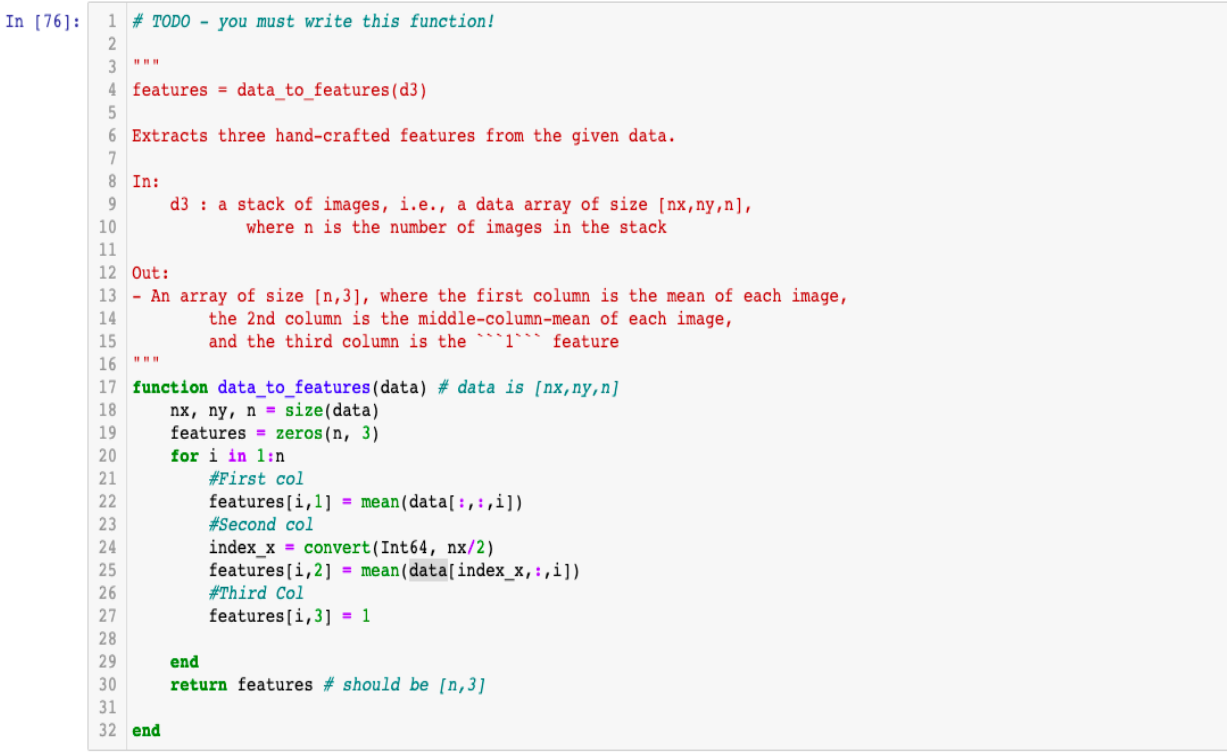
Group member names:
Yuzhan Jiang
Moyan Lyu
Yu Gan

Group member uniquenames/emails:
moyanlyu@umich.edu
yuzhanji@umich.edu
arkhamyu@umich.edu

Part 1

Code for your `data_to_features` function (a screenshot or copy-paste are both fine):



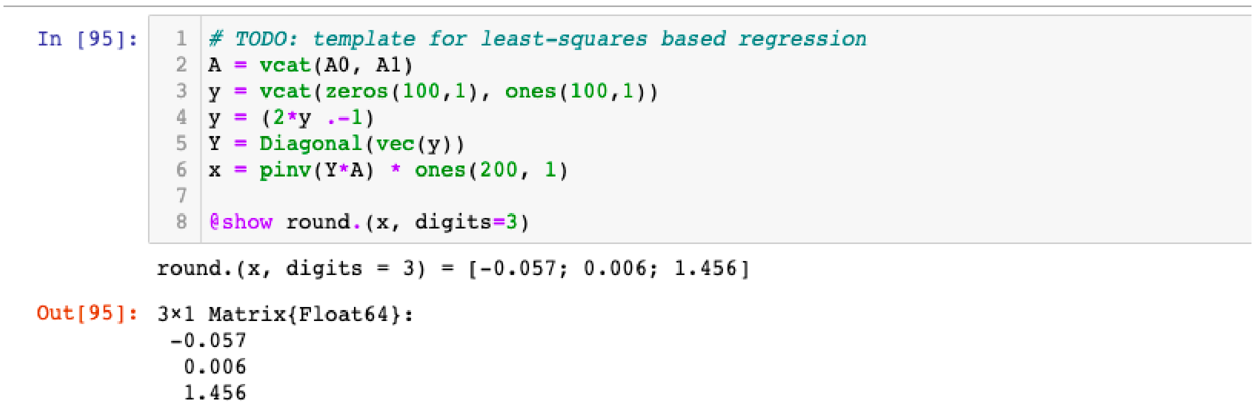
In [76]:

```
# TODO - you must write this function!

"""
features = data_to_features(d3)

Extracts three hand-crafted features from the given data.

In:
    d3 : a stack of images, i.e., a data array of size [nx,ny,n],
            where n is the number of images in the stack

Out:
- An array of size [n,3], where the first column is the mean of each image,
        the 2nd column is the middle-column-mean of each image,
        and the third column is the ```1``` feature
"""
function data_to_features(data) # data is [nx,ny,n]
    nx, ny, n = size(data)
    features = zeros(n, 3)
    for i in 1:n
        #First col
        features[i,1] = mean(data[:,:,i])
        #Second col
        index_x = convert(Int64, nx/2)
        features[i,2] = mean(data[index_x,:,i])
        #Third Col
        features[i,3] = 1

    end
    return features # should be [n,3]

end
```

Name of group member who wrote this part: Yuzhan Jiang /yuzhanji@umich.edu

Part 2

x values (these are your learned coefficients for doing the linear regression) =



In [95]:

```
# TODO: template for least-squares based regression
A = vcat(A0, A1)
y = vcat(zeros(100,1), ones(100,1))
y = (2*y .-1)
Y = Diagonal(vec(y))
x = pinv(Y*A) * ones(200, 1)

@show round.(x, digits=3)
```

```
round.(x, digits = 3) = [-0.057; 0.006; 1.456]
```

Out[95]:

```
3×1 Matrix{Float64}:
 -0.057
  0.006
  1.456
```

Explain why x[1] is negative (2-3 sentences):
X[1] is negative because the first LS cost function's minimum value belongs to 0digit class. Therefore, y is -1 for the 0 digits. So the first element of x, (Diag(y)A)+ is negative.

From another perspective, based on the scatter plot. It is obvious that the separating line should go through the first, third and fourth quadrant. Since the representation of this line is :

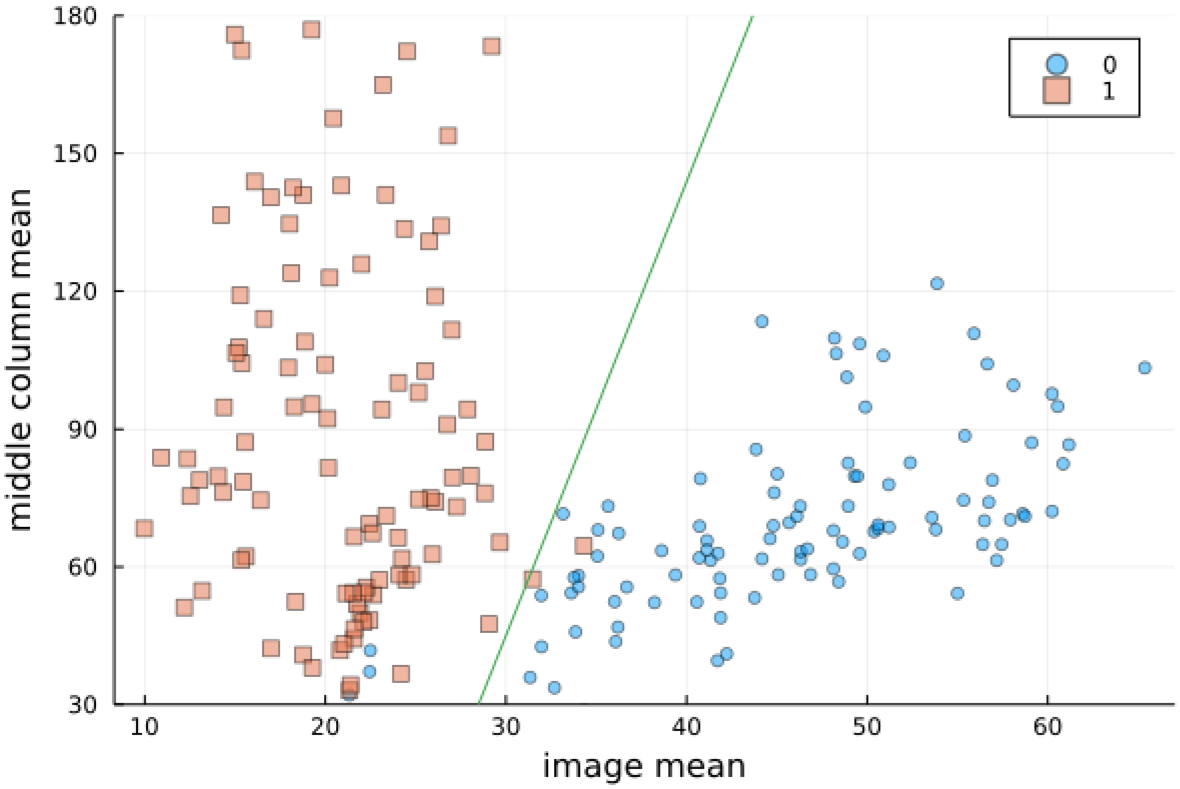
y = - x[1]/x[2]*x - x[3]/x[2]

So,

-x[1]/x[2]>0,
x[3]/x[2]>0.

As a conclusion, x[2] and x[3] should have the same sign which x[1] is different.

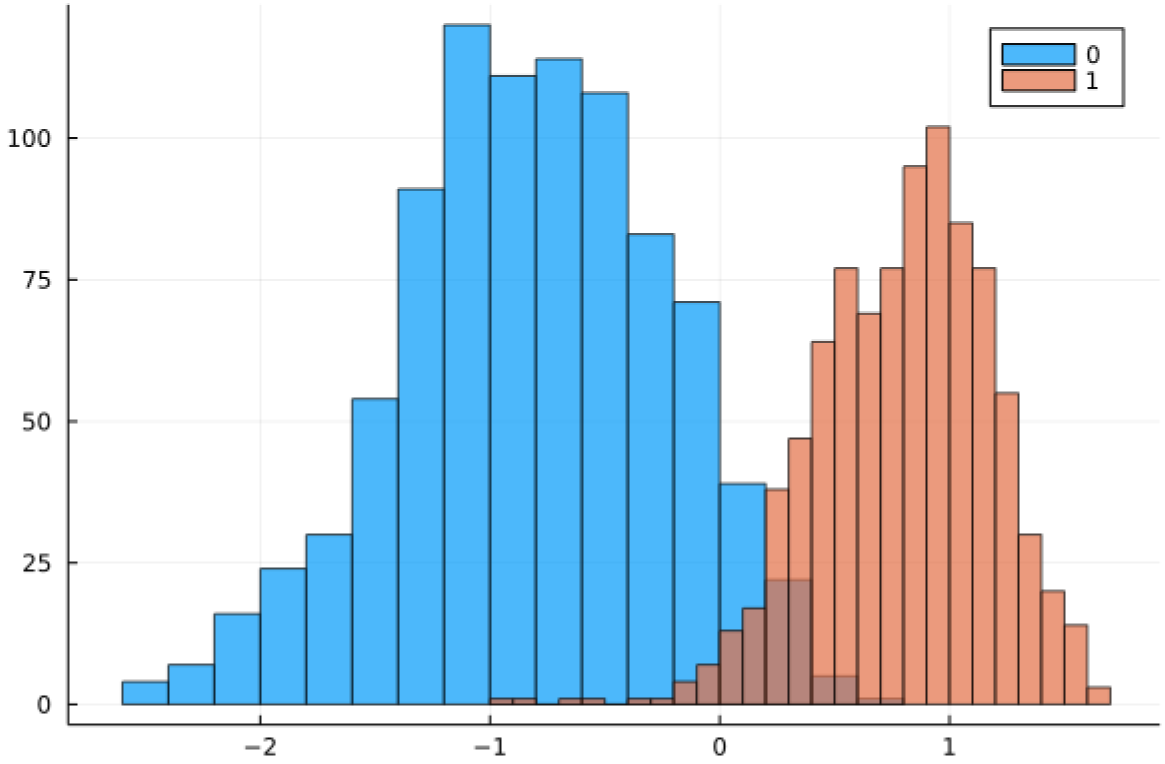
Scatter plot with separating line:



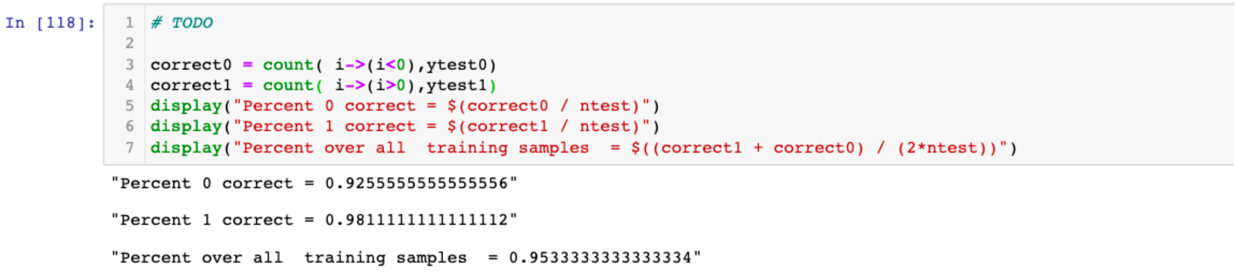
Name of group member who wrote this part: Moyan Lyu/moyanlyu@umich.edu

Part 3

Histogram:



Percent of correctly classified digits:

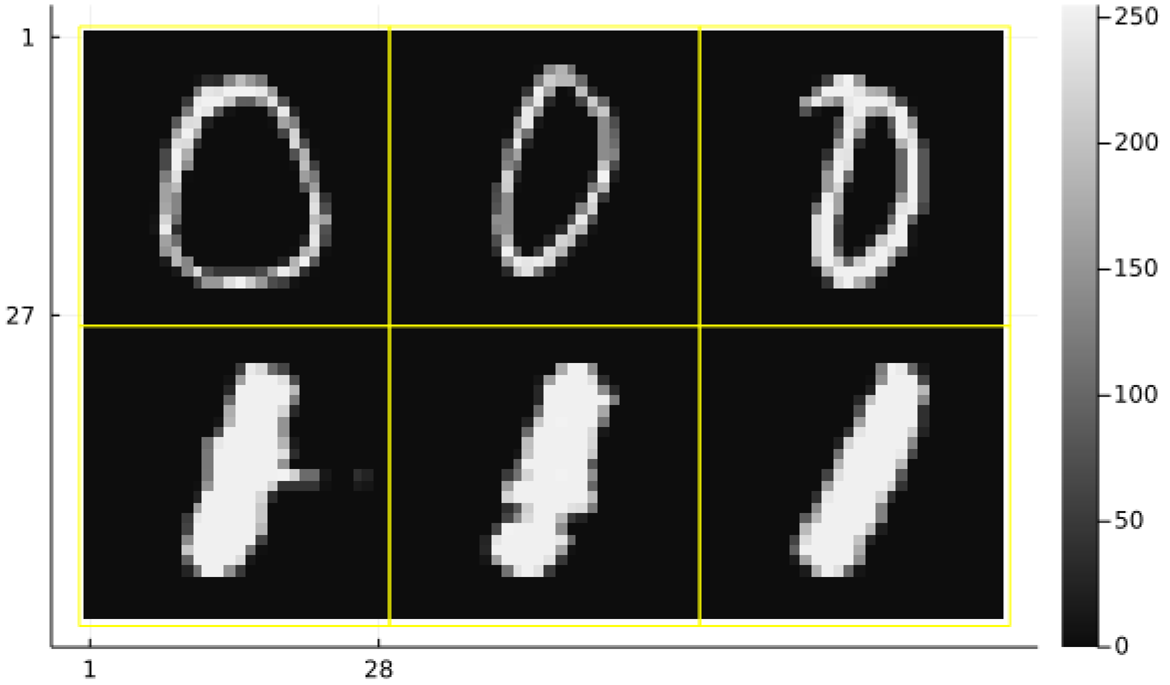

```
In [118]:
# TODO

correct0 = count( i->(i<0),ytest0)
correct1 = count( i->(i>0),ytest1)
display("Percent 0 correct = $(correct0 / ntest)")
display("Percent 1 correct = $(correct1 / ntest)")
display("Percent over all  training samples  = $((correct1 + correct0) / (2*ntest))")
```

```
"Percent 0 correct = 0.9255555555555556"

"Percent 1 correct = 0.9811111111111112"

"Percent over all  training samples  = 0.9533333333333334"
```

Name of group member who wrote this part: Yu Gan/ arkhamyu@umich.edu

Part 4

Show some of the misclassified images:

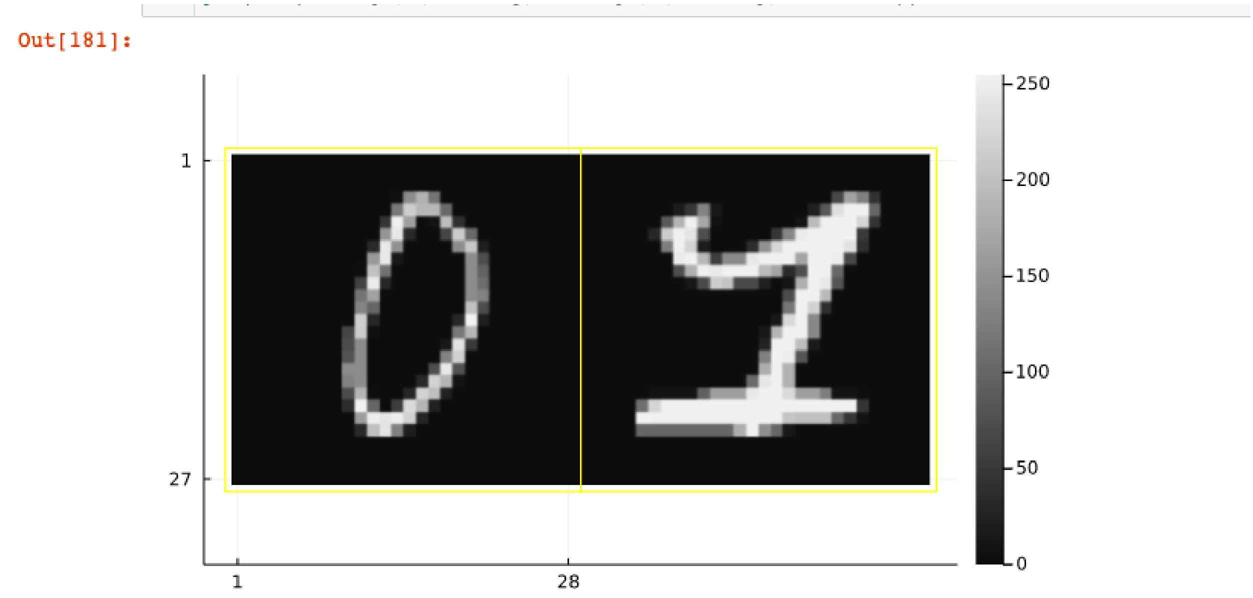


Comment on why these were misclassified (2-3 sentences):
These numbers were misclassified may because their shape is out of the league of the mean Shape of 0 digits and 1 digits. In specific, like the number 0 in the up right hand corner, it has an Extra part than the average 0. And the digits 1 in the lower row are wider than the normal 1.

Name of group member who wrote this part: Moyan Lyu/moyanlyu@umich.edu

Part 5

Worst classified digits using Tikhonov Regularization when beta = 10



Out[181]:

How do you think our classifier will perform as beta tends to infinity?
In my opinion, our classifier will fail to work as expected if beta tends to infinity. Since β influence the effect of regularization. When β is too large, the new x will be away from the estimated x we got previously. So the classifier will become too underfitting to classify 0 and 1.

Name of group member who wrote this part: Yuzhan Jiang /yuzhanji@umich.edu