Homework Week 11

```python
In [ ]:  import numpy as np
         import matplotlib.pyplot as plt
         from scipy.linalg import eigh
         np.random.seed(1)
```

```python
In [ ]:  def data_generation1(n=100):
             return np.concatenate([np.random.randn(n, 1) * 2, np.random.randn(n, 1)], axis=
         
         def data_generation2(n=100):
             return np.concatenate([np.random.randn(n, 1) * 2, 2 * np.round(np.random.rand(
```

```python
In [ ]:  def pca(x, n_components=1):
             x = x - np.mean(x, axis=0)
             w, v = np.linalg.eig(x.T.dot(x))
         
             return w[:n_components], v[:n_components, :]
```

```python
In [ ]:  def llp(x, n_components):
             x = x - np.mean(x, axis=0)
             W = np.exp(-np.sum((x[:, None] - x[None]) ** 2, axis=2))
         
             D = np.diag(np.sum(W, axis=1))
             L = D - W
             z = x.T @ D @ x
             z = (z + z.T) / 2
         
             w, v = eigh(x.T @ L @ x, z, eigvals_only=False)
         
             return w[:n_components], v[:n_components, :]
```
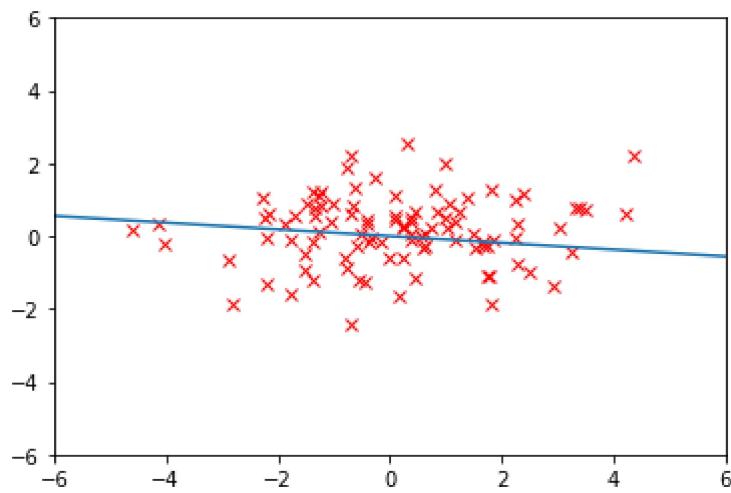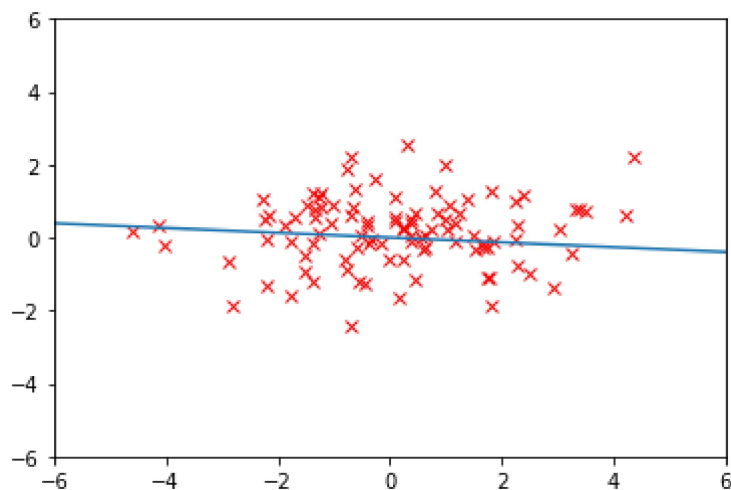
Data 1

LLP

```python
In [ ]:  n = 100
         n_components = 1
         x = data_generation1(n)
         #x = data_generation2(n)
         w, v = llp(x, n_components)
         #w, v = pca(x, n_components)
         plt.xlim(-6., 6.); plt.ylim(-6., 6.)
         plt.plot(x[:, 0], x[:, 1], 'rx')
         plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
         plt.show()
         #plt.savefig('example.png')
```
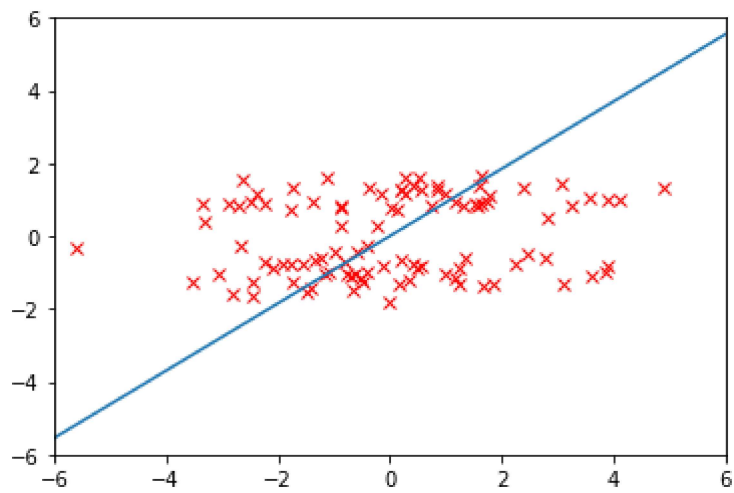
PCA

```
In [ ]:  w, v = pca(x, n_components)
         plt.xlim(-6., 6.); plt.ylim(-6., 6.)
         plt.plot(x[:, 0], x[:, 1], 'rx')
         plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
         plt.show()
         #plt.savefig('example.png')
```


Data 2

LLP

```
In [ ]:  x = data_generation2(n)
         w, v = llp(x, n_components)
         #w, v = pca(x, n_components)
         plt.xlim(-6., 6.); plt.ylim(-6., 6.)
         plt.plot(x[:, 0], x[:, 1], 'rx')
         plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
         plt.show()
         #plt.savefig('example.png')
```

PCA

```
In [ ]:  w, v = pca(x, n_components)
         plt.xlim(-6., 6.); plt.ylim(-6., 6.)
         plt.plot(x[:, 0], x[:, 1], 'rx')
         plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
         plt.show()
         #plt.savefig('example.png')
```