Matt Hall
Homework 3

Problem 1:

Question 1 -

dat:

```
mean [-0.009481 -0.008284 -0.009672]
variance   [[.9983 .0001687 .001237]
            [.0001687 1.003 .001553]
            [.001237 .001553 1.005]]
```

dat1:

```
mean [0.9905 1.992 2.990]
variance   [[.9983 .0001687 .001237]
            [.0001687 1.003 .001553]
            [.001237 .001553 1.005]]
```

dat2:

```
mean [9.905 5.975 2.990]
variance   [[99.83 .005061 .01237]
            [.005061 9.031 .004658]
            [.01237 .004658 1.005]]
```

dat3:

```
mean [11.26 3.532 -1.886]
variance   [[49.15 44.62 6.870]
            [44.62 58.64 7.514]
            [6.870 7.514 2.067]]
```

Question 2 –

The direction of maximum variance for dat2, from the eigenvector, is along the x-axis (the first dimension, which was multipled by 10). This makes sense since we adjusted variances by multiplying dat1 by the diag[10 3 1] vector.

The direction of max variance for dat3, from its eigenvector, is approaching the 1st row of the rotation matrix R. This makes sense since we rotated our dataset using the R matrix.

Question 3 –

The first principal component of dat3 is [0.6651, 0.7427, -0.0770]. Yes this is what would be expected since the original data was rotated, which directly affects the covariance of the data, the principal component(s) would tend towards this rotation matrix.

Question 4 –

The mean and variance from dat are as expected, since the data was generated using N(0,1), the mean should be around 0 the diagonals of the covariance matrix around 1.

The manipulation of dat to get dat1 would only change the means, by 1, 2 and 3 respectively which is what we confirmed, and variances would remain unchanged.

The manipulation of dat1 to dat2, by multiplying by diag[10 3 1] impacts the variance and we confirmed this with our covariance matrix. The variance matrix should be approximately [100 9 1], which is confirmed in our dat2 covariance matrix. As the number increases the variance will approach the [100 9 1] variances.

Matt Hall
Homework 3

Rotating dat2 to obtain dat3 generates expected results. Calculating the eigenvalues/vectors shows that the principal components will tend towards this rotation matrix as the number of samples is increased.

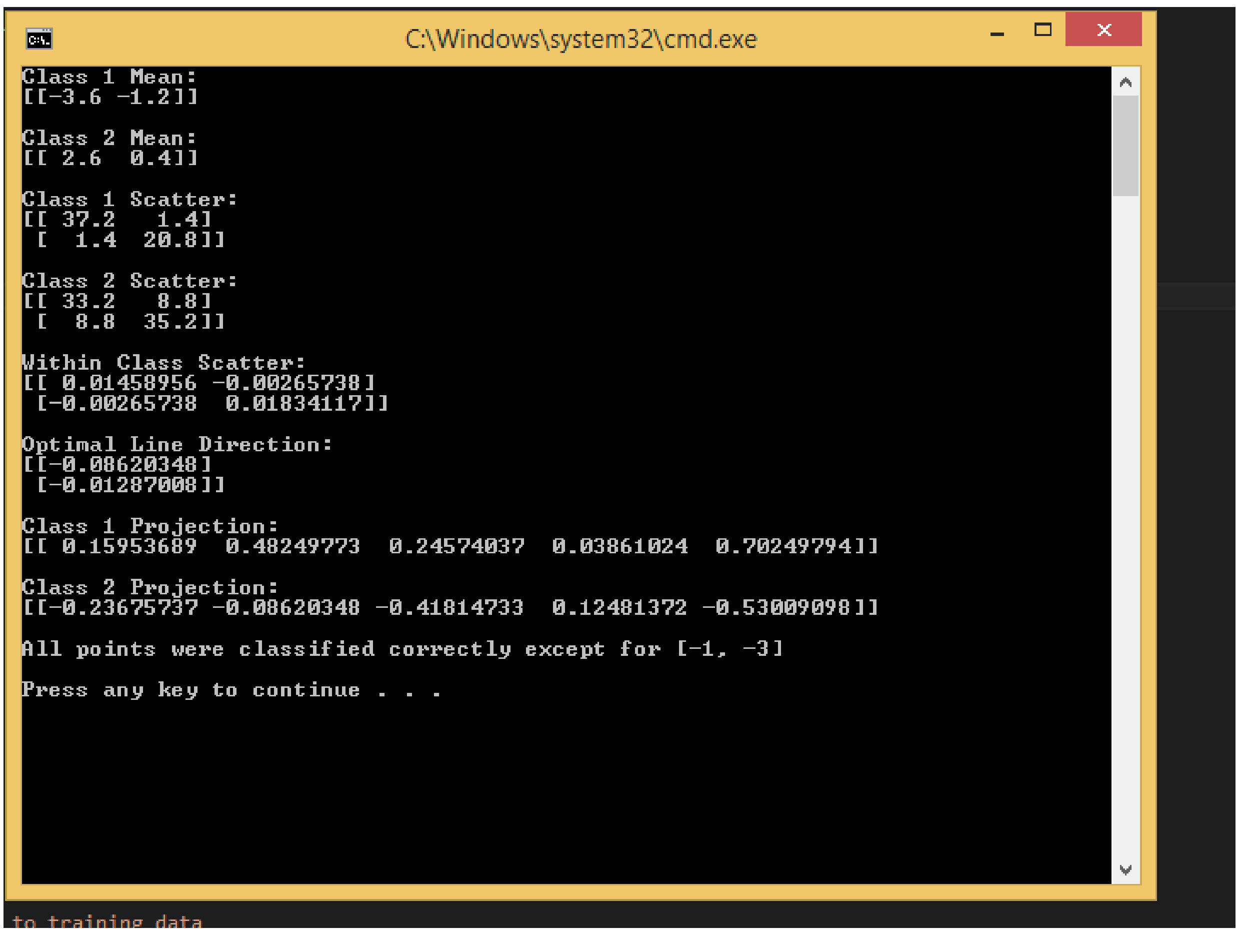
Problem 2:

Average Accuracy: 74.79%
Std Dev of Accuracy: 2.456

In order to get the 3 principal components, the eigenvalues were calculated from the training data covariance matrix. The eigenvectors corresponding to the 3 largest eigenvalues were selected as the principal components.

For the final run the 3 components were:
1 [0.002445 -0.02436 0.02302 0.07141 0.1672 0.001913 -0.9827 0.003658]
2 [-0.09163 -0.9746 -0.1275 -0.1460 -0.04565 0.04682 0.002648 -0.001121]
3 [-0.01265 -0.1263 0.9293 0.1637 -0.2857 0.1084 -0.01164 0.0009828]

Problem 3:



```
C:\Windows\system32\cmd.exe

Class 1 Mean:
[[-3.6 -1.2]]

Class 2 Mean:
[[ 2.6  0.4]]

Class 1 Scatter:
[[ 37.2   1.4]
 [  1.4  20.8]]

Class 2 Scatter:
[[ 33.2   8.8]
 [  8.8  35.2]]

Within Class Scatter:
[[ 0.01458956 -0.00265738]
 [-0.00265738  0.01834117]]

Optimal Line Direction:
[[-0.08620348]
 [-0.01287008]]

Class 1 Projection:
[[ 0.15953689  0.48249773  0.24574037  0.03861024  0.70249794]]

Class 2 Projection:
[[-0.23675737 -0.08620348 -0.41814733  0.12481372 -0.53009098]]

All points were classified correctly except for [-1, -3]

Press any key to continue . . .
```

Matt Hall
Homework 3

Problem 4:

Average Accuracy: 71.80%
Std Dev of Accuracy: 1.782

Optimal projection:
[-0.1401 -0.04205 0.01861 -0.008783 0.0009270 -0.08736 -0.9849 -0.02385]

Problem 5:

(a)
Want to minimize training error.
Training error will be zero if:
    {a.transpose() * yi > 0 for all yi in c1
    {a.transpose() * yi < 0 for all yi in c2

After normalization training error will be zero if:
    {a.transpose() * yi > 0 for all yi in c1
    {a.transpose() * (-yi) > 0 for all yi in c2
so, we want to find a vector `a` such that a.transpose() * yi > 0 for all yi.

(b)
Resulting 'a' vector:
[1, 3, 1, -1, 3, -5]