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

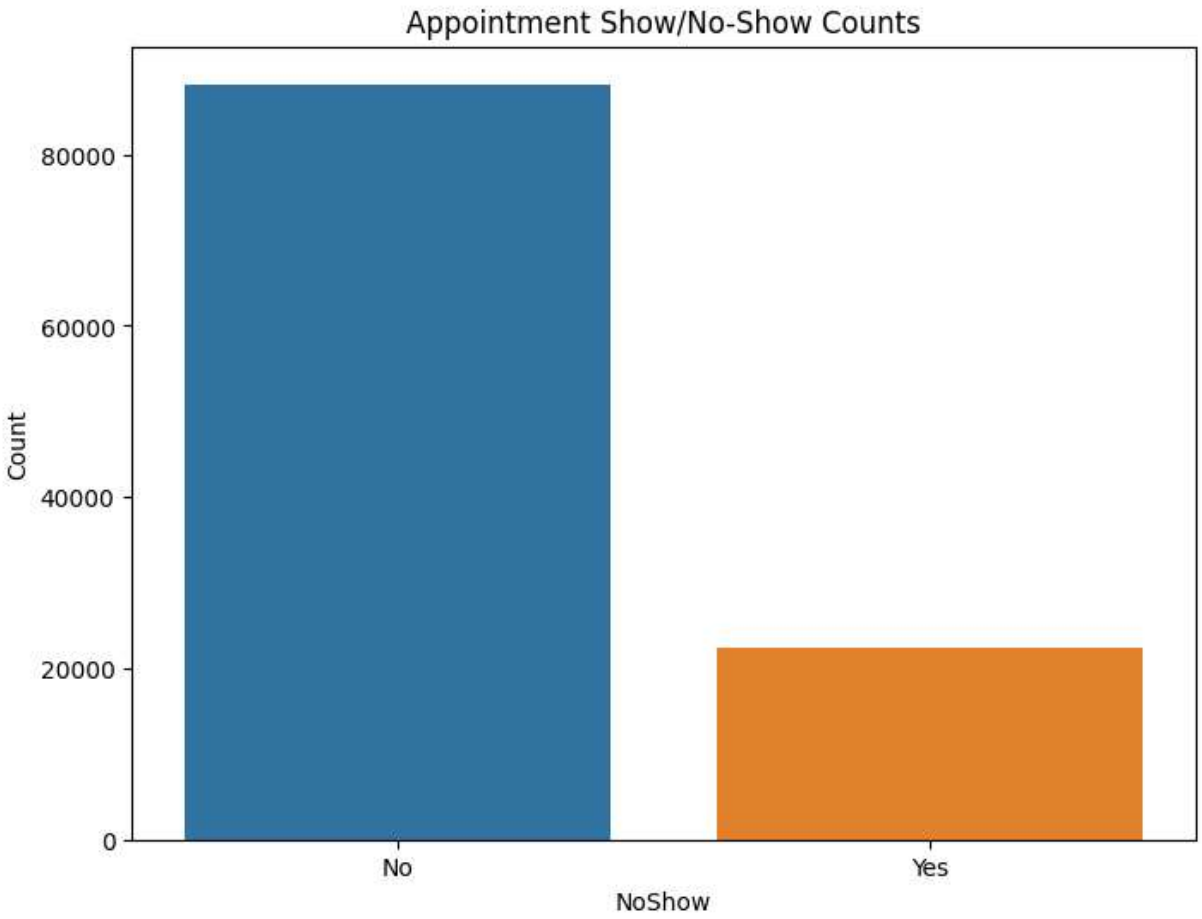
In [3]:
```
# Step 1: Import the dataset
df = pd.read_csv("Data.csv")
```

In [4]:
```
# Step 2: Modify the date and time to standard format
df['ScheduledDay'] = pd.to_datetime(df['ScheduledDay'])
df['AppointmentDay'] = pd.to_datetime(df['AppointmentDay'])
```

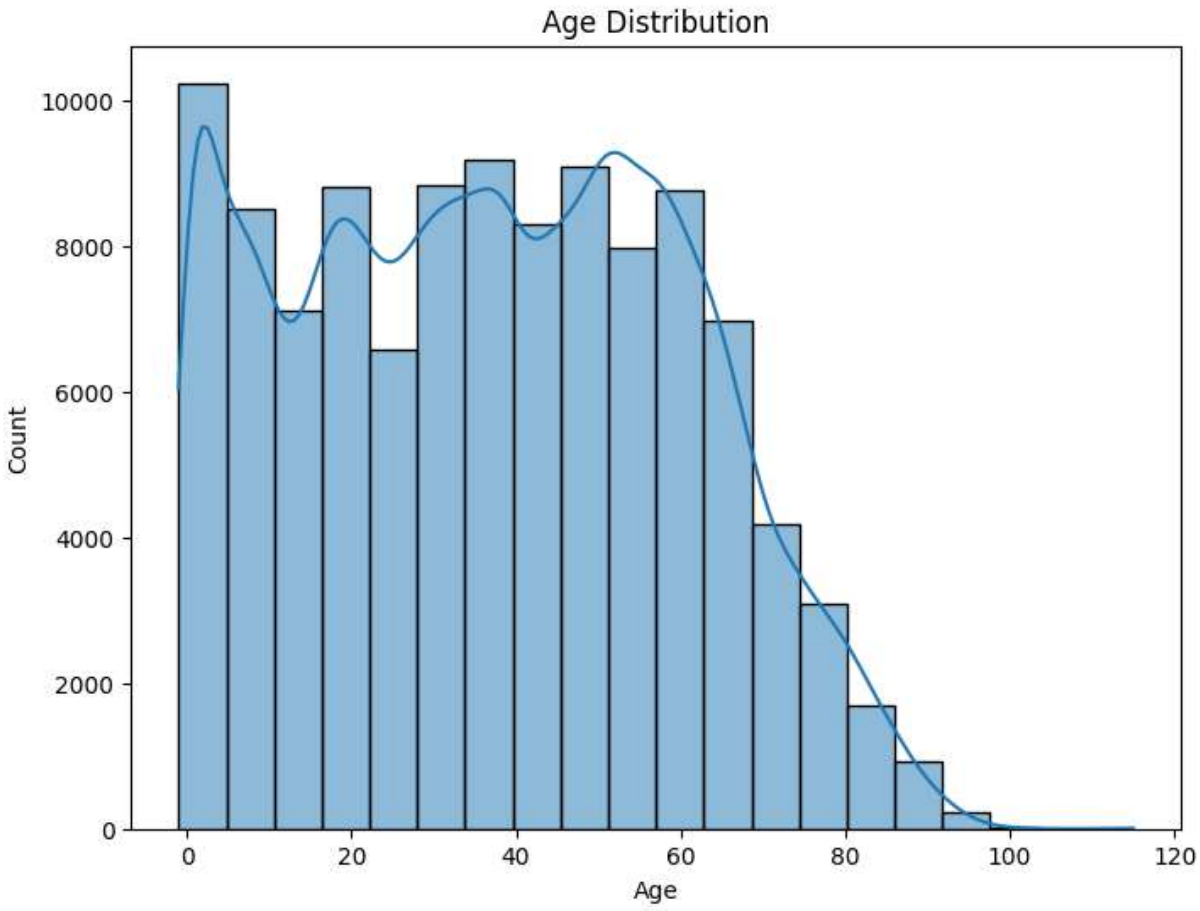
In [5]:
```
# Step 3: Store the days into sch_day and app_day variables (Monday is 0, Sunday is
df['sch_day'] = df['ScheduledDay'].dt.dayofweek
df['app_day'] = df['AppointmentDay'].dt.dayofweek
```

In [7]:
```
# Step 4: Rename columns with improper or unusual usage
# You can use the .rename() method to rename specific columns
df = df.rename(columns={'No-show': 'NoShow', 'Hipertension': 'Hypertension', 'Handc
```

In [8]:
```
# Step 5: Create a bar chart to count categories against the count
# Example: Counting 'NoShow' categories
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='NoShow')
plt.title("Appointment Show/No-Show Counts")
plt.xlabel("NoShow")
plt.ylabel("Count")
plt.show()
```

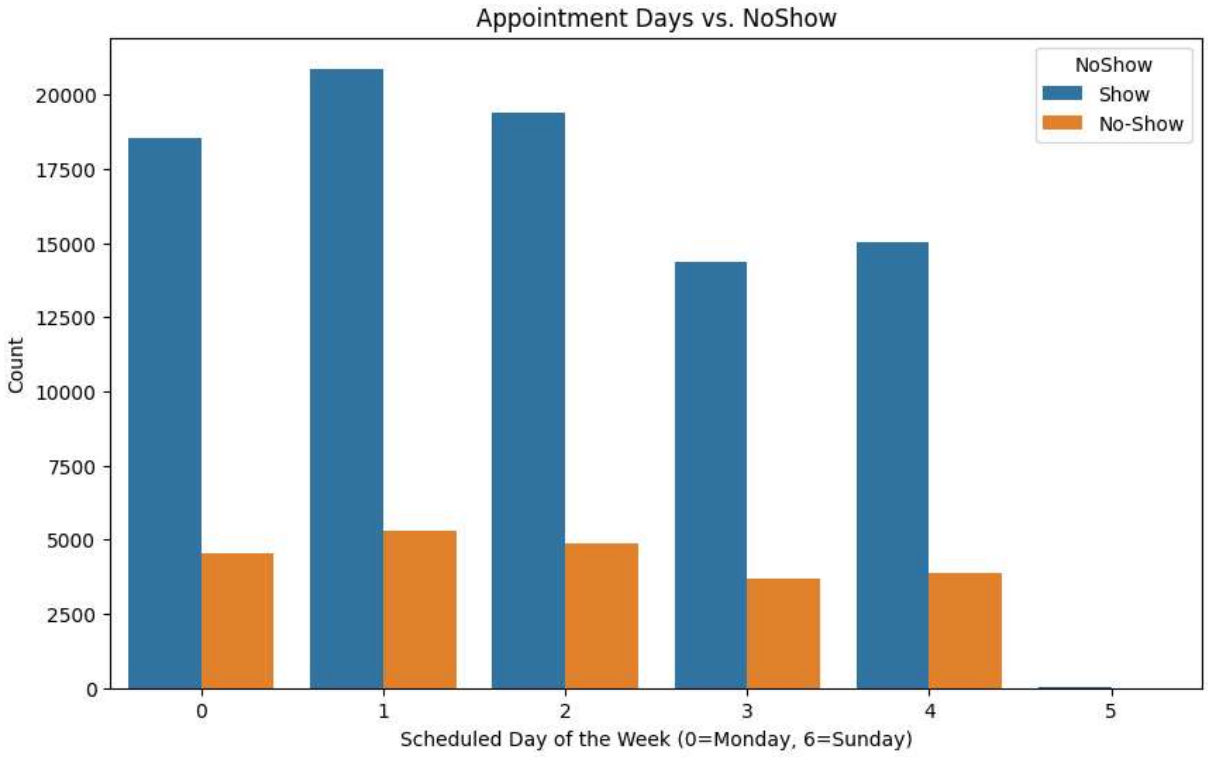


In [9]:
```
# Step 6: Plot other charts for insights about the data
# You can explore various types of plots depending on your data and research questi
# Here's an example of a histogram for Age distribution.
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```

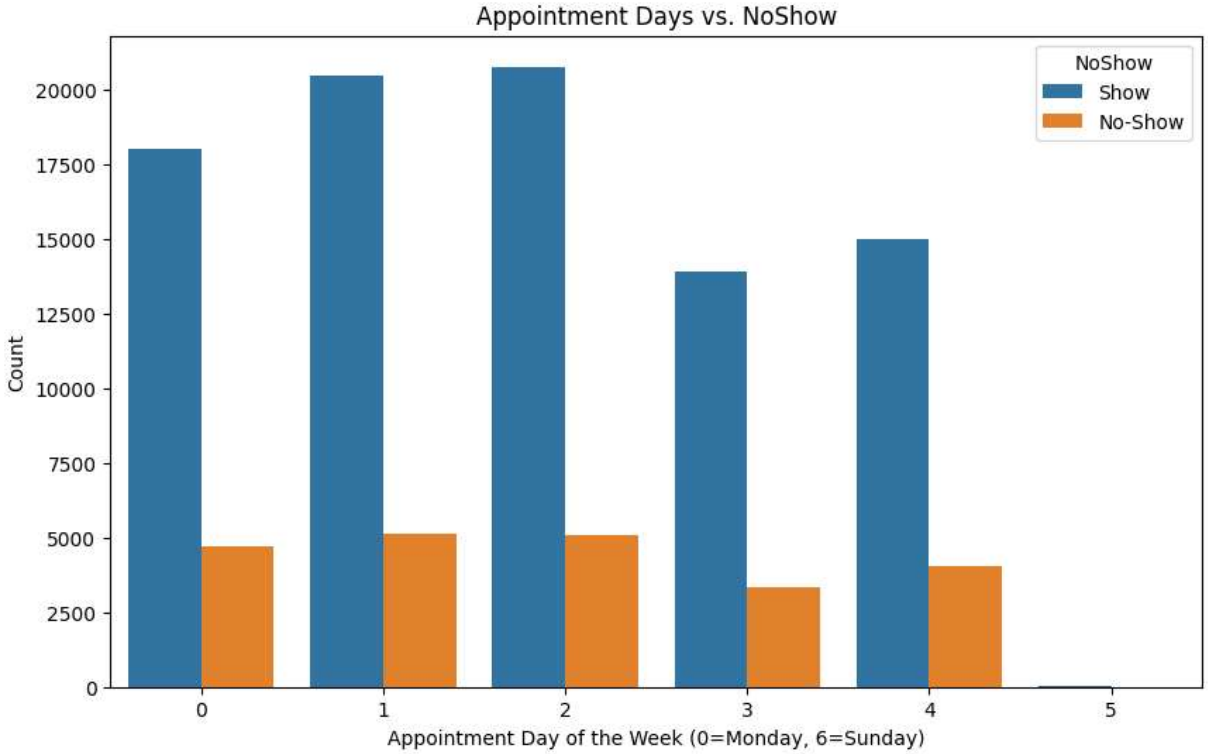


In [10]:
```
# You can create other plots based on the columns and insights you want to explore.

# For instance, to visualize appointment days (sch_day and app_day):
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='sch_day', hue='NoShow')
plt.title("Appointment Days vs. NoShow")
plt.xlabel("Scheduled Day of the Week (0=Monday, 6=Sunday)")
plt.ylabel("Count")
plt.legend(title="NoShow", loc="upper right", labels=["Show", "No-Show"])
plt.show()
```



In [11]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='app_day', hue='NoShow')
plt.title("Appointment Days vs. NoShow")
plt.xlabel("Appointment Day of the Week (0=Monday, 6=Sunday)")
plt.ylabel("Count")
plt.legend(title="NoShow", loc="upper right", labels=["Show", "No-Show"])
plt.show()
```



In [ ]: