Practice: Matrix Algebra

Jay Wei

2023-09-10

Contents

1 Exercise 1

Calculate the inverse of a matrix.

1.1 Question a

```
A <- matrix(data = c(1, 0.5, 0.5, 1.25), nrow = 2, ncol = 2, byrow = TRUE)
A
```

```
##      [,1] [,2]
## [1,]  1.0 0.50
## [2,]  0.5 1.25
```

```
solve(A)
```

```
##       [,1] [,2]
## [1,]  1.25 -0.5
## [2,] -0.50  1.0
```

```
A%*%solve(A)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

1.2 Question b

```
B <- matrix(data = c(1, 0.5, 0.5, 0.26), nrow = 2, ncol = 2, byrow = TRUE)
B
```

```
##      [,1] [,2]
## [1,]  1.0 0.50
## [2,]  0.5 0.26
```

```
solve(B)
```

```
##      [,1] [,2]
## [1,]   26  -50
## [2,]  -50  100
```

```
B%*%solve(B)
```

```
##               [,1] [,2]
## [1,]  1.000000e+00    0
## [2,] -1.776357e-17    1
```

1.3 Question c

```r
C <- matrix(data = c(1, 0.5, 0.5, 0.01), nrow = 2, ncol = 2, byrow = TRUE)
C
```

```
##      [,1] [,2]
## [1,]  1.0 0.50
## [2,]  0.5 0.01
```

```r
solve(C)
```

```
##             [,1]      [,2]
## [1,] -0.04166667  2.083333
## [2,]  2.08333333 -4.166667
```

```r
C%*%solve(C)
```

```
##               [,1] [,2]
## [1,]  1.000000e+00    0
## [2,] -3.509346e-17    1
```

2 Exercise 2

For each matrix in 1), also complete the following:

2.1 Question a. Find the eigenvalues

```r
eigen(A)$values
```

```
## [1] 1.6403882 0.6096118
```

```r
eigen(B)$values
```

```
## [1] 1.252012862 0.007987138
```

```r
eigen(C)$values
```

```
## [1]  1.2085801 -0.1985801
```

2.2 Question b. Find the eigenvectors (just using R) and verify they satisfy their corresponding equality

```r
eigen(A)$vectors
```

```
##           [,1]       [,2]
## [1,] 0.6154122 -0.7882054
## [2,] 0.7882054  0.6154122
```

```r
eigen(B)$vectors
```

```
##            [,1]       [,2]
## [1,] -0.8929846  0.4500872
## [2,] -0.4500872 -0.8929846
```

```r
eigen(C)$vectors
```

```
##            [,1]       [,2]
## [1,] -0.9229151  0.3850035
## [2,] -0.3850035 -0.9229151
```

2.3 Question c. Find the length of the eigenvectors

```r
# matrix A
sqrt(sum(eigen(A)$vectors[,1]^2))
```

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

```r
sqrt(sum(eigen(A)$vectors[,2]^2))
```

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

```r
# matrix B
sqrt(sum(eigen(B)$vectors[,1]^2))
```

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

```r
sqrt(sum(eigen(B)$vectors[,2]^2))
```

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

```r
# matrix C
sqrt(sum(eigen(C)$vectors[,1]^2))
```

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

```r
sqrt(sum(eigen(C)$vectors[,2]^2))
```

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

You shouldn't be surprised. This is R's default of finding eigenvectors.

2.4 Question d. Show that the eigenvectors are perpendicular to each other (use a plot or show that the product of the vectors is 0)

```r
eigen(A)$vectors[,1]%*%eigen(A)$vectors[,2]
```

```
##               [,1]
## [1,] -1.028921e-17
```

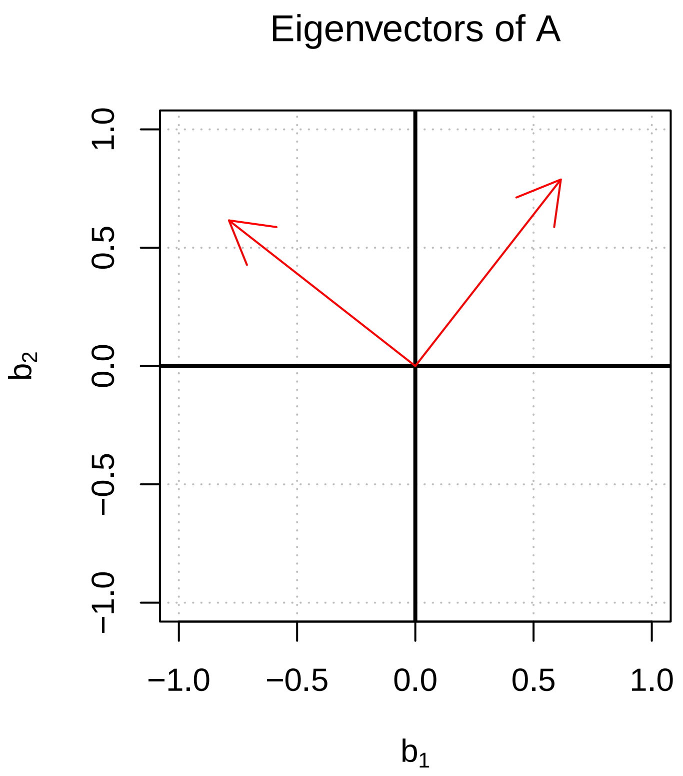
```r
eigen(B)$vectors[,1]%*%eigen(B)$vectors[,2]
```

```
##              [,1]
## [1,] 2.313851e-17
```

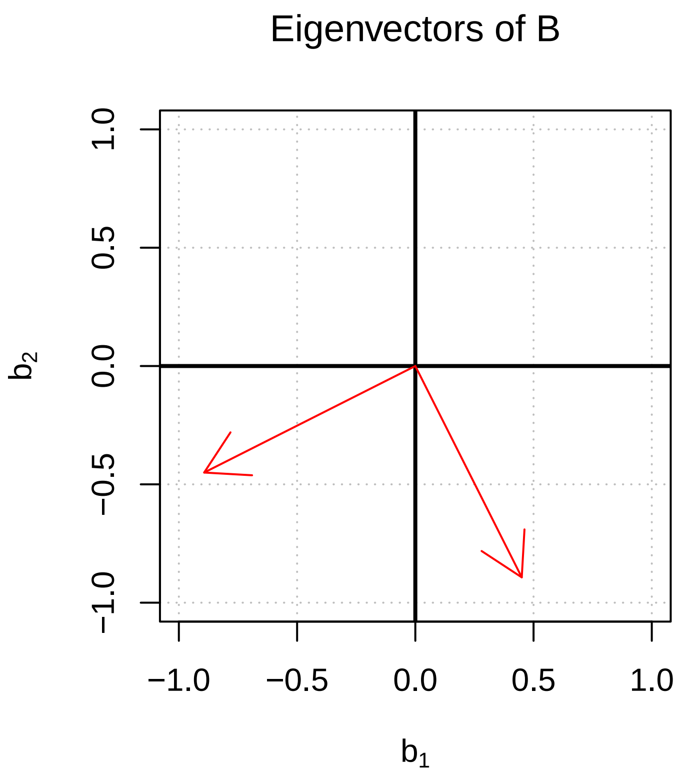
```r
eigen(C)$vectors[,1]%*%eigen(C)$vectors[,2]
```

```
##              [,1]
## [1,] 1.612918e-17
```

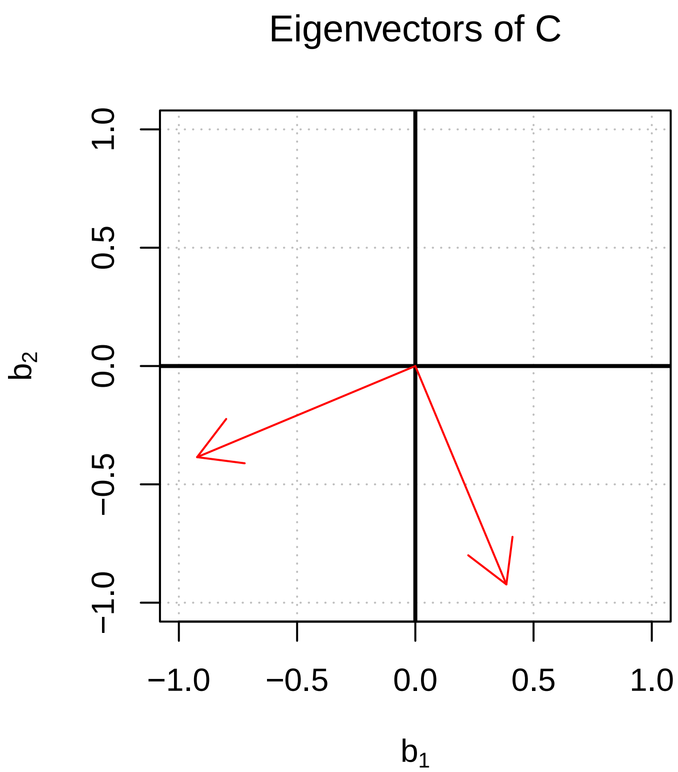
```r
par(pty = "s") # plot type = square

  #Set up some dummy values for plot
  b1 <- c(-1,1)
  b2 <- c(-1,1)

  plot(x = b1, y = b2, type = "n", main =
       expression(paste("Eigenvectors of ", A)),
       xlab = expression(b[1]), ylab = expression(b[2]) ,
       panel.first=grid(col="gray", lty="dotted"))
  # type = "n" don't plot
  #Run demo(plotmath) for help on mathematical notation

  #draw line on plot - h specifies a horizontal line
  abline(h = 0, lty = "solid", lwd = 2)

  #v specifies a vertical line
  abline(v = 0, lty = "solid", lwd = 2)

  arrows(x0 = 0, y0 = 0,
         x1 = eigen(A)$vectors[1,1], y1 = eigen(A)$vectors[2,1],
         col = "red",
         lty = "solid")
  arrows(x0 = 0, y0 = 0,
         x1 = eigen(A)$vectors[1,2], y1 = eigen(A)$vectors[2,2],
         col = "red",
         lty = "solid")
```




```
par(pty = "s") # plot type = square

  #Set up some dummy values for plot
  b1 <- c(-1,1)
  b2 <- c(-1,1)

  plot(x = b1, y = b2, type = "n", main =
       expression(paste("Eigenvectors of ", B)),
       xlab = expression(b[1]), ylab = expression(b[2]) ,
       panel.first=grid(col="gray", lty="dotted"))
  # type = "n" don't plot
  #Run demo(plotmath) for help on mathematical notation

  #draw line on plot - h specifies a horizontal line
  abline(h = 0, lty = "solid", lwd = 2)

  #v specifies a vertical line
  abline(v = 0, lty = "solid", lwd = 2)

  arrows(x0 = 0, y0 = 0,
         x1 = eigen(B)$vectors[1,1], y1 = eigen(B)$vectors[2,1],
         col = "red",
         lty = "solid")
  arrows(x0 = 0, y0 = 0,
         x1 = eigen(B)$vectors[1,2], y1 = eigen(B)$vectors[2,2],
         col = "red",
         lty = "solid")
```

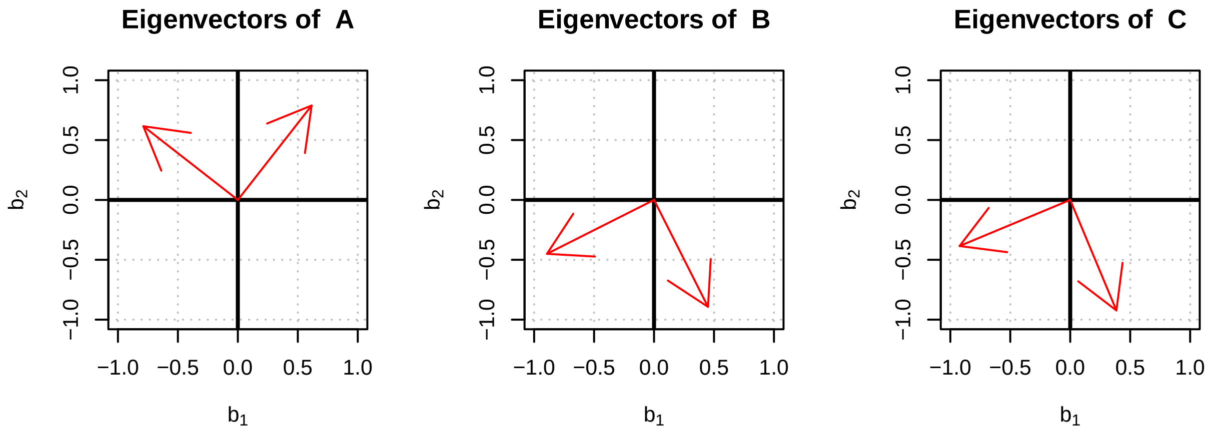



```
par(pty = "s") # plot type = square

  #Set up some dummy values for plot
  b1 <- c(-1,1)
  b2 <- c(-1,1)

  plot(x = b1, y = b2, type = "n", main =
       expression(paste("Eigenvectors of ", C)),
       xlab = expression(b[1]), ylab = expression(b[2]) ,
       panel.first=grid(col="gray", lty="dotted"))
  # type = "n" don't plot
  #Run demo(plotmath) for help on mathematical notation

  #draw line on plot - h specifies a horizontal line
  abline(h = 0, lty = "solid", lwd = 2)

  #v specifies a vertical line
  abline(v = 0, lty = "solid", lwd = 2)

  arrows(x0 = 0, y0 = 0,
         x1 = eigen(C)$vectors[1,1], y1 = eigen(C)$vectors[2,1],
         col = "red",
         lty = "solid")
  arrows(x0 = 0, y0 = 0,
         x1 = eigen(C)$vectors[1,2], y1 = eigen(C)$vectors[2,2],
         col = "red",
         lty = "solid")
```




```
par(pty = "s", mfrow=c(1,3)) # plot type = square
matrix.list <- list("A"=A,"B"=B,"C"=C)
name.list <- c("A","B","C")

#Set up some dummy values for plot
  b1 <- c(-1,1)
  b2 <- c(-1,1)

for (i in 1:length(matrix.list)){

  plot(x = b1, y = b2, type = "n", main =
       paste("Eigenvectors of ", name.list[i]),
       xlab = expression(b[1]), ylab = expression(b[2]) ,
       panel.first=grid(col="gray", lty="dotted"))
  # type = "n" don't plot
  #Run demo(plotmath) for help on mathematical notation

  #draw line on plot - h specifies a horizontal line
  abline(h = 0, lty = "solid", lwd = 2)

  #v specifies a vertical line
  abline(v = 0, lty = "solid", lwd = 2)

  arrows(x0 = 0, y0 = 0,
         x1 = eigen(matrix.list[[i]])$vectors[1,1],
         y1 = eigen(matrix.list[[i]])$vectors[2,1],
         col = "red",
         lty = "solid")
  arrows(x0 = 0, y0 = 0,
```

```r
         x1 = eigen(matrix.list[[i]])$vectors[1,2],
         y1 = eigen(matrix.list[[i]])$vectors[2,2],
         col = "red",
         lty = "solid")
}
```



2.5 Question e. Show the determinant of the matrix is equal to the product of the eigenvalues

```r
det(A)
```

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

```r
eigen(A)$values[1]*eigen(A)$values[2]
```

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

```r
det(B)
```

```
## [1] 0.01
```

```r
eigen(B)$values[1]*eigen(B)$values[2]
```

```
## [1] 0.01
```

```r
det(C)
```

```
## [1] -0.24
```

```r
eigen(C)$values[1]*eigen(C)$values[2]
```

```
## [1] -0.24
```

2.6 Question f. Show the trace of the matrix is equal to the sum of the eigenvalues

```r
sum(diag(A))==sum(eigen(A)$values)
```

```
## [1] TRUE
```

```r
sum(diag(B))==sum(eigen(B)$values)
```

```
## [1] TRUE
```

```r
sum(diag(C))==sum(eigen(C)$values)
```

```
## [1] TRUE
```

2.7 g. Determine if the matrix is positive definite

```r
det(A)>0
```

```
## [1] TRUE
```

```r
det(B)>0
```

```
## [1] TRUE
```

```r
det(C)>0
```

```
## [1] FALSE
```

Matrix A and B are postive definite, but not matrix C.

3 Exercise 3

Continues the diamond data set problem from the previous homework set.

3.1 Question a

$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)' = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} = (-1316.73, 6645.02)'$

```
diamonds <- read.csv("../1-intro-R/DiamondPrices.csv")
head(diamonds)
```

```
##   carat color clarity certify    price
## 1  0.30     D     VS2     GIA 745.9184
## 2  0.30     E     VS1     GIA 865.0820
## 3  0.30     G    VVS1     GIA 865.0820
## 4  0.30     G     VS1     GIA 721.8565
## 5  0.31     D     VS1     GIA 940.1322
## 6  0.31     E     VS1     GIA 890.8626
```

```
X <- cbind(1, diamonds$carat)
Y <- diamonds$price
```

```
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y
beta.hat
```

```
##           [,1]
## [1,] -1316.734
## [2,]  6645.024
```

3.2 Question b

Estimated price for carat = 0.5.

```
X.h <- c(1, 0.5) #Vector containing the carat = 0.5 value
Y.hat <- X.h%*%beta.hat
Y.hat
```

```
##          [,1]
## [1,] 2005.778
```

$\hat{Y}_h = (1 \quad 0.5)\hat{\beta} = \$2005.78$