Performing data cleaning and Analysis

1. Understanding meaning of each column:

Data Dictionary: Variable Description

Survived - Survived (1) or died (0) Pclass - Passenger's class (1 = 1st, 2 = 2nd, 3 = 3rd) Name - Passenger's name Sex - Passenger's sex Age - Passenger's age SibSp - Number of siblings/spouses aboard Parch - Number of parents/children aboard (Some children travelled only with a nanny, therefore parch=0 for them.) Ticket - Ticket number Fare - Fare Cabin - Cabin Embarked - Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

2. Analysing which columns are completely useless in predicting the survival and deleting them

Note - Don't just delete the columns because you are not finding it useful. Or focus is not on deleting the columns. Our focus is on analysing how each column is affecting the result or the prediction and in accordance with that deciding whether to keep the column or to delete the column or fill the null values of the column by some values and if yes, then what values.

In [1]:
```
# import libraries

import numpy as np
import pandas as pd
```

In [5]:
```
titanic = pd.read_csv(r'E:\One_Drive(Microsoft)\OneDrive\Data_Science_cource\Module_1_Python_29_July\D36_18-19Sep_([M
```

In [8]:
```
titanic.tail()
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [10]:
```
titanic.head()
```

Out[10]:

	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	NaN	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	NaN	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	NaN	S

In [26]:
```
# descrbe

titanic.describe().T
```

Out[26]:

	count	mean	std	min	25%	50%	75%	max
PassengerId	891.0	446.000000	257.353842	1.00	223.5000	446.0000	668.5	891.0000
Survived	891.0	0.383838	0.486592	0.00	0.0000	0.0000	1.0	1.0000
Pclass	891.0	2.308642	0.836071	1.00	2.0000	3.0000	3.0	3.0000
Age	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
SibSp	891.0	0.523008	1.102743	0.00	0.0000	0.0000	1.0	8.0000
Parch	891.0	0.381594	0.806057	0.00	0.0000	0.0000	0.0	6.0000
Fare	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

In [32]:
```
titanic.describe()
```

Out[32]:

	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 [34]:
```
titanic.columns
```

Out[34]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [36]:
```
# The person's name in the dataset can not help to identify wether the person survived or not so we will safely delet

del titanic['Name']
titanic.head()

# Name variable deleted successfully from datset
```

Out[36]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	male	35.0	0	0	373450	8.0500	NaN	S

In [38]:
```
# Ticket also not useful for us
del titanic['Ticket']
titanic.head()
```

Out[38]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	female	38.0	1	0	71.2833	C85	C
2	3	1	3	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	female	35.0	1	0	53.1000	C123	S
4	5	0	3	male	35.0	0	0	8.0500	NaN	S

In [40]:
```
# Also Fare and Cabin
del titanic['Fare']
del titanic['Cabin']

titanic.head()
```

Out[40]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked
0	1	0	3	male	22.0	1	0	S
1	2	1	1	female	38.0	1	0	C
2	3	1	3	female	26.0	0	0	S
3	4	1	1	female	35.0	1	0	S
4	5	0	3	male	35.0	0	0	S

In [58]:
```
# Here we will change the categorical or String values 'Male' and 'Female' as 1 and 2 respectively


def getNumber(str):
    if str=="male":
        return 1
    else:
        return 2
titanic["Gender"]=titanic["Sex"].apply(getNumber)

# Here we created one column name Gender based on Sex columns

titanic.head()
```

Out[58]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked	Gender
0	1	0	3	male	22.0	1	0	S	1
1	2	1	1	female	38.0	1	0	C	2
2	3	1	3	female	26.0	0	0	S	2
3	4	1	1	female	35.0	1	0	S	2
4	5	0	3	male	35.0	0	0	S	1

In [60]:
```
del titanic['Sex']
titanic.head()
```

Out[60]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Embarked	Gender
0	1	0	3	22.0	1	0	S	1
1	2	1	1	38.0	1	0	C	2
2	3	1	3	26.0	0	0	S	2
3	4	1	1	35.0	1	0	S	2
4	5	0	3	35.0	0	0	S	1

In [66]:
```
# titanic.isnull()
titanic.isna().sum()
```

Out[66]:
```
PassengerId      0
Survived         0
Pclass           0
Age            177
SibSp            0
Parch            0
Embarked         2
Gender           0
dtype: int64
```

Fill the null values of the Age column. Fill mean Survived age(mean age of the survived people) in the column where the person has survived and mean not Survived age (mean age of the people who have not survived) in the column where person has not survived###

In [69]:
```
# Fill the missing values of survived of Age

meanS = titanic[titanic.Survived == 1].Age.mean() ###
meanS
```

Out[69]:
```
28.343689655172415
```

Creating a new "Age" column , filling values in it with a condition if goes True then given values (here meanS) is put in place of last values else nothing happens, simply the values

are copied from the "Age" column of the dataset###

In [84]:
```
titanic['age'] = np.where(pd.isnull(titanic.Age) & titanic['Survived'] == 1, meanS, titanic['Age'])
titanic.head()
```

Out[84]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Embarked	Gender	age
0	1	0	3	22.0	1	0	S	1	22.0
1	2	1	1	38.0	1	0	C	2	38.0
2	3	1	3	26.0	0	0	S	2	26.0
3	4	1	1	35.0	1	0	S	2	35.0
4	5	0	3	35.0	0	0	S	1	35.0

In [86]:
```
titanic.isnull().sum()
```

Out[86]:
```
PassengerId    0
Survived       0
Pclass         0
Age            0
SibSp          0
Parch          0
Embarked       2
Gender         0
age            0
dtype: int64
```

In [88]:
```
#  Finding the mean of not survived people

meanNS = titanic[titanic.Survived == 0].Age.mean()
meanNS
```

Out[88]:
```
30.626179245283016
```

In [90]:
```
#  Filling missing not survived values with age mean

titanic.Age.fillna(meanNS, inplace= True)
titanic.head()
```

Out[90]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Embarked	Gender	age
0	1	0	3	22.0	1	0	S	1	22.0
1	2	1	1	38.0	1	0	C	2	38.0
2	3	1	3	26.0	0	0	S	2	26.0
3	4	1	1	35.0	1	0	S	2	35.0
4	5	0	3	35.0	0	0	S	1	35.0

In [92]:
```
titanic.isna().sum()
```

Out[92]:
```
PassengerId    0
Survived       0
Pclass         0
Age            0
SibSp          0
Parch          0
Embarked       2
Gender         0
age            0
dtype: int64
```

In [ ]:
```
del titanic['Age']
titanic.head()
```

In [101…
```
import warnings
warnings.filterwarnings('ignore')
```

We want to check if "Embarked" column is is important for analysis or not, that is whether survival of the person depends on the Embarked column value or not###

In [111…
```
# Finding the number of people who survived from different ports , only who are survived
# Q,C,S
survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 1].shape[0]
survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 1].shape[0]
survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 1].shape[0]
```

```
print(survivedQ)
print(survivedC)
print(survivedS)
```

```
30
93
217
```

In [127…

```
#  Finding the people who are not survived by considering ports QCS
# this can don by applying two condition on dataset

# survivedQ0 = titanic[titanic.Embarked == 'Q'].shape[0] # Survived and not of Q => 77
survivedQ0 = titanic[titanic.Embarked == 'Q'][titanic.Survived == 0].shape[0] # not survived of Q 47

# survivedC0 = titanic[titanic.Embarked == 'C'].shape[0] # survied and not of 'C'  => 168
survivedC0 = titanic[titanic.Embarked == 'C'][titanic.SibSp == 0].shape[0] # not survived of 'C' => 109

# survivedS0 = titanic[titanic.Embarked == 'S'].shape[0] # total survived and not -> 644
survivedS0 = titanic[titanic.Embarked == 'S'][titanic.Survived == 0].shape[0]


# diplaying
print(survivedQ0)
print(survivedC0)
print(survivedS0)
```

```
47
109
427
```

As there are significant changes in the survival rate based on which port the passengers aboard the ship. We cannot delete the whole embarked column(It is useful). Now the Embarked column has some null values in it and hence we can safely say that deleting some rows from total rows will not affect the result. So rather than trying to fill those null values with some vales. We can simply remove them.

In [130…

```
# dropping the missing null values

titanic.dropna(inplace= True)
titanic.head()
```

Out[130…

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Gender	age
0	1	0	3	1	0	S	1	22.0
1	2	1	1	1	0	C	2	38.0
2	3	1	3	0	0	S	2	26.0
3	4	1	1	1	0	S	2	35.0
4	5	0	3	0	0	S	1	35.0

In [132…
```
titanic.isnull().sum()
```

Out[132…
```
PassengerId    0
Survived       0
Pclass         0
SibSp          0
Parch          0
Embarked       0
Gender         0
age            0
dtype: int64
```

Renaming 'age' and 'Gendr' columns as 'age' => 'Age' and 'Gender' => 'Sex'

In [136…
```
# We can rename the column by rename function
titanic.rename(columns={'Gender':'Sex','age':'Age'}, inplace= True)
titanic.head()
```

Out[136…

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Sex	Age
0	1	0	3	1	0	S	1	22.0
1	2	1	1	1	0	C	2	38.0
2	3	1	3	0	0	S	2	26.0
3	4	1	1	1	0	S	2	35.0
4	5	0	3	0	0	S	1	35.0

In [138…
```
#  In Embarked categorical data is there we need to convert it into numeric

def getEmb(str):
    if str == 'Q':
        return 1
    elif str == 'C':
        return 2
    else:
        return 3
titanic['temp-embark'] = titanic.Embarked.apply(getEmb)
```

In [140…
```
titanic
```

Out[140…

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Sex	Age	temp-embark
0	1	0	3	1	0	S	1	22.000000	3
1	2	1	1	1	0	C	2	38.000000	2
2	3	1	3	0	0	S	2	26.000000	3
3	4	1	1	1	0	S	2	35.000000	3
4	5	0	3	0	0	S	1	35.000000	3
...	...	...	...	...	...	...	...	...	...
886	887	0	2	0	0	S	1	27.000000	3
887	888	1	1	0	0	S	2	19.000000	3
888	889	0	3	1	2	S	2	30.626179	3
889	890	1	1	0	0	C	1	26.000000	2
890	891	0	3	0	0	Q	1	32.000000	1

889 rows × 9 columns

In [142…
```
# Embarked
del titanic['Embarked']
titanic.rename(columns={'temp-embark':'Embarked'}, inplace= True)
```

```
titanic.head()
```

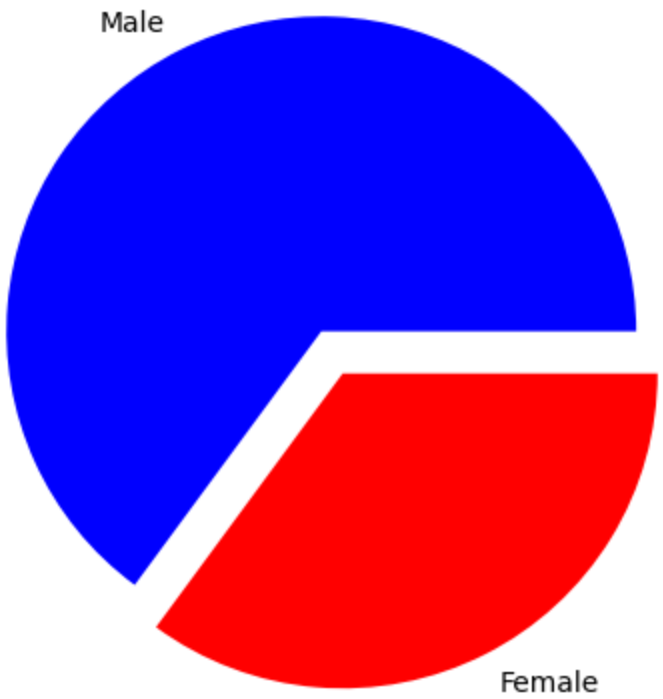
Out[142…

	PassengerId	Survived	Pclass	SibSp	Parch	Sex	Age	Embarked
0	1	0	3	1	0	1	22.0	3
1	2	1	1	1	0	2	38.0	2
2	3	1	3	0	0	2	26.0	3
3	4	1	1	1	0	2	35.0	3
4	5	0	3	0	0	1	35.0	3

In [184…

```
# Creating piechart for Sex data 'Male' and 'Female, 1, 2

import matplotlib.pyplot as plt
from matplotlib import style

# printing total number of male 1 and females 2

males = (titanic['Sex'] == 1).sum()
females = (titanic['Sex'] == 2).sum()

print("No of Males in ship : ",males)
print("No of Females in ship : ",females)

# creating the list of males and females variables
p = [males, females]

# provied the data to pie
plt.pie(p
       ,labels=['Male','Female'] # Adding labels
       ,colors=['blue','red']
        , explode = (0.15,0),
        startangle= 0 )

plt.axis('equal')
plt.show()
```

```
No of Males in ship :  577
No of Females in ship :  312
```



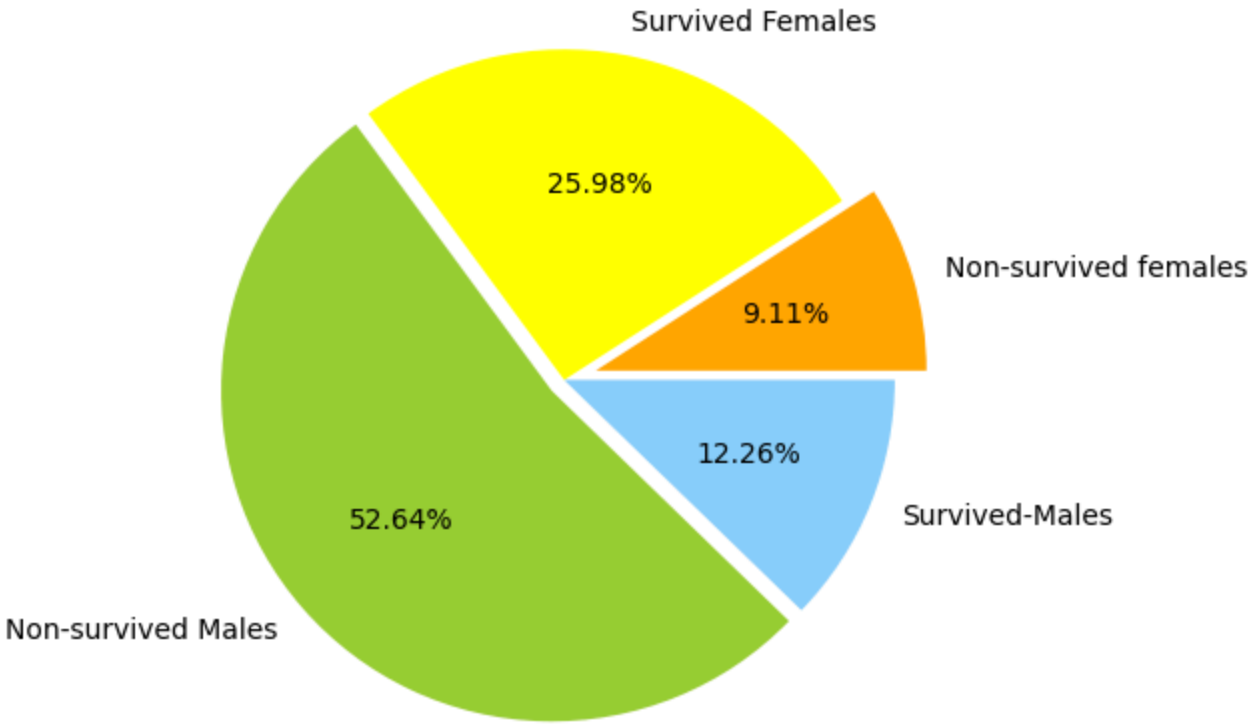
In [192…

```
#  More precise piechart where add survived & non survived males-females

# Survived male and non-survived males
maleS = titanic[titanic.Sex == 1][titanic.Survived == 1].shape[0] # survived males
maleN  = titanic[titanic.Sex == 1][titanic.Survived == 0].shape[0] # non survived males

# Survived females and non-survived females
femaleS = titanic[titanic.Sex == 2][titanic.Survived == 1].shape[0] # survived females
femaleN = titanic[titanic.Sex == 2][titanic.Survived == 0].shape[0] # non-survived females

# pritting data
print("Numbers of survived males  : ",maleS)
print("Numbers of non-survived males : ",maleN)

print("Number of survived females : ",femaleS)
print("Number of non-survived females : ",femaleN)
```

```
Numbers of survived males  :  109
Numbers of non-survived males :  468
Number of survived females :  231
Number of non-survived females :  81
```

In [202…

```
p1 = [maleS,maleN, femaleS, femaleN]
labels=['Survived-Males', 'Non-survived Males', 'Survived Females', 'Non-survived females']
explode=[0,0.05,0,0.1]
colors=['lightskyblue','yellowgreen','Yellow','Orange']
plt.pie(p1, labels = labels, colors= colors, explode = explode ,counterclock=False,autopct="%.2f%%")

plt.axis('equal')
plt.show()
```


Survived Females
25.98%
Non-survived females
9.11%
12.26%
Survived-Males
52.64%
Non-survived Males

In [ ]: