

Lab 4: Randomization schemes

13/01/2026

Contents

Άσκηση 1 – Απλή τυχαιοποίηση (Simple Randomization)	2
i. Πιθανότητα ακριβούς ισορροπίας	2
ii. Προσομοίωση μιας τυχαιοποίησης	2
iii. Επανάληψη του πειράματος 1000 φορές	3
Άσκηση 2 – Τυχαιοποίηση σε blocks	4
i. Πλήθος δυνατών αναθέσεων εντός block	4
ii. Block randomization με τυχαία block sizes	4
Άσκηση 3 – Blocking μέσα σε strata (π.χ. φύλο)	5
Άσκηση 4 – Urn scheme	6
i. Υλοποίηση ως συνάρτηση	6
ii. Προσομοίωση 2500 αναθέσεων και γράφημα πιθανότητας	6
Άσκηση 5 – Σύγκριση SR vs Block vs Urn σε μικρό N	8

Άσκηση 1 – Απλή τυχαιοποίηση (Simple Randomization)

i. Πιθανότητα ακριβούς ισορροπίας

Θέλουμε να τυχαιοποιήσουμε $N = 100$ άτομα σε 2 σκέλη (A/B) με πιθανότητα 0.5 για κάθε σκέλος.

1. Υπολόγισε την πιθανότητα να πάρουμε ακριβώς 50 στο A και 50 στο B.
2. Ερμήνευσε το αποτέλεσμα: είναι «μεγάλη» ή «μικρή» πιθανότητα;

Υπόδειξη: $N_A \sim \text{Binomial}(N, 0.5)$ και μπορείς να χρησιμοποιήσεις τη `dbinom()`.

```
N <- 100
# P(N_A = 50)
dbinom(x = N/2, size = N, p = 0.5)
```

ii. Προσομοίωση μιας τυχαιοποίησης

Προσομοίωσε μία απλή τυχαιοποίηση $N = 100$ ατόμων σε A/B, μέσω `runif()`:

1. Φτιάξε ένα dataset με `id` και `trt` (θεραπεία).
2. Θέσε `seed` (π.χ. 125) για αναπαραγωγιμότητα.
3. Δημιούργησε `u <- runif(N)` και ανάθεσε:
 - A αν $u < 0.5$
 - B αν $u \geq 0.5$
4. Υπολόγισε πόσοι πήγαν σε κάθε σκέλος και τα ποσοστά.

```
N <- 100
data <- data.frame(id = paste0("id", 1:N), trt = NA_character_)

set.seed(125)
u <- runif(N)

data$trt[u < 0.5] <- "A"
data$trt[u >= 0.5] <- "B"

head(data, 10)
table(data$trt)
table(data$trt) / N
```

iii. Επανάληψη του πειράματος 1000 φορές

Επανάλαβε την απλή τυχαιοποίηση $K = 1000$ φορές, αποθήκευσε για κάθε replication το πλήθος στο A και στο B και υπολόγισε:

1. Τη μέση τιμή και τη διακύμανση του πλήθους στο A.
2. Το ποσοστό replications όπου το A είναι >60 ή <40 (ανισορροπία χειρότερη από 60/40).

```
N <- 100
K <- 1000
set.seed(2342)

results <- data.frame(replication = 1:K, NumberA = NA_integer_, NumberB =
  ↪ NA_integer_)

for (i in 1:K) {
  u <- runif(N)
  trt <- ifelse(u < 0.5, "A", "B")
  tab <- table(trt)
  results$NumberA[i] <- tab["A"]
  results$NumberB[i] <- tab["B"]
}

head(results)

# P(imbalance > 60/40)
mean(results$NumberA > 60 | results$NumberA < 40)

# Means & variances
colMeans(results[, c("NumberA", "NumberB")])
apply(results[, c("NumberA", "NumberB")], 2, var)
```

Άσκηση 2 – Τυχαιοποίηση σε blocks

Θεωρούμε 2 σκέλη (A/B) και block size 4.

i. Πλήθος δυνατών αναθέσεων εντός block

Πόσες διαφορετικές αναθέσεις υπάρχουν σε block των 4 ατόμων με ακριβώς 2 A και 2 B;

```
choose(4, 2)
```

ii. Block randomization με τυχαία block sizes

Χρησιμοποίησε το πακέτο `blockrand` για να τυχαιοποιήσεις 100 άτομα με τυχαία block sizes (π.χ. 2, 4, 6, 8 μέσω `block.sizes = 1:4`).

1. Πόσες συνολικά γραμμές επιστρέφει η `blockrand()` (μπορεί να είναι >100);
2. Ποια είναι η τελική κατανομή A/B στα **πρώτα 100** άτομα;

```
library(blockrand)

set.seed(1322)
block_rand <- blockrand(
  n = 100,
  num.levels = 2,
  levels = c("A", "B"),
  block.sizes = 1:4
)

# Preview πρώτων blocks
block_rand[block_rand$block.id %in% 1:2, ]

nrow(block_rand)
table(block_rand$treatment[1:100])
```

Άσκηση 3 – Blocking μέσα σε strata (π.χ. φύλο)

Θέλουμε 55 άνδρες και 45 γυναίκες (σύνολο 100). Κάνουμε block randomization *ξεχωριστά* σε κάθε stratum και μετά ενώνουμε.

1. Κάνε block randomization για Male και Female.
2. Κράτα τους πρώτους 55 άνδρες και 45 γυναίκες.
3. Έλεγξε ότι:
 - Συνολικά A/B είναι κοντά στο 50/50
 - Μέσα σε κάθε φύλο, η κατανομή είναι παρόμοια.
4. Κάνε έλεγχο χ^2 ανεξαρτησίας μεταξύ θεραπείας και φύλου (με Monte Carlo p-value).

```
library(blockrand)

set.seed(2435)
male   <- blockrand(n = 100, id.prefix = "M", block.prefix = "M", stratum =
  ↪ "Male")
female <- blockrand(n = 100, id.prefix = "F", block.prefix = "F", stratum =
  ↪ "Female")

male   <- male[1:55, ]
female <- female[1:45, ]

mystudy <- rbind(male, female)

table(mystudy$treatment)

x <- table(mystudy$treatment, mystudy$stratum)
x

chisq.test(x, simulate.p.value = TRUE, B = 10000)
```

Άσκηση 4 – Urn scheme

Υλοποίησε ένα απλό urn scheme όπου:

- Ξεκινάμε με 1 μπάλα A και 1 μπάλα B.
- Σε κάθε draw, τραβάμε A με πιθανότητα $\frac{n_A}{n_A + n_B}$.
- **Μετά** προσθέτουμε 1 μπάλα από την *αντίθετη* θεραπεία (δηλ. αν βγει A, προσθέτουμε μία B).

i. Υλοποίηση ως συνάρτηση

Γράψε `urnModel(n)` που επιστρέφει: - `treatment`: η ακολουθία αναθέσεων - `probA`: η πιθανότητα A σε κάθε βήμα

```
urnModel <- function(n, na0 = 1, nb0 = 1) {  
  na <- na0  
  nb <- nb0  
  
  probA <- rep(NA_real_, n)  
  trt <- rep(NA_character_, n)  
  u <- runif(n)  
  
  for (i in 1:n) {  
    probA[i] <- na / (na + nb)  
  
    if (u[i] < probA[i]) {  
      trt[i] <- "A"  
      nb <- nb + 1 # add opposite ball  
    } else {  
      trt[i] <- "B"  
      na <- na + 1 # add opposite ball  
    }  
  }  
  
  data.frame(treatment = trt, probA = probA)  
}
```

ii. Προσομοίωση 2500 αναθέσεων και γράφημα πιθανότητας

1. Τρέξε `urnModel(2500)` και δες πόσοι καταλήγουν σε A/B.
2. Σχεδίασε `probA` ως προς το βήμα και πρόσθεσε οριζόντια γραμμή στο 0.5.

```
set.seed(12)
draws <- urnModel(n = 2500)

table(draws$treatment)

plot(draws$probA, type = "l",
      xlab = "Draw",
      ylab = "P(A)",
      main = "Urn scheme: probability of assigning A")
abline(h = 0.5, lty = 2)
```

Άσκηση 5 – Σύγκριση SR vs Block vs Urn σε μικρό N

Θεώρησε μικρή μελέτη $N = 14$ και επανάλαβε 200 φορές:

- SR: απλή τυχαιοποίηση
- BR: block randomization (τυχαία block sizes)
- Urn: urn scheme

Σύγκρινε μέση τιμή και τυπική απόκλιση της αναλογίας που παίρνει A.

```
set.seed(5)
N <- 14
it <- 200

results <- data.frame(
  SR = rep(NA_real_, it),
  BR = rep(NA_real_, it),
  urn = rep(NA_real_, it)
)

for (i in 1:it) {
  # SR
  trt_sr <- ifelse(runif(N) < 0.5, "A", "B")
  results$SR[i] <- mean(trt_sr == "A")

  # BR
  br <- blockrand(n = N, num.levels = 2, levels = c("A", "B"),
                 block.sizes = 1:4)[1:N, ]
  results$BR[i] <- mean(br$treatment == "A")

  # urn
  trt_urn <- urnModel(N)$treatment
  results$urn[i] <- mean(trt_urn == "A")
}

colMeans(results)
apply(results, 2, sd)
```