X

```
Trial>> X=[-3,5,0;-3,4,-1;-4,0,-1;-1,-3,-3]

X =

   -3    5    0

   -3    4   -1

   -4    0   -1

   -1   -3   -3
```

Z

```
Trial>> mu=mean(X)

mu =

  -2.7500    1.5000   -1.2500


Trial>> [nrows,ncols]=size(X);

[nrows,ncols]

ans =

    4    3


Trial>> Z=zeros(nrows,ncols);

Trial>>  for row=1:nrows
                    for col=1:ncols
                                Z(row,col)=X(row,col)-mu(col);
                    end
          end
```

Trial>> Z

Z =

-0.2500 3.5000 1.2500

-0.2500 2.5000 0.2500

-1.2500 -1.5000 0.2500

1.7500 -4.5000 -1.7500

C

Trial>> c=cov(Z)

c =

1.5833 -2.5000 -1.2500

-2.5000 13.6667 4.1667

-1.2500 4.1667 1.5833

V

Trial>> [V,D]=eig(c);V

V =

-0.4125 0.8900 -0.1945

0.1969 0.2955 0.9348

-0.8894 -0.3473 0.2971

Is each column of V *an eigenvector? Or is it each row? Let us check .. (we will examine* D *in a little while).*

Trial>> (c*V(:,1))./(V(:,1))

ans =

0.0813

0.0813

0.0813

The fact that identical entries appear in the above result suggests that λ=0.0813 is a solution of c*V(:,1)= λ V(:,1). So the columns of V must be eigenvectors.

Trial>> (c*V(1,:))./(V(1,:))

Error using *

Inner matrix dimensions must agree.

Trial>> (c*V(1,:)')./(V(1,:)')

ans =

6.3878

13.9149

-20.1366

The fact that differing entries appear in the above result suggests that no λ can satisfy

c*V(1,:)'=λ V(1,:)'. So the rows of V cannot be eigenvectors.

P

Trial>> P=Z*V

P =

-0.3196	0.3777	3.6919
0.3730	0.4295	2.4600
-0.0021	-1.6426	-1.0849
-0.0513	0.8354	-5.0670

Note that the PCA process decorrelates the features (columns) in the original data i.e. cov(P) must be a diagonal matrix (or very nearly so)

Trial>> cov(P)

ans =

0.0813	0.0000	0.0000
0.0000	1.2410	-0.0000
0.0000	-0.0000	15.5110

Note that the features contained in P *are arranged in increasing order of importance. Examine the eigenvalues returned as the diagonal entries in* D

```
Trial>> D

D =

   0.0813         0         0

        0    1.2410         0

        0         0   15.5110
```

R

```
Trial>> R=P*V'

R =

  -0.2500    3.5000    1.2500

  -0.2500    2.5000    0.2500

  -1.2500   -1.5000    0.2500

   1.7500   -4.5000   -1.7500
```

```
Trial>> Xrecovered=zeros(nrows,ncols);

Trial>>  for row=1:nrows

                for col=1:ncols

                        Xrecovered(row,col)=R(row,col)+mu(col);

                end

        end

Trial>> Xrecovered

Xrecovered =

  -3.0000    5.0000    0.0000

  -3.0000    4.0000   -1.0000

  -4.0000   -0.0000   -1.0000

  -1.0000   -3.0000   -3.0000
```