DATA LOADING

```
#Data Loading
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

How My Data Looks

```
df = pd.read_csv("/content/haberman.csv", header = None)
print(df)
```

```
       0   1  2  3
0     30  64  1  1
1     30  62  3  1
2     30  65  0  1
3     31  59  2  1
4     31  65  4  1
..    ..  .. .. ..
301   75  62  1  1
302   76  67  0  1
303   77  65  3  1
304   78  65  1  2
305   83  58  2  2

[306 rows x 4 columns]
```

Attribute Information:

Patient's age (numerical) at the time of the procedure The patient's year of surgery (numerical year - 1900) The number (numerical) of positive axillary nodes found

Status of survival (class attribute) 1 = The patient lived for five years or more 2: The patient passed away in five years.

```
df.head()
```

	0	1	2	3
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1
3	31	59	2	1
4	31	65	4	1

IS THERE ANY STEP I NEED TO TAKE TO AVOID DATA REDUNDANCY?

```
df = pd.read_csv("/content/haberman.csv", header = None, names=['Age','Year','Nodes','Sur
print(df)
```

```
     Age  Year  Nodes  Survival
0     30    64      1         1
1     30    62      3         1
2     30    65      0         1
3     31    59      2         1
4     31    65      4         1
..   ...   ...    ...       ...
301   75    62      1         1
302   76    67      0         1
303   77    65      3         1
304   78    65      1         2
305   83    58      2         2

[306 rows x 4 columns]
```

```
df.head()
```

	Age	Year	Nodes	Survival
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1
3	31	59	2	1
4	31	65	4	1

What is the dimension of my data ?

```
df.shape
```

```
(306, 4)
```

What is the datatype of my columns?

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Age       306 non-null    int64
 1   Year      306 non-null    int64
 2   Nodes     306 non-null    int64
 3   Survival  306 non-null    int64
dtypes: int64(4)
memory usage: 9.7 KB
```

What is the mathematical overview of my dataset ?

```
df.describe()
```

	Age	Year	Nodes	Survival
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

Is there any null value in my dataset ?

```
df.isnull().sum()
```

```
Age         0
Year        0
Nodes       0
Survival    0
dtype: int64
```

Is there any duplicated?

```
df.duplicated().sum()
```

```
17
```

Is there any co-relation in my dataset?

```
df.corr()["Age"]
```

```
Age         1.000000
Year        0.089529
Nodes      -0.063176
Survival    0.067950
Name: Age, dtype: float64
```

DATA MANIPULATION

```
df['Survival']=df['Survival'].map({1:"Yes",2:"No"})
df
```

	Age	Year	Nodes	Survival
0	30	64	1	Yes
1	30	62	3	Yes
2	30	65	0	Yes
3	31	59	2	Yes
4	31	65	4	Yes
...	...	...	...	...
301	75	62	1	Yes
302	76	67	0	Yes
303	77	65	3	Yes
304	78	65	1	No
305	83	58	2	No

306 rows × 4 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Age       306 non-null    int64
 1   Year      306 non-null    int64
 2   Nodes     306 non-null    int64
 3   Survival  306 non-null    object
dtypes: int64(3), object(1)
memory usage: 9.7+ KB
```

```
df['Survival'] = df['Survival'].astype('category')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Age       306 non-null    int64
 1   Year      306 non-null    int64
 2   Nodes     306 non-null    int64
 3   Survival  306 non-null    category
dtypes: category(1), int64(3)
memory usage: 7.7 KB
```

```
df.shape[0]
```

```
306
```

```
df.shape[1]
```

```
4
```

```
df['Survival'].value_counts(normalize=True)
```

```
Yes    0.735294
No     0.264706
Name: Survival, dtype: float64
```

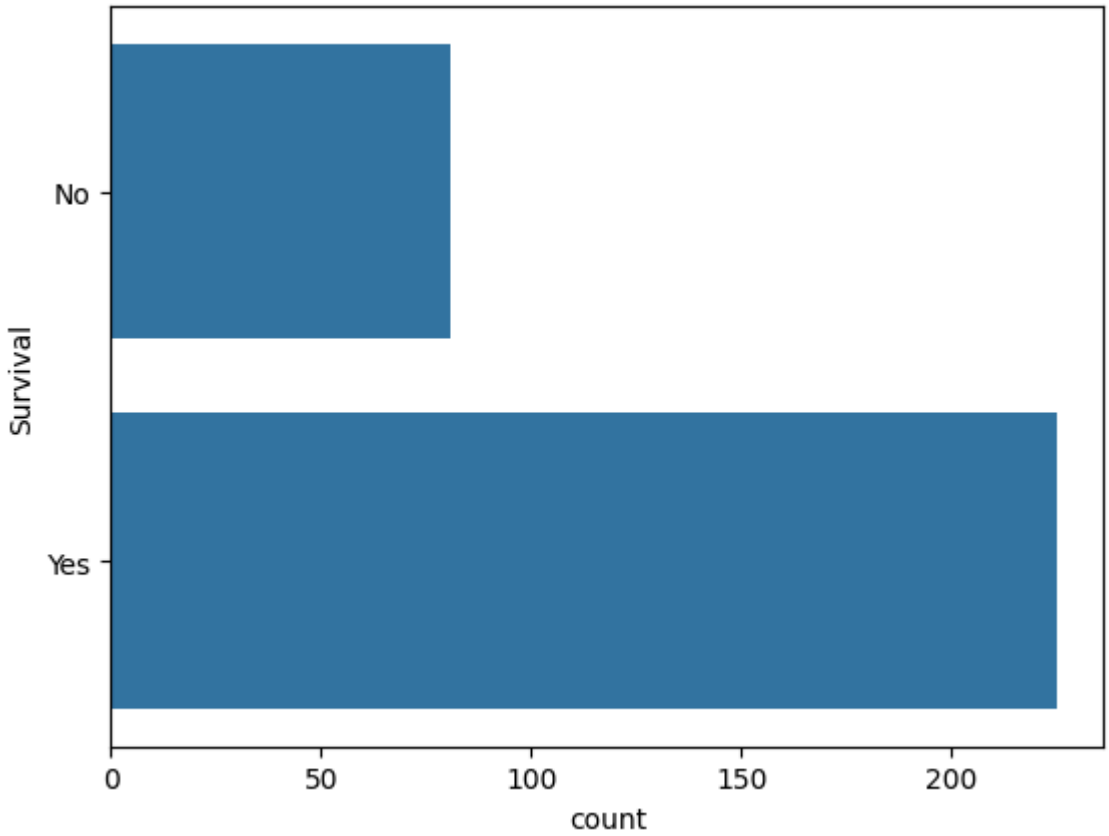
Univarient Analysis

```
import seaborn as sns
```

CATEGORICAL ANALYSIS

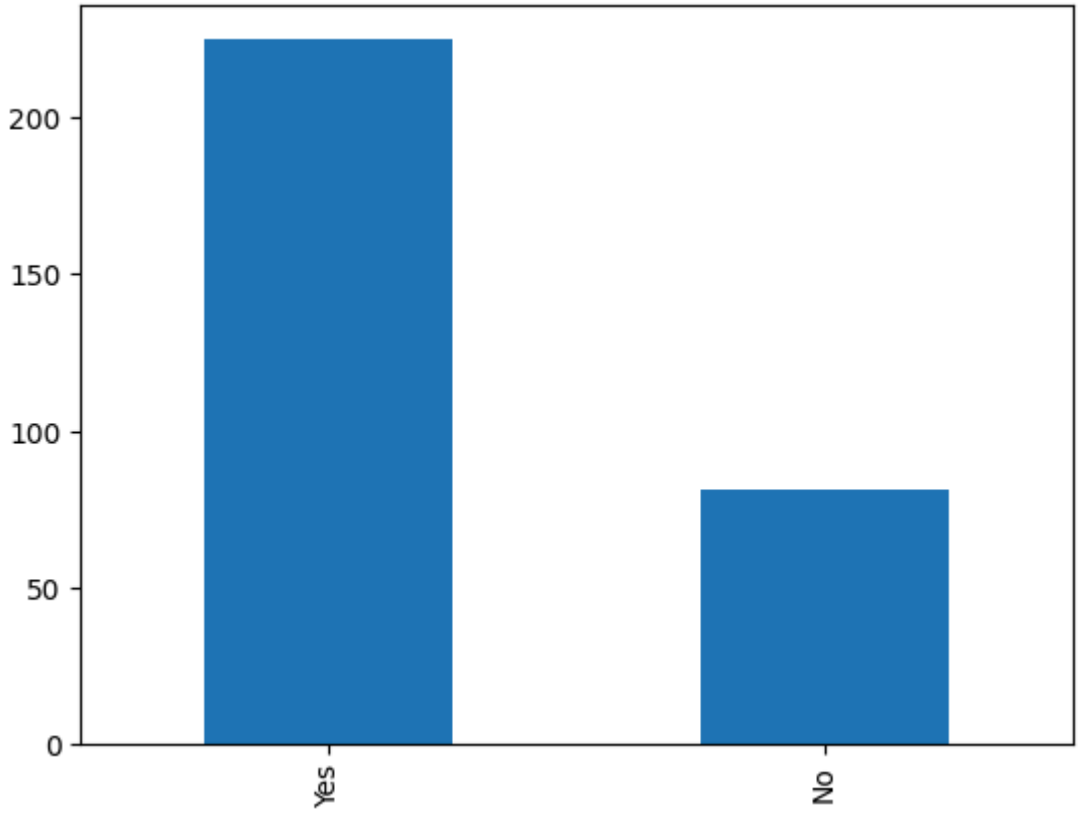
```
sns.countplot(df["Survival"])
```

```
<Axes: xlabel='count', ylabel='Survival'>
```



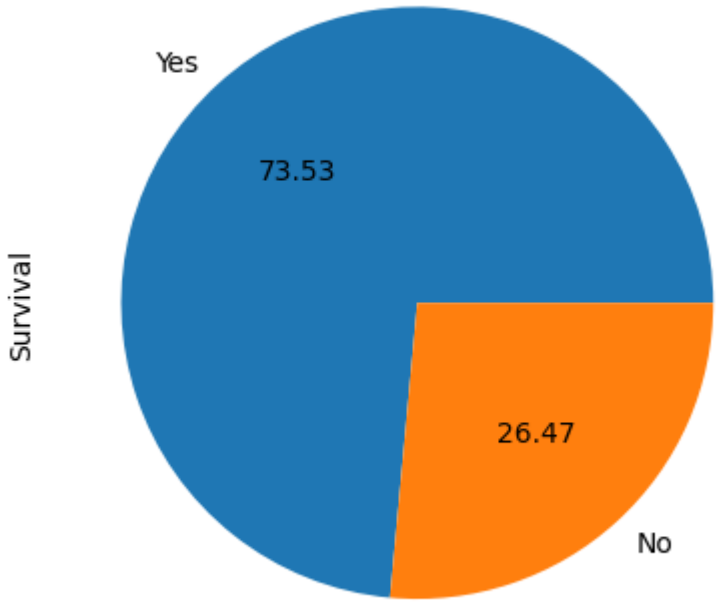

```
df["Survival"].value_counts().plot(kind="bar")
```

```
<Axes: >
```




```
df["Survival"].value_counts().plot(kind="pie",autopct="%.2f")
```

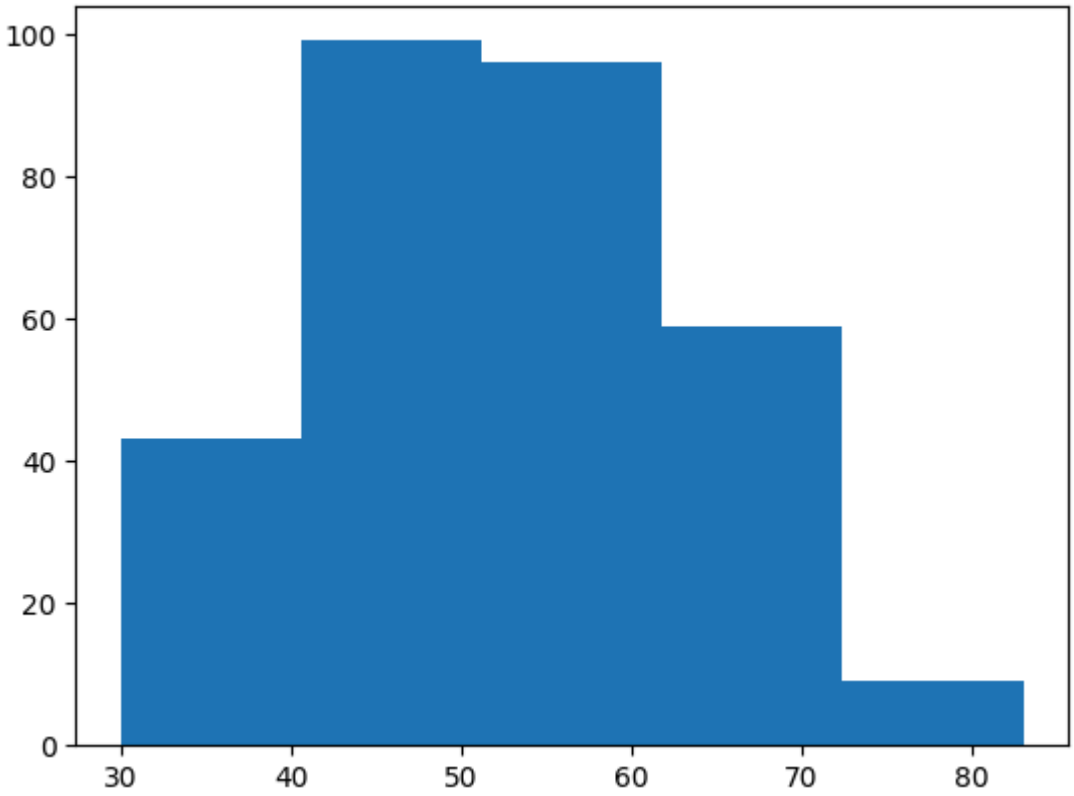
```
<Axes: ylabel='Survival'>
```



Numerical Data

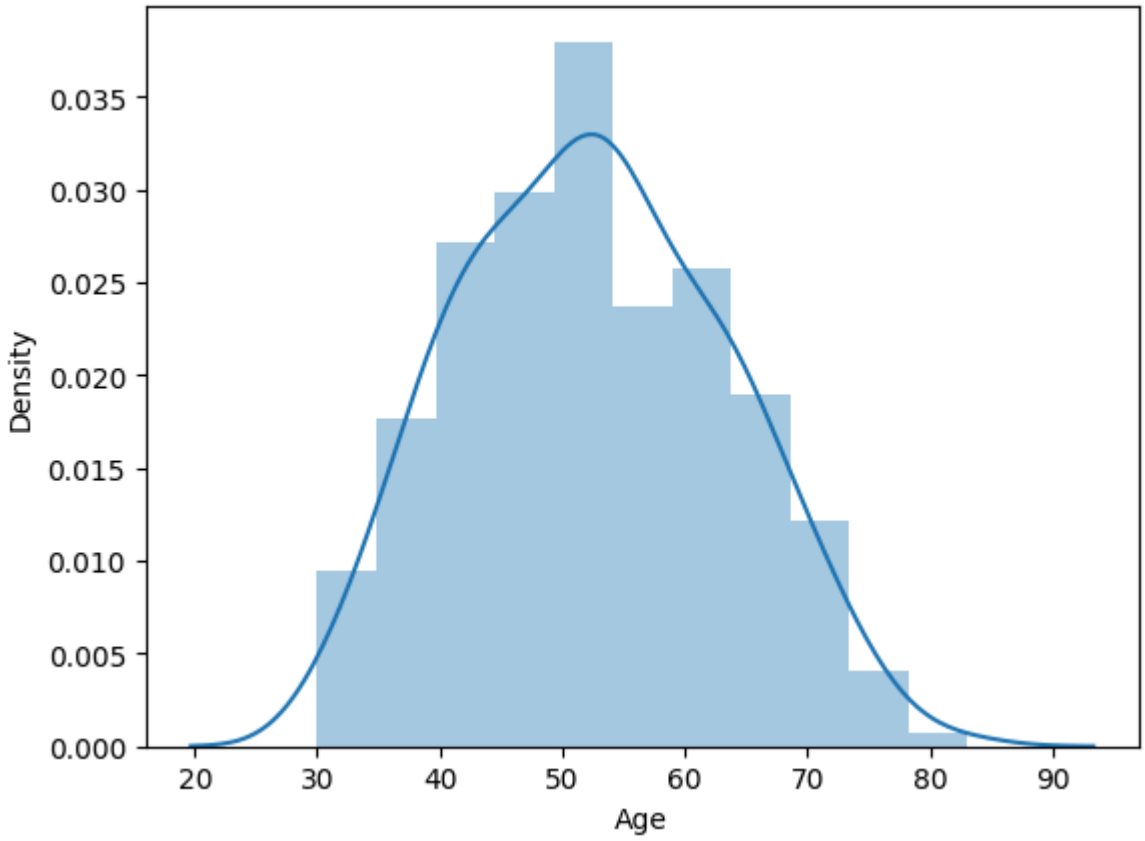
```
plt.hist(df["Age"], bins=5)
```

```
(array([43., 99., 96., 59.,  9.]),
 array([30. , 40.6, 51.2, 61.8, 72.4, 83. ]),
 <BarContainer object of 5 artists>)
```



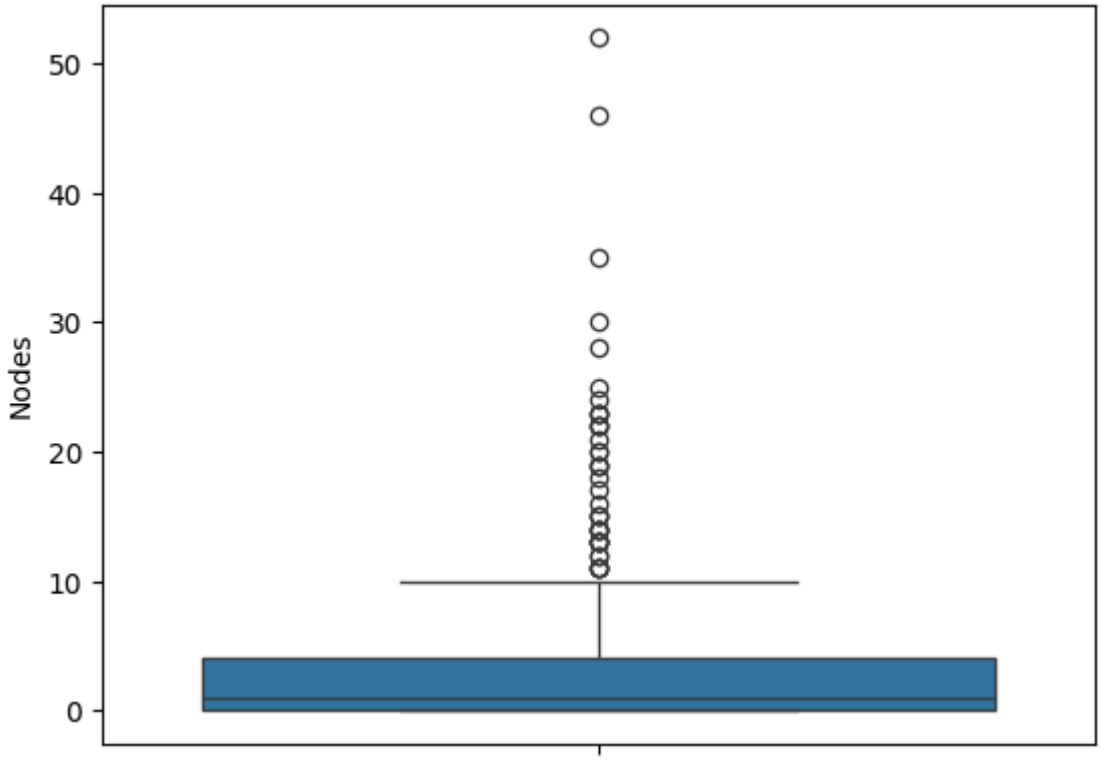

```
sns.distplot(df["Age"])
```

```
<ipython-input-182-cf0334540b62>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Age"])
<Axes: xlabel='Age', ylabel='Density'>
```




```
sns.boxplot(df["Nodes"])
```

```
<Axes: ylabel='Nodes'>
```

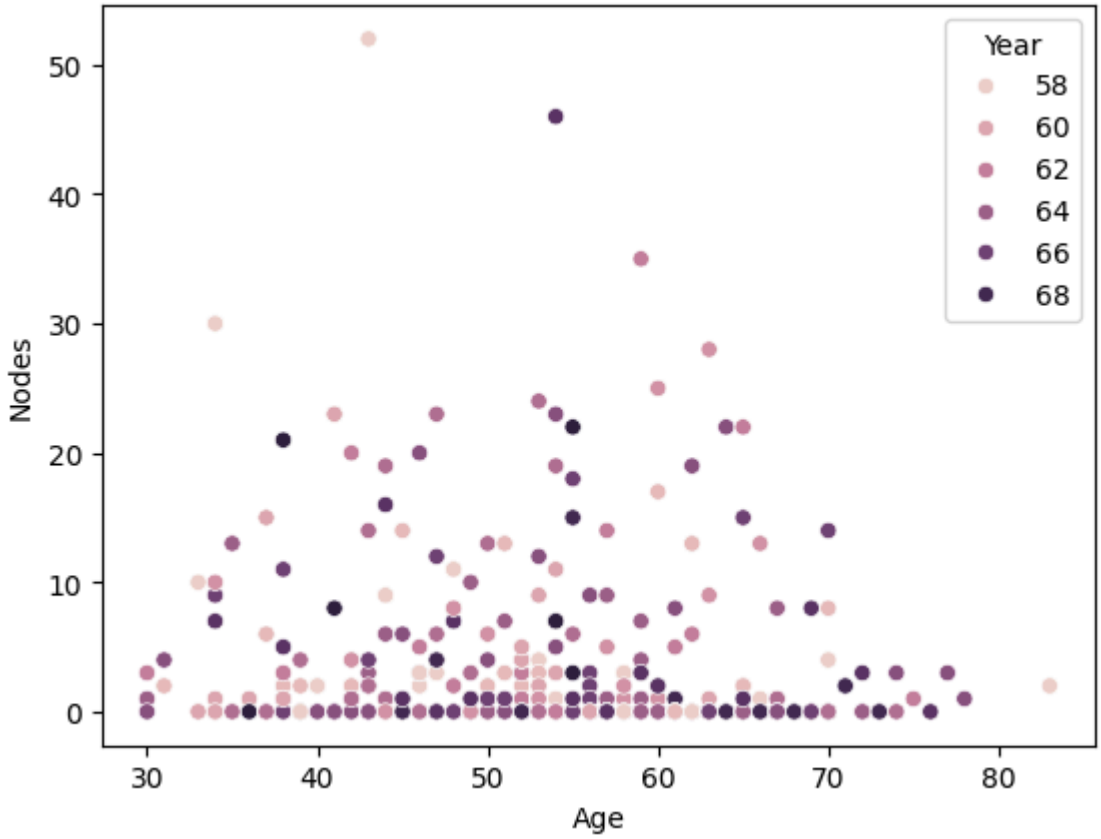


Multivarient Analysis

Numerical-Numerical

```
sns.scatterplot(x="Age", y="Nodes", data=df,hue="Year")
```

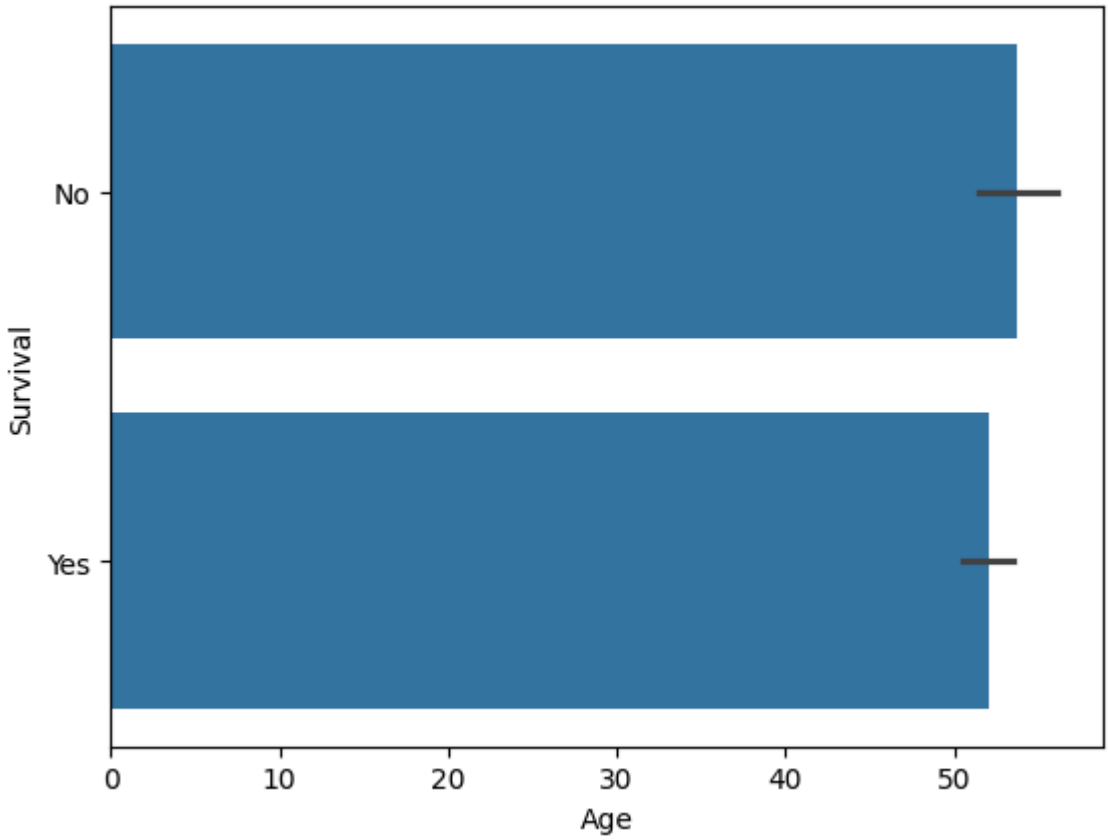
```
<Axes: xlabel='Age', ylabel='Nodes'>
```



Numericl-Categorical

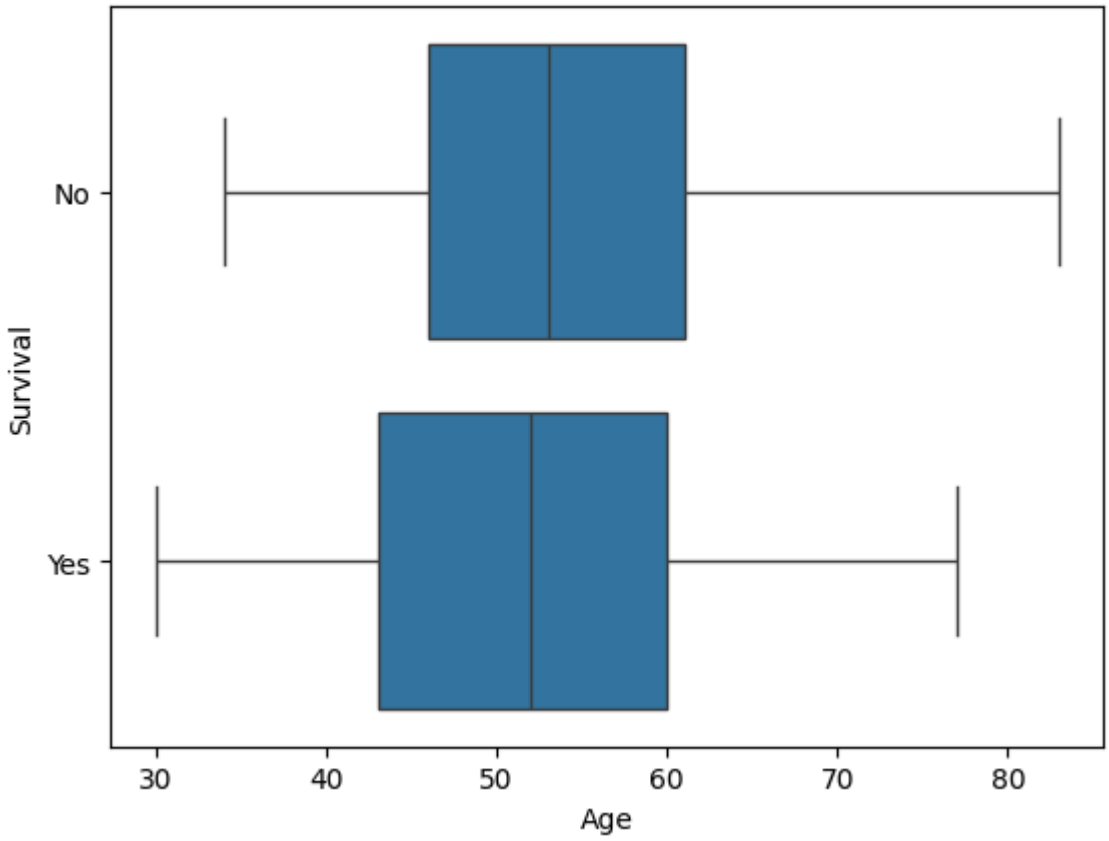
```
sns.barplot(x = "Age", y = "Survival", data = df)
```

```
<Axes: xlabel='Age', ylabel='Survival'>
```



```
sns.boxplot(x="Age",y="Survival",data=df)
```

```
<Axes: xlabel='Age', ylabel='Survival'>
```




```
sns.distplot(df["Age"],hist=False)
sns.distplot(df["Nodes"],hist=False)
```

```
<ipython-input-196-39b0852b716c>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```