Study and analysis of ANOVA Data for IRIS Data

211090073 Astik Sonawane

In [3]:
```
import sklearn
import pandas as pd
from sklearn import datasets

#load digit dataset
iris = datasets.load_iris()
```

In [ ]:
```
iris
```

In [4]:
```
df = pd.DataFrame(iris.data)
df.head()
```

Out[4]:

	0	1	2	3
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [5]:
```
iris['feature_names']
```

Out[5]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [6]:
```
import scipy.stats as stats
stats.f_oneway(df.iloc[:,0],df.iloc[:,1],df.iloc[:,2],df.iloc[:,3])
#pvalue less than alpha therefore we reject the hypothesis
```

Out[6]:
```
F_onewayResult(statistic=482.91531656927964, pvalue=4.660592480454751e-159)
```

Visualization with seaborn library

In [8]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [9]:
```
mtcars = pd.read_csv(r'C:\Users\HOME\Desktop\DSA\Lab5\mtcars.csv')
```

In [12]:
```
mtcars.head()
```

Out[12]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [13]:
```
mtcars.columns
```

Out[13]:
```
Index(['mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear',
       'carb'],
      dtype='object')
```

In [14]:
```
mtcars.shape
```

Out[14]:
```
(32, 11)
```

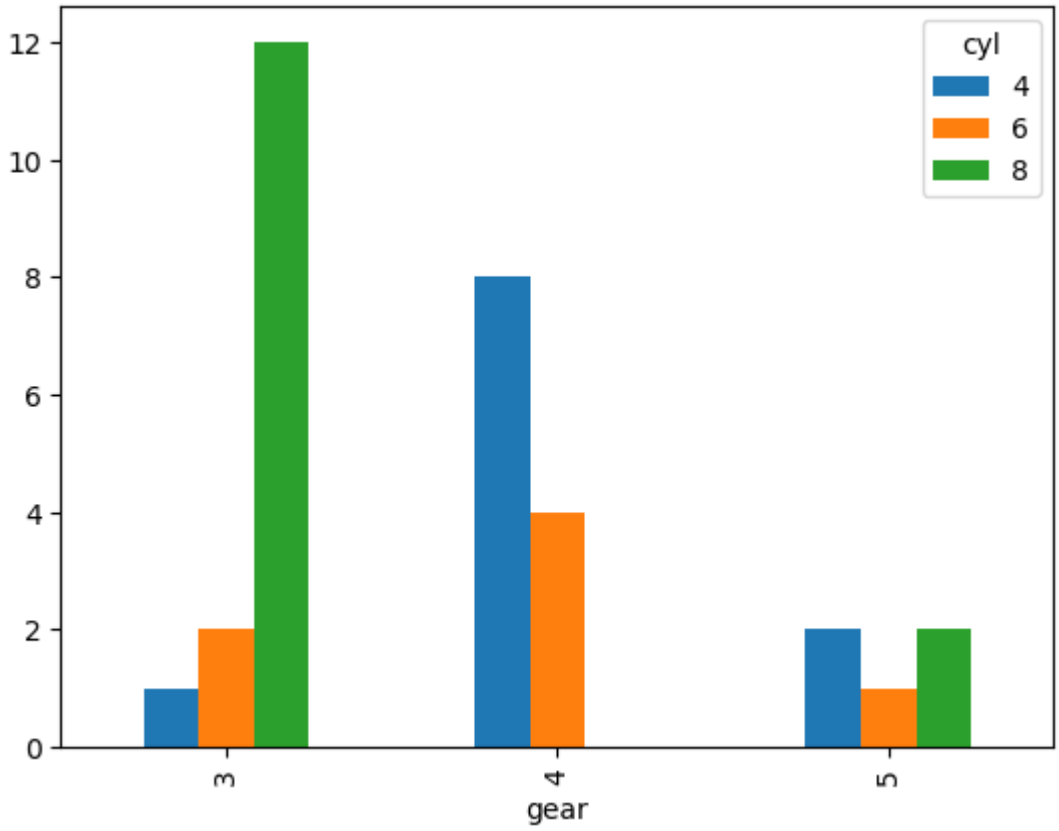
In [15]:
```
pd.crosstab(mtcars.gear,mtcars.cyl)
```

Out[15]:

cyl	4	6	8
gear			
3	1	2	12
4	8	4	0
5	2	1	2

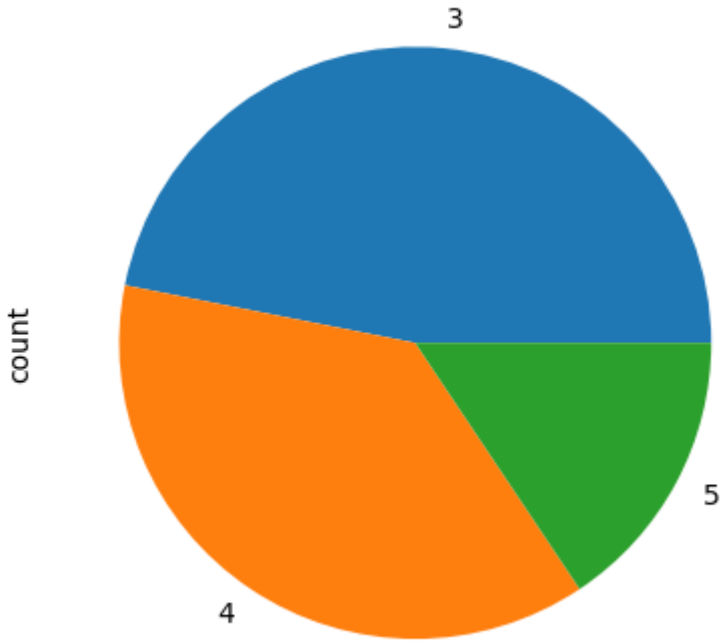
In [21]:
```
pd.crosstab(mtcars.gear,mtcars.cyl).plot(kind='bar')
```

Out[21]:
```
<Axes: xlabel='gear'>
```



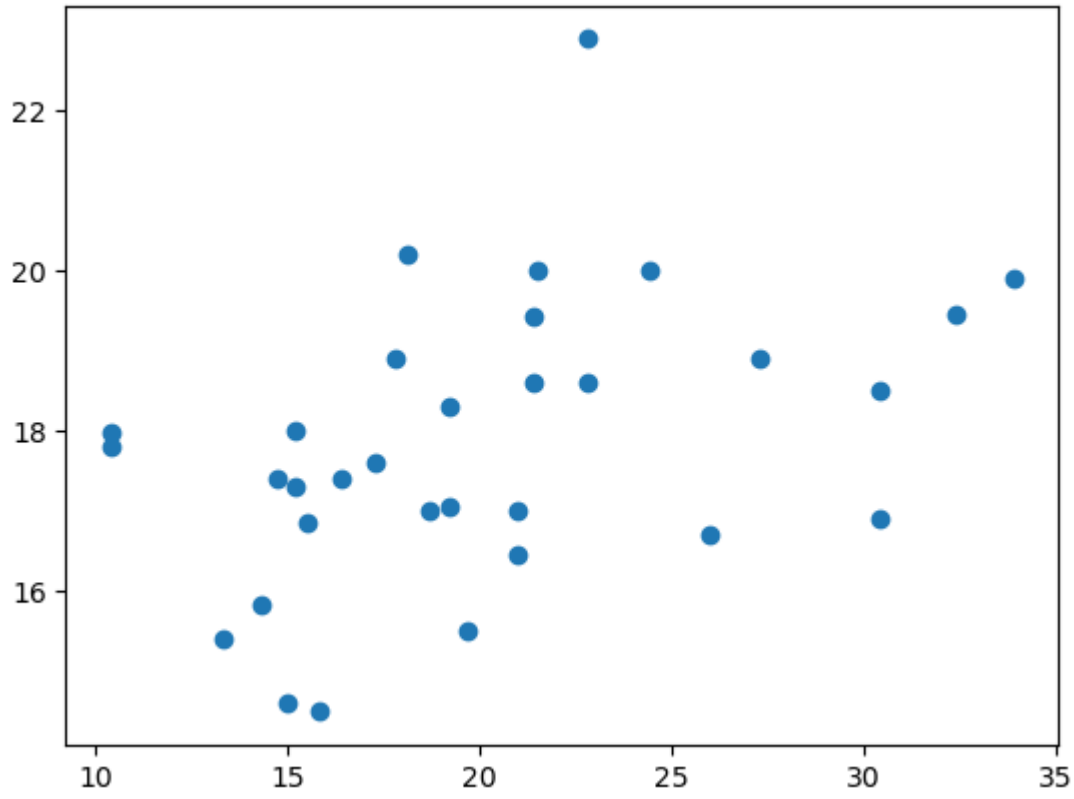
In [24]:
```
mtcars['gear'].value_counts()
mtcars.gear.value_counts().plot(kind="pie")
```

Out[24]: <Axes: ylabel='count'>



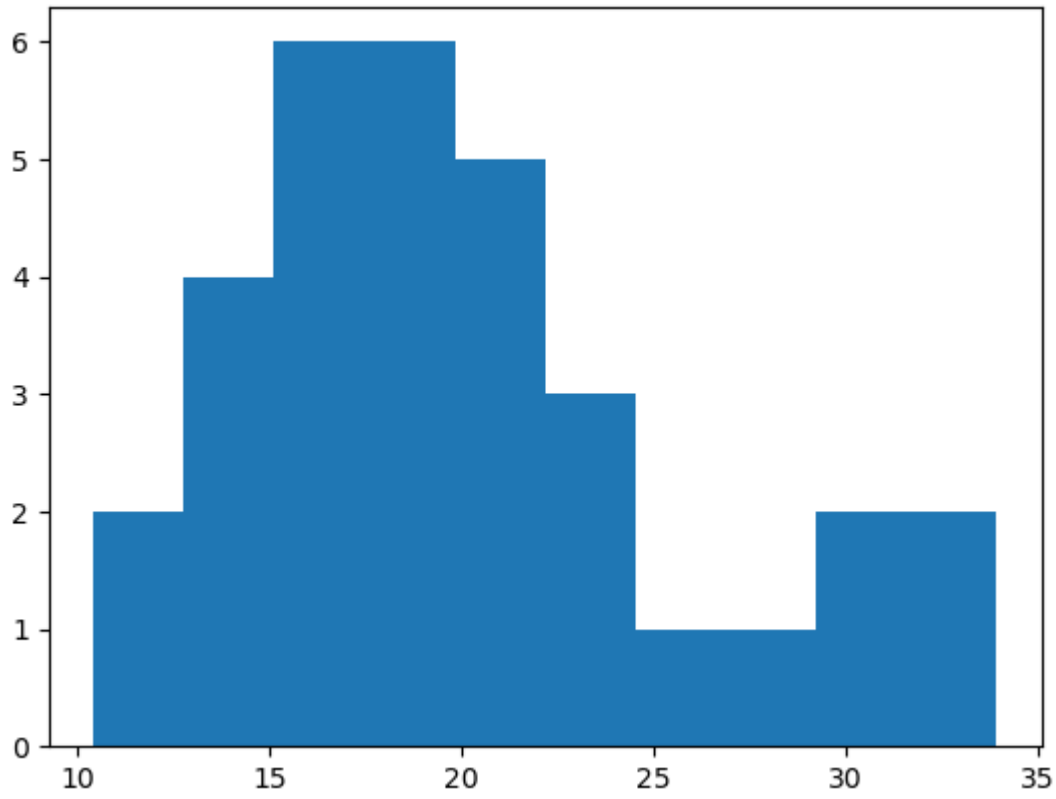
In [25]:
```
plt.scatter(mtcars.mpg,mtcars.qsec)
```

Out[25]:
```
<matplotlib.collections.PathCollection at 0x19be73efe90>
```



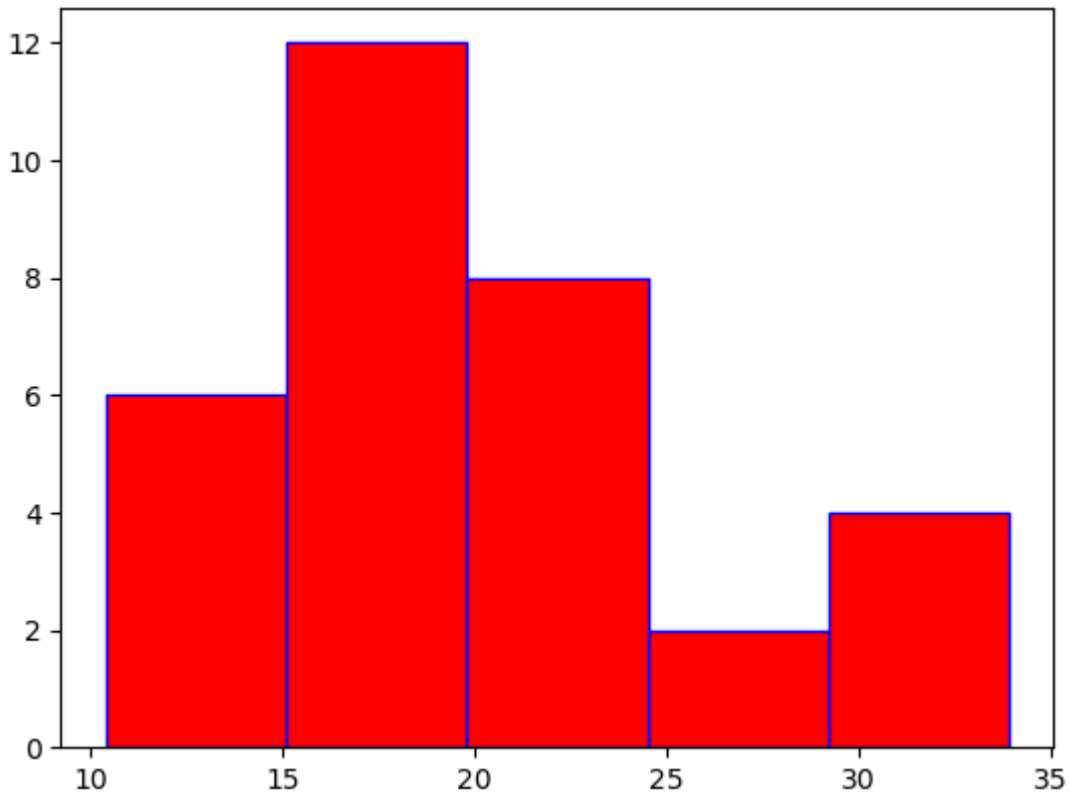
In [27]:
```
#Graphical representation of data
#histogram
plt.hist(mtcars['mpg'])
```

Out[27]:
```
(array([2., 4., 6., 6., 5., 3., 1., 1., 2., 2.]),
 array([10.4 , 12.75, 15.1 , 17.45, 19.8 , 22.15, 24.5 , 26.85, 29.2 ,
        31.55, 33.9 ]),
 <BarContainer object of 10 artists>)
```



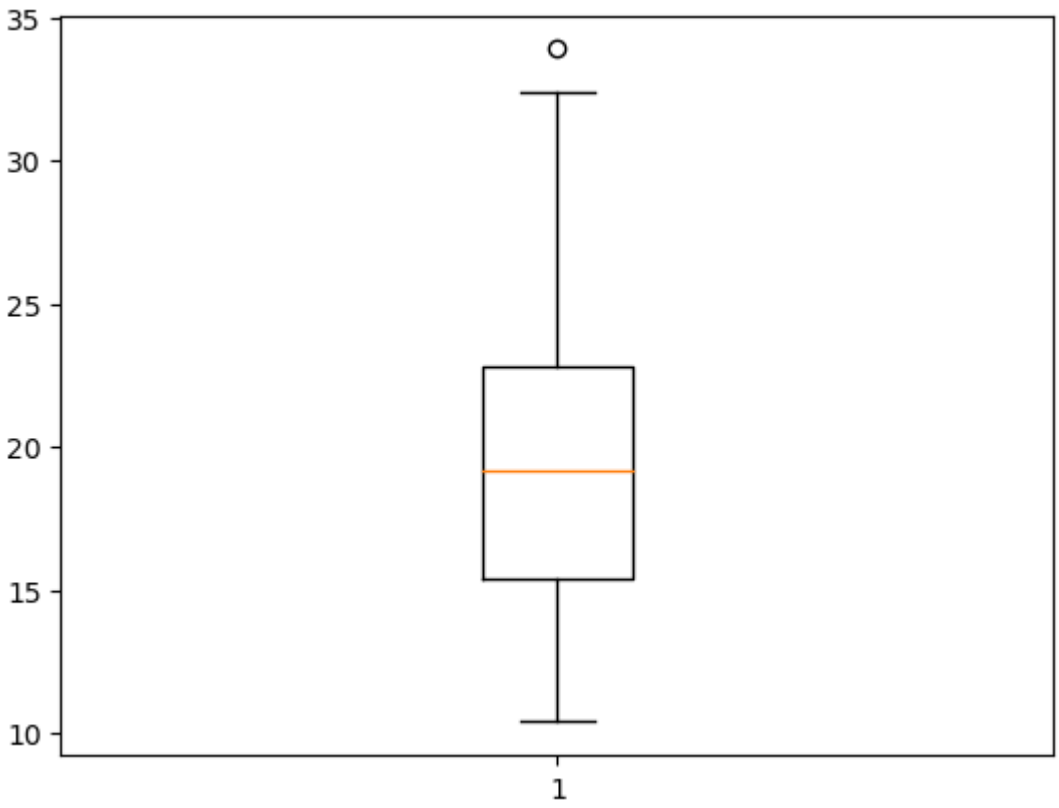
In [31]:
```
plt.hist(mtcars['mpg'], facecolor='red',edgecolor='blue',bins = 5)
```

Out[31]:
```
(array([ 6., 12.,  8.,  2.,  4.]),
 array([10.4, 15.1, 19.8, 24.5, 29.2, 33.9]),
 <BarContainer object of 5 artists>)
```



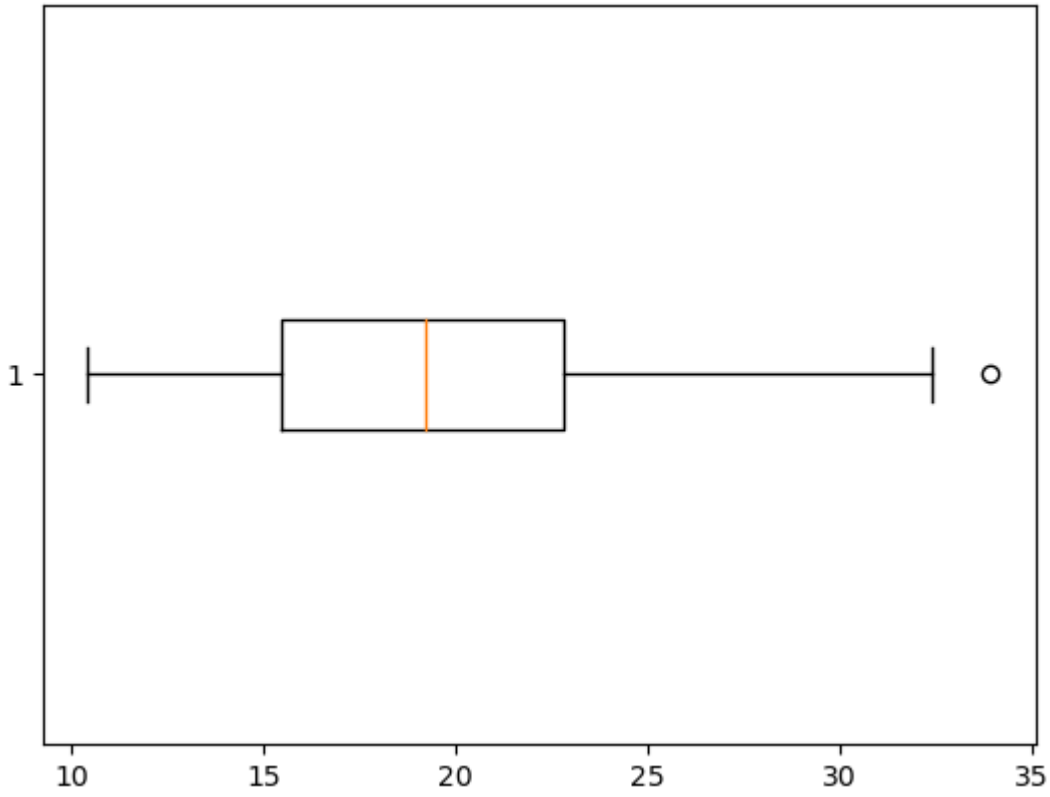
In [32]:
```
#boxplot
plt.boxplot(mtcars['mpg'],vert=True)
#gives median value
```

Out[32]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x19be85cc690>,
  <matplotlib.lines.Line2D at 0x19be85cd110>],
 'caps': [<matplotlib.lines.Line2D at 0x19be85cd3d0>,
  <matplotlib.lines.Line2D at 0x19be73de690>],
 'boxes': [<matplotlib.lines.Line2D at 0x19be85bfb50>],
 'medians': [<matplotlib.lines.Line2D at 0x19be85cf210>],
 'fliers': [<matplotlib.lines.Line2D at 0x19be85bfa10>],
 'means': []}
```



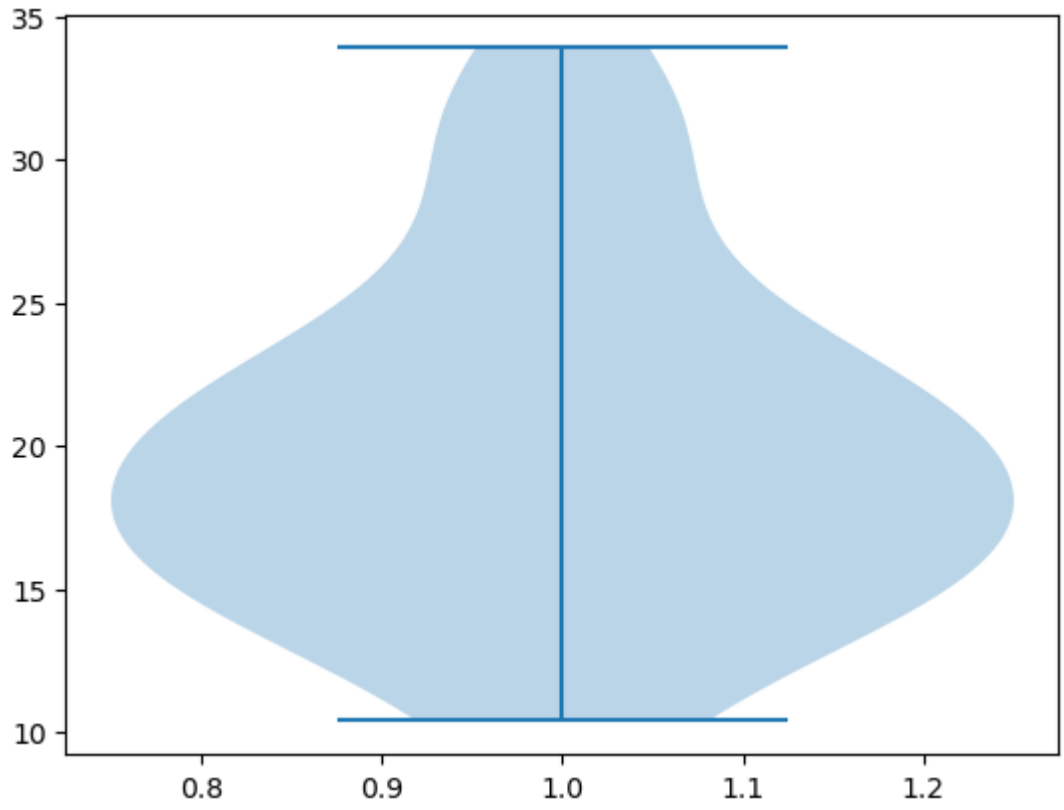
In [33]:
```
plt.boxplot(mtcars['mpg'],vert=False)
```

Out[33]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x19be86227d0>,
  <matplotlib.lines.Line2D at 0x19be86231d0>],
 'caps': [<matplotlib.lines.Line2D at 0x19be8623d90>,
  <matplotlib.lines.Line2D at 0x19be8630990>],
 'boxes': [<matplotlib.lines.Line2D at 0x19be73da810>],
 'medians': [<matplotlib.lines.Line2D at 0x19be8631550>],
 'fliers': [<matplotlib.lines.Line2D at 0x19be86238d0>],
 'means': []}
```



In [34]:
```
#violin plot
plt.violinplot(mtcars['mpg'])
```

Out[34]:
```
{'bodies': [<matplotlib.collections.PolyCollection at 0x19be85f36d0>],
 'cmaxes': <matplotlib.collections.LineCollection at 0x19be8605cd0>,
 'cmins': <matplotlib.collections.LineCollection at 0x19be868e050>,
 'cbars': <matplotlib.collections.LineCollection at 0x19be868f550>}
```



In [35]:
```
import seaborn as sns
```

In [36]:
```
tips =sns.load_dataset('tips')
```

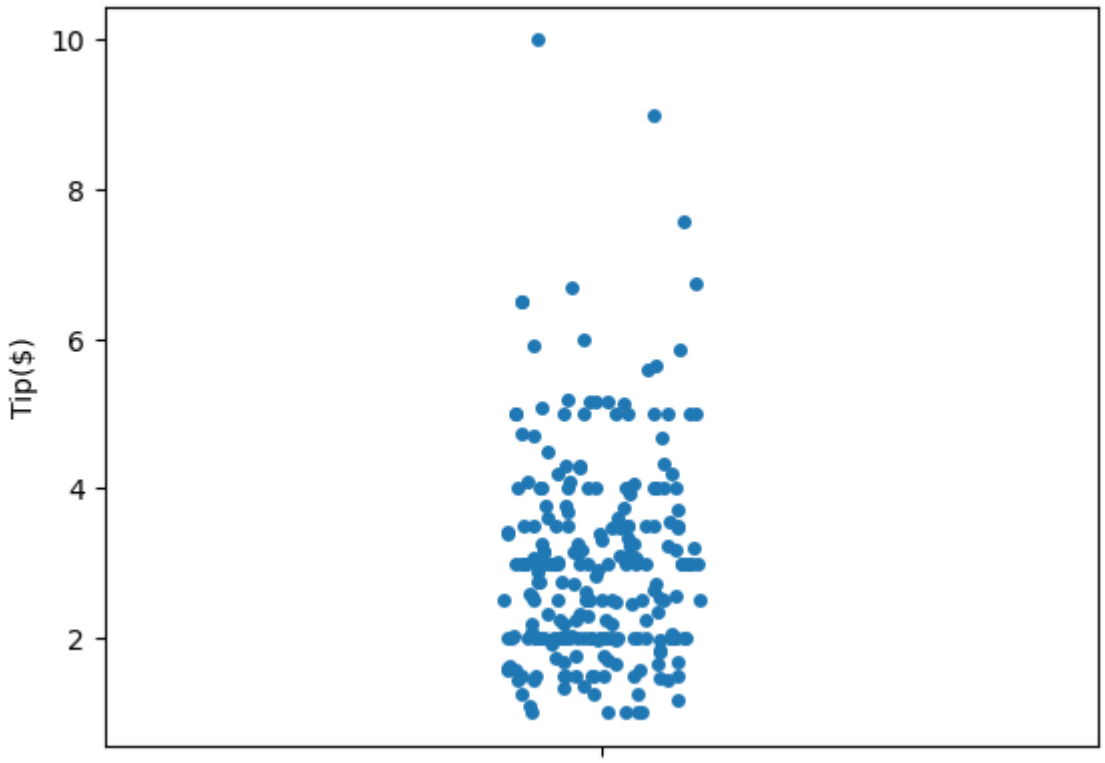
In [37]:
```
tips
```

Out[37]:

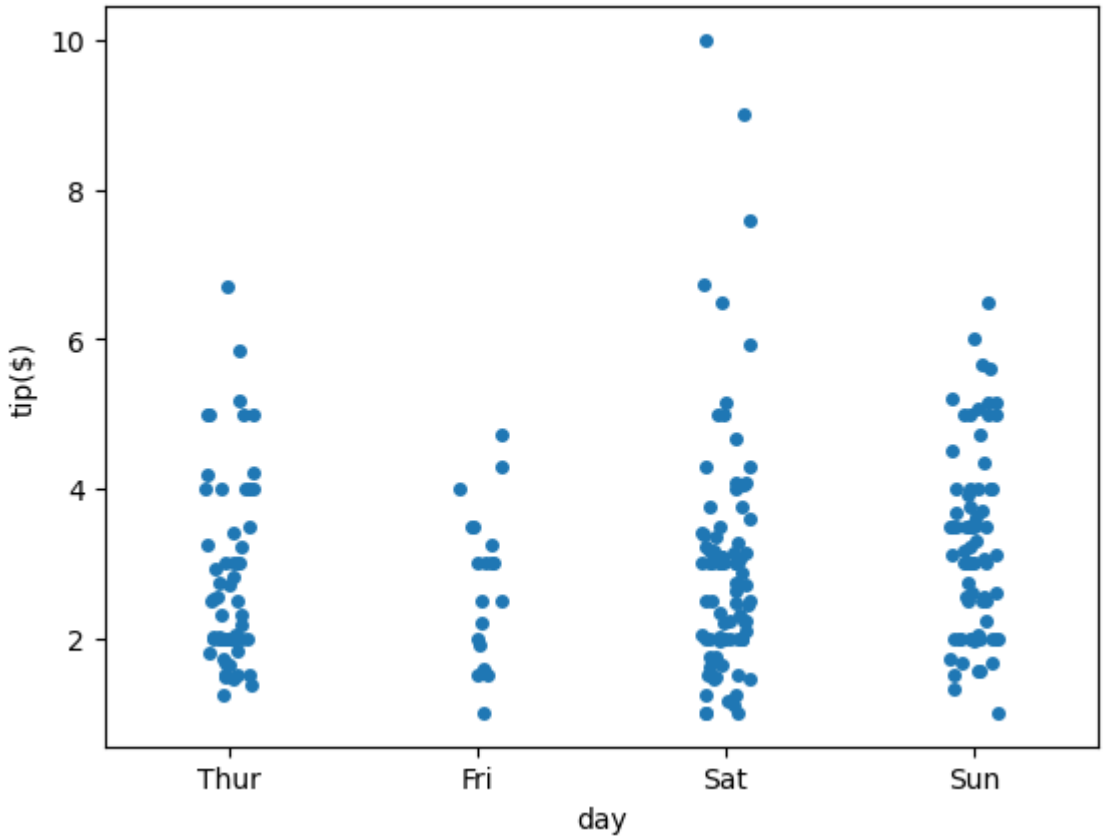
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

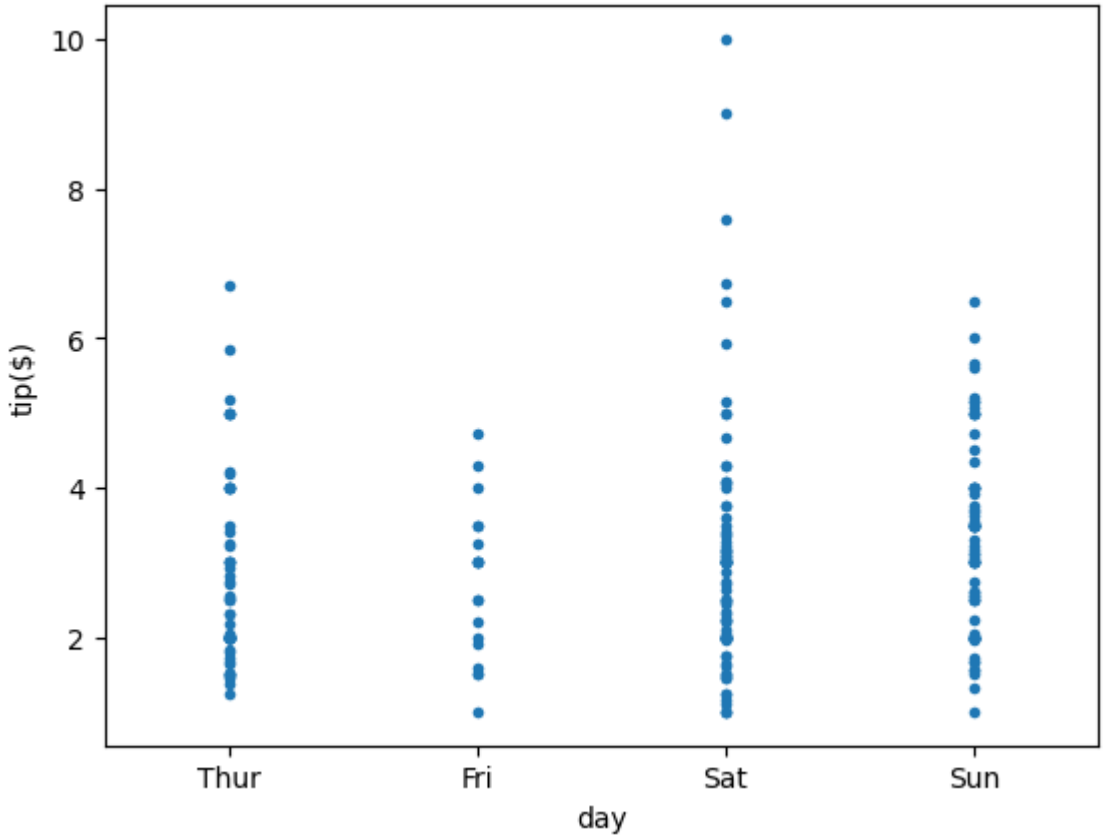
In [44]:
```
#strip plot

sns.stripplot(y='tip', data=tips, jitter=True)
plt.ylabel('Tip($)')
plt.show()
```



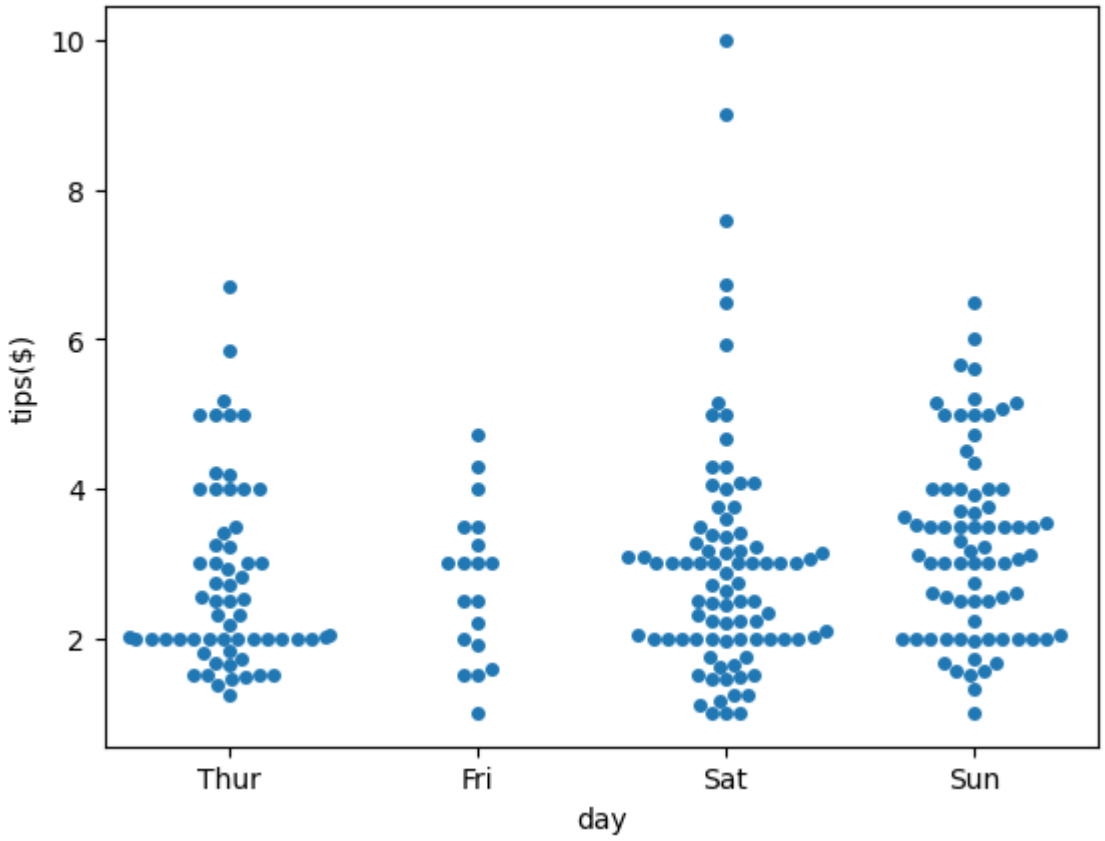
In [45]:
```
#grouping
sns.stripplot(x='day',y='tip',data=tips)
plt.ylabel('tip($)')
plt.show()
```



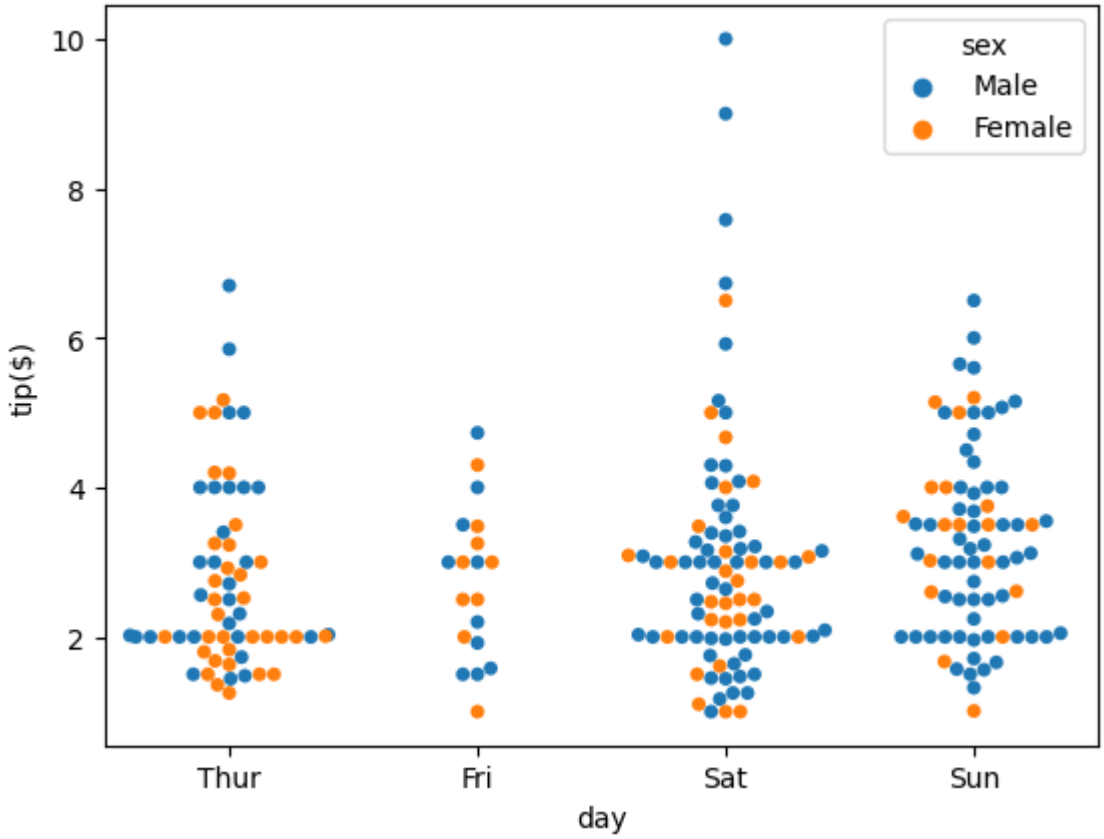
In [46]:
```
sns.stripplot(x='day',y='tip',data=tips,size=4,jitter=False)
plt.ylabel('tip($)')
plt.show()
```



In [47]:
```
#swarm plot

sns.swarmplot(x='day',y='tip',data=tips)
plt.ylabel('tips($)')
plt.show()
```

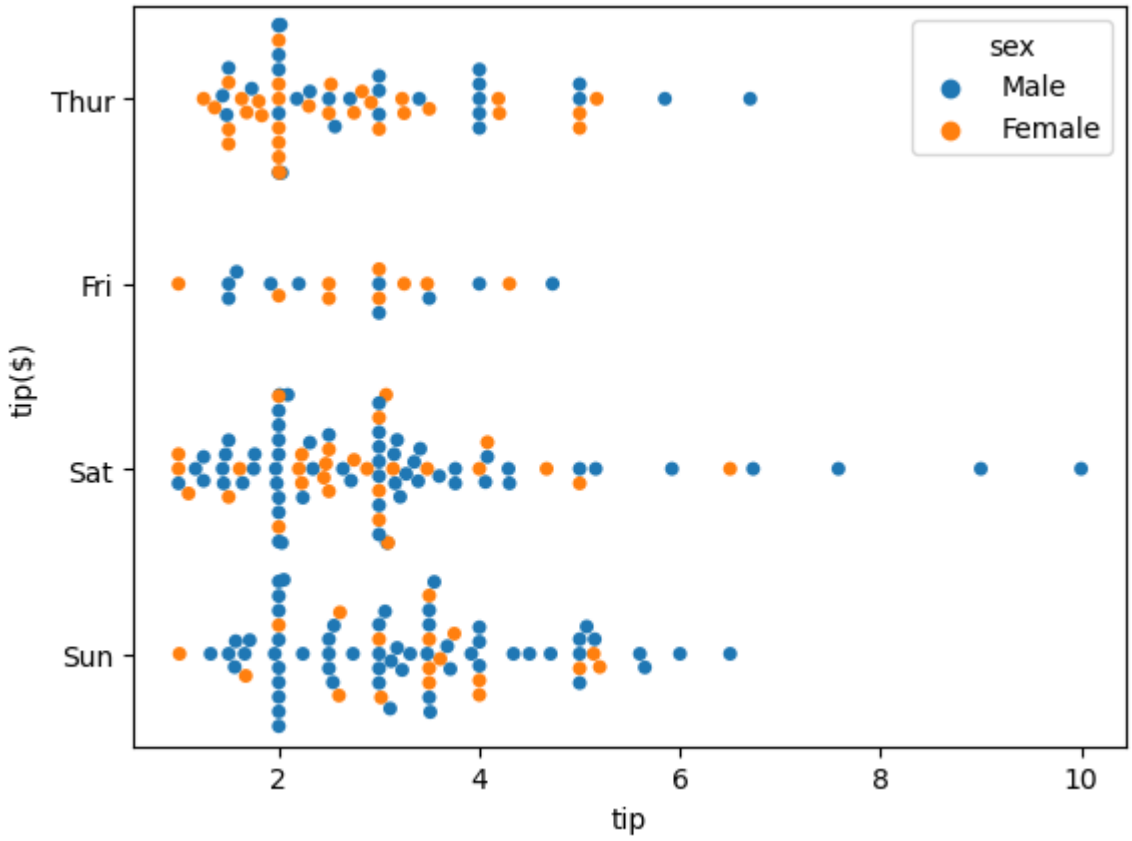


In [48]:
```
sns.swarmplot(x='day',y='tip',data=tips,hue='sex')
plt.ylabel('tip($)')
plt.show()
```



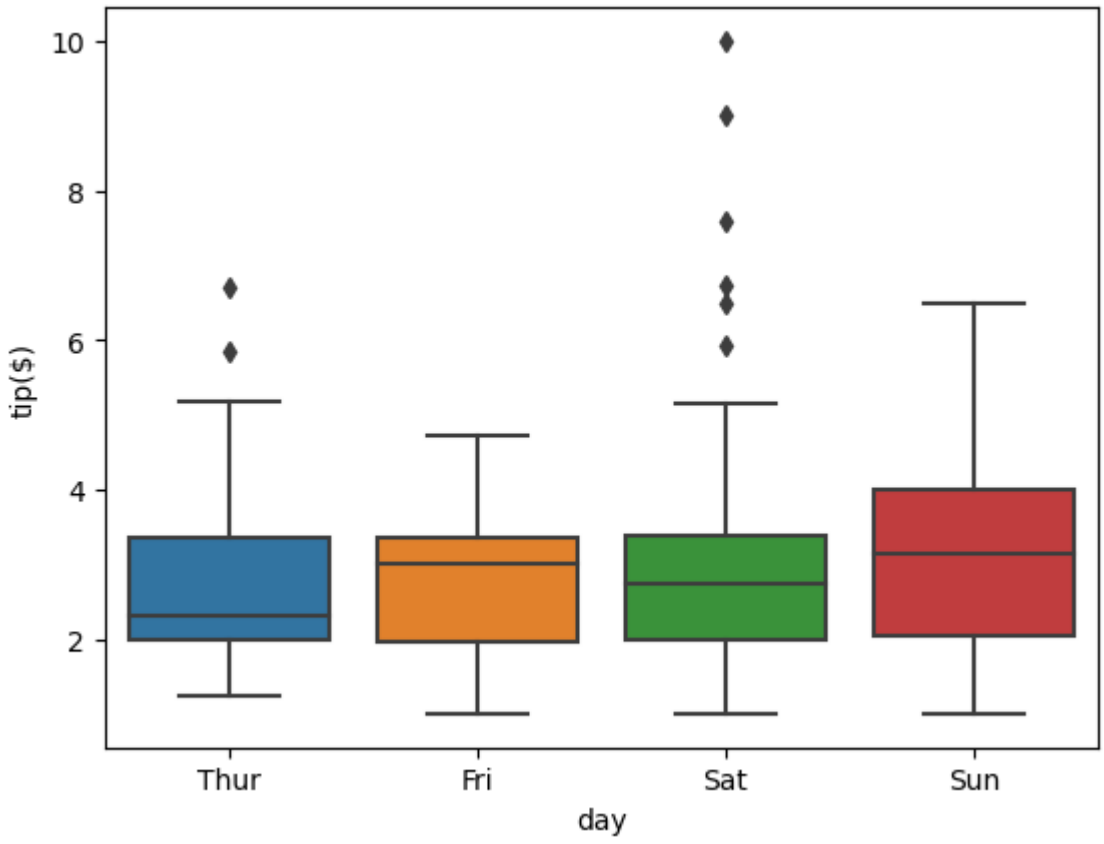
In [50]:
```
#horizontal swarm
sns.swarmplot(x='tip',y='day',data=tips,hue='sex',orient='h')
plt.ylabel('tip($)')
plt.show()
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWa
rning: 8.1% of the points cannot be placed; you may want to decrease the size
of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
c:\Users\HOME\anaconda3\Lib\site-packages\seaborn\categorical.py:3544: UserWa
rning: 6.9% of the points cannot be placed; you may want to decrease the size
of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

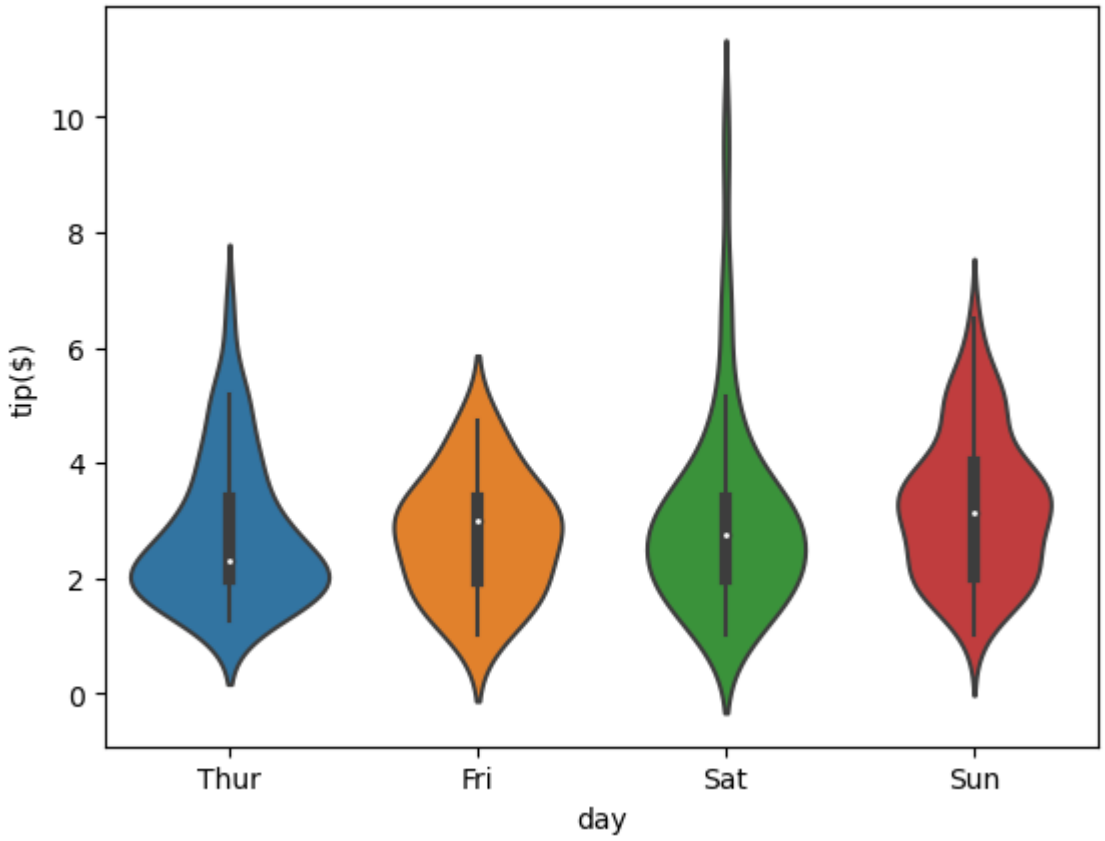


In [52]:
```
#box and violin plot
sns.boxplot(x='day',y='tip',data=tips)
plt.ylabel('tip($)')
#horizontal line are median value
#vertical dots are outliers
```

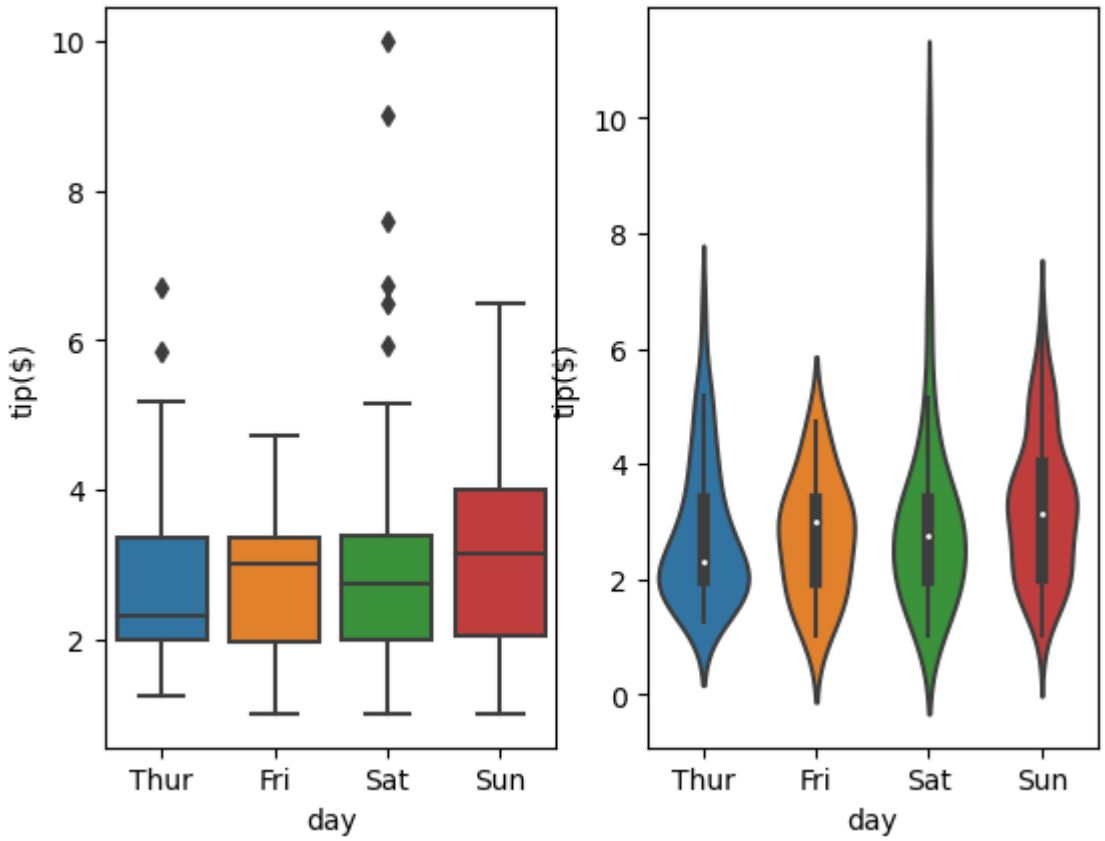
Out[52]:
```
Text(0, 0.5, 'tip($)')
```



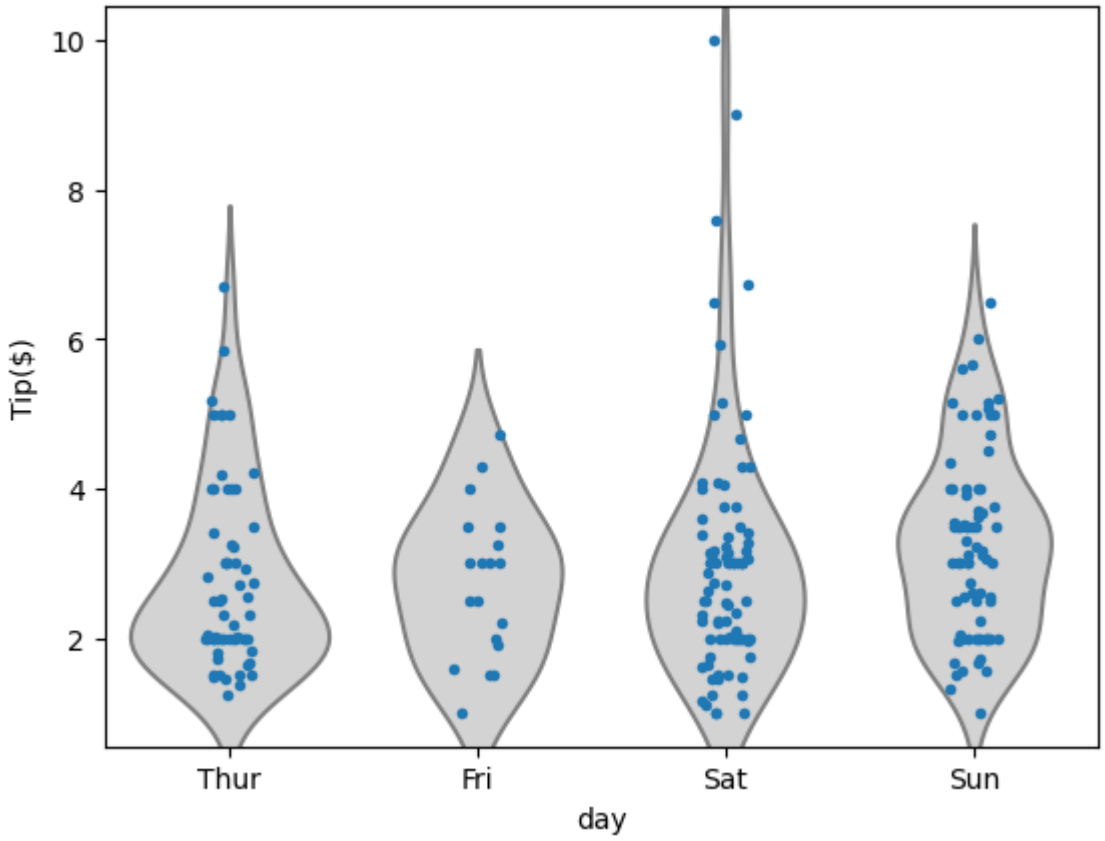
In [57]:
```
#plt.subplot(1,2,2)
sns.violinplot(x='day',y='tip',data=tips)
plt.ylabel('tip($)')
plt.show()
```



In [59]:
```
plt.subplot(1,2,1)
sns.boxplot(x='day',y='tip',data=tips)
plt.ylabel('tip($)')
plt.subplot(1,2,2)
sns.violinplot(x='day',y='tip',data=tips)
plt.ylabel('tip($)')

plt.show()
```

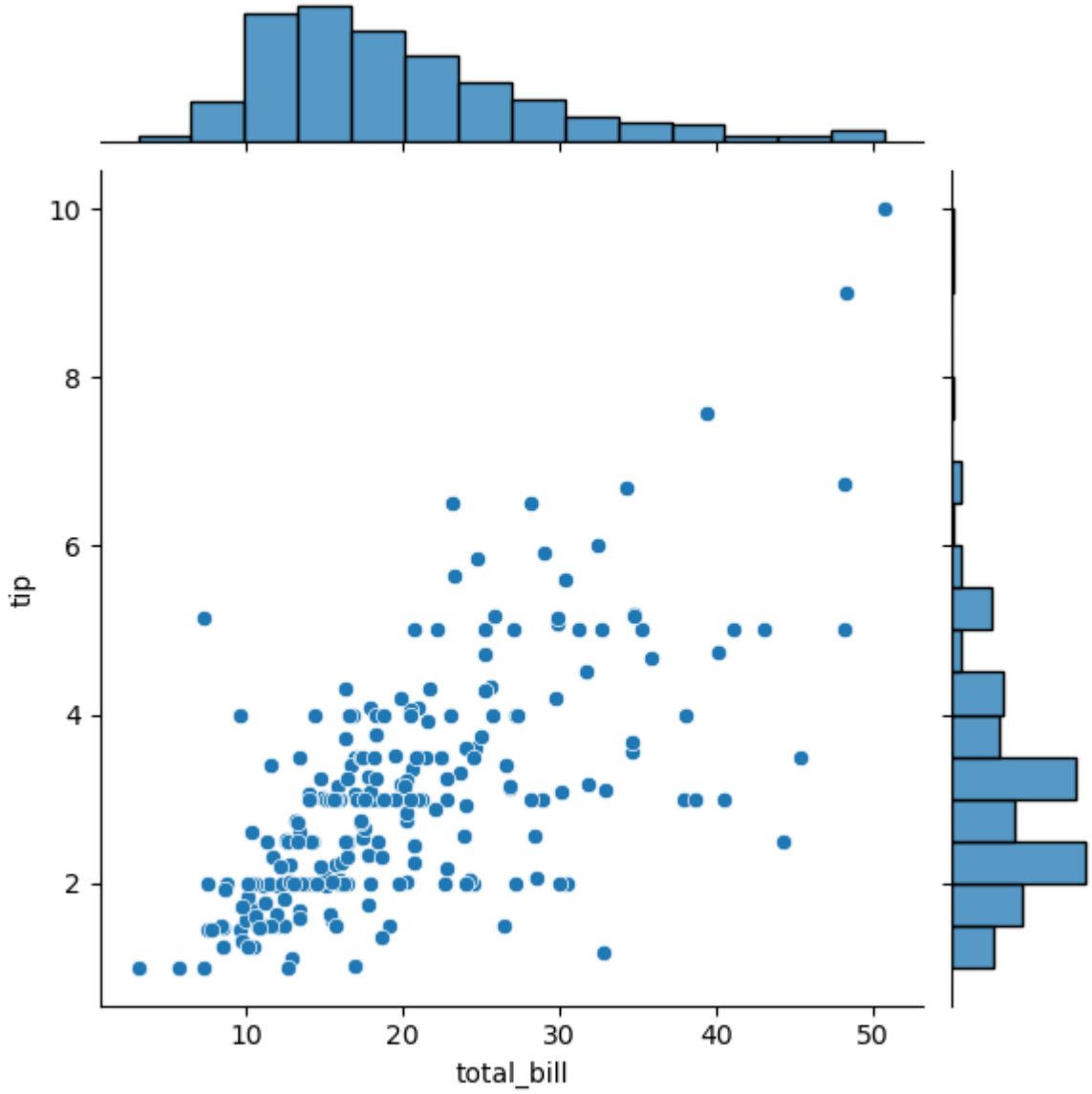


In [63]:
```
#combining plots

sns.violinplot(x='day',y='tip',data=tips,inner=None,color='lightgray')
sns.stripplot(x='day',y='tip',data=tips,size=4,jitter=True)
plt.ylabel('Tip($)')
plt.show()
```

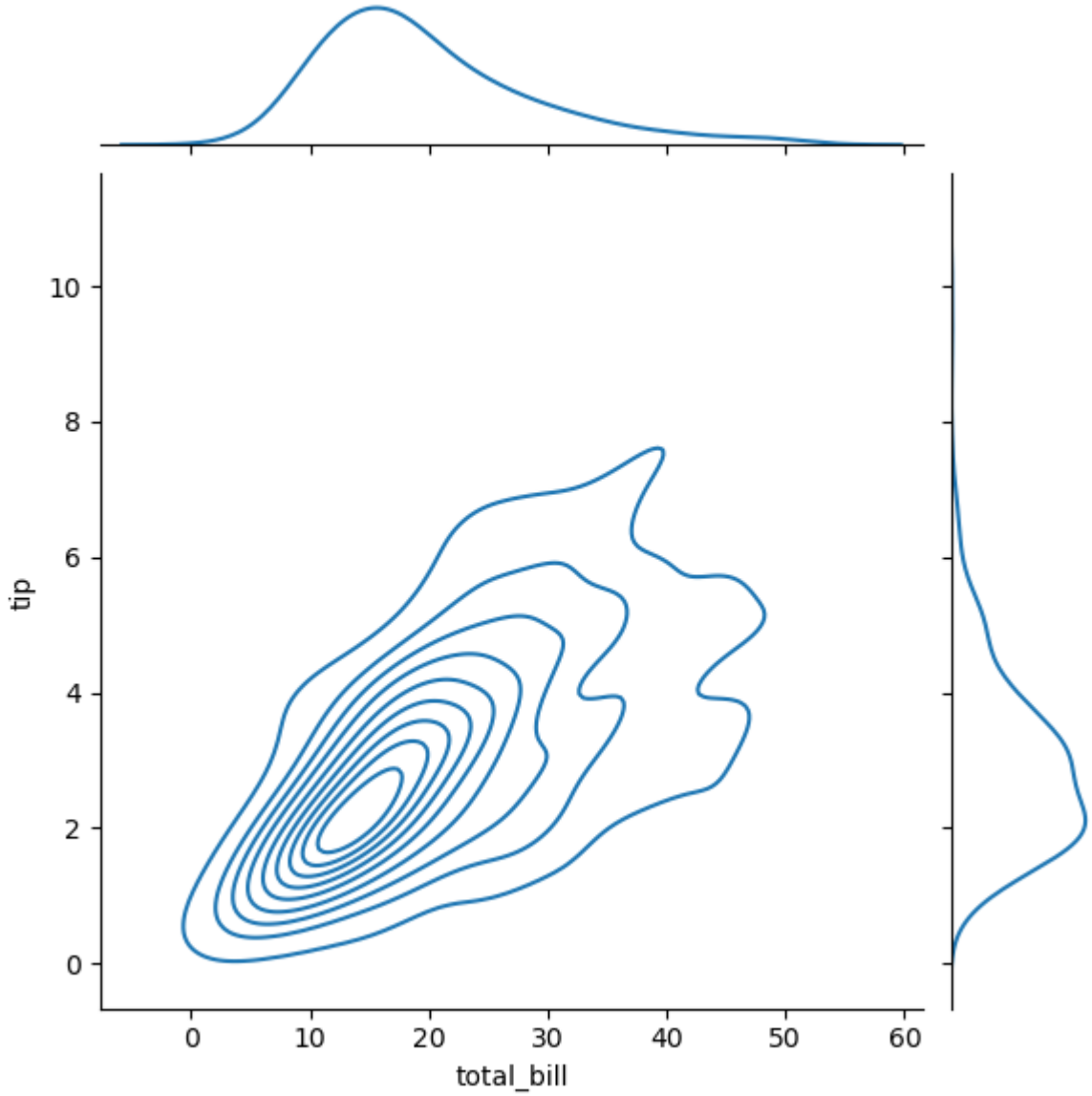


bivariate

In [64]:
```
#Joint plot
sns.jointplot(x='total_bill',y='tip',data=tips)
plt.show()
```

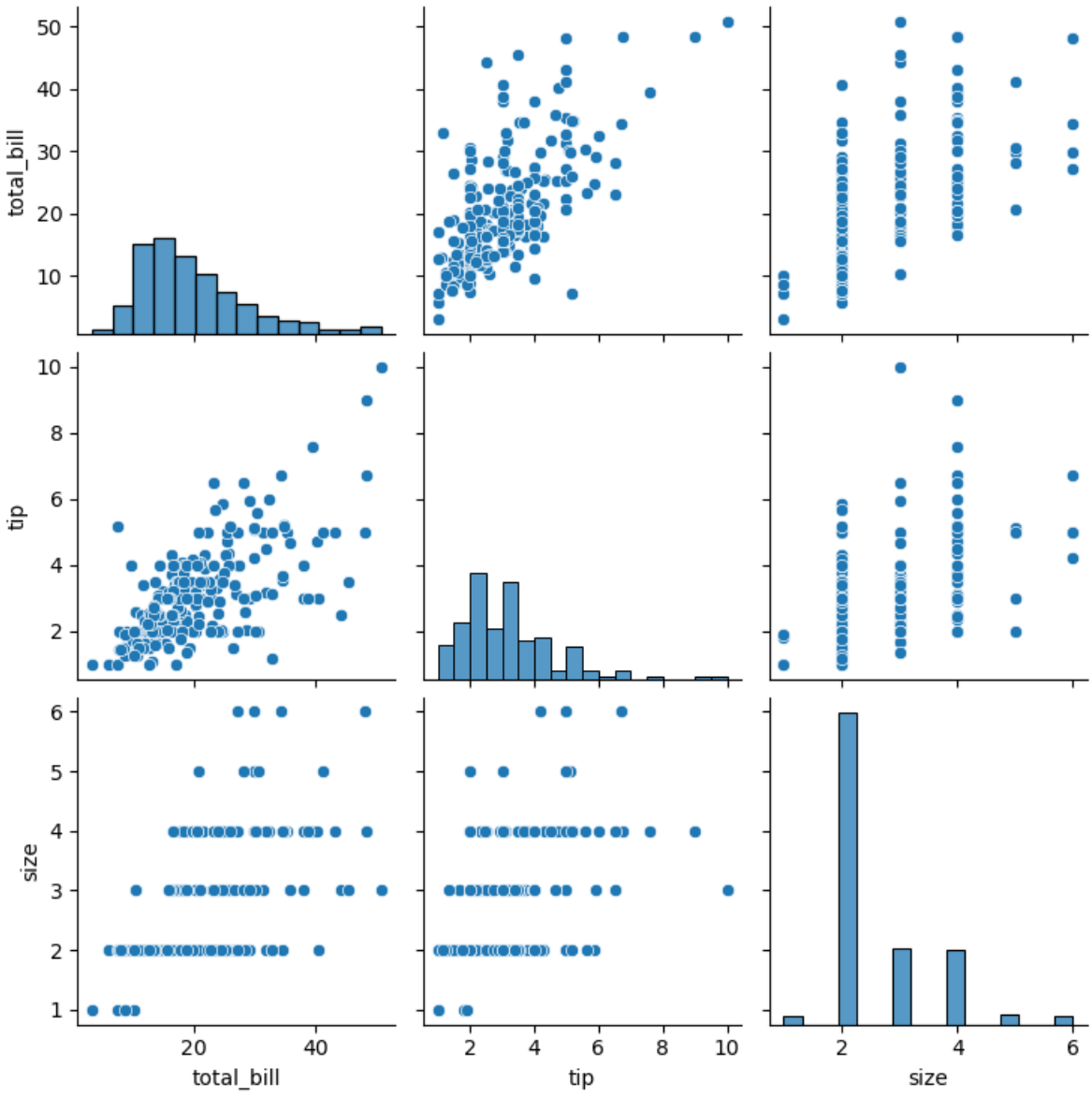


In [69]:
```
#density plot
sns.jointplot(x='total_bill',y='tip',data=tips,kind='kde')
plt.show()
```



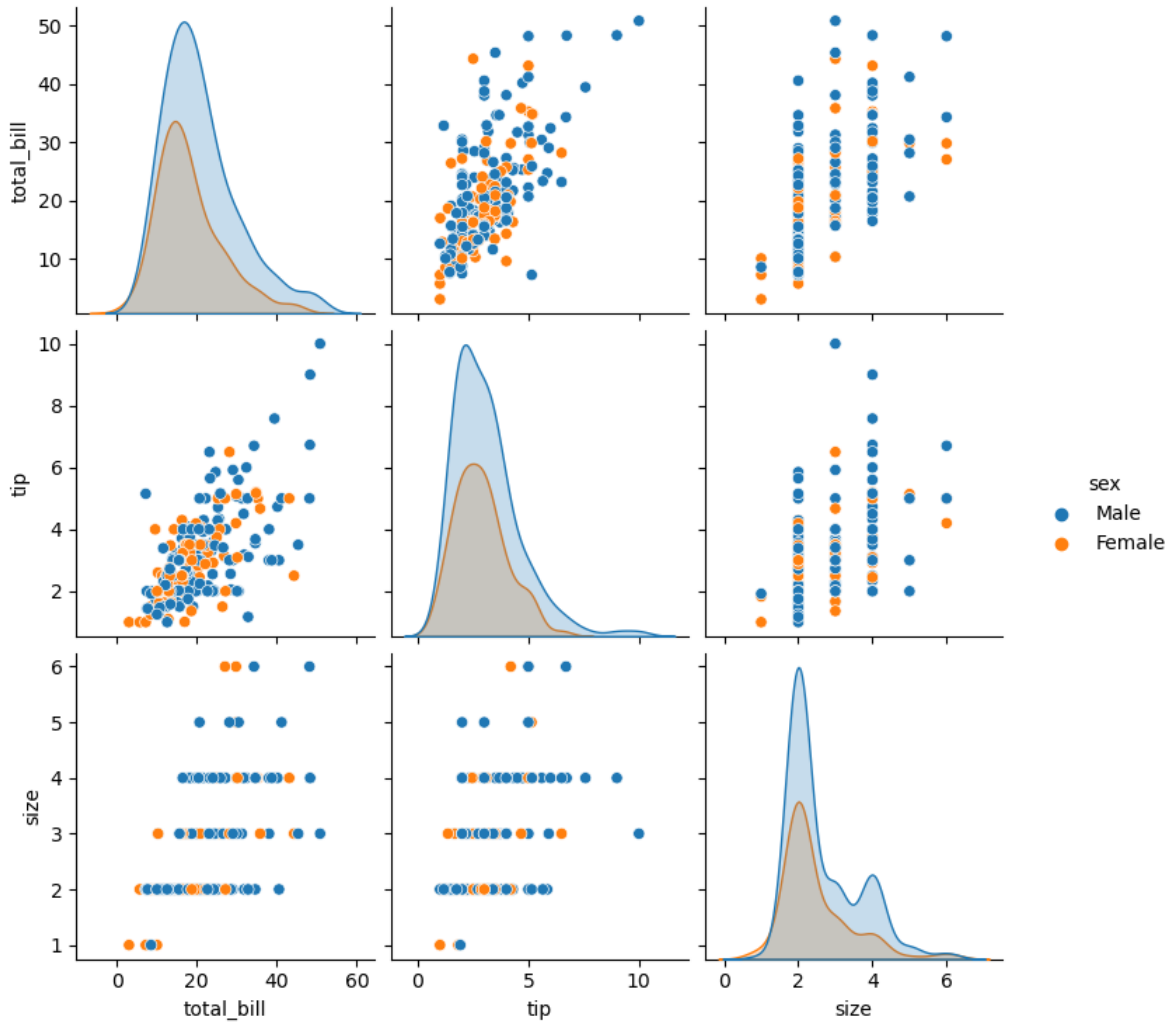
In [72]:
```
#pair plot

sns.pairplot(tips)
plt.show()
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarnin
g: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [73]:
```
sns.pairplot(tips,hue='sex')
plt.show()
```

```
c:\Users\HOME\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarnin
g: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Train Data Visualization

In [74]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

In [75]:
```
data=pd.read_csv(r'C:\Users\HOME\Desktop\DSA\Lab5\train.csv')
data.head()
```

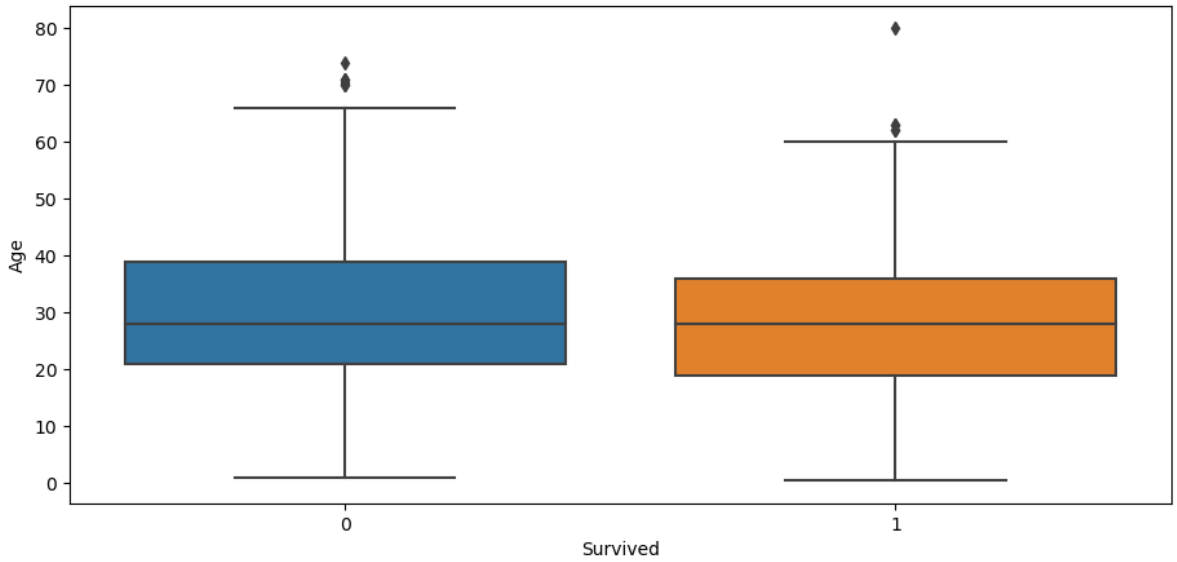
Out[75]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
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	N
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	N

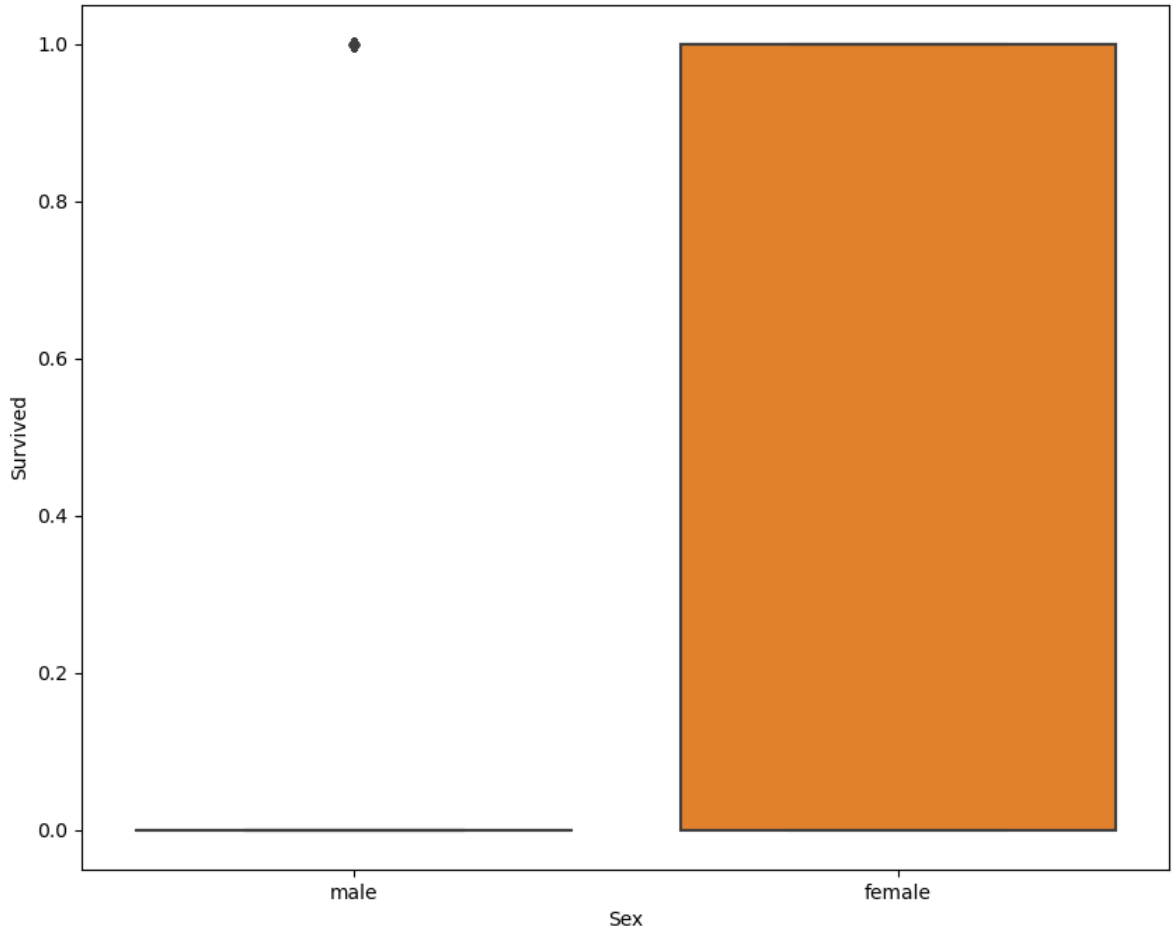
In [76]:
```
data.groupby('Survived')['PassengerId'].count()
```

Out[76]:
```
Survived
0    549
1    342
Name: PassengerId, dtype: int64
```

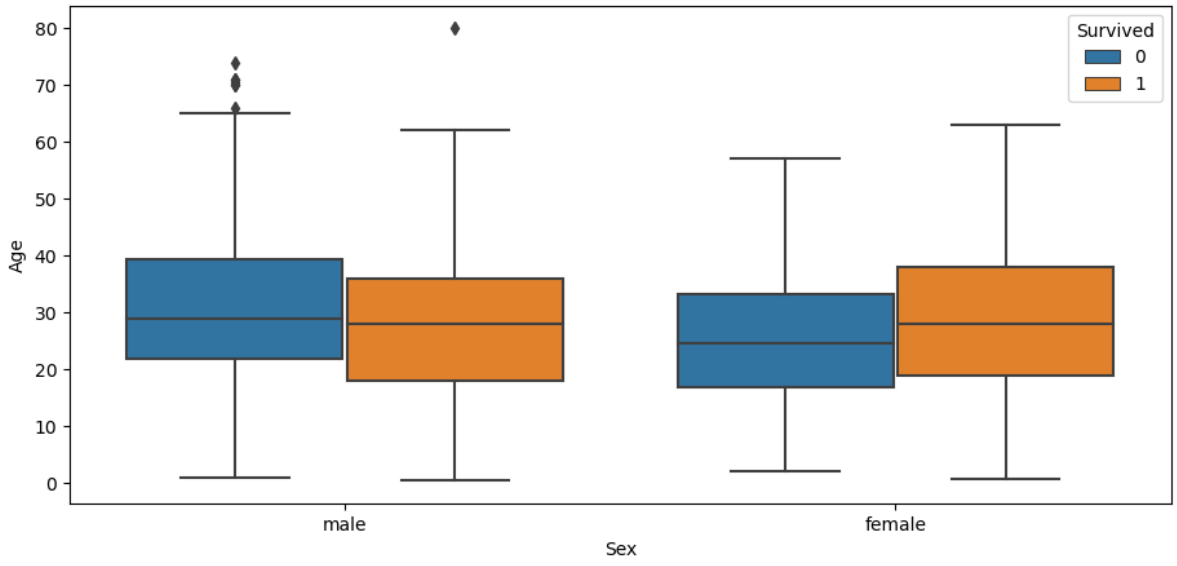
In [77]:
```
f,ax=plt.subplots(figsize=(11,5))
sns.boxplot(x='Survived',y='Age',data=data);
```



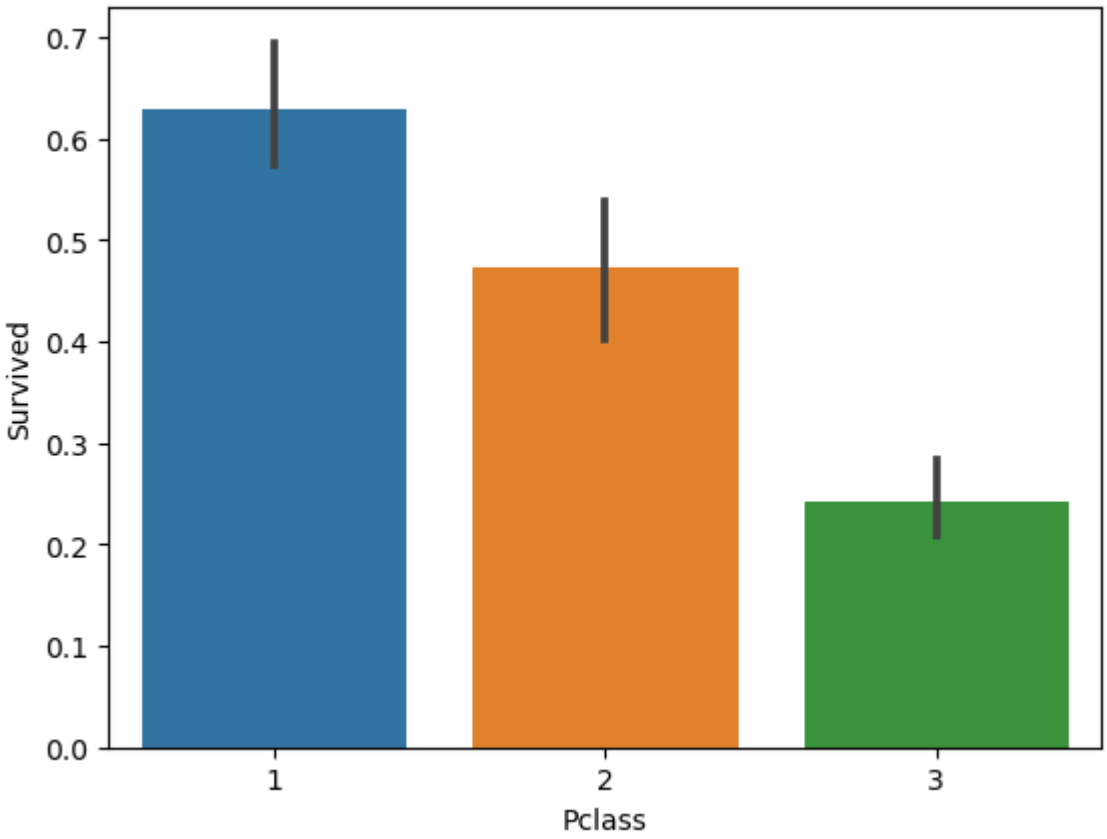
In [83]:
```
f,ax=plt.subplots(figsize=(10,8))
sns.boxplot(x='Sex',y='Survived',data=data);
```



In [80]:
```
f,ax=plt.subplots(figsize=(11,5))
sns.boxplot(x='Sex',y='Age',hue='Survived', data=data);
```

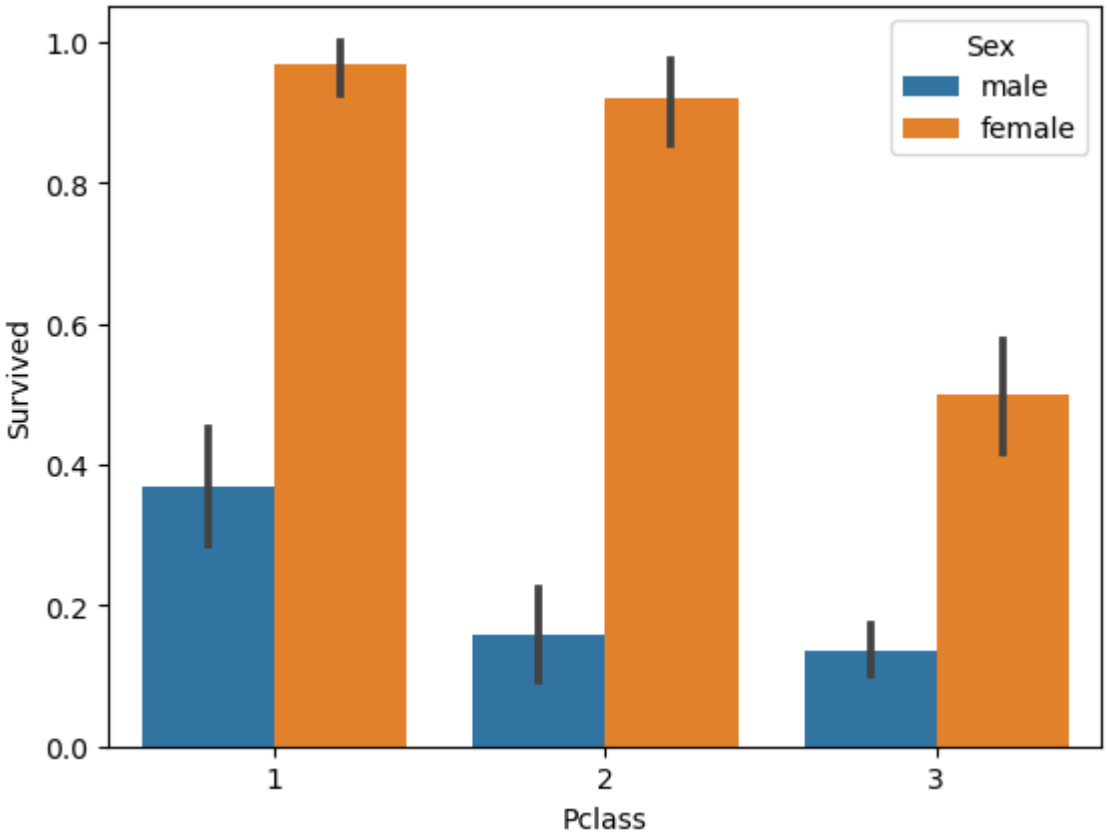


In [81]:
```
sns.barplot(x='Pclass',y='Survived',data=data);
```

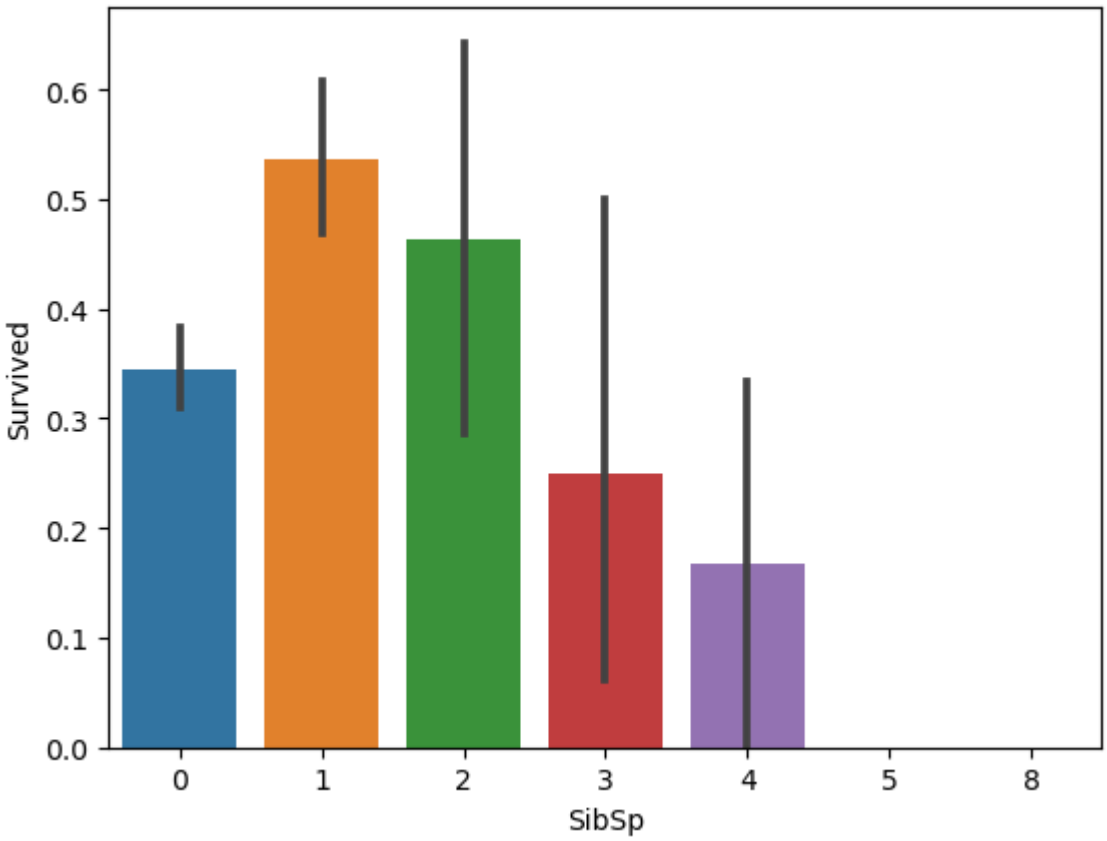


In [82]:
```
sns.barplot(x='Pclass',y='Survived',hue='Sex',data=data)
```

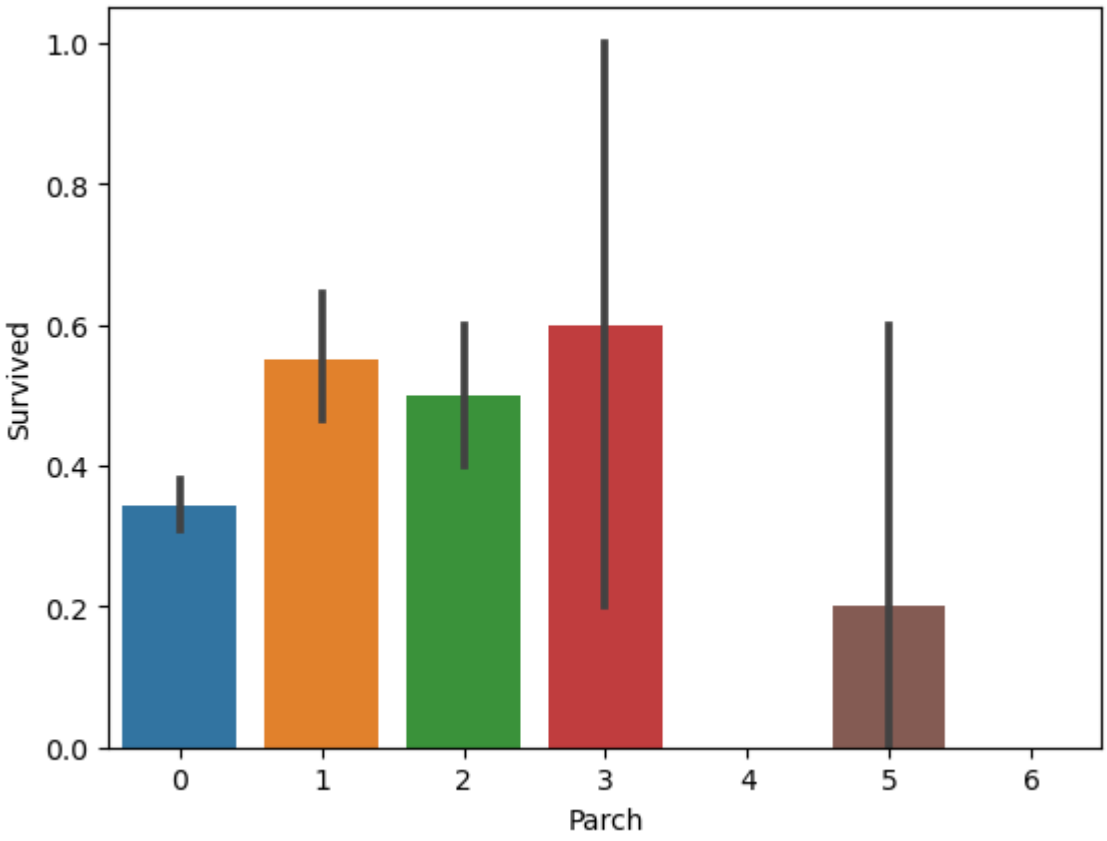
Out[82]:
```
<Axes: xlabel='Pclass', ylabel='Survived'>
```



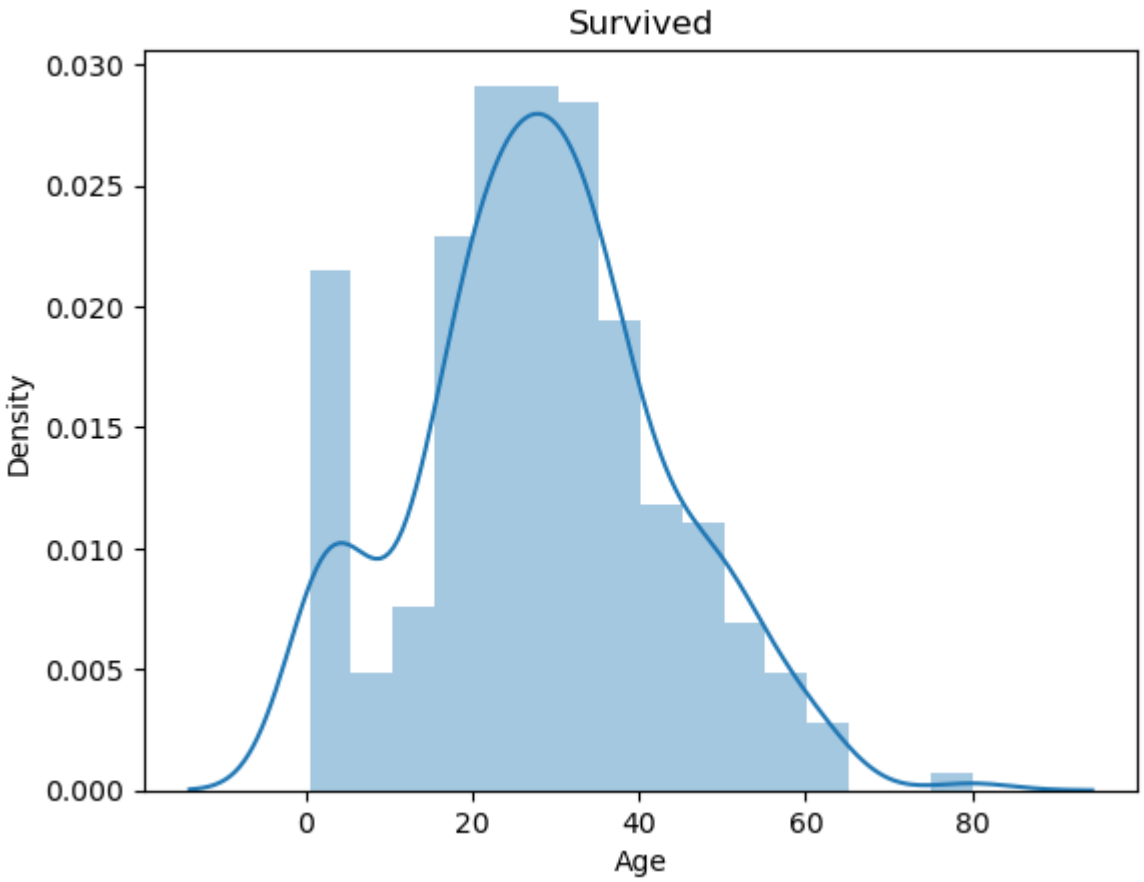
In [85]:
```
sns.barplot(x='SibSp',y='Survived',data=data);
```



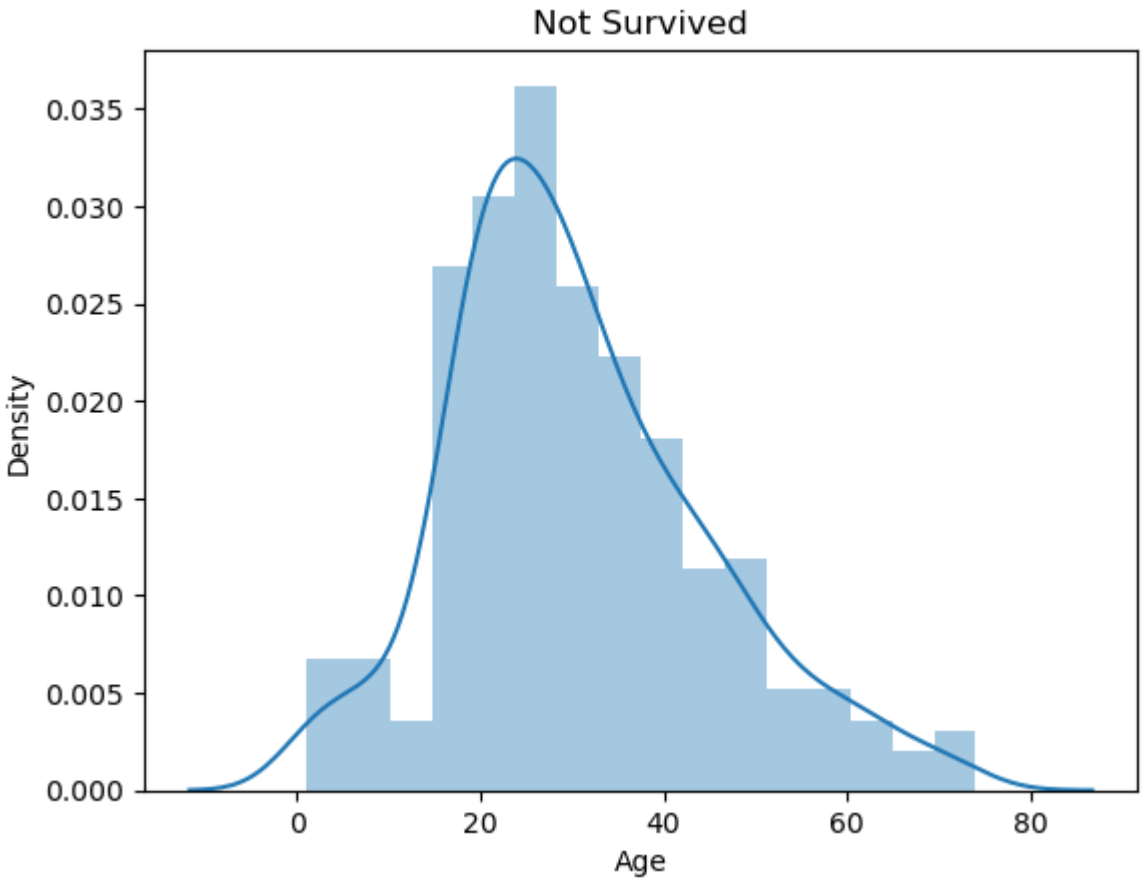
In [86]:
```
sns.barplot(x='Parch',y='Survived',data=data);
```



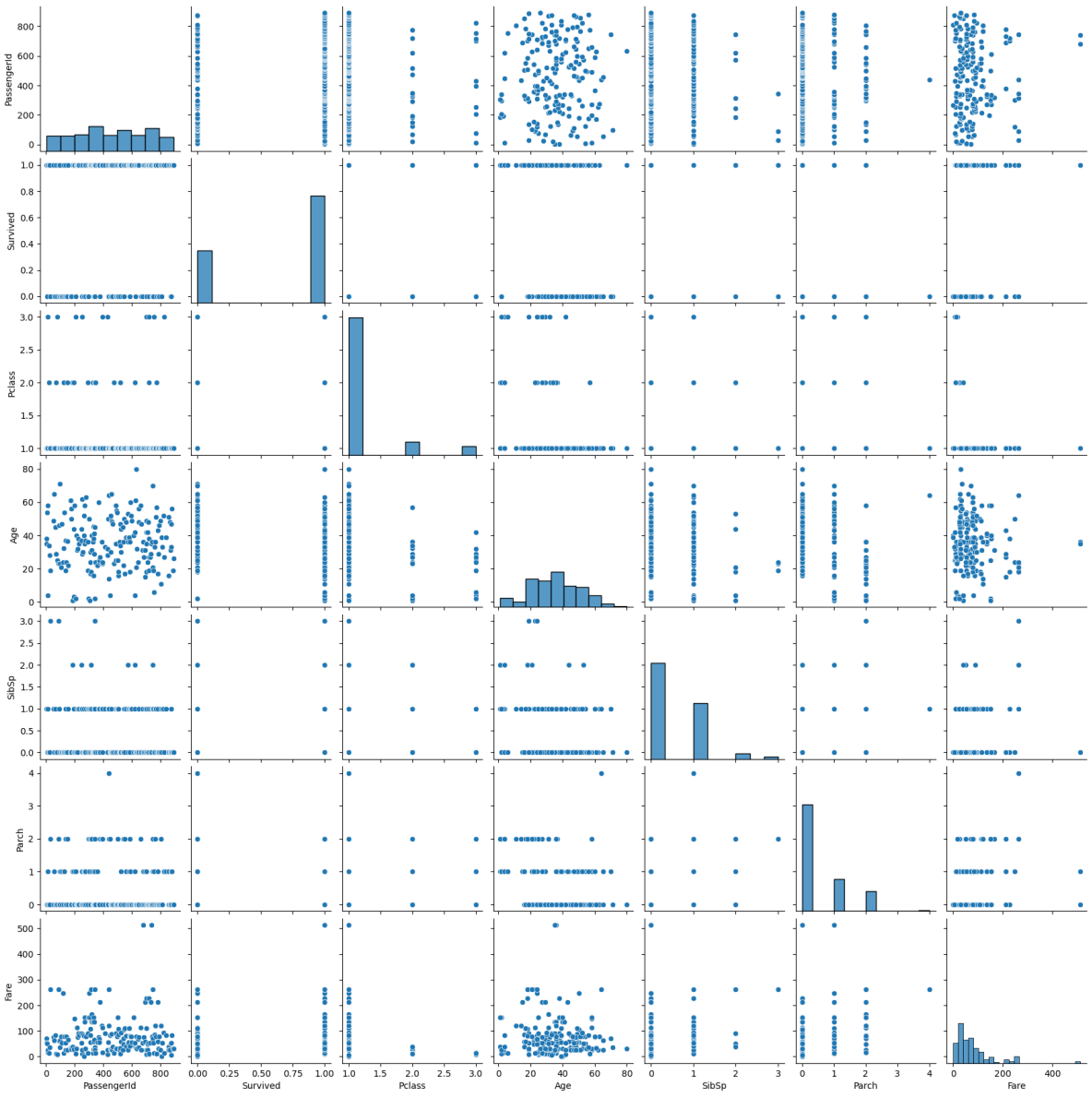
In [87]:
```
survived = data.loc[data['Survived']==1,'Age'].dropna()
sns.distplot(survived)
plt.title('Survived');
```



In [89]:
```
not_survived= data.loc[data['Survived']==0,'Age'].dropna()
sns.distplot(not_survived)
plt.title('Not Survived');
```

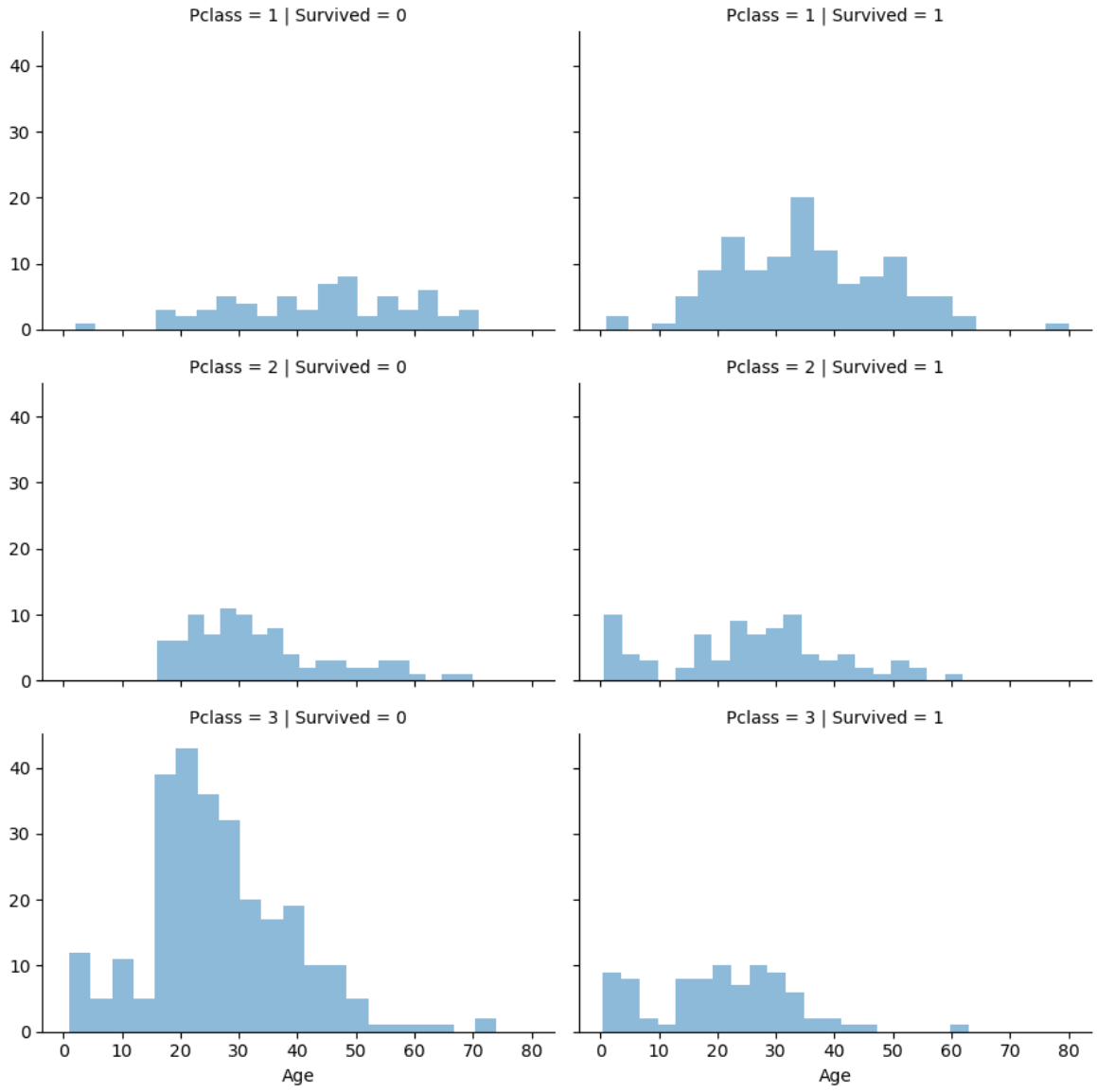


In [90]:
```
sns.pairplot(data.dropna());
```

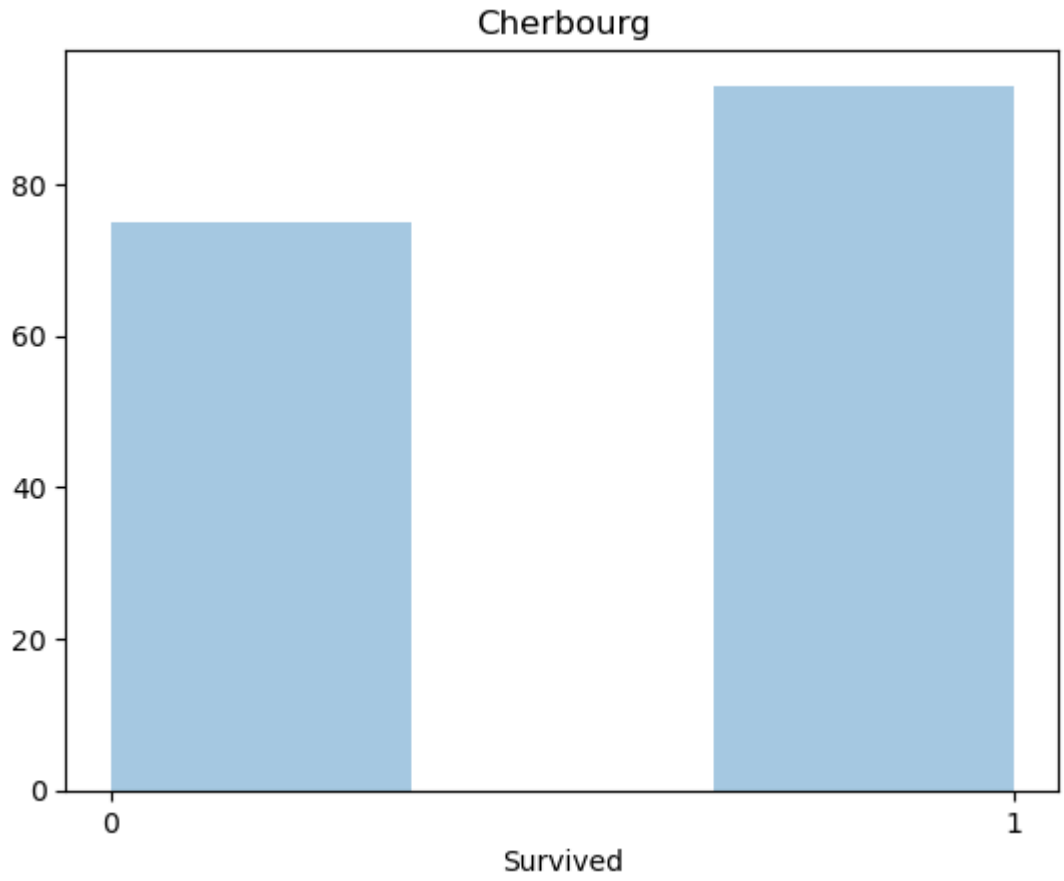


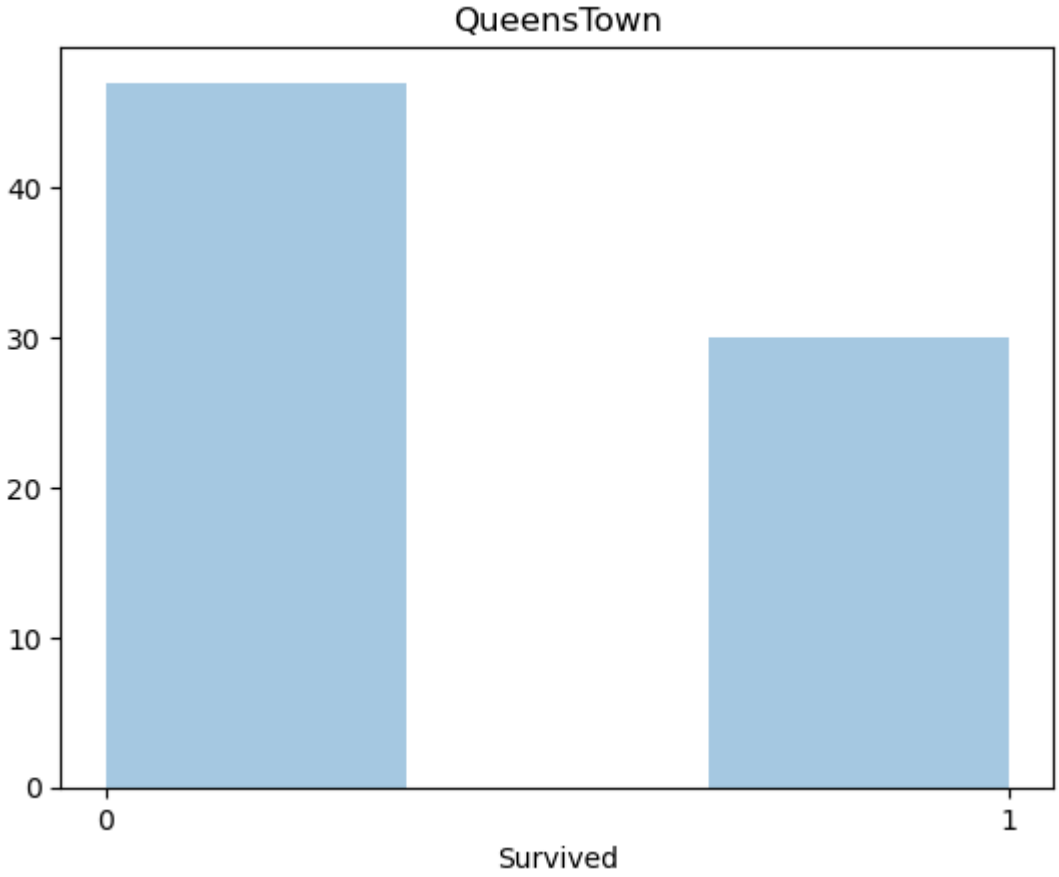
In [103]:
```
#pclass vs survived
grid = sns.FacetGrid(data, col='Survived', row='Pclass', aspect=1.5)
grid.map(plt.hist,'Age',alpha=.5,bins=20)
grid.add_legend()
```

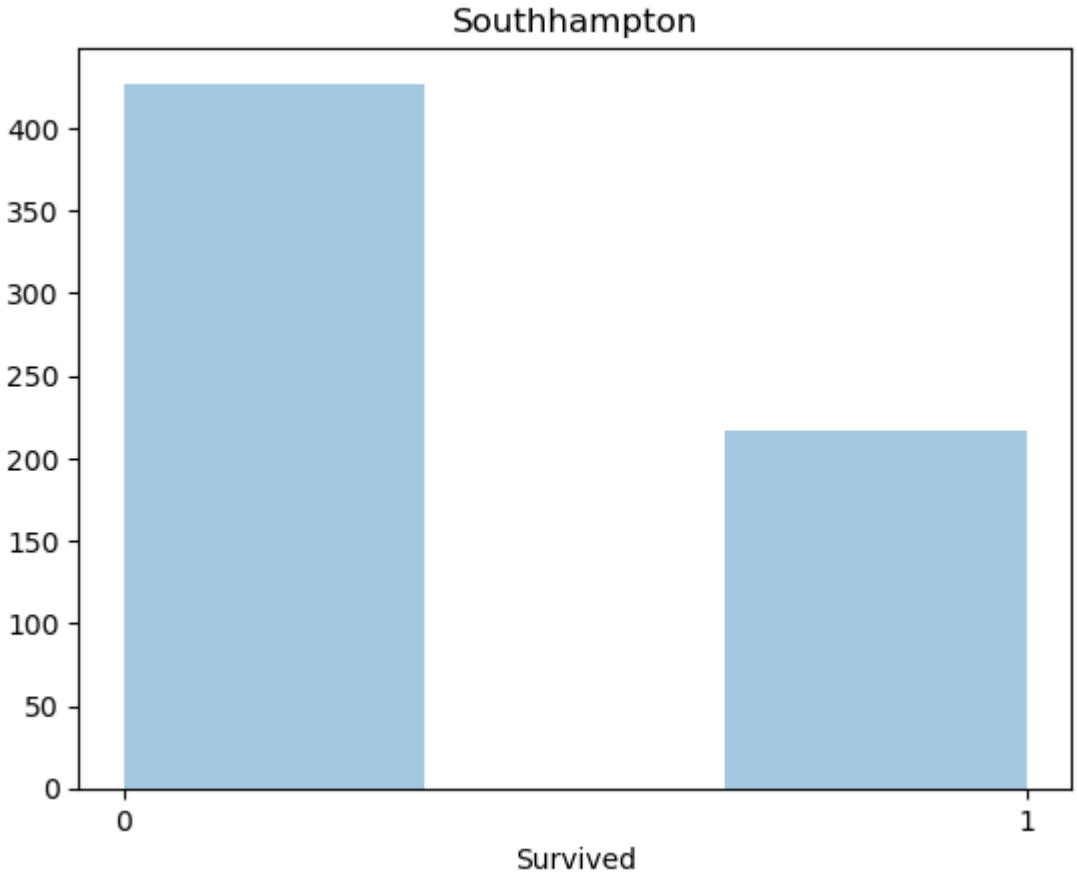
Out[103]: <seaborn.axisgrid.FacetGrid at 0x19bf8da2650>



In [104]:
```
sns.distplot(a=data[data['Embarked']=='C']['Survived'],bins=3,kde=False)
plt.title('Cherbourg')
plt.xticks([0,1])
plt.show()

plt.title('QueensTown')
sns.distplot(a=data[data['Embarked']=='Q']['Survived'],bins=3,kde=False)
plt.xticks([0,1])
plt.show()

plt.title('Southhampton')
sns.distplot(a=data[data['Embarked']=='S']['Survived'],bins=3,kde=False)
plt.xticks([0,1])
plt.show()
```




QueensTown
40
30
20
10
0
0
1
Survived


Southhampton
400
350
300
250
200
150
100
50
0
0
1
Survived

In [105]:
```
figbi,axesbi = plt.subplots(2,4,figsize=(16,10))
data.groupby('Pclass')['Survived'].mean().plot(kind='barh',ax=axesbi[0,0],xlim
data.groupby('SibSp')['Survived'].mean().plot(kind='barh',ax=axesbi[0,1],xlim=
data.groupby('Parch')['Survived'].mean().plot(kind='barh',ax=axesbi[0,2],xlim=
data.groupby('Sex')['Survived'].mean().plot(kind='barh',ax=axesbi[0,3],xlim=[0
data.groupby('Embarked')['Survived'].mean().plot(kind='barh',ax=axesbi[1,0],xl
sns.boxplot(x='Survived',y="Age",data=data,ax=axesbi[1,1])
sns.boxplot(x='Survived',y='Fare',data=data,ax=axesbi[1,2])
```

Out[105]: <Axes: xlabel='Survived', ylabel='Fare'>