R PROGRAMMING

ROOT FINDING

For real or complex polynomials, one can use the `polyroot()` function. To solve f(x)=x^2−2x−3=0

```
polyroot(c(-3, -2, 1))
```

EIGEN VALUES

```
A<-matrix(c(13, -4, 2, -4, 11, -2, 2, -2, 8), 3, 3, byrow=TRUE)

A

print(eigen(A))
```

LINEAR REGRESSION

- The steps to create the relationship is −
  - Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
  - Create a relationship model using the **lm()** functions in R.
  - Find the coefficients from the model created and create the mathematical equation using these
  - Get a summary of the relationship model to know the average error in prediction. Also called **residuals**.
  - To predict the weight of new persons, use the **predict()** function in R.

LINEAR REGRESSION

- lm() Function
  - This function creates the relationship model between the predictor and the response variable.

- Syntax
  - The basic syntax for **lm()** function in linear regression is −

    lm(formula,data)

- Following is the description of the parameters used −
  - **formula** is a symbol presenting the relation between x and y.
  - **data** is the vector on which the formula will be applied.

LINEAR REGRESSION

- predict() Function
- Syntax
  - The basic syntax for predict() in linear regression is − predict(object,newdata)
- Following is the description of the parameters used −
  - **object** is the formula which is already created using the lm() function.
  - **newdata** is the vector containing the new value for predictor variable.

LINEAR REGRESSION: GET THE COEFFICEINTS

```
x <- c(15, 17, 18, 16, 12, 13, 19, 10, 15, 11)

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)


# Apply the lm() function.

relation<-lm(y~x)


Print(relation)
```

LINEAR REGRESSION:PREDICTING FOR NEW DATA

```
x <- c(15, 17, 18, 16, 12, 13, 19, 10, 15, 11)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
# Apply the lm() function.
relation<-lm(y~x)
print(summary(relation))
a<-data.frame(x = 170)
result<-predict(relation,a)
print(result)
```

SINGULAR VALUES

- Singular values of a M×N matrix A are the square roots of the eigen values of the N×N matrix A(A^T).

SINGULAR VALUE DECOMPOSITION (SVD)

- Handy mathematical technique that has application to many problems
- Given any $m \times n$ matrix **A**, algorithm to find matrices **U**, **V**, and **W** such that

$$\mathbf{A} = \mathbf{U}\,\mathbf{W}\,\mathbf{V}^T$$

U is $m \times n$ and orthonormal

W is $n \times n$ and diagonal

V is $n \times n$ and orthonormal

SVD

- The w_i are called the singular values of $\mathbf{A}$
- If $\mathbf{A}$ is singular, some of the w_i will be 0
- In general $rank(\mathbf{A})$ = number of nonzero w_i
- SVD is mostly unique (up to permutation of singular values, or if some w_i are equal)

SVD

```
A<-matrix(c(13, -4, 2, -4, 11, -2, 2, -2, 8), 3, 3, byrow=TRUE)

(s <- svd(A))

D <- diag(s$d)

print(D)

s$u %*% D %*% t(s$v) #  X = U D V'

t(s$u) %*% A %*% s$v #  D = U' X V
```