Import libraries

In [44]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Explore the data

In [45]:
```
df=pd.read_csv("Doctor_visit.csv")
```

In [46]:
```
df.head(20)
```

Out[46]:

	Unnamed: 0	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepat
0	1	1	female	0.19	0.55	1	4	1	yes	no	no
1	2	1	female	0.19	0.45	1	2	1	yes	no	no
2	3	1	male	0.19	0.90	3	0	0	no	no	no
3	4	1	male	0.19	0.15	1	0	0	no	no	no
4	5	1	male	0.19	0.45	2	5	1	no	no	no
5	6	1	female	0.19	0.35	5	1	9	no	no	no
6	7	1	female	0.19	0.55	4	0	2	no	no	no
7	8	1	female	0.19	0.15	3	0	6	no	no	no
8	9	1	female	0.19	0.65	2	0	5	yes	no	no
9	10	1	male	0.19	0.15	1	0	0	yes	no	no
10	11	1	male	0.19	0.45	1	0	0	no	no	no
11	12	1	male	0.19	0.25	2	0	2	no	no	yes
12	13	2	male	0.19	0.55	3	13	1	no	no	no
13	14	1	male	0.19	0.45	4	7	6	no	no	no
14	15	1	male	0.19	0.25	3	1	0	yes	no	no
15	16	1	male	0.19	0.55	2	0	7	no	no	no
16	17	2	male	0.19	0.45	1	0	5	yes	no	no
17	18	1	female	0.19	0.45	1	1	0	no	no	no
18	19	2	female	0.19	0.45	1	0	0	yes	no	no
19	20	1	female	0.19	0.35	1	0	0	yes	no	no

In [47]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  20 non-null     int64  
 1   visits      20 non-null     int64  
 2   gender      20 non-null     object 
 3   age         20 non-null     float64
 4   income      20 non-null     float64
 5   illness     20 non-null     int64  
 6   reduced     20 non-null     int64  
 7   health      20 non-null     int64  
 8   private     20 non-null     object 
 9   freepoor    20 non-null     object 
 10  freerepat   20 non-null     object 
 11  nchronic    20 non-null     object 
 12  lchronic    20 non-null     object 
dtypes: float64(2), int64(5), object(6)
memory usage: 2.2+ KB
```

In [48]:

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

Out[48]:

```
1    9
3    4
2    4
4    2
5    1
Name: illness, dtype: int64
```

In [49]:

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

Out[49]:

```
male      11
female     9
Name: gender, dtype: int64
```

Data cleaning

In [50]:

```
# Handling missing values
df.isnull()
```

Out[50]:

	Unnamed: 0	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepa
0	False	False	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	Fals
5	False	False	False	False	False	False	False	False	False	False	Fals
6	False	False	False	False	False	False	False	False	False	False	Fals
7	False	False	False	False	False	False	False	False	False	False	Fals
8	False	False	False	False	False	False	False	False	False	False	Fals
9	False	False	False	False	False	False	False	False	False	False	Fals
10	False	False	False	False	False	False	False	False	False	False	Fals
11	False	False	False	False	False	False	False	False	False	False	Fals
12	False	False	False	False	False	False	False	False	False	False	Fals
13	False	False	False	False	False	False	False	False	False	False	Fals
14	False	False	False	False	False	False	False	False	False	False	Fals
15	False	False	False	False	False	False	False	False	False	False	Fals
16	False	False	False	False	False	False	False	False	False	False	Fals
17	False	False	False	False	False	False	False	False	False	False	Fals
18	False	False	False	False	False	False	False	False	False	False	Fals
19	False	False	False	False	False	False	False	False	False	False	Fals

In [51]:

```
df.dropna()
```

Out[51]:

	Unnamed: 0	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepat
0	1	1	female	0.19	0.55	1	4	1	yes	no	no
1	2	1	female	0.19	0.45	1	2	1	yes	no	no
2	3	1	male	0.19	0.90	3	0	0	no	no	no
3	4	1	male	0.19	0.15	1	0	0	no	no	no
4	5	1	male	0.19	0.45	2	5	1	no	no	no
5	6	1	female	0.19	0.35	5	1	9	no	no	no
6	7	1	female	0.19	0.55	4	0	2	no	no	no
7	8	1	female	0.19	0.15	3	0	6	no	no	no
8	9	1	female	0.19	0.65	2	0	5	yes	no	no
9	10	1	male	0.19	0.15	1	0	0	yes	no	no
10	11	1	male	0.19	0.45	1	0	0	no	no	no
11	12	1	male	0.19	0.25	2	0	2	no	no	yes
12	13	2	male	0.19	0.55	3	13	1	no	no	no
13	14	1	male	0.19	0.45	4	7	6	no	no	no
14	15	1	male	0.19	0.25	3	1	0	yes	no	no
15	16	1	male	0.19	0.55	2	0	7	no	no	no
16	17	2	male	0.19	0.45	1	0	5	yes	no	no
17	18	1	female	0.19	0.45	1	1	0	no	no	no
18	19	2	female	0.19	0.45	1	0	0	yes	no	no
19	20	1	female	0.19	0.35	1	0	0	yes	no	no

In [52]:

```
# print duplicate records/rows
import pandas as pd
df=pd.read_csv("Doctor_visit.csv")
duplicate_rows=df[df.duplicated()]
print(duplicate_rows)
```

```
Empty DataFrame
Columns: [Unnamed: 0, visits, gender, age, income, illness, reduced, health,
private, freepoor, freerepat, nchronic, lchronic]
Index: []
```

In [53]:
```
# identify duplicate records
df.duplicated()
```

Out[53]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
dtype: bool
```

In [54]:

```
# Removal of duplicates
df.drop_duplicates()
```

Out[54]:

	Unnamed: 0	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepat
0	1	1	female	0.19	0.55	1	4	1	yes	no	no
1	2	1	female	0.19	0.45	1	2	1	yes	no	no
2	3	1	male	0.19	0.90	3	0	0	no	no	no
3	4	1	male	0.19	0.15	1	0	0	no	no	no
4	5	1	male	0.19	0.45	2	5	1	no	no	no
5	6	1	female	0.19	0.35	5	1	9	no	no	no
6	7	1	female	0.19	0.55	4	0	2	no	no	no
7	8	1	female	0.19	0.15	3	0	6	no	no	no
8	9	1	female	0.19	0.65	2	0	5	yes	no	no
9	10	1	male	0.19	0.15	1	0	0	yes	no	no
10	11	1	male	0.19	0.45	1	0	0	no	no	no
11	12	1	male	0.19	0.25	2	0	2	no	no	yes
12	13	2	male	0.19	0.55	3	13	1	no	no	no
13	14	1	male	0.19	0.45	4	7	6	no	no	no
14	15	1	male	0.19	0.25	3	1	0	yes	no	no
15	16	1	male	0.19	0.55	2	0	7	no	no	no
16	17	2	male	0.19	0.45	1	0	5	yes	no	no
17	18	1	female	0.19	0.45	1	1	0	no	no	no
18	19	2	female	0.19	0.45	1	0	0	yes	no	no
19	20	1	female	0.19	0.35	1	0	0	yes	no	no

In [55]:

```
# Handling inconsistent data
df.replace()
```

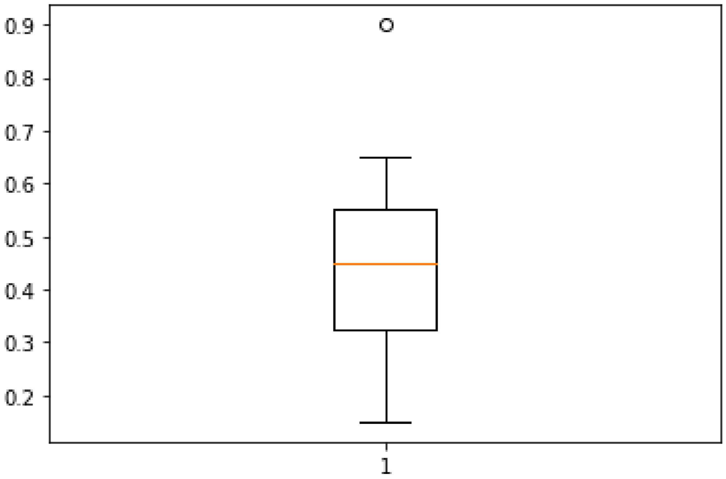
Out[55]:

	Unnamed: 0	visits	gender	age	income	illness	reduced	health	private	freepoor	freerepat
0	1	1	female	0.19	0.55	1	4	1	yes	no	no
1	2	1	female	0.19	0.45	1	2	1	yes	no	no
2	3	1	male	0.19	0.90	3	0	0	no	no	no
3	4	1	male	0.19	0.15	1	0	0	no	no	no
4	5	1	male	0.19	0.45	2	5	1	no	no	no
5	6	1	female	0.19	0.35	5	1	9	no	no	no
6	7	1	female	0.19	0.55	4	0	2	no	no	no
7	8	1	female	0.19	0.15	3	0	6	no	no	no
8	9	1	female	0.19	0.65	2	0	5	yes	no	no
9	10	1	male	0.19	0.15	1	0	0	yes	no	no
10	11	1	male	0.19	0.45	1	0	0	no	no	no
11	12	1	male	0.19	0.25	2	0	2	no	no	yes
12	13	2	male	0.19	0.55	3	13	1	no	no	no
13	14	1	male	0.19	0.45	4	7	6	no	no	no
14	15	1	male	0.19	0.25	3	1	0	yes	no	no
15	16	1	male	0.19	0.55	2	0	7	no	no	no
16	17	2	male	0.19	0.45	1	0	5	yes	no	no
17	18	1	female	0.19	0.45	1	1	0	no	no	no
18	19	2	female	0.19	0.45	1	0	0	yes	no	no
19	20	1	female	0.19	0.35	1	0	0	yes	no	no

visualize maximum, minimum, and medium income

In [56]:

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



Find out the no of days of reduced activity of male and female seperately due to illness

In [57]:

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

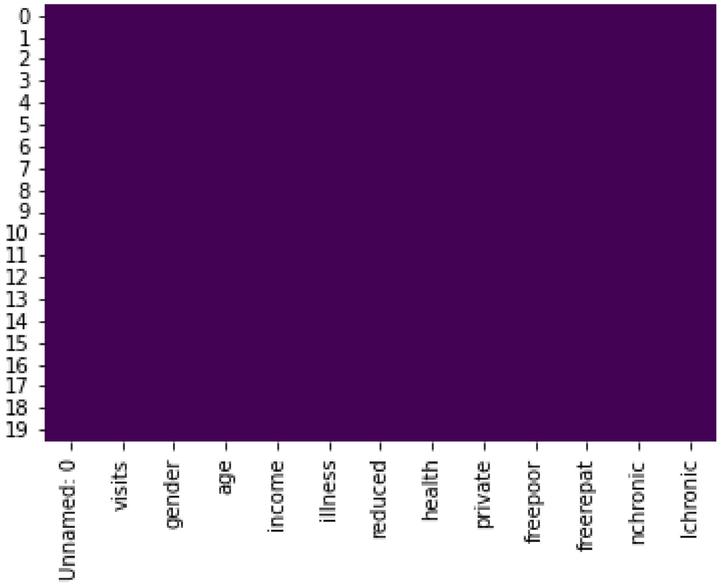
Out[57]:

		Unnamed: 0	visits	age	income	illness	health
gender	**reduced**						
female	**0**	12.600000	1.200000	0.19	0.430000	2.200000	2.6
	1	12.000000	1.000000	0.19	0.400000	3.000000	4.5
	2	2.000000	1.000000	0.19	0.450000	1.000000	1.0
	4	1.000000	1.000000	0.19	0.550000	1.000000	1.0
male	**0**	10.428571	1.142857	0.19	0.414286	1.571429	2.0
	1	15.000000	1.000000	0.19	0.250000	3.000000	0.0
	5	5.000000	1.000000	0.19	0.450000	2.000000	1.0
	7	14.000000	1.000000	0.19	0.450000	4.000000	6.0
	13	13.000000	2.000000	0.19	0.550000	3.000000	1.0

visualize is there any missing values in the dataset

In [58]:
```
sns.heatmap(df.isnull(),cbar=False,cmap='viridis')
```

Out[58]:
```
<AxesSubplot:>
```



correlation between different variables in the given dataset

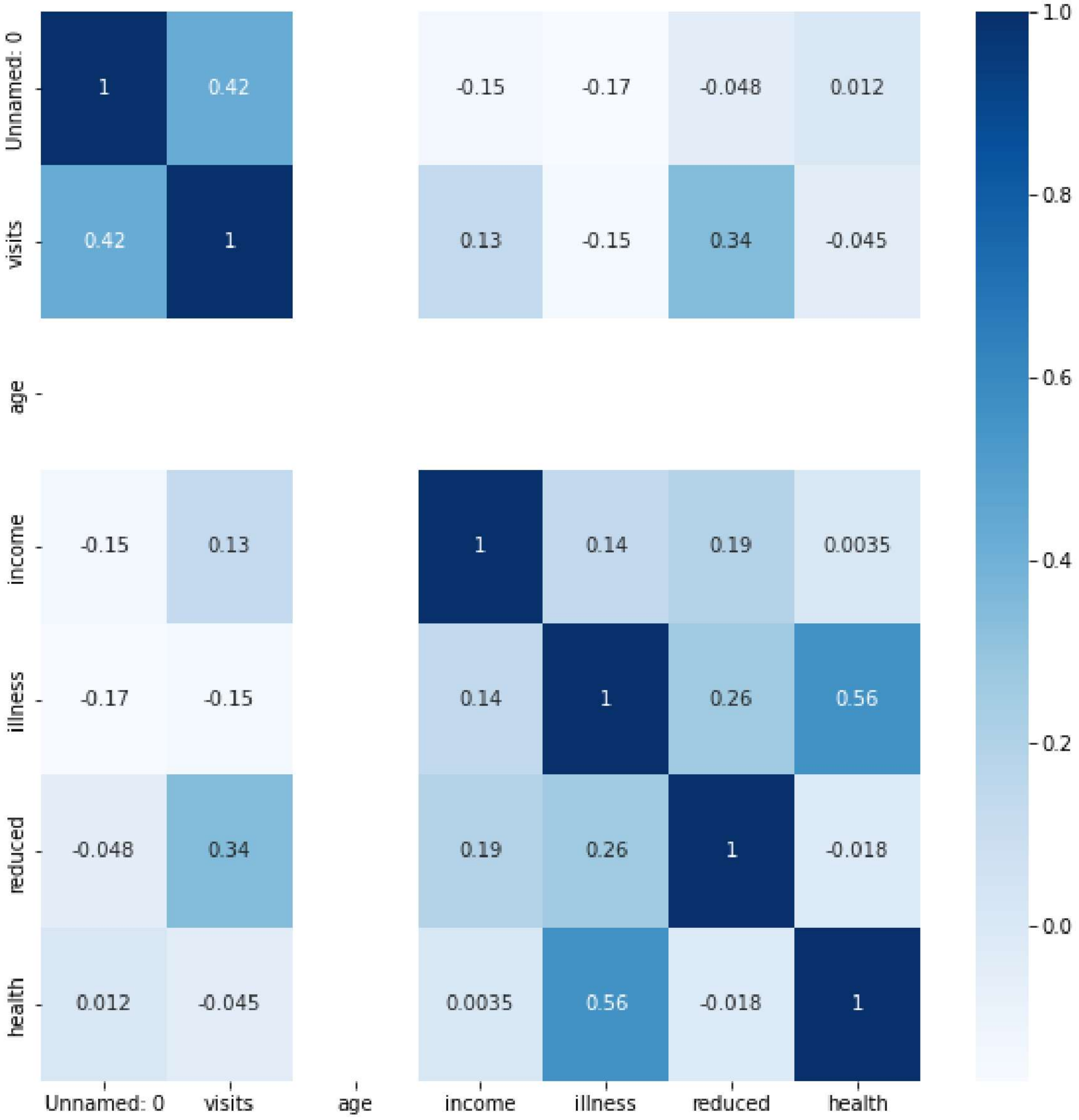
plt figure(figsize=(10,10)) sns.heatmap(df.corr(),cbar=True,annot=True,cmap='Blues')

In [59]:

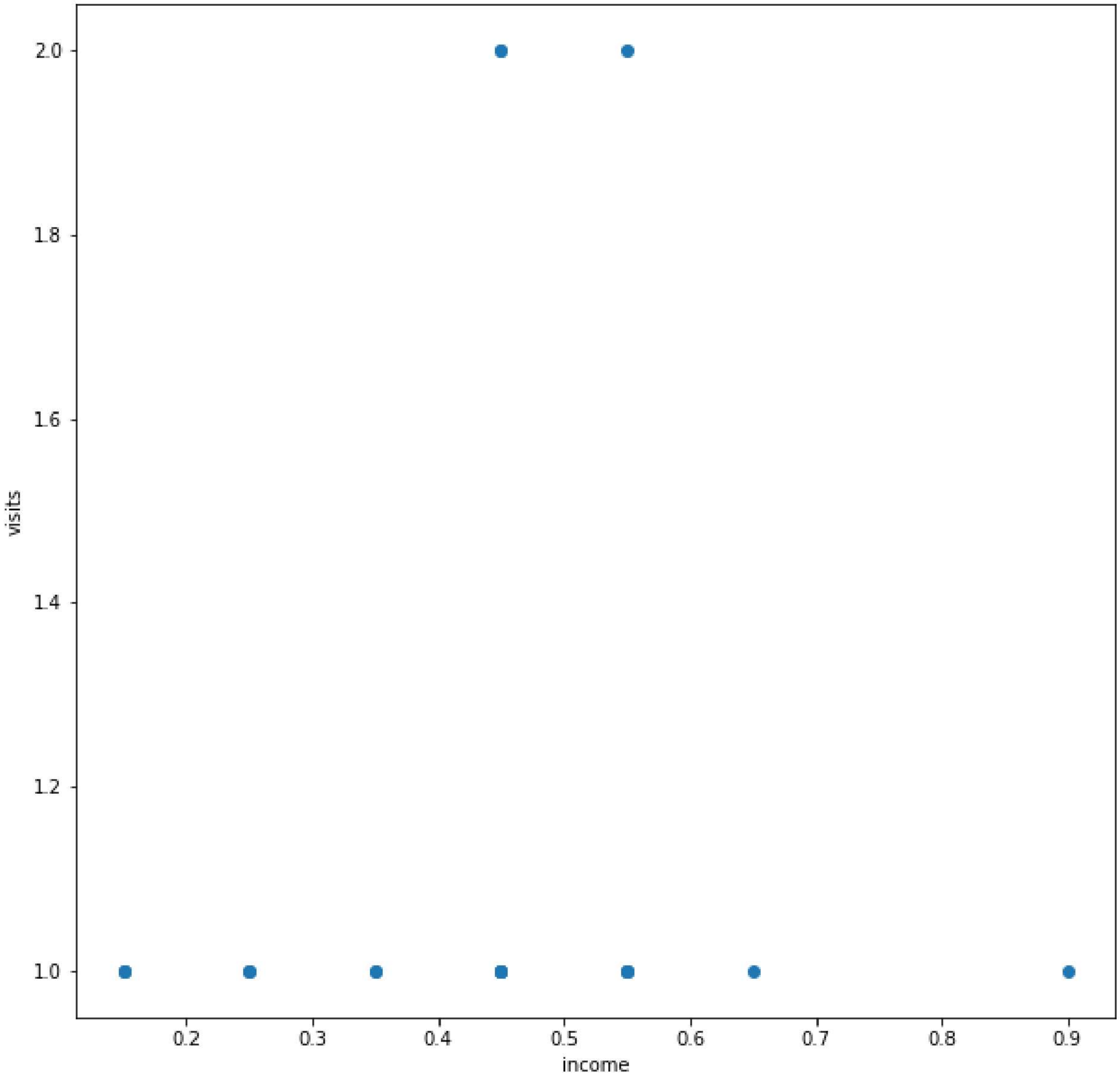
```
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),cbar=True,annot=True,cmap="Blues")
```

Out[59]:

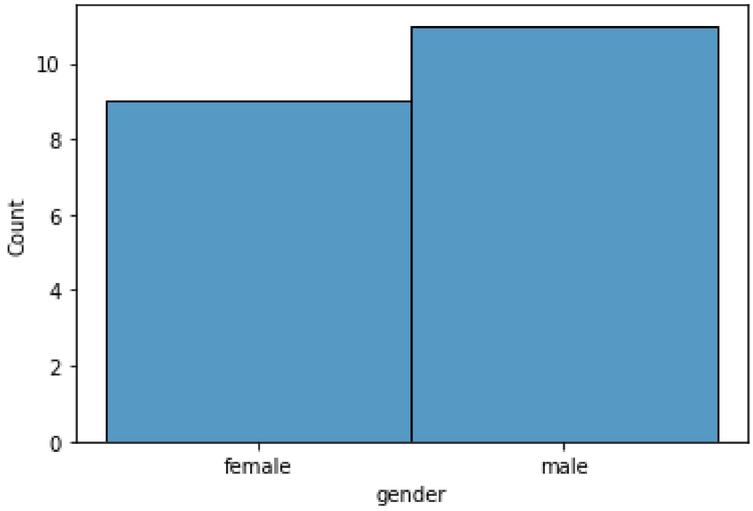
```
<AxesSubplot:>
```



```
In [60]: #relation between income and visits
         plt.figure(figsize=(10,10))
         plt.scatter(x='income',y='visits',data=df)
         plt.xlabel('income')
         plt.ylabel('visits')

Out[60]: Text(0, 0.5, 'visits')
```

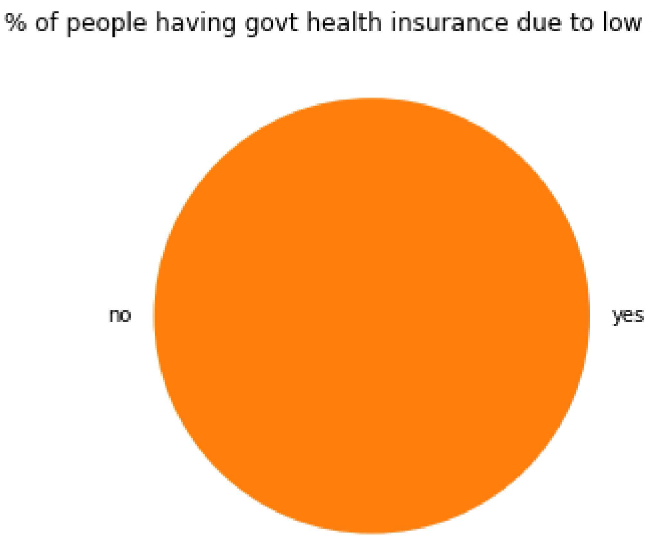


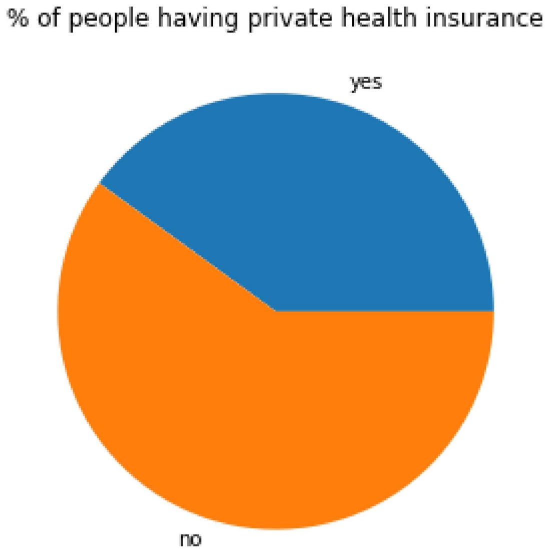
```
In [61]: #no of males and females affected by illness
         sns.histplot(df.gender,bins=2)

Out[61]: <AxesSubplot:xlabel='gender', ylabel='Count'>
```

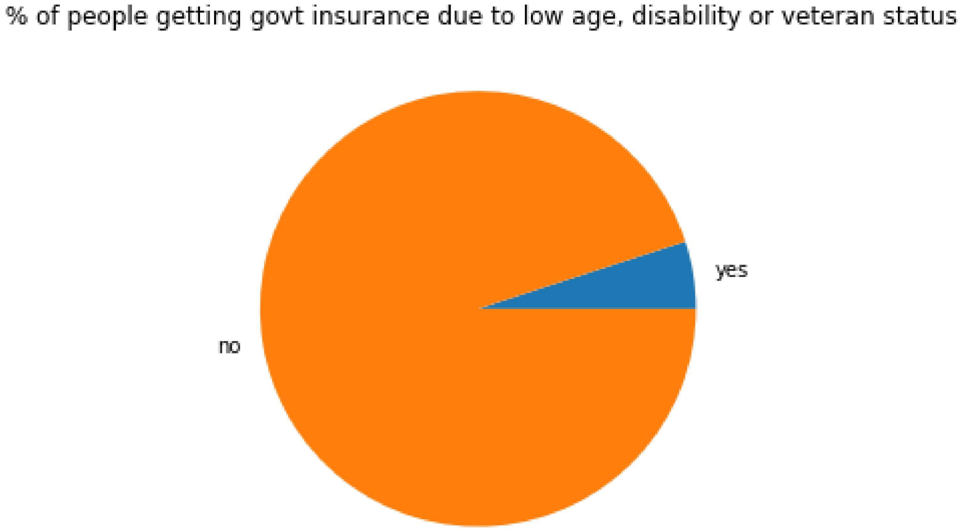


visualize the percentage of people getting govt health insurance due to low income, due to low age and also the percentage of people getting private health insurance

In [62]:
```
#percentage of people having govt 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=(5,5))
plt.pie(x,labels=label)
plt.title("% of people having govt health insurance due to low income")
plt.show()
#percentage of people having private health insurance
Y=df[df['private']=='yes']
N=df[df['private']=='no']
x=[Y.shape[0],N.shape[0]]
plt.figure(figsize=(5,5))
plt.pie(x,labels=label)
plt.title("% of people having private health insurance")
plt.show()
# % of people getting govt insurance due to low age, disability or veteran stat
Y=df[df['freerepat']=='yes']
N=df[df['freerepat']=='no']
x=[Y.shape[0],N.shape[0]]
plt.figure(figsize=(5,5))
plt.pie(x,labels=label)
plt.title(" % of people getting govt insurance due to low age, disability or ve
plt.show()
```

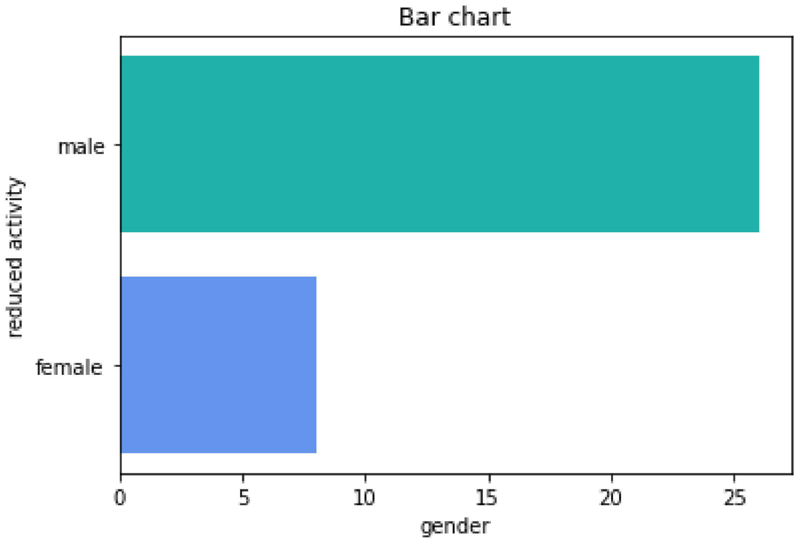






plot a horizontal bar chart to analyze the reduced days of activity due to illness based on gender

In [68]:
```
import matplotlib.pyplot as plt
db=df.groupby('gender')['reduced'].sum().to_frame().reset_index()
# creating the bar chart
plt.barh(db['gender'],db['reduced'],color=['cornflowerblue','lightseagreen'])
# Adding the aesthetics
plt.title('Bar chart')
plt.xlabel('gender')
plt.ylabel('reduced activity')
#show the plot
plt.show()
```


Bar chart
male
female
reduced activity
gender