**St. Xavier’s College (Autonomous), Kolkata**

**Department of Statistics**

**MSc in Data Science**

**Paper code: MDSC 4113**

**Module 1**

**Linear Algebra**

**Handout for Problem sheet 3**

**3.1 Explore the following functions under Matlib**

install.packages('matlib')

library(matlib)

* tr() - trace of a matrix
* R() - rank of a matrix
* Det(A)
* Inverse(X), inv() - uses gaussianElimination to find the inverse of X
* len() - Euclidean length of a vector or columns of a matrix
* vec() - vectorize a matrix
* Proj(y, X) - projection of vector y on columns of X
* mpower(A, p) - matrix powers for a square symmetric matrix
* The minor of an element is equal to the determinant of the matrix remaining after excluding the row and column containing the element:

minor(A,i,j)-Minor of A[i,j]

rowMinors(A,i): gives all the minors corresponding to ith row of A

* The cofactor of an element is equal to the product of the minor of the element, and -1 to the power of the row +column of the element:

cofactor(A,i,j) - Cofactor of A[i,j]

rowCofactors(A,i): gives all the cofactors corresponding to the ith row of A

Let A be a *K×K* dimension matrix, the cofactor expansion along the i-th row is defined with the following formula: Det(A) =

**3.2 Gram-Scmidt Orthogonalization (matlib)**

Carries out simple Gram-Schmidt orthogonalization of a matrix. Treating the columns of the matrix X in the given order, each successive column after the first is made orthogonal to all previous columns by subtracting their projections on the current column.

GramSchmidt(

X,

normalize = [TRUE](https://rdrr.io/r/base/logical.html),

verbose = [FALSE](https://rdrr.io/r/base/logical.html),

tol = [sqrt](https://rdrr.io/r/base/MathFun.html)([.Machine](https://rdrr.io/r/base/zMachine.html)$double.eps)

)

# tol: the tolerance for detecting linear dependencies in the columns of a. The default is .Machine$double.eps

**3.3 Row Reduced Echelon Form**

A matrix that has undergone [Gaussian elimination](https://mathworld.wolfram.com/GaussianElimination.html) is said to be in row echelon form or, more properly, "reduced echelon form" or "row-reduced echelon form."

a). install.packages('pracma')

library(pracma)

# enter matrix A

A <- matrix(1:9, 3, 3)

help(rref)

rref(A)

b). install.packages('matlib')

library(matlib)

# enter matrix A

A <- matrix(1:9, 3, 3)

echelon(A, reduced=T, verbose=TRUE, fractions=T) # row reduced, verbose: shows all the steps, fractions: gives the outputs in fractions