In [81]:
```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import KFold
```

# Problem a

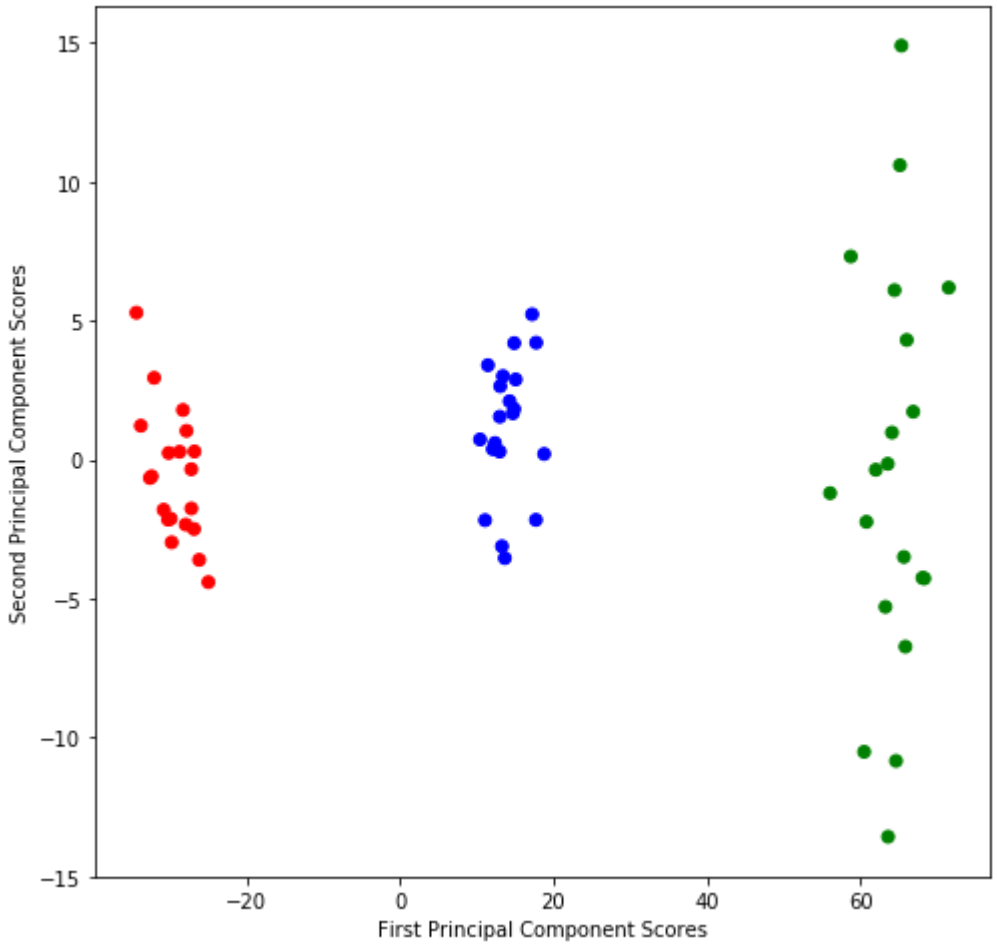
In [82]:
```
# Generate Dataset
X1 = np.random.normal(loc=9.0,scale=3.0,size=(20,50))
X2 = np.random.normal(loc=-4.0,scale=2.0,size=(20,50))
X3 = np.random.normal(loc=2.0,scale=2.0,size=(20,50))
y1 = 0*np.ones(shape=(20,1))
y2 = 1*np.ones(shape=(20,1))
y3 = 2*np.ones(shape=(20,1))

X = np.concatenate((X1,X2),axis=0)
X = np.concatenate((X,X3),axis=0)
y = np.concatenate((y1,y2),axis=0)
y = np.concatenate((y,y3),axis=0)
```

# Problem b

In [83]:
```
# Create a Principal Component Analysis model and fit it to dataset
pca = PCA(n_components=2)
pca.fit(X);

# Recall ith score vector is projection of observations onto ith PC vector
# Create Principal Component Score Vectors
score_vectors = np.dot(pca.components_, X.T)
```

In [84]:
```
# Define colors for the class labels: RED for 1, GREEN for 0
colors = ['green','red', 'blue']

fig = plt.figure(figsize=(8,8))
plt.scatter(score_vectors[0].ravel(), score_vectors[1].ravel(), c=y.ravel(), cmap=matplotlib.colors.ListedCol
ormap(colors))
plt.xlabel('First Principal Component Scores')
plt.ylabel('Second Principal Component Scores')
```

Out[84]: Text(0,0.5,'Second Principal Component Scores')



# Problem c

In [85]:
```
# Create K-Means Clustering model and fit to raw data (3 Clusters)
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)

num_class1 = np.sum(1*(kmeans.labels_ == 0))
num_class2 = np.sum(1*(kmeans.labels_ == 1))
num_class3 = np.sum(1*(kmeans.labels_ == 2))

print('Number of Observations labeled as Class 1 is ' + repr(num_class1))
print('Number of Observations labeled as Class 2 is ' + repr(num_class2))
print('Number of Observations labeled as Class 3 is ' + repr(num_class3))
```

```
Number of Observations labeled as Class 1 is 20
Number of Observations labeled as Class 2 is 20
Number of Observations labeled as Class 3 is 20
```

# Problem d

In [86]:
```
# Create K-Means Clustering model and fit to raw data (2 Clusters)
kmeans = KMeans(n_clusters=2)
kmeans.fit(X)

num_class1 = np.sum(1*(kmeans.labels_ == 0))
num_class2 = np.sum(1*(kmeans.labels_ == 1))

print('Number of Observations labeled as Class 1 is ' + repr(num_class1))
print('Number of Observations labeled as Class 2 is ' + repr(num_class2))
```

```
Number of Observations labeled as Class 1 is 40
Number of Observations labeled as Class 2 is 20
```

# Problem e

In [87]:
```
# Create K-Means Clustering model and fit to raw data (4 Clusters)
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)

num_class1 = np.sum(1*(kmeans.labels_ == 0))
num_class2 = np.sum(1*(kmeans.labels_ == 1))
num_class3 = np.sum(1*(kmeans.labels_ == 2))
num_class4 = np.sum(1*(kmeans.labels_ == 3))

print('Number of Observations labeled as Class 1 is ' + repr(num_class1))
print('Number of Observations labeled as Class 2 is ' + repr(num_class2))
print('Number of Observations labeled as Class 3 is ' + repr(num_class3))
print('Number of Observations labeled as Class 4 is ' + repr(num_class4))
```

```
Number of Observations labeled as Class 1 is 20
Number of Observations labeled as Class 2 is 13
Number of Observations labeled as Class 3 is 20
Number of Observations labeled as Class 4 is 7
```

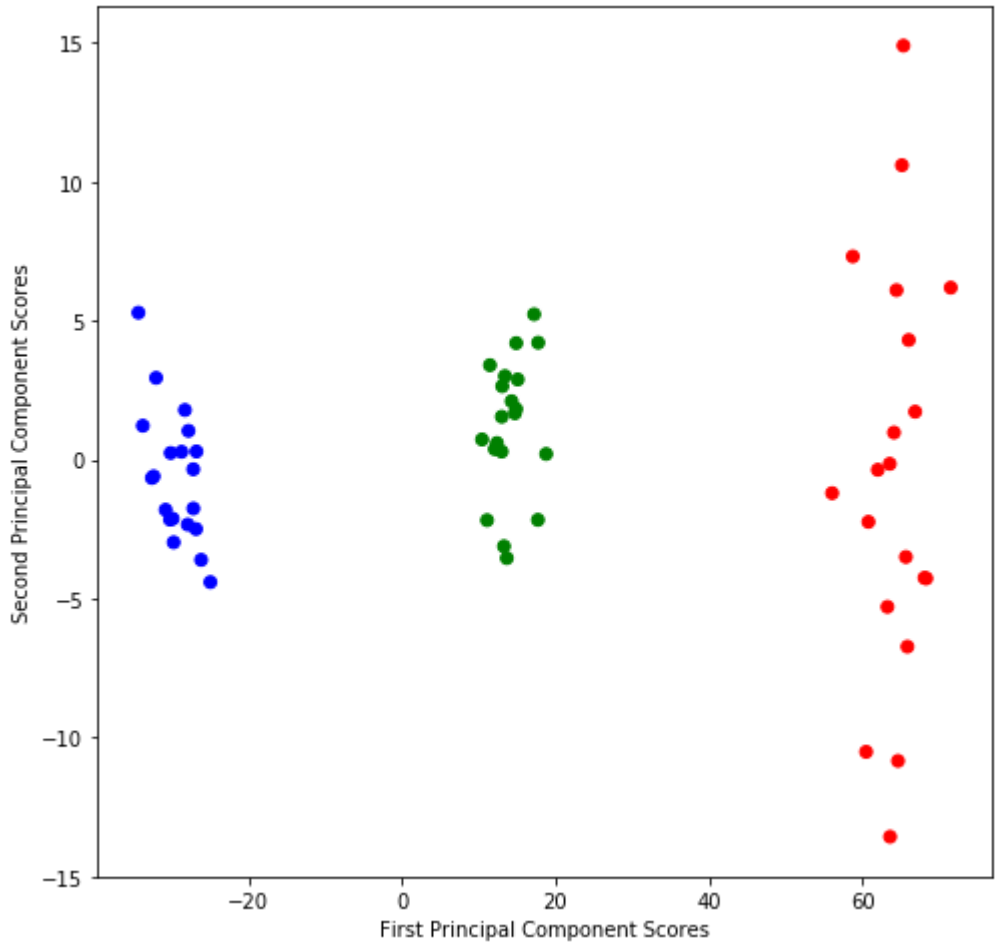
# Problem f

In [88]:
```
# Create K-Means Clustering model and fit to PC score vectors (3 Clusters)
kmeans = KMeans(n_clusters=3)
kmeans.fit(score_vectors.T)

num_class1 = np.sum(1*(kmeans.labels_ == 0))
num_class2 = np.sum(1*(kmeans.labels_ == 1))
num_class3 = np.sum(1*(kmeans.labels_ == 2))

print('Number of Observations labeled as Class 1 is ' + repr(num_class1))
print('Number of Observations labeled as Class 2 is ' + repr(num_class2))
print('Number of Observations labeled as Class 3 is ' + repr(num_class3))
```

```
Number of Observations labeled as Class 1 is 20
Number of Observations labeled as Class 2 is 20
Number of Observations labeled as Class 3 is 20
```

In [89]:
```
# Define colors for the class labels: RED for 1, GREEN for 0
colors = ['green','red', 'blue']

fig = plt.figure(figsize=(8,8))
plt.scatter(score_vectors[0].ravel(), score_vectors[1].ravel(), c=kmeans.labels_.ravel(), cmap=matplotlib.colors.ListedColormap(colors))
plt.xlabel('First Principal Component Scores')
plt.ylabel('Second Principal Component Scores')
```

Out[89]: Text(0,0.5,'Second Principal Component Scores')



# Problem g

In [90]:
```
from scipy import stats
# Generate Standardized Dataset (using Z-score)
X1_z_score = stats.zscore(X1)
X2_z_score = stats.zscore(X2)
X3_z_score = stats.zscore(X3)
y1_z = y1
y2_z = y2
y3_z = y3

X_z_score = np.concatenate((X1_z_score, X2_z_score), axis=0)
X_z_score = np.concatenate((X_z_score, X3_z_score), axis=0)
y_z = np.concatenate((y1_z, y2_z), axis=0)
y_z = np.concatenate((y_z, y3_z), axis=0)
```

In [91]:
```
# Create K-Means Clustering model and fit to standardized data (3 Clusters)
kmeans = KMeans(n_clusters=3)
kmeans.fit(X_z_score)

num_class1 = np.sum(1*(kmeans.labels_ == 0))
num_class2 = np.sum(1*(kmeans.labels_ == 1))
num_class3 = np.sum(1*(kmeans.labels_ == 2))

print('Number of Observations labeled as Class 1 is ' + repr(num_class1))
print('Number of Observations labeled as Class 2 is ' + repr(num_class2))
print('Number of Observations labeled as Class 3 is ' + repr(num_class3))
```

```
Number of Observations labeled as Class 1 is 10
Number of Observations labeled as Class 2 is 18
Number of Observations labeled as Class 3 is 32
```

In [ ]:
```

```