Darshan UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Lab - 3

Name: Harmik Rathod

Enrollment No: 24010101680

1) First, you need to read the titanic dataset from local disk and display first five records

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
data=pd.read_csv("titanic.csv")
```

In [4]:
```
data.head(5)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
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
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

2) Identify Nominal, Ordinal, Binary and Numeric attributes from data sets and display all values.

In [5]:
```
nominal=['Name','Cabin','Embarked','Ticket','Sex']
ordinal=['Pclass']
binary=['Survived','Sex']
numerical=['PassengerId','Age','Fare','Parch','SibSp']

print("Nominal: ", nominal)
print("ordinal: ", ordinal)
print("Binary: ", binary)
print("Numerical: ", numerical)
```

```
Nominal:  ['Name', 'Cabin', 'Embarked', 'Ticket', 'Sex']
ordinal:  ['Pclass']
Binary:  ['Survived', 'Sex']
Numerical:  ['PassengerId', 'Age', 'Fare', 'Parch', 'SibSp']
```

3) Identify symmetric and asymmetric binary attributes from data sets and display all values.

In [21]:
```
symmetricAttr=data.Sex
asymmetricAttr=data.Survived

print('count symmetricAttr',symmetricAttr.value_counts())
```

```
print('\ncount asymmetricAttr',asymmetricAttr.value_counts())


print('\nsymmetricAttr\n',symmetricAttr)
print('\nasymmetricAttr\n',asymmetricAttr)
```

```
count symmetricAttr Sex
male      577
female    314
Name: count, dtype: int64

count asymmetricAttr Survived
0    549
1    342
Name: count, dtype: int64

symmetricAttr
 0        male
1      female
2      female
3      female
4        male
        ...
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object

asymmetricAttr
 0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

4) For each quantitative attribute, calculate its average, standard deviation, minimum, mode, range and maximum values.

In [52]:
```
numerical=['PassengerId','Age','Fare','Parch','SibSp']

for i in numerical:
    print("::::::",i,"::::::");
    print("Mean: ",data[i].mean())
    print("Standard Deviation: ",data[i].std())
    print("Minimum: ", data[i].min())
```

```
    print("Maxmum: ", data[i].max())
    print("Mode: ", data[i].mode())
    print("Range: ", (data[i].max() - data[i].min() + 1))
    print()
```

```
::::: PassengerId :::::
Mean:  446.0
Standard Deviation:  257.3538420152301
Minimum:  1
Maxmum:  891
Mode:  0        1
1        2
2        3
3        4
4        5
      ...
886    887
887    888
888    889
889    890
890    891
Name: PassengerId, Length: 891, dtype: int64
Range:  891

::::: Age :::::
Mean:  29.69911764705882
Standard Deviation:  14.526497332334044
Minimum:  0.42
Maxmum:  80.0
Mode:  0    24.0
Name: Age, dtype: float64
Range:  80.58

::::: Fare :::::
Mean:  32.204207968574636
Standard Deviation:  49.693428597180905
Minimum:  0.0
Maxmum:  512.3292
Mode:  0    8.05
Name: Fare, dtype: float64
Range:  513.3292

::::: Parch :::::
Mean:  0.38159371492704824
Standard Deviation:  0.8060572211299559
Minimum:  0
Maxmum:  6
Mode:  0    0
Name: Parch, dtype: int64
Range:  7

::::: SibSp :::::
Mean:  0.5230078563411896
Standard Deviation:  1.1027434322934275
Minimum:  0
Maxmum:  8
Mode:  0    0
Name: SibSp, dtype: int64
Range:  9
```

In [ ]:

6) For the qualitative attribute (class), count the frequency for each of its distinct values.

In [54]:
```
print("Distinct Value Pclass: ", data['Pclass'].value_counts())
```

```
Distinct Value Pclass:  Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

7) It is also possible to display the summary for all the attributes simultaneously in a table using the describe() function. If an attribute is quantitative, it will display its mean, standard deviation and various quantiles (including minimum, median, and maximum) values. If an attribute is qualitative, it will display its number of unique values and the top (most frequent) values.

In [64]:
```
data.describe(include=object)
```

Out[64]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	B96 B98	S
freq	1	577	7	4	644

8) For multivariate statistics, you can compute the covariance and correlation between pairs of attributes.

In [66]:
```
data.cov(numeric_only=True)
```

Out[66]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
PassengerId	66231.000000	-0.626966	-7.561798	138.696504	-16.325843	-0.342697	161.8
Survived	-0.626966	0.236772	-0.137703	-0.551296	-0.018954	0.032017	6.2
Pclass	-7.561798	-0.137703	0.699015	-4.496004	0.076599	0.012429	-22.8
Age	138.696504	-0.551296	-4.496004	211.019125	-4.163334	-2.344191	73.8
SibSp	-16.325843	-0.018954	0.076599	-4.163334	1.216043	0.368739	8.7
Parch	-0.342697	0.032017	0.012429	-2.344191	0.368739	0.649728	8.6
Fare	161.883369	6.221787	-22.830196	73.849030	8.748734	8.661052	2469.4

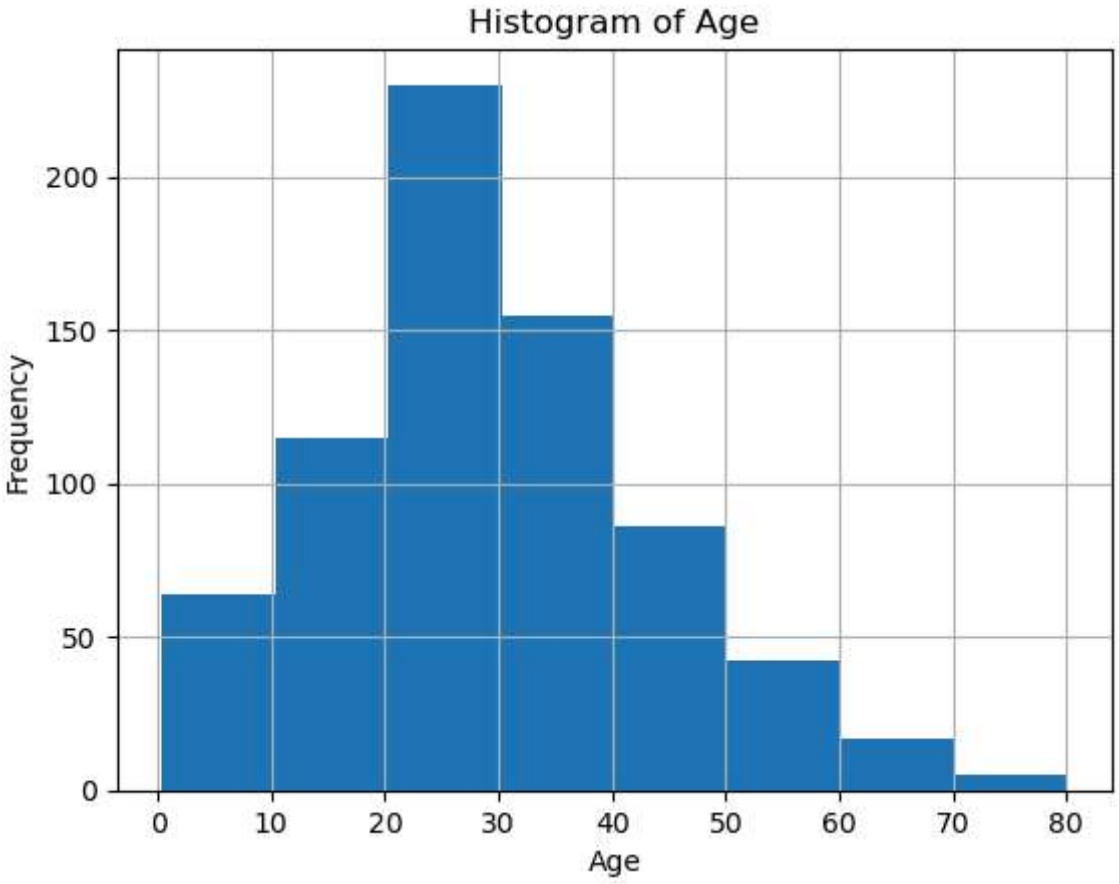
In [67]:
```
data.corr(numeric_only=True)
```

Out[67]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

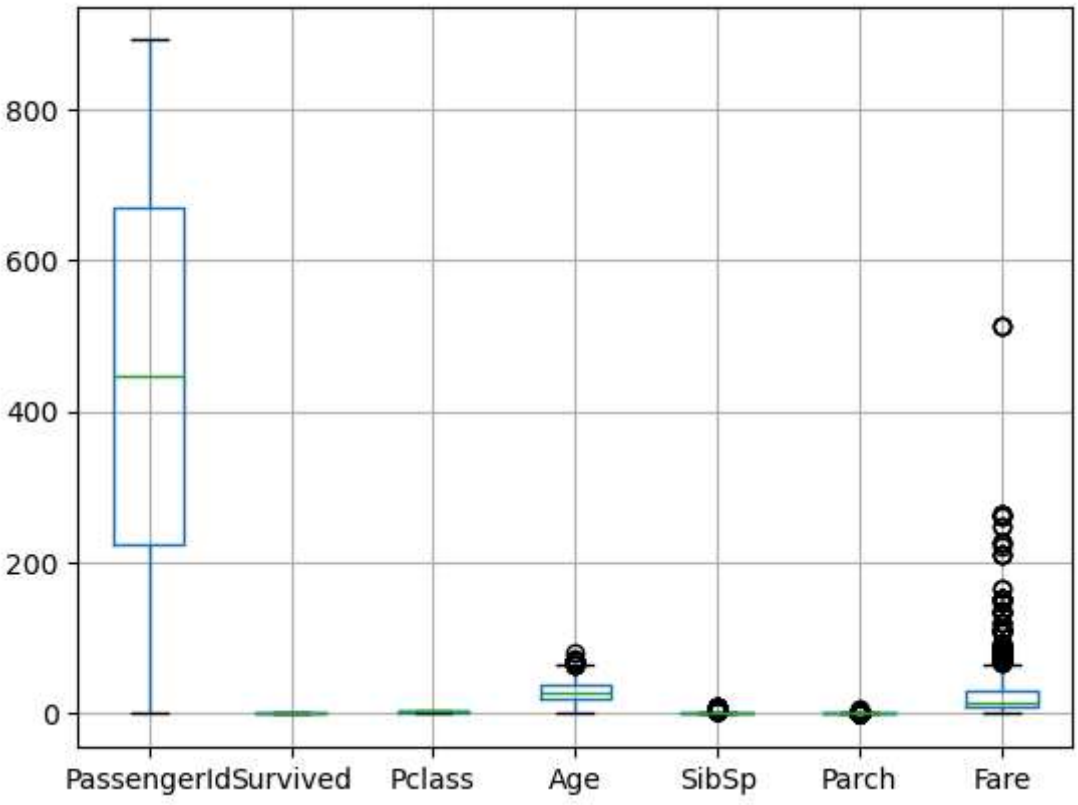
9) Display the histogram for Age attribute by discretizing it into 8 separate bins and counting the frequency for each bin.

In [83]:
```
plt.hist(data['Age'].dropna(),bins=8)
plt.title('Histogram of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



In [84]:

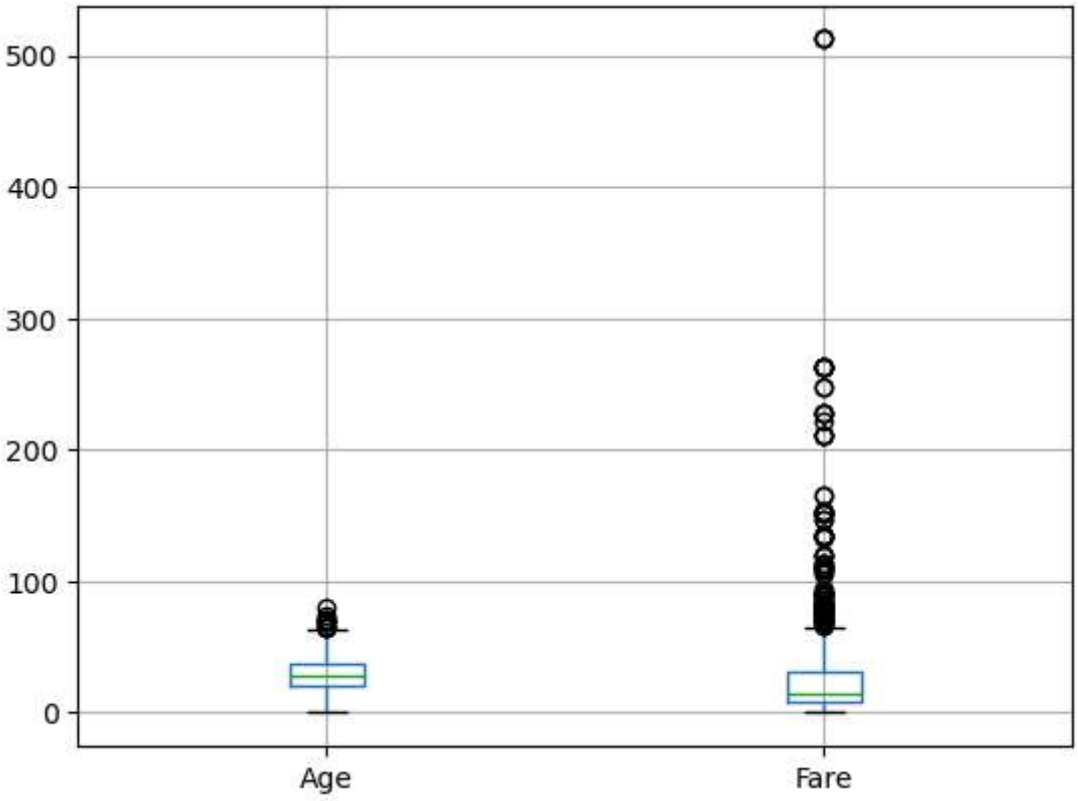
Out[84]: <Axes: >



10) A boxplot can also be used to show the distribution of values for each attribute.

In [86]:
```
data.boxplot(column=['Age','Fare'])
```

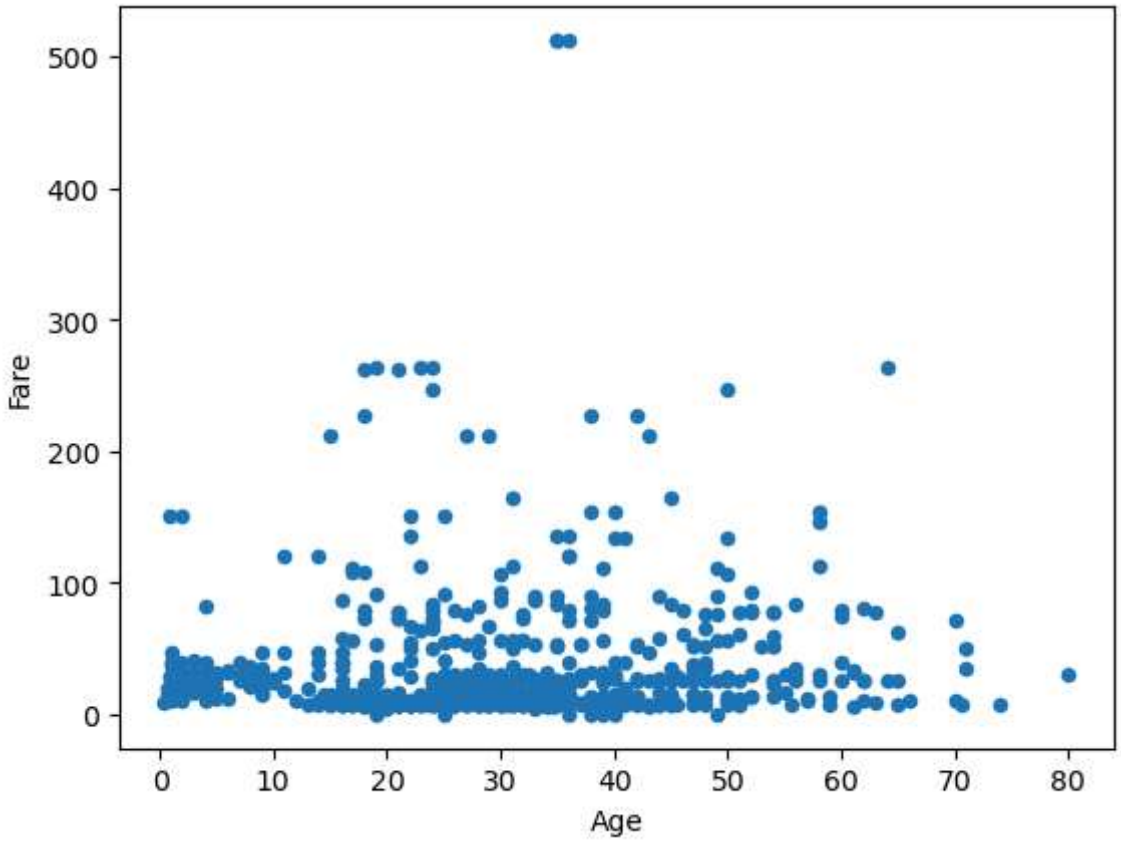
Out[86]: <Axes: >



11) Display scatter plot for any 5 pair of attributes , we can use a scatter plot to visualize their joint distribution.

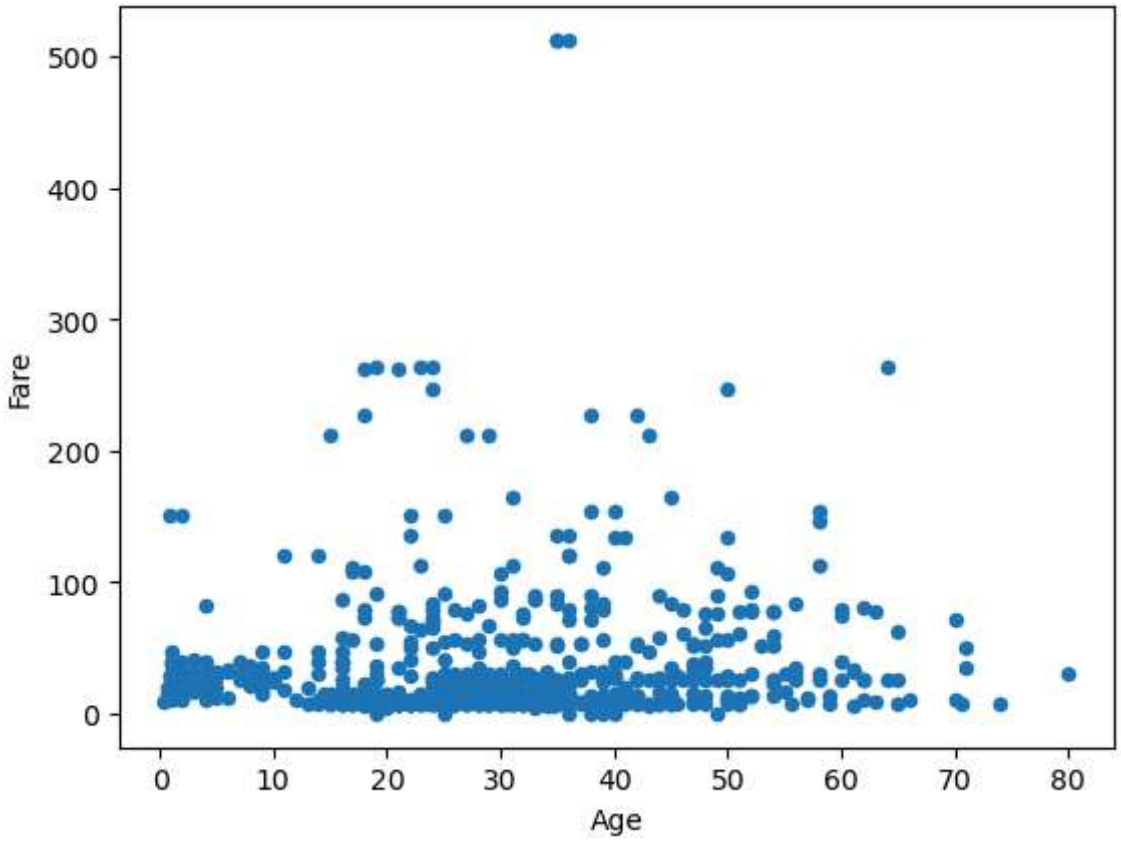
In [89]:
```
data.plot.scatter(x='Age',y='Fare')
```

Out[89]: <Axes: xlabel='Age', ylabel='Fare'>



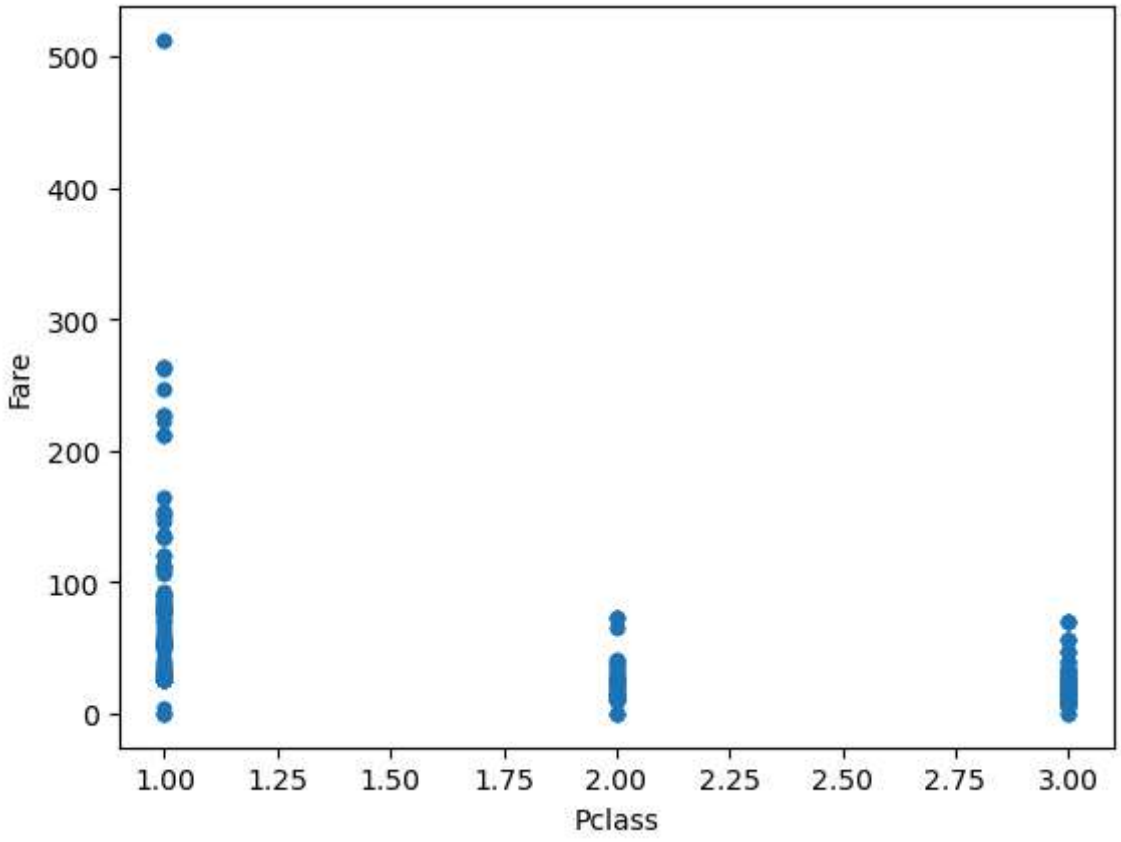
In [90]:
```
data.plot.scatter(x='Age', y='Fare')
```

Out[90]: <Axes: xlabel='Age', ylabel='Fare'>



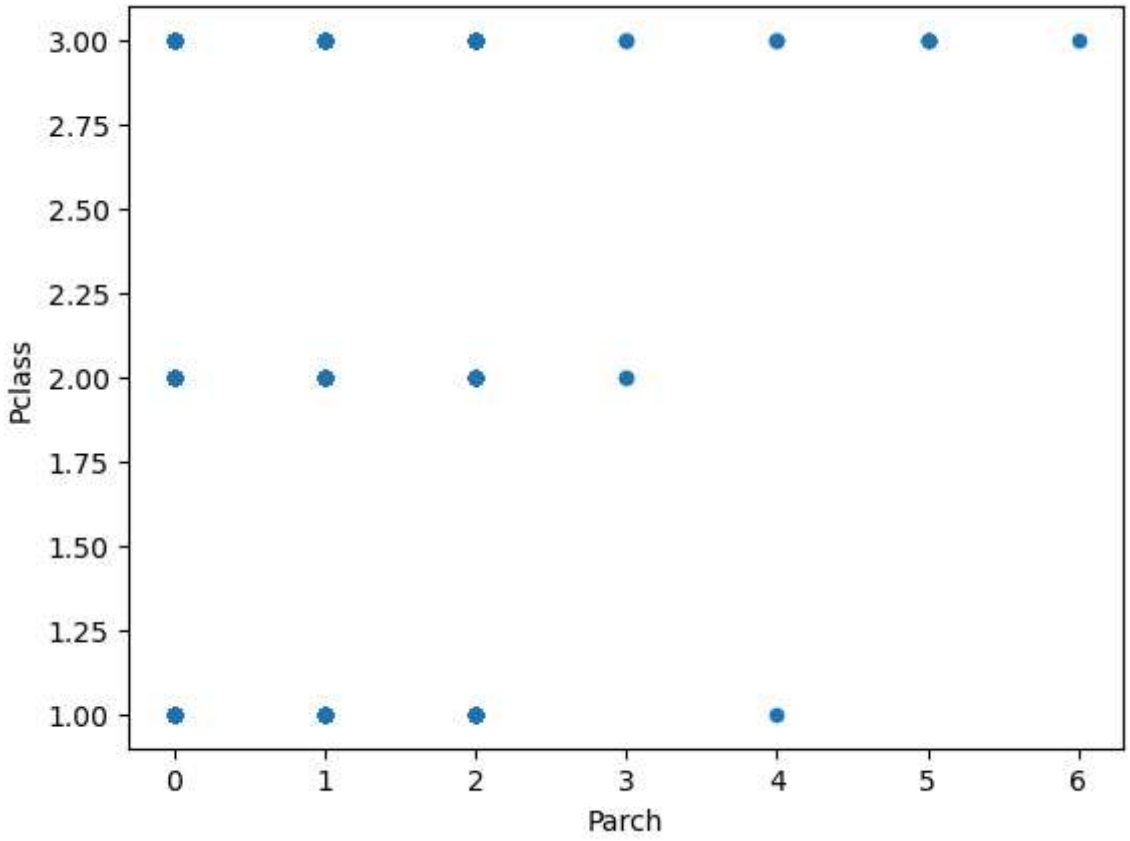
In [91]:
```
data.plot.scatter(x='Pclass', y='Fare')
```

Out[91]:
```
<Axes: xlabel='Pclass', ylabel='Fare'>
```



In [93]:
```
data.plot.scatter(x='Parch', y='Pclass')
```

Out[93]:
```
<Axes: xlabel='Parch', ylabel='Pclass'>
```



In [ ]:
```
data.plot.scatter(x='Age', y='Fare')
```