Darshan UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Lab - 3

Raj Vekariya | 23010101298

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

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

In [4]:
```
df=pd.read_csv("titanic.csv")
df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

891 rows × 12 columns

In [6]:
```
df.head(5)
```

Out[6]:

	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 [8]:
```
nominal=['Name','Sex','Cabin','Embarked','Ticket']
ordinal=['Pclass']
binary=['Sex','Survived']
numeric=['Age','Fare','Sibsp','Parch']

print("Nominal:",nominal)
print("Ordinal:",ordinal)
print("Binary:",binary)
print("Numeric:",numeric)
```

```
Nominal: ['Name', 'Sex', 'Cabin', 'Embarked', 'Ticket']
Ordinal: ['Pclass']
Binary: ['Sex', 'Survived']
Numeric: ['Age', 'Fare', 'Sibsp', 'Parch']
```

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

In [11]:
```
print("Survived values(asymmetric binary)")
print(df['Survived'].value_counts())
print("---------------------------------")
```

```
print("Gender count(symmetric binary)")
print(df['Sex'].value_counts())
```

```
Survived values(asymmetric binary)
0    549
1    342
Name: Survived, dtype: int64
--------------------------------
Gender count(symmetric binary)
male      577
female    314
Name: Sex, dtype: int64
```

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

In [14]:
```
quantitative=['PassengerId','Survived','Pclass','Age','SibSp','Parch','Fare']

for i in quantitative:
    print(":::",i,":::")
    print("Mean",df[i].mean())
    print("Standard Deviation",df[i].std())
    print("Minimum",df[i].min())
    print("Maximum",df[i].max())
    print("Mode",df[i].mode()[0])
    print("Range",df[i].max()-df[i].min())
    print("--------------------------------")
```

```
::: PassengerId :::
Mean 446.0
Standard Deviation 257.3538420152301
Minimum 1
Maximum 891
Mode 1
Range 890
------------------------------
::: Survived :::
Mean 0.3838383838383838
Standard Deviation 0.4865924542648575
Minimum 0
Maximum 1
Mode 0
Range 1
------------------------------
::: Pclass :::
Mean 2.308641975308642
Standard Deviation 0.836071240977049
Minimum 1
Maximum 3
Mode 3
Range 2
------------------------------
::: Age :::
Mean 29.69911764705882
Standard Deviation 14.526497332334042
Minimum 0.42
Maximum 80.0
Mode 24.0
Range 79.58
------------------------------
::: SibSp :::
Mean 0.5230078563411896
Standard Deviation 1.1027434322934317
Minimum 0
Maximum 8
Mode 0
Range 8
------------------------------
::: Parch :::
Mean 0.38159371492704824
Standard Deviation 0.8060572211299483
Minimum 0
Maximum 6
Mode 0
Range 6
------------------------------
::: Fare :::
Mean 32.204207968574636
Standard Deviation 49.6934285971809
Minimum 0.0
Maximum 512.3292
Mode 8.05
Range 512.3292
------------------------------
```

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

In [19]:
```
print("Frequency of Pclass\n",df['Pclass'].value_counts())
```

```
Frequency of Pclass
 3    491
1    216
2    184
Name: Pclass, 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 [23]:
```
print("Summary for numeric attributes:")
print(df.describe())

print("\nSummary for categorical attributes")
print(df.describe(include="object"))
```

```
Summary for numeric attributes:
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200

Summary for categorical attributes
                           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 [33]:
```
print("Covariance matrix:")
print(df.cov())
```

```
Covariance matrix:
             PassengerId  Survived     Pclass         Age      SibSp  \
PassengerId  66231.000000 -0.626966  -7.561798  138.696504 -16.325843
Survived        -0.626966  0.236772  -0.137703   -0.551296  -0.018954
Pclass          -7.561798 -0.137703   0.699015   -4.496004   0.076599
Age            138.696504 -0.551296  -4.496004  211.019125  -4.163334
SibSp          -16.325843 -0.018954   0.076599   -4.163334   1.216043
Parch           -0.342697  0.032017   0.012429   -2.344191   0.368739
Fare           161.883369  6.221787 -22.830196   73.849030   8.748734

                Parch         Fare
PassengerId -0.342697   161.883369
Survived     0.032017     6.221787
Pclass       0.012429   -22.830196
Age         -2.344191    73.849030
SibSp        0.368739     8.748734
Parch        0.649728     8.661052
Fare         8.661052  2469.436846
```

In [34]:
```
print("Correlation matrix:")
print(df.corr())
```

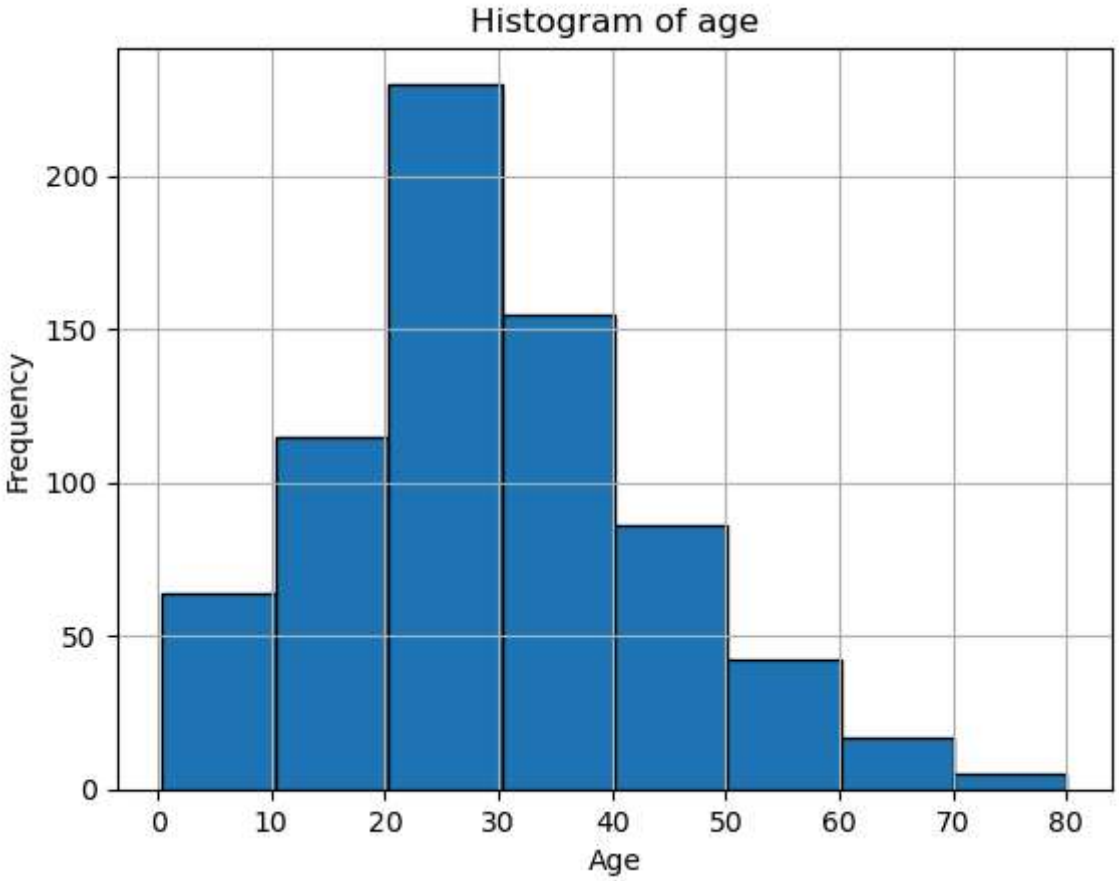
```
Correlation matrix:
             PassengerId  Survived    Pclass       Age     SibSp     Parch  \
PassengerId     1.000000 -0.005007 -0.035144  0.036847 -0.057527 -0.001652
Survived       -0.005007  1.000000 -0.338481 -0.077221 -0.035322  0.081629
Pclass         -0.035144 -0.338481  1.000000 -0.369226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

                 Fare
PassengerId  0.012658
Survived     0.257307
Pclass      -0.549500
Age          0.096067
SibSp        0.159651
Parch        0.216225
Fare         1.000000
```

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

In [6]:
```
import matplotlib.pyplot as plt
```

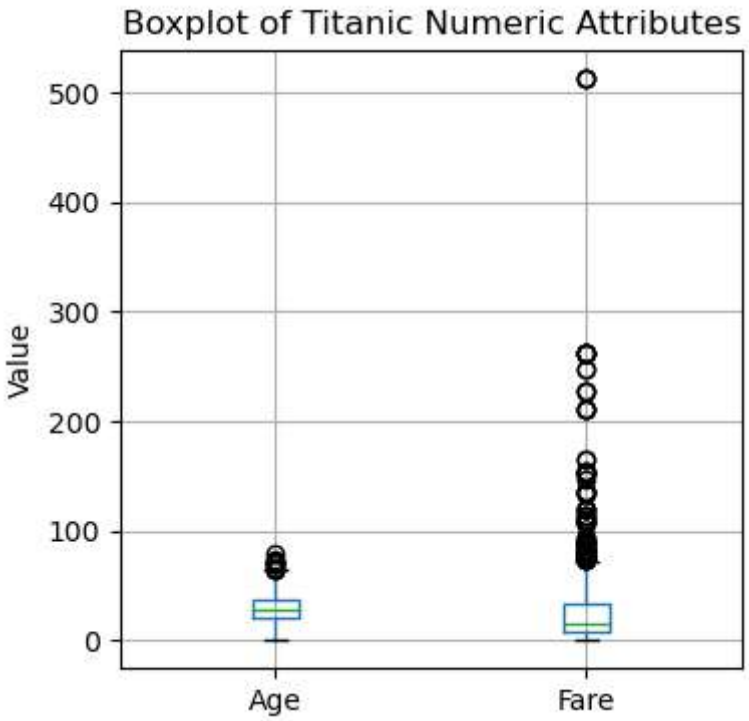
In [37]:
```
df['Age'].dropna().hist(bins=8,edgecolor='black')
plt.title("Histogram of age")
```

```
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



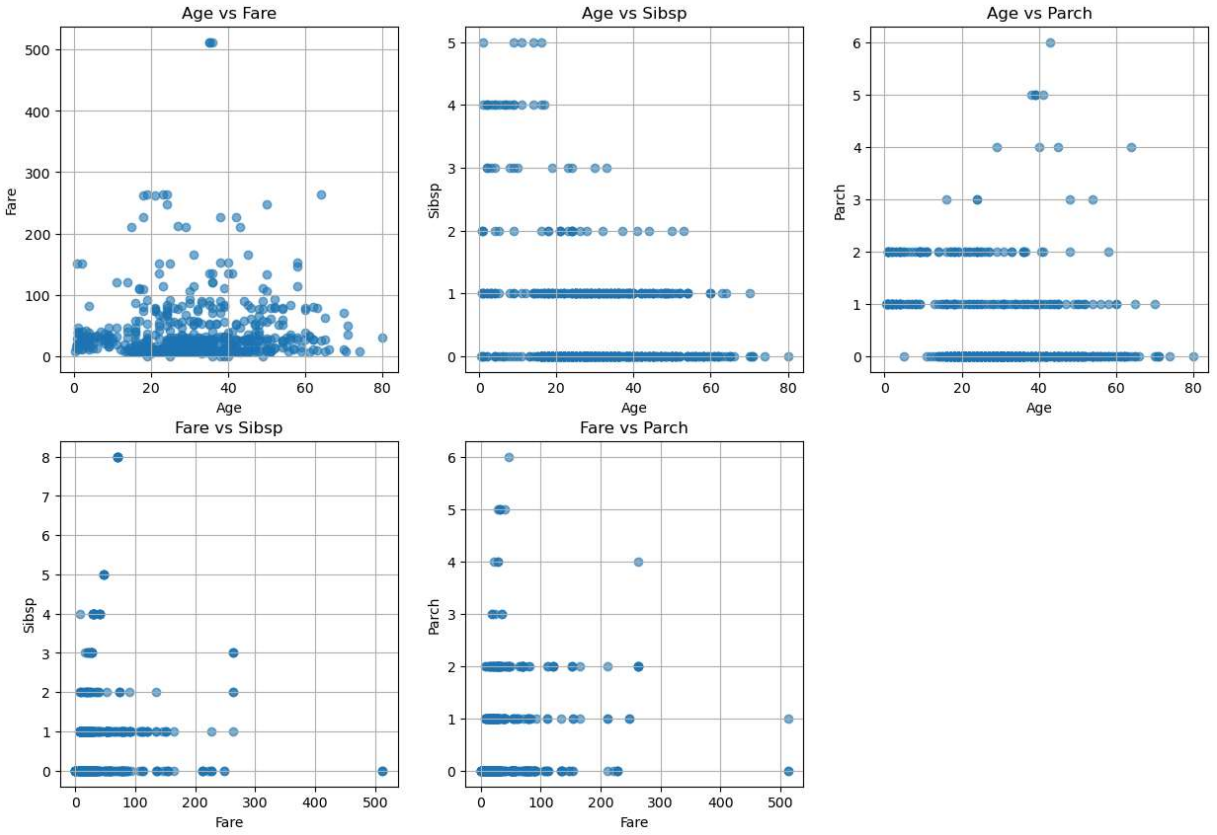
10) A boxplot can also be used to show the distribution of values for each attribute.

In [16]:
```
numeric = ['Age', 'Fare']
data = df[numeric].dropna()

plt.figure(figsize = (4,4))
data.boxplot(column = numeric)
plt.title("Boxplot of Titanic Numeric Attributes")
plt.ylabel("Value")
plt.grid(True)
plt.show()
```



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

In [15]:
```
pairs = [('Age', 'Fare'), ('Age', 'SibSp'), ('Age', 'Parch'), ('Fare', 'SibSp'), ('

plt.figure(figsize = (15, 10))

for i, (x, y) in enumerate(pairs):
    plt.subplot(2, 3, i + 1)
    plt.scatter(df[x], df[y], alpha=0.6)
    plt.xlabel(x.capitalize())
    plt.ylabel(y.capitalize())
    plt.title(f"{x.capitalize()} vs {y.capitalize()}")
    plt.grid(True)
```


Age vs Fare
Fare
Age
Age vs Sibsp
Sibsp
Age
Age vs Parch
Parch
Age
Fare vs Sibsp
Sibsp
Fare
Fare vs Parch
Parch
Fare

In [ ]: