```python
import numpy
import scipy
import scipy.spatial
import scipy.sparse
import matplotlib
from matplotlib import pyplot as plt

def cc_fit_predict(self,X):

    D = scipy.spatial.distance.cdist(X,X)
    A = (D < self.delta)*1.0
    n_components, Y = scipy.sparse.csgraph.connected_components(A)
    return n_components, Y


def correspondence(names,T,n_components,Y):

    s = f"{'':10s}"
    for j in range(n_components):
        s += '  cluster_'+str(j)
    print(s)

    for i,name in enumerate(names):
        s = f"{name:10s}"
        for j in range(n_components):
            count = (((T==i)*(Y==j)).sum())
            s += f"{count:11d}"
        print(s)


def kmeans_fit(self,X):

    self.cluster_centers_ = X[:self.n_clusters]*1.0

    for t in range(100):
        D = scipy.spatial.distance.cdist(X,self.cluster_centers_)
        C = numpy.argmin(D,axis=1)
        C_hot = numpy.eye(self.n_clusters)[C]
        w_hot = C_hot / (C_hot.sum(axis=0) + 1e-9)
        self.cluster_centers_ = w_hot.T.dot(X)


def view_cluster_centers(model):

    prototypes = model.cluster_centers_
    mosaic = prototypes.reshape(2,10,28,28).transpose(0,2,1,3).reshape(2*28,10*28)

    plt.figure(figsize=(10,2))
    plt.axis('off')
    plt.imshow(mosaic,cmap='gray',vmin=0,vmax=1)


def analyze_variance(X,model):

    Y = model.predict(X)
    centers = model.cluster_centers_

    var_tot     = numpy.var(X,axis=0).sum()
    var_between = numpy.var(X*0 + centers[Y],axis=0).sum()
    var_within  = numpy.var(X - centers[Y],axis=0).sum()
    var_expl_f  = var_between / var_tot

    print(f"Total variance           {var_tot:8.3f}")
    print(f"Between-cluster variance {var_between:8.3f}")
    print(f"Within-cluster variance  {var_within:8.3f}")
    print('')
    print(f"Explained variance (%)   {var_expl_f*100:7.2f}%")
```