jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[43]:
```
# Step 1: Introduction
# The dataset involves details such as salary, educational background, demographics, and job-related variables. The main objective is to:

# Explore data patterns and relationships.
# Answer specific research questions like salary trends and gender preference in specializations.
# The target variable is Salary.
```

[41]:
```
#Perform Exploratory Data Analysis (EDA) on the AMCAT Dataset.
```

[ ]:
```
# Step 2: Importing Data
# Start by importing necessary libraries and loading the dataset.
```

[20]:
```
import pandas as pd
import numpy as np
import seaborn as sn
import matplotlib.pyplot as plt
```

[26]:
```
df = pd.read_csv('data.xlsx - Sheet1.csv')
```

[29]:
```
df.head()
```

[29]:

	Unnamed: 0	ID	Salary	DOJ	DOL	Designation	JobCity	Gender	DOB	10percentage	...	ComputerScience	MechanicalEngg	ElectricalEngg	Telecom
0	train	203097	420000.0	6/1/12 0:00	present	senior quality engineer	Bangalore	f	2/19/90 0:00	84.3	...	-1	-1	-1	
1	train	579905	500000.0	9/1/13 0:00	present	assistant manager	Indore	m	10/4/89 0:00	85.4	...	-1	-1	-1	

2	train	810601	325000.0	6/1/14 0:00	present	systems engineer	Chennai	f	8/3/92 0:00	85.0	...	-1	-1	-1
3	train	267447	1100000.0	7/1/11 0:00	present	senior software engineer	Gurgaon	m	12/5/89 0:00	85.6	...	-1	-1	-1
4	train	343523	200000.0	3/1/14 0:00	3/1/15 0:00	get	Manesar	m	2/27/91 0:00	78.0	...	-1	-1	-1

5 rows × 39 columns

[45]:
```
df.shape
```

[45]:
```
(3998, 39)
```

[47]:
```
df.describe()
```

[47]:

	ID	Salary	10percentage	12graduation	12percentage	CollegeID	CollegeTier	collegeGPA	CollegeCityID	CollegeCityTier	...	ComputerScie
count	3.998000e+03	3.998000e+03	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	3998.000000	...	3998.000
mean	6.637945e+05	3.076998e+05	77.925443	2008.087544	74.466366	5156.851426	1.925713	71.486171	5156.851426	0.300400	...	90.742
std	3.632182e+05	2.127375e+05	9.850162	1.653599	10.999933	4802.261482	0.262270	8.167338	4802.261482	0.458489	...	175.273
min	1.124400e+04	3.500000e+04	43.000000	1995.000000	40.000000	2.000000	1.000000	6.450000	2.000000	0.000000	...	-1.000
25%	3.342842e+05	1.800000e+05	71.680000	2007.000000	66.000000	494.000000	2.000000	66.407500	494.000000	0.000000	...	-1.000
50%	6.396000e+05	3.000000e+05	79.150000	2008.000000	74.400000	3879.000000	2.000000	71.720000	3879.000000	0.000000	...	-1.000
75%	9.904800e+05	3.700000e+05	85.670000	2009.000000	82.600000	8818.000000	2.000000	76.327500	8818.000000	1.000000	...	-1.000
max	1.298275e+06	4.000000e+06	97.760000	2013.000000	98.700000	18409.000000	2.000000	99.930000	18409.000000	1.000000	...	715.000

jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code

[49]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3998 entries, 0 to 3997
Data columns (total 39 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             3998 non-null   object
 1   ID                     3998 non-null   int64
 2   Salary                 3998 non-null   float64
 3   DOJ                    3998 non-null   object
 4   DOL                    3998 non-null   object
 5   Designation            3998 non-null   object
 6   JobCity                3998 non-null   object
 7   Gender                 3998 non-null   object
 8   DOB                    3998 non-null   object
 9   10percentage           3998 non-null   float64
 10  10board                3998 non-null   object
 11  12graduation           3998 non-null   int64
 12  12percentage           3998 non-null   float64
 13  12board                3998 non-null   object
 14  CollegeID              3998 non-null   int64
 15  CollegeTier            3998 non-null   int64
 16  Degree                 3998 non-null   object
 17  Specialization         3998 non-null   object
 18  collegeGPA             3998 non-null   float64
 19  CollegeCityID          3998 non-null   int64
 20  CollegeCityTier        3998 non-null   int64
 21  CollegeState           3998 non-null   object
 22  GraduationYear         3998 non-null   int64
 23  English                3998 non-null   int64
 24  Logical                3998 non-null   int64
 25  Quant                  3998 non-null   int64
 26  Domain                 3998 non-null   float64
```

```
 21  CollegeState            3998 non-null   object
 22  GraduationYear          3998 non-null   int64
 23  English                 3998 non-null   int64
 24  Logical                 3998 non-null   int64
 25  Quant                   3998 non-null   int64
 26  Domain                  3998 non-null   float64
 27  ComputerProgramming     3998 non-null   int64
 28  ElectronicsAndSemicon   3998 non-null   int64
 29  ComputerScience         3998 non-null   int64
 30  MechanicalEngg          3998 non-null   int64
 31  ElectricalEngg          3998 non-null   int64
 32  TelecomEngg             3998 non-null   int64
 33  CivilEngg               3998 non-null   int64
 34  conscientiousness       3998 non-null   float64
 35  agreeableness           3998 non-null   float64
 36  extraversion            3998 non-null   float64
 37  nueroticism             3998 non-null   float64
 38  openess_to_experience   3998 non-null   float64
dtypes: float64(10), int64(17), object(12)
memory usage: 1.2+ MB
```

[51]:
```
# Step 3: Univariate Analysis
# A. Numerical Columns:
# For numerical columns like Salary, 10th percentage, 12th percentage, and College GPA:

# Plot histograms, boxplots to detect outliers.
# Check probability and frequency distribution.
```

[62]:
```
# Plotting histogram for numerical column
numerical_columns = df.select_dtypes(include=['int64', 'float64'])
numerical_columns.hist(figsize=(16, 14), bins=15, layout=(7, 4))

plt.tight_layout()
plt.show()
```

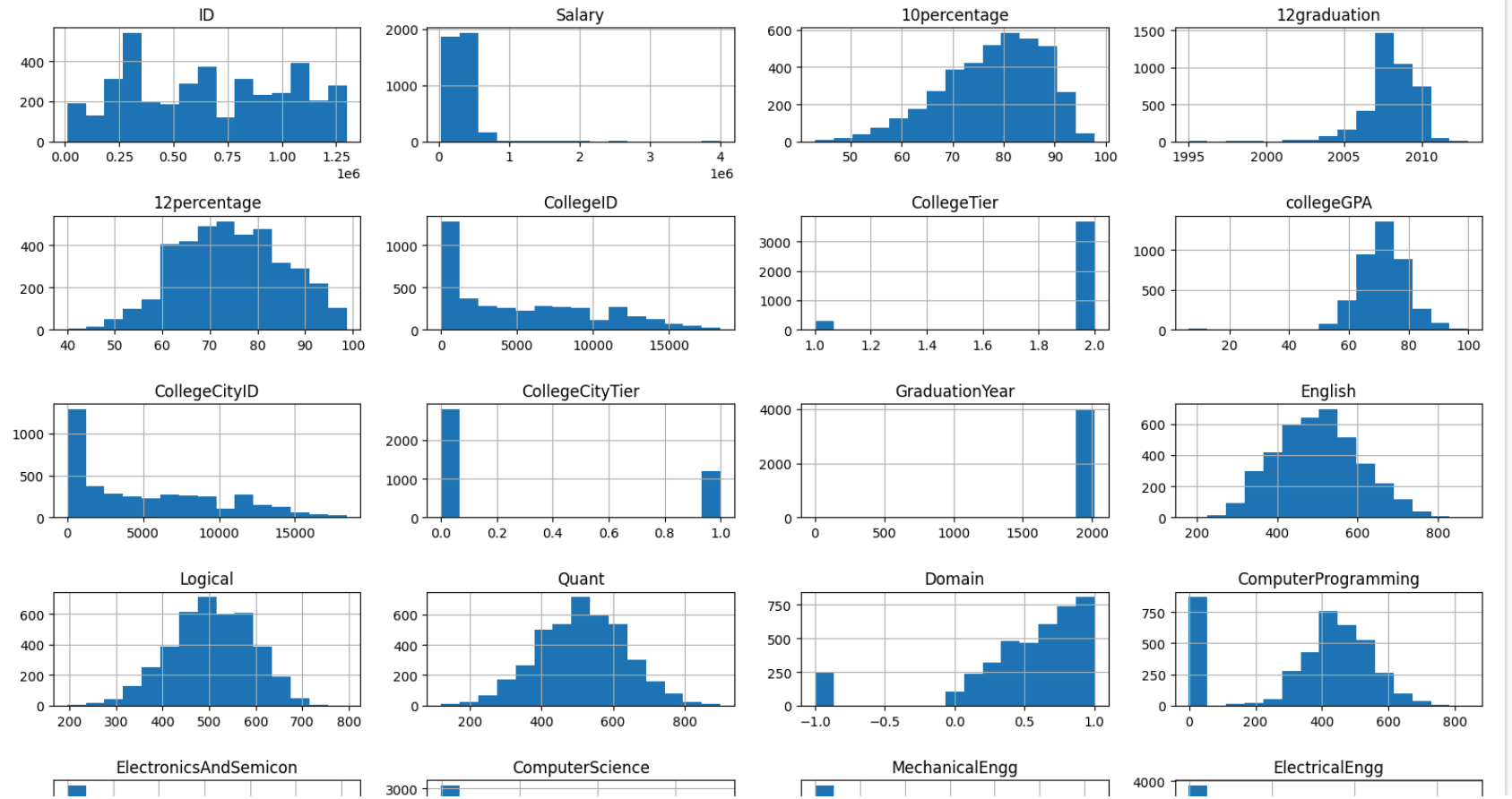
jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

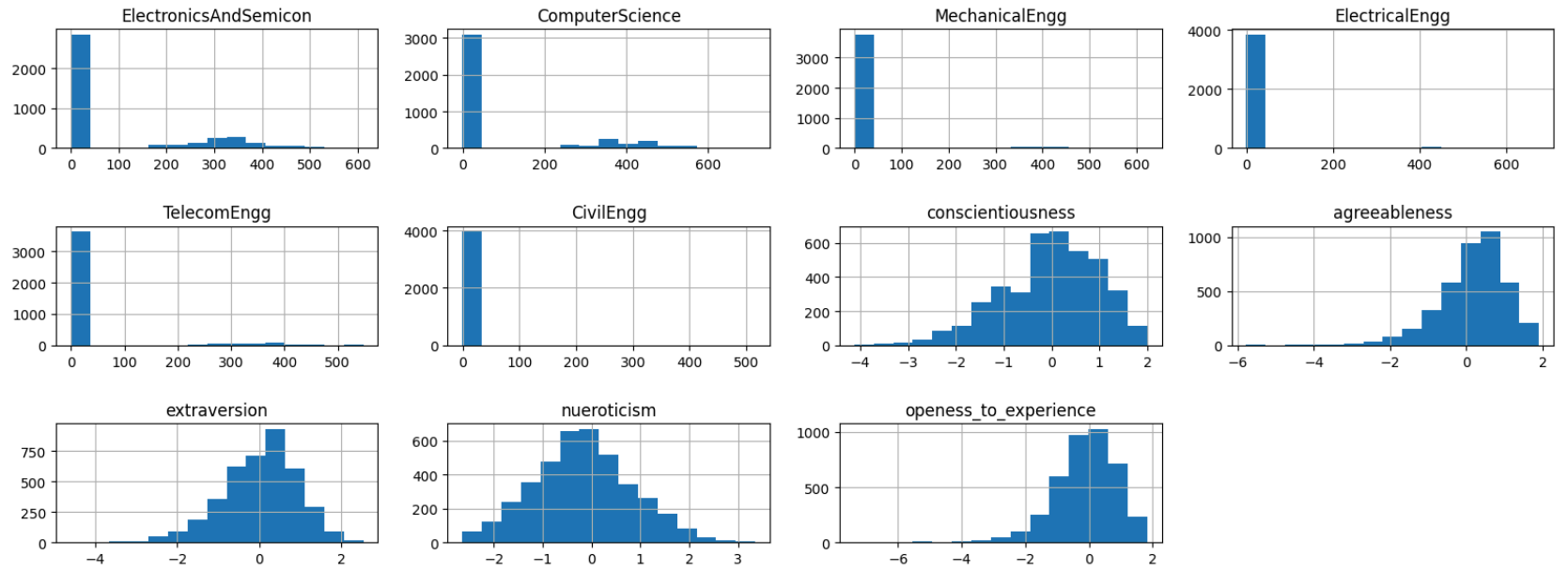
File Edit View Run Kernel Settings Help

Trusted

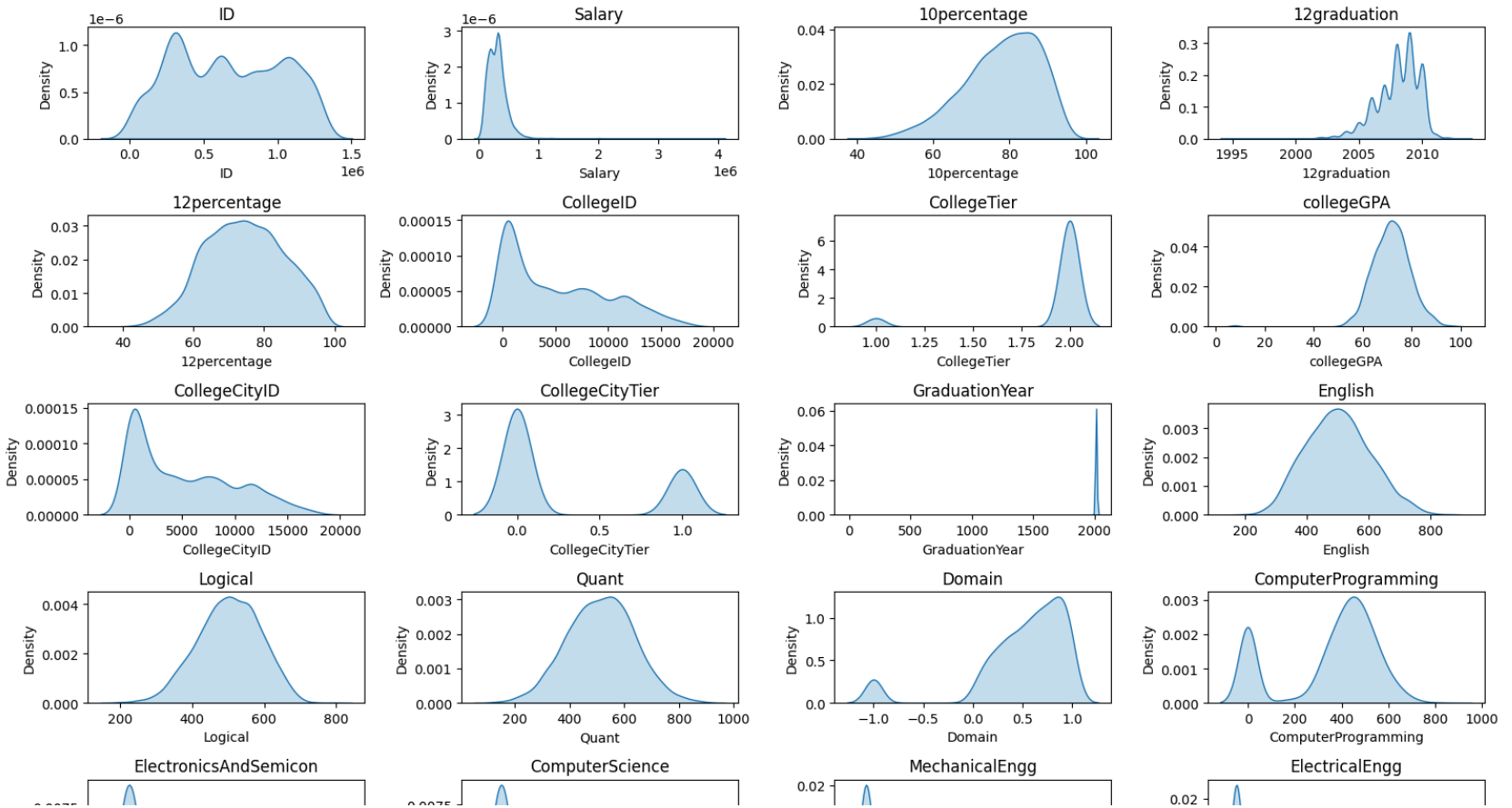
Code

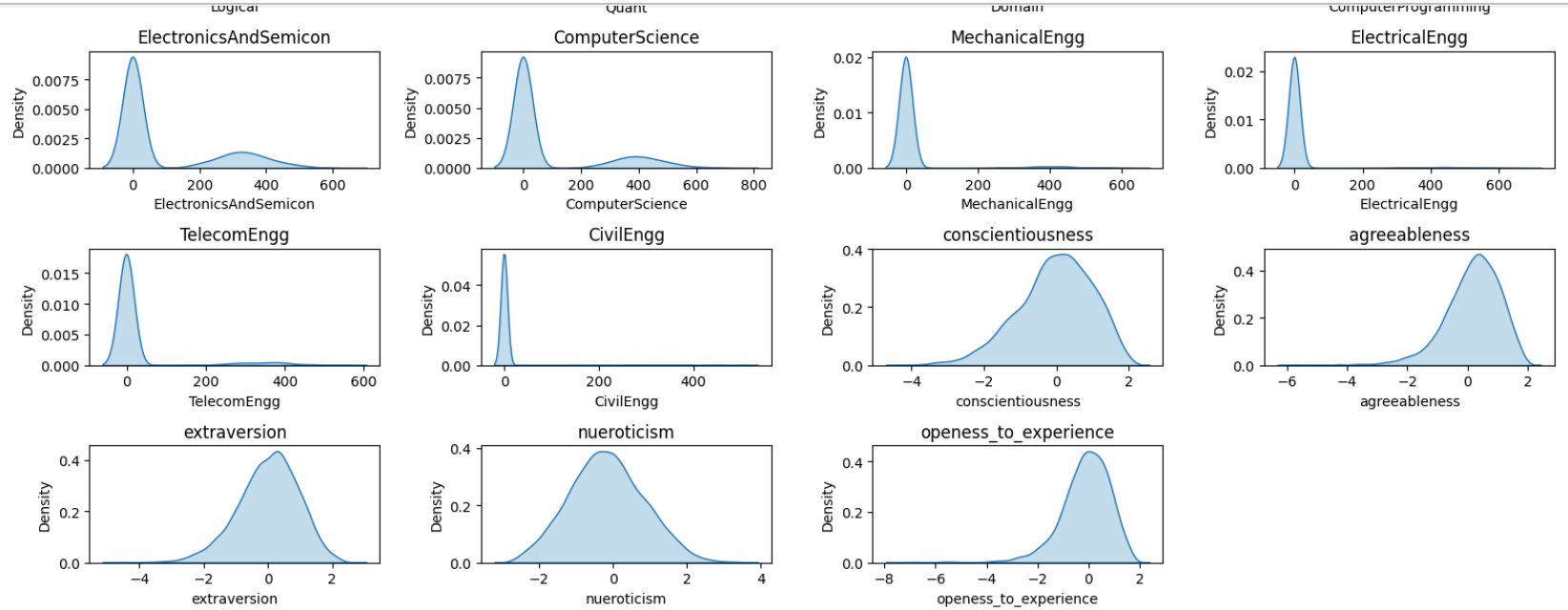
JupyterLab Python 3 (ipykernel)



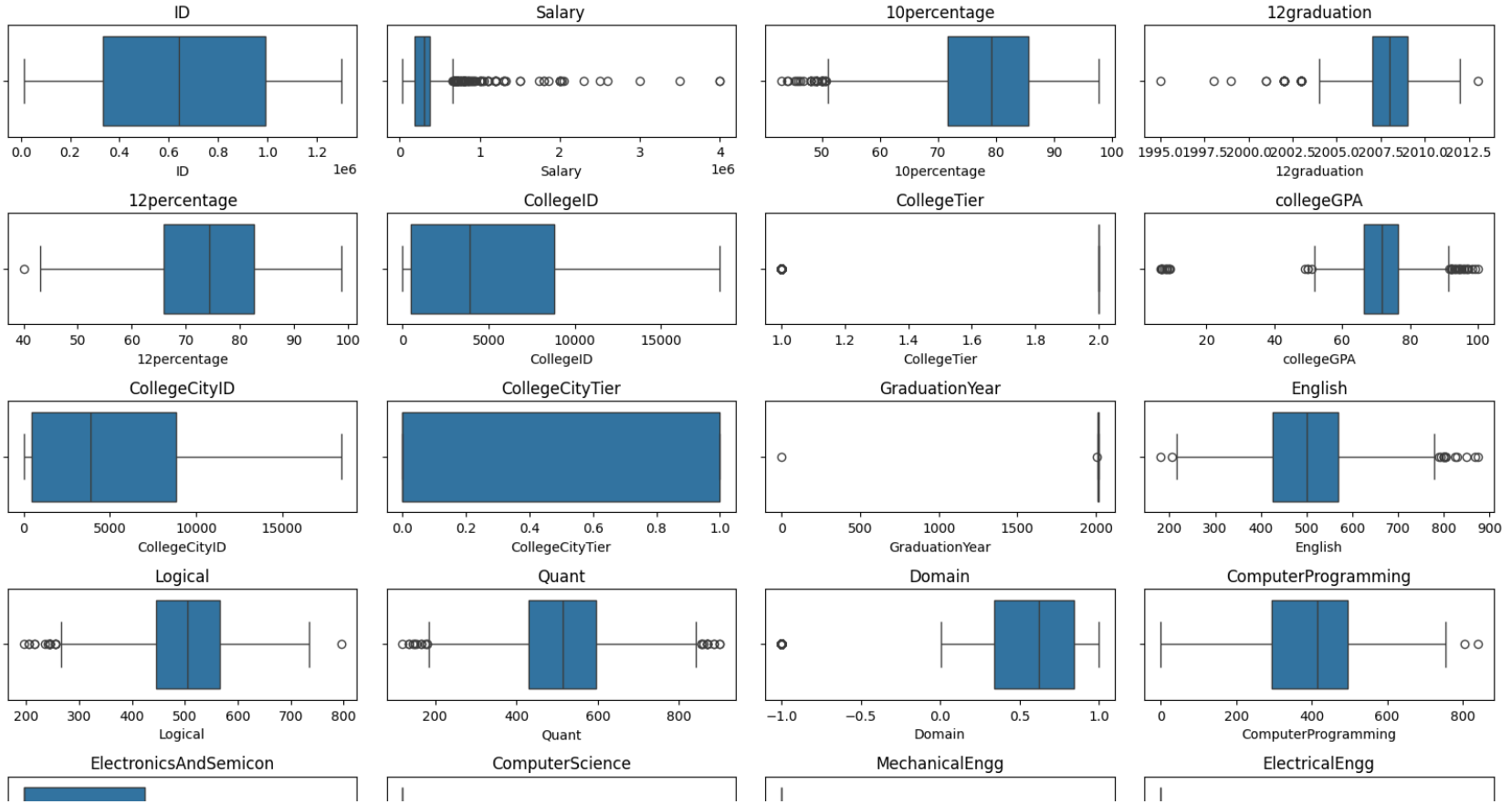


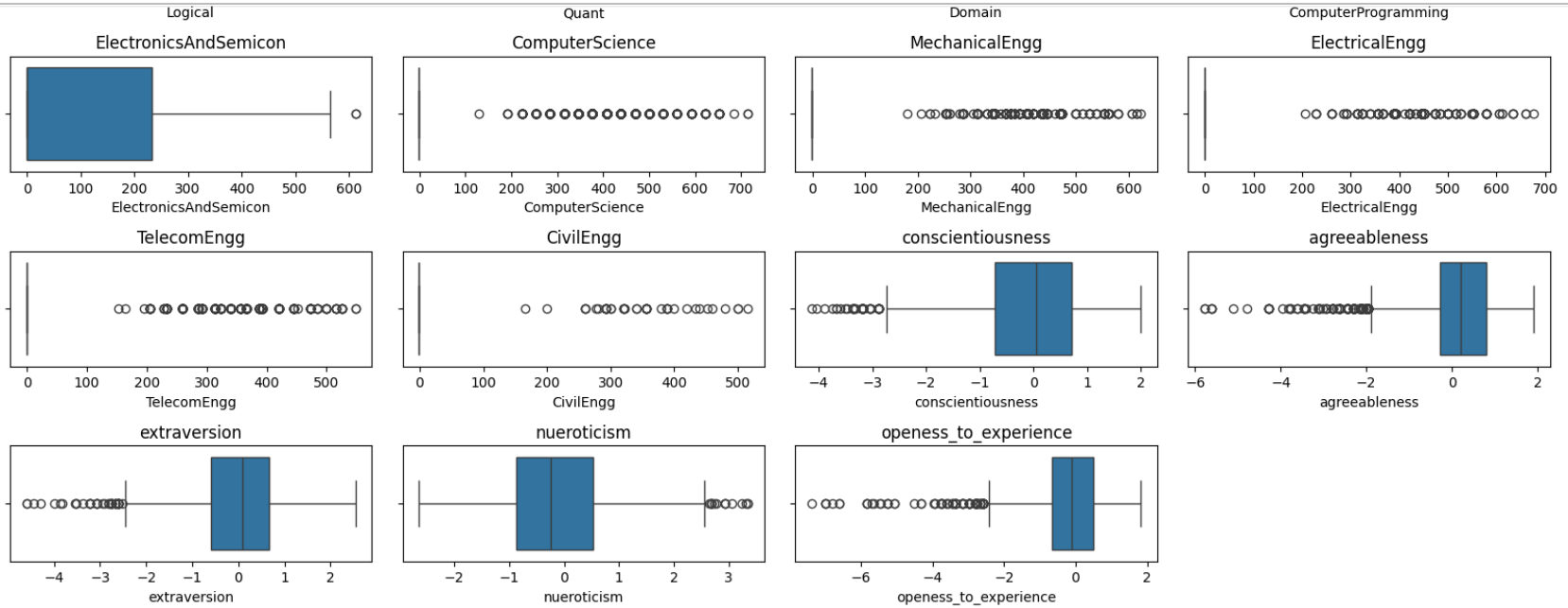
[69]:
```
categorical_columns = df.select_dtypes(include=object).columns
numerical_columns = [col for col in df.columns if col not in categorical_columns]
plt.figure(figsize=(16, 14))
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(7, 4, i)  # Adjust layout to fit all columns
    sns.kdeplot(df[col], fill=True)
    plt.title(col)
plt.tight_layout()
plt.show()
```


ID
Salary
10percentage
12graduation
12percentage
CollegeID
CollegeTier
collegeGPA
CollegeCityID
CollegeCityTier
GraduationYear
English
Logical
Quant
Domain
ComputerProgramming
ElectronicsAndSemicon
ComputerScience
MechanicalEngg
ElectricalEngg
Density



[71]:
```
# Plotting BOX Plot for numerical columns
plt.figure(figsize=(16, 14))
for i, col in enumerate(numerical_columns, 1):
    plt.subplot(7, 4, i)  # Adjust layout to fit all columns
    sns.boxplot(x=df[col])
    plt.title(col)
plt.tight_layout()
plt.show()
```


ID
0.0 0.2 0.4 0.6 0.8 1.0 1.2
ID
1e6
Salary
0 1 2 3 4
Salary
1e6
10percentage
50 60 70 80 90 100
10percentage
12graduation
1995.0 1997.5 2000.0 2002.5 2005.0 2007.5 2010.0 2012.5
12graduation
12percentage
40 50 60 70 80 90 100
12percentage
CollegeID
0 5000 10000 15000
CollegeID
CollegeTier
1.0 1.2 1.4 1.6 1.8 2.0
CollegeTier
collegeGPA
20 40 60 80 100
collegeGPA
CollegeCityID
0 5000 10000 15000
CollegeCityID
CollegeCityTier
0.0 0.2 0.4 0.6 0.8 1.0
CollegeCityTier
GraduationYear
0 500 1000 1500 2000
GraduationYear
English
200 300 400 500 600 700 800 900
English
Logical
200 300 400 500 600 700 800
Logical
Quant
200 400 600 800
Quant
Domain
−1.0 −0.5 0.0 0.5 1.0
Domain
ComputerProgramming
0 200 400 600 800
ComputerProgramming
ElectronicsAndSemicon
ComputerScience
MechanicalEngg
ElectricalEngg



[ ]:
```
### Univariate Visual Analysis on Categorical Columns
```

[93]:
```
num_plots = len(categorical_columns)
num_cols = 3  # Adjust number of columns per figure (optional)
num_rows = int(np.ceil(num_plots / num_cols))  # Calculate required rows

# Plotting bar charts on categorical columns
fig_num = 1
```

jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

```
# Plotting bar charts on categorical columns
fig_num = 1
for i in range(0, num_plots, num_cols * num_rows):
    # Create a new figure with increased width
    plt.figure(figsize=(14, num_rows * 5))
    for j in range(min(num_cols * num_rows, len(categorical_columns) - i)):
        col_index = i + j
        if col_index >= len(categorical_columns):
            break  # Avoid creating empty subplots
        plt.subplot(num_rows, num_cols, j + 1)
        sns.countplot(y=df[categorical_columns[col_index]], order=df[categorical_columns[col_index]].value_counts().index)
        plt.title(f'Bar Plot for {categorical_columns[col_index]}', fontsize=14)  # Increase title font size
        plt.xticks(rotation=45)  # Optional: Rotate x-axis labels
        plt.xlabel('Categories', fontsize=12)  # Add x-axis label with adjusted font size
        plt.xticks(fontsize=10)  # Adjust x-axis tick labels font size

    # Adjust spacing between subplots
    plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.85, wspace=0.4, hspace=0.5)
    plt.suptitle(f'Figure {fig_num}', y=1.02, fontsize=14)  # Add supertitle with adjusted font size
    plt.show()
    fig_num += 1
```

Figure 1

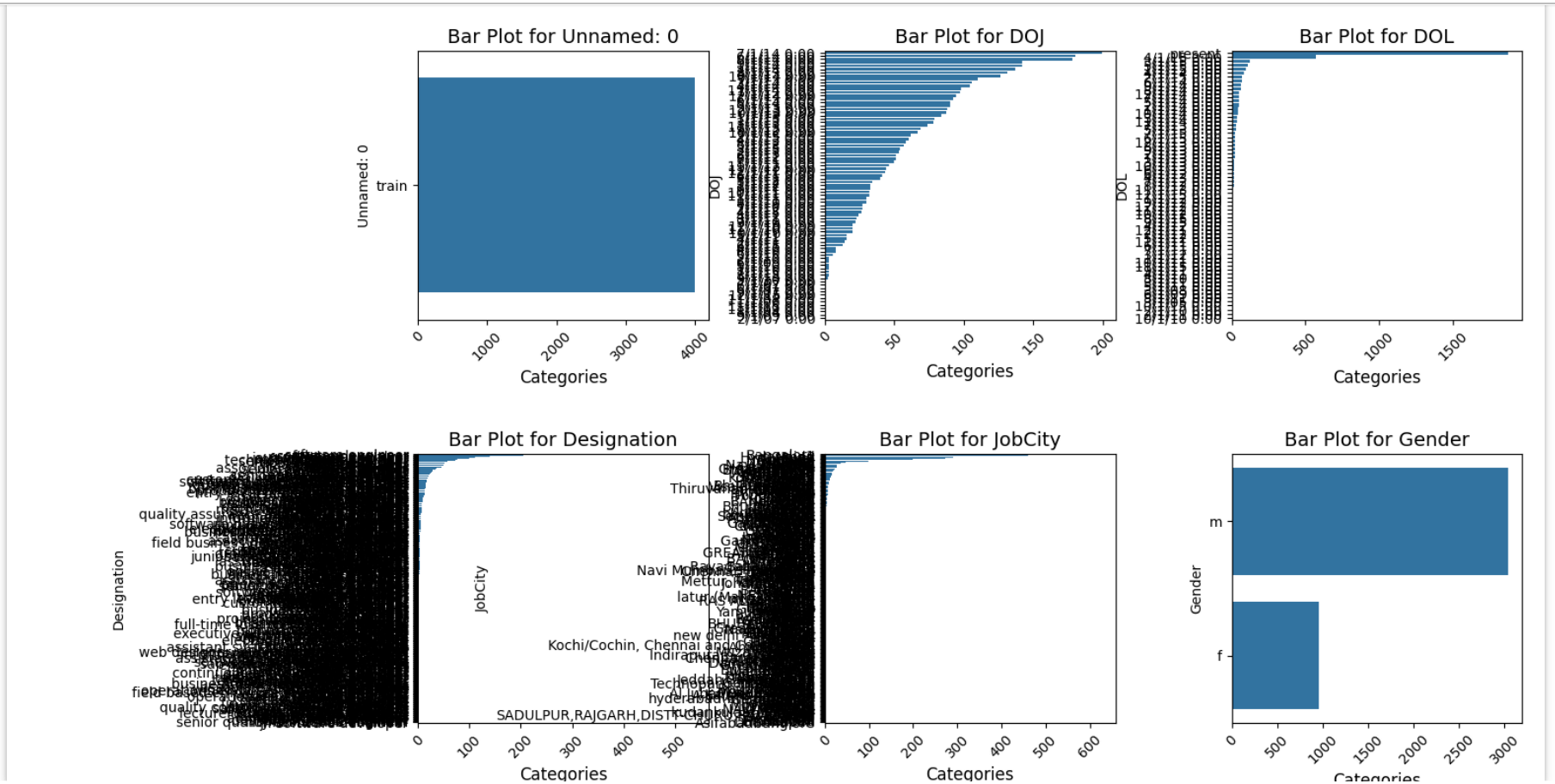
jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

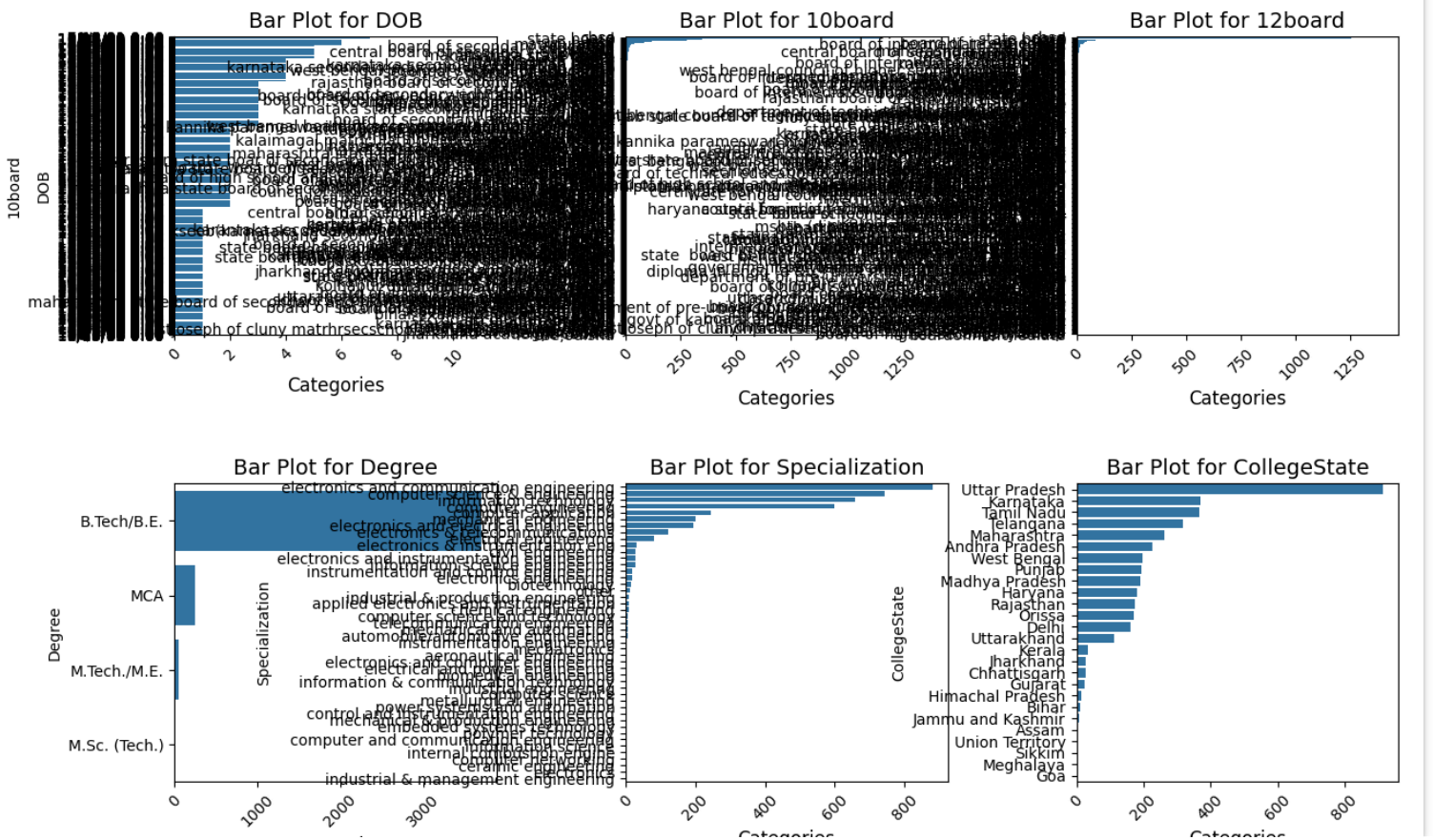
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

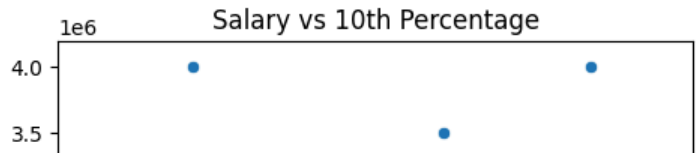


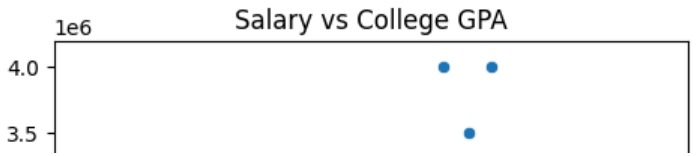

Bar Plot for DOB
10board
DOB
Categories
Bar Plot for 10board
Categories
Bar Plot for 12board
Categories
Bar Plot for Degree
B.Tech/B.E.
MCA
M.Tech./M.E.
M.Sc. (Tech.)
Degree
Bar Plot for Specialization
Specialization
Bar Plot for CollegeState
Uttar Pradesh
Karnataka
Tamil Nadu
Telangana
Maharashtra
Andhra Pradesh
West Bengal
Punjab
Madhya Pradesh
Haryana
Rajasthan
Orissa
Delhi
Uttarakhand
Kerala
Jharkhand
Chhattisgarh
Gujarat
Himachal Pradesh
Bihar
Jammu and Kashmir
Assam
Union Territory
Sikkim
Meghalaya
Goa
CollegeState

[ ]:
```
# Observations:
# Salary distribution shows outliers (right-skewed).
# Gender distribution is imbalanced.
# Specialization and Designation columns show that certain fields are more popular.
```

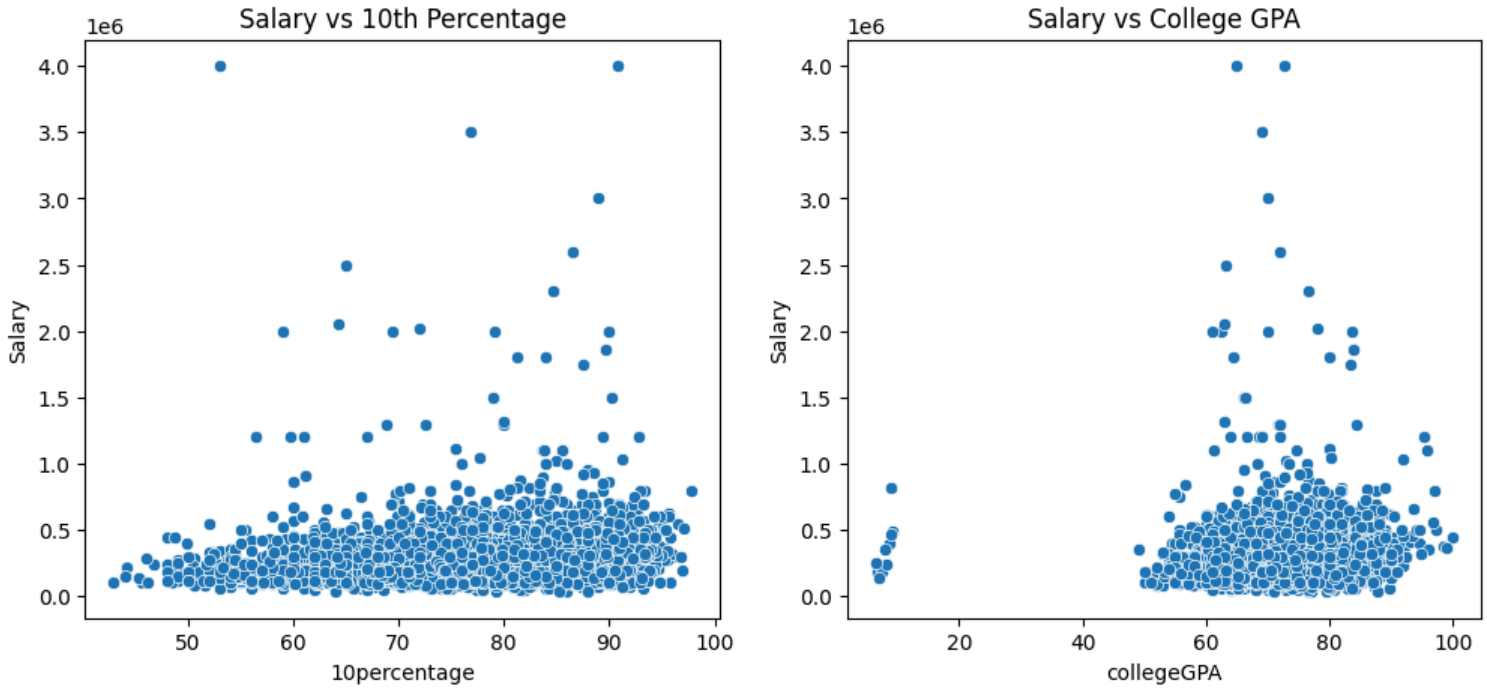
[ ]:
```
## Step -4 : Bivariate Analysis on Numerical and Categorical Columns
# Numerical vs Numerical: Using scatter plots, hexbin plots, and pair plots.
# Categorical vs Numerical: Using swarm plots, boxplots, and barplots.
# Categorical vs Categorical: Using stacked bar plots.
```

[97]:
```
# Numerical vs Numerical
# Scatter plot for Salary vs 10percentage and Salary vs College GPA
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.scatterplot(data=df, x='10percentage', y='Salary')
plt.title('Salary vs 10th Percentage')

plt.subplot(1, 2, 2)
sns.scatterplot(data=df, x='collegeGPA', y='Salary')
plt.title('Salary vs College GPA')

plt.show()
```





```
plt.show()
```



[103]:

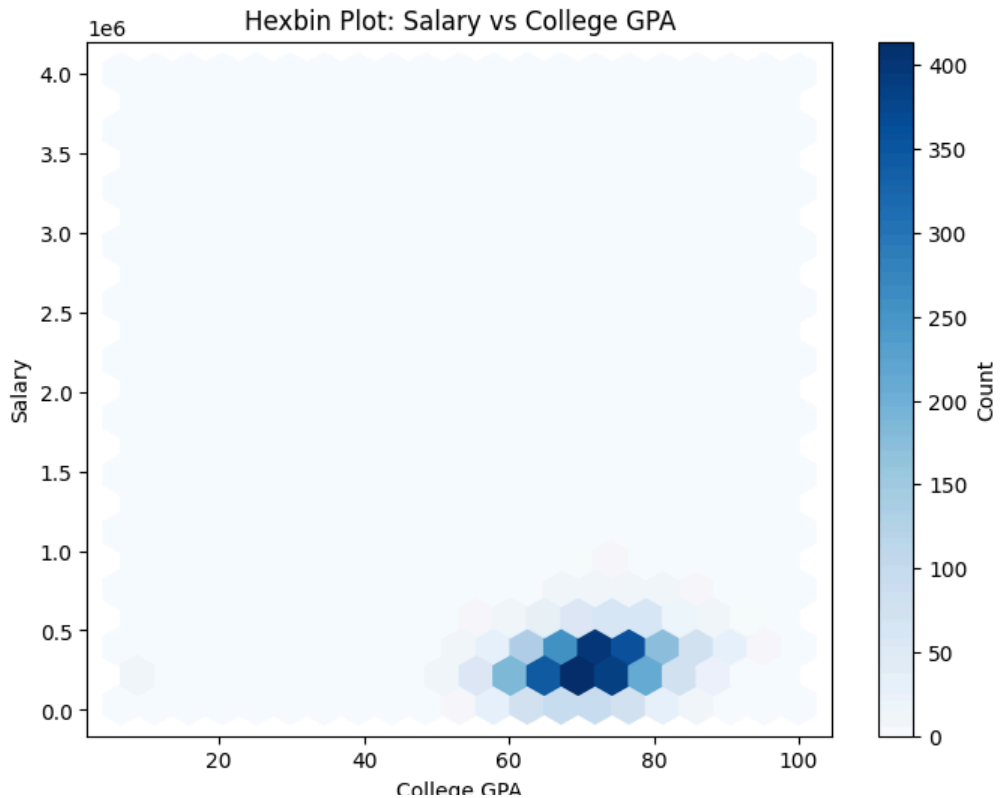
```
# Hexbin plot for Salary vs College GPA (Density based)
plt.figure(figsize=(8, 6))
plt.hexbin(df['collegeGPA'], df['Salary'], gridsize=20, cmap='Blues')
plt.colorbar(label='Count')
```

jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

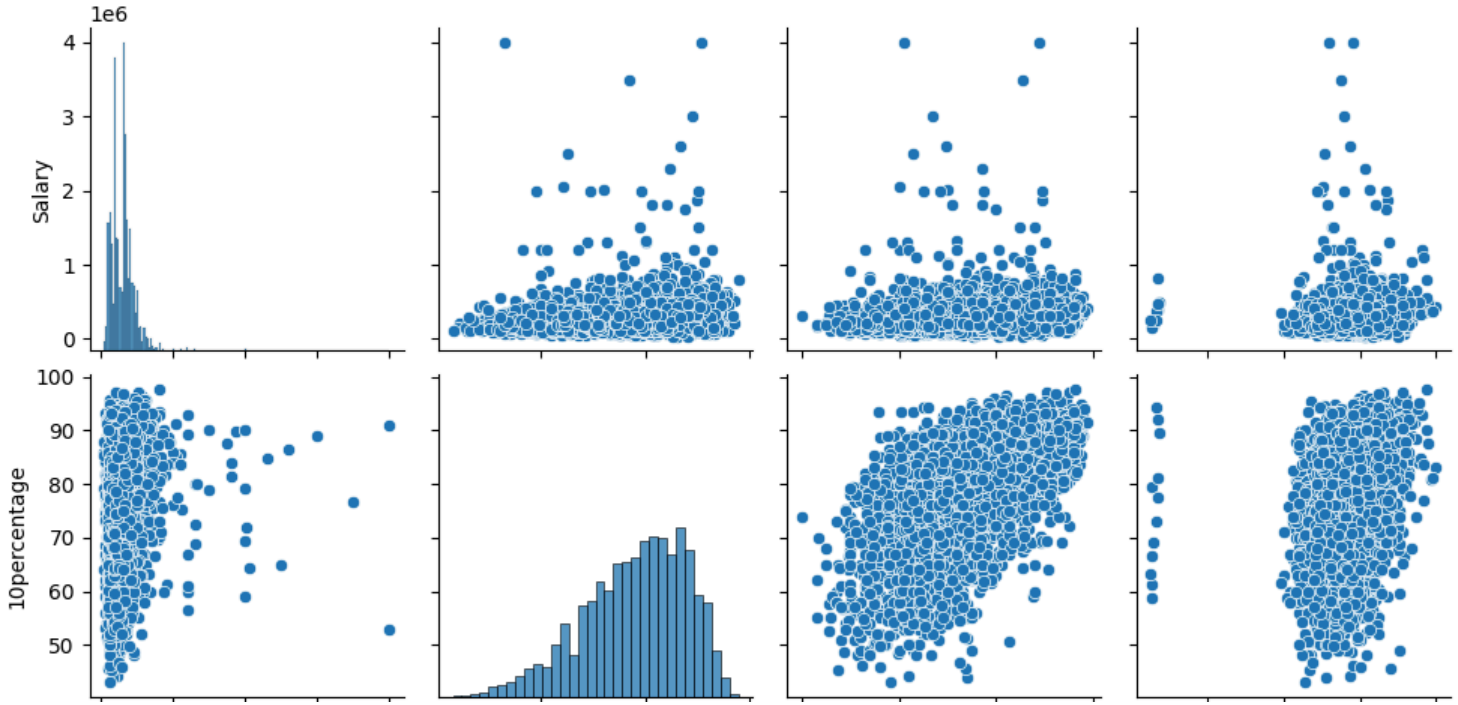
Code JupyterLab Python 3 (ipykernel)

```
plt.figure(figsize=(8, 6))
plt.hexbin(df['collegeGPA'], df['Salary'], gridsize=20, cmap='Blues')
plt.colorbar(label='Count')
plt.xlabel('College GPA')
plt.ylabel('Salary')
plt.title('Hexbin Plot: Salary vs College GPA')
plt.show()
```



College GPA

[107]:
```
# Pair plot for all numerical columns
sns.pairplot(df[['Salary', '10percentage', '12percentage', 'collegeGPA']])
plt.show()
```



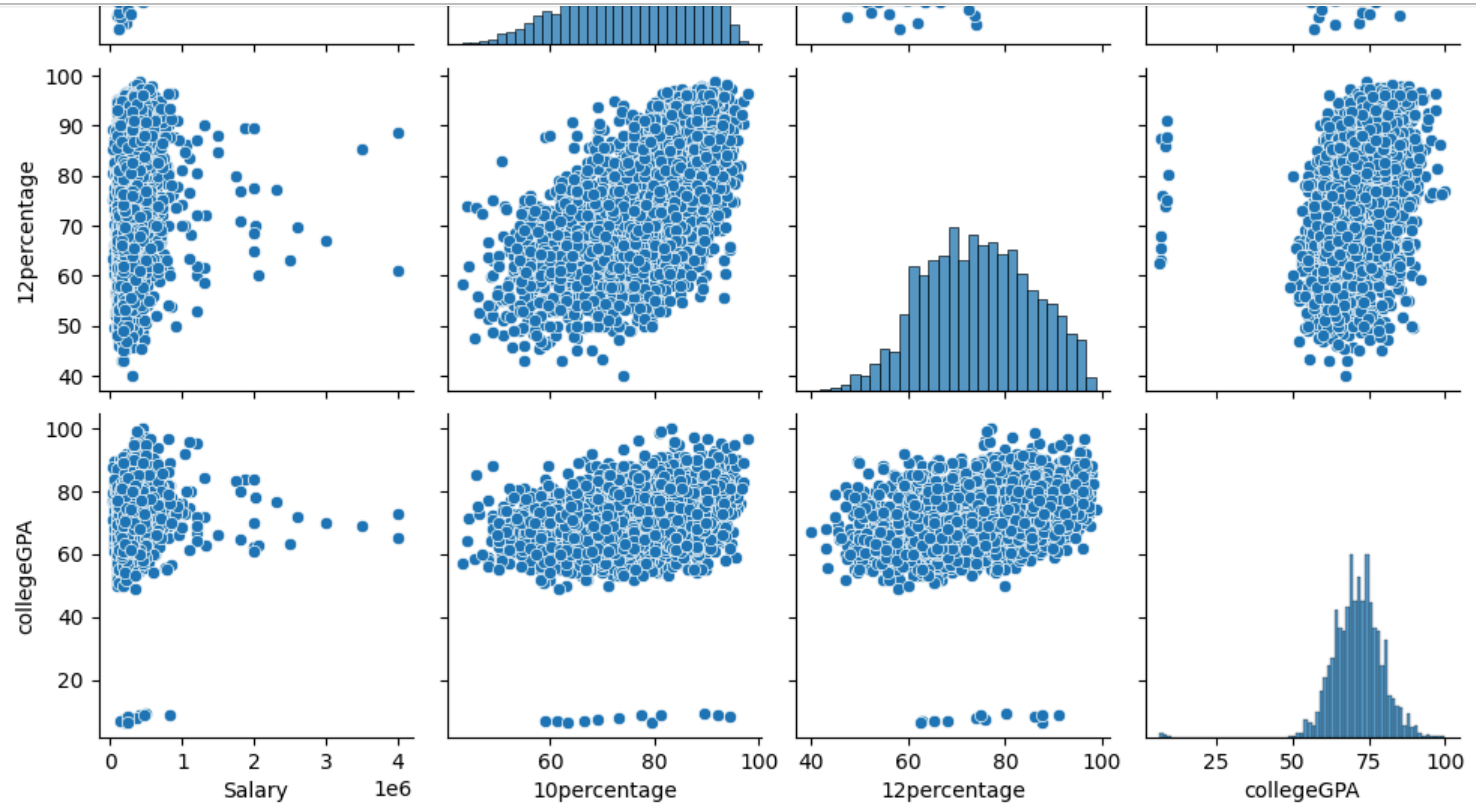
jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)



[109]:
```
# Categorical vs Numerical
```

jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

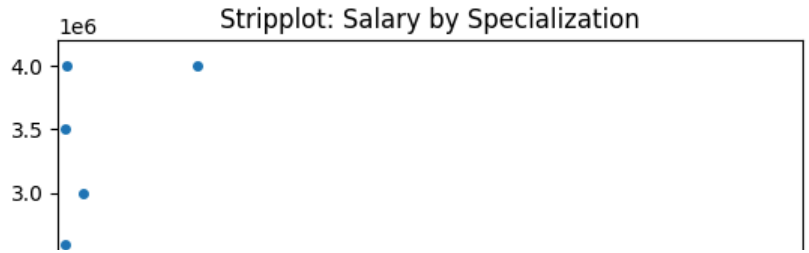
Code

JupyterLab Python 3 (ipykernel)

Salary 1e6 10percentage 12percentage collegeGPA

[109]:
```
# Categorical vs Numerical
```

[117]:
```
# Swarmplot for Salary by Specializationplt.figure(figsize=(12, 6))
sns.stripplot(data=df, x='Specialization', y='Salary')  # Use stripplot instead of swarmplot
plt.title('Stripplot: Salary by Specialization')
plt.xticks(rotation=90)
plt.show()

# Boxplot for Salary by Specialization and Gender
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.boxplot(data=df, x='Specialization', y='Salary')
plt.title('Boxplot: Salary by Specialization')
plt.xticks(rotation=90)

plt.subplot(1, 2, 2)
sns.boxplot(data=df, x='Gender', y='Salary')
plt.title('Boxplot: Salary by Gender')

plt.show()
```

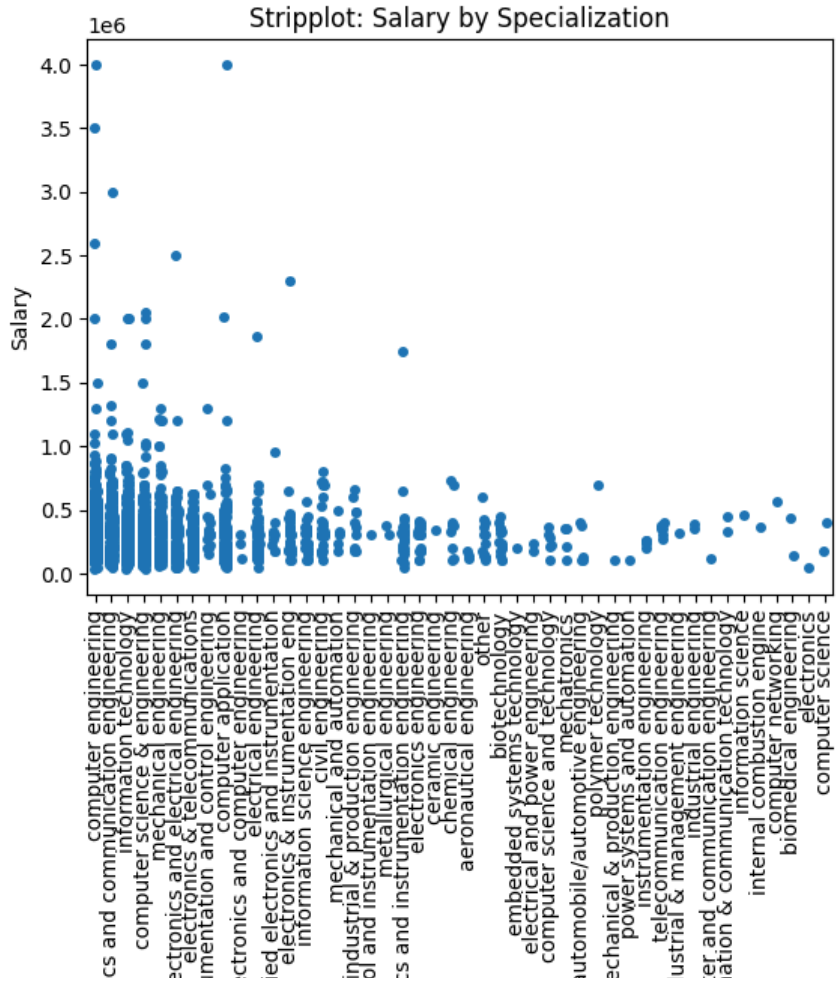


jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipykernel)

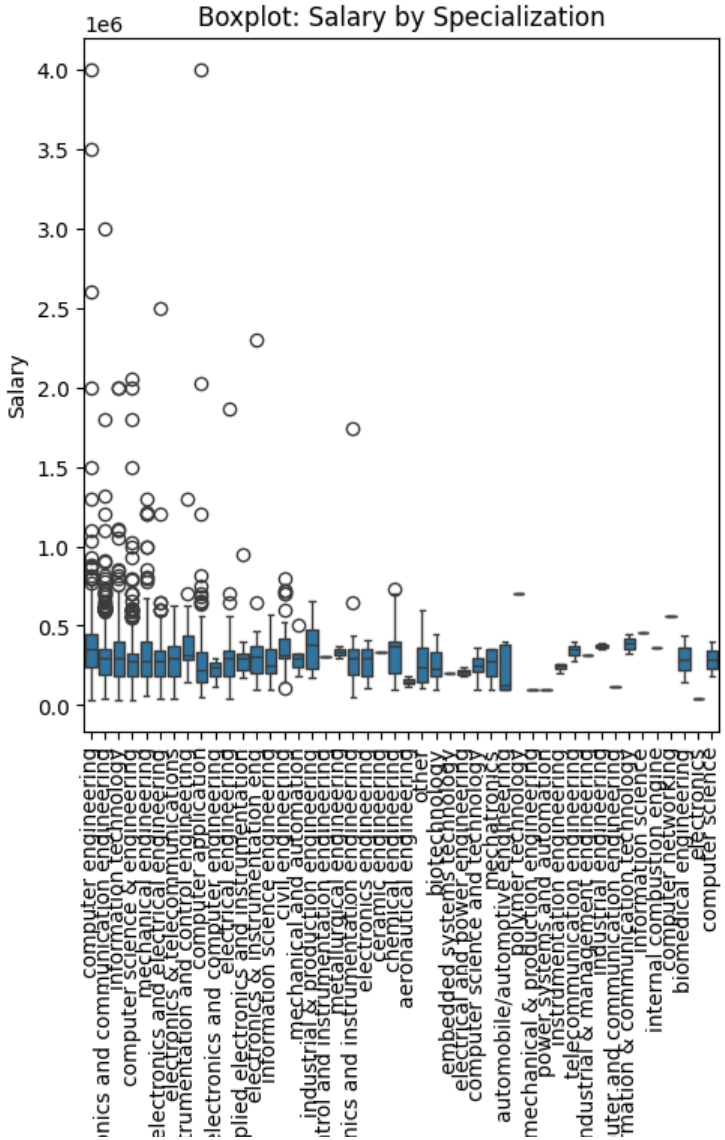
```
plt.show()
```

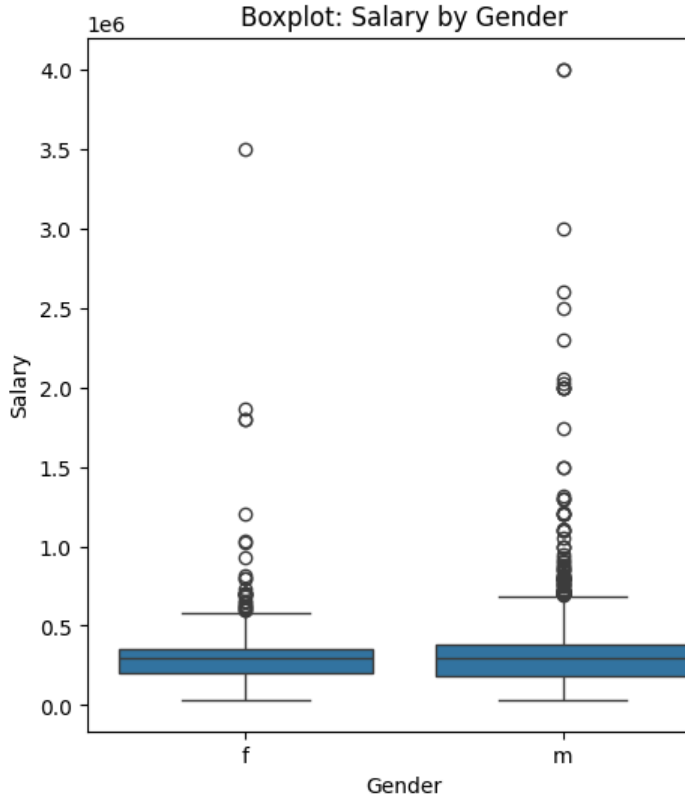


jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab P





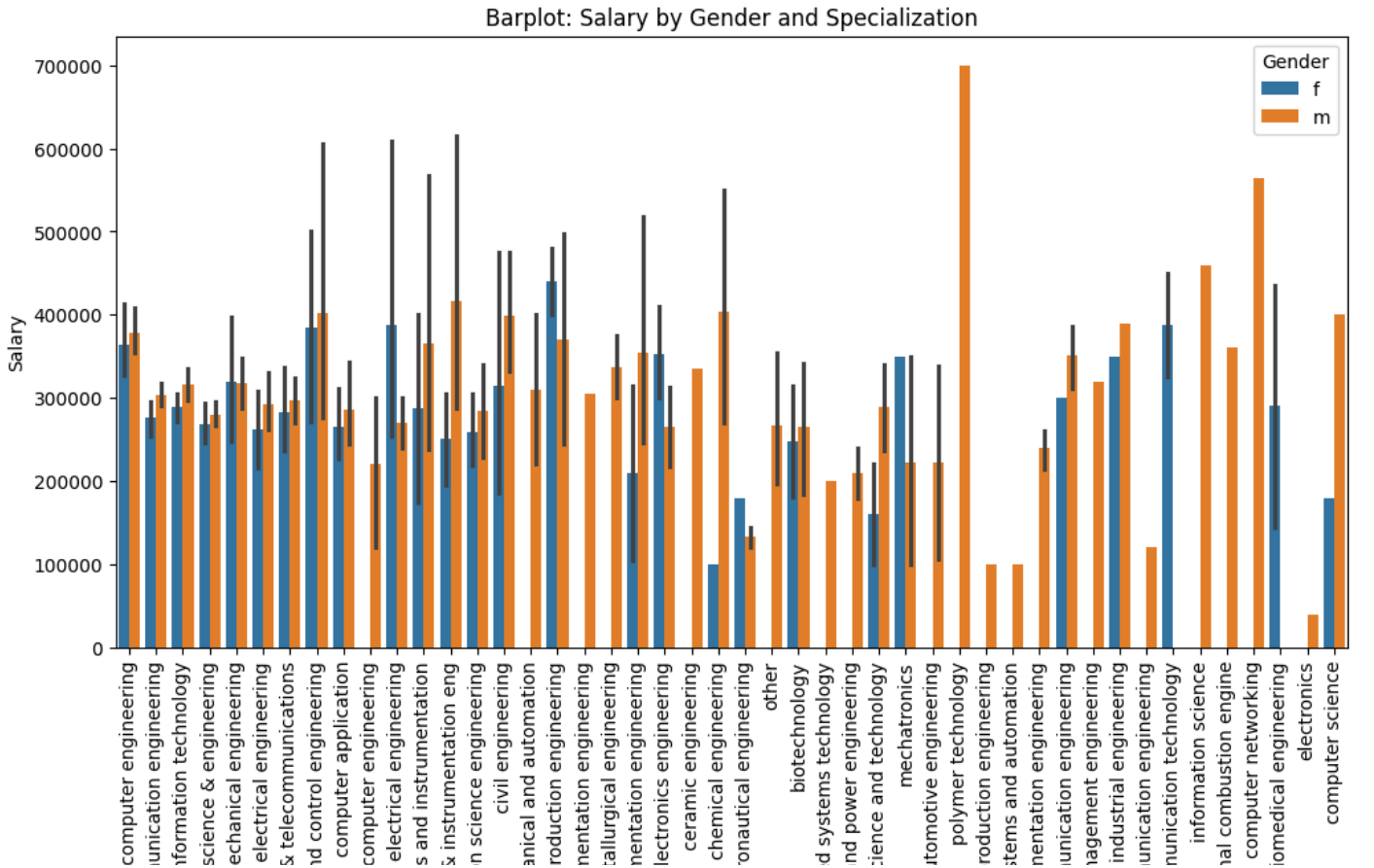
jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipyke

Specialization

[121]:
```
# Barplot for Salary by Gender and Specialization
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='Specialization', y='Salary', hue='Gender')
plt.title('Barplot: Salary by Gender and Specialization')
plt.xticks(rotation=90)
plt.show()
```



jupyter EDA on Salary Dataset Last Checkpoint: 2 hours ago

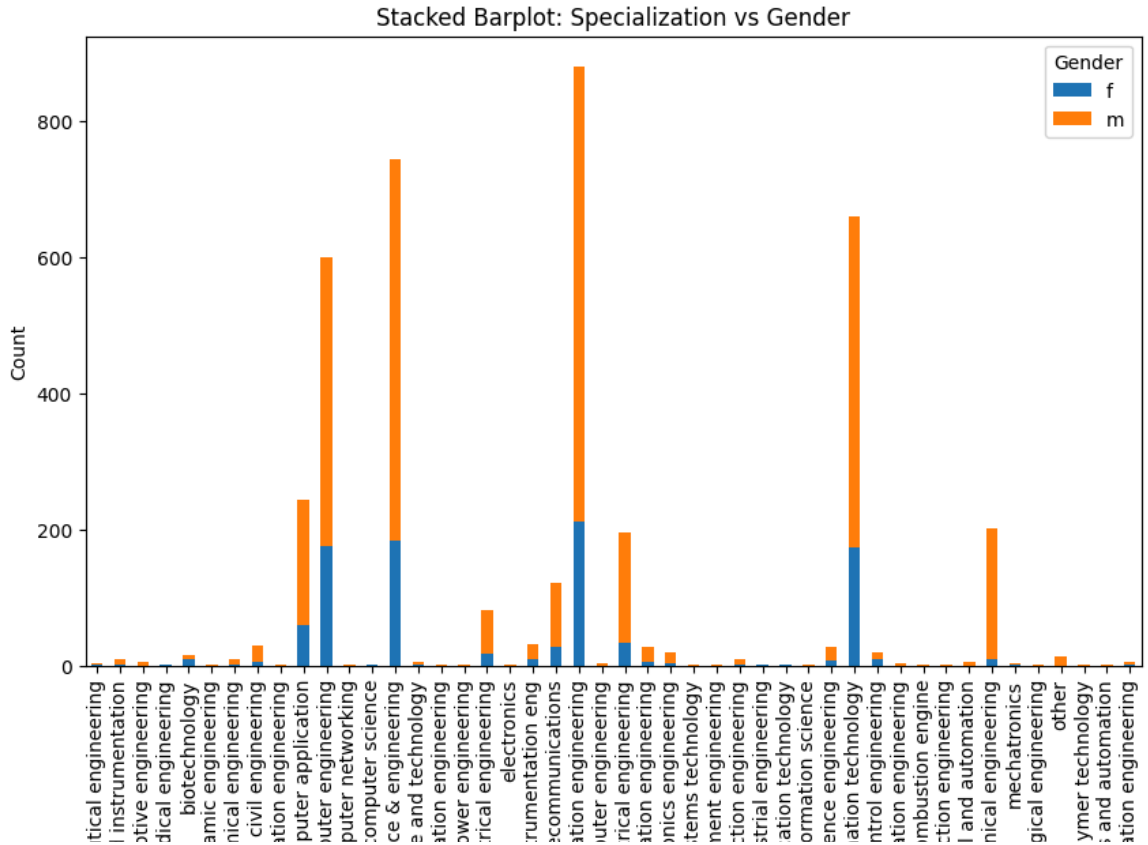
File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipyke

Specialization

[125]:
```
# Categorical vs Categorical
```

[131]:
```
# Relationship between Gender and Specialization using a stacked barplot
cross_tab = pd.crosstab(df['Specialization'], df['Gender'])
cross_tab.plot(kind='bar', stacked=True, figsize=(10, 6))
plt.title('Stacked Barplot: Specialization vs Gender')
plt.ylabel('Count')
plt.show()
```



[133]:
```
# Observations:
# Stacked Bar Plot: Shows that males dominate fields like Computer Science, while other specializations have a more even distribution of gender.
```

[135]:
```
# Step 5: Research Questions
```

[233]:
```
# Display all the column names
print(df.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')
```

[237]:
```
cs_grads = df[df['Specialization'] == 'Computer Science Engineering']

job_roles = ['Programming Analyst', 'Software Engineer', 'Hardware Engineer', 'Associate Engineer']
cs_grads_filtered = cs_grads[cs_grads['Designation'].isin(job_roles)]

# Define relevant job roles
job_roles = ['Programming Analyst', 'Software Engineer', 'Hardware Engineer', 'Associate Engineer']

average_salary = cs_grads_filtered['Salary'].mean()
print(f"Average Salary: {average_salary} Lakhs")

lower_limit = 2.5
upper_limit = 3.0

if lower_limit <= average_salary <= upper_limit:
    print("The average salary falls within the claimed range of 2.5-3 Lakhs.")
else:
    print("The average salary does NOT fall within the claimed range of 2.5-3 Lakhs.")
```

```
Average Salary: nan Lakhs
The average salary does NOT fall within the claimed range of 2.5-3 Lakhs.
```

[147]:
```
# To determine whether specialization depends on gender
```

```
Average Salary: nan Lakhs
The average salary does NOT fall within the claimed range of 2.5-3 Lakhs.
```

[147]:
```
# To determine whether specialization depends on gender
```

[151]:
```
from scipy.stats import chi2_contingency

# Create a contingency table for Specialization vs Gender
contingency_table = pd.crosstab(df['Specialization'], df['Gender'])

# Perform Chi-Square test of independence
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

# Display results
print(f"Chi-Square Statistic: {chi2}")
print(f"P-Value: {p_value}")

# Interpretation
if p_value < 0.05:
    print("Reject the null hypothesis - Specialization depends on Gender")
else:
    print("Fail to reject the null hypothesis - No relationship between Specialization and Gender")
```

```
Chi-Square Statistic: 104.46891913608454
P-Value: 1.2453868176977014e-06
Reject the null hypothesis - Specialization depends on Gender
```

[165]:
```
# Step - 6 - Conclusion


# **Key Findings:**
'''
- **Salary Distribution:** Median salary is below 3 lakhs for most fresh graduates.
```

```
'''
- **Salary Distribution:** Median salary is below 3 lakhs for most fresh graduates.
- **Specialization and Salary:** Computer Science and Information Technology fields offer higher salaries.
- **Gender and Salary:** Males generally earn more than females.
- **Gender and Specialization:** Gender influences specialization choice, with more males in Computer Science.

**Limitations and Recommendations:**

- Data limitations and potential biases exist.
- Further analysis could explore additional factors.
- Students should focus on technical specializations.
- Employers should ensure fair hiring practices.
- Policymakers should promote diversity in STEM fields.

These insights can inform career guidance and contribute to a more equitable job market.
'''
```

[165]:
```
'\n- **Salary Distribution:** Median salary is below 3 lakhs for most fresh graduates.\n- **Specialization and Salary:** Computer Science and Information Technology fields offer higher salaries.\n- **Gender and Salary:** Males generally earn more than females.\n- **Gender and Specialization:** Gender influences specialization choice, with more males in Computer Science.\n\n**Limitations and Recommendations:**\n\n- Data limitations and potential biases exist.\n- Further analysis could explore additional factors.\n- Students should focus on technical specializations.\n- Employers should ensure fair hiring practices.\n- Policymakers should promote diversity in STEM fields.\n\nThese insights can inform career guidance and contribute to a more equitable job market.\n'
```

[169]:
```
#Step - 7 - Bonus Research Question:
# What is the average salary of Computer Science graduates based on their gender?
```

[258]:
```
# Filter only Computer Science Engineering graduates
cs_grads = df[df['Specialization'] == 'Computer Science Engineering']

# Group by gender and calculate the average salary
avg_salary_by_gender = cs_grads.groupby('Gender')['Salary'].mean().reset_index()
```

```
avg_salary_by_gender = cs_grads.groupby('Gender')['Salary'].mean().reset_index()

# Display the average salaries
print("Average Salary by Gender:")
print(avg_salary_by_gender)

# Check for differences in average salaries
if len(avg_salary_by_gender) == 1:
    print("No data available for one of the genders.")
else:
    male_salary = avg_salary_by_gender.loc[avg_salary_by_gender['Gender'] == 'Male', 'Salary'].values
    female_salary = avg_salary_by_gender.loc[avg_salary_by_gender['Gender'] == 'Female', 'Salary'].values

    if male_salary.size > 0 and female_salary.size > 0:
        print(f"Average Salary for Males: {male_salary[0]} Lakhs")
        print(f"Average Salary for Females: {female_salary[0]} Lakhs")

        if male_salary[0] == female_salary[0]:
            print("No difference in average salaries based on gender.")
        else:
            salary_difference = abs(male_salary[0] - female_salary[0])
            print(f"Difference in average salaries: {salary_difference} Lakhs")
    else:
        print("No data available for one or both genders.")
```

```
Average Salary by Gender:
Empty DataFrame
Columns: [Gender, Salary]
Index: []
No data available for one or both genders.
```

[ ]:

```
Conclusion
Key Findings:
```

```
Average Salary by Gender:
Empty DataFrame
Columns: [Gender, Salary]
Index: []
No data available for one or both genders.
```

[ ]:

```
Conclusion
Key Findings:

Salary Distribution: Median salary is below 3 lakhs for most fresh graduates.
Specialization and Salary: Computer Science and Information Technology fields offer higher salaries.
Gender and Salary: Males generally earn more than females.
Gender and Specialization: Gender influences specialization choice, with more males in Computer Science.
Limitations and Recommendations:

Data limitations and potential biases exist.
Further analysis could explore additional factors.
Students should focus on technical specializations.
Employers should ensure fair hiring practices.
Policymakers should promote diversity in STEM fields.
These insights can inform career guidance and contribute to a more equitable job market.
```