```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
healthcare=pd.read_excel(r"C:\Users\DELL\Downloads\healthcare_patient_data (1).xlsx")
healthcare.head()
```

	Patient ID	Age	Gender	BMI	Blood Pressure	Cholesterol	Smoking	Exercise Hours	Diagnosis	Treatment Cost	Region
0	P00001	32	M	20.3	135/88	259	No	3	Heart Disease	5802	East
1	P00002	61	F	18.2	100/65	230	Yes	4	Diabetes	3443	East
2	P00003	48	M	28.3	138/90	257	Yes	7	Diabetes	3302	East
3	P00004	35	F	30.4	120/80	235	Yes	6	Heart Disease	4996	North
4	P00005	43	M	33.6	100/65	218	No	6	Diabetes	3288	East

```
healthcare.shape
```

```
(35000, 11)
```

```
healthcare.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35000 entries, 0 to 34999
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Patient ID      35000 non-null  object
 1   Age             35000 non-null  int64
 2   Gender          35000 non-null  object
 3   BMI             35000 non-null  float64
 4   Blood Pressure  35000 non-null  object
 5   Cholesterol     35000 non-null  int64
 6   Smoking         35000 non-null  object
 7   Exercise Hours  35000 non-null  int64
 8   Diagnosis       26251 non-null  object
 9   Treatment Cost  35000 non-null  int64
 10  Region          35000 non-null  object
dtypes: float64(1), int64(4), object(6)
memory usage: 2.9+ MB
```

```
healthcare.isnull().sum()
```

```
Patient ID           0
Age                  0
Gender               0
BMI                  0
Blood Pressure       0
Cholesterol          0
Smoking              0
Exercise Hours       0
Diagnosis         8749
Treatment Cost       0
Region               0
dtype: int64
```

```
healthcare["Diagnosis"].value_counts()
```

```
Diagnosis
Hypertension     8836
Diabetes         8711
Heart Disease    8704
Name: count, dtype: int64
```

this is your first way to replace null values-- central tendenciees -- mean,medain mode

```
healthcare["Diagnosis"].fillna(healthcare["Diagnosis"].mode()[0],inplace=True)
```

```
healthcare.isnull().sum()
```

```
Patient ID       0
Age              0
Gender           0
BMI              0
Blood Pressure   0
Cholesterol      0
Smoking          0
Exercise Hours   0
Diagnosis        0
Treatment Cost   0
Region           0
dtype: int64
```

```
healthcare["Diagnosis"].value_counts()
```

```
Diagnosis
Hypertension     17585
Diabetes          8711
Heart Disease     8704
Name: count, dtype: int64
```

second way of doing is if you are working with helath care data is to remove the rows in which these confusion are.

```
healthcare2=pd.read_excel(r"C:\Users\DELL\Downloads\healthcare_patient_data (1).xlsx")
healthcare2.head()
```

	Patient ID	Age	Gender	BMI	Blood Pressure	Cholesterol	Smoking	Exercise Hours	Diagnosis	Treatment Cost	Region
0	P00001	32	M	20.3	135/88	259	No	3	Heart Disease	5802	East
1	P00002	61	F	18.2	100/65	230	Yes	4	Diabetes	3443	East
2	P00003	48	M	28.3	138/90	257	Yes	7	Diabetes	3302	East
3	P00004	35	F	30.4	120/80	235	Yes	6	Heart Disease	4996	North
4	P00005	43	M	33.6	100/65	218	No	6	Diabetes	3288	East

```
healthcare2.isnull().sum()
```

```
Patient ID          0
Age                 0
Gender              0
BMI                 0
Blood Pressure      0
Cholesterol         0
Smoking             0
Exercise Hours      0
Diagnosis        8749
Treatment Cost      0
Region              0
dtype: int64
```

```
healthcare2=healthcare2.dropna()
```

```
healthcare2.isnull().sum()
```

```
Patient ID       0
Age              0
Gender           0
BMI              0
Blood Pressure   0
Cholesterol      0
Smoking          0
Exercise Hours   0
Diagnosis        0
Treatment Cost   0
Region           0
dtype: int64
```

```
healthcare2.shape
```

```
(26251, 11)
```

if missing values are less than 5-7% thn consider dropping them ..

ex- you hve 4 coulmns missing vales -- 23,4,12,1-- firstly 23% imputation-- thn 12%

```
healthcare2[['Systolic','Diastolic']]=healthcare2['Blood Pressure'].str.split('/',expand=True).astype(float)
```

```
healthcare2.head()
```

	Patient ID	Age	Gender	BMI	Blood Pressure	Cholesterol	Smoking	Exercise Hours	Diagnosis	Treatment Cost	Region	Systolic	Diastolic
0	P00001	32	M	20.3	135/88	259	No	3	Heart Disease	5802	East	135.0	88.0
1	P00002	61	F	18.2	100/65	230	Yes	4	Diabetes	3443	East	100.0	65.0
2	P00003	48	M	28.3	138/90	257	Yes	7	Diabetes	3302	East	138.0	90.0
3	P00004	35	F	30.4	120/80	235	Yes	6	Heart Disease	4996	North	120.0	80.0

```
healthcare2.drop(columns=["Blood Pressure"],inplace=True)
```

```
healthcare2.head()
```

	Patient ID	Age	Gender	BMI	Cholesterol	Smoking	Exercise Hours	Diagnosis	Treatment Cost	Region	Systolic	Diastolic
0	P00001	32	M	20.3	259	No	3	Heart Disease	5802	East	135.0	88.0
1	P00002	61	F	18.2	230	Yes	4	Diabetes	3443	East	100.0	65.0
2	P00003	48	M	28.3	257	Yes	7	Diabetes	3302	East	138.0	90.0
3	P00004	35	F	30.4	235	Yes	6	Heart Disease	4996	North	120.0	80.0
4	P00005	43	M	33.6	218	No	6	Diabetes	3288	East	100.0	65.0

```
healthcare2["Diagnosis"].value_counts()
```

```
Diagnosis
Hypertension     8836
Diabetes         8711
Heart Disease    8704
Name: count, dtype: int64
```

```
healthcare.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Age	35000.0	49.001200	18.205521	18.0	33.0	49.0	65.0	80.0
BMI	35000.0	26.490106	4.894053	18.0	22.2	26.5	30.7	35.0
Cholesterol	35000.0	215.022600	31.961938	160.0	187.0	215.0	243.0	270.0
Exercise Hours	35000.0	3.514800	2.297897	0.0	1.0	4.0	6.0	7.0
Treatment Cost	35000.0	2469.280886	1776.684292	100.0	1200.0	1993.0	3494.0	6000.0

```
average_cost_Treatment=healthcare2.groupby('Diagnosis')['Treatment Cost'].mean()
average_cost_Treatment
```

```
Diagnosis
Diabetes         3001.512570
Heart Disease    5002.425437
Hypertension     1598.295835
Name: Treatment Cost, dtype: float64
```

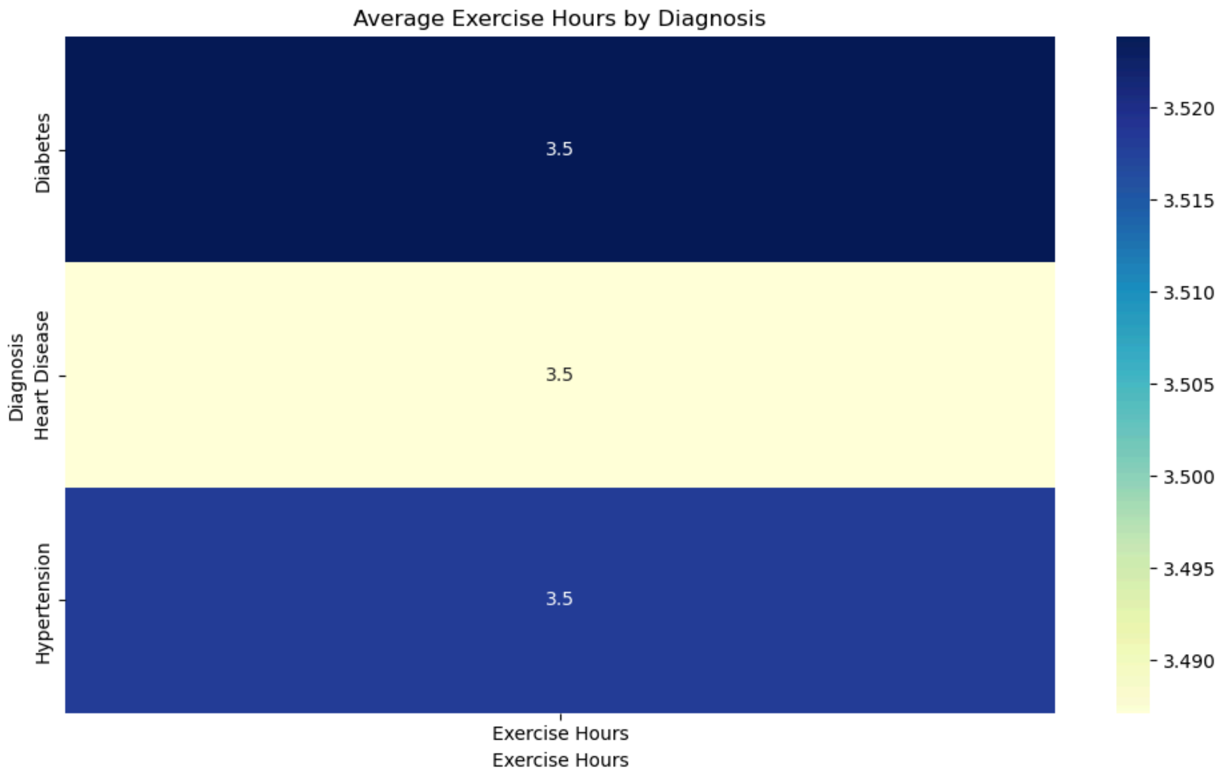
Health Trends and Lifestyle Analysis

```
# Grouping the data correctly
heatmap_data = healthcare2.groupby('Diagnosis')['Exercise Hours'].mean().to_frame()

# Double-check the data
print(heatmap_data.shape)
print(heatmap_data.head())

# Plotting
plt.figure(figsize=(10, 6))
sns.heatmap(heatmap_data, annot=True, cmap='YlGnBu')
plt.title('Average Exercise Hours by Diagnosis')
plt.xlabel('Exercise Hours')
plt.ylabel('Diagnosis')
plt.tight_layout()
plt.show()
```

```
(3, 1)
               Exercise Hours
Diagnosis
Diabetes             3.523820
Heart Disease        3.487132
Hypertension         3.517995
```



Start coding or generate with AI.