1. (a) $\vec{x} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$ $\vec{y} = \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}$

$$x^T y = \begin{bmatrix} 2 & 2 & 3 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \end{bmatrix}$$

(b) $\|x\|_1 = 2+2+3 = 7$

$\|y\|_1 = 1+0+2 = 3$

(c) $\|x\|_2 = \sqrt{4+4+9} = \sqrt{17}$

$\|y\|_2 = \sqrt{1+0+4} = \sqrt{5}$

(d) $\cos\theta = \frac{\vec{u}\cdot\vec{v}}{\|u\|\cdot\|v\|} = \frac{\vec{x}\cdot\vec{y}}{\|x\|_2 \cdot \|y\|_2}$

$$= \frac{4}{\sqrt{17}\cdot\sqrt{5}} = \frac{4}{\sqrt{85}}$$

2. ~~P(def | A)~~ =

$$P(A \mid def) = \frac{P(A \cap def)}{P(def)}$$

$P(def) = 25\% \cdot 5\% + 35\% \cdot 4\% + 40\% \cdot 2\%$

$= 0.0345$

$P(A \mid def) = \frac{0.25 \cdot 0.05}{0.0345} = 36.2\%$

$P(B \mid def) = \frac{0.35 \cdot 0.04}{0.0345} = 40.6\%$

$P(C \mid def) = \frac{0.40 \cdot 0.02}{0.0345} = 23.2\%$

3. (a) $$\mathbb{E}[X+Y] = \sum_x \sum_y (x+y) \cdot P_{XY}(x,y) = \sum_x \sum_y x \cdot P_{XY}(x,y) + \sum_x \sum_y y \cdot P_{XY}(x,y)$$

$$= \sum_x \left(x \cdot \sum_y P_{XY}(x,y) \right) + \sum_y \left(y \cdot \sum_x P_{XY}(x,y) \right)$$

$$= \sum_x x \cdot P_x(x) + \sum_y y\, P_y(y) = \mathbb{E}[x] + \mathbb{E}[y]$$

(b) if independent: $\mathrm{COV}[X,Y] = E[(X-\mu_X)(Y-\mu_Y)] = E[X-\mu_X]\, E[Y-\mu_Y]$

$= (E[X]-\mu_X)(E[Y]-\mu_Y) = 0.$

$\mathrm{VAR}[X+Y] = E[(X+Y - E[X+Y])^2]$

$= E[(X-\mu_X)^2] + E[(Y-\mu_Y)^2]$

$+ E[2(X-\mu_X)(Y-\mu_Y)] \rightarrow 0 = \mathrm{VAR}[X] + \mathrm{VAR}[Y]$

4. (a)

Let $A = \begin{bmatrix} A_{11} & A_{21} & & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ A_{13} & \vdots & & \vdots \\ \vdots & & & \end{bmatrix}$

$= \begin{bmatrix} \vec{A}_1 & \vec{A}_2 & \cdots & \vec{A}_n \end{bmatrix}$

where A_i is a column vector...

$E[Ax+b] = E[Ax] + b$

$= E[A_1 \cdot x_1 + A_2 \cdot x_2 + A_3 \cdot x_3 \cdots A_n \cdot x_n] + b$

$= A_1 \cdot E[x_1] + A_2 \cdot E[x_2] \ldots + b$

$= A \cdot E[x] + b$

(b) $E[(x-E[x])(x-E[x])^T]$

$= E[(Ax + \cancel{b} - A \cdot E[x] - \cancel{b})(\cancel{x} A\cancel{x} - A \cdot E[x] - b)^T]$

$= E[(Ax - A \cdot E[x])(Ax - A \cdot E[x])^T]$

$= E[A(x - E[x]) \cdot (x - E[x])^T \cdot A^T]$

A and x are ~~un~~ independent.

We prove that $E[A \cdot B] = E[A] \cdot E[B]$

$E[A \cdot B] = E\left(\begin{bmatrix} y_{00} & \cdots \\ y_{01} & \\ y_{02} & \\ \vdots & \end{bmatrix}\right)$ where $y_{ij} = \sum_k A_{ik} B_{k,j}$

~~E[A]~~. $= \left[y_{ij} = E\left[\sum_k A_{ik} B_{kj}\right] \right]$

$E[A] = \begin{bmatrix} E[A_{00}] & \cdots \\ E[A_{01}] & \\ \vdots & \end{bmatrix}$

$E[B] = \begin{bmatrix} E[B_{00}] & \cdots \\ E[B_{01}] & \\ \vdots & \end{bmatrix}$

$E[A] \cdot E[B] = [z_{ij}]$

when

$z_{ij} = \sum_k E[A_{ik}] \cdot E[B_{kj}]$

↓ b/c indep.

$= \sum_k E[A_{ik} \cdot B_{kj}]$

therefore $E[A \cdot B] = E[A] \cdot E[B]$ holds for matrix mult. if A, B indep.

$E[A(x - E[x]) \cdot (x - E[x])^T \cdot A^T)$

$= E[A] \cdot E[(x - E[x])(x - E[x])^T] \cdot E[A^T]$

$= \boxed{A \cdot \mathrm{cov}(x) \cdot A^T}$ ∎

(c) y is a multivariate gaussian distrib. b/c $Ax+b$ is a affine transformation

y has mean $A \cdot E[x] + b$

& Covariance $A \, \mathrm{cov}(x) \cdot A^T$.

Part (a): red represents students who get in and blue represents students who get rejected

```
stat30 = readtable("UCLA_EE_grad_2030.csv");
figure(1);
GPA = stat30{:,1};
GRE = stat30{:,2};
in = stat30{:,3};
hold on;
scatter(GPA(in == -1), GRE(in == -1), 'blue');
scatter(GPA(in == 1), GRE(in == 1), 'red');
xlabel('GPA for 2030');
ylabel('GRE for 2030');
title('Normalized GRE score vs GPA of applicants in 2030');


stat31 = readtable("UCLA_EE_grad_2031.csv");
figure(2);
GPA = stat31{:,1};
GRE = stat31{:,2};
in = stat31{:,3};
hold on;
scatter(GPA(in == -1), GRE(in == -1), 'blue');
scatter(GPA(in == 1), GRE(in == 1), 'red');
xlabel('GPA for 2031');
ylabel('GRE for 2031');
title('Normalized GRE score vs GPA of applicants in 2031');
```

Looking at these two graphs, it seems like 2031's data is linearly separable, while 2030's is not.

(b)

```
figure(1);
D = 2;
u = 0;
maxiter = 1000;
wd1 = zeros(1, D);
b1 = 0;
stat30 = table2array(stat30);
for iter = 1:maxiter
    for row = 1:length(stat30)
        a = wd1 * stat30(row,1:2)' + b1;
        if stat30(row,3) * a <= 0
            u = u + 1;
            wd1 = wd1 + stat30(row,3) * stat30(row,1:2);
            b1 = b1 + stat30(row,3);
        end
    end
end
xval = (0:0.01:5);
yval = (-b1 - wd1(1) * xval) / wd1(2);
plot(xval, yval);
```


```
b1
```

```
b1 = -61
```

```
wd1
```

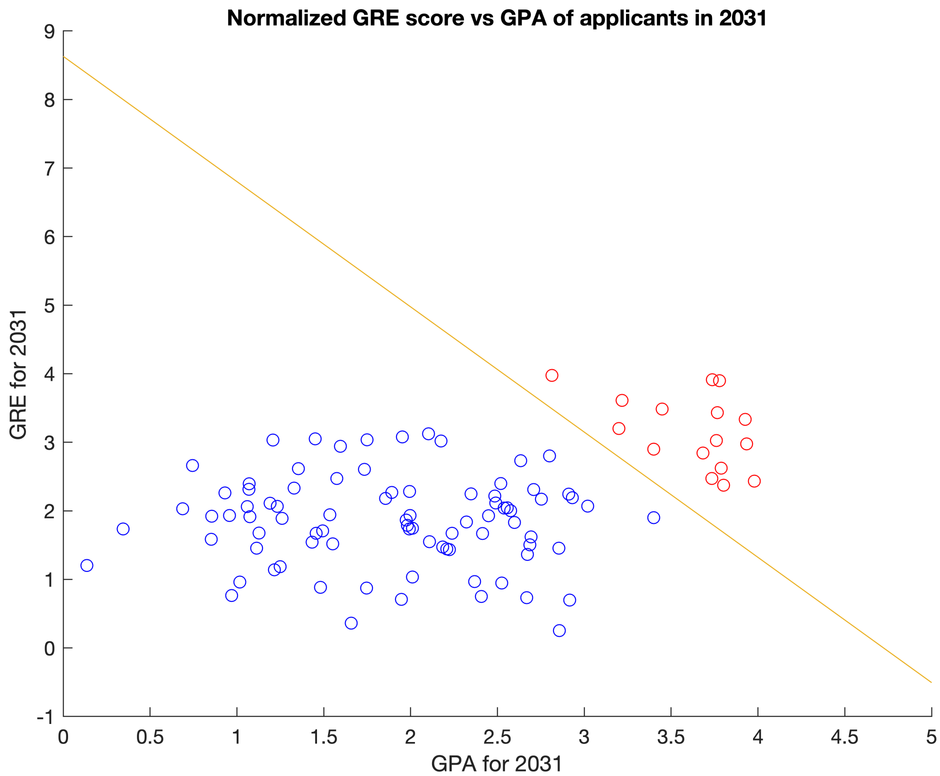
```
wd1 = 1×2
   14.3003    1.0938
```

```
u
```

```
u = 11299
```

```
figure(2);
D = 2;
u = 0;
maxiter = 1000;
wd2 = zeros(1, D);
b2 = 0;
stat31 = table2array(stat31);
for iter = 1:maxiter
    for row = 1:length(stat31)
        a = wd2 * stat31(row,1:2)' + b2;
        if stat31(row,3) * a <= 0
            u = u + 1;
            wd2 = wd2 + stat31(row,3) * stat31(row,1:2);
            b2 = b2 + stat31(row,3);
        end
```

```
    end
end
xval = (0:0.01:5);
yval = (-b2 - wd2(1) * xval) / wd2(2);
plot(xval, yval);
```




```
b2
```

```
b2 = -28
```

```
wd2
```

```
wd2 = 1x2
    5.9282    3.2447
```

```
u
```

```
u = 158
```

We see that for the 2031 dataset, the perceptron algorithm converges quickly with only 158 iterations, while the 2030 dataset has over 11000 iterations. This is because there IS no clearly defined line separating the two groups of data, so the perceptron algorithm actually doesn't work here. It doesn't converge and will never (if we didn't specify max 1000 loops).

(c)

```
gamma = inf;
for row = 1:length(stat31)
    x = stat31(row,1);
```

```
    y = stat31(row,2);
    gamma = min(gamma,abs(x*wd2(1)+y*wd2(2)+b2)/sqrt(wd2(1)*wd2(1)+wd2(2)*wd2(2)));
end
gamma
```

```
dist = 0.2002
```

Here, we can see that the nearest distance is 0.2002 using the formula for distance from point to line.