Week 1: Vectors, Matrices & Systems of Equations – IS602

Sharon Morris

2/2/2017

Problem Set 1 (1) Calculate the dot product u.v where $u = [0.5; 0.5] and v = [3; 4]$

```
u <- c(0.5,0.5)
v <- c(3,-4)
product <- sum(u*v)
product
```

```
## [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 definition.

```
lenghtU <- sqrt(sum(u*u))
lenghtU
```

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

```
lengthV <- sqrt(sum(v*v))
lengthV
```

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

(3) What is the linear combination: $3u2v$?

```
linearComb <- 3*u - 2*v
linearComb
```

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

(4) What is the angle between u and v?

```
theta <- acos( sum(u*v) / ( sqrt(sum(u * u)) * sqrt(sum(v * v)) ) )
theta
```

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

```
#http://stackoverflow.com/questions/1897704/angle-between-two-vectors-in-r
```

Problem set 2 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 $x = -1.55, -0.32, 0.95$

```
x = function(A, b){
  r <- dim(A)[1]
```

```r
  c <- dim(A)[2]+dim(b)[2]
  upperT <- matrix(c(A, b), nrow=r, ncol=c)
  for (j in 1:(c-2)) {
    for (i in (j+1):r) {
      upperT[i,] <- upperT[i,]-upperT[j,]*upperT[i,j]/upperT[j,j]
    }
  }
  upperT[r,] <- upperT[r,]/upperT[r,r]
  n <- numeric(r)
  n[r] = upperT[r,c]
  for (k in (r-1):1) {
    t = 0
    for (m in (k+1):r) {
      s = upperT[k,m]*n[m]
      t = t + s
    }
    n[k] = (upperT[k,c] - t) / upperT[k,k]
  }
  x <- round(n,2)
return(x)
}
```

```r
#Please test it with the system below and it should produce a solution x=[-1.55,-0.32,0.95]

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)
x(A,b)
```

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