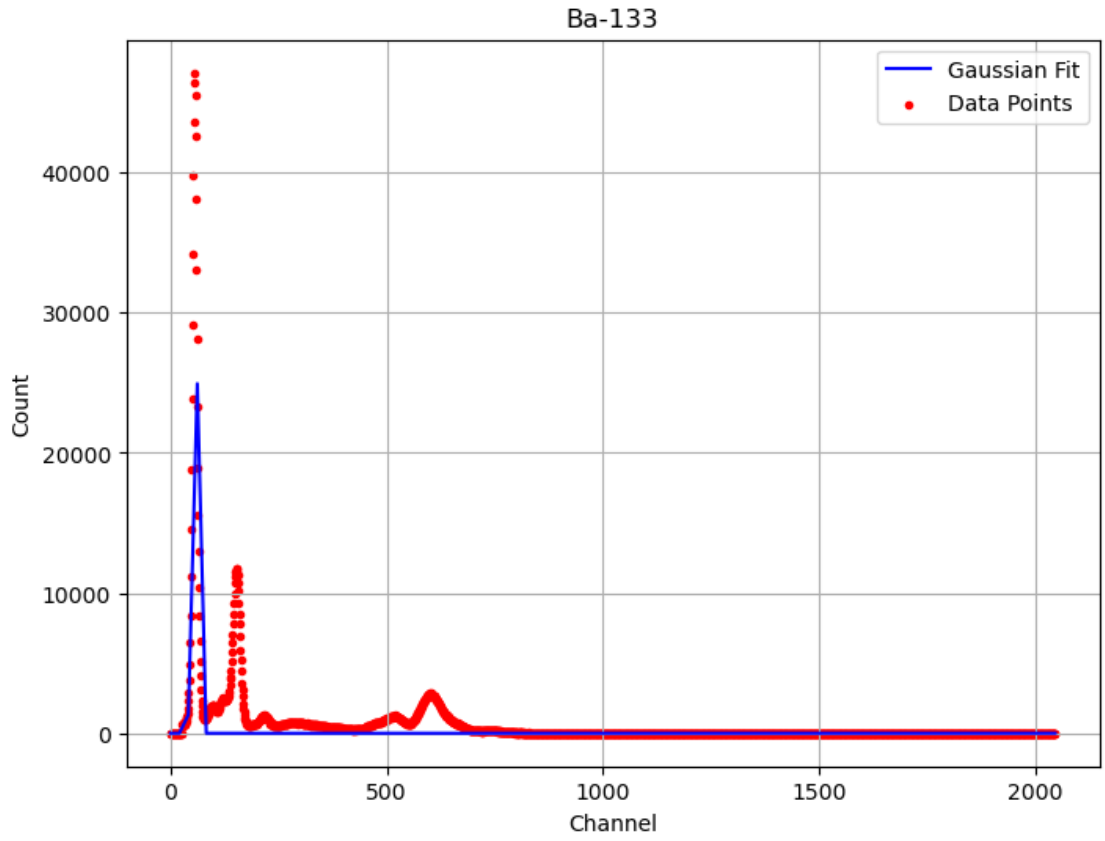
09/15/2023 - James Amidei

These are some preliminary scatter plots with Gaussian fits from the data collected on 09/15/2023.

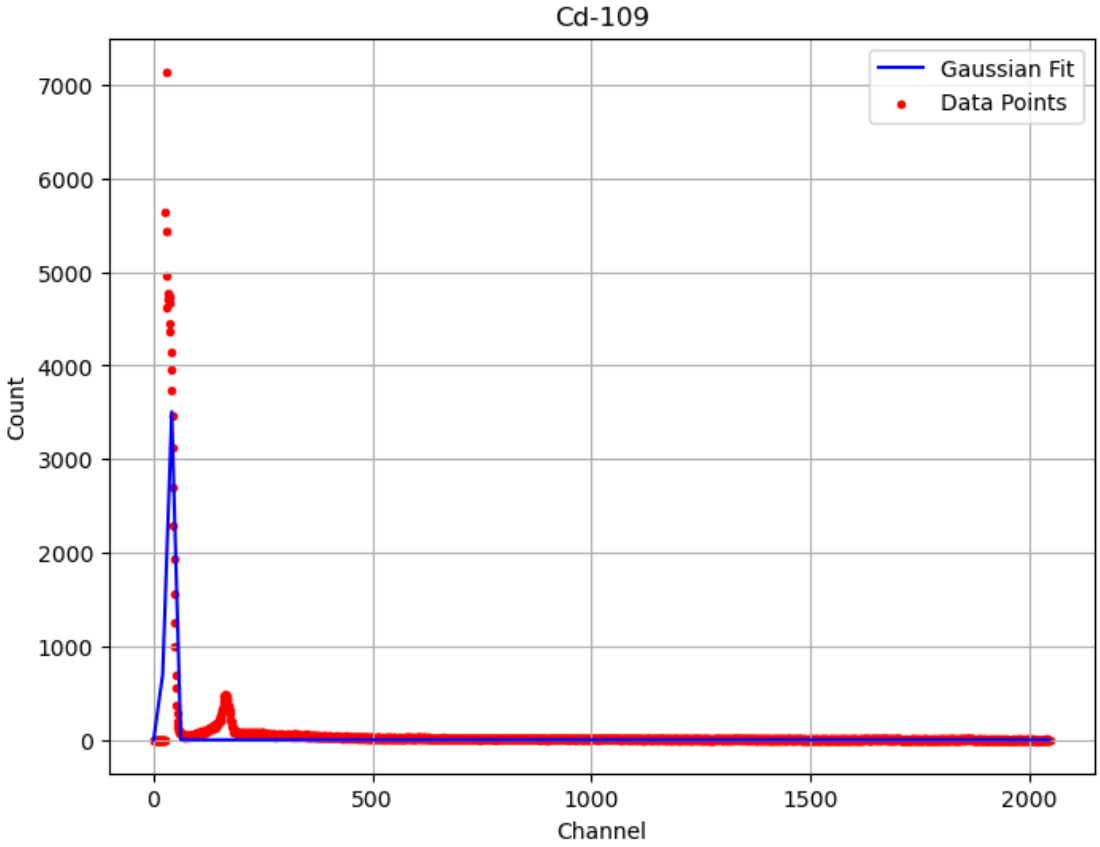
In [40]:

```
import numpy as np
import pylab as py
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

data_1 = py.loadtxt('Ba-133data.csv', skiprows=22, delimiter=',', usecols=
data_2 = py.loadtxt('Cd-109data.csv', skiprows=22, delimiter=',', usecols=
data_3 = py.loadtxt('Co-57data.csv', skiprows=22, delimiter=',', usecols=(
data_4 = py.loadtxt('Co-60data.csv', skiprows=22, delimiter=',', usecols=(
data_5 = py.loadtxt('Mn-54data.csv', skiprows=22, delimiter=',', usecols=(
data_6 = py.loadtxt('Na-22data.csv', skiprows=22, delimiter=',', usecols=(
data_7 = py.loadtxt('Unknowndata.csv', skiprows=22, delimiter=',', usecols

x_1 = data_1[:,0]
y_1 = data_1[:,1]
x_2 = data_2[:,0]
y_2 = data_2[:,1]
x_3 = data_3[:,0]
y_3 = data_3[:,1]
x_4 = data_4[:,0]
y_4 = data_4[:,1]
x_5 = data_5[:,0]
y_5 = data_5[:,1]
x_6 = data_6[:,0]
y_6 = data_6[:,1]
x_7 = data_7[:,0]
y_7 = data_7[:,1]

def gauss_function(x, amplitude, mean, stddev):
    return amplitude * np.exp(-(x - mean) ** 2 / (2 * stddev ** 2))

initial_guess_1 = [1.0, np.mean(x_1), np.std(x_1)]
initial_guess_2 = [1.0, np.mean(x_2), np.std(x_2)]
initial_guess_3 = [1.0, np.mean(x_3), np.std(x_3)]
initial_guess_4 = [1.0, np.mean(x_4), np.std(x_4)]
initial_guess_5 = [1.0, np.mean(x_5), np.std(x_5)]
initial_guess_6 = [1.0, np.mean(x_6), np.std(x_6)]
initial_guess_7 = [1.0, np.mean(x_7), np.std(x_7)]

params_1, params_covariance_1 = curve_fit(gauss_function, x_1, y_1, p0=ini
params_2, params_covariance_2 = curve_fit(gauss_function, x_2, y_2, p0=ini
params_3, params_covariance_3 = curve_fit(gauss_function, x_3, y_3, p0=ini
params_4, params_covariance_4 = curve_fit(gauss_function, x_4, y_4) #, p0=
params_5, params_covariance_5 = curve_fit(gauss_function, x_5, y_5, p0=ini
params_6, params_covariance_6 = curve_fit(gauss_function, x_6, y_6, p0=ini
params_7, params_covariance_7 = curve_fit(gauss_function, x_7, y_7, p0=ini

amplitude_1, mean_1, stddev_1 = params_1
amplitude_2, mean_2, stddev_2 = params_2
amplitude_3, mean_3, stddev_3 = params_3
amplitude_4, mean_4, stddev_4 = params_4
amplitude_5, mean_5, stddev_5 = params_5
amplitude_6, mean_6, stddev_6 = params_6
amplitude_7, mean_7, stddev_7 = params_7

x_1_fit = np.linspace(min(x_1), max(x_1), 100)
y_1_fit = gauss_function(x_1_fit, amplitude_1, mean_1, stddev_1)
```

```
x_2_fit = np.linspace(min(x_2), max(x_2), 100)
y_2_fit = gauss_function(x_2_fit, amplitude_2, mean_2, stddev_2)
x_3_fit = np.linspace(min(x_3), max(x_3), 100)
y_3_fit = gauss_function(x_3_fit, amplitude_3, mean_3, stddev_3)
x_4_fit = np.linspace(min(x_4), max(x_4), 100)
y_4_fit = gauss_function(x_4_fit, amplitude_4, mean_4, stddev_4)
x_5_fit = np.linspace(min(x_5), max(x_5), 100)
y_5_fit = gauss_function(x_5_fit, amplitude_5, mean_5, stddev_5)
x_6_fit = np.linspace(min(x_6), max(x_6), 100)
y_6_fit = gauss_function(x_6_fit, amplitude_6, mean_6, stddev_6)
x_7_fit = np.linspace(min(x_7), max(x_7), 100)
y_7_fit = gauss_function(x_7_fit, amplitude_7, mean_7, stddev_7)

plt.figure(figsize=(8, 6))
plt.plot(x_1_fit, y_1_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_1, y_1, label='Data Points', color='red', marker='.')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Ba-133')
plt.legend()
plt.grid(True)
plt.show()
```

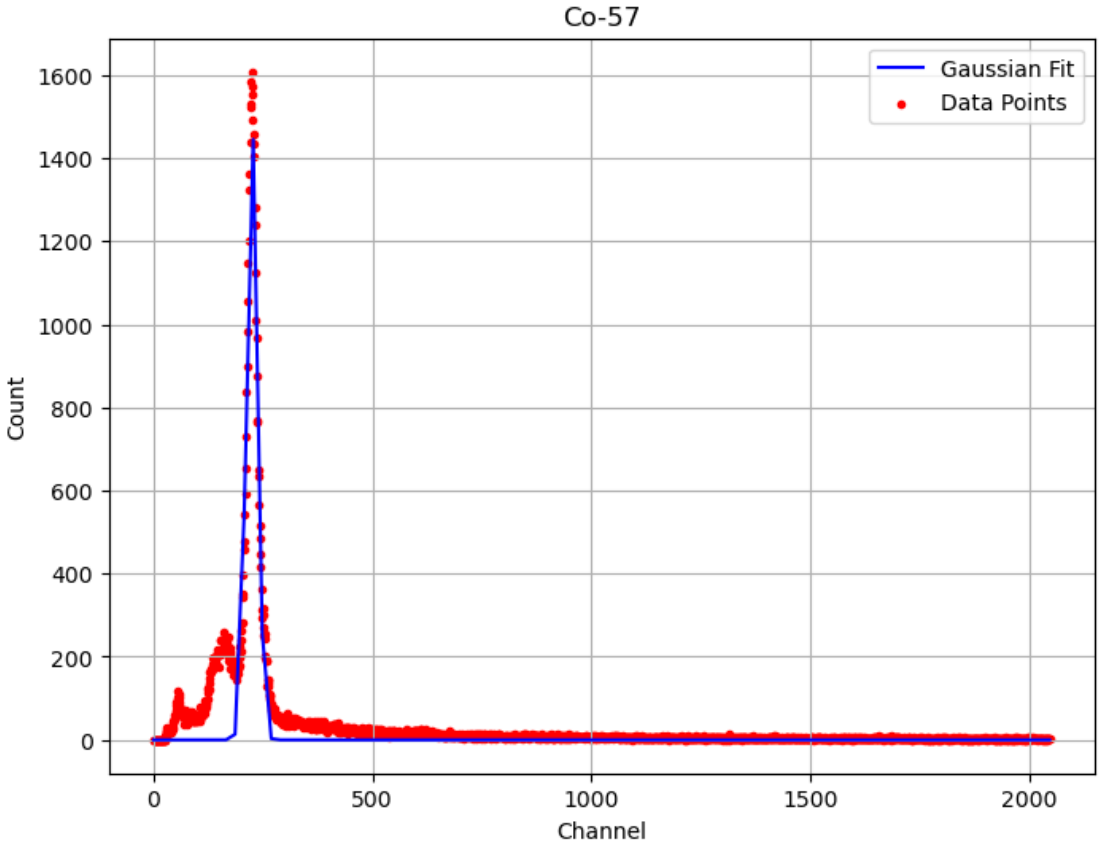


In [41]:
```
plt.figure(figsize=(8, 6))
plt.plot(x_2_fit, y_2_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_2, y_2, label='Data Points', color='red', marker='.')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Cd-109')
plt.legend()
plt.grid(True)
plt.show()
```



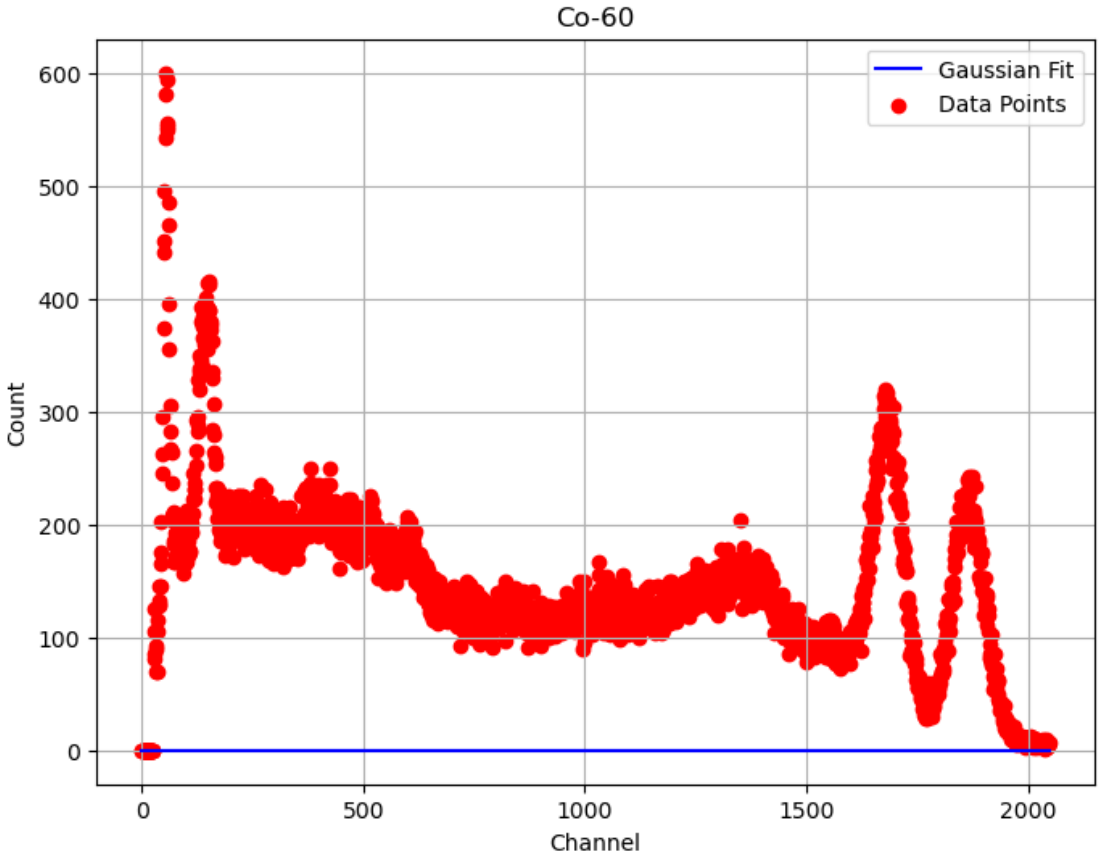
In [42]:

```
plt.figure(figsize=(8, 6))
plt.plot(x_3_fit, y_3_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_3, y_3, label='Data Points', color='red', marker='.')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Co-57')
plt.legend()
plt.grid(True)
plt.show()
```



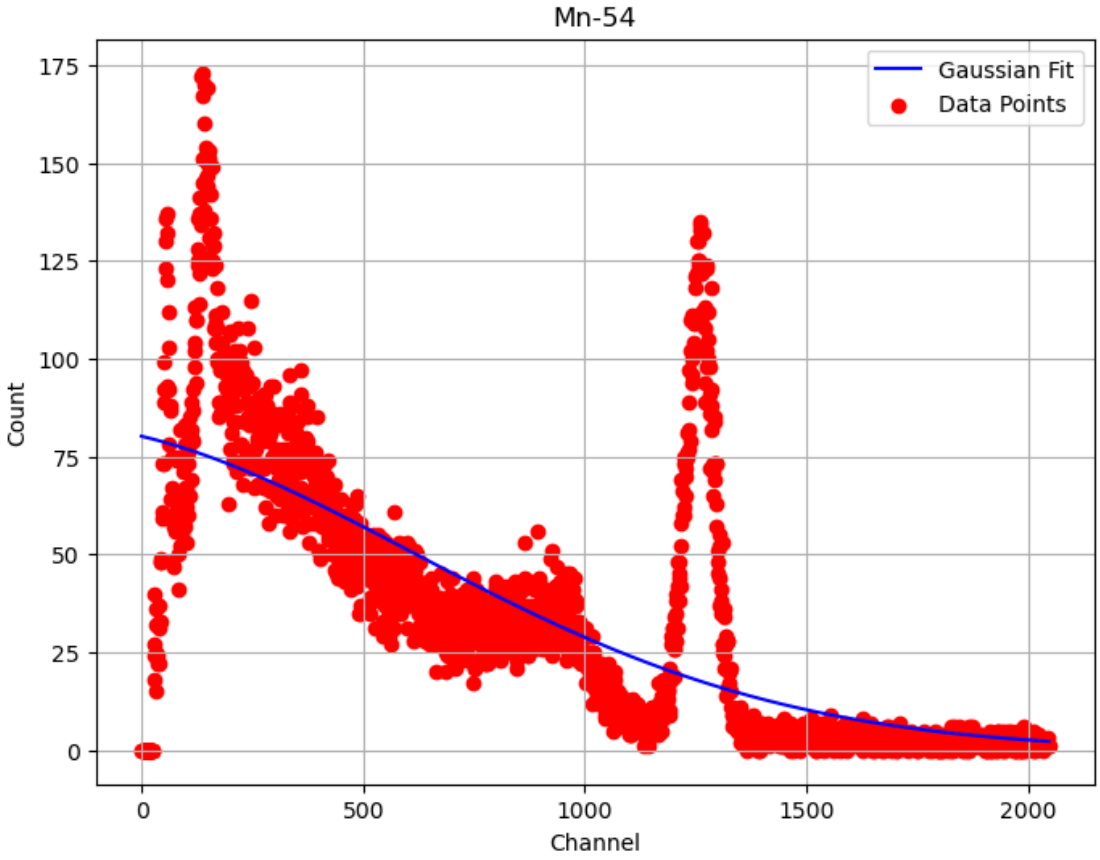
In [43]:

```
# For whatever reason, I wasn't able to get a Gaussian curve to work on th
# Will try again later
plt.figure(figsize=(8, 6))
plt.plot(x_4_fit, y_4_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_4, y_4, label='Data Points', color='red', marker='o')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Co-60')
plt.legend()
plt.grid(True)
plt.show()
```

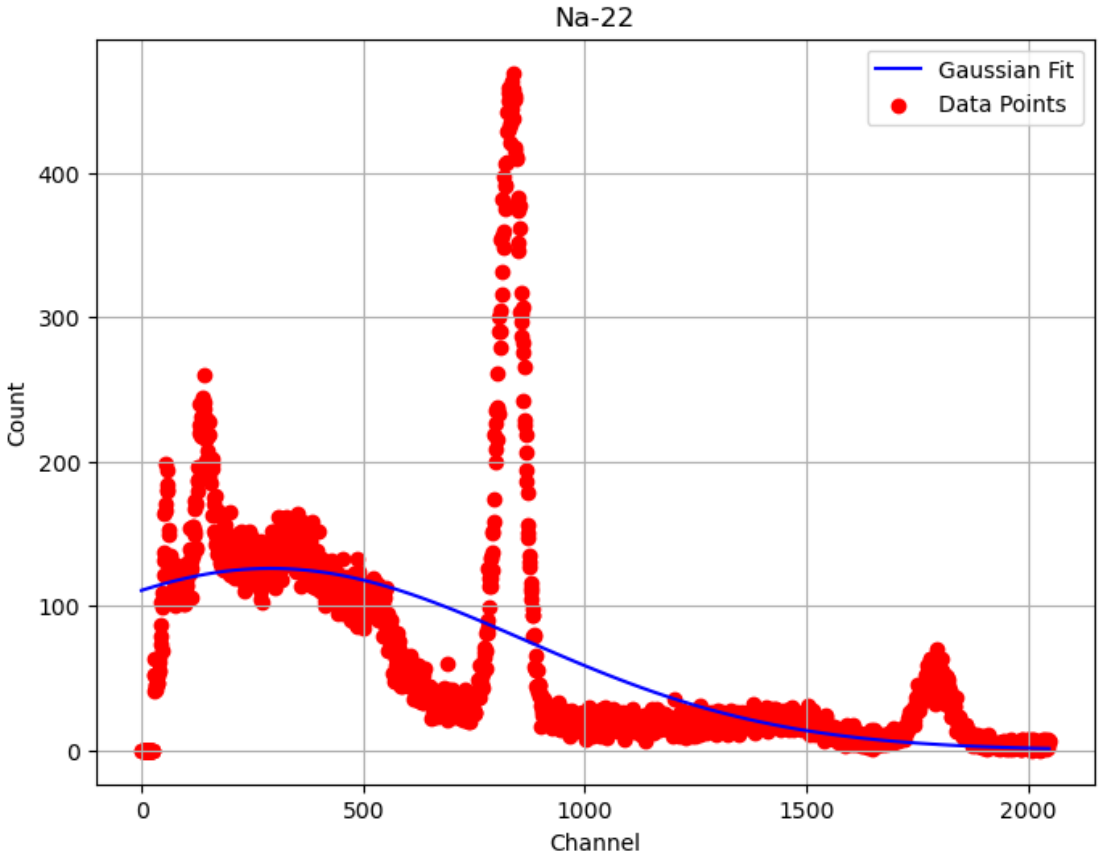


In [44]:

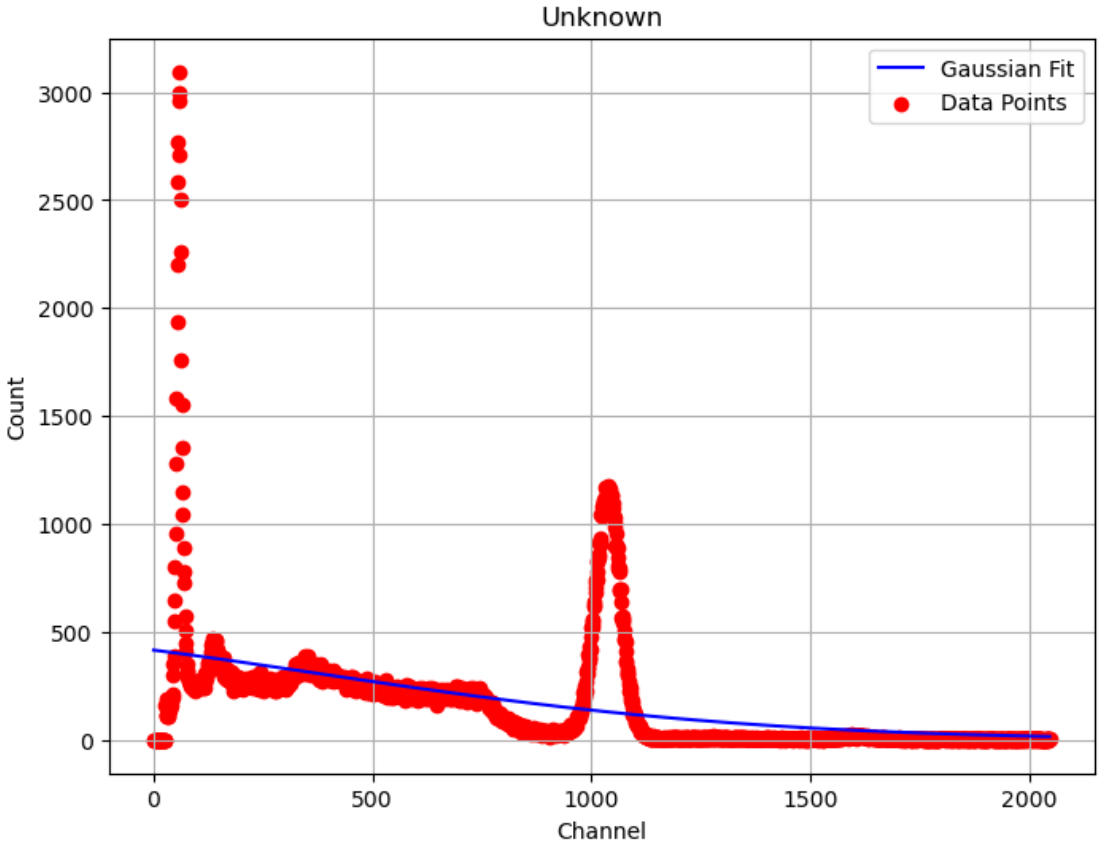
```
plt.figure(figsize=(8, 6))
plt.plot(x_5_fit, y_5_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_5, y_5, label='Data Points', color='red', marker='o')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Mn-54')
plt.legend()
plt.grid(True)
plt.show()
```



In [45]:
```
plt.figure(figsize=(8, 6))
plt.plot(x_6_fit, y_6_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_6, y_6, label='Data Points', color='red', marker='o')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Na-22')
plt.legend()
plt.grid(True)
plt.show()
```



In [46]:
```
plt.figure(figsize=(8, 6))
plt.plot(x_7_fit, y_7_fit, label='Gaussian Fit', color='blue')
plt.scatter(x_7, y_7, label='Data Points', color='red', marker='o')
plt.xlabel('Channel')
plt.ylabel('Count')
plt.title('Unknown')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]:

In [ ]: