

In [57]:

```
1 lreg = LogisticRegression()
2 lreg.fit(pd.DataFrame(X_train), y_train)
```

Out[57]:

LogisticRegression()

In [58]:

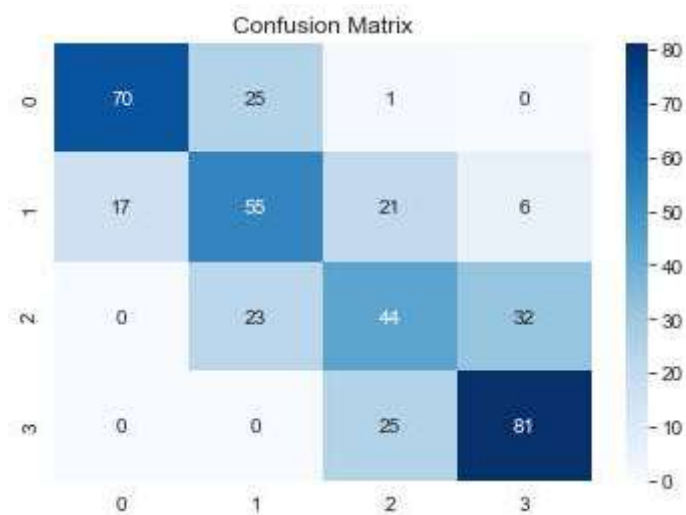
```
1 lreg_preds = lreg.predict(X_test)
```

In [59]:

```
1 from sklearn.metrics import accuracy_score , confusion_matrix
2 # what do they look like? (show me some of the predictions)
3 # create confusion matrix plot
4 cf_matrix = confusion_matrix(y_test, lreg_preds)
5 sns.heatmap(cf_matrix, annot=True, fmt='', cmap='Blues').set_title("Confusion Matrix")
```

Out[59]:

Text(0.5, 1.0, 'Confusion Matrix')



In [60]:

```
1 accuracy_score(y_true=y_test, y_pred=lreg_preds)
```

Out[60]:

0.625

In [61]:

```
1 lreg_mae = mean_absolute_error(y_test , lreg_preds)
2 lreg_mse = mean_squared_error(y_test, lreg_preds)
3 lreg_rse = mean_squared_error(y_test, lreg_preds , squared=False)
4 lreg_r2 = r2_score(y_test , lreg_preds)
```

In [75]:

```
1 print('Logistic Regression Mean Absolute Error:', lreg_mae)
2 print('Logistic Regression Mean squared Error:', lreg_mse)
3 print('Logistic Regression Squared Error:', lreg_rse)
4 print('Logistic Regression R2 Score:', lreg_r2)
```

Logistic Regression Mean Absolute Error: 0.3925

Logistic Regression Mean squared Error: 0.4275

Logistic Regression Squared Error: 0.653834841531101

Logistic Regression R2 Score: 0.6603351955307262

Creating & Training KNN Model

In [63]:

```
1 from sklearn.neighbors import KNeighborsClassifier
2 knn = KNeighborsClassifier(n_neighbors=15)
3 knn.fit(X_train,y_train)
```

Out[63]:

KNeighborsClassifier(n_neighbors=15)

In [64]:

```
1 knn.score(X_test,y_test)
```

Out[64]:

0.935

Elbow Method For optimum value of K

In [65]:

```
1 # Scale Data
2 ss2=StandardScaler()
3 sc2=ss2.fit_transform(X)
```