Moving towards R package **Matrix** version 2.0-0

Mikael Jagan[1,2] Martin Mächler[1,3]

[1]R Core Team

[2]Department of Data Science
Dana-Farber Cancer Institute
Boston, Massachusetts, U.S.A.

[3]Seminar for Statistics
ETH Zurich
Zurich, Switzerland

useR! @ Warsaw
July 9, 2026

What is **Matrix**?

- Classes for representing matrices
  - Data type: numeric, logical, ...
  - Structure: "general", symmetric, ...
  - Storage format: sparse or dense
- Methods for efficient operations on objects
- Interoperable with **base** vectors, matrices
- S4-based: multiple inheritance, multiple dispatch

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

An abridged history

2000	0.2-1	Douglas Bates publishes initial version on CRAN
2004	0.8-2	Use of S4, C libraries for sparse linear algebra
2005	0.95-1	Martin Mächler joins as coauthor
2006	0.9975-0	Machinery for mixed models is moved to **lme4**
2009	0.999375-24	**Matrix** is "recommended" with R 2.9.0
2011	1.0-0	Stable development begins after 156 0.y-z releases
2022	1.4-1	Mikael Jagan joins as coauthor
2026	1.7-5	2175 reverse dependencies (13448 recursively)
202?	2.0-0	This talk

The class hierarchy

- Virtual class `Matrix`
- Virtual subclasses
  - Data type: `dMatrix`, `lMatrix`, ...
  - Structure: `generalMatrix`, `symmetricMatrix`, ...
  - Storage format: `sparseMatrix` or `denseMatrix`
- Further extension by `[CRT]sparseMatrix`, ...
- Nonvirtual subclasses *xyy*`Matrix`, by "restricted crossing"

Nonvirtual subclasses *xyy*Matrix

```
> library(Matrix)
> packageVersion("Matrix")
[1] '1.7.5'
> M <- getClass("Matrix")
> Msub <- sort(Filter(Negate(isVirtualClass),
+                     names(M@subclasses)))
> matrix(grepv("^.g.Matrix$", Msub), nrow = 4)
     [,1]         [,2]         [,3]
[1,] "dgCMatrix" "lgCMatrix" "ngCMatrix"
[2,] "dgRMatrix" "lgRMatrix" "ngRMatrix"
[3,] "dgTMatrix" "lgTMatrix" "ngTMatrix"
[4,] "dgeMatrix" "lgeMatrix" "ngeMatrix"
> extends("dgCMatrix")
[1] "dgCMatrix"     "CsparseMatrix" "dsparseMatrix"
[4] "generalMatrix" "dMatrix"       "sparseMatrix"
[7] "Matrix"
> length(Msub)
[1] 51
```

Nonvirtual subclasses *xyy*Matrix

```
> library(Matrix)
> packageVersion("Matrix")
[1] '1.7.5'
> M <- getClass("Matrix")
> Msub <- sort(Filter(Negate(isVirtualClass),
+                     names(M@subclasses)))
> matrix(grepv("^.g.Matrix$", Msub), nrow = 4)
     [,1]         [,2]         [,3]
[1,] "dgCMatrix" "lgCMatrix" "ngCMatrix"
[2,] "dgRMatrix" "lgRMatrix" "ngRMatrix"
[3,] "dgTMatrix" "lgTMatrix" "ngTMatrix"
[4,] "dgeMatrix" "lgeMatrix" "ngeMatrix"
> extends("dgCMatrix")
[1] "dgCMatrix"     "CsparseMatrix" "dsparseMatrix"
[4] "generalMatrix" "dMatrix"       "sparseMatrix"
[7] "Matrix"
> length(Msub)
[1] 51
```

Nonvirtual subclasses *xyy*Matrix

```
> library(Matrix)
> packageVersion("Matrix")
[1] '1.7.5'
> M <- getClass("Matrix")
> Msub <- sort(Filter(Negate(isVirtualClass),
+                     names(M@subclasses)))
> matrix(grepv("^.g.Matrix$", Msub), nrow = 4)
     [,1]         [,2]         [,3]
[1,] "dgCMatrix" "lgCMatrix" "ngCMatrix"
[2,] "dgRMatrix" "lgRMatrix" "ngRMatrix"
[3,] "dgTMatrix" "lgTMatrix" "ngTMatrix"
[4,] "dgeMatrix" "lgeMatrix" "ngeMatrix"
> extends("dgCMatrix")
[1] "dgCMatrix"     "CsparseMatrix" "dsparseMatrix"
[4] "generalMatrix" "dMatrix"       "sparseMatrix"
[7] "Matrix"
> length(Msub)
[1] 51
```

Nonvirtual subclasses *xyy*Matrix

```
> library(Matrix)
> packageVersion("Matrix")
[1] '1.7.5'
> M <- getClass("Matrix")
> Msub <- sort(Filter(Negate(isVirtualClass),
+                     names(M@subclasses)))
> matrix(grepv("^.g.Matrix$", Msub), nrow = 4)
     [,1]         [,2]         [,3]
[1,] "dgCMatrix" "lgCMatrix" "ngCMatrix"
[2,] "dgRMatrix" "lgRMatrix" "ngRMatrix"
[3,] "dgTMatrix" "lgTMatrix" "ngTMatrix"
[4,] "dgeMatrix" "lgeMatrix" "ngeMatrix"
> extends("dgCMatrix")
[1] "dgCMatrix"      "CsparseMatrix" "dsparseMatrix"
[4] "generalMatrix" "dMatrix"       "sparseMatrix"
[7] "Matrix"
> length(Msub)
[1] 51
```

Nonvirtual subclasses *xyy*Matrix

```
> library(Matrix)
> packageVersion("Matrix")
[1] '1.7.5'
> M <- getClass("Matrix")
> Msub <- sort(Filter(Negate(isVirtualClass),
+                     names(M@subclasses)))
> matrix(grepv("^.g.Matrix$", Msub), nrow = 4)
     [,1]         [,2]         [,3]
[1,] "dgCMatrix" "lgCMatrix" "ngCMatrix"
[2,] "dgRMatrix" "lgRMatrix" "ngRMatrix"
[3,] "dgTMatrix" "lgTMatrix" "ngTMatrix"
[4,] "dgeMatrix" "lgeMatrix" "ngeMatrix"
> extends("dgCMatrix")
[1] "dgCMatrix"     "CsparseMatrix" "dsparseMatrix"
[4] "generalMatrix" "dMatrix"       "sparseMatrix"
[7] "Matrix"
> length(Msub)
[1] 51
```

Deprecated "specific" coercions

Since **Matrix** version 1.5-0 (released in 2022):

```
> (L <- new("lgeMatrix"))
0 x 0 Matrix of class "lgeMatrix"
<0 x 0 matrix>
> as(L, "dgeMatrix")
'as(<lgeMatrix>, "dgeMatrix")' is deprecated.
Use 'as(., "dMatrix")' instead.
See help("Deprecated") and help("Matrix-deprecated").
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> as(L, "dMatrix")
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> com <- capture.output(showMethods("coerce", includeDefs=TRUE))
> sum(grepl("DeprecatedCoerce", com))
[1] 186
```

Deprecated "specific" coercions

Since **Matrix** version 1.5-0 (released in 2022):

```
> (L <- new("lgeMatrix"))
0 x 0 Matrix of class "lgeMatrix"
<0 x 0 matrix>
> as(L, "dgeMatrix")
'as(<lgeMatrix>, "dgeMatrix")' is deprecated.
Use 'as(., "dMatrix")' instead.
See help("Deprecated") and help("Matrix-deprecated").
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> as(L, "dMatrix")
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> com <- capture.output(showMethods("coerce", includeDefs=TRUE))
> sum(grepl("DeprecatedCoerce", com))
[1] 186
```

Deprecated "specific" coercions

Since **Matrix** version 1.5-0 (released in 2022):

```
> (L <- new("lgeMatrix"))
0 x 0 Matrix of class "lgeMatrix"
<0 x 0 matrix>
> as(L, "dgeMatrix")
'as(<lgeMatrix>, "dgeMatrix")' is deprecated.
Use 'as(., "dMatrix")' instead.
See help("Deprecated") and help("Matrix-deprecated").
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> as(L, "dMatrix")
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> com <- capture.output(showMethods("coerce", includeDefs=TRUE))
> sum(grepl("DeprecatedCoerce", com))
[1] 186
```

Deprecated "specific" coercions

Since **Matrix** version 1.5-0 (released in 2022):

```
> (L <- new("lgeMatrix"))
0 x 0 Matrix of class "lgeMatrix"
<0 x 0 matrix>
> as(L, "dgeMatrix")
'as(<lgeMatrix>, "dgeMatrix")' is deprecated.
Use 'as(., "dMatrix")' instead.
See help("Deprecated") and help("Matrix-deprecated").
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> as(L, "dMatrix")
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> com <- capture.output(showMethods("coerce", includeDefs=TRUE))
> sum(grepl("DeprecatedCoerce", com))
[1] 186
```

Deprecated "specific" coercions

Since **Matrix** version 1.5-0 (released in 2022):

```
> (L <- new("lgeMatrix"))
0 x 0 Matrix of class "lgeMatrix"
<0 x 0 matrix>
> as(L, "dgeMatrix")
'as(<lgeMatrix>, "dgeMatrix")' is deprecated.
Use 'as(., "dMatrix")' instead.
See help("Deprecated") and help("Matrix-deprecated").
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> as(L, "dMatrix")
0 x 0 Matrix of class "dgeMatrix"
<0 x 0 matrix>
> com <- capture.output(showMethods("coerce", includeDefs=TRUE))
> sum(grepl("DeprecatedCoerce", com))
[1] 186
```

Matrix objects with integer data

- Under **Matrix**-release (1.7-5) and **Matrix**-devel (to become 1.8-0):

```
> (J <- Matrix(1:6, ncol = 3))
2 x 3 Matrix of class "dgeMatrix"
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> typeof(J@x)
[1] "double"
```

- Under **Matrix**-devel:

```
> (J <- Matrix(1:6, ncol = 3, keepInteger = TRUE))
2 x 3 Matrix of class "igeMatrix"
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> typeof(J@x)
[1] "integer"
```

Matrix objects with integer data

- Under **Matrix**-release (1.7-5) and **Matrix**-devel (to become 1.8-0):

```
> (J <- Matrix(1:6, ncol = 3))
2 x 3 Matrix of class "dgeMatrix"
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> typeof(J@x)
[1] "double"
```

- Under **Matrix**-devel:

```
> (J <- Matrix(1:6, ncol = 3, keepInteger = TRUE))
2 x 3 Matrix of class "igeMatrix"
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> typeof(J@x)
[1] "integer"
```

Matrix objects with complex data

- Under **Matrix**-release:

```
> (Z <- Matrix(c(1, -1i, 1i, 1), ncol = 2))
Error in [........] : invalid 'data'
```

- Under **Matrix**-devel:

```
> (Z <- Matrix(c(1, -1i, 1i, 1), ncol = 2))
2 x 2 Matrix of class "zsyMatrix"
     [,1] [,2]
[1,] 1+0i 0+1i
[2,] 0-1i 1+0i
> Z@trans
[1] "C"
> Z@trans <- "T"
> Z
2 x 2 Matrix of class "zsyMatrix"
     [,1] [,2]
[1,] 1+0i 0+1i
[2,] 0+1i 1+0i
```

Matrix objects with complex data

- Under **Matrix**-release:

```
> (Z <- Matrix(c(1, -1i, 1i, 1), ncol = 2))
Error in [........] : invalid 'data'
```

- Under **Matrix**-devel:

```
> (Z <- Matrix(c(1, -1i, 1i, 1), ncol = 2))
2 x 2 Matrix of class "zsyMatrix"
     [,1] [,2]
[1,] 1+0i 0+1i
[2,] 0-1i 1+0i
> Z@trans
[1] "C"
> Z@trans <- "T"
> Z
2 x 2 Matrix of class "zsyMatrix"
     [,1] [,2]
[1,] 1+0i 0+1i
[2,] 0+1i 1+0i
```

New idiom for linear system solution, determinant, etc.

- Old idiom, relying on factorization caching:

```
A <- Matrix(...)
X1 <- solve(A, B1)
X2 <- solve(A, B2)
detA <- determinant(A)
```

- New idiom, since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A) # maybe BunchKaufman(), Cholesky(), ...
X1 <- solve(luA, B1)
X2 <- solve(luA, B2)
detA <- determinant(luA)
```

New idiom for linear system solution, determinant, etc.

- Old idiom, relying on factorization caching:

```
A <- Matrix(...)
X1 <- solve(A, B1)
X2 <- solve(A, B2)
detA <- determinant(A)
```

- New idiom, since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A) # maybe BunchKaufman(), Cholesky(), ...
X1 <- solve(luA, B1)
X2 <- solve(luA, B2)
detA <- determinant(luA)
```

Nicer expansions of factorizations

Methods for generic function `expand` are deprecated in favour of ones for `expand2` and `expand1`, new since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand(luA)
##  dense: list(   L, U, P) where   A = P L U
## sparse: list(P, L, U, Q) where P A = L U Q

all.equal(A, Reduce(`%*%`, listOfFactors)) # *not* in general
```

Nicer expansions of factorizations

Methods for generic function `expand` are deprecated in favour of ones for `expand2` and `expand1`, new since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand(luA)
##  dense: list(   L, U, P) where   A = P L U
## sparse: list(P, L, U, Q) where P A = L U Q

all.equal(A, Reduce(`%*%`, listOfFactors)) # *not* in general
```

Nicer expansions of factorizations

Methods for generic function `expand` are deprecated in favour of ones for `expand2` and `expand1`, new since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand(luA)
##  dense: list(   L, U, P) where   A = P L U
## sparse: list(P, L, U, Q) where P A = L U Q

all.equal(A, Reduce(`%*%`, listOfFactors)) # *not* in general
```

Nicer expansions of factorizations

Methods for generic function `expand` are deprecated in favour of ones for `expand2` and `expand1`, new since **Matrix** version 1.6-0 (released in 2023):

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand(luA)
##  dense: list(   L, U, P) where   A = P L U
## sparse: list(P, L, U, Q) where P A = L U Q

all.equal(A, Reduce(`%*%`, listOfFactors)) # *not* in general
```

Nicer expansions of factorizations

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand2(luA)
##  dense: list(P1., L, U     ) where P1 A    = L U
## sparse: list(P1., L, U, P2.) where P1 A P2 = L U

all.equal(A, Reduce(`%*%`, listOfFactors)) # TRUE

expand1(luA, "L") # or "U", "P1", "P1.", ...
```

Nicer expansions of factorizations

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand2(luA)
##  dense: list(P1., L, U     ) where P1 A    = L U
## sparse: list(P1., L, U, P2.) where P1 A P2 = L U

all.equal(A, Reduce(`%*%`, listOfFactors)) # TRUE

expand1(luA, "L") # or "U", "P1", "P1.", ...
```

Nicer expansions of factorizations

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand2(luA)
##  dense: list(P1., L, U     ) where P1 A    = L U
## sparse: list(P1., L, U, P2.) where P1 A P2 = L U

all.equal(A, Reduce(`%*%`, listOfFactors)) # TRUE

expand1(luA, "L") # or "U", "P1", "P1.", ...
```

Nicer expansions of factorizations

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand2(luA)
##  dense: list(P1., L, U     ) where P1 A    = L U
## sparse: list(P1., L, U, P2.) where P1 A P2 = L U

all.equal(A, Reduce(`%*%`, listOfFactors)) # TRUE

expand1(luA, "L") # or "U", "P1", "P1.", ...
```

Nicer expansions of factorizations

```
A <- Matrix(...)
luA <- lu(A)

listOfFactors <- expand2(luA)
##  dense: list(P1., L, U     ) where P1 A    = L U
## sparse: list(P1., L, U, P2.) where P1 A P2 = L U

all.equal(A, Reduce(`%*%`, listOfFactors)) # TRUE

expand1(luA, "L") # or "U", "P1", "P1.", ...
```

More systematic documentation

Under **Matrix**-devel, consolidated `help("*-class")`.
E.g., `help("CsparseMatrix-class")`:

Subclasses

The nonvirtual subclasses of `CsparseMatrix` defined in package **Matrix** are listed below with links to virtual superclasses defining their data type and structure.

Nonvirtual subclass	Data type	Structure
ngCMatrix	nMatrix	generalMatrix
lgCMatrix	lMatrix	generalMatrix
igCMatrix	iMatrix	generalMatrix
dgCMatrix	dMatrix	generalMatrix
zgCMatrix	zMatrix	generalMatrix
nsCMatrix	nMatrix	symmetricMatrix
lsCMatrix	lMatrix	symmetricMatrix
isCMatrix	iMatrix	symmetricMatrix
dsCMatrix	dMatrix	symmetricMatrix
zsCMatrix	zMatrix	symmetricMatrix
dpCMatrix	dMatrix	posdefMatrix
zpCMatrix	zMatrix	posdefMatrix
ntCMatrix	nMatrix	triangularMatrix
ltCMatrix	lMatrix	triangularMatrix
itCMatrix	iMatrix	triangularMatrix
dtCMatrix	dMatrix	triangularMatrix
ztCMatrix	zMatrix	triangularMatrix

More systematic documentation

Under **Matrix**-devel, `help("*-methods")` works for every generic function for which we define methods.

E.g., `help("length-methods")`:

length-methods {Matrix} R Documentation

Length of an Object

Description

S4 methods defined in package **Matrix** for S4 generic functions `length` and `length<-`.

Usage

```
## S4 method for signature 'Matrix'
length(x)
## S4 method for signature 'MatrixFactorization'
length(x)
## S4 method for signature 'sparseVector'
length(x)

## S4 replacement method for signature 'denseMatrix,numeric'
length(x) <- value
## S4 replacement method for signature 'sparseMatrix,numeric'
length(x) <- value
## S4 replacement method for signature 'sparseVector,numeric'
length(x) <- value
```

Arguments

x
: a `Matrix` object, a `MatrixFactorization` object, or a `sparseVector` object.

Debugging fixes (R bug #18822)

Until R 4.5:

```
> selectMethod("dim", signature = c(x="Matrix"))
Method Definition:

function (x)
x@Dim
<bytecode: 0x8f6f3e238>
<environment: namespace:Matrix>

Signatures:
        x
target  "Matrix"
defined "Matrix"
> debugonce("dim", signature = c(x="Matrix"))
Error in [........] :
  Function must be an S4 generic
```

Debugging fixes (R bug #18822)

Until R 4.5:

```
> selectMethod("dim", signature = c(x="Matrix"))
Method Definition:

function (x)
x@Dim
<bytecode: 0x8f6f3e238>
<environment: namespace:Matrix>

Signatures:
        x
target  "Matrix"
defined "Matrix"
> debugonce("dim", signature = c(x="Matrix"))
Error in [........] :
  Function must be an S4 generic
```

Debugging fixes (R bug #18822)

Until R 4.5:

```
> selectMethod("dim", signature = c(x="Matrix"))
Method Definition:

function (x)
x@Dim
<bytecode: 0x8f6f3e238>
<environment: namespace:Matrix>

Signatures:
        x
target  "Matrix"
defined "Matrix"
> debugonce("dim", signature = c(x="Matrix"))
Error in [........] :
  Function must be an S4 generic
```

Debugging fixes (R bug #18824)

Until R 4.5:

```
> debugonce(dim, signature = c(x="Matrix"))
Tracing specified method for function "dim" in environment
<namespace:base>
> dim(new("dgCMatrix"))
Tracing dim(new("dgCMatrix")) on entry
Untracing specified method for function "dim" in package
"namespace:base"
Called from: eval(expr, p)
Browse[1]> Q
> debugonce(dim, signature = c(x="Matrix"))
Error in [........] :
  cannot use 'at' argument unless the function body has the
  form '{ ... }'
```

Debugging fixes (R bug #18824)

Until R 4.5:

```
> debugonce(dim, signature = c(x="Matrix"))
Tracing specified method for function "dim" in environment
<namespace:base>
> dim(new("dgCMatrix"))
Tracing dim(new("dgCMatrix")) on entry
Untracing specified method for function "dim" in package
"namespace:base"
Called from: eval(expr, p)
Browse[1]> Q
> debugonce(dim, signature = c(x="Matrix"))
Error in [.......] :
  cannot use 'at' argument unless the function body has the
  form '{ ... }'
```

Debugging fixes (R bug #18824)

Until R 4.5:

```
> debugonce(dim, signature = c(x="Matrix"))
Tracing specified method for function "dim" in environment
<namespace:base>
> dim(new("dgCMatrix"))
Tracing dim(new("dgCMatrix")) on entry
Untracing specified method for function "dim" in package
"namespace:base"
Called from: eval(expr, p)
Browse[1]> Q
> debugonce(dim, signature = c(x="Matrix"))
Error in [.......] :
  cannot use 'at' argument unless the function body has the
  form '{ ... }'
```

Debugging fixes (R bug #18824)

Until R 4.5:

```
> debugonce(dim, signature = c(x="Matrix"))
Tracing specified method for function "dim" in environment
<namespace:base>
> dim(new("dgCMatrix"))
Tracing dim(new("dgCMatrix")) on entry
Untracing specified method for function "dim" in package
"namespace:base"
Called from: eval(expr, p)
Browse[1]> Q
> debugonce(dim, signature = c(x="Matrix"))
Error in [........] :
  cannot use 'at' argument unless the function body has the
  form '{ ... }'
```

Debugging fixes (R bug #18823)

Until R 4.5:

```
> ## debug as(<matrix>, "CsparseMatrix")
> debugonce(coerce, signature = c(from="matrix", to="Matrix"))
Tracing specified method for function "coerce" in environment
<namespace:methods>
> as(diag(6L), "Matrix")
Tracing asMethod(object) on entry
Error in eval(expr, p) : object '.Generic' not found
```

Debugging fixes (R bug #18823)

Until R 4.5:

```
> ## debug as(<matrix>, "CsparseMatrix")
> debugonce(coerce, signature = c(from="matrix", to="Matrix"))
Tracing specified method for function "coerce" in environment
<namespace:methods>
> as(diag(6L), "Matrix")
Tracing asMethod(object) on entry
Error in eval(expr, p) : object '.Generic' not found
```

Debugging fixes (R bug #18823)

Until R 4.5:

```
> ## debug as(<matrix>, "CsparseMatrix")
> debugonce(coerce, signature = c(from="matrix", to="Matrix"))
Tracing specified method for function "coerce" in environment
<namespace:methods>
> as(diag(6L), "Matrix")
Tracing asMethod(object) on entry
Error in eval(expr, p) : object '.Generic' not found
```

Moving towards **Matrix** version 2.0-0

- Publish **Matrix**-devel on CRAN as version 1.8-0
- Submit 2.0-0 once obvious bugs have been squashed
- Explore sparse format matrices with $> 2^{31} - 1$ nonzero entries