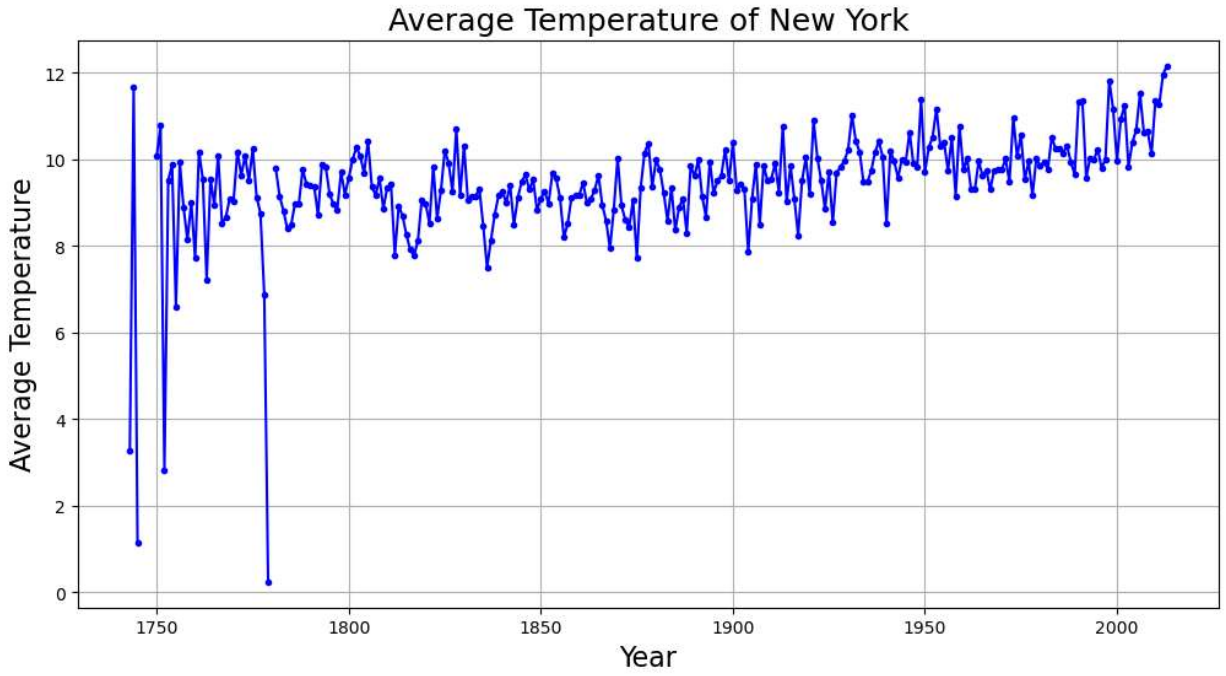
In [7]:
```
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv('GlobalLandTemperaturesByCity.csv')
df
```

Out[7]:

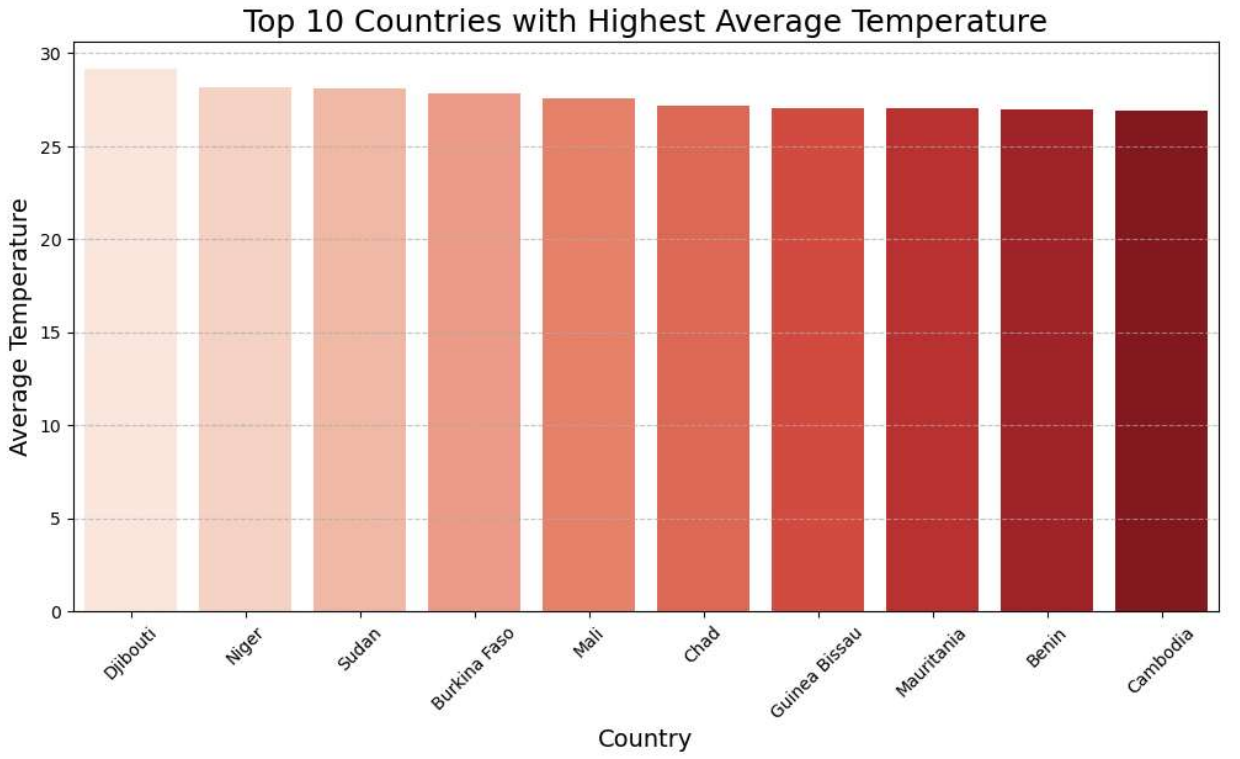
	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1743-11-01	6.068	1.737	Århus	Denmark	57.05N	10.33E
1	1743-12-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
2	1744-01-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
3	1744-02-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
4	1744-03-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
...	...	...	...	...	...	...	...
8599207	2013-05-01	11.464	0.236	Zwolle	Netherlands	52.24N	5.26E
8599208	2013-06-01	15.043	0.261	Zwolle	Netherlands	52.24N	5.26E
8599209	2013-07-01	18.775	0.193	Zwolle	Netherlands	52.24N	5.26E
8599210	2013-08-01	18.025	0.298	Zwolle	Netherlands	52.24N	5.26E
8599211	2013-09-01	NaN	NaN	Zwolle	Netherlands	52.24N	5.26E

8599212 rows × 7 columns

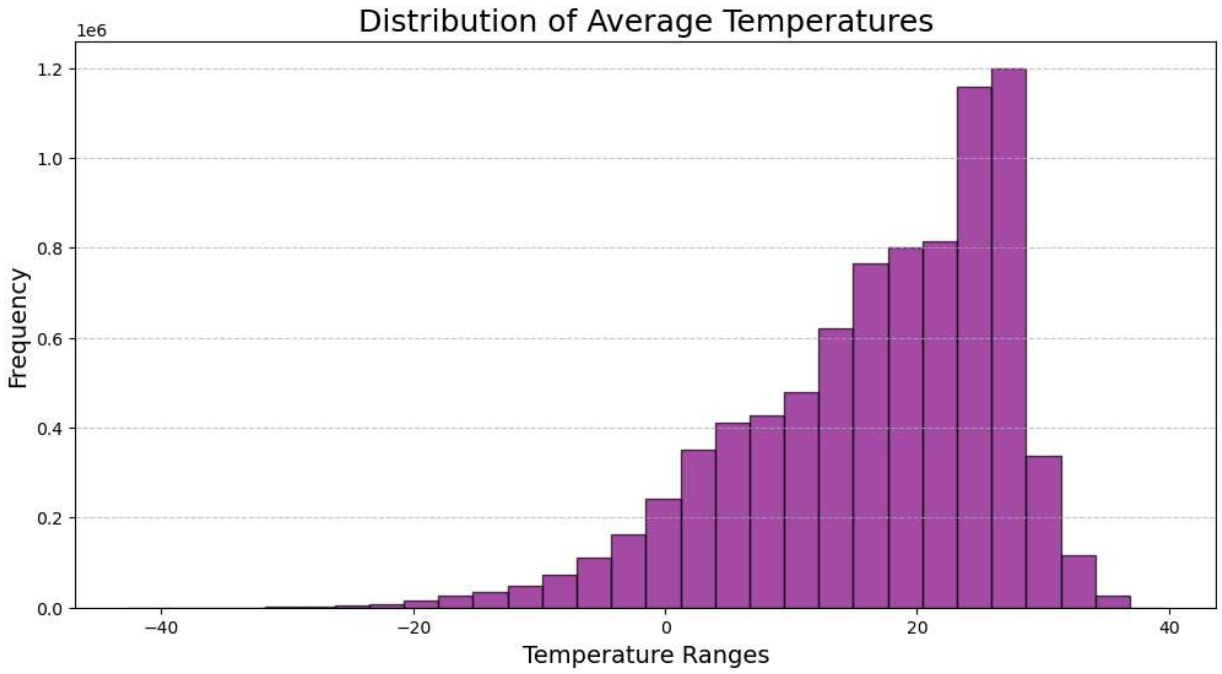
In [13]:
```
# Filter data for New York City
city_data = df[df['City'] == 'New York'].copy()

# Convert 'dt' to datetime
city_data['dt'] = pd.to_datetime(city_data['dt'], errors='coerce')

# Extract the year
city_data['Year'] = city_data['dt'].dt.year

# Group by year and calculate mean temperature
annual = city_data.groupby('Year')['AverageTemperature'].mean().reset_index()

# Plot the annual average temperature
plt.figure(figsize=(12, 6))
plt.plot(annual['Year'], annual['AverageTemperature'], linestyle='-', color='blue',marker='.')
plt.title('Average Temperature of New York', fontsize=18)
plt.xlabel('Year', fontsize=16)
plt.ylabel('Average Temperature', fontsize=16)
plt.grid(True)
plt.show()
```



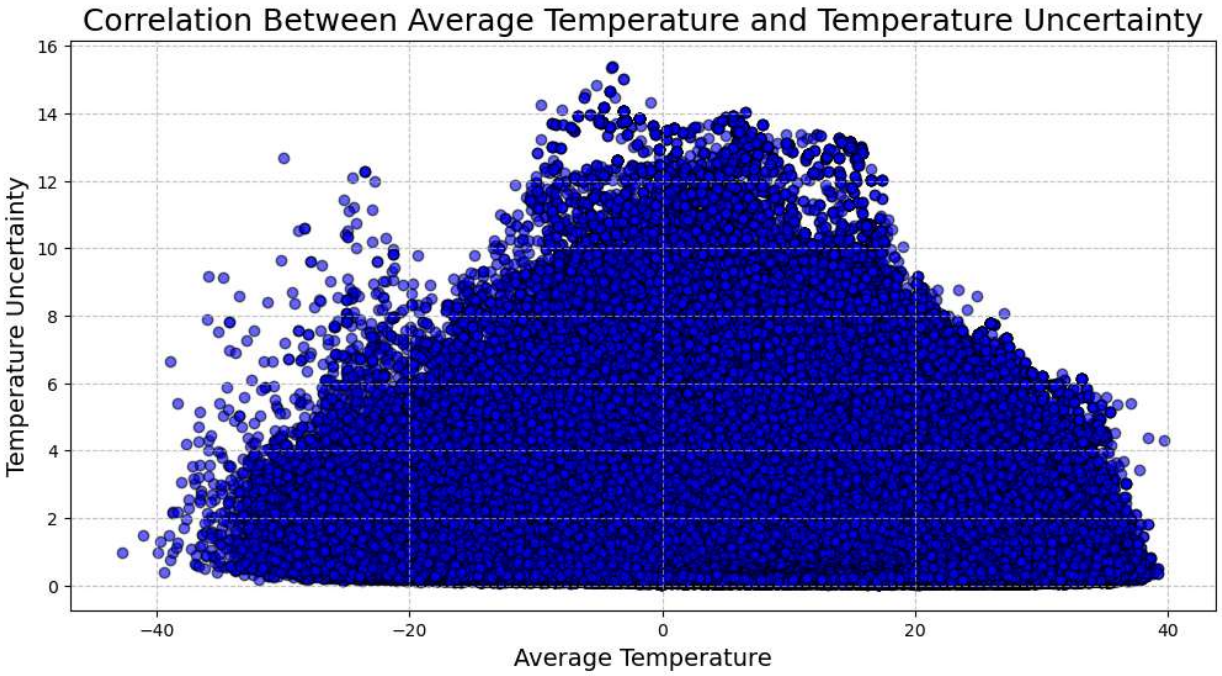
In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Group by country and calculate the average temperature
annual = df.groupby('Country')['AverageTemperature'].mean().reset_index()

# Sort by average temperature in descending order
annual_sorted = annual.sort_values(by='AverageTemperature', ascending=False).head(10)

# Plot a bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x='Country', y='AverageTemperature', data=annual_sorted, palette='Reds')
plt.title('Top 10 Countries with Highest Average Temperature', fontsize=18)
plt.xlabel('Country', fontsize=14)
plt.ylabel('Average Temperature', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



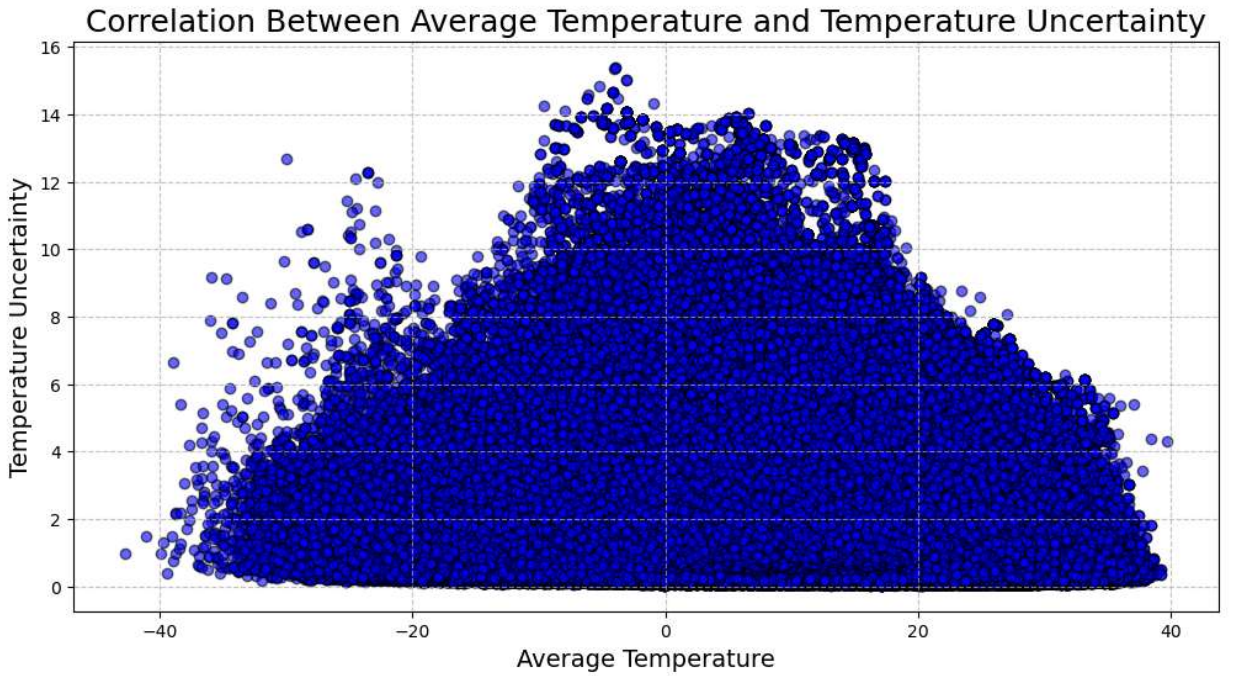
In [20]:
```
import matplotlib.pyplot as plt

# Drop missing or invalid values in AverageTemperature column
temperature_data = df['AverageTemperature'].dropna()

# Plot histogram
plt.figure(figsize=(12, 6))
plt.hist(temperature_data, bins=30, color='purple', edgecolor='black', alpha=0.7)
plt.title('Distribution of Average Temperatures', fontsize=18)
plt.xlabel('Temperature Ranges', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [25]:

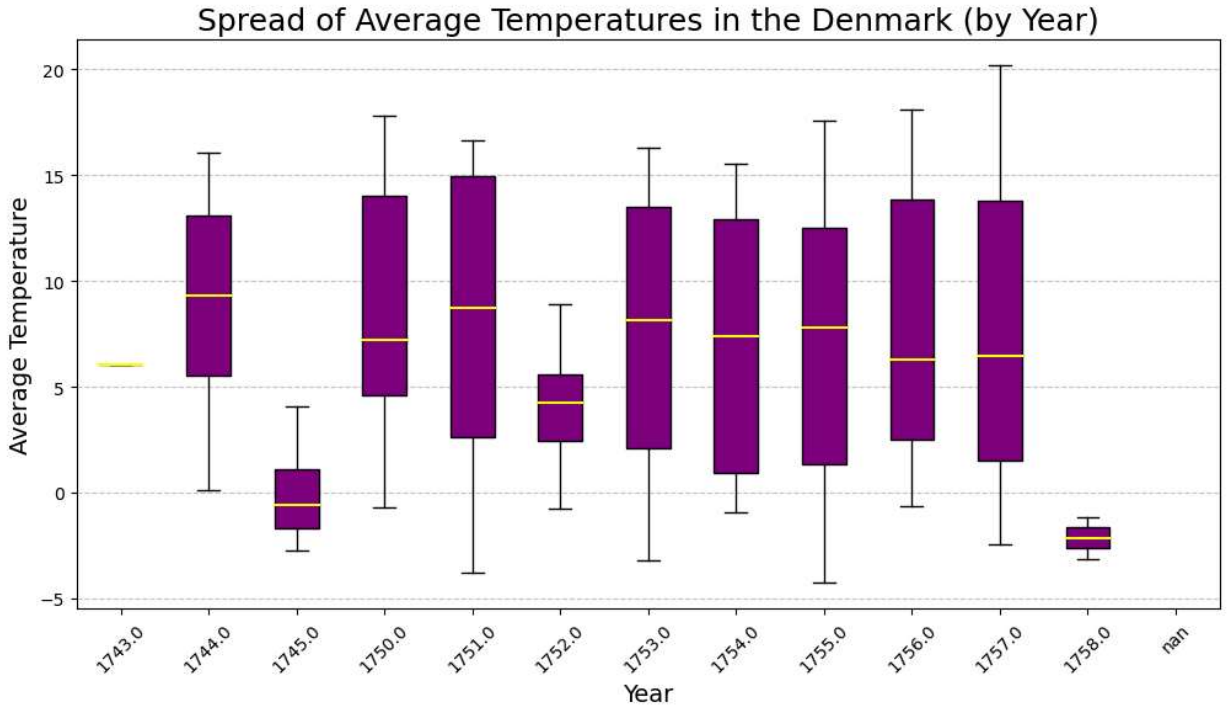


In [34]:
```
import matplotlib.pyplot as plt

# Drop rows with missing values in relevant columns
data = df[['AverageTemperature', 'AverageTemperatureUncertainty']].dropna()

# Plot scatter plot
plt.figure(figsize=(12, 6))
plt.scatter(data['AverageTemperature'], data['AverageTemperatureUncertainty'], alpha=0.6, color='blu
plt.title('Correlation Between Average Temperature and Temperature Uncertainty', fontsize=18)
plt.xlabel('Average Temperature', fontsize=14)
plt.ylabel('Temperature Uncertainty', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
correlation = data.corr()
print(correlation)
```




```
                            AverageTemperature  \
AverageTemperature                    1.000000
AverageTemperatureUncertainty        -0.228159

                            AverageTemperatureUncertainty
AverageTemperature                              -0.228159
AverageTemperatureUncertainty                    1.000000
```

In [41]:
```
import matplotlib.pyplot as plt

# Filter the dataset for the specific country
country_data = df[df['Country'] == 'Denmark'].dropna()

# Extract the relevant data: Year and AverageTemperature
country_data['Year'] = pd.to_datetime(country_data['dt']).dt.year.head(100)
temperature_data = country_data[['Year', 'AverageTemperature']]

# Plot boxplot for temperature spread by year
plt.figure(figsize=(12, 6))
plt.boxplot(
    [temperature_data[temperature_data['Year'] == year]['AverageTemperature'] for year in temperatur
    labels=temperature_data['Year'].unique(),
    patch_artist=True,
    boxprops=dict(facecolor='purple', color='black'),
    medianprops=dict(color='yellow', linewidth=1.5),
    flierprops=dict(marker='o', color='red', alpha=0.5),
)

plt.title('Spread of Average Temperatures in the Denmark (by Year)', fontsize=18)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Average Temperature', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



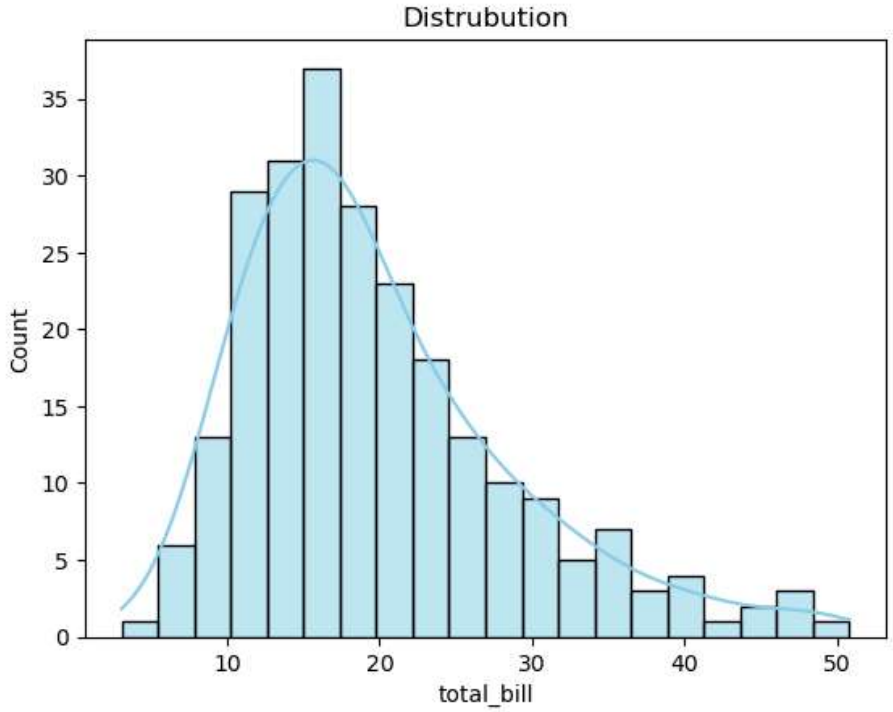
In [42]:
```
import seaborn as sns
```

In [43]:
```
tips=sns.load_dataset('tips')
tips.head()
```

Out[43]:

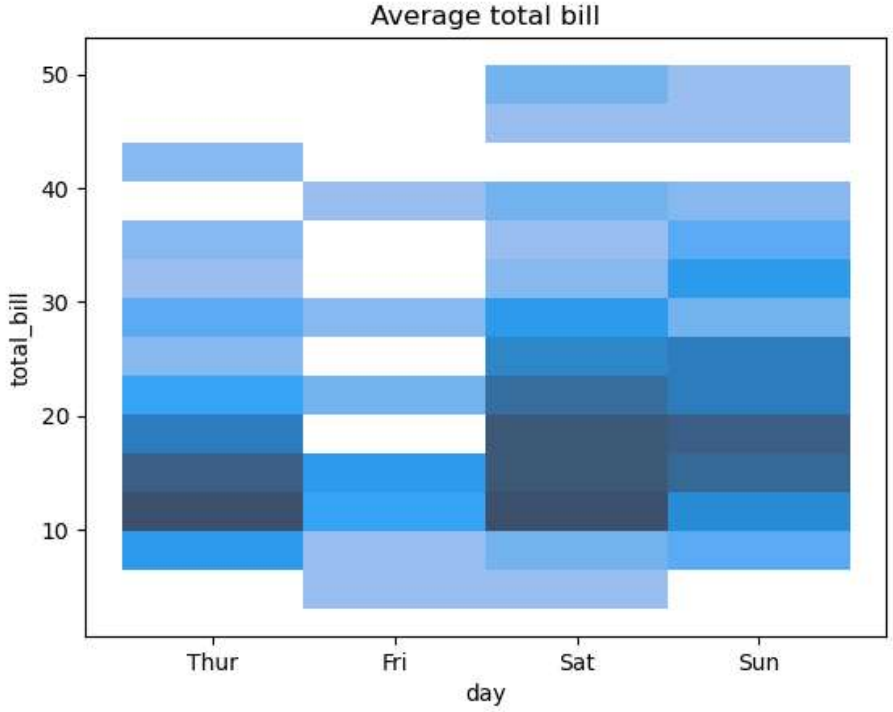
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [44]:
```
sns.histplot(data=tips,x='total_bill',kde=True,bins=20,color='skyblue')
plt.title('Distrubution')
plt.show()
```

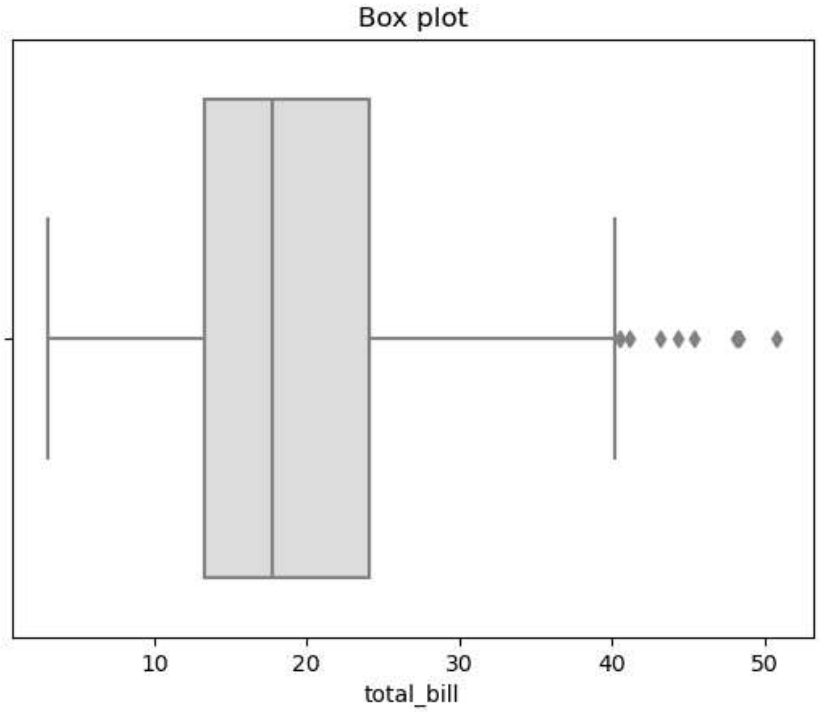


In [45]:
```
sns.histplot(data=tips,x='day',y='total_bill',kde=True,palette='viridis')
plt.title('Average total bill')
plt.show()
```

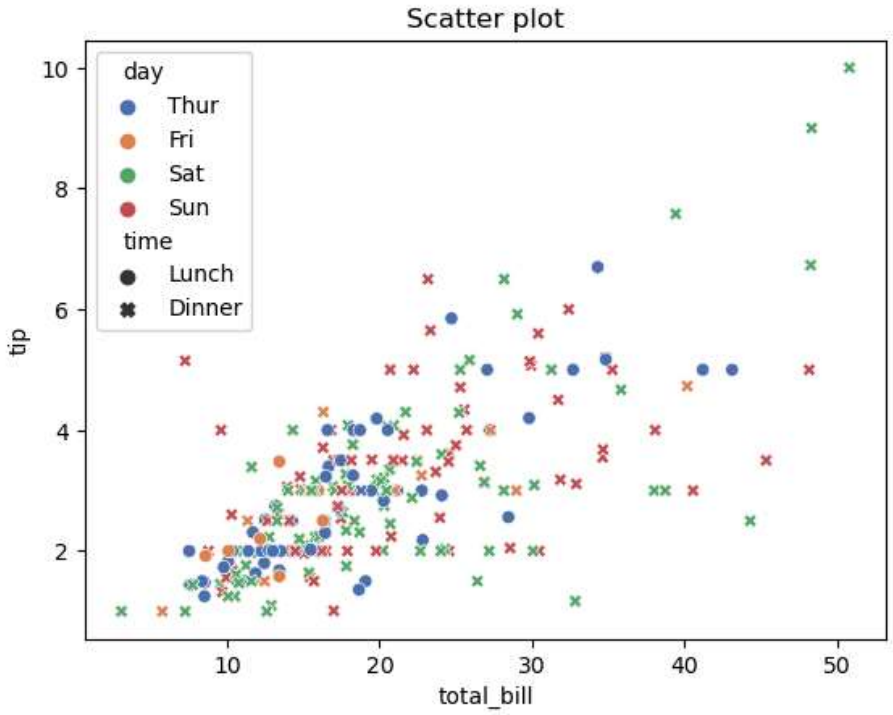
```
C:\Users\cvr\AppData\Local\Temp\ipykernel_13652\2739178180.py:1: UserWarning: Ignoring `palette` be
cause no `hue` variable has been assigned.
  sns.histplot(data=tips,x='day',y='total_bill',kde=True,palette='viridis')
```



In [46]:
```
sns.boxplot(data=tips,x='total_bill',hue='sex',palette='coolwarm')
plt.title('Box plot')
plt.show()
```



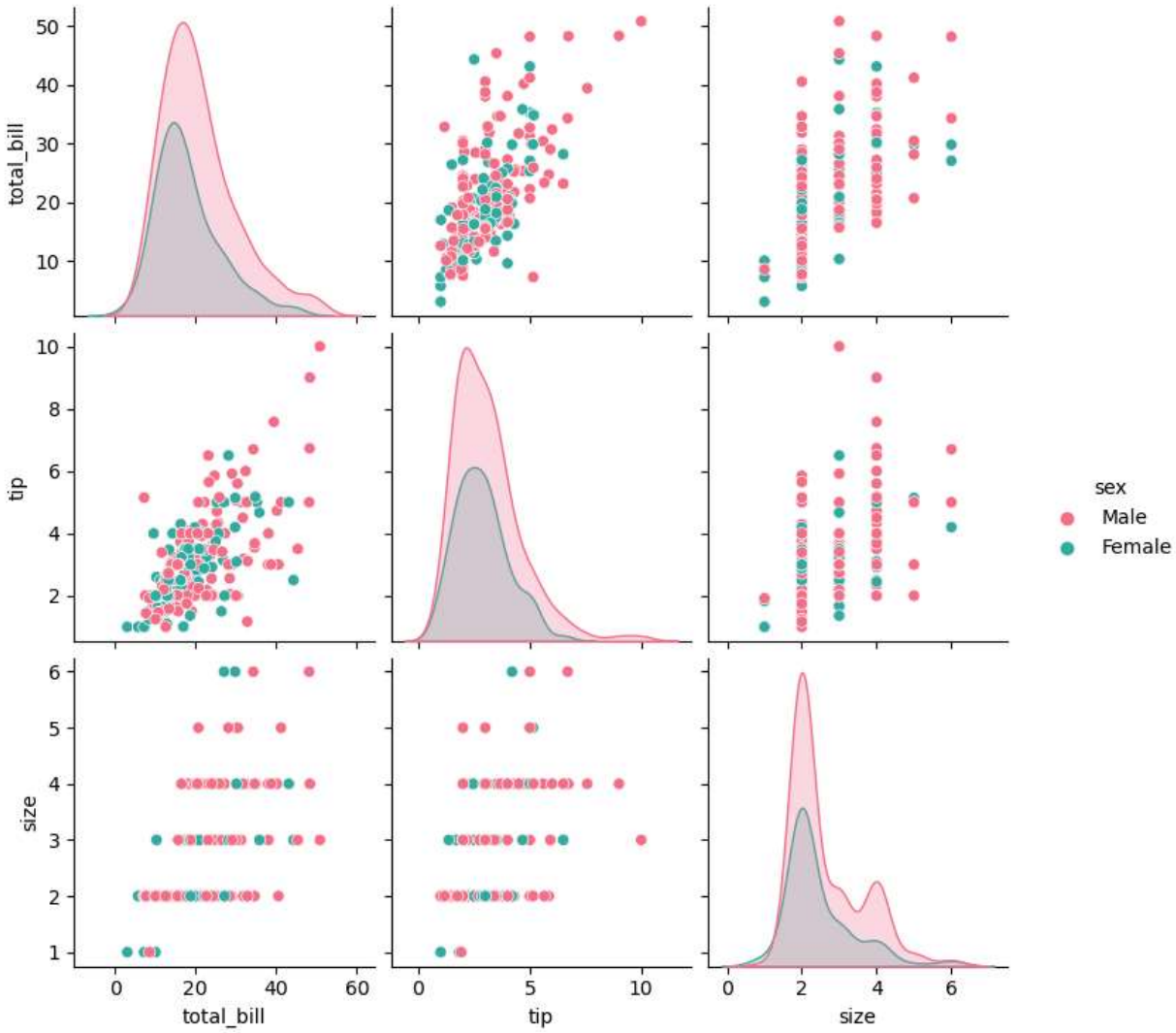
In [47]:
```
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time',palette='deep')
plt.title('Scatter plot')
plt.show()
```



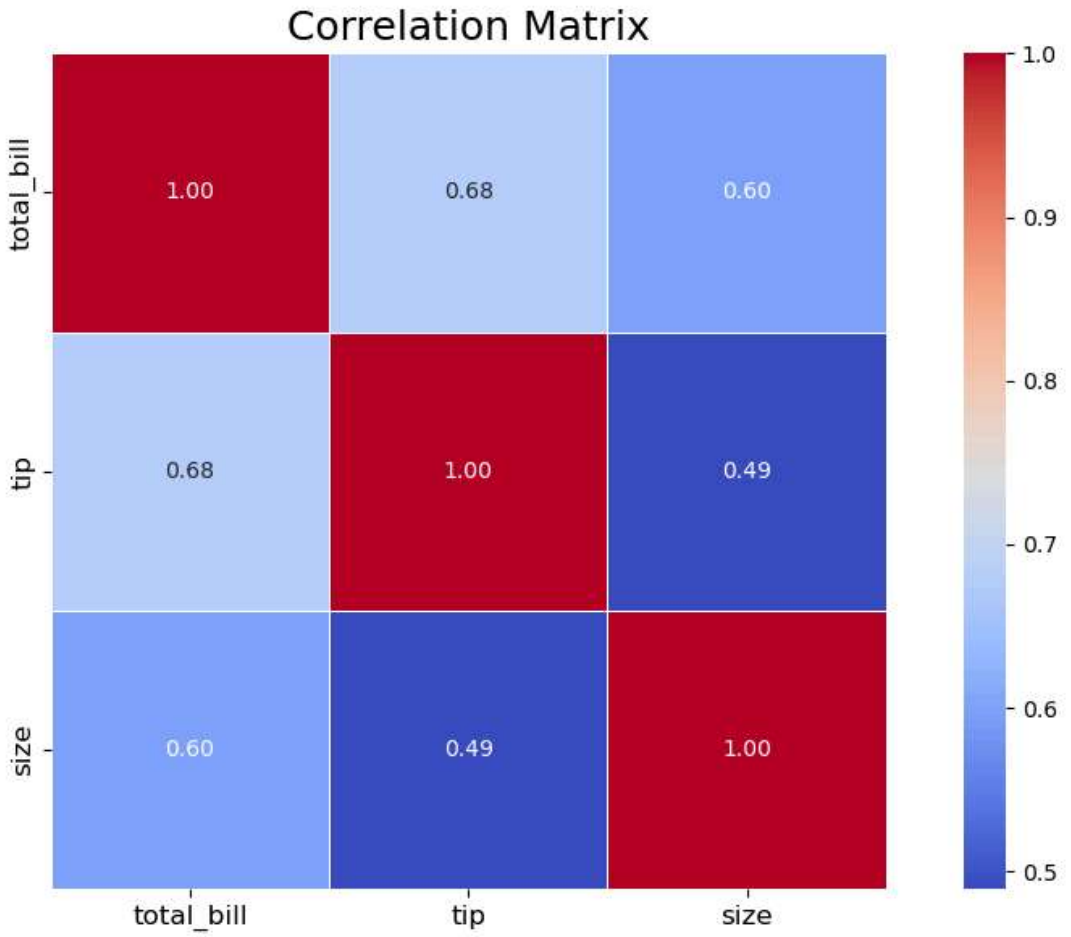
In [48]:
```
sns.lineplot(data=tips,x='size',y='tip',marker='*',color='green')
plt.title('Line plot')
plt.show()
```

In [49]:
```
sns.pairplot(data=tips,hue='sex',palette='husl')
plt.show()
```

```
C:\Users\cvr\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [54]:
```
import seaborn as sns
import matplotlib.pyplot as plt
n=tips.select_dtypes(include='number')
# Compute the correlation matrix
c =n.corr()

# Create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(
    c,
    annot=True,
    cmap='coolwarm',
    fmt='.2f',
    linewidths=0.5,  # Add gridlines between cells
    square=True      # Ensure cells are square-shaped
)
plt.title('Correlation Matrix', fontsize=18)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight_layout()
plt.show()
```

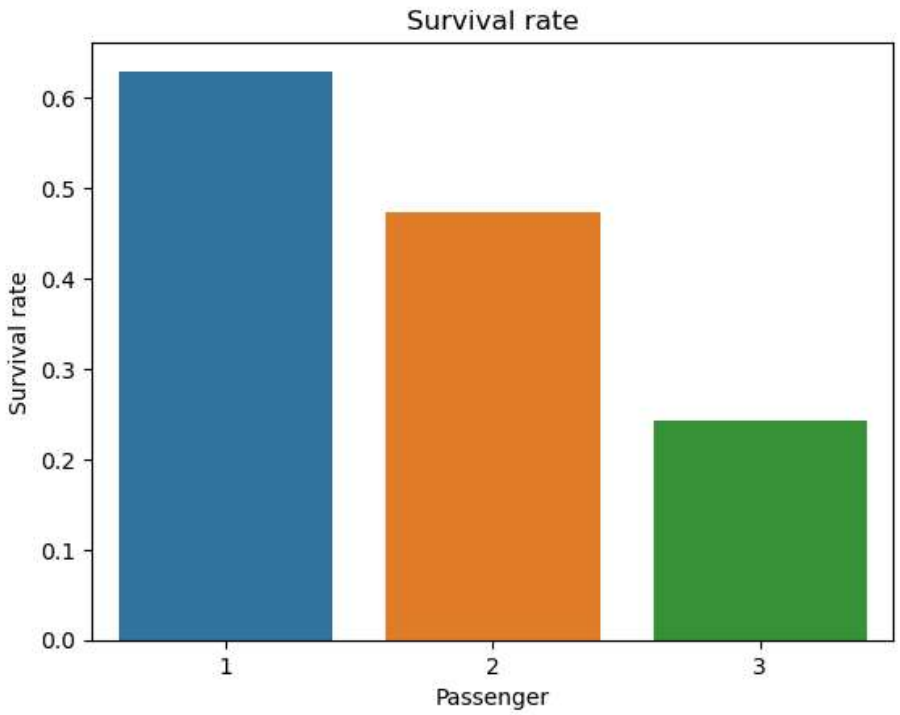


In [55]:
```
train=pd.read_csv('train.csv')
train.head()
```

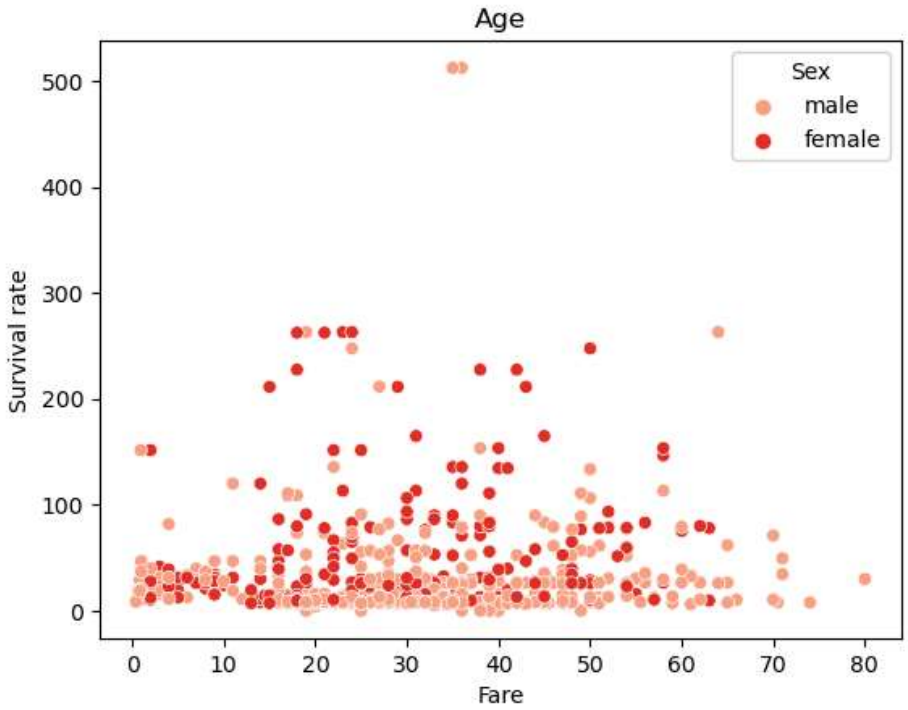
Out[55]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [56]:
```
sns.barplot(data=train,x='Pclass',y='Survived',errorbar=None)
plt.title('Survival rate')
plt.xlabel('Passenger')
plt.ylabel('Survival rate')
plt.show()
```

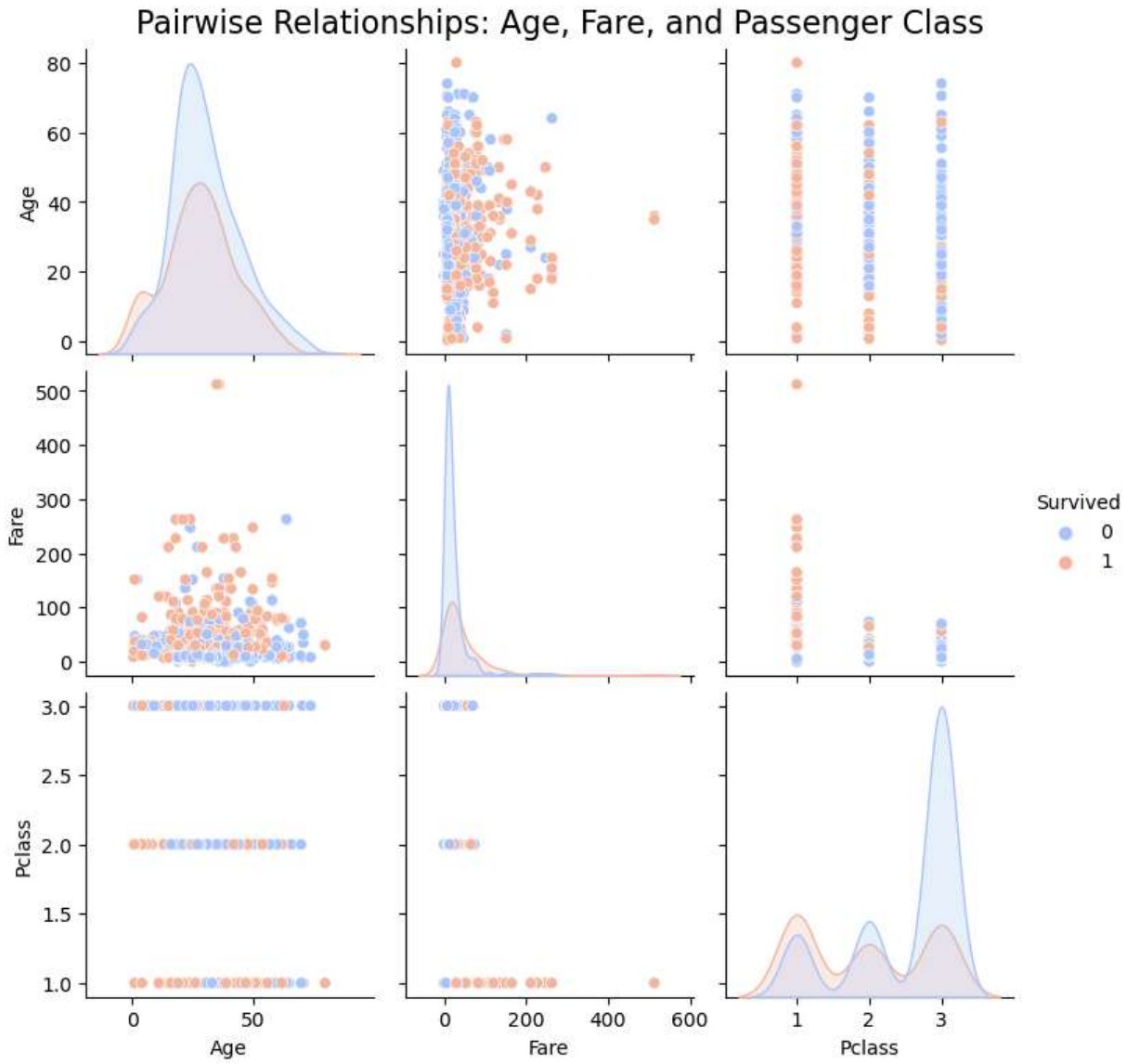


In [60]:
```
sns.scatterplot(data=train,x='Age',y='Fare',hue='Sex',palette='Reds')
plt.title('Age')
plt.xlabel('Fare')
plt.ylabel('Survival rate')
plt.show()
```

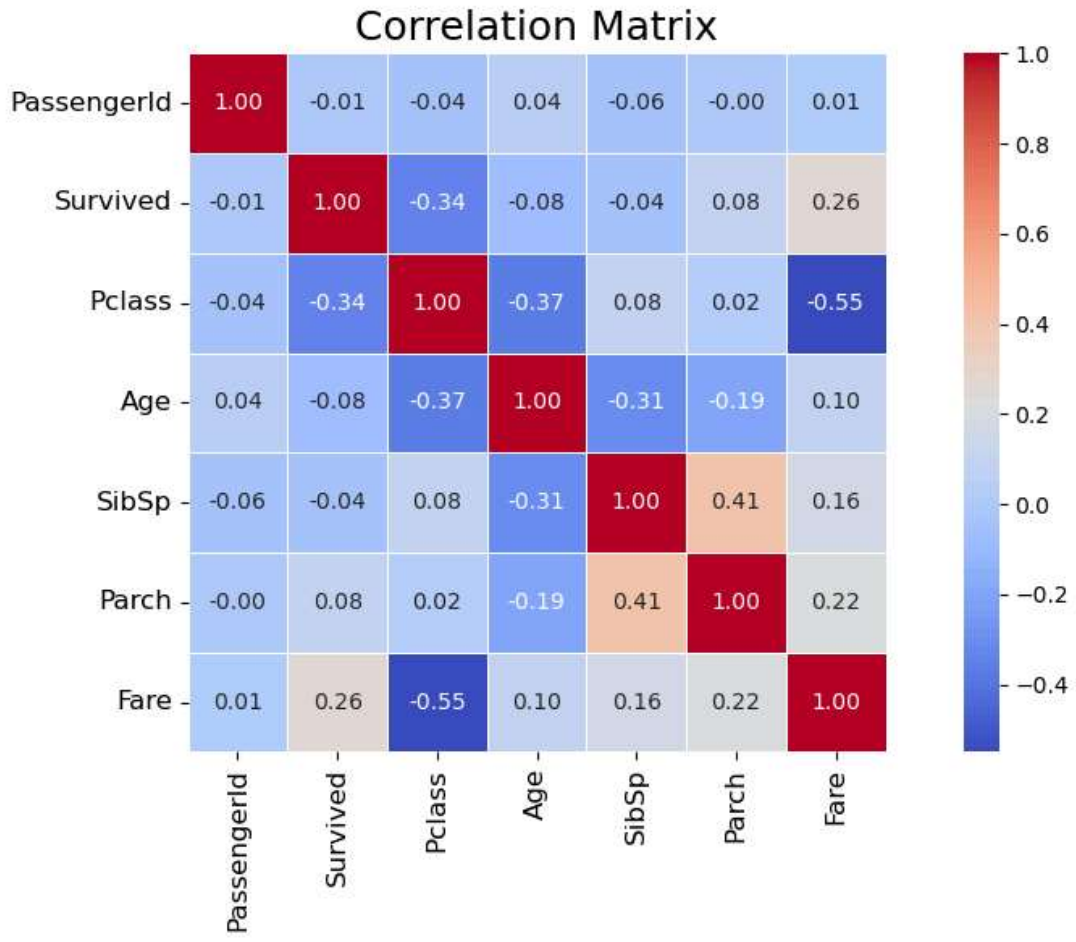


In [68]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Subset the DataFrame to include only the relevant columns
subset = train[['Age', 'Fare', 'Pclass','Survived']]

# Create the pairplot
sns.pairplot(
    subset,
    hue='Survived',  # Use 'Pclass' for color-coding
    palette='coolwarm',
    diag_kind='kde',  # Use Kernel Density Estimate for diagonal plots
     # Different markers for each class
)

# Add a title to the pairplot
plt.suptitle('Pairwise Relationships: Age, Fare, and Passenger Class', y=1.02, fontsize=16)
plt.show()
```

```
C:\Users\cvr\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout ha
s changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [73]:
```
import seaborn as sns
import matplotlib.pyplot as plt
n=train.select_dtypes(include='number')
# Compute the correlation matrix
c =n.corr()

# Create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(
    c,
    annot=True,
    cmap='coolwarm',
    fmt='.2f',
    linewidths=0.5,  # Add gridlines between cells
    square=True      # Ensure cells are square-shaped
)
plt.title('Correlation Matrix', fontsize=18)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight_layout()
plt.show()
```



In [ ]:
```

```