PC6a - Matrices

Question 1

The following statements generate the desired matrix A. Other solutions are possible. Please, don't use `A[1,1]<-1, A[1,2]<-2` etc.

```
r <- 1:4
A <- matrix(0,4,4)
A[1,] <- r
A[2:4,4] <- r[3:1]
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    0    0    0    3
## [3,]    0    0    0    2
## [4,]    0    0    0    1
```

Question 2

You only need to consider the dimensions of the matrices: A is 3×3, B is 3×2 and C is 2×3.

$[\underset{3\times3}{A} \;\; \underset{3\times2}{B}]$ OK

$[\underset{3\times3}{A} \;\; \underset{2\times3}{B'}]$ not OK

$[\underset{3\times3}{A} \;\; \underset{2\times3}{C}]$ not OK

$[\underset{3\times3}{A} \;\; \underset{3\times2}{C'}]$ OK

$\begin{bmatrix} \underset{3\times3}{A} \\ \underset{2\times3}{C} \end{bmatrix}$ OK

$\begin{bmatrix} \underset{3\times3}{A} \\ \underset{2\times3}{B'} \end{bmatrix}$ OK

```
A <- matrix(1,3,3)
B <- 2*matrix(1,3,2)
C <- 3*matrix(1,2,3)
cbind(A,B)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    2    2
## [2,]    1    1    1    2    2
## [3,]    1    1    1    2    2
```

```
cbind(A,t(B))
```

```
## Error in cbind(A, t(B)): number of rows of matrices must match (see arg 2)
```

```
cbind(A,C)
```

```
## Error in cbind(A, C): number of rows of matrices must match (see arg 2)
```

```
cbind(A,t(C))
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    3    3
## [2,]    1    1    1    3    3
## [3,]    1    1    1    3    3
```

```
rbind(A,C)
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
## [3,]    1    1    1
## [4,]    3    3    3
## [5,]    3    3    3
```

```
rbind(A,t(B))
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
## [3,]    1    1    1
## [4,]    2    2    2
## [5,]    2    2    2
```

Question 3

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 3 & 4 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 \cdot 3 + 2 \cdot -1 = 1 & 1 \cdot 4 + 2 \cdot 2 = 8 \\ 3 \cdot 3 + 4 \cdot -1 = 5 & 3 \cdot 4 + 4 \cdot 2 = 20 \end{pmatrix}$$

```r
A <- matrix(1:4,2,2,byrow=T)
B <- matrix(c(3,-1,4,2),2,2)
A%*%B
```

```
##      [,1] [,2]
## [1,]    1    8
## [2,]    5   20
```

$$2\begin{pmatrix} 3 & 5 \\ 6 & -2 \end{pmatrix} - 4\begin{pmatrix} -1 & 0 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 6 & 10 \\ 12 & -4 \end{pmatrix} - \begin{pmatrix} -4 & 0 \\ 8 & 4 \end{pmatrix} = \begin{pmatrix} 10 & 10 \\ 4 & -8 \end{pmatrix}$$

```r
A <- matrix(c(3,6,5,-2),2,2)
B <- matrix(c(-1,2,0,1),2,2)
2*A-4*B
```

```
##      [,1] [,2]
## [1,]   10   10
## [2,]    4   -8
```

$$\begin{pmatrix} 1 & 3 & 5 \end{pmatrix}\begin{pmatrix} 2 & -1 \\ -1 & 0 \\ 7 & -2 \end{pmatrix} = \left(1\cdot 2 + 3\cdot -1 + 5\cdot 7 = 34, 1\cdot -1 + 3\cdot 0 + 5\cdot -2 = \ -11\right)$$

```
A <- c(1,3,5)
B <- matrix(c(2,-1,7,-1,0,-2),3,2)
A%*%B
```

Question 4

```r
A <- matrix(c(1,1,-2,1,-1,-5,2,-3,1),3,3)
b <- c(1,0,4)
solve(A,b)
```

```
## [1]  0.8 -1.0  0.6
```

So, $(x, y, z) = (4/5, -1, 3/5)$.

Question 5

System (i): the system involves 2 variables and 2 equations. Since $rank(A) = rank(A|b) = dim(A)$, there is a unique solution.

```
A <- matrix(c(3,3,2,-2),2,2)
b <- c(7,7)
qr(A)$rank
```

```
## [1] 2
```

```
qr(cbind(A,b))$rank
```

```
## [1] 2
```

System (ii): the system involves 3 variables and 3 equations. Since $rank(A) < rank(A|b)$, there is no solution.

```
A <- matrix(c(1,1,1,1,1,1,1,-1,0),3,3)
b <- c(1,0,0)
qr(A)$rank
```

```
## [1] 2
```

```
qr(cbind(A,b))$rank
```

```
## [1] 3
```

System (iii): the system involves 6 variables and 6 equations. Since $rank(A) = rank(A|b) = dim(A)$, there is a unique solution.

```
A <- matrix(1,6,6)-2*diag(c(0,1,1,1,1,1))
b <- rep(1,6)
qr(A)$rank
```

```
## [1] 6
```

```
qr(cbind(A,b))$rank
```

```
## [1] 6
```

Question 6a

$$\begin{vmatrix} 0 & 1 & s \\ s & 0 & 1 \\ 1 & s & 0 \end{vmatrix} = 0 \cdot \begin{vmatrix} 0 & 1 \\ s & 0 \end{vmatrix} - 1 \cdot \begin{vmatrix} s & 1 \\ 1 & 0 \end{vmatrix} + s \cdot \begin{vmatrix} s & 0 \\ 1 & s \end{vmatrix} = 0 \cdot -s - 1 \cdot -1 + s \cdot s^2 = s^3 + 1$$

You can verify your answer using Ryacas:

```
library(Ryacas)
yac("A:={{0,1,s},{s,0,1},{1,s,0}}")
```

```
## [1] "{{0,1,s},{s,0,1},{1,s,0}}"
```

```
yac("Determinant(A)")
```

```
## [1] "s^3+1"
```

Question 6b

```
library(pracma)
s <- seq(0,3*pi/4,length=4)
y <- rep(0,4)
for(i in 1:4){
  A <- matrix(c(0,1,s[i],s[i],0,1,1,s[i],0),3,3)
  y[i] <- det(A)
}
coef=polyfit(s,y,3)
cat(formatC(coef,format="f",digits=4))
```

```
## 1.0000 -0.0000 0.0000 1.0000
```

These are the coefficients of $1 \cdot s^3 + 0 \cdot s^2 + 0 \cdot s + 1 = s^3 + 1$.

Question 6c

The matrix is singular for $s = -1$.

Question 7a

```r
A <- diag(4)
A[1,4] <- A[4,1] <- -1
r <- eigen(A)
V <- r$vectors; V
```

```
##            [,1] [,2] [,3]      [,4]
## [1,] -0.7071068    0    0 0.7071068
## [2,]  0.0000000    1    0 0.0000000
## [3,]  0.0000000    0    1 0.0000000
## [4,]  0.7071068    0    0 0.7071068
```

```r
lambda <- r$values; lambda
```

```
## [1] 2.000000e+00 1.000000e+00 1.000000e+00 1.110223e-15
```

The eigenvalues are: 0, 1 (twice) and 2.

Question 7b

```r
for(i in 1:nrow(A)){
  cat(sprintf('A*v_%d  lambda_%d*v_%d:\n',i,i,i))
  AV=A%*%V[,i]
  lV=lambda[i]%*%V[,i]
  cat(sprintf("%8.5f %8.5f\n",AV,lV))
}
```

```
## A*v_1  lambda_1*v_1:
## -1.41421 -1.41421
##   0.00000  0.00000
##   0.00000  0.00000
##   1.41421  1.41421
## A*v_2  lambda_2*v_2:
##  0.00000  0.00000
##   1.00000  1.00000
##   0.00000  0.00000
##   0.00000  0.00000
## A*v_3  lambda_3*v_3:
##  0.00000  0.00000
##   0.00000  0.00000
```

```
##   1.00000  1.00000
##   0.00000  0.00000
## A*v_4  lambda_4*v_4:
##  0.00000  0.00000
##   0.00000  0.00000
##   0.00000  0.00000
##  -0.00000  0.00000
```

Question 8a

```
P <- matrix(c(0.9,0.1,0.5,0.5),2,2)
r <- eigen(P)
V <- r$vectors; V
```

```
##            [,1]       [,2]
## [1,] 0.9805807 -0.7071068
## [2,] 0.1961161  0.7071068
```

```
lambda <- r$values
D <- diag(lambda); D
```

```
##      [,1] [,2]
## [1,]    1  0.0
## [2,]    0  0.4
```

```
V %*% D %*% solve(V)
```

```
##      [,1] [,2]
## [1,]  0.9  0.5
## [2,]  0.1  0.5
```

The matrix D with the eigen values on the diagonal has the following limit:

$$\lim_{n\to\infty} D^n = \lim_{n\to\infty} \begin{bmatrix} 1^n & 0 \\ 0 & 0.4^n \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Question 8b

Using a for-loop to calculate P^{10} using built-in functions only.

```r
PP <- P; for(i in 2:10){ PP <- PP %*% P}
PP
```

```
##           [,1]     [,2]
## [1,] 0.8333508 0.833246
## [2,] 0.1666492 0.166754
```

```r
V %*% diag(lambda^10)%*% solve(V)
```

```
##           [,1]     [,2]
## [1,] 0.8333508 0.833246
## [2,] 0.1666492 0.166754
```

Or using the library `matrixcalc`:

```r
library(matrixcalc)
matrix.power(P,10)
```

```
##           [,1]     [,2]
## [1,] 0.8333508 0.833246
## [2,] 0.1666492 0.166754
```

Question 8c

The steady-state is determined by two equations (1 equilibrium + restriction that fractions sum to 1):

$$\begin{bmatrix} -0.1 & 0.5 \\ 1 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

```r
A <- matrix(c(-0.1,1,0.5,1),2,2)
b <- c(0,1)
v <- solve(A,b)
as.matrix(v)
```

```
##           [,1]
## [1,] 0.8333333
## [2,] 0.1666667
```

```r
P %*% v
```

```
##           [,1]
## [1,] 0.8333333
## [2,] 0.1666667
```

Hence, Pv equals v.