Week 5

September 25, 2024

[1]:
```
import pandas as pd
df = pd.DataFrame({'Date':['10/2/2011', '11/2/2011', '12/2/2011', '13/2/2011'],
                   'Event':['Music', 'Poetry', 'Theatre', 'Comedy'],
                   'Cost':[10000, 5000, 15000, 2000]})
print(df)
```

```
        Date    Event   Cost
0  10/2/2011    Music  10000
1  11/2/2011   Poetry   5000
2  12/2/2011  Theatre  15000
3  13/2/2011   Comedy   2000
```

[2]:
```
df['Discounted_Price'] = df.apply(lambda row: row.Cost * 0.9, axis = 1)
print(df)
```

```
        Date    Event   Cost  Discounted_Price
0  10/2/2011    Music  10000            9000.0
1  11/2/2011   Poetry   5000            4500.0
2  12/2/2011  Theatre  15000           13500.0
3  13/2/2011   Comedy   2000            1800.0
```

[3]:
```
df = pd.DataFrame({'Name':['John','Ted','Dove','Brad','Rex'],
                   'Salary':[44000, 35000, 75000, 20000,6000]})
print(df)
```

```
   Name  Salary
0  John   44000
1   Ted   35000
2  Dove   75000
3  Brad   20000
4   Rex    6000
```

[4]:
```
def salary_stats(value):
    if value < 10000:
        return "very low"
    elif 10000 <= value < 25000:
        return "low"
    elif 25000 <= value < 40000:
        return "average"
    elif 40000 <= value < 50000:
        return "better"
    elif value >= 50000:
        return "very good"

df['salary_stats'] = df['Salary'].map(salary_stats)
df
```

[4]:
```
   Name  Salary salary_stats
0  John   44000       better
1   Ted   35000      average
2  Dove   75000    very good
3  Brad   20000          low
4   Rex    6000     very low
```

[9]:
```
import pandas as pd
data = pd.DataFrame({
 'Name' : ['A', 'B', 'C', 'D','E', 'F'],
'Education' : ['High School', 'Masters', 'Doctorate', 'Bachelors','Masters',␣
 ↪'High School']})
data
```

[9]:
```
  Name    Education
0    A  High School
1    B      Masters
2    C    Doctorate
3    D    Bachelors
4    E      Masters
5    F  High School
```

1 Binary Encoding

[10]:
```
education_data = pd.get_dummies(data.Education)
print(education_data)
```

```
   Bachelors  Doctorate  High School  Masters
0      False      False         True    False
1      False      False        False     True
2      False       True        False    False
3       True      False        False    False
4      False      False        False     True
5      False      False         True    False
```

2 Ranking Transformation

[11]:
```
education_map = {
 'High School' : 1,
 'Bachelors' : 2,
 'Masters': 3,
 'Doctorate': 4
 }
education_data = data['Education'].map(education_map)
data['Education'] = education_data
data
```

[11]:
```
  Name  Education
0    A          1
1    B          3
2    C          4
3    D          2
4    E          3
5    F          1
```

[12]:
```
education_map = {
    'High School' : 12,
    'Bachelors' : 16,
    'Masters': 18,
    'Doctorate': 21
 }
education_data = data['Education'].map(education_map)
data['Education'] = education_data
data
```

[12]:
```
  Name  Education
0    A        NaN
1    B        NaN
2    C        NaN
3    D        NaN
4    E        NaN
5    F        NaN
```

3 Adding data objects- rows

[13]:
```
df.loc[len(df.index)]=['Hruthvik', 15000, 'low']
df
```

[13]:
```
        Name  Salary salary_stats
0       John   44000       better
1        Ted   35000      average
2       Dove   75000    very good
```

```
3      Brad   20000          low
4       Rex    6000     very low
5  Hruthvik   15000          low
```

4 Combining two data frames

[15]:
```
import pandas as pd
d1 = {'Name': ['Pankaj', 'Meghna', 'Lisa'], 'Country': ['India', 'India',␣
 ↪'USA']}
df1 = pd.DataFrame(d1)
print('DataFrame 1:\n', df1,'\n')
df2 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Pankaj', 'Anupam', 'Amit']})
print('DataFrame 2:\n', df2,'\n')
df3 = pd.DataFrame({'Name': ['Priya'], 'Country': ['India'], 'Role': ['COO']})
print('DataFrame 3:\n', df3,'\n')
```

```
DataFrame 1:
      Name Country
0  Pankaj   India
1  Meghna   India
2    Lisa     USA

DataFrame 2:
    ID    Name
0   1  Pankaj
1   2  Anupam
2   3    Amit

DataFrame 3:
     Name Country Role
0  Priya   India  COO
```

[16]:
```
same_cols_df = pd.concat([df1,df3],ignore_index=True)
same_cols_df
```

[16]:
```
     Name Country Role
0  Pankaj   India  NaN
1  Meghna   India  NaN
2    Lisa     USA  NaN
3   Priya   India  COO
```

[17]:
```
a_df=df1.append(df2, ignore_index=True)
a_df
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_26088\2048347772.py in ?()
----> 1 a_df=df1.append(df2, ignore_index=True)
      2 a_df

~\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   6295                 and name not in self._accessors
   6296                 and self._info_axis.
 ↪_can_hold_identifiers_and_holds_name(name)
   6297             ):
   6298                 return self[name]
-> 6299         return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'append'
```

[18]:
```
c_df = pd.concat([df1,df2],ignore_index=True)
c_df
```

[18]:
```
     Name Country   ID
0  Pankaj   India  NaN
1  Meghna   India  NaN
2    Lisa     USA  NaN
3  Pankaj     NaN  1.0
4  Anupam     NaN  2.0
5    Amit     NaN  3.0
```

5 Defaut Mergeing - inner join

[19]:
```
df_merged = df1.merge(df2)
print('Result:\n', df_merged)
```

```
Result:
      Name Country  ID
0  Pankaj   India   1
```

6 Mergeing DataFrames with left, Right and outer join

[20]:
```
print('Result Left Join:\n', df1.merge(df2, how='left'))
print('Result Right Join:\n', df1.merge(df2, how='right'))
print('Result Outer Join:\n', df1.merge(df2, how='outer'))
```

```
Result Left Join:
      Name Country   ID
0  Pankaj   India  1.0
1  Meghna   India  NaN
2    Lisa     USA  NaN
Result Right Join:
      Name Country  ID
0  Pankaj   India   1
1  Anupam     NaN   2
2    Amit     NaN   3
Result Outer Join:
      Name Country   ID
0    Amit     NaN  3.0
1  Anupam     NaN  2.0
2    Lisa     USA  NaN
3  Meghna   India  NaN
4  Pankaj   India  1.0
```

7 Mergeing dataframes with specific columns

[30]:
```
import pandas as pd

dict1 = {
    'ID': [1, 2, 3],
    'Name': ['Pankaj', 'Meghna', 'Lisa'],
    'Country': ['India', 'India', 'India'],
    'Role': ['CEO', 'CTO', 'CTO']
}

df1 = pd.DataFrame(dict1)
df2 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Pankaj', 'Anupam', 'Amit']})

print(df1.merge(df2, on='ID'))
print('\n', df1.merge(df2, on='Name'))
```

```
   ID  Name_x Country Role  Name_y
0   1  Pankaj   India  CEO  Pankaj
1   2  Meghna   India  CTO  Anupam
2   3    Lisa   India  CTO    Amit

    ID_x    Name Country Role  ID_y
0     1  Pankaj   India  CEO     1
```

8 Titanic CSV

[41]:
```
import pandas as pd
import missingno as msno
%matplotlib inline
```

[34]:
```
titanic_df = pd.read_csv("titanic.csv")
titanic_df
```

```
[34]:      PassengerId  Survived  Pclass  \
     0              1         0       3
     1              2         1       1
     2              3         1       3
     3              4         1       1
     4              5         0       3
     ..           …       …     …
     886          887         0       2
     887          888         1       1
     888          889         0       3
     889          890         1       1
     890          891         0       3

                                                       Name     Sex   Age  SibSp  \
     0                              Braund, Mr. Owen Harris    male  22.0      1
     1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
     2                               Heikkinen, Miss. Laina  female  26.0      0
     3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
     4                             Allen, Mr. William Henry    male  35.0      0
     ..                                                 …     …   …    …
     886                              Montvila, Rev. Juozas    male  27.0      0
     887                       Graham, Miss. Margaret Edith  female  19.0      0
     888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
     889                              Behr, Mr. Karl Howell    male  26.0      0
     890                                Dooley, Mr. Patrick    male  32.0      0

          Parch            Ticket     Fare Cabin Embarked
     0        0         A/5 21171   7.2500   NaN        S
     1        0          PC 17599  71.2833   C85        C
     2        0  STON/O2. 3101282   7.9250   NaN        S
     3        0            113803  53.1000  C123        S
     4        0            373450   8.0500   NaN        S
     ..     …             …     …   …      …
     886      0            211536  13.0000   NaN        S
     887      0            112053  30.0000   B42        S
     888      2        W./C. 6607  23.4500   NaN        S
     889      0            111369  30.0000  C148        C
     890      0            370376   7.7500   NaN        Q

     [891 rows x 12 columns]
```

```
[35]: titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
```

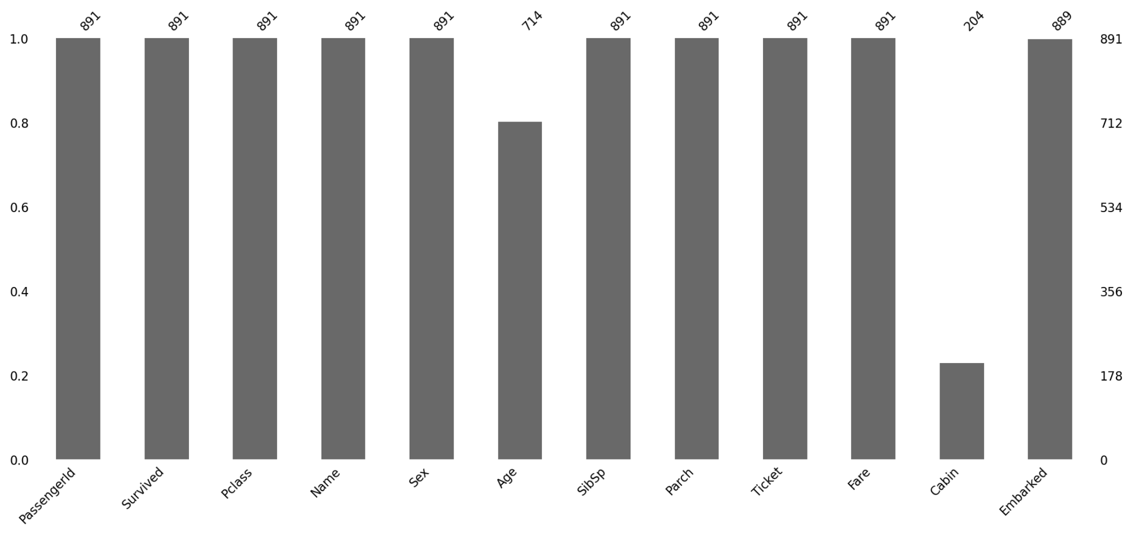
```
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        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
```

[36]:
```
titanic_df.isnull()
```

[36]:
```
     PassengerId  Survived  Pclass   Name    Sex    Age  SibSp  Parch  Ticket  \
0          False     False   False  False  False  False  False  False   False
1          False     False   False  False  False  False  False  False   False
2          False     False   False  False  False  False  False  False   False
3          False     False   False  False  False  False  False  False   False
4          False     False   False  False  False  False  False  False   False
..           …         …       …      …      …      …      …      …       …
886        False     False   False  False  False  False  False  False   False
887        False     False   False  False  False  False  False  False   False
888        False     False   False  False  False   True  False  False   False
889        False     False   False  False  False  False  False  False   False
890        False     False   False  False  False  False  False  False   False

      Fare  Cabin  Embarked
0    False   True     False
1    False  False     False
2    False   True     False
3    False  False     False
4    False   True     False
..     …      …         …
886  False   True     False
887  False  False     False
888  False   True     False
889  False  False     False
890  False   True     False

[891 rows x 12 columns]
```

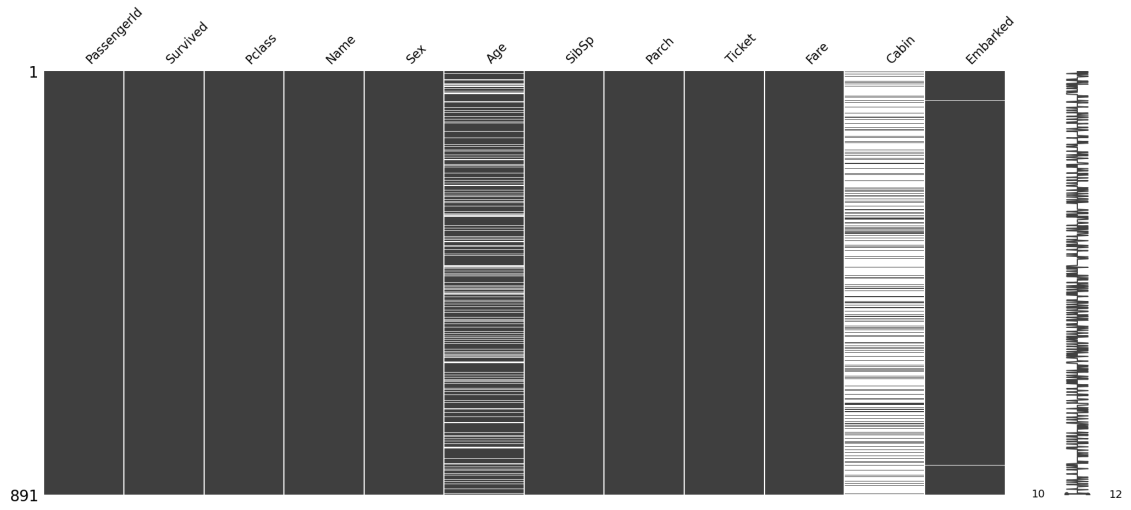
[42]:
```
msno.bar(titanic_df)
```

[42]:
```
<Axes: >
```



[43]:
```
msno.matrix(titanic_df)
```

[43]:
```
<Axes: >
```



[44]:
```
titanic_df.isnull().sum()
```

[44]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
```

```
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

[45]:
```
df = titanic_df.dropna(axis=0)
df.isnull().sum()
```

[45]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

[46]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 183 entries, 1 to 889
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  183 non-null    int64
 1   Survived     183 non-null    int64
 2   Pclass       183 non-null    int64
 3   Name         183 non-null    object
 4   Sex          183 non-null    object
 5   Age          183 non-null    float64
 6   SibSp        183 non-null    int64
 7   Parch        183 non-null    int64
 8   Ticket       183 non-null    object
 9   Fare         183 non-null    float64
 10  Cabin        183 non-null    object
 11  Embarked     183 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 18.6+ KB
```

[47]:
```
titanic_df.columns
```

```
[47]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
              'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
             dtype='object')
```

```
[50]: df = titanic_df.drop(['Pclass'],axis=1)
      df.isnull().sum()
```

```
[50]: PassengerId      0
      Survived         0
      Name             0
      Sex              0
      Age            177
      SibSp            0
      Parch            0
      Ticket           0
      Fare             0
      Cabin          687
      Embarked         2
      dtype: int64
```

```
[51]: titanic_df['Pclass'].unique()
```

```
[51]: array([3, 1, 2], dtype=int64)
```

```
[53]: titanic_df['Pclass'] = titanic_df['Pclass'].fillna('C')
```

```
[54]: titanic_df['Pclass'].isnull().sum()
```

```
[54]: 0
```

```
[55]: titanic_df
```

```
[55]:      PassengerId  Survived  Pclass  \
      0              1         0       3
      1              2         1       1
      2              3         1       3
      3              4         1       1
      4              5         0       3
      ..           …         …       …
      886          887         0       2
      887          888         1       1
      888          889         0       3
      889          890         1       1
      890          891         0       3

                                                       Name     Sex   Age  SibSp  \
      0                                 Braund, Mr. Owen Harris    male  22.0      1
      1     Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                              Heikkinen, Miss. Laina  female  26.0      0
3        Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                            Allen, Mr. William Henry    male  35.0      0
..                                                ...     ...   ...    ...
886                             Montvila, Rev. Juozas    male  27.0      0
887                      Graham, Miss. Margaret Edith  female  19.0      0
888          Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                             Behr, Mr. Karl Howell    male  26.0      0
890                               Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[57]:
```
mean = titanic_df['Age'].mean()
print(mean)
#Replace the missing values for numerical columns with mean
titanic_df['Age'] = titanic_df['Age'].fillna(mean)
titanic_df['Age']
```

```
29.69911764705882
```

[57]:
```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

[58]:
```
titanic_df = pd.read_csv("titanic.csv")
 #Replace the missing values for categorical columns with mode
mode = titanic_df['Pclass'].mode()[0]
print(mode)
titanic_df['Pclass'] = titanic_df['Pclass'].fillna(mode)
```

```
3
```

[59]:
```
titanic_df['Pclass']
```

[59]:
```
0      3
1      1
2      3
3      1
4      3
      ..
886    2
887    1
888    3
889    1
890    3
Name: Pclass, Length: 891, dtype: int64
```

[61]:
```
titanic_df['Age']= titanic_df['Age'].fillna(titanic_df['Age'].median())
titanic_df['Age']
```

[61]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       …
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

[62]:
```
titanic_df = pd.read_csv("titanic.csv")
titanic_df
```

[62]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
```

```
..           …         …      …
886          887        0       2
887          888        1       1
888          889        0       3
889          890        1       1
890          891        0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

[63]:
```python
new_df = titanic_df.fillna(method="ffill")
new_df
```

```
C:\Users\skand\AppData\Local\Temp\ipykernel_26088\3071186871.py:1:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a
future version. Use obj.ffill() or obj.bfill() instead.
  new_df = titanic_df.fillna(method="ffill")
```

[63]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
```

```
4              5          0       3
..           …         …       …
886          887          0       2
887          888          1       1
888          889          0       3
889          890          1       1
890          891          0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female  19.0      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   C85        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500  C123        S
..     …               …      …    …      …
886      0            211536  13.0000   C50        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   B42        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500  C148        Q

[891 rows x 12 columns]
```

[64]:
```python
new_df = titanic_df.fillna(method="ffill",limit=1)
new_df
```

```
C:\Users\skand\AppData\Local\Temp\ipykernel_26088\3418266207.py:1:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a
future version. Use obj.ffill() or obj.bfill() instead.
  new_df = titanic_df.fillna(method="ffill",limit=1)
```

[64]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
```

```
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female  19.0      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   C85        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500  C123        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   B42        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500  C148        Q

[891 rows x 12 columns]
```

[65]:
```python
new_df = titanic_df.fillna(method="bfill")
new_df
```

```
C:\Users\skand\AppData\Local\Temp\ipykernel_26088\348161704.py:1: FutureWarning:
DataFrame.fillna with 'method' is deprecated and will raise in a future version.
Use obj.ffill() or obj.bfill() instead.
  new_df = titanic_df.fillna(method="bfill")
```

[65]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
```

```
2              3         1       3
3              4         1       1
4              5         0       3
..           …       …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                 Name     Sex   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                              Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                            Allen, Mr. William Henry    male  35.0      0
..                                                …     …   …    …
886                             Montvila, Rev. Juozas    male  27.0      0
887                      Graham, Miss. Margaret Edith  female  19.0      0
888          Johnston, Miss. Catherine Helen "Carrie"  female  26.0      1
889                             Behr, Mr. Karl Howell    male  26.0      0
890                               Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   C85        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250  C123        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   E46        S
..     …             …     …   …      …
886      0            211536  13.0000   B42        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500  C148        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

9 Numerosity Data Reduction

[66]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

[67]:
```
customer_df = pd.read_csv('customer_churn.csv')
print(customer_df.shape)
print(customer_df.Churn.value_counts())
```

```
(900, 10)
Churn
0    750
1    150
Name: count, dtype: int64
```

[69]:
```
import pandas as pd

# Example DataFrame creation (replace with your actual DataFrame)
customer_df = pd.DataFrame({
    'Churn': [0, 1] * 500,  # Example data
    'Feature1': range(1000),  # Example feature
    'Feature2': range(1000)   # Example feature
})

# Ensure the sample size is appropriate
sample_size = min(1000, customer_df.shape[0])

# Sample the DataFrame
customer_df_rs = customer_df.sample(sample_size, random_state=1)

# Separate features and target
y = customer_df_rs['Churn']
Xs = customer_df_rs.drop(columns=['Churn'])

print(customer_df_rs.shape)
```

```
(1000, 3)
```

[70]:
```
customer_df_rs
```

[70]:

	Churn	Feature1	Feature2
507	1	507	507
818	0	818	818
452	0	452	452
368	0	368	368
242	0	242	242
..	…	…	…
767	1	767	767
72	0	72	72
908	0	908	908
235	1	235	235
37	1	37	37

```
[1000 rows x 3 columns]
```

[71]:
```
print(customer_df_rs.Churn.value_counts())
```

```
Churn
1    500
0    500
Name: count, dtype: int64
```

10 Stratified Sampling

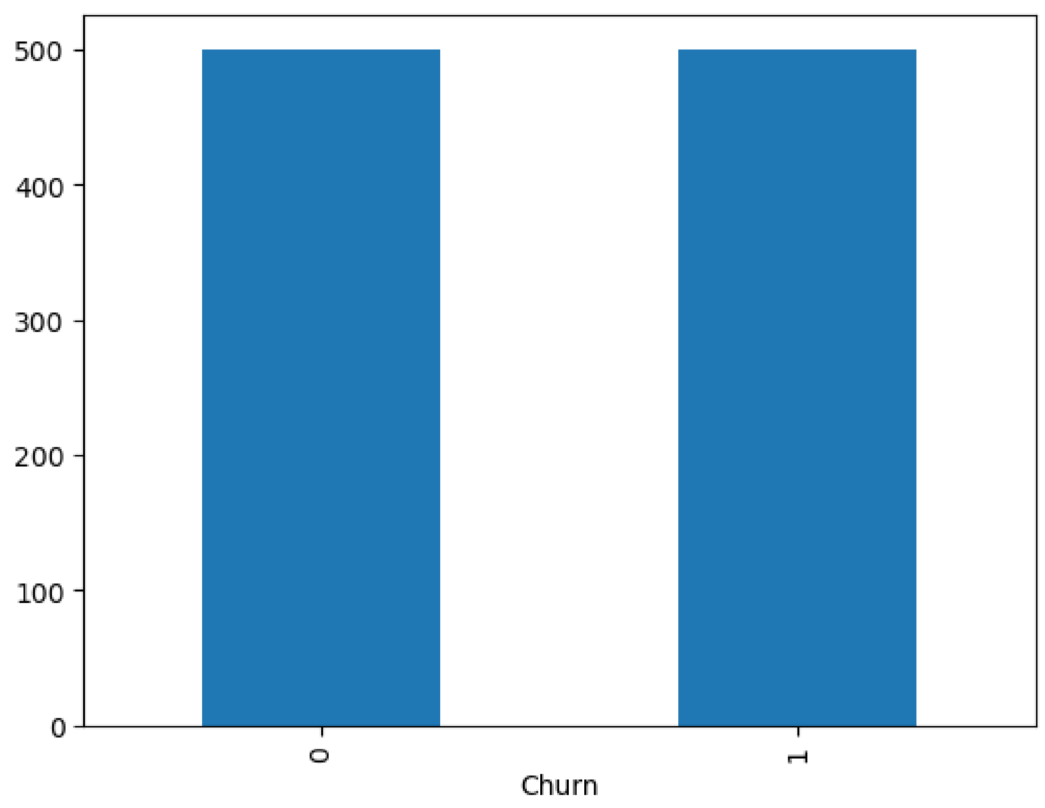
[72]:
```
n,s=len(customer_df),1000
print(n,s)
r = s/n
print('Ratio of each Churn class in sample:',r)
sample_df = customer_df.groupby('Churn').apply(lambda sdf: sdf.
  ↪sample(round(len(sdf))))
print(sample_df.Churn.value_counts())
```

```
1000 1000
Ratio of each Churn class in sample: 1.0
Churn
0    500
1    500
Name: count, dtype: int64
```

```
C:\Users\skand\AppData\Local\Temp\ipykernel_26088\3439575783.py:5:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns.
This behavior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_groups=False`
to exclude the groupings or explicitly select the grouping columns after groupby
to silence this warning.
  sample_df = customer_df.groupby('Churn').apply(lambda sdf:
sdf.sample(round(len(sdf))))
```

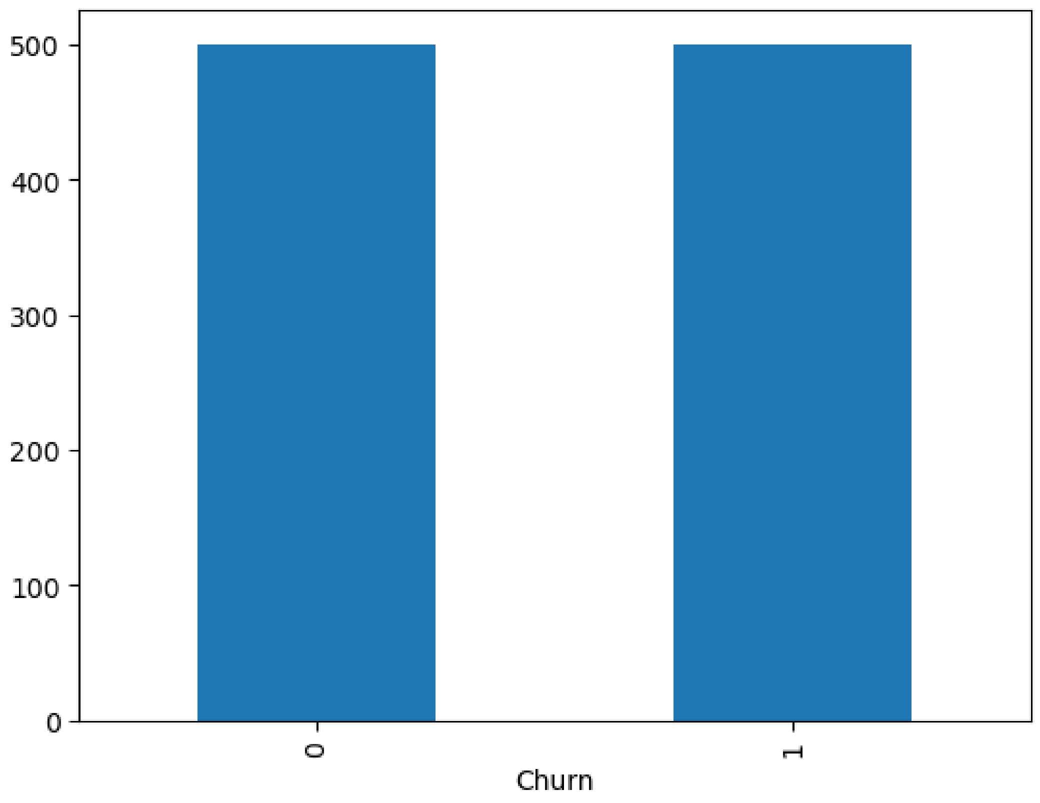
[73]:
```
customer_df.Churn.value_counts().plot.bar()
```

[73]:
```
<Axes: xlabel='Churn'>
```



```
[74]: sample_df.Churn.value_counts().plot.bar()
```

```
[74]: <Axes: xlabel='Churn'>
```



11 Random Over/Under sampling

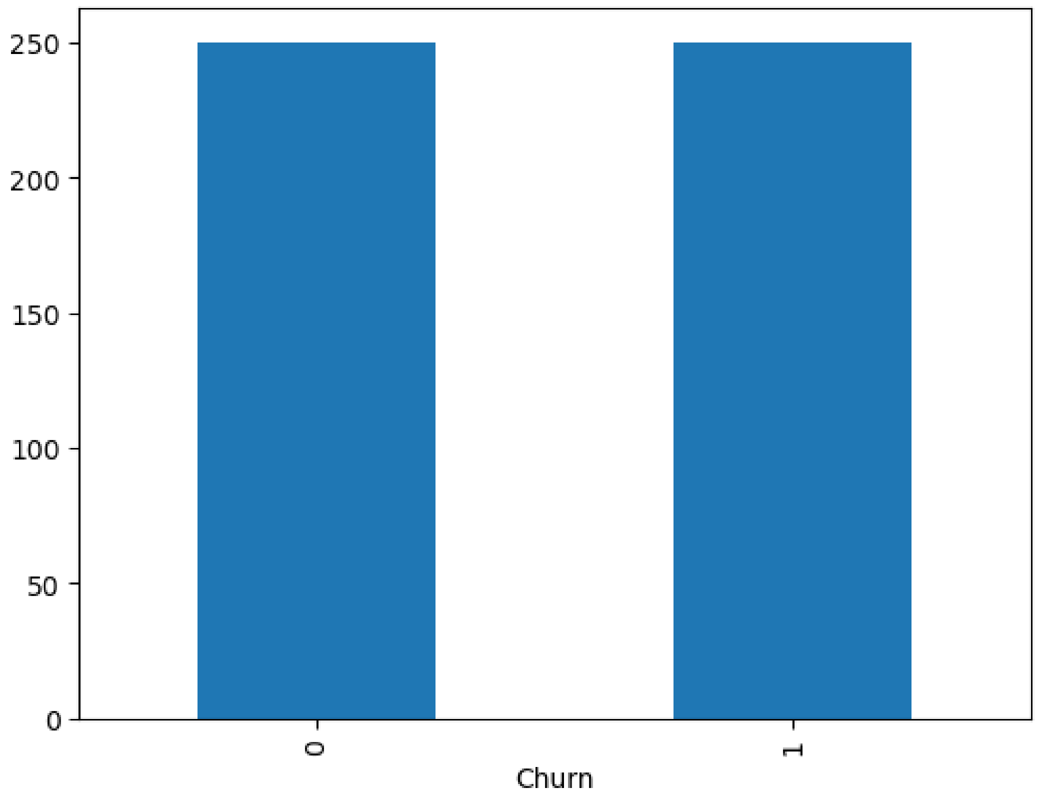
[75]:
```
n,s=len(customer_df),500
sample_df = customer_df.groupby('Churn').apply(lambda sdf: sdf.sample(250))
print(sample_df.Churn.value_counts())
```

```
Churn
0    250
1    250
Name: count, dtype: int64
```

```
C:\Users\skand\AppData\Local\Temp\ipykernel_26088\3198813423.py:2:
DeprecationWarning: DataFrameGroupBy.apply operated on the grouping columns.
This behavior is deprecated, and in a future version of pandas the grouping
columns will be excluded from the operation. Either pass `include_groups=False`
to exclude the groupings or explicitly select the grouping columns after groupby
to silence this warning.
  sample_df = customer_df.groupby('Churn').apply(lambda sdf: sdf.sample(250))
```

[76]:
```
 sample_df.Churn.value_counts().plot.bar()
```

[76]: <Axes: xlabel='Churn'>



[77]:
```
sample_df
```

[77]:
```
             Churn  Feature1  Feature2
Churn
0     136        0       136       136
      134        0       134       134
      892        0       892       892
      784        0       784       784
      970        0       970       970
…            …         …         …
1     383        1       383       383
      305        1       305       305
      559        1       559       559
      115        1       115       115
      11         1        11        11

[500 rows x 3 columns]
```

12 Outliners Detection

[78]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.cm as cm
```

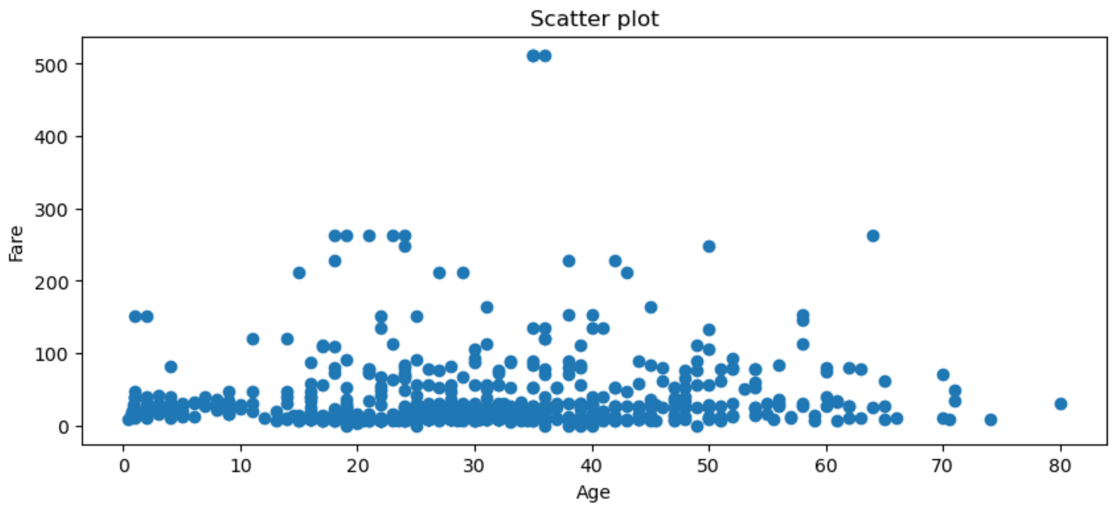
[79]:
```
titanic_df = pd.read_csv("titanic.csv")
titanic_df
```

[79]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           …         …     …
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 …     …   …    …
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     …               …      …   …      …
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
```

```
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

13 Scatter plot to detect outliners

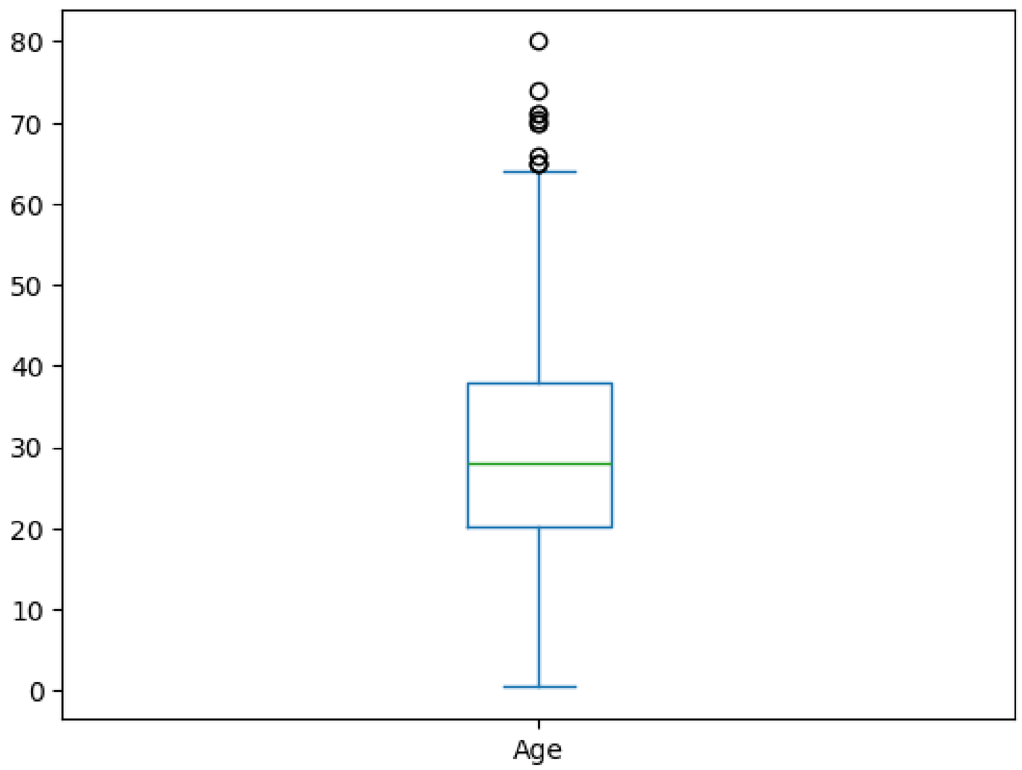
[81]:
```
fig,ax = plt.subplots(figsize=(10,4))
ax.scatter(titanic_df['Age'],titanic_df['Fare'])
ax.set_xlabel('Age')
ax.set_ylabel('Fare')
plt.title("Scatter plot")
plt.show()
```



14 Box plot to detect outliners

[83]:
```
titanic_df['Age'].plot(kind='box')
```

[83]:
```
<Axes: >
```



[85]:
```
q1 = titanic_df["Age"].quantile(0.25)
# finding the 3rd quartile
q3 = titanic_df['Age'].quantile(0.75)
# finding the iqr region
iqr = q3-q1
# finding upper and lower whiskers
upper_bound = q3+(1.5*iqr)
lower_bound = q1-(1.5*iqr)
```

[86]:
```
age_arr = titanic_df["Age"]
outliers = age_arr[(age_arr <= lower_bound) | (age_arr >= upper_bound)]
print('The following are the outliers in the boxplot of age:\n',outliers)
```

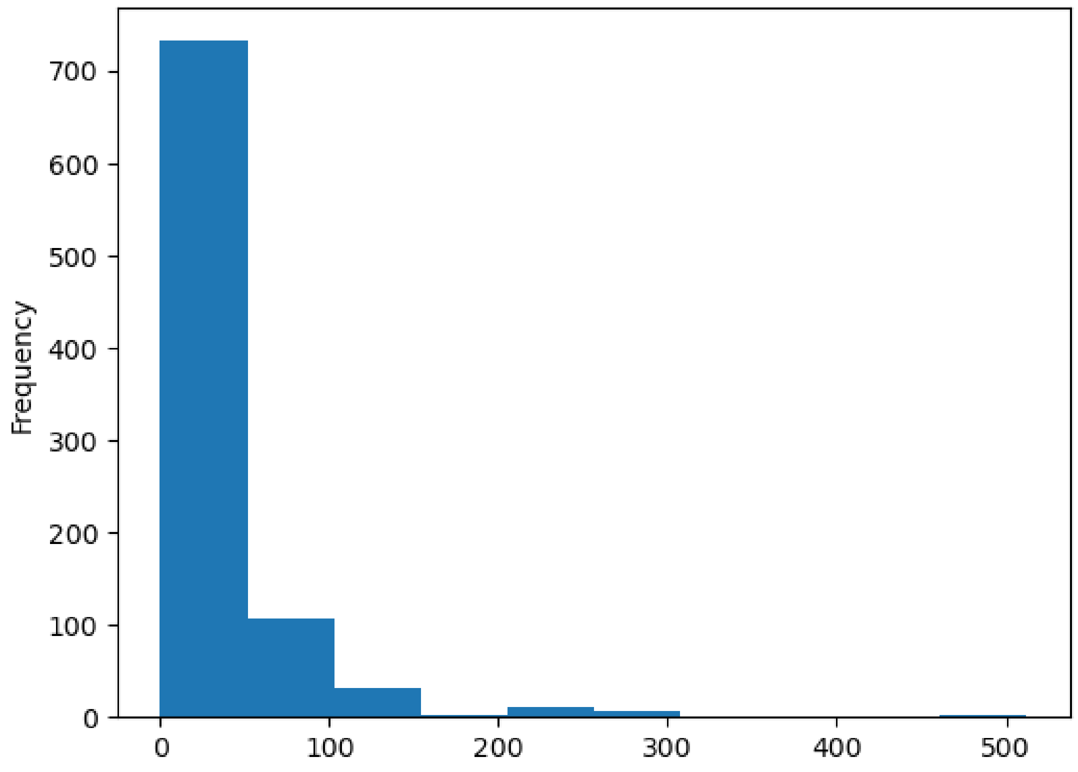
```
The following are the outliers in the boxplot of age:
 33     66.0
54     65.0
96     71.0
116    70.5
280    65.0
456    65.0
493    71.0
630    80.0
```

```
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

15 Histogram to detect outliners

[88]:
```
titanic_df['Fare'].plot(kind='hist')
```

[88]:
```
<Axes: ylabel='Frequency'>
```



16 Remove Data Objects with outliners

[90]:
```
upperIndex = titanic_df[titanic_df['Age']>upper_bound].index
titanic_df.drop(upperIndex,inplace=True)
lowerIndex = titanic_df[titanic_df['Age']<lower_bound].index
titanic_df.drop(lowerIndex,inplace=True)
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 880 entries, 0 to 890
```

```
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  880 non-null    int64
 1   Survived     880 non-null    int64
 2   Pclass       880 non-null    int64
 3   Name         880 non-null    object
 4   Sex          880 non-null    object
 5   Age          703 non-null    float64
 6   SibSp        880 non-null    int64
 7   Parch        880 non-null    int64
 8   Ticket       880 non-null    object
 9   Fare         880 non-null    float64
 10  Cabin        199 non-null    object
 11  Embarked     878 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 89.4+ KB
```

17 Replcing Outliners with upper/lower caps

[91]:
```
titanic_df = pd.read_csv("titanic.csv")
```

[93]:
```
fare_arr = titanic_df["Fare"]
upper_cap = np.percentile(fare_arr,1)
lower_cap = np.percentile(fare_arr,99)
outliers = fare_arr[(fare_arr < upper_cap) | (fare_arr > lower_cap)]
print('The following are the outliers in the boxplot of fare:\n',outliers)
```

```
The following are the outliers in the boxplot of fare:
 27     263.0000
88     263.0000
258    512.3292
311    262.3750
341    263.0000
438    263.0000
679    512.3292
737    512.3292
742    262.3750
Name: Fare, dtype: float64
```

[96]:
```
for i in titanic_df['Fare']:
    if i<lower_bound :
        titanic_df['Fare'] = titanic_df['Fare'].replace(i,lower_cap)
    elif i>upper_bound :
        titanic_df['Fare'] = titanic_df['Fare'].replace(i,upper_cap)
```

[97]:
```
titanic_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
```

18 replacing outliners with mean

[98]:
```
titanic_df = pd.read_csv("titanic.csv")
```

[101]:
```
m = np.mean(titanic_df['Age'])
print('mean:',m)
for i in titanic_df['Age']:
    if i<lower_bound or i>upper_bound :
        titanic_df['Age'] = titanic_df['Age'].replace(i,m)
```

```
mean: 29.69911764705882
```

19 Replacing Outliners with median

[2]:
```
import pandas as pd
titanic_df = pd.read_csv("titanic.csv")
```

[5]:
```
q1 = titanic_df["Age"].quantile(0.25)
 # finding the 3rd quartile
q3 = titanic_df['Age'].quantile(0.75)
 # finding the iqr region
iqr = q3-q1
 # finding upper and lower whiskers
upper_bound = q3+(1.5*iqr)
lower_bound = q1-(1.5*iqr)
```

[8]:
```
m = titanic_df['Age'].median()
print(m)
for i in titanic_df['Age']:
    if i<lower_bound or i>upper_bound :
        titanic_df['Age'] = titanic_df['Age'].replace(i,m)
```

```
28.0
```

[ ]:
```

```