In [1]:

```
#using Pandas start by showing the head of the dataset
import pandas as pd
df=pd.read_csv('titanic-passengers.csv')
df.head()
```

Out[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
343	No	2	Collander	Mr. Erik Gustaf	male	28.0	0	0	248740	
76	No	3	Moen	Mr. Sigurd Hansen	male	25.0	0	0	348123	
641	No	3	Jensen	Mr. Hans Peder	male	20.0	0	0	350050	7.854200
568	No	3	Palsson	Mrs. Nils (Alma Cornelia Berglund)	female	29.0	0	4	349909	
672	No	1	Davidson	Mr. Thornton	male	31.0	1	0	F.C. 12750	

In [2]:

```
#some general information about the data columns and values
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 891 entries, 343 to 428
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    object 
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    object 
 3   Name         891 non-null    object 
 4   Sex          873 non-null    object 
 5   Age          732 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    object 
 8   Ticket       891 non-null    object 
 9   Fare         846 non-null    object 
 10  Cabin        249 non-null    object 
 11  Embarked     836 non-null    object 
dtypes: float64(1), int64(2), object(9)
memory usage: 90.5+ KB
```

In [3]:

```
#Finding missing values
df.isnull().sum()
```

Out[3]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex             18
Age            159
SibSp            0
Parch            0
Ticket           0
Fare            45
Cabin          642
Embarked        55
dtype: int64
```

In [ ]:

```

```

In [4]:

```
df1=df.dropna(axis=0,how='any',thresh=None,inplace=False)
df1.isnull().sum()
```

Out[4]:

```
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
```

In [5]:

```
#number_of_elements per category
print(df1["Sex"].value_counts())
```

```
male      92
female    83
Name: Sex, dtype: int64
```

In [6]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 175 entries, 76 to 699
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  175 non-null    object 
 1   Survived     175 non-null    int64  
 2   Pclass       175 non-null    object 
 3   Name         175 non-null    object 
 4   Sex          175 non-null    object 
 5   Age          175 non-null    float64
 6   SibSp        175 non-null    int64  
 7   Parch        175 non-null    object 
 8   Ticket       175 non-null    object 
 9   Fare         175 non-null    object 
 10  Cabin        175 non-null    object 
 11  Embarked     175 non-null    object 
dtypes: float64(1), int64(2), object(9)
memory usage: 17.8+ KB
```

In [7]:

```
#Categorical to Numerical
print(df1['Survived'].value_counts())
```

```
1    153
2     12
3     10
Name: Survived, dtype: int64
```

In [8]:

```
df1.head()
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
76	No	3	Moen	Mr. Sigurd Hansen	male	25.0	0	0	348123	7.65	F G
672	No	1	Davidson	Mr. Thornton	male	31.0	1	0	F.C. 12750	52.0	E
378	No	1	Widener	Mr. Harry Elkins	male	27.0	0	2	113503	211.5	C
225	Yes	1	Hoyt	Mr. Frederick Maxfield	male	38.0	1	0	19943	90.0	C
588	Yes	1	Frolicher-Stehli	Mr. Maxmillian	male	60.0	1	1	13567	79.2	E

In [9]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df1['Sex']=encoder.fit_transform(df1['Sex'])
df1.head()
```

```
<ipython-input-9-bf0d7ca5ab81>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df1['Sex']=encoder.fit_transform(df1['Sex'])
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
76	No	3	Moen	Mr. Sigurd Hansen	1	25.0	0	0	348123	7.65	F G
672	No	1	Davidson	Mr. Thornton	1	31.0	1	0	F.C. 12750	52.0	B
378	No	1	Widener	Mr. Harry Elkins	1	27.0	0	2	113503	211.5	C
225	Yes	1	Hoyt	Mr. Frederick Maxfield	1	38.0	1	0	19943	90.0	C
588	Yes	1	Frolicher-Stehli	Mr. Maxmillian	1	60.0	1	1	13567	79.2	B

In [10]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df1['Age']=encoder.fit_transform(df1['Age'])
df1.head()
```

```
<ipython-input-10-455ce9413233>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df1['Age']=encoder.fit_transform(df1['Age'])
```

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
76	No	3	Moen	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	F G
672	No	1	Davidson	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	B
378	No	1	Widener	Mr. Harry Elkins	1	19	0	2	113503	211.5	C
225	Yes	1	Hoyt	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	C
588	Yes	1	Frolicher-Stehli	Mr. Maxmillian	1	53	1	1	13567	79.2	B

In [11]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df1['Survived']=encoder.fit_transform(df1['Survived'])
df1.head()
```

```
<ipython-input-11-2d53ad516555>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df1['Survived']=encoder.fit_transform(df1['Survived'])
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
76	No	2	Moen	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	F G7
672	No	0	Davidson	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	B7
378	No	0	Widener	Mr. Harry Elkins	1	19	0	2	113503	211.5	C8
225	Yes	0	Hoyt	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	C9
588	Yes	0	Frolicher-Stehli	Mr. Maxmillian	1	53	1	1	13567	79.2	B4

In [12]:

```
df2=df1.dropna(axis=1,how='any',thresh=None,inplace=False)
df2.head()
```

Out[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
76	No	2	Moen	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	F G7
672	No	0	Davidson	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	B7
378	No	0	Widener	Mr. Harry Elkins	1	19	0	2	113503	211.5	C8
225	Yes	0	Hoyt	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	C9
588	Yes	0	Frolicher-Stehli	Mr. Maxmillian	1	53	1	1	13567	79.2	B4

In [13]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Pclass']=encoder.fit_transform(df2['Pclass'])
df2.head()
```

Out[13]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
76	No	2	87	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	F G73
672	No	0	38	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	B71
378	No	0	134	Mr. Harry Elkins	1	19	0	2	113503	211.5	C82
225	Yes	0	65	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	C93
588	Yes	0	48	Mr. Maxmillian	1	53	1	1	13567	79.2	B41

In [14]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Cabin']=encoder.fit_transform(df2['Cabin'])
df2.head()
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
76	No	2	87	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	124
672	No	0	38	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	31
378	No	0	134	Mr. Harry Elkins	1	19	0	2	113503	211.5	68
225	Yes	0	65	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	75
588	Yes	0	48	Mr. Maxmillian	1	53	1	1	13567	79.2	22

In [15]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 175 entries, 76 to 699
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   PassengerId  175 non-null    object
 1   Survived     175 non-null    int64 
 2   Pclass       175 non-null    int32 
 3   Name         175 non-null    object
 4   Sex          175 non-null    int32 
 5   Age          175 non-null    int64 
 6   SibSp        175 non-null    int64 
 7   Parch        175 non-null    object
 8   Ticket       175 non-null    object
 9   Fare         175 non-null    object
 10  Cabin        175 non-null    int32 
 11  Embarked     175 non-null    object
dtypes: int32(3), int64(3), object(6)
memory usage: 15.7+ KB
```

In [16]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['PassengerId']=encoder.fit_transform(df2['PassengerId'])
df2.head()
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
76	0	2	87	Mr. Sigurd Hansen	1	17	0	0	348123	7.65	124
672	0	0	38	Mr. Thornton	1	23	1	0	F.C. 12750	52.0	31
378	0	0	134	Mr. Harry Elkins	1	19	0	2	113503	211.5	68
225	1	0	65	Mr. Frederick Maxfield	1	31	1	0	19943	90.0	75
588	1	0	48	Mr. Maxmillian	1	53	1	1	13567	79.2	22

In [17]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Name']=encoder.fit_transform(df2['Name'])
df2.head()
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
76	0	2	87	114	1	17	0	0	348123	7.65	124	
672	0	0	38	120	1	23	1	0	F.C. 12750	52.0	31	
378	0	0	134	87	1	19	0	2	113503	211.5	68	
225	1	0	65	80	1	31	1	0	19943	90.0	75	
588	1	0	48	103	1	53	1	1	13567	79.2	22	

In [18]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Ticket']=encoder.fit_transform(df2['Ticket'])
df2.head()
```

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
76	0	2	87	114	1	17	0	0	75	7.65	124	
672	0	0	38	120	1	23	1	0	89	52.0	31	
378	0	0	134	87	1	19	0	2	18	211.5	68	
225	1	0	65	80	1	31	1	0	61	90.0	75	
588	1	0	48	103	1	53	1	1	49	79.2	22	

In [19]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 175 entries, 76 to 699
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  175 non-null    int32
 1   Survived     175 non-null    int64
 2   Pclass       175 non-null    int32
 3   Name         175 non-null    int32
 4   Sex          175 non-null    int32
 5   Age          175 non-null    int64
 6   SibSp        175 non-null    int64
 7   Parch        175 non-null    object
 8   Ticket       175 non-null    int32
 9   Fare         175 non-null    object
 10  Cabin        175 non-null    int32
 11  Embarked     175 non-null    object
dtypes: int32(6), int64(3), object(3)
memory usage: 13.7+ KB
```

In [20]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Parch']=encoder.fit_transform(df2['Parch'])
df2.head()
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
76	0	2	87	114	1	17	0	0	75	7.65	124	
672	0	0	38	120	1	23	1	0	89	52.0	31	
378	0	0	134	87	1	19	0	2	18	211.5	68	
225	1	0	65	80	1	31	1	0	61	90.0	75	
588	1	0	48	103	1	53	1	1	49	79.2	22	

In [21]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Fare']=encoder.fit_transform(df2['Fare'])
df2.head()
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Eml
76	0	2	87	114	1	17	0	0	75	69	124	
672	0	0	38	120	1	23	1	0	89	55	31	
378	0	0	134	87	1	19	0	2	18	20	68	
225	1	0	65	80	1	31	1	0	61	87	75	
588	1	0	48	103	1	53	1	1	49	79	22	

In [22]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 175 entries, 76 to 699
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  175 non-null    int32
 1   Survived     175 non-null    int64
 2   Pclass       175 non-null    int32
 3   Name         175 non-null    int32
 4   Sex          175 non-null    int32
 5   Age          175 non-null    int64
 6   SibSp        175 non-null    int64
 7   Parch        175 non-null    int32
 8   Ticket       175 non-null    int32
 9   Fare         175 non-null    int32
 10  Cabin        175 non-null    int32
 11  Embarked     175 non-null    object
dtypes: int32(8), int64(3), object(1)
memory usage: 12.3+ KB
```

In [23]:

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df2['Embarked']=encoder.fit_transform(df2['Embarked'])
df2.head()
```

Out[23]:

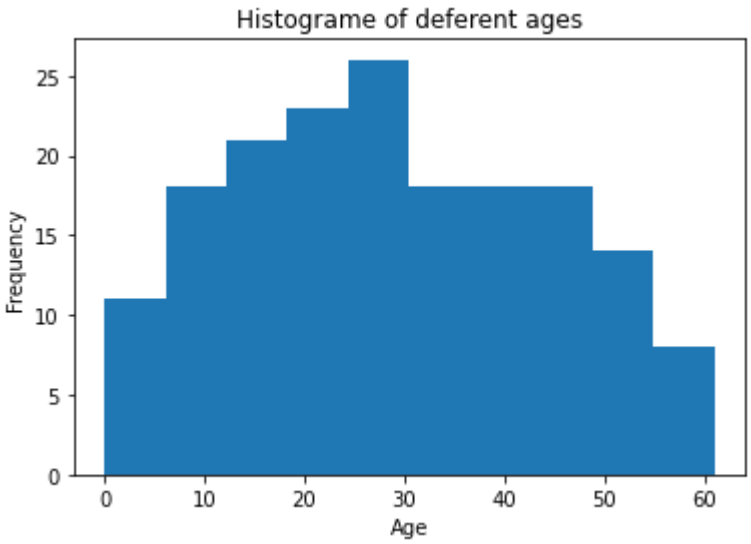
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
76	0	2	87	114	1	17	0	0	75	69	124	
672	0	0	38	120	1	23	1	0	89	55	31	
378	0	0	134	87	1	19	0	2	18	20	68	
225	1	0	65	80	1	31	1	0	61	87	75	
588	1	0	48	103	1	53	1	1	49	79	22	

In [24]:

```
import pandas as pd
import matplotlib.pyplot as plt
plt.title("Histograme of deferent ages")
plt.xlabel("Age")
df2['Age'].plot.hist()
```

Out[24]:

```
<AxesSubplot:title={'center':'Histograme of deferent ages'}, xlabel='Age', y
label='Frequency'>
```

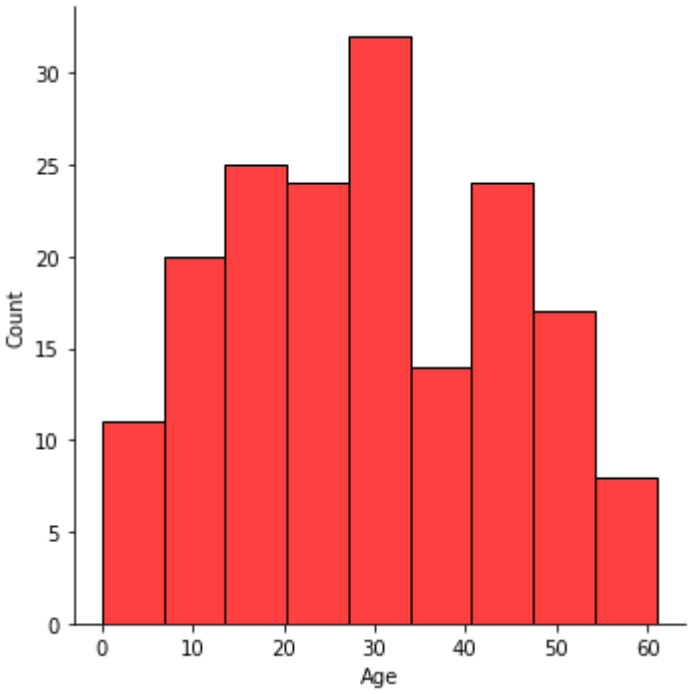


In [25]:

```
import seaborn as sns
sns.displot(df2['Age'],color="red")
```

Out[25]:

```
<seaborn.axisgrid.FacetGrid at 0xf79e12a940>
```

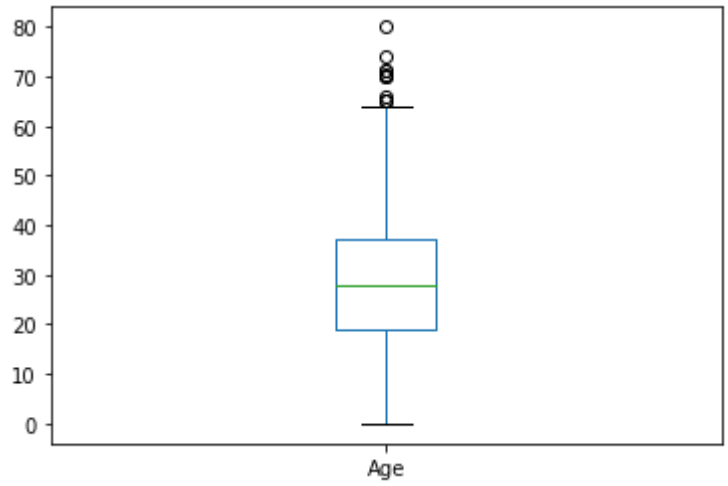


In [26]:

```
df['Age'].plot.box()
```

Out[26]:

```
<AxesSubplot:>
```

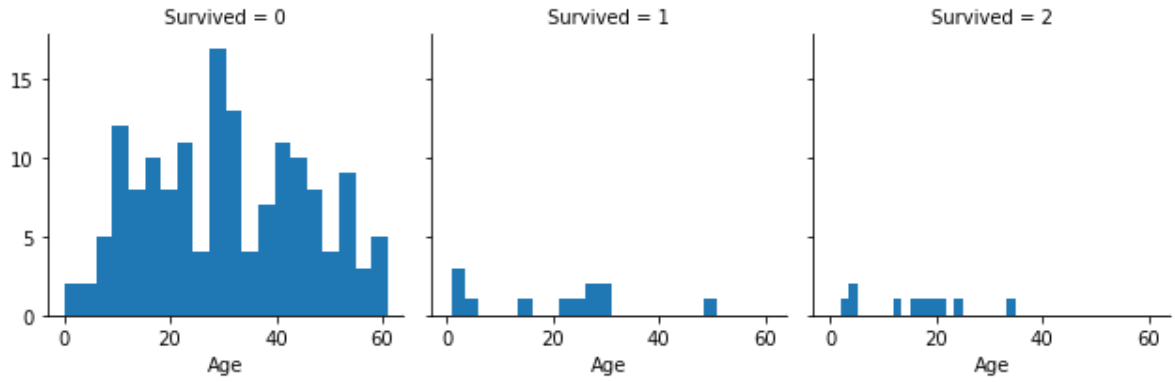


In [27]:

```
g=sns.FacetGrid(df2,col='Survived')
g.map(plt.hist,'Age',bins=20)
```

Out[27]:

```
<seaborn.axisgrid.FacetGrid at 0xf79e868310>
```

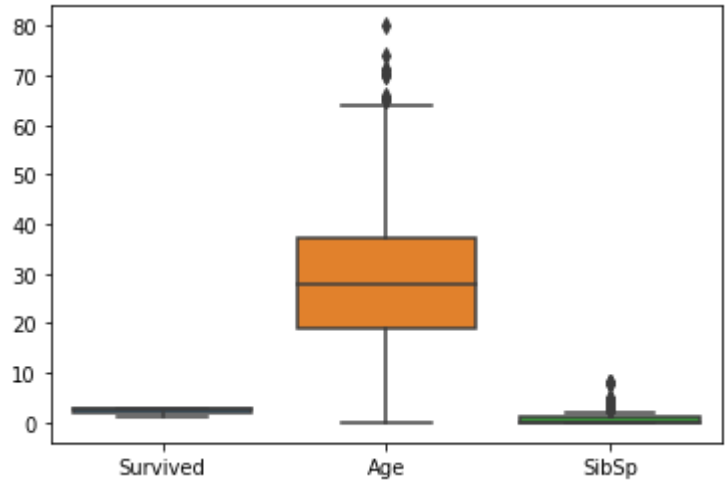


In [28]:

```
#Standardization
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
df2['Age']=scaler.fit_transform(df2[['Age']].values)
df2['Survived']=scaler.fit_transform(df2[['Survived']].values)
sns.boxplot(data=df)
```

Out[28]:

<AxesSubplot:>

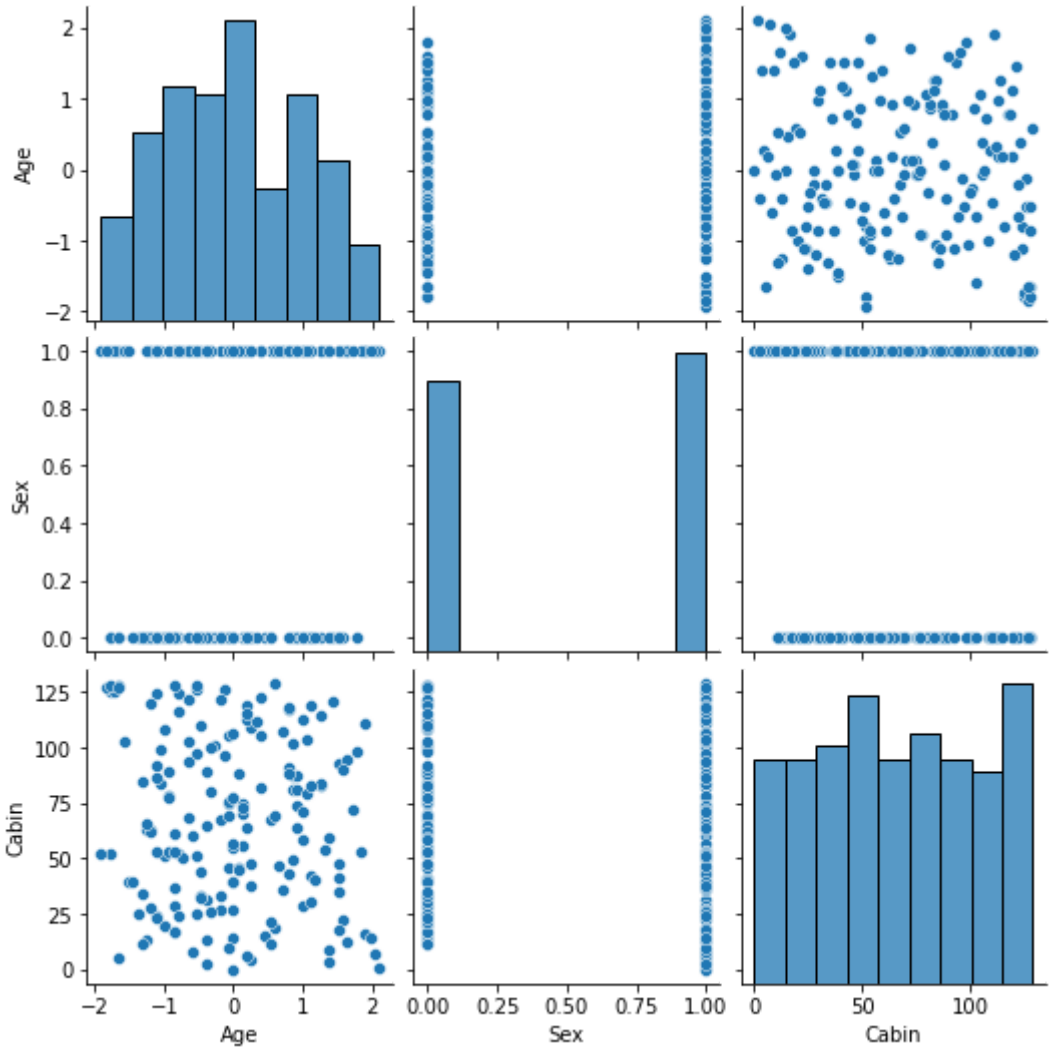


In [29]:

```
sns.pairplot(df2[['Age','Sex','Cabin']])
```

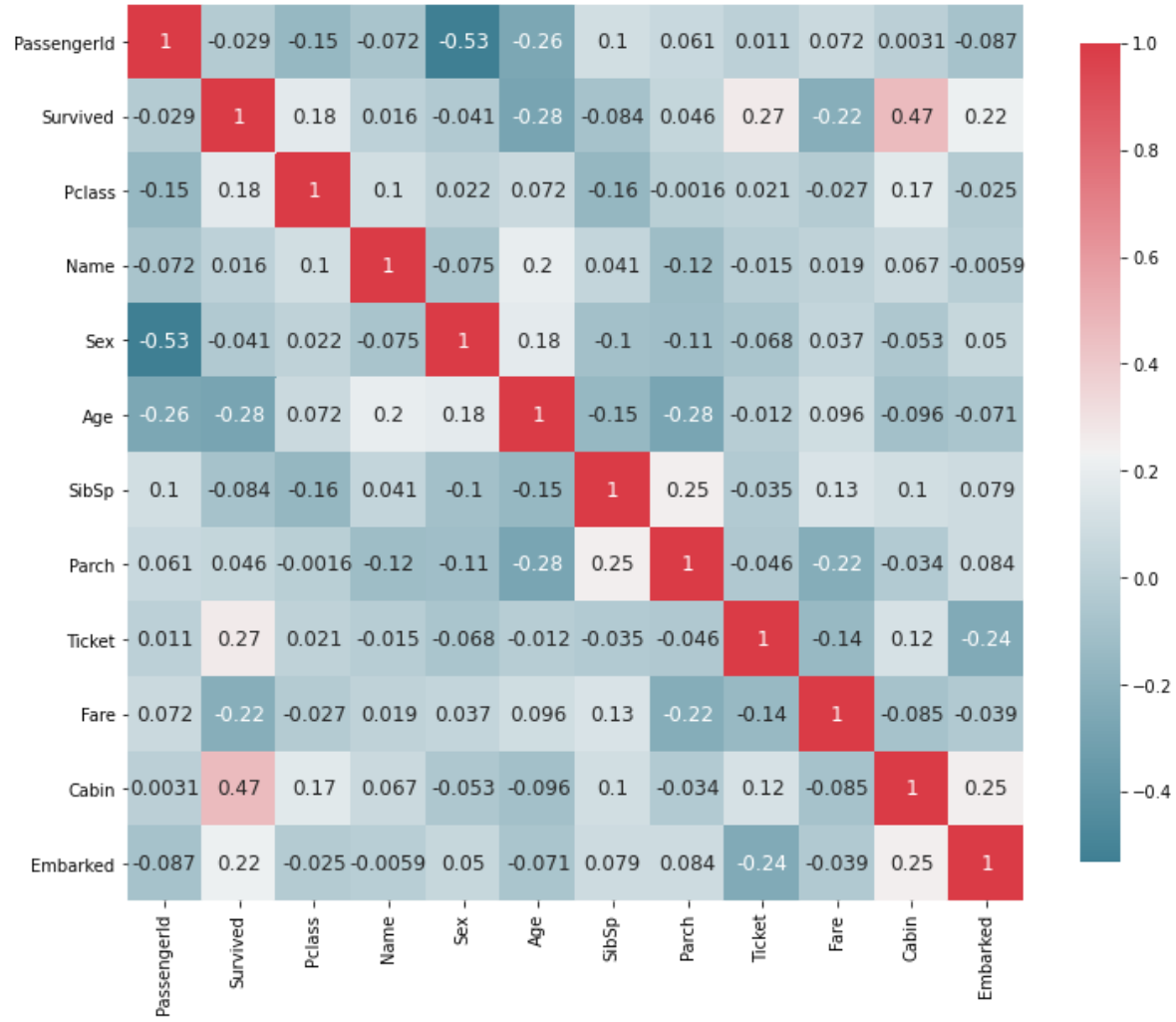
Out[29]:

```
<seaborn.axisgrid.PairGrid at 0xf79ea795e0>
```



In [35]:

```
# visualize the correlation matrix
def plot_correlation_map(df):

    corr = df.corr()

    s , ax = plt.subplots( figsize =( 12 , 10 ) )

    cmap = sns.diverging_palette( 220 , 10 , as_cmap = True )

    s = sns.heatmap(

        corr,

        cmap = cmap,

        square=True,

        cbar_kws={ 'shrink' : .9 },

        ax=ax,

        annot = True,

        annot_kws = { 'fontsize' : 12 }

        )
plot_correlation_map(df2)
```



In [ ]:

```
```