Bayesian Go/No-Go: Normal Distribution

Erina Paul

Contents

1 GoNoGo

The purpose of this `R` implementation is to perform Bayesian Go/No-Go decision-making based on specific model assumptions. It specifically covers scenarios involving single normal mean, two independent normal mean differences, and two independent normal mean ratio analysis, considering both informative and non-informative priors. For a more in-depth understanding of the technical aspects, please refer to the `Details` section within the documentation.

1.1 Dependencies

`GoNoGo` requires the following `R` package: `devtools` (for installation only). Please install it before installing `GoNoGo`, which can be done as follows (execute from within a fresh `R` session):

```
install.packages("devtools")
library(devtools)
```

1.2 Installation

Once the dependencies are installed, `GoNoGo` can be loaded using the following command:

```r
devtools::install_github("Merck/GoNoGo")
library(GoNoGo)
```

1.3 Usage

This implementation provides functionality for performing Bayesian Go/No-Go decision-making in various scenarios. The four main functions for normal distribution included are:

- Normal1SampleGNG(n.t, mu.t, sd.t, hypothesis, threshold, PP.cutoffGo, PP.cutoffNoGo, prior, mu0t, nu0t, kappa0t, var0t, nsim, reps): This function allows for the analysis of the mean of a single population with a normal endpoint. It requires parameters such as the sample size (n.t), the true mean (mu.t), the standard deviation (sd.t), the hypothesis being tested, the threshold for decision-making, the cutoff values for Go and No-Go decisions, prior information, and the number of simulations and repetitions.
- Normal2SampleGNG(n.t, n.c, mu.t, mu.c, sd.t, sd.c, measure, hypothesis, threshold, PP.cutoffGo, PP.cutoffNoGo, prior, mu0t, nu0t, kappa0t, var0t, mu0c, nu0c, kappa0c, var0c, nsim, reps): This function is used to analyze the difference/ratio in means between two independent populations with normal endpoints. It requires parameters such as the sample sizes for the two populations (n.t and n.c), the true means (mu.t and mu.c), the standard deviations (sd.t and sd.c), measure (difference/ratio), the hypothesis, the decision threshold, the cutoff values for Go and No-Go decisions, prior information for both populations, and the number of simulations and repetitions.
- Normal1SamplePP(n.t, mu.t, sd.t, hypothesis, threshold, PP.cutoffGo, PP.cutoffNoGo, prior, mu0t, nu0t, kappa0t, var0t, reps): This function allows the posterior probability calculation of the mean of a single population with a normal endpoint. It requires parameters such as the sample size (n.t), the true mean (mu.t), the standard deviation (sd.t), the hypothesis being tested, the threshold for decision-making, the cutoff values for Go and No-Go decisions, prior information, and the number of repetitions.
- Normal2SamplePP(n.t, n.c, mu.t, mu.c, sd.t, sd.c, measure, hypothesis, threshold, PP.cutoffGo, PP.cutoffNoGo, prior, mu0t, nu0t, kappa0t, var0t, mu0c, nu0c, kappa0c, var0c, nsim, reps): This function is used to calculate the posterior probability of the difference/ratio in means between two independent populations with normal endpoints. It requires parameters such as the sample sizes for the two populations (n.t and n.c), the true means (mu.t and mu.c), the standard deviations (sd.t and sd.c), measure (difference/ratio), the hypothesis, the decision threshold, the cutoff values for Go and No-Go decisions, prior information for both populations, and the number of repetitions.

These functions enable users to perform Bayesian Go/No-Go decision-making in scenarios involving single population mean analysis, difference/ratio in means between two populations.

1.4 Input

For mean of single population with normal endpoint, the inputs are:

- n.t: Sample size in treatment arm.
- mu.t: True mean of the treatment arm.
- sd.t: True sd of the treatment arm.
- hypothesis: “greater than” (default), “less than”, “between”.
- threshold: Target region cut-off; k is scaler for “greater than” and “less than” hypotheses; k is a vector of k1 and k2 for “between” hypothesis.
- PP.cutoffGo: Cut-off for probability of Go; Default is 0.60.
- PP.cutoffNoGo: Cut-off for probability of No-Go; Default is 0.60.
- prior: Prior choices: “non-informative” (Default), “informative”.
- mu0t: Prior mean for treatment hyper-parameter if prior = “informative”.
- nu0t: Prior sample size for treatment hyper-parameter if prior = “informative”.
- kappa0t: Prior degrees of freedom for treatment hyper-parameter if prior = “informative”.

- var0t: Prior variance for treatment hyper-parameter if prior = "informative".
- nsim: Number of simulated data sets; Default is 10000.
- reps: Number of repetition for samples from the posterior using simulation; Default is 10000.

In addition to the previous input variables, the calculation for difference or ratio of means of two independent population with normal endpoint includes

- n.c: Sample size in control arm.
- mu.c: True mean of the control arm.
- sd.c: True sd of the control arm.
- mu0c: Prior mean for control hyper-parameter if prior = "informative".
- nu0c: Sample size for control hyper-parameter if prior = "informative".
- kappa0c: Degrees of freedom for control hyper-parameter if prior = "informative".
- var0c: Prior variance for control hyper-parameter if prior = "informative".

1.5 Output

A list containing the following components is returned in operating characteristic table:

- pos
  - Input variables
  - probGo: Probability of Go
  - probNoGo: Probability of No-Go
- pp: Posterior probabilities

A list containing the following components is returned in posterior probability output:

- Input variables
- pp: Posterior probabilities

1.6 Details

To calculate the posterior probabilities with probabilities of Go and No-Go, the code follows the steps as below:

- Generate $n.t$ data from Normal$(mu.t, sd.t^2)$
- Calculate sample mean $(\bar{y})$ and sample variance (s_y^2)
- Then draw samples from the posterior distribution using simulation
  - Generate variances from posterior
    - Non-informative prior (See Appendix in Gelman et al. for scaled inverse chi square generation)
      - $s^2 \sim \chi^2(df = (n.t - 1), ncp = 0)$
      - $\sigma^2 = \frac{(n.t-1)*s_y^2}{s^2}$
      - $prior.mu = \bar{y}$
      - $prior.sd = \sqrt{\frac{\sigma^2}{n.t}}$
      - $prior.df = n.t$
    - Informative prior
      - $\mu_{0t}, \nu_{0t}, \kappa_{0t}, var0t$: Prior mean, sample size, degrees of freedom, variance for hyper-parameter
      - $var1 = (\frac{1}{(\nu_{0t}+n.t)} * (\nu_{0t} * var0t + (n.t-1) * s_y^2 + (\frac{\kappa_{0t}*n.t}{\kappa_{0t}+n.t} * (\bar{y} - \mu_{0t})^2)$
      - $prior.mu = \frac{\kappa_{0t}*\mu_{0t}+n.t*\bar{y}}{\kappa_{0t}+n.t}$
      - $prior.sd = \sqrt{\frac{var1}{\kappa_{0t}+n.t}}$
      - $prior.df = \nu_{0t} + n.t$
  - Generate mean from posterior using normal distribution with mean = prior.mu and sd = prior.sd
- Calculate the posterior probabilities
- Now, based on `nsim` iterations, calculate the probabilities of Go and No-Go using the pre-defined posterior probability cut-offs.

For difference and ratio of means of two independent population with normal endpoint, we have used the same formulation for control arm and calculated posterior probability using mean difference or mean ratio or treatment versus control arm followed by calculating the probabilities of Go and No-Go using the pre-defined posterior probability cut-offs based on `nsim` iterations. Note that, for the posterior probability calculation, we do not use `nsim`.

2 Illustration of Operationg Characteristics of `GoNoGo`

In this document, we describe the following scenarios.

2.1 Mean of single normal distribution using non-informative prior with hypothesis = "between"

This example illustrates the steps to solve a single mean problem in which the endpoint data is normally distributed and a non-informative prior is used. First, we load the necessary GoNoGo library. We then define the parameters, such as the sample size (n.t), true mean (mu.t), and standard deviation (sd.t) of the population. Next, we specify the hypothesis as "greater" and set the desired threshold for decision-making. The cutoff probabilities for making a Go decision (PP.cutoffGo) and No-Go decision (PP.cutoffNoGo) are also established. To incorporate a non-informative prior, we set the prior parameter to "noninformative". Finally, we call the `Normal1SampleGNG` function with the provided parameters to perform the analysis.

```
##################
# Load libraries #
##################

library(GoNoGo)
library(ggplot2)
library(gridExtra)
library(grid)
```

```
#################
# Assign inputs #
#################

n.t = rep(8, 9) # Sample size in treatment arm
mu.t = rep(c(40, 45, 50), each = 3) # True mean of the treatment arm
sd.t = rep(15, 9) # True sd of the treatment arm
hypothesis = "between" # "greater than", "less than", "between"
threshold =  matrix(rep(c(35, 45), each = 9), ncol = 2) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "non-informative" # Prior choice
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

It is important to note that if the hypothesis is set to `between`, the threshold should be provided in a matrix format (e.g., `matrix(rep(c(35, 45), 9), ncol = 2)`). In the `Normal1SampleGNG` function, use `threshold = threshold[i,]`. On the other hand, if the hypothesis is either `greater than` or `less than`, the threshold should be in a vector format (e.g., `rep(45, 9)`). In the `Normal1SampleGNG` function, use `threshold = threshold[i]`. This distinction allows for appropriately handling the threshold values based on the specific hypothesis being tested.

```
####################################
# Normal-noninformative: 1 sample #
####################################
```

```r
set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal1SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  hypothesis = hypothesis, threshold = threshold[i,],
                                  PP.cutoffGo = PP.cutoffGo[i],
                                  PP.cutoffNoGo = PP.cutoffNoGo[i],
                                  nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.t = unlist(data$mu.t)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.t = as.factor(data$mu.t)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t   threshold PP.cutoffGo PP.cutoffNoGo hypothesis
#> 1   8   40   15 ( 35 , 45 )         0.6           0.6    between
#> 2   8   40   15 ( 35 , 45 )         0.7           0.7    between
#> 3   8   40   15 ( 35 , 45 )         0.8           0.8    between
#> 4   8   45   15 ( 35 , 45 )         0.6           0.6    between
#> 5   8   45   15 ( 35 , 45 )         0.7           0.7    between
#> 6   8   45   15 ( 35 , 45 )         0.8           0.8    between
#> 7   8   50   15 ( 35 , 45 )         0.6           0.6    between
#> 8   8   50   15 ( 35 , 45 )         0.7           0.7    between
#> 9   8   50   15 ( 35 , 45 )         0.8           0.8    between
#>             prior nsim reps      time probGo probNoGo
#> 1 non-informative 1000 1000 0.6952741   28.0     72.0
#> 2 non-informative 1000 1000 0.6718066   10.9     89.1
#> 3 non-informative 1000 1000 0.6739888    3.1     96.9
#> 4 non-informative 1000 1000 0.6728902   19.4     80.6
#> 5 non-informative 1000 1000 0.7030933    8.2     91.8
#> 6 non-informative 1000 1000 0.6755729    2.0     98.0
#> 7 non-informative 1000 1000 0.6646812    5.7     94.3
#> 8 non-informative 1000 1000 0.6817379    1.9     98.1
#> 9 non-informative 1000 1000 0.7276332    0.7     99.3
```

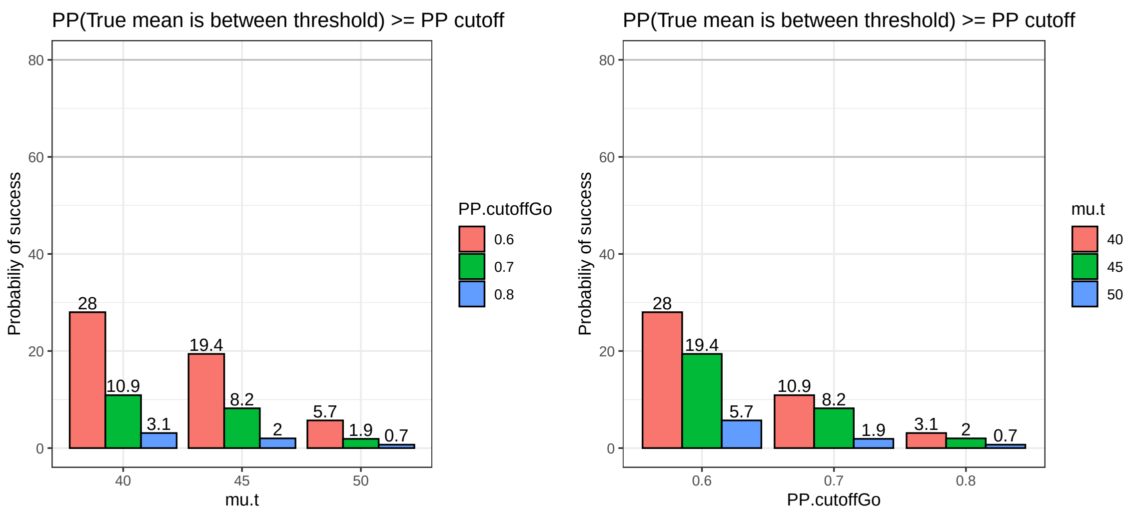
Now, let's obtain the output for various scenarios using two visualizations:

- Visualizing True Means with Different Posterior Probability Cut-offs for Go Decision: We can create a bar plot to show the relationship between different true means and the corresponding posterior probability cut-offs for making a Go decision.
- Visualizing Posterior Probability Cut-offs for Go Decision with Different True Means: We can also create a bar plot to display the relationship between different posterior probability cut-offs for making a Go decision and the corresponding true means.

These visualizations will provide insights into the impact of different parameters on the decision-making process.

```r
p1 = GNGplot(data = data, x.var = data$mu.t,
                     fill.var = data$PP.cutoffGo,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.t",
```

```
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                     fill.var = data$mu.t,
                     hypothesis = data$hypothesis,
                     x.lab = "PP.cutoffGo",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "mu.t" )

grid.arrange(p1, p2, nrow = 1)
```



2.2 Mean of single normal distribution using informative prior with hypothesis = "between"

In this example, we set the parameters such as the sample size (n.t), true mean (mu.t), standard deviation (sd.t), hypothesis, decision threshold, and cutoff probabilities for Go and No-Go decisions.To consider an informative prior, we set the prior mean (mu0t), prior degrees of freedom (nu0t), prior scale (kappa0t), and prior variance (var0t) accordingly.

```
#################
# Assign inputs #
#################

n.t = rep(8, 9) # Sample size in treatment arm
mu.t = rep(c(40, 45, 50), each = 3) # True mean of the treatment arm
sd.t = rep(15, 9) # True sd of the treatment arm
hypothesis = "between" # "greater than", "less than", "between"
threshold = matrix(rep(c(35, 45), each = 9), ncol = 2) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "informative" # Prior choice
mu0t = 35 # Prior mean for treatment hyper-parameter
nu0t = 7 # Prior sample size for treatment hyper-parameter
kappa0t = 4 # Prior DF for treatment hyper-parameter
var0t = 1  # Prior variance for treatment hyper-parameter
```

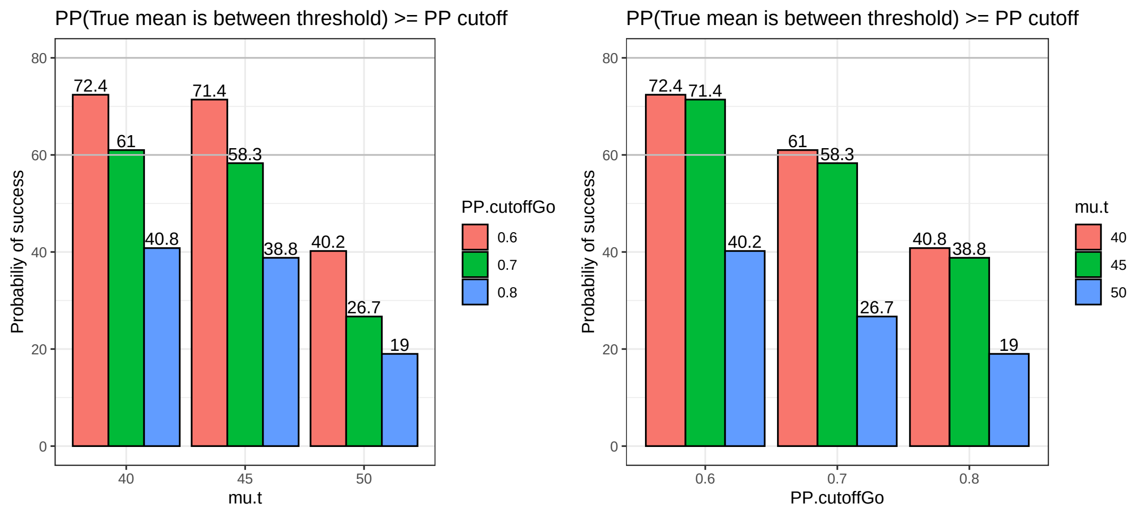
```r
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

Finally, we call the `Normal1SampleGNG` function with all the specified parameters, including the informative prior, to perform the analysis.

```r
################################
# Normal-informative: 1 sample #
################################

set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal1SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  hypothesis = hypothesis, threshold = threshold[i,],
                                  PP.cutoffGo = PP.cutoffGo[i],
                                  PP.cutoffNoGo = PP.cutoffNoGo[i],
                                  prior = prior, mu0t = mu0t, nu0t = nu0t,
                                  kappa0t = kappa0t, var0t = var0t,
                                  nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.t = unlist(data$mu.t)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.t = as.factor(data$mu.t)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t   threshold PP.cutoffGo PP.cutoffNoGo hypothesis       prior
#> 1   8   40   15 ( 35 , 45 )         0.6           0.6    between informative
#> 2   8   40   15 ( 35 , 45 )         0.7           0.7    between informative
#> 3   8   40   15 ( 35 , 45 )         0.8           0.8    between informative
#> 4   8   45   15 ( 35 , 45 )         0.6           0.6    between informative
#> 5   8   45   15 ( 35 , 45 )         0.7           0.7    between informative
#> 6   8   45   15 ( 35 , 45 )         0.8           0.8    between informative
#> 7   8   50   15 ( 35 , 45 )         0.6           0.6    between informative
#> 8   8   50   15 ( 35 , 45 )         0.7           0.7    between informative
#> 9   8   50   15 ( 35 , 45 )         0.8           0.8    between informative
#>   mu0t nu0t kappa0t var0t nsim reps      time probGo probNoGo
#> 1   35    7       4     1 1000 1000 0.3557458   72.4     27.6
#> 2   35    7       4     1 1000 1000 0.3600399   61.0     39.0
#> 3   35    7       4     1 1000 1000 0.3521979   40.8     59.2
#> 4   35    7       4     1 1000 1000 0.3382215   71.4     28.6
#> 5   35    7       4     1 1000 1000 0.3459163   58.3     41.7
#> 6   35    7       4     1 1000 1000 0.3643799   38.8     61.2
#> 7   35    7       4     1 1000 1000  0.345886   40.2     59.8
#> 8   35    7       4     1 1000 1000 0.3575602   26.7     73.3
#> 9   35    7       4     1 1000 1000 0.3561292   19.0     81.0
```

Now, we will obtain the output for different scenarios using two bar plot visualizations: one for true means with varying posterior probability cut-offs for Go decisions, and another for posterior probability cut-offs with differing true means.

```r
p1 = GNGplot(data = data, x.var = data$mu.t,
                      fill.var = data$PP.cutoffGo,
                      hypothesis = data$hypothesis,
                      x.lab = "mu.t",
                      title.text =  paste("PP(True mean is", hypothesis,
                                          "threshold) >= PP cutoff"),
                      legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                      fill.var = data$mu.t,
                      hypothesis = data$hypothesis,
                      x.lab = "PP.cutoffGo",
                      title.text =  paste("PP(True mean is", hypothesis,
                                          "threshold) >= PP cutoff"),
                      legend.title.text = "mu.t" )

grid.arrange(p1, p2, nrow = 1)
```



2.3 Difference of means of two independent normal distributions using non-informative prior with hypothesis = "less than"

In this example, we set the parameters such as the sample sizes for the two populations (n.t and n.c), the true means for the populations (mu.t and mu.c), the standard deviations for the populations (sd.t and sd.c), the hypothesis, the decision threshold, and the cutoff probabilities for Go and No-Go decisions. To utilize a non-informative prior, we set prior to `noninformative`.

```r
#################
# Assign inputs #
#################

n.t = rep(9, 9) # Sample size in treatment arm
mu.t = rep(70, each = 9) # True mean of the treatment arm
sd.t = rep(10, 9) # True sd of the treatment arm
n.c = rep(3, 9) # Sample size in control arm
mu.c = rep(c(40, 45, 50), each = 3) # True mean of the control arm
sd.c = rep(15, 9) # True sd of the control arm
measure = "difference" # "difference" or "ratio" of normal means
hypothesis = "less than" # "greater than", "less than", "between"
```

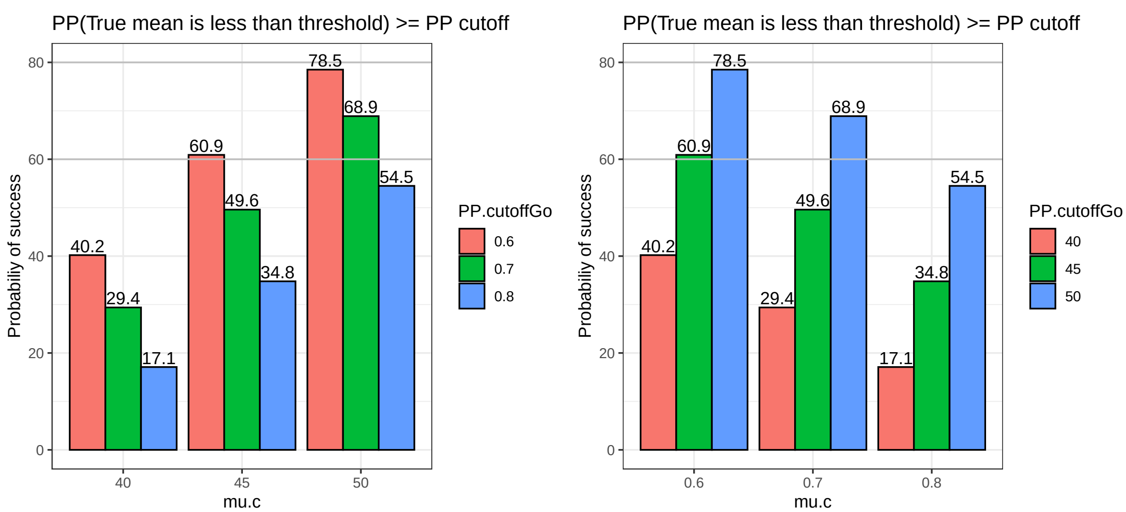
```r
threshold = rep(30, 9) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "non-informative" # Prior choice
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

Finally, we call the `Normal2SampleGNG` function with the specified parameters to perform the analysis and obtain the result.

```r
##############################################
# Normal-noninformative: 2 sample difference #
##############################################

set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal2SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  n.c = n.c[i], mu.c = mu.c[i], sd.c = sd.c[i],
                                  measure = "difference",
                                  hypothesis = hypothesis, threshold = threshold[i],
                                  prior = prior, PP.cutoffGo = PP.cutoffGo[i],
                                  PP.cutoffNoGo = PP.cutoffNoGo[i],
                                  nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.c = unlist(data$mu.c)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.c = as.factor(data$mu.c)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t n.c mu.c sd.c    measure threshold PP.cutoffGo PP.cutoffNoGo
#> 1   9   70   10   3   40   15 difference        30         0.6           0.6
#> 2   9   70   10   3   40   15 difference        30         0.7           0.7
#> 3   9   70   10   3   40   15 difference        30         0.8           0.8
#> 4   9   70   10   3   45   15 difference        30         0.6           0.6
#> 5   9   70   10   3   45   15 difference        30         0.7           0.7
#> 6   9   70   10   3   45   15 difference        30         0.8           0.8
#> 7   9   70   10   3   50   15 difference        30         0.6           0.6
#> 8   9   70   10   3   50   15 difference        30         0.7           0.7
#> 9   9   70   10   3   50   15 difference        30         0.8           0.8
#>   hypothesis           prior nsim reps     time probGo probNoGo
#> 1  less than non-informative 1000 1000 1.218525   40.2     59.8
#> 2  less than non-informative 1000 1000 1.200987   29.4     70.6
#> 3  less than non-informative 1000 1000 1.207733   17.1     82.9
#> 4  less than non-informative 1000 1000 1.181813   60.9     39.1
#> 5  less than non-informative 1000 1000 1.212354   49.6     50.4
#> 6  less than non-informative 1000 1000 1.187761   34.8     65.2
#> 7  less than non-informative 1000 1000 1.209599   78.5     21.5
#> 8  less than non-informative 1000 1000 1.170644   68.9     31.1
#> 9  less than non-informative 1000 1000 1.173244   54.5     45.5
```

In order to analyze different scenarios, we will generate two bar plot visualizations: the first will display true means with varying posterior probability cut-offs for making Go decisions, and the second will show posterior probability cut-offs with differing true means.

```
p1 = GNGplot(data = data, x.var = data$mu.c,
                     fill.var = data$PP.cutoffGo,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                     fill.var =  data$mu.c,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

grid.arrange(p1, p2, nrow = 1)
```



2.4 Difference of means of two independent normal distributions using informative prior with hypothesis = "less than"

In this example, we set the parameters for the analysis, including the sample sizes for the two populations (n.t and n.c), the true means for each population (mu.t and mu.c), the standard deviations for each population (sd.t and sd.c), the hypothesis being tested, the decision threshold, and the cutoff probabilities for making a Go or No-Go decision. To utilize a informative prior, we set the prior parameter to `informative`. These parameter settings allow us to perform the analysis considering the difference of means between two independent populations with a normal endpoint using a informative prior.

```
#################
# Assign inputs #
#################

n.t = rep(9, 9) # Sample size in treatment arm
mu.t = rep(70, each = 9) # True mean of the treatment arm
sd.t = rep(10, 9) # True sd of the treatment arm
```

```r
n.c = rep(3, 9) # Sample size in control arm
mu.c = rep(c(40, 45, 50), each = 3) # True mean of the control arm
sd.c = rep(15, 9) # True sd of the control arm
measure = "difference" # "difference" or "ratio" of normal means
hypothesis = "less than" # "greater than", "less than", "between"
threshold = rep(30, 9) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "informative" # Prior choice
mu0t = 70 # Prior mean for treatment hyper-parameter
nu0t = 7 # Prior sample size for treatment hyper-parameter
kappa0t = 4 # Prior DF for treatment hyper-parameter
var0t = 1  # Prior variance for treatment hyper-parameter
mu0c = 40 # Prior mean for control hyper-parameter
nu0c = 2 # Prior sample size for control hyper-parameter
kappa0c = 2 # Prior DF for control hyper-parameter
var0c = 1  # Prior variance for control hyper-parameter
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

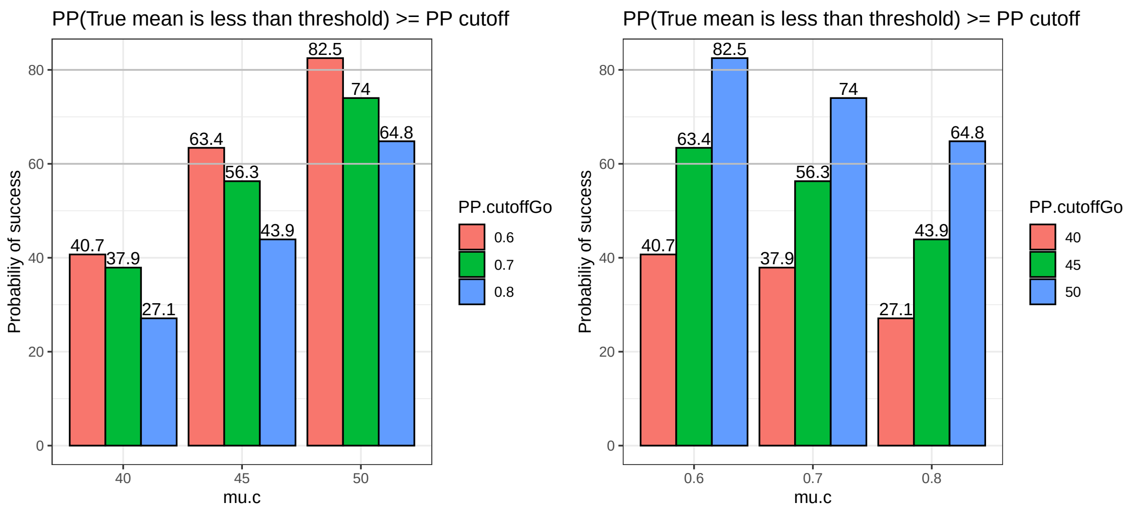
Now, calculate the probabilities of Go and No-Go using informative prior.

```r
###########################################
# Normal-informative: 2 sample difference #
###########################################

set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal2SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  n.c = n.c[i], mu.c = mu.c[i], sd.c = sd.c[i],
                                  measure = "difference",
                                  hypothesis = hypothesis, threshold = threshold[i],
                                  PP.cutoffGo = PP.cutoffGo[i],
                                  PP.cutoffNoGo = PP.cutoffNoGo[i],
                                  prior = prior, mu0t = mu0t, nu0t = nu0t,
                                  kappa0t = kappa0t, var0t = var0t,
                                  mu0c = mu0c, nu0c = nu0c,
                                  kappa0c = kappa0c, var0c = var0c,
                                  nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.c = unlist(data$mu.c)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.c = as.factor(data$mu.c)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t n.c mu.c sd.c    measure threshold PP.cutoffGo PP.cutoffNoGo
#> 1   9   70   10   3   40   15 difference        30         0.6           0.6
#> 2   9   70   10   3   40   15 difference        30         0.7           0.7
#> 3   9   70   10   3   40   15 difference        30         0.8           0.8
#> 4   9   70   10   3   45   15 difference        30         0.6           0.6
```

```
#> 5   9   70   10   3   45   15 difference        30         0.7           0.7
#> 6   9   70   10   3   45   15 difference        30         0.8           0.8
#> 7   9   70   10   3   50   15 difference        30         0.6           0.6
#> 8   9   70   10   3   50   15 difference        30         0.7           0.7
#> 9   9   70   10   3   50   15 difference        30         0.8           0.8
#>   hypothesis       prior mu0t nu0t kappa0t var0t mu0c nu0c kappa0c var0c nsim
#> 1  less than informative   70    7       4     1   40    2       2     1 1000
#> 2  less than informative   70    7       4     1   40    2       2     1 1000
#> 3  less than informative   70    7       4     1   40    2       2     1 1000
#> 4  less than informative   70    7       4     1   40    2       2     1 1000
#> 5  less than informative   70    7       4     1   40    2       2     1 1000
#> 6  less than informative   70    7       4     1   40    2       2     1 1000
#> 7  less than informative   70    7       4     1   40    2       2     1 1000
#> 8  less than informative   70    7       4     1   40    2       2     1 1000
#> 9  less than informative   70    7       4     1   40    2       2     1 1000
#>   reps      time probGo probNoGo
#> 1 1000 0.6205294   40.7     59.3
#> 2 1000 0.5814629   37.9     62.1
#> 3 1000 0.5764706   27.1     72.9
#> 4 1000 0.5814173   63.4     36.6
#> 5 1000 0.5742276   56.3     43.7
#> 6 1000 0.5651298   43.9     56.1
#> 7 1000 0.5764289   82.5     17.5
#> 8 1000 0.5756996   74.0     26.0
#> 9 1000 0.5743225   64.8     35.2
```

Here, both visualizations provide insights into the relationship between true means and posterior probability cut-offs for Go decisions.

```r
p1 = GNGplot(data = data, x.var = data$mu.c,
                     fill.var = data$PP.cutoffGo,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                     fill.var =  data$mu.c,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

grid.arrange(p1, p2, nrow = 1)
```



2.5 Ratio of means of two independent normal distribution using non-informative prior with hypothesis = "greater than"

In this example, we set the parameters such as the sample sizes for the two populations (n.t and n.c), the true means for the populations (mu.t and mu.c), the standard deviations for the populations (sd.t and sd.c), the hypothesis, the decision threshold, and the cutoff probabilities for Go and No-Go decisions. To utilize a non-informative prior, we set prior to `noninformative`.

```
#################
# Assign inputs #
#################

n.t = rep(9, 9) # Sample size in treatment arm
mu.t = rep(70, each = 9) # True mean of the treatment arm
sd.t = rep(10, 9) # True sd of the treatment arm
n.c = rep(3, 9) # Sample size in control arm
mu.c = rep(c(40, 45, 50), each = 3) # True mean of the control arm
sd.c = rep(15, 9) # True sd of the control arm
measure = "ratio" # "difference" or "ratio" of normal means
hypothesis = "greater than" # "greater than", "less than", "between"
threshold = rep(1, 9) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "non-informative" # Prior choice
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

Now, calculate the probabilities of Go and No-Go.

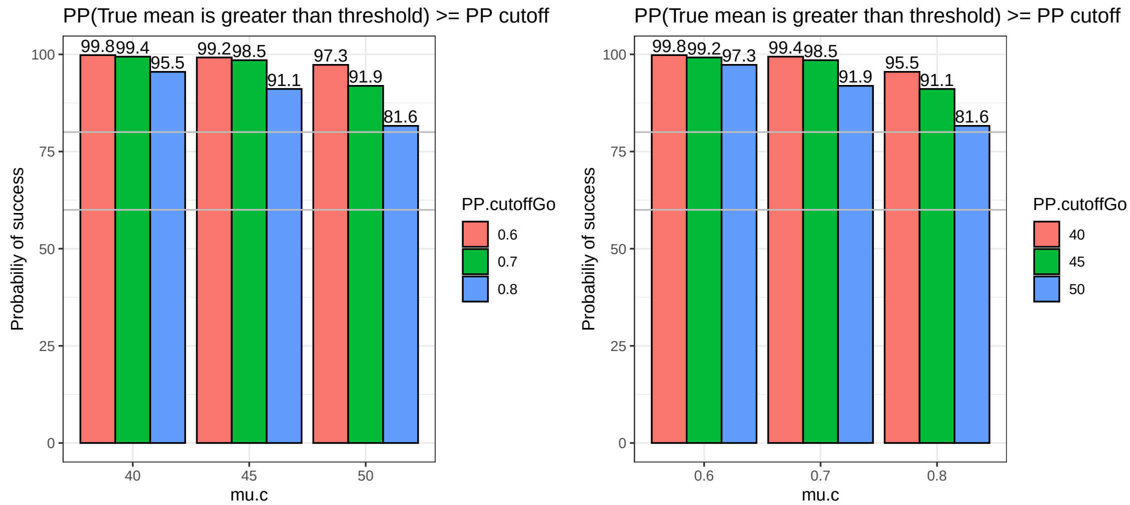
```
#########################################
# Normal-noninformative: 2 sample ratio #
#########################################

set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal2SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  n.c = n.c[i], mu.c = mu.c[i], sd.c = sd.c[i],
                                  measure = "ratio",
```

```r
                                    hypothesis = hypothesis, threshold = threshold[i],
                                    prior = prior, PP.cutoffGo = PP.cutoffGo[i],
                                    PP.cutoffNoGo = PP.cutoffNoGo[i],
                                    nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.c = unlist(data$mu.c)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.c = as.factor(data$mu.c)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t n.c mu.c sd.c measure threshold PP.cutoffGo PP.cutoffNoGo
#> 1   9   70   10   3   40   15   ratio         1         0.6           0.6
#> 2   9   70   10   3   40   15   ratio         1         0.7           0.7
#> 3   9   70   10   3   40   15   ratio         1         0.8           0.8
#> 4   9   70   10   3   45   15   ratio         1         0.6           0.6
#> 5   9   70   10   3   45   15   ratio         1         0.7           0.7
#> 6   9   70   10   3   45   15   ratio         1         0.8           0.8
#> 7   9   70   10   3   50   15   ratio         1         0.6           0.6
#> 8   9   70   10   3   50   15   ratio         1         0.7           0.7
#> 9   9   70   10   3   50   15   ratio         1         0.8           0.8
#>     hypothesis           prior nsim reps     time probGo probNoGo
#> 1 greater than non-informative 1000 1000 1.211634   99.8      0.2
#> 2 greater than non-informative 1000 1000 1.175131   99.4      0.6
#> 3 greater than non-informative 1000 1000 1.171894   95.5      4.5
#> 4 greater than non-informative 1000 1000  1.16974   99.2      0.8
#> 5 greater than non-informative 1000 1000  1.17096   98.5      1.5
#> 6 greater than non-informative 1000 1000 1.186371   91.1      8.9
#> 7 greater than non-informative 1000 1000 1.363475   97.3      2.7
#> 8 greater than non-informative 1000 1000 1.163795   91.9      8.1
#> 9 greater than non-informative 1000 1000 1.195335   81.6     18.4
```

These visualizations help us comprehend the impact of varying true means on the decision-making process, as well as the effect of different posterior probability cut-offs on the same decisions. By examining these relationships, we can gain a better understanding of the interplay between true means and the corresponding posterior probability cut-offs for Go decisions.

```r
p1 = GNGplot(data = data, x.var = data$mu.c,
                      fill.var = data$PP.cutoffGo,
                      hypothesis = data$hypothesis,
                      x.lab = "mu.c",
                      title.text =  paste("PP(True mean is", hypothesis,
                                          "threshold) >= PP cutoff"),
                      legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                      fill.var =  data$mu.c,
                      hypothesis = data$hypothesis,
                      x.lab = "mu.c",
                      title.text =  paste("PP(True mean is", hypothesis,
                                          "threshold) >= PP cutoff"),
                      legend.title.text = "PP.cutoffGo")
```

```r
grid.arrange(p1, p2, nrow = 1)
```



2.6 Ratio of means of two independent normal distributions using informative prior with hypothesis = "greater than"

This example shows how to solve the ratio of means of two independent normal distribution using informative prior:

```r
#################
# Assign inputs #
#################

n.t = rep(9, 9) # Sample size in treatment arm
mu.t = rep(70, each = 9) # True mean of the treatment arm
sd.t = rep(10, 9) # True sd of the treatment arm
n.c = rep(3, 9) # Sample size in control arm
mu.c = rep(c(40, 45, 50), each = 3) # True mean of the control arm
sd.c = rep(15, 9) # True sd of the control arm
measure = "ratio" # "difference" or "ratio" of normal means
hypothesis = "greater than" # "greater than", "less than", "between"
threshold = rep(1, 9) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "informative" # Prior choice
mu0t = 70 # Prior mean for treatment hyper-parameter
nu0t = 5 # Prior mean for treatment hyper-parameter
kappa0t = 8 # Prior DF for treatment hyper-parameter
var0t = 1  # Prior variance for treatment hyper-parameter
mu0c = 45 # Prior mean for control hyper-parameter
nu0c = 2 # Prior mean for control hyper-parameter
kappa0c = 2 # Prior DF for control hyper-parameter
var0c = 1  # Prior variance for control hyper-parameter
reps = 1000 # Number of repetitions for samples from the posterior using simulation
nsim = 1000 # Number of simulations
```

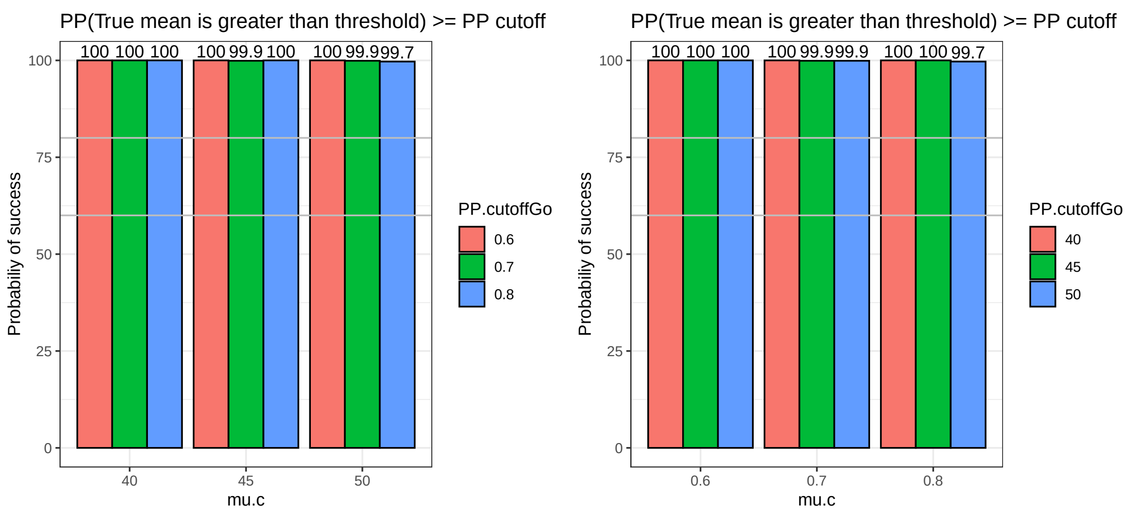
Let's apply normal with mean ratio and calculate the probabilities of Go and No-Go for informative prior.

```r
######################################
# Normal-informative: 2 sample ratio #
######################################

set.seed(1234)
pos.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pos.val[[i]] = Normal2SampleGNG(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                  n.c = n.c[i], mu.c = mu.c[i], sd.c = sd.c[i],
                                  measure = "ratio",
                                  hypothesis = hypothesis, threshold = threshold[i],
                                  PP.cutoffGo = PP.cutoffGo[i],
                                  PP.cutoffNoGo = PP.cutoffNoGo[i],
                                  prior = prior, mu0t = mu0t, nu0t = nu0t,
                                  kappa0t = kappa0t, var0t = var0t,
                                  mu0c = mu0c, nu0c = nu0c,
                                  kappa0c = kappa0c, var0c = var0c,
                                  nsim = nsim, reps = reps)$pos
}
data = dataoutput(prob.val = pos.val)
data$mu.c = unlist(data$mu.c)
data$PP.cutoffGo = unlist(data$ PP.cutoffGo)

data$mu.c = as.factor(data$mu.c)
data$PP.cutoffGo = as.factor(data$ PP.cutoffGo)

data
#>   n.t mu.t sd.t n.c mu.c sd.c measure threshold PP.cutoffGo PP.cutoffNoGo
#> 1   9   70   10   3   40   15   ratio         1         0.6           0.6
#> 2   9   70   10   3   40   15   ratio         1         0.7           0.7
#> 3   9   70   10   3   40   15   ratio         1         0.8           0.8
#> 4   9   70   10   3   45   15   ratio         1         0.6           0.6
#> 5   9   70   10   3   45   15   ratio         1         0.7           0.7
#> 6   9   70   10   3   45   15   ratio         1         0.8           0.8
#> 7   9   70   10   3   50   15   ratio         1         0.6           0.6
#> 8   9   70   10   3   50   15   ratio         1         0.7           0.7
#> 9   9   70   10   3   50   15   ratio         1         0.8           0.8
#>     hypothesis       prior mu0t nu0t kappa0t var0t mu0c nu0c kappa0c var0c nsim
#> 1 greater than informative   70    5       8     1   45    2       2     1 1000
#> 2 greater than informative   70    5       8     1   45    2       2     1 1000
#> 3 greater than informative   70    5       8     1   45    2       2     1 1000
#> 4 greater than informative   70    5       8     1   45    2       2     1 1000
#> 5 greater than informative   70    5       8     1   45    2       2     1 1000
#> 6 greater than informative   70    5       8     1   45    2       2     1 1000
#> 7 greater than informative   70    5       8     1   45    2       2     1 1000
#> 8 greater than informative   70    5       8     1   45    2       2     1 1000
#> 9 greater than informative   70    5       8     1   45    2       2     1 1000
#>   reps      time probGo probNoGo
#> 1 1000 0.6155603  100.0      0.0
#> 2 1000 0.5999141  100.0      0.0
#> 3 1000 0.5864863  100.0      0.0
#> 4 1000 0.5852897  100.0      0.0
#> 5 1000  0.583169   99.9      0.1
```

```
#> 6 1000 0.5447373  100.0       0.0
#> 7 1000 0.4865179  100.0       0.0
#> 8 1000 0.4953742   99.9       0.1
#> 9 1000 0.4844463   99.7       0.3
```

The visualizations of two scenarios are as follows:

```
p1 = GNGplot(data = data, x.var = data$mu.c,
                     fill.var = data$PP.cutoffGo,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

p2 = GNGplot(data = data, x.var = data$PP.cutoffGo,
                     fill.var =  data$mu.c,
                     hypothesis = data$hypothesis,
                     x.lab = "mu.c",
                     title.text =  paste("PP(True mean is", hypothesis,
                                         "threshold) >= PP cutoff"),
                     legend.title.text = "PP.cutoffGo")

grid.arrange(p1, p2, nrow = 1)
```



3 Illustration of Posterior Probability of `GoNoGo`

In this example, we consider one sample mean using non-informative prior with `greater than` hypothesis to set the parameters.

```
#################
# Assign inputs #
#################

n.t = rep(10, 9) # Sample size in treatment arm
mu.t = rep(70, each = 9) # True mean of the treatment arm
sd.t = rep(10, 9) # True sd of the treatment arm
hypothesis = "greater than" # "greater than", "less than", "between"
```

```r
threshold = rep(c(60, 65, 70), each = 3) # k: Target region cut-off
PP.cutoffGo = rep(c(0.6, 0.7, 0.8), 3) # c1: Cut-off for probability of Go
PP.cutoffNoGo = rep(c(0.6, 0.7, 0.8), 3) # c2: Cut-off for probability of No-Go
prior = "non-informative" # Prior choice
reps = 10000 # Number of repetitions for samples from the posterior using simulation
```

Finally, we call the R function with the specified parameters to perform the analysis and obtain the posterior probabilities.

```r
##############################################
# Normal-noninformative: 2 sample difference #
##############################################

set.seed(1234)
pp.val = vector("list", length(n.t))
for(i in 1:length(n.t)){
  pp.val[[i]] = Normal1SamplePP(n.t = n.t[i], mu.t = mu.t[i], sd.t = sd.t[i],
                                hypothesis = hypothesis,
                                threshold = threshold[i], prior = prior,
                                PP.cutoffGo = PP.cutoffGo[i],
                                PP.cutoffNoGo = PP.cutoffNoGo[i], reps = reps)
}
data = dataoutput(prob.val = pp.val, output.format = "PP")
data
#>   n.t mu.t sd.t threshold PP.cutoffGo PP.cutoffNoGo   hypothesis
#> 1  10   70   10        60         0.6           0.6 greater than
#> 2  10   70   10        60         0.7           0.7 greater than
#> 3  10   70   10        60         0.8           0.8 greater than
#> 4  10   70   10        65         0.6           0.6 greater than
#> 5  10   70   10        65         0.7           0.7 greater than
#> 6  10   70   10        65         0.8           0.8 greater than
#> 7  10   70   10        70         0.6           0.6 greater than
#> 8  10   70   10        70         0.7           0.7 greater than
#> 9  10   70   10        70         0.8           0.8 greater than
#>             prior  reps        time     pp
#> 1 non-informative 10000  0.00447607 0.9944
#> 2 non-informative 10000 0.005071878 0.9936
#> 3 non-informative 10000 0.004446268 0.9950
#> 4 non-informative 10000 0.004471302 0.9222
#> 5 non-informative 10000 0.004408836 0.9251
#> 6 non-informative 10000 0.004419804 0.9232
#> 7 non-informative 10000 0.005378962 0.5076
#> 8 non-informative 10000 0.004435539 0.4952
#> 9 non-informative 10000 0.004752398 0.5106
```