DATA VISUALIZATIONS

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
# Load your dataset (replace with your dataset)
# For example, using seaborn's built-in dataset
# Load the dataset
df_1=pd.read_csv(r"C:\Users\arumu\Downloads\cardio_train.csv",delimiter=';')
df_1
```

Out[6]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69995	99993	19240	2	168	76.0	120	80	1	1	1	0	1	0
69996	99995	22601	1	158	126.0	140	90	2	2	0	0	1	1
69997	99996	19066	2	183	105.0	180	90	3	1	0	1	0	1
69998	99998	22431	1	163	72.0	135	80	1	2	0	0	0	1
69999	99999	20540	1	170	72.0	120	80	2	1	0	0	1	0

70000 rows × 13 columns

In [7]:
```
# For example, using seaborn's built-in dataset
df_ = sns.load_dataset('iris')
df_
```

Out[7]:

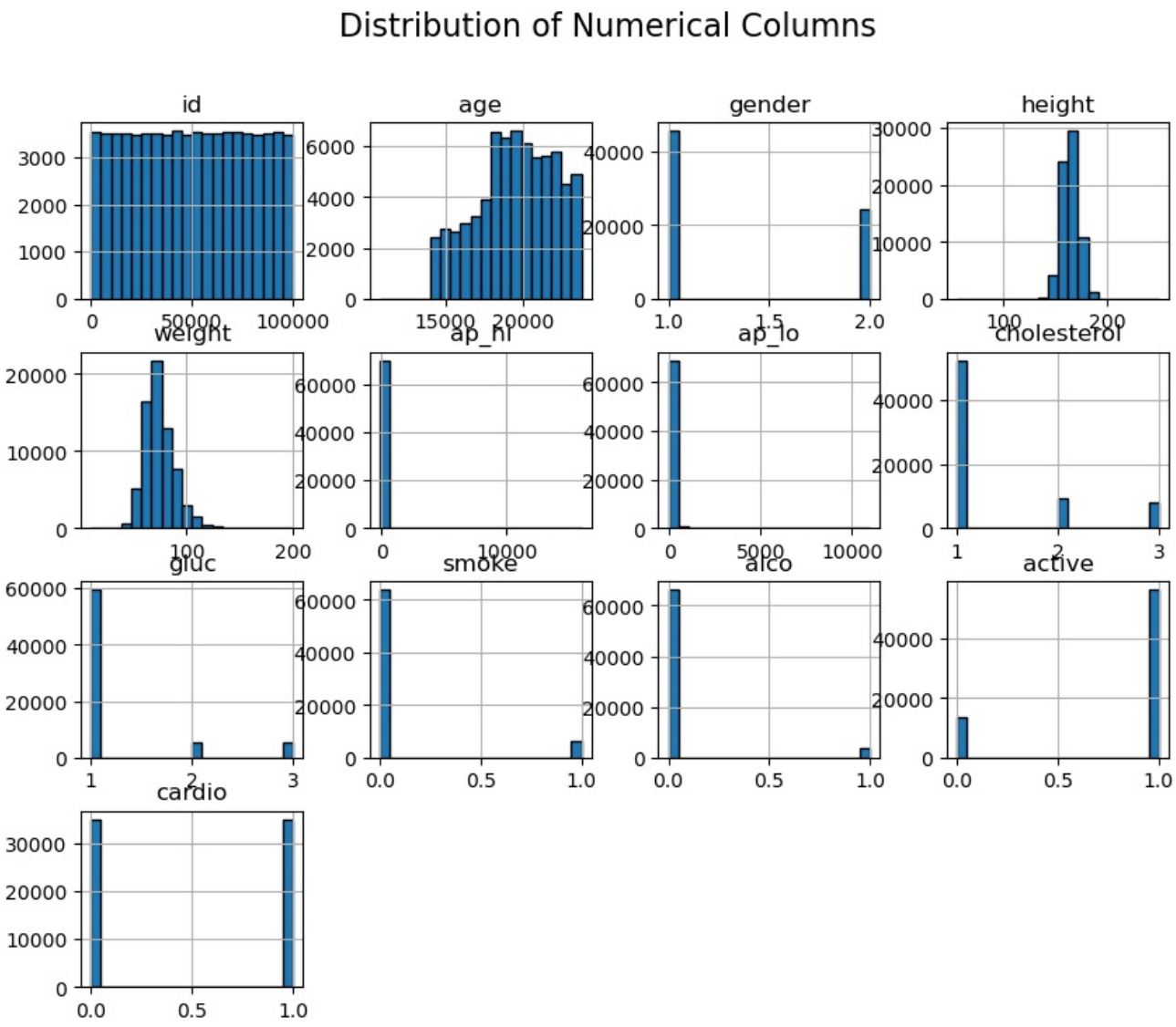
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

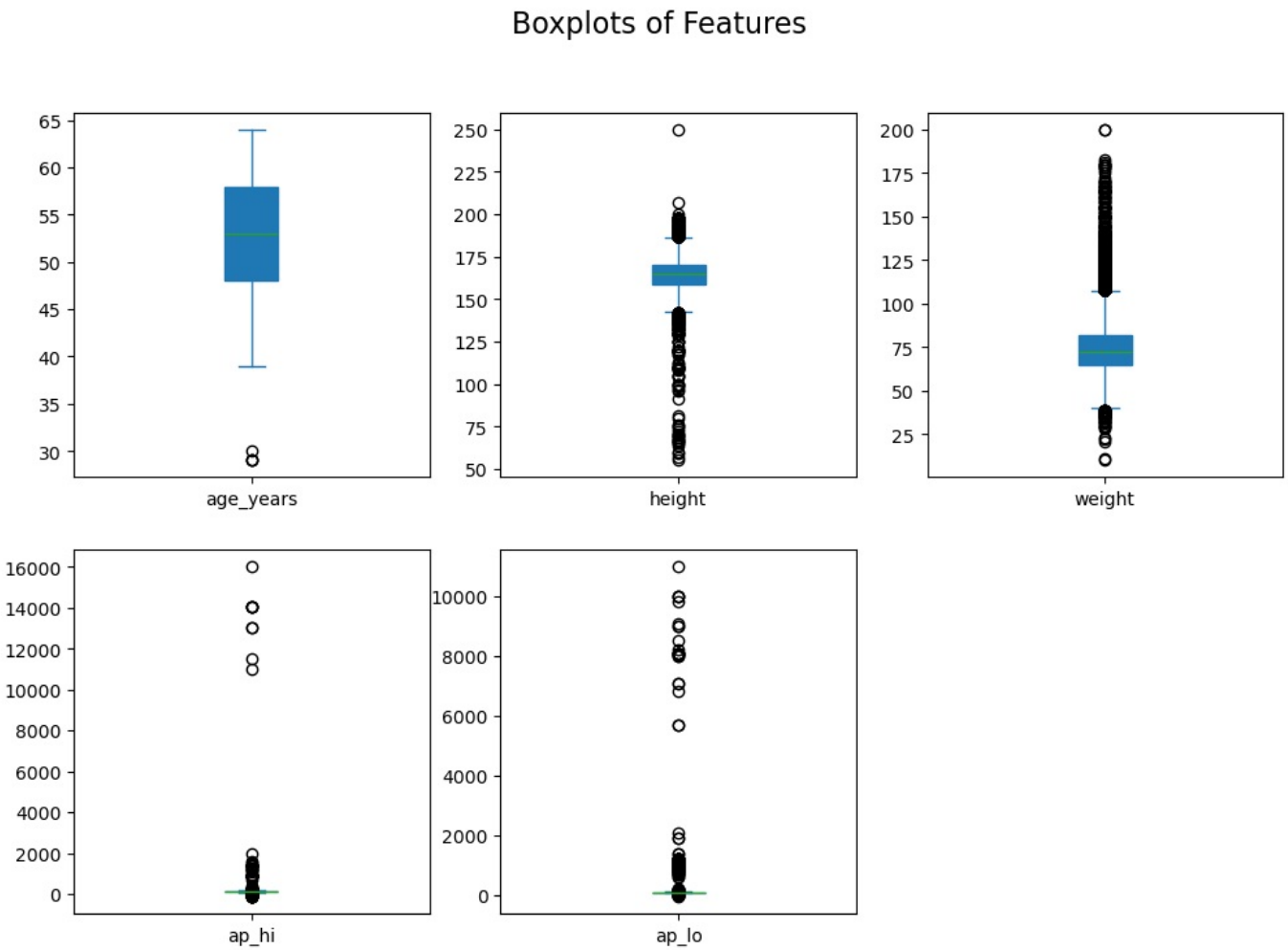
In [17]:
```
# Checking for missing values
df_1.isnull().sum()

# Replace 'X' column names if there are unnamed columns
df_1.columns = df_1.columns.str.strip()

# Convert any necessary columns to appropriate types (e.g., if age is in days)
df_1['age_years'] = df_1['age'] // 365  # Assuming 'age' is in days
df_1['age_years']
```

Out[17]:
```
0        50
1        55
2        51
3        48
4        47
         ..
69995    52
69996    61
69997    52
69998    61
69999    56
Name: age_years, Length: 70000, dtype: int64
```

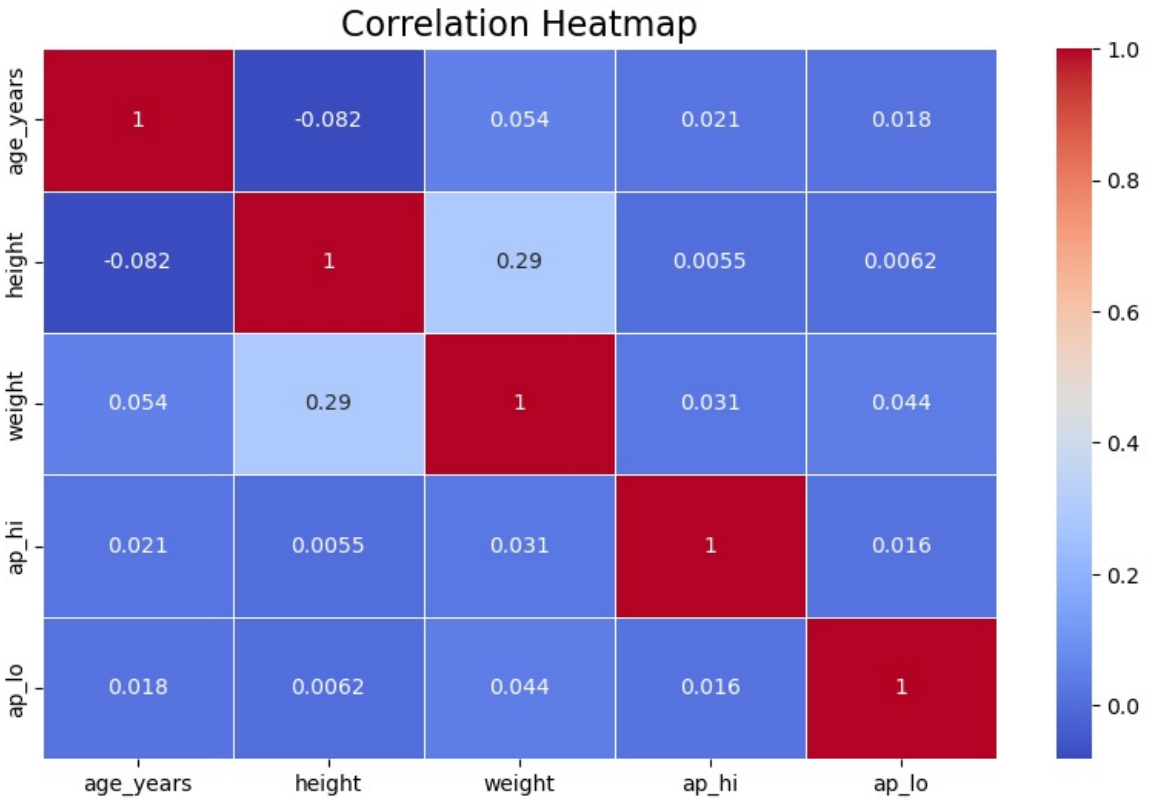
In [9]:
```
# 1. Histogram - Distribution of numerical columns
def plot_histogram(df_1):
    df_1.hist(figsize=(10, 8), bins=20, edgecolor='black')
    plt.suptitle('Distribution of Numerical Columns', fontsize=16)
    plt.show()
plot_histogram(df_1)
```



In [18]:
```
# Boxplot - To identify outliers in continuous features
def plot_boxplots(df_1):
    df_1[['age_years', 'height', 'weight', 'ap_hi', 'ap_lo']].plot(kind='box', subplots=True, layout=(2, 3), fi
    plt.suptitle('Boxplots of Features', fontsize=16)
    plt.show()

plot_boxplots(df_1)
```

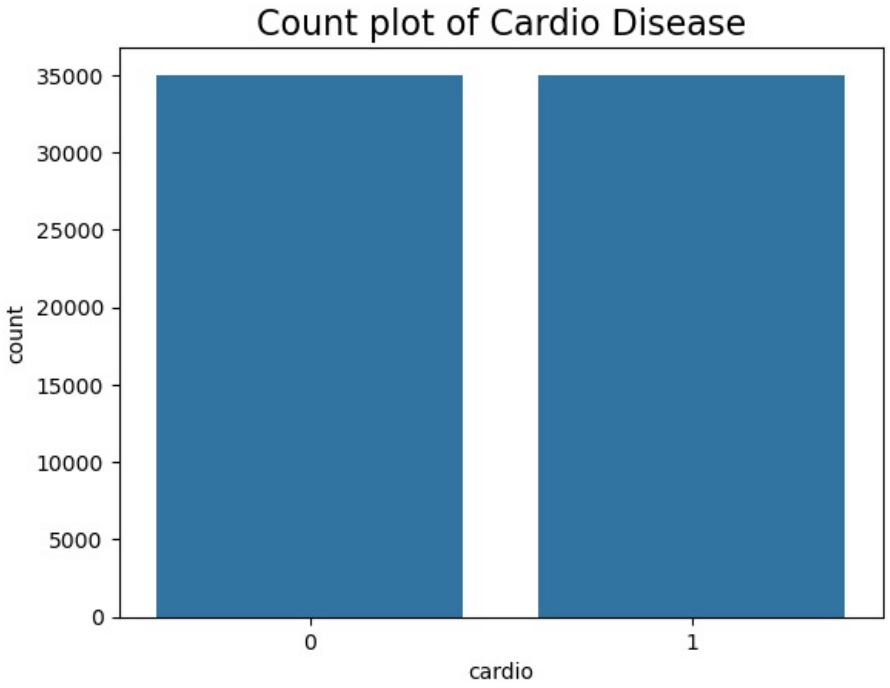


In [19]:
```
# Correlation Heatmap
def plot_correlation(df_1):
    plt.figure(figsize=(10, 6))
    correlation = df_1[['age_years', 'height', 'weight', 'ap_hi', 'ap_lo']].corr()
    sns.heatmap(correlation, annot=True, cmap='coolwarm', linewidths=0.5)
    plt.title('Correlation Heatmap', fontsize=16)
    plt.show()

plot_correlation(df_1)
```

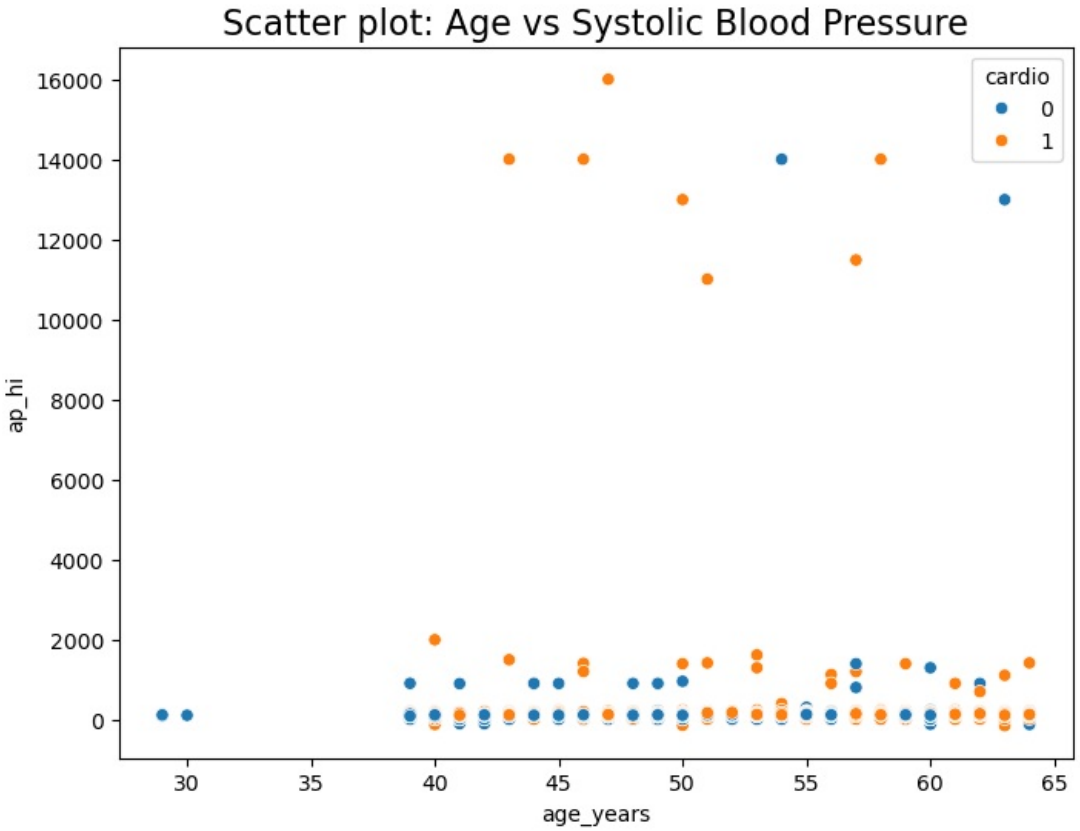


In [20]:
```
# Count plot - Distribution of the target variable 'cardio'
def plot_countplot(df_1):
```

```
    sns.countplot(x='cardio', data=df_1)
    plt.title('Count plot of Cardio Disease', fontsize=16)
    plt.show()

plot_countplot(df_1)
```



In [22]:
```
# Scatter plot - Relationship between age and systolic blood pressure (ap_hi)
def plot_scatter(df_1):
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x='age_years', y='ap_hi', hue='cardio', data=df_1)
    plt.title('Scatter plot: Age vs Systolic Blood Pressure', fontsize=16)
    plt.show()

plot_scatter(df_1)
```



In [ ]:
```

```