```
# Importing Necessary Libraries

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

```
# 1. Read the dataset

dataset = pd.read_csv("/content/titanic.csv")
dataset.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2

Next steps: Generate code with dataset View recommended plots

```
# 2. Observe the shape of dataset

dataset.shape
```

```
(891, 12)
```

```
# 3. Observe the statistics of the dataset

dataset.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329

```
# 4.Observe the number of Non-NULL and datatype of each feature of the dataset

dataset.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
```

```
# 5. Bifurcate the categorical and numerical features of the dataset

# Separate the features into categorical and numerical

categorical_features = [column for column in dataset.columns if dataset[column].dtype == 'object']
numerical_features = [column for column in dataset.columns if dataset[column].dtype == 'int64' or dataset[column].dtype == 'float64

# Print the results

print("Categorical features:", categorical_features)
print("Numerical features:", numerical_features)
```

```
Categorical features: ['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
Numerical features: ['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
```

```
# 6. Observe the number of null (N/A) values for each feature

dataset.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
```

```
# 7. Observe the percentage of null (N/A) values for each feature

null_percentage = round(dataset.isnull().sum() * (100 / dataset.shape[0]),2)
null_percentage
```

```
PassengerId     0.00
Survived        0.00
Pclass          0.00
Name            0.00
Sex             0.00
Age            19.87
SibSp           0.00
Parch           0.00
Ticket          0.00
Fare            0.00
Cabin          77.10
Embarked        0.22
dtype: float64
```

```
# 8. Drop the "Ticket" and "Name" features from the dataset

dataset = dataset.drop(['Ticket', 'Name'], axis=1)
dataset.columns
```

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

```
# 9. Drop the feature corresponding to the highest missing values

# Identify the feature with the highest percentage of missing values

max_null_feature = null_percentage.idxmax()
max_null_percentage = null_percentage.max()

print(f"The feature with the highest percentage of missing values is '{max_null_feature}' with {max_null_percentage:.2f}% missing v

# Drop the feature with the highest percentage of missing values

dataset = dataset.drop(max_null_feature, axis=1)
dataset.columns
```

```
The feature with the highest percentage of missing values is 'Cabin' with 77.10% missing values.
Index(['PassengerId', 'Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch',
       'Fare', 'Embarked'],
      dtype='object')
```

```
dataset.columns
```

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

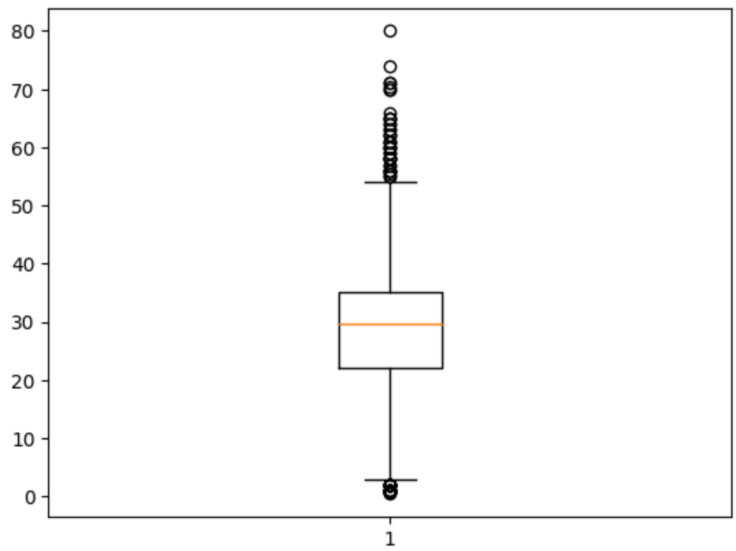
```
# 10. Drop the observations with missing values in the "Embarked" feature

dataset = dataset.dropna(subset=['Embarked'] , axis=0)

# Checking the NULL Values after dropping the Observations

dataset.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age            177
SibSp            0
Parch            0
Fare             0
Embarked         0
dtype: int64
```

```
# 11. Fill the missing values of the "Age" feature with mean value

dataset['Age'] = dataset['Age'].fillna(dataset['Age'].mean())

# Checking the NULL Values after dropping the Observations

dataset.isnull().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       0
dtype: int64
```

```
# 12. Observe the boxplot of the "Age" feature

plt.boxplot(dataset['Age'])
plt.show()
```




```
# 13. Nomalize the features with the numerical values using MinMaxScaler

feature_to_be_scaled = numerical_features

# Importing the Scaler
```

```
from sklearn.preprocessing import MinMaxScaler

# Creating the Scaler

scaler = MinMaxScaler()

# Scaling the features

dataset[feature_to_be_scaled] = scaler.fit_transform(dataset[feature_to_be_scaled])

# Showing the Result

dataset.head()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0.000000	0.0	1.0	male	0.271174	0.125	0.0	0.014151	S
1	0.001124	1.0	0.0	female	0.472229	0.125	0.0	0.139136	C
2	0.002247	1.0	1.0	female	0.321438	0.000	0.0	0.015469	S
3	0.003371	1.0	0.0	female	0.434531	0.125	0.0	0.103644	S
4	0.004494	0.0	1.0	male	0.434531	0.000	0.0	0.015713	S

Next steps: Generate code with `dataset` View recommended plots

```
# Visulizing the Normalized Data

dataset.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000
mean	0.500000	0.382452	0.655793	0.367204	0.065523	0.063742	0.062649
std	0.288762	0.486260	0.417350	0.162960	0.137963	0.134460	0.097003
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.250562	0.000000	0.500000	0.271174	0.000000	0.000000	0.015412
50%	0.500000	0.000000	1.000000	0.367204	0.000000	0.000000	0.028213
75%	0.749438	1.000000	1.000000	0.434531	0.125000	0.000000	0.060508
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000