

Task 2. Calculate summary statistics

```
In [29]: import pandas as pd
import statistics
import numpy as np
import seaborn as sns

In [32]: df=pd.read_csv("Techno Task 2 data.csv")

In [25]: df.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	NaN	S
1	2	1	1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8	NaN	S

```
In [14]: df.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                  294 non-null    object  
11  Embarked               889 non-null    object  
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

In [15]: df['Fare'] = df['Fare'].astype('int')

In [17]: df['Age'] = df['Age'].astype(str).str.lstrip('<').str.rstrip('>')

In [22]: df.describe()
```

	PassengerId	Survived	Pclass	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	0.523008	0.381594	31.785634
std	257.353842	0.486592	0.836071	1.102743	0.806057	49.703730
min	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	0.000000	0.000000	7.000000
50%	446.000000	0.000000	3.000000	0.000000	0.000000	14.000000
75%	668.500000	1.000000	3.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	8.000000	6.000000	512.000000

```
In [21]: df.describe(include = "all")

Out[21]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	891	891.000000	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	89	NaN	NaN	681	NaN	147	3
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	nan	NaN	NaN	347082	NaN	B96 B98	S
freq	NaN	NaN	NaN	1	577	177	NaN	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	NaN	0.523008	0.381594	NaN	31.785634	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	NaN	1.102743	0.806057	NaN	49.703730	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	NaN	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	NaN	0.000000	0.000000	NaN	7.000000	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	NaN	0.000000	0.000000	NaN	14.000000	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	NaN	1.000000	0.000000	NaN	31.000000	NaN	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	NaN	8.000000	6.000000	NaN	512.000000	NaN	NaN

```
In [26]: df.describe(include=['object'])

Out[26]:
```

	Name	Sex	Age	Ticket	Cabin	Embarked
count	891	891	891	891	204	889
unique	891	2	89	681	147	3
top	Braund, Mr. Owen Harris	male	nan	347082	B96 B98	S
freq	1	577	177	7	4	644

Calculation for PassengersId

```
In [34]: Passenger_data = df['PassengerId']
type(Passenger_data)
```

Out[34]: pandas.core.series.Series

MEAN

```
In [35]: statistics.mean(Passenger_data)
```

Out[35]: 446

MODE

```
In [38]: statistics.mode(Passenger_data)
```

Out[38]: 1

MEDIAN

```
In [39]: statistics.median(Passenger_data)
```

Out[39]: 446

STANDARD DEVIATION

```
In [42]: statistics.stdev(Passenger_data)
```

Out[42]: 257.3538420152301

Calculation of survived passengers

```
In [89]: Survived_passenger = df['Survived']
type(Survived_passenger)
```

Out[89]: pandas.core.series.Series

MEAN

```
In [90]: statistics.mean(Survived_passenger)
```

Out[90]: 0.3838383838383838

MEDIAN

```
In [91]: statistics.median(Survived_passenger)
```

Out[91]: 0

MODE

```
In [92]: statistics.mode(Survived_passenger)
```

Out[92]: 0

Standard Deviation

```
In [93]: statistics.stdev(Survived_passenger)
```

Out[93]: 0.48659245426485753

Calculation of sibling & Spouse

```
In [70]: SiblingSpouse_data = df['SibSp']
type(SiblingSpouse_data)
```

Out[70]: pandas.core.series.Series

MEAN

```
In [71]: statistics.mean(SiblingSpouse_data)
```

Out[71]: 0.5230078563411896

MEDIAN

```
In [72]: statistics.median(SiblingSpouse_data)
```

Out[72]: 0

MODE

```
In [73]: statistics.mode(SiblingSpouse_data)
```

Out[73]: 0

STANDARD DEVIATION

```
In [74]: statistics.stdev(SiblingSpouse_data)
```

Out[74]: 1.1027434322934317

Calculation of Parents & Children

```
In [76]: Parentschildren_data = df['Parch']
type(Parentschildren_data)
```

Out[76]: pandas.core.series.Series

MEAN

```
In [78]: statistics.mean(Parentschildren_data)
```

Out[78]: 0.38159371492704824

MEDIAN

```
In [79]: statistics.median(Parentschildren_data)
```

Out[79]: 0

MODE

```
In [80]: statistics.mode(Parentschildren_data)
```

Out[80]: 0

STANDARD DEVIATION

```
In [81]: statistics.stdev(Parentschildren_data)
```

Out[81]: 0.8060572211299483

Calculation of Fare

```
In [82]: Fare_data = df['Fare']
type(Fare_data)
```

Out[82]: pandas.core.series.Series

MEAN

```
In [84]: statistics.mean(Fare_data)
```

Out[84]: 32.204207968574636

MEDIAN

```
In [85]: statistics.median(Fare_data)
```

Out[85]: 14.4542

MODE

```
In [86]: statistics.mode(Fare_data)
```

Out[86]: 8.05

Standard deviation

```
In [87]: statistics.stdev(Fare_data)
```

Out[87]: 49.6934285971809

Calculated mean, median, mode, and standard deviation of passenger id, survived passenger, Sibling/spouse, parents/children, and Fare.

