

```

In [1]: import pandas as pd
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

# Cargar el archivo CSV en un DataFrame de pandas
df = pd.read_csv('datasauRUS.csv')

# Mostrar las primeras filas del DataFrame para verificar la carga de datos
print(df.head())

# Obtener las categorías únicas de la columna 'dataset'
categorias = df['dataset'].unique()

# Crear gráficos para cada categoría en la columna 'dataset'
for categoria in categorias:
    # Filtrar los datos por la categoría actual
    df_categoria = df[df['dataset'] == categoria]

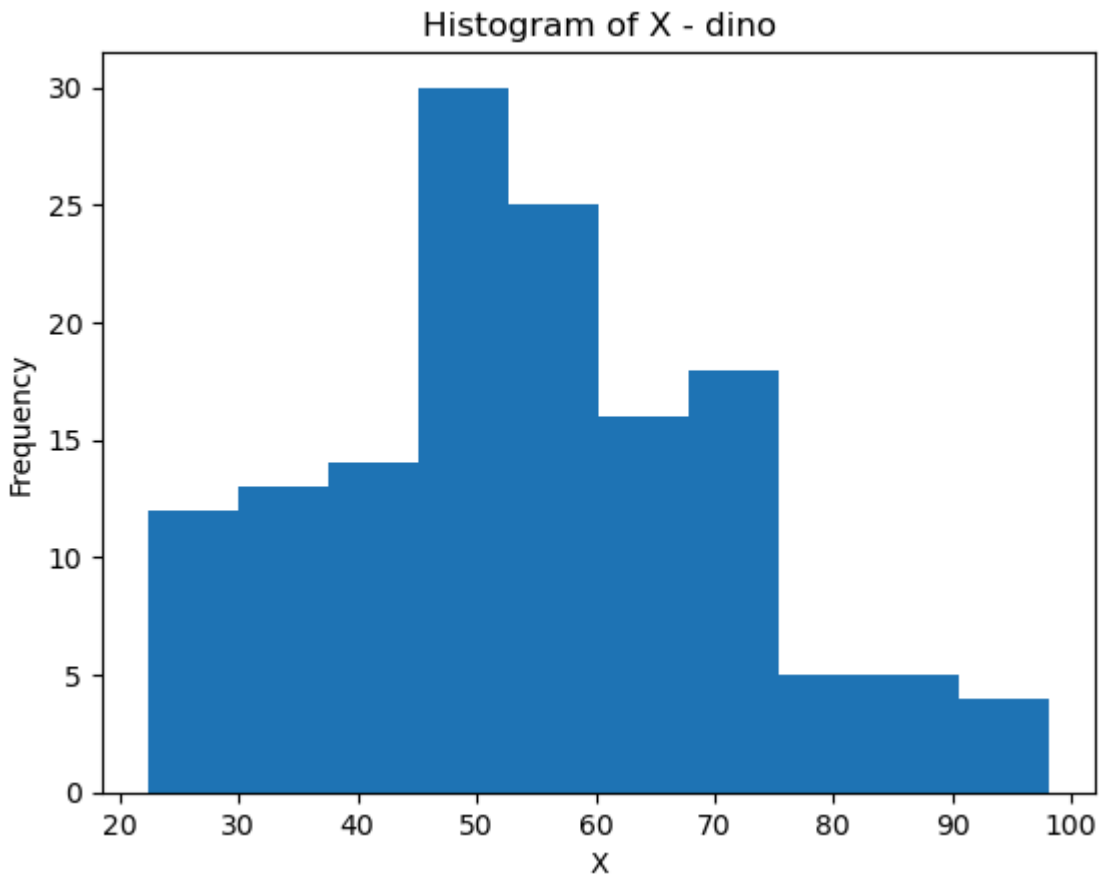
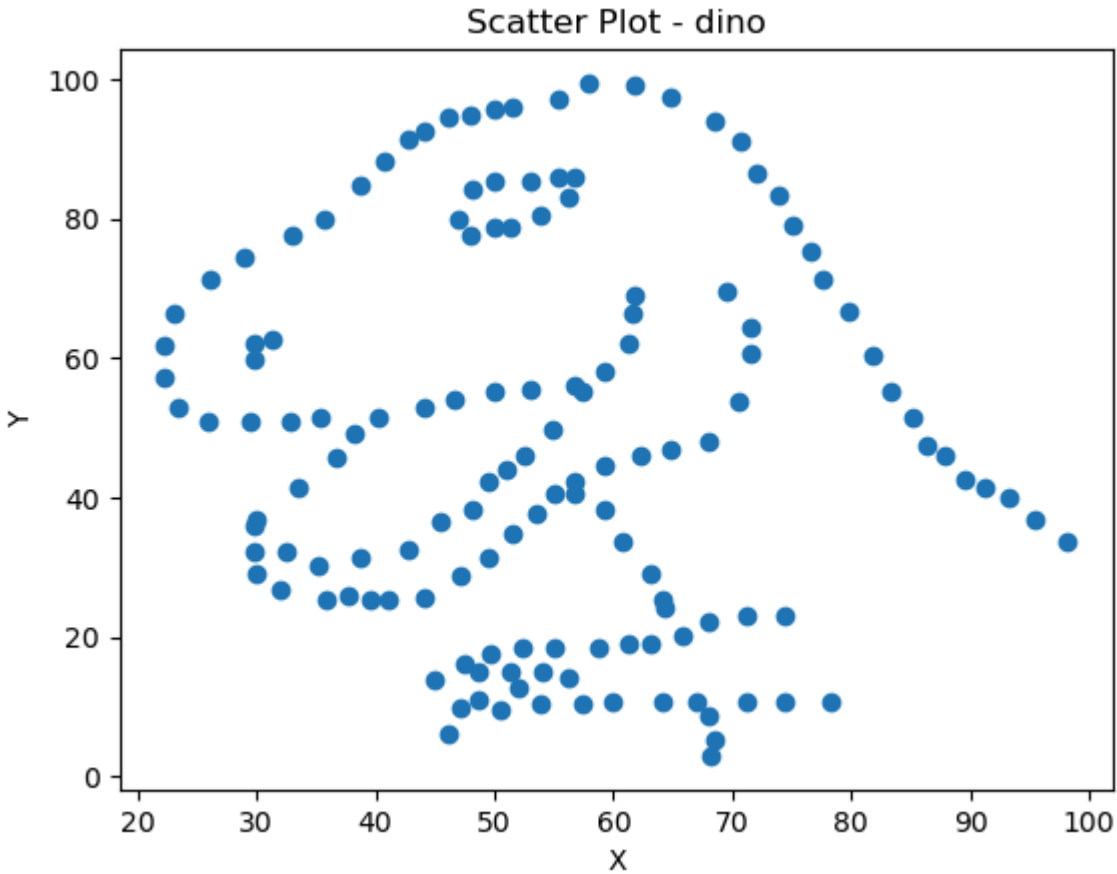
    # Crear un gráfico de dispersión para las columnas 'x' y 'y' con nombre "Scatter Plot - {categoria}"
    plt.scatter(df_categoria['x'], df_categoria['y'])
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title(f'Scatter Plot - {categoria}')
    plt.show()

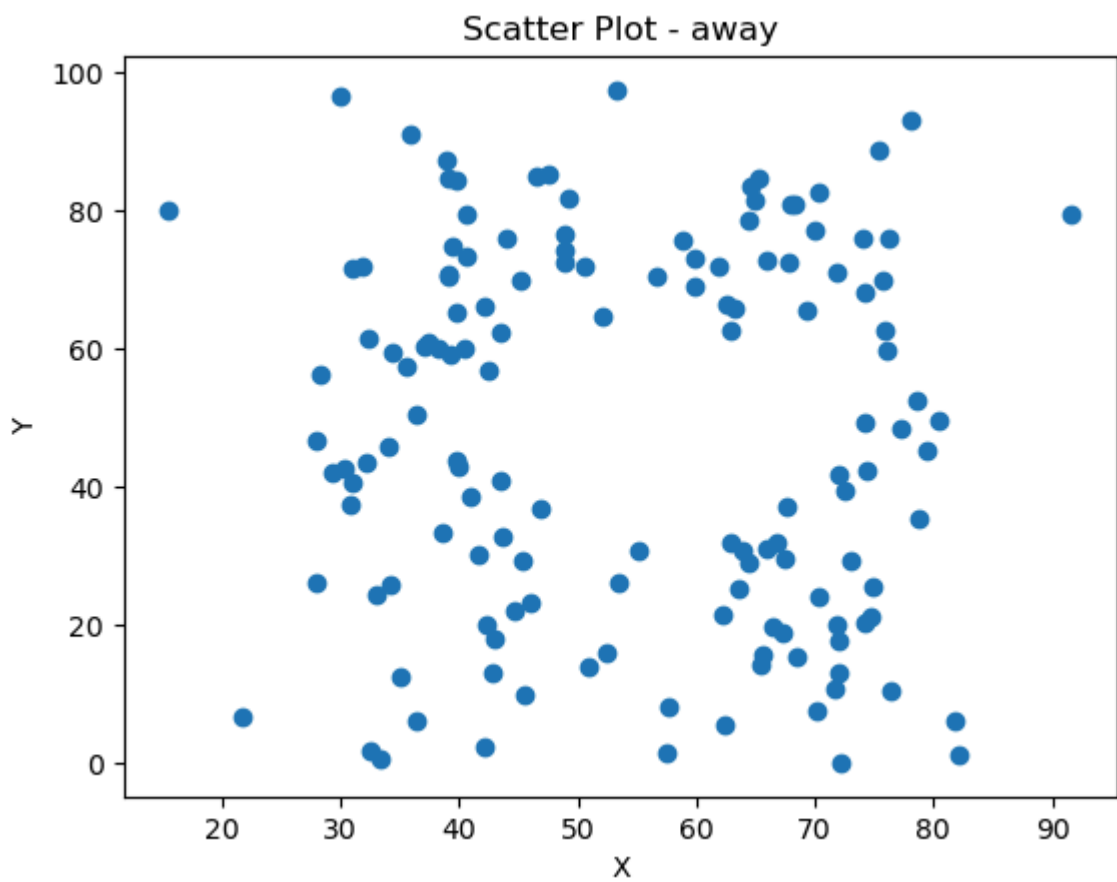
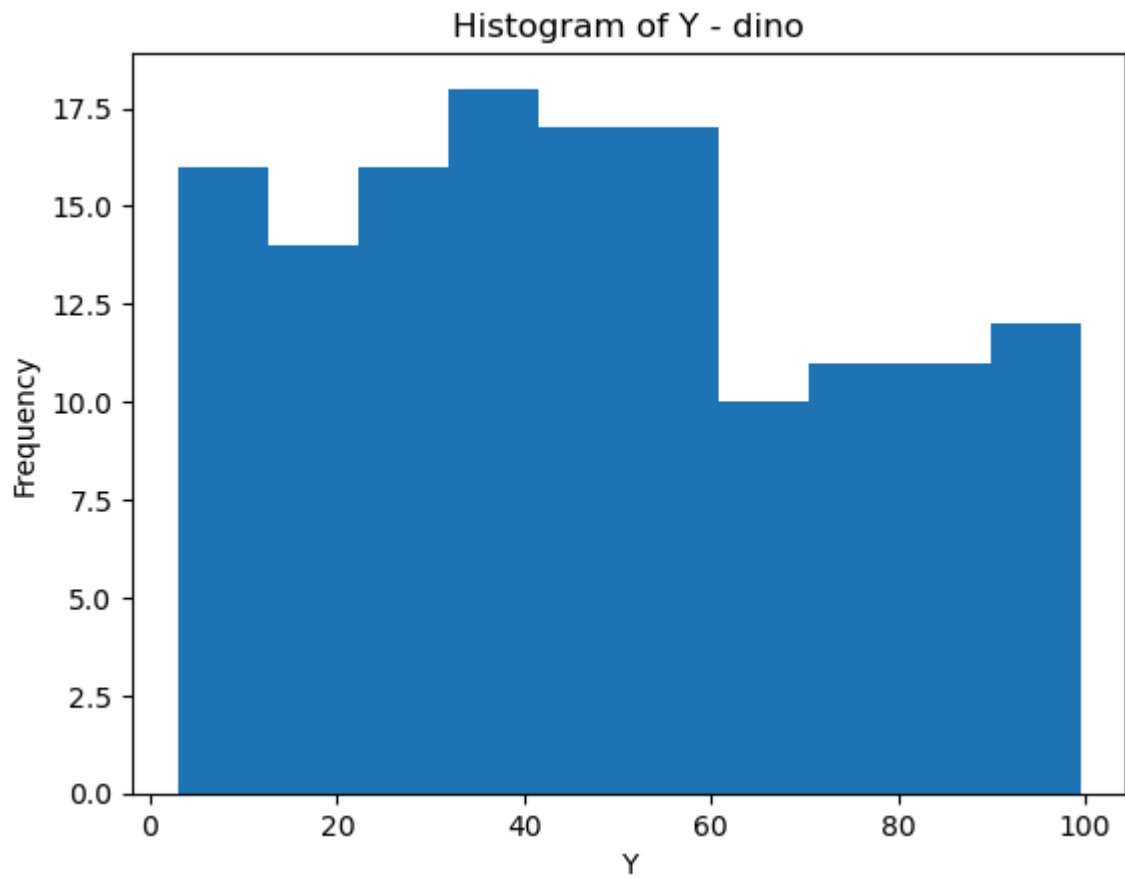
    # Crear un histograma para la columna 'x' con nombre "Histogram of X - {categoria}"
    plt.hist(df_categoria['x'])
    plt.xlabel('X')
    plt.ylabel('Frequency')
    plt.title(f'Histogram of X - {categoria}')
    plt.show()

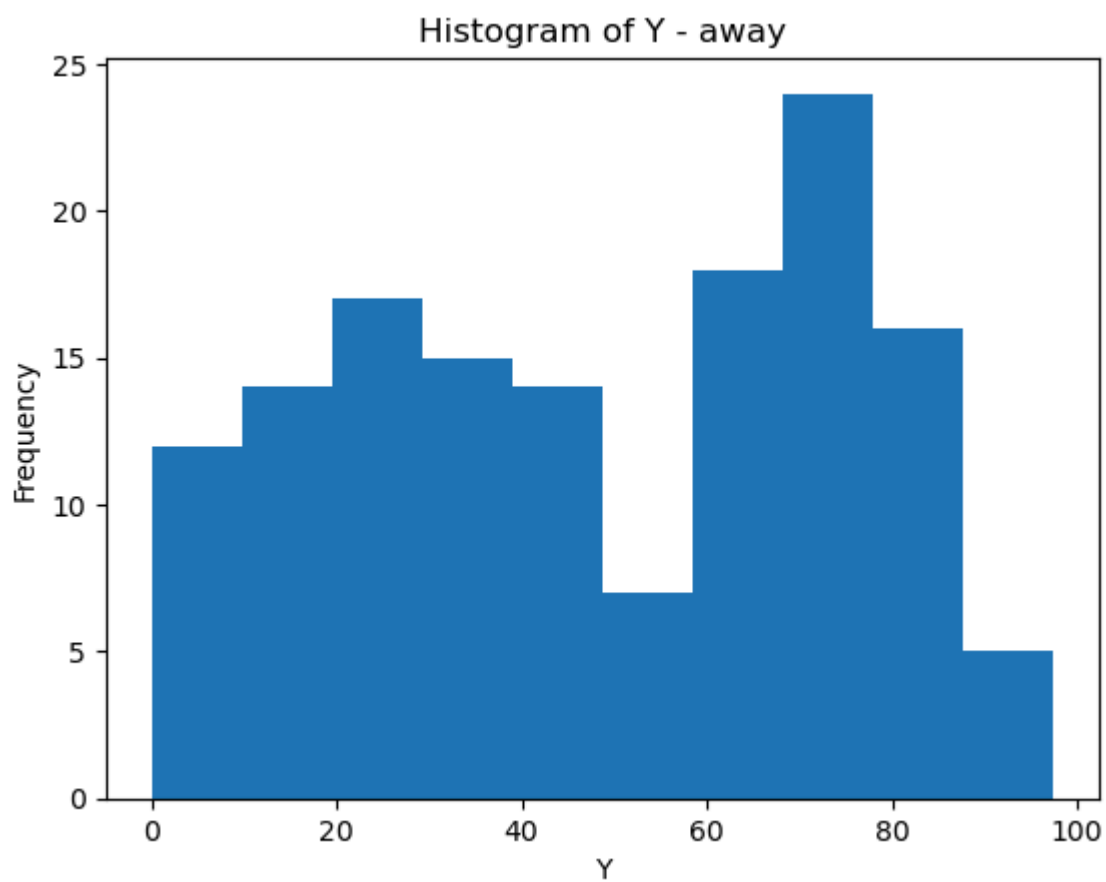
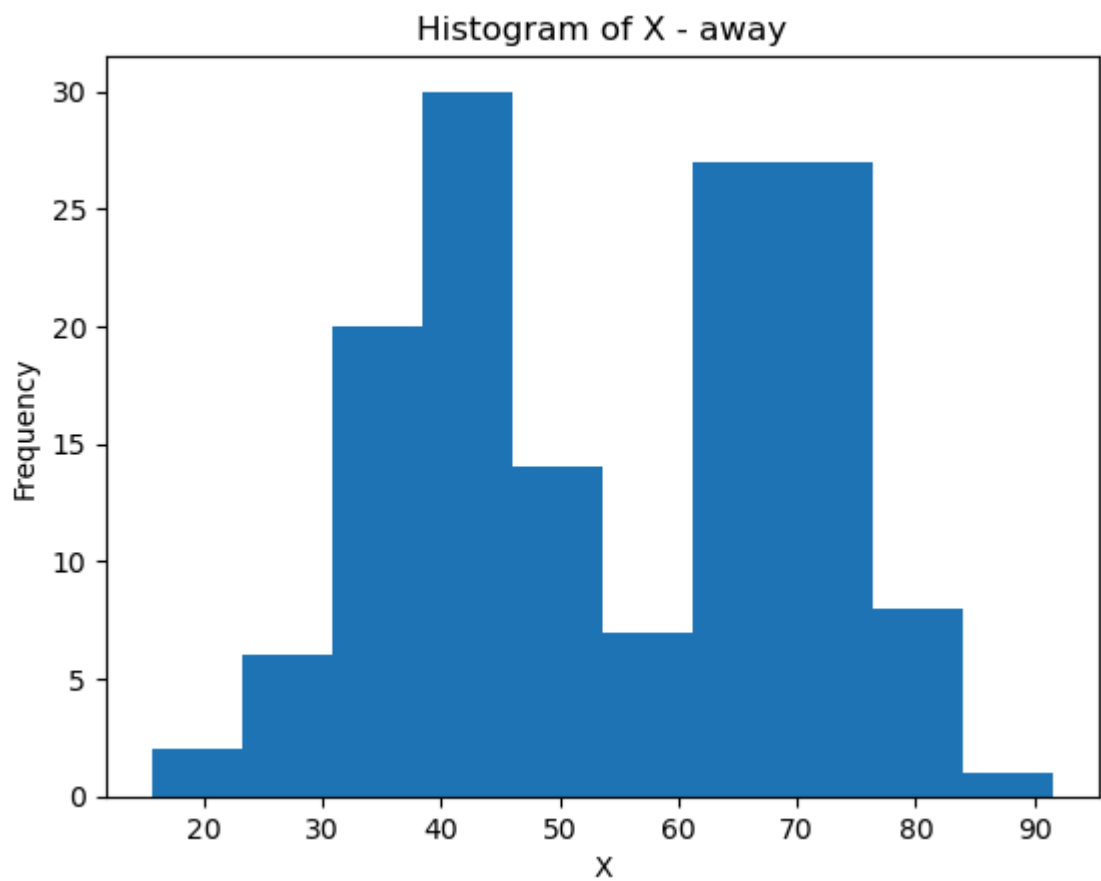
    # Crear un histograma para la columna 'y' con nombre "Histogram of Y - {categoria}"
    plt.hist(df_categoria['y'])
    plt.xlabel('Y')
    plt.ylabel('Frequency')
    plt.title(f'Histogram of Y - {categoria}')
    plt.show()

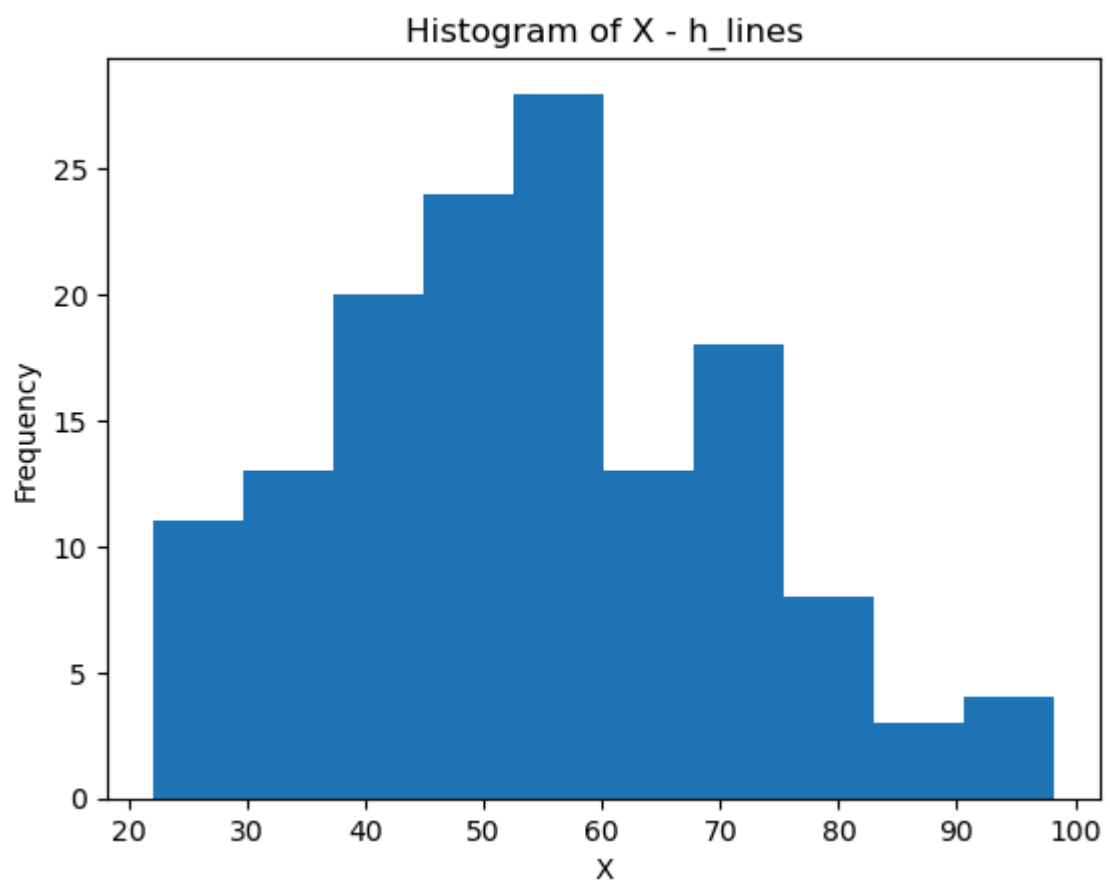
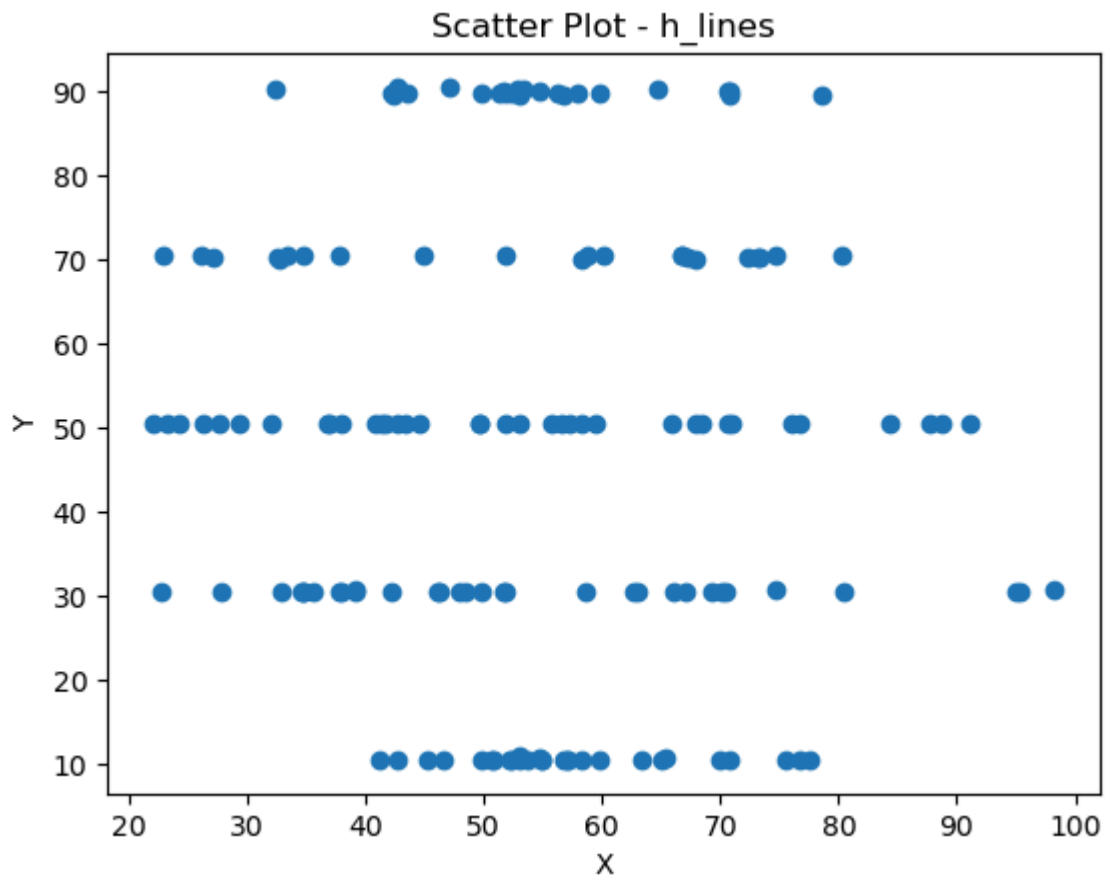
```

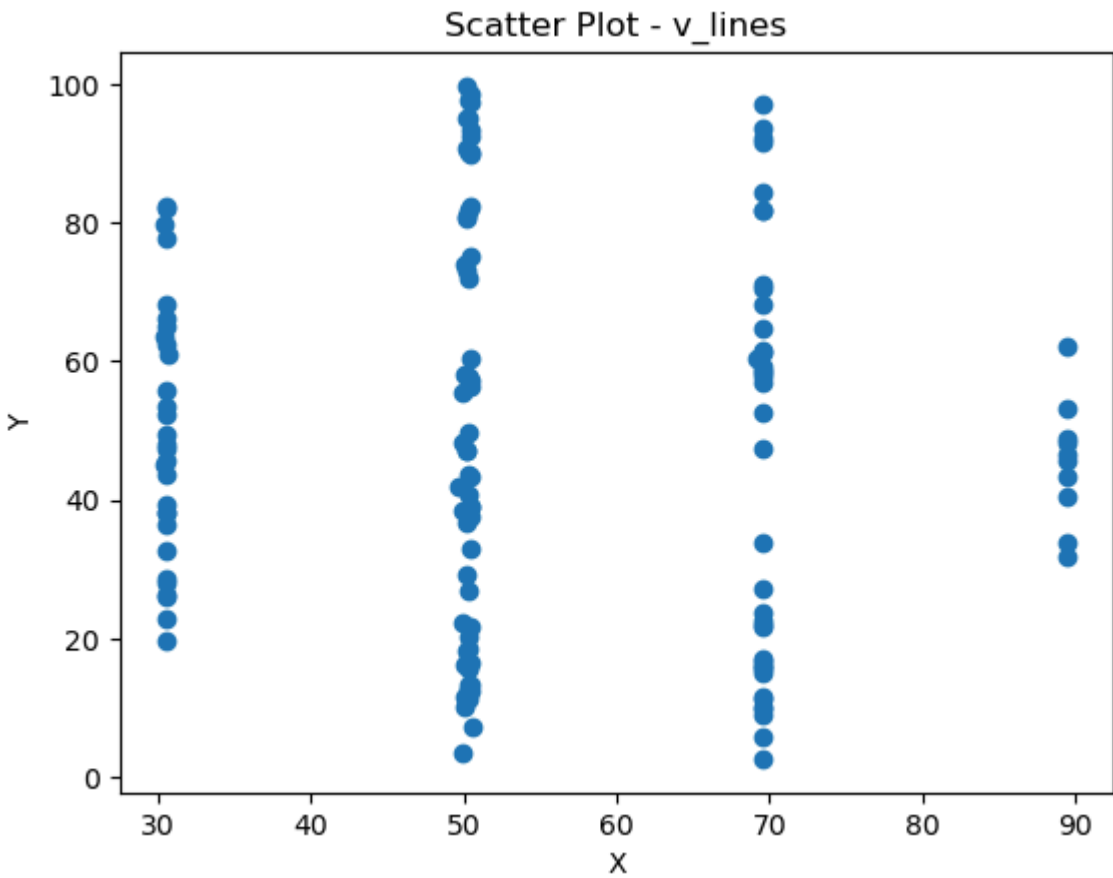
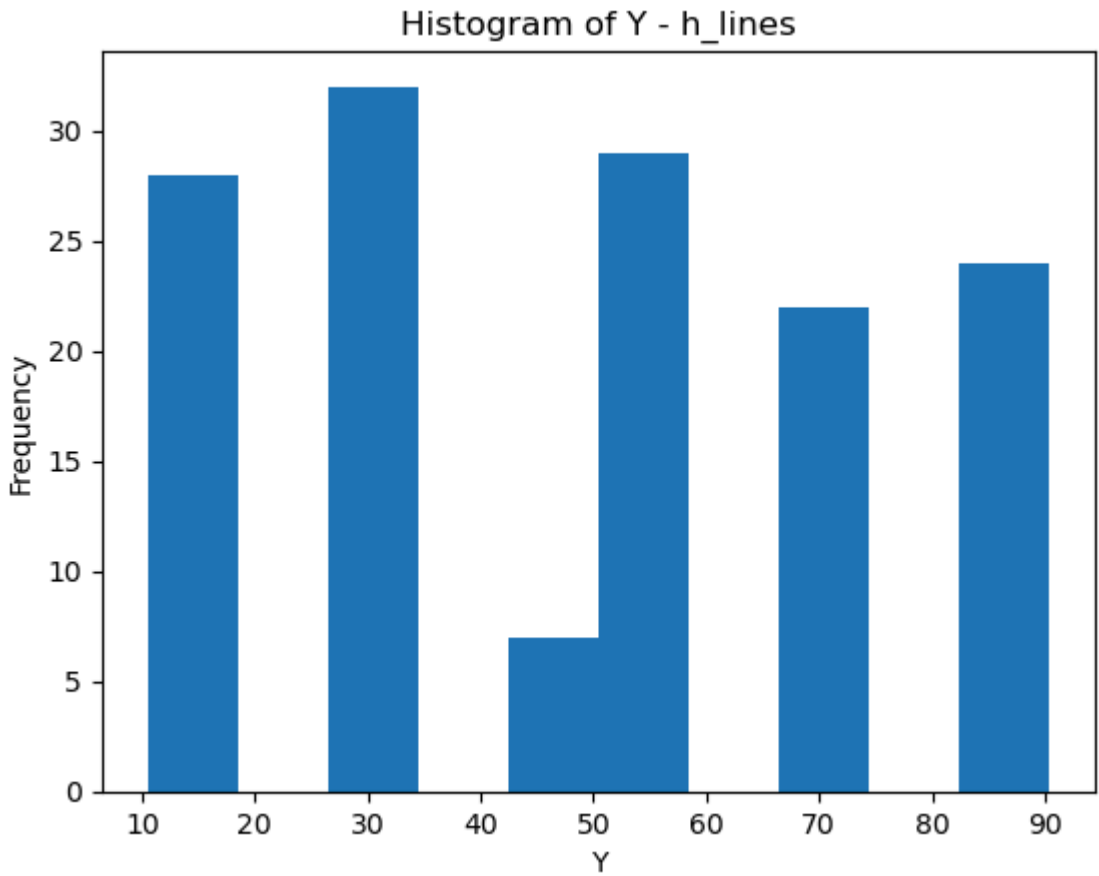
	dataset	x	y
0	dino	55.3846	97.1795
1	dino	51.5385	96.0256
2	dino	46.1538	94.4872
3	dino	42.8205	91.4103
4	dino	40.7692	88.3333

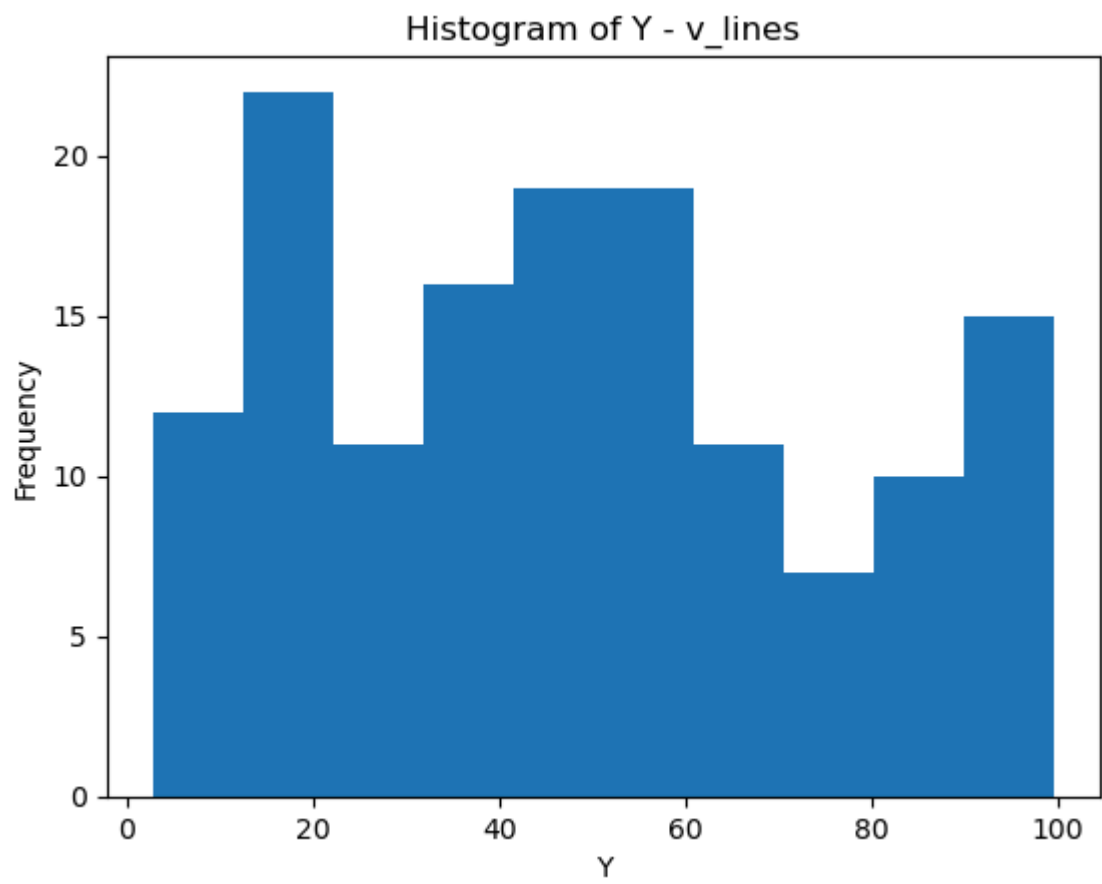
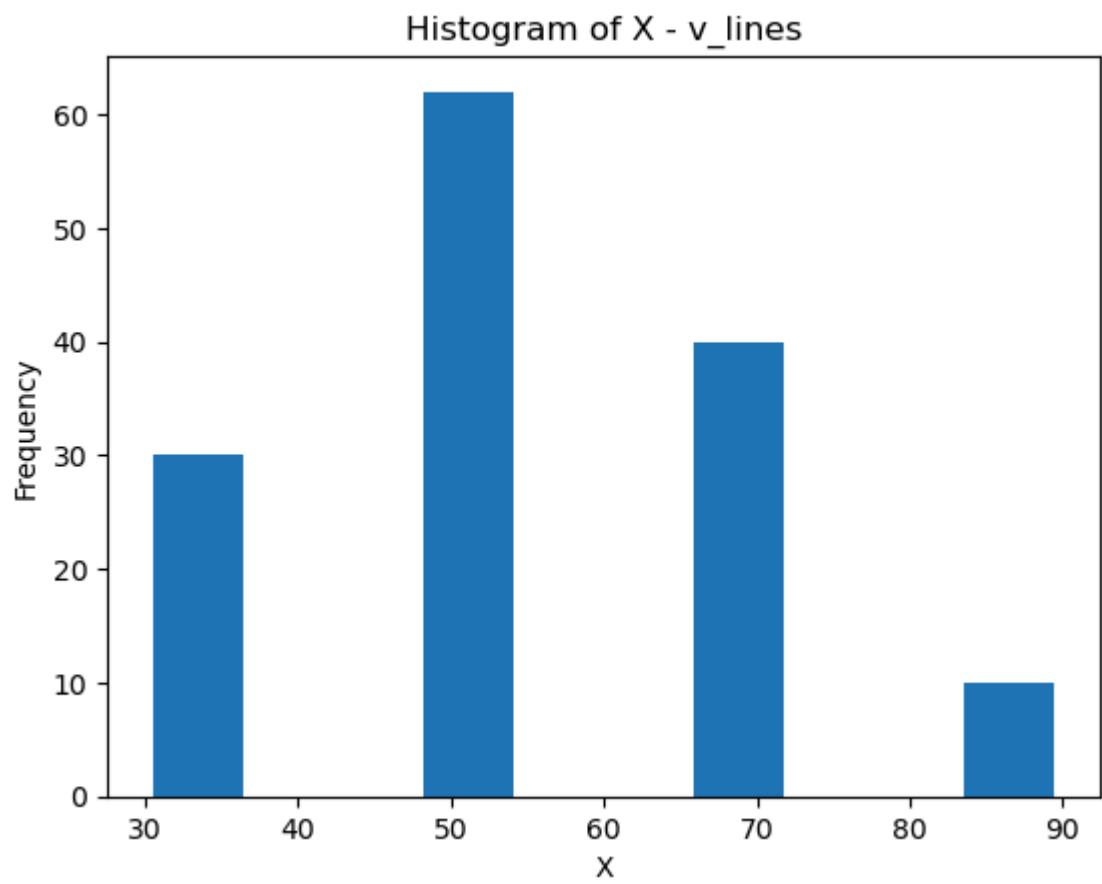


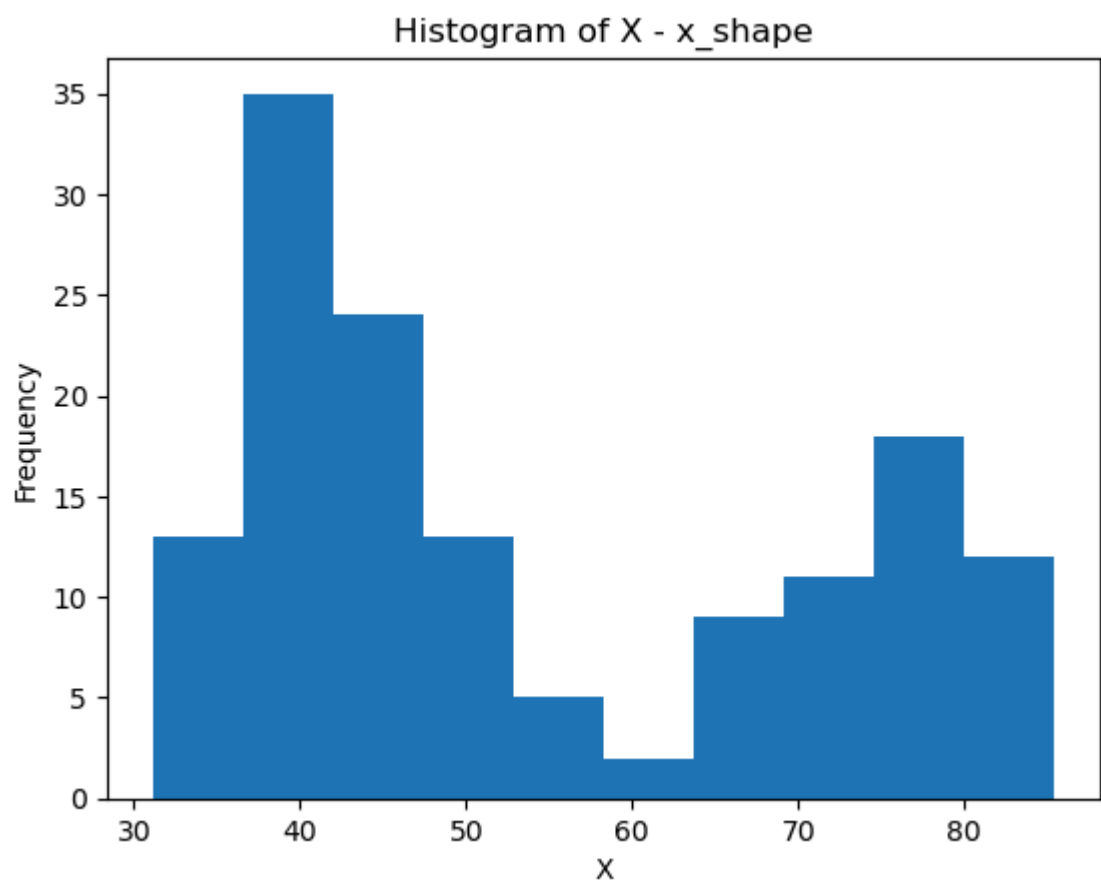
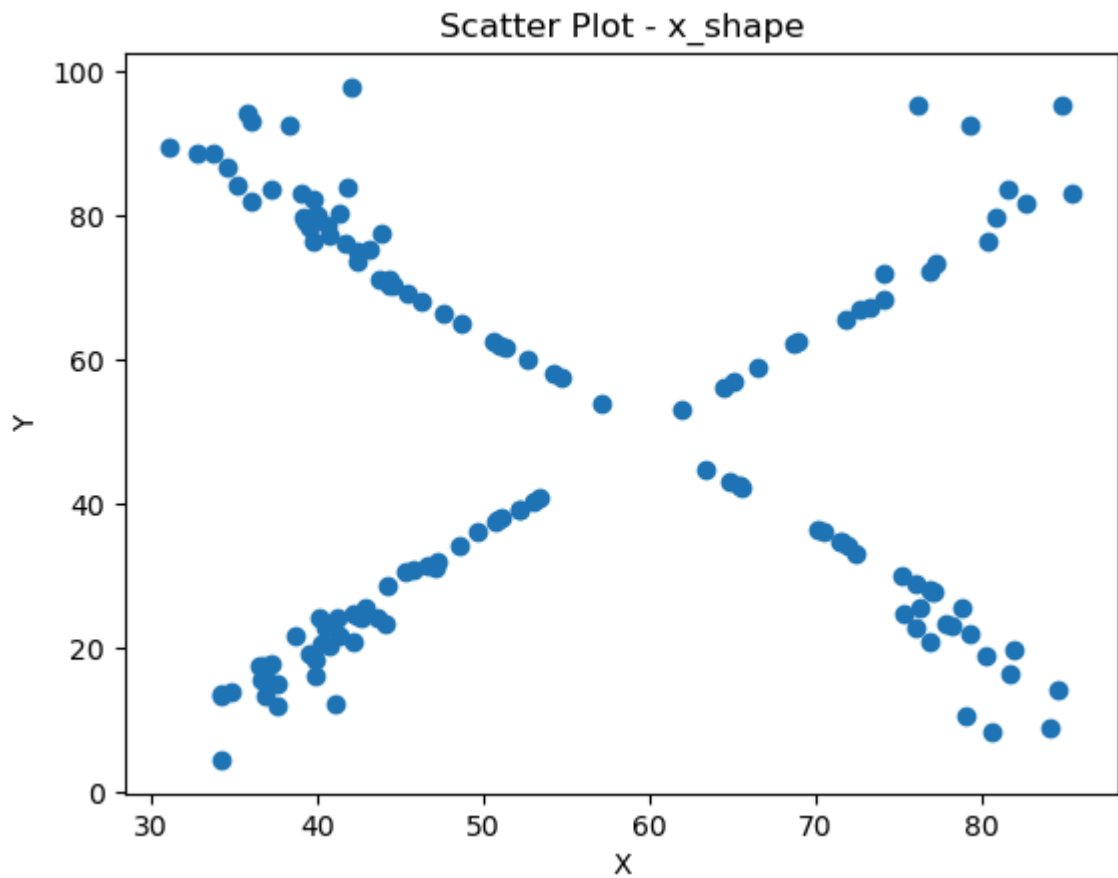




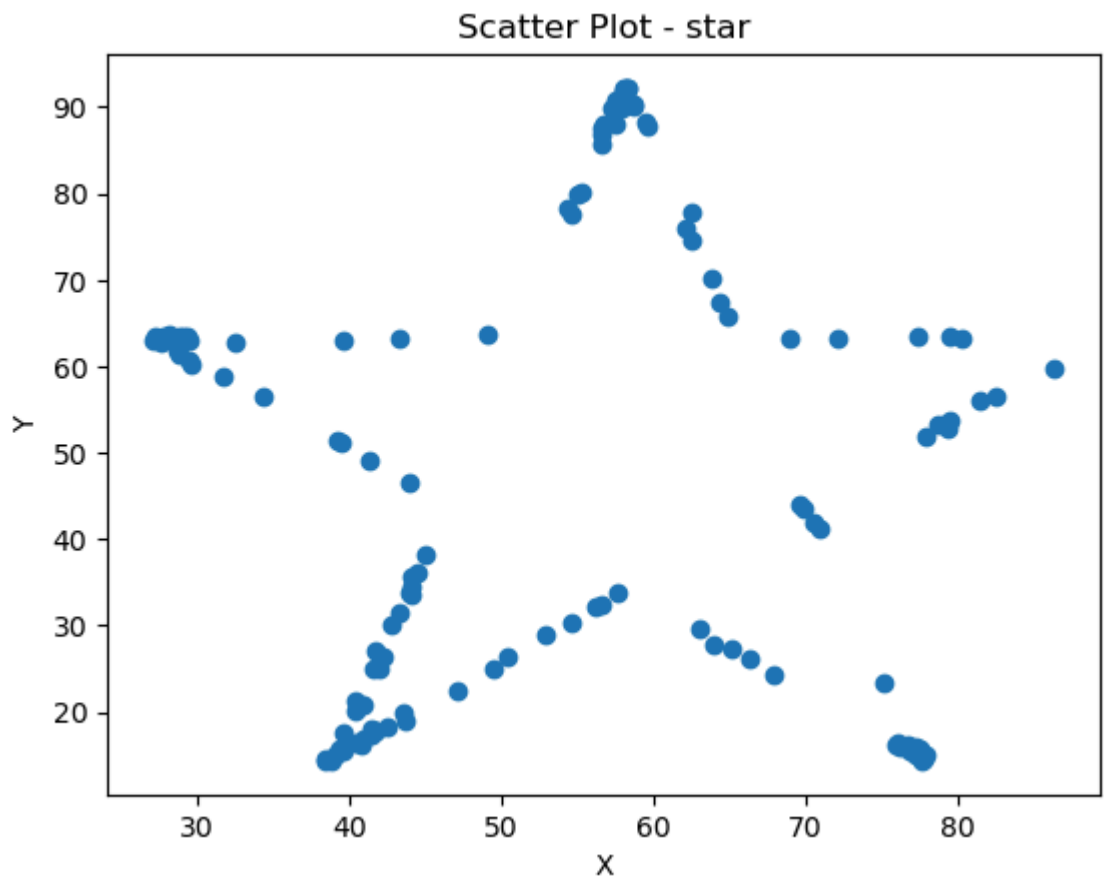
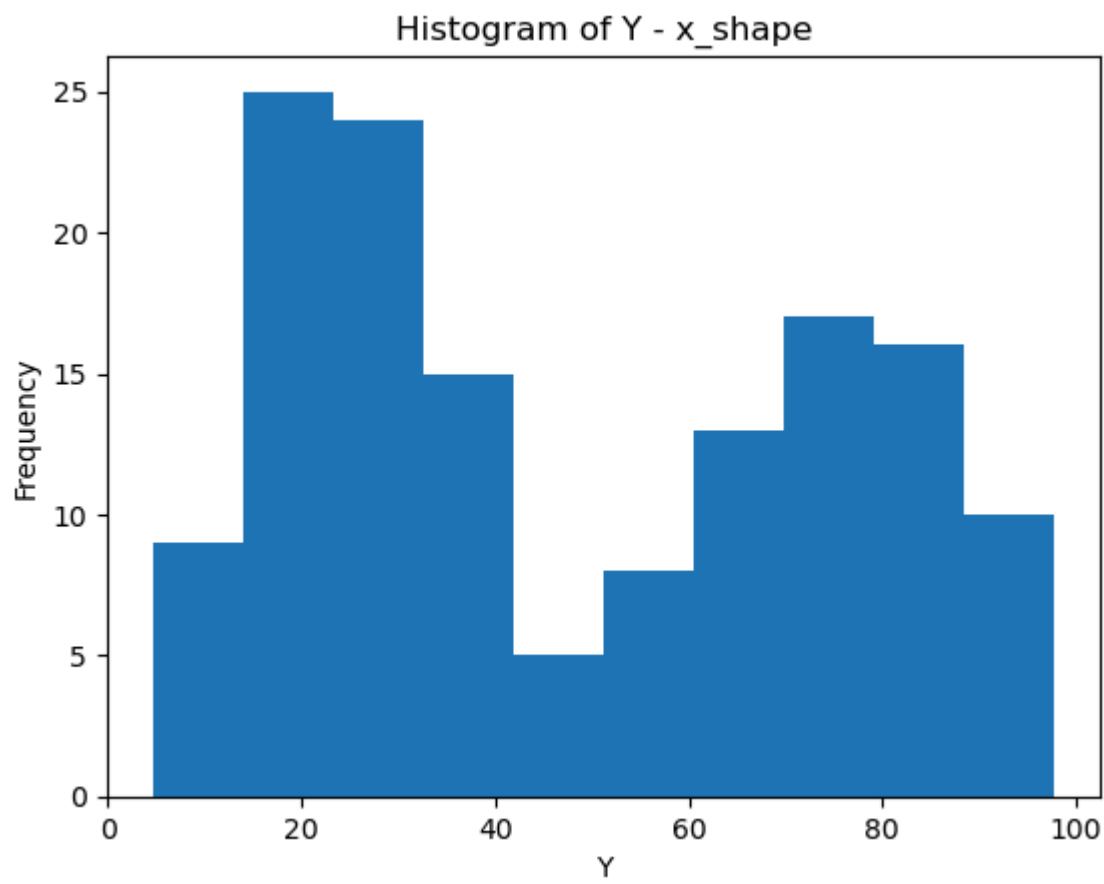




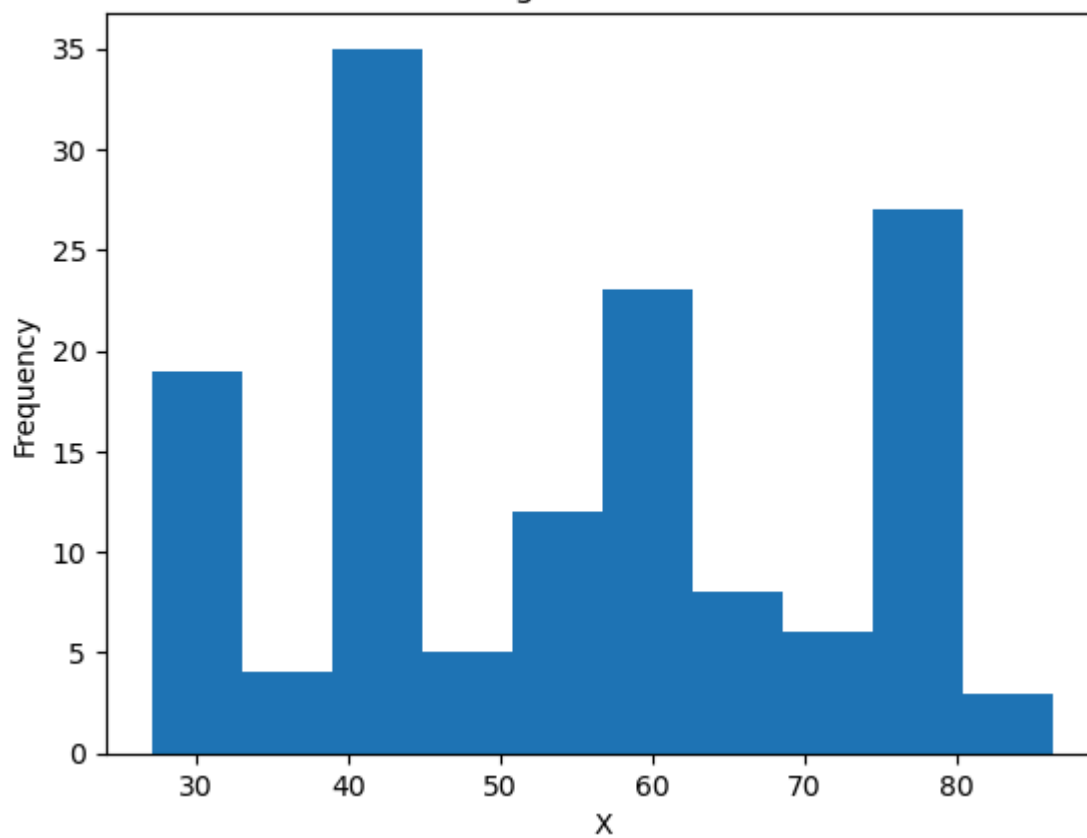




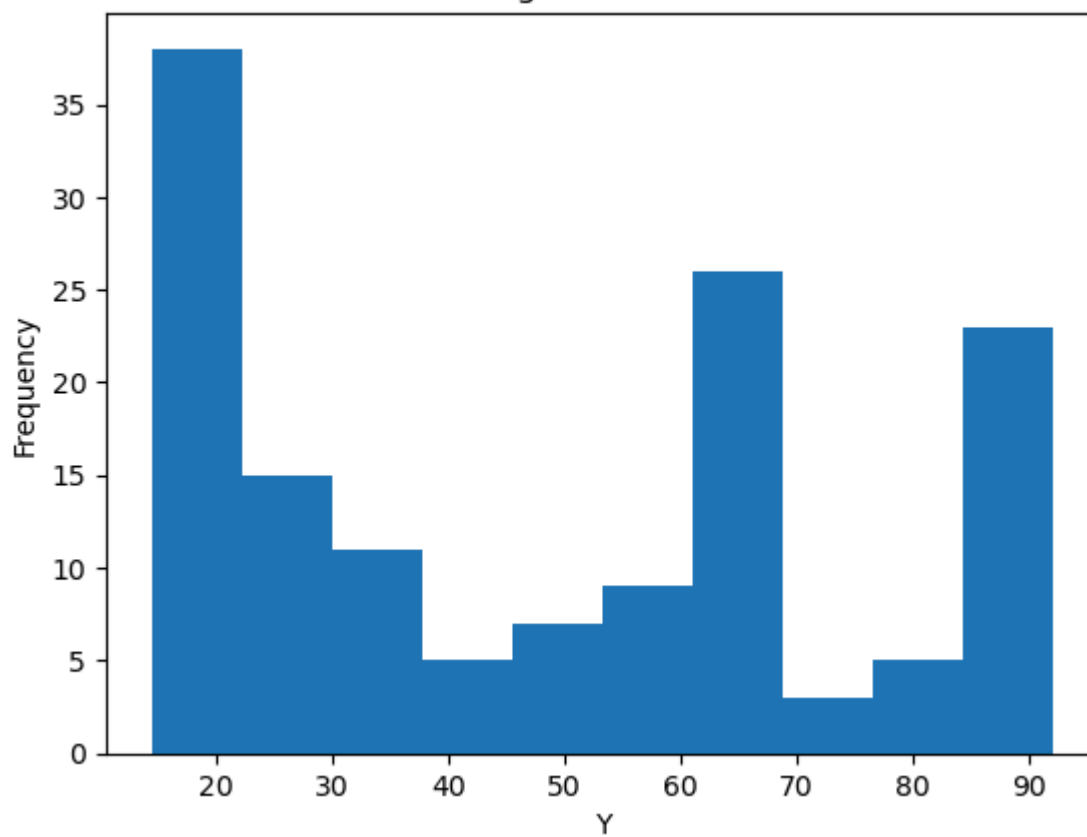


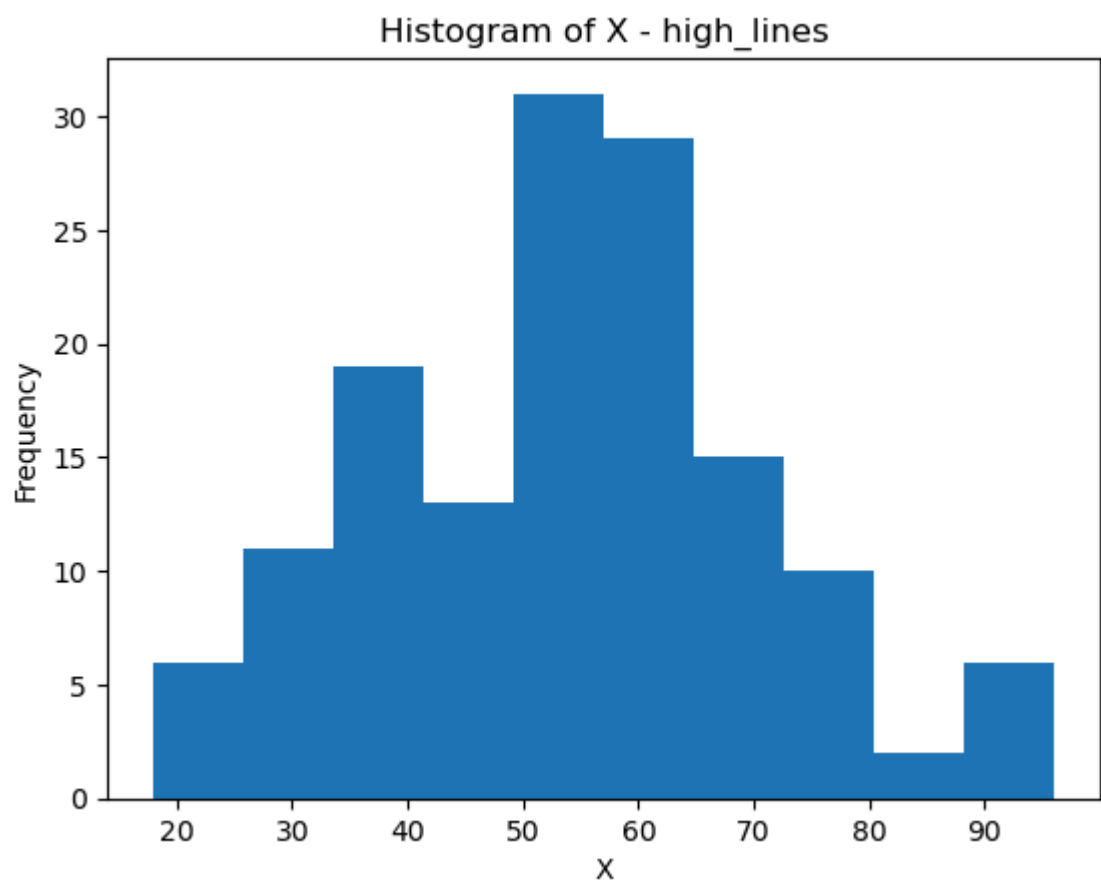
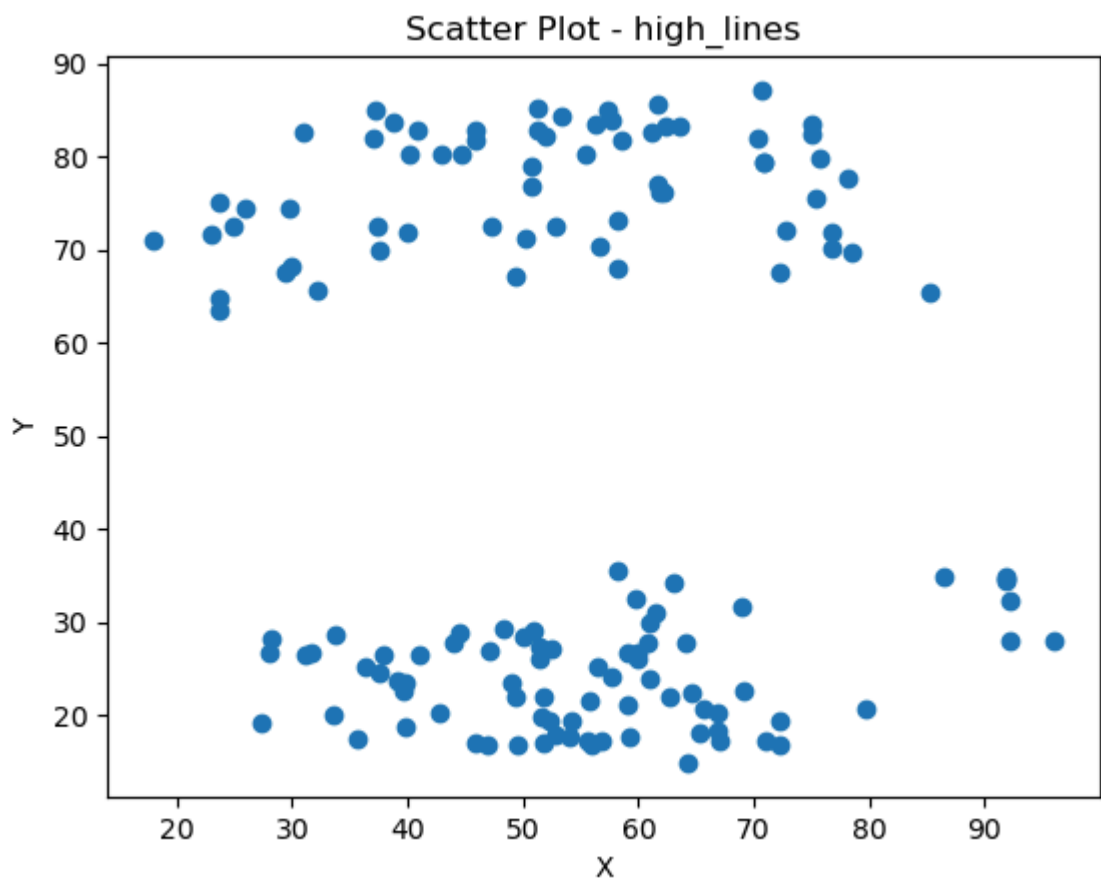


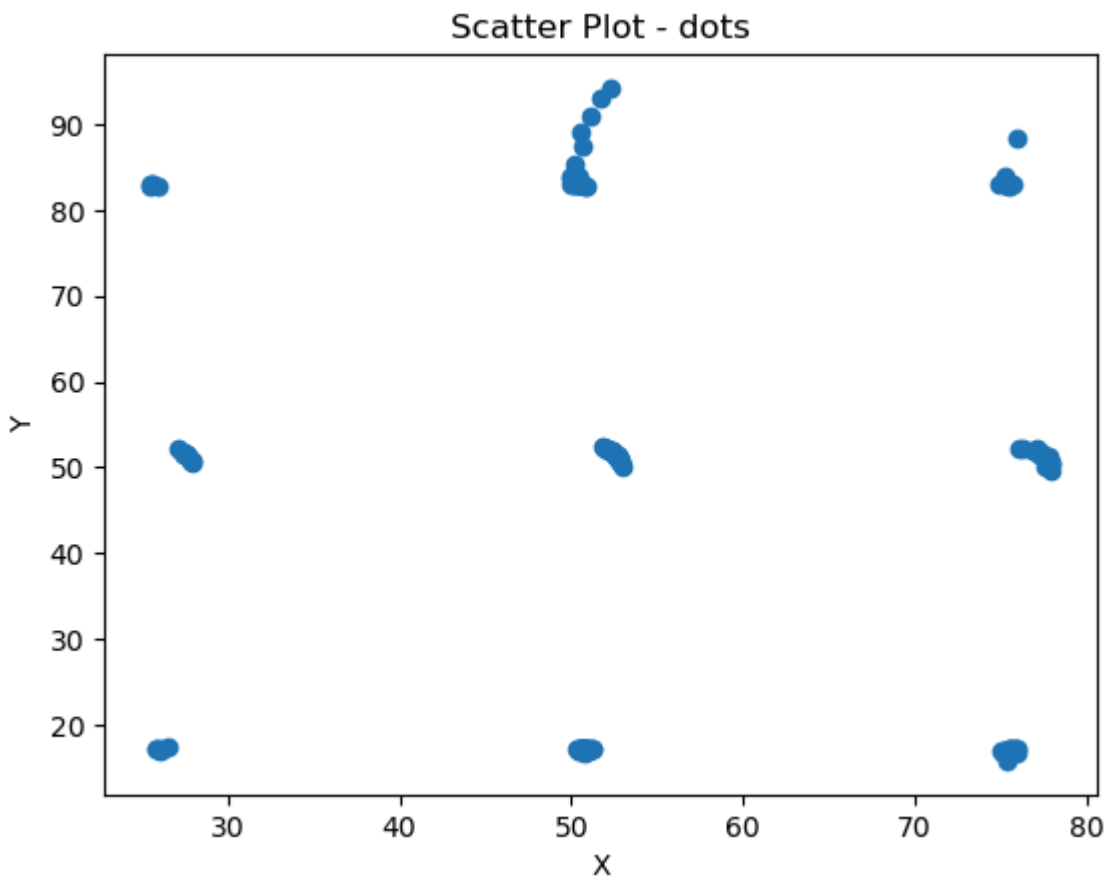
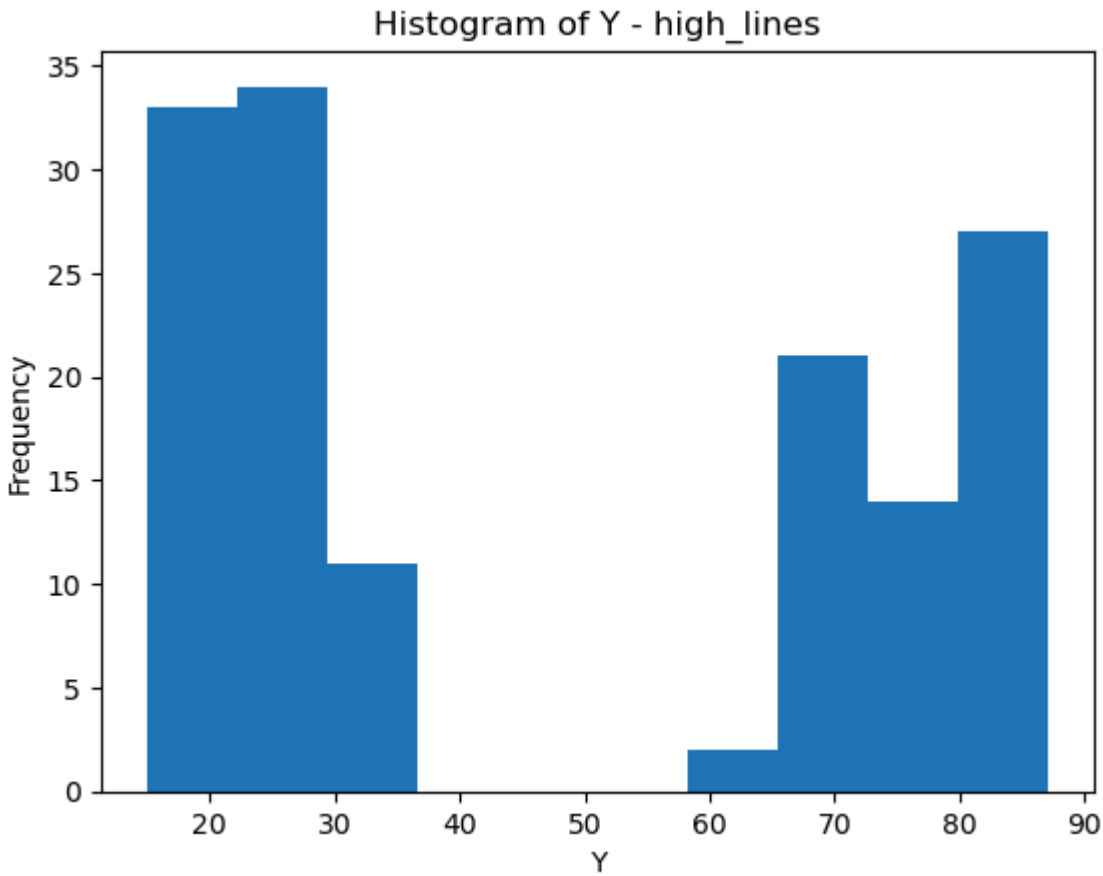
Histogram of X - star

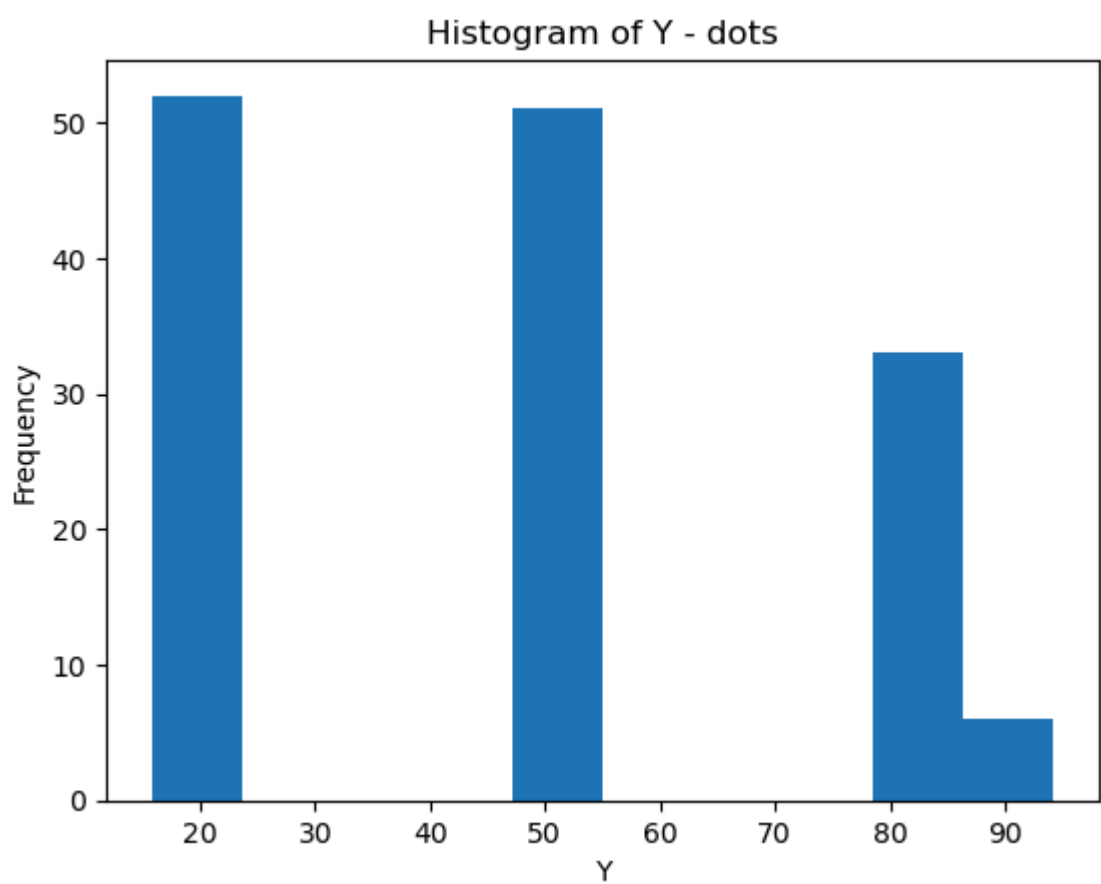
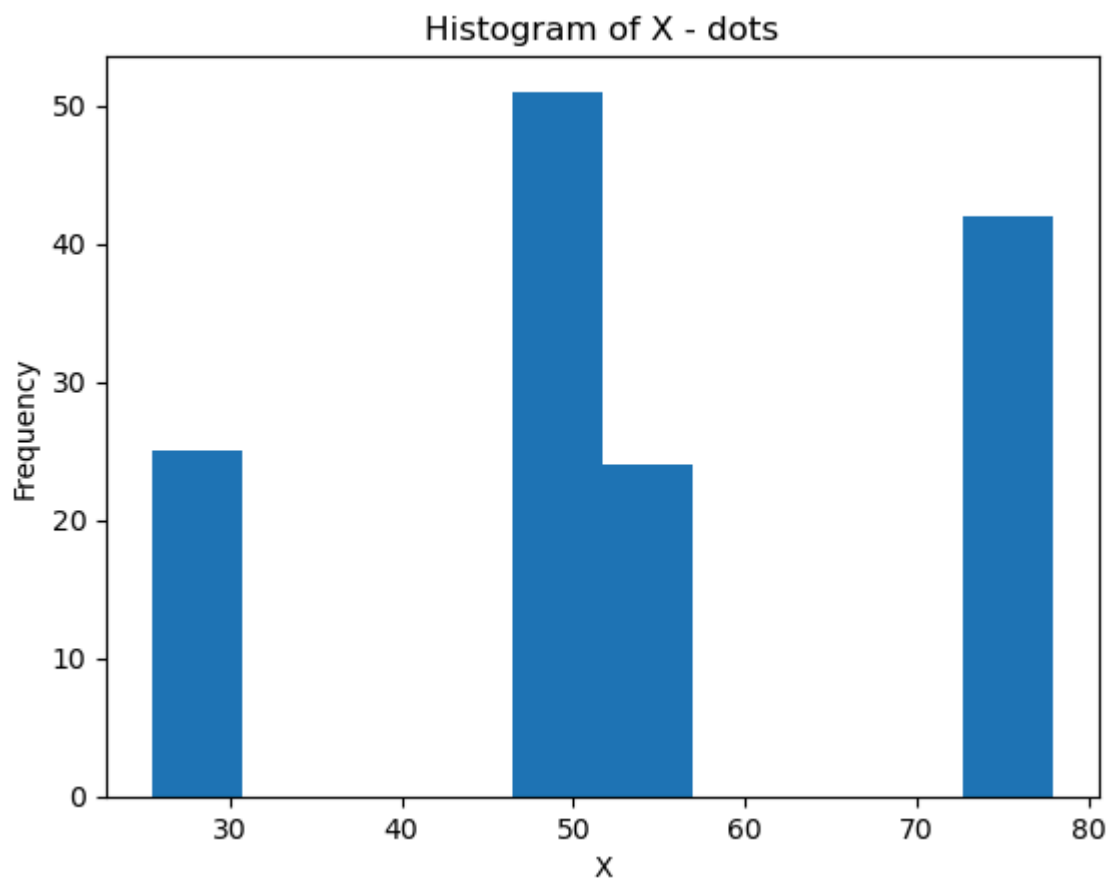


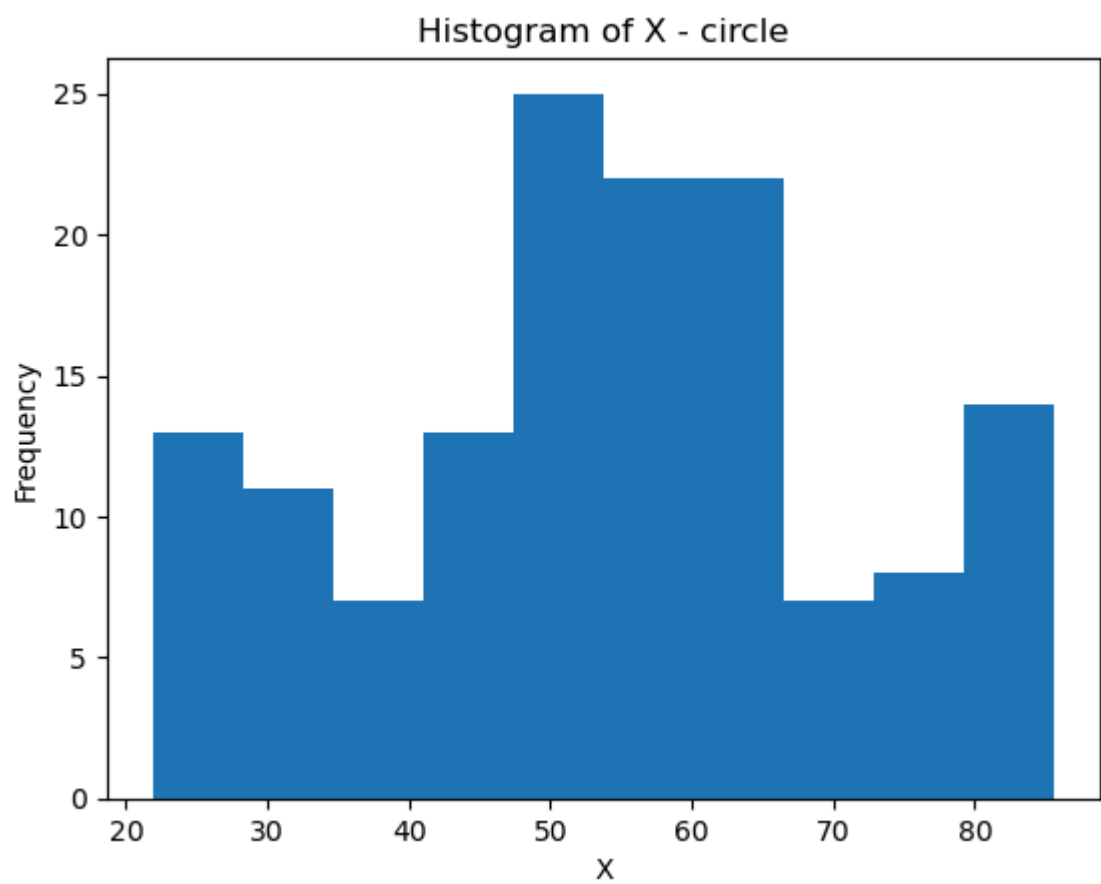
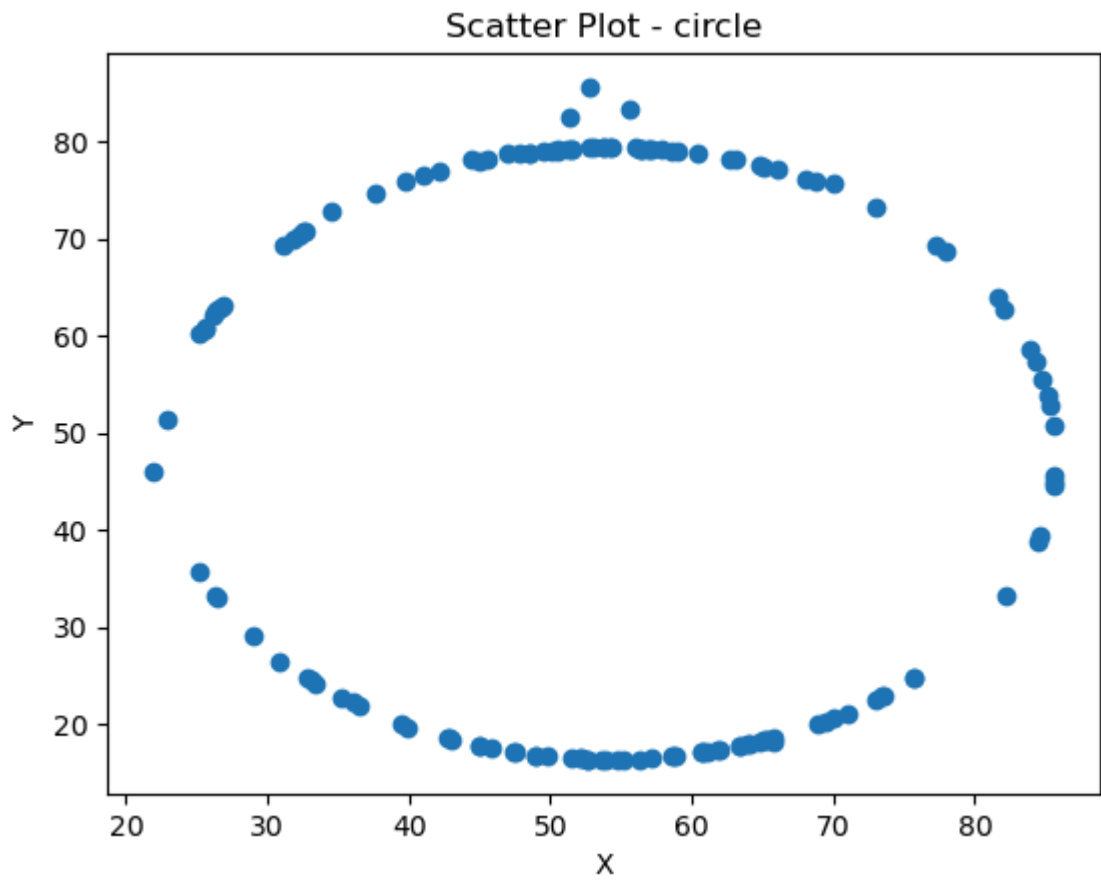
Histogram of Y - star

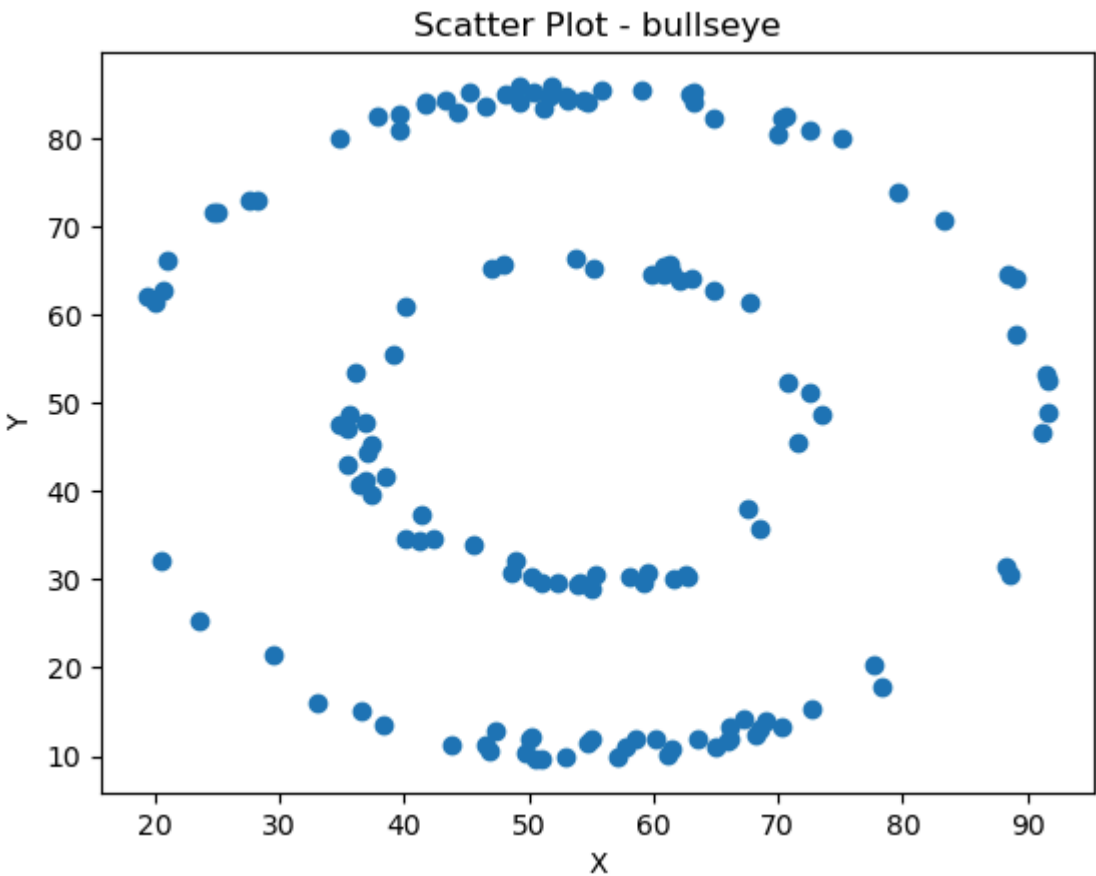
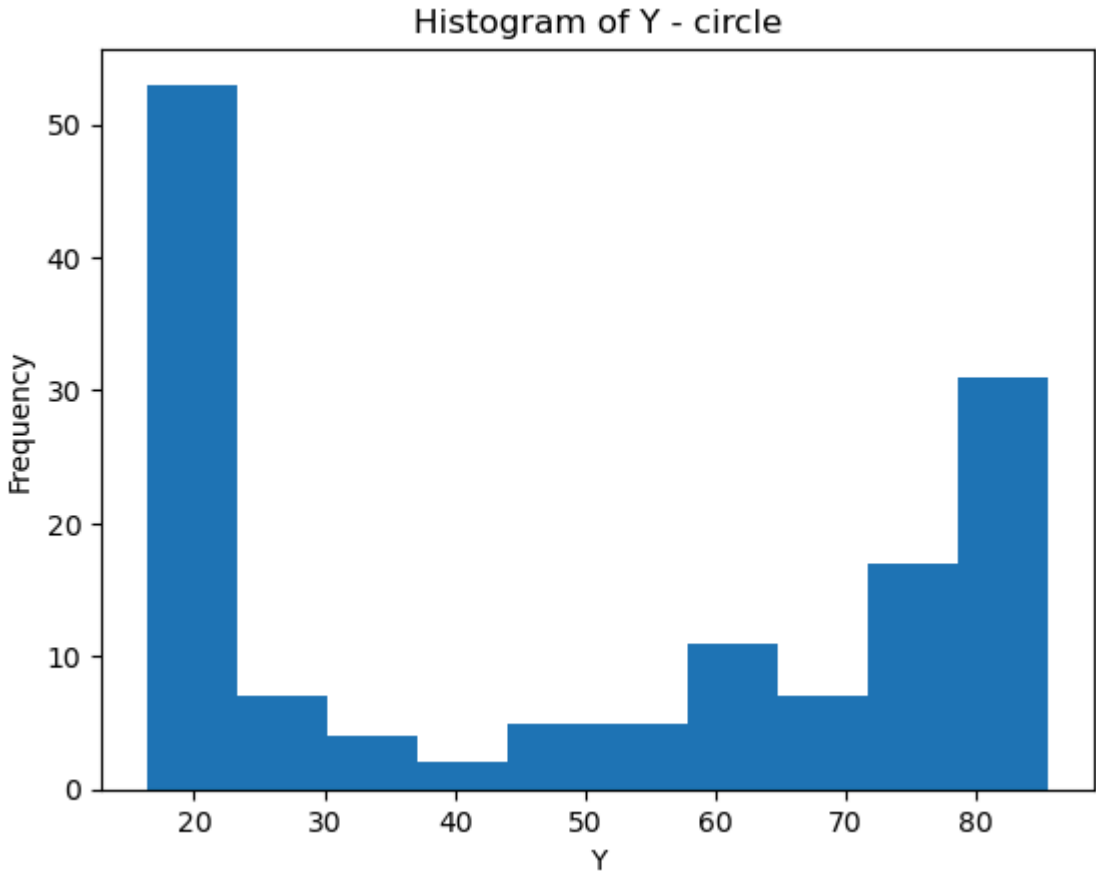


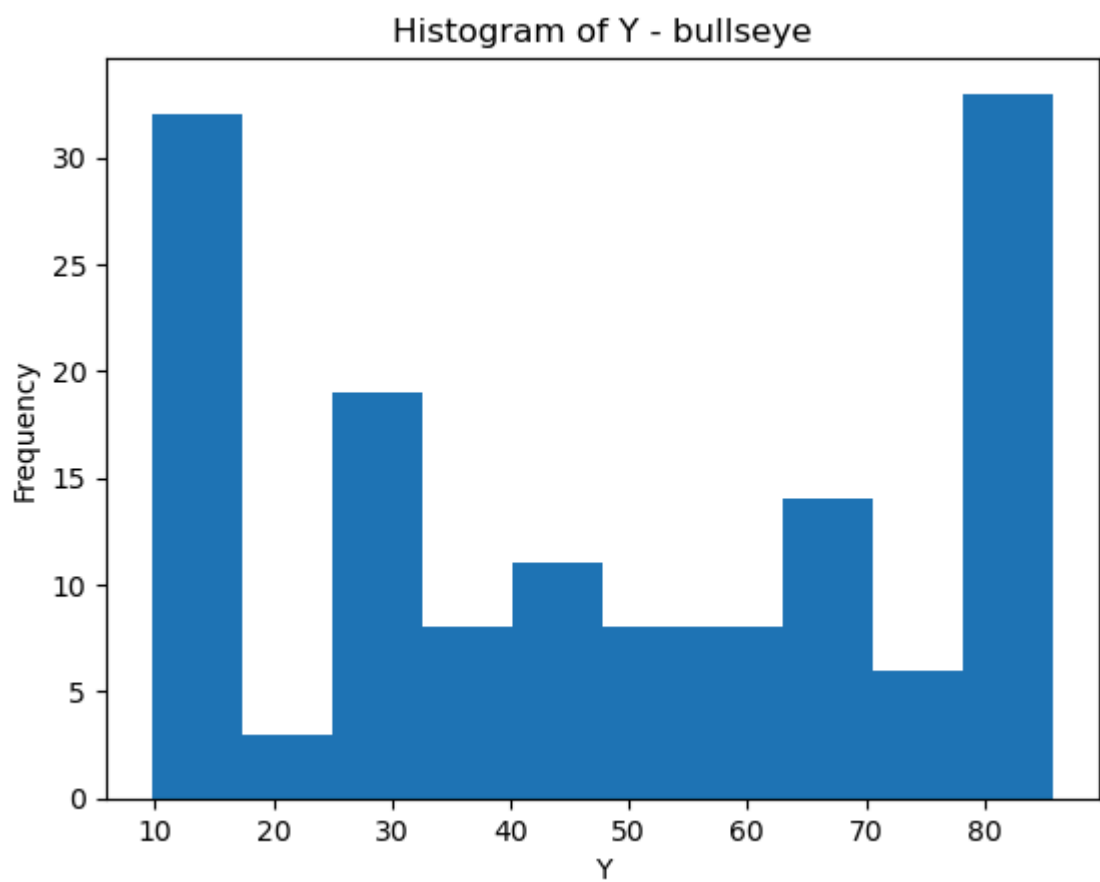
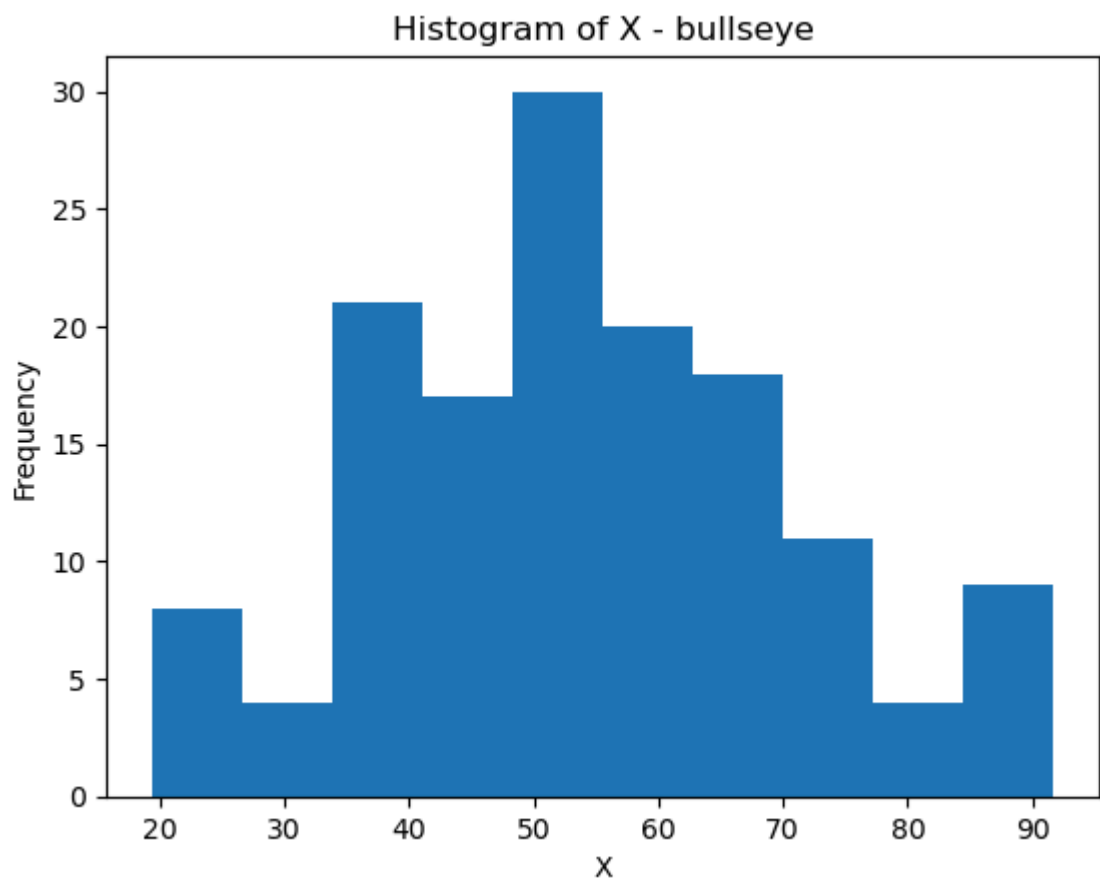




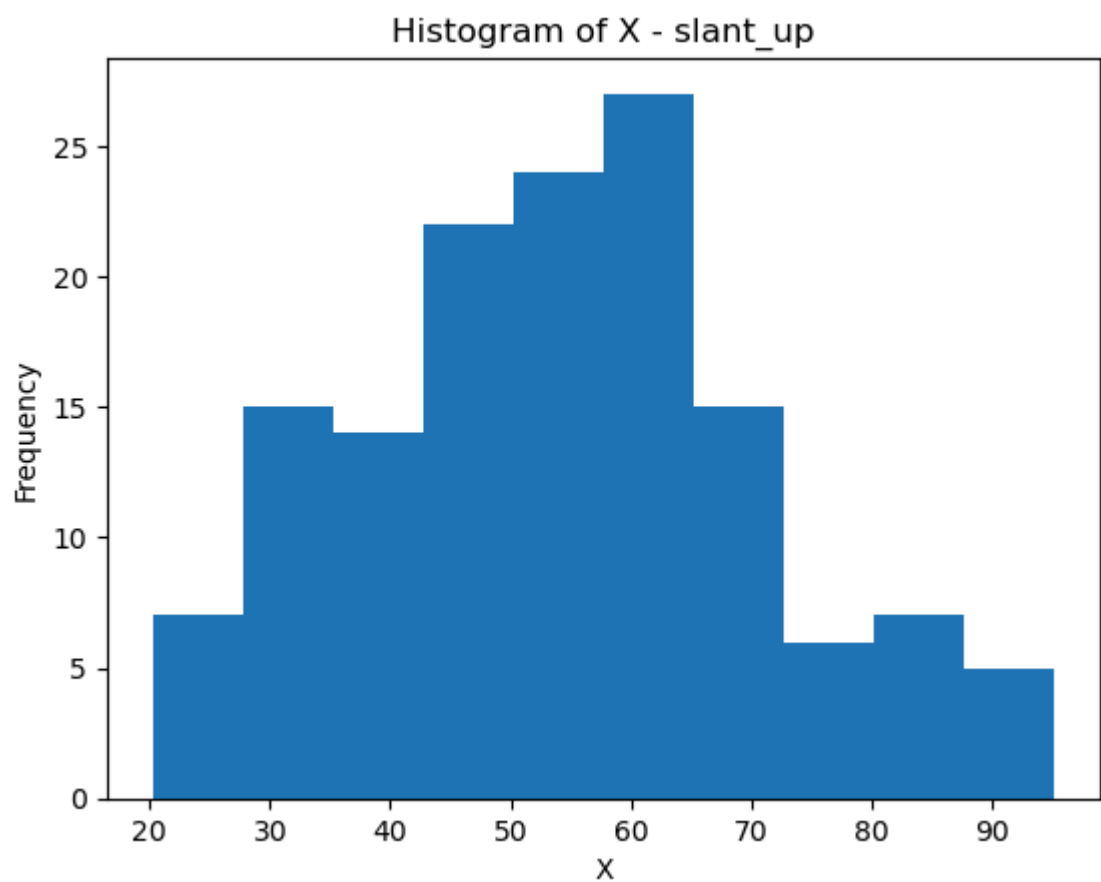
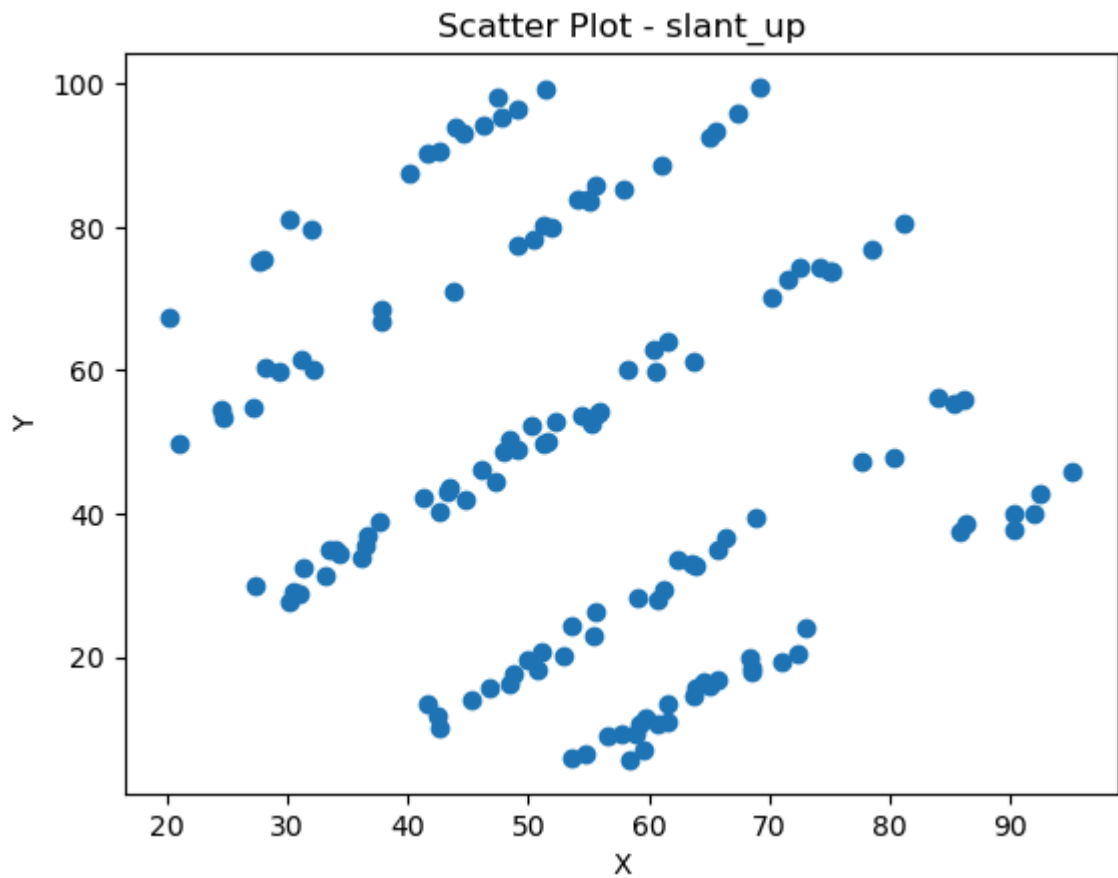


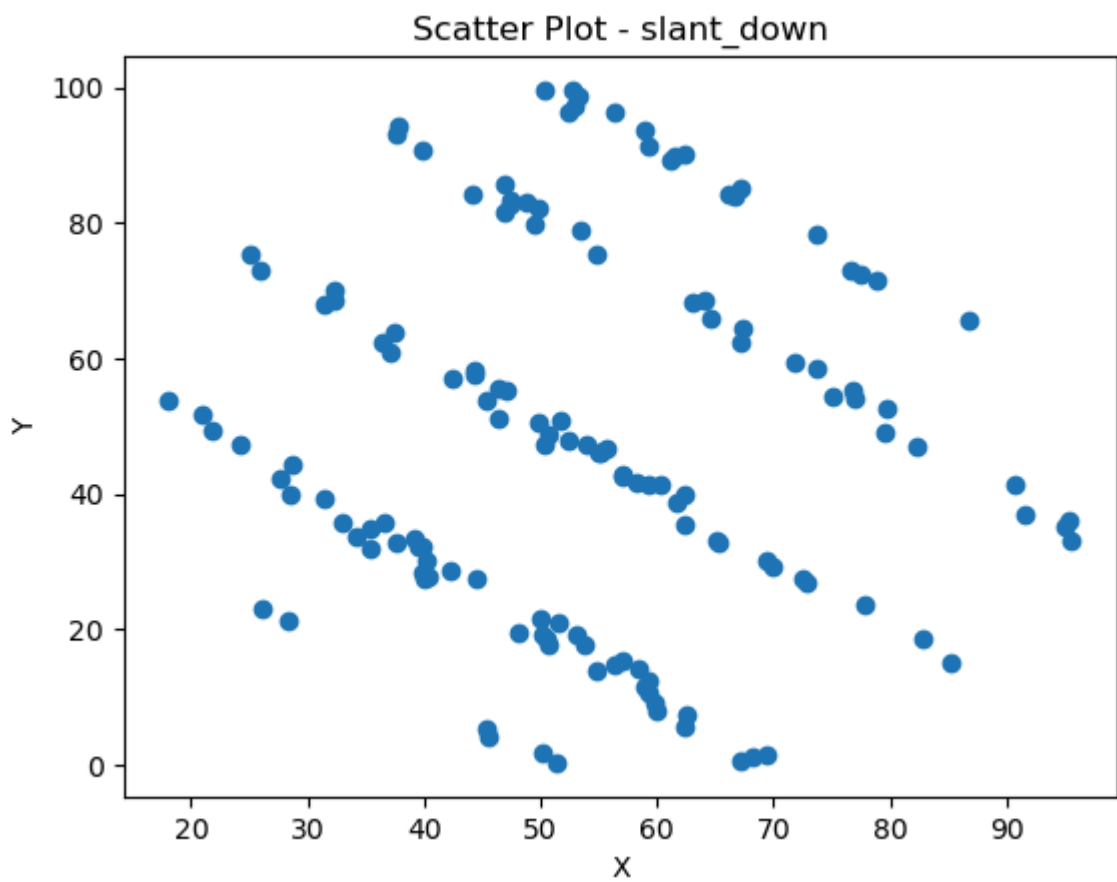
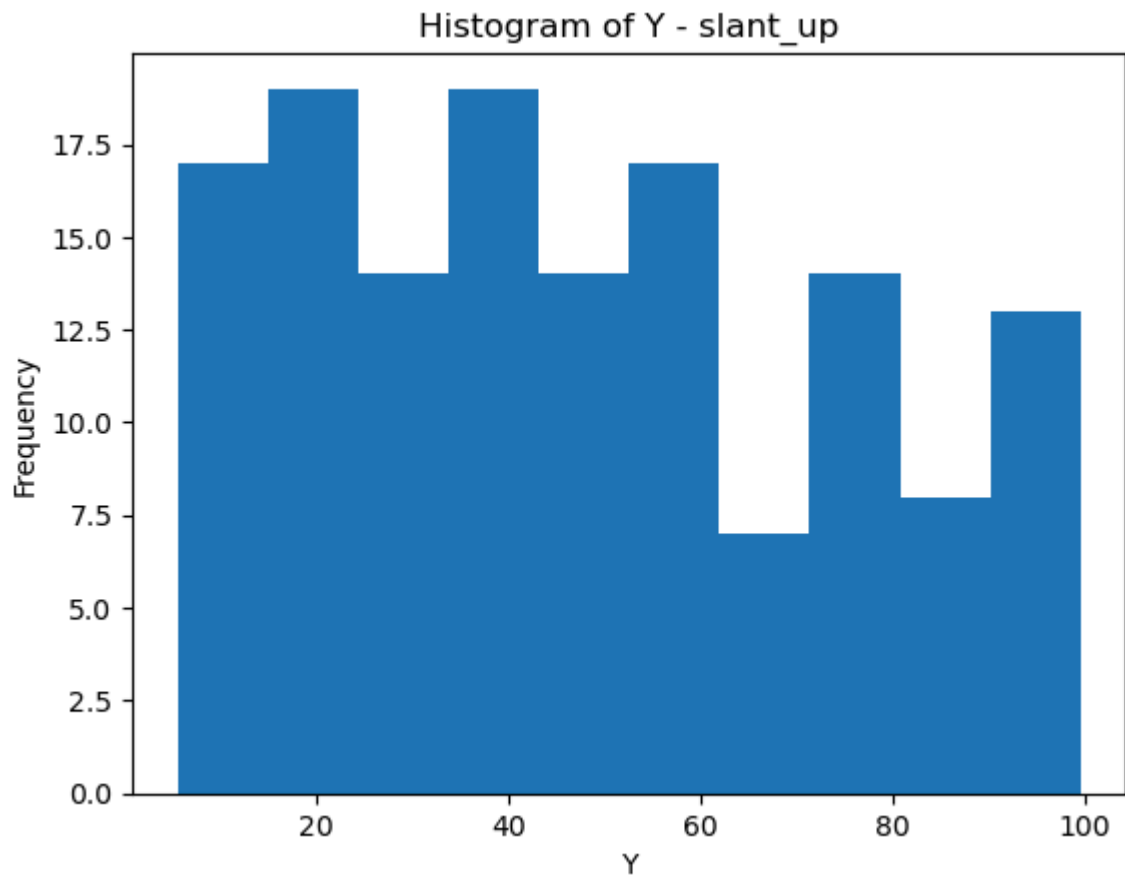


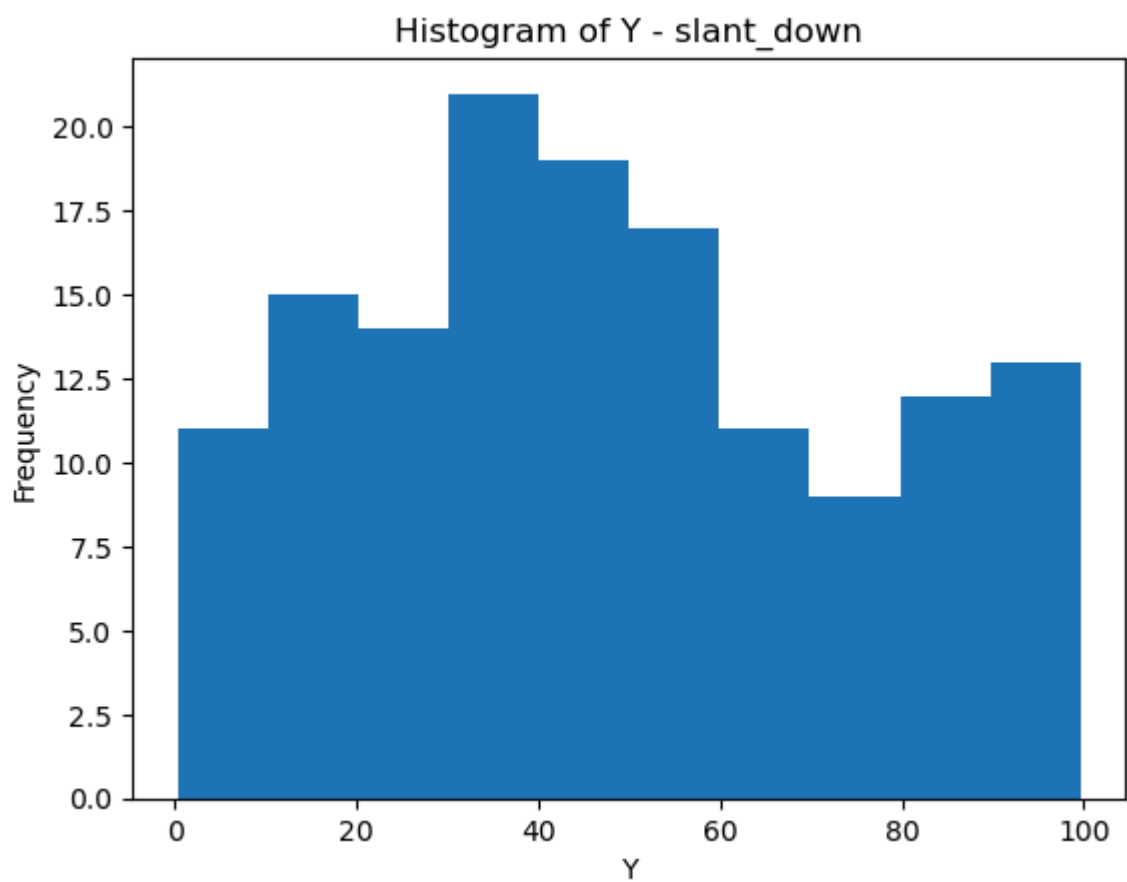
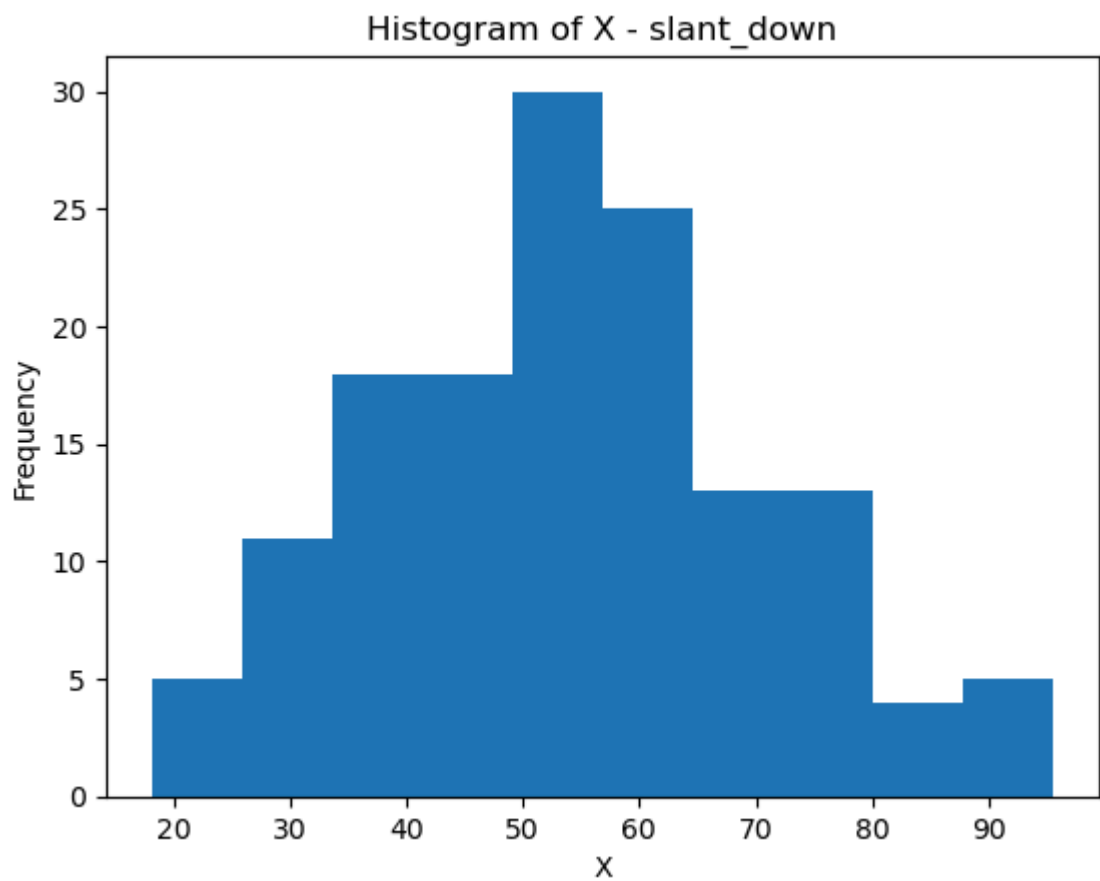


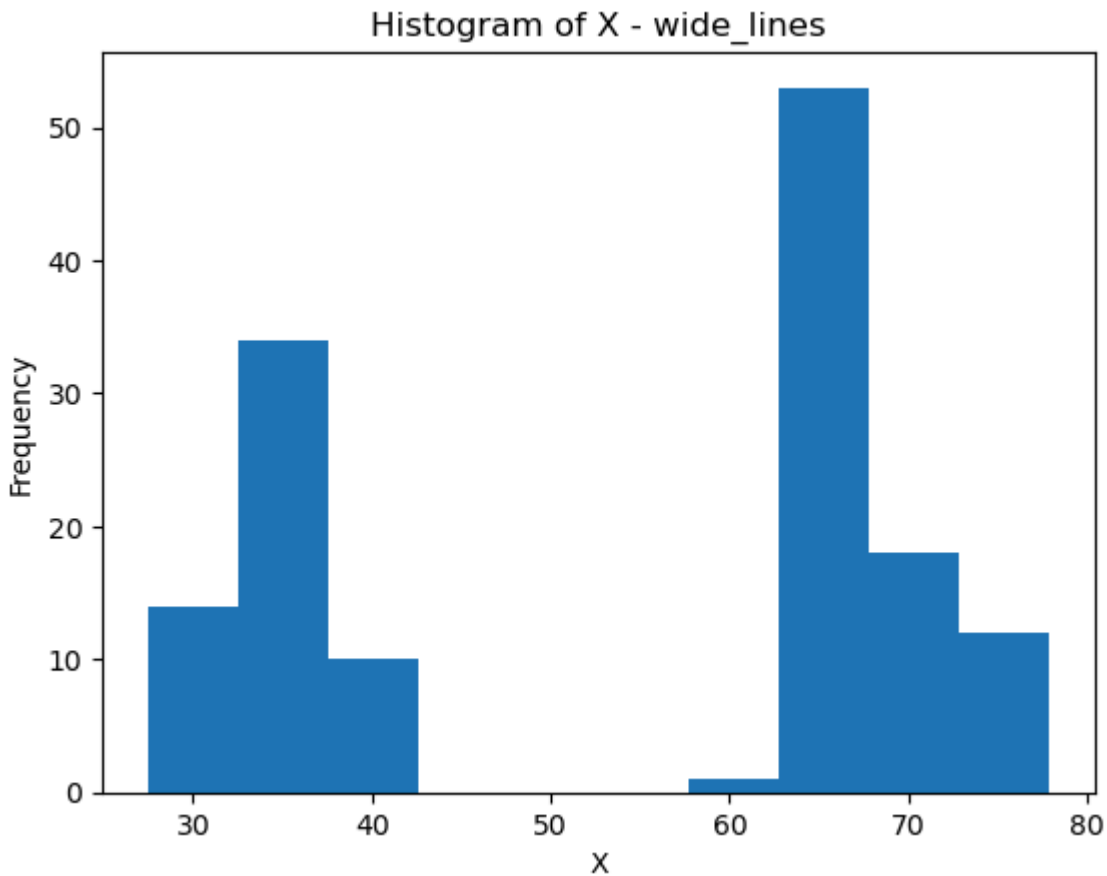
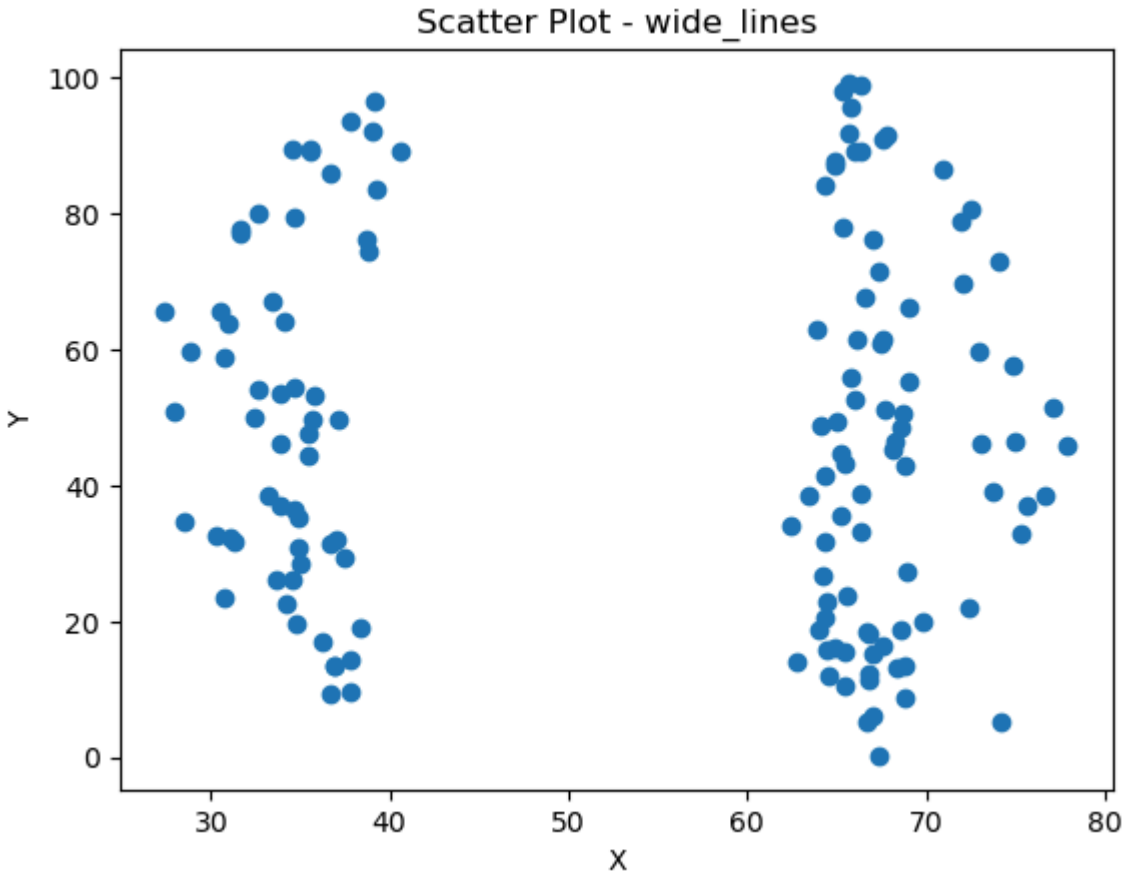


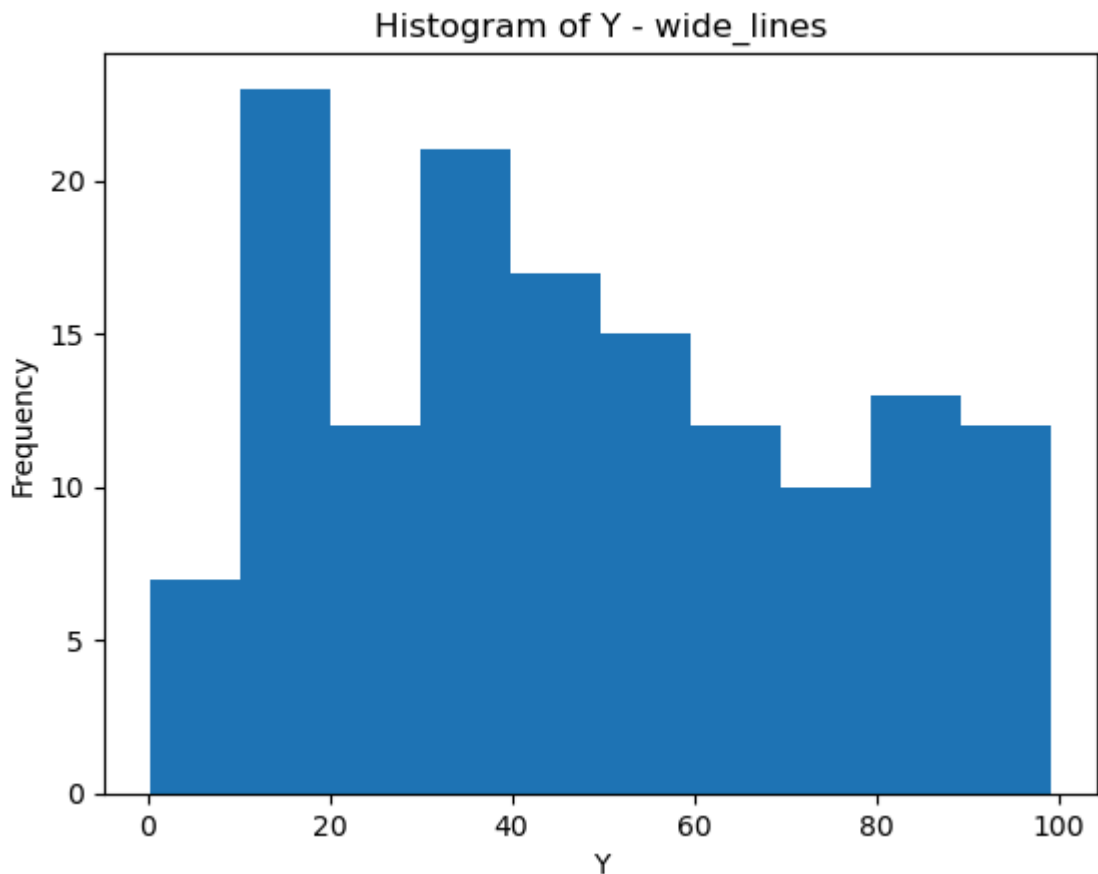












```
In [2]: import pandas as pd
import matplotlib.pyplot as plt

# Cargar el archivo CSV en un DataFrame de pandas
df = pd.read_csv('datasauRUS.csv')

# Mostrar las primeras filas del DataFrame para verificar la carga de datos
print(df.head())

# Filtrar el DataFrame para obtener solo las filas correspondientes al dataset "estrella"
df_estrella = df[df['dataset'] == 'star']

# Mostrar las primeras filas del DataFrame filtrado para verificar
print(df_estrella.head())

# Crear un gráfico de dispersión para las columnas 'x' y 'y' del dataset "estrella"
plt.scatter(df_estrella['x'], df_estrella['y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot - Dataset Estrella')
plt.show()

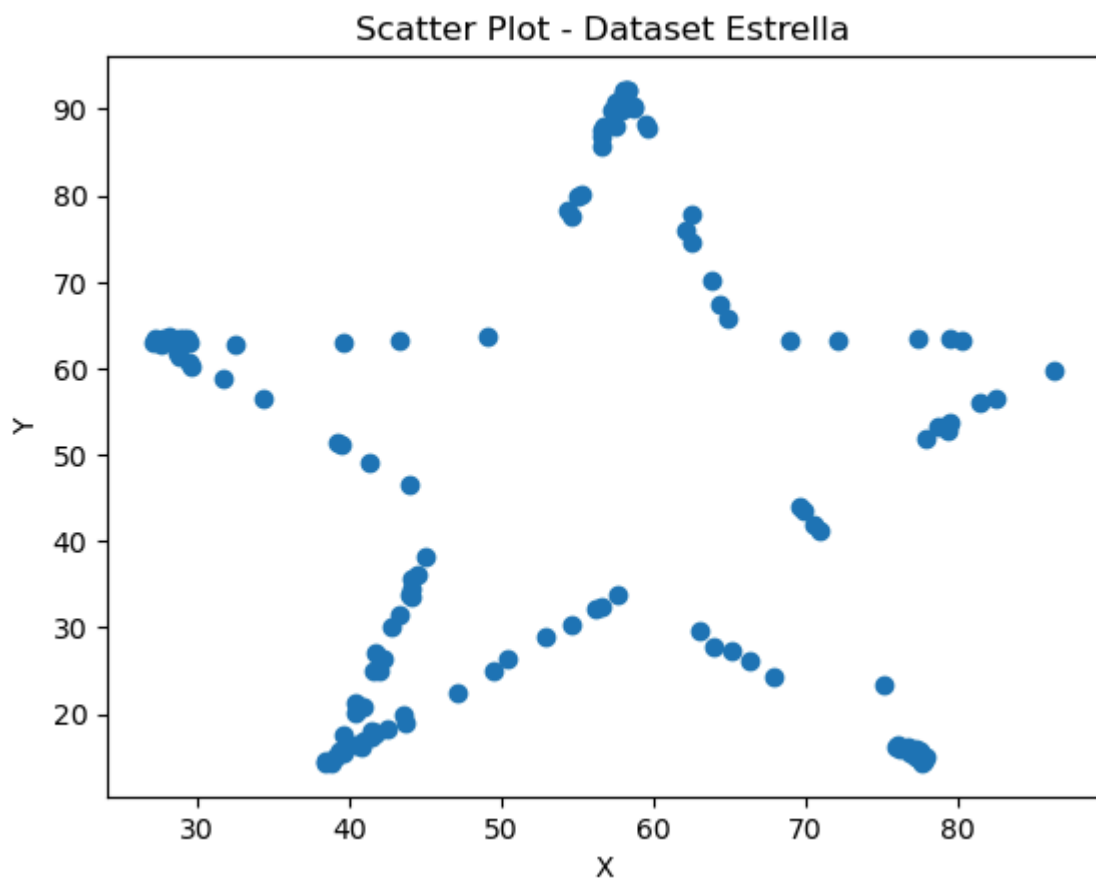
# Crear un histograma para la columna 'x' del dataset "estrella"
plt.hist(df_estrella['x'])
plt.xlabel('X')
plt.ylabel('Frequency')
plt.title('Histogram of X - Dataset Estrella')
plt.show()

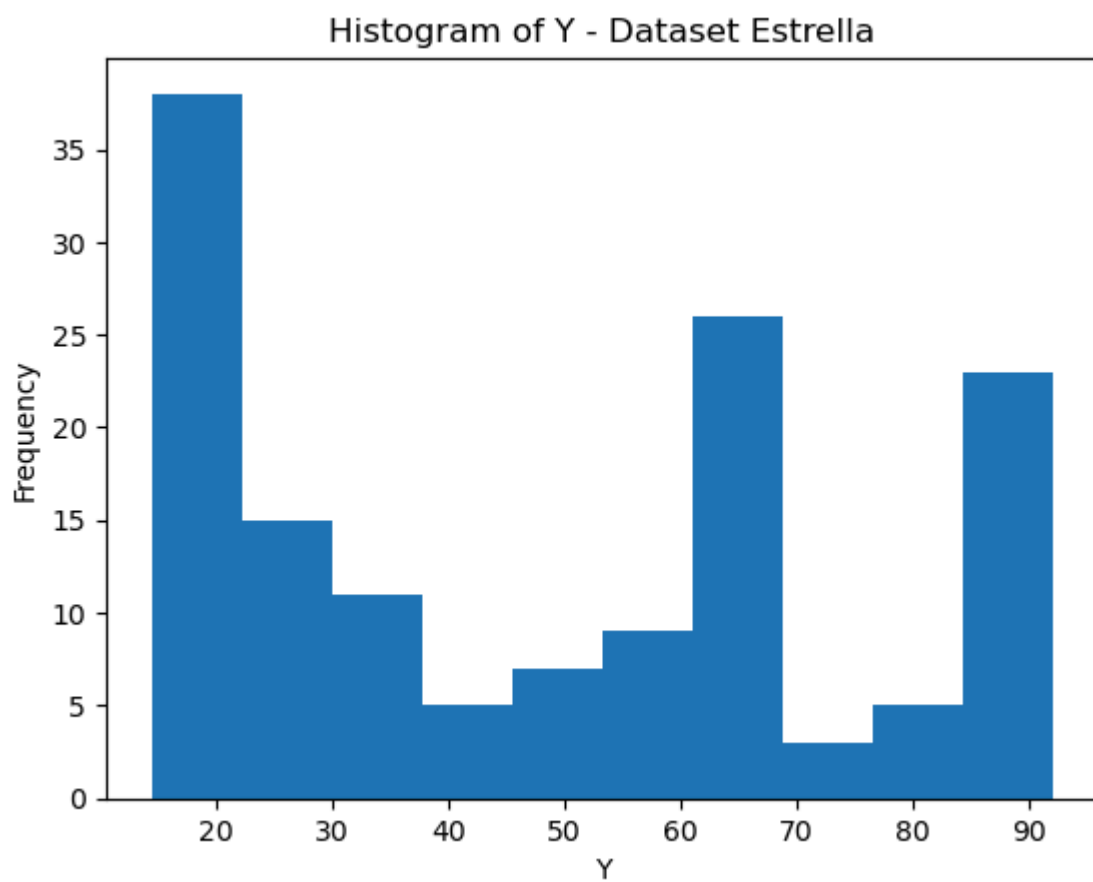
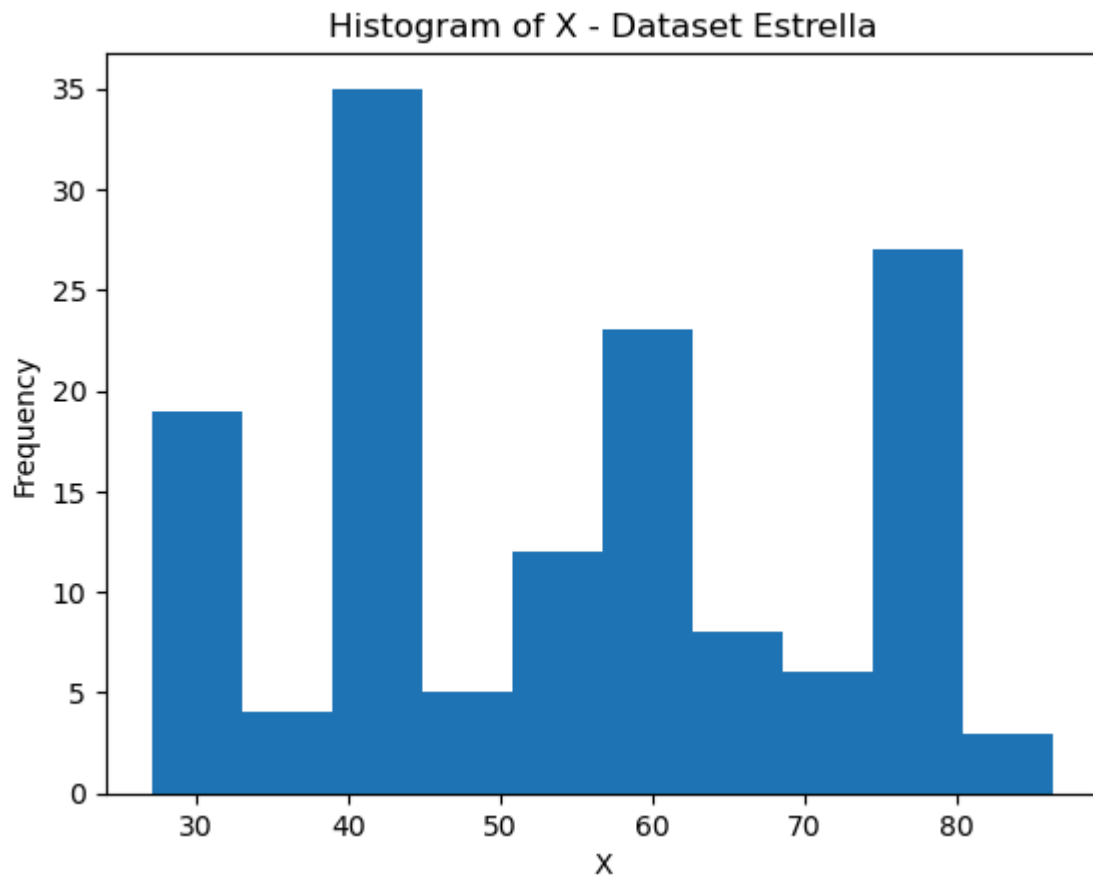
# Crear un histograma para la columna 'y' del dataset "estrella"
plt.hist(df_estrella['y'])
plt.xlabel('Y')
```

```
plt.ylabel('Frequency')
plt.title('Histogram of Y - Dataset Estrella')
plt.show()
```

	dataset	x	y
0	dino	55.3846	97.1795
1	dino	51.5385	96.0256
2	dino	46.1538	94.4872
3	dino	42.8205	91.4103
4	dino	40.7692	88.3333

	dataset	x	y
710	star	58.213608	91.881892
711	star	58.196054	92.214989
712	star	58.718231	90.310532
713	star	57.278373	89.907607
714	star	58.082020	92.008145



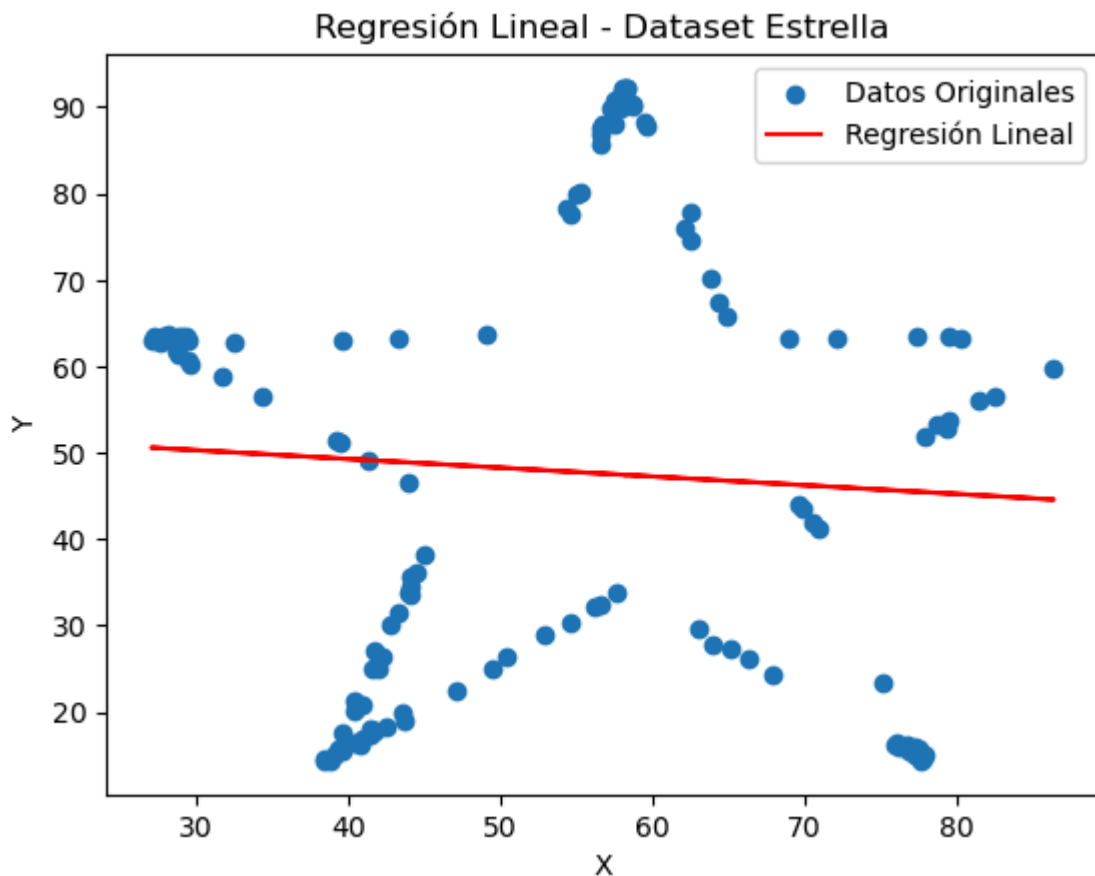


```
In [3]: print("Modelo de regresion LINEAL")
# Crear un modelo de regresión lineal
model_lineal = LinearRegression()
model_lineal.fit(df_estrella[['x']], df_estrella['y'])
```

```
# Predecir valores utilizando el modelo de regresión lineal
y_pred_lineal = model_lineal.predict(df_estrella[['x']])

# Graficar los datos originales y la línea de regresión lineal
plt.scatter(df_estrella['x'], df_estrella['y'], label='Datos Originales')
plt.plot(df_estrella['x'], y_pred_lineal, color='red', label='Regresión Lineal')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Regresión Lineal - Dataset Estrella')
plt.legend()
plt.show()
```

Modelo de regresion LINEAL



```
In [4]: print("Modelo de regresion CUADRATICA")
# Cargar el archivo CSV en un DataFrame de pandas
df = pd.read_csv('datasauRUS.csv')

# Filtrar el DataFrame para obtener solo las filas correspondientes al dataset "
df_estrella = df[df['dataset'] == 'star']

# Crear un modelo de regresión cuadrática
poly_features = PolynomialFeatures(degree=2)
X_poly = poly_features.fit_transform(df_estrella[['x']])
model_cuadratico = LinearRegression()
model_cuadratico.fit(X_poly, df_estrella['y'])

# Predecir valores utilizando el modelo de regresión cuadrática
y_pred_cuadratico = model_cuadratico.predict(X_poly)

# Ordenar los valores para graficar la curva de regresión cuadrática
sort_axis = np.argsort(df_estrella['x'])
x_sorted = df_estrella['x'].values[sort_axis]
```



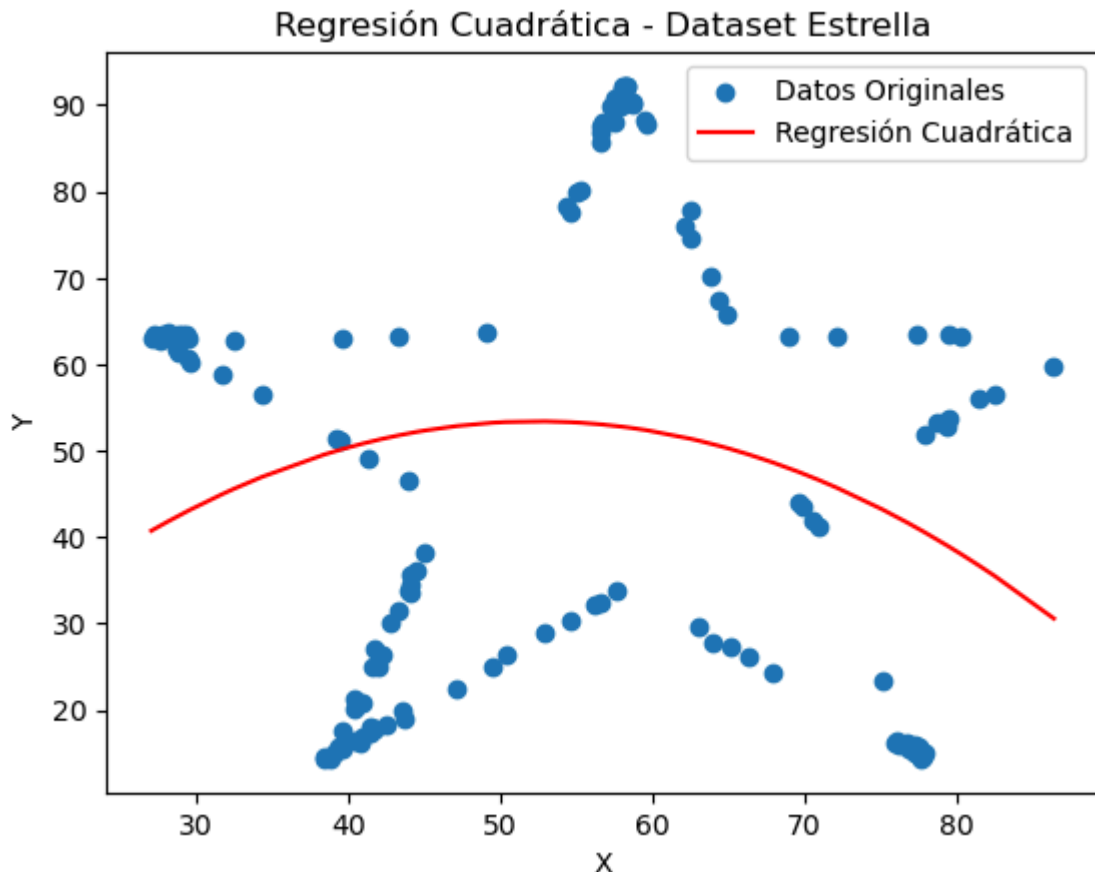
```

y_pred_sorted = y_pred_cuadratico[sort_axis]

# Graficar los datos originales y la curva de regresión cuadrática
plt.scatter(df_estrella['x'], df_estrella['y'], label='Datos Originales')
plt.plot(x_sorted, y_pred_sorted, color='red', label='Regresión Cuadrática')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Regresión Cuadrática - Dataset Estrella')
plt.legend()
plt.show()

```

Modelo de regresion CUADRATICA



```

In [5]: print("Modelo de regresion POLINOMIAL")
# Crear un modelo de regresión polinomial
poly_features = PolynomialFeatures(degree=3) # Puedes ajustar el grado del poli
X_poly = poly_features.fit_transform(df_estrella[['x']])
model_polinomial = LinearRegression()
model_polinomial.fit(X_poly, df_estrella['y'])

# Predecir valores utilizando el modelo de regresión polinomial
y_pred_polinomial = model_polinomial.predict(X_poly)

# Ordenar los valores para graficar la curva de regresión polinomial
sort_axis = np.argsort(df_estrella['x'])
x_sorted = df_estrella['x'].values[sort_axis]
y_pred_sorted = y_pred_polinomial[sort_axis]

# Graficar los datos originales y la curva de regresión polinomial
plt.scatter(df_estrella['x'], df_estrella['y'], label='Datos Originales')
plt.plot(x_sorted, y_pred_sorted, color='red', label='Regresión Polinomial')
plt.xlabel('X')
plt.ylabel('Y')

```

```
plt.title('Regresión Polinomial - Dataset Estrella')  
plt.legend()  
plt.show()
```

Modelo de regresion POLINOMIAL

