Table of Contents

In [1]:
```
# import autograd-wrapped numpy
import autograd.numpy as np

# datapath to data
datapath = '../mlrefined_datasets/unsuperlearn_datasets/'
```

Exercise 8.2. Encoding data

In [2]:
```
# load in dataset
csvname = datapath + '2d_span_data.csv'
x = np.loadtxt(csvname, delimiter = ',')

print(np.shape(x))
```

```
(2, 50)
```

Exercise 8.4. Nonconvexity of the linear Autoencoder

In [3]:
```
# load in dataset
csvname = datapath + '2d_span_data_centered.csv'
x = np.loadtxt(csvname, delimiter = ',')

print(np.shape(x))
```

```
(2, 50)
```

Exercise 8.5. Minimizing the linear Autoencoder over a toy dataset

In [4]:
```
# load in dataset
csvname = datapath + '2d_span_data_centered.csv'
x = np.loadtxt(csvname, delimiter = ',')

print(np.shape(x))
```

```
(2, 50)
```

Exercise 8.6. Producing a PCA basis

In [5]:
```
# load in dataset
csvname = datapath + '2d_span_data.csv'
x = np.loadtxt(csvname, delimiter = ',')

print(np.shape(x))
```

```
(2, 50)
```

Exercise 8.7. A warning example

In [6]:
```
# load in dataset
csvname = datapath + 'PCA_class_data.csv'
data = np.loadtxt(csvname,delimiter = ',');
x = data[:,:2]
y = data[:,-1]

print(np.shape(x))
print(np.shape(y))
```

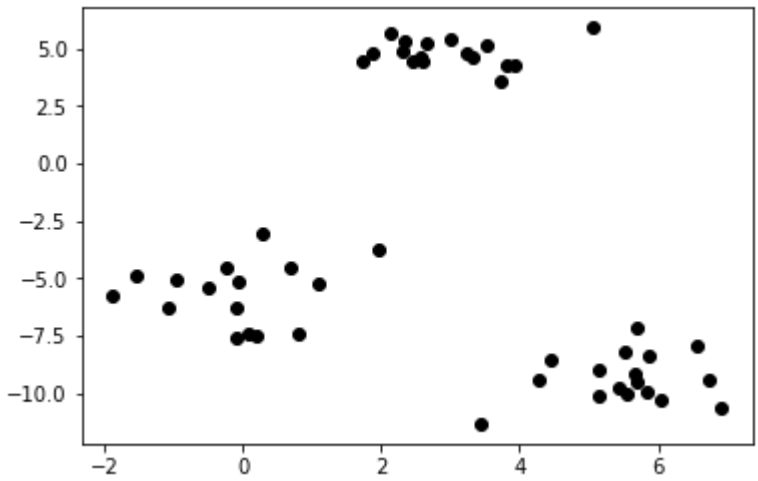
```
(23, 2)
(23,)
```

Exercise 8.8. Perform K-Means

In [7]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
%matplotlib inline

# Loading the data
P = 50 # Number of data points
blobs = datasets.make_blobs(n_samples=P,centers = 3,random_state=1
data = np.transpose(blobs[0])

# scatter plot the dataset
plt.scatter(data[0,:],data[1,:],c = 'k')
plt.show()
```



Exercise 8.9. Making a scree plot

In [8]:
```
# Loading the data
P = 50 # Number of data points
data = datasets.make_blobs(n_samples=P, random_state=1, centers =
```