(n=nv,ef=efv,...) rnorm(n,ef),
             function(n=nv,ef=efv,...) rnorm(n,ef),
             function(n=nv,ef=efv,...) rnorm(n,ef),
             function(n=nv,ef=efv,...) rnorm(n,ef),
             function(n=nv,ef=efv,...) rnorm(n,ef))
par(mfrow=c(2,2))
eg1aB <- pwAnova(dd1B,replics=reps,varyn=seq(100,500,20),
          varyef=0,pcon=.05,dfv=3)
eg1aB <- pwAnova(dd1B,replics=reps,varyn=seq(100,500,20),
          varyef=0,pcon=.05,numcon=1,dfv=3)
eg1cB <- pwAnova(dd1B,replics=reps,varyn=seq(100,500,20),
               varyef=c(0,.1,.1,.3,.3),pcon=.05,dfv=5)
eg1dB <- pwAnova(dd1B,replics=reps,varyn=seq(100,500,20),
               varyef=c(0,.1,.1,.3,.3),numcon=1,pcon=.05)
abline(h=.8,lty=3)
```

There are two take-home messages from this example. First, as evident from comparing the left panels with the right panels, when the probability of a single contrast being significant, call this x, is small, x^4 is very small. Therefore large samples are needed if the researcher wants all contrasts to be significant. However, as seen in the right-side panels, if only one contrast is necessary to declare 'success' this will often occur even when the groups are sampled from the same distribution. This is why adjusted p values are often used. Second, if a data matrix is used to enter data into `pwAnova`, the population distribution should be entered because if sample data are entered idiosyncratic variability in the sample will be re-sampled in the function.

Example 2. Exploratory methods

The researcher is asking if there are any significant pairwise differences for the means from a five-group ANOVA. As there are five conditions, there are 10 pairwise comparisons. This cannot be represented in the contrast matrix with 4 contrasts, so the `extrasuccess` slot is used in conjunction with `pairwise.t.test` function. Holm's method is used to adjust for the number of p values calculated (this is the default for `pairwise.t.test`), so that the Type 1 error rate should be near the nominal rate. Holm's methods is preferred over competitors like Bonferroni's method because it maintains family-wise Type 1 error rate while having more power. For the left panel the null hypothesis is true. In the right panel $\mu_1 = 0; \mu_2 = \mu_3 = .1; \mu_4 = \mu_5 = .3$ in standard deviation units. All distributions are normal. The left panel shows that Type 1 error is maintained, and right panel shows the power for different sample sizes. The bottom row uses no adjustment for the number of p-values and the proportion of 'successes' increases substantially.

```
reps <- 1000
dd3 <- list(function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef))
par(mfrow=c(2,2))
eg3a <- pwAnova(dd3,dfv=4,
        replics=reps,varyef=rep(0,5),varyn=seq(100,500,20),r2p=.05,
        extrasuccess = list(
          function() any(pairwise.t.test(dd[,2],dd[,1],
              p.adjust.method="holm")$p.value < .05,na.rm=TRUE)))
eg3b <- pwAnova(dd3,dfv=4,
        replics=reps,varyef=c(0,.1,.1,.3,.3),varyn=seq(100,500,20),
        r2p=.05,extrasuccess = list(
          function() any(pairwise.t.test(dd[,2],dd[,1],
              p.adjust.method="holm")$p.value < .05,na.rm=TRUE)))
eg3c <- pwAnova(dd3,dfv=4,
        replics=reps,varyef=rep(0,5),varyn=seq(100,500,20),
        extrasuccess = list(
          function() any(pairwise.t.test(dd[,2],dd[,1],
              p.adjust.method="none")$p.value < .05,na.rm=TRUE)))
eg3d <- pwAnova(dd3,dfv=4,
        replics=reps,varyef=c(0,.1,.1,.3,.3),varyn=seq(100,500,20),
        extrasuccess = list(
          function() any(pairwise.t.test(dd[,2],dd[,1],
              p.adjust.method="none")$p.value < .05,na.rm=TRUE)))
```

Example 3. Estimating a single power value

Sometimes researchers want to estimate a single power value. Suppose the researcher assumes that in a three group study two groups are drawn from the same normally distributed population and the other group from a normal distribution with a mean half of a standard deviation higher. Suppose also that the researcher knows there are 30 people in each group. For 'success' the researcher wants to observe that the third group is significantly different from each of the other two. Because only a single situation is explored no plot is made. The function outputs a matrix (not a vector) with one row. Statistics can be done on this row. Here Wilson's 95% confidence interval is found using the `binconf` function in **Hmisc** (Jr et al., 2015). It is worth stressing that this confidence interval is just the variability in this procedure and does not take into account the uncertainty in the assumptions (increasing the sample size to extremely large numbers will not tell you much more about the true value of the power because of the uncertainty in the assumptions about the model and effect sizes).

A second example is shown where the researcher is interested just in whether the overall test statistic for the ANOVA is significant. This less elaborate definition of 'success' produces has higher power. This is the only definition of 'success' for traditional power analyses. Because normal distributions were assumed, G*Power (Faul et al., 2007) can also be used for this situation. It yields a power of .489, so is within the confidence interval found with `pwAnova` (a large number of replications is used here to provide a more stringent test of these two approaches yielding consistent estimates).

```
reps <- 1000
dd4 <- list(function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef),
            function(n=nv,ef=efv,...) rnorm(n,ef))
(eg4a <- pwAnova(dd4,replics = reps,conmat=contr.treatment(3,3),
               pcon=.05,varyef = c(0,0,.5),varyn = 90))

##      rep propsucc ef1 ef2 ef3 n1 n2 n3
## [1,]   1    0.307   0   0 0.5 30 30 30

binconf(eg4a[1,2]*reps,reps)

##  PointEst     Lower     Upper
##     0.307 0.2791958 0.3362814

reps <- 10000
(eg4b <- pwAnova(dd4,replics = reps,conmat=contr.treatment(3,3),
               r2p=.05,varyef = c(0,0,.5),varyn = 90))

##      rep propsucc ef1 ef2 ef3 n1 n2 n3
## [1,]   1   0.4865   0   0 0.5 30 30 30

binconf(eg4b[1,2]*reps,reps)

##  PointEst     Lower     Upper
##    0.4865 0.4767108 0.4962996
```

Example 4. A Robust Oneway ANOVA

The normal distribution is rare with real data (Micceri, 1989) and an important advantage of `pwAnova` is not having any distributional requirements. Suppose a five condition study is done but the researcher expects the data to be skewed (here non-central χ^2 distributions are used with the effect size determined by the non-central parameter). R has several procedures for robust ANOVA. For consistency with the in-built 'success' definition in `pwAnova`, the `lmRob` (robust linear model) function from **robust** (Wang et al., 2014) is used. Two power analyses are shown in Figure 6. In the left panel all the data are drawn from the same central χ^2 distribution with one degree of freedom so that the null hypothesis is true. Results are declared a 'success' if any of the four contrasts is statistically significant using the p values produced by `summary.lmRob`. As shown, the Type 1 error exceeds the nominal $\alpha = .05$ and is not systematically related to the sample size (the range of the y axis is small so exaggerates the appearance of differences). In the right panel the data are drawn from different distributions based on the non-central parameter (1 for groups 2 and 3, and 2 for groups 3 and 4, so that their expected means are 2 and 3, respectively). Here the results are deemed 'successful' if the first condition differs significantly from all of the other four. This begins with a lower proportion of 'successes' compared with the left panel, but it increases with sample size.

```
reps <- 1000
par(mfrow=c(1,2))
dd5 <- list(function(n=nv,ef=efv,...) rchisq(n,1,ef),
            function(n=nv,ef=efv,...) rchisq(n,1,ef),
            function(n=nv,ef=efv,...) rchisq(n,1,ef),
            function(n=nv,ef=efv,...) rchisq(n,1,ef),
            function(n=nv,ef=efv,...) rchisq(n,1,ef))
egRobustany <- pwAnova(dd5,replics=reps,varyef=c(0,0,0,0,0),
  varyn=seq(100,1000,30),dfv=3,extrasuccess = list(function()
   any(summary(lmRob(dd[,2]~as.factor(dd[,1])))$coef[2:5,4] < .05)))
egRobustall <- pwAnova(dd5,replics=reps,varyef=c(0,1,1,2,2),
  varyn=seq(100,1000,30),extrasuccess = list(function()
   all(suppressWarnings(summary(lmRob(dd[,2]~
                      as.factor(dd[,1]))))$coef[2:5,4] < .05)))
```

Figure 3: Replicating sample data. The top row shows the Type 1 error when requiring all $k-1$ contrasts to be significant (here $k=5$) in the left panel and requiring only one to be significant in the right panel. The curves are related to sample size, which reveals a problem using sample data. In the bottom row two of the groups have μs of .1 and two have μs of .3 of a standard deviation higher than the control group.

Figure 4: Using functions rather than sample data. The top row shows differences in Type 1 errors between requiring all $k-1$ contrasts to be significant (here $k=5$) and requiring only one to be significant. As expected these are not related to sample size. In the bottom row two of the groups have μs of .3 of a standard deviation higher than the other three.

Figure 5: Exploratory data analysis of a five-group oneway ANOVA when the null hypothesis is true (left panels) and when it is false with $\mu_1 = 0; \mu_2 = \mu_3 = .1; \mu_4 = \mu_5 = .3$ in standard deviation units (right panels). The top row uses Holm's adjustment for the p values, the bottom row uses no adjustment.

Figure 6: Robust ANOVA. Comparing the control condition with each of four experimental groups. In the left panel the data are all drawn from a central χ^2 distribution with one degree of freedom (null hypothesis is true) and 'success' is defined as any of the four experimental conditions being significantly different from the control condition. In the right panel the control condition is drawn from a central χ^2_1 distribution, but the other conditions are drawn fron non-central χ^2_1 distributions with the non-central parameter being 1 for groups 2 and 3 and 2 for groups 4 and 5. For the right panel 'success' requires all four contrasts being significant.