Exploring Asthma Risk Factors: A Data Analysis and visualization Approach using python

Submitted By:
Kadeejathul Kubra.CA

Submitted To:
Shilpa Miss

Date of submission: 21/10/25

CONTENTS

INTRODUCTION

This project focuses on analyzing a Synthetic Asthma Dataset using Python to explore the various factors that influence asthma occurrence, control, and severity

❖ Python libraries used

To perform data cleaning, analysis, and visualization, several Python libraries were utilized:

pandas – for data loading, cleaning, and manipulation

numpy – for numerical computations and handling missing values

matplotlib – for basic charts and visualizations

seaborn – for advanced and aesthetically appealing statistical visualizations

❖ Aim

The main aim of this project is to analyze and visualize asthma-related data to identify key factors affecting asthma prevalence, patient outcomes, and control levels.

❖ Objectives

1. To clean and preprocess the dataset by handling missing or inconsistent values.
2. To explore demographic variables such as age, gender, and BMI among asthma patients.
3. To identify environmental and behavioral risk factors such as smoking and air pollution.
4. To visualize the relationship between medication adherence and asthma control.
5. To interpret clinical indicators like FeNO levels and peak expiratory flow for asthma diagnosis.

DATASET DESCRIPTION

```
import pandas as pd
df=pd.read_csv("synthetic_asthma_dataset.csv")
df
```

	Patient_ID	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence
0	ASTH100000	52	Female	27.6	Former	1	NaN	Moderate	Sedentary	Outdoor	Diabetes	0.38
1	ASTH100001	15	Male	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60
2	ASTH100002	72	Female	17.6	Never	0	NaN	Moderate	Moderate	Indoor	NaN	0.38
3	ASTH100003	61	Male	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60
4	ASTH100004	21	Male	30.2	Never	0	NaN	Moderate	Active	Indoor	NaN	0.82
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	ASTH109995	70	Male	25.0	Never	0	NaN	Low	Sedentary	Indoor	NaN	0.67
9996	ASTH109996	78	Female	24.8	Never	0	Pollen	Low	Moderate	Indoor	Diabetes	0.72
9997	ASTH109997	58	Male	30.1	Former	1	Pollen	Low	Moderate	Indoor	NaN	0.28
9998	ASTH109998	88	Female	31.2	Former	0	Pollen	Moderate	Moderate	Indoor	NaN	0.44
9999	ASTH109999	13	Female	16.4	Former	1	Multiple	High	Moderate	Outdoor	NaN	0.23

10000 rows × 17 columns

The Synthetic Asthma Dataset is designed to analyze the various factors that influence asthma occurrence, severity, and management among individuals. It contains detailed information about patients, including age, gender, smoking status, BMI, family history, air pollution exposure, FeNO levels, and number of ER visits, medication adherence, and asthma control levels. These attributes help in understanding the relationships between lifestyle, environmental, and biological factors affecting asthma.

This dataset provides a realistic foundation for conducting data cleaning, visualization, and statistical analysis to explore patterns such as how air pollution or smoking affects asthma, how medication adherence improves control, and how genetic history contributes to disease risk. The insights gained from this dataset can support public health decisions, preventive strategies, and better asthma management in both clinical and community settings.

COLUMN NAME	DATA TYPE	DESCRIPTION
Patient Id	Object	Unique identifier assigned to each patient
Age	Int	Age of the individuals in years
Gender	Object	Gender of the patient
Bmi	Float	Bmi calculated from height and weight
Smoking Status	Object	Smoking behavior
Family History	Int	Indicates the family history
Allergies	Object	Types of allergies to patient
Comorbidities	Object	Presence of other health issues
Air Pollution Level	Object	Air quality level in patients area
Feno Level	Float	Fractional exhaled nitric oxide
Peak Expiratory Flow	Float	Measure how fast person can exhale
Medication Adherence	Float	Pearsons consistency in following prescribed medication
Er Visits	Int	Emergency room visit
Asthma Control Level	Object	Level of control
Occupation type	Object	Occupation of patient
Physical activity level	Object	Level of activities
Has Asthma	Int	Indicates whether the patient has been diagnosed with asthma

No of rows:10000

No of columns:17

Data set source: kaggle

DATA CLEANING

CHECK IF THERE ANY NULL VALUES:

```
df.isnull().any()
```

	0
Patient_ID	False
Age	False
Gender	False
BMI	False
Smoking_Status	False
Family_History	False
Allergies	True
Air_Pollution_Level	False
Physical_Activity_Level	False
Occupation_Type	False
Comorbidities	True
Medication_Adherence	False
Number_of_ER_Visits	False
Peak_Expiratory_Flow	False
FeNO_Level	False
Has_Asthma	False
Asthma_Control_Level	True

FIRST FIVE ROWS:

```
df.head()
```

	Patient_ID	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	N
0	ASTH100000	52	Female	27.6	Former	1	NaN	Moderate	Sedentary	Outdoor	Diabetes	0.38	
1	ASTH100001	15	Male	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60	
2	ASTH100002	72	Female	17.6	Never	0	NaN	Moderate	Moderate	Indoor	NaN	0.38	
3	ASTH100003	61	Male	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60	
4	ASTH100004	21	Male	30.2	Never	0	NaN	Moderate	Active	Indoor	NaN	0.82	

LAST FIVE ROWS:

```
df.tail()
```

	Patient_ID	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication
9995	ASTH109995	70	Male	25.0	Never	0	NaN	Low	Sedentary	Indoor	NaN	
9996	ASTH109996	78	Female	24.8	Never	0	Pollen	Low	Moderate	Indoor	Diabetes	
9997	ASTH109997	58	Male	30.1	Former	1	Pollen	Low	Moderate	Indoor	NaN	
9998	ASTH109998	88	Female	31.2	Former	0	Pollen	Moderate	Moderate	Indoor	NaN	
9999	ASTH109999	13	Female	16.4	Former	1	Multiple	High	Moderate	Outdoor	NaN	

INFO:

```
df.info()
```

```
<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
```

DESCRIBE:

```
df.describe()
```

	Age	BMI	Family_History	Medication_Adherence	Number_of_ER_Visits	Peak_Expiratory_Flow	FeNO_Level	Has_Asthma
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	44.930700	25.053320	0.303400	0.497998	1.015900	400.884090	25.101420	0.243300
std	25.653559	4.874466	0.459749	0.224809	1.020564	97.531113	9.840184	0.429096
min	1.000000	15.000000	0.000000	0.000000	0.000000	150.000000	5.000000	0.000000
25%	23.000000	21.600000	0.000000	0.320000	0.000000	334.800000	18.200000	0.000000
50%	45.000000	25.000000	0.000000	0.500000	1.000000	402.500000	25.000000	0.000000
75%	67.000000	28.400000	1.000000	0.670000	2.000000	468.700000	31.700000	0.000000
max	89.000000	45.000000	1.000000	0.990000	6.000000	600.000000	63.900000	1.000000

DROP THE EMPLOYEE ID:

```
df.drop(columns=["Patient_ID"],inplace=True)
df
```

	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	Number_of_
0	52	Female	27.6	Former	1	NaN	Moderate	Sedentary	Outdoor	Diabetes	0.38	
1	15	Male	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60	
2	72	Female	17.6	Never	0	NaN	Moderate	Moderate	Indoor	NaN	0.38	
3	61	Male	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60	
4	21	Male	30.2	Never	0	NaN	Moderate	Active	Indoor	NaN	0.82	
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	70	Male	25.0	Never	0	NaN	Low	Sedentary	Indoor	NaN	0.67	
9996	78	Female	24.8	Never	0	Pollen	Low	Moderate	Indoor	Diabetes	0.72	
9997	58	Male	30.1	Former	1	Pollen	Low	Moderate	Indoor	NaN	0.28	
9998	88	Female	31.2	Former	0	Pollen	Moderate	Moderate	Indoor	NaN	0.44	
9999	13	Female	16.4	Former	1	Multiple	High	Moderate	Outdoor	NaN	0.23	

10000 rows × 16 columns

REPLACE THE GENDERS:

```
df["Gender"].replace({"Male":"M"},inplace=True)
df
```

```
/tmp/ipython-input-3523758271.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operat
```

df["Gender"].replace({"Male":"M"},inplace=True)

	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	Number_of_Ef
0	52	F	27.6	Former	1	NaN	Moderate	Sedentary	Outdoor	Diabetes	0.38	
1	15	M	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60	
2	72	F	17.6	Never	0	NaN	Moderate	Moderate	Indoor	NaN	0.38	
3	61	M	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60	
4	21	M	30.2	Never	0	NaN	Moderate	Active	Indoor	NaN	0.82	
...	...	...	...	...	...	...	...	...	...	...	...	
9995	70	M	25.0	Never	0	NaN	Low	Sedentary	Indoor	NaN	0.67	
9996	78	F	24.8	Never	0	Pollen	Low	Moderate	Indoor	Diabetes	0.72	
9997	58	M	30.1	Former	1	Pollen	Low	Moderate	Indoor	NaN	0.28	
9998	88	F	31.2	Former	0	Pollen	Moderate	Moderate	Indoor	NaN	0.44	
9999	13	F	16.4	Former	1	Multiple	High	Moderate	Outdoor	NaN	0.23	

```
df["Gender"].replace({"Female":"F"},inplace=True)
df
```

```
/tmp/ipython-input-1431184987.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame o
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate obje

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace
```

df["Gender"].replace({"Female":"F"},inplace=True)

	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity
0	52	F	27.6	Former	1	NaN	Moderate	Se
1	15	Male	24.6	Former	0	Dust	Low	M
2	72	F	17.6	Never	0	NaN	Moderate	M
3	61	Male	16.8	Never	0	Multiple	High	Se
4	21	Male	30.2	Never	0	NaN	Moderate	
...	...	...	...	...	...	...	...	
9995	70	Male	25.0	Never	0	NaN	Low	Se
9996	78	F	24.8	Never	0	Pollen	Low	M
9997	58	Male	30.1	Former	1	Pollen	Low	M
9998	88	F	31.2	Former	0	Pollen	Moderate	M
9999	13	F	16.4	Former	1	Multiple	High	M

FILL WITH UNKNOWN:

```
df["Comorbidities"].fillna("unknown",inplace=True)
df
```

```
/tmp/ipython-input-1300163578.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operati
```

```
df["Comorbidities"].fillna("unknown",inplace=True)
```

	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	Number_of_ER
0	52	F	27.6	Former	1	NaN	Moderate	Sedentary	Outdoor	Diabetes	0.38	
1	15	M	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60	
2	72	F	17.6	Never	0	NaN	Moderate	Moderate	Indoor	unknown	0.38	
3	61	M	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60	
4	21	M	30.2	Never	0	NaN	Moderate	Active	Indoor	unknown	0.82	
...	...	...	...	...	...	...	...	...	...	...	...	
9995	70	M	25.0	Never	0	NaN	Low	Sedentary	Indoor	unknown	0.67	

```
df["Asthma_Control_Level"].fillna("unknown",inplace=True)
df
```

```
on a copy of a DataFrame or Series through chained assignment using an inplace method.
cause the intermediate object on which we are setting values always behaves as a copy.

ethod({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.
```

on_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	Number_of_ER_Visits	Peak_Expiratory_Flow	FeNO_Level	Has_Asthma	Asthma_Control_Level
Moderate	Sedentary	Outdoor	Diabetes	0.38	0	421.0	46.0	0	unknown
Low	Moderate	Indoor	Both	0.60	2	297.6	22.9	0	unknown
Moderate	Moderate	Indoor	unknown	0.38	0	303.3	15.3	0	unknown
High	Sedentary	Outdoor	Both	0.60	1	438.0	40.1	1	Poorly Controlled
Moderate	Active	Indoor	unknown	0.82	3	535.0	27.7	0	unknown
...	...	...	...	...	...	...	...	...	...
Low	Sedentary	Indoor	unknown	0.67	0	580.6	18.7	0	unknown
Low	Moderate	Indoor	Diabetes	0.72	1	417.6	40.8	0	unknown
Low	Moderate	Indoor	unknown	0.28	0	459.1	20.3	1	Not Controlled
Moderate	Moderate	Indoor	unknown	0.44	0	415.9	25.0	0	unknown

```
df["Allergies"].fillna("unknown",inplace=True)
df
```

```
/tmp/ipython-input-2771968839.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operati
```

```
df["Allergies"].fillna("unknown",inplace=True)
```

	Age	Gender	BMI	Smoking_Status	Family_History	Allergies	Air_Pollution_Level	Physical_Activity_Level	Occupation_Type	Comorbidities	Medication_Adherence	Number_of_ER
0	52	F	27.6	Former	1	unknown	Moderate	Sedentary	Outdoor	Diabetes	0.38	
1	15	M	24.6	Former	0	Dust	Low	Moderate	Indoor	Both	0.60	
2	72	F	17.6	Never	0	unknown	Moderate	Moderate	Indoor	unknown	0.38	
3	61	M	16.8	Never	0	Multiple	High	Sedentary	Outdoor	Both	0.60	
4	21	M	30.2	Never	0	unknown	Moderate	Active	Indoor	unknown	0.82	
...	...	...	...	...	...	...	...	...	...	...	...	
9995	70	M	25.0	Never	0	unknown	Low	Sedentary	Indoor	unknown	0.67	
9996	78	F	24.8	Never	0	Pollen	Low	Moderate	Indoor	Diabetes	0.72	
9997	58	M	30.1	Former	1	Pollen	Low	Moderate	Indoor	unknown	0.28	

CHECK IF THERE ANY NULL VALUES:

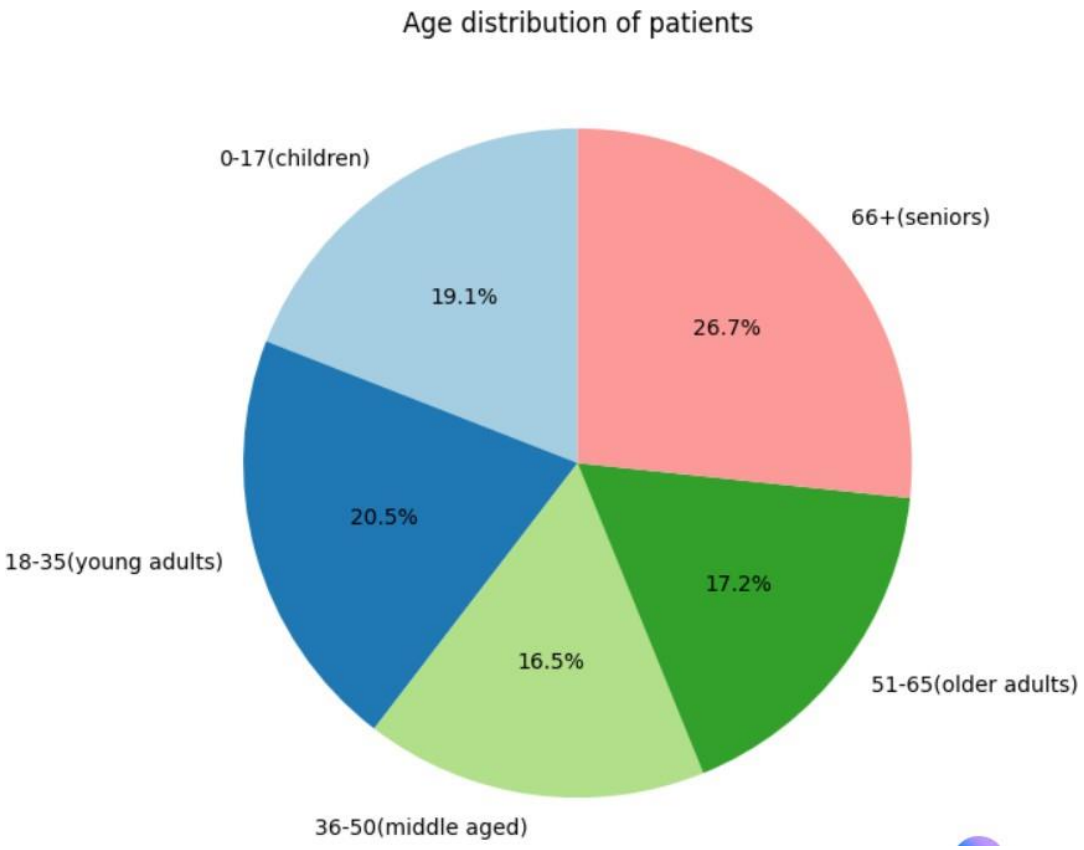
```
df.isnull().any()
```

	0
Age	False
Gender	False
BMI	False
Smoking_Status	False
Family_History	False
Allergies	False
Air_Pollution_Level	False
Physical_Activity_Level	False
Occupation_Type	False
Comorbidities	False
Medication_Adherence	False
Number_of_ER_Visits	False
Peak_Expiratory_Flow	False
FeNO_Level	False
Has_Asthma	False
Asthma_Control_Level	False

EXPLORATORY DATA ANALYSIS (EDA)

1. Which age group has highest number of patients?

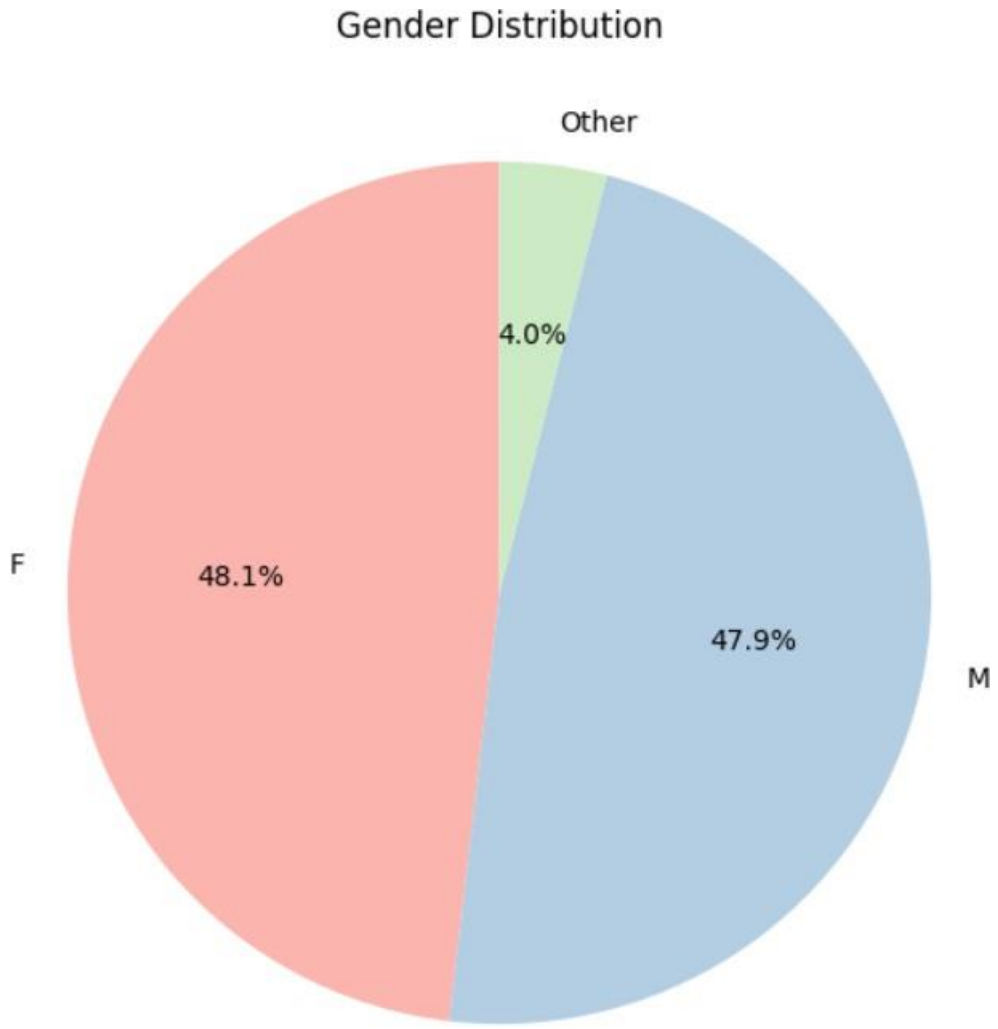
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
bins=[0,18,36,51,66,100]
labels=["0-17(children)","18-35(young adults)","36-50(middle aged)","51-65(older adults)","66+(seniors)"]
df["age_group"]=pd.cut(df["Age"],bins=bins,labels=labels,right=False)
age_counts=df["age_group"].value_counts().sort_index()
plt.figure(figsize=(7,7))
plt.pie(age_counts,labels=age_counts.index,autopct="%1.1f%%",startangle=90,colors=plt.cm.Paired.colors)
plt.title("Age distribution of patients")
plt.show()
```



This pie chart shows the distribution of different age groups in a population. The largest group is seniors (66+) at 26.7%, followed by young adults (18–35) at 20.5%. Children (0–17) make up 19.1%, older adults (51–65) account for 17.2%, and middle-aged adults (36–50) form 16.5%. Overall, the chart indicates that seniors represent the highest proportion, while middle-aged adults represent the lowest among the age groups.

2. Most asthma patients consist in which gender?

```
gender_counts = df["Gender"].value_counts()
plt.figure(figsize=(7, 7))
plt.pie(gender_counts, labels=gender_counts.index, autopct="%1.1f%%", startangle=90, colors=plt.cm.Pastel1.colors)
plt.title("Gender Distribution")
plt.show()
```

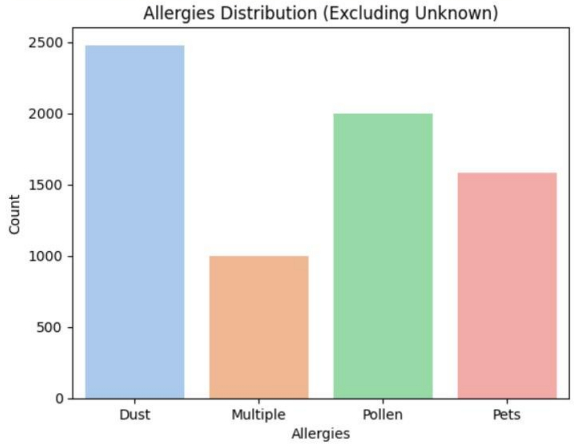


This pie chart shows the gender distribution of the population. Females (F) make up the largest portion at 48.1%, followed closely by males (M) at 47.9%, while others represent a small share of 4%. Overall, the distribution between males and females is nearly equal, with a slight majority of females.

3. Does allergies is a main reason for asthma?

```
df_filtered = df[df["Allergies"] != "unknown"]

plt.figure()
sns.countplot(x="Allergies", data=df_filtered, palette="pastel")
plt.title("Allergies Distribution (Excluding Unknown)")
plt.xlabel("Allergies")
plt.ylabel("Count")
plt.show()
```

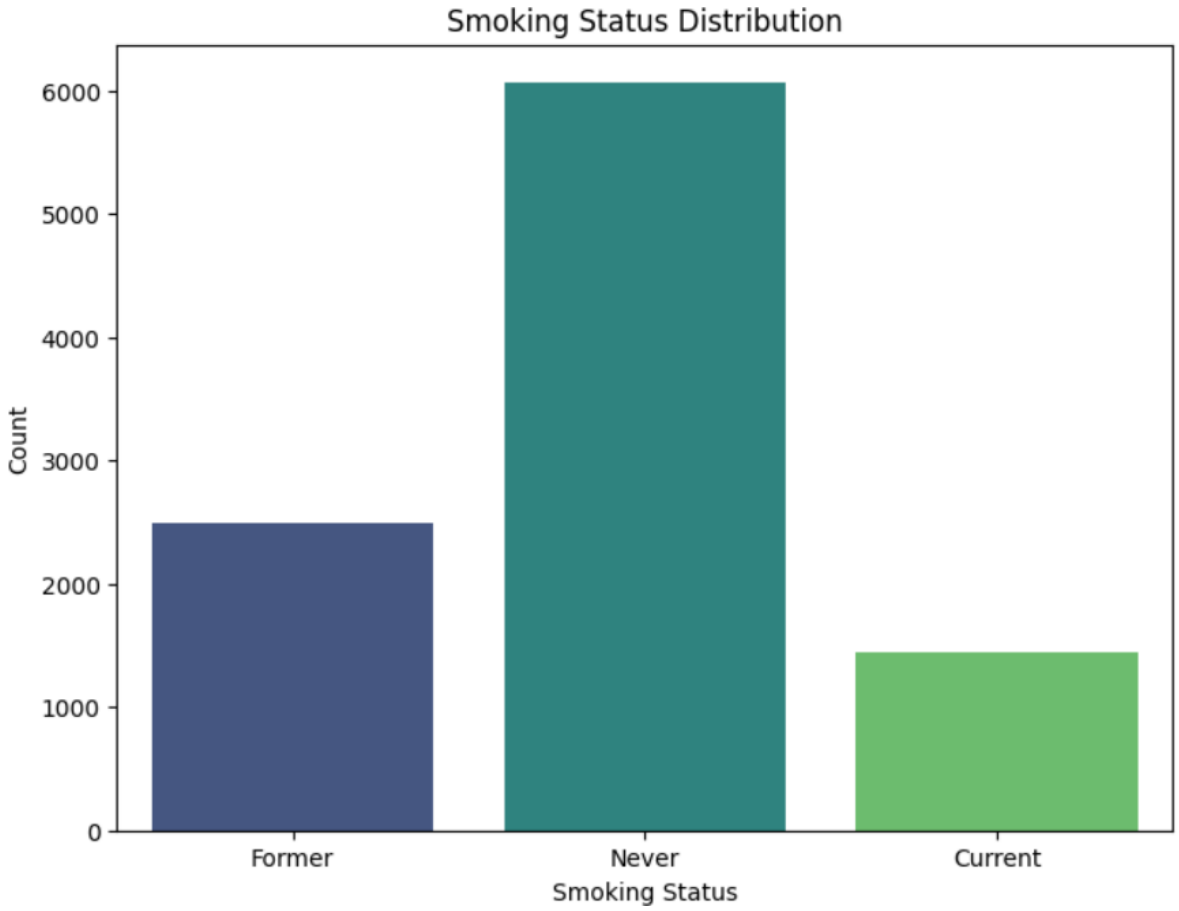
```
/tmp/ipython-input-1808769649.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect

  sns.countplot(x="Allergies", data=df_filtered, palette="pastel")
```



The bar chart shows that dust allergies are the most common among patients, followed by pollen, pet, and multiple allergies. This suggests that environmental allergens like dust play a major role in triggering asthma or allergic reactions in this dataset.

4. Are smokers more likely to have asthma compared to non-smokers?

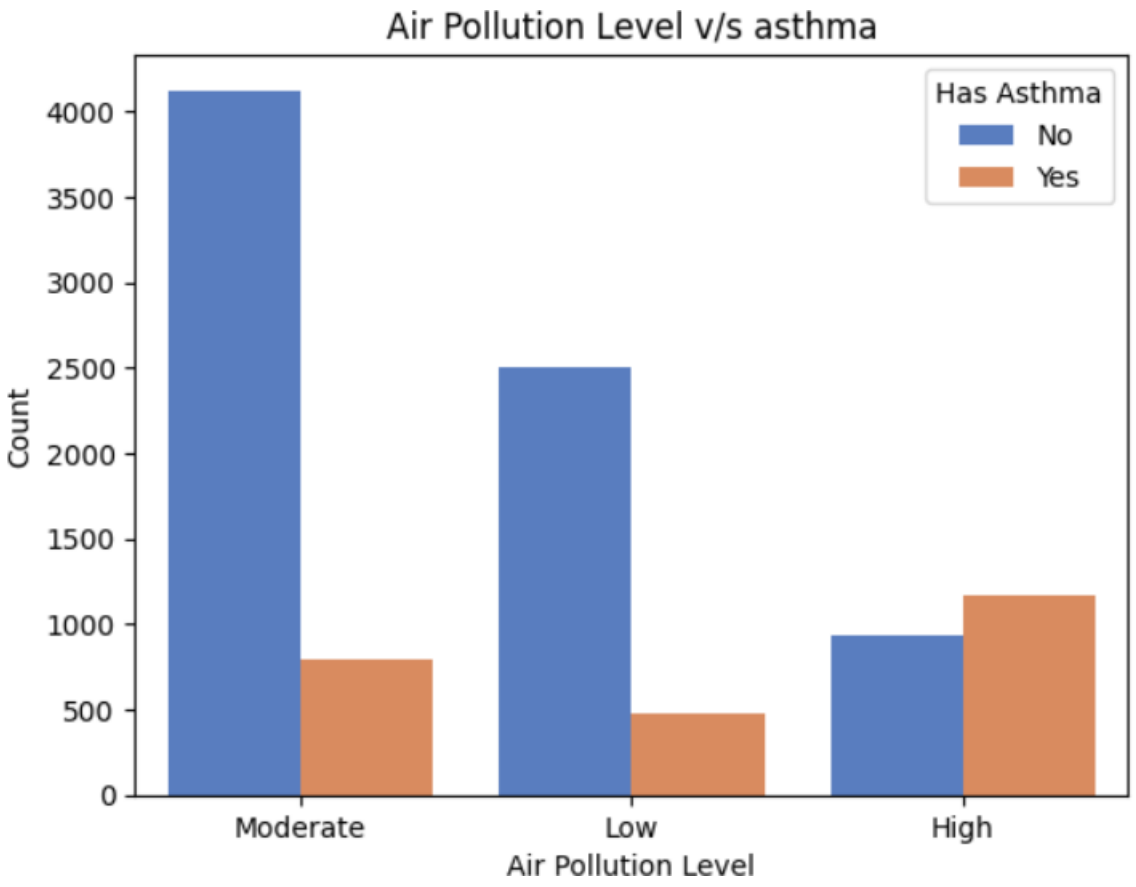
```
plt.figure(figsize=(8, 6))
sns.countplot(x="Smoking_Status", data=df, palette="viridis")
plt.title("Smoking Status Distribution")
plt.xlabel("Smoking Status")
plt.ylabel("Count")
plt.show()
```



The chart shows that the majority of individuals are never smokers, followed by former smokers, while current smokers form the smallest group. This indicates that most people in the dataset have no active smoking habit, though a smaller portion has a history of smoking, which could still influence asthma risk.

5. **Does living in areas with higher air pollution levels increase asthma rates?**

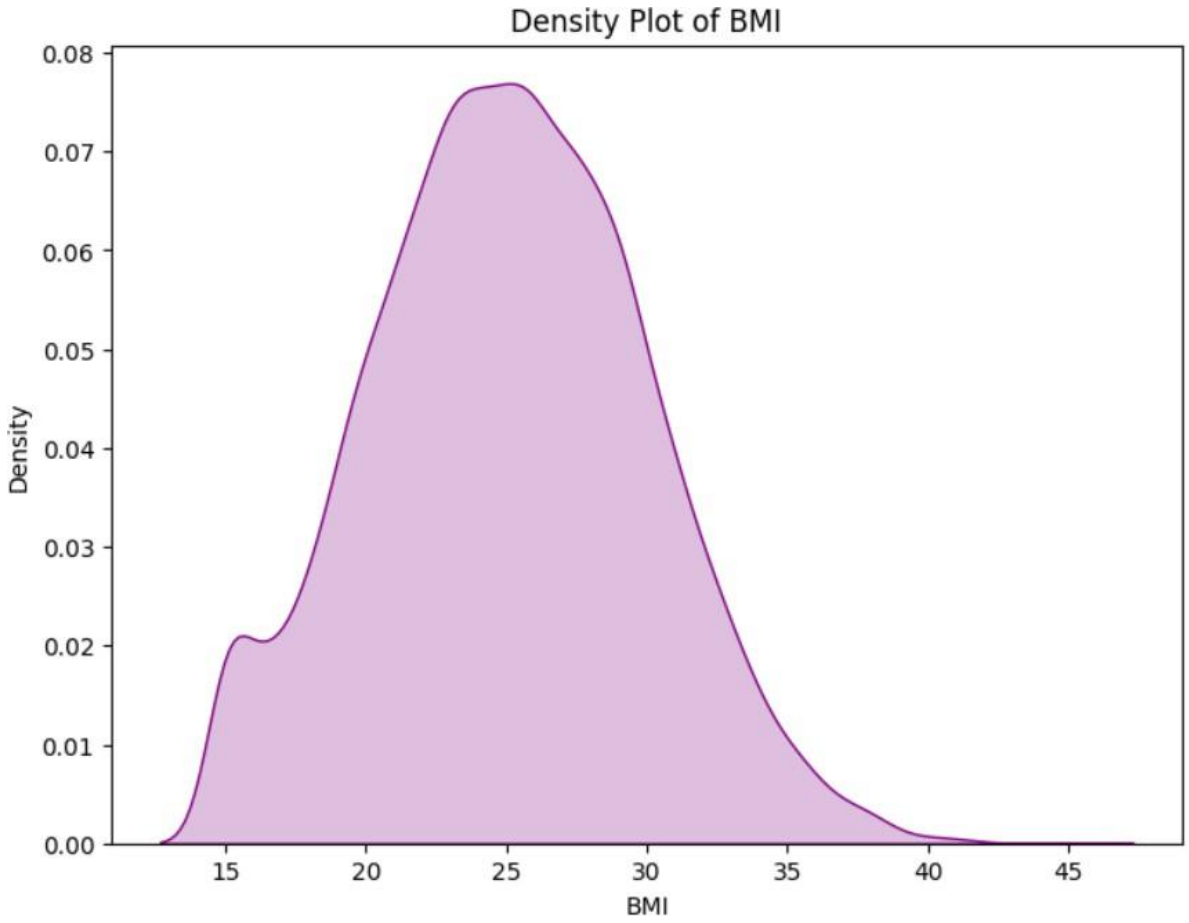
```
plt.figure()
sns.countplot(x="Air_Pollution_Level",hue="Has_Asthma",data=df,palette="muted")
plt.title("Air Pollution Level v/s asthma")
plt.xlabel("Air Pollution Level")
plt.ylabel("Count")
plt.legend(title="Has Asthma",labels=["No","Yes"])
plt.show()
```



The graph shows that in areas with Moderate and Low air pollution, fewer people have asthma than those who do not. However, in areas with High air pollution, the number of people who have asthma is greater than the number of people who do not, indicating a clear link between high air pollution levels and increased asthma prevalence.

6. Do people with high asthma have higher BMI?

```
plt.figure(figsize=(8, 6))
sns.kdeplot(df["BMI"], fill=True, color="purple")
plt.title("Density Plot of BMI")
plt.xlabel("BMI")
plt.ylabel("Density")
plt.show()
```

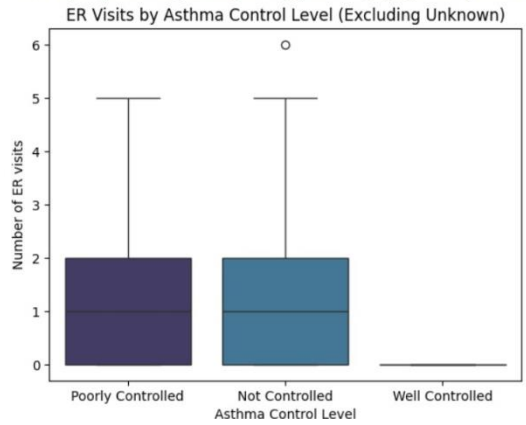


The Density Plot shows the distribution of Body Mass Index (BMI) data. The distribution is mainly unimodal, peaking sharply around 25 kg/m², indicating this is the most common BMI in the dataset (borderline normal/overweight). The data is somewhat positively skewed, with a long tail extending toward higher BMI values (obesity). A small initial peak is visible on the far left, suggesting a minor subgroup of individuals with very low, underweight BMIs (around 15-17).

7. How does the number of ER visits vary among well-Controlled and poorly controlled Asthma patients?

```
df_filtered = df[df["Asthma_Control_Level"] != "unknown"]

plt.figure()
sns.boxplot(x="Asthma_Control_Level", y="Number_of_ER_Visits", data=df_filtered, palette="mako")
plt.title("ER Visits by Asthma Control Level (Excluding Unknown)")
plt.xlabel("Asthma Control Level")
plt.ylabel("Number of ER visits")
plt.show()
```

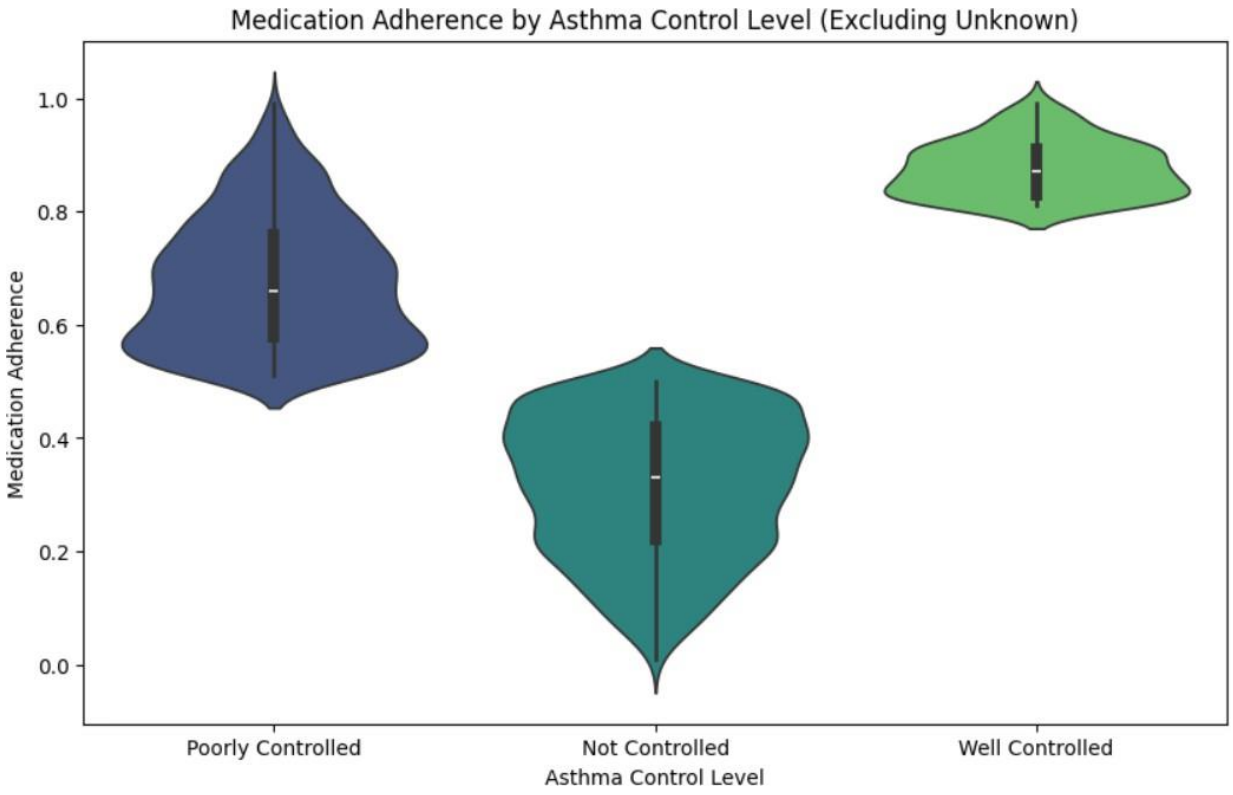
```
/tmp/ipython-input-1939208008.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x="Asthma_Control_Level", y="Number_of_ER_Visits", data=df_filtered, palette="mako")
```



If people control their asthma well, they almost never have to visit the emergency room. However, if their asthma is poorly or not controlled, they are much more likely to have to go to the emergency room, sometimes several times.

8. How does medication adherence affect asthma control?

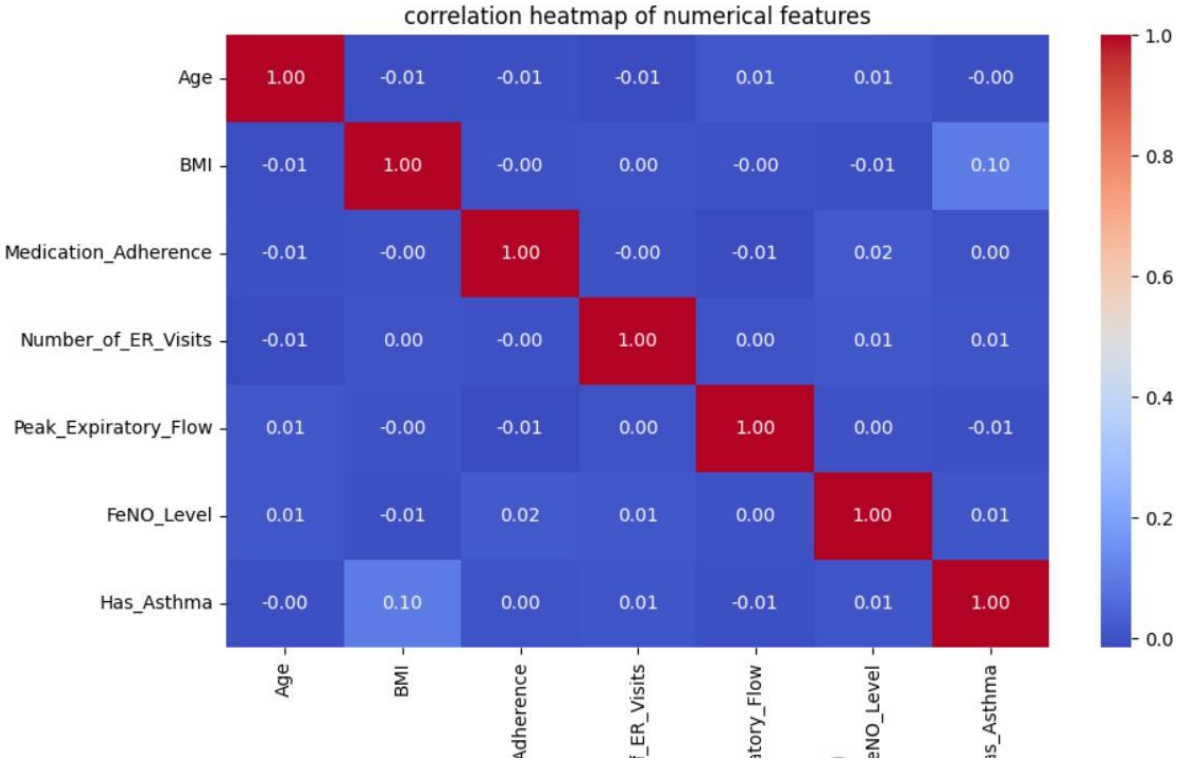
```
df_filtered = df[df["Asthma_Control_Level"] != "unknown"]

plt.figure(figsize=(10, 6))
sns.violinplot(x="Asthma_Control_Level", y="Medication_Adherence", data=df_filtered, palette="viridis")
plt.title("Medication Adherence by Asthma Control Level (Excluding Unknown)")
plt.xlabel("Asthma Control Level")
plt.ylabel("Medication Adherence")
plt.show()
```



High medication adherence is strongly associated with Well Controlled asthma, and low adherence is most prominent among those with Not Controlled asthma. However, it is noteworthy that even a large group of people with Poorly Controlled asthma report high adherence, suggesting that adherence alone may not be the only factor, and other issues (like correct technique, environmental triggers, or prescribed dosage) might also be at play in this group.

9. Which numerical factors are most strongly correlated with having asthma?

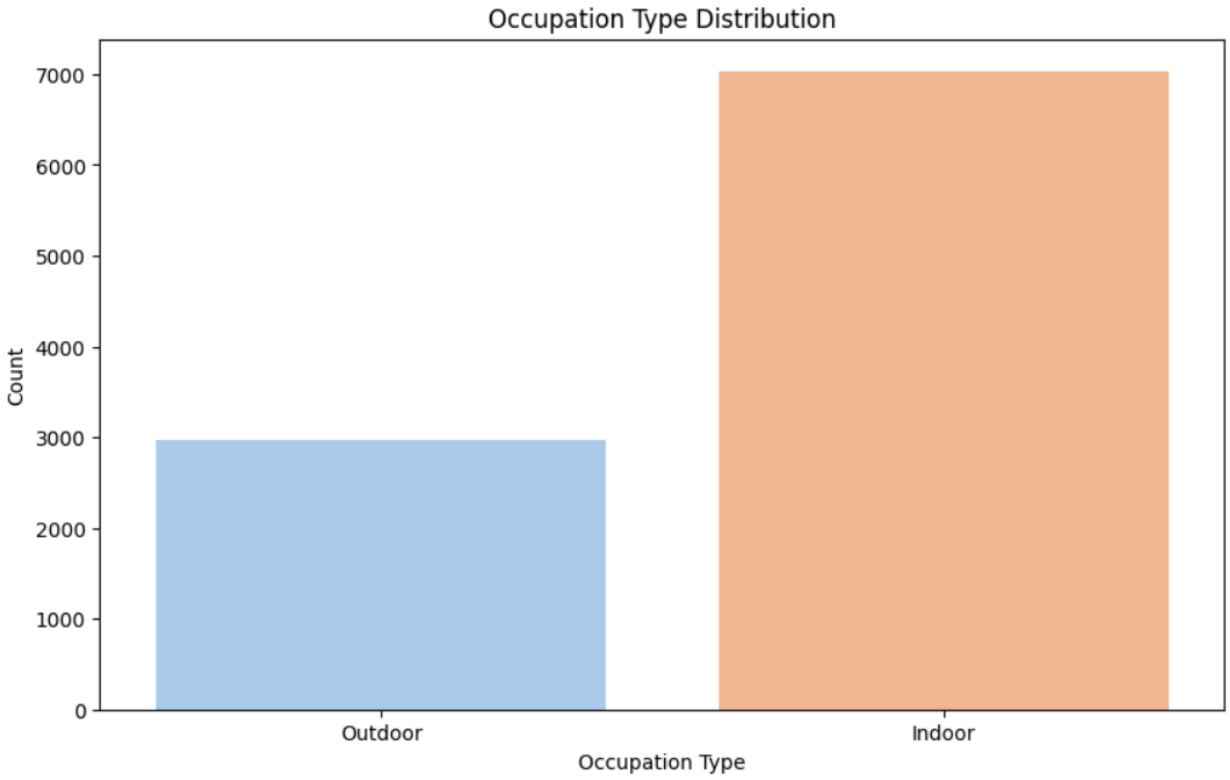
```
plt.figure(figsize=(10,6))
corr=df[["Age","BMI","Medication_Adherence","Number_of_ER_Visits","Peak_Expiratory_Flow","FeNO_Level","Has_Asthma"]].corr()
sns.heatmap(corr,annot=True,cmap="coolwarm",fmt=".2f")
plt.title("correlation heatmap of numerical features")
plt.show()
```



the heatmap reveals that the numerical features in the dataset are largely independent of one another, with only a very weak positive association between BMI and the presence of asthma.

10. What is the distribution of occupation types among the patients?

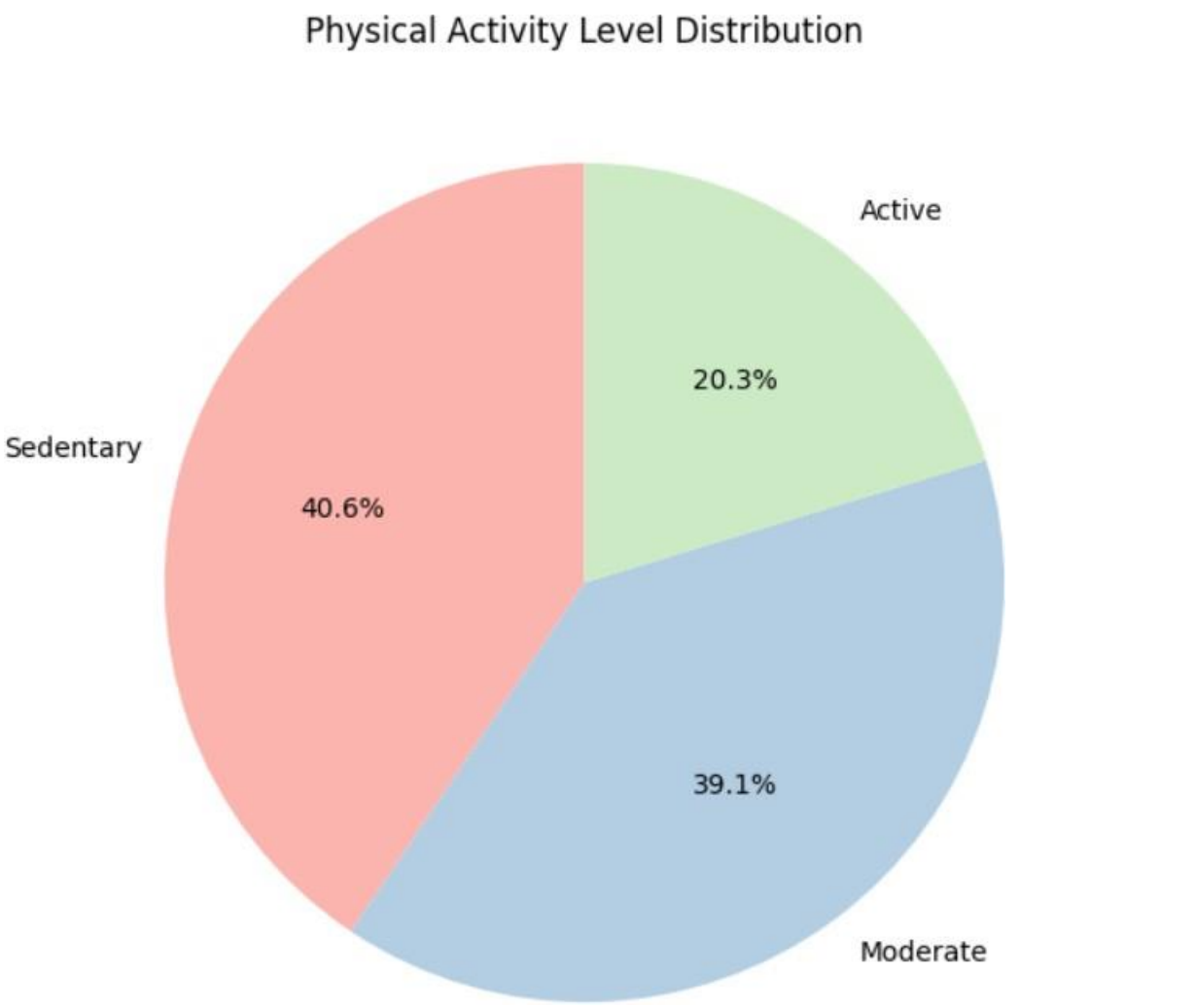
```
plt.figure(figsize=(10, 6))
sns.countplot(x="Occupation_Type", data=df, palette="pastel")
plt.title("Occupation Type Distribution")
plt.xlabel("Occupation Type")
plt.ylabel("Count")
plt.show()
```



The count plot reveals a significant skew in the dataset's occupation distribution. There are substantially more individuals working in Indoor occupations (around 7,000) compared to those in Outdoor occupations (approximately 3,000). This means the dataset has more than twice as many people in indoor jobs, a factor that should be considered when analyzing any occupation-related health trends.

11. What is the distribution of physical activity levels among the patients?

```
activity_counts = df["Physical_Activity_Level"].value_counts()
plt.figure(figsize=(7, 7))
plt.pie(activity_counts, labels=activity_counts.index, autopct="%1.1f%%", startangle=90, colors=plt.cm.Pastel1.colors)
plt.title("Physical Activity Level Distribution")
plt.show()
```



This pie chart shows the distribution of physical activity levels among patients. The largest group is Sedentary at 35.4%, followed by Moderate at 34.3%, and Active at 30.3%.This indicates that a significant portion of the patients have low levels of physical activity.

CONCLUSION

After cleaning and visualizing the synthetic asthma dataset, several important insights were observed. The data revealed that environmental, lifestyle, and genetic factors play a key role in influencing asthma occurrence and control. Dust allergies and air pollution exposure were among the most common triggers, while family history and high FeNO levels showed strong associations with asthma diagnosis.

Most individuals in the dataset were non-smokers, suggesting that while smoking increases risk, other factors such as poor air quality and obesity (high BMI) also contribute significantly. Additionally, patients with better medication adherence had fewer ER visits and better asthma control, highlighting the importance of consistent treatment.

Overall, the dataset emphasizes that asthma is a multi-factorial condition influenced by genetics, environment, and behavior. Data-driven analysis like this can help healthcare professionals identify high-risk groups and develop preventive and management strategies to improve patient outcomes.