```
from tensorflow.keras.datasets import fashion_mnist
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

```
print('Training data shape:', X_train.shape)
print('Testing data shape:', X_test.shape)
```

```
Training data shape: (60000, 28, 28)
Testing data shape: (10000, 28, 28)
```

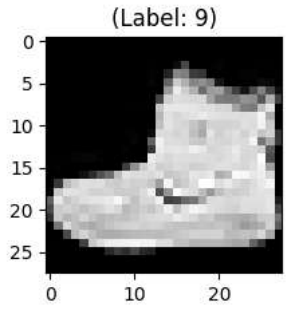
```
y_train.shape, y_test.shape
```

```
((60000,), (10000,))
```

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

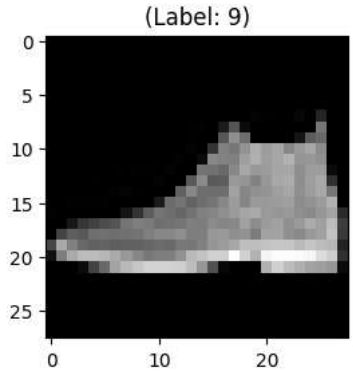
```
# Display the first image in training data
plt.figure(figsize=[5,5])
plt.subplot(121)
curr_img = np.reshape(X_train[0], (28,28))  # Fashion MNIST images are 28x28
plt.imshow(curr_img, cmap='gray')
plt.title(f"(Label: {y_train[0]})")
```

```
Text(0.5, 1.0, '(Label: 9)')
```




```
# Display the first image in testing data
plt.subplot(122)
curr_img = np.reshape(X_test[0], (28,28))
plt.imshow(curr_img, cmap='gray')
plt.title(f"(Label: {y_test[0]})")
```

```
Text(0.5, 1.0, '(Label: 9)')
```



```
np.min(X_train), np.max(X_train)
```

```
(np.uint8(0), np.uint8(255))
```

```
X_train = X_train / 255.0
np.min(X_train), np.max(X_train)
```

```
(np.float64(0.0), np.float64(1.0))
```

```
X_train.shape
```

```
(60000, 28, 28)
```

```
# Flatten the images
x_train_flat = X_train.reshape(-1, 28*28)
feat_cols = ['pixel'+str(i) for i in range(x_train_flat.shape[1])]
```

```
import pandas as pd
df_fashion = pd.DataFrame(x_train_flat, columns=feat_cols)
df_fashion['label'] = y_train
print('Size of the dataframe:', df_fashion.shape)
```

```
Size of the dataframe: (60000, 785)
```

```
from sklearn.decomposition import PCA
pca_fashion = PCA(n_components=2)
principalComponents_fashion = pca_fashion.fit_transform(df_fashion.iloc[:,:-1])
principal_fashion_Df = pd.DataFrame(data=principalComponents_fashion,
                                    columns=['principal component 1', 'principal component 2'])
principal_fashion_Df['y'] = y_train
```

```
principal_fashion_Df.head()
```

	principal component 1	principal component 2	y
0	-0.486250	6.404213	9
1	5.521290	-1.771142	0
2	-2.846709	-4.320934	0
3	0.123132	-3.847324	3
4	3.153409	-4.710466	0

Next steps: Generate code with principal_fashion_Df | View recommended plots | New interactive sheet

```
print('Explained variation per principal component:', pca_fashion.explained_variance_ratio_)
```

```
Explained variation per principal component: [0.29039228 0.1775531 ]
```

```
import seaborn as sns
plt.figure(figsize=(16,10))
sns.scatterplot(
    x="principal component 1", y="principal component 2",
    hue="y",
    palette=sns.color_palette("hls", 10),
    data=principal_fashion_Df,
    legend="full",
    alpha=0.3
)
```

```
<Axes: xlabel='principal component 1', ylabel='principal component 2'>
```