

R Generate and Combine Fixed and Random Matrix

Fan Wang

2022-07-14

Contents

1	Generate Matrixes	1
1.1	Create a N by 2 Matrix from 3 arrays	1
1.2	Name Matrix Columns and Rows	1
1.3	Generate NA Matrix	1
1.4	Generate Random Matrixes	2
1.5	Replace a Subset of Matrix Values by NA_real_	2
1.6	Sort Each Matrix Row or Column	3
1.7	Compute Column and Row Statistics	3
1.8	Add Column to Matrix with Common Scalar Value	4

1 Generate Matrixes

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) research support package, [R4Econ](#) examples page, [PkgTestR](#) packaging guide, or [Stat4Econ](#) course page.

1.1 Create a N by 2 Matrix from 3 arrays

Names of each array become row names automatically.

```
ar_row_one <- c(-1,+1)
ar_row_two <- c(-3,-2)
ar_row_three <- c(0.35,0.75)

mt_n_by_2 <- rbind(ar_row_one, ar_row_two, ar_row_three)
kable(mt_n_by_2) %>%
  kable_styling_fc()
```

ar_row_one	-1.00	1.00
ar_row_two	-3.00	-2.00
ar_row_three	0.35	0.75

1.2 Name Matrix Columns and Rows

```
# An empty matrix with Logical NA
mt_named <- matrix(data=NA, nrow=2, ncol=2)
colnames(mt_named) <- paste0('c', seq(1,2))
rownames(mt_named) <- paste0('r', seq(1,2))
mt_named
```

```
##      c1 c2
## r1 NA NA
## r2 NA NA
```

1.3 Generate NA Matrix

- Best way to allocate matrix in R, NULL vs NA?

Allocate with NA or NA_real_ or NA_int_. Clarity in type definition is preferred.

```
# An empty matrix with Logical NA
mt_na <- matrix(data=NA, nrow=2, ncol=2)
str(mt_na)

##  logi [1:2, 1:2] NA NA NA NA

# An empty matrix with numerical NA
mt_fl_na <- matrix(data=NA_real_, nrow=2, ncol=2)
mt_it_na <- matrix(data=NA_integer_, nrow=2, ncol=2)

str(mt_fl_na)

##  num [1:2, 1:2] NA NA NA NA
str(mt_fl_na)

##  num [1:2, 1:2] NA NA NA NA
```

1.4 Generate Random Matrixes

Random draw from the normal distribution, random draw from the uniform distribution, and combine resulting matrixes.

```
# Generate 15 random normal, put in 5 rows, and 3 columns
mt_rnorm <- matrix(rnorm(15,mean=0,sd=1), nrow=5, ncol=3)

# Generate 15 random normal, put in 5 rows, and 3 columns
mt_runif <- matrix(runif(15,min=0,max=1), nrow=5, ncol=5)

# Combine
mt_rnorm_runif <- cbind(mt_rnorm, mt_runif)

# Display
kable(round(mt_rnorm_runif, 3)) %>% kable_styling_fc()
```

-0.059	-0.566	2.431	0.762	0.560	0.553	0.762	0.560
-0.422	-2.415	0.213	0.426	0.508	0.656	0.426	0.508
0.928	0.117	-0.405	0.210	0.044	0.068	0.210	0.044
0.117	-1.314	0.741	0.152	0.541	0.347	0.152	0.541
-0.594	-0.013	2.213	0.711	0.871	0.184	0.711	0.871

1.5 Replace a Subset of Matrix Values by NA_real_

For values in matrix that fall below or above some thresholds, we will replace these values by NA_real_.

```
fl_max_val <- 0.8
fl_min_val <- 0.2
mt_rnorm_runif_bd <- mt_rnorm_runif
```

```
mt_rnorm_runif_bd[which(mt_rnorm_runif < fl_min_val)] <- NA_real_
mt_rnorm_runif_bd[which(mt_rnorm_runif > fl_max_val)] <- NA_real_
# Print
print(mt_rnorm_runif_bd)
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
## [1,]  NA  NA      NA 0.7617321 0.5599549 0.5525358 0.7617321 0.5599549
## [2,]  NA  NA 0.2130297 0.4261897 0.5075784 0.6564292 0.4261897 0.5075784
## [3,]  NA  NA      NA 0.2097634      NA      NA 0.2097634      NA
## [4,]  NA  NA 0.7412710      NA 0.5413033 0.3466240      NA 0.5413033
## [5,]  NA  NA      NA 0.7114064      NA      NA 0.7114064      NA
```

1.6 Sort Each Matrix Row or Column

Now we sort within each row or within each column of the random matrix.

```
# Within row sort
mt_rnorm_runif_row_sort <- t(apply(
  mt_rnorm_runif, 1, sort
))
# Within column sort, note no transpose
mt_rnorm_runif_col_sort <- apply(
  mt_rnorm_runif, 2, sort
)
# Display
kable(round(mt_rnorm_runif_row_sort, 3),
      caption="Each row sort low to high") %>%
  kable_styling_fc()
```

Each row sort low to high

-0.566	-0.059	0.553	0.560	0.560	0.762	0.762	2.431
-2.415	-0.422	0.213	0.426	0.426	0.508	0.508	0.656
-0.405	0.044	0.044	0.068	0.117	0.210	0.210	0.928
-1.314	0.117	0.152	0.152	0.347	0.541	0.541	0.741
-0.594	-0.013	0.184	0.711	0.711	0.871	0.871	2.213

```
kable(round(mt_rnorm_runif_col_sort, 3),
      caption="Each column sort low to high") %>%
  kable_styling_fc()
```

Each column sort low to high

-0.594	-2.415	-0.405	0.152	0.044	0.068	0.152	0.044
-0.422	-1.314	0.213	0.210	0.508	0.184	0.210	0.508
-0.059	-0.566	0.741	0.426	0.541	0.347	0.426	0.541
0.117	-0.013	2.213	0.711	0.560	0.553	0.711	0.560
0.928	0.117	2.431	0.762	0.871	0.656	0.762	0.871

1.7 Compute Column and Row Statistics

Compute column and row means, and also column and row sums

```

print(paste0('colSums=',
             paste(round(
               colSums(mt_rnorm_runif),3), collapse=',')
             ))

## [1] "colSums=-0.03,-4.192,5.193,2.261,2.524,1.808,2.261,2.524"

print(paste0('colMeans=',
             paste(round(
               colMeans(mt_rnorm_runif),3), collapse=',')
             ))

## [1] "colMeans=-0.006,-0.838,1.039,0.452,0.505,0.362,0.452,0.505"

print(paste0('rowSums=',
             paste(round(
               rowSums(mt_rnorm_runif),3), collapse=',')
             ))

## [1] "rowSums=5.001,-0.1,1.215,1.278,4.954"

print(paste0('rowMeans=',
             paste(round(
               rowMeans(mt_rnorm_runif),3), collapse=',')
             ))

## [1] "rowMeans=0.625,-0.013,0.152,0.16,0.619"

```

1.8 Add Column to Matrix with Common Scalar Value

Given some matrix of information, add a column, where all rows of the column have the same numerical value. Use the matrix created prior. - R add column to matrix - r append column to matrix constant value

```

fl_new_first_col_val <- 111
fl_new_last_col_val <- 999
mt_with_more_columns <- cbind(rep(fl_new_first_col_val, dim(mt_rnorm_runif)[1]),
                              mt_rnorm_runif,
                              rep(fl_new_last_col_val, dim(mt_rnorm_runif)[1]))

# Display
kable(mt_with_more_columns) %>% kable_styling_fc_wide()

```

111	-0.0592404	-0.5659461	2.4307278	0.7617321	0.5599549	0.5525358	0.7617321	0.5599549	999
111	-0.4217741	-2.4152576	0.2130297	0.4261897	0.5075784	0.6564292	0.4261897	0.5075784	999
111	0.9278903	0.1171755	-0.4048819	0.2097634	0.0438403	0.0680209	0.2097634	0.0438403	999
111	0.1167738	-1.3144461	0.7412710	0.1523807	0.5413033	0.3466240	0.1523807	0.5413033	999
111	-0.5939258	-0.0131548	2.2126501	0.7114064	0.8708764	0.1839549	0.7114064	0.8708764	999