How to create analysis result (one- and multi-arm) plots with rpact

Gernot Wassmer and Friedrich Pahlke

Last change: 01 Dezember, 2020

Contents

Summary

This R Markdown document provides many different examples for creating one- and multi-arm analysis result plots with rpact and ggplot2.

1 Preparation and design

First, load the rpact package

```
library(rpact)
packageVersion("rpact")
```

```
## [1] '3.0.3'
```

1.1 Create a design

```
designIN <- getDesignInverseNormal(kMax = 4, alpha = 0.02,
    futilityBounds = c(-0.5, 0, 0.5),   bindingFutility = FALSE,
    typeOfDesign = "asKD", gammaA = 1.2,
    informationRates = c(0.15, 0.4, 0.7, 1))
```

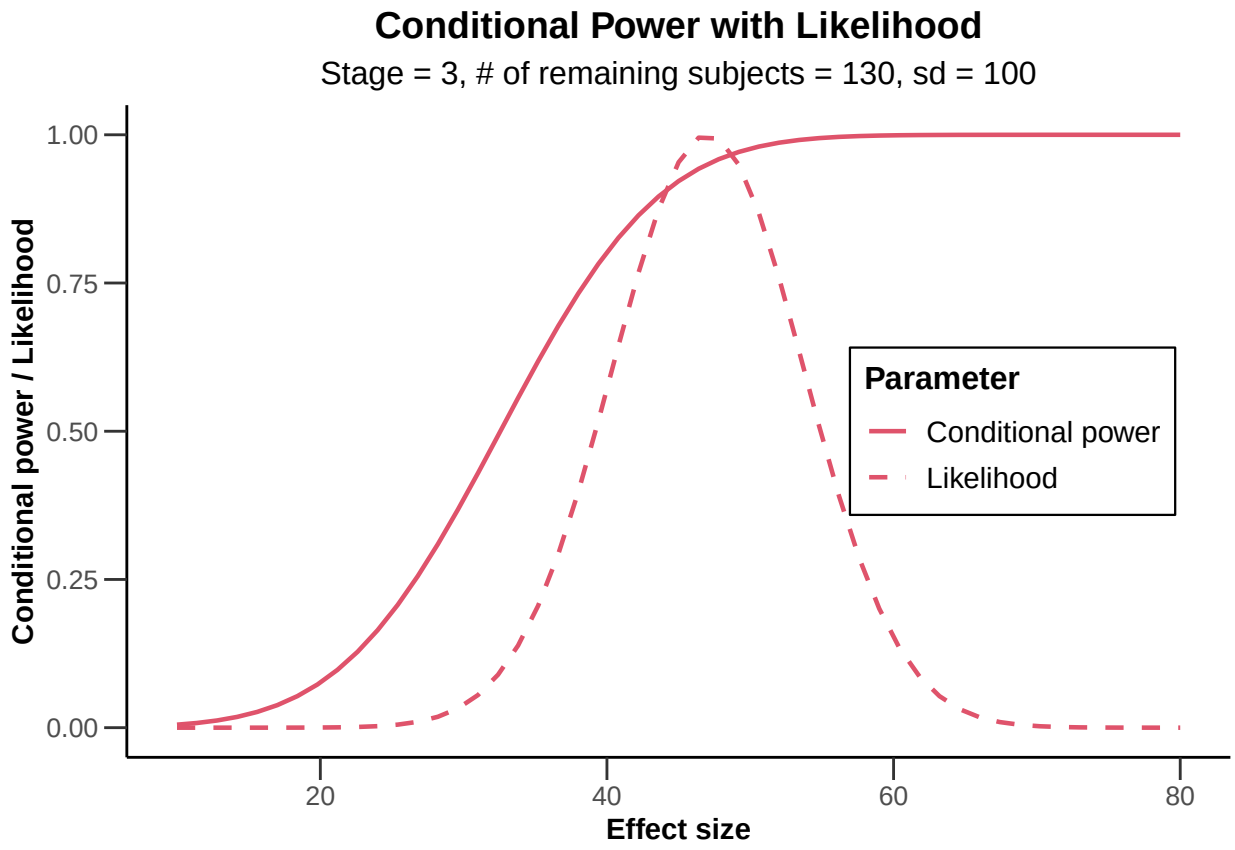
2 Analysis results base

2.1 Analysis results base - means

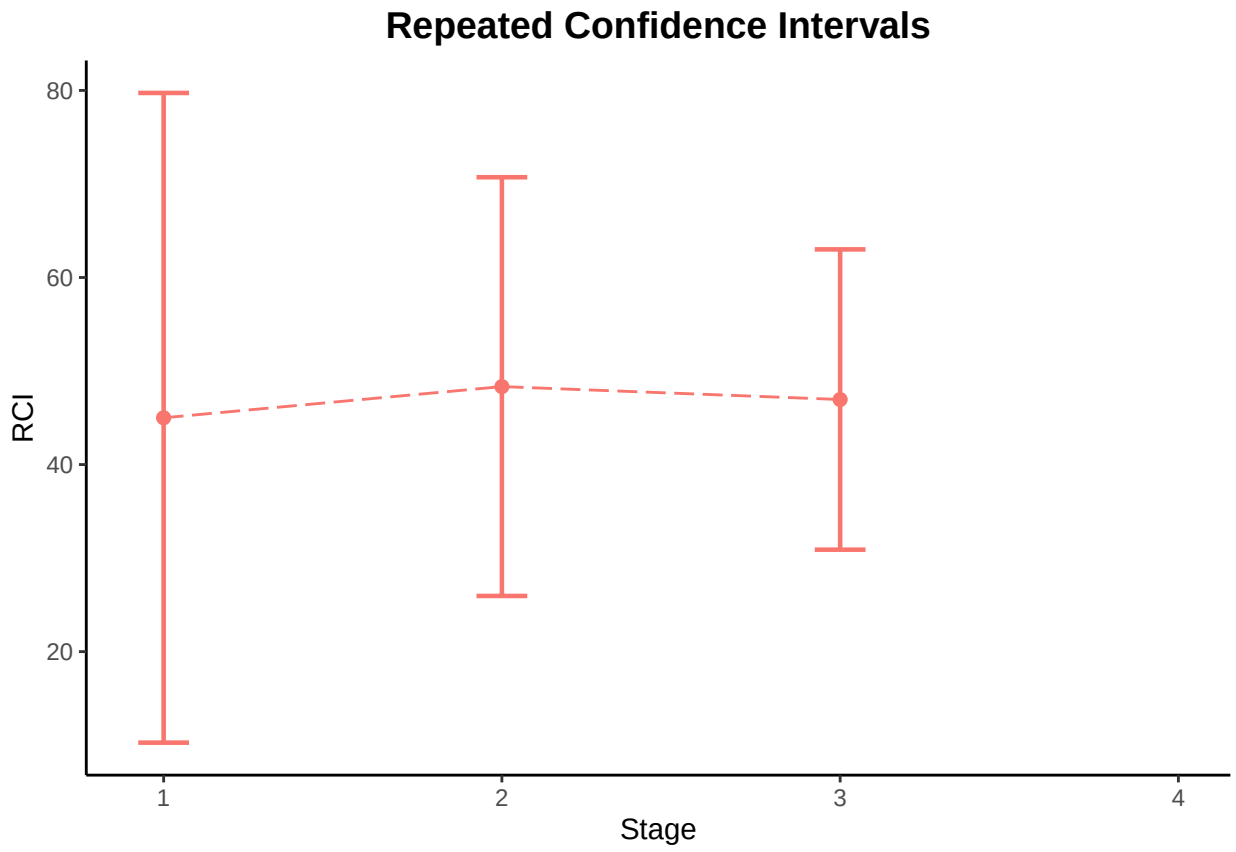
```
simpleDataExampleMeans1 <- getDataset(
    n = c(120, 130, 130),
    means = c(0.45, 0.51, 0.45) * 100,
    stDevs = c(1.3, 1.4, 1.2) * 100
)

x <- getAnalysisResults(design = designIN, dataInput = simpleDataExampleMeans1,
    nPlanned = 130, thetaH0 = 30, thetaH1 = 60, assumedStDev = 100)
```

```
## Calculation of final confidence interval performed for kMax = 4 (for kMax > 2, it is theoretically sl
```

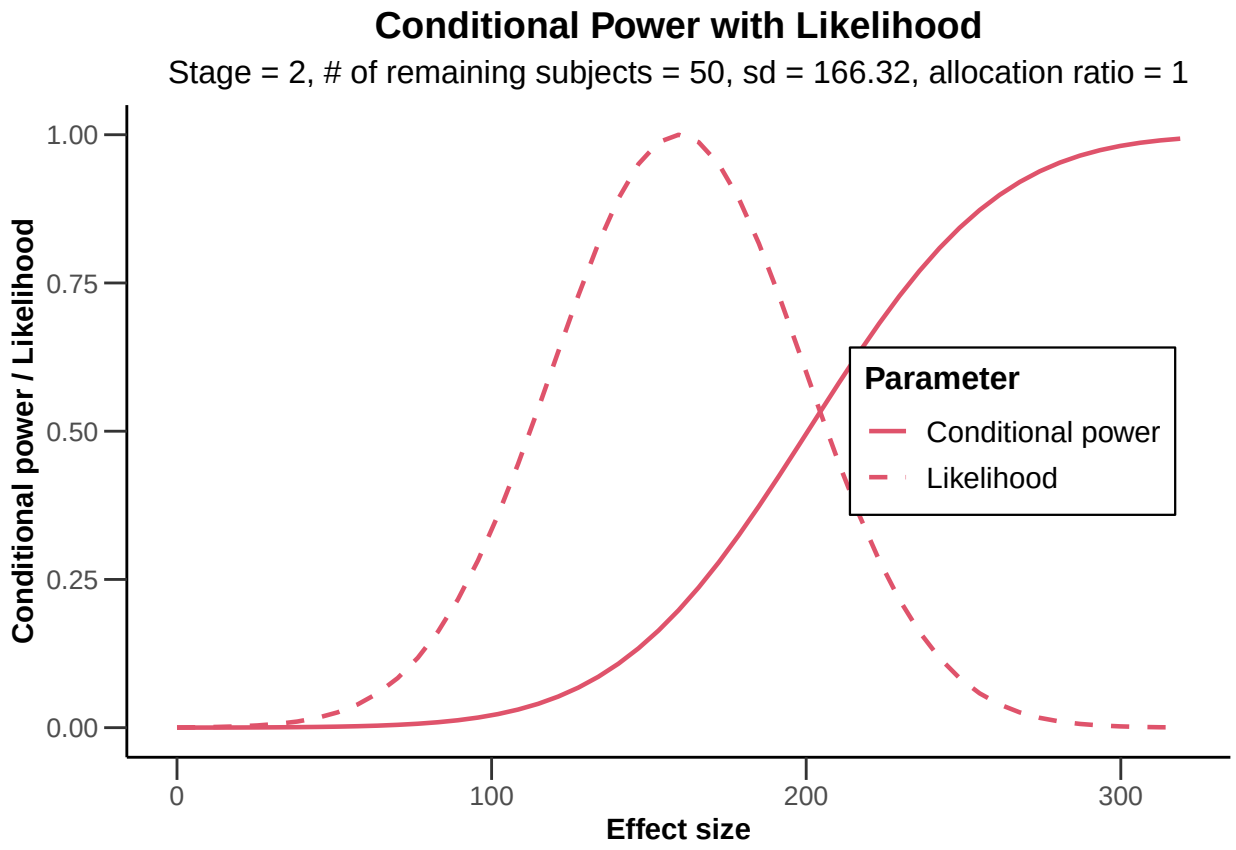
```
plot(x, thetaRange = c(10,80))
```



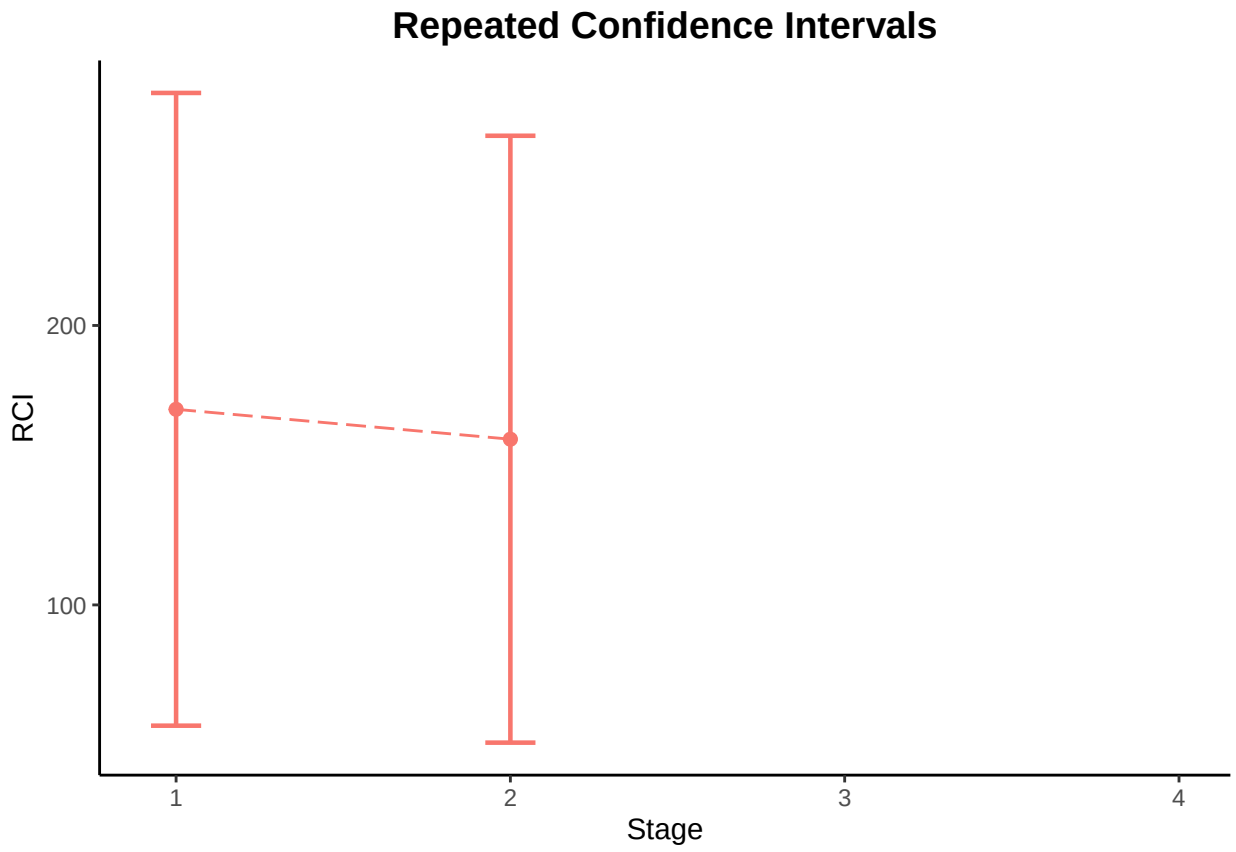

```
plot(x, type = 2)
```




```
simpleDataExampleMeans2 <- getDataset(
    n1 = c(23, 13, 22, 13),
    n2 = c(22, 11, 22, 11),
    means1 = c(2.7, 2.5, 4.5, 2.5) * 100,
    means2 = c(1, 1.1, 1.3, 1) * 100,
    stds1 = c(1.3, 2.4, 2.2, 1.3) * 100,
    stds2 = c(1.2, 2.2, 2.1, 1.3) * 100
)

x <- getAnalysisResults(design = designIN, dataInput = simpleDataExampleMeans2,
    thetaH0 = 110, equalVariances = TRUE, directionUpper = TRUE, stage = 2)
plot(x, nPlanned = c(20, 30))
```


Conditional Power with Likelihood
Stage = 2, # of remaining subjects = 50, sd = 166.32, allocation ratio = 1
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
0
100
200
300
Effect size
Parameter
Conditional power
Likelihood

```
plot(x, type = 2)
```



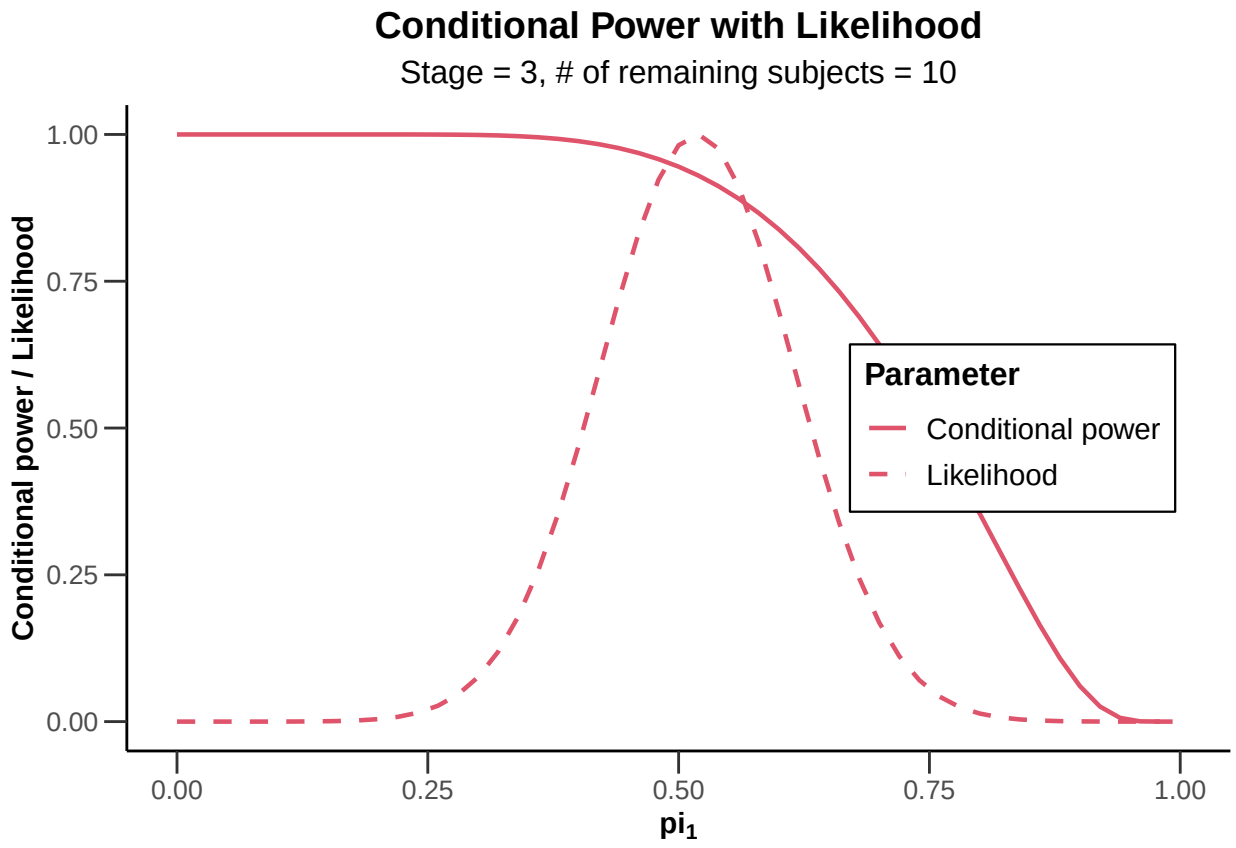
2.2 Analysis results base - rates

```
simpleDataExampleRates1 <- getDataset(
    n = c(8, 10, 9, 11),
    events = c(4, 5, 5, 6)
)

x <- getAnalysisResults(design = designIN, dataInput = simpleDataExampleRates1,
    stage = 3,  thetaH0 = 0.75, normalApproximation = TRUE,
    directionUpper = FALSE, nPlanned = 10)
```

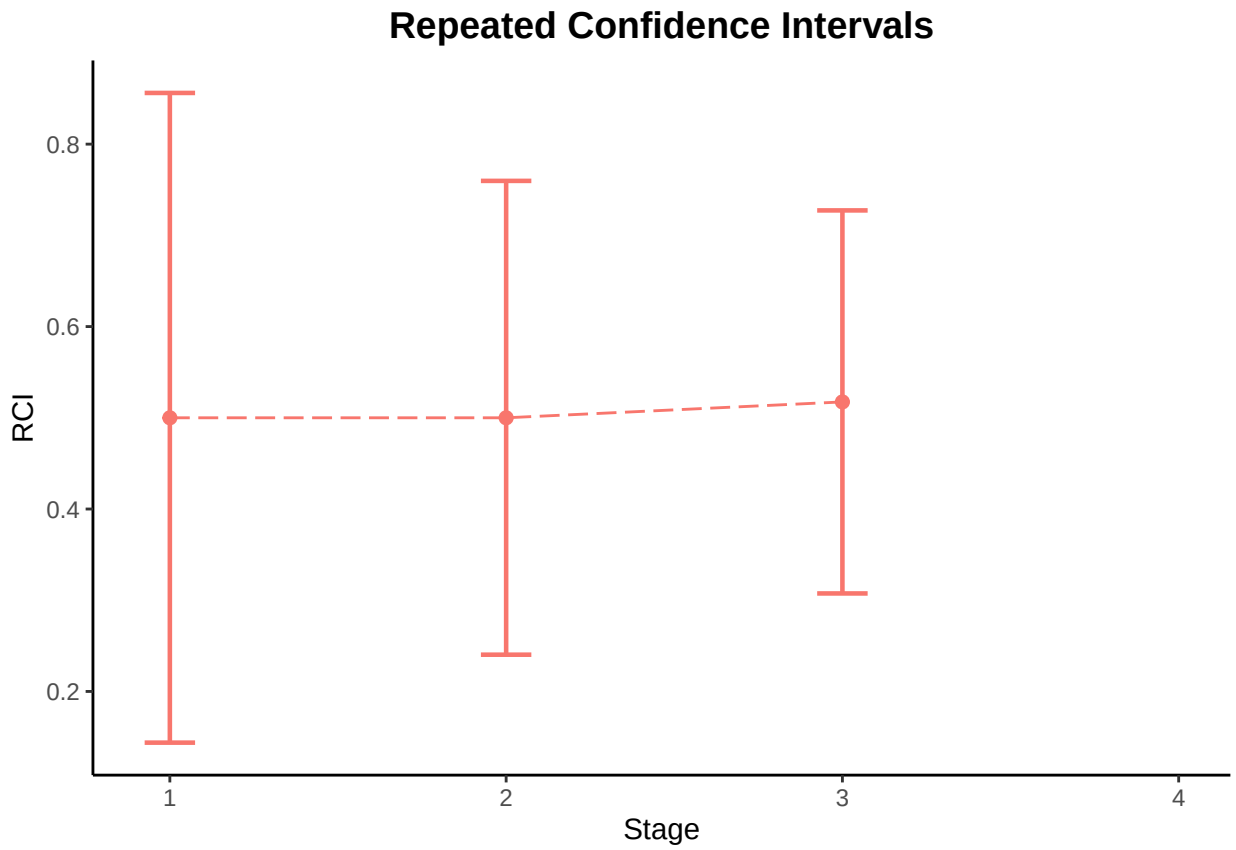
```
## Calculation of final confidence interval performed for kMax = 4 (for kMax > 2, it is theoretically sl
```

```
plot(x)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

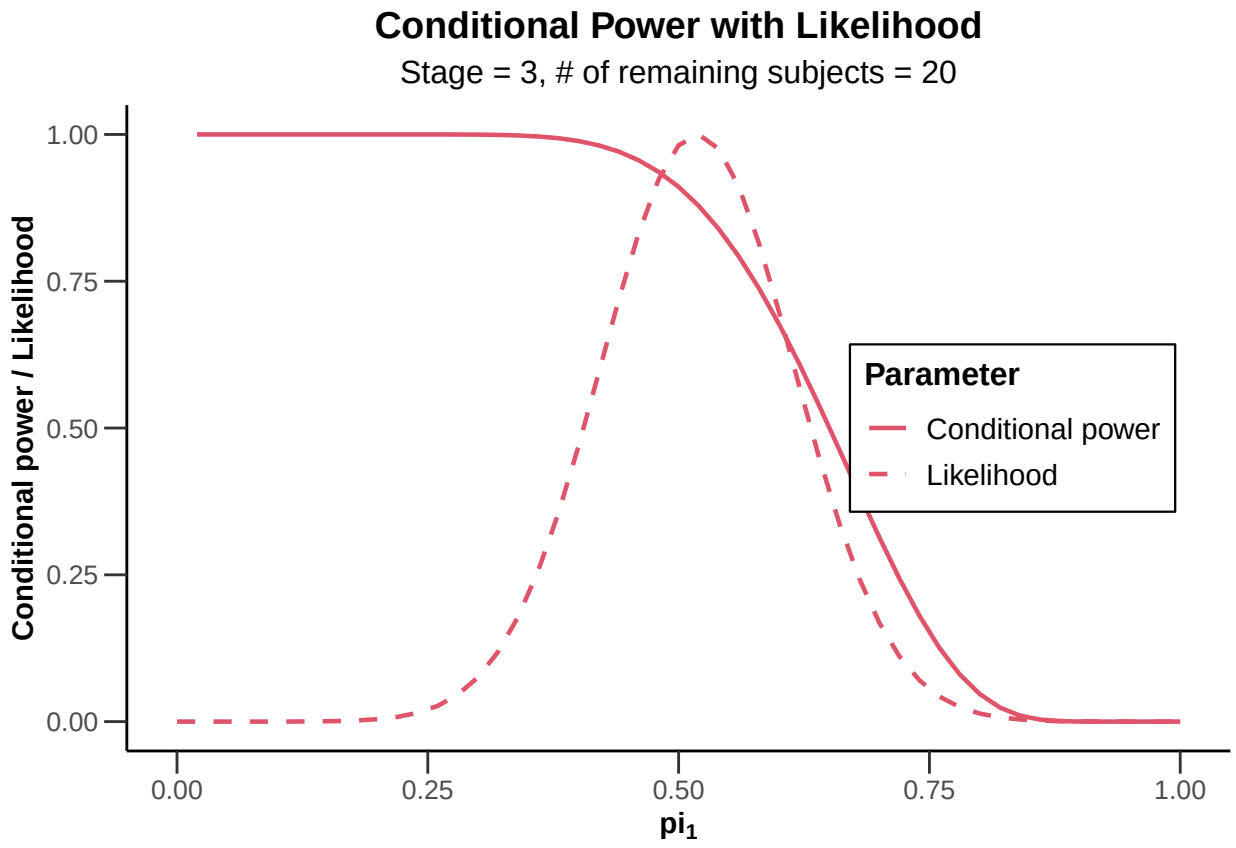

Conditional Power with Likelihood
Stage = 3, # of remaining subjects = 10
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
pi1
Parameter
Conditional power
Likelihood

```
plot(x, type = 2)
```

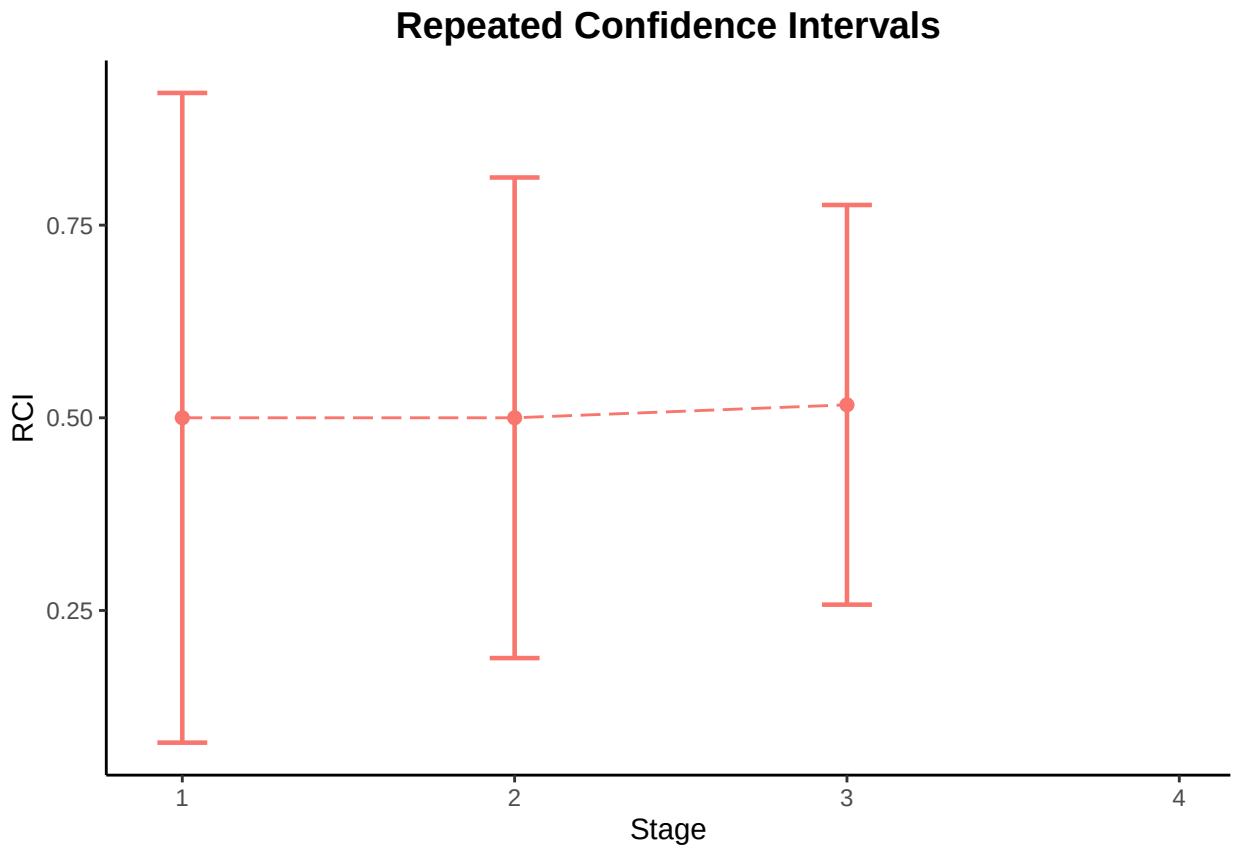



```
x <- getAnalysisResults(design = designIN, dataInput = simpleDataExampleRates1,
    stage = 3,  thetaH0 = 0.75, normalApproximation = FALSE,
    directionUpper = FALSE)
plot(x, nPlanned = 20)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```


Conditional Power with Likelihood
Stage = 3, # of remaining subjects = 20
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
pi1
Parameter
Conditional power
Likelihood

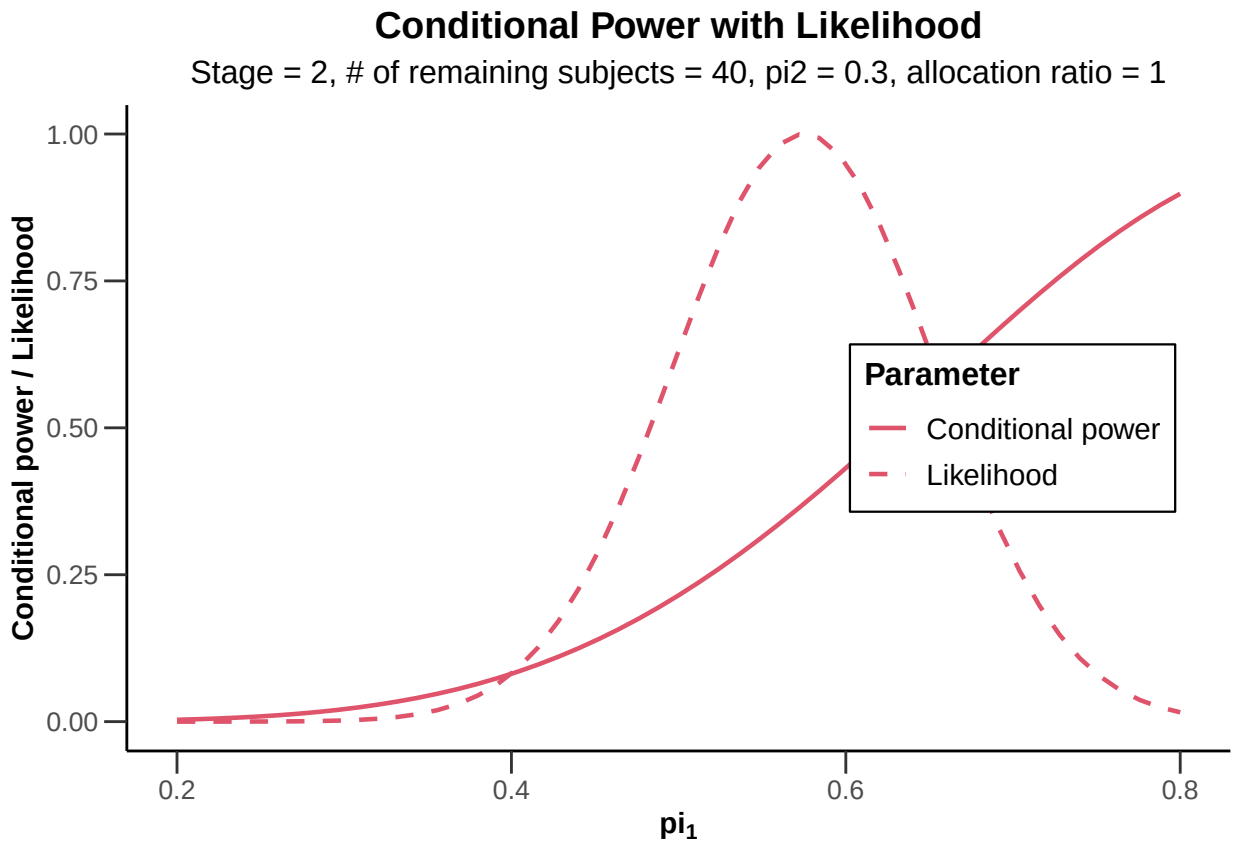
```
plot(x, type = 2)
```



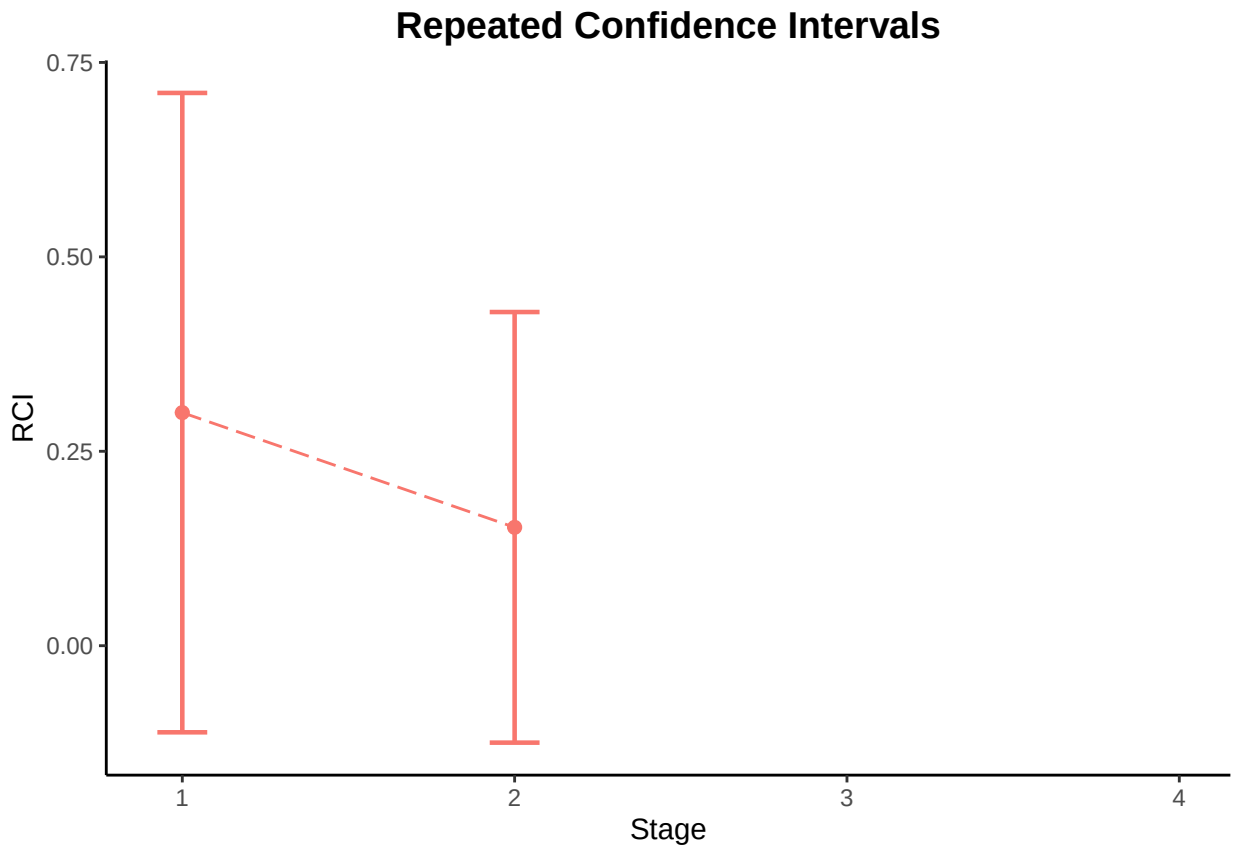

```
simpleDataExampleRates2 <- getDataset(
    n1 = c(17, 23, 22),
    n2 = c(18, 20, 19),
    events1 = c(11, 12, 17),
    events2 = c(5, 10, 7)
)

x <- getAnalysisResults(designIN, simpleDataExampleRates2, thetaH0 = 0,
    stage = 2, directionUpper = TRUE, normalApproximation = FALSE,
    pi1 = 0.9, pi2 = 0.3, nPlanned = c(20, 20))
```

```
## Repeated confidence intervals will be calculated under the normal approximation
```

```
plot(x, piTreatmentRange = c(0.2,0.8))
```


Conditional Power with Likelihood
Stage = 2, # of remaining subjects = 40, pi2 = 0.3, allocation ratio = 1
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
0.2
0.4
0.6
0.8
pi1
Parameter
Conditional power
Likelihood

```
plot(x, type = 2)
```

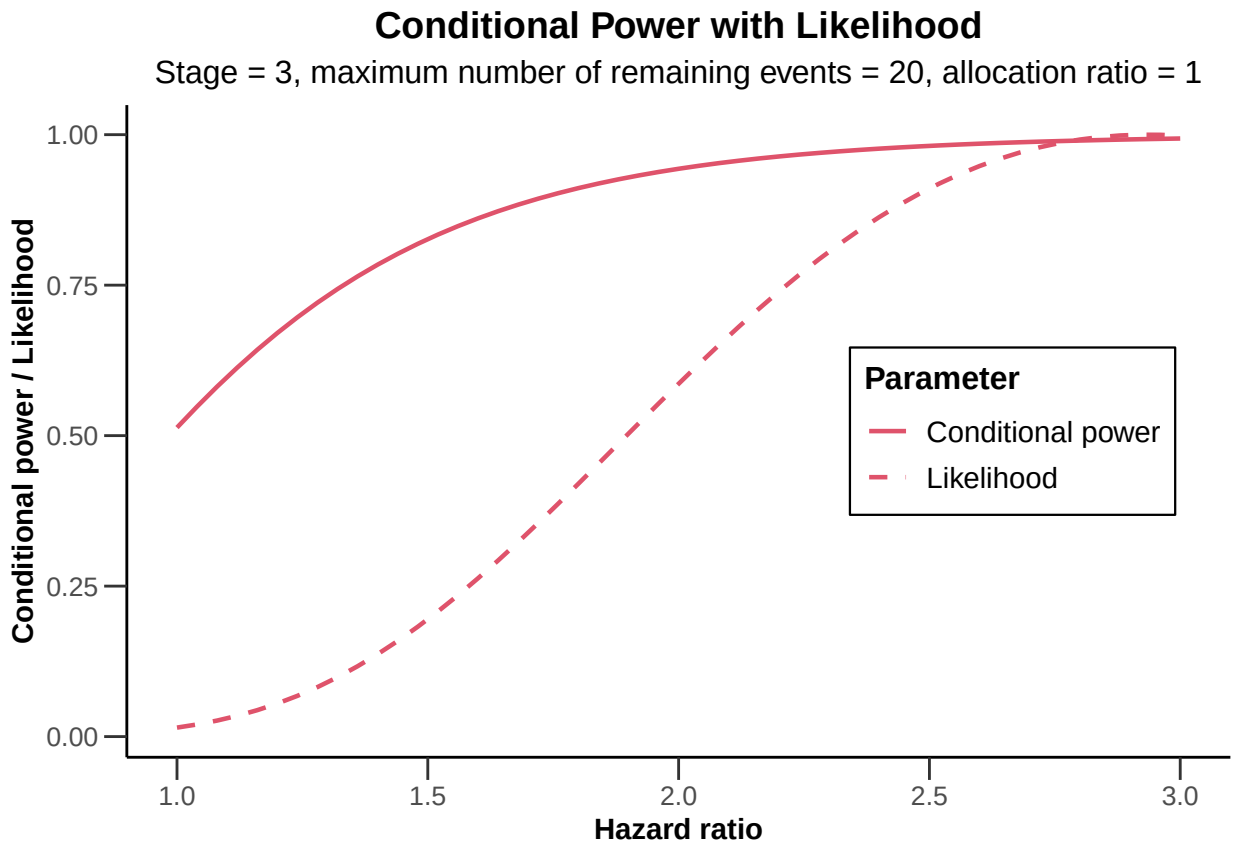


2.3 Analysis results base - survival

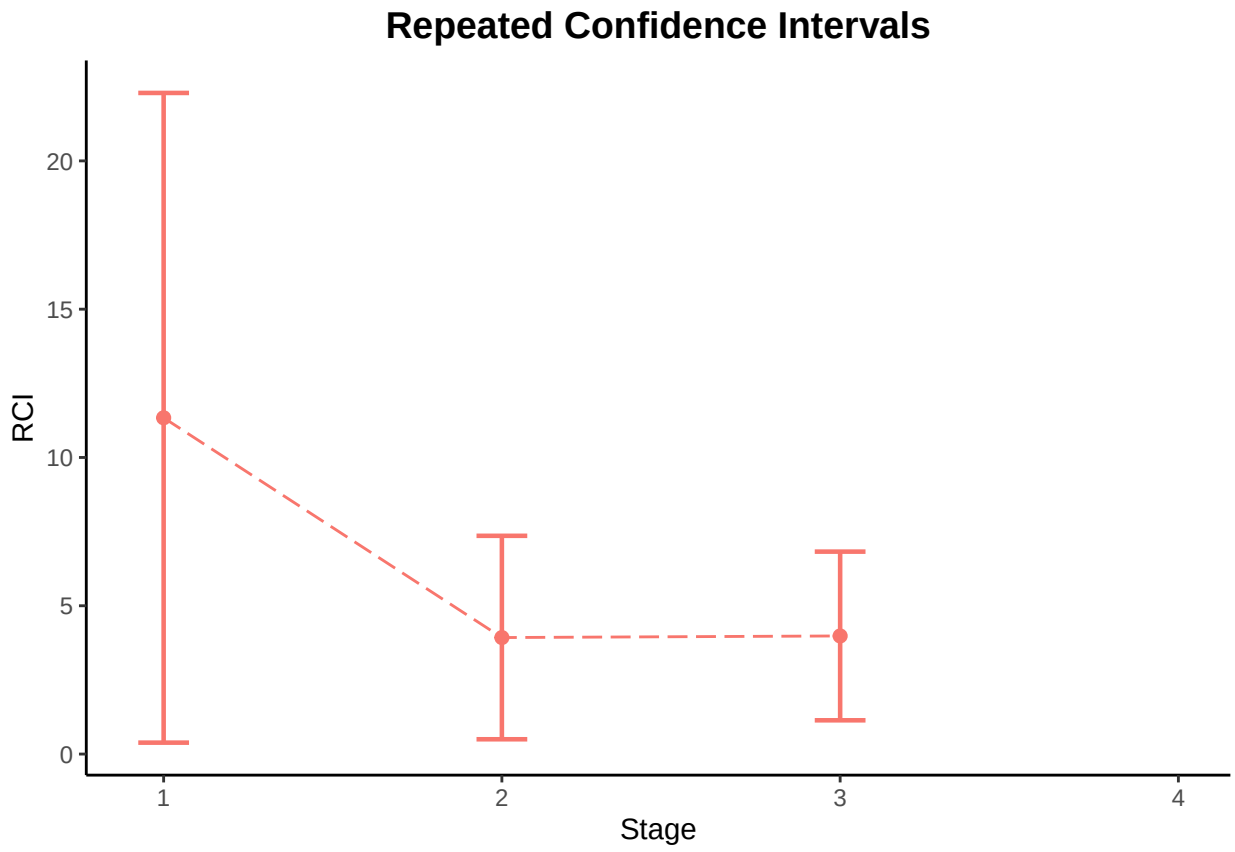
```
simpleDataExampleSurvival <- getDataset(
    overallEvents = c(8, 15, 29),
    overallAllocationRatios = c(1, 1, 1),
    overallLogRanks = c(1.52, 1.38, 2.9)
)

x <- getAnalysisResults(designIN, simpleDataExampleSurvival,
    directionUpper = TRUE, nPlanned = 20)
```

```
## Calculation of final confidence interval performed for kMax = 4 (for kMax > 2, it is theoretically sl
```

```
plot(x, thetaRange = c(1,3))
```


Conditional Power with Likelihood
Stage = 3, maximum number of remaining events = 20, allocation ratio = 1
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
1.0
1.5
2.0
2.5
3.0
Hazard ratio
Parameter
Conditional power
Likelihood

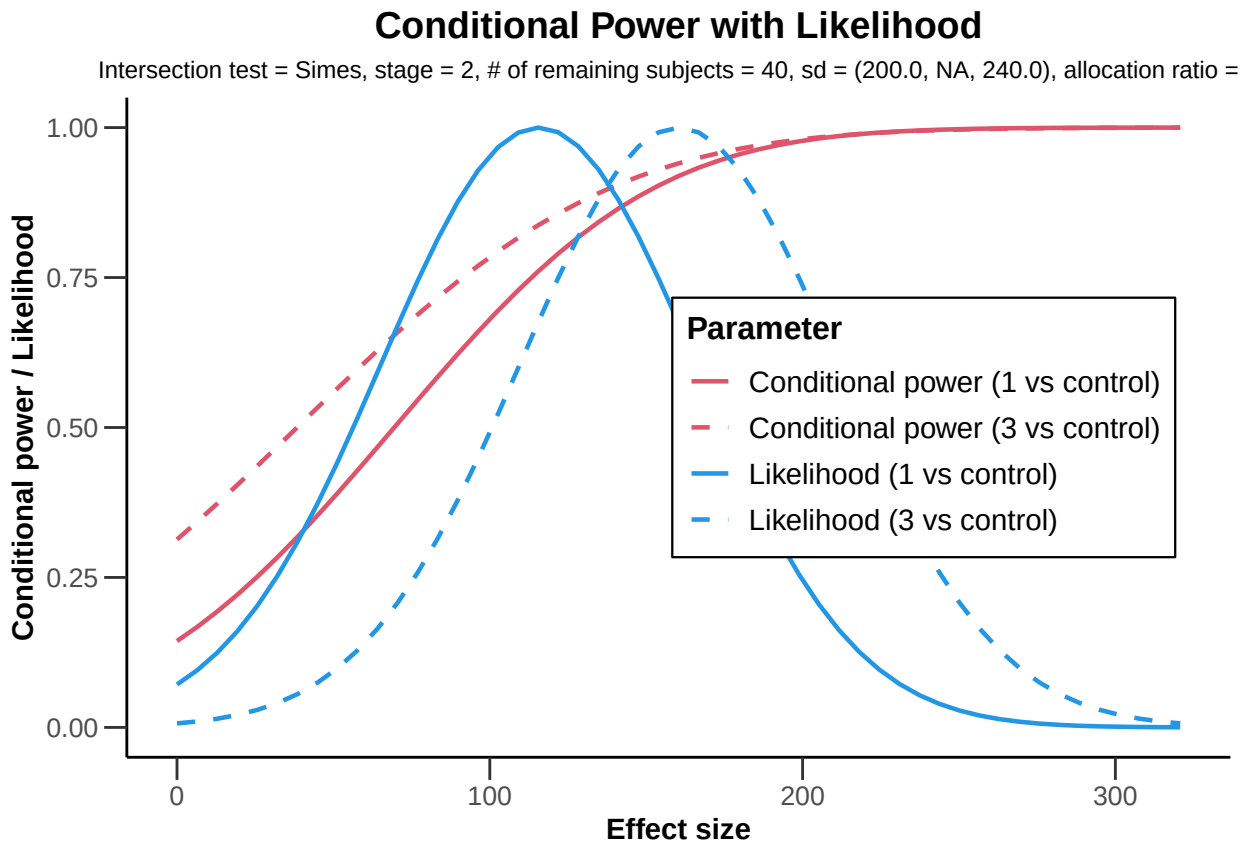
```
plot(x, type = 2)
```



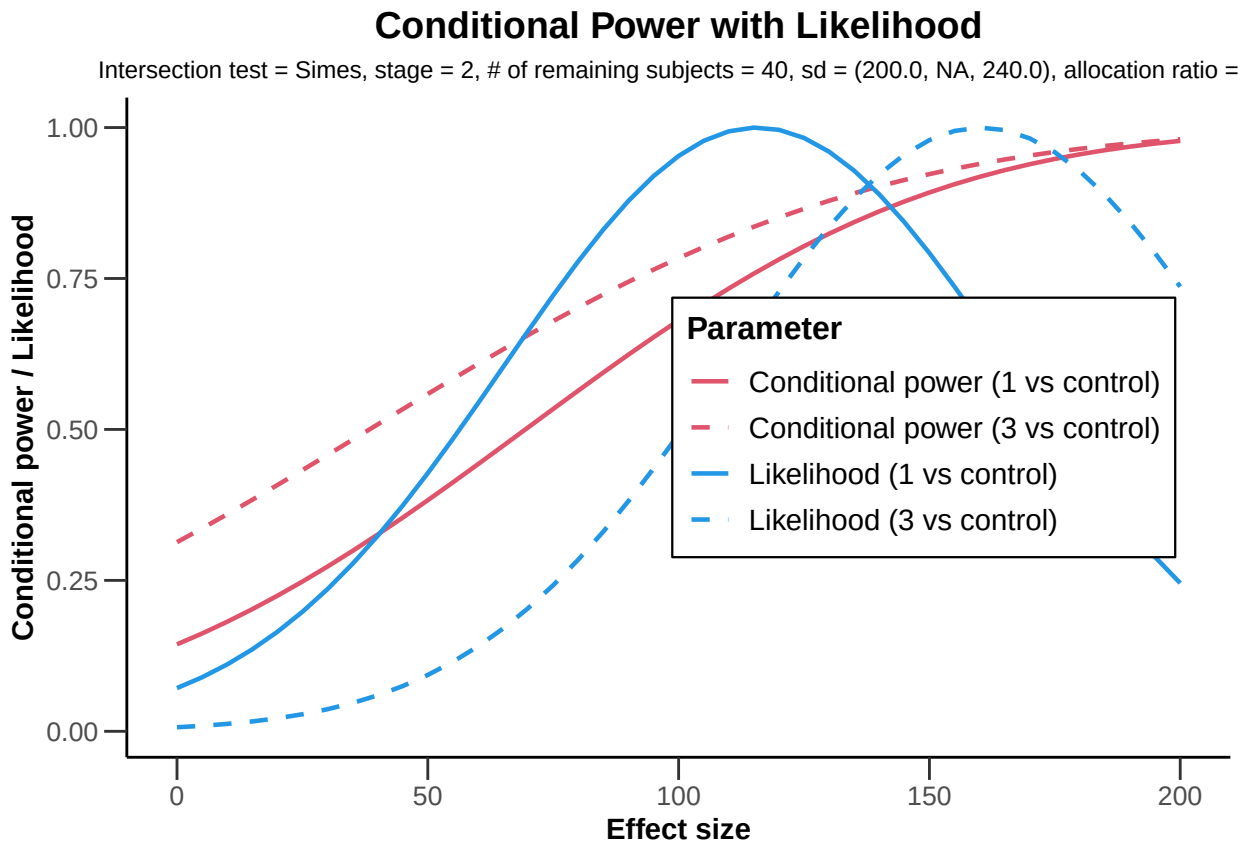
3 Analysis results multi-arm

3.1 Analysis results multi-arm - means

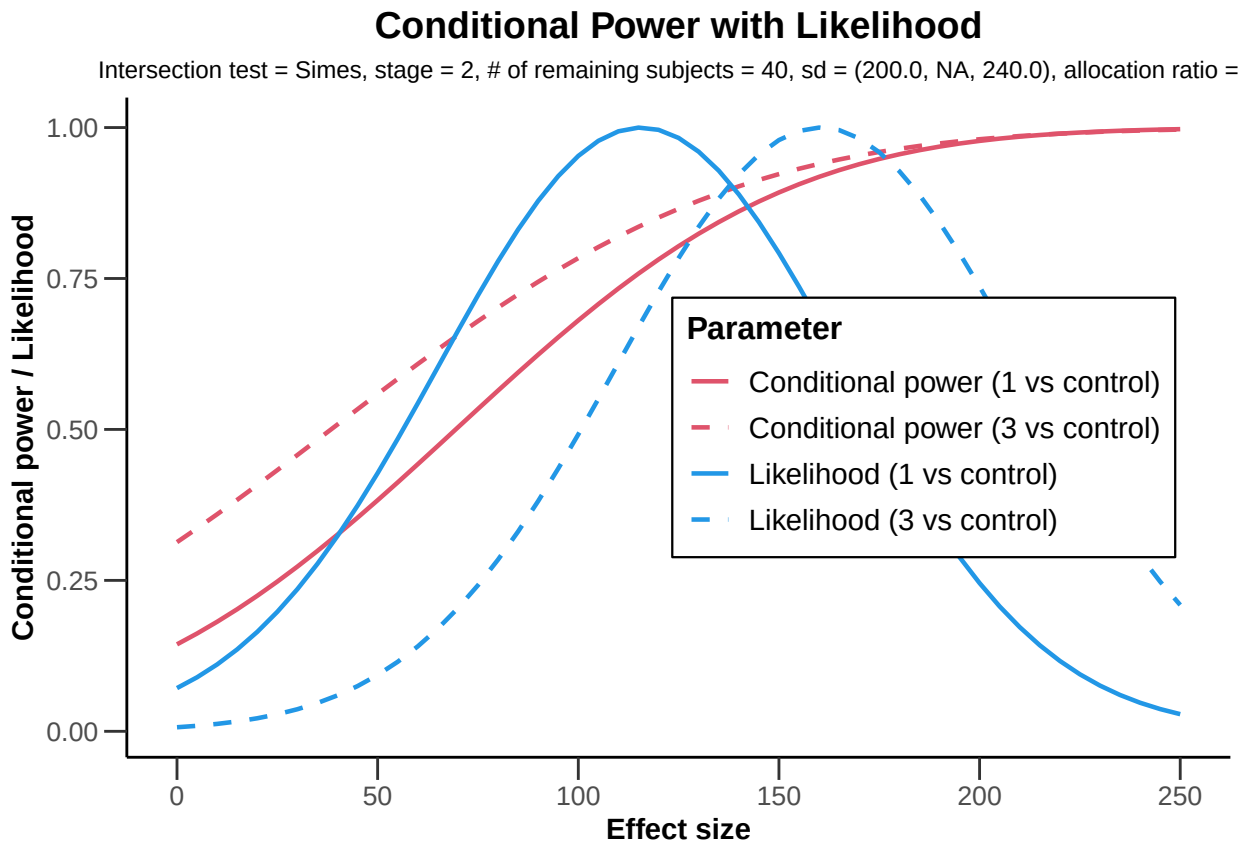
```
dataExampleMeans <- getDataset(
    n1 = c(13, 25),
    n2 = c(15, NA),
    n3 = c(14, 27),
    n4 = c(12, 29),
    means1 = c(242, 222),
    means2 = c(188, NA),
    means3 = c(267, 277),
    means4 = c(92, 122),
    stDevs1 = c(244, 221),
    stDevs2 = c(212, NA),
    stDevs3 = c(256, 232),
    stDevs4 = c(215, 227))

x <- getAnalysisResults(design = designIN, dataInput = dataExampleMeans,
    intersectionTest = "Simes", directionUpper = TRUE,
    varianceOption = "notPooled", nPlanned = c(32, 8),
    assumedStDevs = c(200, NA, 240))

plot(x)
```



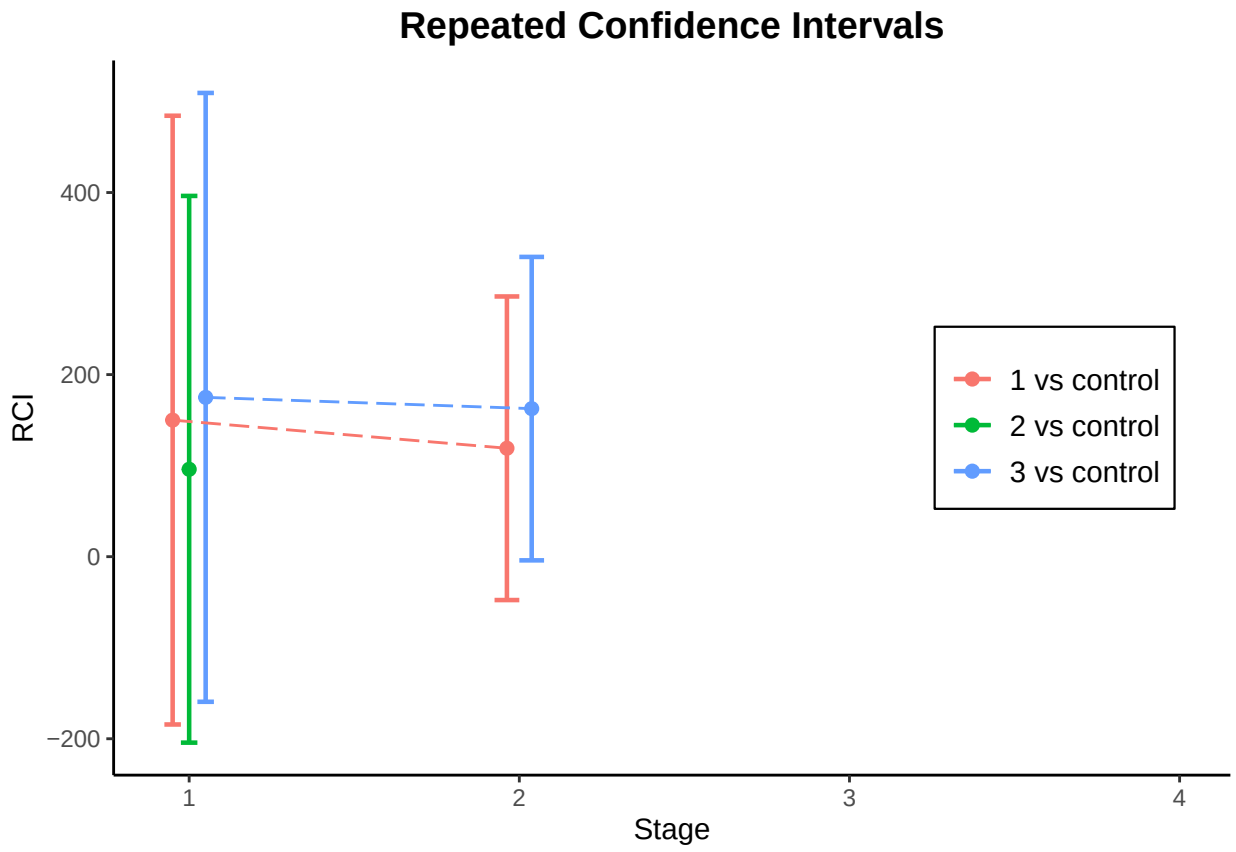

```
plot(x, nPlanned = c(32, 8), thetaRange = seq(0, 200, 5),
    assumedStDevs = c(200, NA, 240), treatmentArms = c(1, 3))
```



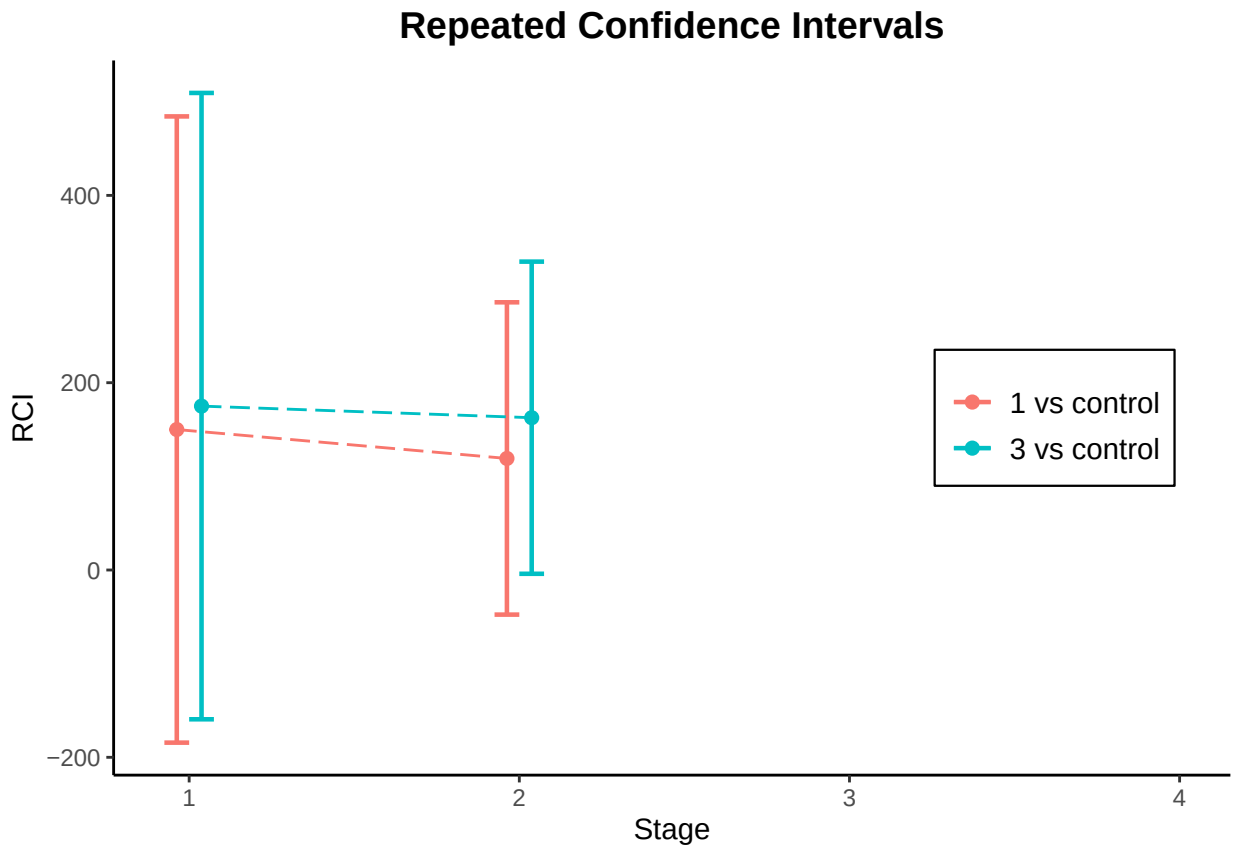

```
plot(x, nPlanned = c(32, 8), thetaRange = c(0, 250),
    assumedStDevs = c(200, NA, 240), treatmentArms = c(1, 3))
```


Conditional Power with Likelihood
Intersection test = Simes, stage = 2, # of remaining subjects = 40, sd = (200.0, NA, 240.0), allocation ratio =
Conditional power / Likelihood
1.00
0.75
0.50
0.25
0.00
0
50
100
150
200
250
Effect size
Parameter
Conditional power (1 vs control)
Conditional power (3 vs control)
Likelihood (1 vs control)
Likelihood (3 vs control)

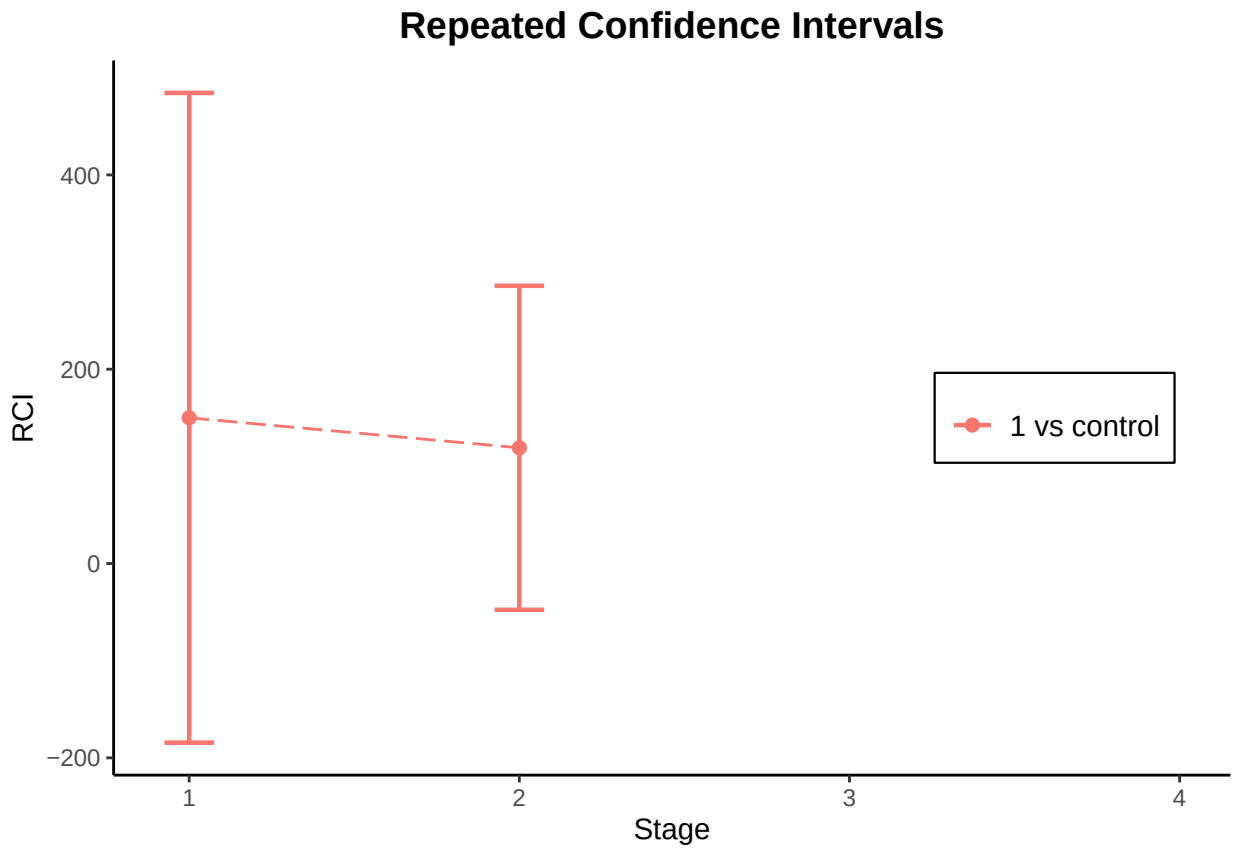
```
plot(x, type = 2)
```




```
plot(x, type = 2, treatmentArms = c(1, 3))
```

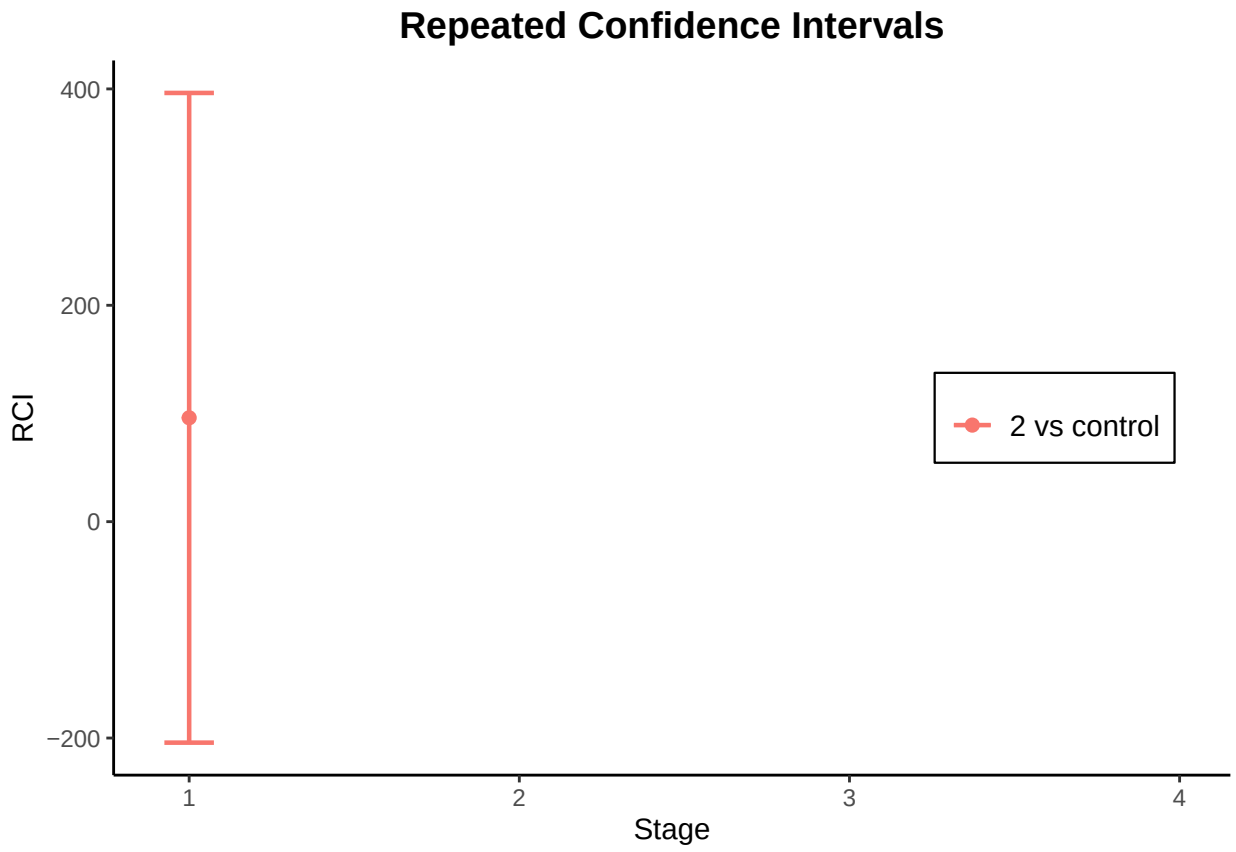



```
plot(x, type = 2, treatmentArms = c(1))
```

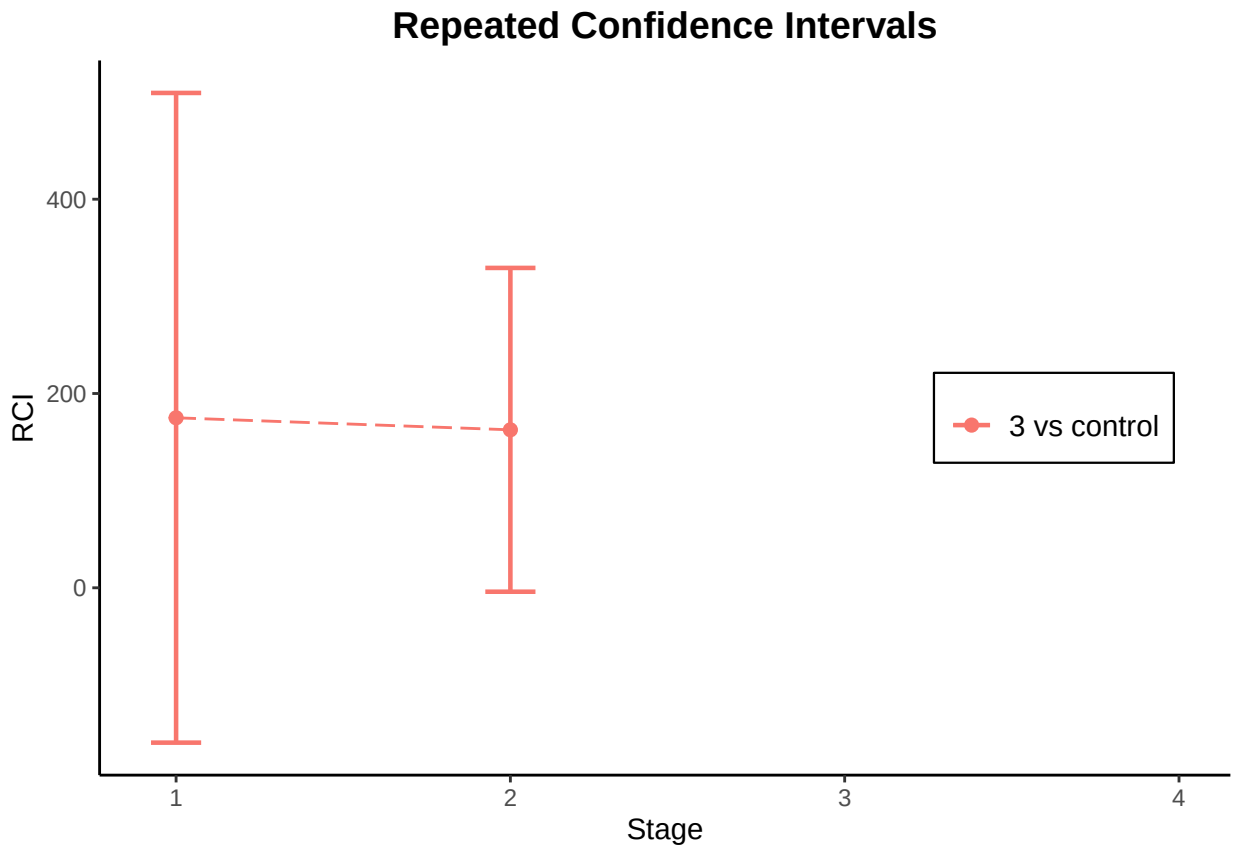



```
plot(x, type = 2, treatmentArms = c(2))
```

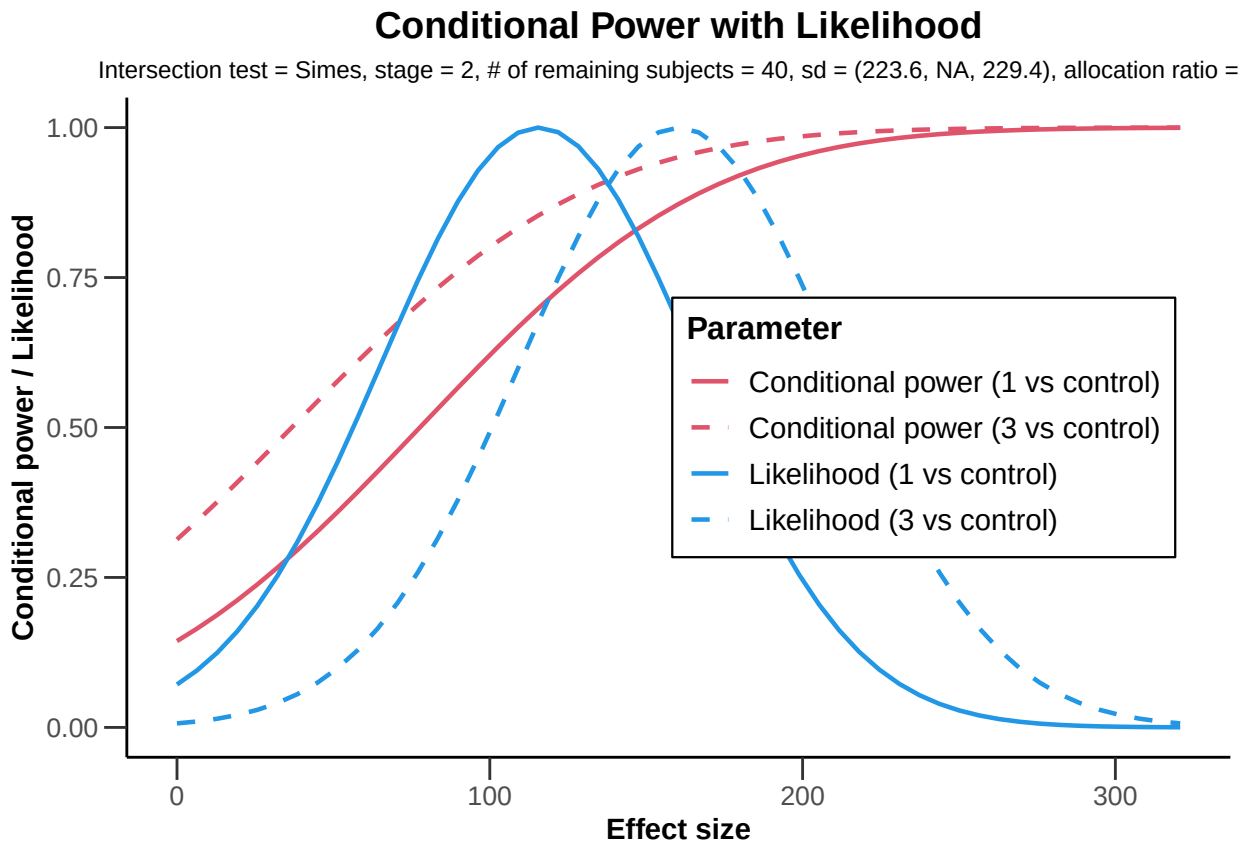
```
## geom_path: Each group consists of only one observation. Do you need to adjust the group aesthetic?
```




```
plot(x, type = 2, treatmentArms = c(3))
```

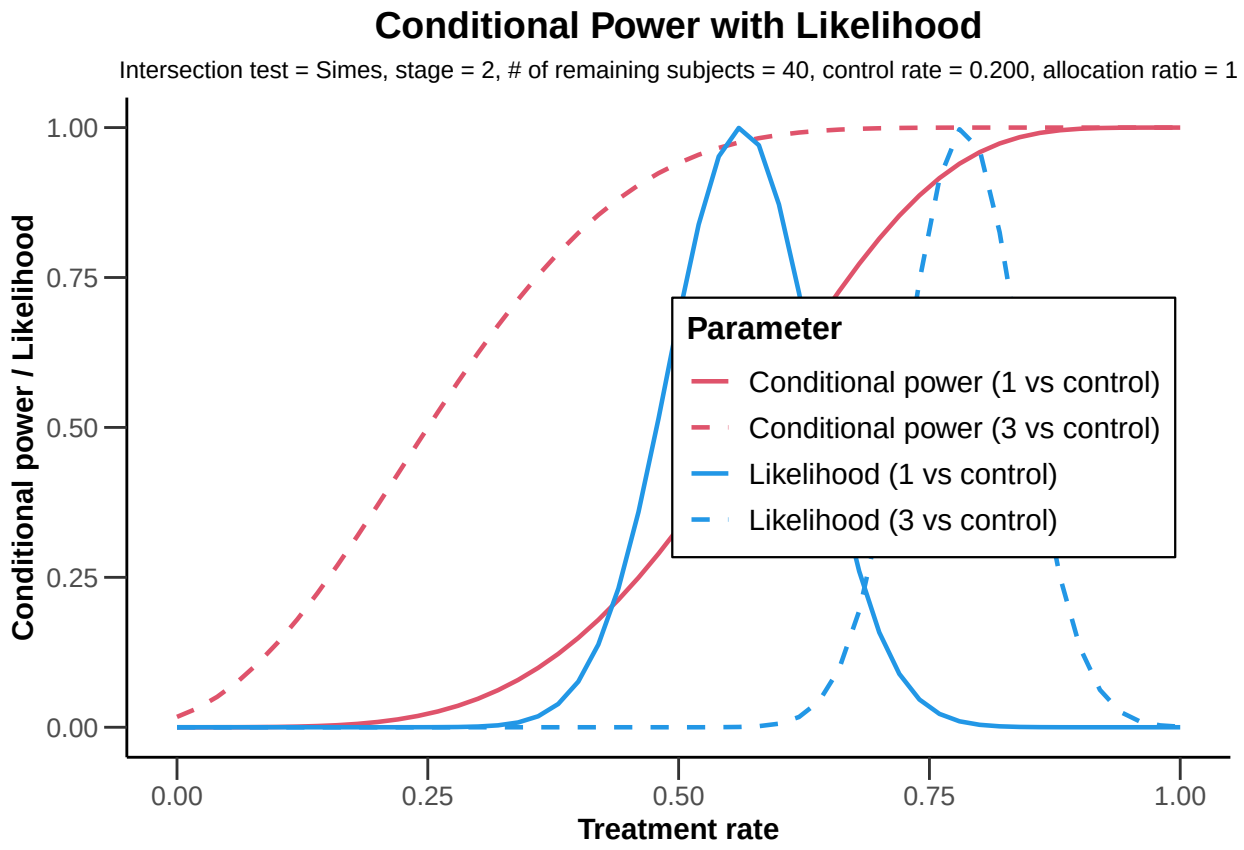



```
x2 <- getAnalysisResults(design = designIN, dataInput = dataExampleMeans,
    intersectionTest = "Simes", directionUpper = TRUE,
    varianceOption = "notPooled", nPlanned = c(32, 8))
# Observed standard deviations will be used
plot(x2)
```

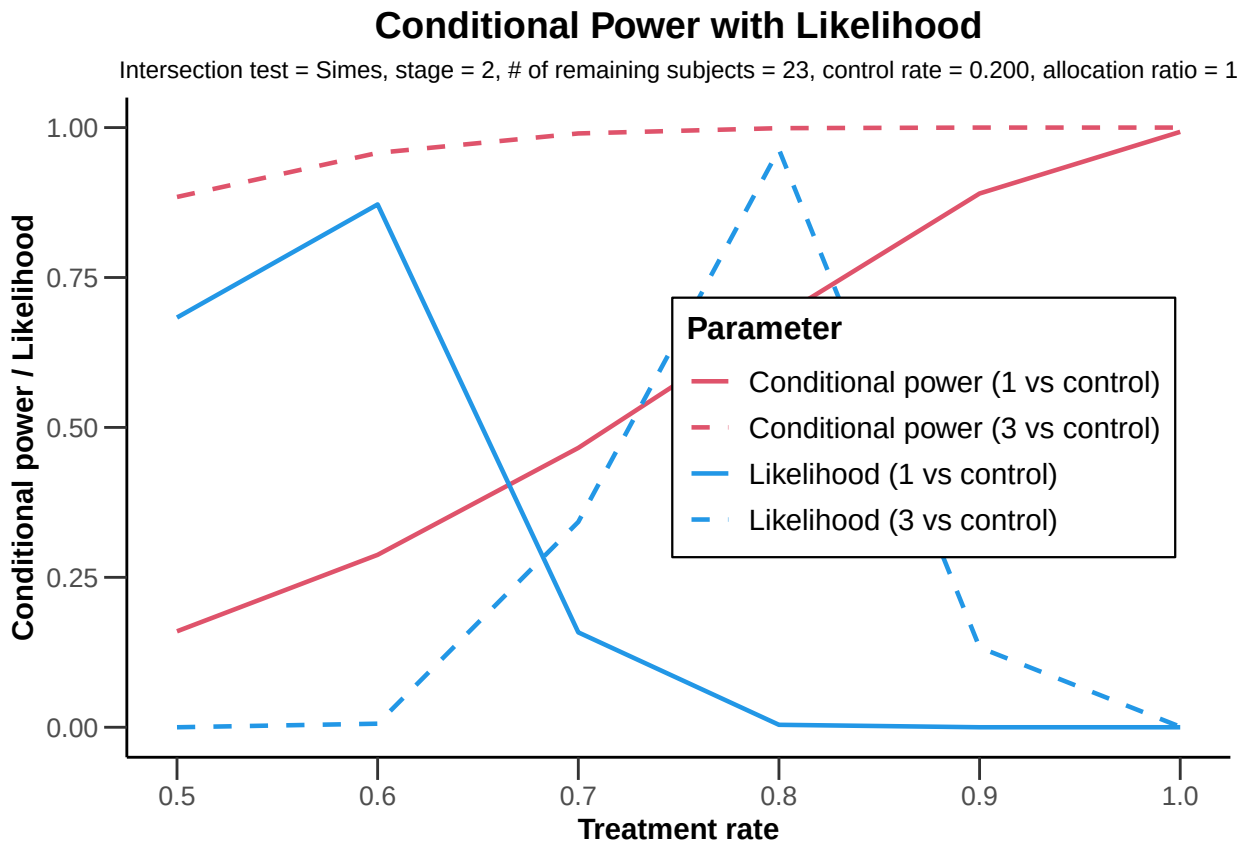


3.2 Analysis results multi-arm - rates

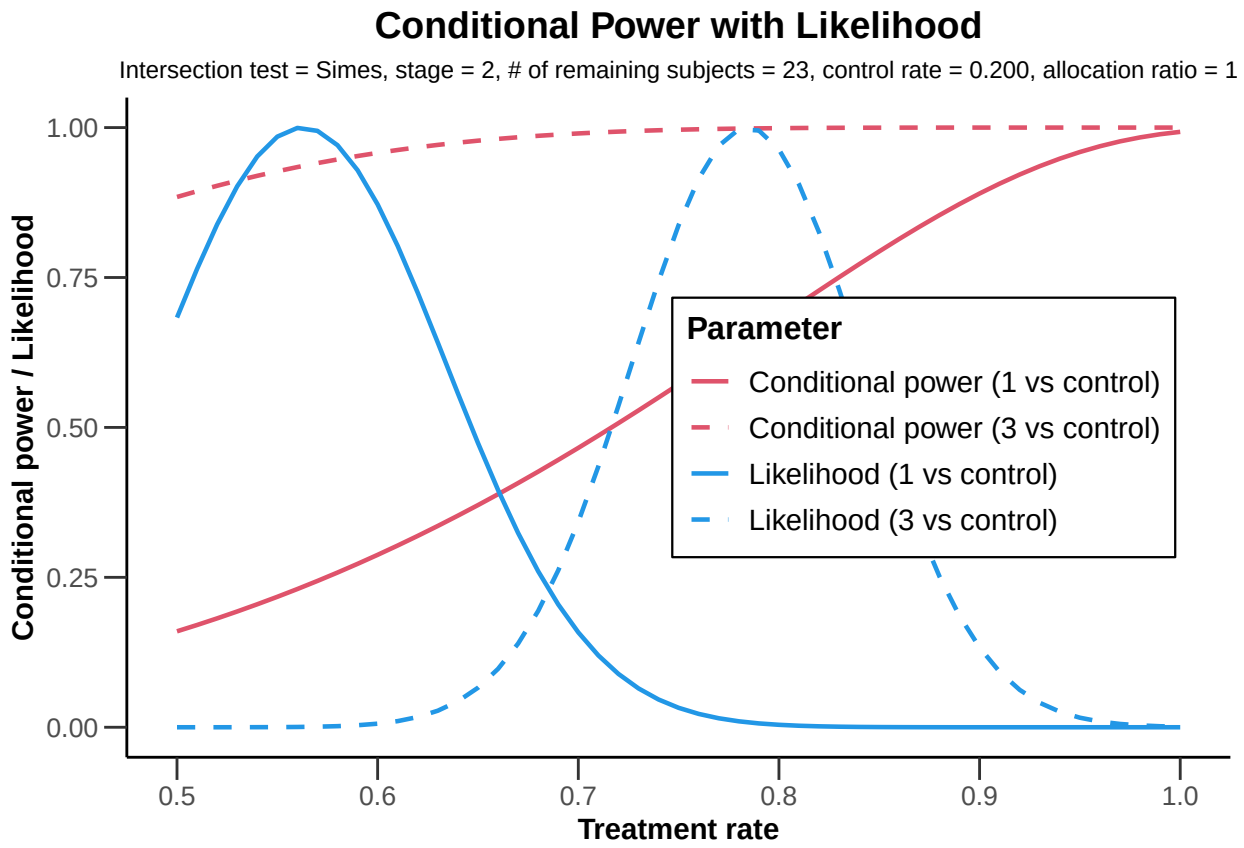
```
dataExampleRates <- getDataset(
    n1 = c(23, 25),
    n2 = c(25, NA),
    n3 = c(24, 27),
    n4 = c(22, 29),
    events1 = c(15, 12),
    events2 = c(19, NA),
    events3 = c(18, 22),
    events4 = c(12, 13))

analysisResultsRates <- getAnalysisResults(design = designIN,
    dataInput = dataExampleRates, intersectionTest = "Simes",
    nPlanned = c(20, 20), directionUpper = TRUE, piControl = 0.2)

plot(analysisResultsRates)
```



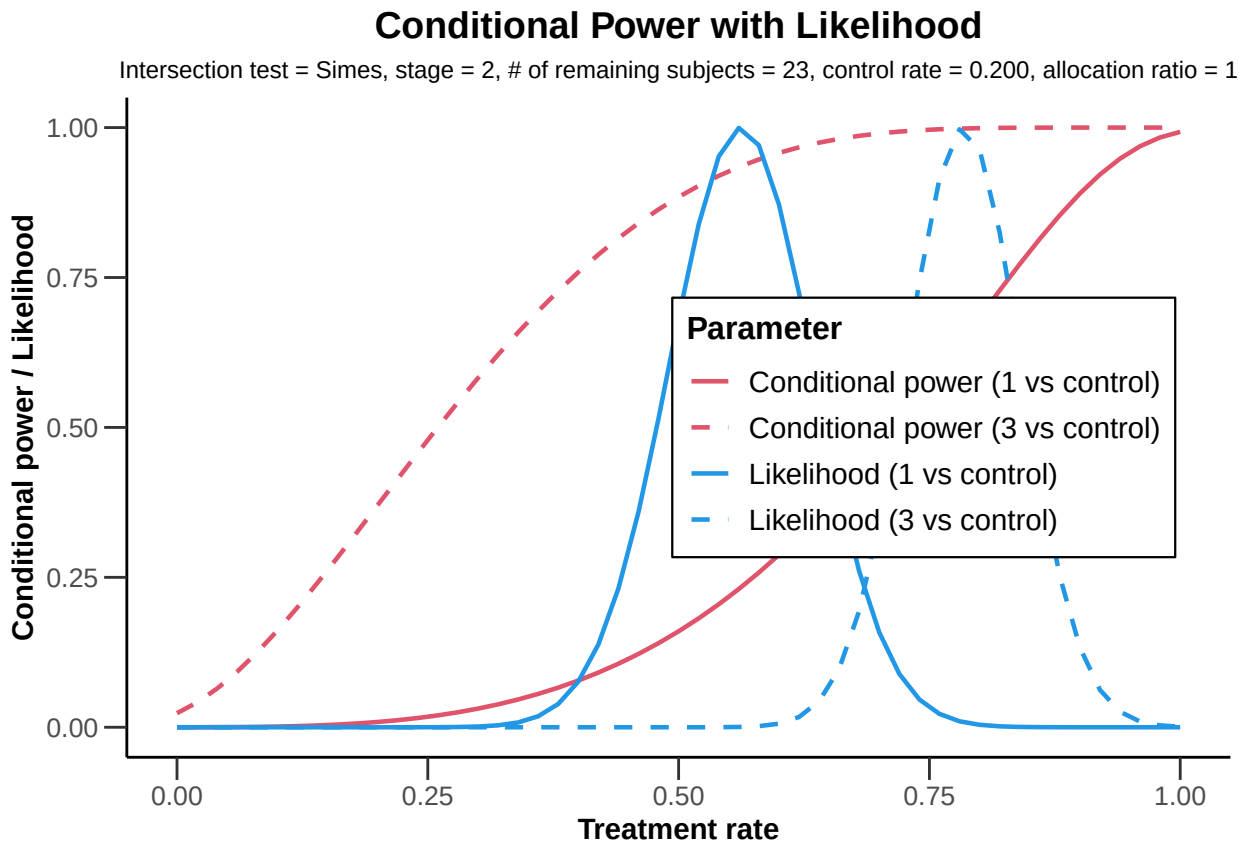

```
plot(analysisResultsRates, nPlanned = c(20, 3), piTreatmentRange = seq(0.5, 1, 0.1))
```



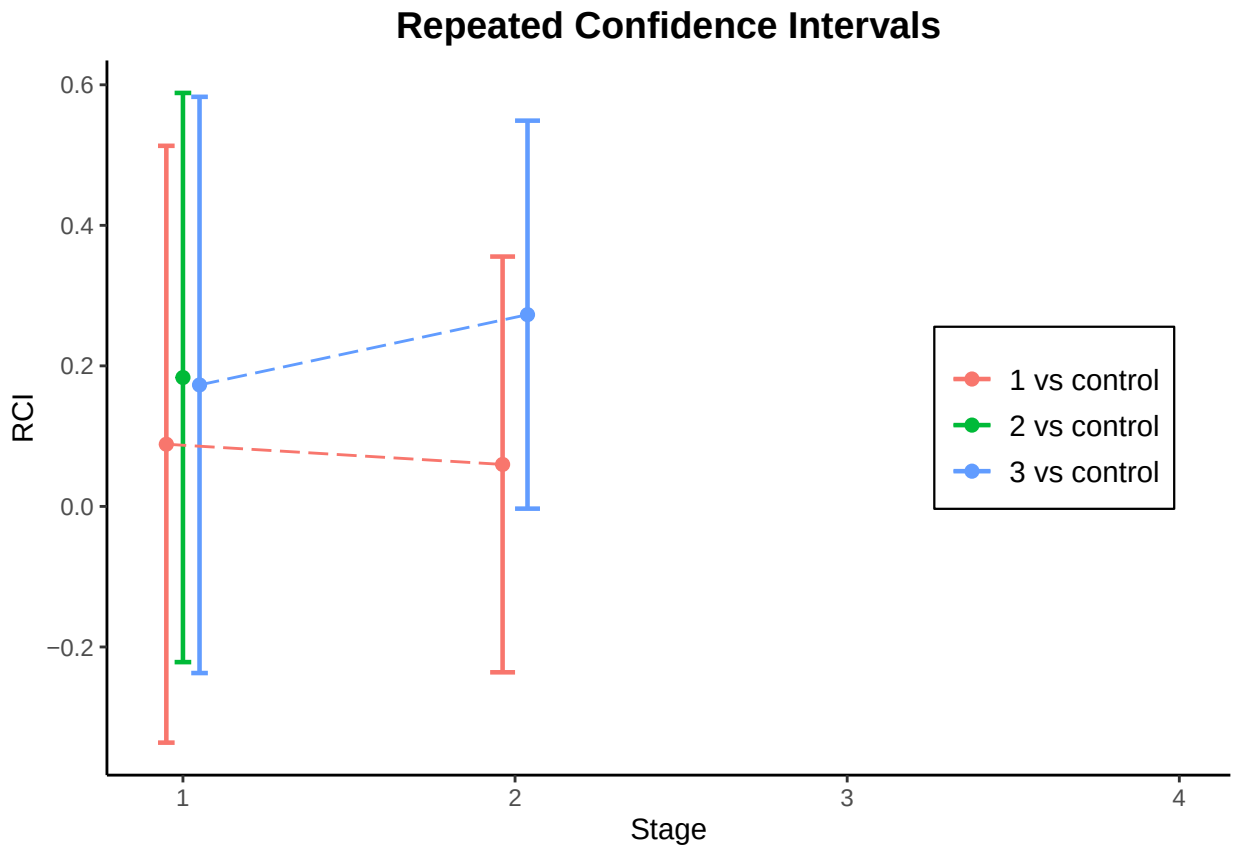

```
plot(analysisResultsRates, nPlanned = c(20, 3), piTreatmentRange = c(0.5, 1))
```




```
plot(analysisResultsRates, nPlanned = c(20, 3))
```

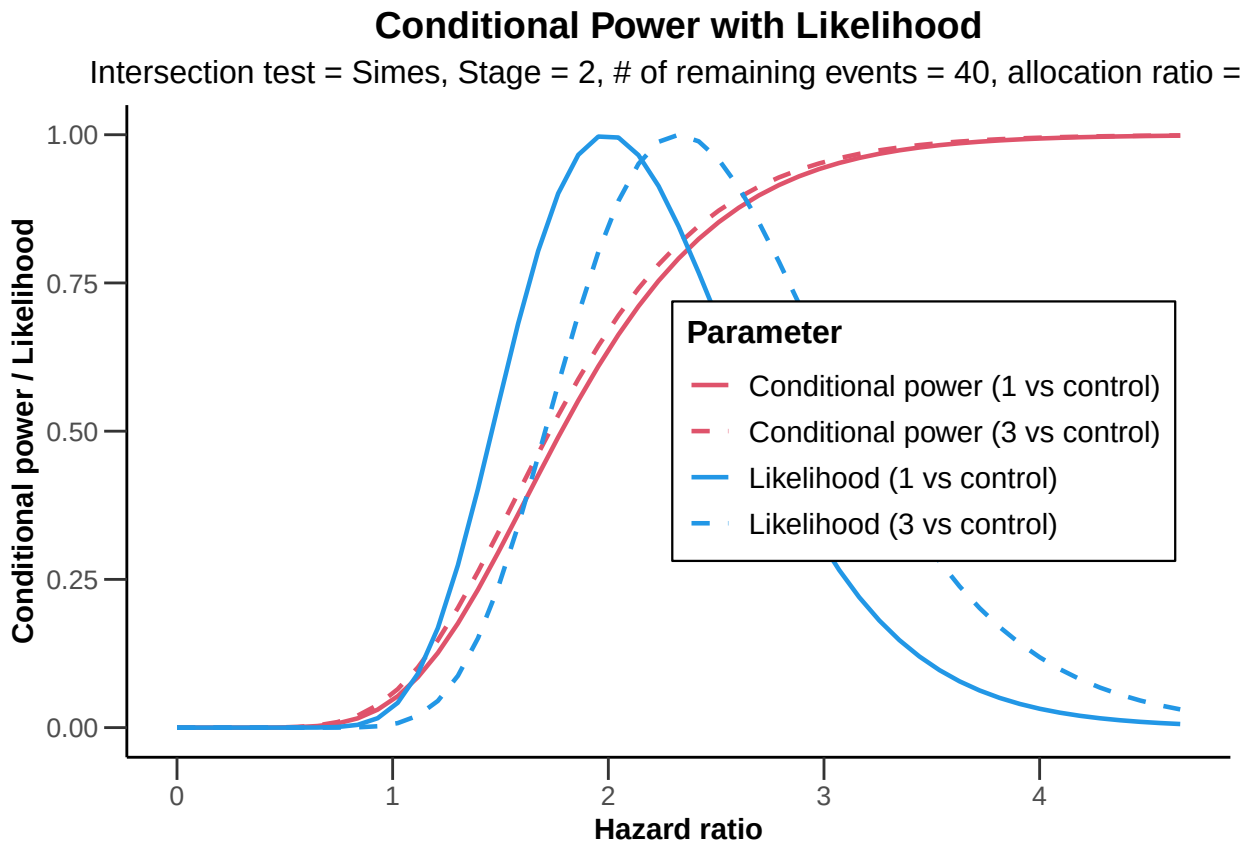



```
plot(analysisResultsRates, type = 2)
```

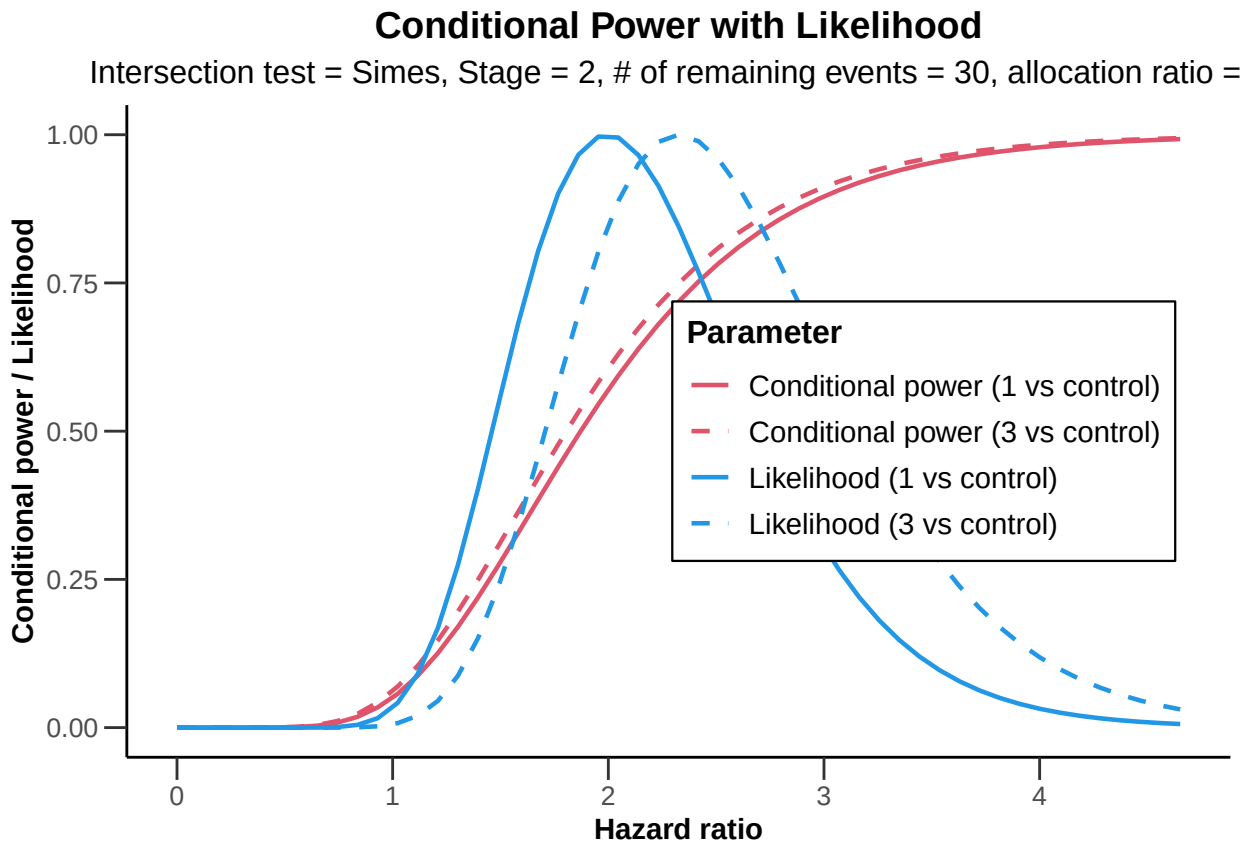


3.3 Analysis results multi-arm - survival

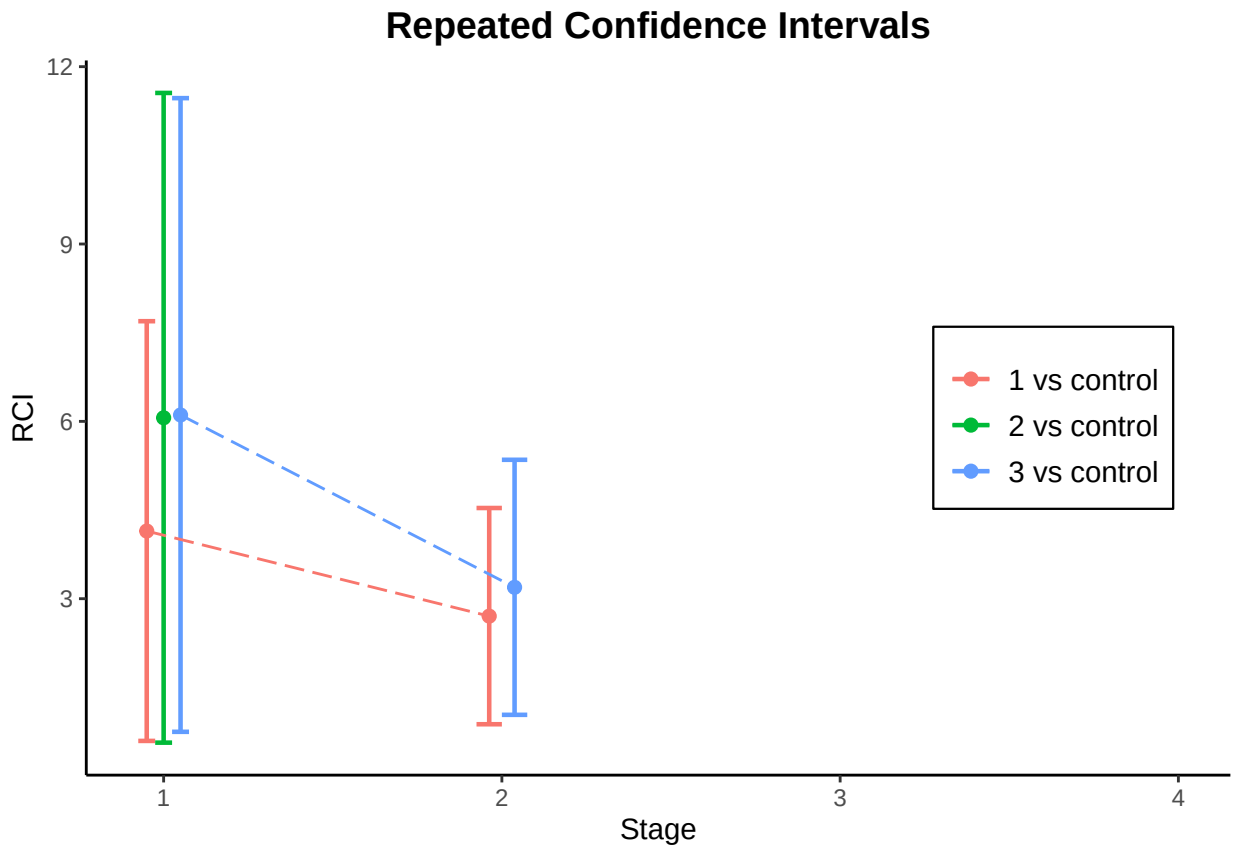
```
dataExampleSurvival <- getDataset(
    events1 = c(25, 32),
    events2 = c(18, NA),
    events3 = c(22, 36),
    logRanks1 = c(1.9,1.8),
    logRanks2 = c(1.99, NA),
    logRanks3 = c(2.52, 2.11)
)

analysisResultsSurvival <- getAnalysisResults(design = designIN,
    dataInput = dataExampleSurvival, intersectionTest = "Simes",
    nPlanned = c(20, 20), thetaH0 = 1.2, directionUpper = TRUE)
plot(analysisResultsSurvival)
```




```
plot(analysisResultsSurvival, nPlanned = c(20, 10))
```




```
plot(analysisResultsSurvival, type = 2)
```



System: rpact 3.0.3, R version 4.0.3 (2020-10-10), platform: x86_64-w64-mingw32

To cite R in publications use:

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

To cite package 'rpact' in publications use:

Gernot Wassmer and Friedrich Pahlke (2020). rpact: Confirmatory Adaptive Clinical Trial Design and Analysis. R package version 3.0.3. https://www.rpact.org

License