Assignment 4

Max Wagner

February 23, 2015

PS1. Storing and displaying A, X, and Y:

```
A <- matrix(c(-1,1,0,1,3,-2,5,0,1), nrow = 3);A
```

```
##      [,1] [,2] [,3]
## [1,]   -1    1    5
## [2,]    1    3    0
## [3,]    0   -2    1
```

```
X <- A %*% t(A);X
```

```
##      [,1] [,2] [,3]
## [1,]   27    2    3
## [2,]    2   10   -6
## [3,]    3   -6    5
```

```
Y <- t(A) %*% A;Y
```

```
##      [,1] [,2] [,3]
## [1,]    2    2   -5
## [2,]    2   14    3
## [3,]   -5    3   26
```

Getting and printing eigenvalues/vectors of X and Y:

```
x.eigen <- eigen(X);x.eigen
```

```
## $values
## [1] 27.4907376 14.0000000  0.5092624
##
## $vectors
##            [,1]       [,2]       [,3]
## [1,] 0.99086430  0.0000000  0.1348627
## [2,] 0.07480836 -0.8320503 -0.5496326
## [3,] 0.11221254  0.5547002 -0.8244489
```

```
y.eigen <- eigen(Y);y.eigen
```

```
## $values
## [1] 27.4907376 14.0000000  0.5092624
##
```

```
## $vectors
##            [,1]       [,2]       [,3]
## [1,] -0.1747144  0.2223748  0.9591790
## [2,]  0.1889822  0.9636241 -0.1889822
## [3,]  0.9663129 -0.1482499  0.2103839
```

Getting the singular, left singular, and right singular values of A using `svd()`:

```
a.svd <- svd(A)
a.svd$d
```

```
## [1] 5.2431610 3.7416574 0.7136263
```

```
a.svd$u
```

```
##             [,1]          [,2]       [,3]
## [1,] -0.99086430 -7.771561e-16  0.1348627
## [2,] -0.07480836  8.320503e-01 -0.5496326
## [3,] -0.11221254 -5.547002e-01 -0.8244489
```

```
a.svd$v
```

```
##            [,1]       [,2]       [,3]
## [1,]  0.1747144  0.2223748 -0.9591790
## [2,] -0.1889822  0.9636241  0.1889822
## [3,] -0.9663129 -0.1482499 -0.2103839
```

We can compare the two sets of singular vectors with the eigenvectors of X and Y. If the comparison works, they should be the same(can be negative).

For X:

```
x.eigen$vectors
```

```
##            [,1]       [,2]       [,3]
## [1,] 0.99086430  0.0000000  0.1348627
## [2,] 0.07480836 -0.8320503 -0.5496326
## [3,] 0.11221254  0.5547002 -0.8244489
```

```
a.svd$u
```

```
##             [,1]          [,2]       [,3]
## [1,] -0.99086430 -7.771561e-16  0.1348627
## [2,] -0.07480836  8.320503e-01 -0.5496326
## [3,] -0.11221254 -5.547002e-01 -0.8244489
```

For Y:

```r
y.eigen$vectors
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.1747144  0.2223748  0.9591790
## [2,]  0.1889822  0.9636241 -0.1889822
## [3,]  0.9663129 -0.1482499  0.2103839
```

```r
a.svd$v
```

```
##            [,1]       [,2]       [,3]
## [1,]  0.1747144  0.2223748 -0.9591790
## [2,] -0.1889822  0.9636241  0.1889822
## [3,] -0.9663129 -0.1482499 -0.2103839
```

And finally for the singular values:

```r
x.eigen$values
```

```
## [1] 27.4907376 14.0000000  0.5092624
```

```r
y.eigen$values
```

```
## [1] 27.4907376 14.0000000  0.5092624
```

```r
a.svd$d ^ 2
```

```
## [1] 27.4907376 14.0000000  0.5092624
```

PS2 First step is to write a function to find the inverse of a matrix.

```r
myinverse <- function(A) {
  # Takes matrix A, and returns the inverse(B).

  if (det(A) != 0) { #checking for 0 determinant
    B <- matrix(data = NA,nrow = dim(A)[1], ncol = dim(A)[2])
    for (i in 1:dim(A)[1]) {
      for (j in 1:dim(A)[2]) {
        B[i,j] <- -1^(i+j) * det(A[-i,-j])
      }
    }
    return(t(B) / det(A))
  }
}
```

Let's test with the matrix from the first problem set.

```r
B <- myinverse(A)
A %*% B
```

```
##            [,1]       [,2]       [,3]
## [1,] -0.8571429 -0.1428571 -0.7142857
## [2,]  0.4285714  0.5714286 -2.1428571
## [3,] -0.2857143  0.2857143  0.4285714
```

The test for matrix A seems a bit off, but the inverse function itself seems to work fine.