Step 1: Introduction

Data Description:

The dataset is sourced from the Aspiring Mind Employment Outcome 2015 (AMEO) study, focusing on engineering graduates. It contains approximately 4000 data points across 40 independent variables, which include demographic information, educational performance metrics, and employment outcomes.

Objective:

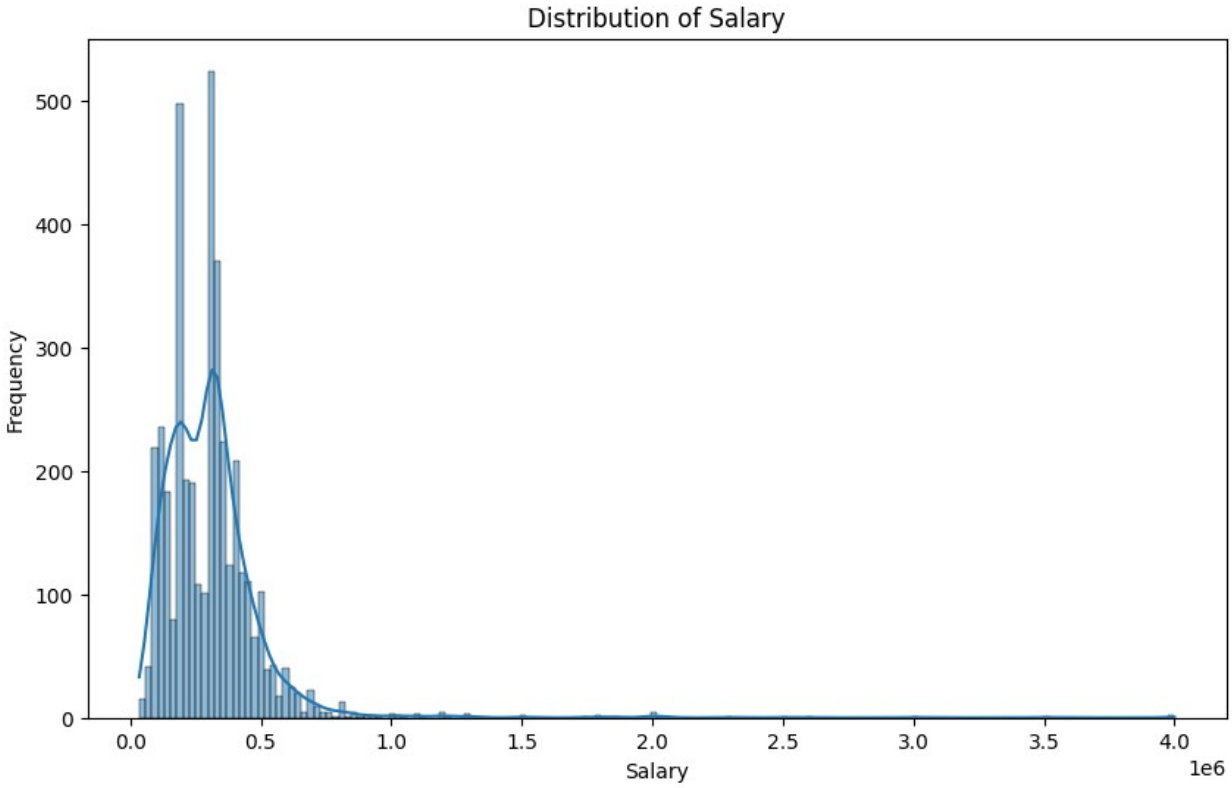
The primary goal of this analysis is to explore how various independent variables impact the target variable (Salary) while also examining other relationships within the dataset. Specifically, this report will analyze univariate and bivariate relationships, test a claim regarding salary expectations for recent engineering graduates, and explore the relationship between gender and specialization.

Step 1: Importing Libraries and Loading the Dataset

```
# importing necessary libraries

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

# load the dataset
data = pd.read_excel('/content/data.xlsx')

data.head()

{"type":"dataframe","variable_name":"data"}

data.tail()

{"type":"dataframe"}

# shape
# Displaying the shape of the dataset (rows, columns)
print(f"Shape of the dataset: {data.shape}")


Shape of the dataset: (3998, 39)

# Summary statistics of the dataset
data.describe(include='all')

{"type":"dataframe"}
```

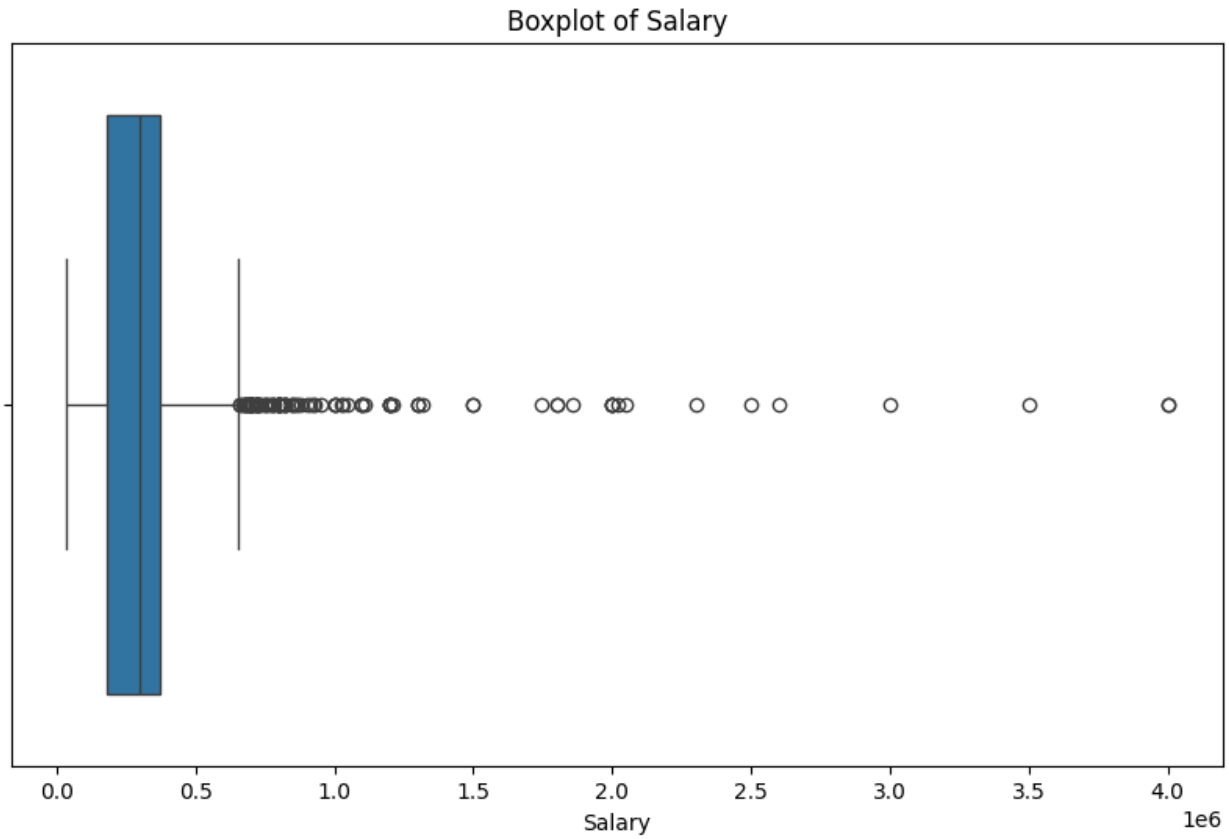
Step 3: Univariate Analysis

```
# Histogram for Salary
plt.figure(figsize=(10, 6))
sns.histplot(data['Salary'], kde=True)
plt.title('Distribution of Salary')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.show()
```



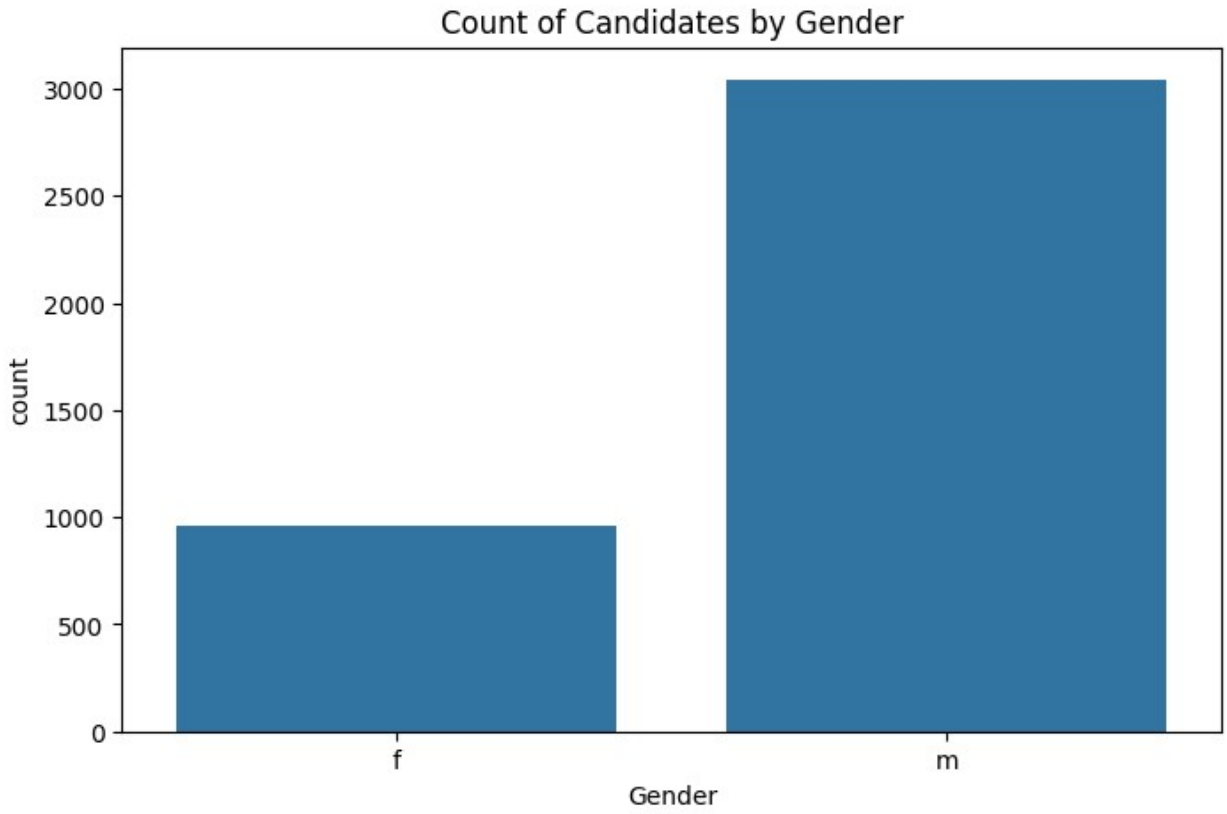

```
# Boxplot for Salary to check for outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Salary'])
plt.title('Boxplot of Salary')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed
in a future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```



3.2 Countplot for Gender

```
# Countplot for Gender
plt.figure(figsize=(8, 5))
sns.countplot(x='Gender', data=data)
plt.title('Count of Candidates by Gender')
plt.show()
```



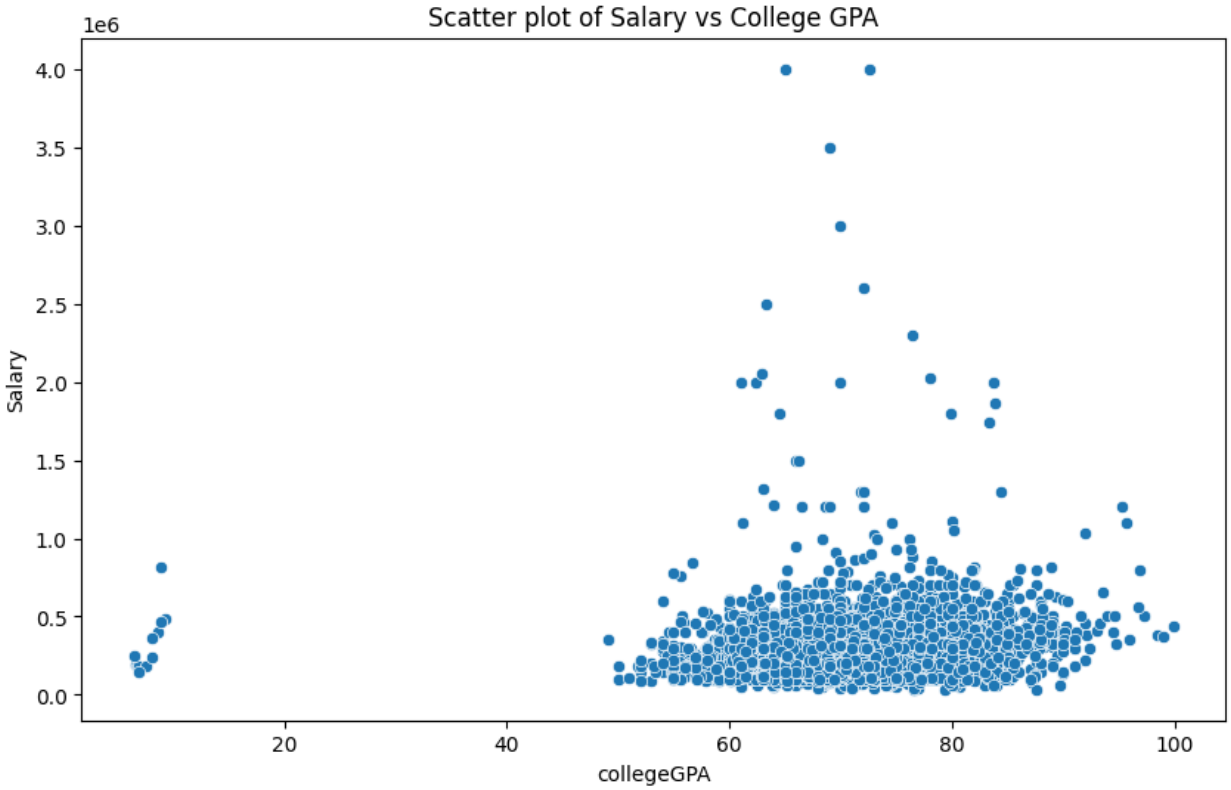
Step 4: Bivariate Analysis

```
# Check the column names in the dataset
print(data.columns)
```

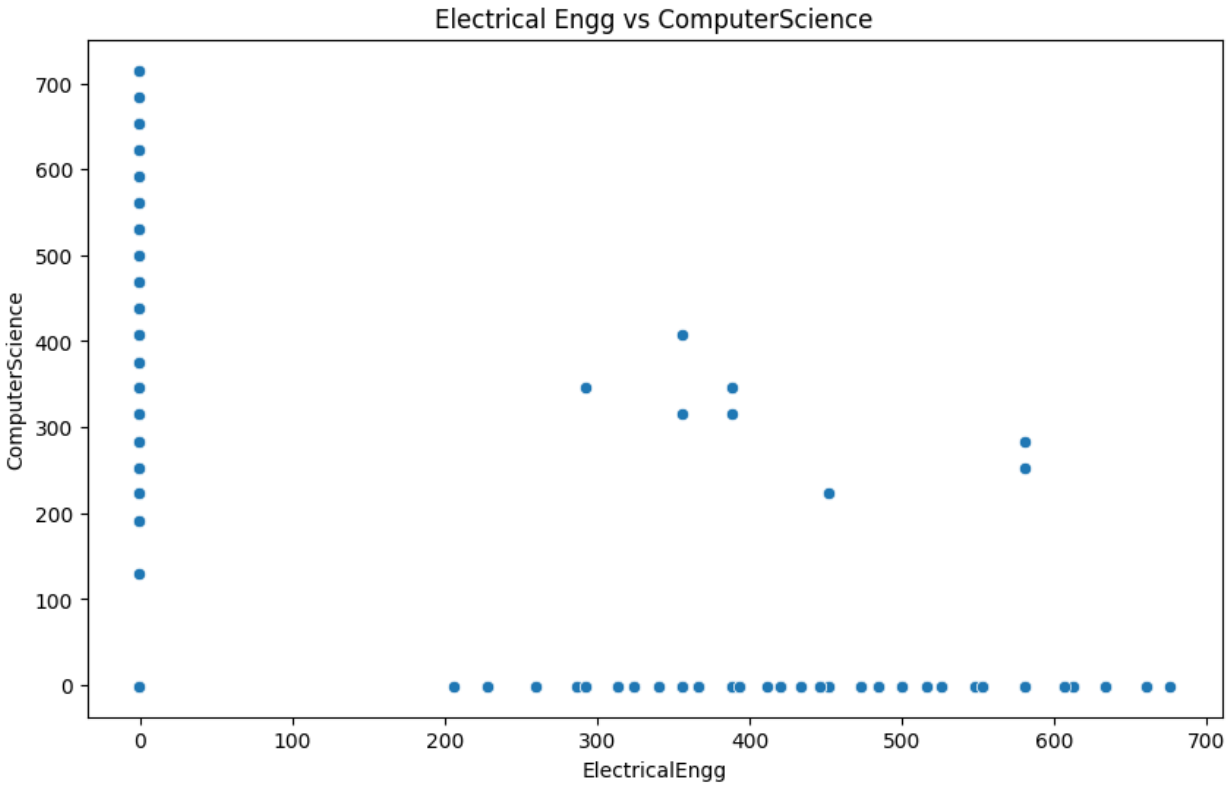
```
Index(['Unnamed: 0', 'ID', 'Salary', 'DOJ', 'DOL', 'Designation',
'JobCity',
       'Gender', 'DOB', '10percentage', '10board', '12graduation',
       '12percentage', '12board', 'CollegeID', 'CollegeTier',
'Degree',
       'Specialization', 'collegeGPA', 'CollegeCityID',
'CollegeCityTier',
       'CollegeState', 'GraduationYear', 'English', 'Logical',
'Quant',
       'Domain', 'ComputerProgramming', 'ElectronicsAndSemicon',
       'ComputerScience', 'MechanicalEngg', 'ElectricalEngg',
'TelecomEngg',
       'CivilEngg', 'conscientiousness', 'agreeableness',
'extraversion',
       'nueroticism', 'openess_to_experience'],
      dtype='object')
```

```
# Scatter plot to check relationship between College GPA and Salary
plt.figure(figsize=(10, 6))
sns.scatterplot(x='collegeGPA', y='Salary', data=data)
```

```
plt.title('Scatter plot of Salary vs College GPA')
plt.show()
```



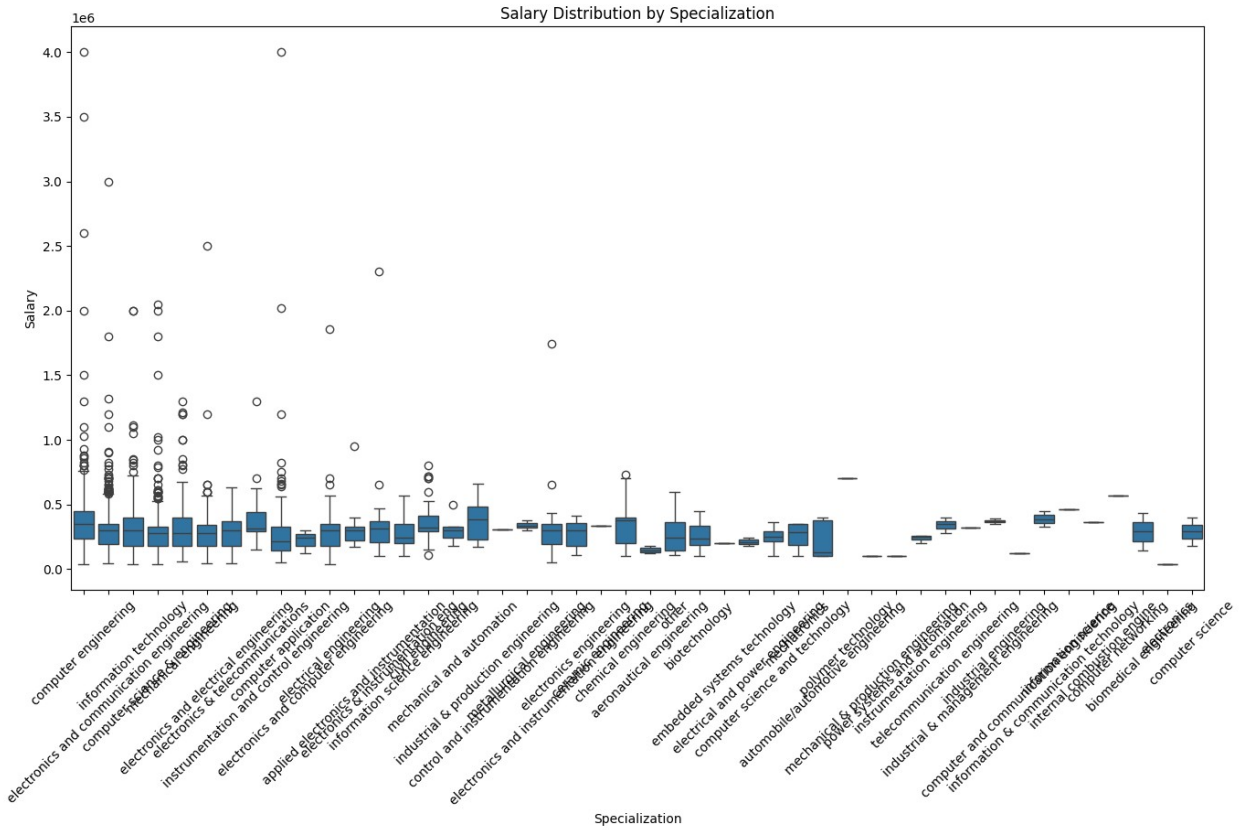

```
# Scatter plot to check relationship between College GPA and Salary
plt.figure(figsize=(10, 6))
sns.scatterplot(x='ElectricalEngg', y='ComputerScience', data=data)
plt.title('Electrical Engg vs ComputerScience')
plt.show()
```



Histogram

```
# Boxplot to visualize how Salary varies across Specializations
plt.figure(figsize=(16, 8))
sns.boxplot(x='Specialization', y='Salary', data=data)
plt.title('Salary Distribution by Specialization')
plt.xticks(rotation=45)
plt.show()
```

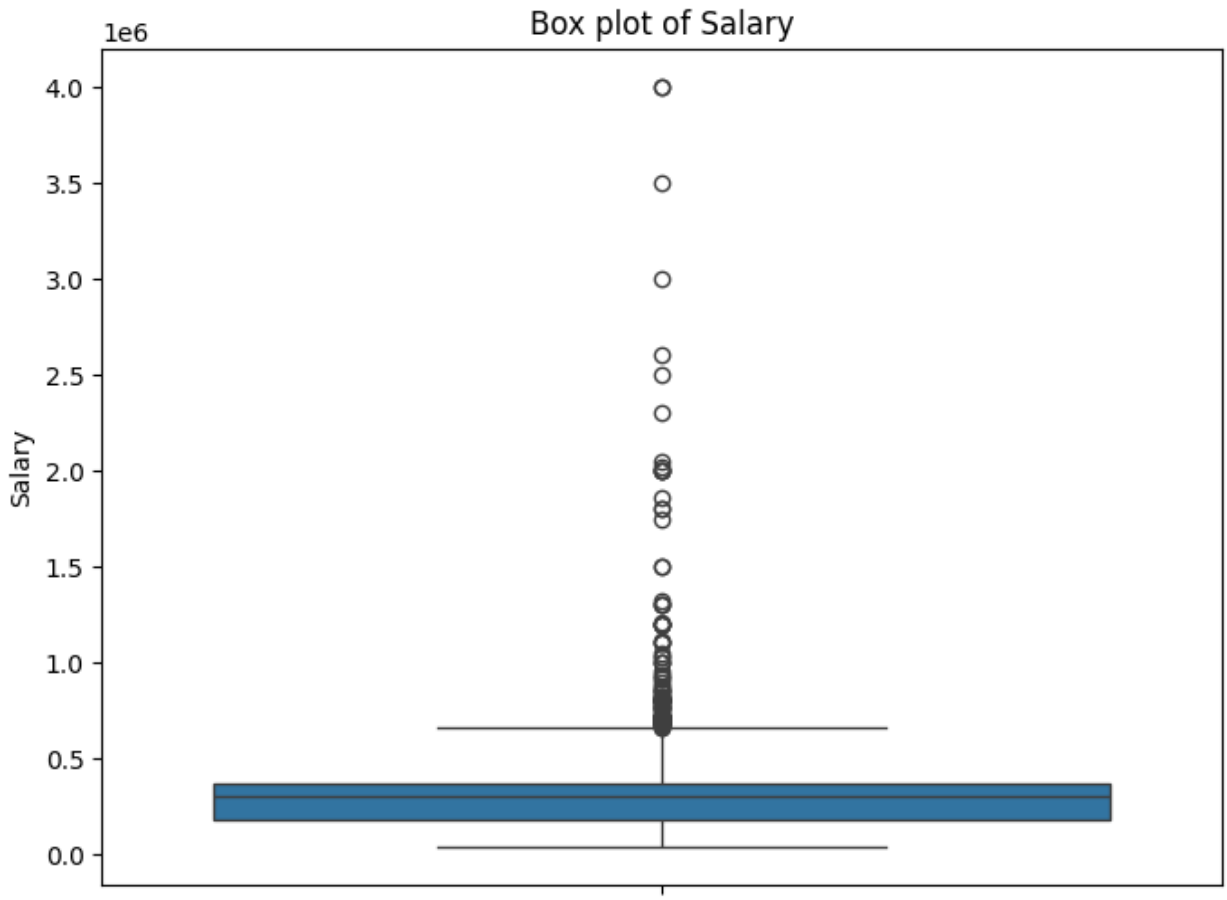
```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed
in a future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```



Box Plot

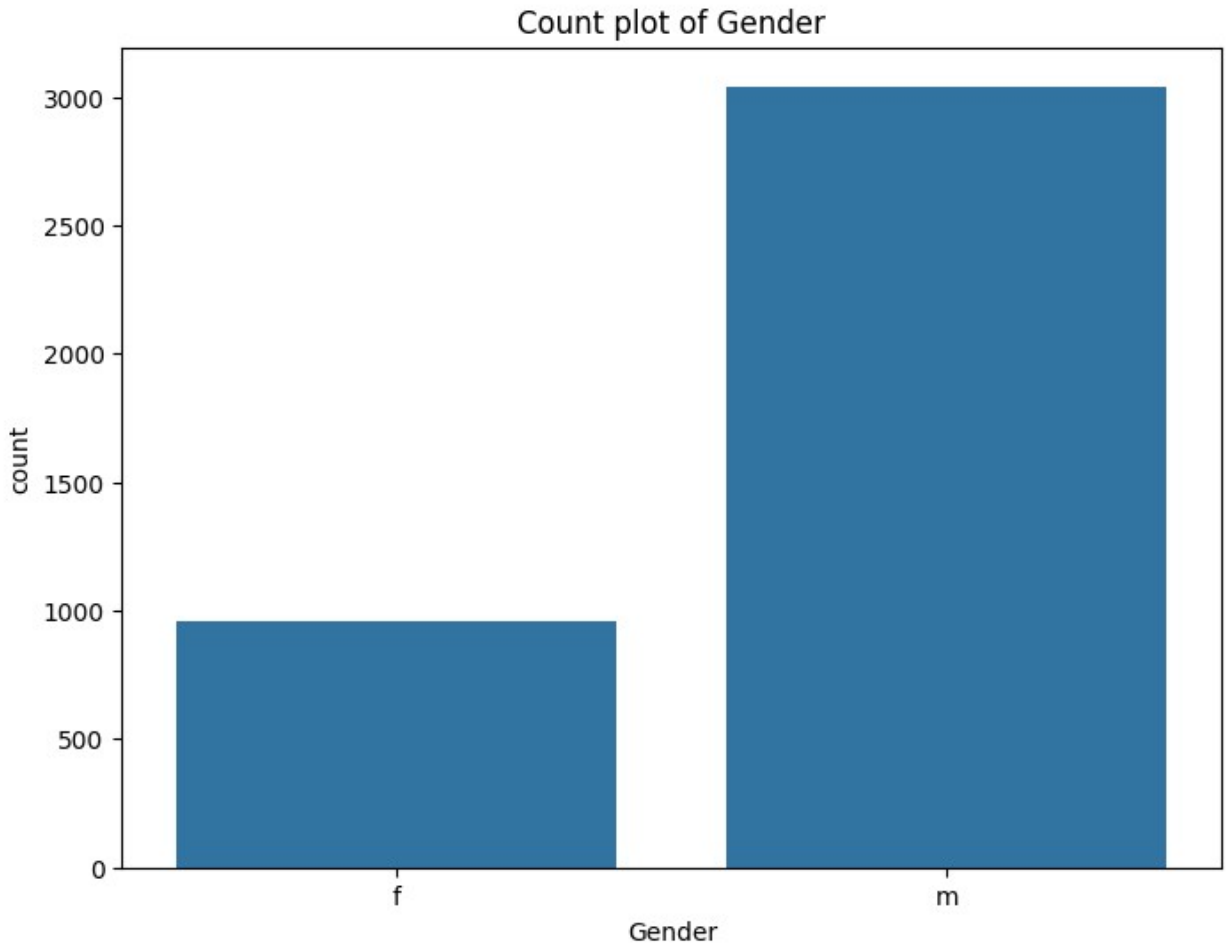
```
# Box plot for Salary
plt.figure(figsize=(8, 6))
sns.boxplot(y='Salary', data=data)
plt.title('Box plot of Salary')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed
in a future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```

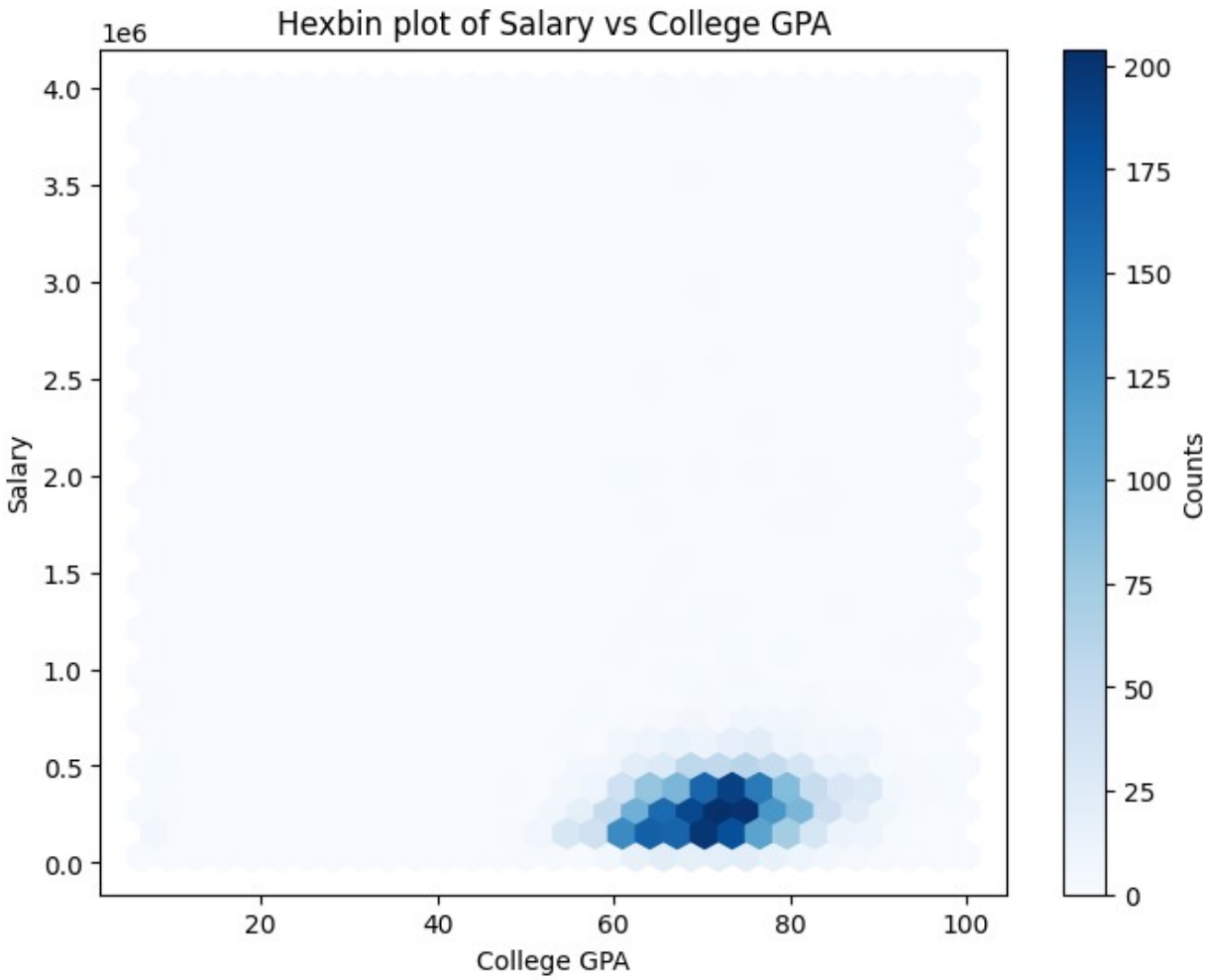


Count Plot

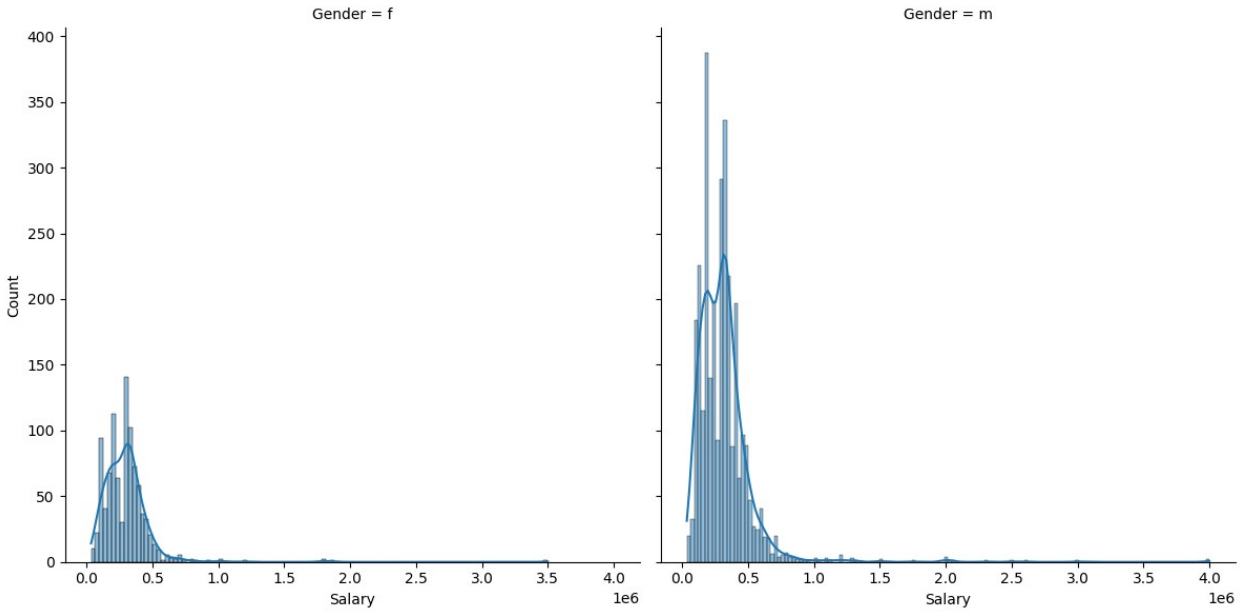
```
# Count plot for Gender

plt.figure(figsize=(8, 6))
sns.countplot(x='Gender', data=data)
plt.title('Count plot of Gender')
plt.show()
```



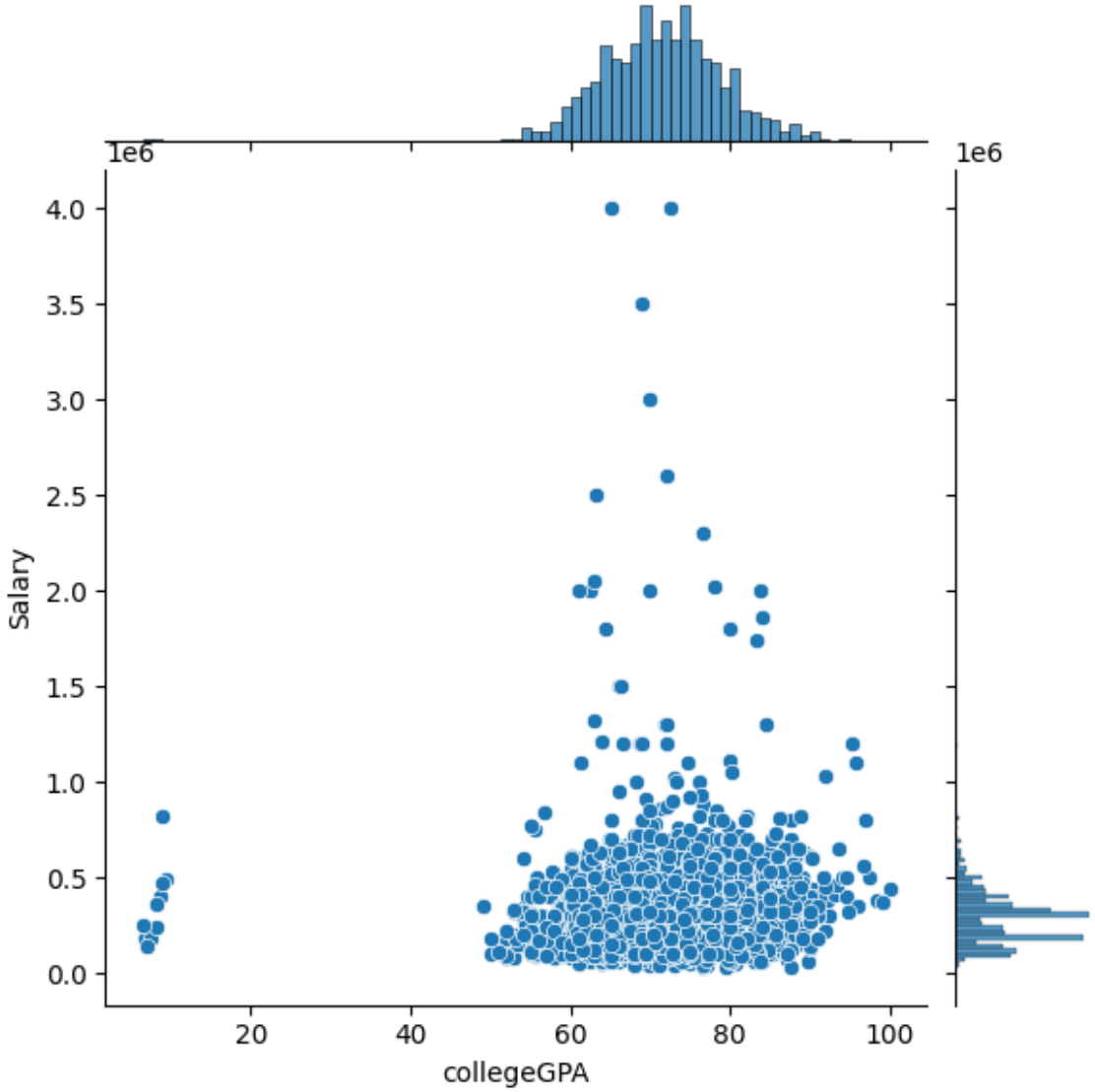

```
# Hexbin plot for College GPA vs Salary
plt.figure(figsize=(8, 6))
plt.hexbin(data['collegeGPA'], data['Salary'], gridsize=30,
cmap='Blues')
plt.colorbar(label='Counts')
plt.title('Hexbin plot of Salary vs College GPA')
plt.xlabel('College GPA')
plt.ylabel('Salary')
plt.show()
```




```
# FacetGrid for Salary distribution by Gender
g = sns.FacetGrid(data, col='Gender', height=6, aspect=1)
g.map(sns.histplot, 'Salary', kde=True)
plt.show()
```



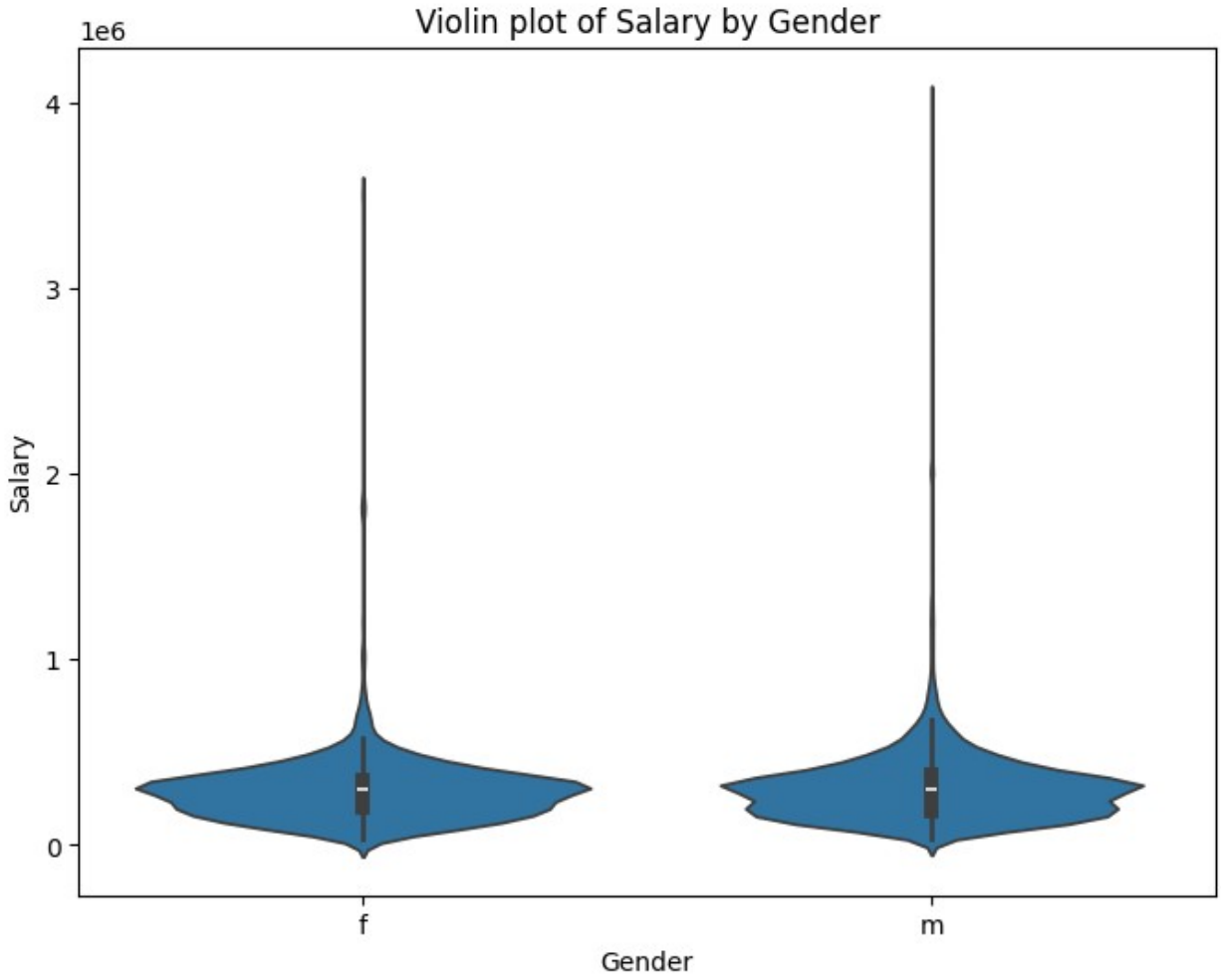

```
# Joint plot of College GPA vs Salary
sns.jointplot(x='collegeGPA', y='Salary', data=data, kind='scatter')
plt.show()
```



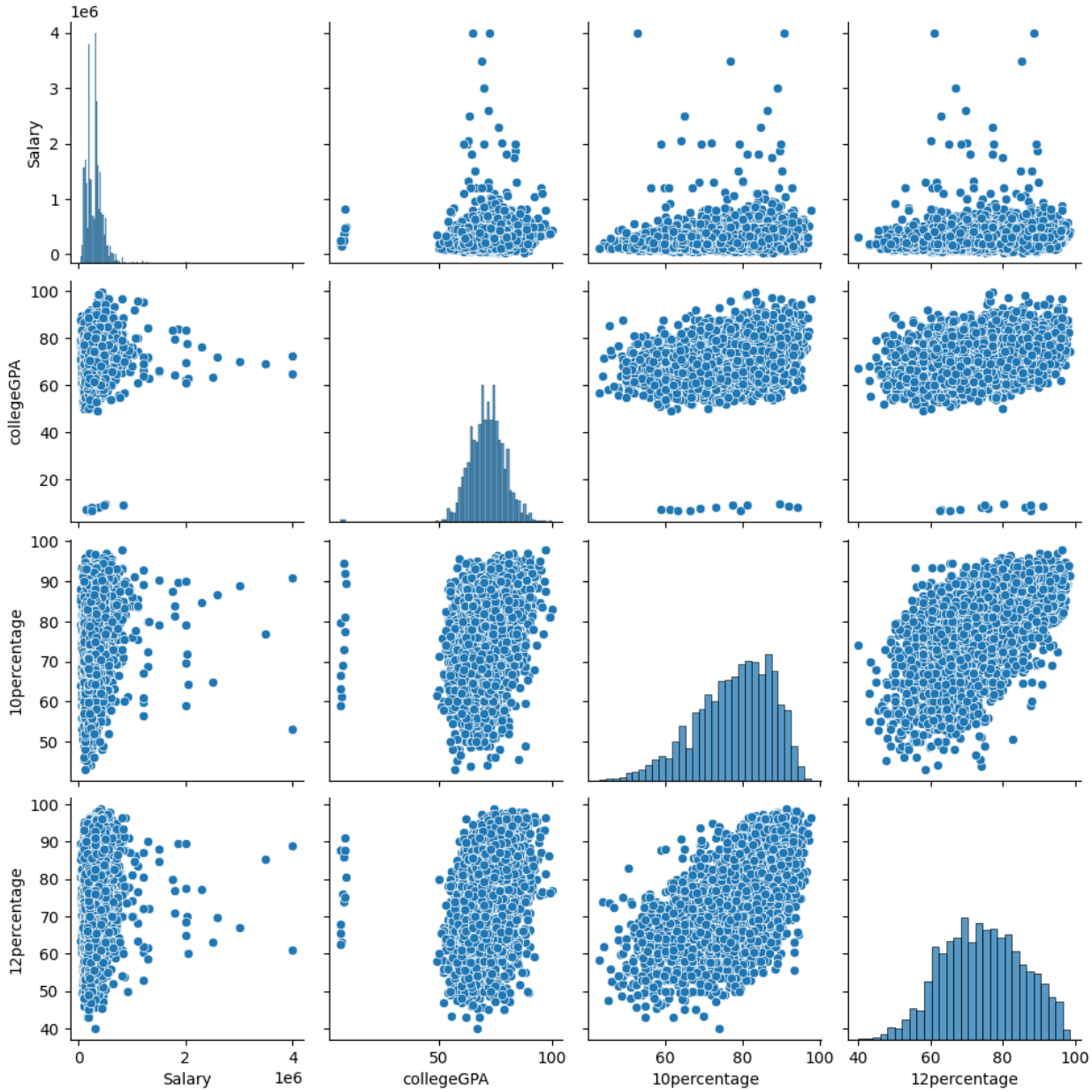

```
# Violin plot of Salary vs Gender
plt.figure(figsize=(8, 6))
sns.violinplot(x='Gender', y='Salary', data=data)
plt.title('Violin plot of Salary by Gender')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
```

```
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```



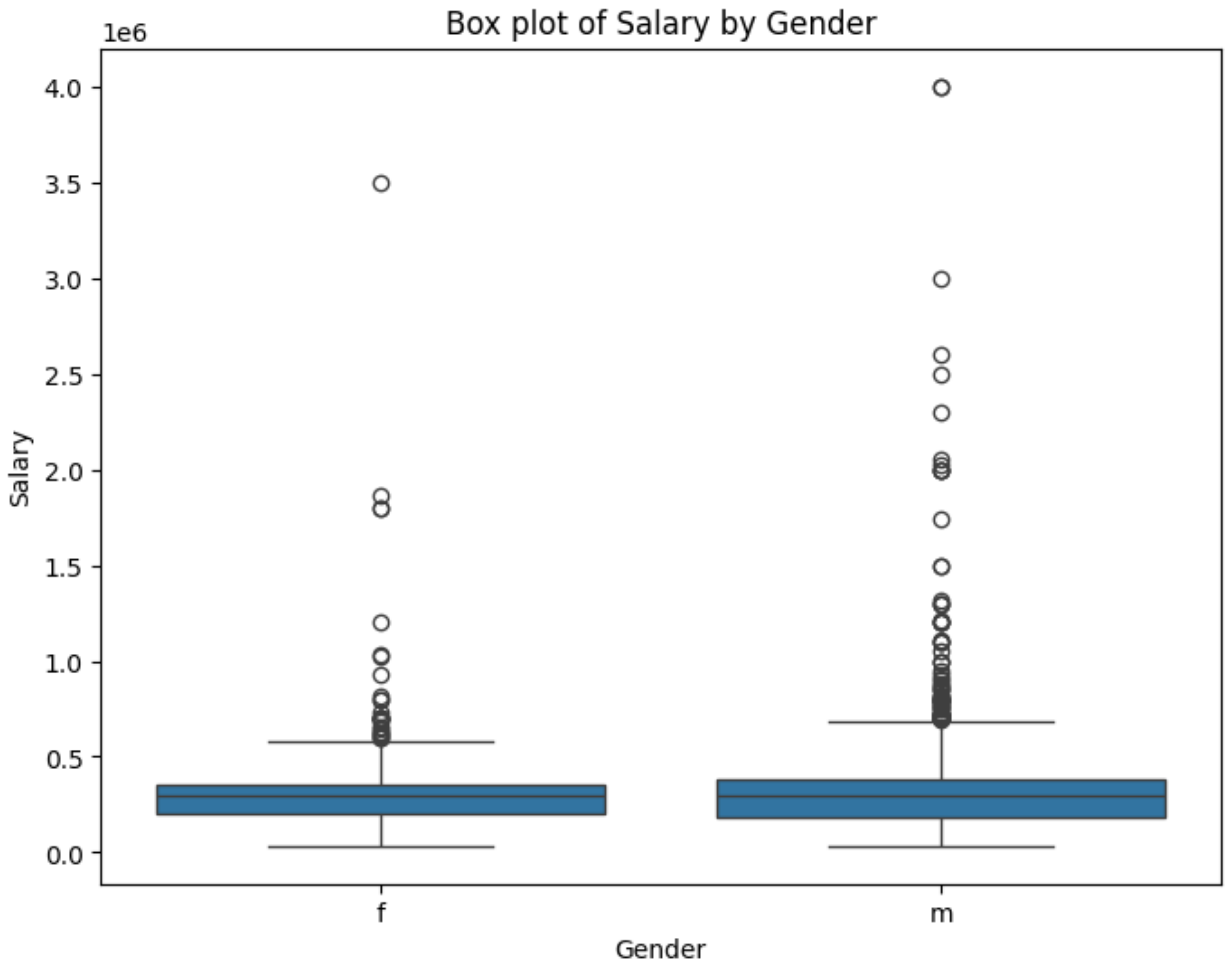

```
# Pair plot for selected numerical columns
sns.pairplot(data[['Salary', 'collegeGPA', '10percentage',
'12percentage']])
plt.show()
```



Categorical-Numerical Relationships

```
# Box plot of Salary by Gender
plt.figure(figsize=(8, 6))
sns.boxplot(x='Gender', y='Salary', data=data)
plt.title('Box plot of Salary by Gender')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:640:
FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed
in a future version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```



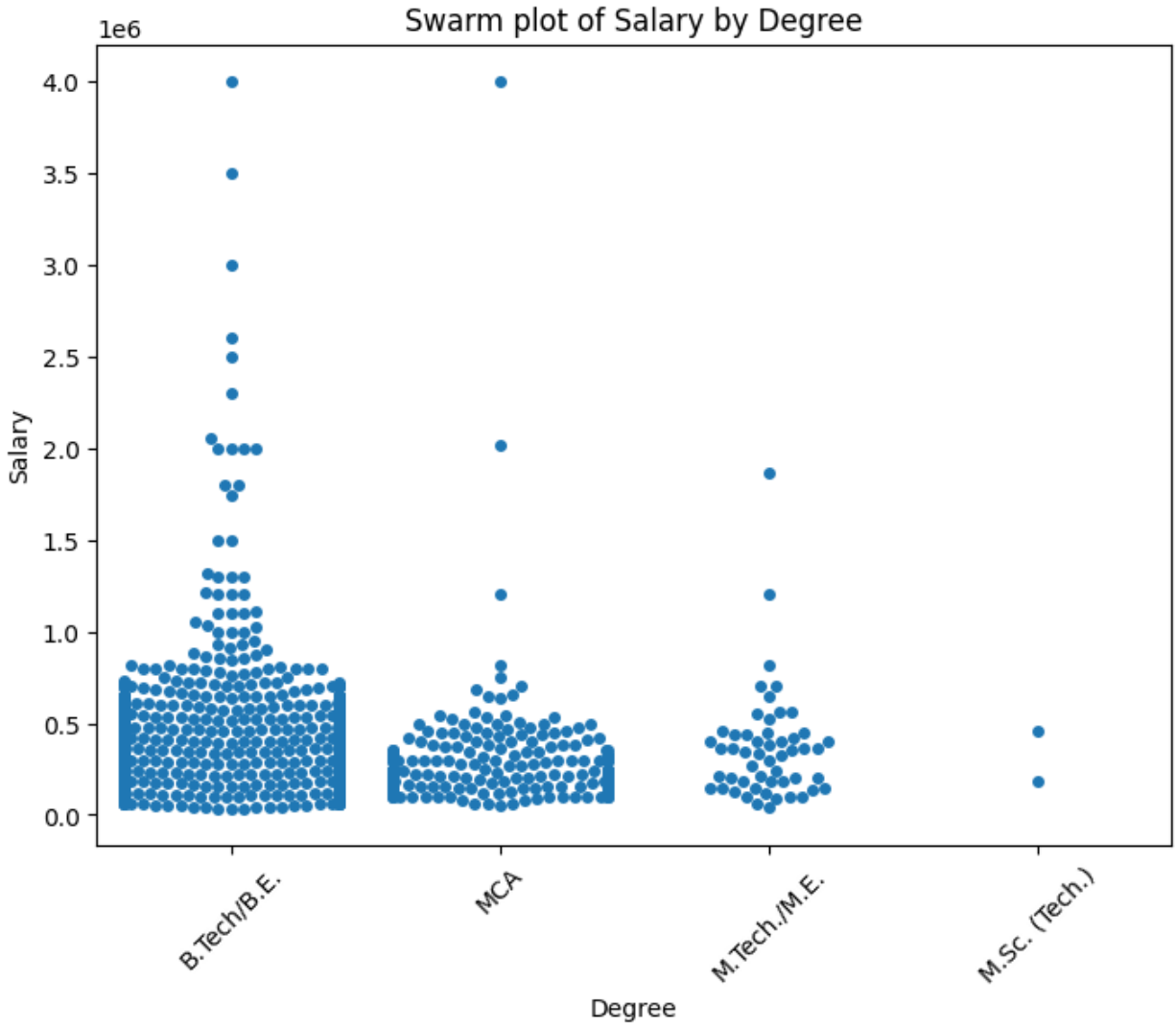
Swarm Plot:

Swarm plots help visualize the spread of numerical data points across categorical variables without overlapping.

```
# Swarm plot of Salary by Degree
plt.figure(figsize=(8, 6))
sns.swarmplot(x='Degree', y='Salary', data=data)
plt.title('Swarm plot of Salary by Degree')
plt.xticks(rotation=45)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 91.9% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
```

```
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 42.0% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 93.0% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 48.6% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



Multiple Scatter Plots in Subplots

```
# Importing necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns

# List of numerical columns to plot against Salary
numerical_columns = ['collegeGPA', '10percentage', '12percentage',
'English', 'Logical', 'Quant']

# Creating subplots
plt.figure(figsize=(15, 12))

# Loop over the numerical columns and create scatter plots
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(3, 2, i)  # Adjust the grid (3 rows, 2 columns)
    sns.scatterplot(x=col, y='Salary', data=data,
color=sns.color_palette()[i % 6])  # Using different colors
    plt.title(f'Salary vs {col}')
    plt.xlabel(col)
    plt.ylabel('Salary')

# Adjust layout for clarity
plt.tight_layout()
plt.show()
```

Step 5: Research Questions

5.1 Testing Salary Claim (2.5-3 Lakhs for Certain Job Titles)

```
# Testing the claim for specific job titles
job_titles = ['Programming Analyst', 'Software Engineer', 'Hardware
Engineer', 'Associate Engineer']
salaries = data[data['Designation'].isin(job_titles)]['Salary']

# Descriptive statistics for the salary of selected job titles
salaries.describe()
```

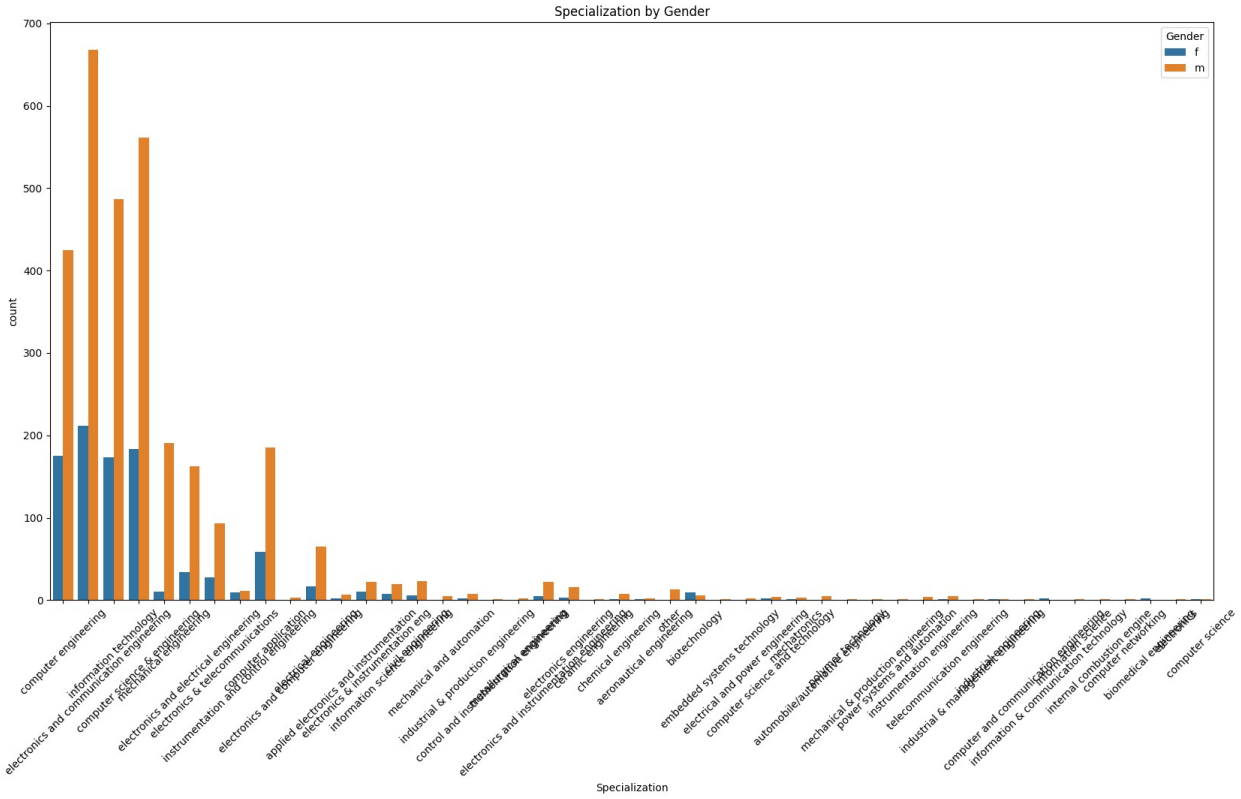
```
count    0.0
mean     NaN
std      NaN
min      NaN
25%      NaN
50%      NaN
75%      NaN
```

```
max      NaN
Name: Salary, dtype: float64
```

5.2 Analyzing Relationship Between Gender and Specialization

```
# Countplot to analyze specialization preferences by gender
plt.figure(figsize=(20, 10))
sns.countplot(x='Specialization', hue='Gender', data=data)
plt.title('Specialization by Gender')
plt.xticks(rotation=45)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```



Step 6: Conclusion

In this section, summarize the key insights and conclusions derived from the exploratory data analysis. Below is an example structure of the conclusion based on the steps you've followed:

Salary Distribution: The salary data is right-skewed, with most candidates earning below the median salary and a few earning significantly higher, indicating some outliers.

Key Insights:

College GPA vs Salary: A weak positive correlation was observed, suggesting higher GPAs do not necessarily lead to significantly higher salaries. Specialization: Graduates from technical fields, such as Computer Science and Electronics, tend to earn higher salaries compared to other specializations. Research Questions:

Salary Claim: Fresh graduates in roles like Software Engineer and Programming Analyst generally earn between 2.5-3 lakhs, as claimed. Gender and Specialization: Both genders tend to choose similar specializations, with a slight male preference for mechanical fields. Final Insight: Strong technical skills, particularly in programming, are key factors in securing higher salaries for engineering graduates.

This concise summary highlights the main findings from the exploratory data analysis.