

# Untitled

July 3, 2025

```
[3]: # Step 1: Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Step 2: Load the Titanic dataset
df = pd.read_csv("train.csv.csv") # make sure the file is uploaded and named
↳ correctly

# Step 3: Preview the data
df.head()
```

```
[3]:
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cummings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

```
[5]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Load dataset
df = pd.read_csv("train.csv.csv")

# Display first few rows
df.head()
```

```
[5]:
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

```
[6]: # Basic info
df.info()

# Statistical summary
df.describe()

# Missing values
df.isnull().sum()
```

```
<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

```

```

[6]: 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

```

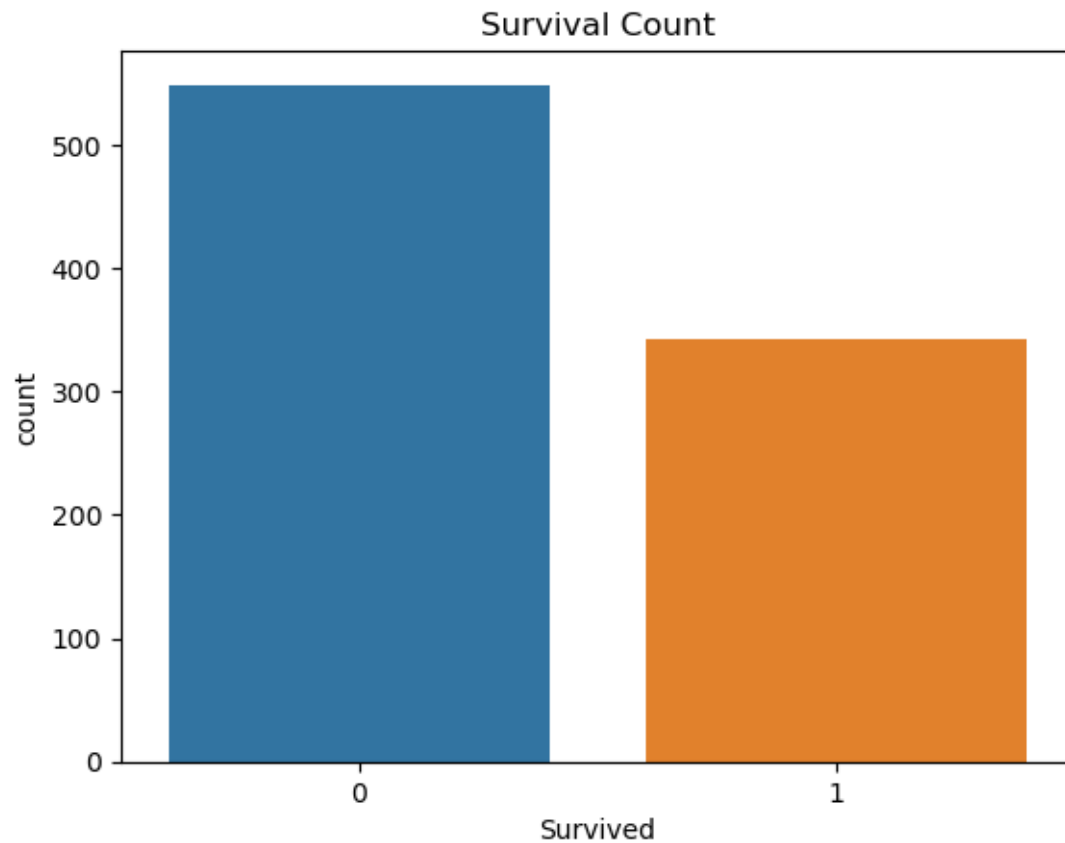
```

[7]: # Countplot for 'Survived'
     sns.countplot(x='Survived', data=df)
     plt.title('Survival Count')
     plt.show()

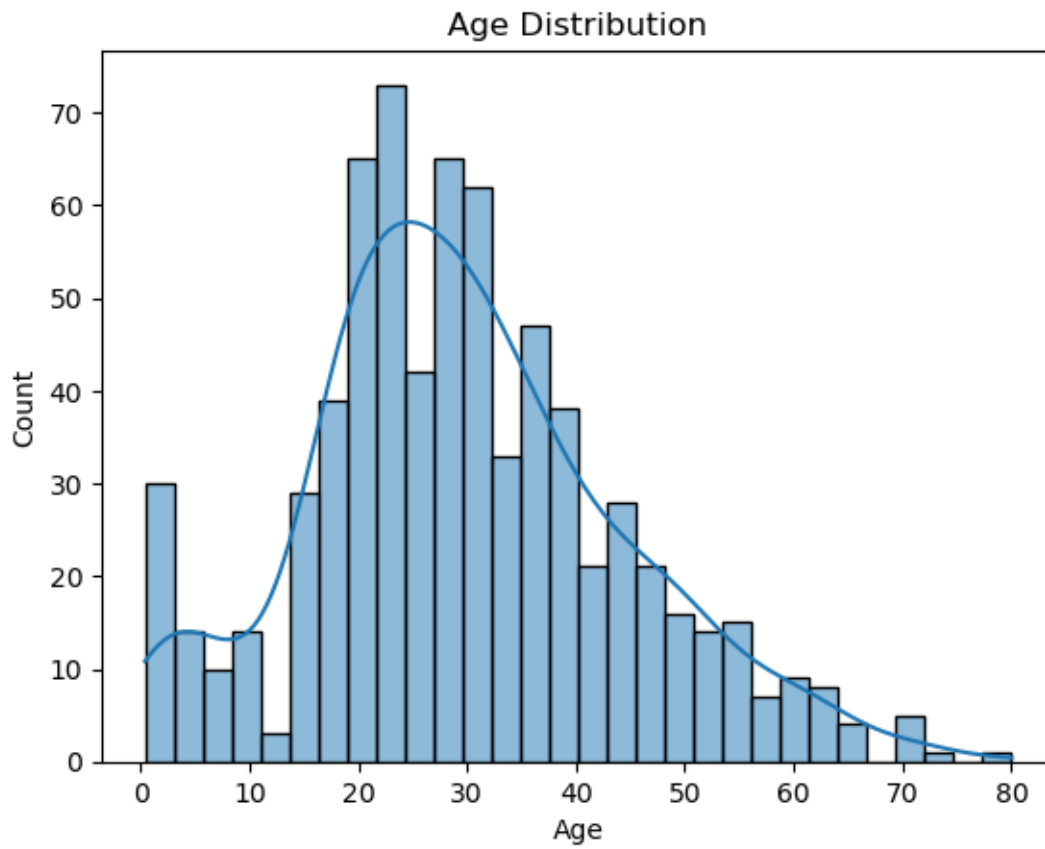
     # Histogram for 'Age'
     sns.histplot(df['Age'].dropna(), bins=30, kde=True)
     plt.title('Age Distribution')
     plt.show()

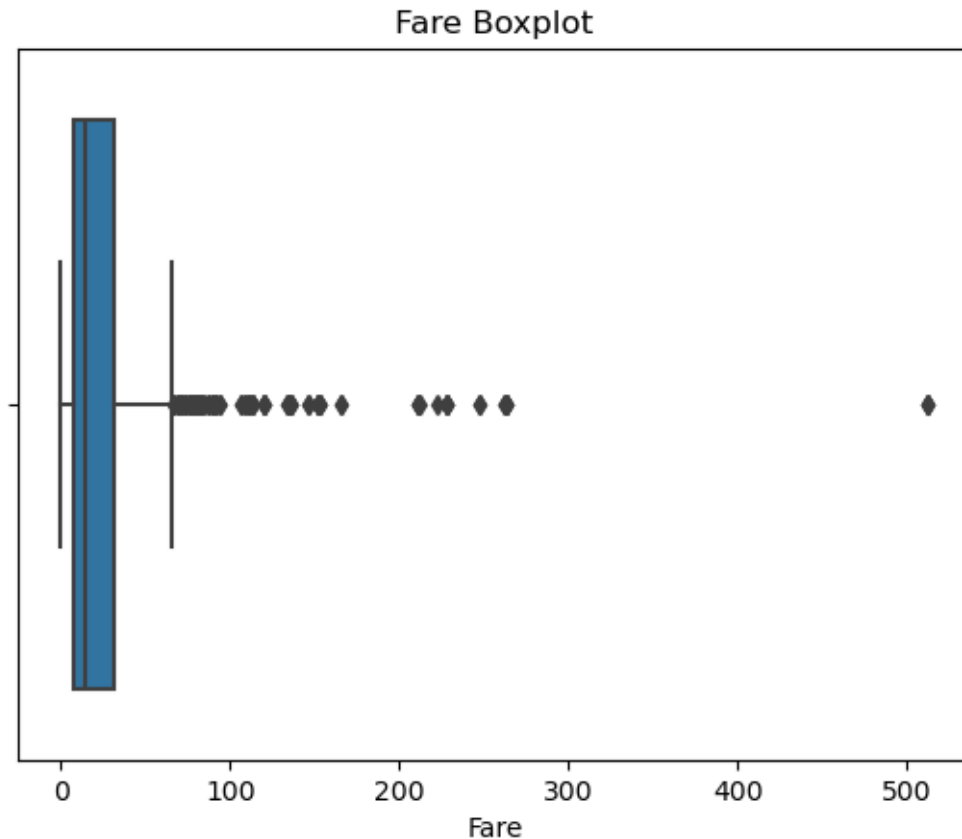
     # Boxplot for 'Fare'
     sns.boxplot(x=df['Fare'])
     plt.title('Fare Boxplot')
     plt.show()

```



```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is  
deprecated and will be removed in a future version. Convert inf values to NaN  
before operating instead.  
    with pd.option_context('mode.use_inf_as_na', True):
```





```
[17]: # Plot: Survival vs Passenger Class
import seaborn as sns
import matplotlib.pyplot as plt

# Create the plot
sns.countplot(x='Pclass', hue='Survived', data=df)

# Add titles and labels
plt.title('Passenger Class vs Survival')
plt.xlabel('Passenger Class')
plt.ylabel('Number of Passengers')
plt.legend(title='Survived', labels=['No', 'Yes'])

# Show the plot
plt.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[17], line 6
      3 import matplotlib.pyplot as plt
```

```

5 # Create the plot
----> 6 sns.countplot(x='Pclass', hue='Survived', data=df)
8 # Add titles and labels
9 plt.title('Passenger Class vs Survival')

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:2955, in countplot(data, x, y, hue, order, hue\_order, orient, color, palette, saturation, width, dodge, ax, \*\*kwargs)

```

2952 if ax is None:
2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
2956 return ax

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:1587, in \_BarPlotter.plot(self, ax, bar\_kws)

```

1585 """Make the plot."""
1586 self.draw_bars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
1588 if self.orient == "h":
1589     ax.invert_yaxis()

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:767, in \_CategoricalPlotter.annotate\_axes(self, ax)

```

764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/matplotlib/axes/\_axes.py:322, in Axes.legend(self, \*args, \*\*kwargs)

```

204 @_docstring.dedent_interpd
205 def legend(self, *args, **kwargs):
206     """
207     Place a legend on the Axes.
208
209     (...)
210
211     .. plot:: gallery/text_labels_and_annotations/legend.py
212     """
--> 322     handles, labels, kwargs =
    mlegend._parse_legend_args([self], *args, **kwargs)
323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
324     self.legend_.remove_method = self._remove_legend

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/matplotlib/legend.py:1361, in \_parse\_legend\_args(axes, handles, labels, \*args, \*\*kwargs)

```

1357 handles = [handle for handle, label
1358             in zip(_get_legend_handles(axes, handlers), labels)]
1360 elif len(args) == 0: # 0 args: automatically detect labels and handles

```

```

-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
    1362     if not handles:
    1363         log.warning(
    1364             "No artists with labels found to put in legend. Note that
    1365             "artists whose label start with an underscore are ignored "
    1366             "when legend() is called with no argument.")

```

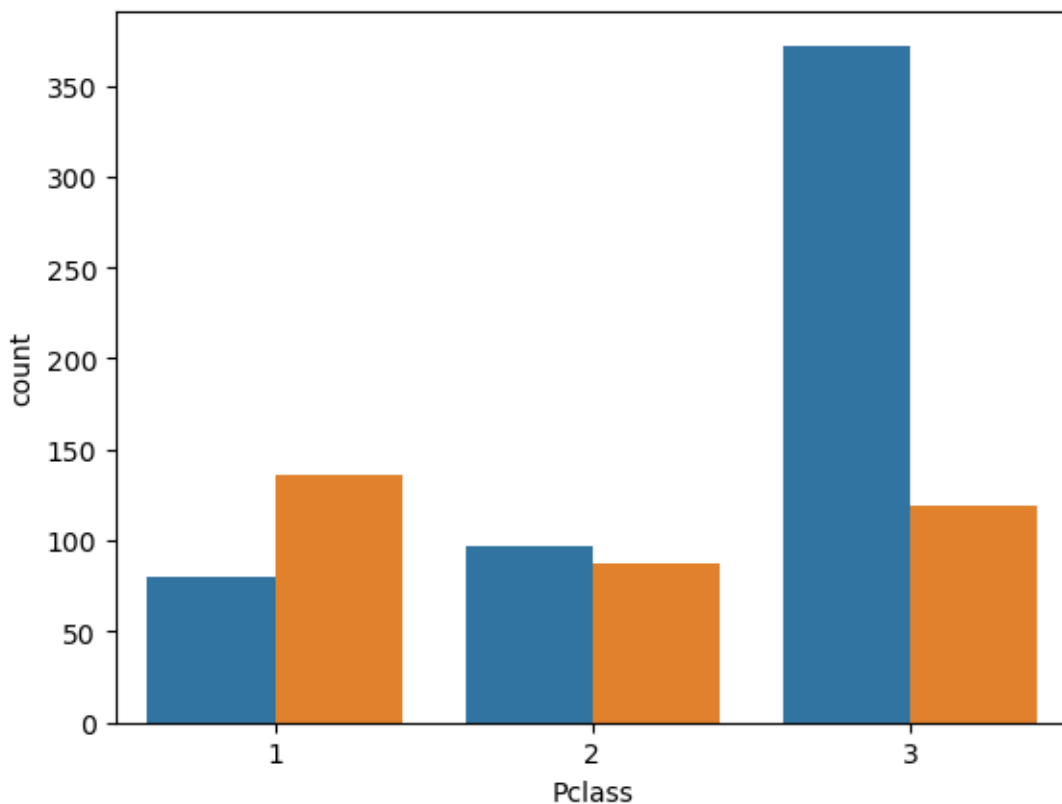
File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/  
↳matplotlib/legend.py:1291, in \_get\_legend\_handles\_labels(axs,  
↳legend\_handler\_map)

```

    1289 for handle in _get_legend_handles(axs, legend_handler_map):
    1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
    1292         handles.append(handle)
    1293         labels.append(label)

```

AttributeError: 'numpy.int64' object has no attribute 'startswith'



```

[18]: print(df.head())
      print(df.columns)

```



	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

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

```
[19]: import pandas as pd

# Reload dataset
df = pd.read_csv("train.csv.csv") # or train.csv if renamed

# Confirm load
df.head()
```

```
[19]: PassengerId  Survived  Pclass  \
0            1         0         3
1            2         1         1
2            3         1         3
3            4         1         1
4            5         0         3
```

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

Parch	Ticket	Fare	Cabin	Embarked
-------	--------	------	-------	----------

0	0	A/5	21171	7.2500	NaN	S
1	0	PC	17599	71.2833	C85	C
2	0	STON/O2.	3101282	7.9250	NaN	S
3	0		113803	53.1000	C123	S
4	0		373450	8.0500	NaN	S

```
[20]: import seaborn as sns
import matplotlib.pyplot as plt

# Confirm df exists before plotting
if 'df' in globals():
    sns.countplot(x='Pclass', hue='Survived', data=df)
    plt.title('Survival by Passenger Class')
    plt.xlabel('Passenger Class')
    plt.ylabel('Number of Passengers')
    plt.legend(title='Survived', labels=['No', 'Yes'])
    plt.show()
else:
    print("DataFrame 'df' is not loaded. Please load the dataset first.")
```

-----  
AttributeError Traceback (most recent call last)

Cell In[20], line 6

```
4 # Confirm df exists before plotting
5 if 'df' in globals():
----> 6     sns.countplot(x='Pclass', hue='Survived', data=df)
      7     plt.title('Survival by Passenger Class')
      8     plt.xlabel('Passenger Class')
```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:2955, in countplot(data, x, y, hue, order, hue\_order, orient, color, palette, saturation, width, dodge, ax, \*\*kwargs)

```
2952 if ax is None:
2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
2956 return ax
```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:1587, in \_BarPlotter.plot(self, ax, bar\_kws)

```
1585 """Make the plot."""
1586 self.drawBars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
1588 if self.orient == "h":
1589     ax.invert_yaxis()
```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/categorical.py:767, in \_CategoricalPlotter.annotate\_axes(self, ax)

```

764     ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/

```

↳matplotlib/axes/_axes.py:322, in Axes.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
    (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
    321     """
--> 322     handles, labels, kwargs = _
↳mlegend._parse_legend_args([self], *args, **kwargs)
    323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    324     self.legend_.remove_method = self._remove_legend

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/

```

↳matplotlib/legend.py:1361, in _parse_legend_args(axs, handles, labels, *args,
↳**kwargs)
    1357     handles = [handle for handle, label
    1358                 in zip(_get_legend_handles(axs, handlers), labels)]
    1360 elif len(args) == 0: # 0 args: automatically detect labels and handles
-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
    1362     if not handles:
    1363         log.warning(
    1364             "No artists with labels found to put in legend. Note that
    1365             "artists whose label start with an underscore are ignored "
    1366             "when legend() is called with no argument.")

```

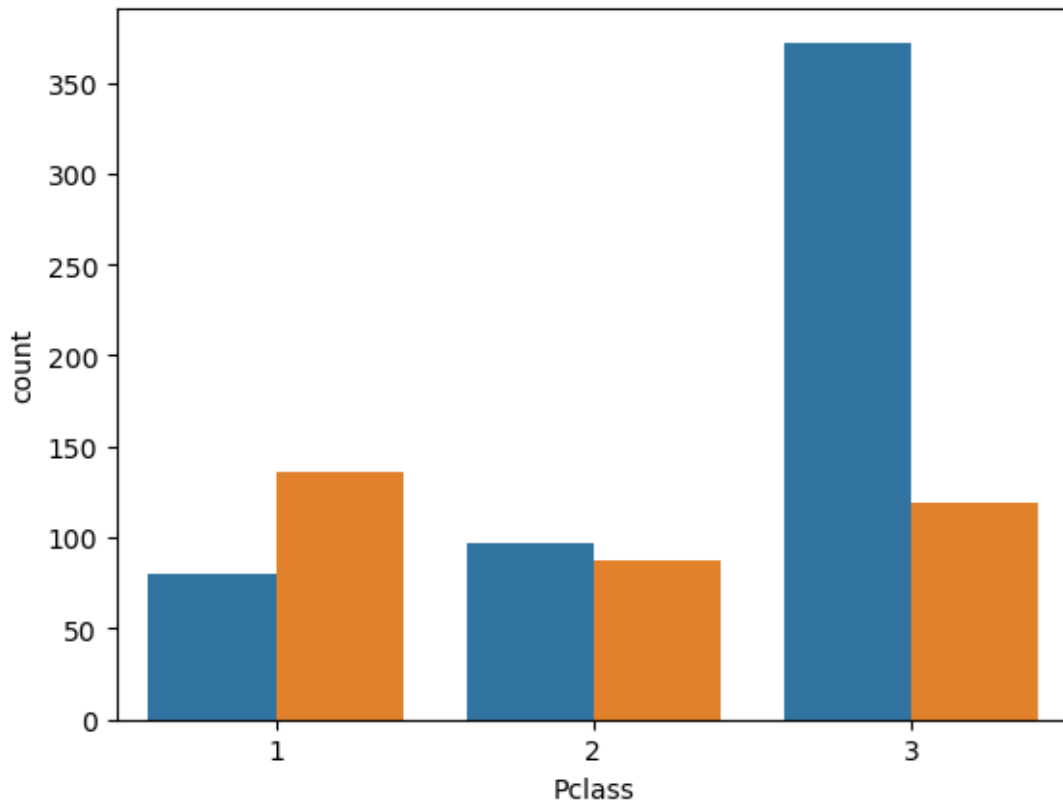
File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/

```

↳matplotlib/legend.py:1291, in _get_legend_handles_labels(axs,
↳legend_handler_map)
    1289 for handle in _get_legend_handles(axs, legend_handler_map):
    1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
    1292         handles.append(handle)
    1293         labels.append(label)

```

AttributeError: 'numpy.int64' object has no attribute 'startswith'



```
[21]: # Check actual column names
print(df.columns)

# Plot with exact column names
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Passenger Count')
plt.legend(title='Survived', labels=['No', 'Yes'])
plt.show()
```

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

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[21], line 5
      2 print(df.columns)
      4 # Plot with exact column names
----> 5 sns.countplot(x='Pclass', hue='Survived', data=df)
```

```

6 plt.title('Survival by Passenger Class')
7 plt.xlabel('Passenger Class')

```

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
↳ categorical.py:2955, in countplot(data, x, y, hue, order, hue_order, orient,
↳ color, palette, saturation, width, dodge, ax, **kwargs)
    2952 if ax is None:
    2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
    2956 return ax

```

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
↳ categorical.py:1587, in _BarPlotter.plot(self, ax, bar_kws)
    1585 """Make the plot."""
    1586 self.drawBars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
    1588 if self.orient == "h":
    1589     ax.invert_yaxis()

```

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
↳ categorical.py:767, in _CategoricalPlotter.annotate_axes(self, ax)
    764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
    766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

```

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/
↳ matplotlib/axes/_axes.py:322, in Axes.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
    (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
    321     """
--> 322     handles, labels, kwargs =
↳ mlegend._parse_legend_args([self], *args, **kwargs)
    323     self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
    324     self.legend_.remove_method = self._remove_legend

```

```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/
↳ matplotlib/legend.py:1361, in _parse_legend_args(axs, handles, labels, *args,
↳ **kwargs)
    1357 handles = [handle for handle, label
    1358              in zip(_get_legend_handles(axs, handlers), labels)]
    1360 elif len(args) == 0: # 0 args: automatically detect labels and handles
-> 1361     handles, labels = _get_legend_handles_labels(axs, handlers)
    1362     if not handles:

```

```

1363         log.warning(
1364             "No artists with labels found to put in legend. Note that
1365             "artists whose label start with an underscore are ignored "
1366             "when legend() is called with no argument.")

```

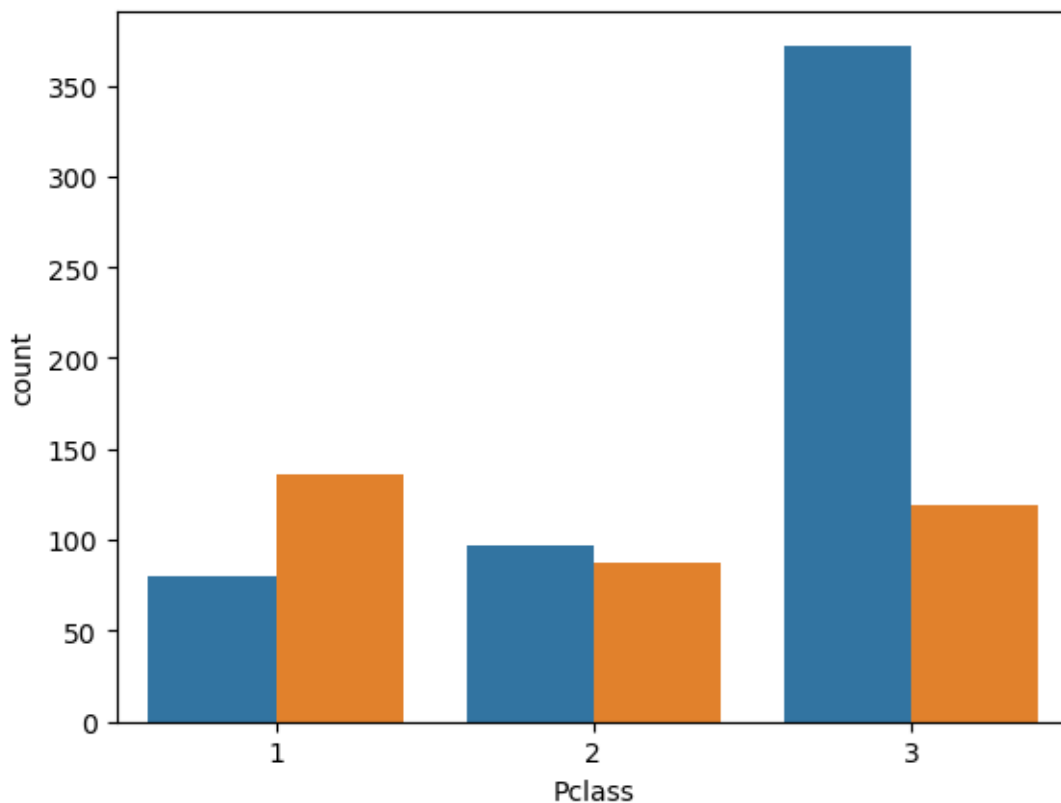
File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/  
↳matplotlib/legend.py:1291, in \_get\_legend\_handles\_labels(axs,   
↳legend\_handler\_map)

```

1289 for handle in _get_legend_handles(axs, legend_handler_map):
1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
1292         handles.append(handle)
1293         labels.append(label)

```

AttributeError: 'numpy.int64' object has no attribute 'startswith'



```

[22]: import seaborn as sns
import matplotlib.pyplot as plt

# Plot: Survival by Passenger Class
plt.figure(figsize=(6, 4))

```

```
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Passenger Count')
plt.legend(title='Survived') # remove labels param to avoid error
plt.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[22], line 6
      4 # Plot: Survival by Passenger Class
      5 plt.figure(figsize=(6, 4))
----> 6 sns.countplot(x='Pclass', hue='Survived', data=df)
      7 plt.title('Survival by Passenger Class')
      8 plt.xlabel('Passenger Class')

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
categorical.py:2955, in countplot(data, x, y, hue, order, hue_order, orient,
color, palette, saturation, width, dodge, ax, **kwargs)
    2952 if ax is None:
    2953     ax = plt.gca()
-> 2955 plotter.plot(ax, kwargs)
    2956 return ax

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
categorical.py:1587, in _BarPlotter.plot(self, ax, bar_kws)
    1585 """Make the plot."""
    1586 self.draw_bars(ax, bar_kws)
-> 1587 self.annotate_axes(ax)
    1588 if self.orient == "h":
    1589     ax.invert_yaxis()

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/seaborn/
categorical.py:767, in _CategoricalPlotter.annotate_axes(self, ax)
    764 ax.set_ylim(-.5, len(self.plot_data) - .5, auto=None)
    766 if self.hue_names is not None:
--> 767     ax.legend(loc="best", title=self.hue_title)

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/
matplotlib/axes/_axes.py:322, in Axes.legend(self, *args, **kwargs)
    204 @_docstring.dedent_interpd
    205 def legend(self, *args, **kwargs):
    206     """
    207     Place a legend on the Axes.
    208
    (...)
    320     .. plot:: gallery/text_labels_and_annotations/legend.py
```

```

321         """
--> 322         handles, labels, kwargs = mlegend._parse_legend_args([self], *args, **kwargs)
323         self.legend_ = mlegend.Legend(self, handles, labels, **kwargs)
324         self.legend_.remove_method = self._remove_legend

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/
↳matplotlib/legend.py:1361, in _parse_legend_args(axs, handles, labels, *args,
↳**kwargs)
1357         handles = [handle for handle, label
1358                     in zip(_get_legend_handles(axs, handlers), labels)]
1360 elif len(args) == 0: # 0 args: automatically detect labels and handles
-> 1361         handles, labels = _get_legend_handles_labels(axs, handlers)
1362         if not handles:
1363             log.warning(
1364                 "No artists with labels found to put in legend. Note that
1365                 "artists whose label start with an underscore are ignored "
1366                 "when legend() is called with no argument.")

```

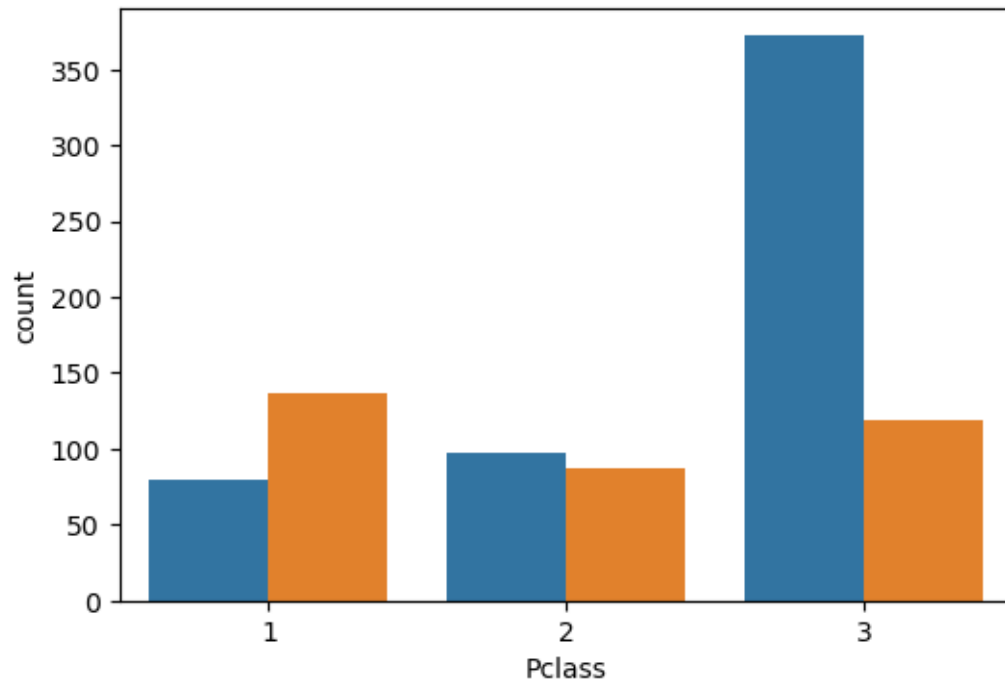
```

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/
↳matplotlib/legend.py:1291, in _get_legend_handles_labels(axs,
↳legend_handler_map)
1289 for handle in _get_legend_handles(axs, legend_handler_map):
1290     label = handle.get_label()
-> 1291     if label and not label.startswith('_'):
1292         handles.append(handle)
1293         labels.append(label)

```

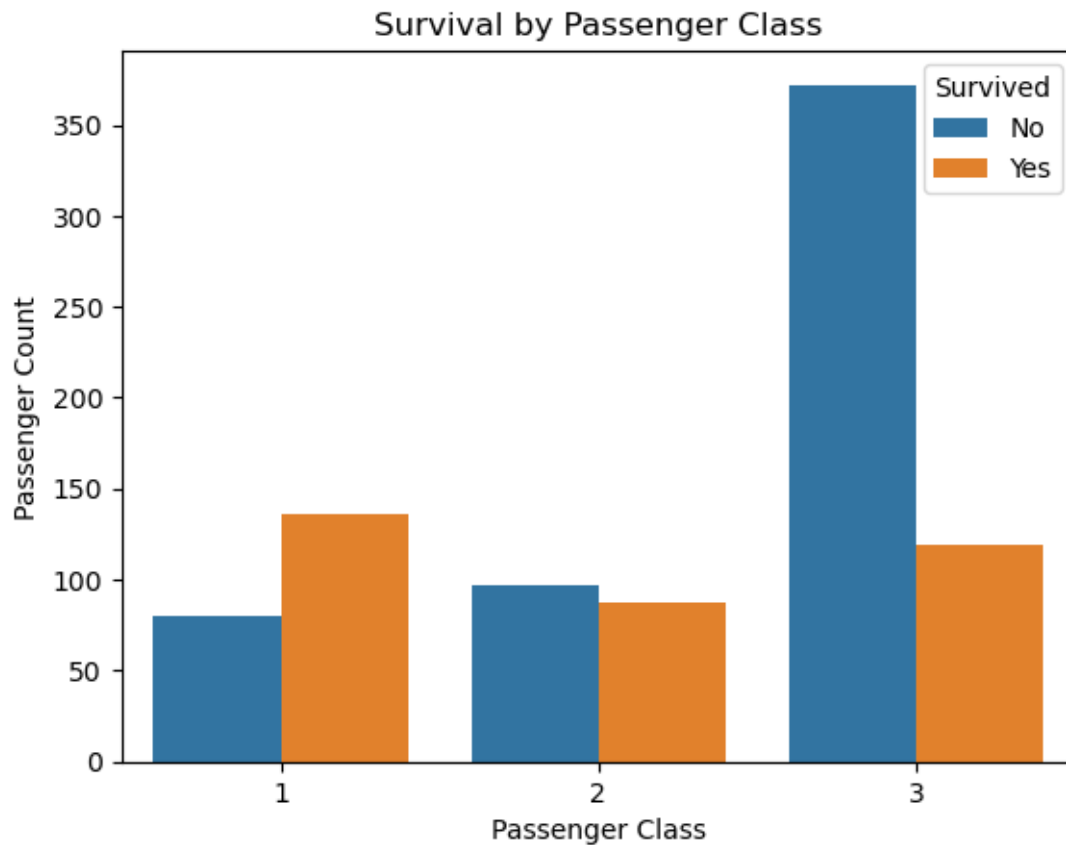
AttributeError: 'numpy.int64' object has no attribute 'startswith'





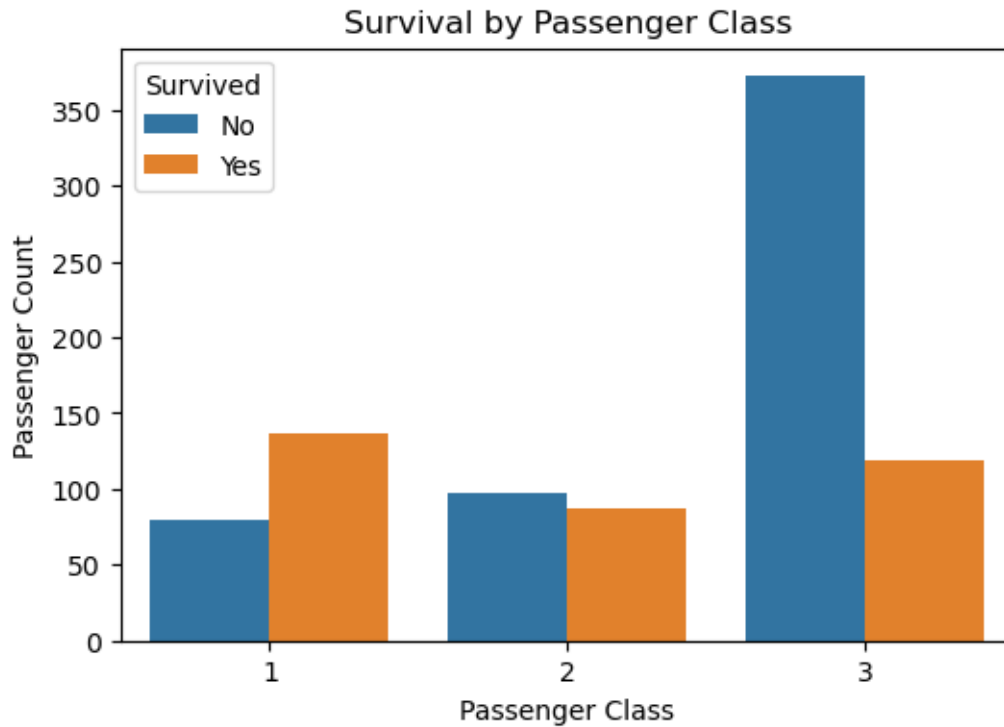
```
[23]: # Replace 0 and 1 with 'No' and 'Yes'
df['Survived'] = df['Survived'].map({0: 'No', 1: 'Yes'})

# Now plot with strings
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Passenger Count')
plt.legend(title='Survived')
plt.show()
```



```
[24]: import seaborn as sns
import matplotlib.pyplot as plt

# Final Fixed Code (No custom labels)
plt.figure(figsize=(6, 4))
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Passenger Count')
plt.legend(title='Survived') # Don't pass custom labels!
plt.show()
```



```
[25]: import seaborn as sns
import matplotlib.pyplot as plt

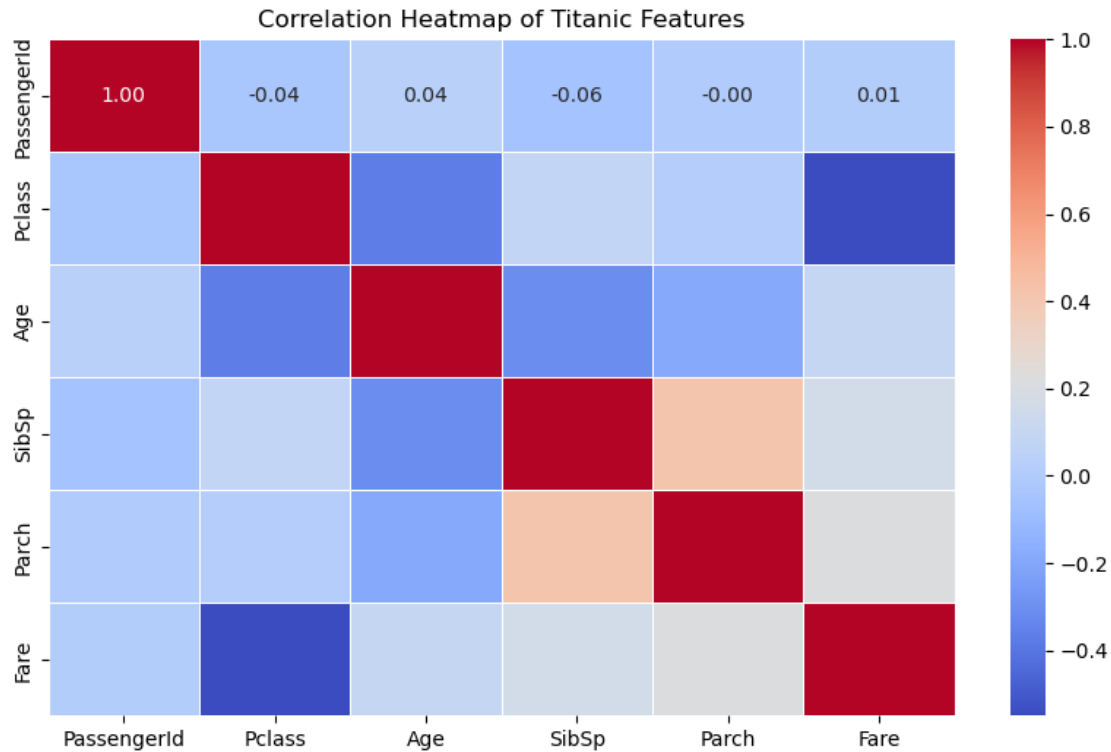
# Compute correlation matrix (only for numeric columns)
corr = df.corr(numeric_only=True)

# Set up the figure
plt.figure(figsize=(10, 6))

# Draw the heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm', linewidths=0.5, fmt=".2f")

# Add title
plt.title("Correlation Heatmap of Titanic Features")

# Show the plot
plt.show()
```



```
[26]: import seaborn as sns
import matplotlib.pyplot as plt

# Subset data for pairplot
pairplot_data = df[['Survived', 'Pclass', 'Age', 'Fare']].dropna()

# Convert Survived to string for better coloring
pairplot_data['Survived'] = pairplot_data['Survived'].map({0: 'No', 1: 'Yes'})

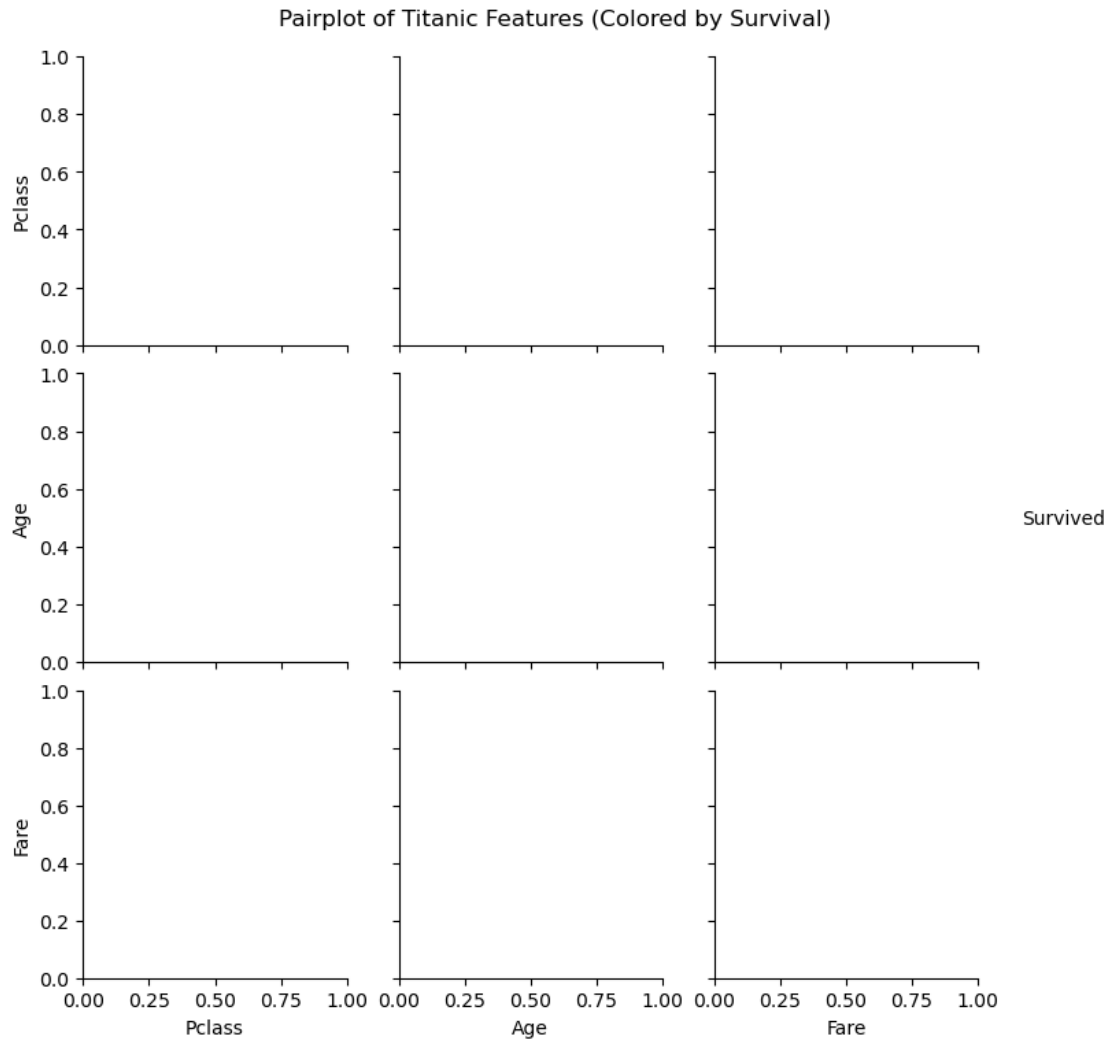
# Plot
sns.pairplot(pairplot_data, hue='Survived', palette='Set1')
plt.suptitle("Pairplot of Titanic Features (Colored by Survival)", y=1.02)
plt.show()
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```

/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)

```



```
[27]: print(pairplot_data['Survived'].unique())
      print(pairplot_data['Survived'].isnull().sum())
```

```
[nan]
```

```
714
```

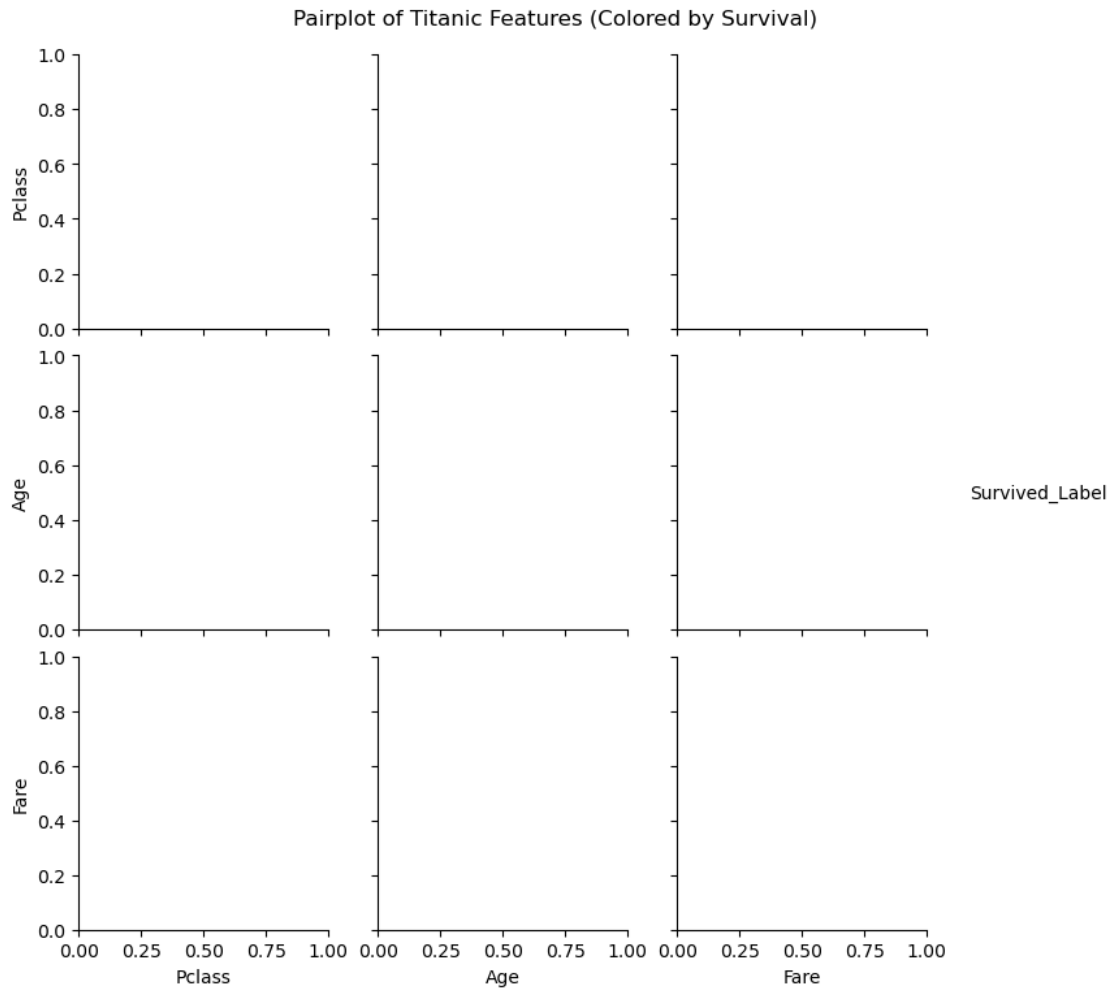
```
[28]: # Step 1: Clean and convert BEFORE dropping NaN
      df['Survived_Label'] = df['Survived'].map({0: 'No', 1: 'Yes'})

      # Step 2: Create pairplot dataset and drop NaNs
      pairplot_data = df[['Survived_Label', 'Pclass', 'Age', 'Fare']].dropna().copy()

      # Step 3: Now plot
      import seaborn as sns
      import matplotlib.pyplot as plt
```

```
sns.pairplot(pairplot_data, hue='Survived_Label', palette='Set1')
plt.suptitle("Pairplot of Titanic Features (Colored by Survival)", y=1.02)
plt.show()
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=vector, **plot_kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
```



```
[29]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Create a copy of the original dataframe
df_copy = df.copy()

# Step 2: Map the 'Survived' column to readable labels before dropping any rows
df_copy['Survived_Label'] = df_copy['Survived'].map({0: 'No', 1: 'Yes'})

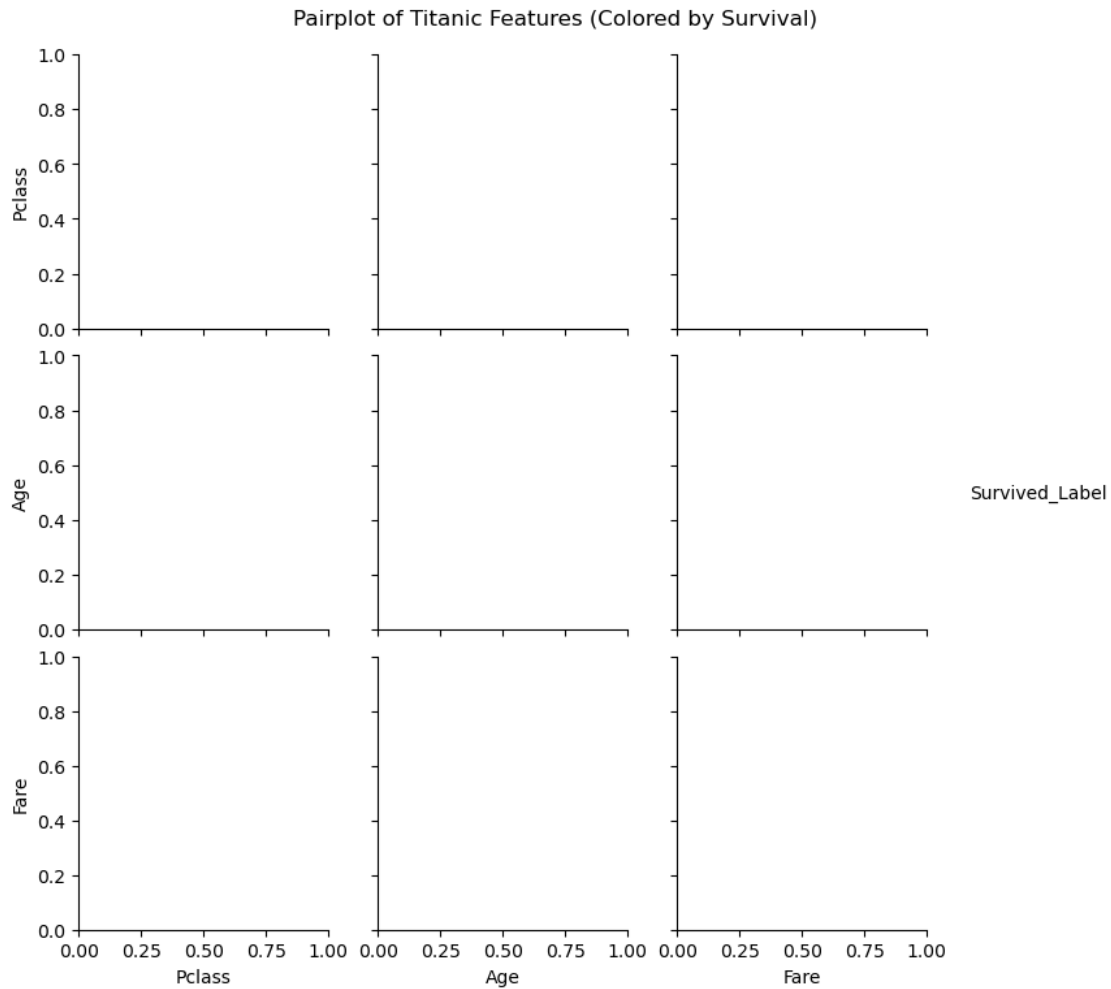
# Step 3: Drop rows with missing values from the selected columns
pairplot_data = df_copy[['Survived_Label', 'Pclass', 'Age', 'Fare']].dropna()

# Step 4: Plot the pairplot
sns.pairplot(data=pairplot_data, hue='Survived_Label', palette='Set1')
plt.suptitle("Pairplot of Titanic Features (Colored by Survival)", y=1.02)
```



```
plt.show()
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=vector, **plot_kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=vector, **plot_kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1507: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=vector, **plot_kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=x, y=y, **kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=x, y=y, **kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=x, y=y, **kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=x, y=y, **kwargs)  
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/axisgrid.py:1609: UserWarning: Ignoring `palette` because no  
`hue` variable has been assigned.  
    func(x=x, y=y, **kwargs)
```



```
[30]: # Full reset to make sure nothing's wrong
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Reload original dataset
df = pd.read_csv("train.csv.csv") # or 'train.csv' if renamed

# Step 1: Map Survived to readable labels
df['Survived_Label'] = df['Survived'].map({0: 'No', 1: 'Yes'})

# Step 2: Drop missing values only from needed columns
pairplot_data = df[['Survived_Label', 'Pclass', 'Age', 'Fare']].dropna()

# Step 3: Check it's clean
print(pairplot_data['Survived_Label'].unique())
```

```
print(pairplot_data.isnull().sum())
```

```
['No' 'Yes']  
Survived_Label    0  
Pclass            0  
Age              0  
Fare             0  
dtype: int64
```

```
[31]: # Final working pairplot  
sns.pairplot(data=pairplot_data, hue='Survived_Label', palette='Set1')  
plt.suptitle("Pairplot of Titanic Features (Colored by Survival)", y=1.02)  
plt.show()
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is  
deprecated and will be removed in a future version. Convert inf values to NaN  
before operating instead.
```

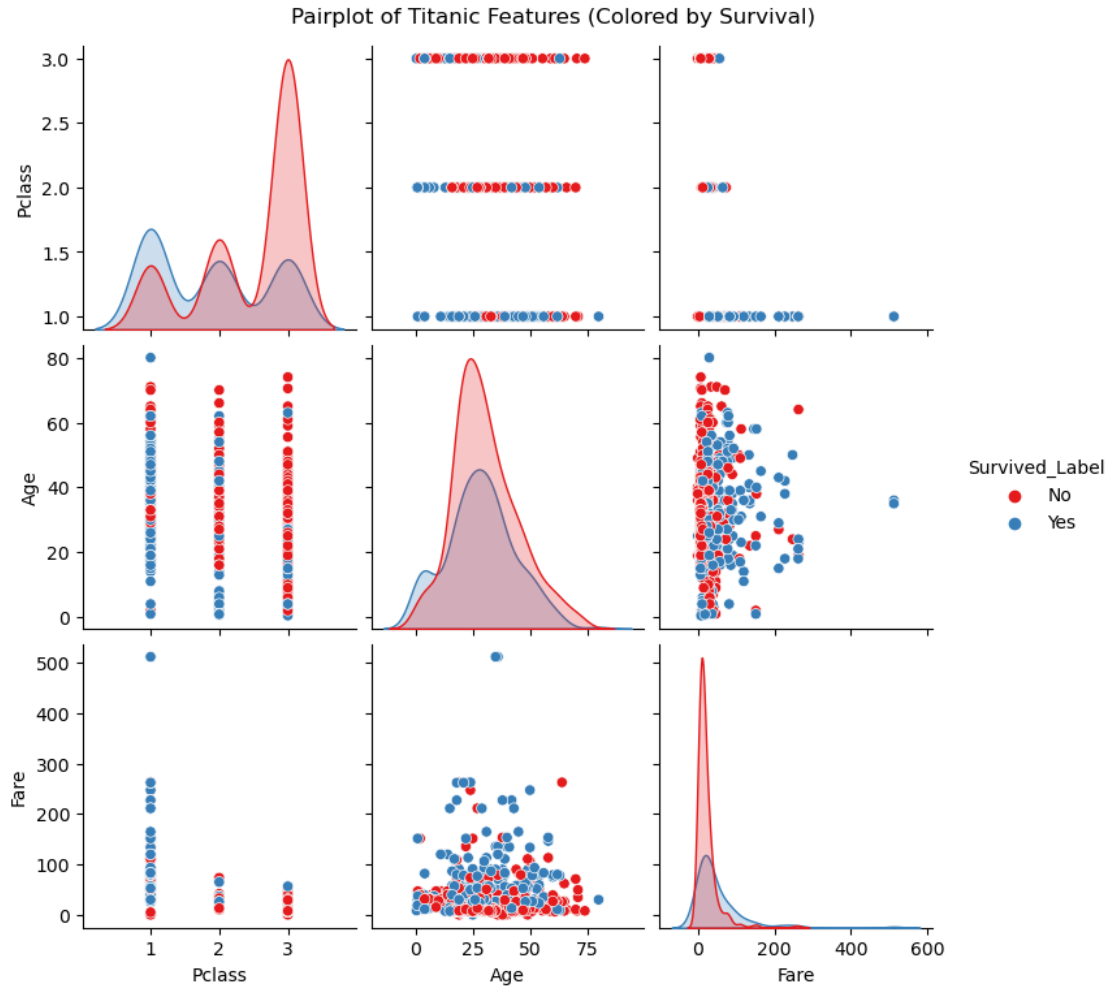
```
    with pd.option_context('mode.use_inf_as_na', True):
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is  
deprecated and will be removed in a future version. Convert inf values to NaN  
before operating instead.
```

```
    with pd.option_context('mode.use_inf_as_na', True):
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-  
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is  
deprecated and will be removed in a future version. Convert inf values to NaN  
before operating instead.
```

```
    with pd.option_context('mode.use_inf_as_na', True):
```



[ ]:

## 0.1 Summary of Key Insights from Titanic EDA

### 1. Gender and Survival:

- Female passengers had a significantly higher survival rate than males.
- Most women survived, especially from 1st and 2nd class.

### 2. Passenger Class and Survival:

- 1st class passengers had the highest chance of survival.
- 3rd class passengers had the lowest survival rate overall.

### 3. Age Distribution:

- Children and younger passengers had a better survival chance.
- Some elderly passengers also survived, but in lower numbers.

### 4. Fare and Survival:

- Passengers who paid higher fares were more likely to survive.
- Fare is positively correlated with survival (related to class).

### 5. Heatmap Correlation:

- Fare and Survived showed positive correlation.
- Pclass and Survived were negatively correlated.
- Age and Survived had a weaker but visible relationship.

### 6. Missing Data:

- Age and Cabin had missing values.
- Missing Age rows were dropped during visualizations.

This analysis helps us understand how social status (class), gender, and age influenced survival in the Titanic disaster.

[32]: File → Download as → HTML (.html)

```
Cell In[32], line 1
  File → Download as → HTML (.html)
  ~
SyntaxError: invalid character '→' (U+2192)
```

[ ]: