R Generate and Combine Fixed and Random Matrix

Fan Wang

2022-07-23

Contents

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.

```r
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

```r
# 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.

```r
# 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
```

```r
# An empty matrix with numerica 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
```

```r
str(mt_fl_na)
```

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

1.4 Generate Matrixes with values

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

```r
# 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()
```

-0.869	-0.145	-0.727	0.341	0.949	0.646	0.341	0.949
0.564	-0.258	-1.269	0.574	0.495	0.235	0.574	0.495
0.389	0.633	-0.175	0.107	0.299	0.410	0.107	0.299
1.494	1.259	0.175	0.241	0.166	0.712	0.241	0.166
0.455	0.087	-0.351	0.759	0.600	0.117	0.759	0.600

Now we generate a matrix with sequential integers, and either fill matrix by columns or fill matrix by rows.

```r
# with byrow set to FALSE, will fill first col, then second col, etc..
mt_index_colbycol <- matrix(seq(0, 15), nrow=4, ncol=4, byrow=FALSE)
# Display
kable(mt_index_colbycol,
  caption= "with byrow=FALSE, the default, will fill col by col") %>%
  kable_styling_fc()
```

with byrow=FALSE, the default, will fill col by col

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

```
# with byrow set to TRUE, will fill row by row
mt_index_rowbyrow <- matrix(seq(0, 15), nrow=4, ncol=4, byrow=TRUE)
# Display
kable(mt_index_rowbyrow,
  caption= " with byrow=TRUE, will fill row by row") %>%
  kable_styling_fc()
```

with byrow=TRUE, will fill row by row

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

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 0.3406442        NA 0.6461887 0.3406442        NA
## [2,] 0.5636412       NA   NA 0.5740784 0.4954167 0.2352184 0.5740784 0.4954167
## [3,] 0.3888201 0.632777   NA        NA 0.2990886 0.4102063        NA 0.2990886
## [4,]        NA       NA   NA 0.2409072        NA 0.7118945 0.2409072        NA
## [5,] 0.4548058       NA   NA 0.7587683 0.6003126        NA 0.7587683 0.6003126
```

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

-0.869	-0.727	-0.145	0.341	0.341	0.646	0.949	0.949
-1.269	-0.258	0.235	0.495	0.495	0.564	0.574	0.574
-0.175	0.107	0.107	0.299	0.299	0.389	0.410	0.633
0.166	0.166	0.175	0.241	0.241	0.712	1.259	1.494
-0.351	0.087	0.117	0.455	0.600	0.600	0.759	0.759

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

Each column sort low to high

-0.869	-0.258	-1.269	0.107	0.166	0.117	0.107	0.166
0.389	-0.145	-0.727	0.241	0.299	0.235	0.241	0.299
0.455	0.087	-0.351	0.341	0.495	0.410	0.341	0.495
0.564	0.633	-0.175	0.574	0.600	0.646	0.574	0.600
1.494	1.259	0.175	0.759	0.949	0.712	0.759	0.949

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=2.033,1.576,-2.347,2.022,2.51,2.12,2.022,2.51"
```

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

```
## [1] "colMeans=0.407,0.315,-0.469,0.404,0.502,0.424,0.404,0.502"
```

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

```
## [1] "rowSums=1.485,1.41,2.07,4.454,3.026"
```

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

```
## [1] "rowMeans=0.186,0.176,0.259,0.557,0.378"
```

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

```r
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	-0.8691096	-0.1445619	-0.7268035	0.3406442	0.9487628	0.6461887	0.3406442	0.9487628	999
111	0.5636412	-0.2580963	-1.2692544	0.5740784	0.4954167	0.2352184	0.5740784	0.4954167	999
111	0.3888201	0.6327770	-0.1753339	0.1074422	0.2990886	0.4102063	0.1074422	0.2990886	999
111	1.4943747	1.2587139	0.1753828	0.2409072	0.1660050	0.7118945	0.2409072	0.1660050	999
111	0.4548058	0.0873159	-0.3512261	0.7587683	0.6003126	0.1169283	0.7587683	0.6003126	999