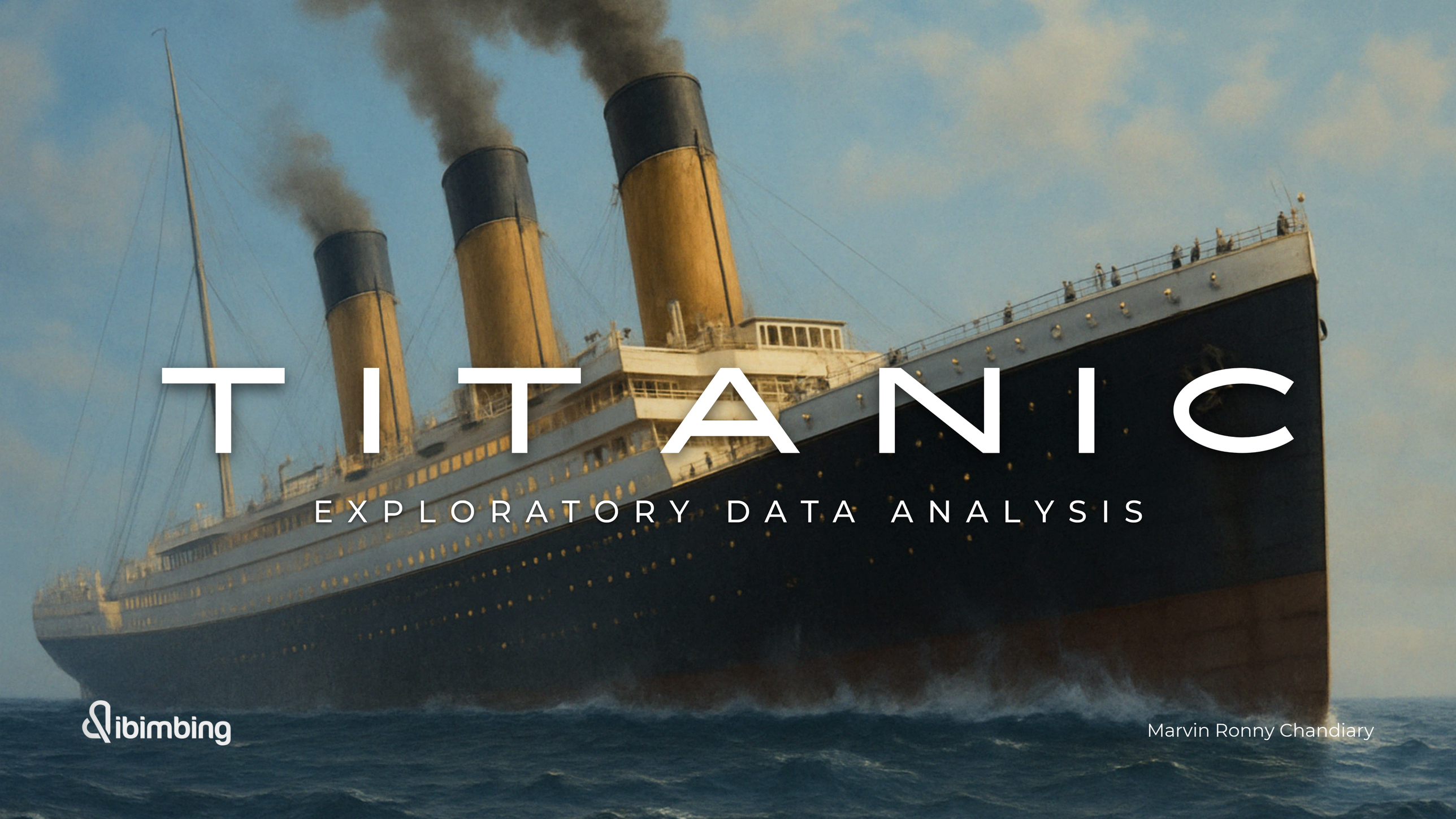


TITANIC

EXPLORATORY DATA ANALYSIS

ibimbing

Marvin Ronny Chandiary

Project Introduction

Objective

The main goal of this project is to learn and apply the core steps of data analysis, with a **focus on data cleaning** and **exploratory data analysis (EDA)**. It aims to build a strong foundation in handling real-world datasets.

Dataset

The dataset used is the classic Titanic dataset, which contains information about passengers on the Titanic, including features such as name, gender, age, and survival status.

Scope of Work

- Understand the dataset by checking samples and structure.
- Summarize key statistics to explore data patterns.
- Clean the data by handling duplicates and missing values.
- Create basic visualizations to support data understanding.

Tools

This project is completed using **Python**, including **pandas**, **matplotlib**, and **seaborn**.

01

Understanding the Dataset

Head & Tail

```
[3] data.head()
```

	survived	name	sex	age
0	1	Allen, Miss. Elisabeth Walton	female	29.0000
1	1	Allison, Master. Hudson Trevor	male	0.9167
2	0	Allison, Miss. Helen Loraine	female	2.0000
3	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000
4	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000

Showing top 5 rows of the data

```
[4] data.tail()
```

	survived	name	sex	age
495	1	Mallet, Mrs. Albert (Antoinette Magnin)	female	24.0
496	0	Mangiavacchi, Mr. Serafino Emilio	male	NaN
497	0	Matthews, Mr. William John	male	30.0
498	0	Maybery, Mr. Frank Hubert	male	40.0
499	0	McCrae, Mr. Arthur Gordon	male	32.0

Showing last 5 rows of the data

01

Understanding the Dataset

Random Sample

```
[5] data.sample(5)
```

	survived	name	sex	age
191	0	Loring, Mr. Joseph Holland	male	30.0
185	0	Lindeberg-Lind, Mr. Erik Gustaf ("Mr Edward Li...	male	42.0
454	0	Hood, Mr. Ambrose Jr	male	21.0
47	1	Calderhead, Mr. Edward Pennington	male	42.0
245	1	Rothes, the Countess. of (Lucy Noel Martha Dye...	female	33.0

Observations:

1. The dataset contains **numerical columns** (survived, age) and **categorical columns** (name, sex).
2. The ‘survived’ column appears to be binary (0 or 1).
3. The ‘sex’ column contains only two categories: 'male' and 'female'.
4. The ‘age’ column is in float format, likely representing exact years.

01

Understanding the Dataset

Info Data

```
[6]  data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   survived  500 non-null    int64
 1   name      500 non-null    object
 2   sex       500 non-null    object
 3   age       451 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 15.8+ KB
```

Observations:

1. The dataset has 4 columns and 500 rows.
2. Only the ‘age’ column contains missing values (49 rows).
3. The data types appear appropriate for each column.

02

Explore the Data

Classify Data Columns

```
[7] data.columns
```

```
Index(['survived', 'name', 'sex', 'age'], dtype='object')
```

```
[8] categoricals = ['name', 'sex']

    numericals = ['survived', 'age']
```

Splitting columns into **categorical** and **numerical** types to apply appropriate statistical summaries

02

Explore the Data

Describe Numerical Columns

```
[9] data[numericals].describe()
```

	survived	age
count	500.000000	451.000000
mean	0.540000	35.917775
std	0.498897	14.766454
min	0.000000	0.666700
25%	0.000000	24.000000
50%	1.000000	35.000000
75%	1.000000	47.000000
max	1.000000	80.000000

Observations:

1. The min and max values make sense for each column.
2. Mean ~ 50%(median) in ‘age’ value indicating somewhat a symmetrical distribution.
3. The ‘survived’ column is boolean/binary column since the value is either 0 or 1, so we don't need to conclude its simmetricity. We only need to check the balance level.
4. Given the mean of ‘survived’ = 0.54 (just a bit over 0.5), we can conclude that there are slightly more survivors (1s) than non-survivors (0).

02

Explore the Data

Describe Categorical Columns

```
[10] data[categoricals].describe()
```

	name	sex
count	500	500
unique	499	2
top	Eustis, Miss. Elizabeth Mussey	male
freq	2	288

Observations:

1. The ‘name’ column has 2 duplicated values, shown by the 499 unique values instead of 500.
2. The Titanic survivors are mostly male (288 rows), and the rest are female.
3. There are no missing values in categorical columns.

02

Explore the Data

Numerical Details

```
[12] for col in numericals:
         print(f"==== {col} ====")
         display(data[col].value_counts())
         print()
```

```
==== survived ====
```

survived	count
1	270
0	230

dtype: int64

```
==== age ====
```

age	count
24.0000	23
30.0000	20
36.0000	19
18.0000	14
42.0000	14

Observations:

1. The dataset shows that 270 passengers survived, which is slightly higher than the 230 passengers who did not survive. This suggests a relatively balanced but survival-favored distribution.
2. The ages 24, 30, and 36 occur most frequently among passengers, indicating these ages were more common on board.

02

Explore the Data

Categorical Details

```
[11] for col in categoricals:
         print(f"Value counts of {col} column")
         display(data[col].value_counts())
         print()
```

```
Value counts of name column
```

name	count
Eustis, Miss. Elizabeth Mussey	2
Becker, Miss. Ruth Elizabeth	1
Becker, Miss. Marion Louise	1

```
Value counts of sex column
```

sex	count
male	288
female	212

dtype: int64

Observations:

1. Two duplicate entries were found with the name "Eustis, Miss. Elizabeth Mussey".
2. The number of male passengers (288) is higher than female passengers (212), indicating a gender imbalance in the dataset.

03

Data Cleansing

Identifying Duplicated Values

```
[13] len(data.drop_duplicates())/len(data)
```

```
0.998
```

There are duplicated values in this dataset, proved by the above simple equation that is **below 1.**

```
[15] duplicates = data[data.duplicated(keep=False)]

     duplicate_counts = duplicates.groupby(list(data.columns)).size().reset_index(name='duplicate_count')

     sorted_duplicates = duplicate_counts.sort_values(by='duplicate_count', ascending=False)

     sorted_duplicates
```

	survived	name	sex	age	duplicate_count
0	1	Eustis, Miss. Elizabeth Mussey	female	54.0	2

The only duplicate found in the dataset is the name **"Eustis, Miss. Elizabeth Mussey"**, which appears twice. No other duplicate records were detected.

03

Data Cleansing

Handling Duplicated Values

```
[16] data = data.drop_duplicates()
```

Handling the duplicated values using **.drop_duplicates()**

```
[17] len(data.drop_duplicates())/len(data)

1.0
```

The value of this simple equation turns out to be 1 and that means **there's no duplicated value anymore.**

03

Data Cleansing

Identifying Null Values

```
[18] data.isna().sum()
```

	0
survived	0
name	0
sex	0
age	49

dtype: int64

```
Column: survived, Missing Count: 0, Missing Percentage: 0.00%
Column: name, Missing Count: 0, Missing Percentage: 0.00%
Column: sex, Missing Count: 0, Missing Percentage: 0.00%
Column: age, Missing Count: 49, Missing Percentage: 9.82%
```

The percentage of missing values **below 20%**, so we can handle numericals using **median** and categorical using mode. Since the categoricals (name & sex) and one of numericals (survived) don't have any missing values, **we're focusing on the age numericals column.**

03

Data Cleansing

Handling Null Values

```
[24] data['age'] = data['age'].fillna(data['age'].median())
```

Handling the missing value specifically on 'age' column using **.fillna()**

```
[26] total_rows = len(data)

     for col in data.columns:
         missing_count = data[col].isna().sum()
         missing_percentage = (missing_count / total_rows) * 100
         print(f"Column: {col}, Missing Count: {missing_count}, Missing Percentage: {missing_percentage:.2f}%")
```

```
Column: survived, Missing Count: 0, Missing Percentage: 0.00%
Column: name, Missing Count: 0, Missing Percentage: 0.00%
Column: sex, Missing Count: 0, Missing Percentage: 0.00%
Column: age, Missing Count: 0, Missing Percentage: 0.00%
```

All the missing percentage value of each column are 0% and that means there's **no missing value anymore.**

03

Data Cleansing

Data Cleaning Completed

```
[25] data.info()
```

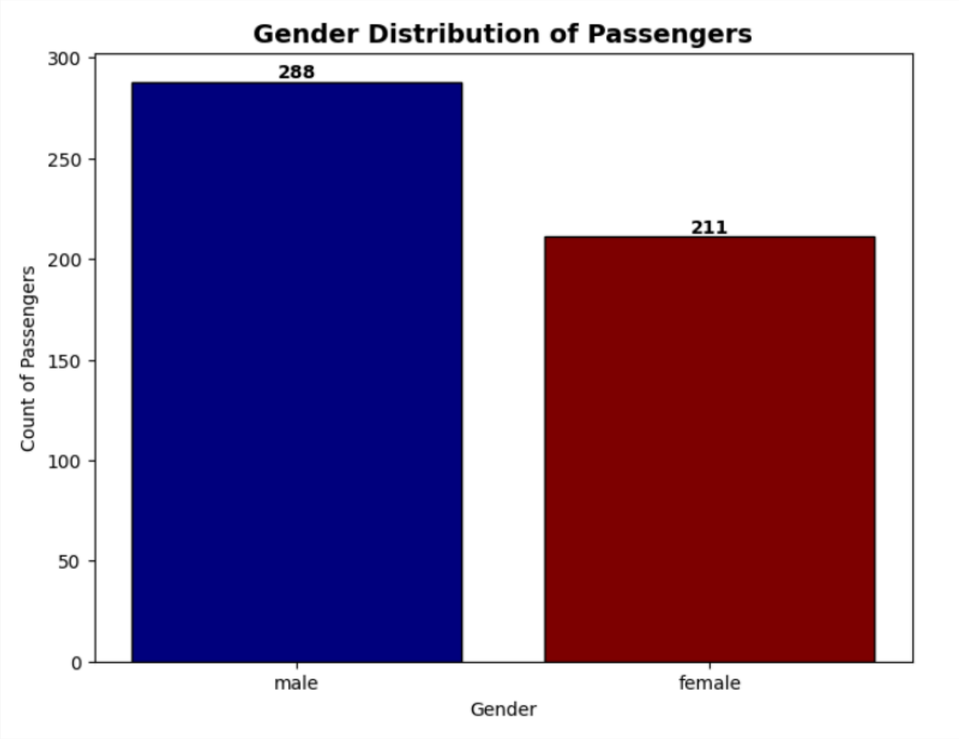
```
<class 'pandas.core.frame.DataFrame'>
Index: 499 entries, 0 to 499
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   survived  499 non-null    int64
 1   name      499 non-null    object
 2   sex       499 non-null    object
 3   age       499 non-null    float64
dtypes: float64(1), int64(1), object(2)
memory usage: 19.5+ KB
```

After completing the data cleansing process, especially by handling duplicate entries and missing values, the dataset is now cleaner, more consistent, and **ready to be used for the next stage of analysis or visualization.**

04

Get to Know the Data

Gender Distribution

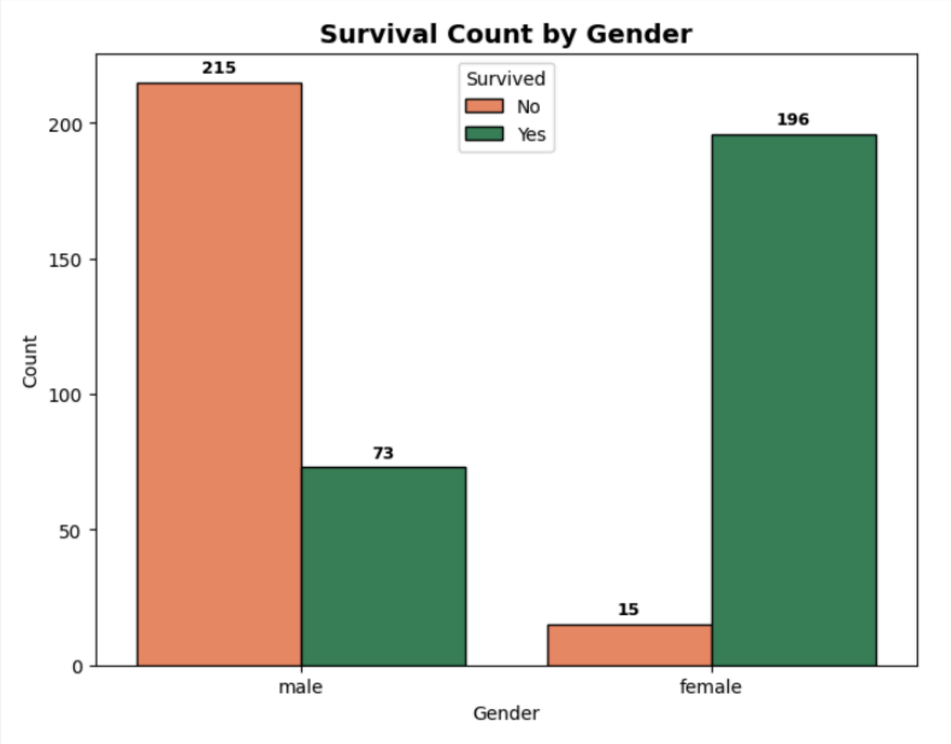


The gender distribution reveals that **male passengers made up the majority on board**, with 288 males compared to 211 females. This imbalance could play a role in survival outcomes, especially considering historical lifeboat priorities during the Titanic tragedy.

04

Get to Know the Data

Survival Count by Gender

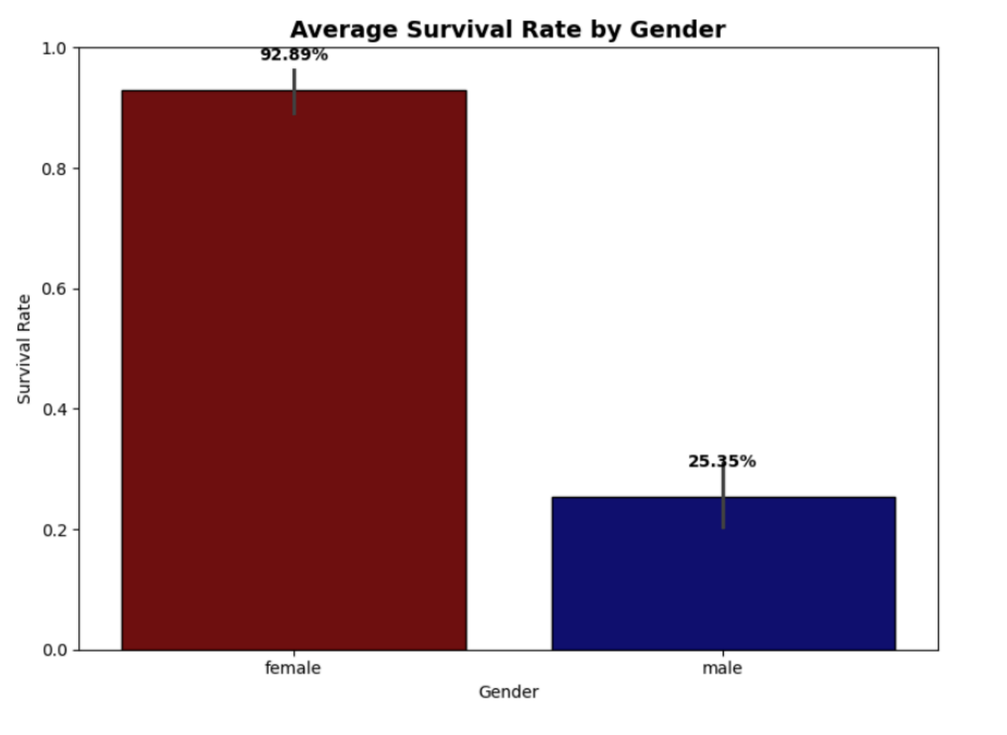


The chart above reveals a striking gender gap in survival outcomes. While most male passengers did not survive (only 73 out of 288 survived), **female passengers were far more likely to survive**, with 196 out of 211 making it. This pattern supports the historical **“women and children first”** policy during the Titanic evacuation.

04

Get to Know the Data

Average Survival Rate by Gender



The survival rate by gender **highlights a significant disparity.** 92.89% of female passengers survived, compared to only 25.35% of male passengers. This reinforces the **strong likelihood that gender played a major role in survival chances,** possibly due to evacuation protocols that prioritized women during the disaster.

THANK YOU

ibimbing

Marvin Ronny Chandiary