CT 31 第一期臨床試驗

Phase One Design

臨床試驗: 設計與分析

Clinical Trials: Design and Alaysis

December 11, 2019

目錄 I

Operating Characteristic Analysis

- Monte Carlo Method
- C: MTD
- $P(C)$: P(dose C is found to be MTD)
- Estimate $P(C)$
- Confirm: $P(C) < 33\%$
- Hope: $P(C)$ is close to 33%
- 3+3 with k dose levels, must specify k probabilities

$$\pi = P(\text{Toxicity} \mid d_i), i = 1, 2, \ldots, k. \tag{1.1}$$

- An OC is conditional on particular truth

Traditional 3+3 Design: Simulation I

- M = 1000 simulations
- c_m = indicator as RP2D in each M

$$\widehat{P}(C) \approx= \sum m = 1^{M} \frac{c_m}{M}. \tag{1.2}$$

- Suppose two dose levels, with $\pi_1 = 0.1$ and $\pi_2 = 0.25$.

Traditional 3+3 Design: Simulation II

```
set.seed(1234);
nsim = 1000;
true_prob = c(0.1, 0.25)
cohort1a = rbinom(nsim, 3, true_prob[1])
cohort1b = rbinom(nsim, 3, true_prob[1])
cohort2a = rbinom(nsim, 3, true_prob[2])
cohort2b = rbinom(nsim, 3, true_prob[2])
```

Traditional 3+3 Design: Simulation III

```
# Recommend dose 1 as MTD
mean((cohort1a == 0 | (cohort1a + cohort1b <= 1))
     & (cohort2a + cohort2b > 1))

## [1] 0.422
```

Traditional 3+3 Design: Simulation IV

```
# Recommend dose 2 as MTD
mean((cohort1a == 0 | (cohort1a + cohort1b <= 1))
     & (cohort2a + cohort2b  <= 1))

## [1] 0.472
```

Question

練習 1

- Replace the random seed with your ID.
- Suppose 3 dose levels, $\pi_1 = 0.1$, $\pi_2 = 0.25$ and $\pi_3 = 0.4$.
- Report the simulation-based OC correspondng to each π.

Biased Coin Design: Simulation I

```
sm_biasedcoin = function(true_probs,
                         start = 1,
                         q_esc = 1,
                         n = 20,
                         nsim = 100,
                         seed=sample(.Machine$integer.max, 1)) {
# Put code in here
}
```

目錄 I

MTPI: Simulation I

- $\pi_1, \pi_2, \ldots, \pi_k$: parameters
- $p(\pi)$: conjugate prior with $Beta(a_1, a_2)$ distribution
- $L(\boldsymbol{y} \mid \pi)$: likelihood for binomial distribution
- $p(\pi \mid \boldsymbol{y})$: posterior with $Beta(a_1 + y, a_2 + (n - y))$
- ε: equivalence interval around target

MTPI: Simulation II

- Beta is a conjugate prior
- Start with $a_1 + a_2$ effective subjects
  - a_1 DLTs
  - a_2 non-DLTs
- Distribution of each π_j remains Beta

  after (real) data are collected (independent across j)
- For each dose level j, need only

  number of DLTs (y_j) and non-DLTs ($n_j - y_j$)
- ε: equivalence interval around target

```r
mtpi_decision = function(y,
                         n,
                         target = 0.30, # DLT rate
                         epsilon = 0.05, # equivalence interval around target
                         prior_shape1 = 0.5, # a1
                         prior_shape2 = 0.5) # a2
{
  post_shape1 = y + prior_shape1
  post_shape2 = n - y + prior_shape2
  E = pbeta(target - epsilon, post_shape1, post_shape2) /
    (target - epsilon)
  S = (pbeta(target + epsilon, post_shape1, post_shape2) -
         pbeta(target - epsilon, post_shape1, post_shape2)) /
    (2 * epsilon)
  D = pbeta(target + epsilon, post_shape1, post_shape2, lower.tail = F) /
    (1 - target - epsilon)
  c(E = E, S = S, D = D)
}
```

```r
mtpi_decision(0, 2)
```

```
##         E         S         D
## 2.9873200 0.9141251 0.2488577
```

```r
mtpi_decision(1, 2)
```

```
##         E         S         D
## 0.7820044 1.1641772 1.0585864
```

```r
mtpi_decision(1, 6)
```

```
##         E         S         D
## 2.6092659 1.7132933 0.2713141
```

```r
mtpi_decision(1, 7)
```

```
##         E         S         D
## 2.8992506 1.5349618 0.1872172
```

```r
mtpi_decision(2, 7)
```

```
##         E         S         D
## 1.5459888 2.3671754 0.5796696
```

```r
mtpi_decision(2, 2)
```

```
##          E          S          D
## 0.04689924 0.16788961 1.49459420
```

- Data are shared within a dose level but not between.
- After $n_1 = 6$ patients at dose level 1, $y_1 = 1$ DLT,

  and MTPI says to escalate to dose level 2 for patient 7
- After DLT / no DLT is observed for patient 7,

  $p((\pi_2 \mid \boldsymbol{y}) = Beta(\pi_2 \mid y_2 + a_1, 1y_2 + a_2)$, where $y_2 = 0$ or $y_2 = 1$
- No knowledge used about 1/6 DLT rate at dose level 1.

Behavior Can be Completely Pre-specified I

```
n = 20; target = 0.30; epsilon = 0.05; shape_both = 0.5;
recommendations = unacceptable =
  matrix(NA, n+1, n, dimnames = list(0:(n), 1:n));
for(i in 1:n) {
  recommendations[1:(i+1), i] =
    c("E","S","D")[apply(matrix(mtpi_decision(0:i, i,
                                              target = target,
                                              epsilon = epsilon,
                                              prior_shape1 = shape_both,
                                              prior_shape2 = shape_both),
                                nrow = i+1), 1, which.max)];
  unacceptable[1:(i+1), i] =
    pbeta(target + epsilon,
          shape_both + (0:i),
          shape_both + i - (0:i),
          lower.tail = F) > 0.95;
}
```

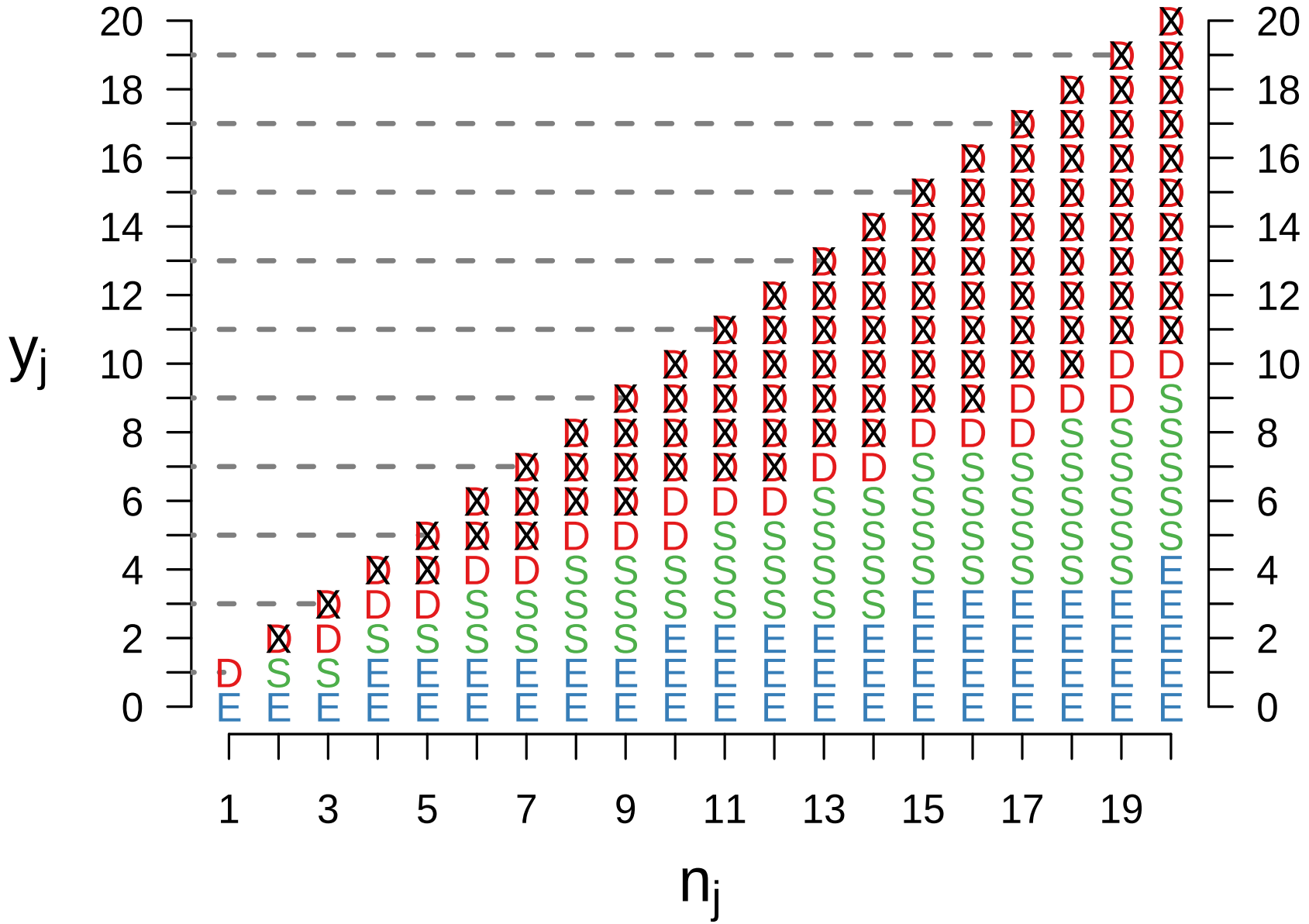
Behavior Can be Completely Pre-specified II

```
plot_code = function() {
  par(mar = c(5, 4, 0, 2), oma = c(0, 0.1, 0, 0.1), las = 1);
  plot.new(); plot.window(xlim = c(1, n), c(0, n));
  axis(1, at = 1:n);
  axis(2, at = 0:n, las = 2); axis(4, at = 0:n, las = 2);
  for(i in 1:n) {
    if(i%%2) segments(0, i, i-0.1, i, lty = 2, lwd = 2, col = "grey50");
    text(recommendations[1:(i+1), i], x = i, y = 0:i,
         col = ifelse(recommendations[1:(i+1), i] == "E", "#377EB8",
                      ifelse(recommendations[1:(i+1),i] == "S", "#4DAF4A",
                             "#E41A1C")));
    text("X", x = i, y = which(unacceptable[1:(i+1), i]) - 1,
         col = "black");
  }
  mtext(expression(n[j]), side = 1, line = 3, cex = 1.5);
  mtext(expression(y[j]), side = 2, line = 3, cex = 1.5);
  }
```

Behavior Can be Completely Pre-specified III

```
plot_code()
```

Behavior Can be Completely Pre-specified IV



Sample from Posterior and Enforce Montonicity I

```
# (Gelfand, Smith and Lee, 1992)

mu_draws = cbind(rbeta(1e6, 3+shape_both, 5+shape_both),
                 rbeta(1e6, 1+shape_both, 3+shape_both),
                 rbeta(1e6, 4+shape_both, 4+shape_both));
satisfied = which((mu_draws[, 1] < mu_draws[, 2]) & (mu_draws[, 2] < mu_draws[, 3
mu_draws_mono = mu_draws[satisfied, ];
nrow(mu_draws_mono) / nrow(mu_draws); # proportion of draws retained

## [1] 0.191398
```

Unconstrained Posterior I

- $p(\pi \mid \boldsymbol{y})$

```
# posterior mean (closer to target is better)
colMeans(mu_draws);

## [1] 0.3888056 0.2997745 0.5000954

# posterior pr(mu_j > p_t) (smaller is better)
colMeans(mu_draws > target);

## [1] 0.691105 0.446549 0.888239

# posterior pr(p_t - eps < mu_j < p_t + eps) (larger is better)
colMeans((mu_draws > target - epsilon) * ((mu_draws < target + epsilon)));

## [1] 0.226422 0.182889 0.126424
```

Constrained Posterior I

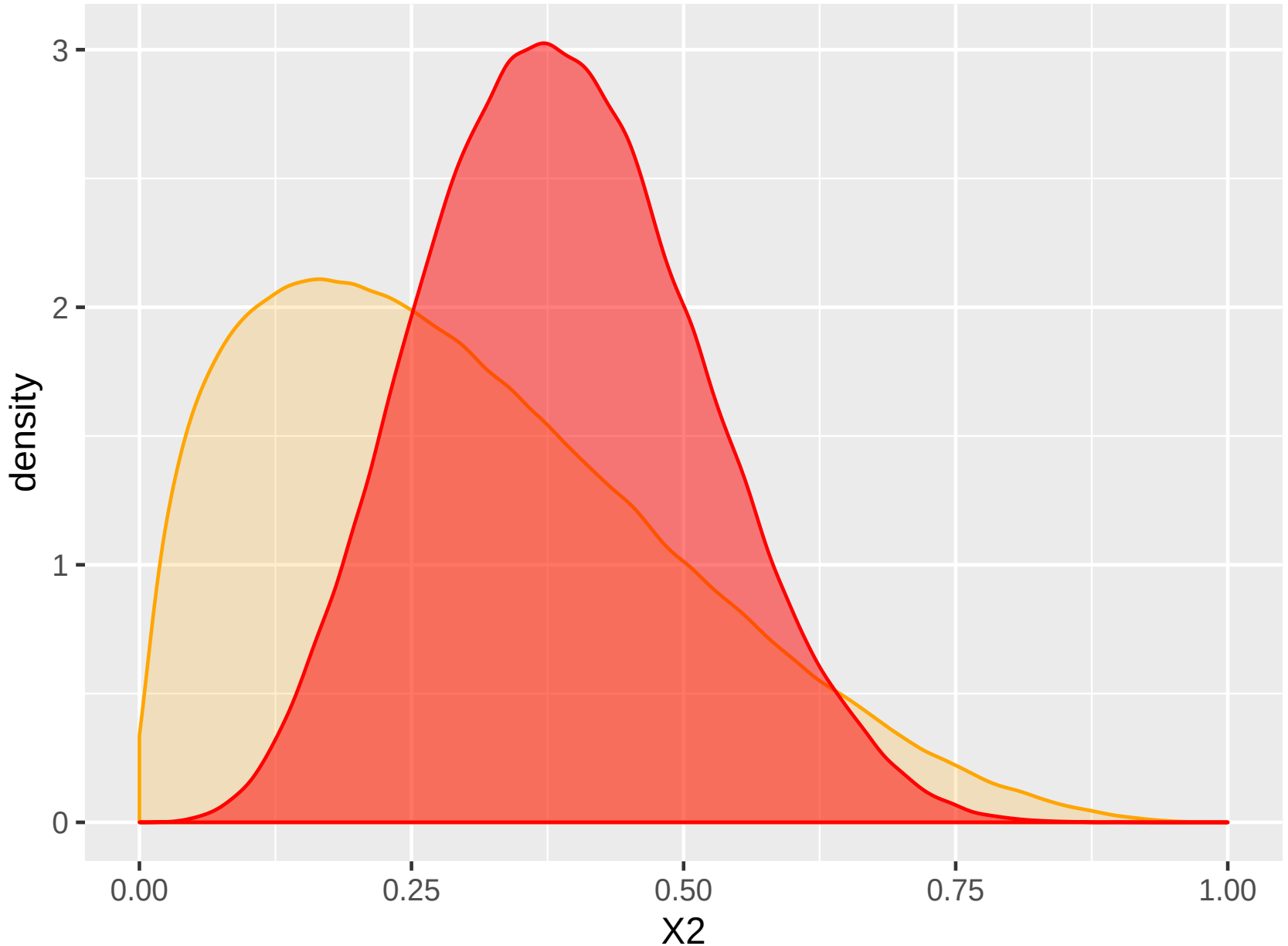
- $p(\pi \mid \boldsymbol{y}, \pi_1 < \pi_2 < \pi_3)$

```
# posterior mean given monotonicity
colMeans(mu_draws_mono);

## [1] 0.2593899 0.3883762 0.5800689

# posterior pr(mu_j > p_t) given monotonicity
colMeans(mu_draws_mono > target);

## [1] 0.3336764 0.7426723 0.9842214

# posterior pr(p_t - eps < mu_j < p_t + eps) given monotonicity
colMeans((mu_draws_mono > target - epsilon) * ((mu_draws_mono < target + epsilon)

## [1] 0.29961651 0.25777176 0.03758137
```

UnConstrained vs Constrained Posterior I

```
library(ggplot2)
ggplot() +
  geom_density(data = data.frame(mu_draws), aes(x = X2),
               fill = "orange", color = "orange", alpha = 0.2) +
  geom_density(data = data.frame(mu_draws_mono), aes(x = X2),
               fill = "red", color = "red", alpha = 0.5) +
  xlim(0, 1)
```

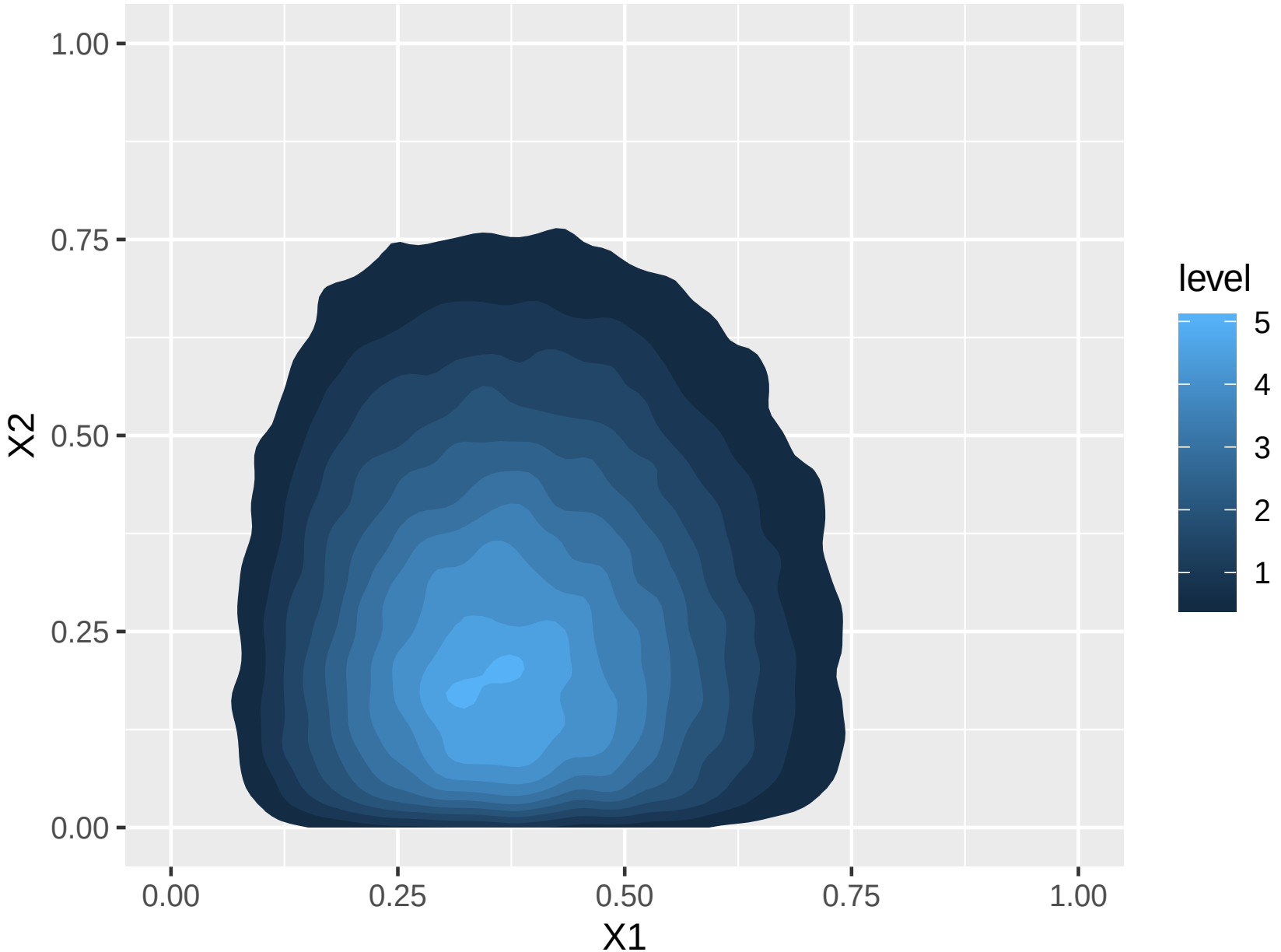
UnConstrained vs Constrained Posterior II



UnConstrained Posterior I

```
# Unconstrained Posterior
library(ggplot2)
ggplot(data = data.frame(mu_draws[1:1e5, ])) +
  lims(x = c(0, 1), y = c(0, 1)) +
  stat_density_2d(aes(x = X1, y = X2, fill = ..level..),
                  geom = ”polygon”)
```

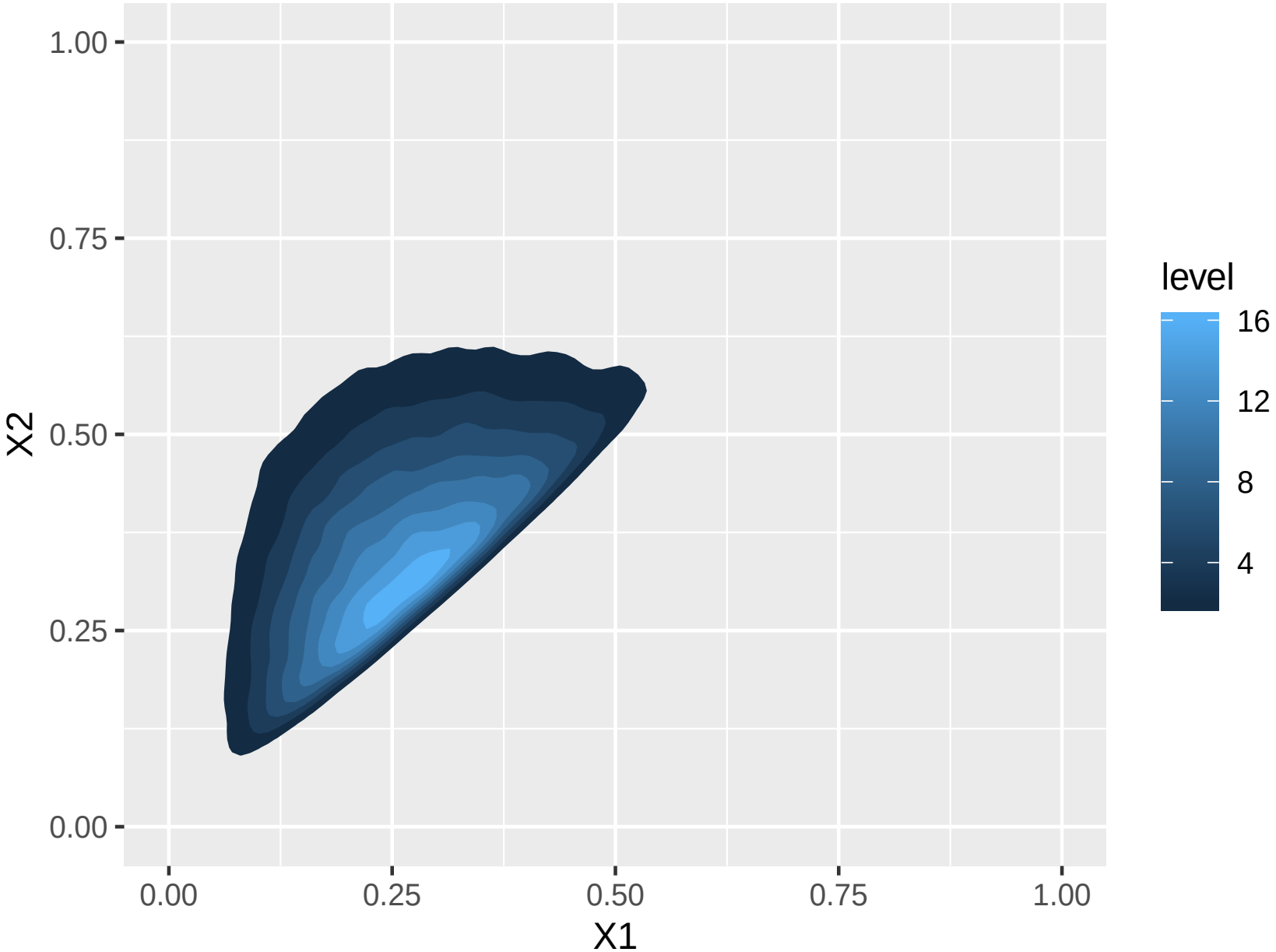
UnConstrained Posterior II



Constrained Posterior I

```
# Constrained Posterior
library(ggplot2)
ggplot(data = data.frame(mu_draws_mono[1:1e5, ])) +
  lims(x = c(0, 1), y = c(0,1)) +
  stat_density_2d(aes(x = X1, y = X2, fill = ..level..),
                  geom = ”polygon”)
```

Constrained Posterior II



目錄 I

```r
antic = c(12,16,22,30,40,52)/100 # skeleton
truth1 = c(2,12,20,23,26,30)/100 # True curve
truth2 = c(12,16,22,30,40,52)/100 + 0.10 # True curve

library(dfcrm)

## Error in library(dfcrm): there is no package called 'dfcrm'

set.seed(1234)
sim1 = titesim(
  PI = truth1, # True curve
  prior = antic, # skeleton
  x0 = 2, # starting dose level
  n = 30, # sample size
  target = 0.30, # target rate of DLT
  nsim = 100, # number of simulated trials
    count = F, # Don't display progress of simulations
    restrict = T, # place restrictions on dose escalation
    scale = 1) # sd of normal prior on beta

## Error in titesim(PI = truth1, prior = antic, x0 = 2, n = 30, target = 0.3,
: 沒有這個函數"titesim"
```

```
sim1
```

```
## # A tibble: 30 x 2
##        x     y
##    <int> <dbl>
##  1     1  4.20
##  2     1  7.51
##  3     1  2.13
##  4     2  8.99
##  5     2 10.2
##  6     2 11.3
##  7     3  7.36
##  8     3 10.5
##  9     3 10.5
## 10     4 12.4
## # ... with 20 more rows
```

目錄 I

TITE-CRM implemented in `dfcrm` package (Cheung, 2013)

- function `titesim()`
- Different choices for *w*
- Simulate arrival process of patients
- Calculate estimated trial duration

```
# 3 subjects for >1 cycle
## each at dose level 3 free of DLT (001-003)
# 1 subject at dose level 4 with DLT (004)
# 1 subject at dose level 4 without DLT (005)
# 1 subject at dose level 4 without DLT for 9/21 days
## (dropped due to progression) (006)
# 1 subject enrolled to dose level 5, never received drug (007)
# 1 sujbect enrolled to dose level 5 without DLT for 7/21 days
## (dropped due to progression) (008)
# 1 subject enrolled to dose level 5 without DLT (009)
```

```r
# Data
tox_dat = c(0,0,0,1,0,0,0,0,0)
level = c(3,3,3,4,4,4,5,5,5)
n = length(tox_dat)
weights = c(1,1,1,1,1,9/21,0,7/21,1)

library(dfcrm)

## Error in library(dfcrm): there is no package called 'dfcrm'

set.seed(1234)
sim2 = titecrm(prior = c(2,5,10,15,20,30,40,45)/100,
               target = 0.30,
               tox = tox_dat,
               level = level,
               weights = weights,
               method = "bayes",
               model = "logistic",
               intcpt = 0,
               scale = sqrt(1.34))

## Error in titecrm(prior = c(2, 5, 10, 15, 20, 30, 40, 45)/100, target =
0.3, : 沒有這個函數"titecrm"
```

目錄 I

Phase II with Single Arm: Normal-based Sample Size

- $H_0 : \pi = \pi_0$ versus $H_A : \pi = \pi_1$.
- Effect size: $\delta = \pi_1 - \pi_0$.
- n: sample size.

$$n = \frac{\left(Z_{1-\alpha} \sqrt{\pi_0 (1 - \pi_0)} + Z_{1-\beta} \sqrt{\pi_A (1 - \pi_1)} \right)^2}{(\pi_1 - \pi_0)^2}. \tag{5.1}$$

```r
# Phase II + Single Arm
simp_sampsize = function(p0, p1,
                         alpha = 0.05, beta = 0.20) {
  ceiling((qnorm(1 - alpha) * sqrt(p0*(1 - p0)) +
             qnorm(1 - beta) * sqrt(p1*(1 - p1)))^2 /
            (p1 - p0)^2)
}

x = matrix(mapply(simp_sampsize,
                  p0 = rep((1:6)/10, each = 6),
                  p1 = rep((2:7)/10, times = 6)),
           nrow=6,
           byrow=T,
           dimnames = list((1:6)/10, (2:7)/10))
```

```
x
```

```
##     0.2 0.3 0.4 0.5 0.6 0.7
## 0.1  69  20  10   6   4   3
## 0.2 Inf 109  29  13   8   5
## 0.3 119 Inf 136  35  16   9
## 0.4  33 142 Inf 151  38  16
## 0.5  15  37 153 Inf 153  37
## 0.6   9  16  38 151 Inf 142
```

- π_0: rows
- π_1: columns
- $\alpha = 0.05$ and $\beta = 0.20$
- $n \propto (\pi_0 - \pi_1)^2$
- Infeasible to test for improvement of 0.1 or less

```r
# po = 0.2, p1 = 0.35
# alpha = 0.05, beta = 0.20
# n = 50, r = 15
# reject when R > 15
# calculate type I error
pbinom(15, 50, 0.20, lower = F) # type I error

## [1] 0.03080342

pbinom(15, 50, 0.35, lower = F) # type I error

## [1] 0.7198956
```

```r
# calculate exact one-sided CI
binom.test(15, 50, p = 0.20,
           alternative = "greater", conf.level = 0.95)

##
##  Exact binomial test
##
## data:  15 and 50
## number of successes = 15, number of trials = 50, p-value = 0.06072
## alternative hypothesis: true probability of success is greater than 0.2
## 95 percent confidence interval:
##  0.1948849 1.0000000
## sample estimates:
## probability of success
##                    0.3
```

```r
# calculate exact two-sided CI
binom.test(15, 50, p = 0.20,
           alternative = "two.sided", conf.level = 0.90)
```

```
##
##  Exact binomial test
##
## data:  15 and 50
## number of successes = 15, number of trials = 50, p-value = 0.1087
## alternative hypothesis: true probability of success is not equal to 0.2
## 90 percent confidence interval:
##  0.1948849 0.4237330
## sample estimates:
## probability of success
##                    0.3
```

```r
# calculate approximate two-sided CI
prop.test(15, 50, p = 0.20,
          alternative = "two.sided", conf.level = 0.90)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  15 out of 50, null probability 0.2
## X-squared = 2.5312, df = 1, p-value = 0.1116
## alternative hypothesis: true p is not equal to 0.2
## 90 percent confidence interval:
##  0.1974083 0.4249925
## sample estimates:
##   p
## 0.3
```

Comparison of One-stage, Modified Gehan Designs

- One stage
- Continue to phase 3 if $R > r$ responses out of n patients

```r
n = 50; r = 14;
p0 = 0.20; p1 = 0.35;

# Type I error = P (R > r | \pi = p0)
pbinom(q = r, size = n, prob = p0, lower.tail = F)

## [1] 0.06072208

# Power = P (R > r | \pi = p1)
pbinom(q = r, size = n, prob = p1, lower.tail = F)

## [1] 0.812223
```

```r
# Modified Gehan designs
n1 = 7; n = 50; r = 14
p0 = 0.20; p1 = 0.35

(summand1 = pbinom(r - (1:n1), n - n1, p0, lower.tail = F));

## [1] 0.03622801 0.07332134 0.13554241 0.22887402 0.35331616 0.49971868
## [7] 0.65030413

(summand2 = dbinom(1:n1, n1, p0));

## [1] 0.3670016 0.2752512 0.1146880 0.0286720 0.0043008 0.0003584 0.0000128

# Type I error
sum(summand1 * summand2);

## [1] 0.05729186

# Power
sum(pbinom(r - (1:n1), n - n1, p1, lower.tail = F) *
      dbinom(1:n1, n1, p1))

## [1] 0.7846002
```

目錄 I

Simon's two-Stage Design

- Implemented in `clinfun` package (Seshan, 2015)

```r
library(clinfun);
ph2simon(pu = 0.20, pa = 0.35, ep1 = 0.05, ep2 = 0.20);

##
##  Simon 2-stage Phase II design
##
## Unacceptable response rate:  0.2
## Desirable response rate:  0.35
## Error rates: alpha =  0.05 ; beta =  0.2
##
##         r1 n1  r  n EN(p0) PET(p0)
## Optimal  5 22 19 72  35.37  0.7326
## Minimax  6 31 15 53  40.44  0.5711
```

- Compared to one-stage design ($n = 50$; $r = 14$), there is cost in terms of numbers of subjects.

Implemented in `ph2mult` package (Zhu and Qin, 2016)

```r
library(ph2mult)

## Error in library(ph2mult): there is no package called 'ph2mult'

par(mar = c(4, 4, 3, 1))
binom.design(type = "admissible", p0 = 0.20, p1 = 0.35,
             signif.level = 0.05, power.level = 0.8, plot.out = T)

## Error in binom.design(type = "admissible", p0 = 0.2, p1 = 0.35,
signif.level = 0.05, : 沒有這個函數"binom.design"
```

Inference depends on design

```r
# Inference depends on design: Ex 1
r1 = 6; n1 = 31; R = 15; n = 53;
p0 = 0.20
sum(pbinom(R - ((r1+1):n1), n - n1, p0, lower.tail = F) *
                dbinom((r1+1):n1, n1, p0))

## [1] 0.04979161
```

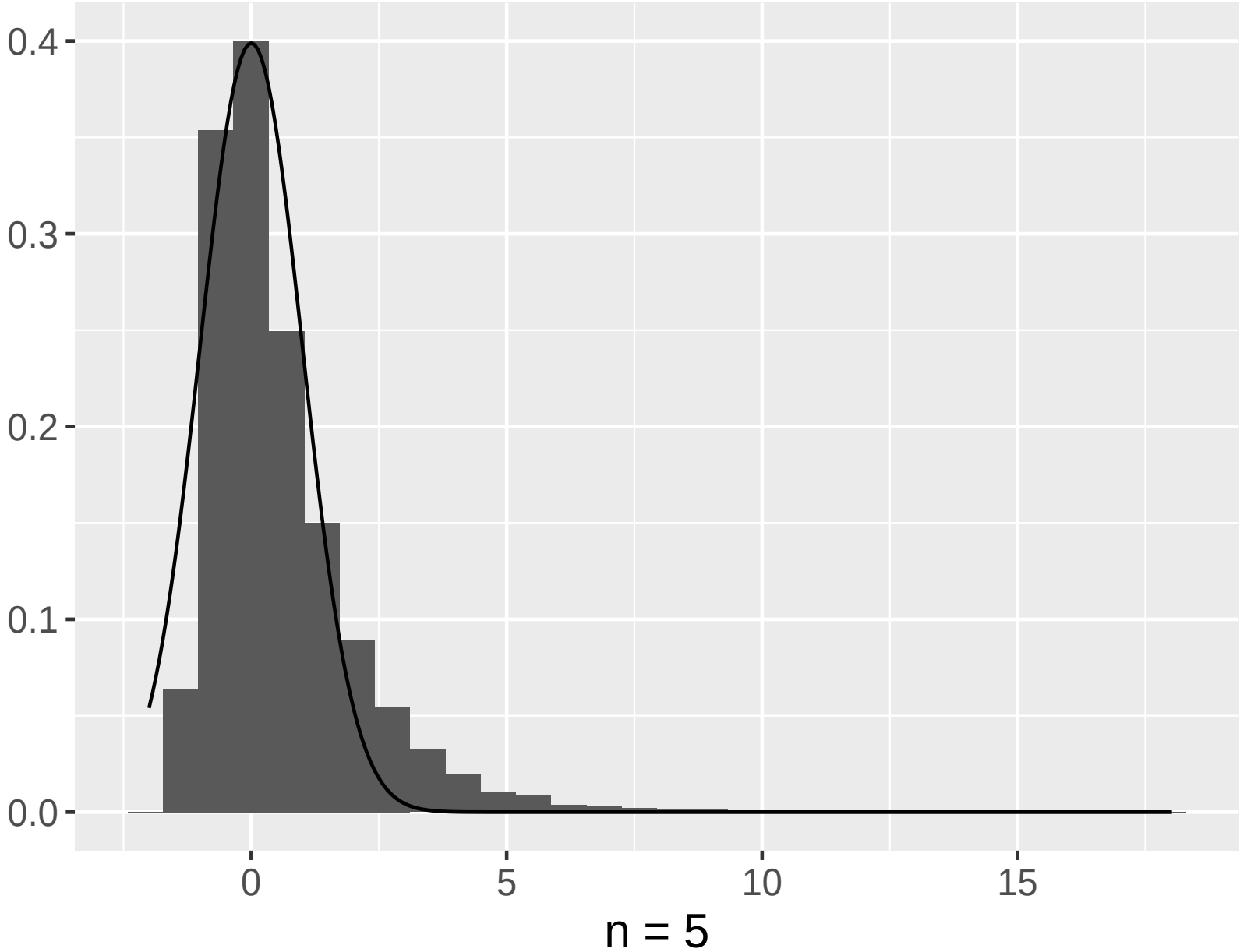
```r
# Inference depends on design: Ex 2
# Design 1
pbinom(7, 25, 0.2, lower.tail = F) # 1-binomial CDF

## [1] 0.1091228
```

```r
# Inference depends on design: Ex 2
# Design 2
pnbinom(7, 17, 1-0.2, lower.tail = F) # 1 - neg-binom CDF

## [1] 0.08917126
```

目錄 I

Sample Size Time to Event I

```r
set.seed(1234)
lambda = 0.025
base_draws = matrix(rexp(100 * 1e4), nrow = 1e4)
nsamp = 5
library(ggplot2)
ggplot() +
  geom_histogram(aes(
    x = ( 1/rowMeans(base_draws[ ,1:nsamp,drop=F]/lambda) - lambda) /
      (lambda / sqrt(nsamp)),
    y = ..density..),
    bins=30) +
  stat_function(aes(x = c(-2,2)), fun = dnorm, n = 301,
                args = list(mean = 0, sd = 1)) +
  theme(text = element_text(size = 14)) +
  labs(x = "n = 5", y = "")
```


0.4
0.3
0.2
0.1
0.0
0
5
10
15
n = 5

- try $n = 50$ and $n = 100$.

Thanks!