OUTPUTS FOR TASK_5

OUTPUTS FOR .SHAPE()

```
Shape: (891, 12)
```

	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

OUTPUT FOR .DESCRIBE()

	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

OUTPUT FOR .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
```

OUTPUT FOR VALUE_COUNT()

```
PassengerId  Survived  Pclass  Name                                          Sex     Age   SibSp  Parch  Ticket    Fare     Cabin        Embarke
d
2            1         1       Cumings, Mrs. John Bradley (Florence Briggs Thayer)  female  38.0  1      0      PC 17599  71.2833  C85          C
1
4            1         1       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0  1      0      113803    53.1000  C123         S
1
7            0         1       McCarthy, Mr. Timothy J                       male    54.0  0      0      17463     51.8625  E46          S
1
11           1         3       Sandstrom, Miss. Marguerite Rut               female  4.0   1      1      PP 9549   16.7000  G6           S
1
12           1         1       Bonnell, Miss. Elizabeth                      female  58.0  0      0      113783    26.5500  C103         S
1

..
872          1         1       Beckwith, Mrs. Richard Leonard (Sallie Monypeny)  female  47.0  1      1      11751     52.5542  D35          S
1
873          0         1       Carlsson, Mr. Frans Olof                      male    33.0  0      0      695       5.0000   B51 B53 B55  S
1
880          1         1       Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  female  56.0  0      1      11767     83.1583  C50          C
1
888          1         1       Graham, Miss. Margaret Edith                  female  19.0  0      0      112053    30.0000  B42          S
1
890          1         1       Behr, Mr. Karl Howell                         male    26.0  0      0      111369    30.0000  C148         C
1
Name: count, Length: 183, dtype: int64
```

OUTPUT FOR MISSING VALUES

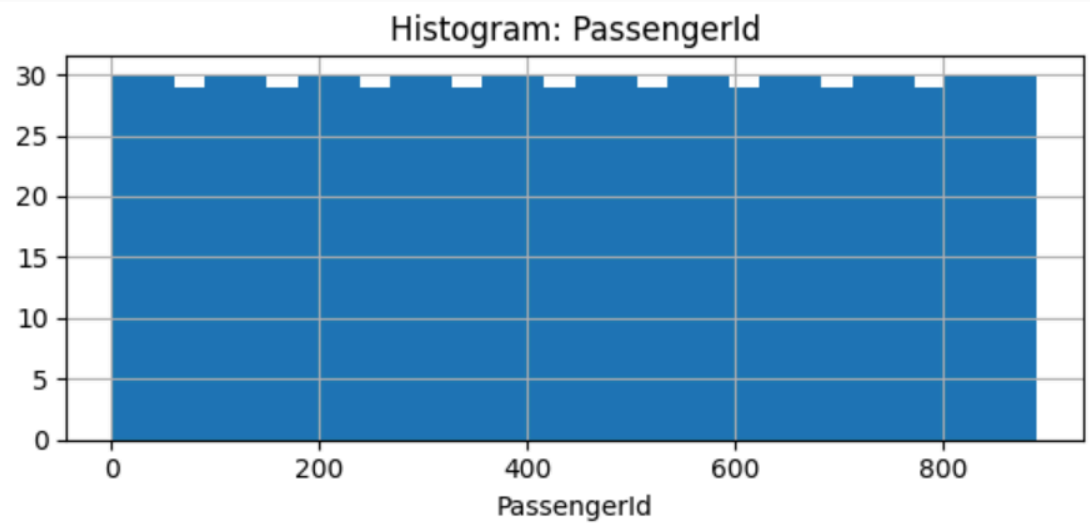
	missing_count	missing_pct
Cabin	687	77.104377
Age	177	19.865320
Embarked	2	0.224467
PassengerId	0	0.000000
Name	0	0.000000
Pclass	0	0.000000
Survived	0	0.000000
Sex	0	0.000000
Parch	0	0.000000
SibSp	0	0.000000
Fare	0	0.000000
Ticket	0	0.000000

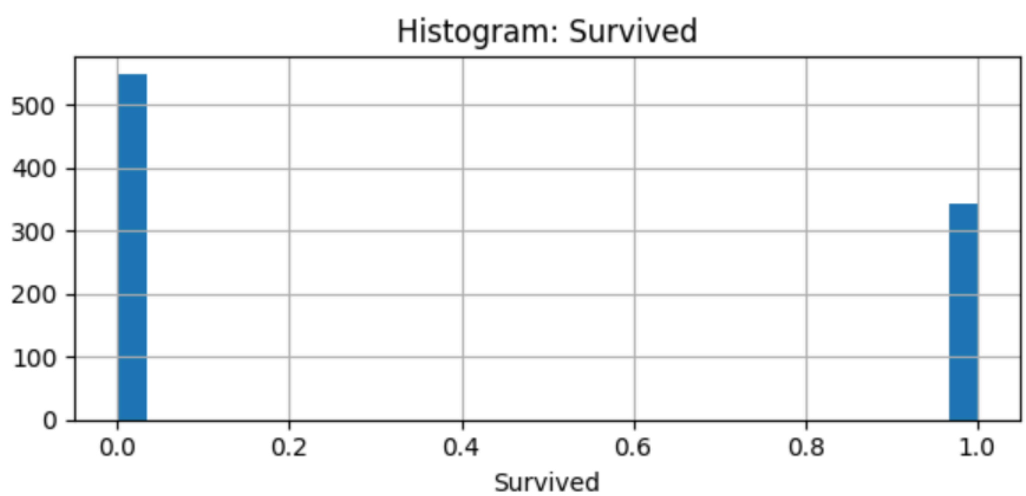
OUTPUT FOR CLEANING THE DATA

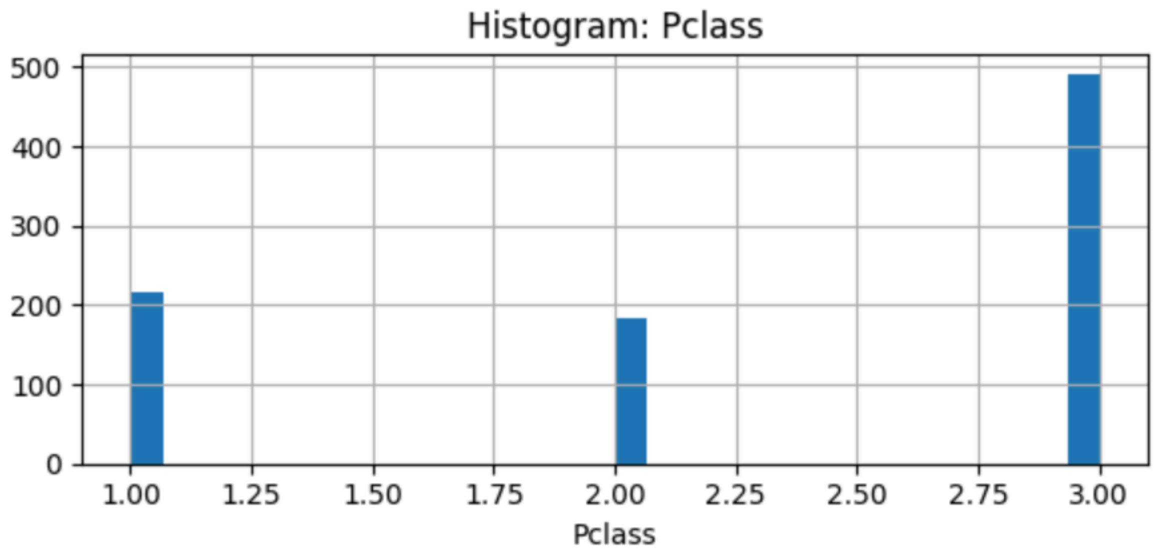
]:

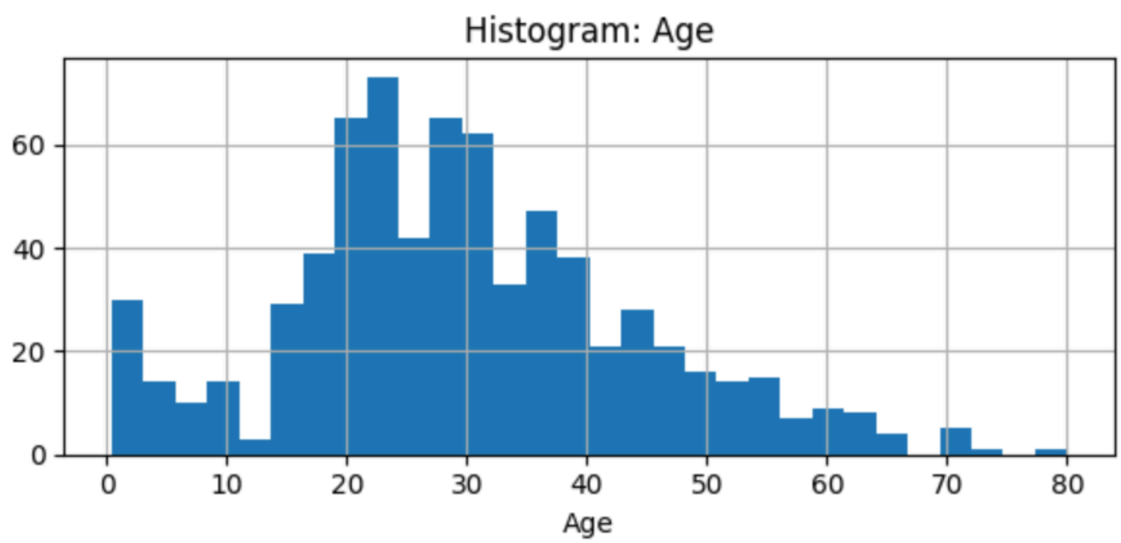
	Age	Age_median_fill	Cabin	HasCabin
0	22.0	22.0	NaN	0
1	38.0	38.0	C85	1
2	26.0	26.0	NaN	0
3	35.0	35.0	C123	1
4	35.0	35.0	NaN	0

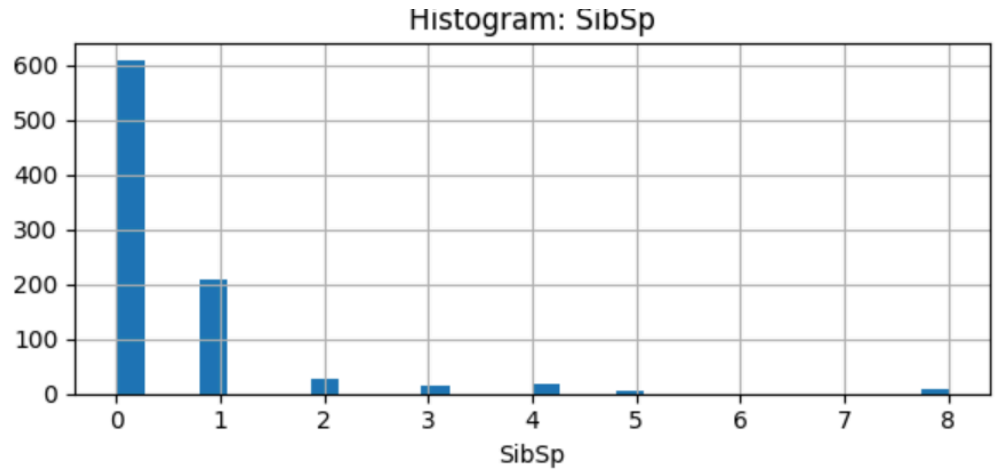
OUTPUTS FOR VISUALIZATION



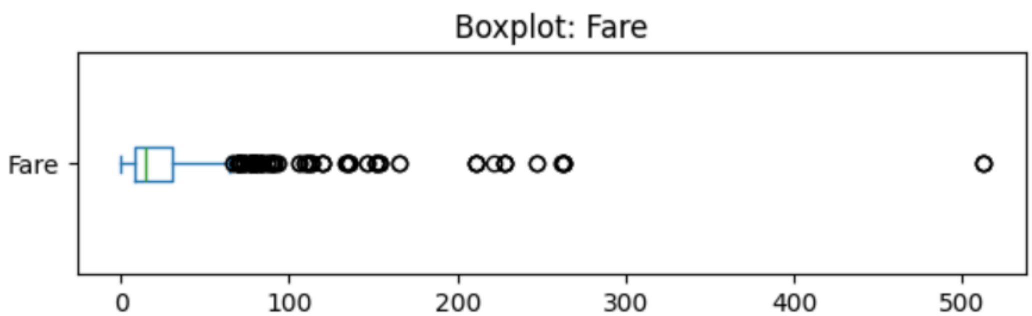



Histogram: Pclass
500
400
300
200
100
0
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Pclass

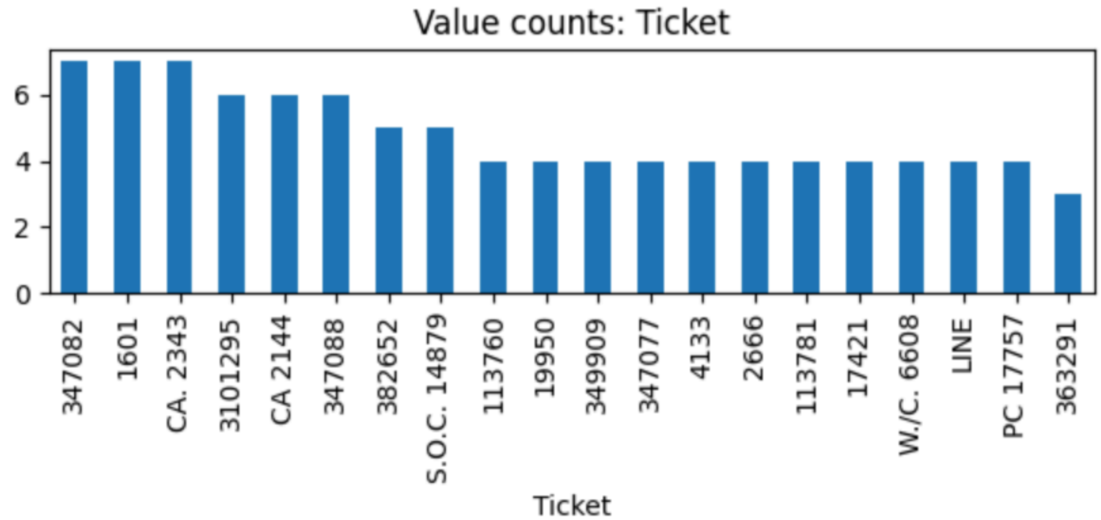

Histogram: Age
60
40
20
0
0
10
20
30
40
50
60
70
80
Age

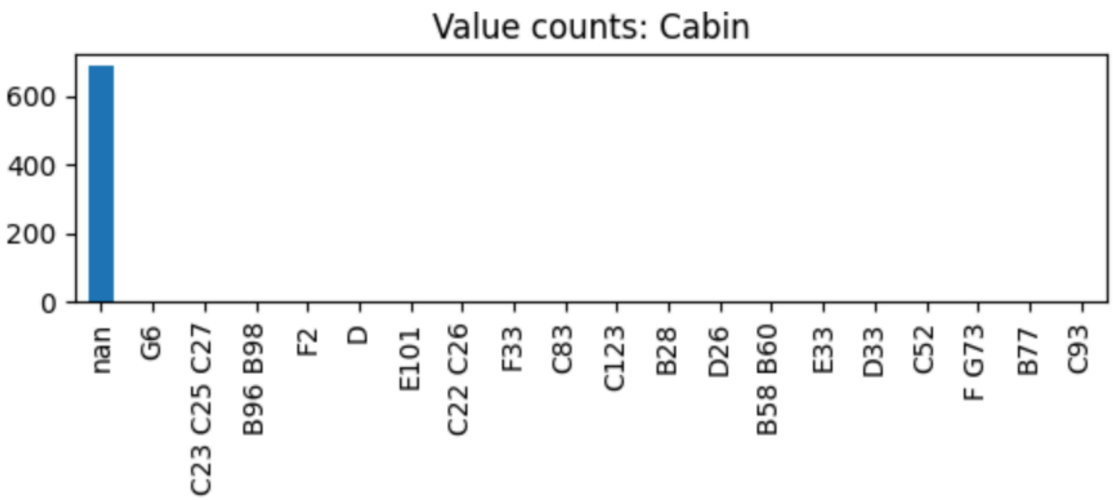


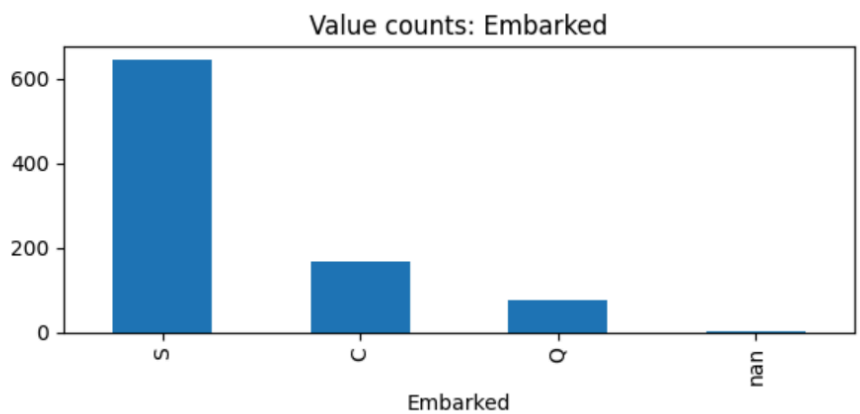
Histogram: SibSp
600
500
400
300
200
100
0
0
1
2
3
4
5
6
7
8
SibSp

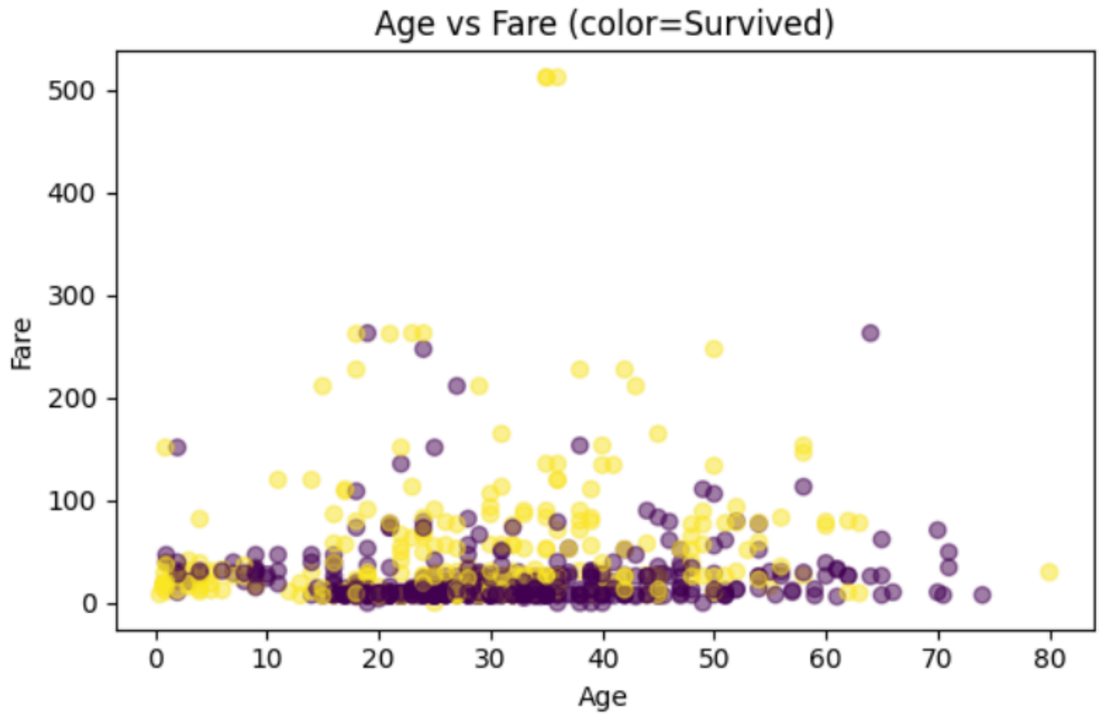


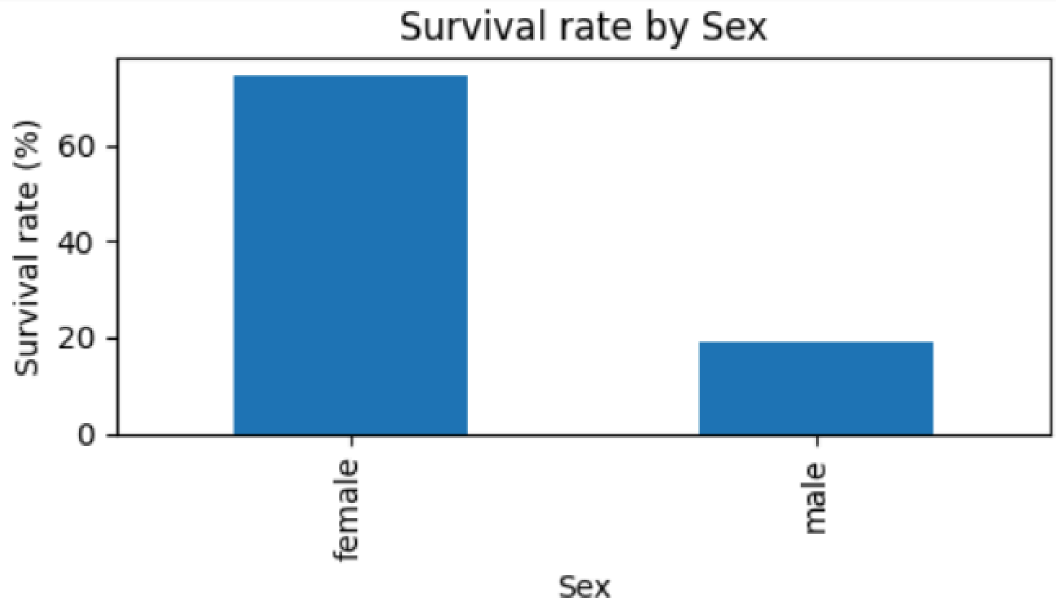
Boxplot: Fare
Fare
0
100
200
300
400
500

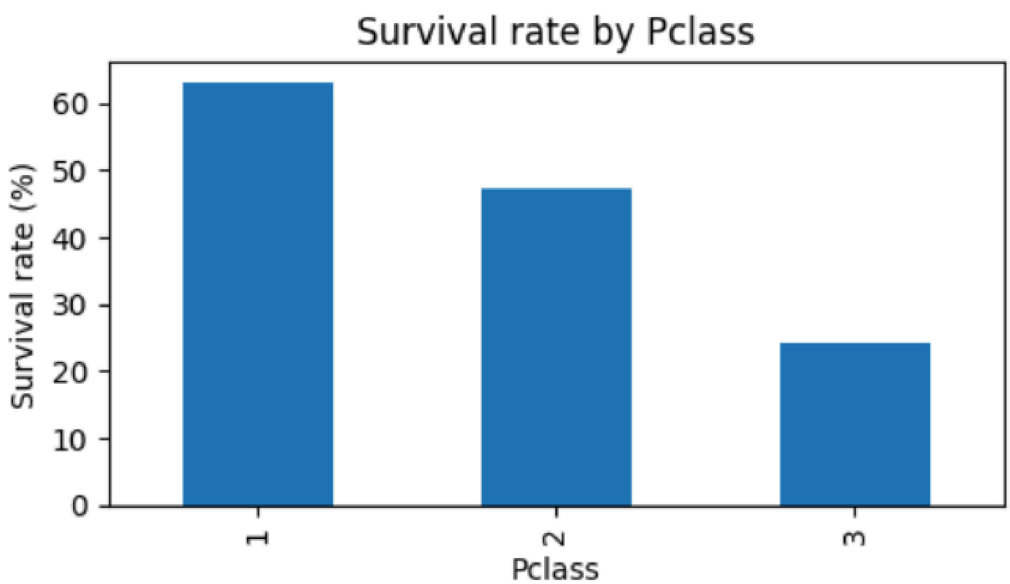

Value counts: Ticket
0
2
4
6
347082
1601
CA. 2343
3101295
CA 2144
347088
382652
S.O.C. 14879
113760
19950
349909
347077
4133
2666
113781
17421
W./C. 6608
LINE
PC 17757
363291
Ticket


Value counts: Cabin
0
200
400
600
nan
G6
C23 C25 C27
B96 B98
F2
D
E101
C22 C26
F33
C83
C123
B28
D26
B58 B60
E33
D33
C52
F G73
B77
C93


Value counts: Embarked
0
200
400
600
S
C
Q
nan
Embarked


Age vs Fare (color=Survived)
Fare
Age
0
100
200
300
400
500
0
10
20
30
40
50
60
70
80


Survival rate by Sex
Survival rate (%)
0
20
40
60
female
male
Sex


Survival rate by Pclass
Survival rate (%)
0
10
20
30
40
50
60
1
2
3
Pclass

CONCLUSION

To explore the dataset, I started by importing a few essential Python libraries — `pandas` to load and manipulate the data, `numpy` for quick numerical calculations, and `matplotlib.pyplot` to create visualizations. I loaded the dataset using `pd.read_csv('/mnt/data/train.csv')`, which stored everything in a DataFrame called `df`. To get a quick overview of what I was working with, I ran `df.info()`, which showed me how many rows and columns the dataset has, along with the data types of each column. This also helped me spot missing values and understand which features were numeric vs. categorical. I used `df.shape` to confirm the dataset size and `df.columns.tolist()` to list out all the column names. After that, I separated numeric and categorical columns using `select_dtypes`. To check data quality, I calculated how many values were missing in each column with `df.isnull().sum()` and also computed their percentages using `df.isnull().mean()*100`. This made it obvious that features like *Age* and *Cabin* had a lot of missing entries. Next, I ran `df[numeric_cols].describe()` to get summary statistics such as mean, median, standard deviation, and quartile ranges, which helped me understand how values were distributed. For visualization, I plotted histograms (`df[col].plot(kind='hist')`) to see how features like Age and Fare were spread out, and created boxplots (`df[col].plot(kind='box')`) to detect any extreme outliers. To check relationships between variables, I generated a correlation heatmap using `plt.imshow(df.corr())`, which showed patterns like higher fares being associated with passengers from better classes. I also created a scatter matrix to visually compare multiple numeric features against each other. Finally, to get real-world insights, I calculated survival rates across different groups using `value_counts(normalize=True)`, which clearly showed that women and first-class passengers had much higher chances of survival. All the plots were saved into a folder for easy access, and everything was organized into a Jupyter Notebook for reporting.