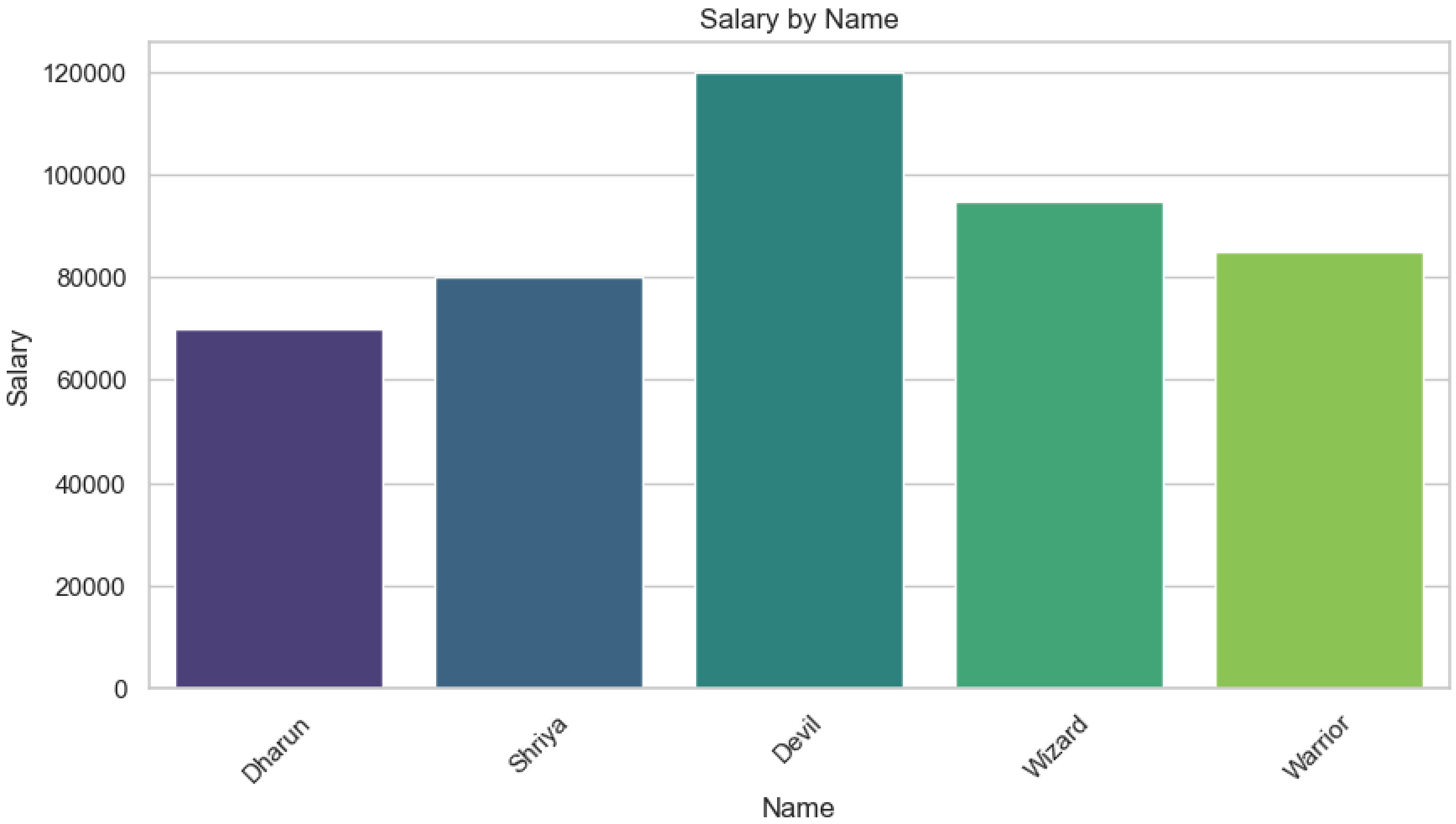
In [2]:

```
#Data Visualization

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data creation with specified names
data = {
    'Name': ['Dharun', 'Shriya', 'Devil', 'Wizard', 'Warrior'],
    'Age': [25, 29, 30, 35, 32],
    'City': ['New York', 'Los Angeles', 'Chicago', 'New York', 'Los Angeles'],
    'Salary': [70000, 80000, 120000, 95000, 85000]
}

# Create a DataFrame
df = pd.DataFrame(data)
print(df)
```

```
      Name  Age         City  Salary
0   Dharun   25     New York   70000
1   Shriya   29  Los Angeles   80000
2    Devil   30      Chicago  120000
3   Wizard   35     New York   95000
4  Warrior   32  Los Angeles   85000
```

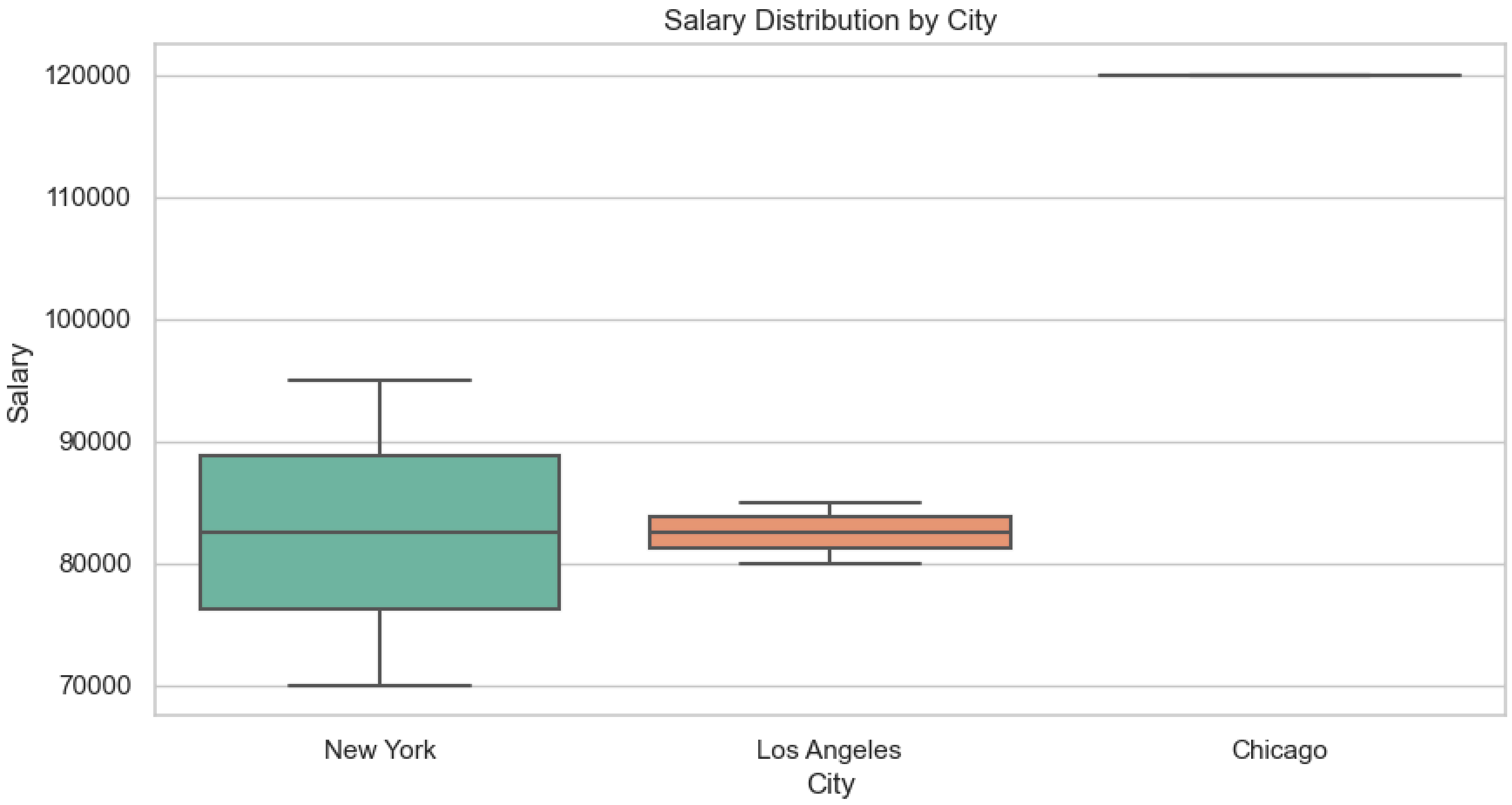
In [3]:

```
# Set the aesthetic style of the plots
sns.set(style="whitegrid")

# 1. Bar Plot: Salary by Name
plt.figure(figsize=(10, 5))
sns.barplot(x='Name', y='Salary', data=df, palette='viridis')
plt.title('Salary by Name')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.xticks(rotation=45)
plt.show()
```


Salary by Name
Salary
120000
100000
80000
60000
40000
20000
0
Dharun
Shriya
Devil
Wizard
Warrior
Name

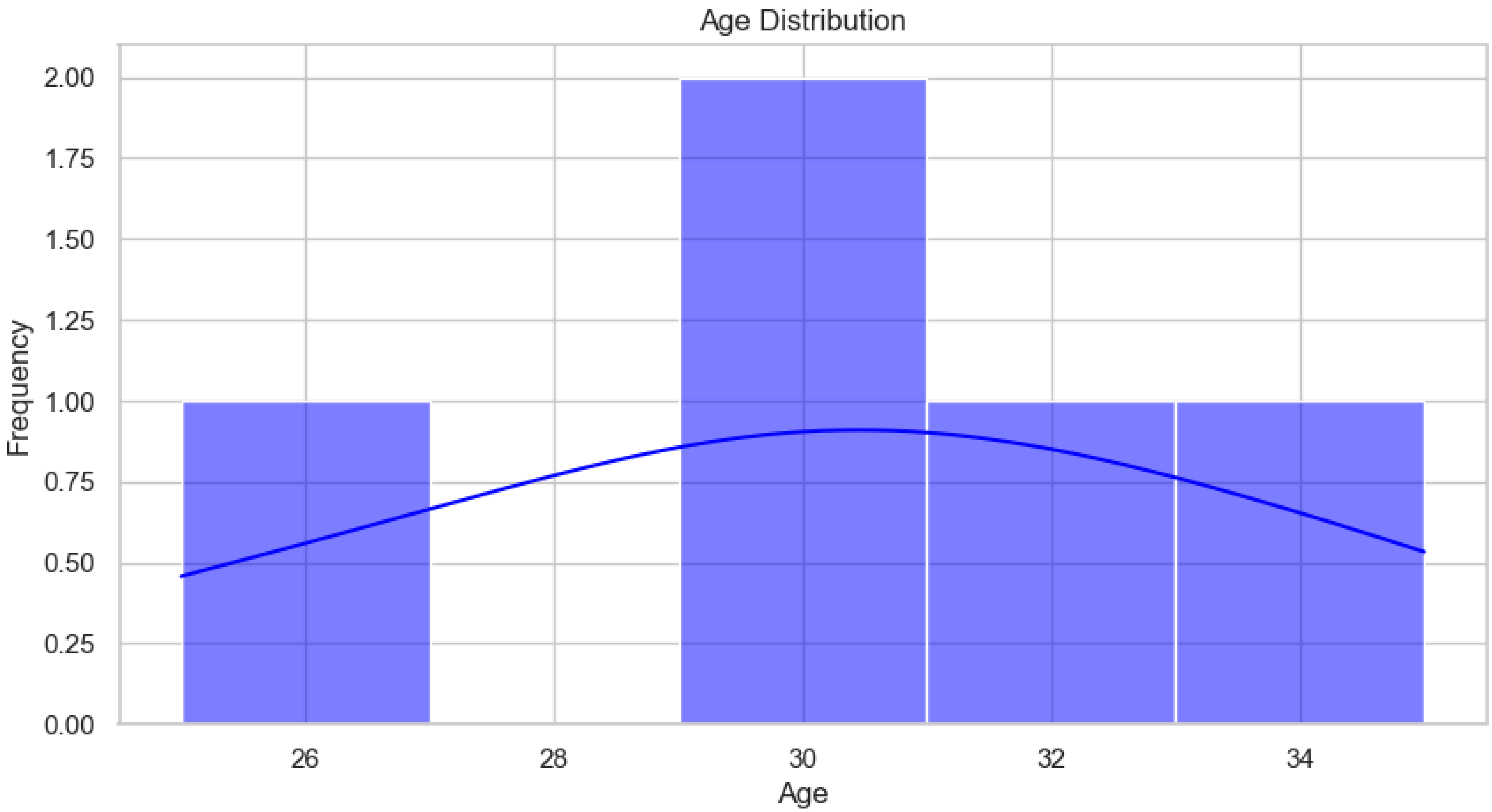
In [4]:

```python
# 2. Box Plot: Salary Distribution
plt.figure(figsize=(10, 5))
sns.boxplot(x='City', y='Salary', data=df, palette='Set2')
plt.title('Salary Distribution by City')
plt.xlabel('City')
plt.ylabel('Salary')
plt.show()
```

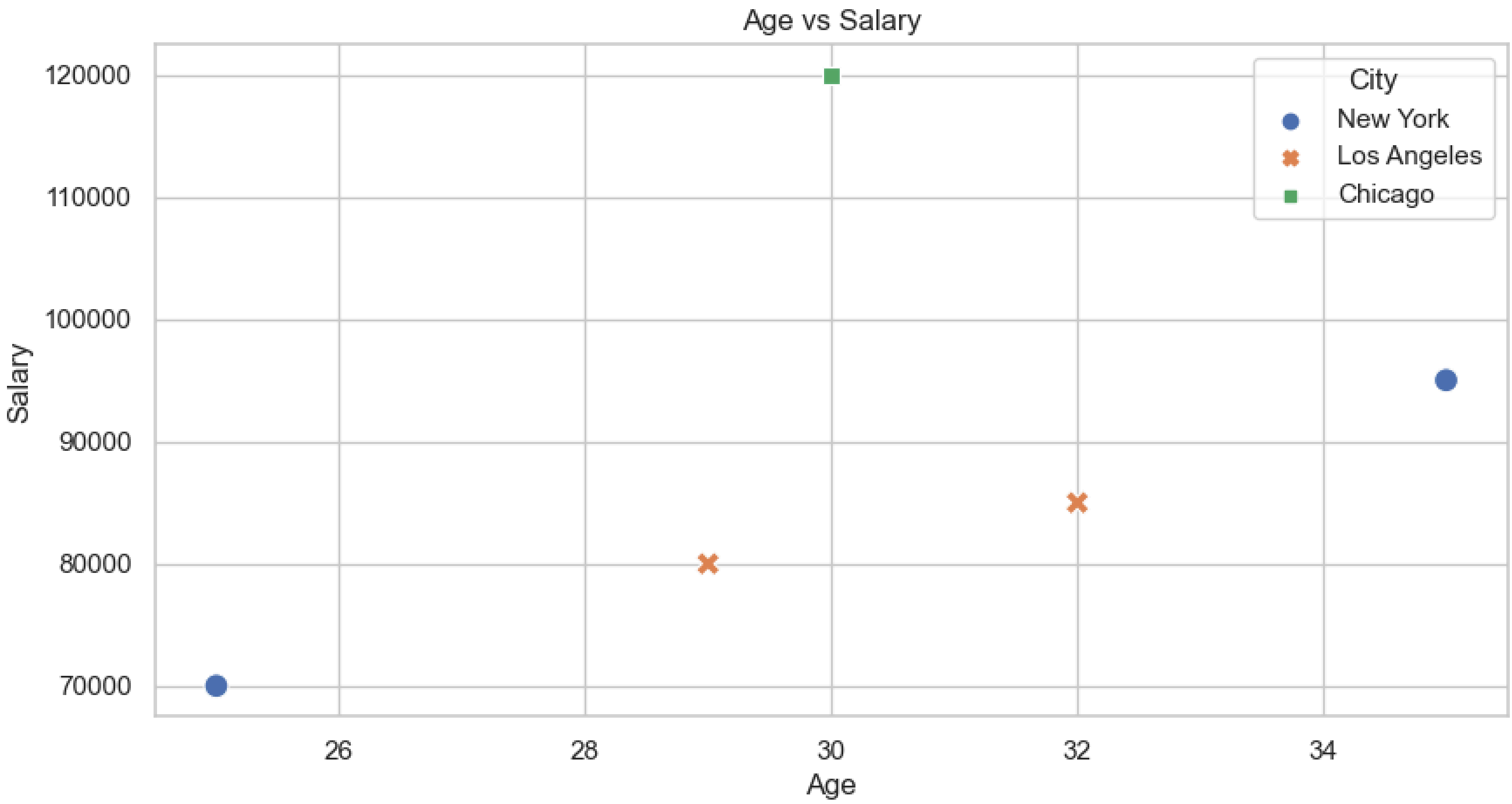


In [5]:

```
# 3. Histogram: Age Distribution
plt.figure(figsize=(10, 5))
sns.histplot(df['Age'], bins=5, kde=True, color='blue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

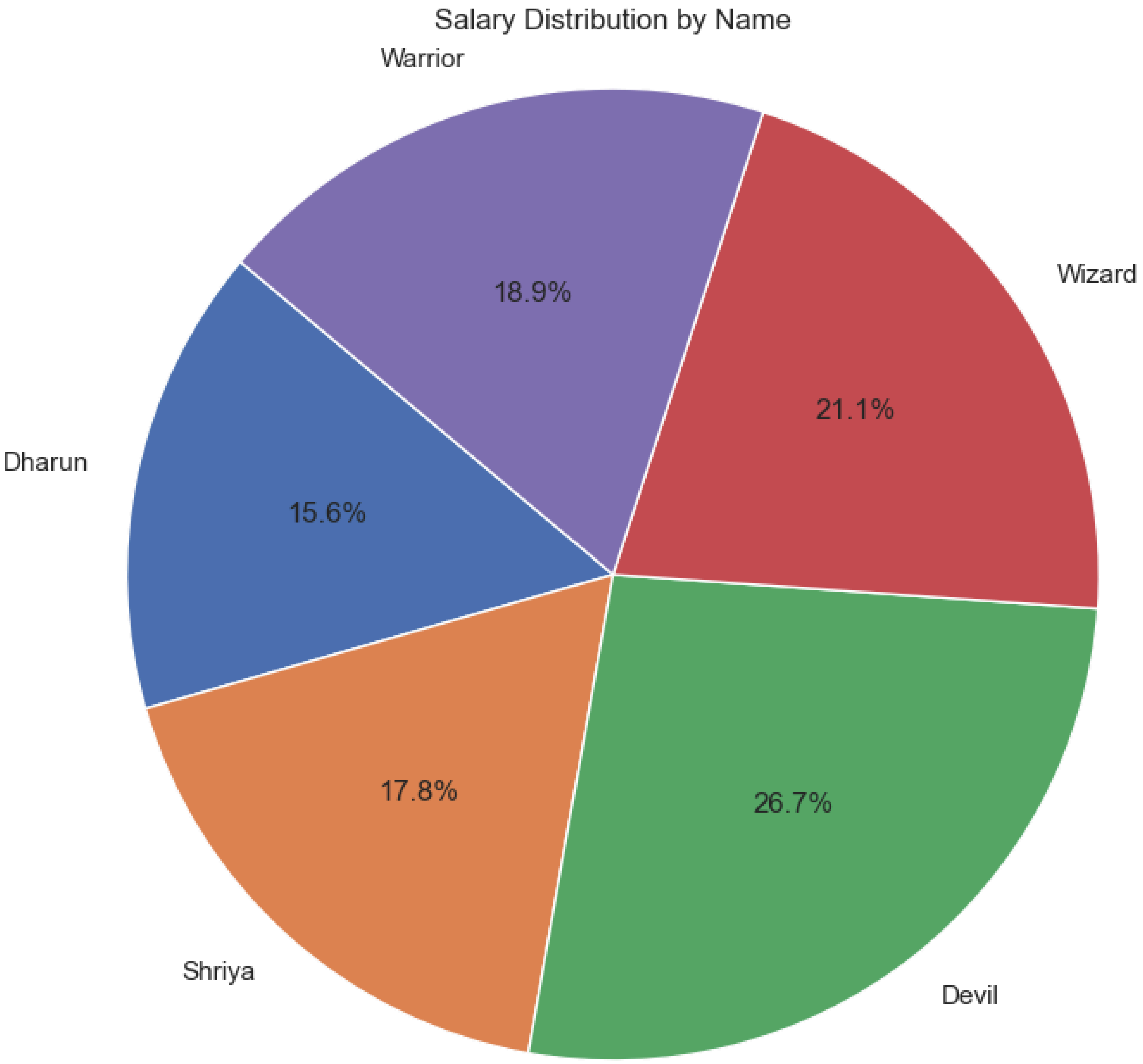


```
In [6]: # 4. Scatter Plot: Age vs Salary
        plt.figure(figsize=(10, 5))
        sns.scatterplot(x='Age', y='Salary', hue='City', style='City', data=df, s=100)
        plt.title('Age vs Salary')
        plt.xlabel('Age')
        plt.ylabel('Salary')
        plt.legend(title='City')
        plt.show()
```



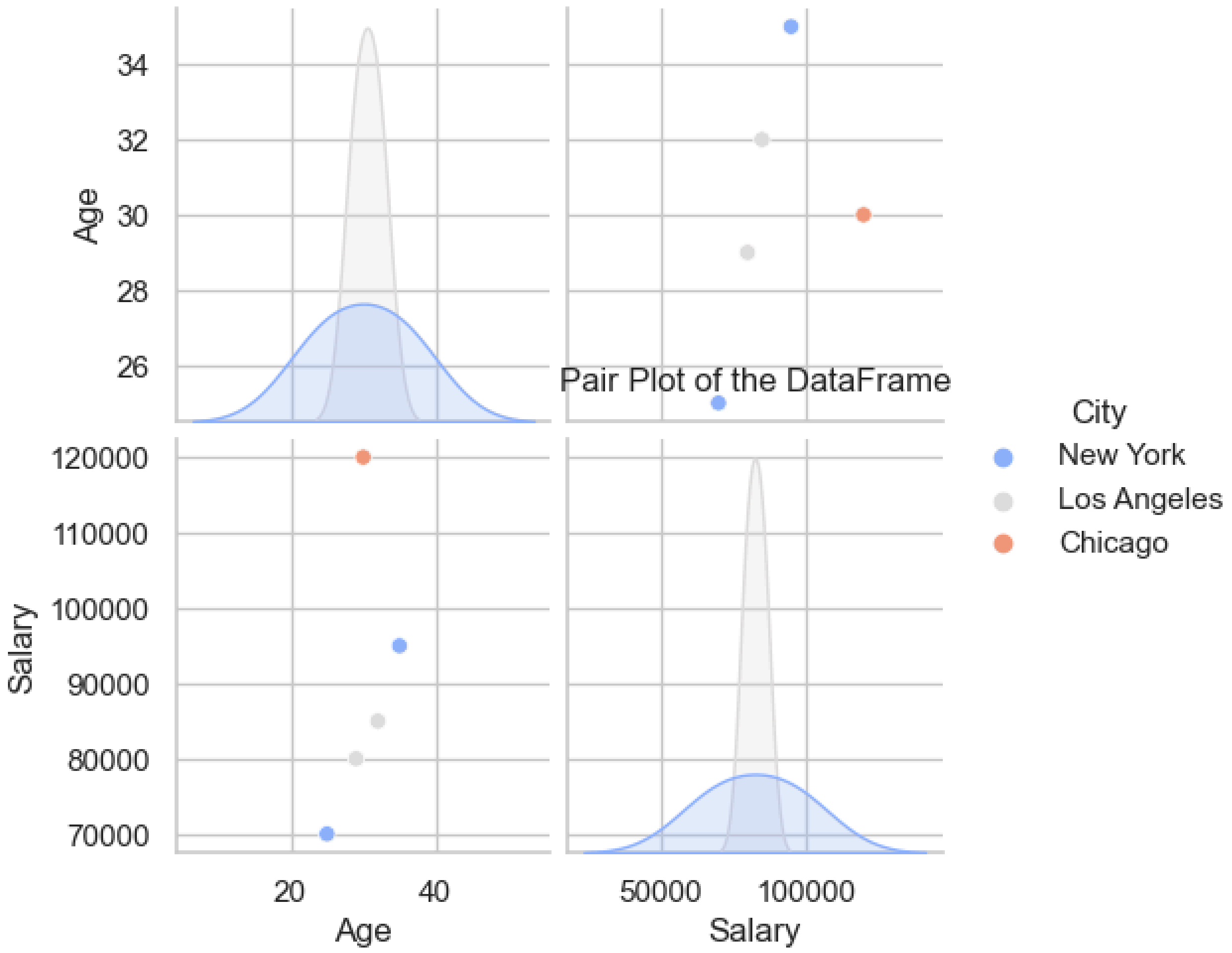
In [7]:

```
# 5. Pie Chart: Salary Distribution
plt.figure(figsize=(8, 8))
plt.pie(df['Salary'], labels=df['Name'], autopct='%1.1f%%', startangle=140)
plt.title('Salary Distribution by Name')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```


Salary Distribution by Name
Warrior
18.9%
Wizard
21.1%
Dharun
15.6%
17.8%
26.7%
Shriya
Devil

```
In [9]: print("Pair Plot: Visualizing pairwise relationships")
        sns.pairplot(df, hue='City', palette='coolwarm')
        plt.title('Pair Plot of the DataFrame')
        plt.show()
```

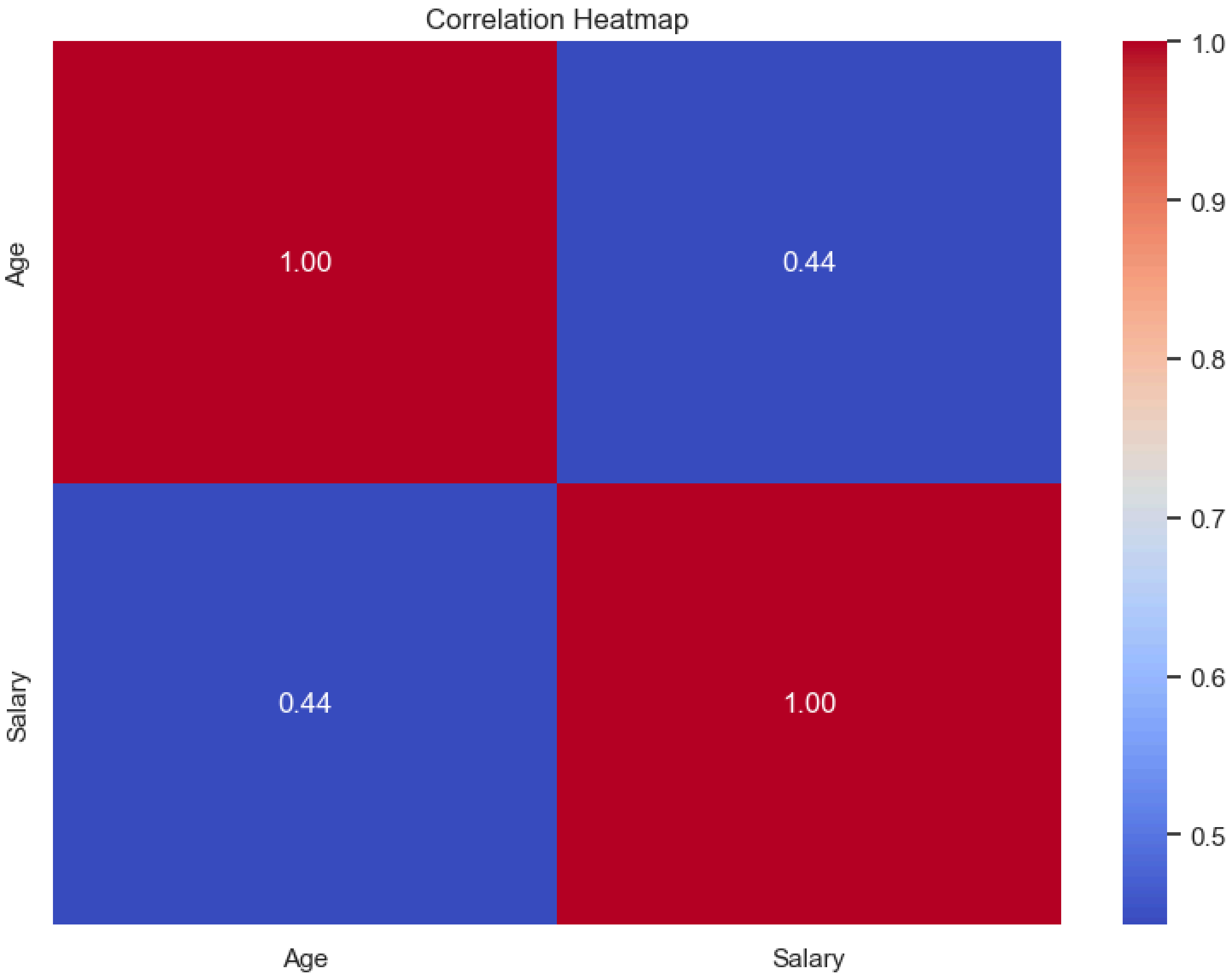
```
Pair Plot: Visualizing pairwise relationships

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



In [11]:

```
print("Correlation Heatmap")
plt.figure(figsize=(8, 6))
correlation = df[['Age', 'Salary']].corr()  # Select only numeric columns for correlation
sns.heatmap(correlation, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```

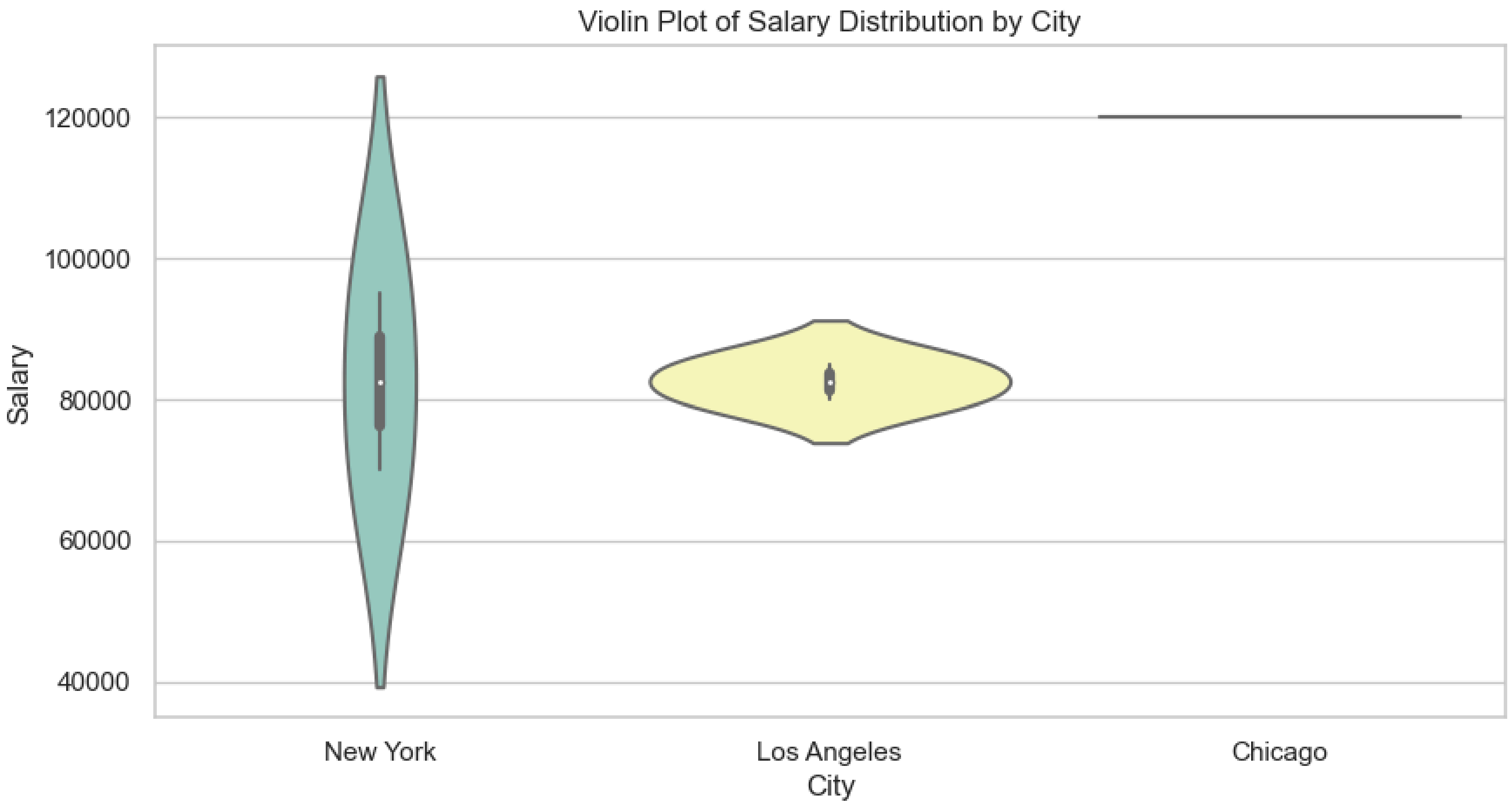
```
Correlation Heatmap
```


Correlation Heatmap
Age
Salary
1.00
0.44
0.44
1.00
Age
Salary
1.0
0.9
0.8
0.7
0.6
0.5

In [12]:

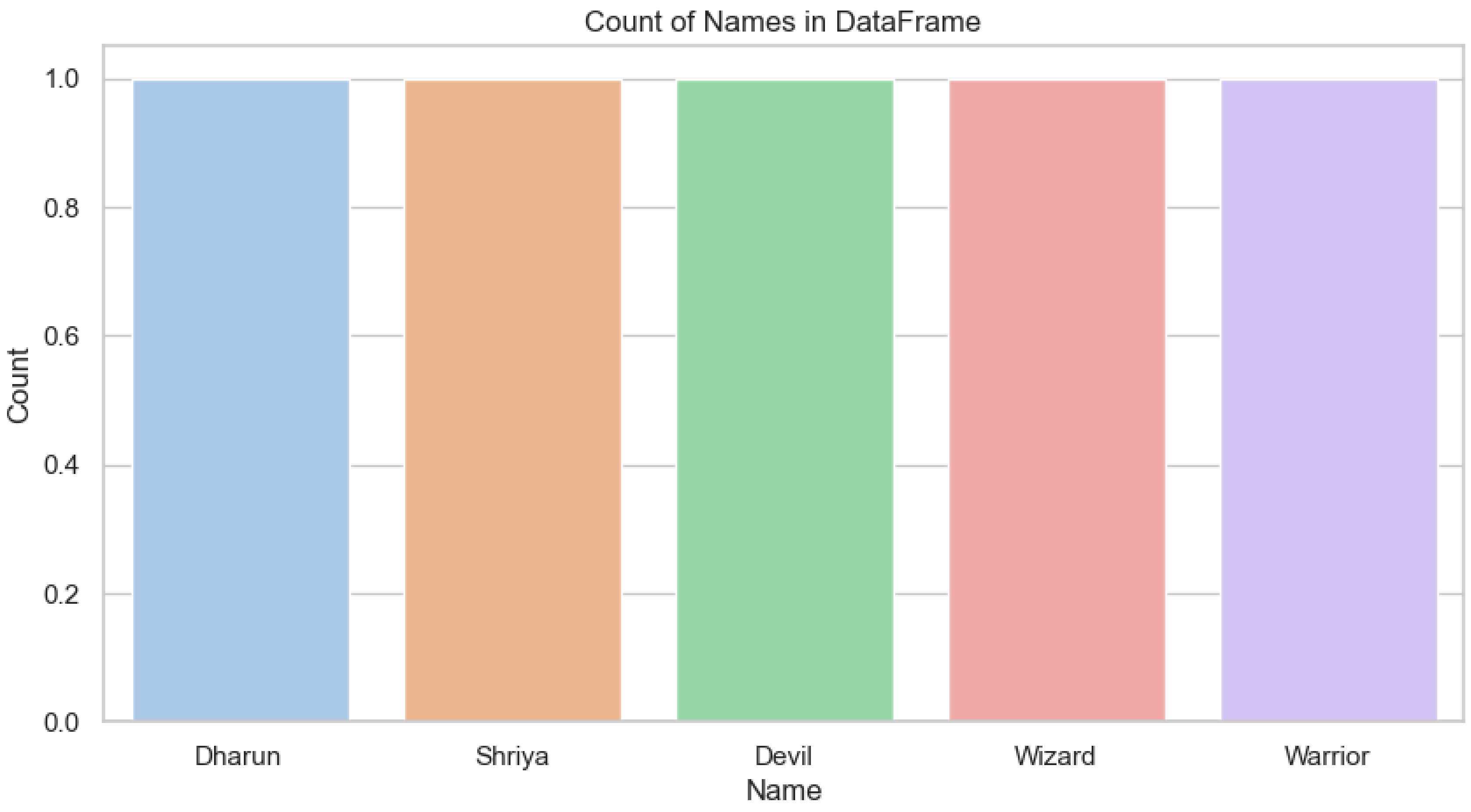
```
print("Violin Plot: Salary Distribution by City")
plt.figure(figsize=(10, 5))
sns.violinplot(x='City', y='Salary', data=df, palette='Set3')
plt.title('Violin Plot of Salary Distribution by City')
plt.xlabel('City')
plt.ylabel('Salary')
plt.show()
```

```
Violin Plot: Salary Distribution by City
```



In [13]:
```
print("Count Plot: Count of Names")
plt.figure(figsize=(10, 5))
sns.countplot(x='Name', data=df, palette='pastel')
plt.title('Count of Names in DataFrame')
plt.xlabel('Name')
plt.ylabel('Count')
plt.show()
```

```
Count Plot: Count of Names
```



In [14]:

```
print("Exporting DataFrame to Excel")
df.to_excel('data_visualization_output.xlsx', index=False)
print("DataFrame exported successfully!")
```

```
Exporting DataFrame to Excel
DataFrame exported successfully!
```

In [15]:

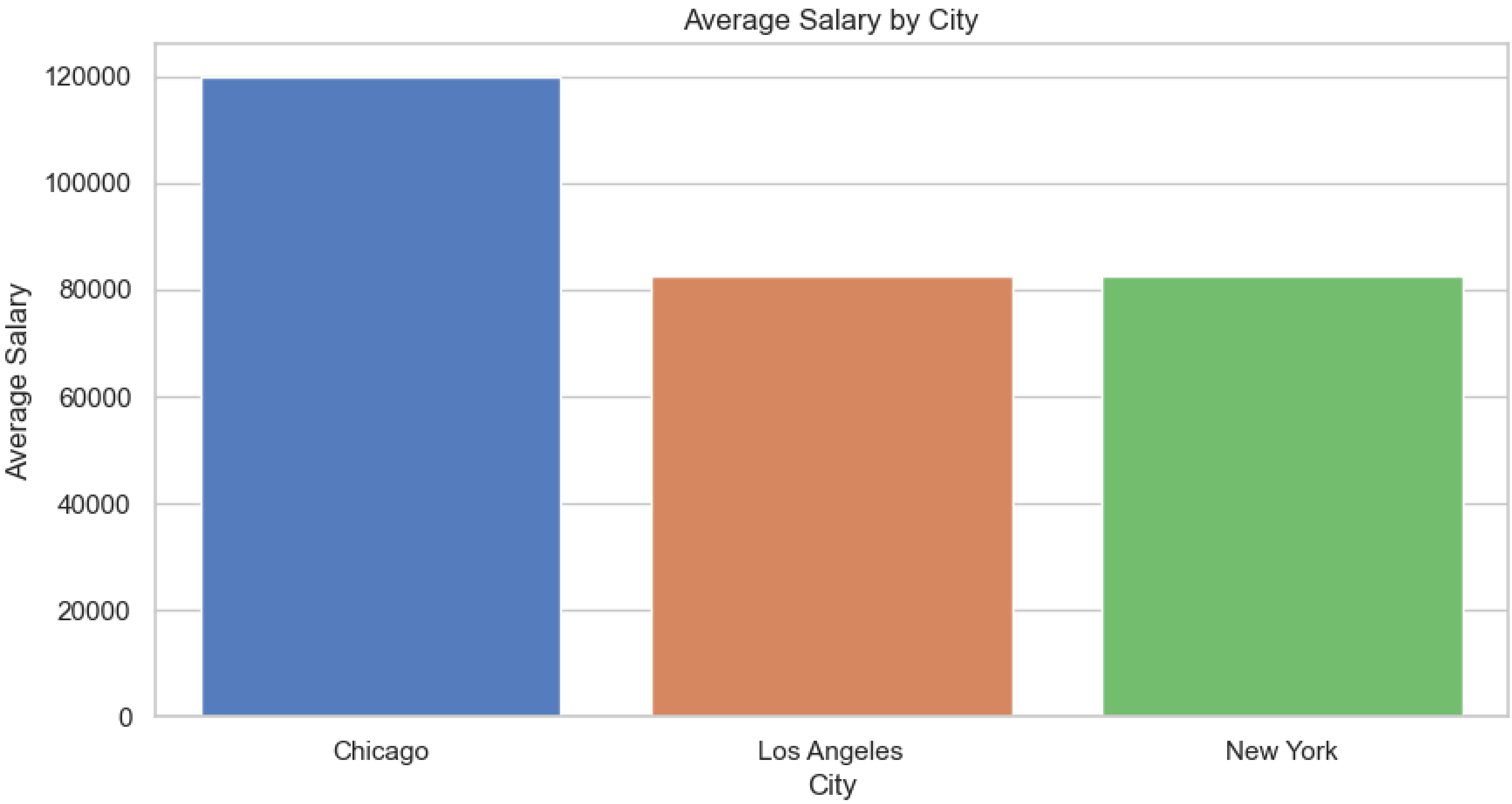
```
print("Descriptive Statistics of the DataFrame")
stats = df.describe()
print(stats)
```

```
Descriptive Statistics of the DataFrame
             Age         Salary
count   5.000000       5.000000
mean   30.200000   90000.000000
std     3.701351   19039.432765
min    25.000000   70000.000000
25%    29.000000   80000.000000
50%    30.000000   85000.000000
75%    32.000000   95000.000000
max    35.000000  120000.000000
```

In [16]:

```
print("Average Salary by City")
avg_salary_by_city = df.groupby('City')['Salary'].mean().reset_index()
print(avg_salary_by_city)

# Bar Plot for Average Salary by City
plt.figure(figsize=(10, 5))
sns.barplot(x='City', y='Salary', data=avg_salary_by_city, palette='muted')
plt.title('Average Salary by City')
plt.xlabel('City')
plt.ylabel('Average Salary')
plt.show()
```

```
Average Salary by City
          City    Salary
0      Chicago  120000.0
1  Los Angeles   82500.0
2     New York   82500.0
```



```
In [ ]: 220901020 - DHARUN J
```