Francis I. Proctor Foundation Guide to Randomization

Contributors: Ben Arnold

Compiled: 2021-03-03

Contents

Welcome!

This is a guide for best (essential!) practices in trial randomization and masking. At the Proctor Foundation, we lead many randomized, controlled trials to study intervention effects. Many of our trials are masked (aka “blinded”), whereby treatment allocation is concealed from participants, investigators, and/or outcome assessors. This short guide is a compendium our team’s best practices around these activities.

IMPORTANT: The randomization and masking steps are among the most important activities to ensure a trial’s validity. Jeopardizing one or both can undermine a trial. Team members involved in generating an allocation sequence and masking a trial should work directly with at least one of our faculty biostatisticians. At present, those faculty members include Ben Arnold and Travis Porco.

Chapter 1

Overview and Introduction

Contributors: Ben Arnold

Random allocation of treatment to units (individuals or clusters) is perhaps the single strongest design tool we have in epidemiology and clinical research to estimate the causal effect of a treatment on outcomes. Randomization ensures that individuals who receive treatment are, on average, exchangeable with those who do not (Altman and Bland, 1999b). Without randomization, individuals who seek or receive treatment are almost inevitably different from those who do not, often in immeasurable ways. This leads to confounding of the treatment-outcome relationship.

Importantly (Altman and Bland, 1999b):

> "The term random does not mean the same as haphazard but has a precise technical meaning. By random allocation we mean that each patient has a known chance, usually an equal chance, of being given each treatment, but the treatment to be given cannot be predicted."

In practice, generating an allocation sequence involves the following steps:

Table 1.1: Randomization Steps

Steps	
1	Finalize the study design and randomization plan, including specifics about allocation ratio, any blocking/stratification, and masking.
2	Create a randomization subdirectory within the trial's project directory to save the randomization files. If randomization is masked, you will need to save the randomization files in a separate, tightly controlled directory (sync'd to the cloud for secure backup).

Steps	
3	Write a script to generate a random sequence. If the trial is masked, use temporary letters.
4	Assess randomization diagnostics to ensure that the randomization sequence behaves as expected.
5	Share the randomization sequence and diagnostics with the PI and trial's biostatistician. Have at least 2 people review the randomization script and diagnostics to check for any errors.
6	If the trial is masked, work with the trial's unmasked biostatistician to assign the final letters to each treatment group using the agreed upon, private mapping between letters and treatment group.
7	Set a new seed, and generate the final sequence. Store the sequence in a `.csv` file in the randomization directory and lock the file to ensure it cannot be over-written.

This short guide will cover details for each step, and will elaborate in some areas where appropriate (e.g., providing examples of how to generate stratified or blocked sequences).

Additionally, the guide will cover the important topics of Masking and Allocation Concealment, which are key steps to ensuring the validity of the randomization sequence in the conduct of the trial.

This guide does not currently include guidance for advanced topics, such as response-adaptive allocation.

Chapter 2

A few comments on computing

Contributors: Ben Arnold

Like all of our data science workflows, generating a random sequence needs to be transparent and reproducible. See Proctor's handbook on data science. The principles we describe there are germane for randomization sequences as well!

https://proctor-ucsf.github.io/dcc-handbook/intro.html

All of the examples in this guide use R software. There are surely many other effective ways to generate sequences in other software, but R includes many convenient functions for pseudo-random number generation.

Our advice for generating the sequence is to not rely on any packages beyond base `R`. The R language and packages evolve rapidly. Using base R ensures that functions will behave consistently over time. For example, many tidyverse packages such as the `dplyr` package are incredible for data manipulation. `dplyr` includes many convenient pseudo-random sampling routines, but as of this writing its syntax rapidly evolves – some commonly used functions seem to be replaced or deprecated every few months. This makes the code more fragile. As an exception, we do use `ggplot2` for graphics in randomization diagnostics examples below.

Chapter 3

Creating a Randomization Directory

Contributors: Ben Arnold

The files used to generate a randomization sequence should live in one directory. They should be clearly labeled. They should include lots of comments and documentation to orient a new reader to their contents. The directory should include, at minimum:

Table 3.1: Checklist of items that should be in the Randomization Directory

Item	
______ 1.	A metadata README file that describes all files in the directory.
______ 2.	The script that generates the randomization sequence(s).
______ 3.	Randomization diagnostics (typically an output file from the script).
______ 4.	The randomization sequence(s) generated, stored as `.csv` files.
______ 5.	The key that maps group labels to masked codes (masked trials only)
______ 6.	A list of team members with access to the masked codes (masked trials only) and randomization sequence file(s)

For unmasked randomization, we recommend creating a subdirectory within a trial's parent directory called `Randomization`. In the hypothetical `MyTrial` study below, create a new subdirectory nested within:

```
~/Box Sync/MyTrial/Randomization
```

If the randomization sequence is masked, then we need to keep the randomization files separate from the main trial directory. Otherwise team members who should not know the mapping between treatment labels and masking codes could discover the link. The sequence could also be discoverable, and if a team member is involved in treating patients that could bias the allocation.

In masked trials, we recommend creating a parallel, shadow directory for the trial with restricted access permissions. We typically use the suffix `-unmasked-materials` to identify the restricted access directories. They should live on the same encrypted server as the trial's main directory, e.g.:

```
~/Box Sync/MyTrial-unmasked-materials/Randomization
```

Note that an `-unmasked-materials` directory can contain other, sensitive, restricted access materials. Such as: interim analyses requested by the trial's Data and Safety Monitoring Committee, and the primary analysis files/results up until the trial is unmasked.

We recommend the `-unmasked-materials` directory reside on a secure, cloud server that meets UCSF's IT security standards. At this time, that includes Box and Microsoft OneDrive. The trial's biostatistician should "own" the directory and should carefully control team member access. Restrict permissions/access to only team members who should have access (item #6, above). Do not create a shareable link to the directory or individual files within the directory.

If the randomization script is developed on GitHub, then the trial should maintain a secure repostitory that includes unmasked materials and is highly restricted. Additional details in the chapter on Masking.

Table 3.2: Key Points for Restricted Access Directories in Masked Trials

Key Points	
1	The trial's biostatistician should control access to a trial's directory of unmasked materials
2	At least 3 team members should have access to unmasked materials at all times

Chapter 4

Generating Randomization Sequences

Contributors: Ben Arnold

4.1 Introduction

This chapter includes guides for specific types of randomization. It includes code snippets to illustrate the different types of randomization. In the examples below, we focus on trials with two groups and equal allocation. Multi-arm trials are conceptually similar. Unequal allocation is similar as well, but raises particular issues around masking, described in the Masking chapter.

In all of the examples, below, we will use masked examples with 2 letter per arm (so 4 letters total). The rationale for 2+ letters per arm is that if a single letter is accidentally unmasked, then the entire trial will not be unmasked.

These examples will carry forward into the next chapter on Randomization Diagnostics. The template files stored in this guide's Github respository (**TBD**) provide complete examples of how to put everything together.

Essential background reading:

Altman, D. G. & Martin Bland, J. Treatment allocation in controlled trials: why randomise? BMJ 318, 1209–1209 (1999). https://pubmed.ncbi.nlm.nih.gov/10221955/ (Altman and Bland, 1999b)

Altman, D. G. & Bland, J. M. How to randomise. BMJ 319, 703–704 (1999). https://pubmed.ncbi.nlm.nih.gov/10480833/ (Altman and Bland, 1999a)

Schulz, K. F. & Grimes, D. A. Generation of allocation sequences in randomised trials: chance, not choice. Lancet 359, 515–519 (2002). https://pubmed.ncbi.nl m.nih.gov/11853818/ (Schulz and Grimes, 2002)

4.2 Simple randomization

> *No other allocation generation approach, irrespective of its complexity and sophistication, surpasses the unpredictability and bias prevention of simple randomisation.*

High praise from masters of clinical trials! (Schulz and Grimes, 2002) Simple (unrestricted) randomization is the purest form of randomization. Patients (or clusters) are allocated to each arm with a known (usually equal) probability. For larger trials (hundreds of units) without specific design considerations this might be just the right tool. Many trials led by Proctor faculty favor simple randomization.

Schulz and Grimes highlight a downside, which can be particularly germane for smaller trials. With simple randomization, at any point in the sequence the numbers of patients allocated to each treatment group will probably differ, which is not a major problem in terms of statistical efficiency (though perfectly balanced is statistically ideal). For a two-arm trial, the chance of pronounced imbalance becomes negligible with trial sizes greater than 200 (Lachin, 1988; Schulz and Grimes, 2002). But, if 20 participants are randomized with equal probability to two treatment groups, the chance of a 12:8 split (i.e., 60% A, 40% B) or worse is approximately 50%. For 100 participants, the chance of the same ratio (60:40 split) or worse is only 5% (Friedman et al., 2015). For smaller trials and multi-center trials this chance imbalance could be a problem, and so many trials use blocking and/or stratification in the randomization sequence.

Here is an example of generating a simple randomization sequence:

```r
#----------------------------
# Example of simple randomization
#
# 2 arm trial
# 2 letters per arm (masked)
#----------------------------

#----------------------------
# create a list of 1000 IDs.
# in a real trial, we may already
# have a list or sampling frame
# to work with
#
# pro tip: always ensure your IDs start with a character
# to prevent accidental conversion to numeric variables!
#----------------------------
nobs <- 1000
```

```
id <- paste0("ID",1:nobs)

#----------------------------
# Identify letters for
# the randomization
#
# In a masked trial, we would
# start with placeholder letters
# to vet the sequence generation
# and then change them for the
# final randomization.
# See step 3 in the Overview
#
# In this example, we use
# A, B, T, V
#----------------------------
rand_letters <- c("A","B","T","V")

#----------------------------
# set a seed
#
# this is essential for
# creating a reproducible
# sequence!  After vetting the
# randomization with the broader
# study team (steps 4-5 in the Overview),
# we can change the seed to create a new
# (protected) sequence
#----------------------------
set.seed(123)

#----------------------------
# generate the sequence
#
# the sample() function
# samples the rand_letters
# with equal probability
# up to the number of IDs
# using replacement
#----------------------------
tr_masked <- sample(rand_letters, size = length(id), replace = TRUE)

#----------------------------
# combine the IDs with the
# sequence in a data.frame
```

```
#
# this object would be stored
# in the Randomization Directory
# as a .csv file
#----------------------------
rand_seq <- data.frame(id = id, tr_masked = tr_masked)


#----------------------------
# for the sake of example,
# unmask and examine the
# allocation.
#
# Pretend letters A and T
# are for placebo
#----------------------------
tr <- ifelse(tr_masked %in% c("A","T"),"Placebo","Treatment")
table(tr)
```

```
## tr
##   Placebo Treatment
##       506       494
```

In this example, 51% of the participants were allocated to placebom and 49% were allocated to treatment — a pretty good balance! This would be expected given the trial's large size.

Try running the code above with `nobs <- 80` to see what a typical randomization looks like:

```
#----------------------------
# From the above example
# create a function
# that takes as arguments
# a vector of IDs and
# masked treatment letters
#----------------------------
randomize_simple <- function(ids, letters) {
  tr_masked <- sample(rand_letters, size = length(id), replace = TRUE)
  return( data.frame(id = ids, tr_masked = tr_masked) )
}

#----------------------------
# setting size, creating IDs
# and identifying letters,
# as in the above example
#----------------------------
```

```
nobs <- 80
id <- paste0("ID",1:nobs)
rand_letters <- c("A","B","T","V")
#----------------------------
# generating a new sequence
# using the randomize_simple()
# function
#----------------------------
set.seed(123)
rand_seq2 <- randomize_simple(ids = id, letters = rand_letters)
rand_seq2$tr <- ifelse(rand_seq2$tr_masked %in% c("A","T"),"Placebo","Treatment")
table(rand_seq2$tr)
```

```
##
##   Placebo Treatment
##        48        32
```

With the same seed, 48/80 (60%) are allocated to placebo, which is not ideal. Blocking can prevent chance imbalances.

4.3 Blocked randomization

4.4 Stratified randomization

4.5 Matched randomization

Chapter 5

Randomization Diagnostics

Contributors: Ben Arnold

Chapter 6

Masking

Contributors: Ben Arnold

Chapter 7

Allocation Concealment

Contributors: Ben Arnold

Bibliography

Altman, D. G. and Bland, J. M. (1999a). How to randomise. *BMJ*, 319(7211):703–704.

Altman, D. G. and Bland, M. J. (1999b). Treatment allocation in controlled trials: why randomise? *BMJ*, 318(7192):1209–1209.

Friedman, L. M., Furberg, C. D., DeMets, D. L., Reboussin, D. M., and Granger, C. B. (2015). *Fundamentals of Clinical Trials*. Springer, Cham.

Lachin, J. M. (1988). Properties of simple randomization in clinical trials. *Control. Clin. Trials*, 9(4):312–326.

Schulz, K. F. and Grimes, D. A. (2002). Generation of allocation sequences in randomised trials: chance, not choice. *Lancet*, 359(9305):515–519.