Übung 8 PCA - Rainier Robles & Valentin Wolf

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
train = pd.read_table('zip.train', delim_whitespace=True,header=None)
test = pd.read_table('zip.test', delim_whitespace=True,header=None)
data = np.concatenate((train,test))
#split labels y from data X
y = pd.DataFrame(data)[0].as_matrix()
X = pd.DataFrame(data).drop(0, axis=1).as_matrix()
```

In [4]:

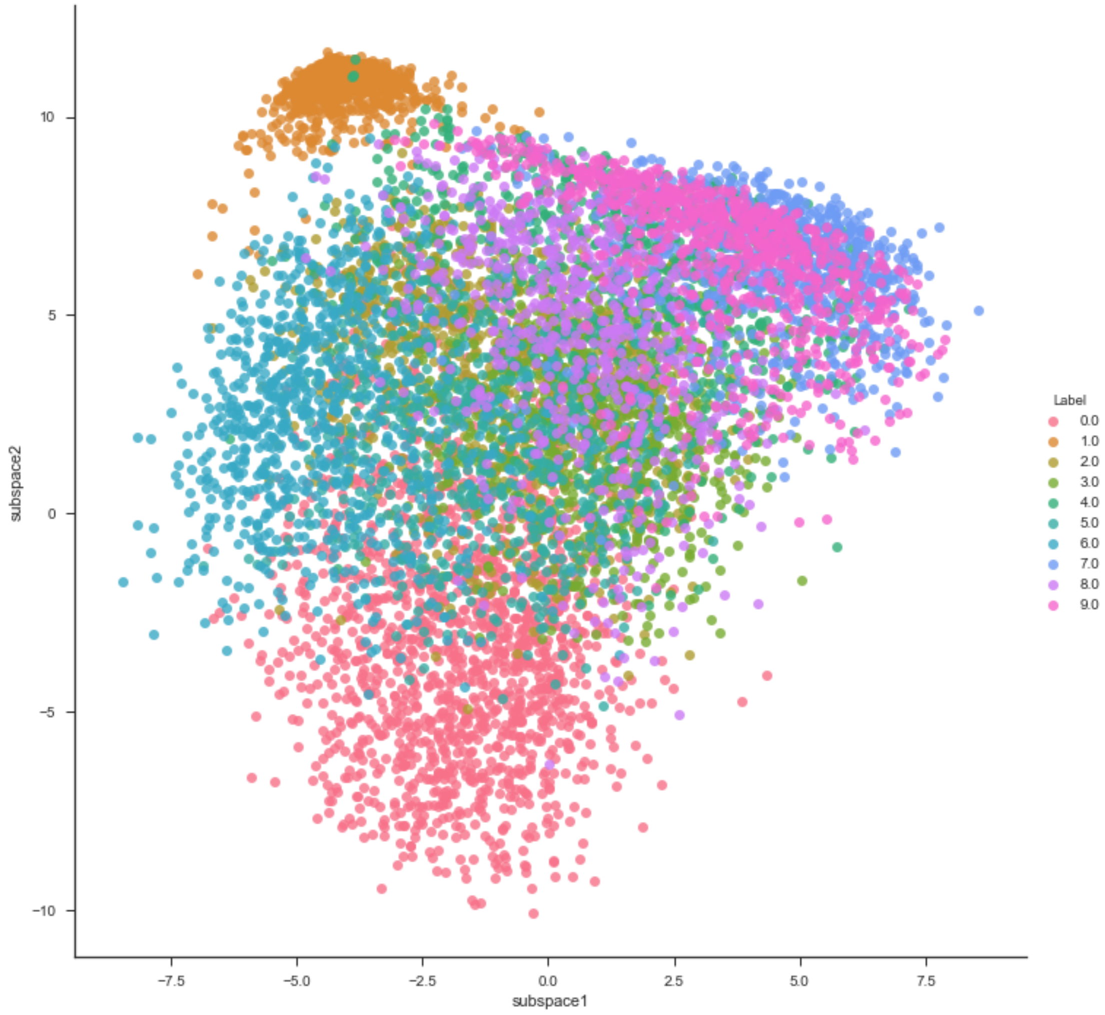
```
from scipy.linalg import eig

class PCA():
    def fit(self,X,dims=10):

        #Standardization -- weird results
        #N,features = X.shape
        #for i in range(features-1):
        #    X[i] = (X[i] - X[i].mean()) / np.sqrt(X[i].var())


        covariance = np.cov(X,rowvar=False)
        print("cov finished")
        w,v = eig(covariance)#np.linalg.eig(covariance)
        print("eig finished")
        highest = w.argpartition(-dims)[-dims:]
        print("found biggest eig")
        self.big_v = v[:,highest]
        print("selected vecs")
        #self.big_w_normalized =  (w[highest] - w[highest].mean()) / np.sqrt(w[
highest].var())
        return self.big_v#, self.big_w_normalized

    def transform(self,x):
        return x.dot((self.big_v))
```

In [5]:

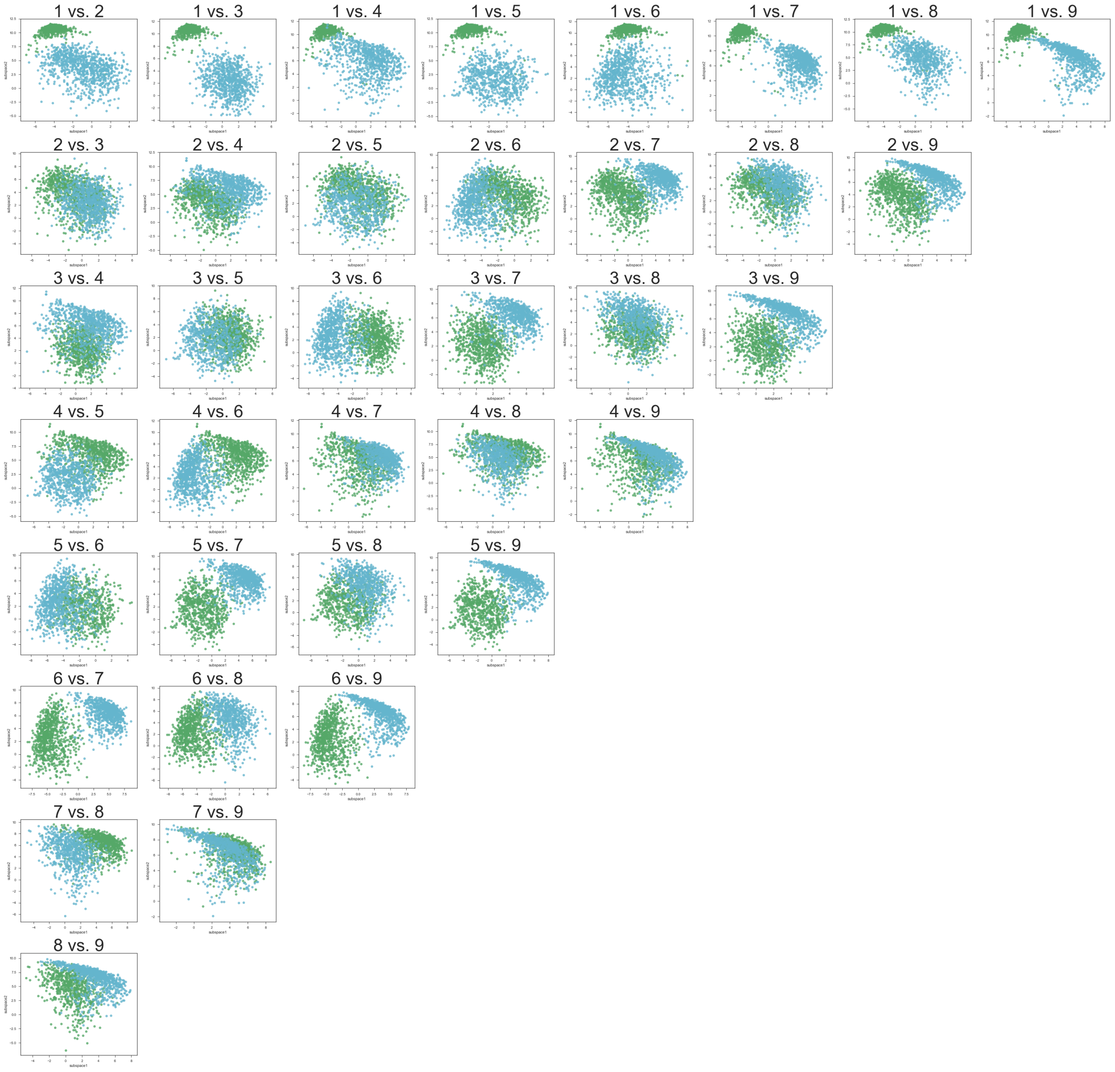
```
zip_PCA = PCA()
v= zip_PCA.fit(X,dims=2)
reduced = zip_PCA.transform(X)
```

```
cov finished
eig finished
found biggest eig
selected vecs
```

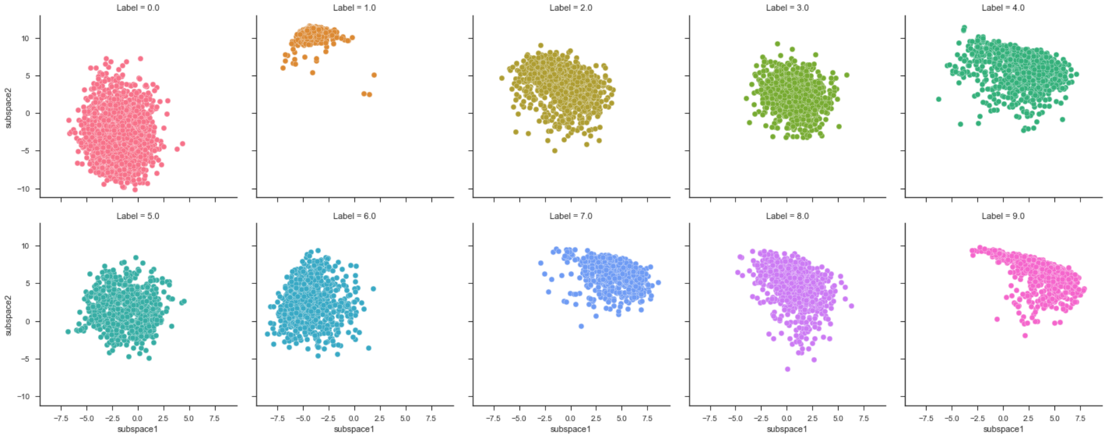
```
In [6]: subspaces_df = pd.DataFrame(reduced,columns=["subspace1","subspace2"])
        y_df = pd.DataFrame(y,columns=["Label"])
        df = subspaces_df.join(y_df)
        sns.set(style="ticks", color_codes=True)

        sns.lmplot(x="subspace1",y="subspace2", data=df, hue='Label', fit_reg=False,siz
        e=10)

Out[6]: <seaborn.axisgrid.FacetGrid at 0x10ce5bf98>
```



```
In [7]: nplots = 10
        plt.subplots(squeeze=False, figsize=(5*(nplots-1), 6*(nplots-2)))
        for i in range(nplots-1):

            for j in range(i+1,nplots):
                bin_dat1 = df[(y==i)]
                bin_dat2 = df[(y==j)]

                try:
                    curr_axis = plt.subplot2grid((nplots-1, nplots-2), (i, j-1-i))
                except TypeError:
                    curr_axis = axes

                sns.regplot(ax=curr_axis,x="subspace1",y="subspace2", data=bin_dat1, fi
        t_reg=False,scatter=True,color='g')
                sns.regplot(ax=curr_axis,x="subspace1",y="subspace2", data=bin_dat2, fi
        t_reg=False,scatter=True,color='c')
                curr_axis.set_title(str(i) + " vs. " + str(j),fontsize= 50)

        plt.tight_layout()
```



In [8]:
```
g = sns.FacetGrid(data=df, col="Label", hue="Label",size=4,col_wrap=5)
g = g.map(plt.scatter, "subspace1", "subspace2",edgecolor="w")
```



Aufgabe 2 - Eigenfaces

In [9]:
```
import PIL
from os import listdir
import matplotlib.image as mpimg

def show_img(img):
    plt.figure(figsize=(3, 3))
    plt.imshow(img,cmap='gray')
    plt.show()
```

In [10]:
```
directory = "/Users/valentinwolf/Documents/Studium/Machine Learning/Übung 8/lfw
crop_grey/faces/"
npics = len(listdir(directory))
rows = cols = 64
```

In [11]:
```
# Reading the pictures and saving to csv. only once
#
#pics = np.zeros((rows*cols,npics))
#
#for i in range(npics):
#    #print("reading ", i)
#    j = listdir(directory)[i]
#    pics[:,i] = mpimg.imread(directory + j).reshape(rows * cols)
#
#np.savetxt("faces.csv", pics, delimiter=",")
```

In [12]:
```
all_pics = pd.read_csv("faces.csv", index_col=False, header=None).as_matrix()
```

In [13]:
```
# Limiting the ampunt of picures as it gety very slow/inaccurate (returning com
plex numbers)
# to calculate ht eigenvectors
pics = all_pics[:,:3000]
```

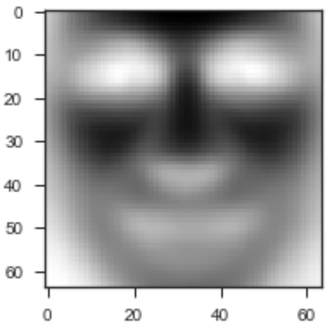
In [14]:
```
eigenfaces = PCA()
x = eigenfaces.fit(pics,dims=10)
```

```
cov finished
eig finished
found biggest eig
selected vecs
```

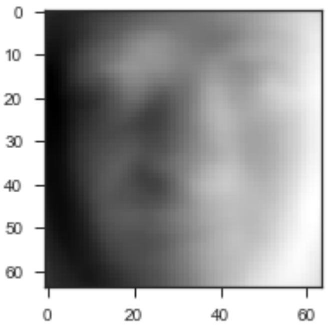
In [15]:

```
ef_subsp = eigenfaces.transform(pics)

for i in range(3):#len(ef_subsp[0])):
    print("Eigenface ", i)
    im2 = ef_subsp[:,-(i+1)].reshape(64,64)
    show_img(im2)
```

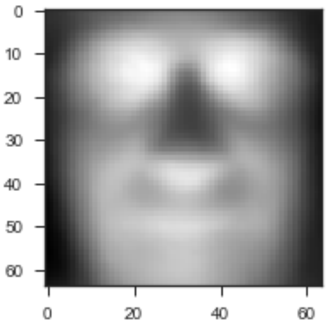
```
Eigenface  0
```



```
Eigenface  1
```

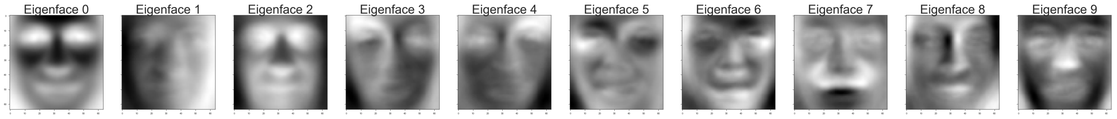


```
Eigenface  2
```



In [16]:

```
fig, axes = plt.subplots(ncols=len(ef_subsp[0]), sharey=True ,figsize=(len(ef_s
ubsp[0])*8, 7))
for i in range(len(ef_subsp[0])):
    #print("Eigenface ", i)
    im2 = ef_subsp[:,-(i+1)].reshape(64,64)
    axes[i].imshow(im2,cmap='gray',)
    axes[i].set_title("Eigenface " + str(i) ,fontsize= 50)
#plt.subplots(squeeze=False, figsize=(5*(nplots-1), 6*(nplots-2)))
```


Eigenface 0
Eigenface 1
Eigenface 2
Eigenface 3
Eigenface 4
Eigenface 5
Eigenface 6
Eigenface 7
Eigenface 8
Eigenface 9