```
Start coding or generate with AI.
```

Write a Program to determine the following in the Titanic Survival data.

Determine the data type of each column.

Double-click (or enter) to edit

```
# importing all the necessary libraries
import pandas as pd
import numpy as np
#we need to read the data
data = pd.read_csv("/content/drive/MyDrive/AI Tools Lab/train.csv")
#print top 5 rows
print(data.head())
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         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

   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
```

```
# to get the datatype of all columns we can use Dataframe.dtypes
print(data.dtypes)
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
```

```
Embarked        object
dtype: object
```

Find the number of non-null values in each column.

```
# Dataframe.info() gives all information about every column in our dataset
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
```

Start coding or generate with AI.

Find out the unique values in each categorical column and frequency of each unique value.

Start coding or generate with AI.

```
# categorical is nothing but the datatype which is other than numerical datatype (i.e int
# to get the all categorical columns, we can use Dataframe.select_dtypes and we have to s
#datatype we required.
# In our case it would be "object" datatype
categorical_cols = data.select_dtypes(include=['object']).columns.tolist()
print("Categorical columns are : ",categorical_cols)
print("printing the results")
for i in categorical_cols:
 print("========== Column '"+i+"' =============")
 print(data[i].value_counts())
```

```
Categorical columns are :  ['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
printing the results
========== Column 'Name' =============
Name
Braund, Mr. Owen Harris                      1
Boulos, Mr. Hanna                            1
```

```
Frolicher-Stehli, Mr. Maxmillian            1
Gilinski, Mr. Eliezer                       1
Murdlin, Mr. Joseph                         1
                                           ..
Kelly, Miss. Anna Katherine "Annie Kate"    1
McCoy, Mr. Bernard                          1
Johnson, Mr. William Cahoone Jr             1
Keane, Miss. Nora A                         1
Dooley, Mr. Patrick                         1
Name: count, Length: 891, dtype: int64
========== Column 'Sex' =============
Sex
male      577
female    314
Name: count, dtype: int64
========== Column 'Ticket' =============
Ticket
347082      7
CA. 2343    7
1601        7
3101295     6
CA 2144     6
           ..
9234        1
19988       1
2693        1
PC 17612    1
370376      1
Name: count, Length: 681, dtype: int64
========== Column 'Cabin' =============
Cabin
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: count, Length: 147, dtype: int64
========== Column 'Embarked' =============
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

```
Start coding or generate with AI.
```

d. Find the number of rows where age is greater than the mean age of data.

```
# to get mean of age column
age_mean = data['Age'].mean()
print("Mean of Age is : ",age_mean)
print("printing the result")
print(np.sum(data['Age']>age_mean))
```

```
Mean of Age is :  29.69911764705882
printing the result
330
```

Start coding or generate with AI.

e. Delete all the rows with missing values.

```
print("length of dataframe before deleting rows with missing values",len(data))
# deletes the rows where at least one element is missing
data.dropna(inplace=True)
print("length of dataframe after the deletion of missing value rows",len(data))
```

```
length of dataframe before deleting rows with missing values 891
length of dataframe after the deletion of missing value rows 183
```

```
data.info()
```

Start coding or generate with AI.

Find the number of rows where age is greater than the mean age of data.

```
# to get mean of age column
age_mean = data['Age'].mean()
print("Mean of Age is : ",age_mean)
print("printing the result")
print(np.sum(data['Age']>age_mean))
```

```
Mean of Age is :  35.6744262295082
printing the result
93
```

```
import pandas as pd
import numpy as np
df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/boo.csv")
print(df)
```

```
  name  rollno Gender
0    A    11.0      M
1    B    22.0    NaN
2    C    33.0      F
3    D     NaN      F
```

```
print("length of dataframe before deleting rows with missing values",len(df))
# deletes the rows where at least one element is missing
print(df)
print(df.dropna())
print(df)
print("length of dataframe after the deletion of missing value rows",len(df))
```

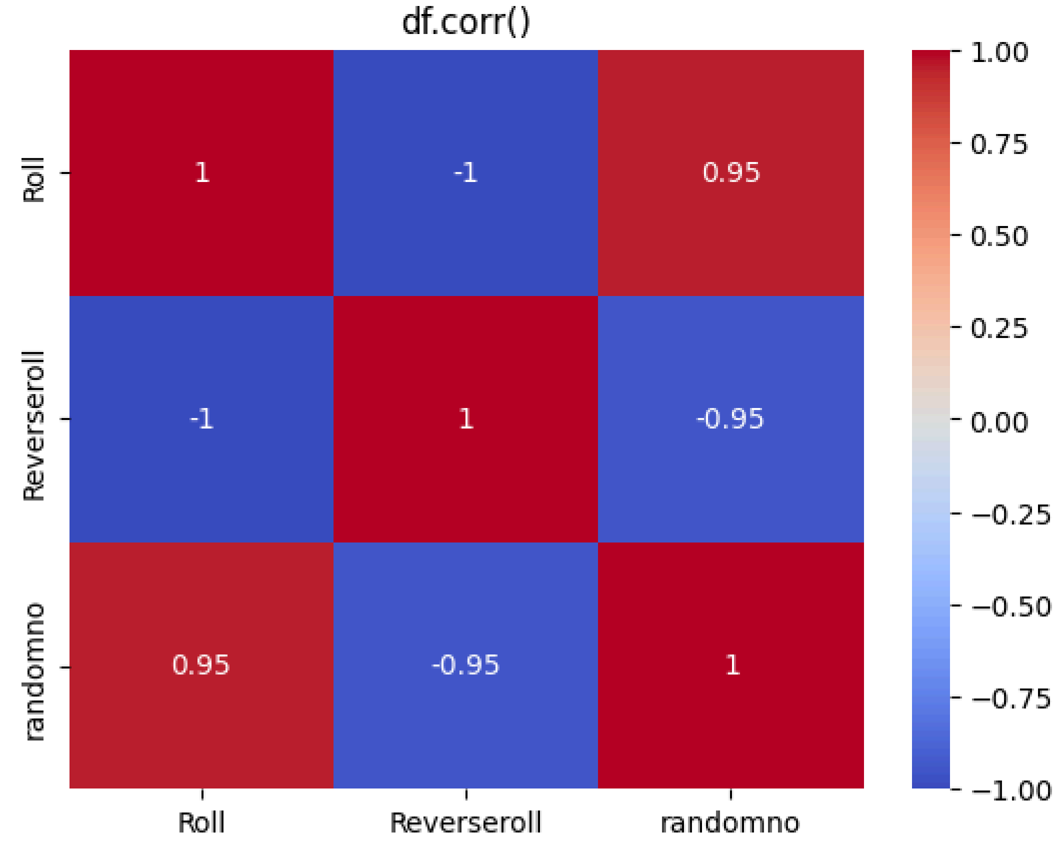
```
length of dataframe before deleting rows with missing values 4
  name  rollno Gender
0    A    11.0      M
1    B    22.0    NaN
2    C    33.0      F
3    D     NaN      F
  name  rollno Gender
0    A    11.0      M
2    C    33.0      F
  name  rollno Gender
0    A    11.0      M
1    B    22.0    NaN
2    C    33.0      F
3    D     NaN      F
length of dataframe after the deletion of missing value rows 4
```

```
print(df)
print(df.dropna(inplace=True))
print(df)
```

```
  name  rollno Gender
0    A    11.0      M
1    B    22.0    NaN
2    C    33.0      F
3    D     NaN      F
None
  name  rollno Gender
0    A    11.0      M
2    C    33.0      F
```

```
#Correlation between each column
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/numbercorr.csv")
print(df)
print(df.corr())
sns.heatmap(df.corr(),cmap='coolwarm',xticklabels=True,annot=True)
plt.title('df.corr()')
```

```
   Roll  Reverseroll  randomno
0     1           10         2
1     2            9         3
2     3            8         1
3     4            7         4
4     5            6         5
5     6            5         7
6     7            4         6
7     8            3         8
8     9            2         9
9    10            1        10
                 Roll  Reverseroll  randomno
Roll         1.000000    -1.000000  0.951515
Reverseroll -1.000000     1.000000 -0.951515
randomno     0.951515    -0.951515  1.000000
Text(0.5, 1.0, 'df.corr()')
```



Start coding or generate with AI.

Perform Data Analysis on the Titanic Data Set to answer the following.

```
df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/boo.csv")
print(df)
df.corr()
```

```
  name  rollno Gender
0    A    11.0      M
1    B    22.0    NaN
2    C    33.0      F
3    D     NaN      F
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-24-a710215580fc> in <cell line: 3>()
      1 df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/boo.csv")
      2 print(df)
----> 3 df.corr()

                              3 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in
_interleave(self, dtype, na_value)
   1792             else:
   1793                 arr = blk.get_values(dtype)
-> 1794             result[rl.indexer] = arr
   1795             itemmask[rl.indexer] = 1
   1796

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

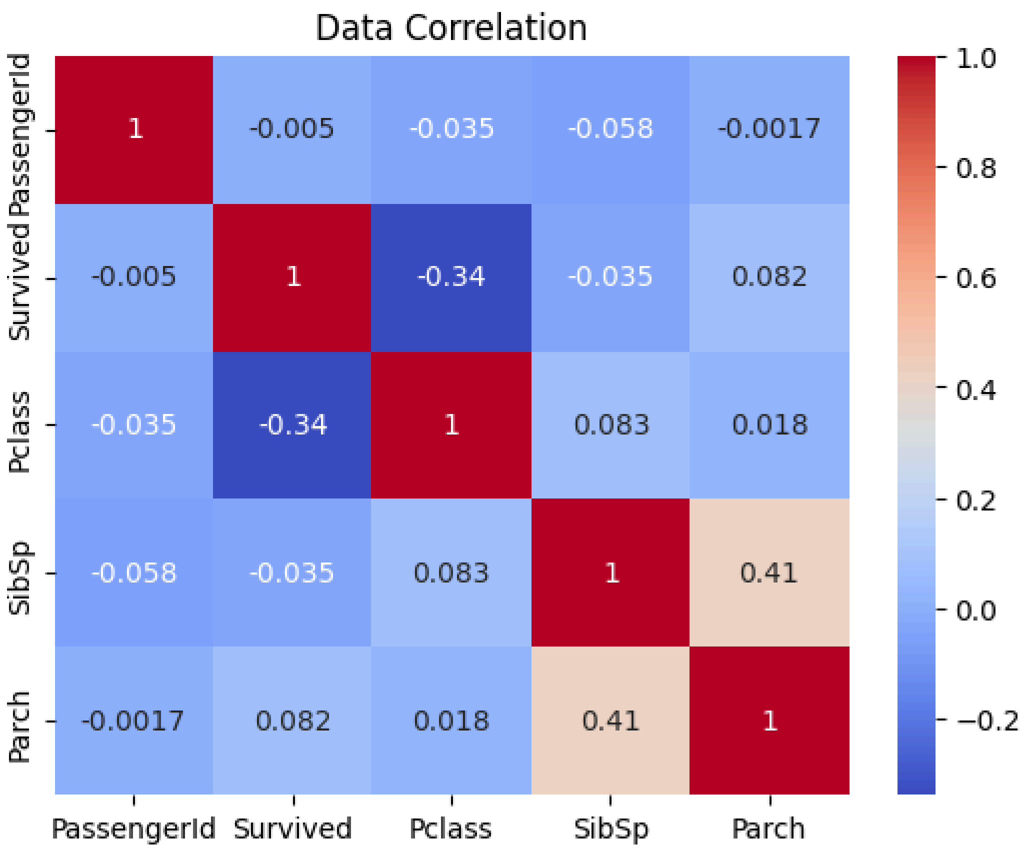
```
df=pd.read_csv("/content/drive/MyDrive/AI Tools Lab/new_boo.csv")
print(df)
categorical_cols = df.select_dtypes(include=['int64'])
categorical_cols.corr()
```

```
  name  rollno Gender  order
0   AA      11      M      1
1   BB      22      M      2
2   CC      33      F      3
3   DD      44      F      4
```

	rollno	order
rollno	1.0	1.0
order	1.0	1.0

Perform correlation on the data related to Titanic Data set

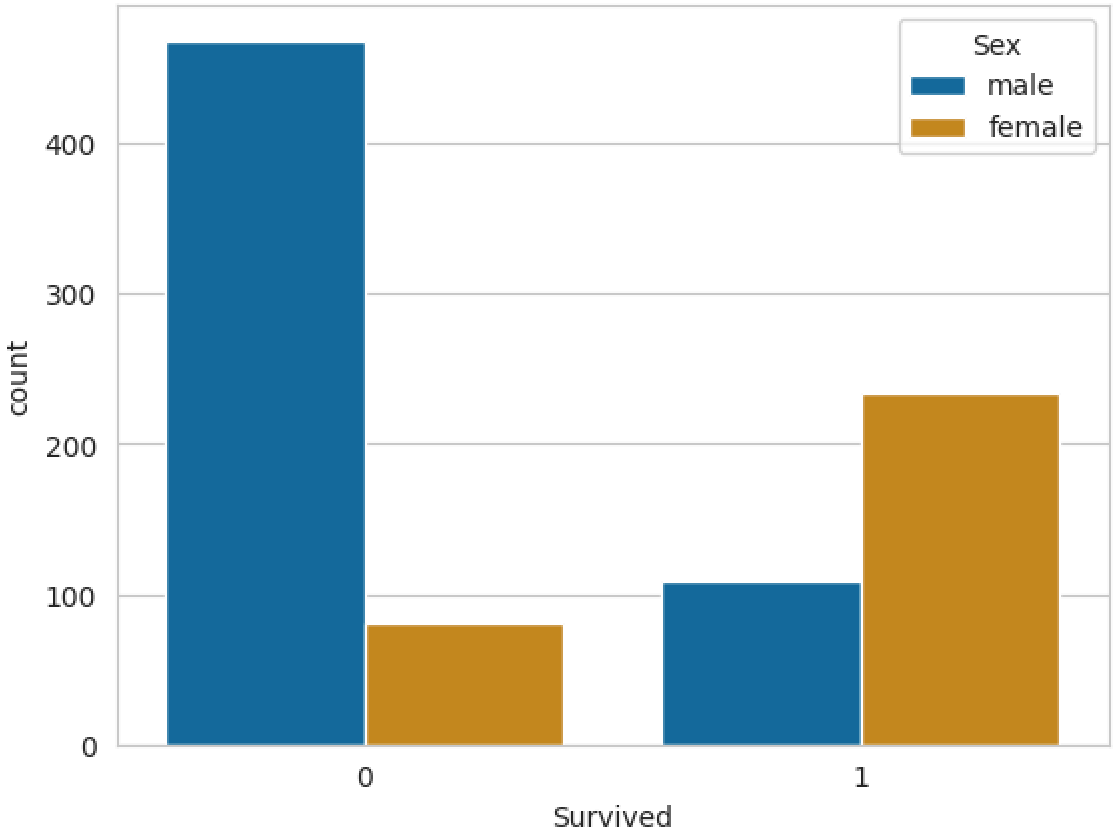
```
#importing all the necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
#reading data
data = pd.read_csv("/content/drive/MyDrive/AI Tools Lab/train.csv")
categorical_cols = data.select_dtypes(include=['int64'])
sns.heatmap(categorical_cols.corr(),cmap='coolwarm',xticklabels=True,annot=True)
plt.title('Data Correlation')
plt.show()
```



Number of survivals in each gender

```
# plotting countplot for Each gender who has survived and not survived
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Sex',data=data,palette='colorblind')
```

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



Number of survivals in each passenger class

```
#plotting count plot for no of survivals in each class
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=data,palette='bright')
```

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

Double-click (or enter) to edit

e. The number of people who are not alone.

```
# count plot for who has siblings/spouse
sns.countplot(x = 'SibSp', data = data,palette="bright",hue='SibSp')
plt.show()
```