```
% **Exercise 17** -- file 'PCAsup.m'
% Principal Component Analysis of LakeSuperiorAirTemp

% IMPORT FILE 'LakeSuperiorAirTemperature.dat.txt' AND
% RENAME THE RESULTING VARIABLE TO 'LakeSuperiorAirTemp'.

% The file contains hourly measurements of air temperature above Lake
% Superior at four locations:
%  Station-1: Lat 47.08 Lon -90.73
%  Station-2: Lat 47.87 Lon -89.31
%  Station-3: Lat 48.22 Lon -88.37
%  Station-4: Lat 47.18 Lon -87.22

% First, copy the data into separate variables
        aa = LakeSuperiorAirTemp(:,1);
        bb = LakeSuperiorAirTemp(:,2);
        cc = LakeSuperiorAirTemp(:,3);
        dd = LakeSuperiorAirTemp(:,4);

% Remove the NaN's via interpolation
% In this example, I'm using my very simple interpolation method:
        va = sergeiinterpolate(aa);
        vb = sergeiinterpolate(bb);
        vc = sergeiinterpolate(cc);
        vd = sergeiinterpolate(dd);

        plot([va vb vc vd])
%%
% ----------- BEFORE YOU BEGIN THE ANALYSIS ------------
% Look at the data. Think about how the PCA analysis could be useful here.
% Ask yourself a series of questions.
% What factors do you think may generate variance in this data? (Come up with at least 3)
% (Variance here is anything that departs from the average temperature for the dataset.)
% How many of these factors are independent?
% ------------------------------------------------------
%%
% Now let's proceed. Choose a record with least sinister gaps.
% You can later play with these numbers
        % mybegin=7000; myend=22000;
        % mybegin=3000; myend=10000;

        mybegin=8000; myend=10000;
        va = va(mybegin:myend);
        vb = vb(mybegin:myend);
        vc = vc(mybegin:myend);
        vd = vd(mybegin:myend);
        plot([va vb vc vd])
%%
% Make a scatter plot to see the dispersion of data
% (We don't have a 4-dimensional space handy, so have to settle for 3D)

        plot3(va,vb,vc,'.');grid on

% Rotate the plot and investigate the dimensionality of the data.
%%
% Subtract off the mean from each vector.
```

```
% If there was a sensible and easy trend, I'd subtract that too.
% Stuff them all into a n row x 4 column matrix
        mymatrix = [va-mean(va), vb-mean(vb), vc-mean(vc), vd-mean(vd)];
        size(mymatrix)
        sr=mymatrix;       % here, 'sr' is the variable used in the analysis

% Create labels
        categories={'St1' 'St2' 'St3' 'St4'};
%%
% Get a quick impression about the data
        boxplot(mymatrix,'orientation','horizontal','labels',categories)
% You could also use 'plot' to compare pairs of variables,

%%
% Get a quick idea of correlations

        corr(mymatrix)

%%
% ---- SKIP THIS STEP ON THE FIRST RUN ----
% If the four datasets were very different, we'd need to standardize their
% variance. Since they are not, on the first pass it's possible to retain
% the original scale and units. You can run this cell later and compare the
% results.
% Divide the data by the corresponding standard deviations
        stdr = std(mymatrix);
        sr = mymatrix./repmat(stdr,max(size(mymatrix)),1);
% The standardized rankings are now in variable 'sr'
        boxplot(sr,'orientation','horizontal','labels',categories)
% -------------------------------------------
%%

% Now find the principal components!
        [coefs,scores,variances,t2] = princomp(sr);
%%
% First, look at vectors of principal component coefficients

        coefs

% The first column is the first principal component
% Note the weights (loadings) for each station
%%
% Component scores (variable 'scores') are the original data that have been
% mapped into the new variables, i.e. projected on principal components.
% Projection on the first two (most significant) principal components:
        plot(scores(:,1),scores(:,2),'+')
        xlabel('1st Principal Component');
        ylabel('2nd Principal Component');
%%
% Using the 'variances' output, calculate the percent of variance
% in the data explained by each principal component
        percent_explained = 100*variances/sum(variances)
        pareto(percent_explained)
        xlabel('Principal Component')
        ylabel('Variance Explained (%)')
%%
```

```
% Visualize the results of the principal component analysis

        biplot(coefs(:,1:2), 'scores',scores(:,1:2),'varlabels',categories);

% Each of the variables is represented in this plot by a vector, and
% the direction and length of the vector indicates how each variable
% contributes to the two principal components in the plot.
% What can you tell about the nature of the first PC?
%%
% Let's visualize the principal components in another way.
% Plot the data along the principal components against time
        hold off
        plot(scores)
        %hold on
        %plot(-sr(:,1),'black')
% Doesn't this look like the first PC is seasonal variation?
% Of course, you knew this from the beginning, right?
% Is it only seasonal?
%%
% But what is the second PC? Daily cycle? Latitude?
% Use Google Earth or Google Maps to find out the locations of the four stations.
% (Both accept comma-separated (Lat,Lon) pairs.
% Compare the latitudes against the biplot. Is it now clearer?
% Let's compare the magnitudes of the second and third components

        plot(scores(:,2:3))
%%
% Hm, about the same... But can they really be reflections of the same
% underlying thing? Well, no! They are orthogonal by construction, remember?

        corr(scores)

%%
% Plot the second and third PCs on a biplot.

        biplot(coefs(:,2:3), 'scores',scores(:,2:3),'varlabels',categories);

% Does this give you a clue? Compare to the Google Earth map.
% (This is the part that I find most amazing about this analysis.)
% Explain what you see.

% Additional food for thought: Could daily cycle be a separate PC?
% If it were, how would the contributions from the four factors look like?
% For example, could any two of them have different signs?
% Or would all four stations contribute to the component in the same way?
% If so, wouldn't the daily component be included in the first PC?
% Check if you can see the daily cycle in the first principal component.
% We have 2000 hours (data points) in the series, that's 83 days.
% Compare the first PC against the actual temperature data.

% Repeat the analysis using another part of the series (or a longer
% series). Do you get the same results?
```