*Journal of Statistical Software*

*November 2021, Volume 100, Issue 21.* *doi: 10.18637/jss.v100.i21*

# BayesCTDesign: An R Package for Bayesian Trial Design Using Historical Control Data

**Barry S. Eggleston**
RTI International

**Joseph G. Ibrahim**
UNC Chapel Hill

**Becky McNeil**
RTI International

**Diane Catellier**
RTI International

**Abstract**

This article introduces the R package **BayesCTDesign** for two-arm randomized Bayesian trial design using historical control data when available, and simple two-arm randomized Bayesian trial design when historical control data is not available. The package **BayesCTDesign**, which is available from the Comprehensive R Archive Network, has two simulation functions, `historic_sim()` and `simple_sim()` for studying trial characteristics under user-defined scenarios, and two methods `print()` and `plot()` for displaying summaries of the simulated trial characteristics. The package **BayesCTDesign** works with two-arm trials with equal sample sizes per arm. The package **BayesCTDesign** allows a user to study Gaussian, Poisson, Bernoulli, Weibull, lognormal, and piecewise exponential outcomes. Power for two-sided hypothesis tests at a user-defined $\alpha$ is estimated via simulation using a test within each simulation replication that involves comparing a 95% credible interval for the outcome specific treatment effect measure to the null case value. If the 95% credible interval excludes the null case value, then the null hypothesis is rejected, else the null hypothesis is accepted. In the article, the idea of including historical control data in a Bayesian analysis is reviewed, the estimation process of **BayesCTDesign** is explained, and the user interface is described. Finally, the **BayesCTDesign** is illustrated via several examples.

*Keywords*: Bayesian statistics, clinical trials, historical controls, power prior, R.

## 1. Introduction: BayesCTDesign

A controlled clinical trial is "an experiment performed on human subjects to assess the efficacy of a new treatment for some condition" (Matthews 2006, p. 1). In its most basic form, subjects

are assigned to one of two groups, a treated group which receives an experimental treatment, and a control group which receives a placebo, standard of care, or some comparator treatment (Matthews 2006, p. 1). If treatment assignment is random, then the trial is a randomized controlled clinical trial (Matthews 2006, p. 1).

When designing a clinical trial, investigators must address questions such as (Evans and Ting 2016, p. 71–86):

- What clinical question is the trial going to answer?
- What data is necessary to address the clinical question?
- What subject population is appropriate for the treatment?
- What inclusion/exclusion criteria are necessary to obtain a sample from the chosen population?
- What logistical controls will be implemented such as standardized outcome definitions, will central labs be used, will central image/evaluations be used, will standardized adjudication procedures be used, etc.?
- What is the primary outcome?
- What sample size/level of type I error control will be used?
- What statistical power is required?
- What interim analysis methods will be used?
- Will adaptive randomization be used?
- Will historical control data be used?

When designing a Bayesian clinical trial, additional questions must be addressed such as:

- What priors will be placed on model parameters?
- What criteria will be used to declare a successful trial?
- What method of interim analyses will be used (posterior vs. predictive distribution based)?
- How will historical control data be included, if it will be used?

The package **BayesCTDesign** gives the investigator a set of tools to select primary outcome type as well as address sample size and power issues within the context of a Bayesian randomized two-arm controlled trial, as well as address issues related to historical control utilization. Just because a clinical trialist is designing a Bayesian randomized clinical trial, the clinical trialist is not exempt from studying power and sample size. The clinical trialist needs to make decisions about power and sample size within the context of reasonable hypothesized treatment effects (Berry, Carlin, Lee, and Müller 2011, p. 70). Now, the process of designing a Bayesian trial involves defining priors, and this is called prior elicitation. Prior elicitation

is a very crucial part of Bayesian statistics (Ibrahim, Chen, Gwon, and Chen 2015). One approach to defining the prior is to incorporate historical data, when available. If appropriate historical data is available, it should be included in the analysis, because the result will be a more efficient trial (Viele *et al.* 2014). One method to incorporate historical data is to use a power prior, which uses actual historical data to define the prior. Using a power prior makes prior elicitation more systematic and somewhat objective in the sense that it is based on information contained in real data (Ibrahim *et al.* 2015). Inclusion of historical data using a power prior, however, requires additional design decisions to be made (Viele *et al.* 2014). Simulations can help the clinical trialist to make decisions with respect to sample size needed and with respect to incorporation of historical data.

One very important consideration is concerned with the relationship between the population from which came the historical controls and the randomized controls. Ideally, when a researcher runs a clinical trial that incorporates historical control data, the randomized controls should be from the same population as the historical controls (Ibrahim, Chen, and Chu 2012; Psioda, Soukup, and Ibrahim 2018). As such, one major question that the researcher needs to address is the appropriateness of any historical control data relative to the population from which a future randomized trial will be drawn. When proper historical control data is available it is important to consider including it in the trial design, because utilizing the information about the outcome within the historical data can result in more accurate point estimates, can improve power, and reduced type I error thus generating stronger evidence for any conclusion drawn from the trial results or reduce trial sample size while maintaining sufficient power (Viele *et al.* 2014). Yet inclusion of historical control data does have its risks. Inclusion of improper historical control data may bias the results and inflate type I error, depending on the differences between the historical and randomized controls (Viele *et al.* 2014). Similarly, inclusion of improper historical data can result in decreased power (Psioda *et al.* 2018). These risks can be mitigated by using a power prior, because the power prior can be set to include all or partial amounts of information from the historical data. How to change the amount of information encoded into the power prior will be described later in this article. Choices about inclusion of historical data and how much information to draw from historical data can be easily aided by simulations of trial scenarios where each trial scenario represents a potential real life context for the actual trial.

Using a power prior and simulations, the package **BayesCTDesign** gives the investigator some tools to determine how data from historical controls should be utilized when available and gain an understanding of the vulnerability of the final design to inclusion of improper historical control data. Via simulation, **BayesCTDesign** helps the investigator to determine trial sample size and make a decision about power prior settings before trial initiation. The package **BayesCTDesign** is a set of simulation tools that can help a clinical trialist to plan Bayesian two-arm randomized clinical trials by estimating power and other operational characteristics such as type I error, treatment effect estimate and variance, bias, and mean square error (MSE). By setting up realistic scenarios via defined design and population characteristics such as $\alpha$ level, treatment effect, sample size, and outcome, the trialist can run simulations on these scenarios to learn how the design may behave in an actual trial. **BayesCTDesign** also has the functionality for simple two-arm trials with no historical data, but its real strength is studying trial designs that incorporate historical control data.

# 2. Background

## 2.1. Package review

As of 2019-07-22, **BayesCTDesign** was only one of a few design R (R Core Team 2021) packages set up for inclusion of historical control data (Eggleston, Wilson, McNeil, Ibrahim, and Catellier 2021). A search of packages available from the Comprehensive R Archive Network (CRAN) shows several packages that can help a user to evaluate various Bayesian clinical trial designs. Some packages such as **bacistool**, **BAEssd**, **BDP2**, and **ph2bayes** can help a user to define phase II or general Bayesian trials (Chen and Lee 2020; Reyes and Ghosh 2012; Kopp-Schneider, Wiesenfarth, and Abel 2018; Kopp-Schneider, Wiesenfarth, Ruth, Edelmann, Witt, and Abel 2019; Nagashima 2018; Gsponer, Gerber, Bornkamp, Ohlssen, Vandemeulebroecke, and Schmidli 2014). Yet, these do not allow for designs that include historical controls and are limited to binary or normal outcomes. Other packages such as **BOIN**, **dfpk**, **EurosarcBayes**, **ph2bye**, **phase1RMD**, and **bcrm** can help a user design Bayesian phase I or single arm trials (Yan, Zhang, Zhou, Pan, Liu, and Yuan 2020; Toumazi, Zohar, and Ursino 2018; Dutton 2017; Zhu and Qin 2016; Yin, Du, and Mandrekar 2020; Sweeting, Mander, and Sabin 2013; Sweeting and Wheeler 2019). Being specific to early phase trials, these packages are not general design tools. Many packages are available for general clinical trial design, but these do not allow for partial inclusion of historical data information. Packages that fit into this last category are **clinfun**, **Mediana**, **rpact**, **SampleSize4ClinicalTrials**, **gsDesign**, **sp23design**, **SurvGSD**, **tsdf**, **TrialSize**, **pwr**, **pwr2**, **pwrGSD**, and **experiment** (Seshan 2018; Paux and Dmitrienko. 2018; Wassmer and Pahlke 2019; Qi 2021; Anderson 2021; Narasimhan, Shih, and He 2014; Hsu and Chen 2018; Guo and Zhong 2020; Zhang, Wu, Chow, and G.Zhang 2020; Champely 2020; Lu, Liu, and Koestler 2017; Izmirlian 2021; Imai and Jiang 2019). The **Mediana** package is very interesting and similar to **BayesCTDesign**, because it uses simulation to calculate trial characteristics. Unlike **BayesCTDesign**, **Mediana** is not designed for inclusion of historical data. We only found three packages which included historical control data in the estimation: **bayesDP**, **BACCT** and **hctrial** (Balcome, Musgrove, Haddad, and Jackson 2021; Zhang and Tang 2016; Edelmann 2018). For binary outcomes, the **BACCT** package will calculate type I error and power when historical controls are used, but it requires JAGS 4.0.0 to also be installed on the computer. The **hctrial** package can help in designing trials with historical controls, but it is designed only for binary outcomes. **bayesDP** allows for historical control data to be incorporated using a power prior where $a_0$, a parameter that determines how much of the information in the historical data is embedded in the power prior, is determined dynamically using a discount function. The **bayesDP** works with Bernoulli, Gaussian and survival outcomes, but it is set up for trial data analysis. To use **bayesDP** in design, the package results would need to be incorporated into a design focused software structure. Not only does **BayesCTDesign** have functionality to design Bayesian trials that use historical controls with binary (Bernoulli) or Gaussian outcomes, but it also has functionality for Weibull, lognormal, piecewise exponential (PWE), and Poisson outcomes.

## 2.2. Bayesian estimation

Before we get into the details of **BayesCTDesign** and its use, we will review Bayesian estimation with inclusion of historical control data by going through the computational concepts involved in Bayesian estimation with historical data and a power prior, as well as go through a simple mathematical example.

At a high level, the conceptual steps in Bayesian analysis are actually very simple. One takes prior information and combines it with information embedded in thoughtfully collected data to generate an evidence-based update about the information of interest (Bolstad 2007, p. 6). In general, one first defines a prior distribution for parameters of interest, $\boldsymbol{\theta}$, such as the probability of success in two arms of a clinical trial. This prior embodies information about what parameter values are plausible and which parameter values have a higher plausibility compared to other parameter values (Bolstad 2007, p. 6). Combining these two concepts, the prior embodies the amount of uncertainty a researcher has about the parameter. Next, one multiplies this prior by the likelihood of collected data, where the likelihood is considered a function of the parameters once the data is collected (Spiegelhalter, Abrams, and Myles 2004, p. 57):

$$f(\boldsymbol{\theta} \mid Data) \propto L(Data \mid \boldsymbol{\theta}) \times f(\boldsymbol{\theta}), \tag{1}$$

where the product of the prior and the likelihood is proportional to what is called a posterior distribution, $f(\boldsymbol{\theta} \mid Data)$. The posterior distribution for the parameters or any function proportional to it indicates which values or range of values have a non-negligible chance of being the correct value given information embedded in the collected data and the prior. Sometimes the prior in Equation 1 is assumed flat and assigned a value of 1. Although a prior of 1 is called an improper prior since it does not integrate to 1 over the real numbers, this flat prior works for many cases since the likelihood can be integrated (Berry *et al.* 2011, p. 23–24). By using such a flat improper prior, an analyst can treat the likelihood as proportional to the posterior.

Incorporating historical control data into an analysis of clinical trial data might at first seem like a difficult step; however, it is not. In the Bayesian framework, a posterior can in turn be used as a prior in future analysis (Spiegelhalter *et al.* 2004, p. 79). If you have a collection of historical control data, then that data can be used to generate a likelihood for a parameter that represents the controls. This likelihood can in turn be used to generate a posterior for the control parameter given some initial prior for the parameter that is embedded in the control likelihood. **BayesCTDesign** uses a power prior to incorporate such historical control data (Ibrahim, Chen, and Sinha 2001, p. 23–25). A power prior is simply the product of a modified likelihood and a base prior. The idea of a base prior will be described below. This product, which is a posterior with respect to the control parameter embedded in the historical control likelihood, can in turn be used as a prior for a future trial. In a power prior, some information may come from the historical controls, while other information about other parameters will be embedded in the other base priors. Historical control information about control group related parameters is embedded in the modified likelihood and the base prior, while information about other parameters only comes from information embedded in the original base prior (Ibrahim *et al.* 2015):

$$f(\boldsymbol{\theta} \mid HistControlData) \propto [L(HistControlData \mid \theta_{Control}]^{a_0}. \times f(\boldsymbol{\theta}) \tag{2}$$

A power prior is the product of a weighted likelihood calculated from historical control data and a base prior (Equation 2). The base prior is a prior on all the parameters being considered, and it embeds all information about the parameters available before current or historical data is collected. The weighting comes from raising the historical control likelihood to a power, $a_0$, which ranges from 0 to 1. If $a_0 = 0$, then the information in the historical control data is ignored. If $a_0 = 1$, then all the information in the historical control data is embedded in the resulting power prior. One way of interpreting $a_0$ is to consider that this parameter

controls the heaviness of the power prior tails (Ibrahim *et al.* 2015). Decreasing $a_0$ makes the power prior tails more heavier, which in turn makes the power prior less informative (Ibrahim *et al.* 2015). The result of this product is a function that is proportional to a posterior distribution that retains information from the base prior about all parameters but also incorporates additional information about control group related parameters that was present in the historical control data. Note: $a_0$ can be random or fixed, but using $a_0$ as random makes the computations very difficult and not necessarily better than using $a_0$ as fixed and using simulation to make a choice on the value of $a_0$ (Ibrahim *et al.* 2015; Psioda *et al.* 2018). As such, **BayesCTDesign** follows the line of reasoning that it is sufficient to consider $a_0$ as a fixed parameter, but use simulation to determine the desired value.

The final step of analyzing clinical trial data while incorporating historical control data through a power prior is to multiply the likelihood of new trial data by the power prior:

$$f(\boldsymbol{\theta} \mid Data) \propto [L(RandomizedData \mid \boldsymbol{\theta})] \times f(\boldsymbol{\theta}|HistControlData). \tag{3}$$

In Equation 3, $\boldsymbol{\theta}$ is a vector of parameters, one component is $\theta_{Control}$, and the other is $\theta_{Experimental}$. The function $f(\boldsymbol{\theta} \mid Data)$ is proportional to the posterior distribution for relevant parameters that incorporates information collected during the randomized trial, information embedded in the base prior, and information about control group parameters embedded in the historical control data.

Earlier, the idea of a "flat" or improper prior was mentioned. In **BayesCTDesign**, the base priors are always equal to improper priors with a value of 1. The choice of flat priors was made for simplicity, the authors are investigating options for inclusion of informative priors for future releases of **BayesCTDesign**. As a result of these "flat" base priors, the only parameters that have information embedded into the power priors of **BayesCTDesign** are the control group related parameters, and this information is derived from the historical control data alone, Equation 4.

$$f(\boldsymbol{\theta} \mid Data) \propto [L(RandomizedData \mid \boldsymbol{\theta})] \times [L(HistControlData \mid \theta_{Control})]^{a_0} \tag{4}$$

Note that even this historical control information is dampened if $a_0 < 1$. Given that $a_0$ is fixed for any given trial design scenario in **BayesCTDesign**, it is interesting to note that the analysis implied by a **BayesCTDesign** design, with flat base priors, is closely related to weighted maximum likelihood analysis where historical control data is given a weight of $a_0$ and new trial data are given a weight of one (Psioda and Ibrahim 2018). Finally, with these flat priors, all the information about the experimental group related parameters that is embedded in the posterior comes from the trial data alone.

## 2.3. Mathematical example

Now we will consider a simple example to illustrate the ideas behind incorporating historical control data using a power prior. As we go through this example, keep in mind that the example does not illustrate the computational process that **BayesCTDesign** uses to estimate power and other trial characteristics. **BayesCTDesign** uses simulation and the `optim()` function in R to produce a numerical based estimate of power, while this example goes through all the details of an analytic approach to calculate an estimate of posterior treatment effects. This section is intended for readers who are not familiar with Bayesian analysis using power

priors and has a purpose of illustrating the broad strokes involved in power prior use. It can be skipped by anybody familiar with power priors.

*Example setup*

Consider a scenario where:

- we have historical data from 100 controls treated with standard of care,
- among the historical controls, 65 experienced a successful response,
- the outcome is a binary (yes/no) outcome,
- we have trial data comparing a novel treatment to standard of care,
- 200 subjects are randomized into each arm of the trial,
- 150 subjects randomized to the novel therapy experienced a successful response,
- 135 subjects randomized to the control group experienced a successful response,
- we want to incorporate historical data using a power prior, $a_0 = 0.4$,
- we want to calculate $\mathsf{P}(\theta_{experimental} > \theta_{Control} \mid Data)$ for final analysis.

The scenario is based on an example described in Viele *et al.* (2014). We will use this scenario to demonstrate an analytical process for incorporating historical control data into the analysis of clinical trial data. Given that subjects are randomized into the experimental and control group component of the trial, the randomized experimental group and the randomized control group are independent. Given this independence, the posterior distribution of the parameters after trial data is analyzed can be expressed as in Equation 5.

$$\begin{aligned} f((\theta_{Experimental}, \theta_{Control}) \mid Data) \quad \propto \quad & [L(RandomizedControlData \mid \theta_{Control}] \times \\ & [L(HistControlData \mid \theta_{Control})]^{a_0} \times \\ & f(\theta_{Control}) \times \\ & [L(RandomizedExpData \mid \theta_{Experimental})] \times \\ & f(\theta_{Experimental}) \end{aligned} \tag{5}$$

In this analytical example, we will use conjugate analysis for a binary outcome. Also, as we mentioned in the previous section, the base prior, $f(\boldsymbol{\theta})$, is a prior for control group related and treatment group related parameters. For a two-arm clinical trial, $\boldsymbol{\theta}$ will contain two sets of parameters, $\theta_{control}$ and $\theta_{experimental}$, where $\theta_{control}$ and $\theta_{experimental}$ are independent, due to randomization, and equal to the probability of the outcome in the control and experimental groups, respectively. Because of this independence due to randomization, the base prior will be equal to $f(\boldsymbol{\theta}) = f(\theta_{control}) \times f(\theta_{experimental})$, and the posterior represented in Equation 5 can therefore be separated into two posteriors, one for $\theta_{control}$ and $\theta_{experimental}$. This example of posterior estimation will take advantage of this independence between treatment groups and contruct the posterior distributions for $\theta_{control}$ and $\theta_{experimental}$ separately and then multiply the two separate posteriors to construct the joint posterior of $\theta_{control}$ and $\theta_{experimental}$.

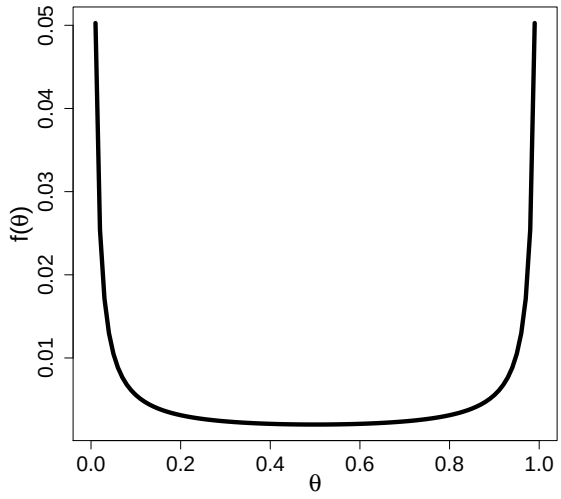


Figure 1: Base Beta prior, Beta$(0.001, 0.001)$.

Once the trial data is incorporated into the parameter estimation and individual parameter posteriors are calculated, then the joint posterior distribution of $\theta_{control}$ and $\theta_{experimental}$ will be constructed as the product of the two independent posteriors. Given this independence and the lack of any information in the power prior regarding $\theta_{experimental}$, only the base prior component $f(\theta_{control})$ will be given attention when the power prior is constructed. The other base prior component, $f(\theta_{experimental})$, will be brought into play when the posterior of $\theta_{experimental}$ is calculated.

For this example we will use Beta distributions for priors and the Bernoulli distribution to build up the likelihoods for historical controls, randomized controls, and randomized experimental group members. The formula for the Beta$(a, b)$ distribution is given by Equation 6.

$$f(\theta; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1}, \text{ where } 0 < \theta < 1,\ a > 0,\ b > 0. \tag{6}$$

### *Base prior*

Before we learn anything from the historical control data, we need to define our base prior components for thecontrol and experimental group event probabilities, $\theta_{control}$ and $\theta_{experimental}$. In this analysis, we will use a prior that is relatively flat over the $(0, 1)$ range for both $\theta_{control}$ and $\theta_{experimental}$, i.e., Beta$(0.001, 0.001)$. Figure 1 shows this base prior which is somewhat odd because it gives highest probabilities to extreme values of $\theta$ while it is rather flat between 0 and 1. This means that in the range of the most reasonable values of $\theta$ the prior is at least somewhat non-informative.

### *Modified historical control likelihood*

We now can build up the power prior, focusing on the component of the power prior that incorporates historical control information. However, before we do that, let us first take a look at the effect of $a_0$ on the historical control likelihood. Of the 100 historical controls, 65 successful responses were observed, so the likelihood for the historical control data has a

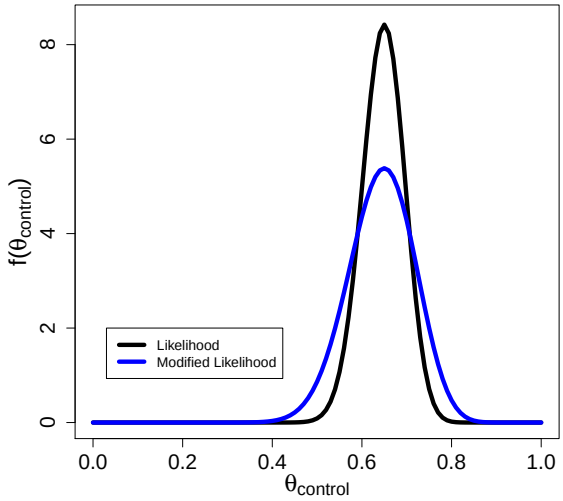


Figure 2: Historical control likelihood and modified historical control likelihood.

shape proportional to a Beta$(66, 36)$ distribution defined in Equation 7:

$$\begin{aligned} L(\theta_{control}) &= \theta_{control}^{65}(1-\theta_{control})^{(100-65)} \\ &\propto \frac{\Gamma(66+36)}{\Gamma(66)\Gamma(36)}\theta_{control}^{((65+1)-1)}(1-\theta_{control})^{((100-65+1)-1)}. \end{aligned} \tag{7}$$

We will use this likelihood as part of the power prior by raising the likelihood to the power of $a_0$: $L(\theta_{control})^{a_0}$. For a likelihood that is the product of a set of Bernoulli likelihoods as in Equation 7, apart from the proportionality constant, the modified historical control likelihood is obtained by simply multiplying the success and failure values by $a_0$. For the historical control data, the down weighted number of successes is $0.4 \cdot 65 = 26$, while the down weighted number of failures is $0.4 \cdot 35 = 14$. Notice how the use of $a_0$ down weights the value of information in the historical control likelihood. The original historical control likelihood was based on data from 100 subjects, but raising the historical control likelihood to the $a_0$ power reduces the effective sample size of the historical control data. The modified historical control likelihood contains information from only 40 subjects. This reduction in effective sample size increases the uncertainty represented by the historical control data. Down weighting the historical control likelihood by $a_0 = 0.4$, affects the graph of the historical control likelihood in an interesting way. The down weighted likelihood is given in Equations 8 and 9 and is visualized in Figure 2. Figure 2 shows both the likelihood of the historical control data and the resulting modified likelihood created by down weighting the historical control likelihood using $a_0 = 0.4$.

$$\begin{aligned} L(\theta_{control})^{0.4} &= [\theta_{control}^{65}(1-\theta_{control})^{(100-65)}]^{0.4} \\ &= \theta_{control}^{0.4\cdot 65}(1-\theta_{control})^{0.4\cdot(100-65)} \\ &= \theta_{control}^{26}(1-\theta_{control})^{14}. \end{aligned} \tag{8}$$

$$L(\theta_{control})^{0.4} \propto \frac{\Gamma(27+15)}{\Gamma(27)\Gamma(15)}\theta_{control}^{(27-1)}(1-\theta_{control})^{(15-1)}. \tag{9}$$

Notice how the historical likelihood has been spread out by raising it to the $a_0$ power of 0.4,

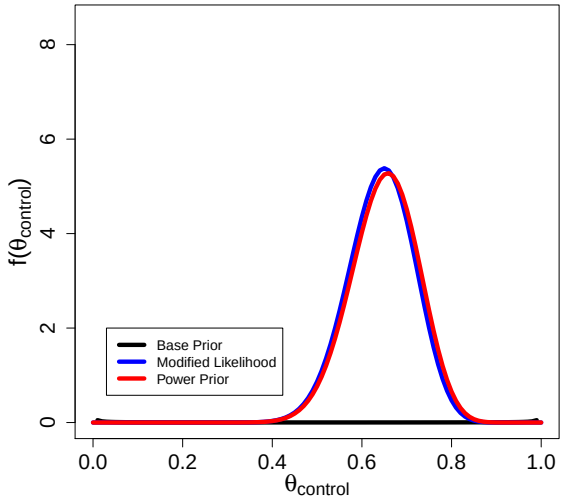


Figure 3: Power prior for randomized controls.

resulting in more values of $\theta_{control}$ with non-negligible likelihood values. Down weighting the historical control data simply made the information about the true success probability among controls, $\theta_{control}$, a little more uncertain. One way of interpreting this procedure is to say that we believe the historical data has good information about the central location of the true value for $\theta_{control}$ among controls receiving standard of care, but for analysis of the trial data we assume it is overly optimistic in its precision of the success probability among controls.

*Power prior*

At this point, we are ready to calculate the component of the power prior that concerns randomized controls and the historical control data down weighted by 0.4, by multiplying the component of the base prior related to the controls and the modified historical control likelihood. The component of the power prior for the randomized controls is given in Equation 10:

$$\begin{aligned} f(\theta_{control} \mid HistData) &= f(\theta_{control}) \times L(\theta_{control})^{0.4} \\ &\propto [\theta_{control}^{(0.001-1)}(1-\theta_{control})^{(0.001-1)}][\theta_{control}^{(27-1)}(1-\theta_{control})^{(15-1)}] \\ &\propto \frac{\Gamma(26.001+14.001)}{\Gamma(26.001)\Gamma(14.001)}\theta_{control}^{(26.001-1)}(1-\theta_{control})^{(14.001-1)}. \end{aligned} \tag{10}$$

Multiplying the base prior $\text{Beta}(0.001, 0.001)$ by the power prior using $a_0 = 0.4$ generates a prior for randomized controls that is proportional to a $\text{Beta}(26.001, 14.001)$. Figure 3 shows this power prior, along with the base prior and the modified historical control likelihood. This power prior embeds information about the control group parameter, $\theta_{control}$, that was contained in the historical control likelihood and the original base prior. It is a posterior distribution given the historical control data, but we will use it as a prior for the analysis of randomized control data.

*Posterior construction*

In the trial, we have 200 subjects in each arm, with 150 successes in the experimental arm and 135 successes in the control arm. Figure 4 shows us what the likelihood for the randomized control group looks like relative to the power prior.

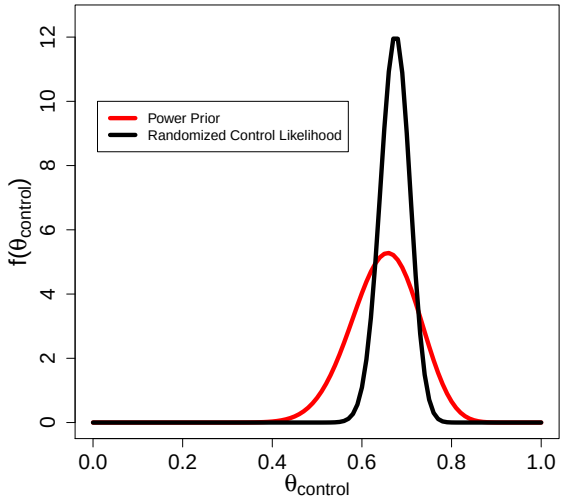


Figure 4: Power prior and likelihood for randomized controls.

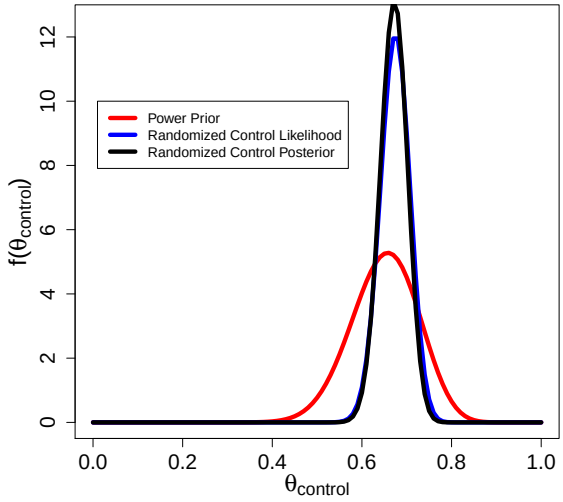


Figure 5: Randomized control group posterior.

By multiplying the likelihood of the randomized control data and the power prior for randomized controls, we get the posterior for the control group regarding the probability of success, $\theta_{control}$. This posterior is a Beta$(26.001 + 135, 14.001 + 65)$. Figure 5 shows this posterior for the randomized control group along with the randomized control likelihood and the power prior. In Figure 5 we see very good alignment between the power prior and the randomized control likelihood, so the posterior is very similar to the randomized control likelihood, giving slightly more precise information about the probability of success among controls. Similarly, the likelihood for the randomized experimental group is calculated by multiplying the base prior, Beta$(0.001, 0.001)$, times the likelihood for the experimental group data, which is proportional to a Beta$(151, 51)$. The posterior is proportional to a Beta$(0.001 + 150, 0.001 + 50)$ distribution. Figure 6 shows both the posterior for the randomized control group and the posterior for the randomized experimental group. Since the experimental group and the control are independent, we can look at the joint posterior distribution of the success probability

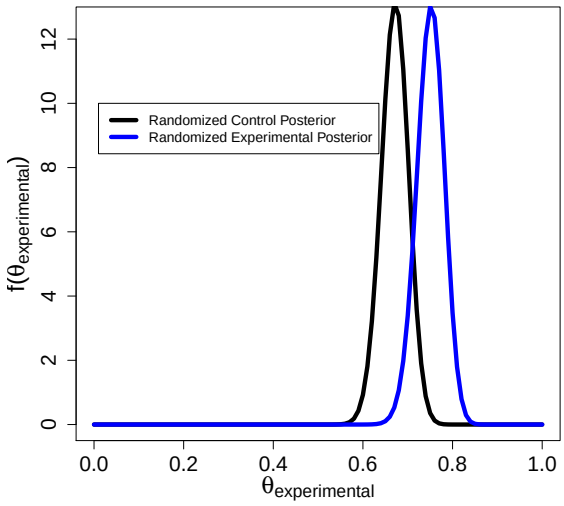


Figure 6: Randomized control and experimental group posteriors.

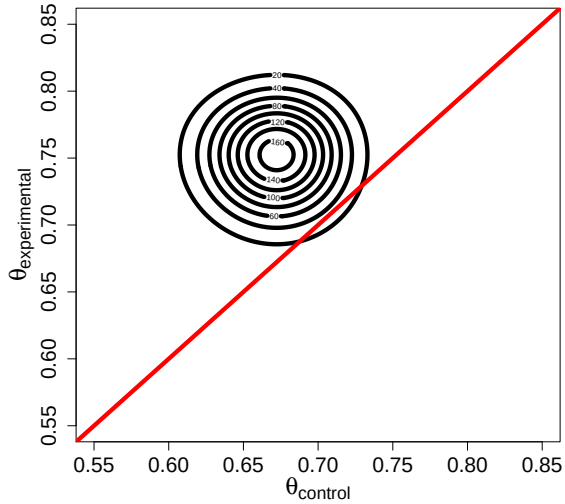


Figure 7: Contour plot of posterior joint distribution for randomized experimental and control groups.

in each group by using $\theta_{control}$ for the success probability in the controls and $\theta_{experimental}$ for the success probability in the experimental group and then simply multiplying the individual group posteriors. The joint posterior distribution is given in Equation 11, and the contour plot of the joint distribution is given in Figure 7.

$$\begin{aligned} f(\theta_{control}, \theta_{experimental} \mid Data) \quad = \quad & \frac{\Gamma(161.001 + 79.001)}{\Gamma(161.001)\Gamma(79.001)} \times \\ & [\theta_{control}^{(161.001-1)}(1-\theta_{control})^{(79.001-1)}] \times \\ & \frac{\Gamma(150.001 + 50.001)}{\Gamma(150.001)\Gamma(50.001)} \times \\ & [\theta_{experimental}^{(150.001-1)}(1-\theta_{experimental})^{(50.001-1)}]. \end{aligned} \tag{11}$$

Figure 7 shows that the posterior mean for the experimental treatment success probability is around 0.75, while the posterior mean for the control success probability is around 0.67. What is $\mathsf{P}(\theta_{experimental} > \theta_{control} \mid Data)$? We apply double integration to Equation 11 to calculate this probability, which results in 0.97. The double integration setup is given in Equation 12.

$$Pr(\theta_{experimental} > \theta_{control} \mid Data) = \int_{\theta_{control}=0}^{1} \int_{\theta_{experimental}=\theta_{control}}^{1} f(\theta_{control}, \theta_{experimental} \mid Data) d\theta_{experimental} d\theta_{control} \approx 0.97. \quad (12)$$

We have a very high posterior probability (of about 0.97) of $\theta_{experimental} > \theta_{control}$. If we wanted to, we could continue to use double integration to estimate the posterior difference in success probabilities, $\theta_{experimental} - \theta_{control}$, and calculate a 95% credible interval for this difference, see Equations 13 through 15.

$$E(\theta_{experimental} - \theta_{control}) = \int_{\theta_{control}=0}^{1} \int_{\theta_{experimental}=0}^{1} (\theta_{experimental} - \theta_{control}) \times f(\theta_{control}, \theta_{experimental} \mid Data) d\theta_{experimental} d\theta_{control} \approx 0.079 \quad (13)$$

$$E(\theta_{experimental} - \theta_{control})^2 = \int_{\theta_{control}=0}^{1} \int_{\theta_{experimental}=0}^{1} (\theta_{experimental} - \theta_{control})^2 \times f(\theta_{control}, \theta_{experimental} \mid Data) d\theta_{experimental} d\theta_{control} \approx 0.0081 \quad (14)$$

$$\sigma_{(\theta_{experimental}, \theta_{control})} = \sqrt{(E(\theta_{experimental} - \theta_{control})^2 - (E(\theta_{experimental} - \theta_{control}))^2} \approx 0.043 \quad (15)$$

In summary, the posterior probability that the difference in success proportion is greater than 0 is 0.97. The posterior estimate of the difference in success probability is 0.079, with a large sample 95% credible interval of $(-0.01, 0.16)$. Yet, not all Bayesian analyses are this simple, and as analysts we do not want to work with double integrals unless necessary. Even worse, as the number of parameters increases, the analytical approach via integrals become intractable. Markov chain Monte Carlo (MCMC) techniques are available for accurate estimation of parameters and thus available for use in simulation studies of power; however powering Bayesian trial via MCMC and simulation can take a very long time. **BayesCTDesign** was created to assist clinical trialists in studying the consequences of including historical controls into the study through simulation and numerical approximation, and reduce the amount of time it takes to get results via simulation relative to MCMC utilization.

## 3. BayesCTDesign overview

**BayesCTDesign** is a simulation based package that helps clinical trialists to estimate power and make sample size determinations about Bayesian two-arm trial designs that may or may not include historical control data, partial or in full. The package will allow a user to define

$\alpha$, but at present only two-sided hypothesis tests can be considered. Also, only equal sized trial arms can be considered. The package will allow a clinical trialist to consider Gaussian, Poisson, Bernoulli, Weibull, lognormal, and piecewise exponential outcomes. As noted earlier, flat priors are used for the base priors, so informative prior knowledge is embedded in the likelihood of the historical controls, and this information is potentially down weighted by the power prior parameter, $a_0$. For a given simulation set-up, the package will simulate a user specified number of trials, which in turn can be summarized to estimate trial operational characteristics. Within each trial replication the package will estimate a Gaussian posterior for a function of the treatment effect. The treatment effect that is implemented depends on the outcome:

- Gaussian: Estimated effect is a difference in two means.
- Bernoulli: Estimated effect is an odds ratio (experimental over control).
- Poisson: Estimated effect is a mean ratio (experimental over control).
- Weibull: Estimated effect is a hazard ratio (experimental over control).
- Lognormal: Estimated effect is a mean ratio (experimental over control).
- Piecewise exponential: Estimated effect is a hazard ratio (experimental over control).

The function of the treatment effect that has a posterior generated also depends on the outcome. Because log transformation of ratios can improve the Gaussian approximation of a posterior, actual posteriors for ratio effects are on the log scale:

- Gaussian: Estimated posterior is for the difference in two means.
- Bernoulli: Estimated posterior is for the log odds ratio (experimental over control).
- Poisson: Estimated posterior is for the log mean ratio (experimental over control).
- Weibull: Estimated posterior is for the log hazard ratio (experimental over control).
- Lognormal: Estimated posterior is for the log mean ratio (experimental over control).
- Piecewise exponential: Estimated posterior is for the log hazard ratio (experimental over control).

The package uses the Bayesian central limit theorem (BCLT) to generate trial characteristics such as power and type I error (Berry *et al.* 2011, p. 28). As noted earlier, the appropriateness of applying the BCLT is enhanced by the using the log transformations in the estimation when treatment effect is a ratio.

The package uses the BCLT to estimate trial effect posteriors, which in turn are used to make decisions about rejecting/accepting a null hypothesis. The BCLT states that if:

1. the data collected from each subject is independent from data collected from other subjects,
2. the data collected are generated from the same distribution,

3. the priors have positive value for all real numbers and are twice differentiable near the posterior mode,

4. the joint distribution of the data is positive and twice differentiable near the posterior mode,

then under suitable regularity conditions, the posterior distribution for large sample sizes can be approximated by a normal distribution (perhaps multivariate) with mean equal to the posterior mode and the covariance matrix equal to negative one times the inverse Hessian matrix of the log posterior evaluated at the log posterior mode, which will equal the posterior mode.

For posterior mode and hessian matrix estimation, **BayesCTDesign** uses the R `optim()` function, using the Nelder-Mead optimization method to estimate the posterior mode and hessian matrix at the posterior mode. Once the covariance matrix is determined from the hessian matrix, posterior standard deviations are calculated from the diagonals of the covariance matrix. The use of log transformations on treatment effect estimates when the treatment effects are in the form of ratios increases the quality of the Gaussian approximation when applying the BCLT. The concern of how large sample size needs to be in order for the approximation to be acceptable will be considered in Section 3.2.

If the treatment effect is on the log scale, then a large sample $100 \cdot (1 - \alpha/2)\%$ credible interval is calculated by using the posterior mode as the mean and the posterior standard deviation as the measure of variability and the result is exponentiated to calculate a large sample $100 \cdot (1 - \alpha/2)\%$ credible interval for the treatment effect. If the treatment effect is not estimated on the log scale, the credible interval is calculated but not exponentiated. Within a trial replication, a decision about rejecting the null hypothesis of no treatment effect is made by determining if the credible interval excludes the null value, null value for ratios is 1 and null value for differences is 0. The package has two primary simulation functions, `simple_sim()` and `historic_sim()`. The function `simple_sim()` is for studying simple two-arm clinical trials where no historical data is being included. The function `historic_sim()` is for studying two-arm clinical trials where historical data is being included. In addition to these two primary simulation functions, the package has `print()` and `plot()` methods. These functions will be described in more detail in Section 4.

### 3.1. Simulation process

The process of simulation that is used in **BayesCTDesign** can be explained by focusing on one trial replication that incorporates historical control data and uses a Bernoulli outcome. First, the historical control data is analyzed to estimate the base success probability, $\theta_{histc}$, for controls. With $\theta_{histc}$ defined the user can decide if the randomized controls and the historical controls will differ in success probabilities or not. If one assumes no difference between historical and randomized controls, then $\theta_{randc}$, the success probability among randomized controls, is set equal to $\theta_{histc}$. If one needs to assume a difference exists between historical and randomized controls, then $\theta_{randc}$ is calculated using a user-defined odds ratios, $OR_c$, that defines the difference between the two control groups:

$$\theta_{randc} = \frac{OR_c \times \frac{\theta_{histc}}{1-\theta_{histc}}}{1 + OR_c \times \frac{\theta_{histc}}{1-\theta_{histc}}}, \tag{16}$$

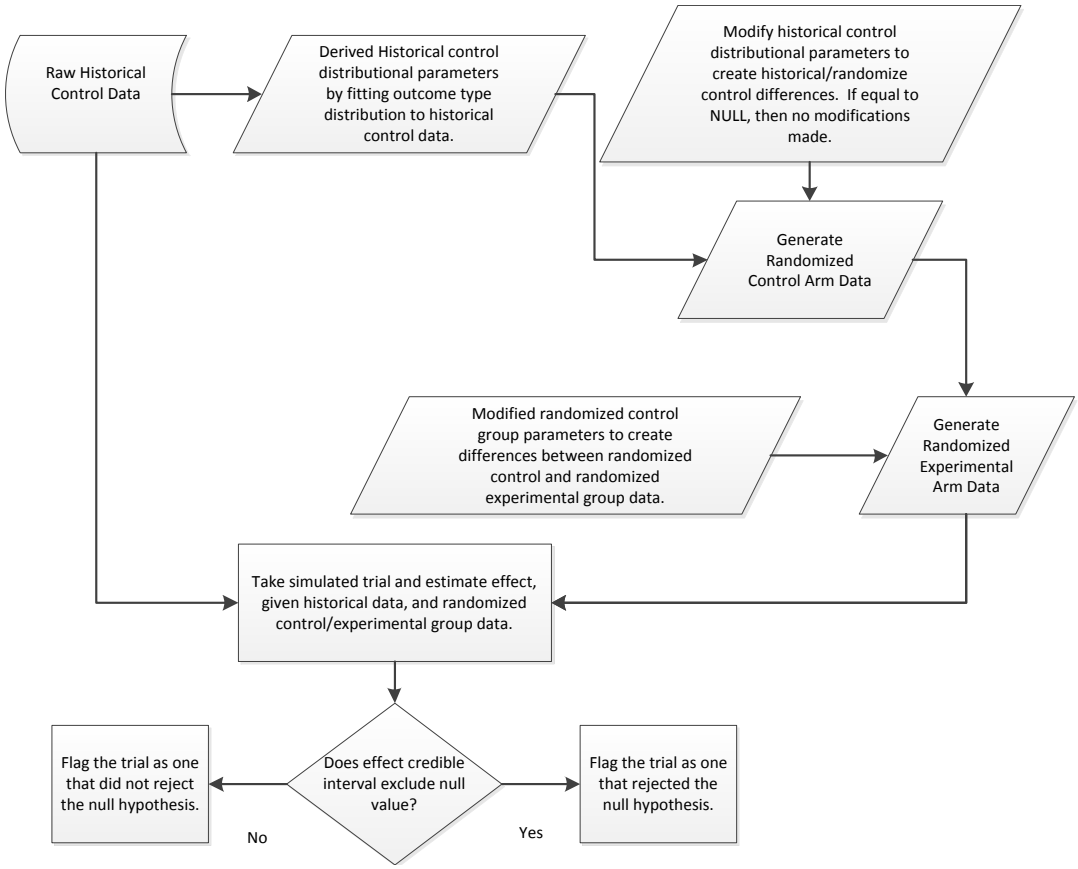


Figure 8: Basic data generation and trial simulation for `historic_sim()`.

where $OR_c$ is the odds ratio between randomized and historical control success probabilities, randomized over historic. With the two control group success probabilities $\theta_{histc}$ and $\theta_{randc}$ defined, the success probability for the randomized experimental treatment group, $\theta_{randexp}$ is calculated. In **BayesCTDesign**, the value of $\theta_{randexp}$ is always set relative to the value of $\theta_{randc}$. The value of $\theta_{randexp}$ is determined using a user-defined treatment odds ratio, $OR_{trt}$, see Equation 17.

$$\theta_{randexp} = \frac{OR_{trt} \times \frac{\theta_{randc}}{1-\theta_{randc}}}{1 + OR_{trt} \times \frac{\theta_{randc}}{1-\theta_{randc}}}. \tag{17}$$

With the parameters defined for all three groups, data is simulated for the randomized control group and the randomized experimental group. Finally, the historical control data and the simulated trial data are modeled to estimate the posterior of $\log(OR_{trt})$ using `optim()` and then a credible interval is calculated for $\log(OR_{trt})$. Finally, the credible interval for $\log(OR_{trt})$ is exponentiated to determine if the credible interval for $OR_{trt}$ excludes 1. To estimate trial characteristics, the above process is repeated many times, the characteristics are recorded each time, and averages across all trial replicates are calculated.

Figure 8 contains a flow chart of the basic data generating and trial simulation process for the most complicated simulation function, `historic_sim()`. The process for `simple_sim()` is very similar, except historical data is not involved, so randomized control arm data are generated based on distributional parameters given by the user.

### 3.2. Time savings and accuracy

The goal of **BayesCTDesign** is to reduce the time it takes to run simulations that quantify trial characteristics relative to MCMC. One can study complex Bayesian trial characteristics using simulations and MCMC for estimation, but the time it takes to run the number of replications needed to get good estimates of the posteriors within each trial replication is very time consuming. Using MCMC techniques in the simulations for power calculations involves simulating the trial by creating a hypothetical dataset of trial data and then analyzing the hypothetical data using MCMC techniques. The large amount of time taken to study trial characteristics using MCMC is mostly taken up by the time it takes to create the MCMC chains that are used to approximate the treatment effect posterior. **BayesCTDesign** reduces the time necessary for running the replications by using the BCLT and numerical optimization to approximate the posterior with a Gaussian distribution centered on the posterior mode. Since **BayesCTDesign** still uses simulations, a very large and complex simulation can still take a long time, but as the following discussion will show, the time will always be a fraction of the time necessary when using MCMC.

Using the BCLT saves a lot of time relative to MCMC use, but such time savings are only good if the simplification does not lead to inaccuracies in posterior estimation. In this section we will look at a set of simulations that demonstrate not only the time savings of using the BCLT for estimation, but also that its use does not impose inaccuracies relative to an MCMC approach when sample sizes are reasonable large. Note, the MCMC approach used for these comparisons involved the same simulation process as used by **BayesCTDesign** except MCMC estimation results were used to construct the credible interval for treatment effect estimate.

In all simulations except for the piecewise exponential, flat $N(\mu = 0, \sigma = 100)$ priors were used for base priors of model parameters in the MCMC models. In the MCMC model for the piecewise exponential the treatment effect base prior was a flat $N(\mu = 0, \sigma = 100)$; however, a multivariate gamma prior was used for the set of hazards defined for the time intervals. Each MCMC call used the R package **rjags** and the **JAGS** software to generate the MCMC chains (Plummer 2003). For posterior approximation using MCMC, 2 chains were created, using 1000 adaptations, 1000 burn-ins, and then 10000 samples were collected. No thinning was used on the final chains. The total estimation approaches were compared using a Windows machine with an i7-3770 process runing at 40GHz and 8 GB of RAM. In these time and accuracy assessment simulations, 2 cores were used by utilizing the R packages **doParallel** and **foreach** (Wallig, Corporation, Weston, and Tenenbaum 2020a, Wallig, Microsoft, and Weston 2020b, see also Kane, Emerson, and Weston 2013).

Table 1 contains the time and accuracy assessment results for Bernoulli, Gaussian, Poisson, Weibull, lognormal, and piecewise exponential outcomes respectively. The following is reported in the table: outcome type, true effect value, estimation mode (MCMC or **BayesCTDesign**), posterior treatment effect estimates on the log scale if applicable, posterior standard deviation on the log scale if applicable, posterior treatment effect estimate on the untransformed scale, posterior standard deviation on untransformed scale if applicable, power, single trial replication run time (in seconds), ratio of MCMC run time to **BayesCTDesign** run time, total time for 100 trial replications (in minutes). In the table, the treatment effect estimates (on log scale and untransformed scale) are averages based on 100 trial replications. In all simulations, the outcome distributions for the randomized controls was the same as the outcome distributions for the historical controls.

| Outcome | True effect | Mode | Est. log effect | sd (est. log effect) | Est. effect | sd (est. effect) | Power | Run time (sec/iter) | Run time ratio (MCMC/BayesCT) | Run time (Min. per 1000) |
|---|---|---|---|---|---|---|---|---|---|---|
| Gaussian | 1.1 | MCMC | | | 1.108 | 0.368 | 0.87 | 17.114 | 2226.812 | 285.227 |
| | | BayesCT | | | 1.097 | 0.365 | 0.86 | 0.008 | | 0.128 |
| Binomial | 0.45 ($OR_{trt}$) | MCMC | −0.786 | 0.290 | 0.486 | | 0.88 | 3.394 | 514.520 | 56.561 |
| | | BayesCT | −0.777 | 0.288 | 0.470 | | 0.87 | 0.007 | | 0.110 |
| Poisson | 0.6 ($MR_{trt}$) | MCMC | −0.519 | 0.172 | 0.61 | | 0.91 | 3.448 | 540.734 | 57.461 |
| | | BayesCT | −0.512 | 0.172 | 0.606 | | 0.91 | 0.006 | | 0.106 |
| Weibull | 0.6 ($HR_{trt}$) | MCMC | −0.523 | 0.188 | 0.61 | | 0.91 | 17.480 | 482.519 | 291.341 |
| | | BayesCT | −0.515 | 0.187 | 0.604 | | 0.91 | 0.036 | | 0.604 |
| Lognormal | 0.6 ($MR_{trt}$) | MCMC | −0.508 | 0.173 | 0.62 | | 0.88 | 25.915 | 1619.706 | 431.921 |
| | | BayesCT | −0.505 | 0.170 | 0.612 | | 0.88 | 0.016 | | 0.267 |
| PWE | 0.6 ($HR_{trt}$) | MCMC | −0.503 | 0.192 | 0.624 | | 0.83 | 38.723 | 51.381 | 645.382 |
| | | BayesCT | −0.494 | 0.191 | 0.618 | | 0.80 | 0.754 | | 12.561 |

Table 1: Time and Accuracy trial Results.

For Gaussian outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The mean in the historical controls was 25.93, standard deviation was 2.60, and the treatment effect was set to a mean difference of 1.1. Table 1 shows that the posterior treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Using **BayesCTDesign**, 100 trial replications were ran and summarized in about 0.13 minutes of clock time, while the MCMC approach took about 285 minutes.

For Bernoulli outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The true proportion with the outcome among the historical controls was 0.57, and the true treatment effect was 0.45. Table 1 shows that the posterior log treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Even though the estimates are about equal, using **BayesCTDesign** to run 100 trial replications and summarize the results took only about 0.11 minutes of clock time, while the MCMC approach took up to 57 minutes.

For Poisson outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The mean in the historical controls was 0.95, and the treatment effect was set to a mean ratio of 0.6. Table 1 shows that the posterior log treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Using **BayesCTDesign**, 100 trial replications were ran and summarized in about 0.11 minutes of clock time, while the MCMC approach took about 57 minutes.

For Weibull outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The median event-time among the historical controls was 2.5 years, and the treatment effect was set to a hazard ratio of 0.6. Weibull parameters for the historical controls were `scale = 2.814651` and `shape = 3.091710` (using `rweibull()` parameterization). Among historical controls and randomized trial data, the event times were right censored at 3 years. Table 1 shows that the posterior log treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Using **BayesCTDesign**, 100 trial replications were ran and summarized in about 0.60 minutes of clock time, while the MCMC approach took up to 291 minutes.

For lognormal outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The median event-time among the historical controls was 2.54 years, and the treatment effect was set to a mean ratio of 0.6. Lognormal parameters for the historical controls were `meanlog = 0.9332408` and `sdlog = 1.147586` (using `rlnorm()` parameterization). Among historical controls and randomized trial data, the event times were right censored at 3 years. Table 1 shows that the posterior log treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Using **BayesCTDesign**, 100 trial replications were ran and summarized in about 0.27 minutes of clock time, while the MCMC approach took about 432 minutes.

Finally, for piecewise exponential outcomes, a trial with 80 subjects per arm and 60 historical groups were simulated. The time intervals among historical controls were created by using cutpoints at 0.3, 0.9, 1.5, 2.1, and 2.4 years. The corresponding interval hazards were 0.1707802, 0.3213363, 0.5089973, 0.4216200, 0.2620553, and 0.3884450. The median event-time among the historical controls was 1.92 years, and the treatment effect was set to a hazard ratio of 0.6. Among historical controls and randomized trial data, the event times were right censored at 3 years. The `rpch()` function in the **eha** package is used to generate draws from

a piecewise exponential distribution (Broström 2012). Table 1 shows that the posterior log treatment effect, standard deviation, and power estimates from the MCMC and **BayesCTDesign** approaches are within 0.02 of each other. Using **BayesCTDesign**, 100 trial replications were ran and summarized in about 12.6 minutes of clock time, while the MCMC approach took up to 645 minutes. The piecewise exponential simulation process is slower than the other processes, because extra data processing is needed to determine if each combination of time interval and control/experimental treatment group combination has at least 5 events. This extra processing is a conservative step to ensure evaluable likelihoods across many simulated trials.

Although the timing differences are dependent on the particular computer used and on the MCMC model and MCMC generating software used, Table 1 shows that substantial time savings can be obtained using **BayesCTDesign** relative to MCMC, reducing run time by at least 97%. Table 1 only briefly illustrates the time savings and accuracy assessments of **BayesCTDesign** relative to MCMC. In addition to Table 1, the authors constructed several tables to study the time savings and accuracy of each outcome in a more thorough matter than could be reported in this article. These additional tables along with the underlying code are provided as supplementary files.

## 4. Package user interface overview

The package user interface is an R command that can be called from the R command line or coded into an R script. The package has four primary functions that are accessible by the user:

- `simple_sim()`: A command line interface for simulating two-arm Bayesian clinical trials that do not incorporate historical data.
- `historic_sim()`: A command line interface for simulating two-arm Bayesian clinical trials that incorporate historical data.
- `print()`: A command line interface for printing out tables from the object created by `simple_sim()` or `historic_sim()`.
- `plot()`: A command line interface for plotting and/or smoothing, uses `loess()`, tabulated results from the object created by `simple_sim()` or `historic_sim()`.

The functions `simple_sim()` and `historic_sim()` are called first to generate a set of simulated trial results. The function `simple_sim()` has 13 parameters. A call to `simple_sim()` will have the following form:

```
simple_sim(trial_reps = 100, outcome_type = "weibull",
  subj_per_arm = c(50, 100, 150, 200, 250), effect_vals = c(0.6, 1, 1.4),
  control_parms = NULL, time_vec = NULL, censor_value = NULL,
  alpha = 0.05, get_var = FALSE, get_bias = FALSE, get_mse = FALSE,
  seedval = NULL, quietly = TRUE)
```

The `trial_reps` parameter determines how many trials to simulate. The `outcome_type` parameter determines what outcome type will be studied. Possible values for `outcome_type` are:

"weibull", "lognormal", "pwe" (for piecewise exponential survival outcome), "gaussian", "bernoulli" (for binary outcome), "poisson".

The parameters subj_per_arm and effect_vals are vectors of sample sizes and treatment effect values respectively to explore via simulation. The treatment effect value that is used depends on the outcome. For each outcome type, the following effects are used:

- Gaussian: Estimated positive difference in two means.
- Bernoulli: Estimated odds ratio (experimental over control).
- Poisson: Estimated mean ratio (experimental over control).
- Weibull: Estimated hazard ratio (experimental over control).
- Lognormal: Estimated mean ratio (experimental over control).
- Piecewise exponential: Estimated hazard ratio (experimental over control).

For binary outcomes, the current version of **BayesCTDesign** only allows the odds ratio to be studied as a treatment effect, an estimated difference in proportions is being considered for a future release of **BayesCTDesign**.

The control_parms parameter is a vector of distributional parameters that defines the outcome distribution of the randomized controls. The parameters listed in control_parms must be sufficient to define a distribution of type 'outcome_type'. For each outcome type, the following information is required for control_parms:

- Gaussian: (mean, sd), where mean is the mean parameter for the control group used in a call to rnorm(), and sd is the common standard deviation parameter for both groups used in a call to rnorm().
- Bernoulli: (prob), where prob is the event probability for the control group used in a call to rbinom().
- Poisson: (lambda), where lambda is the lambda parameter for the control group used in a call to rpois() and is equal to the mean of a Poisson distribution.
- Weibull: (scale, shape), where scale is the scale parameter for the control group used in a call to rweibull(), and shape is the shape parameter for both groups used in a call to rweibull().
- Lognormal: (meanlog, sdlog), where meanlog is the mean parameter for the control group used in a call to rlnorm(), and sdlog is the sd parameter for both groups used in a call to rlnorm().
- Piecewise exponential: A vector of lambdas used in a call to eha::rpch(), where each lambda is a hazard (numerical value representing the failure rate) for an interval defined by the time_vec parameter.

The time_vec parameter is a vector of time cut-offs and is used only for the piecewise exponential. If time_vec has 4 values $t_1$, $t_2$, $t_3$, $t_4$, then five intervals are created, $(0, t_1)$, $(t_1, t_2)$,

$(t_2, t_3)$, $(t_3, t_4)$, and $(t_4, +\infty)$. In such a case, `control_parms` should have 5 values, a hazard value for each interval. Between the values of `time_vec`, the hazard is assumed constant. If the `outcome_type` is a survival outcome and the user wants to study a trial where event times are right censored at some value, then `censor_value` is used to define when simulated event times are right censored. The parameter `alpha` is used to define the test wise type I error rate. All simulation runs using `simple_sim()` will return results for power and effect estimation; however, the user has to indicate whether or not variance, bias, and MSE results are to be returned as well. The parameters `get_var`, `get_bias` and `get_mse` are `TRUE/FALSE` indicators that determine if variance, bias, and MSE results are returned in addition to estimated power and effect estimate results. The parameter `seedval` allows the user to set the seed necessary for repeatable results. When `simple_sim()` runs it can print a short list of numbers indicating which trial scenario is being simulated. This simple report will help the user to know the program is running and how far along it is. When **BayesCTDesign** is run in a notebook or a log file is generated from the output, this simple report can create a very large amount of unnecessary information in the results file. The parameter `quietly` can be used to turn this report off.

The function `simple_sim()` will take each combination of `subj_per_arm` and `effect_vals`, run `trial_reps` replicates of the trial, then calculate the number of replicates where the null hypothesis was rejected, calculate the average effect, and if requested by the user, estimate values of variance, bias, and MSE on appropriate scales. Once done, `simple_sim()` returns a list that is an S3 object of class '`bayes_ctd_array`'. A '`bayes_ctd_array`' class object has 6 elements:

- a list containing simulation results (called `data`),
- the `subj_per_arm` vector,
- the `effect_vals` vector,
- the `a0_vals` vector (which is a single value of 1 for `simple_sim()`),
- the `rand_control_diff` vector (which is a single value of 1 for `simple_sim()`),
- an indicator of whether `simple_sim()` was used.

Because `simple_sim()` returns the same type of object as `historic_sim()`, each element of `data` returned from `simple_sim()` is a four-dimensional array. The first dimension contains information related to levels of `subj_per_arm` studied in the simulation, and the third dimension contains information related to levels of `effect_vals` studied in the simulation. Note, however, that the second and fourth dimensions are not relevant for results from `simple_sim()`. The size of the first and third dimensions are determined by the vector lengths of parameters `subj_per_arm` and `effect_vals` as requested by the user. At a minimum, at least one of `subj_per_arm` or `effect_vals` must contain at least 2 values. The simulation results `data` will always contain two elements: An array of power results (`power`) and an array of estimation results (`est`). In addition to `power` and `est`, data may also contain elements `var`, `bias`, or `mse`, depending on the values of `get_var`, `get_bias`, and `get_mse`. The values returned in `est` are in the form of hazard ratios, mean ratios, odds ratios, or mean differences depending on the value of `outcome_type` as previously described. The values returned in `bias`, `var`, and `mse` are on the scale of the values returned in `est`.

Results from the simulation contained in the `bayes_ctd_array` object can be printed or plotted using the `print_table()` and `plot_table()` methods. The results can also be accessed using basic list element identification and array slicing. For example, to get the power results from a simulation, one could use the code `bayes_ctd_arrayname$data$power`, where `bayes_ctd_arrayname` is replaced with the name of the variable containing the `bayes_ctd_array` object. Even though the arrays returned in the `data` element are 4-dimensional arrays, for `simple_sim()` simulations the power and estimate results really only occupy a single 2-dimensional table. To print this 2-dimensional table without using the `print_table()` method, one could use the code `bayes_ctd_arrayname$data$power[ , 1, , 1]`, where
`bayes_ctd_arrayname` is replaced with the name of the variable containing the `bayes_ctd_array` object.

The function, `historic_sim()` has 15 parameters. A call to `historic_sim()` will have the following form:

```
historic_sim(trial_reps = 100, outcome_type = "weibull",
  subj_per_arm = c(50, 100, 150, 200, 250), a0_vals = c(0, 0.33, 0.67, 1),
  effect_vals = c(0.6, 1, 1.4), rand_control_diff = c(0.8, 1, 1.2),
  hist_control_data = NULL, time_vec = NULL, censor_value = NULL,
  alpha = 0.05, get_var = FALSE, get_bias = FALSE, get_mse = FALSE,
  seedval = NULL, quietly = TRUE)
```

The function `historic_sim()` shares the following parameters with `simple_sim()`: `trial_reps`, `outcome_type`, `subj_per_arm`, `effect_vals`, `censor_value`, `alpha`, `get_var`, `get_bias`, `get_mse`, `seedval`, `quietly`. For a description of these shared parameters, see the above description of the `simple_sim()` user interface. Parameters that are unique to `historic_sim()` are `a0_vals`, `rand_control_diff`, and `hist_control_data`. The `hist_control_data` parameter needs to be set equal to the name of the historical control dataset.

For survival outcomes, the historical control dataset must have 4 columns: `id`, `treatment` (must equal 0), `event_time` (positive valued), and `status` (0 = right censored, 1 = observed event). For other outcomes, historical control datasets must have columns: `id`, `treatment` (must equal 0), and `y`. The parameter, `rand_control_diff` defines differences between historical controls and randomized controls. The meaning of `rand_control_diff` depends on the value of `outcome_type`. The following list details the meaning of `rand_control_diff` for each outcome type:

- Gaussian: Difference in two means (randomized controls minus historical controls).
- Bernoulli: An odds ratio (randomized controls over historical controls).
- Poisson: A mean ratio (randomized controls over historical controls).
- Weibull: A hazard ratio (randomized controls over historical controls).
- Lognormal: A mean ratio (randomized controls over historical controls).
- Piecewise exponential: A hazard ratio (randomized controls over historical controls).

Finally, the `a0_vals` parameter is a vector of values that defines the amount of information from the historical controls that will be included in the posterior estimation of the treatment effect. If an element of `a0_vals` is 1, then all the historical control information will be used. If an element of `a0_vals` is less than 1, then partial information will be used. If an element of `a0_vals` is 0, then no information will be used.

The function `historic_sim()` will take each combination of `subj_per_arm`, `effect_vals`, `a0_vals`, and `rand_control_diff` and run `trial_reps` replicates of the trial, then calculate the number of replicates where the null hypothesis was rejected, calculate the average effect, and if requested by the user, estimate values of variance, bias, and MSE on appropriate scales. Just like `simple_sim()`, the function `historic_sim()` will return a list. The contents of this list are the same as those returned by `simple_sim()` and are described above; however, the size of the resulting list can be much larger than the resulting list of `simple_sim()` because the dimensions of `a0_vals`, and `rand_control_diff` may not have size 1.

Each element of `data` is a four-dimensional array, where each dimension is determined by the length of parameters `subj_per_arm`, `a0_vals`, `effect_vals`, and `rand_control_diff`. The size of each four-dimensional array depends on which results are requested by the user. At a minimum, at least one of `subj_per_arm`, `a0_vals`, `effect_vals`, or `rand_control_diff` must contain at least 2 values, while the other three must contain at least 1 value. The simulation results `data` will always contain two elements: an array of power results (`power`) and an array of estimation results (`est`). In addition to `power` and `est`, data may also contain elements `var`, `bias`, or `mse`, depending on the values of `get_var`, `get_bias`, and `get_mse`. The values returned in `est` are in the form of hazard ratios, mean ratios, odds ratios, or mean differences depending on the value of `outcome_type` as previously described. The values returned in `bias`, `var`, and `mse` are on the scale of the values returned in `est`.

The end product from using `simple_sim()` or `historic_sim()` will be a set of tables or figures, each containing information about one trial operational characteristic as a function of a trial design characteristic, stratifying on a second trial design characteristic. For `simple_sim()` the table will always be power by sample size, stratified by a set of treatment effects, where power in this context is the power of a hypothesis test to reject the null hypothesis and not the power of the power prior $a_0$. For `historic_sim()`, the output will most likely be at least a table of power (power of a test) by sample size for a set of $a_0$ values (power for the power prior), given a specific effect and historical/randomized control difference. However, if a user passes to `historic_sim()` multiple sample sizes, multiple effect values, and multiple differences between historical and randomized controls, then the generated object will be a 4-dimensional array with all dimensions equal to 2 or more. The first dimension will be sample size, the second dimension will be $a_0$ values, the third dimension will be treatment effect, and the last dimension will be differences between randomized and historical controls.

As already mentioned, at a minimum two arrays of simulation results will be generated by `historic_sim()` and `simple_sim()`, one for power and one for treatment effect estimate. In `historic_sim()`, if two of the parameters (sample size, $a_0$ value, treatment effect, and control differences) are set to only one level, then output results will still be a 4-dimensional array but only two dimensions will have more than one level. As a result, the resulting arrays will basically be 2-dimensional. Like power and effect estimate, identical structures will occur in the variance, bias, and MSE arrays when requested.

In order to print and/or plot information contained in these arrays, a user can extract them

- Example table for power
- Effect size equal to 0.6

| | | $a_0$ | | |
|---|---|---|---|---|
| | | 0 | 0.5 | 1.0 |
| SS | 20 | 0.10 | 0.11 | 0.12 |
| | 40 | 0.20 | 0.24 | 0.40 |
| | 60 | 0.70 | 0.75 | 0.85 |
| | 80 | 0.90 | 0.92 | 0.95 |

- Example table for type I error
- Effect size equal to null case

| | | $a_0$ | | |
|---|---|---|---|---|
| | | 0 | 0.5 | 1.0 |
| SS | 20 | 0.06 | 0.05 | 0.04 |
| | 40 | 0.05 | 0.04 | 0.05 |
| | 60 | 0.05 | 0.06 | 0.03 |
| | 80 | 0.03 | 0.05 | 0.04 |

Table 2: Representation of output structure from `historic_sim()`. Historical and Randomized controls are from the same population.

from the `bayes_ctd_array` object and slice them like any other `R` array, or the user can use the **BayesCTDesign** slicing system implemented in the print and plot methods. Let `W` represent the sample size dimension, `X` the $a_0$ dimension, `Y` the effect dimension, and `Z` the historical/randomized control difference dimension. Moreover, let `AB|CD` represent a table that summarizes a trial operational characteristic with rows made from values of `A` and columns made from values of `B` while holding `C` and `D` constant. Then:

- `WX|YZ` will represent a power (estimate/var/bias/MSE) table with rows representing sample size values and columns representing power prior $a_0$ values, while holding effect size and historical/randomized control group difference constant.
- `WY|XZ` will represent a power (estimate/var/bias/MSE) table with rows representing sample size values and columns representing effect size values, while holding power prior $a_0$ parameter and historical/randomized control group difference constant.
- `WZ|XY` will represent a power (estimate/var/bias/MSE) table with rows representing sample size values and columns representing historical/randomized control group differences, while holding power prior $a_0$ parameter and effect size constant.
- `XY|WZ` will represent a power (estimate/var/bias/MSE) table with rows representing power prior $a_0$ parameter values and columns representing effect size values, while holding sample size and historical/randomized control group difference constant.
- `XZ|WY` will represent a power (estimate/var/bias/MSE) table with rows representing power prior $a_0$ parameter values and columns representing historical/randomized control group differences, while holding sample size and effect size constant.
- `YZ|WX` will represent a power (estimate/var/bias/MSE) table with rows representing effect size values and columns representing historical/randomized control group differences, while holding sample size and power prior $a_0$ parameter constant.

Since `simple_sim()` will return a 4 dimensional array with the `X` dimension and the `Z` dimension equal to 1, only the `WY|XZ` table from `simple_sim()` is meaningful. This table will contain all the information generated by a trial simulation call to `simple_sim()`.

The **BayesCTDesign** package can also produce results regarding type I error, when the null effect case is represented in the set of possible treatment effects. Table 2 shows the struc-

tural form of two possible output tables for two treatment effects when no differences between historical and randomized controls are present, one for power and the other for type I error. In the tables, SS refers to sample size and the columns represent 3 values of $a_0$. Finally, the **BayesCTDesign** package will generate similar arrays of treatment effect estimate, treatment effect variance, bias and MSE on an appropriate scale given the `outcome_type`, if the user requests such output. The `bayes_ctd_array` object created by `simple_sim()` or `historic_sim()` will have two methods. A print method is available to print two-dimensional tables from the `bayes_ctd_array` object. A plotting method is also available to plot data either contained in the `bayes_ctd_array` object or derived from the `bayes_ctd_array` object via `loess()` smoothing. The plot method calls the print method to generate the table prior to plotting. Both the print and the plot methods allow the user to print and plot data for power and treatment effect, as well as variance, bias, or MSE when available.

The user interface for the print method is:

```
print(bayes_ctd_array = NULL, measure = "power", tab_type = "WX|YZ",
  subj_per_arm_val = NULL, a0_val = NULL, effect_val = NULL,
  rand_control_diff_val = NULL, print_chg_warn = 1)
```

The object `bayes_ctd_array` is of class '`bayes_ctd_array`' that has been created with `simple_sim()` or `historic_sim()`. The value of `measure` can be either `"power"`, `"est"`, `"var"`, `"bias"`, or `"mse"`. The value of `tab_type` can be:

- `"WX|YZ"`: Trial operational characteristic by sample size (`W`), stratifying by $a_0$ (`X`), while holding treatment effect (`Y`) and control differences (`Z`) constant.
- `"WY|XZ"`: Trial operational characteristic by sample size (`W`), stratifying by treatment effect (`Y`), while holding $a_0$ (`X`) and control differences (`Z`) constant.
- `"WZ|XY"`: Trial operational characteristic by sample size (`W`), stratifying by control differences (`Z`), while holding $a_0$ (`X`) and treatment effect values (`Y`) constant.
- `"XY|WZ"`: Trial operational characteristic by $a_0$ (`X`), stratifying by treatment effect (`Y`), while holding sample size (`W`) and control differences constant (`Z`).
- `"XZ|WY"`: Trial operational characteristic by $a_0$ (`X`), stratifying by control differences (`Z`), while holding sample size (`W`) and treatment effect (`Y`) constant.
- `"YZ|WX"`: Trial operational characteristic by treatment effect (`Y`), stratifying by control differences (`Z`), while holding sample size (`W`) and $a_0$ (`X`) constant.
- `"ZX|WY"`: Trial operational characteristic by control differences (`Z`), stratifying by $a_0$ (`X`), while holding sample size ( `W`) and treatment effect (`Y`) constant.
- `"XW|YZ"`: Trial operational characteristic by $a_0$ ( `X`), stratifying by sample size (`W`), while holding treatment effect (`Y`) and control differences (`Z`) constant.
- `"YW|XZ"`: Trial operational characteristic by treatment effect (`Y`), stratifying by sample size (`W`), while holding $a_0$ (`X`) and control differences (`Z`) constant.
- `"YX|WZ"`: Trial operational characteristic by treatment effect (`Y`), stratifying by $a_0$ (`X`), while holding sample size (`W`) and control differences (`Z`) constant.

- "ZW|XY": Trial operational characteristic by control differences (Z), stratifying by sample size (W), while holding $a_0$ (X) and treatment effect constant (Y).
- "ZY|WX": Trial operational characteristic by control differences (Z), stratifying by treatment effect (Y), while holding sample size (W) and $a_0$ (X) constant.

W represents the sample size, X $a_0$, Y the treatment effect, and Z the historical/randomized control difference. The first letter in the tab_type indicates what parameter will be on the $x$-axis. The second letter in the tab_type indicates what parameter will be the stratifying parameter. The two letters after '|' indicate which parameters are being held constant. When sample size per arm is being held constant, subj_per_arm_val must be a value that was given to subj_per_arm in the call to simple_sim() or historic_sim(). Similarly, when a0_val is held constant it must be equal to a value for a_0 that was used in simple_sim() or historic_sim(). Finally, effect_val and rand_control_diff_val must be set equal to a value of effect_vals or rand_control_diff respectively that was used in simple_sim() or historic_sim(). The last parameter of print() is print_chg_warn. This last parameter is not primarily for the user, it is used by plot() to ensure warnings are not printed twice.

The user interface for the graph method will have a form such as:

```
plot(bayes_ctd_array, measure = "power", tab_type = "WX|YZ", smooth = FALSE,
  plot_out = TRUE, subj_per_arm_val = NULL, a0_val = NULL,
  effect_val = NULL, rand_control_diff_val = NULL, span = 0.75, degree = 2,
  family = "gaussian", title = NULL, ylim = NULL)
```

Most of the parameters for plot() are the same as for print() and they are used in the same manner. The parameters that are unique are smooth, plot_out, span, degree, and family. The parameter smooth is a TRUE/FALSE parameter indicating whether smoothed results should be plotted. Smoothing is done through a call to loess() and requires the length of the trial design characteristic (subj_per_arm or a0_val or effect_val or rand_control_diff_val) that populates the $x$-axis on the graph to contain enough elements to justify the smoothing. The method plot() does not check to see if enough elements are present to justify smoothing. The parameter plot_out is a TRUE/FALSE parameter indicating whether the plot should be produced. This toggle parameter is useful if the user only wants a table of smoothed values. The other parameters that are unique to plot() (span, degree and family) are parameters required for a call to loess() and are explained in the stats::loess() help page.

## 5. Examples

In this section we will illustrate the use of **BayesCTDesign** by working through an initial simple trial example followed by three more complex trial design investigations where one uses a Weibull outcome, another uses a piecewise exponential, and a third uses a Bernoulli outcome. The simple trial example will not take much time to run; however, the reader may want to read the other examples to get a sense of the **BayesCTDesign** functionality prior to running them. The purpose of the simple trial example is to give the reader a sense of what **BayesCTDesign** can do without the code taking a long time to run. The complex examples take longer to run, but are much richer in results. The purpose of the first complex example is to illustrate how **BayesCTDesign** can be used to power a study

which incorporates historical control data, and to illustrate how **BayesCTDesign** can be used to investigate effects of differences between historical and randomized controls on bias and power. These investigations can often lead to unusual results, and **BayesCTDesign** can be used to tabulate or visualize these results. The second complex example demonstrates the piecewise exponential capabilities of **BayesCTDesign**. Finally, the third complex example illustrates more of the plotting capabilities of the package, while showing that unusual results coming from differences between historical and randomized controls are not unique to the Weibull outcome. The reader may want to run a few examples of his own using a smaller set of trial characteristic combinations and starting off with a small number of replicates, and then continue to increase the number of replicates to get a sense of how long different simulation setups take to run. After the reader has a good sense of time needed to run simulations within his own computing environment, the reader may run the more complex trial examples in this paper, but even then the reader will want to reduce the number of replications substantially before running them until the reader is aware of how long the full examples will take to run.

### 5.1. Simple trial example

For our simple example of using **BayesCTDesign**, consider a scenario where we have historical control Weibull data from 60 subjects, which are right censored at 3.0. We want to determine the sample size needed for a two-arm clinical trial that will utilize information from these 60 control subjects to detect a treatment hazard ratio of 0.6 with 80% power and a two-sided $\alpha$ of 0.05 when all of the information in the historical controls is used ($a_0 = 1$) and no differences exist between randomized and historical controls. The following code will help the clinical trialist determine the necessary sample size. The results from this small simulation show us that about 80 subjects per arm are needed for 80% power. It took about 5 seconds to run on a 2.6 GHz i7-6700HQ Lenovo ThinkPad using only one core.

```
R> library("BayesCTDesign")
R> set.seed(2250)
R> SampleHistData <- genweibulldata(sample_size = 60, scale1 = 2.82487,
+    hazard_ratio = 1.0, common_shape = 3, censor_value = 3)
R> histdata <- subset(SampleHistData, subset = (treatment == 0))
R> histdata$id <- histdata$id + 10000
R> weibull_test <- historic_sim(trial_reps = 100, outcome_type = "weibull",
+    subj_per_arm = c(40, 70, 100), a0_vals = 1, effect_vals = 0.6,
+    rand_control_diff = 1, hist_control_data = histdata, censor_value = 3,
+    alpha = 0.05, get_var = TRUE, get_bias = TRUE, get_mse = TRUE,
+    seedval = 123)
R> print(weibull_test)
```

```
[1] "Since only subj_per_arm vector has more than 1 element,
 tab_type was set to WX|YZ"
[1] "This works towards putting all results in a single table"
  40   70  100
0.68 0.81 0.86
```

### 5.2. Complex trial example 1

If a user wants to use `historic_sim()` he should have access to historical control data with the required structure. However, **BayesCTDesign** also has several data generating functions that can be used to generate hypothetical control data, which can be used for exploratory purposes. For example, `genweibulldata()` can be used to simulate a trial where the outcome is a Weibull time-to-event variable. To use `genweibulldata()` to generate hypothetical historical control data, a user would create a simulated trial and then retain only the control data created. Now the historical control dataset needs to have a certain structure. For survival outcomes like a Weibull, the historical control dataset must have 4 columns: `id`, `treatment` (must equal 0), `event_time` (positive valued) and `status` (0 = right censored, 1 = observed event). When you run `genweibulldata()` the result will be hypothetical data for a control arm (`treatment = 0`) as well as an experimental arm (`treatment = 1`). To generate a hypothetical historical control group dataset, we use `genweibulldata()` to generate a full trial dataset and subset the data so that we keep only records where `treatment = 0`. In the following examples, we will use this process to generate the historical control datasets.

As our first complex example of using **BayesCTDesign**, consider a scenario where we have historical control Weibull data from 60 subjects, which are right censored at 3.0. We want to determine the sample size needed for a two-arm clinical trial that will utilize information from these 60 control subjects to detect a treatment hazard ratio of 0.7 with 80% power and a two-sided $\alpha$ of 0.05. Although the targeted treatment hazard ratio is 0.7, assume the clinical trialist believes the effect could range between 0.6 and 1.0. Initially, the clinical trialist believes the required sample size per arm will be in the range of 75 to 175 subjects per arm.

Since the clinical trialist might incorporate historical control data into the trial design, the clinical trialist needs to assess the risk of including these historical control data. The model will assume historical control and randomized control data are samples from the same control population. Yet, the historical control data might be very different from the randomized control data. If the control groups are different, including the historical control data may significantly bias the results, because the model will produce a biased estimate of the control group hazard. Assume the clinical trialist believes the randomized and historical control data might differ in such a way that the hazard ratio between the two control groups will be in the range of 0.8 to 1.2. Will such differences create unacceptable bias in the final Bayesian estimate of the treatment hazard ratio? If so, how might the clinical trialist mitigate against this possible bias?

The clinical trialist will need to determine if differences between randomized and historical controls within this range will have a significant biasing effect on power and other trial characteristics. If a significant bias is possible, the clinical trialist will need to determine what value of $a_0$ will mitigate the effects of differences between historic and randomized controls.

Putting the simulation setup together, the clinical trialist needs to study power to detect treatment hazard ratios (experimental over control) ranging from 0.6 to 1.0 while allowing sample size to range from 75 to 175 and including a sample of 60 historical controls. At the same time, the clinical trialist also needs to study the effects of randomized/control differences (control hazard ratio ranging from 0.8 to 1.2) when the power prior parameter, $a_0$, ranges from 0 (no historical control information included) to 1 (all historical control information included). Finally, assume the outcome is a Weibull distributed time-to-event variable, and the expected treatment hazard ratio is 0.7. The test will be two-sided: $H_0$: Treatment hazard

ratio = 1 vs $H_a$: Treatment hazard ratio $\neq 1$. With this information, the clinical trialist can run `historic_sim()` and generate several arrays of simulated output, each array containing information about a trial characteristic. In turn, these arrays of simulated output can be investigated to determine what sample size the trial should use and make a decision on what value of $a_0$ should be used. The code for this exploration is shown below.

```
R> set.seed(2250)
R> SampleHistData <- genweibulldata(sample_size = 60, scale1 = 2.82487,
+    hazard_ratio = 1.0, common_shape = 3, censor_value = 3)
R> histdata <- subset(SampleHistData, subset = (treatment == 0))
R> histdata$id <- histdata$id + 10000
R> weibull_test <- historic_sim(trial_reps = 500, outcome_type = "weibull",
+    subj_per_arm = c(75, 100, 125, 150, 175),
+    a0_vals = c(0, 0.25, 0.50, 0.75, 1),
+    effect_vals = c(0.6, 0.7, 0.8, 0.9, 1),
+    rand_control_diff = c(0.8, 1, 1.2),
+    hist_control_data = histdata, censor_value = 3, alpha = 0.05,
+    get_var = TRUE, get_bias = TRUE, get_mse = TRUE, seedval = 123)
```

As mentioned above, the call to `genweibulldata()` is used to generate the dataset of historical controls for illustration purposes only. Normally, the data generating functions in **BayesCTDesign** are only used by `simple_sim()` and `historic_sim()` to generate hypothetical trial data; however, the data generating functions are made available to the user so one can explore the capabilities of **BayesCTDesign** even if real historical control data is not available. In the call to `genweibulldata()`, we create a sample of 60 subjects per group where the Weibull scale parameter is 2.82487 and the shape parameter is 3. We right censor the data at 3. Since this use of `genweibulldata()` is for generating control group data only, we assign the hazard ratio to 1. After the data has been created, we subset the data to include only those subjects who are in the control group (`treatment = 0`).

On a 2.6 GHz i7-6700HQ Lenovo ThinkPad the above call to `historic_sim()` took about one hour to complete; however, this is much less than it would have taken if MCMC estimation was used within the simulation process. Once complete, this simulation gives a rich set of results to investigate via printing or plotting the results. In all, the simulation produces results for 375 different trial scenarios (5 sample sizes by 5 $a_0$ values by 5 effect values by 3 randomized and historical control differences).

First look at power by sample size, stratifying by $a_0$, while holding the effect to 0.6 and assuming no randomized/historical control differences. Code needed to create this table is shown below. Notice that we have `measure = "power"`, since we want a table of power. Also `tab_type = "WX|YZ"`, because for a table we want sample size (`W`) represented in the rows and we want $a_0$ (`X`) values in the columns, while holding treatment effect (`Y`) and historical/randomized control differences (`Z`) constant. Remember that for a Weibull outcome, the parameter used in **BayesCTDesign** to define a difference between historical and randomized controls is a hazard ratio, so we set `rand_control_diff_val=1.0` to look at the scenario where historical and randomized controls are not different. In this table, both rows and columns have appropriate labels to identify which sample size and $a_0$ value is represented by a power estimate in the table. On the one hand, we see that when the historical control data is ignored (`a0 = 0`), we need between 100 and 125 subjects per arm for 80% power (first

column with a heading). On the other hand, if all of the information in the historical control data is used (`a0 = 1`), then only 75 to 100 subjects per arm are needed for 80% power (last column with a heading).

```
R> test_table0_6 <- print(weibull_test, measure = "power",
+    tab_type = "WX|YZ", effect_val = 0.6, rand_control_diff_val = 1.0)
R> test_table0_6

        0  0.25   0.5  0.75     1
75  0.670 0.728 0.756 0.796 0.780
100 0.776 0.814 0.854 0.880 0.892
125 0.876 0.906 0.940 0.918 0.940
150 0.932 0.944 0.944 0.970 0.964
175 0.960 0.972 0.980 0.980 0.988
```

Next consider a similar table of power by sample size, stratifying by $a_0$, but this time holding the treatment effect to 0.7 and assuming no randomized/historical control differences. Code needed to create this table is shown below. In this scenario, we see that when the historical control data is ignored ($a_0 = 0$), we need over 175 subjects per arm for 80% power. In contrast, if all of the information in the historical control data is used ($a_0 = 1$), then 150 subjects per arm are needed for 80% power. As expected, reducing the treatment effect that we want to power requires us to collect more data to maintain a specified amount of power. Also, including the historical control data does give us 80% power within the expected sample size range.

```
R> test_table0_7 <- print(weibull_test, measure = "power",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_val = 1.0)
R> test_table0_7

        0  0.25   0.5  0.75     1
75  0.434 0.394 0.458 0.488 0.520
100 0.524 0.528 0.574 0.576 0.592
125 0.602 0.654 0.642 0.662 0.730
150 0.692 0.744 0.758 0.744 0.804
175 0.788 0.742 0.816 0.822 0.834
```

Now consider a table of power by sample size, stratifying by $a_0$, while holding the effect to 1.0 and assuming no randomized/historical control differences. Because the treatment effect (hazard ratio) is set to 1.0, this is a study of type I error. Code needed to create this table is shown below. In this table we see that regardless of how much information is used from the historical controls, the type I error is controlled. Average type I error when $a_0 = 0$ is 0.055, when $a_0 = 0.25$ the average type I error is 0.043. When $a_0 = 0.5$, 0.75, and 1, the average type I error is 0.038, 0.037, and 0.032 respectively. This table shows the interesting observation that when historical control data and randomized control data are from the same population, inclusion of historical control data using the power prior can reduce type I error below the nominal level. Note that, in a real trial design context, the trialist will want to rerun the simulation using many more replications than 500, especially for a study of type I error.

```
R> test_table1_0 <- print(weibull_test, measure = "power",
+    tab_type = "WX|YZ", effect_val = 1.0, rand_control_diff_val = 1.0)
R> test_table1_0
```

```
        0  0.25   0.5  0.75     1
75  0.044 0.048 0.044 0.030 0.028
100 0.060 0.044 0.040 0.028 0.030
125 0.052 0.052 0.032 0.036 0.028
150 0.058 0.036 0.046 0.048 0.030
175 0.060 0.036 0.028 0.044 0.042
```

The clinical trialist may also think that it possible that randomized and historical controls could differ on a hazard ratio scale of 0.8 to 1.2, so the trialist must also look into issues related to such differences. The following code and subsequent tables extract information needed from the simulations to study the effects of randomized/historical control non-compatibility on power, when the true treatment effect is 0.7.

When the hazard ratio between randomized to historical controls is 1.2, randomized controls tend to experience more events than historical controls. The effect of this on power is very interesting. Inclusion of the historical control data under this scenario decreases power!

```
R> test_table0_7_1_2 <- print(weibull_test, measure = "power",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_val = 1.2)
R> test_table0_7_1_2
```

```
        0  0.25   0.5  0.75     1
75  0.430 0.402 0.362 0.364 0.344
100 0.536 0.530 0.548 0.472 0.444
125 0.700 0.638 0.630 0.584 0.574
150 0.688 0.724 0.720 0.644 0.702
175 0.798 0.782 0.804 0.750 0.728
```

This decrease in power is due to a biased treatment effect estimate towards the null value of a hazard ratio equal to 1, which can be seen by replacing the `measure` parameter with `"est"` instead of `"power"`.

```
R> test_table0_7_1_2e <- print(weibull_test, measure = "est",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_val = 1.2)
R> test_table0_7_1_2e
```

```
        0  0.25   0.5  0.75     1
75  0.719 0.737 0.746 0.755 0.762
100 0.708 0.723 0.727 0.746 0.762
125 0.695 0.720 0.725 0.737 0.746
150 0.712 0.716 0.723 0.737 0.731
175 0.703 0.712 0.720 0.731 0.740
```

Why is there a bias towards the null? If we focus on expected values, then we can see part of the answer. For a given hazard among historical controls, call it $h_c$, the assumed hazard among randomized controls is $1.2 \times h_c$. Since the overall control hazard will be a weighted average of $h_c$ and $1.2 \times h_c$, the control hazard will be in the interval $(h_c, 1.2 \times h_c)$. Now, since the treatment effect is 0.7, which is relative to the randomized control hazard, the hazard among the experimental group is $0.7 \times 1.2 \times h_c$ or $0.84 \times h_c$. As such, we see the final hazard ratio will be in the interval $(0.84 \times h_c)/(b \times h_c)$, where $b \in (1, 1.2)$. It follows that the treatment effect will be a hazard ratio between 0.7 and 0.84 and biased towards the null.

Now, including the historical control data should have some effect on the posterior variance. The following code and table demonstrate, however, that the variance is not affected that much in the current context and actually decreases slightly as $a_0$ increases.

```
R> test_table0_7_1_2v <- print(weibull_test, measure = "var",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_val = 1.2)
R> test_table0_7_1_2v
```

```
        0  0.25   0.5  0.75     1
75  0.021 0.020 0.019 0.019 0.018
100 0.015 0.015 0.014 0.014 0.014
125 0.011 0.012 0.011 0.011 0.011
150 0.010 0.010 0.009 0.010 0.009
175 0.008 0.008 0.008 0.008 0.008
```

Now, when the hazard ratio between randomized to historical controls is 0.8, randomized controls tend to experience fewer events than historical controls and inclusion of the historical control data increases power.

```
R> test_table0_7_0_8 <- print(weibull_test, measure = "power",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_val = 0.8)
R> test_table0_7_0_8
```

```
        0  0.25   0.5  0.75     1
75  0.380 0.470 0.530 0.606 0.702
100 0.470 0.524 0.608 0.684 0.738
125 0.556 0.636 0.686 0.776 0.842
150 0.628 0.664 0.746 0.824 0.854
175 0.694 0.758 0.818 0.830 0.862
```

Of course this time, the increase in power is a result of biased estimation of treatment effect away from the null value of a hazard ratio equal to 1, as shown when the call to `print()` using `"est"` instead of `"power"` for the `measure` parameter.

```
R> test_table0_7_0_8e <- print(weibull_test, measure = "est",
+    tab_type = "WX|YZ", effect_val = 0.7, rand_control_diff_va = 0.8)
R> test_table0_7_0_8e
```

```
        0  0.25   0.5  0.75     1
75  0.716 0.679 0.678 0.651 0.634
100 0.712 0.700 0.678 0.668 0.644
125 0.706 0.693 0.682 0.669 0.651
150 0.708 0.699 0.684 0.665 0.662
175 0.703 0.701 0.683 0.673 0.670
```

This time, the hazard among randomized controls is $0.8 \times h_c$. Since the overall control hazard will be a weighted average of $h_c$ and $0.8 \times h_c$, the control hazard will be in the interval $(0.8 \times h_c, h_c)$. Now, since the treatment effect is still 0.7, the hazard among the experimental group is $0.7 \times 0.8 \times h_c$ or $0.56 \times h_c$. As such, we see the final hazard ratio will be around $(0.56 \times h_c)/(b \times h_c)$, where $b \in (0.8, 1)$. It follows that the estimated treatment effect will be a hazard ratio between 0.56 and 0.7 and biased away from the null.

Based on the tables seen so far, it seems that when all the information in the historical control data is used, the trial needs to enroll about 150 subjects per arm for 80 percent power to detect a 0.7 hazard ratio between experimental and controls groups. Yet, what are the effects of randomized/historical control differences? A set of different tables can be used to address this question. One the one hand, table `test_table0_7_1_2` showed that if the true randomized experimental to control group hazard ratio is 0.7, but the randomized and historical controls differ by a hazard ratio of 1.2, then the power drops from 0.80 to 0.70 (fourth row, last column) when sample size remains at 150. On the other hand, table `test_table0_7_0_8` showed that if the true randomized experiment to control group hazard ratio is 0.7, but the randomized and historical controls differ by a hazard ratio of 0.8, then the power increases from 0.80 to 0.85 (fourth row, last column). In both cases, the change in power is due to bias in treatment effect estimation caused by the differences in randomized and historical controls. As we saw in tables `test_table0_7_0_8e` and `test_table0_7_1_2e`, this bias can go in either direction depending on the true treatment effect among randomized in relation to the randomized and historical control group differences.

To get a final view of the effect of historical/randomized control differences, consider table `test_table150_0_7` given below. In the following code `tab_type = "ZX|WY"` is used, so it is a table where the rows will be values of historical/randomized control differences and columns will be different $a_0$ values. Notice that in this table, we see that when $a_0$ is equal to 0.25 (second column), the historical/randomized control difference of 0.8 tended to have lower power than the historical/randomized control difference of 1.2. Also, when $a_0$ is equal to 0.75 (fourth column), the historical/randomized control difference of 0.8 tended to have higher power than the historical/randomized control difference of 1.2. Finally, when $a_0$ is equal to 0.5, power is rather close regardless of the historical/randomized control differences.

```
R> test_table150_0_7 <- print(weibull_test, measure = "power",
+    tab_type = "ZX|WY", subj_per_arm_val = 150, effect_val = 0.7)
R> test_table150_0_7
```

```
        0  0.25   0.5  0.75     1
0.8 0.628 0.664 0.746 0.824 0.854
1   0.692 0.744 0.758 0.744 0.804
1.2 0.688 0.724 0.720 0.644 0.702
```

If the randomized control trial sample size could be increased so that this third column was roughly 0.80, then a trial with $a_0 = 0.5$ and that sample size per arm would be rather robust to historical/randomized control differences within the range of 0.8 to 1.2. This is illustrated in `test_table175_0_7`.

```
R> test_table175_0_7 <- print(weibull_test, measure = "power",
+    tab_type = "ZX|WY", subj_per_arm_val = 175, effect_val = 0.7)
R> test_table175_0_7

        0  0.25   0.5  0.75     1
0.8 0.694 0.758 0.818 0.830 0.862
1   0.788 0.742 0.816 0.822 0.834
1.2 0.798 0.782 0.804 0.750 0.728
```

When a sample size of 175 per arm is considered, we find that at $a_0$=0.5 the estimated power is always 0.80 or greater. In conclusion, a clinical trialist could conclude that at least 80% power will be obtained when 175 subjects are randomized into each of two arms (experimental and control groups) and the true effect is equal to 0.7, and when data from 60 historical controls are included in the trial using a power prior with $a_0$ equal to 0.5 and the historical controls do not differ from randomized controls in terms of a hazard ratio beyond the range 0.8 to 1.2.

Earlier, we saw that a simple randomized trial of 175 subjects per arm would not be sufficient to have 80% power to detect a true hazard ratio of 0.7 with a two-sided test and an $\alpha$ of 0.05. With the addition of 60 historical controls using a power prior with $a_0 = 0.5$, we see that the trial now has at least 80% power to detect the true hazard ratio of 0.7 as long as the historical and randomized controls do not differ more than what was explored in these simulations. Finally, `test_table175_1_0` shows that when the sample size is set to 175 and the true treatment effect is 1.0, the power (which now is equal to type I error) is always less than 0.05, which demonstrates the conservatism of the test. We say the test is conservative since the probability of committing a type 1 error is less than the pre-specified value of 0.05. Based on these investigations, the clinical trialist has a good idea of how to incorporate the historical controls and gain some power without inflating type I error as long as the differences between historical and randomized controls do not go beyond what is expected.

```
R> test_table175_1_0 <- print(weibull_test, measure = "power",
+    tab_type = "ZX|WY", subj_per_arm_val = 175, effect_val = 1.0)
R> test_table175_1_0

        0  0.25   0.5  0.75     1
0.8 0.076 0.048 0.040 0.064 0.074
1   0.060 0.036 0.028 0.044 0.042
1.2 0.052 0.036 0.046 0.058 0.056
```

In addition to tabulating the results as we have done, we can also plot similar slices (two-dimensional tables) of the results. For example, the following code creates a plot of power as a function of sample size and $a_0$ value, when the true experiment to control hazard ratio is 0.7 and the randomized and historical controls differ by a hazard ratio of 1.2, see Figure 9. In this call to `plot()`, we have set `smooth = FALSE`, so no smoothing is applied to the simulation results. Also, `plot_out = TRUE`, therefore, we request that the graph is created.

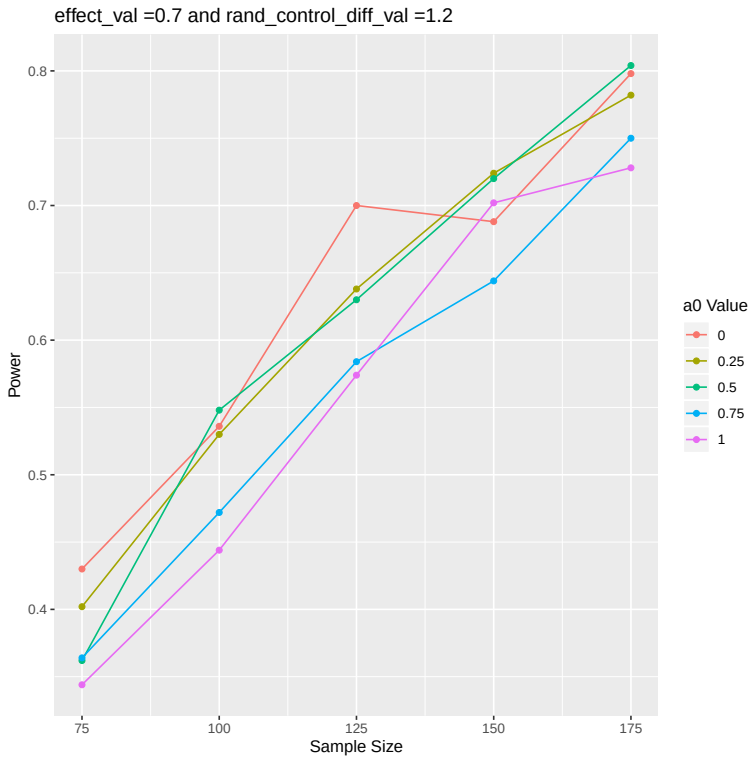


Figure 9: Power as a function of sample size, stratified by $a_0$, while holding effect to a hazard ratio of 0.7 and a hazard ratio between randomized and historical controls = 1.2.

```
R> plot(weibull_test, measure = "power", tab_type = "WX|YZ",
+     smooth = FALSE, plot_out = TRUE, effect_val = 0.7,
+     rand_control_diff_val = 1.2)
```

Although Figure 9 clearly indicates that an insufficient number of replications (500) were used to estimate each point on the graph, we can nevertheless see the general tendency for power to drop when the treatment effect is less than 1 and the randomized controls have a higher hazard for the outcome than the historical controls.

Although this simulation took about an hour to run, the results were very rich in information to not only design parts of the trial, but also gain in understanding how differences between randomized and historical controls can affect posterior estimation.

### 5.3. Complex trial example 2

For the next example, we will look at a trial that will use the piecewise exponential outcome. A clinical trialist may want to simply design a two-arm Bayesian trial that does not include historical data. Consider the case where a clinical trialist wants to use a piecewise exponential model with 6 time intervals (0 to 0.3 years, 0.3 to 0.9 years, 0.9 to 1.5 years, 1.5 to 2.1 years, 2.1 to 2.4 years, and 2.4 years or higher). Assume the vector of constant hazards for this six-piece PWE is (0.19, 0.35, 0.56, 0.47, 0.38, 0.34). Finally, assume the true hazard ratio (experimental over control) within any time interval is 0.8, but the clinical trialist wants to study hazard ratios ranging from 0.6 to 1.0. We now are ready to use `simple_sim()` to

determine the required sample size if the clinical trialist wants the trial to have 80% power to detect a hazard ratio of 0.8 using a two-sided test and $\alpha = 0.05$. The clinical trialist believes the required sample size will be in the range of 400 to 550 subjects. With `simple_sim()`, we need to assign `control_parms` the assumed parameters for the randomized control group. The code for the call to `simple_sim()` is given below. On a 2.6 GHz i7-6700HQ Lenovo ThinkPad, this simulation took about 4.1 hours.

```
R> set.seed(2250)
R> time.vec <- c(0.3, 0.9, 1.5, 2.1, 2.4)
R> lambdaHC.vec <- c(0.19, 0.35, 0.56, 0.47, 0.38, 0.34)
R> pwe_test <- simple_sim(trial_reps = 500, outcome_type = "pwe",
+    subj_per_arm = c(400, 425, 450, 475, 500, 525, 550),
+    effect_vals = c(0.6, 0.7, 0.8, 0.9, 1.0),
+    control_parms = lambdaHC.vec, time_vec = time.vec,
+    censor_value = 3, alpha = 0.05, get_var = TRUE,
+    get_bias = TRUE, get_mse = TRUE, seedval = 123)
```

The following code and table shows the simulation results. Remember that when results from `simple_sim()` are being printed or plotted, there is no need to define a value for `tab_type`. The `simple_sim()` function creates an array that really only has information in two-dimensions. The `print()` method automatically extracts the needed dimensions. Notice how `print()` prints out a few messages to indicate that it reduced the simulation results down to a single table.

```
R> pwe_test_table <- print(pwe_test, measure = "power")

[1] "Since simple_sim was used, tab_type was set to WX|YZ"
[1] "Values for tab_type, subj_per_arm_val, a0_val, effect_val,
and rand_control_diff_val were ignored"
[1] "This works towards putting all results in a single table,
effect by sample size"

R> pwe_test_table

    0.6   0.7   0.8   0.9     1
400   1 0.974 0.734 0.212 0.060
425   1 0.978 0.754 0.230 0.052
450   1 0.992 0.812 0.250 0.048
475   1 0.990 0.796 0.260 0.054
500   1 0.994 0.820 0.266 0.078
525   1 0.988 0.828 0.278 0.068
550   1 0.998 0.848 0.318 0.064
```

Based on these results, the trial will need to enroll between 475 and 500 subjects per arm to detect a hazard ratio of 0.8 with 80% power at an $\alpha$ of 0.05.

This second example took a good bit longer because it used the PWE outcome. Anytime the PWE outcome is used, the total simulation time will increase.

## 5.4. Complex trial example 3

The third and final example will give us another chance to see how **BayesCTDesign** allows us to explore randomized/historical control differences and their sometimes unexpected results. Consider a scenario where a hematologist wants to design a two-arm clinical trial to study the efficacy of a novel therapy for Immune Thrombocytopenia (ITP) relative to standard of care. The outcome is a (0/1) outcome where 0 indicates no ITP related bleeding during 1 year of treatment (no relapse). The trial needs to detect an odds ratio of 0.7 with at least 80% power with a two-sided $\alpha$ of 0.05. Budget constraints limit enrollment to 480 subjects per arm. Finally, the clinical trialist has data from 60 historical controls who received standard of care. Probability of relapse among historical controls is 0.6.

The clinical trialist has a few questions that need answers.

1. Will we have enough power to detect an odds ratio of 0.7 without incorporating the historical control data?
2. If a simple randomized trial will not have enough power, will incorporation of historical data result in at least 80% power?
3. If the historical control data is necessary, what are the consequences of reasonable differences between historical and randomized controls?

The code needed to answer the first question and the resulting table of simulation results are given below. Since we do not use historical control data to answer this first question, `simple_sim()` is used. On a 2.6 GHz i7-6700HQ Lenovo ThinkPad, this simulation took about 27 minutes. Notice that in this example the replications have been increased from 500 to 10000.

```
R> BasicTwoArm.Bernoulli <- simple_sim(trial_reps = 10000,
+    outcome_type = "bernoulli",
+    subj_per_arm = c(180, 220, 240, 280, 320, 360, 400, 440,
+    480, 520), effect_vals = c(0.6, 0.7, 0.8, 0.9, 1.0),
+    control_parms = c(0.6), alpha = 0.05, get_bias = TRUE,
+    get_mse = TRUE, seedval = 123, quietly = FALSE)
R> print(BasicTwoArm.Bernoulli)

[1] "Since simple_sim was used, tab_type was set to WX|YZ"
[1] "Values for tab_type, subj_per_arm_val, a0_val, effect_val,"
[1] "  and rand_control_diff_val were ignored"
[1] "This works towards putting all results in a single table,"
[1] "  effect by sample size"
       0.6    0.7    0.8    0.9      1
180 0.6723 0.3845 0.1766 0.0734 0.0458
220 0.7676 0.4535 0.2090 0.0793 0.0458
240 0.7950 0.4911 0.2217 0.0813 0.0424
280 0.8390 0.5475 0.2560 0.0914 0.0524
320 0.9028 0.6093 0.2819 0.0991 0.0500
360 0.9274 0.6462 0.3154 0.1060 0.0530
```

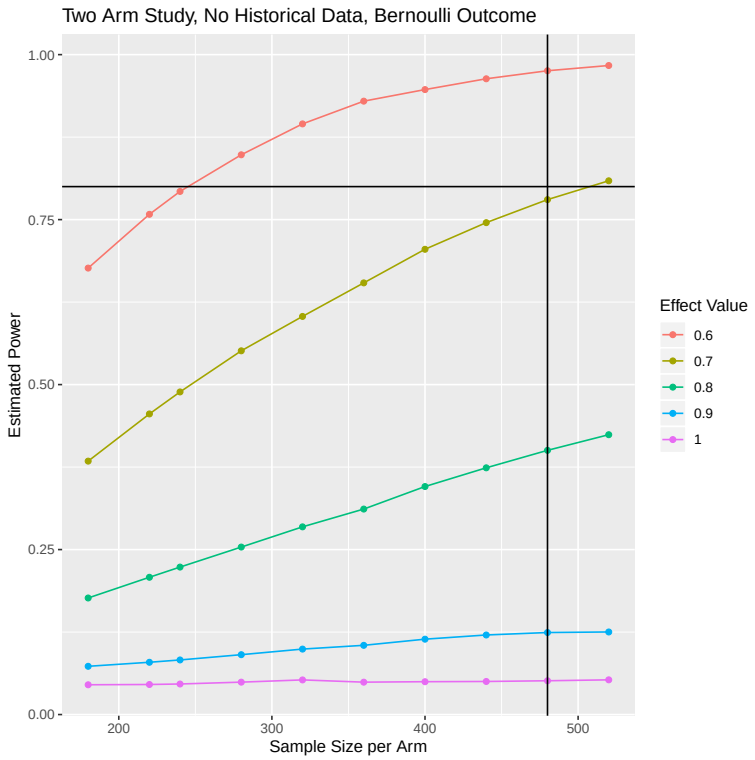


Figure 10: Power by sample size, stratified by $a_0$ value.

```
400 0.9460 0.7097 0.3383 0.1108 0.0440
440 0.9645 0.7488 0.3842 0.1263 0.0562
480 0.9753 0.7726 0.3932 0.1211 0.0481
520 0.9835 0.8119 0.4261 0.1259 0.0533
```

From this table of results, we see that between 480 and 520 subjects per arm are needed to detect an OR of 0.7, if we ignore the historical data and simply run a two-arm randomized trial. These same results can be plotted using `plot()`. Since `plot()` uses **ggplot2**, the output from `plot()` can be modified using **ggplot2** commands once the **ggplot2** package is loaded. The following code creates the basis plot using a call to `plot()`; however, additional calls to other **ggplot2** functions add a vertical line at a sample size of 480 and a horizontal line at 0.80, and the title is improved. As with the table, Figure 10 shows us that 480 subjects per arm is insufficient to produce a design with 80% power.

```
R> library("ggplot2")
R> BasicPlot <- plot(BasicTwoArm.Bernoulli, measure = "power",
+    tab_type = "WX|YZ", smooth = TRUE)
R> BasicPlot <- BasicPlot + geom_hline(yintercept = 0.80)
+    geom_vline(xintercept = 480)
R> ggtitle("Two Arm Study, No Historical Data, Bernoulli Outcome")
+    xlab("Sample Size per Arm") + ylab("Estimated Power")
R> BasicPlot
```

Given 480 subjects was the maximum number of subjects per arm the hematologist could enroll, the next step is to address question 2. Will including the historical control data, increase the power to at least 80%? To answer this question, `historic_sim()` must be used. The simulation code to address question 2 as well as the results are given below. On a 2.6 GHz i7-6700HQ Lenovo ThinkPad, this simulation took about 1.7 days. Yes, this code takes a very long time to run even using the BCLT, but the simulation setup looks at 1250 trial scenarios, the same code would take weeks to run with MCMC, and each trial characteristic combination is estimated very accurately with 10000 replicates. Notice in this example, hypothetical historical control data is being used and it is generated with the function `genbernoullidata()` that is available in **BayesCTDesign**.

```
R> set.seed(2250)
R> samplehistdata <- genbernoullidata(sample_size = 60, prob1 = 0.6,
+    odds_ratio = 1.0)
R> histdata <- subset(samplehistdata, subset = (treatment == 0))
R> histdata$id <- histdata$id + 10000
R> HistoricTwoArm.Bernoulli <- historic_sim(trial_reps = 10000,
+    outcome_type = "bernoulli",
+    subj_per_arm = c(180, 220, 240, 280, 320, 360, 400, 440, 480, 520),
+    a0_vals = c(0, 0.25, 0.5, 0.75, 1),
+    effect_vals = c(0.6, 0.7, 0.8, 0.9, 1.0),
+    rand_control_diff = c(0.6, 0.8, 1.0, 1.2, 1.4),
+    hist_control_data = histdata, time_vec = NULL, alpha = 0.05,
+    get_var = TRUE, get_bias = TRUE, get_mse = TRUE, seedval = 123,
+    quietly = FALSE)
R> print(HistoricTwoArm.Bernoulli, measure = "power",
+        tab_type = "WY|XZ", a0_val = 1.0,
+        rand_control_diff_val = 1.0)

       0.6    0.7    0.8    0.9      1
180 0.7494 0.4275 0.1910 0.0693 0.0368
220 0.8128 0.5077 0.2223 0.0741 0.0362
240 0.8523 0.5413 0.2431 0.0827 0.0386
280 0.8910 0.5995 0.2770 0.0890 0.0421
320 0.9279 0.6586 0.3006 0.0999 0.0421
360 0.9480 0.7064 0.3418 0.1038 0.0432
400 0.9640 0.7437 0.3593 0.1084 0.0419
440 0.9780 0.7841 0.3988 0.1177 0.0433
480 0.9853 0.8110 0.4247 0.1215 0.0424
520 0.9894 0.8364 0.4577 0.1373 0.0430
```

As with question 1, the graphical results needed to answer question 2 can be modified using **ggplot2** commands. Figure 11 shows that power is above 80% at 480 subjects per arm when historical control data is included in the estimation process.

```
R> HistoricPlot <- plot(HistoricTwoArm.Bernoulli, measure = "power",
+    tab_type = "WY|XZ", a0_val = 1.0,
+    rand_control_diff_val = 1.0, smooth = TRUE)
```

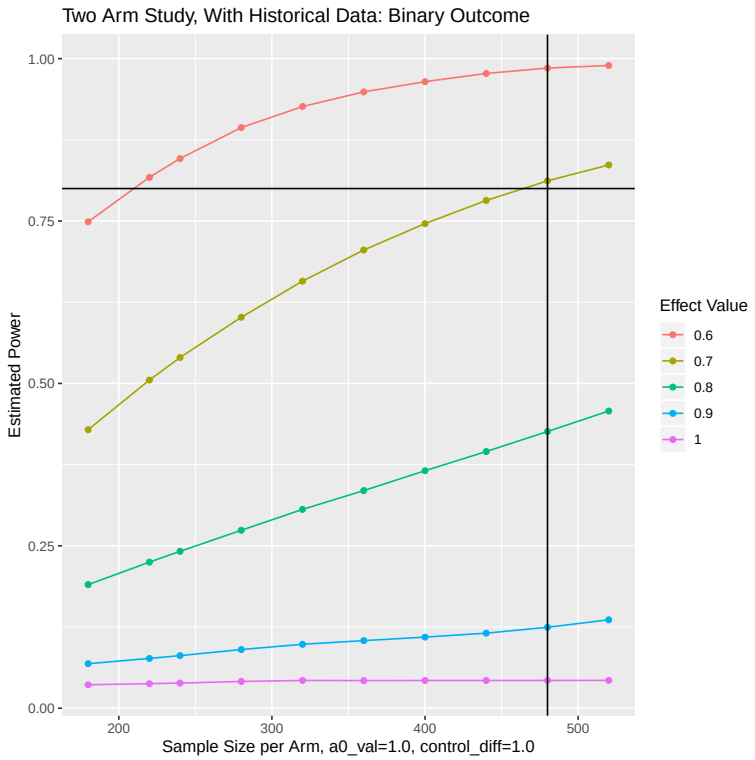


Figure 11: Power by sample size, stratified by $a_0$ value.

```
R> HistoricPlot <- HistoricPlot + geom_hline(yintercept = 0.80)
+    geom_vline(xintercept = 480)
+    ggtitle("Two Arm Study, With Historical Data: Binary Outcome")
+    xlab("Sample Size per Arm, a0_val=1.0, control_diff=1.0")
+    ylab("Estimated Power")
R> HistoricPlot
```

Now we are ready to study the consequences of differences between randomized and historical controls. First consider a plot of power by sample size, stratifying on control differences, Figure 12.

```
R> HistoricPlot2 <- plot(HistoricTwoArm.Bernoulli, measure = "power",
+    tab_type = "WZ|XY", a0_val = 1.0, effect_val = 0.7, smooth = TRUE)
R> HistoricPlot2 <- HistoricPlot2
+    ggtitle("Two Arm Study, With Historical Data: Binary Outcome")
+    xlab("Sample Size per Arm, a0_val = 1.0, effect = 0.7")
+    ylab("Estimated Power")
R> HistoricPlot2
```

Figure 12 has the interesting result that as the control effect $OR_c$ decreases power increases, where $OR_c$ is the odds ratio between randomized controls and historical controls. Similarly, while control effect $OR_c$ increases power decreases. Why are these changes occurring? Consider the case where control differences are not present ($OR_c = 1.0$), and let $a_0 > 0$ and the

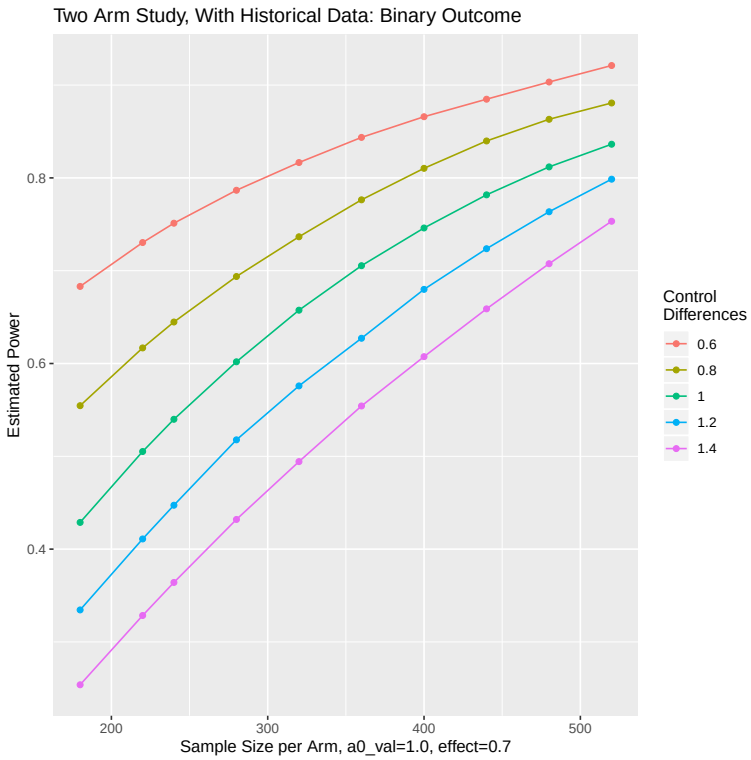


Figure 12: Power by sample size, stratified by control differences.

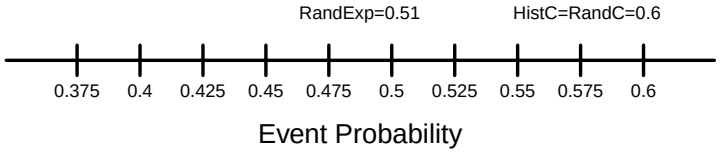


Figure 13: Success probabilities when no control differences are present. HistC = event probability for historical controls, RandC = event probability for randomized controls, RandExp = event probability for randomized experimental group.

treatment effect be $OR_t = 0.7$, see Figure 13. Since the control groups are not different, the effect estimate of experimental group over control group will be unbiased. Also, as sample size increases, power naturally increases. If control differences are present and the control effect is $OR_c = 0.6$, $a_0 > 0$, and the treatment effect is $OR_t = 0.7$, then what is the consequence of these control differences on power as a function of sample size? The consequence is illustrated in Figure 14. In this case, the probability of event in the randomized controls will be less than the probability in historical controls, since $OR_c = 0.6$. When the overall estimate of the control probability of event is estimated it will be a weighted average of the event probability in the randomized controls and the historical controls. As a result, the overall control probability will be pulled upward. All the while, the event probability in the randomized experimental group will be estimated without bias. It will represent the event probability needed to produce a treatment effect of 0.7 relative to randomized controls. The result will be an odds ratio experimental group over control that is biased downward to OR values closer

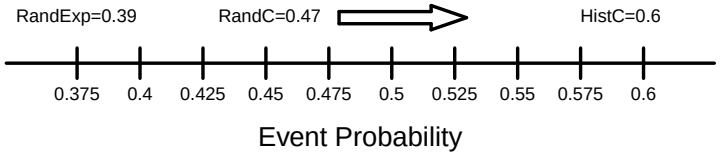


Figure 14: Success probabilities when control differences are present and randomized controls at less risk than historical controls. HistC = event probability for historical controls, RandC = event probability for randomized controls, RandExp = event probability for randomized experimental group.

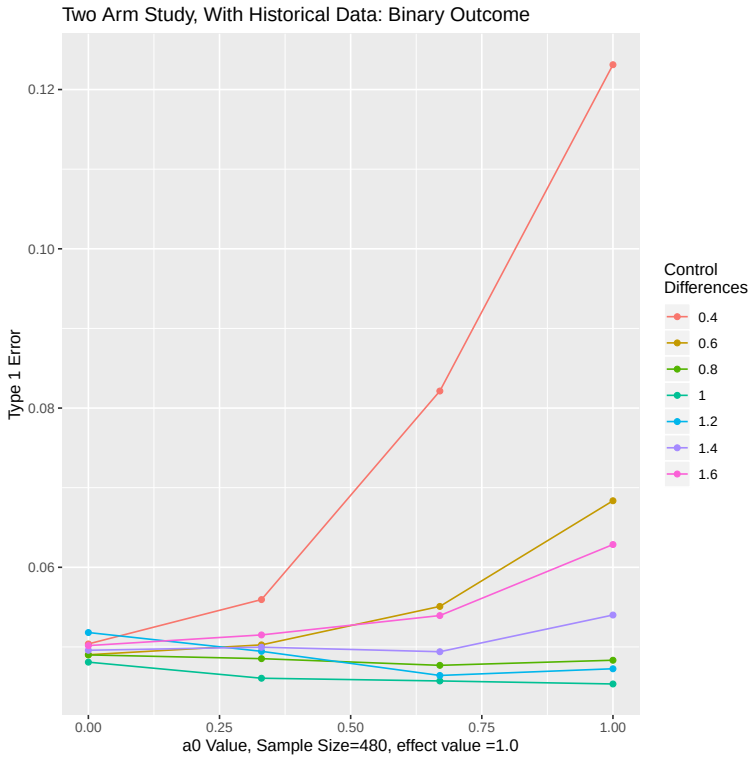


Figure 15: Type I error by control differences, stratified by $a_0$ values.

to zero and away from the true value. The treatment odds ratio is biased downward and the increase in power is artificial. Next, let us look at what happens to type I error when historic and randomized controls are different. Code for studying the type I error is given below. On a 2.6 GHz i7-6700HQ Lenovo ThinkPad, this code took about 11.2 hours to run. The plot is shown in Figure 15. Notice that `trial_reps = 100000`. As noted before, this code will take several hours to run, but the results will be rich in detail and take much less time than a similar MCMC based study.

```
R> HistoricTwoArm.Bernoulli2 <- historic_sim(trial_reps = 100000,
+    outcome_type = "bernoulli", subj_per_arm = c(480),
+    a0_vals = c(0, 0.33, 0.67, 1), effect_vals = c(1.0),
+    rand_control_diff = c(0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6),
```

```
+    hist_control_data = histdata, time_vec = NULL,
+    alpha = 0.05, get_var = TRUE, get_bias = TRUE,
+    get_mse = TRUE, seedval = 123, quietly = FALSE)
R> HistoricPlot3 <- plot(HistoricTwoArm.Bernoulli2,
+    measure = "power", tab_type = "XZ|WY", effect_val = 1.0,
+    subj_per_arm_val = 480, smooth = FALSE)
R> HistoricPlot3 <- HistoricPlot3 +
+    ggtitle("Two Arm Study, With Historical Data: Binary Outcome")
+    xlab("a0 Value, Sample Size = 480, effect value = 1.0")
+    ylab("Type 1 Error")
R> HistoricPlot3
```

When historic and randomized controls are not different, there is a tendency for type I error to decrease as $a_0$ increases (more historical information is included in estimation). In contrast, when historic and randomized controls are different, large control differences along with large $a_0$ values can result in highly inflated type I error.

However, simulations may show that type I erroris not unduly inflated for a range of control differences given a specific $a_0$. If background knowledge supports the belief that controls will not differ more than what is predicted by this range, then $a_0$ can be pre-specified and type I error controlled.

To further explore power when historic and randomized controls are different, we perform a simulation looking at effects on power due to control group differences and different values of $a_0$. The simulation code and figure code is given below and results are shown in Figure 16. In this simulation, we look at 9 difference values of randomized and control differences. On a 2.6 GHz i7-6700HQ Lenovo ThinkPad, this code took about 1.5 hours to run. Note that we hold sample size to 480 subjects and we set the treatment effect, $OR_t$, equal to 0.7.

```
R> HistoricTwoArm.Bernoulli3 <- historic_sim(trial_reps = 10000,
+    outcome_type = "bernoulli",
+    subj_per_arm = c(480),
+    a0_vals = c(0, 0.33, 0.66, 1),
+    effect_vals = c(0.7),
+    rand_control_diff = c(0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4),
+    hist_control_data = histdata, time_vec = NULL,
+    alpha = 0.05, get_var = TRUE, get_bias = TRUE, get_mse = TRUE,
+    seedval = 123, quietly=FALSE)
R> HistoricPlot4 <- plot(HistoricTwoArm.Bernoulli3,
+    measure = "power", tab_type = "ZX|WY",
+    subj_per_arm = 480, effect_val = 0.7,
+    smooth=TRUE)
R> HistoricPlot4 <- HistoricPlot4
+    ggtitle("Two Arm Study, With Historical Data: Binary Outcome")
+    xlab("Control Differences (OR), subj_per_arm = 480, effect = 0.7")
+    ylab("Power")
R> HistoricPlot4
```

In Figure 16, we see that when $OR_c = 1$, as $a_0$ increases, power increases as expected. In contrast, when $OR_c$ is large, in this example $OR_c > 1.2$, as $a_0$ increases, power decreases. Going

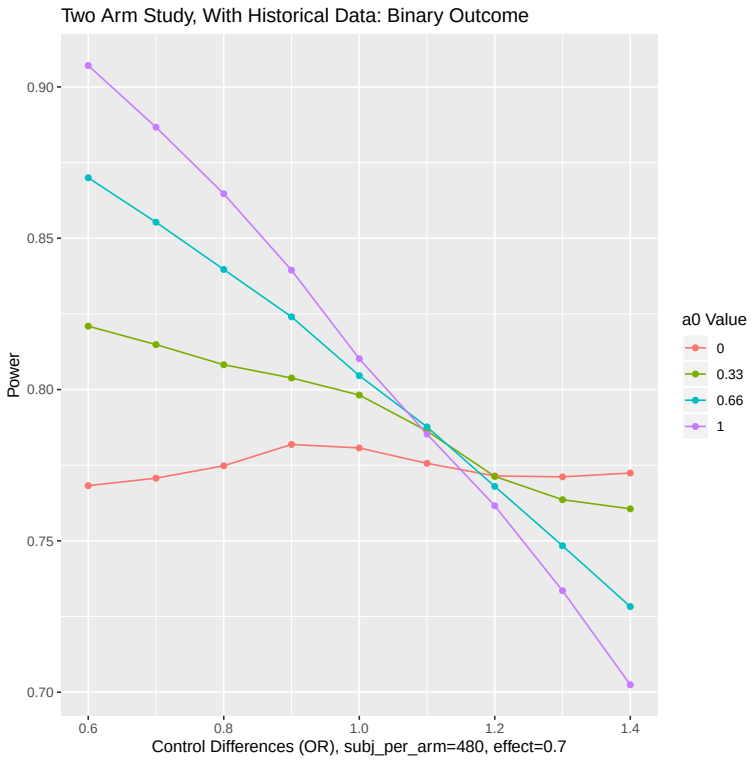


Figure 16: Plot of power by control differences stratified on $a_0$ values.

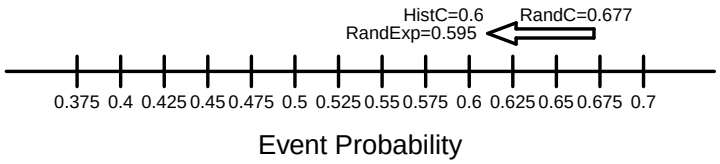


Figure 17: Success probabilities when control differences are present and randomized controls are at a higher risk than historical controls. HistC = event probability for historical controls, RandC = event probability for randomized controls, RandExp = event probability for randomized experimental group.

in the other direction, when $OR_c < 1.0$ and small, we see that power continues to increases as $a_0$ increases. Figure 17 explains why power decreases if $OR_c$ is large, while Figure 14 explains why power increases if $OR_c$ is small and $< 1.0$. Figure 14 was discussed earlier. When $OR_c$ is large and greater than 1, the probability of event is greater in the randomized controls than in the historical controls. Yet, the treatment effect is still 0.7, so the event probability in the randomized experimental group is less than the event probability in the randomized controls. In this example, the event probability in the randomized experimental group is slightly smaller than the event probability among historical controls. Overall, the estimated control event probability will be a weighted average of the two control groups, thus the estimated control event probability will be pulled downward towards the randomized experimental group event probability. The consequence will be lower power than when historical controls were ignored.

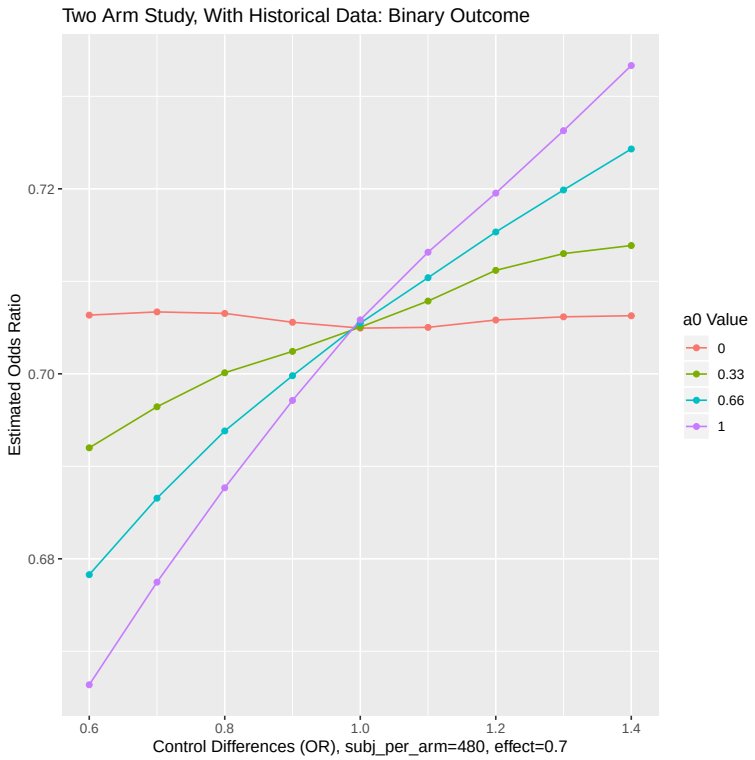


Figure 18: OR estimation by control differences stratifying on $a_0$ values.

The following code looks at the same simulation from the perspective of estimated treatment odds ratio, $OR_t$.

```
R> HistoricPlot5 <- plot(HistoricTwoArm.Bernoulli3, measure = "est",
+    tab_type = "ZX|WY", subj_per_arm = 480, effect_val = 0.7, smooth = TRUE)
R> HistoricPlot5 <- HistoricPlot5
+    ggtitle("Two Arm Study, With Historical Data: Binary Outcome")
+    xlab("Control Differences (OR), subj_per_arm = 480, effect = 0.7")
+    ylab("Estimated Odds Ratio")
R> HistoricPlot5
```

In Figure 18, we see that when $a_0 = 1$, all estimation routines are roughly unbiased. The estimated treatment effect is just over 0.7. In contrast, when $OR_c < 1$, we see that the estimated $OR_t$ is biased downward. Likewise, when $OR_c > 1$, we see that the estimated $OR_t$ is biased upward.

## 6. Discussion

**BayesCTDesign** was developed to allow users to study two-arm Bayesian designs that might include historical control data. The package uses simulation to estimate trial design characteristics under user-defined scenarios. Given simulation is used, the time it takes to generate results is longer than the time required by closed form solutions; however, by using Bayesian

central limit theorem, **BayesCTDesign** saves a substantial amount of time relative to simulation methods that make calls to external MCMC programs like WinBUGS or **JAGS**. The package allows the user to study designs with Gaussian, Poisson, Bernoulli, Weibull, lognormal, and piecewise exponential outcomes.

Future development of the package will allow users to study unequal arm sizes, informative initial priors, and add a Poisson model for relative risk estimation. One possible development will be potentially dynamic determination of power prior parameter. For example, incorporating the estimation process contained in the **bayesDP** package to allow for designing trials that use a discount formula for dynamic borrowing, (Balcome *et al.* 2021). Another potential development is to add functionality that allows for inclusion of historical data from actively treated subjects as well as controls and follow the framework for Bayesian P-value calculations as described in Psioda and Ibrahim (2019).

## Computational details

The results in this paper were obtained using R 4.1.2 (R Core Team 2021). R itself and all packages used are available from the CRAN at https://CRAN.R-project.org/.

## Acknowledgments

The development of this package was funded by Clinical Trials Development Resource for Hematologic Disorders (U24) 1U24HL114577.

## References

Anderson K (2021). ***gsDesign****: Group Sequential Design*. R package version 3.2.1, URL https://CRAN.R-project.org/package=gsDesign.

Balcome S, Musgrove D, Haddad T, Jackson C (2021). ***bayesDP****: Tools for the Bayesian Discount Prior Function*. R package version 1.3.5, URL https://CRAN.R-project.org/package=bayesDP.

Berry SM, Carlin BP, Lee JJ, Müller P (2011). *Adaptive Methods for Clinical Trials*. Chapman & Hall/CRC, Boca Raton.

Bolstad WM (2007). *Introduction to Bayesian Statistics*. 2nd edition. John Wiley & Sons, Hoboken.

Broström G (2012). *Event History Analysis with R*. Chapman & Hall/CRC, Boca Raton. doi:10.1201/9781315373942.

Champely S (2020). ***pwr****: Basic Functions for Power Analysis*. R package version 1.3-0, URL https://CRAN.R-project.org/package=pwr.

Chen N, Lee JJ (2020). ***bacistool****: Bayesian Classification and Information Sharing (BaCIS) Tool for the Design of Multi-Group Phase II Clinical Trials*. R package version 1.0.0, URL https://CRAN.R-project.org/package=bacistool.

Dutton P (2017). ***EurosarcBayes**: Bayesian Single Arm Sample Size Calculation Software*. R package version 1.1, URL https://CRAN.R-project.org/package=EurosarcBayes.

Edelmann D (2018). ***hctrial**: Using Historical Controls for Designing Phase II Clinical Trials*. R package version 0.1.0, URL https://CRAN.R-project.org/package=hctrial.

Eggleston BS, Wilson D, McNeil B, Ibrahim JG, Catellier D (2021). ***BayesCTDesign**: Two Arm Bayesian Clinical Trial Design with and Without Historical Control Data*. R package version 0.6.1, URL https://CRAN.R-project.org/package=BayesCTDesign.

Evans S, Ting N (2016). *Fundamental Concepts for New Clinical Trialists*. Chapman & Hall/CRC, Boca Raton.

Gsponer T, Gerber F, Bornkamp B, Ohlssen D, Vandemeulebroecke M, Schmidli H (2014). "A Practical Guide to Bayesian Group Sequential Designs." *Pharmaceutical Statistics*, **13**(1), 71–80. doi:10.1002/pst.1593.

Guo W, Zhong B (2020). ***tsdf**: Two-/Three-Stage Designs for Phase 1&2 Clinical Trials*. R package version 1.1-8, URL https://CRAN.R-project.org/package=tsdf.

Hsu CY, Chen CH (2018). ***SurvGSD**: Group Sequential Design for a Clinical Trial with Censored Survival Data*. R package version 1.0.0, URL https://CRAN.R-project.org/package=SurvGSD.

Ibrahim JG, Chen MH, Chu H (2012). "Bayesian Methods in Clincial Trials: A Bayesian Analysis of ECOG Trials E1684 and E1690." *BMC Medical Research Methodology*, **12**(183), 1–12. doi:10.1186/1471-2288-12-183.

Ibrahim JG, Chen MH, Gwon Y, Chen F (2015). "The Power Prior: Theory and Application." *Statistics in Medicine*, **34**(28), 3724–3749. doi:10.1002/sim.6728.

Ibrahim JG, Chen MH, Sinha D (2001). *Bayesian Survival Analysis*. Springer-Verlag, New York.

Imai K, Jiang Z (2019). ***experiment**: R Package for Designing and Analyzing Randomized Experiments*. R package version 1.2.0, URL https://CRAN.R-project.org/package=experiment.

Izmirlian G (2021). ***PwrGSD**: Power in a Group Sequential Design*. R package version 2.3.3, URL https://CRAN.R-project.org/package=PwrGSD.

Kane MJ, Emerson J, Weston S (2013). "Scalable Strategies for Computing with Massive Data." *Journal of Statistical Software*, **55**(14), 1–19. doi:10.18637/jss.v055.i14.

Kopp-Schneider A, Wiesenfarth M, Abel U (2018). ***BDP2**: Bayesian Adaptive Designs for Phase II Trials with Binary Endpoint*. R package version 0.1.3, URL https://CRAN.R-project.org/package=BDP2.

Kopp-Schneider A, Wiesenfarth M, Ruth W, Edelmann D, Witt O, Abel U (2019). "Monitoring Futility and Efficacy in Phase II Trials with Bayesian Posterior Distributions - a Calibration Approach." *Biometrical Journal*, **61**(3), 488–502. doi:10.1002/bimj.201700209.

Lu P, Liu J, Koestler D (2017). ***pwr2****: Power and Sample Size Analysis for One-Way and Two-Way ANOVA Models*. R package version 1.0, URL https://CRAN.R-project.org/package=pwr2.

Matthews JNS (2006). *Introduction to Randomized Controlled Clinical Trials*. Chapman & Hall/CRC, Boca Raton.

Nagashima K (2018). ***ph2bayes****: Bayesian Single-Arm Phase II Designs*. R package version 0.0.2, URL https://CRAN.R-project.org/package=ph2bayes.

Narasimhan B, Shih MC, He P (2014). ***sp23design****: Design and Simulation of Seamless Phase II-III Clinical Trials*. R package version 0.9, URL https://CRAN.R-project.org/package=sp23design.

Paux G, Dmitrienko A (2018). ***Mediana****: Clinical Trial Simulations*. R package version 1.0.7, URL https://CRAN.R-project.org/package=Mediana.

Plummer M (2003). "**JAGS**: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Technische Universität Wien, Vienna, Austria. URL https://www.R-project.org/conferences/DSC-2003/Proceedings/Plummer.pdf.

Psioda M, Ibrahim JG (2018). "Bayesian Design of a Survival Trial with a Cured Fraction Using Historical Data." *Statistics in Medicine*, **37**(26), 3814–3831. doi:10.1002/sim.7846.

Psioda M, Ibrahim JG (2019). "Bayesian Clinical Trial Design Using Historical Data That Inform the Treatment Effect." *Biostatistics*, **20**(3), 400–415. doi:10.1093/biostatistics/kxy009.

Psioda M, Soukup M, Ibrahim JG (2018). "A Practical Bayesian Adaptive Design Incorporating Data from Historical Controls." *Statistics in Medicine*, **37**(27), 4054–4070. doi:10.1002/sim.7897.

Qi H (2021). ***SampleSize4ClinicalTrials****: Sample Size Calculation for Mean and Proportion Comparisons in Phase 3 Clinical Trials*. R package version 0.2.3, URL https://CRAN.R-project.org/package=SampleSize4ClinicalTrials.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reyes EM, Ghosh SK (2012). ***BAEssd****: Bayesian Average Error Approach to Sample Size Determination*. R package version 1.0.1, URL https://CRAN.R-project.org/package=BAEssd.

Seshan VE (2018). ***clinfun****: Clinical Trial Design and Data Analysis Functions*. R package version 1.0.15, URL https://CRAN.R-project.org/package=clinfun.

Spiegelhalter DJ, Abrams KR, Myles JP (2004). *Bayesian Approaches to Clinical Trials and Health-Care Evaluation*. John Wiley & Sons, West Sussex.

Sweeting M, Mander A, Sabin T (2013). "**bcrm**: Bayesian Continual Reassessment Method Designs for Phase I Dose-Finding Trials." *Journal of Statistical Software*, **54**(13), 1–26. doi:10.18637/jss.v054.i13.

Sweeting M, Wheeler G (2019). ***bcrm***: *Bayesian Continual Reassessment Method for Phase I Dose-Escalation Trials*. R package version 0.5.4, URL https://CRAN.R-project.org/package=bcrm.

Toumazi A, Zohar S, Ursino M (2018). ***dfpk***: *Bayesian Dose-Finding Designs Using Pharmacokinetics (PK) for Phase I Clinical Trials*. R package version 3.5.1, URL https://CRAN.R-project.org/package=dfpk.

Viele K, Berry S, Neuenschwander B, Amzal B, Chen F, Enas N, Hobbs B, Ibrahim JG, Kinnersley N, Lindborg S, Micallef S, Roychoudhury S, Thompson L (2014). "Use of Historical Control Data for Assessing Treatment Effects in Clinical Trials." *Pharmaceutical Statistics*, **13**(1), 41–54. doi:10.1002/pst.1589.

Wallig M, Corporation M, Weston S, Tenenbaum D (2020a). ***doParallel***: *Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.16, URL https://CRAN.R-project.org/package=doParallel.

Wallig M, Microsoft, Weston S (2020b). ***foreach***: *Provides Foreach Looping Construct*. R package version 1.5.1, URL https://CRAN.R-project.org/package=foreach.

Wassmer G, Pahlke F (2019). ***rpact***: *Confirmatory Adaptive Clinical Trial Design and Analysis*. R package version 1.0.8, URL https://CRAN.R-project.org/package=rpact.

Yan F, Zhang L, Zhou Y, Pan H, Liu S, Yuan Y (2020). "**BOIN**: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs." *Journal of Statistical Software*, **94**(13), 1–32. doi:10.18637/jss.v094.i13.

Yin J, Du Y, Mandrekar S (2020). ***phase1RMD***: *Repeated Measurement Design for Phase I Clinical Trial*. R package version 1.0.9, URL https://CRAN.R-project.org/package=phase1RMD.

Zhang E, Wu VQ, Chow SC, GZhang H (2020). ***TrialSize***: *R Functions for Chapter 3, 4, 6, 7, 9, 10, 11, 12, 14, 15 of Sample Size Calculation in Clinical Research*. R package version 1.4, URL https://CRAN.R-project.org/package=TrialSize.

Zhang H, Tang Q (2016). ***BACCT***: *Bayesian Augmented Control for Clinical Trials*. R package version 1.0, URL https://CRAN.R-project.org/package=BACCT.

Zhu Y, Qin R (2016). ***ph2bye***: *Phase II Clinical Trial Design Using Bayesian Methods*. R package version 0.1.4, URL https://CRAN.R-project.org/package=ph2bye.

**Affiliation:**

Barry S. Eggleston
RTI International
3040 East Cornwallis Road
P.O. Box 12194
Research Triangle Park, NC 27709-2194, United States of America
E-mail: beggleston@rti.org

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2021, Volume 100, Issue 21 *Submitted:* 2019-04-01
doi:10.18637/jss.v100.i21 *Accepted:* 2020-09-08