Darshan UNIVERSITY
योग: कर्मसु कौशलम्

Data Mining

Lab - 3

137 | Vishal Baraiya | 23010101014

https://tinyurl.com/dm12025

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

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [5]:
```
df= pd.read_csv("titanic.csv")
```

In [3]:
```
df.head(5)
```

Out[3]:

	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 [4]:
```
nominal  = ["Name","Sex","Cabin","Embarked","Ticket"]
ordinal = ["Pclass"]
binary = ["Sex","Survived"]
numaric = ["Age","Fare","SibSp","Parch"]

print("Nominal : ",nominal)
print("Ordinal : ",ordinal)
print("Binary : ",binary)
print("Numaric : ",numaric)
```

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

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

In [6]:
```
print("Survived values (Asymmetric binary):")
print(df['Survived'].value_counts())
print("-"*50)
```

```
print("Gender Count (Symmetric binary):")
print(df['Sex'].value_counts())
```

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

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

In [10]:
```
quantitative = ['PassengerId','Survived','Pclass','Age','SibSp','Parch','Fare']

for col in quantitative:
    print("::::::",col,"::::::")
    print("Mean : ",df[col].mean())
    print("Standard Devition : ",df[col].std())
    print("Minimum : ",df[col].min())
    print("Maximum : ",df[col].max())
    print("Mode : ",df[col].mode()[0])
    print("Range : ",df[col].max()-df[col].min())
    print("-"*50)
```

```
:::::: PassengerId ::::::
Mean :  446.0
Standard Devition :  257.3538420152301
Minimum :  1
Maximum :  891
Mode :  1
Range :  890
--------------------------------------------------
:::::: Survived ::::::
Mean :  0.3838383838383838
Standard Devition :  0.4865924542648585
Minimum :  0
Maximum :  1
Mode :  0
Range :  1
--------------------------------------------------
:::::: Pclass ::::::
Mean :  2.308641975308642
Standard Devition :  0.8360712409770513
Minimum :  1
Maximum :  3
Mode :  3
Range :  2
--------------------------------------------------
:::::: Age ::::::
Mean :  29.69911764705882
Standard Devition :  14.526497332334044
Minimum :  0.42
Maximum :  80.0
Mode :  24.0
Range :  79.58
--------------------------------------------------
:::::: SibSp ::::::
Mean :  0.5230078563411896
Standard Devition :  1.1027434322934275
Minimum :  0
Maximum :  8
Mode :  0
Range :  8
--------------------------------------------------
:::::: Parch ::::::
Mean :  0.38159371492704824
Standard Devition :  0.8060572211299559
Minimum :  0
Maximum :  6
Mode :  0
Range :  6
--------------------------------------------------
:::::: Fare ::::::
Mean :  32.204207968574636
Standard Devition :  49.693428597180905
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 [11]:
```
print("Passenger Class Frequency : ")
print(df["Pclass"].value_counts())
```

```
Passenger Class Frequency :
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 [22]:
```
print("\nSummary for Numeric Attributes : ")
df.describe()
```

```
Summary for Numeric Attributes :
```

Out[22]:

	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 [23]:
```
print("\nSummary for All attributes : ")
df.describe(include='all')
```

```
Summary for All attributes :
```

Out[23]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	F
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.00
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.38
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.80
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.00
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.00
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.00
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.00
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.00

In [24]:
```
print("\nSummary for Categorical Attributes : ")
df.describe(include="object")
#df.describe(include=["object"]) # Also valid
```

```
Summary for Categorical Attributes :
```

Out[24]:

	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 [26]:
```
print("Covariance Matrix : ")
df.cov(numeric_only=True)
```

```
Covariance Matrix :
```

Out[26]:

	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 [27]:
```
print("Correlation Matrix : ")
df.corr(numeric_only=True)
```

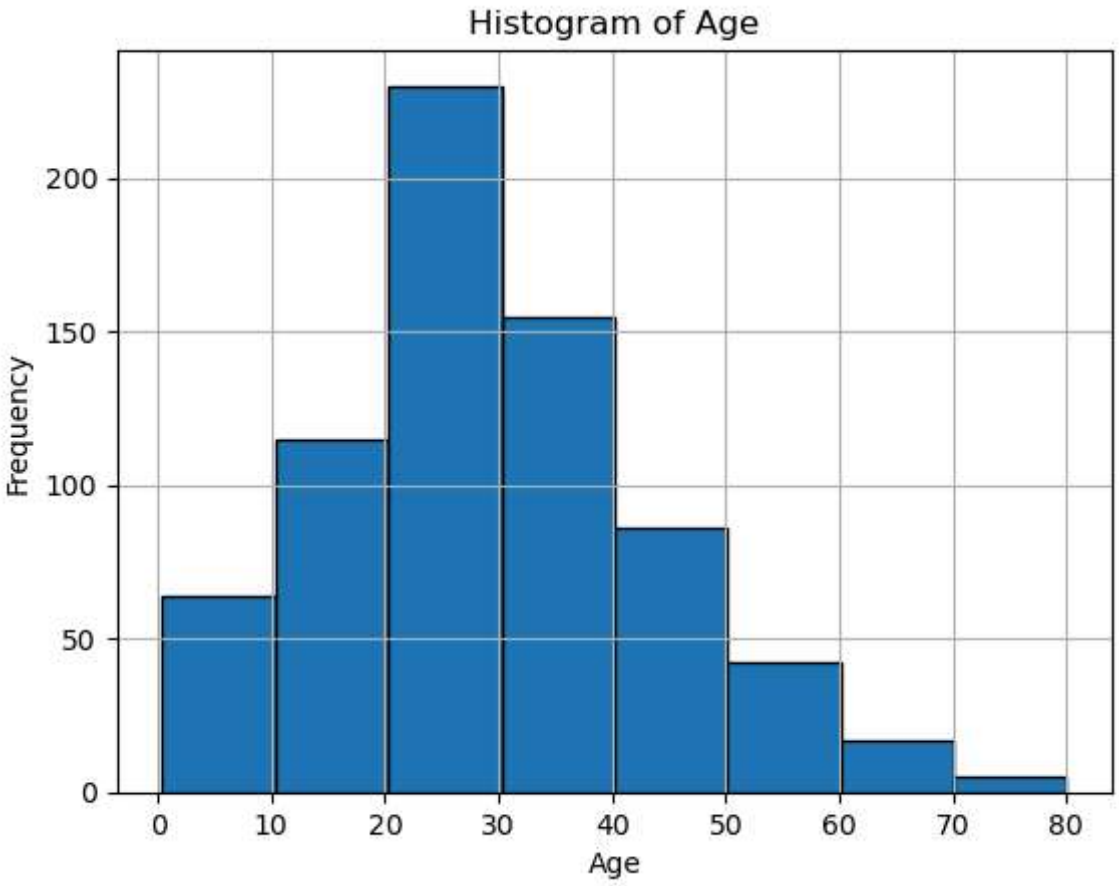
```
Correlation Matrix :
```

Out[27]:

	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.

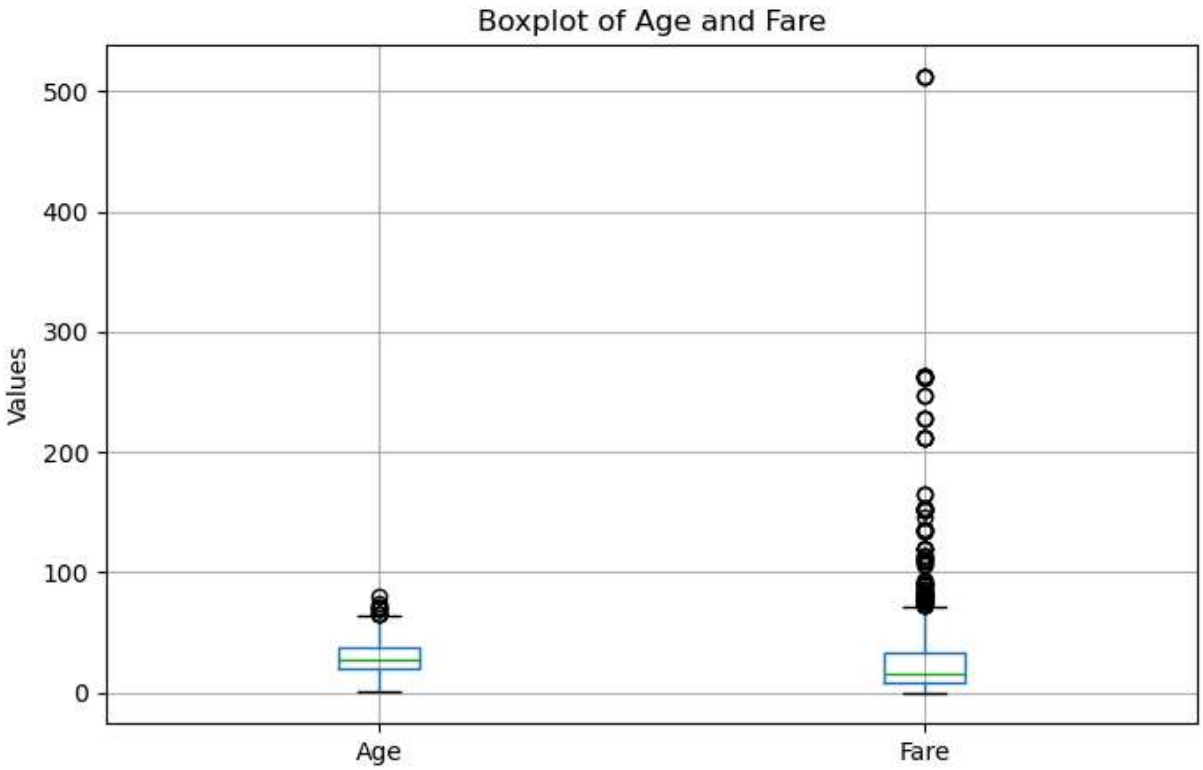
In [6]:
```
df["Age"].dropna().hist(bins=8, edgecolor='black')
plt.title("Histogram of Age")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()
```



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

In [8]:
```
df_filtered = df[['Age', 'Fare']].dropna()

# Plot boxplots for Age and Fare
plt.figure(figsize=(8, 5))
df_filtered.boxplot(column=['Age', 'Fare'])
plt.title('Boxplot of Age and Fare')
plt.ylabel('Values')
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.

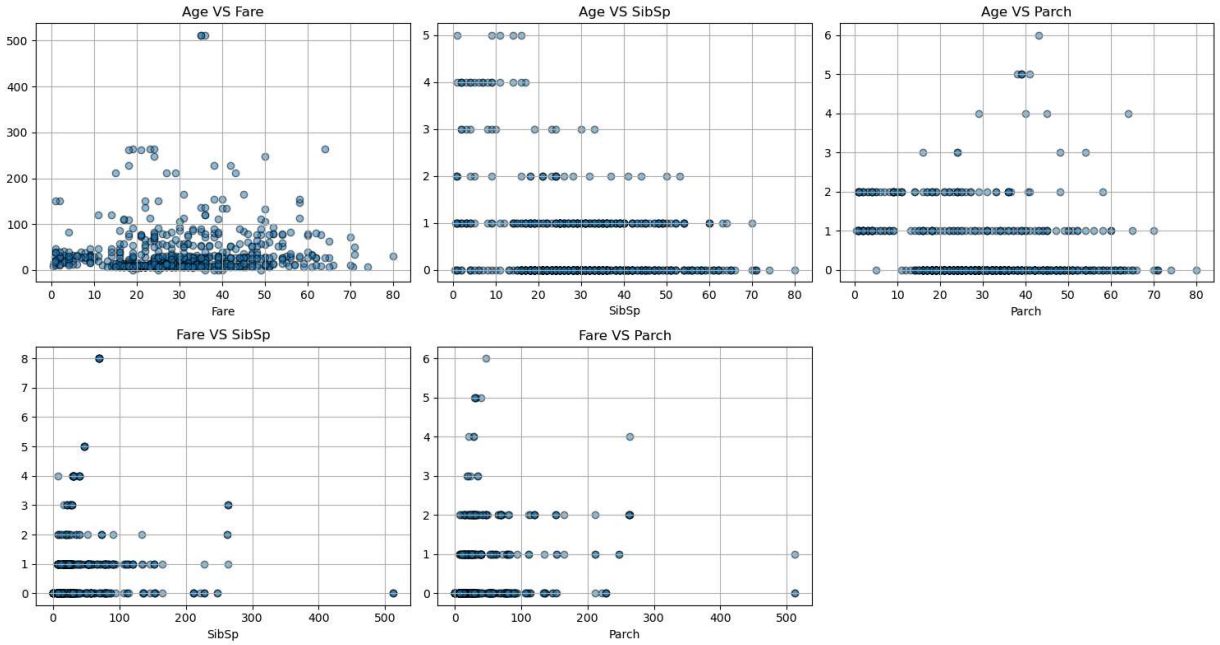
In [43]:
```
pairs = [
    ('Age','Fare'),
    ('Age','SibSp'),
    ('Age','Parch'),
    ('Fare','SibSp'),
    ('Fare','Parch')
]

plt.figure(figsize=(15,8))

for i, (x, y) in enumerate(pairs):
    plt.subplot(2, 3, i+1)
    plt.scatter(df[x],df[y],alpha=0.5,edgecolors='k')
    plt.xlabel(x)
    plt.xlabel(y)
    plt.title(f"{x} VS {y}")
    plt.grid(True)

plt.tight_layout()
plt.suptitle("Scatter Of Attributes Pairs",fontsize=16, y=1.10)
plt.show()
```

Scatter Of Attributes Pairs



In [ ]: