preprocessing_DLM_basic2

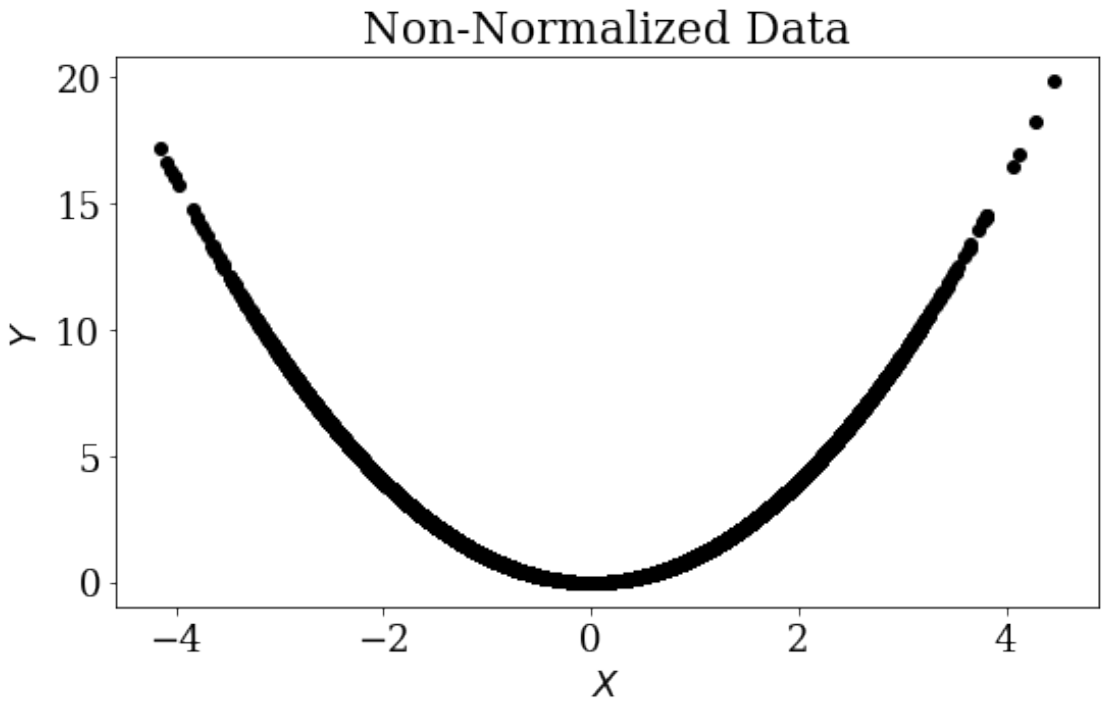
February 28, 2019

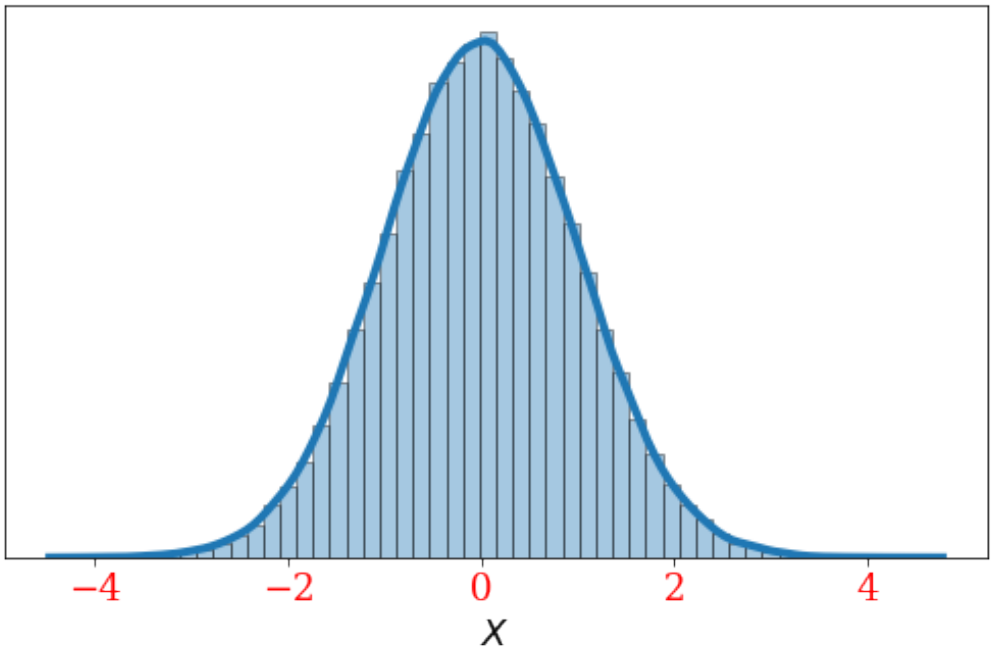
In [1]:
```
# import packages
import os
import numpy as np
from sklearn import preprocessing
import h5py
from random import shuffle
from matplotlib import rcParams  # next 3 lines set font family for plotting
rcParams['font.family'] = 'serif'
rcParams['font.sans-serif'] = ['TImes New Roman']
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 18})
import seaborn as sns

# set working directory (change the following path to match your directory structure)
main = 'C:\\Users\\Kathy_Breen\\Documents\\DL_Seminar\\Week3'  # set directory path where this
file is saved
os.chdir(main)  # make sure the Spyder is pointing to the correct folder
```

In [2]:
```
#%% Create some simple data for a known function (y = x^2)

# generate 100,000 samples (x dimension) with 1 features each (y dimension)

# create random input data on a Gaussian distribution
X = np.random.randn(100000,1)
Y = np.square(X)
```

In [3]:
```
#%% Plot non-normalized data data

fig = plt.figure(num=1, figsize=(8,10))
ax1 = fig.add_subplot(211)
ax1.plot(X,Y,'ko',label=r'$Y_{true}$')
ax1.set_xlabel(r'$X$')
ax1.set_ylabel(r'$Y$')
plt.title('Non-Normalized Data')
```

```
ax2 = fig.add_subplot(212)
sns.distplot(X,  # data to plot
             hist=True,  # plot histogram
             kde=True,   # overlay kernel density function (PDF)
             ax=ax2,  # plot on the existing axis object created for this figure
             hist_kws={'edgecolor':'black'},  # set color to outline hist bins
             kde_kws={'linewidth': 4}  # use a thick line for the kde
             )
ax2.xaxis.set_tick_params(labelcolor='r')
ax2.set_xlabel(r'$X$')
ax2.get_yaxis().set_visible(False)
plt.tight_layout()
plt.savefig('ytrue_nonorm.png')
plt.show()
```





In [4]:

```
#%% Preprocess data

# normalize inputs for each feature (column) across all samples (rows)
X = preprocessing.normalize(X,norm='l2',axis=0)
```

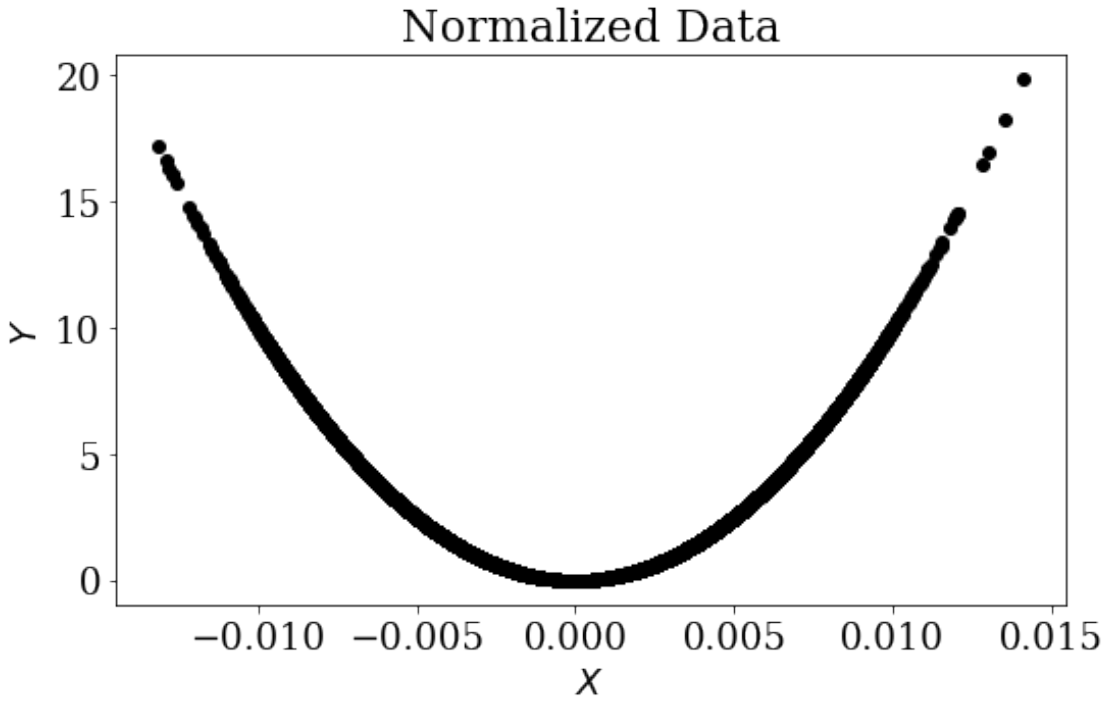
```
# shuffle samples
ismpls = list(i for i in range(0,X.shape[0]))
shuffle(ismpls)
ismpls = np.argsort(ismpls)
X = X[ismpls,:]
Y = Y[ismpls,:]

# partition X and Y into training (90%) and test (10%) sets
split_idx = int(round(X.shape[0]*0.9))
X_train = X[:split_idx,:]
X_test = X[split_idx:,:]
Y_train = Y[:split_idx,:]
Y_test = Y[split_idx:,:]
```

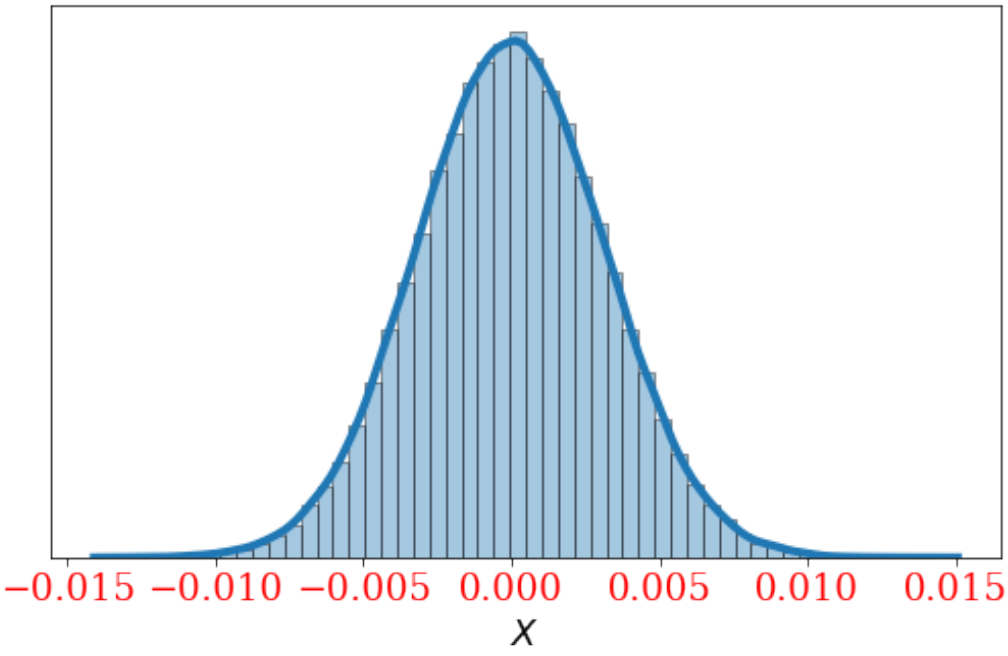
In [5]:

```
#%% Write data to *.hdf5 file

with h5py.File('X.hdf5','w') as f:
    xtrain = f.create_dataset("X_train",X_train.shape,data=X_train)
    xtest = f.create_dataset("X_test",X_test.shape,data=X_test)

with h5py.File('Y.hdf5','w') as f:
    ytrain = f.create_dataset("Y_train",Y_train.shape,data=Y_train)
    ytest = f.create_dataset("Y_test",Y_test.shape,data=Y_test)
```

In [6]:

```
#%% Plot normalized data

fig = plt.figure(num=1, figsize=(8,10))
ax1 = fig.add_subplot(211)
ax1.plot(X,Y,'ko',label=r'$Y_{true}$')
ax1.set_xlabel(r'$X$')
ax1.set_ylabel(r'$Y$')
plt.title('Normalized Data')
ax2 = fig.add_subplot(212)
sns.distplot(X,  # data to plot
             hist=True,  # plot histogram
             kde=True,   # overlay kernel density function (PDF)
             ax=ax2,  # plot on the existing axis object created for this figure
             hist_kws={'edgecolor':'black'},  # set color to outline hist bins
             kde_kws={'linewidth': 4}  # use a thick line for the kde
             )
ax2.xaxis.set_tick_params(labelcolor='r')
ax2.set_xlabel(r'$X$')
ax2.get_yaxis().set_visible(False)
plt.tight_layout()
plt.savefig('ytrue_norm.png')
plt.show()
```


Normalized Data
20
15
10
5
0
Y
−0.010
−0.005
0.000
0.005
0.010
0.015
X


−0.015
−0.010
−0.005
0.000
0.005
0.010
0.015
X