Data605-Week1-HomeWork1-kamath

Vinayak Kamath

8/30/2020

Problem set 1

(1) Calculate the dot product u:v where u = [0.5; 0.5] and v = [3;-4]

```
u <- matrix(c(0.5, 0.5), 1, 2)
v <- matrix(c(3, -4), 1, 2)

#print("dot product u:v is ")
(0.5 *  3) + (0.5 * -4)
```

```
## [1] -0.5
```

(2) What are the lengths of u and v? Please note that the mathematical notion of the length of a vector is not the same as a computer science defnition.

```
ulen <- sqrt(0.5**2 + 0.5**2)
#print("lengths of u is ")
ulen
```

```
## [1] 0.7071068
```

```
vlen <- sqrt(3**2 + (-4)**2 )
#print("lengths of v is ")
vlen
```

```
## [1] 5
```

(3) What is the linear combination: 3u - 2v?

```
#print("linear combination: 3u - 2v is ")
3*u - 2*v
```

```
##      [,1] [,2]
## [1,] -4.5  9.5
```

(4) What is the angle between u and v

```
#print("angle between u and v is ")
acos ( ( (0.5 *  3) + (0.5 * -4) )  / ( ulen  *  vlen  ) )
```

```
## [1] 1.712693
```

Problem set 2

(1) Set up a system of equations with 3 variables and 3 constraints and solve for x. Please write a function in R that will take two variables (matrix A & constraint vector b) and solve using elimination. Your function should produce the right answer for the system of equations for any 3-variable, 3-equation system. You don't have to worry about degenerate cases and can safely assume that the function will only be tested with a system of equations that has a solution. Please note that you do have to worry about zero pivots, though. Please note that you should not use the built-in function solve to solve this system or use matrix inverses. The approach that you should employ is to construct an Upper Triangular Matrix and then back-substitute to get the solution. Alternatively, you can augment the matrix A with vector b and jointly apply the Gauss Jordan elimination procedure.Please test it with the system below and it should produce a solution x = [-1.55, -0.32, 0.95]

$$\begin{bmatrix} 1 & 1 & 3 \\ 2 & -1 & 5 \\ -1 & -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix}$$

```r
solve_x = function(A, b){
  r <- dim(A)[1]
  c <- dim(A)[2]+dim(b)[2]

  UT <- matrix(c(A, b), nrow=r, ncol=c)

  for (j in 1:(c-2)) {
    for (i in (j+1):r) {
      UT[i,] <- UT[i,]-UT[j,]*UT[i,j]/UT[j,j]
    }
  }

  UT[r,] <- UT[r,]/UT[r,r]
  xn <- numeric(r)
  xn[r] = UT[r,c]

  for (k in (r-1):1) {
    t = 0
    for (m in (k+1):r) {
      s = UT[k,m]*xn[m]
      t = t + s
    }
    xn[k] = (UT[k,c] - t) / UT[k,k]
  }

  x <- round(xn,2)

return(x)

}

A <- matrix(c(1, 2, -1, 1, -1, -2, 3, 5, 4), nrow=3, ncol=3)
b <- matrix(c(1, 2, 6), nrow=3, ncol=1)
solve_x(A,b)
```

```
## [1] -1.55 -0.32  0.95
```