In [1]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [24]:
```
dataset = pd.read_csv('Heart.csv')
```

In [25]:
```
dataset.head(10)
```

Out[25]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0
5	6	56	1	nontypical	120	236	0	0	178	0	0.8	1	0.0
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6	3	2.0
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6	1	0.0
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4	2	1.0
9	10	53	1	asymptomatic	140	203	1	2	155	1	3.1	3	0.0

In [26]:
```
dataset2 = pd.read_csv('heart.csv')
```

In [27]:
```
dataset2.head(10)
```

Out[27]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0
5	6	56	1	nontypical	120	236	0	0	178	0	0.8	1	0.0
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6	3	2.0
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6	1	0.0
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4	2	1.0
9	10	53	1	asymptomatic	140	203	1	2	155	1	3.1	3	0.0

In [28]:

```
dataset2.isna().sum()
```

Out[28]:

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

In [29]:

```
dataset2 = dataset2.dropna(axis=0)
```

In [30]:

```
dataset2.isnull().sum()
```

Out[30]:

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            0
Thal          0
AHD           0
dtype: int64
```

In [31]:

```
dataset2.describe()
```

Out[31]:

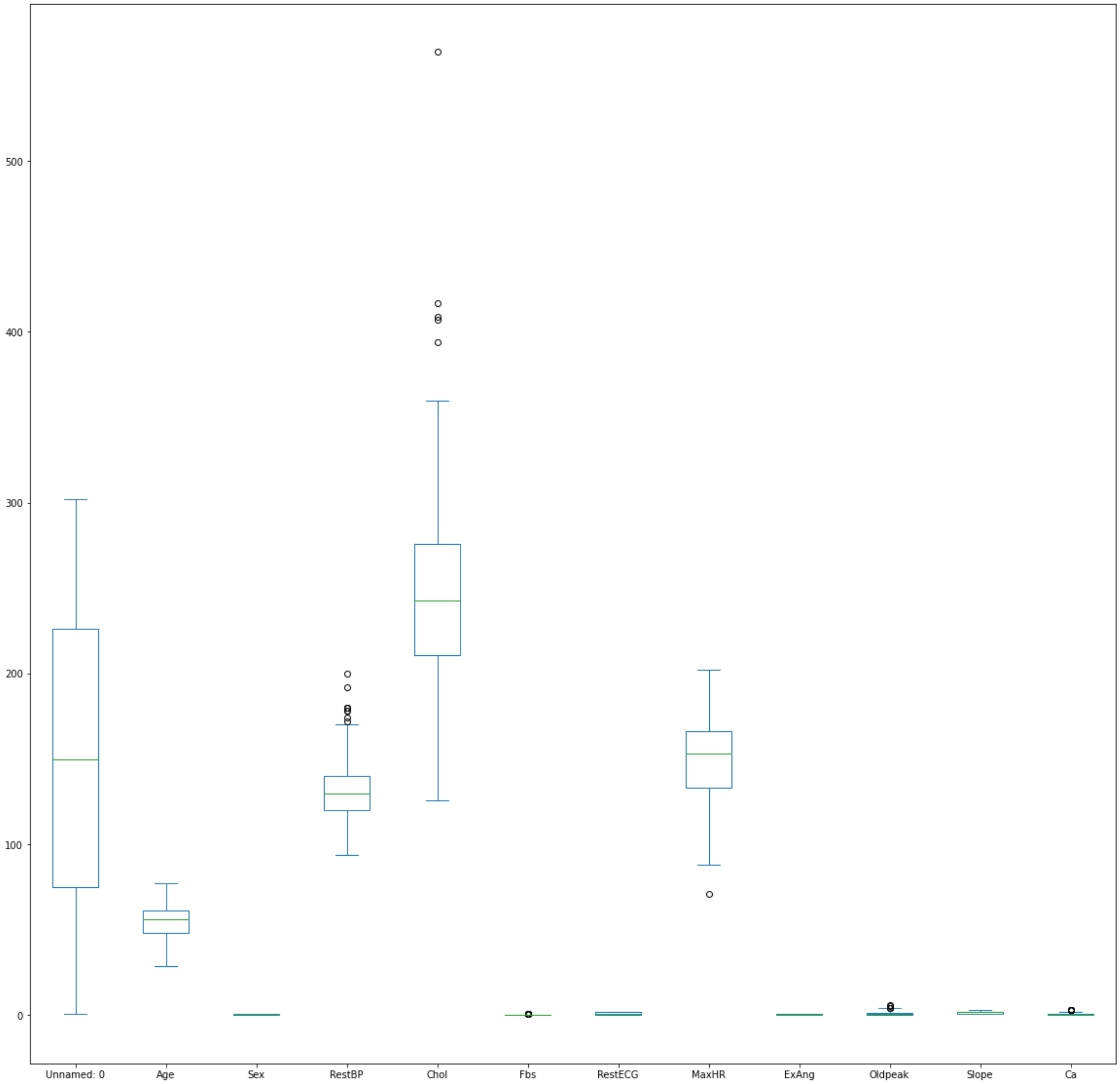
	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	MaxHR	
count	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	29
mean	150.673401	54.542088	0.676768	131.693603	247.350168	0.144781	0.996633	149.599327	
std	87.323283	9.049736	0.468500	17.762806	51.997583	0.352474	0.994914	22.941562	
min	1.000000	29.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	
25%	75.000000	48.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.000000	
50%	150.000000	56.000000	1.000000	130.000000	243.000000	0.000000	1.000000	153.000000	
75%	226.000000	61.000000	1.000000	140.000000	276.000000	0.000000	2.000000	166.000000	
max	302.000000	77.000000	1.000000	200.000000	564.000000	1.000000	2.000000	202.000000	

```
In [32]: dataset.count()
```

```
Out[32]: Unnamed: 0    303
         Age           303
         Sex           303
         ChestPain     303
         RestBP        303
         Chol          303
         Fbs           303
         RestECG       303
         MaxHR         303
         ExAng         303
         Oldpeak       303
         Slope         303
         Ca            299
         Thal          301
         AHD           303
         dtype: int64
```

```
In [33]: plt.figure(figsize=(9,3))
         dataset2.plot(kind='box',figsize=(20,20))
```

```
Out[33]: <AxesSubplot:>

         <Figure size 648x216 with 0 Axes>
```



```
In [34]: percentile25 = dataset2['Chol'].quantile(0.25)
         percentile75 = dataset2['Chol'].quantile(0.75)
```

```
In [35]: iqr = percentile75 - percentile25
```

```
In [36]: upper_limit = percentile75 + 1.5 * iqr
         lower_limit = percentile25 - 1.5 * iqr
```

In [37]:

```
dataset2[dataset2['Chol'] > upper_limit]
dataset2[dataset2['Chol'] < lower_limit]
```

Out[37]:

Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Th

In [38]:

```
dataset2['Chol'] = dataset2[dataset2['Chol'] < upper_limit]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_6648/3294842726.py in <module>
----> 1 dataset2['Chol'] = dataset2[dataset2['Chol'] < upper_limit]

~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py in __
setitem__(self, key, value)
   3600             self._setitem_array(key, value)
   3601         elif isinstance(value, DataFrame):
-> 3602             self._set_item_frame_value(key, value)
   3603         elif (
   3604             is_list_like(value)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py in _s
et_item_frame_value(self, key, value)
   3727             len_cols = 1 if is_scalar(cols) else len(cols)
   3728             if len_cols != len(value.columns):
-> 3729                 raise ValueError("Columns must be same length as key")
   3730 
   3731             # align right-hand-side columns if self.columns

ValueError: Columns must be same length as key
```

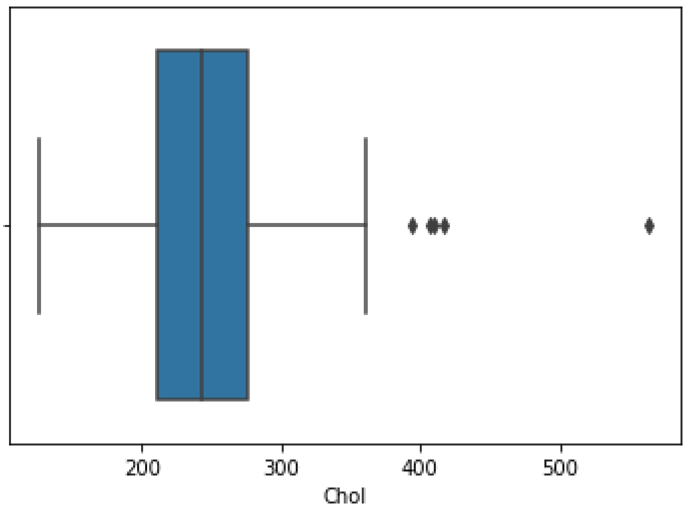
In [39]:

```
sns.boxplot(dataset2['Chol'])
```

```
C:\Users\rapen\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_deco
rators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ver
sion 0.12, the only valid positional argument will be `data`, and passing other argume
nts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[39]:

```
<AxesSubplot:xlabel='Chol'>
```



In [40]:
```
percentile25 = dataset2['RestBP'].quantile(0.25)
percentile75 = dataset2['RestBP'].quantile(0.75)
```

In [41]:
```
iqr = percentile75 - percentile25
```

In [42]:
```
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr
```

In [43]:
```
dataset2[dataset2['RestBP'] > upper_limit]
dataset2[dataset2['RestBP'] < lower_limit]
```

Out[43]:

Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Th

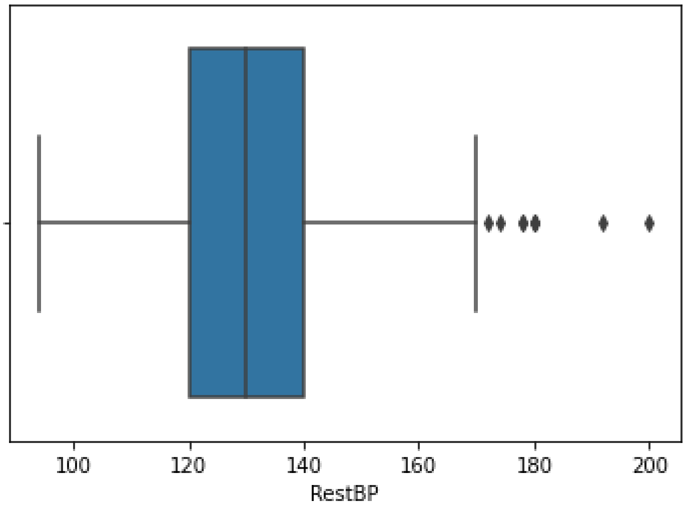
In [44]:
```
dataset2['RestBP'] = dataset2[dataset2['RestBP'] < upper_limit]
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_6648/2386516796.py in <module>
----> 1 dataset2['RestBP'] = dataset2[dataset2['RestBP'] < upper_limit]

~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py in __
setitem__(self, key, value)
   3600             self._setitem_array(key, value)
   3601         elif isinstance(value, DataFrame):
-> 3602             self._set_item_frame_value(key, value)
   3603         elif (
   3604             is_list_like(value)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py in _s
et_item_frame_value(self, key, value)
   3727             len_cols = 1 if is_scalar(cols) else len(cols)
   3728             if len_cols != len(value.columns):
-> 3729                 raise ValueError("Columns must be same length as key")
   3730 
   3731             # align right-hand-side columns if self.columns

ValueError: Columns must be same length as key
```

In [45]:
```
sns.boxplot(dataset2['RestBP'])
```

```
C:\Users\rapen\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_deco
rators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ver
sion 0.12, the only valid positional argument will be `data`, and passing other argume
nts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[45]:
```
<AxesSubplot:xlabel='RestBP'>
```



In [ ]:

In [46]:
```
print(dataset2.ChestPain.unique())
print(dataset2.Thal.unique())
print(dataset2.AHD.unique())
```

```
['typical' 'asymptomatic' 'nonanginal' 'nontypical']
['fixed' 'normal' 'reversable']
['No' 'Yes']
```

In [47]:

```
heart_encoding = pd.get_dummies(dataset2[['ChestPain', 'Thal', 'AHD']])
heart_final = pd.concat([dataset2, heart_encoding],1)
heart_final = heart_final.drop(['ChestPain', 'Thal', 'AHD'], axis = 1)
heart_final.head(10)
```

```
C:\Users\rapen\AppData\Local\Temp/ipykernel_6648/3224836008.py:2: FutureWarning: In a
future version of pandas all arguments of concat except for the argument 'objs' will b
e keyword-only
  heart_final = pd.concat([dataset2, heart_encoding],1)
```

Out[47]:

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	...	Ca	ChestPain_asym
0	1	63	1	145	233	1	2	150	0	2.3	...	0.0	
1	2	67	1	160	286	0	2	108	1	1.5	...	3.0	
2	3	67	1	120	229	0	2	129	1	2.6	...	2.0	
3	4	37	1	130	250	0	0	187	0	3.5	...	0.0	
4	5	41	0	130	204	0	2	172	0	1.4	...	0.0	
5	6	56	1	120	236	0	0	178	0	0.8	...	0.0	
6	7	62	0	140	268	0	2	160	0	3.6	...	2.0	
7	8	57	0	120	354	0	0	163	1	0.6	...	0.0	
8	9	63	1	130	254	0	2	147	0	1.4	...	1.0	
9	10	53	1	140	203	1	2	155	1	3.1	...	0.0	

10 rows × 21 columns

In [48]:

```
heart_final.Sex.value_counts()
```

Out[48]:

```
1    201
0     96
Name: Sex, dtype: int64
```

In [49]:

```
pd.crosstab(heart_final.AHD_Yes,heart_final.Sex)
```

Out[49]:

Sex	0	1
AHD_Yes		
0	71	89
1	25	112

In [50]:

```
heart_final.columns
```

Out[50]:

```
Index(['Unnamed: 0', 'Age', 'Sex', 'RestBP', 'Chol', 'Fbs', 'RestECG', 'MaxHR',
       'ExAng', 'Oldpeak', 'Slope', 'Ca', 'ChestPain_asymptomatic',
       'ChestPain_nonanginal', 'ChestPain_nontypical', 'ChestPain_typical',
       'Thal_fixed', 'Thal_normal', 'Thal_reversable', 'AHD_No', 'AHD_Yes'],
      dtype='object')
```

In [51]:
```
heart_final.dtypes
```

Out[51]:
```
Unnamed: 0                 int64
Age                        int64
Sex                        int64
RestBP                     int64
Chol                       int64
Fbs                        int64
RestECG                    int64
MaxHR                      int64
ExAng                      int64
Oldpeak                  float64
Slope                      int64
Ca                       float64
ChestPain_asymptomatic     uint8
ChestPain_nonanginal       uint8
ChestPain_nontypical       uint8
ChestPain_typical          uint8
Thal_fixed                 uint8
Thal_normal                uint8
Thal_reversable            uint8
AHD_No                     uint8
AHD_Yes                    uint8
dtype: object
```

In [52]:
```
df = heart_final.drop('AHD_Yes', axis=1)
df_norm = (df-df.min())/(df.max()-df.min())
df_norm = pd.concat((df_norm, heart_final.AHD_Yes), 1)
```

```
C:\Users\rapen\AppData\Local\Temp/ipykernel_6648/1126837166.py:3: FutureWarning: In a
future version of pandas all arguments of concat except for the argument 'objs' will b
e keyword-only
  df_norm = pd.concat((df_norm, heart_final.AHD_Yes), 1)
```

In [53]:
```
df_norm.head(10)
```

Out[53]:

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	...	Ca
0	0.000000	0.708333	1.0	0.481132	0.244292	1.0	1.0	0.603053	0.0	0.370968	...	0.000000
1	0.003322	0.791667	1.0	0.622642	0.365297	0.0	1.0	0.282443	1.0	0.241935	...	1.000000
2	0.006645	0.791667	1.0	0.245283	0.235160	0.0	1.0	0.442748	1.0	0.419355	...	0.666667
3	0.009967	0.166667	1.0	0.339623	0.283105	0.0	0.0	0.885496	0.0	0.564516	...	0.000000
4	0.013289	0.250000	0.0	0.339623	0.178082	0.0	1.0	0.770992	0.0	0.225806	...	0.000000
5	0.016611	0.562500	1.0	0.245283	0.251142	0.0	0.0	0.816794	0.0	0.129032	...	0.000000
6	0.019934	0.687500	0.0	0.433962	0.324201	0.0	1.0	0.679389	0.0	0.580645	...	0.666667
7	0.023256	0.583333	0.0	0.245283	0.520548	0.0	0.0	0.702290	1.0	0.096774	...	0.000000
8	0.026578	0.708333	1.0	0.339623	0.292237	0.0	1.0	0.580153	0.0	0.225806	...	0.333333
9	0.029900	0.500000	1.0	0.433962	0.175799	1.0	1.0	0.641221	1.0	0.500000	...	0.000000

10 rows × 21 columns

In [54]:
```
df_norm = df_norm.dropna()
```

In [55]:
```
X = df_norm.drop(['AHD_Yes', 'Unnamed: 0'], axis=1)
Y = df_norm.AHD_Yes
```

In [56]:
```
#data celeaning done
X.isnull().sum()
```

Out[56]:
```
Age                     0
Sex                     0
RestBP                  0
Chol                    0
Fbs                     0
RestECG                 0
MaxHR                   0
ExAng                   0
Oldpeak                 0
Slope                   0
Ca                      0
ChestPain_asymptomatic  0
ChestPain_nonanginal    0
ChestPain_nontypical    0
ChestPain_typical       0
Thal_fixed              0
Thal_normal             0
Thal_reversable         0
AHD_No                  0
dtype: int64
```

In [ ]:
```

```

In [57]:
```
Y.isnull().sum()
```

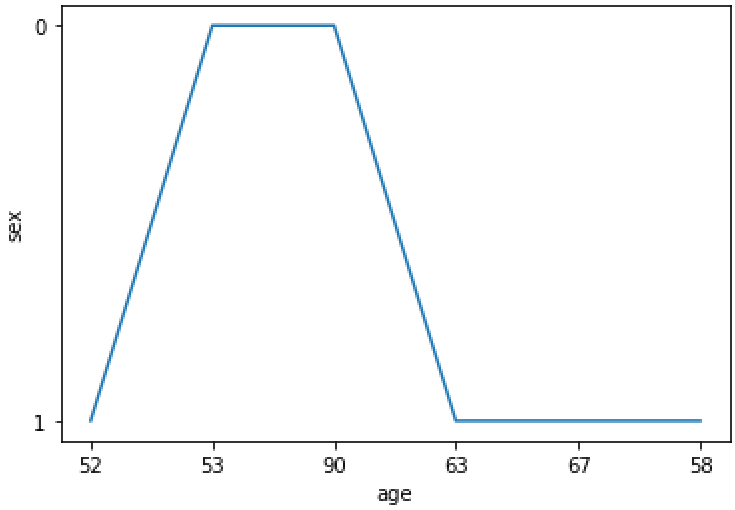
Out[57]:
```
0
```

In [58]:
```
df_norm.isnull().sum()
```

Out[58]:
```
Unnamed: 0              0
Age                     0
Sex                     0
RestBP                  0
Chol                    0
Fbs                     0
RestECG                 0
MaxHR                   0
ExAng                   0
Oldpeak                 0
Slope                   0
Ca                      0
ChestPain_asymptomatic  0
ChestPain_nonanginal    0
ChestPain_nontypical    0
ChestPain_typical       0
Thal_fixed              0
Thal_normal             0
Thal_reversable         0
AHD_No                  0
AHD_Yes                 0
dtype: int64
```

In [59]:

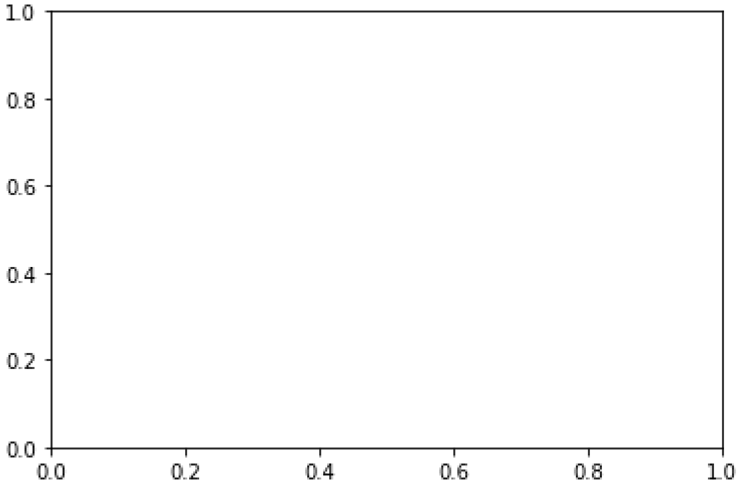
```
# d] line charts
age=["52","53","90","63","67","58"]
sex=['1','0','0','1','1','1']
plt.plot(age,sex)
plt.xlabel("age")
plt.ylabel("sex")
x = np.array([52, 53, 90, 63,67])  # X-axis points
y = x*2  # Y-axis points

plt.show()
```



In [61]:

```
#b]histogram
plt.hist(dataset['Age'],dataset['Sex'])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_6648/1216958322.py in <module>
      1 #b]histogram
----> 2 plt.hist(dataset['Age'],dataset['Sex'])

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\pyplot.py in hi
st(x, bins, range, density, weights, cumulative, bottom, histtype, align, orientation,
rwidth, log, color, label, stacked, data, **kwargs)
   2588         orientation='vertical', rwidth=None, log=False, color=None,
   2589         label=None, stacked=False, *, data=None, **kwargs):
-> 2590     return gca().hist(
   2591         x, bins=bins, range=range, density=density, weights=weights,
   2592         cumulative=cumulative, bottom=bottom, histtype=histtype,

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\__init__.py in
inner(ax, data, *args, **kwargs)
   1410     def inner(ax, *args, data=None, **kwargs):
   1411         if data is None:
-> 1412             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1413 
   1414         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\matplotlib\axes\_axes.py i
n hist(self, x, bins, range, density, weights, cumulative, bottom, histtype, align, or
ientation, rwidth, log, color, label, stacked, **kwargs)
   6637             # this will automatically overwrite bins,
   6638             # so that each histogram uses the same bins
-> 6639             m, bins = np.histogram(x[i], bins, weights=w[i], **hist_kwargs)
   6640             tops.append(m)
   6641         tops = np.array(tops, float)  # causes problems later if it's an int

<__array_function__ internals> in histogram(*args, **kwargs)

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\histograms.py in
histogram(a, bins, range, normed, weights, density)
    791     a, weights = _ravel_and_check_weights(a, weights)
    792 
--> 793     bin_edges, uniform_bins = _get_bin_edges(a, bins, range, weights)
    794 
    795     # Histogram is an integer or a float array depending on the weights.

~\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\lib\histograms.py in
_get_bin_edges(a, bins, range, weights)
    429         bin_edges = np.asarray(bins)
    430         if np.any(bin_edges[:-1] > bin_edges[1:]):
--> 431             raise ValueError(
    432                 '`bins` must increase monotonically, when an array')
    433 

ValueError: `bins` must increase monotonically, when an array
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0