```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

1. Load the dataset

> Indented block

```
df = pd.read_excel("/DoctorVisits (2).xlsx")
```

2. Display first 15 rows

```
print(df.head(15))
```

```
    Unnamed: 0  visits  gender   age  income  illness  reduced  health  \
0            1       1  female  0.19    0.55        1        4       1
1            2       1  female  0.19    0.45        1        2       1
2            3       1    male  0.19    0.90        3        0       0
3            4       1    male  0.19    0.15        1        0       0
4            5       1    male  0.19    0.45        2        5       1
5            6       1  female  0.19    0.35        5        1       9
6            7       1  female  0.19    0.55        4        0       2
7            8       1  female  0.19    0.15        3        0       6
8            9       1  female  0.19    0.65        2        0       5
9           10       1    male  0.19    0.15        1        0       0
10          11       1    male  0.19    0.45        1        0       0
11          12       1    male  0.19    0.25        2        0       2
12          13       2    male  0.19    0.55        3       13       1
13          14       1    male  0.19    0.45        4        7       6
14          15       1    male  0.19    0.25        3        1       0

   private freepoor freerepat nchronic lchronic
0      yes       no         no       no       no
1      yes       no         no       no       no
2       no       no         no       no       no
3       no       no         no       no       no
4       no       no         no      yes       no
5       no       no         no      yes       no
6       no       no         no       no       no
7       no       no         no       no       no
8      yes       no         no       no       no
9      yes       no         no       no       no
10      no       no         no       no       no
11      no       no        yes       no       no
12      no       no         no      yes       no
13      no       no         no      yes       no
14     yes       no         no      yes       no
```

3. Display complete information about the columns of the dataset such as Column name,Count, Data type and overall memory usage

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5190 entries, 0 to 5189
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  5190 non-null   int64
 1   visits      5190 non-null   int64
 2   gender      5190 non-null   object
 3   age         5190 non-null   float64
 4   income      5190 non-null   float64
 5   illness     5190 non-null   int64
```

```
 6   reduced     5190 non-null   int64
 7   health      5190 non-null   int64
 8   private     5190 non-null   object
 9   freepoor    5190 non-null   object
 10  freerepat   5190 non-null   object
 11  nchronic    5190 non-null   object
 12  lchronic    5190 non-null   object
dtypes: float64(2), int64(5), object(6)
memory usage: 527.2+ KB
```

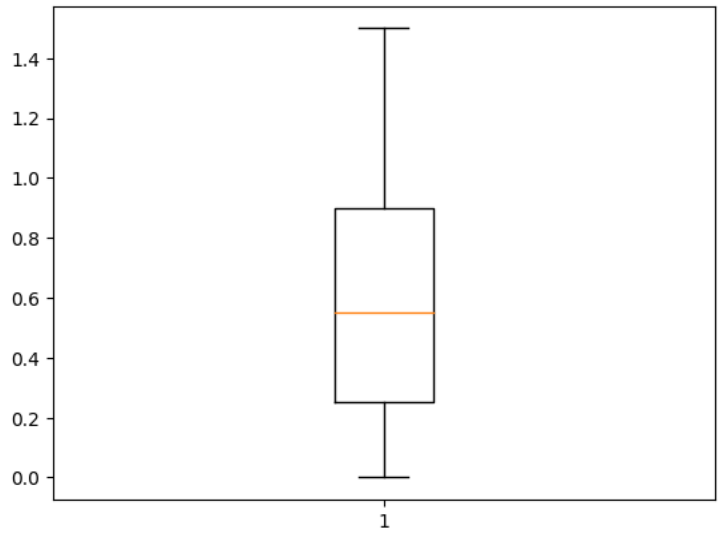
4. Find out the total no: of people based on their count of illness

```
df["illness"].value_counts()
```

```
1    1638
0    1554
2     946
3     542
4     274
5     236
Name: illness, dtype: int64
```

5. Visualize and analyse the maximum,minimum and medium income

```
y = list(df.income)
plt.boxplot(y)
plt.show()
```



6. Find out the no of days of reduced activity of male and female seperatly due to illness

```
df.groupby(['gender', 'reduced']).mean()
```

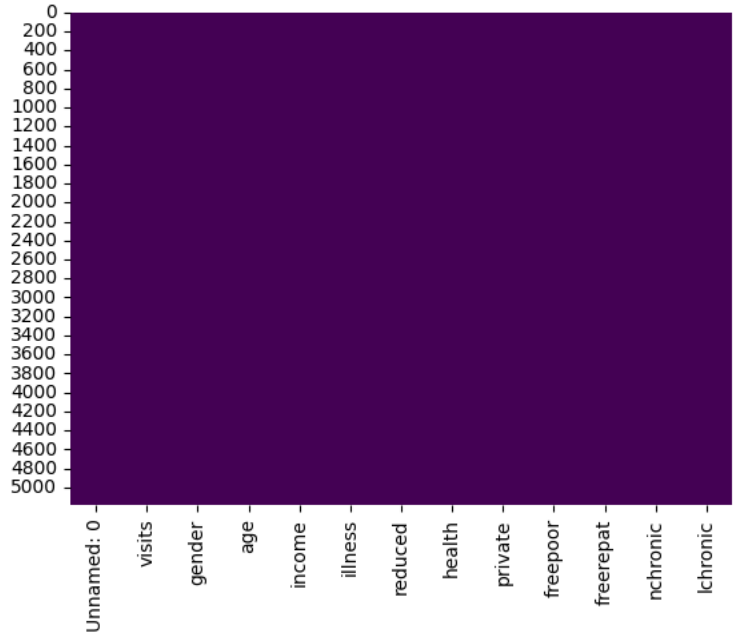
```
<ipython-input-9-40781631630e>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a future v
  df.groupby(['gender', 'reduced']).mean()
```

gender	reduced	Unnamed: 0	visits	age	income	illness	health
female	0	2524.038512	0.229322	0.465755	0.482735	1.462144	1.115098
	1	1985.768421	0.400000	0.325684	0.542105	2.242105	1.610526
	2	1622.618182	0.672727	0.391455	0.560182	2.236364	1.781818
	3	997.311111	1.333333	0.403111	0.516000	2.733333	1.733333
	4	1237.740741	0.851852	0.458889	0.466667	2.222222	2.074074
	5	1169.055556	1.444444	0.401667	0.614444	2.222222	2.500000
	6	1382.545455	1.363636	0.426364	0.622727	2.363636	1.363636
	7	1034.846154	1.384615	0.436154	0.473462	2.653846	2.230769
	8	1883.090909	1.090909	0.471818	0.404545	2.181818	4.000000
	9	1349.000000	0.500000	0.570000	0.825000	3.000000	1.000000
	10	1099.428571	2.142857	0.512857	0.421429	2.571429	2.000000
	12	1661.000000	2.000000	0.720000	0.250000	3.500000	5.500000
	13	906.000000	4.000000	0.720000	0.300000	4.500000	3.500000
	14	1392.112069	1.543103	0.551724	0.427586	2.534483	4.112069
male	0	3008.911019	0.136007	0.344703	0.694398	1.099585	0.924850
	1	2485.158537	0.304878	0.286220	0.676341	1.743902	1.256098
	2	2007.679245	0.471698	0.343585	0.653019	2.358491	1.547170
	3	1909.068966	0.724138	0.334138	0.741379	2.137931	1.689655
	4	1424.000000	0.722222	0.309444	0.869444	2.055556	2.000000
	5	1437.272727	1.136364	0.331818	0.570455	2.272727	2.818182
	6	562.000000	0.833333	0.340000	0.591667	2.500000	2.000000
	7	1716.750000	0.750000	0.314167	0.655000	2.583333	4.333333

7. Visualize is there is any missing value in the dataset based based on a heat map

```
sns.heatmap(df.isnull(),cbar=False,cmap='viridis')
```

```
<Axes: >
```



8. Find out the correlation between variables in the given dataset correlation between different variables

```
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),cbar=True,annot=True,cmap='Blues')
```

```
<ipython-input-14-545168e7e9ec>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versior
  sns.heatmap(df.corr(),cbar=True,annot=True,cmap='Blues')
```

9. Analyse how the income of a patient affects the no of visits to the hospital

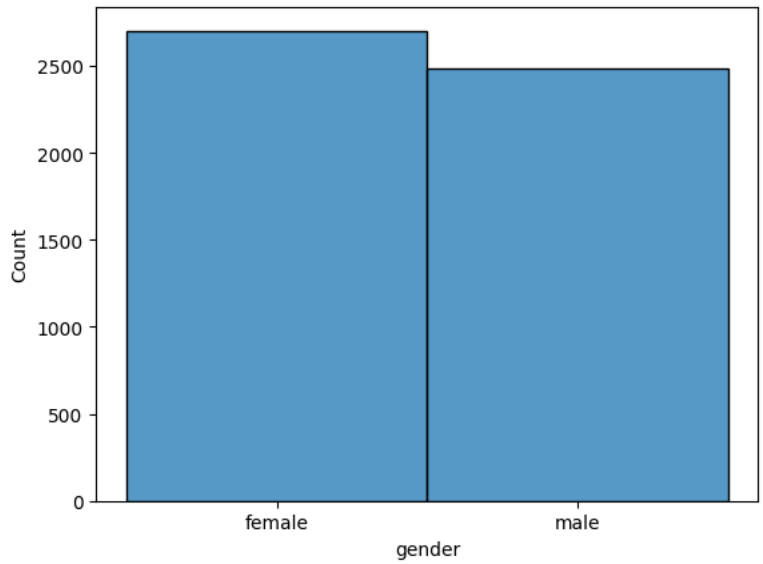
```
plt.figure(figsize=(15,15))
plt.scatter(x='income',y='visits',data=df)
plt.xlabel('income')
plt.ylabel('visits')
```

```
Text(0, 0.5, 'visits')
```

10. Count and visualize the number of males and females affected by illness

```
sns.histplot(df.gender,bins=2)
```

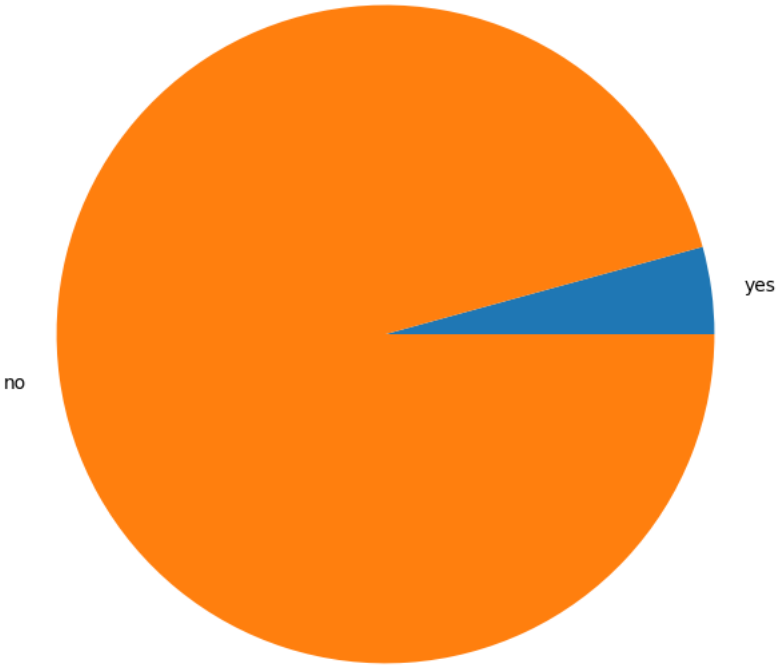
```
<Axes: xlabel='gender', ylabel='Count'>
```



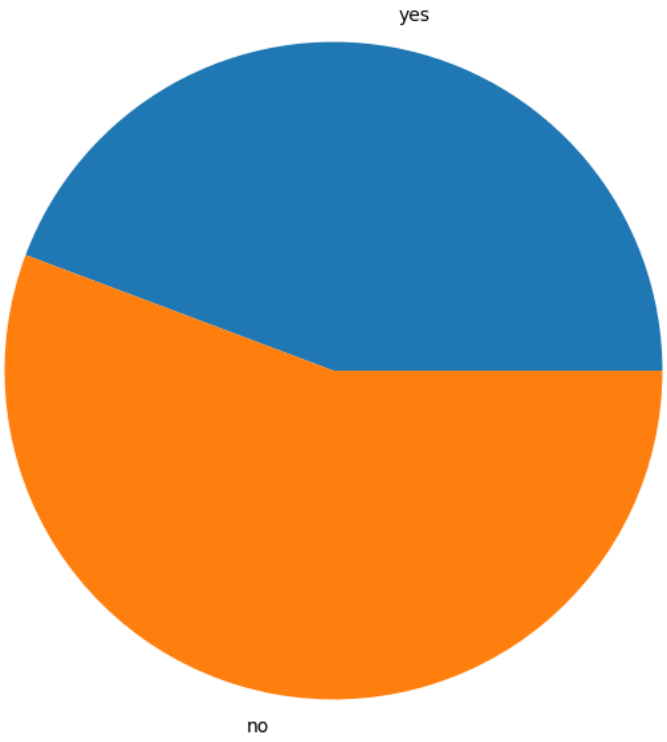
11. Visualize the percentage of people getting govt health Insurance due to low income, due to old age and also the percentage of people having private health insurance

```
label=['yes','no']
Y = df[df['freepoor']=='yes']
N = df[df['freepoor']=='no']
x = [Y.shape[0],N.shape[0]]
plt.figure(figsize=(8,8))
plt.pie(x,labels=label)
plt.title("% of people getting govt health Insurance due to low income")
plt.show()
Y = df[df['private']=='yes']
N = df[df['private']=='no']
x = [Y.shape[0],N.shape[0]]
plt.figure(figsize=(8,8))
plt.pie(x,labels=label)
plt.title("% of people having private health Insurance ")
plt.show()
Y = df[df['freerepat']=='yes']
N = df[df['freerepat']=='no']
x = [Y.shape[0],N.shape[0]]
plt.figure(figsize=(8,8))
plt.pie(x,labels=label)
plt.title("% of people getting govt health Insurance due to old age, disability or veteran status ")
plt.show()
```

% of people getting govt health Insurance due to low income



% of people having private health Insurance



% of people getting govt health Insurance due to old age, disability or veteran status