

# Package ‘CARM’

June 7, 2023

**Title** Covariate-Adjusted Adaptive Randomization via Mahalanobis-Distance

**Version** 1.1.0

**Author** Haoyu Yang [aut],  
Fanglu Chen [aut, cre],  
Yichen Qin [aut],  
Yang Li [aut]

**Maintainer** Fanglu Chen <chenfanglu1031@qq.com>

**Description** In randomized controlled trial (RCT), balancing covariate is often one of the most important concern. CARM package provides functions to balance the covariates and generate allocation sequence by covariate-adjusted Adaptive Randomization via Mahalanobis-distance (ARM) for RCT. About what ARM is and how it works please see Y. Qin, Y. Li, W. Ma, H. Yang, and F. Hu (2022). "Covariate-adjusted adaptive randomization via Mahalanobis distance" *Statistica Sinica*. <doi:10.5705/ss.202020.0440>. In addition, the package is also suitable for the randomization process of multi-arm trials. For details, please see Yang H, Qin Y, Wang F, et al. (2023). "Balancing covariates in multi-arm trials via adaptive randomization" *Computational Statistics & Data Analysis*.<doi:10.1016/j.csda.2022.107642>.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R(>= 3.6.0),stats

**Imports** arrangements,dplyr,MASS

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-06-07 16:40:14 UTC

## R topics documented:

CARM-package . . . . .	2
ARM . . . . .	2
ARMM . . . . .	4

**Index****6**


---

CARM-package	<i>CARM: Covariate-adjusted Adaptive Randomization via Mahalanobis-distance</i>
--------------	---

---

**Description**

The CARM package provides function of implement of randomization:

**ARM functions**

please ?ARM and ?ARMM to view function usage

---

ARM	<i>Adaptive Randomization via Mahalanobis Distance</i>
-----	--

---

**Description**

Allocates patients to one of two treatments using Adaptive Randomization via Mahalanobis Distance proposed by Yichen Qin, Yang Li, Wei Ma, Haoyu Yang, and Feifang Hu.(2022)

**Usage**

ARM(covariate, assignment, q = 0.75)

**Arguments**

covariate	a data frame. A row of the dataframe corresponds to the covariate profile of a patient.
assignment	a vector. If partial patients had been allocated, please input their allocation. IF all the patients are not be allocated, please input 'assignment = NA' directly.
q	the biased coin probability. $q$ should be larger than 1/2 and less than 1, default = 0.75

**Details**

Suppose that  $n$  patients are to be assigned to two treatment groups. Consider  $p$  continuous covariates for each patient.  $T_i$  is the assignment of the  $i$ th patient. The proposed procedure to assign units to treatment groups, namely adaptive randomization via Mahalanobis distance (ARM), is outlined below.

- (1) Arrange all  $n$  units randomly into a sequence  $x_1, \dots, x_n$ .
- (2) Assign the first two units with  $T_1 = 1$  and  $T_2 = 2$ .
- (3) Suppose that  $2i$  units have been assigned to treatment groups, for the  $2i + 1$ -th and  $2i + 2$ -th units:

(3a) If the  $2i + 1$ -th unit is assigned to treatment 1 and the  $2i + 2$ -th unit to treatment 2, then calculate the potential Mahalanobis distance, between the updated treatment groups. with  $2i + 2$  units,  $M_1(2i + 2)$ .

(3b) Similarly, if the  $2i + 1$ -th unit is assigned to treatment 2 and the  $2i + 2$ -th unit to treatment 1, then calculate the other potential Mahalanobis distance,  $M_2(2i + 2)$ .

(4) Assign the  $2i + 1$ -th unit to treatment groups according to the following probabilities:

if  $M_1(2i + 2) < M_2(2i + 2)$ ,  $P(T_{2i+1} = 1) = q$ ;

if  $M_1(2i + 2) > M_2(2i + 2)$ ,  $P(T_{2i+1} = 1) = 1 - q$ ;

if  $M_1(2i + 2) = M_2(2i + 2)$ ,  $P(T_{2i+1} = 1) = 0.5$ .

(5) Repeat the last two steps until all units are assigned. If  $n$  is odd, assign the last unit to two treatments with equal probabilities.

Mahalanobis distance  $M(n)$  between the sample means across different treatment groups is:

$$M(n) = np(1 - p)(\hat{x}_1 - \hat{x}_2)^T cov(x)^{-1}(\hat{x}_1 - \hat{x}_2)$$

See the reference for more details.

### Value

An object of class "ARM" is a list containing the following components:

assignment	Allocation of patients.
sample_size	The number of patients in treatment 1 and treatment 2 respectively.
Mahalanobis_Distance	Mahalanobis distance between treatment groups 1 and 2.

### References

Qin, Y., Y. Li, W. Ma, H. Yang, and F. Hu (2022). Adaptive randomization via mahalanobis distance. *Statistica Sinica*.DOI:<10.5705/ss.202020.0440>.

### Examples

```
library(MASS)
#simulate covariates of patients
p <- 6; n <- 30
sigma <- diag(p); mean <- c(rep(0,p))
data <- mvrnorm(n, mean, sigma)
covariate <- as.data.frame(data)
#IF all the patients are not be allocated
ARM(covariate = covariate, assignment = NA, q=0.75)
#IF you had allocated partial patients
ARM(covariate = covariate,assignment = c(1,2),q=0.75)
```

ARMM

*Adaptive Randomization via Mahalanobis distance for Multi-arm design*

### Description

Randomize patients into treatment groups for multi-arm trials using ARMM proposed by Haoyu Yang, Yichen Qin, Yang Li, Fan Wang, and Feifang Hu.(2022)

### Usage

ARMM(covariate, assignment, K, q = 0.75, method)

### Arguments

covariate	a data frame. A row of the dataframe corresponds to the covariate profile of a patient.
assignment	a vector. If partial patients had been allocated, please input their allocation. IF all the patients are not be allocated, please input 'assignment = NA' directly.
K	an integer; number of arms of the trial.
q	the biased coin probability. $q$ should be larger than 1/2 and less than 1, default = 0.75
method	Methods for calculating Mahalanobis distance, input one of these texts: 'mean', 'max' or 'median'.

### Details

Suppose  $n$  units (participants) are to be assigned to  $K$  treatment groups. For each unit  $i, i = 1, \dots, n$  and treatment  $j, j = 1, \dots, K$ , define the assignment matrix  $[T_{ij}]^{n \times K}$ , where  $T_{ij} = 1$  indicates unit  $i$  receives treatment  $j$ . Consider  $p$  continuous covariates, let  $x_i = (x_{i1}, \dots, x_{in})^T$ .

The proposed method, namely the adaptive randomization via Mahalanobis distance for multi-arm design (ARMM), is outlined below. The implement of ARMM is similar to ARM.

First assume that  $n$  units are in a sequence and then assign the first  $K$  units to  $K$  treatment groups randomly as the initialization. Then, the following units are assigned in blocks of  $K$  sequentially and adaptively until all the units are assigned. For  $K$  units are assigned to  $K$  groups, there are in total  $K!$  possible allocations. Calculate  $K!$  potential overall covariate imbalance measurement according to pairwise Mahalanobis distance under the  $K!$  possible allocations. Choose the allocation which corresponds to the smallest Mahalanobis distance with a probability of  $q$  across all potential allocations. Repeat the process until all units are assigned.

For any pair of treatments  $s$  and  $t$  among the  $K$  treatment groups, calculate the Mahalanobis distance by:

$$M_{s,t}(n) = 2n/K/K(\hat{x}_1 - \hat{x}_2)^T cov(x)^{-1}(\hat{x}_1 - \hat{x}_2)$$

In total, there are  $C_K^2$  pairs of Mahalanobis distances among  $K$  treatment groups. Finally, calculate the mean, the median or the maximum to represent the total imbalance.

See the reference for more details.

### Value

An object of class "ARMM" is a list containing the following components:

assignment	Allocation of patients.
sample_size	The number of patients from treatment 1 to treatment $K$ respectively.
Mahalanobis_Distance	Mahalanobis distance among treatment groups .

### References

Yang H, Qin Y, Wang F, et al. Balancing covariates in multi-arm trials via adaptive randomization. *Computational Statistics & Data Analysis*, 2023, 179: 107642. <https://doi.org/10.1016/j.csda.2022.107642>

### Examples

```
library(MASS)
#simulate covariates of patients
p <- 6; n <- 30
sigma <- diag(p); mean <- c(rep(0,p))
data <- mvrnorm(n, mean, sigma)
covariate <- as.data.frame(data)
#IF all the patients are not be allocated
ARMM(covariate = covariate, assignment = NA, K = 3, q = 0.75, method = 'mean')
#IF you had allocated partial patients
ARMM(covariate = covariate, assignment = c(1,2), K=4, q=0.75, method = 'max')
```

# Index

ARM, [2](#)

ARMM, [4](#)

CARM-package, [2](#)