MAS3114 MATLAB Assignment 4

Name: Samantha Bennett

UF ID: ####-####

WARNING: You will recieve a zero on this assignment if you fail to enter your name, UF ID, and the seed in the random number generator.

```
clear
% Initialize the random number generator by typing the command
rng(########,'twister')
% where ######## must be your 8-digit UF ID number
```

Exercise 1 -- Eigenvalues & Eigenvectors

1A

```
B = randi([-8,8],4,4)
```

```
B = 4×4
     6     5     3     2
    -4     2     1     5
     2     3     1    -3
    -4    -1     4    -1
```

```
% The command tril returns a triangular matrix of the lower triangle part
% of matrix B
A1 = tril(B)
```

```
A1 = 4×4
     6     0     0     0
    -4     2     0     0
     2     3     1     0
    -4    -1     4    -1
```

The eigenvalues of A1 are 6, 2, 1, and -1 beacuse eigenvalues of a triangular matrix are the entries on the main diagonal.

% Find eigenvalues and eigenvectors using eig:

```
% P is an invertible matrix
% D is a diagonal matrix
[P,D] = eig(A1)
```

```
P = 4×4
         0         0         0    0.6545
         0         0    0.2065   -0.6545
         0    0.4472    0.6196   -0.1309
    1.0000    0.8944    0.7573   -0.3553
D = 4×4
    -1     0     0     0
     0     1     0     0
     0     0     2     0
     0     0     0     6
```

```
% check if A1 = P*D*inv(P), Yes
P*D*inv(P)
```

```
ans = 4×4
    6.0000         0         0         0
   -4.0000    2.0000         0         0
    2.0000    3.0000    1.0000         0
   -4.0000   -1.0000    4.0000   -1.0000
```

```
% Find eigenvalues and eigenvectors using eigvec:
[P,D] = eigvec(A1)
```

```
P = 4×4
   -1.8421         0         0         0
    1.8421    0.2727         0         0
    0.3684    0.8182    0.5000         0
    1.0000    1.0000    1.0000    1.0000
D = 4×4
     6     0     0     0
     0     2     0     0
     0     0     1     0
     0     0     0    -1
```

```
% check if A1 = P*D*inv(P)?, Yes
P*D*inv(P)
```

```
ans = 4×4
    6.0000         0         0         0
   -4.0000    2.0000         0         0
    2.0000    3.0000    1.0000         0
   -4.0000   -1.0000    4.0000   -1.0000
```

Conclusion: A1 is diagonalizable since $A_1 = PDP^{-1}$.

Note that P and D are not unique. Decompose A1 using two different sets of P and D.

A1 = $\begin{bmatrix} 0 & 0 & 0 & 0.6545 \\ 0 & 0 & 0.2065 & -0.6545 \\ 0 & 0.4472 & 0.6196 & -0.1309 \\ 1 & 0.8944 & 0.7573 & -0.3553 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}$*inv(P)

A1 = $\begin{bmatrix} -1.8421 & 0 & 0 & 0 \\ 1.8421 & 0.2727 & 0 & 0 \\ 0.3684 & 0.8182 & 0.5 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$*inv(P)

1B

```
A2 = [5 0 0 0; 1 5 0 0; 0 1 5 0; 0 0 1 5]
```

```
A2 = 4×4
     5     0     0     0
     1     5     0     0
     0     1     5     0
```

```
     0     0     1     5
```

```
[P,D] = eig(A2)
```

```
P = 4×4
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    1.0000   -1.0000    1.0000   -1.0000
D = 4×4
     5     0     0     0
     0     5     0     0
     0     0     5     0
     0     0     0     5
```

```
A2*P
```

```
ans = 4×4
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    5.0000   -5.0000    5.0000   -5.0000
```

```
P*D
```

```
ans = 4×4
         0         0         0    0.0000
         0         0    0.0000   -0.0000
         0    0.0000   -0.0000    0.0000
    5.0000   -5.0000    5.0000   -5.0000
```

```
P*D*inv(P)
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =  6.842278e-46.
ans = 4×4
     5     0     0     0
     0     5     0     0
     0     0     5     0
     0     0     0     5
```

```
% check if A2*P = P*D, Yes
% check if A2 = P*D*inv(P), No
% write a single command for a basis of the eigenspace of A2 corresponding
% to lambda = 5 using NulBasis
N = NulBasis(A2 - 5*eye(4))
```

```
N = 4×1
     0
     0
     0
     1
```

Conclusion: A2 is NOT diagonalizable because

- P*D*inv(P) does not equal A2. Therefore, the columns in P are not linearly independent eigenvectors of A2. A2 can only be diagonalizable if and only if A2 have n linearly indepdent eigenvectors.
- the dimension of eigenspace (dimension = 1) does not equal the multiplicity of lambda = 5 (multiplicity = 4)

1C

```
A = [6 3; 3 6]
```

```
A = 2×2
     6     3
     3     6
```

```
[P,D] = eigvec(A)
```

```
P = 2×2
     1    -1
     1     1
D = 2×2
     9     0
     0     3
```

```
dot(P(:, 1),P(:, 2))
```

```
ans = 0
```

Let v1 and v2 be two linearly indepedent eigenvectors of A. v1 and v2 are orthogonal because their dot product is zero.

The solution to the system $\begin{array}{l} x_1' = 6x_1 + 3x_2 \\ x_2' = 3x_1 + 6x_2 \end{array}$ is �.

1D

```
A = [4 0 2; 2 3 4; 0 0 -2]
```

```
A = 3×3
     4     0     2
     2     3     4
     0     0    -2
```

```
[P,D] = eigvec(A)
```

```
P = 3×3
    0.5000         0   -0.3333
    1.0000    1.0000   -0.6667
         0         0    1.0000
D = 3×3
     4     0     0
     0     3     0
     0     0    -2
```

```
P*D*inv(P)
```

```
ans = 3×3
     4     0     2
     2     3     4
     0     0    -2
```

A is diagonalizable because $A = \mathrm{PDP}^{-1}$. Therefore, the columns of P are linearly independent eigenvectors of A and diagonal entries of D are the eigenvalues of A that correspond to eigenvectors of P. So A has all linearly independent eigenvectors which means A is diagonalizable.

The solution to the system $\begin{aligned} x_1' &= 4x_1 + 2x_3 \\ x_2' &= 2x_1 + 3x_2 + 4x_3 \\ x_3' &= -2x_3 \end{aligned}$ is �.

Exercise 2 -- The Markov Chain

2A

```
A = [.7 .20 .10; .20 .70 .20; .10 .10 .70]
```

```
A = 3×3
    0.7000    0.2000    0.1000
    0.2000    0.7000    0.2000
    0.1000    0.1000    0.7000
```

```
x = [.70; .15; .15]  %x0 (probability vector for two years ago)
```

```
x = 3×1
    0.7000
    0.1500
    0.1500
```

```
x = A*x  %x1 (probability vector for right now)
```

```
x = 3×1
    0.5350
    0.2750
    0.1900
```

```
x = A*x  %x2 (probability vector for two years later)
```

```
x = 3×1
    0.4485
    0.3375
    0.2140
```

Now, 53.5% of those surveyed drive cars, 27.5% minivans, and 19% suv.

Two years later, 44.85% of those surveyed will drive cars, 33.75% minivans, and 21.4% suv.

2B see Sol_DiffEq.m

2C

```
% Method 1
x = [.70; .15; .15];  %x0 (initial probability vector)
Sol_DiffEq(A,x)
```

```
ans = 3×1
    0.3500
    0.4000
    0.2500
```

```
%%%%%
% Method 2
```

```
[P,D] = eigvec(A)
```

```
P = 3×3
    1.4000   -1.0000   -1.0000
    1.6000    0.0000    1.0000
    1.0000    1.0000         0
D = 3×3
    1.0000         0         0
         0    0.6000         0
         0         0    0.5000
```

```
x = [.70; .15; .15];  %x0 (initial probability vector)
% write a single command to solve for Pc = x0, where P = [v1 v2 v3], v_i is
% an eigenvector of A
c = rref([P x])
```

```
c = 3×4
    1.0000         0         0    0.2500
         0    1.0000         0   -0.1000
         0         0    1.0000   -0.2500
```

```
c(:, end)
```

```
ans = 3×1
    0.2500
   -0.1000
   -0.2500
```

Express x0 as a linear combination of eigenvectors of A:

$$\mathbf{x}_0 = 0.25\begin{bmatrix}1.4\\1.6\\1\end{bmatrix} + -0.1\begin{bmatrix}-1\\0\\1\end{bmatrix} + -0.25\begin{bmatrix}-1\\1\\0\end{bmatrix}$$

Therefore, a general solution to the difference equation is $\mathbf{x}_k =$

$$0.25(1)^k\begin{bmatrix}1.4\\1.6\\1\end{bmatrix} + (-0.1)(0.6)^k\begin{bmatrix}-1\\0\\1\end{bmatrix} + (-0.25)(0.5)^k\begin{bmatrix}-1\\1\\0\end{bmatrix}.$$

Finally, �

Do you have the same solution from both methods? Yes. As k approaches infinity, the second and third eigenvectors in the linear combination approach 0. Therefore, only the first eigenvector is left and

$$0.25\begin{bmatrix}1.4\\1.6\\1\end{bmatrix} = \begin{bmatrix}0.35\\0.4\\0.25\end{bmatrix}.$$

Conclusion: In the long run, we expect 35% of those surveyed will drive cars, 40% minivans, and 25% suv.

The steady-state vector for matrix A is $\begin{bmatrix}0.35\\0.4\\0.25\end{bmatrix}$.

2D

```
x = [.60; .20; .20]; %x0 (enter entries for a different initial probability vector)
Sol_DiffEq(A,x)
```

```
ans = 3×1
    0.3500
    0.4000
    0.2500
```

```
x = [.30; .40; .30]; %x0 (enter entries for another initial probability vector)
Sol_DiffEq(A,x)
```

```
ans = 3×1
    0.3500
    0.4000
    0.2500
```

The steady-state vector is $\begin{bmatrix} 0.35 \\ 0.4 \\ 0.25 \end{bmatrix}$ for any initial vector x0 because every stochastic matrix A has a unique probability vector called p that satisfies the equation $\mathrm{Ap} = p$. This probability vector is called the steady-state vector.

Exercise 3 -- Inner Products & Orthogonal Projections

3A

```
u1 = [2 1 3 -2].'
```

```
u1 = 4×1
     2
     1
     3
    -2
```

```
u2 = [1 2 3 4].'
```

```
u2 = 4×1
     1
     2
     3
     4
```

```
v1 = [1 -2 3 -4].'
```

```
v1 = 4×1
     1
    -2
     3
    -4
```

```
v2 = [2 4 6 8].'
```

```
v2 = 4×1
     2
     4
     6
```

```
     8
```

```
v3 = [-3 -6 -9 -12].'
```

```
v3 = 4×1
    -3
    -6
    -9
   -12
```

```
%do the calcuations for dot product and norms
dot(u1, v1)
```

```
ans = 17
```

```
norm(u1)
```

```
ans = 4.2426
```

```
norm(v1)
```

```
ans = 5.4772
```

```
%Verify the Cauchy-Schwarz inequality for u1 and v1
abs(dot(u1,v1))
```

```
ans = 17
```

```
norm(u1)*norm(v1)
```

```
ans = 23.2379
```

```
%repeat the process for u2, v2, and u2, v3
abs(dot(u2,v2))
```

```
ans = 60
```

```
norm(u2)*norm(v2)
```

```
ans = 60
```

```
abs(dot(u2,v3))
```

```
ans = 90
```

```
norm(u2)*norm(v3)
```

```
ans = 90
```

The Cauchy-Schwarz inequality:

- For u1 and v1: $|\mathbf{u_1} \cdot \mathbf{v}_1| \;<\; \|\mathbf{u}_1\|\;\|\mathbf{v}_1\|$
- For u2 and v2: $|\mathbf{u_2} \cdot \mathbf{v}_2| \;<\; \|\mathbf{u}_2\|\;\|\mathbf{v}_2\|$
- For u2 and v3: $|\mathbf{u_2} \cdot \mathbf{v}_3| \;=\; \|\mathbf{u}_2\|\;\|\mathbf{v}_3\|$
- "=" when u and v are parallel.

3B see projection.m

3C

```
[y_hat,z] = projection(v1,u1)
```

```
y_hat = 4×1
    1.8889
    0.9444
    2.8333
   -1.8889
z = 4×1
   -0.8889
   -2.9444
    0.1667
   -2.1111
```

```
y_hat+z
```

```
ans = 4×1
     1
    -2
     3
    -4
```

```
dot(z,u1) %rounded to zero
```

```
ans = 8.8818e-16
```

Verify if v1 = y_hat + z and if z is orthogonal to u1, it is

```
[y_hat,z] = projection(v2,u2)
```

```
y_hat = 4×1
     2
     4
     6
     8
z = 4×1
     0
     0
     0
     0
```

```
y_hat+z
```

```
ans = 4×1
     2
     4
     6
     8
```

```
dot(z,u2)
```

```
ans = 0
```

Verify if v2 = y_hat + z and if z is orthogonal to u2, it is

z is the zero vector because y is parallel to y_hat. Since z is the distance between y and y_hate and z = 0, there is no distance between y and y_hat. Therefore, they are parallel.