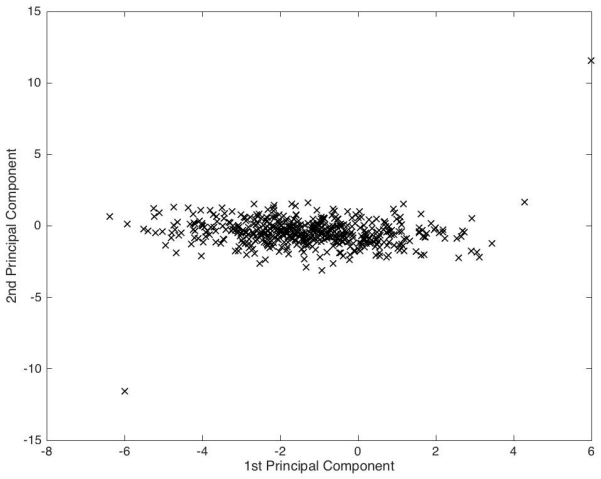
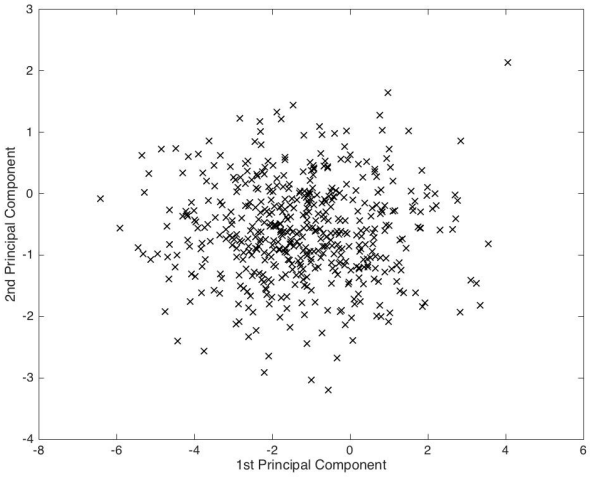
```
function ueb03a01()

    clear all;

    % a)

    data = dlmread('pca2.csv',',',1,0);

    [vectors,values] = pca_selfmade(data);
    data_translated = data*vectors;

    figure;
    plot(data_translated(:,1),data_translated(:,2),'xk');
    xlabel('1st Principal Component');
    ylabel('2nd Principal Component');
```

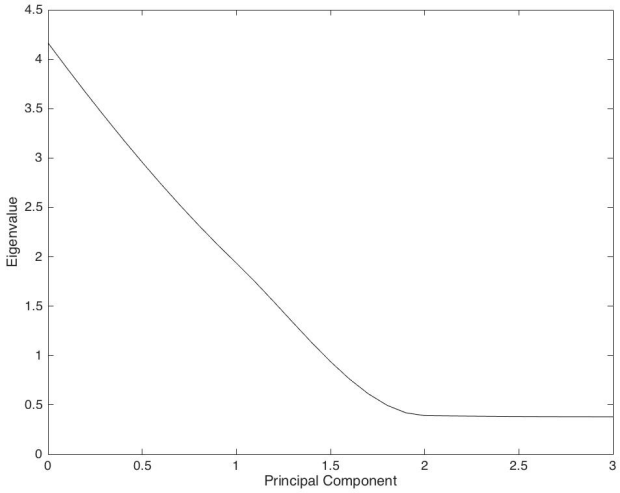



```
    % b)

    data_truncated = data([1:16,18:156,158:end],:);

    [vectors,values] = pca_selfmade(data_truncated);
    data_truncated_translated = data_truncated*vectors;

    figure;
    plot(data_truncated_translated(:,1),data_truncated_translated(:,2),'xk');
    xlabel('1st Principal Component');
    ylabel('2nd Principal Component');
```



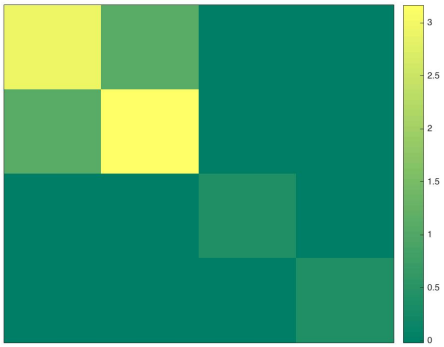

```
end
```

```
function ueb31a02()

    clear all;

    % a)

    data = dlmread('pca4.csv',',',1,0);

    outlierIndexes = [];

    for k = 1:size(data,2)

        variable = data(:,k);

        % compute the median absolute difference
        meanValue = mean(variable);
        % compute the absolute differences
        absoluteDeviation = abs(variable - meanValue);
        % compute the median of the absolute differences
        mad = median(absoluteDeviation);

        % if the absolute difference is more than some factor times the mad value it's an outlier
        sensitivityFactor = 9;
        thresholdValue = sensitivityFactor * mad;
        outlierIndexes = [outlierIndexes ; find(abs(absoluteDeviation) > thresholdValue)];

    end

    % b)

    data_truncated = data;
    sort(outlierIndexes,'descend');

    for k = 1:size(outlierIndexes)

        % remove outliers
        data_truncated = data_truncated([1:(outlierIndexes(k)-1),(outlierIndexes(k)+1):end],:);

    end

    [vectors,values] = pca_selfmade(data_truncated);
    screeplot(values);
```

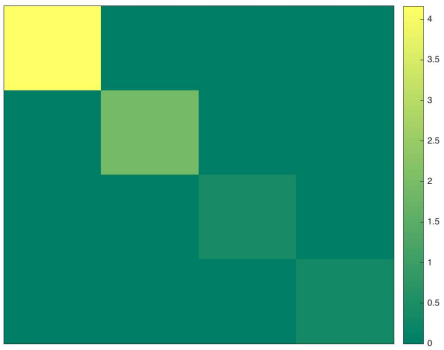



```
    % c)

    data_truncated_centered =
               data_truncated-repmat(mean(data_truncated),[size(data_truncated,1),1]);
    [vectors,values] = pca_selfmade(data_truncated_centered);

    data_whitened = data_truncated_centered*vectors*diag(values.^(-1/2));
```

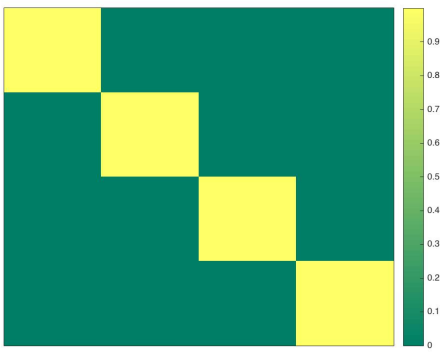
```
% d) i)

figure;
colormap(summer);
heatmap(cov(data_truncated_centered));
colorbar;
```




```
% d) ii)

data_translated = data_truncated_centered*vectors;

figure;
colormap(summer);
heatmap(cov(data_translated));
colorbar;
```

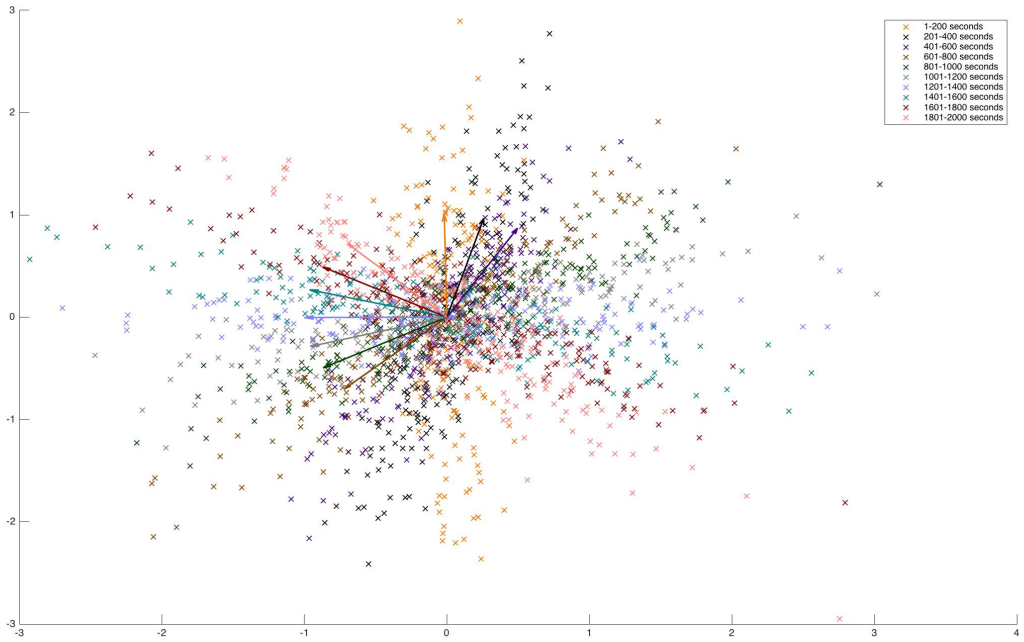



```
% d) iii)
figure;
colormap(summer);
heatmap(cov(data_whitened));
colorbar;
```

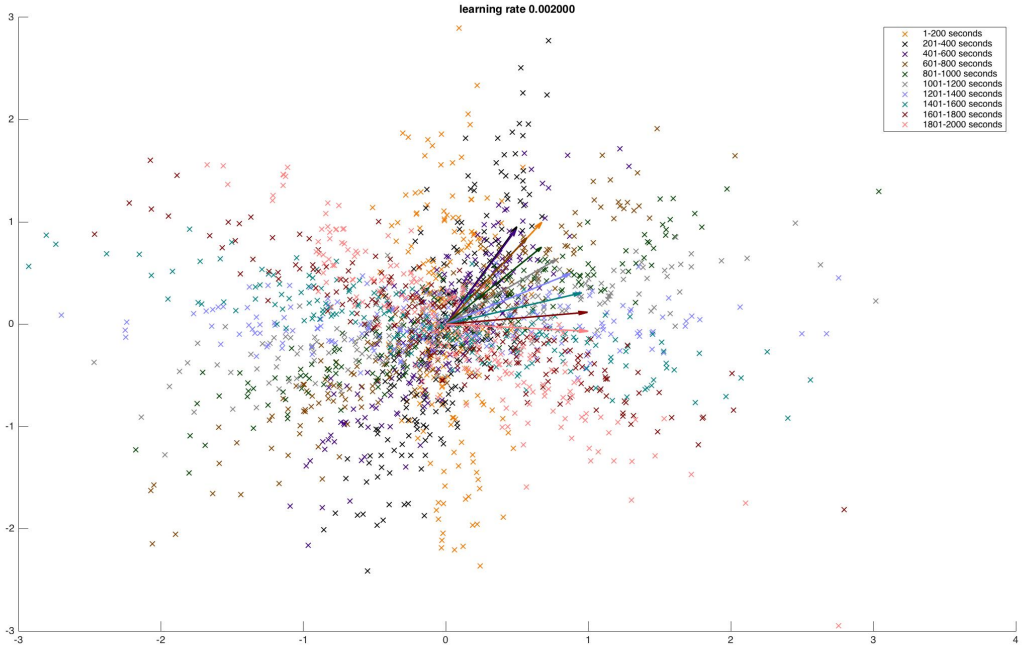


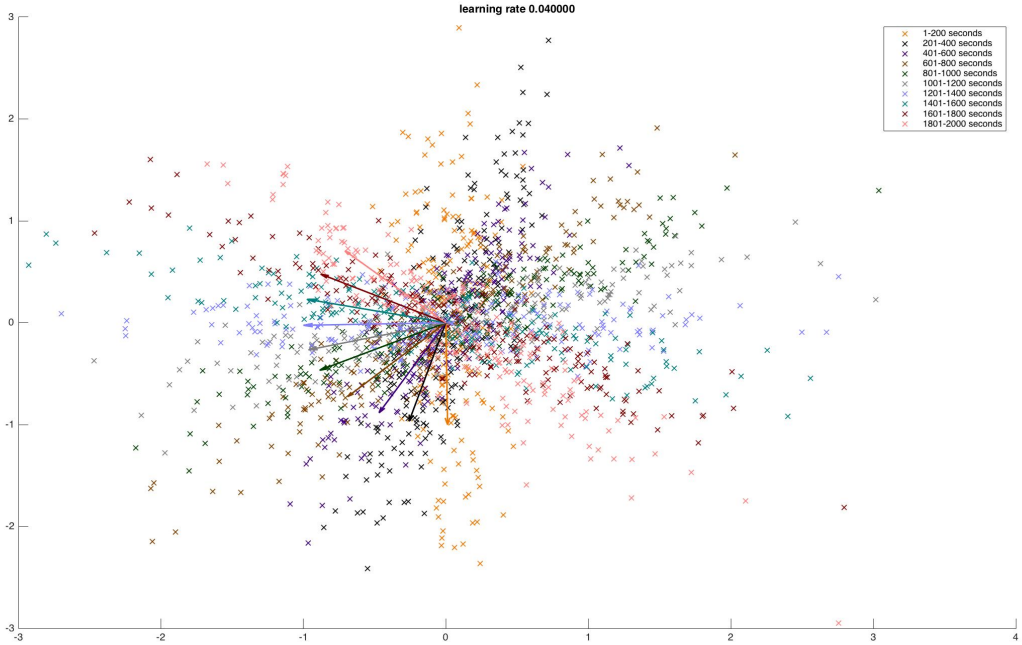

```
end
```

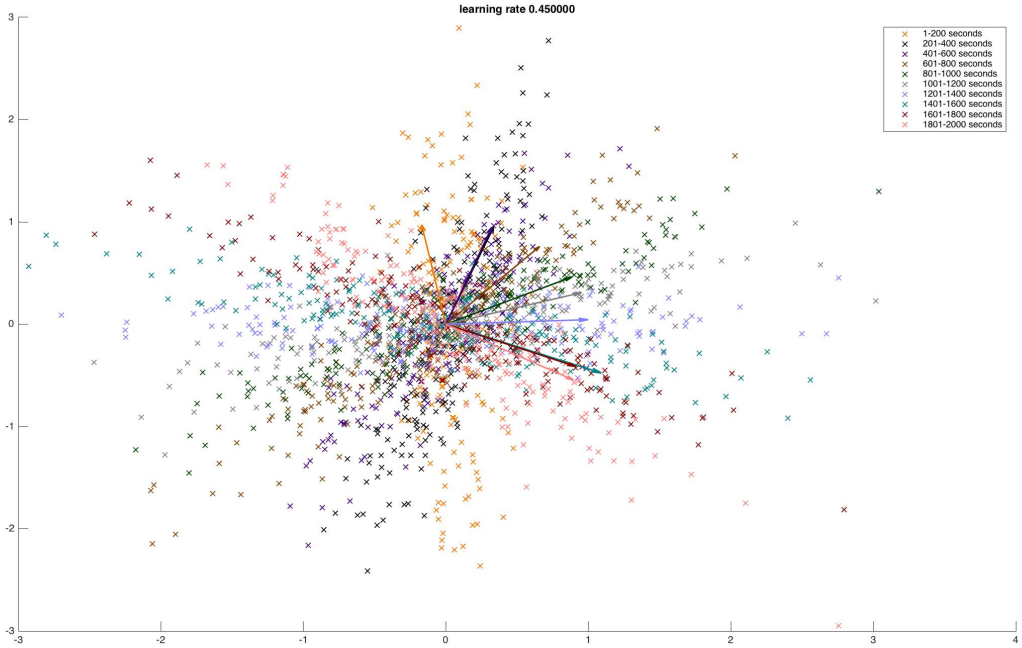
```
function ueb03a04()

    clear all;

    color_selection = [0.9412 0.4706 0 ; 0 0 0 ; 0.251 0 0.502 ; 0.502 0.251 0 ; 0 0.251 0 ;
                0.502 0.502 0.502 ; 0.502 0.502 1 ; 0 0.502 0.502 ; 0.502 0 0 ; 1 0.502 0.502];

    % 1.

    data = dlmread('data-onlinePCA.txt',',',1,1);

    plot_data(data,color_selection);

    % 2.

    plot_vectors(data,color_selection);
```




```
    % 3.

    learning_rate = [.002 ; .04 ; .45];
    w = [1 , 1];
    output = zeros(size(data));

    for k = 1:size(learning_rate)

        plot_data(data,color_selection);

        for l = 1:size(data,1)

            s = w * data(l,:)';
            w = w + learning_rate(k) * s * (data(l,:) - (s * w));

            if (mod(l,200) == 0)
                plot_arrow(w,color_selection(l/200,:));
            end

        end

    end

end
```


learning rate 0.002000
1-200 seconds
201-400 seconds
401-600 seconds
601-800 seconds
801-1000 seconds
1001-1200 seconds
1201-1400 seconds
1401-1600 seconds
1601-1800 seconds
1801-2000 seconds


learning rate 0.040000
1-200 seconds
201-400 seconds
401-600 seconds
601-800 seconds
801-1000 seconds
1001-1200 seconds
1201-1400 seconds
1401-1600 seconds
1601-1800 seconds
1801-2000 seconds


learning rate 0.450000
1-200 seconds
201-400 seconds
401-600 seconds
601-800 seconds
801-1000 seconds
1001-1200 seconds
1201-1400 seconds
1401-1600 seconds
1601-1800 seconds
1801-2000 seconds

```
function plot_data(data,color_selection)

    figure;
    hold on;

    for k = 1:10
        block = data(((k-1)*200+1):(k*200),:);
        plot(block(:,1),block(:,2),'Color',color_selection(k,:),'Marker','x','LineStyle','none');
    end

    legend('1-200 seconds', '201-400 seconds', '401-600 seconds', '601-800 seconds', '801-1000
seconds', '1001-1200 seconds', '1201-1400 seconds', '1401-1600 seconds', '1601-1800 seconds',
'1801-2000 seconds');

end

function plot_vectors(data,color_selection)

    for k = 1:10
        [vectors,values] = pca_selfmade(data(((k-1)*200+1):(k*200),:));
        plot_arrow(vectors,color_selection(k,:));
    end

end

function plot_arrow(vector,color)

    ursprung = [0;0];
    quiver(ursprung(1),ursprung(2),vector(1),vector(2),0,'Color',color,'LineWidth',1.5);

end

function [vectors,values] = pca_selfmade(data)

    [d,v] = eigs(cov(data));

    vectors = fliplr(d);
    values = nonzeros(fliplr(v));

end

function screeplot(values)

    X = 0:3;
    Xi = 0:.1:3;
    Y = pchip(X,values,Xi);

    figure;
    plot(Xi,Y,'-k');
    xlabel('Principal Component');
    ylabel('Eigenvalue');

end
```