zl9901 Homework 1

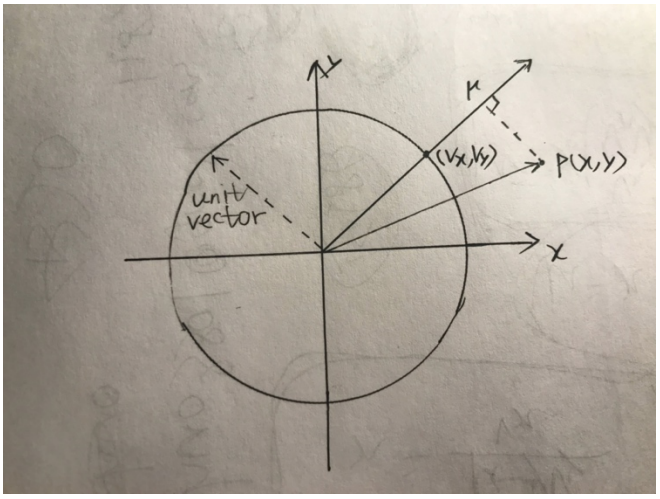
1 1) $a^Tb = 1*3 - 4*1 + 5*9 = 44$

2) $ab^T = \begin{pmatrix} 3 & -1 & 9 \\ 12 & -4 & 36 \\ 15 & -5 & 45 \end{pmatrix}$

3) $(ab^T)c = \begin{pmatrix} 3 & -1 & 9 \\ 12 & -4 & 36 \\ 15 & -5 & 45 \end{pmatrix}\begin{pmatrix} -4 \\ -1 \\ 3 \end{pmatrix} = \begin{pmatrix} 16 \\ 64 \\ 80 \end{pmatrix}$

4) $c^T(ab^T) = \begin{pmatrix} -4 & -1 & 3 \end{pmatrix}\begin{pmatrix} 3 & -1 & 9 \\ 12 & -4 & 36 \\ 15 & -5 & 45 \end{pmatrix} = \begin{pmatrix} 21 & -7 & 63 \end{pmatrix}$

2 a



b According to the formula, $\alpha = x^T u$

where α is the projection, x is an arbitrary point, u is the unit vector.

$$\mu = \begin{pmatrix} x & y \end{pmatrix}\begin{pmatrix} v_x \\ v_y \end{pmatrix} = x * v_x + y * v_y$$

$$x = \frac{\mu - yv_y}{v_x}$$

3 b

Table 1 statistics

	Experiment1	Experiment2	Experiment3	Experiment4
number	10000	7554	1448	2617
Standard deviation	0.0	0.4299	0.3519	0.4396
mean	1.0	0.7554	0.1448	0.2617
confidence	1.0±0.0	0.7554±0.0070	0.1448±0.0058	0.2617±0.0072

Since the confidence level is 90%, z* should be 1.645

For large samples, we can approximate the binomial using a normal distribution. This is justified by the Central Limit Theorem.

For experiment 1, 90% confidence interval should be calculated as followed:

$$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = 1.0 \pm 1.645 \frac{0.0}{\sqrt{10000}} = 1.0 \pm 0.0$$

For experiment 2, 90% confidence interval should be calculated as followed:

$$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = 0.7554 \pm 1.645 \frac{0.4299}{\sqrt{10000}} = 0.7554 \pm 0.0070$$

For experiment 3, 90% confidence interval should be calculated as followed:

$$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = 0.1448 \pm 1.645 \frac{0.3519}{\sqrt{10000}} = 0.1448 \pm 0.0058$$

For experiment 4, 90% confidence interval should be calculated as followed:

$$\bar{x} \pm z^* \frac{\sigma}{\sqrt{n}} = 0.2617 \pm 1.645 \frac{0.4396}{\sqrt{10000}} = 0.2617 \pm 0.0072$$

Table 2 time consuming of different matrix

	Experiment1	Experiment2	Experiment3	Experiment4
10*10	0.1316	0.3437	0.5233	1.0719
20*20	0.2235	0.4459	1.6266	3.7843
30*30	0.3330	0.6267	3.4465	8.0948

We can see from the table above, with the increase of array size, the time of invertible operation becomes much longer. I choose to import time module to calculate the time. There is a shortcoming in my solution. In experiment 3 and experiment 4, I traversed each element of the matrix to make the value of each entry as 0 or 1. This will require extra time and generate differences from first two experiments. As a result, the result might not be so accurate.

Table 3 number of invertible matrices of different size

	Experiment1	Experiment2	Experiment3	Experiment4
10*10	10000	7554	1448	2617
20*20	10000	10000	8551	8789
30*30	10000	10000	9869	9913

We can see from the table above, with the increase of array size, the number of invertible matrices becomes larger. The shortcoming of this approach is space complexity. We can only calculate certain size of matrices. If the size of matrix becomes more than 100*100, the computational time will be longer.

When the size of samples is too large, we can use normal distribution to approximate binomial distribution. I have noticed that the mean value of this Gaussian distribution moves right along x axis as the size of matrix increases which x axis represents the number of matrices that can be inverted. Also, with the increasing size of matrix, the possibility that one row or one column will be much lower.