

#Max Reinhard - 359417

#Ozge Sahin 389691

#Vincent Ulitzsch - 365672

ANSWERS TO QUESTIONS

```
1 ) def pca(X,ncomp=10):
    """ Principal Component Analysis
        INPUT:      X      - DxN array of N data points with D features
                   ncomp  - number of principal components to estimate
        OUTPUT: W      - D x ncomp array of directions of maximal variance,
                   sorted by their eigenvalues
                   H      - ncomp x N array of projected data """

    ncomp = min(np.hstack((X.shape, ncomp)))

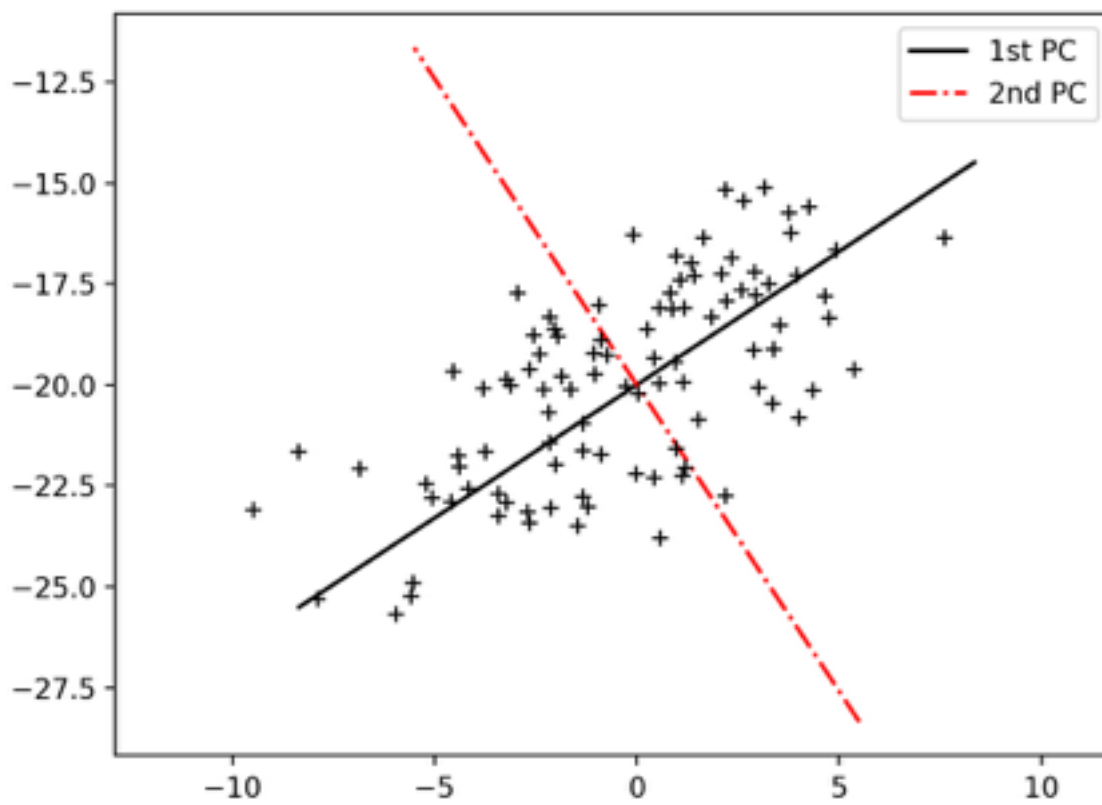
    #center the data
    # ...

    mu =(sp.mean(X,axis= 1)).T
    mu = mu[:,sp.newaxis]
    X=X-(mu)

    # compute linear kernel
    #...
    K = (X.T).dot(X)
    # compute eigenvectors and sort them according to their eigenvalues
    # ...
    eval,evalvec = np.linalg.eig(K)
    idx = np.argsort(eval)[::-1]
    MaxEigenV = evalvec[:,idx[0:ncomp]]
    # compute W and H
    W = X.dot(MaxEigenV)
    H = (W.T).dot(X)
    return W,H
    #...
```

2) Angles of the two PCA components are changed. This time direction of second PCA component becomes dominated appears as first PCA component when we use non-centered data.

a) Graph of PCA with centered data :



b) Graph of PCA with non-centered data :

