Power of mRCT using Japan's Method 1 or 2

Based on Japan's Method 1 or 2, given the global and target region sample sizes, calculate and simulate the marginal probabilities, conditional probabilities, and joint probabilities of global success and efficacy consistency between target region and globally, in clinical trials using superiority, non-inferiority, and equivalence designs.

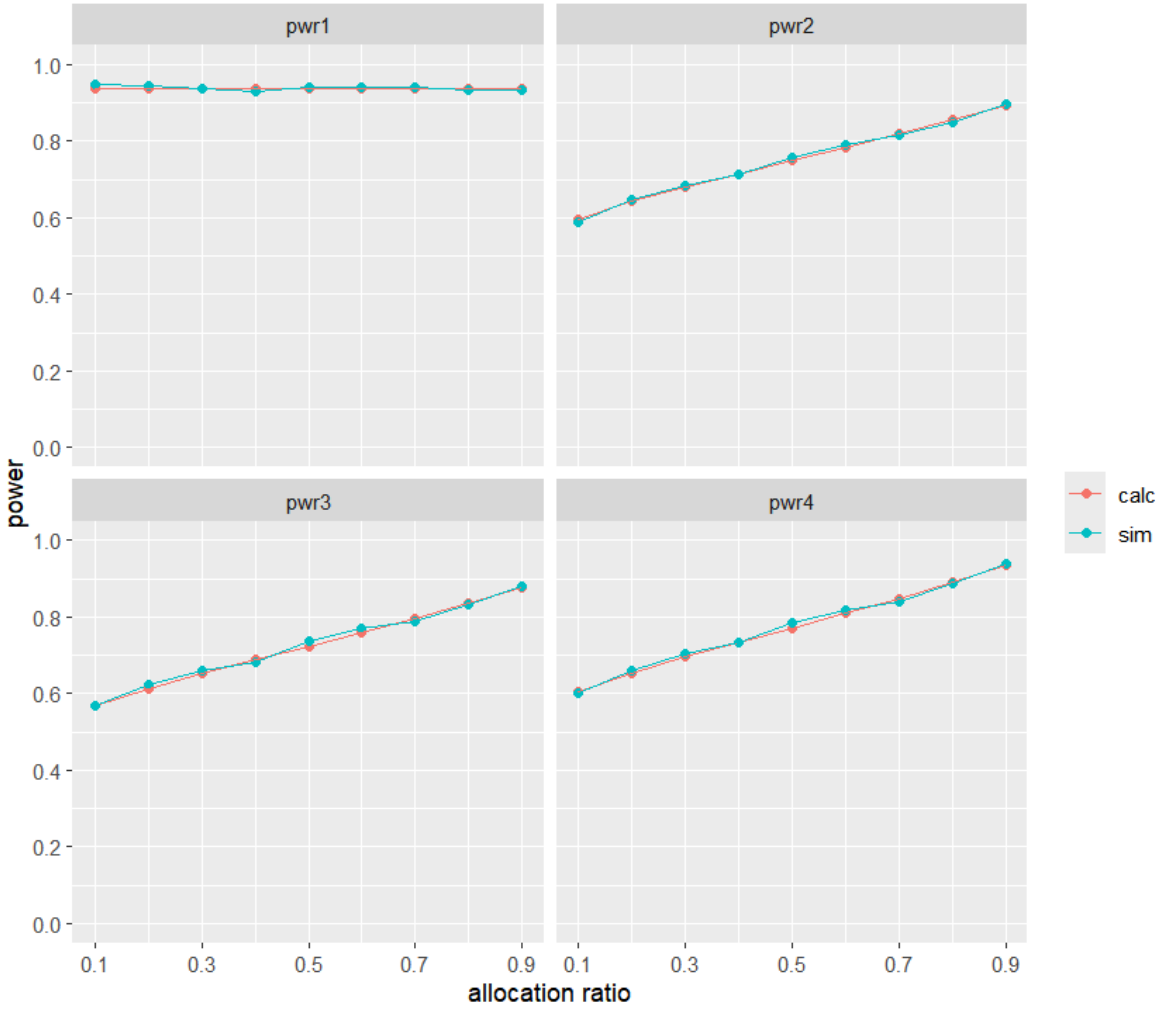
```
library(tidyverse)
library(SSmRCT)
rm(list = ls())

pplot <- function(a, b) {
  a <- pivot_longer(data = a, cols = starts_with("pwr"))
  b <- pivot_longer(data = b, cols = starts_with("pwr"))
  ggplot() +
    geom_point(dat = a, aes(x = f, y = value, color = "calc")) +
    geom_point(dat = b, aes(x = f, y = value, color = "sim")) +
    geom_line(dat = a, aes(x = f, y = value, color = "calc")) +
    geom_line(dat = b, aes(x = f, y = value, color = "sim")) +
    facet_wrap(vars(name)) +
    scale_x_continuous(breaks = seq(0.1, 0.9, 0.2), limits = c(0.1, 0.9)) +
    scale_y_continuous(breaks = seq(0, 1, 0.2), limits = c(0, 1)) +
    labs(x = "allocation ratio", y = "power", color = "")
}
```

getPwr_Con_Super_JM1

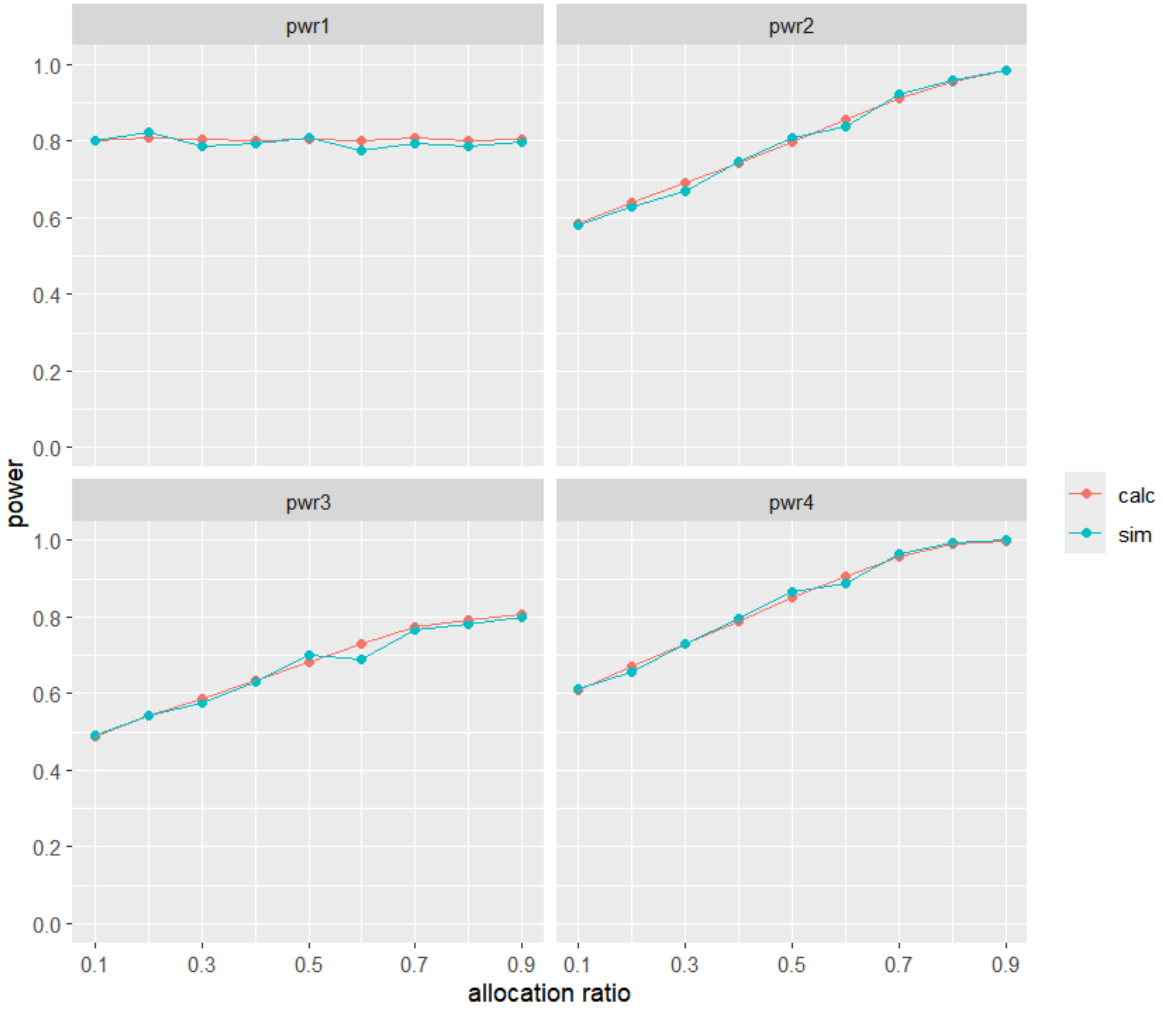
高优

```
a <- getPwr_Con_Super_JM1(delta_j = 0.5, delta_a = 0.7, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 100, r = 1, sim = FALSE)
b <- getPwr_Con_Super_JM1(delta_j = 0.5, delta_a = 0.7, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 100, r = 1, sim = TRUE)
pplot(a, b)
```



低优

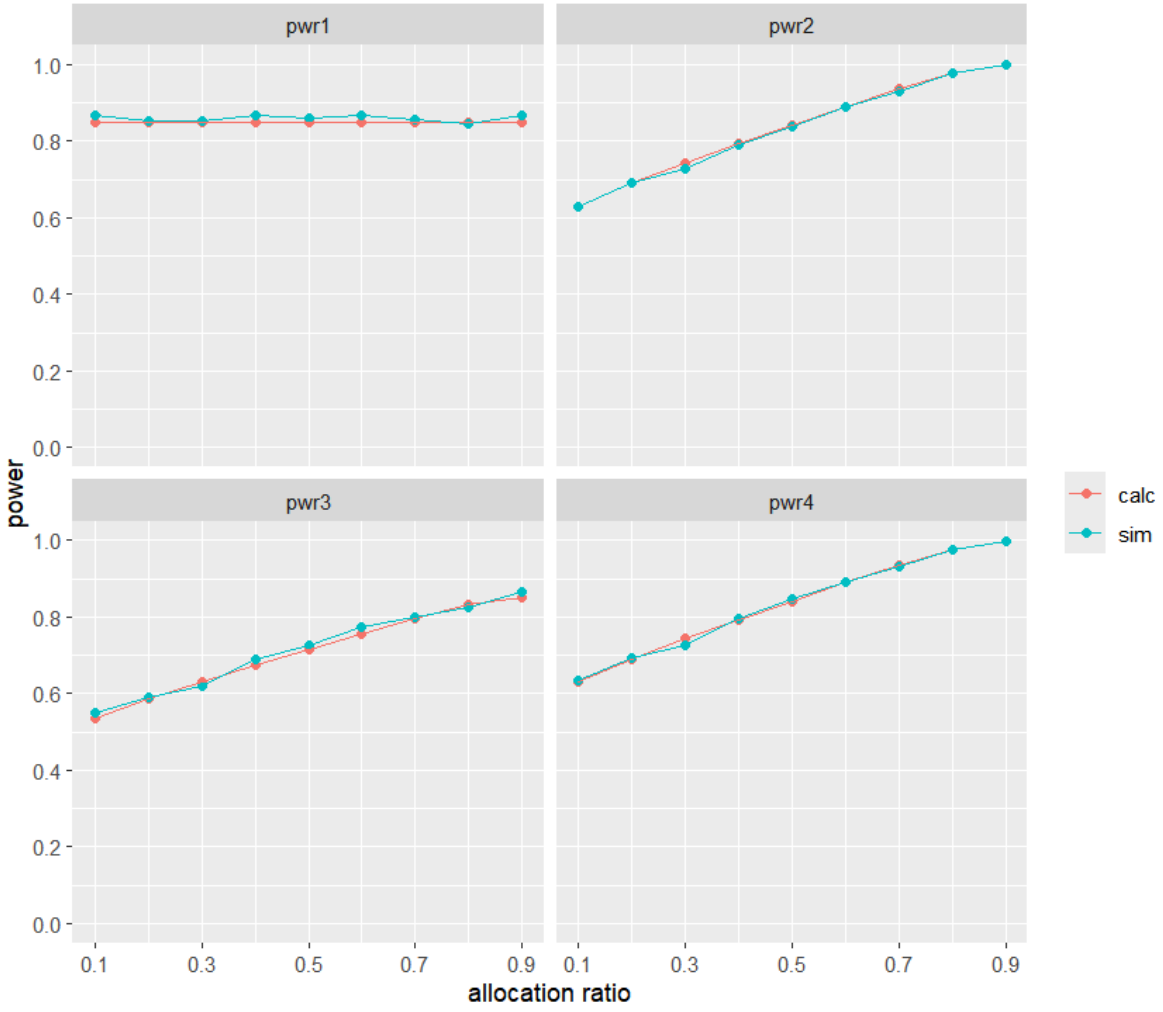
```
c <- getPwr_Con_Super_JM1(delta_j = -0.5, delta_nj = -0.7, sigma = 1, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim =
FALSE)
d <- getPwr_Con_Super_JM1(delta_j = -0.5, delta_nj = -0.7, sigma = 1, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim =
TRUE)
pplot(c, d)
```



getPwr_Con_Noninf_JM1

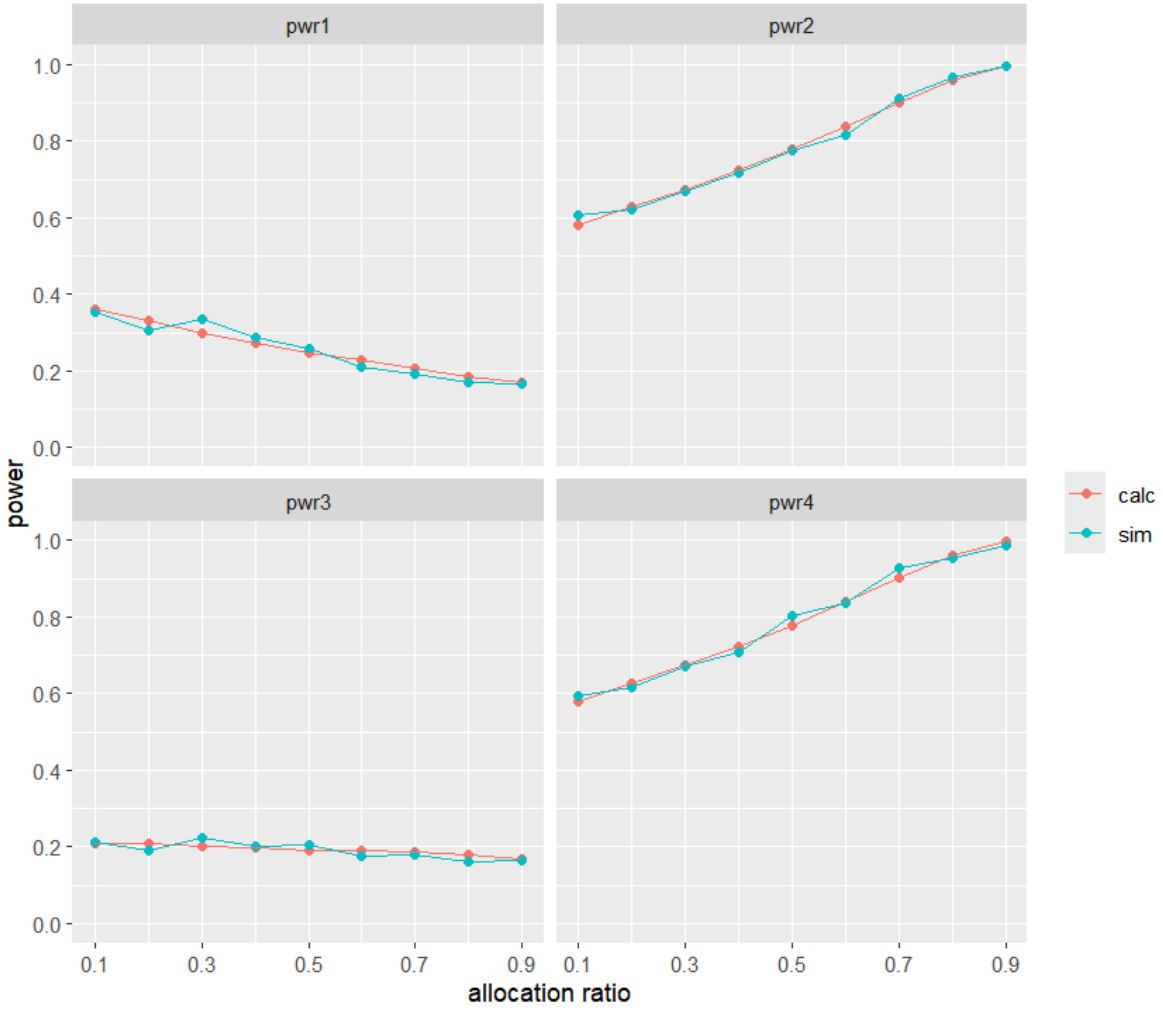
高优

```
a <- getPwr_Con_Noninf_JM1(delta_j = -0.2, delta_a = -0.1, sigma = 1, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = NA, N = 400, r =
1, direct = 1, sim = FALSE)
b <- getPwr_Con_Noninf_JM1(delta_j = -0.2, delta_a = -0.1, sigma = 1, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = NA, N = 400, r =
1, direct = 1, sim = TRUE)
pplot(a, b)
```



低优

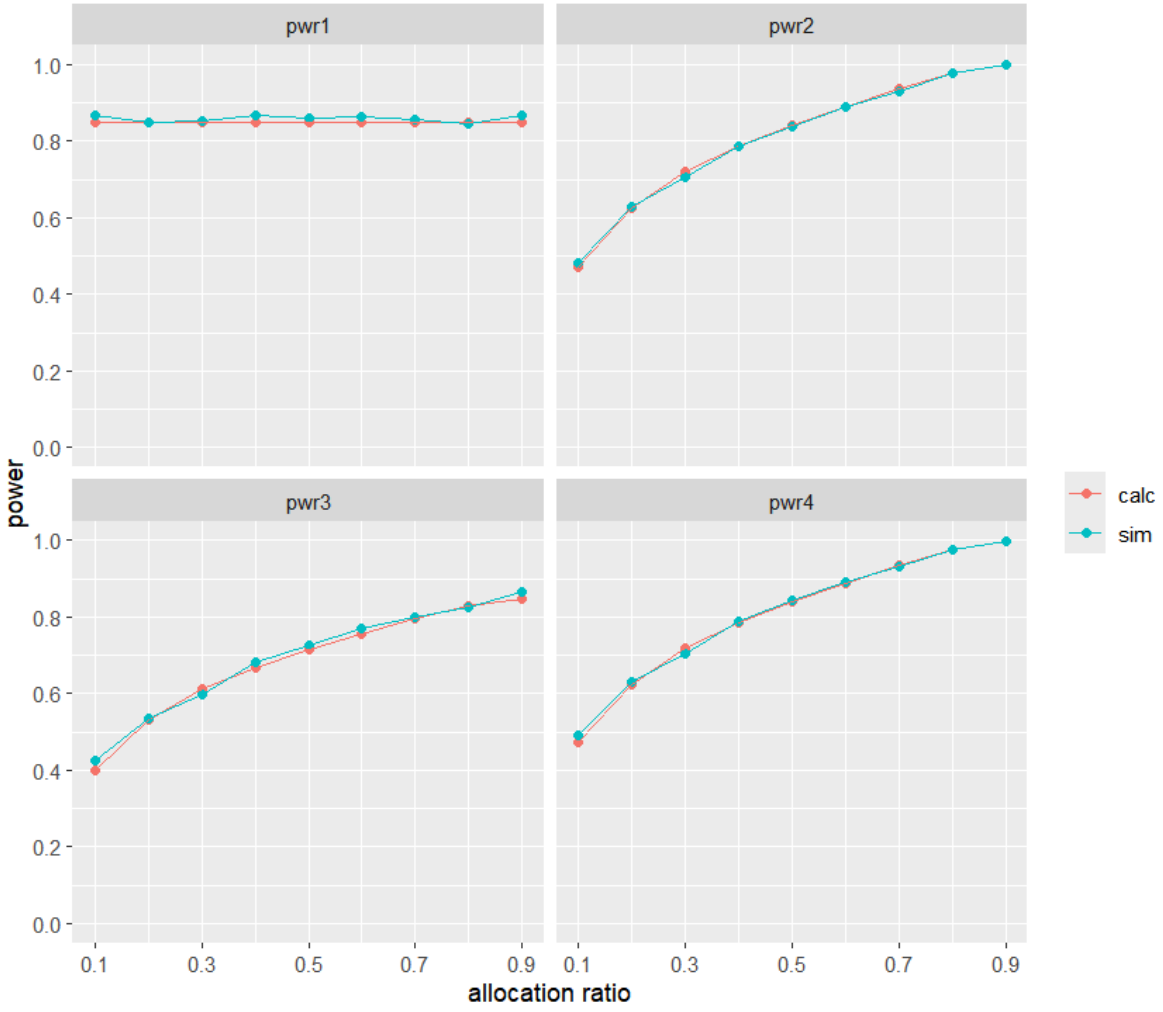
```
c <- getPwr_Con_Noninf_JM1(delta_j = 0.2, delta_nj = 0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = FALSE)
d <- getPwr_Con_Noninf_JM1(delta_j = 0.2, delta_nj = 0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = TRUE)
pplot(c, d)
```



getPwr_Con_Equi_JM1

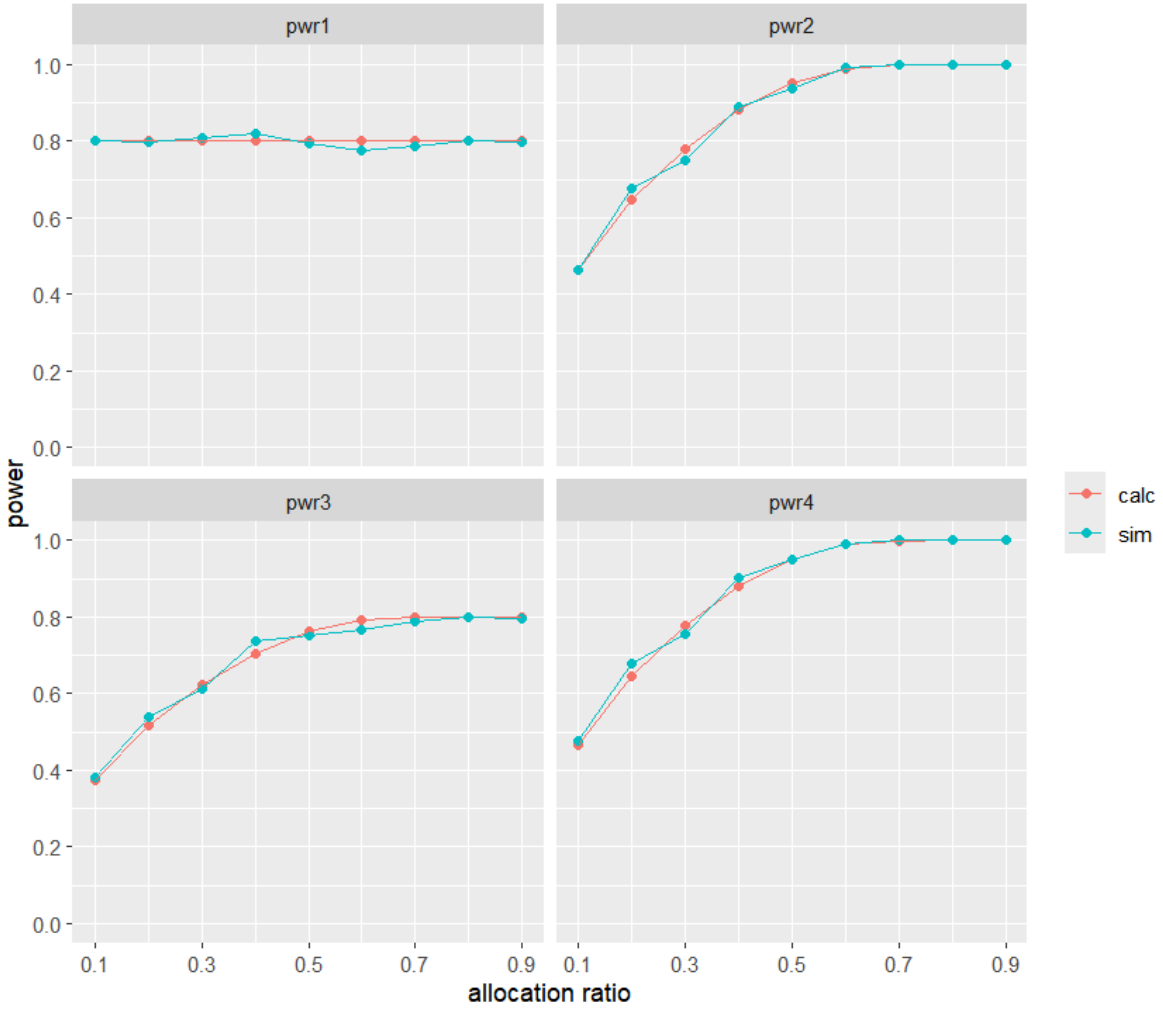
高优

```
a <- getPwr_Con_Equi_JM1(delta_j = -0.2, delta_a = -0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = NA, N = 400, r = 1, sim = FALSE)
b <- getPwr_Con_Equi_JM1(delta_j = -0.2, delta_a = -0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = NA, N = 400, r = 1, sim = TRUE)
pplot(a, b)
```



低优

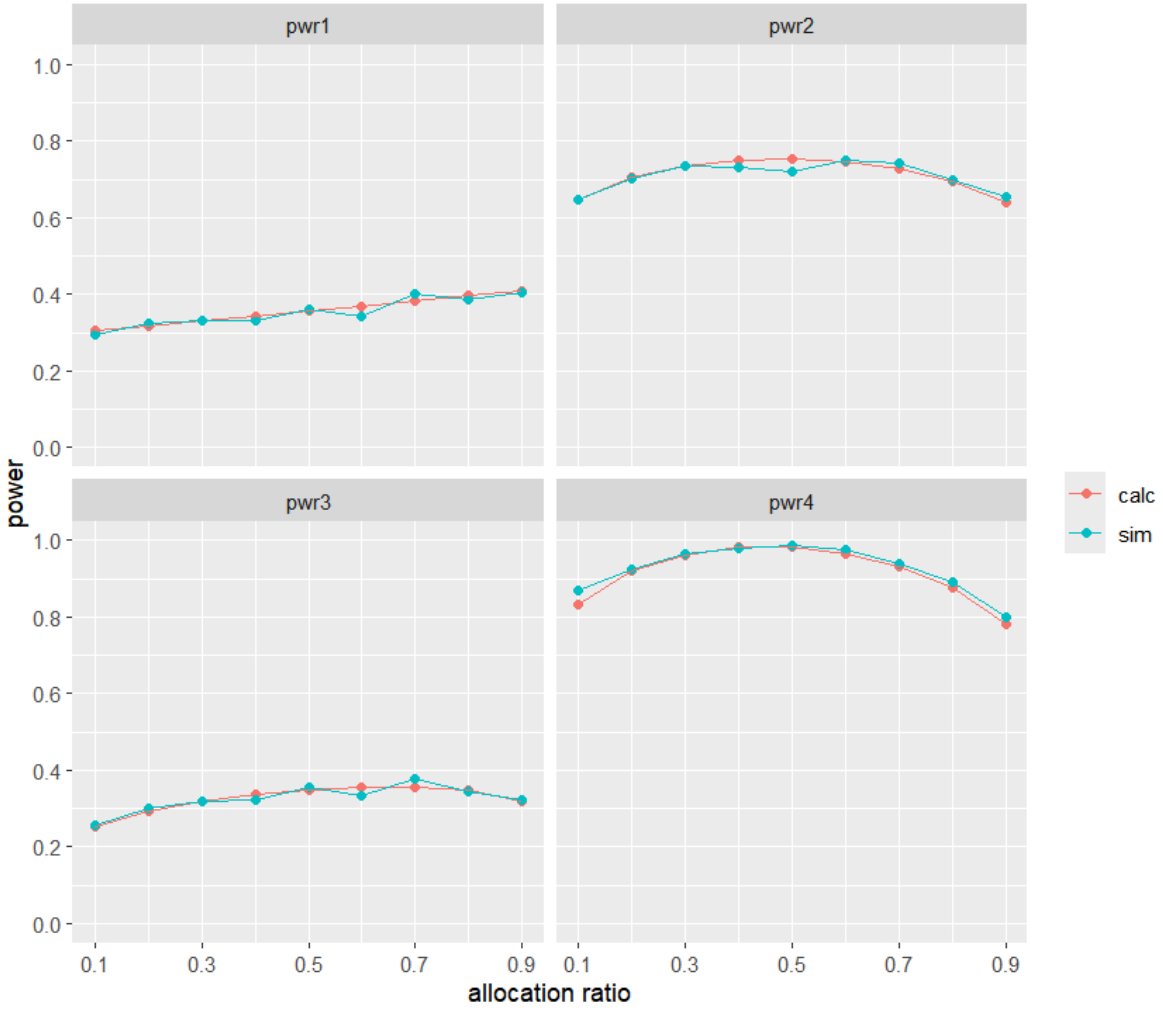
```
c <- getPwr_Con_Equi_JM1(delta_j = 0.2, delta_nj = 0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)
d <- getPwr_Con_Equi_JM1(delta_j = 0.2, delta_nj = 0.1, sigma = 1, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.4, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)
pplot(c, d)
```



getPwr_Con_Super_JM2

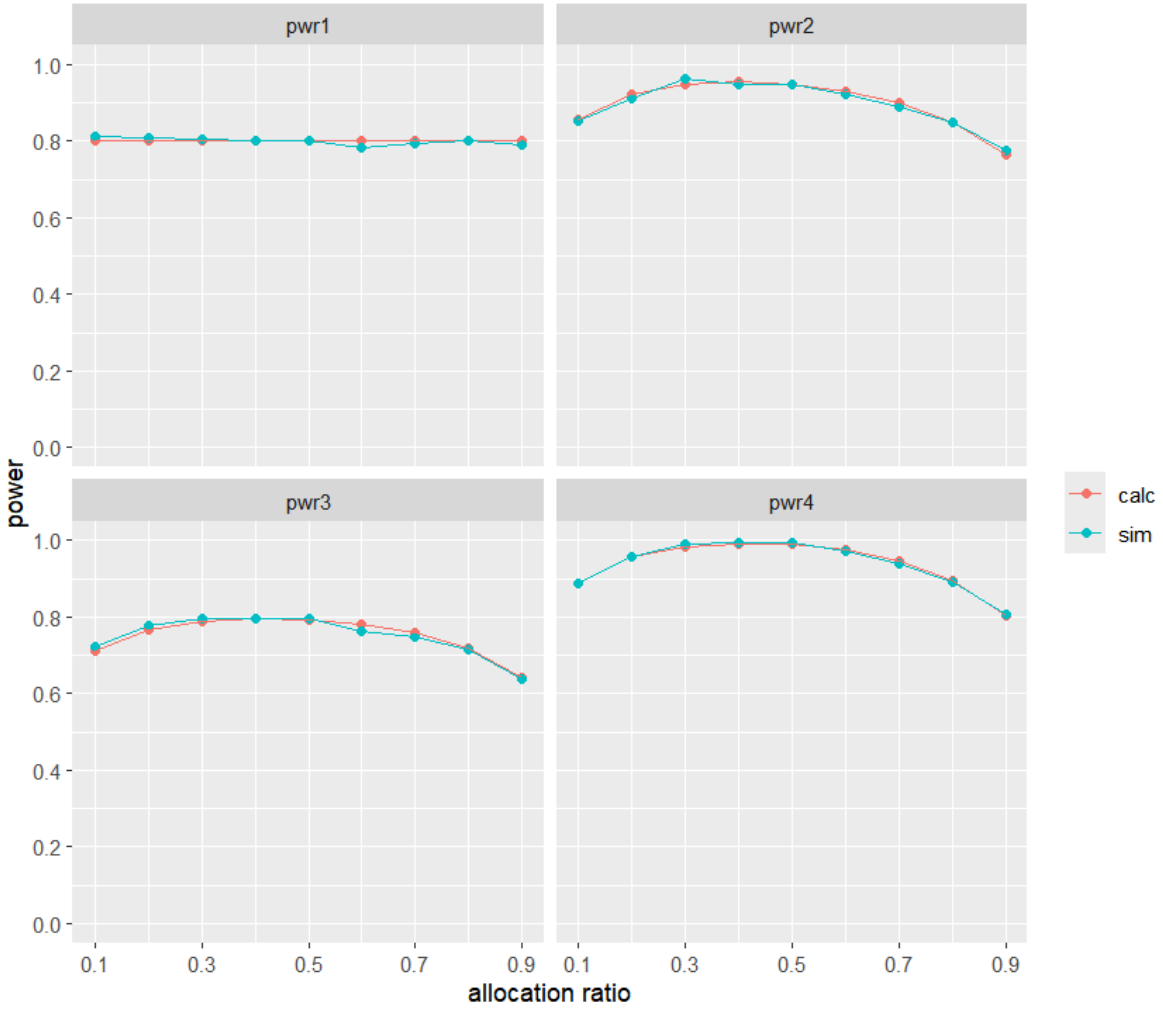
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Super_JM2(delta_i = c(1, 0.8), sigma = 4, fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 200, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Super_JM2(delta_i = c(1, 0.8), sigma = 4, fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 200, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

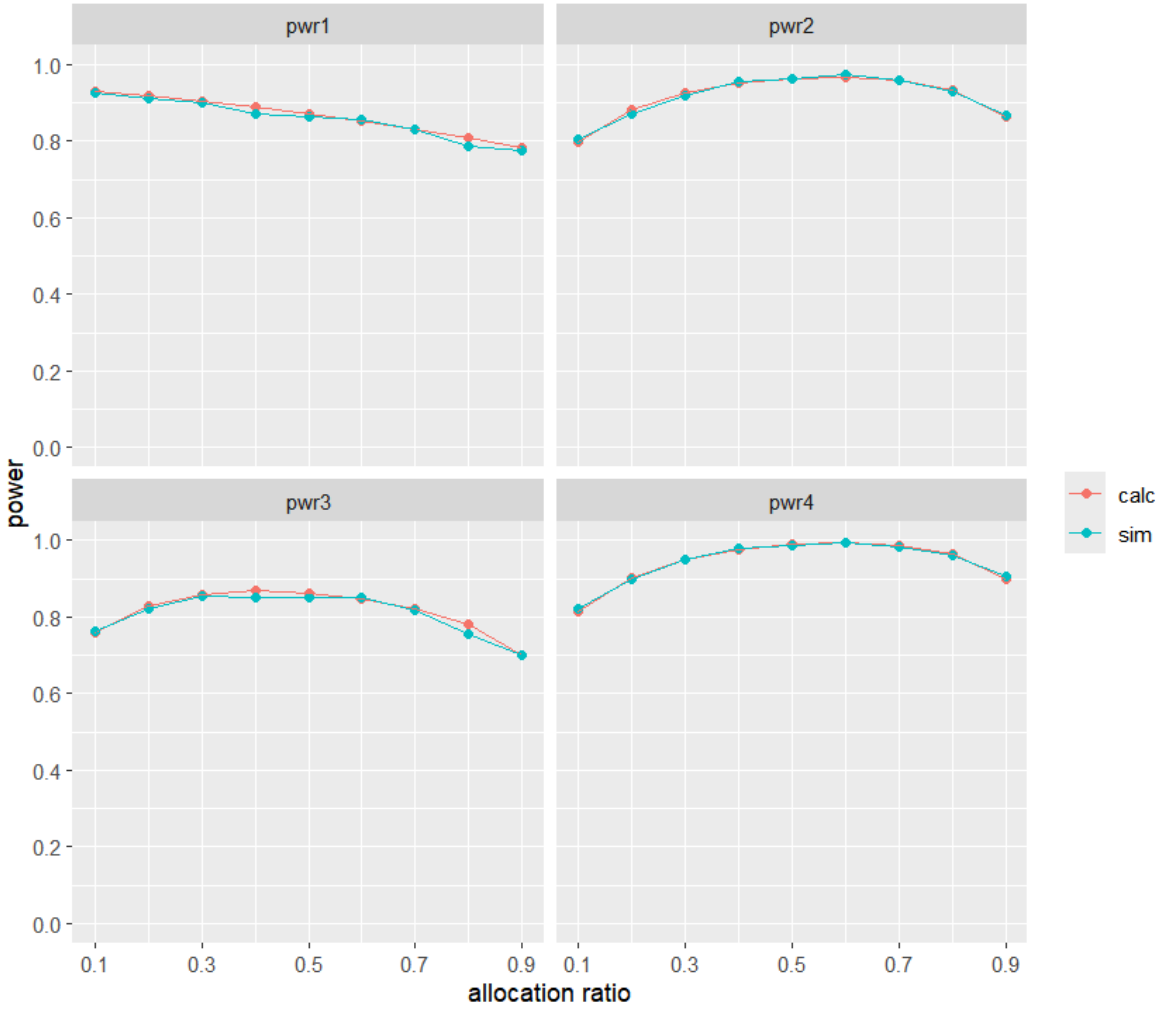
```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Super_JM2(delta_i = c(-1, -0.8), sigma = 4, fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Super_JM2(delta_i = c(-1, -0.8), sigma = 4, fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```



getPwr_Con_Noninf_JM2

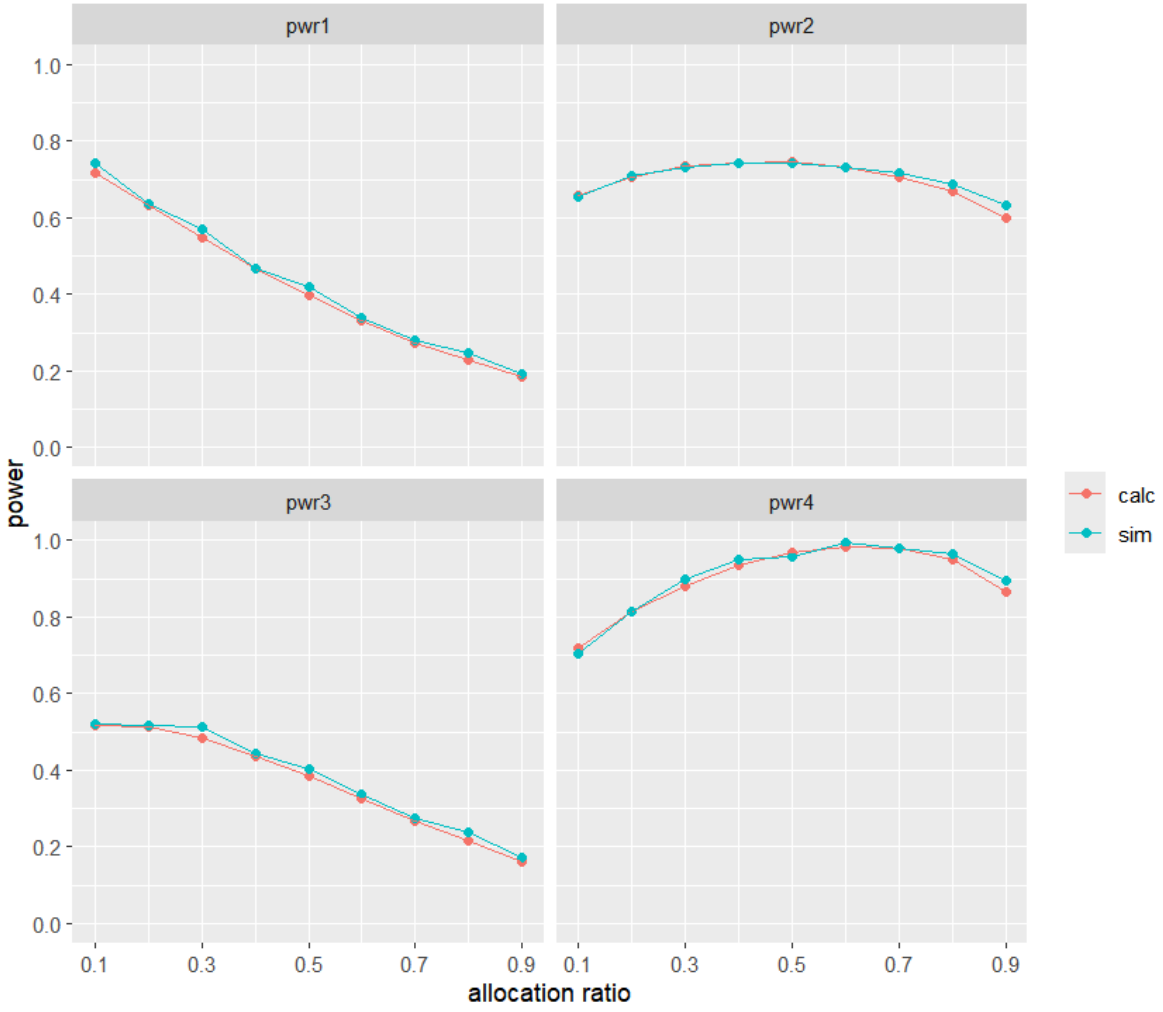
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Noninf_JM2(delta_i = c(-0.5, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1, direct = 1, sim =
FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Noninf_JM2(delta_i = c(-0.5, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1, direct = 1, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

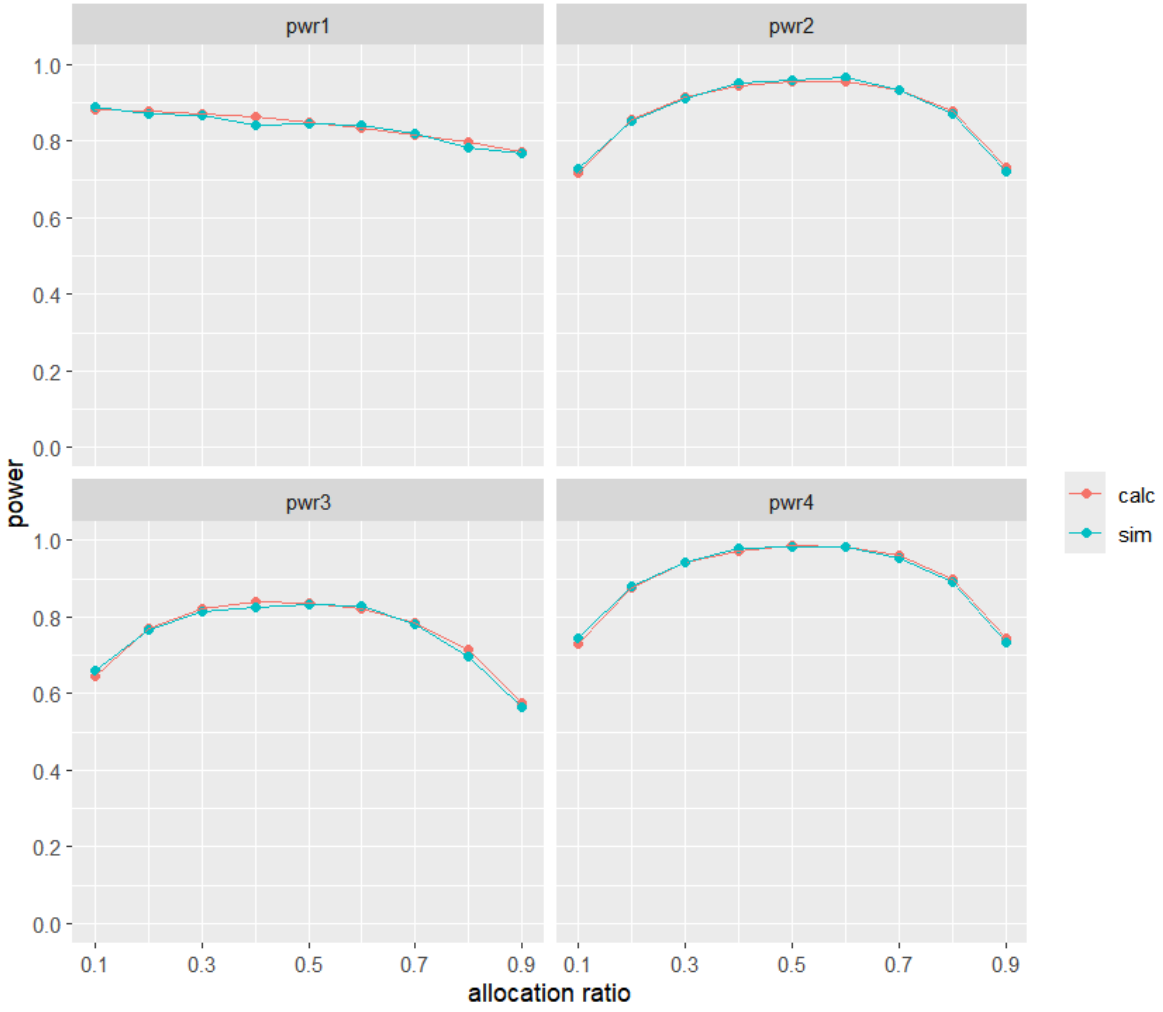
```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Noninf_JM2(delta_i = c(1, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim =
FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Noninf_JM2(delta_i = c(1, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```



getPwr_Con_Equi_JM2

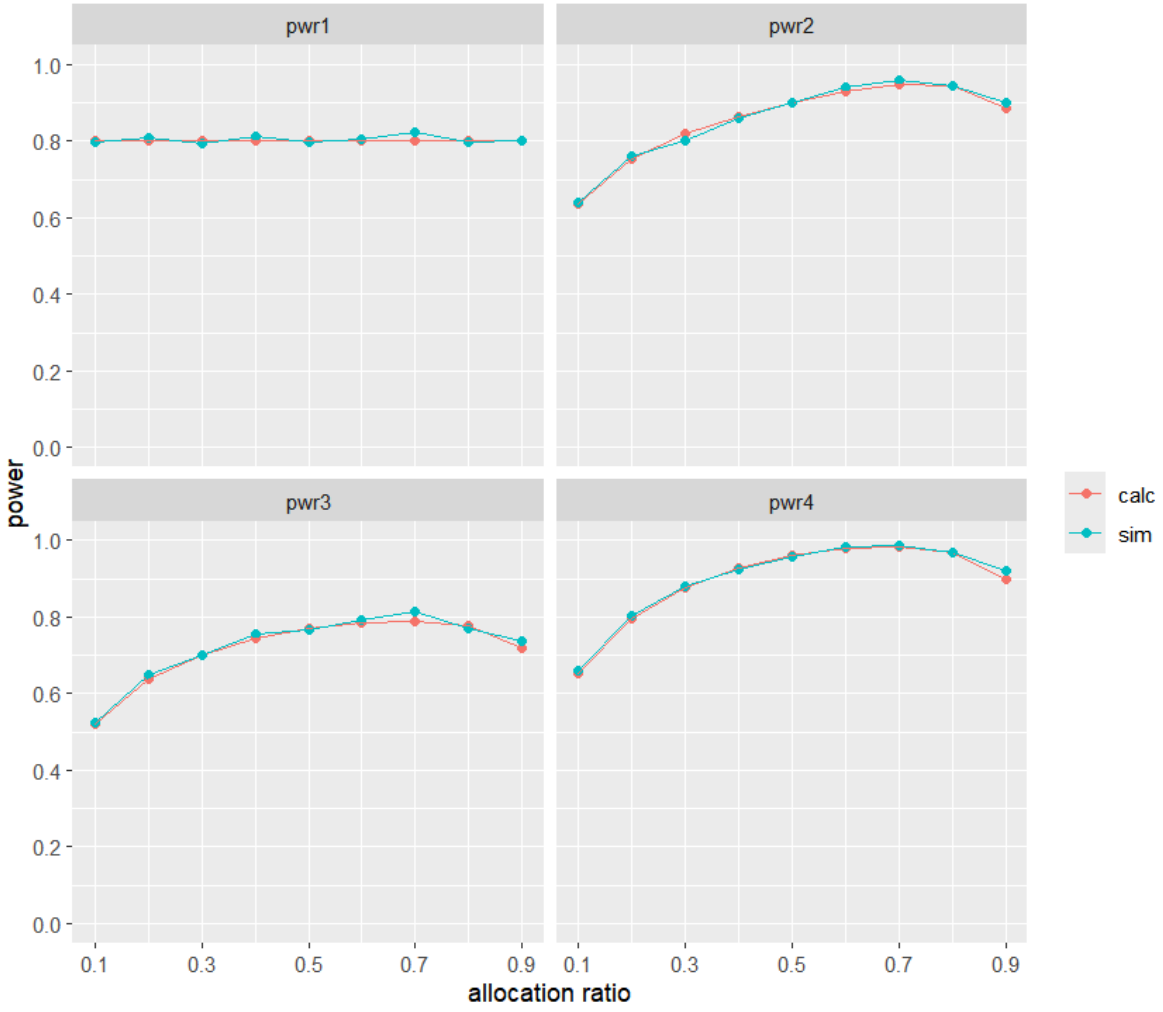
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Equi_JM2(delta_i = c(-0.5, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Equi_JM2(delta_i = c(-0.5, 0), sigma = 4, fi = c(f, 1 - f),
cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Equi_JM2(delta_i = c(1, 0), sigma = 4, fi = c(f, 1 - f), cut
= 2, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Con_Equi_JM2(delta_i = c(1, 0), sigma = 4, fi = c(f, 1 - f), cut
= 2, alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```

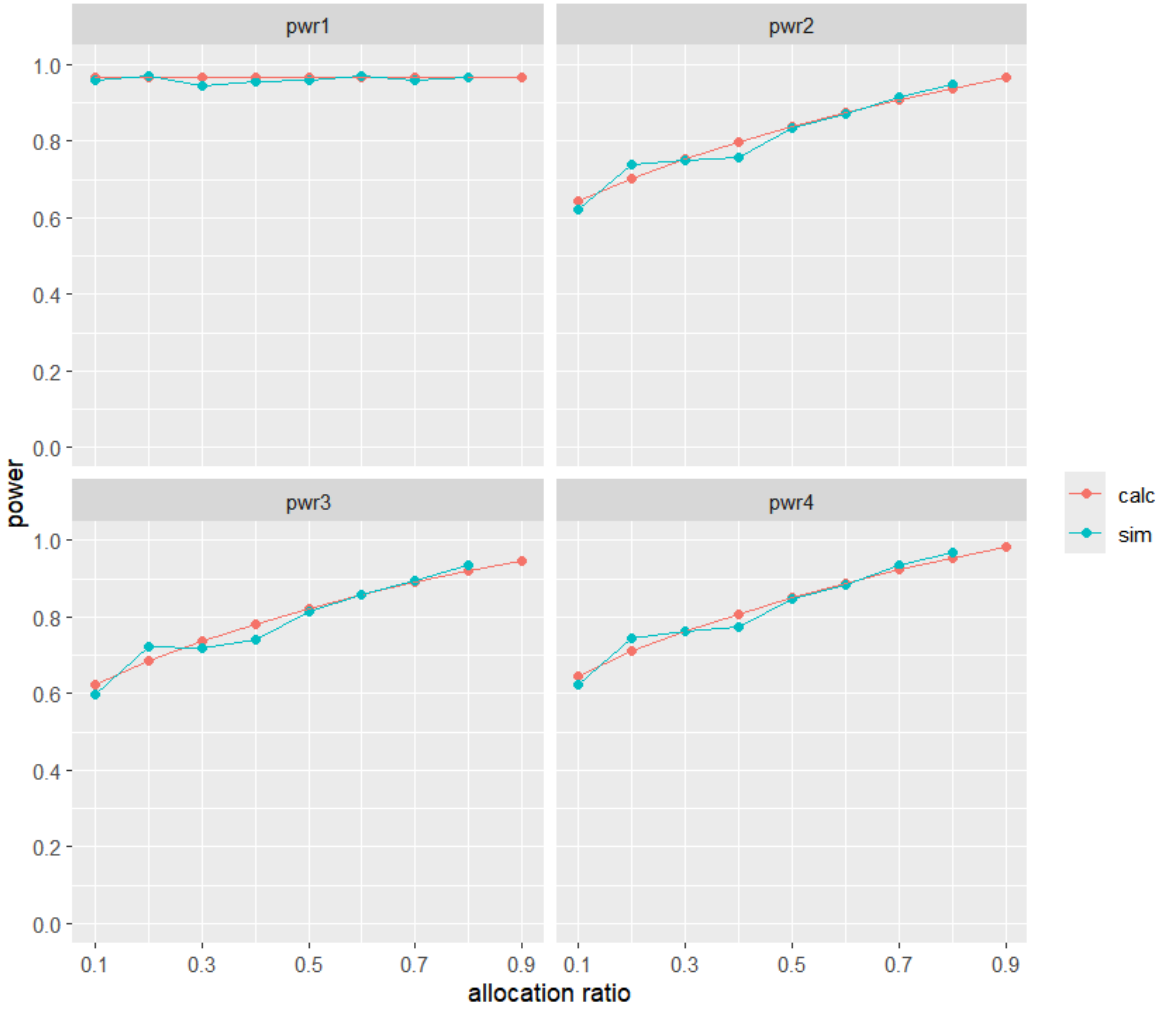


getPwr_Bin_Super_JM1

高优

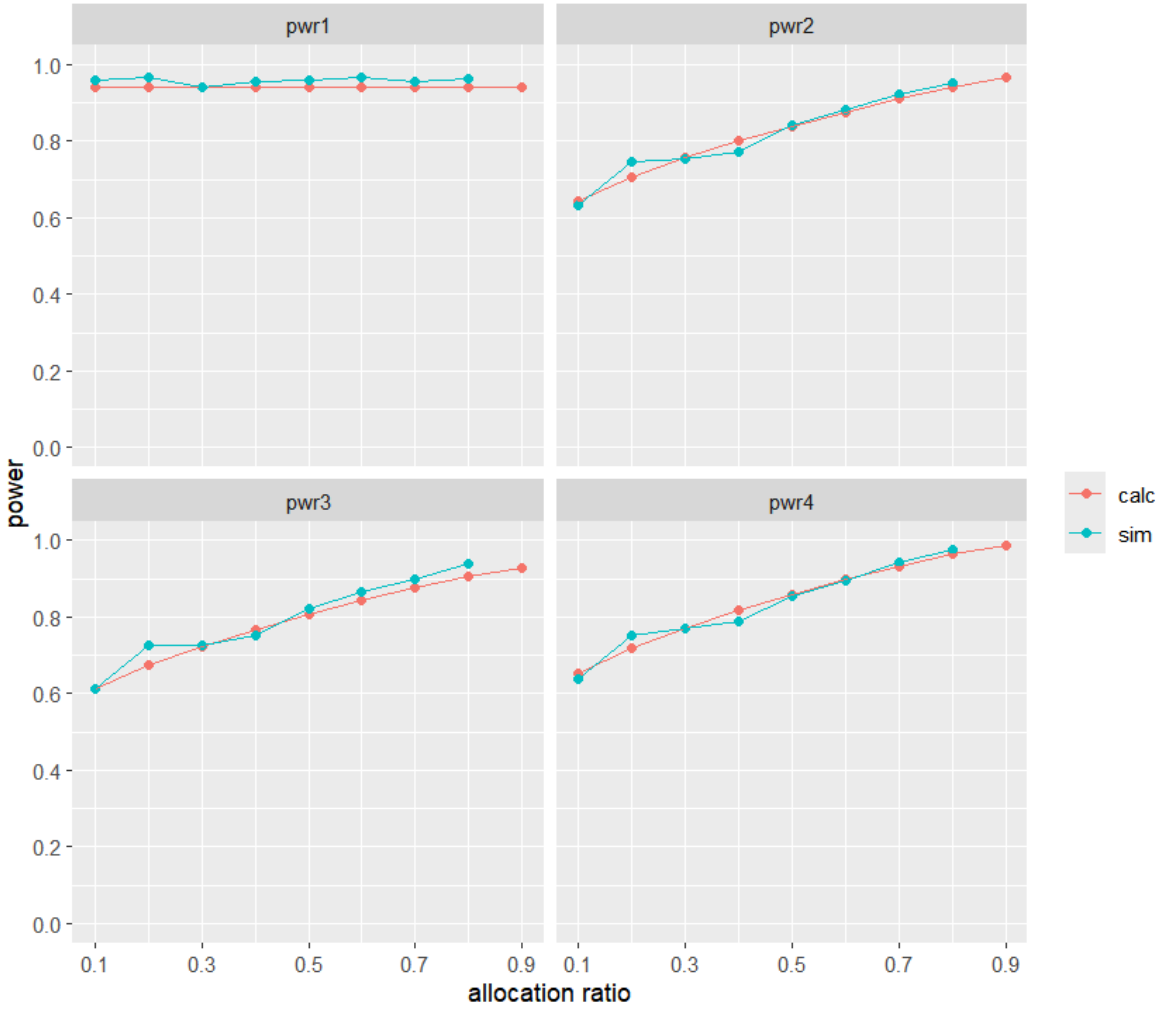
RD

```
a1 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RD", sim = FALSE)
b1 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RD", sim = TRUE)
pplot(a1, b1)
```



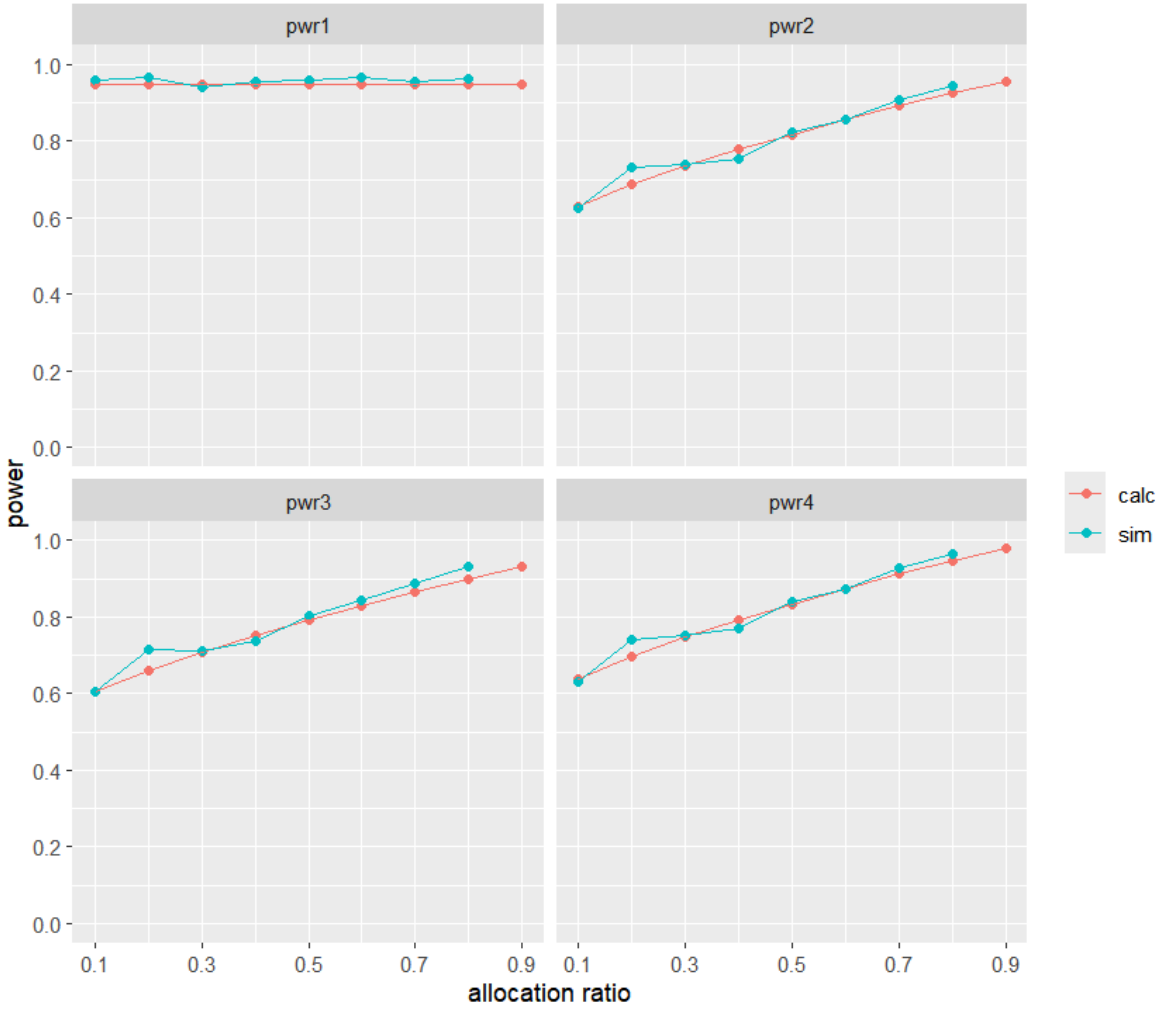
RR

```
a2 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RR", sim = FALSE)
b2 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RR", sim = TRUE)
pplot(a2, b2)
```



OR

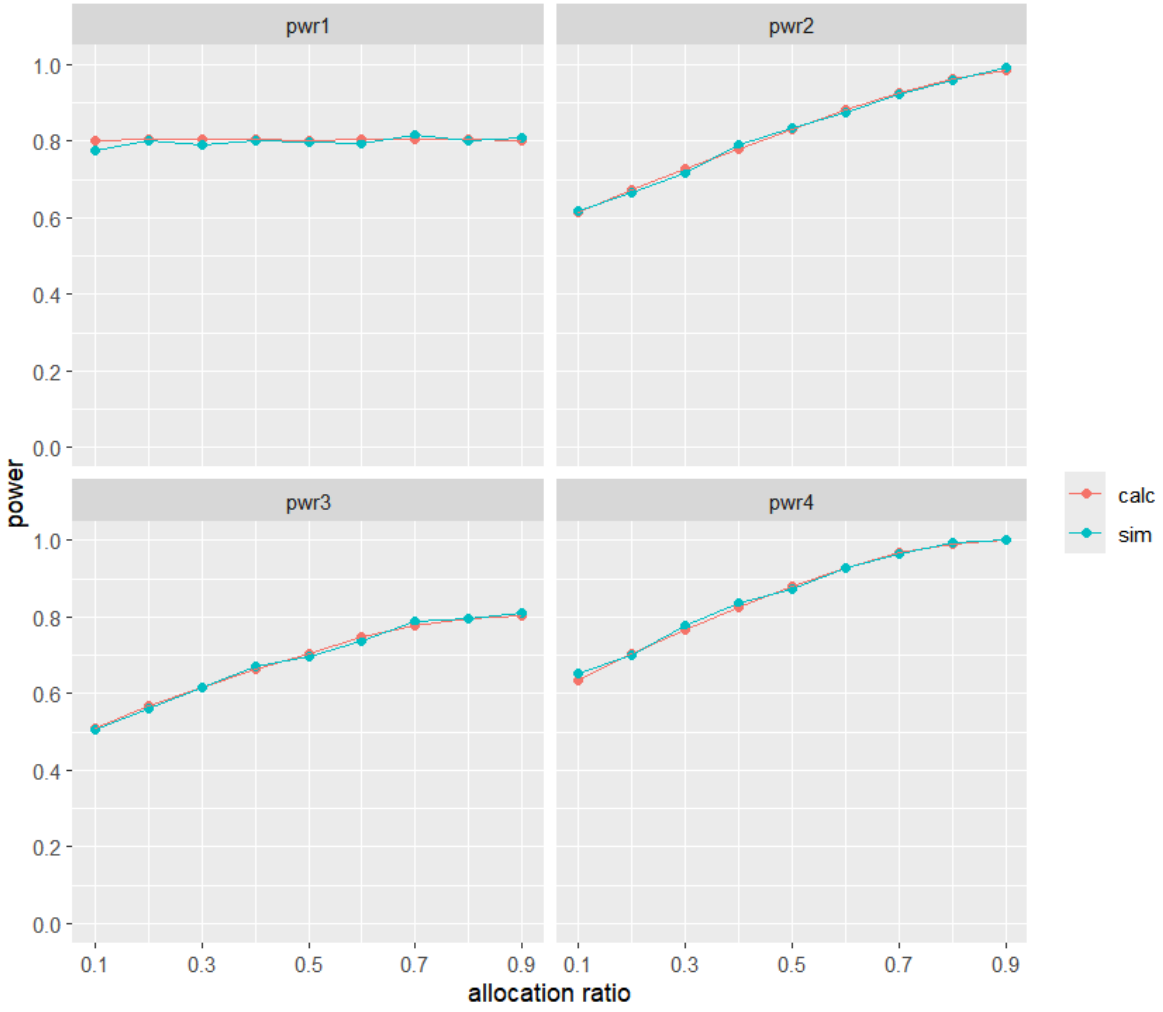
```
a3 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"OR", sim = FALSE)
b3 <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"OR", sim = TRUE)
pplot(a3, b3)
```



低优

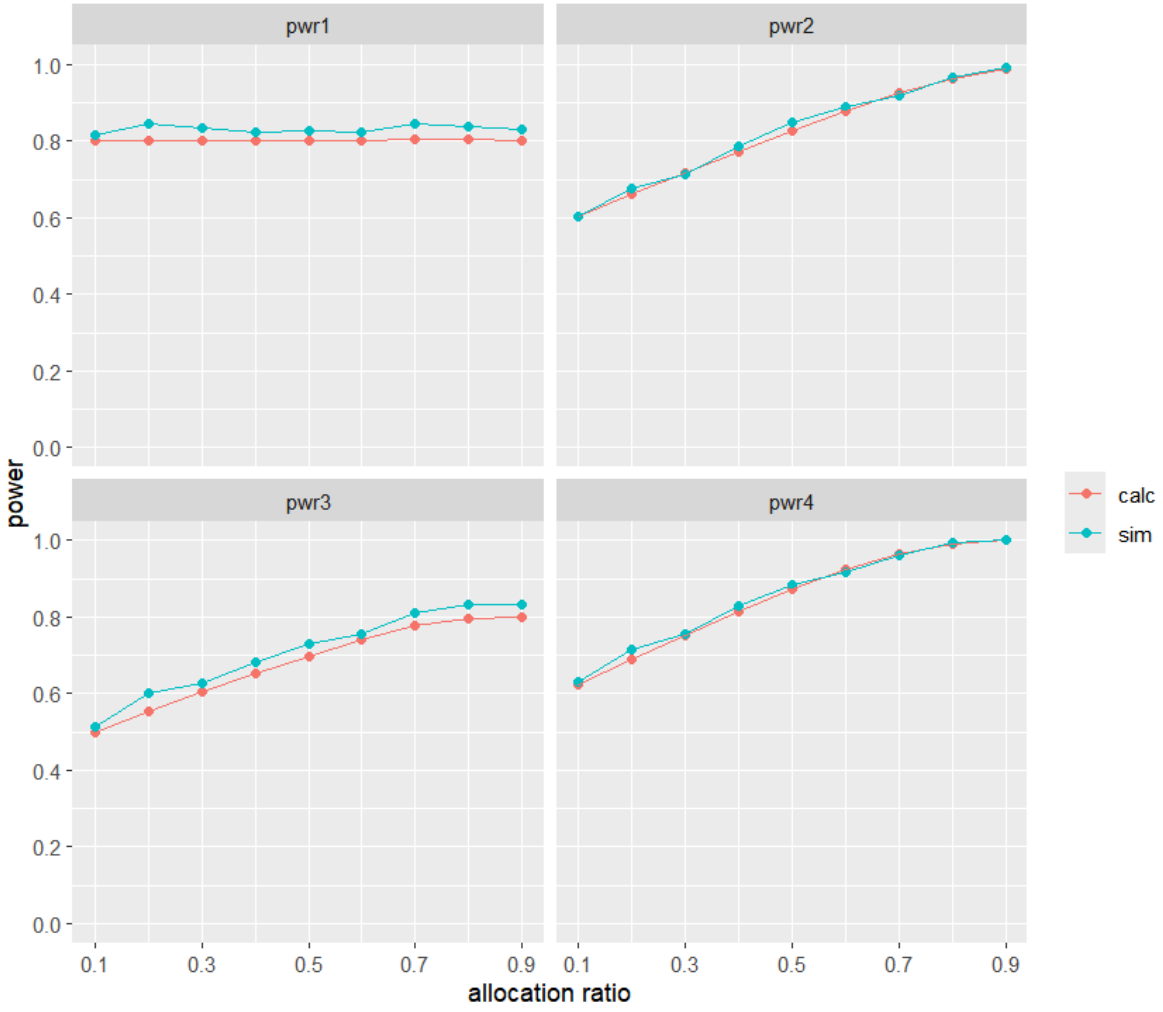
RD

```
c1 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"RD", sim = FALSE)
d1 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"RD", sim = TRUE)
pplot(c1, d1)
```



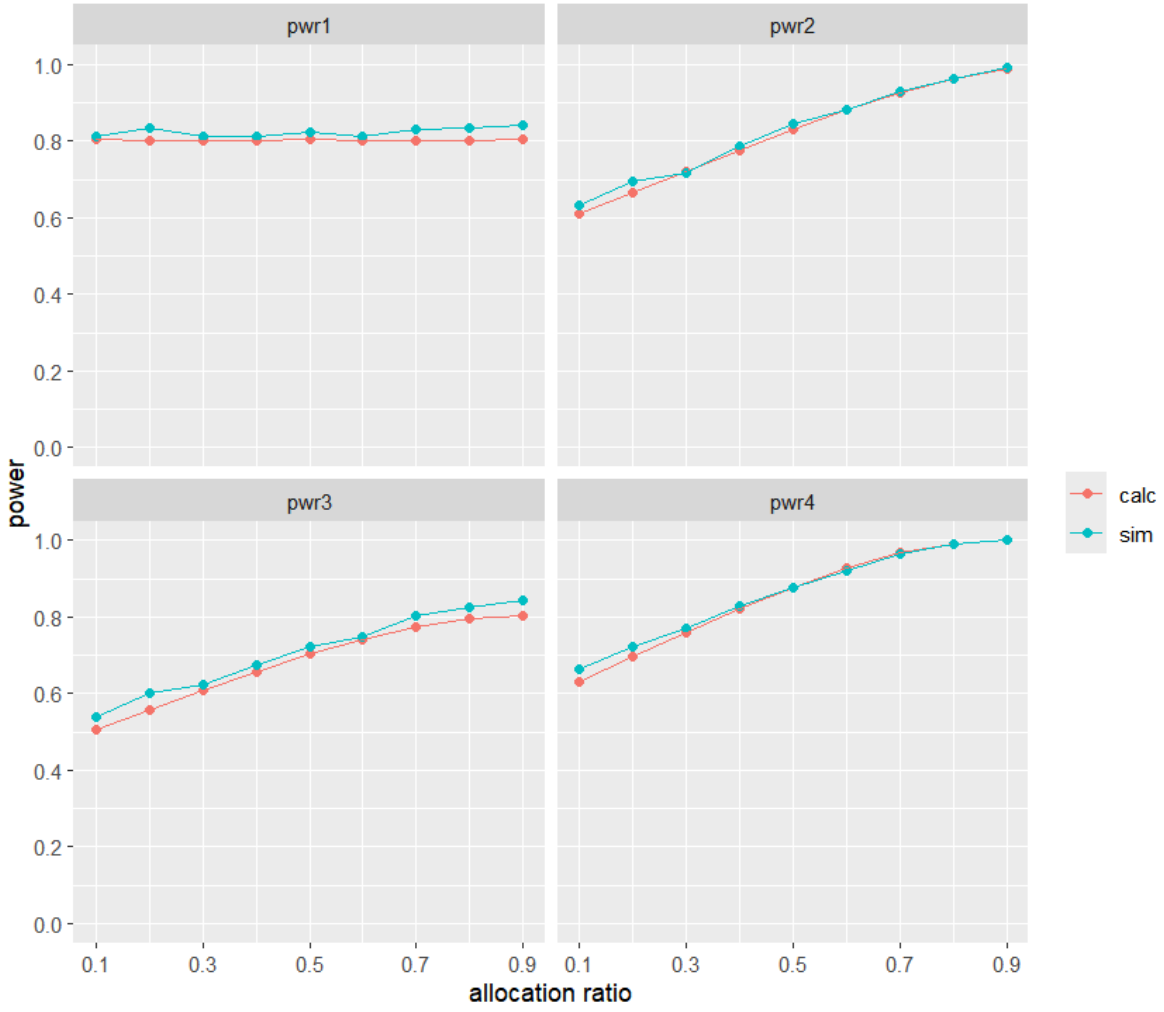
RR

```
c2 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"RR", sim = FALSE)
d2 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"RR", sim = TRUE)
pplot(c2, d2)
```



OR

```
c3 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"OR", sim = FALSE)
d3 <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_nj = 0.25, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale =
"OR", sim = TRUE)
pplot(c3, d3)
```

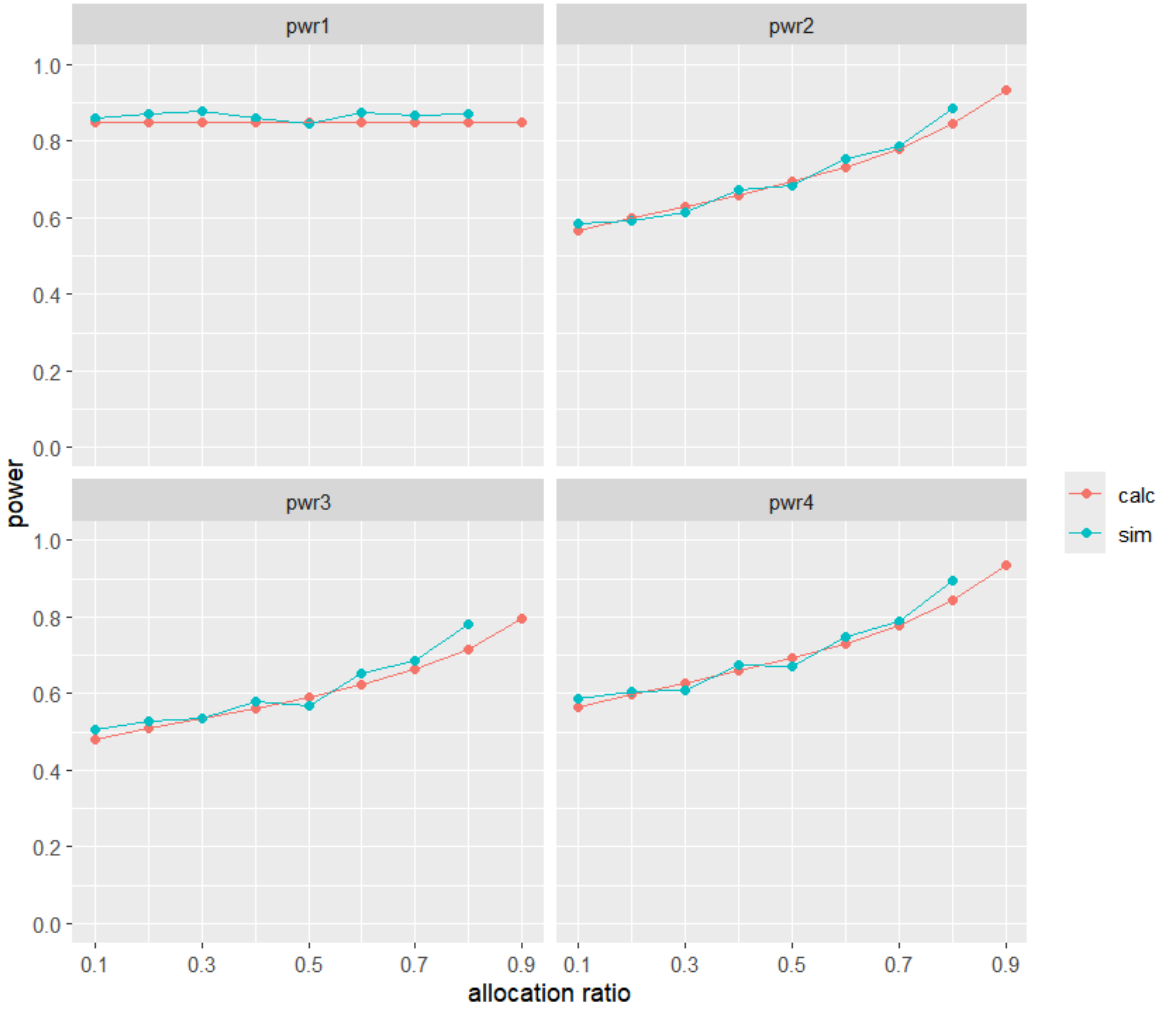


getPwr_Bin_Noninf_JM1

高优

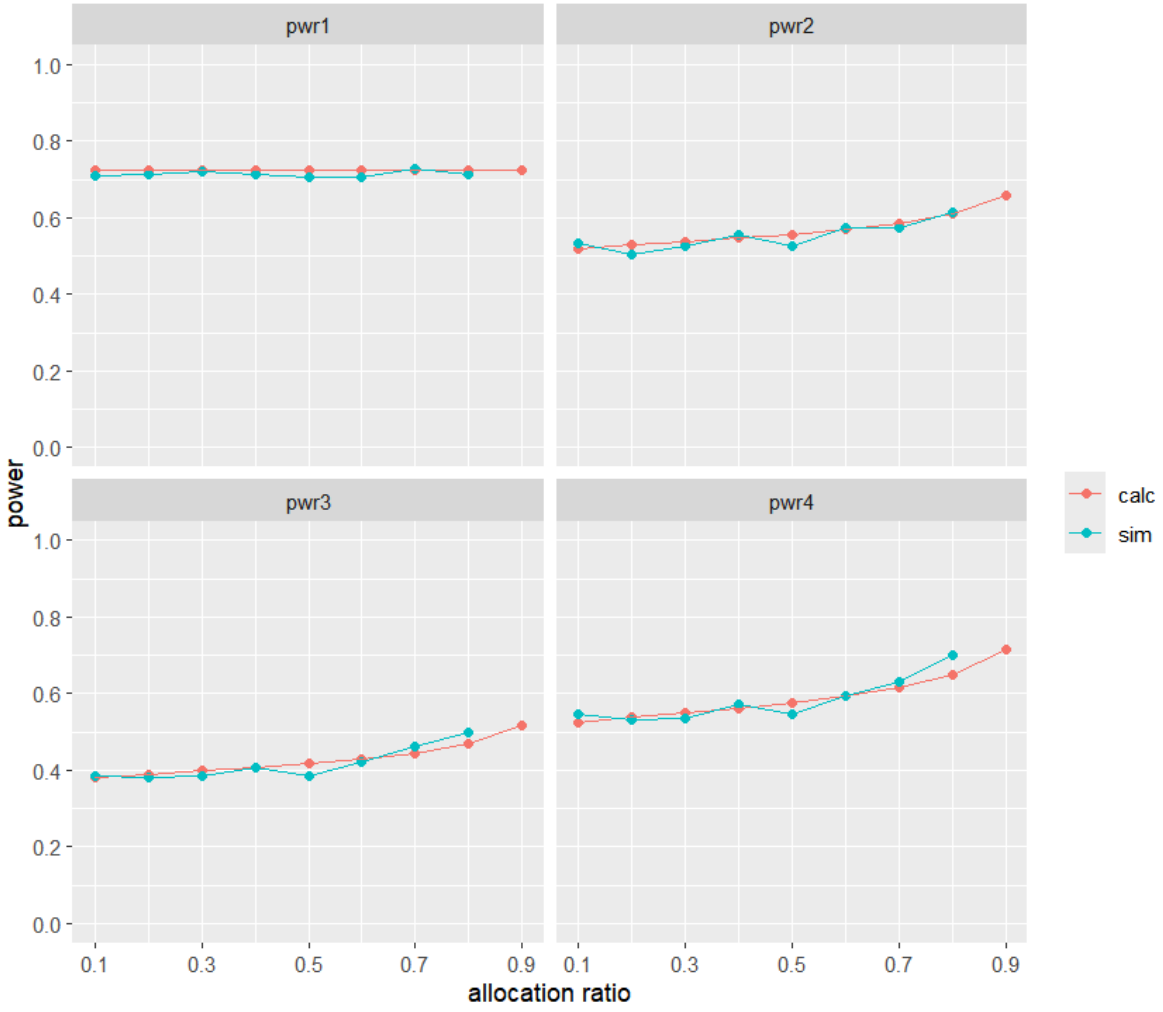
RD

```
a1 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 100, r =
1, scale = "RD", direct = 1, sim = FALSE)
b1 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 100, r =
1, scale = "RD", direct = 1, sim = TRUE)
pplot(a1, b1)
```



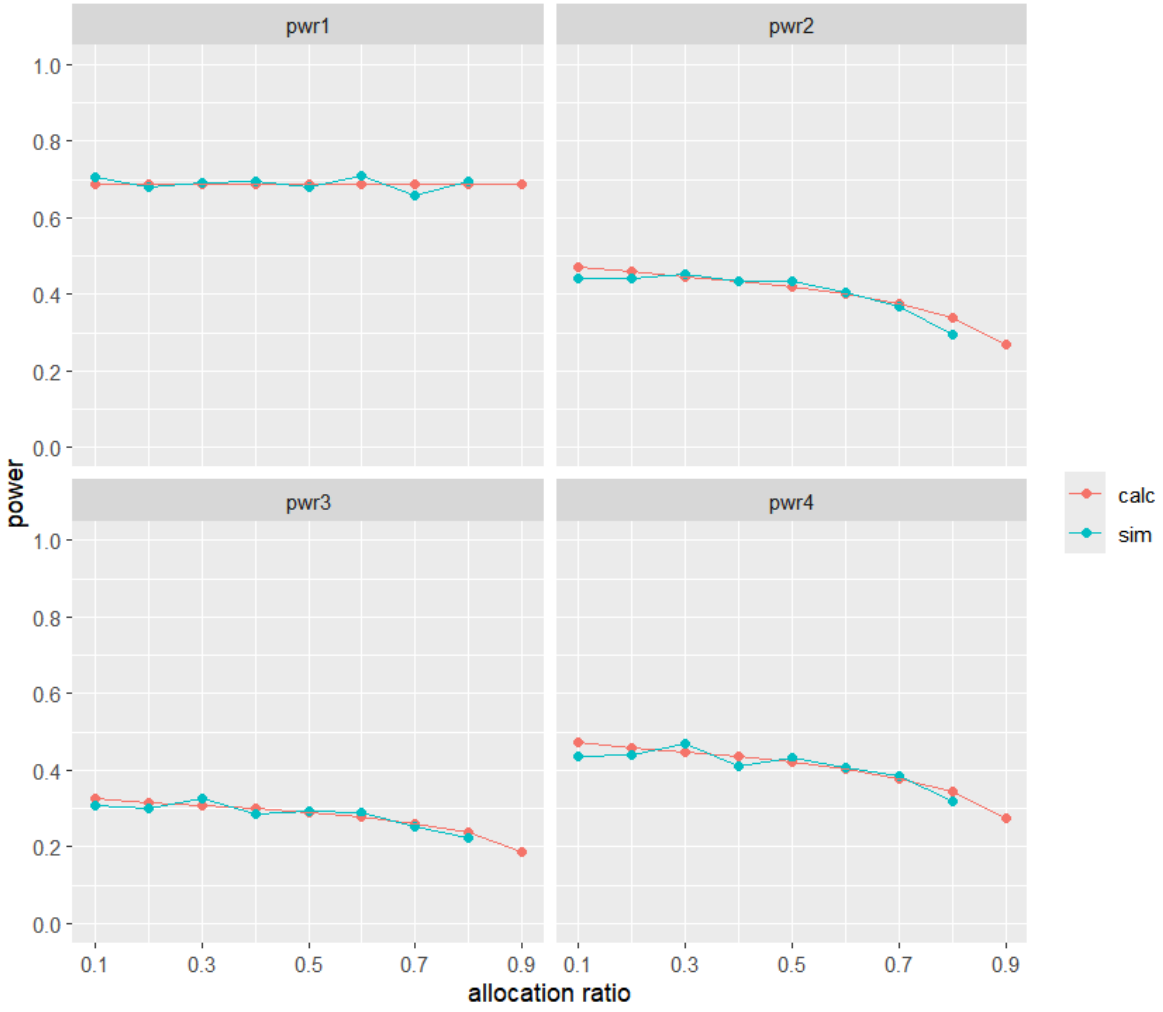
RR

```
a2 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 100,
r = 1, scale = "RR", direct = 1, sim = FALSE)
b2 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 100,
r = 1, scale = "RR", direct = 1, sim = TRUE)
pplot(a2, b2)
```



OR

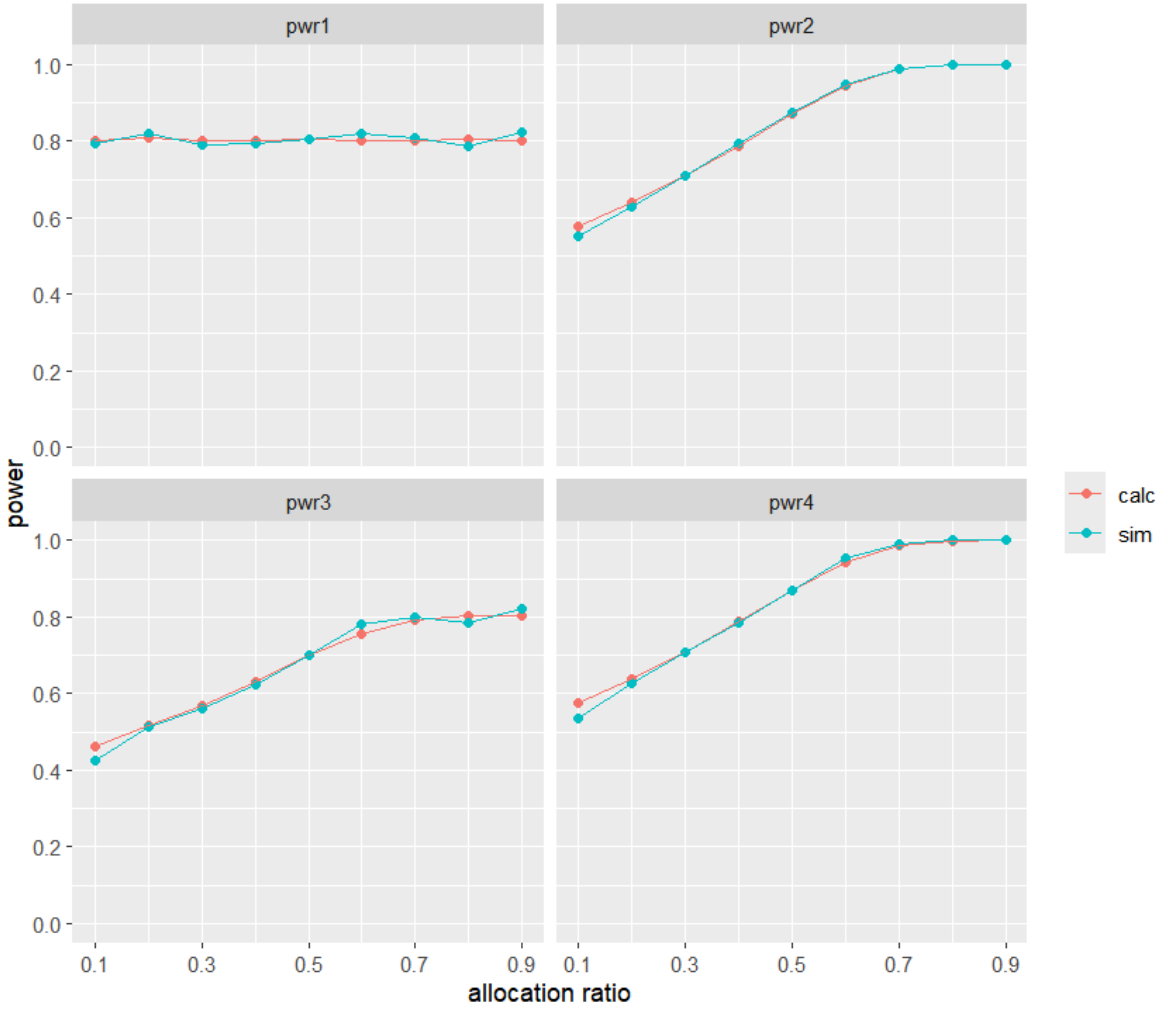
```
a3 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", direct = 1, sim = FALSE)
b3 <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", direct = 1, sim = TRUE)
pplot(a3, b3)
```



低优

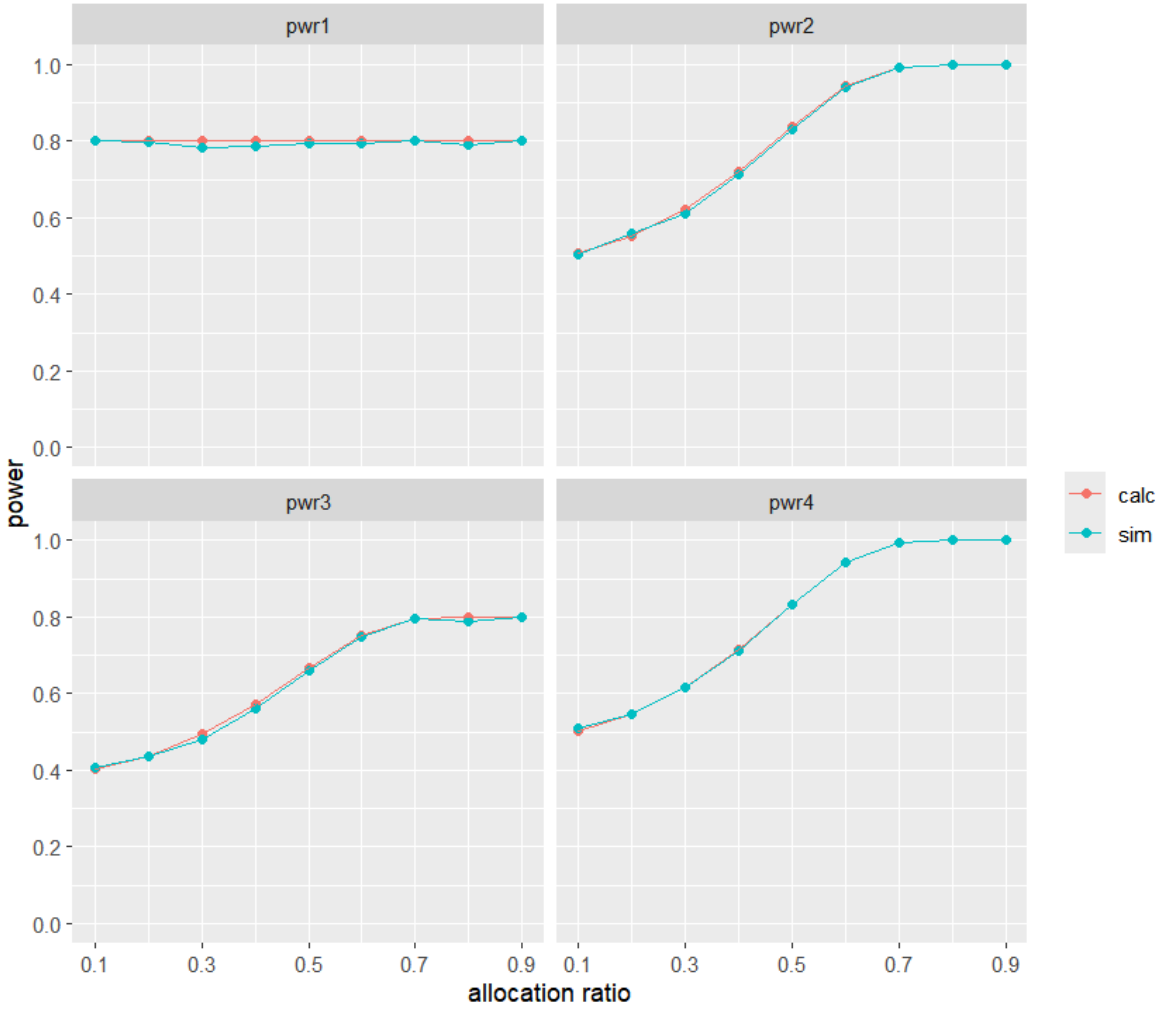
RD

```
c1 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = 0.2, N = NA, r =
1, scale = "RD", direct = -1, sim = FALSE)
d1 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = 0.2, N = NA, r =
1, scale = "RD", direct = -1, sim = TRUE)
pplot(c1, d1)
```



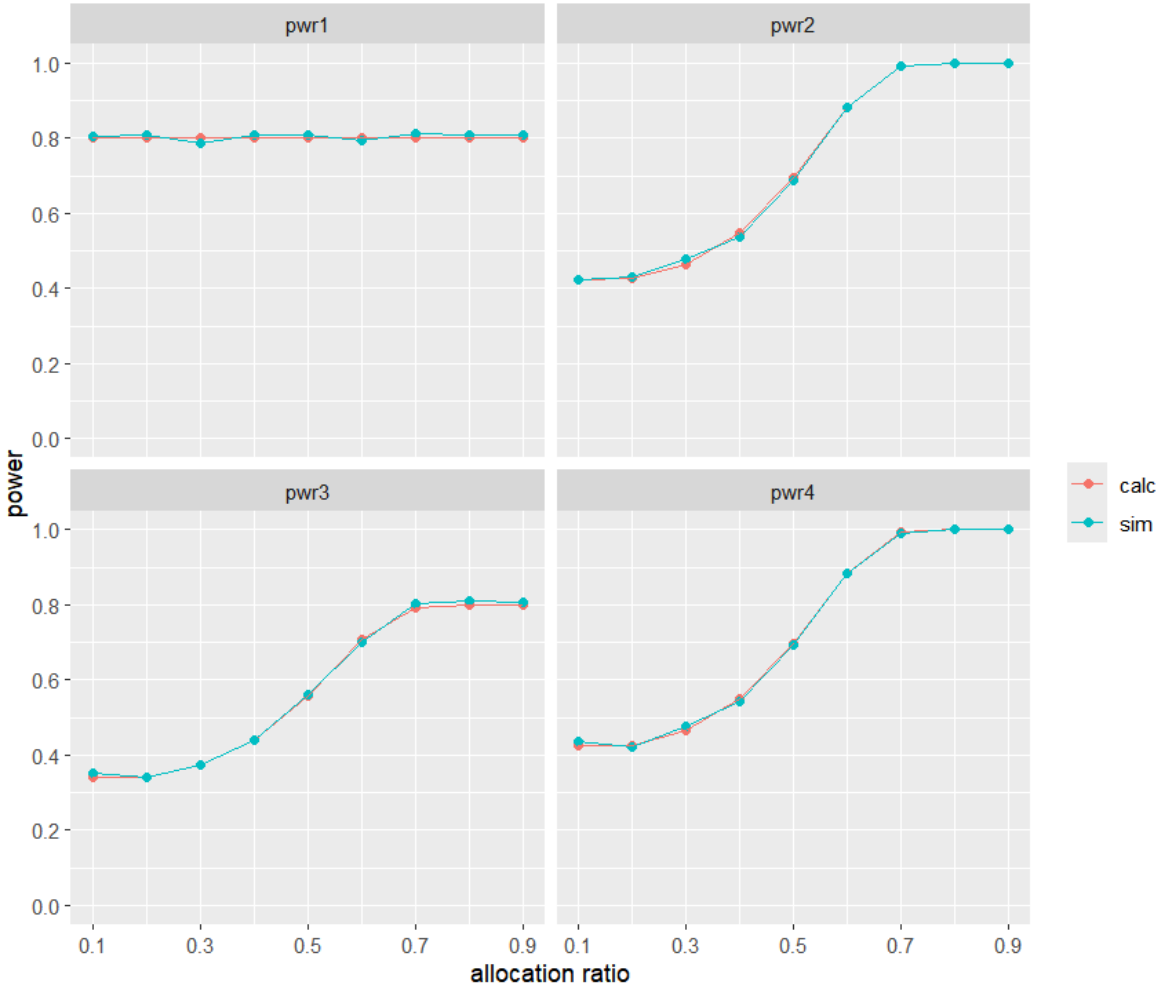
RR

```
c2 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "RR", direct = -1, sim = FALSE)
d2 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "RR", direct = -1, sim = TRUE)
pplot(c2, d2)
```



OR

```
c3 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "OR", direct = -1, sim = FALSE)
d3 <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "OR", direct = -1, sim = TRUE)
pplot(c3, d3)
```

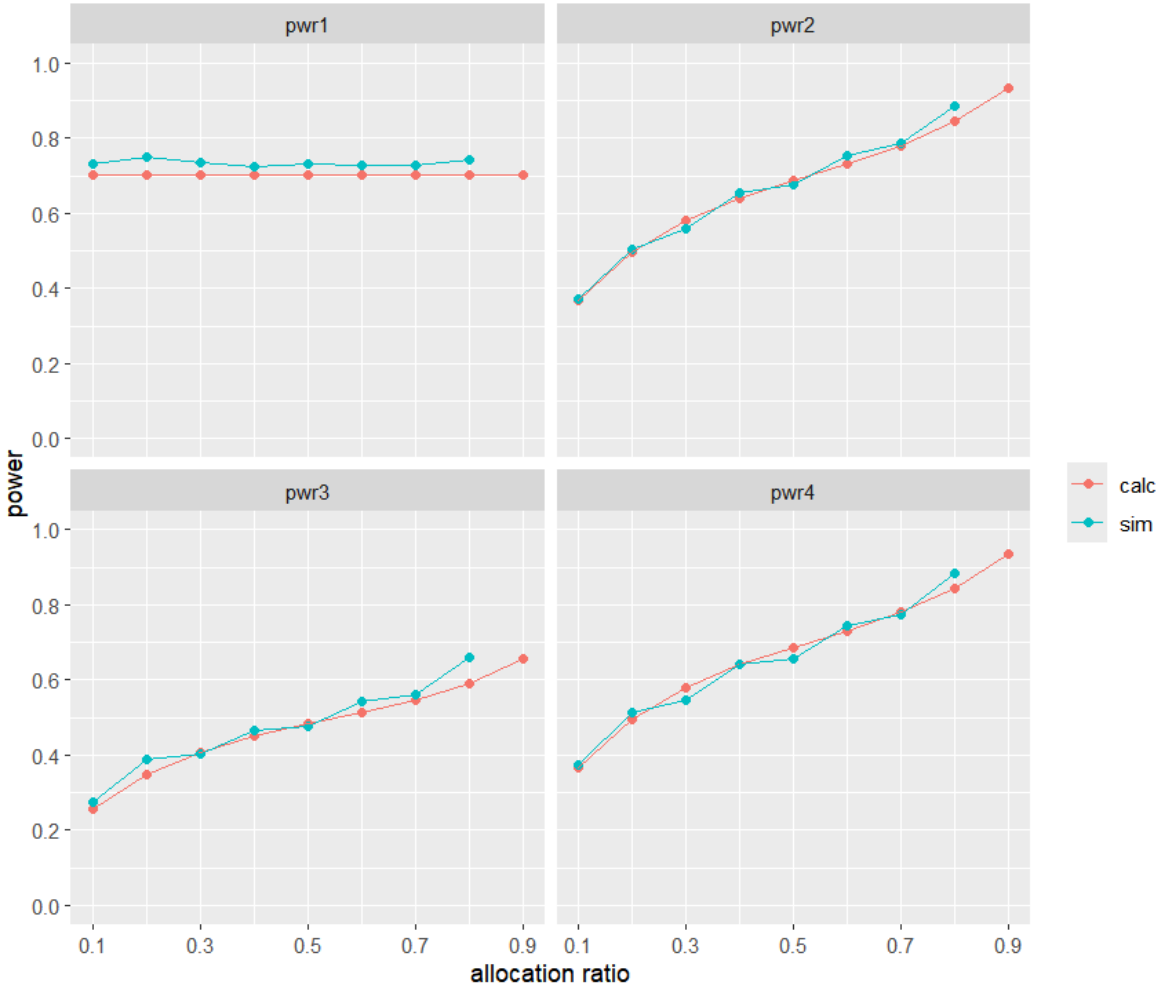


getPwr_Bin_Equi_JM1

高优

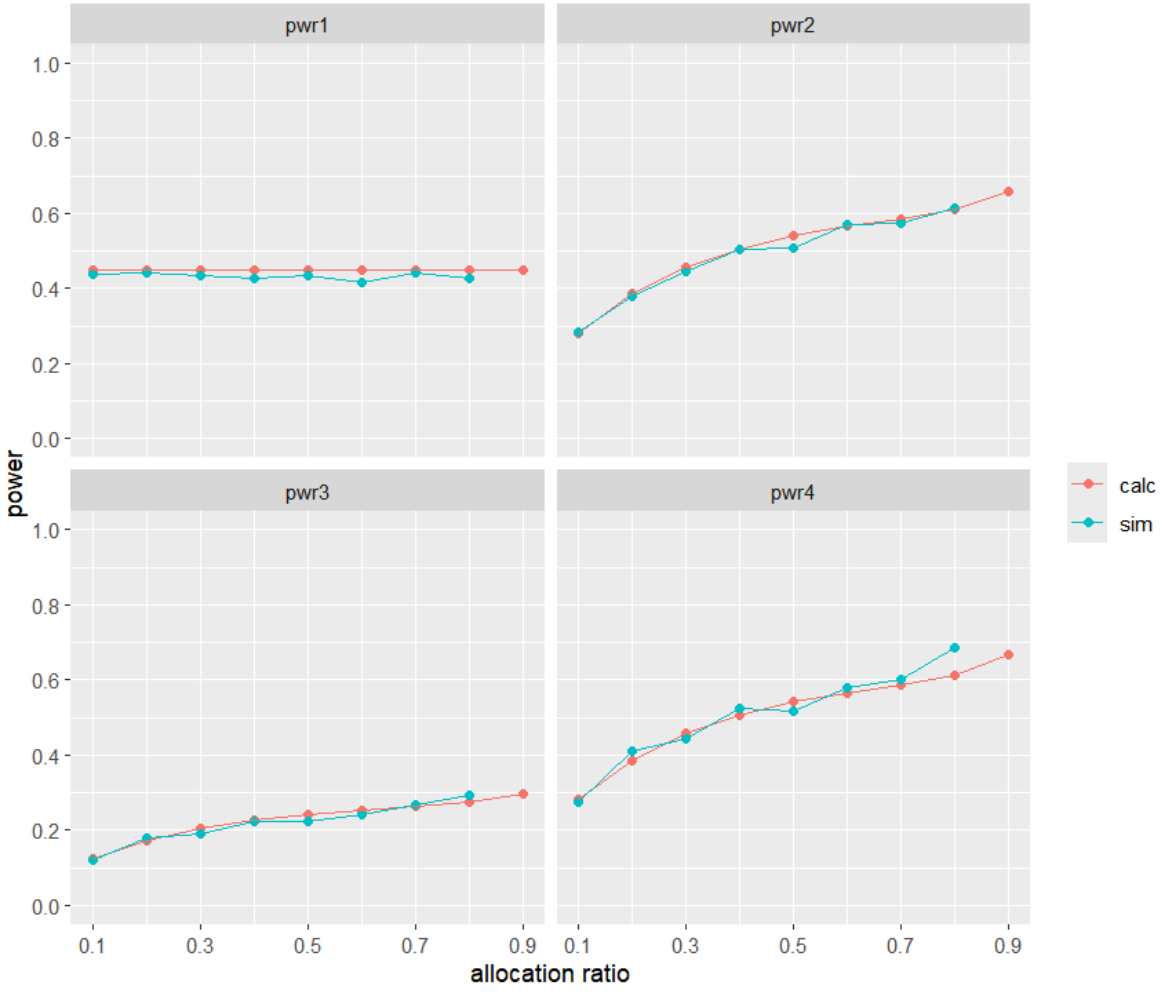
RD

```
a1 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 100, r =
1, scale = "RD", sim = FALSE)
b1 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 100, r =
1, scale = "RD", sim = TRUE)
pplot(a1, b1)
```



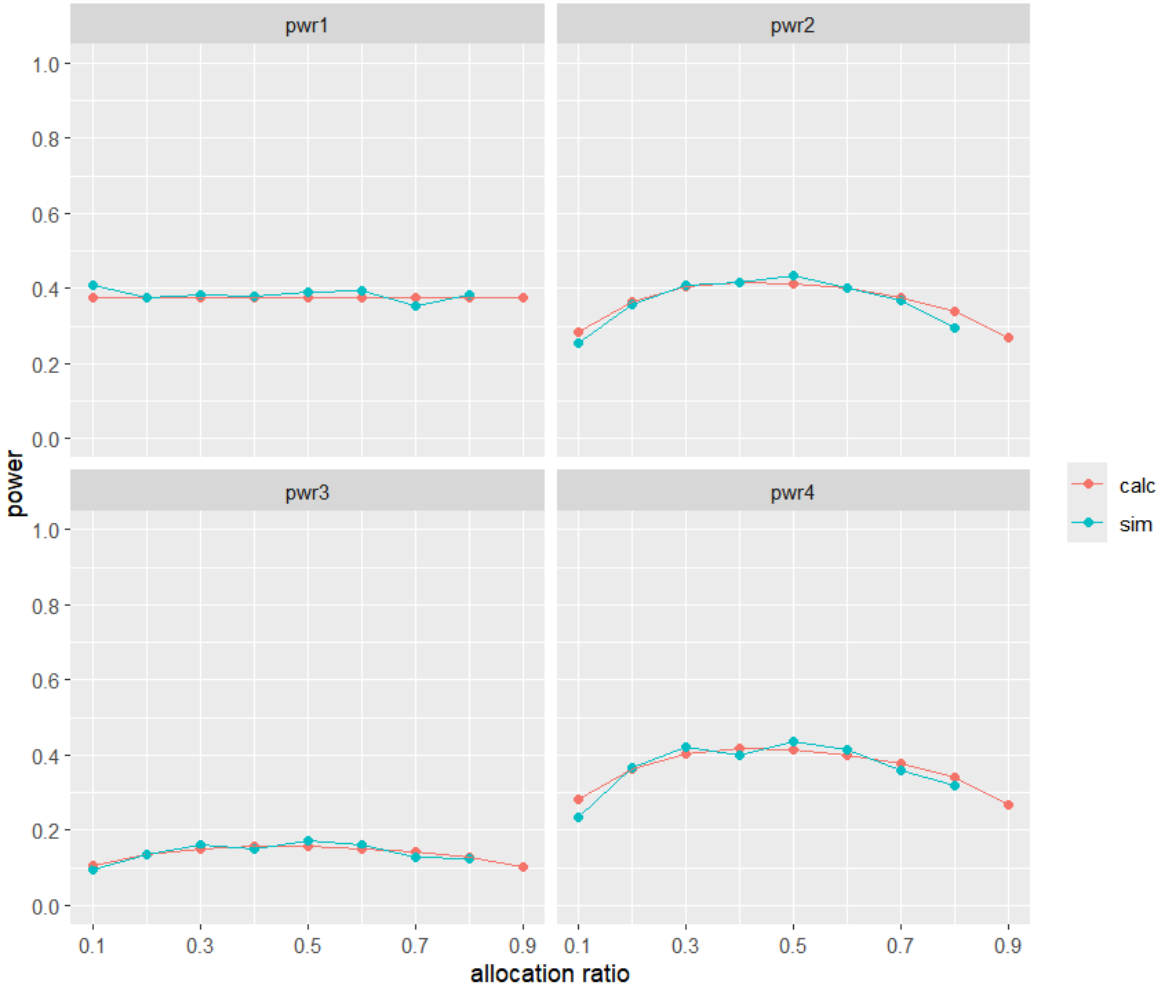
RR

```
a2 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 100,
r = 1, scale = "RR", sim = FALSE)
b2 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 100,
r = 1, scale = "RR", sim = TRUE)
pplot(a2, b2)
```



OR

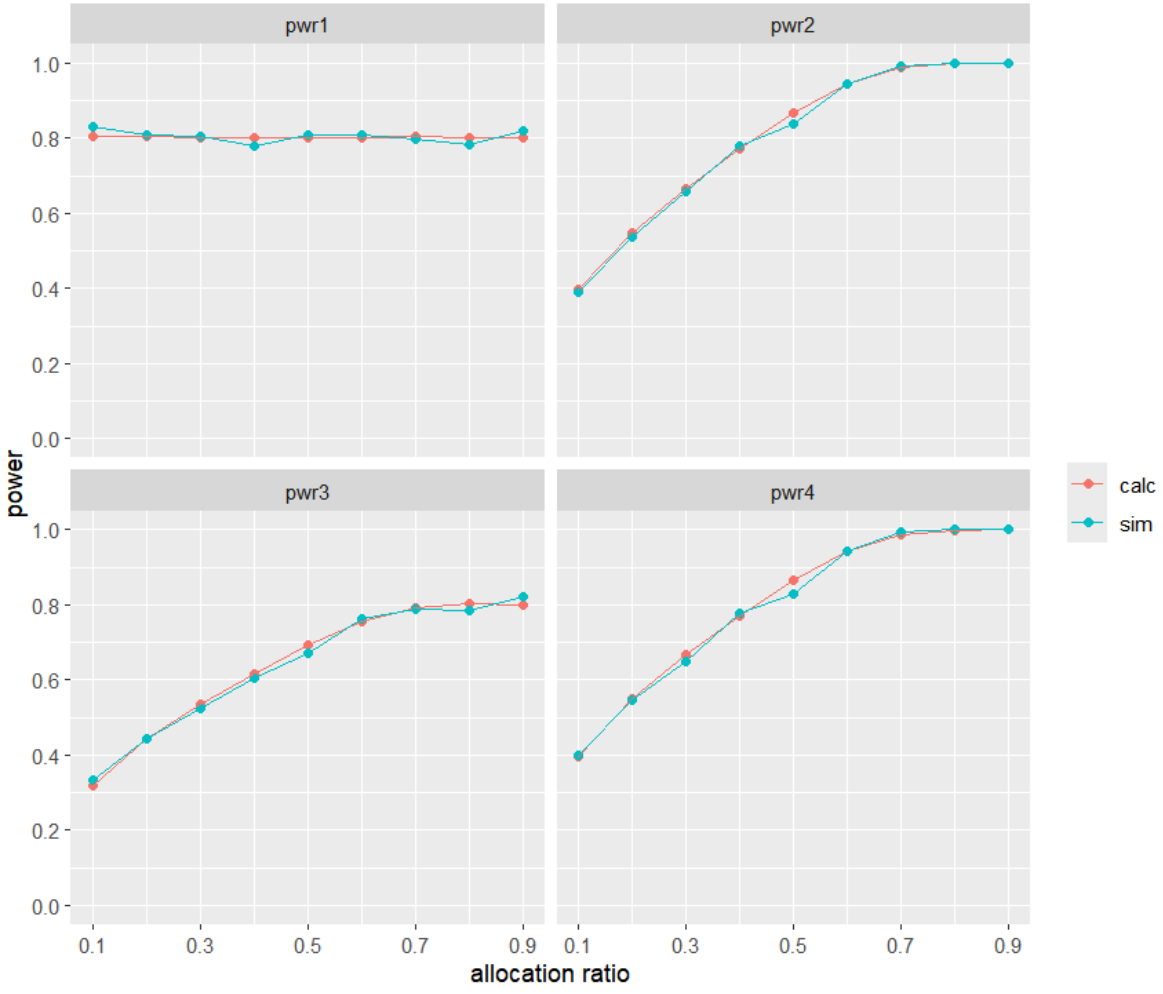
```
a3 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", sim = FALSE)
b3 <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", sim = TRUE)
pplot(a3, b3)
```



低优

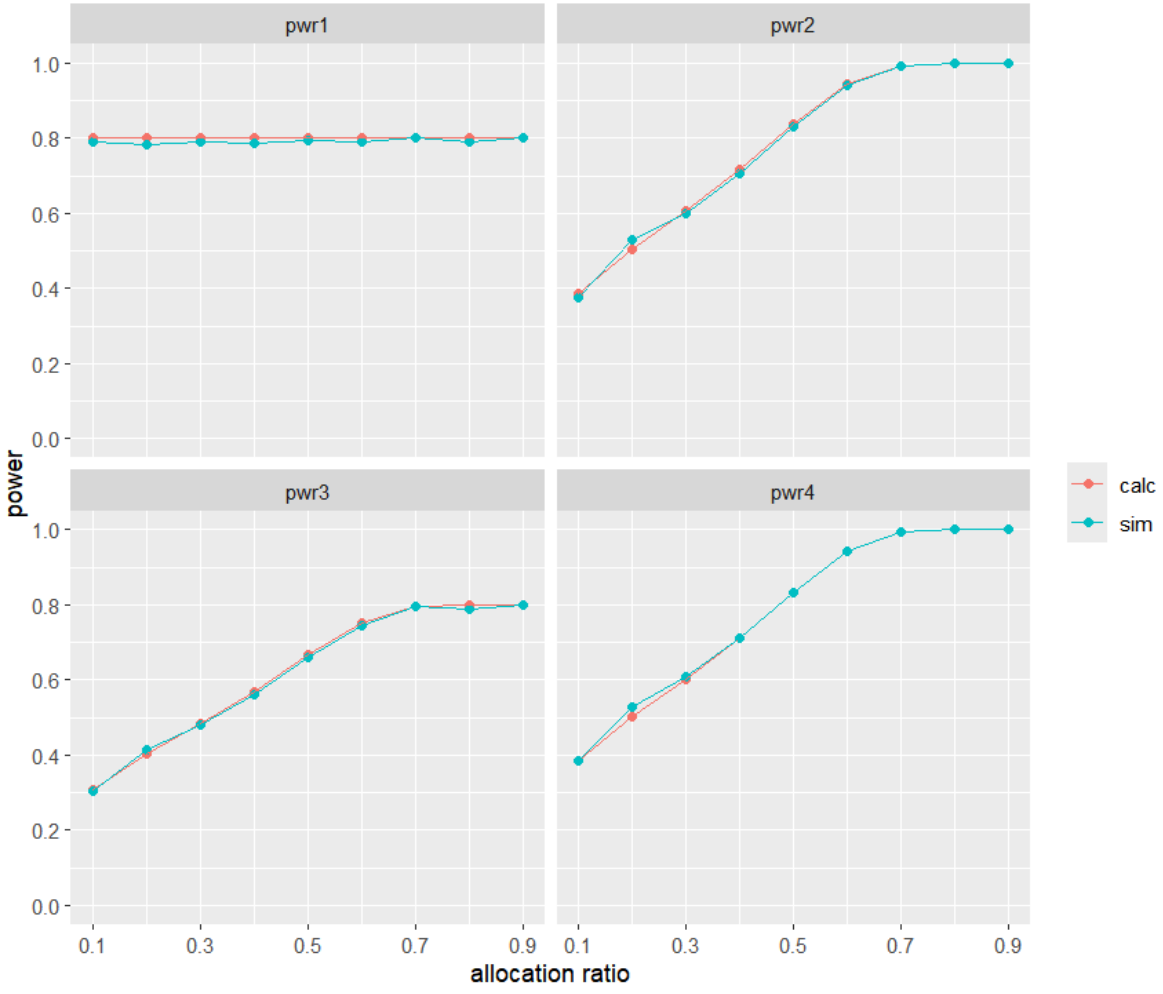
RD

```
c1 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = 0.2, N = NA, r =
1, scale = "RD", sim = FALSE)
d1 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = 0.2, N = NA, r =
1, scale = "RD", sim = TRUE)
pplot(c1, d1)
```



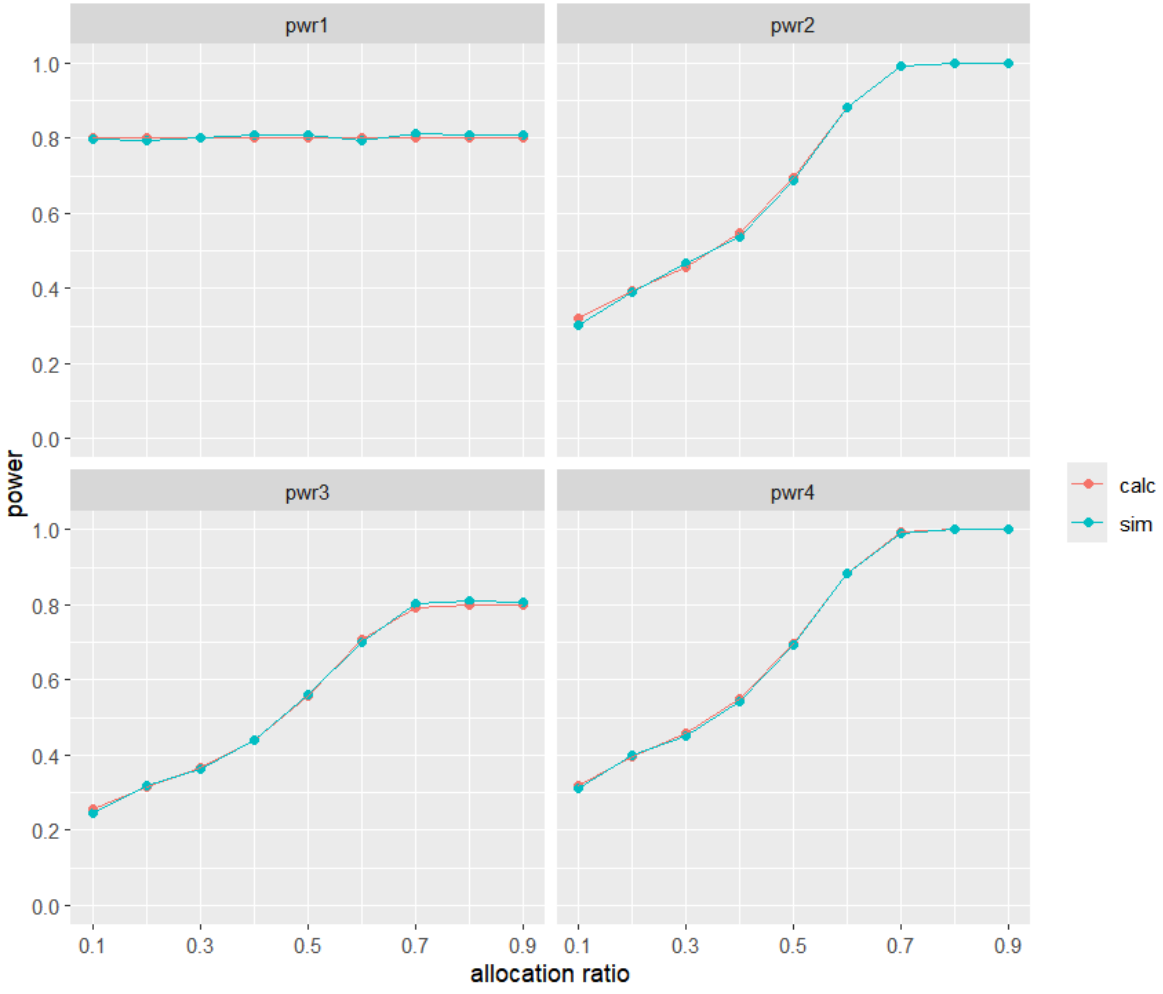
RR

```
c2 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "RR", sim = FALSE)
d2 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "RR", sim = TRUE)
pplot(c2, d2)
```



OR

```
c3 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "OR", sim = FALSE)
d3 <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_nj = 0.5, p0_nj = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA,
r = 1, scale = "OR", sim = TRUE)
pplot(c3, d3)
```

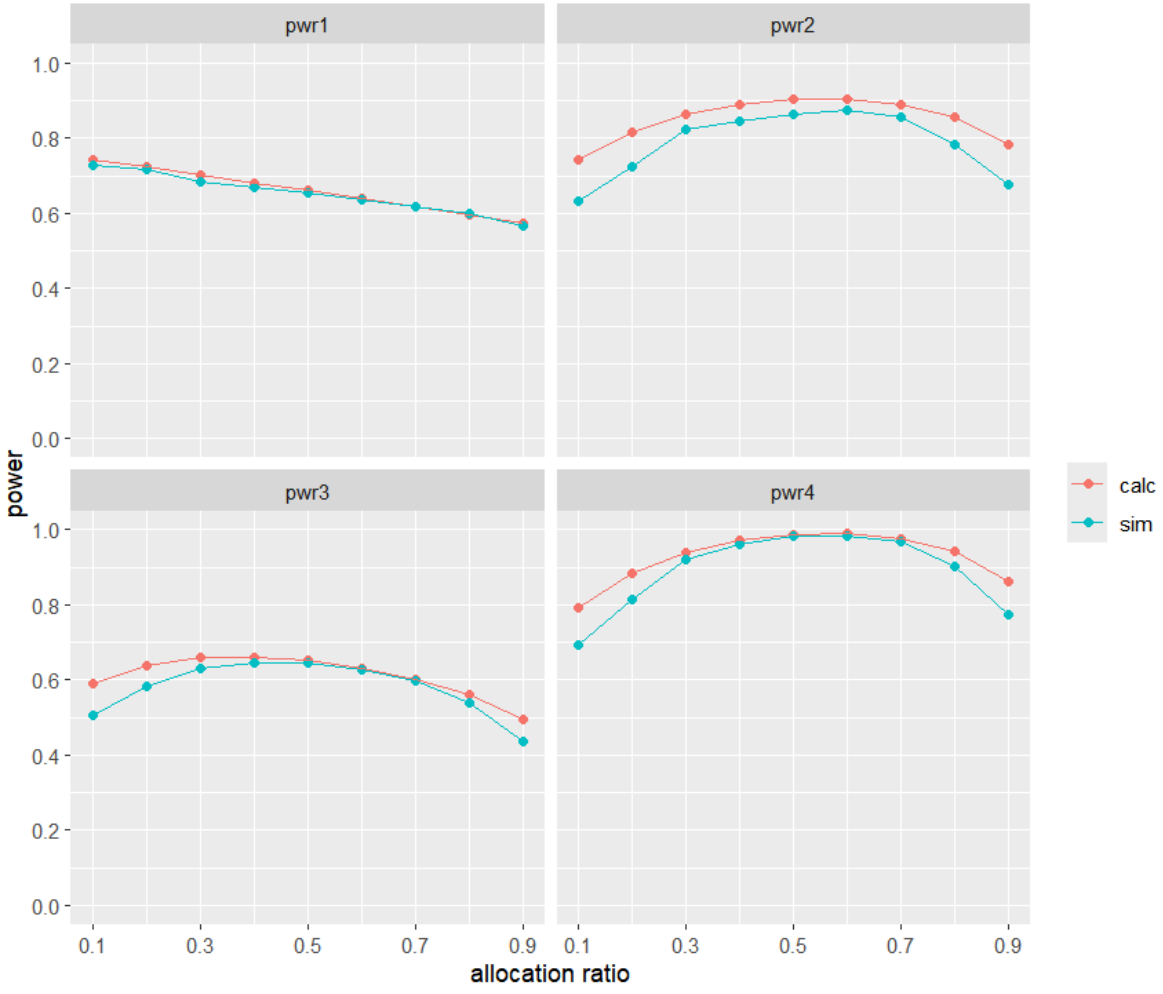


getPwr_Bin_Super_JM2

高优

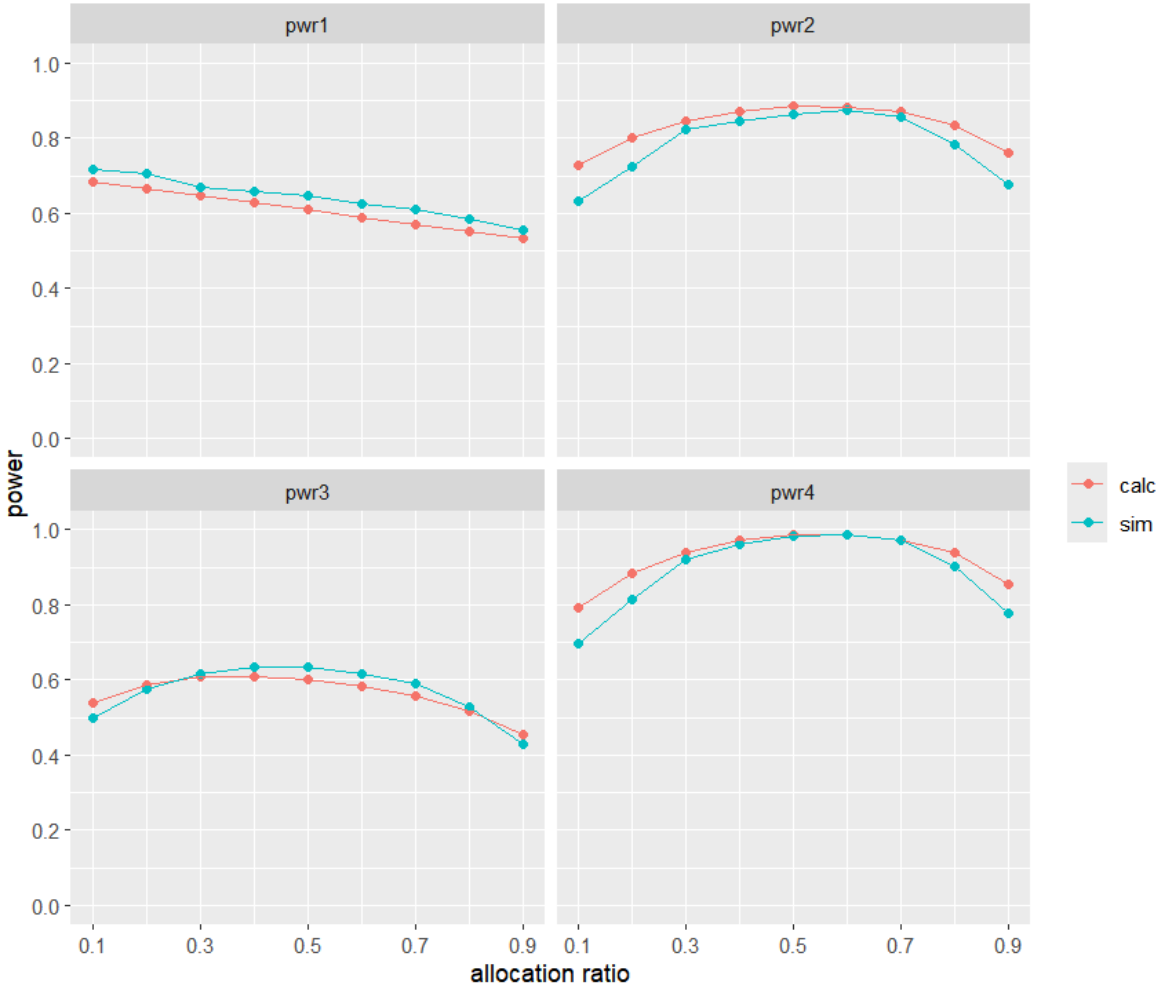
RD

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD", sim =
FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



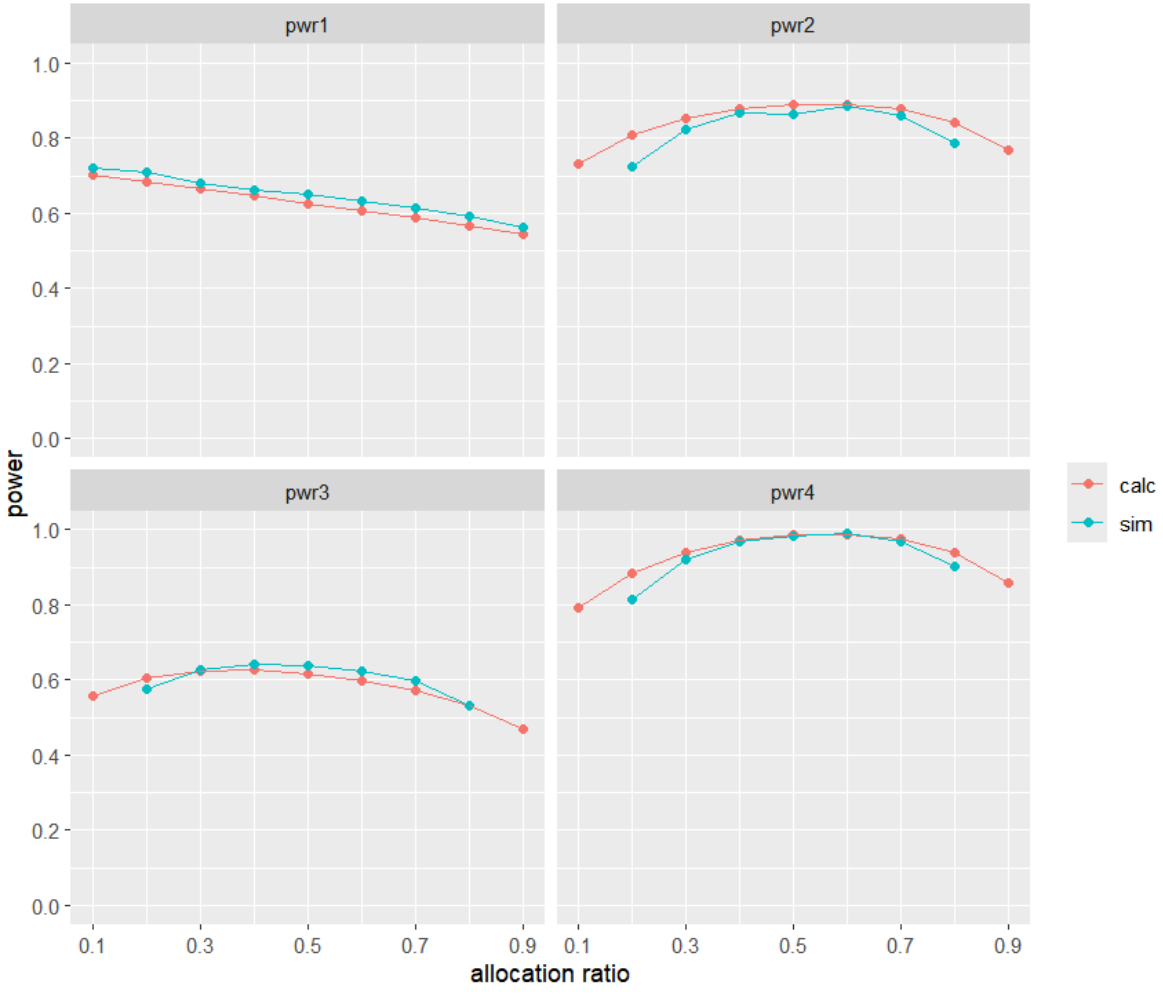
RR

```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR", sim =
FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



OR

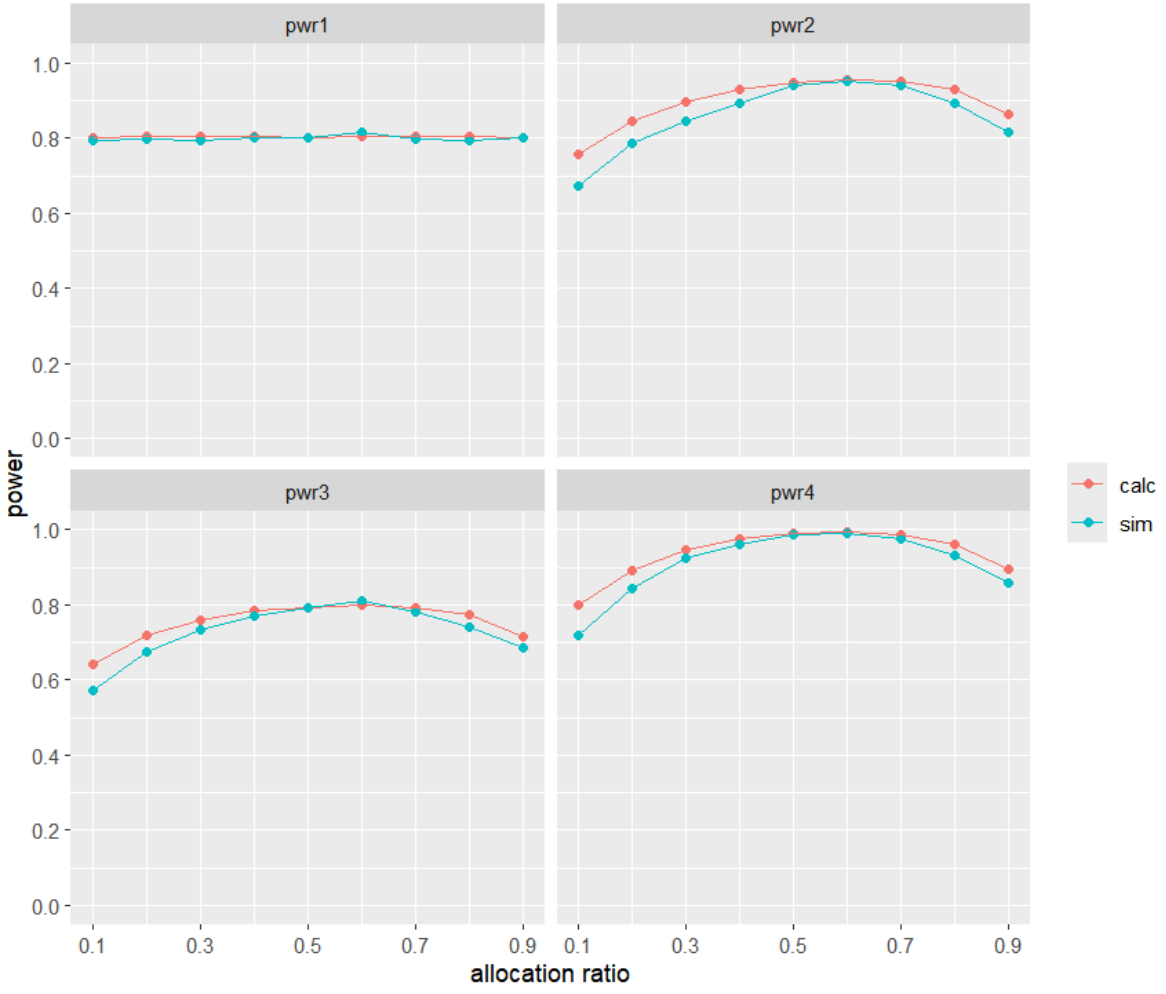
```
f_set <- seq(0.1, 0.9, 0.1)
a3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "OR", sim =
FALSE)$overall
  res$f <- f
  res
})
b3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.7, 0.75), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "OR", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a3, b3)
```



低优

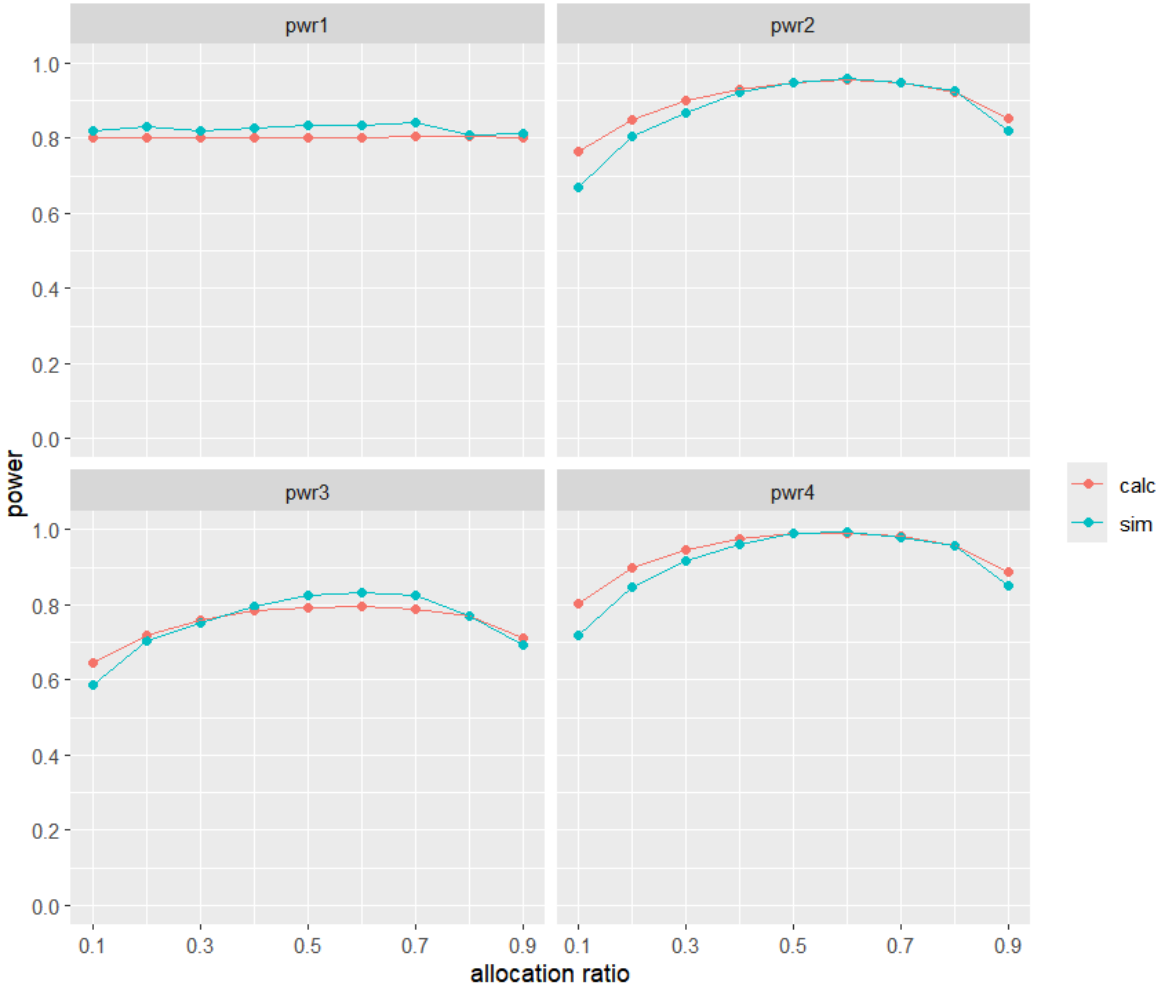
RD

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", sim =
FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



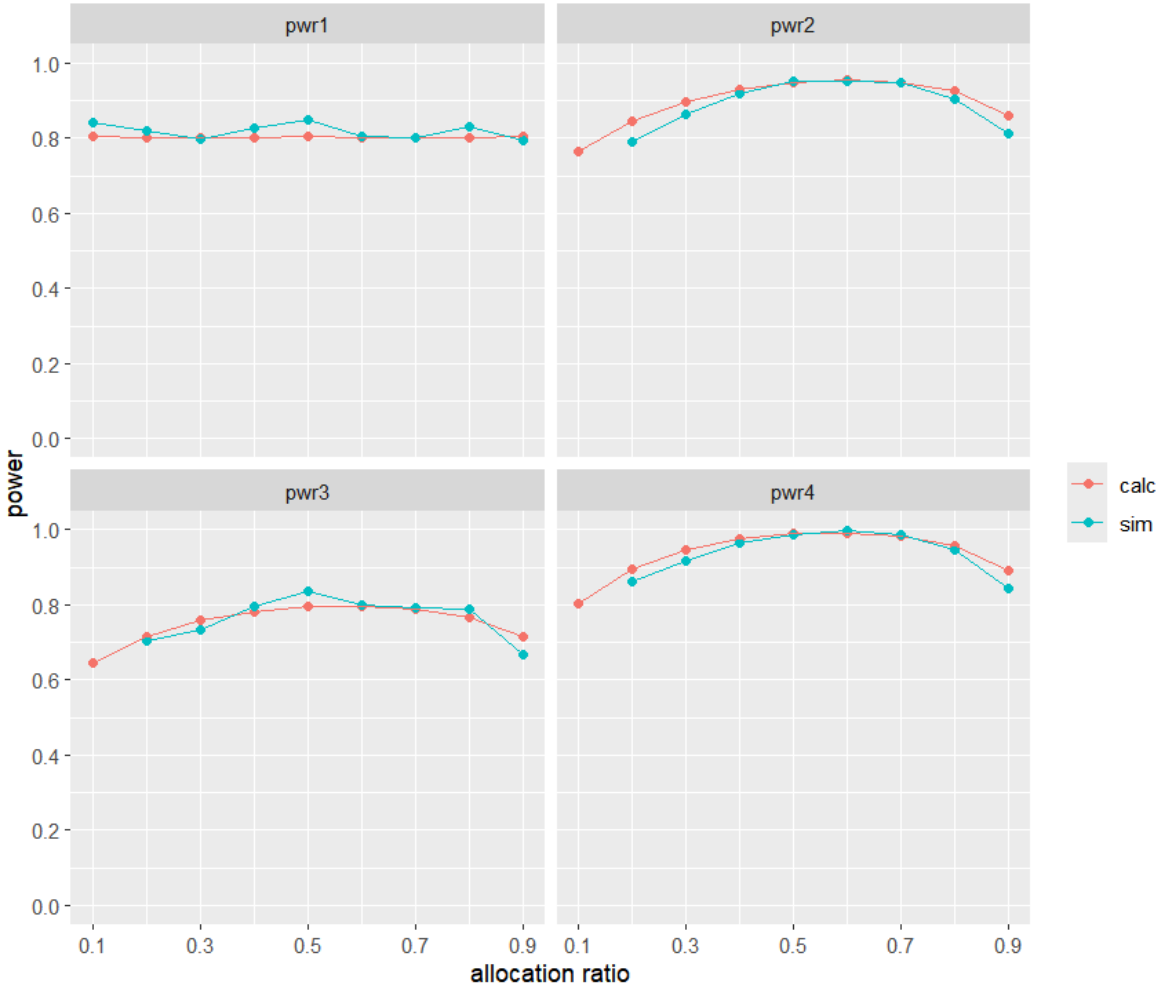
RR

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", sim =
FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```



OR

```
f_set <- seq(0.1, 0.9, 0.1)
c3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", sim =
FALSE)$overall
  res$f <- f
  res
})
d3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Super_JM2(p1_i = c(0.3, 0.25), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c3, d3)
```

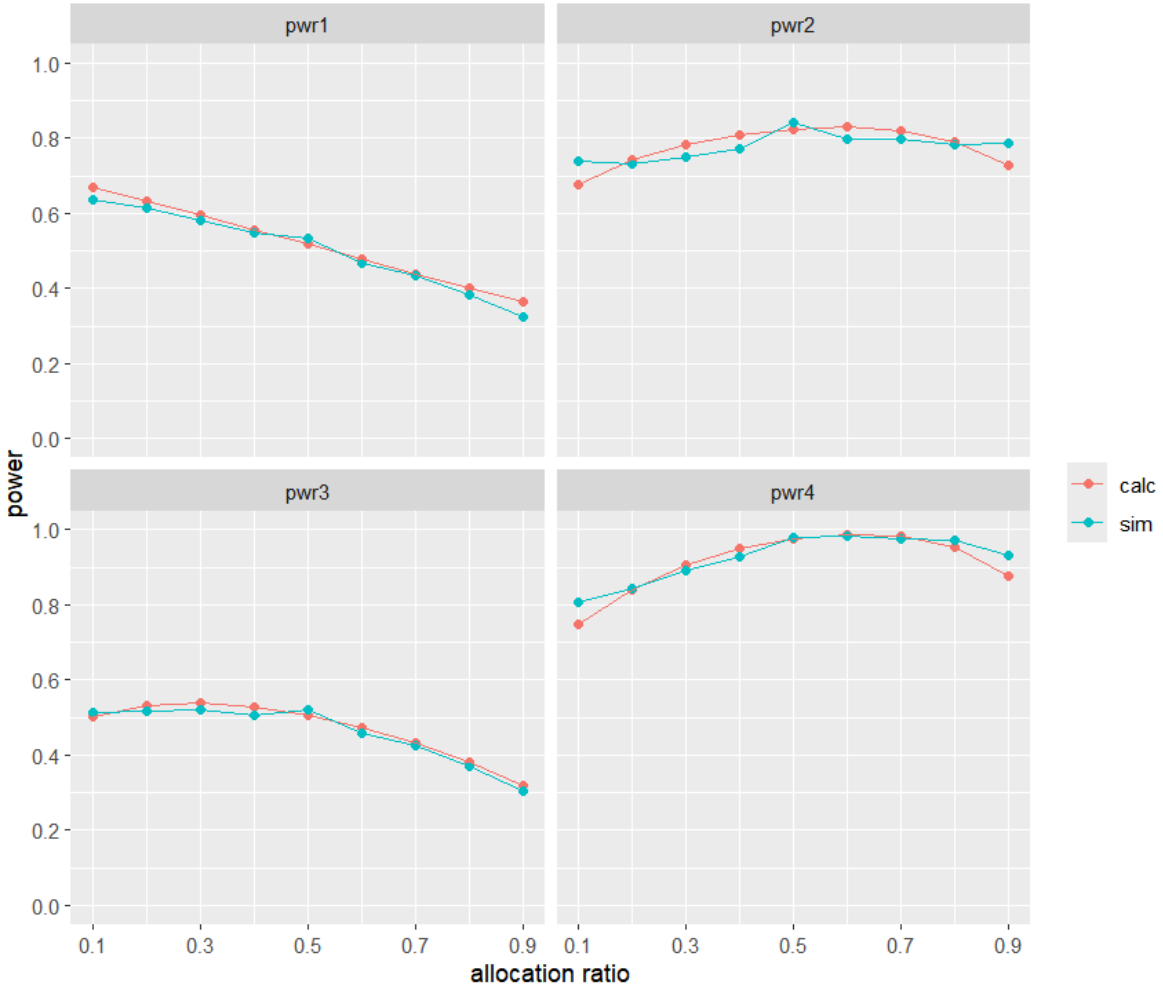


getPwr_Bin_Noninf_JM2

高优

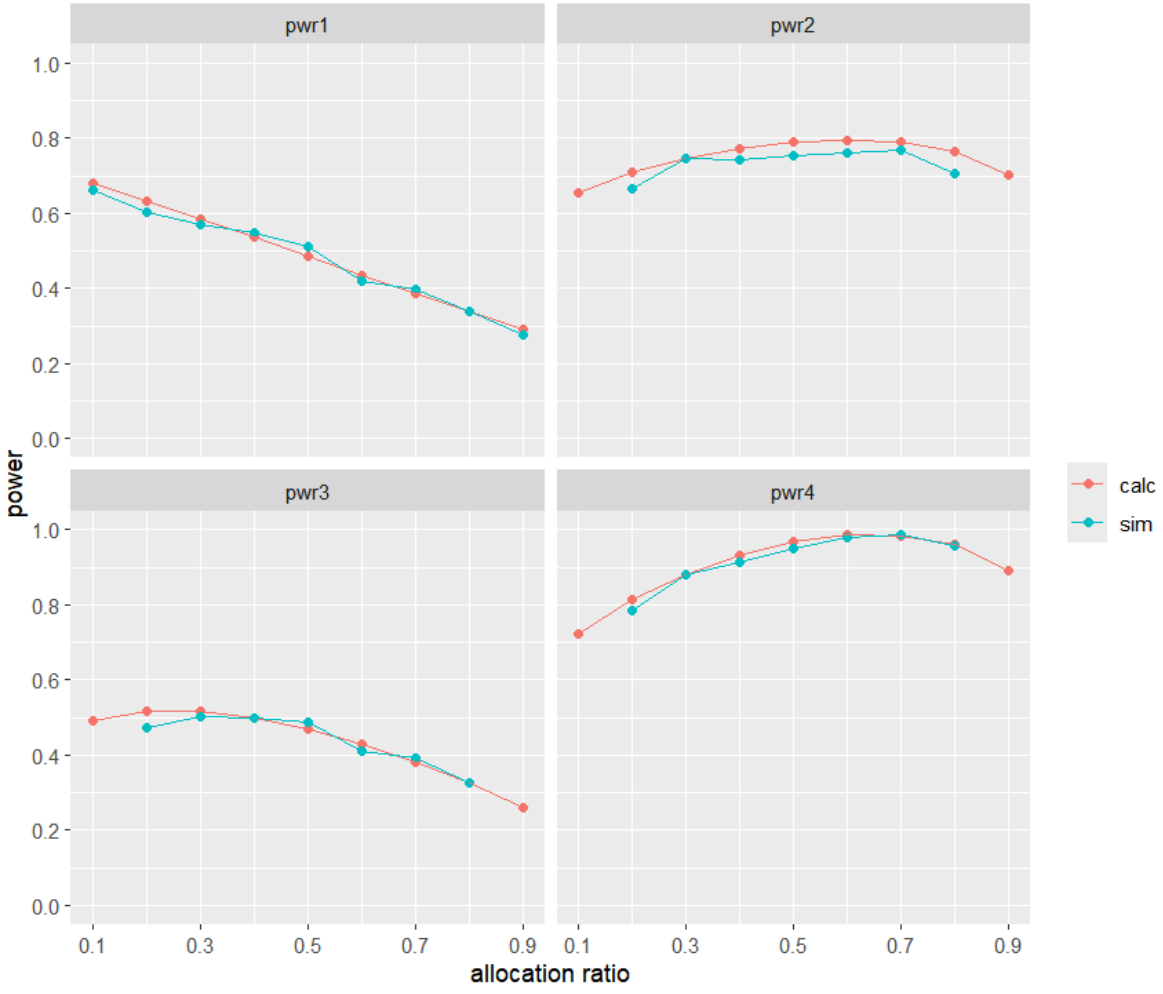
RD

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = 0.25, alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD",
direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = 0.25, alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD",
direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



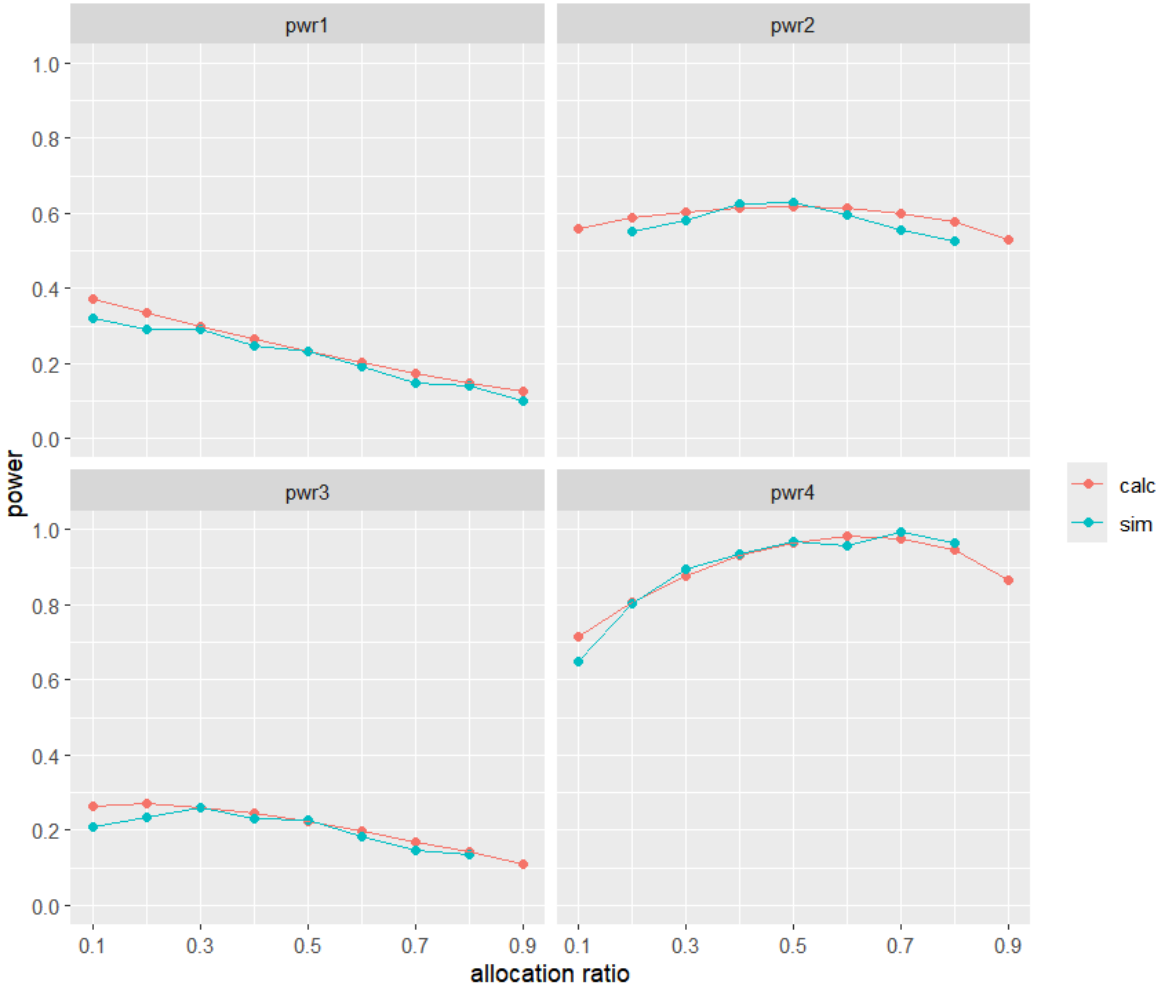
RR

```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = -log(0.6), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR",
direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = -log(0.6), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR",
direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



OR

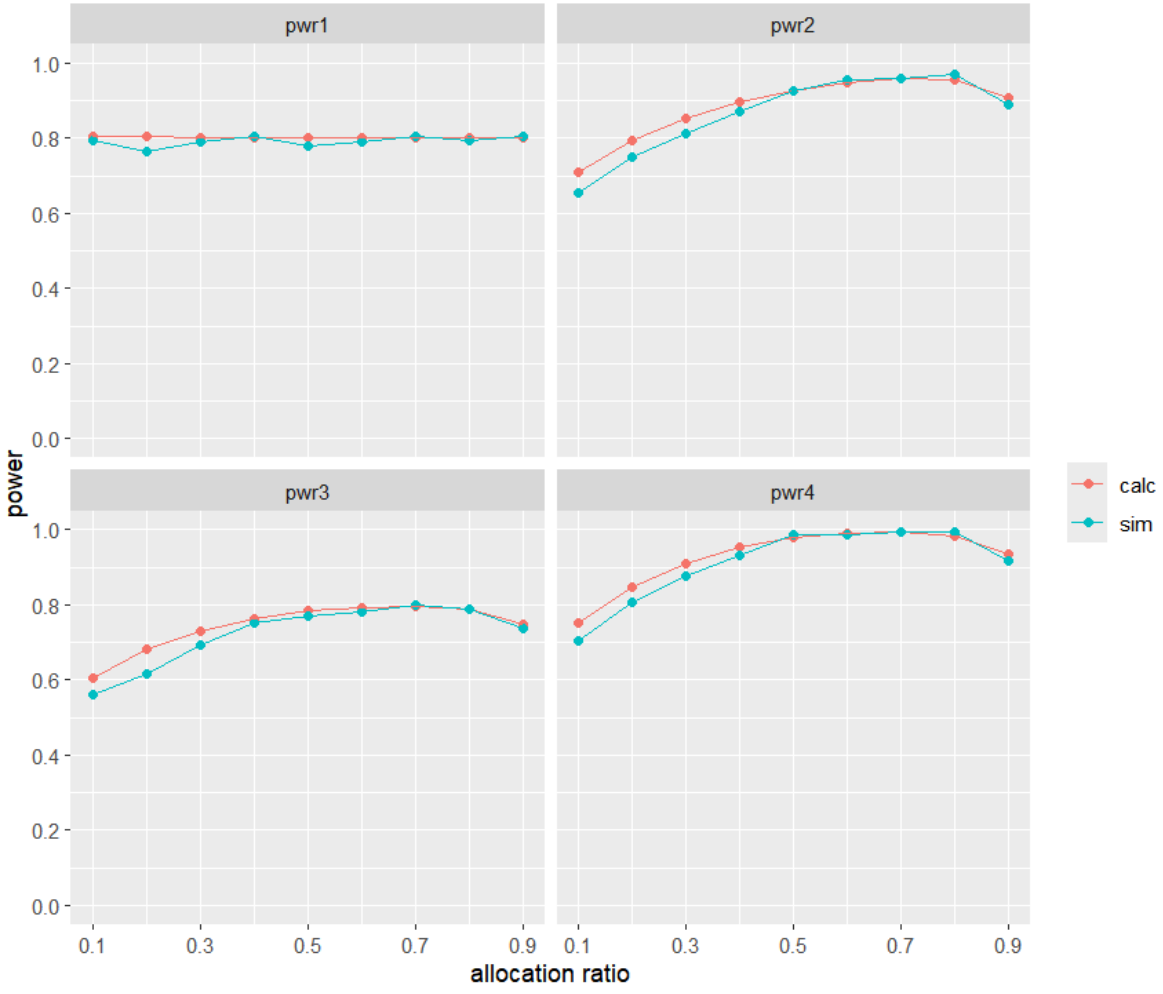
```
f_set <- seq(0.1, 0.9, 0.1)
a3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = -log(0.5), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "OR",
direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = -log(0.5), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "OR",
direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a3, b3)
```



低优

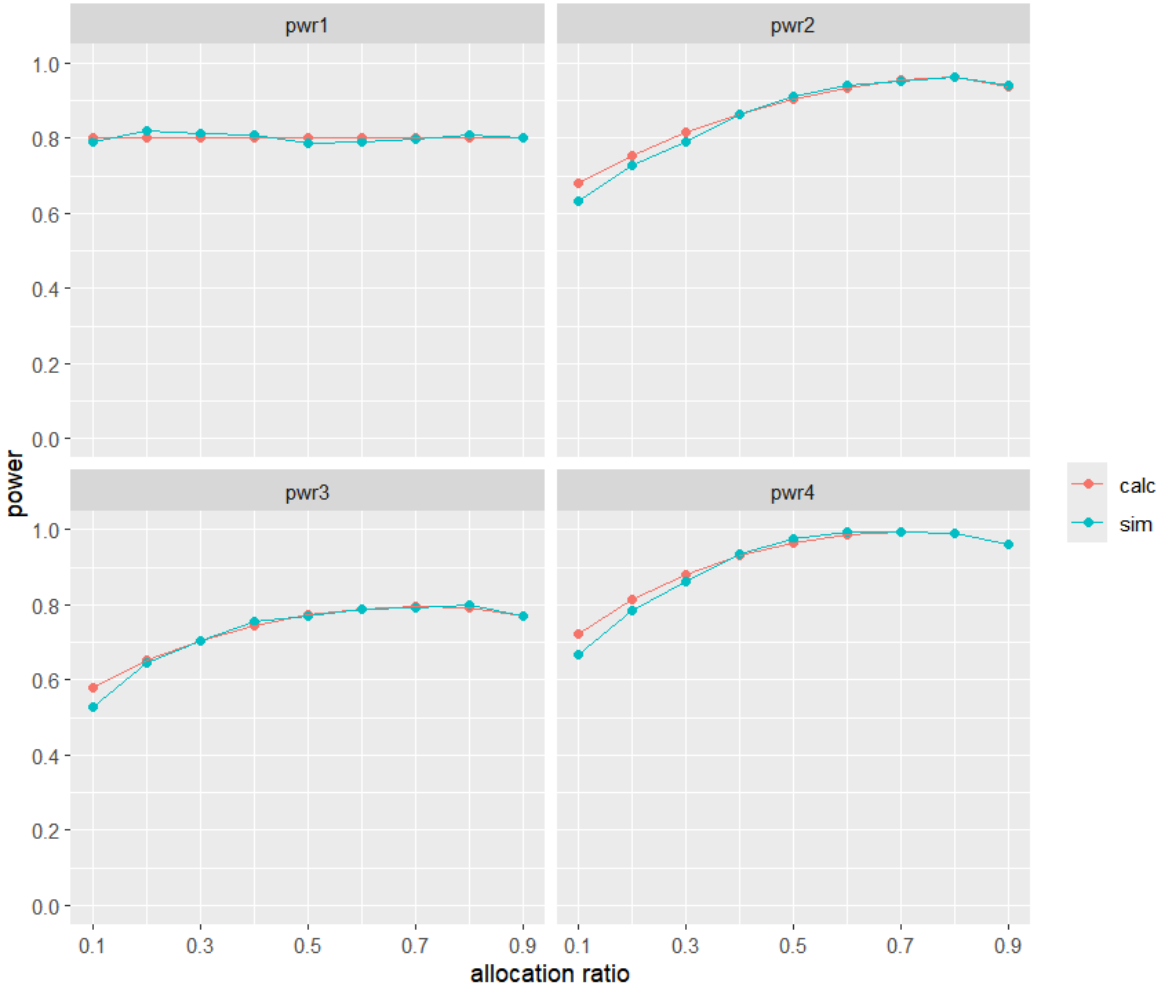
RD

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = 0.25, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD",
direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = 0.25, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD",
direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



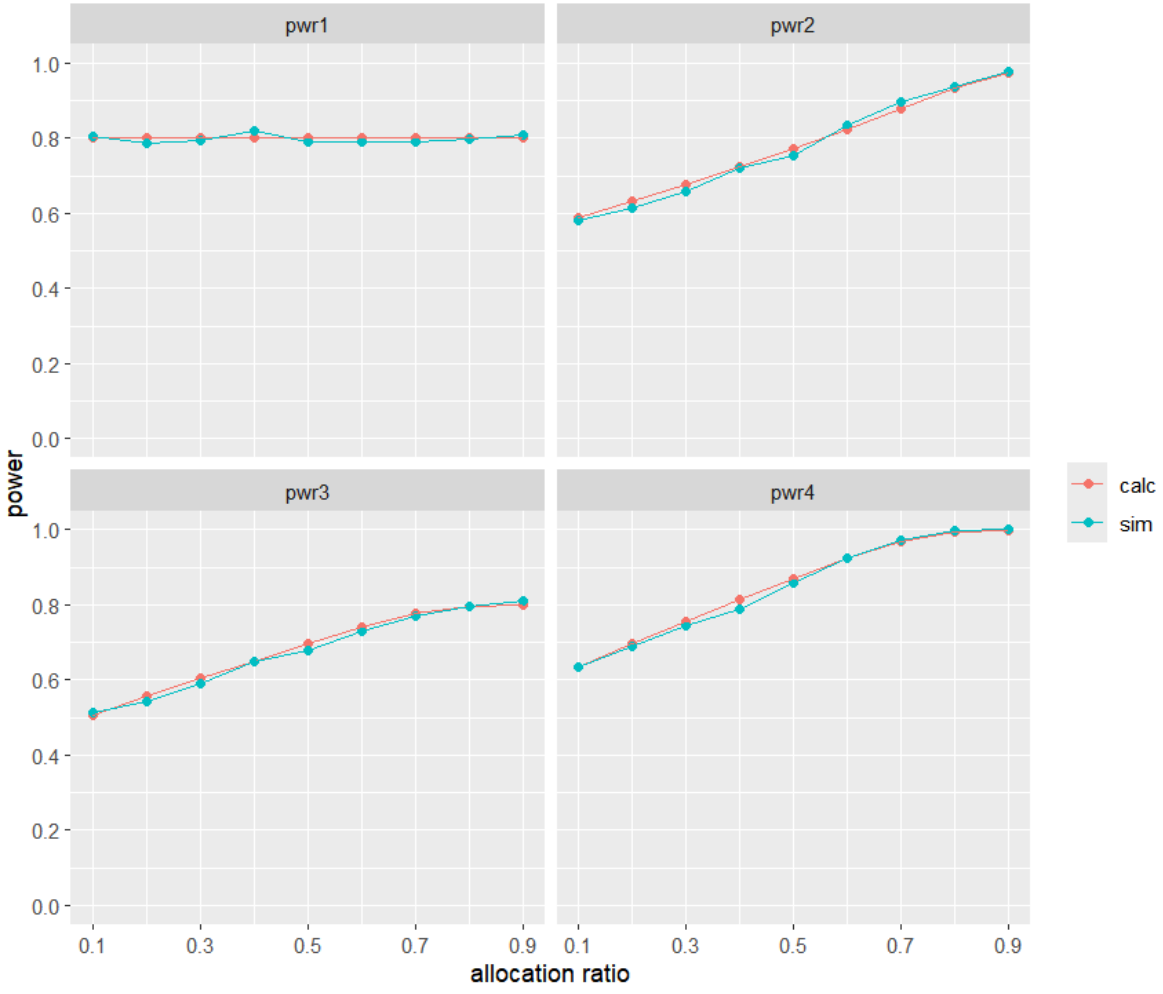
RR

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR",
direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR",
direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```



OR

```
f_set <- seq(0.1, 0.9, 0.1)
c3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR",
direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Noninf_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f,
1 - f), cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR",
direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c3, d3)
```

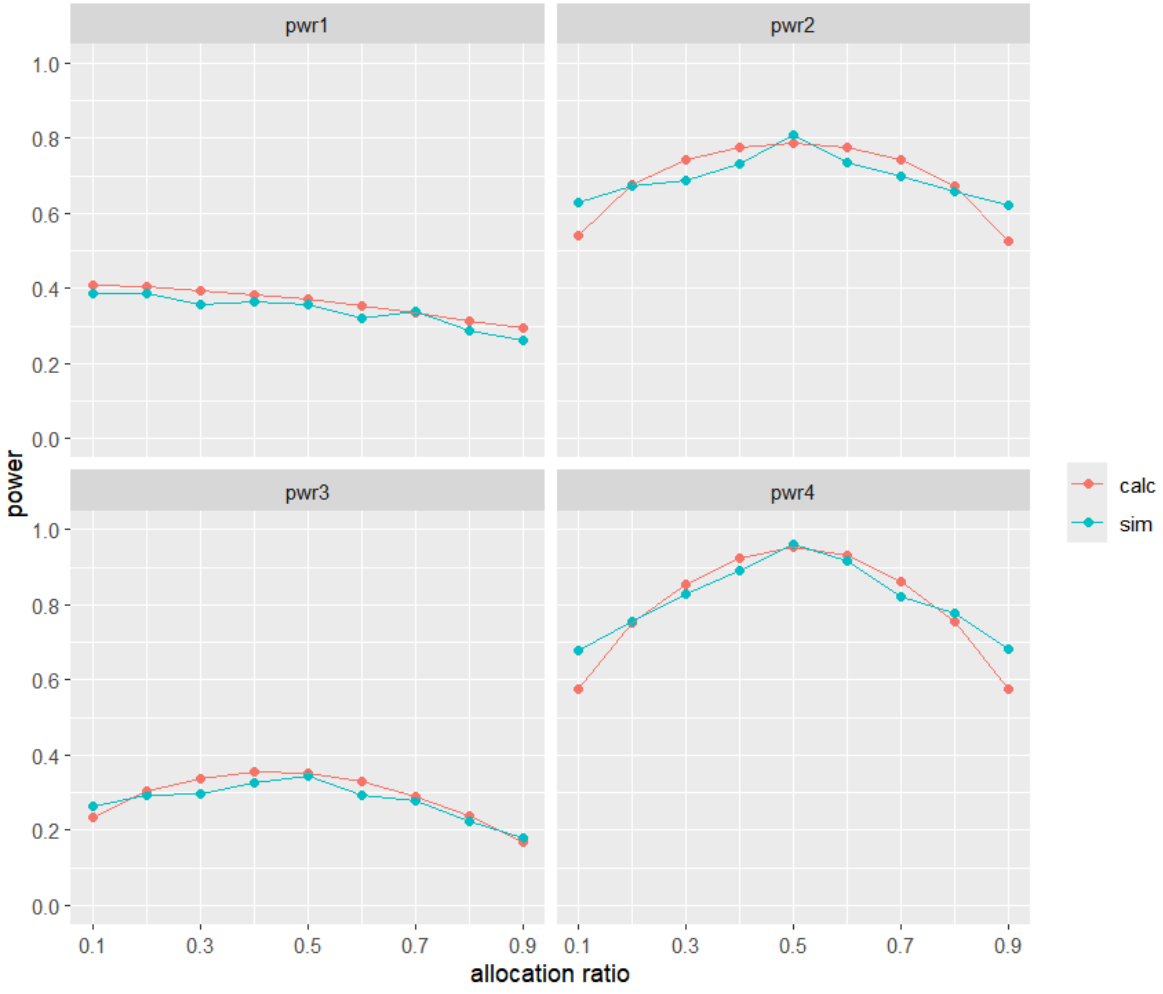


getPwr_Bin_Equi_JM2

高优

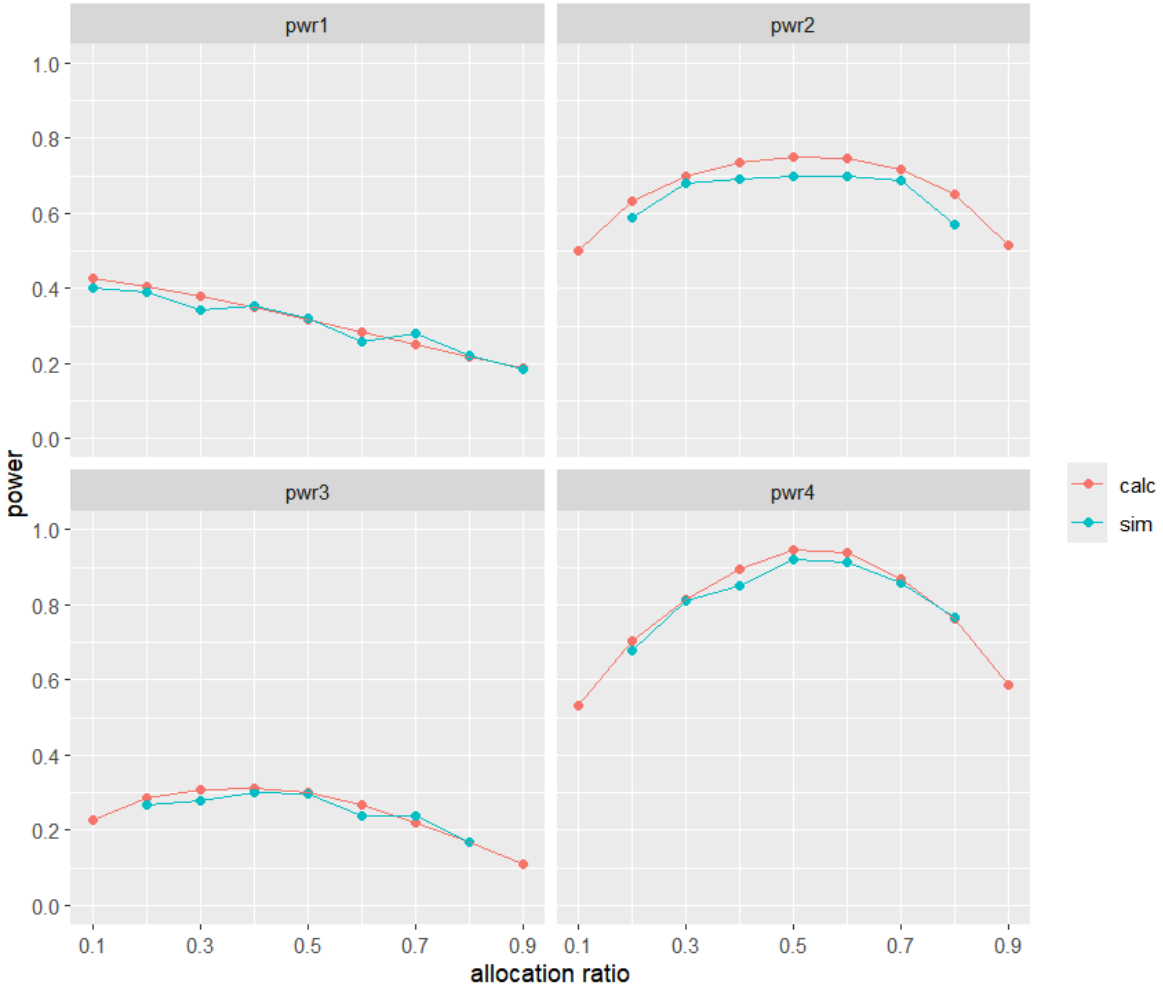
RD

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = 0.25, alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD", sim =
FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = 0.25, alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RD", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



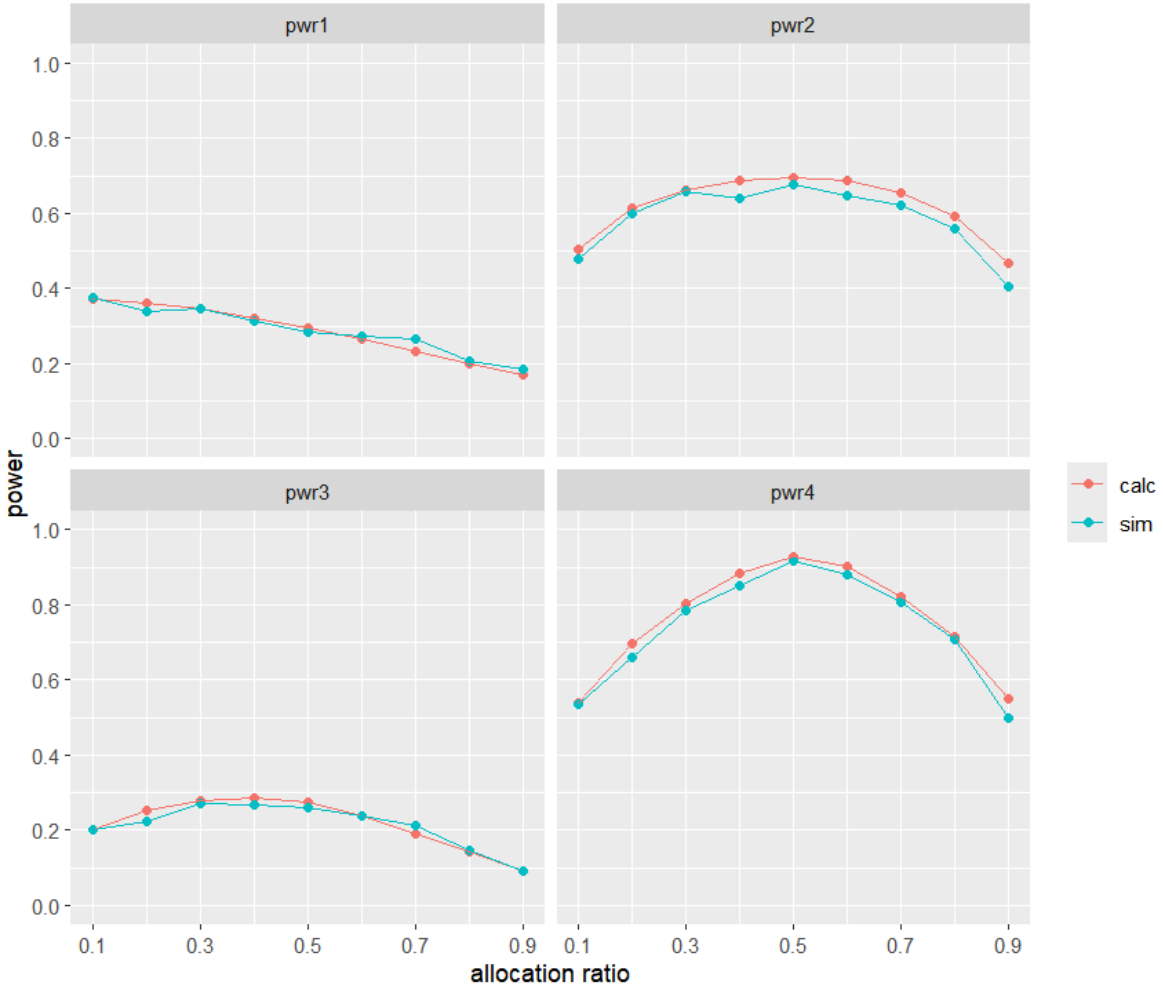
RR

```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = -log(0.6), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR",
sim = FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = -log(0.6), alpha = 0.025, beta = NA, N = 100, r = 1, scale = "RR",
sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



OR

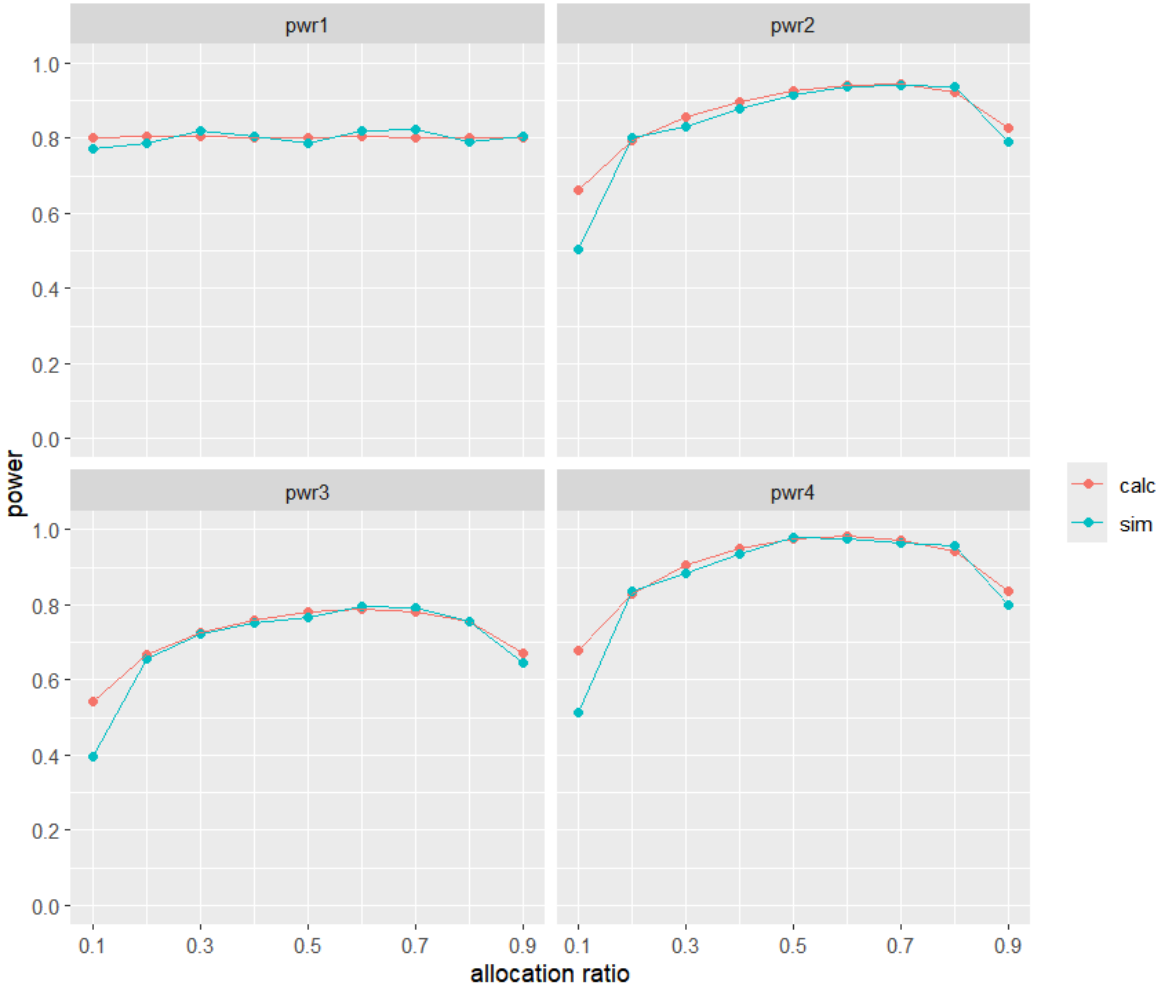
```
f_set <- seq(0.1, 0.9, 0.1)
a3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = -log(0.5), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "OR",
sim = FALSE)$overall
  res$f <- f
  res
})
b3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.4, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = -log(0.5), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "OR",
sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a3, b3)
```



低优

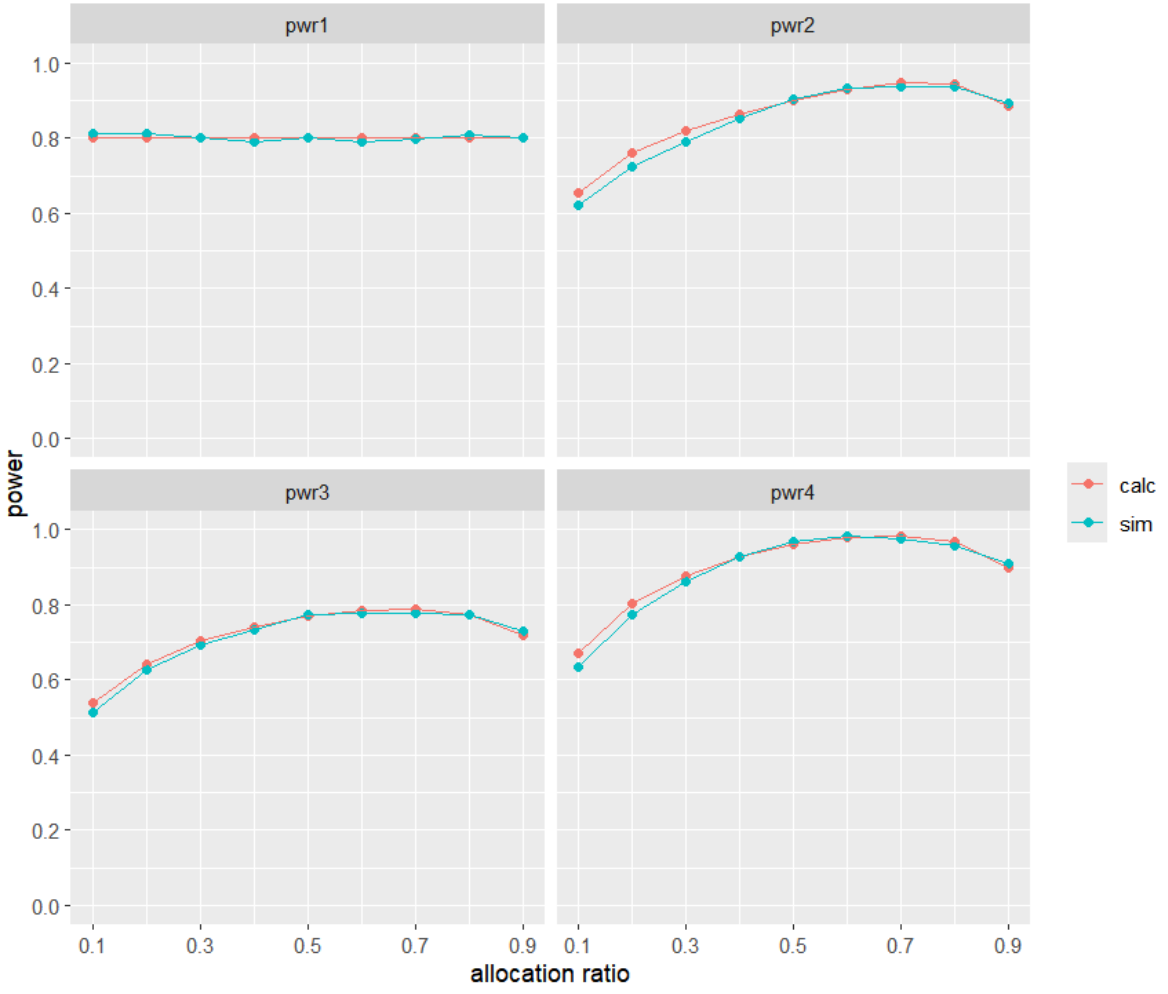
RD

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = 0.25, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", sim =
FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = 0.25, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



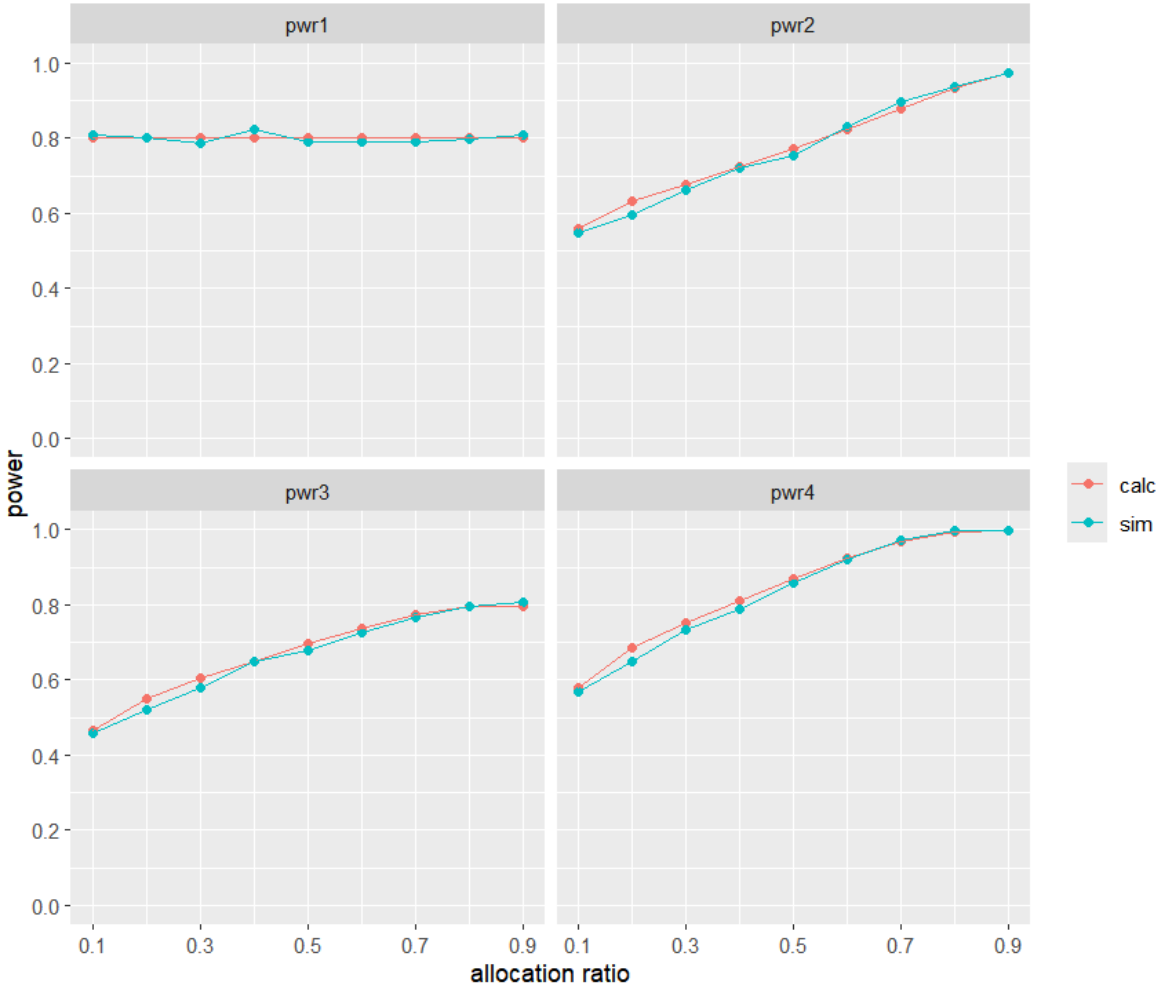
RR

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", sim
= FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = log(1.4), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", sim
= TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```



OR

```
f_set <- seq(0.1, 0.9, 0.1)
c3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", sim
= FALSE)$overall
  res$f <- f
  res
})
d3 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Bin_Equi_JM2(p1_i = c(0.6, 0.5), p0_i = c(0.5, 0.5), fi = c(f, 1
- f), cut = log(1.7), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", sim
= TRUE)$overall
  res$f <- f
  res
})
pplot(c3, d3)
```

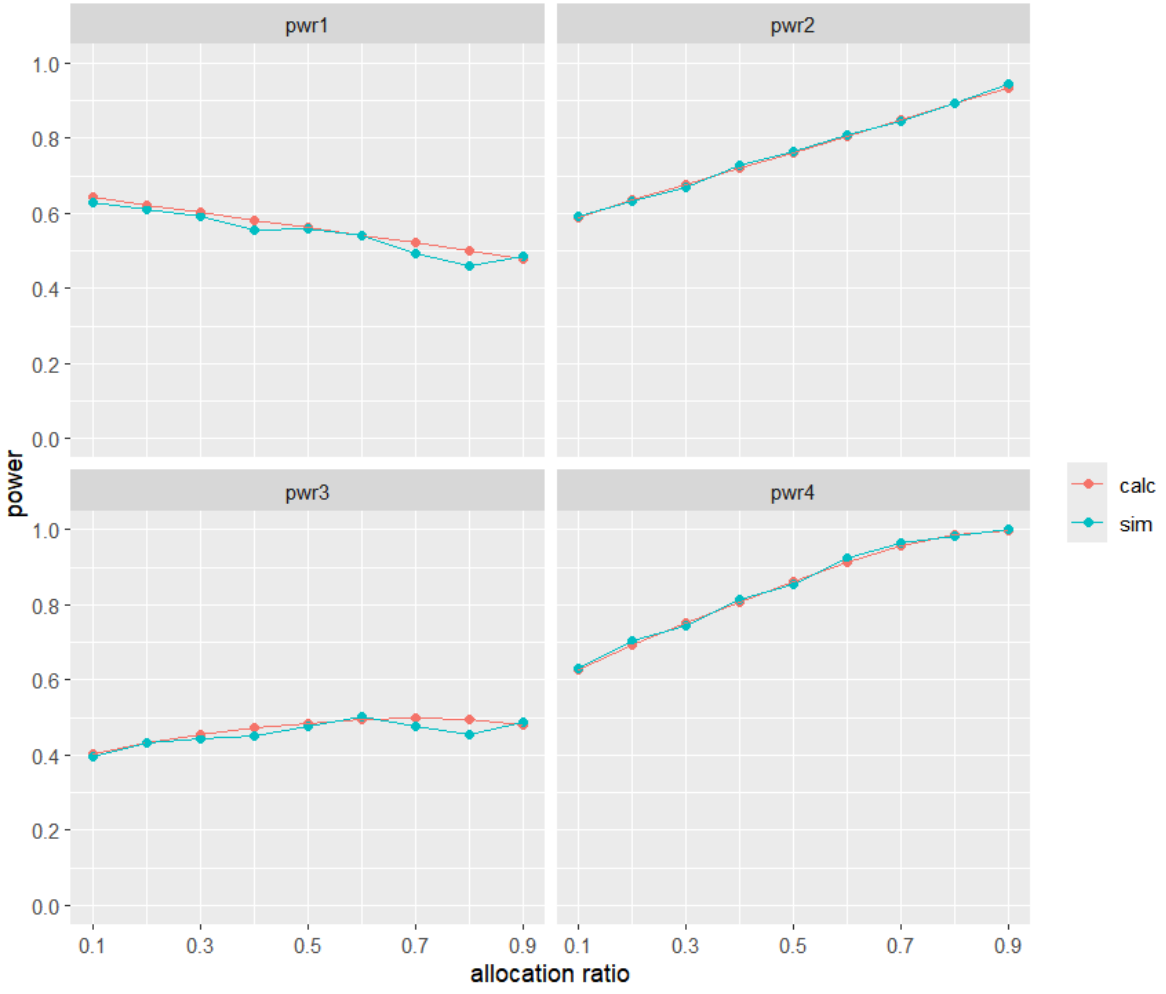


getPwr_Surv_Super_JM1

高优

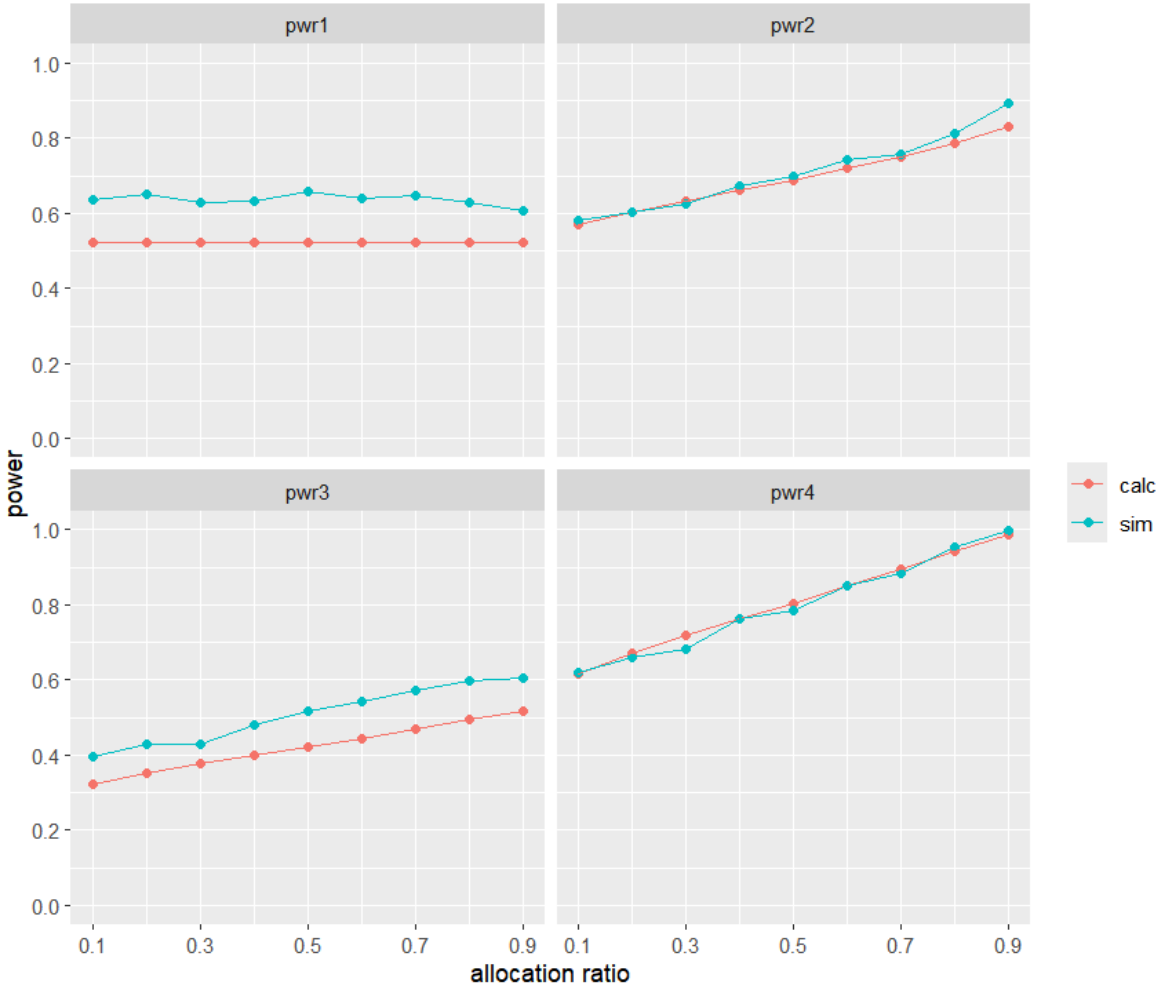
criterion=1

```
a1 <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_nj = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 1, sim
= FALSE)
b1 <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_nj = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 1, sim
= TRUE)
pplot(a1, b1)
```



criterion=2

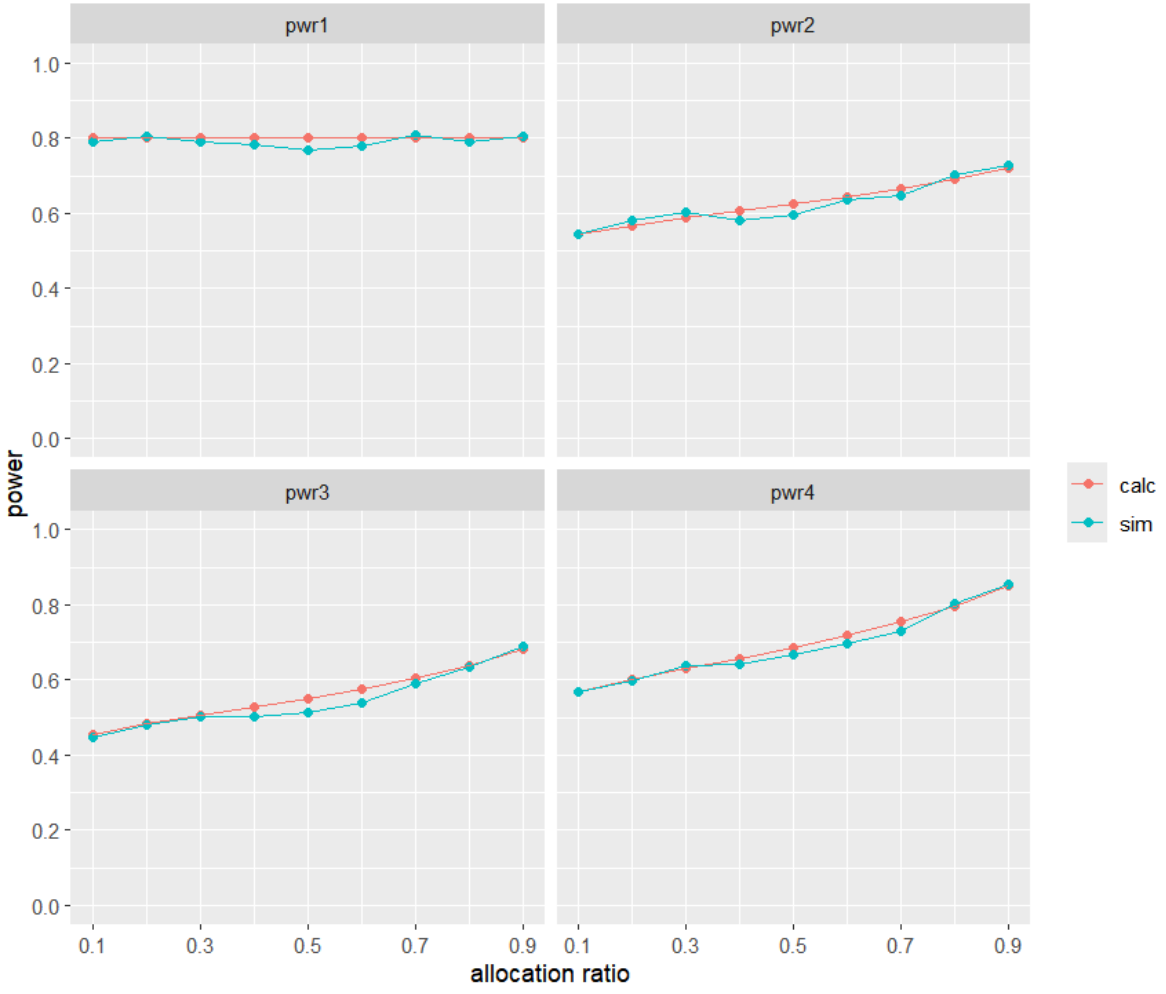
```
a2 <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_a = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 2, sim
= FALSE)
b2 <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_a = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 2, sim
= TRUE)
pplot(a2, b2)
```



低优

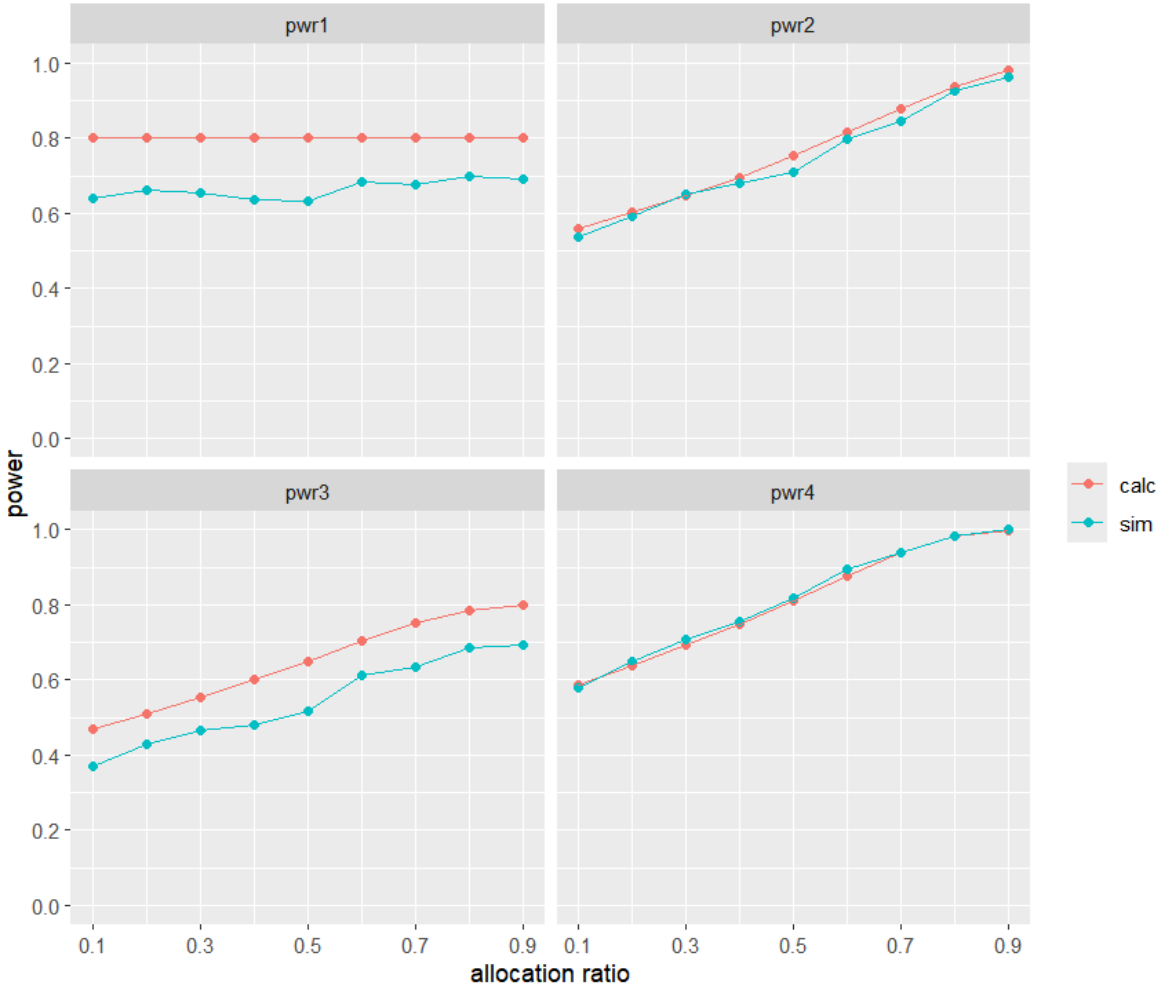
criterion=1

```
c1 <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim
= FALSE)
d1 <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim
= TRUE)
pplot(c1, d1)
```



criterion=2

```
c2 <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_nj = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim
= FALSE)
d2 <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_nj = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim
= TRUE)
pplot(c2, d2)
```

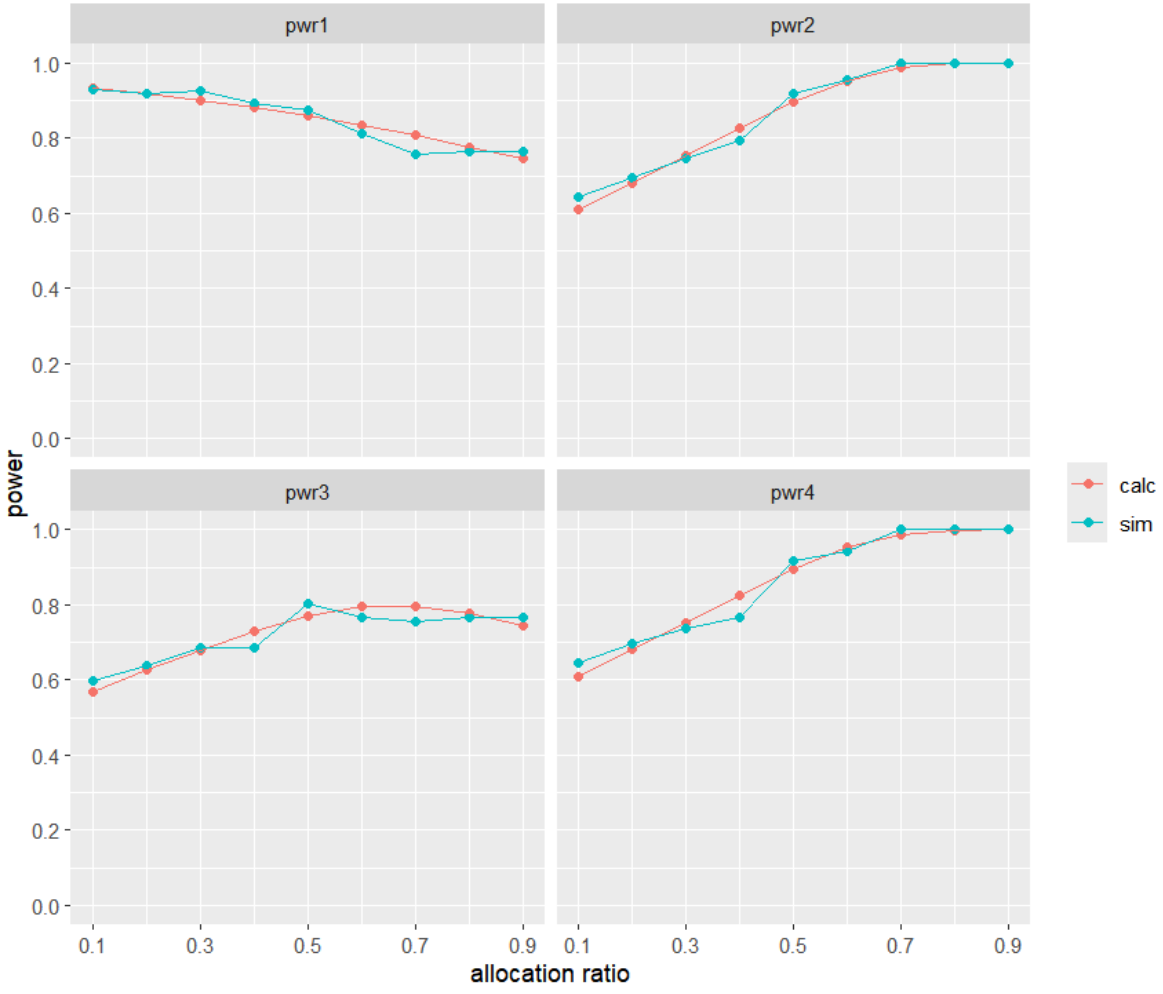


getPwr_Surv_Noninf_JM1

高优

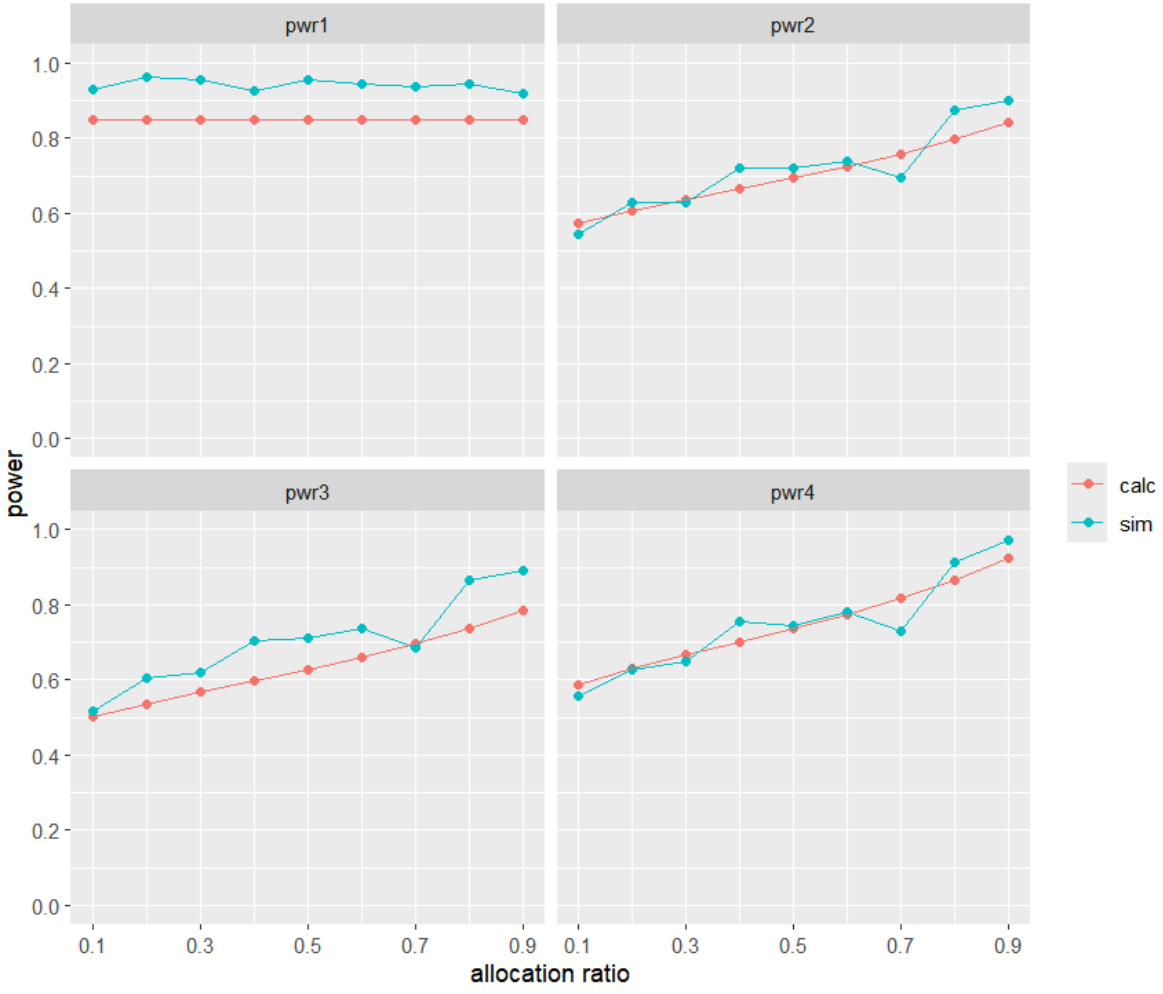
criterion=1

```
a1 <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_nj = log(1.0), f =
seq(0.1, 0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400,
r = 1, criterion = 1, direct = 1, sim = FALSE)
b1 <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_nj = log(1.0), f =
seq(0.1, 0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400,
r = 1, criterion = 1, direct = 1, sim = TRUE)
pplot(a1, b1)
```



criterion=2

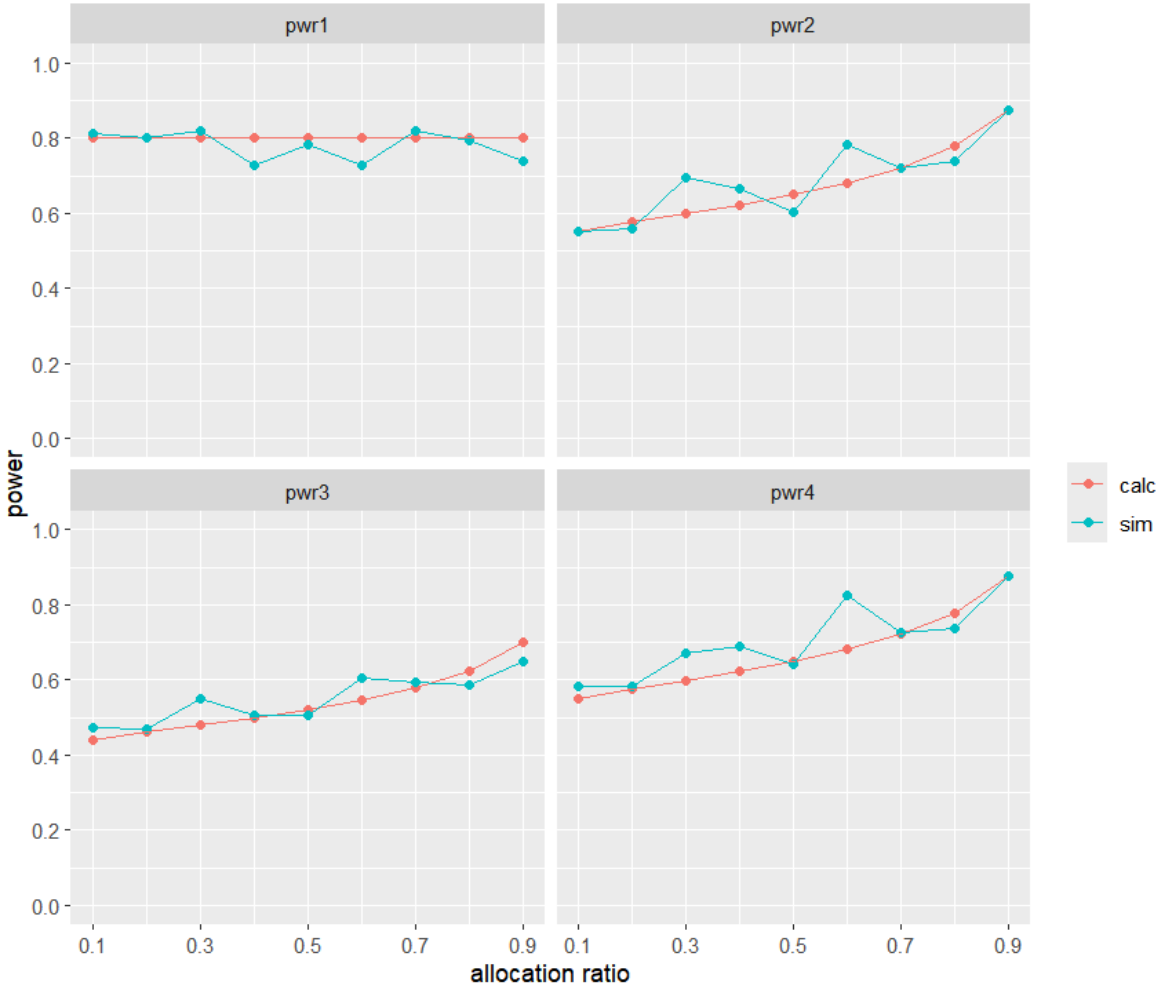
```
a2 <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 2, direct = 1, sim = FALSE)
b2 <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 2, direct = 1, sim = TRUE)
pplot(a2, b2)
pplot(a2, b2)
```



低优

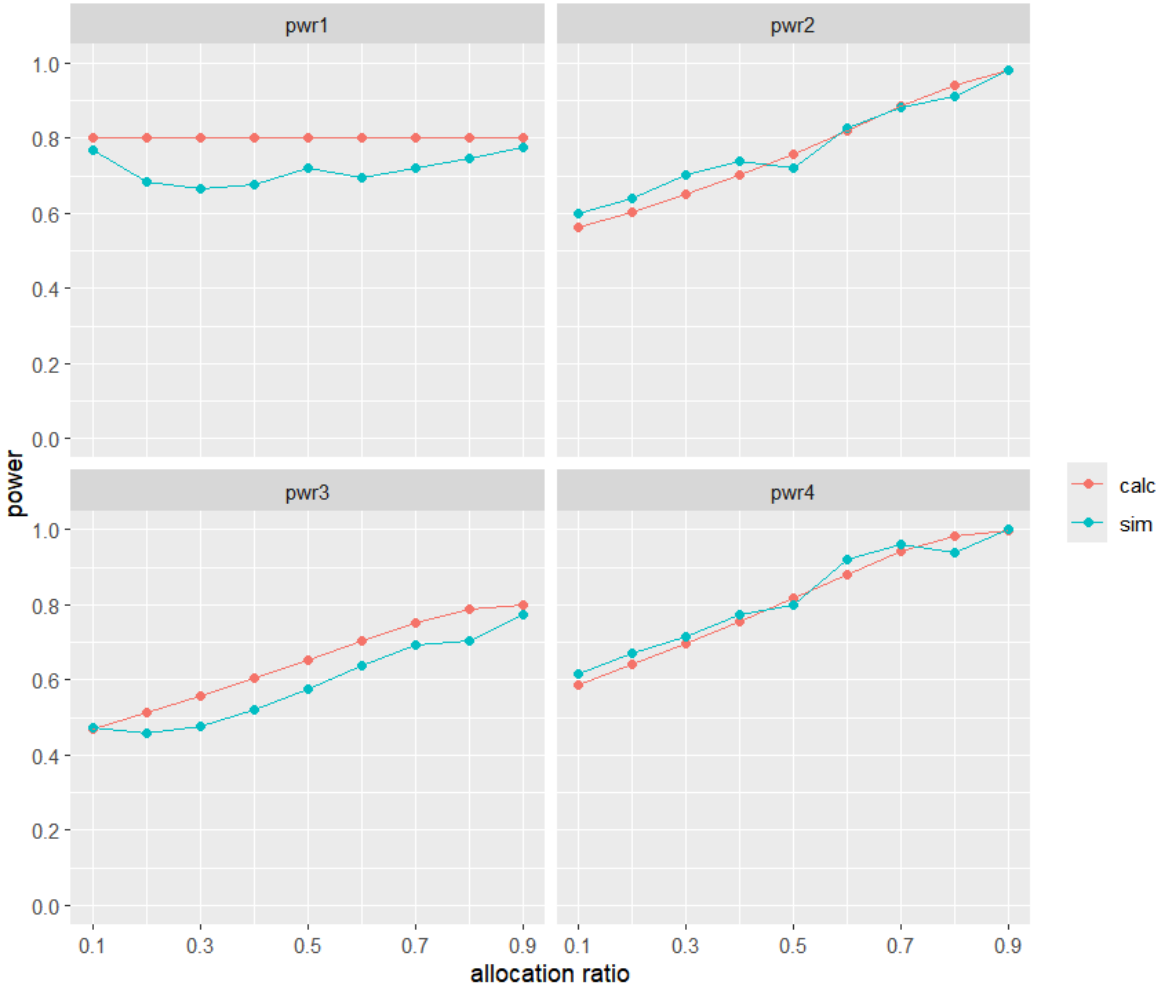
criterion=1

```
c1 <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 1, direct = -1, sim = FALSE)
d1 <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 1, direct = -1, sim = TRUE)
pplot(c1, d1)
```



criterion=2

```
c2 <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_nj = log(1.0), f =
seq(0.1, 0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA,
r = 1, criterion = 2, direct = -1, sim = FALSE)
d2 <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_nj = log(1.0), f =
seq(0.1, 0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA,
r = 1, criterion = 2, direct = -1, sim = TRUE)
pplot(c2, d2)
```

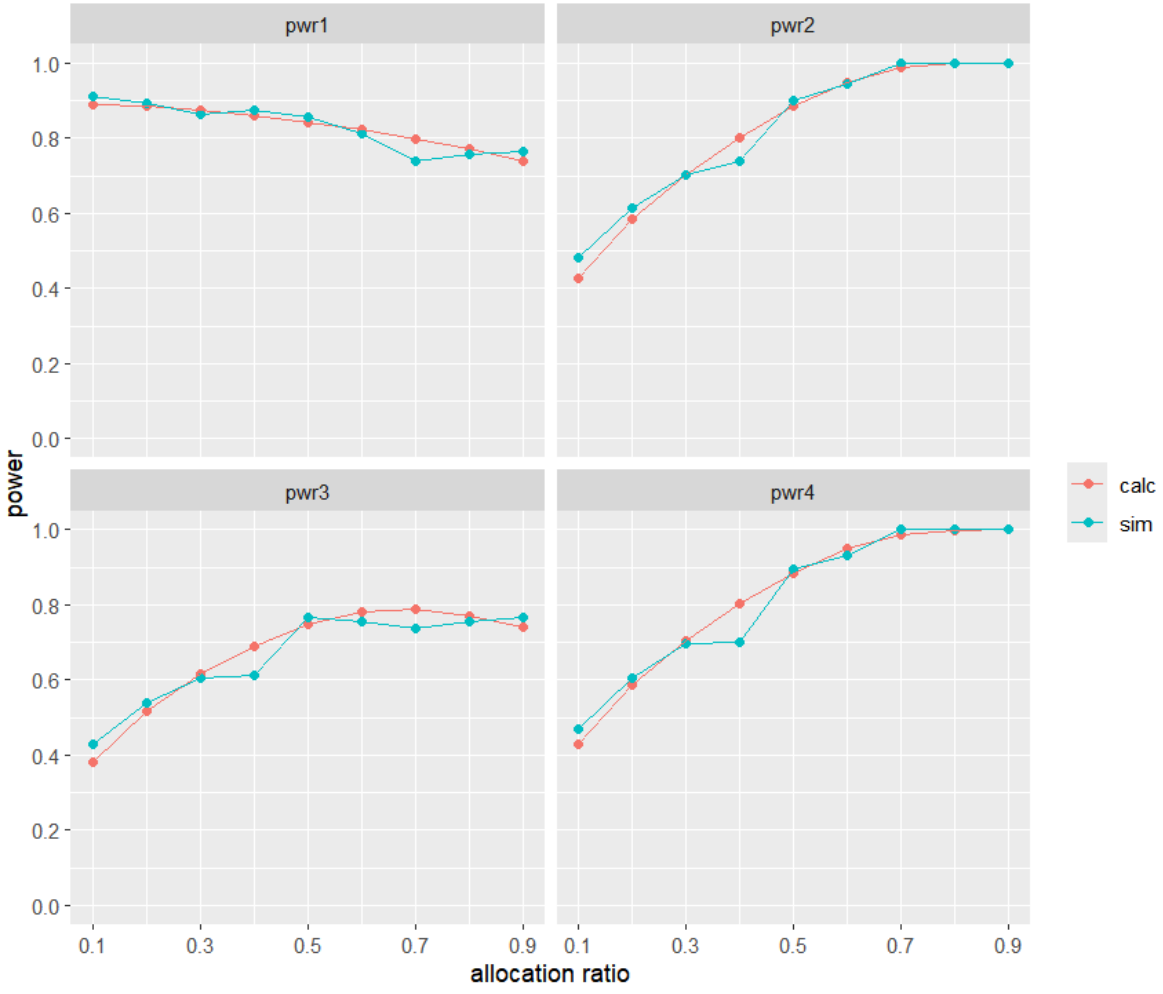


getPwr_Surv_Equi_JM1

高优

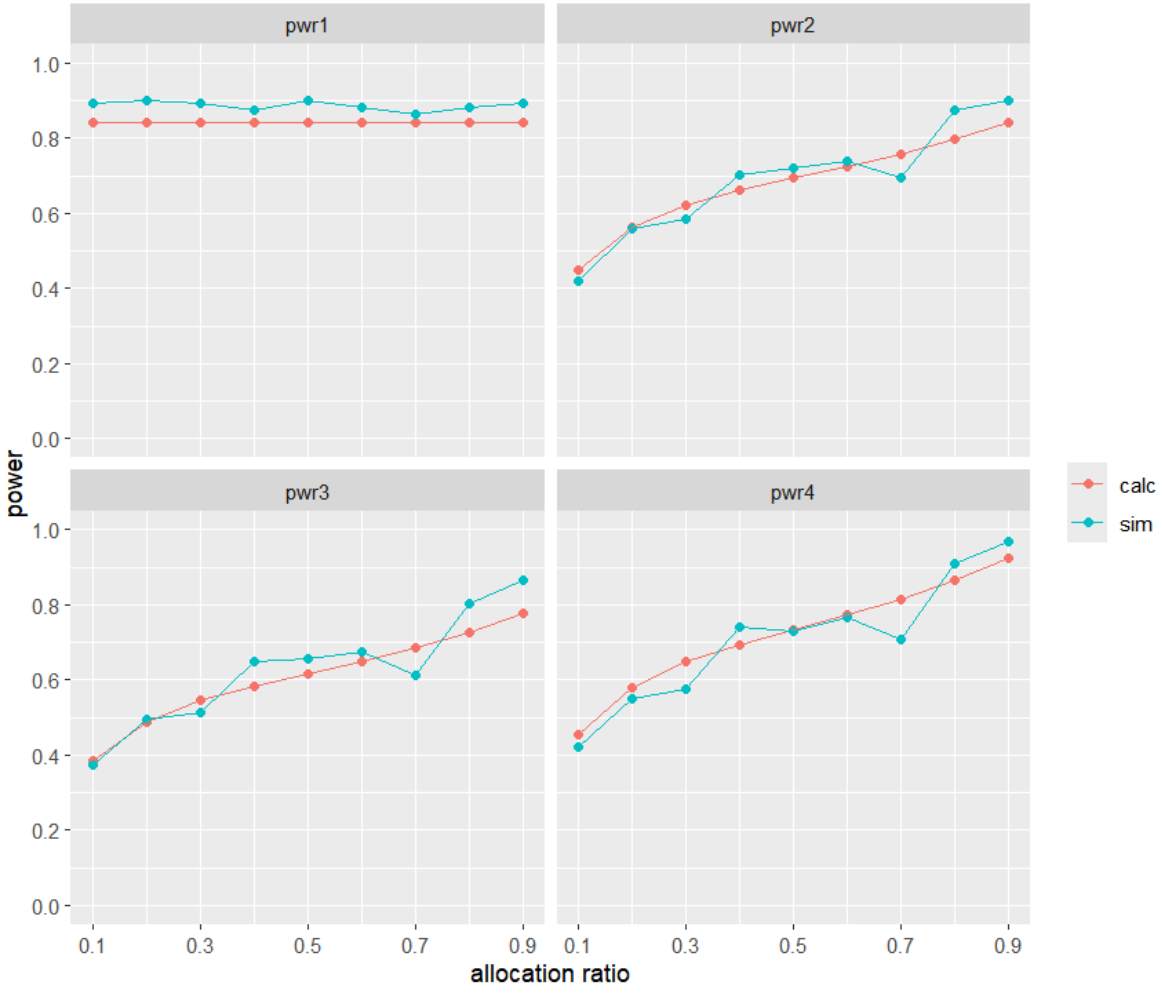
criterion=1

```
a1 <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_nj = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 1, sim = FALSE)
b1 <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_nj = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 1, sim = TRUE)
pplot(a1, b1)
```



criterion=2

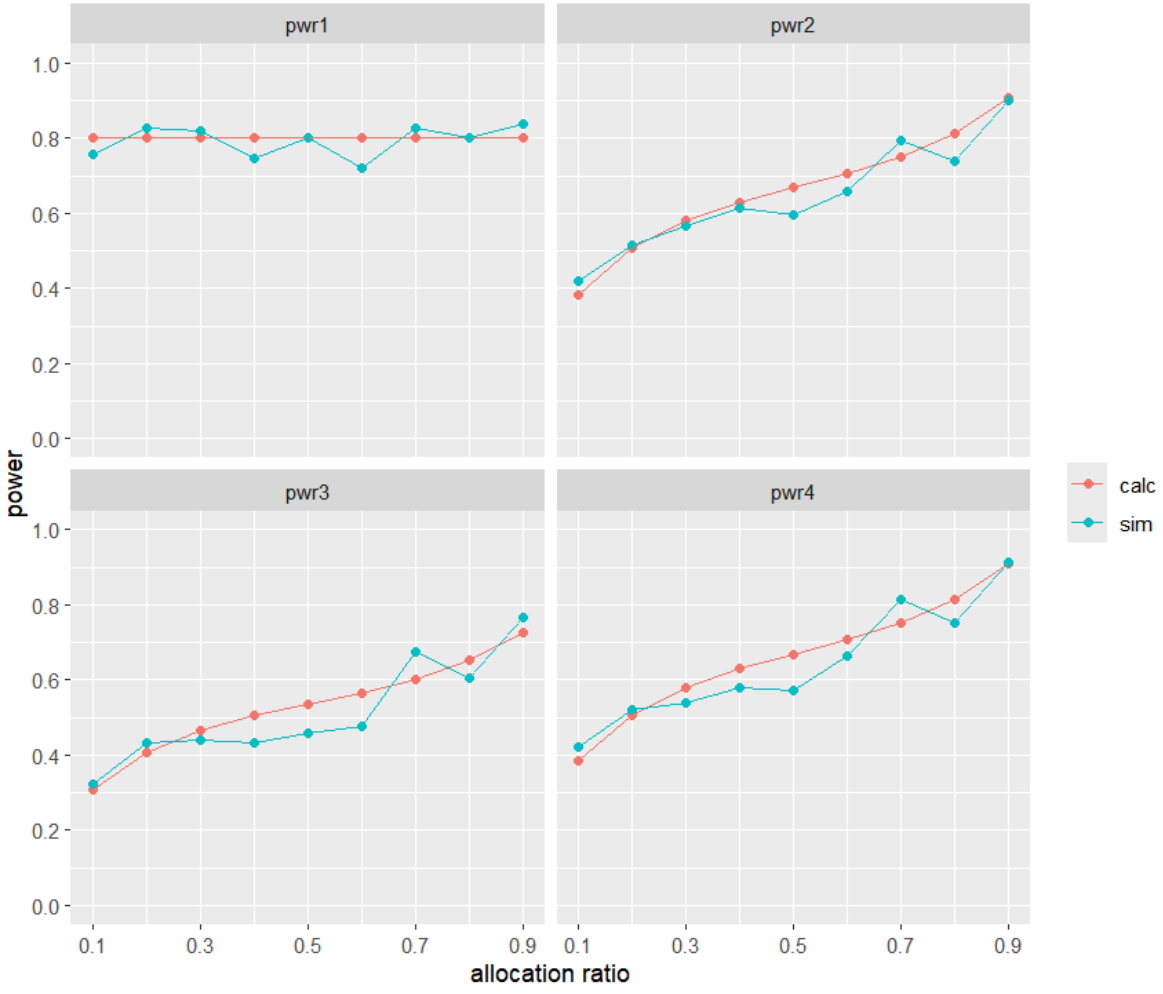
```
a2 <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 2, sim = FALSE)
b2 <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = -log(0.7), pi = 0.5, alpha = 0.025, beta = NA, N = 400, r = 1,
criterion = 2, sim = TRUE)
pplot(a2, b2)
```



低优

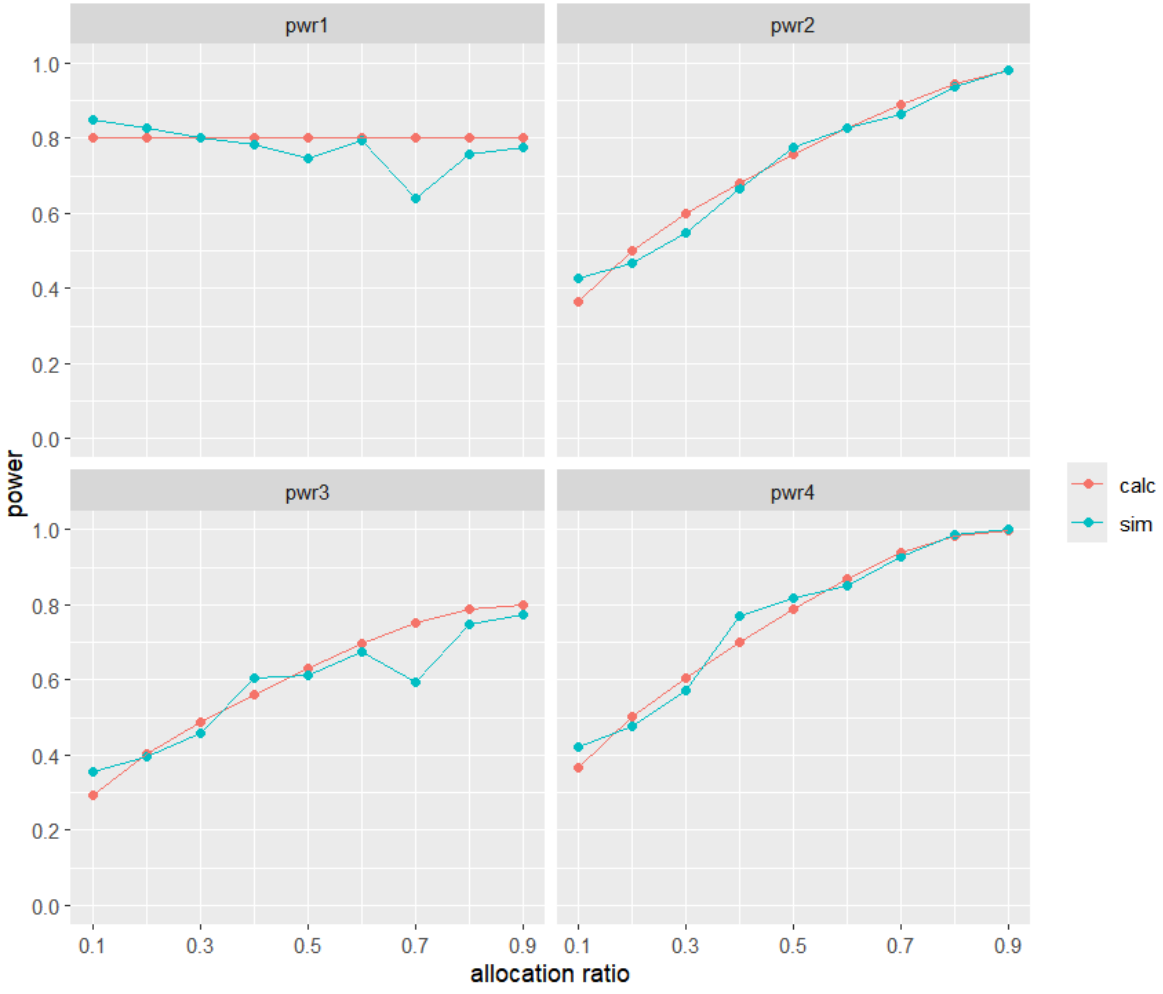
criterion=1

```
c1 <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 1, sim = FALSE)
d1 <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 1, sim = TRUE)
pplot(c1, d1)
```



criterion=2

```
c2 <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_nj = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 2, sim = FALSE)
d2 <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_nj = log(1.0), f = seq(0.1,
0.9, 0.1), cut = log(1.3), pi = 0.5, alpha = 0.025, beta = 0.2, N = NA, r = 1,
criterion = 2, sim = TRUE)
pplot(c2, d2)
```

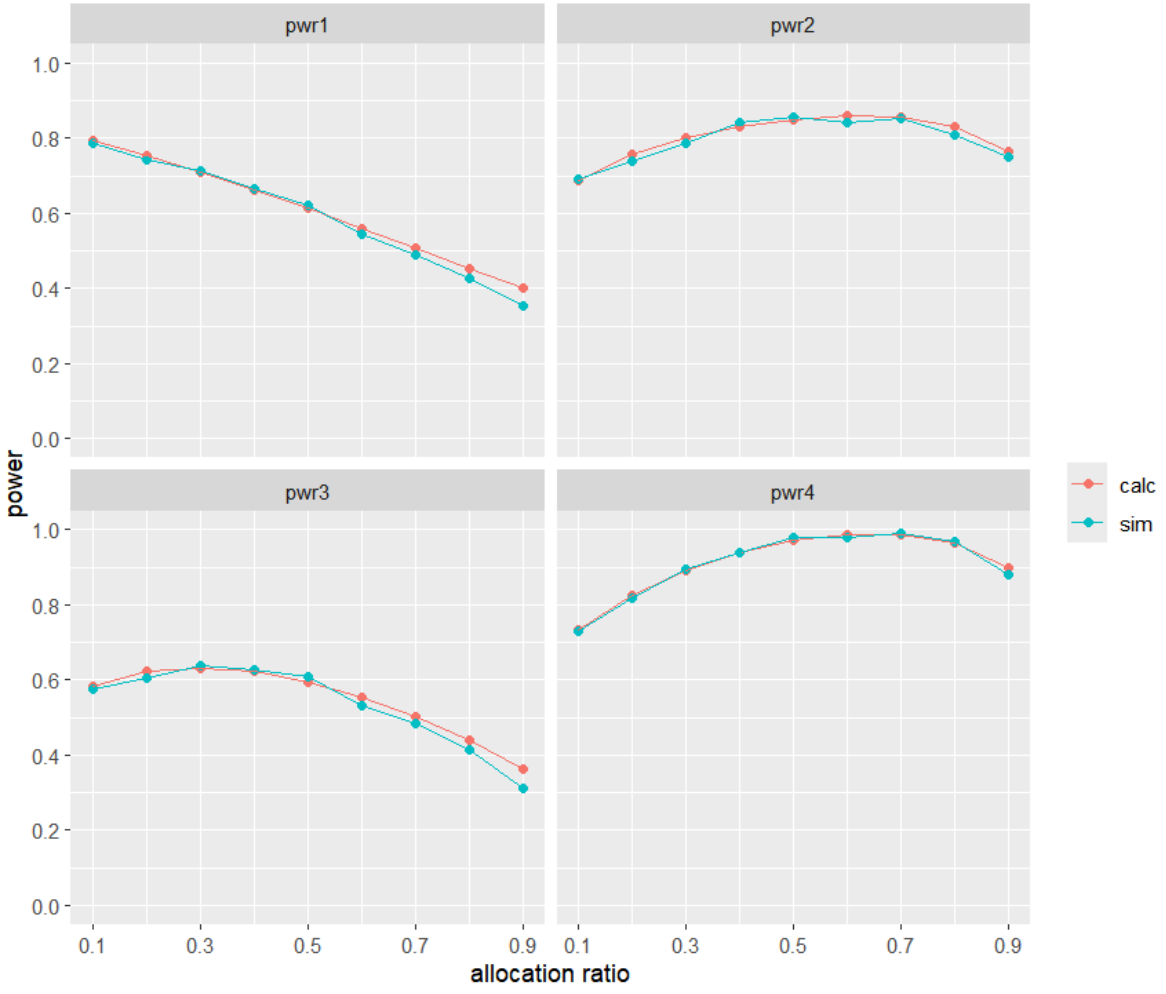


getPwr_Surv_Super_JM2

高优

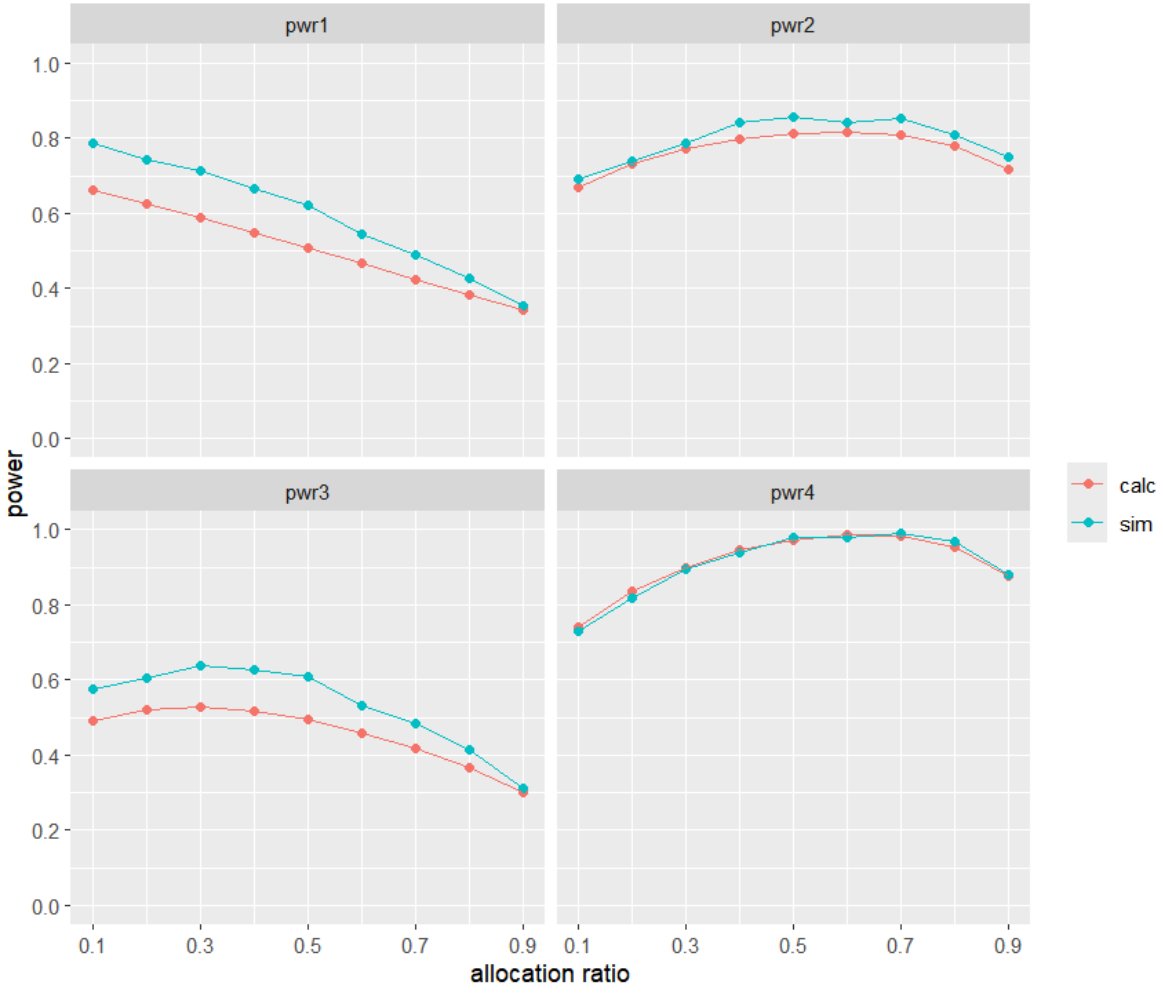
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(1.2), log(1.4)), fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(1.2), log(1.4)), fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



criterion=2

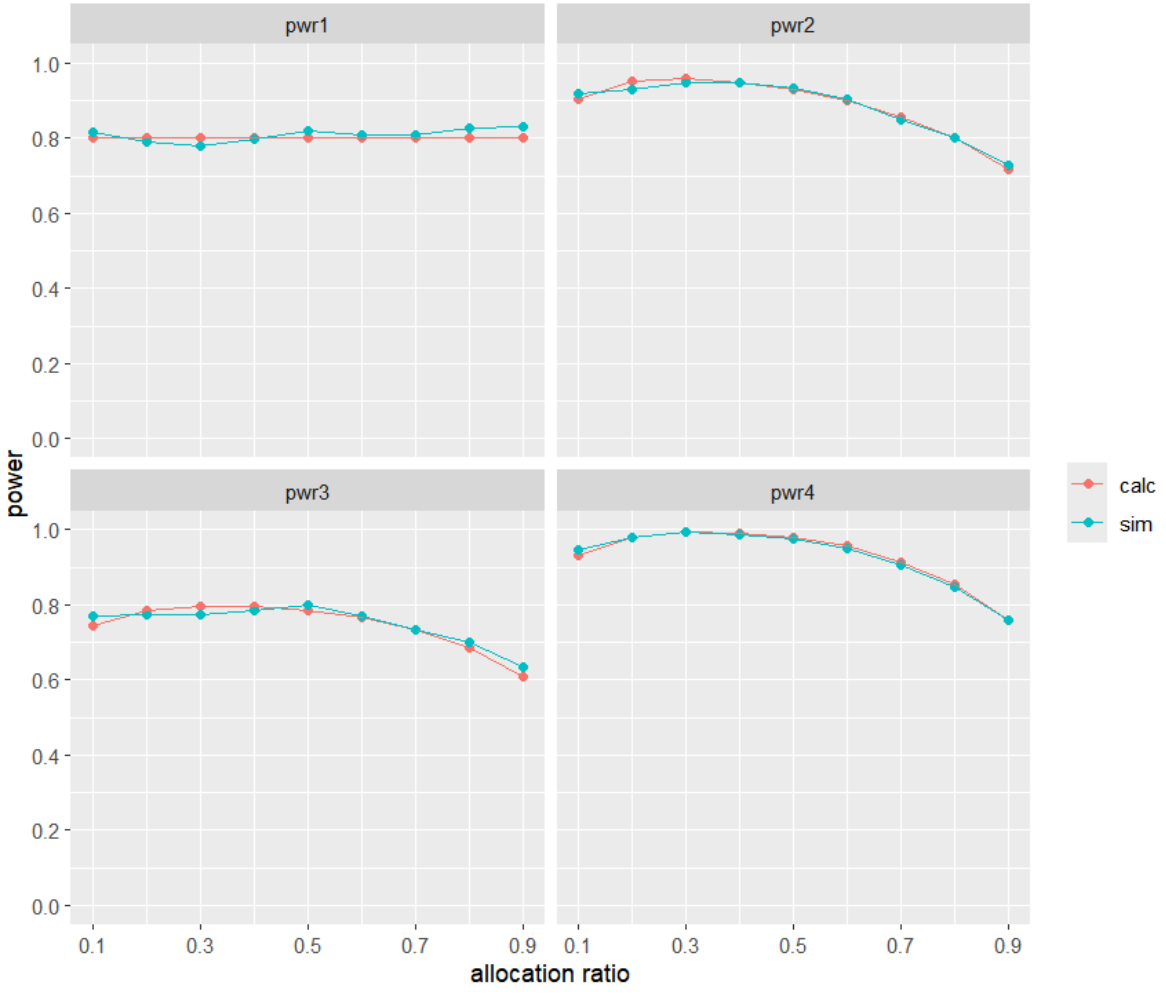
```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(1.2), log(1.4)), fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2, sim = FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(1.2), log(1.4)), fi = c(f, 1 - f),
alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



低优

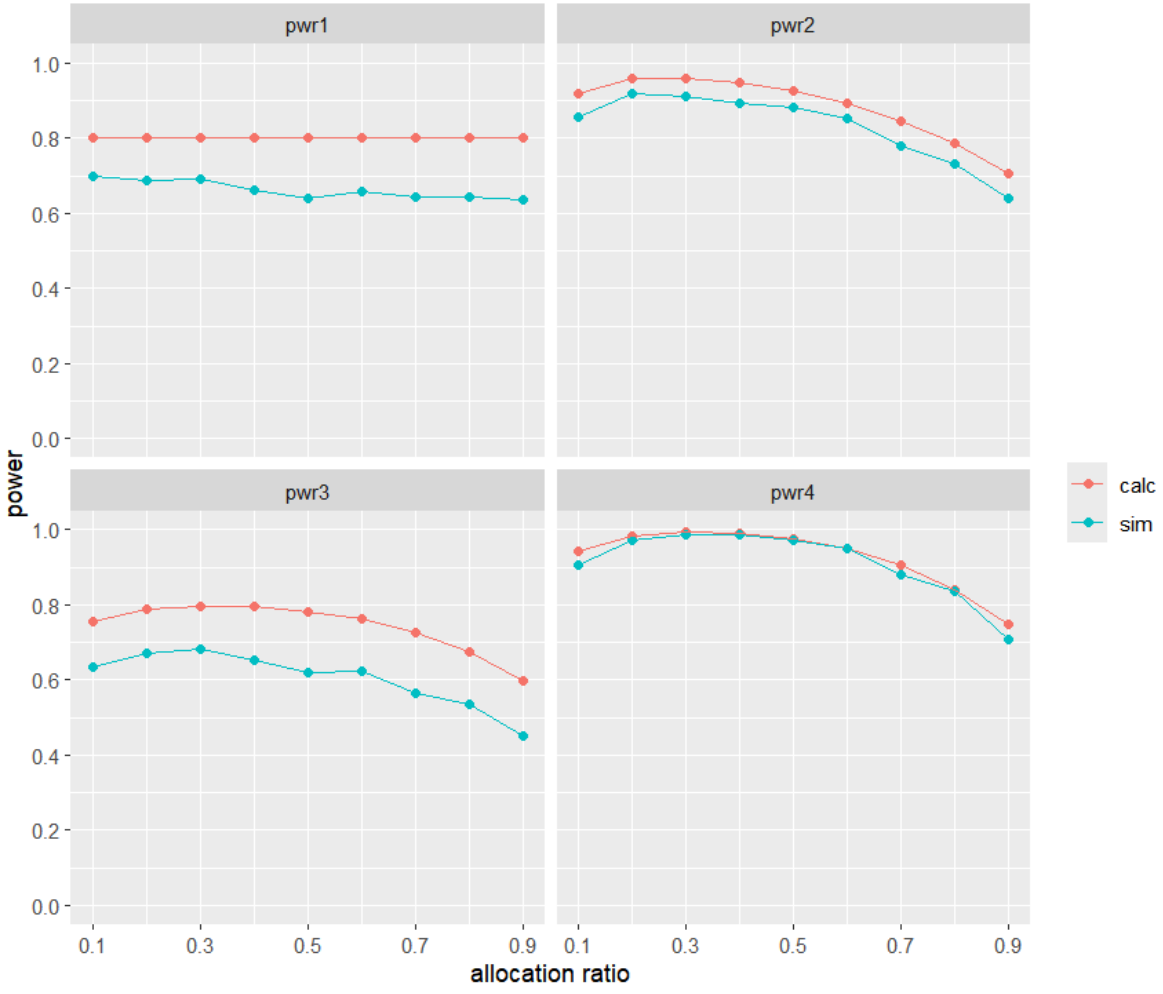
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(0.7), log(0.8)), fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim = FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(0.7), log(0.8)), fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



criterion=2

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(0.7), log(0.8)), fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim = FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Super_JM2(delta_i = c(log(0.7), log(0.8)), fi = c(f, 1 - f),
alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```

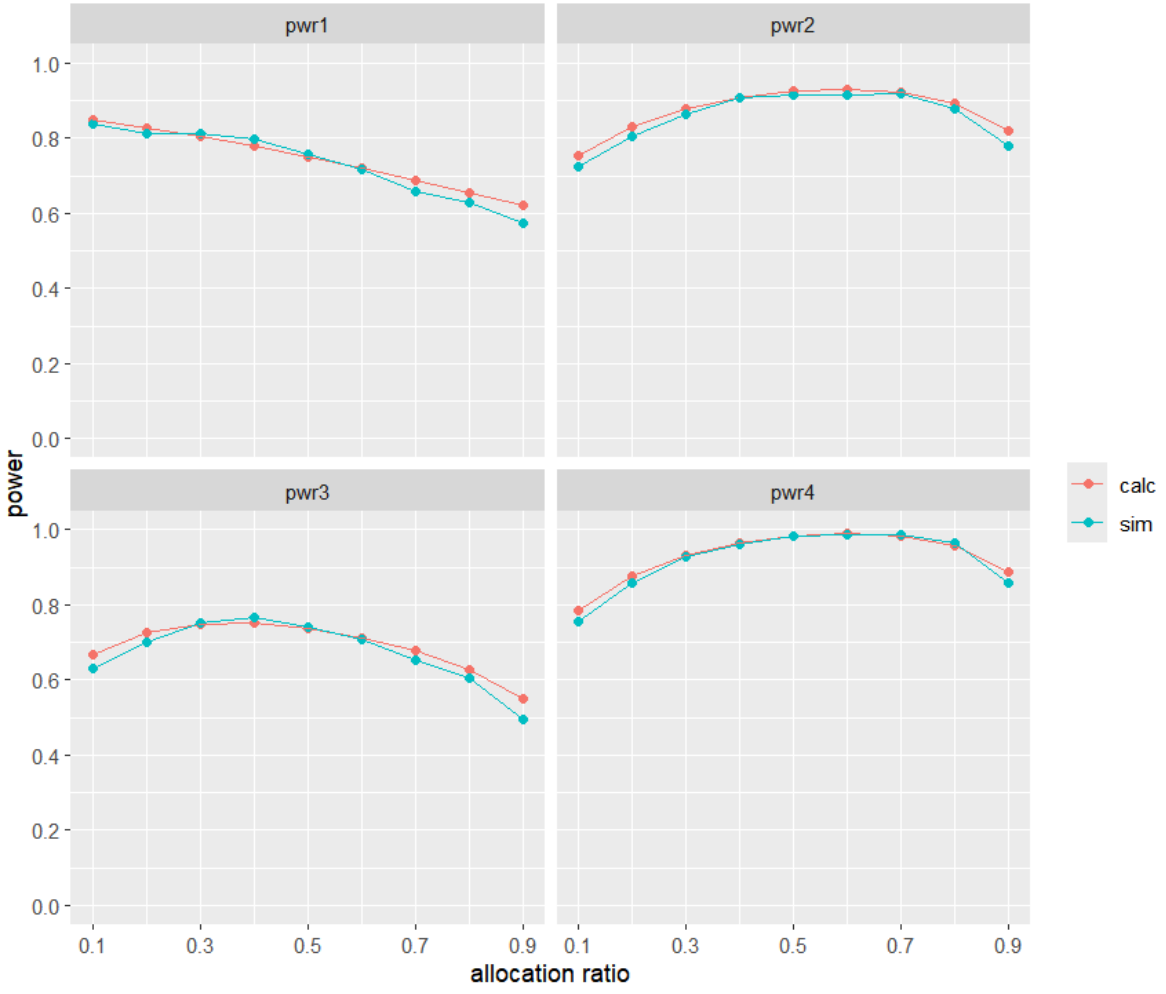


getPwr_Surv_Noninf_JM2

高优

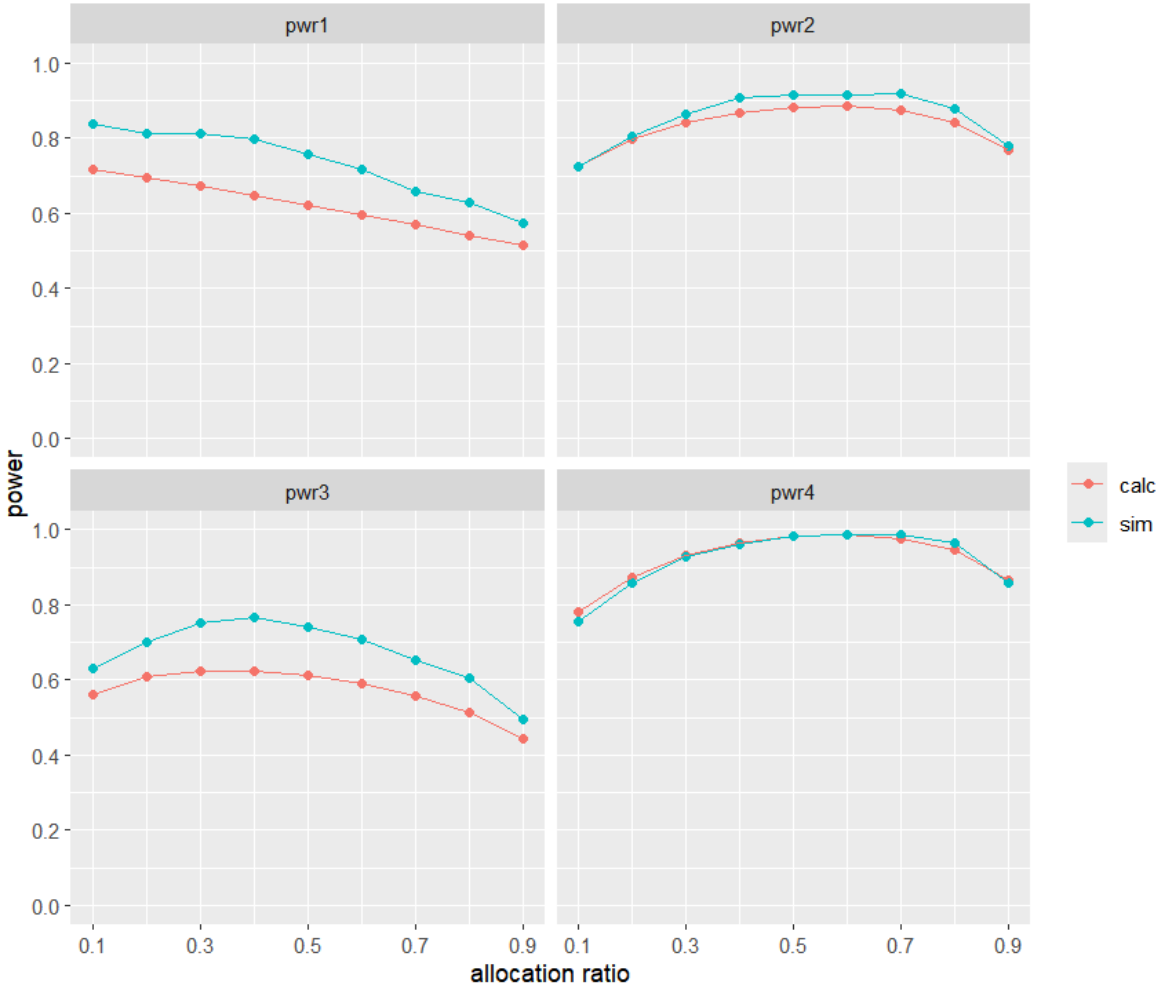
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 -
f), cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1,
direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 -
f), cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1,
direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



criterion=2

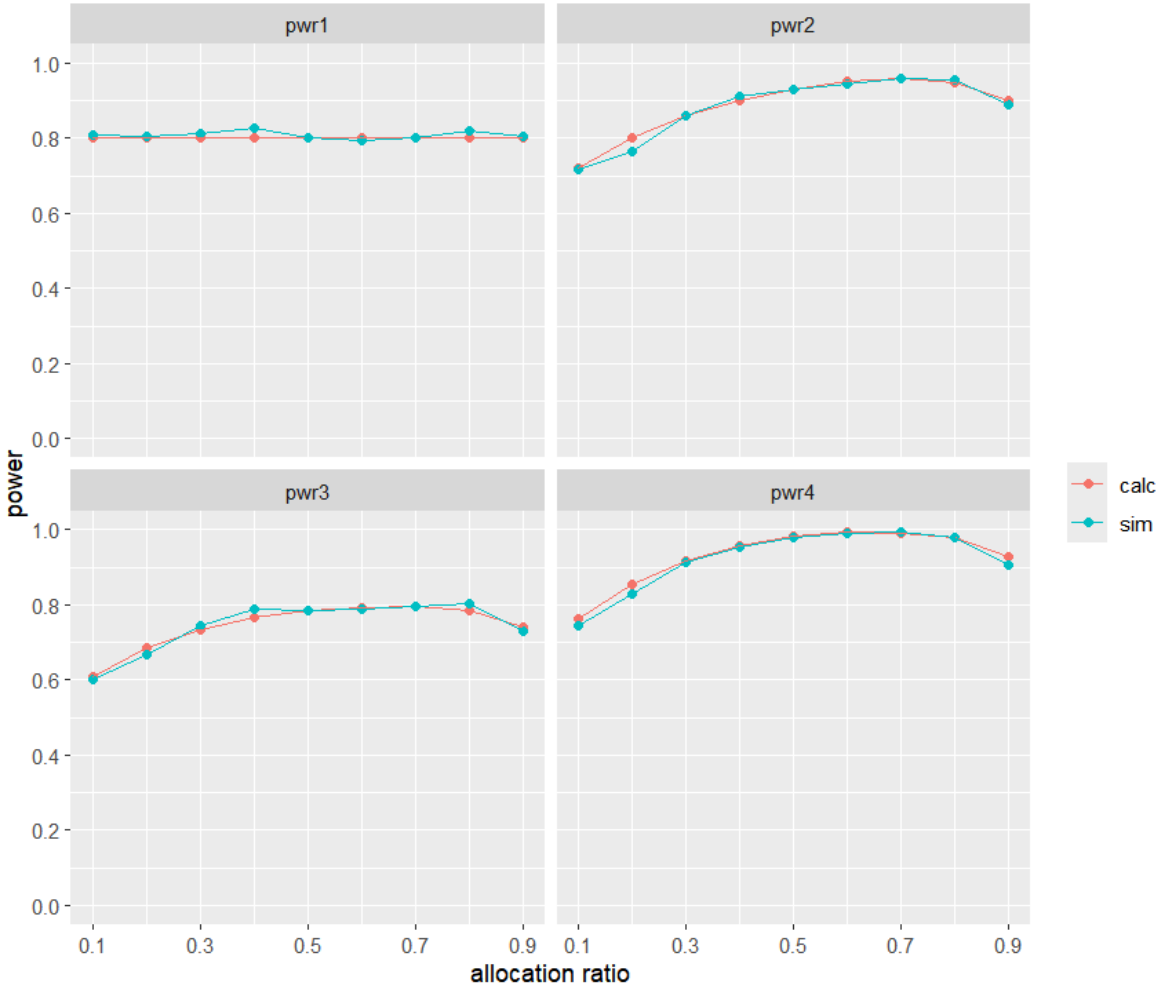
```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 -
f), cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2,
direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 -
f), cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2,
direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



低优

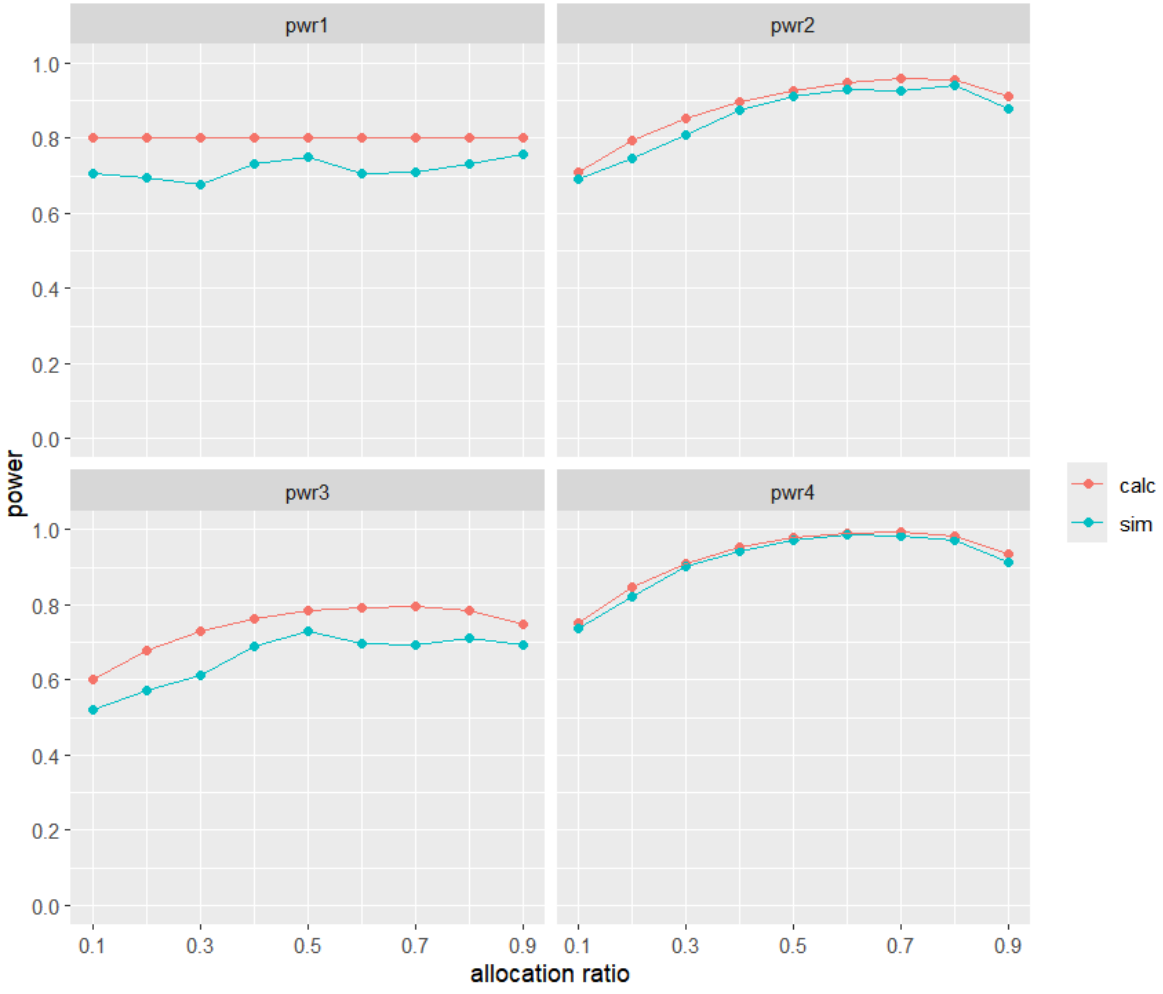
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 -
f), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1,
direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 -
f), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1,
direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



criterion=2

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 -
f), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2,
direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 -
f), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2,
direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```

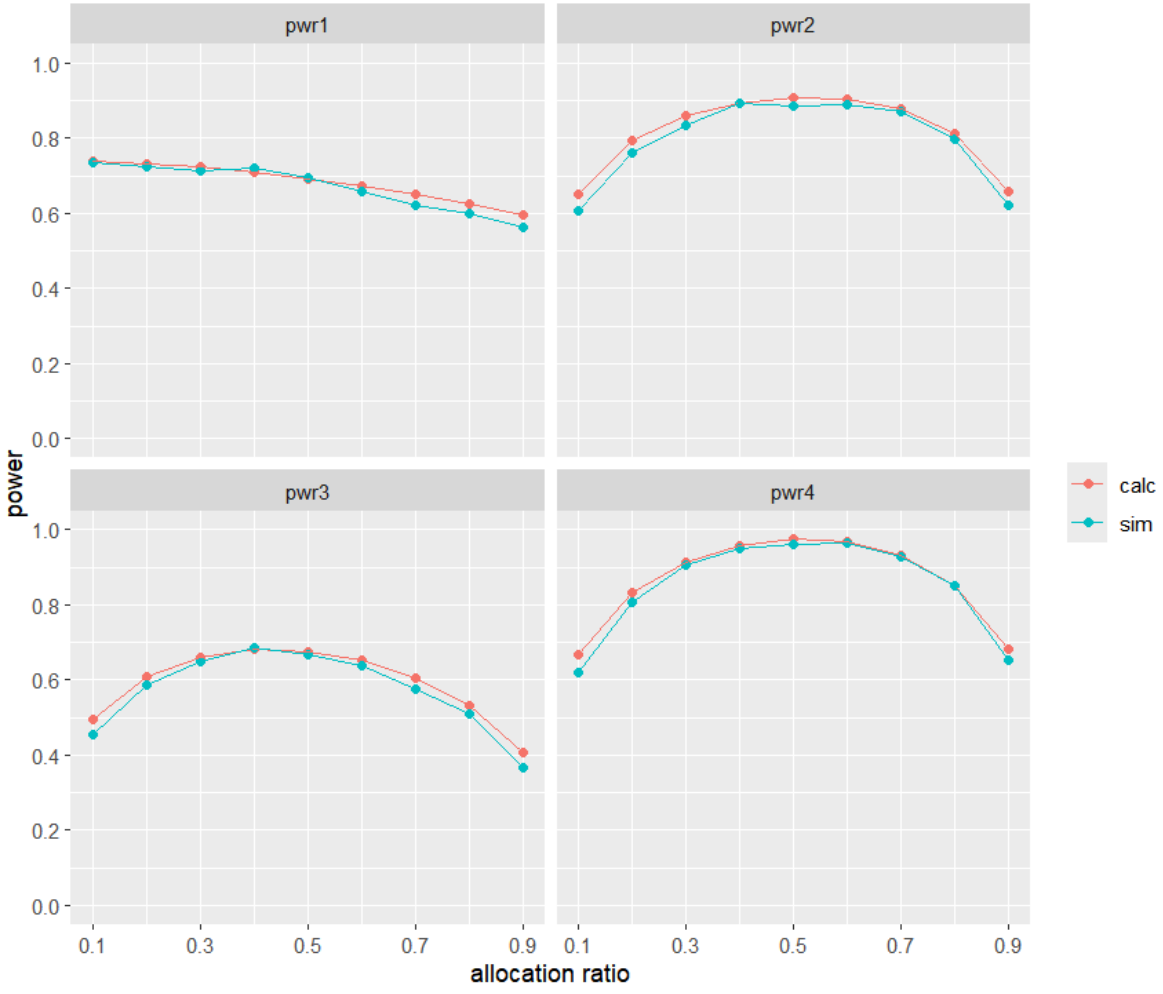


getPwr_Surv_Equi_JM2

高优

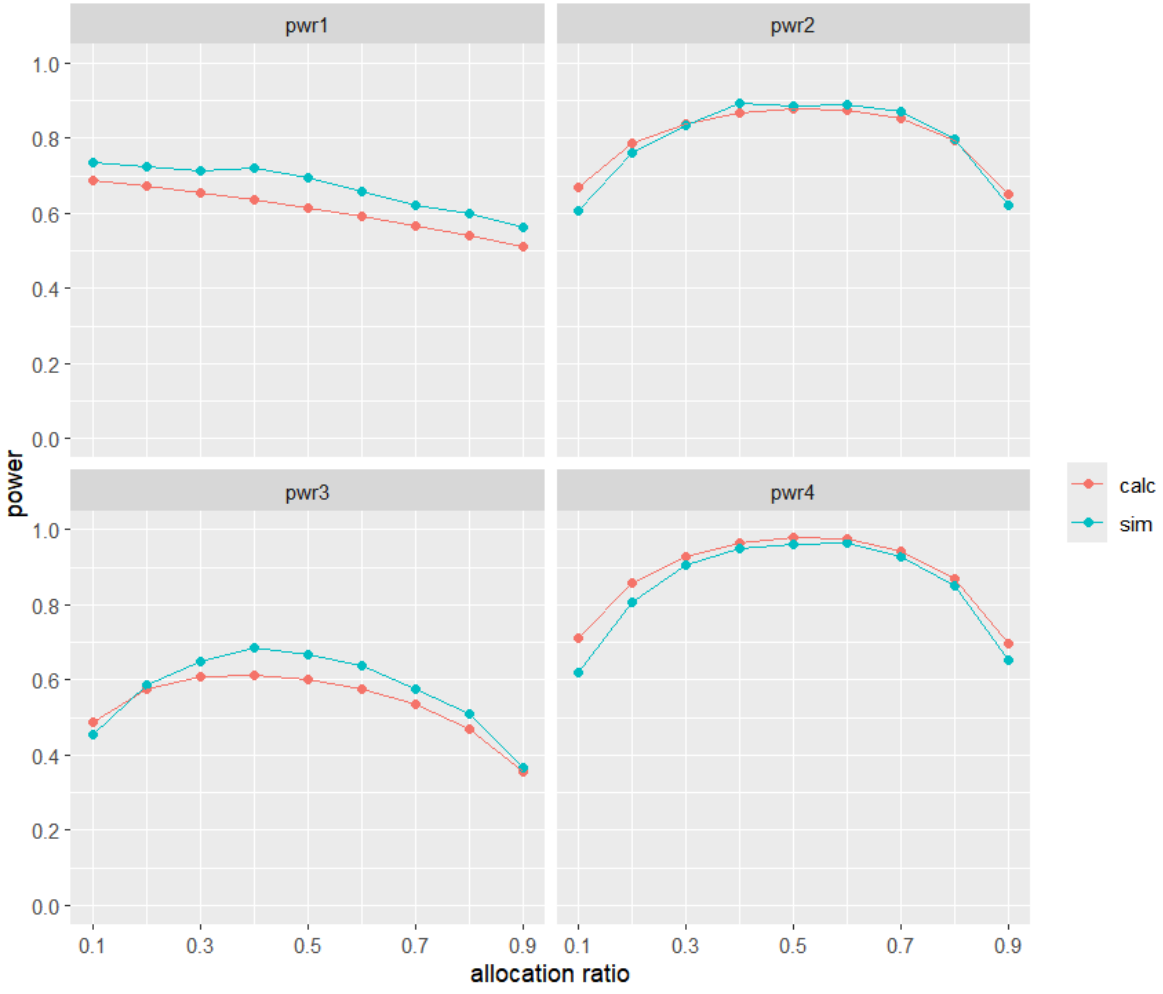
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
a1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 - f),
cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1, sim =
FALSE)$overall
  res$f <- f
  res
})
b1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 - f),
cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a1, b1)
```



criterion=2

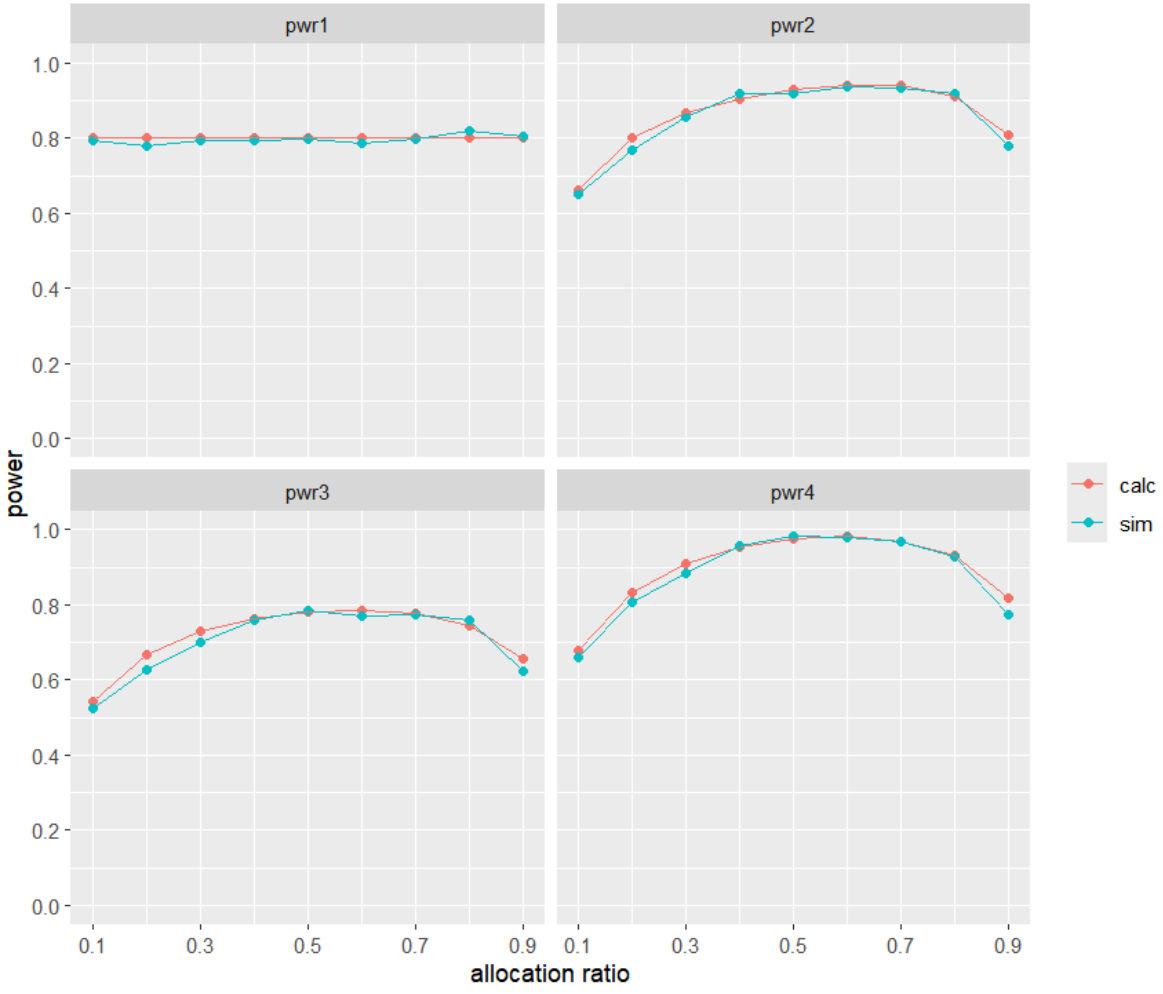
```
f_set <- seq(0.1, 0.9, 0.1)
a2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 - f),
cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2, sim =
FALSE)$overall
  res$f <- f
  res
})
b2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(0.9), log(1.0)), fi = c(f, 1 - f),
cut = -log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 2, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(a2, b2)
```



低优

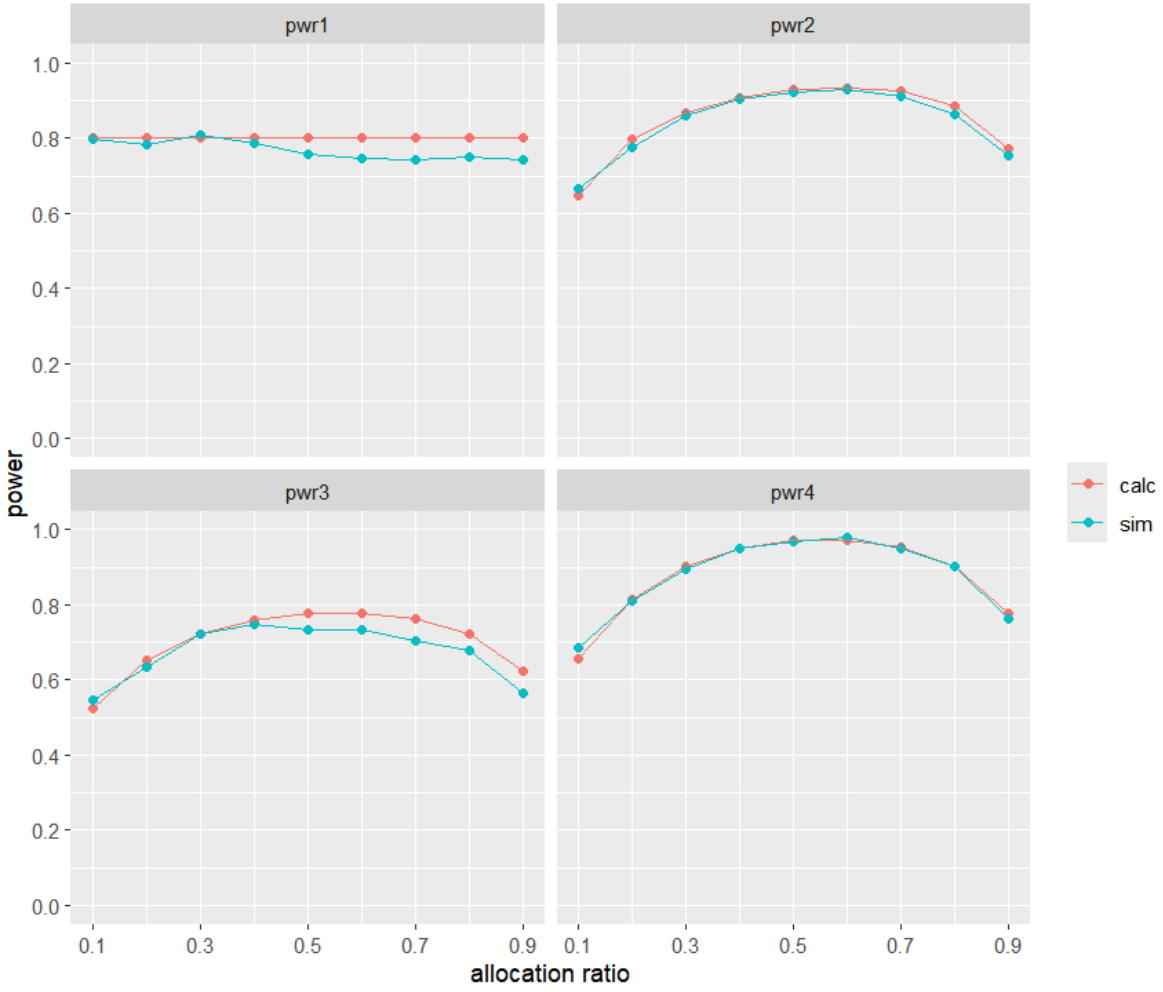
criterion=1

```
f_set <- seq(0.1, 0.9, 0.1)
c1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 - f),
cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim =
FALSE)$overall
  res$f <- f
  res
})
d1 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 - f),
cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c1, d1)
```



criterion=2

```
f_set <- seq(0.1, 0.9, 0.1)
c2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 - f),
cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim =
FALSE)$overall
  res$f <- f
  res
})
d2 <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Surv_Equi_JM2(delta_i = c(log(1.1), log(1.0)), fi = c(f, 1 - f),
cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 2, sim =
TRUE)$overall
  res$f <- f
  res
})
pplot(c2, d2)
```

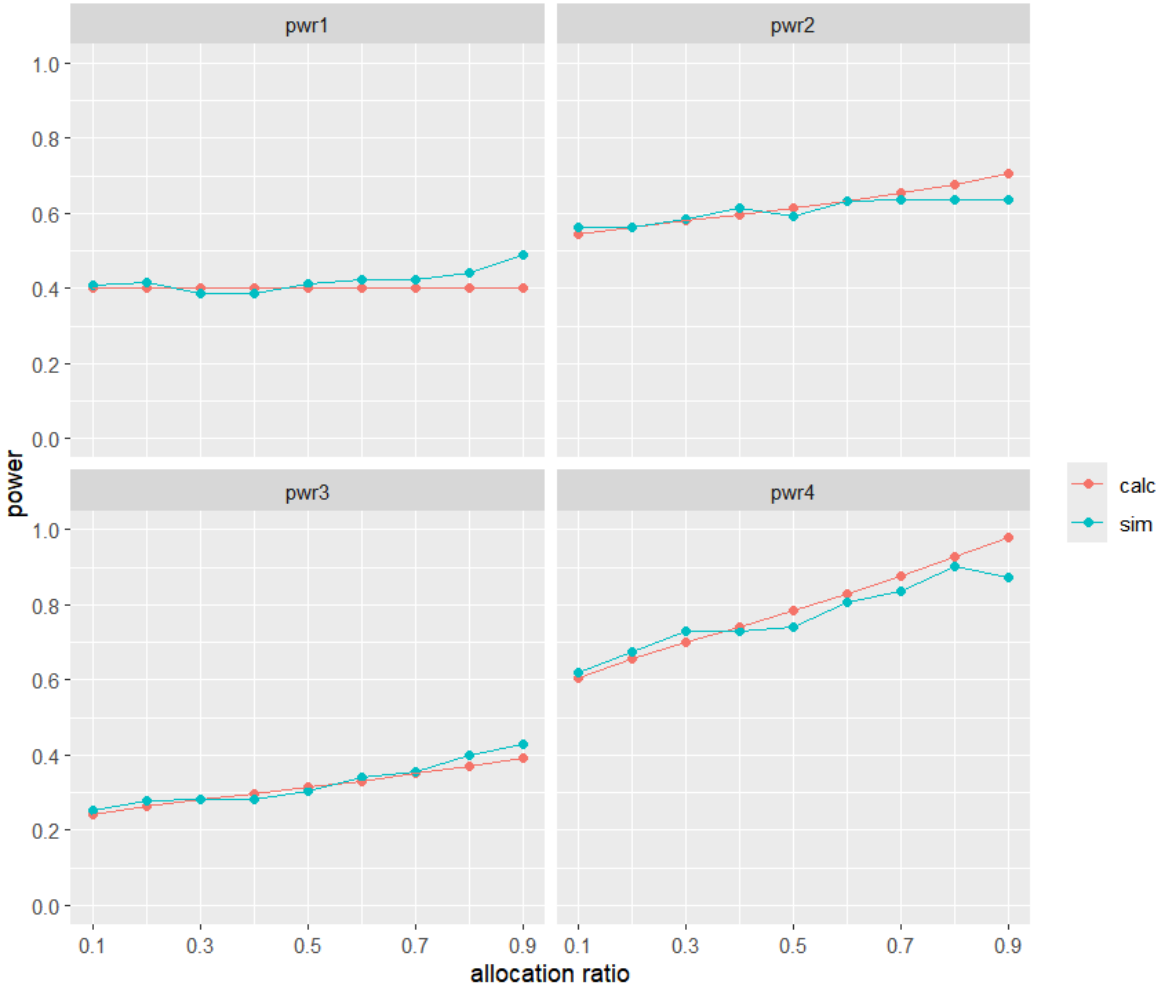


getPwr_Count_Super_JM1

高优

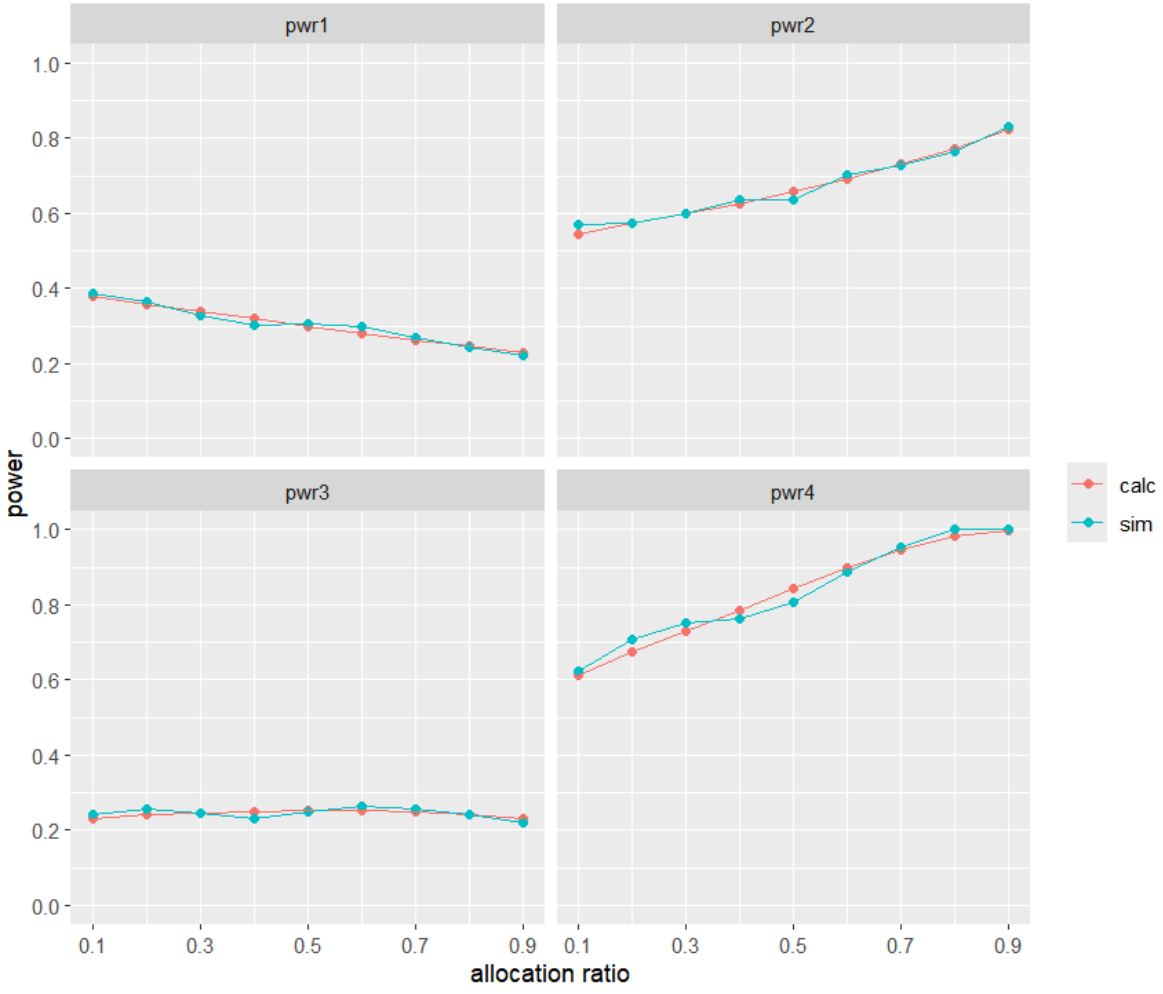
j & a

```
a1 <- getPwr_Count_Super_JM1(delta_j = log(1.2), delta_a = log(1.3), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = FALSE)
b1 <- getPwr_Count_Super_JM1(delta_j = log(1.2), delta_a = log(1.3), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = TRUE)
pplot(a1, b1)
```



j & nj

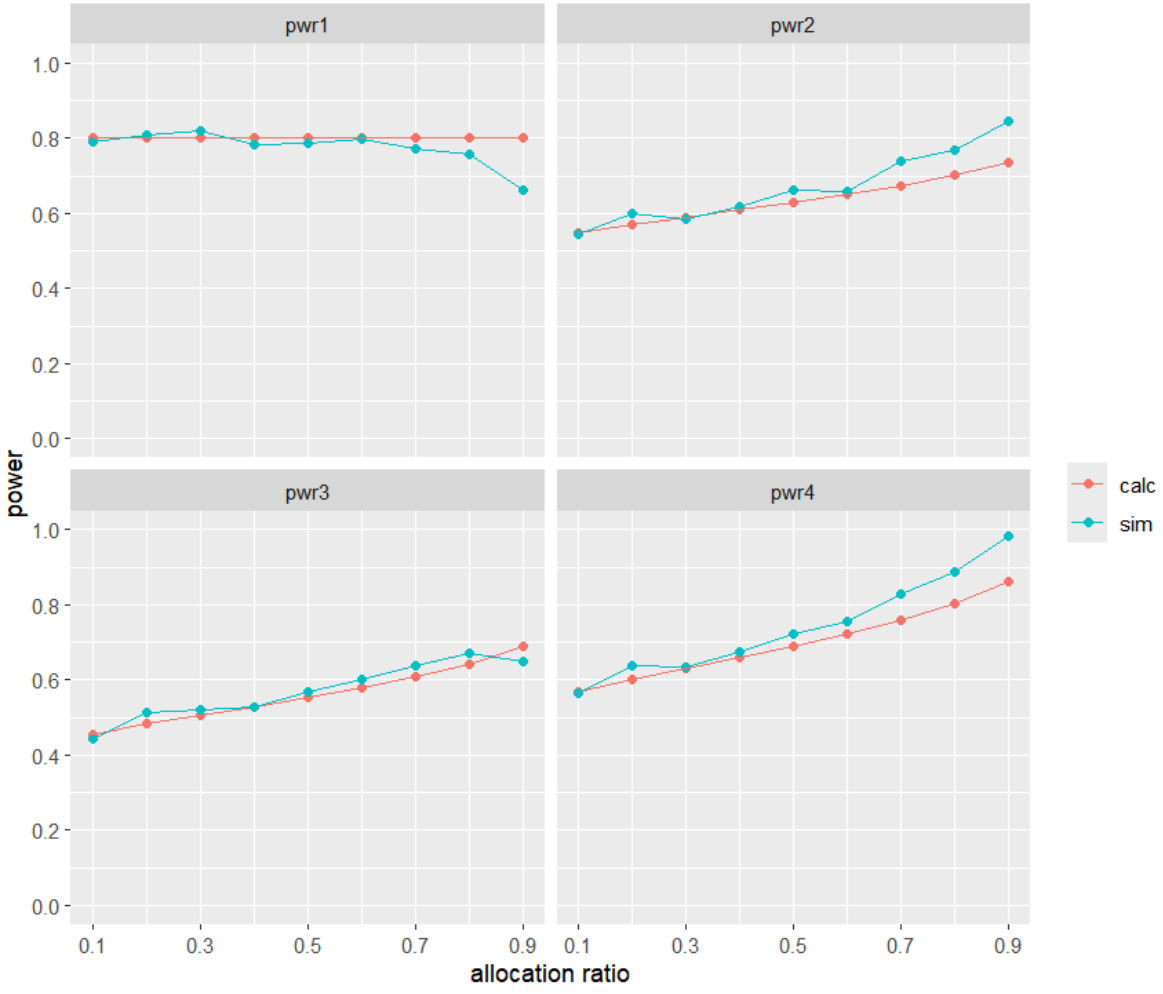
```
a2 <- getPwr_Count_Super_JM1(delta_j = log(1.2), delta_nj = log(1.3), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = FALSE)
b2 <- getPwr_Count_Super_JM1(delta_j = log(1.2), delta_nj = log(1.3), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = TRUE)
pplot(a2, b2)
```



低优

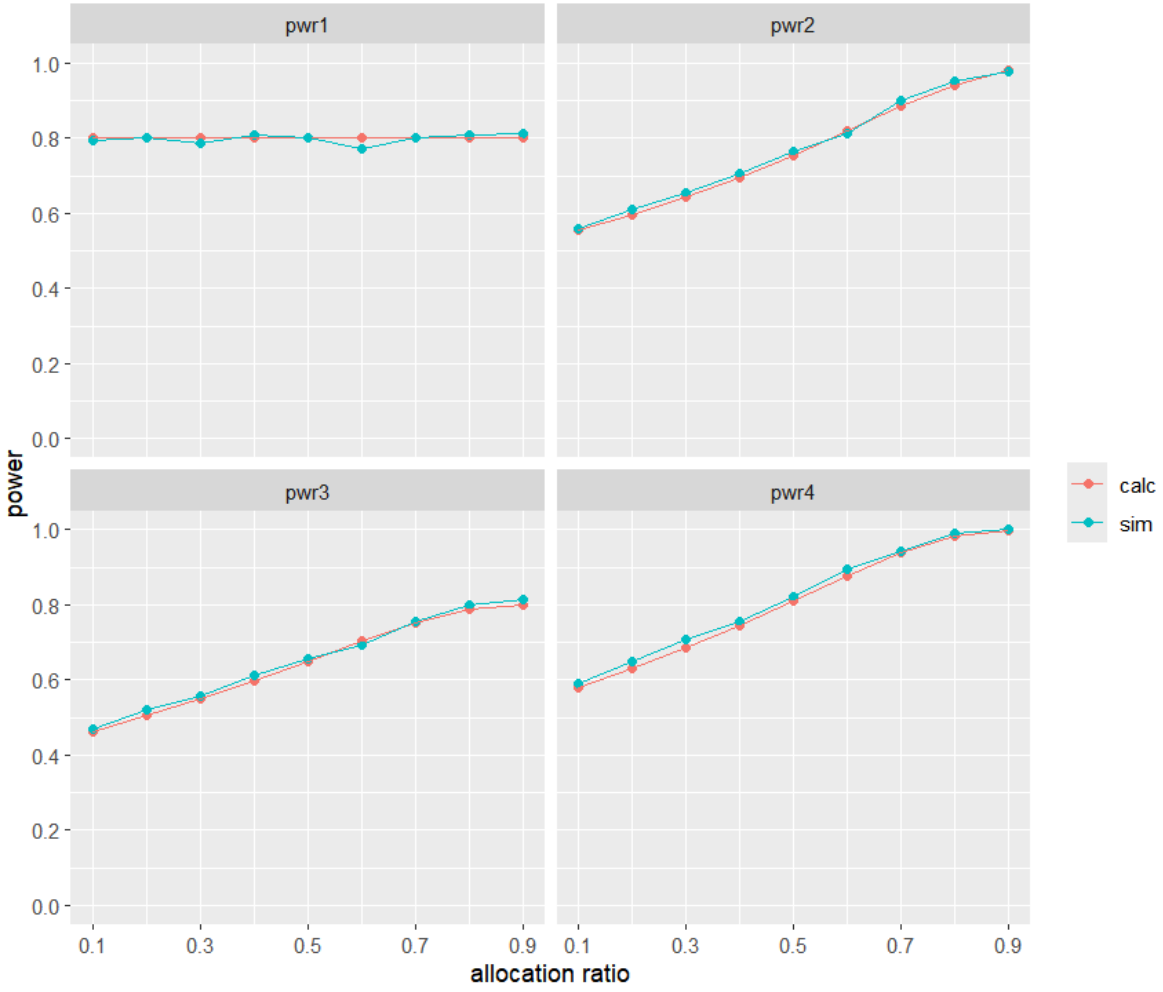
j & a

```
c1 <- getPwr_Count_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)
d1 <- getPwr_Count_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)
pplot(c1, d1)
```



j & nj

```
c2 <- getPwr_Count_Super_JM1(delta_j = log(0.8), delta_nj = log(0.7), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)
d2 <- getPwr_Count_Super_JM1(delta_j = log(0.8), delta_nj = log(0.7), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)
pplot(c2, d2)
```

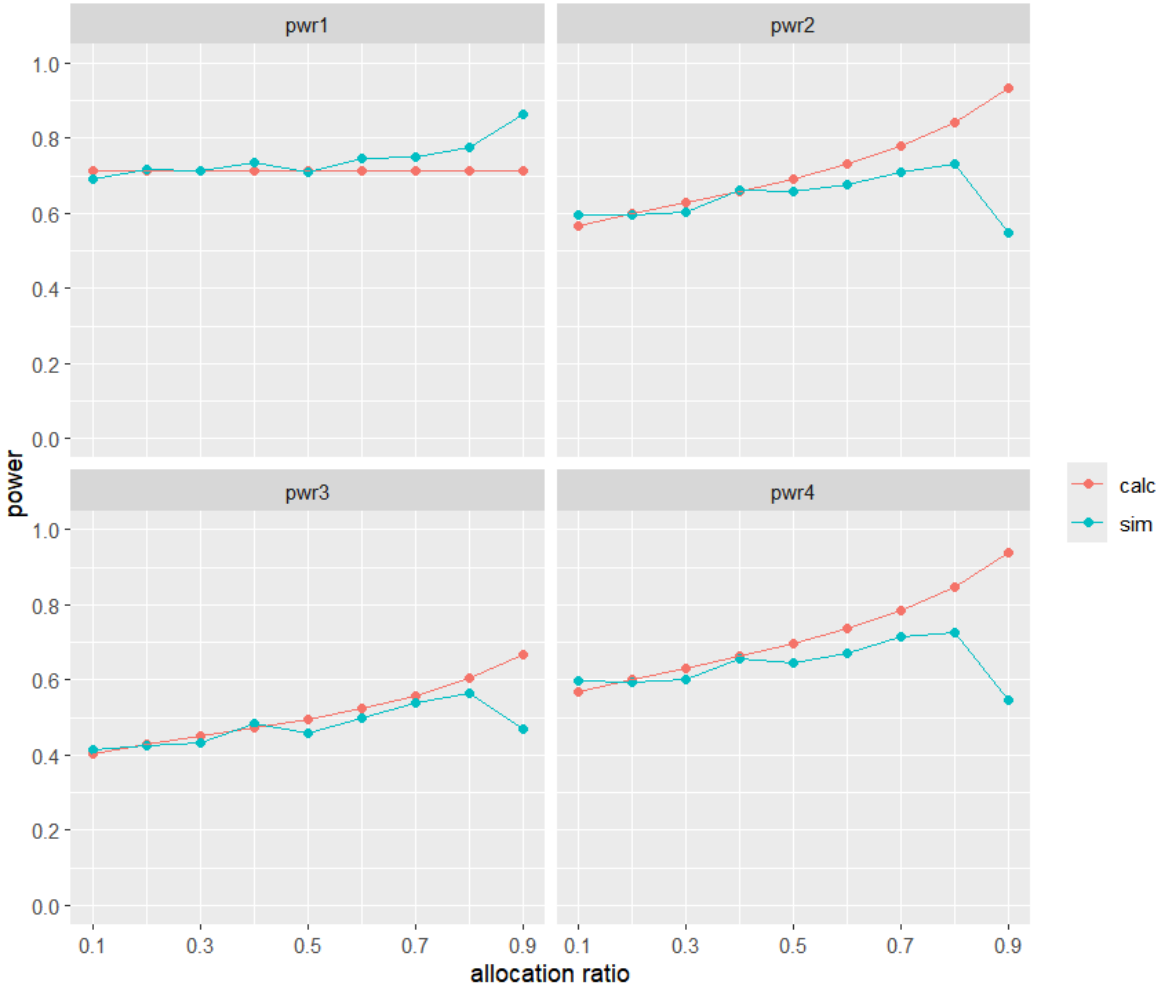


getPwr_Count_Noninf_JM1

高优

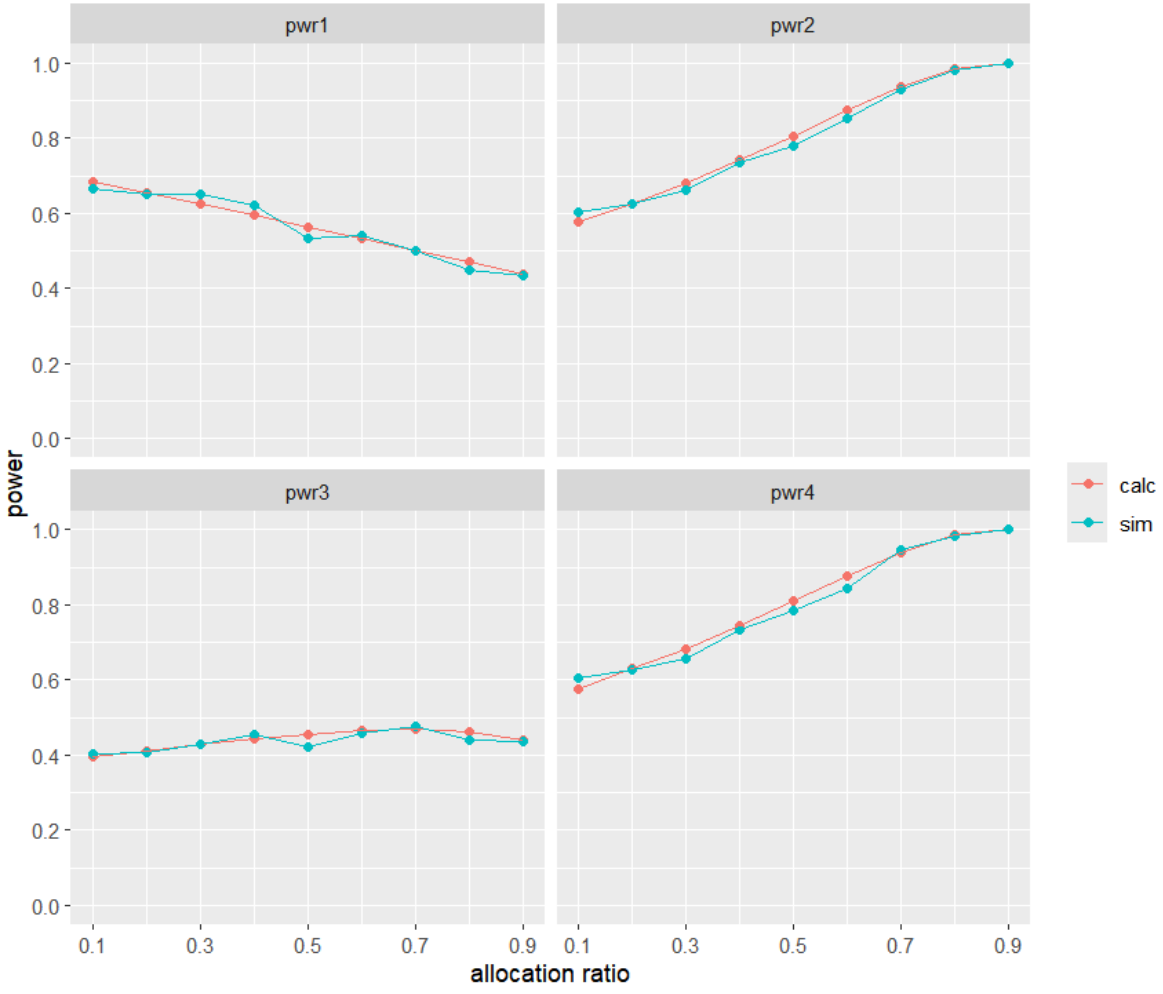
j & a

```
a1 <- getPwr_Count_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, direct = 1, sim = FALSE)
b1 <- getPwr_Count_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, direct = 1, sim = TRUE)
pplot(a1, b1)
```



j & nj

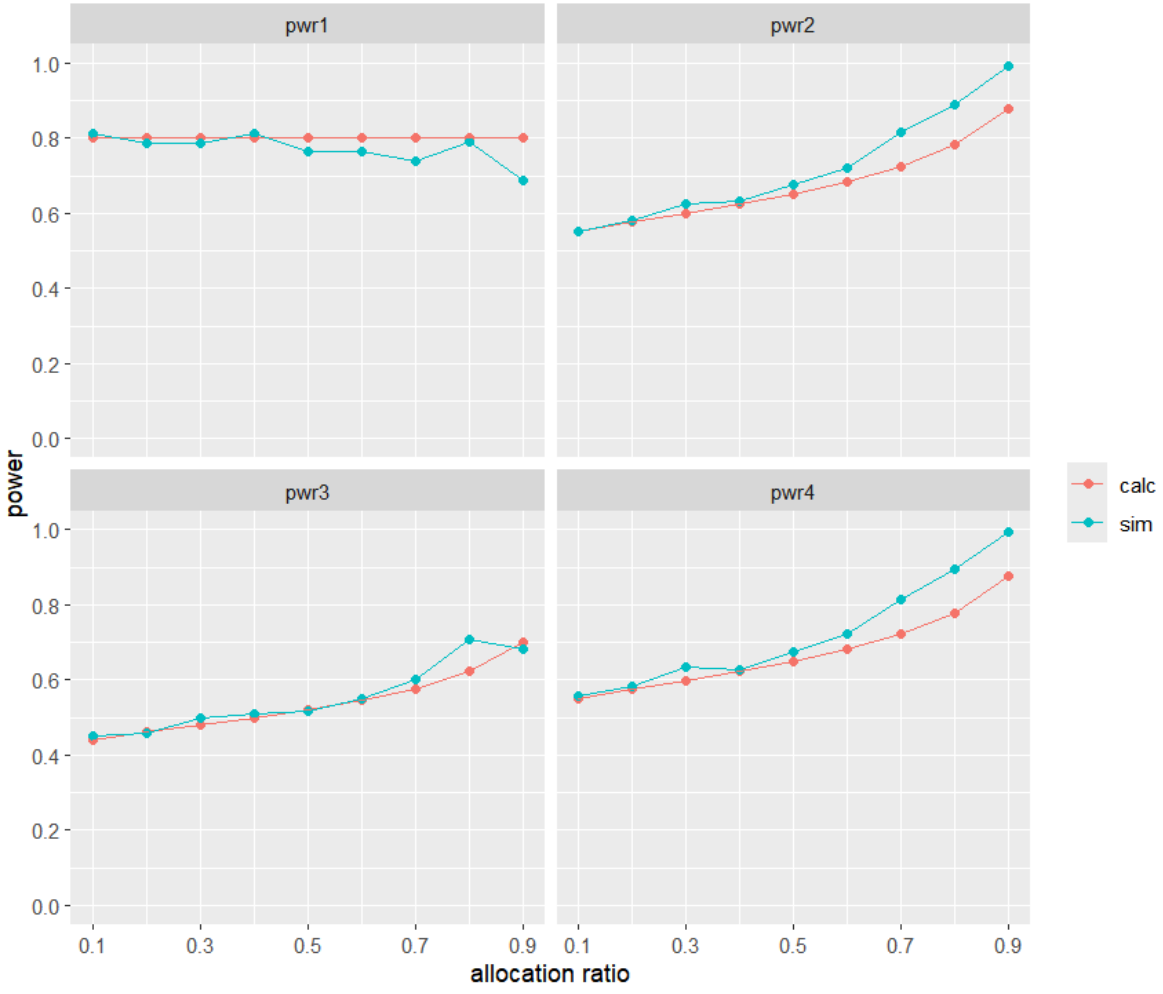
```
a2 <- getPwr_Count_Noninf_JM1(delta_j = log(0.9), delta_nj = log(1.0), lambda0_j
= 0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, direct = 1, sim = FALSE)
b2 <- getPwr_Count_Noninf_JM1(delta_j = log(0.9), delta_nj = log(1.0), lambda0_j
= 0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, direct = 1, sim = TRUE)
pplot(a2, b2)
```



低优

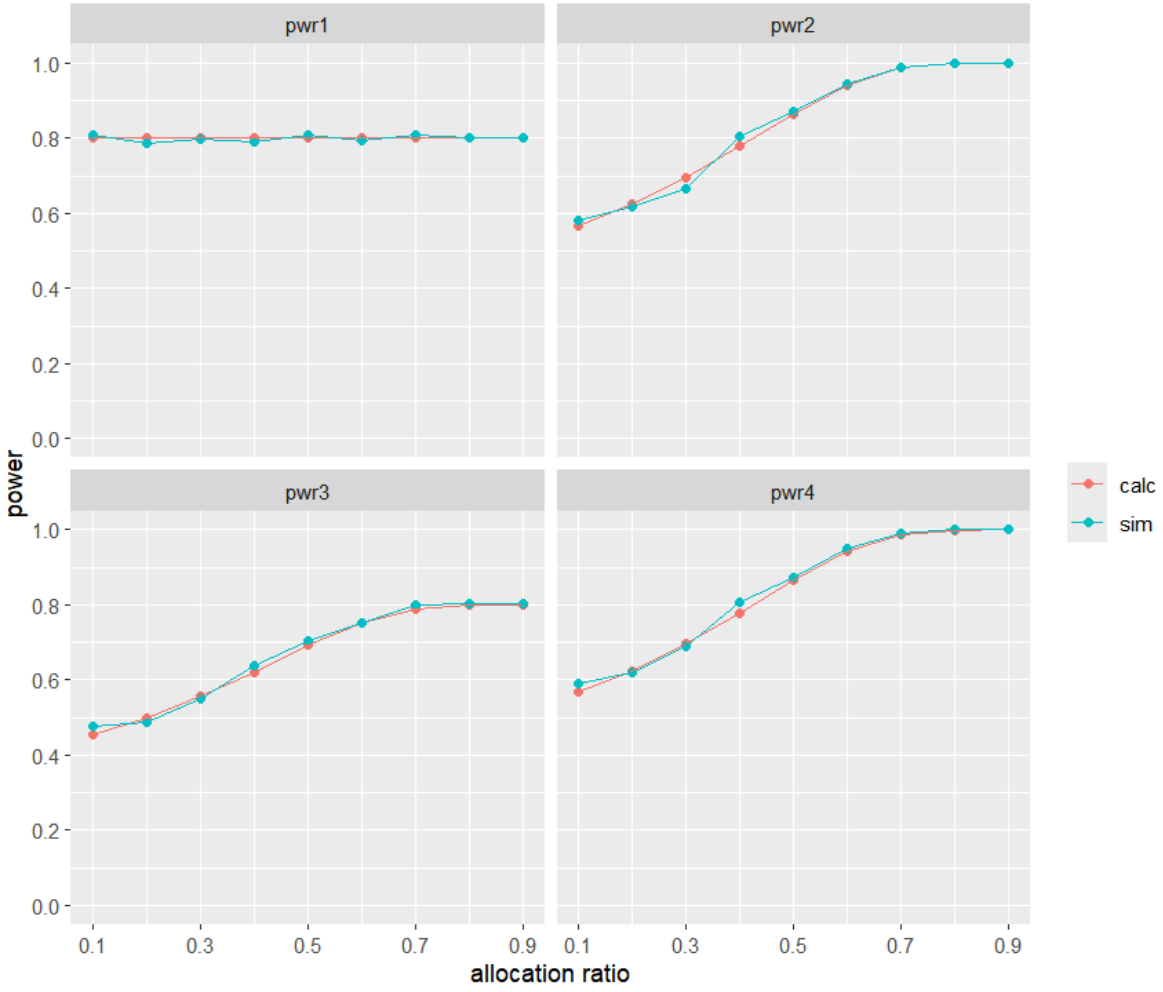
j & a

```
c1 <- getPwr_Count_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = FALSE)
d1 <- getPwr_Count_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = TRUE)
pplot(c1, d1)
```



j & nj

```
c1 <- getPwr_Count_Noninf_JM1(delta_j = log(1.1), delta_nj = log(1.0), lambda0_j
= 0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = FALSE)
d1 <- getPwr_Count_Noninf_JM1(delta_j = log(1.1), delta_nj = log(1.0), lambda0_j
= 0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, sim = TRUE)
pplot(c1, d1)
```

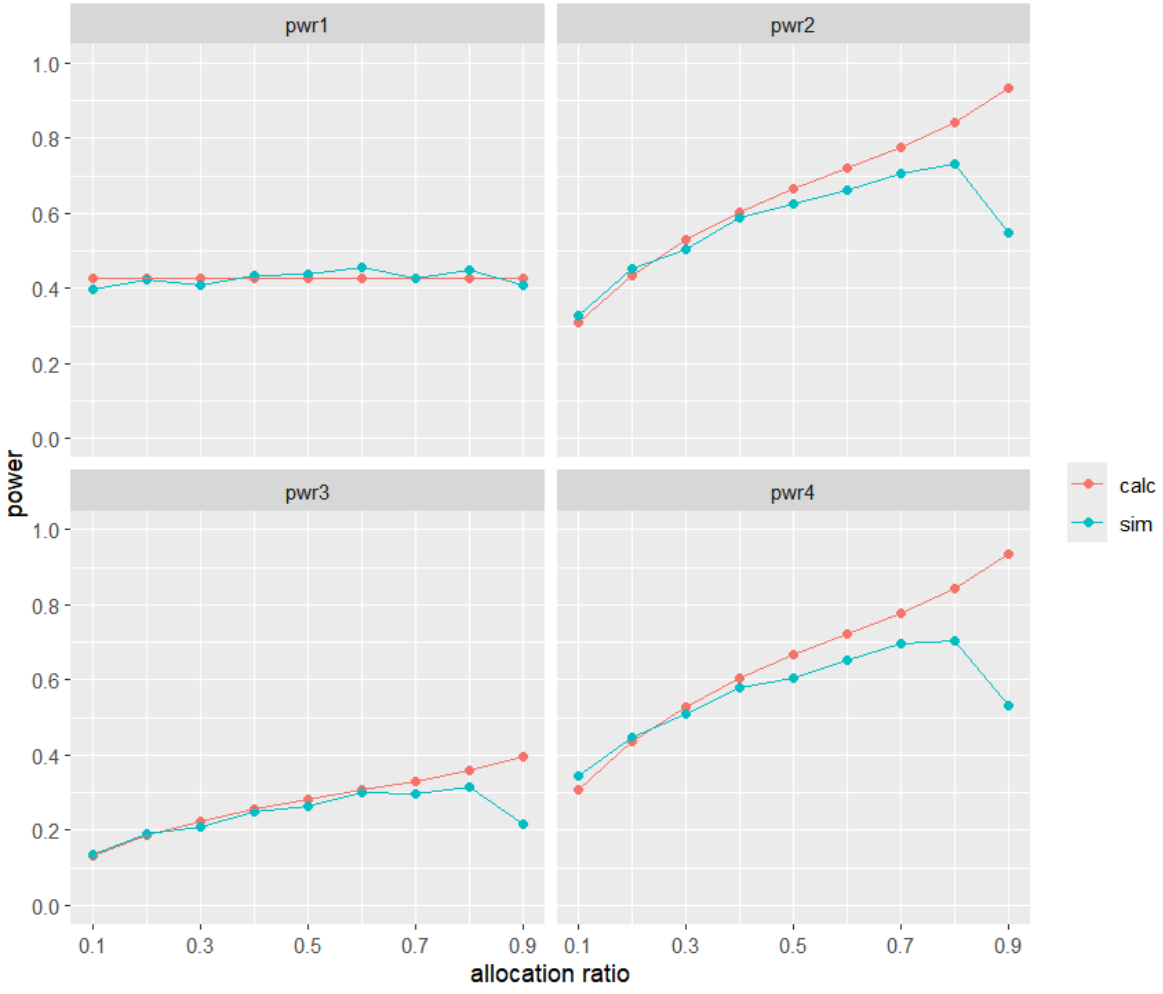


getPwr_Count_Equi_JM1

高优

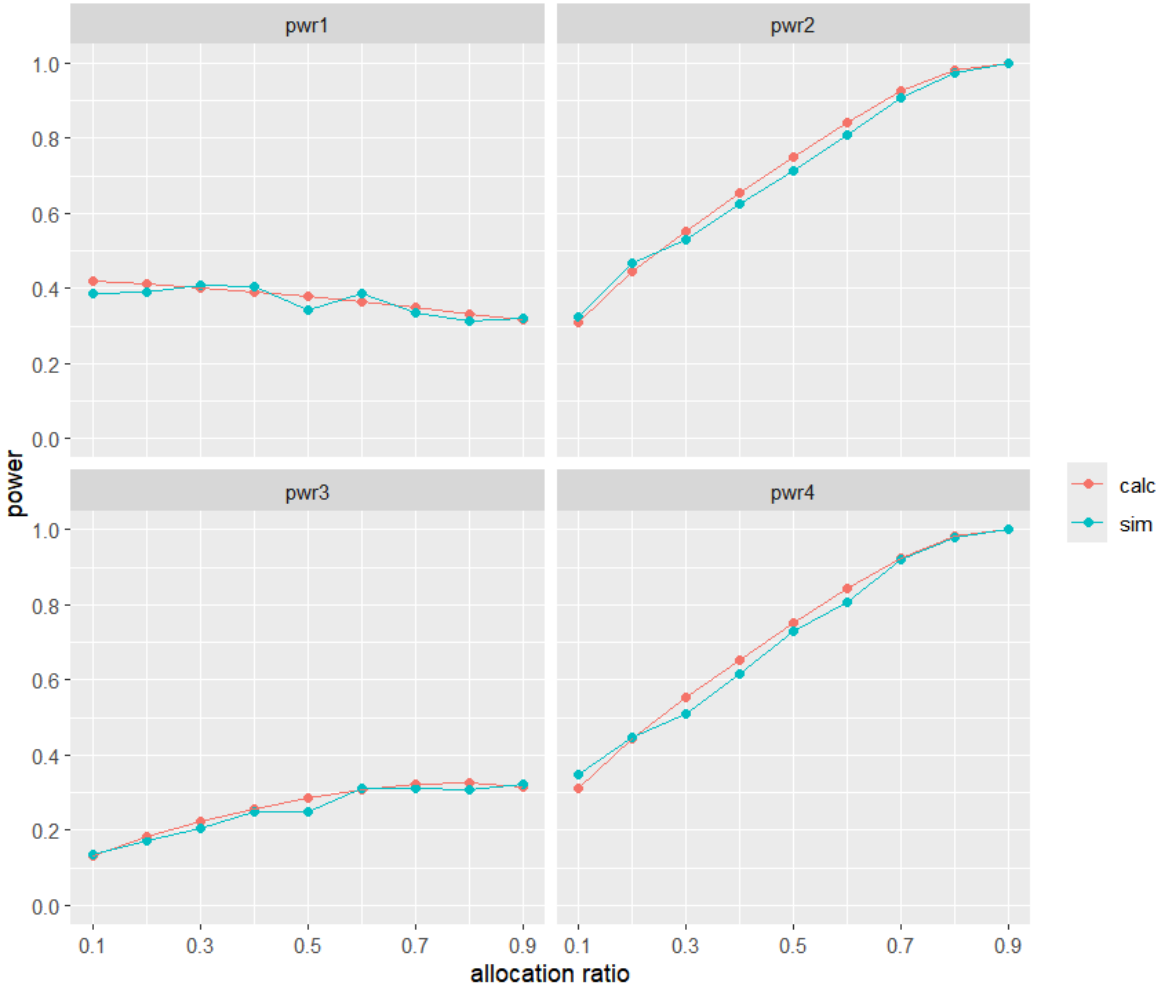
j & a

```
a1 <- getPwr_Count_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, sim = FALSE)
b1 <- getPwr_Count_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, sim = TRUE)
pplot(a1, b1)
```



j & nj

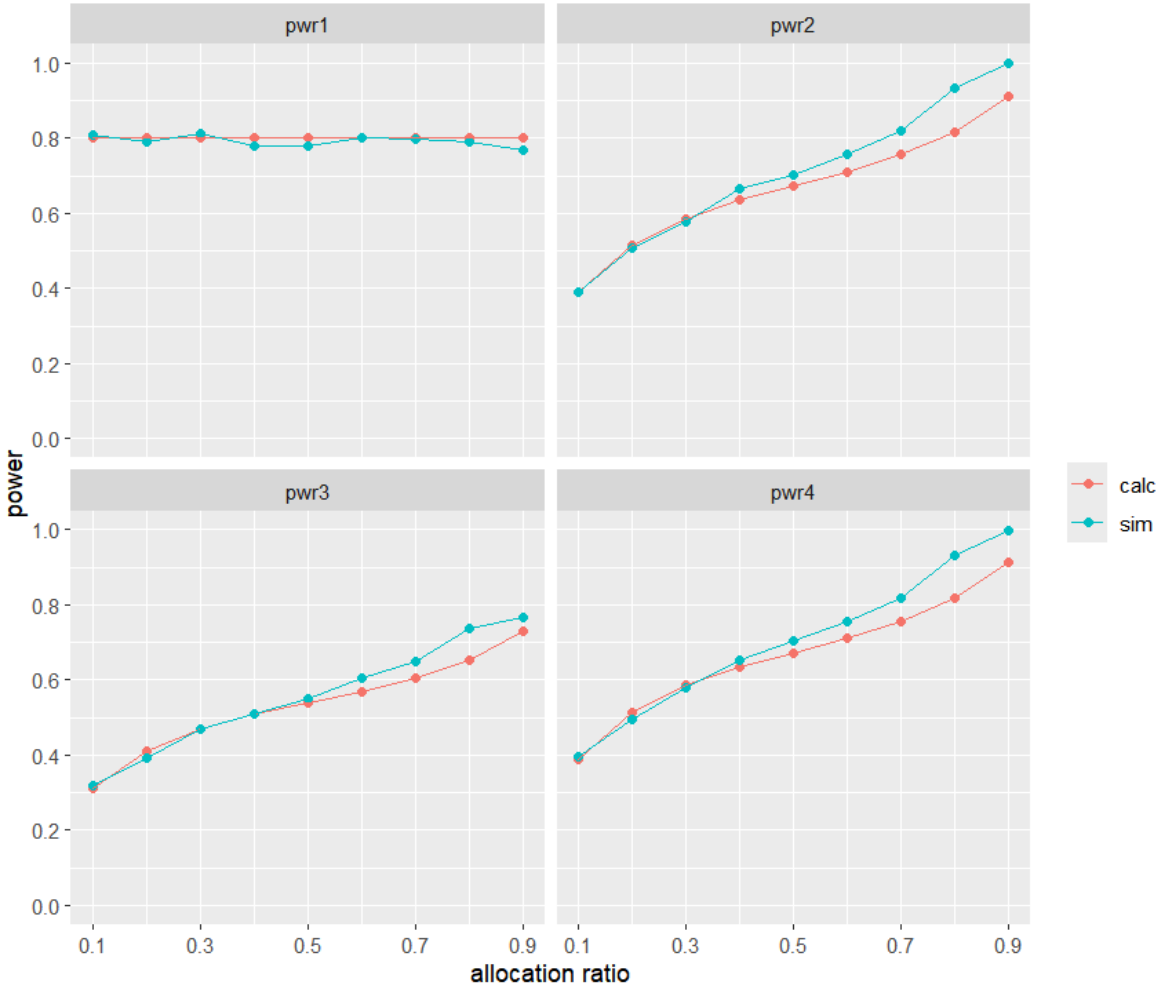
```
a2 <- getPwr_Count_Equi_JM1(delta_j = log(0.9), delta_nj = log(1.0), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, sim = FALSE)
b2 <- getPwr_Count_Equi_JM1(delta_j = log(0.9), delta_nj = log(1.0), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 400, r = 1, sim = TRUE)
pplot(a2, b2)
```



低优

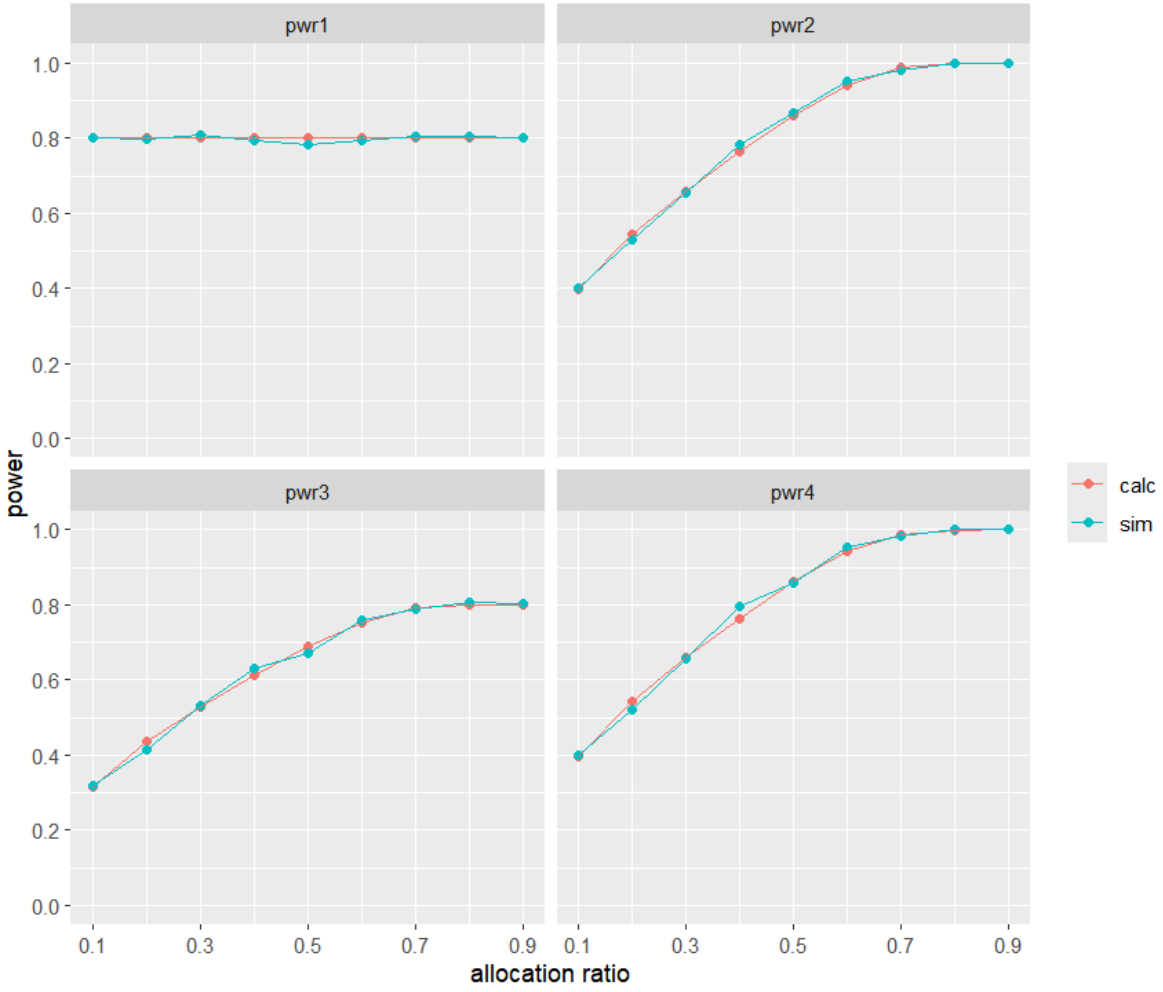
j & a

```
c1 <- getPwr_Count_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)
d1 <- getPwr_Count_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)
pplot(c1, d1)
```



j & nj

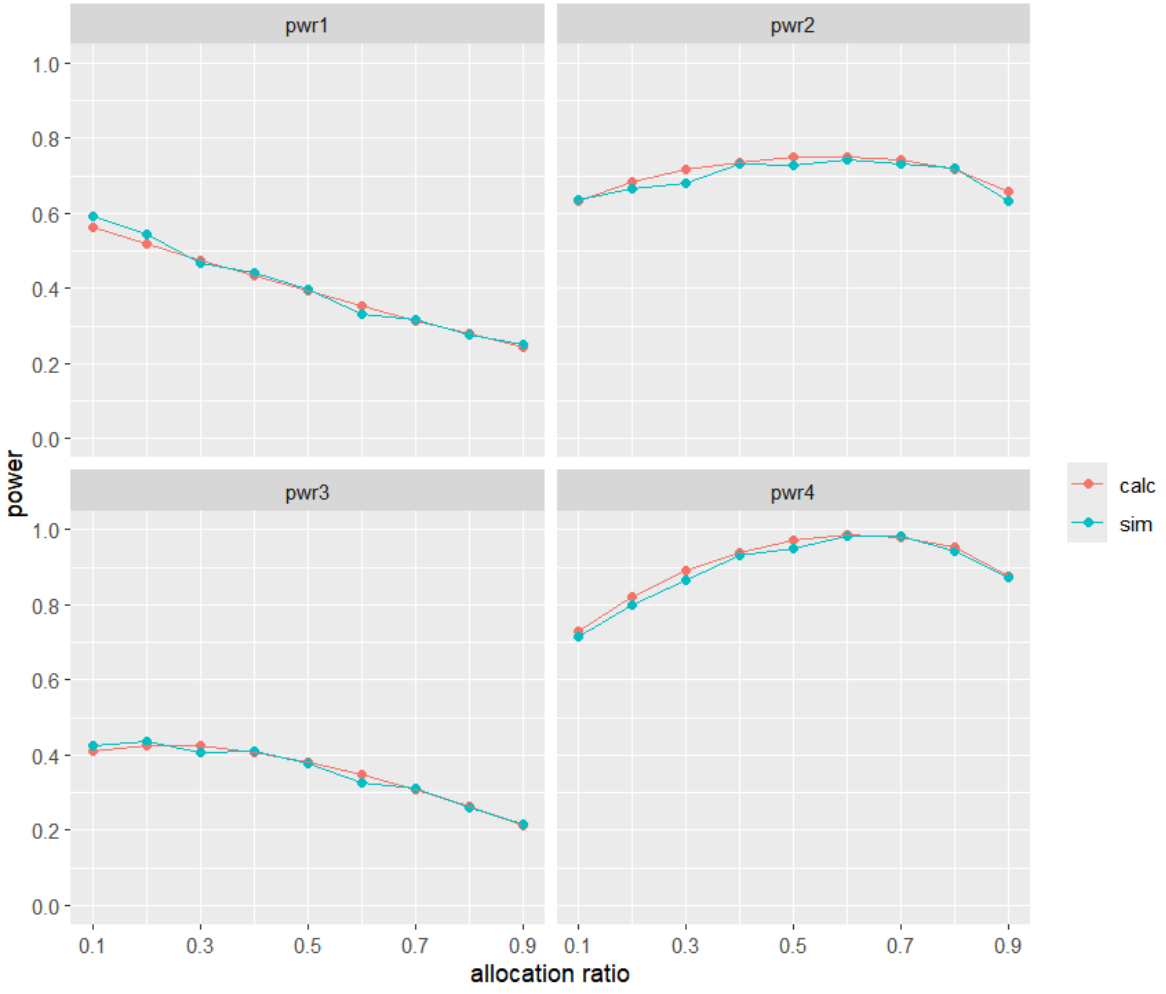
```
c1 <- getPwr_Count_Equi_JM1(delta_j = log(1.1), delta_nj = log(1.0), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = FALSE)
d1 <- getPwr_Count_Equi_JM1(delta_j = log(1.1), delta_nj = log(1.0), lambda0_j =
0.1, lambda0_nj = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, sim = TRUE)
pplot(c1, d1)
```



getPwr_Count_Super_JM2

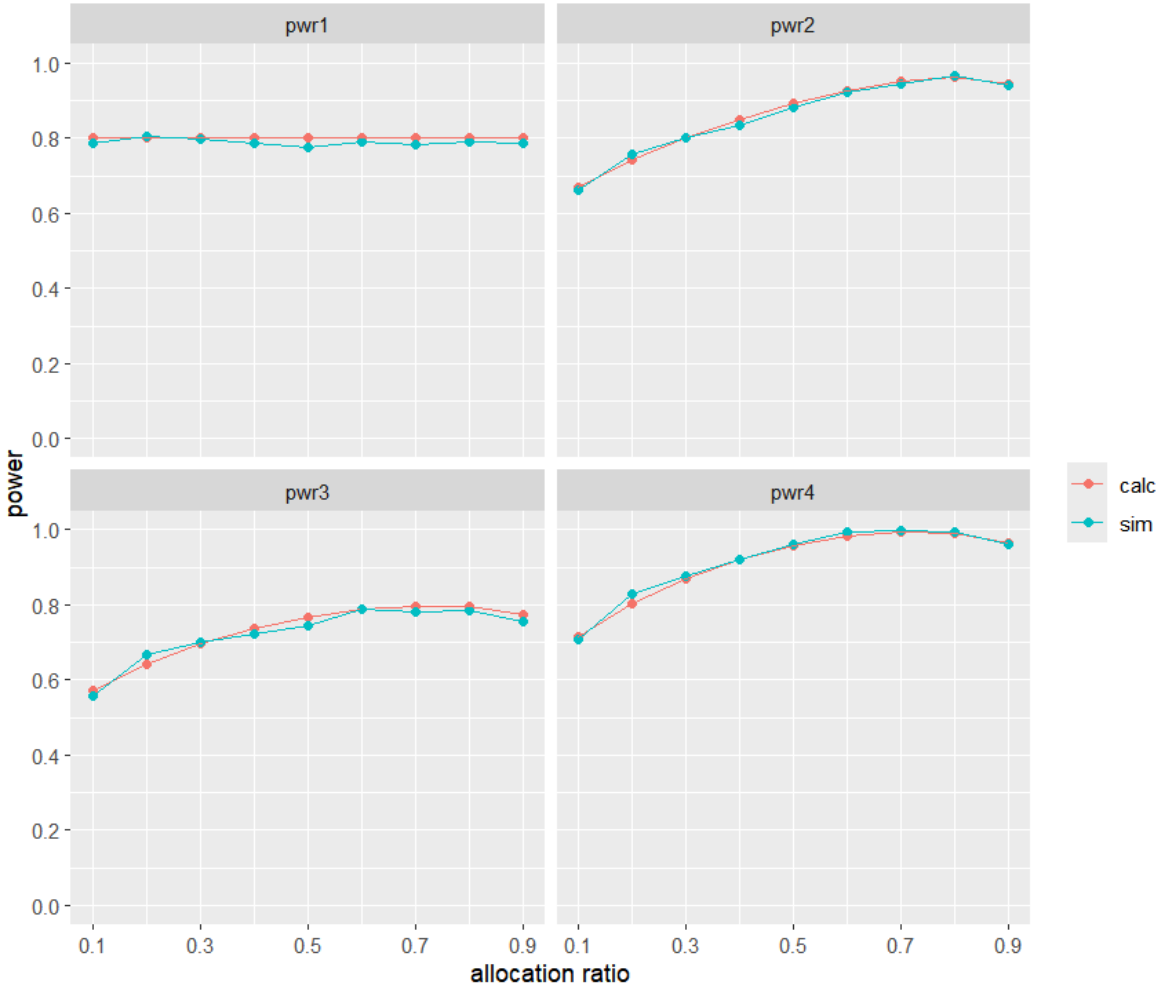
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Super_JM2(delta_i = c(log(1.2), log(1.4)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), alpha = 0.025, beta = NA, N = 300, r
= 1, sim = FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Super_JM2(delta_i = c(log(1.2), log(1.4)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), alpha = 0.025, beta = NA, N = 300, r
= 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

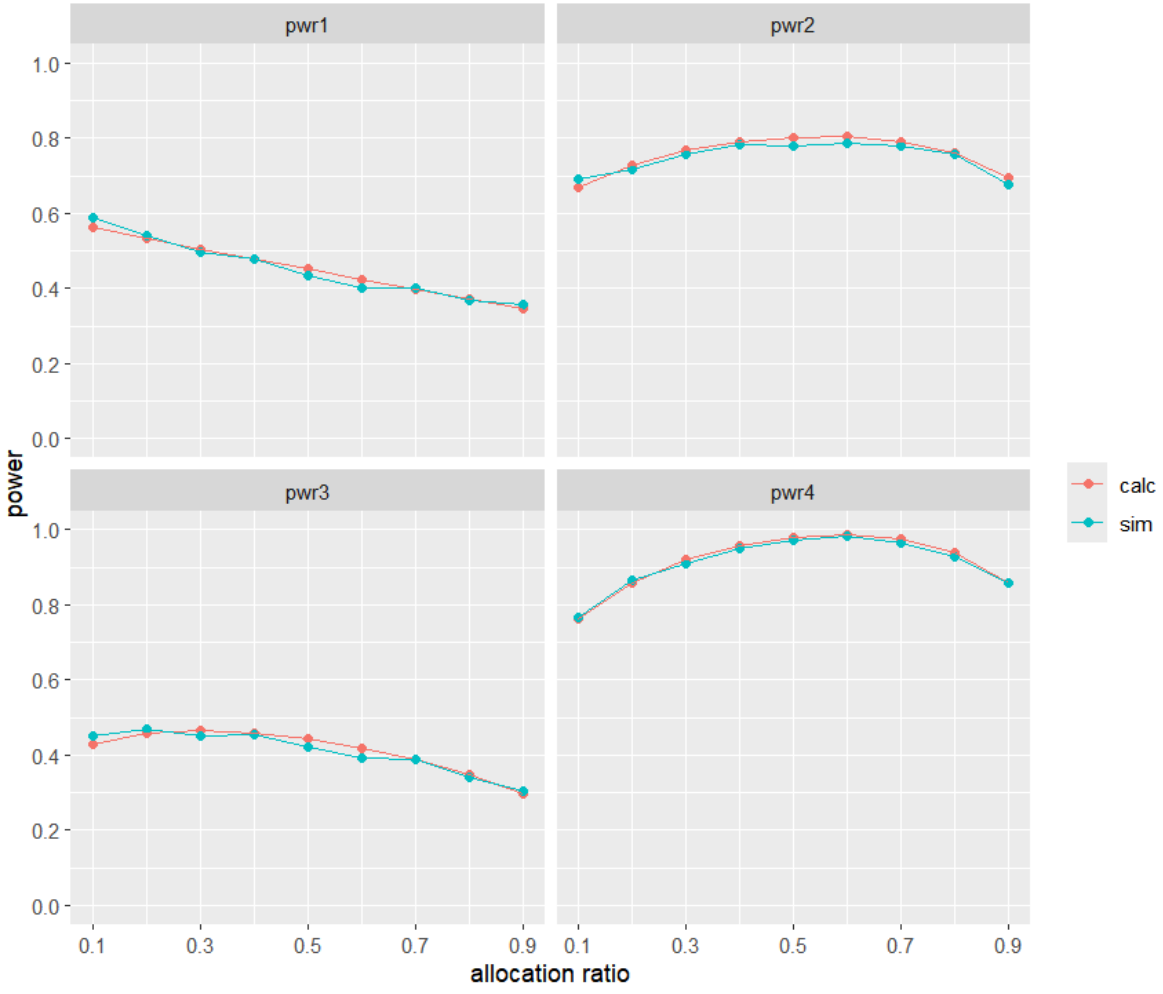
```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Super_JM2(delta_i = c(log(0.8), log(0.6)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), alpha = 0.025, beta = 0.2, N = NA, r
= 1, sim = FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Super_JM2(delta_i = c(log(0.8), log(0.6)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), alpha = 0.025, beta = 0.2, N = NA, r
= 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```



getPwr_Count_Noninf_JM2

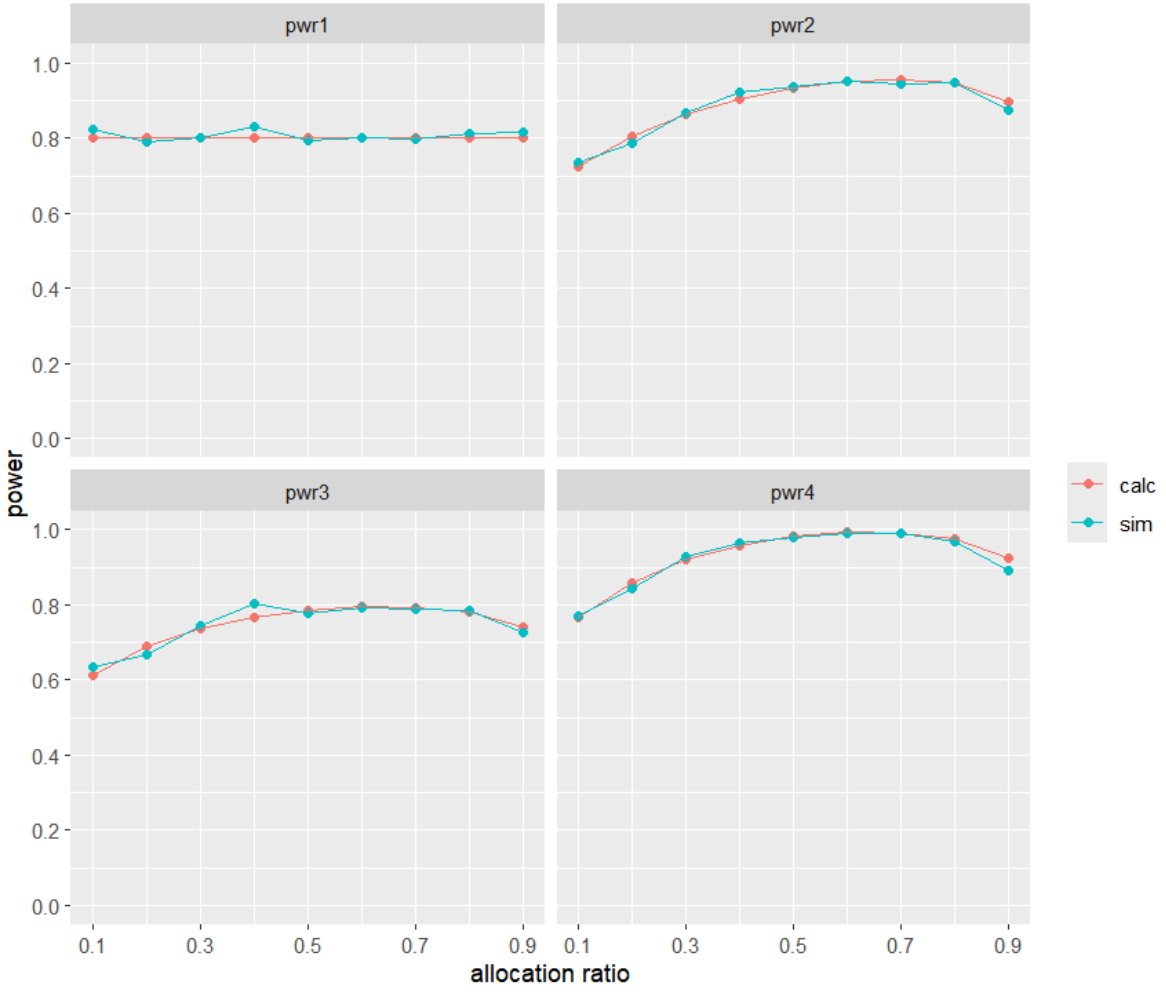
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = -log(0.7), alpha = 0.025, beta
= NA, N = 300, r = 1, direct = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Noninf_JM2(delta_i = c(log(0.9), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = -log(0.7), alpha = 0.025, beta
= NA, N = 300, r = 1, direct = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

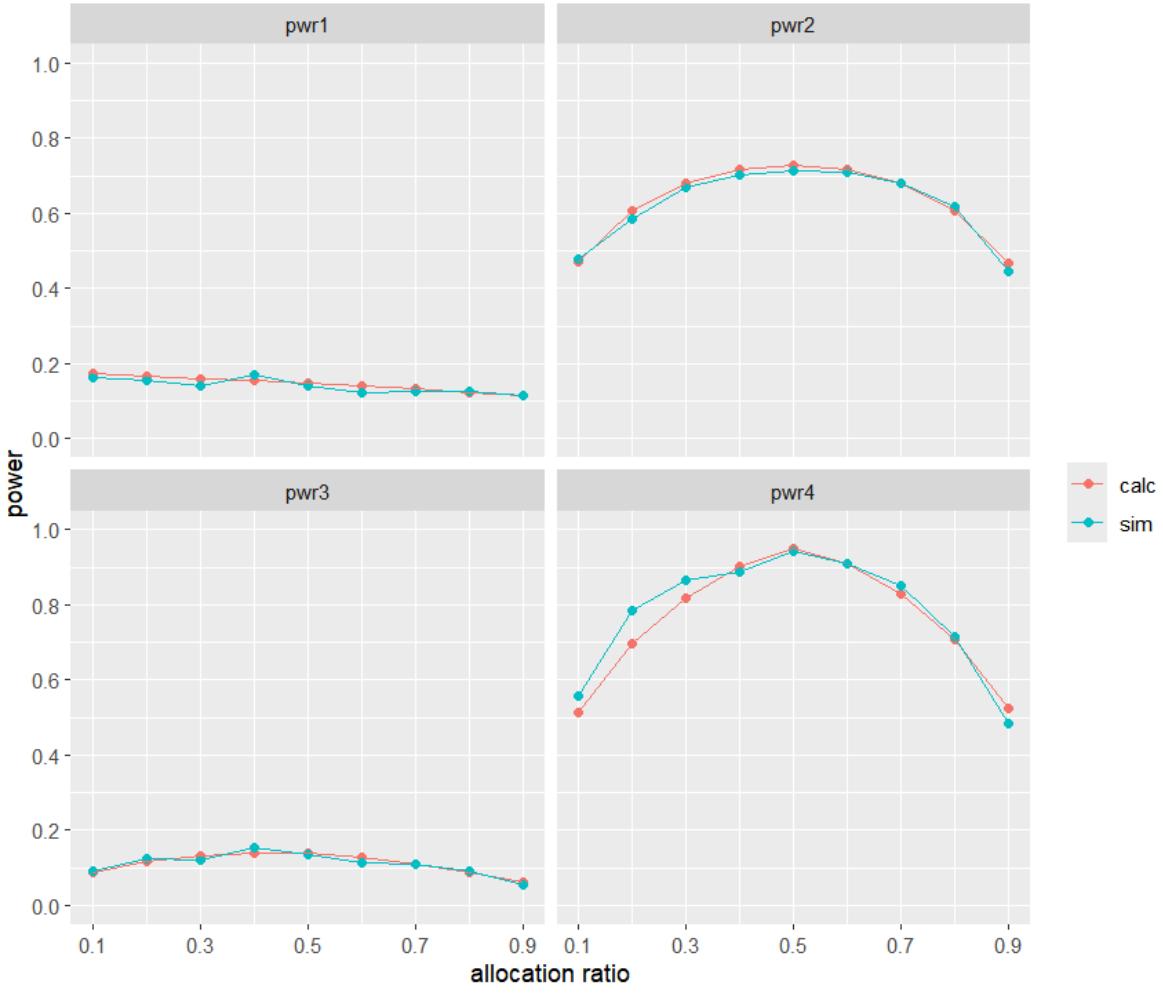
```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = log(1.3), alpha = 0.025, beta
= 0.2, N = NA, r = 1, direct = -1, sim = FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Noninf_JM2(delta_i = c(log(1.1), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = log(1.3), alpha = 0.025, beta
= 0.2, N = NA, r = 1, direct = -1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```



getPwr_Count_Equi_JM2

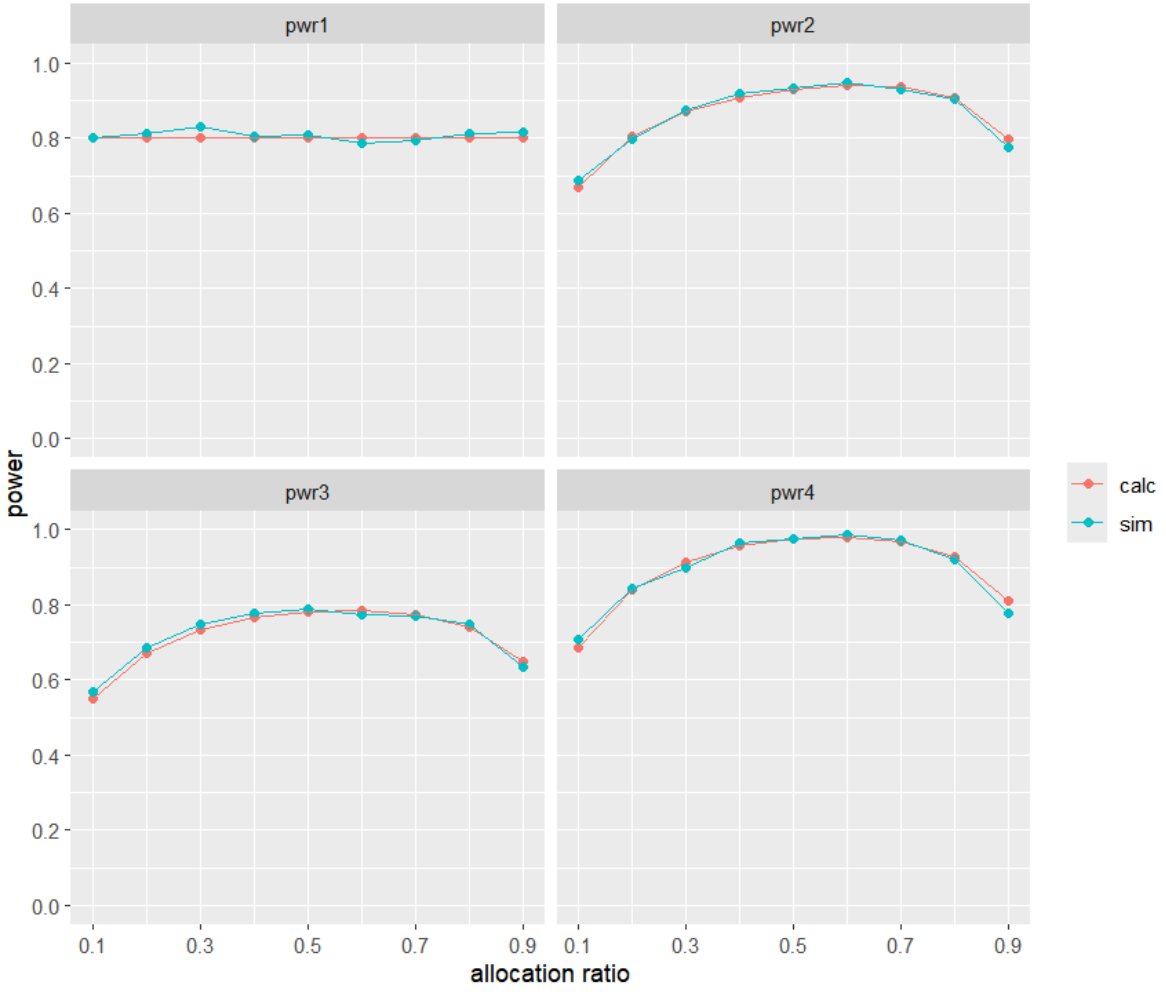
高优

```
f_set <- seq(0.1, 0.9, 0.1)
a <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Equi_JM2(delta_i = c(log(0.9), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = -log(0.7), alpha = 0.025, beta
= NA, N = 300, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
b <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Equi_JM2(delta_i = c(log(0.9), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = -log(0.7), alpha = 0.025, beta
= NA, N = 300, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(a, b)
```



低优

```
f_set <- seq(0.1, 0.9, 0.1)
c <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Equi_JM2(delta_i = c(log(1.1), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = log(1.3), alpha = 0.025, beta
= 0.2, N = NA, r = 1, sim = FALSE)$overall
  res$f <- f
  res
})
d <- map_dfr(.x = 1:length(f_set), .f = function(i) {
  f <- f_set[i]
  res <- getPwr_Count_Equi_JM2(delta_i = c(log(1.1), log(1.0)), lambda0_i =
c(0.1, 0.1), t = 5, k = 0, fi = c(f, 1 - f), cut = log(1.3), alpha = 0.025, beta
= 0.2, N = NA, r = 1, sim = TRUE)$overall
  res$f <- f
  res
})
pplot(c, d)
```



Regional sample size allocation using Japan's Method 1 or 2

Based on Japan's Method 1 or 2, given the global sample size and marginal probability (power) of efficacy consistency between target region and globally, calculate the required sample size allocated to the target region, in clinical trials using superiority, non-inferiority, and equivalence designs.

getPwr_Con_Super_JM1

```
v <- getPwr_Con_Super_JM1(delta_j = 0.5, delta_a = 0.7, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 100, r = 1, sim = FALSE)
getN_Con_Super_JM1(delta_a = 0.7, delta_j = 0.5, sigma = 1, pi = 0.5, beta1 = 1 -
v$pwr2, N = 100, r = 1)

v <- getPwr_Con_Super_JM1(delta_j = -0.5, delta_a = -0.7, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 100, r = 1, sim = FALSE)
getN_Con_Super_JM1(delta_a = -0.7, delta_j = -0.5, sigma = 1, pi = 0.5, beta1 = 1
- v$pwr2, N = 100, r = 1)
```

getPwr_Con_Noninf_JM1

```
v <- getPwr_Con_Noninf_JM1(delta_j = -0.5, delta_a = 0, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = 2, alpha = 0.025, beta = NA, N = 100, r = 1, direct =
1, sim = FALSE)
getN_Con_Noninf_JM1(delta_a = 0, delta_j = -0.5, sigma = 1, pi = 0.5, cut = 2,
beta1 = 1 - v$pwr2, N = 100, r = 1, direct = 1)

v <- getPwr_Con_Noninf_JM1(delta_j = 0.5, delta_a = 0, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = 2, alpha = 0.025, beta = NA, N = 100, r = 1, direct =
-1, sim = FALSE)
getN_Con_Noninf_JM1(delta_a = 0, delta_j = 0.5, sigma = 1, pi = 0.5, cut = 2,
beta1 = 1 - v$pwr2, N = 100, r = 1, direct = -1)
```

getPwr_Con_Equi_JM1

```
v <- getPwr_Con_Equi_JM1(delta_j = -0.5, delta_a = 0, sigma = 1, f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = 2, alpha = 0.025, beta = NA, N = 100, r = 1, sim =
FALSE)
getN_Con_Equi_JM1(delta_a = 0, delta_j = -0.5, sigma = 1, pi = 0.5, cut = 2,
beta1 = 1 - v$pwr2, N = 100, r = 1)

v <- getPwr_Con_Equi_JM1(delta_j = 0.5, delta_a = 0, sigma = 1, f = seq(0.1, 0.9,
0.1), pi = 0.5, cut = 2, alpha = 0.025, beta = NA, N = 100, r = 1, sim = FALSE)
getN_Con_Equi_JM1(delta_a = 0, delta_j = 0.5, sigma = 1, pi = 0.5, cut = 2, beta1
= 1 - v$pwr2, N = 100, r = 1)
```

getPwr_Bin_Super_JM1

```
v <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RD", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.75, p0_a = 0.5, p1_j = 0.7, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD")

v <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RR", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.75, p0_a = 0.5, p1_j = 0.7, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RR")

v <- getPwr_Bin_Super_JM1(p1_j = 0.7, p0_j = 0.5, p1_a = 0.75, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"OR", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.75, p0_a = 0.5, p1_j = 0.7, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "OR")

v <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_a = 0.25, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RD", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.25, p0_a = 0.5, p1_j = 0.3, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD")
```

```
v <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_a = 0.25, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"RR", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.25, p0_a = 0.5, p1_j = 0.3, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RR")

v <- getPwr_Bin_Super_JM1(p1_j = 0.3, p0_j = 0.5, p1_a = 0.25, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, scale =
"OR", sim = FALSE)
getN_Bin_Super_JM1(p1_a = 0.25, p0_a = 0.5, p1_j = 0.3, p0_j = 0.5, pi = 0.5,
alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "OR")
```

getPwr_Bin_Noninf_JM1

```
v <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 200, r =
1, scale = "RD", direct = 1, sim = FALSE)
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut
= 0.3, alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD",
direct = 1)

v <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "RR", direct = 1, sim = FALSE)
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut
= -log(0.6), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"RR", direct = 1)

v <- getPwr_Bin_Noninf_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", direct = 1, sim = FALSE)
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut
= -log(0.5), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"OR", direct = 1)

v <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 200, r =
1, scale = "RD", direct = -1, sim = FALSE)
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut
= 0.3, alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD",
direct = -1)

v <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.4), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "RR", direct = -1, sim = FALSE)
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut
= log(1.4), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"RR", direct = -1)

v <- getPwr_Bin_Noninf_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.8), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", direct = -1, sim = FALSE)
```

```
getN_Bin_Noninf_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut
= log(1.8), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"OR", direct = -1)
```

getPwr_Bin_Equi_JM1

```
v <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 200, r =
1, scale = "RD", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut =
0.3, alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD")

v <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.6), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "RR", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut =
-log(0.6), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"RR")

v <- getPwr_Bin_Equi_JM1(p1_j = 0.4, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = -log(0.4), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.4, p0_j = 0.5, pi = 0.5, cut =
-log(0.4), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"OR")

v <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = 0.3, alpha = 0.025, beta = NA, N = 200, r =
1, scale = "RD", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut =
0.3, alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale = "RD")

v <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(1.5), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "RR", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut =
log(1.5), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"RR")

v <- getPwr_Bin_Equi_JM1(p1_j = 0.6, p0_j = 0.5, p1_a = 0.5, p0_a = 0.5, f =
seq(0.1, 0.9, 0.1), pi = 0.5, cut = log(2.3), alpha = 0.025, beta = NA, N = 200,
r = 1, scale = "OR", sim = FALSE)
getN_Bin_Equi_JM1(p1_a = 0.5, p0_a = 0.5, p1_j = 0.6, p0_j = 0.5, pi = 0.5, cut =
log(2.3), alpha = NA, beta = NA, beta1 = 1 - v$pwr2, N = 200, r = 1, scale =
"OR")
```

getPwr_Surv_Super_JM1

```
v <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_a = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 1, sim
= FALSE)
getN_Surv_Super_JM1(delta_a = log(1.4), delta_j = log(1.3), pi = 0.5, beta1 = 1 -
v$pwr2, N = 200, r = 1, criterion = 1)

v <- getPwr_Surv_Super_JM1(delta_j = log(1.3), delta_a = log(1.4), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 2, sim
= FALSE)
getN_Surv_Super_JM1(delta_a = log(1.4), delta_j = log(1.3), pi = 0.5, beta1 = 1 -
v$pwr2, N = 200, r = 1, criterion = 2)

v <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 1, sim
= FALSE)
getN_Surv_Super_JM1(delta_a = log(0.7), delta_j = log(0.8), pi = 0.5, beta1 = 1 -
v$pwr2, N = 200, r = 1, criterion = 1)

v <- getPwr_Surv_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), f = seq(0.1,
0.9, 0.1), pi = 0.5, alpha = 0.025, beta = NA, N = 200, r = 1, criterion = 2, sim
= FALSE)
getN_Surv_Super_JM1(delta_a = log(0.7), delta_j = log(0.8), pi = 0.5, beta1 = 1 -
v$pwr2, N = 200, r = 1, criterion = 2)
```

getPwr_Surv_Noninf_JM1

```
v <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = -log(0.7), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 1, direct = 1, sim = FALSE)
getN_Surv_Noninf_JM1(delta_a = log(1.0), delta_j = log(0.9), pi = 0.5, cut = -
log(0.7), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 1, direct = 1)

v <- getPwr_Surv_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = -log(0.7), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 2, direct = 1, sim = FALSE)
getN_Surv_Noninf_JM1(delta_a = log(1.0), delta_j = log(0.9), pi = 0.5, cut = -
log(0.7), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 2, direct = 1)

v <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = log(1.3), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 1, direct = -1, sim = FALSE)
getN_Surv_Noninf_JM1(delta_a = log(1.0), delta_j = log(1.1), pi = 0.5, cut =
log(1.3), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 1, direct = -1)

v <- getPwr_Surv_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = log(1.3), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 2, direct = -1, sim = FALSE)
getN_Surv_Noninf_JM1(delta_a = log(1.0), delta_j = log(1.1), pi = 0.5, cut =
log(1.3), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 2, direct = -1)
```

getPwr_Surv_Equi_JM1

```
v <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = -log(0.7), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 1, sim = FALSE)
getN_Surv_Equi_JM1(delta_a = log(1.0), delta_j = log(0.9), pi = 0.5, cut = -
log(0.7), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 1)

v <- getPwr_Surv_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = -log(0.7), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 2, sim = FALSE)
getN_Surv_Equi_JM1(delta_a = log(1.0), delta_j = log(0.9), pi = 0.5, cut = -
log(0.7), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 2)

v <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = log(1.3), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 1, sim = FALSE)
getN_Surv_Equi_JM1(delta_a = log(1.0), delta_j = log(1.1), pi = 0.5, cut =
log(1.3), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 1)

v <- getPwr_Surv_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), f = seq(0.1,
0.9, 0.1), pi = 0.5, cut = log(1.3), alpha = 0.025, beta = NA, N = 200, r = 1,
criterion = 2, sim = FALSE)
getN_Surv_Equi_JM1(delta_a = log(1.0), delta_j = log(1.1), pi = 0.5, cut =
log(1.3), beta1 = 1 - v$pwr2, N = 200, r = 1, criterion = 2)
```

getPwr_Count_Super_JM1

```
v <- getPwr_Count_Super_JM1(delta_j = log(1.2), delta_a = log(1.3), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = FALSE)
getN_Count_Super_JM1(delta_a = log(1.3), delta_j = log(1.2), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, beta1 = 1 - v$pwr2, N = 300, r = 1)

v <- getPwr_Count_Super_JM1(delta_j = log(0.8), delta_a = log(0.7), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, alpha =
0.025, beta = NA, N = 300, r = 1, sim = FALSE)
getN_Count_Super_JM1(delta_a = log(0.7), delta_j = log(0.8), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, beta1 = 1 - v$pwr2, N = 300, r = 1)
```

getPwr_Count_Noninf_JM1

```
v <- getPwr_Count_Noninf_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, direct = 1, sim = FALSE)
getN_Count_Noninf_JM1(delta_a = log(1.0), delta_j = log(0.9), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, cut = -log(0.7), beta1 = 1 - v$pwr2, N =
300, r = 1, direct = 1)

v <- getPwr_Count_Noninf_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = NA, N = 300, r = 1, direct = -1, sim = FALSE)
getN_Count_Noninf_JM1(delta_a = log(1.0), delta_j = log(1.1), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, cut = log(1.3), beta1 = 1 - v$pwr2, N =
300, r = 1, direct = -1)
```

getPwr_Count_Equi_JM1

```
v <- getPwr_Count_Equi_JM1(delta_j = log(0.9), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut = -
log(0.7), alpha = 0.025, beta = NA, N = 300, r = 1, sim = FALSE)
getN_Count_Equi_JM1(delta_a = log(1.0), delta_j = log(0.9), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, cut = -log(0.7), beta1 = 1 - v$pwr2, N =
300, r = 1)

v <- getPwr_Count_Equi_JM1(delta_j = log(1.1), delta_a = log(1.0), lambda0_j =
0.1, lambda0_a = 0.1, t = 5, k = 0, f = seq(0.1, 0.9, 0.1), pi = 0.5, cut =
log(1.3), alpha = 0.025, beta = NA, N = 300, r = 1, sim = FALSE)
getN_Count_Equi_JM1(delta_a = log(1.0), delta_j = log(1.1), lambda0_a = 0.1,
lambda0_j = 0.1, t = 5, k = 0, pi = 0.5, cut = log(1.3), beta1 = 1 - v$pwr2, N =
300, r = 1)
```

Sample size and power

Calculating sample size when given power or power when given sample size.

getN_Con_Super

```
(v <- getN_Con_Super(delta = seq(0.5, 1.5, 0.5), sigma = 4, alpha = 0.025, beta =
0.2, N = NA, r = 1))
getN_Con_Super(delta = seq(0.5, 1.5, 0.5), sigma = 4, alpha = 0.025, beta = NA, N
= 300, r = 1)
```

```
> (v <- getN_Con_Super(delta = seq(0.5, 1.5, 0.5), sigma = 4, alpha = 0.025, beta = 0.2, N = NA, r = 1))
  delta sigma alpha beta       pwr r    N   n1   n0
1   0.5     4 0.025  0.2 0.8001340 1 2010 1005 1005
2   1.0     4 0.025  0.2 0.8013015 1  504  252  252
3   1.5     4 0.025  0.2 0.8013015 1  224  112  112
> getN_Con_Super(delta = seq(0.5, 1.5, 0.5), sigma = 4, alpha = 0.025, beta = NA, N = 300, r = 1)
  delta sigma alpha beta       pwr r   N  n1  n0
1   0.5     4 0.025   NA 0.1901260 1 300 150 150
2   1.0     4 0.025   NA 0.5812528 1 300 150 150
3   1.5     4 0.025   NA 0.9010628 1 300 150 150
```

Two-Sample Z-Tests Assuming Equal Variance

Numeric Results for Two-Sample Z-Test Assuming Equal Variance
Alternative Hypothesis: H1: δ = μ1 - μ2 ≠ 0

Target Power	Actual Power	N1	N2	N	μ1	μ2	δ	σ	Alpha
0.8	0.80013	1005	1005	2010	0.5	0	0.5	4	0.05
0.8	0.80130	252	252	504	1.0	0	1.0	4	0.05
0.8	0.80130	112	112	224	1.5	0	1.5	4	0.05

Two-Sample Z-Tests Assuming Equal Variance

Numeric Results for Two-Sample Z-Test Assuming Equal Variance
Alternative Hypothesis: H1: δ = μ1 - μ2 ≠ 0

Power	N1	N2	N	μ1	μ2	δ	σ	Alpha
0.19130	150	150	300	0.5	0	0.5	4	0.05
0.58127	150	150	300	1.0	0	1.0	4	0.05
0.90106	150	150	300	1.5	0	1.5	4	0.05

getN_Con_Noninf

```
(v <- getN_Con_Noninf(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1))
getN_Con_Noninf(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1)
```

```
> (v <- getN_Con_Noninf(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1))
  delta sigma cut alpha beta       pwr r    N   n1   n0 direct
1   0.0     4   2 0.025  0.2 0.8013015 1  126   63   63      1
2  -0.5     4   2 0.025  0.2 0.8013015 1  224  112  112      1
3  -1.0     4   2 0.025  0.2 0.8013015 1  504  252  252      1
4  -1.5     4   2 0.025  0.2 0.8001340 1 2010 1005 1005      1
> getN_Con_Noninf(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1)
  delta sigma cut alpha beta       pwr r   N  n1  n0 direct
1   0.0     4   2 0.025   NA 0.9424375 1 200 100 100      1
2  -0.5     4   2 0.025   NA 0.7554329 1 200 100 100      1
3  -1.0     4   2 0.025   NA 0.4237939 1 200 100 100      1
4  -1.5     4   2 0.025   NA 0.1409456 1 200 100 100      1
```

Two-Sample T-Tests for Non-Inferiority Assuming Equal Variance

Numeric Results for an Equal-Variance T-Test
δ = μ1 - μ2 = μT - μR
Higher Means are Better
Hypotheses: H0: δ ≤ -NIM vs. H1: δ > -NIM

Target Power	Actual Power	N1	N2	N	-NIM	δ	σ	Alpha
0.8	0.80009	1005	1005	2010	-2	-1.5	4	0.025
0.8	0.80136	253	253	506	-2	-1.0	4	0.025
0.8	0.80141	113	113	226	-2	-0.5	4	0.025
0.8	0.80146	64	64	128	-2	0.0	4	0.025

Two-Sample T-Tests for Non-Inferiority Assuming Equal Variance

Numeric Results for an Equal-Variance T-Test
δ = μ1 - μ2 = μT - μR
Higher Means are Better
Hypotheses: H0: δ ≤ -NIM vs. H1: δ > -NIM

Power	N1	N2	N	-NIM	δ	σ	Alpha
0.13999	100	100	200	-2	-1.5	4	0.025
0.42044	100	100	200	-2	-1.0	4	0.025
0.75137	100	100	200	-2	-0.5	4	0.025
0.94043	100	100	200	-2	0.0	4	0.025

getN_Con_Equi

```
(v <- getN_Con_Equi(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1))
getN_Con_Equi(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1)
```

```
> (v <- getN_Con_Equi(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = 0.2, N = NA, r = 1))
  delta sigma cut alpha beta       pwr r    N   n1   n0
1   0.0     4   2 0.025  0.2 0.8062747 1  170   85   85
2  -0.5     4   2 0.025  0.2 0.8016775 1  226  113  113
3  -1.0     4   2 0.025  0.2 0.8013015 1  504  252  252
4  -1.5     4   2 0.025  0.2 0.8001340 1 2010 1005 1005
> getN_Con_Equi(delta = seq(0, -1.5, -0.5), sigma = 4, cut = 2, alpha = 0.025, beta = NA, N = 200, r = 1)
  delta sigma cut alpha beta       pwr r   N  n1  n0
1   0.0     4   2 0.025   NA 0.8848750 1 200 100 100
2  -0.5     4   2 0.025   NA 0.7484754 1 200 100 100
3  -1.0     4   2 0.025   NA 0.4233800 1 200 100 100
4  -1.5     4   2 0.025   NA 0.1409338 1 200 100 100
```

Two-Sample T-Tests for Equivalence Assuming Equal Variance

Numeric Results for Two One-Sided Equal-Variance T-Tests

δ = μ1 - μ2 = μT - μR
Hypotheses: H0: δ ≤ EL or δ ≥ EU vs. H1: EL < δ < EU

Target Power	Actual Power	N1	N2	N	Lower Equiv Limit EL	Upper Equiv Limit EU	δ	σ	Alpha
0.8	0.80009	1005	1005	2010	-2	2	-1.5	4	0.025
0.8	0.80136	253	253	506	-2	2	-1.0	4	0.025
0.8	0.80179	114	114	228	-2	2	-0.5	4	0.025
0.8	0.80646	86	86	172	-2	2	0.0	4	0.025

Two-Sample T-Tests for Equivalence Assuming Equal Variance

Numeric Results for Two One-Sided Equal-Variance T-Tests

δ = μ1 - μ2 = μT - μR
Hypotheses: H0: δ ≤ EL or δ ≥ EU vs. H1: EL < δ < EU

Power	N1	N2	N	Lower Equiv Limit EL	Upper Equiv Limit EU	δ	σ	Alpha
0.13998	100	100	200	-2	2	-1.5	4	0.025
0.41998	100	100	200	-2	2	-1.0	4	0.025
0.74398	100	100	200	-2	2	-0.5	4	0.025
0.88085	100	100	200	-2	2	0.0	4	0.025

getN_Bin_Super

RD

```
(v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD"))
getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N = 300, r = 1, scale = "RD")
```

```
> (v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD"))
    p1  p0 scale delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    RD  0.10 0.025  0.2 0.8004125 1 770 385 385
2 0.55 0.4    RD  0.15 0.025  0.2 0.8021598 1 342 171 171
3 0.60 0.4    RD  0.20 0.025  0.2 0.8033625 1 190  95  95
> getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N = 300, r = 1, scale = "RD")
    p1  p0 scale delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    RD  0.10 0.025   NA 0.4167057 1 300 150 150
2 0.55 0.4    RD  0.15 0.025   NA 0.7489566 1 300 150 150
3 0.60 0.4    RD  0.20 0.025   NA 0.9424375 1 300 150 150
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	δ1	Variance
1		Des1	Superiority	1	1-Sided	0.025	0.8	1	770	0.4	0.5	0.1	Unpooled Estimate
2		Des2	Superiority	1	1-Sided	0.025	0.801	1	341	0.4	0.55	0.15	Unpooled Estimate
3		Des3	Superiority	1	1-Sided	0.025	0.801	1	189	0.4	0.6	0.2	Unpooled Estimate

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	δ1	Variance
1		Des1	Superiority	1	1-Sided	0.025	0.417	1	300	0.4	0.5	0.1	Unpooled Estimate
2		Des2	Superiority	1	1-Sided	0.025	0.749	1	300	0.4	0.55	0.15	Unpooled Estimate
3		Des3	Superiority	1	1-Sided	0.025	0.942	1	300	0.4	0.6	0.2	Unpooled Estimate

RR

```
(v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta =
0.2, N = NA, r = 1, scale = "RR"))
getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N =
300, r = 1, scale = "RR")
```

```
> (v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR"))
    p1  p0 scale     delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    RR 0.2231436 0.025  0.2 0.8009186 1 790 395 395
2 0.55 0.4    RR 0.3184537 0.025  0.2 0.8012718 1 360 180 180
3 0.60 0.4    RR 0.4054651 0.025  0.2 0.8021096 1 208 104 104
> getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N = 300, r = 1, scale = "RR")
    p1  p0 scale     delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    RR 0.2231436 0.025   NA 0.4084626 1 300 150 150
2 0.55 0.4    RR 0.3184537 0.025   NA 0.7263053 1 300 150 150
3 0.60 0.4    RR 0.4054651 0.025   NA 0.9212764 1 300 150 150
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Variance	ρ1
4		Des4	Superiority	1	1-Sided	0.025	0.8	1	789	0.4	0.5	Unpooled Estimate	1.25
5		Des5	Superiority	1	1-Sided	0.025	0.8	1	359	0.4	0.55	Unpooled Estimate	1.375
6		Des6	Superiority	1	1-Sided	0.025	0.8	1	207	0.4	0.6	Unpooled Estimate	1.5

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Variance	ρ1
7		Des7	Superiority	1	1-Sided	0.025	0.408	1	300	0.4	0.5	Unpooled Estimate	1.25
8		Des8	Superiority	1	1-Sided	0.025	0.726	1	300	0.4	0.55	Unpooled Estimate	1.375
9		Des9	Superiority	1	1-Sided	0.025	0.921	1	300	0.4	0.6	Unpooled Estimate	1.5

OR

```
(v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta =
0.2, N = NA, r = 1, scale = "OR"))
getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N =
300, r = 1, scale = "OR")
```

```
> (v <- getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR"))
    p1  p0 scale     delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    OR 0.4054651 0.025  0.2 0.8001075 1 780 390 390
2 0.55 0.4    OR 0.6061358 0.025  0.2 0.8014941 1 352 176 176
3 0.60 0.4    OR 0.8109302 0.025  0.2 0.8021096 1 200 100 100
> getN_Bin_Super(p0 = 0.4, p1 = seq(0.5, 0.6, 0.05), alpha = 0.025, beta = NA, N = 300, r = 1, scale = "OR")
    p1  p0 scale     delta alpha beta       pwr r   N  n1  n0
1 0.50 0.4    OR 0.4054651 0.025   NA 0.4120572 1 300 150 150
2 0.55 0.4    OR 0.6061358 0.025   NA 0.7360970 1 300 150 150
3 0.60 0.4    OR 0.8109302 0.025   NA 0.9306329 1 300 150 150
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	τc	Prop. Treatment (Alt.)	ψ1
10		Des10	Superiority	1	1-Sided	0.025	0.8	1	780	0.4	0.5	1.5
11		Des11	Superiority	1	1-Sided	0.025	0.8	1	351	0.4	0.55	1.833
12		Des12	Superiority	1	1-Sided	0.025	0.8	1	199	0.4	0.6	2.25

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	τc	Prop. Treatment (Alt.)	ψ1
13		Des13	Superiority	1	1-Sided	0.025	0.412	1	300	0.4	0.5	1.5
14		Des14	Superiority	1	1-Sided	0.025	0.736	1	300	0.4	0.55	1.833
15		Des15	Superiority	1	1-Sided	0.025	0.931	1	300	0.4	0.6	2.25

getN_Bin_Noninf

RD

```
(v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", direct = 1))
getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = NA, N = 200, r = 1, scale = "RD", direct = 1)
```

```
> (v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD", direct = 1))
    p1  p0 cut scale delta alpha beta       pwr r   N  n1  n0 direct
1 0.40 0.5 0.2    RD -0.10 0.025  0.2 0.8004125 1 770 385 385      1
2 0.45 0.5 0.2    RD -0.05 0.025  0.2 0.8010204 1 348 174 174      1
3 0.50 0.5 0.2    RD  0.00 0.025  0.2 0.8035266 1 198  99  99      1
> getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = NA, N = 200, r = 1, scale = "RD", direct = 1)
    p1  p0 cut scale delta alpha beta       pwr r   N  n1  n0 direct
1 0.40 0.5 0.2    RD -0.10 0.025   NA 0.2975734 1 200 100 100      1
2 0.45 0.5 0.2    RD -0.05 0.025   NA 0.5661889 1 200 100 100      1
3 0.50 0.5 0.2    RD  0.00 0.025   NA 0.8074296 1 200 100 100      1
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	τc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	δ1	δ0
28		Des28	Noninferiority	1	1-Sided	0.025	0.8	1	770	0.5	0.4	0.3	-0.1	-0.2
29		Des29	Noninferiority	1	1-Sided	0.025	0.801	1	348	0.5	0.45	0.3	-0.05	-0.2
30		Des30	Noninferiority	1	1-Sided	0.025	0.802	1	197	0.5	0.5	0.3	0	-0.2

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	τc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	δ1	δ0
31		Des31	Noninferiority	1	1-Sided	0.025	0.298	1	200	0.5	0.4	0.3	-0.1	-0.2
32		Des32	Noninferiority	1	1-Sided	0.025	0.566	1	200	0.5	0.45	0.3	-0.05	-0.2
33		Des33	Noninferiority	1	1-Sided	0.025	0.807	1	200	0.5	0.5	0.3	0	-0.2

RR

```
(v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha
= 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", direct = 1))
getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha =
0.025, beta = NA, N = 200, r = 1, scale = "RR", direct = 1)
```

```
> (v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR", direct = 1))
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0 direct
1 0.40 0.5 0.5108256    RR -0.2231436 0.025  0.2 0.8014924 1 476 238 238      1
2 0.45 0.5 0.5108256    RR -0.1053605 0.025  0.2 0.8033274 1 214 107 107      1
3 0.50 0.5 0.5108256    RR  0.0000000 0.025  0.2 0.8054258 1 122  61  61      1
> getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "RR", direct = 1)
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0 direct
1 0.40 0.5 0.5108256    RR -0.2231436 0.025   NA 0.4441314 1 200 100 100      1
2 0.45 0.5 0.5108256    RR -0.1053605 0.025   NA 0.7763663 1 200 100 100      1
3 0.50 0.5 0.5108256    RR  0.0000000 0.025   NA 0.9507448 1 200 100 100      1
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	ρ1	ρ0	Test Statistic
34		Des34	Noninferiority	1	1-Sided	0.025	0.801	1	475	0.5	0.4	0.3	0.8	0.6	Wald
35		Des35	Noninferiority	1	1-Sided	0.025	0.801	1	213	0.5	0.45	0.3	0.9	0.6	Wald
36		Des36	Noninferiority	1	1-Sided	0.025	0.802	1	121	0.5	0.5	0.3	1	0.6	Wald

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	ρ1	ρ0	Test Statistic
37		Des37	Noninferiority	1	1-Sided	0.025	0.444	1	200	0.5	0.4	0.3	0.8	0.6	Wald
38		Des38	Noninferiority	1	1-Sided	0.025	0.776	1	200	0.5	0.45	0.3	0.9	0.6	Wald
39		Des39	Noninferiority	1	1-Sided	0.025	0.951	1	200	0.5	0.5	0.3	1	0.6	Wald

OR

```
(v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha
= 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", direct = 1))
getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha =
0.025, beta = NA, N = 200, r = 1, scale = "OR", direct = 1)
```

```
> (v <- getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR", direct = 1))
   p1  p0       cut scale      delta alpha beta       pwr r    N  n1  n0 direct
1 0.40 0.5 0.6931472    OR -0.4054651 0.025  0.2 0.8002481 1 1550 775 775      1
2 0.45 0.5 0.6931472    OR -0.2006707 0.025  0.2 0.8011961 1  522 261 261      1
3 0.50 0.5 0.6931472    OR  0.0000000 0.025  0.2 0.8009244 1  262 131 131      1
> getN_Bin_Noninf(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "OR", direct = 1)
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0 direct
1 0.40 0.5 0.6931472    OR -0.4054651 0.025   NA 0.1702225 1 200 100 100      1
2 0.45 0.5 0.6931472    OR -0.2006707 0.025   NA 0.4116989 1 200 100 100      1
3 0.50 0.5 0.6931472    OR  0.0000000 0.025   NA 0.6881741 1 200 100 100      1
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	ψ1	ψ0
40		Des40	Noninferiority	1	1-Sided	0.025	0.8	1	1550	0.5	0.4	0.333	0.667	0.5
41		Des41	Noninferiority	1	1-Sided	0.025	0.8	1	521	0.5	0.45	0.333	0.818	0.5
42		Des42	Noninferiority	1	1-Sided	0.025	0.801	1	262	0.5	0.5	0.333	1	0.5

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	Prop. Treatment (Null)	ψ1	ψ0
46		Des46	Noninferiority	1	1-Sided	0.025	0.17	1	200	0.5	0.4	0.333	0.667	0.5
47		Des47	Noninferiority	1	1-Sided	0.025	0.412	1	200	0.5	0.45	0.333	0.818	0.5
48		Des48	Noninferiority	1	1-Sided	0.025	0.688	1	200	0.5	0.5	0.333	1	0.5

getN_Bin_Equi

RD

```
(v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025,
beta = 0.2, N = NA, r = 1, scale = "RD"))
getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta
= NA, N = 200, r = 1, scale = "RD")
```

```
> (v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RD"))
   p1  p0 cut scale delta alpha beta       pwr r   N  n1  n0
1 0.40 0.5 0.2    RD -0.10 0.025  0.2 0.8004125 1 770 385 385
2 0.45 0.5 0.2    RD -0.05 0.025  0.2 0.8000815 1 350 175 175
3 0.50 0.5 0.2    RD  0.00 0.025  0.2 0.8028283 1 264 132 132
> getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = 0.2, alpha = 0.025, beta = NA, N = 200, r = 1, scale = "RD")
   p1  p0 cut scale delta alpha beta       pwr r   N  n1  n0
1 0.40 0.5 0.2    RD -0.10 0.025   NA 0.2875575 1 200 100 100
2 0.45 0.5 0.2    RD -0.05 0.025   NA 0.5096424 1 200 100 100
3 0.50 0.5 0.2    RD  0.00 0.025   NA 0.6148592 1 200 100 100
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	δ1	Equivalence Margin	Expected Difference
49		Des49	Equivalence	1	2-Sided	0.025	0.8	1	754	0.5	0.4	-0.1	0.2	-0.1
50		Des50	Equivalence	1	2-Sided	0.025	0.8	1	341	0.5	0.45	-0.05	0.2	-0.05
51		Des51	Equivalence	1	2-Sided	0.025	0.802	1	257	0.5	0.5	0	0.2	0

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	δ1	Equivalence Margin	Expected Difference
52		Des52	Equivalence	1	2-Sided	0.025	0.298	1	200	0.5	0.4	-0.1	0.2	-0.1
53		Des53	Equivalence	1	2-Sided	0.025	0.528	1	200	0.5	0.45	-0.05	0.2	-0.05
54		Des54	Equivalence	1	2-Sided	0.025	0.636	1	200	0.5	0.5	0	0.2	0

Equivalence Tests for the Difference Between Two Proportions

Numeric Results

Test Statistic: Z-Test with Unpooled Variance
Hypotheses: H0: P1 - P2 ≤ D0.L or P1 - P2 ≥ D0.U vs. H1: D0.L < P1 - P2 < D0.U

Target Power	Actual Power*	N1	N2	N	Ref. P2	P1.0L	P1.0U	D0.L	D0.U	D1	Alpha
0.8	0.80041	385	385	770	0.5	0.3	0.7	-0.2	0.2	-0.10	0.025
0.8	0.80008	175	175	350	0.5	0.3	0.7	-0.2	0.2	-0.05	0.025
0.8	0.80283	132	132	264	0.5	0.3	0.7	-0.2	0.2	0.00	0.025

* Power was computed using the normal approximation method.

Equivalence Tests for the Difference Between Two Proportions

Numeric Results

Test Statistic: Z-Test with Unpooled Variance
Hypotheses: H0: P1 - P2 ≤ D0.L or P1 - P2 ≥ D0.U vs. H1: D0.L < P1 - P2 < D0.U

Power*	N1	N2	N	Ref. P2	P1.0L	P1.0U	D0.L	D0.U	D1	Alpha
0.28756	100	100	200	0.5	0.3	0.7	-0.2	0.2	-0.10	0.025
0.50964	100	100	200	0.5	0.3	0.7	-0.2	0.2	-0.05	0.025
0.61486	100	100	200	0.5	0.3	0.7	-0.2	0.2	0.00	0.025

* Power was computed using the normal approximation method.

RR

```
(v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha =
0.025, beta = 0.2, N = NA, r = 1, scale = "RR"))
getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha = 0.025,
beta = NA, N = 200, r = 1, scale = "RR")
```

```
> (v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "RR"))
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0
1 0.40 0.5 0.5108256    RR -0.2231436 0.025  0.2 0.8014923 1 476 238 238
2 0.45 0.5 0.5108256    RR -0.1053605 0.025  0.2 0.8012518 1 218 109 109
3 0.50 0.5 0.5108256    RR  0.0000000 0.025  0.2 0.8032652 1 162  81  81
> getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.6), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "RR")
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0
1 0.40 0.5 0.5108256    RR -0.2231436 0.025   NA 0.4404729 1 200 100 100
2 0.45 0.5 0.5108256    RR -0.1053605 0.025   NA 0.7614964 1 200 100 100
3 0.50 0.5 0.5108256    RR  0.0000000 0.025   NA 0.9014896 1 200 100 100
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	ρ1	Equivalence Margin	Test Statistic
55		Des55	Equivalence	1	2-Sided	0.025	0.803	1	162	0.5	0.5	1	1.667	Wald

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	πc	Prop. Treatment (Alt.)	ρ1	Equivalence Margin	Test Statistic
56		Des56	Equivalence	1	2-Sided	0.025	0.901	1	200	0.5	0.5	1	1.667	Wald

OR

```
(v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha =
0.025, beta = 0.2, N = NA, r = 1, scale = "OR"))
getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha = 0.025,
beta = NA, N = 200, r = 1, scale = "OR")
```

```
> (v <- getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha = 0.025, beta = 0.2, N = NA, r = 1, scale = "OR"))
   p1  p0       cut scale      delta alpha beta       pwr r    N  n1  n0
1 0.40 0.5 0.6931472    OR -0.4054651 0.025  0.2 0.8002481 1 1550 775 775
2 0.45 0.5 0.6931472    OR -0.2006707 0.025  0.2 0.8003296 1  522 261 261
3 0.50 0.5 0.6931472    OR  0.0000000 0.025  0.2 0.8001346 1  350 175 175
> getN_Bin_Equi(p0 = 0.5, p1 = seq(0.4, 0.5, 0.05), cut = -log(0.5), alpha = 0.025, beta = NA, N = 200, r = 1, scale = "OR")
   p1  p0       cut scale      delta alpha beta       pwr r   N  n1  n0
1 0.40 0.5 0.6931472    OR -0.4054651 0.025   NA 0.1404656 1 200 100 100
2 0.45 0.5 0.6931472    OR -0.2006707 0.025   NA 0.2951093 1 200 100 100
3 0.50 0.5 0.6931472    OR  0.0000000 0.025   NA 0.3763482 1 200 100 100
```

Equivalence Tests for the Odds Ratio of Two Proportions

Numeric Results

Test Statistic: Farrington & Manning Likelihood Score Test
Hypotheses: H0: OR ≤ OR0.L or OR ≥ OR0.U vs. H1: OR0.L < OR < OR0.U

Target Power	Actual Power*	N1	N2	N	Ref. P2	P1.0L	P1.0U	OR0.L	OR0.U	OR1	Alpha
0.8	0.80028	769	769	1538	0.5	0.33333	0.66667	0.5	2	0.66667	0.025
0.8	0.80120	258	258	516	0.5	0.33333	0.66667	0.5	2	0.81818	0.025
0.8	0.80269	172	172	344	0.5	0.33333	0.66667	0.5	2	1.00000	0.025

* Power was computed using the normal approximation method.

Equivalence Tests for the Odds Ratio of Two Proportions

Numeric Results

Test Statistic: Farrington & Manning Likelihood Score Test
Hypotheses: H0: OR ≤ OR0.L or OR ≥ OR0.U vs. H1: OR0.L < OR < OR0.U

Power*	N1	N2	N	Ref. P2	P1.0L	P1.0U	OR0.L	OR0.U	OR1	Alpha
0.14266	100	100	200	0.5	0.33333	0.66667	0.5	2	0.66667	0.025
0.30394	100	100	200	0.5	0.33333	0.66667	0.5	2	0.81818	0.025
0.39032	100	100	200	0.5	0.33333	0.66667	0.5	2	1.00000	0.025

* Power was computed using the normal approximation method.

getN_Surv_Super

```
(v <- getN_Surv_Super(delta = log(seq(0.7, 0.8, 0.05)), alpha = 0.025, beta =
0.2, N = NA, r = 1, criterion = 1))
getN_Surv_Super(delta = log(seq(0.7, 0.8, 0.05)), alpha = 0.025, beta = NA, N =
300, r = 1, criterion = 1)
```

```
> (v <- getN_Surv_Super(delta = log(seq(0.7, 0.8, 0.05)), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1))
      delta alpha beta       pwr r   N  n1  n0 criterion
1 -0.3566749 0.025  0.2 0.8019195 1 248 124 124         1
2 -0.2876821 0.025  0.2 0.8006692 1 380 190 190         1
3 -0.2231436 0.025  0.2 0.8009186 1 632 316 316         1
> getN_Surv_Super(delta = log(seq(0.7, 0.8, 0.05)), alpha = 0.025, beta = NA, N = 300, r = 1, criterion = 1)
      delta alpha beta       pwr r   N  n1  n0 criterion
1 -0.3566749 0.025   NA 0.8705367 1 300 150 150         1
2 -0.2876821 0.025   NA 0.7024416 1 300 150 150         1
3 -0.2231436 0.025   NA 0.4890368 1 300 150 150         1
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	Expected SS (H0)	Expected SS (H1)	Maximum Events	Exp. Events (H0)	Exp. Events (H1)
57		Des57	Superiority	1	1-Sided	0.025	0.8	1	461	461	461	247	247	247
58		Des58	Superiority	1	1-Sided	0.025	0.801	1	694	694	694	380	380	380
59		Des59	Superiority	1	1-Sided	0.025	0.8	1	1130	1130	1130	631	631	631

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	Expected SS (H0)	Expected SS (H1)	Maximum Events	Exp. Events (H0)	Exp. Events (H1)
63		Des63	Superiority	1	1-Sided	0.025	0.871	1	560	560	560	300	300	300
64		Des64	Superiority	1	1-Sided	0.025	0.702	1	548	548	548	300	300	300
65		Des65	Superiority	1	1-Sided	0.025	0.489	1	537	537	537	300	300	300

getN_Surv_Noninf

```
(v <- getN_Surv_Noninf(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha =
0.025, beta = 0.2, N = NA, r = 1, direct = -1, criterion = 1))
getN_Surv_Noninf(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025,
beta = NA, N = 200, r = 1, direct = -1, criterion = 1)
```

```
> (v <- getN_Surv_Noninf(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, direct = -1, criterion = 1))
      delta       cut alpha beta       pwr r    N  n1  n0 criterion direct
1 0.00000000 0.2623643 0.025  0.2 0.8016296 1  458 229 229         1     -1
2 0.04879016 0.2623643 0.025  0.2 0.8009732 1  690 345 345         1     -1
3 0.09531018 0.2623643 0.025  0.2 0.8003475 1 1126 563 563         1     -1
> getN_Surv_Noninf(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025, beta = NA, N = 200, r = 1, direct = -1, criterion = 1)
      delta       cut alpha beta       pwr r   N  n1  n0 criterion direct
1 0.00000000 0.2623643 0.025   NA 0.4582798 1 200 100 100         1     -1
2 0.04879016 0.2623643 0.025   NA 0.3264392 1 200 100 100         1     -1
3 0.09531018 0.2623643 0.025   NA 0.2180743 1 200 100 100         1     -1
```

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	Expected SS (H0)	Expected SS (H1)	Maximum Events	Exp. Events (H0)	Exp. Events (H1)
72		Des72	Noninferiority	1	1-Sided	0.025	0.801	1	765	765	765	457	457	457
73		Des73	Noninferiority	1	1-Sided	0.025	0.8	1	1136	1136	1136	689	689	689
74		Des74	Noninferiority	1	1-Sided	0.025	0.8	1	1831	1831	1831	1126	1126	1126

Output Preview

Output Preview

Sr. No.		ID	Design Type	No. of Looks	Test Type	Specified α	Power	nt/nc	Sample Size	Expected SS (H0)	Expected SS (H1)	Maximum Events	Exp. Events (H0)	Exp. Events (H1)
75		Des75	Noninferiority	1	1-Sided	0.025	0.458	1	335	335	335	200	200	200
76		Des76	Noninferiority	1	1-Sided	0.025	0.326	1	330	330	330	200	200	200
77		Des77	Noninferiority	1	1-Sided	0.025	0.218	1	326	326	326	200	200	200

getN_Surv_Equi

EAST and PASS do not support.

```
(v <- getN_Surv_Equi(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha =
0.025, beta = 0.2, N = NA, r = 1, criterion = 1))
getN_Surv_Equi(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025,
beta = NA, N = 400, r = 1, criterion = 1)
```

```
> (v <- getN_Surv_Equi(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025, beta = 0.2, N = NA, r = 1, criterion = 1))
      delta       cut alpha beta       pwr r    N  n1  n0 criterion
1 0.00000000 0.2623643 0.025  0.2 0.8013135 1  612 306 306         1
2 0.04879016 0.2623643 0.025  0.2 0.8001932 1  714 357 357         1
3 0.09531018 0.2623643 0.025  0.2 0.8003209 1 1126 563 563         1
> getN_Surv_Equi(delta = log(seq(1, 1.1, 0.05)), cut = log(1.3), alpha = 0.025, beta = NA, N = 400, r = 1, criterion = 1)
      delta       cut alpha beta       pwr r   N  n1  n0 criterion
1 0.00000000 0.2623643 0.025   NA 0.4931040 1 400 200 200         1
2 0.04879016 0.2623643 0.025   NA 0.4450186 1 400 200 200         1
3 0.09531018 0.2623643 0.025   NA 0.3331660 1 400 200 200         1
```

getN_Count_Super

```
(v <- getN_Count_Super(delta = log(seq(0.7, 0.8, 0.05)), lambda0 = 0.1, t = 5, k
= 0, alpha = 0.025, beta = 0.2, N = NA, r = 1))
getN_Count_Super(delta = log(seq(0.7, 0.8, 0.05)), lambda0 = 0.1, t = 5, k = 0,
alpha = 0.025, beta = NA, N = 1000, r = 1)
```

```
> (v <- getN_Count_Super(delta = log(seq(0.7, 0.8, 0.05)), lambda0 = 0.1, t = 5, k = 0, alpha = 0.025, beta = 0.2, N = NA, r = 1))
      delta lambda0 t k alpha beta       pwr r    N  n1  n0
1 -0.3566749     0.1 5 0 0.025  0.2 0.8004314 1  600 300 300
2 -0.2876821     0.1 5 0 0.025  0.2 0.8003745 1  886 443 443
3 -0.2231436     0.1 5 0 0.025  0.2 0.8003673 1 1420 710 710
> getN_Count_Super(delta = log(seq(0.7, 0.8, 0.05)), lambda0 = 0.1, t = 5, k = 0, alpha = 0.025, beta = NA, N = 1000, r = 1)
      delta lambda0 t k alpha beta       pwr r    N  n1  n0
1 -0.3566749     0.1 5 0 0.025   NA 0.9514277 1 1000 500 500
2 -0.2876821     0.1 5 0 0.025   NA 0.8456202 1 1000 500 500
3 -0.2231436     0.1 5 0 0.025   NA 0.6525358 1 1000 500 500
```

Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Testing the Ratio of Two Negative Binomial Rates using the Wald or LR Test

Alternative Hypothesis: One-Sided (H0: RR ≥ 1 vs. Ha: RR < 1)
Null Variance Calculation Method: Use True Event Rates (λ1 and λ2)

Power	N1	N2	N	Ave Expos Time μ(t)	Grp 1 Event Rate λ1	Grp 2 Event Rate λ2	Event Rate Ratio RR	Neg Binom Disp κ	Alpha
0.80043	300	300	600	5	0.1	0.070	0.70	0	0.025
0.80037	443	443	886	5	0.1	0.075	0.75	0	0.025
0.80037	710	710	1420	5	0.1	0.080	0.80	0	0.025

Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Testing the Ratio of Two Negative Binomial Rates using the Wald or LR Test

Alternative Hypothesis: One-Sided (H0: RR ≥ 1 vs. Ha: RR < 1)
Null Variance Calculation Method: Use True Event Rates (λ1 and λ2)

Power	N1	N2	N	Ave Expos Time μ(t)	Grp 1 Event Rate λ1	Grp 2 Event Rate λ2	Event Rate Ratio RR	Neg Binom Disp κ	Alpha
0.95143	500	500	1000	5	0.1	0.070	0.70	0	0.025
0.84562	500	500	1000	5	0.1	0.075	0.75	0	0.025
0.65254	500	500	1000	5	0.1	0.080	0.80	0	0.025

References

Zhu, H. and Lakkis, H. 2014. 'Sample Size Calculation for Comparing Two Negative Binomial Rates.' Statistics in Medicine, Volume 33, Pages 376-387.

getN_Count_Noninf

```
(v <- getN_Count_Noninf(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0
= 0.1, t = 5, k = 1, alpha = 0.025, beta = 0.2, N = NA, r = 1))
getN_Count_Noninf(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1,
t = 5, k = 1, alpha = 0.025, beta = NA, N = 1000, r = 1)
```

```
> (v <- getN_Count_Noninf(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = 0.2, N = NA, r = 1))
       delta lambda0 t k       cut alpha beta       pwr r    N  n1  n0 direct
1 -0.10536052     0.1 5 1 0.3566749 0.025  0.2 0.8003815 1 1548 774 774      1
2 -0.05129329     0.1 5 1 0.3566749 0.025  0.2 0.8001236 1 1028 514 514      1
3  0.00000000     0.1 5 1 0.3566749 0.025  0.2 0.8008664 1  742 371 371      1
> getN_Count_Noninf(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = NA, N = 1000, r = 1)
       delta lambda0 t k       cut alpha beta       pwr r    N  n1  n0 direct
1 -0.10536052     0.1 5 1 0.3566749 0.025   NA 0.6151902 1 1000 500 500      1
2 -0.05129329     0.1 5 1 0.3566749 0.025   NA 0.7891974 1 1000 500 500      1
3  0.00000000     0.1 5 1 0.3566749 0.025   NA 0.9025153 1 1000 500 500      1
```

Non-Inferiority Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Non-Inferiority Tests of the Ratio of Two Negative Binomial Rates

Test Direction Assumption: Higher Negative Binomial Rates Are Better
Hypotheses: H0: λ2 / λ1 ≤ R0 vs. H1: λ2 / λ1 > R0
Variance Calculation Method: Using Assumed True Rates

Power	N1	N2	N	Average Exposure Time μ(t)	Grp 1 Cntrl Event Rate λ1	Grp 2 Trt Event Rate λ2	Event Rate Ratio λ2 / λ1	Non-Inf-eriority Ratio R0	Disper-sion φ	Alpha
0.80038	774	774	1548	5	0.1	0.090	0.90	0.7	1	0.025
0.80012	514	514	1028	5	0.1	0.095	0.95	0.7	1	0.025
0.80087	371	371	742	5	0.1	0.100	1.00	0.7	1	0.025

Non-Inferiority Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Non-Inferiority Tests of the Ratio of Two Negative Binomial Rates

Test Direction Assumption: Higher Negative Binomial Rates Are Better
Hypotheses: H0: λ2 / λ1 ≤ R0 vs. H1: λ2 / λ1 > R0
Variance Calculation Method: Using Assumed True Rates

Power	N1	N2	N	Average Exposure Time μ(t)	Grp 1 Cntrl Event Rate λ1	Grp 2 Trt Event Rate λ2	Event Rate Ratio λ2 / λ1	Non-Inf-eriority Ratio R0	Disper-sion φ	Alpha
0.61519	500	500	1000	5	0.1	0.090	0.90	0.7	1	0.025
0.78920	500	500	1000	5	0.1	0.095	0.95	0.7	1	0.025
0.90252	500	500	1000	5	0.1	0.100	1.00	0.7	1	0.025

References

Zhu, H. 2017. 'Sample Size Calculation for Comparing Two Poisson or Negative Binomial Rates in Non-Inferiority or Equivalence Trials.' Statistics in Biopharmaceutical Research, 9(1), 107-115, doi:10.1080/19466315.2016.1225594.

getN_Count_Equi

```
(v <- getN_Count_Equi(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = 0.2, N = NA, r = 1))
getN_Count_Equi(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = NA, N = 1000, r = 1)
```

```
> (v <- getN_Count_Equi(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = 0.2, N = NA, r = 1))
       delta lambda0 t k       cut alpha beta       pwr r    N  n1  n0
1 -0.10536052     0.1 5 1 0.3566749 0.025  0.2 0.8001918 1 1550 775 775
2 -0.05129329     0.1 5 1 0.3566749 0.025  0.2 0.8007784 1 1106 553 553
3  0.00000000     0.1 5 1 0.3566749 0.025  0.2 0.8004966 1  992 496 496
> getN_Count_Equi(delta = log(seq(0.9, 1, 0.05)), cut = -log(0.7), lambda0 = 0.1, t = 5, k = 1, alpha = 0.025, beta = NA, N = 1000, r = 1)
       delta lambda0 t k       cut alpha beta       pwr r    N  n1  n0
1 -0.10536052     0.1 5 1 0.3566749 0.025   NA 0.6006288 1 1000 500 500
2 -0.05129329     0.1 5 1 0.3566749 0.025   NA 0.7475619 1 1000 500 500
3  0.00000000     0.1 5 1 0.3566749 0.025   NA 0.8050307 1 1000 500 500
```

Equivalence Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Equivalence Tests of the Ratio of Two Negative Binomial Rates

Hypotheses: H0: λ2 / λ1 ≤ RL or λ2 / λ1 ≥ RU vs. H1: RL < λ2 / λ1 < RU
Variance Calculation Method: Using Assumed True Rates

Power	N1	N2	N	Average Exposure Time μ(t)	Grp 1 Cntrl Event Rate λ1	Grp 2 Trt Event Rate λ2	Event Rate Ratio λ2 / λ1	Lower Equiv. Limit RL	Upper Equiv. Limit RU	Disper-sion φ	Alpha
0.80019	775	775	1550	5	0.1	0.090	0.90	0.7	1.42857	1	0.025
0.80078	553	553	1106	5	0.1	0.095	0.95	0.7	1.42857	1	0.025
0.80050	496	496	992	5	0.1	0.100	1.00	0.7	1.42857	1	0.025

Equivalence Tests for the Ratio of Two Negative Binomial Rates

Numeric Results for Equivalence Tests of the Ratio of Two Negative Binomial Rates

Hypotheses: H0: λ2 / λ1 ≤ RL or λ2 / λ1 ≥ RU vs. H1: RL < λ2 / λ1 < RU
Variance Calculation Method: Using Assumed True Rates

Power	N1	N2	N	Average Exposure Time μ(t)	Grp 1 Cntrl Event Rate λ1	Grp 2 Trt Event Rate λ2	Event Rate Ratio λ2 / λ1	Lower Equiv. Limit RL	Upper Equiv. Limit RU	Disper-sion φ	Alpha
0.60063	500	500	1000	5	0.1	0.090	0.90	0.7	1.42857	1	0.025
0.74756	500	500	1000	5	0.1	0.095	0.95	0.7	1.42857	1	0.025
0.80503	500	500	1000	5	0.1	0.100	1.00	0.7	1.42857	1	0.025