

Homework 3

1. Screenshot of 50 MSE numbers

0N	1N	2N	3N	4N	0c	1c	2c	3c	4c
4.542470667	0.383450312	0.175563	0.141783648	0.160838362	4.543119029	0.384613534	0.177815283	0.144440506	0.160838362
4.542470667	0.641093185	0.715628488	0.908392907	1.115657858	4.549538993	0.648642108	0.750621129	0.941972819	1.115657858
4.542470667	1.290372451	1.967240392	2.650841135	3.653279733	4.557472964	1.323462148	2.119748049	3.02737992	3.653279733
4.542470667	0.799942744	0.828082555	0.984949768	1.194	4.566198667	0.840614157	1.207089797	1.271191967	1.194
4.542470667	1.91776775	3.331722104	4.548257197	5.139266667	4.919928	2.835679428	4.651434503	4.971247272	5.139266667

2. Screenshot of code:

```

def reconstruct(data_in, data_mean, iris_cov, useNoiseless):
    # NORMALIZATION
    data_mean_repeat = np.tile(data_mean, [samples, 1])
    data_norm = (data_in - data_mean_repeat)

    # COVARIANCE
    if(useNoiseless == False):      # c
        data_cov = np.cov(data_norm, rowvar=0)
    else:                            # n
        data_cov = iris_cov

    # EIGENVALUE, EIGENVECTOR
    data_eigval, data_eigvec = np.linalg.eig(data_cov)
    index = data_eigval.argsort()[::-1]
    data_eigval = data_eigval[index]
    data_eigvec = data_eigvec[:, index]

    # REDUCE DIMENSION AND RECONSTRUCTION
    data_reconstructions = []
    for dim in range(dims + 1):
        data_feature = data_eigvec[:, :dim].reshape(dims, dim)
        data_reduce = np.dot(data_norm, data_feature).reshape(samples, dim)
        data_reconstruction = np.dot(data_reduce, data_feature.T) + data_mean_repeat
        data_reconstructions.append(data_reconstruction)

    return data_reconstructions

def MSE(x1, x2):
    mse = []
    for dim in range(dims+1):
        square = pow((x1[dim] - x2), 2)
        mean = np.sum(square)/samples
        mse.append(mean)
    return mse

```

```

if __name__ == '__main__':
    mse = []

    # iris
    iris_in = readInData('./Data/iris.csv')
    iris_mean = getMean(iris_in)
    iris_mean_repeat = np.tile(iris_mean, [samples, 1])
    iris_norm = (iris_in - iris_mean_repeat)
    iris_cov = np.cov(iris_norm, rowvar=0)

    # dataI
    mse_I = []
    dataI_in = readInData('./Data/dataI.csv')
    dataI_mean = getMean(dataI_in)
    # dataI + n
    dataI_n_reconstructions = reconstruct(dataI_in, iris_mean, iris_cov, True)
    dataI_n_mse = MSE(dataI_n_reconstructions, iris_in)
    mse_I.append(dataI_n_mse)
    # dataI + c
    dataI_c_reconstructions = reconstruct(dataI_in, dataI_mean, iris_cov, False)
    dataI_c_mse = MSE(dataI_c_reconstructions, iris_in)
    mse_I.append(dataI_c_mse)
    # mse
    mse.append(mse_I)

    # dataXX

```