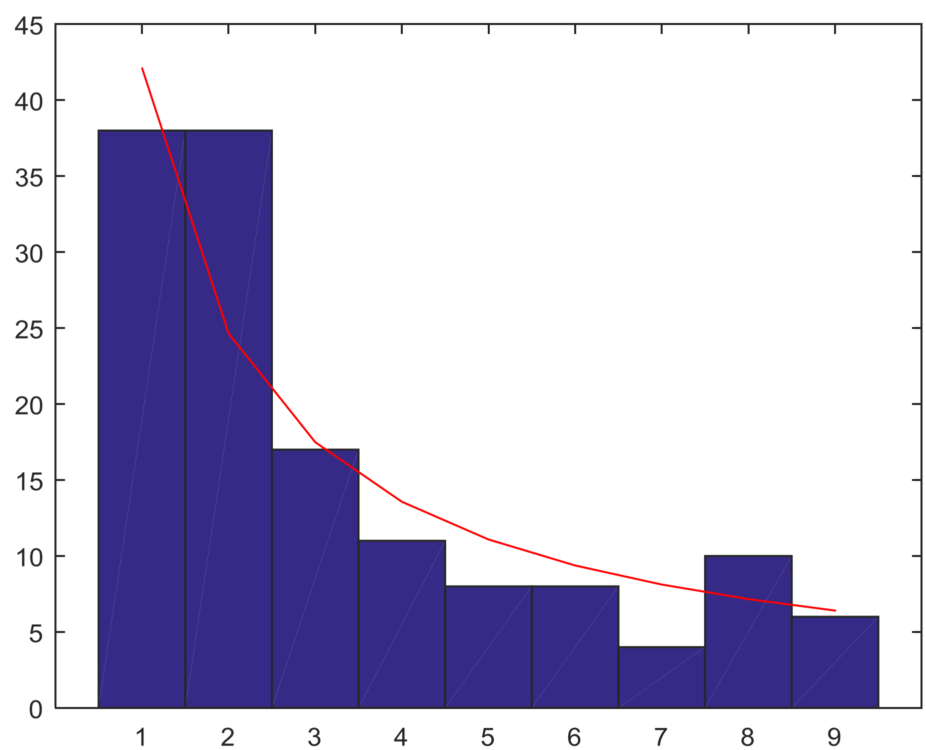
Hw5

Problem 1:

A.

```
vals = textread('incomeTax/HR_Clinton_2014_tax_return_numbers.txt', '%s');

numbers = unique(str2double(vals));

firstdigit = @(x) floor(x ./ (10 .^ floor(log10(x)))); %obtain first digits

number_of_bins = 9;
nu = number_of_bins - 1;

Histogram = hist(firstdigit(numbers), 1:9);
hist(firstdigit(numbers), 1:9);

BenfordProbabilities = diff(log10(1:10));
N = length(numbers);
BenfordHistogram = N * BenfordProbabilities;
hold on
plot(1:9, BenfordHistogram, 'r')

ChiSquareStatistic = sum((Histogram - BenfordHistogram) .^2 ./
BenfordHistogram)
ChiSquareProbability = cdf('Chisquare', ChiSquareStatistic, nu)
```




```
ChiSquareStatistic =
```

```
    12.4337


ChiSquareProbability =

    0.8671
```

The chi-square probability is a little high, but still below 90%. I would say that the tax return is probably fine and not clearly fraudulent.

Problem 2:

1. The simplified derivative is Summation[2*(f(t) – P) * f'(t)]

    => 2*b*cos(t)*(b*sin(t) – pY) – 2*a*sin(t)*(a*cos(t) - pX).

2.

Timestep `deltaT = 0.01`, represents the time elapsed in each iteration when the simulation is updated.

The outside force acting on the particles is gravity. `gravity = [0 -9.81];`

A particle is outside the ellipse when `implicitEllipse( pX, pY ) > 1`

3.

```
% Read in the csv file here.
P = csvread('particleData.csv', 1, 0);

for I = 1:particlesCount
     particles(I).position = [P(I, 1), P(I, 2)];        % Position.
     particles(I).velocity = [P(I, 3), P(I, 4)];        % Velocity.
     particles(I).color = [P(I, 5), P(I, 6), P(I, 7)];          % Color.
     particles(I).mass = P(I, 8);           % Mass.
end;
```

4.

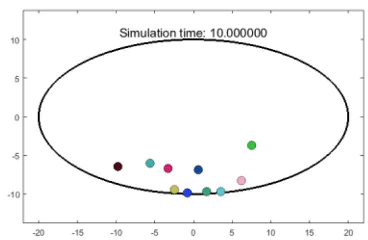
```
oldV = particles(I).velocity;
         particles(I).velocity = oldV + deltaT * gravity;        % Integrate
acceleration to find new velocity.;
         particles(I).position = particles(I).position + deltaT * oldV;
% Integrate velocity to find new position.
```

5.

```
% Function for which you want to find a root.
             df = @(t) 2*b*cos(t)*(b*sin(t) - pY) - 2*a*sin(t)*(a*cos(t) -
pX);
            %
            % Pick an initial value to launch Newton's method.
             t0 = atan2(pY, pX);
```

```
            %
            % t value for which the distance between ellipse and particle is
minimal.
             tStar = fzero( df, t0 );
            %
            % Use tStar to relocate particle ON the ellipse
             particles(I).position = [a*cos(tStar) b*sin(tStar)];
```

6.



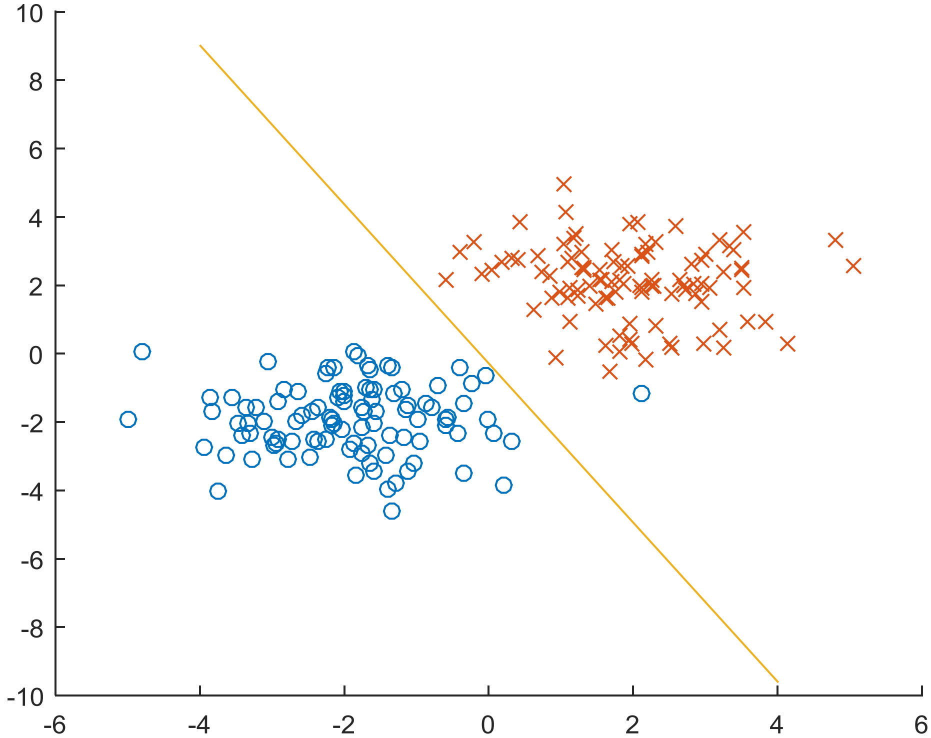
Problem 3:

1.

```
function J = cost(X, y, theta)
      h = 1./(1+exp(-theta*X'));
      J = (1/200) * (-log(h)*y - log(1-h)*(1-y));
```

2.

```
best_theta = fminsearch(@(theta)cost(data, label, theta), [0, 0, 0]);
```


10
8
6
4
2
0
-2
-4
-6
-8
-10
-6
-4
-2
0
2
4
6