In [98]:
```
import pandas as pa
```

In [99]:
```
import numpy as np
```

In [100…
```
c=pa.read_csv(r"C:\Users\CVR\Downloads\clevelanda.csv")
```

In [101…
```
c.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   gender    303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fps       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    object 
 12  thal      303 non-null    object 
 13  class     303 non-null    int64  
dtypes: float64(1), int64(11), object(2)
memory usage: 33.3+ KB
```

In [102…
```
c.describe
```

Out[102]:
```
<bound method NDFrame.describe of      age  gender  cp  trestbps  chol  fps  reste
cg  thalach  exang  oldpeak  \
0     63       1   1       145   233    1        2      150      0      2.3
1     67       1   4       160   286    0        2      108      1      1.5
2     67       1   4       120   229    0        2      129      1      2.6
3     37       1   3       130   250    0        0      187      0      3.5
4     41       0   2       130   204    0        2      172      0      1.4
..   ...     ...  ..       ...   ...  ...      ...      ...    ...      ...
298   45       1   1       110   264    0        0      132      0      1.2
299   68       1   4       144   193    1        0      141      0      3.4
300   57       1   4       130   131    0        0      115      1      1.2
301   57       0   2       130   236    0        2      174      0      0.0
302   38       1   3       138   175    0        0      173      0      0.0

     slope ca thal  class
0        3  0    6      0
1        2  3    3      2
2        2  2    7      1
3        3  0    3      0
4        1  0    3      0
..     ... ..  ...    ...
298      2  0    7      1
299      2  2    7      2
300      2  1    7      3
301      2  1    3      1
302      1  ?    3      0

[303 rows x 14 columns]>
```

In [103…
```
c.columns
```

Out[103]:
```
Index(['age', 'gender', 'cp', 'trestbps', 'chol', 'fps', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'class'],
      dtype='object')
```

In [104…
```
c.head()
```

Out[104]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	class
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

In [105…
```
c.tail()
```

Out[105]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	cla
298	45	1	1	110	264	0	0	132	0	1.2	2	0	7	
299	68	1	4	144	193	1	0	141	0	3.4	2	2	7	
300	57	1	4	130	131	0	0	115	1	1.2	2	1	7	
301	57	0	2	130	236	0	2	174	0	0.0	2	1	3	
302	38	1	3	138	175	0	0	173	0	0.0	1	?	3	

In [106…
```
c
```

Out[106]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	cla
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	45	1	1	110	264	0	0	132	0	1.2	2	0	7	
299	68	1	4	144	193	1	0	141	0	3.4	2	2	7	
300	57	1	4	130	131	0	0	115	1	1.2	2	1	7	
301	57	0	2	130	236	0	2	174	0	0.0	2	1	3	
302	38	1	3	138	175	0	0	173	0	0.0	1	?	3	

303 rows × 14 columns

In [107…
```
c.shape
```

Out[107]:
```
(303, 14)
```

In [108…
```
c.columns
```

Out[108]:
```
Index(['age', 'gender', 'cp', 'trestbps', 'chol', 'fps', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'class'],
      dtype='object')
```

In [109…
```
c.isnull()
```

Out[109]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	t
0	False	False	False	False	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	False	False	Fa
…	…	…	…	…	…	…	…	…	…	…	…	…	
298	False	False	False	False	False	False	False	False	False	False	False	False	Fa
299	False	False	False	False	False	False	False	False	False	False	False	False	Fa
300	False	False	False	False	False	False	False	False	False	False	False	False	Fa
301	False	False	False	False	False	False	False	False	False	False	False	False	Fa
302	False	False	False	False	False	False	False	False	False	False	False	False	Fa

303 rows × 14 columns

In [110…
```
c["age"].unique()
```

Out[110]:
```
array([63, 67, 37, 41, 56, 62, 57, 53, 44, 52, 48, 54, 49, 64, 58, 60, 50,
       66, 43, 40, 69, 59, 42, 55, 61, 65, 71, 51, 46, 45, 39, 68, 47, 34,
       35, 29, 70, 77, 38, 74, 76], dtype=int64)
```

In [111…
```
c["ca"].unique()
```

Out[111]:
```
array(['0', '3', '2', '1', '?'], dtype=object)
```

In [112…
```
c["class"].unique()
```

Out[112]:
```
array([0, 2, 1, 3, 4], dtype=int64)
```

In [113…
```
c["thal"].unique()
```

Out[113]:
```
array(['6', '3', '7', '?'], dtype=object)
```

In [114…
```
col_qm=[col for col in c.columns if c[col].astype(str).str.contains('?',regex=False
col_qm



#to check the columns with ? as ther evalues
```

Out[114]:
```
['ca', 'thal']
```

In [115…
```
c["ca"].replace('?',np.nan)


#to replace ? with nan values
```

Out[115]:
```
0        0
1        3
2        2
3        0
4        0
      ...
298      0
299      2
300      1
301      1
302    NaN
Name: ca, Length: 303, dtype: object
```

In [116…
```
c["ca"].replace('?',np.nan,inplace=True)


#to replace ? with nan values in exisiting dataset
```

In [117…
```
c
```

Out[117]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	c
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	
298	45	1	1	110	264	0	0	132	0	1.2	2	0	7	
299	68	1	4	144	193	1	0	141	0	3.4	2	2	7	
300	57	1	4	130	131	0	0	115	1	1.2	2	1	7	
301	57	0	2	130	236	0	2	174	0	0.0	2	1	3	
302	38	1	3	138	175	0	0	173	0	0.0	1	NaN	3	

303 rows × 14 columns

In [118…
```
c["thal"].replace('?',np.nan,inplace=True)
```

In [119…
```
c
```

Out[119]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	c
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	45	1	1	110	264	0	0	132	0	1.2	2	0	7	
299	68	1	4	144	193	1	0	141	0	3.4	2	2	7	
300	57	1	4	130	131	0	0	115	1	1.2	2	1	7	
301	57	0	2	130	236	0	2	174	0	0.0	2	1	3	
302	38	1	3	138	175	0	0	173	0	0.0	1	NaN	3	

303 rows × 14 columns

In [120…
```
c["thal"].unique(),c["ca"].unique()


#checking if its being modified to nan
```

Out[120]:
```
(array(['6', '3', '7', nan], dtype=object),
 array(['0', '3', '2', '1', nan], dtype=object))
```

In [121…
```
c.fillna(0,inplace=True)

#fillimg 0 in place of nan to convet the dtype of columns from object to int so tha
```

In [122…
```
c["thal"].unique(),c["ca"].unique()
```

Out[122]:
```
(array(['6', '3', '7', 0], dtype=object),
 array(['0', '3', '2', '1', 0], dtype=object))
```

In [144…
```
c['ca']=c['ca'].astype(int)

#changing the dtype to column to int
```

In [145…
```
c['thal']=c['thal'].astype(int)
```

In [146…
```
c["ca"].mean()
```

Out[146]:
```
0.6633663366336634
```

In [147…
```
c["ca"].mode()
```

Out[147]:
```
0    0
Name: ca, dtype: int32
```

In [148…
```
c["ca"].median()
```

Out[148]:
```
0.0
```

In [149…
```
c["thal"].mean()
```

Out[149]:
```
4.729372937293729
```

In [150…
```
c["thal"].mode()
```

Out[150]:
```
0    3
Name: thal, dtype: int32
```

In [151…
```
c["thal"].median()
```

Out[151]:
```
3.0
```

In [131…
```
#c['ca']=c['ca'].replace(0,0.6633663366336634)

#replacing the 0 values in column to the mean value of the respective column
```

In [135…
```
#c['thal']=c['thal'].replace(0,4.702970297029703)
```

In [136…
```
#c["ca"].mean(),c["thal"].mean()

#chceking if data is being affted or is consistent
#as its effecting lets replace it with mode
```

Out[136]:
```
(1.0574453484952455, 4.7340130052609215)
```

In [138…
```
#c['ca']=c['ca'].replace(0,0)
```

In [152…
```
c['thal']=c['thal'].replace(0,3)

#replacing the 0 values in column to the mode value of the respective column
```

In [153…
```
c["ca"].mode(),c["thal"].mode()

#chceking if data is being affted or is consistent
```

Out[153]:
```
(0    0
 Name: ca, dtype: int32,
 0    3
 Name: thal, dtype: int32)
```

In [154…
```
c
```

Out[154]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	cla
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	45	1	1	110	264	0	0	132	0	1.2	2	0	7	
299	68	1	4	144	193	1	0	141	0	3.4	2	2	7	
300	57	1	4	130	131	0	0	115	1	1.2	2	1	7	
301	57	0	2	130	236	0	2	174	0	0.0	2	1	3	
302	38	1	3	138	175	0	0	173	0	0.0	1	0	3	

303 rows × 14 columns

In [155…
```
c["ca"].mean(),c["ca"].mode(),c["ca"].median()
```

Out[155]:
```
(0.6633663366336634,
 0    0
 Name: ca, dtype: int32,
 0.0)
```

In [157…
```
c["thal"].mean(),c["thal"].mode(),c["thal"].median()
```

Out[157]:
```
(4.729372937293729,
 0    3
 Name: thal, dtype: int32,
 3.0)
```

the mean mode median values of both columns are unchanged when replaced with mode so in this case i.e in this data set replacing with mode is the best fit

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [33]:
```
c['ca'].value_counts()
```

Out[33]:
```
ca
0    176
1     65
2     38
3     20
?      4
Name: count, dtype: int64
```

In [34]:
```
c['class'].value_counts()
```

Out[34]:
```
class
0    164
1     55
2     36
3     35
4     13
Name: count, dtype: int64
```

In [ ]:

In [14]:
```
(c.isnull()).sum()
```

Out[14]:
```
age         0
gender      0
cp          0
trestbps    0
chol        0
fps         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
class       0
dtype: int64
```

In [15]:
```
dups=c[c.duplicated()]
```

In [16]:
```
dups
```

Out[16]:

	age	gender	cp	trestbps	chol	fps	restecg	thalach	exang	oldpeak	slope	ca	thal	class

In [17]:
```
c=c.drop_duplicates()
```

In [18]:
```
c["fps"].mean()
```

Out[18]:
```
0.1485148514851485
```

In [19]:
```
c["fps"].median()
```

Out[19]:
```
0.0
```

In [20]:
```
c["fps"].mode()
```

Out[20]:
```
0    0
Name: fps, dtype: int64
```

In [22]:
```
c["fps"].fillna(0.1485148514851485)
```

Out[22]:
```
0      1
1      0
2      0
3      0
4      0
      ..
298    0
299    1
300    0
301    0
302    0
Name: fps, Length: 303, dtype: int64
```

In [36]:
```
pip install plotly-express
```

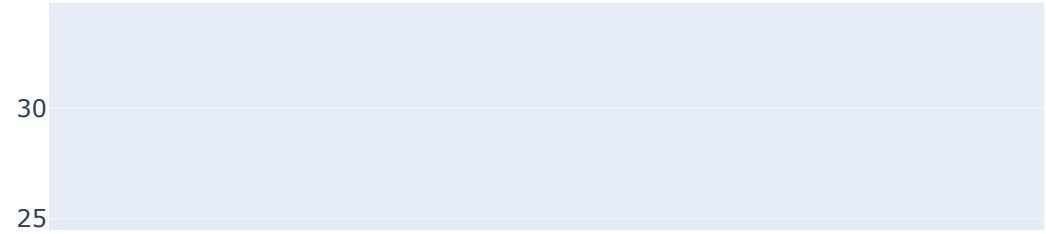
```
Collecting plotly-expressNote: you may need to restart the kernel to use updated p
ackages.

  Obtaining dependency information for plotly-express from https://files.pythonhos
ted.org/packages/d4/d6/8a2906f51e073a4be80cab35cfa10e7a34853e60f3ed5304ac470852a08
d/plotly_express-0.4.1-py2.py3-none-any.whl.metadata
  Downloading plotly_express-0.4.1-py2.py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: pandas>=0.20.0 in c:\users\cvr\anaconda3\lib\site-p
ackages (from plotly-express) (2.0.3)
Requirement already satisfied: plotly>=4.1.0 in c:\users\cvr\anaconda3\lib\site-pa
ckages (from plotly-express) (5.9.0)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\cvr\anaconda3\lib\si
te-packages (from plotly-express) (0.14.0)
Requirement already satisfied: scipy>=0.18 in c:\users\cvr\anaconda3\lib\site-pack
ages (from plotly-express) (1.11.1)
Requirement already satisfied: patsy>=0.5 in c:\users\cvr\anaconda3\lib\site-packa
ges (from plotly-express) (0.5.3)
Requirement already satisfied: numpy>=1.11 in c:\users\cvr\anaconda3\lib\site-pack
ages (from plotly-express) (1.24.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\cvr\anaconda3\li
b\site-packages (from pandas>=0.20.0->plotly-express) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\cvr\anaconda3\lib\site-pac
kages (from pandas>=0.20.0->plotly-express) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\cvr\anaconda3\lib\site-p
ackages (from pandas>=0.20.0->plotly-express) (2023.3)
Requirement already satisfied: six in c:\users\cvr\anaconda3\lib\site-packages (fr
om patsy>=0.5->plotly-express) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\cvr\anaconda3\lib\site-
packages (from plotly>=4.1.0->plotly-express) (8.2.2)
Requirement already satisfied: packaging>=21.3 in c:\users\cvr\anaconda3\lib\site-
packages (from statsmodels>=0.9.0->plotly-express) (23.1)
Downloading plotly_express-0.4.1-py2.py3-none-any.whl (2.9 kB)
Installing collected packages: plotly-express
Successfully installed plotly-express-0.4.1
```

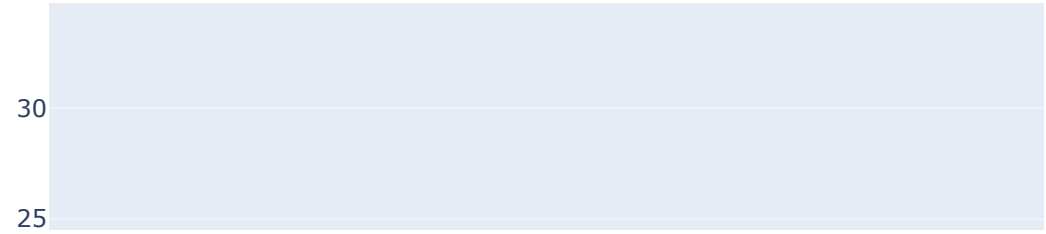
In [37]:
```
import plotly.express as px
```

In [38]:
```
h=px.histogram(c,x="age",nbins=25)
```

In [39]:
```
h
```

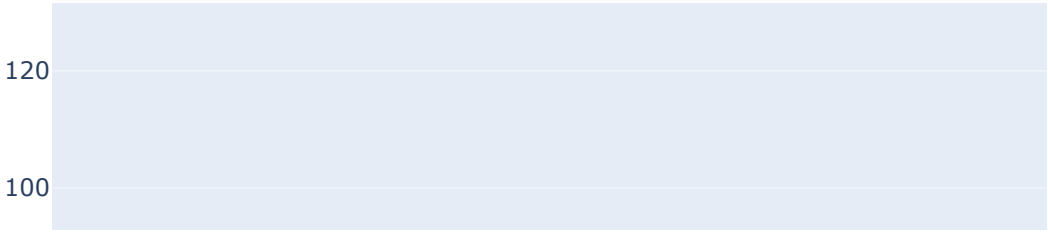


In [40]:
```
h.show()
```



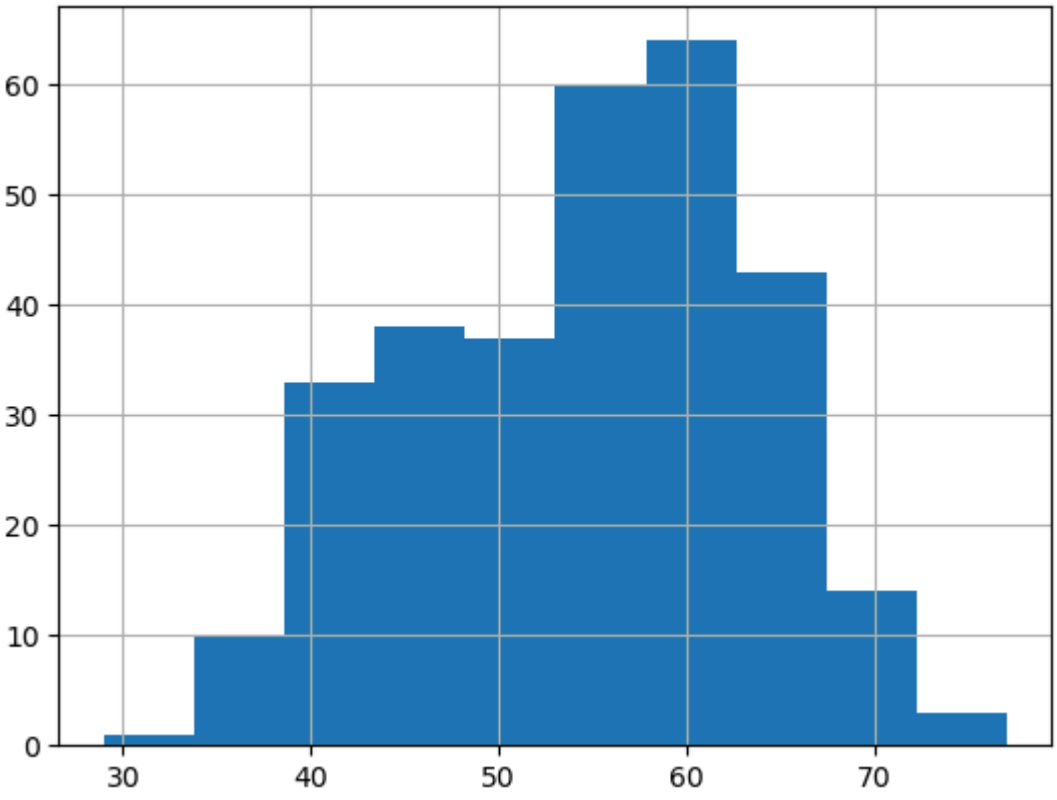
In [41]:
```
h=px.histogram(c,x="age",nbins=5)
```

In [42]:
```
h.show()
```



In [43]:
```
c["age"].hist()
```

Out[43]: <Axes: >



In [ ]:

In [ ]:

In [ ]: