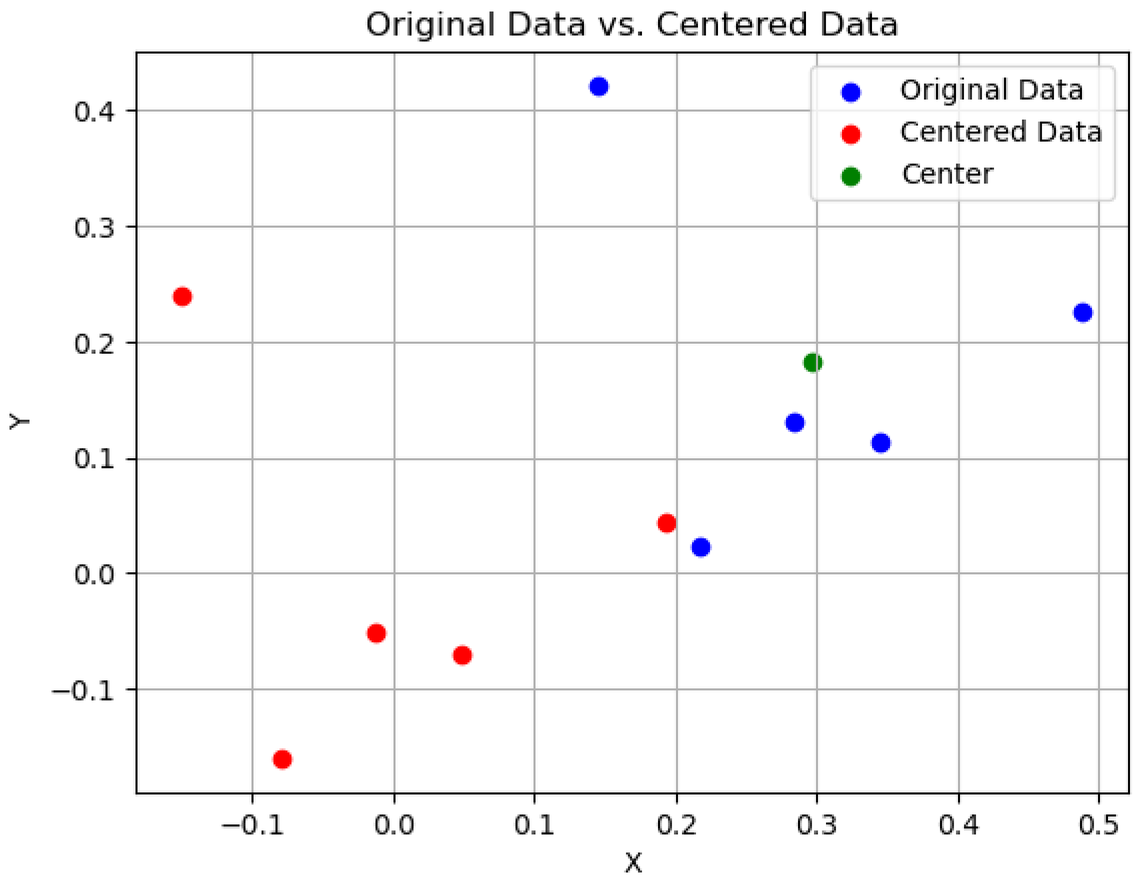
In [74]:
```
import numpy as np
import matplotlib.pyplot as plt

# Create data matrix Ao with random values
m = 2
n = 5
Ao = np.random.rand(m, n)
# Calculate the mean along each column to center the data
mean = np.mean(Ao, axis=1, keepdims=True)
Ao_centered = Ao -mean

# Plot the original data
plt.scatter(Ao[0], Ao[1], color='b', label='Original Data')

# Plot the centered data
plt.scatter(Ao_centered[0], Ao_centered[1], color='r', label='Centered Data')

# Plot the mean as a point at the center
plt.scatter(mean[0], mean[1], color='g', label='Center')

# Set axis labels and legend
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Original Data vs. Centered Data')
plt.grid(True)

# Show the plot
plt.show()
```




In [75]:
```
# Calculate the covariance between the two rows of the data
covariance = np.cov(Ao)
```

```python
# Extract the covariance between the two rows
covariance_between_rows = covariance[0, 1]

# Calculate the correlation between the two rows of the data
correlation = np.corrcoef(Ao)

# Extract the correlation between the two rows
correlation_between_rows = correlation[0, 1]

print("Covariance between the two rows: ", covariance_between_rows)
print("Correlation between the two rows: ", correlation_between_rows)
```

```
Covariance between the two rows:  -0.004428644101878765
Correlation between the two rows:  -0.2224269569166325
```

In [76]:

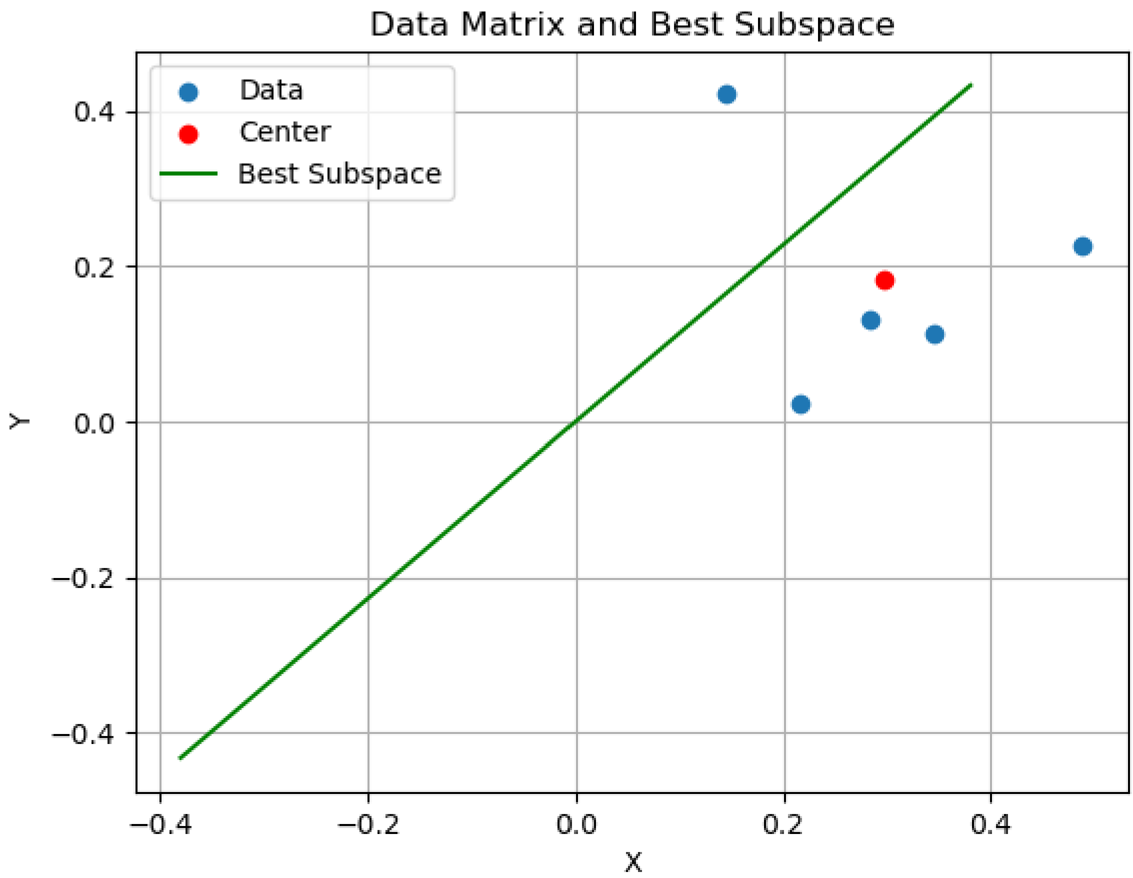
```python
import numpy as np

# Perform SVD on the data matrix Ao
U, s, VT = np.linalg.svd(Ao, full_matrices=False)

# Extract the first right singular vector, which represents the best one-dimensiona
best_eigenvector = VT[0, :]

# The one-dimensional subspace is spanned by the best eigenvector
best_approximation = np.outer(best_eigenvector, best_eigenvector.T)

print("The one-dimensional subspace that best approximates the data:")
print(best_approximation)
```

```
The one-dimensional subspace that best approximates the data:
[[0.14469139 0.16454818 0.16169668 0.24936974 0.09137071]
 [0.16454818 0.18713002 0.1838872  0.2835921  0.10391001]
 [0.16169668 0.1838872  0.18070058 0.27867767 0.10210933]
 [0.24936974 0.2835921  0.27867767 0.42977862 0.15747371]
 [0.09137071 0.10391001 0.10210933 0.15747371 0.0576994 ]]
```

In [77]:

```python
# The one-dimensional subspace is spanned by the best eigenvector
print(mean.shape,best_eigenvector.shape,np.linspace(-1,1,num=100).shape)
mean = mean.reshape(-1, 1)
best_eigenvector = best_eigenvector.reshape(-1, 1)
print(mean.shape,best_eigenvector.shape,np.linspace(-1,1,num=100).shape)
best_line = best_eigenvector* np.linspace(-1, 1, num=100)
best_line = mean + best_line

# Plot the data matrix, its center, and the one-dimensional subspace
plt.scatter(Ao[0, :], Ao[1, :], label='Data')
plt.scatter(mean[0], mean[1], color='red', marker='o', label='Center')
plt.plot(best_line[0, :], best_line[1, :], color='green', label='Best Subspace')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Data Matrix and Best Subspace')
plt.grid(True)
plt.show()
```

```
(2, 1) (5,) (100,)
(2, 1) (5, 1) (100,)
```


Data Matrix and Best Subspace
Data
Center
Best Subspace
0.4
0.2
0.0
−0.2
−0.4
Y
−0.4
−0.2
0.0
0.2
0.4
X

```
In [ ]:
```