

TASK 2: SUMMARY STATISTICS

The `describe()` function is used to obtain summary statistics: mean, count, standard deviation, percentiles and the minimum and maximum values in each of the columns containing numerical values. Before calculating summary statistics, the data is cleaned.

```
1  # Import libraries
2  import pandas as pd
3
4  # Load the Titanic dataset
5  TITANIC = pd.read_csv('train.csv')
6
7  # Checking for missing values
8  print(TITANIC.isnull().sum())
9
10 # Dropping columns that are unlikely to be useful in analysis
11 TITANIC.drop(columns=["Cabin", "Name", "Ticket"], axis=1, inplace=True)
12
13 # Filling missing values
14 TITANIC["Embarked"] = TITANIC["Embarked"].fillna(TITANIC["Embarked"].mode()[0]) # Mode for 'Embarked'
15 TITANIC['Age'].fillna(TITANIC["Age"].median(), inplace=True) # Median for 'Age'
16
17 # Confirming that there are no null values
18 print(TITANIC.isnull().sum())
19
20 # Detecting outliers using IQR (Interquartile Range) for Age
21 Q1_age = TITANIC['Age'].quantile(0.25)
22 Q3_age = TITANIC['Age'].quantile(0.75)
23 IQR_age = Q3_age - Q1_age
24 lower_bound_age = Q1_age - 1.5 * IQR_age
25 upper_bound_age = Q3_age + 1.5 * IQR_age
26
27 # Identifying outliers in Age
28 outliers_age = TITANIC[(TITANIC['Age'] < lower_bound_age) | (TITANIC['Age'] > upper_bound_age)]
29 print(f'Outliers in Age:\n{outliers_age}')
30
31 # Detecting outliers using IQR for Fare
32 Q1_fare = TITANIC['Fare'].quantile(0.25)
33 Q3_fare = TITANIC['Fare'].quantile(0.75)
34 IQR_fare = Q3_fare - Q1_fare
35 lower_bound_fare = Q1_fare - 1.5 * IQR_fare
36 upper_bound_fare = Q3_fare + 1.5 * IQR_fare
37
38 # Identifying outliers in Fare
39 outliers_fare = TITANIC[(TITANIC['Fare'] < lower_bound_fare) | (TITANIC['Fare'] > upper_bound_fare)]
40 print(f'Outliers in Fare:\n{outliers_fare}')
41
42 # Summary statistics after handling missing values
43 print(TITANIC.describe())
44
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.361582	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.019697	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	22.000000	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	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200