The NA of character type is distinct from the string "NA". Programmers who need to specify an explicit missing string should use NA\_character\_ (rather than "NA") or set elements to NA using is.na<-.

"**Boolean Indexing**":

x <- 1:20 # Creates a vector from 1 to 20

x %% 5 == 0 #This basically says: "Is x divisible by 5?"

[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE

## Now, key step -- we assign the previous line to a variable called "mask":

mask <- x %% 5 == 0

## The reason why this is significant is because we can now INDEX "x" with "mask":

x[mask]

[1] 5 10 15 20

Invisible(x)- stops autoprinting, but will still return x

library(datasets)

data(iris)

to strip a column vector from a dataspace:

eye <- iris[,5]

seq1 <- seq(1:4)

> mat1 <- matrix(seq1, 2)

> solve(mat1)

[,1] [,2]

[1,] -2 1.5

[2,] 1 -0.5

> mat1

[,1] [,2]

[1,] 1 3

tf<-matrix(1:4,2,2)   # creates simple 2x2 matrix  
tf                              # display it  
tf1<-solve(tf)           # calculate the inverse  
tf1                          # display it  
tf \* tf1                   # this does element based multiplication - not what we want  
tf %\*% tf1         # this one does true matrix multiplication and yields the Identity matrix, proving that tf1 is the inverse of tf

amatrix = makeCacheMatrix(matrix(c(1,2,3,4), nrow=2, ncol=2))

amatrix$get() # Returns original matrix

cacheSolve(amatrix) # Computes, caches, and returns matrix inverse

amatrix$getinverse() # Returns matrix inverse

cacheSolve(amatrix) # Returns cached matrix inverse using previously computed matrix inverse

amatrix$set(matrix(c(0,5,99,66), nrow=2, ncol=2)) # Modify existing matrix

cacheSolve(amatrix) # Computes, caches, and returns new matrix inverse

amatrix$get() # Returns matrix

amatrix$getinverse() # Returns matrix inverse

> # Here's another example...

> x <- 5

> changeX <- function(newXValue) x <- newXValue

> changeX(100)

> x

[1] 5

>

> reallyChangeX <- function(newXValue) x <<- newXValue

> reallyChangeX(100)

> x

[1] 100