```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv('/content/Titanic-Dataset.csv')
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
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	
				Heikkinen								

```
data.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
```

```
data.describe()
```

	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

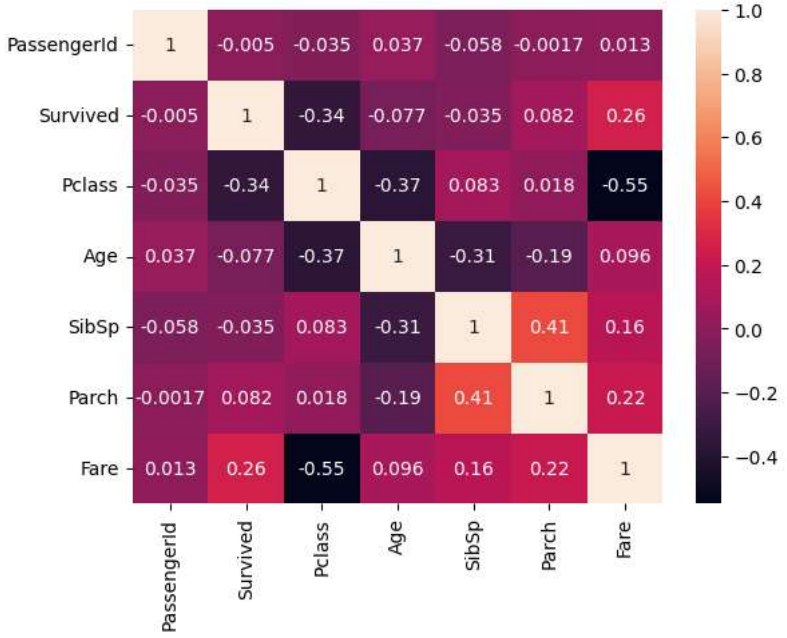
```
corr=data.corr()
corr
```

```
<ipython-input-5-0d3ae1d0be10>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
  corr=data.corr()
```

	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

```
sns.heatmap(corr,annot=True)
```

```
<Axes: >
```



```
data.Cabin.value_counts()
```

```
B96 B98        4
G6             4
C23 C25 C27    4
C22 C26        3
F33            3
              ..
E34            1
C7             1
C54            1
E36            1
C148           1
Name: Cabin, Length: 147, dtype: int64
```

```
data.Embarked.value_counts()
```

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

```
data.Parch.value_counts()
```

```
0    678
1    118
2     80
5      5
3      5
4      4
6      1
Name: Parch, dtype: int64
```

```
data.isnull().any()
```

```
PassengerId    False
Survived       False
Pclass         False
Name           False
```

```
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

```
data.isnull().sum()
```

```
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
```

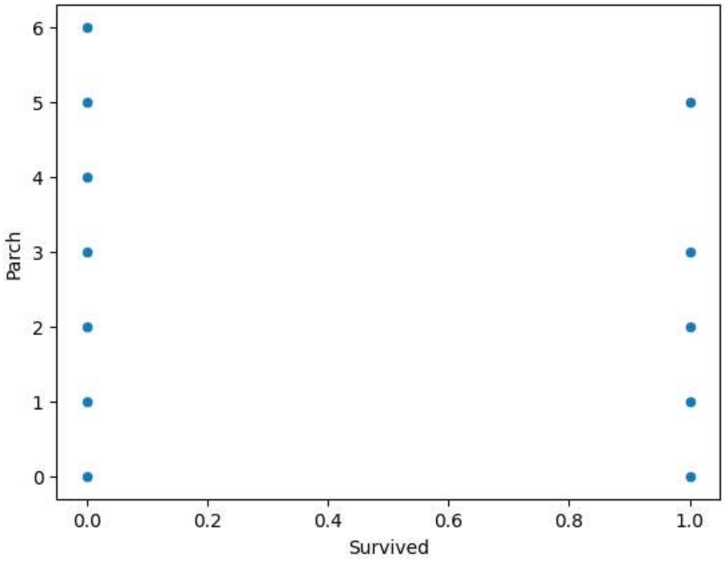
```
data["Age"].fillna(data["Age"].mean(),inplace=True)
data["Cabin"].fillna(data["Cabin"].mode()[0],inplace=True)
data["Embarked"].fillna(data["Embarked"].mode()[0],inplace=True)
```

```
data.isnull().sum()#I removed all null values
```

```
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
```

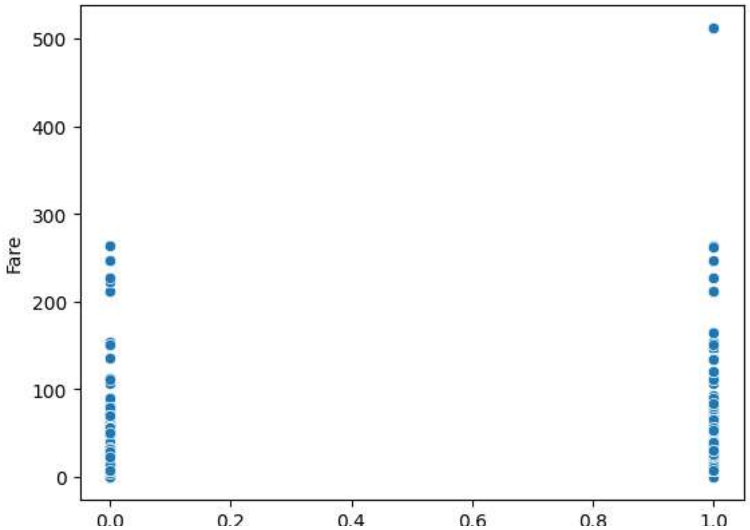
```
sns.scatterplot(x=data["Survived"],y=data["Parch"])
```

```
<Axes: xlabel='Survived', ylabel='Parch'>
```



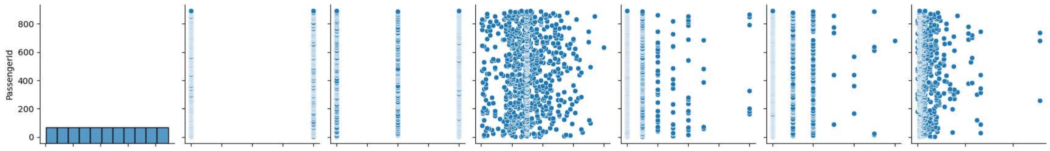
```
sns.scatterplot(x=data["Survived"],y=data["Fare"])
```

```
<Axes: xlabel='Survived', ylabel='Fare'>
```



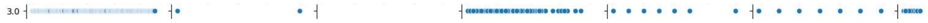
```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7c726f239390>
```



```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
data["Sex"]=le.fit_transform(data["Sex"])
```



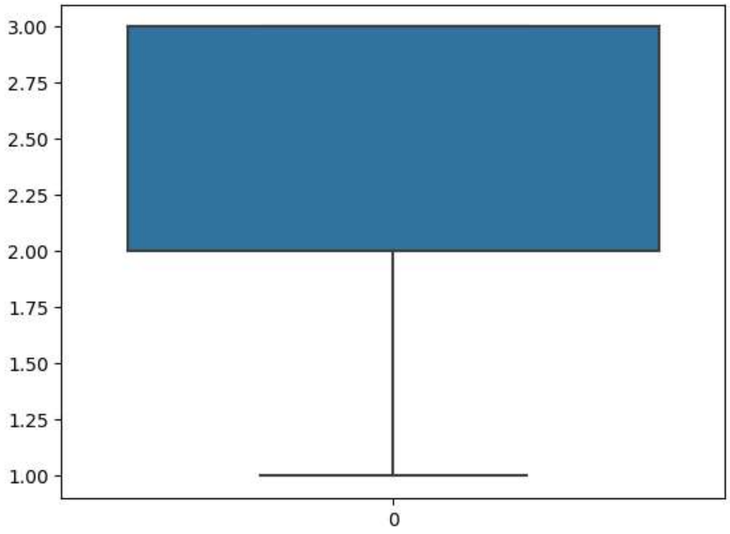
```
data["Embarked"]=le.fit_transform(data["Embarked"])
```

```
data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S

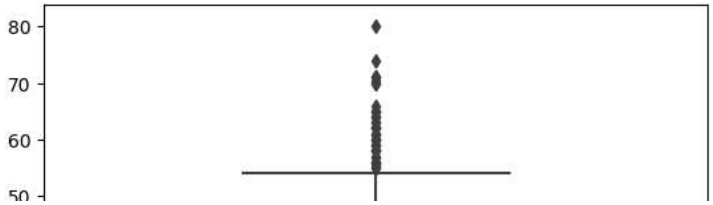
```
sns.boxplot(data['Pclass'])
```

```
<Axes: >
```



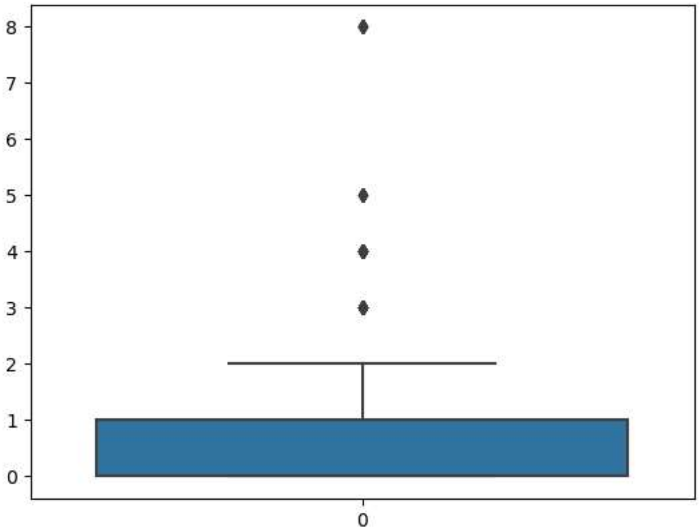
```
sns.boxplot(data['Age'])
```

```
<Axes: >
```



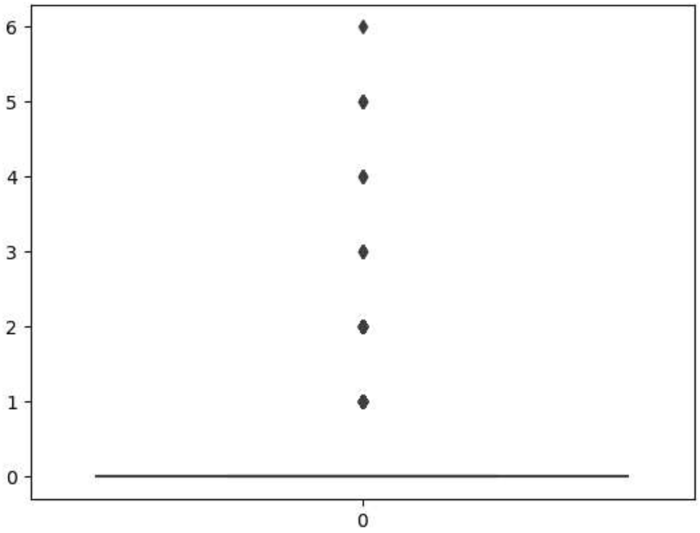
```
sns.boxplot(data['SibSp'])
```

```
<Axes: >
```



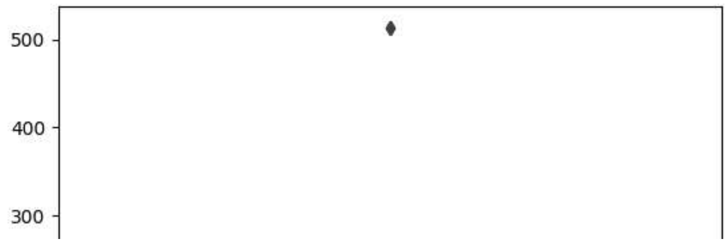
```
sns.boxplot(data['Parch'])
```

```
<Axes: >
```



```
sns.boxplot(data['Fare'])
```

```
<Axes: >
```



```
sns.boxplot(data['Embarked'])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-24-50a4e8c6f085> in <cell line: 1>()
----> 1 sns.boxplot(data['Embarked'])

                    4 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/series.py in __array__(self, dtype)
    891                 dtype='datetime64[ns]')
    892             """
--> 893             return np.asarray(self._values, dtype)
    894 
    895     # ----------------------------------------------------------------------

ValueError: could not convert string to float: 'S'
```

SEARCH STACK OVERFLOW

```
q1=data.Age.quantile(0.25)
q3=data.Age.quantile(0.75)
print(q1)
print(q3)
```

```
22.0
35.0
```

```
iqr=q3-q1
iqr
```

```
13.0
```

```
upperlimit = q3+1.5*iqr
upperlimit
```

```
54.5
```

```
lowerlimit=q1-1.5*iqr
lowerlimit
```

```
2.5
```

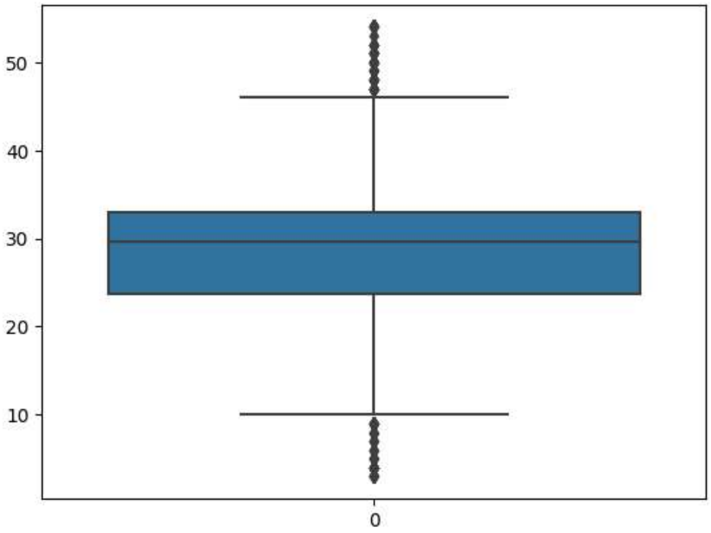
```
data.median()
```

```
<ipython-input-29-135339ac59ce>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future vers
  data.median()
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

```
data['Age']=np.where(data['Age']>upperlimit,29.699118,data['Age'])
data['Age'] = np.where(data['Age'] < lowerlimit,29.699118, data['Age'])
```

```
sns.boxplot(data['Age'])
```

```
<Axes: >
```



```
q1=data.SibSp.quantile(0.25)
q3=data.SibSp.quantile(0.75)
print(q1)
print(q3)
```

```
0.0
1.0
```

```
iqr=q3-q1
iqr
```

```
1.0
```

```
upperlimit = q3+1.5*iqr
upperlimit
```

```
2.5
```

```
lowerlimit=q1-1.5*iqr
lowerlimit
```

```
-1.5
```

```
data['SibSp']=np.where(data['SibSp']>upperlimit,0.000000,data['SibSp'])
```

```
sns.boxplot(data['SibSp'])
```

```
q1=data.Parch.quantile(0.25)
q3=data.Parch.quantile(0.75)
print(q1)
print(q3)
```

```
0.0
0.0
```

```
iqr=q3-q1
iqr
```

```
0.0
```

```
upperlimit = q3+1.5*iqr
upperlimit
```

```
0.0
```

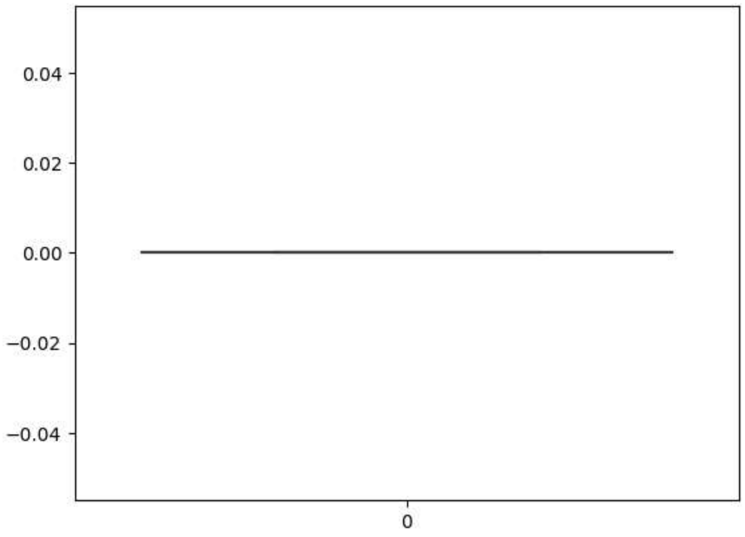
```
lowerlimit=q1-1.5*iqr
lowerlimit
```

```
0.0
```

```
data['Parch']=np.where(data['Parch']>upperlimit,0.000000,data['Parch'])
```

```
sns.boxplot(data['Parch'])
```

```
<Axes: >
```



```
q1=data.Fare.quantile(0.25)
q3=data.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
31.0
```

```
iqr=q3-q1
iqr
```

```
23.0896
```

```
upperlimit = q3+1.5*iqr
upperlimit
```

```
65.6344
```

```
lowerlimit=q1-1.5*iqr
lowerlimit
```

```
-26.724
```

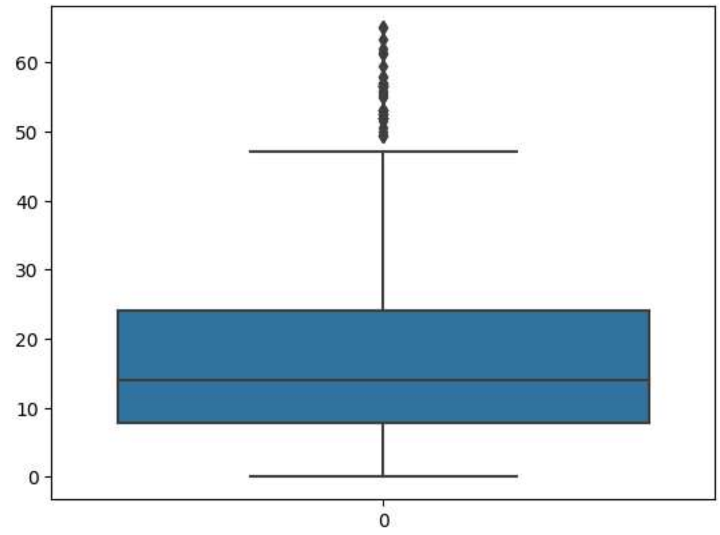
```
data.median()
```

```
<ipython-input-49-135339ac59ce>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future vers
  data.median()
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

```
data['Fare']=np.where(data['Fare']>upperlimit,14.054150,data['Fare'])
```

```
sns.boxplot(data.Fare)
```

```
<Axes: >
```



```
y=data["Survived"]
```

```
X=data.drop(columns=["Name","PassengerId","Survived","Ticket","Cabin"],axis=1)
```

```
y.head()
```

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

```
X_Scaled=ms.fit_transform(X)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-57-8621e43fe6dc> in <cell line: 1>()
----> 1 X_Scaled=ms.fit_transform(X)

  ⌃⌄ 7 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in __array__(self, dtype)
   2068
```

```
X_Scaled=pd.DataFrame(ms.fit_transform(X),columns=X.columns)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-58-d92c04273673> in <cell line: 1>()
----> 1 X_Scaled=pd.DataFrame(ms.fit_transform(X),columns=X.columns)

  ⌃⌄ 7 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in __array__(self, dtype)
   2068
   2069     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.ndarray:
-> 2070         return np.asarray(self._values, dtype=dtype)
   2071
   2072     def __array_wrap__(

ValueError: could not convert string to float: 'male'
```

SEARCH STACK OVERFLOW

```
X_Scaled.head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-59-717f179f34cc> in <cell line: 1>()
----> 1 X_Scaled.head()

NameError: name 'X_Scaled' is not defined
```

SEARCH STACK OVERFLOW

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X_Scaled,y,test_size =0.2,random_state =0)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-60-fdc851923b8c> in <cell line: 2>()
      1 from sklearn.model_selection import train_test_split
----> 2 x_train,x_test,y_train,y_test = train_test_split(X_Scaled,y,test_size =0.2,random_state =0)

NameError: name 'X_Scaled' is not defined
```

SEARCH STACK OVERFLOW

```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-62-08fa712edb3b> in <cell line: 1>()
----> 1 print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)

NameError: name 'x_train' is not defined
```

SEARCH STACK OVERFLOW