Matrix Refactor

Ali Taqi

Symmetric Normal Matrices

```
RM_normal <- function(M, normal_args = c(0,1), symm = F){
  # Extract parameters
  mu <- normal_args[1]
  sd <- normal_args[2]
  # Create [M x M] transition matrix
  P <- matrix(rep(NA, M * M), ncol = M)
  # Generate rows
  for(i in 1:M){
    P[i,] <- rnorm(n = M, mean = mu, sd = sd)
  }
  # Make symmetric if prompted
  if(symm == T){
    P[lower.tri(P)] <- P[upper.tri(P)]
    }
  # Return the matrix
  P
}
```

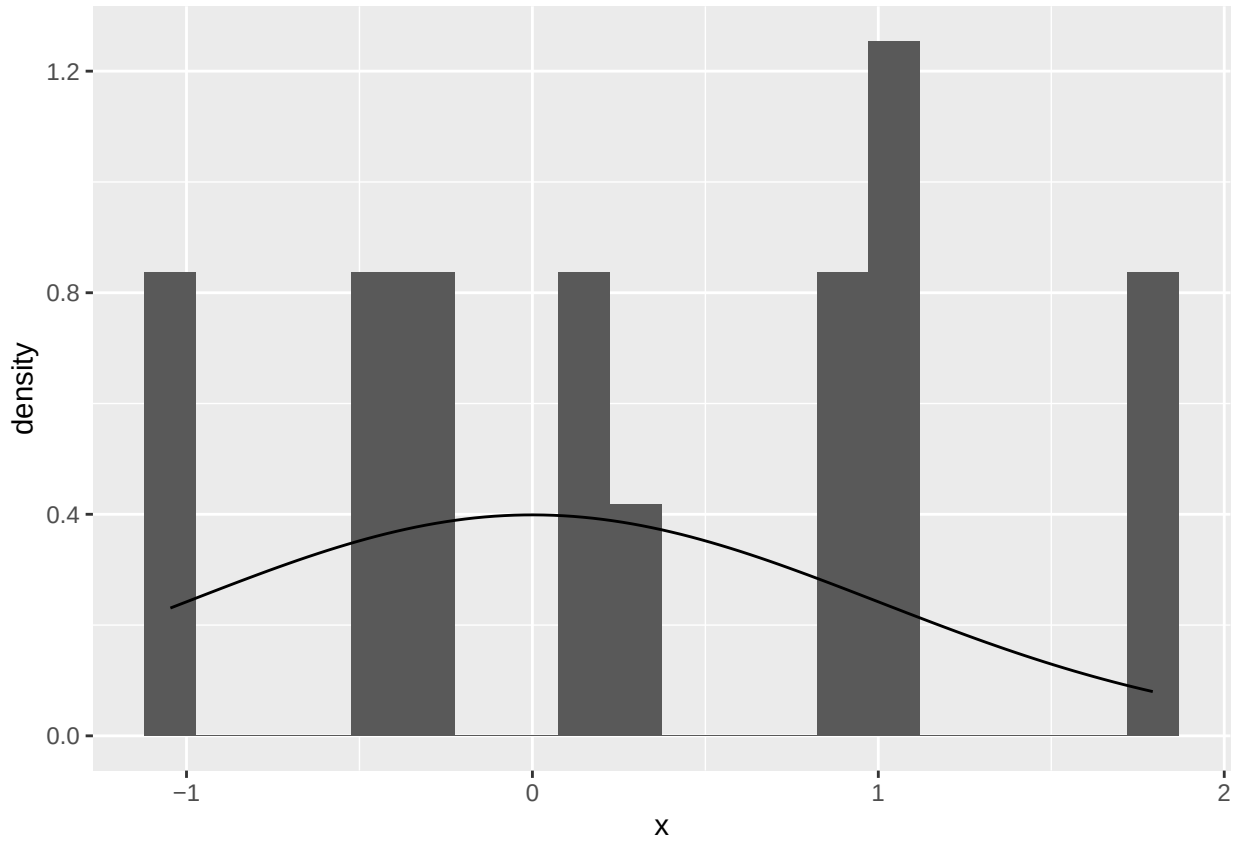
```
# Set seed
set.seed(23)
# Set parameters
M <- 4
mu <- 0
sd <- 1
normal_args <- c(mu, sd)
# Generate matrix
P <- RM_normal(M, normal_args, symm = T)
P
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,]  0.1932123 -0.4346821  0.9132671  1.7933881
## [2,] -0.4346821  1.1074905 -0.2780863  1.0192055
## [3,]  0.9132671  1.7933881  0.2182885 -1.0465353
## [4,] -0.2780863  1.0192055 -1.0465353  0.3081369
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: FALSE"
```

```
visualize_normal_entries(P, normal_args)
```



Non-symmetric Normal Matrices

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
mu <- 1
sd <- 2
# Generate matrixx
P <- RM_normal(M, normal_args = c(mu, sd), symm = F)
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: FALSE"
```

Stochastic Matrices

Sparse

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
mu <- 1
sd <- 2
# Generate matrixx
P <- RM_stoch(M, symm = F, sparsity = T)
P
```

```
##      [,1]      [,2]      [,3]
## [1,]    0 0.4019552 0.5980448
## [2,]    0 1.0000000 0.0000000
## [3,]    1 0.0000000 0.0000000
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: FALSE"
```

```
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: TRUE"
```

Non-sparse

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
mu <- 1
sd <- 2
# Generate matrix
P <- RM_stoch(M, symm = F, sparsity = F)
P
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5095594 0.1971352 0.2933055
## [2,] 0.3637477 0.4193927 0.2168595
## [3,] 0.3463251 0.3515677 0.3021073
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: FALSE"
```

```
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: TRUE"
```

Symmetric Stochastic Matrices

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
# Generate matrix
P <- RM_stoch(M, symm = T, sparsity = T)
P
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.0000000 0.4019552 0.5980448
## [2,] 0.4019552 0.5980448 0.0000000
## [3,] 0.5980448 0.0000000 0.4019552
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: TRUE"
```

```
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: TRUE"
```

Non-sparse

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
mu <- 1
sd <- 2
# Generate matrix
P <- RM_stoch(M, symm = T, sparsity = F)
P
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5095594 0.1971352 0.2933055
## [2,] 0.1971352 0.5860053 0.2168595
## [3,] 0.2933055 0.2168595 0.4898350
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: TRUE"
```

```
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: TRUE"
```

Tridiagonal Matrices

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
# Generate matrix
P <- RM_trid(M)
P
```

```
##           [,1]       [,2]      [,3]
## [1,] 0.3864247  1.7933881 0.0000000
## [2,] 1.7933881 -0.8693642 0.9966051
## [3,] 0.0000000  0.9966051 1.8265342
```

```
paste("Matrix is symmetric: ",isSymmetric(P),sep="")
```

```
## [1] "Matrix is symmetric: TRUE"
```

p-Sparse Matrices

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
# Set parameters
p <- 0.2
# Generate matrix
P <- RM_erdos(M, p, stoch = F)
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: FALSE"
```

Stochastic p-Sparse Matrices

```
# Set seed
set.seed(23)
# Set parameters
M <- 3
p <- 0.2
# Generate matrix
P <- RM_erdos(M, p, stoch = T)
paste("Matrix is row-stochastic: ",is_row_stochastic(P),sep="")
```

```
## [1] "Matrix is row-stochastic: TRUE"
```

Suppose we have a $M \times M$ square matrix $\mathbf{P}$ (for some $M \in \mathbb{N}$) on a field F. We notate $\mathbf{P} \in \mathcal{M}_F[M^2]$. Take $\mathbf{P} \in \mathcal{M}_F[M^2]$.

Structural Properties of Matrices

If $\mathbf{P}$ is symmetric, then its upper triangle is equal to the lower triangle.

If $\mathbf{P}$ is tridiagonal, then it is a band matrix of width 1.

Entry-wise Properties of Matrices

If $\mathbf{P}$ is row-stochastic, then $\forall i : \sum_j p_{ij} = 1$.

```
RM_stoch <- function(M, symm = F, sparsity = F){...}
```

If $\mathbf{P}$ is $\mathcal{N}(\mu, \sigma^2)$, then its entries satisfy $p_{ij} \sim \mathcal{N}(\mu, \sigma^2)$.

```
RM_normal <- function(M, normal_args = c(0,1), symm = F){...}
```

If $\mathbf{P}$ is $p-$sparse, then $\forall i, j \in S_M : p_{ij}/c \sim \text{Bern}(p)$ for some $c \in \mathbb{R}$.

```
RM_erdos <- function(M, p_sparse){...}
```