How to Create One- and Multi-Arm Simulation Result Plots with rpact

Friedrich Pahlke and Gernot Wassmer

Last change: 15 November, 2022

Contents

Summary

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

1 Preparation

First, load the rpact package

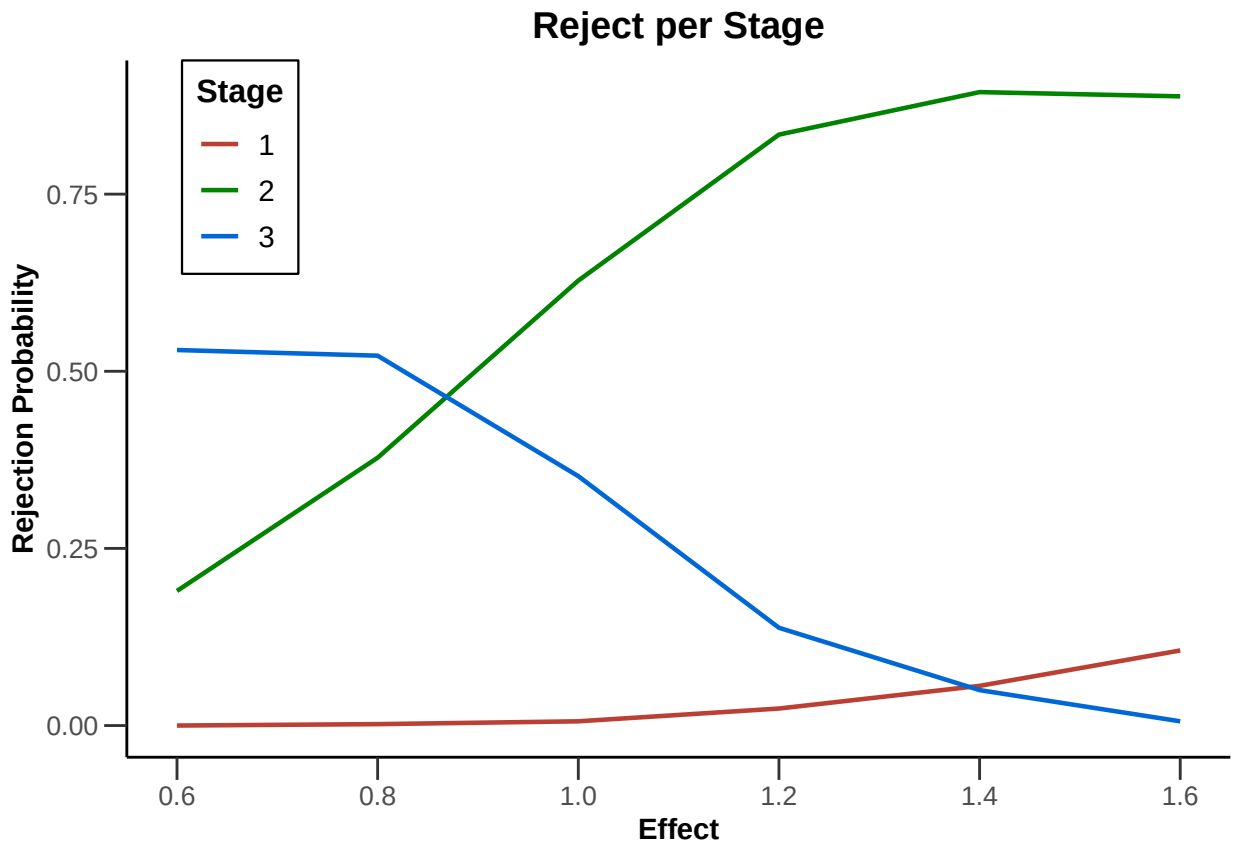
```
library(rpact)
packageVersion("rpact") # version should be version 3.0 or later
```

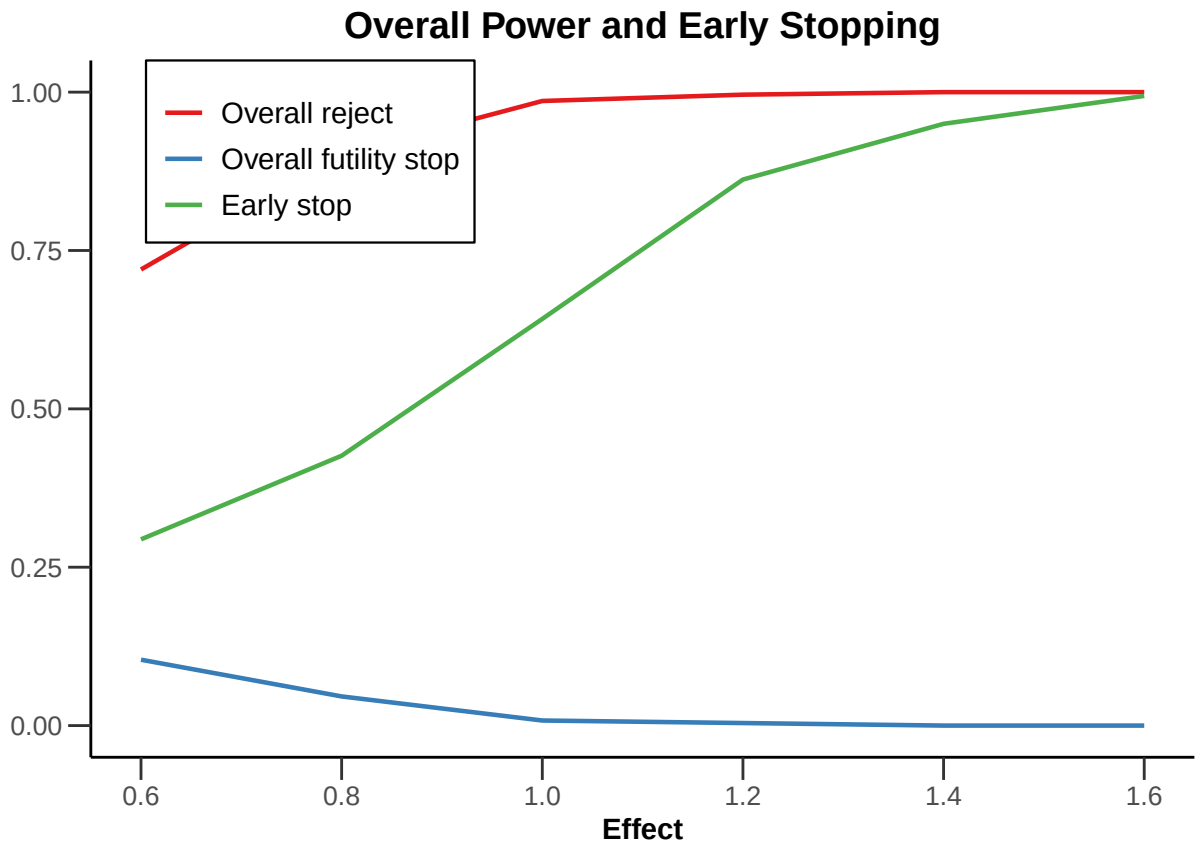
```
## [1] '3.3.2'
```

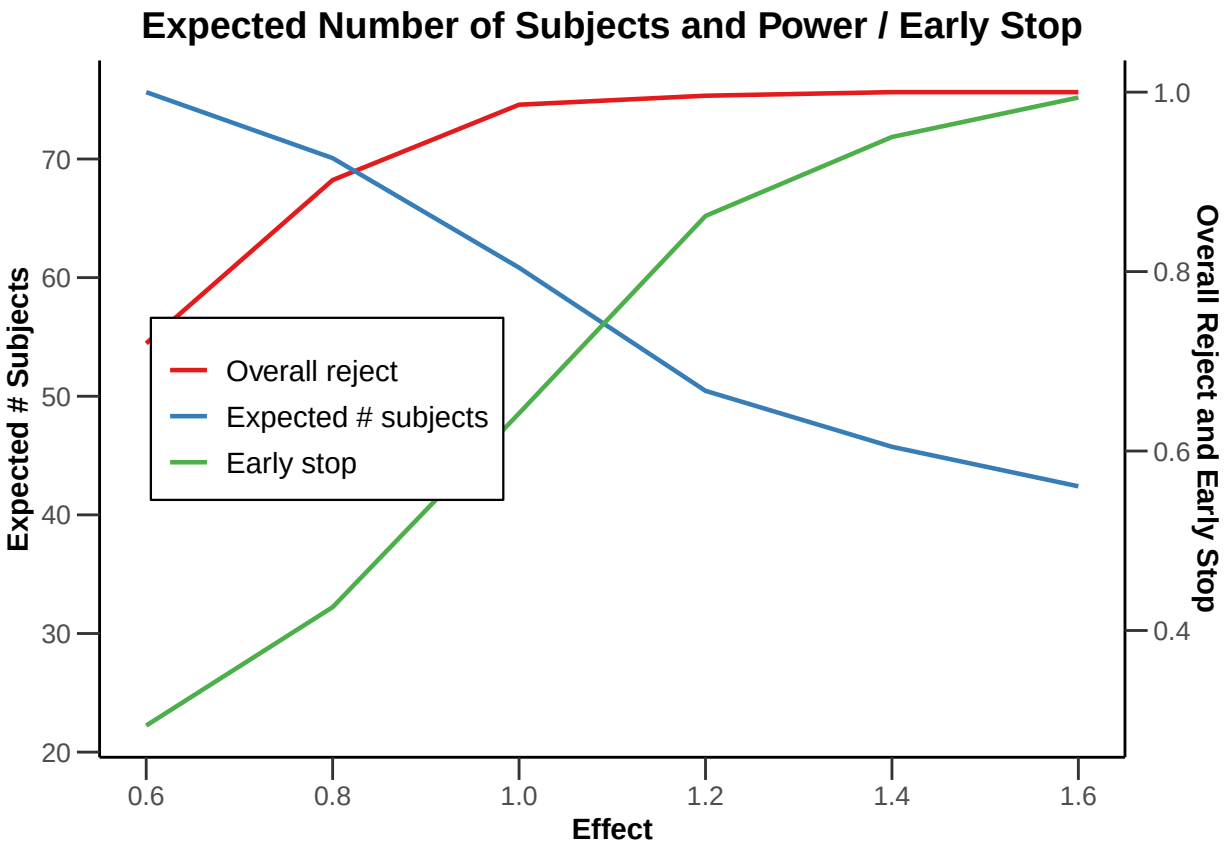
2 Simulation results base

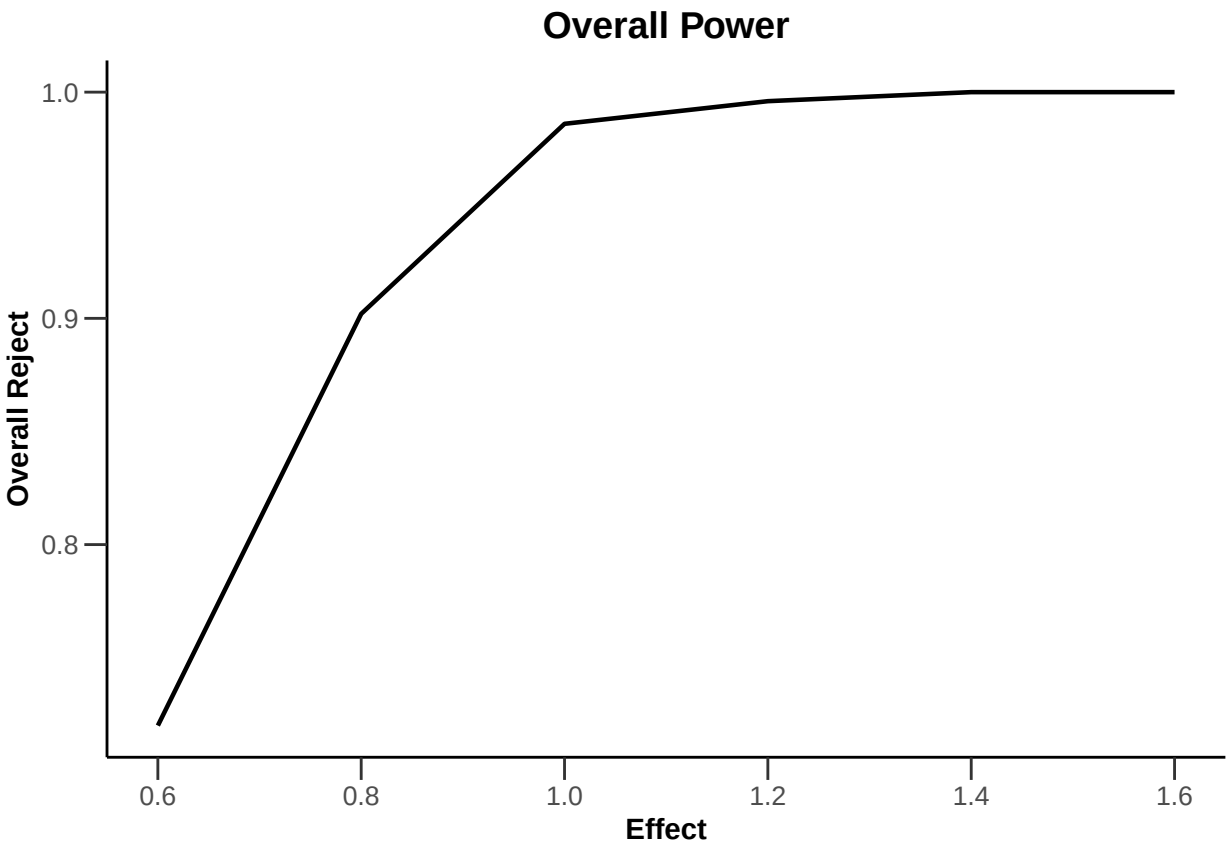
2.1 Simulation results base - means

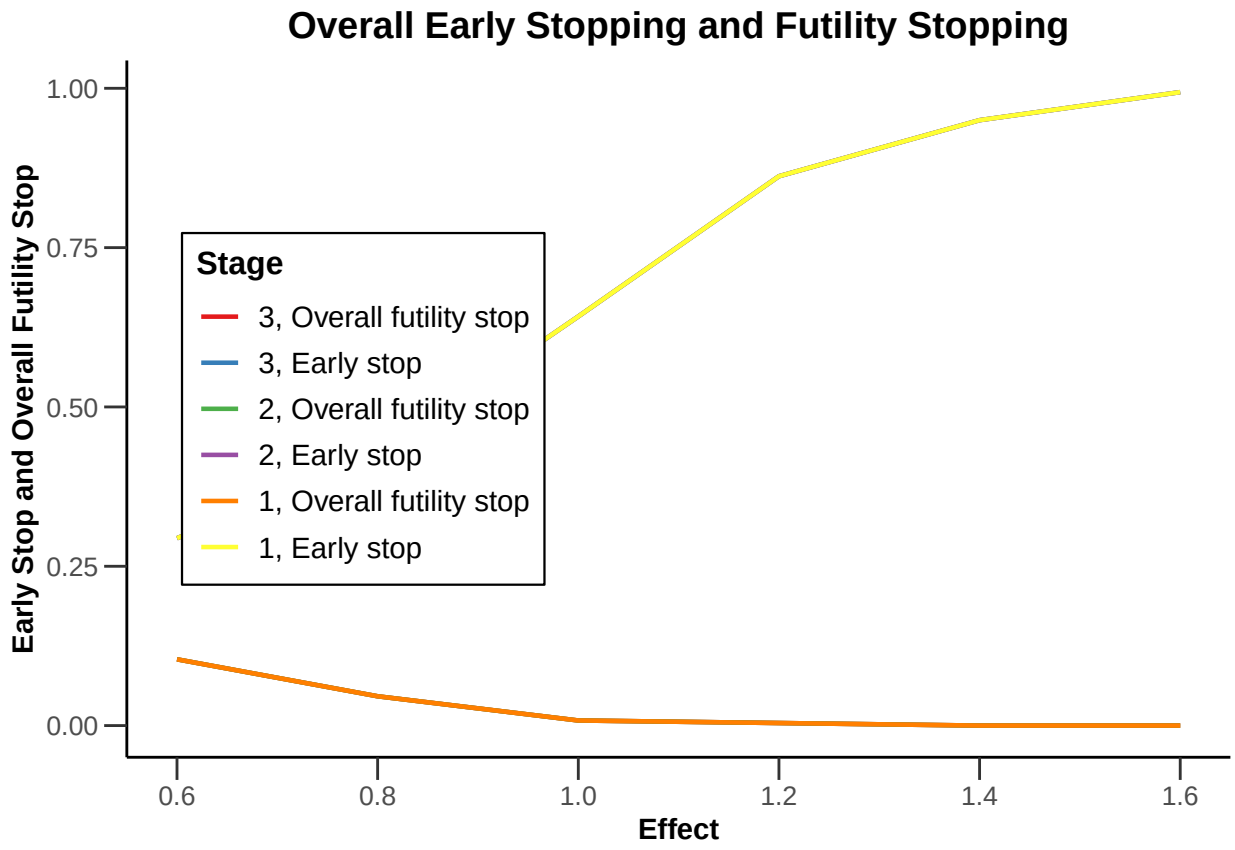
```
maxNumberOfSubjects <- 90
informationRates <- c(0.2, 0.5, 1)
plannedSubjects <- round(informationRates * maxNumberOfSubjects)
design <- getDesignInverseNormal(
    futilityBounds = c(-0.5, 0.5),
    informationRates = informationRates
)
x <- getSimulationMeans(
    design = design, groups = 2, meanRatio = TRUE,
    thetaH0 = 0.4, plannedSubjects = plannedSubjects,
    maxNumberOfIterations = 500, allocationRatioPlanned = 3,
    stDev = 1.5, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

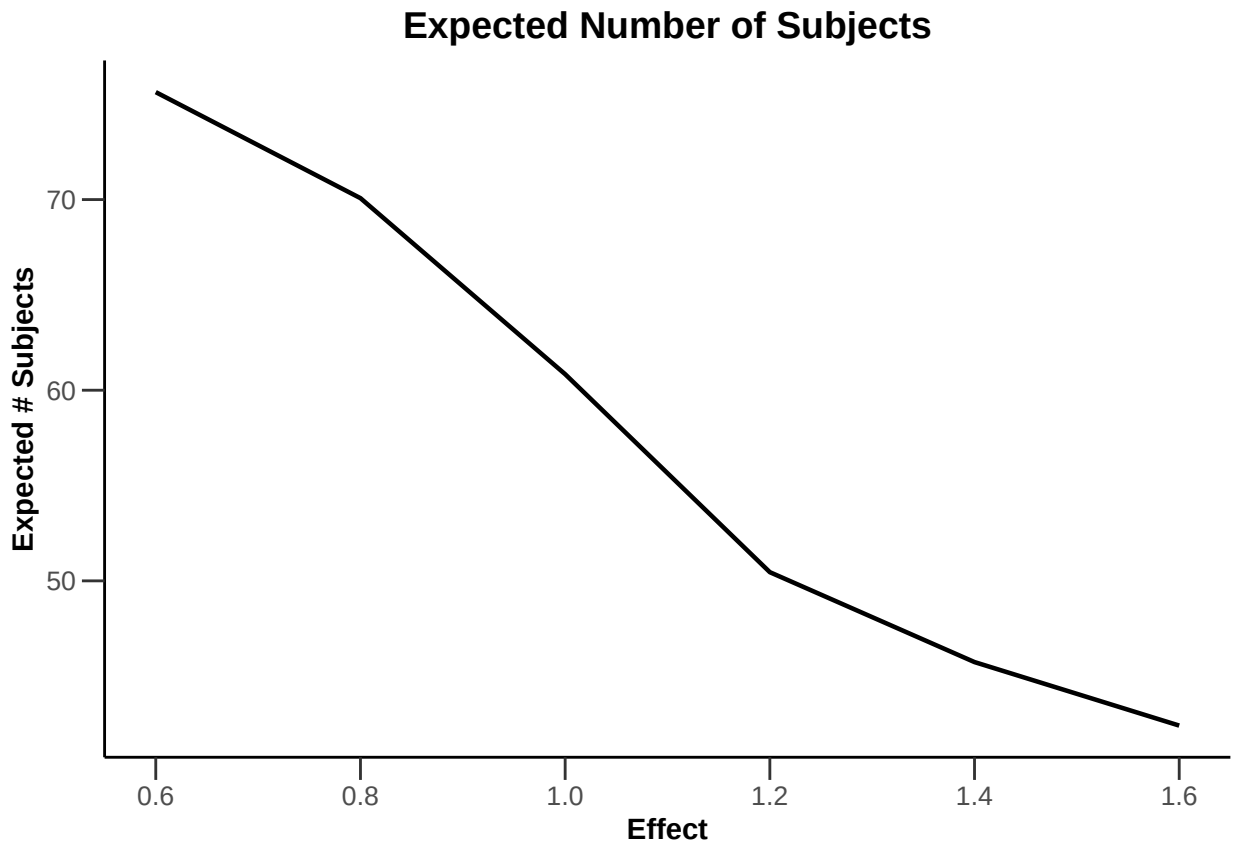



Overall Power and Early Stopping
Overall reject
Overall futility stop
Early stop
1.00
0.75
0.50
0.25
0.00
0.6
0.8
1.0
1.2
1.4
1.6
Effect


Expected Number of Subjects and Power / Early Stop
Expected # Subjects
Overall Reject and Early Stop
Effect
Overall reject
Expected # subjects
Early stop
20
30
40
50
60
70
0.4
0.6
0.8
1.0
0.6
0.8
1.0
1.2
1.4
1.6

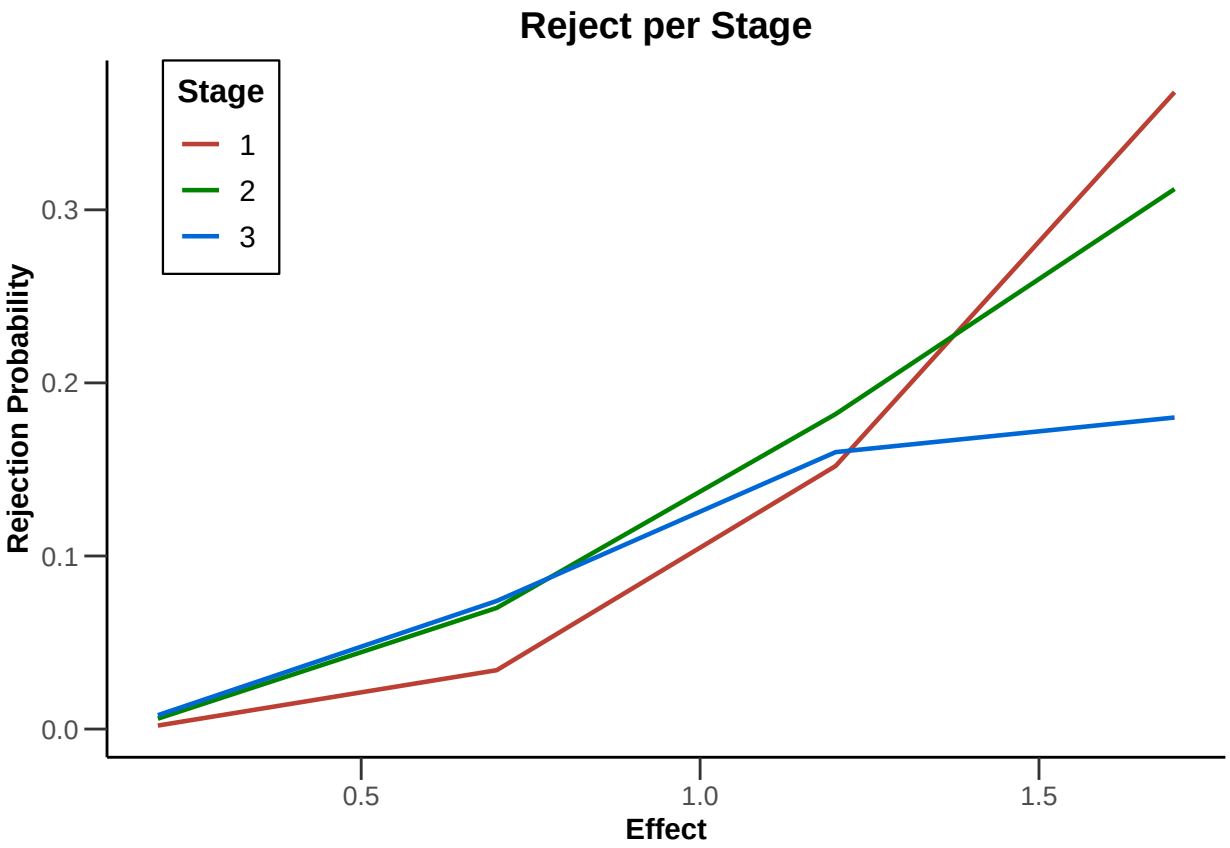

Overall Power
Overall Reject
1.0
0.9
0.8
0.6
0.8
1.0
1.2
1.4
1.6
Effect

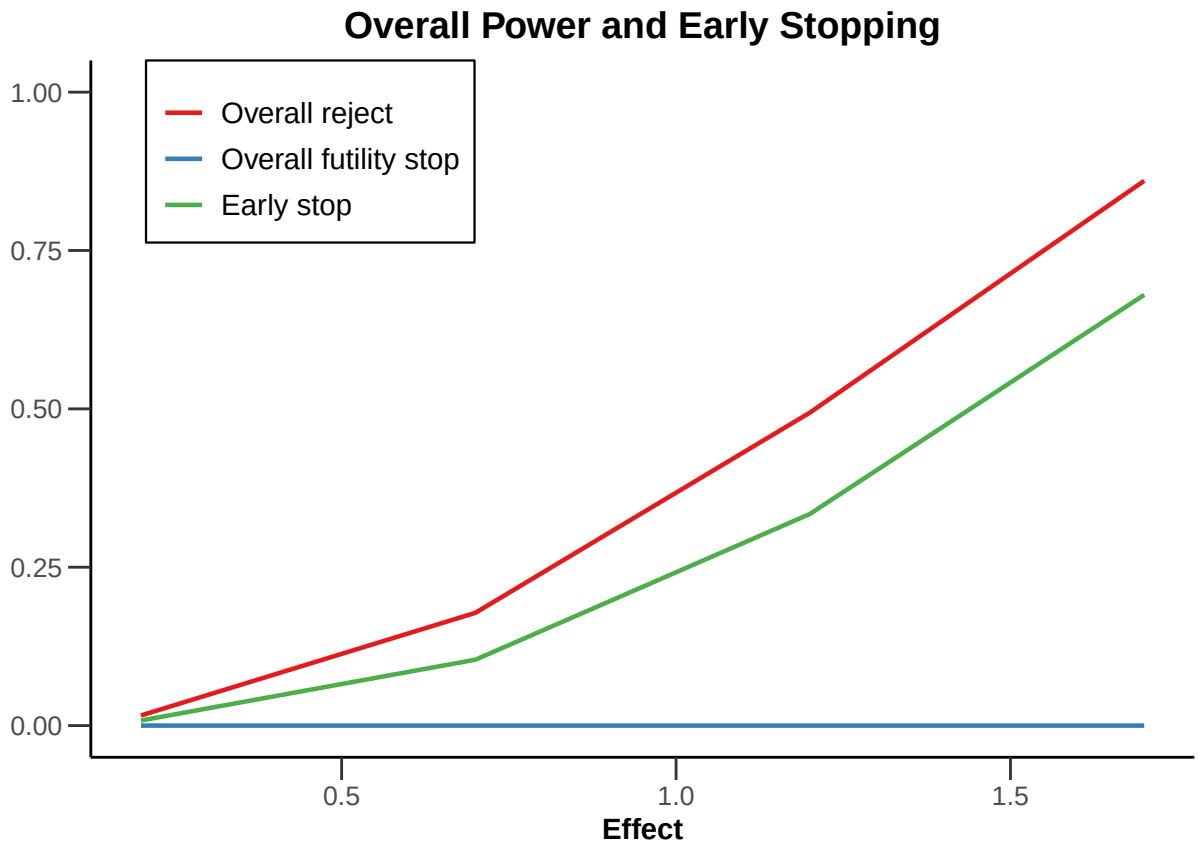

Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
1.00
0.75
0.50
0.25
0.00
Stage
3, Overall futility stop
3, Early stop
2, Overall futility stop
2, Early stop
1, Overall futility stop
1, Early stop
0.6
0.8
1.0
1.2
1.4
1.6
Effect

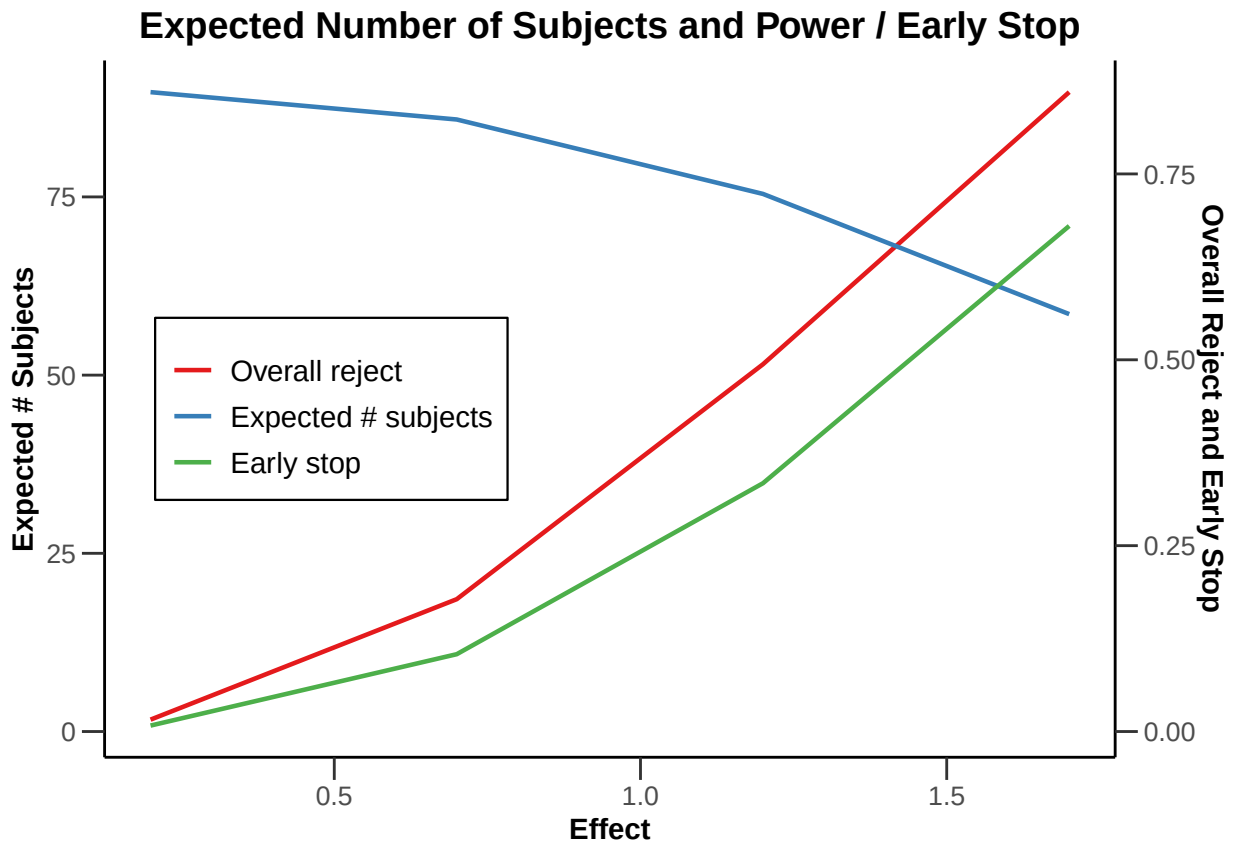


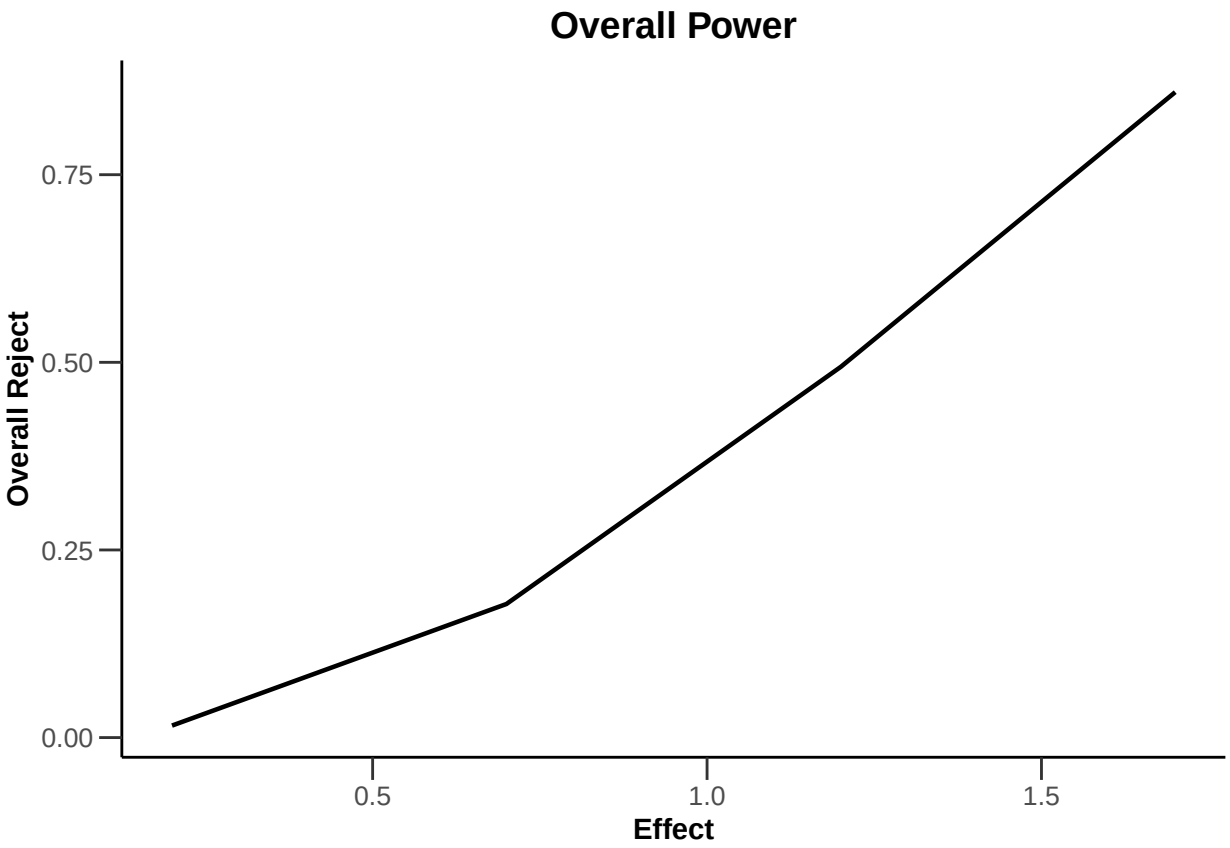
2.2 Simulation results base - rates

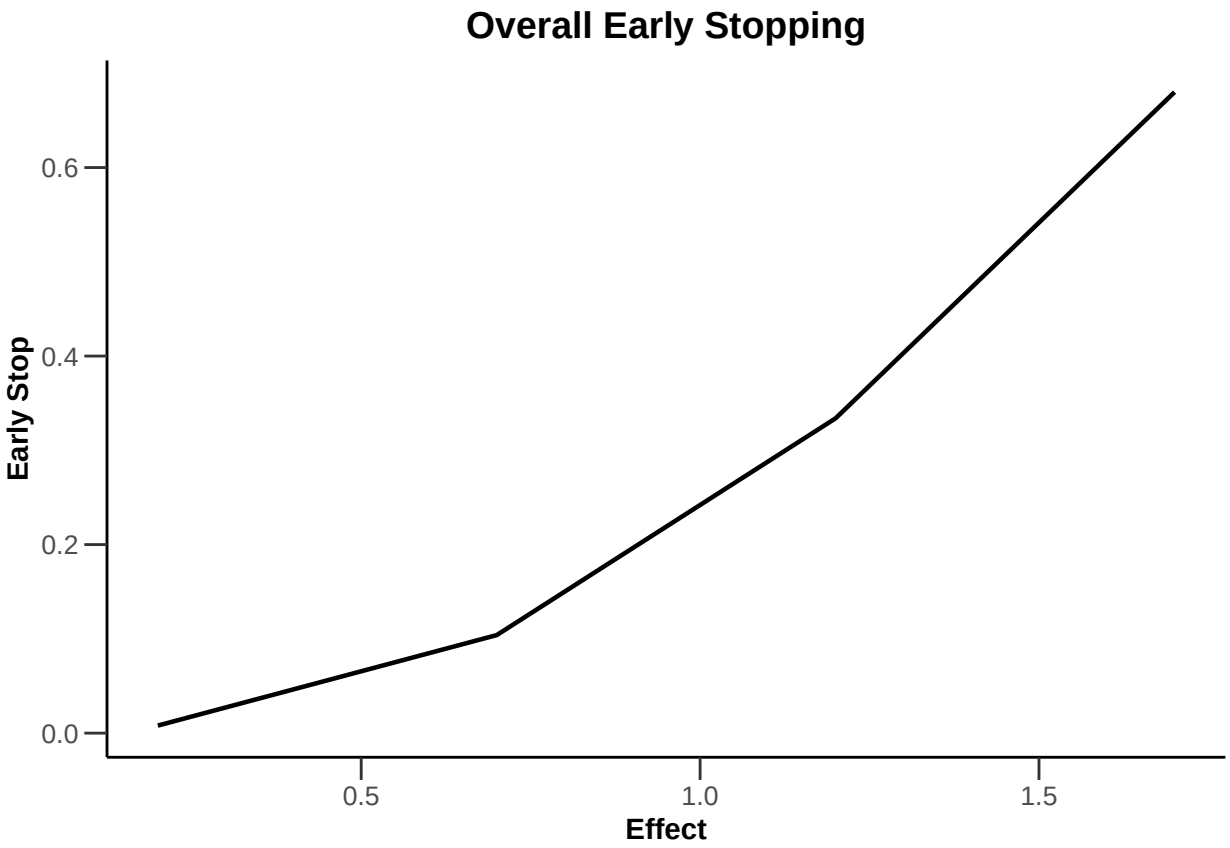
```
maxNumberOfSubjects <- 90
informationRates <- (1:3) / 3
plannedSubjects <- round(informationRates * maxNumberOfSubjects)
design <- getDesignInverseNormal(
    futilityBounds = c(-0.5, 0.5),
    informationRates = informationRates
)
x <- getSimulationRates(
    design = getDesignFisher(),
    groups = 2, riskRatio = TRUE,
    thetaH0 = 0.8, plannedSubjects = plannedSubjects,
    maxNumberOfIterations = 500, allocationRatioPlanned = 3,
    seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

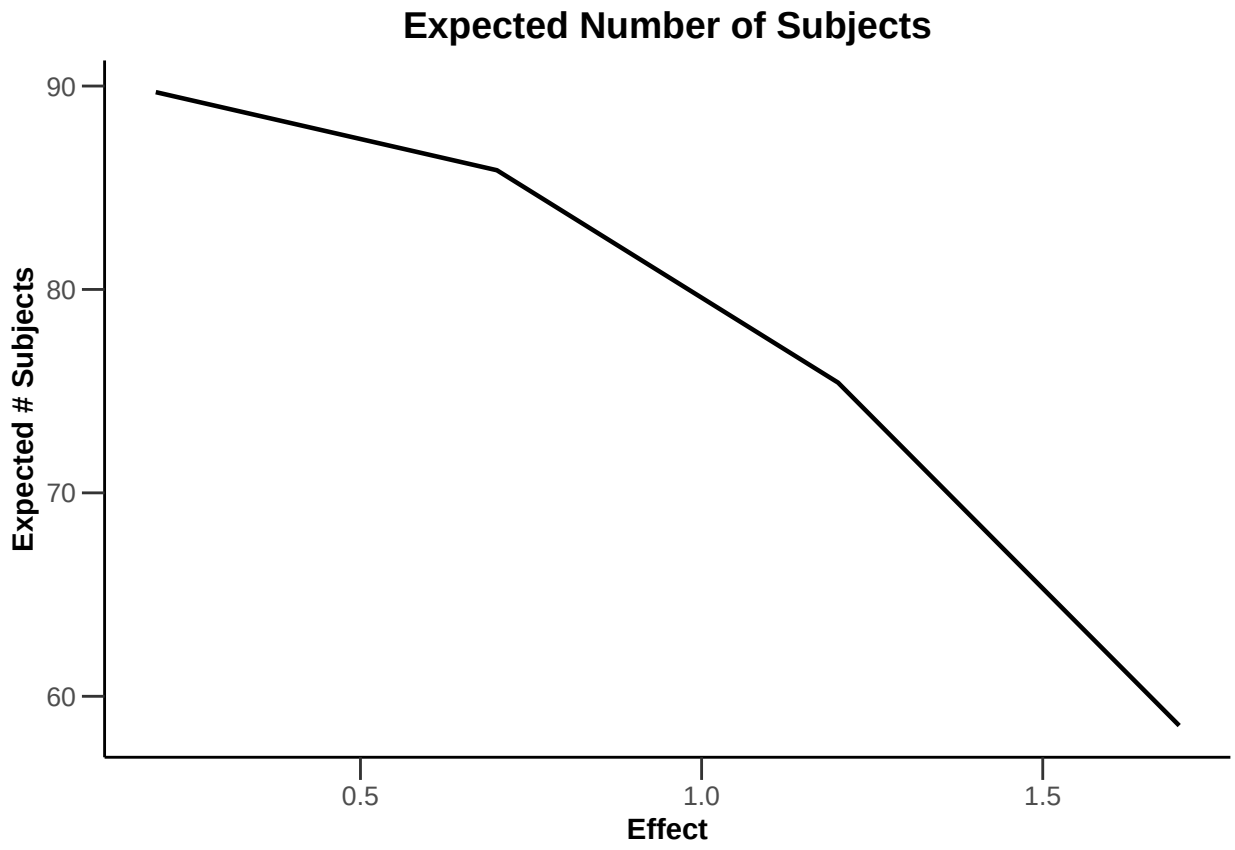

Reject per Stage
Stage
1
2
3
Rejection Probability
0.0
0.1
0.2
0.3
0.5
1.0
1.5
Effect


Overall Power and Early Stopping
Overall reject
Overall futility stop
Early stop
1.00
0.75
0.50
0.25
0.00
0.5
1.0
1.5
Effect


Expected Number of Subjects and Power / Early Stop
Expected # Subjects
Overall Reject and Early Stop
Effect
Overall reject
Expected # subjects
Early stop
0
25
50
75
0.00
0.25
0.50
0.75
0.5
1.0
1.5

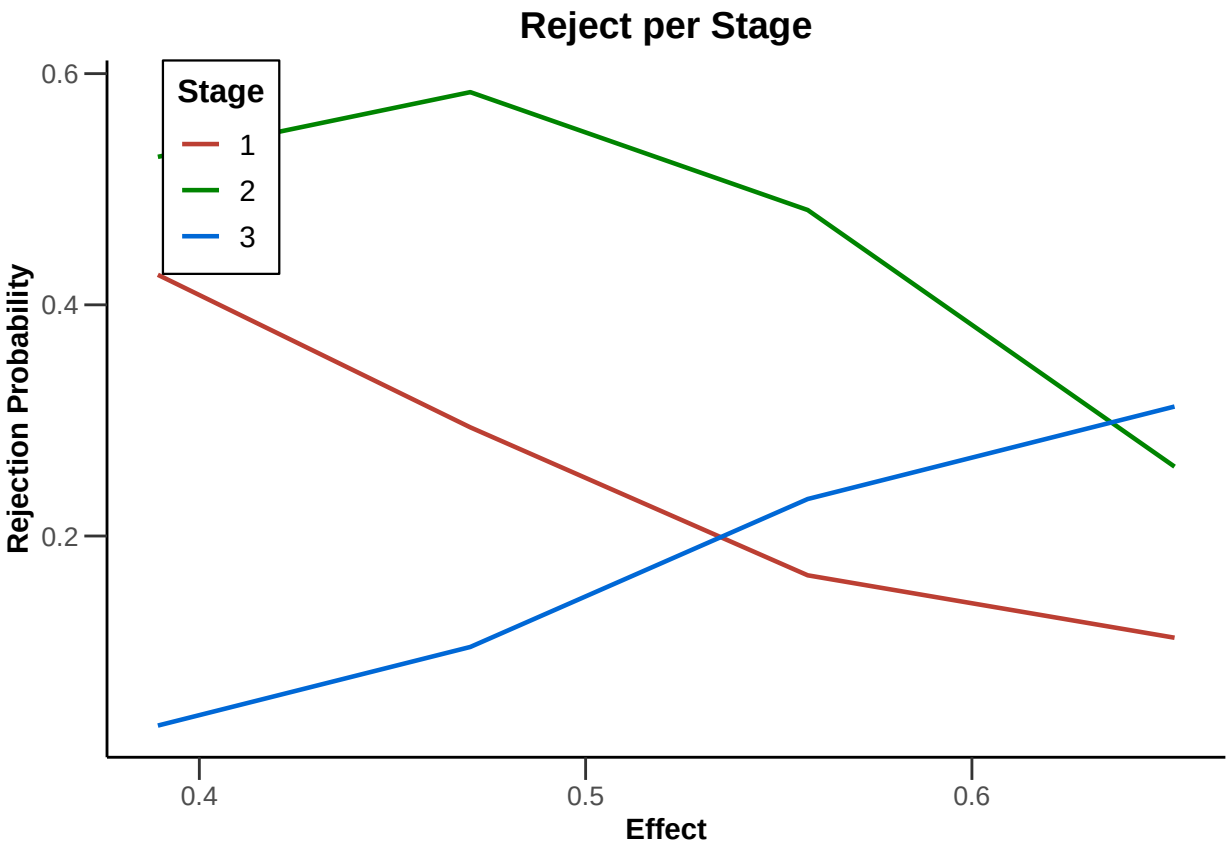

Overall Power
Overall Reject
0.75
0.50
0.25
0.00
0.5
1.0
1.5
Effect

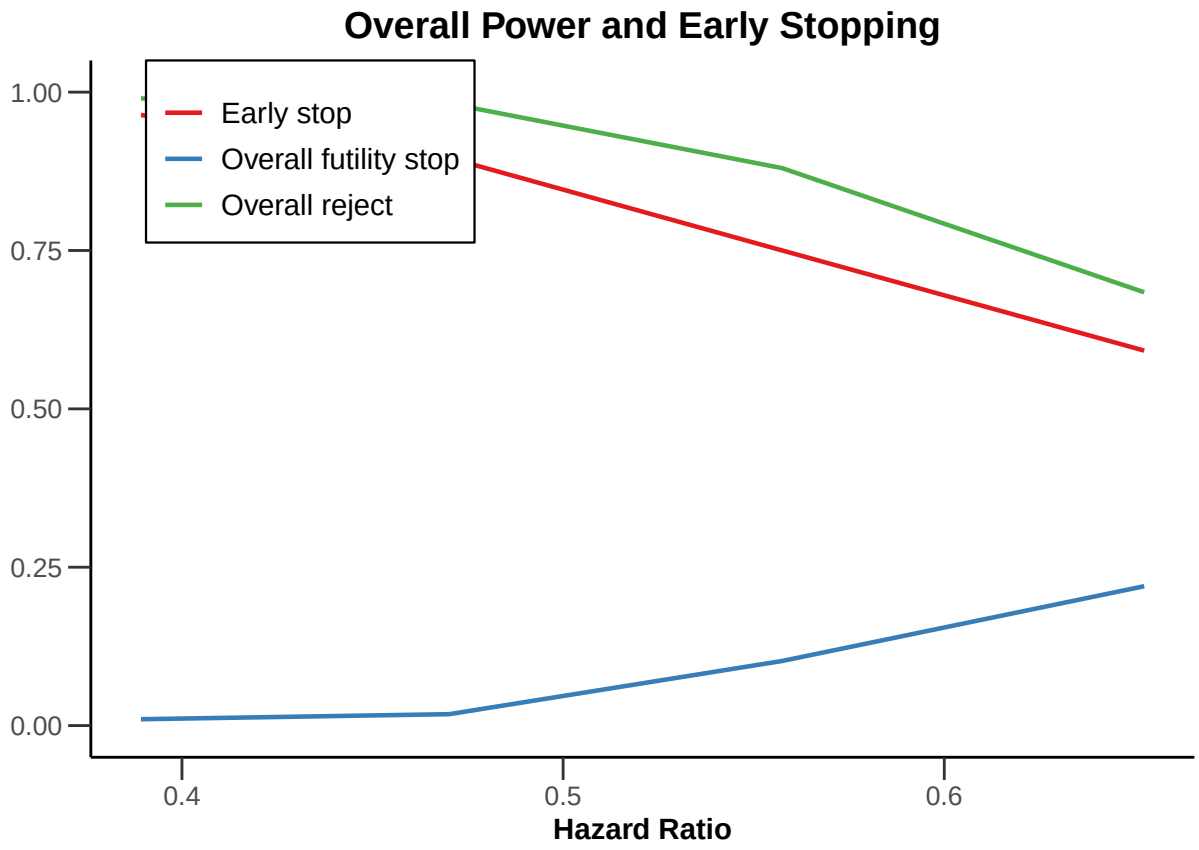

Overall Early Stopping
Early Stop
0.6
0.4
0.2
0.0
0.5
1.0
1.5
Effect

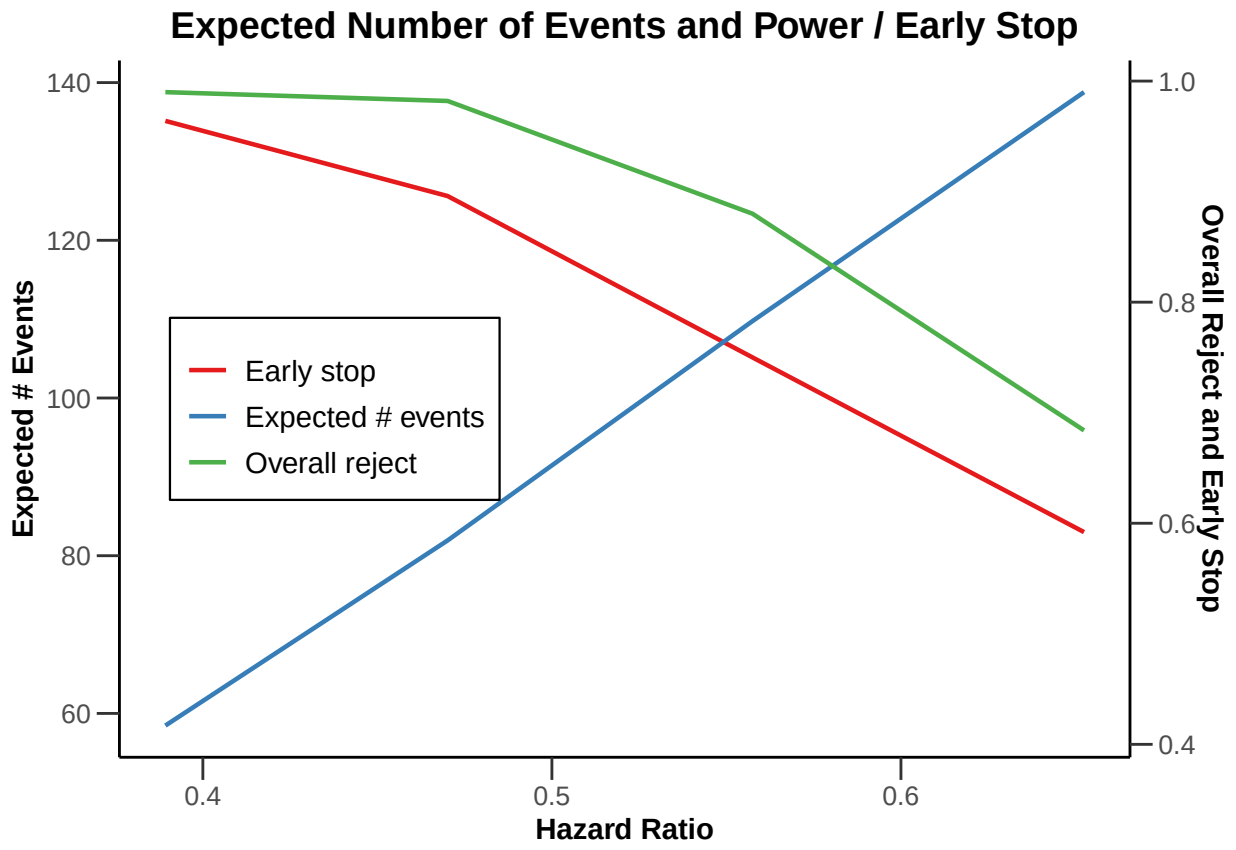


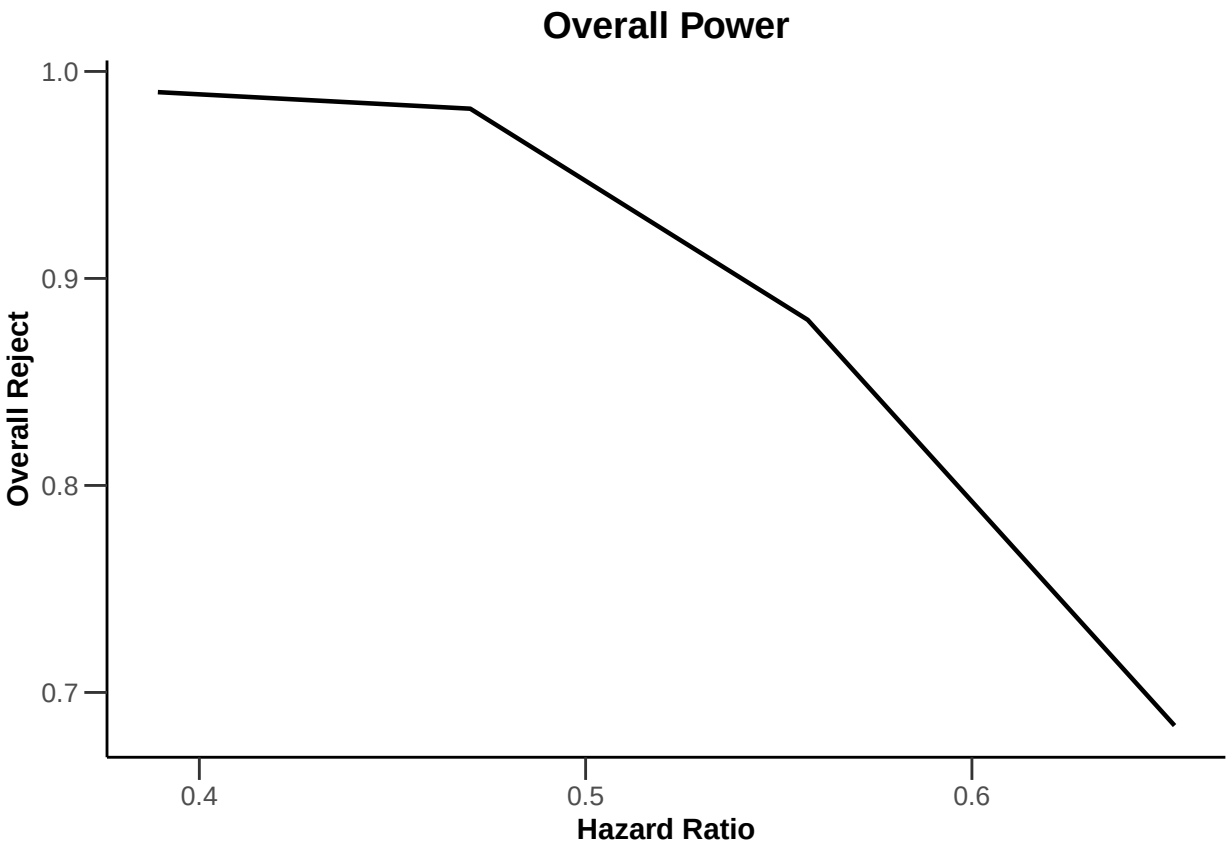
2.3 Simulation results base - survival

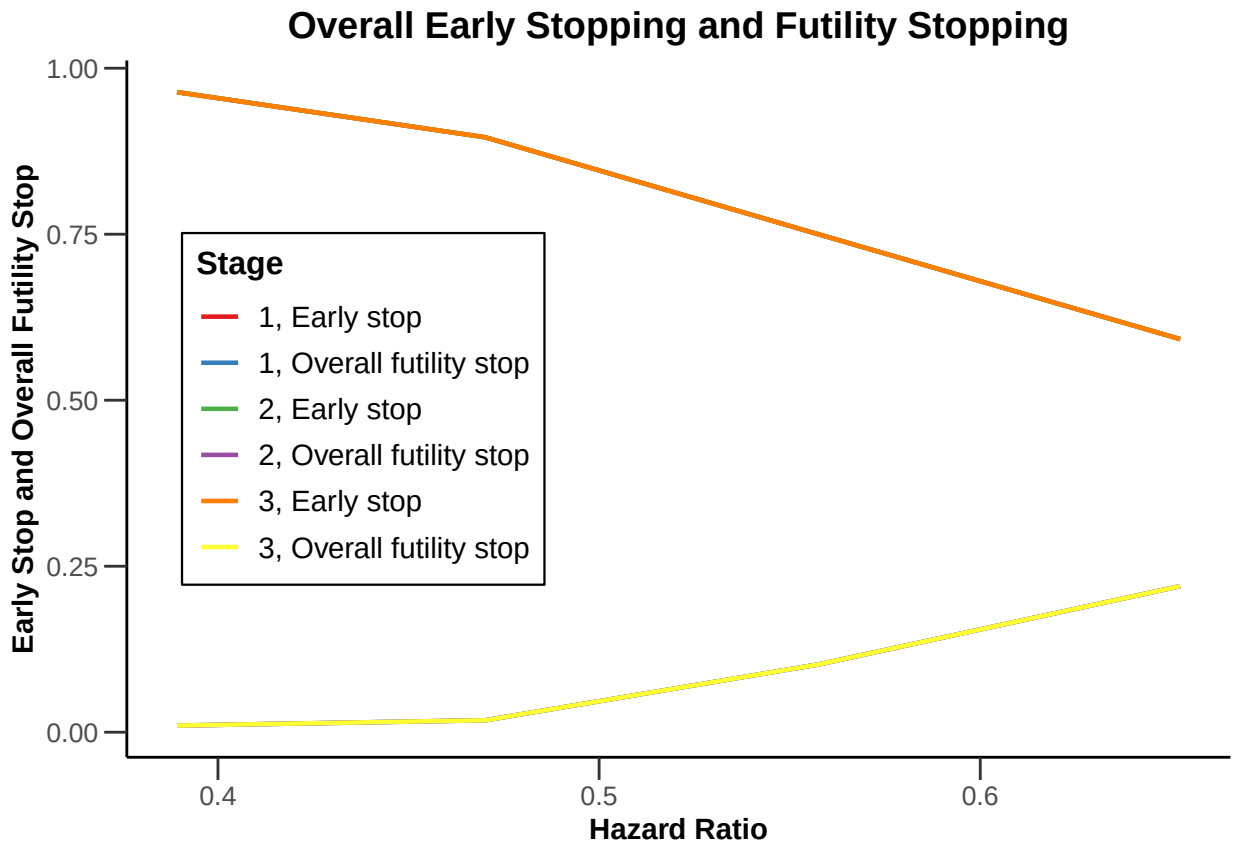
```
design <- getDesignFisher(kMax = 3, alpha0Vec = c(0.5, 0.5))
x <- getSimulationSurvival(
    design = design, pi2 = 0.6,
    pi1 = seq(0.3, 0.45, 0.05), directionUpper = FALSE,
    maxNumberOfSubjects = 500, plannedEvents = (1:design$kMax) * 20,
    allocation1 = 1, allocation2 = 1, accrualTime = c(0, 3, 6, 12),
    accrualIntensity = c(0.1, 0.2, 0.2), dropoutRate1 = 0,
    dropoutRate2 = 0, dropoutTime = 12, conditionalPower = 0.8,
    minNumberOfEventsPerStage = c(NA_real_, 10, 10),
    maxNumberOfEventsPerStage = c(NA_real_, 100, 200),
    maxNumberOfIterations = 500, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

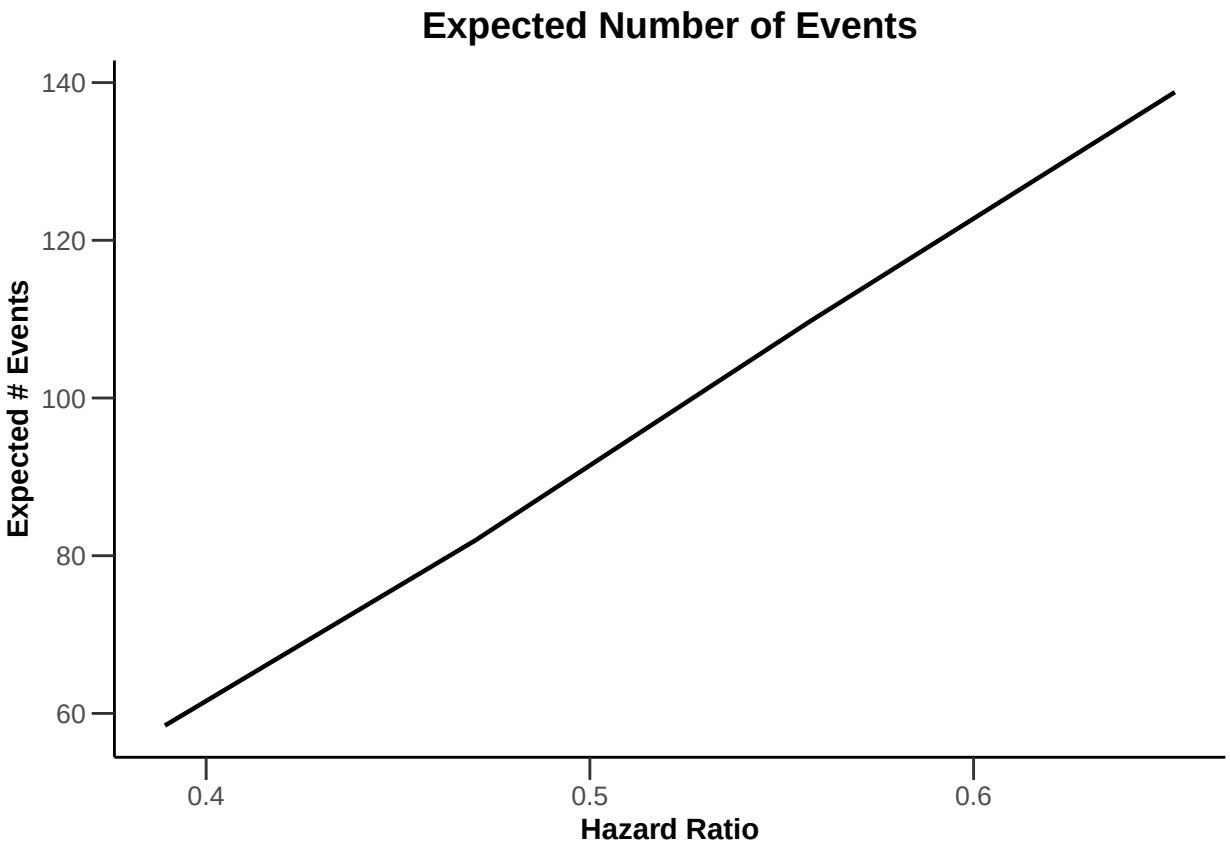

Reject per Stage
Stage
1
2
3
Rejection Probability
0.6
0.4
0.2
0.4
0.5
0.6
Effect

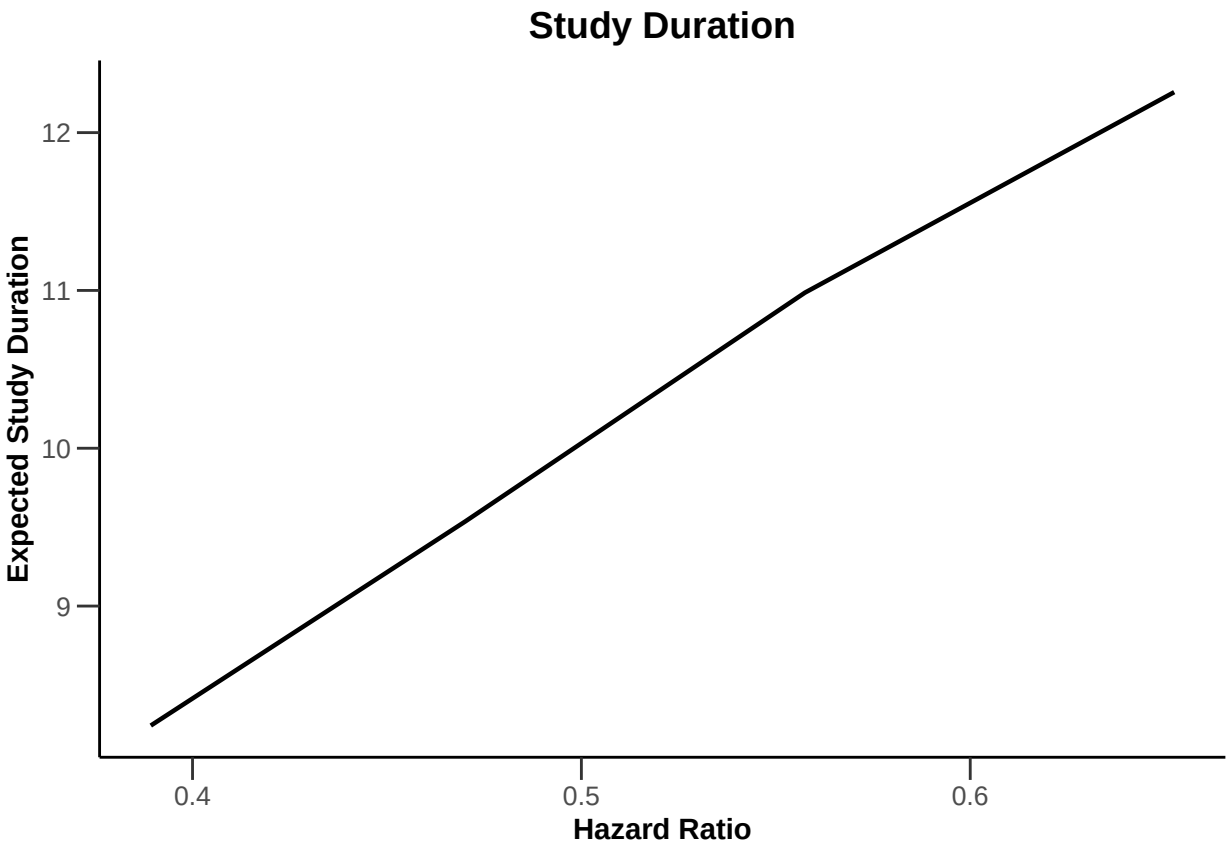

Overall Power and Early Stopping
Early stop
Overall futility stop
Overall reject
1.00
0.75
0.50
0.25
0.00
0.4
0.5
0.6
Hazard Ratio

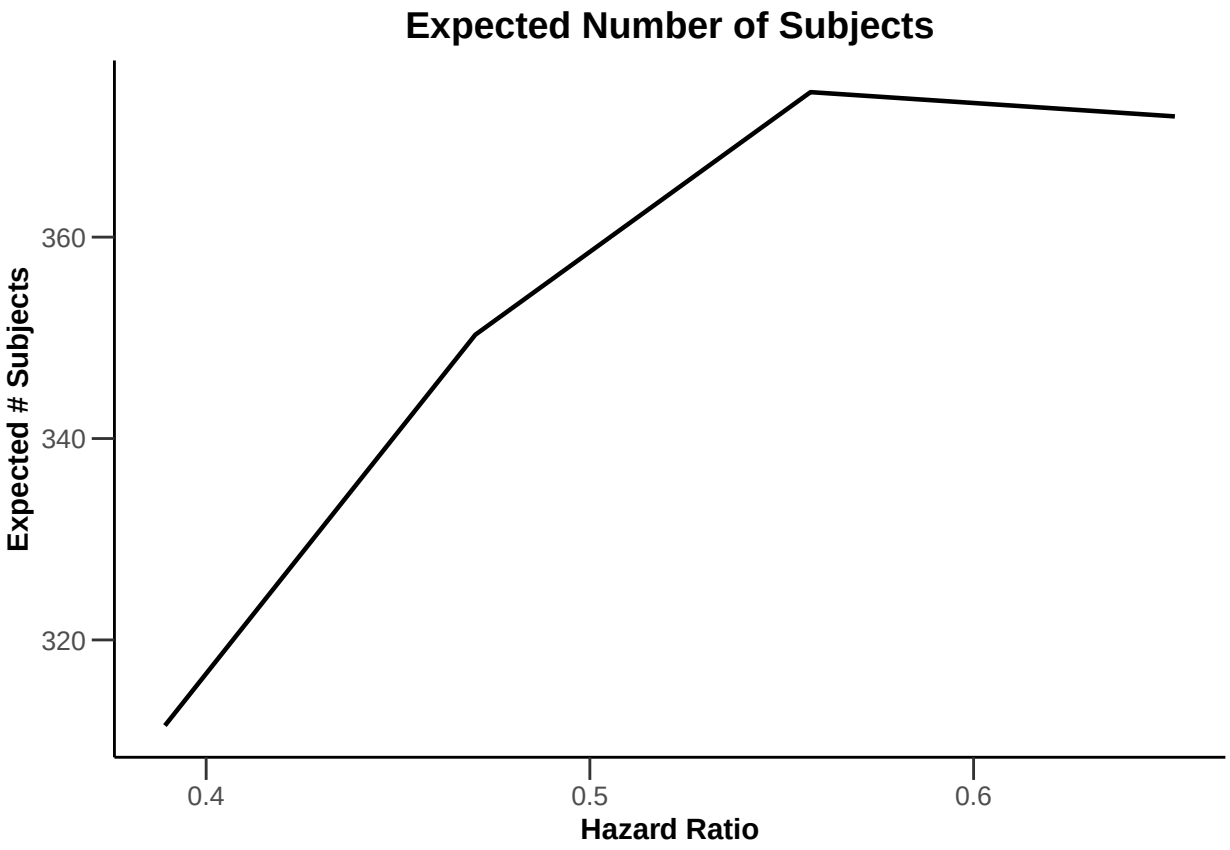

Expected Number of Events and Power / Early Stop
Expected # Events
Overall Reject and Early Stop
Hazard Ratio
Early stop
Expected # events
Overall reject
140
120
100
80
60
1.0
0.8
0.6
0.4
0.4
0.5
0.6

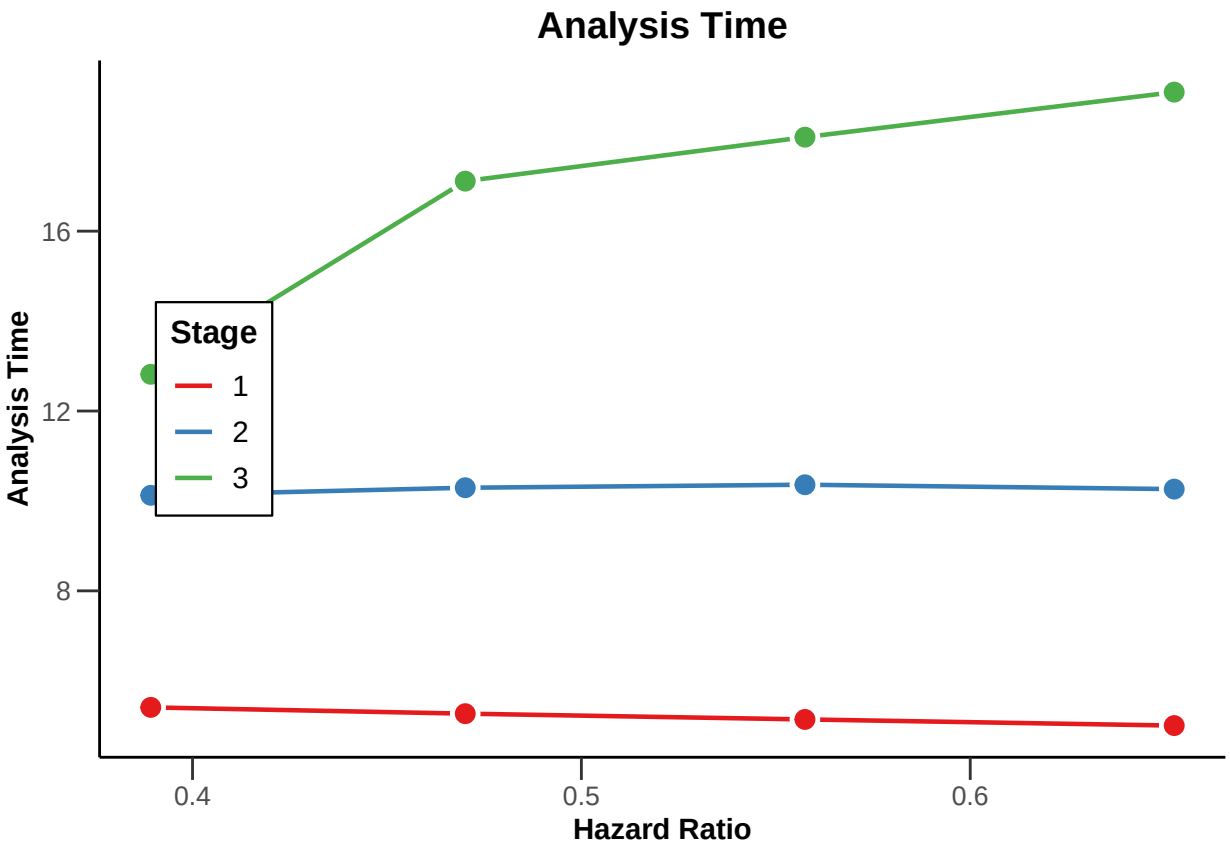

Overall Power
1.0
0.9
0.8
0.7
Overall Reject
0.4
0.5
0.6
Hazard Ratio

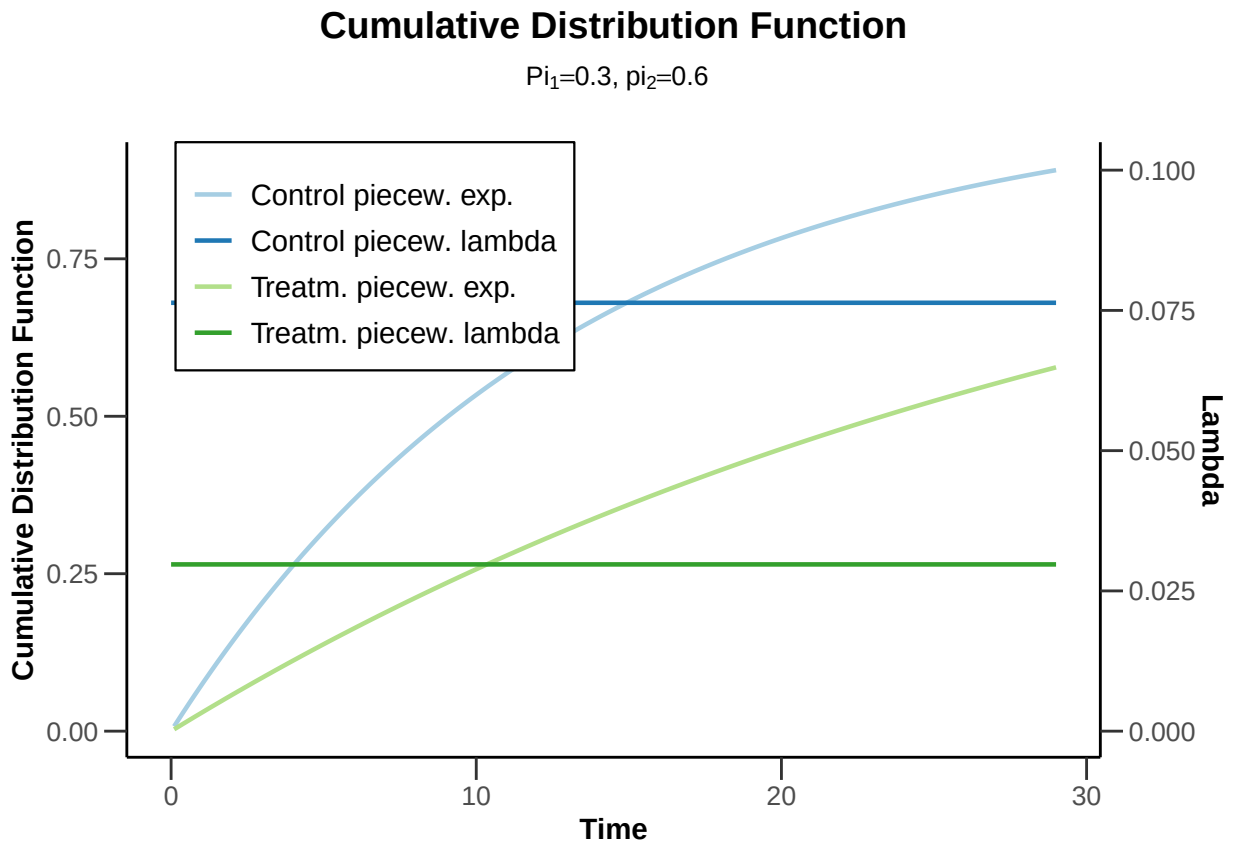

Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
1.00
0.75
0.50
0.25
0.00
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Early stop
3, Overall futility stop
0.4
0.5
0.6
Hazard Ratio

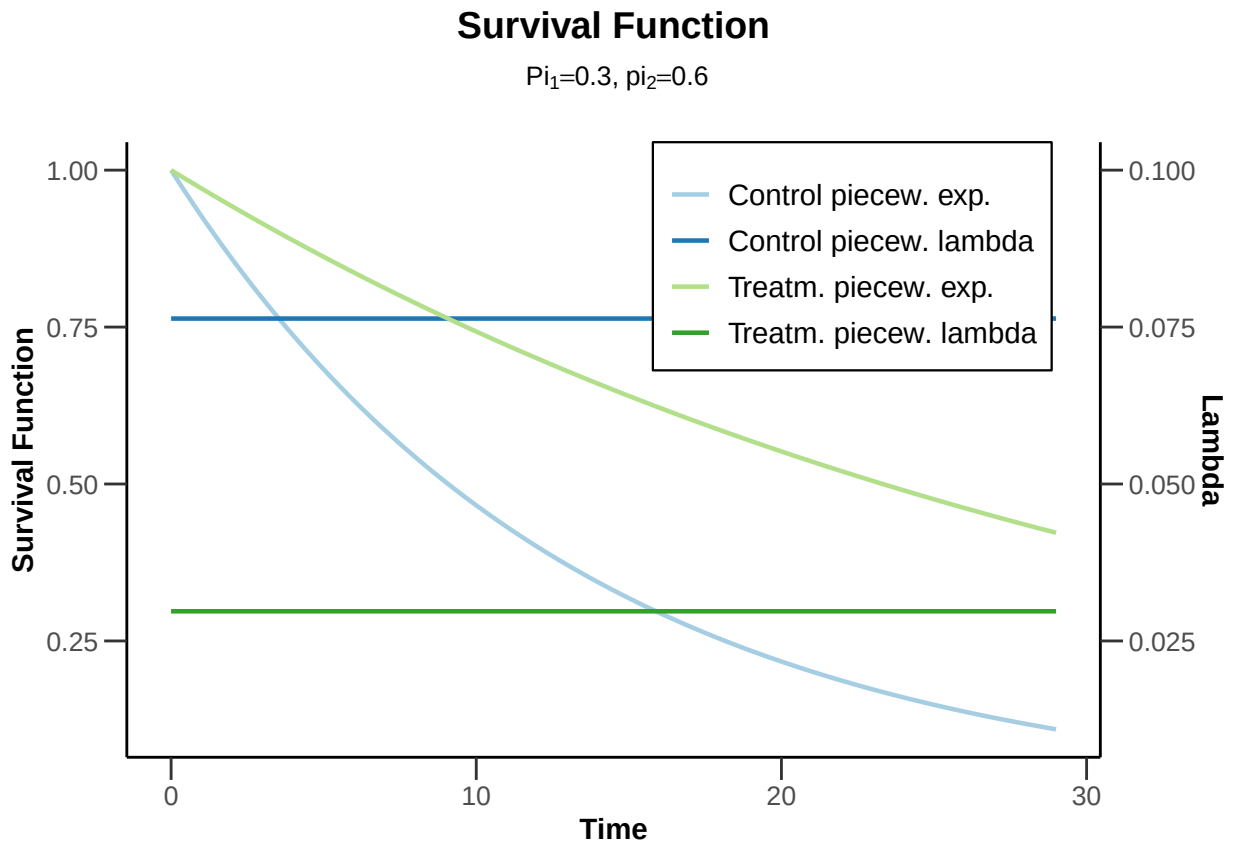

Expected Number of Events
Expected # Events
140
120
100
80
60
0.4
0.5
0.6
Hazard Ratio


Study Duration
Expected Study Duration
12
11
10
9
0.4
0.5
0.6
Hazard Ratio


Expected Number of Subjects
Expected # Subjects
360
340
320
0.4
0.5
0.6
Hazard Ratio


Analysis Time
Analysis Time
16
12
8
Stage
1
2
3
0.4
0.5
0.6
Hazard Ratio

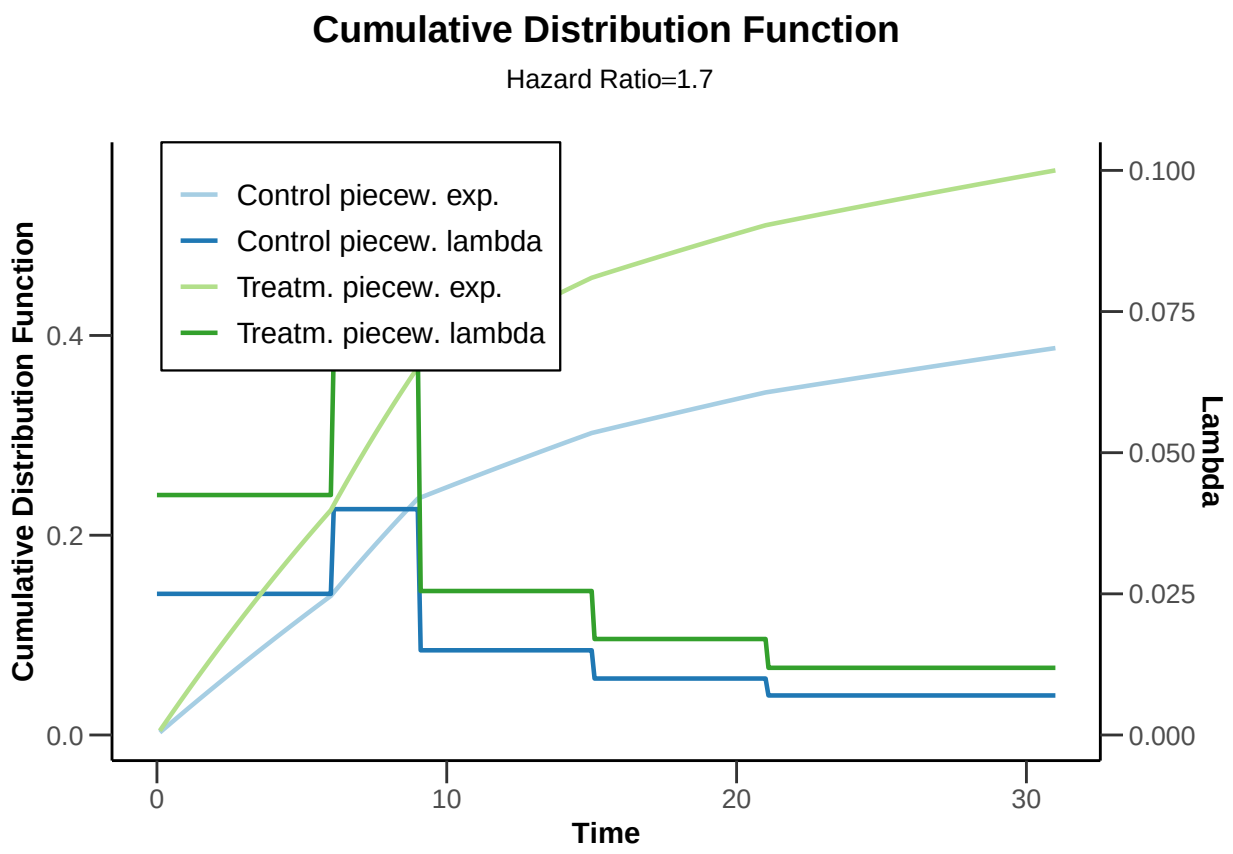

Cumulative Distribution Function
Pi1=0.3, pi2=0.6
Control piecew. exp.
Control piecew. lambda
Treatm. piecew. exp.
Treatm. piecew. lambda
Cumulative Distribution Function
0.00
0.25
0.50
0.75
0.000
0.025
0.050
0.075
0.100
Lambda
0
10
20
30
Time

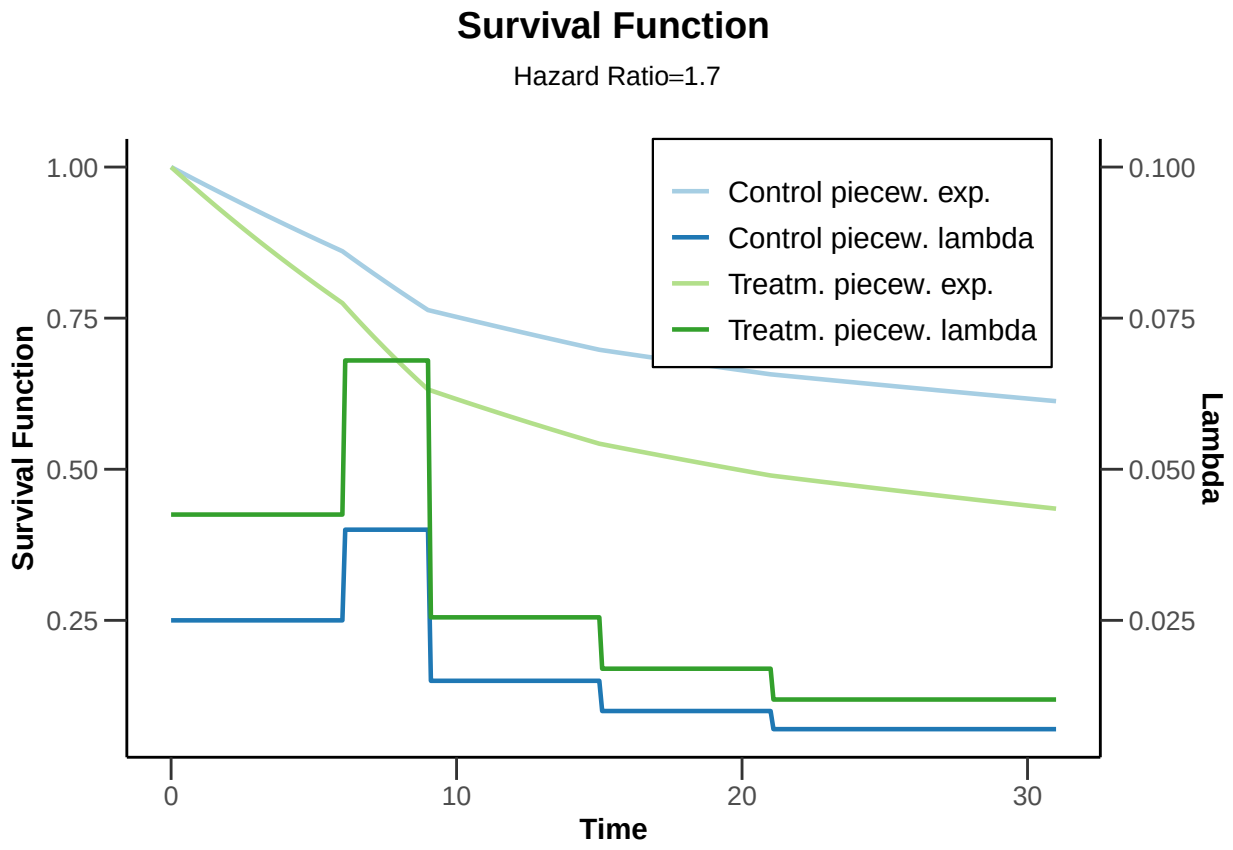



```
design <- getDesignGroupSequential(kMax = 3, typeOfDesign = "WT", deltaWT = 0.25)
piecewiseSurvivalTime <- list(
    "<6" = 0.025,
    "6 - <9" = 0.04,
    "9 - <15" = 0.015,
    "15 - <21" = 0.01,
    ">=21" = 0.007
)
x <- getSimulationSurvival(
    design = design,
    directionUpper = TRUE, maxNumberOfSubjects = 500,
    plannedEvents = (1:design$kMax) * 20, allocation1 = 1,
    allocation2 = 1, accrualTime = c(0, 3, 6, 12),
    piecewiseSurvivalTime = piecewiseSurvivalTime, hazardRatio = 1.7,
    accrualIntensity = c(0.1, 0.2, 0.2), dropoutRate1 = 0,
    dropoutRate2 = 0, dropoutTime = 12, conditionalPower = 0.8,
    minNumberOfEventsPerStage = c(NA_real_, 10, 10),
    maxNumberOfEventsPerStage = c(NA_real_, 100, 200),
    maxNumberOfIterations = 500, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

```
## Warning in !is.null(lambda1) && !is.na(lambda1): 'length(x) = 5 > 1' in coercion
## to 'logical(1)'

## Warning in !is.null(lambda1) && !is.na(lambda1): 'length(x) = 5 > 1' in coercion
```

```
## to 'logical(1)'

## Warning in !is.null(lambda1) && !is.na(lambda1): 'length(x) = 5 > 1' in coercion
## to 'logical(1)'

## Warning in !is.null(lambda1) && !is.na(lambda1): 'length(x) = 5 > 1' in coercion
## to 'logical(1)'
```

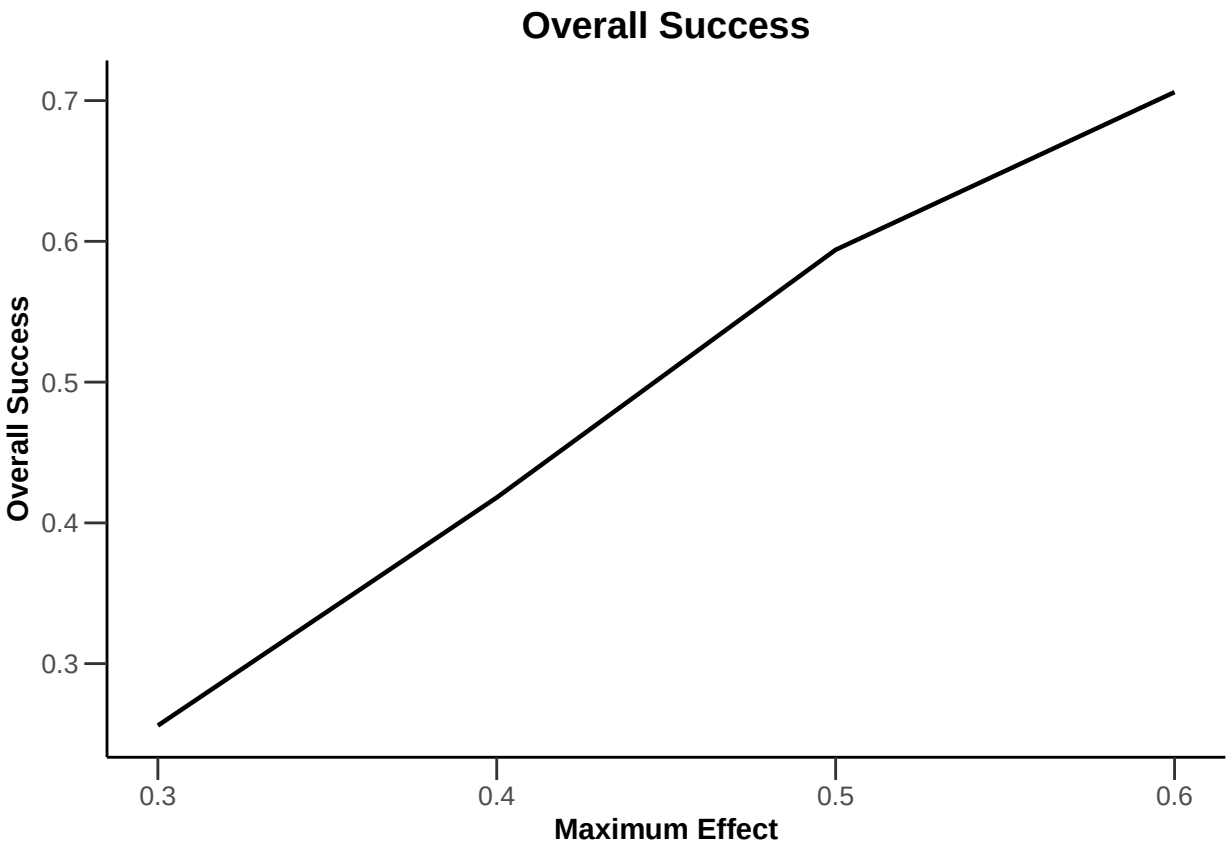


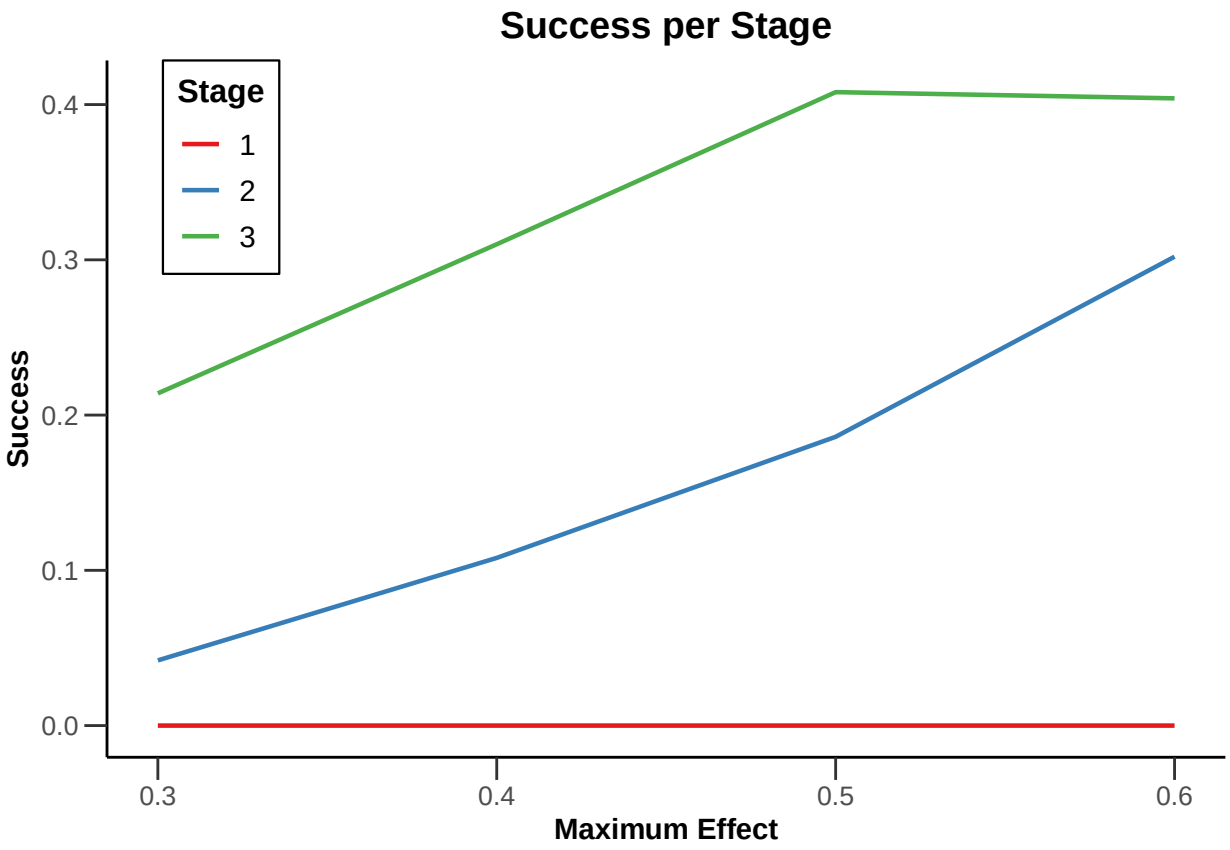


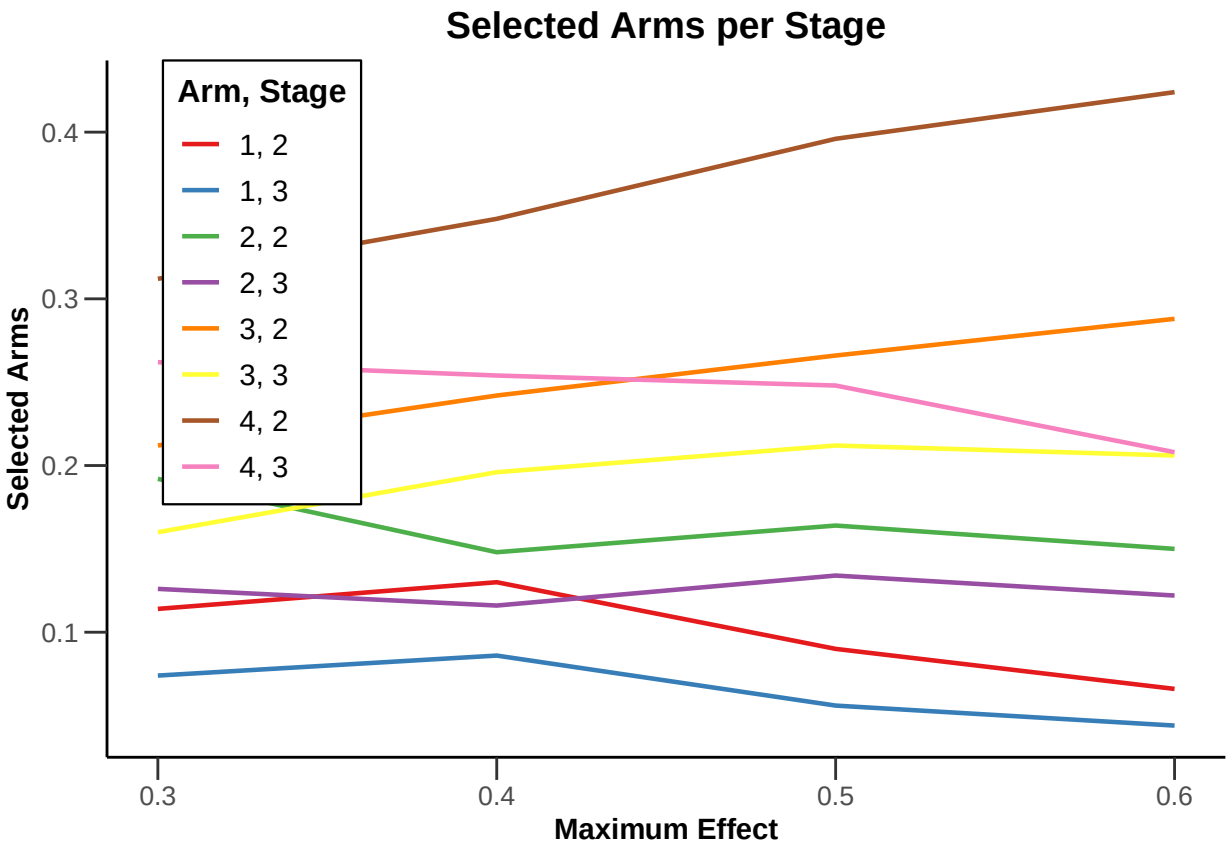
3 Simulation results multi-arm

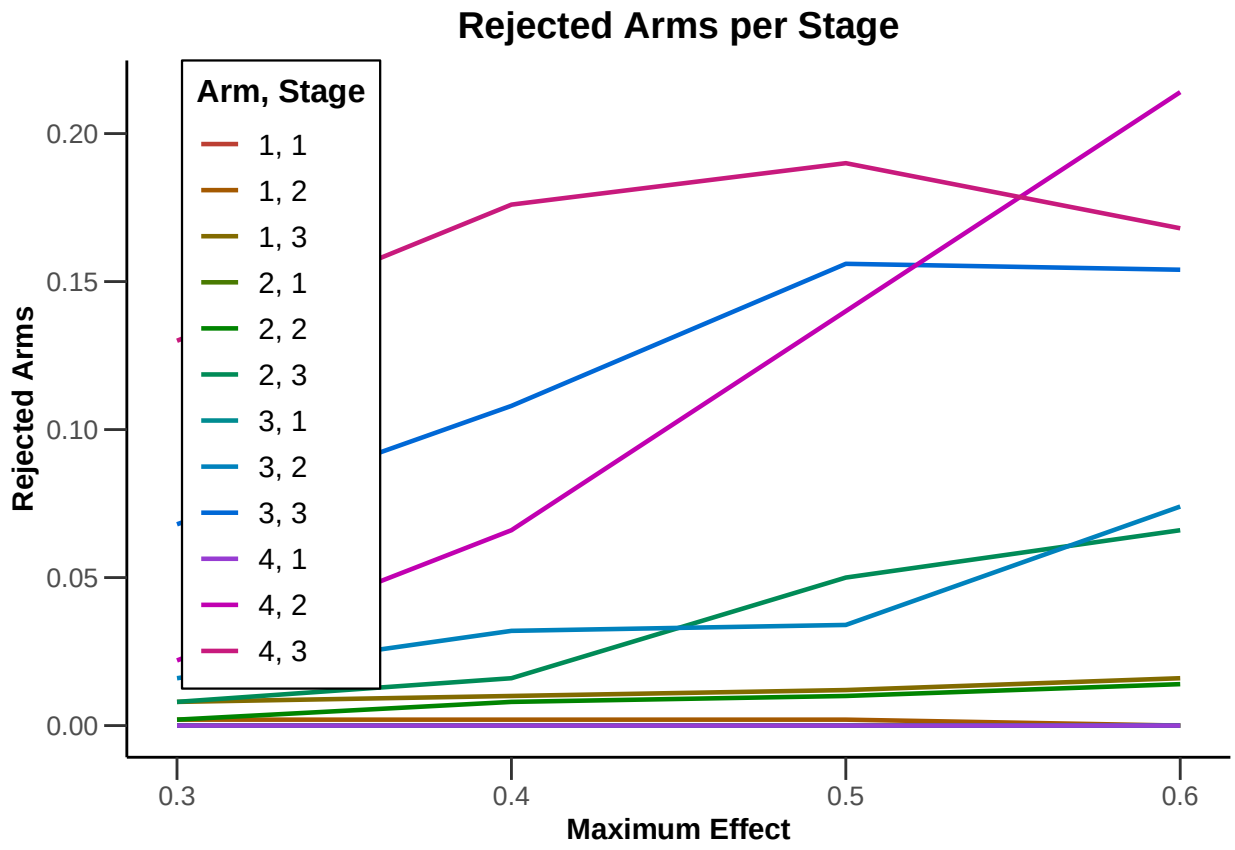
3.1 Simulation results multi-arm - means

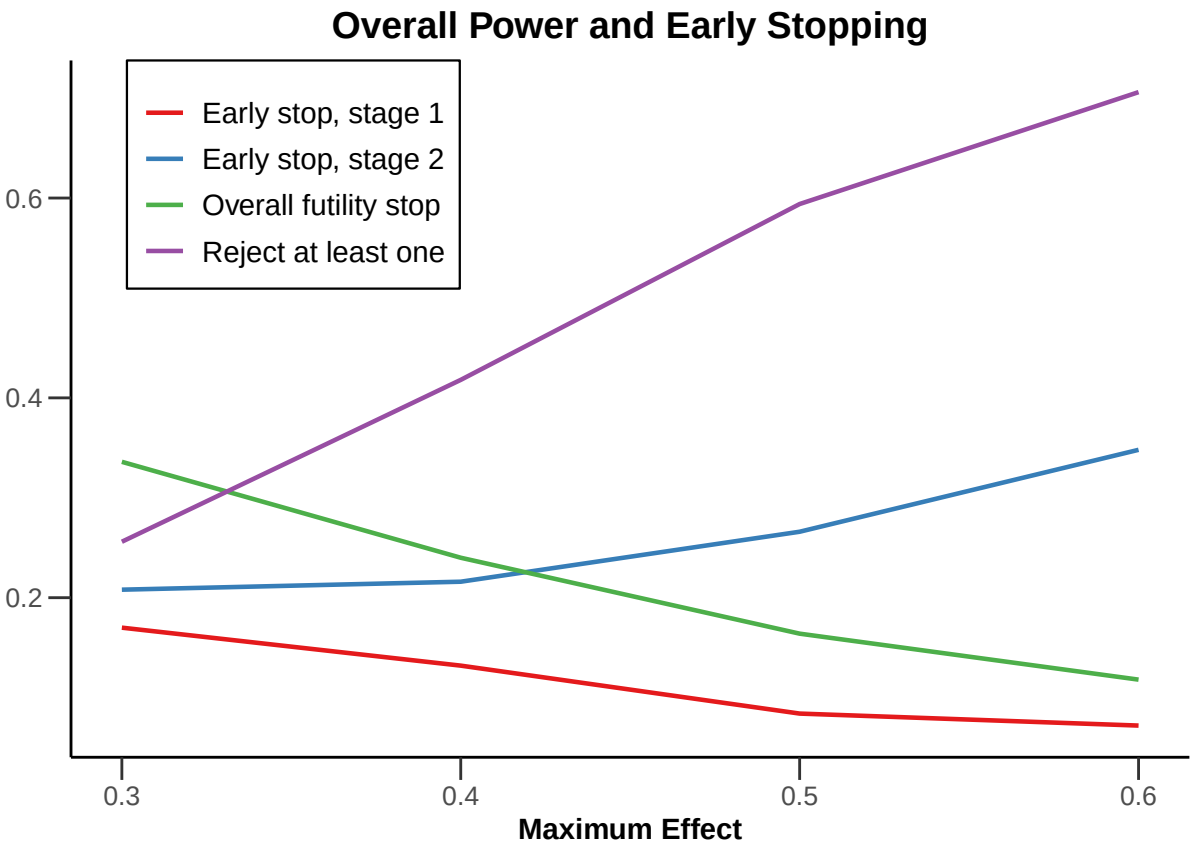
```
design <- getDesignInverseNormal(
    informationRates = c(0.2, 0.6, 1),
    futilityBounds = c(-0.5, 0.5)
)
x <- getSimulationMultiArmMeans(
    design = design, typeOfShape = "linear",
    activeArms = 4, plannedSubjects = c(10, 30, 50), stDev = 1.2,
    muMaxVector = seq(0.3, 0.6, 0.1), adaptations = rep(TRUE, 2),
    conditionalPower = 0.8, minNumberOfSubjectsPerStage = c(10, 4, 4),
    maxNumberOfSubjectsPerStage = c(10, 100, 100),
    maxNumberOfIterations = 500, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

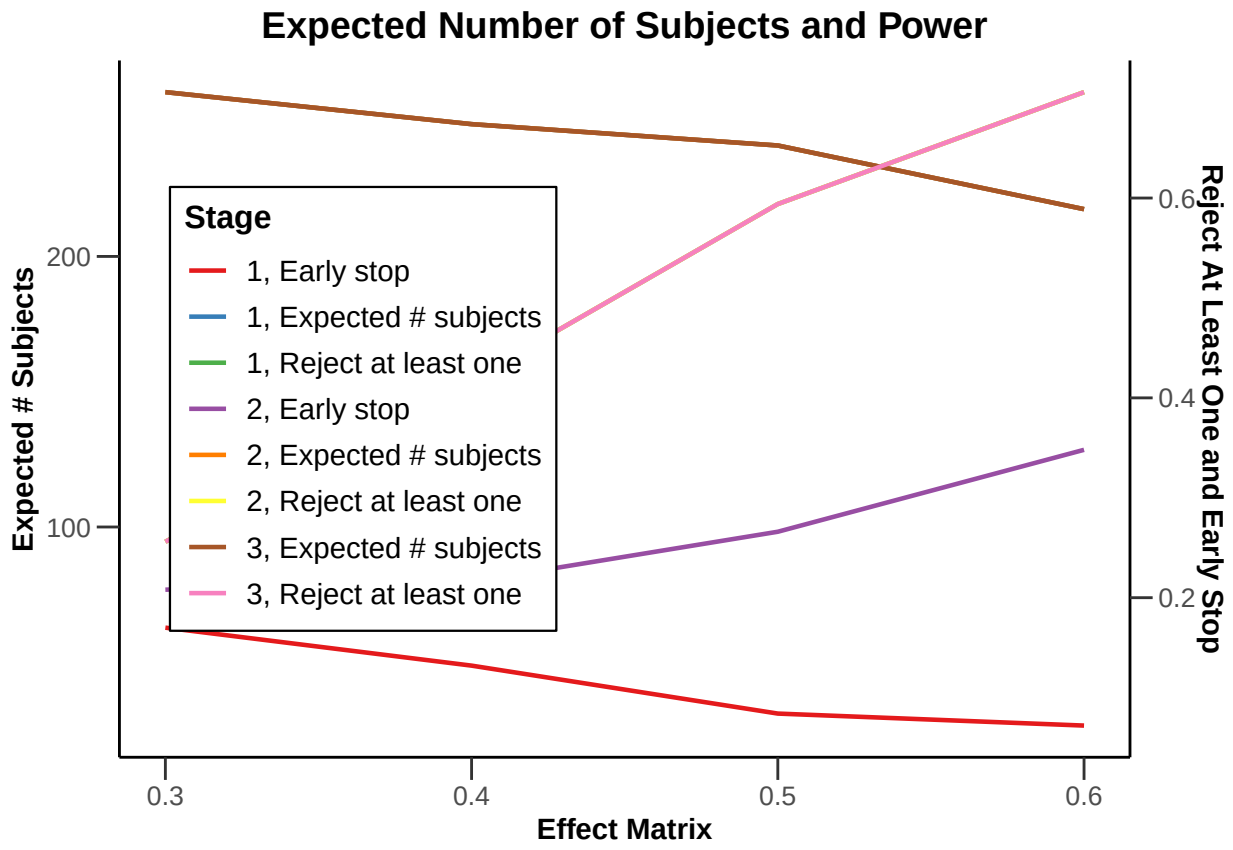

Overall Success
0.7
0.6
0.5
0.4
0.3
Overall Success
0.3
0.4
0.5
0.6
Maximum Effect

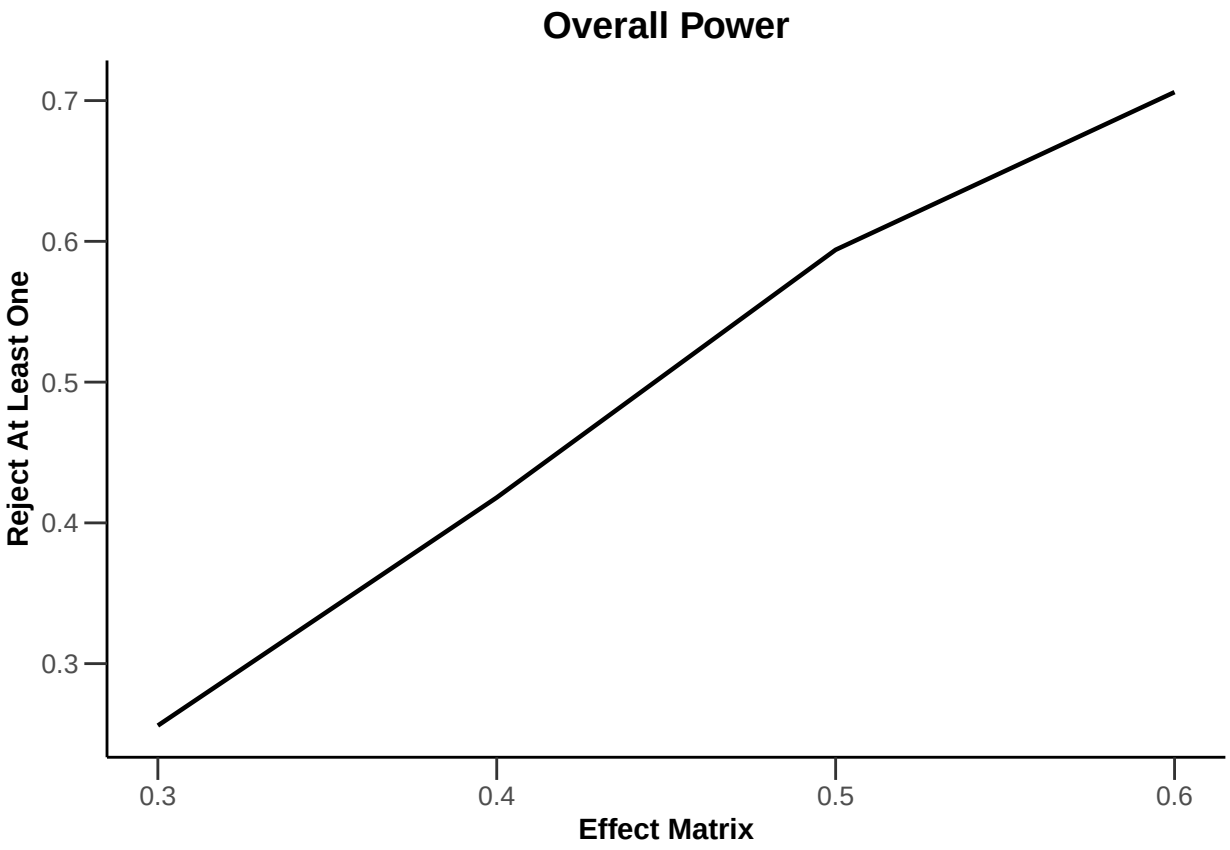

Success per Stage
Stage
1
2
3
Success
0.0
0.1
0.2
0.3
0.4
0.3
0.4
0.5
0.6
Maximum Effect

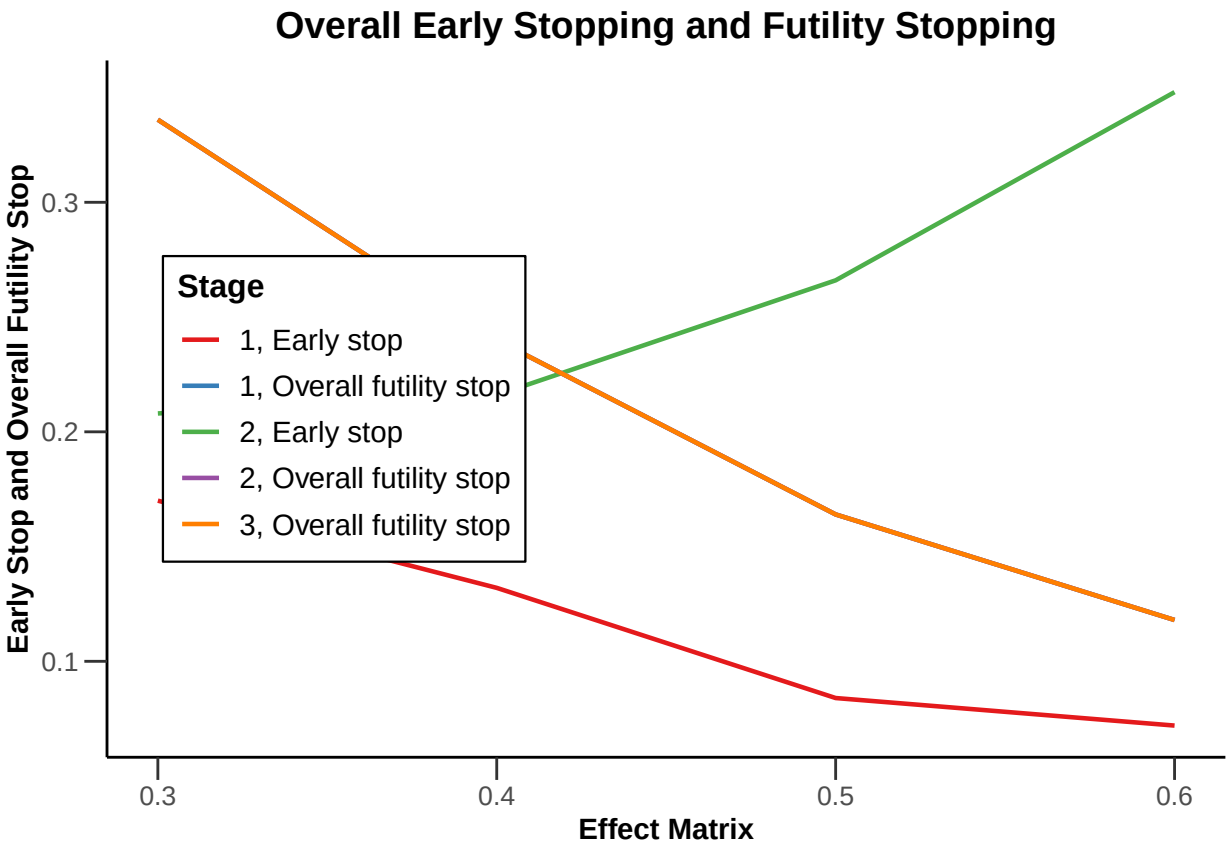

Selected Arms per Stage
Arm, Stage
1, 2
1, 3
2, 2
2, 3
3, 2
3, 3
4, 2
4, 3
Selected Arms
0.4
0.3
0.2
0.1
0.3
0.4
0.5
0.6
Maximum Effect

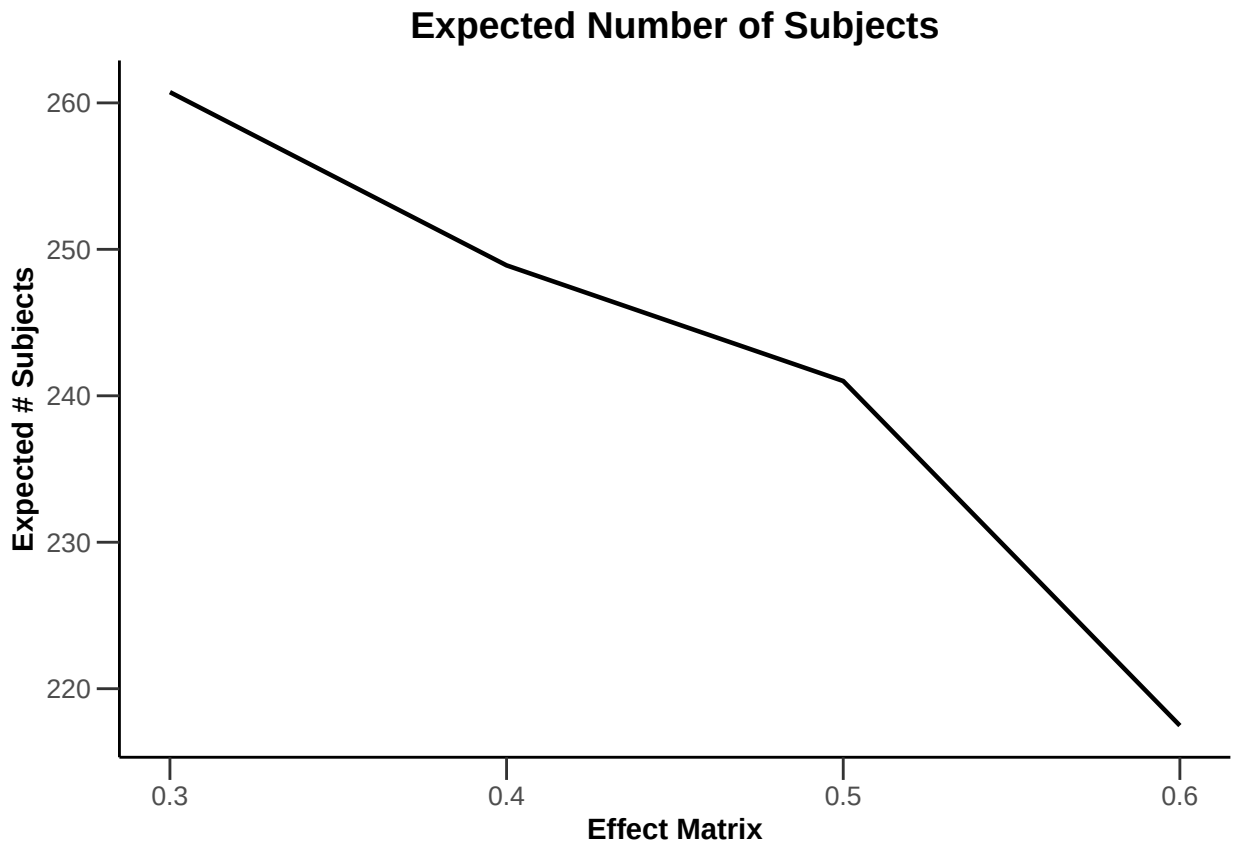

Rejected Arms per Stage
Arm, Stage
1, 1
1, 2
1, 3
2, 1
2, 2
2, 3
3, 1
3, 2
3, 3
4, 1
4, 2
4, 3
Rejected Arms
0.20
0.15
0.10
0.05
0.00
0.3
0.4
0.5
0.6
Maximum Effect


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.6
0.4
0.2
0.3
0.4
0.5
0.6
Maximum Effect


Expected Number of Subjects and Power
Stage
1, Early stop
1, Expected # subjects
1, Reject at least one
2, Early stop
2, Expected # subjects
2, Reject at least one
3, Expected # subjects
3, Reject at least one
Expected # Subjects
200
100
Reject At Least One and Early Stop
0.6
0.4
0.2
0.3
0.4
0.5
0.6
Effect Matrix

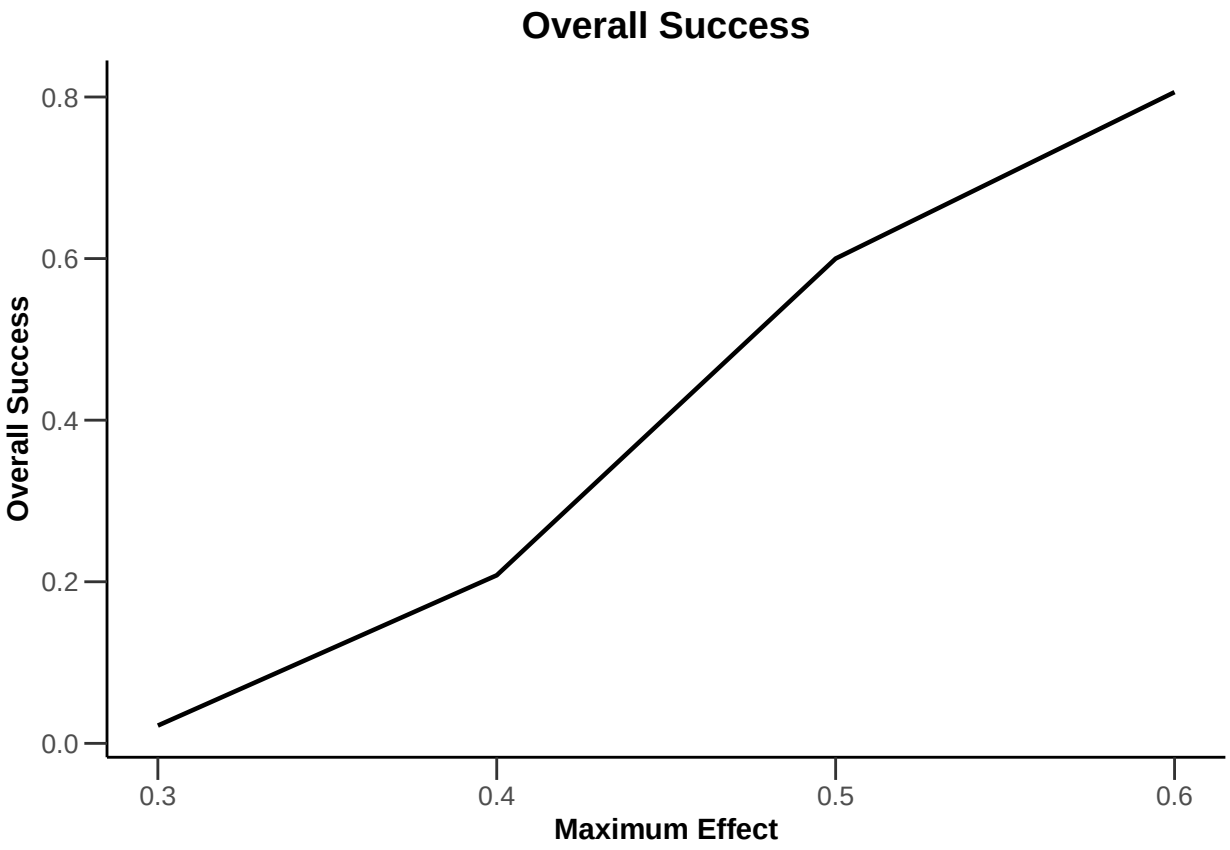

Overall Power
Reject At Least One
0.7
0.6
0.5
0.4
0.3
0.3
0.4
0.5
0.6
Effect Matrix

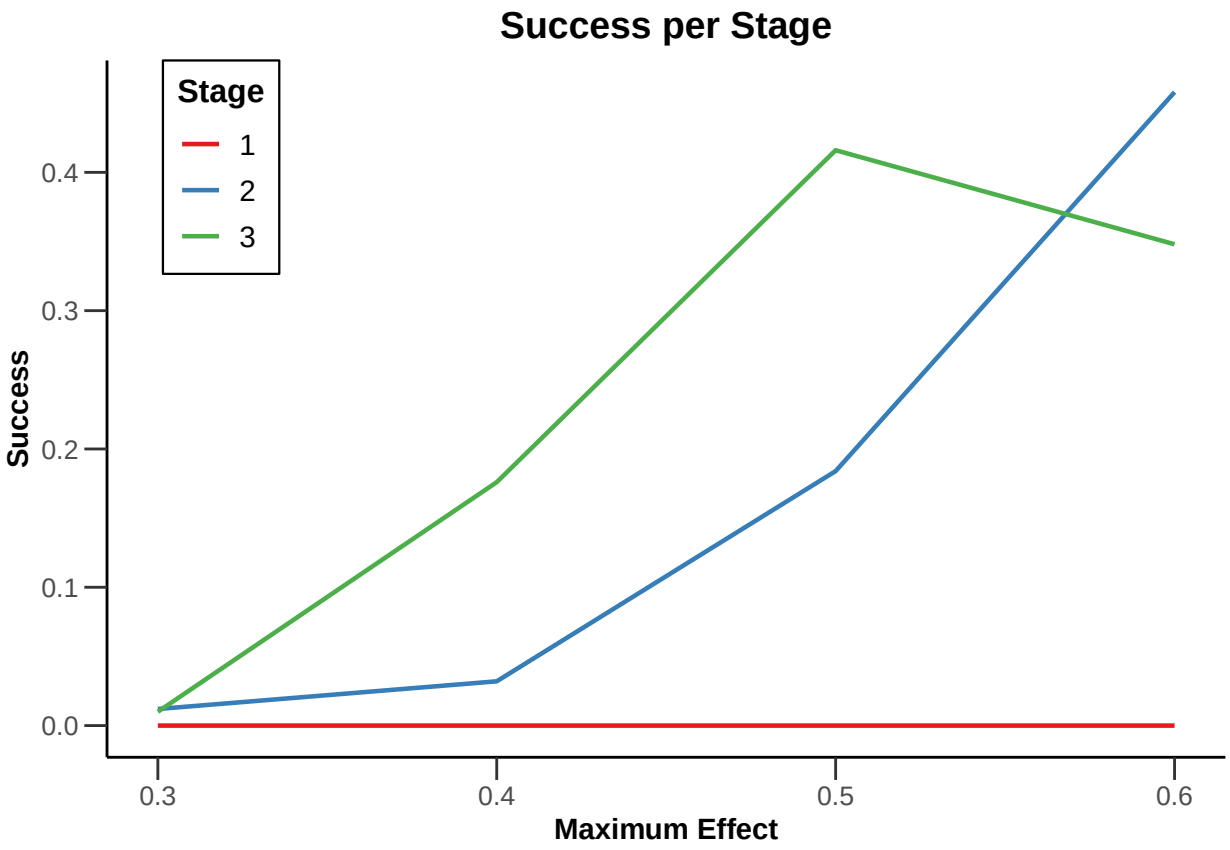

Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
0.3
0.2
0.1
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Overall futility stop
0.3
0.4
0.5
0.6
Effect Matrix

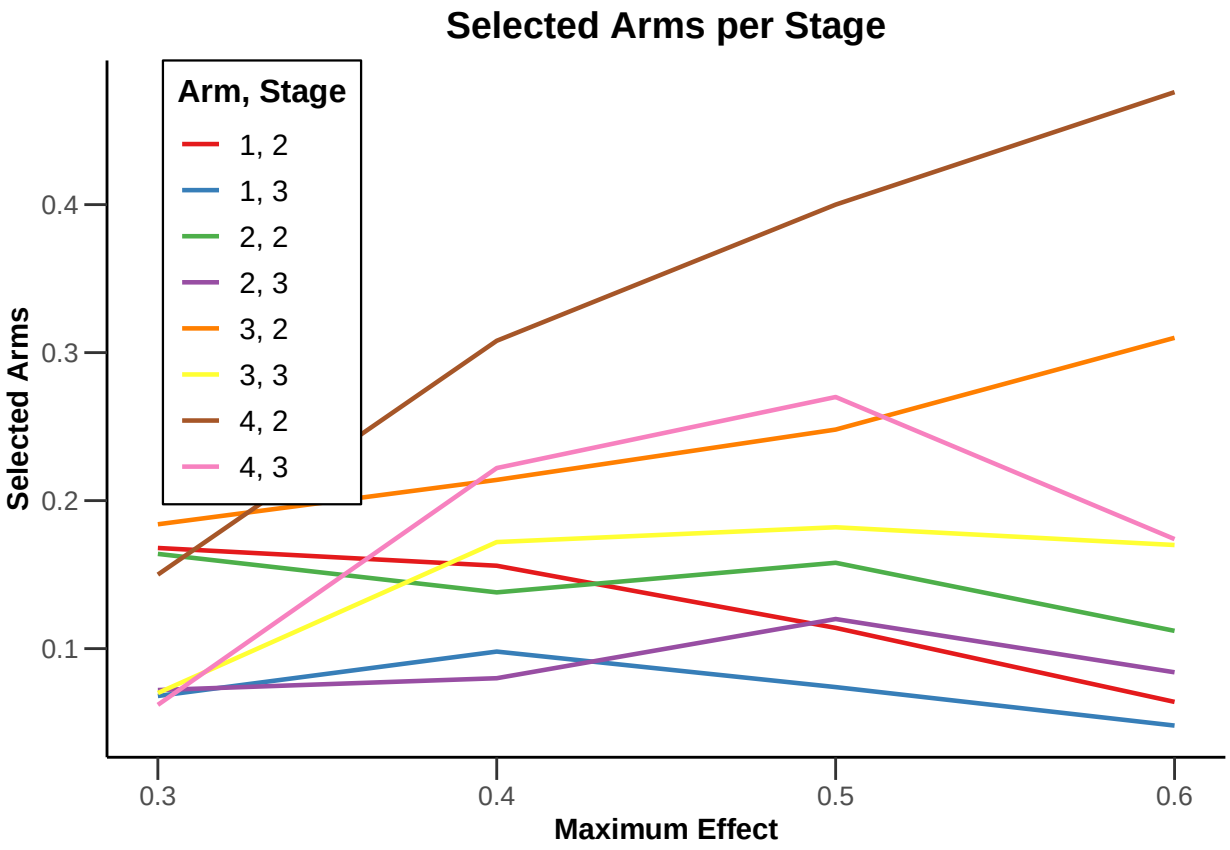


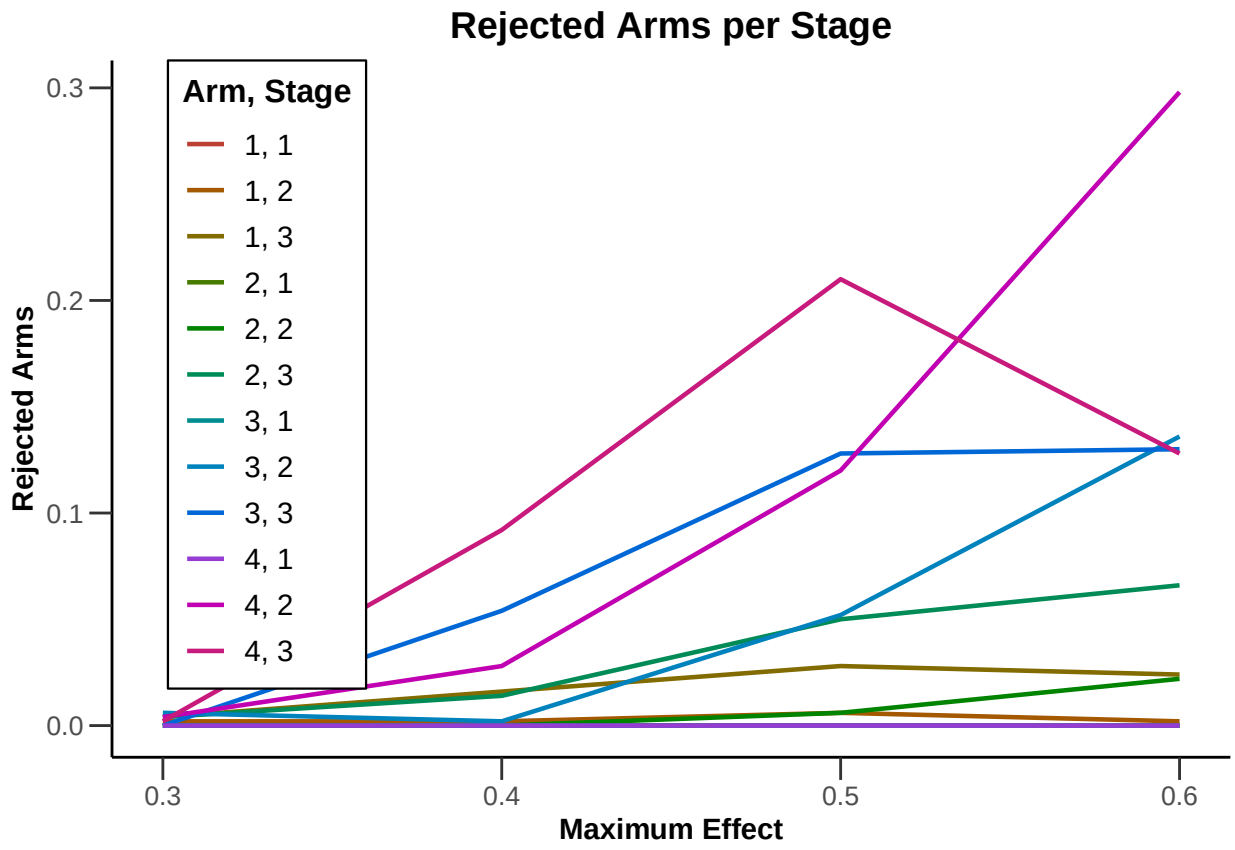
3.2 Simulation results multi-arm - rates

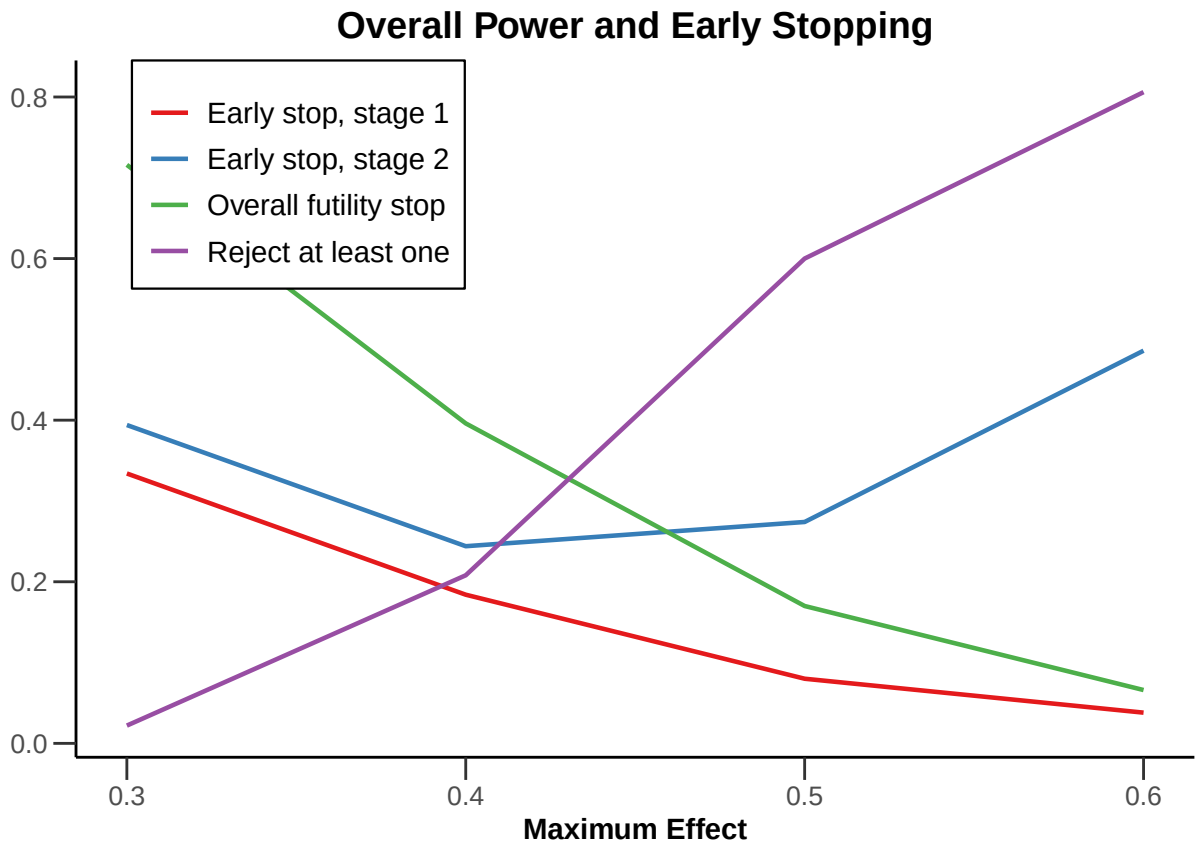
```
design <- getDesignInverseNormal(
    informationRates = c(0.2, 0.6, 1),
    futilityBounds = c(-0.5, 0.5)
)
x <- getSimulationMultiArmRates(
    design = design, typeOfShape = "linear",
    activeArms = 4, plannedSubjects = c(10, 30, 50),
    piControl = 0.3, piMaxVector = seq(0.3, 0.6, 0.1),
    adaptations = rep(TRUE, 2), conditionalPower = 0.8,
    minNumberOfSubjectsPerStage = c(10, 4, 4),
    maxNumberOfSubjectsPerStage = c(10, 100, 100),
    maxNumberOfIterations = 500, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

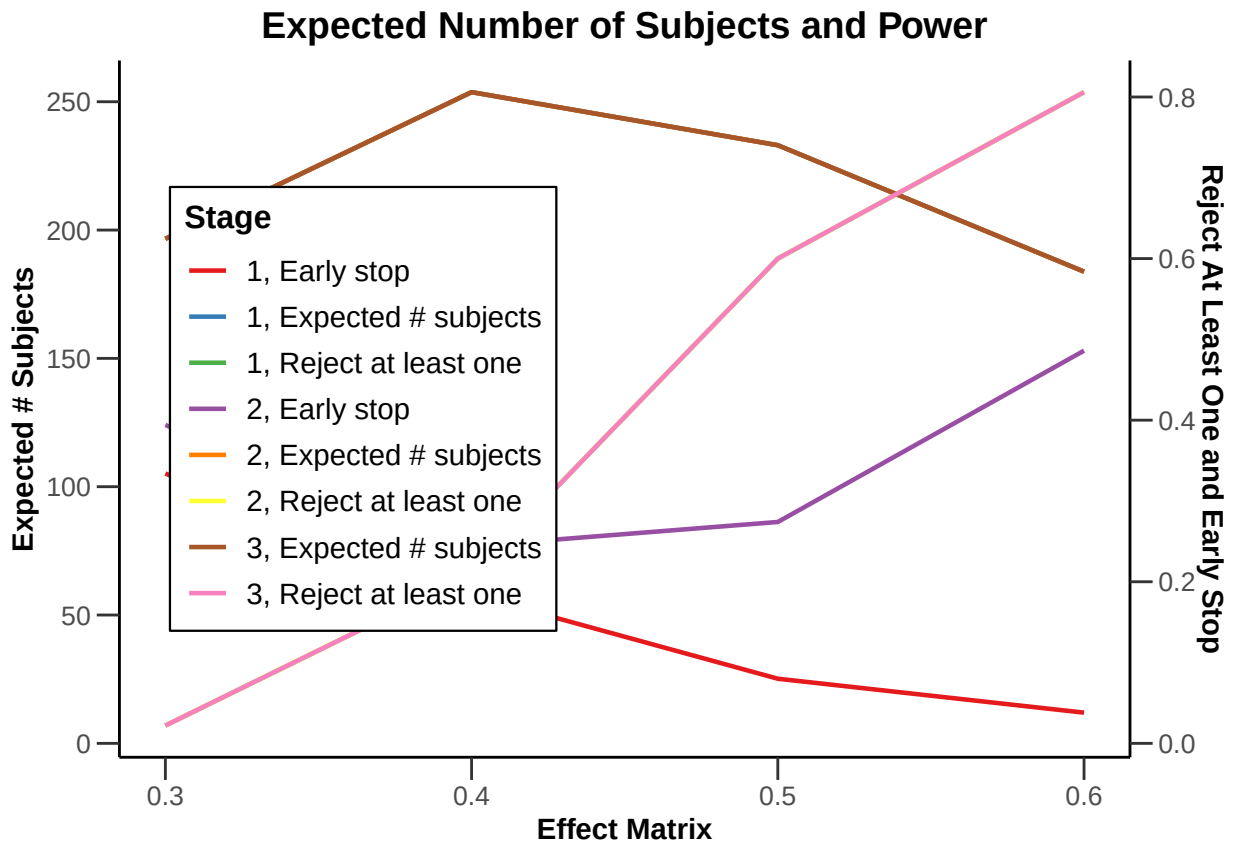

Overall Success
Overall Success
0.8
0.6
0.4
0.2
0.0
0.3
0.4
0.5
0.6
Maximum Effect

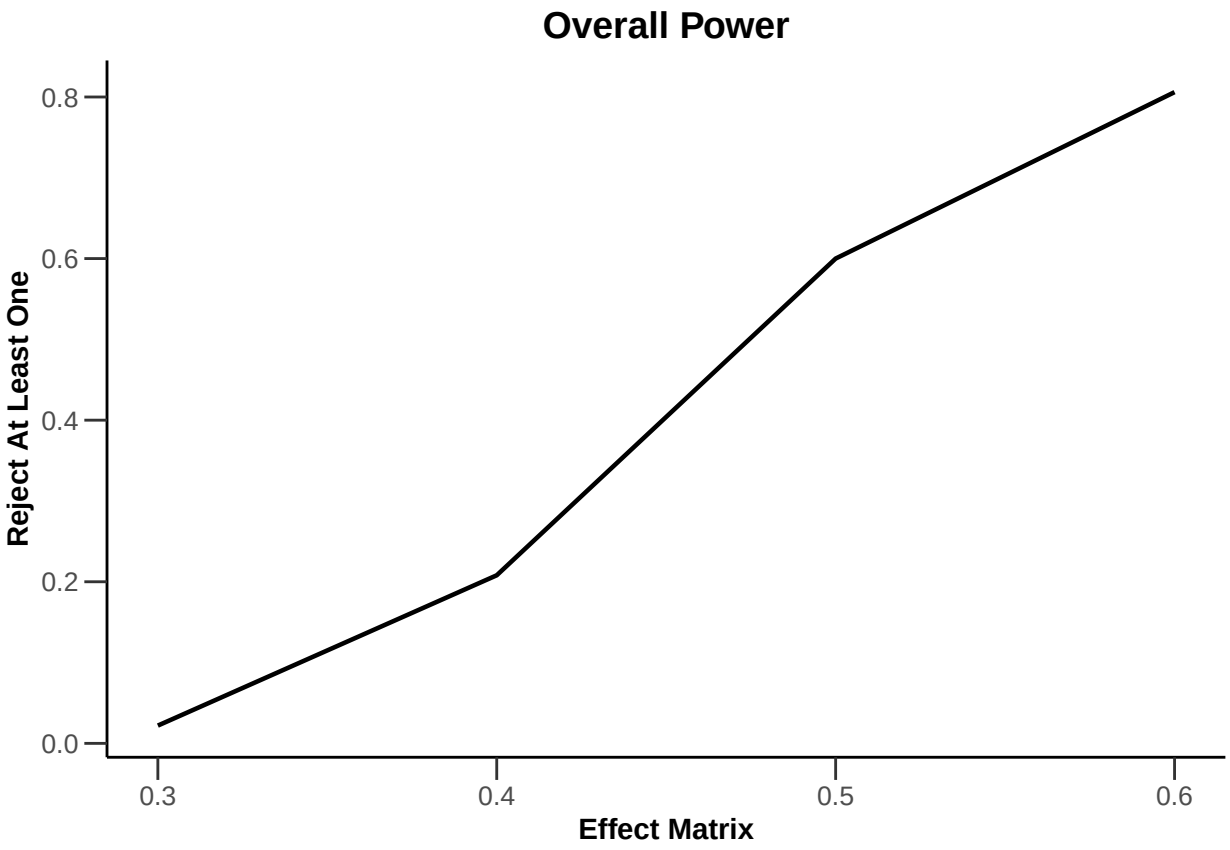

Success per Stage
Stage
1
2
3
Success
0.0
0.1
0.2
0.3
0.4
0.3
0.4
0.5
0.6
Maximum Effect

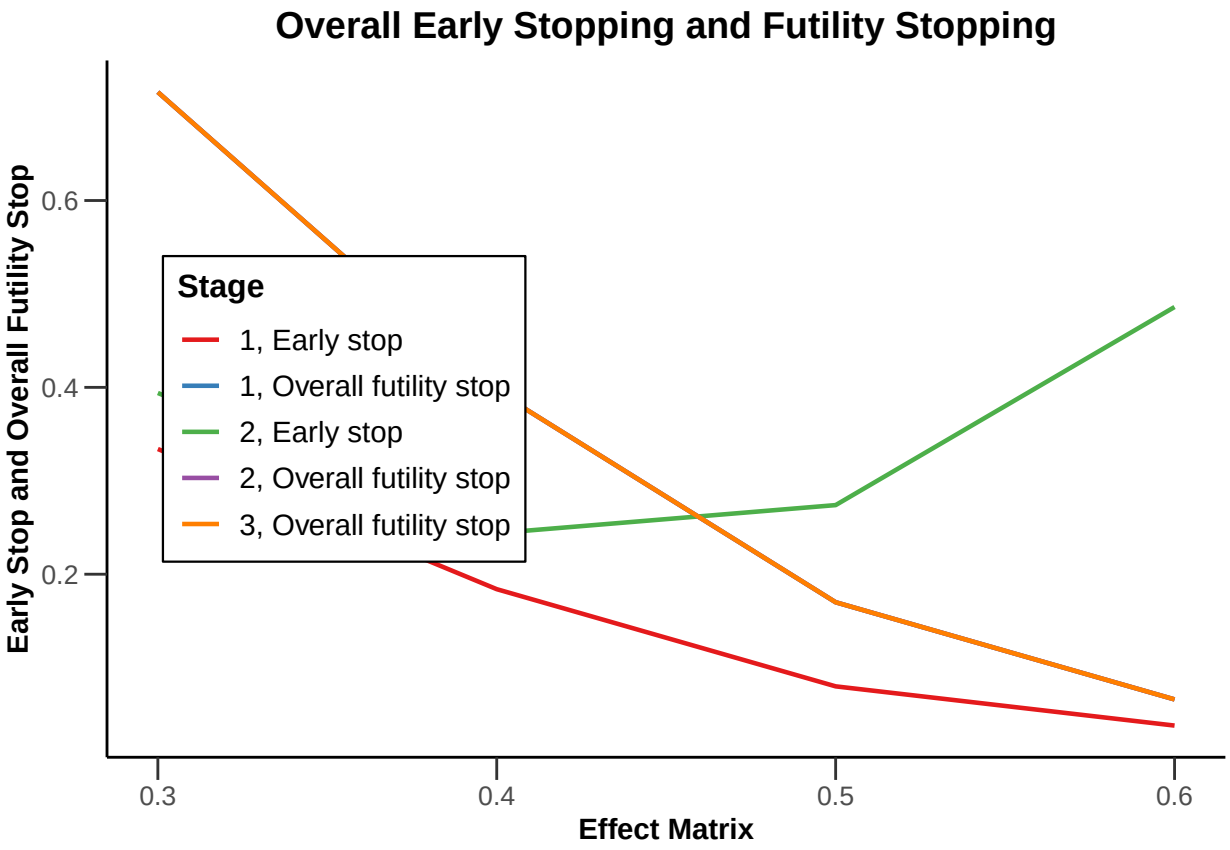

Selected Arms per Stage
Arm, Stage
1, 2
1, 3
2, 2
2, 3
3, 2
3, 3
4, 2
4, 3
Selected Arms
0.4
0.3
0.2
0.1
0.3
0.4
0.5
0.6
Maximum Effect

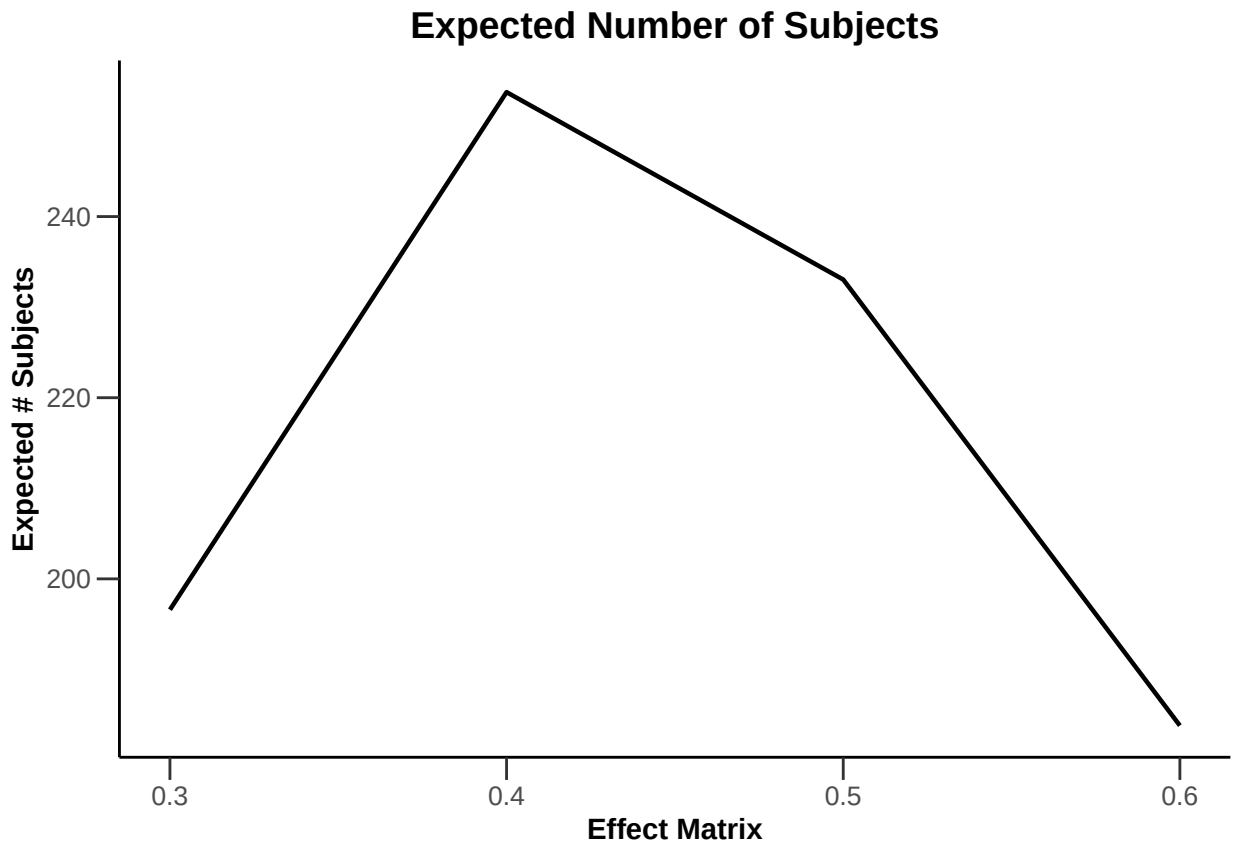

Rejected Arms per Stage
Arm, Stage
1, 1
1, 2
1, 3
2, 1
2, 2
2, 3
3, 1
3, 2
3, 3
4, 1
4, 2
4, 3
Rejected Arms
0.0
0.1
0.2
0.3
0.3
0.4
0.5
0.6
Maximum Effect


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.8
0.6
0.4
0.2
0.0
0.3
0.4
0.5
0.6
Maximum Effect


Expected Number of Subjects and Power
Expected # Subjects
Reject At Least One and Early Stop
Effect Matrix
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
0.3
0.4
0.5
0.6
Stage
1, Early stop
1, Expected # subjects
1, Reject at least one
2, Early stop
2, Expected # subjects
2, Reject at least one
3, Expected # subjects
3, Reject at least one

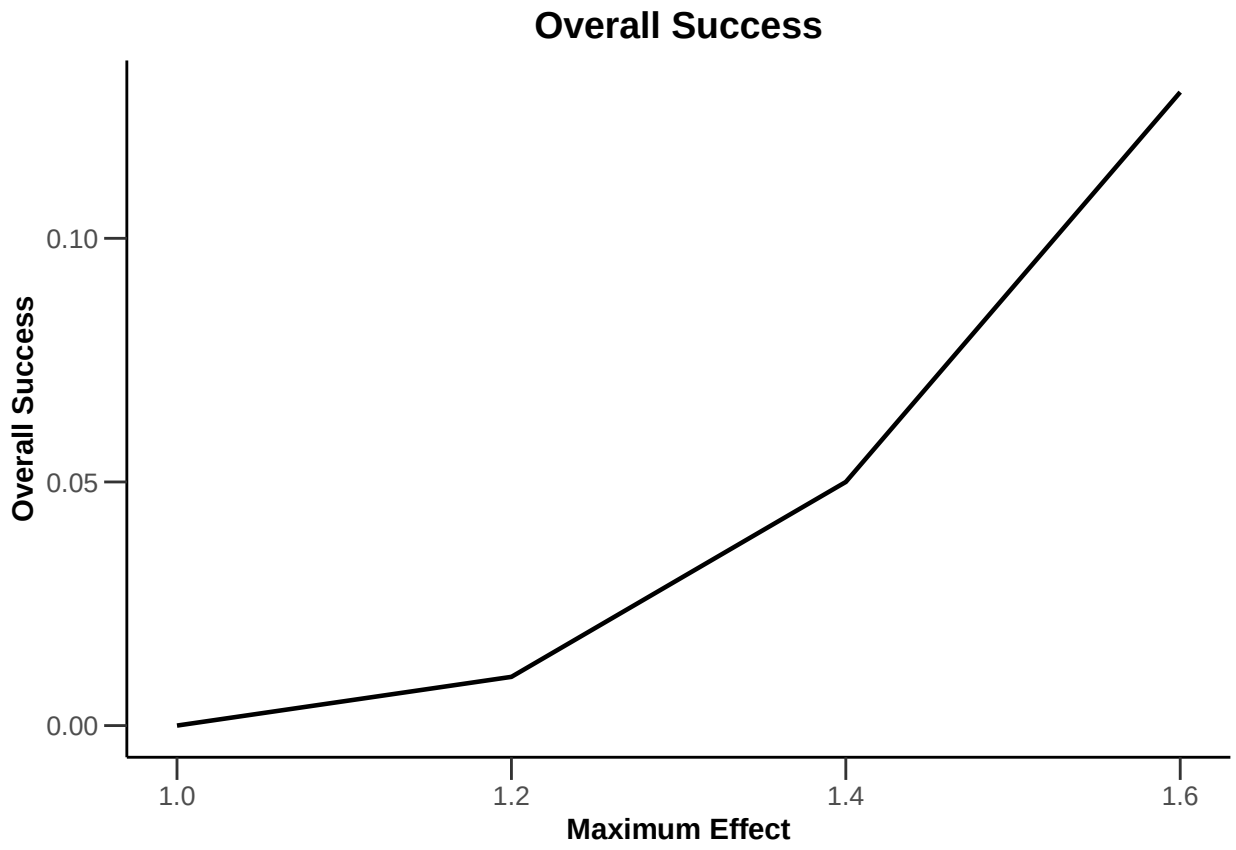

Overall Power
Reject At Least One
0.0
0.2
0.4
0.6
0.8
0.3
0.4
0.5
0.6
Effect Matrix

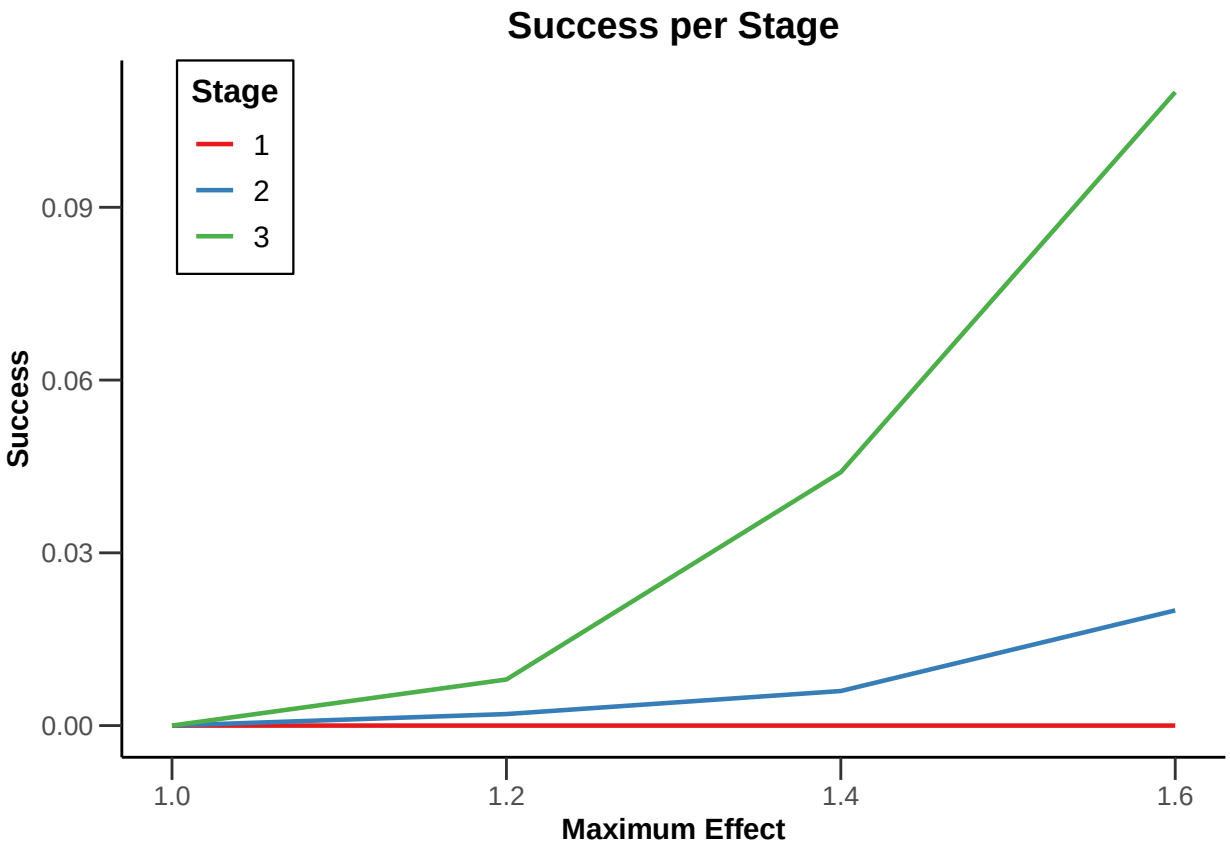

Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
0.6
0.4
0.2
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Overall futility stop
0.3
0.4
0.5
0.6
Effect Matrix

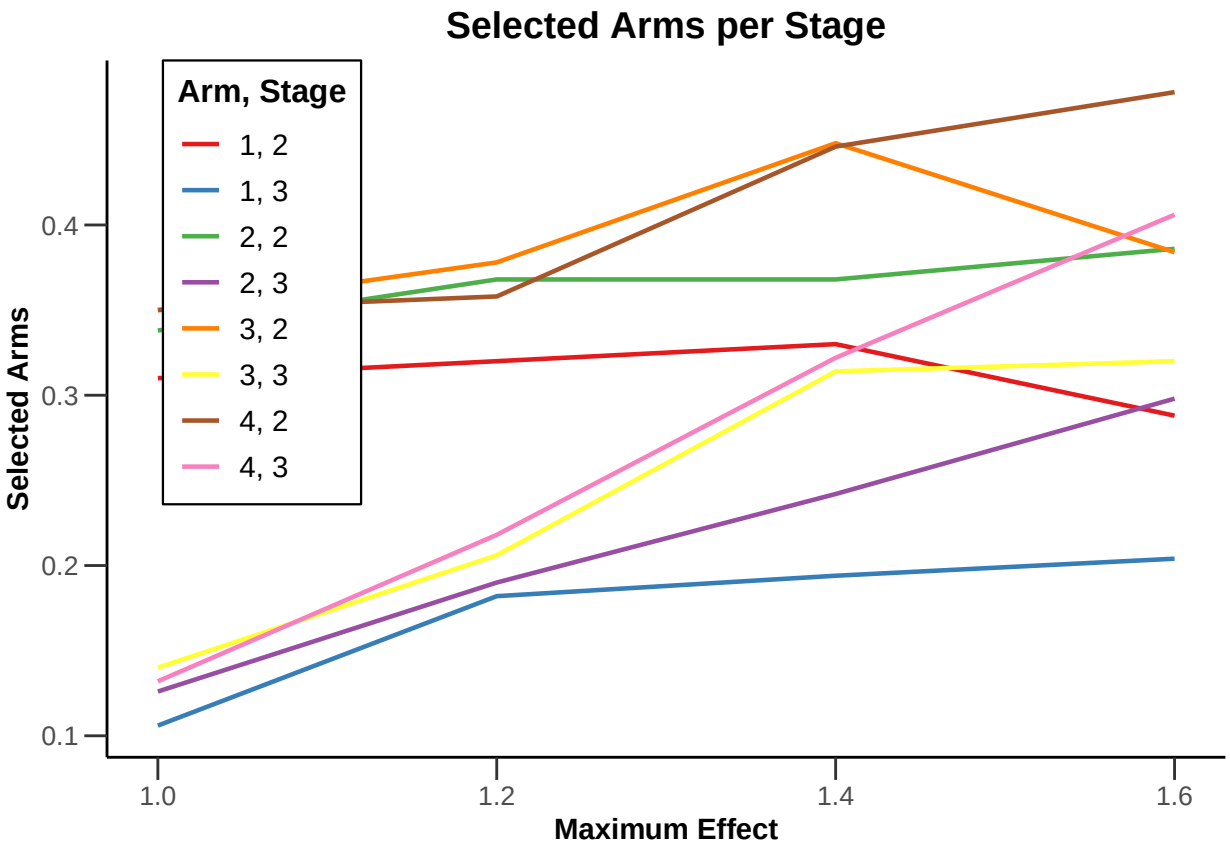


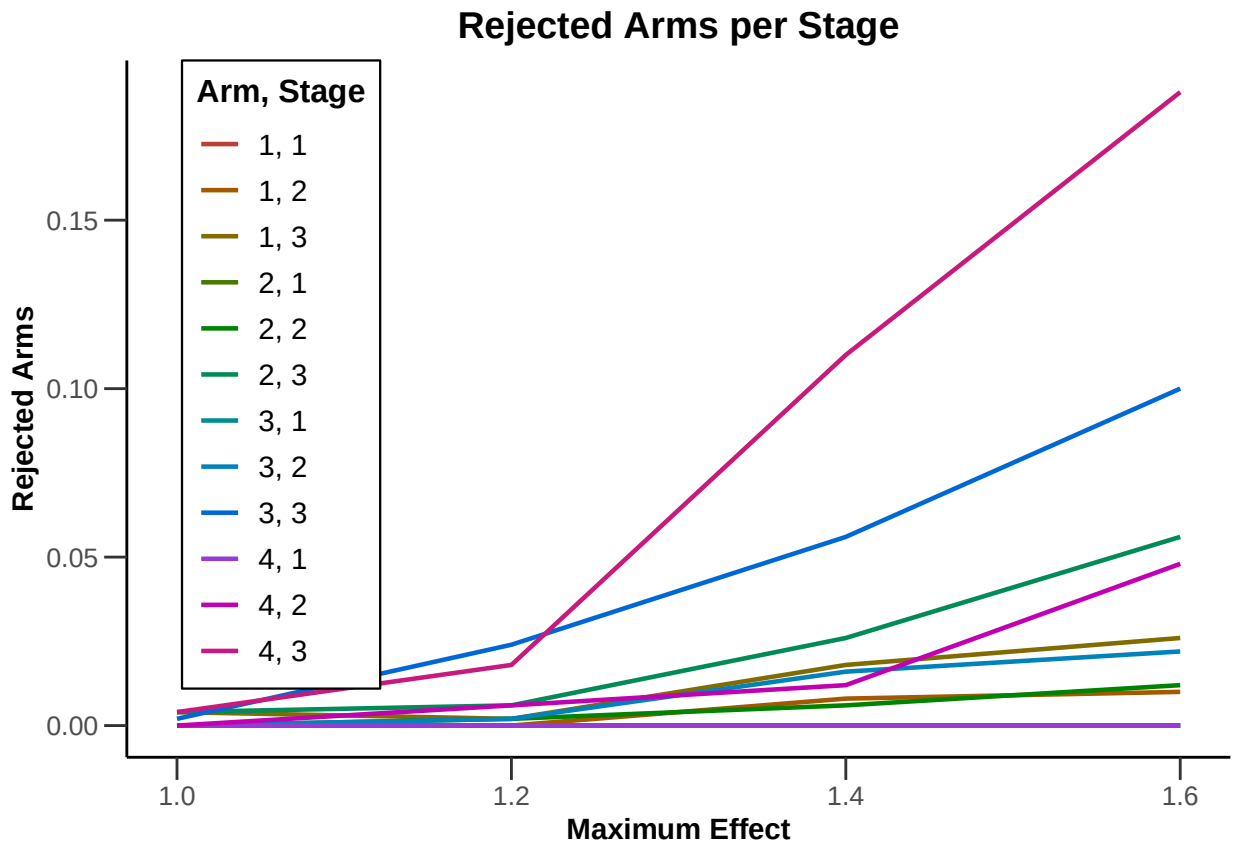
3.3 Simulation results multi-arm - survival

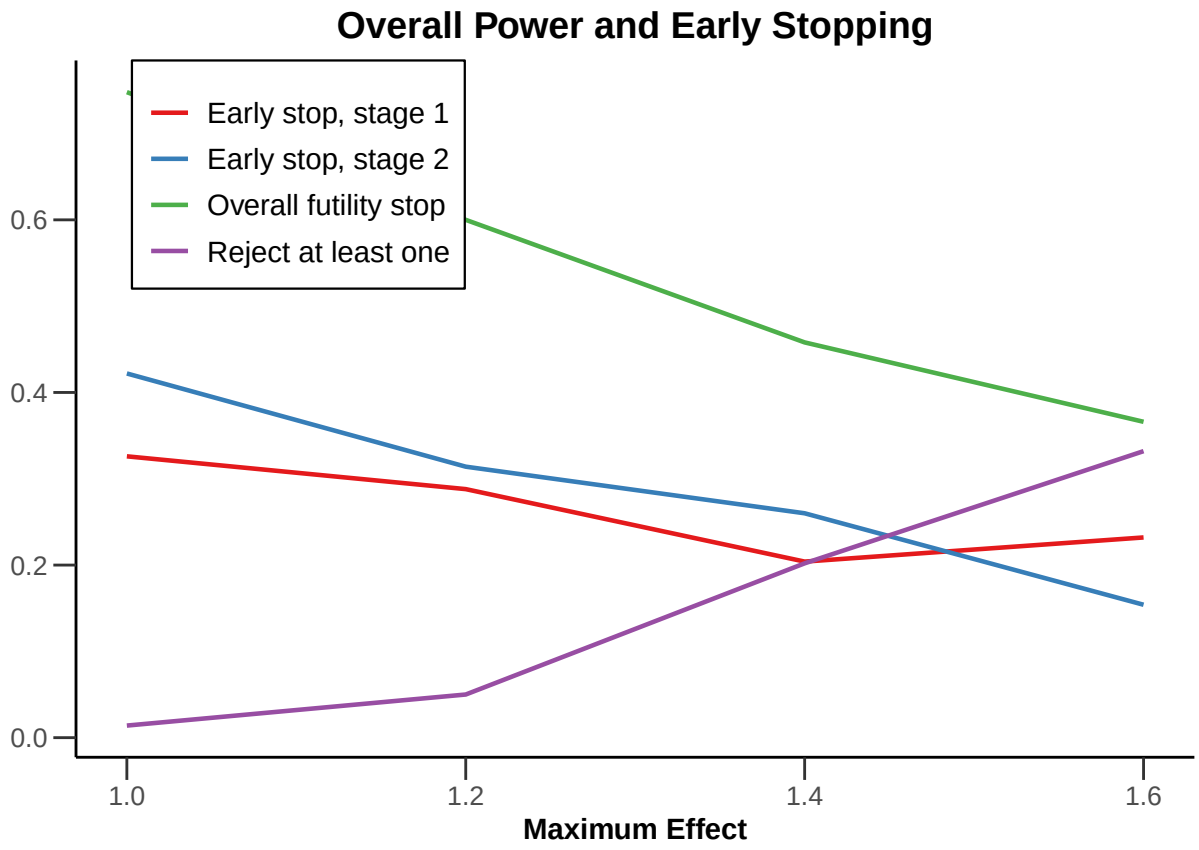
```
design <- getDesignInverseNormal(
    informationRates = c(0.2, 0.6, 1),
    futilityBounds = c(-0.5, 0.5)
)
x <- getSimulationMultiArmSurvival(
    design = design, activeArms = 4,
    typeOfSelection = "rBest", rValue = 2, plannedEvents = c(10, 30, 50),
    omegaMaxVector = seq(1, 1.6, 0.2), adaptations = rep(TRUE, 2),
    conditionalPower = 0.8, minNumberOfEventsPerStage = c(10, 4, 4),
    maxNumberOfEventsPerStage = c(10, 100, 100),
    maxNumberOfIterations = 500, seed = 1234567890
)
plot(x, type = "all", grid = 0)
```

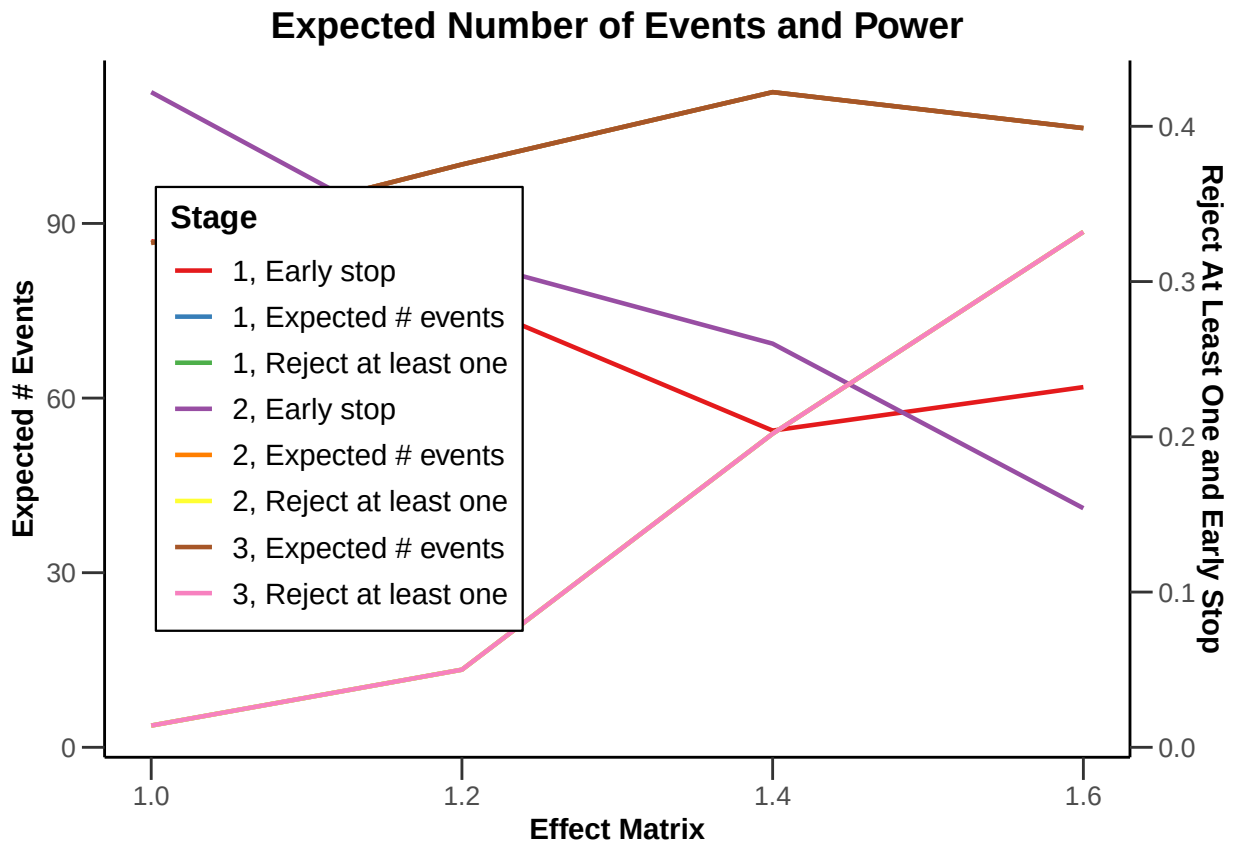

Overall Success
Overall Success
0.10
0.05
0.00
1.0
1.2
1.4
1.6
Maximum Effect

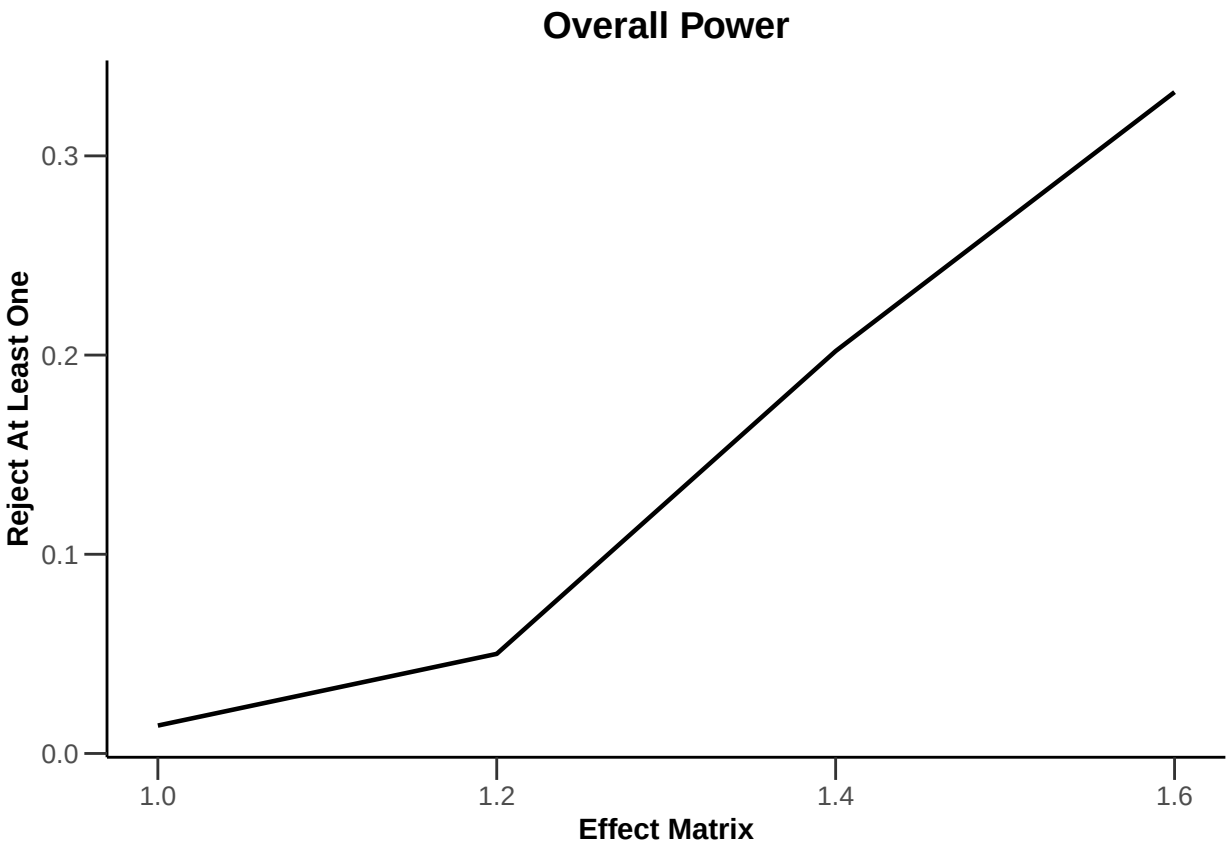

Success per Stage
Stage
1
2
3
Success
0.09
0.06
0.03
0.00
1.0
1.2
1.4
1.6
Maximum Effect

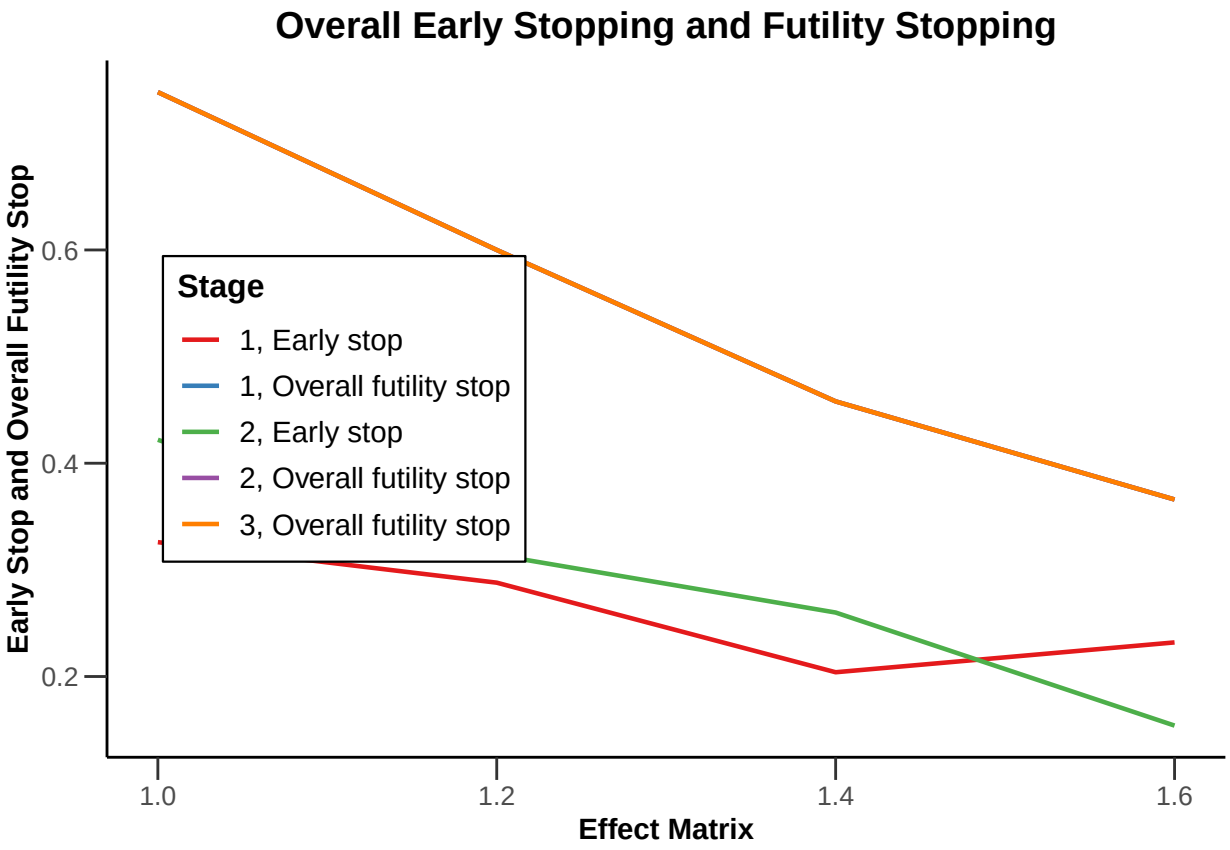

Selected Arms per Stage
Arm, Stage
1, 2
1, 3
2, 2
2, 3
3, 2
3, 3
4, 2
4, 3
Selected Arms
0.4
0.3
0.2
0.1
1.0
1.2
1.4
1.6
Maximum Effect

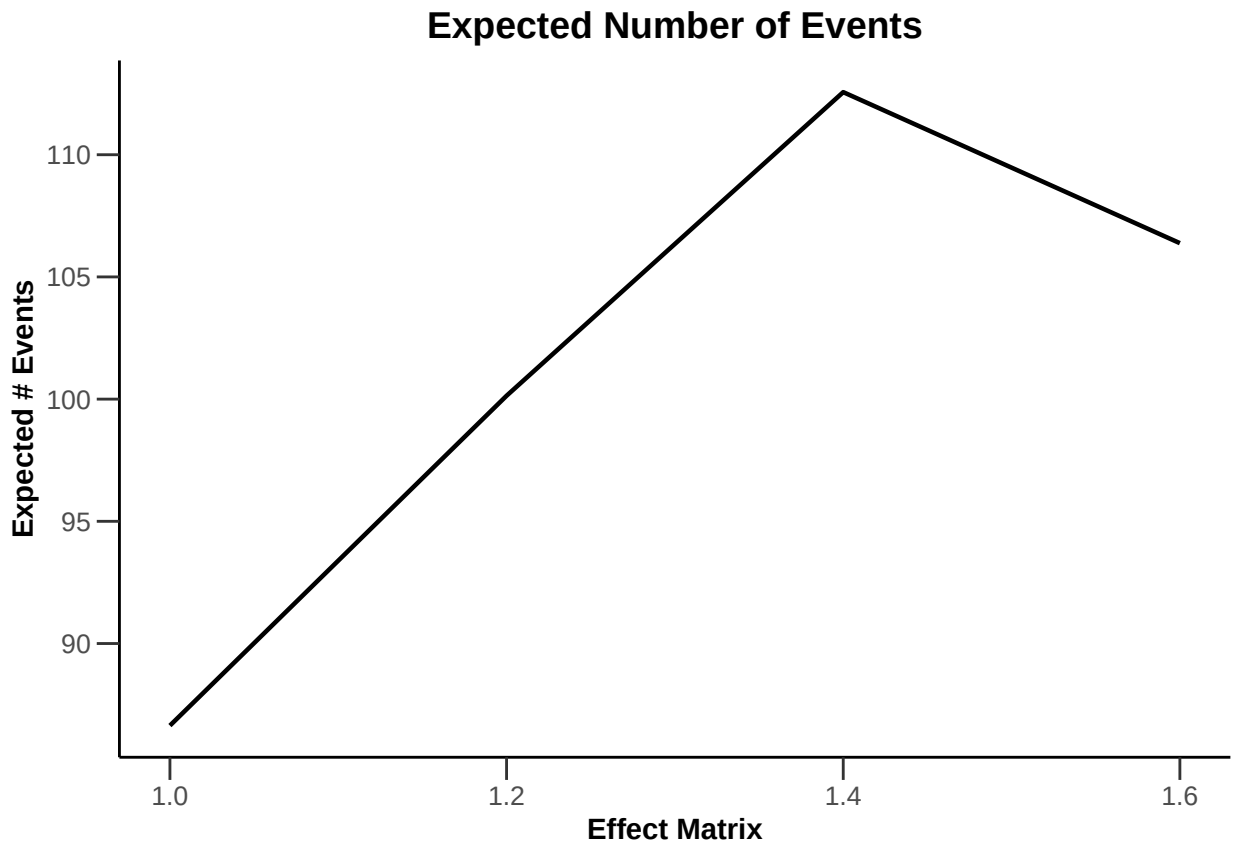

Rejected Arms per Stage
Arm, Stage
1, 1
1, 2
1, 3
2, 1
2, 2
2, 3
3, 1
3, 2
3, 3
4, 1
4, 2
4, 3
Rejected Arms
0.15
0.10
0.05
0.00
1.0
1.2
1.4
1.6
Maximum Effect


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.6
0.4
0.2
0.0
1.0
1.2
1.4
1.6
Maximum Effect


Expected Number of Events and Power
Stage
1, Early stop
1, Expected # events
1, Reject at least one
2, Early stop
2, Expected # events
2, Reject at least one
3, Expected # events
3, Reject at least one
Expected # Events
Reject At Least One and Early Stop
Effect Matrix
0
30
60
90
0.0
0.1
0.2
0.3
0.4
1.0
1.2
1.4
1.6


Overall Power
Reject At Least One
0.0
0.1
0.2
0.3
1.0
1.2
1.4
1.6
Effect Matrix


Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
0.6
0.4
0.2
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Overall futility stop
1.0
1.2
1.4
1.6
Effect Matrix



4 Simulation results enrichment

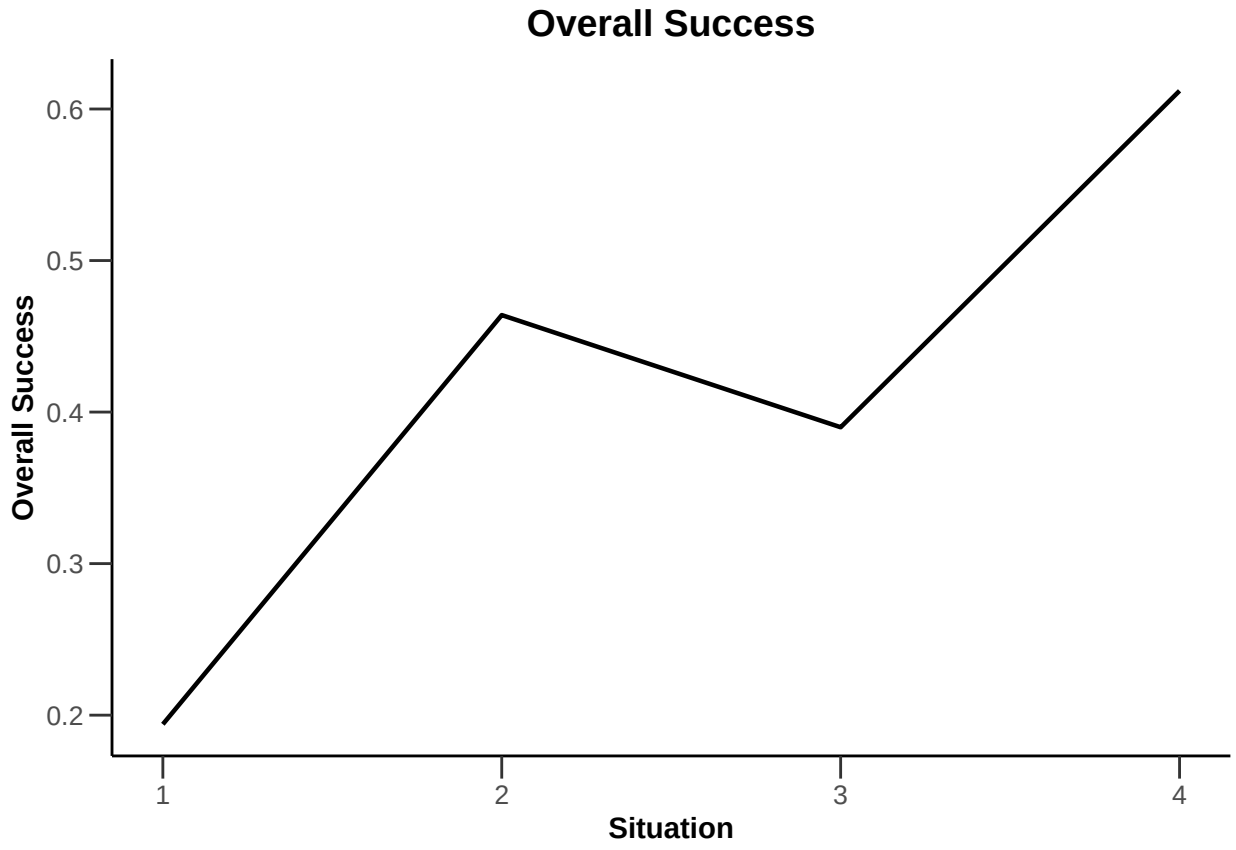
4.1 Simulation results enrichment - means

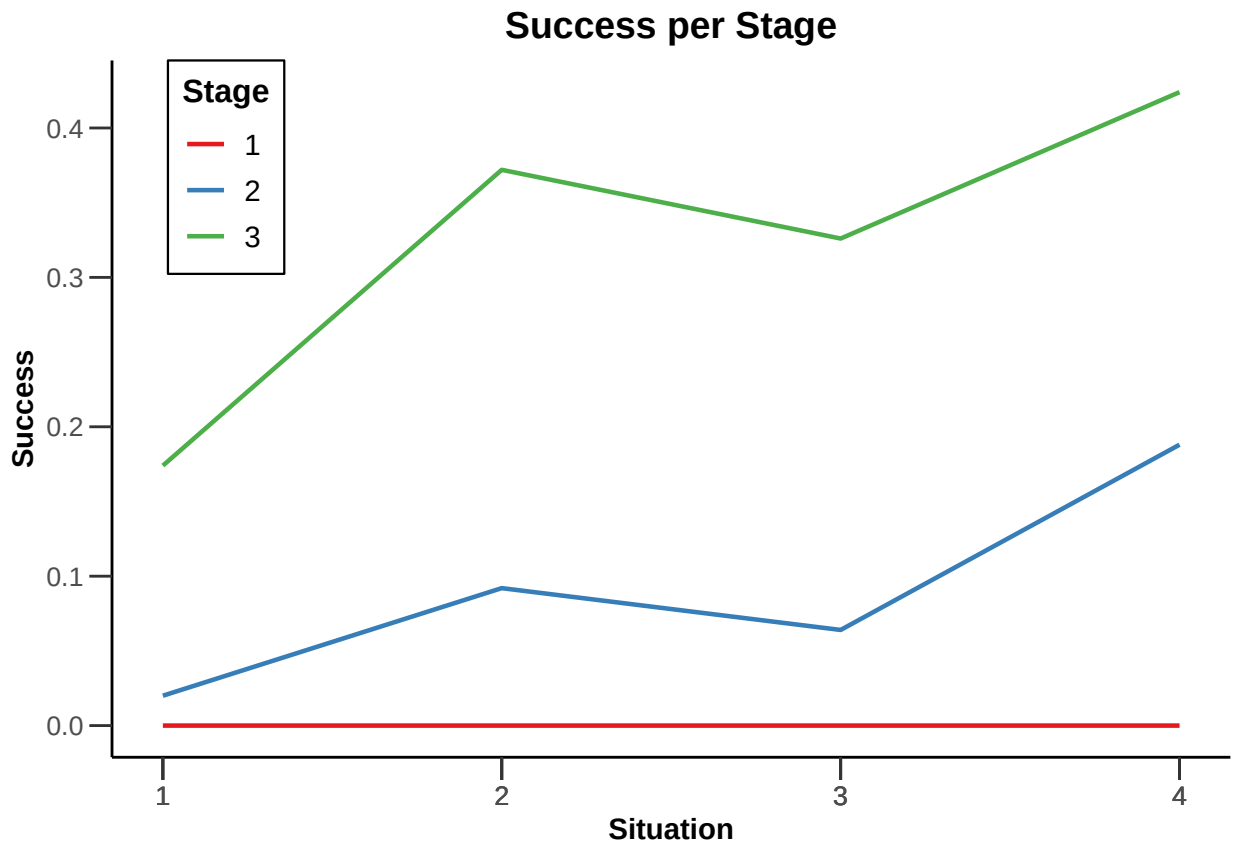
```
design <- getDesignInverseNormal(
    informationRates = c(0.2, 0.6, 1),
    futilityBounds = c(-0.5, 0.5)
)
# Define subgroups and their prevalences
subGroups <- c("S1", "S12", "S2", "R") # fixed names!
prevalences <- c(0.2, 0.3, 0.4, 0.1)

effectR <- 1.5
effectS12 <- 5
m <- c()
for (effectS1 in seq(0, 5, 5)) {
    for (effectS2 in seq(0, 5, 5)) {
        m <- c(m, effectS1, effectS12, effectS2, effectR)
    }
}
effects <- matrix(m, byrow = TRUE, ncol = 4)
stDev <- 10
# Define effect list
el <- list(
    subGroups = subGroups, prevalences = prevalences,
```

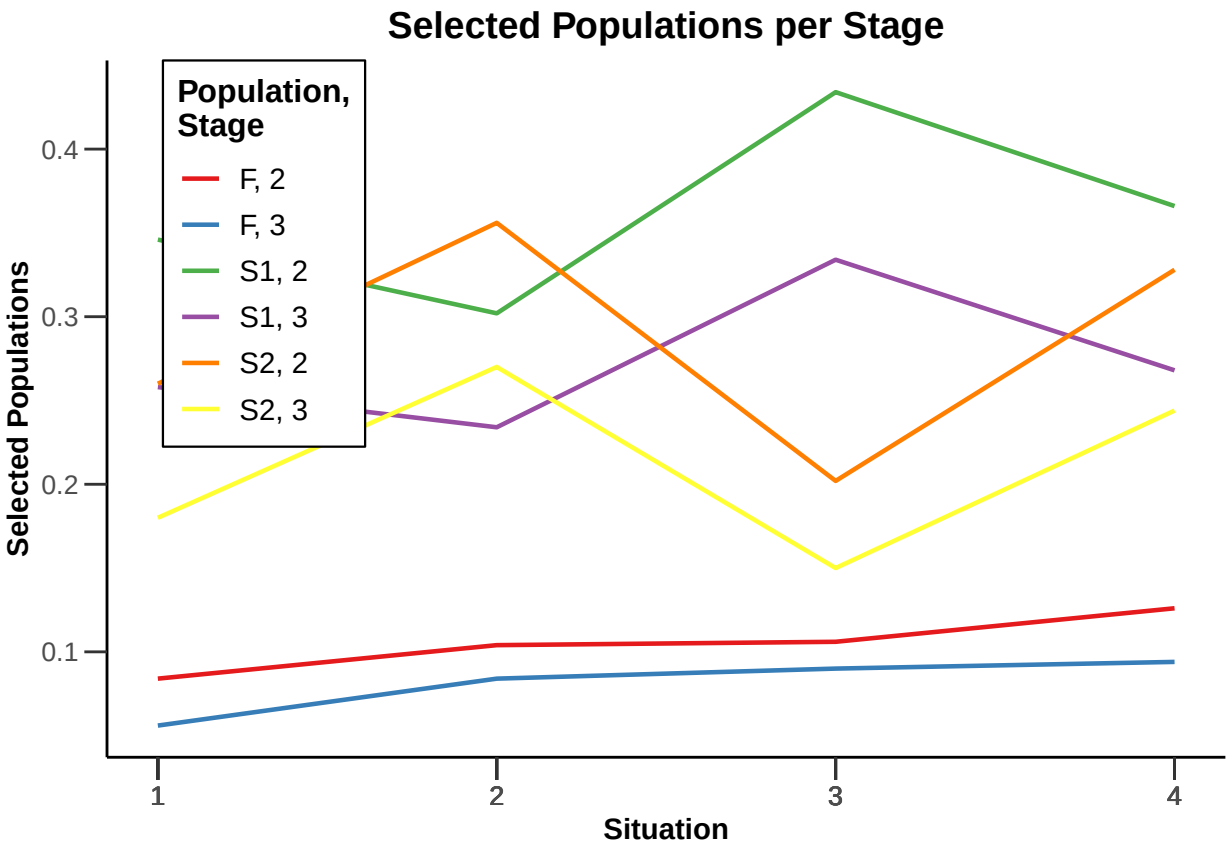
```
    stDevs = stDev, effects = effects
)

x <- getSimulationEnrichmentMeans(
    design = design,
    plannedSubjects = c(10, 30, 50),
    effectList = el,
    adaptations = rep(TRUE, 2),
    conditionalPower = 0.8,
    minNumberOfSubjectsPerStage = c(10, 4, 4),
    maxNumberOfSubjectsPerStage = c(10, 100, 100),
    maxNumberOfIterations = 500,
    seed = 1234567890
)
```

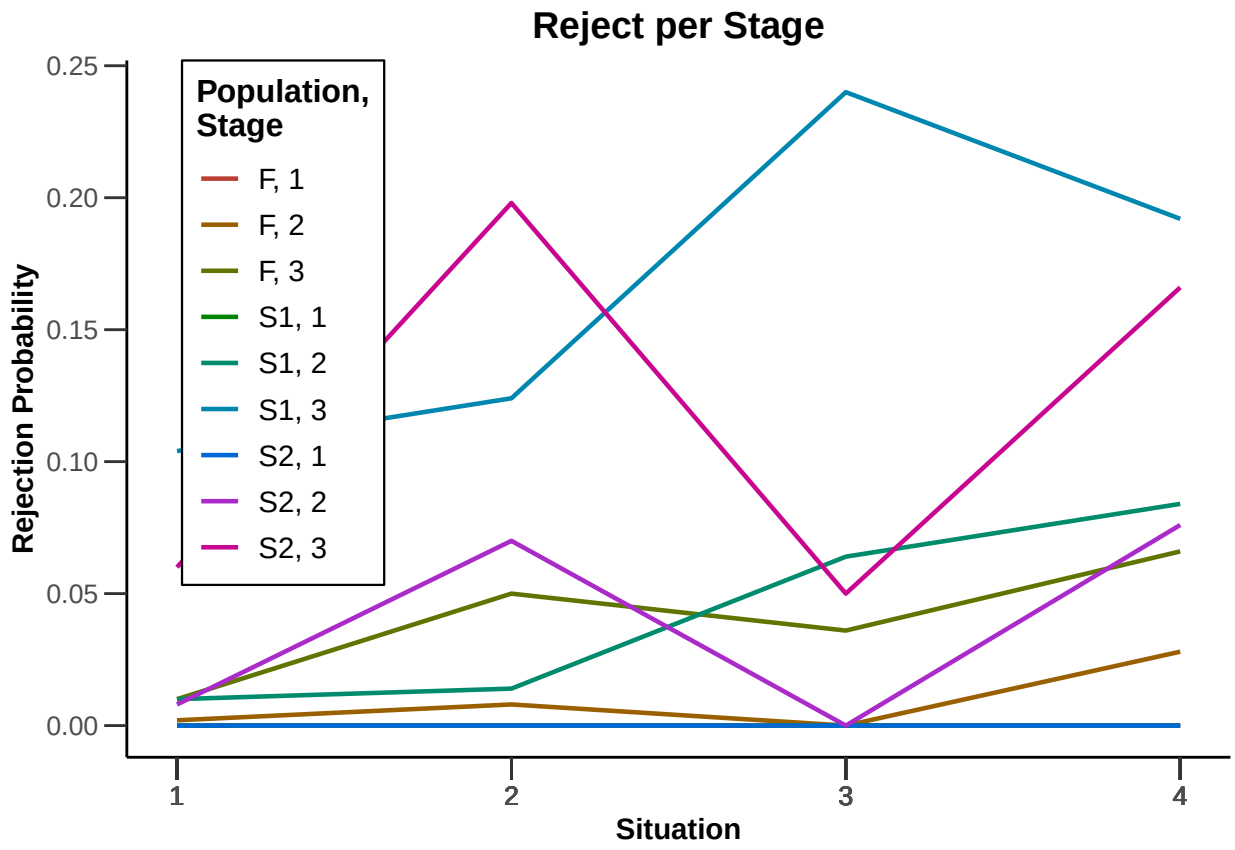
```
## Warning: Simulation of enrichment designs is experimental and hence not fully
## validated (see www.rpact.com/experimental)
```

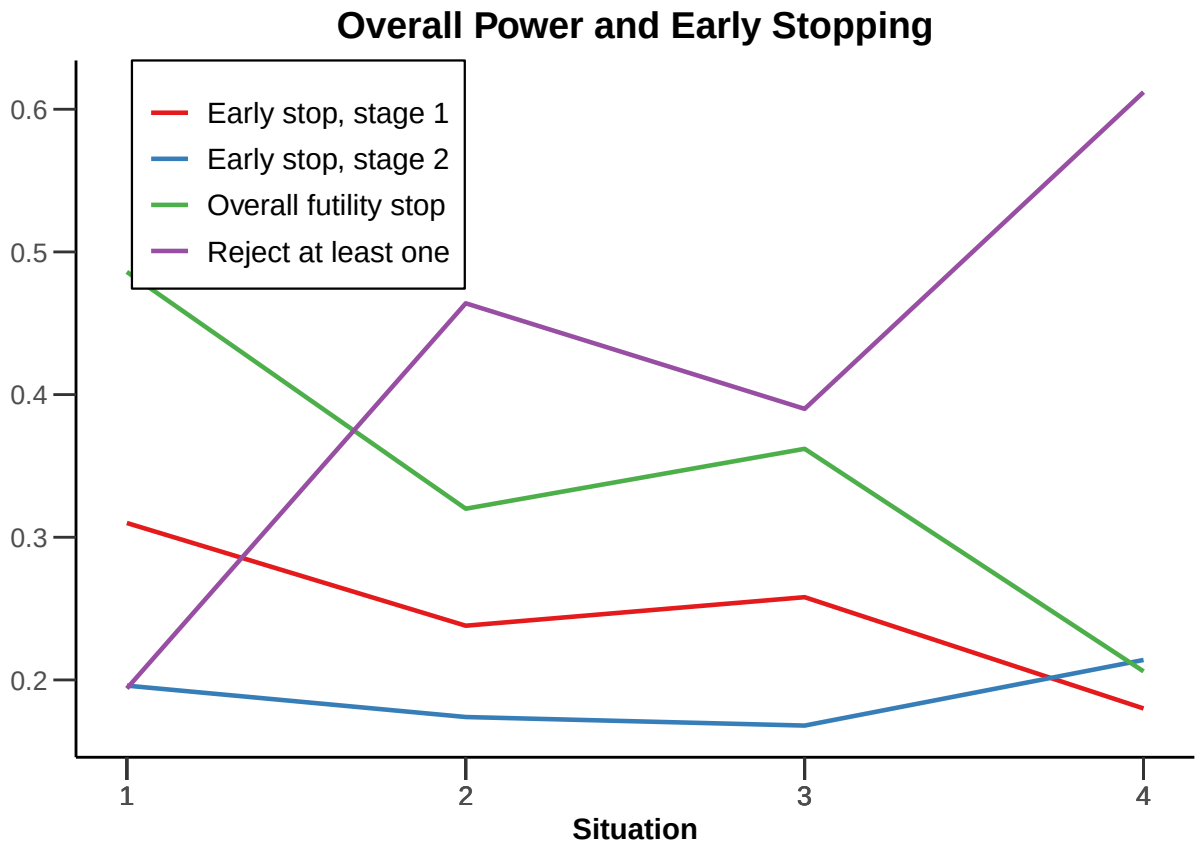
```
plot(x, type = "all", grid = 0)
```

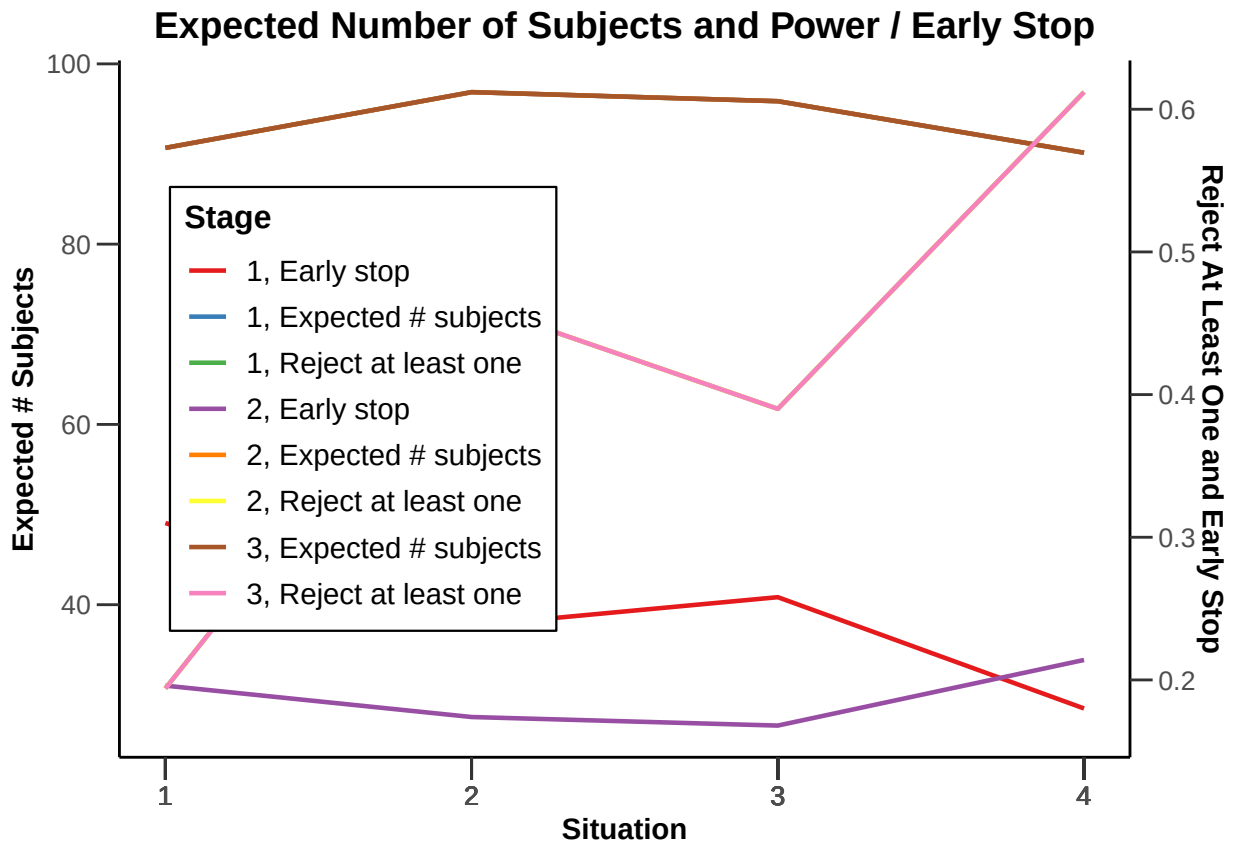


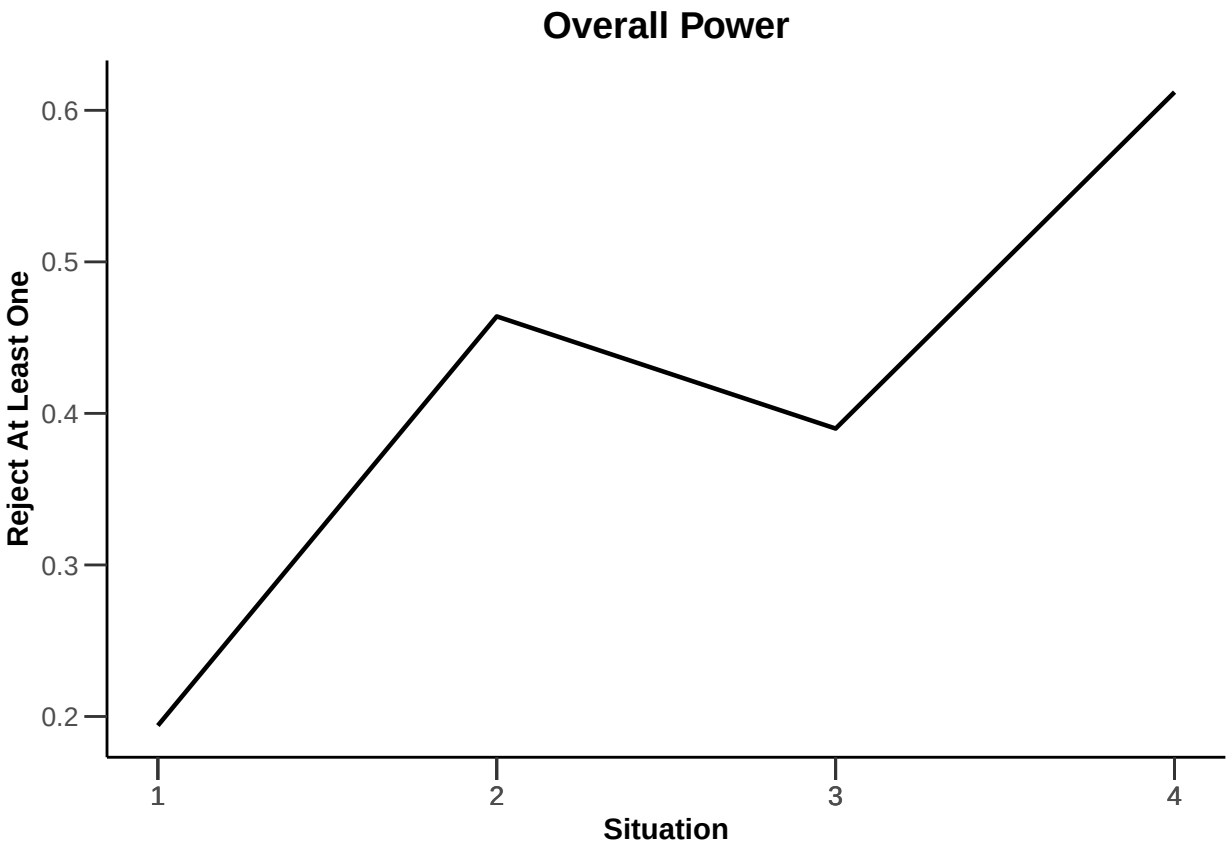

Success per Stage
Stage
1
2
3
Success
0.0
0.1
0.2
0.3
0.4
1
2
3
4
Situation

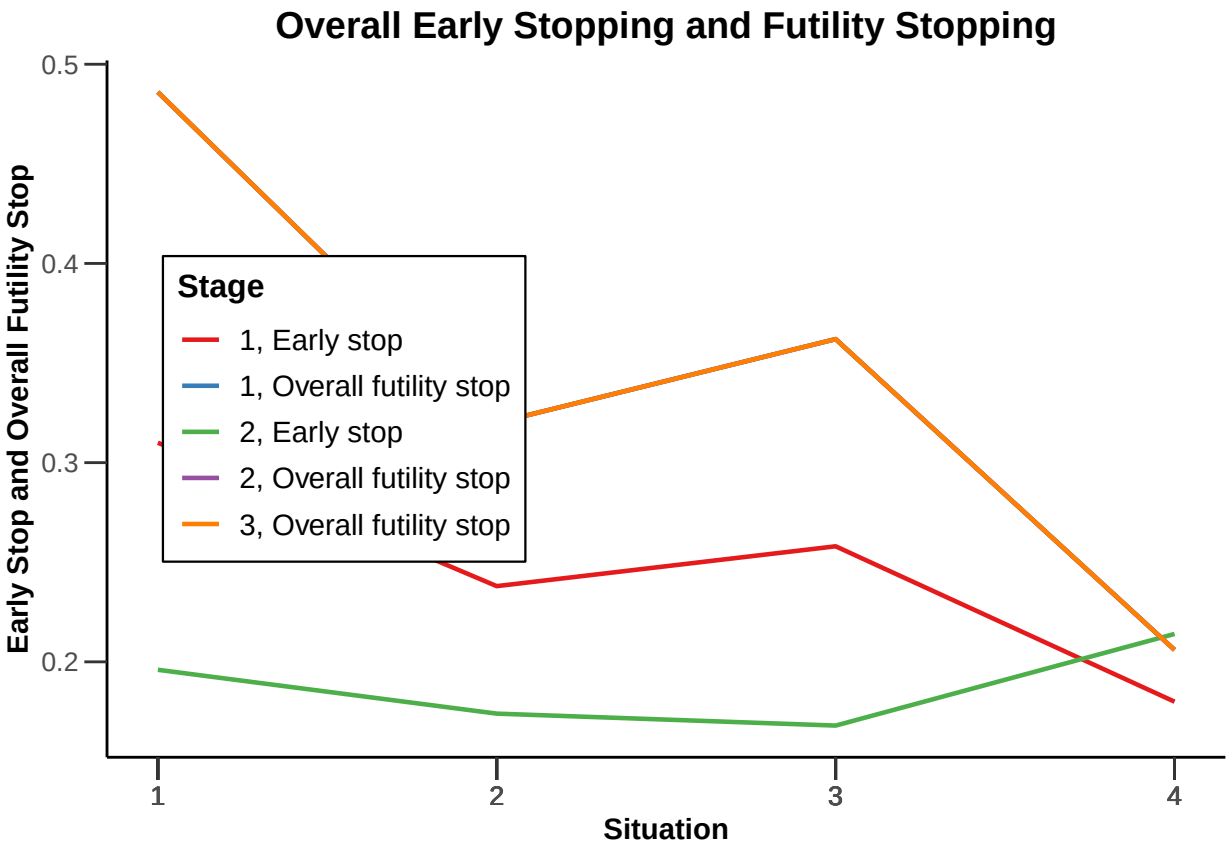

Selected Populations per Stage
Population, Stage
F, 2
F, 3
S1, 2
S1, 3
S2, 2
S2, 3
Selected Populations
0.4
0.3
0.2
0.1
1
2
3
4
Situation

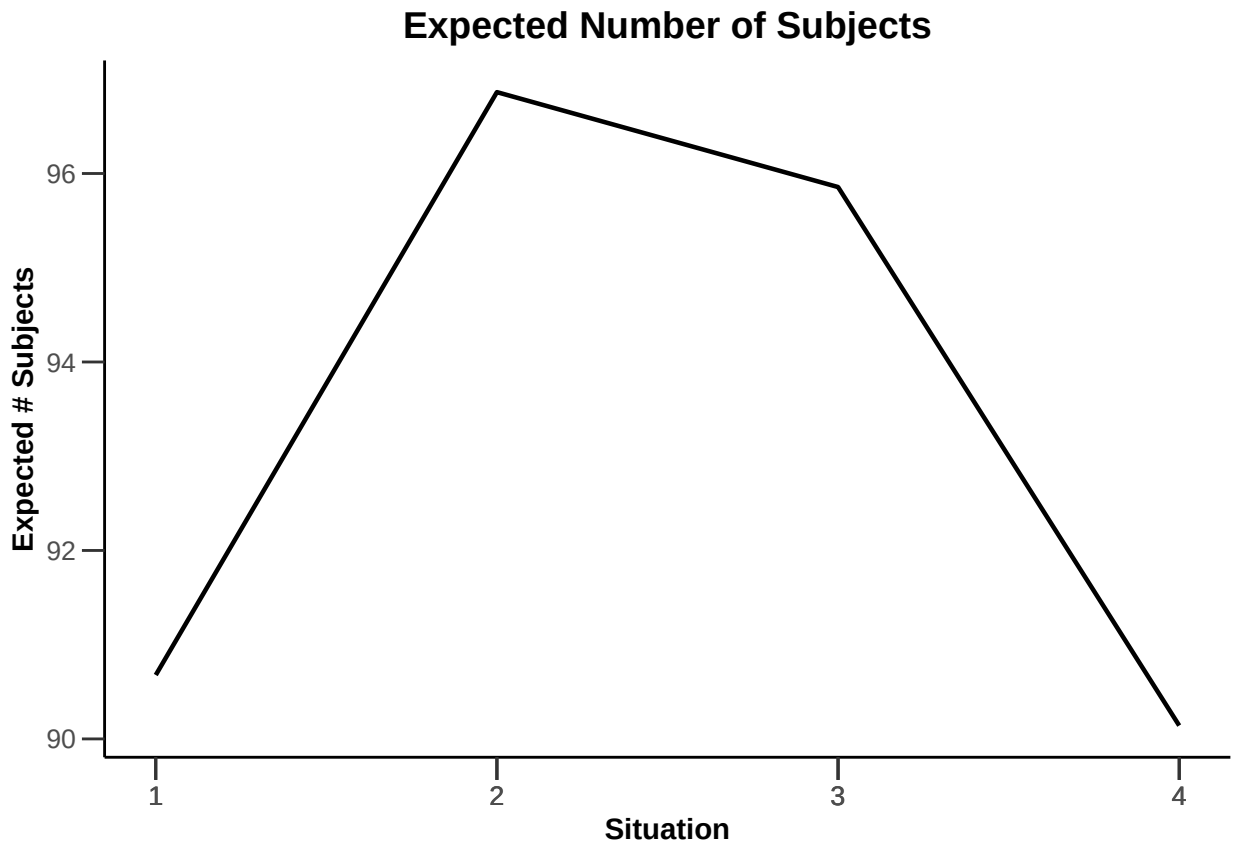

Reject per Stage
Population, Stage
F, 1
F, 2
F, 3
S1, 1
S1, 2
S1, 3
S2, 1
S2, 2
S2, 3
Rejection Probability
0.25
0.20
0.15
0.10
0.05
0.00
1
2
3
4
Situation


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.6
0.5
0.4
0.3
0.2
1
2
3
4
Situation


Expected Number of Subjects and Power / Early Stop
Expected # Subjects
Reject At Least One and Early Stop
Situation
Stage
1, Early stop
1, Expected # subjects
1, Reject at least one
2, Early stop
2, Expected # subjects
2, Reject at least one
3, Expected # subjects
3, Reject at least one
100
80
60
40
0.6
0.5
0.4
0.3
0.2
1
2
3
4


Overall Power
Reject At Least One
0.6
0.5
0.4
0.3
0.2
1
2
3
4
Situation


Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
0.5
0.4
0.3
0.2
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Overall futility stop
1
2
3
4
Situation



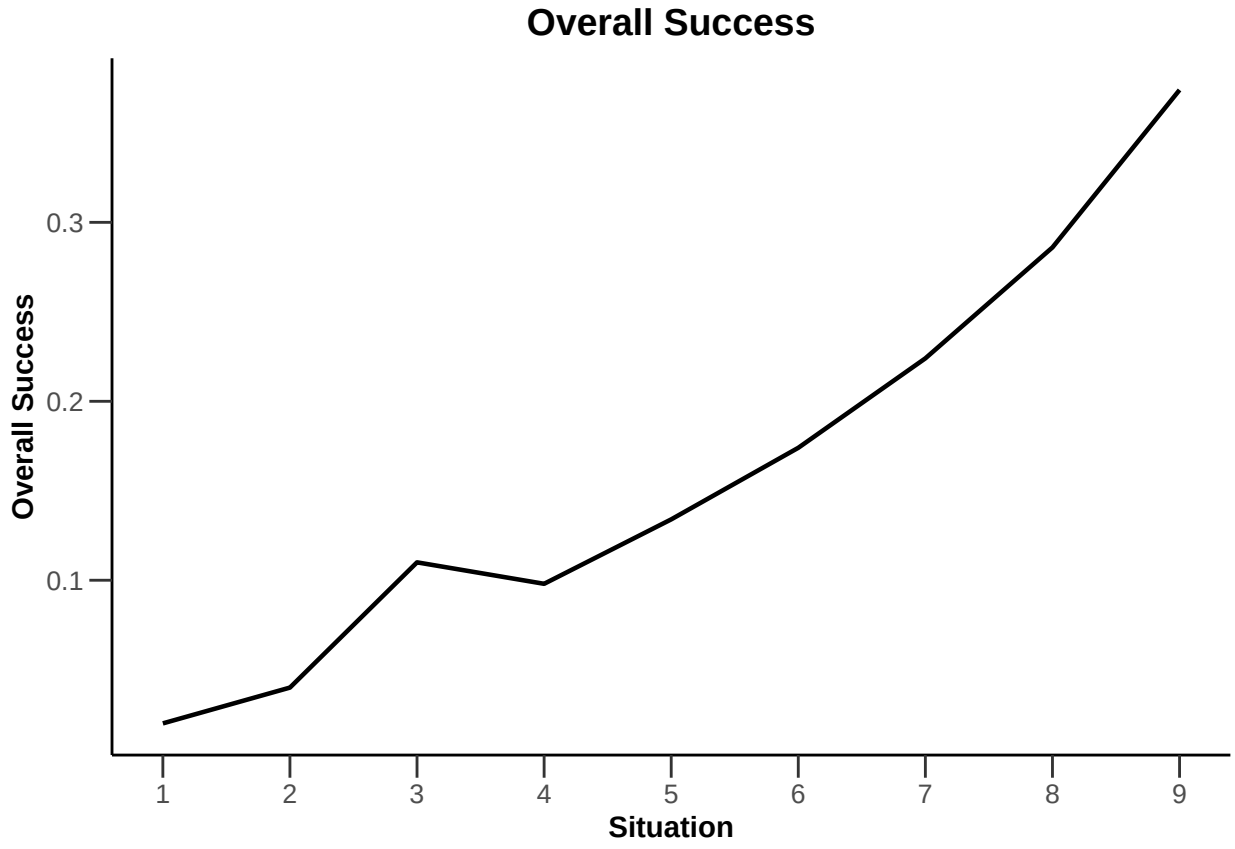
4.2 Simulation results enrichment - rates

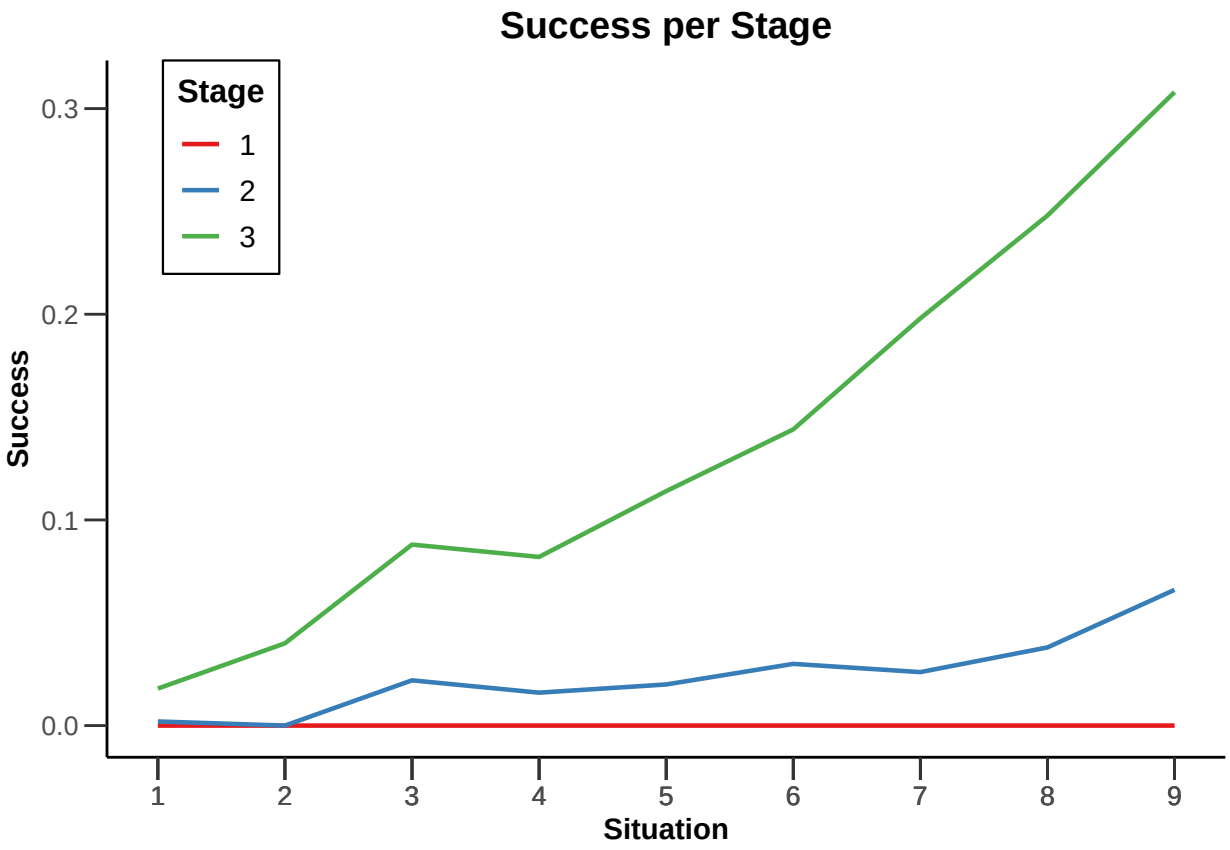
```
design <- getDesignInverseNormal(
    informationRates = c(0.2, 0.6, 1),
    futilityBounds = c(-0.5, 0.5)
)
# Define effect list
subGroups <- c("S", "R")
prevalences <- c(0.4, 0.6)
piControl <- c(0.1, 0.4)
range1 <- piControl[1] + seq(0.0, 0.2, 0.1)
range2 <- piControl[2] + seq(0.0, 0.2, 0.1)
piTreatments <- c()
for (x1 in range1) {
    for (x2 in range2) {
        piTreatments <- c(piTreatments, x1, x2)
    }
}
el <- list(
    subGroups = subGroups,
    prevalences = prevalences,
    piControl = piControl,
    piTreatments = matrix(piTreatments,
        byrow = TRUE, ncol = 2
    )
```

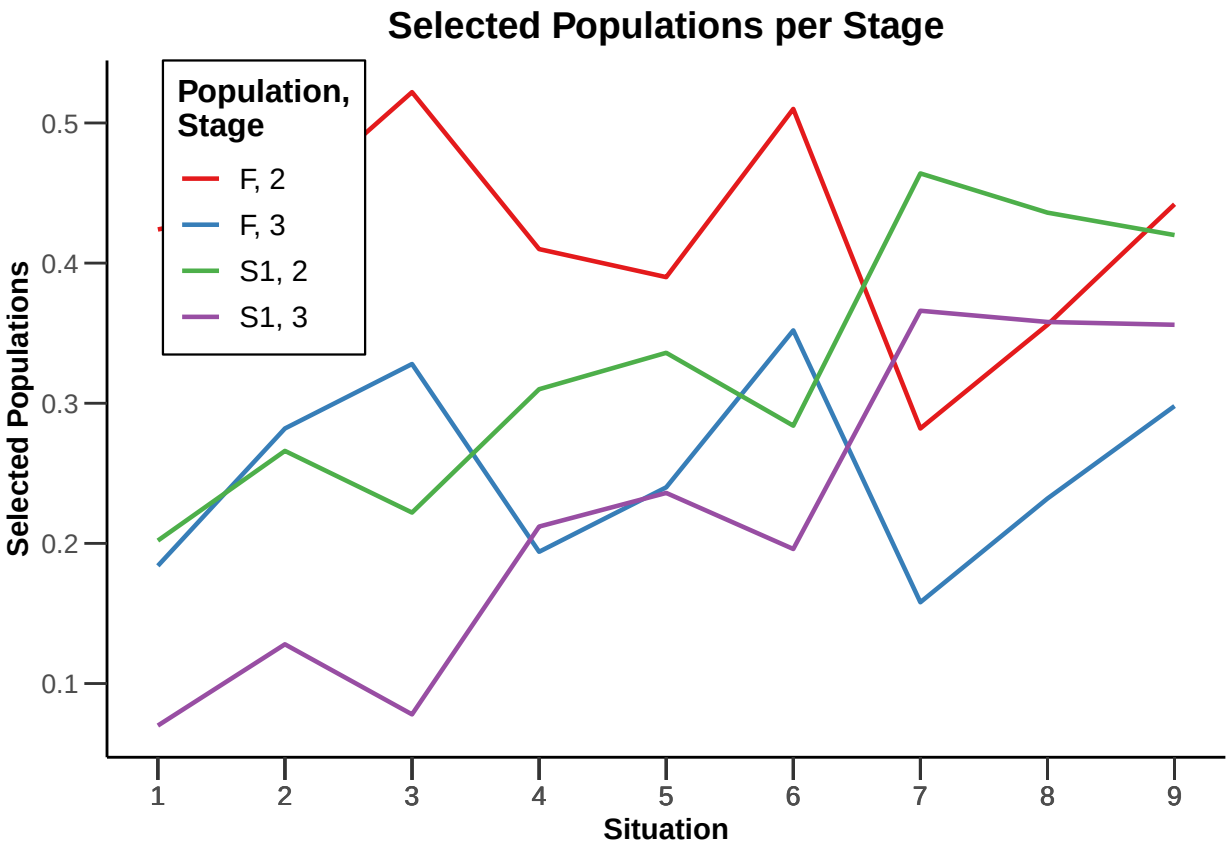
```
)

x <- getSimulationEnrichmentRates(
    design = design,
    plannedSubjects = c(10, 30, 50),
    effectList = el,
    maxNumberOfIterations = 500,
    seed = 1234567890
)
```

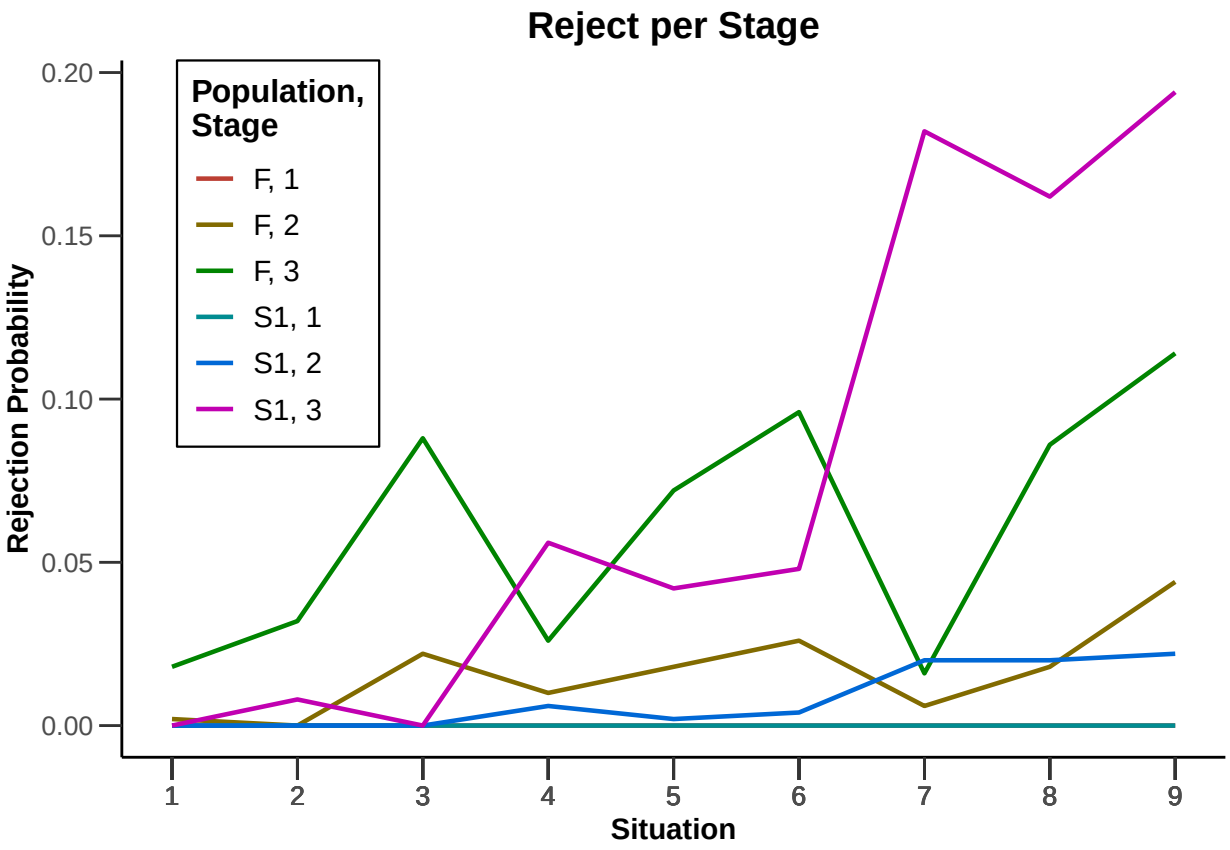
```
## Warning: Simulation of enrichment designs is experimental and hence not fully
## validated (see www.rpact.com/experimental)
```

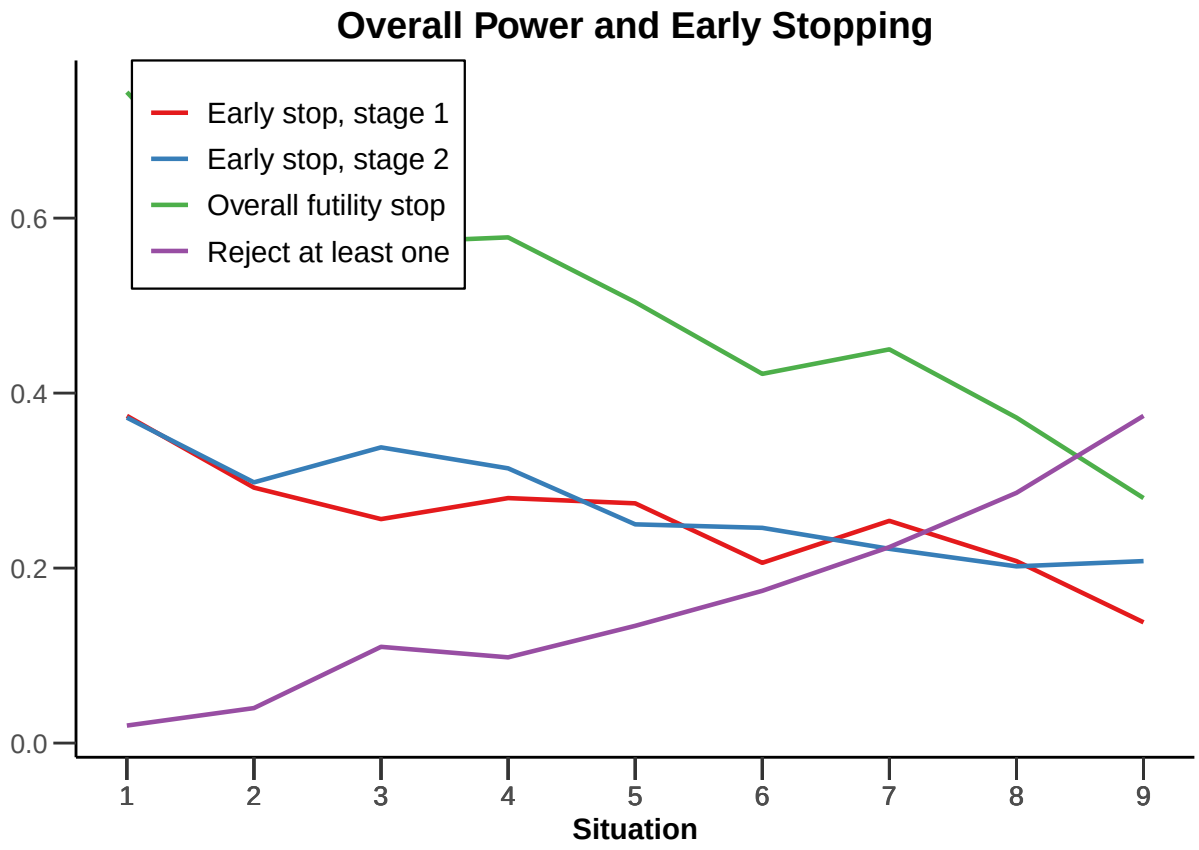
```
plot(x, type = "all", grid = 0)
```

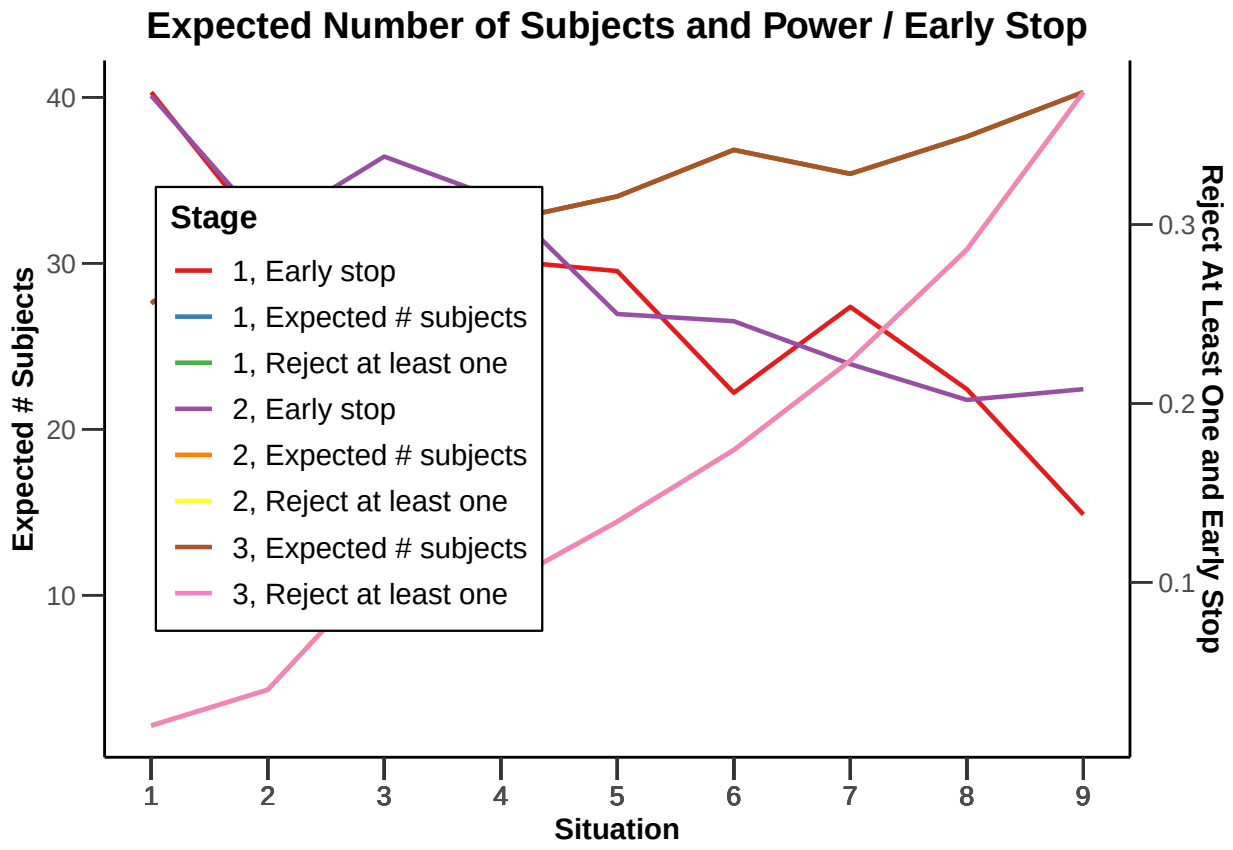


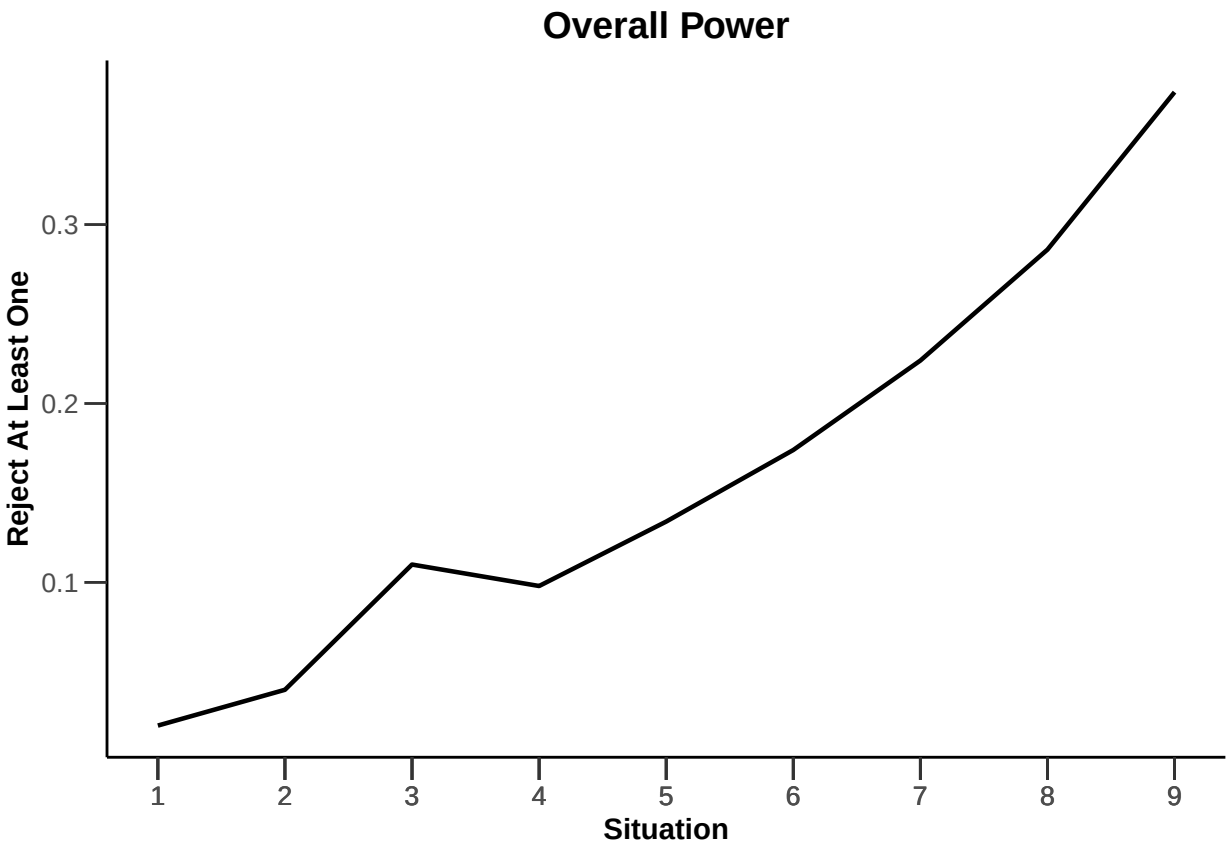

Success per Stage
Stage
1
2
3
Success
0.0
0.1
0.2
0.3
1
2
3
4
5
6
7
8
9
Situation

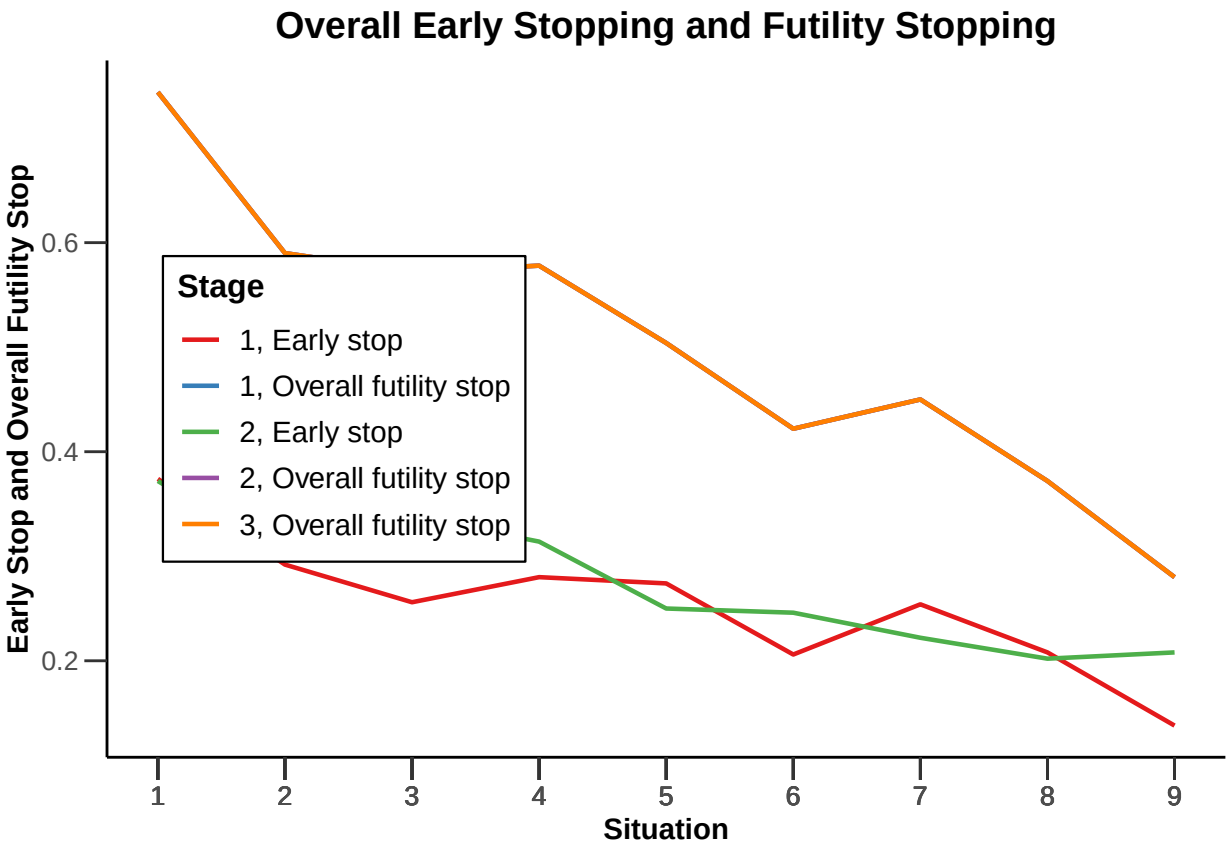

Selected Populations per Stage
Population, Stage
F, 2
F, 3
S1, 2
S1, 3
Selected Populations
0.5
0.4
0.3
0.2
0.1
1
2
3
4
5
6
7
8
9
Situation

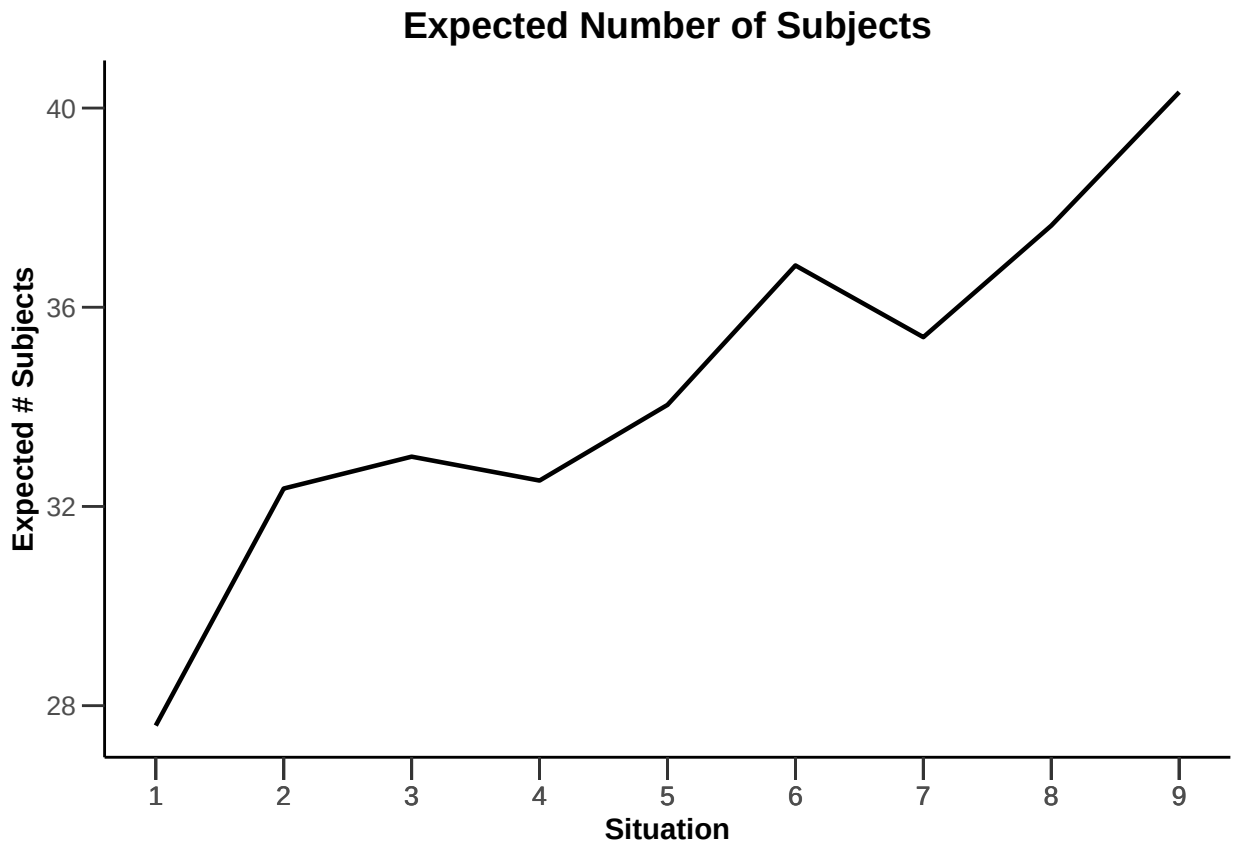

Reject per Stage
Population, Stage
F, 1
F, 2
F, 3
S1, 1
S1, 2
S1, 3
Rejection Probability
0.20
0.15
0.10
0.05
0.00
1
2
3
4
5
6
7
8
9
Situation


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
Situation


Expected Number of Subjects and Power / Early Stop
Expected # Subjects
40
30
20
10
Reject At Least One and Early Stop
0.3
0.2
0.1
Stage
1, Early stop
1, Expected # subjects
1, Reject at least one
2, Early stop
2, Expected # subjects
2, Reject at least one
3, Expected # subjects
3, Reject at least one
1
2
3
4
5
6
7
8
9
Situation


Overall Power
Reject At Least One
0.3
0.2
0.1
1
2
3
4
5
6
7
8
9
Situation


Overall Early Stopping and Futility Stopping
Early Stop and Overall Futility Stop
0.6
0.4
0.2
Stage
1, Early stop
1, Overall futility stop
2, Early stop
2, Overall futility stop
3, Overall futility stop
1
2
3
4
5
6
7
8
9
Situation



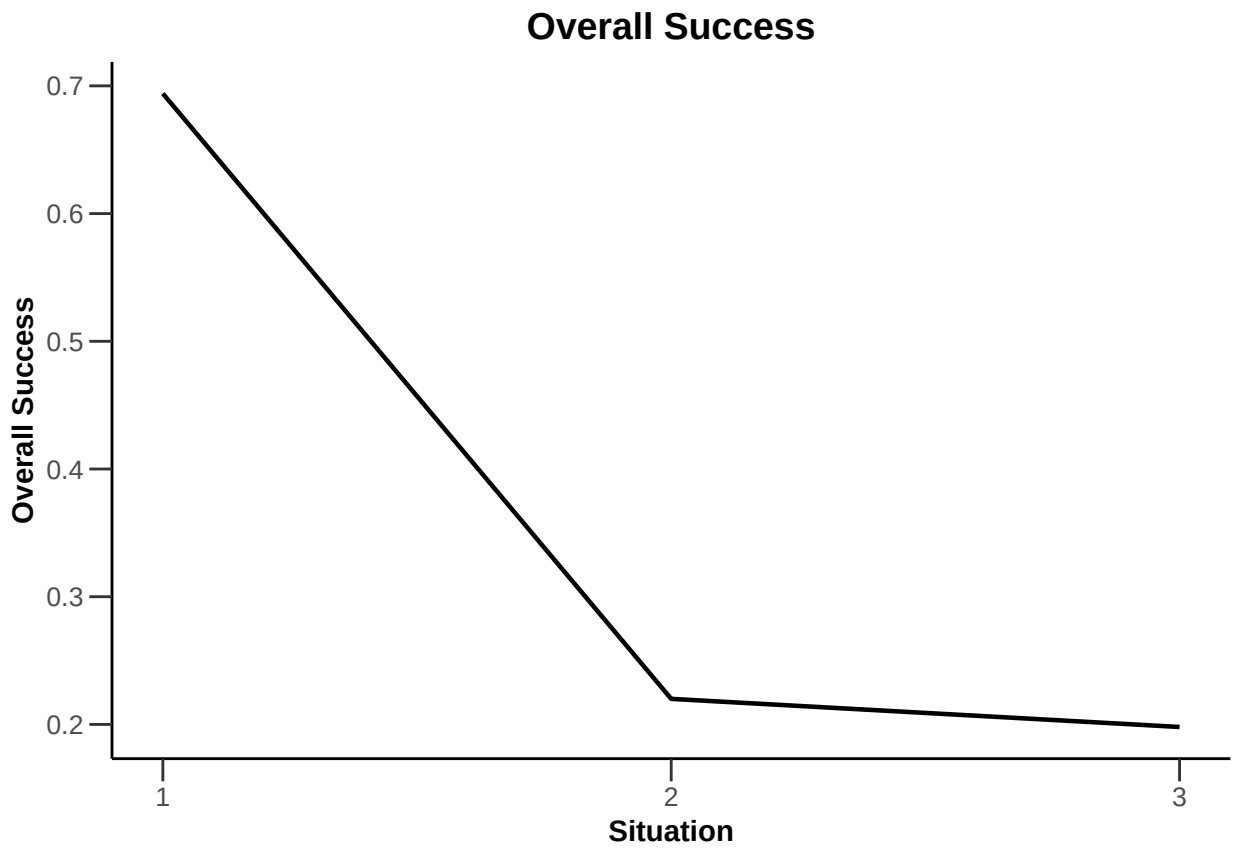
4.3 Simulation results enrichment - survival

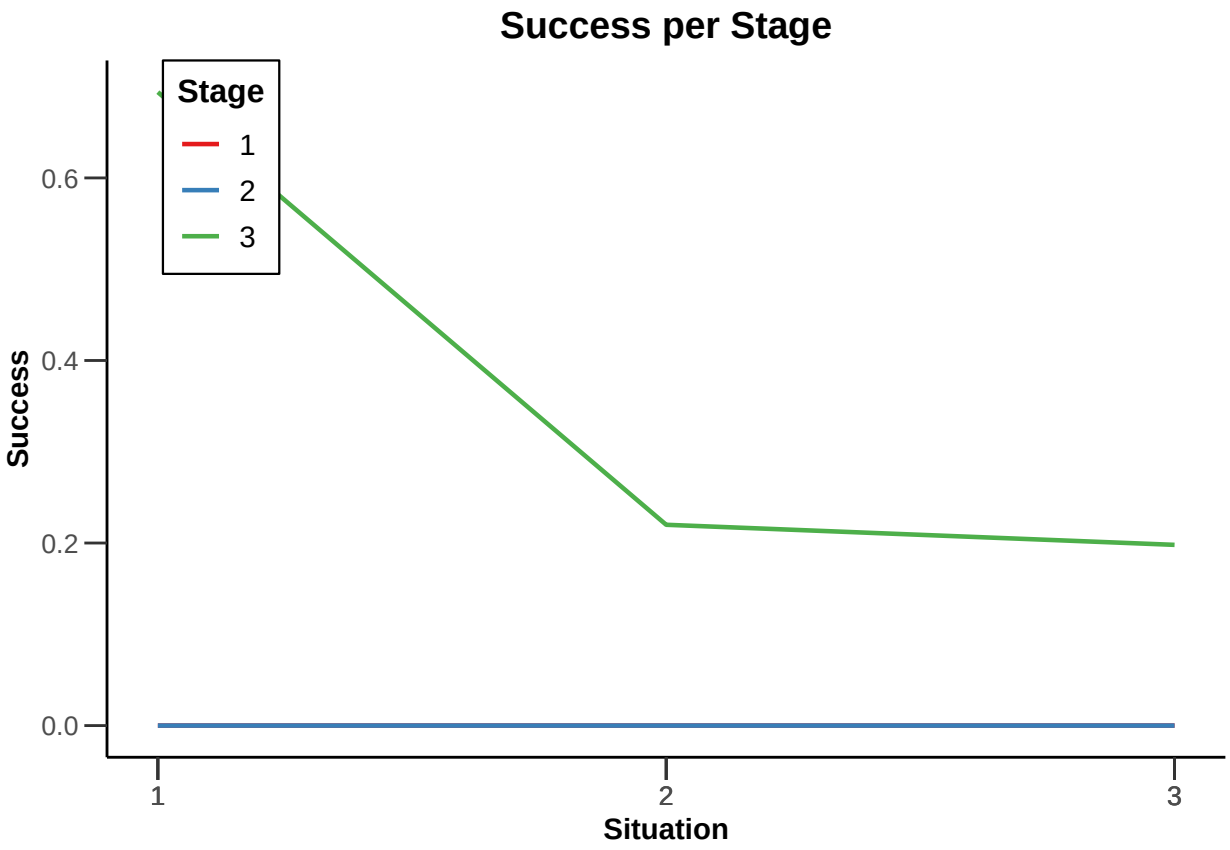
```
# Define subgroups and their prevalences
subGroups <- c("S1", "S2", "S12", "R") # fixed names!
prevalences <- c(0.2, 0.3, 0.4, 0.1)

piControls <- c(0.2, 0.4, 0.15, 0.3)
effect <- c(-0.05, -0.02, -0.10, -0.10)
piTreatments <- piControls + effect

hr <- log(1 - piTreatments) / log(1 - piControls)

# Define effect list
el <- list(
    subGroups = subGroups, prevalences = prevalences,
    piControls = piControls, hazardRatios = matrix(rep(hr, 3), nrow = 3)
)

# Perform simulation
x <- getSimulationEnrichmentSurvival(
    design = getDesignInverseNormal(typeOfDesign = "noEarlyEfficacy"),
    effectList = el,
    typeOfSelection = "rbest",
    rValue = 2,
    intersectionTest = "Simes",
```

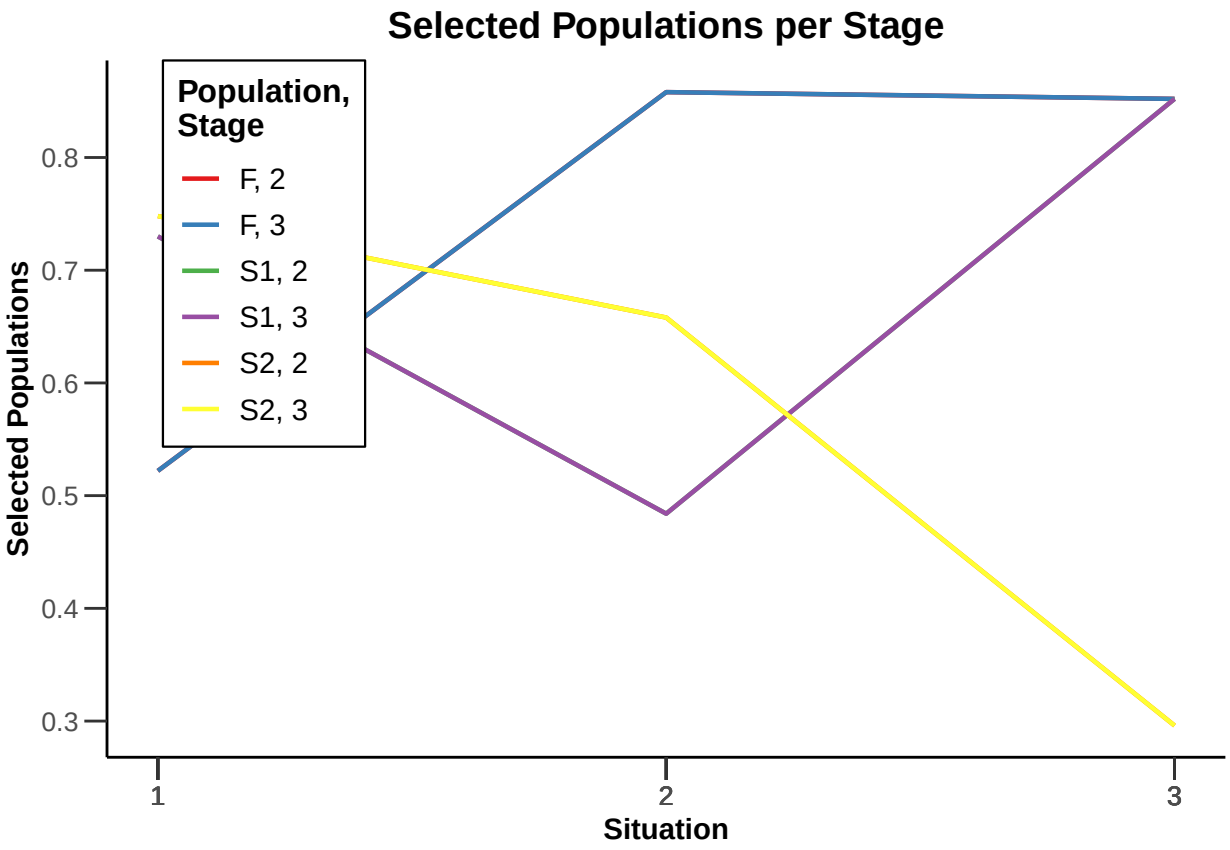
```
    plannedEvents = c(30, 80, 120),
    maxNumberOfIterations = 500,
    directionUpper = FALSE
)
```

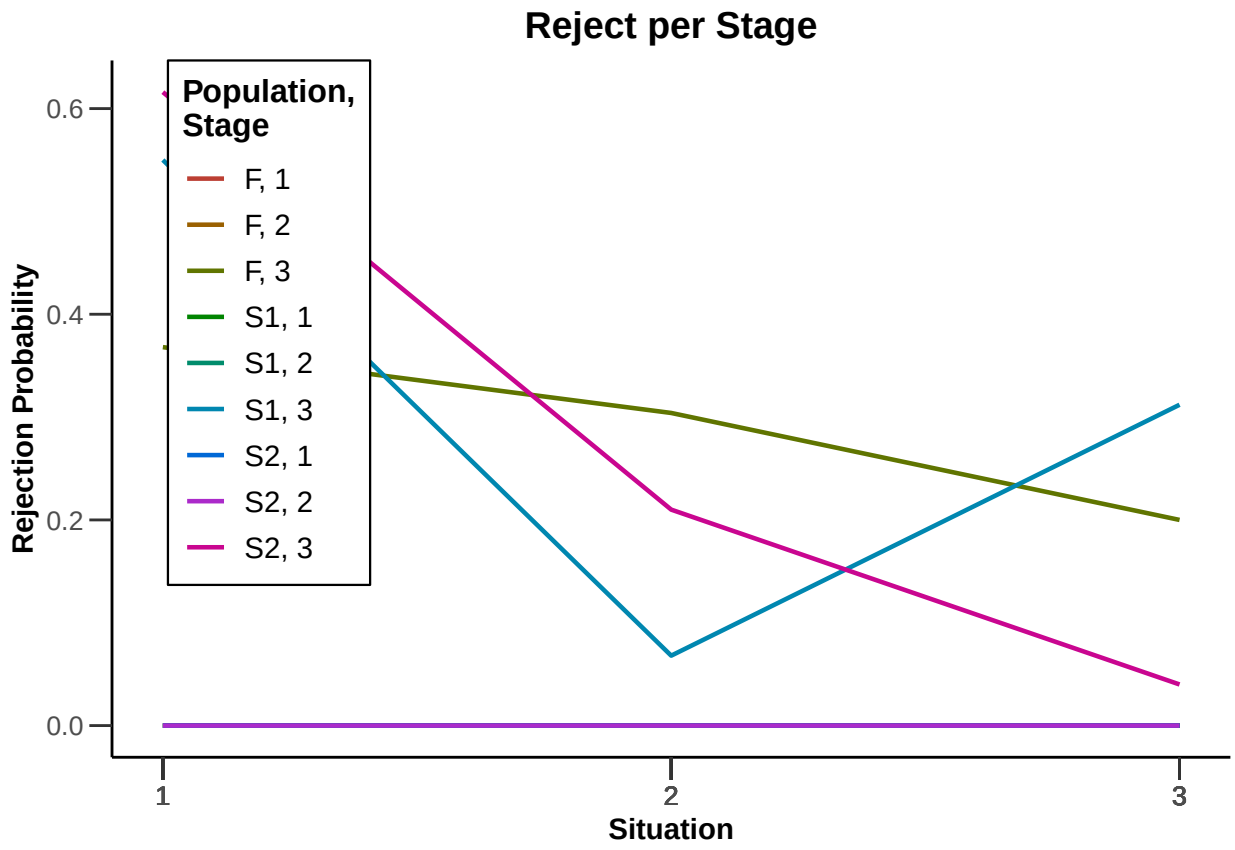
```
## Warning: Simulation of enrichment designs is experimental and hence not fully
## validated (see www.rpact.com/experimental)
```

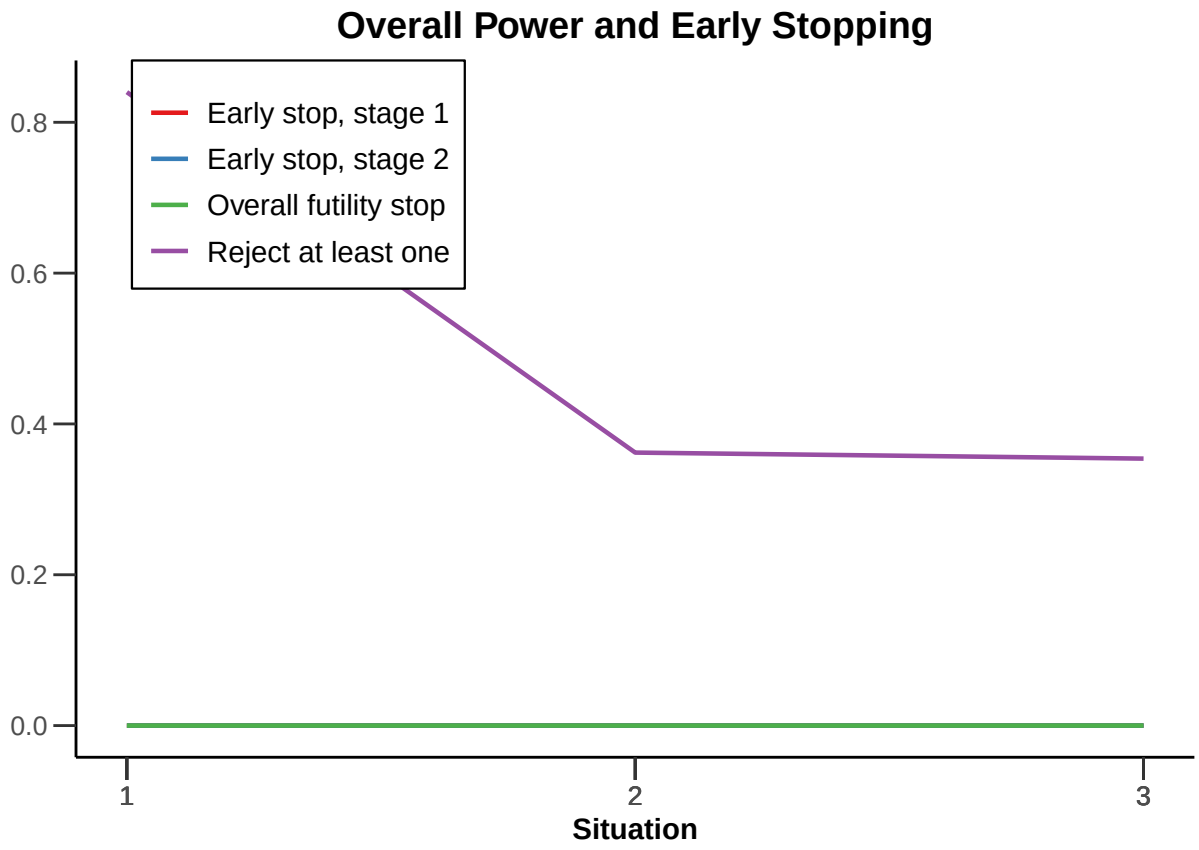
```
plot(x, type = "all", grid = 0)
```

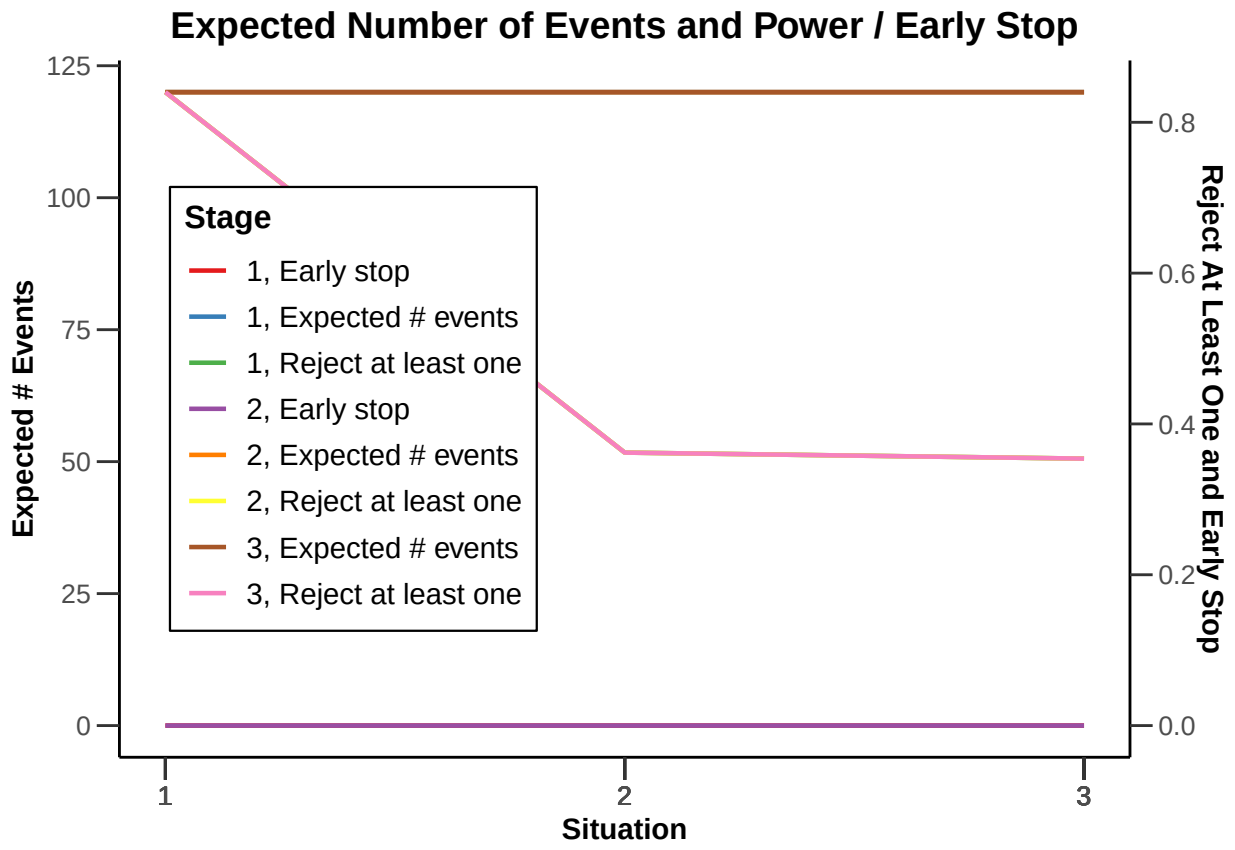


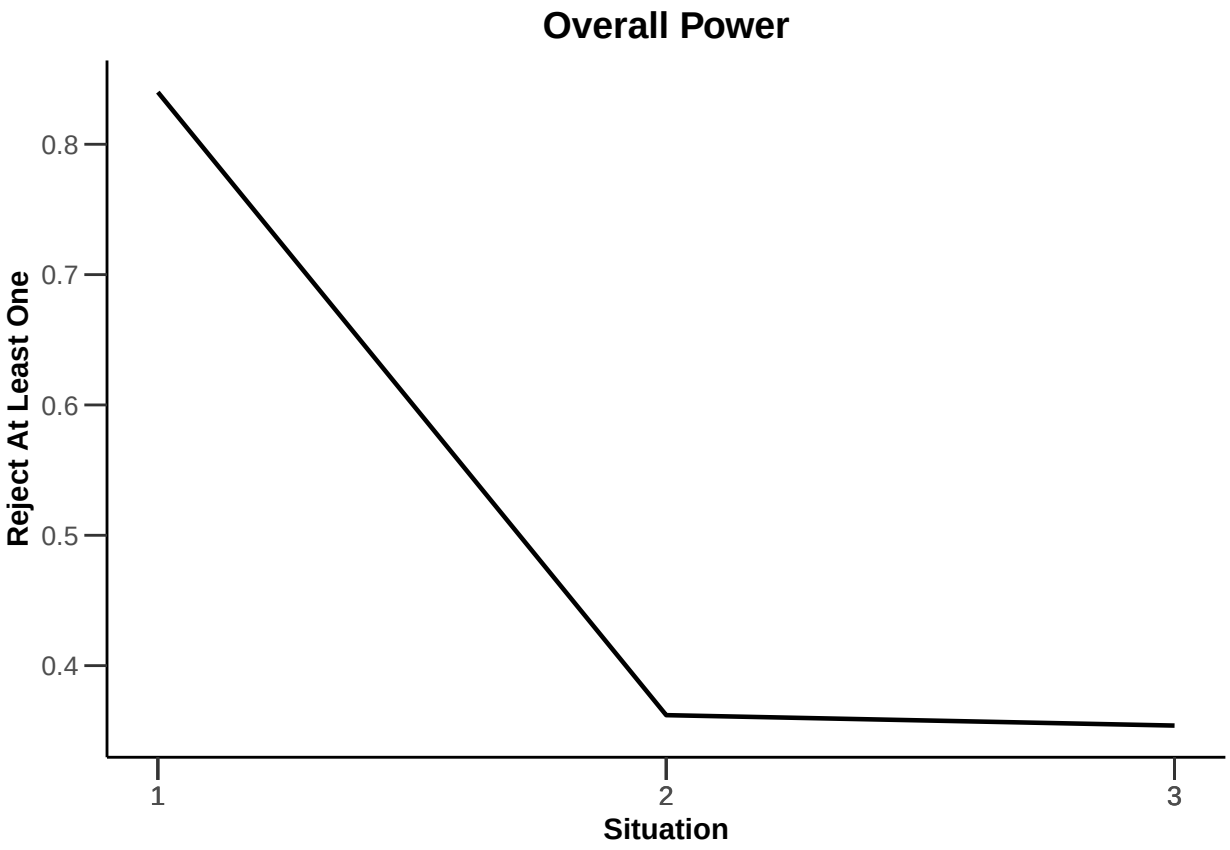

Success per Stage
Stage
1
2
3
Success
0.6
0.4
0.2
0.0
1
2
3
Situation

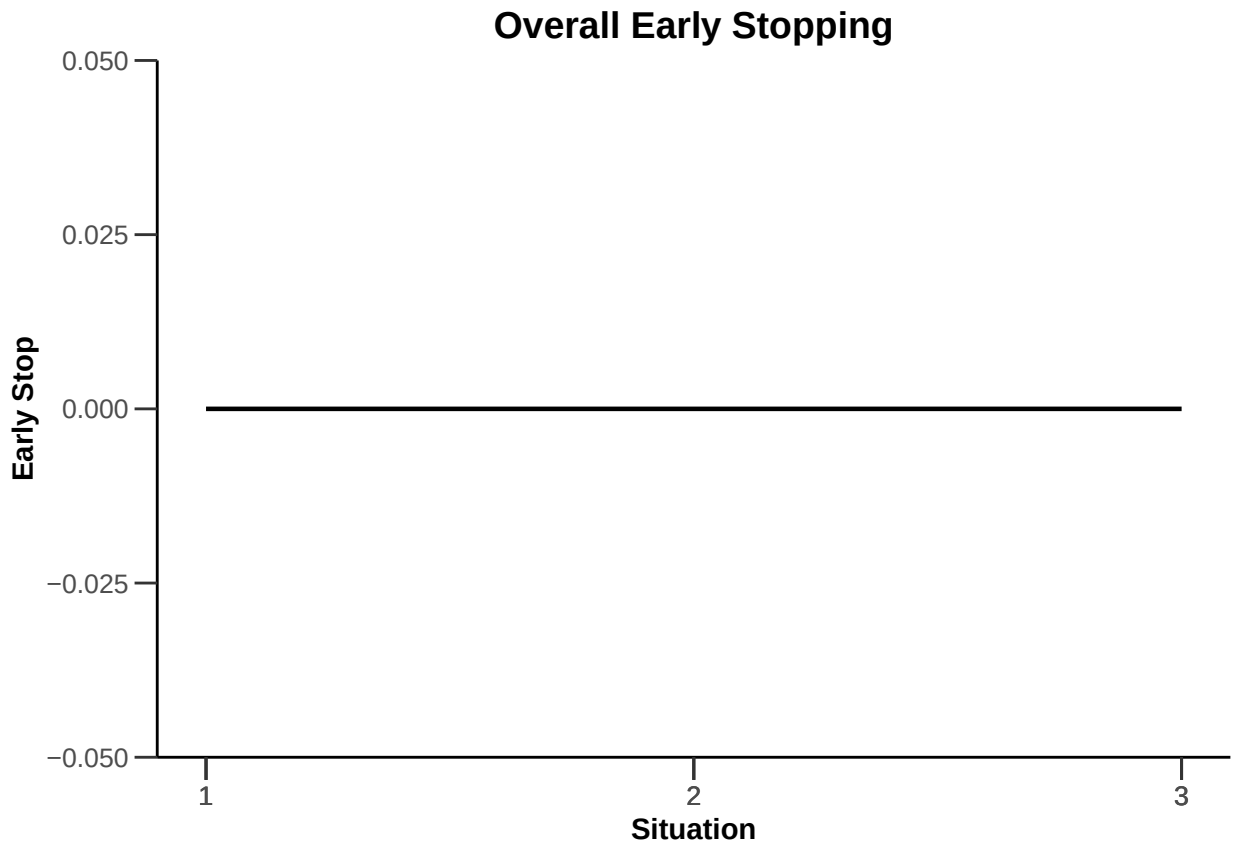

Selected Populations per Stage
Population, Stage
F, 2
F, 3
S1, 2
S1, 3
S2, 2
S2, 3
Selected Populations
0.8
0.7
0.6
0.5
0.4
0.3
1
2
3
Situation

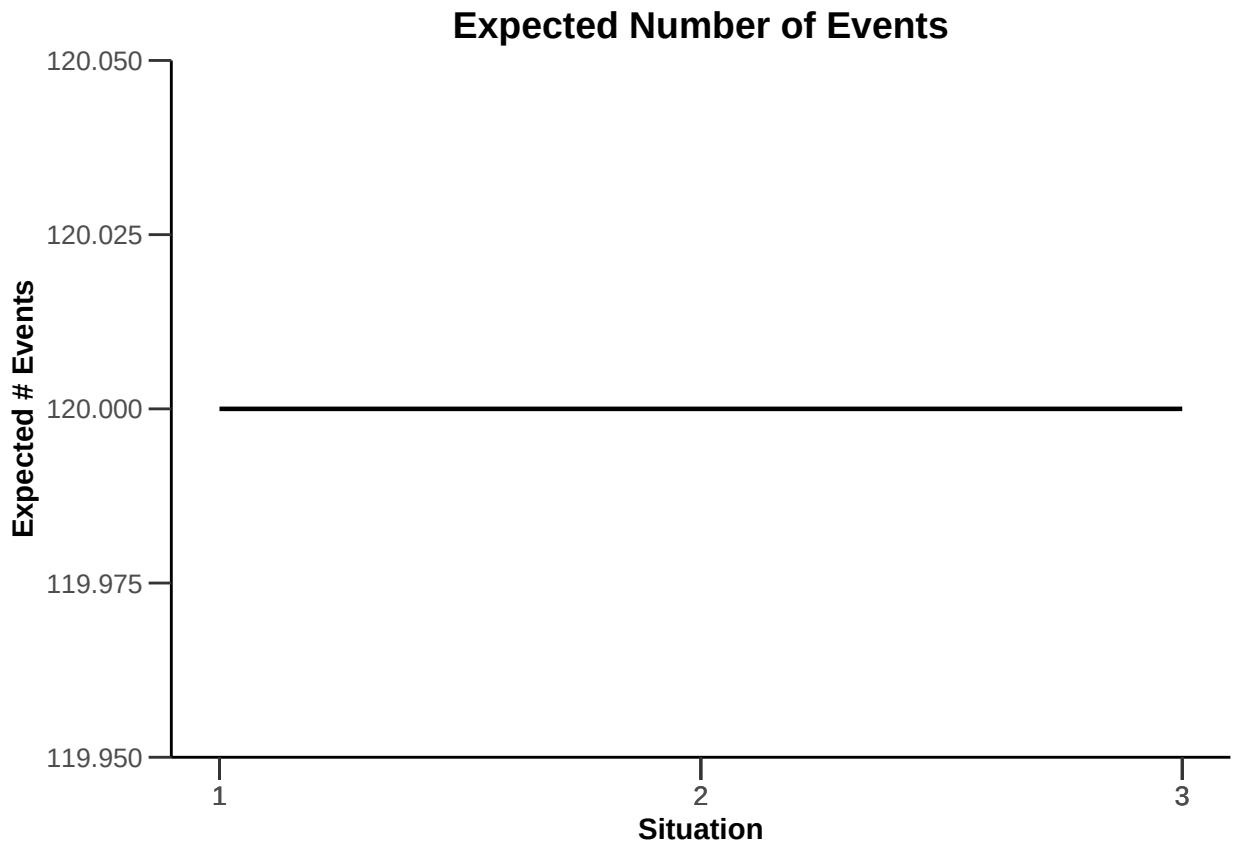

Reject per Stage
Population, Stage
F, 1
F, 2
F, 3
S1, 1
S1, 2
S1, 3
S2, 1
S2, 2
S2, 3
Rejection Probability
0.6
0.4
0.2
0.0
1
2
3
Situation


Overall Power and Early Stopping
Early stop, stage 1
Early stop, stage 2
Overall futility stop
Reject at least one
0.8
0.6
0.4
0.2
0.0
1
2
3
Situation


Expected Number of Events and Power / Early Stop
Expected # Events
125
100
75
50
25
0
Reject At Least One and Early Stop
0.8
0.6
0.4
0.2
0.0
Stage
1, Early stop
1, Expected # events
1, Reject at least one
2, Early stop
2, Expected # events
2, Reject at least one
3, Expected # events
3, Reject at least one
1
2
3
Situation


Overall Power
Reject At Least One
0.8
0.7
0.6
0.5
0.4
1
2
3
Situation


Overall Early Stopping
Early Stop
0.050
0.025
0.000
−0.025
−0.050
1
2
3
Situation



System: rpact 3.3.2, R version 4.2.1 (2022-06-23 ucrt), platform: x86_64-w64-mingw32

To cite R in publications use:

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Um Paket 'rpact' in Publikationen zu zitieren, nutzen Sie bitte:

Wassmer G, Pahlke F (2022). *rpact: Confirmatory Adaptive Clinical Trial Design and Analysis*. https://www.rpact.org, https://www.rpact.com, https://github.com/rpact-com/rpact.

file