

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

df = pd.read_csv('heart.csv')

-----
-----
FileNotFoundError                         Traceback (most recent call
last)
Cell In[4], line 1
----> 1 df = pd.read_csv('heart.csv')

File ~\AppData\Local\Programs\Python\Python39\lib\site-packages\
pandas\io\parsers\readers.py:912, in read_csv(filepath_or_buffer, sep,
delimiter, header, names, index_col, usecols, dtype, engine,
converters, true_values, false_values, skipinitialspace, skiprows,
skipfooter, nrows, na_values, keep_default_na, na_filter, verbose,
skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col,
date_parser, date_format, dayfirst, cache_dates, iterator, chunksize,
compression, thousands, decimal, lineterminator, quotechar, quoting,
doublequote, escapechar, comment, encoding, encoding_errors, dialect,
on_bad_lines, delim_whitespace, low_memory, memory_map,
float_precision, storage_options, dtype_backend)
    899 kwds_defaults = _refine_defaults_read(
    900     dialect,
    901     delimiter,
    (...),
    908     dtype_backend=dtype_backend,
    909 )
    910 kwds.update(kwds_defaults)
--> 912 return _read(filepath_or_buffer, kwds)

File ~\AppData\Local\Programs\Python\Python39\lib\site-packages\
pandas\io\parsers\readers.py:577, in _read(filepath_or_buffer, kwds)
    574 _validate_names(kwds.get("names", None))
    576 # Create the parser.
--> 577 parser = TextFileReader(filepath_or_buffer, **kwds)
    579 if chunksize or iterator:
    580     return parser

File ~\AppData\Local\Programs\Python\Python39\lib\site-packages\
pandas\io\parsers\readers.py:1407, in TextFileReader.__init__(self, f,
engine, **kwds)
    1404     self.options["has_index_names"] = kwds["has_index_names"]
    1406 self.handles: I0Handles | None = None
-> 1407 self._engine = self._make_engine(f, self.engine)

File ~\AppData\Local\Programs\Python\Python39\lib\site-packages\
pandas\io\parsers\readers.py:1661, in
```

```
TextFileReader._make_engine(self, f, engine)
1659     if "b" not in mode:
1660         mode += "b"
-> 1661 self.handles = get_handle(
1662     f,
1663     mode,
1664     encoding=self.options.get("encoding", None),
1665     compression=self.options.get("compression", None),
1666     memory_map=self.options.get("memory_map", False),
1667     is_text=is_text,
1668     errors=self.options.get("encoding_errors", "strict"),
1669     storage_options=self.options.get("storage_options", None),
1670 )
1671 assert self.handles is not None
1672 f = self.handles.handle
```

```
File ~\AppData\Local\Programs\Python\Python39\lib\site-packages\
pandas\io\common.py:859, in get_handle(path_or_buf, mode, encoding,
compression, memory_map, is_text, errors, storage_options)
    854     elif isinstance(handle, str):
    855         # Check whether the filename is to be opened in binary
mode.
    856         # Binary mode does not support 'encoding' and 'newline'
    857         if ioargs.encoding and "b" not in ioargs.mode:
    858             # Encoding
--> 859             handle = open(
    860                 handle,
    861                 ioargs.mode,
    862                 encoding=ioargs.encoding,
    863                 errors=errors,
    864                 newline="",
    865             )
    866         else:
    867             # Binary mode
    868             handle = open(handle, ioargs.mode)
```

```
FileNotFoundException: [Errno 2] No such file or directory: 'heart.csv'
```

df

4	57	0	0	120	354	0	1	163	1
0.6									
..	..	..	..	..	..	..	..	..	..
..									
298	57	0	0	140	241	0	1	123	1
0.2									
299	45	1	3	110	264	0	1	132	0
1.2									
300	68	1	0	144	193	1	1	141	0
3.4									
301	57	1	0	130	131	0	1	115	1
1.2									
302	57	0	1	130	236	0	0	174	0
0.0									

	slope	ca	thal	target
0	0	0	1	1
1	0	0	2	1
2	2	0	2	1
3	2	0	2	1
4	2	0	2	1
..	..	..	..	..
298	1	0	3	0
299	1	0	3	0
300	1	2	3	0
301	1	1	3	0
302	1	1	2	0

[303 rows x 14 columns]

df.dtypes

age	int64
sex	int64
cp	int64
trestbps	int64
chol	int64
fbs	int64
restecg	int64
thalach	int64
exang	int64
oldpeak	float64
slope	int64
ca	int64
thal	int64
target	int64
dtype:	object

```
# df['sex']=df['sex'].astype('object')
df['sex']=df['sex'].astype('int64')
```

```

df.dtypes

age          int64
sex          object
cp           int64
trestbps    int64
chol         int64
fbs          int64
restecg     int64
thalach     int64
exang        int64
oldpeak     float64
slope        int64
ca           int64
thal         int64
target       int64
dtype: object

df.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   sex         303 non-null    object  
 2   cp          303 non-null    int64  
 3   trestbps   303 non-null    int64  
 4   chol         303 non-null    int64  
 5   fbs          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    int64  
 12  thal         303 non-null    int64  
 13  target       303 non-null    int64  
dtypes: float64(1), int64(12), object(1)
memory usage: 33.3+ KB

df.describe()

      age          cp      trestbps        chol        fbs
restecg \
count  303.000000  303.000000  303.000000  303.000000  303.000000
303.000000
mean   54.366337   0.966997  131.623762  246.264026   0.148515
0.528053

```

```

std      9.082101    1.032052   17.538143   51.830751    0.356198
0.525860
min     29.000000    0.000000   94.000000   126.000000   0.000000
0.000000
25%    47.500000    0.000000  120.000000  211.000000   0.000000
0.000000
50%    55.000000    1.000000  130.000000  240.000000   0.000000
1.000000
75%    61.000000    2.000000  140.000000  274.500000   0.000000
1.000000
max    77.000000    3.000000  200.000000  564.000000   1.000000
2.000000

          thalach      exang      oldpeak      slope      ca
thal \
count  303.000000  303.000000  303.000000  303.000000  303.000000
303.000000
mean   149.646865  0.326733   1.039604   1.399340   0.729373
2.313531
std    22.905161   0.469794   1.161075   0.616226   1.022606
0.612277
min    71.000000   0.000000   0.000000   0.000000   0.000000
0.000000
25%   133.500000   0.000000   0.000000   1.000000   0.000000
2.000000
50%   153.000000   0.000000   0.800000   1.000000   0.000000
2.000000
75%   166.000000   1.000000   1.600000   2.000000   1.000000
3.000000
max   202.000000   1.000000   6.200000   2.000000   4.000000
3.000000

          target
count  303.000000
mean   0.544554
std    0.498835
min    0.000000
25%   0.000000
50%   1.000000
75%   1.000000
max   1.000000

df['sex'] = df.sex.replace({1:'Male',0:'Female'})

df['target'] = df.target.replace({1:'Disease',0:'No_disease'})

sns.heatmap(df.corr(),annot=True,linewidths=2)
plt.tight_layout()

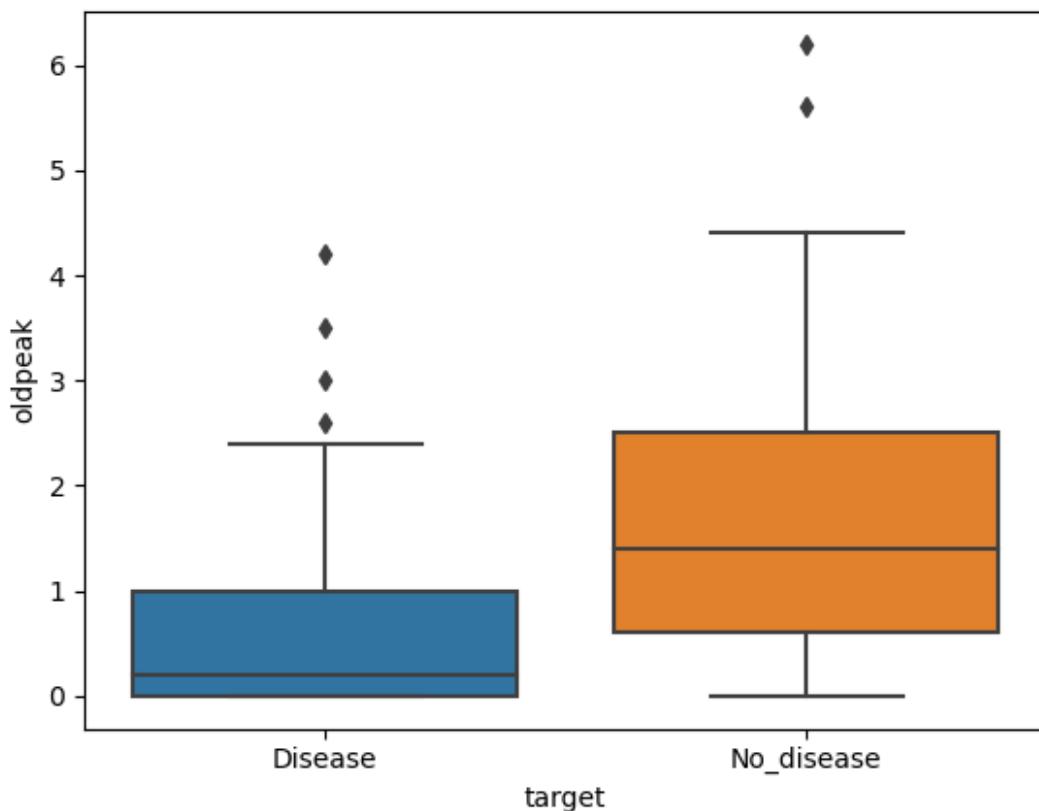
```

```
-----  
NameError Traceback (most recent call  
last)  
Cell In[1], line 1  
----> 1 sns.heatmap(df.corr(), annot=True, linewidths=2)  
      2 plt.tight_layout()  
-----
```

```
NameError: name 'sns' is not defined
```

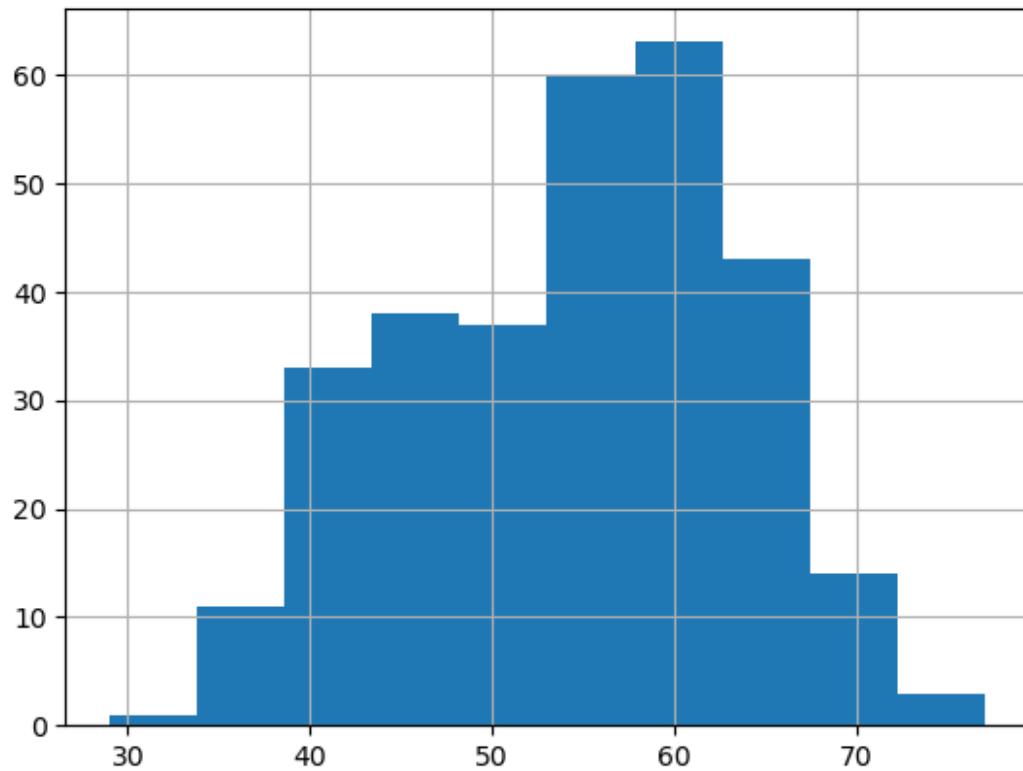
```
sns.boxplot(x='target', y='oldpeak', data=df)
```

```
<AxesSubplot:xlabel='target', ylabel='oldpeak'>
```

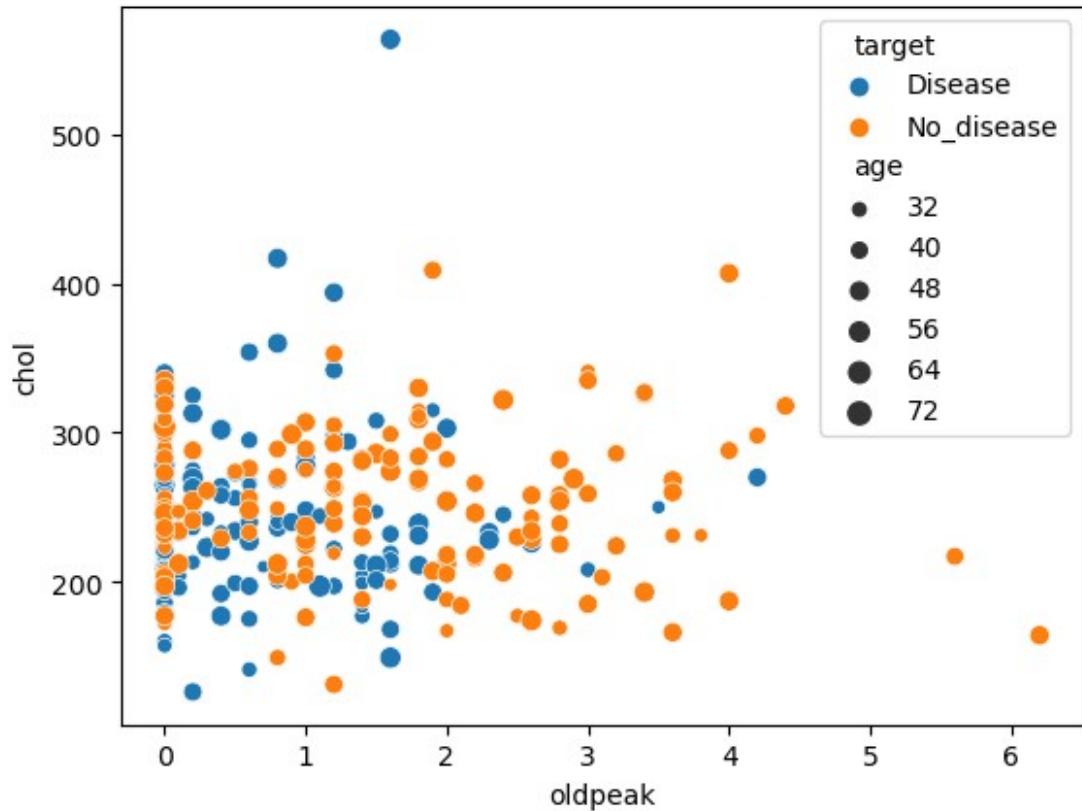


```
df['age'].hist().plot(kind='bar')
```

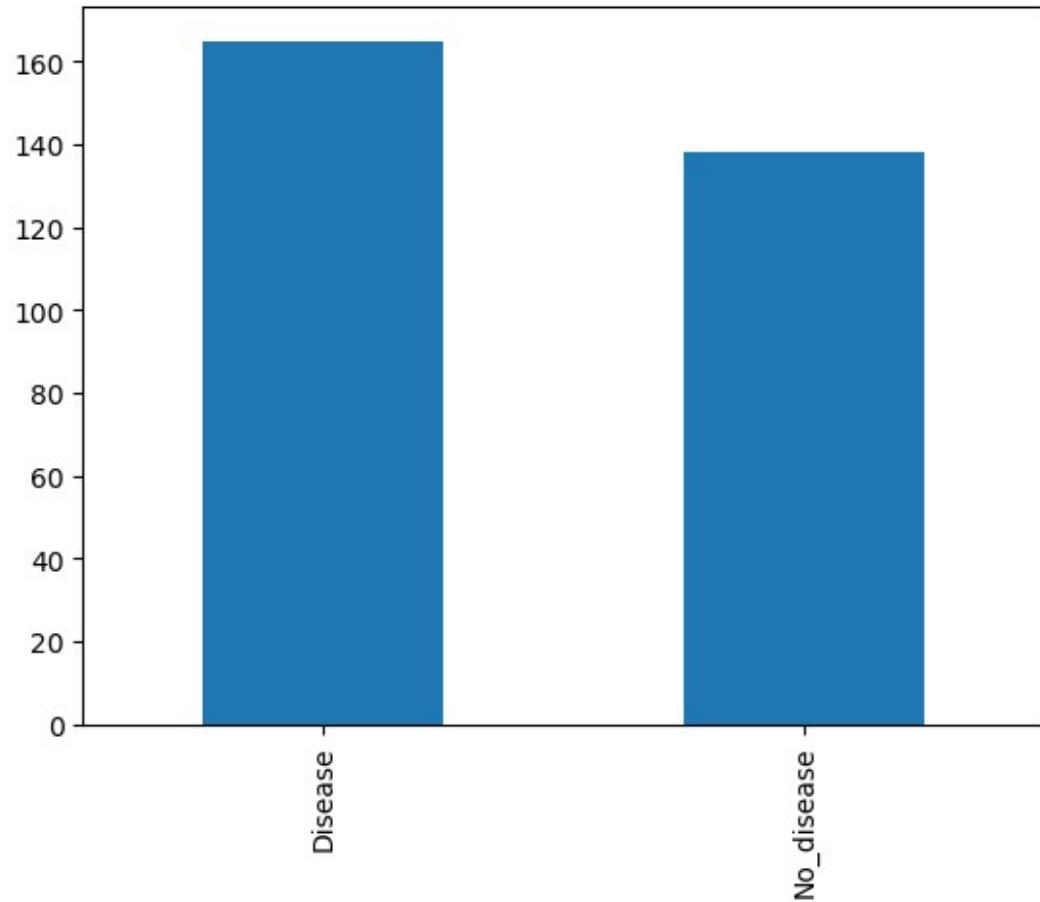
```
[]
```



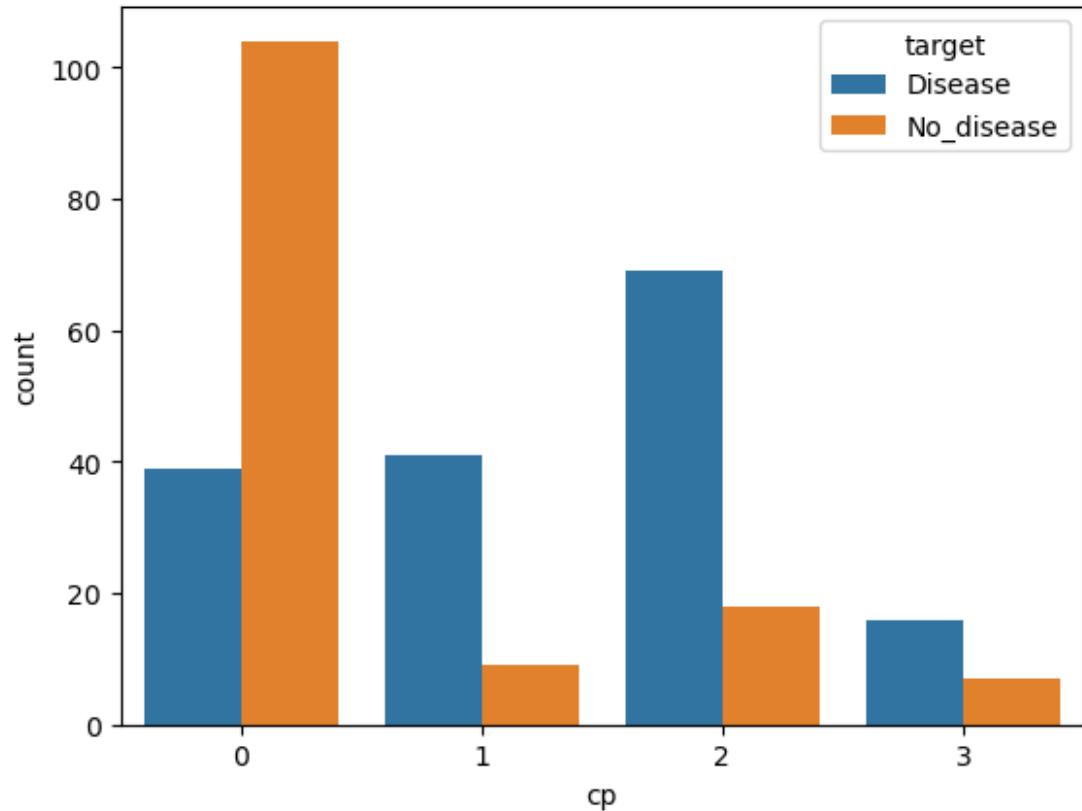
```
sns.scatterplot(x=df['oldpeak'],y=df['chol'],hue=df['target'],size=df['age'])  
<AxesSubplot:xlabel='oldpeak', ylabel='chol'>
```



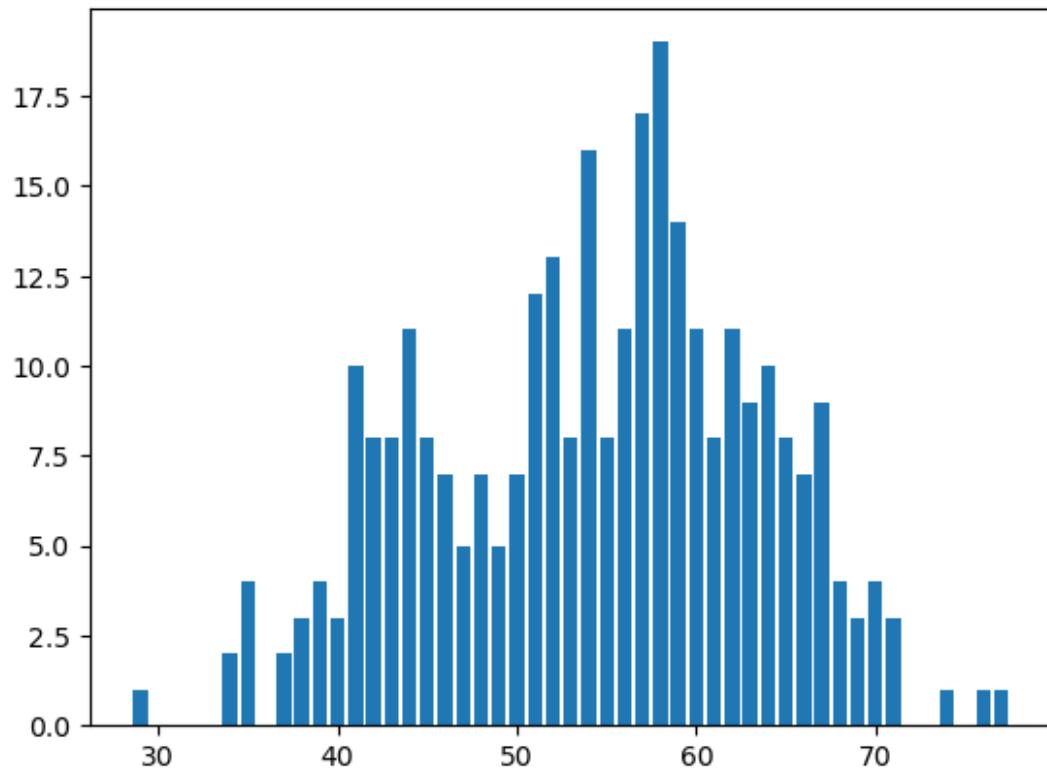
```
df.target.value_counts().plot(kind='bar')  
<AxesSubplot:>
```



```
sns.countplot(x='cp',hue='target',data=df)  
<AxesSubplot:xlabel='cp', ylabel='count'>
```

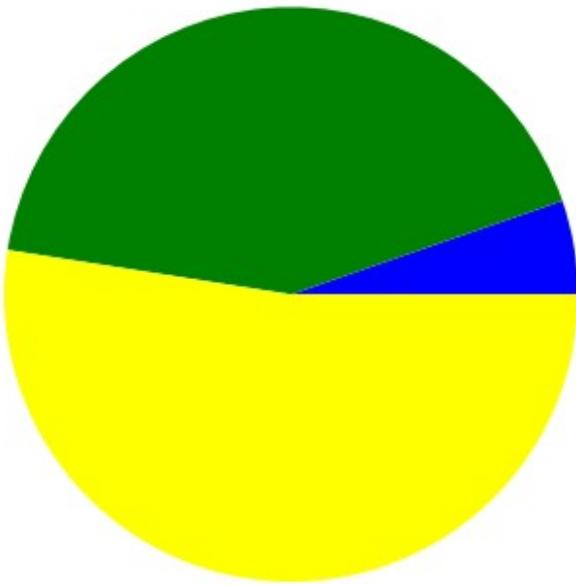


```
plt.bar(df['age'].value_counts().index,df['age'].value_counts())
<BarContainer object of 41 artists>
```



```
young = df[(df['age'] >=29) & (df['age'] < 40)]
middle = df[(df['age'] >=40) & (df['age'] < 55)]
old = df[(df['age'] >=55)]

colors = ['blue','green','yellow']
plt.pie([len(young),len(middle),len(old)],colors=colors);
```



```
df['age'].valu
-----
-----
AttributeError                               Traceback (most recent call
last)
~\AppData\Local\Temp\ipykernel_5032\2875843194.py in <module>
----> 1 df['age'].valuse_count()

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in
__getattr__(self, name)
    5573         ):
    5574             return self[name]
-> 5575         return object.__getattribute__(self, name)
    5576
    5577     def __setattr__(self, name: str, value) -> None:
```

AttributeError: 'Series' object has no attribute 'valuse\_count'