A Bayesian single-arm design using predictive probability monitoring (v0.3)

Replication of Results

2022-02-21

Article Outline

[1 Reference 3](#_Toc96415588)

[2 Study Design 4](#_Toc96415589)

[3 Import R packages 5](#_Toc96415590)

[4 Functions to be used 6](#_Toc96415591)

[4.1 Beta prior 6](#_Toc96415592)

[4.2 Predictive power monitoring from a specified start patient until IA 7](#_Toc96415593)

[4.3 Main function to perform simulation 9](#_Toc96415594)

[4.3.1 Decision boundary from literature 9](#_Toc96415595)

[4.3.2 Decision boundary described in this paper 12](#_Toc96415596)

[5 Figure 2 17](#_Toc96415597)

[6 Figure 3 20](#_Toc96415598)

[7 Table 1 23](#_Toc96415599)

[8 Table 2 28](#_Toc96415600)

[9 Table 3 31](#_Toc96415601)

[9.1 Timing of stop decision: Final or before 31](#_Toc96415602)

[9.2 Timing of stop decision: IA or before 32](#_Toc96415603)

[9.3 Timing of stop decision: Pre IA 32](#_Toc96415604)

[9.4 Overall table 32](#_Toc96415605)

[10 Table 4 37](#_Toc96415606)

[10.1 Timing of stop decision: Final or before 37](#_Toc96415607)

[10.2 Timing of stop decision: IA or before 37](#_Toc96415608)

[10.3 Timing of stop decision: Pre IA 38](#_Toc96415609)

[10.4 Overall table 38](#_Toc96415610)

[11 Table 5 43](#_Toc96415611)

[11.1 NO IA 43](#_Toc96415612)

[11.2 Consistent decision 45](#_Toc96415613)

[11.3 Inconsistent decision 47](#_Toc96415614)

[12 Table 6 49](#_Toc96415615)

[12.1 NO IA 49](#_Toc96415616)

[12.2 Consistent decision 51](#_Toc96415617)

[12.3 Inconsistent decision 53](#_Toc96415618)

[12.4 Inconsistency phenomenon True Effect Means near TV/LRV 55](#_Toc96415619)

[13 Figure 4 55](#_Toc96415620)

[13.1 , sample size equivalent=1 57](#_Toc96415621)

[13.2 , sample size equivalent=20 62](#_Toc96415622)

[13.3 , sample size equivalent=1 67](#_Toc96415623)

[13.4 , sample size equivalent=20 72](#_Toc96415624)

[14 Guidance and Practical Implications 77](#_Toc96415625)

[14.1 Setting 77](#_Toc96415626)

[14.2 TV/LRV and the lack of “PAUSE” results 78](#_Toc96415627)

[14.3 The effect of the timing of beginning predictive probability monitoring 78](#_Toc96415628)

[14.4 Practical implications 79](#_Toc96415629)

[14.5 Multiplicity 79](#_Toc96415630)

# 1 Reference

This document aims to replicate the results from the paper:

*Mitchell PD. A Bayesian single-arm design using predictive probability monitoring. Biom Biostat Int J. 2018;7(4):299-309. DOI: 10.15406/bbij.2018.07.00222*

This design is an additional option for single arm designs where **there is a desire to stop early for futility in a flexible way**.

With this design, fewer patients will be exposed to ineffective compounds than would be exposed using either a single arm trial with either no or a fixed interim.

As with all designs the operating characteristics should be carefully understood before conducting the corresponding experiment. In the case of compounds that show more activity, more patients will be exposed to the compound in anticipation of progressing into later phase development.

**Predictive probability** can be used as an effective means to continuously monitor the chance of success in an ongoing single-arm trial.

# 2 Study Design

This design is a single arm design in which patients either respond to an experimental treatment or not after being followed for a fixed amount of time. Patients will be enrolled in the trial in **four sequential phases**. The overall sample will be N=Na+Nb+Nc+Nd patients.

The responses of the individual patients to treatment will be denoted by Yi (i=1 … N). The chance that patient i responds is . The behavior of is described in the section below on **“True Effect Distribution”**.

Na patients will be enrolled and evaluated unconditionally based on efficacy response. Once the efficacy response is observed in patient Na, trial monitoring using predictive probability including all Na patients begins.

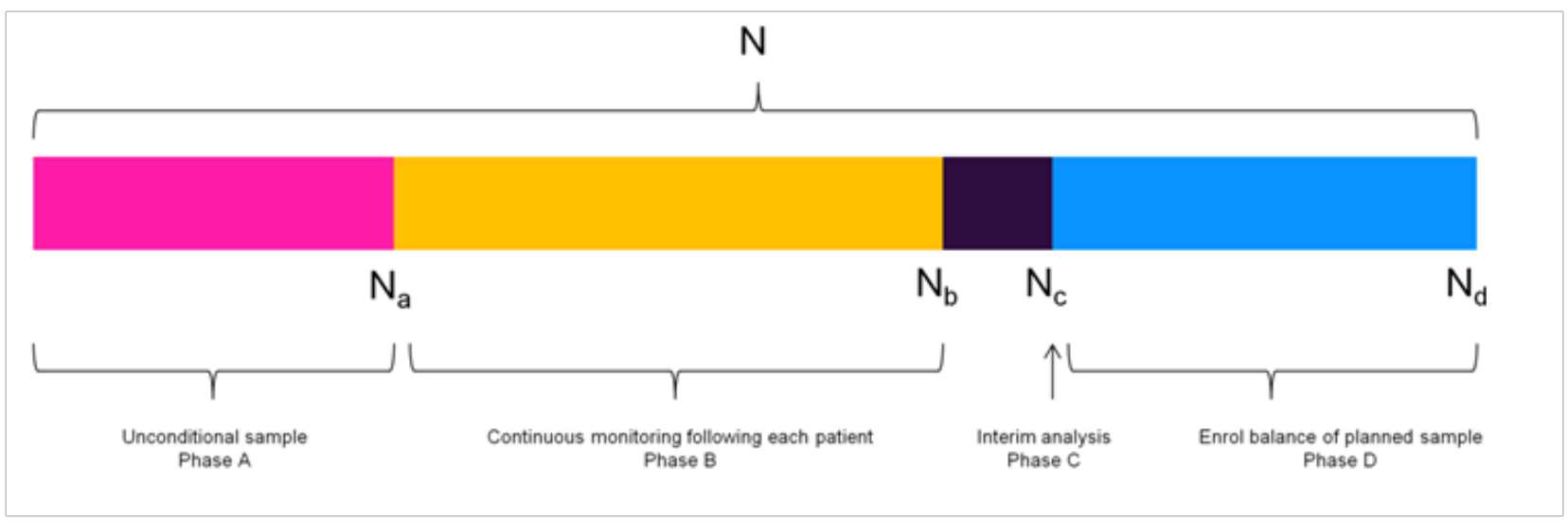
If predictive probability that the proportion of patients responding is at least falls below some level then the trial will be permanently stopped for futility. If the trial is not stopped then the efficacy response for the first patient in Phase b (Patient Na+1) will be observed, predictive probability will then be recalculated for all Na+1 patients and the same criterion will be applied (if predictive probability then continue to the next patient. Otherwise, the trial will be stopped.). The observation of efficacy response in each patient enrolled during Phase b followed by the calculation of predictive probability will continue for all patients in this phase for as long as predictive probability remains above .

If the trial continues through all patients planned in Phase b, then following the observation of the response of the next patient (Phase c), an interim analysis with both futility and administrative facets including all patients up through Phase c will be conducted. This interim will include two criteria:

* a ‘stop’ criterion (S) where the trial will stop if the proportion of patients who respond to treatment is equal to or below this level and
* a ‘go’ (G) criterion where other actions associated with more robust efficacy could be taken.

If the trial does not stop, as a result of the interim analysis, then the balance of the planned Nd patients, will be enrolled and a final analysis including all N patients will be conducted.

*Temporary stopping is not assumed during the trial*. This analysis will be similar to the interim in that the proportion of patients who respond will be evaluated using two criteria (‘STOP’ and ‘GO’).



A schema of study design.

# 3 Import R packages

# load package  
library(ggplot2)  
library(dplyr)  
library(rtables)  
library(knitr)  
library(tidyr)  
library(gridExtra)  
library(reshape)  
library(gcookbook)  
library(flextable)  
library(cowplot)

# 4 Functions to be used

## 4.1 Beta prior

The selection of a prior other than one that is non-informative is beyond the scope of this paper, if there is enough information in patients with the disease under study available that information can be used to construct an empirical prior for the true response proportion.

The number of responses in the sample will follow a Binomial distribution with true response parameter . A natural choice for the True Effect distribution of is Beta with parameters and . These can be selected as functions of the mean proportion of responders and an equivalent sample size **Morita**.

Morita S, Thall PF, Müller P. Determining the Effective Sample Size of a Parametric Prior. Biometrics. 2008;64(2):595–602.

The Nt is **the strength of the True Effect distribution** in terms of a sample size equivalent which will allow for rational values (ranging from 0, equivalent to a Uniform (0,1) to large values which yield distributions where almost all of the density is within a small neighborhood of the mean).

Prop is **the mean of the True Effect distribution**. When evaluating the opearting characteristics of the design, Nt should be no more than 10% of the overall sample size. A range of Prop values (selected to cover the parameter space) are used to assess design performance.

###################  
## Prior  
###################  
Beta\_ab <- function(Nt, Prop) {  
 # Nt : Sample size (numeric > 0, REAL)  
 # Prop : Mean Proportion ( (0,1) )  
 if ((Nt >= 0) & ((Prop >= 0) & (Prop <= 1))) {  
 alpha <- Nt \* Prop  
 beta <- Nt \* (1-Prop)  
 param <- c(alpha, beta)  
 }  
 else  
 param <- NA  
 return(param)  
}

## 4.2 Predictive power monitoring from a specified start patient until IA

Jennison C, Turnbull BW. Group Sequential Methods Applications to Clinical Trials. USA: Chapman & Hall/CRC Press; 2000.

Predictive power is presented in Jennison & Turnbull and is similar to the expression shown below.

For a single arm trial with planned patients and an interim analysis to occur following patients ( ), where of the patients included in the interim respond and we need to observe () patients in order to declare the trial a success then the predictive or the chance to observe at least patients out of patients (at the end of the trial) is:

The important difference is that predictive power implies that the target is a function of a critical value. Predictive probability allows for the selection of a target other than the critical value. The specifics of the application to non-comparative trials with binomial endpoints are described here.

The left hand factor in the expression being summed above is a vector of probabilities that responses are observed assuming that the observed response proportion is actually true.

The right hand factor is the probability that at least the balance of the required responses needed is observed in the remaining patients to be observed in the planned sample.

and ranges from 0 to and is the possible number of successes observed at an interim with total observations.

is the number of succeses that remain following the interim necessary to achieve the target at the end of the trial.

###################  
## Predictive prob  
###################  
  
pred\_power\_bin <- function(N, N\_I, ns, ni) {  
 # N : Total N planned in sample  
 # N\_I : N (sample size) at interim  
 # ns : Number of success needed in total sample  
 # ni : Number of successes observed at interim  
   
 ISS <- 0:N\_I # Interim sample space  
 LIK <- dbinom(ISS, N\_I, ni / N\_I, log = FALSE) # Likelihood of interim result given interim observed result  
   
   
 mid\_e <- length(ISS)  
 mid <- ISS / N\_I  
   
 # Exception when ISS=0  
 mid[1] <- ifelse(mid[1] == 0, 0.5 / N\_I, ISS / N\_I)  
 # Exception when ISS=N\_I  
 mid[mid\_e] <- (ISS[mid\_e] - 0.5) / N\_I  
   
 #Prob of Success  
 POS <-  
 1 - pbinom((ns - ni) - 1,  
 N - N\_I ,  
 mid,  
 lower.tail = TRUE,  
 log.p = FALSE)  
   
 # probability of success at final given interim over all interim results  
 assu <- LIK \* POS  
   
 return(round(sum(assu), 2))  
}

## 4.3 Main function to perform simulation

Before introducing the simulation function, the strategy of making decisions should be summarized.

A question may be raised firstly: is beta(1,1) using for calculating the posterior probability of response rate when making decisions? Take beta(1,1) as the prior distribution, the posterior distribution for response rate in final analysis is calculated as beta(1+k,1+N-k), where k is the number of responses and N is the total sample size.

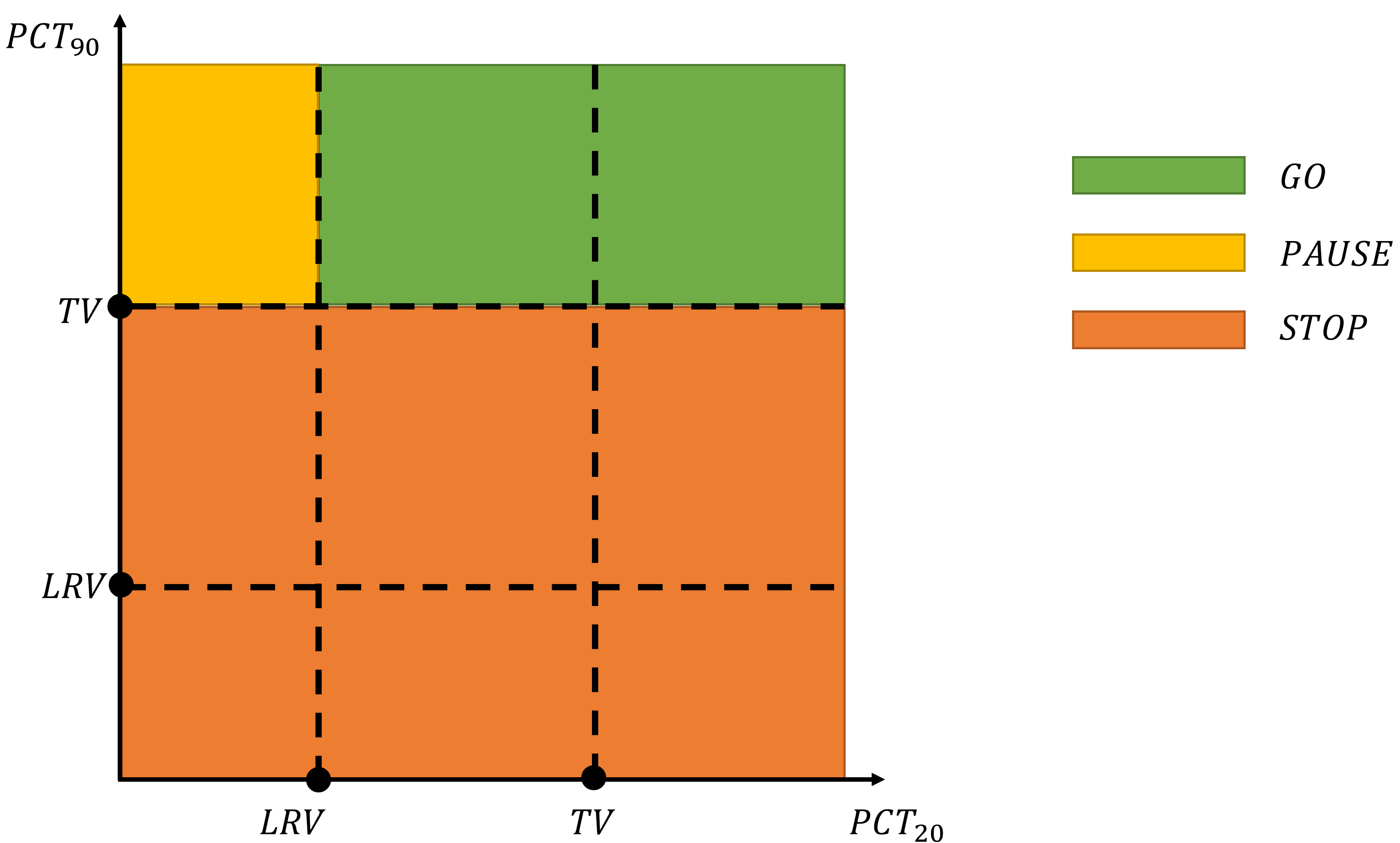
For the decision boundaries, for example, can be used to construct rules for advancing compound development.

### 4.3.1 Decision boundary from literature

For example, **80% confidence that p>LRV** might be desired to limit the risk of progressing a compound with relatively little efficacy.

The drug development team will also desire a compound that is commercially viable (**p>TV**). To avoid terminating such a compound, **10% risk** might be chosen.

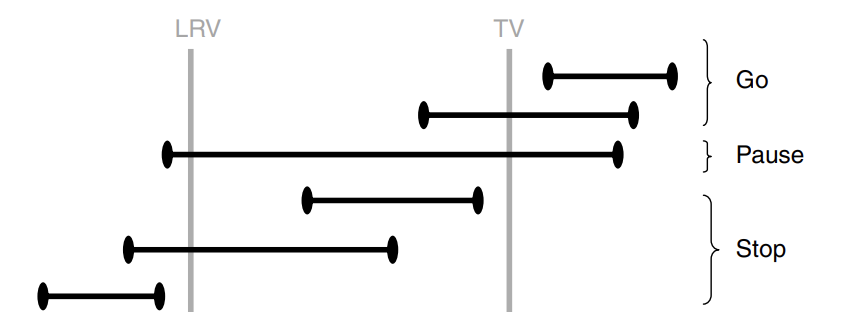
* Go decision if : PCT20>LRV and PCT90>TV
* Pause if : PCT20<=LRV and PCT90>TV
* Stop if : PCT90<=TV



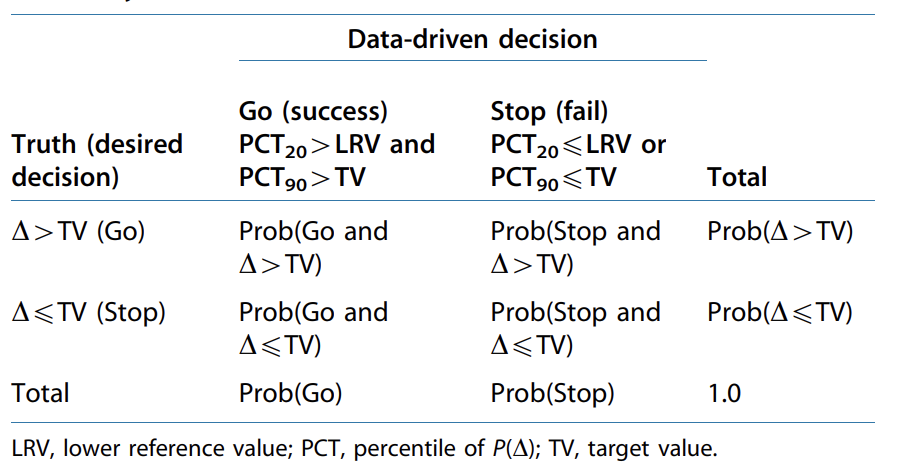
An illustration of decision making

where PCT denotes the -th percentile of (response rate).

The dual criteria can be used to formulate a **Go/Pause/Stop** decision structure as follows:



Example of a decision rule based on the dual criteria. The criteria and decisions will vary depending on the target responses that are important for a particular compound and the phase of clinical development.



An example of a design performance (trial metrics) summary

### 4.3.2 Decision boundary described in this paper

The decision criteria used in the design described in this paper are not set directly but are **a function of two values** selected by the project and are a reflection of the performance necessary to support decisions to go forward in development or to stop development.

These are the **target value (TV)** and **lower reference value (LRV)**.

In this design:

1. A GO decision will be made if

* the posterior probability that the true response proportion is at least LRV is at least DC\_LRV.

1. A STOP decision is made if

* the posterior probability that the true response proportion is at least the TV is no more than AR\_TV.

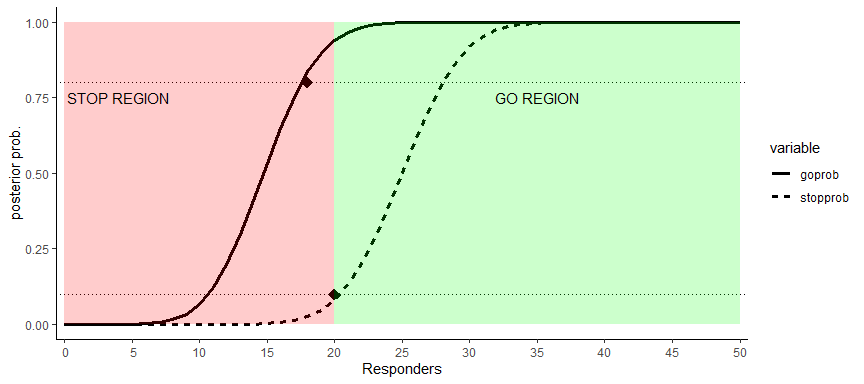
(3)\*In cases where **both conditions are true (a Go decision and a Stop decision are made simultaneously)**, a STOP decision is made.

Here the DC\_LRV is the **desired confidence** that the true response proportion is at least as high as the LRV given a GO decision. The AR\_TV is the **acceptable risk** that the true response proportion is at least as high as the TV given a decision to STOP.

###################  
## Simulation  
###################  
  
trials <- function(Nt,  
 N\_i,  
 N\_p,  
 true\_1,  
 tv,  
 lrv,  
 dc\_lrv,  
 ar\_tv,  
 Tar, # Proportion of responders to be observed at final that will lead to a non-stop decision  
 strt, # Begin predictive power calculation with this observation  
 stop, # Last patient to monitor via predictive power  
 PPmon ) { # Minimum predicted power needed to continue arm  
   
 # Sample response rate from prior  
 Bpar1 <- Beta\_ab(N\_p, true\_1)  
 theta\_1 <- rbeta(1, Bpar1[1] + 1, Bpar1[2] + 1)  
   
 # Generate simulated data for total N patients  
 arm1 <- rbinom(Nt, 1, theta\_1)  
   
 # Take the first N\_i data from the complete simulated data arm1 as the data for IA   
 int\_arm1 <- arm1[1:N\_i]  
   
 #########################################################################  
   
 # set data container for storing the results of PP monitoring  
 PP1 <- vector(mode = "numeric", length = stop - strt)  
   
 # set data container for storing the number of patients when monitoring PP  
 PP1\_N <- vector(mode = "numeric", length = stop - strt)  
   
 # set data container for recording the step when monitoring PP  
 h <- vector(mode = "numeric", length = stop - strt)  
   
 for (g in 1:(stop - strt)) {  
 h[g] <- g  
 PP1\_N[g] <- strt + (g - 1)  
 PP1[g] <-  
 pred\_power\_bin(Nt, strt + (g - 1), round(Nt \* Tar, 0), max(sum(int\_arm1[1:(strt + (g - 1))])))  
 }  
   
 # PP || #pts when doing this PP  
 pptab <- data.frame(PP1, PP1\_N)  
   
 # Store the results of NOT to continue arm  
 ppck1 <- pptab[pptab$PP1 < PPmon, ]  
   
 # exceptions  
 ppN1 <- ifelse(length(ppck1$PP1\_N) == 0, 0, min(ppck1$PP1\_N))  
   
 # records how many patients WIN before futility stopping during PP monitoring  
 # a1NPP: the number of patients before NO Continue decision is made  
 if (min(pptab$PP1) >= PPmon){  
 a1NPP <- N\_i   
 } else {  
 a1NPP <- ppN1  
 }  
   
 #########################################################################  
   
 # calculate the REAL number of corresponders among all patients  
 sc1 <- sum(arm1)  
   
 # calculate the REAL number of NON-corresponders among all patients  
 fa1 <- Nt - sc1  
   
 # calculate the REAL number of corresponders among patients in Interim Analysis  
 isc1 <- sum(int\_arm1)  
   
 # calculate the REAL number of NON-corresponders among patients in Interim Analysis  
 ifa1 <- N\_i - isc1  
   
 #########################################################################  
   
 # Decide  
   
 # For final analysis  
 ### 1 for GO, 2 for STOP  
 arm1\_go <- pbeta(lrv, 1 + sc1, 1 + fa1, lower.tail = FALSE)  
 arm1\_stop <- pbeta(tv, 1 + sc1, 1 + fa1, lower.tail = FALSE)  
 FA\_GO\_FLAG <- ifelse(arm1\_go >= dc\_lrv, 1,0)  
 FA\_STOP\_FLAG <- ifelse(arm1\_stop <= ar\_tv, 2,0)  
 arm1\_rag <- ifelse(FA\_GO\_FLAG\*FA\_STOP\_FLAG==0,max(FA\_GO\_FLAG,FA\_STOP\_FLAG),2)  
   
 # For interim analysis  
 iarm1\_go <- pbeta(lrv, 1 + isc1, 1 + ifa1, lower.tail = FALSE)  
 iarm1\_stop <- pbeta(tv, 1 + isc1, 1 + ifa1, lower.tail = FALSE)  
 IA\_GO\_FLAG <- ifelse(iarm1\_go >= dc\_lrv, 1,0)  
 IA\_STOP\_FLAG <- ifelse(iarm1\_stop <= ar\_tv, 2,0)  
 iarm1\_rag <- ifelse(IA\_GO\_FLAG\*IA\_STOP\_FLAG==0,max(IA\_GO\_FLAG,IA\_STOP\_FLAG),2)  
   
 #########################################################################  
   
 # Summarize the results   
   
 ## For interim analysis:  
 ## if trial early stop since STOP is made in IA, record the sample size for IA  
 a1N <-  
 ifelse(iarm1\_rag == 2, ifelse(a1NPP < N\_i, a1NPP, N\_i), Nt)  
 tot\_N <- a1N  
 pre\_int1 <- ifelse(a1NPP < N\_i, "STOP", "GO")  
 p <- true\_1  
 ne <- N\_p  
   
 # change flag to char  
 FA\_FL <- recode(arm1\_rag, `2` = "STOP", `1` = "GO")  
 IA\_FL <- recode(iarm1\_rag, `2` = "STOP", `1` = "GO")  
   
 # Package results  
 tr <-  
 data.frame(  
 ne,  
 p,  
 sc1, # the REAL number of corresponders among all patients  
 fa1, # the REAL number of NON-corresponders among all patients  
 arm1\_go, # boundary for GO at FA  
 arm1\_stop, # boundary for STOP at FA  
 FA\_FL, # Decision at FA  
 isc1, # the REAL number of corresponders among patients in IA  
 ifa1, # the REAL number of NON-corresponders among patients in IA  
 iarm1\_go, # boundary for GO at IA  
 iarm1\_stop, # boundary for STOP at IA  
 IA\_FL, # Decision at IA  
 a1NPP, # the number of patients before NO Continue decision is made  
 pre\_int1, # indicate if trial stops before IA  
 a1N, # the sample size when trial stops  
 tot\_N # the same as a1N  
 )  
   
 return(tr)  
}

# 5 Figure 2

ss <- 50  
responders <- seq(0,ss,by=1)  
lrv <- 0.3  
tv <- 0.5  
  
# function to calculate posterior probability  
PostProb <- function(x, boundary,ss){  
 return(pbeta(boundary, 1 + x, 1 + ss-x, lower.tail = FALSE))  
}  
  
##############  
## Figure 2  
##############  
  
# create data frame container  
PostProbDF1 <-  
 data.frame(  
 responders = responders,  
 goprob = rep(NA, length(responders)),  
 stopprob = rep(NA, length(responders))  
 )  
  
# retrieve results  
for (index in 1:length(responders)){  
 PostProbDF1[index,"goprob"] <- PostProb(PostProbDF1[index,"responders"],lrv,ss)  
 PostProbDF1[index,"stopprob"] <- PostProb(PostProbDF1[index,"responders"],tv,ss)  
}  
  
# reshape data to plot  
PostProbDF1\_t <- melt(PostProbDF1,id=c("responders"))  
  
# Plot  
CritPlot1 <- ggplot(PostProbDF1\_t, aes(x=responders, y=value, group=variable)) +  
 geom\_line(aes(linetype=variable),size=1.2,color="black")+  
 geom\_hline(yintercept = c(0.1,0.8),linetype=3) +   
 geom\_point(aes(x=18,y=0.8),colour="black",size=4,shape=18) +  
 geom\_point(aes(x=20,y=0.1),colour="black",size=4,shape=18) +  
 annotate("rect", xmin = 20, xmax = ss, ymin = 0, ymax = 1,alpha = .2,fill = "green") +  
 annotate("rect", xmin = 0, xmax = 20, ymin = 0, ymax = 1,alpha = .2,fill = "red") +  
 annotate(geom="text", x=4, y=0.75, label="STOP REGION",color="black") +  
 annotate(geom="text", x=35, y=0.75, label="GO REGION",color="black") +  
 scale\_x\_discrete(name ="Responders", limits=seq(0,ss,by=5)) +  
 scale\_y\_continuous(name="posterior prob.", limits=c(0, 1))+  
 theme\_classic()  
  
CritPlot1



Then we want to find the correponding number of responders for GO/STOP decision:

# find the number of responders for GO decision  
PostProbDF1\_t\_GO <- PostProbDF1\_t %>%  
 filter(variable=="goprob" & value>0.8)  
  
min\_go\_resp <- PostProbDF1\_t\_GO[1,"responders"]  
cat("The minimum responders for GO is: ", min\_go\_resp, "\n")

## The minimum responders for GO is: 18

# find the number of responders for STOP decision  
PostProbDF1\_t\_STOP <- PostProbDF1\_t %>%  
 filter(variable=="stopprob" & value<=0.1)  
  
max\_stop\_resp <- PostProbDF1\_t\_STOP[length(PostProbDF1\_t\_STOP$responders),"responders"]  
cat("The maximum responders for STOP is: ", max\_stop\_resp, "\n")

## The maximum responders for STOP is: 20

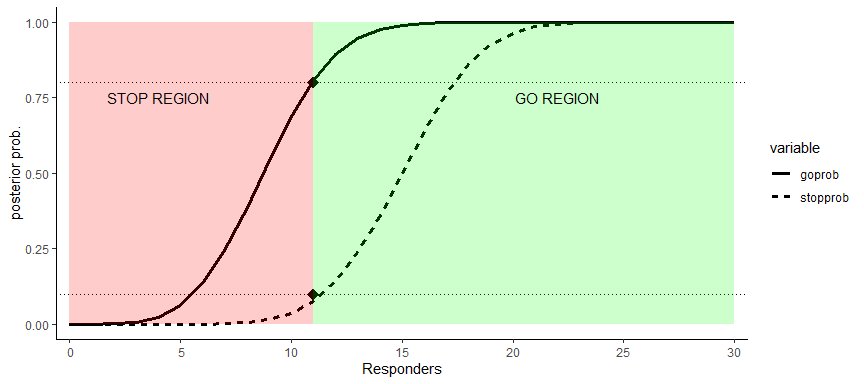
If the number of responders , then a GO decision will be made.

If the number of responders , then a STOP decision will be made.

In this case, **a STOP decision will be made** if the number of responders falls in the interval based on the decision making strategy described in the above section.

# 6 Figure 3

##############  
## Figure 3  
##############  
  
ss <- 30  
responders <- seq(0,ss,by=1)  
  
# create data frame container  
PostProbDF2 <-  
 data.frame(  
 responders = responders,  
 goprob = rep(NA, length(responders)),  
 stopprob = rep(NA, length(responders))  
 )  
  
# retrieve results  
for (index in 1:length(responders)){  
 PostProbDF2[index,"goprob"] <- PostProb(PostProbDF2[index,"responders"],lrv,ss)  
 PostProbDF2[index,"stopprob"] <- PostProb(PostProbDF2[index,"responders"],tv,ss)  
}  
  
# reshape data to plot  
PostProbDF2\_t <- melt(PostProbDF2,id=c("responders"))  
  
# Plot  
CritPlot2 <- ggplot(PostProbDF2\_t, aes(x=responders, y=value, group=variable)) +  
 geom\_line(aes(linetype=variable),size=1.2,color="black")+  
 geom\_hline(yintercept = c(0.1,0.8),linetype=3) +   
 geom\_point(aes(x=11,y=0.8),colour="black",size=4,shape=18) +  
 geom\_point(aes(x=11,y=0.1),colour="black",size=4,shape=18) +  
 annotate("rect", xmin = 11, xmax = ss, ymin = 0, ymax = 1,alpha = .2,fill = "green") +  
 annotate("rect", xmin = 0, xmax = 11, ymin = 0, ymax = 1,alpha = .2,fill = "red") +  
 annotate(geom="text", x=4, y=0.75, label="STOP REGION",color="black") +  
 annotate(geom="text", x=22, y=0.75, label="GO REGION",color="black") +  
 scale\_x\_discrete(name ="Responders", limits=seq(0,ss,by=5))+  
 scale\_y\_continuous(name="posterior prob.", limits=c(0, 1))+  
 theme\_classic()  
  
CritPlot2



Then we want to find the correponding number of responders for GO/STOP decision:

# find the number of responders for GO decision  
PostProbDF2\_t\_GO <- PostProbDF2\_t %>%  
 filter(variable=="goprob" & value>0.8)  
  
min\_go\_resp <- PostProbDF2\_t\_GO[1,"responders"]  
cat("The minimum responders for GO is: ", min\_go\_resp, "\n")

## The minimum responders for GO is: 11

# find the number of responders for STOP decision  
PostProbDF2\_t\_STOP <- PostProbDF2\_t %>%  
 filter(variable=="stopprob" & value<=0.1)  
  
max\_stop\_resp <- PostProbDF1\_t\_STOP[length(PostProbDF2\_t\_STOP$responders),"responders"]  
cat("The maximum responders for STOP is: ", max\_stop\_resp, "\n")

## The maximum responders for STOP is: 11

If the number of responders , then a GO decision will be made.

If the number of responders , then a STOP decision will be made.

In this case, **a STOP decision will be made** if the number of responders equals to based on the decision making strategy described in the above section.

# 7 Table 1

# set seed  
set.seed(125)  
  
# Target value  
TV <- 0.5   
  
# Lower reference value  
LRV <- 0.3   
  
# Range of true responses to be explored (combination) in increasing order  
True\_range <- c(0.8, 0.6, 0.4, 0.2)   
  
# Strength of prior as a sample size equivalent  
PN <- c(0,0.5,1,5,10,20)  
  
# Planned total sample size  
N <- 50   
  
# Planned interim sample size  
Ni <- 30   
  
# Number of simulated trials  
iter <- 4000   
  
# Minimum predicted power needed to continue arm  
MON\_stop <- 0.05   
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
## Target <- 20 / 50   
Target <- 0.52   
  
# Last patient to monitor via predictive power  
PP\_stop <- 29   
  
# Begin predictive power calculation with this observation  
PP\_start <- 12   
  
# Go: at least 80% chance that the true is greater than the LRV based on the data  
DC\_LRV <- 0.8  
  
# Stop: no more than 10% chance that the true is greater than the TV based on the data  
AR\_TV <- 0.1  
  
# Initiate parameters/data containers to be used in simulations  
LenTrueResp <- length(True\_range)  
  
SimDataTab1 <-  
 data.frame(matrix(NA, length(PN) \* length(True\_range) \* iter, 16)) # create matrix container  
  
a <- 0  
row <- 1  
  
if (file.exists("SimDataTab1.RData")) {  
 load("SimDataTab1.RData")  
} else {  
 for (ne in (PN)) {  
 for (i in 1:LenTrueResp) {  
 # loop through different REAL response rate  
 a <- a + 1  
 for (z in 1:iter) {  
 # loop to generate multiple times of trial  
 sim\_t <-  
 trials(  
 N,  
 Ni,  
 ne,  
 True\_range[i],  
 TV,  
 LRV,  
 DC\_LRV,  
 AR\_TV,  
 Target,  
 PP\_start,  
 PP\_stop,  
 MON\_stop  
 )  
 SimDataTab1[row, 1:16] <- sim\_t[1,]  
 row <- row + 1  
 }  
 }  
 }  
 colnames(SimDataTab1) <- colnames(sim\_t)  
 save(SimDataTab1, file = "SimDataTab1.RData")  
 load("SimDataTab1.RData")  
}  
  
Table1DFMeanSample <- SimDataTab1 %>%  
 group\_by(ne, p) %>%  
 summarise(mean = mean(a1N))  
  
Table1DFProp <- SimDataTab1 %>%  
 group\_by(ne, p, FA\_FL) %>%  
 summarise(cnt = n()) %>%  
 mutate(freq = round(100\*cnt/iter,3)) %>%   
 select(-c(cnt)) %>%  
 spread(FA\_FL, freq) %>%  
 arrange(ne,desc(p))  
  
Table1 <- Table1DFMeanSample %>%  
 inner\_join(Table1DFProp, by = c("ne","p")) %>%  
 arrange(ne,desc(p))  
  
flextable(Table1) %>%  
 FitFlextableToPage()

| ne | p | mean | GO | STOP |
| --- | --- | --- | --- | --- |
| 0.0 | 0.8 | 35.81275 | 59.225 | 40.775 |
| 0.0 | 0.6 | 35.79075 | 59.175 | 40.825 |
| 0.0 | 0.4 | 35.85800 | 58.350 | 41.650 |
| 0.0 | 0.2 | 35.79200 | 58.725 | 41.275 |
| 0.5 | 0.8 | 39.04450 | 67.075 | 32.925 |
| 0.5 | 0.6 | 37.14850 | 62.150 | 37.850 |
| 0.5 | 0.4 | 34.88125 | 55.650 | 44.350 |
| 0.5 | 0.2 | 33.09600 | 50.950 | 49.050 |
| 1.0 | 0.8 | 41.45075 | 73.500 | 26.500 |
| 1.0 | 0.6 | 38.90525 | 66.600 | 33.400 |
| 1.0 | 0.4 | 35.24575 | 56.075 | 43.925 |
| 1.0 | 0.2 | 31.19750 | 45.950 | 54.050 |
| 5.0 | 0.8 | 48.41575 | 94.975 | 5.025 |
| 5.0 | 0.6 | 43.66650 | 79.900 | 20.100 |
| 5.0 | 0.4 | 34.88825 | 54.325 | 45.675 |
| 5.0 | 0.2 | 23.59450 | 23.225 | 76.775 |
| 10.0 | 0.8 | 49.58125 | 98.775 | 1.225 |
| 10.0 | 0.6 | 45.67225 | 86.150 | 13.850 |
| 10.0 | 0.4 | 34.38025 | 49.725 | 50.275 |
| 10.0 | 0.2 | 19.80475 | 12.875 | 87.125 |
| 20.0 | 0.8 | 49.95800 | 99.850 | 0.150 |
| 20.0 | 0.6 | 47.55200 | 91.625 | 8.375 |
| 20.0 | 0.4 | 34.91675 | 50.000 | 50.000 |
| 20.0 | 0.2 | 17.17400 | 5.600 | 94.400 |

# 8 Table 2

# set seed  
set.seed(125)  
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
Target <- 0.40   
  
# Initiate parameters/data containers to be used in simulations  
SimDataTab2 <-  
 data.frame(matrix(NA, length(PN) \* length(True\_range) \* iter, 16)) # create matrix container  
  
a <- 0  
row <- 1  
  
if (file.exists("SimDataTab2.RData")) {  
 load("SimDataTab2.RData")  
} else {  
 for (ne in (PN)) {  
 for (i in 1:LenTrueResp) {  
 # loop through different REAL response rate  
 a <- a + 1  
 for (z in 1:iter) {  
 # loop to generate multiple times of trial  
 sim\_t <-  
 trials(  
 N,  
 Ni,  
 ne,  
 True\_range[i],  
 TV,  
 LRV,  
 DC\_LRV,  
 AR\_TV,  
 Target,  
 PP\_start,  
 PP\_stop,  
 MON\_stop  
 )  
 SimDataTab2[row, 1:16] <- sim\_t[1,]  
 row <- row + 1  
 }  
 }  
 }  
 colnames(SimDataTab2) <- colnames(sim\_t)  
 save(SimDataTab2, file = "SimDataTab2.RData")  
 load("SimDataTab2.RData")  
}  
  
Table2DFMeanSample <- SimDataTab2 %>%  
 group\_by(ne, p) %>%  
 summarise(mean = mean(a1N))  
  
Table2DFProp <- SimDataTab2 %>%  
 group\_by(ne, p, FA\_FL) %>%  
 summarise(cnt = n()) %>%  
 mutate(freq = round(100\*cnt/iter,3)) %>%   
 select(-c(cnt)) %>%  
 spread(FA\_FL, freq) %>%  
 arrange(ne,desc(p))  
  
Table2 <- Table2DFMeanSample %>%  
 inner\_join(Table2DFProp, by = c("ne","p")) %>%  
 arrange(ne,desc(p))  
  
flextable(Table2) %>%  
 FitFlextableToPage()

| ne | p | mean | GO | STOP |
| --- | --- | --- | --- | --- |
| 0.0 | 0.8 | 37.00125 | 59.225 | 40.775 |
| 0.0 | 0.6 | 37.03625 | 59.175 | 40.825 |
| 0.0 | 0.4 | 37.08175 | 58.350 | 41.650 |
| 0.0 | 0.2 | 36.95525 | 58.725 | 41.275 |
| 0.5 | 0.8 | 40.16425 | 67.075 | 32.925 |
| 0.5 | 0.6 | 38.40975 | 62.150 | 37.850 |
| 0.5 | 0.4 | 36.37225 | 55.650 | 44.350 |
| 0.5 | 0.2 | 34.52375 | 50.950 | 49.050 |
| 1.0 | 0.8 | 42.51750 | 73.500 | 26.500 |
| 1.0 | 0.6 | 40.15075 | 66.600 | 33.400 |
| 1.0 | 0.4 | 36.70725 | 56.075 | 43.925 |
| 1.0 | 0.2 | 32.86850 | 45.950 | 54.050 |
| 5.0 | 0.8 | 48.72875 | 94.975 | 5.025 |
| 5.0 | 0.6 | 44.74025 | 79.900 | 20.100 |
| 5.0 | 0.4 | 36.93575 | 54.325 | 45.675 |
| 5.0 | 0.2 | 25.92500 | 23.225 | 76.775 |
| 10.0 | 0.8 | 49.68075 | 98.775 | 1.225 |
| 10.0 | 0.6 | 46.52775 | 86.150 | 13.850 |
| 10.0 | 0.4 | 36.88875 | 49.725 | 50.275 |
| 10.0 | 0.2 | 22.40900 | 12.875 | 87.125 |
| 20.0 | 0.8 | 49.96850 | 99.850 | 0.150 |
| 20.0 | 0.6 | 48.13375 | 91.625 | 8.375 |
| 20.0 | 0.4 | 37.60800 | 50.000 | 50.000 |
| 20.0 | 0.2 | 19.99775 | 5.600 | 94.400 |

# 9 Table 3

## 9.1 Timing of stop decision: Final or before

Table3DF\_1 <- SimDataTab1 %>%   
 mutate(Timing="Final",NOIAfl=ifelse(a1N<30,1,0),IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table3DF\_1\_a <- Table3DF\_1 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(NOIA = round(sum(NOIAfl)/iter,3),IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 9.2 Timing of stop decision: IA or before

Table3DF\_2 <- SimDataTab1 %>%   
 filter((IA\_FL=="STOP")|(pre\_int1=="STOP")) %>%  
 mutate(Timing="IA",IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table3DF\_2\_a <- Table3DF\_2 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 9.3 Timing of stop decision: Pre IA

Table3DF\_3 <- SimDataTab1 %>%   
 filter(pre\_int1=="STOP") %>%  
 mutate(Timing="Pre",IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table3DF\_3\_a <- Table3DF\_3 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 9.4 Overall table

Table3 <- Table3DF\_1\_a %>%  
 bind\_rows(Table3DF\_2\_a,Table3DF\_3\_a) %>%  
 arrange(Timing,ne,desc(p))  
  
flextable(Table3) %>%  
 FitFlextableToPage()

| Timing | ne | p | NOIA | IAPP |
| --- | --- | --- | --- | --- |
| Final | 0.0 | 0.8 | 0.386 | 0.450 |
| Final | 0.0 | 0.6 | 0.388 | 0.459 |
| Final | 0.0 | 0.4 | 0.384 | 0.454 |
| Final | 0.0 | 0.2 | 0.386 | 0.463 |
| Final | 0.5 | 0.8 | 0.301 | 0.377 |
| Final | 0.5 | 0.6 | 0.351 | 0.430 |
| Final | 0.5 | 0.4 | 0.413 | 0.496 |
| Final | 0.5 | 0.2 | 0.459 | 0.540 |
| Final | 1.0 | 0.8 | 0.236 | 0.319 |
| Final | 1.0 | 0.6 | 0.305 | 0.387 |
| Final | 1.0 | 0.4 | 0.402 | 0.493 |
| Final | 1.0 | 0.2 | 0.513 | 0.595 |
| Final | 5.0 | 0.8 | 0.045 | 0.085 |
| Final | 5.0 | 0.6 | 0.180 | 0.266 |
| Final | 5.0 | 0.4 | 0.419 | 0.546 |
| Final | 5.0 | 0.2 | 0.718 | 0.819 |
| Final | 10.0 | 0.8 | 0.012 | 0.037 |
| Final | 10.0 | 0.6 | 0.124 | 0.215 |
| Final | 10.0 | 0.4 | 0.438 | 0.599 |
| Final | 10.0 | 0.2 | 0.819 | 0.909 |
| Final | 20.0 | 0.8 | 0.001 | 0.012 |
| Final | 20.0 | 0.6 | 0.072 | 0.160 |
| Final | 20.0 | 0.4 | 0.430 | 0.638 |
| Final | 20.0 | 0.2 | 0.891 | 0.964 |
| IA | 0.0 | 0.8 |  | 0.435 |
| IA | 0.0 | 0.6 |  | 0.447 |
| IA | 0.0 | 0.4 |  | 0.437 |
| IA | 0.0 | 0.2 |  | 0.445 |
| IA | 0.5 | 0.8 |  | 0.359 |
| IA | 0.5 | 0.6 |  | 0.414 |
| IA | 0.5 | 0.4 |  | 0.478 |
| IA | 0.5 | 0.2 |  | 0.522 |
| IA | 1.0 | 0.8 |  | 0.299 |
| IA | 1.0 | 0.6 |  | 0.368 |
| IA | 1.0 | 0.4 |  | 0.470 |
| IA | 1.0 | 0.2 |  | 0.577 |
| IA | 5.0 | 0.8 |  | 0.078 |
| IA | 5.0 | 0.6 |  | 0.248 |
| IA | 5.0 | 0.4 |  | 0.520 |
| IA | 5.0 | 0.2 |  | 0.792 |
| IA | 10.0 | 0.8 |  | 0.034 |
| IA | 10.0 | 0.6 |  | 0.199 |
| IA | 10.0 | 0.4 |  | 0.561 |
| IA | 10.0 | 0.2 |  | 0.886 |
| IA | 20.0 | 0.8 |  | 0.012 |
| IA | 20.0 | 0.6 |  | 0.146 |
| IA | 20.0 | 0.4 |  | 0.595 |
| IA | 20.0 | 0.2 |  | 0.941 |
| Pre | 0.0 | 0.8 |  | 0.435 |
| Pre | 0.0 | 0.6 |  | 0.447 |
| Pre | 0.0 | 0.4 |  | 0.437 |
| Pre | 0.0 | 0.2 |  | 0.445 |
| Pre | 0.5 | 0.8 |  | 0.359 |
| Pre | 0.5 | 0.6 |  | 0.414 |
| Pre | 0.5 | 0.4 |  | 0.478 |
| Pre | 0.5 | 0.2 |  | 0.522 |
| Pre | 1.0 | 0.8 |  | 0.299 |
| Pre | 1.0 | 0.6 |  | 0.368 |
| Pre | 1.0 | 0.4 |  | 0.470 |
| Pre | 1.0 | 0.2 |  | 0.577 |
| Pre | 5.0 | 0.8 |  | 0.078 |
| Pre | 5.0 | 0.6 |  | 0.248 |
| Pre | 5.0 | 0.4 |  | 0.520 |
| Pre | 5.0 | 0.2 |  | 0.792 |
| Pre | 10.0 | 0.8 |  | 0.034 |
| Pre | 10.0 | 0.6 |  | 0.199 |
| Pre | 10.0 | 0.4 |  | 0.561 |
| Pre | 10.0 | 0.2 |  | 0.886 |
| Pre | 20.0 | 0.8 |  | 0.012 |
| Pre | 20.0 | 0.6 |  | 0.146 |
| Pre | 20.0 | 0.4 |  | 0.595 |
| Pre | 20.0 | 0.2 |  | 0.941 |

# 10 Table 4

## 10.1 Timing of stop decision: Final or before

Table4DF\_1 <- SimDataTab2 %>%   
 mutate(Timing="Final",NOIAfl=ifelse(a1N<30,1,0),IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table4DF\_1\_a <- Table4DF\_1 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(NOIA = round(sum(NOIAfl)/iter,3),IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 10.2 Timing of stop decision: IA or before

Table4DF\_2 <- SimDataTab2 %>%   
 filter((IA\_FL=="STOP")|(pre\_int1=="STOP")) %>%  
 mutate(Timing="IA",IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table4DF\_2\_a <- Table4DF\_2 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 10.3 Timing of stop decision: Pre IA

Table4DF\_3 <- SimDataTab2 %>%   
 filter(pre\_int1=="STOP") %>%  
 mutate(Timing="Pre",IAPPfl=ifelse((FA\_FL=="STOP")|(IA\_FL=="STOP")|(pre\_int1=="STOP"),1,0))  
  
Table4DF\_3\_a <- Table4DF\_3 %>%  
 group\_by(Timing,ne, p) %>%  
 summarise(IAPP = round(sum(IAPPfl)/iter,3)) %>%  
 arrange(Timing,ne,desc(p))

## 10.4 Overall table

Table4 <- Table4DF\_1\_a %>%  
 bind\_rows(Table4DF\_2\_a,Table4DF\_3\_a) %>%  
 arrange(Timing,ne,desc(p))  
  
flextable(Table4) %>%  
 FitFlextableToPage()

| Timing | ne | p | NOIA | IAPP |
| --- | --- | --- | --- | --- |
| Final | 0.0 | 0.8 | 0.333 | 0.421 |
| Final | 0.0 | 0.6 | 0.328 | 0.424 |
| Final | 0.0 | 0.4 | 0.330 | 0.426 |
| Final | 0.0 | 0.2 | 0.333 | 0.426 |
| Final | 0.5 | 0.8 | 0.245 | 0.344 |
| Final | 0.5 | 0.6 | 0.290 | 0.395 |
| Final | 0.5 | 0.4 | 0.345 | 0.460 |
| Final | 0.5 | 0.2 | 0.395 | 0.508 |
| Final | 1.0 | 0.8 | 0.183 | 0.280 |
| Final | 1.0 | 0.6 | 0.245 | 0.350 |
| Final | 1.0 | 0.4 | 0.334 | 0.456 |
| Final | 1.0 | 0.2 | 0.436 | 0.561 |
| Final | 5.0 | 0.8 | 0.026 | 0.061 |
| Final | 5.0 | 0.6 | 0.119 | 0.224 |
| Final | 5.0 | 0.4 | 0.320 | 0.488 |
| Final | 5.0 | 0.2 | 0.623 | 0.789 |
| Final | 10.0 | 0.8 | 0.006 | 0.019 |
| Final | 10.0 | 0.6 | 0.072 | 0.163 |
| Final | 10.0 | 0.4 | 0.314 | 0.535 |
| Final | 10.0 | 0.2 | 0.717 | 0.883 |
| Final | 20.0 | 0.8 | 0.000 | 0.002 |
| Final | 20.0 | 0.6 | 0.033 | 0.106 |
| Final | 20.0 | 0.4 | 0.278 | 0.548 |
| Final | 20.0 | 0.2 | 0.784 | 0.951 |
| IA | 0.0 | 0.8 |  | 0.390 |
| IA | 0.0 | 0.6 |  | 0.392 |
| IA | 0.0 | 0.4 |  | 0.389 |
| IA | 0.0 | 0.2 |  | 0.392 |
| IA | 0.5 | 0.8 |  | 0.306 |
| IA | 0.5 | 0.6 |  | 0.356 |
| IA | 0.5 | 0.4 |  | 0.418 |
| IA | 0.5 | 0.2 |  | 0.465 |
| IA | 1.0 | 0.8 |  | 0.240 |
| IA | 1.0 | 0.6 |  | 0.312 |
| IA | 1.0 | 0.4 |  | 0.406 |
| IA | 1.0 | 0.2 |  | 0.520 |
| IA | 5.0 | 0.8 |  | 0.048 |
| IA | 5.0 | 0.6 |  | 0.187 |
| IA | 5.0 | 0.4 |  | 0.428 |
| IA | 5.0 | 0.2 |  | 0.729 |
| IA | 10.0 | 0.8 |  | 0.015 |
| IA | 10.0 | 0.6 |  | 0.132 |
| IA | 10.0 | 0.4 |  | 0.452 |
| IA | 10.0 | 0.2 |  | 0.824 |
| IA | 20.0 | 0.8 |  | 0.002 |
| IA | 20.0 | 0.6 |  | 0.079 |
| IA | 20.0 | 0.4 |  | 0.448 |
| IA | 20.0 | 0.2 |  | 0.896 |
| Pre | 0.0 | 0.8 |  | 0.338 |
| Pre | 0.0 | 0.6 |  | 0.332 |
| Pre | 0.0 | 0.4 |  | 0.336 |
| Pre | 0.0 | 0.2 |  | 0.339 |
| Pre | 0.5 | 0.8 |  | 0.250 |
| Pre | 0.5 | 0.6 |  | 0.296 |
| Pre | 0.5 | 0.4 |  | 0.350 |
| Pre | 0.5 | 0.2 |  | 0.401 |
| Pre | 1.0 | 0.8 |  | 0.186 |
| Pre | 1.0 | 0.6 |  | 0.252 |
| Pre | 1.0 | 0.4 |  | 0.338 |
| Pre | 1.0 | 0.2 |  | 0.444 |
| Pre | 5.0 | 0.8 |  | 0.028 |
| Pre | 5.0 | 0.6 |  | 0.126 |
| Pre | 5.0 | 0.4 |  | 0.329 |
| Pre | 5.0 | 0.2 |  | 0.634 |
| Pre | 10.0 | 0.8 |  | 0.009 |
| Pre | 10.0 | 0.6 |  | 0.080 |
| Pre | 10.0 | 0.4 |  | 0.328 |
| Pre | 10.0 | 0.2 |  | 0.723 |
| Pre | 20.0 | 0.8 |  | 0.001 |
| Pre | 20.0 | 0.6 |  | 0.040 |
| Pre | 20.0 | 0.4 |  | 0.296 |
| Pre | 20.0 | 0.2 |  | 0.789 |

# 11 Table 5

Table5DF\_1 <- SimDataTab1 %>%  
 mutate(  
 CAT1 = ifelse(FA\_FL == "STOP", "STOP AT FINAL", "GO AT FINAL"),  
 CAT2 = ifelse(  
 FA\_FL == "GO" & IA\_FL == "GO",  
 "GO AT BOTH IA AND FINAL",  
 ifelse(FA\_FL == "STOP" & IA\_FL == "STOP","STOP AT BOTH IA AND FINAL","OTHER")  
 ),  
 CAT3 = ifelse(  
 FA\_FL == "GO" & IA\_FL == "STOP",  
 "STOP AT IA GO AT FA",  
 ifelse(FA\_FL == "STOP" & IA\_FL == "GO","GO AT IA STOP AT FA","OTHER")  
 )  
 )

## 11.1 NO IA

Table5DF\_1\_single <- Table5DF\_1 %>%  
 group\_by(ne,p,CAT1) %>%  
 summarise(cnt = n()) %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT1, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table5DF\_1\_single) %>%  
 FitFlextableToPage()

| ne | p | GO AT FINAL | STOP AT FINAL |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.592 | 0.408 |
| 0.0 | 0.6 | 0.592 | 0.408 |
| 0.0 | 0.4 | 0.584 | 0.416 |
| 0.0 | 0.2 | 0.587 | 0.413 |
| 0.5 | 0.8 | 0.671 | 0.329 |
| 0.5 | 0.6 | 0.622 | 0.378 |
| 0.5 | 0.4 | 0.556 | 0.444 |
| 0.5 | 0.2 | 0.509 | 0.490 |
| 1.0 | 0.8 | 0.735 | 0.265 |
| 1.0 | 0.6 | 0.666 | 0.334 |
| 1.0 | 0.4 | 0.561 | 0.439 |
| 1.0 | 0.2 | 0.460 | 0.540 |
| 5.0 | 0.8 | 0.950 | 0.050 |
| 5.0 | 0.6 | 0.799 | 0.201 |
| 5.0 | 0.4 | 0.543 | 0.457 |
| 5.0 | 0.2 | 0.232 | 0.768 |
| 10.0 | 0.8 | 0.988 | 0.012 |
| 10.0 | 0.6 | 0.862 | 0.138 |
| 10.0 | 0.4 | 0.497 | 0.503 |
| 10.0 | 0.2 | 0.129 | 0.871 |
| 20.0 | 0.8 | 0.999 | 0.002 |
| 20.0 | 0.6 | 0.916 | 0.084 |
| 20.0 | 0.4 | 0.500 | 0.500 |
| 20.0 | 0.2 | 0.056 | 0.944 |

## 11.2 Consistent decision

Table5DF\_2\_dual <- Table5DF\_1 %>%  
 group\_by(ne,p,CAT2) %>%  
 summarise(cnt = n()) %>%  
 filter(CAT2!="OTHER") %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT2, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table5DF\_2\_dual) %>%  
 FitFlextableToPage()

| ne | p | GO AT BOTH IA AND FINAL | STOP AT BOTH IA AND FINAL |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.581 | 0.374 |
| 0.0 | 0.6 | 0.579 | 0.374 |
| 0.0 | 0.4 | 0.577 | 0.377 |
| 0.0 | 0.2 | 0.578 | 0.377 |
| 0.5 | 0.8 | 0.659 | 0.290 |
| 0.5 | 0.6 | 0.609 | 0.338 |
| 0.5 | 0.4 | 0.544 | 0.400 |
| 0.5 | 0.2 | 0.496 | 0.446 |
| 1.0 | 0.8 | 0.722 | 0.224 |
| 1.0 | 0.6 | 0.654 | 0.293 |
| 1.0 | 0.4 | 0.547 | 0.388 |
| 1.0 | 0.2 | 0.443 | 0.496 |
| 5.0 | 0.8 | 0.942 | 0.037 |
| 5.0 | 0.6 | 0.782 | 0.163 |
| 5.0 | 0.4 | 0.517 | 0.393 |
| 5.0 | 0.2 | 0.216 | 0.702 |
| 10.0 | 0.8 | 0.983 | 0.008 |
| 10.0 | 0.6 | 0.843 | 0.106 |
| 10.0 | 0.4 | 0.473 | 0.414 |
| 10.0 | 0.2 | 0.118 | 0.808 |
| 20.0 | 0.8 | 0.998 | 0.001 |
| 20.0 | 0.6 | 0.899 | 0.055 |
| 20.0 | 0.4 | 0.461 | 0.391 |
| 20.0 | 0.2 | 0.050 | 0.885 |

## 11.3 Inconsistent decision

Table5DF\_3\_dual <- Table5DF\_1 %>%  
 group\_by(ne,p,CAT3) %>%  
 summarise(cnt = n()) %>%  
 filter(CAT3!="OTHER") %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT3, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table5DF\_3\_dual) %>%  
 FitFlextableToPage()

| ne | p | GO AT IA STOP AT FA | STOP AT IA GO AT FA |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.034 | 0.011 |
| 0.0 | 0.6 | 0.034 | 0.013 |
| 0.0 | 0.4 | 0.040 | 0.007 |
| 0.0 | 0.2 | 0.036 | 0.009 |
| 0.5 | 0.8 | 0.040 | 0.012 |
| 0.5 | 0.6 | 0.041 | 0.013 |
| 0.5 | 0.4 | 0.044 | 0.013 |
| 0.5 | 0.2 | 0.045 | 0.014 |
| 1.0 | 0.8 | 0.042 | 0.013 |
| 1.0 | 0.6 | 0.041 | 0.012 |
| 1.0 | 0.4 | 0.051 | 0.013 |
| 1.0 | 0.2 | 0.044 | 0.016 |
| 5.0 | 0.8 | 0.013 | 0.008 |
| 5.0 | 0.6 | 0.038 | 0.017 |
| 5.0 | 0.4 | 0.064 | 0.026 |
| 5.0 | 0.2 | 0.066 | 0.016 |
| 10.0 | 0.8 | 0.004 | 0.004 |
| 10.0 | 0.6 | 0.033 | 0.019 |
| 10.0 | 0.4 | 0.089 | 0.024 |
| 10.0 | 0.2 | 0.063 | 0.010 |
| 20.0 | 0.8 | 0.001 | 0.000 |
| 20.0 | 0.6 | 0.028 | 0.017 |
| 20.0 | 0.4 | 0.109 | 0.039 |
| 20.0 | 0.2 | 0.059 | 0.006 |

# 12 Table 6

Table6DF\_1 <- SimDataTab2 %>%  
 mutate(  
 CAT1 = ifelse(FA\_FL == "STOP", "STOP AT FINAL", "GO AT FINAL"),  
 CAT2 = ifelse(  
 FA\_FL == "GO" & IA\_FL == "GO",  
 "GO AT BOTH IA AND FINAL",  
 ifelse(FA\_FL == "STOP" & IA\_FL == "STOP","STOP AT BOTH IA AND FINAL","OTHER")  
 ),  
 CAT3 = ifelse(  
 FA\_FL == "GO" & IA\_FL == "STOP",  
 "STOP AT IA GO AT FA",  
 ifelse(FA\_FL == "STOP" & IA\_FL == "GO","GO AT IA STOP AT FA","OTHER")  
 )  
 )

## 12.1 NO IA

Table6DF\_1\_single <- Table6DF\_1 %>%  
 group\_by(ne,p,CAT1) %>%  
 summarise(cnt = n()) %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT1, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table6DF\_1\_single) %>%  
 FitFlextableToPage()

| ne | p | GO AT FINAL | STOP AT FINAL |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.592 | 0.408 |
| 0.0 | 0.6 | 0.592 | 0.408 |
| 0.0 | 0.4 | 0.584 | 0.416 |
| 0.0 | 0.2 | 0.587 | 0.413 |
| 0.5 | 0.8 | 0.671 | 0.329 |
| 0.5 | 0.6 | 0.622 | 0.378 |
| 0.5 | 0.4 | 0.556 | 0.444 |
| 0.5 | 0.2 | 0.509 | 0.490 |
| 1.0 | 0.8 | 0.735 | 0.265 |
| 1.0 | 0.6 | 0.666 | 0.334 |
| 1.0 | 0.4 | 0.561 | 0.439 |
| 1.0 | 0.2 | 0.460 | 0.540 |
| 5.0 | 0.8 | 0.950 | 0.050 |
| 5.0 | 0.6 | 0.799 | 0.201 |
| 5.0 | 0.4 | 0.543 | 0.457 |
| 5.0 | 0.2 | 0.232 | 0.768 |
| 10.0 | 0.8 | 0.988 | 0.012 |
| 10.0 | 0.6 | 0.862 | 0.138 |
| 10.0 | 0.4 | 0.497 | 0.503 |
| 10.0 | 0.2 | 0.129 | 0.871 |
| 20.0 | 0.8 | 0.999 | 0.002 |
| 20.0 | 0.6 | 0.916 | 0.084 |
| 20.0 | 0.4 | 0.500 | 0.500 |
| 20.0 | 0.2 | 0.056 | 0.944 |

## 12.2 Consistent decision

Table6DF\_2\_dual <- Table6DF\_1 %>%  
 group\_by(ne,p,CAT2) %>%  
 summarise(cnt = n()) %>%  
 filter(CAT2!="OTHER") %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT2, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table6DF\_2\_dual) %>%  
 FitFlextableToPage()

| ne | p | GO AT BOTH IA AND FINAL | STOP AT BOTH IA AND FINAL |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.581 | 0.374 |
| 0.0 | 0.6 | 0.579 | 0.374 |
| 0.0 | 0.4 | 0.577 | 0.377 |
| 0.0 | 0.2 | 0.578 | 0.377 |
| 0.5 | 0.8 | 0.659 | 0.290 |
| 0.5 | 0.6 | 0.609 | 0.338 |
| 0.5 | 0.4 | 0.544 | 0.400 |
| 0.5 | 0.2 | 0.496 | 0.446 |
| 1.0 | 0.8 | 0.722 | 0.224 |
| 1.0 | 0.6 | 0.654 | 0.293 |
| 1.0 | 0.4 | 0.547 | 0.388 |
| 1.0 | 0.2 | 0.443 | 0.496 |
| 5.0 | 0.8 | 0.942 | 0.037 |
| 5.0 | 0.6 | 0.782 | 0.163 |
| 5.0 | 0.4 | 0.517 | 0.393 |
| 5.0 | 0.2 | 0.216 | 0.702 |
| 10.0 | 0.8 | 0.983 | 0.008 |
| 10.0 | 0.6 | 0.843 | 0.106 |
| 10.0 | 0.4 | 0.473 | 0.414 |
| 10.0 | 0.2 | 0.118 | 0.808 |
| 20.0 | 0.8 | 0.998 | 0.001 |
| 20.0 | 0.6 | 0.899 | 0.055 |
| 20.0 | 0.4 | 0.461 | 0.391 |
| 20.0 | 0.2 | 0.050 | 0.885 |

## 12.3 Inconsistent decision

Table6DF\_3\_dual <- Table6DF\_1 %>%  
 group\_by(ne,p,CAT3) %>%  
 summarise(cnt = n()) %>%  
 filter(CAT3!="OTHER") %>%  
 mutate(freq = round(cnt/iter,3)) %>%  
 select(-c(cnt)) %>%  
 spread(CAT3, freq) %>%  
 arrange(ne,desc(p))  
  
flextable(Table6DF\_3\_dual) %>%  
 FitFlextableToPage()

| ne | p | GO AT IA STOP AT FA | STOP AT IA GO AT FA |
| --- | --- | --- | --- |
| 0.0 | 0.8 | 0.034 | 0.011 |
| 0.0 | 0.6 | 0.034 | 0.013 |
| 0.0 | 0.4 | 0.040 | 0.007 |
| 0.0 | 0.2 | 0.036 | 0.009 |
| 0.5 | 0.8 | 0.040 | 0.012 |
| 0.5 | 0.6 | 0.041 | 0.013 |
| 0.5 | 0.4 | 0.044 | 0.013 |
| 0.5 | 0.2 | 0.045 | 0.014 |
| 1.0 | 0.8 | 0.042 | 0.013 |
| 1.0 | 0.6 | 0.041 | 0.012 |
| 1.0 | 0.4 | 0.051 | 0.013 |
| 1.0 | 0.2 | 0.044 | 0.016 |
| 5.0 | 0.8 | 0.013 | 0.008 |
| 5.0 | 0.6 | 0.038 | 0.017 |
| 5.0 | 0.4 | 0.064 | 0.026 |
| 5.0 | 0.2 | 0.066 | 0.016 |
| 10.0 | 0.8 | 0.004 | 0.004 |
| 10.0 | 0.6 | 0.033 | 0.019 |
| 10.0 | 0.4 | 0.089 | 0.024 |
| 10.0 | 0.2 | 0.063 | 0.010 |
| 20.0 | 0.8 | 0.001 | 0.000 |
| 20.0 | 0.6 | 0.028 | 0.017 |
| 20.0 | 0.4 | 0.109 | 0.039 |
| 20.0 | 0.2 | 0.059 | 0.006 |

## 12.4 Inconsistency phenomenon True Effect Means near TV/LRV

There is an increase in the chance to observe a decision that is inconsistent at or before the interim and final when **the True Effect mean is near the TV(0.5)/LRV(0.3)**. This chance appears to increase with the sample size equivalent used (See **Table 5 & Table 6** where the True Effect mean is either 0.4 or 0.6.

This would seem to suggest a risk in using this design when there is reason to **select higher sample size equivalents when the True Effect mean is in the neighborhood of the targets set** for the development plan.

# 13 Figure 4

* Simple PP trial simulation function

PPtrials <- function(Nt, N\_i, N\_p, true\_1, Tar, strt, stop, PPmon){  
  
 # Sample from prior  
 Bpar1 <- Beta\_ab(N\_p, true\_1)  
 theta\_1 <- rbeta(1, Bpar1[1], Bpar1[2])  
  
 # Generate Data  
 arm1 <- rbinom(Nt, 1, theta\_1)  
 int\_arm1 <- arm1[1:N\_i]  
  
 #################################################  
  
 PP1 <- vector(mode = "numeric", length = stop - strt)  
 PP1\_N <- vector(mode = "numeric", length = stop - strt)  
  
 h <- vector(mode = "numeric", length = stop - strt)  
 for (g in 1:(stop - strt)){  
 h[g] <- g  
 PP1\_N[g] <- strt + (g - 1)  
 PP1[g] <- pred\_power\_bin(Nt, strt + (g - 1), round(Nt\*Tar,0), max(sum(int\_arm1[1:(strt + (g-1))])))  
 }  
  
 pptab <- data.frame(PP1, PP1\_N)  
 ppck1 <- pptab[pptab$PP1 < PPmon,]  
 ppN1 <- ifelse(length(ppck1$PP1\_N) == 0, 0, min(ppck1$PP1\_N))  
 if(min(pptab$PP1) >= PPmon){  
 a1NPP <- N\_i  
 } else {  
 a1NPP <- ppN1   
 }  
   
 # if meet target  
 tmt <- (sum(arm1)/Nt)>=Tar  
  
 # Package results   
 tr <- list(a1NPP,pptab,tmt)  
 return(tr)   
}

* Begin PP monitoring

PPrange <- seq(4,20,2)

* True effect mean

TrueEffRange <- seq(0.1,0.8,0.05)

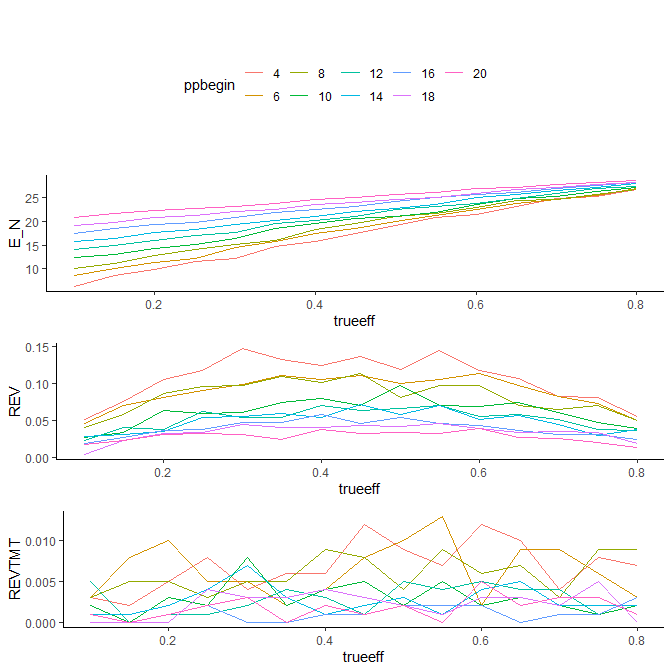
## 13.1 , sample size equivalent=1

* Generate simulation dataset

# set seed  
set.seed(125)  
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
Target <- 0.52  
  
# iteration  
iter <- 1000  
  
# PP stop/start  
PP\_stop <- 30  
  
# Equivalent sample size  
ne <- 1  
  
# Simulation  
TR <- length(TrueEffRange)  
  
# Initiate parameters/data containers to be used in simulations  
SimDataFigA <-  
 data.frame(matrix(NA, length(PPrange) \* length(TrueEffRange) \* iter, 5)) # create matrix container  
  
colnames(SimDataFigA) <- c("ppbegin","trueeff","stopN","reverseflag","meettargt")  
  
stop\_N <- vector(mode = "numeric", length = iter)  
reverse <- vector(mode = "numeric", length = iter)  
revtmt <- vector(mode = "numeric", length = iter)  
  
row <- 1  
  
if (file.exists("SimDataFigA.RData")){  
 load("SimDataFigA.RData")  
} else {  
 for (pstat in PPrange){  
 for (a in 1:length(TrueEffRange)){  
 for (b in 1:iter){  
 tr\_out <- PPtrials(N, Ni, ne, TrueEffRange[a], Target, pstat, PP\_stop, MON\_stop)  
  
 # the number of patients before NO Continue decision is made during iter times of sim  
 stop\_N[b] <- tr\_out[[1]]  
 # pp and corresponding #pt during ppstart to ppstop  
 tr\_hist <- tr\_out[[2]]  
 # meeting target?  
 tmtfl <- tr\_out[[3]]  
   
 # flag the combination of pp/pp\_N after the timepoint where pp monitoring is stopped  
 after\_stop <- tr\_hist[tr\_hist$PP1\_N > stop\_N[b],]  
 # flag the reverse; but why stop\_N[b] < PP\_stop - 2 ????  
 reverse[b] <- ifelse(length(after\_stop[(after\_stop$PP1 >= MON\_stop),][,2]) == 0, 0, 1)  
 # Number of times PP > MON\_stop at end of monitoring | at least one reverse  
 revtmt[b] <- ifelse(reverse[b]==1 & tmtfl==1, 1, 0)  
   
 # package data.frame  
 SimDataFigA[row,1] <- pstat  
 SimDataFigA[row,2] <- TrueEffRange[a]  
 SimDataFigA[row,3] <- stop\_N[b]  
 SimDataFigA[row,4] <- reverse[b]  
 SimDataFigA[row,5] <- revtmt[b]  
   
 row <- row+1  
 }  
 }  
}  
  
SimDataFigA[,1] <- as.factor(SimDataFigA[,1])  
  
save(SimDataFigA,file = "SimDataFigA.RData")  
}

* Analysis and plot

# Avg sample size  
AnalyAvgSS\_A <- SimDataFigA %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(E\_N=mean(stopN)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_A1 <- ggplot(AnalyAvgSS\_A, aes(x=trueeff, y=E\_N, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +  
 theme(legend.position = "top")  
  
# Reversal  
AnalyRev\_A <- SimDataFigA %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REV=mean(reverseflag)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_A2 <- ggplot(AnalyRev\_A, aes(x=trueeff, y=REV, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +  
 theme(legend.position = "none")  
  
# Reversal + trial meets target  
AnalyTmt\_A <- SimDataFigA %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REVTMT=mean(meettargt)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_A3 <- ggplot(AnalyTmt\_A, aes(x=trueeff, y=REVTMT, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +  
 theme(legend.position = "none")  
  
# save the common legend  
legend <- get\_legend(p\_A1)  
  
# remove the legend from the p\_D1  
p\_A1 <- p\_A1 + theme(legend.position="none")  
  
# Create a blank plot  
blankPlot <- ggplot()+geom\_blank(aes(1,1)) + cowplot::theme\_nothing()  
  
# overall  
grid.arrange(legend, p\_A1, p\_A2, p\_A3, nrow = 4)



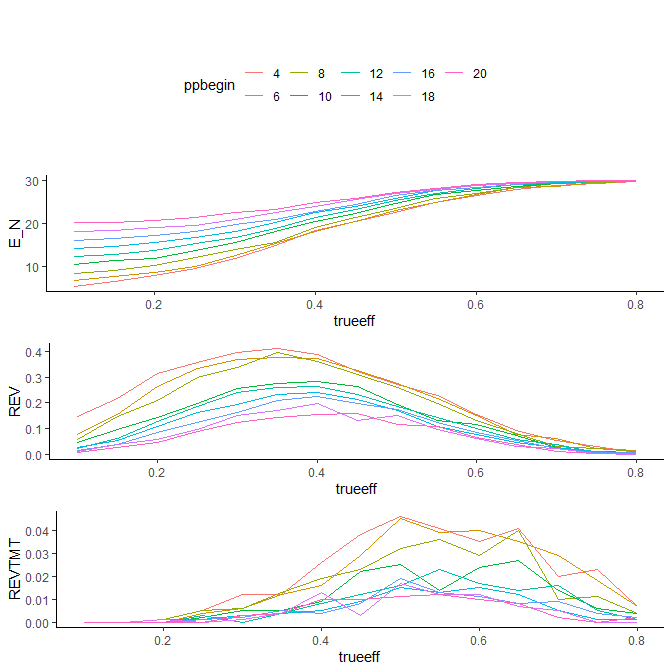
## 13.2 , sample size equivalent=20

* Generate simulation dataset

# set seed  
set.seed(125)  
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
Target <- 0.52  
  
# iteration  
iter <- 1000  
  
# PP stop/start  
PP\_stop <- 30  
  
# Equivalent sample size  
ne <- 20  
  
# Simulation  
TR <- length(TrueEffRange)  
  
# Initiate parameters/data containers to be used in simulations  
SimDataFigB <-  
 data.frame(matrix(NA, length(PPrange) \* length(TrueEffRange) \* iter, 5)) # create matrix container  
  
colnames(SimDataFigB) <- c("ppbegin","trueeff","stopN","reverseflag","meettargt")  
  
stop\_N <- vector(mode = "numeric", length = iter)  
reverse <- vector(mode = "numeric", length = iter)  
revtmt <- vector(mode = "numeric", length = iter)  
  
row <- 1  
  
if (file.exists("SimDataFigB.RData")){  
 load("SimDataFigB.RData")  
} else {  
 for (pstat in PPrange){  
 for (a in 1:length(TrueEffRange)){  
 for (b in 1:iter){  
 tr\_out <- PPtrials(N, Ni, ne, TrueEffRange[a], Target, pstat, PP\_stop, MON\_stop)  
  
 # the number of patients before NO Continue decision is made during iter times of sim  
 stop\_N[b] <- tr\_out[[1]]  
 # pp and corresponding #pt during ppstart to ppstop  
 tr\_hist <- tr\_out[[2]]  
 # meeting target?  
 tmtfl <- tr\_out[[3]]  
   
 # flag the combination of pp/pp\_N after the timepoint where pp monitoring is stopped  
 after\_stop <- tr\_hist[tr\_hist$PP1\_N > stop\_N[b],]  
 # flag the reverse; but why stop\_N[b] < PP\_stop - 2 ????  
 reverse[b] <- ifelse(length(after\_stop[(after\_stop$PP1 >= MON\_stop),][,2]) == 0, 0, 1)  
 # Number of times PP > MON\_stop at end of monitoring | at least one reverse  
 revtmt[b] <- ifelse(reverse[b]==1 & tmtfl==1, 1, 0)  
   
 # package data.frame  
 SimDataFigB[row,1] <- pstat  
 SimDataFigB[row,2] <- TrueEffRange[a]  
 SimDataFigB[row,3] <- stop\_N[b]  
 SimDataFigB[row,4] <- reverse[b]  
 SimDataFigB[row,5] <- revtmt[b]  
   
 row <- row+1  
 }  
 }  
}  
  
SimDataFigB[,1] <- as.factor(SimDataFigB[,1])  
 save(SimDataFigB,file = "SimDataFigB.RData")  
}

* Analysis and plot

# Avg sample size  
AnalyAvgSS\_B <- SimDataFigB %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(E\_N=mean(stopN)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_B1 <- ggplot(AnalyAvgSS\_B, aes(x=trueeff, y=E\_N, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# Reversal  
AnalyRev\_B <- SimDataFigB %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REV=mean(reverseflag)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_B2 <- ggplot(AnalyRev\_B, aes(x=trueeff, y=REV, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# Reversal + trial meets target  
AnalyTmt\_B <- SimDataFigB %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REVTMT=mean(meettargt)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_B3 <- ggplot(AnalyTmt\_B, aes(x=trueeff, y=REVTMT, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# overall  
grid.arrange(legend, p\_B1, p\_B2, p\_B3, nrow = 4)



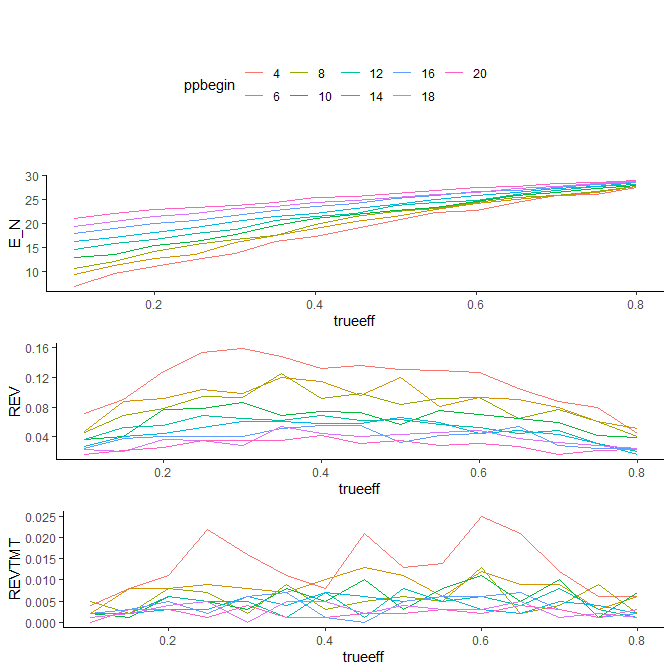
## 13.3 , sample size equivalent=1

* Generate simulation dataset

# set seed  
set.seed(125)  
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
Target <- 0.40  
  
# iteration  
iter <- 1000  
  
# PP stop/start  
PP\_stop <- 30  
  
# Equivalent sample size  
ne <- 1  
  
# Simulation  
TR <- length(TrueEffRange)  
  
# Initiate parameters/data containers to be used in simulations  
SimDataFigC <-  
 data.frame(matrix(NA, length(PPrange) \* length(TrueEffRange) \* iter, 5)) # create matrix container  
  
colnames(SimDataFigC) <- c("ppbegin","trueeff","stopN","reverseflag","meettargt")  
  
stop\_N <- vector(mode = "numeric", length = iter)  
reverse <- vector(mode = "numeric", length = iter)  
revtmt <- vector(mode = "numeric", length = iter)  
  
row <- 1  
  
if (file.exists("SimDataFigC.RData")){  
 load("SimDataFigC.RData")  
} else {  
 for (pstat in PPrange){  
 for (a in 1:length(TrueEffRange)){  
 for (b in 1:iter){  
 tr\_out <- PPtrials(N, Ni, ne, TrueEffRange[a], Target, pstat, PP\_stop, MON\_stop)  
  
 # the number of patients before NO Continue decision is made during iter times of sim  
 stop\_N[b] <- tr\_out[[1]]  
 # pp and corresponding #pt during ppstart to ppstop  
 tr\_hist <- tr\_out[[2]]  
 # meeting target?  
 tmtfl <- tr\_out[[3]]  
   
 # flag the combination of pp/pp\_N after the timepoint where pp monitoring is stopped  
 after\_stop <- tr\_hist[tr\_hist$PP1\_N > stop\_N[b],]  
 # flag the reverse; but why stop\_N[b] < PP\_stop - 2 ????  
 reverse[b] <- ifelse(length(after\_stop[(after\_stop$PP1 >= MON\_stop),][,2]) == 0, 0, 1)  
 # Number of times PP > MON\_stop at end of monitoring | at least one reverse  
 revtmt[b] <- ifelse(reverse[b]==1 & tmtfl==1, 1, 0)  
   
 # package data.frame  
 SimDataFigC[row,1] <- pstat  
 SimDataFigC[row,2] <- TrueEffRange[a]  
 SimDataFigC[row,3] <- stop\_N[b]  
 SimDataFigC[row,4] <- reverse[b]  
 SimDataFigC[row,5] <- revtmt[b]  
   
 row <- row+1  
 }  
 }  
}  
  
SimDataFigC[,1] <- as.factor(SimDataFigC[,1])  
 save(SimDataFigC,file = "SimDataFigC.RData")  
}

* Analysis and plot

# Avg sample size  
AnalyAvgSS\_C <- SimDataFigC %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(E\_N=mean(stopN)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_C1 <- ggplot(AnalyAvgSS\_C, aes(x=trueeff, y=E\_N, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# Reversal  
AnalyRev\_C <- SimDataFigC %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REV=mean(reverseflag)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_C2 <- ggplot(AnalyRev\_C, aes(x=trueeff, y=REV, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# Reversal + trial meets target  
AnalyTmt\_C <- SimDataFigC %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REVTMT=mean(meettargt)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_C3 <- ggplot(AnalyTmt\_C, aes(x=trueeff, y=REVTMT, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +   
 theme(legend.position = "none")  
  
# overall  
grid.arrange(legend,p\_C1, p\_C2, p\_C3, nrow = 4)



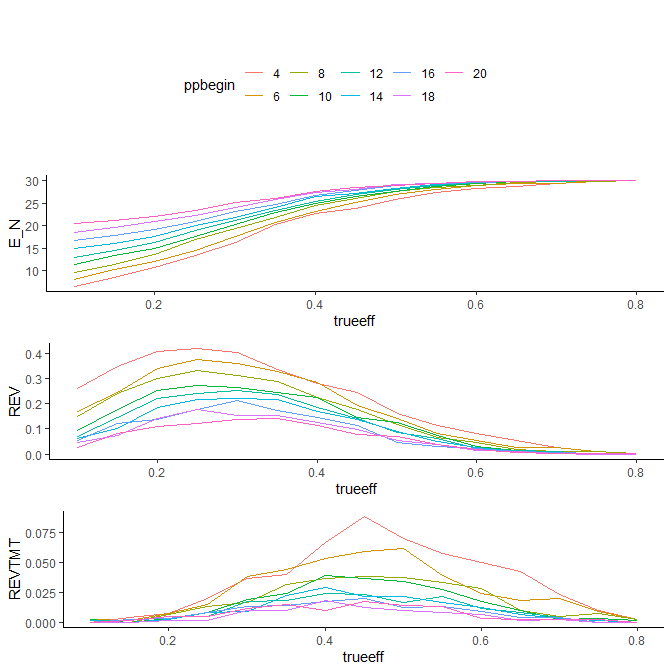
## 13.4 , sample size equivalent=20

* Generate simulation dataset

# set seed  
set.seed(125)  
  
# Proportion of responders to be observed at final that will lead to a non-stop decision  
Target <- 0.40  
  
# iteration  
iter <- 1000  
  
# PP stop/start  
PP\_stop <- 30  
  
# Equivalent sample size  
ne <- 20  
  
# Simulation  
TR <- length(TrueEffRange)  
  
# Initiate parameters/data containers to be used in simulations  
SimDataFigD <-  
 data.frame(matrix(NA, length(PPrange) \* length(TrueEffRange) \* iter, 5)) # create matrix container  
  
colnames(SimDataFigD) <- c("ppbegin","trueeff","stopN","reverseflag","meettargt")  
  
stop\_N <- vector(mode = "numeric", length = iter)  
reverse <- vector(mode = "numeric", length = iter)  
revtmt <- vector(mode = "numeric", length = iter)  
  
row <- 1  
  
if (file.exists("SimDataFigD.RData")){  
 load("SimDataFigD.RData")  
} else {  
 for (pstat in PPrange){  
 for (a in 1:length(TrueEffRange)){  
 for (b in 1:iter){  
 tr\_out <- PPtrials(N, Ni, ne, TrueEffRange[a], Target, pstat, PP\_stop, MON\_stop)  
  
 # the number of patients before NO Continue decision is made during iter times of sim  
 stop\_N[b] <- tr\_out[[1]]  
 # pp and corresponding #pt during ppstart to ppstop  
 tr\_hist <- tr\_out[[2]]  
 # meeting target?  
 tmtfl <- tr\_out[[3]]  
   
 # flag the combination of pp/pp\_N after the timepoint where pp monitoring is stopped  
 after\_stop <- tr\_hist[tr\_hist$PP1\_N > stop\_N[b],]  
 # flag the reverse; but why stop\_N[b] < PP\_stop - 2 ????  
 reverse[b] <- ifelse(length(after\_stop[(after\_stop$PP1 >= MON\_stop),][,2]) == 0, 0, 1)  
 # Number of times PP > MON\_stop at end of monitoring | at least one reverse  
 revtmt[b] <- ifelse(reverse[b]==1 & tmtfl==1, 1, 0)  
   
 # package data.frame  
 SimDataFigD[row,1] <- pstat  
 SimDataFigD[row,2] <- TrueEffRange[a]  
 SimDataFigD[row,3] <- stop\_N[b]  
 SimDataFigD[row,4] <- reverse[b]  
 SimDataFigD[row,5] <- revtmt[b]  
   
 row <- row+1  
 }  
 }  
}  
  
SimDataFigD[,1] <- as.factor(SimDataFigD[,1])  
 save(SimDataFigD,file = "SimDataFigD.RData")  
}

* Analysis and plot

# Avg sample size  
AnalyAvgSS\_D <- SimDataFigD %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(E\_N=mean(stopN)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_D1 <- ggplot(AnalyAvgSS\_D, aes(x=trueeff, y=E\_N, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic()+  
 theme(legend.position = "none")  
  
# Reversal  
AnalyRev\_D <- SimDataFigD %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REV=mean(reverseflag)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_D2 <- ggplot(AnalyRev\_D, aes(x=trueeff, y=REV, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +  
 theme(legend.position="none")  
  
# Reversal + trial meets target  
AnalyTmt\_D <- SimDataFigD %>%  
 group\_by(ppbegin,trueeff) %>%  
 summarise(REVTMT=mean(meettargt)) %>%  
 arrange(ppbegin,trueeff)  
  
p\_D3 <- ggplot(AnalyTmt\_D, aes(x=trueeff, y=REVTMT, group=ppbegin)) +  
 geom\_line(aes(color=ppbegin))+  
 theme\_classic() +  
 theme(legend.position="none")  
  
# overall  
grid.arrange(legend, p\_D1, p\_D2, p\_D3, nrow = 4)



# 14 Guidance and Practical Implications

## 14.1 Setting

There are **two** considerations regarding the choice of .

* The first is that while can relate to the choice of particular decision criteria at the end of the trial, in some cases this can lead to unacceptable operating characteristics.
  + In particular, setting too high can result in an unacceptably high probability of premature stopping.

Setting higher than the target of interest is not recommended and will lead to discarding what would otherwise be good candidates for further development.

In the example above is set according to **the observed proportion of responders required out of the full planned sample that will lead to a non-stop decision at that point**.

When comparing Table 3 () and Table 4 () the losses Pre IA and IA or before are the same when . This suggests that there may be a point in any similar design where for of at least some value, that all of the futility related decision making could be made based on the predictive probability monitoring.

An administrative look could then be incorporated if there is a desire to include a program decision that allows acceleration of planning or other activity within the program but outside the trial.

**However, setting too high can lead to an increase of inconsistent early stopping**.

So, while was chosen as the non-stop criterion at the final analysis, it is not automatically set in this way. **As with the values of any of the parameters, the operating characteristics of the corresponding design should be carefully understood before trial conduct**.

## 14.2 TV/LRV and the lack of “PAUSE” results

With the decision framework that is used in these examples **there are only two decision categories (GO/STOP)**. Results falling into one of these two categories suggest a clear basis for either continuing development or stopping development respectively.

With designs that include fewer patients or where the TV and LRV are set closer together, there can be a third decision category between the GO and STOP categories (**PAUSE**).

Results falling into this category are not by themselves enough to decide. This third category has typically meant that additional information from the trial should be considered in order to support a clear STOP or GO decision.

*Reference*

* *Lalonde RL, Kowalski KG, Hutmacher MM, et al. Model–based Drug Development. Clinical Pharmacology & Therapeutics. 2007;82(1):21–32.*
* *Frewer P, Mitchell P, Watkins C, et al. Decision Making in Early Clinical Drug Development. Pharmaceutical Statistics. 2016;15(3):255–263.*

## 14.3 The effect of the timing of beginning predictive probability monitoring

Selecting the beginning of the predictive probability monitoring phase of the design does have an impact on some of its operating characteristics (Figure 4).

The effect on **average sample size** is obvious and more pronounced in trials which terminate early since a certain number of patients are observed before the beginning of the predictive probability monitoring phase. Beyond this, average sample sizes rise to the maximum planned sample size with increasing True Effect mean.

The chance of **observing a reversal** increases with fewer patients observed before the predictive probability monitoring phase begins.

Further in this design there another decision point at the end of the trial that would have to be met before a compound would proceed further controlling the probability of an ineffective compound proceeding in development with borderline interim results. Though it is not something that was explored explicitly here, **simulations are the best way to examine the effect of the start of the predictive probability phase of the trial on the overall risk of stopping the trial at various phases** (looking specifically at the difference between the chance to stop without interim analysis and the corresponding chance to stop before IA, at IA or before or at any time in the trial).

## 14.4 Practical implications

In order for this method to be successfully implemented, strict adherence to the design as planned is necessary.

**Changes to the timing of the start of any of the phases within the design** are possible but would require simulations to fully understand the changes in the risks from the base case. This is necessary in order to avoid missing reversals in the response proportion. *A reversal is suggestive of a STOP decision and should be taken seriously since the true mean proportion of patients responding is likely much lower than may otherwise be suggested when ignoring the reversal*. This can lead to higher costs and more patients exposed to an otherwise ineffective compound than necessary.

It is also important to note that good ongoing data management is vital to the success of this design in particular when considering the possibility of stopping during the phase of the design concerned with the assessment of predictive probability. In particular, **timely data entry and cleaning** will quickly inform the possibility of a stop decision early in predictive probability monitoring. Changes in response due to data cleaning can increase the risk of incorrect decisions early and as an extension have unpredictable effects on average sample size.

As in all study designs, **several assumptions are made for the purposes of planning the trial**. In cases where more patients are recruited than originally planned, then as part of the procedures described in Frewer (2016) the criteria should be re-estimated with the current assumptions.

* *Frewer P, Mitchell P, Watkins C, et al. Decision Making in Early Clinical Drug Development. Pharmaceutical Statistics. 2016;15(3):255–263.*

## 14.5 Multiplicity

It is the case that in these examples there are **20 tests (predictive probability after patients 12–29 an interim at patient 30 and the final at patient 50) being conducted**.

**Were this trial to be designed using frequentist methods**, there would be quite a number of adjustments made along with stopping boundaries specified for each of the 20 assessments.

**Using Bayesian methodology and simulations, there is a much simpler way to address the problem**. The operating characteristics are fully characterized over a number of True Effect distributions in terms of both sample size equivalents and range of True Effect mean. Since predictive probability is calculated using an algorithm, there is no need to calculate the individual stopping boundaries and include those in a protocol ahead of time. The algorithm is simply run following each patient until either a decision to stop the trial is made or until the planned number of patients in the predictive probability monitoring phase is reached.

Even though simulations are used along with an algorithm to estimate the chance of inappropriate early stopping, **it is important to note that with more interim analyses (or longer periods of predictive probability monitoring), the risk of inappropriate stopping will increase**.

Multiple simulations may be needed to select design parameters that lead to an acceptable level of risk.