In [1]:
```
import numpy as np
from numpy import percentile
from numpy.random import rand

input_string = input('Enter elements of a list separated by space ')
print("\n")

user_list = input_string.split()

# print list
print('list: ', user_list)

# convert each item to int type
for i in range(len(user_list)):
    # convert each item to int type
    user_list[i] = int(user_list[i])

arr1 = np.array(user_list)
print('arr1 elements: \n')
print(arr1)
print(type(arr1))


quartiles = percentile(arr1, [25,50,75])

data_min, data_max = arr1.min(), arr1.max()

print('         5 NUMBER SUMMARY                      ')

print('**************************************************')
print('Minimum = ',data_min)
print('Q1: %.3f',quartiles[0])
print('Median: %.3f',quartiles[1])
print('Q3: %.3f',quartiles[2])
print('Q1: %.3f',data_max)
print('**************************************************')
```

```
Enter elements of a list separated by space 1 2 3 4 5 6 7 8 9


list:  ['1', '2', '3', '4', '5', '6', '7', '8', '9']
arr1 elements:

[1 2 3 4 5 6 7 8 9]
<class 'numpy.ndarray'>
         5 NUMBER SUMMARY
**************************************************
Minimum =  1
Q1: %.3f 3.0
Median: %.3f 5.0
Q3: %.3f 7.0
Q1: %.3f 9
**************************************************
```

In [3]:
```
import pandas as pd

print(arr1)

df = pd.DataFrame(arr1)

df.describe()
```

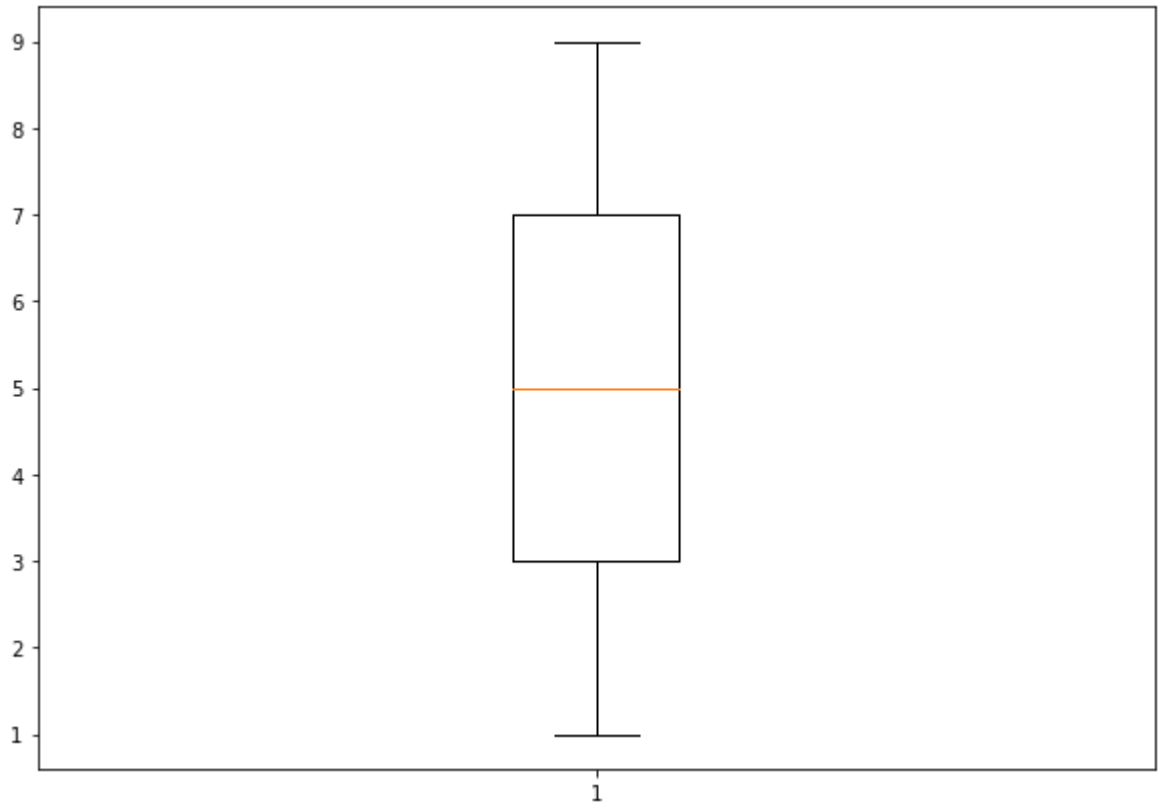
```
[1 2 3 4 5 6 7 8 9]
```

Out[3]:

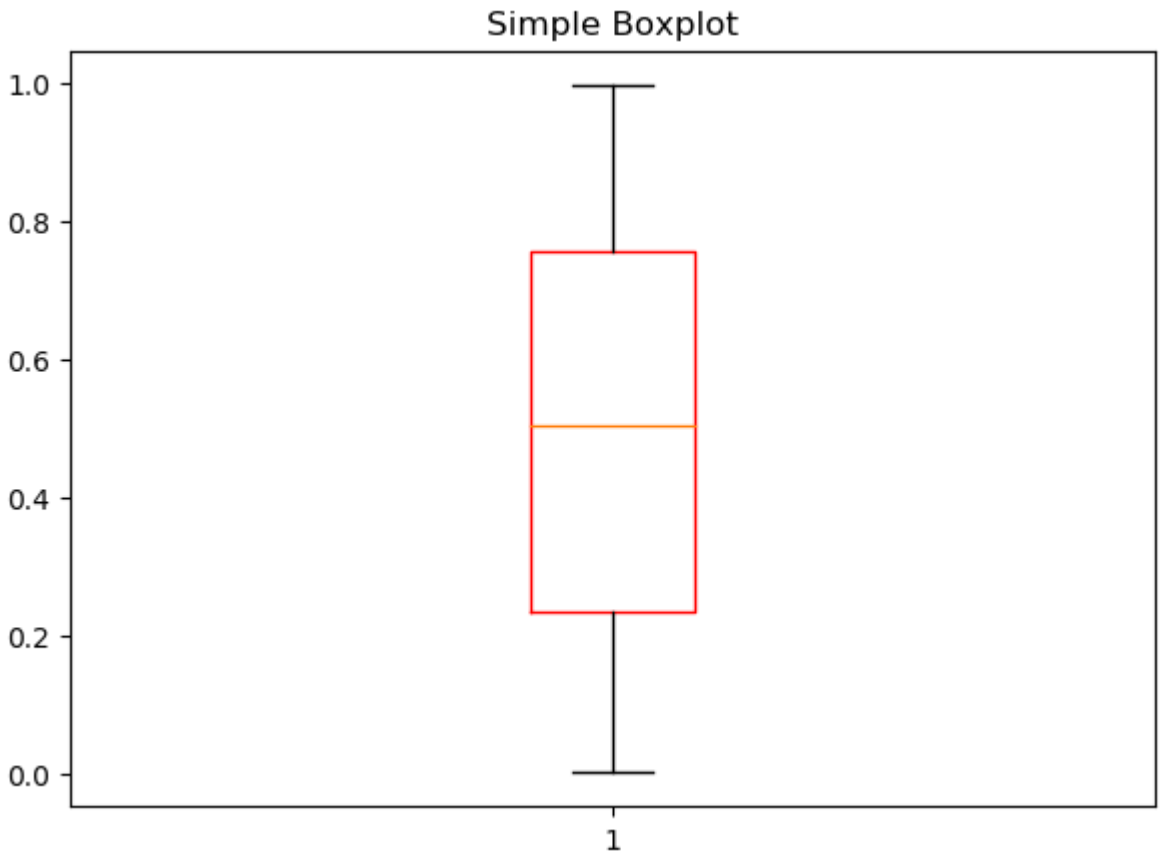
	0
count	9.000000
mean	5.000000
std	2.738613
min	1.000000
25%	3.000000
50%	5.000000
75%	7.000000
max	9.000000

In [10]:
```
# Import libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

print('data is \n')
print(arr1)
print('\n')

quartiles = percentile(arr1, [25,50,75])

data_min, data_max = arr1.min(), arr1.max()

print('         5 NUMBER SUMMARY                        ')

print('************************************************')
print('Minimum = ',data_min)
print('Q1: %.3f',quartiles[0])
print('Median: %.3f',quartiles[1])
print('Q3: %.3f',quartiles[2])
print('Q1: %.3f',data_max)
print('************************************************')

fig = plt.figure(figsize =(10, 7))


# Creating plot
plt.boxplot(arr1)

# show plot
plt.show()
```

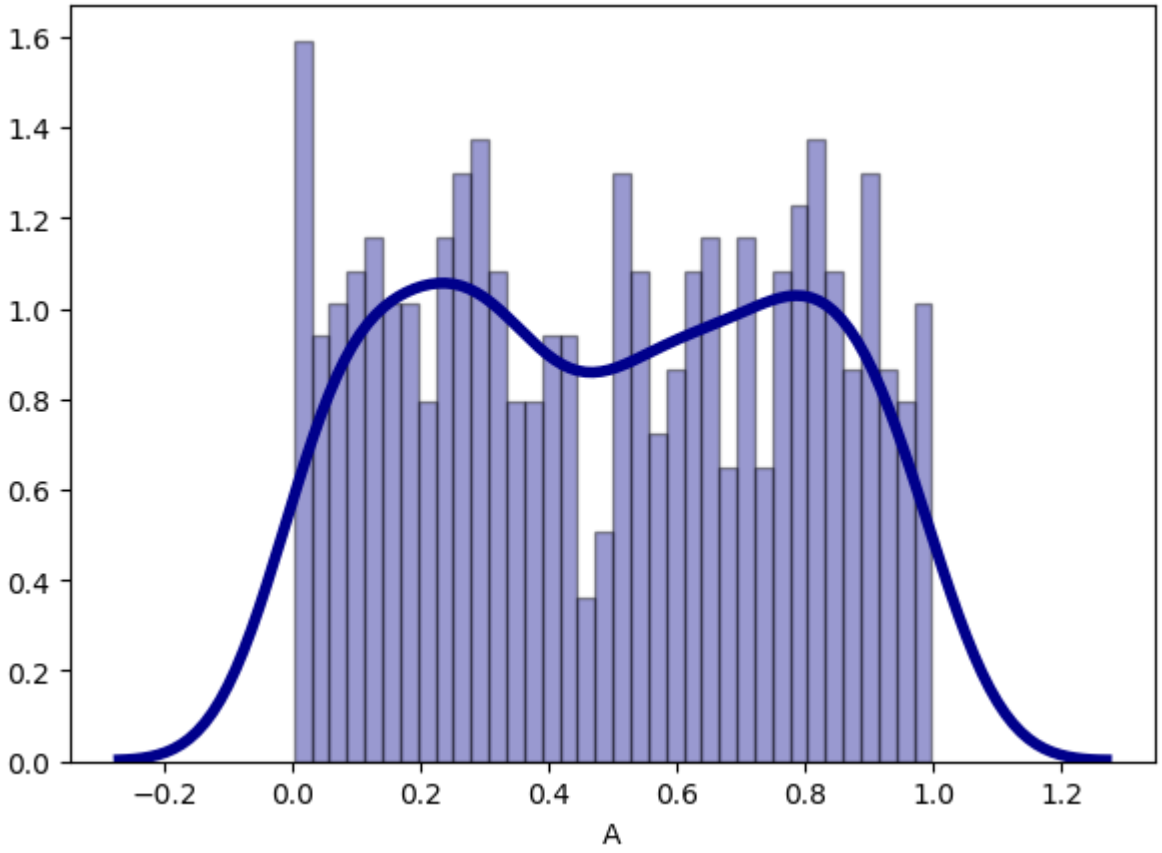
```
data is

[1 2 3 4 5 6 7 8 9]


         5 NUMBER SUMMARY
**************************************************
Minimum =  1
Q1: %.3f 3.0
Median: %.3f 5.0
Q3: %.3f 7.0
Q1: %.3f 9
**************************************************
```



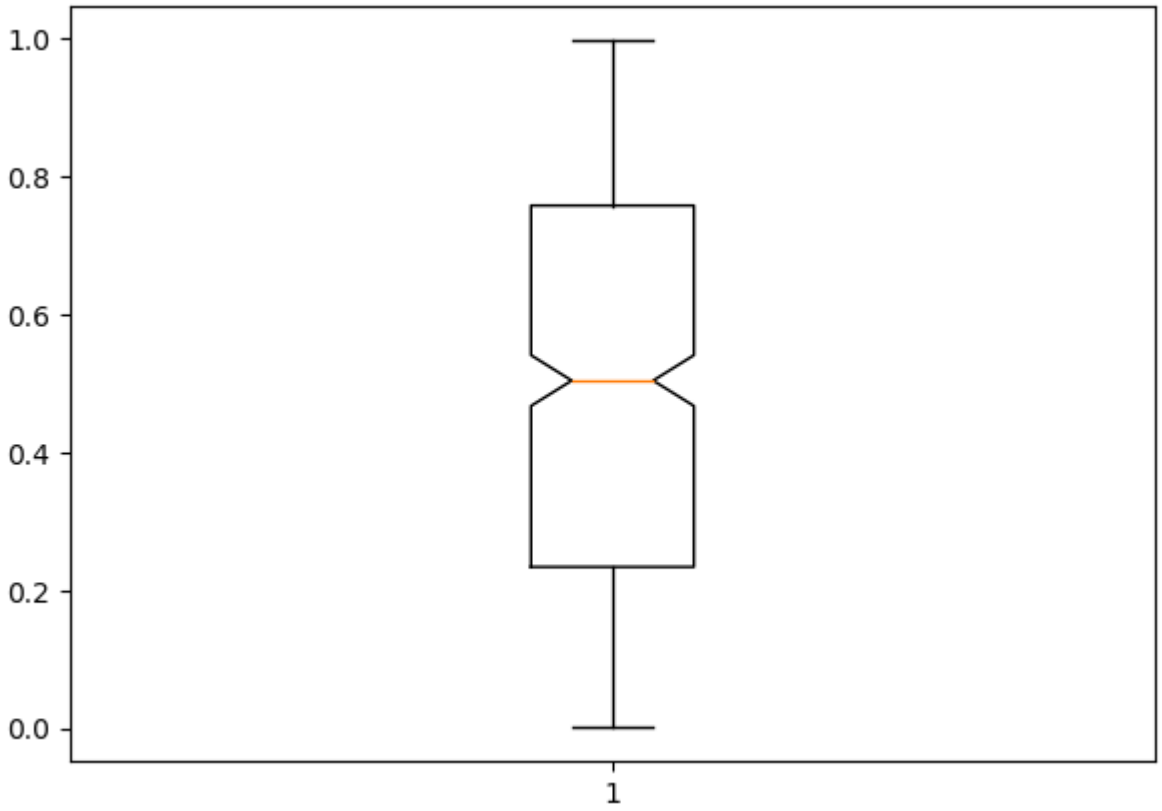
In [13]:
```
# Load package
import pandas as pd, numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})

# Creating dataset
df = pd.DataFrame(np.random.rand(500, 1), columns=['A'])

# plot
plt.boxplot(df['A'], boxprops=dict(color='red'))
plt.title('Simple Boxplot');
```



In [14]:
```
# Histogram and density
import seaborn as sns

sns.distplot(df['A'], hist=True, kde=True,
             bins=int(180/5), color = 'darkblue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 4});
```



In [15]:
```
# Notched box plot
plt.boxplot(df['A'],notch=True);
```

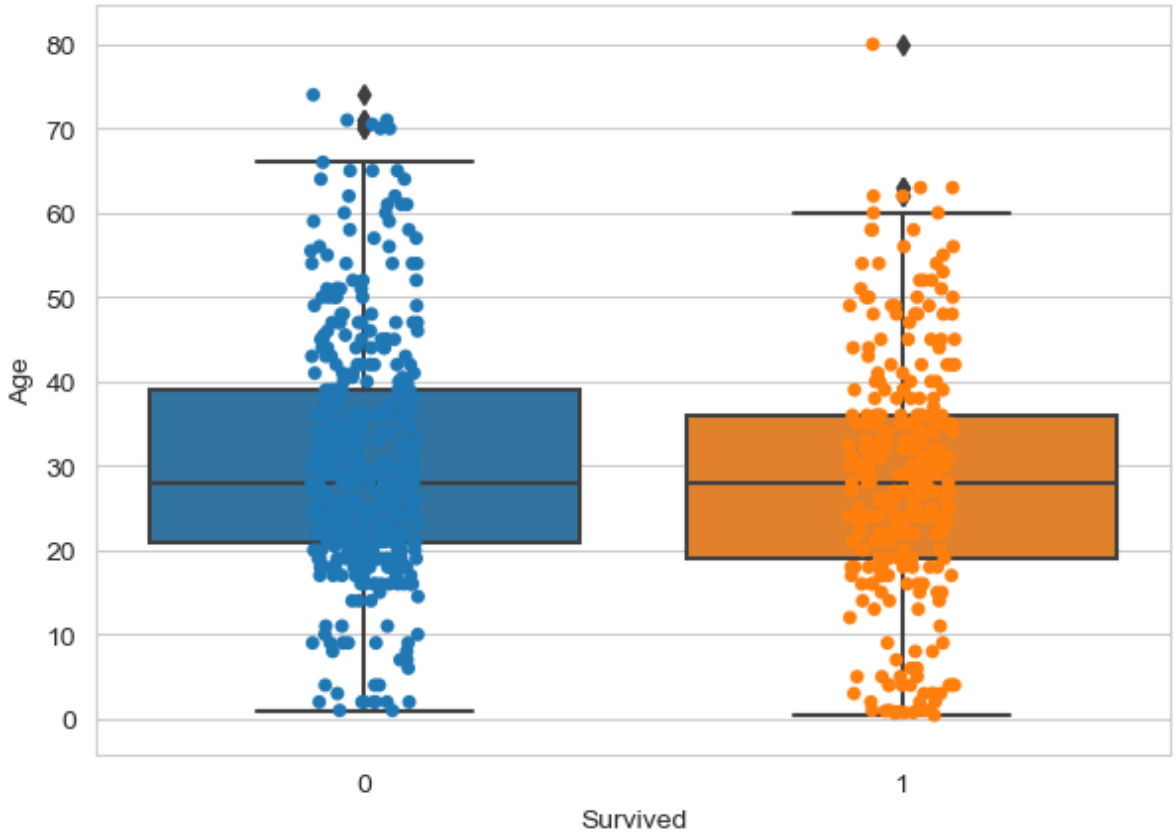


In [16]:
```
# Import data
df=pd.read_csv("https://raw.githubusercontent.com/ven-27/datasets/master/titanic.csv")
df.head()
```

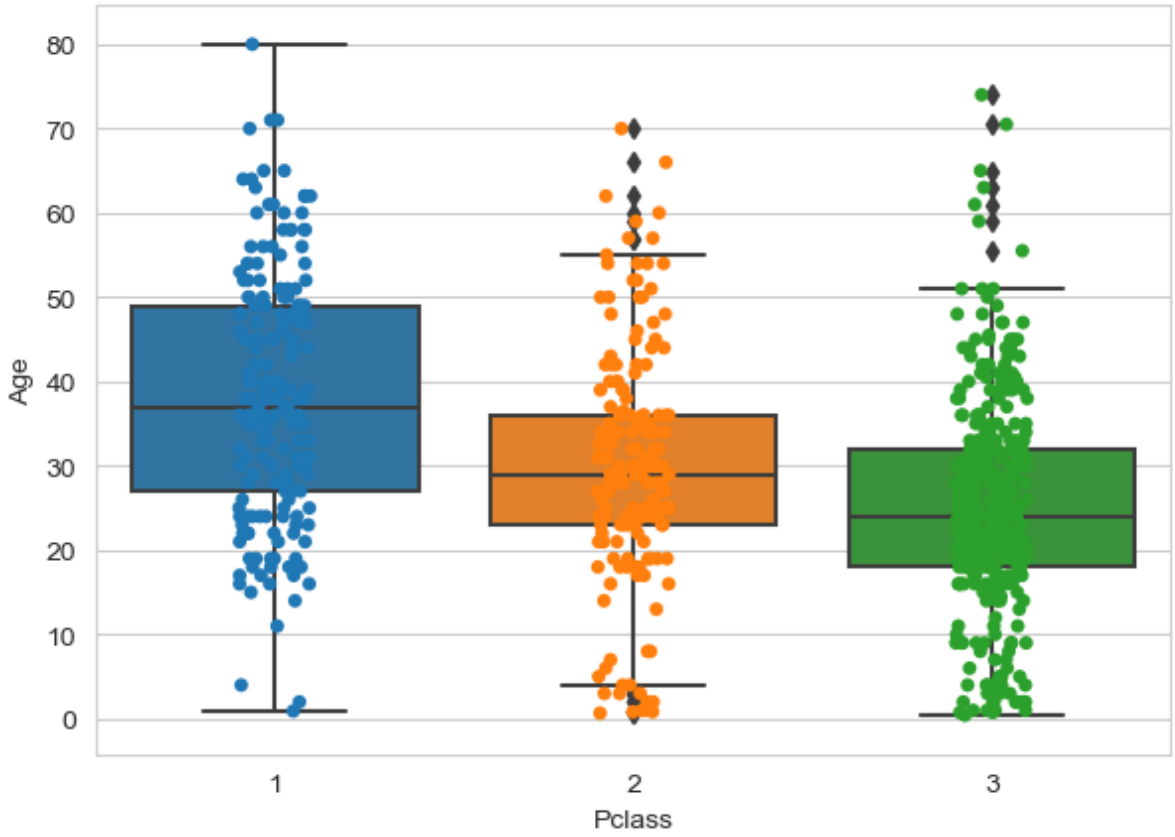
Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

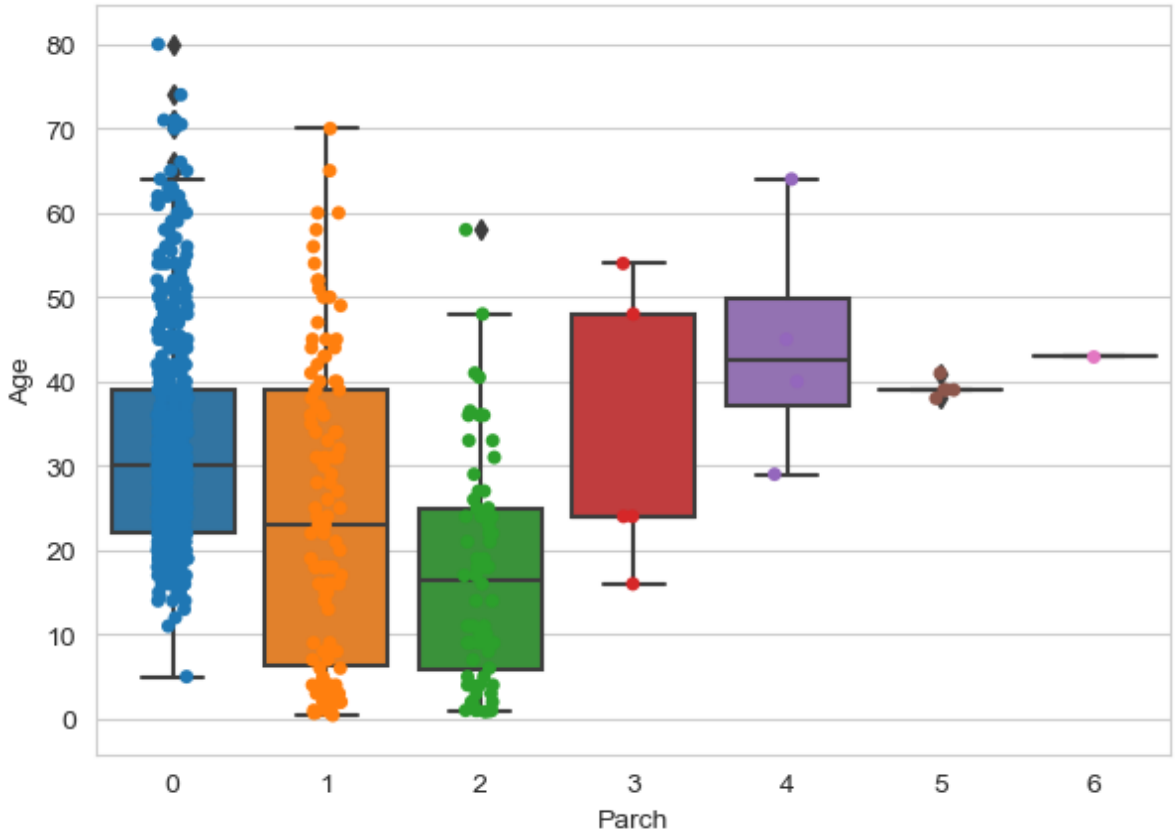
In [17]:
```
#et's look into the distribution of survived based on the age of the passenge
r.
# Boxplot with Seaborn
import seaborn as sns
sns.set_style('whitegrid')
ax= sns.boxplot(x='Survived',y='Age',data=df)
ax = sns.stripplot(x="Survived", y="Age",data=df)
```



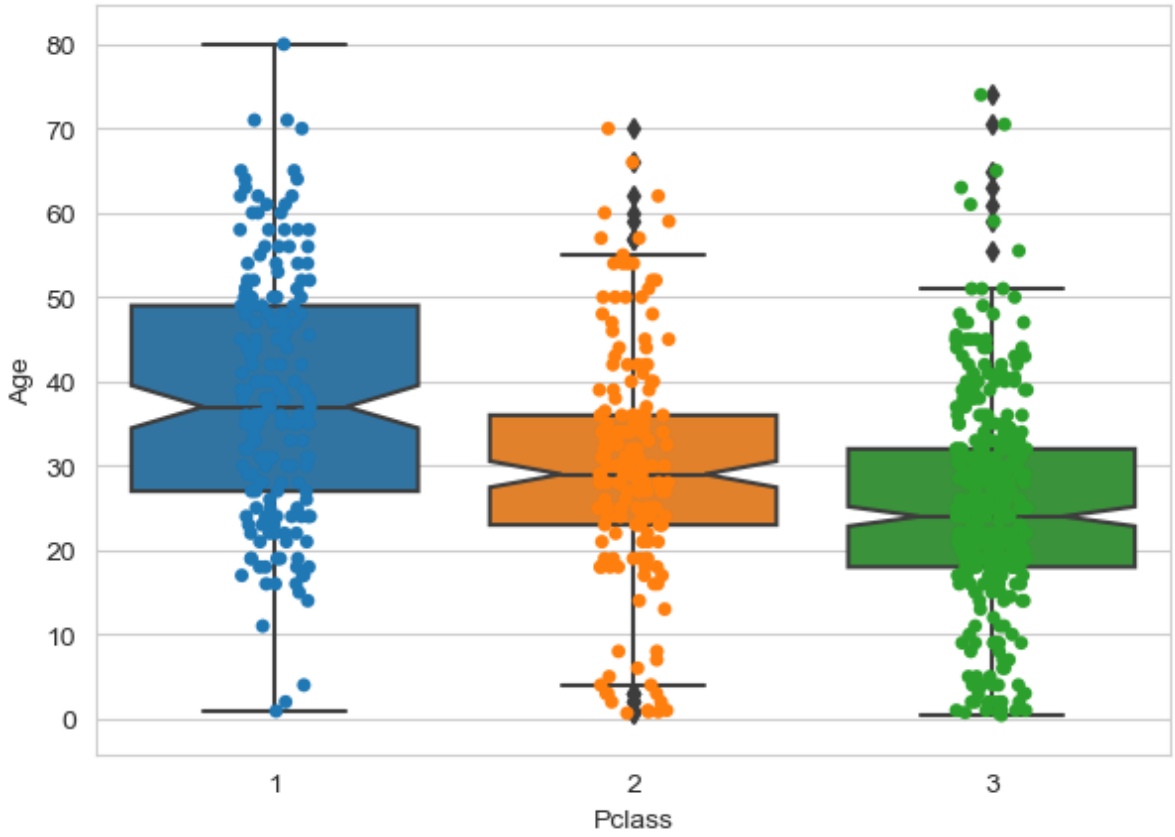
In [18]:
```
sns.set_style('whitegrid')
ax= sns.boxplot(x='Pclass',y='Age',data=df)
ax = sns.stripplot(x="Pclass", y="Age",data=df)
```



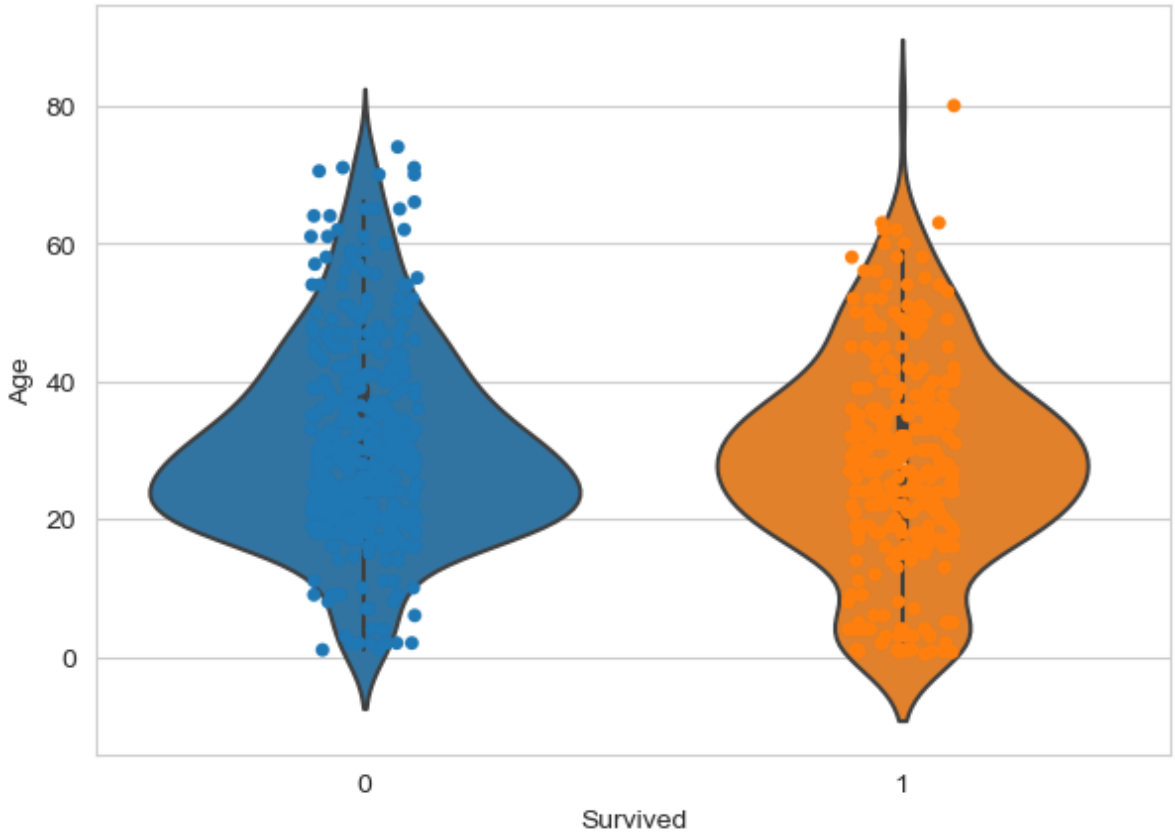
In [19]:
```
sns.set_style('whitegrid')
ax= sns.boxplot(x='Parch',y='Age',data=df)
ax = sns.stripplot(x="Parch", y="Age",data=df)
```



In [20]:
```
sns.set_style('whitegrid')
ax= sns.boxplot(x='Pclass',y='Age',data=df,notch=True)
ax = sns.stripplot(x="Pclass", y="Age",data=df)
```



In [21]:
```
import seaborn as sns
sns.set_style('whitegrid')
ax= sns.violinplot(x='Survived',y='Age',data=df)
ax = sns.stripplot(x="Survived", y="Age",data=df)
```



In [ ]:
```
```