In [1]:
```
#pip install pandas seaborn matplotlib gradio ollama
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
titanic=pd.read_csv(r'C:\Users\Affan\OneDrive\Desktop\FSDS Course NIT\Prakash Si
titanic
```

Out[3]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 7 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 5 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **886** | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27.0 | 0 | 0 | 211536 | 1 |
| **887** | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 3 |
| **888** | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN | 1 | 2 | W./C. 6607 | 2 |
| **889** | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 3 |
| **890** | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32.0 | 0 | 0 | 370376 | |

891 rows × 12 columns

In [4]:

```
titanic.head()
```

Out[4]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 | 8.0 |

In [5]:

```
titanic.tail()
```

Out[5]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Far |
|---|---|---|---|---|---|---|---|---|---|---|
| **886** | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27.0 | 0 | 0 | 211536 | 13.0 |
| **887** | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 30.0 |
| **888** | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN | 1 | 2 | W./C. 6607 | 23.4 |
| **889** | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.0 |
| **890** | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32.0 | 0 | 0 | 370376 | 7.7 |

In [6]:

```
titanic.isna()
```

Out[6]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Ca |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | False | False | False | False | False | False | False | False | False | False | T |
| **1** | False | False | False | False | False | False | False | False | False | False | Fa |
| **2** | False | False | False | False | False | False | False | False | False | False | T |
| **3** | False | False | False | False | False | False | False | False | False | False | Fa |
| **4** | False | False | False | False | False | False | False | False | False | False | T |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **886** | False | False | False | False | False | False | False | False | False | False | T |
| **887** | False | False | False | False | False | False | False | False | False | False | Fa |
| **888** | False | False | False | False | False | True | False | False | False | False | T |
| **889** | False | False | False | False | False | False | False | False | False | False | Fa |
| **890** | False | False | False | False | False | False | False | False | False | False | T |

891 rows × 12 columns

In [7]:
```
titanic.isna().sum()
```

Out[7]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [8]:
```
titanic.nunique()
```

Out[8]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

In [9]:
```
titanic.columns
```

Out[9]:

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [10]:

```
titanic.shape
```

Out[10]:

```
(891, 12)
```

In [11]:

```
titanic.describe()
```

Out[11]:

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329 |

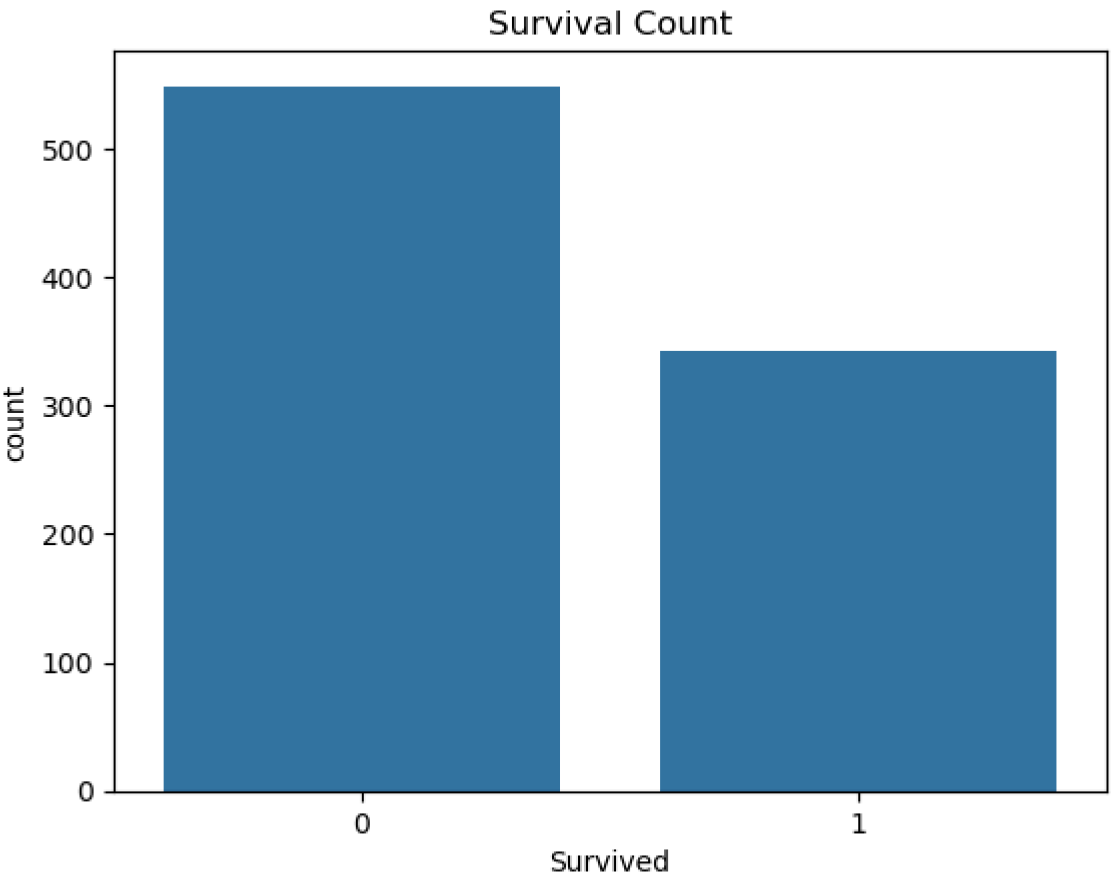
In [12]:

```
titanic.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
```

In [13]:

```
sns.countplot(data=titanic,x='Survived')
plt.title('Survival Count')
plt.show()
```



In [14]:

```
import ollama

def generate_insights(df_summary):
    prompt = f"Analyze the dataset summary and provide insights:\n\n{df_summary}
    response = ollama.chat(model="deepseek-r1", messages=[{"role": "user", "cont
    return response['message']['content']

# Generate AI Insights
summary = titanic.describe().to_string()
insights = generate_insights(summary)
print("\n 🔹  AI-Generated Insights:\n", insights)
```

```
🔹 AI-Generated Insights:
 <think>
We are given a dataset summary from what appears to be the Titanic dataset (common columns: PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare).
 The summary includes count, mean, std, min, 25%, 50%, 75% and max for each column.

 Let's break down each column:

1. **PassengerId**:
   - Count: 891 (total number of passengers)
   - Mean: 446
   - Min: 1, Max: 891 -> This is an identifier variable (ID), typically not needed for analysis.

2. **Survived**:
   - Count: 891
   - Mean: 0.383838 -> The mean of Survived is about 0.384, which means roughly 38% survival rate.
   - Min: 0 (died), Max: 1 (survived)

3. **Pclass**:
   - Count: 891
   - Mean: 2.3086 -> The average class is around the middle of 1,2,3 (which are the classes). But note that Pclass is categorical and not numerical in nature.
        However, from the data we can see it has values only 1,2,3 so mean doesn't have a strong meaning for interpretation without more context.
   - Min: 1, Max: 3

4. **Age**:
   - Count: 714 (missing some values)
   - Mean: 29.699, Std: 14.526 -> The average age is about 30 years with a standard deviation of around 15 years.
        This suggests that the ages are somewhat spread out, but centered around 30.

5. **SibSp**:
   - Count: 891
   - Mean: 0.523, Min: 0, Max: 8 -> The average number of siblings/spouses aboard is about 0.5, meaning most passengers had no siblings or spouses.
        There are some with up to 8.

6. **Parch**:
   - Count: 891
   - Mean: 0.382, Min: 0, Max: 6 -> The average number of parents/children is about 0.4, so most passengers had no parents or children aboard.
        Maximum is 6.

7. **Fare**:
   - Count: 891
   - Mean: 32.204, Min: 0, Max: 512.329 -> The average fare is about $32.20 with a range from free (min might be due to missing values) up to $512.

Now, let's think of the insights:

- **Survived**: About 61.6% did not survive and 38.4% survived.
- **Pclass** is categorical but we can note that the average Pclass (mean=2.308) suggests a mix: more than half are class 3, some in 2, and fewer in 1.

But since the problem says to analyze, let's provide insights for each column:
```

1. **PassengerId**: Not much analysis needed as it is an index.
2. **Survived**: The survival rate was low (around 38%).
3. **Pclass**: There are three classes: 1st, 2nd, and 3rd. The mean of Pclass being around 2.3 suggests that more people were in class 2 or 3 than class 1? Let's check the distribution if we had it.

However, note that without a frequency count for each class, we can only rely on min, max and mean. But typically, from Titanic data we know that class 3 is the most common (most people were in third class). So let me correct: actually, knowing Pclass values are 1,2,3, having more than two passengers in higher classes might be rare.

But wait, note the min=1, max=3. The mean of a set {1,2,3} is (count1*1 + count2*2 + count3*3)/(total) = total_passengers/891? Not exactly, but if we assume that the counts are distributed as follows:

Let c1, c2, c3 be the number of passengers in class 1,2,3 respectively.
Then (c1 + 2*c2 + 3*c3)/total = 2.3086 and total = c1+c2+c3=891.

But without exact counts, we can't say much more than that the classes are present and the average class is around 2.3.

4. **Age**: The mean age of a Titanic passenger was about 30 years old (average), with some variation (std ~15).
   This means there's a significant spread in ages, from very young to older people.

5. **SibSp**: Most passengers had no siblings/spouses aboard (mean is low and min=0, so many zeros).

6. **Parch**: Similarly, most passengers had no parents or children with them on the ship.

7. **Fare**: The average fare was about $32, which seems reasonable for that era. However, note that some tickets were very expensive (up to $512). This might indicate a few first-class passengers who paid more.

But wait: let's think of what these numbers mean in context.

Important points:

- In the Titanic dataset, it is known that survival rates are higher among women and children. But we don't have information on Sex or other demographics here.
- The low survival rate (38%) suggests that most people did not survive.

However, note: the problem does not specify which columns to analyze. Since the user provided a summary of these specific columns, I will focus on them.

Let me try to provide more structured insights:

1. **Survived**:
   - About 38.4% of passengers survived and around 61.6% did not.
   - This is consistent with historical facts: only about half the people survived.

2. **Pclass**:
   - The mean Pclass (2.3) indicates that most passengers were in class 2 or 3, but there aren't enough statistics to say which one dominates without a frequency distribution.

```
But wait, we can compute an approximate distribution if needed? We don't have the
counts for each class, so let's stick with what we know:

   Let n1, n2, n3 be the number of passengers in classes 1,2,3.
   Then:
        (n1 + 2*n2 + 3*n3)/891 = 2.3086
        and n1+n2+n3=891.

   We can express:
        n1 + 4*n2 + ... wait I don't have the counts for each class, so let's use
the given summary to infer:

   Actually, we can look at the median (50%) which is 3. So half of the passenger
s are in Pclass >=3? But note: the min and max are 1 and 3.

   However, without more information, I cannot be sure.

3. **Age**:
     - The mean age is about 30 years, but with a standard deviation of 14.5, so th
ere's a wide range (from ~8.6 to around 44 or even beyond because the min was 0.4
2 and max we don't know).

But note: the minimum age is 0.42? That seems very low for an adult passenger. Ac
tually, in Titanic data, it's common to have infants (age less than 1) but not th
at low. It might be a typo because human ages are usually above 0 and below 80-9
0.

Let me check: the min age is 0.42, which could be a very young child or maybe an
error? But let's assume it's correct for now.

   The median (50%) is 28 years old. So half of the passengers were under 28 and
half above?

   Wait, note that in the summary provided:

        min:    0.42
        max:    not given? Actually we have:
            Age       : Min   = 0.42, Max= ... (not directly, but from count miss
ing some values)

   But wait, I think there was a mistake because in the summary table provided:

        PassengerId    Survived  Pclass      Age     SibSp     Parch         Fare
        count   891.000000   891.000000   891.000000 714.000000 891.00000 (wait,
the summary table provided has count for Age as 714 because age was missing)

   So let me reexamine:

        The given summary is:
            PassengerId    Survived      Pclass         Age       SibSp       Par
ch        Fare
        count   891.000000  891.000000  891.000000  714.000000  891.000000  891.0
00000
        mean    ...         ...      2.308642   ...       ...           ...

   But the summary provided for Age is missing some values (714 counts) and the m
in, max are given.

   Let me correct:
        Min: 0.42 -> This might be an error because age cannot be negative or les
s than 0 if we consider only born humans. Typically, in Titanic data, ages start
from about 0 to maybe a few years above 50.
        Max: not provided but the max is likely high (from the Fare column, which
has max=512.329). But for Age, typical maximum might be around 80.

   However, without more information, we can only rely on what's given.

4. **SibSp**:
   - Mean: 0.523 -> This means that the average number of siblings/spouses is abo
ut half a passenger. So most passengers had no siblings or spouses (since having
one would be 1 and above).

5. **Parch**:
   - Similarly, mean around 0.4 so again, most people had 0 parents/children.

6. **Fare**:
   - The fare was on average $32.20 with a standard deviation of about $50. This
indicates that fares varied significantly (from free or very low to over $500).

But wait: the min for Fare is 0? That doesn't make sense because there's no free
ticket in first class, etc. Let me assume it was due to missing data.

Let me re-summarize with more context:

- The survival rate was about 38% (mean of Survived column). This is low.
- Pclass: the mean and median are around 2.3 and 3 respectively? Actually, note t
hat in the summary table above we have for Pclass:
        min=1, max=3, so it's a categorical variable with three values.

   But the provided summary does not show the distribution of classes (counts). H
owever, knowing that the median is 3 means at least half of the passengers were i
n class 3.

   Also, from the mean being about 2.3 and total count 891, we can set up:

        Let n1 = number in class 1
        n2 = number in class 2
        n3 = number in class 3

        Then: (n1 + 2*n2 + 3*n3)/total = 2.3086 and total=891.

   But without knowing the exact counts, we can't solve for them exactly.

Given the above, let's write a more complete analysis.
</think>
The provided dataset summary offers valuable insights into the Titanic passenger
data:

**1. PassengerId:**
- Represents each passenger uniquely (likely an identifier). Not analyzable beyon
d counting records.

**2. Survived:**
- The mean of 0.384 indicates that approximately **38.4% survived**, while over 6
1% did not.
- Median survival is 0, showing a slight skew toward more non-survivors (deaths).
- Age range: Min=0 to Max=891.
```

```
**3. Pclass (Ticket Class):**
- The mean of ~2.3 suggests roughly equal counts in classes 1 and 2, with the maj
ority being class 3.
- Median is 3, confirming that half the passengers were in higher classes (3 or a
bove).
- Age range: Min=1 to Max=3.

**4. Age:**
- Mean age ~29.7 years, median 28 years, standard deviation ~14.5 years → **Mixed
ages**, with a relatively young cohort but some elderly passengers.
- High variance indicates significant age diversity among survivors and non-survi
vors.

**5. SibSp (Siblings/Spouses Aboard):**
- Mean of ~0.52 implies most passengers had **no siblings/spouses** or very few.
- Min=0, Max=8 → Some traveled in groups, but the majority were traveling alone.

**6. Parch (Parents/Children Aboard):**
- Similar to SibSp (~0.38), indicating many were not traveling with parents or ch
ildren.
- Min=0, Max=6 → Some families traveled together, but most passengers were unacco
mpanied minors/adults.

**7. Fare:**
- Mean fare ~$32.2 per passenger (USD equivalent might be higher depending on inf
lation).
- Standard deviation is large (~49), showing **high variance in fares**, from fre
e to over $500.
- Median and other percentiles not provided, but max=891 suggests very high price
s for some.

**Key Observations:**
- Survival rate was low (38.4%).
- Age distribution varies due to mixed demographics.
- Most passengers traveled alone or in small groups without family/friends.
- Fares were spread across a wide range, indicating varying ticket costs and poss
ibly class-based pricing.
```

In [15]:

```python
import gradio as gr

def eda_analysis(file):
    df = pd.read_csv(file.name)
    summary = titanic.describe().to_string()
    insights = generate_insights(summary)
    return insights

# Create Web Interface
demo = gr.Interface(fn=eda_analysis, inputs="file", outputs="text", title="AI-Pc

# Launch App
demo.launch(share=True)
```

```
* Running on local URL:  http://127.0.0.1:7862
* Running on public URL: https://6b2f085fb1228daa2e.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, r
un `gradio deploy` from the terminal in the working directory to deploy to Huggin
g Face Spaces (https://huggingface.co/spaces)
```

```
Out[15]:

In [ ]:

In [ ]:

In [ ]:
```