Titanic Database Exploration Report

1.0 Introduction

Titanic cruise is well-known in history for its tragic sinking on its first voyage. On 15th April 1912, the British passenger liner RMS Titanic, which was widely considered as *'unsinkable'*, ironically sank after ploughing into an iceberg. Limited number of lifeboats on the ship had resulted in the death of 1,502 out of 2,224 passengers and crews.

1.1 Data background

The database consists of a single table separated into **12 different columns** and contains **891 records** in total. The data type for each column was identified and any field that was different to the data format will be addressed later, if any, during the data wrangling process. Several variables had been determined as the predictor and target variables for analysis purpose. All variables were categorised accordingly into categorical and quantitative variables.

The summary of the data types and variable categories are as follows:

Column	***Data Type***	***Type of Variable***		***Variable Category***
		Predictor variable	Target variable	
PassengerID	Integer			-
Survived	Boolean (in binary)		√	Dichotomous (Categorical)
Pclass	Integer	√		Ordinal (Categorical)
Name	Text			-
Sex	Text	√		Dichotomous (Categorical)
Age	Decimal	√		Continuous (Quantitative)
SibSp	Integer			Discrete (Quantitative)
Parch	Integer			Discrete (Quantitative)
Ticket	Text			-
Fare	Decimal			Continuous (Quantitative)
Cabin	Text			Nominal (Categorical)
Embarked	Text			Nominal (Categorical)

1.2 Research questions

While there was some element of luck involved in surviving, a few questions had been posed to investigate whether any of the predictor variables had any correlation with the target variable. The research questions are as of the following:

1. Does socioeconomic status have any correlation with surviving Titanic crash?
2. Does gender have any correlation with surviving Titanic crash?
3. Does age have any correlation with surviving Titanic crash?

These questions were used to form statistical hypotheses for the analysis purpose. Along with that, descriptive statistics were also employed to lay out descriptive information on the data.

1.3 Statistical hypotheses

1	Null hypothesis, H_0	:	There is no correlation between socioeconomic status and survival in the population
	Alternative hypothesis, H_a	:	There is a correlation between socioeconomic status and survival in the population
2	Null hypothesis, H_0	:	There is no correlation between gender and survival in the population
	Alternative hypothesis, H_a	:	There is a correlation between gender and survival in the population
3	Null hypothesis, H_0	:	There is no correlation between age and survival in the population
	Alternative hypothesis, H_a	:	There is a correlation between age and survival in the population

2.0 Data Collection

Data was **collected retrospectively** and is available online through Kaggle platform (*https://www.kaggle.com/competitions/titanic/overview*) and TalentLabs learning management system. A metadata of the database is tabled below:

Column	Definition and Notes
PassengerID	• Passenger ID • Acts as the primary key
Survived	• Survival status of the passenger • 1 for '*survived*' • 0 for '*did not survive*'
Pclass	• A proxy variable for socioeconomic status • 1 is for first or upper class • 2 is for second or middle class • 3 is for third or lower class
Name	• Name of the passenger
Sex	• Gender
Age	• For passengers with age less than one year, the age is represented in a decimal that was converted from a fraction of $\frac{x}{12}$. For example, 5 months old is 0.42 years old. • Several passengers' age cannot be verified and had their ages estimated instead. Estimated age is presented in the form of xx.5
SibSp	• Number of family relations onboard in terms of siblings (brother, sister, stepbrother, stepsister) and spouse (husband and wife) • Mistresses and fiancés were excluded
Parch	• Number of family relations onboard in terms of parent (mother, father) and child (daughter, son, stepdaughter, stepson) • For children travelling with only a nanny, *Parch = 0*
Ticket	• Ticket number
Fare	• Passenger fare
Cabin	• Cabin number
Embarked	• Port of embarkation

	• C stands for Cherbourg port • Q stands for Queenstown port • S stands for Southampton port

3.0 Data Wrangling

3.1 Data validation

The database schema was examined and format for *Age* variable was in text format when supposedly to be in numerical. Changing the data format directly in SQLite is not possible, so a copy of *Age* variable was created instead and formatted in integers. The old *Age* variable was deleted.

```
ALTER TABLE passengers
RENAME COLUMN Age TO Age_text
# Renaming Age to something else so a new variable named Age can be added
ALTER TABLE passengers
ADD Age INTEGER
# Create a new Age variable in integers format
UPDATE passengers
SET Age = Age_text
WHERE Age_text IS NOT NULL
# Copy all values from the old variable onto the new one
ALTER TABLE passengers
DROP COLUMN Age_text
# Delete the old variable
```

3.2 Data accuracy

Several records in age variable were age estimation, represented by xx.5. The 0.5 here may also be interpreted as 6 months old. This may affect the results of age variable when running for descriptive and inferential statistics. To address this, all records with 0.5 decimal in *Age* variable were rounded down to the closest integer. For example, age of 28.5 will be rounded off to 28. However, data records for passengers less than 1 year old are accurate to months period, so the data were not modified.

```
WITH estimated_age AS (
        SELECT PassengerId FROM passengers
        WHERE Age LIKE '%.5')
# Select all IDs where the passengers' ages were estimated
UPDATE passengers
SET Age = CAST(Age AS INT)
WHERE PassengerId IN estimated_age IS NOT NULL
# Update the records in age variable by removing the decimals (rounding down)
```

Similarly, a few records with fare price of 0 in *Fare* variable were ambiguous. This could mean the fare was free of charge, or the fare was actually null. To avoid any ambiguity and to ensure data accuracy, this data will be treated as null.

```
UPDATE passengers
SET Fare = NULL
WHERE Fare = 0
# Update all zero fares to NULL
```

Otherwise, other data columns looked fine.

3.3 Data completeness

```
SELECT
(SELECT COUNT(*) FROM passengers WHERE PassengerId IS NOT NULL) AS PassengerId,
(SELECT COUNT(*) FROM passengers WHERE Survived IS NOT NULL) AS Survived,
(SELECT COUNT(*) FROM passengers WHERE Pclass IS NOT NULL) AS Pclass,
(SELECT COUNT(*) FROM passengers WHERE Name IS NOT NULL) AS Name,
(SELECT COUNT(*) FROM passengers WHERE Sex IS NOT NULL) AS Sex,
(SELECT COUNT(*) FROM passengers WHERE SibSp IS NOT NULL) AS SibSp,
(SELECT COUNT(*) FROM passengers WHERE Parch IS NOT NULL) AS Parch,
(SELECT COUNT(*) FROM passengers WHERE Ticket IS NOT NULL) AS Ticket,
(SELECT COUNT(*) FROM passengers WHERE Fare IS NOT NULL) AS Fare,
(SELECT COUNT(*) FROM passengers WHERE Cabin IS NOT NULL) AS Cabin,
(SELECT COUNT(*) FROM passengers WHERE Embarked IS NOT NULL) AS Embarked,
(SELECT COUNT(*) FROM passengers WHERE Age IS NOT NULL) AS Age
```

PassengerID	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age
891	891	891	891	891	891	891	891	876	204	889	714

```
# Return the number of records for each column
```

Number of records highlighted in green indicate the data are complete and there is no missing value. Meanwhile, number of records highlighted in green scale mean there were some missing values, whereas number of record highlighted in red shows that there were significant portion of null values in the data. Summary of data completeness is shown on the right table.

Column	Data Completeness (%)
PassengerID	100
Survived	100
Pclass	100
Name	100
Sex	100
SibSp	100
Parch	100
Ticket	100
Fare	98.32
Cabin	22.90
Embarked	99.78
Age	80.13

Since *Cabin* variable had poor completeness, it will not give any value to our dataset; hence the variable was removed completely from the table.

```
ALTER TABLE passengers
DROP COLUMN Cabin
# Remove Cabin variable from the table entirely
```

Even though *Ticket* column has perfect completeness, it stored useless data. Removing this data will allow database processing a little faster and saving costs for data storage.

```
ALTER TABLE passengers
DROP COLUMN Ticket
# Remove Ticket column from the table entirely
```

3.4 Data consistency

All columns were checked for data consistency in a quick look. What was meant by consistency here is to check whether a record was actually properly reflected in all variables or not. For instance, a first class ticket should reflect a high fare price.

During this process, one record was found to be inconsistent and argumentative. The record in concern was a first class ticket (*Pclass = 1*) but with an extremely low fare price (*Fare = 5*) as compared to other first class tickets. This may imply three different things:

i. The ticket fare was low because it was discounted for infant passenger
ii. There was an error on the fare price (for instance, missing one or two digits)
iii. There was a typing error when recording the class number (inputting 1 instead of 2 or 3)

Upon checking, the passenger was 33 years old, so this rules out the first implication and leaves out 2 remaining possibilities. Since the error couldn't be pinpointed on which field, this record was deleted from the database due to inconsistency.

```
DELETE FROM passengers
WHERE Fare = 5 AND Pclass = 1
# Delete the inconsistent record from the database
```

On top of that, any spelling error, capitalization, and other structural errors were looked for to ensure consistency throughout the database. No other inconsistencies were spotted.

3.5 Data uniqueness

Database was scanned for any duplicate entries.

```
SELECT COUNT(DISTINCT PassengerId), COUNT(DISTINCT Name)
FROM passengers
# Both count returns all 890 distinctive records
```

No duplicate entries were found in this process.

3.6 Data uniformity

Database was inspected once more for any discrepancy in the measurement units among each column. All data records were found to be uniform.

3.7 Data making

A new variable was added by categorising data in *Age* variable into 4 age groups, which are as follows:

i. Children (0-12 years old)
ii. Adolescents (13-18 years old)
iii. Adults (19-59 years old)
iv. Seniors (60 years and above)

```
ALTER TABLE passengers
ADD "Age_Group" TEXT
# Add a new column Age_Group
UPDATE passengers
SET Age_Group = (
        CASE
                WHEN (Age < 13) THEN 'children'
                WHEN (Age >= 13) AND (Age < 19) THEN 'adolescents'
                WHEN (Age >= 19) AND (Age < 60) THEN 'adults'
                WHEN (Age >= 60) THEN 'seniors'
                ELSE NULL
        END)
# Categorise age into groups based on specified ranges
```

3.8 Summary of data cleansing

Data Inspection	Details
Validation	• Format for *Age* variable was changed from text to integers.
Accuracy	• Data in *Age* and *Fare* variable were found to be ambiguous. • For *Age* variable, data for passengers aged 1 year and above that contains .5 decimals were rounded down to the closest integer. • For *Fare* variable, fare prices of 0 were converted to null.
Completeness	• *Cabin* variable was deleted due to poor data completeness. • *Ticket* column had complete data but it was rendered useless, so it was removed to save storage cost and reduce processing time. • Three other variables – *Age, Fare,* and *Embarked* had slight incompleteness but no imputation or deletion were made.
Consistency	• One record with *Pclass* = 1, *Fare* = 5 was deleted for being inconsistent, leaving a total of 890 records.
Uniqueness	• No duplicate entry was found. All data were unique.
Uniformity	• All data were uniform. No changes was made.
Making	• A new variable named *Age_Group* was added. • Children – 0 to 12 years old • Adolescents – 13 to 18 years old • Adults – 19 to 59 years old • Seniors – 60 years old and above

Before Cleansing		After Cleansing	
Column	Data type (SQL)	Column	Data type (SQL)
PassengerID	Integer	*PassengerID*	Integer
Survived	Integer	*Survived*	Integer
Pclass	Integer	*Pclass*	Integer
Name	Text	*Name*	Text
Sex	Text	*Sex*	Text
Age	Text	*SibSp*	Integer
SibSp	Integer	*Parch*	Integer
Parch	Integer	*Fare*	Integer
Ticket	Text	*Embarked*	Text
Fare	Integer	*Age*	Integer
Cabin	Text	*Age_Group*	Text
Embarked	Text		
Number of records: 891		*Number of records: 890*	

**Red boxes indicate columns that were deleted during data cleansing*

***Green boxes indicate new columns that were added during data cleansing*

4.0 Data Analysis

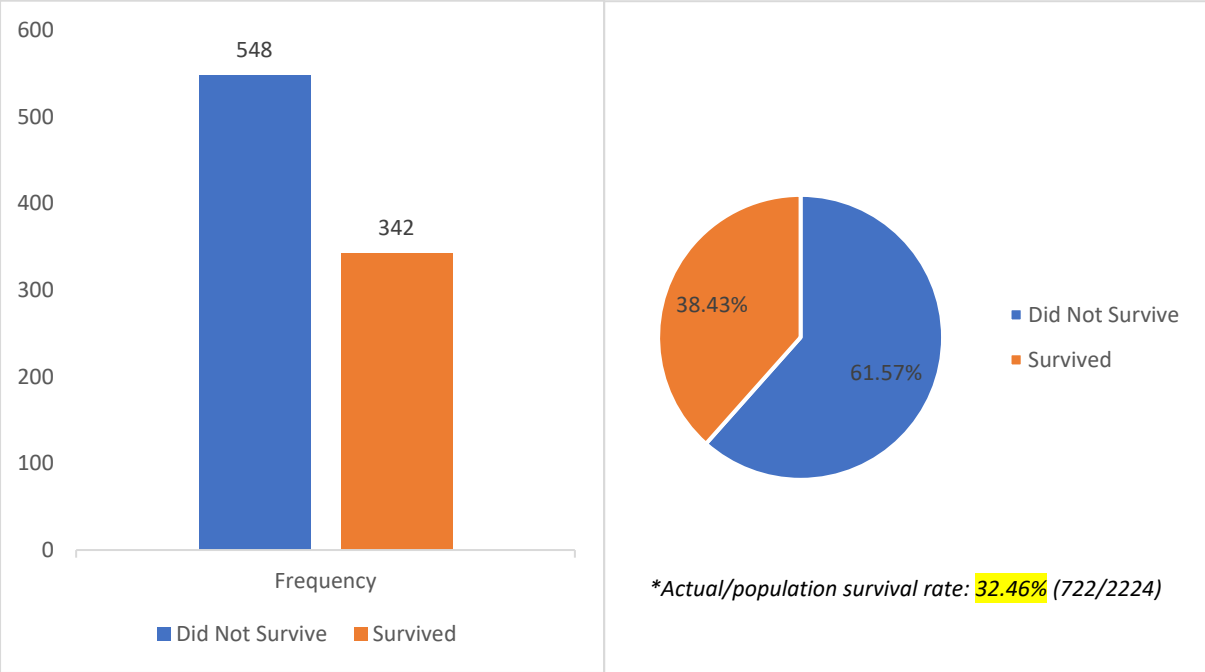
4.1 Univariate analysis

4.1.1 Data distribution

Survived variable

```
SELECT
(SELECT COUNT(Survived) FROM passengers WHERE Survived = 1) AS survived,
(SELECT COUNT(Survived) FROM passengers WHERE Survived = 0) AS not_survived
# Returns the following table:
```

survived	not_survived
342	548



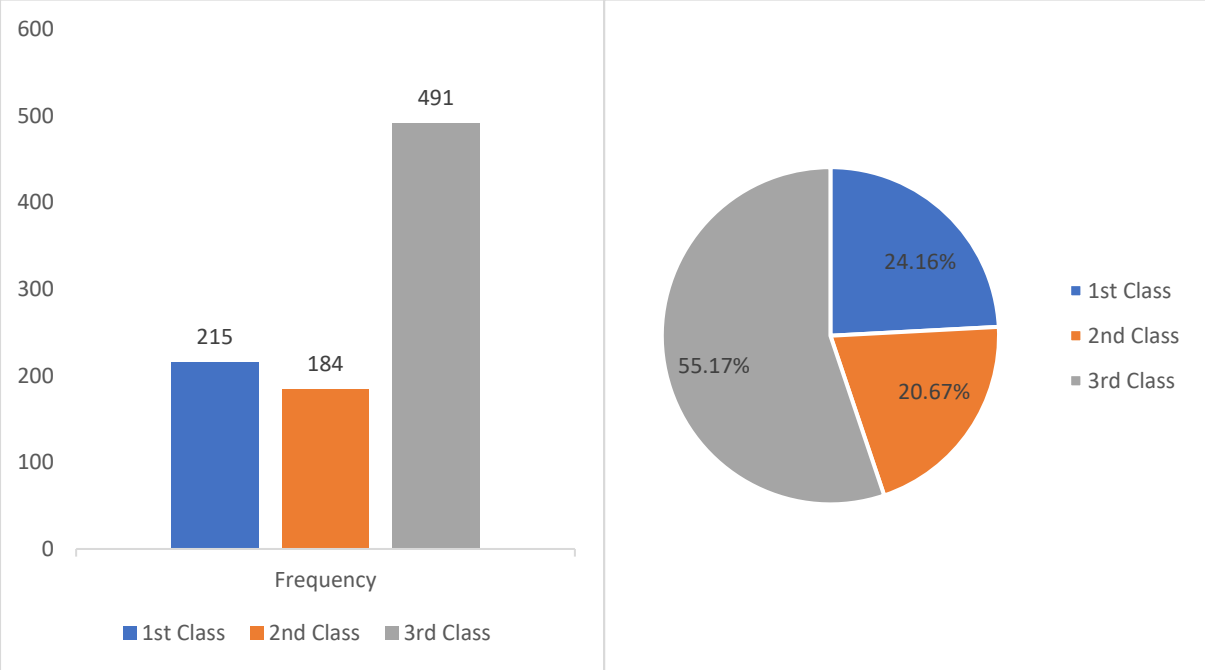
Simple and percentage frequency distribution of survival

Almost two thirds of the sample data did not survive the Titanic crash, recording 548 deaths out of 890 passengers. Only around one third (38.43% or 342/890) of the passengers did survive. This doesn't deviate much from the actual survival rate which was at 32.46%. We could say the data was not really biased and data sampling was totally random.

Pclass variable

```
SELECT
(SELECT COUNT(Pclass) FROM passengers WHERE Pclass = 1) AS first_class,
(SELECT COUNT(Pclass) FROM passengers WHERE Pclass = 2) AS second_class,
(SELECT COUNT(Pclass) FROM passengers WHERE Pclass = 3) AS third_class
# Returns the following table:
```

first_class	second_class	third_class
215	184	491



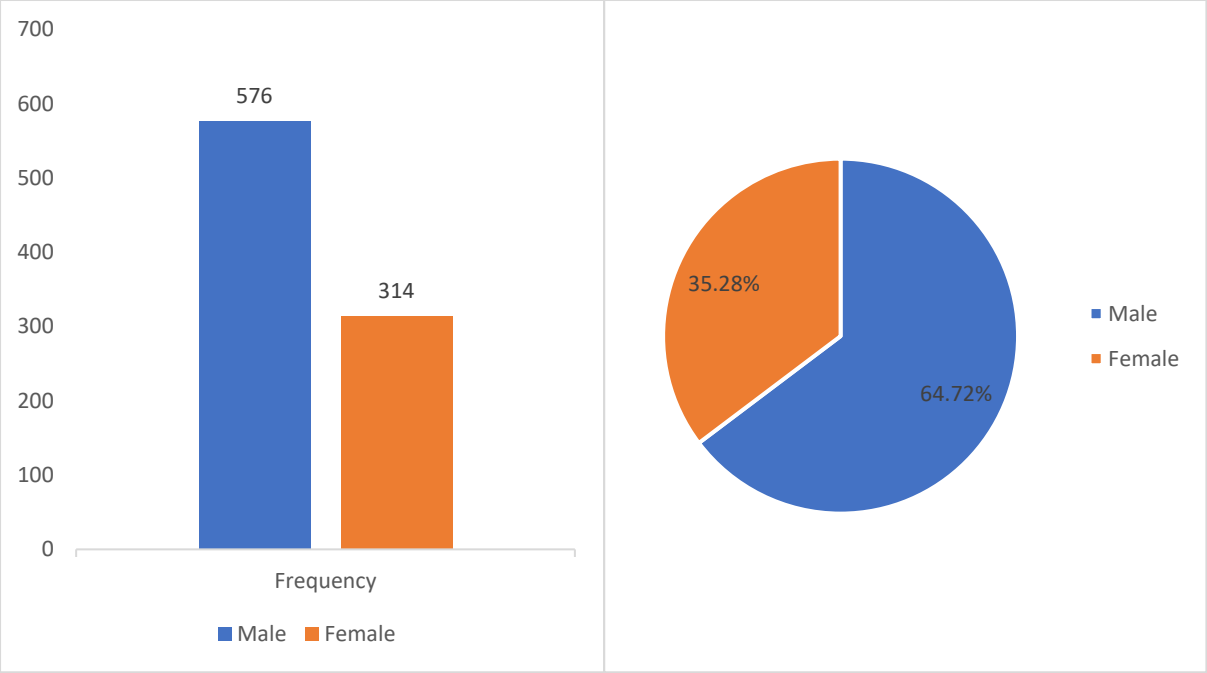
Simple and percentage frequency distribution of socioeconomic status

First class and second class passengers were almost equally distributed, whereas third class passengers constitute more than half of the total passengers, which was at 55.17%. It was followed by first class passengers with 24.16% and finally, second class passengers with 20.67% of total passengers.

Sex variable

```
SELECT
(SELECT COUNT(Sex) FROM passengers WHERE Sex = 'male') AS male,
(SELECT COUNT(Sex) FROM passengers WHERE Sex = 'female') AS female
# Returns the following table:
```

male	female
576	314



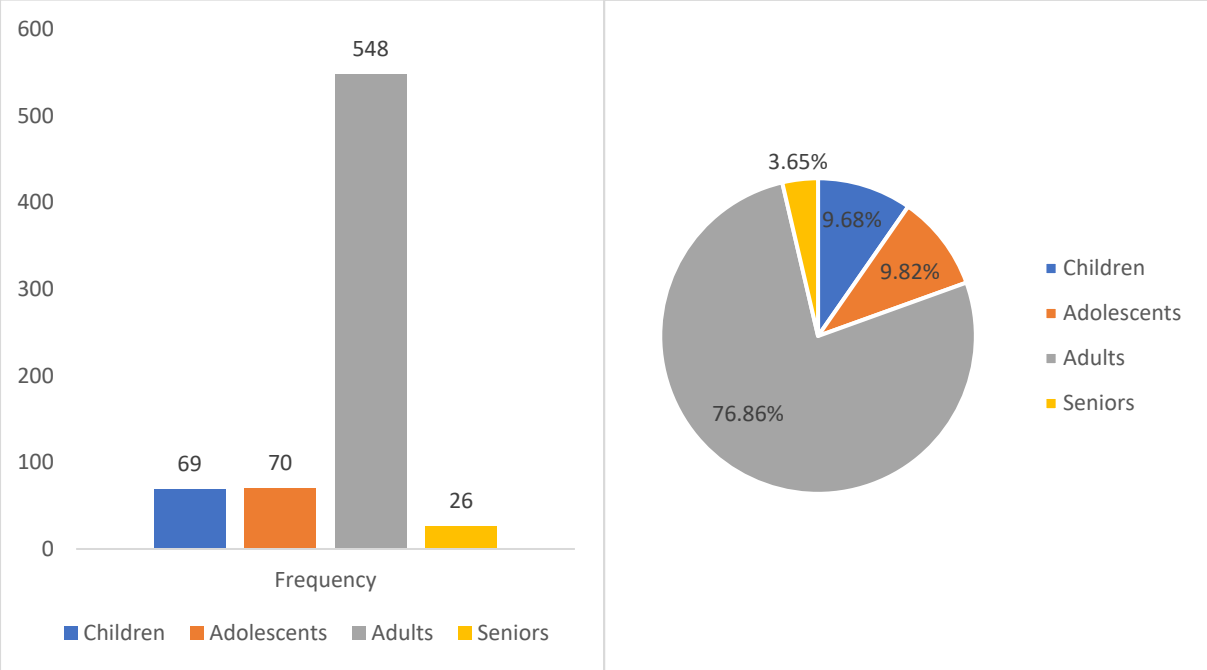
Simple and percentage frequency distribution of gender

Most of the passengers were male, with close to two thirds (64.72%) of the total passengers. Female passengers only make up around one third of the total passengers with 35.28% constitution.

Age variable

```
SELECT
(SELECT COUNT(Age_Group) FROM passengers
        WHERE Age_Group = 'children') AS 'children',
(SELECT COUNT(Age_Group) FROM passengers
        WHERE Age_Group = 'adolescents') AS 'adolescents',
(SELECT COUNT(Age_Group) FROM passengers
        WHERE Age_Group = 'adults') AS 'adults',
(SELECT COUNT(Age_Group) FROM passengers
        WHERE Age_Group = 'seniors') AS 'seniors'
# Returns the following table:
```

children	adolescents	adults	seniors
69	70	548	26



Simple and percentage frequency distribution of age by groups

Adults were dominating in age distribution with more than three quarters of overall (76.86%). This makes sense since people who went for a cruise vacation are typically working adults. Children, adolescents, and seniors are usually only tagging along with their family members, which may explain why they only comprise a small portion of the pie.

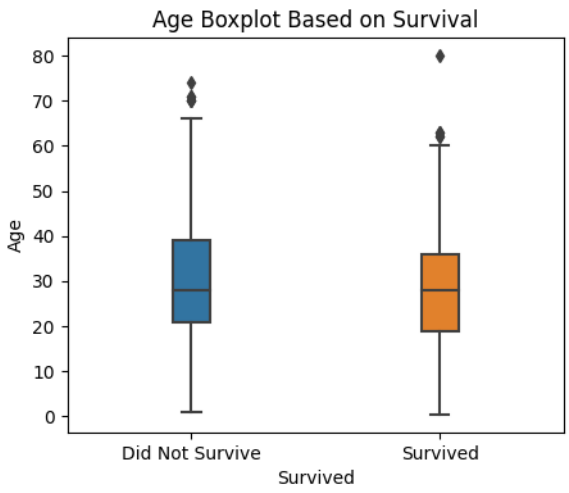
For further analysis, the database was exported as a CSV file and processed using Python with libraries for data analytics such as NumPy, SciPy, ResearchPy, Pandas, Seaborn, and Matplotlib.

```
# Import relevant libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

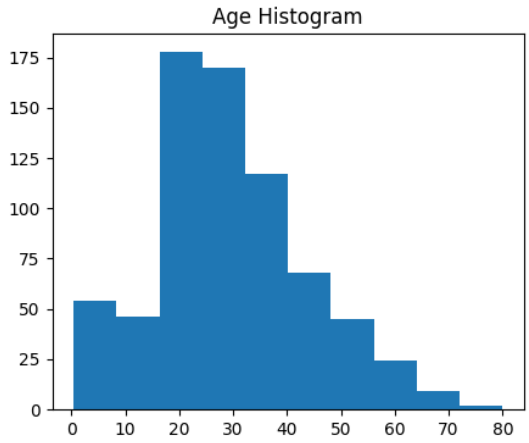
To check for outliers in *Age* variable, the data was plotted in Python using Pandas and Matplotlib.

```
# Load the database and relevant variables
dataframe = pd.read_csv('passengers.csv')
survived = dataframe['Survived']
age = dataframe['Age']
filtered_age = age[~np.isnan(age)]  # Exclude null values in age variable

# Set the figure size
plt.rcParams['figure.figsize'] = (5, 4)

# Plot the graph
sns.boxplot(x=survived, y=filtered_age, data=dataframe, width=0.15)

# Labelling
plt.xticks([0, 1], ['Did Not Survive', 'Survived'])
plt.title('Age Boxplot Based on Survival')

# Display the graph
plt.show()
```

A number of outliers were noted from the graph. Positive skewness or right-skewed histogram is expected due to existence of the outliers on the top end.



```
# Plot the graph
plt.hist(filtered_age)

# Labelling
plt.title('Age Histogram')

# Display the graph
plt.show()
```



From here, it was clearly seen the data did not follow normal distribution and skewed to the right as expected. This can be testified further using Shapiro-Wilk normality test.

```
# Shapiro-Wilk normality test
shapiro = stats.shapiro(filtered_age)
print(shapiro)

# Returns p-value = 6.8177x10-8
```

Since p-value 6.8177×10^{-8} is less than significance value $\alpha = 0.05$, we reject the null hypothesis of Shapiro-Wilk test. The sample data did not come from a normal distribution. Since most people who went for a cruise vacation were working adults and pensioners, the occurrence of children and adolescents boarding as passengers were much lower, hence resulting in positive skewness of the data. Thus, a nonparametric test is recommended for a correlation test.

4.1.2 Central tendency

```
# Measure of central tendency
mean_age = np.mean(filtered_age)
median_age = np.median(filtered_age)
mode_age = stats.mode(filtered_age, keepdims=True)

print(f'''Mean: {mean_age}
Median: {median_age}
Mode: {mode_age}''')

# Output:
# Mean: 29.681865357643755
# Median: 28.0
# Mode: ModeResult(mode=array([24.]), count=array([31]))
```

Mean age is 29.68 years, while median is 28 years. The most frequent age is 24 years, with 31 frequencies. This can also be calculated using SQL.

```
SELECT AVG(Age) AS Mean
FROM passengers
WHERE Age IS NOT NULL
# Returns mean value of 29.6818653576438
SELECT AVG(Age) AS Median
FROM (SELECT Age
        FROM passengers
        WHERE Age IS NOT NULL
        ORDER BY Age ASC
        LIMIT 2 - (SELECT COUNT(Age) FROM passengers) % 2
        OFFSET (SELECT (COUNT(Age) - 1) / 2
                FROM passengers
                WHERE Age IS NOT NULL))
# Returns median value of 28.0
SELECT Age AS Mode, COUNT(Age) AS Frequency
FROM passengers
WHERE Age IS NOT NULL
GROUP BY Age
ORDER BY Frequency DESC
LIMIT 1
# Returns mode as 24 with frequency of 31
```

4.1.3 Variability

```
# Measure of variability
variance_age = filtered_age.var()
std_age = filtered_age.std()

print(f'''Variance: {variance_age}
Standard deviation: {std_age}''')

# Calculate range and inter-quartile range
sort_age = sorted(filtered_age)
range1 = max(sort_age) - min(sort_age)
print(f'Range: {range1}')

q1_index = (len(sort_age) + 1) * 0.25
q3_index = (len(sort_age) + 1) * 0.75
if q1_index != int(q1_index):
    Q1 = (sort_age[int(q1_index - 1.5)] + sort_age[int(q1_index - 0.5)]) / 2
    Q3 = (sort_age[int(q3_index - 1.5)] + sort_age[int(q3_index - 0.5)]) / 2
else:
    Q1 = sort_age[q1_index]
    Q3 = sort_age[q3_index]

IQR = Q3 - Q1
print(f'IQR: {IQR}')

# Output:
# Variance: 211.16518542004826
# Standard deviation: 14.531523850582508
# Range: 79.58
# IQR: 18.0
```

The variance, σ^2 and standard deviation, σ for age are 211.17 and 14.53 respectively. Whereas the range and inter-quartile range are 79.58 and 18.0 respectively.

4.2 Bivariate analysis

Correlation between socioeconomic status and survival

```
# Import relevant libraries
import pandas as pd
import scipy.stats as stats
import researchpy as rp

# Load the database and relevant variables
dataframe = pd.read_csv('passengers.csv')
survived = dataframe['Survived']
pclass = dataframe['Pclass']

# Crosstab Pclass * Survived
pcl_sur_tab = pd.crosstab(pclass, survived, margins=True)
print(f'{pcl_sur_tab}\n')

# Output:
Survived    0    1  All
Pclass
1          79  136  215
2          97   87  184
3         372  119  491
All       548  342  890

# Pearson's Chi-Squared correlation test
chi2_stat, p, dof, expected = stats.chi2_contingency(chisqt)
print(f'''Chi-square statistics: {chi2_stat:.5g}
p-value: {p:.5g}
Degrees of freedom: {dof}
Expected frequencies:\n
{expected}''')

# Output:
Chi-square statistics: 103.91
p-value: 3.8382e-20
Degrees of freedom: 6
Expected frequencies:

[[132.38202247 113.29438202 302.32359551 548.        ]
 [ 82.61797753  70.70561798 188.67640449 342.        ]
 [215.         184.         491.         890.        ]]
```

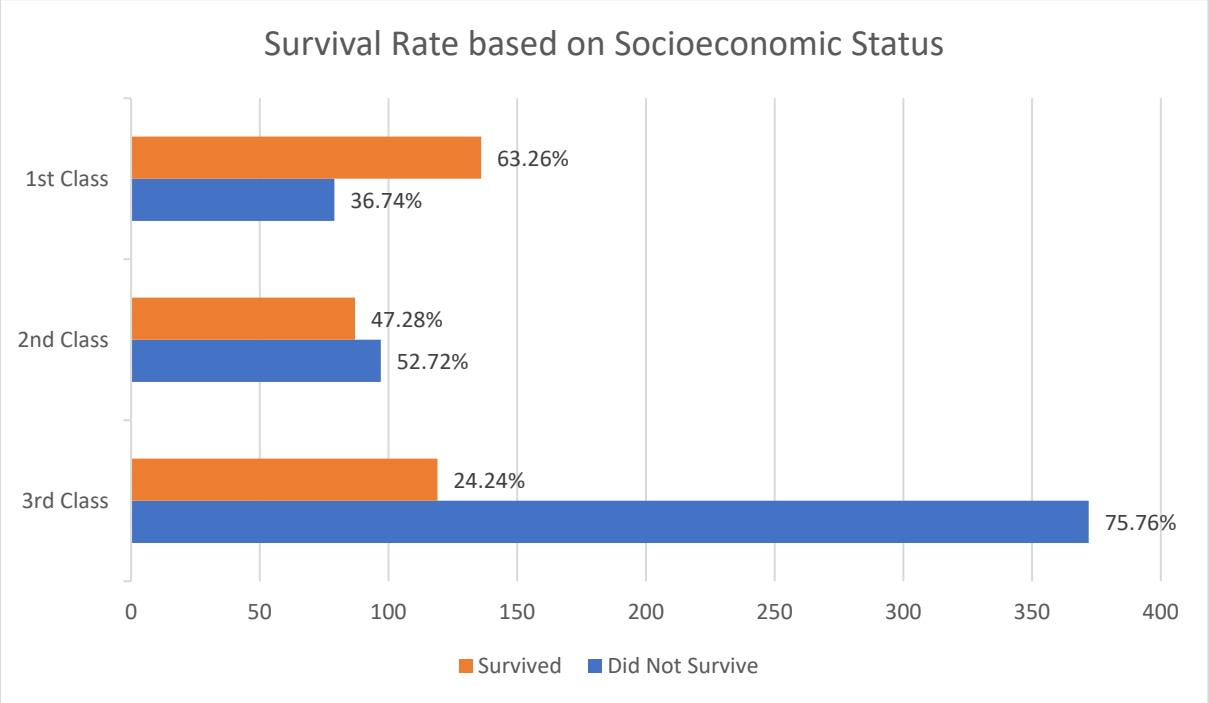
Since p-value, 3.8382×10^{-20} is lower than significance value, $\alpha = 0.05$, we reject the null hypothesis of Pearson's chi-squared correlation test. There is a correlation between socioeconomic status and surviving Titanic crash. The strength of the relationship can be measured using Cramer's V test.

```
# Cramer's V test
crosstab, res = rp.crosstab(pclass, survived, test='chi-square')
print(res)

# Output:
                Chi-square test    results
0  Pearson Chi-square ( 2.0) =   103.9054
1                     p-value =     0.0000
```

```
2                 Cramer's V =     0.3417

# Degree of freedom
deg_free = min(pcl_sur_tab.shape[0], pcl_sur_tab.shape[1]) - 1
print(f'Degree of freedom: {deg_free}')

# Output: Degree of freedom: 2

# Correlation strength
V = res.iloc[2, 1]
if deg_free == 1:
    if V < 0.1:
        strength = 'negligible'
    elif 0.1 <= V < 0.3:
        strength = 'weak'
    elif 0.3 <= V < 0.5:
        strength = 'moderate'
    elif V >= 0.5:
        strength = 'strong'
if deg_free == 2:
    if V < 0.07:
        strength = 'negligible'
    elif 0.07 <= V < 0.21:
        strength = 'weak'
    elif 0.21 <= V < 0.35:
        strength = 'moderate'
    elif V >= 0.35:
        strength = 'strong'
if deg_free == 3:
    if V < 0.06:
        strength = 'negligible'
    elif 0.06 <= V < 0.17:
        strength = 'weak'
    elif 0.17 <= V < 0.29:
        strength = 'moderate'
    elif V >= 0.29:
        strength = 'strong'
if deg_free == 4:
    if V < 0.05:
        strength = 'negligible'
    elif 0.05 <= V < 0.15:
        strength = 'weak'
    elif 0.15 <= V < 0.25:
        strength = 'moderate'
    elif V >= 0.25:
        strength = 'strong'
if deg_free == 5:
    if V < 0.05:
        strength = 'negligible'
    elif 0.05 <= V < 0.13:
        strength = 'weak'
    elif 0.13 <= V < 0.22:
        strength = 'moderate'
    elif V >= 0.22:
        strength = 'strong'

print(f'The correlation strength is {strength}')
# Output:
The correlation strength is moderate
```

```
WITH table1 AS (
        SELECT COUNT(Survived) AS survive, Pclass
        FROM passengers
        WHERE Survived = 1
        GROUP BY Pclass),
table2 AS (
        SELECT COUNT(Survived) AS didnotsurvive, Pclass
        FROM passengers
        WHERE Survived = 0
        GROUP BY Pclass)
SELECT table1.Pclass, survive, didnotsurvive
FROM table1 JOIN table2
        ON table1.Pclass = table2.Pclass
# Returns the following table:
```

Pclass	survive	didnotsurvive
1	136	79
2	87	97
3	119	372



Based on the result of the statistical tests, there is a **moderate correlation** between socioeconomic status and survival rate. As seen in the bar chart, first class passengers were more likely to survive with 63.26% survival rate, followed by second class passengers and third class passengers with 47.28% and 24.24% survival rate respectively.

This data may suggest that passengers of higher class were given utmost priority to board the lifeboats or their cabins were located much closer to one of the emergency exits, as compared with passengers from lower class.

Correlation between gender and survival

```
# Import relevant libraries
import pandas as pd
import scipy.stats as stats
import researchpy as rp

# Load the database and relevant variables
dataframe = pd.read_csv('passengers.csv')
survived = dataframe['Survived']
gender = dataframe['Sex']

# Crosstab Gender * Survived
sex_sur_tab = pd.crosstab(gender, survived, margins=True)
print(f'{sex_sur_tab}\n')

# Output:
Survived    0    1  All
Sex
female     81  233  314
male      467  109  576
All       548  342  890

# Pearson's Chi-Squared correlation test
chi2_stat, p, dof, expected = stats.chi2_contingency(sex_sur_tab)
print(f'''Chi-square statistics: {chi2_stat:.5g}
p-value: {p:.5g}
Degrees of freedom: {dof}
Expected frequencies:\n
{expected}''')

# Output:
Chi-square statistics: 262.47
p-value: 1.3407e-55
Degrees of freedom: 4
Expected frequencies:

[[193.33932584 120.66067416 314.        ]
 [354.66067416 221.33932584 576.        ]
 [548.         342.         890.        ]]
```

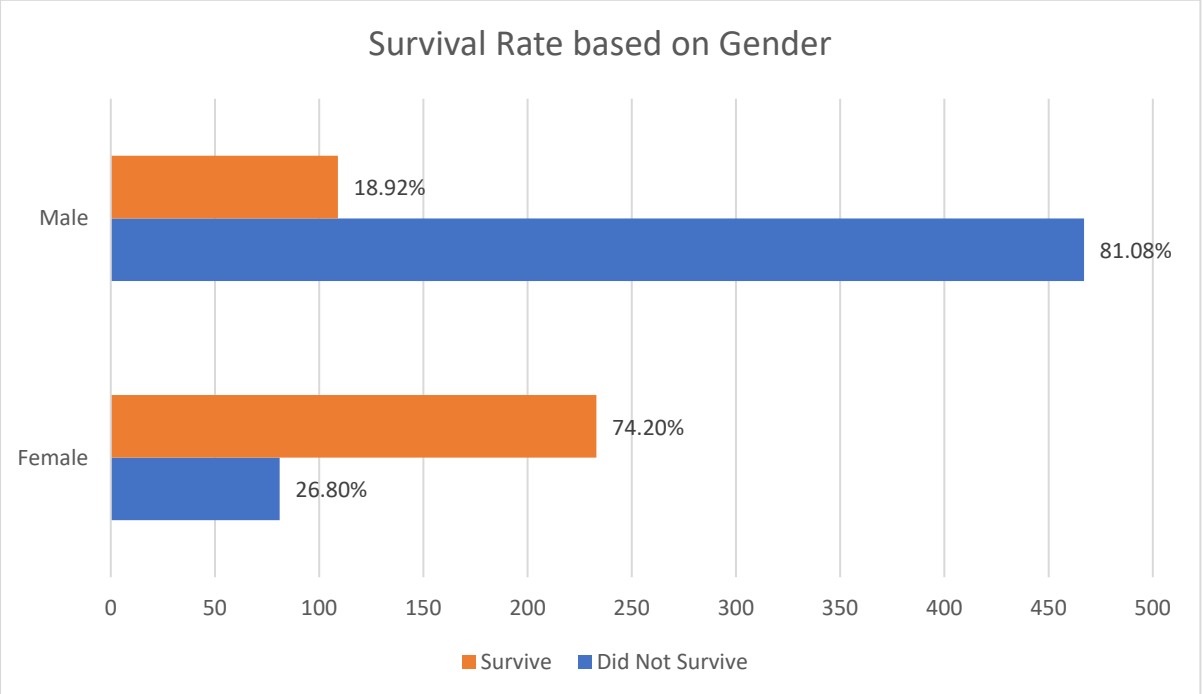
Since p-value, 1.3407×10^{-55} is lower than significance value, $\alpha = 0.05$, we reject the null hypothesis of Pearson's chi-squared correlation test. There is a correlation between gender and surviving Titanic crash. The strength of the relationship can be measured using Cramer's V test.

```
# Cramer's V test
crosstab, res = rp.crosstab(gender, survived, test='chi-square')
print(res)

# Output:
                Chi-square test   results
0  Pearson Chi-square ( 1.0) =   262.4671
1                       p-value =     0.0000
2                Cramer's phi =     0.5431
```

```python
# Degree of freedom
deg_free = min(sex_sur_tab.shape[0], sex_sur_tab.shape[1]) - 1
print(f'Degree of freedom: {deg_free}')

# Output:
Degree of freedom: 2

# Correlation strength
V = res.iloc[2, 1]
if deg_free == 1:
    if V < 0.1:
        strength = 'negligible'
    elif 0.1 <= V < 0.3:
        strength = 'weak'
    elif 0.3 <= V < 0.5:
        strength = 'moderate'
    elif V >= 0.5:
        strength = 'strong'
if deg_free == 2:
    if V < 0.07:
        strength = 'negligible'
    elif 0.07 <= V < 0.21:
        strength = 'weak'
    elif 0.21 <= V < 0.35:
        strength = 'moderate'
    elif V >= 0.35:
        strength = 'strong'
if deg_free == 3:
    if V < 0.06:
        strength = 'negligible'
    elif 0.06 <= V < 0.17:
        strength = 'weak'
    elif 0.17 <= V < 0.29:
        strength = 'moderate'
    elif V >= 0.29:
        strength = 'strong'
if deg_free == 4:
    if V < 0.05:
        strength = 'negligible'
    elif 0.05 <= V < 0.15:
        strength = 'weak'
    elif 0.15 <= V < 0.25:
        strength = 'moderate'
    elif V >= 0.25:
        strength = 'strong'
if deg_free == 5:
    if V < 0.05:
        strength = 'negligible'
    elif 0.05 <= V < 0.13:
        strength = 'weak'
    elif 0.13 <= V < 0.22:
        strength = 'moderate'
    elif V >= 0.22:
        strength = 'strong'

print(f'The correlation strength is {strength}')

# Output:
The correlation strength is strong
```

```
WITH table1 AS (
        SELECT COUNT(Survived) AS survive, Sex
        FROM passengers
        WHERE Survived = 1
        GROUP BY Sex),
table2 AS (
        SELECT COUNT(Survived) AS didnotsurvive, Sex
        FROM passengers
        WHERE Survived = 0
        GROUP BY Sex)
SELECT table1.Sex, survive, didnotsurvive
FROM table1 JOIN table2
        ON table1.Sex = table2.Sex
# Returns the following table:
```

Sex	survive	didnotsurvive
female	233	81
male	109	467



Based on the result of the statistical tests, there is a **strong correlation** between gender and survival rate. As illustrated in the bar chart, women were more likely to survive with a whopping 74.20% survival rate. In contrast, there was only 18.92% survival rate for men.

The most plausible explanation of why male passengers had underwhelming survival rate is because men were (and still are) considered as figure of leaders who would automatically assume responsibilities in a situation where there is no clear figure of authority or person in charge. In such an unexpected emergency situation, men are most likely to stay behind and prioritize evacuating others whom they think are more vulnerable to danger – children, elderlies, and women. Putting others before themselves, in turn, had put them in a higher risk of death, and eventually lower chance of surviving.

Correlation between age and survival

The correlation between age and survival was conducted using two different variables. One is by using categorical variable (*Age_Group*), and another using quantitative variable (*Age*).

For correlation between age group and survival, Pearson's chi-squared correlation test was used.

```
# Import relevant libraries
import pandas as pd
import scipy.stats as stats
import researchpy as rp
import statsmodels.formula.api as smf
import numpy as np
from sklearn.metrics import confusion_matrix

# Load the database and relevant variables
dataframe = pd.read_csv('passengers.csv')
subset = dataframe.iloc[:, [1, 9, 10]]
subset1 = subset.copy()
subset1.dropna(inplace=True)
survived = subset1['Survived']
age = subset1['Age']
age_group = subset1['Age_Group']

# Crosstab Age Group * Survived
agroup_sur_tab = pd.crosstab(age_group, survived, margins=True)
print(f'{agroup_sur_tab}\n')
```

Output:

Age_Group / Survived	0	1	All
Adolescents	40	30	70
Adults	335	213	548
Children	29	40	69
Seniors	19	7	26
All	423	290	713

```
# Pearson's Chi-Squared correlation test
chi2_stat, p, dof, expected = stats.chi2_contingency(agroup_sur_tab)
print(f'''Chi-square statistics: {chi2_stat:.5g}
p-value: {p:.5g}
Degrees of freedom: {dof}
Expected frequencies:\n
{expected}''')

# Output:
Chi-square statistics: 11.471
p-value: 0.17641
Degrees of freedom: 8
```

```
Expected frequencies:

[[ 41.52875175  28.47124825  70.        ]
 [325.11079944 222.88920056 548.        ]
 [ 40.93548387  28.06451613  69.        ]
 [ 15.42496494  10.57503506  26.        ]
 [423.         290.         713.        ]]
```

The p-value is 0.17641, which is greater than significance value, α = 0.05. Hence, the result is not statistically significant. However, it's important to note that the result may be inaccurate and inconclusive as Pearson's chi-squared test performs poorly when the sample size is too small. As in *Age_Group* variable, we have 3 groups – adolescents, children, and seniors with sample size less than 100.

Cramer's V test can be used to test the correlation between two categorical variables without being affected by sample sizes.

```
# Cramer's V test
crosstab, res = rp.crosstab(age_group, survived, test='chi-square')
print(res)

# Degree of freedom
deg_free = min(agroup_sur_tab.shape[0], agroup_sur_tab.shape[1]) - 1
print(f'Degree of freedom: {deg_free}')

# Correlation strength
V = res.iloc[2, 1]
if deg_free == 1:
    if V < 0.1:
        strength = 'negligible'
    elif 0.1 <= V < 0.3:
        strength = 'weak'
    elif 0.3 <= V < 0.5:
        strength = 'moderate'
    elif V >= 0.5:
        strength = 'strong'
if deg_free == 2:
    if V < 0.07:
        strength = 'negligible'
    elif 0.07 <= V < 0.21:
        strength = 'weak'
    elif 0.21 <= V < 0.35:
        strength = 'moderate'
    elif V >= 0.35:
        strength = 'strong'
if deg_free == 3:
    if V < 0.06:
        strength = 'negligible'
    elif 0.06 <= V < 0.17:
        strength = 'weak'
    elif 0.17 <= V < 0.29:
        strength = 'moderate'
    elif V >= 0.29:
        strength = 'strong'
if deg_free == 4:
    if V < 0.05:
```

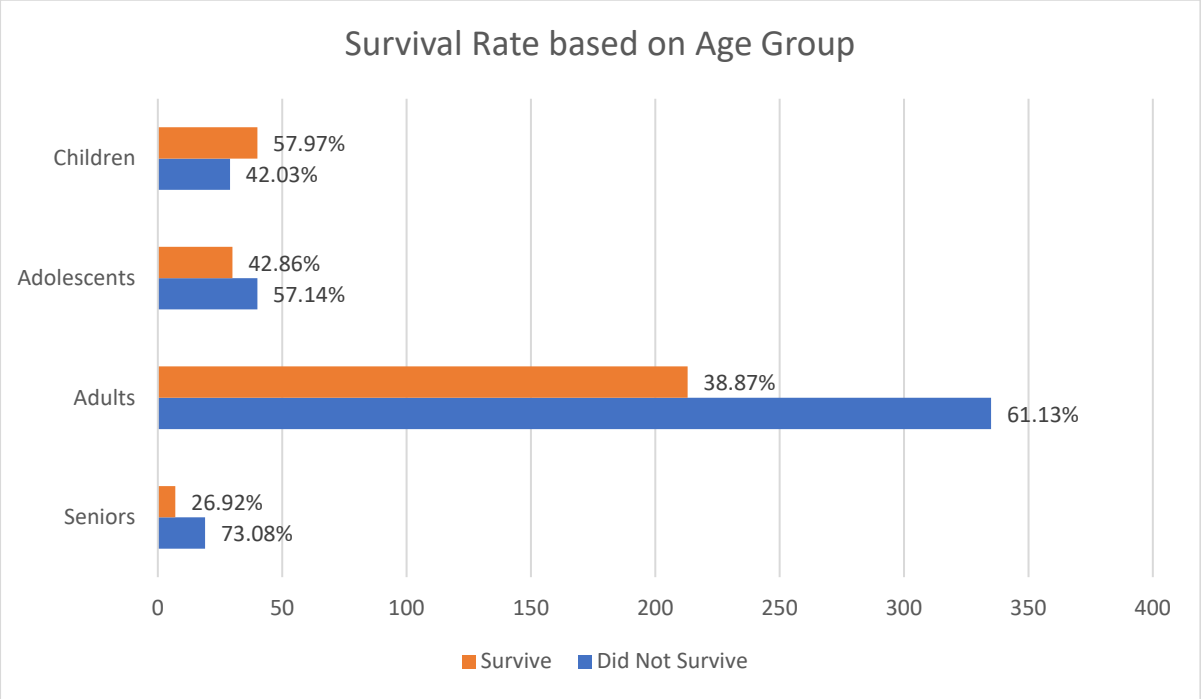
```
        strength = 'negligible'
    elif 0.05 <= V < 0.15:
        strength = 'weak'
    elif 0.15 <= V < 0.25:
        strength = 'moderate'
    elif V >= 0.25:
        strength = 'strong'
if deg_free == 5:
    if V < 0.05:
        strength = 'negligible'
    elif 0.05 <= V < 0.13:
        strength = 'weak'
    elif 0.13 <= V < 0.22:
        strength = 'moderate'
    elif V >= 0.22:
        strength = 'strong'

print(f'The correlation strength is {strength}')

# Output:
                Chi-square test  results
0  Pearson Chi-square ( 3.0) =   11.4711
1                   p-value =    0.0094
2                 Cramer's V =    0.1268
Degree of freedom: 2
The correlation strength is weak
```

The p-value this time is 0.0094, which is statistically significant. The V coefficient, however, is only 0.1268 with degree of freedom of 2, indicating that there is a **weak correlation** between age group and survival rate.

```
WITH table1 AS (
        SELECT COUNT(Survived) AS survive, Age_Group
        FROM passengers
        WHERE Survived = 1 AND Age_Group IS NOT NULL
        GROUP BY Age_Group),
table2 AS (
        SELECT COUNT(Survived) AS didnotsurvive, Age_Group
        FROM passengers
        WHERE Survived = 0 AND Age_Group IS NOT NULL
        GROUP BY Age_Group)
SELECT table1.Age_Group, survive, didnotsurvive
FROM table1 JOIN table2
        ON table1.Age_Group = table2.Age_Group
```

 # Returns the following table:

Age_Group	survive	didnotsurvive
adolescents	30	40
adults	213	335
children	40	29
seniors	7	19



As we can see on the chart, the survival rate is decreasing as the group progresses from children towards seniors. Children had the highest survival rate with 57.97%, followed by adolescents with 42.86%, adults 38.87%, and seniors make up the rear with 26.92% survival rate.

Although it can be seen that age is negatively correlated with survival, this data is not conclusive as 3 out of 4 groups had small sample sizes. For further analysis, binomial logistic regression was fitted to study the relationship between age and survival rate.

```
# Logistic regression model
lr_model = smf.logit(formula='Survived ~ Age', data=subset1).fit()
print(lr_model.summary())
```

Output:

```
                           Logit Regression Results
==============================================================================
Dep. Variable:                Survived   No. Observations:                  713
Model:                           Logit   Df Residuals:                      711
Method:                            MLE   Df Model:                            1
Date:                 Mon, 15 Aug 2022   Pseudo R-squ.:                0.004349
Time:                         14:26:47   Log-Likelihood:                -479.64
converged:                        True   LL-Null:                       -481.74
Covariance Type:            nonrobust   LLR p-value:                    0.04065
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0582      0.174     -0.335      0.737      -0.398       0.282
Age           -0.0108      0.005     -2.034      0.042      -0.021      -0.000
==============================================================================
```

Age is statistically significant with p-value = 0.042.

```
# Confidence interval
CI = lr_model.conf_int()
CI['Odd Ratio'] = lr_model.params
CI.columns = ['2.5%', '97.5%', 'Odd Ratio']
print(np.exp(CI))
```

Output:

	2.5%	97.5%	Odd Ratio
Intercept	0.671460	1.325661	0.943466
Age	0.978942	0.999607	0.989220

Threshold moving was used to find the optimum threshold for the logistic regression model.

```
# Threshold moving
predicted_values = lr_model.predict()
step_factor = 0.05
threshold = 0.1
model_accuracy = 0
while threshold <= 0.8:
    predicted_survived = np.zeros(predicted_values.shape)
    predicted_survived[predicted_values > threshold] = 1
    cm = confusion_matrix(survived, predicted_survived)
    accuracy = (cm[0, 0] + cm[1, 1]) / len(survived)
    print(f'Threshold {threshold} -- {accuracy}')
    if model_accuracy < accuracy:
        model_accuracy = accuracy
        thrsh_score = threshold
        best_cm = cm
    threshold += step_factor
print(f'---Optimum Threshold--- {thrsh_score} --Accuracy--
{model_accuracy}')
```

```
# Output:
Threshold 0.1 -- 0.4067321178120617
Threshold 0.15000000000000002 -- 0.4067321178120617
Threshold 0.2 -- 0.4067321178120617
Threshold 0.25 -- 0.4067321178120617
Threshold 0.3 -- 0.4067321178120617
Threshold 0.35 -- 0.4305750350631136
Threshold 0.39999999999999997 -- 0.48106591865357645
Threshold 0.44999999999999996 -- 0.6115007012622721
Threshold 0.49999999999999994 -- 0.5932678821879382
Threshold 0.5499999999999999 -- 0.5932678821879382
Threshold 0.6 -- 0.5932678821879382
Threshold 0.65 -- 0.5932678821879382
Threshold 0.7000000000000001 -- 0.5932678821879382
Threshold 0.7500000000000001 -- 0.5932678821879382
---Optimum Threshold--- 0.44999999999999996 --Accuracy--
0.6115007012622721
```

The optimum threshold is 0.45 with an accuracy of 61.15%. The confusion matrix of this logistic regression with threshold 0.45 is as follows:

```
# Confusion matrix
print(f'''Confusion matrix:
{best_cm}''')

# Sensitivity and specificity
sensitivity = best_cm[1, 1] / (best_cm[1, 0] + best_cm[1, 1])
specificity = best_cm[0, 0] / (best_cm[0, 0] + best_cm[0, 1])
print(f'''Sensitivity: {sensitivity}
Specificity: {specificity}
Accuracy: {model_accuracy}''')

# Output:
Confusion matrix:
[[394  29]
 [248  42]]
Sensitivity: 0.14482758620689656
Specificity: 0.9314420803782506
Accuracy: 0.6115007012622721
```

394 samples were identified as true negative, 29 samples as false positive, 248 as false negative, and 42 as true positive. From this confusion matrix, we can compute the sensitivity, specificity, and accuracy of the prediction model. This gave us a sensitivity of 14.48%, specificity of 93.14%, and overall accuracy of 61.15%. Our prediction model is excellent in predicting who would not survive in a Titanic crash, but very poor in detecting who would indeed survive.

5.0 Conclusion

Based on the results of all bivariate analyses, we could conclude that socioeconomic status, gender, and age are correlated with survival.

Gender is strongly correlated with survival, where females were more likely to survive than males, with 74.20% survival rate against 18.92%.

Whereas socioeconomic status is moderately correlated with survival. First class passengers had higher chance of surviving with rate of 63.26%, as compared with second class and third class passengers with 47.28% and 24.24% survival rate respectively.

On the other hand, age is weakly correlated with survival, where children had better chance of surviving out of all 4 age groups. Children had the highest survival rate with 57.97%, followed by adolescents with 42.86%, adults 38.87%, and seniors with 26.92% survival rate.

Surviving Titanic crash can also be predicted using binomial logistic regression model, with age as the predictor. This model, however, only had 61.15% accuracy with specificity of 93.14% and sensitivity of 14.48%, which would be very good in predicting people who would not survive, but not the otherwise.

Although these 3 variables were correlated with survival, this doesn't imply any causation relationship. There were other confounding and latent variables that influenced the survival but were not measured or unable to be measured, such as body habitus, stamina, ability to swim or stay afloat, and most importantly, luck.