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

Gernot Wassmer and Friedrich Pahlke

Last change: 26 Oktober, 2021

Contents

Summary 1

1 Preparation and design 1
  1.1 Create a design ................................................................. 1

2 Analysis results base 2
  2.1 Analysis results base - means ............................................. 2
  2.2 Analysis results base - rates ............................................... 5
  2.3 Analysis results base - survival ......................................... 11

3 Analysis results multi-arm 13
  3.1 Analysis results multi-arm - means ...................................... 13
  3.2 Analysis results multi-arm - rates ....................................... 22
  3.3 Analysis results multi-arm - survival .................................... 27

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

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

# [1] '3.1.1'

1.1 Create a design

```r
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

```r
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 shown that it is valid only if no sample size change was performed)

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

---

**Conditional Power with Likelihood**

Stage = 3, # of remaining subjects = 130, sd = 100

```
plot(x, type = 2)
```
2.1 Analysis results base - means

Repeated Confidence Intervals

```
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))
```
2.1 Analysis results base - means

Conditional Power with Likelihood
Stage = 2, # of remaining subjects = 50, sd = 166.32, allocation ratio = 1

```
plot(x, type = 2)
```
2.2 Analysis results base - rates

```r
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 shown that it is valid only if no sample size change was performed)

plot(x)

## Warning: Removed 1 row(s) containing missing values (geom_path).
```
Conditional Power with Likelihood
Stage = 3, # of remaining subjects = 10

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

plot(x, type = 2)
2.2 Analysis results base - rates

Repeated Confidence Intervals

```
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, $\pi_2 = 0.3$, allocation ratio = 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conditional power</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Plot](plot(x, type = 2))
2.3 Analysis results base - survival

```r
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 shown
## that the confidence interval is valid only if no sample size change was performed)
plot(x, thetaRange = c(1,3))
```
Conditional Power with Likelihood

Stage = 3, maximum number of remaining events = 20, allocation ratio = 1

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)
Conditional Power with Likelihood

Intersection test = Simes, stage = 2, # of remaining subjects = 40, sd = (200, NA, 240), allocation ratio = 1

Parameter
- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

```
plot(x, nPlanned = c(32, 8), thetaRange = seq(0, 200, 5),
     assumedStDevs = c(200, NA, 240), treatmentArms = c(1, 3))
```
3.1 Analysis results multi-arm - means

Conditional Power with Likelihood

Intersection test = Simes, stage = 2, # of remaining subjects = 40, sd = (200, NA, 240), allocation ratio = 1

Parameter
- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

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, NA, 240), allocation ratio = 1

![Conditional Power with Likelihood graph]

Parameter
- Red line: Conditional power (1 vs control)
- Dashed red line: Likelihood (1 vs control)
- Blue line: Conditional power (3 vs control)
- Dashed blue line: Likelihood (3 vs control)

plot(x, type = 2)
3.1 Analysis results multi-arm - means

Repeated Confidence Intervals

plot(x, type = 2, treatmentArms = c(1, 3))
3.1 Analysis results multi-arm - means

Repeated Confidence Intervals

plot(x, type = 2, treatmentArms = c(1))
Repeated Confidence Intervals

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

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

plot(x, type = 2, treatmentArms = c(3))
3.1 Analysis results multi-arm - means

Repeated Confidence Intervals

Stage

RCI

3 vs control

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)
### Conditional Power with Likelihood

Intersection test = Simes, stage = 2, # of remaining subjects = 40, control rate = 0.2, allocation ratio = 1

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

Intersection test = Simes, stage = 2, # of remaining subjects = 23, control rate = 0.2, allocation ratio

Parameter

- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

plot(analysisResultsRates, nPlanned = c(20, 3), piTreatmentRange = c(0.5, 1))
Conditional Power with Likelihood

Intersection test = Simes, stage = 2, # of remaining subjects = 23, control rate = 0.2, allocation ratio

Parameter
- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

plot(analysisResultsRates, nPlanned = c(20, 3))
3.2 Analysis results multi-arm - rates

Conditional Power with Likelihood

Intersection test = Simes, stage = 2, # of remaining subjects = 23, control rate = 0.2, allocation ratio = 1

plot(analysisResultsRates, type = 2)
3.3 Analysis results multi-arm - survival

```r
# Example survival data
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)

# Analysis results
analysisResultsSurvival <- getAnalysisResults(design = designIN,
                                           dataInput = dataExampleSurvival, intersectionTest = "Simes",
                                           nPlanned = c(20, 20), thetaH0 = 1.2, directionUpper = TRUE)
plot(analysisResultsSurvival)
```
3.3 Analysis results multi-arm - survival

Conditional Power with Likelihood
Intersection test = Simes, Stage = 2, # of remaining events = 40, allocation ratio =

Parameter
- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

plot(analysisResultsSurvival, nPlanned = c(20, 10))
Conditional Power with Likelihood

Intersection test = Simes, Stage = 2, # of remaining events = 30, allocation ratio =

Parameter
- Conditional power (1 vs control)
- Likelihood (1 vs control)
- Conditional power (3 vs control)
- Likelihood (3 vs control)

plot(analysisResultsSurvival, type = 2)
System: rpact 3.1.1, R version 4.1.0 (2021-05-18), platform: x86_64-w64-mingw32
To cite R in publications use:
To cite package ‘rpact’ in publications use:

License

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/.