HW 6 CSM146

1) a)

$$p(C_0|x) = \frac{p(x|C_0)\,p(C_0)}{p(x|C_0)\,p(C_0) + p(x|C_1)\,p(C_1)}$$

$$= \frac{1}{1 + \dfrac{p(x|C_1)\,p(C_1)}{p(x|C_0)\,p(C_0)}} = \frac{1}{1+\exp(-a)} = \sigma(a)$$

yes it can ~~we~~ be written in that form IF $a = -\ln\left(\frac{p(x|C_1)\,p(C_1)}{p(x|C_0)\,p(C_0)}\right) = \ln\left(\frac{p(x|C_0)\,p(C_0)}{p(x|C_1)\,p(C_1)}\right)$

b)

$$a = \ln\left(\frac{p(C_0)}{p(C_1)}\right) - \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) = \ln\left(\frac{p(C_0)}{p(C_1)}\right) + \left(\mu_1^T\Sigma^{-1} - \mu_0^T\Sigma^{-1}\right)x + \frac{1}{2}\left(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1\right)$$

$$\begin{cases} w = (\mu_1 - \mu_0)\Sigma^{-1} \\ b = \frac{1}{2}(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1) + \ln\left(\frac{p(C_0)}{p(C_1)}\right) \end{cases}$$

c)

$$a = \ln\left(\frac{|\Sigma_1|^{1/2}}{|\Sigma_0|^{1/2}}\cdot\frac{p(C_0)}{p(C_1)}\exp\left(\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - \frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\right)\right)$$

$$= \ln\left(\frac{|\Sigma_1|^{1/2}}{|\Sigma_0|^{1/2}}\right) + \ln\left(\frac{p(C_0)}{p(C_1)}\right) + \frac{1}{2}x^T\Sigma_1^{-1}x - \mu_1^T\Sigma_1^{-1}x - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \mu_0^T\Sigma_0^{-1}x - \frac{1}{2}x^T\Sigma_0^{-1}x - \frac{1}{2}\mu_0^T\Sigma_0^{-1}\mu_0$$

$$= \frac{1}{2}x^T(\Sigma_0^{-1} - \Sigma_1^{-1})x + (\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})x + \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 - \mu_1^T\Sigma_1^{-1}\mu_1) + \ln\left(\frac{p(C_0)}{p(C_1)}\right) + \frac{1}{2}\ln\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right)$$

$A = \frac{1}{2}(\Sigma_0^{-1} - \Sigma_1^{-1})$ $\quad b = \frac{1}{2}(\mu_0^T\Sigma_0^{-1}\mu_0 - \mu_1^T\Sigma_1^{-1}\mu_1) + \ln\left(\frac{p(C_0)}{p(C_1)}\right) + \frac{1}{2}\ln\left(\frac{|\Sigma_1|}{|\Sigma_0|}\right)$

$w = \mu_1\Sigma_1^{-1} - \mu_0\Sigma_0^{-1}$

2)

(a) $$p(x^{(1)}, x^{(2)} \dots x^{(m)}, y^{(1)}, y^{(2)} \dots y^{(m)}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)})$$

$$= \prod_{i=1}^{m} p(x^{(i)} | y^{(i)}) p(y^{(i)})$$

$$= \prod_{i=1}^{m} \left[\frac{1-\phi}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T \Sigma^{-1} (x-\mu_0)\right) \right]^{1-y^{(i)}}$$

$$\left[\frac{\phi}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma^{-1} (x-\mu_1)\right) \right]^{y^{(i)}}$$

$$L(\phi, \mu_0, \mu_1 \Sigma) = \ln(\uparrow) = \sum_{i=1}^{m} (1-y^{(i)}) \ln(1-\phi) + (1-y^{(i)}) \ln p(x^{(i)} | y^{(i)} = 0)$$

$$+ y^{(i)} \ln\phi + y^{(i)} \ln p(x^{(i)} | y^{(i)} = 1)$$

$$= \sum_{i=1}^{m} \Big\{ (1-y^{(i)}) \Big[\ln(1-\phi) - \frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(|\Sigma|) - \frac{1}{2}(x^{(i)} - \mu_0)^T \Sigma^{-1}$$

$$(x^{(i)} - \mu_0) \Big] + y^{(i)} \Big[\ln(\phi) - \frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln(|\Sigma|)$$

$$- \frac{1}{2}(x^{(i)} - \mu_1)^T \Sigma^{-1} (x^{(i)} - \mu_1) \Big]$$

b) $\max_{\phi} L(\phi, \mu_0, \mu_1, \Sigma)$

$$\frac{\partial L}{\partial \phi} = 0 = \frac{N_1}{\phi} - \frac{N - N_1}{1-\phi} \Rightarrow \boxed{\phi = \frac{N_1}{N}}$$

now we have to fig. out if this is max or min

$$\frac{\partial^2 L}{\partial \phi^2} = \frac{-N_1}{\phi^2} - \frac{N - N_1}{(1-\phi)^2}$$

N_1, N, ϕ are all positive

so $\frac{\partial^2 L}{\partial \phi^2} < 0$ and $\underline{\underline{\phi}}$ is a min

(c) $$\frac{\partial L}{\partial \mu_0} = 0 = \sum_{i=1}^{m} (1-y^{(i)}) \frac{\partial}{\partial \mu_0} \Big[\underbrace{-\frac{1}{2}(x^{(i)} - \mu_0)^T \Sigma^{-1} (x^{(i)} - \mu_0)}_{} \Big] \rightarrow x^{(i)T} \Sigma^{-1} x^{(i)} - x^{(i)T} \Sigma^{-1} \mu_0 - \mu_0^T \Sigma^{-1} x^{(i)} + \mu_0^T \Sigma^{-1} \mu_0 \Big]$$

$$= -\frac{1}{2} \sum_{i=1}^{m} (1-y^{(i)}) \left[-2\Sigma^{-1} x^{(i)} + 2\Sigma^{-1} \mu_0 \right] = 0 \Rightarrow$$

$$\sum_{i=1}^{m} (1-y^{(i)}) \cancel{\Sigma^{-1}} x^{(i)} = \sum_{i=1}^{m} (1-y^{(i)}) \cancel{\Sigma^{-1}} \mu_0 \Rightarrow \mu_0 = \frac{\sum_{i=1}^{m} (1-y^{(i)}) x^{(i)}}{N_0}$$

$$\frac{\partial^2 L}{\partial \mu_0^{k^2}} = -\cancel{N_0} (1-y^{(i)}) \Sigma^{-1} \rightarrow -H = \begin{bmatrix} (1-y^{(i)})\Sigma^{-1} & & 0 \\ & (1-y^{(i)})\Sigma^{-1} & \\ 0 & & \ddots \end{bmatrix}$$

the diag of $-H$ is >0 so $-H$ is ~~PSD~~ positive definite. thus, H is negative definite. Since H w.r.t μ_0 is negative definite, μ_0 is the ~~best~~/maximum.

3(a)

```
stat = readtable("UCLA_EE_grad_2030.csv")
```

```
x = stat{:,1:2};
y = (stat{:,3} + 1) / 2;
N = length(y);

admit = x(y == 1,:);
rejec = x(y == 0,:);

P0 = length(rejec) / N;
mu0 = mean(rejec);
mu1 = mean(admit);
mu = [mu0; mu1];

P0
```

```
P0 = 0.7900
```

```
mu0
```

```
mu0 = 1×2
    1.8678    1.9673
```

```
mu1
```

```
mu1 = 1×2
    3.1637    2.9590
```

```
covar = zeros(2, 2);
for i = 1:N
    covar = covar + (x(i,:) - mu(y(i) + 1,:))' * (x(i,:) - mu(y(i) + 1,:));
end
covar = covar / N;

covar
```

```
covar = 2×2
    0.4457    0.0731
    0.0731    0.4745
```

```
w = covar \ (mu0' - mu1')
```

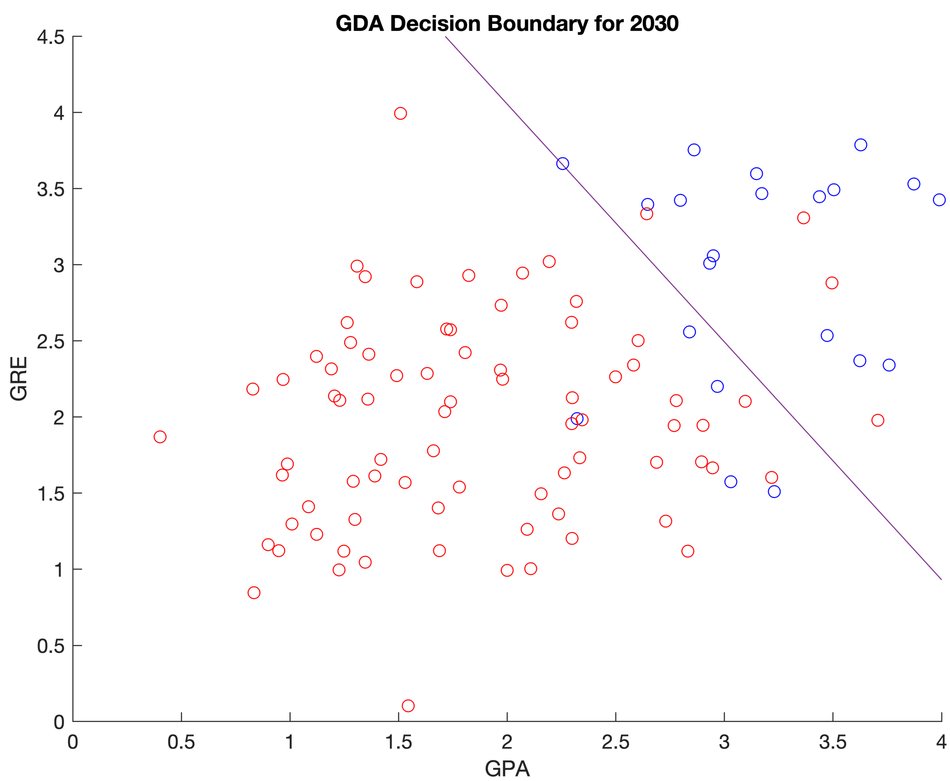
```
w = 2×1
   -2.6314
   -1.6845
```

```
b = -0.5 * (mu0 / covar * mu0' - mu1 / covar * mu1') + log(P0) - log(1 - P0)
```

```
b = 12.0941
```

```
xval = 0:0.1:4;
```

```
yval = 0:0.1:4;
figure(1);
hold on;
plot(xval, -1 / w(2) * (w(1) * xval + b));
ylim([0 4.5]);
scatter(admit(:,1), admit(:,2), 'blue');
scatter(rejec(:,1), rejec(:,2), 'red');
title('GDA Decision Boundary for 2030');
xlabel('GPA');
ylabel('GRE');
```



(c)

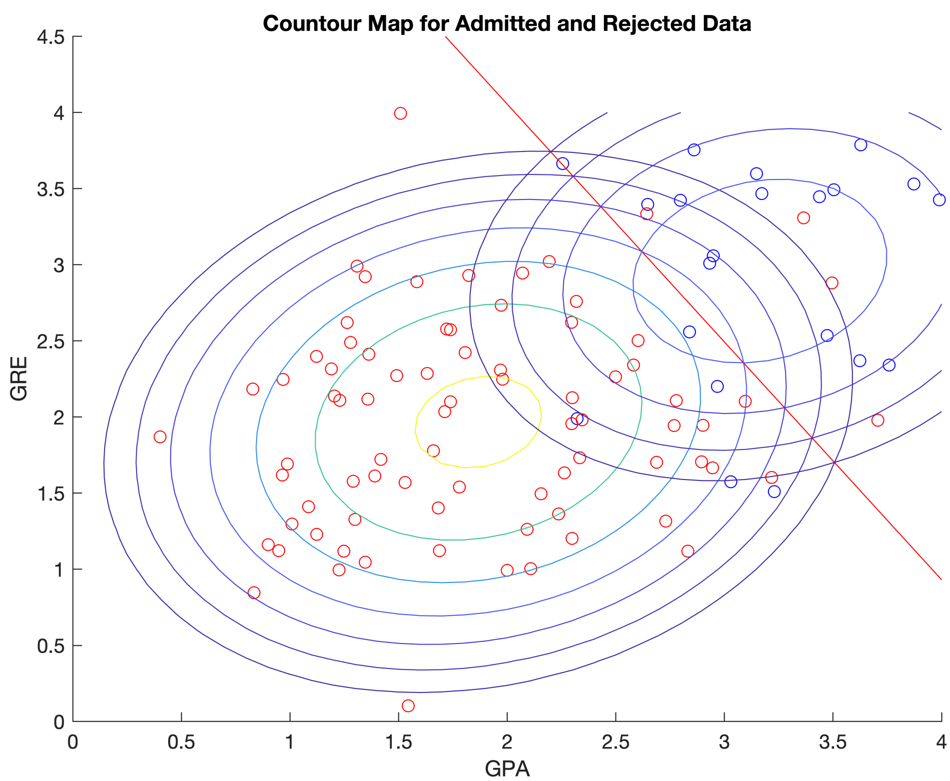
```
M = length(xval);
[GPA, GRE] = meshgrid(xval, yval);

Z0 = zeros(M);
Z1 = zeros(M);
for i = 1:M
    for j = 1:M
        k = [GPA(i, j); GRE(i, j)];
        Z0(i, j) = P0 * mvnpdf(k, mu0', covar);
        Z1(i, j) = (1 - P0) * mvnpdf(k, mu1', covar);
    end
end

figure(2);
hold on;
```

```
scatter(admit(:,1), admit(:,2), 'blue');
scatter(rejec(:,1), rejec(:,2), 'red');
plot(xval, -1 / w(2) * (w(1) * xval + b), 'red');
contour(GPA, GRE, Z0, 'LevelList', logspace(-2, -0.6, 7));
contour(GPA, GRE, Z1, 'LevelList', logspace(-2, -0.6, 7));
ylim([0 4.5]);

title('Countour Map for Admitted and Rejected Data');
xlabel('GPA');
ylabel('GRE');
```



4) $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$ $r_{nk} = 0, 1$

(a) if $k = n$, $x_n = \mu_k \Rightarrow \|x_n - \mu_k\| = 0$ min value = $J = 0$

(b) $\underset{\mu_k}{\operatorname{argmin}} \; \lambda \|\mu_k\|^2 + \sum_{n=1}^{N} r_{nk} \|x_n - \mu_k\|^2$

$$\frac{\partial J}{\partial \mu_k} = 2\lambda \mu_k + 2 \sum_{n=1}^{N} r_{nk}(\mu_k - x_n) = 0$$

When you regularize, J is bigger than without the λ and ~~adds~~ increases even if $r_{nk} = 0$.

$$\mu_k \left(\lambda + \sum_{n=1}^{N} r_{nk}\right) = \sum_{n=1}^{N} r_{nk} x_n$$

$\downarrow$

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\lambda + \sum_{n=1}^{N} r_{nk}}$$

5. L_1 distance

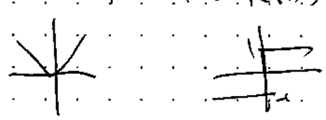
$J = \sum_{n=1}^{N} \sum_{k=1}^{K} |x_{ni} - \mu_{ki}|$ $\|z\|_1 = \sum_{i=1}^{M} |z_i|$ $z \in \mathbb{R}^M$ $r_{nk} \in \{0, 1\}$ $r_{nk} = 1$

$$f(\mu_k) = \sum_{n \in C_k} \|x_n - \mu_k\|_1$$

a) $f(x) = |x|$ $x < 0 \Rightarrow f(x) = -1$ $x > 0$ $\partial f(x) = \{1\}$ y_c $x < 0 \Rightarrow f(x) = -x$
$x > 0 \Rightarrow f(x) = x$

$x = 0$ subdifferential:

$f(z) = |z|$ ~~$f(z)$~~ $|x| \geq gz$ $z = [-1, 1]$



$\partial f(x)$

$$\partial f(x) = \begin{cases} -1 & x < 0 \\ [-1, 1] & x = 0 \\ 1 & x > 0 \end{cases}$$

b)

	y
1	1
2	2
3	3
4	5
5	

$f(\bar{y}) = \sum_{1}^{5} |y_i - g|$ $g = 2, 3, 2.5$

$f(2) = |-1| + |0| + |1| + |2| + |3| = \boxed{7}$ $f(3) = |-2| + |-1| + |0| + |1| + |2| = \boxed{6}$

$f(2.5) = |-2.5| + |-1.5| + |-0.5| + |0.5| + |1.5| = \boxed{6.5}$

$\partial f(2) = -1 + [-1, 1] + 1 + 1 + 1 = 2 + [-1, 1] = \boxed{[1, 3]}$

$\partial f(3) = -1 - 1 + [-1, 1] + 2 = \boxed{[-1, 1]}$

$\partial f(2.5) = -1 - 1 - 1 + 1 + 1 = \boxed{-1}$

c) y_1 1
y_2 2
y_3 3
y_4 4
y_5 5
y_6 6

$f(\bar{y}) = \sum_{i=1}^{6} |y_i - \bar{y}|$

$\bar{y} = 3, 3.5, 4, 5$

$f(3) = |-2|+|-1|+|0|+|1|+|2|+|3| = 9$

$f(3.5) = |-2.5|+|-1.5|+|-0.5|+|0.5|+|1.5|+|2.5| = 9$

$f(4) = |-3|+|-2|+|-1|+|0|+|1|+|2| = 9$

$f(5) = |-4|+|-3|+|-2|+|-1|+|0|+|1| = 11$

$\partial f(3) = -1-1+[-1,1]+1+1 = [0,2]$

$\partial f(3.5) = -1-1-1+1+1+1 = 0$

$\partial f(4) = -1-1-1+[-1,1]+2 = [-2,0]$

$\partial f(5) = -1-1-1-1+[-1,1]+1 = [-4,-2]$

d) $\partial f(\bar{y}^*) = 0 \rightarrow \bar{y}^* = \frac{y_N - y_1}{2} + 1$

value minimizes $\text{median}(y_1, y_2, y_3 \ldots y_N)$

half of $y - \bar{y} < 0$ and half > 0. thus,

derivative $\partial f(\bar{y}^*) \approx 0$.

e) $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|$

1) Find all x_n where r_{nk} is nonzero

set this set to A

2) find $\text{median}(A)$ and set it to μ_k^*

also faster than K means and easier

6.

a) $\det(A) = 9a - 36 > 0$

$\boxed{a > 4}$

b) $z^T B z > 0$

let $y = Bz$ and $z = B^{-1}y$
therefore

Since B is invertible

$z^T B z = z^T B^T B^{-1} B z$

$= x^T B^{-1} x > 0$ so B^{-1} is PSD.

c) $S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T$

$$S = \begin{bmatrix} \sigma_1^2 & \mathrm{cov}(x_1, x_2) & \mathrm{cov}(x_1, x_3) \\ & \sigma_2^2 & \\ & & \ddots \quad \sigma_n^2 \end{bmatrix} = E\left[(x - \bar{x})(x - \bar{x})^T\right]$$

$$y^T S y = y^T \left(\frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T \right) y$$

$$= \frac{1}{N} \sum_{n=1}^{N} \underbrace{\left((x_n - \bar{x})^T y \right)^2}_{\text{positive so}}$$

$y^T S y$ is always > 0

for all y $\rightarrow$ S is PSD. $\square$

5(a)

```
bruin = imread("UCLA_Bruin.jpg");
image(bruin);
```


```
n = size(bruin, 1);
m = size(bruin, 2);
```

(b)

```
K = 4;
mu = zeros(K, 3);
mu(1,:) = bruin(1,1,:);

for k = 2:K
    maxd = -1;
    for x = 1:n
        for y = 1:m
            mind = 1e9;
            cxy = zeros(1, 3);
            cxy(:) = bruin(x,y,:);
            for kp = 1:(k-1)
                mind = min(norm(cxy - mu(kp,:))^2, mind);
            end
            if (maxd < mind)
                maxd = mind;
                mu(k,:) = cxy(:);
            end
```

```
        end
    end
end
mu
```

```
mu = 4x3
   147   200   250
     0     0     0
   137   141    57
    31    70   125
```

```
r = zeros(300, 400, K);
J = zeros(10, 1);
for iter = 1:10
    for x = 1:n
        for y = 1:m
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            mind = 1e9;
            mink = 0;
            for kp = 1:K
                dist = norm(cxy - mu(kp,:)');
                if (mind > dist)
                    mind = dist;
                    mink = kp;
                end
            end
            for kp = 1:K
                r(x, y, kp) = 0;
            end
            r(x, y, mink) = 1;
        end
    end
    for k = 1:K
        mean = zeros(1, 3);
        count = 0;
        for x = 1:n
            for y = 1:m
                cxy = cast(squeeze(bruin(x,y,:)), 'double');
                mean = mean + r(x, y, k) * cxy';
                count = count + r(x, y, k);
            end
        end
        mu(k,:) = mean / count;
    end
    sum = 0;
    for x = 1:n
        for y = 1:m
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            for k = 1:K
                sum = sum + r(x, y, k) * norm(cxy - mu(k,:))^2;
            end
        end
    end
    J(iter) = sum;
end
```

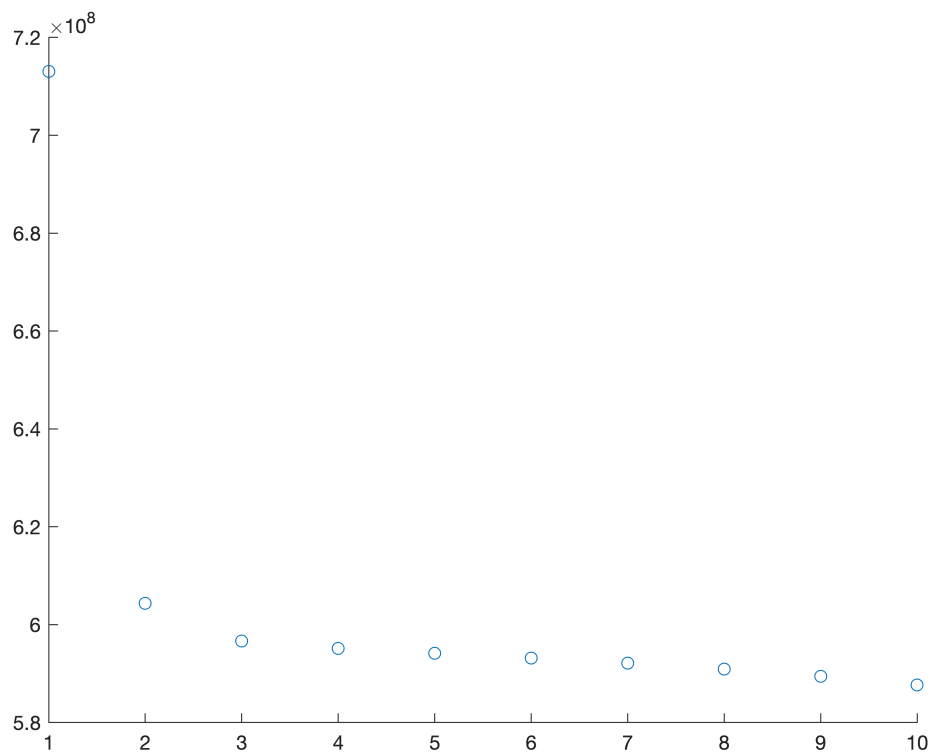
```
J(10)
```

```
ans = 5.8770e+08
```

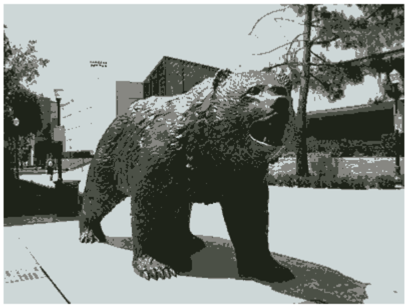
```
mu
```

```
mu = 4×3
  205.3092  218.2813  218.4876
   33.6186   36.2153   27.3633
  134.5668  136.7124  130.2239
   77.8911   83.4243   72.4920
```

```
newbruin = bruin;
for x = 1:n
    for y = 1:m
        kp = 0;
        for k = 1:K
            if r(x, y, k) == 1
                newbruin(x, y, :) = mu(k,:);
            end
        end
    end
end
scatter(1:10, J);
```




```
imshow(newbruin);
```



(c)

```
K = 8;
mu = zeros(K, 3);
mu(1,:) = bruin(1,1,:);

for k = 2:K
    maxd = -1;
    for x = 1:n
        for y = 1:m
            mind = 1e9;
            cxy = zeros(1, 3);
            cxy(:) = bruin(x,y,:);
            for kp = 1:(k-1)
                mind = min(norm(cxy - mu(kp,:))^2, mind);
            end
            if (maxd < mind)
                maxd = mind;
                mu(k,:) = cxy(:);
            end
        end
    end
end
mu
```

```
mu = 8×3
   147   200   250
     0     0     0
   137   141    57
    31    70   125
   251   214   159
   109   139   165
    69    81     0
   242   255   255
```

```
r = zeros(300, 400, K);
J = zeros(10, 1);
for iter = 1:10
    for x = 1:n
        for y = 1:m
```

```
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            mind = 1e9;
            mink = 0;
            for kp = 1:K
                dist = norm(cxy - mu(kp,:)');
                if (mind > dist)
                    mind = dist;
                    mink = kp;
                end
            end
            for kp = 1:K
                r(x, y, kp) = 0;
            end
            r(x, y, mink) = 1;
        end
    end
    for k = 1:K
        mean = zeros(1, 3);
        count = 0;
        for x = 1:n
            for y = 1:m
                cxy = cast(squeeze(bruin(x,y,:)), 'double');
                mean = mean + r(x, y, k) * cxy';
                count = count + r(x, y, k);
            end
        end
        mu(k,:) = mean / count;
    end
    sum = 0;
    for x = 1:n
        for y = 1:m
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            for k = 1:K
                sum = sum + r(x, y, k) * norm(cxy - mu(k,:))^2;
            end
        end
    end
    J(iter) = sum;
end
J(10)
```

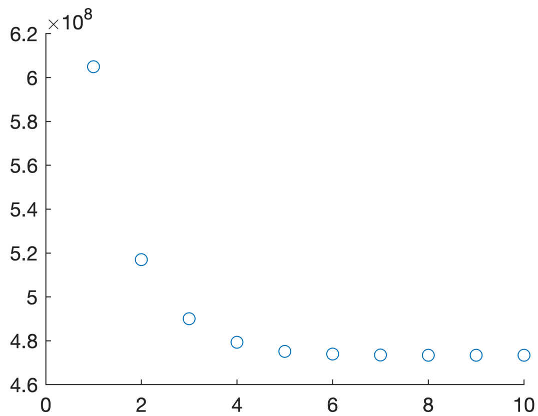
```
ans = 4.7337e+08
```

```
mu
```

```
mu = 8x3
  168.8847  217.7350  250.7417
   22.0897   25.8861   19.4480
  105.9684  111.8957  105.6784
   78.3522   84.3161   72.1193
  207.4103  199.6460  179.9658
  153.6260  155.0707  150.9627
   53.1262   54.7085   42.1512
  235.8857  228.0374  209.6141
```

```
newbruin = bruin;
```

```
for x = 1:n
    for y = 1:m
        kp = 0;
        for k = 1:K
            if r(x, y, k) == 1
                newbruin(x, y, :) = mu(k,:);
            end
        end
    end
end
scatter(1:10, J);
```




```
imshow(newbruin);
```

```
K = 16;
mu = zeros(K, 3);
mu(1,:) = bruin(1,1,:);

for k = 2:K
    maxd = -1;
    for x = 1:n
        for y = 1:m
            mind = 1e9;
            cxy = zeros(1, 3);
            cxy(:) = bruin(x,y,:);
            for kp = 1:(k-1)
```

```
                mind = min(norm(cxy - mu(kp,:))^2, mind);
            end
            if (maxd < mind)
                maxd = mind;
                mu(k,:) = cxy(:);
            end
        end
    end
end
mu
```

```
mu = 16×3
   147   200   250
     0     0     0
   137   141    57
    31    70   125
   251   214   159
   109   139   165
    69    81     0
   242   255   255
   196   144   130
    96    69    76
     ⋮
```

```
r = zeros(300, 400, K);
J = zeros(10, 1);
for iter = 1:10
    for x = 1:n
        for y = 1:m
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            mind = 1e9;
            mink = 0;
            for kp = 1:K
                dist = norm(cxy - mu(kp,:)');
                if (mind > dist)
                    mind = dist;
                    mink = kp;
                end
            end
            for kp = 1:K
                r(x, y, kp) = 0;
            end
            r(x, y, mink) = 1;
        end
    end
    for k = 1:K
        mean = zeros(1, 3);
        count = 0;
        for x = 1:n
            for y = 1:m
                cxy = cast(squeeze(bruin(x,y,:)), 'double');
                mean = mean + r(x, y, k) * cxy';
                count = count + r(x, y, k);
            end
        end
```

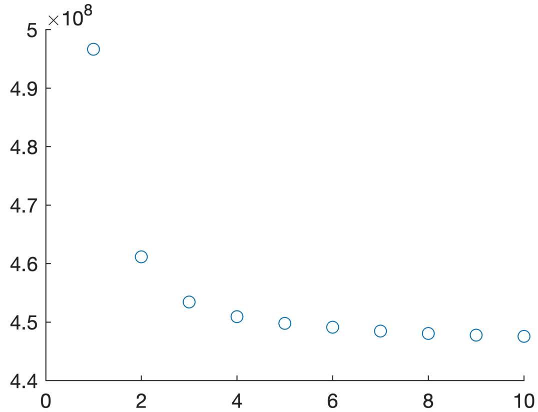
```
        mu(k,:) = mean / count;
    end
    sum = 0;
    for x = 1:n
        for y = 1:m
            cxy = cast(squeeze(bruin(x,y,:)), 'double');
            for k = 1:K
                sum = sum + r(x, y, k) * norm(cxy - mu(k,:))^2;
            end
        end
    end
    J(iter) = sum;
end
J(10)
```

```
ans = 4.4759e+08
```

```
mu
```

```
mu = 16×3
  169.4358  218.6251  251.6993
   17.7295   21.7464   16.2366
  109.1421  109.8681   70.8890
   63.4478   77.9096   86.0944
  233.2694  222.1816  198.6191
  129.6679  146.0076  157.9573
   58.2489   58.8509   45.5631
  238.0500  234.0214  221.4212
  188.7162  172.5177  152.1534
   81.2063   81.7605   58.6404
      ⋮
```

```
newbruin = bruin;
for x = 1:n
    for y = 1:m
        kp = 0;
        for k = 1:K
            if r(x, y, k) == 1
                newbruin(x, y, :) = mu(k,:);
            end
        end
    end
end
scatter(1:10, J);
```




```
imshow(newbruin);
```

(d)

The original image is 300 * 400 * 3 * 8 = 2880000 bits needed. After the K-means algorithm the image is compressed and only relates to the size of the r matrix and the number of centers.

K = 4 needs 4 * 3 * 8 for the 4 centers, and 300 * 400 * 2 = 240000 bits approximately. The compression ratio is about 12:1

K = 8 needs approximately 300 * 400 * 3 = 360000 bits. The compression ratio is about 8:1

K = 16 needs approx 300 * 400 * 4 = 480000 bits. The compression ratio is about 6:1.