```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x purrr::%||%()   masks base::%||%()
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(knitr)
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```r
library(patchwork)
```

```
Attaching package: 'patchwork'

The following object is masked from 'package:MASS':

    area
```

```r
library(broom)
library(rpact)
```

```
Attaching package: 'rpact'
```

```
The following object is masked from 'package:knitr':

    kable

The following object is masked from 'package:dplyr':

    pull
```

```r
library(effsize)
library(pwr)
```

```
::: {.cell}

```{.r .cell-code}
data <- read.csv("data/cookiecats.csv") # import dataset or subset
data$retention_1 <- as.logical(data$retention_1)
data$retention_7 <- as.logical(data$retention_7)
data$retention_1 <- as.numeric(data$retention_1)
data$retention_7 <- as.numeric(data$retention_7)
# calculate the sum users and users per group
nall <- nrow(data)
groupnumber <- table(data$version)
groupnumber
```

```
gate_30 gate_40
  44700   45489
```

```r
# the average game rounds per group and whole
agrall <- mean(data$sum_gamerounds)
agr <- tapply(data$sum_gamerounds, data$version, mean)
# the retention rate per group and whole
# day1
ret.1 <- mean(data$retention_1)
ret34.1 <- tapply(data$retention_1,data$version, mean)
Count.1 <- table(data$retention_1, data$version)
# day7
ret.7 <- mean(data$retention_7)
ret34.7 <- tapply(data$retention_7,data$version, mean)
Count.7 <- table(data$retention_7, data$version)
# SRM checking
```
```

```r
n30 <- 44700
n40 <- 45489
total_number <- n40 + n30
observed <- c(n30, n40)
expected <- c(total_number/2, total_number/2)
srm_test <- chisq.test(observed, p = expected / sum(expected))
print(srm_test)
```

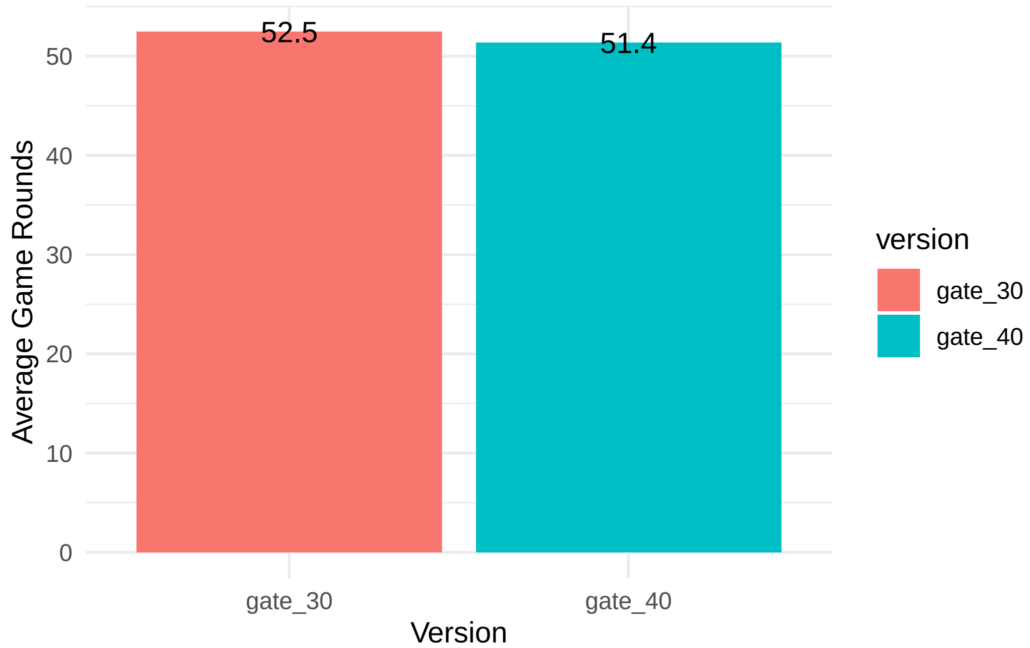
```
    Chi-squared test for given probabilities

data:  observed
X-squared = 6.9024, df = 1, p-value = 0.008608
```

:::

```r
data <- read.csv("data/cookiecats_subset.csv")
data$retention_1 <- as.logical(data$retention_1)
data$retention_7 <- as.logical(data$retention_7)
data$retention_1 <- as.numeric(data$retention_1)
data$retention_7 <- as.numeric(data$retention_7)
ret.1 <- mean(data$retention_1)
total_number <- n40 + n30
ret.7 <- mean(data$retention_7)
agrall <- mean(data$sum_gamerounds)
```

```r
# plot pie and bar
grouped_data <- group_by(data, version)
average_gamerounds <- summarise(grouped_data, mean_gamerounds =
                                  mean(sum_gamerounds))
ggplot(average_gamerounds, aes(x = version, y = mean_gamerounds,
                               fill = version)) +
  geom_bar(stat = "identity") +
  labs(x = "Version",
       y = "Average Game Rounds") +
  geom_text(aes(label = round(mean_gamerounds, 1)) ) +
  theme_minimal()
```



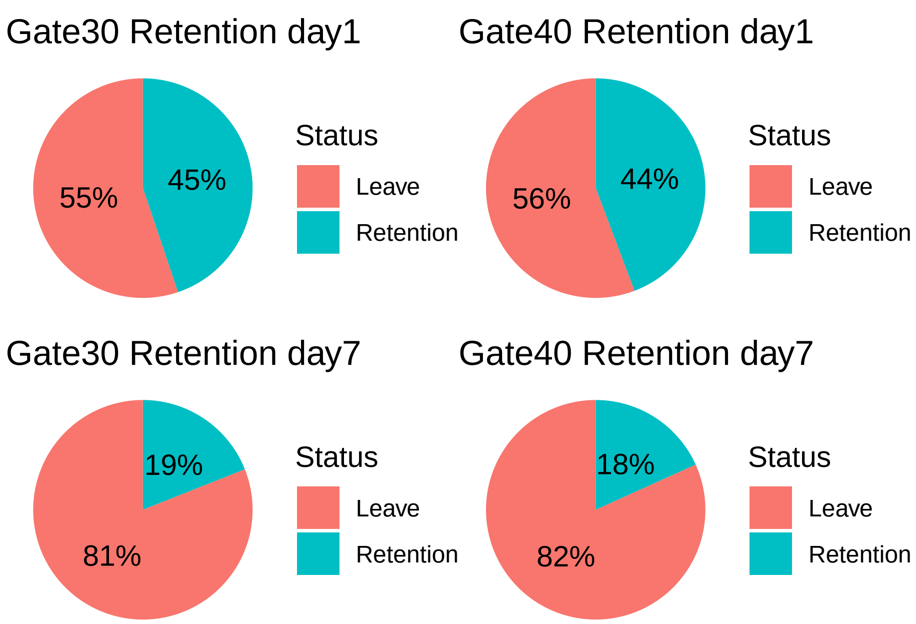

```
nall <- nrow(data)
n30 <- 44000
n40 <- 44000
gate_30_data <- data[data$version == "gate_30", ] # version gate_30
gate_40_data <- data[data$version == "gate_40", ] # version gate_40
avgr30 <- mean(gate_30_data$sum_gamerounds)
avgr40 <- mean(gate_40_data$sum_gamerounds)
reten_1_30 <- mean(gate_30_data$retention_1)
reten_1_40 <- mean(gate_40_data$retention_1)
reten_1_all <- mean(data$retention_1)
reten_7_30 <- mean(gate_30_data$retention_7)
reten_7_40 <- mean(gate_40_data$retention_7)
reten_7_all <- mean(data$retention_7)
leave_1_30 <- 1 - reten_1_30
leave_1_40 <- 1 - reten_1_40
leave_1_all <- 1 - reten_1_all
leave_7_30 <- 1 - reten_7_30
leave_7_40 <- 1 - reten_7_40
leave_7_all <- 1 - reten_7_all

pie <- function(reten_value, leave_value, title) {
  df <- data.frame(
    category = c("Retention", "Leave"),
```

```
    value = c(reten_value, leave_value)
  )

  ggplot(df, aes(x = "", y = value, fill = category)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar("y", start = 0) +
    geom_text(aes(label = scales::percent(value/sum(value))),
    position = position_stack(vjust = 0.5))+
    labs(title = title, fill = "Status") +
    theme_void()
}
p1 <- pie(reten_1_30, leave_1_30, "Gate30 Retention day1")
p2 <- pie(reten_1_40, leave_1_40, "Gate40 Retention day1")
p3 <- pie(reten_7_30, leave_7_30, "Gate30 Retention day7")
p4 <- pie(reten_7_40, leave_7_40, "Gate40 Retention day7")

(p1 | p2) / (p3 | p4)
```




```
library(pwr)
library(effsize)
# using equation 5.11
var30 <- var(gate_30_data$sum_gamerounds)
```

```r
var40 <- var(gate_40_data$sum_gamerounds)
se30gr <-sd(gate_30_data$sum_gamerounds)
se40gr <- sd(gate_40_data$sum_gamerounds)
std_pooled_gr <- sqrt((((n30 - 1)*var30)+ ((n40 - 1)*var40))
                      /(n30 + n40 - 2))
d.gr <- abs(avgr30 - avgr40)
r <- 1
n_gr <- ((1 + r)*((qnorm(0.975) + qnorm(0.8))^2) * (std_pooled_gr^2))/
  (1 *(d.gr^2))
#using package by cohend's d
cohend <- cohen.d(gate_30_data$sum_gamerounds,gate_40_data$sum_gamerounds,
                  pooled = TRUE)
 cohen.d <- cohend$estimate
n_gr1 <- pwr.t.test(d = cohen.d, sig.level = 0.05, power = 0.8,
                   type = "two.sample",
                    alternative = "two.sided")
n.gr <- n_gr1$n

power_value<- pwr.t.test(d = cohen.d, n = 44000, sig.level = 0.05,
                         type = "two.sample", alternative = "two.sided")

# using equation 5.12
# For day1 retention rate
std_pooled_1 <- sqrt((reten_1_30*(1 - reten_1_30)) +
                       (reten_1_40*(1 - reten_1_40)*r))
# std_pooled_1^2 is the variance day1
d.ren1 <- abs(reten_1_30 - reten_1_40)
N_ren1 <- ((qnorm(0.975) + (qnorm(0.8)))^2 * (std_pooled_1^2))/
  (d.ren1^2)
## std_pooled_7^2 is the variance day7
std_pooled_7 <- sqrt((reten_7_30*(1 - reten_7_30)) +
                       (reten_7_40*(1 - reten_7_40)*r))
d.ren7 <- abs(reten_7_30 - reten_7_40)
N_ren7 <- ((qnorm(0.975) + (qnorm(0.8)))^2 * (std_pooled_7^2))/
  (d.ren7^2)
# using cohen'h
cohen.h1 <- ES.h(reten_1_30,reten_1_40)
N.reten1 <- pwr.2p.test(h = cohen.h1, sig.level = 0.05, power = 0.8,
                      alternative = "two.sided")
N.ren1 <- N.reten1$n
cohen.h7 <- ES.h(reten_7_30,reten_7_40)
N.reten7 <- pwr.2p.test(h = cohen.h7, sig.level = 0.05, power = 0.8,
```

```r
                     alternative = "two.sided")
N.ren7 <- N.reten7$n
power_value1 <- pwr.2p.test(h = cohen.h1, n = 44000, sig.level = 0.05,
                         alternative = "two.sided")
```

```r
#build welch T statistic
grgate30 <- gate_30_data$sum_gamerounds
grgate40 <- gate_40_data$sum_gamerounds
welch.t <- t.test(grgate30, grgate40, alternative = "two.sided",
                  var.equal = FALSE, conf.level = 0.95)
# build Z statistic
se.1 <- sqrt(reten_1_all*(1 - reten_1_all)*(1/n30+1/n40))
z.value <- (reten_1_30 - reten_1_40)/se.1
p.value <- 2*(1 - pnorm(abs(z.value)))
print(p.value)
```

```
[1] 0.0912466
```

```r
se.7 <- sqrt(reten_7_all*(1 - reten_7_all)*(1/n30+1/n40))
z.value <- (reten_7_30 - reten_7_40)/se.7
p.value <- 2*(1 - pnorm(abs(z.value)))
print(p.value)
```

```
[1] 0.004891535
```

```r
# buile chi-squre statistic
n30true1 <- sum(gate_30_data$retention_1 ==1)
n40true1 <- sum(gate_40_data$retention_1 ==1)
n30true7 <- sum(gate_30_data$retention_7 ==1)
n40true7 <- sum(gate_40_data$retention_7 ==1)
stay1 <- c(n30true1,n40true1)
stay7 <- c(n30true7,n40true7)
n <- c(n30,n40)
ztest <- prop.test(stay7,n,alternative = "two.sided")# change stay1 or stay7
print(ztest)
```

```

    2-sample test for equality of proportions with continuity correction

data:  stay7 out of n
```

```
X-squared = 7.8705, df = 1, p-value = 0.005025
alternative hypothesis: two.sided
95 percent confidence interval:
 0.002219398 0.012553329
sample estimates:
   prop 1    prop 2
0.1899545 0.1825682
```

```r
# established logistical model
data$version <- factor(data$version, levels = c("gate_40", "gate_30"))
logmodel <- glm(retention_7 ~ version, family = binomial(link = "logit"),
                data = data) # retention_7 or retention_1
summary(logmodel)
```

```
Call:
glm(formula = retention_7 ~ version, family = binomial(link = "logit"),
    data = data)

Coefficients:
               Estimate Std. Error  z value Pr(>|z|)
(Intercept)    -1.49904    0.01234 -121.473  < 2e-16 ***
versiongate_30  0.04874    0.01732    2.814  0.00489 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 84613  on 87999  degrees of freedom
Residual deviance: 84605  on 87998  degrees of freedom
AIC: 84609

Number of Fisher Scoring iterations: 4
```

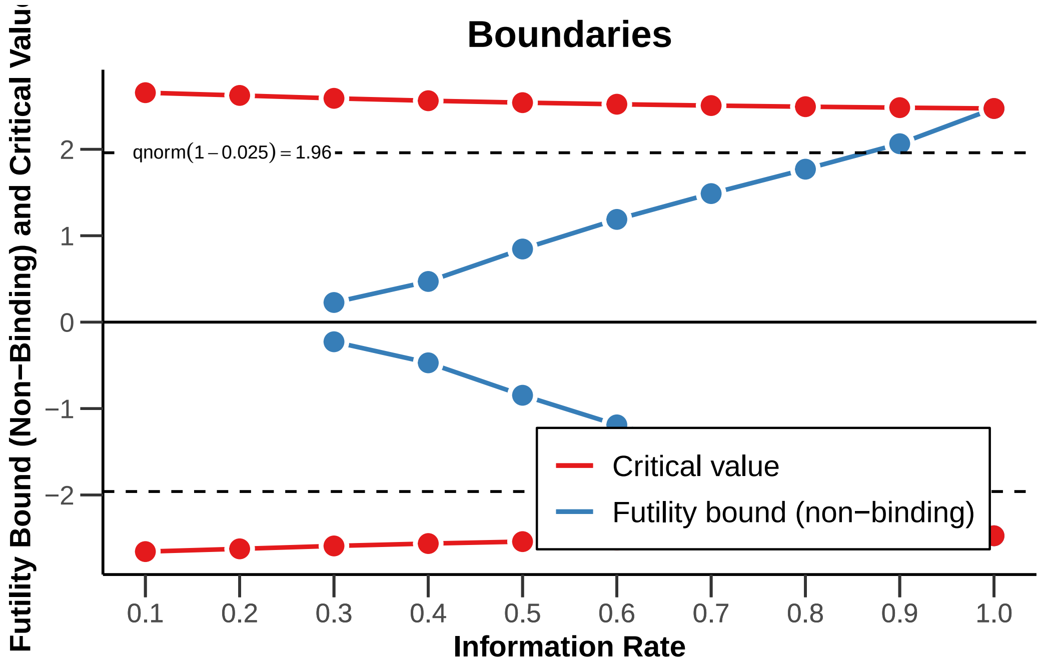
```r
RAhat <- predict(logmodel, type = "response")
mean(RAhat)
```

```
[1] 0.1862614
```

```r
odds <- exp(-1.49904 + 0.04874 * 1)
```

```r
# loading new data with time variable
data1 <- read.csv("data/cookiecats_subset1.csv")
data1$retention_1_t <- as.logical(data1$retention_1_t)
data1$retention_7_t <- as.logical(data1$retention_7_t)
data1$retention_1_t <- as.numeric(data1$retention_1_t)
data1$retention_7_t <- as.numeric(data1$retention_7_t)
data1$retention_1_c <- as.logical(data1$retention_1_c)
data1$retention_7_c <- as.logical(data1$retention_7_c)
data1$retention_1_c <- as.numeric(data1$retention_1_c)
data1$retention_7_c <- as.numeric(data1$retention_7_c)
```

```r
# choosing pocock boundary
library(rpact)
# Example: beta-spending function approach with Pocock spending function
# function and Pocock beta-spending function
designgr <- getDesignGroupSequential(
    sided = 2, alpha = 0.05, beta = 0.2, kMax = 10,
    informationRates = c(0.1, 0.2, 0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),
    typeOfDesign = "asP",
    typeBetaSpending = "bsP",bindingFutility = FALSE,twoSidedPower = FALSE
)
plot(designgr, type = 1)
```




```r
summary(designgr)
```

Sequential analysis with a maximum of 10 looks (group sequential design)

Pocock type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.4339, ASN H1 0.7287, ASN H01 0.7985, ASN H0 0.7059.

Stage	1	2	3	4	5	6	7	8	9	10
Planned information rate	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Efficacy boundary (z-value scale)	2.655	2.623	2.590	2.562	2.540	2.521	2.506	2.493	2.482	2.472

Stage	1	2	3	4	5	6	7	8	9	10
Stage levels (one-sided)	0.0040	0.0044	0.0048	0.0052	0.0055	0.0058	0.0061	0.0063	0.0065	0.0067
Futility boundary (z-value scale)			0.227	0.471	0.846	1.189	1.488	1.770	2.065	
Cumulative alpha spent	0.0079	0.0148	0.0208	0.0262	0.0310	0.0354	0.0395	0.0432	0.0467	0.0500
Cumulative beta spent	0	0	0.0342	0.0646	0.0922	0.1173	0.1403	0.1616	0.1815	0.2000
Cumulative power	0.0554	0.1499	0.2640	0.3821	0.4937	0.5922	0.6737	0.7362	0.7785	0.8000
Futility probabilities under H1	0	0	0.034	0.030	0.028	0.025	0.023	0.021	0.020	

```r
# ASE and maximum sample size
sampleSizeResultgr <- getSampleSizeMeans(designgr, groups = 2,
                        normalApproximation = FALSE, meanRatio = FALSE,
                        thetaH0 = 0, alternative = d.gr,stDev = std_pooled_gr,
                        allocationRatioPlanned = 1 )
summary(sampleSizeResultgr)
```

Sample size calculation for a continuous endpoint

Sequential analysis with a maximum of 10 looks (group sequential design), overall significance level 5% (two-sided). The results were calculated for a two-sample t-test, H0: mu(1) - mu(2) = 0, H1: effect = 1.098, standard deviation = 196.877, power 80%.

Stage	1	2	3	4	5	6	7	8	9	10
Planned information rate	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Efficacy boundary (z-value scale)	2.655	2.623	2.590	2.562	2.540	2.521	2.506	2.493	2.482	2.472
Futility boundary (z-value scale)			0.227	0.471	0.846	1.189	1.488	1.770	2.065	
Cumulative power	0.0554	0.1499	0.2640	0.3821	0.4937	0.5922	0.6737	0.7362	0.7785	0.8000
Number of subjects	144687.6	289375.2	434062.9	578750.5	723438.1	868125.7	1012813.3	1157501.0	1302188.6	1446876.2
Expected number of subjects under H1	735308.4									
Cumulative alpha spent	0.0079	0.0148	0.0208	0.0262	0.0310	0.0354	0.0395	0.0432	0.0467	0.0500
Cumulative beta spent	0	0	0.0342	0.0646	0.0922	0.1173	0.1403	0.1616	0.1815	0.2000

Stage	1	2	3	4	5	6	7	8	9	10
Two-sided local significance level	0.0079	0.0087	0.0096	0.0104	0.0111	0.0117	0.0122	0.0127	0.0131	0.0134
Lower futility boundary (t)			-0.136	-0.244	-0.392	-0.502	-0.582	-0.648	-0.712	
Upper futility boundary (t)			0.136	0.244	0.392	0.502	0.582	0.648	0.712	
Overall exit probability (under H0)	0.0079	0.0068	0.1855	0.2514	0.2360	0.1487	0.0843	0.0463	0.0236	
Overall exit probability (under H1)	0.0554	0.0945	0.1482	0.1486	0.1391	0.1235	0.1046	0.0838	0.0621	

Stage	1	2	3	4	5	6	7	8	9	10
Exit probability for efficacy (under H0)	0.0079	0.0068	0.0060	0.0054	0.0048	0.0043	0.0038	0.0033	0.0025	
Exit probability for efficacy (under H1)	0.0554	0.0945	0.1141	0.1181	0.1116	0.0985	0.0815	0.0625	0.0423	
Exit probability for futility (under H0)	0	0	0.1795	0.2460	0.2312	0.1443	0.0805	0.0430	0.0211	
Exit probability for futility (under H1)	0	0	0.0341	0.0305	0.0275	0.0251	0.0230	0.0213	0.0198	

Legend:

- *(t)*: treatment effect scale

```r
designChargr <- getDesignCharacteristics(designgr)
summary(designChargr)
```

Sequential analysis with a maximum of 10 looks (group sequential design)

Pocock type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.4339, ASN H1 0.7287, ASN H01 0.7985, ASN H0 0.7059.

Stage	1	2	3	4	5	6	7	8	9	10
Planned information rate	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Efficacy boundary (z-value scale)	2.655	2.623	2.590	2.562	2.540	2.521	2.506	2.493	2.482	2.472
Stage levels (one-sided)	0.0040	0.0044	0.0048	0.0052	0.0055	0.0058	0.0061	0.0063	0.0065	0.0067
Futility boundary (z-value scale)			0.227	0.471	0.846	1.189	1.488	1.770	2.065	
Cumulative alpha spent	0.0079	0.0148	0.0208	0.0262	0.0310	0.0354	0.0395	0.0432	0.0467	0.0500
Cumulative beta spent	0	0	0.0342	0.0646	0.0922	0.1173	0.1403	0.1616	0.1815	0.2000
Cumulative power	0.0554	0.1499	0.2640	0.3821	0.4937	0.5922	0.6737	0.7362	0.7785	0.8000

Stage	1	2	3	4	5	6	7	8	9	10
Futility probabilities under H1	0	0	0.034	0.030	0.028	0.025	0.023	0.021	0.020	

```
sampleSizeResultgr[["earlyStop"]]
```

```
[1] 0.959857
```

```
sampleSizeResultgr[["numberOfSubjects"]]
```

```
           [,1]
 [1,]  144687.6
 [2,]  289375.2
 [3,]  434062.9
 [4,]  578750.5
 [5,]  723438.1
 [6,]  868125.7
 [7,] 1012813.3
 [8,] 1157501.0
 [9,] 1302188.6
[10,] 1446876.2
```

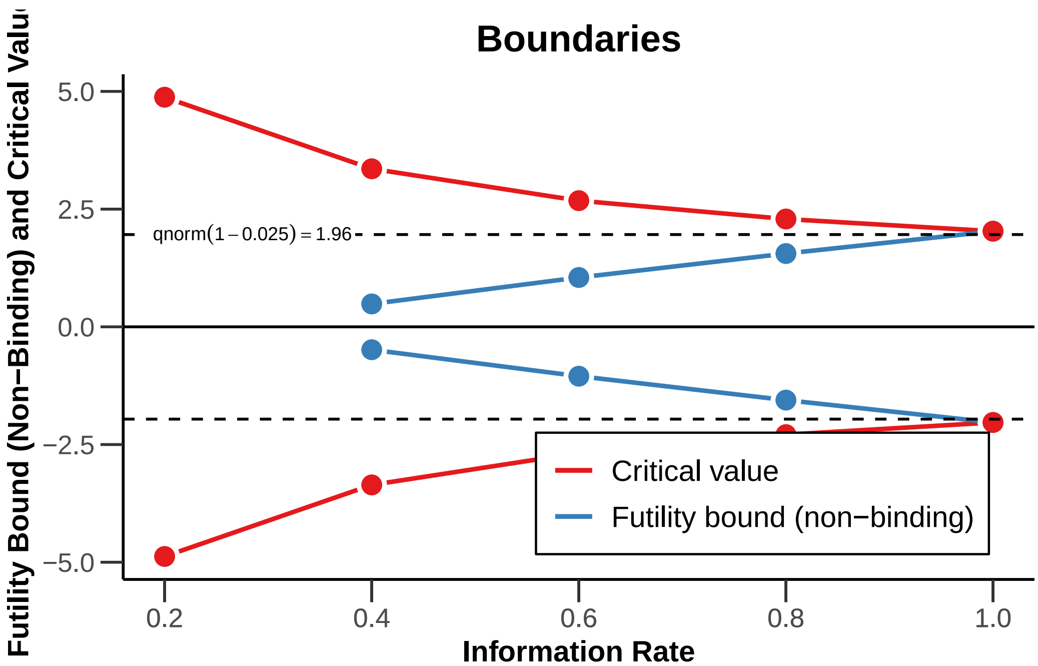
```
sampleSizeResultgr[["maxNumberOfSubjects"]]
```

```
[1] 1446876
```

```
ASNgr <- sampleSizeResultgr[["expectedNumberOfSubjectsH01"]]
fixed_ngr <- 2*n_gr
saverate.gr <-  abs((ASNgr - fixed_ngr))/fixed_ngr
(1009039-712317)/1009039
```

```
[1] 0.294064
```

```
# the total sample size for RA1
fixed_nr1 <- 2*N_ren1
# beta-spending function Pocock; O'Brien & Fleming alpha-spending function
# k = 5 keeping
design1 <- getDesignGroupSequential( kMax = 5,
    sided = 2, alpha = 0.05, beta = 0.2,
    informationRates = c(0.2, 0.4, 0.6,0.8,1),
    typeOfDesign = "asOF",
    typeBetaSpending = "bsP",bindingFutility = FALSE,twoSidedPower = FALSE
)
plot(design1, 1)
```




```
summary(design1)
```

Sequential analysis with a maximum of 5 looks (group sequential design)

O'Brien & Fleming type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.2066, ASN H1 0.8486, ASN H01 0.8209, ASN H0 0.7126.

Stage	1	2	3	4	5
Planned information rate	20%	40%	60%	80%	100%
Efficacy boundary (z-value scale)	4.877	3.357	2.680	2.290	2.031
Stage levels (one-sided)	<0.0001	0.0004	0.0037	0.0110	0.0211
Futility boundary (z-value scale)		0.486	1.048	1.556	
Cumulative alpha spent	<0.0001	0.0008	0.0076	0.0244	0.0500
Cumulative beta spent	0	0.0646	0.1173	0.1616	0.2000
Cumulative power	0.0002	0.0792	0.3855	0.6749	0.8000
Futility probabilities under H1	0	0.065	0.053	0.044	

```
# (pi2 control) vs (pi1 treatment) in intervention
sampleSizeResultGS1 <- getSampleSizeRates(design1, pi1 = reten_1_30,
                                          pi2 = reten_1_40,
                                     allocationRatioPlanned = 1,
                                     groups = 2,normalApproximation = TRUE)
# Standard rpact output
summary(sampleSizeResultGS1)
```

Sample size calculation for a binary endpoint

Sequential analysis with a maximum of 5 looks (group sequential design), overall significance level 5% (two-sided). The results were calculated for a two-sample test for rates (normal approximation), H0: pi(1) - pi(2) = 0, H1: treatment rate pi(1) = 0.448, control rate pi(2) = 0.443, power 80%.

Stage	1	2	3	4	5
Planned information rate	20%	40%	60%	80%	100%
Efficacy boundary (z-value scale)	4.877	3.357	2.680	2.290	2.031
Futility boundary (z-value scale)		0.486	1.048	1.556	
Cumulative power	0.0002	0.0792	0.3855	0.6749	0.8000
Number of subjects	58435.3	116870.6	175306.0	233741.3	292176.6
Expected number of subjects under H1	205498.2				
Cumulative alpha spent	<0.0001	0.0008	0.0076	0.0244	0.0500
Cumulative beta spent	0	0.0646	0.1173	0.1616	0.2000
Two-sided local significance level	<0.0001	0.0008	0.0074	0.0220	0.0423
Lower futility boundary (t)		-0.001	-0.002	-0.003	
Upper futility boundary (t)		0.001	0.002	0.003	
Overall exit probability (under H0)	<0.0001	0.3740	0.3730	0.1789	
Overall exit probability (under H1)	0.0002	0.1436	0.3589	0.3338	

Stage	1	2	3	4	5
Exit probability for efficacy (under H0)	<0.0001	0.0008	0.0068	0.0160	
Exit probability for efficacy (under H1)	0.0002	0.0789	0.3063	0.2894	
Exit probability for futility (under H0)	0	0.3732	0.3661	0.1629	
Exit probability for futility (under H1)	0	0.0646	0.0526	0.0444	

Legend:

- *(t)*: treatment effect scale

```r
designCharr1 <- getDesignCharacteristics(design1)
summary(designCharr1)
```

Sequential analysis with a maximum of 5 looks (group sequential design)

O'Brien & Fleming type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.2066, ASN H1 0.8486, ASN H01 0.8209, ASN H0 0.7126.

Stage	1	2	3	4	5
Planned information rate	20%	40%	60%	80%	100%
Efficacy boundary (z-value scale)	4.877	3.357	2.680	2.290	2.031
Stage levels (one-sided)	<0.0001	0.0004	0.0037	0.0110	0.0211

Stage	1	2	3	4	5
Futility boundary (z-value scale)		0.486	1.048	1.556	
Cumulative alpha spent	<0.0001	0.0008	0.0076	0.0244	0.0500
Cumulative beta spent	0	0.0646	0.1173	0.1616	0.2000
Cumulative power	0.0002	0.0792	0.3855	0.6749	0.8000
Futility probabilities under H1	0	0.065	0.053	0.044	

```r
sampleSizeResultGS1[["numberOfSubjects"]]
```

```
          [,1]
[1,]  58435.32
[2,] 116870.65
[3,] 175305.97
[4,] 233741.29
[5,] 292176.62
```

```r
sampleSizeResultGS1[["earlyStop"]]
```

```
[1] 0.8365378
```

```r
sampleSizeResultGS1[["maxNumberOfSubjects"]]
```

```
[1] 292176.6
```

```r
sampleSizeResultGS1[["expectedNumberOfSubjectsH1"]]
```

```
[1] 205498.2
```

```r
sampleSizeResultGS1[["expectedNumberOfSubjectsH01"]]
```

```
[1] 198792.3
```

```r
sampleSizeResultGS1[["expectedNumberOfSubjectsH0"]]
```

```
[1] 172571.6
```

```r
ASNgr <- sampleSizeResultGS1[["expectedNumberOfSubjectsH01"]]
#stage 1 for retention rate diff testing on day1
data_stage1d1 <- data1[1:29218, ]
meand1stage1 <-apply(data_stage1d1[,c("retention_1_t","retention_1_c")],2,mean)
rate1.t <- meand1stage1[1]
rate1.c <- meand1stage1[2]
staystage1_t <- sum(data_stage1d1$retention_1_t)
staystage1_c <- sum(data_stage1d1$retention_1_c)
sizestage1 <- nrow(data_stage1d1)
d1ra.stage1 <- (staystage1_c+staystage1_t)/(2*sizestage1)
se.11 <- sqrt(d1ra.stage1*(1 - d1ra.stage1)*(1/sizestage1+1/sizestage1))
z.value.11 <- (rate1.t - rate1.c)/se.11
unname(z.value.11)
```

```
[1] 1.290337
```

```r
#p.value.11 <- 2*(1 - pnorm(abs(z.value.11)))
#print(p.value.11)
#saving rates and inflation rate
(242153-172572)/242153
```

```
[1] 0.2873431
```

```r
(292177-242153)/242153
```

```
[1] 0.2065801
```

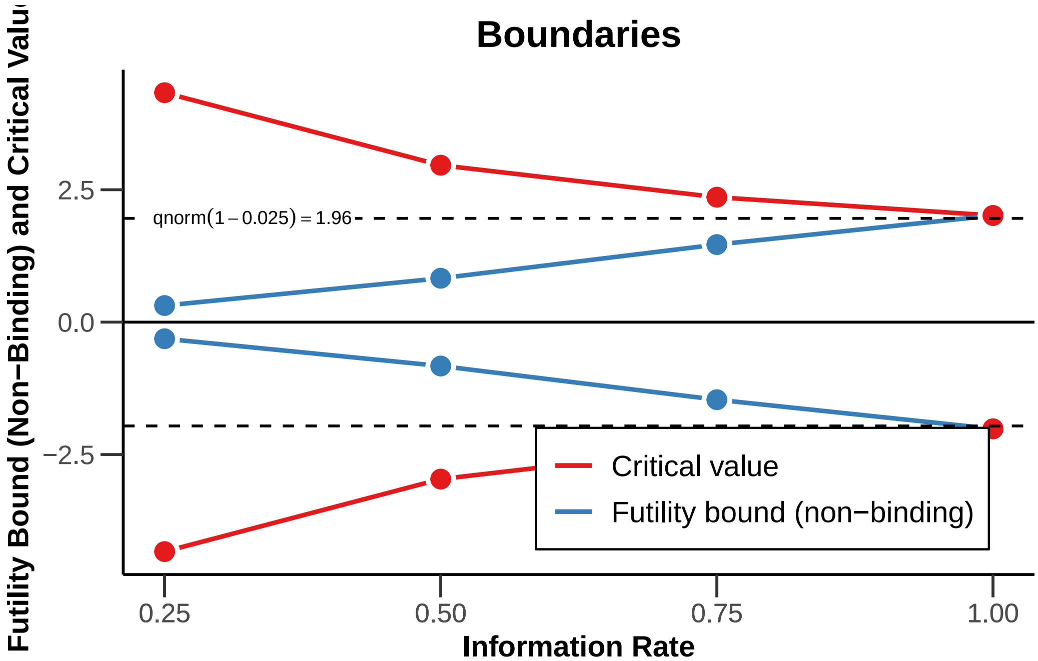
```r
# the total sample size for RA
fixed_nr7 <- 2*N_ren7
# beta-spending function Pocock; O'Brien & Fleming alpha-spending function
# k = 4
design7 <- getDesignGroupSequential( kMax = 4,
    sided = 2, alpha = 0.05, beta = 0.2,
    informationRates = c(0.25, 0.5, 0.75,1),
    typeOfDesign = "asOF",
    typeBetaSpending = "bsP",bindingFutility = FALSE,twoSidedPower = FALSE
)
summary(design7)
```

Sequential analysis with a maximum of 4 looks (group sequential design)

O'Brien & Fleming type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.3012, ASN H1 0.8963, ASN H01 0.8438, ASN H0 0.7126.

Stage	1	2	3	4
Planned information rate	25%	50%	75%	100%
Efficacy boundary (z-value scale)	4.333	2.963	2.359	2.014
Stage levels (one-sided)	<0.0001	0.0015	0.0092	0.0220
Futility boundary (z-value scale)	0.313	0.828	1.465	
Cumulative alpha spent	<0.0001	0.0031	0.0193	0.0500
Cumulative beta spent	0.0715	0.1240	0.1656	0.2000
Cumulative power	0.0031	0.2405	0.6398	0.8000
Futility probabilities under H1	0.071	0.053	0.042	

```r
plot(design7, type =1 )
```




```r
# (pi2 control) vs (pi1 treatment) in intervention
sampleSizeResultGS7 <- getSampleSizeRates(design7, pi1 = reten_7_30,
                                  pi2 = reten_7_40,
                             allocationRatioPlanned = 1,
                             groups = 2,normalApproximation = TRUE)
# Standard rpact output
summary(sampleSizeResultGS7)
```

Sample size calculation for a binary endpoint

Sequential analysis with a maximum of 4 looks (group sequential design), overall significance level 5% (two-sided). The results were calculated for a two-sample test for rates (normal approximation), H0: pi(1) - pi(2) = 0, H1: treatment rate pi(1) = 0.19, control rate pi(2) = 0.183, power 80%.

Stage	1	2	3	4
Planned information rate	25%	50%	75%	100%
Efficacy boundary (z-value scale)	4.333	2.963	2.359	2.014

Stage	1	2	3	4
Futility boundary (z-value scale)	0.313	0.828	1.465	
Cumulative power	0.0031	0.2405	0.6398	0.8000
Number of subjects	28371.7	56743.4	85115.1	113486.8
Expected number of subjects under H1	78176.8			
Cumulative alpha spent	<0.0001	0.0031	0.0193	0.0500
Cumulative beta spent	0.0715	0.1240	0.1656	0.2000
Two-sided local significance level	<0.0001	0.0030	0.0183	0.0440
Lower futility boundary (t)	-0.001	-0.003	-0.004	
Upper futility boundary (t)	0.001	0.003	0.004	
Overall exit probability (under H0)	0.2460	0.4109	0.2499	
Overall exit probability (under H1)	0.0746	0.2900	0.4408	
Exit probability for efficacy (under H0)	<0.0001	0.0030	0.0151	
Exit probability for efficacy (under H1)	0.0031	0.2374	0.3992	
Exit probability for futility (under H0)	0.2459	0.4079	0.2348	
Exit probability for futility (under H1)	0.0715	0.0525	0.0416	

Legend:

- *(t)*: treatment effect scale

```r
designCharr1 <- getDesignCharacteristics(design1)
summary(designCharr1)
```

Sequential analysis with a maximum of 5 looks (group sequential design)

O'Brien & Fleming type alpha spending design and Pocock type beta spending, non-binding futility, two-sided overall significance level 5%, power 80%, undefined endpoint, inflation factor 1.2066, ASN H1 0.8486, ASN H01 0.8209, ASN H0 0.7126.

Stage	1	2	3	4	5
Planned information rate	20%	40%	60%	80%	100%
Efficacy boundary (z-value scale)	4.877	3.357	2.680	2.290	2.031
Stage levels (one-sided)	<0.0001	0.0004	0.0037	0.0110	0.0211
Futility boundary (z-value scale)		0.486	1.048	1.556	
Cumulative alpha spent	<0.0001	0.0008	0.0076	0.0244	0.0500
Cumulative beta spent	0	0.0646	0.1173	0.1616	0.2000
Cumulative power	0.0002	0.0792	0.3855	0.6749	0.8000
Futility probabilities under H1	0	0.065	0.053	0.044	

```r
sampleSizeResultGS7[["numberOfSubjects"]]
```

```
         [,1]
[1,] 28371.69
```

```
[2,]  56743.39
[3,]  85115.08
[4,] 113486.78
```

```r
# stage 1 for RA7
sampleSizeResultGS7[["numberOfSubjects1"]]
```

```
         [,1]
[1,] 14185.85
[2,] 28371.69
[3,] 42557.54
[4,] 56743.39
```

```r
data_stage1d7 <- data1[1:14186, ]
meand7stage1 <-apply(data_stage1d7[,c("retention_7_t","retention_7_c")],2,mean)
rate7_1.t <- meand7stage1[1]
rate7_1.c <- meand7stage1[2]
staystage1_t7 <- sum(data_stage1d7$retention_7_t)
staystage1_c7 <- sum(data_stage1d7$retention_7_c)
sizestage1.7 <- nrow(data_stage1d7)
d7ra.stage1 <- (staystage1_c7+staystage1_t7)/(2*sizestage1.7)
se.71 <- sqrt(d7ra.stage1*(1 - d7ra.stage1)*(1/sizestage1.7+1/sizestage1.7))
z.value.71 <- (rate7_1.t - rate7_1.c)/se.71
unname(z.value.71)
```

```
[1] 1.383307
```

```r
#stage 2 for RA7
data_stage1d7 <- data1[1:28372, ]
meand7stage1 <-apply(data_stage1d7[,c("retention_7_t","retention_7_c")],2,mean)
rate7_1.t <- meand7stage1[1]
rate7_1.c <- meand7stage1[2]
staystage1_t7 <- sum(data_stage1d7$retention_7_t)
staystage1_c7 <- sum(data_stage1d7$retention_7_c)
sizestage1.7 <- nrow(data_stage1d7)
d7ra.stage1 <- (staystage1_c7+staystage1_t7)/(2*sizestage1.7)
se.71 <- sqrt(d7ra.stage1*(1 - d7ra.stage1)*(1/sizestage1.7+1/sizestage1.7))
z.value.71 <- (rate7_1.t - rate7_1.c)/se.71
unname(z.value.71)
```

```
[1] 2.763375
```

```
#stage 3 for RA7
data_stage1d7 <- data1[1:42558, ]
meand7stage1 <-apply(data_stage1d7[,c("retention_7_t","retention_7_c")],2,mean)
rate7_1.t <- meand7stage1[1]
rate7_1.c <- meand7stage1[2]
staystage1_t7 <- sum(data_stage1d7$retention_7_t)
staystage1_c7 <- sum(data_stage1d7$retention_7_c)
sizestage1.7 <- nrow(data_stage1d7)
d7ra.stage1 <- (staystage1_c7+staystage1_t7)/(2*sizestage1.7)
se.71 <- sqrt(d7ra.stage1*(1 - d7ra.stage1)*(1/sizestage1.7+1/sizestage1.7))
z.value.71 <- (rate7_1.t - rate7_1.c)/se.71
unname(z.value.71)
```

```
[1] 2.621098
```

```
(87212-85116)/87212
```

```
[1] 0.02403339
```

```
# finding a eariler stopping
design7a <- getDesignGroupSequential( kMax = 5,
    sided = 2, alpha = 0.05, beta = 0.2,
    informationRates = c(0.2, 0.4,0.6,0.8,1),
    typeOfDesign = "asOF",
    typeBetaSpending = "bsP",bindingFutility = FALSE,twoSidedPower = FALSE
)
# (pi2 control) vs (pi1 treatment) in intervention
sampleSizeResultGS7 <- getSampleSizeRates(design7a, pi1 = reten_7_30,
                                 pi2 = reten_7_40,
                                allocationRatioPlanned = 1,
                                groups = 2,normalApproximation = TRUE)
summary(sampleSizeResultGS7)# 31569.6 is about 31570
```

Sample size calculation for a binary endpoint

Sequential analysis with a maximum of 5 looks (group sequential design), overall significance level 5% (two-sided). The results were calculated for a two-sample test for rates (normal approximation), H0: pi(1) - pi(2) = 0, H1: treatment rate pi(1) = 0.19, control rate pi(2) = 0.183, power 80%.

Stage	1	2	3	4	5
Planned information rate	20%	40%	60%	80%	100%
Efficacy boundary (z-value scale)	4.877	3.357	2.680	2.290	2.031
Futility boundary (z-value scale)		0.486	1.048	1.556	
Cumulative power	0.0002	0.0792	0.3855	0.6749	0.8000
Number of subjects	21046.4	42092.8	63139.2	84185.6	105232.0
Expected number of subjects under H1	74013.4				
Cumulative alpha spent	<0.0001	0.0008	0.0076	0.0244	0.0500
Cumulative beta spent	0	0.0646	0.1173	0.1616	0.2000
Two-sided local significance level	<0.0001	0.0008	0.0074	0.0220	0.0423
Lower futility boundary (t)		-0.002	-0.003	-0.004	
Upper futility boundary (t)		0.002	0.003	0.004	
Overall exit probability (under H0)	<0.0001	0.3740	0.3730	0.1789	
Overall exit probability (under H1)	0.0002	0.1436	0.3589	0.3338	

Stage	1	2	3	4	5
Exit probability for efficacy (under H0)	<0.0001	0.0008	0.0068	0.0160	
Exit probability for efficacy (under H1)	0.0002	0.0789	0.3063	0.2894	
Exit probability for futility (under H0)	0	0.3732	0.3661	0.1629	
Exit probability for futility (under H1)	0	0.0646	0.0526	0.0444	

Legend:

- *(t)*: treatment effect scale

```
data_stage1d7 <- data1[1:31570, ]
meand7stage1 <-apply(data_stage1d7[,c("retention_7_t","retention_7_c")],2,mean)
rate7_1.t <- meand7stage1[1]
rate7_1.c <- meand7stage1[2]
staystage1_t7 <- sum(data_stage1d7$retention_7_t)
staystage1_c7 <- sum(data_stage1d7$retention_7_c)
sizestage1.7 <- nrow(data_stage1d7)
d7ra.stage1 <- (staystage1_c7+staystage1_t7)/(2*sizestage1.7)
se.71 <- sqrt(d7ra.stage1*(1 - d7ra.stage1)*(1/sizestage1.7+1/sizestage1.7))
z.value.71 <- (rate7_1.t - rate7_1.c)/se.71
unname(z.value.71)
```

```
[1] 2.985519
```

```
63140/2
```

```
[1] 31570
```

```r
(87212-63140)/87212
```

```
[1] 0.2760171
```

```r
sampleSizeResultGS7[["expectedNumberOfSubjectsH1"]]# compare ASN under H1
```

```
[1] 74013.43
```

```r
sampleSizeResultGS7[["maxNumberOfSubjects"]]
```

```
[1] 105232
```

```r
(105232-87212)/87212
```

```
[1] 0.2066229
```