```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import chi2_contingency, ttest_ind
```

LOADING DATA

```
df = pd.read_csv("synthetic_asthma_dataset.csv")
```

```
print(df.shape)
print(df.columns)
print(df.info())
print(df.describe(include='all'))
```

```
(10000, 17)
Index(['Patient_ID', 'Age', 'Gender', 'BMI', 'Smoking_Status',
       'Family_History', 'Allergies', 'Air_Pollution_Level',
       'Physical_Activity_Level', 'Occupation_Type', 'Comorbidities',
       'Medication_Adherence', 'Number_of_ER_Visits', 'Peak_Expiratory_Flow',
       'FeNO_Level', 'Has_Asthma', 'Asthma_Control_Level'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Patient_ID               10000 non-null  object
 1   Age                      10000 non-null  int64
 2   Gender                   10000 non-null  object
 3   BMI                      10000 non-null  float64
 4   Smoking_Status           10000 non-null  object
 5   Family_History           10000 non-null  int64
 6   Allergies                7064 non-null   object
 7   Air_Pollution_Level      10000 non-null  object
 8   Physical_Activity_Level  10000 non-null  object
 9   Occupation_Type          10000 non-null  object
 10  Comorbidities            5033 non-null   object
 11  Medication_Adherence     10000 non-null  float64
 12  Number_of_ER_Visits      10000 non-null  int64
 13  Peak_Expiratory_Flow     10000 non-null  float64
 14  FeNO_Level               10000 non-null  float64
 15  Has_Asthma               10000 non-null  int64
 16  Asthma_Control_Level     2433 non-null   object
dtypes: float64(4), int64(4), object(9)
memory usage: 1.3+ MB
None
        Patient_ID           Age  Gender           BMI Smoking_Status  \
count        10000  10000.000000   10000  10000.000000          10000
unique       10000           NaN       3           NaN              3
top     ASTH109983           NaN  Female           NaN          Never
freq             1           NaN    4814           NaN           6070
mean           NaN     44.930700     NaN     25.053320            NaN
std            NaN     25.653559     NaN      4.874466            NaN
min            NaN      1.000000     NaN     15.000000            NaN
25%            NaN     23.000000     NaN     21.600000            NaN
50%            NaN     45.000000     NaN     25.000000            NaN
75%            NaN     67.000000     NaN     28.400000            NaN
max            NaN     89.000000     NaN     45.000000            NaN

        Family_History Allergies Air_Pollution_Level Physical_Activity_Level  \
count     10000.000000      7064               10000                    10000
unique             NaN         4                   3                        3
top                NaN      Dust            Moderate               Sedentary
freq               NaN      2479                4915                     4062
mean          0.303400       NaN                 NaN                      NaN
std           0.459749       NaN                 NaN                      NaN
min           0.000000       NaN                 NaN                      NaN
25%           0.000000       NaN                 NaN                      NaN
50%           0.000000       NaN                 NaN                      NaN
75%           1.000000       NaN                 NaN                      NaN
max           1.000000       NaN                 NaN                      NaN
```

```
df.head()
```

oking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupatio
Former	1	NaN	Moderate	Sedentary	(
Former	0	Dust	Low	Moderate	
Never	0	Dust	Moderate	Moderate	
Never	0	Multiple	High	Sedentary	(
Never	0	Multiple	Moderate	Active	

Next steps: Generate code with df | View recommended plots | New interactive sheet

** DATA CLEANING**

```
print(df.isnull().sum())
```

```
Patient_ID                   0
Age                          0
Gender                       0
BMI                          0
Smoking_Status               0
Family_History               0
Allergies                 2936
Air_Pollution_Level          0
Physical_Activity_Level      0
Occupation_Type              0
Comorbidities             4967
Medication_Adherence         0
Number_of_ER_Visits          0
Peak_Expiratory_Flow         0
FeNO_Level                   0
Has_Asthma                   0
Asthma_Control_Level      7567
dtype: int64
```

```
df.fillna(method='ffill', inplace=True)
```

```
/tmp/ipython-input-3970806690.py:1: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version.
  df.fillna(method='ffill', inplace=True)
```

```
df.drop_duplicates(inplace=True)
```

```
df.head()
```

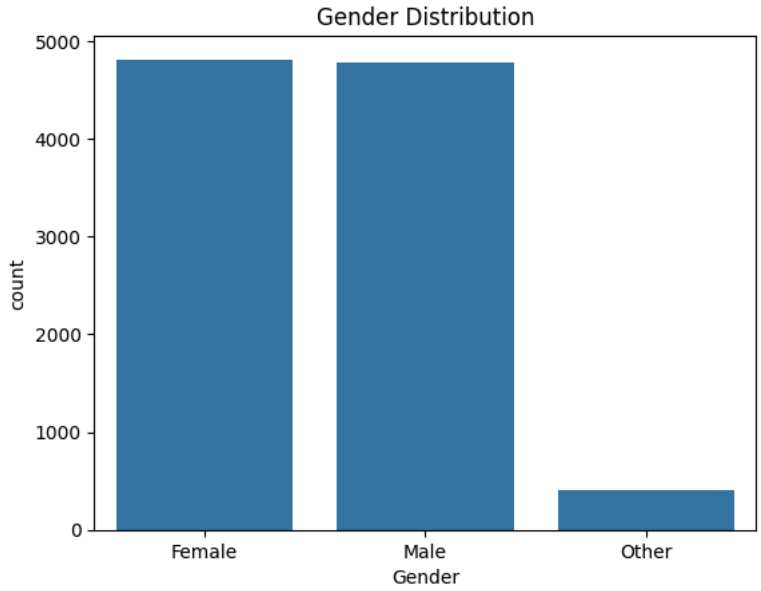
oking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupatio
Former	1	NaN	Moderate	Sedentary	(
Former	0	Dust	Low	Moderate	
Never	0	Dust	Moderate	Moderate	
Never	0	Multiple	High	Sedentary	(
Never	0	Multiple	Moderate	Active	

Next steps: Generate code with df | View recommended plots | New interactive sheet

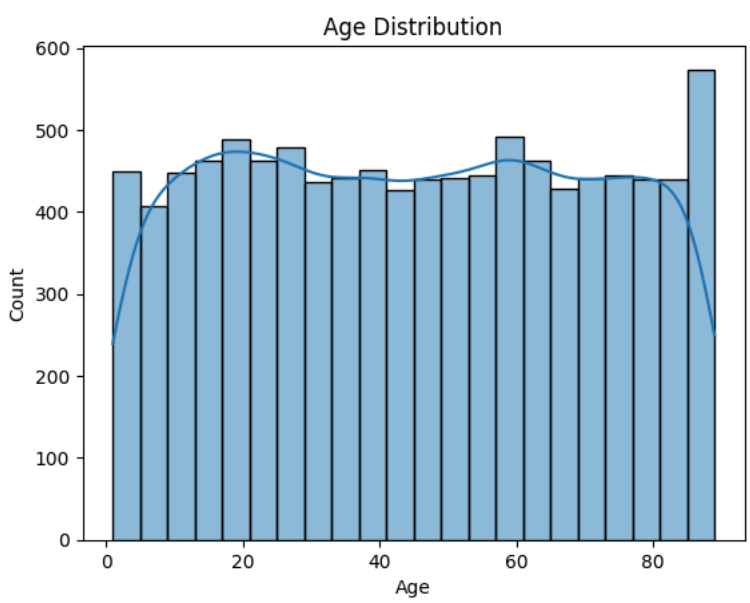
EXPLORATORY DATA ANALYSIS (EDA)

** UNIVARIATE ANALYSIS**

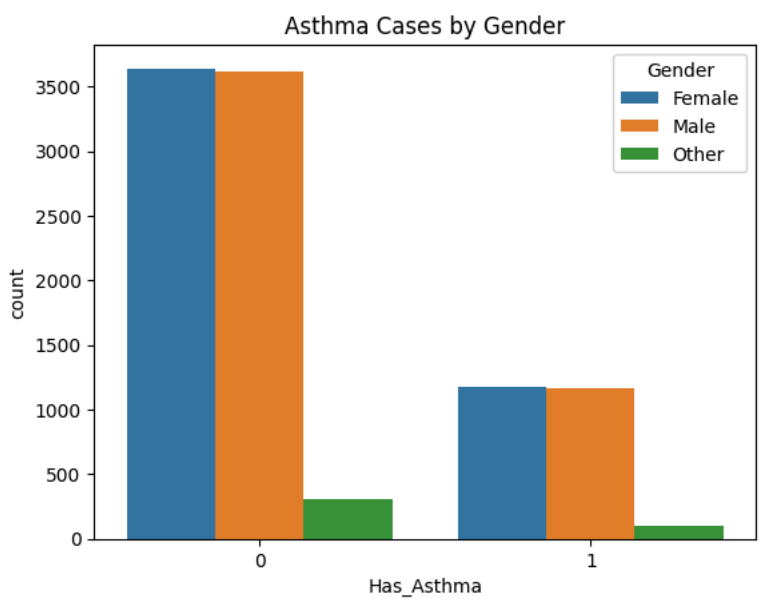
```
# 1. Distribution of gender
sns.countplot(data=df, x='Gender')
plt.title("Gender Distribution")
plt.show()
```



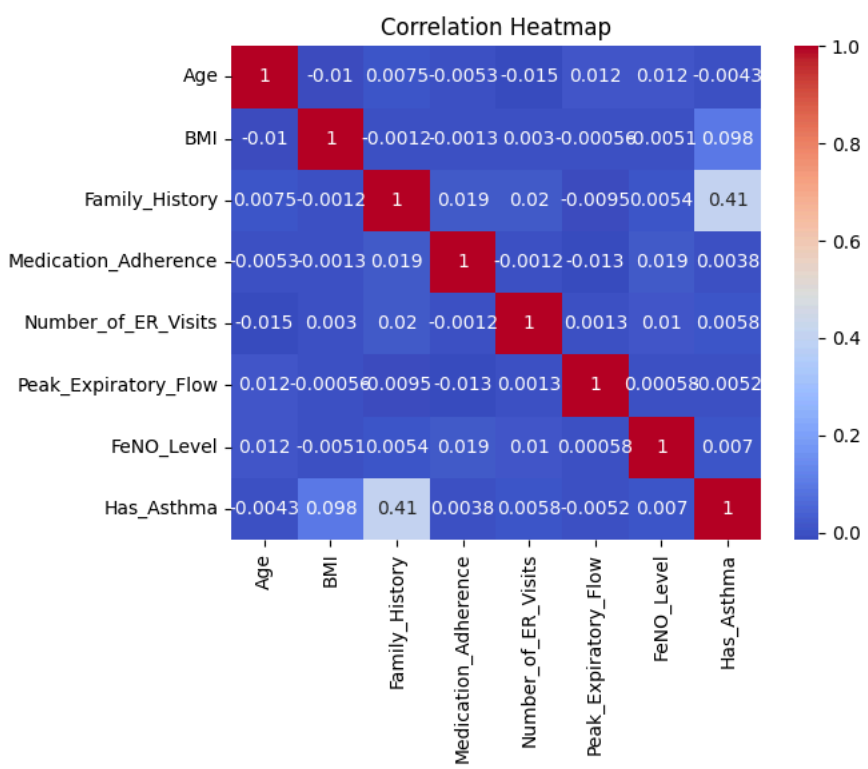

```
# Age distribution
sns.histplot(data=df, x='Age', kde=True)
plt.title("Age Distribution")
plt.show()
```



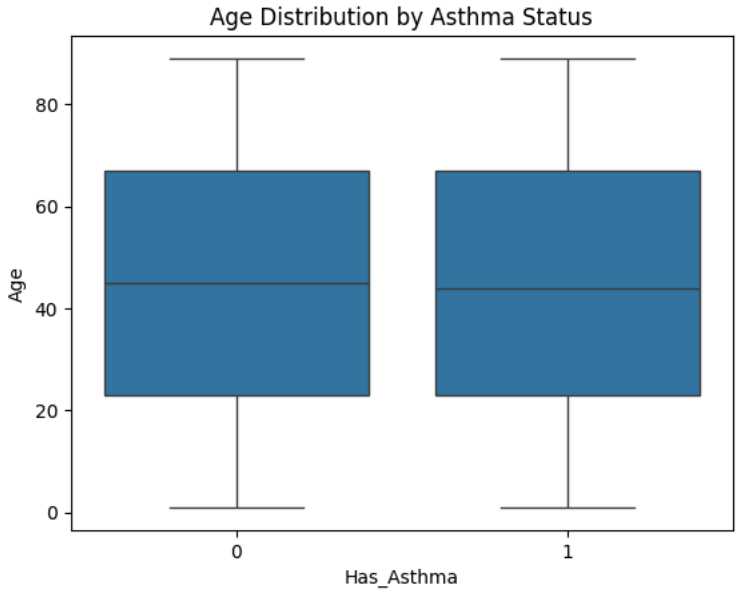

```
#Asthma prevalence by gender
sns.countplot(data=df, x='Has_Asthma', hue='Gender')
plt.title("Asthma Cases by Gender")
plt.show()
```



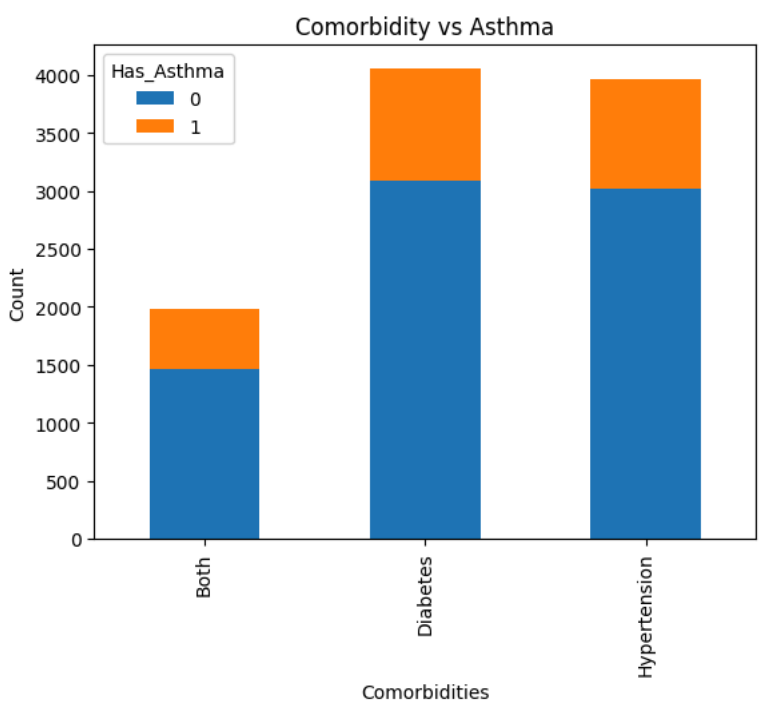

```
#Correlation matrix (for numeric variables)
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```




```
#Age vs Has Asthma
sns.boxplot(x='Has_Asthma', y='Age', data=df)
plt.title("Age Distribution by Asthma Status")
plt.show()
```




```
#Comorbidity vs Has Asthma
pd.crosstab(df['Comorbidities'], df['Has_Asthma']).plot(kind='bar', stacked=True)
plt.title("Comorbidity vs Asthma")
plt.ylabel("Count")
plt.show()
```



STATISTICAL TESTING

Chi-square test: Gender vs Asthma

```
from scipy.stats import chi2_contingency

contingency = pd.crosstab(df['Gender'], df['Has_Asthma'])
chi2, p, dof, expected = chi2_contingency(contingency)

print("Chi-square Test")
print("Chi2 Stat:", chi2)
print("p-value:", p)
if p < 0.05:
    print("Reject H0: Gender and asthma are associated.")
else:
    print("Fail to reject H0: No significant association.")
```

```
Chi-square Test
Chi2 Stat: 0.15592975133514347
```

```
p-value: 0.9249969158721212
Fail to reject H0: No significant association.
```

T-test: Age of asthma vs non-asthma Hypothesis:

```
asthma_age = df[df['Has_Asthma'] == 1]['Age']
non_asthma_age = df[df['Has_Asthma'] == 0]['Age']

t_stat, p_val = ttest_ind(asthma_age, non_asthma_age, equal_var=False)

print("T-test for Age")
print("T-statistic:", t_stat)
print("p-value:", p_val)
if p_val < 0.05:
    print("Reject H0: Age differs between asthma and non-asthma patients.")
else:
    print("Fail to reject H0: No age difference.")
```

```
T-test for Age
T-statistic: -0.43514227581166276
p-value: 0.6634819556405647
Fail to reject H0: No age difference.
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.