COMP4107 - Assignment 1

Student Name: Yunkai Wang
Student Number: 100968473

Student Name: Jules Kuehn
Student Number: 100661464

Fall 2018

1. Question 1

2. Question 2
   Implementation for question 2 can be found in q2.py. The results is

```
('A=', array([[1, 2, 3],
       [2, 3, 4],
       [4, 5, 6],
       [1, 1, 1]]))
('U=', array([[-0.33306893, -0.73220483,  0.20999988, -0.55573485],
       [-0.48640367, -0.34110504,  0.13689238,  0.79266594],
       [-0.79307315,  0.44109455, -0.34689227, -0.23693109],
       [-0.15333474,  0.39109979,  0.90378442, -0.08187267]]))
('S=', array([[  1.10528306e+01,   0.00000000e+00,   0.00000000e+00],
       [  0.00000000e+00,   9.13748280e-01,   0.00000000e+00],
       [  0.00000000e+00,   0.00000000e+00,   1.10715576e-16]]))
('V=', array([[-0.41903326, -0.56492763, -0.71082199],
       [ 0.81101447,  0.11912225, -0.57276996],
       [ 0.40824829, -0.81649658,  0.40824829]]))
```

3. Question 3
   Implementation for question 3 can be found in q3.py. The rank-2 approximation and $||A-A_2||$ is

```
yunkaideMacBook-Pro:yunkai jeremy$ python3 q3.py
A2= [[0.17447807 0.17754332 0.18056607 ... 0.18056607 0.17754332 0.1744
 [0.17754332 0.18059153 0.18359756 ... 0.18359756 0.18059153 0.17754332
 [0.18056607 0.18359756 0.18658718 ... 0.18658718 0.18359756 0.18056607
 ...
 [0.18056607 0.18359756 0.18658718 ... 0.18658718 0.18359756 0.18056607
 [0.17754332 0.18059153 0.18359756 ... 0.18359756 0.18059153 0.17754332
 [0.17447807 0.17754332 0.18056607 ... 0.18056607 0.17754332 0.17447807

||A - A2|| = 1.3311896328587232
```

4. Question 4

   Implementation for question 4 can be found in q4.py. The only learning rate that will work is when $\varepsilon = 0.01$, which will lead to the correct result with ≈ 420 iterations. The other ones won't work as we are descending too quickly, and therefore we will miss the correct answer and failed to come back. We set the program to stop once the norm of the vector is greater than 100000, which is obviously too big and therefore it's impossible to get the result.

```
 P-inv sol'n:    -0.147059     0.058824     0.264706
|    Step     | RESULT |                  x                   |  Iter  |
============================================================
 step: 0.010:  success    -0.156771     0.057397     0.271565    420
 step: 0.050:  failed  -5133.149689 -6919.777001 -8706.404313    5
 step: 0.100:  failed  23356.638100 31489.229900 39621.821700    4
 step: 0.150:  failed  -7673.780375 -10345.031125 -13016.281875    3
 step: 0.200:  failed  -18987.992000 -25598.536000 -32209.080000    3
 step: 0.250:  failed  -38043.265625 -51288.296875 -64533.328125    3
 step: 0.500:  failed  -320060.375000 -431495.125000 -542929.875000    3
dhcp-93-141:yunkai jeremy$
```

5. Question 5

   The implementation for q5 can be found in q5.py. $A = \begin{bmatrix} 3 & 2 & -1 & 4 \\ 1 & 0 & 2 & 3 \\ -2 & -2 & 3 & 3 \end{bmatrix}$. It's clear that the rows of A are not linearly independent. To see this, we reduce A to its RREF form and we will get $A = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & -7/2 & -5/2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, rank of rows of A equals to 2. The columns of A are not linearly independent as well. Since as we reduce A^T to RREF form, we will get $A = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, clearly that rank of the columns of A equal to 2, so the columns are not linearly independent. The inverse of A cannot be calculated as A is not a square matrix, and the rows and columns of A are not linearly independent.

```
A in rref:
 Matrix([[1, 0, 2, 3], [0, 1, -7/2, -5/2], [0, 0, 0, 0]])
Independent rows: 2

A^T in rref:
 Matrix([[1, 0, -1], [0, 1, 1], [0, 0, 0], [0, 0, 0]])
Independent cols: 2
```

6. Question 6

7. Question 7