In [3]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline
```

In [6]:
```
df = pd.read_csv('train.csv')

# take a look at the dataset
df.head()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [8]:
```
# number of rows and columns
df.shape
```

Out[8]:
```
(891, 12)
```

In [9]:
```
# Index, Datatype and Memory information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
```

```
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [10]:
```
# Summary statistics for numerical columns
df.describe()
```

Out[10]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [28]:
```
# Returns the number of non-null values in each DataFrame column
df.count()
```

Out[28]:
```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

In [16]:
```
# Returns the highest value in each column
df.max()
```

Out[16]:
```
PassengerId                             891
Survived                                  1
Pclass                                    3
Name           van Melkebeke, Mr. Philemon
Sex                                    male
Age                                      80
SibSp                                     8
Parch                                     6
Ticket                            WE/P 5735
Fare                                512.329
dtype: object
```

In [20]:
```
# Returns the correlation between columns in a DataFrame
pd.isnull(df).head()
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False

In [32]:
```
# Returns the number of non-null values in each DataFrame column and convert to array u
arr = df.count().to_numpy()
print(arr)
```

```
[891 891 891 891 891 714 891 891 891 891 204 889]
```

In [36]:
```
# sorting using numpy
sorted_arr = np.sort(arr)
print(sorted_arr)
```

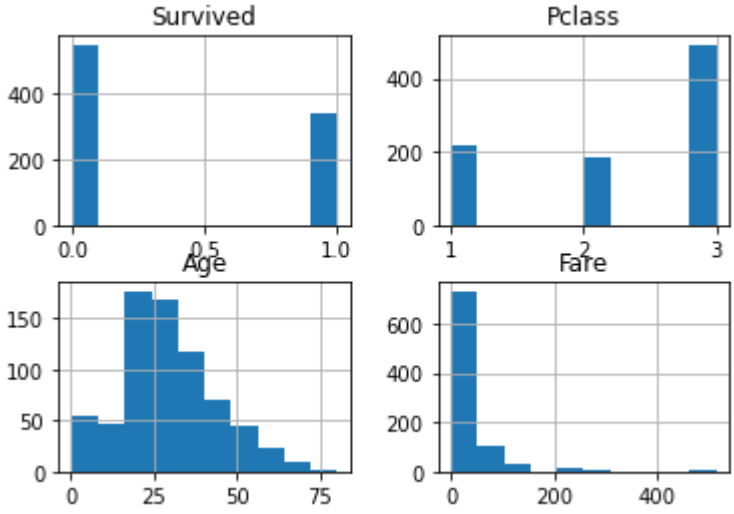
```
[204 714 889 891 891 891 891 891 891 891 891 891]
```

In [38]:
```
cdf = df[['Survived','Pclass','Age','Fare']]
cdf.head(9)
```

Out[38]:

	Survived	Pclass	Age	Fare
0	0	3	22.0	7.2500
1	1	1	38.0	71.2833
2	1	3	26.0	7.9250
3	1	1	35.0	53.1000
4	0	3	35.0	8.0500
5	0	3	NaN	8.4583
6	0	1	54.0	51.8625
7	0	3	2.0	21.0750
8	1	3	27.0	11.1333

In [40]:
```
cdf.hist()
plt.show()
```


Survived
Pclass
Age
Fare
0
200
400
0.0
0.5
1.0
1
2
3
50
100
150
25
50
75
600
200
400

In [ ]: