// The program I made requires 3 files, the driver (pa1.cpp), the class header (PatternSet.h), and the class definition(PatternSet.cpp) I wrote it in c++ (noway) and leveraged gsl for most of the matrix manipulation. Ended up doing the euclidean and angular comparison functions by hand because I was having more trouble figuring out how to do it using gsl or something similar than it would be to just write it all by hand. I've included a make file (written by hand, im so fancy) that should let you compile it simply

//Trace of program. Side note; I apparently did this in 2011, how amazing is that. Can I get super bonus credit? (engapps00 time fail)

```
john@zakum:~$ ssh jlusby@engapps00
jlusby@engapps00's password:
Linux engapps00.ucmerced.edu 2.6.32.28 #6 SMP Tue Feb 22 06:16:56 UTC 2011 x86_64

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

_____

The use of this server (engapps00.ucmerced.edu) is limited only to
instructional
related activities. For information on School of Engineering research
computational resources/support please contact support@eng.ucmerced.edu
_____
Last login: Wed Feb 20 05:28:22 2013 from ucmerced-169-236-101-152.ucmerced.edu
$ bash
jlusby@engapps00:/mnt/home10/jlusby$ cd machine\ Learning\ CSE\ 176/pa1/source/
jlusby@engapps00:/mnt/home10/jlusby/machine Learning CSE 176/pa1/source$ make
g++ -c -Wall -pedantic -I/usr/include/gsl pa1.cpp
pa1.cpp: In function ‘int main(int, char**)’:
pa1.cpp:18: warning: statement has no effect
pa1.cpp:42: warning: statement has no effect
g++ -c -Wall -pedantic -I/usr/include/gsl PatternSet.cpp
PatternSet.cpp: In member function ‘void PatternSet::ReadPatterns(FILE*)’:
PatternSet.cpp:58: warning: statement has no effect
PatternSet.cpp:63: warning: statement has no effect
PatternSet.cpp: In member function ‘void PatternSet::GenerateRandomOrderingSequence()’:
PatternSet.cpp:87: warning: statement has no effect
PatternSet.cpp:94: warning: statement has no effect
PatternSet.cpp:96: warning: statement has no effect
```

```
PatternSet.cpp: In member function ‘PatternSet* PatternSet::ProjectOntoPCA()’:
PatternSet.cpp:212: warning: statement has no effect
g++ -L/usr/local/lib -lgsl -lgslcblas -lm pa1.o PatternSet.o -o pa1
jlusby@engapps00:/mnt/home10/jlusby/machine Learning CSE 176/pa1/source$ make
make: Nothing to be done for `all'.
jlusby@engapps00:/mnt/home10/jlusby/machine Learning CSE 176/pa1/source$ ./pa1
Enter the input vector dimensionality:
2
Enter the number of patterns:
60
Enter pattern file pathname:
trace-input.txt
Enter the target vector (dim:2), with elements separated by whitespace:
100.0 0.0


Random ordering
56.512100 51.466900
84.108100 54.566600
60.318200 37.061000
58.531600 33.858600
79.004100 61.111900
73.697800 38.815700
40.674700 23.385700
76.152000 41.949700
60.975900 52.210500
31.046200 19.739200
52.074100 31.787700
57.969500 37.322600
68.875600 43.002000
50.868400 35.777000
51.694000 35.245500
34.420700 19.450600
66.799600 44.421400
81.470000 50.130900
55.884600 35.867000
72.166300 43.858900
70.881900 56.635200
79.087700 59.005500
56.038900 32.992800
57.258300 43.576000
57.400700 43.112200
57.157000 45.031300
```

52.254600 35.901300
67.591200 44.174300
56.821100 57.439100
64.563000 27.950100
42.916900 27.928900
55.149700 45.838000
54.437500 45.307600
70.389000 42.600100
37.776300 37.079600
79.917100 55.734800
61.619000 49.739100
53.533200 28.872100
61.959900 40.622900
77.571100 50.224700
71.296200 40.458700
89.156600 51.620100
46.741800 31.515300
56.075100 39.676700
62.484000 37.099800
79.187500 46.268700
46.315600 42.068900
53.367800 31.391800
63.140900 44.440800
76.969000 48.216000
59.158300 46.894300
30.564000 19.083300
68.595500 42.243800
63.382600 39.672900
82.572100 45.343600
63.640200 44.246000
63.012100 34.253400
55.274400 41.399800
72.062800 42.325400
88.853200 54.515900

Original Ordering
61.959900 40.622900
79.917100 55.734800
76.152000 41.949700
30.564000 19.083300
57.258300 43.576000
63.012100 34.253400

50.868400 35.777000
72.166300 43.858900
51.694000 35.245500
53.367800 31.391800
68.875600 43.002000
58.531600 33.858600
52.254600 35.901300
62.484000 37.099800
54.437500 45.307600
57.969500 37.322600
64.563000 27.950100
67.591200 44.174300
55.274400 41.399800
81.470000 50.130900
70.881900 56.635200
52.074100 31.787700
89.156600 51.620100
79.087700 59.005500
55.884600 35.867000
76.969000 48.216000
57.400700 43.112200
66.799600 44.421400
77.571100 50.224700
73.697800 38.815700
56.075100 39.676700
60.318200 37.061000
37.776300 37.079600
61.619000 49.739100
70.389000 42.600100
79.187500 46.268700
72.062800 42.325400
55.149700 45.838000
56.821100 57.439100
79.004100 61.111900
71.296200 40.458700
63.140900 44.440800
31.046200 19.739200
34.420700 19.450600
56.038900 32.992800
63.382600 39.672900
40.674700 23.385700
57.157000 45.031300
84.108100 54.566600

59.158300 46.894300
68.595500 42.243800
60.975900 52.210500
88.853200 54.515900
53.533200 28.872100
42.916900 27.928900
82.572100 45.343600
46.741800 31.515300
56.512100 51.466900
63.640200 44.246000
46.315600 42.068900

Euclidean Ordering
64.563000 27.950100
73.697800 38.815700
76.152000 41.949700
82.572100 45.343600
71.296200 40.458700
63.012100 34.253400
72.062800 42.325400
79.187500 46.268700
70.389000 42.600100
72.166300 43.858900
68.595500 42.243800
89.156600 51.620100
62.484000 37.099800
68.875600 43.002000
76.969000 48.216000
81.470000 50.130900
58.531600 33.858600
63.382600 39.672900
60.318200 37.061000
53.533200 28.872100
67.591200 44.174300
56.038900 32.992800
77.571100 50.224700
66.799600 44.421400
88.853200 54.515900
61.959900 40.622900
57.969500 37.322600
53.367800 31.391800
84.108100 54.566600

55.884600 35.867000
63.640200 44.246000
52.074100 31.787700
63.140900 44.440800
56.075100 39.676700
79.917100 55.734800
52.254600 35.901300
51.694000 35.245500
57.400700 43.112200
50.868400 35.777000
55.274400 41.399800
57.258300 43.576000
46.741800 31.515300
57.157000 45.031300
59.158300 46.894300
79.087700 59.005500
61.619000 49.739100
42.916900 27.928900
70.881900 56.635200
40.674700 23.385700
55.149700 45.838000
54.437500 45.307600
79.004100 61.111900
60.975900 52.210500
56.512100 51.466900
46.315600 42.068900
34.420700 19.450600
31.046200 19.739200
56.821100 57.439100
30.564000 19.083300
37.776300 37.079600

Angular Ordering
64.563000 27.950100
73.697800 38.815700
53.533200 28.872100
63.012100 34.253400
82.572100 45.343600
76.152000 41.949700
34.420700 19.450600
71.296200 40.458700
40.674700 23.385700

58.531600 33.858600
89.156600 51.620100
79.187500 46.268700
72.062800 42.325400
53.367800 31.391800
56.038900 32.992800
62.484000 37.099800
70.389000 42.600100
72.166300 43.858900
52.074100 31.787700
88.853200 54.515900
60.318200 37.061000
81.470000 50.130900
68.595500 42.243800
68.875600 43.002000
30.564000 19.083300
63.382600 39.672900
76.969000 48.216000
31.046200 19.739200
55.884600 35.867000
57.969500 37.322600
77.571100 50.224700
84.108100 54.566600
42.916900 27.928900
67.591200 44.174300
61.959900 40.622900
66.799600 44.421400
46.741800 31.515300
51.694000 35.245500
52.254600 35.901300
63.640200 44.246000
79.917100 55.734800
50.868400 35.777000
63.140900 44.440800
56.075100 39.676700
79.087700 59.005500
55.274400 41.399800
57.400700 43.112200
57.258300 43.576000
79.004100 61.111900
57.157000 45.031300
59.158300 46.894300
70.881900 56.635200

61.619000 49.739100
55.149700 45.838000
54.437500 45.307600
60.975900 52.210500
46.315600 42.068900
56.512100 51.466900
37.776300 37.079600
56.821100 57.439100

PCA projected patterns in their original order:
-0.437097 -0.633755
22.818270 2.531440
12.264350 -7.102861
-38.485374 -2.042867
-2.830418 4.376021
-2.954554 -6.578506
-12.400724 1.203568
9.917644 -3.358022
-11.987376 0.312922
-12.634104 -3.838506
6.678826 -2.322144
-6.951574 -4.515842
-11.162953 0.567231
-1.878472 -3.890955
-4.287806 7.347783
-5.573916 -1.288256
-5.015222 -12.734035
6.220519 -0.644668
-5.670620 3.598233
21.133345 -3.034189
15.665379 8.124446
-13.515502 -2.812088
28.424717 -5.886806
23.866687 5.738651
-8.114069 -1.403145
16.306025 -2.244994
-2.958144 3.907968
5.683798 -0.012514
17.889071 -0.869725
8.514510 -8.438433
-5.915602 1.714041
-3.729251 -2.765429

-22.766455 9.306128
4.150372 7.251449
7.742654 -3.471132
17.139130 -5.076894
9.010045 -4.598424
-3.402353 7.415055
4.214415 16.323700
24.922591 7.563199
7.363946 -5.765731
2.602694 1.960621
-37.727143 -1.746543
-35.030152 -3.795148
-9.520868 -3.914270
0.256958 -2.197361
-27.641164 -3.814879
-2.137690 5.659878
25.734749 -0.697077
0.549709 6.163717
6.036651 -2.812995
4.928726 9.683641
29.717084 -3.277688
-13.841930 -6.056027
-23.316790 -1.175199
19.504243 -7.668722
-18.166802 -0.190441
0.759273 11.442650
2.920403 1.528986
-12.882659 8.954938
jlusby@engapps00:/mnt/home10/jlusby/machine Learning CSE 176/pa1/source$
jlusby@engapps00:/mnt/home10/jlusby/machine Learning CSE 176/pa1/source$

//What did I learn? I got a bit more familiar with c++ class syntax. Was already familiar with python and java syntax so it wasn't exactly a huge learning experience. I still don't fully understand how PCA works mathwise. I just used the functions to find the important values and put it together how the PCA definition said to (Reference 1 and 2). I got to play around with a debug macro which is something I learned about when I sat in on my friends OOP class. Other than that this really wasn't a difficult project or particularly ground breaking for me understanding-wise

//References
1 - https://docs.google.com/viewer?a=v&q=cache:CvPTvqJoCAAJ:www.ce.yildiz.edu.tr/personal/songul/file/1097/principal_components.pdf+&hl=en&gl=us&pid=bl&srcid=ADGEESiFY4dT-qVBKKl

_Di4gRgIbsnItxk5doeeARYvQrYQnmPoOSWCiH1wcH7KHnsLfN4rTZ0IUobAwpwvhm9gNDiMUIYRX0UWzrpt0pvwBQItiJuKhBQPNUC2I71_NMksQx3HmB2aT&sig=AHIEtbQZzwo22qJvMUszGiBAkyxFFOteXg
2 - http://en.wikipedia.org/wiki/Principal_component_analysis
3 - http://www.gnu.org/software/gsl/
4 - http://www.cplusplus.com/
5 - and other random links from www.google.com that helped point me in the right direction when figuring out which functions to use.

Done.
john@zakum:~$ ./start_next_assignment
//I apologize for any informalities in this report, I was just trying to entertain myself